

UNCLASSIFIED

AD NUMBER

ADB134931

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; AUG 1988. Other requests shall be referred to Air Force Geophysics Laboratory, Attn: PHG, Hanscom AFB, MA 01731-5000. This document contains export-controlled technical data.

AUTHORITY

AFGL ltr dtd 26 Dec 1989

THIS PAGE IS UNCLASSIFIED

DTIC FILE COPY

WARNING - This document contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751, et seq.) or The Export Administration Act of 1979, as amended. Title 50, U.S.C., App. 2401, et seq. Violations of these export laws are subject to severe criminal penalties. Disseminate in accordance with the provisions of AFR 80-34.

AFGL-TR-88-0163

DESTRUCTION NOTICE: For classified documents, follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX. For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.



The CRRES Langmuir Probe and Fluxgate Magnetometer Instrument

Peter R. Harvey

University of California
Space Sciences Laboratory
Berkeley, CA 94720

9 May 1989

Final Report
21 March 1983 - 31 December 1987

DTIC
SELECTED
AUG 4 1988
A D

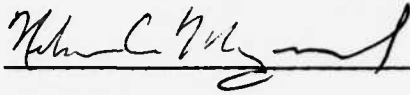
Distribution limited to US Government agencies and their contractors; Critical Technology; August 1988. Other requests for this document shall be referred to AFGL/PHG, Hanscom AFB, MA 01731-5000.

Air Force Geophysics Laboratory
Air Force Systems Command
United States Air Force
Hanscom Air Force Base, MA 01731-5000

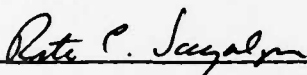
89 8 03 042

" This technical report has been reviewed and is approved for publication"

MICHAEL SMIDDY
Contract Manager


NELSON C. MAYNARD
Branch Chief

FOR THE COMMANDER


RITA C. SAGALYN
Division Director

Qualified requestors may obtain additional copies from the Defense Technical Information Center.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify AFGL/DAA, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFGL-TR-88-0163	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE CRRES LANGMUIR PROBE AND FLUXGATE MAGNETOMETER INSTRUMENT	5. TYPE OF REPORT & PERIOD COVERED Final 3/21/83 - 12/31/87	
	6. PERFORMING ORG. REPORT NUMBER N/A	
7. AUTHOR(s) Peter R. Harvey	8. CONTRACT OR GRANT NUMBER(s) F19628-83-K-0029	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Space Sciences Laboratory University of California Berkeley, CA 94720	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62101F 760112AM	
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratory Hanscom AFB, MA 01731 Contract Manager: Michael Smiddy/PHG	12. REPORT DATE 5/9/89	
	13. NUMBER OF PAGES 400	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Resident Representative University of California at Berkeley Office Richmond Field Station Richmond, CA 94804-0001	15. SECURITY CLASS. (of this report) Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
16. DISTRIBUTION STATEMENT (of this Report) Distribution limited to US Government agencies and their contractors; Critical Technology; August 1988. Other requests for this document shall be referred to AFGL/PHG, Hanscom AFB, MA 01731-5000.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) CRRES, Langmuir Probe Instrument; Electric Field Instrument; Fluxgate Magnetometer; Hardware; Software. (jhd)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A description of the CRRES Langmuir Probe Instrument is given in three parts. The first part is a general description and contains sensor infor- mation, telemetry and command formats, and top-level capabilities of the instrument. The second and third sections describe the hardware and soft- ware designs. A software listing is included in the appendix. <i>Keywords:</i>		

Contents

	<u>Page</u>
1. General Information.....	1
1.1 Sensor Description.....	2
1.2 Instrument Interfaces.....	7
1.3 Analog Electronics.....	8
1.4 Digital Electronics.....	17
2. Software.....	29
2.1 Main Executive Module.....	31
2.2 Background Management.....	34
2.3 Electric Field Management.....	46
2.4 Magnetic Field Management.....	53
2.5 Plasma Instrument Data Management.....	56
2.6 Burst Processing Management.....	61
2.7 Spin-Fitting Management.....	68
2.8 Spin-Fitting Computations.....	73
2.9 Sawtooth Generation.....	80
2.10 Bias Sweeps.....	84
2.11 Main Program Loader.....	96
2.12 Boom Deployment.....	98
2.13 General Utilities.....	101
2.14 Main Input/Output Utilities.....	102
2.15 Fast Floating Point Utilities.....	106
2.16 Matrix Utilities.....	109
2.17 Trigonometric Utilities.....	110
2.18 Burst Executive Module.....	111
2.19 Burst Input/Output Utilities.....	113
2.20 Burst Format Control.....	116
2.21 Burst Program Loader.....	118

Contents

	<u>Page</u>
2.22 Burst Sampling Module.....	119
2.23 Burst Compiler.....	123
2.24 Burst Floating Point Utilities.....	126
3. Hardware.....	127
3.1 The Analog Board.....	128
3.2 The Filter Board.....	133
3.3 The Input/Output Board.....	136
3.4 The Main Processor Board.....	143
3.5 The Burst Processor Board.....	154
3.6 The Memory Boards.....	166
3.7 The Power Converter.....	168
3.8 Inter-Processor Communications.....	169
4. Contributing Engineers and Scientists.....	174
Appendix A. Flight Software Listing.....	175
Appendix B. Block Diagrams.....	390



<u>Figure</u>	Illustrations	<u>Page</u>
1.	Synchronous Sampling Format.....	35
2.	General Telemetry Format.....	36
3.	Fluxgate Magnetometer Telemetry Format.....	37
4.	Digital Subcommutator Telemetry Format.....	38
5.	Electric Field Telemetry Buffering.....	48
6.	Electric Field Quantity Lists.....	49
7.	RAM Quantities.....	50
8.	Burst Playback Telemetry Format.....	67
9.	Spin-Fit Result Telemetry Format.....	70
10.	Spin-Fit Difference Function.....	77
11.	Spin-Fitting Matrices.....	78
12.	Input/Output Data Blocks for Spin Fitting.....	79
13.	Orientation of Booms During Bias Sweeps.....	88
14.	Bias Output Timing for Spheres.....	89
15.	Bias Output Timing for Cylinders.....	90
16.	Bias Sweep Analysis.....	91
17.	Relay Set and Reset Logic.....	131
18.	Extended Relay Set and Reset Logic.....	132
19.	Filter Board Control Register Definition.....	141
20.	Analog Board Control Register Definition.....	142
21.	Burst A/D Pipeline Operation.....	163
22.	Burst to Main Communication Timing.....	171

Tables

<u>Table</u>		<u>Page</u>
1. Summary of Physical Attributes.....		6
2. Programmable Filter Response Characteristics...		12
3. Multiplexor Settings.....		13
4. Bias, Stub and Guard Characteristics.....		16
5. Burst Default Sampling Formats.....		26
6. Burst Frequencies.....		28
7. Main and Burst Software Modules.....		30
8. Module Sampling Frequencies.....		34
9. Fast Floating Point Execution Times.....		108
10. Filter Rolloffs.....		135
10. Main Multiplexor Quantities.....		172
11. Burst Multiplexor Quantities.....		172
12. Analog to Digital Conversion Times.....		173

1. General Information

The AFGL-701-14 Langmuir Probe Instrument on CRRES has two separate functions, namely to act as 1) a combined Langmuir-Probe/Electric-Field instrument and 2) as the control element for the Fluxgate Magnetometer (AFGL-701-13). This chapter describes the general capabilities of the instrument while later chapters describe the software and hardware in more detail.

The electric-field/langmuir probe part of the instrument connects to two orthogonal double probes, each of which is a pair of separated conductors whose potential difference is measured. One pair of separated conductors are spheres that are located in the spin plane on the ends of wire booms and separated by 100 meters. The other pair of conductors are cylindrical wire boom elements also in the spin plane that are separated by an effective distance of 90 meters.

The fluxgate magnetometer part of the instrument consists of a 3-axis magnetometer sensor which is attached to the end of a 20' rigid boom in the spin plane. The sensor has a plus or minus 45000 nT range and sensitive to less than 1/2 nT.

Both parts of the instrument are controlled by a central computer which telemeters to the ground approximately 96 Langmuir Probe/Electric Field samples and 48 Magnetometer samples per second.

In addition, the instrument coordinates with two other instruments on the spacecraft. The electric field analog signals are repeated to both an AC electric field instrument and a plasma analyser, and magnetic field data is sent in reduced digital form to the plasma analyser.

1.1 Sensor Descriptions

Wire Booms with Spherical Sensors (AFGL-701-14D and 14E)

The spherical sensors consist of opposing wire booms that are positioned by centrifugal force. Each boom has four major components: the spherical sensor, "stub" and "guard" segments, the cable and the deployment hardware.

The sensor consists of an aluminum sphere which is coated with a conductive material. Inside this shell is a small circuit board containing circuits for both voltage and current measurements.

The cable consists of 8 conductors surrounding a coaxial conductor. These wires are surrounded by a stainless steel braid which acts as an electrically conducting outer shield. The conductors feed voltages and accept signals from the preamplifiers located in the spheres. The mechanical member which supports the centrifugal force load of the probes is stranded Kevlar that is located between the wires and the braid. The wires and outer shield connect electrically at the sensor sphere.

The cable's outer shield is broken into three sections. The section closest to the sphere, which is called the STUB section, is electrically connected to a voltage equal to the preamplifier output plus or minus a small DC offset so as to force its potential to be near that of the sphere, and, thus, to minimize the perturbing effect of the cable on the plasma. To guard against the possibility of oscillations being set up by this arrangement due to resonances in the plasma, it is possible to

insert a low pass RC filter with 100 Hz rolloff between the preamplifier and the stub section by actuating a latching relay in the main electronic box.

To prevent a positively charged spacecraft from attracting electrons away from the sensors, a small section (10cm), called the GUARD, is placed between the shield and the stub sections. The guard section is adjusted to be more negative than the sphere and stub sections and its potential with respect to the sphere is controlled by the microprocessor via ground command.

Stubs and guards exists symmetrically on both sides of each sphere, with the outer stub and guard being restrained at launch via a "tophat" mechanism. Basically, the outward cable segment is wrapped inside the "hat" while the "brim" of the hat holds onto the sphere. When the centrifugal force on the tophat exceeds its ability to hold on, the tophat releases and the cable unravels as the tophat floats away.

The deployment units incorporate two methods of measuring boom length as well as microswitches to reveal both sphere-release and end-of-wire conditions. One method of length measurement involves a simple potentiometer tied to the cable spool, while the second method uses a microswitch tied to cam on the cable feed roller.

Boom deployment is accomplished under microprocessor control that monitors the boom lengths and temporarily stops the deployment of any unit whose length differs from that of its mate by more than a few inches. As a backup in case of microprocessor failure in launch, the boom deployment can be accomplished

directly by the spacecraft control system.

Wire Booms with Cylindrical Sensors
(AFGL-701-15D and 15E)

Each cylindrical sensor unit is driven by a 28-volt brushless DC motor which powers both a storage reel and a drive roller assembly. The driver roller incorporates a slip clutch which allows the roller to be driven at a slightly faster rate than the storage spool. This feature ensures that a positive tension is maintained on the wire while being dispensed from the mechanism. A potentiometer is driven by the moving wire to give a continuous indication of deployed length. Microswitches are used to signal the full extension point. In addition, at full extension, a lever trips another microswitch which cuts power to the motor to provide automatic shut-off of the mechanism. This function is backed up by a positive mechanical stop to prevent the possibility of backwrapping the wire through switch failure.

The deployment of cylindrical boom units is under direct spacecraft control, not instrument control.

Electric Field measurements using the cylindrical units involve attaching small preamplifier circuits near the base of each antenna unit. Each preamplifier box is then connected to the main electronics box.

DC Preamplifiers for Cylindrical Booms
(AFGL-701-14B and 14C)

The DC measurement of cylindrical sensors is accomplished by means of preamplifier circuits located close to the base of the cylinder deployment mechanisms. Bias current to the sensors is supplied by means of a relay inside the DC preamplifier unit. This relay connects a bias voltage to the sensor through a large valued resistor.

Shonstedt Fluxgate Magnetometer
(AFGL-701-13-1A, 1B, 2)

The magnetometer measurement is made using a standard fluxgate instrument of which there are two units: a main electronics box and sensor. The sensor is mounted on a rigid 20' long boom in order to get it far enough away from the spacecraft body that the total spacecraft generated magnetic field will have a strength less than 2 nT along the spin axis and 4 nT in the spin plane at the sensor location. The sensor is oriented such that the outputs called BX, BY and BZ are in the spacecraft coordinate system -X, -Z, and -Y. The sensor is also tipped slightly in order to give a spin frequency waveform in the other measurements. This allows for the spacecraft z-axis offset to be calculated.

Table 1. Summary of Physical Attributes

BOX DESCRIPTION	DIMENSIONS (in)	WEIGHT (lbs)	AVG. POWER (W)
13-1a:Fluxgate Elect.	5.6 x 6.5 x 2.1	1.5	.500
13-1b:Sensor	4.7 x 2.8 x 2.8	.7	.050
14A :Electronics	10.5 x 10.5 x 5.0	16.0	8.250
14B :DC Preamp	4.0 x 2.4 x 1.3	0.6	.125
14C :DC Preamp	4.0 x 2.4 x 1.3	0.6	.125
14D :Spherical Boom	14.8 x 6.5 x 7.5	8.2	.500
14E :Spherical Boom	14.8 x 6.5 x 7.5	8.2	.500
15D :Cylindrical Boom	7.5 x 12.2 x 7.3	5.6	----
15E :Cylindrical Boom	7.5 x 12.2 x 7.3	5.6	----

Notes:

1. 14D&E weights include 50 meters of wire and spherical sensor.
2. 15D&E weights include 50 meters of wire.
3. Peak power of 14A is 10.5 Watts (Burst collecting).

1.2 Instrument Interfaces

The 701-14 instrument has a number of interfaces which share signals and data with other experiments. These are described below:

IOWA SOUNDER INTERFACE. The IOWA Sounder is an AC electric field instrument capable of frequencies up to several hundred thousand Hertz. Two analog signals are sent from the -14A box to the IOWA instrument (701-15A). In the voltage mode, these are the Voltages on spheres 1 and 2, while in the current mode these are the current on sphere 1 and ground.

The IOWA instrument incorporates a search coil magnetometer, whose signal is buffered to the -14A box. This signal is available to the Burst computer system only.

LEPA INTERFACE. The Low Energy Plasma Analyser (LEPA) interface consists of a few open collector digital lines which are used to communicate reduced magnetometer information from the Langmuir Probe to the LEPA instrument. This data points out the loss cone to the LEPA instrument, so that it can take high resolution samples in this region.

SPACECRAFT INTERFACE. The spacecraft interface consists of both digital and analog wires. Timing signals provided include telemetry shift clocks at 16 KHz, a 2 KHz clock (8-bit telemetry word timing), and a major frame spike every 4.096 seconds. Data is shifted out in 16-bit packages. The 2 KHz clock is used to maintain timing in the MAIN and BURST computers.

Commands are shifted into the instrument in 16-bit packages using a shift clock and envelope signals. The envelope is wired

into a MAIN computer interrupt to provide command ready information.

A sun pulse is also provided by the spacecraft electronics. Using this signal and the 2 KHz clock, the MAIN computer continuously calculates the spacecraft sun angle for use by internal functions such as spin-fitting and bias sweeping.

Analog and bi-level monitors of the instrument health are provided through the main spacecraft interface.

1.3 Analog Electronics

The following is an overview of the analog electronics in the Langmuir Probe instrument (701-14A). For more details, refer to the hardware description in chapter 3.

E-field Sensor Interfacing

The instrument has four main electric field sensors, namely the two spheres and the two cylinders. The spheres are capable of being operated in two modes, one which measures electric fields and the other which measures the current.

The electronics which measure the sphere voltages have a bandwidth of about 1 MHz and the differential signal dynamic range of +/- 200 Volts. This dynamic range is achieved by operating +/- 12 volt preamplifiers from floating power supplies. The output voltage of the sensor is used to driver a unity gain preamplifier which operates from a +/- 100 Volt supply that is referenced to the experiment ground. The output of this supply drives the common terminals of its floating preamp supplies. This circuit allows the sensor potentials to swing plus or minus 100 volts with respect to the spacecraft. To reduce power

consumption, the large signal bandwidth is limited to 1000 Hz.

The cylinder measurement electronics differ significantly from the above by operating from fixed +/- 35 Volt supplies rather than from floating power supplies. This limits the dynamic range of the cylinder measurement to about +/- 33 Volts.

These four sensor measurements are called V1 thru V4 and are available to be digitized by both the MAIN and the BURST computer systems (see the digital electronics below).

Sensor Differencing

The main measurement of the electric field is the difference in voltage between the two spheres and the two cylinders. By convention, the voltage difference between V1 and V2 is called V12 and the difference between V3 and V4 is V34.

Difference Amplification

While a "times 50" amplifier is available to any quantity on the MAIN multiplexor, backup "times 50" amplifiers are provided for V12 and V34. These signals are called V12H and V34H on the MAIN multiplexor.

Differencing Trims

The difference measurements V12 and V34 are trimmable by the operation of a pair of DAC's so that offsets which occur as the result of radiation damage may be adjusted so the measurement stays in the range of the high gain amplifier (see the A/D section below). To adjust these trims use the following commands:

.VTRIM n xx

where n is 1 for V12 trim, and 2 for the V34 trim,
and xx is a 2's complement 8-bit value (+127 to -128).

Band Filters

The V12 signal is fed into a bank of 3 bandpass filters with center frequencies at 32, 256 and 2048 Hz. The filters are called F1, F2 and F3 and are connected to the main multiplexor only.

Each of these bank filters consists of a 2-pole bandpass function followed by a logarithmic amplifier, a full wave rectifier and an integrator. The overall response of the filter bank constant in the frequency range from 32 to 2048 Hz; i.e. the filter band widths are made sufficiently large that there is no loss of signal with frequencies between the center frequencies of the filters. The result of this is to provide for each of the frequency ranges a voltage which is proportional to the logarithm of the power in that frequency range.

Magnetic Field Measurements

The fluxgate magnetometer interface converts the +/- 10 Volt signal from the Shonstedt unit into +/- 5 Volts for the A/D

circuitry of the Langmuir Probe. At the same time, these signals are rolled off at 6 Hz for the MAIN telemetry sampling system.

The additional capability is that of amplifying the BY signal from the magnetometer by a factor of 6. BY is in the spacecraft Z axis and may at times be very small. To select this amplification, use the command

```
.BMODE x
```

where x = 1 turns ON the amplification

and x = 0 turns OFF the amplifier.

The instrument defaults to OFF (BY not amplified).

Filtering Electronics

There are two basic types of anti-aliasing filters used by the Langmuir Probe instrument, fixed and variable. The fixed filters are used on nearly all quantities which are fed into the MAIN analog multiplexor since the sample frequencies are pretty much dictated by the telemetry capabilities.

The variable filters are used for quantities which are fed into the BURST computer system. Since the Burst system can vary its sample frequency from 10 Hz to 62500 Hz, the completely general anti-aliasing filter should optimally be tunable from 5 to 31250 Hz. Using knowledge about what was reasonable to expect for sampling frequencies for each quantity, the filters were set as shown in the table below.

To set the value of a given filter, use the command as follows:

```
.FILTER n xx
```

where n is the filter number (1 thru 7) and "xx" is a value

between 1 and 255. The value 1 provides the maximum rolloff while 255 opens the low pass filter to its maximum value.

Note: Do not use a filter value of 0! The filters do not work when programmed with 0. The output simply saturates.

Table 2. Programmable Filter Response Characteristics

FILTER	SIGNAL	BURST NAME	MAX(255)	MIN(1)
1.	V12/RI1 AC	BV12AC	11.5 KHZ	45 HZ
2.	VS2/RI2	BV2	11.5	45
3.	V12/RI1	BV12	25.5	98
4.	DIRECT AC	BDIRECT	11.5	45
5.	VS1/-SC	BV1SC	15.0	59
6.	V34	BV34	18.0	70
7.	V34 AC	BV34AC	12.0	47

Filters corresponding to Burst multiplexor quantities and the names on the block diagram.

Multiplexing

In order to maximize capabilities of the instrument using only 7 filters, a number of multiplexors were added to provide options as to which value should be filtered and then sampled. These multiplexors are responsible for some quantity names to have a "/" in them. For example, the quantity "V1/SC" is either V1 or SC (Search Coil) depending upon the setting of multiplexor 3. To set these multiplexors, use the command as follows:

```
.MUX n x
```

where n is the multiplexor (0 thru 3) and x is the value.

Table 3. Multiplexor Settings

MUX	QTY	x=0	x=1	x=2	x=3
0	V12/RI1	RI1	V12		
1	V2/RI2	RI2	V2		
2	KAGC	V2/RI2	V12/RI1	SC	V34
3	V1/SC	SC	V1		

Note:

1. The multiplexor setting is available in the DSC data.
2. MUX 0 and MUX 1 are operated when changing from the Voltage to the Current mode and vice versa. (See Mode Switching.)

Relay Control

A number of the options of the sensor measurement and control electronics are implemented using relays as the switching elements. These switches are shown on the instrument block diagram. To set or reset relay number "n", use the command

.SET n or .RESET n

It is important to note:

1. Relays 0 and 1 are used to "steer" current for setting and resetting other relays, so their state may change when operating other relays.
2. Relays 0, 1, 7, 8, 9, 16, 18 and 19 are changed by the switching from the Voltage to the Current mode or vice versa. See Mode Switching.

Sensor Bias Electronics

An important capability of the instrument is that of applying bias currents to the sensors. The impedance between the sensor and the plasma is a non-linear function of the current flowing between them and it exhibits a minimum at a value of bias current which depends upon the plasma conditions. Thus, the accuracy of the electric field instrument can be maximized by applying the optimum value of bias current to the sensor.

The analog circuitry which accomplishes sensor, stub and guard biasing consists of eight 8-bit DACs which produce a bipolar effect upon the eight biasing circuits (2 spheres, 2 guards, 2 stubs and 2 cylinders). The value of the sensor bias currents are set either by ground command or by on-board algorithms.

The sensor bias voltages are connected to the sensors through 100 MegaOhm resistors. Ground commanded relays can be operated to remove the biasing capability for the cylinders only. The spheres are always biased when in the voltage mode.

To set any of these DAC's, use the appropriate command:

.BIAS n xx

.STUB m xx

.GUARD m xx

where n is the sensor number (1 thru 4),

m is the sphere number (1 or 2),

and xx is a 2's complement 8-bit value (+127 to -128).

The voltages output are shown in Table 4.

Table 4. Bias, Stub and Guard Characteristics

DAC	-128	+127	GAIN	OFFSET
BIAS1	-35.31	35.50	.2777	.2345
BIAS2	-35.24	35.55	.2776	.2960
BIAS3	-35.23	35.45	.2772	.2515
BIAS4	-35.19	35.58	.2776	.3360
STUB1	- 1.21	1.25	.0096	.0210
STUB2	- 1.23	1.22	.0096	-.0030
GUARD1	-35.20	35.41	.2769	.2415
GUARD2	-35.33	35.47	.2776	.2075

Analog to Digital Conversion

Selected potential difference measurements as well as analog outputs of filter banks, the potentials of each sensor, and other analog quantities such as boom lengths, motor currents, etc. are fed through two multiplexors, one for telemetry sampling by the MAIN computer and one for high rate sampling by the BURST computer.

Each multiplexor is followed by a pair of op amp circuits, one with unity gain and the other with a gain of about 50. The outputs of these circuits are then fed into a final multiplexor with which the processor can select one signal for digitization. The gain decision for the MAIN computer is performed in software by actually digitizing the low gain value and then re-digitizing either the high or low gain value. On the BURST computer system, the gain decision is made automatically by comparators whose outputs drive the last multiplexor.

The output of these final multiplexors go to fast 12-bit A/D converters, one for each computer system.

The computed gain of the "times 50" amplifiers is 51.12 and -49.75 for the MAIN and BURST systems, respectively.

1.4 Digital Electronics

General

The instrument digital section consists of two microprocessor systems arranged in a master-slave relationship. The master processor, called the MAIN processor, is responsible for most of the mission operations, namely telemetry formatting, command reception and execution, sensing burst conditions, boom deployment, current sweeps and other control items. The slave microprocessor, called the BURST, is responsible for high frequency data sampling and storage.

The MAIN system consists of a SANDIA 3000 microprocessor which is a radiation tolerant version of the Intel 8085 only in CMOS. On the system buss is 8K bytes of ROM and 4K of RAM, plus a host of input and output ports, of course.

The BURST system also uses a SANDIA 3000 processor but has only half the amounts of ROM and RAM in which to store programs; i.e. 4K bytes of ROM and 2K of RAM. In addition to the normal RAM, the BURST system includes a memory unit of 192K bytes used for storing bursts of digitizations which it later plays back to the MAIN system on request.

Mode Control

The Main computer controls the mode of the spherical sensors in two ways, either automatically or manually. The mode of these sensors appears in the Fast Digital Monitor data.

The executive will automatically switch modes back and forth at programmable time intervals (measured in the number of spacecraft rotations). These intervals are selected using the command:

```
.EMODE n m
```

This instructs the instrument to operate for $2^{*(n-1)}$ spins of the current mode and $2^{*(m-1)}$ spins of the voltage mode. If either n or m is zero, the corresponding mode is not used. If $n = m = 0$, mode switching is disabled. For example, to select voltage mode only use ".EMODE 0 1". The default is ".EMODE 7 7" which means the instrument will flip modes every 64 spins (32 minutes).

The executive changes the modes only at the beginning of a spin period, phased with the spin-fitting software so that no data is lost. The first mode change occurs at the first spin period boundary following the EMODE command entry.

Two vestigial commands worth noting are as follows:

```
.VMODE    selects the Voltage mode and  
.IMODE    selects the Current mode.
```

As soon as these commands are entered, the instrument will flip all the necessary relays and multiplexors in order to configure itself for that mode. However, these commands control neither the current mode sawtooth nor the spin-fit calculations (both of these ARE controlled in the automatic mode). Hence, if

you ask for the IMODE, you should also enable the sawtooth using SAW commands. And if you ask for the VMODE, you probably will want to enable the spin-ftting.

The MAIN computer can configure the instrument such that the health of nearly all of the sensor electronics can be determined. This automatic test sequence can be invoked using either the ".TEST" or ".CALIBRATE" commands. The sequencer begins at the start of a telemetry major cycle (every 32 seconds) and lasts about a minute. Simultaneous measurements of V1 thru V4, their voltage differences, the BIASing circuitry and so forth are recorded by both the MAIN and the BURST circuitry. The TEST sequence requires about 1 minute, including the BURST playback time.

Telemetry Processing

The MAIN computer processes telemetry for both the Langmuir Probe/Electric Field part of the instrument and the Fluxgate Magnetometer part. Hence, there are two separate and independent telemetry formats, one for each function, which share the total telemetry allotment from the spacecraft. These formats are detailed in Figures 2 through 6 in the software chapter and are further described below.

L-Probe/E-Field (LPEF) Formatting. The LPEF part of the telemetry is a table driven (and therefor programmable) format. Two tables, called HX and LX, define the High Rate and Low Rate sampling profile, respectively. The HX table defines 16 channels through which any quantity available to the MAIN processor can be sampled at 4 Hz. Similarly, the LX table defines 32 channels

through which any quantity can be sampled at 1 Hz.

The MAIN computer manages 16 16-byte formatting tables in memory, 10 in ROM and 6 in RAM. The ten ROM formats are numbered 0 thru 9, while the six RAM formats start at 10. (Of the 10 ROM formats designed into the software, only 3 were actually filled, namely 0, 1 and 2.) To select which format tables to use, there are three commands:

```
.FORMAT n m
.VFMT n m
.IFMT n m
```

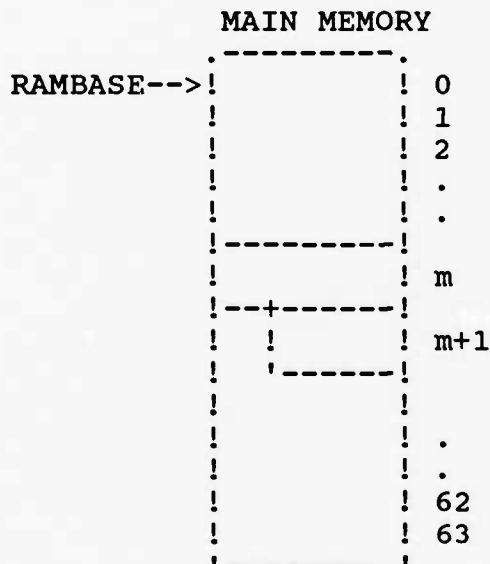
where n is the table to use for the HX list and m is the table for the LX list. FORMAT says that the list is to be used in both Voltage and Current modes. VFMT says the list is to be used ONLY in the Voltage mode, and IFMT says the list is to be used ONLY in the Current mode. These provide the capacity to have different sampling formats for different modes.

The six RAM lists are located in a continuous section of memory which is 64 bytes long. To load a format one uses the "INDEX n" command which selects location n ($0 < n < 64$) and "QTY q" which describes what to sample. For example, to load format 11 with alternating V1 and V2F samples:

```
.INDEX 16
.QTY V1
.QTY V2F
.QTY V1
.QTY V2F
.QTY V1
.QTY V2F
.
.
.
.QTY V1
.QTY V2F
```

Each channel descriptor "q" can be either a MAIN multiplexor quantity or what we call a RAM quantity. To select a multiplexor value, simply name the quantity, such as "V1" or "V2F" above.

RAM quantities are simply values taken from the memory of the MAIN computer system. These are of interest mainly for diagnostic purposes when it is important to have a high bandwidth of information regarding some variables in the computer memory. Since there is a lot of RAM and only 6-bits of possible indexation, the RAM quantities use a programmable 16-bit base address called RAMBASE. The 6-bits are added to RAMBASE to produce an effective address from which 13-bits are retrieved (low byte first then 5-bits of the high byte).



To select a RAM quantity 0 thru 63, use the for ".QTY RAM+n". To set the value of RAMBASE, use the command

```
.RAMBASE n
```

(This allows 11-bit values to be loaded into RAMBASE. For other values, use the .LOAD facility).

Finally, quantity descriptors include one bit which, if enabled, allows playbacks to preempt them. To indicate that playbacks may preempt the channel, add "PE" to the command. For example,

.QTY V1 PE

Sawtooth Generation. In the Current mode, the MAIN computer's SAWTOOTH module is enabled by the executive program to generate linear sweeps on the sphere BIAS voltages.

Playbacks. There are two basic types of playbacks, those from the MAIN and those from the BURST. MAIN transmissions always take priority over BURST playbacks since the latter take much longer. One bit, called the "MAIN/BURST XMIT" bit, is used to distinguish between these two in the telemetry stream.

While decoding playback telemetry, one must watch for transitions between the states of this "MAIN/BURST XMIT" bit. For example, if the BURST is playing back, one must switch to a MAIN playback if the MAIN/BURST XMIT bit goes to a 1 (MAIN).

If the "MAIN/BURST XMIT" bit starts out as 1, the lower priority BURST transmission will not override it and will follow immediately after the MAIN playback finishes.

Command Reception

The chief command capabilities of the instrument are implemented via the serial digital commands. Each command is a 16-bit value which is shifted into the instrument using a standard CLOCK, DATA, and STROBE protocol (see the CRRES-225 document). Each command interrupts the MAIN processor which either executes it or passes it to the BURST processor for interpretation.

Hex Digit Commands. The command capabilities while in orbit are incredibly limited compared to those available through the Ground Support Equipment.

Commands are uplinked to the spacecraft in what is called a

"command pass", which ranges from a few minutes to hours in duration. Prior to a command pass is a command planning meeting in which all command sets are determined. These command sets are relayed to the specific ground station (worldwide) which will be in contact with the spacecraft. Voice communication is used between the Sunnyvale operation and the ground station in order to invoke specific command sets as needed.

The primary limitation with the system is that one cannot send an arbitrary command to an instrument in real-time. (One could conceive a plan in which one sent all permutations of 16-bits to the ground station, but there is a 1000 command limit per instrument.) To get around this problem, seventeen "hex-digit" commands were incorporated into the instrument. These commands are simply the digits plus an "enter" command. To send any 16-bit command, one must convert the command bits into a series of hex digit commands. The on-board microprocessor will act on a string of digit commands just as if it had received the standard serial input command.

For example, to enter the 16-bit command '5678' one would send the following sequence of hex digit commands:

```
DIGIT 5  
DIGIT 6  
DIGIT 7  
DIGIT 8  
ENTER
```

This method is obviously slow (as many as five commands will have to be sent instead of one), and error prone (five times the error rate of one), but it is the only way to do real-time commanding given the design of the ground stations we must use.

Burst Sampling Formats

Just as there are 16 sampling formats available to the MAIN processor, 16 have been implemented in the BURST system. As in the MAIN, 10 formats (0 thru 9) are located in ROM and therefore cannot change. The remaining six sample formats (A thru F) are programmable on the fly. But unlike the MAIN, the BURST system provides sampling formats of varying length. They can have as few as zero and as many as 64 quantities. (Note: The total number of quantities in the 6 RAM lists is limited to 64.)

The procedure for requesting a sampling list involves two separate operations, that of 1) list selection and 2) list definition. To select a format to be sampled, one uses the command "BFMT n", where n is 0 thru 9 for the ROM formats and 10 thru 15 for the RAM formats.

To define a RAM list, first select the list you want to define and then enter the sample series Q1..Qn using the "BQTY q" command as follows:

```
BFMT f
BQTY q1
BQTY q2
BQTY q3
.
.
BQTY qn
```

The format number "f" must be in the range 10 thru 15 decimal in order for anything to change. (You can't change ROM, of course).

Once defined, sample formats may be selected at will by using the BFMT command; i.e. you don't have to re-define the RAM list each time you use it. You can switch around between the different ROM and RAM formats as conditions may warrant.

Upon reset, the BURST cpu defines the RAM formats by copying an area of its ROM over into the RAM lists. This provides for the immediate use of all 16 formats by the user from the start. These formats are given in table 5.

Table 5. Burst Default Sampling Formats

FORMAT	QTY LIST	DESIRED FREQUENCY
0.	V12/RI1	60 KHZ
1.	V12/RI1 V1/SC	30 KHZ
2.	V12/RI1 V34	30 KHZ
3.	V12/RI1 V34 V1/SC	20 KHZ
4.	V12/RI1 V34 BX BY BZ	120 HZ
5.	BZ BX BY V3 V4 V34 V34 AC V1SC V12/RI1 AC V2 V1 V12/RI1 DIRECT AGCU GUARD STUB	

6-F. [EMPTY]

Burst Sample Frequency Control

The BURST computer is capable of sampling a list of quantities at a number of frequencies up to 60 KHz. These frequencies are listed in the Table below. It is important to observe that the frequency is for the whole list, not individual quantities, and thus the size of the sample list defines the maximum frequency at which it can be sampled.

The command which set the BURST sample frequency is

BFREQ f

where f is a frequency code from 0 to 15. (See the table below for the equivalent frequency.) If a frequency is requested which is greater than the maximum for a given list, the BURST processor will sample the list at the highest frequency possible for that list. Thus "BFREQ 15" always guarantees that a list will be sampled at the very highest sample frequency possible.

Burst playbacks always contain the ACTUAL frequency code of the playback data, not what was commanded. Thus, there can be no confusion over the sample frequency of playback data.

The frequency code is independent of the sampling lists so that one does not need to re-command the frequency when one changes lists. Also, the frequency code is not modified by the BURST cpu even when it describes a frequency which is impossible for a given list. Thus, one can set the frequency either before or after one defines or selects the sample list without fear that these other commands may effect the frequency.

Table 6. Burst Frequencies

<u>CODE</u>	<u>FREQUENCY (Hz)</u>	<u>QTY LIMIT</u>
15	62,500	1
14	62,500	1
13	31,250	2
12	20,833	3
11	15,625	4
10	10,417	6
9	6,250	10
8	3,125	20
7	2,000	30
6	1,000	60
5	500	64
4	200	64
3	100	64
2	50	64
1	20	64
0	10	64

2. Software

This chapter describes the software for both the Langmuir Probe Instrument (AFGL-701-14) and the Fluxgate Magnetometer (AFGL-701-13). This is such a large task that it requires its own chapter, one that is separate from the scientific and hardware oriented descriptions of the instrument. At the same time it would be unwise to clutter these other descriptions with the particulars of the instrument's inner mechanisms.

Both the MAIN computer and the BURST computer programs are modularized according to the best description of their function. For example, the electric field module (ELE.A) handles all E-Field/Langmuir telemetry formatting and commands dealing with that part of the instrument. The magnetometer data formatting and commands are handled by the MAG.A module, etc. Table 7 provides a list of the modules in the system. The balance of the chapter describes the detail for each module.

Table 7. MAIN and BURST software modules

MAIN	Major Function
EXEC.A	Executive Program
IO.A	Input/Output Module
BKG.A	Background Processing Manager
ELE.A	Electric Field/Langmuir Probe Manager
MAG.A	Magnetic Field Manager
UTIL.A	Small Utilities
DEP.A	Deployment Manager
PLA.A	Plasma (Low Energy) Instrument Manager
LD.A	Program Load Manager
BUR.A	BURST Triggering Manager
SWP.A	Bias Sweep Manager
FIT.A	Spin Fitting Manager
SPIN.A	Spin Fit Calculator
MATRIX.A	Matrix Solver
TRIG.A	Trigonometric Functions
FFP.A	Fast Floating Point Utility

BURST	Major Function
BEXEC.A	Executive Program
BIO.A	Input/Output Module
BCMP.A	Burst Microcode Compiler
BSMP.A	Sampling Control Module
BFMT.A	Format Manager
BLD.A	Program Loading Manager

2.1 Main Executive Module

The executive module is responsible for coordinating the activities of the Main computer system and its slave, the Burst system. It forms the "foreground" part of the instrument, that part in which non-real-time calculations such as spin-fits can be done. While instrument "autonomic" functions like sampling and telemetry formatting proceed in the "background" under interrupts, the foreground is free for data analysis, mode switching decisions, ground loaded programs, etc.

The EXEC module is responsible for the following functions:

- 1) Defining the instrument initial state;
- 2) Controlling the instrument mode switching (Langmuir Probe versus E-Field versus CALIBRATE);
- 3) Coordinating the mode switching with on-orbit spin fitting and bias sweeps;
- 4) Running ground loaded programs.

The MAIN module has the following entry points :

EXEINIT Jumped to when the processor is reset.

EXEANG Called when the background has changed the sun angle.
On Entry: [A] = new sun angle.

EXEDSC Called when the Digital Subcom wants a status word from the EXEC module. On Entry: [A] is the index into the status word requested. On exit, [A] contains the status byte.

INITIALIZATION / EXECUTIVE LOOP. The module 1) clears the RAM, 2) initializes all of the modules it controls, 3) sends in a

command sequence to define the initial instrument state, 4) resets the "EXEVECT" to null and 5) begins the MAIN executive loop. In this loop, it checks to see if the executive vector has been armed, and executes the vector if so. This allows ground loaded programs to gain control of the foreground (otherwise programs run 1 interrupt deep). The executive loop mainly just calls three routines to share the CPU between the spin fitting (FITEXEC), the bias sweeping (SWPEXEC) and the mode decision (DECMODE).

A technique is used in this executive loop to lower the power of the system by 1) halting the processor when it is not being used and 2) by stopping in RAM not ROM. This latter part works because the ROMS are turned ON only when they are addressed, so to turn them OFF, one simply has to stop the CPU in the RAM.

MODE DETERMINATION. The executive mode switching is controlled by two 16 bit counters called VTIME and ITIME and a 16 bit register called MODTIM. VTIME and ITIME are the number of spins in which the instrument should be in either the Voltage or the Current mode. MODTIM is the count remaining in the present mode.

The EMODE command simply loads VTIME and ITIME with the $2^{**N}-1$ calculations (as described in the LP.DOC). It also sets MODTIM to 1 so that the mode will switch in the next spin period. If one chooses to operate the instrument with other timing, one merely has to load values for VTIME and ITIME using the loader

commands.

Mode switching is performed by a combination of the EXEANG and DECMODE routines. The EXEANG routine compares the current sun angle reported by the background to the angle at which the mode can be switched (CHGANG). This mode-change-angle is initially set to the beginning of V12 spin fits less 11 degrees, or 1/2 the period between fit samples. When the sun angle equals the selected angle, MODTIM is decremented. DECMODE simply checks for MODTIM == 0 to decide to change the mode. (Note: EXEANG runs in the background and DECMODE in the foreground. Since spin fits and bias sweeps are occurring in the foreground, one cannot be sure to "see" every spin angle from the foreground.)

CALIBRATE SEQUENCING. The TEST/CALIBRATE command is implemented by sending in a series of commands just like the initializing sequence. The CALSEQ is the list of commands which implement the test mode. Delay timing is implemented using a modified CALIBRATE command (91xx instead of 90xx). The delay routine "SYNCWT" implements this command which simply waits until the minor frame matches the data portion of the command. By inserting these "SYNC" commands in the CALSEQ, one can set the timing of the events.

2.2 Background Management

The Main Computer system cpu time is subdivided into the standard foreground/background processing profile. Management of the background (interrupts) is accomplished in a single module which organizes and further subdivides cpu time between the several modules which operate in the background.

The major functions of background management are in telemetry formatting, command reception, time and sun angle determination. Other functions include cycling the Kelley AutoGain circuit, and sampling instrument temperatures for the digital subcom. The background manager calls each of its modules when its time for them to sample some data or to telemeter some data.

The timing requirements implemented by the module are in Table 8. The sampling schedule is defined in Figure 1, and the real-time telemetry formats are given in Figures 2 through 4.

Table 8. Module Sampling Frequencies

Frequency	Action
16 Hz	: Sampling of BX, BY, BZ
32 Hz	: Sampling of LEPA
32 Hz	: Sampling for burst Triggering
64 Hz	: High Rate Electric Field (HX)
32 Hz	: Low Rate Electric Field (LX)
64 Hz	: Sawtooth Generator for Langmuir Mode
64 Hz	: Command Execution
64 Hz	: Sun Angle Phase Lock Loop
8 Hz	: Deployment

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
04	SYND	SWP	T/M	SWP	T/M	SWP	T/M	SWP	HY	SWP	BUR	SWP	SUN	SWP	CD/SW	SWP																
30+	PLA	SWP	T/M	SWP	T/M	SWP	T/M	SWP	HY	SWP	LY	SWP	SUN	SWP	CD/SW	SWP																
34+	MG	SWP	TM/MS	SWP	T/M	SWP	T/M	ME	SWP	HY	SWP	BUR	SWP	SUN	SWP	CD/SW	SWP															
38+	PLA	SWP	T/M	SWP	T/M	SWP	T/M	SWP	HY	SWP	LY	SWP	SUN	SWP	CD/SW	SWP																
120	AUTO	SWP	T/M	SWP		SWP		SWP	HY	SWP	BUR	SWP	SUN	SWP	CD/SW	SWP																
16+	PLA	SWP	T/M	SWP		SWP		SWP	HY	SWP	LY	SWP	SUN	SWP	CD/SW	SWP																
170	MG	SWP	TM/MS	SWP		SWP	ME	SWP	HY	SWP	BUR	SWP	SUN	SWP	CD/SW	SWP																
224	PLA	SWP	T/M	SWP		SWP		SWP	HY	SWP	LY	SWP	SUN	SWP	CD/SW	SWP																

MAIN TELEMETRY SAMPLING FORMAT

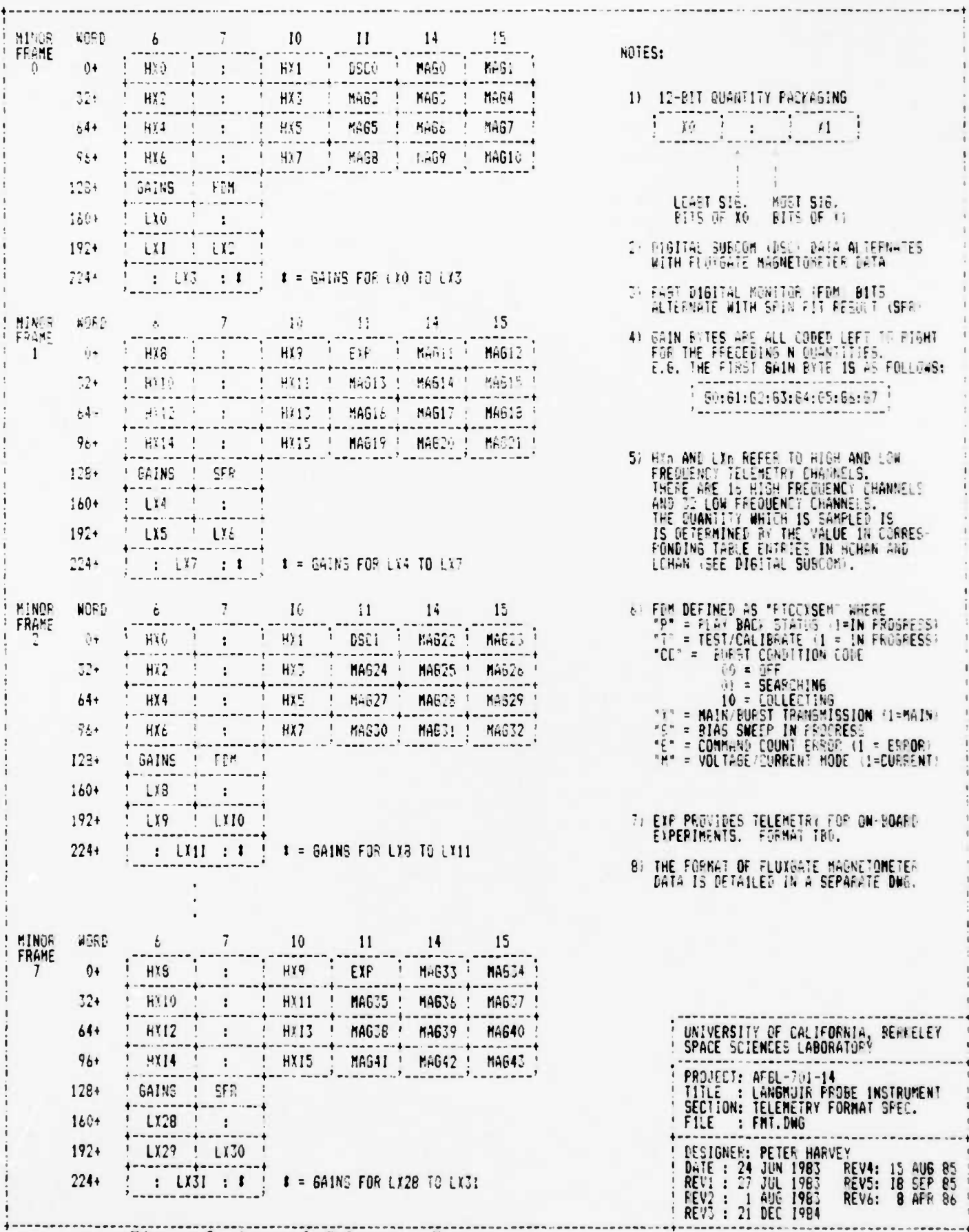
- NOTES: SYND = WINDOW FRAME SYND ORAL MADE TO TELEMETRY MODULES
T/M = TIME TO LOAD THE OUTPUT SHIFT REGISTERS IN PREPARATION FOR TELEMETRY CLOCKS
HY = ELECTRIC FIELD HIGH FREQUENCY LIST SAMPLING
LY = ELECTRIC FIELD LOW FREQUENCY LIST SAMPLING
SUN = SUN ANGLE UPDATES
CD/SW = COMMAND EXECUTION IF COMMAND READY / SAWTOOTH GENERATION
BUR = BURST TRIGGER ALGORITHM
AUTO = KELLEY AUTO GAIN CIRCUIT UPDATE / DEPLOYMENT CHECKING / TEMPERATURE MONITORS
PLA = LOW ENERGY PLASMA SAMPLING/CALCULATION
MG = MAGNETOMETER GAIN SAMPLES TAKEN AND EVALUATED
MS = MAGNETOMETER SAMPLES TAKEN
ME = MAGNETOMETER TELEMETRY ENCODING PERFORMED
SWP = SWEEP SAMPLING PERFORMED (BACKGROUND PROGRAMMABLE VECTOR)

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY

PROJECT: AFGL-101-14
TITLE: LANGMUIR PROBE INSTRUMENT
SECTION: SYNCHRONOUS SAMPLING FORMAT
FILE: SAMP.DWG

DESIGNER: PETER HARVEY
DATE: 27 JUL 1983 REV: 19 SEP 1985
REV1: 21 DEC 1984 REV2: 8 APR 1986
REV3: 23 APR 1985

Figure 1. Synchronous Sampling Format



NOTES:

- 1) 12-BIT QUANTITY PACKAGING

X0	:	X1
----	---	----

LEAST SIG. BITS OF X0 MOST SIG. BITS OF X1
- 2) DIGITAL SUBCOM (DSC) DATA ALTERNATES WITH FLUXGATE MAGNETOMETER DATA
- 3) FAST DIGITAL MONITOR (FDM) BITS ALTERNATE WITH SPIN PIT RESULT (SFR)
- 4) GAIN BYTES ARE ALL CODED LEFT TO RIGHT FOR THE PRECEDING N QUANTITIES. E.G. THE FIRST GAIN BYTE IS AS FOLLOWS:

G0	G1	G2	G3	G4	G5	G6	G7
----	----	----	----	----	----	----	----
- 5) Hx0 AND Lx0 REFER TO HIGH AND LOW FREQUENCY TELEMETRY CHANNELS. THERE ARE 15 HIGH FREQUENCY CHANNELS AND 22 LOW FREQUENCY CHANNELS. THE QUANTITY WHICH IS SAMPLED IS DETERMINED BY THE VALUE IN CORRESPONDING TABLE ENTRIES IN HCHAN AND LCHAN (SEE DIGITAL SUBCOM).
- 6) FDM DEFINED AS "FDCYSEN" WHERE
 "P" = PLAY BACK STATUS (1=IN PROGRESS)
 "T" = TEST/CALIBRATE (1 = IN PROGRESS)
 "CC" = FDMST CONDITION CODE
 00 = OFF
 01 = SEARCHING
 10 = COLLECTING
 "Y" = MAIN/BURST TRANSMISSION (1=MAIN)
 "S" = BIAS SWEEP IN PROGRESS
 "E" = COMMAND COUNT ERROR (1 = ERROR)
 "M" = VOLTAGE/CURRENT MODE (1=CURRENT)
- 7) EXP PROVIDES TELEMETRY FOR ON-BOARD EXPERIMENTS. FORMAT TBD.
- 8) THE FORMAT OF FLUXGATE MAGNETOMETER DATA IS DETAILED IN A SEPARATE DWG.

UNIVERSITY OF CALIFORNIA, BERKELEY
 SPACE SCIENCES LABORATORY

PROJECT: AFGL-701-14
 TITLE : LANGMUIR PROBE INSTRUMENT
 SECTION: TELEMETRY FORMAT SPEC.
 FILE : FMT.DWG

DESIGNER: PETER HARVEY
 DATE : 24 JUN 1983 REV4: 15 AUG 85
 REV1 : 27 JUL 1983 REV5: 18 SEP 85
 REV2 : 1 AUG 1983 REV6: 8 APR 86
 REV3 : 21 DEC 1984

Figure 2. General Telemetry Format 36

FLUXGATE MAGNETOMETER FORMAT
LOGICAL ARRANGEMENT (1/2 SECOND)

B10L	B10H	B20L	MO
B11	B11	B21	
B12	B12	B22	
B13	B13	B23	
B14	B14	B24	
B15	B15	B25	
B16	B16	B26	
B17	B17	B27	
B1			
B2			
B3			

FLUXGATE MAGNETOMETER FORMAT
PHYSICAL ARRANGEMENT (1/2 SECOND)

1	B10L	5	B10	23	B14	41	B1
2	B11	6	B11	24	B15	42	B2
3	B12	7	B12	25	B16	43	B3
4	B13	8	B13	26	B17		
	MO	9		27			
		10	B14	28	B18		
		11	B15	29	B19		
		12	B16	30	B20		
		13	B17	31	B21		
		14	B18	32	B22		
		15	B19	33	B23		
		16	B20	34	B24		
		17	B21	35	B25		
		18	B22	36	B26		
		19	B23	37	B27		
		20	B24	38	B28		
		21	B25	39	B29		
		22	B26	40	B30		

NOTES:

- 1) B10L, B10H, B20L : LOW GAIN VALUES (12-BIT) WHICH ARE TAKEN AT THE SAME TIME AS THE B10-B20 SET.
- 2) B11-B17 : AUTO-GAIN VALUES (12-BIT) (APPROPRIATE GAIN IS DECIDED WHEN EACH SAMPLE IS TAKEN)
- 3) B1, B2, B3 : GAIN BITS FOR B1, B2 (1-HIGH GAIN) CODED LEFT TO RIGHT AS:

B1	B2	B3	B4	B5	B6	B7
----	----	----	----	----	----	----
- 4) MO : 4-BIT MODE INFORMATION (...Y) WHERE Y=1 MEANS B1 IS AMPLIFIED $\times 2$ TIMES
- 5) THE FORMAT TOTALS 903 12-BIT VALUES, 903 1-BIT VALUES, PLUS 4 MODE BITS. THIS REQUIRES 552 BITS OR 69 BYTES.
- 6) EACH 12-BIT VALUE IS IN 2'S COMPLEMENT FORM HAVING A RANGE THEREFORE OF -2048 TO +2047. THE LOW GAIN 12-BIT VALUES CORRESPOND TO A ± 45000 nT RANGE AND THE HIGH GAIN VALUES CORRESPOND TO A ± 900 nT RANGE.

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY

PROJECT: AFGL-791-14
TITLE : LANGMUIR PROBE INSTRUMENT
SECTION: FLUXGATE MAGNETOMETER FORMAT
VERSION 2

NAME : PETER R HARVEY
DATE : 21 DEC 1984
REV1 : 6 FEB 1985

Figure 3. Fluxgate Magnetometer Telemetry Format

MINOR FRAME	DSC	HEX	QUANTITY
0	0	0	MAJOR FRAME COUNT (MOD 8=0)
1	1	1	CHAN(0)
2	2	2	CHAN(1)
3	3	3	CHAN(2)
4	4	4	CHAN(3)
5	5	5	CHAN(4)
6	6	6	CHAN(5)
7	7	7	CHAN(6)
8	8	8	CHAN(7)
9	9	9	CHAN(8)
10	A	A	CHAN(9)
11	B	B	CHAN(10)
12	C	C	CHAN(11)
13	D	D	CHAN(12)
14	E	E	CHAN(13)
15	F	F	CHAN(14)
16	0	10	MAJOR FRAME COUNT (MOD 8=1)
17	1	11	CHAN(15)
18	2	12	CHAN(16)
19	3	13	CHAN(17)
20	4	14	CHAN(18)
21	5	15	CHAN(19)
22	6	16	CHAN(20)
23	7	17	CHAN(21)
24	8	18	CHAN(22)
25	9	19	CHAN(23)
26	A	1A	CHAN(24)
27	B	1B	CHAN(25)
28	C	1C	CHAN(26)
29	D	1D	CHAN(27)
30	E	1E	CHAN(28)
31	F	1F	CHAN(29)
32	0	20	MAJOR FRAME COUNT (MOD 8=2)
33	1	21	CHAN(30)
34	2	22	CHAN(31)
35	3	23	CHAN(32)
36	4	24	CHAN(33)
37	5	25	CHAN(34)
38	6	26	CHAN(35)
39	7	27	CHAN(36)
40	8	28	CHAN(37)
41	9	29	CHAN(38)
42	A	2A	CHAN(39)
43	B	2B	CHAN(40)
44	C	2C	CHAN(41)
45	D	2D	CHAN(42)
46	E	2E	CHAN(43)
47	F	2F	CHAN(44)
48	0	30	MAJOR FRAME COUNT (MOD 8=3)
49	1	31	CHAN(45)
50	2	32	CHAN(46)
51	3	33	CHAN(47)
52	4	34	CHAN(48)
53	5	35	CHAN(49)
54	6	36	CHAN(50)
55	7	37	CHAN(51)
56	8	38	CHAN(52)
57	9	39	CHAN(53)
58	A	3A	CHAN(54)
59	B	3B	CHAN(55)
60	C	3C	CHAN(56)
61	D	3D	CHAN(57)
62	E	3E	CHAN(58)
63	F	3F	CHAN(59)

MINOR FRAME	DSC	HEX	QUANTITY
0	64	40	MAJOR FRAME COUNT (MOD 8=4)
2	65	41	BIAS 1
4	66	42	BIAS 2
6	67	43	BIAS 3
8	68	44	BIAS 4
10	69	45	STUB 1
12	70	46	STUB 2
14	71	47	GUARD 1
16	72	48	GUARD 2
18	73	49	FILTER 1
20	74	4A	FILTER 2
22	75	4B	FILTER 3
24	76	4C	FILTER 4
26	77	4D	FILTER 5
28	78	4E	FILTER 6
30	79	4F	FILTER 7
32	80	50	MAJOR FRAME COUNT (MOD 8=5)
34	81	51	FILTER MUX
36	82	52	RELAY STATUS 0
38	83	53	RELAY STATUS 1
40	84	54	RELAY STATUS 2
42	85	55	SPHERE SWEEP OPTIONS
44	86	56	SPHERE SWEEP ANGLE
46	87	57	SPHERE SWEEP ALGORITHM
48	88	58	SPHERE SWEEP ALTERNATE BIAS
50	89	59	SPHERE SWEEP RESULT
52	90	5A	SPHERE SWEEP M AVERAGE
54	91	5B	CYLINDER SWEEP OPTIONS
56	92	5C	CYLINDER SWEEP ANGLE
58	93	5D	CYLINDER SWEEP ALGORITHM
60	94	5E	CYLINDER SWEEP ALTERNATE BIAS
62	95	5F	CYLINDER SWEEP RESULT
64	96	60	MAJOR FRAME COUNT (MOD 8=6)
66	97	61	CYLINDER SWEEP M AVERAGE
68	98	62	SAWTOOTH OFFSET
70	99	63	SAWTOOTH DELTA
72	100	64	SAWTOOTH PERIOD
74	101	65	SAWTOOTH DIVIDER
76	102	66	BURST MODE/REFRESH STATUS
78	103	67	BURST TRIGGER MODE
80	104	68	BURST DELAY TIME 0
82	105	69	BURST DELAY TIME 1
84	106	6A	DEPLOYMENT STATUS
86	107	6B	BOOM MICROSWITCHES
88	108	6C	TURNS COUNT BOOM 1
90	109	6D	TURNS COUNT BOOM 2
92	110	6E	TURNS COUNT LIMIT
94	111	6F	PLASMA MODE REGISTER
96	112	70	MAJOR FRAME COUNT (MOD 8=7)
98	113	71	32.7 SEC CLOCK
100	114	72	2.25 HOUR CLOCK
102	115	73	24 DAY CLOCK
104	116	74	GOOD COMMAND COUNT
106	117	75	BAD COMMAND COUNT
108	118	76	SUN ANGLE 1
110	119	77	SUN ANGLE 2
112	120	78	SUN PERIOD 1
114	121	79	SUN PERIOD 2
116	122	7A	BOOM LENGTH 1
118	123	7B	BOOM LENGTH 2
120	124	7C	TEMPERATURE 1
122	125	7D	TEMPERATURE 2
124	126	7E	TEMPERATURE 3
126	127	7F	TEMPERATURE 4

NOTE: (0) INDICATES A 2-BYTE VALUE WHICH IS GIVEN LOW BYTE FIRST.

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY

PROJECT: AFGL-701-14
TITLE: LANGMUIR PROBE INSTRUMENT
SECTION: TELEMETRY FORMAT SPEC.
DIGITAL SUBCOMMUTATOR FORMAT

DESIGNER: PETER HARVEY
DATE: 24 JUN 1983 REV3: 8 APR 86
REV1: 31 JAN 1985
REV2: 24 APR 1985

Figure 4. Digital SubCommutator Telemetry Format

Theory of Operation

The background manager handles hardware interrupts from instrument commands, major frames, word timing and the watchdog circuit as described below:

WORD RATE INTERRUPTS. Every other word clock from the spacecraft provides a Word Rate interrupt to the background manager. This is used to update the instrument word count within the minor frame. If it is an even interrupt, one of 64 routines will be chosen depending upon the value of the count. If armed, the background manager vectors to the address contained in BKGVECT on all odd interrupts.

The BKG module has been implemented in such a way as to keep the telemetry formatting modules ignorant of the specific telemetry format. Sampling calls and calls for data bytes are done separately so that each sub-module, like ELE.A or MAG.A, doesn't know where in the minor frame the data is going. ELE.A and MAG.A only know how much data to produce per sample call. In this way, changes to the telemetry format (for the next instrument) should be simple.

The Digital SubCommutator (DSC) output is formatted in the BKG module by the "DSC" routine. Using the list DSCTAB, this routine calls in turn each of the many modules which report status in the DSC data. Each module is handed an index in [A] and returns a byte in [A] which goes into the telemetry. Each module is therefore ignorant of the format of the DSC other than that of its own data. No module knows where in the DSC its data

appears.

Along with telemetry formatting and data sampling calls, the word interrupts provide the basic timing for the sun angle phase locked loop, command execution, sawtooth generation, calls to the low energy plasma calculator, sampling of boom lengths and temperature monitors, and Kelley gain circuit pulsing.

MAJOR FRAME INTERRUPTS. Every four seconds, the major frame interrupt occurs. These interrupts used to synchronize the telemetry word count with the spacecraft as well as to internally synchronize the data sampling packages such as ELE and MAG. The 40-bit instrument clock is updated during the major frame interrupts.

COMMAND INTERRUPTS. Command interrupts occur as soon as the spacecraft command begins to shift into the instrument. The software merely notes that a new command will be arriving soon. The actual command processor is a routine which runs under the word rate clock interrupt.

WATCH DOG INTERRUPT. The "watchdog" is a simple circuit which counts major frames and is reset by a bit on one of the output ports. If two major frames occur while the watchdog has not been reset, the watchdog fires a TRAP type interrupt. Normal processing of the telemetry includes periodic pulsing to the reset line on the watchdog. A loss of this reset pulse means that the software has crashed for some reason, so the background manager simply resets the processor when this occurs.

There are several entry points for the background manager,

most of which perform some utility function of the background. These as described below:

CMDGO. This entry point, called by software restart 6, executes a command in the [HL] registers as if the command just came from the ground. This is used for internal module to module controls.

BKGFN(1). This call (software restart 4 with [A] = 1) is a batch command processor which executes a list of commands terminated with a 0FFFFH command. On entry, [DE] address a list of commands.

BKGFN(2). Function 2 is a request of the command count status. If the command count matches what it is supposed to match then a zero is returned in the [A] register. If the command count is incorrect, [A] is returned with 1.

BKGFN(3). Restart 4 with [A]=3 is a call to the STVECT routine. If [HL] contain a non-zero address, the background vector BKGVECT is set to [HL]. If [HL] is zero, BKGVECT is reset and disarmed. As described above, the BKGVECT routine is called every odd word interrupt (which is every 2 milliseconds).

BKGFN(4). The fourth function on restart 4 is the STEXP function. On entry, [HL] points to a data block of [DE] bytes. The STEXP routine starts playing bytes from this block into the EXP telemetry slot. (EXP is the undefined, EXPerimental output which resulted from a decrease in the magnetometer telemetry allocation.)

INITIALIZATION. The BKGINIT entry enables the interrupts and initializes variables internal to the module so that it works correctly. Modules which are subordinate to the background module are initialized so they too are guaranteed to work when called upon.

The BKG module waits for the first major frame spike to enable word interrupts so that the telemetry comes "up" in synchronization. Depending upon when the instrument is turned ON, this may take up to one major frame time to start processing (4.096 seconds). During this time, commands will not be recognized.

COMMAND EXECUTION. Commands are executed by vectoring through a 32 element table called CMDTAB using the upper 5 bits of the command. Each command routine is executed with [HL] holding the command bits as well as [A] redundantly holding the low 8 bits. Command routines return carry if the command was BAD and no-carry if the command was GOOD. The BKG module keeps 8-bit counters GOODCNT and BADCNT of the commands which were so noted. If the command table has no vector there (the address is 0), the BADCNT is incremented.

BACKGROUND COMMANDS. The module implements two types of commands, namely, the "digit" and the "command count" commands. Digit commands emulate a hexadecimal shift register having the digits 0 thru F and an ENTER key. Digit commands shift a hex register left one digit and add the new digit. The ENTER command sends the contents of the register into the instrument as a new

command. This is designed to be used with an old command capability of the Air Force facilities which could not readily send arbitrary bit patterns to our instrument. Digit commands would be used to create arbitrary bit patterns inside the instrument.

Command-count commands are used to check for lost commands within a long command series from the ground. The satellite control facilities cannot check each command if they are grouped in series. The "CMDS" command enters a count of the commands which will be arriving in a block. This data is put into a register called CMDCNT while GOODCNT and BADCNT are zeroed. When the block of the expected length is finally entered, the Fast Digital Monitor will report the result of BKGFN(2), which compares the expected count to the good count. If a command is missed (satellite uplink problems), the good count will be too small and the Fast Digital status will report the error.

Some of the BKG variables are described below:

ON-BOARD CLOCK. The BKG module maintains a 40 bit clock which counts 0.5 milliseconds (2 KHz). This clock is defined as follows:

```
"DAY24" "HR225" "CYCLE" "FRAME" "WORD"  
-----  
!ddddddd!hhhhhhh!ccccccc!jjjmmmm!wwwwwww!  
-----
```

where w is the telemetry word within the minor frame,

m is the minor frame within the major,

j is the major frame count (internal),

c is the count of digital subcom cycles since reset,

h is the count of 2.25 hours since reset,

d is the count of 24 days since reset.

The clock is zeroed when the instrument is turned on and overflows after approximately 17 years.

SPIN PERIOD. The module keeps a 16-bit value of the spin period by counting the number of 16 millisecond pulses between sun pulses. This yields a value which can describe spin periods of up to 1024 seconds.

SUN-ANGLE. The BKG module keeps a 16-bit value of the angle between the sun sensor and the sun. The low eight bits subdivide a spin in 256 equal parts of 1.41 degrees each. The upper byte can be used as an overflow byte which shows that the sun sensor is not working (this occurs in shadow). The algorithm uses the spin period value calculated above divided by 256 (with remaindering) as a counter to decide when to step the sun angle. Below 4.096 second spin periods, the division underflows and

causes the sun angle to stay at zero.

For a 30 second spin, the sun angle is accurate to 1 part in 2000 or .05% . During shadow periods, however, this error will accumulate since the sun pulse will not reset the angle. One hour of shadow (120 spins) times an accumulated .05% error is about $.06 \times 360$ or 21.6 degrees (worst case). A typical error value would be half of that, or 10.8 degrees.

2.3 Electric Field Management

The ELE package manages the Electric Field/Langmuir Probe part of instrument operations. The functions include sampling, telemetry formatting and commanding the related sections of the instrument. For this module there are the following entry points:

ELEINIT : This entry requires no parameters and initializes the module for subsequent calls. The only thing this entails is zeroing the electric field RAM area.

ELEFRAME: This entry tells the module the minor frame time. On entry, [A] contains the 8-bit frame count. The module uses this information to decide whether to output SFR or FDM data, to synchronize the HX and LX list pointers, etc.

ELESAMP : The sample entry tells the module when to take one of its analog samples. The ELE package requires 16 sample calls per frame call. In addition, both the even calls and the odd calls must be periodic; i.e. even-to-even and odd-to-odd is a constant length of time.

ELETELEM: The telemetry call requests the ELE package to give some of its formatted data. On entry, the [A] register contains 0 if 1 byte is requested, 1 if 2 bytes are requested. On exit, the [L] register contains the first byte and [H] contains the optional second byte of data. The ELE package produces 20 bytes of data per 16 samples.

ELEDSC : Slow status of the package can be obtained by calling the DSC entry point. On entry, [A] is an index into the lists and variables which the ELE package maintains. ELEDSC returns its data in the [A] register.

When [A] is 0 to 15, the HX list is returned.

When [A] is 16 to 47, the LX list is returned.

When [A] is 48 or more, the ELE variables are returned.

Note that the HX and LX lists for E-field and Langmuir modes are output in alternating fashion. The LP lists are output on odd major cycles.

ELEXMIT : Accesses the MAIN playback feature of the ELE package. On entry : [HL]-> data block of 13-bit data, [DE] is the number of samples to play back. On the first even minor frame, the playback will begin, overriding Burst playbacks if need be.

ELESTAT : Returns the Fast Digital Monitor in [A].

Theory of Operation

TELEMETRY SAMPLING/FORMATting. The telemetry formatting uses a double-buffered scheme with each buffer representing one minor frame worth of data. Each minor frame toggles which buffer is to be used as the input buffer and which to be used as the output buffer.

The two types of samples, namely High Rate (HX) and Low Rate (LX), are dumped into the input buffer using two pointers called

HBPTR and LBPTR. Data is read out of the buffer using a single pointer called TMPTR. This results in a one minor frame delay in the data stream as samples which are taken in even frames are played in odd frames and vice versa.

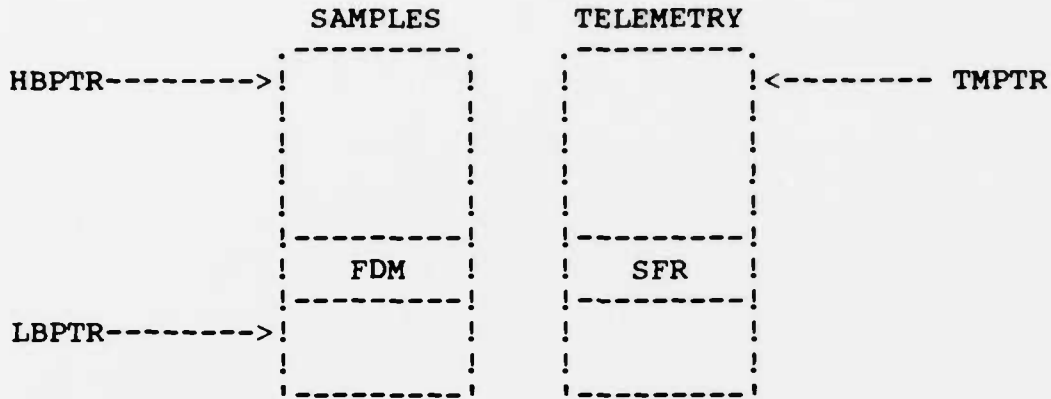


Figure 5. Electric Field Telemetry Buffering

Corresponding with each high or low rate sample is a quantity descriptor which details what value to sample at that time. The quantity descriptors are in lists which are read using HQPTR and LQPTR as the samples are taken. These pointers are reset to the top of their lists as required and incremented along with each use.

RAM quantities are simply values taken from the memory of the MAIN computer system. These are of interest mainly for diagnostic purposes when it is important to have a high bandwidth of information regarding some variables in the computer memory. Since there is a lot of RAM and only 6-bits of possible indexation, the RAM quantities use a programmable 16-bit base address called RAMBASE. The 6-bits are added to RAMBASE to produce an effective address from which 13-bits are retrieved (low byte first then 5-bits of the high byte).

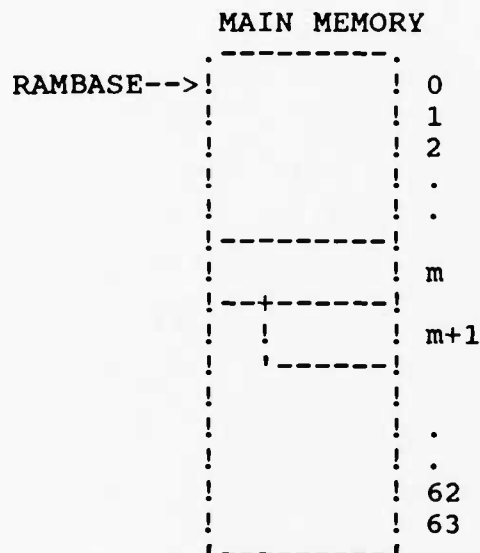


 Figure 7. Ram Quantities

When the Burst playback is indicated by the Fast Digital Monitor, sampling of channels with their playback bit "b" enabled is replaced with a BURPLAY call to the BUR package. Depending upon the state of the playback, the BUR module may offer some of its Burst Header or may request data from the Burst computer. In either case, thirteen bits are stored in the sample buffer as described above.

When a Main playback is indicated, the ELEXMIT facility is invoked. Samples are played from the buffer pointed at by the XMPTR variable until the count held in XMTCNT goes to zero. When XMTCNT is zero, zero fill data is returned as a sample and the next FDM calculation clears the MAIN transmit state.

HQPTR/LQPTR. The quantity list pointers are reset when necessary to the value contained in VHXPTR and VLXPTR when in the E-field mode or to IHXPTR and ILXPTR when in the Langmuir Probe mode. Thus, if the instrument switches modes, the sampling list is changed along with it.

As all of these pointers are 16-bit values, any buffer in memory can be used for a telemetry format. Six buffers of 16 bytes each have been allocated for this purpose in the ELE module RAM area. This is enough for complete redefinitions of both the V and the I mode telemetry formats. Originally, the package was defined with the idea of having multiple ROM loaded formats from which to choose. This would facilitate various tests, sampling modes, etc. Ten format buffers of 16 bytes each were planned for the ROM, but only 3 (0,1,and 2) could be accomodated. The "format" command was devised as an address independent method to access these buffers. See the "format" command in the general instrument description.

FAST DIGITAL MONITOR CALCULATION. Once every other minor frame the Fast Digital Monitor is calculated and stored into the samples buffer. The FDM is a conglomeration of a number of

status bits which indicate the instrument mode (E-field or Langmuir Probe), playback mode (Main, Burst or Real-time), etc. The format of the FDM is given in Figure ELE-1. These status bits are collected by calling the various modules responsible for the bits; i.e. the BUR, SWP, and BKG modules.

COMMANDS. Commands are vectored through the command vector table to routines contained in the ELE package. Most of these commands are simple calls to the IO package to perform some function such as setting the bias voltage. The software in the ELE package simply makes up the differences between the external command specifications and the internal IO call specifications. Where no differences occur, the ELE package directly references the IO call.

The VMODE and IMODE commands are interesting since they are actually command macros. When a VMODE command is executed, a list of commands are sent into the command-list processing facility of the BKG module. Since command lists are executed only when real commands are not ready, the execution of a VMODE command may occur after the series of real commands which follow the VMODE. One should probably not use relay commands in series right after either a VMODE or IMODE command (i.e. within 25 milliseconds). This could cause contention for the relay hardware.

2.4 Magnetic Field Management

The MAG package manages the Fluxgate Magnetometer part of instrument operations. The functions include sampling, telemetry formatting and commanding the related sections of the instrument. For this module there are the following entry points:

MAGINIT : This entry requires no parameters and initializes the module for subsequent calls. It also sets the mode of the BY amplifier to 0 (BY*1 operation).

MAGFRAME: This entry tells the module the minor frame time. On entry, [A] contains the frame count. On minor frames 2, 6, 10,... the sampling pointers are resynchronized while on frames 0, 4, 8,... the output pointers are synchronized.

MAGGAIN : This entry tells the package to determine the gains for the three axes. Each axis is sampled in low gain and compared to 1/51th of full scale. If less than that, high gain will be selected for the SAMP call. If more, the low gain sample will be selected.

MAGSAMP : The sample entry tells the module when to take the three axis measurement using the gains determined in the GAIN call.

MAGENCD : This entry tells the module to buffer the data sampled by the last SAMP call.

MAGTELEM: The telemetry call requests the MAG package to give

some of its formatted data. On entry, the [A] register contains 0 if 1 byte is requested, 1 if 2 bytes are requested. On exit, the [L] register contains the first byte and [H] contains the optional second byte of data.

The GAIN/SAMP/ENCD calls are actually three parts of 1 process. They are split in order to meet the system requirement to stay under 1 millisecond for background processes. The package requires two SAMP calls per FRAME call and produces 11 bytes of data per FRAME. (See Figures 1 through 3).

Theory of Operation

The magnetometer format involves buffering 8 triplets of autogain (BX, BY, BZ), eight triplets of gain bits, as well as 1 triplet of (BX, BY, BZ) taken in low gain. Four bits of mode information fill out the format to 44 bytes.

The package uses a single buffer scheme in formatting its data. Three pointers are required as well as temporary storage for gain registers. One pointer, called OTPTR, is used to read out of the buffer. It is reset to the beginning of the buffer at every frame divisible by 4 (1/2 second) and is stepped every time a byte of magnetometer data is required.

Two pointers, LGPTR and AGPTR, are used to store data into the buffer. Both pointers are reset at every frame mod 4 equal to 2. This causes the input buffering to be filling a different half of the buffer that is being read from by the OTPTR. The LGPTR is used to index low gain storage while the AGPTR indexes

the auto gain storage. Both pointers actually count the number of nibbles from the beginning of the buffer since this facilitates storing 12 bit quantities.

The number of samples in the autogain buffer is counted by SMPCNT, which is reset when the storage pointers are. On the first sample the low gain values (which are taken for the auto-gain decision) are stored using LGPTR. Otherwise, all sample triplets are stored using AGPTR.

Since the sampling of autogain quantities occurs simultaneous to the output of previous gain decisions, the gain bits for a given buffer are stored in temporary locations TMPX thru TMPZ. On the last sample of the buffer (all quantities finished) these values are copied into the output buffer.

Mode information (4-bits) is buffered at the end of the first sample, following on the heels of the low gain storage. Mode bits are set by the BMODE command to the package.

MAG COMMANDS. The single command which the MAG module knows about is the BMODE command. The least significant bit of this command determines whether or not the BY amplifier is engaged on the Filter board. The software in the MAG module simply calls the IO module SETMUX facility in response to the BMODE command.

2.5 Plasma Instrument Data Management

The PLA package manages the Low Energy Plasma part of the instrument operations. The science requirements of this package are as follows:

- 1) Whenever the ABS(SCBY) is less than 1/8th ABS(SCBX), send the LEPA instrument a compressed calculation of SCBZ/SCBX plus current BURST mode information.
- 2) The software should produce results at least every 64 milli-seconds.
- 3) Use appropriate gain (high or low) in producing the calculations.

Note that SCBX, SCBY, and SCBZ listed above refer to the mag field in the spacecraft coordinate system, not the fluxgate outputs. The fluxgate sensor is turned such that its axis outputs correspond to spacecraft coordinates as follows:

$$\begin{aligned} \text{SCBX} &= - \text{FLUXGATE BX} \\ \text{SCBY} &= - \text{FLUXGATE BZ} - .044 \text{ FLUXGATE BY} \\ \text{SCBZ} &= - \text{FLUXGATE BY} \end{aligned}$$

The LEPA aperture lies in the spacecraft XZ plane with Z along the spin axis.

The system requirements for the package are as follows:

- 1) The sample/calculation time cannot exceed 1 millisecond.
- 2) Since the spin-axis measurement has a mode of amplification, adjust it back to unity gain whenever the

amplifier is used.

Specifically, the format for the information transferred to the LEPA instrument is as follows:

16-bit LEPA shift register: :-----:-----:
 !10qqffff!vsnnnnnn!
 '-----'-----'

where qq is a status code which indicates the condition of the BURST sampling.

qq= 00 = BURST OFF

01 = BURST ON, SEARCHING AT FREQUENCY F

10 = BURST ON, COLLECTING AT FREQUENCY F

f is a frequency code at which the burst is operating.

(See General Description, "Burst Sample Frequency Control")

v is a range error bit if $ABS(BZ/BX) \geq 2$.

s is a sign bit of BZ/BX (1=negative)

n is $32*ABS(BZ/BX)$

For example, if $BZ=BX$, $n=100000$; if $BZ=1/2 BX$, $n=010000$.

For this module there are the following entry points:

PLAINIT : This entry requires no parameters and initializes the module for subsequent calls.

PLASAMP : The sample entry tells the module when to take one of its analog samples.

PLADSC : Status of the package can be obtained by calling the DSC entry point. The PLA package has 8 bits of status returned in the [A] register as "esqqffff" where e=0 indicates the package is enabled

s=1 indicates the package is currently sending.

q and f are as described above.

PLA PACKAGE COMMANDS. The PLA package has a single command, called LMODE, whose routine "PLACMD" is referenced in the command vector table CMDTAB. The format of that command is as follows:

.LMODE d where d= "eeqqffff"

will set the q and f fields of the mode register. Field "ee" enables/disables the entire package in case a ram-loaded algorithm is being used.

If ee = 10 the package is enabled.

= 11 the package is disabled.

= 0x (no change).

The operation of the "ee" field seems a little strange. Why go to the trouble of an arming bit for another bit? Well, the problem is simply that the BUR module communicates its status via the LMODE command. It of course uses the "no change" form of the command in case the PLA module has been disarmed earlier.

Theory of Operation

With the exception of PLASAMP, the PLA entry points are already described. PLASAMP is a routine which performs the following (note that BX, BY and BZ are fluxgate measurements and SCBX, SCBY and SCBZ are spacecraft coordinates):

1) For each axis, the PLA package uses the MAG package samples to determine the field values to 16-bit accuracy (2nT per bit).

a) If low gain sample, it adds a low gain offset. If a high gain sample it adds a high gain offset.

b) If a low gain sample, it then multiplies by 51 (18-bit result).

c) For either gain, it divides by 4 (16-bit result).

2) For the BY measurement, if the BY amplifier is ON, the BY field value is scaled by 39/256 (1/6.2) and inverted.

3) Rotation to spacecraft coordinates is approximated by

a) $ABS(SCBX) = BX$

b) $ABS(SCBY) = BZ + (11/256)*BY$

c) $ABS(SCBZ) = BY$

4) Compares $ABS(SCBX)$ with $8*ABS(SCBY)$ and performs the $SCBZ/SCBX$ calculation if the former is greater.

4a) Calculates $32*ABS(SCBZ/SCBX)$ by performing an 8-bit divide of $ABS(SCBZ) / 2*ABS(SCBX)$. The 8-bit result is then rounded and shifted to produce a 7-bit result which exceeds 64 if $ABS(BZ) > 2*ABS(BX)$.

4b) Valid 6-bit results and masked 7-bit results are put into the 6-bit mantissa field along with an appropriate overflow bit (0 or 1).

4c) The sign of SCBZ/SCBX is calculated using an exclusive-or of the signs of their field values.

2.6 Burst Processing Management

The BUR package coordinates the Burst Sampling and Playback part of the instrument. The functions include sampling of conditions to determine when to take a burst of data, communicating with the burst computer system to receive stored data, and accepting commands for both the module and the BURST computer.

The science requirements for this function are as follows:

- 1) Sampling of conditions must operate at a fixed rate with a period no greater than 32 milliseconds, and must continue even while playing back data to the telemetry system.
- 2) Ram algorithms must be loadable from the ground and there should be a selection of ROM algorithms to choose from. One algorithm must be a simple clock which will be used for the timely Bursting during the chemical releases.

The system requirements for the module are:

- 1) The module must be provided some capability for starting and stopping burst playback transmissions.

The module has the following entry points:

BURINIT : This entry requires no parameters and initializes the module for subsequent calls. Playback requests are cleared and the default burst duration set to 4 seconds.

BURSAMP : The sample entry tells the module when to take one of its analog samples with which it will determine whether or not to take a burst collection, start the playback, etc.

BURPLAY : The telemetry call requests the BUR package to give some of its stored data. On exit, [HL] contain a 13-bit value for the telemetry stream if playback has been requested. Otherwise, [HL] will contain zero.

BURDSC : Status of the package can be obtained by calling the DSC entry point. On entry, [A] is an index into the variables which the package maintains. Data is returned in the [A] register.

Of special interest is the most significant nibble of DSC(0) which contains the internal mode of the package. These 4 bits are intended to be used in the fast digital monitor as an indicator of burst conditions.

Theory of Operation

The burst module operates completely in the background (under interrupts) trying to decide when to take bursts of data. As the controller of the Burst computer, all commands to the Burst computer go through the BUR module in order to keep the module aware of what is going on with the Burst.

The module has a state variable called MODFREQ (mode and frequency) which is used to remember from one SAMP call to another what the state of the Burst is. Three of the bits of

MODFREQ end up in the Fast Digital Monitor since they describe the internal mode of the BUR module. Eight states are defined for the module as follows:

- 0 OFF
- 1 SEARCH
- 2 COLLECT
- 3 WAIT
- 4 R1--SENDING "BGO" COMMAND TO BURST
- 5 R2--RECEIVING "REAL FREQUENCY" DATA
- 6 R3--RECEIVING "DURATION" DATA
- 7 R0--DELAYING BEFORE ENTERING R1 STATE

When any algorithm is selected using the control command (see below), the state of the module is set to R1. The module goes thru states R1, R2, R3 and ends up in SEARCH.

In SEARCH mode, each SAMP call simply vectors to the selected algorithm to determine whether searching is over or not. When the decision is made to end the SEARCH mode, the algorithm calls the "TRIGGER" routine which copies the burst duration parameter into the delay timer DTIME, saves the "EVENT TIME" and sets the state to COLLECT.

In COLLECT mode, each SAMP call simply decrements the delay timer. When zero, the collection phase is over, the "BSTOP" command is issued to the Burst computer and the "END TIME" is recorded.

Finally, the state is set to "WAIT" for the Burst computer

to close its Burst memory file and get ready to play back. When this occurs, the state is set to "OFF", the Burst is commanded to play its data back and the playback bit is turned on in the MODFREQ status byte.

While the playback bit is set, the PLAY entry point will be called each time there is a channel armed for playback. The first thing played back into the telemetry is the header which shows the algorithm information and so forth. After its been played out, the BUR module requests data from the Burst computer system. This is the data portion of the Burst playback. See Figure 8.

When the Burst no longer has any data to play back, the BURPLAY routine decides whether to go into the OFF state or back into the R1 state. This depends upon the AUTOSEARCH bit in the trigger mode word (set by the BTRIG command). If AUTOSEARCH is armed, then the BUR module uses the R0 state to delay 1/2 second before going into R1. This makes sure that the playback bit in the Fast Digital Monitor returns to zero in between burst playbacks---a fact required by the telemetry decoding in order to distinguish between burst playbacks.

COMMANDS. BUR commands fall into three categories: (1) Burst CPU commands, (2) BUR algorithm commands and (3) BUR control commands. The first category of commands is simply those commands which the Burst CPU knows how to perform, such as BFMT and BANKS, etc. To each Burst computer command, the BUR module has a response in terms of its own state. That is to say, if the user commands the Burst computer to play back some data, the BUR

module goes into the playback state as well.

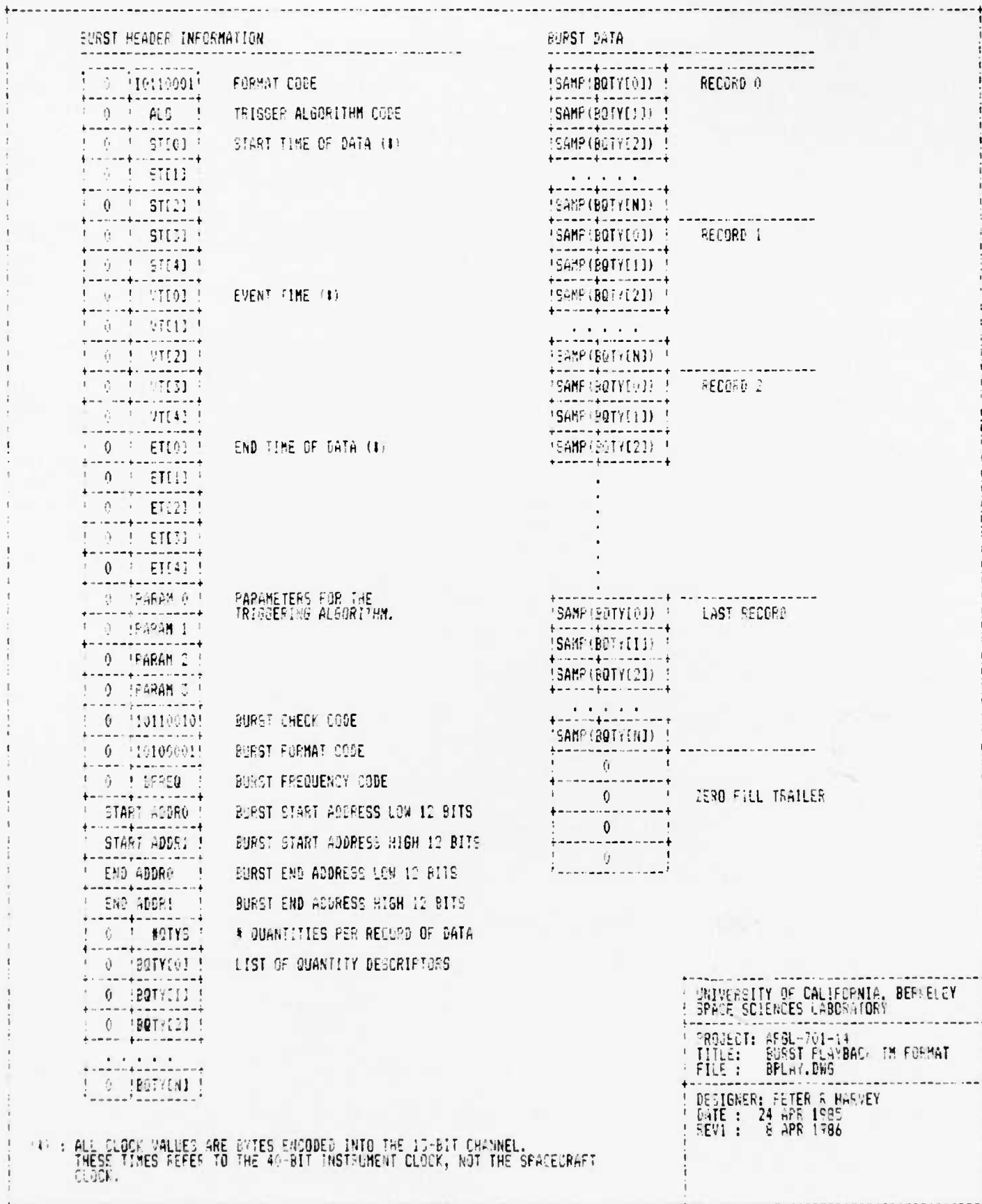
There are four BUR module control parameters: (0) trigger mode, (1) timer delay Low, (2) timer delay High and (3) spare. Setting the trigger mode turns ON and OFF the BUR module, etc. See the general information for more detail on what the commands do.

The algorithm commands simply set the four variables which are available for use by the algorithms. These are mostly undefined and are intended for use by the RAM algorithms.

Whenever any of the control or algorithm parameters are going to be changed, the burst trigger is first set to "OFF". This keeps algorithms from being executed using partial parameters and so forth. The impact to the user is that the last command should be the setting of the burst trigger (BTRIG command).

ROM TRIGGERS. The triggers available in the flight ROM are as follows:

- 0 OFF
- 1 IMMEDIATE Immediately begins collecting.
- 2 VALCHECK Begins collecting when magnitude of SAMPLE(PARAM0) exceeds PARAM1.
- 3 MAGCHECK Begins collecting when the PLA module detects the loss cone (it is sending). The status of the PLA package is requested through its DSC function.
- 4-7 RAM RAM loaded algorithm.



2.7 Spin-Fitting Management

The FIT package controls the spin-fitting of electric field data. The functions include sampling electric field data at particular sun angles, calculating the sine wave least squares fit, and telemetering this information. Unlike other modules, these operations are synchronized with the spin period of the spacecraft rather than with telemetry timing.

Specifications for the module are as follows:

- 1) 32 points taken at even angles with respect to the sun.
- 2) Full period capability only.
- 3) No bias changes should be made during the sampling.
- 4) Perform fits in voltage mode only
- 5) Sine wave fit the boom systems at 180 degrees out of phase with each other. This will give better temporal resolution (1/2 spin) of the electric field if the sphere and cylinder measurements are comparable. If they are not comparable, it doesn't hurt.
- 6) Each fit has the following results:
 - Sine and Cosine components (floating pt)
 - High and Low gain offsets to the fit (floating pt)
 - Standard deviation (floating pt)
 - A code to distinguish sphere and cylinder fits.
 - The number of points used in the fit.

For this module there are the following entry points:

FITINIT : This entry requires no parameters and initializes the module for subsequent calls by copying initial values

from ROM into the parameter blocks for both the V12 and V34 sensor pairs. For both blocks, it NULLs all old AHI and ALO values so that the SPIN package starts off with no history of AHI or ALO to contend with.

FITSMP : The sample tells the package of a particular sun angle at which the module may decide to take a sample. On entry, [A] is the 8-bit sun angle. This function samples and stores the data from both boom systems when the angle is a 360/32 boundary.

FITTELEM: The telemetry call requests the FIT package to give some of its formatted data. On exit, the [A] register contains the byte of data. The FIT package produces an asynchronous format of data (zero filled) with up to 34 bytes of non-fill data every spin. See Figure 9.

FITEXEC : Performs the spin fits if ready to do so. Returns minus if its not ready. This entry point should be called from the foreground since it may take up to 1/2 second to complete the pair of fits.

SFR DATA

```

+-----+
|1111000X| FIT TYPE (X = 0 FOR SPHERES, X = 1 FOR CYLINDERS)
+-----+
| AHI(0) | AHI OFFSET (FLOATING POINT)
+-----+
| AHI(1) |
+-----+
| AHI(2) |
+-----+
| ALO(0) | ALO OFFSET (FLOATING POINT)
+-----+
| ALO(1) |
+-----+
| ALO(2) |
+-----+
| B(0) | B COMPONENT (FLOATING POINT)
+-----+
| B(1) |
+-----+
| B(2) |
+-----+
| C(0) | C COMPONENT (FLOATING POINT)
+-----+
| C(1) |
+-----+
| C(2) |
+-----+
| SIGMA(0) | STANDARD DEVIATION (FLOATING POINT)
+-----+
| SIGMA(1) |
+-----+
| SIGMA(2) |
+-----+
| N | NUMBER OF POINTS REMAINING IN FIT
+-----+

```

NOTE: THE SFR TELEMETRY DATA IS ASYNCHRONOUS WITH THE NORMAL TELEMETRY DATA. BLOCKS OF SPIN FIT DATA ARE SENT SYNCHRONOUS TO THE SPIN PERIOD, TWO BLOCKS SENT EACH ROTATION (ONE FOR SPHERES, ONE FOR CYLINDERS).

THE SFR BLOCKS ARE SEPARATED FROM OTHER SFR BLOCKS BY ZERO FILL.

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY

PROJECT: AFGL-701-14
TITLE: SPIN FIT RESULT TELEMETRY
FILE: FIT.DWG

DESIGNER: PETER R HARVEY
DATE: 8 APR 1986

Figure 9. Spin-Fit Result Telemetry Format

Theory of Operation

BUFFERING. At each selected angle which is a multiple of $360/32$ the FIT module samples the V12 and V34 measurements (unless disabled of course). These data are stored in two double buffers, one pair for each type of data, using pointers V12IN and V34IN, which are incremented with each use.

For each boom system there is an angle at which fit sampling is started and ended. When this angle is met for a given boom pair, the buffer pointer is checked to see if the previous buffer has been completed. This is a reasonable question since changes in the spin period may cause short cycling of the sun angle phase locked loop (i.e. skips from 280 degrees to 360 degrees for example). If the buffer has been filled with the correct number of points, a variable called V12OUT (or V34OUT) is pointed at it.

These pointers are checked by the foreground process FITEXEC. If they point at a buffer, then the SPIN module is called with the appropriate parameter and result pointers. When the spin fit is finished, the V12OUT or V34OUT pointer is set to a "READY" value to indicate that new spin fit results are ready to be transmitted.

TELEMETRY. Sending fit results into the telemetry stream is accomplished by the FITTELEM routine. It is called by the ELE module each time the SFR TM slot comes up (every other minor frame). The FITTELEM routine hands out bytes one by one from memory pointed at by PTR, using a count contained in COUNT. When COUNT is zero, the PTR is reloaded with any results marked

"READY" by their output pointer (V12OUT or V34OUT). Once the pointer PTR and COUNT are loaded, the "OUT" pointer is set to "DONE", so that the results will be transmitted only once.

FIT COMMAND. Options in the package operations are selectable using the single FITMODE command (see the general description). One option completely disables the FIT package so that a RAM loaded procedure can be substituted.

A second control bit determines whether V12 fits should be performed. This is used by the EXEC mode switcher when going into the Langmuir Probe Mode since it is nonsense to spin fit V12 at that time.

A third option is a diagnostic dump of the data points used in the spin fit. This feature uses the ELEXMIT routine to form a MAIN transmission of the 32 samples. Unfortunately, there is no header protocol to distinguish it from bias sweep results except for the lack of the sweep header.

INITIAL PARAMETERS. The parameter block for each boom system is copied from the ROM into the RAM at initialization so that adjustments may be made as desired. The initial values for both boom systems are given below:

GAINFACTOR	1/50.9
ALPHA	1.40
BETA	0.40

2.8 Spin-Fitting Computations

The SPIN.A module is devoted to only one function, namely producing sine wave least squares fits of some measured data. This function is by far the most complicated of those in the instrument (although bias sweep analysis is a close second). It is called solely by the FIT.A module which handles the sampling of the data for the two boom systems. All CRRES dependent stuff (if there is any) is contained in FIT.

The SPIN module takes 32 points of evenly sampled data which represents one spin period. It produces the sine, cosine and offset parameters which approximate the waveform. Also produced is the standard deviation for the points.

Assume $E(t_i)$ is the t_i 'th measured value of the electric field between the two sensors (either V12 or V34). The approximating waveform's expression in general would be

$$E(t_i) = A + B \cdot \cos(wt_i) + C \cdot \sin(wt_i)$$

where w is $2 \cdot \pi \cdot v$ and v is the spacecraft spin frequency. However, since the instrument samples data in two gain states, the true approximating formulas are

$$E(t_i) = A_{hi} + B \cdot \cos(wt_i) + C \cdot \sin(wt_i)$$

for high gain points and

$$E(t_i) = A_{lo} + B \cdot \cos(wt_i) + C \cdot \sin(wt_i)$$

for the low gain points.

What we want to do is form a difference function between the approximate expression and the sampled data. The best fit occurs when we minimize this function. Figure 10a thru 10e show the

difference function and its associated differentials. Equations (b) thru (e) give four equations in four unknowns which can, of course, be solved by standard matrix methods. The matrix elements are shown in Figures 11a and 11b.

The procedure of SPIN is actually more complicated than simply solving the function once. Error points should be removed in the process of determining the best fit of the DC field. These "error" points are real data, of course, but reveal AC activity which we want to remove from a DC measurement. The procedure followed by the SPIN module is as follows:

1. Least squares fit the input data to find A_{hi} , A_{lo} , B , C and the standard deviation Σ .
2. Discard all points more than $\text{Alpha} * \Sigma$ from the least squares curve, where Alpha is an input constant.
3. Repeat the least squares fit using the remaining points. For the j th fit, throw away points more than $(\text{Alpha} + \text{Beta}) * \Sigma$ from the curve. (Beta is also an input constant.)
4. Stop the above procedure when no more points have been removed.
5. Average A_{hi} with the previous AVPTS values of A_{hi} to produce A_{hi}' . Do the same for A_{lo} .
6. Use A_{hi}' and A_{lo}' as input values (not to be determined by the fit), remove these offsets from the remaining data points and perform a least squares fit to obtain B and C only.

7. Proceed as above until no more points are removed.
8. Transmit Ahi', Alo', B, C, Sigma and N, the number of points remaining in the fit.

It is important to note:

1. The algorithm does not use Ahi or Alo when fewer than 3 points occur in that gain. To do so would cause the fit to errantly report the value of the offset. For example, if 1 point of high gain were in the buffer, the high gain offset (Ahi) would be set to that value. This is wrong, of course, since the value of the point is simply a small electric field value at the sampled time, not the offset of the high gain amplifier.
2. When rejecting points from the curve, the algorithm always removes their effects from the matrix sums by subtraction. It does not simply recalculate the sums for the remaining points. While being a simpler procedure, this would take at least fifteen times longer to perform.
3. Relatively large offset values (A.I and ALO) produce large sigmas and poor fits. It is easy to see why. In each summation of the points, ALO will be summed 32 times while each point only once. If ALO is large relative to the sum of the points, the floating point value will more or less reflect only the ALO value and lose track of the point sum.

The procedure is invoked with three parameters. On entry, [HL] address the sampled data block, [DE] points to the input

parameter block, and [BC] point to where SPIN should put the results (see Figure 12). The SPIN calculation takes 500 milliseconds or less when running the cpu at 5 MHz (crystal).

EQ(a):

$$F = \sum_{i=1}^M [E(t_i) - (A_{HI} + B\cos(\omega t_i) + C\sin(\omega t_i))]^2$$

$$+ \sum_{i=M+1}^N [E(t_i) - (A_{LO} + B\cos(\omega t_i) + C\sin(\omega t_i))]^2$$

EQ(b):

$$\frac{\Delta F}{\Delta A_{HI}} = \sum_{i=1}^M 2[E(t_i) - (A_{HI} + B\cos(\omega t_i) + C\sin(\omega t_i))]$$

EQ(c):

$$\frac{\Delta F}{\Delta A_{LO}} = \sum_{i=M+1}^N 2[E(t_i) - (A_{LO} + B\cos(\omega t_i) + C\sin(\omega t_i))]$$

EQ(d):

$$\frac{\Delta F}{\Delta B} = \sum_{i=1}^M [E(t_i) - (A_{HI} + B\cos(\omega t_i) + C\sin(\omega t_i))] \cos(\omega t_i)$$

$$+ \sum_{i=M+1}^N [E(t_i) - (A_{LO} + B\cos(\omega t_i) + C\sin(\omega t_i))] \cos(\omega t_i)$$

EQ(e):

$$\frac{\Delta F}{\Delta C} = \sum_{i=1}^M [E(t_i) - (A_{HI} + B\cos(\omega t_i) + C\sin(\omega t_i))] \sin(\omega t_i)$$

$$+ \sum_{i=M+1}^N [E(t_i) - (A_{LO} + B\cos(\omega t_i) + C\sin(\omega t_i))] \sin(\omega t_i)$$

EQ(f):

$$\text{Sigma} = \sqrt{\frac{F}{N-1}}$$

Spin Fit Difference Function F of M high gain points and (N-M) low gain points.

Figure 10. Spin Fit Difference Function

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY
PROJECT : AFGL-701-14
TITLE : LANGMUIR PROBE INST.
SECTION : SPIN FIT DIFFERENCE FUNCTION
FILE : SPIN1.CAD
DESIGNER: PETER R HARVEY
DATE : 7 MAY 86

	A _{HI}	A _{LO}	B	C
$\frac{\Delta F}{\Delta C}$	$\sum^M \cos$	$\sum_{M+1}^N \cos$	$\sum^N \cos^2$	$\sum^N \text{sincos}$ $\sum^N E(t_i)\cos$
$\frac{\Delta F}{\Delta B}$	$\sum^M \sin$	$\sum_{M+1}^N \sin$	$\sum^N \text{sincos}$	$\sum^N \sin^2$ $\sum^N E(t_i)\sin$
$\frac{\Delta F}{\Delta A_{LO}}$	0	N-M	$\sum_{M+1}^N \cos$	$\sum_{M+1}^N \sin$ $\sum_{M+1}^N E(t_i)$
$\frac{\Delta F}{\Delta A_{HI}}$	M	0	$\sum^M \cos$	$\sum^M \sin$ $\sum^M E(t_i)$

Spin Fitting Matrix including offset parameters

	B	C
$\frac{\Delta F}{\Delta C}$	$\sum^N \cos^2$	$\sum^N \text{sincos}$ $\sum^N E(t_i)\cos - A_{HI} \sum^M \cos - A_{LO} \sum_{M+1}^N \cos$
$\frac{\Delta F}{\Delta B}$	$\sum^N \text{sincos}$	$\sum^N \sin^2$ $\sum^N E(t_i)\sin - A_{HI} \sum^M \sin - A_{LO} \sum_{M+1}^N \sin$

Spin Fitting Matrix with offsets removed

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY
PROJECT : AFGL-701-14
TITLE : LANGMUIR PROBE INST.
SECTION : SPIN FITTING MATRICES
FILE : SPIN2.CAD
DESIGNER: PETER R HARVEY
DATE : 7 MAY 86

Figure 11. Spin-Fitting Matrices

2.9 Sawtooth Generator

The SAW module is used to generate sawtooth waveforms on the sphere bias lines. This is nominally useful in the Langmuir Probe mode but for test purposes, can produce diagnostic waveforms on the bias DACs in the electric field mode as well. The module parameters are defined to generate either sawtooth waveforms or square waves. The module has the following entry points:

SAWINIT : The initialization entry requires no parameters and simply copies its default parameters from ROM into RAM where the user commands can change them. Its initial state is OFF (no waveform).

SAWSTEP : The step input does everything. It synchronizes itself word 224 of every minor frame 30. It does the calculation of the bias value and the setting of that value on both boom bias circuits.

SAWDSC : The Digital SubCom function returns the slow status of the package (what has been commanded). On entry, the index into this data is in the [A] register. On exit, the [A] register contains one byte of data.

Theory of Operation

GENERATION. The Sawtooth generation is a simple process complicated by the many things going on in the rest of the system. The process itself involves just the four values:

1. SAWOFF The bias offset DAC value
2. SAWDEL The size of each change to the bias
3. SAWPER The number of steps up and down
4. SAWDIV The divider of the input frequency

Each of these has a temporary value as follows:

1. BIASREG The present bias value
2. DELREG The delta for this series
3. PERCNT The period counter
4. DIVCNT The divider counter

The basic input calls to SAWSTEP are divided by the count in SAWDIV using the temporary variable DIVCNT. Each time DIVCNT is reloaded from the commanded value in SAWDIV, the present bias value is put on the DACs and a new step of the bias value is made.

Before each step the period counter is first decremented and zero checked. If zero, the DELREG is flipped in sign to make the waveform go in the opposite direction, and the period counter PERCNT is reloaded from SAWPER. If not zero, the procedure simply adds the DELREG to the BIASREG. When flipping directions, the BIASREG is not changed.

This causes the BIASREG to repeat the values at the top and

the bottom of the waveform. This is a nice feature since the waveform has exactly the number of steps asked for rather than $N+1$ or $N-1$, etc. If you ask for 128 steps, you get 128 steps up and 128 steps down. Simple.

SYNCHRONIZATION. A very important part of the package is its synchronization code since, for data analysis purposes, one must know what values are on the BIAS DACs at all times. The synchronization accomplishes this by watching the internal word and frame clocks in the BKG module and waiting for minor frame 30 and word 224. This is just before the ELE package sampling begins for the next major frame. Thus, data within major frames always reflect multiples of the sawtooth starting at the beginning of the waveform.

The synchronization operation is performed by simply copying the four commanded values SAWOFF thru SAWDIV into the temporaries BIASREG thru DIVCNT. Commands changing these values are not guaranteed of affecting the waveform until the first sync time (every four seconds).

COMMANDS. The commands for the package simply set one of the six parameters SAWOFF thru SAWDIV, OPTION and SENSOR. The first four of these have already been described.

"OPTION" has two enable bits, one for stepping and the other for biasing. This flexibility allows the package to run the bias stepping algorithm without actually setting the bias DACs. This is required for the 2 seconds when the bias sweeps occur. We want the bias sawtooth to maintain its place in the waveform

regardless of the fact that it shouldn't set the biases for a couple of seconds. The OPTION parameter is used internally by the EXEC mode switcher and by the SWP module. To completely disarm the package, one must set the SAWDEL to zero.

"SENSOR" simply holds the bias DAC pair to set. Of course, since only the spheres have a current mode, this value is set to 1. Bias DACs 1 and 2 get the sawtooth. Just for fun the user can set this value to 3, sending the sawtooth out to the cylinders.

2.10 Bias Sweeps

The SWP module manages the timing and analysis of Bias sweeps on the spheres and cylinders. Sweeps are used to calculate the proper sensor biasing while in the electric field mode and to calculate the temperature and density information in both E-field and Langmuir modes.

The requirements for the voltage mode sweeps are

1. Perform a current sweep in the voltage mode at a programmable time interval, initially 120 seconds.
2. Sweep both sensors (spheres or cylinders) when that boom system is perpendicular to the sun-spacecraft line. Do not sweep both spheres and cylinders at the same time. See Figure 13.
3. Sweep from -360 to +360 nanoamperes in 128 steps that take 500 milliseconds. At each step, measure and store both sensor outputs in LOW GAIN (V1 and V2 or V3 and V4). Note that V2 must be inverted to be consistent with the rest. See Figure 14A.
4. Allow maximum settling time between setting the current level and measuring the value.
5. Analyze both V1S and V2S (or V3S and V4S) curves as defined below.
6. Average the BIAS results from the two curves and set both BIAS DAC's to that value.
7. Transmit the sweep curves, the parameters and results of the analysis if enabled for playback.

Screening out noise from the curves is required and specified as follows:

1. Do not analyse regions of the curve where the measurement is within 10 percent of full scale, positive or negative.
2. Let $V(i)$ be the i th measurement on a scale of -2048 to +2047. If $V(i+1) - V(i) < -N$ then if $V(i+2) - V(i+1) > V(i) - V(i-1)$, replace $V(i+1)$ by $(V(i) + V(i+2))/2$. Otherwise, replace $V(i)$ by $(V(i+1) + V(i-1))/2$. See Figure 16.

Analysis of the curves follows the noise step:

1. Form a new function whose i th value is defined as shown in Figure 16B. Find the minimum value of this function, since the bias changes make the smallest effects on the measured field.
2. If there are two or more minima with the same ΔV , select the one having the smallest bias value (most negative bias current).
3. If the algorithm fails, set the bias current to an alternate value (IBALT). This can only occur if the signal is within 10 percent of full scale the entire time.

Requirements for the current mode voltage sweeps are

1. Perform current mode voltage sweeps at programmable time intervals, initially 120 seconds.
2. Wait until the sensors are perpendicular to the sun-spacecraft line.

3. Sweep both sensors (spheres 1 and 2 only) through 128 voltage steps from -35 to +35 volts. Measure RI1 and RI2 at these 128 points in LOW GAIN only. See Figure 14B.
4. Transmit the curves.

Note: Analysis of the voltage sweeps is left TBD by ground loaded analysis programs. Though the initial specifications for this analysis was finished prior to instrument completion, there was no longer enough memory available in the ROM to do the calculations required.

There are a number of system requirements placed on the operation of this module. Among them are:

1. Sweeps must inform the Fast Digital Monitor bit when they begin. As illustrated by Figure 14, the software finds the correct sun angle, and then waits until the Fast Digital Monitor is calculated. This keeps the sweep action synchronous with the telemetry while not causing too much error in sun angle.
2. The SWP module must coordinate with the sawtooth generator so that the SAW module doesn't continue to set the BIAS DAC's while the sweep is occurring.
3. The SWP module must not place BIAS results on the sensors until the beginning/ending of that sensor pair's spin fit. Otherwise, part of the first fit will be done at one bias value while part of the second will be done at the new bias

value.

The SWP module has the following entry points

SWPINIT This entry has no parameters and simply sets the initial values for the module.

SWPANG This entry is called to report a new sun angle. On entry: [A] is the new sun angle. This routine checks for sun angle related events such as the start of a sweep or the setting of the bias result.

SWPSTAT This entry is used in the Fast Digital Monitor calculation since the SWP module provides one bit of the FDM word. On exit: [A] = 1 for SWEEP-IN-PROGRESS.

SWPEXEC This is a foreground entry point for performing sweeps. If it is the proper time, a sweep will be performed. If not, the code will simply return.

SWPDSC This routine returns the status of the module to the Digital SubCom processor. On entry: [A] is the index into the status block. On exit : [A] holds the status byte.

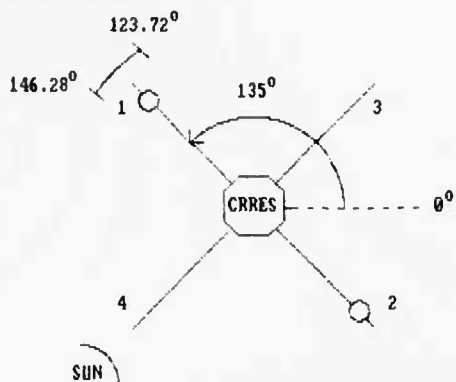


Fig A. U12 I and U Bias Sweeps are taken when the spheres are perpendicular to the sun. Since they take 1-2 seconds to complete, they begin 11.28 degrees prior to perpendicular.

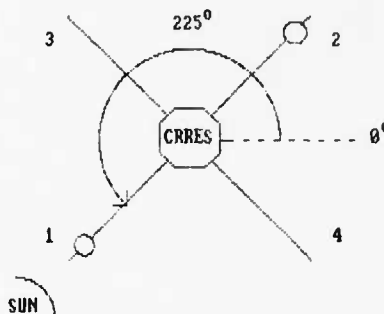


Fig B. Bias results are applied 1.41° prior to when sphere 1 points at the sun (223.59). Spin fits take their first data point at 225 degrees.

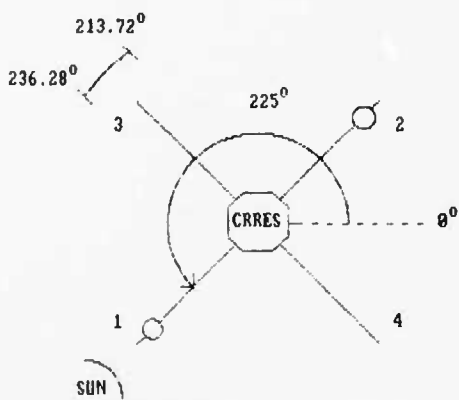


Fig C. Cylinder Bias Sweeps are taken when they are perpendicular to the sun.

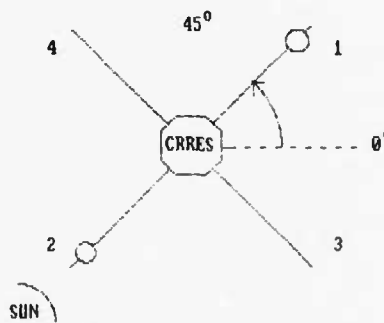


Fig D. Cylinder Bias results are output 1.41 degrees prior to the start of cylinder spin fits.

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY

PROJECT : AFGL-701-14
TITLE : LANGMUIR PROBE INSTR.
SECTION : ORIENTATION OF BOOMS
DURING BIAS SWEEPS
FILE : SWEEP1.CAD

DESIGNER: PETER R HARVEY
DATE : 9 MAY 1986

Figure 13. Orientation of Booms During Sweeps

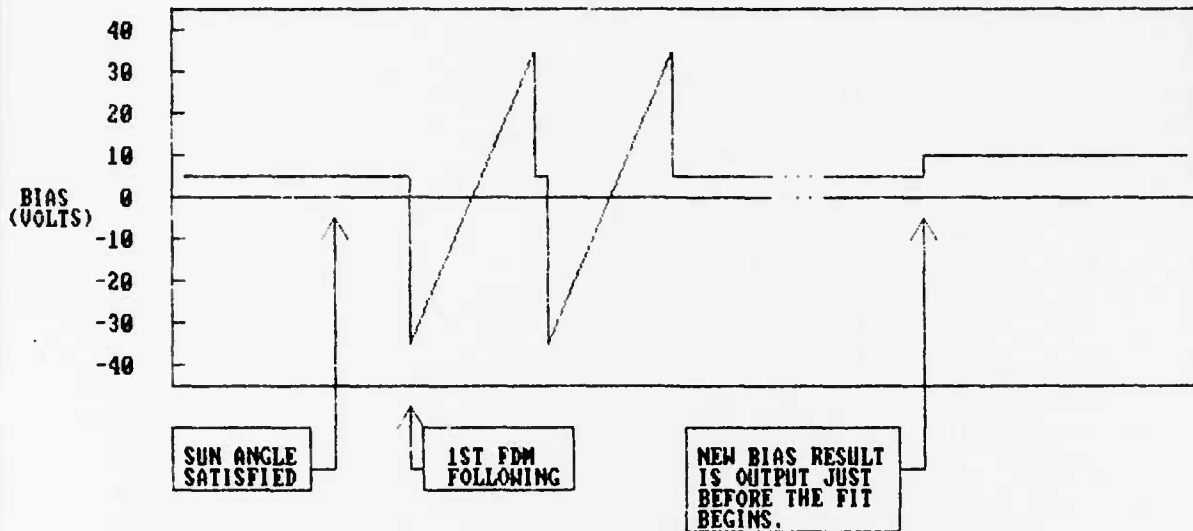


Fig A. Voltage mode bias output on the spheres when controlled by the sweep module.
 Note: Delay between sun angle and 1st FDM varies between 0 and 256 msec.
 Each of the two sweeps takes 512 msec for 128 points.

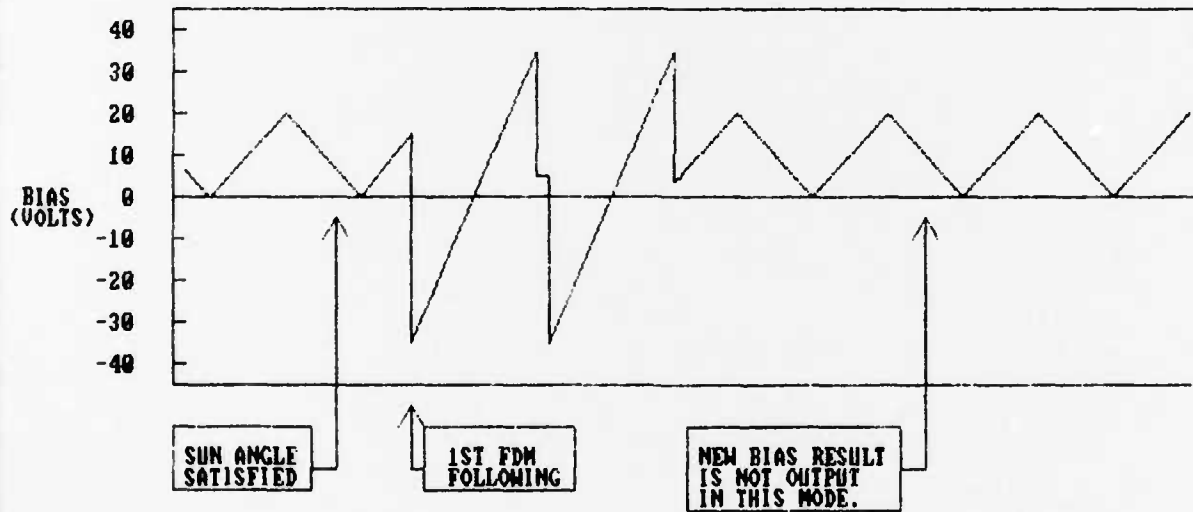


Fig B. Current mode bias output on the spheres when controlled by the sweep module.
 Note: Delay between sun angle and 1st FDM varies between 0 and 256 msec.
 Each of the two sweeps takes 512 msec for 128 points.

UNIVERSITY OF CALIFORNIA, BERKELEY SPACE SCIENCES LABORATORY
PROJECT : AFGL-701-14 TITLE : LANGMUIR PROBE INSTR. SECTION : BIAS OUTPUT TIMING
FILE : SWEEP2.CAD
DESIGNER: PETER R HARVEY DATE : 9 MAY 1986

Figure 14. Bias Output Timing for Spheres

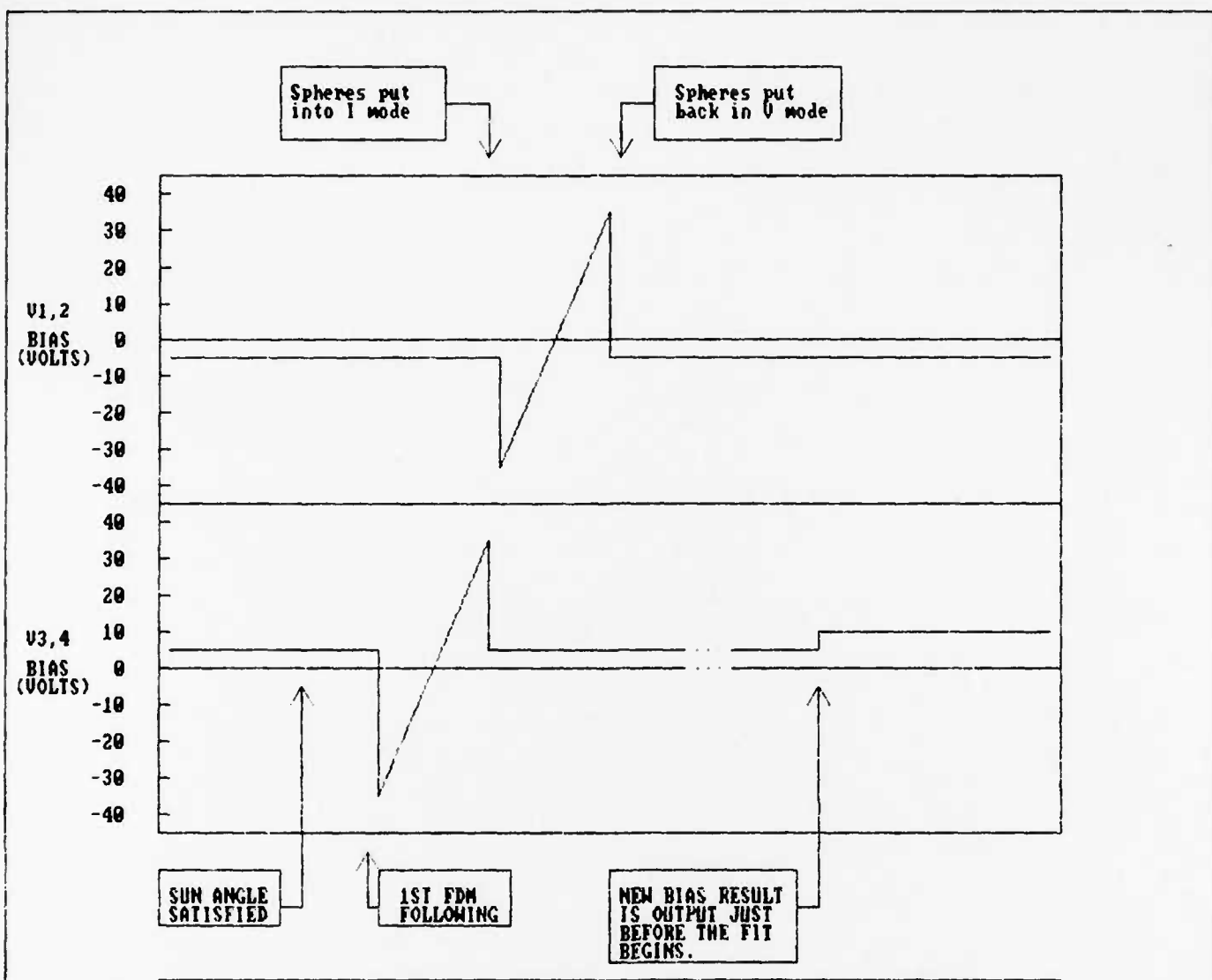


Fig A. Voltage mode bias output on the spheres and cylinders in a cylinder sweep.

Figure 15. Bias Output Timing for Cylinders

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY

PROJECT : AFGL-701-14
TITLE : LANGMUIR PROBE INSTR.
SECTION : BIAS OUTPUT TIMING
CYLINDER SWEEPS
FILE : SWEEP3.CAD

DESIGNER: PETER R HARVEY
DATE : 15 MAY 86

$$\Delta V(i) = \sum_{j=i}^M V1S(i+j) - V1S(i-j) \quad \text{for } (M+1) < i < (128-M)$$

Fig A. Sweep differential function with M point averaging.
For sensors V2S, V3S and V4S, substitute for V1S above.

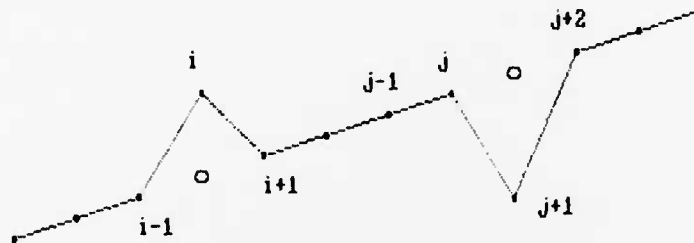


Fig B. Noise points on a bias sweep curve. Circles show replacement point.

UNIVERSITY OF CALIFORNIA, BERKELEY
SPACE SCIENCES LABORATORY

PROJECT : AFGL-701-14
TITLE : LANGMUIR PROBE INSTR.
SECTION : BIAS SWEEP ANALYSIS

FILE : SWEEP4.CAD

DESIGNER: PETER R HARVEY
DATE : 15 MAY 86

Figure 16. Bias Sweep Analysis

Theory of Operation

GENERAL. The SWP module is a complicated module controlling two boom systems in two modes, while conforming to a number of system timing constraints, etc. A large number of options are provided for the sake of flexibility that probably won't be used. If things happen the way they did in the ISEE instrument, the algorithm will be eventually replaced by a commanded program, and that will be that. Nevertheless, the RAM loaded programs may find parts of this code useful and thus these should be described.

COMMANDS. Commands are vectored to the module through the CMDTAB vector table as are all other instrument commands (see BKG). To handle the large number of options and values for the two boom systems' bias sweeps, the SWP module used a simple "index and value" commanding format like the ELE INDEX and QTY commands. One command sets the INDEX variable to point to one of the 32 RAM variables which can be changed. The second command sets the value of the variable and then increments the INDEX.

BOOM SWITCHING. The module works on one boom system at a time. The variable BOOM is either 1 or 3 to indicate spheres and cylinders, respectively. At the end of a sweep cycle, the boom system is switched by the routine SWITCH.

SWEEP TIMING. Sweeps are performed at multiples of the spacecraft spin period. The commanded variable SPINMAX holds the number of spins to wait between sweeps. SPINCNT is loaded from SPINMAX when BOOM is set and is decremented each time the SUN

ANGLE is 0.

ONE SWEEP CYCLE. When SPINCNT is zero (the last spin) and the SUN ANGLE matches the SWPANG value for that boom system, the STATE is set to 1. This indicates to the foreground routine SWPEXEC that a sweep is ready to be done, and this will occur as soon as SWPEXEC is called.

SWPEXEC performs the sweep management which is pretty well described in the software listing. It is worth noting :

1. The SAW module status is saved by SWPEXEC before doing the sweeps, and later re-commanded after the sweeps.
2. The SWPPTR variable is set at the beginning of the sweeps and is incremented by the action of the SAMSTO routine. Since sweeps can be disabled, the number of data points is variable. Hence, the TRANSMIT routine is smart enough to telemeter from the start of the buffer to wherever SWPPTR ends up.
3. When the sweep analysis is done, STATE is set to 2. This allows the BIAS setting to occur.

BIAS SETTING. If the STATE is 2 (analysis done), the SWPANG code waits for the sun angle to match the spin fit angle for that boom system. When these are equal, either the RESULT or the ALTERNATE values are sent to the boom system DAC's, depending upon one of the option bits. Note that the code also checks if the mode is correct for setting bias DAC's. It will not set the sphere DAC's in the Langmuir (current) mode.

SWEEP DETAILS. Here are some notes the sweeps:

1. Sweeps are synch'd with the telemetry system by SWPREQ and SWPOK variables. The FDM routine copies REQuest into the OK variable, so to start a sweep the code sets REQ=1 and waits for OK=1.
2. Relays are flipped only if necessary. For example, if performing a voltage sweep and the instrument were already in the voltage mode, the code will switch to the current mode. It is smart enough (because of table lookup) to know that a current sweep on the cylinders can be performed while the spheres are in the current mode.
3. The first BIAS step is output 4*4 or 16 milliseconds before the first measurement is taken. Thereafter, all measurements are taken at 4.00 millisecond intervals (at words 2, 10, 18 etc).
4. Buffered data is sign extended from 12 to 16 bits. This is needed to simplify the analysis phase and doesn't effect the transmit phase except negative values show high gain.

SWEEP ANALYSIS. The sweep analysis routine (for the voltage mode current sweeps) is contained in the SWP module. Users may override this function by using the ANAVECT vector. When ANAVECT is armed (=0AAH), then the ANA routine used is the one pointed to by ANAVECT+1.

Note that there are 4 undefined variables which are available in each of the two sweep parameter blocks. These are zeroed at SWPINIT and are transmitted as spares in the sweep transmissions. If the user reprograms the bias sweep, these variables should be utilized to indicate that this has been done, as well as possibly to show the parameters of the new function.

MODE BITS. The large number of option bits are handled by two routines CHKENA and CHKWTD ("check enable" and "check-what-to-do"), the two distinguished by the first referencing RAM options and the second ROM options. There is one byte of RAM and one byte of ROM enable/disable bits per boom system. These routines form a one bit mask depending upon the instrument mode and the value of [C]. This mask is ANDed with the option bits and NZ if indicated if a 1 is found.

For example, to determine if transmissions of V1,2 sweeps are enabled, a call is made to CHKENA with [C]=40H. In the voltage mode, the mask will be 40H while in the current mode the mask will be shifted to 80H. These are the bit positions for transmission enables in those modes.

2.11 MAIN Program Loader

One of the most important, yet simplest, pieces of the flight code is its program loader in the LD.A module. Together with various vectors tucked away in critical places, much of the MAIN CPU operations can be changed or increased, and any location in the MAIN memory can be modified. (To load bytes into the Burst computer, one invokes the Burst Program Loader BLD.)

For this module there is only one entry point:

LDINIT : This entry initializes by setting the load address to the beginning of the user area defined for programs.

The available commands, as described in the general description, are

ADRL	Set Low byte of memory address
ADRH	Set High byte of memory address
LOAD	Load byte into memory
EXEC	Execute program

It is important to note that programs must begin with a code "AA" in hexadecimal. This is to prevent an errant command from crashing the system by executing a program not ready to be run. (Had the command count error bit been invented prior to this module, it would have been a better check of correct program loading.) This byte is zeroed before the program is executed, so one cannot re-execute a program by sending a second EXEC command.

MAIN CPU programs must always begin at "USER" (address 2930H). If you want to run something elsewhere, you must load a jump at

USER.

MAIN programs are executed as part of the command execution. They are therefore part of the background and have suspended the foreground until they execute a RET instruction. To get to the foreground, one can change the foreground vector in the EXEC module and then RET.

Without using any leftover data areas in the other modules, programs of about 1.6 to 1.7 KBytes can be loaded. The stack is going to operate in the first 30 to 50 bytes from 2FFFH, but one should never get too close if one can help it.

2.12 Boom Deployment

The deployment of spherical booms is performed by the Deployment module. This software is intended to monitor the lengths of the boom systems as measured by their "turns counter" microswitches and to deploy the boom systems as commanded.

Requirements for this module are outlined as follows:

- 1) Never start a boom motor if its cover is on.
- 2) Turn off a boom unit immediately if any end-of-wire indicator is tripped (not just it's end-of-wire).
- 3) When deploying both booms at the same time, if one boom gets too far ahead of the other, turn it off until they are the same length again.
- 4) An override command can disable points 1 and 2.

For this module there are the following entry points:

DEPINIT : This entry initializes the boom deployment relays and the module by executing a deploy stop command.

DEPSAMP : This entry is used for monitoring the boom status information and making decisions to turn on or off the boom unit motors.

DEPDSC : The DSC entry returns the status of the boom deployment module. On entry: A is an index into the variables used by the deploy module. On return: A contains the value at that location.

Theory of Operation

DEPLOY STATE. The deploy module handles the two booms as separate devices using two nibbles of a state variable DEPSTAT.

Each nibble has four possible states:

OFF The motor has been commanded OFF and is OFF.
PAUSED The motor is currently OFF, but is commanded ON.
STOPPING The motor is currently ON, but is commanded OFF.
RUNNING The motor is currently ON and commanded ON.

SWITCHING. On each call to DEPSAMP, only one boom system is checked for what to do. This is controlled by the LSB in DEPCNT which is incremented on each call. DEPSAMP first samples the microswitches in order to determine the boom lengths. It then looks at the state of the one boom system. If OFF, it does nothing. If STOP, it calls the IO system to turn OFF that boom motor relay. If PAUSED, it compares the boom lengths and restarts the current boom when the other boom is longer. And if already RUNNING, it checks both the microswitches as well as compares the length of this boom versus the other.

LIMIT CHECKING. The microswitch limits which show the covers ON or the End-of-Wire status is done by the routine LIMCHK. This code returns no-carry if there is a problem, such as a cover which is ON. Limit checking can be disabled if there is a sticky switch by a command which sets the OVERRIDE bits. These are OR'd into the boom status bits in the limit checking process and prevent a zero condition (error) in any of these bits.

COMMANDS. The deploy command simply sets the value of the delay limit and then uses the two bits from the deploy command (0, 1, 2 or 3) to set the DEPSTAT variable with the appropriate states

of each boom. To turn ON a boom system, one sets the boom in the PAUSED state from which it will try to turn ON. To turn OFF a boom, one sets its state to STOPPING from which it will turn OFF its motor and enter the OFF state.

2.13 General Utilities

The smallest of the modules has to be the UTIL.A module. Many of the modules in the MAIN computer needed the same very low level functions to be performed. These were grouped into this file since it made no sense to put them elsewhere.

The functions are as follows:

ZERO Zeroes C bytes of memory from [HL] on.
COPY Copies C bytes of memory from [DE] to [HL].
REF Adds [A] to [HL] in sixteen bit fashion. This is
 useful for array referencing.
UNARY Calculates $2^{**}[A]$ and returns the value in [HL]
NEG16 Negates [HL] in 2's complement form.
MARK This routine outputs [HL] to the diagnostic LEDS of the
 development system

ZERO, COPY and REF have been assigned software restarts 1, 2 and 3 respectively. This conserves on memory since the restart instrument is one byte whereas subroutine calls are three.

2.14 Main Input/Output Utilities

The Input/Output module for the Main computer system is contained in a file called "IO.A". The entry points for it are described below:

- IOINIT Initializes the input/output variables so that the module functions correctly.
- GETMASK Reads the interrupt mask of the processor.
On exit: A is the interrupt mask.
- SETMASK Sets the interrupt mask of the processor.
On entry: A is the interrupt mask to set.
- CMDIN Reads the command input shift register.
On exit: [HL] is the current shifted value.
- TMOUT Sets the telemetry shift register.
On entry: [HL] is the value to send to the telemetry shift register.
- SUNSTAT Returns the status of the sun pulses.
On exit: Status is non zero if a sun pulse has occurred since the last call.
- BOOMSTAT Reads the microswitches in the boom deployment units.
On exit: A is E2.R2.T2.L2.E1.R1.T1.L1 where E is the Endwire indicator, R is the right cover, L is the left cover and T is the turns counter. The endwire and covers are active low.
- SAMPLE Sample an analog quantity on the MAIN multiplexor.
On entry: A is the multiplexor address
On exit: [HL] contains the 12-bit value (0XXX).
If Kelley qty, [HL] = 0XXG, where G is the gain.

AGC Sample an analog quantity after selecting its gain.
 On entry: A is a multiplexor address.
 On exit : D is the proper address to digitize.

SETBIAS Sets a bias D/A converter.
 On entry: H is the boom number (0 thru 3).
 L is the 8-bit dac value (-128 to 127).

SETGUARDS Sets a guard D/A converter.
 On entry: H is the boom number (0 thru 3).
 L is the 8-bit dac value (-128 to 127).

SETSTUBS Sets a stub D/A converter.
 On entry: H is the boom number (0 thru 3).
 L is the 8-bit dac value (-128 to 127).

SETVTRIM Sets a Vtrim D/A converter.
 On entry: H is the boom pair (0 or 1).
 L is the 8-bit dac value (-128 to 127).

SETRELAY Sets or resets a relay in the analog/filtering.
 On entry: A is the relay number (0 thru 17).
 Carry is 1 to set, 0 to reset.

SETFILTER Sets a Filter register.
 On entry: H is the filter number (0 thru 6).
 L is the filter value (0 to 255).

SETMUX Sets the multiplexor bits which steer the filtering.

On entry: H holds a 3-bit code for which mux to set.

L holds the value (1 to 5 bits)

Code	Mux effected
000	V12/RI1
001	V2/RI2
010	6-POLE SELECT (2 BITS)
	00 : V2/RI2
	01 : V12/RI1 RAW
	10 : SC
	11 : V34 RAW
011	V1/SC
100	COMBINES CODES 0 AND 1 (L IS 2 BIT)
101	COMBINES CODES 0 THRU 2 (L IS 4 BIT)
110	COMBINES CODES 0 THRU 3 (L IS 5 BIT)

TSTMUX Requests a copy of the multiplexor bits which steer the filtering. On entry: a holds a 3-bit code for which mux to request. On exit: a = the bits not right justified, condition code set.

SETMOTOR Turns on/off the boom motors.

On entry: A is the motor number (0 or 1).

Carry is 1 for on, 0 for off.

SETKLY Cycle the Kelley AutoGain Circuit.

On entry: no parameters.

SETPLA Load the LEPA shift register.

On entry: [HL] is the value to send.

SEND Send data to the Burst computer system.

On entry: [HL] is the value to send.

Note: if the burst is already trying to send to the main system, this information will be lost. See the LPHW.DOC for timing information on SEND and RECEIVE.

RECEIVE Receive data from the Burst computer system.
On exit: If zero returned, the Burst is not ready.
If non-zero, [HL] contain the data.

RWATCH Resets the watchdog timer.
On entry : no parameters.

IODSC Requests the digital status of relays, and dacs.
On entry: [A] is the index
On exit:

A =	0-1	:	VTRIM1-VTRIM2
	2-5	:	BIAS1-BIAS4
	6-7	:	STUB1-STUB2
	8-9	:	GUARD1-GUARD2
	10-16:		FILTER1-FILTER7
	17	:	FILTER MULTIPLEXOR (SEE LPHW.DOC)
	18-20:		RELAYS (K0 IS LSB OF 1ST BYTE)

2.15 Fast Floating Point Utilities

The FFP.A module fulfills the needs of the instrument in performing on-orbit data analysis of the DC electric field. Specifically, the Sine Wave least squares fit subroutine requires the range and precision of floating point. Originally designed to produce least squares fits in 500 milliseconds on an 8085 running at 2.5 MHz, this package is roughly 30 times as fast as it needs to be for CRRES. The one drawback to the package is the fact that it uses only a two byte mantissa (instead of three) and therefore has less precision than full implementations. Nevertheless, the package is ideal for scientific applications of this sort.

The format of the data is SIGN (S), 7-bit EXPONENT (E), and 16 bits of MANTISSA (HL) as follows:

```
-----
!SEEEEEEE!HHHHHHHH!LLLLLLLL!
-----
MSB                               LSB
```

The registers are organized with the current value held in [CDE] and the second parameter pointed at by [HL]. When floating values are stored in memory, they are stored with the exponent byte first and low mantissa last.

The functions available in the package are as follows:

LODFP Loads [CDE] from memory at [HL]
STOFP Stores the result in [CDE] in memory at [HL]
FMUL Multiplies [CDE] by value at [HL], leaving the result
 in [CDE].

FDIV Divides [CDE] by value at [HL]; leaves result in [CDE].
FADD Adds value at [HL] to [CDE].
FSUB Subtracts value at [HL] from value in [CDE].
FCMP Compares values in [HL] and [CDE] using subtraction.
Returns carry and zero flags as appropriate.
FNEG Negates the value in [CDE].
FLT32 Floats a signed 32-bit value in [DEHL] leaving the
result in [CDE].
FIX32 Fixes a floating value in [CDE] leaving the signed
result in [DEHL].
FSQUA Squares the value in [CDE].
FSQRT Takes the square root of [CDE].
MU21 Fast fixed point 8 bit by 16 bit unsigned multiply. On
entry: [A] is the 8 bit value and [DE] hold the 16 bit
value. On exit: [AHL] hold the 24 bit result. This is
a useful utility though it isn't a floating point call.

Underflow and overflow conditions are treated by returning zero
and maximum values respectively.

Useful timing information has been collected under 8085
simulations of the package. These are listed below with respect
to their minimum, average and worst cases. (Multiplying by zero
would be a minimum case for example).

FUNCTION	MINIMUM (cycles)	AVERAGE (cycles)	WORST (cycles)
FADD	76 cyc	300	465
FSUB	97	400	716
FMUL	48	600	1003
FDIV	48	1600	2030
MU21	197	250	298

Table 9. Fast Floating Point Execution Times

These data are useful for estimating the amount of time it will take for the CPU to calculate the floating point result. To convert these cycle times into microseconds, multiply by 0.4. For example, a worst case FMUL will take 1003 X .4 or 400 microseconds.

2.16 Matrix Utilities

The MATRIX.A package is really just one routine which solves up to 4 X 5 matrices in floating point. It has two entry points, IMATX, for defining where the matrix is, its size, and where the solution should go. The second entry point, SOLVE, calculates the solution to the matrix defined earlier and stores it where it was told.

The procedure the standard one which first diagonalizes the matrix to produce the result for one unknown. That value is then substituted back into the equations to remove the unknown value. This gives the result for the second unknown and so forth.

The matrix solver operates on either 4X5, 3X4 or 2X3 matrices. To save time in the spin-fitting solver and to keep the indexing in this module simple, matrices smaller than 4X5 are still stored in the same amount of space as a 4X5 matrix would take. In other words, the 1st element of the 2nd row is always stored in the 6th memory position as if 5 elements were in the top row.

Finally, one other note about the module is its definition of "0.0" when trying to find non-zero elements in the diagonalization process. A common problem with floating point is that small errors get significant when large multipliers are used. (The same problem exists in a smaller way with integers, but that's another story.) As a result, the diagonalization process must stay away from choosing very small non-zero numbers as the radix. This package uses $1/2^{**10}$ as the limit defining a practical zero.

2.17 Trigonometric Functions

The TRIG.A package provides floating point subroutines for the common trigonometric functions. These are used by the spin-fitting subroutine to calculate sin and cosine terms in the matrices. Since data is sampled at fixed intervals in the spin package, the trigonometric functions did not need to be complete and, in fact, only work for 32 discrete angles. Their only real requirement is to be very fast. Hence, the trig functions merely play small games with the angle parameter and then reference a floating point table. Note: the angle ranges between 0 and 31×3 , in steps of 3.

The functions available are:

SIN	Returns [HL] -> Sin(A)
COS	Returns [HL] -> Cosine(A)
SINSQ	Returns [HL] -> Sin(A)**2
COSSQ	Returns [HL] -> Cosine(A)**2
SNCS	Returns [HL] -> Sin(A) * Cosine(A)

2.18 Burst Executive Module

The BEXEC.A module contains the Burst computer executive logic. This software is responsible for initializing all the modules and then distributing any commands which come into the system. Except for some interrupt-timed sampling modes, the Burst computer operates almost completely in the foreground.

The executive has only 1 entry point, namely the cpu reset. All other modules are called by BEXEC. None call it.

Commands from the MAIN cpu enter the system by an interrupt which causes the transfer of data from the MAIN to the BURST IO system. The executive polls the IO system to see if there is a command using the RECEIVE function. When this returns the NZ flag, [HL] contain the command bits.

With a valid command in hand, the executive calls the other Burst modules to see whose command it is. Each module returns a carry when the command is not theirs. If it is theirs, they execute it, of course. Command errors are recorded in the BEXEC RAM area but NO status is reported to the MAIN computer. This could be a place for future improvement.

In order to cut down the power of the Burst system, the executive uses the same trick as the MAIN executive when it has nothing to do. If no command is ready, the BEXEC module puts a HLT (halt) instruction followed by a RET (return) in RAM memory, and then executes it. This causes the CPU to shut down, until the command interrupt from the MAIN system, at which time the Return instruction is executed. The Burst then recognizes the

command-ready and executes it.

Two vestigial diagnostic routines remain in the Burst executive module. Since the Burst memory was no cramped, these were left in for the sake of future problem solving. One is a memory test program which tests the buss memories, not the Burst Memory bank. The other is a diagnostic output routine which displays [HL] to the diagnostic LEDS of the development system.

2.19 Burst Input/Output Module

The Input/Output routines for the Burst computer system are contained in file "BIO.A." The entry points for it are described below:

- BIOINIT : Initializes the input/output variables so that the module functions correctly.
- GETMASK : Reads the interrupt mask of the processor.
On exit: A is the interrupt mask.
- SETMASK : Sets the interrupt mask of the processor.
On entry: A is the interrupt mask to set.
- SETVECT : Sets/Resets the 2KHz interrupt vector.
On entry: If [HL] is zero, the 2KHz int is disabled.
Else the vector is set to [HL] and the int is enabled.
- RECSTAT : Returns not zero if RECEIVE data is ready.
- RECEIVE : Receives data from the MAIN processor over the interprocessor communication lines. If no data is ready, the zero flag is set on return.
On exit : If zero, no data ready.
If not zero, [HL]= 16-bit data.
- SEND : Sends data to the MAIN processor.
On entry: [HL] contain the data to be sent.
- ADPWR : Controls power to the A/D converter circuitry.
On entry: Carry = 1 to turn ON the A/D.
Carry = 0 to turn it OFF.

SAMPLE : Sample an analog quantity on the BURST multiplexor.
On entry: [A] is the multiplexor address.
On exit : [HL] contains the 13-bit value in the format
 (...G xxxx xxxx xxxx)

MEMPWR : Controls power to the memory banks.
On entry: [A] = the bank number to turn ON or OFF.
 carry = 1 for ON, 0 for OFF.

MARSET : Sets the Memory Address Register.
On entry: [BHL] = 18-bit address to set

BANKSET : Sets the start and end banks to use.
On entry: [B] = the start bank to use (0..5)
 [C] = the end bank to use (0..5)

MODESET : Turns on/off the memory autowrite mode.
On entry: [A] = 1 for autowrite, 0 for normal memory.

SECOND : Delays 1 second.

D5MS : Delays 5 milliseconds.

READ : Performs a memory read of the burst memory and returns
the value in [HL].

WRITE : Writes [HL] to the burst memory.

REWIND : Resets the burst memory address register to the start
bank.

MARGET : Returns the value of the memory address register in
[AHL]. Note: since the hardware counter itself cannot
actually be read back, the software simulates the
action of the MAR whenever READ, WRITE, MARSET and
REWIND are used.

SETIO : This function sets the "IOMODE" to parameter [A]. In
IOMODE 1, the carry flag will be set when a command

interrupt occurs. This is used by the BURST sampling procedure so it need only check carry to decide when to stop.

2.20 Burst Format Control

The BFMT.A module controls the sampling format lists. As described elsewhere, the Burst can remember 16 formats, 10 of which are in ROM while 6 are in RAM. This allows for sophisticated programming which might take different Bursts depending upon conditions seen by the MAIN cpu decision maker. The ability to hold multiple lists lowers the time needed to switch between lists since one need only refer to the list, rather than define it each time.

The module has the following functions:

- INIFMT Initializes the module and defines the default RAM formats 10 thru 15.
- SETFMT Sets which format to use. On entry: [A]=format number.
- ADDFMT Add a quantity to the current format. On entry: [A] holds the quantity's multiplexor address.
- ADRFMT Returns the address of the current format in [HL].
- LNGFMT Returns the length of the current format in [A].
- ENDFMT Returns the end address of current format in [HL].

Theory of Operation

The sampling list are contained in two separate areas, one for ROM and one for RAM. Each format is simply a list of bytes ended by an EOL (End-of-List) marker. The 10 ROM lists and 6 RAM lists are placed contiguously in memory so they occupy the minimum amount of space. The code provides only 64+6 bytes of total space for the 6 RAM lists, so this limits the RAM

quantities to a maximum of 64.

Finding the start and end addresses as well as the length of any list is done by linear search. Adding to a given list in RAM simply moves all RAMLIST bytes down one. There isn't much to this, so I won't labor the description. Just look at the listing for more details.

2.21 Burst Program Loader

Just like the MAIN system, programs can be loaded into the memory of the BURST computer system and then executed. Even more so than in the MAIN system, BURST programs can completely change the way the BURST operates, since almost all of the BURST is run in the foreground.

For this module there are two entry points:

BLDINIT Initializes the module and resets the load address register to point to available memory.

BLDCMD Accepts loading type commands:

 BADRH Set high byte of address register

 BADRL Set low byte of address register

 BLOAD Load a byte into memory, increment register

 BEXEC Execute the program.

It is important to note that programs must begin with a code "AA" in hexadecimal. This is to prevent errant commands from crashing the system. As in the MAIN system, this byte is zeroed before the program is executed.

BURST programs load at address 1202H. That is where the code "AA" must appear. The first executable opcode must be loaded at 1203H. The user program area extends from 1203H to nearly 17FFH less 20 bytes or so for the stack. This amounts to approximately 1.5 KBytes.

2.22 Burst Sampling Module

Burst sampling functions are contained in the BSMP.A module. These include controlling the frequency of Bursts, the memory banks used, starting, stopping and playing back data to the MAIN system. This module uses the BFMT module for its format control.

The BSMP module has two entry points:

BSMPINIT This entry initializes the module, sets initial default values such as the frequency and so forth.

BSMPCMD This entry executes Burst commands which are in [HL] registers.

Theory of Operation

INITIALIZATION. The initialization of the sampling module sets up the defaults as follows: (1) use the V12 only format (2) frequency to maximum, (3) burst A/D turned OFF to save power, (4) memory banks 4 and 5 turned ON while banks 0 thru 3 are turned OFF. This allows for small bursts of V12 data to be run with just a BGO command.

BANK SELECTION. In response to Bank commands, the software sets the memory address control hardware so that the sampling uses all of the memory banks which are powered. Routine BSELECT extracts the STBANK and ENBANK (start and end banks) while routine MEMPC turns those memory banks ON (and the others OFF).

A more sophisticated sampling program may wish to utilize only part of the memory at a time. For example, one could take 6 separate Bursts and then play them all out at the same time, or

perhaps choose which to play back and which to junk. The data returned would be HELL to analyze of course.

BURSTING. The process of taking a Burst is as follows:

1. Make sure the A/D converter is turned ON.
2. Compile the current format into a program.
3. Send the MAIN computer information about this burst:
 - a) the real frequency of the burst
 - b) the number of milliseconds to fill the burst
4. Rewind the memory and write a "START" marker
5. Check that the memory is working by reading back the "START" marker
6. Calculate the amount of delay required for this frequency
7. Mark the memory as an "OPEN" file.
8. Execute the compiled program.

STOPPING. The Bursting stops when the MAIN sends anything to the Burst. The BSTOP command turns OFF the A/D converter and "CLOSES" the Burst memory by obtaining the Start and End addresses of the data in the memory. Since closing a file may take the Burst processor several seconds, the MAIN computer waits for the BSTOP command to signal the end of the file closing process.

PLAYING BACK. The PLAY command starts the playback of header and data to the MAIN system. The playback begins with the header which is played by routine PHEAD. Once the header is finished, the data section begins.

The data section is played from the start address (STADR) to

the end address (ENADR) which were determined at the time of file closing. Since memory errors due to cosmic ray upsets, the code does not simply look for the End Mark which was written into the memory.

Playing back may be interrupted by any command from the MAIN processor. Bursts may be played back over and over again, but this is more of a testing level feature than one useful in orbit.

DURATION CALCULATION. The calculation of the time it takes to fill the memory at a given frequency is errant by a small fraction as a result of the late changes to the burst frequency. The calculation tends to overestimate the time by about 5 percent.

This calculation was originally going to be used by the Burst controller module in the MAIN cpu as a way of deciding how long to wait for a Burst. This turned out to be very hard to use. Nevertheless, it is still available for more sophisticated Burst triggering algorithms.

FILE CLOSING. Closing a memory file of data means to find the start and the end addresses of the data in the memory. This is obviously necessary in order to play back the data starting from the first data point and ending with the last. Since the memory has a "wrap-around" capability, the first data point to be played out may occur anywhere in the memory and it takes this routine to figure out where it is.

The first step is to find the End address (where in the

memory the last data point is). This is done by writing an END marker into the memory, rewinding and searching for the END marker.

Step two is to determine if the Burst was short or long. Short bursts occupy less than the amount of available memory while long bursts last long enough to overflow the memory. Short bursts are the easy case, detected by the fact that their "START" marks are still in the memory (at the rewind point). To close the file on a short burst, the routine simply records where it found the START and the END markers.

Long bursts are much harder to close. They have no "START" marker since the wrap-around feature of the memory wiped it out. Long bursts have 1 END marker and an unknown number of PAUSE markers. We know that the very next memory location following the END marker is some data, but we don't know which of the quantities in the sample list it corresponds to. If there are three quantities in the sample record, it could be any one of them. In other words, we must skip over the partial record of data following the END marker (if there IS a partial record).

One way to compute the partial record is to divide the available memory by the length of each record. The remainder is the amount to skip. The available memory would be calculated from the memory size less the END marker and PAUSE markers.

A simpler method was to simply search for the first END or PAUSE marker following the END marker while keeping track of the record elements.

2.23 Burst Compiler

The Burst sampling program compiler is contained in the "BCMP.A" module. It has 1 entry point as follows:

BCMP : Compiles a high-speed sampling program from a sample list.

On entry: [HL] points to an available code area;

[DE] points to a sample list;

[C] = size of the sample list.

[B] = 2 for interrupt type timing

1 for software delay timing

0 for no delay at all.

On exit: [HL] points to the next available memory loc.

Theory of Operation

Naturally, one might ask "Why is there a compiler in the BURST computer system?" The reason is that it is too difficult to write sampling programs which take into account all the right gate delays and parallelism provided by the hardware. The compiled code runs extremely fast because it takes full advantage of these characteristics.

Much of the details on the burst sampling are discussed in the hardware chapter and I won't repeat these. Rather, I list below only a few examples of the code produced by sampling lists of various sizes (the compiler produces binary of course):

SINGLE QTY SAMPLING ROUTINE (NO DELAY).

```
MVI B,f          ;INIT( FREQ= f )
RST 3            ;
LOOP LHL MEM+ADC+QTY1 ;SAMPLE/STORE 1 POINT
JNC LOOP        ;WAIT FOR COMMAND READY
JMP ENDBURST
```

The execution time of the loop is 40 cycles (16 microseconds) which matches the hardware capabilities.

SINGLE QTY SAMPLING WITH A DELAY OF C SAMPLE PERIODS

```
MVI B,f          ;INIT( FREQ= f )
RST 3            ;
LOOP LHL MEM+ADC+QTY1 ;SAMPLE/STORE 1 POINT
MOV L,C         ;DELAY C SAMPLE PERIODS
CALL SOFDLA
JNC LOOP
JMP ENDBURST
```

The sampling frequency is approximately 62.5 KHz divided by (C+1). See the section below for more details.

MULTIPLE QTY (4) SAMPLING PROGRAM WITH INTERRUPT TIMING

```
MVI B,f          ;INIT( FREQ= f )
RST 3            ;
LOOP LHL MEM+ADC+QTY3 ;MEM=QTY1,DIGITIZE QTY2, MUX=QTY3
IN 7FH          ;(DELAY 10 CYCLES)

LHL MEM+ADC+QTY4 ;MEM=QTY2,DIGITIZE QTY3, MUX=QTY4
IN 7FH          ;DELAY 10 CYCLES

LHL MEM+ADC+QTY1 ;MEM=QTY3,DIGITIZE QTY4, MUX=QTY1
IN 7FH          ;(DELAY 10 CYCLES)

LHL MEM+ADC+QTY2 ;MEM=QTY4,DIGITIZE QTY1, MUX=QTY2

MOV L,C         ;DELAY C INTERRUPTS AT 2KHZ
CALL INTDLA

JNC LOOP        ;LOOP UNTIL CARRY SET (CMD RDY)
JMP ENDBURST    ;THEN QUIT
```

The sampling frequency for the loop is $2/C$ KHz, where C is the value contained in the C register.

BURST FREQUENCY UPDATE. A late modification to the Burst frequency changed it from 59.5 to 62.5 KHz and the software was not able to be updated. This causes the SOFT-DELAY routine to delay more than the equivalent sample by 5 percent. If C=1 in the above case , this would cause the overall frequency to be 2.5 percent below 31.25 KHz or 30.49 KHz. A correct calculation of Burst frequencies corresponding to the parameter RFREQ and the size of the record NREC is given below:

1. Find NMAX in the table corresponding to RFREQ.

RFREQ	14	13	12	11	10	9	8
-----	-----	-----	-----	-----	-----	-----	-----
NMAX	1	2	3	4	6	10	20

2. Calculate the number of delays used:

$$NDLA = NMAX - NREC$$

3. Calculate the Frequency as follows:

$$\text{Frequency} = 1000000 / (NREC * 16 + NDLA * 16.8)$$

Only high frequency bursts (over 3 KHz) use software delays to regulate the frequency. Interrupt regulated frequencies (2 KHz and under) are not affected.

2.23 Burst Floating Point Utilities

The Burst computer system comes complete with its own floating point package for the simple reason that there was plenty of room in the ROM, that the DURATION calculation was facilitated and third, some ground loaded program may be able to use it. For the sake of sanity in file-keeping and in order to simplify the assembly process, it is contained in the BFFP.A file, not in the FFP.A file.

The entry points are exactly the same as those in the MAIN floating point package so please refer to that section for further information.

3. Hardware

The function of this chapter is to describe the circuitry and physical characteristics of the CRRES Langmuir Probe Instrument (AFGL-701-14A) and its spherical sensors. It is assumed that the reader understands related parts of the general instrument description.

This chapter describes each of the seven boards of electronics in 701-14A. The functions of each board are summarized below:

Analog Board	:	Sphere and Cylinder Sensor Interfaces AC Instrument Interface
Filter Board	:	Filtering of sensor signals Fluxgate and Search Coil Interfaces
IO Board	:	D/A conversion for bias control Relay controls for sensors and Analog Filter board control
MAIN CPU Board	:	Telemetry Formatting, Command Reception MAIN A/D Conversion
BURST CPU Board:		High Speed Data Sampling Burst Memory control circuitry
Memory Boards	:	3 Banks of 32 Kbytes on each board Individually powered

3.1 The Analog Board

The "Analog" board is so called because it handles the primary analog inputs and outputs of the instrument. The primary input signals are the sensor inputs from the two spheres and two cylinders. The outputs are the sphere voltages which are repeated to the IOWA Sounder instrument.

SPHERE BIASING. Each spherical sensor has three control signals called BIAS, STUB and GUARD. These are generated by "offset" circuits which add the +/- 2.5 Volt signals (supplied by the IO board) to the voltage output from the sphere preamplifier. As a result, these voltages track the sphere potential up or down. The range of the STUB and GUARD voltages is approximately +/- 37 Volts, while the BIAS voltage ranges between +/- 1.25 Volts. For example, if the IO board supplies a +2.5 Volt signal to BIAS 1, and the sphere 1 preamp is 1 Volt, then the BIAS 1 output will be 37+1 or 38 Volts. The maximum range of these outputs is +/- 100 Volts.

SPHERE MEASUREMENTS. Each sphere produces two outputs called V/I and I/V. The V/I output represents the voltage measured on the sphere when the sphere is in the voltage mode, and the current collected when in the current mode. The I/V output is just the reverse of these.

The "voltage" output from the sphere 1, for example, is current amplified by the opamp U5 and fed four places: 1) the GUARD offset circuit, 2) the STUB offset circuit, 3) the floating power supply driver and 4) the signal divider. The floating power supply driver (sphere 1) is comprised of U5, Q13, Q14, etc.

and simply keeps the ground for the sphere components near the preamp's output voltage. In this way, the sphere preamps can float between -100 and +100 Volts with respect to the instrument ground, while all the sphere op amps need only operate from +/- 12 Volts. The +/-100 Volt signal from U5 is divided down to +/-5 Volts by the resistor divider R79/R80 and is then buffered by U1 for exit to the filtering board.

The "current" output from sphere 1 is really the voltage from the DAC plus the collected current times the resistor in the sphere. The op amp U4 is used to remove the DAC voltage from this measurement while amplifying the signal a factor of ten. The resulting voltage, RI1, is fed to the multiplexor U3 and optionally to the Sounder interface circuit.

SPHERE RELAY CONTROL. Each sphere contains two relays which determine whether it's in the voltage or the current mode. For lack of wires both relays are controlled using the same wire. Relays K2 and K3 control whether power is sent down the mode control line. Relay K1 determines whether +35V or -35V will be sent. Each sphere relay circuit uses diodes to determine which coils should get the power based upon the polarity of the signal. Note that the sphere relays are 12 Volt relays, but since the 50 meter wire is so resistive, 35 Volts had to be used on this end.

CYLINDERS. Control of the bias current to the cylinders and receiving the measurements proceeds in much the same way as the spheres. However, no floating power supplies are used as the cylinder amplifiers work from fixed +/- 35 Volt supplies.

Relays K4 and K5 pulse the relays inside the cylinder preamplifiers to either put biases on the cylinders or not. Relay K6 holds closed the calibrate relay inside the cylinder preamp.

SENSOR DIFFERENCING. The main measurement of the instrument is the difference in voltage between the two spheres and between the two cylinders. This difference is done by opamp U4 (for V1 - V2) in a way that allows for the introduction of a trimming potential from the IO board. This trim potential will allow for the near-zeroing of any offsets which arise because of radiation effects to the opamps.

DIFFERENCING AMPLIFICATION. As a precaution against possible noise problems in the system, the Analog board provided an extra amplifier for getting a high gain version of the difference measurement. These signals are called V12X50 and V34X50, and proceed directly to the inputs of the MAIN analog multiplexor.

DERIVATIVES. Both difference measurements are also differentiated to make what are called "AC" signals V12/RI1 AC and V34 AC.

RELAYS. There are two types of relays on the Analog board, latching and non-latching. The former require current to be applied to their coils for the entire time they are flipped. They flip within 2 milliseconds after current is supplied.

The latching type require only a 2 millisecond pulse directed down one of its two coils, one to set and one to reset the relay. In order to cut down on the number of drive

transistors required for all these coils, a separate relay (either K1 or K0) is used to determine whether current will flow down the set or the reset coil of a relay. This is illustrated below:

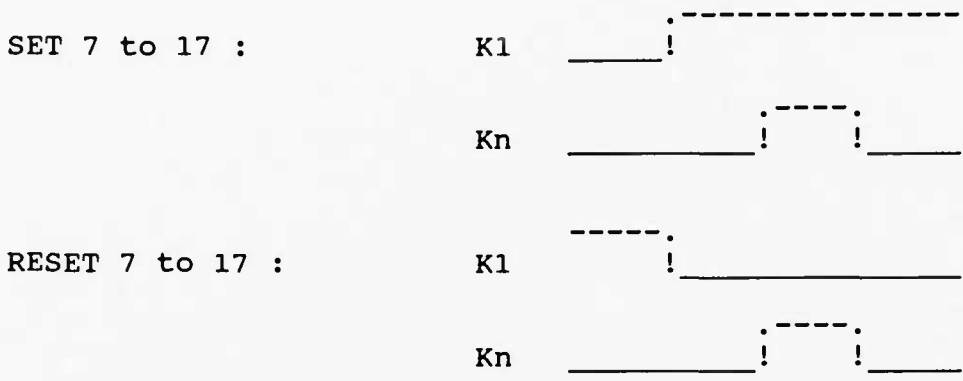


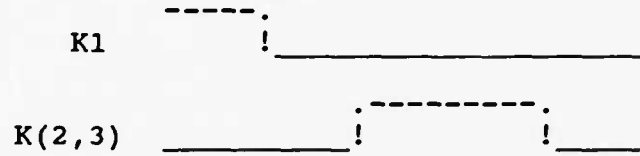
Figure 17. Relay Set and Reset Logic

The relays which are in the spheres and cylinders require extra time to flip since they are only controlled by relays on the analog board. For example, to flip relay K18 (sphere 1), relay K2 must go into the set position. We must wait for K2 to settle, then wait more for K18 to settle. Extended relays like these are flipped as seen below:

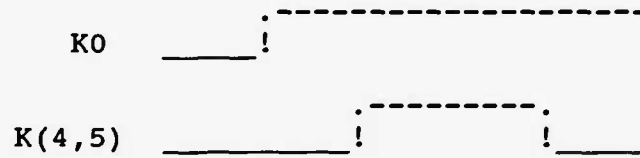
SET (18, 19) :



RESET (18, 19) :



SET (20, 21) :



RESET (20, 21) :

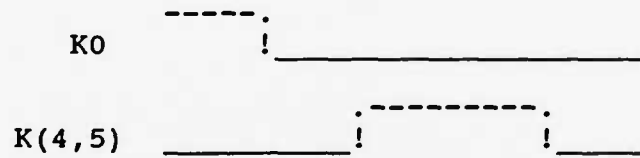


Figure 18. Extended Relay Set and Reset

3.2 The Filter Board

The filter board has three major functions: 1) to interface with the Fluxgate Magnetometer, 2) to interface with the Search Coil Magnetometer and 3) to filter the above signals as well as the signals from the Analog board.

FLUXGATE INTERFACE. The Fluxgate signals (X,Y, and Z) come into the board in +/- 10 Volt form (see page 2 of the Filter drawings). The BY input can be multiplied by a factor of 6 using the amp U27 when selected by multiplexor U31. The three signals are rolled off to 60 Hz and converted to +/- 5 Volts by one stage of opamps U32 and U34. These three signals are called BXFAST, BYFAST and BZFAST, and are sent to the IO board for multiplexing into the Burst computer system.

A second stage of filtering, again using U32 and U34, rolls off these signals to 6 Hz for input to the MAIN multiplexor U29.

SEARCH COIL INTERFACE. The Search Coil Magnetometer measurement comes to the Filter board as a differential signal. Amp U22 converts this to a single ended signal, ready to be used by multiplexor U23.

VARIABLE FILTERS. On the board are seven programmable filters which are used for signals heading for the Burst computer system. Since burst collections can be taken at frequencies from 10 to 60 KHz, the programmable filters are used to stop aliasing while sampling at the lower rates. The heart of each filter is the D/A converter located in the feedback loop. The resistance in the DAC changes the rolloff characteristics. (See the filter

rolloff figure below).

FIXED FILTERS. For the sake of telemetry sampling, signals input to the MAIN multiplexor are rolled off at 1/2 their approximate sampling frequencies.

BAND FILTERS. As a gauge of AC activity on V12/RI1 the Filter board has three comb filters called F1, F2 and F3, whose notch frequencies are 32, 256 and 2048 Hz, respectively. These are drawn on page 2 of the filter diagrams.

KELLEY GAIN CIRCUIT. Finally, the analog or clean part of the KELLEY automatic gain circuit is found on the Filter board. The digital section on the IO board controls the operations of this circuit. See below for the description.

Table 10. Filter Rolloffs

<u>FILTER</u>	<u>BURST QTY</u>	<u>FILTER MAX FREQUENCY</u>
1.	V12/RI1 AC	11.5 KHZ
2.	V2/RI2	11.5 KHZ
3.	V12/RI1	25.5 KHZ
4.	DIRECT	11.5 KHZ
5.	V1/SC	15.0 KHZ
6.	V34	18.0 KHZ
7.	V34 AC	12.0 KHZ

3.3 The Input/Output (IO) Board

The primary functions of the IO board is to interface the central processor boards (MAIN and BURST) with both the ANALOG and FILTER boards. The fundamental reason for a separate board comes from the requirement that all analog signals must be kept as clean as possible of digital noise.

There are six subcircuits to the IO board:

- 1) Bias control
- 2) Relay control
- 3) Boom status information
- 4) Filter control
- 5) Kelley Automatic Gain Control (Digital part)
- 6) BURST multiplexing and Gain control

SERIAL/PARALLEL CONVERSION. The IO board is controlled by a total of 6 lines over which information is passed serially. The historical reasoning behind this was to minimize wires travelling between the digital and analog sections of the instrument and, by such an arrangement, optimize the ability to shield fewer wires, etc. As it stands, this vestigial protocol could be removed with little overall effects to cleanliness, while vastly improving the bandwidth of processor control to the board.

BIAS CONTROL. The bias currents which go to both the spherical and cylindrical boom systems are controlled by eight 8-bit DAC's (packaged in 4 AD7528 dual-DAC's).

RELAY CONTROL. The relay control circuit is responsible for providing power to the relay coils on the analog board. The

circuit can simultaneously provide coil power to a total of three relays: K0, K1 and 1 of K2 thru K17. (Note: there are no relays K12 and K15). This is accomplished using 7 bits from the IO shift register: 2 bits directly control the K0 and K1 coils through coil-driver circuits; 5 bits control relays K2 thru K17 using one disable line and 4 address lines to select 1 relay coil using 2 3-to-8 decoders.

Each coil driver consists of a 4050 gate, a base current limiting resistor and a PNP transistor. The normal "OFF" condition is a high level of the 4050 which causes no current to be drawn thru the transistor. A low level on the 4050 turns "ON" the transistor and pulls roughly 20 times that current through the relay coil which is attached between the transistor and the -7 Volt supply.

The minimum output sink current of the 4050B is roughly 2ma when the 4050 is commanded low and its output held at 0.4 Volts. The minimum current delivered thru the base is limited by the 2.2K resistor which at 4 Volts is around 2ma. With a gain of 20 through the transistor, deliverable current to the relay coil is approximately 40ma. Smaller resistors (3.3K) are used for the latching type relay drivers and limit deliverable current to 25 ma each. Since the relay coil resistances are 390 and 500 Ohms for the non-latching and latching types, respectively, each coil will see 12 volts across it when commanded. This retains a 25% margin over the minimum guaranteed switching voltage for these relays.

The power dissipated in each transistor itself is $.25V \times$

40ma = 10 milliwatts which is safe for continuous operation.

BIAS CONTROL. The eight biases which can be applied to the spheres, the stubs, the guards and the cylinders, are controlled by circuitry on the IO board. Each bias voltage generated is a bipolar signal which gets added to other signals on the analog board.

Each bias voltage is made using 1/2 of a 7528 dual D/A converter whose reference is tied to +5 Volts. The current output is converted to a voltage by an inverting amplifier whose output is in the 0 to -5 Volt range. This signal is then subtracted from a -2.5 Volt reference by a second op amp whose output is bipolar and centered on zero Volts.

Each 7528 is programmed by its 8 parallel input lines and chip select which are attached to the serial shift register mentioned earlier. The write and DAC select (A/B) are directly controlled by MAIN computer output pins.

Note that the sharing of 3 lines between the bias circuitry and relay control has the limitation that one cannot hold relay coil K2 thru K17 while changing the bias D/A values.

BOOM STATUS. The IO board is also responsible for making motor currents, length potentiometers and thermistors into voltages for the MAIN A/D multiplexor. Converting motor current into voltages involves simple op amps. Measuring the lengths and temperatures merely requires pull-up resistors.

FILTER CONTROL. The IO board controls the low pass filters and multiplexors of the Filter board with 8 strobe signals, 8

data lines, and 5 multiplexor bits. These are shifted into a 24-bit shift register composed of 3 4094 shift registers. This 24-bit register re-uses the same clock and data lines as the 16-bit register for biasing and relay control, but has a separate strobe line. Thus, one can program only the 24-bit or the 16-bit register at a time.

KELLEY GAIN CONTROL. The digital control section for the automatic gain circuit consists of an up-down counter (54193), a latch (4042) and some random logic. On a rising edge of "GAIN CTL MAIN", the 4042 latches in the OVER and UNDER signals which are determined by comparators on the Filter board. These latched values are combined with the strobe signal and limit checking logic to provide either a "count up" or "count down" signal to the 54193. Limit check is performed to keep the gain in the range of 0 to 13.

BURST MULTIPLEXING AND AUTOGAIN CONTROL. The BURST multiplexor circuit is placed on the IO board instead of the Filter board because of the need to switch its mux address at high speed. Between the Filter board and the IO board are isolation ("feedthru") filters which keep the Filter board free from digital noise.

The multiplexor has 16 input channels which are addressed by BURST MUX A0 thru A3. The output goes to both "times 50" and "times 1" amplifiers as well as a bipolar comparator operation. The comparator is formed by 2 CMP04 comparator stages, one for positive and one for negative signals. Their outputs are open collector and thus pull down when they are active. The limit

used to decide whether the signal is "high gain-able" is roughly 1/65th of full scale.

The "times 50" amplifier has one additional feature which speeds up the process of preparing a quantity for digitization. The feedback loop is clamped so that the op amp cannot saturate. Once in saturation, this type of amplifier can take many microseconds to work properly again. Thus, no matter the prior state of the multiplexor, the high gain amp will be ready for the new quantity.

Figure 19. Filter Board Control Register Definition

```
-----  
!K MMMMM!DDDDDD!FFFFFF!  
-----  
MSB                               LSB
```

where F represents the Filter Strobes as follows:

(LSB) 0: VTRIM
1: V12/RI1AC
2: V2/RI2
3: V12/RI1
4: DIRECT AC
5: V1/SC
6: V34
(MSB) 7: V34AC

and D is 8-bit data for the filters as follows:

1: FILTER AT LOWEST ROLLOFF VALUE
255: FILTER AT HIGHEST ROLLOFF VALUE
0: FILTER UNDEFINED (DO NOT USE)

and M represents the Filter Multiplexor bits as follows:

(LSB) 0: V12/RI1 CONTROL --- 0 = RI1 1 = V12
2,1: 6-POLE SELECT --- 00 = V2/RI2
01 = V12/RI1 RAW
10 = SC
11 = V34 RAW
3: V1/SC CONTROL --- 0 = SC 1 = V1
4: BY/BY6 CONTROL --- 0 = BY 1 = BY*(-6)
5: V2/RI2 CONTROL --- 0 = RI2 1 = V2

and K represents the Kelley Gain Control circuit reset

1: Holds Kelley Gain Control circuit reset
0: Allows gain circuit to operate

Figure 20. Analog Board Control Register Definition

```
-----  
!DDDDDDDD!AAAAAAA!  
-----
```

WHERE D is the data to the DAC's or the relay number

A represents the Analog select bits as follows:

0:	Bias 1/2 DAC Select	(Active low)
1:	Stub DAC Select	(Active low)
2:	Guard DAC Select	(Active low)
3:	Bias 3/4 DAC Select	(Active low)
4:	Disables Relays K2-K18	(1=disabled)
5:	Relay Set Select	(0 for K2-9)
6:	K0 coil	(Active low)
7:	K1 coil	(Active low)

3.4 The Main Processor Board

The main processor board is the heart of the instrument. Information flows to and from the outside world to all parts of the instrument through this board. It's basic purpose is in digitizing data, formatting telemetry, and receiving commands from the ground. Associated with those services, this processor controls the analog and filter boards, calculates magnetic field data for the LEPA instrument, and controls the Burst computer. The following paragraphs describe the subcircuits of this board.

THE PROCESSOR. Central to the design is the processor itself (U2), a Sandia SA3000 which is really just the radiation tolerant version of the Intel 8085 (although in CMOS). The input crystal frequency used is 5 MHz which produces an internal cycle time of 400 nanoseconds. This is a lower frequency than is possible with the part (post-radiation) and was chosen since it would both increase the reliability of the part as well as lower its power consumption.

Connected to the processor buss for compatibility with the OKI processor are pullup resistors (U29) which give the buss some direction whenever the OKI 8085 part is tristated. This occurs at reset and whenever the cpu executes a HLT instruction. The board thus supports both the Sandia and OKI manufactured parts. However, if the Sandia part is to be used, the resistor package should not be installed since the Sandia has weak output latches which might fight with the weak resistors.

One item to note about the Sandia cpu, however, is that it tends to not start up using the standard crystal interface (2

10pF capacitors on pins 1 and 2). Instead, Sandia uses the less common interface as shown on the drawing (1 10 Mohm between pins 1 and 2, 1 20 pF capacitor on pin 1 only).

The 8085 multiplexes its low 8 address bits with its data bits and this causes problems for ROM and RAM devices which require the low address bits to be stable for the entire READ cycle. For this reason a buss latch (U3) and pullups (U47) are included. The latch holds the low 8 address lines which are strobed out by the processor while ALE is high.

INTERRUPTS. The processor can be interrupted by any of five types of events. In order of priority from top to bottom they are 1) WatchDog Timer Overflow, 2) 1 KHz clock, 3) Major Frame pulse, 4) Command Envelope, and 5) RC Int. Interrupts 3 and 4 are simply the filtered signals from the spacecraft interface. The 1 KHz clock is the 2 KHz telemetry synchronous clock divided by 2 by the flip-flop U9A. The WatchDog Timer Interrupt is the very highest priority interrupt and is connected to the Non-Maskable-Interrupt of the 8085 processor (see below).

The RC Interrupt is used by the software to shut off the cpu for a time while the A/D converter is operating. The software simply outputs a "1" to the RC using U14 pin 1 and goes into HALT. The interrupt wakes up the processor after approximately 1 RC time of 33 microseconds so it can get the results of the conversion. (the conversion completes in 11.2 microseconds.)

WATCHDOG TIMER. The Watchdog timer is a small circuit (U5) which is intended to "wake-up" software which has crashed or gotten lost as a result of a Cosmic Ray upset, other radiation

damage or errant commanding. In actuality, it is a counter which is incremented by the Major Frame pulse (every 4 seconds) and reset by software. If the software does not reset this timer/counter in any major frame, the WatchDog will stop the software by issuing a Non-Maskable-Interrupt.

WAIT STATE GENERATOR. Since some of the design uses 4000 series CMOS devices which are not fast enough (over temperature) to drive the 8085 bus at 5 MHz, a wait state generator is included to provide a small delay when these devices are addressed. More specifically, the wait state is applied whenever an address of 8000H or greater is used. All CMOS devices which need a wait state are given an address in this range.

The wait state works as follows. The falling edge of ALE clocks valid A15's into the flip-flop U10B. If A15 is a 1 (address above 8000H) then the inverted Q goes low which declares a NOT-READY condition. This will cause the 8085 to delay its READ or WRITE cycle until RDY is 1 when it begins a clock cycle. The wait state generator releases control of the 8085 on the first rising edge of CLK since U10A takes A15(=1) into its D input, causing its -Q output to go low, thus resetting U10B. This action causes the -Q of U10B to go high, which re-enables the RDY input.

ADDRESS DECODING. Devices are distinguished from one another by the 3-to-8 line decoders U22 and U25. Both are Sandia 2995's which are emulations of the standard 74LS138 device. U22 handles addresses over 8000H while U25 handles addresses below

8000H. A full address map is given below.

READ ONLY MEMORY (ROM). The program memory for the computer is contained in a pair of Raytheon 29673 SMB ROMS, each holding 4096 8-bit bytes of information. To lower the power consumption of these devices, 5 Volts is provided only while they are selected. This is accomplished using the transistor circuits Q1/Q2 and Q3/Q4. U23C and U23D provide logic level 1 whenever the respective ROM is being selected by the decoder. This high level causes current to flow through the base resistor (R55 or R52) turning ON the npn transistor (Q1 or Q3). The collector is pulled near ground which causes the base of the pnp transistor (Q2 or Q4) to turn ON. This causes the ROM power pin to be pulled up near 5 Volts. For full TTL-CMOS compatibility, bus pullups (U30) are used on the outputs of these ROMS.

RANDOM ACCESS MEMORY. The RAM for the main computer system is comprised of 8 5114's, each of which organized as 4 bits by 1024. These are arranged in pairs, one taking the high nibble and one taking the low nibble. The addressing of these pairs is done with another SA2995 decoder (U26), coupled with a READ-or-WRITE signal from gate U23A. This strange configuration is brought about by the fact that the 5114 memories are unlike most microprocessor compatible memories; i.e. they have no READ input. To READ a 5114, one merely has to select it and not WRITE to it. To WRITE to a 5114, one must signal WRITE to it before one selects it. Hence, the select signals for these chips must arrive during the processor READ cycle or WRITE cycle.

As a final note, both the NAND gate and the decoder must be fast in order for this circuit to work. The response time of the circuit is the sum of the NAND, the decoder, and the RAM itself. Since the processor takes 3/2 cycle (400 ns each) the response of this circuit must be under 600 nanoseconds. The 5114 responds in 400 nanoseconds and so the NAND and decoder must total less than 200 nanoseconds (worst case).

SERIAL CONTROL REGISTER. The main processor controls the operations of the Analog, Filter and IO boards using a serial protocol in order to minimize the number of interboard connections needed. U4 is a 4034 register which is employed to latch data from the processor for the other boards. Software actually serializes the data and presents it to the port.

BOOM STATUS INPUTS. Microswitch closures from the boom units are input to the microprocessor via U18 which is another 4034 register. These inputs are not filtered in hardware (which saves some components) and are expected to be software de-bounced.

BOOM MOTOR CONTROL. The two boom motor relays are contained in the power supply unit of the instrument (for height reasons). The control lines for those relays consist of four lines, one wire for each coil; i.e. an ON COIL and OFF COIL for two relays. Each control wire uses an inverter (for current), a base resistor and a transistor. A logic level 1 from the port to any of these four wires energizes the respective relay coil. At reset, the 4034 is tri-stated so that the 100K resistors will guarantee the

motor coils will be off.

COMMAND SHIFT REGISTER. U24 and U27 are a pair of 4094 shift registers each having 8 bits. The resulting 16-bit register is clocked into the register by the falling edge of "COMMAND CLOCK" (after it has been filtered etc). The COMMAND ENVELOPE signals an interrupt to the processor all the time the command is being shifted into the register. Software can acknowledge the interrupt when shifting begins but must wait until the shifting ends to receive the data.

TELEMETRY SHIFT REGISTER. U31 and U32 form a 16-bit telemetry register using a pair of 4021 parallel-to-serial converters. Each is clocked out by the rising edge of GATED SHIFT CLOCK. The very first bit is therefore valid only until that first rising edge.

LEPA SHIFT REGISTER. The main processor sends reduced magnetometer data to the LEPA instrument via a 16-bit register (U19 and U20) using exactly the same protocol as the telemetry shift register. The LEPA instrument provides the pulses when needed to shift data out. Status of the register is provided by the main processor using the MAG BUSY signal (U34 and Q6).

SUN PULSE COUNTER. Once per spin while the spacecraft is in sunlight, a sun pulse will occur. Each pulse toggles a flip-flop (U9B) which can be read in the most significant bit of U12. The software defines a sun pulse as a change in the polarity of the flip-flop.

BURST CONTROL. The Burst computer can be reset using the "BRESET" output of U34. This is useful for stopping whatever the Burst computer was told to do, errant commanding, crashed programs, or whatever.

THE MAIN MULTIPLEXOR. The multiplexor circuit starts with a register to hold the multiplexor quantity (U13). Some of these bits go to another part of the multiplexor which is on the IO board (in a clean area). Two eight line multiplexors (Harris 508A's) are on the CPU board (U8 and U21). U21 is used for selecting analog monitors while U8 is used to select between the offboard multiplexor outputs. The output signal is held by a .01 microfarad capacitor (C8) while U8 is disabled (holding pin 2 low).

The signal is amplified by op-amp U16B since the multiplexors have a large impedance over temperature which would contribute to signal noise. Also, the input to the AM6112 is a very low impedance to the internal DAC. Pretty hefty spikes come out of the AM6112 input pin and the opamp takes care of these.

Another amplifier is needed to buffer the analog housekeeping values which pass through U21. Something that isn't apparent when one first looks at the U8 to C8 connection is that it has a very low impedance (maybe 200 to 1000 Ohms). Any input to U8 must be buffered by an amplifier. This is true of any science quantity already, but the analog monitors are simply resistors and such. U16c buffers the signals from those monitors before going into U8. (It is amazing to me that the ISEE

instrument had no such buffer.)

ANALOG TO DIGITAL CONVERSION. The analog to digital conversion circuit is comprised of three parts, namely, the A/D itself, a clock, and a 5 Volt reference. The A/D converter is a microprocessor buss compatible device made by Advanced Micro Devices (the AM6112). It can digitize a 12-bit quantity in as little as 4 microseconds (according to the company promises) but is in this circuit set to convert in 10.0 microseconds. The flip-flop (U46A) divides the processor clock by 2 in order to clock the 6112. Each clock period is 800 nanoseconds and a conversion requires 12.5 cycles.

Initially, the A/D timing was a simple delay loop in the conversion software. After the prototype was completely assembled and closed up against outside noise, it was found that the A/D was missing codes. Testing proved that the 6112 was sensitive to the operation of the CPU, even though the manufacturer claims this isn't true. With the addition of the RC interrupt circuit, conversions were greatly improved.

Finally, the reference circuit for the 6112 converter is composed of a Precision Monolithics REF-02 (U7) followed by an op amp (U16A). One may notice if one reads the information on the 6112 that it has an internal reference circuit. While this is true, it has been shown to be rather rad-soft; i.e. the reference voltage drifts rather badly after radiation. The REF-02 doesn't have such a problem in radiation. Variable resistor R70 is provided to trim the reference circuit so that U6 pin 20 is

2.5000 Volts.

SUMMARY OF MAIN COMPUTER I/O. Below is a short summary of the input and output ports decoded on the MAIN processor buss.

Analog to Digital Converter Control (STA 5001H)

!.....MMB!

!!`----- 1 = 2'S COMPLEMENT OUTPUT, 0 = OFFSET BINARY
!`----- MODE = 00 : WRITE STARTS CONVERIONS
`----- 01 : READ STARTS CONVERSIONS
10 : READ STARTS CONVERSIONS
11 : WRITE STARTS CONVERSIONS

Analog to Digital Converter Read (LHLD 5000H)

!LLLLLLLL! : 5000H --- LOW BYTE OF CONVERSION
+-----+
!....HHHH! : 5001H --- HIGH 4 BITS
!-----!

Multiplexor Control (OUT 0E0H)

!TAAAAAAA!

!!!!!!!`----- A is the MAIN multiplexor address
!`----- described by Table 1-1
!
!----- T is the TRACK/HOLD signal (0 = TRACK)

Command Input Register (LHLD 0AFFH)

!LLLLLLLL! : AFFFH --- LOW BYTE OF THE COMMAND
+-----+
!HHHHHHHH! : B000H --- HIGH BYTE OF THE COMMAND
!-----!

Telemetry Output Register (SHLD 0AFFFH)

!LLLLLLLL! : AFFFH --- LOW BYTE OF THE TELEMETRY WORD
+-----+
!HHHHHHHH! : B000H --- HIGH BYTE OF THE TELEMETRY WORD

Deployment Unit Microswitches (IN 90H)

!22221111!

!!!!!!`----- BOOM 1 LEFT COVER (0 = COVER ON)
!!!!!!`----- BOOM 1 TURNS COUNTER
!!!!!!`----- BOOM 1 RIGHT COVER (0 = COVER ON)
!!!!!!`----- BOOM 1 END-OF-WIRE (0 = END-OF-WIRE)
!!!!!!`----- BOOM 2 LEFT COVER (0 = COVER ON)
!!!!!!`----- BOOM 2 TURNS COUNTER
!!!!!!`----- BOOM 2 RIGHT COVER (0 = COVER ON)
!!!!!!`----- BOOM 2 END-OF-WIRE (0 = END-OF-WIRE)

General Status Input (IN 80H)

!S...KKKK!

!`----- KELLEY GAIN BIT (LSB)
!`----- KELLEY GAIN BIT
!`----- KELLEY GAIN BIT
!`----- KELLEY GAIN BIT (MSB)
!`----- SUN PULSE INDICATOR

General Control Output (OUT C0H)

!KLBWMMMM!

!!!!!!`----- MOTOR 1 RELAY OFF COIL
!!!!!!`----- MOTOR 1 RELAY ON COIL
!!!!!!`----- MOTOR 2 RELAY OFF COIL
!!!!!!`----- MOTOR 2 RELAY ON COIL
!!!!!!`----- WATCHDOG RESET CIRCUIT (1=RESET)
!!!!!!`----- BURST COMPUTER RESET (0 = RESET, 1=RUN)
!!!!!!`----- LEPA REGISTER LOAD STATUS (1 = LOADING)
!!!!!!`----- KELLEY GAIN CHANGE STOBE (RISING EDGE)

3.5 The Burst Processor Board

The Burst computer system is a slave of the MAIN processor and is dedicated to the singular task of high frequency sampling and storage of field data. Equipped with its own multiplexor, analog to digital converter and a very large memory, this processor non-deterministically records data in commanded formats at frequencies of up to 60 KHz. The following paragraphs describe the subcircuits of the Burst system.

THE PROCESSOR. The Burst processor is the same computer used by the MAIN system, namely a Sandia SA3000, running at 5 MHz input frequency just like the MAIN system. And similar to that system, the low address pins are latched by an 8-bit latch (U19) on the high level of ALE (see the MAIN PROCESSOR description).

INTERRUPTS. The BURST system has two types of interrupts: 1) the BCMD line from the MAIN system and 2) the word rate clock interrupts (2 KHz) from the telemetry system. The BCMD line is the higher priority of the two and is used to signal the beginning of some communication from the MAIN processor. The 2 KHz interrupt is used to sample data synchronous with the telemetry system whenever frequencies of the word rate or less are requested.

ADDRESS DECODING. The BURST system devices are distinguished from each other by U25, a 3-to-8 line decoder the SA2995. The eight decoded outputs operate as 0, 1000H, 2000H, system operates on ANY address between 8000H and FFFFH so that it can be activated at the same time as any other device (see the

memory circuit below).

ROM/RAM. The Read Only Memory for the BURST computer is contained in a single 4096x8 chip, a Raytheon 29673SMB (U26). A power controller circuit similar to those on the MAIN system is used here (U24A, Q12 and Q13).

The Random Access Memory for the BURST system consists of 4 5114's (U27 thru U30) which together comprise a total of 2048 bytes of memory and are addressed as 1000H to 17FFH. The RAM is selected in a manner similar to that used in the MAIN system.

KELLEY GAIN CONTROL. The Kelley automatic gain control circuit has 4-bits of digital information which describe its current gain state. These bits are input to the BURST system by the 8-bit register U36 (four bits unused).

THE BURST MULTIPLEXOR. The BURST system multiplexor address is controlled by an 8-bit register (U37) which provides 4 bits to the multiplexor located elsewhere and 3 bits of local control (1 bit is not used). See the BURST MUX address table.

The way in which data is written to this latch dramatically differs from normal buss operations. Instead of responding to a write signal, this latch grabs bits from the low address buss when the A/D low byte is read. This facilitates high speed sampling which is described a little bit later.

ANALOG TO DIGITAL CONVERSION. The BURST A/D circuitry is quite a bit more complicated than the equivalent circuit in the

MAIN system, basically resulting from the very high sampling rate requirement. There are 3 major features added to this circuit as compared to the MAIN: 1) a high speed sample and hold is used, 2) automatic gain control is accomplished in hardware, and 3) most of this circuit can be powered-down.

At the heart of all this is, of course, the A/D itself --- an AM6112 (U21) which is the same device used in the MAIN system. On its digital side are two devices namely a buss transceiver (U20) and a 4050 (U31) which isolate the AM6112 from the processor buss when the A/D is turned off. This is a necessary step since the 6112 diode clamps its input pins when it has no power. The 6112 is clocked by BADCLK, a signal which comes from the MAIN system and is the Burst CPU clock out divided by two.

BURST AUTO-GAIN CIRCUIT. The analog signal to digitize comes into the board both in low and high gain forms, just as in the MAIN system. In the BURST system, however, a third line called the "GAIN DECISION" is provided. This is a digital signal which indicates whether high gain is in range of the A/D or not.

This gain decision is latched by the flip-flop U17B when clocked by the timing circuit (described later). The output of this latch determines which of the two input signals to choose for digitization by controlling the multiplexor U33.

The output of this high/low gain multiplexor is connected to a fast (5 microsecond) sample and hold, the Precision Monolithics SMP-11. The sample/hold timing is provided once again by the timing circuit described later. The value of the "hold"

capacitor, 5000 pF, was chosen for the fact that the SMP11 is internally trimmed for that value. Hence, no offsets will result in the output signal.

To indicate which gain state was used, the gain bit is jammed onto the processor buss when the high order bits of the 6112 are read. This is accomplished with U43B which is high when the A/D high byte is read. This gates the gain bit, driven by U39A, onto the processor buss by multiplexor U66, replacing the 4th most significant bit of the byte. This results in a 12-bit sample plus gain packed neatly in 13 contiguous bits.

Finally, the timing for all of this is accomplished using a single 4015 (shift register U38), configured as a unary counter. Reset by the reading of the A/D low byte, it is incremented with every A/D clock cycle. On the 6th thru 14th clocks, the CPU is stopped by the nand gate U2D. This keeps digital noise from the A/D converter while converting. On the 10th clock, this timer strobes the new gain decision into U17B (see above) and the old gain decision into U17A. Gain bits require this double buffering in order to track the pipeline correctly. On the 14th clock (A/D is done), the sample and hold is instructed to sample again.

A/D POWER DOWN. The BURST analog to digital conversion circuitry described above involves a number of high power devices, specifically, the 6112 A/D, the 4602 quad opamp, the SMP11 sample and hold, and the REF02 reference. Since during the long playback periods these parts will not be needed, the power to these devices has been routed through a pair of relays K7 and

K8. Relay K7 handles the +/- 5 Volts to the 6112 and the 4050 isolation buffer. Relay K8 switches OFF the +/- 12 Volts to the circuit.

THE MEMORY UNIT. The memory circuit for the BURST system is a rather autonomous unit which functions more like a digital tape recorder than a typical memory. Complete with its own memory address register (MAR) and bank wrap-around logic, this memory unit features total-dose radiation tolerance, protection against cosmic ray induced power surges, individual bank power-down, and a special "autowrite" mode which effectively doubles the memory transfer rate.

The unit acts like a tape recorder for two reasons. First, whenever reading or writing, the memory unit uses its own memory address register (U3, U6, U9, U11 and U13) rather than one supplied by the processor address buss. This allows the central processor to spend its time sampling rather than incrementing an address which is larger than the processor is designed for. The MAR keeps track of the address for the cpu and increments itself whenever either a read or a write to the memory is done.

Second, when the MAR reaches a specified end address, the next read or write causes the start address to be jammed into the MAR. This is accomplished by the upper address comparison (U7 and U8A) combined with the detection of an all 1's condition in the rest of the address (U8B, U15D and U2C). Thus, the "tape" automatically rewinds to the beginning and starts recording again.

Setting a particular address into the MAR is achieved by writing directly to the lower 12 address bits (the counters are on the processor buss as output ports). The upper six bits have to be written to the start address latch (U1) and then strobed into the counters U3 and U6 by toggling bit 6 (2nd MSB) of the start address port while doing one memory access. Note: the 40163 counters accept input data only when clocked while their load pins are down.

The decoder U16 is used to select which of the six memory banks the MAR is addressing. It is delected whenever the MAR is written into by virtue of a high level on pin 5 (2nd MSB of the start address port). This keeps accesses to the MAR from writing anything into the memory banks.

One peculiar feature of the memory unit is referred to as its "autowrite" mode. Initiated by a high level on the MSB of the start address port, this mode causes the memory unit to perform a WRITE operation when the processor buss says to READ. This seemingly useless configuration actually has a tremendous advantage in high speed data recording of data from another buss device like an a/d converter. The important thing to notice is that the selection of the memory unit can be overlapped with any other device on the processor buss. In the autowrite mode, if one addresses both the memory and another device and one reads from the device, the memory unit will catch whatever is read from that device! With an extremely fast a/d, this can effectively double the memory transfer capability of the system.

The swapping of the read and write lines is accomplished using a multiplexor and 4049 drivers (U14, U15A, U15B). Note that it is not a complete swap since writing to the memory in the autowrite mode still works --- the memory won't try to read.

The memory banks are protected from SCR latchups (caused by cosmic radiation) by a current foldback circuit on their +5V power. Consisting of Q10, Q11, and some discrete components, this circuit removes power from the entire bank of memory as soon as the SCR power surge is detected and long before the RAM device burns out. Measurements of this response show it to go into current limiting when the load exceeds 800 mW (20 Ohms at 4 Volts). Radiation tests with the RAM devices show them to take 4 times this amount of power in the latched condition. The entire memory bank requires only one tenth this amount of power when operating normally (pre-radiation).

A total dose radiation effect which is expected to occur with the type of memories used is a growth in power consumption. Radiation tests performed on these devices have shown a factor of 100 growth in power consumption before the devices actually fail to work. As a counter measure, the banks can be individually turned OFF in case one or all become excessively power hungry. If all banks are extremely hungry, the decision might be to turn all of them OFF and use the processor's 1K memory.

Bank power is controlled by relays K1 thru K6 in a Set/Reset scheme using K9. This is similar to the arrangement on the analog board. Basically, to power OFF a bank, e.g. bank 1, one

must apply current through the right coil or relay K1. To do this, a digital value is applied to the base resistor R1 which turns on Q1. This runs +5 to -7 Volts through the K1 reset coil. Similarly, to turn ON a bank, one simply flips Relay K9 before powering relay K1. The current then flows through the set coil instead of the reset coil.

BURST SAMPLE/HOLD/CONVERT PIPELINE. Unlike the MAIN computer system which is constrained to sample data only as fast as it can telemeter it, the BURST computer is designed for high speed data collection and playback. The BURST analog to digital conversion circuitry meets the high speed requirement by having its own multiplexor, a/d converter, AND a fast sample and hold (the SMP 11).

While the circuit design is capable of sampling rates in excess of 150 KHz, both the substitution of radiation tolerant devices and the slowing of the converter clock pulses has resulted in the current 60 KHz conversion frequency maximum. For example, the specifications for the AMD6112 converter say it should be capable of converting in 3 microseconds but the manufacturer will deliver only parts spec'd at 8 microseconds. The closest clock frequency which we have available is 1.25 MHz and this results in a convert time of 10.0 microseconds.

The procedure for sampling a qty in general is 1) address the quantity on the multiplexor, 2) set the sample-hold to SAMPLE for the required time, 3) set the sample-hold to HOLD, 4) start the

a/d converter and wait until the it's ready, 5) read out the converted data, and 6) store the result.

In the BURST processor, it takes a single instruction per sample to accomplish this series of events. In pipeline fashion, a new value can be addressed on the multiplexor and have its gain decision prepared while the prior value is being converted. To do this, the BURST software configures the a/d to start a new conversion as soon as the last converted value is read out (see the specifications on the AMD6112). BURST software also configures the MEMORY circuit to AUTOWRITE, which means it can take data directly off the buss when the a/d converter gives it to the cpu.

The process of converting a value is shown in figure AD-1. Any read of the a/d converter causes four things to happen. First, the sample/hold is placed in the HOLD condition. At the same time, the a/d converter puts its last result on the buss for the cpu and begins converting the quantity which the sample/hold is now holding. Lastly, the multiplexor register will take a new value described by some of the address bits used to read the a/d converter.

At the end of the 12.5th cycle after the a/d read, the conversion will complete. At the 14th cycle, an automatic timer will force the sample/hold back into the SAMPLE mode where it will track the value (we just set) on the multiplexor. This returns the circuit to its original state, ready to convert another quantity.

The net effect of the timing circuit is that the gain decision is latched after 8 microseconds from the start and the sample-hold is released to track the new value after 11.2 microseconds. If the sampling loop is 16.0 usec, then the hold time is 4.8 usec.

It is important to note that it actually takes three READS of the a/d converter to get the converted value into memory. After the first read, the sample/hold will acquire it. The second read will cause it to be converted but it will only be inside the converter. The third READ actually obtains the value and stores it.

INSTR.	CYC	MUX	S/H	A/D	MEMORY
LHLD Q1	0	--	HOLD --	CONVERT --	STORE --
	1	Q1	HOLD --	CONVERT --	
	14	Q1	SAMPLE --	FINISHED	
LHLD Q2	0	Q1	HOLD Q1	CONVERT Q1	STORE --
	1	Q2	HOLD Q1	CONVERT Q1	
	14	Q2	SAMPLE Q1	FINISHED	
LHLD Q3	0	Q2	HOLD Q2	CONVERT Q2	STORE Q1
	1	Q3	HOLD Q2	CONVERT Q2	
	14	Q3	SAMPLE Q2	FINISHED	

Figure 21. Burst A/D Pipeline Operation

SUMMARY OF BURST COMPUTER I/O. Described below are the input and output ports available to the Burst computer system:

Analog to Digital Converter Control (STA 3001H)

!.....MMB!
!-----!
!-----!
!-----!

!!`----- 1 = 2'S COMPLEMENT OUTPUT, 0 = OFFSET BINARY
!`----- MODE = 00 : WRITE STARTS CONVERIONS
!`----- 01 : READ STARTS CONVERSIONS
!`----- 10 : READ STARTS CONVERSIONS
!`----- 11 : WRITE STARTS CONVERSIONS

Analog to Digital Converter Read (LHLD 30xxH)

!LLLLLLLL! : 3000H --- LOW BYTE OF CONVERSION
+-----+
!...GHHHH! : 3001H --- HIGH 4 BITS PLUS GAIN BIT
!-----!

Multiplexor Control (LHLD 30xxH)

!AAAAAAA!
!-----!
!-----!

!!!!!!`----- A is the 7-bit BURST multiplexor address
!----- described by Table 1-2

High Memory Address Register (OUT 70H)

!ALMMMMMM!
!-----!
!-----!

!!!!!!`----- Upper Memory Address bit A12
!!`----- Upper Memory Address bit A17
!`----- Load upper MAR (1=load)
!----- Autowrite control (1=autowrite)

Low Memory Address Registers

!LLLLLLLL! : 0CFFFH
+-----+
!...HHHH! : 0D000H
!-----!

Relay Control Bits (OUT 20H)

!AABBBBBB!

!!!!!!`----- Bank 0 Relay
!!!!!!`----- Bank 1 Relay
!!!!!!`----- Bank 2 Relay
!!!!!!`----- Bank 3 Relay
!!!!!!`----- Bank 4 Relay
!!!!!!`----- Bank 5 Relay
!!!!!!`----- A/D +5V Relay
!!!!!!`----- A/D +12V Relay

General Control Bits (OUT 63H)

!..R..EEE!

! !!`----- End Memory address (A15)
! !`----- End Memory address (A16)
! `----- End Memory address (A17)
`----- Set/Reset Relay Control for above relays

3.6 The Memory Boards

The Burst memory itself (not the control circuitry) is located on two memory cards, each of which contains 3 banks of 32 KBytes each. Originally there had been 4 banks on each board, but we couldn't fit in that many chips. Hence, the total memory capacity of the Burst memory is 6 X 32 K or 192 KBytes.

As was mentioned earlier in the Burst CPU description, each memory bank can be individually powered. Both transient and total-dose radiation effects are expected to be encountered, particularly in view of the fact that the memory chips used are not "radiation tolerant" in the full definition of the term. While several rad-hard memories exist, they are either outrageously expensive or did not have sufficient density to be useful. These memories, made by Integrated Device Technologies, have survived more than 10000 rads which is the expected "box-level" total dose for the instrument. Assuming the shielding which the box provides, we shouldn't have too much radiation problems with them.

Nevertheless, each bank of memory is separated by a buss tranceiver and can be turned off completely by the Burst computer system. The trick to doing this is in isolating the memory address and control signals from the chips while powered down. (One must do this with most CMOS since the chips will take power from their input pins if their input pins have higher voltage than their power pins.)

Each memory bank's address and control lines are buffered through

CD4049 and CD4050 packages which are powered by the bank's power. When the bank power goes down, the 4049 and 4050 devices cannot drive the private bank buss. Unlike most CMOS devices, CD4049 and 4050 packages are made to accept input signals higher than their power pins.

Within a memory bank, one of the 16 memory chips is selected by a pair of 3-to-8 line decoders (SA2995). These are only enabled if the Bank Enable line is low (this signal comes from the Burst CPU board).

Six jumpers on each memory board allow for the address definition of the banks on that board. For example, the first bank may be configured as either bank 0 or 3, the second bank as either 1 or 4, and the third bank as either 2 or 5. For each bank, one must make two jumpers: 1) for power and 2) for the digital bank enable line. The standard way to configure the two boards is Banks 0, 1 and 2 together, and banks 3, 4 and 5 together.

3.7 The Power Converter

High level commands are pulsed commands which have 28 Volts on them (and considerable power capability). These are used in the instrument for switching on and off the power converter relays. Four commands (1 wire each) are used to control the MAIN and BACKUP power converters. The converters can both be on at the same time without damage to the converter.

1. MAIN CONVERTER POWER ON (CONNECTOR J8 PIN 18)
2. MAIN CONVERTER POWER OFF (CONNECTOR J8 PIN 17)
3. BACKUP CONVERTER POWER ON (CONNECTOR J8 PIN 5)
4. BACKUP CONVERTER POWER OFF (CONNECTOR J8 PIN 6)

The immediate effect of these four commands are as follows:

1. POWER ON/OFF -LP (CONNECTOR J8 PIN 4) IS SET TO BILEVEL 1
2. POWER ON/OFF -LP (CONNECTOR J8 PIN 4) IS SET TO BILEVEL 0
3. POWER ON/OFF -MAG (CONNECTOR J8 PIN 10) IS SET TO BILEVEL 1
4. POWER ON/OFF -MAG (CONNECTOR J8 PIN 10) IS SET TO BILEVEL 0

3.8 Inter-Processor Communications

The two computer systems communicate over two wires labelled BCMD and BRDY. BCMD is controlled by the MAIN processor and BRDY is controlled by the BURST. Both commands and data are transferred between the systems in 16-bit serial strings using BCMD and BRDY.

The protocol employed to send data from the MAIN to the BURST is the same as that used to send data the other way, so one need only describe one transmission to describe both. Initially both the BCMD and BRDY lines are low. From the MAIN cpu point of view, these names mean "BURST COMMAND" and "BURST READY". To send data from the MAIN processor to the BURST, the MAIN cpu sets BCMD to 1 which requests the BURST's attention. When the BURST is ready to accept the data, it sets BRDY to 1. The falling edge of BCMD is used as the starting time after which the MAIN cpu presents data bits on the BCMD line every 45 cycles. The BURST samples the BCMD line every 45 cycles and stores the bits in its software shift register. Finally, the MAIN processor restores the BCMD line low as does the BURST processor its status line BRDY.

These transmissions have a maximum operating frequency of approximately 56 Kbps with both cpu's using 5.0 MHz crystals. (Each transmission takes around 290 microseconds.)

Contention between the two processors for use of these lines is possible since both processors may want to send information to the other at the same time. For example, the BURST may be playing

data back to the MAIN processor at the same time as the MAIN gets a command from the ground with the BURST cpu as the destination. The situation in which both cpu's raise their request lines at the same time is handled by the BURST relenting and receiving data instead of sending it. (The BURST will immediately try again to send its data at the conclusion of the MAIN-to-BURST transmission unless explicitly stopped by command.)

Commands are distinguished from data only by context; i.e. after RESET to the Burst processor it enters the command state which expects to get command data. If the command has data following it, then the Burst will interpret the next transmissions as data. When the data portion is finished, it will again interpret incoming data as commands. This context sensitive protocol has one minor problem, namely that all command-data sequences from the MAIN cpu to the BURST must supply all of the data required. If the MAIN tries to stop the last command and start a new one before all of the data is sent, the BURST will confuse the new commands as if they were data. To stop a command-data sequence one must RESET the BURST processor.

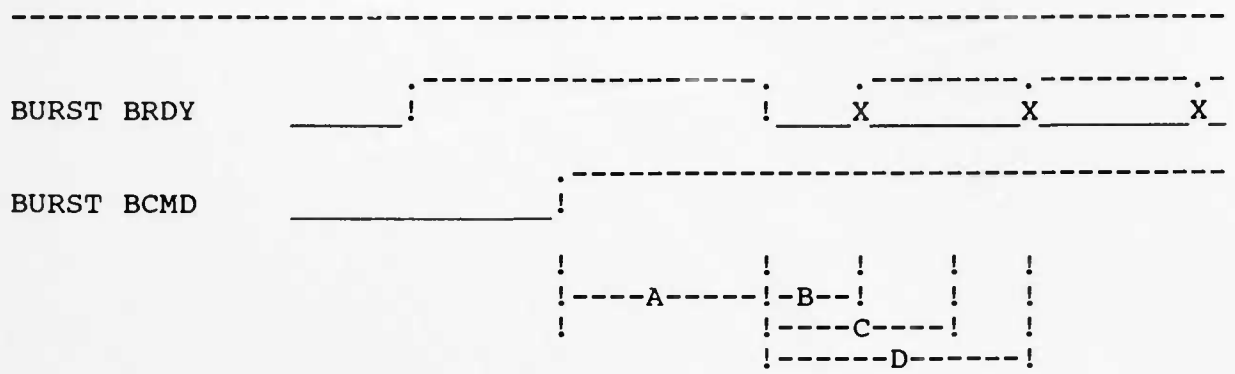


Figure 22. BURST to MAIN communication timing

	MIN	MAX
A : BURST RECOGNIZES MAIN READY	45 cyc	69
B : BURST FIRST BIT TRANSITION	31	31
C : TRANSITION TO TRANSITION	45	45
D : MAIN BIT SAMPLE FROM START	32	60

Table 11. Main Multiplexor Quantities

ADDR	0x LOW GAIN	1x HI GAIN	2x	3x	4x
x0	BZ	BZ	MTR1	V12X50	V34X50
x1	BY	BY	MTR2	V12X50	V34X50
x2	BX	BX	LEN1	V12X50	V34X50
x3	V3	V3	LEN2	V12X50	V34X50
x4	V2/RI2	V2/RI2	TMP1	V12X50	V34X50
x5	V1/SC	V1/SC	TMP2	V12X50	V34X50
x6	V1	V1	TMP3	V12X50	V34X50
x7	AGC	AGC**	TMP4	V12X50	V34X50
x8	V12/RI1	V12/RI1	MTR1	V12X50	V34X50
x9	F3	F3	MTR2	V12X50	V34X50
xA	F2	F2	LEN1	V12X50	V34X50
xB	F1	F1	LEN2	V12X50	V34X50
xC	V4	V4	TMP1	V12X50	V34X50
xD	AGC*	AGC*	TMP2	V12X50	V34X50
xE	V12/RI1*	V12/RI1*	TMP3	V12X50	V34X50
xF	V34	V34	TMP4	V12X50	V34X50

* : UNFILTERED QUANTITIES
 **: NON-SENSE QUANTITY

Table 12. Burst Multiplexor Quantities

ADDR	BQTY
x0	BZFAST
x1	BXFAST
x2	BYFAST
x3	V3
x4	V4
x5	V34
x6	V34 AC
x7	V1/SC
x8	V12 AC
x9	V2/RI2
xA	V1
xB	V12
xC	DIRECT
xD	AGC*
xE	GUARD 1
xF	STUB 1

* : UNFILTERED QUANTITIES

x : 0 -- AUTOMATIC GAIN 5 -- GROUND
 1 -- FORCED HIGH GAIN 6 -- BURST MEM+5V MONITOR
 2 -- FORCED LOW GAIN
 3,4,7 -- UNDEFINED

Table 13. Analog to Digital Conversion Times

	MAIN	BURST	UNITS
A/D CLOCK FREQUENCY :	1.25	1.25	MHz
A/D CONVERT TIME : (12.5 clock cycles)	10.0	10.0	microsec
SAMPLE TIME REQD :			
(10V step to .02%)	153(1)	4.8	microsec
(10V step to .10%)	---	3.5	
MINIMUM A/D TIME :	163.0	15.0	microsec
ACTUAL A/D TIME :	165	16.0	microsec

(1) The impedance of a Harris 508A (1.8K Ohms max) and a .01 microfarad capacitor have an RC of 18 microseconds. Twelve bit accuracy translates into 8.5 RC periods or 153 microseconds.

4. Contributing Engineers and Scientists

The CRRES Langmuir Probe and Fluxgate Magnetometer required the efforts of a number of engineers and scientists at the University of California, Regis College in Massachusetts, the Air Force Geophysics Lab in Massachusetts and Analytyx Corporation in New Hampshire. The following is a list of those people and their functions:

Name	Org	Functions
Peter R Harvey	UCB	Project Management at UCB Flight Hardware and Software GSE Hardware and Software
Dr. John Wygant	UCB	Project Scientist at UCB Flight Sphere Design Analog Section Design
Dr. Forrest Mozer	UCB	Co-Investigator for 701-14A
Dr. David Pankow	UCB	Boom Deployment Systems Engineering
Dr. Roy Torbert	UCSD	Flight Filtering section
Peter Anderson	Regis	Parts Engineering Test Management
Bill Sullivan	AFGL	Project Management at AFGL
Dr. Howard Singer	AFGL	Scientist for Fluxgate Magnetometer
Dr. Michael Smiddy	AFGL	Scientist for Langmuir Probe UCB Contract Monitor
Dr. Nelson Maynard	AFGL	Principle Investigator for 701-14A
Ken Fredholm	Analytyx	Project Management at Analytyx Systems Engineering
Bob Hayes	Analytyx	Power Supply Engineering
Paul Murray	Analytyx	Main Box Mechanical Engineering

Appendix A.

CRRES FLIGHT SOFTWARE

RELEASE 2.1, 12-6-88

MAIN SOFTWARE : 12-6-88
BURST SOFTWARE : 2-1-85

PETER R. HARVEY

```

0000      0001 *
0000      0002 * CRRES FLIGHT PROGRAM---CONFIGURATION
0000      0003 * WRITTEN BY PETER R HARVEY
0000      0004 * FILE CONFIG.A
0000      0005 *
0000      0006 * 8085 SPECIFIC INFORMATION
0000      0007 *
0000      0008 PSW EQU 6
0000      0009 SP EQU 6
0000      0010 *
0000      0011 * ROM CONFIGURATION OF PACKAGES
0000      0012 *
0000      0013 ORG 40H
0040      0014 CMDTAB DS 40H
0080      0015 NEXT DS 0C00H
0050      0016 MAG DS 148H
0E98      0017 PLA DS 170H
1008      0018 BUR DS 1E0H
11E8      0019 FIT DS 148H
1330      0020 SAW DS 90H
13C0      0021 ELE DS 2C0H
1680      0022 LD DS 44H
16C4      0023 DEP DS 120H
17E4      0024 SWP DS 380H
1B64      0025 EXEC DS 17CH
1CE0      0026 BKG DS 320H
2000      0027 *
2000      0028 * RAM CONFIGURATION
2000      0029 *
2000      0030 RAM EQU 2000H
2000      0031 RAMSIZE EQU 1000H
2000      0032 *
2000      0033 ORG RAM
2000      0034 IGRAM DS 1CH
201C      0035 BKGRAM DS 28H
2044      0036 DEPRAM DS 0CH
2050      0037 ELERAM DS 0A8H
20FB      0038 *
20FB      0039 ORG $/256+1*256
2100      0040 MAGRAM DS 50H MUST BE ALIGNED ON 256
2150      0041 PLARAM DS 20H
2170      0042 BURRAM DS 30H
21A0      0043 SAWRAM DS 10H
21B0      0044 FITRAM DS 170H
2320      0045 SPINRAM DS 170H
2490      0046 MATRAM DS 10H
24A0      0047 EXERAM DS 20H
24C0      0048 SWPRAM DS 460H
2920      0049 LDRAM EQU $
  
```


2920	0050	STACK	EQU	RAM+RAMSIZE-1
2920	0051	*		
2920	0052		COM	CMDTAB
2920	0053		COM	NEXT
2920	0054		COM	BKG
2920	0055		COM	ELE
2920	0056		COM	MAG
2920	0057		COM	PLA
2920	0058		COM	DEP
2920	0059		COM	LD
2920	0060		COM	EXEC
2920	0061		COM	BUR
2920	0062		COM	SAW
2920	0063		COM	SWP
2920	0064		COM	FIT
2920	0065	*		
2920	0066		COM	IORAM
2920	0067		COM	BKGRAM
2920	0068		COM	ELEGRAM
2920	0069		COM	MAGRAM
2920	0070		COM	PLARAM
2920	0071		COM	DEPRAM
2920	0072		COM	BURRAM
2920	0073		COM	SAWRAM
2920	0074		COM	SWPRAM
2920	0075		COM	FITRAM
2920	0076		COM	SPINRAM
2920	0077		COM	MATRAM
2920	0078		COM	LDRAM
2920	0079		COM	EXERAM

0000	0001 :		
0000	0002 :	CRRES FLIGHT SOFTWARE --- UTILITIES	
0000	0003 :	WRITTEN BY PETER R HARVEY	
0000	0004 :		
0000	0005	ORG	188
0008 36 00	0006 ZERO	MVI	M,0
000A 23	0007	INX	H
000B 0D	0008	DCR	C
000C C2 08 00	0009	JNZ	ZERO
000F	0010 :		
000F	0011	ORG	218-1
000F C8	0012 CPI	RZ	.
0010 1A	0013 COPY	LDAX	D
0011 77	0014	MOV	M,A
0012 13	0015	INX	D
0013 23	0016	INX	H
0014 0D	0017	DCR	C
0015 C3 0F 00	0018	JMP	CPI
0018	0019 :		
0018	0020	ORG	388
0018 85	0021 REF	ADD	L
0019 6F	0022	MOV	L,A
001A D0	0023	RNC	.
001B 24	0024	INR	H
001C B7	0025	ORA	A
001D C9	0026	RET	.
001E	0027 :		
001E	0028 UTIL	EQU	NEXT
001E	0029	ORG	UTIL
0080	0030	COM	ZERO
0080	0031	COM	COPY
0080	0032	COM	REF
0080	0033	COM	UNARY
0080	0034	COM	NEG16
0080	0035	COM	MARK
0080	0036 :		
0080	0037 :	CONVERT [HL] INTO 2**A-1	
0080	0038 :		
0080 21 01 00	0039 UNARY	LXI	H,1
0083 E6 0F	0040	ANI	15
0085 C8	0041 UNA1	RZ	
0086 29	0042	DAD	H
0087 3D	0043	DCR	A
0088 C3 85 00	0044	JMP	UNA1
0088	0045 :		
0088	0046 :	NEGATE [HL]	
0088	0047 :		
008B 7C	0048 NEG16	MOV	A,H
008C 2F	0049	CMA	

0088 57	0050	MOV	H,A
008E 7D	0051	MOV	A,L
008F 2F	0052	CMA	
0090 6F	0053	MOV	L,A
0091 27	0054	INX	H
0092 09	0055	RET	
0093	0056	*	
0093	0057	* OUTPUT (HL) TO THE DIAGNOSTIC LEDS	
0093	0058	*	
0093 7C	0059 MARK	MOV	A,H
0094 03 01	0060	OUT	:
0095 78	0061	MOV	A,L
0097 03 00	0062	OUT	0
0097 09	0063	RET	
009A	0064 NEXT	EQU	*
009A		CON	NEXT

```

0000      0001 *
0000      0002 * CRRES FLIGHT SOFTWARE---INPUT/OUTPUT DRIVER SECTION
0000      0003 * WRITTEN BY PETER R HARVEY
0000      0004 *
0000      0005 * FILE IO.4
0000      0006 *
0000      0007 * IO SYSTEM DESCRIPTION
0000      0008 *
0000      0009 SERIAL EQU 0F3H SERIAL CONTROL OUTPUT
0000      0010 SERCLK EQU 2 SERIAL CLOCK
0000      0011 ANASTB EQU 4 ANALOG SHIFT REGISTER STROBE
0000      0012 FILSTB EQU 8 FILTER SHIFT REGISTER STROBE
0000      0013 NWF EQU 16 7528 WRITE (INVERTED)
0000      0014 AB EQU 32 7528 A/B CONTROL
0000      0015 INT EQU 128 INTERRUPT CIRCUIT
0000      0016 *
0000      0017 ADC EQU 5000H A/D CONVERTER
0000      0018 ADCTL EQU ADC+1 A/D CONTROL
0000      0019 *
0000      0020 MUX EQU 0E0H MUX, TRACK/HOLD, AND POWER
0000      0021 ADDR EQU 07FH MUX ADDRESS BITS
0000      0022 HIGAIN EQU 10H HIGH GAIN SELECT
0000      0023 TRACK EQU 080H TRACK BIT
0000      0024 PRVAL EQU 079H PREFERRED RESET VALUE FOR MUX
0000      0025 AGCF EQU 7 KELLEY QUANTITIES ON TM MUX
0000      0026 AGCU EQU 13
0000      0027 V12H EQU 2EH "V12 X50" QTY, V34H = V12H+1
0000      0028 V12 EQU 0EH V12 LOW GAIN USED IF X50 SAT.
0000      0029 V12X50 EQU 30H
0000      0030 V34 EQU 0FH V34 LOW USED IF X50 SAT.
0000      0031 V34X50 EQU 40H
0000      0032 *
0000      0033 COMMAND EQU 0AFFFH COMMAND REGISTER (MEMORY MAPPED)
0000      0034 TELEM EQU 0AFFFH TELEMETRY BUFFER OUTPUT
0000      0035 *
0000      0036 MICROS EQU 90H BOOM MICROSWITCHES
0000      0037 MOTOPS EQU 0C0H MOTOR CONTROL
0000      0038 KLYGC EQU 080H KELLEY GAIN CONTROL
0000      0039 MAGLOAD EQU 040H MAG LOAD STATUS
0000      0040 BRUN EQU 20H BURST PROCESSOR RESET/RUN
0000      0041 WDRES EQU 10H WATCHDOG RESET (1=RESET)
0000      0042 *
0000      0043 KLYIN EQU 080H KELLEY GAIN BITS
0000      0044 KLYGN EQU 0FH
0000      0045 SUNBIT EQU 90H SUN PULSE FLOP STATUS
0000      0046 FLIGHT EQU 10H PROTOFLIGHT IF 1
0000      0047 *
0000      0048 LEPAH EQU 0D0H LEPA SHIFT REGISTER OUTPUT
0000      0049 LEPAL EQU 090H
  
```

```

0000
0000 0050 #
0000 0051 PSW EQU 6 9035 INFORMATION
0000 0052 SP EQU 6
0000 0053 PIM EQU 20H
0000 0054 SIM EQU 30H
0000 0055 SLDE EQU 18H
0000 0056 SDG EQU 80H
0000 0057 SODEN EQU 40H
0000 0058 RES75 EQU 10H
0000 0059 MSE EQU 9
0000 0060 DIS75 EQU 4
0000 0061 DIS65 EQU 2
0000 0062 DIS55 EQU 1
0000 0063 #
0000 0064 # RESTART 7 (CONVERSION DONE INTERRUPT)
0000 0065 #
0000 0066 ORG 743
0000 0067 FET
0000 0068 #
0000 0069 # IO CONTROL ENTRY POINTS
0000 0070 #
0000 0071 IO EQU NEXT
0000 0072 ORG IO
0000 0073 COM IOINIT
0000 0074 COM GETMASK
0000 0075 COM SETMASK
0000 0076 COM CMDIN
0000 0077 COM TMOU
0000 0078 COM SUNSTAT
0000 0079 COM BOOMSTAT
0000 0080 #
0000 0081 COM SAMPLE
0000 0082 COM AGC
0000 0083 COM SETBIAS
0000 0084 COM SETGUARDS
0000 0085 COM SETSTUBS
0000 0086 COM SETVTRIM
0000 0087 COM SETRELAY
0000 0088 COM SETFILTER
0000 0089 COM SETMUX
0000 0090 #
0000 0091 COM SETMOTOR
0000 0092 COM SETKLY
0000 0093 COM SETPLA
0000 0094 #
0000 0095 COM SEND
0000 0096 COM RECEIVE
0000 0097 COM TSTMUX
0000 0098 COM RWATCH
0000 0099 COM TODSC
  
```

```

009A          0100      COM   RBURST
009A          0101 *
009A          0102 * INITIALIZE THE INPUT/OUTPUT SO THE MODULE
009A          0103 * FUNCTIONS CORRECTLY.
009A          0104 *
009A 3E 20    0105 IOINIT MVI   A,BRUN  CLEAR MOTORS
009C 32 00 20 0106      STA   MKBCOPY  KELLEY,RUN BURST
009F CD B5 03 0107      CALL  RWATCH  RESET THE WATCHDOG
00A2 3E 40    0108      MVI   A,SDOEN  RESET BCMD LINE
00A4 30      0109      DB     SIM
00A5 00      0110      NOP
00A6 00      0111      NOP
00A7 00      0112      NOP
00A8          0113 *
00A9 97      0114      SUB   A     RESET MOTOR 0
00A9 CD 18 03 0115      CALL  SETMOTOR
00AC 3E 01    0116      MVI   A,1    AND MOTOR 1
00AE CD 18 03 0117      CALL  SETMOTOR
00B1          0118 *
00B1 3E 3F    0119      MVI   A,3FH  RESET THE SERIAL CTL
00B3 D3 F3    0120      OUT   SERIAL
00B5 32 14 20 0121      STA   MUXCOPY  CLEAR RESET TO KELLEY
00B9 3E 01    0122      MVI   A,1    SET A/D FOR 2'S COMPLEMENT
00BA 32 01 50 0123      STA   ADCTL
00BD          0124 *
00BD CD FA 02 0125      CALL  FILDCLEAR
00C0 97      0126      SUB   A     CLEAR THE RELAY BITS
00C1 32 15 20 0127      STA   RS
00C4 32 16 30 0128      STA   RS+1
00C7 32 17 20 0129      STA   RS+2
00CA C3 08 02 0130      JMP   PLYOFF
00CD          0131 *
00CD          0132 * SYSTEM INTERRUPT STATUS CALLS.
00CD          0133 *
00CD 20      0134 GETMASK DB   RIM   READ THE INTERRUPT MASK
00CE 7       0135      RET
00CF 30      0136 SETMASK DB   SIM   SET THE INTERRUPT MASK
00D0 C9      0137      RET
00D1          0138 *
00D1          0139 * COMMAND INPUT.
00D1          0140 * ON EXIT: [HL]= COMMAND
00D1          0141 *
00D1 2A FF AF 0142 CMDIN LHLD  COMMAND  PICK THE COMMAND REGISTER
00D4 C9      0143      RET
00D5          0144 *
00D5          0145 * TELEMETRY OUTPUT.
00D5          0146 * ON ENTRY: [HL]=TELEMETRY OUTPUT
00D5          0147 *
00D5 22 FF AF 0148 TMOUT SHLD  TELEM  SET THE SHIFT REGISTER
00D8 C9      0149      RET

```

```

00D9          0150 *
00E9          0151 * SUN AND BOOM STATUS CALLS
00D9          0152 *
00D9 DB 80    0153 SUNSTAT IN  SUNBIT  THE SUN STATUS IS
00E8 E6 80    0154      ANI  BOH   IN THE MSB
00D0 21 00 20 0155      LXI  H,SUNCOPY IF SAME AS WE'VE SEEN
00E0 BE       0156      CMP  M    THEN JUST RETURN ZERO
00E1 77       0157      MOV  M,A   ELSE RECORD NEW STATUS
00E2 09       0158      RET
00E3         0159 *
00E3 DB 90    0160 BOOMSTAT IN  MICROS  READ THE BOOM
00E5 09       0161      RET   .    MICROSWITCHES.
00E6         0162 *
00E6         0163 * SAMPLE THE A/D CONVERTER
00E6         0164 * ON ENTRY: [A]=QTY TO ADDRESS ON MULTIPLEXOR
00E6         0165 * ON EXIT : [HL]=THE 12-BIT VALUE
00E6         0166 *
00E6 6F       0167 SAMPLE MOV  L,A   SAVE MUX #
00E7 3E 0D    0168      MVI  A,MSE+DIS75+DIS55 TURN OFF INTS (NOT SYNC)
00E9 30       0169      DE   SIM
00EA         0170 *
00EA 7D       0171      MOV  A,L
00E9 F3 0D    0172      ORI  TRACK START SAMPLING
00E0 D3 E0    0173      OUT  MUX
00EF         0174 *
00EF 3E 19    0175      MVI  A,153*5/2-21/14 DELAY 153 MICROSECS
00F1 3E       0176 SDLA  DCR  A    (14 CYCLE LOOP)
00F2 02 F1 00 0177      JNZ  SDLA  /21 OTHER CYCLES)
00F5 7D       0178      MOV  A,L  REMOVE TRACK
00F6 D3 E0    0179      OUT  MUX
00F8         0180 *
00F8 3E 90    0181      MVI  A,NWR+INT REQUEST INTERRUPT
00FA D3 F3    0182      OUT  SERIAL
00FC 3E FF    0183      MVI  A,OFFH PUT OUT RESTART 7 ON BUSS
00FE D3 50    0184      OUT  ADD/256 WRITE 10 START CONVERSION
0100 76       0185      HLT
0101 3E 10    0186      MVI  A,NWF
0103 D3 F3    0187      OUT  SERIAL
0105 3E 08    0188      MVI  A,MSE ENABLE WORD CLOCKS
0107 30       0189      DE   SIM
0108         0190 *
0108 7D       0191      MOV  A,L  GET MUX ADDRESS
0109 24 00 50 0192      LHLD  ADC  GET THE 12 BITS
010C FE 07    0193      CPI  AGCF IF KELLEY, GO TO IT
010E CA 16 01 0194      JZ   GETFLY
0111 FE 0D    0195      CPI  AGCU
0113 02 20 01 0196      JNZ  MASK
0116         0197 *
0116 7D       0198 GETKLY MOV  A,L  MASK THE LOW BITS
0117 E6 F0    0199      ANI  -1-KLYGN

```

```
0119 6F      0200      MOV    L,A
011A 08 80    0201      IN     KLYIN
011C E6 0F    0202      ANI   KLYGN
011E B5      0203      ORA   L
011F 6F      0204      MOV    L,A
0120          0205 *
0120 7C      0206 MASK MOV    A,H    MASK UPPERS
0121 E6 0F    0207      ANI   OFH
0123 67      0208      MOV    H,A
0124 3E F9    0209      MVI   A,TRACK+PRVAL RETURN TO ZERO
0126 D3 E0    0210      OUT   MUX
0128 FB      0211      EI
0129 C9      0212      RET   .
012A          0213 *
012A          0214 * AUTOMATIC GAIN CONTROL LOGIC
012A          0215 * ON ENTRY: A=LOW GAIN QTY ADDRESS
012A          0216 * ON EXIT:  [D]=PROPER QTY TO DIGITIZE
012A          0217 * ZERO FLAG SET IF LOW GAIN, NZ FOR H1
012A          0218 *
012A 57      0219 ABC  MOV    D,A    ASSUME LOW GAIN
0128 FE 07    0220      CPI   AGCF    IF EITHER KELLEY QTY, TAKE A
0120 CA 5E 01 0221      JZ    AGC2    DUMMY SAMPLE.
0130 FE 0D    0222      CPI   AGCU
0132 CA 5E 01 0223      JZ    AGC2
0135 FE 10    0224      CPI   HIGAIN IF AUTOGAIN QTY, GO
0137 DA 4C 01 0225      JC    AUTO
013A FE 2E    0226      CPI   V12H  IF FORCED LOW GAIN AND NOT X50
013C 0A 58 01 0227      JC    FLG    THEN GO
013F 11 30 0E 0228      LXI   D,V12*256+V12X50 ELSE DO THE X50
0142 CA 48 01 0229      JZ    X5060
0145 11 40 0F 0230      LXI   D,V34*256+V34X50
0148 7B      0231 X5060 MOV    A,E
0149 C3 4F 01 0232      JMP   AGC1
014C          0233 *
014C F6 10    0234 AUTO ORI   HIGAIN SAMPLE THE H1 GAIN
014E 5F      0235      MOV    E,A
014F C0 E6 00 0236 ABC1 CALL  SAMPLE [HL]=SAMPLE
0152 7C      0237      MOV    A,H    IF SAMPLE=7XX THEN IT'S
0153 FE 07    0238      CPI   7      VERY HIGH POSITIVE.
0155 C8      0239      RZ    .      SO USE LOW GAIN
0156 FE 08    0240      CPI   8      IF SAMPLE=8XX THEN IT'S
0158 C8      0241      RZ    .      VERY NEGATIVE. USE LOW
0159 53      0242      MOV    D,E    ELSE HIGH IS OK
015A C9      0243      RET   .
015B          0244 *
015B E6 EF    0245 FLG  ANI   -1-HIGAIN REMOVE HIGAIN FROM QTY
015D 57      0246      MOV    D,A
015E C0 E6 00 0247 ABC2 CALL  SAMPLE AND TAKE DUMMY SAMPLE
0161 97      0248      SUB   A      AND RETURN(O) FOR LOW GAIN
0162 C9      0249      RET
```


0163	0250 *		
0163	0251 *	BIAS, STUB AND GUARD CONTROL LOGIC.	
0163	0252 *	ON ENTRY: [L]=8-BIT DAC VALUE	
0163	0253 *	[H]=BOOM NUMBER 1-4	
0163	0254 *		
0163	0255	SPHAD EQU 0EH	BIT 1 SELECT
0163	0256	CYLAD EQU 07H	BIT 4 SELECT
0163	0257	GRDAD EQU 0BH	BIT 2 SELECT
0163	0258	STBAD EQU 0DH	BIT 3 SELECT
0163	0259	NILAD EQU 0FH	ALL OFF
0163	0260	RLYDBL EQU 01GH	RELAY DISABLE
0163	0261 *		
0163 1E 05	0262	SETBIAS MVI	E,SBIAS#256/256 RECORD IT
0165 3E 03	0263	MVI	A,3 ONLY 4 BIASES MAX
0167 CD AF 01	0264	CALL	RECORD RECORD
016A FE 02	0265	CPI	2 IF SPHERE BOOM, USE
016C 2E 0E	0266	MVI	L,SPHAD THE 1ST SELECT BIT
016E DA 87 01	0267	JC	ANADAC SHIFT INTO ANALOG BOARD
0171 2E 07	0268	MVI	L,CYLAD CYLINDERS ARE ANOTHER BIT
0173 C3 87 01	0269	JMP	ANADAC
0176	0270 *		
0176 1E 0B	0271	SETGUARD MVI	E,SGUARD#256/256
0178 CD AD 01	0272	CALL	RECD1
017B 2E 0B	0273	MVI	L,GRDAD GUARDS ARE AT
017D C3 87 01	0274	JMP	ANADAC 2ND SELECT BIT
0180	0275 *		
0180 1E 09	0276	SETSTUB MVI	E,SSTUB#256/256
0182 CD AD 01	0277	CALL	RECD1
0185 2E 0D	0278	MVI	L,STBAD STUBS ARE 3RD.
0187 F5	0279	ANADAC PUSH	PSW
0188 CD 3D 02	0280	CALL	NLOFF GET SELECT FOR RELAYS OFF
018B E6 F0	0281	ANI	0F0H
018D E5	0282	ORA	L PUT IN SELECT BIT
018E 6F	0283	MOV	L,A
018F 7C	0284	MOV	A,H CONVERT TO 0 TO FF
0190 EE 7F	0285	XRI	7FH FROM +/- 127
0192 67	0286	MOV	H,A
0193 F1	0287	POP	PSW
0194 CD 81 02	0288	CALL	SHF16 SHIFT HL TO I/O REGS
0197 2E 14	0289	MVI	L,NWR+ANASTB STROBE ANALOG 4094'S
0199 0F	0290	STROBE RRC	. SELECT EITHER A OR B SIDE
019A 7D	0291	MOV	A,L
019B D2 A0 01	0292	JNC	AS1
019E F6 20	0293	ORI	AB (B SIDE)
01A0 F3	0294	AS1 DI	DON'T LET SAMPLING INTERRUPT
01A1 D3 F3	0295	OUT	SERIAL
01A3 E6 EF	0296	LATCH ANI	-NWR-1 REMOVE WRITE BAR
01A5 D3 F3	0297	OUT	SERIAL
01A7 F6 10	0298	ORI	NWR AND REPLACE IT
01A9 D3 F3	0299	OUT	SERIAL

01AB F9	0300	EI		
01AC C9	0301	RET	.	
01AD	0302	*		
01AD 3E 01	0303	RECD1	MV1	A,1 MASK TO 0 OF 1
01AF 25	0304	RECORD	DCR	H DECREMENT THE BOOM TO 0-N
01B0 A4	0305	ANA	H	MASK(A) BITS
01B1 F5	0306	PUSH	PSW	
01B2 83	0307	ADD	E	ADD OFFSET TO E
01B3 5F	0308	MOV	E,A	
01B4 16 20	0309	MV1	D,10STAT/256	
01B6 7D	0310	MOV	A,L	STORE THE VALUE
01B7 12	0311	STAX	D	
01B8 F1	0312	POP	PSW	
01B9 65	0313	MOV	H,L	PREPARE TO SHIFT
01BA C9	0314	RET		
01BB	0315	*		
01BE	0316	*	RELAY CONTROL LOGIC.	
01BB	0317	*	ON ENTRY: A=RELAY NUMBER 0 TO 21	
01BB	0318	*	CRY=1 TO SET, 0 TO RESET	
01BB	0319	*		
01BB F5	0320	SETRELAY	PUSH	PSW
01BC CD 4E 02	0321	CALL	RECRLY	RECORD RELAY
01BF F1	0322	POP	PSW	
01C0 17	0323	RAL		
01C1 FE 04	0324	CP1	282	IF K0 OR K1, DIRECTIONAL
01C3 DA 02 02	0325	JC	RLYON	
01C6 FE 0E	0326	CP1	247	IF K2-K6, NONLATCHING TYPE
01C8 DA FD 01	0327	JC	NONLAT	
01CB FE 24	0328	CP1	248	IF K7-K17, LATCHING
01CD DA EF 01	0329	JC	LAT	
01D0 2E 01	0330	MV1	L,1	IF K18,K19 THEN EXTERNAL(K1)
01D2 FE 2B	0331	CP1	2420	
01D4 DA DB 01	0332	JC	EXT	
01D7 2D	0333	DCR	L	ELSE K20-K21, THEN EXTERNAL(K0)
01DB	0334	*		
01D8 F5	0335	EXT	PUSH	PSW SAVE THE REAL RELAY#
01D9 1F	0336	RAR	.	SETRELAY(KL,ONOFF)
01DA 7D	0337	MOV	A,L	
01DB CD BB 01	0338	CALL	SETRELAY	
01DE F1	0339	POP	PSW	
01DF	0340	*		
01DF D6 20	0341	SUI	16#2	SETRELAY(K-16,ON)
01E1 1F	0342	RAR	.	
01E2 37	0343	STC	.	
01E3 F5	0344	PUSH	PSW	
01E4 CD BB 01	0345	CALL	SETRELAY	
01E7 CD A7 03	0346	CALL	DLA5	DELAY EXTRA
01EA F1	0347	POP	PSW	
01EB B7	0348	DRA	A	SETRELAY(K-16,OFF)
01EC C3 BB 01	0349	JMP	SETRELAY	

01EF	0350 #			
01EF F5	0351 LAT	PUSH	PSW	SETRELAY(K1,ONOFF)
01F0 1F	0352	RAR		
01F1 3E 01	0353	MVI	A,1	
01F3 CD BB 01	0354	CALL	SETRELAY	
01F6 F1	0355	POP	PSW	
01F7 CD 02 02	0356	CALL	RLYON	PULSE(K) ON/OFF
01FA C3 08 02	0357	JMP	RLYOFF	
01FD	0358 #			
01FD 0F	0359 NONLAT	RRC	.	FOR NONLATCHING, SIMPLY
01FE 07	0360	RLC	.	TEST WHETHER TO TURN THE CURRENT
01FF D2 08 02	0361	JNC	RLYOFF	ON OR OFF
0202	0362 #			
0202 CD 19 02	0363 RLYON	CALL	SELECT	A=SELECT BITS FOR POWER
0205 C3 08 02	0364	JMP	SRLY	
0208 C9 3D 02	0365 RLYOFF	CALL	NLOFF	A=TURN OFF BITS
020B 67	0366 SRLY	MOV	H,A	PUT DECODER BITS IN MSBYTE
020C F5 0F	0367	ORI	NILAD	DESELECT DACS FOR LOW POWER
020E 6F	0368	MOV	L,A	
020F CD 81 02	0369	CALL	SHF16	
0212 3E 14	0370	MVI	A,NWR+ANASTE	
0214 D3 F3	0371	OUT	SERIAL	
0216 C3 A7 03	0372	JMP	DLAS	
0219	0373 #			
0219	0374 #	RELAY SELECT ROUTINE		
0219	0375 #	ON ENTRY: [A]=RELAY # TIMES 2		
0219	0376 #	ON EXIT: [A]=DECODER SELECT BITS PLUS K0,K1		
0219	0377 #			
0219 B7	0378	SELECT	DRA	A A=RELAY NUMBER AGAIN
021A 1F	0379	RAR	.	
021B B6 02	0380	SUI	2	IF K0 OR K1, USE OLD SELECT BITS
021D DA 29 02	0381	JC	SEL01	
0220 E6 0F	0382	ANI	0FH	ELSE CONVERT A INTO DECODER #
0222 C6 18	0383	ADI	18H	
0224 E6 27	0384	ANI	27H	MASK AND LOWER RLYDEL
0226 C3 2C 02	0385	JMP	SETSEL	
0229 3A 01 20	0386 SEL01	LDA	OLDSEL	GET PRIOR SELECTION
022C	0387 #			
022C E6 3F	0388	SETSEL	ANI	03FH MASK THE SELECTION BITS
022E 32 01 20	0389	STA	OLDSEL	
0231 5F	0390	MOV	E,A	
0232 3A 15 20	0391	LDA	RS	PUT IN K0 AND K1
0235 E6 03	0392	ANI	3	
0237 0F	0393	RRC		
0238 0F	0394	RRC		
0239 EE C0	0395	XRI	0C0H	INVERT FOR PNP TRANSISTORS
023B B3	0396	ORA	E	
023C C9	0397	RET		
023D	0398 #			
023D	0399 #	TURN OFF POWER TO RELAYS FROM K2 THRU K21		

023D	0400 *
023D CD 45 02	0401 NLOFF CALL CLRNL CLEAR NONLATCHING STATUS
0240 3E 10	0402 MVI A,RLYD8L DISABLE RELAYS
0242 C3 2C 02	0403 JMP SETSEL
0245	0404 *
0245 3A 15 20	0405 CLRNL LDA RS SHOW NON-LATCHING OFF
024B E6 83	0406 ANI 083H
024A 32 15 20	0407 STA RS
024D C9	0408 RET
024E	0409 *
024E	0410 * RECORD RELAY ON/OFF
024E	0411 *
024E F5	0412 RECRLY PUSH PSW
024F FE 02	0413 CPI 2 ANY RELAYS BUT K0/K1
0251 D4 45 02	0414 CNC CLRNL CAUSE NONLATCHING TO RESET
0254 F1	0415 POP PSW
0255 11 15 20	0416 LXI D,RS
0258	0417 *
0258	0418 * SBIT: [DE]-> ARRAY, [A]=#, CARRY=BIT
0258	0419 *
0258 F5	0420 SBIT PUSH PSW SAVE CARRY
0259 FE 0B	0421 SBI CPI 8 SET HL->BYTE
025B DA 64 02	0422 JC SB2
025E 13	0423 INX D
025F D6 0B	0424 SUI 8
0261 C3 59 02	0425 JMP SBI
0264 CD 80 00	0426 SB2 CALL UNARY L=BIT MASK
0267 EB	0427 XCHG .
026B F1	0428 POP PSW GET CARRY
0269 7B	0429 MOV A,E
026A DA 71 02	0430 JC SBSET
026D 2F	0431 CMA . CLEARING A BIT
026E A6	0432 ANA M
026F 77	0433 MOV M,A
0270 C9	0434 RET .
0271 B6	0435 SBSET ORA M SETTING A BIT
0272 77	0436 MOV M,A
0273 C9	0437 RET
0274	0438 *
0274 21 03 20	0439 IODSC LXI H,I0STAT REFERENCE IO DIGITAL STAT
0277 85	0440 ADD L
0278 6F	0441 MOV L,A
0279 7E	0442 MOV A,M
027A C9	0443 RET
027B	0444 *
027B	0445 * SEND BITS TO THE ANALOG I/O REGISTER
027B	0446 * ON ENTRY: [HL]=DATA TO SHIFT
027B	0447 *
027B C5	0448 SHFB PUSH B
027C 0E 0B	0449 MVI C,B COUNT OF BITS

027E C3 B4 02	0450	JMP	ASHF	
0281 C5	0451 SHF16	PUSH	B	
0282 0E 10	0452	MVI	C,16	
0284	0453	*		
0284 F5	0454 ASHF	PUSH	PSW	SAVE ACCUM FOR RETURN
0285 29	0455 ASH1	DAD	H	SHIFT ONE BIT INTO CARRY
0286 3E 08	0456	MVI	A,NWR/2	THEN INTO THE LSB OF A
0288 17	0457	RAL	.	
0289 F3	0458	DI	.	DON'T ALLOW SAMPLING TO INTERRUPT
028A D3 F3	0459	OUT	SERIAL	PUT DATA OUT WITH CLOCK=0
028C F6 02	0460	ORI	SERCLK	THEN WITH RISING EDGE
028E D3 F3	0461	OUT	SERIAL	
0290 FB	0462	EI		
0291 0D	0463	DCR	C	UNTIL --COUNT=0
0292 C2 B5 02	0464	JNZ	ASH1	
0295 F1	0465	POP	PSW	
0296 C1	0466	POP	B	
0297 C9	0467	RET		
0298	0468	*		
0298	0469	*	VTRIM CONTROL LOGIC.	
0298	0470	*	ON ENTRY: H=1 FOR VTRIM12, 2 FOR VTRIM34	
0298	0471	*	[L]=DAC VALUE (8-BITS)	
0298	0472	*		
0298 1E 03	0473	SETVTRIM	MVI E,SVTRIM#256/256	
029A CD AD 01	0474	CALL	RECD1	
029D F5	0475	PUSH	PSW	SAVE WHICH VTRIM
029E EB	0476	XCHG	.	
029F 7A	0477	MOV	A,D	CONVERT TO 0 TO FF
02A0 EE 7F	0478	XRI	7FH	FROM +/- 127
02A2 57	0479	MOV	D,A	
02A3 2E 01	0480	MVI	L,1	SELECT VTRIM PAIR DAC
02A5 CD 00 03	0481	CALL	FSEND	
02A8 F1	0482	POP	PSW	NOW IF VTRIM12,USE SIDE A
02A9 2E 18	0483	MVI	L,NWR+FILSTB	
02AB C3 99 01	0484	JMP	STROBE	
02AE	0485	*		
02AE	0486	*	FILTER CONTROL LOGIC.	
02AE	0487	*	ON ENTRY: H=FILTER NUMBER 1-7	
02AE	0488	*	[L]=FILTER VALUE	
02AE	0489	*		
02AE 1E 0D	0490	SETFILTER	MVI E,SFILTER#256/256	
02B0 3E 07	0491	MVI	A,7	SET MASK
02B2 CD AF 01	0492	CALL	RECORD	
02B5 54	0493	MOV	D,H	PUT DATA IN D REG
02B6 3C	0494	INR	A	TURN ON BIT 1-7
02B7 E6 07	0495	ANI	7	
02B9 CD 80 00	0496	CALL	UNARY	L=BIT SELECT
02BC CD 00 03	0497	CALL	FSEND	
02BF 3E 18	0498	MVI	A,NWR+FILSTB	CHOOSE A SIDE
02C1 CD A0 01	0499	CALL	ASI	

02C4 3E 38	0500	MVI	A,NWR+FILSTR+AR	THEN B SIDE
02C6 CD A0 01	0501	CALL	AS1	
02C9 C3 FA 02	0502	JMP	FILCLEAR	
02CC	0503	*		
02CC	0504	*	SET MULTIPLEXOR BITS WHICH STEER THE FILTERING	
02CC	0505	*	ON ENTRY: A=MULTIPLEXOR NUMBER ON FILTER BOARD	
02CC	0506	*	L=VALUE TO ADDRESS ON THAT MUX	
02CC	0507	*		
02CC C5	0508	SETMUX	PUSH B	CALL THE BIT REPLACEMENT
02CD 5D	0509	MOV	E,L	ROUTINE TO REPLACE ONLY
02DE 21 F2 02	0510	LXI	H,MUXFIELD	THE FIELD FOR THAT REG
02D1 E6 07	0511	ANI	7	MASK THE REGISTER NUMBER
02D3 4F	0512	MOV	C,A	
02D4 06 00	0513	MVI	B,0	
02D6 09	0514	DAD	B	
02D7	0515	*		
02D7 3A 14 20	0516	LDA	MUXCOPY	GET THE MUX REGISTER
02DA CD C1 03	0517	CALL	REPFIELD	REPLACE THE ONE FIELD
02DD 32 14 20	0518	STA	MUXCOPY	AND THEN SEND IT OUT
02E0 C1	0519	POP	B	
02E1 C3 FA 02	0520	JMP	FILCLEAR	
02E4	0521	*		
02E4	0522	*	TEST MULTIPLEXOR BITS	
02E4	0523	*	ON ENTRY: (A)=FIELD NUMBER	
02E4	0524	*	ON EXIT: ZERO FLAG IF FIELD=0	
02E4	0525	*		
02E4 E6 07	0526	TSTMUX	ANI 7	(HL)->MUXFIELD(A)
02E6 5F	0527	MOV	E,A	
02E7 16 00	0528	MVI	D,0	
02E9 21 F2 02	0529	LXI	H,MUXFIELD	
02EC 19	0530	DAD	D	
02ED 3A 14 20	0531	LDA	MUXCOPY	PICK UP ALL BITS
02F0 A6	0532	ANA	M	MASK TO SET STATUS
02F1 C9	0533	RET		
02F2	0534	*		
02F2 20	0535	MUXFIELD	DB 32	V2/R12 MUX
02F3 01	0536	DB	1	V12/R11
02F4 06	0537	DB	6	6-POLE SELECT
02F5 08	0538	DB	8	V1/SC
02F6 21	0539	DB	32+1	V2+V12 TOGETHER
02F7 27	0540	DB	32+1+6	V2+V12+6POLE
02F8 2F	0541	DB	32+1+6+8	V2+V12+6POLE+V1
02F9 10	0542	DB	16	B AMPLIFIER
02FA	0543	*		
02FA	0544	*	FILTER BOARD CONTROL ROUTINES	
02FA	0545	*		
02FA 21 00 00	0546	FILCLEAR	LXI H,0	CLEAR THE SELECTS
02FD 11 00 00	0547	LXI	D,0	CLEAR THE DATA REGS
0300	0548	*		
0300	0549	*	FILTER SEND. ON ENTRY (DE)=DATA AND (L)=SELECT BITS	

0300	0550 †
0300 E5	0551 FSEND PUSH H SEND MUX BITS DOWN
0301 3A 14 20	0552 LDA MUXCOPY WITHOUT THE LATCHING
0304 67	0553 MOV H,A
0305 CD 7B 02	0554 CALL SHFB
0308 E1	0555 POP H
0309 EB	0556 XCHG . NOW SEND BITS FROM [DE]
030A CD 7B 02	0557 CALL SHFB
030D	0558 †
030E 7B	0559 MOV A,E INVERT SELECT BITS
030E 2F	0560 CMA .
030F 67	0561 MOV H,A
0310 CD 7B 02	0562 CALL SHFB
0313	0563 †
0313 3E 12	0564 MVI A,NWR+FILSTB STROBE THE FILTER
0315 D3 F3	0565 OUT SERIAL BOARD
0317 C9	0566 RET .
0318	0567 †
0318	0568 † MOTOR CONTROL SECTION
0318	0569 † ON ENTRY: A=MOTOR#, CARRY=1 FOR ON, 0 FOR OFF
0318	0570 †
0318	0571 †
0318 6F	0572 SETMOTOR ADC A CONVERT # AND ON/OFF
0319 CD 80 00	0573 CALL UNARY INTO BIT NUMBER
031C CD 49 03	0574 CALL FLIP INVERT THAT BIT
031F 3E 0A	0575 MVI A,10 DELAY 50 MILLISECONDS
0321 CD A7 03	0576 SMDLA CALL DLAS
0324 3D	0577 DCR A
0325 C2 21 03	0578 JNZ SMDLA
0328 C3 49 03	0579 JMP FLIP AND RESET
032B	0580 †
032B	0581 † CYCLE THE KELLEY GAIN CHANGE.
032B	0582 †
032B 2E 80	0583 SETKLY MVI L,KLYGC
032D C3 46 03	0584 JMP CYCLE
0330	0585 †
0330	0586 † OUTPUT THE MAGNETOMETER DATA TO THE PLASMA EXP.
0330	0587 † ON ENTRY: [HL]=16 BIT VALUE TO SEND
0330	0588 †
0330 E5	0589 SETPLA PUSH H
0331 2E 40	0590 MVI L,MAGLOAD INDICATE REGISTER LOADING
0332 CD 49 03	0591 CALL FLIP
0336 E1	0592 POP H
0337 7C	0593 MOV A,H SET HIGH BYTE
0338 2F	0594 CMA . INVERT FOR THE OPEN COLLECTOR
0339 03 D0	0595 OUT LEPAH
033B 7D	0596 MOV A,L AND LOW BYTE
033C 2F	0597 CMA .
033D D3 90	0598 OUT LEPAL
033F 2E 40	0599 MVI L,MAGLOAD RE-ENABLE SHIFTING

0341 C3 49 03	0600 JMP FLIP
0344	0601 †
0344 2E 20	0602 RBURST MVI L, BRUN RESET BURST
0346 C0 49 03	0603 CYCLE CALL FLIP CYCLE=FLIP TWICE
0349 3A 00 20	0604 FLIP LDA MKBCOPY PICK UP COPY OF PORT
034C AD	0605 XRA L INVERT THE BIT IN L
034D 03 C0	0606 OUT MOTORS
034F 32 00 20	0607 STA MKBCOPY
0352 C9	0608 RET .
0353	0609 †
0353	0610 † SEND [HL] TO THE BURST PROCESSOR
0353	0611 † ON EXIT: CARRY SET IF FAILURE TO COMMUNICATE
0353	0612 †
0353 20	0613 SEND DB RIM IF BURST RDY NOW
0354 07	0614 RLC . RECOVER BY RECEIVING
0355 DC 7E 03	0615 CC RECOVER
0358	0616 †
0358 E5	0617 PUSH H SAVE HL AND BC
0359 C5	0618 PUSH B
035A 01 10 00	0619 LXI B, 16 C=#BITS TO SEND
035D 3E C0	0620 MVI A, SDOEN+SDOEN SDO=1 (REQUEST ATTN)
035F 30	0621 DB SIM
0360 05	0622 WTBRS DCR B WAIT MAX OF 3.9 MSEC
0361 37	0623 STC .
0362 CA 77 03	0624 JZ SENDX
0365 20	0625 DB RIM WAITING FOR BURST TO RESPOND
0366 07	0626 RLC . (READY IF 1)
0367 D2 60 03	0627 JNC WTBRS
036A 00	0628 NOP . DISABLE WORD AND CMD INTERRUPTS ONLY
036B 3E 4D	0629 MVI A, SDOEN+MSE+DIS75+DIS55 SDO=0 TO START
036D 30	0630 DB SIM
036E 3E 80	0631 SEND1 MVI A, SDOEN*2 GET THE NEXT BIT
0370 29	0632 DAD H BY SHIFTING THRU MSB OF HL
0371 1F	0633 RAR . THEN INTO MSB OF ACCUM
0372 30	0634 DB SIM
0373 0D	0635 DCR C COUNT # BITS
0374 C2 6E 03	0636 JNZ SEND1
0377 3E 48	0637 SENDX MVI A, SDOEN+MSE LEAVE SDO=0, INTS ON
0379 30	0638 DB SIM
037A FB	0639 EI . RESTART INTERRUPTS
037B C1	0640 POP B
037C E1	0641 POP H
037D C9	0642 RET
037E	0643 †
037E E5	0644 RECOVER PUSH H GRAB THE INFO
037F CD 84 03	0645 CALL RECEIVE FROM THE BURST
0382 E1	0646 POP H AND THROW AWAY
0383 C9	0647 RET
0384	0648 †
0384	0649 † RECEIVE DATA FROM THE BURST


```

0384          0650 * ON EXIT: IF NOT ZERD, [HL]=16 BITS FROM BURST
0384          0651 *           ELSE NOTHING READY
0384          0652 *
0384 20       0653 RECEIVE DB   RIM   IF REQUEST FROM BURST
0385 E6 80   0654          ANI   SDO   IS NOT PRESENT, RETURN NOW.
0387 C8     0655          RZ    .
0388        0656 *
0388 C5     0657          PUSH  B     SAVE REGISTERS
0389 EB     0658          XCHG  .
038A        0659 *
038A 01 10 00 0660          LXI   B,16  SHIFT # BITS
038D 00     0661          NOF   .     DISABLE ONLY WORD AND CMD INTERRUPTS
038E 3E CD   0662          MVI   A,SDO+SDDEN+MSE+DIS75+DIS55  ANSWER WITH SDO=I
0390 30     0663          DB    SIM
0391 20     0664 RECVT DB   RIM   WAIT FOR START EDGE
0392 04     0665          INR   B     BUT DON'T WAIT FOREVER
0393 AE     0666          XRA   B     IF B=12B, QUIT
0394 FA 91 03 0667          JM    RECVT
0397        0668 *
0397 0A     0669 RECBIT LDAX  B     7 CYCLE DELAY
0398 20     0670          DB    RIM   GET THE BIT
0399 07     0671          RLC
039A 18     0672          DB    SLDE  SHIFT INTD DE
039B 0D     0673          DCR   C
039C C2 97 03 0674          JNZ   RECBIT
039F EB     0675          XCHG  .     PUT RESULT INTO HL
03A0 3E 4B   0676          MVI   A,SDDEN+MSE  SDO=0, INTS ON
03A2 30     0677          DB    SIM
03A3 B7     0678          ORA   A     RETURN NOT-ZERO
03A4 C1     0679          POP   B
03A5 FB     0680          EI    .     RE-ENABLE INTERRUPTS
03A6 C9     0681          RET
03A7        0682 *
03A7        0683 * DELAY ROUTINE
03A7        0684 *
03A7 F5     0685 DLAS  PUSH  PSW   WAIT 5 MILLISECONDS
03A8 D5     0686          PUSH  D
03A9 11 FE 01 0687          LXI   D,510  2.5 MHZ/23 CYCLES
03AC 18     0688 DLI  DCX   D     BY COUNTING
03AD 7B     0689          MOV   A,E
03AE B2     0690          ORA   D
03AF C2 AC 03 0691          JNZ   DLI
03B2 D1     0692          POP   D
03B3 F1     0693          POP   PSW
03B4 C9     0694          RET    .
03B5        0695 *
03B5        0696 * RESET WATCHDOG CIRCUIT
03B5        0697 *
03B5 3A 00 20 0698 RWATCH LDA  MRBCOPY PULSE THE RESET
03B8 F6 10   0699          ORI   WDRES  LINE TO THE 4015

```

03BA D3 C0	0700	OUT	MOTORS	COUNTER
03BC EE 10	0701	XRI	WDRES	
03BE D3 C0	0702	OUT	MOTORS	
03C0 C9	0703	RET	.	
03C1	0704	‡		
03C1	0705	‡	FIELD REPLACEMENT ROUTINE	
03C1	0706	‡	ON ENTRY: [A] = ORIGINAL 8-BIT VALUE	
03C1	0707	‡	[HL]->MASK OF THE FIELD	
03C1	0708	‡	[E] = NEW VALUE RIGHT ADJUSTED	
03C1	0709	‡		
03C1 4F	0710	REPFIELD MOV	C,A	SAVE ORIGINAL
03C2 7E	0711	MOV	A,M	REMOVE THE MASKED FIELD
03C3 2F	0712	CMA		
03C4 A1	0713	ANA	C	
03C5 4F	0714	MOV	C,A	
03C6	0715	‡		
03C6 EP	0716	XCHG	.	[DE]->MASK, L=VALUE
03C7 IA	0717	LDAX	D	SHIFT THE VALUE INTO POSITION
03C8 0F	0718	RF1 RRC	.	BY SHIFTING THE MASK
03C9 DA D0 03	0719	JC	RF2	THE OTHER WAY UNTIL A BIT
03CC 29	0720	DAD	H	
03CD C3 CB 03	0721	JMP	RF1	
03D0 IA	0722	RF2 LDAX	D	MASK THE SHIFTED VALUE
03D1 A5	0723	ANA	L	
03D2 B1	0724	ORA	C	AND APPLY TO MODIFIED ORIGINAL
03D3 C9	0725	RET		
03D4	0726	NEXT EQU	‡	
03D4	0727	COM	NEXT	
03D4	0728	‡		
03D4	0729	‡	RAM SECTION	
03D4	0730	‡		
03D4	0731	ORG	IORAM	
2000	0732	MKBCOPY DS	1	COPY OF MOTOR PORT
2001	0733	OLDSEL DS	1	COPY OF RELAY SELECT BITS
2002	0734	SUNCOPY DS	1	COPY OF SUN BIT
2003	0735	IOSTAT EQU	‡	DIGITAL STATUS RETURNED FROM IO
2003	0736	SVTRIM DS	2	STATUS OF VTRIMS,ETC
2005	0737	SBIAS DS	4	
2009	0738	SSTUB DS	2	
200B	0739	SGUARD DS	2	
200D	0740	SFILTER DS	7	
2014	0741	MUXCOPY DS	1	COPY OF FILTER MUX
2015	0742	RS DS	3	RELAY BITS 0-23

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE---FAST FLOATING POINT
0000          0003 * WRITTEN BY PETER HARVEY
0000          0004 * FILE : FFP.A
0000          0005 *
0000          0006 * F.P. REGISTER IS CDE.
0000          0007 * FORMAT IS SIGN(1)+EXP(7)+MANTISSA(16)
0000          0008 * NO HIDDEN BIT
0000          0009 *
0000          0010 PSW EQU 6
0000          0011 SP EQU 6
0000          0012 *
0000          FFP EQU NEXT
0000          0013 ORG FFP
0304          0014 COM LODFP
0304          0015 COM STOFFP
0304          0016 COM FMUL
0304          0017 COM FDIV
0304          0018 COM FADD
0304          0019 COM FSUB
0304          0020 COM FCMP
0304          0021 COM FNEG
0304          0022 COM FLT32
0304          0023 COM FIX32
0304          0024 COM FSQUA
0304          0025 COM FSQRT
0304          0026 COM MU21
0304          0027 *
0304 4E       0028 LODFP MOV C,M
0305 23       0029 INX H
0306 56       0030 MOV D,M
0307 23       0031 INX H
0308 5E       0032 MOV E,M
0309 C9       0033 RET .
030A         0034 *
030A 71       0035 STOFFP MOV M,C
030B 23       0036 INY H
030C 72       0037 MOV M,D
030D 23       0038 INX H
030E 73       0039 MOV M,E
030F C9       0040 RET .
03E0         0041 *
03E0         0042 * F.P. MULTIPLY ROUTINE
03E0         0043 *
03E0 7A       0044 FMUL MOV A,D IF Y=0, QUIT NOW
03E1 97       0045 ORA A
03E2 C8       0046 RZ .
03E3 46       0047 MOV B,M LOAD PARAM FROM MEM
03E4 23       0048 INX H INTO BHL FORMAT
  
```

03E5 7E	0049	MOV	A,M	
03E6 B7	0050	DRA	A	IF ZERO THEN SET TO 0
03E7 CA 70 05	0051	JZ	RET0	
03EA 23	0052	INX	H	ELSE LOAD THE REST
03EB 6E	0053	MOV	L,M	
03EC 67	0054	MOV	H,A	
03ED	0055	*		
03ED 78	0056 FM533	MOV	A,B	IF SAME SIGN, 60
03EE A9	0057	XRA	C	
03EF F2 FD 03	0058	JP	FMU33	
03F2 CD 81 05	0059	CALL	STRIF	REMOVE SIGNS FROM B&C
03F5 CD FD 03	0060	CALL	FMU33	MULTIPLY THEN NEGATE
03F8	0061	*		
03F8	0062 FNEG	EGU	*	
03F8 79	0063 NEGFP	MOV	A,C	AND NEGATE F.P
03F9 EE 80	0064	XRI	80H	
03FB 4F	0065	MOV	C,A	
03FC C9	0066	RET	.	
03FD	0067	*		
03FD	0068	*	F.P.	MULTIPLY POSITIVES ONLY
03FD	0069	*		
03FD 78	0070 FMU33	MOV	A,B	ADD EXPONENTS
03FE B1	0071	ADD	C	
03FF D6 40	0072	SUI	40H	ADJUST BACK TO EXCESS 64
0401 FA 76 05	0073	JM	ERCHK	IF MINUS, CHECK THE ERROR
0404 4F	0074	MOV	C,A	
0405 CD F0 05	0075	CALL	MU22F	[AHL.] = DE X HL
0408 C3 5C 05	0076	JMP	NCHK	SHIFT UNTIL AHL NORMED, ROUND OFF
040B	0077	*		
040B	0078	*	F.P.	DIVIDE
040B	0079	*		
040B 7A	0080 FDIV	MOV	A,D	IF ZERO DIVIDEND, QUIT
040C B7	0081	DRA	A	
040D C8	0082	RZ	.	
040E 46	0083	MOV	B,M	PICK UP DIVISOR
040F 23	0084	INX	H	
0410 7E	0085	MOV	A,M	
0411 B7	0086	DRA	A	IF DIVISOR 0, OVERFLOW
0412 CA 7B 05	0087	JZ	OVERFLOW	
0415 23	0088	INX	H	
0416 6E	0089	MOV	L,M	
0417 67	0090	MOV	H,A	
0418	0091	*		
0418 78	0092	MOV	A,B	IF SAME SIGN, DO
0419 A9	0093	XRA	C	SAME SIGNED VERSION
041A F2 26 04	0094	JP	FDU33	
041D CD 81 05	0095	CALL	STRIF	REMOVE SIGNS
0420 CD 26 04	0096	CALL	FDU33	DIVIDE OUT
0423 C3 FB 03	0097	JMP	NEGFP	AND NEGATE
0426	0098	*		

0426 79	0099 FDU33	MOV	A,C	EXP=C-B+40H
0427 90	0100	SUB	B	
0428 C6 40	0101	AOI	40H	
042A FA 76 05	0102	JM	ERCHK	
042B 4F	0103	MOV	C,A	
042E C5	0104	PUSH	B	SAVE EXPONENT
042F	0105	†		
042F 7C	0106	MOV	A,H	BC=-DIVISOR
0430 2F	0107	CMA		
0431 47	0108	MOV	B,A	
0432 7D	0109	MOV	A,L	
0433 2F	0110	CMA		
0434 4F	0111	MOV	C,A	
0435 03	0112	INX	B	
0436	0113	†		
0436	0114	†	IF THE REMAINDER STARTS AS LARGE AS	
0436	0115	†	THE DIVISOR, THE FIRST BIT IS 1	
0436	0116	†		
0436 62	0117	MOV	H,D	HL=REMAINDER
0437 6B	011E	MOV	L,E	
0438 09	0119	OAD	B	HL=REMAINDER-DIVISOR
0439 DA 56 04	0120	JC	FBIT1	
043C	0121	†		
043C	0122	†	IF REMAINDER LESS THAN DIVISOR, THE FIRST	
043C	0123	†	BIT (INTEGER PART) IS ZERO. DIVIDE FOR	
043C	0124	†	FRACTIONAL PART WHICH WILL BE AUTOMATICALLY	
043C	0125	†	NORMALIZED.	
043C	0126	†		
043C EB	0127	XCHG	.	HL=REMAINDER AGAIN
043D 3E 10	0128	MVI	A,16	
043F CD 6F 04	0129	CALL	FDSHF	[DE]=[HL]*2/[BC]
0442 29	0130	DAD	H	IF REMAINDER > 8000H
0443 DA 4C 04	0131	JC	DVROND	THEN ROUND UP
0446 09	0132	DAD	B	IF NEXT BIT WOULD BE 1
0447 DA 4C 04	0133	JC	DVROND	THEN ROUND UP
044A C1	0134	POP	B	RESTORE EXPONENT
044B C9	0135	RET	.	NO NORMALIZATION REQD
044C	0136	†		
044C C1	0137	DVROND	POP	B C=EXPONENT
044D 1C	0138	ROND	INR	E ROUND OFF OE
044E C0	0139	RNZ	.	BUT DON'T PRODUCE
044F 14	0140	INR	D	A ZERO
0450 C0	0141	RNZ	.	
0451 11 00 00	0142	LXI	D,8000H	IF ZERO, THEN
0454 0C	0143	INR	C	UP THE EXPONENT
0455 C9	0144	RET	.	
0456	0145	†		
0456	0146	†	FIRST BIT=1. DIVIDE OUT 16 MORE BITS	
0456	0147	†	USING WHAT'S LEFT OF THE REMAINDER IN HL	
0456	0148	†		

0456 3E 10	0149 FRIT1 MVI A,16
0458 11 FF FF	0150 LXI B,-1
045B CD 6F 04	0151 CALL FDSHF [DE]=[HL]/[BC]
045E	0152 *
045E C1	0153 POP B RESTORE THE EXPONENT
045F 0C	0154 INR C ADJUST SINCE 1ST BIT=1
0460 37	0155 STC RIGHT SHIFT A 1 INTO DE
0461 7A	0156 MOV A,D
0462 1F	0157 RAR
0463 57	0158 MOV D,A
0464 7B	0159 MOV A,E
0465 1F	0160 RAR
0466 5F	0161 MOV E,A
0467 D0	0162 RNC IF 17TH BIT WAS 0, STOP
0468 C3 4D 04	0163 JMP ROND ELSE ROUND OFF
046B	0164 *
046B	0165 * DIVIDE NORMALIZED INTEGERS FOR F.P.
046B	0166 *
046B 33	0167 FDSTK INX SP REMOVE PARTIAL REMAINDER
046C 33	0168 INX SP FROM STACK
046D 3D	0169 FDTST DCR A DECR BIT COUNTER
046E C8	0170 RZ .
046F 29	0171 FDSHF DAD H BRING DOWN A BIT INTO REM
0470 DA 81 04	0172 JC SUBIT IF >=10000, THEN SUBTRACT
0473 EB	0173 XCHG .
0474 29	0174 DAD H AND SHIFT RESULT REG
0475 EB	0175 XCHG
0476	0176 *
0476 1C	0177 INR E ASSUME RESULT=1
0477 E5	0178 F0V22 PUSH H SAVE REMAINDER ON STK
0478 09	0179 DAD B IF REM/DIVISOR, LEAVE REM ALONE
0479 DA 6B 04	0180 JC FDSHF
047C E1	0181 POP H ELSE RESTORE REMAINDER
047D 1D	0182 DCR E SET RESULT BIT=0
047E C3 6D 04	0183 JMP FDTST
0481	0184 *
0481 EB	0185 SUBIT XCHG . FINISH THE SHIFT
0482 29	0186 DAD H
0483 EB	0187 XCHG .
0484 09	0188 DAD B SUBTRACT DIVISOR
0485 1C	0189 INR E SET RESULT BIT
0486 C3 6D 04	0190 JMP FDTST
0489	0191 *
0489	0192 * F.P. COMPARE
0489	0193 * ON EXIT: ZERO SET IF EQUAL, CARRY IF LESS THAN
0489	0194 * CDE UNTOUCHED
0489	0195 *
0489 C5	0196 F0MP PUSH B SAVE CDE
048A D5	0197 PUSH D
048B CD 9B 04	0198 CALL FSUB SUBTRACT THE TWO

048E 7A	0199	MOV	A,D	IF RESULT=0, RET
048F B7	0200	ORA	A	
0490 CA 95 04	0201	JZ	FCMPX	
0491 79	0202	MOV	A,C	IF NEGATIVE, THEN
0494 07	0203	PLC	.	SET CARRY, ELSE NO CARRY
0495 D1	0204	FCMPX	POP	D RESTORE CDE
0496 C1	0205	POP	B	
0497 C9	0206	RET	.	
0498	0207	*		
0498	0208	* F.P.	SUB	
0498	0209	*		
049B 7E	0210	FSUB	MOV	A,M INVERT SIGN OF 2ND
0499 EE 80	0211	XRI	80H	PARAMETER
049B 47	0212	MOV	B,A	
049C C3 A0 04	0213	JMP	FAD1	
049F	0214	*		
049F	0215	* F.P.	ADD	
049F	0216	*		
049F 46	0217	FADD	MOV	B,M LOAD UP
04A0 23	0218	FAD1	INX	H
04A1 7E	0219	MOV	A,M	
04A2 23	0220	INX	H	
04A3 6E	0221	MOV	L,M	
04A4 67	0222	MOV	H,A	
04A5 97	0223	SUB	A	
04A6 BC	0224	CMF	H	IF BHL=0, QUIT
04A7 C8	0225	RZ	.	
04A8 BA	0226	CMF	D	IF CDE=0, QUIT
04A9 CA 00 05	0227	JZ	SWITCH	
04AC	0228	*		
04AC 79	0229	MOV	A,C	COMPUTE EXP DIFFERENCE
04AD 90	0230	SUB	B	
04AE 87	0231	ADD	A	
04AF F2 58 04	0232	JP	POSDX	
04B2 78	0233	MOV	A,B	SWAP CDE FOR BHL
04B3 41	0234	MOV	B,C	
04B4 4F	0235	MOV	C,A	
04B5 EB	0236	XCHG	.	
04B6 90	0237	SUB	B	COMPUTE EXP DIFFERENCE
04B7 87	0238	ADD	A	AGAIN
04B8 CA C2 04	0239	POSDX	JZ	ADSUB
04BB 0F	0240	RRC	.	DIV BY 2
04BC FE 10	0241	CPI	15	IF CDE>>BHL, QUIT
04BE D9	0242	RNC		
04BF CD F4 04	0243	CALL	SHFHL	REDUCE HL A TIMES
04C2	0244	*		
04C2 78	0245	ADSUB	MOV	A,B IF SIGNS DIFFER, GO
04C3 A9	0246	XRA	C	
04C4 FA D2 04	0247	JM	DIFFER	
04C7 19	0248	DAD	D	ADD DE TO HL

04C8 EB	0249	XCHG	.	IF NO CARRY,
04C9 D0	0250	RNC	.	RETURN CDE
04CA	0251 †			
04CA 7A	0252	RITE1	MOV A,D	ELSE SHIFT RIGHT ONE
04CB 1F	0253	RAR	.	INCLUDING THE CARRY
04CC 57	0254	MOV	D,A	
04CD 7B	0255	MOV	A,E	
04CE 1F	0256	RAR		
04CF 5F	0257	MOV	E,A	
04D0 0C	0258	INR	C	ADJUST EXPONENT
04D1 C9	0259	RET	.	
04D2	0260 †			
04D2 7B	0261	DIFFER	MOV A,E	IF DE<HL,
04D3 95	0262	SUB	L	THEN NORM(B:HL-DE)
04D4 7A	0263	MOV	A,D	
04D5 9C	0264	SBB	H	
04D6 DA E7 04	0265	JC	SUBD	
04D9	0266 †			
04D9 57	0267	MOV	D,A	ELSE NORM(C:DE-HL)
04DA 7B	0268	MOV	A,E	
04DB 95	0269	SUB	L	
04DC 5F	0270	MOV	E,A	
04DD 21 00 00	0271	LXI	H,0	
04E0 82	0272	DRA	D	IF DE<>0,
04E1 C2 21 05	0273	JNZ	NORM	NORMALIZE WITH C
04E4 0E 00	0274	MVI	C,0	IF HL-DE=0, RETURN
04E6 C9	0275	RET	.	
04E7	0276 †			
04E7 7D	0277	SUBD	MOV A,L	DE = DE - HL
04E8 93	0278	SUB	E	
04E9 5F	0279	MOV	E,A	
04EA 7C	0280	MOV	A,H	
04EB 9A	0281	SBB	D	
04EC 57	0282	MOV	D,A	
04ED 4B	0283	MOV	C,B	USE BHL'S EXPONENT
04EE 21 00 00	0284	LXI	H,0	AND SHIFT IN ZEROES
04F1 C3 21 05	0285	JMP	NORM	NORMALIZE
04F4	0286 †			
04F4	0287 †	SHIFT HL RIGHT A TIMES		
04F4	0288 †			
04F4 D6 08	0289	SHFHL	SUI 8	IF LT 8, GO NOW
04F6 DA FF 04	0290	JC	LT8	
04F9 6C	0291	MOV	L,H	SHIFT 8
04FA 26 00	0292	MVI	H,0	
04FC C8	0293	RZ	.	IF EXACTLY 8, RETURN
04FD D6 08	0294	SUI	8	ELSE DO SECOND 8
04FF C5	0295	LT8	PUSH B	SAVE EXPONENTS
0500 47	0296	MOV	B,A	SAVE INVERTED COUNTER
0501 97	0297	SUB	A	CLEAR ACCUM
0502 29	0298	SHF1	DAD H	SHIFT LEFT

0503 8F	0299	ADC	A	INTO A FROM HL
0504 04	0300	INR	B	COUNT UP TO 0
0505 C2 02 05	0301	JNZ	SHF1	
0508 C1	0302	PDF	B	RESTORE EXPS
0509 6C	0303	MOV	L,H	
050A 67	0304	MOV	H,A	
050E C9	0305	RET	.	
050C	0306 *			
050C EB	0307 SWITCH XCHG	.		CDE= BHL
050D 48	0308	MOV	C,B	
050E C9	0309	RET	.	
050F	0310 *			
050F	0311 *	CONVERT 32 BIT DATA TO F.P. FORMAT		
050F	0312 *			
050F 7A	0313 FLT32	MOV	A,D	IF POSITIVE, JUST NORM
0510 B7	0314	ORA	A	
0511 0E 60	0315	MVI	C,64+32	WITH LSB=2440 TO BEGIN
0513 F2 21 05	0316	JP	NORM	
0516 C0 59 06	0317	CALL	NEG32	NEGATE DEHL
0519 C0 21 05	0318	CALL	NORM	NOW NORMALIZE
051C 79	0319	MOV	A,C	AND NEGATE FP
051D F6 80	0320	ORI	80H	
051F 4F	0321	MOV	C,A	
0520 C9	0322	RET	.	
0521	0323 *			
0521	0324 *	NORMALIZE C:DEHL TO F.P. NORMAL FORM		
0521	0325 *			
0521 79	0326 NORM	MOV	A,C	IF C NEGATIVE, TRAP IT
0522 B7	0327	ORA	A	
0523 F2 2F 05	0328	JP	NORMP	
0526 E6 7F	0329	ANI	7FH	
0528 4F	0330	MOV	C,A	
0529 C0 2F 05	0331	CALL	NORMP	
052C C3 F8 03	0332	JMP	NEGFP	AND NEG LATER
052F	0333 *			
052F 7A	0334 NORMP	MOV	A,D	IF WITHIN 8 BITS, GO NOW
0530 B7	0335	ORA	A	
0531 C2 59 05	0336	JNZ	NORM1	
0534 B3	0337	ORA	E	IF WITHIN 16, USE EHL
0535 C2 4E 05	0338	JNZ	NRMEHL	
0538 B4	0339	ORA	H	IF WITHIN 24, USE HL
0539 C2 48 05	0340	JNZ	NRMHL	
053C B5	0341	ORA	L	IF JUST L, USE IT
053D C2 42 05	0342	JNZ	NRML	
0540 4A	0343	MOV	C,D	ELSE CDE=0
0541 C9	0344	RET	.	
0542	0345 *			
0542 55	0346 NRML	MOV	D,L	LOO FOR 3 BYTES
0543 06 18	0347	MVI	B,24	ADJUST EXP BY 24 BITS
0545 C3 53 05	0348	JMP	AJEXP	

054B EB	0349 NRMHL	XCHG	.	HLO FOR 3BYTES
0549 06 10	0350	MVI	B,16	ADJUST EXP 16
054B C3 53 05	0351	JMP	AJEXP	
054E 53	0352 NRMEHL	MOV	D,E	SHIFT EHL TO DEH
054F 5C	0353	MOV	E,H	
0550 65	0354	MOV	H,L	
0551 06 08	0355	MVI	B,8	ADJUST B BITS
0553 79	0356 AJEXP	MOV	A,C	EXP=EXP-B
0554 90	0357	SUB	B	
0555 4F	0358	MOV	C,A	IF PROBLEM, THEN UNDER
0556 DA 70 05	0359	JC		UNDERFLOW
0559	0360	*		
0559	0361	*		BIT BY BIT NORMALIZATION
0559	0362	*		
0559 7A	0363 NORM1	MOV	A,D	AHL=DEH
055A 6C	0364	MOV	L,H	
055B 63	0365	MOV	H,E	
055C B7	0366 NCHK	ORA	A	SHIFT AHL TILL NORMED
055D FA 66 05	0367	JM	NRMFIN	
0560 0D	0368 NCHK1	DCR	C	EXP<-EXP-1
0561 29	0369	DAD	H	
0562 BF	0370	ADC	A	
0563 F2 60 05	0371	JP	NCHK1	
0566 57	0372 NRMFIN	MOV	D,A	DE=AH
0567 5C	0373	MOV	E,H	
056B 7D	0374	MOV	A,L	IF MSB(L)=1, ROUND OFF DE
0569 07	0375	RLC	.	
056A DC 4D 04	0376	CC	ROND	
056D 79	0377	MOV	A,C	IF EXP POSITIVE, OK
056E B7	0378	ORA	A	
056F F0	0379	RP	.	
0570	0380	*		
0570	0381	*		ERRORS : UNDERFLOW AND OVERFLOW
0570	0382	*		
0570	0383	*		UNDERFLOW EQU *
0570 0E 00	0384 RETO	MVI	C,0	RETURN CDE=0
0572 11 00 00	0385	LXI	D,0	
0575 C9	0386	RET	.	
0576	0387	*		
0576 FE C0	0388 ERCHK	CPI	OCHO	IF BETWEEN OBOH AND OBFH
057B D2 70 05	0389	JNC		UNDERFLOW THEN UNDERFLOW, ELSE OVER
057B	0390	*		
057B 0E 7F	0391 OVERFLOW	MVI	C,7FH	RETURN CDE=MAXIMUM
057D 11 FF FF	0392	LXI	D,-1	
0580 C9	0393	RET	.	
05B1	0394	*		
05B1 78	0395 STRIP	MOV	A,B	REMOVE SIGNS FROM B
05B2 E6 7F	0396	ANI	7FH	
05B4 47	0397	MOV	B,A	
0585 79	0398	MOV	A,C	

0586 E6 7F	0399	ANI	7FH	
0588 4F	0400	MOV	C,A	
0589 C9	0401	RET	.	
058A	0402	*		
058A	0403	* FIX32:	FLT TO FIX CONVERSION	
058A	0404	*		
058A 79	0405	FIX32 MOV	A,C	IF NEGATIVE, INVERT
0588 EE 80	0406	XRI	80H	RESULTS
058D FA 97 05	0407	JM	FIXPOS	
0590 4F	0408	MOV	C,A	
0591 CD 97 05	0409	CALL	FIXPOS	
0594 C3 69 06	0410	JMP	NEG32	
0597	0411	*		
0597 E6 7F	0412	FIXPOS ANI	7FH	IF CDE<1, RETURN(0)
0599 FE 41	0413	CPI	41H	
0598 DA C5 05	0414	JC	ZERDH	
059E FE 60	0415	CPI	60H	IF >2**31, MAX IT
05A0 D2 CC 05	0416	JNC	MAXDH	
05A3	0417	*		
05A3 21 00 00	0418	LXI	H,0	ELSE SHIFT MANTISSA
05A6 D6 50	0419	SUI	40H+16	IF 2**16, QUIT
05A8 C8	0420	RZ	.	
05A9 EB	0421	XCHG	.	DEHL=00XX, READY TO SHIFT
05AA D2 BC 05	0422	JNC	SHDH	IF EXP WAS 51 TO 5F, 60
05AD C6 10	0423	ADI	16	ELSE 41-4F, SHIFT THEN
05AF CD BC 05	0424	CALL	SHDH	DIVIDE BY 2**16
0582 EB	0425	XCHG		
0583 11 00 00	0426	LXI	D,0	
0586 C9	0427	RET	.	
0587	0428	*		
0587 29	0429	SHCAR	DAD H	SHIFT DE PART
0588 2C	0430	INR	L	AND PUT IN CARRY
0589 EB	0431	DECRA	XCHG	SWAP BACK HL
058A 3D	0432	DCR	A	IF COUNT=0, QUIT
0588 C8	0433	RZ	.	
058C 29	0434	SHDH	DAD H	SHIFT HL ONE BIT
058D EB	0435	XCHG	.	IF CARRY, THEN
058E DA B7 05	0436	JC	SHCAR	UPDATE DE WITH CARRY
05C1 29	0437	DAD	H	ELSE WITHOUT CARRY
05C2 C3 B9 05	0438	JMP	DECRA	
05C5	0439	*		
05C5 11 00 00	0440	ZERDH	LXI D,0	DEHL=0
05C8 21 00 00	0441	LXI	H,0	
05CB C9	0442	RET	.	
05CC 11 FF 7F	0443	MAXDH	LXI D,7FFFH	DEHL=MAXIMUM
05CF 21 FF FF	0444	LXI	H,-1	
05D2 C9	0445	RET	.	
05D3	0446	*		
05D3	0447	* SQUARE [CDE]		
05D3	0448	*		

05D3 7A	0449 FSQUA	MOV	A,D	CHECK FOR 0
05D4 B7	0450	ORA	A	
05D5 CB	0451	RZ	.	
05D6 41	0452	MOV	B,C	BHL=CDE
05D7 62	0453	MOV	H,D	
05DB 6B	0454	MOV	L,E	
05D9 C3 ED 03	0455	JMP	FMS33	
05DC	0456	*		
05DC 7A	0457 FSQRT	MOV	A,D	IF ZERO, QUIT
05DD B7	045B	ORA	A	
05DE CB	0459	RZ	.	
05DF 79	0460	MOV	A,C	IF ODD EXPONENT, SHIFT
05E0 E6 01	0461	ANI	1	
05E2 C4 CA 04	0462	CNZ	RITE1	
05E5 C5	0463	PUSH	B	SAVE EXPONENT
05E6 CD 44 06	0464	CALL	SQR2	DE=DE**1/2
05E9 C1	0465	POP	B	
05EA 79	0466	MOV	A,C	DIVIDE EXP BY 2
05EB 0F	0467	RRC		
05EC C6 20	046B	ADI	20H	IN EXCESS 64
05EE 4F	0469	MOV	C,A	
05EF C9	0470	RET	.	
05F0	0471	*		
05F0	0472	*	16 X 16 MULTIPLY UNSIGNED. OPTIMIZED FOR F.P.	
05F0	0473	*	[AHL] = [HL] * [DE] TOP 3 BYTES	
05F0	0474	*		
05F0 97	0475 MU22F	SUB	A	IF E=0, DO SHORT MULT
05F1 BB	0476	CMF	E	
05F2 CA 08 06	0477	JZ	SHORD	
05F5 85	0478	ORA	L	IF L=0, DO SHORT WITH H
05F6 CA 09 06	0479	JZ	SHORH	
05F9 E5	0480	PUSH	H	AHL= L*DE
05FA CD 0A 06	0481	CALL	MU21	
05FD 6C	04B2	MOV	L,H	THROW AWAY LS BYTE
05FE 67	0483	MOV	H,A	SAVE UPPER BYTES
05FF E3	04B4	XTHL	.	SAVE EH, GET MS BYTE OF 1ST
0600	04B5	*		
0600 7C	0486	MOV	A,H	AHL=MSB*DE
0601 CD 0A 06	04B7	CALL	MU21	
0604 D1	04B8	POP	D	GRAB THE TWO STORED
0605 19	04B9	DAD	D	ADD PARTIAL RESULTS
0606 BB	0490	ADC	B	FOR THREE BYTES (AHL)
0607 C9	0491	RET	.	
0608	0492	*		
0608 EB	0493 SHORD	XCHG	.	SHORT MULT
0609 7C	0494 SHORH	MOV	A,H	JUST MULT H*DE
060A	0495	*		
060A	0496	*	16 X B MULTIPLY UNSIGNED	
060A	0497	*	[AHL] <- A * [DE]	
060A	0498	*	TAKES 198 TO 297 CYCLES	

060A	0499 ‡				
060A 21 00 00	0500 MU21 LXI	H,0	ZERO RESULT REG		
060D 44	0501 MOV	B,H	B<-0		
060E	0502 ‡				
060E 87	0503 MULTX ADD	A	SHIFT MSB TO CARRY		
060F D2 14 06	0504 JNC	X2			
0612 19	0505 DAD	D	IF C=1, THEN ADD [DE]		
0613 88	0506 ADC	B	IF OVERFLOW,BUMP MSBYTE		
0614 29	0507 X2 DAD	H	SHIFT FOR NEXT TEST		
0615	0508 ‡				
0615 8F	0509 ADC	A	AND SO ON		
0616 D2 1B 06	0510 JNC	X4			
0619 19	0511 DAD	D			
061A 88	0512 ADC	B			
061B 29	0513 X4 DAD	H			
061C	0514 ‡				
061C 8F	0515 ADC	A			
061D D2 22 06	0516 JNC	X8			
0620 19	0517 DAD	D			
0621 88	0518 ADC	B			
0622 29	0519 X8 DAD	H			
0623	0520 ‡				
0623 8F	0521 ADC	A			
0624 D2 29 06	0522 JNC	X10			
0627 19	0523 DAD	D			
0628 88	0524 ADC	B			
0629 29	0525 X10 DAD	H			
062A	0526 ‡				
062A 8F	0527 ADC	A			
062B D2 30 06	0528 JNC	X20			
062E 19	0529 DAD	D			
062F 88	0530 ADC	B			
0630 29	0531 X20 DAD	H			
0631	0532 ‡				
0631 8F	0533 ADC	A			
0632 D2 37 06	0534 JNC	X40			
0635 19	0535 DAD	D			
0636 88	0536 ADC	B			
0637 29	0537 X40 DAD	H			
0638	0538 ‡				
0638 8F	0539 ADC	A			
0639 D2 3E 06	0540 JNC	X80			
063C 19	0541 DAD	D			
063D 88	0542 ADC	B			
063E 29	0543 X80 DAD	H			
063F	0544 ‡				
063F 8F	0545 ADC	A			
0640 D0	0546 RNC				
0641 19	0547 DAD	D			
0642 88	0548 ADC	B			

0643 C9	0549	RET		
0644	0550 *			
0644	0551 *	INTEGER SQUARE ROOT OF DE		
0644	0552 *	[\]		
0644 01 00 80	0553 SQR2	LXI	B,8000H	GUESS=80, ROOT0=0
0647 CD 54 06	0554 SQRA1	CALL	APPX	CHECK APPROXIMATION
064A 78	0555	MOV	A,B	AND SHIFT APPX BIT
064B 0F	0556	RRC		
064C 47	0557	MOV	B,A	
064D D2 47 06	0558	JNC	SQRA1	
0650 51	0559	MOV	D,C	DE=RESULT
0651 1E 00	0560	MVI	E,0	
0653 C9	0561	RET	.	
0654	0562 *			
0654 D5	0563 APPX	PUSH	D	SAVE X
0655 78	0564	MOV	A,B	TRY NEW TEST BIT
0656 81	0565	ADD	C	
0657 5F	0566	MOV	E,A	
0658 16 00	0567	MVI	D,0	
065A C5	0568	PUSH	B	SAVE BC
065B CD 0A 06	0569	CALL	MU21	AHL=A#DE
065E C1	0570	POP	B	
065F D1	0571	POP	D	COMPARE TO X
0660 7B	0572	MOV	A,E	IF X < HL THEN TOO BIG
0661 95	0573	SUB	L	
0662 7A	0574	MOV	A,D	
0663 9C	0575	SBB	H	
0664 D8	0576	RC	.	
0665 78	0577	MOV	A,B	ELSE ADD TEST BIT TO C
0666 81	0578	ADD	C	
0667 4F	0579	MOV	C,A	
0668 C9	0580	RET	.	
0669	0581 *			
0669 CD 77 06	0582 NEG32	CALL	INV16	INVERT DEHL
066C E8	0583	XCHG		
066D CD 77 06	0584	CALL	INV16	
0670 E8	0585	XCHG	.	
0671 23	0586	INX	H	AND ADD 1
0672	0587 *			
0672 7C	0588	MOV	A,H	IF HL=0, INCR DE
0673 B5	0589	ORA	L	
0674 C0	0590	RNZ	.	
0675 13	0591	INX	D	
0676 C9	0592	RET	.	
0677	0593 *			
0677 7C	0594 INV16	MOV	A,H	INVERT HL
0678 2F	0595	CMA		
0679 67	0596	MOV	H,A	
067A 7D	0597	MOV	A,L	
067B 2F	0598	CMA		

067C 6F	0599	MOV	L,A
067D C9	0600	RET	.
067E	0601 NEXT	EQU	\$
067E		COM	NEXT

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE---MATRIX SOLVER
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 * FILE : MATRIX.A
0000          0005 *
0000          0006 MATR1 EQU    NEXT
0000          0007          ORG    MATR1
067E          0008          COM    IMATX  INIT MATRIX ADDR/SIZE
067E          0009          COM    SOLVE  SOLVE MATRIX
067E          0010 *
067E          0011 * INITIALIZE MATRIX SIZE AND ADDRESS
067E          0012 *
067E 22 92 24 0013 IMATX SHLD  MATRX  SAVE ADDRESS OF MATRIX
0681 E8        0014          XCHG
0682 22 94 24 0015          SHLD  RESULT  SAVE ADDRESS OF RESULT
0685 32 90 24 0016          STA   N      SAVE SIZE
0688 3C        0017          INR   A      SAVE SIZE+1
0689 32 91 24 0018          STA   NP1
068C C9        0019          RET   .
068D          0020 *
068D          0021 * MATRIX SOLVER
068D          0022 * RETURNS CARRY SET IF MATRIX ERROR
068D          0023 *
068D 3E 01     0024 SOLVE MV1  A,1   FOR J:=1 TO N-1
068F 32 97 24 0025          STA   J
0692 CD FF 06 0026 SOL1  CALL  LOCNZ  LOCATE NON-ZERO IN COL
0695 D8        0027          RC    .    IF NOT POSS, ERROR
0696 CD 5D 07 0028          CALL  SUBROWJ  SUBTRACT JTH FROM OTHERS
0699 21 97 24 0029          LX1  H,J    J=J+1
069C 34        0030          INR   M
069D 3A 90 24 0031          LDA   N      IF N=J, QUIT
06A0 BE        0032          CMP   M
06A1 C2 92 06 0033          JNZ   SOL1
06A4          0034 *
06A4          0035 * CALCULATE SOLUTION FROM A DIAGONAL MATRIX
06A4          0036 *
06A4 3A 90 24 0037          LDA   N      FOR M2:=N DOWNT0 1
06A7 32 9A 24 0038 CAL1  STA   M2
06AA 67        0039          MOV   H,A    LOAD ELEMENT (M2,N+1)
06AB 3A 91 24 0040          LDA   NP1
06AE 6F        0041          MOV   L,A
06AF CD D7 07 0042          CALL  LODZ
06B2          0043 *
06B2 3A 9A 24 0044          LDA   M2    DIVIDE BY ELEMENT (M2,M2)
06B5 6F        0045          MOV   L,A
06B6 67        0046          MOV   H,A
06B7 CD E3 07 0047          CALL  DIVZ
06BA          0048 *
06BA 2A 94 24 0049          LHLD  RESULT  STORE IN X(M2)

```



```

06BD CD BC 07      0050      CALL  REF XM2
06C0 CD DA 03      0051      CALL  STOFF
06C3                0052 *
06C3                0053 * IF TOP ROW, THEN WE'RE FINISHED.
06C3                0054 * ELSE UPDATE THE CONSTANTS ON THE RIGHT SIDE
06C3                0055 *
06C3 3A 9A 24      0056      LDA   M2      IF M2=1, QUIT
06C6 3D            0057      DCR   A
06C7 C8            0058      RZ   .
06C8 32 96 24      0059 UPCON STA   Q      FOR Q=M2-1 DOWNTD 1
06CB 67            0060      MOV   H,A      Z(Q,N+1)=Z(Q,N+1)-X(M2)*Z(Q,M2)
06CC 3A 9A 24      0061      LDA   M2
06CF 6F            0062      MOV   L,A
06D0 CD D7 07      0063      CALL  LODZ
06D3                0064 *
06D3 2A 94 24      0065      LHLD RESULT MULTIPLY BY X(M2)
06D6 CD BC 07      0066      CALL  REF XM2
06D9 CD E0 03      0067      CALL  FMUL
06DC                0068 *
06DC 79            0069      MOV   A,C      CHANGE SIGN OF RESULT
06DD EE 80          0070      XRI  80H
06DF 4F            0071      MOV   C,A
06E0                0072 *
06E0 3A 96 24      0073      LDA   Q      SUBTRACT FROM Z(Q,N+1)
06E3 67            0074      MOV   H,A
06E4 3A 91 24      0075      LDA   NP1
06E7 6F            0076      MOV   L,A
06E8 E5            0077      PUSH H
06E9 CD E9 07      0078      CALL  ADZ
06EC E1            0079      POP  H      STORE AT Z(Q,N+1)
06ED CD DD 07      0080      CALL  STRZ
06F0                0081 *
06F0 3A 96 24      0082      LDA   Q      Q=Q-1 UNTIL 0
06F3 3D            0083      DCR   A
06F4 C2 C8 06      0084      JNZ  UPCON
06F7                0085 *
06F7 3A 9A 24      0086      LDA   M2      M2=M2-1
06FA 3D            0087      DCR   A      UNTIL 0
06FB C2 A7 06      0088      JNZ  CAL1
06FE C9            0089      RET   .
06FF                0090 *
06FF                0091 * LOCATE A NON-ZERO IN COLUMN J
06FF                0092 *
06FF 3A 97 24      0093 LOCNZ LDA   J      FOR Q:=J TO N
0702 32 96 24      0094 LOC1 STA   Q
0705 67            0095      MOV   H,A      IS-ZERO Z(Q,J)?
0706 3A 97 24      0096      LDA   J
0709 6F            0097      MOV   L,A
070A CD EF 07      0098      CALL  ISZRO
070D C2 1D 07      0099      JNZ  SWAPROW NO. 60

```

0710 3A 96 24	0100	LDA	Q	YES. Q=Q+1
0713 3C	0101	INR	A	
0714 21 91 24	0102	LX1	H,NP1	UNTIL Q=N+1
0717 BE	0103	CMP	M	
071B C2 02 07	0104	JNZ	LOC1	
071B 37	0105	STC	.	ELSE RETURN CARRY
071C C9	0106	RET	.	
071D	0107	‡		
071D	0108	‡	SWAP QTH ROW WITH JTH ROW	
071D	0109	‡		
071D 3A 96 24	0110	SWAPROW LDA	Q	IF Q=J, WE'RE DONE
0720 21 97 24	0111	LX1	H,J	
0723 BE	0112	CMP	M	
0724 C8	0113	RZ	.	
0725 7E	0114	MOV	A,M	FOR KLM:=J TO NP1
0726 32 9B 24	0115	SWP1 STA	KLM	
0729 6F	0116	MOV	L,A	SWAP(Z(Q,KLM),Z(J,KLM))
072A 3A 96 24	0117	LDA	Q	
072D 67	0118	MOV	H,A	
072E CD CB 07	0119	CALL	REFZ	
0731 EB	0120	XCHG		
0732 3A 97 24	0121	LDA	J	
0735 67	0122	MOV	H,A	
0736 3A 9B 24	0123	LDA	KLM	
0739 6F	0124	MOV	L,A	
073A CD CB 07	0125	CALL	REFZ	
073D CD 4F 07	0126	CALL	SWAP3	
0740	0127	‡		
0740 3A 9B 24	0128	LDA	KLM	
0743 3C	0129	INR	A	
0744 21 91 24	0130	LX1	H,NP1	
0747 BE	0131	CMP	M	
074B DA 26 07	0132	JC	SWP1	
074B CA 26 07	0133	JZ	SWP1	
074E C9	0134	RET	.	
074F	0135	‡		
074F 0E 03	0136	SWAP3 MVI	C,3	SWAP F.P. DTYS
0751 1A	0137	SMPLP LDAX	D	GRAB DATA
0752 46	0138	MOV	B,M	FROM BOTH
0753 EB	0139	XCHG		
0754 12	0140	STAX	D	STORE FROM BOTH
0755 70	0141	MOV	M,B	
0756 23	0142	INX	H	
0757 13	0143	INX	D	
0758 0D	0144	DCR	C	
0759 C2 51 07	0145	JNZ	SMPLP	
075C C9	0146	RET	.	
075D	0147	‡		
075D	0148	‡	SUBTRACT JTH ROW FROM SUBSEQUENT ROWS	
075D	0149	‡		

0750 3A 97 24	0150 SUBROWJ LDA J FOR K=J+1 TO N
0760 3C	0151 INR A
0761 32 98 24	0152 SUBK1 STA K
0764 67	0153 MOV H,A LOAD Z(K,J)
0765 3A 97 24	0154 LOA J
0768 6F	0155 MOV L,A
0769 CD 07 07	0156 CALL LOOZ
076C 3A 97 24	0157 LDA J DIVIOE BY Z(J,J)
076F 67	0158 MOV H,A
0770 6F	0159 MOV L,A
0771 CD E3 07	0160 CALL DIVZ
0774 21 9C 24	0161 LXI H,RATIO STORE IN RATIO
0777 CD 0A 03	0162 CALL STOFF
077A	0163 *
077A 3A 97 24	0164 LOA J FOR P=J TO N+1
077D 32 99 24	0165 SUBL1 STA P
0790 6F	0166 MOV L,A LOAD Z(J,P)
0781 3A 97 24	0167 LOA J
0784 67	0168 MOV H,A
0785 CD 07 07	0169 CALL LOOZ
0788 21 9C 24	0170 LXI H,RATIO MULT BY RATIO
078B CD E0 03	0171 CALL FMUL
078E 79	0172 MOV A,C CHANGE SIGN
078F EE 80	0173 XRI 80H
0791 4F	0174 MOV C,A
0792	0175 *
0792 3A 98 24	0176 LDA K ADD TO Z(K,P)
0795 67	0177 MOV H,A
0796 3A 99 24	0178 LOA P
0799 6F	0179 MOV L,A
079A E5	0180 PUSH H
079B CD E9 07	0181 CALL AOZ
079E E1	0182 POP H REPLACE Z(K,P)
079F CD DD 07	0183 CALL STRZ
07A2	0184 *
07A2 3A 99 24	0185 LOA P P=P+1
07A5 3C	0186 INR A
07A6 21 91 24	0187 LXI H,NP1 IF P<N+1, OK
07A9 BE	0188 CMP M
07AA 0A 70 07	0189 JC SUBL1
07AD CA 70 07	0190 JZ SUBL1
07B0	0191 *
07B0 3A 98 24	0192 LOA K K=K+1
07B3 3C	0193 INR A
07B4 21 91 24	0194 LXI H,NP1 UNTIL K=N+1
07B7 BE	0195 CMP M
07B8 C2 61 07	0196 JNZ SUBK1
07BB C9	0197 RET
07BC	0198 *
07BC 2A 94 24	0199 REFYM2 LHLD RESULT

07BF 3A 9A 24	0200	LDA	M2	
07C2 3D	0201	DCR	A	
07C3 47	0202	MOV	B,A	
07C4 87	0203	ADD	A	
07C5 80	0204	ADD	B	
07C6 DF	0205	RST	REF/B	
07C7 C9	0206	RET		
07C8	0207	*		
07C8	0208	*	MATRIX UTILITIES	
07C8	0209	*	H=ROW, L=COL	
07C8	0210	*	IN A 4X5 ARRAY	
07C8	0211	*		
07C8 25	0212 REFZ	DCR	H	
07C9 2D	0213	DCR	L	
07CA 7C	0214	MOV	A,H	ROW#5 ACROSS
07CB 87	0215	ADD	A	
07CC 87	0216	ADD	A	
07CD 84	0217	ADD	H	
07CE 85	0218	ADD	L	+COL NUMBER (0 THRU 4)
07CF 6F	0219	MOV	L,A	TIMES FLT (3)
07D0 87	0220	ADD	A	
07D1 85	0221	ADD	L	
07D2 2A 92 24	0222	LHLD	MATRX	GET ADDRESS OF ARRAY Z
07D5 DF	0223	RST	REF/B	
07D6 C9	0224	RET	.	
07D7	0225	*		
07D7 CD CB 07	0226 LODZ	CALL	REFZ	
07DA C3 D4 03	0227	JMP	LODFP	
07DD CD CB 07	0228 STRZ	CALL	REFZ	
07E0 C3 DA 03	0229	JMP	STOFF	
07E3 CD CB 07	0230 DIVZ	CALL	REFZ	
07E6 C3 0B 04	0231	JMP	FDIV	
07E9 CD CB 07	0232 ADZ	CALL	REFZ	
07EC C3 9F 04	0233	JMP	FADD	
07EF CD D7 07	0234 ISZRO	CALL	LODZ	RETURN(Z) IF REALLY ZERO
07F2 7A	0235	MOV	A,D	
07F3 87	0236	ORA	A	
07F4 CB	0237	RZ	.	
07F5 79	0238	MOV	A,C	OR IF UNDER 1/2**10
07F6 E6 7F	0239	ANI	7FH	
07FB FE 37	0240	CPI	41H-10	
07FA D0	0241	RNC	.	
07FB 97	0242	SUB	A	
07FC C9	0243	RET		
07FD	0244 NEXT	EQU	*	
07FD		COM	NEXT	
07FD	0245	*		
07FD	0246	*	VARIABLES	
07FD	0247	*		
07FD	0248	ORG	MATRAM	

2490	0249 N	DS	1	SIZE OF ARRAY TO SOLVE
2491	0250 NP1	DS	1	SIZE+1
2492	0251 MATRX	DS	2	MATRIX STARTING ADDRESS
2494	0252 RESULT	DS	2	RESULT ADDRESS
2496	0253 Q	DS	1	TEMPORARIES
2497	0254 J	DS	1	
2498	0255 K	DS	1	
2499	0256 P	DS	1	
249A	0257 M2	DS	1	
249B	0258 KLM	DS	1	
249C	0259 RATIO	DS	4	

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE---TRIG FUNCTION
0000          0003 * FILE : TRIG.A
0000          0004 *
0000          0005 NPOINTS EQU 32
0000          0006 FLT EQU 3
0000          0007 *
0000          0008 TRIG EQU NEXT
0000          0009 ORG TRIG
07FD          0010 COM SIN
07FD          0011 COM COS
07FD          0012 COM SINSQ
07FD          0013 COM COSSQ
07FD          0014 COM SNCS
07FD          0015 *
07FD 21 91 08 0016 SNCS LX1 H,SNCS REFER TO SIN*COS
0800 FE 33      0017 CPI SNCSX-SNCS IF IN TABLE OK
0802 DA 07 08 0018 JC REFOK
0805 D6 30      0019 SUI NPOINTS/2*FLT ELSE BEGIN AT
0807 DF          0020 REFOK RST REF/B
0808 C9          0021 RET
0809            0022 *
0809            0023 * SIN SQUARED/COSINE SQUARED
0809            0024 *
0809 CD 16 08 0025 SINSQ CALL SIN REFER TO SINE
080C C3 12 08 0026 JMP ADOF
080F CD 1D 08 0027 COSSQ CALL COS REFER TO COSINE
0812 3E 33      0028 ADOF MVI A,COS2T-COSTAB ADO OFFSET
0814 DF          0029 RST REF/B
0815 C9          0030 RET
0816            0031 *
0816            0032 * SIN AND COS ROUTINES.
0816            0033 * ON ENTRY: A=0 THRU 31 TIMES FLOAT
0816            0034 * ON EXIT : HL-> SIN OR COSINE F.P. VALUE
0816            0035 *
0816 D6 18      0036 SIN SUI NPOINTS/4*FLT REMOVE 1/4 PHASE
0818 D2 1D 08 0037 JNC COS AND USE COSINE ROUTINE
081B C6 60      0038 ADI NPOINTS*FLT MOVE TO END OF PERIOD
081D            0039 *
081D 21 2B 08 0040 COS LX1 H,COSTAB REFERENCE COSINE TABLE
0820 FE 33      0041 CPI COS2T-COSTX IF IN TABLE, OK
0822 DA 29 08 0042 JC REFCOS
0825 2F          0043 CMA ELSE NPOINTS-INDEX
0826 3C          0044 INR A
0827 C6 60      0045 ADI NPOINTS*FLT
0829 DF          0046 REFCOS RST REF/B
082A C9          0047 RET
082B            0048 *
082B            0049 * SIN/COSINE TABLES
  
```

082B	0050	*		
082B	0051	COSTAR EQU	*	
082B 41	0052	COSTX DB	041H	COSINE
082C 80	0053	DB	080H	
082D 00	0054	DB	000H	
082E 40	0055	DB	040H	
082F FB	0056	DB	0FBH	
0830 15	0057	DB	015H	
0831 40	0058	DB	040H	
0832 EC	0059	DB	0ECH	
0833 83	0060	DB	083H	
0834 40	0061	DB	040H	
0835 D4	0062	DB	0D4H	
0836 DB	0063	DB	0DBH	
0837 40	0064	DB	040H	
0838 B5	0065	DB	0B5H	
0839 04	0066	DB	004H	
083A 40	0067	DB	040H	
083B BE	0068	DB	0BEH	
083C 3A	0069	DB	03AH	
083D 3F	0070	DB	03FH	
083E C3	0071	DB	0C3H	
083F EE	0072	DB	0EEH	
0840 3E	0073	DB	03EH	
0841 C7	0074	DB	0C7H	
0842 C5	0075	DB	0C5H	
0843 00	0076	DB	000H	
0844 00	0077	DB	000H	
0845 00	0078	DB	000H	
0846 BE	0079	DB	0BEH	
0847 C7	0080	DB	0C7H	
0848 C5	0081	DB	0C5H	
0849 BF	0082	DB	0BFH	
084A C3	0083	DB	0C3H	
084B EE	0084	DB	0EEH	
084C C0	0085	DB	0C0H	
084D BE	0086	DB	0BEH	
084E 3A	0087	DB	03AH	
084F C0	0088	DB	0C0H	
0850 B5	0089	DB	0B5H	
0851 04	0090	DB	004H	
0852 C0	0091	DB	0C0H	
0853 D4	0092	DB	0D4H	
0854 DB	0093	DB	0DBH	
0855 C0	0094	DB	0C0H	
0856 EC	0095	DB	0ECH	
0857 83	0096	DB	083H	
0858 C0	0097	DB	0C0H	
0859 FB	0098	DB	0FBH	
085A 15	0099	DB	015H	

085B C1	0100	DB	0C1H	
085C B0	0101	DB	0B0H	
085D 00	0102	DB	000H	
085E 41	0103 COS2T	DB	041H	COS SQUARED
085F 90	0104	DB	0B0H	
0860 00	0105	DB	000H	
0861 40	0106	DB	040H	
0862 F6	0107	DB	0F6H	
0863 43	0108	DB	043H	
0864 40	0109	DB	040H	
0865 DA	0110	DB	0DAH	
0866 B3	0111	DB	0B3H	
0867 40	0112	DB	040H	
0868 B0	0113	DB	0B0H	
0869 FC	0114	DB	0FCH	
086A 40	0115	DB	040H	
086B B0	0116	DB	0B0H	
086C 00	0117	DB	000H	
086D 3F	0118	DB	03FH	
086E 9E	0119	DB	09EH	
086F 0B	0120	DB	00BH	
0870 3E	0121	DB	03EH	
0871 95	0122	DB	095H	
0872 F6	0123	DB	0F6H	
0873 3C	0124	DB	03CH	
0874 9B	0125	DB	09BH	
0875 E5	0126	DB	0E5H	
0876 00	0127	DB	000H	
0877 00	0128	DB	000H	
0878 00	0129	DB	000H	
0879 3C	0130	DB	03CH	
087A 9B	0131	DB	09BH	
087B E5	0132	DB	0E5H	
087C 3E	0133	DB	03EH	
087D 95	0134	DB	095H	
087E F6	0135	DB	0F6H	
087F 3F	0136	DB	03FH	
0880 9E	0137	DB	09EH	
0881 0B	0138	DB	00BH	
0882 40	0139	DB	040H	
0883 B0	0140	DB	0B0H	
0884 00	0141	DB	000H	
0885 40	0142	DB	040H	
0886 B0	0143	DB	0B0H	
0887 FC	0144	DB	0FCH	
0888 40	0145	DB	040H	
0889 DA	0146	DB	0DAH	
088A B3	0147	DB	0B3H	
088B 40	0148	DB	040H	
088C F6	0149	DB	0F6H	

088D 43	0150	DB	043H	
088E 41	0151	DB	041H	
088F 80	0152	DB	080H	
0890 00	0153	DB	000H	
0891 00	0154	SNCST DB	000H	SIN*COS
0892 00	0155	DB	000H	
0893 00	0156	DB	000H	
0894 3E	0157	DB	03EH	
0895 C3	0158	DB	0C3H	
0896 EF	0159	DB	0EFH	
0897 3F	0160	DB	03FH	
0898 B5	0161	DB	0B5H	
0899 04	0162	DB	004H	
089A 3F	0163	DB	03FH	
089B EC	0164	DB	0ECH	
089C B3	0165	DB	0B3H	
089D 40	0166	DB	040H	
089E 80	0167	DB	080H	
089F 00	0168	DB	000H	
08A0 3F	0169	DB	03FH	
08A1 EC	0170	DB	0ECH	
08A2 B3	0171	DB	0B3H	
08A3 3F	0172	DB	03FH	
08A4 B5	0173	DB	0B5H	
08A5 04	0174	DB	004H	
08A6 3E	0175	DB	03EH	
08A7 C3	0176	DB	0C3H	
08A8 EF	0177	DB	0EFH	
08A9 00	0178	DB	000H	
08AA 00	0179	DB	000H	
08AB 00	0180	DB	000H	
08AC BE	0181	DB	0BEH	
08AD C3	0182	DB	0C3H	
08AE EF	0183	DB	0EFH	
08AF BF	0184	DB	0BFH	
08B0 B5	0185	DB	0B5H	
08B1 04	0186	DB	004H	
08B2 BF	0187	DB	0BFH	
08B3 EC	0188	DB	0ECH	
08B4 B3	0189	DB	0B3H	
08B5 C0	0190	DB	0C0H	
08B6 80	0191	DB	080H	
08B7 00	0192	DB	000H	
08B8 BF	0193	DB	0BFH	
08B9 EC	0194	DB	0ECH	
08BA B3	0195	DB	0B3H	
08BB BF	0196	DB	0BFH	
08BC B5	0197	DB	0B5H	
08BD 04	0198	DB	004H	
08BE BE	0199	DB	0BEH	

08BF C3	0200	DB	0C3H
09C0 EF	0201	DB	0EFH
08C1 00	0202	DB	000H
08C2 00	0203	DB	000H
08C3 00	0204	DB	000H
09C4	0205 SNCSX	EQU	\$
08C4	0206 NEXT	EQU	\$
08C4	0207	COM	NEXT

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE--- SPIN FITTING ALGORITHM
0000          0003 * FILE : SPIN.A
0000          0004 *
0000          0005 FLT EQU 3 FLOATING POINT LENGTH
0000          0006 NULL EQU 40H NULL FLOAT INDICATOR
0000          0007 NPOINTS EQU 32 NUMBER OF POINTS IN FIT
0000          0008 MINN EQU 3 MINIMUM N OR MHI
0000          0009 AVPTS EQU 4 # POINTS TO RETAIN AHI/ALO
0000          0010 HIGAIN EQU 10H HIGH GAIN SAMPLE
0000          0011 REJBIT EQU 20H REJECTED BIT
0000          0012 *
0000          0013 SAINF EQU 0#FLT INPUT PARAMETER BLOCK
0000          0014 ALPHA EQU 1#FLT
0000          0015 BETA EQU 2#FLT
0000          0016 ATABL EQU 3#FLT AHI/ALO TABLE
0000          0017 ATEND EQU AVPTS#2#FLT+ATABL
0000          0018 PSW EQU 5
0000          0019 *
0000          0020 * SPIN FIT SUBROUTINE.
0000          0021 * ON ENTRY: [HL]-> SAMPLED DATA BLOCK
0000          0022 * [DE]-> PARAM BLOCK
0000          0023 * [BC]-> WHERE TO PUT RESULTS
0000          0024 *
0000          0025 SPIN EQU NEXT
0000          0026 COM SPIN
0000          0027 ORG SPIN
0000          0028 SHLD SAMPTR SAVE DATA IN ADDR
0000          0029 XCHG .
0000          0030 SHLD PRMPTR SAVE PARAMS ADDR
0000          0031 PUSH B STACK RESULTS ADDR
0000          0032 MVI A,0C3H PUT JUMP INTO FUNCTION
0000          0033 STA FN
0000          0034 *
0000          0035 CALL PHAS1 DO 4X4 SOLUTIONS
0000          0036 CALL AVERAGE AVERAGE AHI AND ALG
0000          0037 CALL PHAS2 THEN DO FITS W/ B AND C
0000          0038 *
0000          0039 EXIT POP D MOVE LOCAL RESULTS INTO
0000          0040 LXI H,AHI THE DESTINATION AREA
0000          0041 MVI C,FLT#5+1 (AHI,LO,B,C,SIGMA,N)
0000          0042 JMP COPY
0000          0043 *
0000          0044 * 1ST AND 2ND PHASES.
0000          0045 *
0000          0046 PHAS1 MVI A,4 INIT FOR 4X4 MATRIX
0000          0047 LXI D,AHI RESULTS TO AHI THRU C
0000          0048 CALL IPHASE
0000          0049 CALL GEN44 GENERATE SUMS FOR 4X4

```

```

08EE CD AE 09      0050 PH1  CALL  SOL44  SOLVE THE 4X4
08F1 CD 85 08      0051      CALL  CALCS0  CALCULATE STD DEVIATION
08F4 CD E1 09      0052      CALL  REJ44  REJECT ALL POINTS OFF CURVE
08F7 C2 EE 08      0053      JNZ   PH1    REPEAT UNTIL NO GAREAGE
08FA C9             0054      RET    .
08FB               0055 *
08FB 3E 02         0056 PHAS2 MVI  A,2    INIT FOR SMALLER MATRIX
08FD 11 3B 23     0057      LXI  D,8COMP  RESULTS TO 8 AND C
0900 CD 13 09     0058      CALL  IPHASE
0903 CD 62 09     0059      CALL  GEN22
0906 CD 03 09     0060 PH2  CALL  SOL22  SOLVE SMALLER MATRIX
0909 CD 85 08     0061      CALL  CALCS0  COMPUTE SIGMA
090C CD ED 09     0062      CALL  REJ22  REJECT POINTS
090F C2 06 09     0063      JNZ   PH2
0912 C9             0064      RET    .
0913               0065 *
0913               0066 * INIT A PHASE.
0913               0067 *
0913 21 4B 24     0068 IPHASE LXI  H,SCRATCH  TELL SOLVER WHERE
0916 CD 7E 06     0069      CALL  IMATX  TO FIND MATRIX COPY
0919 3E 03         0070      MVI  A,ALPHA  APJB <- ALPHA
091B CD DB 0C     0071      CALL  REFP
091E 11 46 23     0072      LXI  D,APJB
0921 C3 86 0C     0073      JMP  FMOV
0924               0074 *
0924               0075 * GENERATE THE MATRIX TO BEGIN
0924               0076 *
0924 CD 6D 0A     0077 GEN44 CALL  CLRMAX  ZERO THE MATRIX (A=0)
0927 32 44 23     0078      STA  N      # POINTS=0
092A 32 45 23     0079      STA  MH1   # HI GAIN POINTS=0
092D 32 2D 23     0080      STA  ADDSB  SET ADD MODE FOR SUMS
0930               0081 *
0930 21 39 09     0082      LXI  H,GENM1  REPEAT GENM1 FOR EACH POINT
0933 CD 10 0D     0083      CALL  DOLOOP
0936 C3 0A 08     0084      JMP  Q34    THEN PRODUCE COPIES
0939               0085 *
0939 CD C1 0C     0086 GENM1 CALL  REFSAM  [HL]->SAMDTA(I)
093C 5E             0087      MOV  E,M    [DE]=SAMPLE
093D 23             0088      INX  H
093E 56             0089      MOV  D,M
093F E5             0090      PUSH H
0940 CD 9F 0C     0091      CALL  FLT12  FLOAT 12 BIT FORMAT
0943 E1             0092      POP  H
0944 7E             0093      MOV  A,M    CHECK INDICATOR
0945 E6 10         0094      ANI  HIGAIN  IF HIGH GAIN, DO IT
0947 C2 53 09     0095      JNZ  MULGN
094A               0096 *
094A CD D0 0C     0097      CALL  REFET  ET(I)=CDE
094D CD DA 03     0098      CALL  STOPP
0950 C3 E9 0A     0099      JMP  Q2L01  DO QUAD 2 LO AND Q1

```

0953		0100 †		
0953 2A 22 23		0101 MULGN LHL D	PRMPTR	MULTIPLY BY GAIN FACTOR
0956 CD E0 03		0102	CALL	FMUL TO ADJUST FOR GAIN
0959 CD D0 0C		0103	CALL	REFET STORE IN ET(1)
095C CD DA 03		0104	CALL	STDFP
095F C3 C2 0A		0105	JMP	Q2H11 DO QUAD2 H1 AND Q1
0962		0106 †		
0962		0107 †	GENERATE THE 2X2 MATRIX	
0962		0108 †		
0962 3E 80		0109 GEN22 MVI	A,80H	SUBTRACT MODE
0964 32 2D 23		0110	STA	ADDSB
0967 01 18 24		0111	LXI	B,ETCOS ETCOS=ETCOS-AMI*ECSH1
096A 11 3F 24		0112	LXI	D,ECSH1
096D 21 35 23		0113	LXI	H,AMI
0970 CD 94 09		0114	CALL	REMOVE
0973 01 18 24		0115	LXI	B,ETCOS -ALD*ECSLO
0976 11 30 24		0116	LXI	D,ECSLO
0979 21 38 23		0117	LXI	H,ALO
097C CD 94 09		0118	CALL	REMOVE
097F		0119 †		
097F 01 27 24		0120	LXI	B,ETSIN ETSIN=ETSIN-AMI*ESNH1
0982 11 42 24		0121	LXI	D,ESNH1
0985 21 35 23		0122	LXI	H,AMI
0988 CD 94 09		0123	CALL	REMOVE
098B 01 27 24		0124	LXI	B,ETSIN -ALD*ESNLO
098E 11 33 24		0125	LXI	D,ESNLO
0991 21 38 23		0126	LXI	H,ALO
0994		0127 †		
0994 C5		0128 REMOVE	PUSH	B SAVE DEST
0995 D5		0129	PUSH	D SAVE MULTIPLIER
0996 CD D4 03		0130	CALL	LODFP
0999 7A		0131	MOV	A,D IF MULTIPLIER NULL
099A B7		0132	ORA	A OR ZERO, QUIT
099B CA AB 09		0133	JZ	UNSTK
099E FE 40		0134	CP1	NULL
09A0 CA AB 09		0135	JZ	UNSTK
09A3 E1		0136	POP	H
09A4 CD E0 03		0137	CALL	FMUL MULT
09A7 E1		0138	POP	H
09AB C3 7E 0C		0139	JMP	SUM AND SUBTRACT
09AB E1		0140 UNSTK	POP	H
09AC E1		0141	POP	H
09AD C9		0142	RET	.
09AE		0143 †		
09AE		0144 †	SOLVE 4X4 AND 2X2 MATRICES.	
09AE		0145 †		
09AE 11 48 24		0146 SOL44 LX1	D,SCRATCH	SOLVE 4X4
09B1 21 0C 24		0147	LXI	H,MAT
09B4 0E 3C		0148	MVI	C,FLT#4#5
09B6 CD DB 09		0149	CALL	SLVNN

09B9		0150	*
09B9 3A 45 23		0151	LDA MHI IF NO HI GAIN POINTS
09BC B7		0152	ORA A
09BD C2 C5 09		0153	JNZ CHKLO
09C0 3E 40		0154	MVI A,NULL THEN NULL AHI
09C2 32 36 23		0155	STA AHI+1
09C5		0156	*
09C5 3A 44 23		0157	CHKLO LDA N IF NO LO GAIN POINTS
09CB 21 45 23		0158	LXI H,MHI (N-MHI=0)
09CB 96		0159	SUB M
09CC C0		0160	RNZ .
09CD 3E 40		0161	MVI A,NULL THEN NULL ALO
09CF 32 39 23		0162	STA ALO+1
09D2 C9		0163	RET .
09D3		0164	*
09D3 11 48 24		0165	SOL22 LXI D,SCRATCH COPY SMALLER PART
09D6 21 12 24		0166	LXI H,ECOS2 OF THE MATRIX
09D9 0E 1B		0167	MVI C,ETSIN-ECOS2+FLT
09DB CD BB 0C		0168	SLVNN CALL COPY
09DE C3 BD 06		0169	JMP SOLVE
09E1		0170	*
09E1		0171	* REJECT ALL POINTS OVER DISC AWAY FROM THE CURVE
09E1		0172	*
09E1 21 1D 0A		0173	REJ44 LXI H,RJ4 USE RJ4 REJECT FUNCTION
09E4 CD 0F 0A		0174	CALL REJN
09E7 CD 0A 0B		0175	CALL Q34 UPDATE THE REST FOR 4X4
09EA C3 F3 09		0176	JMP RJFIN INCREASE APJB, RETURN POINTS
09ED		0177	*
09ED 21 29 0A		0178	REJ22 LXI H,RJ2 REJECT POINT FOR 2X2
09F0 CD 0F 0A		0179	CALL REJN
09F3 21 46 23		0180	RJFIN LXI H,APJB APJB = APJB+RETA. (INCREASE
09F6 CD D4 03		0181	CALL LODFP THE DIFFICULTY FACTOR)
09F9 3E 06		0182	MVI A,BETA
09FB CD DB 0C		0183	CALL REFP
09FE CD 9F 04		0184	CALL FADD
0A01 21 46 23		0185	LXI H,APJB
0A04 CD DA 03		0186	CALL STOPP
0A07		0187	*
0A07 3A 2E 23		0188	LOA NP RETURN # POINTS REJECTED
0A0A 21 44 23		0189	LXI H,N
0A0D 96		0190	SUB M
0A0E C9		0191	RET .
0A0F		0192	*
0A0F 3A 44 23		0193	REJN LOA N RECORD # POINTS NOW
0A12 32 2E 23		0194	STA NP
0A15 3E B0		0195	MVI A,BOH SET SUBTRACT MODE
0A17 32 2D 23		0196	STA ADOSB
0A1A C3 10 0D		0197	JMP DOLOOP EXECUTE REJECT FUNCTION
0A1D		0198	*
0A1D CD 33 0A		0199	RJ4 CALL RJTEST TEST FOR NEW REJECTION

0A20 D0	0200	RNC	.	IF OLD OR NO REJECT, QUIT
0A21 E6 10	0201	ANI	HIGAIN	IF NEW REJ, UPDATE SUMS IN Q2 AND Q1
0A23 CA E9 0A	0202	JZ	Q2L01	
0A26 C3 C2 0A	0203	JMP	Q2H11	
0A29	0204	*		
0A29 CD 33 0A	0205	RJ2	CALL	RJTEST TEST FOR REJECTION
0A2C D0	0206	RNC	.	OLD OR NO REJECT
0A2D CD 4D 0B	0207	CALL	Q1X22	UPDATE SUMS IN Q1 FOR 2X2
0A30 C3 7D 0A	0208	JMP	Q1	AND FINISH WITH NORMAL Q1
0A33	0209	*		
0A33	0210	*		RETURNS NO CARRY IF NO CHANGE
0A33	0211	*		CARRY IF NEWLY REJECTED
0A33	0212	*		
0A33 CD C1 0C	0213	RJTEST	CALL	REFSAM IF REJECTED ALREADY,
0A36 23	0214	INX	H	
0A37 7E	0215	MOV	A,M	THEN RETURN NO CHANGE
0A38 FE 20	0216	CPI	REJBIT	
0A3A D0	0217	RNC	.	
0A3B	0218	*		
0A3B E6 10	0219	ANI	HIGAIN	REJECT IF TOO FEW POINTS
0A3D 21 45 23	0220	LXI	H,MHI	
0A40 7E	0221	MOV	A,M	IN THIS GAIN.
0A41 C2 48 0A	0222	JNZ	CM1	
0A44 3A 44 23	0223	LDA	N	(N-MHI)<MINN
0A47 96	0224	SUB	M	
0A48 FE 03	0225	CM1	CPI	MINN
0A4A DA 63 0A	0226	JC	MRKREJ	
0A4D	0227	*		
0A4D CD CA 0C	0228	CALL	REFDIF	IF ABS(DIFF(I))<DISC
0A50 CD D4 03	0229	CALL	LODFP	THEN RETURN(NO CHANGE)
0A53 7A	0230	MOV	A,D	(IF DIFF=0, RETURN(NC))
0A54 87	0231	ORA	A	
0A55 C8	0232	RZ	.	
0A56 79	0233	MOV	A,C	TAKE ABSOLUTE VALUE
0A57 E6 7F	0234	ANI	7FH	
0A59 4F	0235	MOV	C,A	
0A5A 21 49 23	0236	LXI	H,DISC	COMPARE EXPONENTS
0A5D 8E	0237	CMP	M	IF EQUAL EXPS, THEN CALL
0A5E CC 89 04	0238	CZ	FCMP	COMPARE TO SET FLAGS
0A61 3F	0239	CMC	.	RETURN(NC) IF DIFF < DISC
0A62 D0	0240	RNC	.	
0A63	0241	*		
0A63 CD C1 0C	0242	MRKREJ	CALL	REFSAM MARK POINT REJECTED
0A66 23	0243	INX	H	
0A67 7E	0244	MOV	A,M	
0A68 F6 20	0245	ORI	REJBIT	
0A6A 77	0246	MOV	M,A	
0A6B 37	0247	STC	.	RETURN (CHANGED)
0A6C C9	0248	RET	.	
0A6D	0249	*		

0A6D 21 0D 24	0250 CLRMAX LXI H,MAT+1 ZERO BYTE 2
0A70 0E 14	0251 MVI C,4*5 OF EACH IN MATRIX
0A72 11 03 00	0252 LXI D,FLT
0A75 97	0253 SUB A
0A76 77	0254 CLR1 MOV M,A
0A77 19	0255 DAD D
0A78 0D	0256 DCR C
0A79 C2 76 0A	0257 JNZ CLR1
0A7C C9	0258 RET .
0A7D	0259 *
0A7D	0260 * QUAD 1
0A7D	0261 *
0A7D 21 44 23	0262 Q1 LXI H,N N = N (+/-) 1
0A80 CD 88 0C	0263 CALL COUNT
0A83 CD F5 0C	0264 CALL COSISQ SUM COS**2
0A86 21 12 24	0265 LXI H,ECS2
0A89 CD 7E 0C	0266 CALL SUM
0A8C CD 07 0D	0267 CALL SINCOS SUM SINCOS
0A8F 21 15 24	0268 LXI H,ESNCS
0A92 CD 7E 0C	0269 CALL SUM
0A95 21 21 24	0270 LXI H,ECSN SUM COSSIN
0A98 CD DA 03	0271 CALL STOFF
0A98 CD FE 0C	0272 CALL SINESQ SUM SIN**2
0A9E 21 24 24	0273 LXI H,ESIN2
0AA1 CD 7E 0C	0274 CALL SUM
0AA4	0275 *
0AA4 CD E3 0C	0276 CALL COSINE SUM COS*ET(I)
0AA7 CD D0 0C	0277 CALL REFET
0AAA CD E0 03	0278 CALL FMUL
0AAD 21 18 24	0279 LXI H,ETCOS
0AB0 CD 7E 0C	0280 CALL SUM
0AB3	0281 *
0AB3 CD EC 0C	0282 CALL SINE SUM SIN*ET(I)
0AB6 CD D0 0C	0283 CALL REFET
0AB9 CD E0 03	0284 CALL FMUL
0ABC 21 27 24	0285 LXI H,ETSIN
0ABF C3 7E 0C	0286 JMP SUM
0AC2	0287 *
0AC2	0288 * QUAD 2. SUM OF COSINES AND SINES
0AC2	0289 *
0AC2 21 45 23	0290 Q2HI1 LXI H,MHI KEEP TRACK OF HIGH POINTS
0AC5 CD 88 0C	0291 CALL COUNT M=M+1 OR -1
0AC8 CD E3 0C	0292 CALL COSINE ECSHI=ECSHI(+/-)COS
0AC8 21 3F 24	0293 LXI H,ECSHI
0ACE CD 7E 0C	0294 CALL SUM
0AD1 CD EC 0C	0295 CALL SINE ESNHI=ESNHI(+/-)SIN
0AD4 21 42 24	0296 LXI H,ESNHI
0AD7 CD 7E 0C	0297 CALL SUM
0ADA CD D0 0C	0298 CALL REFET EHI=EHI(+/-)ET(I)
0ADD CD D4 03	0299 CALL LODFP

0AE0 21 45 24	0300	LXI	H,EHI
0AE3 CD 7E 0C	0301	CALL	SUM
0AE6 C3 7D 0A	0302	JMP	Q1
0AE9	0303	†	
0AE9 CD E3 0C	0304 Q2L01	CALL	COSINE SAME AS ABOVE FOR LO GAIN
0AEC 21 30 24	0305	LXI	H,ECSLO
0AEF CD 7E 0C	0306	CALL	SUM
0AF2 CD EC 0C	0307	CALL	SINE
0AF5 21 33 24	0308	LXI	H,ESNLO
0AF8 CD 7E 0C	0309	CALL	SUM
0AFB	0310	†	
0AFB CD D0 0C	0311	CALL	REFET
0AFE CD D4 03	0312	CALL	LODFP
0B01 21 36 24	0313	LXI	H,ELO
0B04 CD 7E 0C	0314	CALL	SUM
0B07 C3 7D 0A	0315	JMP	Q1
0B0A	0316	†	
0B0A	0317	†	QUADS 3 AND 4. COPY VALUES FROM QUAD 2
0B0A	0318	†	
0B0A 21 3F 24	0319 Q34	LXI	H,ECSHI ECH=ECSHI
0B0D 11 0C 24	0320	LXI	D,ECH
0B10 CD B6 0C	0321	CALL	FMOV
0B13 11 1B 24	0322	LXI	D,ESH ESH=ESNHI
0B16 CD B6 0C	0323	CALL	FMOV
0B19	0324	†	
0B19 21 30 24	0325	LXI	H,ECSLO ECL=ECSLO
0B1C 11 0F 24	0326	LXI	D,ECL
0B1F CD B6 0C	0327	CALL	FMOV
0B22 11 1E 24	0328	LXI	D,ESL
0B25 CD B6 0C	0329	CALL	FMOV
0B28	0330	†	
0B28	0331	†	QUAD 4.
0B28	0332	†	
0B28 3A 45 23	0333	LDA	MHI FMHI=FLOAT(M)
0B28 B7	0334	ORA	A IF M=0 THEN SET MHI=1
0B2C C2 30 0B	0335	JNZ	Q4A
0B2F 3C	0336	INR	A
0B30 CD 96 0C	0337 Q4A	CALL	FLT8
0B33 21 39 24	0338	LXI	H,FMHI
0B36 CD DA 03	0339	CALL	STOFF
0B39	0340	†	
0B39 3A 44 23	0341	LDA	N NLM=FLOAT(N-M)
0B3C 21 45 23	0342	LXI	H,MHI
0B3F 96	0343	SUB	M
0B40 C2 44 0B	0344	JNZ	Q4B
0B43 3C	0345	INR	A
0B44 CD 96 0C	0346 Q4B	CALL	FLT8
0B47 21 2D 24	0347	LXI	H,NLM
0B4A C3 DA 03	0348	JMP	STOFF
0B4D	0349	†	

0B4D	0350	*	REMOVE SUMS FOR 2X2 FROM Q1 ETSIN AND ETCOS
0B4D	0351	*	
0B4D 21 36 23	0352	QIX2 LXI	H,AHI+I IF HIGH GAIN
0B50 E6 10	0353	ANI	HIGAIN
0B52 C2 5B 0B	0354	JNZ	QIXI
0B55 21 39 23	0355	LXI	H,ALO+1 IF LOW GAIN
0B5B	0356	*	
0B5B 7E	0357	QIXI MOV	A,M CHECK IF AX NULL
0B59 B7	035B	ORA	A IF NULL OR ZERO, QUIT
0B5A CB	0359	RZ	.
0B5B FE 40	0360	CPI	NULL
0B5D C9	0361	RZ	.
0B5E 2B	0362	DCX	H
0B5F	0363	*	
0B5F E5	0364	PUSH	H
0B60 CD D4 03	0365	CALL	LODFP ETCOS=ETCOS-A(GAIN)*COS(I)
0B63 3A 2B 23	0366	LDA	INDEX (HL->COS(INDEX))
0B66 CD 1D 0B	0367	CALL	COS
0B69 CD E0 03	036B	CALL	FMUL
0B6C 21 18 24	0369	LXI	H,ETCOS
0B6F CD 7E 0C	0370	CALL	SUM
0B72	0371	*	
0B72 E1	0372	POP	H
0B73 CD D4 03	0373	CALL	LODFP ETSIN=ETSIN-A(GAIN)*SIN(I)
0B76 3A 2B 23	0374	LDA	INDEX
0B79 CD 16 0B	0375	CALL	SIN
0B7C CD E0 03	0376	CALL	FMUL
0B7F 21 27 24	0377	LXI	H,ETSIN
0BB2 C3 7E 0C	037B	JMP	SUM
0BB5	0379	*	
0BB5	0380	*	CALCULATE THE STANDARD DEVIATION AND THE
0BB5	0381	*	DISCRIMINATION FACTOR
0BB5	0382	*	
0BB5 97	03B3	CALCSD SUB	A INIT SIGMA SUM = 0
0BB6 32 42 23	0384	STA	SIGMA+I
0BB9 3A 44 23	03B5	LDA	N FLTNI = FLOAT(N-1)
0BBC 3D	0386	DCR	A
0BBD CD 96 0C	0387	CALL	FLT8
0B90 21 2F 23	038B	LXI	H,FLTNI
0B93 CD DA 03	0389	CALL	STOFF
0B96 21 8D 0B	0390	LXI	H,CDIFF CALCULATE DIFFERENCES
0B99 CD 10 0D	0391	CALL	DOLOOP AND SUM SQUARES
0B9C 21 41 23	0392	LXI	H,SIGMA SIGMA = SIGMA/(N-1)
0B9F CD D4 03	0393	CALL	LODFP
0BA2 21 2F 23	0394	LXI	H,FLTNI
0BA5 CD 0B 04	0395	CALL	FDIV
0BA8 CD DC 05	0396	CALL	FSQRT TAKE ROOT
0BAB 21 41 23	0397	LXI	H,SIGMA THEN STORE
0BAE CD DA 03	039B	CALL	STOFF
0BB1	0399	*	

0BB1 21 46 23	0400 LXI H,APJB DISC=SIGMA*(ALPHA+J*BETA)
0BB4 CD E0 03	0401 CALL FMUL
0BB7 21 49 23	0402 LXI H,DISC
0BBA C3 DA 03	0403 JMP STOPP
0BBD	0404 *
0BBD	0405 * CALCULATE THE DIFFERENCES ARRAY
0BBD	0406 * FOR POINTS WHICH HAVE NOT BEEN REJECTED
0BBD	0407 * $DIFF(I) = A(GAIN)+B*COS(I)+C*SIN(I) - ET(I)$
0BBD	0408 * ALSO SUM SIGMA AT THE SAME TIME
0BBD	0409 *
0BRD CD C1 0C	0410 CDIFF CALL REFSAM IF POINT(I)=REJECTED,QUIT
0BC0 23	0411 INX H
0BC1 7E	0412 MOV A,M
0BC2 FE 20	0413 CPI REJBIT
0BC4 D0	0414 RNC .
0BC5 F5	0415 PUSH PSW ELSE SAVE GAIN INFO
0BC6 CD E3 0C	0416 CALL COSINE B*COSINE(I)
0BC9 21 3B 23	0417 LXI H,BCOMP
0BCC CD E0 03	0418 CALL FMUL
0BCF 21 32 23	0419 LXI H,FTEMP
0BD2 CD DA 03	0420 CALL STOPP
0BD5 CD EC 0C	0421 CALL SINE + C*SIN(I)
0BD8 21 3E 23	0422 LXI H,CCOMP
0BDB CD E0 03	0423 CALL FMUL
0BDE 21 32 23	0424 LXI H,FTEMP
0BE1 CD 9F 04	0425 CALL FADD
0BE4	0426 *
0BE4 F1	0427 POP PSW IF LO GAIN, USE ALO
0BE5 E6 10	0428 ANI HIGAIN
0BE7 21 38 23	0429 LXI H,ALO
0BEA CA F0 0B	0430 JZ ADOFF
0BED 21 35 23	0431 LXI H,AHI ELSE USE AHI
0BF0 CD 9F 04	0432 ADOFF CALL FADD ADD OFFSET
0BF3 CD D0 0C	0433 CALL REFET SUBTRACT ET(I)
0BF6 CD 98 04	0434 CALL FSUB
0BF9 CD CA 0C	0435 CALL REFDIF STORE IN DIFF(I)
0BFC CD DA 03	0436 CALL STOPP
0BFF CD D3 05	0437 CALL FSQUARE SQUARE DIFF(I)
0C02 21 41 23	0438 LXI H,SIGMA SIGMA=SIGMA+DIFF(I)**2
0C05 CD 9F 04	0439 CALL FADD
0C08 21 41 23	0440 LXI H,SIGMA
0C0B C3 DA 03	0441 JMP STOPP
0C0E	0442 *
0C0E	0443 * MAINTAIN AVERAGES OF AHI AND ALO
0C0E	0444 *
0C0E 3E 09	0445 AVERAGE MVI A,ATABL FORGET THE OLDEST
0C10 CD DB 0C	0446 CALL REFP AHI/ALO PAIR
0C13 EB	0447 XCHG . [DE]->ATABL[0]
0C14 21 06 00	0448 LXI H,FLT#2 [HL]->ATABL[2]
0C17 19	0449 DAD D

0C18 0E 12	0450	MVI	C,AVPTS-1*2*FLT
0C1A CD 8B 0C	0451	CALL	COPY
0C1D	0452 *		
0C1D 21 35 23	0453	LXI	H,AHI COPY AHI/ALO INTO TABLE BOTTOM
0C20 0E 06	0454	MVI	C,2*FLT
0C22 CD 8B 0C	0455	CALL	COPY
0C25	0456 *		
0C25 3E 09	0457	MVI	A,ATABL ADD ALL THE AHI'S
0C27 21 35 23	0458	LXI	H,AHI STORE INTO AHI RESULT
0C2A CD 32 0C	0459	CALL	AVGI
0C2D 3E 0C	0460	MVI	A,ATABL+FLT ADD ALL THE ALO'S
0C2F 21 38 23	0461	LXI	H,ALO
0C32	0462 *		
0C32 E5	0463 AVGI	PUSH	H SAVE ADDRESS OF AHI/ALO
0C33 0E 00	0464	MVI	C,0 SUM=0.0
0C35 11 00 00	0465	LXI	D,0
0C38 21 2C 23	0466	LXI	H,PTCNT
0C3B 36 00	0467	MVI	M,0
0C3D	0468 *		
0C3D F5	0469 SUNLP	PUSH	PSW REFERENCE PARAM[A]
0C3E CD 8B 0C	0470	CALL	REFF
0C41 CD 70 0C	0471	CALL	CHKADD ADD IT
0C44 F1	0472	POP	PSW STEP 2 FLTS DOWN IN TABLE
0C45 C6 06	0473	ADI	FLT*2
0C47 FE 21	0474	CPI	ATEND IF MORE IN TABLE, LOOP
0C49 DA 3D 0C	0475	JC	SUNLP
0C4C	0476 *		
0C4C CD 53 0C	0477	CALL	AVDIV
0C4F E1	0478	POP	H
0C50 C3 DA 03	0479	JMP	STOFF
0C53	0480 *		
0C53 3A 2C 23	0481 AVDIV	LDA	PTCNT IF ZERO PTS, RETURN NULL
0C56 B7	0482	ORA	A
0C57 CA 6D 0C	0483	JZ	AVGNUL
0C5A C5	0484	PUSH	B SAVE SUM
0C5B D5	0485	PUSH	D
0C5C CD 96 0C	0486	CALL	FLT8
0C5F 21 32 23	0487	LXI	H,FTEMP
0C62 CD DA 03	0488	CALL	STOFF
0C65 D1	0489	POP	D DIVIDE SUM BY COUNT
0C66 C1	0490	POP	B
0C67 21 32 23	0491	LXI	H,FTEMP
0C6A C3 0B 04	0492	JMP	FDIV
0C6D 16 40	0493 AVGNUL	MVI	D,NULL
0C6F C9	0494	RET	
0C70	0495 *		
0C70 23	0496 CHKADD	INX	H CHECK IF [HL]-> GOOD FLT
0C71 7E	0497	MOV	A,H
0C72 FE 40	0498	CPI	NULL
0C74 C8	0499	RZ	.

0C75 2B	0500	DCX	H	IF GOOD, ADD TO SUM
0C76 CD 9F 04	0501	CALL	FADD	
0C79 21 2C 23	0502	LXI	H,PTCNT	PTCNT++
0C7C 34	0503	INR	M	
0C7D C9	0504	RET		
0C7E	0505	⋮		
0C7E	0506	⋮	<< UTILITY SECTION >>	
0C7E	0507	⋮	SUM CDE INTO VALUE AT HL.	
0C7E	0508	⋮		
0C7E E5	0509	SUM	PUSH	H SAVE ADDRESS
0C7F 3A 2D 23	0510	LDA	AD08B	PUT ADD/SUB MARK
0C82 A9	0511	XRA	C	INTO CDE (INVERT SIGN FOR SUB)
0C83 4F	0512	MOV	C,A	
0C84 CD 9F 04	0513	CALL	FADD	ADD 'EM
0C87 E1	0514	POP	H	AND STORE
0C88 C3 DA 03	0515	JMP	STOFF	
0C8B	0516	⋮		
0C8B 3A 2D 23	0517	COUNT	LDA	AD08B IF ADD MODE,
0C8E B7	0518	DRA	A	THEN
0C8F FA 94 0C	0519	JM	SUBN6	
0C92 34	0520	INR	M	INCREMENT M
0C93 C9	0521	RET	.	
0C94 35	0522	SUBN6	DCR	M ELSE DECR M
0C95 C9	0523	RET	.	
0C96	0524	⋮		
0C96 6F	0525	FLT8	MOV	L,A FLOAT ACCUM
0C97 26 00	0526	MVI	H,0	
0C99 11 00 00	0527	LXI	D,0	
0C9C C3 0F 05	0528	JMP	FLT32	
0C9F	0529	⋮		
0C9F	0530	⋮	FLOAT 12-BIT 2'S COMPLEMENT IN [DE]	
0C9F	0531	⋮	VALUE RETURNED IS -1 TO 1	
0C9F	0532	⋮		
0C9F 21 00 00	0533	FLT12	LXI	H,0 LOW 16 BITS ARE 0
0CA2 7A	0534	MOV	A,D	STRIP TO 12 BITS
0CA3 E6 0F	0535	ANI	0FH	
0CA5 57	0536	MOV	D,A	
0CA6 FE 08	0537	CPI	8	IF POS THEN FLOAT NOW
0CA8 DA AE 0C	0538	JC	FLT1T	
0CAB F6 F0	0539	ORI	0F0H	ELSE EXTEND SIGN
0CAD 57	0540	MOV	D,A	AND FLOAT IT
0CAE CD 0F 05	0541	FLT1T	CALL	FLT32 CDE=FLOAT(DEHL)
0CB1 79	0542	MOV	A,C	REMOVE EXPONENT
0CB2 D6 18	0543	SUI	27	BIAS WE IMPOSED
0CB4 4F	0544	MOV	C,A	TO YIELD VALUE 0 TO 1
0CB5 C9	0545	RET	.	
0CB6	0546	⋮		
0CB6 0E 03	0547	FMOV	MVI	C,FLT MOVE 1 FLT VALUE
0CB8 7E	0548	COPY	MOV	A,M FROM [HL] TO [DE]
0CB9 12	0549	STAX	D	

OCBA 23	0550	INX	H	
OCBB 13	0551	INX	D	
OCBC 0D	0552	DCR	C	
OCBD C2 88 0C	0553	JNZ	COPY	
OCC0 C9	0554	RET	.	
OCC1	0555	‡		
OCC1	0556	‡	REFERENCE FUNCTIONS FOR ARRAYS	
OCC1	0557	‡		
OCC1 2A 20 23	0558	REFSAM	LHLD	SAMPTR ADDRESS SAMPLE(I2)
OCC4 3A 2C 23	0559	LDA		INDX2
OCC7 C3 D6 0C	0560	JMP		REF2
OCCA 21 AC 23	0561	REFDIF	LXI	H,DIFF ADDRESS DIFF(I)
OCCD C3 D3 0C	0562	JMP		REF
OCD0 21 4C 23	0563	REFET	LXI	H,ET ADDRESS ET(I)
OCD3 3A 2B 23	0564	REF	LDA	INDEX GET THE INDEX
OCD6 85	0565	REF2	ADD	L
OCD7 6F	0566	MOV		L,A
OCD8 D0	0567	RNC		
OCD9 24	0568	INR	H	
OCD A C9	0569	RET	.	
OCD8	0570	‡		
OCD8 2A 22 23	0571	REFP	LHLD	PRMPTR [HL]->PARAMETER BLOCK
OCD E 85	0572	ADD	L	REFER TO PARAM(A)
OCD F 6F	0573	MOV		L,A
OCE0 D0	0574	RNC		
OCE1 24	0575	INR	H	
OCE2 C9	0576	RET	.	
OCE3	0577	‡		
OCE3	0578	‡	LOADING TRIGS	
OCE3	0579	‡		
OCE3 3A 2B 23	0580	COSINE	LDA	INDEX
OCE6 CD 1D 08	0581	CALL		COS
OCE9 C3 D4 03	0582	JMP		LODFP
OCEC 3A 2B 23	0583	SINE	LDA	INDEX
OCE F CD 16 08	0584	CALL		SIN
OCF2 C3 D4 03	0585	JMP		LODFP
OCF5 3A 2B 23	0586	COSISQ	LDA	INDEX
OCF8 CD 0F 08	0587	CALL		COSSQ
OCF8 C3 D4 03	0588	JMP		LODFP
OCFE 3A 2B 23	0589	SINESQ	LDA	INDEX
OD01 CD 09 08	0590	CALL		SINSQ
OD04 C3 D4 03	0591	JMP		LODFP
OD07 3A 2B 23	0592	SINCOS	LDA	INDEX
OD0 A CD FD 07	0593	CALL		SNCS
OD0 D C3 D4 03	0594	JMP		LODFP
OD10	0595	‡		
OD10	0596	‡	DO-LOOP EXECUTOR.	
OD10	0597	‡		
OD10 22 29 23	0598	DOLOOP	SHLD	FN+1 SET FUNCTION ADDRESS
OD13 97	0599	SUB	A	

0D14 32 2C 23	0600	STA	INDX2	
0D17 32 2B 23	0601 DOST	STA	INDEX	
0D1A CD 2B 23	0602	CALL	FN	
0D1D 3A 2C 23	0603	LDA	INDX2	
0D20 C6 20	0604	ADI	2*16	
0D22 32 2C 23	0605	STA	INDX2	
0D25 3A 2B 23	0606	LDA	INDEX	DO THE OTHER SIDE
0D28 C6 30	0607	ADI	3*16	OF THE PERIOD NOW
0D2A 32 2B 23	0608	STA	INDEX	
0D2D CD 2B 23	0609	CALL	FN	
0D30 3A 2C 23	0610	LDA	INDX2	
0D33 D6 1E	0611	SUI	2*16-2	
0D35 32 2C 23	0612	STA	INDX2	
0D38 3A 2B 23	0613	LDA	INDEX	SUBTRACT BACK TO
0D3B D6 2D	0614	SUI	3*16-3	FRONT SIDE, ADD 3
0D3D FE 30	0615	CPI	NPOINTS*FLT/2	ONLY DO HALF
0D3F DA 17 0D	0616	JC	DOST	
0D42 C9	0617	RET	.	
0D43	0618 NEXT	EBU	\$	END OF SPIN.A
0D43	0619	COM	NEXT	
0D43	0620	↑		
0D43	0621	↑	SPIN FIT VARIABLES	
0D43	0622	↑		
0D43	0623	ORG	SPINRAM	
2320	0624	SAMPTR DS	2	SAMPLE POINTER
2322	0625	PRMPTR DS	2	PARAM BLOCK POINTER
2324	0626	AXPTR DS	2	AHI/LO POINTER TEMP
2326	0627	TBPTR DS	2	TABLE POINTER TEMP
2328	0628	FN DS	3	FUNCTION FOR DOLOOP
232B	0629	INDEX DS	1	DOLOOP INDEX FOR FLT
232C	0630	INDX2 DS	1	DOLOOP INDEX FOR SAMPLES
232D	0631	ADDSB DS	1	ADD OR SUB MODE
232E	0632	NP DS	1	TEMP FOR N
232F	0633	FLTNI DS	FLT	FLOAT VALUE OF N-1
2332	0634	FTEMP DS	FLT	TEMPORARY
2335	0635	PTCNT EQU	INDX2	TEMP COUNT FOR AVERAGE
2335	0636	↑		
2335	0637	AHI DS	FLT	
2338	0638	ALO DS	FLT	
233B	0639	BCOMP DS	FLT	B COMPONENT
233E	0640	CCOMP DS	FLT	C COMPONENT
2341	0641	SIGMA DS	FLT	
2344	0642	N DS	1	#POINTS ACTIVE
2345	0643	MHI DS	1	#HIGH GAIN ACTIVE
2346	0644	↑		
2346	0645	APJB DS	FLT	
2349	0646	DISC DS	FLT	
234C	0647	↑		
234C	0648	ET DS	NPOINTS*FLT	POINTS IN F.P.
234C	0649	DIFF DS	NPOINTS*FLT	FIT(I) - ET(I)

240C	0650 †			
240C	0651 †	THE MATRIX FOR SOLVING		
240C	0652 †			
240C	0653	MAT	EQU	‡
240C	0654	ECH	DS	FLT TOP ROW
240F	0655	ECL	DS	FLT
2412	0656	ECOS2	DS	FLT
2415	0657	ESNCS	DS	FLT
2418	0658	ETCOS	DS	FLT
241B	0659 †			
241B	0660	ESH	DS	FLT 2ND ROW
241E	0661	ESL	DS	FLT
2421	0662	ECSSN	DS	FLT
2424	0663	ESIN2	DS	FLT
2427	0664	ETSIN	DS	FLT
242A	0665 †			
242A	0666	Z2	DS	FLT 3RD ROW
242D	0667	NLM	DS	FLT
2430	0668	ECSLO	DS	FLT
2433	0669	ESNLO	DS	FLT
2436	0670	ELO	DS	FLT
2439	0671 †			
2439	0672	FMHI	DS	FLT LAST ROW
243C	0673	Z1	DS	FLT
243F	0674	ECSHI	DS	FLT
2442	0675	ESNHI	DS	FLT
2445	0676	EHI	DS	FLT
2448	0677 †			
2448	0678	SCRATCH	DS	FLT#4#5 SCRATCH AREA FOR SOLVING


```

0000          0001 *
0000          0002 * CRRES FLIGHT PROGRAM---MAGNETIC FIELD MANAGEMENT
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 *
0000          0005 * FILE : MAG.A VERSION 3
0000          0006 *
0000          0007 PSW EQU 6      8085 SPECIFIC INFORMATION
0000          0008 SP EQU 6
0000          0009 *
0000          0010 BX EQU 2      MULTIPLEXOR QTY ADDRESSES
0000          0011 BY EQU 1
0000          0012 BZ EQU 0
0000          0013 HIGAIN EQU 10H HIGH GAIN BIT IN QTY
0000          0014 BAMP EQU 7      CODE FOR B AMPLIFIER
0000          0015 BMODE EQU 0FEH MAG MODE COMMAND PREFIX
0000          0016 *
0000          0017 POSLIM EQU 4I    POSITIVE GAIN LIMIT
0000          0018 NEGLIM EQU 38    NEGATIVE GAIN LIMIT
0000          0019 *
0000          0020          ORG MAG
0050 C3 52 00          0021          JMP MAGINIT INITIALIZATION
0053 C3 96 00          0022          JMP MAGFRAME MINDR FRAME SYNC
0056 C3 A7 00          0023          JMP MAGGAIN GAIN DECISIONS
0059 C3 EA 00          0024          JMP MAGSAMP SAMPLE TIME
005C C3 F7 00          0025          JMP MAGENCD BUFFERING TIME
005F C3 85 00          0026          JMP MAGTELEM TELEMETRY TIME
0062          0027 *
0062          0028 * INITIALIZE THE B-FIELD PACKAGE
0062          0029 *
0062 21 00 FE          0030 MAGINIT LXI H,BMODE*256 BMODE(0)
0065 CD 7D 00          0031          CALL MAGCMD
0068 21 26 29          0032          LXI H,POSLIM*256+NEGLIM
006B 22 A6 24          0033          SHLD LIMITS
006E          0034 *
006E 97          0035 MAGSYNC SUB A RESET THE
006F 32 2F 21          0036          STA SMCNT SAMPLE STATE COUNTER
0072 3E F0          0037          MVI A,LGBUF*256/128-3 SET NIBBLE ADDRESS
0074 32 2C 21          0038          STA LGPTR LESS 3 NIBBLES
0077 3E 07          0039          MVI A,AGBUF*256/128-3 FOR BOTH LOW AND
0079 32 2D 21          0040          STA AGPTR AUTO GAIN POINTERS
007C C9          0041          RET .
007D          0042 *
007D          0043 * PERFORM THE BMODE COMMAND
007D          0044 * DN ENTRY: [HL] = 16-BIT COMMAND. (A=L)
007D          0045 *
007D 32 33 21          0046 MAGCMD STA MODE REMEMBER THE MODE
0080 3E 07          0047          MVI A,BAMP SET THE B AMPLIFIER ON/OFF
0082 C3 CC 02          0048          JMP SETMUX BY A I/O IN THE LSB OF L
0085          0049 *

```

```

0085      0050 * B-FIELD TELEMETRY OUTPUT ROUTINES.
0085      0051 * ON ENTRY: A= 0 FOR BYTE, 1 FOR WORD
0085      0052 * ON EXIT : [L]=BYTE VALUE, [HL]=WORD VALUE
0085      0053 *
0085 E7      0054 MAGTEL ORA    A        IF WORD REQ'D
0086 C4 8F 00 0055      CNZ    READ    THEN GET 2 ELSE 1
0089 53      0056      MOV    D,E
008A C0 8F 00 0057      CALL   READ
008D E8      0058      XCHG   .
008E C9      0059      RET    .
008F
008F 21 2E 21 0060 *
0092 34      0061 READ   LXI    H,OTPTR RETURN E=MEM[+OTPTR]
0093 6E      0062      INR    M
0094 5E      0063      MOV    L,M
0095 C9      0064      MOV    E,M
0096      0065      RET
0096      0066 *
0096      0067 * MINOR FRAME SYNC.
0096      0068 * ON ENTRY: A=FRAME NUMBER
0096      0069 *
0096 E6 03 0070 MAGFRAME ANI 3        IF FRAME 0 MOD 4
0098 CA A1 00 0071      JZ     RESOUT
009B FE 02 0072      CPI    2
009D CA 6E 00 0073      JZ     MAGSYNC
00A0 C9      0074      RET
00A1      0075 *
00A1 3E FF 0076 RESOUT MVI   A,LGBUF+256/256-1 RESET THE
00A3 32 2E 21 0077      STA   OTPTR PREINC'D OUTPUT PTR
00A6 C9      0078      RET
00A7      0079 *
00A7      0080 * MAG GAIN DECISION TIME.
00A7      0081 *
00A7 01 E7 00 0082 MAGGAIN LXI   B,GNSAMPS SAMPLE THE TRIPLET
00AA 11 37 21 0083      LXI   D,BXL
00AD C0 84 0E 0084      CALL  TRIPLET
00B0      0085 *
00B0 11 E7 00 0086      LXI   D,GNSAMPS LOW GAIN BY DEFAULT
00B3 21 34 21 0087      LXI   H,BXQTY
00B6 0E 03 0088      MVI   C,3
00B8 D7      0089      RST   COPY/8
00B9      0090 *
00B9 01 34 21 0091      LXI   B,BXQTY AND DECIDE 3 GAINS
00BC 2A 37 21 0092      LHLD  BXL
00BF C0 CB 00 0093      CALL  DECIDE
00C2 2A 39 21 0094      LHLD  BYL
00C5 C0 CB 00 0095      CALL  DECIDE
00C8 2A 3B 21 0096      LHLD  BZL
00CB      0097 *
00CB      0098 * DECIDE WHICH GAIN TO USE
00CB      0099 *

```

0DCB EB	0100 DECIDE XCHG .	PUT VALUE IN (DE)
0DCC 2A A6 24	0101 LHL D	LIMITS H=POS. L = NEG LIMIT
0DCF 7A	0102 MOV	A,D IF SMALL POSITIVE, GO
0DD0 B7	0103 ORA	A
0DD1 CA BC 0D	0104 JZ	CHECK
0DD4 D6 0F	0105 SUI	0FH IF NOT SMALL NEGATIVE, SET LOW
0DD6 C2 E5 0D	0106 JNZ	LOW
0DD9 93	0107 SUB	E INVERT VALUE
0DDA 5F	0108 MOV	E,A
0DDB 85	0109 MOV	H,L USE NEG LIMIT
0DDC	0110 *	
0DDC 7B	0111 CHECK MOV	A,E IF VAL>= LIMIT, USE LOW GAIN
0DDD BC	0112 CMP	H
0DDE D2 E5 0D	0113 JNC	LOW
0DE1 0A	0114 LDAX	B USE HIGH GAIN
0DE2 F6 10	0115 ORI	HIGAIN
0DE4 02	0116 STAX	B
0DES 03	0117 LOW INX	B
0DE6 09	0118 RET	
0DE7	0119 *	
0DE7 02	0120 GNSAMPS DB	BX
0DES 01	0121 DB	BY
0DE9 00	0122 DB	BZ
0DEA	0123 *	
0DEA	0124 *	SAMPLE TIME. USE THE SAMPLE COUNT TO DETERMINE
0DEA	0125 *	WHICH MAG SAMPLES TO TAKE AND WHEN TO SHIFT, ETC.
0DEA	0126 *	
0DEA 21 2F 21	0127 MAGSAMP LXI	H,SMPCNT SMPCNT++
0DED 34	0128 INR	M
0DEE 01 34 21	0129 LXI	B,BXDTY SAMPLE AUTOGAINS
0DF1 11 3D 21	0130 LXI	D,BYSAMP
0DF4 C3 84 0E	0131 JMP	TRIPLET
0DF7	0132 *	
0DF7	0133 *	MAG ENCODING TIME
0DF7	0134 *	
0DF7 3A 2F 21	0135 MAGENCO LDA	SMPCNT IF 1ST SAMPLE TAKEN
0DFA FE 01	0136 CPI	1
0DFC CC 43 0E	0137 CZ	SAVLOW
0DFF	0138 *	
0DFF 2A 3D 21	0139 LHL D	BYSAMP SAVE AUTOGAIN VALUES
0E02 CD 58 0E	0140 CALL	AGSTORE
0E05 2A 3F 21	0141 LHL D	BYSAMP
0E08 CD 58 0E	0142 CALL	AGSTORE
0E0B 2A 41 21	0143 LHL D	BYSAMP
0E0E CD 58 0E	0144 CALL	AGSTORE
0E11	0145 *	
0E11 21 3D 21	0146 LXI	H,TPMX STORE THE THREE GAINS
0E14 11 34 21	0147 LXI	D,BXDTY
0E17 CD 77 0E	0148 CALL	GAINSET
0E1A CD 77 0E	0149 CALL	GAINSET

0E1D CD 77 0E	0150	CALL	GAINSET
0E20	0151 *		
0E20 3A 2F 21	0152	LDA	SMPCNT IF AFTER THE 1ST SAMPLE
0E23 FE 01	0153	CPI	1 BUFFER THE MODE INFO
0E25 CA 38 0E	0154	JZ	PUTMODE
0E28 FE 08	0155	CPI	8 ON THE 8TH SAMPLE, PUT OUT
0E2A C0	0156	RNZ	. THE GAINS
0E2B 2A 30 21	0157	LHLD	TMPX
0E2E 22 29 21	0158	SHLD	GAINX
0E31 3A 32 21	0159	LDA	TMPZ
0E34 32 2B 21	0160	STA	GAINZ
0E37 C9	0161	RET	.
0E38	0162 *		
0E38 21 04 21	0163	PUTMODE LXI	H,MOBITS
0E3B 3A 33 21	0164	LDA	MODE
0E3E E6 0F	0165	ANI	0FH
0E40 B6	0166	ORA	M
0E41 77	0167	MOV	M,A
0E42 C9	0168	RET	.
0E43	0169 *		
0E43 2A 37 21	0170	SAVLOW LHLD	BXL SAVE ALL LOW GAIN VERSIONS
0E46 CD 52 0E	0171	CALL	LGSTORE
0E49 2A 39 21	0172	LHLD	BYL
0E4C CD 52 0E	0173	CALL	LGSTORE
0E4F 2A 3B 21	0174	LHLD	BZL
0E52 11 2C 21	0175	LGSTORE LXI	D,LGPTR
0E55 C3 5B 0E	0176	JMP	MAGSTORE
0E58	0177 *		
0E58	0178 *	MAG STORE MECHANISM.	
0E58	0179 *	ON ENTRY: [DE]->STORAGE POINTER	
0E59	0180 *	[HL]= 12-BIT VALUE TO STORE	
0E58	0181 *		
0E58 11 2D 21	0182	AGSTORE LXI	D,AGPTR
0E5B 1A	0183	MAGSTORE LDAX D	UPDATE THE BUFFER POINTER
0E5C C6 03	0184	ADI	3 BY THE # NIBBLES
0E5E 12	0185	STAX	D
0E5F B7	0186	ORA	A
0E60 1F	0187	RAR	. DIVIDE TO GET #BYTES
0E61 5F	0188	MOV	E,A [DE]->BUFFER
0E62 DA 6E 0E	0189	JC	ODD
0E65 29	0190	DAD	H ON EVEN STORES, LEFT ADJUST
0E66 29	0191	DAD	H THE 12-BIT VALUE
0E67 29	0192	DAD	H
0E68 29	0193	DAD	H
0E69 EB	0194	XCHG	.
0E6A 72	0195	MOV	M,D
0E6B 23	0196	INX	H
0E6C 73	0197	MOV	M,E
0E6D C9	0198	RET	.
0E6E	0199 *		

0E6E EB	0200 ODD	XCHG .	[DE]=VALUE, HL->BUFFER
0E6F 7A	0201	MOV A,D	"OR" THESE BITS INTO BUFFER
0E70 E6 0F	0202	ANI 0FH	ON THE ODD STORES
0E72 86	0203	ORA M	
0E73 77	0204	MOV M,A	
0E74 23	0205	INX H	
0E75 73	0206	MOV M,E	
0E76 C9	0207	RET .	
0E77	0208 *		
0E77	0209 *	STGRE	GAIN BIT FROM QTY IN MEM[DE]
0E77	0210 *		
0E77 1A	0211	GAINSET LDAX D	IF HIGAIN, SET CARRY
0E78 E6 10	0212	ANI HIGAIN	ELSE CLEAR IT
0E7A CA 7E 0E	0213	JZ MG1	
0E7D 37	0214	STC	
0E7E 7E	0215	MG1 MOV A,M	PUT CARRY INTO MEM[HL]
0E7F 17	0216	RAL	
0E80 77	0217	MOV M,A	
0E81 13	0218	INX D	
0E82 23	0219	INX H	
0E83 C9	0220	RET	
0E84	0221 *		
0E84	0222 *	SAMPLE	AND STORE A TRIPLET
0E84	0223 *		
0E84 CD 8A 0E	0224	TRIPLET CALL S1	
0E87 CD 8A 0E	0225	CALL S1	
0E8A 0A	0226	S1 LDAX B	SET QTY
0E8E 03	0227	INX B	
0E9C CD E6 00	0228	CALL	SAMPLE
0E8F EB	0229	XCHG	
0E90 73	0230	MOV M,E	
0E91 23	0231	INX H	
0E92 72	0232	MOV M,D	
0E93 23	0233	INX H	
0E94 EB	0234	XCHG .	
0E95 C9	0235	RET	
0E96 00	0236	DB 257	END-OF-MAG
0E97	0237 *		
0E97	0238 *	ENTER	COMMAND VECTOR IN TABLE
0E97	0239 *		
0E97	0240	ORG 0FBH/4+CMDTAB	
007E 70 00	0241	DW MAGCMD	
0080	0242 *		
0080	0243 *	RAM	SECTION
0080	0244 *		
0080	0245	ORG MAGRAM	
2100	0246	LGBUF DS 3*12+4/8	LOW GAIN BUFFER
2105	0247	MDBITS EQU 4-1	MODE BITS
2105	0248	AGBUF DS 8*3*12/8	AUTO GAIN BUFFER
2129	0249	GAINX DS 1	GAIN BYTES FOR X,Y,Z

212A	0250	GAINY	DS	1	
212B	0251	GAINZ	DS	1	
212C	0252	*			
212C	0253	LGPTR	DS	1	LOW GAIN POINTER
212D	0254	AGPTR	DS	1	AUTOGAIN POINTER
212E	0255	OTPTR	DS	1	OUTPUT POINTER
212F	0256	SMPCNT	DS	1	SAMPLE COUNTER
2130	0257	TMPX	DS	1	TEMP GAINS FOR X,Y,Z
2131	0258	TMPY	DS	1	
2132	0259	TMPZ	DS	1	
2133	0260	MODE	DS	1	MODE BYTE (4 BITS)
2134	0261	EXQTY	DS	1	
2135	0262	BYQTY	DS	1	
2136	0263	BZQTY	DS	1	
2137	0264	BXL	DS	2	
2139	0265	BYL	DS	2	
213B	0266	BZL	DS	2	
213D	0267	BXSAMP	DS	2	
213F	0268	BYSAMP	DS	2	
2141	0269	BZSAMP	DS	2	
2143	0270	*			
2143	0271	*	DEFINE WHERE MAG SAMPLES ARE		
2143	0272	*			
2143	0273	MAGDTA	EQU		BXQTY
2143	0274		COM		MAGDTA
2143	0275	*			
2143	0276	*	PUT GAIN LIMITS IN DSC (OTHERS,4)		
2143	0277	*			
2143	0278	LIMITS	EQU		024A6H

```

0000      0001 *
0000      0002 * CRRES FLIGHT PROGRAM---PLASMA DATA MANAGEMENT
0000      0003 * WRITTEN BY PETER P HARVEY
0000      0004 *
0000      0005 * FILE : PLA.A  VERSION 6 (DEC 88)
0000      0006 *
0000      0007 PSW   EQU   6      5025 SPECIFIC INFORMATION
0000      0008 SP   EQU   6
0000      0009 *
0000      0010 HIGAIN EQU  10H   HIGAIN INDICATOR
0000      0011 BYAMP EQU  7      SPIN AXIS MEASUREMENT AMPLIFIER
0000      0012 *
0000      0013 * SHIFT REGISTER DEFINITION
0000      0014 *
0000      0015 PVALID EQU  30H   PLA CALCULATION VALID IF 1
0000      0016 FDISABL EQU  80H   PLA PACKAGE DISABLED
0000      0017 FSEND  EQU  40H   PLA SENDING
0000      0018 LPMODE EQU  30H   LP INSTRUMENT INFO
0000      0019 LPFREQ EQU  0FH   LP SAMPLING FFEQUENCY
0000      0020 *
0000      0021 MANT  EQU  3FH   MANTISSA PART OF RESULT
0000      0022 SIGN  EQU  40H   SIGN OF RESULT
0000      0023 OVER  EQU  80H   OVERFLOW ERROR BIT
0000      0024 *
0000      0025 * ENTRY POINTS
0000      0026 *
0000      0027      ORG   PLA
0000      0028      JMP   PLAINIT  INITIALIZATION
0000      0029      JMP   PLASAMP  SAMPLE TIME
0000      0030 *
0000      0031 * RETURN DIGITAL STATUS
0000      0032 *
0000      0033 PLAOSC LDA   PMODE  SHOW THE MODE PART
0000      0034      ANI   LPMODE+LPFREQ
0000      0035      LXI   H,PLSTAT  AND INTERNAL STATUS
0000      0036      ORA   M
0000      0037      RET
0000      0038 *
0000      0039 * INITIALIZE THE PLASMA PACKAGE
0000      0040 *
0000      0041 PLAINIT LXI  H,PLARAM  CLEAR ALL VARS
0000      0042      MVI  C,BYOFF-PLARAM
0000      0043      RST  ZERO/8
0000      0044      LXI  D,INIOFF  COPY INIT OFFSETS
0000      0045      MVI  C,OFFEND-INIOFF
0000      0046      RST  COPY/8
0000      0047      RET
0000      0048 *
0000      0049 * PERFORM PACKAGE COMMANDS
  
```

0E85	0050 †		
0E85 32 56 21	0051 PLACMD STA	RNODE	
0E88 07	0052 RLC	.	IF ALG DISABLE/ENABLE SET IT
0E89 00	0053 RNC		
0E8A E6 80	0054 ANI	PDISABL	
0E8C 32 59 21	0055 STA	PSTAT	
0E8F 0F	0056 RET		
0ED0	0057 †		
0ED0	0058 †	SAMPLE TIME	
0ED0	0059 †		
0ED0 3A 59 21	0060 PLASAMP LDA	FSTAT	IF DISABLED, RETURN
0ED3 E6 80	0061 ANI	PDISABL	
0ED5 00	0062 FMT		
0ED6 05	0063 PUSH	B	
0ED7 21 59 21	0064 LYI	H, TIMER	ALTERNATE BETWEEN PART1 AND 2
0ECA 34	0065 INR	M	
0ECB 7E	0066 MOV	A, M	
0ECC 0F	0067 RRC		
0ECD 02 F4 0E	0068 JNC	FART2	
0ED0	0069 †		
0ED0 00 24 0F	0070 CALL	CVTBX	CONVERT BX INTO nT
0ED3 22 50 21	0071 SHLD	ASCBY	SAVE ABS(B/C BX)
0ED6 32 56 21	0072 STA	SGNBX	
0ED9	0073 †		
0ED9 00 30 0F	0074 CALL	CVTBY	CONVERT BY INTO nT
0EDC 22 54 21	0075 SHLD	ASCBZ	SAVE ABS(S/D BZ)
0EDF 00 57 21	0076 STA	SGNBZ	
0EE2	0077 †		
0EE2 4F	0078 MOV	C, A	SAVE ITS SIGN
0EE3 3E 0B	0079 MVI	A, 11	MULTIPLY BY 11/256 (.044 APPX)
0EE5 EB	0080 XCHG		
0EE6 00 0A 06	0081 CALL	MUZI	
0EE9 6C	0082 MOV	L, H	
0EEA 67	0083 MOV	H, A	
0EEB 79	0084 MOV	A, C	
0EEC 00 05 0F	0085 CALL	APPLY	APPLY SIGN
0EEF 22 52 21	0086 SHLD	ASCBY	SAVE (11/256)*SCBY
0EF2 C1	0087 POP	B	
0EF3 C9	0088 RET		
0EF4	0089 †		
0EF4 00 30 0F	0090 PART2 CALL	CVTBZ	CONVERT BZ INTO nT
0EF7 00 05 0F	0091 CALL	APPLY	[HL]=SIGNED SCBZ
0EFA EB	0092 YCHG	.	ADD IT TO SAVED VAL
0EFB 2A 52 21	0093 LHLD	ASCBY	
0EFE 19	0094 DAD	D	
0EFF 00 04 0F	0095 CALL	ABS16	TAKE ABSOLUTE VALUE
0F02	0096 †		
0F02 97	0097 SUB	A	CLEAR "PSEND" BIT
0F03 32 59 21	0098 STA	PSTAT	
0F06 C1	0099 POP	B	

0F07 29	0100	DAD	H	IF ABS(SCBY)*8 > ABS(SCB1) RETURN
0F08 08	0101	RC		
0F09 29	0102	DAD	H	
0F0A 08	0103	RC		
0F0B 2F	0104	DAD	H	
0F0C 08	0105	RC		
0F0D 3A 50 21	0106	LDA	ASC BX	
0F10 53	0107	SUB	L	
0F11 3A 51 21	0108	LDA	ASC BX+1	
0F14 5C	0109	SBB	H	
0F15 08	0110	RC	.	
0F16	0111			
0F16 05	0112	PUSH	B	
0F17 0D 34 0F	0113	CALL	ZCALC CALCULATE THE VALUE TO SEND	
0F1A 0D 30 03	0114	CALL	BETPLA	
0F1D 01	0115	POP	B	
0F1E 3E 40	0116	MVI	A, PSEND INDICATE SENDING	
0F20 32 5F 21	0117	STA	PSTAT	
0F23 09	0118	RET		
0F24	0119			
0F24	0120		CONVERSION ROUTINES	
0F24	0121			
0F24 3A 34 21	0122	CVTB1	LDA	BXQTY CONVERT BX INTO NANOTESLA
0F27 2A 30 21	0123	LHLD	BYSAMP	
0F2A 01 5C 21	0124	LXI	B, BXOFF	
0F2D 03 5F 0F	0125	JMP	CONVERT A=SIGN, HL=MAGNITUDE	
0F30	0126			
0F30 3A 36 21	0127	CVTB2	LDA	BZQTY
0F33 2A 41 21	0128	LHLD	BZSAMP	
0F36 01 68 21	0129	LXI	B, BZOFF	
0F39 03 5F 0F	0130	JMP	CONVERT	
0F3C	0131			
0F3C 3E 07	0132	CVTB3	MVI	A, BYAMP TEST WHETHER *6 IS ON
0F3E 0D E4 02	0133	CALL	TSTMUX	
0F41 0A 56 0F	0134	JZ	AMPOFF NO: GO	
0F44 01 64 21	0135	AMPON	LXI	B, BYOFF+4 IF AMPLIFIED
0F47 0D 5F 0F	0136	CALL	CONBY CONVERT 1ST	
0F4A 4F	0137	MOV	C, A SAVE SIGN OF BY	
0F4B 3E 27	0138	MVI	A, 39 SCALE BY 39/256	
0F4D E8	0139	XCHG	.	
0F4E 0D 0A 03	0140	CALL	MU21	
0F51 6C	0141	MOV	L, H	
0F52 67	0142	MOV	H, A	
0F53 79	0143	MOV	A, C	
0F54 2F	0144	CMA	.	(AMP IS INVERTING)
0F55 05	0145	RET		
0F56	0146			
0F56 01 60 21	0147	AMPOFF	LXI	B, BYOFF IF UNAMPLIFIED
0F59 3A 35 21	0148	CONBY	LDA	BYQTY
0F5C 2A 3F 21	0149	LHLD	BYSAMP	

0F5F	0150	*
0F5F	0151	* CONVERT MAGNETOMETER SAMPLE TO NANOTESLA
0F5F	0152	* ON ENTRY: [A]= MUX QTY (HI OR LO GAIN)
0F5F	0153	* [HL]= VALUE
0F5F	0154	* [BC]=>OFFSET PAIR
0F5F	0155	*
0F5F E6 10	0156	CONVERT ANI HIGAIN IF IN LOW GAIN, MULTIPLY
0F61 3E 00	0157	MOV A,0
0F63 C2 6A 0F	0158	JNZ CV1 BY THE GAIN FACTOR
0F66 3E 33	0159	MOV A,51
0F68 03	0160	INX B AND USE THE LO OFFSET
0F69 03	0161	INX B
0F6A F5	0162	CV1 PUSH PSW
0F6B	0163	*
0F6B 7C	0164	MOV A,H EXTEND 12 BITS TO 16
0F6C FE 08	0165	CPI B
0F6E DA 74 0F	0166	JC CVPOS
0F71 F6 F0	0167	ORI 0F0H
0F73 67	0168	MOV H,A
0F74	0169	CVPOS EQU *
0F74	0170	*
0F74 0A	0171	LDAX B ADD OFFSET FROM MEM[BC]
0F75 5F	0172	MOV E,A
0F76 03	0173	INX B
0F77 0A	0174	LDAX B
0F78 57	0175	MOV D,A
0F79 19	0176	DAD D
0F7A	0177	*
0F7A 7C	0178	MOV A,H
0F7B 32 5A 21	0179	STA TEMP SAVE SIGN
0F7E CD D4 0F	0180	CALL ABS16 CONVERT TO POSITIVE
0F81 F1	0181	POP PSW
0F82 54	0182	MOV D,H
0F83 5D	0183	MOV E,L
0F84 CC 0A 06	0184	CZ MU21 IF LOW GAIN, AHL=A+DE
0F87 CD 93 0F	0185	CALL DIV4 AHL= AHL/4
0F8A B7	0186	ORA A IF A HAS ANYTHING, OVERFLOW
0F8B 3A 5A 21	0187	LDA TEMP RETURN(SIGN)
0F8E C8	0188	RZ .
0F8F 21 FF 7F	0189	LYI H,7FFFH
0F92 C9	0190	RET
0F93	0191	*
0F93 B7	0192	DIV4 ORA A SHIFT RIGHT TWICE
0F94 1F	0193	RAR
0F95 23	0194	INX H ROUND OFF
0F96 CD 9B 0F	0195	CALL SRHL
0F99 B7	0196	ORA A
0F9A 1F	0197	RAR
0F9B 4F	0198	SRHL MOV C,A
0F9C 7C	0199	MOV A,H

0F9D 1F	0200	RAR	
0F9E 67	0201	MOV	H,A
0F9F 7D	0202	MOV	A,L
0FA0 1F	0203	RAR	
0FA1 6F	0204	MOV	L,A
0FA2 79	0205	MOV	A,C
0FA3 09	0206	RET	
0FA4	0207	*	
0FA4	0208	*	TIME TO MAKE THE CALCULATION OF BZ/BX
0FA4	0209	*	ON EXIT: [HL]= VALUE TO SEND TO LEPA
0FA4	0210	*	
0FA4 2A 50 21	0211	ZCALC	LHLD ASCBX A=ABS(BZ/2)/ABS(BX)
0FA7 EB	0212	XCHG	
0FAB 2A 54 21	0213	LHLD	ASCBZ
0FAB B7	0214	DRA	A SHIFT RIGHT
0FAC 0D 9B 0F	0215	CALL	SRHL
0FAF 0D DA 0F	0216	CALL	QDIV A=[HL]/[DE] (8-BITS)
0FB2 3C	0217	INR	A ROUND OFF LAST BIT
0FB3 B7	0218	DRA	A
0FB4 1F	0219	RAR	.
0FB5	0220	*	
0FB5 FE 40	0221	CFI	MANT+1 IF GREATER THAN MANTISSA
0FB7 DA BE 0F	0222	JC	MNTOK CAN BE.
0FBA E6 3F	0223	ANI	MANT MASK MANT BITS
0FBC F6 80	0224	ORI	OVER THEN SET THE OVERFLOW
0FBE 5F	0225	MNTOK	MOV E,A SAVE THIS IN E
0FBF	0226	*	
0FBF 3A 56 21	0227	LDA	SGNBX COMPUTE THE SIGN DIFFERENCE
0FC2 21 57 21	0228	LXI	H,SGNBZ
0FC5 AE	0229	XRA	M
0FC6 78	0230	MOV	A,E AND INSERT THE PLA SIGN
0FC7 F2 0C 0F	0231	JP	PLAPOS
0FCA F6 40	0232	ORI	SIGN
0FCC	0233	PLAPOS	EQU *
0FCC 6F	0234	MOV	L,A SEND MODE AND RESULTS TO
0FCD	0235	*	
0FCD 3A 58 21	0236	LDA	PNODE THE LEPA INSTRUMENT
0FD0 F6 80	0237	OPI	PVALID SIGNIFY VALID BZ/BX
0FD2 67	0238	MOV	H,A
0FD3 09	0239	RET	.
0FD4	0240	*	
0FD4	0241	*	ABSOLUTE VALUE
0FD4	0242	*	
0FD4 7C	0243	ABS16	MOV A,H IF POSITIVE, RETURN
0FD5 B7	0244	APPLY	DRA A
0FD6 F0	0245	RP	.
0FD7 03 9B 00	0246	JMP	NEG16 ELSE [HL]=-[HL]
0FDA	0247	*	
0FDA	0248	*	QUICK DIVIDER
0FDA	0249	*	A = [HL]/[DE] TO 8 BITS

0FDA	0250 †			
0FDA 7A	0251 QDIV	MOV	A,D	[BC]= -DIVISOR
0FDB 2F	0252	CMA		
0FDC 47	0253	MOV	B,A	
0FDD 78	0254	MOV	A,E	
0FDE 2F	0255	CMA		
0FDF 4F	0256	MOV	C,A	
0FE0 03	0257	INX	B	
0FE1	0258 †			
0FE1 3E 01	0259	MV1	A,1	SET MARKER FOR BTH SHIFT
0FE3 C3 E9 0F	0260	JMP	QDIV1	
0FE6	0261 †			
0FE6 17	0262 QDIVN	RAL	.	SHIFT RESULT INTO ACCUM
0FE7 D8	0263	RC	.	WHEN MARKER SETS CRY. RETURN
0FEB 29	0264	DAD	H	SHIFT REMAINDER
0FE9 54	0265 QDIV1	MOV	D,H	SAVE A COPY OF REMAINDER
0FEA 50	0266	MOV	E,L	
0FEB 09	0267	DAD	B	REM=REM-DIVISOR
0FEC DA E6 0F	0268	JC	QDIVN	IF POS, RESULT=1
0FEF EB	0269	XCHG	.	ELSE RESULT=0, GET OLD
0FF0 C3 E6 0F	0270	JMP	QDIVN	REMAINDER BACK AGAIN
0FF3	0271 †			
0FF3 A5 FF	0272 INIOFF	DW	-91	BXH,BXL OFFSETS
0FF5 FC FF	0273	DW	-4	
0FF7 F4 FF	0274	DW	-12	BY AMP OFF
0FF9 FE FF	0275	DW	-2	
0FFB 60 00	0276	DW	+96	BY AMP ON
0FFD 01 00	0277	DW	+1	
0FFF EA FF	0278	DW	-22	BZ
1001 FE FF	0279	DW	-2	
1003	0280 OFFEND	EQU	\$	
1003 00	V 0281	DB	256	END OF PLA MODULE
1004	0282 †			
1004	0283 †	ENTER COMMAND VECTOR INTO TABLE		
1004	0284 †			
1004	0285	ORG	0DBH/4+CMDTAB	
0076 B5 0E	0286	DW	PLACMD	
0078	0287 †			
0078	0288 †	VARIABLES		
0078	0289 †			
0078	0290	ORG	PLARAM	
2150	0291 ASCBX	DS	2	ABSOLUTE VALUES OF S/C BX, BY, BZ
2152	0292 ASCBY	DS	2	
2154	0293 ASCBZ	DS	2	
2156	0294 SGNBX	DS	1	SIGN OF BX
2157	0295 SGNBZ	DS	1	SIGN OF BZ
2158	0296 PMODE	DS	1	MODE INFORMATION
2159	0297 PSTAT	DS	1	PACKAGE STATUS
215A	0298 TEMP	DS	1	
215B	0299 TIMER	DS	1	

215C	0300	BZOFF	DS	4	
2160	0301	BYOFF	DS	412	
2158	0302	BZOFF	DS	4	
216C	0303	ENDPLA	EQU	4	
216C	0304	↑			
216C	0305	↑	EXTERNAL DATA		
216C	0306	↑			
216C	0307		ORG	MAGDTA	
2134	0308	BZQTY	DS	1	MAG MUX ADDRESSES
2135	0309	BYQTY	DS	1	
2136	0310	BZQTY	DS	1	
2137	0311	BXL	DS	2	AND LOW SAMPLES
2139	0312	BYL	DS	2	
2138	0313	BZL	DS	2	
213D	0314	BXSAMP	DS	2	AND AUTOGAIN SAMPLES
213F	0315	BYSAMP	DS	2	
2141	0316	BZSAMP	DS	2	

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE---BURST TRIGGERING CONTROL
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 *
0000          0005 * FILE BUR.A
0000          0006 *
0000          0007 PSW   EQU   6
0000          0008 EP   EQU   6
0000          0009 *
0000          0010 * BURST PROCESSOR COMMANDS
0000          0011 *
0000          0012 BGO   EQU   0B400H  START SAMPLING
0000          0013 BSTOP EQU   0B500H  STOP SAMPLING
0000          0014 BPAUSE EQU  0B600H  PAUSE SAMPLING
0000          0015 BCONT EQU   0B700H  CONTINUE SAMPLING
0000          0016 BPLAY EQU   0B800H  BEGIN PLAYBACK
0000          0017 BRESET EQU  0B900H  RESET BURST
0000          0018 *
0000          0019      ORG   BUR
100B C3 17 10  0020      JMP   BURINIT  INITIALIZATION
100B C3 65 10  0021      JMP   BURSAMP   SAMPLING
100E C3 38 11  0022      JMP   BURPLAY  PLAYBACK
1011          0023 *
1011          0024 * RETURN BURST DIGITAL STATUS OF MODULE
1011          0025 *
1011 21 70 21  0026 BURDSC LXI   H,BURRAM
1014 DF        0027      RST   REF/8
1015 7E        0028      MOV   A,M
1016 C9        0029      RET
1017          0030 *
1017          0031 * INITIALIZE THE BURST TRIGGERING PACKAGE
1017          0032 *
1017 97        0033 BURINIT SUB  A    CLEAR THE PLAYBACK REQUEST
1018 32 70 21  0034      STA   MODFREQ
1018 21 10 00  0035      LXI   H,16  SET DEFAULT DURATION=4 SECONDS
101E 22 72 21  0036      SHL3  WAITTIME
1021 C3 44 03  0037      JMP   RBURST  RESET BURST TO START
1024          0038 *
1024          0039 * PERFORM TRIGGERING COMMANDS
1024          0040 *
1024 11 96 21  0041 ALBCMD LXI   D,ALGPAMS
1027 C3 29 10  0042      JMP   SETPAM
102A 11 71 21  0043 CTLCMD LXI   D,CTLPAMS
102D 7C        0044 SETPAM MOV  A,H   PUT THE VALUE INTO REGISTER
102E E6 03    0045      ANI   3    DESCRIBED BY THE PEG FIELD
1030 EB        0046      XCHG  .
1031 DF        0047      RST   REF/8
1032          0048 *
1032 97        0049      SUB  A    CLEAR TRIG MODE
  
```

1033 32 71 21	0050	STA	BTMODE
1036 73	0051	MOV	M.E SET NEW PARAMETER
1037	0052 #		
1037 3A 71 21	0053	LDA	BTMODE IF ANY TRIGGER SELECTED
103A E6 07	0054	ANI	ALGBITS THEN RESET IT.
103C 08	0055	RZ	.
103D 3E 40	0056	START MVI	A,R1
103F 0D AE 11	0057	CALL	SETMODE
1042 E7	0058	ORA	A
1043 09	0059	RET	
1044	0060 #		
1044 97	0061	BURCMD SUB	A CLEAR TRIGGER ON ANY DIRECT COMMAND
1045 32 71 21	0062	STA	BTMODE
1048 0D 4D 10	0063	CALL	DOBUR EXECUTE COMMAND
104B B7	0064	ORA	A RETURN NO CARRY
104C 09	0065	EXIT	RET
104D	0066 #		
104D 7C	0067	DOBUR	MOV A,H GET THE 0BXH COMMAND
104E FE B9	0068	CPI	BRESET/256 IF RESET, DO IT
1050 CA 17 10	0069	JZ	BURINIT AND CLEAR THIS PACKAGE
1053 FE B4	0070	CPI	BGD/256
1055 CA 3D 10	0071	JZ	START
1058 FE B5	0072	CPI	BSTOP/256
105A CA E9 10	0073	JZ	TRIGGER
105D FE B8	0074	CPI	BPLAY/256
105F CA 1F 11	0075	JZ	STPLAY
1062 C3 53 03	0076	JMP	SEND
1065	0077 #		
1065	0078 #		SAMPLE CONDITIONS TO DECIDE IF WE SHOULD BURST
1065	0079 #		OR NOT.
1065	0080 #		
1065 0D A8 11	0081	BURSAMP CALL	GETMODE RETRIEVE SAMPLING MODE
1068 0F	0082	RRC	
1069 0F	0083	RRC	
106A 0F	0084	RRC	
106B E6 0E	0085	ANI	7#2
106D 21 73 10	0086	LXI	H,BSVECT
1070 C3 05 10	0087	JMP	BEX
1073	0088 #		
1073 4C 10	0089	BSVECT DW	EXIT OFF
1075 0B 10	0090	DW	BURTST SEARCH
1077 01 11	0091	DW	BURCOL COLLECT
1079 16 11	0092	DW	BURWAIT WAIT
107B 8D 10	0093	DW	BR1
107D 40 10	0094	DW	BR2
107F B9 10	0095	DW	BR3
1081 B3 10	0096	DW	BR0
1083	0097 #		
1083 21 79 21	0098	BR0 LXI	H,TEMP DELAYED BR1
1086 35	0099	DCR	M

1087 C0	0100	RNZ	
1088 3E 40	0101	MVI	A,R1
108A C3 AE 11	0102	JMP	SETMODE
108D	0103	*	
108D	0104	*	RESET TRIGGERING (STAGES 1-3)
108D	0105	*	
108D 21 00 B4	0106 BR1	LXI	H,B60 START THE BURST GOING
1090 CD 53 03	0107	CALL	SEND
1093 97	0108	SUB	A ZERO THE TEMP
1094 32 79 21	0109	STA	TEMP
1097 3A 71 21	0110	LDA	BTMODE SAVE WHICH TRIGGER
109A 32 B6 21	0111	STA	TRIGR
109D C3 C1 11	0112	JMP	INCMODE
10A0	0113	*	
10A0 CD 84 03	0114 BR2	CALL	RECEIVE GET THE 3 WORD INFO
10A3 C8	0115	RZ	. (IF NOT READY, TRY NEXT TIME)
10A4 3A 70 21	0116	LDA	MODFREQ 1ST IS THE "REAL FREQ" INFO
10A7 E6 F0	0117	ANI	-1-FREQBITS
10A9 B5	0118	ORA	L
10AA 32 70 21	0119	STA	MODFREQ
10AD	0120	*	
10AD CD 84 03	0121 BR2W	CALL	RECEIVE NEXT IS THE TOTAL DURATION
10B0 CA AD 10	0122	JZ	BR2W
10B3 22 B0 21	0123	SHLD	DURATION WHICH THE BURST
10B6 C3 C1 11	0124	JMP	INCMODE
10B9	0125	*	
10B9 CD 84 03	0126 BR3	CALL	RECEIVE CAN HOLD IN ITS CURRENT
10BC C8	0127	RZ	
10BD 22 B2 21	0128	SHLD	DURATION+2 STATE.
10C0 21 87 21	0129	LXI	H,ST SAVE TIME WHEN THINGS BEGAN
10C3 CD E0 11	0130	CALL	SAVETIME
10C6 3E 10	0131	MVI	A,SEARCH AND BEGIN SEARCHING
10C8 C3 AE 11	0132	JMP	SETMODE
10CB	0133	*	
10CB	0134	*	SEARCH PHASE. USE COMMANDED ALGORITHM FOR SEARCH
10CB	0135	*	
10CB 3A 71 21	0136 BURTST	LDA	BTMODE GET THE ALGORITHM #
10CE E6 07	0137	ANI	ALGBITS
10D0 C8	0138	RZ	.
10D1 87	0139	ADD	A
10D2 21 D9 10	0140	LXI	H,BATABLE-2 REFERENCE ALGORITHM
10D5 DF	0141 BEX	RST	REF/8
10D6 7E	0142	MOV	A,M
10D7 23	0143	INX	H
10DB 66	0144	MOV	H,M
10D9 8F	0145	MOV	L,A
10DA E9	0146	PCHL	.
10EB	0147	*	
10EB E9 10	0148	BATABLE DW	TRIGGER THE TRIGGERING ALGORITHM LIST
10DD 73 11	0149	DW	VALCHK

10DF 9F 11	0150	DW	MASCHK
10E1 9F 11	0151	DW	RAMALB
10E3 9F 11	0152	DW	RAMALB
10E5 9F 11	0153	DW	RAMALB
10E7 9F 11	0154	DW	RAMALB
10E9	0155	*	
10E9	0156	*	EVENT TRIGGER
10E9	0157	*	
10E9 21 8C 21	0158	TRIGGER LXI	H,VT SAVE THE TIME OF THE EVENT
10EC CD 50 11	0159	CALL	SAVETIME
10EF 5A 72 21	0160	LHLD	WAITTIME SET DELAY FOR THAT COMMANDED
10F2 22 7B 21	0161	SHLD	DTIME+1 AS OPPOSED TO THE
10F5 97	0162	SUB	A MEMORY CAPACITY
10F6 32 7A 21	0163	STA	DTIME
10F9 32 7D 21	0164	STA	DTIME+3
10FC 3E 20	0165	MVI	A,COLLECT MODE="COLLECT"
10FE C3 AE 11	0166	JMP	SETMODE
1101	0167	*	
1101	0168	*	COLLECTION PHASE
1101	0169	*	
1101 CD C9 11	0170	BURDOL CALL	CLKTICK COUNT 1 CLOCK TIME
1104 D0	0171	RNC	. IF NOT TIME TO QUIT, RETURN
1105 21 00 B5	0172	LXI	H,ESTOP COMMAND THE BURST PROCESSOR
1108 CD 53 03	0173	CALL	SEND TO STOP NOW.
110B 21 91 21	0174	LXI	H,ET SAVE THE END TIME
110E CD E0 11	0175	CALL	SAVETIME
1111 3E 20	0176	MVI	A,WAIT
1113 C3 AE 11	0177	JMP	SETMODE
1116	0178	*	
1116	0179	*	WAIT FOR BURST CPU TO PROCESS ITS MEMORY
1116	0180	*	
1116 CD 84 03	0181	BURWAIT CALL	RECEIVE GET THE READY FOLLOWING A STOP
1119 C8	0182	RZ	. IF NOT THERE, TRY NEXT TIME
111A 3E 00	0183	MVI	A,OFF TURN OFF SAMPLING SECTION
111C CD AE 11	0184	CALL	SETMODE
111F	0185	*	
111F	0186	*	START PLAYBACK SEQUENCE IN STANDARD FORMAT
111F	0187	*	
111F 21 00 B8	0188	STPLAY LXI	H,BPLAY COMMAND THE BURST
1122 CD 53 03	0189	CALL	SEND TO PLAY BACK
1125 3E B1	0190	MVI	A,0B1H SET THE FORMAT CODE
1127 32 85 21	0191	STA	FORMAT
112A 3E 85	0192	MVI	A,HEADR#256/256 PUT OFFSET ADDR
112C 32 84 21	0193	STA	H51HX INTO THE HEADER INDEX
112F 3A 70 21	0194	LDA	MODFREQ START UP THE PLAYBACK
1132 F6 80	0195	ORI	PLAYBIT BY TURNING ON ITS BIT
1134 32 70 21	0196	STA	MODFREQ
1137 C9	0197	RET	
1138	0198	*	
1138	0199	*	RETRIEVE PLAYBACK DATA FOR THE TELEMETRY SYSTEM.

```

1138                                0200 *
1138 21 84 21                        0201 BURPLAY LX1 H,HDINX IF HDINX < HDEND,
1138 7E                                0202     MOV  A,M   THEN OUTPUT HEADER INFO
113C FE 9A                            0203     CP1   XHEADR#256/256
113E DA 6D 11                          0204     JC   BPHEAD
1141 CD 84 03                          0205     CALL RECEIVE ELSE GRAB NEXT DATA
1144 C0                                0206     RNZ   .      AND RETURN(HL) IF THERE
1145                                0207 *
1145 3A 70 21                          0208 ENPLAY LDA  MODFREQ IF THE PLAYBACK QUIT A WHILE
1148 E6 90                              0209     ANI  PLAYBIT BACK, JUST RETURN(0)
114A CA 69 11                          0210     JZ   ENPRO
114D 3A 70 21                          0211     LDA  MODFREQ ELSE REMOVE PLAYBACK REQUEST
1150 E6 7F                              0212     ANI  -1-PLAYBIT
1152 32 70 21                          0213     STA  MODFREQ
1155                                0214 *
1155 3E 10                              0215     MVI  A,512/32 DELAY 1/2 SECOND
1157 32 79 21                          0216     STA  TEMP
115A 3A 71 21                          0217     LDA  BTMODE IF AUTO-SEARCH MODE, RESTART
115D E6 80                              0218     ANI  AUTOSEARCH THE SEARCHING
115F 3E 70                              0219     MVI  A,R0
1161 C2 66 11                          0220     JNZ  ENFNM
1164 3E 00                              0221     MVI  A,OFF ELSE TURN OFF
1166 CD AE 11                          0222 ENFNM CALL  SETMODE
1169 21 00 00                          0223 ENPRO LXI  H,0  RETURN(0) AS A TRAILER
116C C9                                0224     RET
116D                                0225 *
116D 3A                                0226 BPHEAD INR  M   HDINX++ FOR NEXT TIME
116E 6F                                0227     MOV  L,A  [HLI->]THIS BYTE
116F 6E                                0228     MOV  L,M  [HLI]=BYTE
1170 26 00                              0229     MVI  H,0
1172 C9                                0230     RET  .
1173                                0231 *
1173                                0232 * BURST TRIGGER SECTION
1173                                0233 * VALUE CHECKING FOR SAMPLE > THRESHOLD
1173                                0234 *
1173 3A 96 21                          0235 VALCHK LDA  MUXAD SAMPLE THE MULTIPLEXOR
1176 CD E6 00                          0236     CALL SAMPLE
1179 29                                0237     DAD  H   SCALE TO 8 BITS
117A 29                                0238     DAD  H
117B 29                                0239     DAD  H
117C 29                                0240     DAD  H
117D 7C                                0241     MOV  A,H
117E CD 8A 11                          0242     CALL ABS
1181 67                                0243     MOV  H,A
1182 3A 97 21                          0244     LDA  THRESHOLD IF THRES < SAMPLE(MUXAD)
1185 BC                                0245     CMP  H   CALL THE TRIGGER START
1186 DC E9 10                          0246     CC   TRIGGER
1189 C9                                0247     RET
118A                                0248 *
118A 87                                0249 ABS  DRA  A

```

```

1188 F0          0250      RP
118C 2F          0251      CMA
1190 3C          0252      INR  A
118E C9          0253      RET
118F             0254      *
118F             0255      * MAG CHECKING ALGORITHM
118F             0256      *
118F             0257  LCONE  EDU  40H  "LOSS CONE BIT"
118F 97          0258  MAGCHK  SUB  A    GET PLA STATUS
1190 CD 9E 0E    0259      CALL  PLADSC  AND TRIGGER WHEN
1193 E6 40       0260      ANI  LCONE  LOSSCONE IS BEGINNING
1195 C8          0261      RZ  .
1196 21 79 21    0262      LXI  H,TEMP
1199 BE          0263      CMP  M
119A C3          0264      RZ
1195 77          0265      MOV  M,A
119C C3 E9 10    0266      JMP  TRIGGER
119F             0267      *
119F             0268      * RAM ALGORITHM
119F             0269      *
119F 3A 75 21    0270  RAMALG  LDA  RAMCODE  CHECK THAT IT'S LOADED
11A2 FE AA       0271      CPI  0AAH  OK
11A4 CA 76 21    0272      JZ   RAMCODE+1
11A7 C9          0273      RET
11A9             0274      *
11A8             0275      * UTILITIES
11A9             0276      *
11A9 3A 70 21    0277  GETMODE  LDA  MODFREQ  RETRIEVE SAMPLING MODE
11AB E6 70       0278      ANI  SMPBITS
11AD C9          0279      RET
11AE             0280      *
11AE E6 70       0281  SETMODE  ANI  SMPBITS  SET SAMPLE MODE
11B0 E5          0282      PUSH H
11B1 5F          0283      MOV  L,A
11B2 3A 70 21    0284      LDA  MODFREQ  INTO THE MODE/FREQ BYTE
11B5 E6 8F       0285      ANI  -1-SMPBITS
11B7 B5          0286      ORA  L
11B8 32 70 21    0287      STA  MODFREQ
11BB 6F          0288      MOV  L,A  NOW FORM A PMODE COMMAND
11BC 26 09       0289      MVI  H,PMODE  TO TELL THE PLASMA
11BE F7          0290      RST  6    INSTRUMENT
11BF E1          0291      POP  H
11C0 C9          0292      RET
11C1             0293      *
11C1 21 70 21    0294  INCMODE  LY1  H,MODFREQ  UPDATE MODE INTERNALLY
11C4 7E          0295      MOV  A,M  BUT DON'T TELL LEFA ABOUT IT
11C5 C6 10       0296      ABI  SMPBITS/7  (3-BIT FIELD)
11C7 77          0297      MOV  M,A
11C8 C9          0298      RET  .
11C9             0299      *
  
```

```

11C9          0300 * CLDCK TIMER.  DECREMENTS DTIME BY 32 MILLISECONDS
11C9          0301 * RETURNS NO CARRY IF READY.
11C9          0302 *
11C9 11 7A 21 0303 CLKTICK LXI  D,DTIME  DECREMENT THE DELAY TIMER
11CC 21 DC 11 0304          LXI  H,P32
11CF B7       0305          ORA  A    CLEAR CARRY
11D0 CD D3 11 0306          CALL SUB2
11D3 CD D6 11 0307 SUB2  CALL  SUB1
11D6 1A       0308 SUB1  LDAX  D
11D7 9E       0309          SBB  M
11D8 12       0310          STAX D
11D9 13       0311          INX  D
11DA 23       0312          INX  H
11DB C9       0313          RET
11DC          0314 *
11DC 20 00    0315 P32   DW    32
11DE 90 00    0316          DW    0
11E0          0317 *
11E0 11 1C 20 0318 SAVETIME LXI  D,SYSCLOCK  GET 5 BYTES OF
11E3 0E 05    0319          MVI  C,5    THE SYSTEM CLOCK AND STORE
11E5 D7       0320          RST  COPY/8
11E6 C9       0321          RET
11E7 00       V 0322          DB    256
11E8          0323 *
11E8          0324 * ENTER COMMAND VECTORS INTO TABLE
11E8          0325 *
11E8          0326          ORG   0A0H/4+CMDTAB
0068 2A 10    0327          DW    CTLCMD #A0
006A 24 10    0328          DW    ALGCMD #A8
006C 44 10    0329          DW    BURCMD #B0
006E 44 10    0330          DW    BURCMD #B8
0070          0331 *
0070          0332 * VARIABLES
0070          0333 *
0070          0334          ORG   BURRAM
2170          0335 MODFREQ DS  1    MODE AND FREQ NIBBLES (MMMMFFFF)
2171          0336 SMPBITS EQU  70H  SAMPLING MODE BITS
2171          0337 PLAYBIT EQU  80H  PLAYBACK BIT
2171          0338 FREQBITS EQU  0FH  FREQUENCY BITS
2171          0339 OFF  EQU    0    SAMPLE MODE VALUES
2171          0340 SEARCH EQU  10H
2171          0341 COLLECT EQU  20H
2171          0342 WAIT  EQU    30H
2171          0343 R1   EQU    40H  (RESETTING STAGES)
2171          0344 R2   EQU    50H
2171          0345 R3   EQU    60H
2171          0346 R0   EQU    70H
2171          0347 *
2171          0348 CTLPARMS EQU $  CONTROL PARAMS (4)
2171          0349 BTMODE DS  1    BURST TRIGGER MODE BITS

```

2172	0350	ALGBITS EQU	7	3 PRELAUNCH ALGORITHMS + OFF
2172	0351	AUTOSEARCH EQU	80H	AUTO-SEARCH MODE IF 1
2172	0352	PULSED EQU	40H	PULSED SAMPLING MODE IF 1
2172	0353	‡		
2172	0354	WAITTIME DS	2	BURST DURATION
2174	0355	SPARE DS	1	
2175	0356	‡		
2175	0357	RAMCODE DS	4	
2179	0358	TEMP DS	1	
217A	0359	DTIME DS	4	DELAY TIMER
217E	0360	RFREQ DS	2	REAL FREQUENCY CODE USED BY BURST
2180	0361	DURATION DS	4	DURATION READ BACK (MSEC)
2184	0362	HDINX DS	1	HEADER INDEX
2185	0363	‡		
2185	0364	‡		PLAYBACK HEADER STORED IN MEMORY
2185	0365	‡		
2185	0366	HEADR EQU	‡	
2185	0367	FORMAT DS	1	FORMAT CODE
2186	0368	TRIGR DS	1	TRIGGER ALGORITHM
2187	0369	ST DS	5	START TIME
2190	0370	VT DS	5	EVENT TIME
2191	0371	ET DS	5	END TIME
2196	0372	ALGPAMS DS	4	
219A	0373	XHEADR EQU	‡	
219A	0374	‡		
219A	0375	MUXAD EQU	ALGPAMS+0	TRIGGER 1 PARAMETERS
219A	0376	THRESHOLD EQU	MUXAD+1	
219A	0377	‡		
219A	0378	‡		EXTERNAL INFORMATION
219A	0379	‡		
219A	0380	SYSCLOCK EQU	BKGRAM+0	
219A	0381	PMODE EQU	008H	PMODE COMMAND CODE
219A	0382	‡		
219A	0383	OR6	PLA	
0E98	0384	PLAINIT DS	3	
0E9B	0385	PLASAMP DS	3	
0E9E	0386	PLADSC DS	3	

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE --- SPIN FIT MANAGER
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 * FILE : FIT.A
0000          0005 *
0000          0006 FMODE EQU 50H FIT MODE *
0000          0007 V12DI EQU 02H DISABLE V12 FITS
0000          0008 FITXMIT EQU 08H TRANSMIT ENABLE
0000          0009 FITDI EQU 20H DISABLE FITS
0000          0010 *
0000          0011 FLT EQU 3 3-BYTE FLOATING POINT
0000          0012 NULL EQU 40H NULL FLOAT VALUE IN 2ND BYTE
0000          0013 NSAMPS EQU 32 * POINTS PER SPIN FIT
0000          0014 AVPTS EQU 4 * AHI/ALO TO AVERAGE
0000          0015 V1ANG EQU 128+32 DEGREES FROM SUN PULSE TO START FIT
0000          0016 V3ANG EQU V1ANG+128 START V34 FIT 1/2 SPIN AWAY
0000          0017 *
0000          0018 V12F EQU 8 MAIN MULTIPLEXOR QUANTITIES
0000          0019 V34 EQU 15
0000          0020 HIGAIN EQU 10H HIGH GAIN INDICATOR
0000          0021 *
0000          0022 V12HEADER EQU 0A1H V12 HEADER BYTE
0000          0023 V34HEADER EQU 0A3H V34 HEADER BYTE
0000          0024 FILL EQU 0 BYTE TO USE WHEN NOT SENDING
0000          0025 READY EQU 0AAH READY CODE IN V12OUT/V34OUT
0000          0026 PLAY EQU 0BBH PLAYING CODE
0000          0027 DONE EQU 0FFH FINISHED CODE
0000          0028 *
0000          0029 ORG FIT
11E9 C3 F4 11 0030 JMP FITINIT
11EB C3 29 12 0031 JMP FITSMP
11EE C3 A3 12 0032 JMP FITTEL
11F1 C3 0D 12 0033 JMP FITEXEC
11F4          0034 *
11F4 21 B9 22 0035 FITINIT LXI H,FITVARS CLEAR ALL VARS/PARAMS
11F7 0E 07 0036 MVI C,FVEND-FITVARS
11F9 CF 0037 RST ZERO/8
11FA          0038 *
11FA 21 D7 22 0039 LXI H,V12PRM INIT THE V12 PARAMETERS
11FD CD 03 12 0040 CALL PRMINIT
1200 21 FB 22 0041 LXI H,V34PRM AND THE V34 PARAMETERS
1203 11 20 13 0042 PRMINIT LXI D,IFARAM
1206 0E 09 0043 MVI C,IPRMX-IFARAM
1208 D7 0044 RST COPY/8
1209          0045 *
1209 11 18 40 0046 TBLNULL LXI D,NULL*256+TBLNG NULL ALL TABLE ENTRIES
120C 72 0047 CLEAR MOV M,D
120D 23 0048 INX H
120E 1D 0049 DCR E

```

```
120F C2 0C 12 0050 JNZ CLEAR
1212 C9 0051 RET
1213 0052 *
1213 0053 * FIT MODE SETTING COMMAND
1213 0054 *
1213 5F 0055 FITCMD MOV E,A A=E=ENABLE BITS
1214 29 0056 DAD H SHIFT DATA BITS LEFT
1215 A5 0057 ANA L D=ENABLE AND DATA
1216 57 0058 MOV D,A
1217 78 0059 MOV A,E A=FITMODE AND (NOT ENABLE)
1218 EE 22 0060 XRI V12DI+FITDI
121A 21 B6 22 0061 LXI H,FITMODE
121D A6 0062 ANA M AND WITH THE ENABLE BITS
121E 82 0063 DRA D OR WITH ENABLED BITS
121F 77 0064 MOV M,A REPLACE
1220 0065 *
1220 E6 02 0066 ANI V12DI IF CHANGING V12, CLEAR TABLE
1222 00 0067 RNZ .
1223 21 E0 22 0068 LXI H,V12TBL
1226 03 0F 12 0069 JMF TBLNULL (AND RETURN NO CARRY)
1229 0070 *
1229 0071 * SPIN SYNCHRONOUS SAMPLING.
1229 0072 * ON ENTRY: [A]=SUN ANGLE
1229 0073 *
1229 04 A0 0074 FITSMP SUI V1ANG REMOVE SUN SENSOR TO BOOM 1 ANGLE
122B 5F 0075 MOV E,A
122C E6 07 0076 ANI 256/NSAMPS-1 IF NOT TIME, RETURN
122E 00 0077 RNZ .
122F 3A B6 22 0078 LDA FITMODE IF FITS DISABLED, RETURN
1232 E6 29 0079 ANI FITDI
1234 00 0080 RNZ .
1235 0081 *
1235 7B 0082 MOV A,E MANAGE BUFFERS
1236 00 6B 12 0083 CALL FITSYNC
1239 0084 *
1239 3E 0F 0085 MVI A,V34 SAVE A SAMPLE OF V34
123B 00 90 12 0086 CALL GETSAMP
123E 3A B2 22 0087 LDA V34IN
1241 11 30 22 0088 LXI D,V34BUF
1244 00 90 12 0089 CALL STORE
1247 21 B2 22 0090 LXI H,V34IN
124A 00 64 12 0091 CALL INCR
124D 0092 *
124D 3A B6 22 0093 LDA FITMODE IF V12 DISABLED, GO
1250 E6 02 0094 ANI V12DI
1252 00 0095 RNZ .
1253 2E 08 0096 MVI A,V12F SAVE A SAMPLE OF V12
1255 00 90 12 0097 CALL GETSAMP
1258 3A B0 22 0098 LDA V12IN
125B 11 B0 21 0099 LXI D,V12BUF
```

125E CD 9D 12	0100	CALL	STORE	
1261 21 B0 22	0101	LXI	H,V12IN	
1264	0102	*		
1264 7E	0103	INCR	MOV	A,M
1265 C6 02	0104		ADI	2
1267 E6 7F	0105		ANI	7FH
1269 77	0106		MOV	M,A
126A C9	0107		RET	
126B	0108	*		
126B	0109	*	FITSYNC: ON 0 DEGREES FOR EITHER ROOM, SWITCH BUFFERS	
126B	0110	*		
126B FE 80	0111	FITSYNC	CPI	V3ANG-VIANG IF V34 AT 0 DEGREES
126D 21 B2 22	0112		LXI	H,V34IN
1270 CA 7E 12	0113		JZ	FSY1
1273 B7	0114		ORA	A IF V12 AT 0 DEGREES
1274 C0	0115		RNZ	.
1275 3A B6 22	0116		LDA	FITMODE AND NOT DISABLED
1278 E6 02	0117		ANI	V12DI
127A C0	0118		RNZ	.
127B 21 B0 22	0119		LXI	H,V12IN RESET V12OUTPUT
127E	0120	*		
127E 7E	0121	FSY1	MOV	A,M GET THE BUFFER POINTER
127F E6 3F	0122		ANI	NSAMPS*2-1 IF FINISHED WITH BUFFER
1281 CC 89 12	0123		CZ	FBUFOK THEN SET THE OUTPUT POINTER
1284 7E	0124		MOV	A,M GOOD OR NOT, CLEAR THE
1285 E6 C0	0125		ANI	-NSAMPS*2 LSB'S OF THE INDEX
1287 77	0126		MOV	M,A
1288 C9	0127		RET	
1289	0128	*		
1289 7E	0129	FBUFOK	MOV	A,M SET OUTPUT POINTER
128A EE 40	0130		XRI	NSAMPS*2 TO THE OTHER BUFFER
128C 2C	0131		INR	L
128D 77	0132		MOV	M,A
128E 2D	0133		DCR	L
128F C9	0134		RET	
1290	0135	*		
1290 CD 2A 01	0136	BETSAMP	CALL	ABC D=0TY TO SAMPLE
1293 7A	0137		MOV	A,D
1294 CD E6 00	0138		CALL	SAMPLE TAKE IT
1297 7A	0139		MOV	A,D PUT THE GAIN BIT INTO [HL]
1298 E6 10	0140		ANI	HIGAIN
129A B4	0141		ORA	H
129B 67	0142		MOV	H,A
129C C9	0143		RET	
129D	0144	*		
129D EB	0145	STORE	XCHG	. DATA TO DE
129E DF	0146		RST	REF/8
129F 73	0147		MOV	M,E
12A0 23	0148		INX	H
12A1 72	0149		MOV	M,D


```
12A2 C9      0150      RET
12A3        0151 *
12A3        0152 * FIT TELEMETRY OUTPUT.
12A3        0153 * ON EXIT: [A] = DATA OR 0 FILL
12A3        0154 *
12A3 21 B4 22 0155 FITTEL LXI  H,COUNT  IF NO DATA REMAINING
12A6 7E      0156      MOV  A,M
12A7 B7      0157      ORA  A      THEN CHECK IF MORE READY
12A8 CA B3 12 0158      JZ   CHKDONE
12A8 B9      0159      DCR  M      ELSE COUNT--
12AC 21 B5 22 0160      LXI  H,PTR  RETURN THE NEXT DATA ITEM
12AF 34      0161      INR  M
12B0 BE      0162      MOV  L,M
12B1 7E      0163      MOV  A,M
12B2 C9      0164      RET
12B3        0165 *
12B3 31 B1 22 0166 CHKDONE LXI  H,V12OUT  IF OUTPUT STATUS=READY
12B6 16 A1    0167      MVI  D,V12HEADER
12B8 1E B6    0168      MVI  E,V12RES*256/256-1
12BA 7E      0169      MOV  A,M
12BB FE AA    0170      CFI  READY
12BD CA D0 12 0171      JZ   STARPLAY
12C0        0172 *
12C0 31 B3 22 0173      LXI  H,V34OUT
12C3 16 A3    0174      MVI  D,V34HEADER
12C5 1E C6    0175      MVI  E,V34RES*256/256-1
12C7 7E      0176      MOV  A,M
12C8 FE AA    0177      CFI  READY
12CA CA D0 12 0178      JZ   STARPLAY
12CD 3E 00    0179      MVI  A,FILL
12CF C9      0180      RET
12D0        0181 *
12D0 36 FE    0182 STARPLAY MVI  M,DONE  CLEAR THE READY FLAG
12D2 7B      0183      MOV  A,E  SET THE POINTER TO OUTPUT BUFFER
12D3 32 B5 22 0184      STA  PTR
12D6 3E 10    0185      MVI  A,V34RES-V12RES  SET COUNT=LENGTH OF RESULTS
12D8 32 B4 22 0186      STA  COUNT
12DB 7A      0187      MOV  A,D  RETURN HEADER CODE TO BEGIN
12DD C9      0188      RET
12DE        0189 *
12DE        0190 * EXECUTIVE FIT CALCULATOR (FOREGROUND)
12DE        0191 *
12DE 3A B1 22 0192 FITEXE LDA  V12OUT  IF OUTPUT BUFFER IS READY
12E0 B7      0193      ORA  A
12E1 F2 FE 12 0194      JP   FIT12
12E4        0195 *
12E4 3A B3 22 0196      LDA  V34OUT  CHECK V34'S BUFFER TOO
12E7 B7      0197      ORA  A
12E8 FB      0198      RM
12E9        0199 *
```

```

12E9 21 30 22      0200 FIT34 LXI  H,V34BUF
12EC 00 13 13      0201      CALL  STFIT
12EF 01 07 22      0202      LXI  B,V34RES
12F2 11 FB 22      0203      LXI  D,V34PRM
12F5 00 04 08      0204      CALL  SPIN
12F8 3E AA         0205      MVI  A,READY
12FA 32 B3 22      0206      STA  V34OUT
12FD 09           0207      RET
12FE             0208 *
12FE 21 80 21      0209 FIT12 LXI  H,V12BUF
1301 00 13 13      0210      CALL  STFIT [HL]->SAMPLE AREA
1304 01 27 22      0211      LXI  B,V12RES [BC]->RESULTS AREA
1307 11 07 22      0212      LXI  D,V12PRM [DE]->PARAMS AREA
130A 00 04 08      0213      CALL  SPIN
130D 3E AA         0214      MVI  A,READY
130F 32 B1 22      0215      STA  V12OUT
1312 09           0216      RET
1313             0217 *
1313 0F           0218 STFIT RST  REF/8 [HL]->SAMPLES
1314 3A 86 22      0219      LDA  FITMODE IF TRANSMIT ENABLED, DO IT
1317 E6 08         0220      ANI  FITXMIT
1319 08           0221      RZ
131A 11 20 00      0222      LXI  D,NSAMPS [DE]=#SAMPLES
131D 03 0F 13      0223      JMP  ELEXMIT TRANSMIT THEM
1320             0224 *
1320 3B           0225 IPARAM DB  3BH  LO GAIN = 1/50.9 HIGAIN
1321 A0           0226      DB  0A0H
1322 F2           0227      DB  0F2H
1323 41           0228      DB  041H  ALPHA = 1.40
1324 E6           0229      DB  0E6H
1325 66           0230      DB  066H
1326 3F           0231      DB  03FH  BETA = 0.4
1327 0C           0232      DB  0CCH
1328 0D           0233      DB  0CDH
1329             0234 IPRM EQU  $
1329 00           V 0235      DB  256  END OF FIT
132A             0236 *
132A             0237 * ENTER COMMAND VECTOR INTO TABLE
132A             0238 *
132A             0239      ORG  FMODE/4+CMDTAB
0054 13 12        0240      DW  FITCMD
0056             0241 *
0056             0242 * RAM
0056             0243 *
0056             0244      ORG  FITRAM
2180             0245 SMPBUF EQU  $  SAMPLES BUFFER
2180             0246 V12BUF DS  NSAMPS*2*2
2230             0247 V34BUF DS  NSAMPS*2*2
2280             0248 *
2280             0249 FITVARS EQU  $

```

2280	0250	V12IN DS	1	
2281	0251	V12OUT DS	1	
2282	0252	V34IN DS	1	
2283	0253	V34OUT DS	1	
2284	0254	COUNT DS	1	COUNT OF BYTES TO SEND OUT
2285	0255	PTR DS	1	POINTER TO RESULTS
2286	0256	FITMODE DS	1	ENABLE/DISABLE BITS
2287	0257	FVEND EQU	*	
2287	0258	*		
2287	0259	V12RES DS	5*FLT+1	RESULT AREAS
2287	0260	V34RES DS	5*FLT+1	
2287	0261	*		
2287	0262	TBLNG EQU	AVPTS*2*FLT	AHI/ALO TABLE LENGTH
2287	0263	V12PRM DS	3*FLT	PARAMETER AREA
2288	0264	V12TBL DS	TBLNG	AHI/ALO TABLE AREA
2288	0265	V34PRM DS	3*FLT	
2301	0266	V34TBL DS	TBLNG	
2319	0267	*		
2319	0268	* EXTERNALS		
2319	0269	*		
2319	0281	ELEXMIT EQU	ELE+15	

0000	0001 #
0000	0002 # CRRES FLIGHT SOFTWARE --- SAWTOOTH GENERATOR
0000	0003 # WRITTEN BY PETER R HARVEY
0000	0004 #
0000	0005 # FILE SAW.A
0000	0006 #
0000	0007 PSW EQU 6
0000	0008 SANCODE EQU 05SH
0000	0009 #
0000	0010 ORG SAW
1330 C3 3C 13	0011 JMP SAWINIT
1333 C3 5F 13	0012 JMP SAWSTEP
1336	0013 #
1336 21 A4 21	0014 SANDSC LXI H,SAWOFF
1339 DF	0015 RST REF/B
133A 7E	0016 MOV A,M
133B C9	0017 RET
133C	0018 #
133C 11 85 13	0019 SAWINIT LXI D,DEFAULT SET DEFAULTS
133F 21 A4 21	0020 LXI H,SAWOFF
1342 0E 06	0021 MVI C,6
1344 D7	0022 RST COPY/B
1345 C9	0023 RET
1346	0024 #
1346	0025 # PERFORM SAWTOOTH COMMANDS
1346	0026 #
1346 7C	0027 SAWCMD MOV A,H SELECT WHICH REG
1347 E6 07	0028 ANI 7
1349 5D	0029 MOV E,L
134A 21 A4 21	0030 LXI H,SAWOFF REFERENCE OPTIONS
134D DF	0031 RST REF/B
134E 73	0032 MOV M,E STORE
134F C9	0033 RET
1350	0034 #
1350	0035 # SAWTOOTH SYNCHRONIZER
1350	0036 #
1350 21 A0 21	0037 SAWSYNC LXI H,BIASREG LATCH IN
1353 11 A4 21	0038 LXI D,SAWOFF THE COMMANDED VALUES
1356 0E 04	0039 MVI C,4
1358 D7	0040 RST COPY/B
1359 3E 01	0041 MVI A,1 AT NEXT STEP, RESET BIAS
135B 32 A3 21	0042 STA DIVCNT
135E C9	0043 RET
135F	0044 #
135F	0045 # SAWTOOTH STEP
135F	0046 #
135F 3A A5 21	0047 SAWSTEP LDA SAWDEL IF NO DELTA, THEN
1362 B7	0048 ORA A THIS PACKAGE IS DISABLED
1363 C8	0049 RZ .

1364	0050 †		
1364 3A 1D 20	0051	LDA	FRAME ON FRAME 30, LAST LINE
1367 FE FE	0052	CPI	-2 PERFORM A SYNC
1369 C2 74 13	0053	JNZ	SSOFT
136C 3A 1C 20	0054	LDA	WORD
136F FE E0	0055	CPI	224
1371 D4 50 13	0056	CNC	SAWSYNC
1374	0057 †		
1374 3A A8 21	0058 SSOFT	LDA	OPTIONS IF STEPPING DISABLED, QUIT
1377 0F	0059	RRC	
1378 D0	0060	RNC	
1379	0061 †		
1379 21 A3 21	0062 STEP	LXI	H, DIVCNT DIVIDE STEPS
137C 35	0063	DCR	M
137D C0	0064	RNZ	
137E 3A A7 21	0065	LDA	SAWDIV THEN RELOAD
1381 77	0066	MOV	M, A
1382	0067 †		
1382 3A A8 21	0068	LDA	OPTIONS IF BIASING ALLOWED
1385 E6 02	0069	ANI	2 DO IT
1387 C4 A5 13	0070	CNZ	BIASEM
138A	0071 †		
138A 21 A2 21	0072	LXI	H, PERCNT
138D 35	0073	DCR	M
138E CA 9A 13	0074	JZ	FLIP
1391	0075 †		
1391 3A A1 21	0076	LDA	DELREG BIASREG += DELREG
1394 21 A0 21	0077	LXI	H, BIASREG
1397 86	0078	ADD	M
1398 77	0079	MOV	M, A
1399 C9	0080	RET	
139A	0081 †		
139A 3A A6 21	0082 FLIP	LDA	SAWPER RESET PERIOD REGISTER
139D 77	0083	MOV	M, A
139E 21 A1 21	0084	LXI	H, DELREG DELREG = -DELREG
13A1 97	0085	SUB	A
13A2 96	0086	SUB	M
13A3 77	0087	MOV	M, A
13A4 C9	0088	RET	
13A5	0089 †		
13A5 2A A0 21	0090 BIASEM	LHLD	BIASREG
13A8 3A A9 21	0091	LDA	SENSOR
13AB 67	0092	MOV	H, A
13AC E5	0093	PUSH	H
13AD CD 63 01	0094	CALL	SETBIAS
13B0 E1	0095	POP	H
13B1 24	0096	INR	H
13B2 C3 63 01	0097	JMP	SETBIAS
13B5	0098 †		
13B5	0099 †		DEFAULT SETTINGS FOR SAWTOOTH

13B5		0100	↓		
13B5 00		0101	DEFALT DB	0	OFFSET
13B6 02		0102	DB	2	DELTA (PKG ENABLE TOO)
13B7 40		0103	DB	64	# STEPS UP THEN DOWN
13B8 02		0104	DB	2	64/N HZ STEPS
13B9 00		0105	DB	0	STEPPING/BIASING DISABLED
13BA 01		0106	DB	1	SENSORS 1 AND 2 GET SAWTOOTH
13BB 00	V	0107	DB	256	SAWTOOTH END
13BC		0108	↓		
13BC		0109	↓		ENTER COMMAND VECTOR INTO TABLE
13BC		0110	↓		
13BC		0111	ORG		SAWCODE/4+CMDTAB
0058 46 13		0112	DW		SAWCMD
0058		0113	↓		
0058		0114	↓		SAWTOOTH VARIABLES
0058		0115	↓		
0058		0116	ORG		SAWRAM
21A0		0117	BIASREG DS	1	CURRENT BIAS VALUE
21A1		0118	DELREG DS	1	CURRENT DELTA
21A2		0119	PERCENT DS	1	PERIOD COUNTER
21A3		0120	DIVCNT DS	1	DIVIDER COUNTER
21A4		0121	↓		
21A4		0122	SAWOFF DS	1	SAWTOOTH OFFSET
21A5		0123	SAWDEL DS	1	DELTA (IF ZERO, PKG DISABLE)
21A6		0124	SAWPER DS	1	#STEPS UP
21A7		0125	SAWDIV DS	1	DIVIDER OF CALLS
21A8		0126	OPTION DS	1BS (B=BIAS ENABLE, S=STEP ENABLE)
21A9		0127	SENSOR DS	1	WHICH BIAS PAIR (1 OR 3)
21AA		0128	SPARE DS	2	
21AC		0129	↓		
21AC		0130	ORG		BYGRAM ACCESS SYSTEM CLOCK
201C		0131	WORD DS	1	
201D		0132	FRAME DS	1	

```

0000          0001 *
0000          0002 * CRRES FLIGHT PROGRAM---ELECTRIC FIELD MANAGEMENT
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 *
0000          0005 * FILE : ELE.A
0000          0006 *
0000          0007 PSW   EQU   6      9085 SPECIFIC INFORMATION
0000          0008 SP    EQU   6
0000          0009 *
0000          0010 * FAST DIGITAL MONITOR DEFINITION
0000          0011 *
0000          0012 MBITS EQU   00FH  MAIN STATUS
0000          0013 BBITS EQU   080H  BURST STATUS
0000          0014 TESTFLAG EQU 40H  TEST/CAL MODE BIT
0000          0015 MODEFLAG EQU   I   VOLTAGE/CURRENT MODE BIT
0000          0016 PLAYFLAG EQU  80H  PLAYBACK ENABLE BIT
0000          0017 MBYMIT EQU   8    MAIN TRANSMIT OVERRIDE
0000          0018 *
0000          0019 MUX   EQU  2800H  MUX SETTING CMD
0000          0020 RESET EQU  3000H  RESET RELAY CMD
0000          0021 SET   EQU  RESET+100H
0000          0022 *
0000          0023     DRG   ELE
0000          0024     JMP   ELEINIT  INITIALIZATION
0000          0025     JMP   ELEFRAME  MINOR FRAME SYNC
0000          0026     JMP   ELESAMP   SAMPLE TIME
0000          0027     JMP   ELETELEM  TELEMETRY TIME
0000          0028     JMP   ELEDSC   DIGITAL SUBCDM TIME
0000          0029     JMP   ELEXMIT  REQUEST MAIN PLAYBACK
0000          0030 *
0000          0031 ELESTAT LDR  FDM   RETURN FAST STATUS
0000          0032     RET
0000          0033 *
0000          0034 * INITIALIZE THE ELECTRIC FIELD PACKAGE
0000          0035 *
0000          0036 ELEINIT LXI  H,ELELRAM
0000          0037     MVI   C,ELELND-ELELRAM
0000          0038     RST   ZERO/8
0000          0039     RET
0000          0040 *
0000          0041 * PERFORM ELECTRIC FIELD CATEGORY COMMANDS
0000          0042 *
0000          0043 SMCMD MOV   A,H    A=MUX NUMBER
0000          0044     ANI   7      CHECK IF IT'S AN ELE
0000          0045     CPI   7      CONTROLLED REGISTER.
0000          0046     CMC   .      IF NOT 0-6, ERROR.
0000          0047     RC    .
0000          0048     JMP   SETMUX  L=VALUE ALREADY
0000          0049 *
1300 03 06 13
1303 03 08 14
1306 03 0A 15
1309 03 0C 15
130C 03 0E 15
130F 03 10 16
1312
1315 03 12 20
1318 03
131B
131E 21 50 20
1321 0E A5
1324 CF
1327 C9
132A
132D
1330 7C
1333 E6 07
1336 FE 07
1339 3F
133C 08
133F 03 CC 02
1342

```

13E7 7C	0050 RLCMD	MOV	A,H	CRY=1 TD SET RELAY
13E8 0F	0051	RRC	.	
13E9 7D	0052	MDV	A,L	A=RELAY NUMBER TD FL1P
13EA C3 88 01	0053	JMP		SETRELAY
13ED	0054 *			
13ED 7C	0055 SETRB	MDV	A,H	SAVE 3 BITS OF MS BYTE
13EE E6 07	0056	ANI	7	
13F0 F6 20	0057	ORI		ELERAM/2048*2048/256 TAKE 5 MSB'S
13F2 67	0058	MDV	H,A	DF THE RAM ADDRESSING
13F3 22 50 20	0059	SHLD		RAMBASE SET THE RAM BASE ADDRESS
13F6 C9	0060	RET		
13F7	0061 *			
13F7 C6 3E	0062 SETMD	ADI		TESTFLAG-2 MDVE TEST FLAG DVER
13F9 32 94 20	0063	STA		FDMNEXT
13FC 0F	0064	RRC	.	
13FD 11 0A 14	0065	LXI	D,	VMLIST
1400 B2 06 14	0066	JNC		SM1
1403 11 1C 14	0067	LXI	D,	IMLIST
1406 3E 01	0068 SM1	MVI	A,1	ASK FOR INTERNAL BATCH CMDS
1408 E7	0069	RST	4	
1409 C9	0070	RET		
140A	0071 *			
140A 12 31	0072 VMLIST	DW		SET+18
140C 13 31	0073	DW		SET+19
140E 07 30	0074	DW		RESET+7
1410 08 30	0075	DW		RESET+8
1412 09 30	0076	DW		RESET+9
1414 10 30	0077	DW		RESET+16
1416 01 28	0078	DW		MUX+0+1
1418 01 29	0079	DW		MUX+100H+1
141A FF FF	0080	DW		-1
141C	0081 *			
141C 07 31	0082 INLIST	DW		SET+7
141E 08 31	0083	DW		SET+8
1420 09 31	0084	DW		SET+9
1422 10 31	0085	DW		SET+16
1424 12 30	0086	DW		RESET+18
1426 13 30	0087	DW		RESET+19
1428 00 28	0088	DW		MUX+0+0
142A 00 29	0089	DW		MUX+100H+0
142C FF FF	0090	DW		-1
142E	0091 *			
142E 7C	0092 SETFOR	MDV	A,H	IF BTY TYPE CMD, GD
142F E6 04	0093	ANI	4	
1431 C2 70 14	0094	JNZ		SETFB
1434 3E 01	0095	MVI	A,1	IF VDLTAGE ENABLED,
1436 11 52 20	0096	LXI	D,	VHXPTR EFFECT VDLT POINTERS
1439 CD 41 14	0097	CALL		SETFM
143C 3E 02	0098	MVI	A,2	IF CURRENT ENABLED,
143E 11 54 20	0099	LXI	D,	IHXPTR EFFECT CURR POINTERS

1441		0100 *		
1441 A4	0101	SETFM	ANA H	IF NOT ENABLED, RETURN
1442 CB	0102	RZ	.	
1443 7D	0103	MOV	A,L	GET LEFT NIBBLE OF DATA
1444 0F	0104	RRC	.	FOR THE 1ST POINTER
1445 0F	0105	RRC		
1446 0F	0106	RRC		
1447 0F	0107	RRC		
1448 CD 50 14	0108	CALL	SF2	
1448 7B	0109	MOV	A,E	MOVE TO LX POINTERS
144C C6 04	0110	ADI	VLXPTR-VHXPTR	
144E 5F	0111	MOV	E,A	
144F 7D	0112	MOV	A,L	GET RIGHT NIBBLE FOR LX'S
1450	0113 *			
1450 E5	0114	SF2	PUSH H	SAVE 1NST COMMAND
1451 CD 5D 14	0115	CALL	ADRFMT	HL= &FORMAT(A)
1454 EB	0116	XCHG	.	
1455 73	0117	MOV	M,E	STORE THE ADDRESS INTO HX/LX PTR
1456 23	0118	INX	H	
1457 72	0119	MOV	M,D	
1458 2B	0120	DCX	H	
1459 EB	0121	XCHG		
145A E1	0122	POP	H	
145B 87	0123	ORA	A	
145C C9	0124	RET		
145D	0125 *			
145D	0126 *	ADDRESS THE	FORMAT(A)	
145D	0127 *			
145D 21 3D 16	0128	ADRFMT	LXI H,ROMFOR	POINT AT ROM FORMATS
1460 E6 0F	0129	ANI	15	
1462 FE 0A	0130	CPI	10	IF FORMAT#10 THRU #15 THEN
1464 DA 6A 14	0131	JC	ADRF1	
1467 21 F5 1F	0132	LXI	H,RAMFOR-160	USE RAM FORMATS
146A 87	0133	ADRF1	ADD A	EACH FORMAT IS 16 BYTES
146B 87	0134	ADD	A	
146C 87	0135	ADD	A	
146D 87	0136	ADD	A	
146E DF	0137	RST	REF/B	HL=HL+A
146F C9	0138	RET		
1470	0139 *			
1470 7C	0140	SETFQ	MOV A,H	IF EVEN, THEN QTY INDEX=L
1471 0F	0141	RRC	.	
1472 7D	0142	MOV	A,L	
1473 D2 83 14	0143	JNC	SETQX	
1476	0144 *			
1476 5D	0145	SETQV	MOV E,L	
1477 3A 5B 20	0146	LDA	TINDEX	RAMFOR(TINDEX)=L
147A 21 95 20	0147	LXI	H,RAMFOR	
147D DF	0148	RST	REF/B	
147E 73	0149	MOV	M,E	

147F 3A 5B 20	0150	LDA	TMINDEX	TMINDEX++
14B2 3C	0151	INR	A	
14B3 FE 60	0152	SETDX	CPI	96
14B5 D0	0153	RNC		
14B6 32 5B 20	0154	STA	TMINDEX	
14B9 B7	0155	DRA	A	RETURN(NC)
14BA C9	0156	RET		
14BB	0157	‡		
14BB	0158	‡	MINOR FRAME SYNC.	
14BB	0159	‡	ON ENTRY: A=MINOR FRAME NUMBER	
14BB	0160	‡		
14BB 21 7E 20	0161	ELEFRAME	LX1	H,BUFF1 PICK UP ADDRESSES
148E 11 6A 20	0162	LXI	D,BUFF0	OF BOTH BUFFERS
1491 0F	0163	RRC	.	IF ODD FRAME, GO
1492 DA A5 14	0164	JC	ODDFRAME	
1495	0165	‡		
1495	0166	‡	EVEN FRAME	
1495	0167	‡		
1495 EB	0168	XCHG	.	SET THE BUFFER POINTERS
1496 CD E0 14	0169	CALL	SETBUF	
1499 CD EE 11	0170	CALL	FITTEL	GET THE SPIN FIT
149C 32 BB 20	0171	STA	SFR	ASYNCHRONOUS DATA
149F 3E FF	0172	MVI	A,-1	RESET THE SAMPLE COUNTER
14A1 32 5A 20	0173	STA	SMPCNT	
14A4 C9	0174	RET	.	
14A5	0175	‡		
14A5	0176	‡	ODD FRAME	
14A5	0177	‡		
14A5 CD E0 14	0178	ODDFRAME	CALL	SETBUF SET THE BUFFERS
14AB 3A 1D 20	0179	LDA	FRAME	
14AB 3C	0180	INR	A	IF FRAME=7 THEN
14AC E6 07	0181	ANI	7	RESET THE LX POINTER
14AE CC 09 15	0182	CZ	RSTLX	
14B1 CD 13 15	0183	CALL	RSTHX	RESET HX EVERY ODD FRAME
14B4	0184	‡		
14B4	0185	‡	CALCULATE THE FAST DIGITAL MONITOR	
14B4	0186	‡		
14B4 97	0187	SUB	A	GET BURST FDM BITS FROM
14B5 CD 11 10	0188	CALL	BURDSC	THE BURST CONTROLLER
14BB E6 E0	0189	ANI	BBITS	
14BA 5F	0190	MOV	E,A	
14BB	0191	‡		
14BB CD EA 17	0192	CALL	SWPSTAT	PUT SWEEP STATUS IN L
14BE E6 01	0193	ANI	I	
14C0 6F	0194	MOV	L,A	
14C1	0195	‡		
14C1 3E 02	0196	MVI	A,2	GET THE COMMAND COUNT BIT
14C3 E5	0197	PUSH	H	
14C4 E7	0198	RST	4	
14C5 E1	0199	POP	H	

```

14C6 29      0200      DAD   H      SHIFT 1T 1N
14C7 DF      0201      RST   REF/B
14CB         0202 *
14CB 3A 94 20 0203      LDA   FDMNEXT GET THE I/V MODE COMMANDED
14CB E6 41    0204      ANI   MODEFLAG+TESTFLAG
14CD 29      0205      DAD   H
14CE DF      0206      RST   REF/B
14CF B3      0207      ORA   E      PUT THESE WITH BURST BITS
14D0 5F      0208      MOV   E,A
14D1         0209 *
14D1 2A 5E 20 0210      LHLD  XMTCNT IF THE TRANSMIT COUNT!=0
14D4 7C      0211      MOV   A,H
14D5 B5      0212      ORA   L
14D6 CA DB 14 0213      JZ    ORFDM
14D9 3E 88    0214      MVI   A,PLAYFLAG+MBXMIT TURN ON XMIT
14DB B3      0215 ORFDM ORA   E
14DC 32 77 20 0216      STA   FDM
14DF C9      0217      RET
14E0         0218 *
14E0         0219 * SET THE BUFFER POINTERS. (SAMPLE & TELEM)
14E0         0220 * ON ENTRY: [HL]=BUFFER FOR TELEM
14E0         0221 * [DE]=BUFFER FOR SAMPLING
14E0         0222 *
14E0 22 68 20 0223 SETBUF SHLD  TMPTR
14E3 EB      0224      XCHG .      AND RESETTING THE
14E4 22 64 20 0225      SHLD  HBPTR HIGH AND LOW SAMPLE
14E7 11 0E 00 0226      LXI   D,LSAMP POINTERS TO THE
14EA 19      0227      DAD   D      OTHER BUFFER.
14EB 22 66 20 0228      SHLD  LBPTR
14EE         0229 *
14EE         0230 * PLACE THE GAIN BITS INTO THE TM BUFFER
14EE         0231 *
14EE 2A 68 20 0232      LHLD  TMPTR PLACE THE GAINS IN
14F1 11 0C 00 0233      LXI   D,12 INTO THE BUFFER AT
14F4 19      0234      DAD   D      AN OFFSET OF 12
14F5 3A 92 20 0235      LDA   HGAINS
14FB 77      0236      MOV   M,A
14F9         0237 *
14F9 11 07 00 0238      LXI   D,7 THEN PUT IN THE LOW
14FC 19      0239      DAD   D      GAINS
14FD 7E      0240      MOV   A,M
14FE E6 F0    0241      ANI   OFOH
1500 5F      0242      MOV   E,A
1501 3A 93 20 0243      LDA   LGAINS
1504 E6 0F    0244      ANI   OFH
1506 B3      0245      ORA   E
1507 77      0246      MOV   M,A
1508 C9      0247      RET
1509         0248 *
1509         0249 * RESET THE HIGH AND LOW SAMPLE LIST POINTERS

```

1509	0250	*	DEPENDING UPON THE CURRENT MODE (E-FIELD OR LANGMUIR)
1509	0251	*	
1509 21 56 20	0252	RSTLX LXI	H,VLXPTR POINT AT THE PAIR
150C CD 1D 15	0253	CALL	INDEX OF LISTS AND CHOOSE
150F 22 62 20	0254	SHLD	LQPTR ACC'ING TO MODE
1512 C9	0255	RET	.
1513	0256	*	
1513 21 52 20	0257	RSTHX LXI	H,VHXPTR SAME FOR THE HIGH
1516 CD 1D 15	0258	CALL	INDEX
1519 22 60 20	0259	SHLD	HQPTR
151C C9	0260	RET	.
151D	0261	*	
151D 3A 77 20	0262	INDEX LDA	FDM ADD 0 OR 2
1520 E6 01	0263	ANI	MODEFLAG TO THE POINTER
1522 1E 02	0264	IND1 MVI	E,2
1524 C2 28 15	0265	JNZ	IND2
1527 5F	0266	MOV	E,A DEPENDING UPON THE
1528 16 00	0267	IND2 MVI	D,0 MODE SETTING.
152A 19	0268	DAD	D
152B	0269	*	
152B 7E	0270	MOV	A,M PICK UP THE POINTER
152C 23	0271	INX	H
152D 66	0272	MOV	H,M
152E 6F	0273	MOV	L,A
152F 2B	0274	DCX	H AND SUBTRACT THE 1ST
1530 C9	0275	RET	.
1531	0276	*	
1531	0277	*	E-FIELD TELEMETRY OUTPUT ROUTINES.
1531	0278	*	
1531 2A 68 20	0279	ELETEL LHL	D TMPTR [HL]->CURRENT DATA
1534 5E	0280	MOV	E,M PICK UP 1 BYTE
1535 23	0281	INX	H
1536 87	0282	ORA	A IF 1 BYTE ONLY
1537 CA 3D 15	0283	JZ	FINTEL
153A 53	0284	MOV	D,E ELSE GRAB ANOTHER
1538 5E	0285	MOV	E,M
153C 23	0286	INX	H
153D 22 68 20	0287	FINTEL SHLD	TMPTR
1540 E8	0288	XCHG	. PUT RESULT IN [HL]
1541 C9	0289	RET	.
1542	0290	*	
1542	0291	*	OUTPUT ELECTRIC FIELD DIGITAL SUBCOM
1542	0292	*	ON ENTRY: A= INDEX INTO ELE DIG SUB COM
1542	0293	*	
1542 FE 10	0294	ELEDSC CPI	16 FROM 0 TO 15 ARE THE
1544 21 52 20	0295	LXI	H,VHXPTR HIGH RATE QTY LIST
1547 DA 54 15	0296	JC	EDINX
154A D6 10	0297	SUI	16
154C FE 20	0298	CPI	32 FROM 16 TO 47 ARE
154E 21 56 20	0299	LXI	H,VLXPTR THE LOW QTY LIST

1551 D2 62 15	0300	JNC	EDVAR	
1554 F5	0301	EDINX	PUSH	PSW INDEX TO GET HL
1555 3A 1E 20	0302	LDA	CYCLE	PUT OUT I'S IN ODD CYCLES
1558 E6 01	0303	ANI	1	
155A CD 22 15	0304	CALL	IND1	
155D F1	0305	POP	PSW	ADDRESS OF QTY LIST
155E 23	0306	INX	H	
155F C3 67 15	0307	JMP	ADDA	
1562 D6 20	0308	EDVAR	SUI	32
1564 21 50 20	0309	LXI	H,RAMBASE	FROM 48 ARE JUST VARS
1567 DF	0310	ADDA	RST	REF/B
1568 7E	0311	MOV	A,M	
1569 C9	0312	RET		
156A	0313	*		
156A	0314	*	E-FIELD SAMPLING ROUTINE.	
156A	0315	*		
156A 21 5A 20	0316	ELESAMP	LXI	H,SMPCNT SAMPLE COUNT++
156D 34	0317	INR	M	
156E 7E	0318	MOV	A,M	
156F 0F	0319	RRC	.	IF EVEN, SAMPLE HX
1570 DA 92 15	0320	JC	LXBUR	ELSE SAMPLE LX
1573	0321	*		
1573 2A 60 20	0322	HXSAMP	LHLD	HQPTR INCREMENT LIST POINTER
1576 23	0323	INX	H	
1577 22 60 20	0324	SHLD	HQPTR	
157A CD DA 15	0325	CALL	GETQTY	[DE]=QUANTITY(MEM[HL])
157D	0326	*		
157D 21 92 20	0327	LXI	H,HGAINS	STORE THE GAIN
1580 CD B9 15	0328	CALL	STOGAIN	
1583	0329	*		
1583 2A 64 20	0330	LHLD	HBPTR	STORE [DE] IN BUFFER
1586 3A 5A 20	0331	LDA	SMPCNT	LEFT OR RIGHT ADJUST
1589 E6 02	0332	ANI	2	DEPENDING UPON COUNT
158B CD C4 15	0333	CALL	STOQTY	
158E 22 64 20	0334	SHLD	HBPTR	
1591 C9	0335	RET	.	
1592	0336	*		
1592 3A 5A 20	0337	LXBUR	LDA	SMPCNT
1595 E6 02	0338	ANI	2	
1597 CA 0B 10	0339	JZ	BURSAMP	
159A	0340	*		
159A 2A 62 20	0341	LXSAMP	LHLD	LQPTR INCREMENT LOW LIST
159D 23	0342	INX	H	
159E 22 62 20	0343	SHLD	LQPTR	
15A1 CD DA 15	0344	CALL	GETQTY	SAMPLE THE QUANTITY
15A4 21 93 20	0345	LXI	H,LGAINS	STORE THE GAIN
15A7 CD B9 15	0346	CALL	STOGAIN	
15AA 2A 66 20	0347	LHLD	LBPTR	STORE IN LOW BUFFER
15AD 3A 5A 20	0348	LDA	SMPCNT	LEFT OR RIGHT ADJUSTED
1580 E6 04	0349	ANI	4	

15B2 CD C4 15	0350	CALL	STOQTY	
15B5 22 66 20	0351	SHLD	LBPTR	
15B8 C9	0352	RET	.	
15B?	0353	†		
15B9	0354	†	STORE GAIN BIT FROM [DE] INTO [M]	
15B9	0355	†		
15B9 7A	0356	STOGAIN	MOV A,D	IF GAIN=0, STORE 0
15BA E6 10	0357	ANI	10H	BY RESETTING CARRY
15BC CA C0 15	0358	JZ	STG0	
15BF 37	0359	STC	.	ELSE SET CARRY
15C0 7E	0360	STG0	MOV A,M	MOVE CRY INTO LSB
15C1 17	0361	RAL	.	
15C2 77	0362	MOV	M,A	
15C3 C9	0363	RET	.	
15C4	0364	†		
15C4	0365	†	STORE A QUANTITY INTO THE BUFFER AT [HL]	
15C4	0366	†		
15C4 C2 D1 15	0367	STOQTY	JNZ ODD	
15C7 EB	0368	EVEN	XCHG .	STORE THE EVEN
15C8 29	0369	DAD	H	BY SHIFTING THE
15C9 29	0370	DAD	H	12 BITS TO THE LEFT
15CA 29	0371	DAD	H	
15CB 29	0372	DAD	H	
15CC EB	0373	XCHG	.	
15CD 72	0374	MOV	M,D	AND STORING
15CE 23	0375	INX	H	THE 12 BITS WITH
15CF 73	0376	MOV	M,E	ZERO FOLLOWING
15D0 C9	0377	RET	.	LEAVE HL->BYTE
15D1	0378	†		
15D1 7A	0379	ODD	MOV A,D	STORE THE ODD
15D2 E6 0F	0380	ANI	0FH	BY PUTTING THE
15D4 B6	0381	DRA	M	MSB'S DOWN FIRST
15D5 77	0382	MOV	M,A	
15D6 23	0383	INX	H	AND THEN THE
15D7 73	0384	MOV	M,E	LSB'S
15D8 23	0385	INX	H	
15D9 C9	0386	RET	.	
15DA	0387	†		
15DA	0388	†	GET A QUANTITY.	
15DA	0389	†	ON ENTRY: [M] IS THE QUANTITY DESCRIPTOR.	
15DA	0390	†	ON EXIT : [DE]= THE 13 BIT VALUE OF THAT QTY	
15DA	0391	†		
15DA 3A 77 20	0392	GETQTY	LDA FDM	IF PLAYBACK MODE ENABLED
15DD A6	0393	ANA	M	AND THE QTY IS ENABLED,
15DE FA 0A 16	0394	JM	GETPLAY	THEN GET PLAYBACK
15E1	0395	†		
15E1 7E	0396	MOV	A,M	IF A RAM QUANTITY,
15E2 E6 40	0397	ANI	40H	THEN GET IT
15E4 C2 FF 15	0398	JNZ	GETRAM	
15E7	0399	†		

```

15E7          0400 * GET AN ANALOG QUANTITY.
15E7          0401 *
15E7 7E      0402 GETANA MOV   A,M   DETERMINE GAIN IN D
15E8 E6 3F   0403          ANI   3FH
15EA CD 2A 01 0404          CALL  AGC
15ED 1E 10   0405          MVI   E,10H SET E TO RECORD HIGAIN/LOGAIN
15EF C2 F4 15 0406          JNZ   GAI
15F2 1E 00   0407          MVI   E,0
15F4          0408 *
15F4 7A      0409 GAI   MOV   A,D   SAMPLE QTY(D)
15F5 CD E6 00 0410          CALL  SAMPLE [HL]=ATOD(A)
15F8 7B      0411          MOV   A,E   PUT IN GAINBIT
15F9 E6 10   0412          ANI   10H
15FB B4      0413          ORA   H
15FC 57      0414          MOV   D,A   AND RETURN(DE)
15FD 5D      0415          MOV   E,L
15FE C9      0416          RET   .
15FF          0417 *
15FF 7E      0418 GETRAM MOV   A,M   ADD THE OFFSET
1600 E6 3F   0419          ANI   03FH FROM THE DESCRIPTOR
1602 2A 50 20 0420          LHLD  RAMBASE
1605 DF      0421          RST   REF/S
1606 5E      0422          MOV   E,M   LOAD TWO BYTES
1607 23      0423          INX   H
1608 56      0424          MOV   D,M
1609 C9      0425          RET   .
160A          0426 *
160A          0427 * GET BURST PLAYBACK DATA
160A          0428 *
160A 3A 77 20 0429 GETPLAY LDA   FDM   IF MAIN XMIT, DO IT
160D E6 08   0430          ANI   MBXMIT
160F C2 17 16 0431          JNZ   GPMAIN
1612 CD 0E 10 0432          CALL  BURPLAY [HL]=BURST PLAYBACK INFO
1615 EB      0433          XCHG  .   PUT IN DE
1616 C9      0434          RET   .
1617          0435 *
1617 2A 5E 20 0436 GPMAIN LHLD  XMTCNT DECREASE COUNT
161A 7C      0437          MOV   A,H   IF ZERO, RETURN(0)
161B B5      0438          ORA   L
161C EB      0439          XCHG  .
161D C8      0440          RZ   .
161E 1B      0441          DCX  D
161F EB      0442          XCHG  .
1620 22 5E 20 0443          SHLD  XMTCNT
1623 2A 5C 20 0444          LHLD  XMTPTR THEN PICK UP 13 BITS
1626 5E      0445          MOV   E,M
1627 23      0446          INX   H
1628 56      0447          MOV   D,M
1629 23      0448          INX   H   AND UPDATE POINTER
162A 22 5C 20 0449          SHLD  XMTPTR

```

```

162D C9      0450      RET
162E        0451 †
162E        0452 † REQUEST TRANSMISSION.
162E        0453 † ON ENTRY: [HLI]->RAM AREA, [EJ]=COUNT OF SAMPLES
162E        0454 †
162E 3A 77 20 0455 ELEXMIT LDA FDM IF FDM XMIT GOING, RETURN
1631 E6 0B    0456      ANI MBXMIT
1633 37      0457      STC
1634 C0      0458      RNZ
1635 22 5C 20 0459      SHLD XMPTR SET POINTER
1638 EB      0460      XCHG .
1639 22 5E 20 0461      SHLD XMCNT AND COUNTER
163C C9      0462      RET .
163D        0463 †
163D        0464 † TELEMETRY TABLE DEFAULTS
163D        0465 †
163D        0466 BZ EQU 0 MULTIPLEXOR ADDRESSES
163D        0467 BY EQU 1
163D        0468 BX EQU 2
163D        0469 V3 EQU 3
163D        0470 V2 EQU 4
163D        0471 V1SC EQU 5
163D        0472 V1 EQU 6
163D        0473 AGCF EQU 7
163D        0474 V12F EQU 8
163D        0475 F3 EQU 9
163D        0476 F2 EQU 10
163D        0477 F1 EQU 11
163D        0478 V4 EQU 12
163D        0479 AGCU EQU 13
163D        0480 V12U EQU 14
163D        0481 V34 EQU 15
163D        0482 †
163D        0483 PE EQU PLAYFLAG
163D        0484 ROMFOR EQU $ ROMFORMATS
163D        0485 HXTABLE EQU $
163D 0B      0486      DB V12F
163E 0F      0487      DB V34
163F 8B      0488      DB V12F+PE
1640 BF      0489      DB V34+PE
1641 0B      0490      DB V12F
1642 0F      0491      DB V34
1643 8B      0492      DB V12F+PE
1644 BF      0493      DB V34+PE
1645 0B      0494      DB V12F
1646 0F      0495      DB V34
1647 8B      0496      DB V12F+PE
1648 BF      0497      DB V34+PE
1649 0B      0498      DB V12F
164A 0F      0499      DB V34

```


1648 88	0500	DB	V12F+PE
164C BF	0501	DB	V34+PE
164D	0502	*	
164D	0503	LXTABLE EQU	*
164D 86	0504	DB	V1+PE
164E 83	0505	DB	V3+PE
164F 87	0506	DB	ABCF+PE
1650 08	0507	DB	F1
1651 84	0508	DB	V2+PE
1652 8C	0509	DB	V4+PE
1653 87	0510	DB	ABCF+PE
1654 09	0511	DB	F3
1655 86	0512	DB	V1+PE
1656 83	0513	DB	V3+PE
1657 87	0514	DB	ABCF+PE
1658 0A	0515	DB	F2
1659 84	0516	DB	V2+PE
165A 8C	0517	DB	V4+PE
165B 87	0518	DB	ABCF+PE
165C 09	0519	DB	F3
165D 86	0520	DB	V1+PE
165E 83	0521	DB	V3+PE
165F 87	0522	DB	ABCF+PE
1660 08	0523	DB	F1
1661 84	0524	DB	V2+PE
1662 8C	0525	DB	V4+PE
1663 87	0526	DB	ABCF+PE
1664 09	0527	DB	F3
1665 86	0528	DB	V1+PE
1666 83	0529	DB	V3+PE
1667 87	0530	DB	ABCF+PE
1668 0A	0531	DB	F2
1669 84	0532	DB	V2+PE
166A 8C	0533	DB	V4+PE
166B 87	0534	DB	ABCF+PE
166C 09	0535	DB	F3
166D 00	V 0536	DB	256 ELE END
166E	0537	*	
166E	0538	*	ENTER COMMAND VECTORS INTO TABLE
166E	0539	*	
166E	0540	ORG	CMDTAB
0040 63 01	0541	DW	SETBIAS #0
0042 80 01	0542	DW	SETSTUB #8
0044 76 01	0543	DW	SETGUARD #10
0046 98 02	0544	DW	SETVTRIM #18
0048 AE 02	0545	DW	SETFILTER #20
004A DD 13	0546	DW	SMCMD #28
004C E7 13	0547	DW	RLCMD #30
004E	0548	*	
004E	0549	ORG	40H/4+CMDTAB

0050 2E 14	0550	DW	SETFOR #40
0052 ED 13	0551	DW	SETRB #48
0054	0552	‡	
0054	0553	ORG	68H/4+CMDTAB
005A F7 13	0554	DW	SETMD #68
005C	0555	‡	
005C	0556	‡	RAM SECTION
005C	0557	‡	
005C	0558	ORG	ELERAM
2050	0559	RAMBASE DS	2 RAM VARIABLES BASE POINTER
2052	0560	‡	
2052	0561	VHXPTR DS	2 SAMPLE LIST POINTERS
2054	0562	IHXPTR DS	2 FOR HIGH AND LOW
2056	0563	VLXPTR DS	2
2058	0564	ILXPTR DS	2
205A	0565	SMPCNT DS	1 SAMPLE COUNT
205B	0566	‡	
2058	0567	TMINDEX DS	1 INDEX INTO RAM FORMAT
205C	0568	XMPTR DS	2 TRANSMIT POINTER
205E	0569	XMCNT DS	2 TRANSMIT COUNTER
2060	0570	‡	
2060	0571	HQPTR DS	2 SAMPLE LIST POINTERS
2062	0572	LQPTR DS	2 FOR THE PRESENT MODE.
2064	0573	HQPTR DS	2 BUFFER POINTERS FOR
2066	0574	LQPTR DS	2 THE PRESENT MODE
2068	0575	TMPTR DS	2 TELEMETRY BUFFER POINTER.
206A	0576	‡	
206A	0577	BUFF0 DS	12+1+1+6 TELEMETRY BUFFER 0
207E	0578	FDM EQU	BUFF0+13 FAST DIGITAL MONITOR
207E	0579	BUFF1 DS	12+1+1+6 TELEMETRY BUFFER 1
2092	0580	SFR EQU	BUFF1+13 SPIN-FIT RESULTS
2092	0581	LSAMP EQU	12+1+1 OFFSET BETWEEN HX AND LX
2092	0582	‡	
2092	0583	HGAINS DS	1 HIGH GAIN BITS
2093	0584	LGAINS DS	1 LOW GAIN BITS
2094	0585	FDMNEXT DS	1 FAST DIGITAL MONITOR (NEXT)
2095	0586	RAMFOR DS	48*2
20F5	0587	ELEND EQU	‡
20F5	0588	‡	
20F5	0589	‡	EXTERNALS
20F5	0590	‡	
20F5	0591	ORG	BUR
1008	0592	BURINIT DS	3
1008	0593	BURSAMP DS	3
100E	0594	BURPLAY DS	3
1011	0595	BURDSC DS	3
1014	0596	‡	
1014	0597	ORG	FIT
11E8	0598	FITINI DS	3
11EB	0599	FITSMP DS	3

11EE	0600	FITTEL DS	3
11F1	0601	†	
11F1	0602	ORG	SWP
17E4	0603	SWPINIT DS	3
17E7	0604	SWPANG DS	3
17EA	0605	SWPSTAT DS	3
17ED	0606	†	
17ED	0607	ORG	BKGRAM
201C	0608	WORD DS	1
201D	0609	FRAME DS	1
201E	0610	CYCLE DS	1

0000		0001	⋆
0000		0002	⋆ CRRES FLIGHT SDFWARE---MAIN PRDGRAM LOADER
0000		0003	⋆ FILE : LD.A
0000		0004	⋆
0000		0005	LDCODE EQU OEBH CMDMND NUMBER (5 BITS)
0000		0006	⋆
0000		0007	DRG LD
16B0		0008	⋆
16B0	21 30 29	0009	LDINIT LXI H,USER PDINT THE ADR REGISTER
16B3	22 20 29	0010	SHLD ADR TD THE USER LDADING AREA
16B6	C9	0011	RET .
16B7		0012	⋆
16B7	7C	0013	LDCMD MDV A,H GET THE CMDMND AGAIN
16B8	D6 EB	0014	SUI LDCDDE REMDVE THE BIAS
16BA	CA 9B 16	0015	JZ SADRL AND CDUNT DFF EACH NUMBER
16BD	3D	0016	DCR A
16BE	CA A0 16	0017	JZ SADRH
1691	3D	0018	DCR A
1692	CA A5 16	0019	JZ LOAD
1695	3D	0020	DCR A
1696	CA AF 16	0021	JZ JUMP
1699	37	0022	STC . IF UNKNKDN, RETURN(CRY)
169A	C9	0023	RET
169B		0024	⋆
169B	7D	0025	SADRL MDV A,L SET LOW ADDRESS
169C	32 20 29	0026	STA ADR
169F	C9	0027	RET
16A0		0028	⋆
16A0	7D	0029	SADRH MDV A,L SET HIGH ADDRESS
16A1	32 21 29	0030	STA ADR+1
16A4	C9	0031	RET .
16A5		0032	⋆
16A5	EB	0033	LDAD XCHG .
16A6	2A 20 29	0034	LHLD ADR MEM[ADR++] = VALUE
16A9	73	0035	MDV M,E
16AA	23	0036	INX H
16AB	22 20 29	0037	SHLD ADR
16AE	C9	0038	RET .
16AF		0039	⋆
16AF	3A 30 29	0040	JUMP LDA USER EXECUTE USER PRDGRAM
16B2	B5	0041	ADD L CHECK CDDE PLUS COMMAND
16B3	FE AA	0042	CPI OAAH IS THE RIGHT VALUE
16B5	C0	0043	RNZ . IF NOT RIGHT, SIGNAL ERROR
16B6	97	0044	SUB A RESET THE CDDE
16B7	32 30 29	0045	STA USER
16B8	C3 31 29	0046	JMP USER+1
16BD	00	0047	DB 256 END OF LD MODULE
16BE		0048	⋆
16BE		0049	⋆ ENTER COMMAND VECTOR INTO TABLE

16BE	0050 †			
1&BE	0051	ORG	LDCODE/4+CMDTAB	
007A 87 16	0052	DW	LDCMD	
007C	0053 †			
007C	0054 †	VARIABLES		
007C	0055 †			
007C	0056	ORG	LDRAM	
2920	0057 ADR	DS	2	USER LOAD ADDRESS
2922	0058	DS	14	
2930	0059 USER	DS	3FOH	USER PROGRAM LOADING AREA

```

0000      0001 †
0000      0002 † CRRES FLIGHT SOFTWARE --- DEPLOYMENT MODULE
0000      0003 † WRITTEN BY PETER R HARVEY
0000      0004 †
0000      0005 † FILE: DEP.A
0000      0006 †
0000      0007 PSH   EQU   6
0000      0008 †
0000      0009 DEPLOY EQU   0DOH  COMMAND NUMBER
0000      0010 DEPOVER EQU  DEPLOY+4  DEPLOY SWITCH OVERRIDE
0000      0011 †
0000      0012 LCOVER EQU   1      BOOM STATUS BITS
0000      0013 TURNS EQU   2
0000      0014 RCOVER EQU   4
0000      0015 ENDWIRE EQU  8
0000      0016 †
0000      0017 † INTERNAL CODES FOR DEPLOY STATUS
0000      0018 †
0000      0019 CMDED EQU   1      MOTOR IS COMMANDED TO RUN
0000      0020 ACTUAL EQU  2      MOTOR IS RUNNING
0000      0021 †
0000      0022 MOFF  EQU   0      NEITHER CMDED NOR RUNNING
0000      0023 MPAUSE EQU  CMDED  COMMANDED BUT PAUSED
0000      0024 MSTOP EQU  ACTUAL  RUNNING BUT SHOULD STOP
0000      0025 MRUN  EQU  ACTUAL+CMDED  RUNNING AS COMMANDED
0000      0026 †
0000      0027 † CODE ENTRY POINTS
0000      0028 †
0000      0029      ORG   DEP
16C4 C3 D0 16 0030      JMP   DEPINIT
16C7 C3 0B 17 0031      JMP   DEPSAMP
16CA      0032 †
16CA      0033 † DIGITAL SUB COMMUTATOR (STATUS)
16CA      0034 †
16CA 21 44 20 0035 DEPDSC LX1  H,DEPSTAT RETURN DEPSTAT ON
16CD DF      0036      RST  REF/B FOR THE DIGITAL STATUS
16CE 7E      0037      MOV  A,M
16CF C9      0038      RET
16D0      0039 †
16D0      0040 † INITIALIZATION. TURN OFF BOTH ROOMS
16D0      0041 †
16D0 CD E3 00 0042 DEPINIT CALL  ROOMSTAT
16D3 32 45 20 0043      STA  BOOMBITS
16D6 21 00 D0 0044      LX1  H,DEPLOY+256+0
16D9 CD 02 17 0045      CALL DOVER CLEAR OVERRIDE
16DC      0046 †
16DC      0047 † ACCEPT DEPLOY COMMANDS
16DC      0048 † ON ENTRY: [HL]= COMMAND
16DC      0049 † ON EXIT: CARRY SET IF NOT A DEPLOYMENT COMMAND

```

```

16DC          0050 *
16DC 7C      0051 DEPCMD MOV  A,H   CHECK THE INCOMING COMMAND
16DD FE D4   0052     CPI  DEPOVER  OVERRIDE SWITCHES COMMAND?
16DF CA 02 17 0053     JZ   DOVER
16E2 E6 FC   0054     ANI  OFCH
16E4 D6 D0   0055     SUI  DEPLOY  IF DEPLOY START/STOP, GO
16E6 37      0056     STC  .      RETURN CARRY IF NOT US
16E7 C0      0057     RNZ  .
16EB          0058 *
16EB 7D      0059     MOV  A,L   SET NEW LIMIT
16E9 32 4B 20 0060     STA  LIMIT
16EC          0061 *
16EC 7C      0062     MOV  A,H   LOOK UP WHAT TO DO WITH MOTORS
16ED E6 03   0063     ANI  3
16EF 21 07 17 0064     LXI  H,DEPTAB
16F2 DF      0065     RST  REF/8
16F3 7E      0066     MOV  A,M
16F4 32 44 20 0067     STA  DEPSTAT
16F7          0068 *
16F7 97      0069     SUB  A
16F8 32 4A 20 0070     STA  DEPCNT  DEPCNT=0
16F8 32 46 20 0071     STA  DLEN1  ZERO DEPLOYED LENGTH COUNTS
16FE 32 47 20 0072     STA  DLEN2
1701 C9      0073     RET
1702          0074 *
1702 7D      0075 DOVER MOV  A,L   SET OVERRIDE BITS
1703 32 49 20 0076     STA  OVERRIDE
1706 C9      0077     RET
1707          0078 *
1707          0079 * 2-BIT CODE TO STATE TABLE
1707          0080 *
1707 22      0081 DEPTAB DB   MSTOP*11H  TURN OFF BOTH MOTORS
1708 21      0082     DB   MSTOP*10H+MPAUSE  TURN ON MOTOR 1
1709 12      0083     DB   MPAUSE*10H+MSTOP  TURN ON MOTOR 2
170A 11      0084     DB   MPAUSE*11H  TURN ON BOTH MOTORS
170B          0085 *
170B          0086 * DEPLOY SAMPLING.  MONITOR LENGTHS/CURRENTS OF MOTORS
170B          0087 *           TURN ON/OFF MOTORS AS COMMANDED.
170B          0088 *
170B D5      0089 DEPSAMP PUSH D   SAVE [DE] IN INTERRUPT
170C 21 4A 20 0090     LXI  H,DEPCNT  UPDATE DEPLOY SAMPLE CNT
170F 34      0091     INR  M      WHICH CHANGES THE MOTOR
1710 CD 68 17 0092     CALL  SMPLNG  SAMPLE LENGTHS
1713 CD 18 17 0093     CALL  DEPMON  CHECK BOOM DEPLOYMENT
1716 D1      0094     POP  D
1717 C9      0095     RET
1718          0096 *
1718          0097 * MONITOR BOOM DEPLOYMENT
1718          0098 *
1718 CD 93 17 0099 DEPMON CALL  GETSTAT  GET CURRENT MOTOR STATE
  
```

```
1718 FE 00      0100      CPI      MOFF  IF OFF, QUIT
171D CB         0101      RZ      .
171E FE 01      0102      CPI      MPAUSE IF MOTOR PAUSED, SEE IF
1720 CA 38 17   0103      JI      TRYGO  IT CAN GO NOW
1723 FE 02      0104      CPI      MSTOP  IF MOTOR SHOULD STOP, DO IT
1725 CA 55 17   0105      JI      STOP
1728           0106 *
1728           0107 * MOTOR IS COMMANDED AND RUNNING
1728           0108 *
1728 CD AB 17   0109 KEEPGO CALL LIMCHK CHECK LIMITS
1728 D2 55 17   0110      JNC     STOP  IF OVER LIMIT, STOP NOW
172E CD CB 17   0111      CALL   CMPLNG IF LENGTH < OTHER, OK
1731 08         0112      RC      .
1732 FE 03      0113      CPI      3      IF LENGTH-OTHER > 3, PAUSE
1734 02 5A 17   0114      JNC     PAUSE
1737 C9         0115      RET
1738           0116 *
1738           0117 * TRY TO GO ONCE PAUSED
1738           0118 *
1738 CD AB 17   0119 TRYGO CALL LIMCHK IF BOOM AT LIMIT, DON'T
1738 3E 00      0120      MVI     A,MOFF EVEN START IT.
173D D2 B2 17   0121      JNC     SETSTAT
1740 CD CB 17   0122      CALL   CMPLNG IF THIS BOOM IS SHORTER THAN
1743 DA 47 17   0123      JC      RUN   THE OTHER OK
1746 C0         0124      RMT     .      IF EQUAL OK
1747           0125 *
1747           0126 * MOTOR CONTROL: A=NEW STATE FOR THE MOTOR
1747           0127 *
1747 3E 03      0128 RUN  MVI     A,MRUN SET THE STATE INFORMATION
1749 CD B2 17   0129      CALL   SETSTAT
174C 3A 4A 20   0130      LDA     DEPCNT GET THE MTR BIT
174F E6 01      0131      ANI     1
1751 37         0132      STC     .      AND TURN ON THE MOTOR
1752 C3 18 03   0133      JMP     SETMOTOR
1755           0134 *
1755 3E 00      0135 STOP  MVI     A,MOFF
1757 C3 5C 17   0136      JMP     MTROFF
175A 3E 01      0137 PAUSE MVI     A,MPAUSE
175C CD B2 17   0138 MTROFF CALL SETSTAT
175F 3A 4A 20   0139      LDA     DEPCNT GET MTR NUMBER
1762 E6 01      0140      ANI     1
1764 B7         0141      ORA     A      CLEAR CARRY FOR OFF
1765 C3 18 03   0142      JMP     SETMOTOR
1768           0143 *
1768           0144 * SAMPLE THE LENGTHS OF THE BOOMS
1768           0145 *
1768 21 45 20   0146 SMPLENG LXI H,BOOMBITS SAVE OLD BITS
176B 5E         0147      MOV     E,M
176C CD E3 00   0148      CALL   BOOMSTAT GET NEW STATUS
176F 77         0149      MOV     M,A
```



```
1770          0150 *
1770 A8       0151      XRA   E      GET THE CHANGES
1771 A6       0152      ANA   M      WHICH ARE POSITIVE
1772 5F       0153      MOV   E,A
1773          0154 *
1773 3E 02    0155      MVI   A,TURNS IF 8DOM1 TURNS CTR
1775 21 46 20 0156      LXI   H,DLEN1 IS A 1, INCREMENT DLEN1
1778 CD 7E 17 0157      CALL  TRNCHK
1778          0158 *
1778 3E 20    0159      MVI   A,TURNS*16 CHECK BOOM2 AS WELL
177D 23       0160      INX   H
177E A3       0161 TRNCHK ANA   E      IF 8IT IS 0, QUIT
177F C8       0162      RZ    .
1780 3A       0163      INR   M      ELSE INCREMENT LENGTH
1781 C9       0164      RET
1782          0165 *
1782          0166 * SET/GET STATUS OF CURRENT MOTOR
1782          0167 *
1782 6F       0168 SETSTAT MOV L,A  SAVE NEW 2-8IT CODE
1783 3A 44 20 0169      LDA   DEPSTAT GET THE OTHER SIDE
1786 CD 9F 17 0170      CALL  SWAP
1789 E6 30    0171      ANI   3*16  SAVE THEM
1788 85       0172      ORA   L      PUT IN NEW 8ITS
178C CD 9F 17 0173      CALL  SWAP  REORIENT
178F 32 44 20 0174      STA   DEPSTAT AND SAVE
1792 C9       0175      RET
1793          0176 *
1793 3A 44 20 0177 GETSTAT LDA DEPSTAT GET CURRENT STATUS
1796 CD 9F 17 0178      CALL  SWAP
1799 E6 03    0179      ANI   3
179B C9       0180      RET
179C          0181 *
179C 3A 45 20 0182 SWITCHES LDA BOOMBITS GET THE RIGHT SWITCHES
179F          0183 *
179F          0184 * SWAP REVERSES THE NIBBLES IN A BYTE IF MOTOR 2
179F          0185 *
179F 5F       0186 SWAP MOV E,A  ADJUSTNPUT
17A0 3A 4A 20 0187      LDA   DEPCNT TEST WHICH MOTOR
17A3 0F       0188      RRC   .
17A4 78       0189      MOV   A,E  RETURN SAME IF MOTOR 1
17A5 D0       0190      RNC   .
17A6 0F       0191      RRC   .  ELSE GET HIGH NIBBLE
17A7 0F       0192      RRC   .
17A8 0F       0193      RRC   .
17A9 0F       0194      RRC   .
17AA C9       0195      RET
17AB          0196 *
17AB          0197 * LIMIT CHECK THE CURRENT BOOM
17AB          0198 *
17AB CD 9C 17 0199 LIMCHK CALL SWITCH GET THE MICROS
```

17AE 21 49 20	0200	LX1	H, OVERRIDE	ADD IN THE OVERRIDE BITS
17B1 B6	0201	ORA	M	TO DEFEAT AN ERRANT SWITCH
17B2 6F	0202	MOV	L, A	
17B3 E6 01	0203	ANI	LCDVER	TEST THE COVERS
17B5 C8	0204	RZ	.	IF CLOSED, RETURN NC
17B6 7D	0205	MOV	A, L	
17B7 E6 04	0206	ANI	RCOVER	
17B9 C8	0207	RZ	.	
17BA 7D	0208	MOV	A, L	
17B8 E6 88	0209	ANI	ENDWIRE*11H	IF END-OF-WIRE ON EITHER BOOM,
17BD EE 88	0210	XRI	ENDWIRE*11H	RETURN NO CARRY
17BF C0	0211	RNZ	.	
17C0	0212 *			
17C0 CD D6 17	0213	CALL	REFLEN	GET BOOM LENGTH
17C3 21 48 20	0214	LX1	H, LIMIT	AND COMPARE TO THE LIMIT
17C6 BE	0215	CMP	M	
17C7 C9	0216	RET		
17C8	0217 *			
17C8	0218 *			COMPARE BOOM LENGTHS IF BOTH BOOMS COMMANDED
17C8	0219 *			IF ONLY 1 COMMANDED, RETURN CARRY
17C8	0220 *			IF BOTH COMMANDED, RETURN THIS-THAT BOOM LENGTH
17C8	0221 *			
17C8 3A 44 20	0222	CMPLENG LDA	DEPSTAT	CHECK FOR BOTH COMMANDED
17C8 E6 11	0223	ANI	CMDED*11H	
17CD FE 11	0224	CPI	CMDED*11H	
17CF 37	0225	STC		
17D0 C0	0226	RNZ	.	
17D1 CD D6 17	0227	CALL	REFLEN	[A]=THIS BOOM, [L]=OTHER
17D4 95	0228	SUB	L	
17D5 C9	0229	RET		
17D6	0230 *			
17D6	0231 *			REFERENCE BOTH BOOM LENGTHS.
17D6	0232 *			[A]=THIS BOOM, [L]=THAT BOOM.
17D6	0233 *			
17D6 2A 46 20	0234	REFLEN LHLD	DLEN1	H=DLEN2, L=DLEN1
17D9 3A 4A 20	0235	LDA	DEPCNT	
17DC 0F	0236	RRC	.	
17DD 7C	0237	MOV	A, H	IF MOTOR 2, RETURN(LEN2, LEN1)
17DE D8	0238	RC	.	
17DF 7D	0239	MOV	A, L	ELSE RETURN(LEN1, LEN2)
17E0 6C	0240	MOV	L, H	
17E1 C9	0241	RET		
17E2 00	V 0242	DB	256	END OF DEPLOYMENT
17E3	0243 *			
17E3	0244 *			ENTER COMMAND VECTOR INTO TABLE
17E3	0245 *			
17E3	0246	ORG	DEPLOY/4+CMDTAB	
0074 DC 16	0247	DW	DEPCMD	
0076	0248 *			
0076	0249 *			VARIABLES

0076	0250 *			
0076	0251	ORG	DEPRAM	
2044	0252	DEPSTAT DS	1	DEPLOY STATE [CODE1:CODE0]
2045	0253	BOOMBITS DS	1	BOOM STATUS BITS
2046	0254	DLEN1 DS	1	LENGTH OF BOOM 1
2047	0255	DLEN2 DS	1	LENGTH OF BOOM 2
2048	0256	LIMIT DS	1	BOOM LENGTH UPPER LIMIT
2049	0257	OVERRIDE DS	1	SWITCH OVERRIDE BITS
204A	0258	DEPCNT DS	1	DEPLOY COUNTER (MTR IN LSB)

```

0000      0001 *
0000      0002 * CRRES FLIGHT PROGRAM --- BIAS SWEEPS
0000      0003 *
0000      0004 * FILE SWP.A
0000      0005 *
0000      0006 SWPCODE EQU 60H  COMMAND CODE FOR SWEEP MODULE
0000      0007 SWPR8 EQU SWPCODE+2  COMMAND CODE FOR REBIASING
0000      0008 V1SET EQU 68H  COMMAND CODE FOR ELE MODE SET
0000      0009 SAWENA EQU 05CH  COMMAND CODE FOR SAWTOOTH OPTIONS
0000      0010 XMTCODE EQU 0EE0H  SWEEEEP XMIT CODE
0000      0011 *
0000      0012 V1ANG EQU 128+32  SUN ANGLE WHEN BOOM1 SUNWARD
0000      0013 V3ANG EQU V1ANG+64  AND BOOM3 SUNWARD
0000      0014 V1FIT EQU V1ANG  V12 FITS START WHEN V1 SUNWARD
0000      0015 V3FIT EQU V1FIT+128  V34 FITS ARE OFF 180 DEGREES
0000      0016 V1SWP EQU V1ANG-64-8  SWEEP AT 112 DEGREES BEFORE SUN
0000      0017 V3SWP EQU V3ANG-64-8
0000      0018 *
0000      0019 V1 EQU 6  MUX ADDRESSES
0000      0020 V2F EQU 4
0000      0021 R11 EQU 8
0000      0022 R12 EQU V2F
0000      0023 V3 EQU 3
0000      0024 V4 EQU 12
0000      0025 *
0000      0026 PSW EQU 6
0000      0027 SP EQU 6
0000      0028 SRHL EQU 10H  SHIFT RIGHT OPCODE
0000      0029 *
0000      0030 ORG SWP
17E4 C3 04 18 0031 JMP SWPINIT
17E7 C3 31 18 0032 JMP SWPANG
17EA C3 FD 17 0033 JMP SWPSTAT
17ED C3 A2 18 0034 JMP SWPEXEC
17F0      0035 *
17F0      0036 * SWEEP DIGITAL STATUS
17F0      0037 * RETURNS A=OPTION[BOOM,A]
17F0      0038 *
17F0 FE 06 0039 SWPDSC CP1 6 1F 0-5, RETURN RAM12
17F2 DA F7 17 0040 JC SD1
17F5 C6 09 0041 ADI RAM34-RAM12-6  ELSE RAM34
17F7 21 C3 24 0042 SD1 LX1 H,RAM12
17FA DF 0043 RST REF/B
17FB 7E 0044 MOV A,M
17FC C9 0045 RET
17FD      0046 *
17FD      0047 * SWEEP FAST DIGITAL STATUS
17FD      0048 *
17FD 3A E2 24 0049 SWPSTAT LDA SWPREQ  RETURN LSB=1 FOR SWEEPING
  
```

```

1800 32 E3 24      0050      STA      SWPOK (SYNC TO WHEN THIS CALL IS MADE)
1803 C9           0051      RET
1804             0052 *
1804             0053 * INITIALIZATION/DEFAULT SETTINGS
1804             0054 *
1804 21 C1 24      0055 SWPINIT LXI  H,STATE COPY FROM ROM TO RAM
1807 11 2D 18      0056      LXI      D,ROMDEF
180A 0E 11         0057      MVI      C,RAM34-RAM12+2
180C D7           0058      RST      COPY/8
180D 11 2F 18      0059      LXI      D,ROMDEF+2
1810 0E 0F         0060      MVI      C,RAM34-RAM12
1812 D7           0061      RST      COPY/8
1813 36 01         0062      MVI      M,1 BOOM = 1
1815 3E 98         0063      MVI      A,V3SWP#256/256 CHANGE SWEEP ANGLE OF V34
1817 32 D3 24      0064      STA      RAM34+ANGSWP
181A C9           0065      RET
1818             0066 *
1818             0067 * COMMAND ENTRY IN [HL]
1818             0068 *
1818 7C           0069 SWPCMD MOV  A,H IF REBIAS COMMAND, SETRESULT(L)
181C FE 62         0070      CPI      SWPRB
181E CA 7F 18      0071      JZ       REBIAS
1821 E8           0072      XCHG   . [DE]=COMMAND
1822 21 C0 24      0073      LXI      H,INDEX [HL]->INDEX
1825 0F           0074      RRC      . IF EVEN, SET INDEX
1826 D2 2F 18      0075      JNC      SI
1829 34           0076      INR      M ELSE SET VALUE(INDEX++)
182A 7E           0077      MOV      A,M
182B FE 22         0078      CPI      BOOM-INDEX+1
182D D0           0079      RNC
182E DF           0080      RST      REF/8
182F 73           0081 S1 MOV  M,E
1830 C9           0082      RET
1831             0083 *
1831             0084 * SWEEP SUN ANGLE
1831             0085 *
1831 5F           0086 SWPANG MOV  E,A SAVE SUN ANGLE
1832 87           0087      ORA      A COUNT ZERO CROSSINGS
1833 CC 67 18      0088      CZ       DNCNT
1836             0089 *
1836 01 C1 24      0090      LXI      B,STATE IN SEARCH STATE,
1839 0A           0091      LDAX   B CHECK ANGLE FOR START OF SWEEP
183A E6 03         0092      ANI      3
183C CA 6F 18      0093      JZ       CHKANG
183F D6 02         0094      SUI      2 IF ANALYSIS NOT DONE, QUIT
1841 C0           0095      RNZ      .
1842             0096 *
1842 CD 02 18      0097      CALL   REFCON IS IT TIME TO SET BIAS?
1845 23           0098      INX      H
1846 7E           0099      MOV      A,M (CONC1)==ANGLE
  
```

1847 93	0100	SUB	E	
1848 C0	0101	RNZ	.	NO:RETURN
1849 02	0102	STAX	B	YES, STATE=0 (SEARCH AGAIN)
184A	0103	*		
184A 0E 10	0104	MVI	C,10H	CHECK IF THIS MODE IS OK
184C CD E2 1A	0105	CALL	CHKMTD	FOR SETTING BIASES
184F 2A E1 24	0106	LHLD	BOOM	
1852 C4 85 18	0107	CNZ	SETRES	YES. SET BIASES(BOOM)
1855	0108	*		
1855 21 E1 24	0109	SWITCH LXI	H,BOOM	SWITCH TO OTHER BOOM
1858 35	0110	DCR	M	IF 1, SET TO 3 AND VICE VERSA
1859 36 03	0111	MVI	M,3	
1858 CA 60 18	0112	JZ	SETHAX	
185E 36 01	0113	MVI	M,1	
1860	0114	*		
1860 3A C2 24	0115	SETMAX LDA	SPINMAX	SET DELAY IN #SPINS
1863 32 E4 24	0116	STA	SPINCNT	
1866 C9	0117	RET		
1867	0118	*		
1867 21 E4 24	0119	DNCNT LXI	H,SPINCNT	COUNT DOWN
186A 35	0120	DCR	M	
186B F0	0121	RP	.	IF ZERO OR GREATER, OK
186C 36 00	0122	MVI	M,0	
186E C9	0123	RET		
186F	0124	*		
186F 3A E4 24	0125	CHKANG LDA	SPINCNT	IF SPIN COUNTER=0
1872 87	0126	ORA	A	
1873 C0	0127	RNZ	.	
1874 3E 01	0128	MVI	A,ANGSWP	AND THE ANGLE=SWPANG
1876 CD FC 1A	0129	CALL	GETOPT	
1879 BB	0130	CMP	E	
187A C0	0131	RNZ	.	
187B 3E 01	0132	MVI	A,1	SET READY TO SWEEP
187D 02	0133	STAX	B	
187E C9	0134	RET		
187F	0135	*		
187F 3A C1 24	0136	REBIAS LDA	STATE	IF NOT DISABLED
1882 FE 03	0137	CPI	3	THEN SET RESULT(BOOM=L) ON SENSORS
1884 C8	0138	RZ	.	
1885	0139	*		
1885	0140	*		SET THE RESULT ON THE SENSORS OF BOOM[L]
1885	0141	*		
1885 55	0142	SETRES MOV	D,L	
1886 0E 10	0143	MVI	C,10H	CHECK WHETHER TO USE RESULT
1888 CD DC 1A	0144	CALL	CHKENA	OR THE ALTERNATE VALUES
188B 3E 03	0145	MVI	A,ALT	
188D CA 92 18	0146	JZ	SR1	
1890 3E 04	0147	MVI	A,RESULT	
1892 6A	0148	SR1 MOV	L,D	HL->VARIABLES(BOOM L)
1893 CD F3 1A	0149	CALL	REFL	

1896 DF	0150	RST	REF/B	L=ALTERNATE OR RESULT
1897 6E	0151	MOV	L,M	
1898 62	0152	MOV	H,D	
1899	0153	*		
1899	0154	*	BIAS A SENSOR PAIR	
1899	0155	*	ON ENTRY: [H] = 1 OR 3, [L]=VALUE	
1899	0156	*		
1899 E5	0157	BIBOTH	PUSH	H
189A CD 63 01	0158	CALL	SETBIAS	
189D E1	0159	POP	H	
189E 24	0160	INR	H	
189F C3 63 01	0161	JMP	SETBIAS	
18A2	0162	*		
18A2	0163	*	SWEEP EXECUTIVE SECTION	
18A2	0164	*		
18A2 3A C1 24	0165	SMPEXEC	LDA	STATE IF SWEEP ANGLE NOT MATCHED
18A5 3D	0166	DCR	A	RETURN.
18A6 C0	0167	RNZ	.	
18A7	0168	*		
18A7 21 10 25	0169	LX1	H,SWPBUF	SET WHERE DATA GOES
18AA 22 EE 24	0170	SHLD	SWPPTR	
18AD	0171	*		
18AD 3E 04	0172	MVI	A,4	REQUEST SAWTOOTH MODE
18AF CD 36 13	0173	CALL	SAWOSC	
18B2 32 14 29	0174	STA	SAWMODE	
18B5	0175	*		
18B5	0176	*	VOLTAGE PHASE (DO A CURRENT SWEEP)	
18B5	0177	*		
18B5 01 01 07	0178	LX1	B,1STEP*256+1	B=1STEP,C=1
18B8 CD DA 18	0179	CALL	SWEEP	
18B8	0180	*		
18B8	0181	*	CURRENT PHASE (DO A VOLTAGE SWEEP)	
18B8	0182	*		
18B8 01 04 09	0183	LXI	B,VSTEP*256+4	B=VSTEP,C=4
18BE CD DA 18	0184	CALL	SWEEP	
18C1	0185	*		
18C1 97	0186	SUB	A	REMOVE REQUEST
18C2 32 E2 24	0187	STA	SWPREQ	
18C5	0188	*		
18C5 2A 14 29	0189	LHLD	SAWMODE	RESTORE THE OLD SAWTOOTH MODE
18C8 26 5C	0190	MVI	H,SAWENA	
18CA F7	0191	RST	6	
18CB	0192	*		
18CB 0E 01	0193	MVI	C,1	IF THERE WAS A VOLTAGE PHASE
18CD CD DC 1A	0194	CALL	CHKENA	THEN ANALYZE THE DATA
18D0 C4 87 19	0195	CNZ	ANALYZE	
18D3	0196	*		
18D3 21 C1 24	0197	LX1	H,STATE	APPLY ANALYSIS DONE.
18D6 34	0198	INR	H	
18D7 C3 A5 1A	0199	JMP	TRANSMIT	

18DA	0200 †		
18DA	0201 †	PERFORM A SWEEP ON THE SELECTED BOOM SYSTEM IF ENABLED	
18DA	0202 †	ON ENTRY: [B]= VSTEP OR 1STEP INDEX	
18DA	0203 †	[C]= 1 FOR V PHASE, 4 FOR 1 PHASE	
18DA	0204 †		
18DA CD DC 1A	0205 SWEEP	CALL	CHKENA CHECK IF SWEEP ENABLED
18DD C8	0206	RZ	
18DE	0207 †		
18DE 21 E2 24	0208	LXI	H,SWPREQ SET REQUEST=1
18E1 36 01	0209	MVI	M,1 SET REQUEST=1
18E3 23	0210	INX	H
18E4 7E	0211 SWPWT	MOV	A,M WAIT FOR THE OK
18E5 3D	0212	DCR	A
18E6 C2 E4 18	0213	JNZ	SWPWT
18E9	0214 †		
18E9 CD E2 1A	0215	CALL	CHKWTD CHECK IF WE'LL NEED TO FLIP
18EC 32 15 29	0216	STA	NEEDFLIP
18EF	0217 †		
18EF 78	0218	MOV	A,B GET 1STEP/VSTEP INDEX
18F0 CD FC 1A	0219	CALL	GETOPT [HL]->1STEP OR VSTEP
18F3 CD 1D 18	0220	CALL	LD2
18F6 22 E9 24	0221	SHLD	SWPDEL (DEL AND BIAS)
18F9	0222 †		
18F9 CD 02 18	0223	CALL	REFCON GET THE SWPPAIR
18FC 79	0224	MOV	A,C AND QTYA/8 FROM TABLE
18FD 3C	0225	INR	A (INDEX=2 OR 5)
18FE DF	0226	RST	REF/8
18FF E8	0227	XCHG	
1900 21 EB 24	0228	LXI	H,SWPPAIR
1903 0E 03	0229	MVI	C,3
1905 D7	0230	RST	COPY/8
1906	0231 †		
1906 21 01 5C	0232	LXI	H,SAMENA*256+1 TURN OFF SAWTOOTH
1909 F7	0233	RST	6 BUT ALLOW IT TO KEEP TIME
190A	0234 †		
190A CD 25 19	0235	CALL	FLIP FLIP RELAYS IF WE HAVE TO.
190D 01 80 04	0236	LXI	B,400H+128 WAIT 4 TIMES AT FIRST
1910 CD 4C 19	0237 SWPLP	CALL	BIASEM
1913 CD 3D 19	0238	CALL	WAIT
1916 CD 58 19	0239	CALL	ISAMP
1919 06 01	0240	MVI	B,1 WAIT ONCE AFTER THAT
1918 0D	0241	DCR	C
191C C2 10 19	0242	JNZ	SWPLP
191F 2A E8 24	0243	LHLD	SWPPAIR PUT PRIOR RESULT FOR THE
1922 CD 85 18	0244	CALL	SETRES SWEPT BOOM SYSTEM BACK OUT
1925	0245 †		
1925 3A 15 29	0246 FLIP	LDA	NEEDFLIP DO WE NEED TO FLIP?
1928 87	0247	ORA	A
1929 C8	0248	RZ	. NO. RETURN.
192A CD D2 13	0249	CALL	ELESTAT YES. INVERT THE MODE

192D 3C	0250	INR	A	
192E E6 01	0251	ANI	1	
1930 6F	0252	MOV	L,A	
1931 26 6B	0253	MVI	H,VISET	
1933 F7	0254	RST	6	
1934 C9	0255	RET		
1935	0256 †			
1935 3A 1C 20	0257 WAITN	LDA	WORD	
193B E6 07	0258	ANI	7	
193A C2 35 19	0259	JNZ	WAITN	
193D	0260 †			
193D 3A 1C 20	0261 WAIT	LDA	WORD	SAMPLE IN WORDS
1940 D6 02	0262	SUI	2	2,10,1B...
1942 E6 07	0263	ANI	7	
1944 C2 3D 19	0264	JNZ	WAIT	
1947 05	0265	DCR	R	
194B C2 35 19	0266	JNZ	WAITN	
194B C9	0267	RET		
194C	0268 †			
194C 2A EA 24	0269 BIASM	LHLD	SMPBIAS	BIAS SENSORS
194F CD 99 1B	0270	CALL	BIBOTH	
1952	0271 †			
1952 3A E9 24	0272 STEPB	LDA	SMPDEL	BIAS += DELTA
1955 21 EA 24	0273	LXI	H,SMPBIAS	
195B B6	0274	ADD	M	
1959 77	0275	MOV	M,A	
195A C9	0276	RET		
195B	0277 †			
195B 3A EC 24	0278 ISAMP	LDA	QTYA	SAMPLE/STORE QTYA
195E CD 64 19	0279	CALL	SAMSTO	
1961 3A ED 24	0280	LDA	QTYB	
1964 CD 7B 19	0281 SAMSTO	CALL	SSAMP	
1967 29	0282	DAD	H	SIGN EXTEND 12 TO 16 BITS
196B 29	0283	DAD	H	(SHIFTING LEFT, THEN RIGHT)
1969 29	0284	DAD	H	
196A 29	0285	DAD	H	
196B 10	0286	DB	SRHL	
196C 10	0287	DB	SRHL	
196D 10	0288	DB	SRHL	
196E 10	0289	DB	SRHL	
196F	0290 †			
196F EB	0291	XCHG	.	TO [DE]
1970 2A EE 24	0292	LHLD	SWPTR	PUT AWAY
1973 73	0293	MOV	M,E	
1974 23	0294	INX	H	
1975 72	0295	MOV	M,D	
1976 23	0296	INX	H	
1977 22 EE 24	0297	SHLD	SWPTR	
197A C9	0298	RET		
197B	0299 †			

197B B7	0300 SSAMP	ORA	A	IF POSITIVE MUX QTY
197C F2 E6 00	0301	JP	SAMPLE	TAKE A NORMAL SAMPLE
197F 2F	0302	CMA	.	INVERT MUX QTY
1980 3C	0303	INR	A	
1981 CD E6 00	0304	CALL	SAMPLE ELSE	TAKE SAMPLE
1984 C3 8B 00	0305	JMP	NEG16	AND INVERT
1987	0306	*		
1987	0307	*	ANALYZE	SWEEPS
1987	0308	*		
1987 3A E5 24	0309	ANALYZE LDA	ANAVECT	IF USER VECTOR
198A FE AA	0310	CPI	OAAH	IS SET, THEN GO
198C CA E6 24	0311	JZ	ANAVECT+1	
198F	0312	*		
198F 3E 02	0313	MVI	A,ALGNO	SET ALGORITHM * 0
1991 CD FC 1A	0314	CALL	GETOPT	
1994 36 00	0315	MVI	M,0	
1996	0316	*		
1996 3E 0B	0317	MVI	A,PTSREJ	ZERO THE * POINTS REJECTED
1998 CD FC 1A	0318	CALL	GETOPT	
199B 36 00	0319	MVI	M,0	
199D	0320	*		
199D 21 10 25	0321	LXI	H,SWPBUF+0	GET ANA V1 OR V3
19A0 CD B6 19	0322	CALL	ANA	
19A3 F5	0323	PUSH	PSW	SAVE RESULT
19A4 21 12 25	0324	LXI	H,SWPBUF+2	ADD ANA V2 OR V4
19A7 CD B6 19	0325	CALL	ANA	
19AA 5F	0326	MOV	E,A	AVERAGE THE RESULTS
19AB F1	0327	POP	PSW	
19AC 83	0328	ADD	E	
19AD 1F	0329	RAR		
19AE 5F	0330	MOV	E,A	
19AF 3E 04	0331	MVI	A,RESULT	SET IT
19B1 CD FC 1A	0332	CALL	GETOPT	
19B4 73	0333	MOV	M,E	
19B5 C9	0334	RET		
19B6	0335	*		
19B6	0336	*	ANALYZE ONE	SENSOR SWEEP
19B6	0337	*		
19B6 22 10 29	0338	ANA	SHLD	BPTR SAVE BUFFER START ADDRESS
19B9 3E 06	0339	MVI	A,NREJ	REJECT(N)
19BB CD FC 1A	0340	CALL	GETOPT	
19BE CD E1 19	0341	CALL	SMOOTH	
19C1	0342	*		
19C1 3E 05	0343	MVI	A,MAVG	FINDMIN(M)
19C3 CD FC 1A	0344	CALL	GETOPT	
19C6 CD 4C 1A	0345	CALL	FINDMIN	A=INDEX OF MINIMUM
19C9 5F	0346	MOV	E,A	
19CA	0347	*		
19CA FE FF	0348	CPI	-1	IF NULL INDEX, USE ALTERNATE
19CC 3E 03	0349	MVI	A,ALT	

```
19CE CA FC 1A      0350      JZ      GETOPT
19D1                0351 *
19D1                0352 * CALCULATE THE CURRENT FROM THE INOEX
19D1                0353 *
19D1 3E 07         0354      MVI    A,ISTEP  A=INITIAL STEP
19D3 CD FC 1A     0355      CALL   GETOPT
19D6 16 00        0356      MVI    D,0      L= STEP*INOEX
19D8 E5           0357      PUSH   H
19D9 CD 0A 06    0358      CALL   MU21
19DC 01           0359      POP    D
19DD              0360 *
19D0 13           0361      INX    D      PICK UP INITIAL BIAS
19DE 1A           0362      LOAX   0
19DF 85           0363      ADD    L      ADD 1T 1N
19E0 C9           0364      RET
19E1              0365 *
19E1              0366 * SMOOTH THE I/V CURVE
19E1              0367 * ON ENTRY: [A] = # TIMES TO OPERATE
19E1              0368 *
19E1 32 12 29    0369 SMOOTH STA  NTEMP
19E4 3C           0370      INR    A      1F -1, NO SMOOTHING
19E5 C8           0371      RZ      .
19E6 06 7E       0372      MVI    B,126  FOR B=126 TO 2
19E8 CD F3 19    0373 SM1  CALL   SMOCHK  CHECK/REPLACE PT[B]
19EB 05           0374      OCR    B
19EC 78           0375      MOV    A,B
19ED FE 02       0376      CPI    2
19EF 02 E8 19    0377      JNC   SM1
19F2 C9           0378      RET
19F3              0379 *
19F3 78           0380 SMOCHK MOV    A,B  IF PT[B]-PT[B-1]+N>0
19F4 CD 0E 18    0381      CALL   LOADV  THEN POINT IS OK.
19F7 C8           0382      RZ      .  (SATURATED)
19F8 3A 12 29    0383      LDA    NTEMP
19F8 0F           0384      RST   REF/B
19FC E8           0385      YCHG  .
19FD 78           0386      MOV    A,B
19FE 30           0387      OCR    A
19FF CD 0E 18    0388      CALL   LOADV
1A02 C8           0389      RZ      .
1A03 CD 88 00    0390      CALL   NEG16
1A06 19           0391      DAD    D
1A07 7C           0392      MOV    A,H
1A08 B7           0393      ORA    A
1A09 F0           0394      RP      .
1A0A              0395 *
1A0A              0396 * SLOPE IS FOUND TO BE TOO NEGATIVE FROM [B-1] TO [B]
1A0A              0397 * DECIDE WHICH OF [B] AND [B-1] IS THE PROBLEM
1A0A              0398 *
1A0A 78           0399      MOV    A,B  CALCULATE -[B]-[B-1]
```

1A08 CD 0E 1B	0400	CALL	LOADV	
1A0E EB	0401	XCHG		
1A0F 78	0402	MOV	A,B	
1A10 3D	0403	OCR	A	
1A11 CD 0E 1B	0404	CALL	LOADV	
1A14 19	0405	DAD	D	
1A15 CD 88 00	0406	CALL	NEG16	
1A18 EB	0407	XCHG	.	
1A19	0408 †			
1A19 78	0409	MOV	A,8	+[B+1]
1A1A 3C	0410	INR	A	
1A18 CD 0E 1B	0411	CALL	LOADV	
1A1E C8	0412	RZ	.	
1A1F 19	0413	DAD	D	
1A20 EB	0414	XCHG		
1A21	0415 †			
1A21 78	0416	MOV	A,8	+[B-2]
1A22 06 02	0417	SUI	2	
1A24 CD 0E 1B	0418	CALL	LOADV	
1A27 C8	0419	RZ	.	
1A28 19	0420	DAD	D	
1A29	0421 †			
1A29 7C	0422	MOV	A,H	IF POSITIVE, REPLACE [B]
1A2A B7	0423	DRA	A	
1A28 78	0424	MOV	A,8	
1A2C F2 30 1A	0425	JP	INTERP	
1A2F 3D	0426	OCR	A	ELSE REPLACE [B-1]
1A30	0427 †			
1A30	0428 †	INTERPOLATE POINT [A] FROM SURROUNDING POINTS		
1A30	0429 †			
1A30 4F	0430	INTERP MOV	C,A	INTERPOLATE PT[X]
1A31 3C	0431	INR	A	
1A32 CD 0E 1B	0432	CALL	LOADV	
1A35 EB	0433	XCHG		
1A36	0434 †			
1A36 79	0435	MOV	A,C	
1A37 3D	0436	OCR	A	
1A38 CD 0E 1B	0437	CALL	LOADV	
1A38 19	0438	DAD	D	
1A3C 10	0439	DB	SRHL	
1A3D EB	0440	XCHG		
1A3E	0441 †			
1A3E 79	0442	MOV	A,C	
1A3F CD 22 18	0443	CALL	REFV	
1A42 73	0444	MOV	M,E	
1A43 23	0445	INX	H	
1A44 72	0446	MOV	M,D	
1A45	0447 †			
1A45 3E 0B	0448	MVI	A,PTSREJ	
1A47 CD FC 1A	0449	CALL	GETOPT	

1A4A 34	0450	INR	M	
1A4B C9	0451	RET		
1A4C	0452	*		
1A4C	0453	*	FIND MIN DELTA-V	
1A4C	0454	*	ON ENTRY: [A] = # PTS TO AVERAGE (M)	
1A4C	0455	*	ON EXIT : [A] = INDEX OF MINIMUM DELTA V	
1A4C	0456	*		
1A4C 11 FF 7F	0457	FINDMIN LXI	D,7FFFH [DEI]=MAX VALUE	
1A4F 4F	045B	MOV	C,A C = M	
1A50 3E FF	0459	MVI	A,-1 MININX = -1	
1A52 32 13 29	0460	STA	MININX	
1A55	0461	*		
1A55 3E 7E	0462	MVI	A,127-1 START AT 127 - M - 1	
1A57 91	0463	SUB	C	
1A58 47	0464	MOV	B,A	
1A59	0465	*		
1A59 D5	0466	DVLOP	PUSH D	
1A5A CD 74 1A	0467	CALL	GETDV COMPUTE DELTA V	
1A5D D1	046B	POP	D	
1A5E 7D	0469	MOV	A,L COMPARE TO OLD MINIMUM	
1A5F 93	0470	SUB	E	
1A60 7C	0471	MOV	A,H	
1A61 9A	0472	SBB	D	
1A62 D2 6A 1A	0473	JNC	DVCONT	
1A65 EB	0474	XCHG	. IF LESS, SAVE NEW MINIMUM	
1A66 7B	0475	MOV	A,B AND RECORD THE INDEX	
1A67 32 13 29	0476	STA	MININX	
1A6A 05	0477	DVCONT	DCR B	
1A6B 7B	047B	MOV	A,B CONTINUE UNTIL B<M	
1A6C B9	0479	CMP	C	
1A6D D2 59 1A	0480	JNC	DVLOP	
1A70 3A 13 29	0481	LDA	MININX RETURN INDEX OF MINIMUM	
1A73 C9	0482	RET		
1A74	0483	*		
1A74	0484	*	COMPUTE DELTA-V FOR POINT AT [B]	
1A74	0485	*	ON ENTRY: [B] = INDEX INTO ARRAY	
1A74	0486	*	[C] = #PTS TO AVERAGE	
1A74	0487	*	ON EXIT : [HL] = DELTA-V OR MAX IF SATURATED	
1A74	048B	*		
1A74 78	0489	GETDV	MOV A,B COMPUTE SUM(B+1,M)	
1A75 3C	0490	INR	A MINUS SUM(B-M,M)	
1A76 CD 87 1A	0491	CALL	SUM	
1A79 D8	0492	RC	.	
1A7A E5	0493	PUSH	H	
1A7B 7B	0494	MOV	A,B	
1A7C 91	0495	SUB	C	
1A7D CD 87 1A	0496	CALL	SUM	
1A80 D1	0497	POP	D	
1A81 DB	0498	RC	.	
1A82 CD 8B 00	0499	CALL	NEG16	

1A85 19	0500	DAD	D	
1A86 C9	0501	RET		
1A87	0502	*		
1A87	0503	*	COMPUTE THE SUM OF 'C' SAMPLES	
1A87	0504	*	ON ENTRY: [A]=INDEX INTO THE ARRAY	
1A87	0505	*	ON EXIT : [HL]= SUM OR 7FFFH IF SATURATED (HIGH OR LOW)	
1A87	0506	*		
1A87 11 00 00	0507	SUM	LXI D,0	SUM=0
1A8A C5	0508	PUSH	B	
1A8B F5	0509	SUM1	PUSH	PSW
1A8C CD 0E 1B	0510	CALL	LOADV	ADD 1 VALUE
1A8F CA 9E 1A	0511	JZ	OVER	IF SATURATED, RETURN
1A92 19	0512	DAD	D	
1A93 EB	0513	XCHG	.	
1A94 F1	0514	POP	PSW	
1A95 3C	0515	INR	A	STEP TO NEXT
1A96 0D	0516	DCR	C	FOR C TIMES
1A97 C2 BB 1A	0517	JNZ	SUM1	
1A9A EB	0518	XCHG	.	RESULT TO [HL]
1A9B C1	0519	POP	B	
1A9C B7	0520	ORA	A	RETURN NO CARRY
1A9D C9	0521	RET		
1A9E	0522	*		
1A9E F1	0523	OVER	POP	PSW
1A9F C1	0524	POP	B	
1AA0 21 FF 3F	0525	LXI	H,03FFFH	RETURN MAX
1AA3 37	0526	STC	.	WITH CARRY SET
1AA4 C9	0527	RET		
1AA5	0528	*		
1AA5	0529	*	SWEEP TRANSMIT	
1AA5	0530	*		
1AA5 0E 40	0531	TRANSMIT	MVI C,40H	CHECK TRANSMIT ENABLE
1AA7 CD DC 1A	0532	CALL	CHKENA	
1AAA CB	0533	RZ		
1AAB CD F0 1A	0534	CALL	REFOPT	
1AAE EB	0535	XCHG	.	COPY THE OPTIONS/RESULTS
1AAF 21 F2 24	0536	LXI	H,SWPHDR+2	INTO THE HEADER
1AB2 0E 0F	0537	MVI	C,SWPBUF-SWPHDR-2/2	CONVERT
1AB4 CD D0 1A	0538	CALL	EXPAND	FROM BYTE TO INT
1AB7	0539	*		
1AB7 21 E0 0E	0540	LXI	H,XMTCODE	SET HEADER FOR WHICH SWEEP
1ABA 3A E1 24	0541	LDA	BOOM	
1ABD DF	0542	RST	REF/B	
1ABE 22 F0 24	0543	SHLD	SWPHDR	
1AC1	0544	*		
1AC1 2A EE 24	0545	LHLD	SWPPTR	COMPUTE LENGTH OF PLAYBACK
1AC4 11 10 DB	0546	LXI	D,-SWPHDR	
1AC7 19	0547	DAD	D	ADD, THEN DIV BY 2
1AC8 10	0548	DB	SRHL	
1AC9 EB	0549	XCHG	.	

1ACA 21 F0 24	0550	LX1	H,SWPHDR	TELL THE TELEMETRY FORMATTER
1ACD C3 CF 13	0551	JMP		ELEXMIT
1AD0	0552	*		
1AD0 1A	0553	EXPAND	LOAX	D CONVERT BYTE TO INT ARRAY
1AD1 13	0554	INX		D
1AD2 77	0555	MOV		M,A
1AD3 23	0556	INX		H
1AD4 36 00	0557	MV1		M,0
1AD6 23	0558	INX		H
1AD7 0D	0559	DCR		C
1ADB C2 D0 1A	0560	JNZ		EXPAND
1ADB C9	0561	RET		
1ADC	0562	*		
1ADC	0563	*	OPTIONS	REFERENCING
1ADC	0564	*		
1ADC CD F0 1A	0565	CHKENA	CALL	REFOPT CHECK ENABLE OPTIONS
1ADF C3 E5 1A	0566	JMP		CHK1
1AE2 CD 02 1B	0567	CHKNTD	CALL	REFCON CHECK ENABLE CONSTANTS
1AE5 CD D2 13	0568	CHK1	CALL	ELESTAT GET THE 1/V MODE
1AEB 0F	0569	RRC		
1AE9 79	0570	MOV		A,C
1AEA D2 EE 1A	0571	JNC		GM1
1AED 07	0572	ADD	A	IN IMODE, MOVE BITS OVER
1AEE A6	0573	GM1	ANA	M
1AEF C9	0574	RET		
1AF0	0575	*		
1AF0 2A E1 24	0576	REFOPT	LHLD	BOOM L=BOOM NUMBER (1,3 = V12/V34)
1AF3 2D	0577	REFL	DCR	L IF 1, RETURN V12 LIST
1AF4 21 C3 24	0578	LX1		H,RAM12
1AF7 C8	0579	RZ		
1AF8 21 D2 24	0580	LX1		H,RAM34
1AFB C9	0581	RET		
1AFC	0582	*		
1AFC CD F0 1A	0583	GETOPT	CALL	REFOPT [HL]->OPTIONS
1AFF DF	0584	RST	REF/8	[HL]-> OPTION[A]
1B00 7E	0585	MOV		A,M
1B01 C9	0586	RET		
1B02	0587	*		
1B02 2A E1 24	0588	REFCON	LHLD	BOOM RETURN CONSTANTS
1B05 2D	0589	DCR		L
1B06 21 3E 1B	0590	LXI		H,CON12
1B09 C8	0591	RZ		
1B0A 21 46 1B	0592	LXI		H,CON34
1B0D C9	0593	RET		
1B0E	0594	*		
1B0E CD 22 1B	0595	LOADV	CALL	REFV LOAD 16 BIT FROM ARRAY
1B11 CD 1D 1B	0596	CALL		LD2
1B14 7C	0597	MOV		A,H
1B15 FE 07	0598	CP1		7
1B17 CB	0599	RZ		

```

1818 E6 0F      0600      ANI      15
181A FE 08      0601      CPI      8
181C C9         0602      RET      .
181D           0603 *
181D 7E         0604 LD2   MOV      A,M
181E 23         0605      INX      H
181F 66         0606      MOV      H,M
1820 6F         0607      MOV      L,A
1821 C9         0608      RET
1822           0609 *
1822 2A 10 29    0610 REFV  LHLD   BPTR  REFERENCE ARRAY[A]
1825 87         0611      ADD      A      (TIMES 2)
1826 87         0612      ADD      A      (TIMES 4)
1827 D2 28 18   0613      JNC     REFV1  IF NO CARRY, HL=HL+A
182A 24         0614      INR      H
182B DF         0615 REFV1 RST   REF/8
182C C9         0616      RET
182D           0617 *
182D           0618 * RAM AREA DEFINITIONS
182D           0619 *
182D           0620 DPT   EQU   0      OPTIONS
182D           0621 ANGSWP EQU   1      SWEEP ANGLE
182D           0622 ALGND EQU   2      ALGORITHM #
182D           0623 ALT   EQU   3      BIAS ALTERNATE
182D           0624 RESULT EQU   4      BIAS RESULT
182D           0625 MAVG  EQU   5      #POINTS TO AVERAGE
182D           0626 NREJ  EQU   6      NOISE PASS LIMIT
182D           0627 ISTEP EQU   7      IBIAS STEP (VMODE)
182D           0628 I80   EQU   8      INITIAL IBIAS (VMODE)
182D           0629 VSTEP EQU   9      VBIAS STEP (IMODE)
182D           0630 V80   EQU  10      INITIAL VOLTAGE (IMODE)
182D           0631 PTSREJ EQU  11      #POINTS REJECTED BY SMOOTHING
182D           0632 *
182D           0633 * DEFAULT FOR V12 AND V34
182D           0634 *
182D 00         0635 ROMDEF DB   0      INITIAL STATE=0
182E 08         0636      DB   8      #SPINS BETWEEN SWITCHING (4 MINUTES)
182F           0637 *
182F CF         0638      DB   OCFH  USE ALTERNATE RESULT, ALL ELSE ENABLED
1830 58         0639      DB   VISWP  ANGLE TO SWEEP AT
1831 00         0640      DB   0      ALG#0
1832 F0         0641      DB  -16    ALTERNATE = -40 NANOAMPS
1833 FF         0642      DB   -1    RESULT
1834 05         0643      DB   5      M=5
1835 08         0644      DB   8      N=8
1836 02         0645      DB   2      ISTEP VALUE
1837 B1         0646      DB  -127   IBIAS0
1838 02         0647      DB*  2      STEP VALUE
1839 B1         0648      DB  -127   VBIAS0
183A 00         0649      DB   0

```


1B3B 00	0650	DB	0	
1B3C 00	0651	DB	0	
1B3D 00	0652	DB	0	
1B3E	0653 †			
1B3E 16	0654 CON12	DB	016H	
1B3F 9F	0655	DB	V1FIT-1	SET BIAS1&2 VALUES JUST BEFORE FIT
1B40 01	0656	DB	1	BIAS 1,2 PAIR IN VOLT PHASE
1E41 06	0657	DB	V1	MEASUREMENT QTYS
1B42 FC	0658	DB	-V2F	
1B43 01	0659	DB	1	BIAS 1,2 PAIR IN CURR PHASE
1B44 0B	0660	DB	RI1	
1B45 04	0661	DB	RI2	
1B46	0662 †			
1B46 34	0663 CON34	DB	034H	
1B47 1F	0664	DB	V3FIT-1*256/256	CHANGE BIAS34 BEFORE FIT
1B4B 03	0665	DB	3	BIAS 3,4 IN VOLT PHASE
1B49 03	0666	DB	V3	
1B4A 0C	0667	DB	V4	
1B4B 01	0668	DB	1	BIAS 1,2 IN CURR PHASE
1B4C 0B	0669	DB	RI1	MEASURE RI1/2
1B4D 04	0670	DB	RI2	
1B4E 00	V 0671	DB	256	END OF SWP
1B4F	0672 †			
1B4F	0673 †			ENTER COMMAND VECTOR INTO TABLE
1B4F	0674 †			
1B4F	0675	ORG	SWPCODE/4+CMOTAB	
005B 1B 1B	0676	OW	SWPCMD	
005A	0677 †			
005A	0678 †			SWP RAM AREA
005A	0679 †			
005A	0680	ORG	SWPRAM	
24C0	0681 INDEX	DS	1	COMMAND VALUE ENTRY
24C1	0682 STATE	DS	1	STATE OF SWEEP MODULE
24C2	0683 SPINMAX	OS	1	#SPINS BETWEEN SWEEPS
24C3	0684 RAM12	OS	15	PARAMETERS FOR BOOMS 1,2
24D2	0685 RAM34	DS	15	PARAMETERS FOR BOOMS 3,4
24E1	0686 BOOM	DS	1	BOOM SYSTEM (0=V12)
24E2	0687 †			
24E2	0688 SWPREQ	DS	1	SWEEP REQUEST
24E3	0689 SWPOK	OS	1	SWEEP REQUEST OK
24E4	0690 SPINCNT	DS	1	SPIN COUNTER
24E5	0691 ANAVECT	OS	4	VECTOR = (AA,C3,XX,XX)
24E9	0692 †			
24E9	0693 SWPOEL	OS	1	BIAS DELTA
24EA	0694 SWPBIAS	DS	1	BIAS VALUE
24EB	0695 SWPPAIR	DS	1	SWP BIAS OACS TO USE
24EC	0696 QTYA	OS	1	MUX QTYS
24EO	0697 QTYB	OS	1	
24EE	0698 †			
24EE	0699 SWPPTR	DS	2	SAMPLE POINTER

24F0	0700 SWPHDR DS	2*16	SWEEP HEADER INFO
2510	0701 SWPBUF DS	2*128*2*2	DATA AREA
2910	0702 SWPEND EQU	\$	
2910	0703 †		
2910	0704 BPTR DS	2	BUFFER POINTER
2912	0705 NTEMP DS	1	N TEMPORARY
2913	0706 MININX DS	1	MINIMUM INDEX CALC
2914	0707 SAWMODE DS	1	SAWTOOTH MODE SAVED
2915	0708 NEEDFLIP DS	1	RELAYS NEED FLIPPING IF NZ
2916	0709 †		
2916	0710 † EXTERNALS		
2916	0711 †		
2916	0712 ELEXMT EQU	ELE+0FH	
2916	0713 ELESTAT EQU	ELE+12H	
2916	0714 SANDSC EQU	SAW+6	
2916	0715 †		
2916	0716 WORD EQU	BKGRAM+0	

```

0000          0001 *
0000          0002 * CRRES FLIGHT PROGRAM---EXECUTIVE CONTROL
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 * FILE EXEC.A
0000          0005 *
0000          0006 * 8085 SPECIFIC INFORMATION
0000          0007 *
0000          0008 PSW   EQU   6
0000          0009 SP    EQU   6
0000          0010 *
0000          0011 * RAM CONFIGURATION
0000          0012 *
0000          0013 RAM   EQU   2000H
0000          0014 RAMSIZE EQU 1000H
0000          0015 STACK EQU  RAM+RAMSIZE-1
0000          0016 *
0000          0017 * RESET VECTOR
0000          0018 *
0000          0019      ORG   0
0000 C3 6D 1B   0020      JMP   EXEINIT FOREGROUND START
0003          0021 BKGINIT DS 3   BACKGROUND START
0006          0022 *
0006          0023 * MAIN PROCESSOR EXECUTIVE CONTROL
0006          0024 *
0006          0025      ORG   EXEC
1B64 C3 FD 1B   0026      JMP   EXEANG
1B67          0027 *
1B67          0028 * EXECUTIVE STATUS
1B67          0029 *
1B67 21 A0 24   0030 EXEDSC LXI  H,VERSION RETURN VARS
1B6A DF        0031      RST  REF/8
1B6B 7E        0032      MOV  A,M
1B6C C9        0033      RET
1B6D          0034 *
1B6D          0035 * BEGIN
1B6D          0036 *
1B6D 31 FF 2F   0037 EXEINIT LXI  SP,STACK INIT STACK POINTER
1B70 21 00 20   0038      LXI  H,RAM ZERO THE RAM
1B73 01 00 0F   0039      LXI  B,RAMSIZE-256 EXCEPT FOR LAST PAGE
1B76 CF        0040 CLEAR RST  ZERO/8
1B77 05        0041      DCR  B
1B78 C2 76 1B   0042      JNZ  CLEAR
1B78          0043 *
1B78 CD 9A 00   0044      CALL IDINIT INIT THE IO MODULE
1B7E CD 03 00   0045      CALL BKGINIT INIT THE BACKGROUND MGR
1B81 CD E4 17   0046      CALL SWPINIT INIT EXECUTIVE MODULES
1B84 CD E8 11   0047      CALL FITINIT
1B87 11 40 1C   0048      LXI  D,INSEQ SEND INITIAL COMMANDS
1B8A CD 2A 1C   0049      CALL CMDSTRING

```

188D 2A AB 24	0050	LHLD	VTIME	STAY IN INITIAL MDDE
1890 22 AC 24	0051	SHLD	MDDCTR	FOR THE FULL TIME
1893 2A C7 1B	0052	LHLD	DEFVCT	RESET THE EXVECTDR
1896 22 80 24	0053	SHLD	EXVECT	
1899	0054	*		
1899	0055	*	MAIN EXECUTIVE LOOP	
1899	0056	*		
1899 3A B0 24	0057	EXLOOP	LDA	EXVECT IF EXECUTIVE VECTOR
189C FE AA	0058	CPI	OAAH	IS ENABLED, CALL THE RROUTINE
189E C2 A7 1B	0059	JNZ	EXWT	
18A1 CD B1 24	0060	CALL	EXVECT+1	
18A4 C3 B5 1B	0061	JMP	EXMGR	
18A7	0062	*		
18A7 3E 21	0063	EXWT	MVI	A,21H VERSION=2.1
18A9 32 A0 24	0064	STA	VERSION	
18AC 2A C5 1B	0065	LHLD	LPPRDG	WAIT - LDW PDWER
18AF 22 AE 24	0066	SHLD	PRDG	
18B2 CD AE 24	0067	CALL	PRDG	
18B5	0068	*		
18B5 CD F1 11	0069	EXMGR	CALL	FITEXEC CYCLE BETWEEN CALCULATING
18BB CD ED 17	0070	CALL	SWPEXEC	FITS AND SWEEP DATA
18BB CD 0A 1C	0071	CALL	DECMDE	DECIDE THE MODE (I/V)
18BE 97	0072	SUB	A	CLEAR THE EXECUTIVE WATCHDDG
18BF 32 84 24	0073	STA	EXWATCH	
18C2 C3 99 1B	0074	JMP	EXLDOP	
18C5	0075	*		
18C5 76	0076	LPPRDG	HLT	. HALT IN RAM
18C6 C9	0077	RET	.	THEN RETURN AFTER INTERRUPT
18C7 55	0078	DEFVCT	DB	055H INVERT VECTDR ENABLE BITS
18CB C9	0079	RET	.	AND PUT A RETURN IN CASE
18C9	0080	*		
18C9	0081	*	CAL/TEST MDDE PROGRAM	
18C9	0082	*		
18C9 7C	0083	CALCMD	MDV	A,H IF EVEN, SEND THE CAL
18CA 0F	0084	RRC	.	COMMAND SEQUENCE
18CB 11 B2 1C	0085	LXI	D,CALSEQ	
18CE D2 2A 1C	0086	JNC	CMDSTRING	
18D1	0087	*		
18D1 3A 1D 20	0088	SYNCWT	LDA	FRAME WAIT FOR FRAME(L)
18D4 8D	0089	CMP	L	
18D5 C2 D1 1B	0090	JNZ	SYNCWT	
18DB C9	0091	RET		
18D9	0092	*		
18D9	0093	*	MODE CDNTRDL SECTION	
18D9	0094	*		
18D9	0095	VIANG	EQU	128+32
18D9	0096	MDDEFLLAG	EQU	I
18D9	0097	*		
18D9 3E 98	0098	MODINI	MVI	A,VIANG-8 CHANGE ANGLE = 11 DEGREES
18DB 32 A2 24	0099	STA	CHGANG	

1BDE	0100 †		
1BDE 32 A1 24	0101	MDDCMD STA	MODTIM RECDRD MDDE TIMING
1BE1 CD B0 00	0102	CALL	UNARY CDNVERT VTIME
1BE4 2B	0103	DCX	H
1BE5 22 AB 24	0104	SHLD	VTIME
1BEB 3A A1 24	0105	LDA	MODTIM THEN ITIME
1BEB 0F	0106	RRC	
1BEC 0F	0107	RRC	
1BED 0F	0108	RRC	
1BEE 0F	0109	RRC	
1BEF CD B0 00	0110	CALL	UNARY
1BF2 2B	0111	DCX	H
1BF3 22 AA 24	0112	SHLD	ITIME
1BF6 21 01 00	0113	LXI	H,1 SET FOR SWITCH NEXT TIME
1BF9 22 AC 24	0114	SHLD	MDDCTR
1BFC C9	0115	RET	
1BFD	0116 †		
1BFD 21 A2 24	0117	EXEANG LXI	H,CHGANG CHECK IF THE ANGLE
1C00 BE	0118	CMP	M 1S CORRECT.
1C01 C0	0119	RNZ	.
1C02 2A AC 24	0120	LHLD	MODCTR THEN DDWN-COUNT THE MDDE
1C05 2B	0121	DCX	H COUNTER.
1C06 22 AC 24	0122	SHLD	MODCTR
1C09 C9	0123	RET	
1C0A	0124 †		
1C0A	0125 †	DECIDE WHEN TO SWITCH MODES	
1C0A	0126 †		
1C0A 2A AC 24	0127	DECMDE LHLD	MODCTR IF CDUNT=0, SWITCH
1C0D 7C	0128	MOV	A,H
1C0E B5	0129	DRA	L
1C0F C0	0130	RNZ	
1C10	0131 †		
1C10 CD D2 13	0132	CALL	ELESTAT GET THE CURRENT MDDE
1C13 E6 01	0133	ANI	MODEFLAG
1C15 2A AB 24	0134	LHLD	VTIME IF 1MDDE, TRY VTIME
1C1B 11 2E 1C	0135	LXI	D,VCMD5 AND SEND V COMMANDS
1C1B C2 24 1C	0136	JNZ	TRYMODE
1C1E 2A AA 24	0137	LHLD	ITIME IN VMODE, TRY IMODE
1C21 11 3B 1C	0138	LXI	D,ICMDS AND SEND I COMMANDS
1C24	0139 †		
1C24 7C	0140	TRYMODE MOV	A,H IF MDDE TIME=0
1C25 B5	0141	DRA	L DDN'T SWITCH AT ALL
1C26 C8	0142	RZ	.
1C27 22 AC 24	0143	SHLD	MDDCTR
1C2A	0144 †		
1C2A 3E 01	0145	CMDSTRING MVI	A,1
1C2C E7	0146	RST	4
1C2D C9	0147	RET	
1C2E	0148 †		
1C2E 02 50	0149	VCMD5 DW	5002H ENABLE V12 FITS

1C30 00 5C	0150	DW	5C00H	DISABLE SAWTOOTH
1C32 00 68	0151	DW	6800H	GO TO VOLTAGE MODE
1C34 01 62	0152	DW	6201H	PUT OLD V12 RESULT OUT
1C36 FF FF	0153	DW	-1	
1C38 03 50	0154	ICMDS DW	5003H	DISABLE V12 FITS
1C3A 01 68	0155	DW	6801H	GO TO LANGMUIR PROBE MODE
1C3C 03 5C	0156	DW	5C03H	ENABLE SAWTOOTH STEP+BIAS
1C3E FF FF	0157	DW	-1	
1C40	0158	†		
1C40 01 43	0159	INISEQ DW	4301H	FORMAT 0 1
1C42 F0 01	0160	DW	01F0H	BIAS 1-4 TO -40NAMPS
1C44 F0 02	0161	DW	02F0H	
1C46 F0 03	0162	DW	03F0H	
1C48 F0 04	0163	DW	04F0H	
1C4A FF 09	0164	DW	09FFH	GUARDS 1-2 TO -1
1C4C FF 0A	0165	DW	0AFFH	
1C4E FF 11	0166	DW	11FFH	STUBS 1-2 TO 0
1C50 FF 12	0167	DW	12FFH	
1C52 FF 21	0168	DW	21FFH	FILTERS 1-7 TO 255
1C54 FF 22	0169	DW	22FFH	
1C56 FF 23	0170	DW	23FFH	
1C58 FF 24	0171	DW	24FFH	
1C5A FF 25	0172	DW	25FFH	
1C5C FF 26	0173	DW	26FFH	
1C5E FF 27	0174	DW	27FFH	
1C60 FF 19	0175	DW	19FFH	VTRIMS 1-2 TO -1
1C62 FF 1A	0176	DW	1AFFH	
1C64 0A 31	0177	DW	310AH	SET 10 (GROUND OUTER BRAID)
1C66 08 30	0178	DW	300BH	RESET 11 (BIAS REFS12=GND NOT P03)
1C68 0D 30	0179	DW	300DH	RESET 13 (GUARD FILTER OFF)
1C6A 0E 30	0180	DW	300EH	RESET 14 (STUB FILTER OFF)
1C6C 11 30	0181	DW	3011H	RESET 17 (BIAS3,4 REF=V3,V4 NOT GND)
1C6E 14 31	0182	DW	3114H	SET 20 (UNBIAS DC PREAMP 3)
1C70 15 31	0183	DW	3115H	SET 21 (UNBIAS DC PREAMP 4)
1C72 00 30	0184	DW	3000H	RESET 0 (STEERING FOR 20 AND 21)
1C74 01 2A	0185	DW	2A01H	MUX 2=1 (KAGC = V12/R11)
1C76 01 2B	0186	DW	2B01H	MUX 3=1 (V1/SC = V1)
1C78 00 4B	0187	DW	4B00H	RAMBASE 0X2000
1C7A 00 68	0188	DW	6800H	START OUT IN VMODE
1C7C 77 38	0189	DW	3877H	SWITCH EVERY 128 SPINS
1C7E FF CB	0190	DW	0CBFFH	COMMAND COUNTER (RESET/ERRGR)
1C80 FF FF	0191	DW	-1	END OF LIST
1C82	0192	†		
1C82 00 60	0193	CALSEQ DW	6000H	DISABLE SWEEPS
1C84 03 61	0194	DW	6103H	
1C86 C0 58	0195	DW	058C0H	SAWTOOTH OFFSET = -64
1C88 02 5C	0196	DW	05C02H	DISABLE SAWTOOTH BIASING
1C8A 03 68	0197	DW	6803H	SWITCH TO 1MODE WITH TEST MARK
1C8C 01 29	0198	DW	2901H	MUX 1=1 TO GET V1=BIAS1
1C8E 00 80	0199	DW	08000H	BURST FREQUENCY 0

1C90 11 31	0200	DW	3111H SET K17
1C92 01 30	0201	DW	3001H RESET K1
1C94 05 B1	0202	DW	0B105H BURST FORMAT = 5 (ALL QTVS)
1C96	0203 *		
1C96 00 91	0204	DW	9100H WAIT FOR FRAME(0)
1C98 9C 03	0205	DW	039CH BIAS 3= -100
1C9A 64 04	0206	DW	0464H BIAS 4= +100
1C9C 06 31	0207	DW	3106H SET RELAY(6)
1C9E 20 91	0208	DW	9120H WAIT 4 SECONDS
1CA0 64 03	0209	DW	0364H BIAS 3= +100
1CA2 9C 04	0210	DW	049CH BIAS 4= -100
1CA4 06 31	0211	DW	3106H SET RELAY(6)
1CA6 40 91	0212	DW	9140H WAIT 4 SECONDS
1CA8	0213 *		
1CA8 03 5C	0214	DW	05C03H SAWOPT = 60
1CAA 00 B4	0215	DW	0B400H BURST 60
1CAC 80 91	0216	DW	09180H WAIT FOR 8 SECONDS
1CAE 00 B5	0217	DW	0B500H BURST STOP/PLAYBACK
1CB0 00 5C	0218	DW	05C00H TURN OFF SAW
1CB2 00 68	0219	DW	6800H VOLTAGE MODE
1CB4 00 60	0220	DW	6000H REENABLE SWEEPS
1CB6 00 61	0221	DW	6100H
1CB8 01 62	0222	DW	6201H PUT OUT BIAS1/2 RESULT
1CBA 03 62	0223	DW	6203H AND BIAS3/4 RESULTS
1CBC 11 30	0224	DW	3011H RESET K17 (BIAS3,4 REF=3,4 NOT GND)
1CBE FF FF	0225	DW	-1
1CC0	0226 *		
1CC0 4D 41 49 4E	0227	ASC	"MAIN2,1-HARVEY" VERSION STAMP
32 2E 31 2D			
48 41 52 56			
45 59			
1CCE 00	V 0228	DB	256 END OF EXEC
1CCF	0229 *		
1CCF	0230 *		RAM AREA FOR THE EXECUTIVE
1CCF	0231 *		
1CCF	0232	ORG	EXERAM
24A0	0233	VERSION DS	1
24A1	0234	MODTIM DS	1
24A2	0235	CHGANG DS	1
24A3	0236	OTHER DS	5
24A8	0237	VTIME DS	2
24AA	0238	ITIME DS	2
24AC	0239	MODCTR DS	2
24AE	0240	PRG DS	2 RAM AREA TO REST
24B0	0241	EYEVECT DS	4 FOREGROUND VECTOR (AA,C3,XX,XX)
24B4	0242 *		
24B4	0243	COM	EXWATCH EXECUTIVE WATCHDOG
24B4	0244	EXWATCH DS	1
24B5	0245 *		
24B5	0246 *		CONTROLLED MODULES

24B5	0247 ‡		
24B5	0248	ORG	LD
1680	0249	LDINIT DS	3
1683	0250 ‡		
1683	0251	ORG	SWP
17E4	0252	SWPINIT DS	3
17E7	0253	SWPANG DS	3
17EA	0254	SWPSTAT DS	3
17ED	0255	SWPEXEC DS	3
17F0	0256 ‡		
17F0	0257	ORG	FIT
11E8	0258	FITINIT DS	3
11E8	0259	FITSMP DS	3
11EE	0260	FITTEL DS	3
11F1	0261	FITEXEC DS	3
11F4	0262 ‡		
11F4	0263	ELESTAT EQU	ELE+12H MODE STATUS
11F4	0264	FRAME EQU	BKGRAM+1 FRAME PART OF CLOCK
11F4	0265 ‡		
11F4	0266 ‡	DEFINE EXECUTIVE COMMANDS	
11F4	0267 ‡		
11F4	0268	ORG	38H/4+CMDTAB
004E DE 1B	0269	DW	MODCMD
0050	0270 ‡		
0050	0271	ORG	90H/4+CMDTAB
0064 C9 1B	0272	DW	CALCMD


```

0000          0001 *
0000          0002 * CRRES FLIGHT PROGRAM---BACKGROUND MANAGEMENT
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 *
0000          0005 * FILE : BKG.A
0000          0006 *
0000          0007 DIGIT EQU 0F0H DIGIT COMMAND CODE
0000          0008 ENTER EQU 010H 'ENTER' SUB-CODE
0000          0009 CSUM EQU 0CBH CHECKSUM SET COMMAND
0000          0010 RESET EQU 070H SOFTWARE RESET COMMAND
0000          0011 *
0000          0012 LEN1 EQU 22H BOOMLEN/TEMPS FROM MUX 22-27H
0000          0013 PSM EQU 6 8085 SPECIFIC INFORMATION
0000          0014 SP EQU 6
0000          0015 RES75 EQU 16
0000          0016 MSE EQU 8
0000          0017 MSK75 EQU 4
0000          0018 MSK65 EQU 2
0000          0019 MSK55 EQU 1
0000          0020 *
0000          0021 WRDBIT EQU MSK75
0000          0022 FRMBIT EQU MSK65
0000          0023 CMDBIT EQU MSK55
0000          0024 *
0000          0025 * BACKGROUND INITIALIZATION
0000          0026 *
0000          0027 ORG 3
0000 C3 E0 IC 0028 JMP BKGINIT
0000          0029 *
0000          0030 * BACKGROUND SERVICE FUNCTIONS
0000          0031 *
0000          0032 ORG 4*8
0000 C3 C1 1D 0033 JMP BKGFN5
0000          0034 *
0000          0035 * WATCHDOG TIMER INTERRUPT
0000          0036 *
0000          0037 ORG 4*8+4
0000 C3 00 00 0038 TRAP JMP 0 RESET THE CPU
0000          0039 *
0000          0040 * COMMAND INTERRUPT (RESTART 5.5)
0000          0041 *
0000          0042 ORG 5*8+4
0000 C5 F5 0043 PUSH PSM SAVE ACCUM
0000 C3 2D 1E 0044 JMP CMDSERV
0000          0045 *
0000          0046 * SOFTWARE COMMAND INTERRUPT
0000          0047 *
0000          0048 ORG 6*8
0000 C3 61 1E 0049 JMP CMDG0

```

0033	0050 *
0033	0051 * MAJOR FRAME INTERRUPT
0033	0052 *
0033	0053 ORG 6*8+4
0034 F5	0054 PUSH PSW
0035 C3 32 1D	0055 JMP MAJINT
0038	0056 *
0038	0057 * WORD RATE CLOCK INTERRUPT (RESTART 7.5)
0038	0058 *
0038	0059 ORG 7*8+4
003C F5	0060 PUSH PSW SAVE PROGRAM STATUS WORD
003D C3 42 1D	0061 JMP WRCINT
0040	0062 *
0040	0063 * BEGIN MODULE
0040	0064 *
0040	0065 ORG BKG
1CE0 21 C0 FF	0066 BKGINIT LXI H,OFFCOH SET FRAME
1CE3 22 1D 20	0067 SHLD FRAME SO 1ST MAJOR IS 7
1CE6 21 FF FF	0068 LXI H,-1 RESET HOURS AND DAYS
1CE9 22 1F 20	0069 SHLD HR225
1CEC	0070 *
1CEC	0071 * INITIALIZE THE PACKAGES WHICH OPERATE IN THE BACKGROUND
1CEC	0072 *
1CEC CD C4 16	0073 CALL DEPINIT DEPLOYMENT
1CEF CD 80 16	0074 CALL LDINIT PROGRAM LOADING
1CF2 CD 50 0D	0075 CALL MAGINIT MAGNETOMETER DATA
1CF5 CD 98 0E	0076 CALL PLAINIT PLASMA DATA
1CF8 CD C0 13	0077 CALL ELEINIT ELECTRIC FIELD/LANGMUIR PROBE DATA
1CFB CD 09 10	0078 CALL BURINIT BURST CONTROL
1CFE CD 30 13	0079 CALL SAWINIT SAWTOOTH CONTROL
ID01	0080 *
ID01 21 00 00	0081 LXI H,0 CLEAR THE EXP OUTPUT
ID04 22 3C 20	0082 SHLD EXPCNT
ID07 CD FB 1D	0083 CALL STVECT AND THE BKG VECTOR
ID0A	0084 *
ID0A 3E 0D	0085 MVI A,MSE+WRDBIT+CMDBIT DON'T ACCEPT
ID0C CD CF 00	0086 CALL SETMASK ANYTHING UNTIL MAJOR FRM
ID0F FB	0087 EI
ID10 76	0088 HLT . LET THE FRAME INT OCCUR
ID11 3E 18	0089 MVI A,RES75+MSE ENABLE WORD AND CMDS
ID13 CD CF 00	0090 CALL SETMASK
ID16	0091 *
ID16	0092 * NOW IN "NEITHERGROUND"
ID16	0093 *
ID16 97	0094 SUB A RESET COMMAND COUNTERS
ID17 32 21 20	0095 STA GOODCNT
ID1A 32 22 20	0096 STA BADCNT
ID1D 32 2D 20	0097 STA STATUS AND STATUS FLAG
ID20 D3 00	0098 OUT 0 RESET DIAGNOSTIC LEDES
ID22 D3 01	0099 OUT I

1024	0100 *
1024 21 53 07	0101 LXI H,30000/16 FAKE A SUNPER100
1027 CD A3 10	0102 CALL SUNRES OF 30 SECS TO START
102A	0103 *
102A 3E 50	0104 KLYINIT MVI A,8*10 10 SECS PER PULSE (.1HZ)
102C 32 32 20	0105 STA KLYCNT
102F C3 2B 03	0106 JMP SETKLY
1032	0107 *
1032	0108 * MAJOR FRAME INTERRUPT
1032	0109 *
1032 3A 10 20	0110 MAJINT LDA FRAME MAKE SURE MINOR=31
1035 F6 1F	0111 ORI 31
1037 32 10 20	0112 STA FRAME
103A 3E FC	0113 MVI A,252 RESET WORD NUMBER
103C 32 10 20	0114 STA WORD
103F F1	0115 POP PSW
1040 FB	0116 EI
1041 C9	0117 RET
1042	0118 *
1042	0119 * WORD INTERRUPT
1042	0120 *
1042 E5	0121 WRCINT PUSH H SAVE (HL) REGISTER
1043 D5	0122 PUSH D AND (DE) TOG
1044 C5	0123 PUSH B
1045 FB	0124 EI
1046	0125 *
1046 3A 10 20	0126 LDA WORD INCREMENT WORD BY 2
1049 C6 02	0127 ADI 2
104B 32 10 20	0128 STA WORD
104E B7	0129 ORA A DIV BY 2 SINCE WE COUNT 16 BITS
104F 1F	0130 RAR . AT A TIME
1050 CB 59 10	0131 CALL VECTOR VECTOR TO TMTABLE(WORD)
1053	0132 *
1053 C1	0133 EXIT POP B
1054 D1	0134 POP D RETURN FROM THE INTERRUPT
1055 E1	0135 POP H
1056 F1	0136 POP PSW
1057 FB	0137 EI .
1058 C9	0138 RET .
1059	0139 *
1059	0140 * VECTOR TO TMTABLE(WORD)
1059	0141 *
1059 0F	0142 VECTOR RRC . IF WORD IS ODD, NULL
105A DA 43 1F	0143 JC NULL
105D 6F	0144 MOV L,A MAKE THE WORD (0-127)
105E 26 1F	0145 MVI H,TMTABLE/256 INTO A TABLE ADDRESS
1060 6E	0146 MOV L,M PICK UP ROUTINE ADDRESS
1061 E9	0147 PCHL .
1062	0148 *
1062	0149 * MINOP FRAME BOUNDARY (BEGINNING OF WORD 0)

1D62	0150 *			
1D62 21 1C 20	0151 MINOR LXI	H,WORD	INCREMENT WORD	
1D65 3E 04	0152 MVI	A,4	COUNTER.	
1D67 CD 80 1D	0153 CALL	INC	(UP TO 4 BYTES)	
1D6A CD B5 03	0154 CALL	RWATCH	RESET HARDWARE WATCHDOG CIRCUIT	
1D6D 21 B4 24	0155 LXI	H,EXWATCH	UPDATE EXECUTIVE LOOP WATCHDOG	
1D70 34	0156 INR	M	AND IF OVERFLOW (256), RESET CPU	
1D71 CA 24 00	0157 JZ		TRAP	
1D74 3A 1D 20	0158 LDA	FRAME	NOTIFY THE MODULES	
1D77 CD 53 0D	0159 CALL	MAGFRAME		
1D7A 3A 1D 20	0160 LDA	FRAME		
1D7D C3 C3 13	0161 JMP	ELEFRAME		
1D80	0162 *			
1D80 23	0163 INC	INX	H	STEP TO NEXT BYTE
1D81 34	0164 INR	M		INC IT
1D82 C0	0165 RNZ	.		RETURN IF DONE
1D83 3D	0166 DCR	A		GO ON
1D84 C2 80 1D	0167 JNZ	INC		
1D87 C9	0168 RET			
1D88	0169 *			
1D88	0170 *	UPDATE SUN ANGLE/ CHECK FOR SUN PULSES		
1D88	0171 *	[\]		
1D88 2A 2E 20	0172 SUNINC	LHLD	SUNCTR	
1D8B 23	0173 INX	H		
1D8C 22 2E 20	0174 SHLD	SUNCTR		
1D8F 2A 30 20	0175 LHLD	SUNDOWN	UPDATE THE SUN ANGLE	
1D92 11 00 FF	0176 LXI	D,-256	BY DIVIDING THE	
1D95 19	0177 DAD	D	SUNPERIOD BY 256	
1D96 D4 B4 1D	0178 CNC	PHASE	IF NEG. BUMP ANGLE	
1D99 22 30 20	0179 SHLD	SUNDOWN		
1D9C	0180 *			
1D9C CD D9 00	0181 CALL	SUNSTAT	IF NOT SUN	
1D9F C8	0182 RZ	.	PULSE YET, QUIT.	
1DA0	0183 *			
1DA0 2A 2E 20	0184 LHLD	SUNCTR	SET THE SUN PERIOD	
1DA3 22 25 20	0185 SUNRES	SHLD	SUNPER	
1DA6 2B	0186 DCX	H	RESET THE DOWN COUNTER	
1DA7 22 30 20	0187 SHLD	SUNDOWN		
1DAA	0188 *			
1DAA 21 00 00	0189 LXI	H,0	RESET THE SUN ANGLE	
1DAD 22 23 20	0190 SHLD	ANGLE		
1DB0 22 2E 20	0191 SHLD	SUNCTR	AND SUNINC COUNTER	
1DB3 C9	0192 RET	.		
1DB4	0193 *			
1DB4 EB	0194 PHASE	XCHG	.	PUT REMAINDER IN [DE]
1DB5 2A 23 20	0195 LHLD	ANGLE	ANGLE++	
1DB8 23	0196 INX	H		
1DB9 22 23 20	0197 SHLD	ANGLE		
1DBC 2A 25 20	0198 LHLD	SUNPER	ADD NEW SUN PERIOD	
1DBF 19	0199 DAD	D	TO THE REMAINDER FROM	

1DC0 C9	0200	RET	.	LAST PERIOD.
1DC1	0201	†		
1DC1	0202	†	SERVICE ROUTINES	
1DC1	0203	†		
1DC1 FE 01	0204	BKGFNS	CPI	I DETERMINE FUNCTION
1DC3 CA 15 1E	0205	JZ	BATCH	
1DC6 FE 02	0206	CPI	2	
1DC8 CA 0A 1E	0207	JZ	CCST	
1DC8 FE 03	0208	CPI	3	
1DCD CA FB 1D	0209	JZ	STVECT	
1DD0 FE 04	0210	CPI	4	
1DD2 C0	0211	RNZ		
1DD3	0212	†		
1DD3	0213	†	START EXPERIMENTAL OUTPUT	
1DD3	0214	†		
1DD3 ES	0215	STEXP	PUSH	H
1DD4 CD E1 1D	0216	CALL	CHKEXP	CHECK IF GOING/STOPPED
1DD7 E1	0217	POP	H	
1DD8 C0	0218	RNZ	.	IF ALREADY GOING, RETURN
1DD9 22 3A 20	0219	SHLD	EXPADR	SET DUMP ADDRESS
1DDC EB	0220	XCHG	.	
1DDD 22 3C 20	0221	SHLD	EXPCNT	AND SIZE
1DE0 C9	0222	RET		
1DE1	0223	†		
1DE1 2A 3C 20	0224	CHKEXP	LHLD	EXPCNT
1DE4 7C	0225	MOV	A,H	
1DE5 B5	0226	ORA	L	
1DE6 C9	0227	RET		
1DE7	0228	†		
1DE7 CD E1 1D	0229	EXPTL	CALL	CHKEXP IF EXPCNT=0, RETURN(0)
1DEA C8	0230	RZ	.	
1DEB 28	0231	DCX	H	ELSE CNT--
1DEC 22 3C 20	0232	SHLD	EXPCNT	
1DEF 2A 3A 20	0233	LHLD	EXPADR	AND RETURN MEM(ADR++)
1DF2 7E	0234	MOV	A,H	
1DF3 23	0235	INX	H	
1DF4 22 3A 20	0236	SHLD	EXPADR	
1DF7 C9	0237	RET		
1DF8	0238	†		
1DF8	0239	†	SET BKG VECTOR	
1DF8	0240	†		
1DF8 7C	0241	STVECT	MOV	A,H IF ZERO, CLEAR BKGVECT
1DF9 B5	0242	ORA	L	
1DFA 3E AA	0243	MVI	A,0AAH	
1DFC C2 03 1E	0244	JNZ	STV1	
1DFF 97	0245	SUB	A	
1E00 21 4D 1F	0246	LXI	H,XTRA	
1E03 22 36 20	0247	STV1	SHLD	BKGVECT
1E06 32 38 20	0248	STA	BVCHK	
1E09 C9	0249	RET		

```

1E0A          0250 †
1E0A          0251 † COMMAND COUNT STATUS. RETURNS 1 IF MORE TO GO.
1E0A          0252 †
1E0A 21 21 20 0253 CCST LX1 H,600DCNT IF COUNT MATCHES
1E0D 3A 35 20 0254 LDA CMDCNT THE #600D COMMANDS,
1E10 96        0255 SUB M
1E11 C8        0256 RZ . RETURN(0) ELSE (1)
1E12 3E 01     0257 MVI A,1
1E14 C9        0258 RET
1E15          0259 †
1E15          0260 † COMMAND BATCH PROCESSOR
1E15          0261 † ON ENTRY: [DE]-> COMMAND LIST ENDING IN -1
1E15          0262 †
1E15 1A        0263 BATCH LDAX D HL=CMD
1E16 6F        0264 MOV L,A
1E17 13        0265 INX D
1E18 1A        0266 LDAX D
1E19 67        0267 MOV H,A
1E1A 13        0268 INX D
1E1B 3C        0269 INR A IF CMD= -1, QUIT
1E1C C8        0270 RZ
1E1D 3A 2D 20 0271 BATWT LDA STATUS IF COMMAND COMING IN
1E20 FE AA     0272 CPI OAAH WAIT TILL DONE
1E22 CA 1D 1E 0273 JZ BATWT
1E25 D5        0274 PUSH D
1E26 CD 61 1E 0275 CALL CMD60
1E29 D1        0276 POP D
1E2A C3 15 1E 0277 JMP BATCH
1E2D          0278 †
1E2D          0279 † COMMAND INTERRUPT SERVICE ROUTINE
1E2D          0280 † [ \ ]
1E2D 3E AA     0281 CMDSERV MVI A,OAAH SET A SERVICE FLAG
1E2F 32 2D 20 0282 STA STATUS
1E32 3E 09     0283 MVI A,MSE+CMDBIT MASK OTHER CMDS
1E34 CD CF 00 0284 CALL SETMASK UNTIL THIS ONE
1E37 F1        0285 POP PSW IS SERVICED.
1E38 FB        0286 EI .
1E39 C9        0287 RET .
1E3A          0288 †
1E3A          0289 † CHECK FOR COMMAND INPUTS.
1E3A          0290 †
1E3A 3A 2D 20 0291 CMDEXEC LDA STATUS IF COMMAND STATUS
1E3D FE AA     0292 CPI OAAH IS ZERO, RETURN
1E3F C0        0293 RNZ .
1E40 CD CD 00 0294 CALL GETMASK IF A COMMAND
1E43 E6 10     0295 ANI CMDBIT#16 SHIFTING IN, QUIT
1E45 C0        0296 RNZ . AND GET IT NEXT TIME.
1E46          0297 †
1E46 97        0298 SUB A RESET THE STATUS
1E47 32 2D 20 0299 STA STATUS

```

1E4A	0300 #			
1E4A CD D1 00	0301	CALL	CMDIN	READ THE COMMAND REG
1E4D CD 93 00	0302	CALL	MARK	
1E50 3E 08	0303	MVI	A,MSE	RE-ENABLE CMDS
1E52 CD CF 00	0304	CALL	SETMASK	
1E55 CD 61 1E	0305	CALL	CMD60	EXECUTE IT
1E58 21 21 20	0306	LXI	H,GOODCNT	COUNT GOOD OR BAD
1E5B D2 5F 1E	0307	JNC	INRCNT	
1E5E 23	0308	INX	H	
1E5F 34	0309	INRCNT	INR	M
1E60 C9	0310	RET		
1E61	0311 #			
1E61	0312 #	COMMAND DISTRIBUTION		
1E61	0313 #			
1E61 EB	0314	CMD60	XCHG	. PUT COMMAND IN [DE]
1E62 7A	0315	MOV	A,D	MASK UPPER 5 BITS
1E63 E6 F8	0316	ANI	0F8H	
1E65 0F	0317	RRC	.	
1E66 0F	0318	RRC	.	
1E67 21 40 00	0319	LXI	H,CMDTAB	REFER TO TABLE
1E6A DF	0320	RST	REF/8	
1E6B	0321 #			
1E6B 7E	0322	MOV	A,M	PICK UP ADDRESS
1E6C 23	0323	INX	H	
1E6D 66	0324	MOV	H,M	
1E6E 6F	0325	MOV	L,A	
1E6F	0326 #			
1E6F 84	0327	ORA	H	IF ADDRESS=0, RETURN CARRY
1E70 37	0328	STC		
1E71 C8	0329	RZ	.	
1E72	0330 #			
1E72 E5	0331	PUSH	H	PUT ON STACK
1E73 EB	0332	XCHG	.	[HL]=COMMAND
1E74 7D	0333	MOV	A,L	[A]=DATA PART
1E75 B7	0334	ORA	A	CLEAR CARRY
1E76 C9	0335	RET	.	
1E77	0336 #			
1E77	0337 #	BACKGROUND MODULE COMMANDS		
1E77	0338 #			
1E77 32 35 20	0339	CSCMD	STA	CMDCNT SET CMDCNT COMPARE REG=DATA
1E7A 21 00 00	0340	LXI	H,0	CLEAR CMD COUNTERS
1E7D 22 21 20	0341	SHLD	GOODCNT	
1E80 C9	0342	RET	.	
1E81	0343 #			
1E81 FE 10	0344	DIGCMD	CPI	ENTER IF THE ENTER COMMAND
1E83 CA 95 1E	0345	JZ	ENTDIG	
1E86	0346 #			
1E86 24 33 20	0347	LHLD	DIGREG	
1E89 29	0348	DAD	H	DIGREG=DIGREG*16 + A
1E8A 29	0349	DAD	H	

1EBB 29	0350	DAD	H	
1EBC 29	0351	DAD	H	
1EBD E6 0F	0352	ANI	15	
1EBF B5	0353	ORA	L	
1E90 6F	0354	MDV	L,A	
1E91 22 33 20	0355	SHLD	DIGREG	
1E94 C9	0356	RET	.	RETURN(NO CARRY)
1E95	0357	*		
1E95 2A 33 20	0358	ENTD16	LHLD	DIGREG HL=DIGIT REGISTER
1E9B 97	0359	SUB	A	DIGIT REGISTER=0
1E99 32 33 20	0360	STA	DIGREG	
1E9C 32 34 20	0361	STA	DIGREG+1	
1E9F C3 61 1E	0362	JMP	CMDG0	EXECUTE COMMAND [HL]
1EA2	0363	*		
1EA2	0364	*		AUTONOMICS: KELLY AND TEMPERATURE UPDATES
1EA2	0365	*		
1EA2 21 32 20	0366	AUTO	LXI	H,KLYCNT
1EA5 35	0367	DCR	M	
1EA6 CC 2A 1D	0368	CZ	KLYINIT	
1EA9 CD C7 16	0369	CALL	DEPSAMP	
1EAC	0370	*		
1EAC 3A 1D 20	0371	LDA	FRAME	EVERY MAJOR FRAME
1EAF E6 1F	0372	ANI	31	SAMPLE 1 ANALDG SUBCOM VAL
1EB1 C0	0373	RNZ		
1EB2	0374	*		
1EB2	0375	*		SAMPLE BODM LENGTHS AND 1 DF 4 TEMPERATURES
1EB2	0376	*		
1EB2 3A 1D 20	0377	TEMPSAMP	LDA	FRAME COMPUTE WHICH ONE
1EB5 07	0378	RLC		
1EB6 07	0379	RLC		
1EB7 07	0380	RLC		
1EB8 E6 07	0381	ANI	7	
1EBA FE 06	0382	CPI	6	IF 6 DR 7, QUIT
1EBC D0	0383	RNC		
1EBD F5	0384	PUSH	PSW	
1EBE C6 22	0385	ADI	LEN1	ADD TD LEN1 MUX ADDRESS
1ECO CD E6 00	0386	CALL	SAMPLE	
1EC3 29	0387	DAD	H	CONVERT TD 8 BITS
1EC4 29	0388	DAD	H	
1EC5 29	0389	DAD	H	
1EC6 29	0390	DAD	H	
1EC7 5C	0391	MDV	E,H	
1ECB F1	0392	PDP	PSW	GET THE 0-3 AGAIN
1EC9 21 27 20	0393	LXI	H,SLEN1	
1ECC DF	0394	RST	REF/8	
1ECD 73	0395	MDV	M,E	
1ECE C9	0396	RET		
1ECF	0397	*		
1ECF	0398	*		DIGITAL SUBCOM TABLE
1ECF	0399	*		

1ECF 32	0400 DSCTAB DB	4B+2	QTY5 IN PKG
1ED0 CC 13	0401	DW	ELEDSC PKG ADDRESS
1ED2 0B	0402	DB	B
1ED3 67 1B	0403	DW	EXEDSC
1ED5 15	0404	DB	2+15+4
1ED6 74 02	0405	DW	IDDSC
1ED8 0C	0406	DB	12
1ED9 F0 17	0407	DW	SMPDSC
1EDB 04	0408	DB	4
1EDC 36 13	0409	DW	SAWDSC
1EDE 04	0410	DB	4
1EDF 11 10	0411	DW	BURDSC
1EE1 05	0412	DB	5
1EE2 CA 16	0413	DW	DEPDSC
1EE4 01	0414	DB	1
1EE5 9E 0E	0415	DW	PLADSC
1EE7 64	0416	DB	100
1EEB F3 1F	0417	DW	BKGDSC
1EEA	0418 *		
1EEA	0419 * SOFTWARE RESET		
1EEA	0420 *		
1EEA 7D	0421 SDFTRRESET MDV A,L		CHECK FOR COMMAND 7007
1EEB FE 07	0422	CPI	07H
1EED CA 00 00	0423	JZ	0
1EF0 37	0424	STC	
1EF1 C9	0425	RET	
1EF2 00	V 0426	DB	256 PART 1 ENDS
1EF3	0427 *		
1EF3	0428 * TELEMETRY WORD FUNCTION TABLE		
1EF3	0429 *		
1EF3	0430	DRG	\$/256+1*256 NEXT PAGE BOUNDARY
1F00	0431 T	EQU	\$
1F00 40	0432 TMTABLE DB		MNFR-T 0
1F01 4E	0433	DB	ET-T
1F02 5E	0434	DB	SPL0-T
1F03 56	0435	DB	MT-T
1F04 B6	0436	DB	ES-T
1F05 B6	0437	DB	ES-T
1F06 BF	0438	DB	SUN-T
1F07 AA	0439	DB	CMD-T
1F0B	0440 * 1		
1F0B BC	0441	DB	PS-T
1F09 4E	0442	DB	ET-T
1F0A 7B	0443	DB	SPLIT-T
1F0B 56	0444	DB	MT-T
1F0C B6	0445	DB	ES-T
1F0D B6	0446	DB	ES-T
1F0E BF	0447	DB	SUN-T
1F0F AA	0448	DB	CMD-T
1F10	0449 * 2		

1F10 B9	0450	DB	MG-T
1F11 B0	0451	DB	ETMS-T
1F12 7B	0452	DB	SPLIT-T
1F13 B6	0453	DB	MTME-T
1F14 B6	0454	DB	ES-T
1F15 B6	0455	DB	ES-T
1F16 BF	0456	DB	SUN-T
1F17 AA	0457	DB	CMD-T
1F18	0458 * 3		
1F18 BC	0459	DB	PS-T
1F19 4E	0460	DB	ET-T
1F1A 7B	0461	DB	SPLIT-T
1F1B 56	0462	DB	MT-T
1F1C B6	0463	DB	ES-T
1F1D B6	0464	DB	ES-T
1F1E BF	0465	DB	SUN-T
1F1F AA	0466	DB	CMD-T
1F20	0467 * 4		
1F20 BC	0468	DB	KLYT-T
1F21 4E	0469	DB	ET-T
1F22 4D	0470	DB	XTRA-T
1F23 4D	0471	DB	XTRA-T
1F24 B6	0472	DB	ES-T
1F25 B6	0473	DB	ES-T
1F26 BF	0474	DB	SUN-T
1F27 AA	0475	DB	CMD-T
1F28	0476 * 5		
1F28 BC	0477	DB	PS-T
1F29 4E	0478	DB	ET-T
1F2A 4D	0479	DB	XTRA-T
1F2B 4D	0480	DB	XTRA-T
1F2C B6	0481	DB	ES-T
1F2D B6	0482	DB	ES-T
1F2E BF	0483	DB	SUN-T
1F2F AA	0484	DB	CMD-T
1F30	0485 * 6		
1F30 B9	0486	DB	MG-T
1F31 B0	0487	DB	ETMS-T
1F32 4D	0488	DB	XTRA-T
1F33 B9	0489	DB	ME-T
1F34 B6	0490	DB	ES-T
1F35 B6	0491	DB	ES-T
1F36 BF	0492	DB	SUN-T
1F37 AA	0493	DB	CMD-T
1F38	0494 * 7		
1F38 BC	0495	DB	PS-T
1F39 4E	0496	DB	ET-T
1F3A 4D	0497	DB	XTRA-T
1F3B 4D	0498	DB	XTRA-T
1F3C B6	0499	DB	ES-T

1F3D 86	0500	DB	ES-T
1F3E 8F	0501	DB	SUN-T
1F3F AA	0502	DB	CMG-T
1F40	0503	*	
1F40	0504	*	ROUTINES FOR EACH WORD
1F40	0505	*	
1F40 C3 62 1D	0506	MNFR	JMP MINOR
1F43	0507	*	
1F43 3A 38 20	0508	NULL	LDA BVCHK CHECK IF ENABLED
1F46 FE AA	0509	CPI	0AAH
1F48 C0	0510	RNZ	.
1F49 2A 36 20	0511	LHLD	RFVVECT
1F4C E9	0512	PCHL	
1F4D	0513	*	
1F4D C9	0514	XTRA	RET .
1F4E	0515	*	
1F4E 3E 01	0516	ET	MVI A,1 GET 2 BYTES
1F50 CD C9 13	0517	CALL	ELETELEM
1F53 C3 D5 00	0518	JMP	TMOUT AND OUTPUT 'EM
1F56	0519	*	
1F56 3E 01	0520	MT	MVI A,1 GET 2 MAG BYTES
1F58 CD 5F 0D	0521	CALL	MAGTELEM
1F5B C3 D5 00	0522	JMP	TMOUT
1F5E	0523	*	
1F5E	0524	*	IN FIRST SPLIT, GET 1 BYTE FROM ELE.
1F5E	0525	*	CHECK WHETHER MAG OR DSC GIVES THE OTHER.
1F5E	0526	*	
1F5E 97	0527	SPLG	SUB A GET A BYTE OF ELE
1F5F CD C9 13	0528	CALL	ELETELEM
1F62 E5	0529	PUSH	H
1F63 3A 1D 20	0530	LDA	FFRAME IF ODD MINOR
1F66 0F	0531	RRC	. THEN GET AN EXP BYTE
1F67 DA 71 1F	0532	JC	SPLXP
1F6A CD BF 1F	0533	CALL	DSC GET A DSC BYTE IN L
1F6D 6F	0534	MOV	L,A
1F6E C3 81 1F	0535	JMP	JOIN AND JOIN BYTES
1F71 CD E7 1D	0536	SPLXP	CALL EXPTL GET EXP BYTE IN L
1F74 6F	0537	MOV	L,A
1F75 C3 81 1F	0538	JMP	JOIN
1F78	0539	*	
1F78 97	0540	SPLIT	SUB A GET 1 BYTE OF E-FIELD
1F79 CD C9 13	0541	CALL	ELETELEM IN L
1F7C E5	0542	PUSH	H
1F7D 97	0543	SPLMAG	SUB A GET 1 FROM MAG
1F7E CD 5F 0D	0544	CALL	MAGTELEM
1F81	0545	*	
1F81 D1	0546	JOIN	POP D PUT TOGETHER
1F82 63	0547	MOV	H,E
1F83 C3 D5 00	0548	JMP	TMOUT AND OUTPUT 'EM
1F86	0549	*	

1F86 C3 C6 13	0550 ES	JMP	ELESAMP
1F89 C3 56 0D	0551 M6	JMP	MAGGAIN
1F8C C3 9B 0E	0552 PS	JMP	PLASAMP
1F8F CD 88 1D	0553 SUN	CALL	SUNINC
1F92 3A 23 20	0554	LDA	ANGLE IF THE ANGLE CHANGED,
1F95 21 39 20	0555	LXI	H,OLDANG TAKE THE VECTOR
1F98 BE	0556	CMP	M
1F99 77	0557	MOV	M,A
1F9A C8	0558	RZ	.
1F9B CD EB 11	0559	CALL	FITSMP IF ANGLE CHANGED, FIT SAMPLE
1F9E 3A 23 20	0560	LDA	ANGLE
1FA1 CD 64 18	0561	CALL	EXEANG ELSE TELL EXECUTIVE
1FA4 3A 23 20	0562	LDA	ANGLE AND THE SWEEP
1FA7 C3 E7 17	0563	JMP	SWPANG
1FAA	0564 †		
1FAA CD 33 13	0565 CMD	CALL	SAWSTEP
1FAD C3 3A 1E	0566	JMP	CMDEXEC
1FB0 CD 4E 1F	0567 ETMS	CALL	ET
1FB3 C3 59 0D	0568	JMP	MAGSAMP
1FB6 CD 56 1F	0569 MTME	CALL	MT
1FB9 C3 5C 0D	0570 ME	JMP	MAGENCD
1FBC C3 A2 1E	0571 KLYT	JMP	AUTO
1FBF	0572 †		
1FBF	0573 †	OUTPUT THE DIGITAL SUBCOMMUTATOR VALUES	
1FBF	0574 †		
1FBF 21 1D 20	0575 DSC	LXI	H,FRAME DECIDE WHETHER TO
1FC2 7E	0576	MOV	A,M OUTPUT THE FRAME COUNTER.
1FC3 E6 1F	0577	ANI	31 THIS IS DONE ON NEW MAJORS
1FC5 C2 D3 1F	0578	JNZ	DSCI
1FC8 66	0579	MOV	H,M PUT THE MAJOR FRAME NUMBER
1FC9 3A 1E 20	0580	LDA	CYCLE TOGETHER WITH HIGHER
1FCC 29	0581	DAD	H TIME BITS. 3 FROM THE FRAME
1FCD 17	0582	RAL	. AND 5 FROM THE NEXT BYTE
1FCE 29	0583	DAD	H
1FCF 17	0584	RAL	.
1FD0 29	0585	DAD	H
1FD1 17	0586	RAL	.
1FD2 C9	0587	RET	
1FD3	0588 †		
1FD3 7E	0589 DSCI	MOV	A,M COMPUTE THE DSC INDEX
1FD4 E6 E0	0590	ANI	0E0H BY REMOVING THE TIMES
1FD6 0F	0591	RRC	. WE OUTPUT THE FRAME COUNT
1FD7 0F	0592	RRC	. PUT -(MAJOR-1) INTO LOW
1FDB 0F	0593	RRC	. BITS AND ADD FRAME COUNT
1FD9 0F	0594	RRC	.
1FDA 2F	0595	CMA	
1FDB 86	0596	ADD	M
1FDC	0597 †		
1FDC 0F	0598	RRC	. CONVERT TO VALUES 0 TO 119
1FDD E6 7F	0599	ANI	127

1FDF 21 CF 1E	0600	LX1	H,OSCTAR	CHECK TABLE FOR WHERE TO GET
1FE2 BE	0601	DSCF	CMF	M THE OSC BYTE. IF < M, GET
1FE3 0A ED 1F	0602	JC	DSCGD	IT FROM THE ROUTINE
1FE6 96	0603	SUB	M	ELSE DECREMENT THAT PART OF
1FE7 23	0604	INX	H	THE OSC INDEX. (EACH ROUTINE EXPECTS
1FE8 23	0605	INX	H	TO SEE 0-N IN ACCUM).
1FE9 23	0606	INX	H	
1FEA C3 E2 1F	0607	JMP	OSCF	
1FED 23	0608	DSCGD	INX	H
1FEE 5E	0609	MOV	E,M	
1FEF 23	0610	INX	H	
1FF0 56	0611	MOV	D,M	
1FF1 EB	0612	XCHG	.	
1FF2 E9	0613	PCHL	.	
1FF3	0614	‡		
1FF3	0615	‡	BKG	MODULE DIGITAL SUBCOM
1FF3	0616	‡		
1FF3 21 1E 20	0617	BKGDSC	LX1	H,CYCLE
1FF6 DF	0618	RST	REF/B	
1FF7 7E	0619	MOV	A,M	
1FF8 C9	0620	RET	.	
1FF9 00	0621	DB		257 BACKGROUND-END
1FFA	0622	‡		
1FFA	0623	‡	VARIABLES	
1FFA	0624	‡		
1FFA	0625	ORG	BKGRAM	
201C	0626	WORD	DS	1 WORD COUNTER
201D	0627	FRAME	DS	1 MINOR/MAJOR FRAME COUNTER
201E	0628	CYCLE	DS	1 SUB-COM CYCLE COUNTER
201F	0629	HR225	DS	1 2.25 HOUR COUNTER
2020	0630	DAY24	DS	1 24 DAY COUNTER
2021	0631	‡		
2021	0632	GOODCNT	DS	1 GOOD COMMAND COUNTER
2022	0633	BADCNT	DS	1 BAD COMMAND COUNTER
2023	0634	‡		
2023	0635	ANGLE	DS	2 SUN ANGLE
2025	0636	SUNPER	DS	2 SUN PERIOD
2027	0637	‡		
2027	0638	SLEN1	DS	1 LENGTH MEASUREMENTS
2028	0639	SLEN2	DS	1
2029	0640	STMP1	DS	1 TEMPERATURE MEASUREMENTS
202A	0641	STMP2	DS	1
202B	0642	STMP3	DS	1
202C	0643	STMP4	DS	1
202D	0644	‡		
202D	0645	STATUS	DS	1 COMMAND READY STATUS BYTE
202E	0646	SUNCTR	DS	2 COUNT OF SUNINC CALLS
2030	0647	SUNDOWN	DS	2 DOWN COUNTER
2032	0648	KLYCNT	DS	1 KELLEY GAIN TIMER
2033	0649	DIGREG	DS	2 DIGIT COMMAND REGISTER

2035	0650	CMOCNT DS	1	COMMAND COUNT COMPARE REG
2036	0651	BKGVECT DS	2	BACKGROUND VECTOR
2038	0652	BVCHK DS	1	CHECK BYTE FOR VECTOR
2039	0653	OLOANG DS	1	OLO SUN ANGLE
203A	0654	EXPAOR DS	2	EXP OUTPUT MEM ADDRESS
203C	0655	EXPCNT DS	2	EXP OUTPUT COUNTER
203E	0656	†		
203E	0657	† EXTERNAL REFERENCES		
203E	0658	†		
203E	0659	ORG	ELE	
1300	0660	ELEINI DS	3	INITIALIZATION
1303	0661	ELEFRAME OS	3	MINOR FRAME
1306	0662	ELESAMP OS	3	SAMPLE TIME
1309	0663	ELETELEM DS	3	TELEMETRY TIME
130C	0664	ELEDSC DS	3	DIGITAL SUBCOM
130F	0665	†		
130F	0666	ORG	MAG	
0050	0667	MAGINIT OS	3	
0053	0668	MAGFRAME DS	3	
0056	0669	MAGGAIN OS	3	
0059	0670	MAGSAMP OS	3	
005C	0671	MAGENCO OS	3	
005F	0672	MAGTELEM OS	3	
0062	0673	†		
0062	0674	ORG	PLA	
0E98	0675	PLAINIT DS	3	
0E9B	0676	PLASAMP OS	3	
0E9E	0677	PLAOSC OS	3	
0EA1	0678	†		
0EA1	0679	ORG	DEP	
16C4	0680	DEPINT DS	3	
16C7	0681	DEPSAMP OS	3	
16CA	0682	DEPOSC OS	3	
16C0	0683	†		
16CD	0684	ORG	LO	
1680	0685	LOINIT OS	3	
1683	0686	†		
1683	0687	ORG	FIT	
11E8	0688	FITINIT DS	3	
11E8	0689	FITSMP OS	3	
11EE	0690	FITTEL DS	3	
11F1	0691	†		
11F1	0692	ORG	BUR	
1008	0693	BURINIT DS	3	
1008	0694	BURSAMP DS	3	
100E	0695	BURTELEM OS	3	
1011	0696	BURDSC DS	3	
1014	0697	†		
1014	0698	ORG	SAW	
1330	0699	SAWINIT DS	3	

1333	0700	SAWSTEP DS	3
1336	0701	SAWDSC DS	3
1339	0702	↓	
1339	0703	ORG SWP	
17E4	0704	SWPINIT DS	3
17E7	0705	SWPANG DS	3
17EA	0706	SWPSTAT DS	3
17ED	0707	SWPEXEC DS	3
17F0	0708	SWPDSC DS	3
17F3	0709	↓	
17F3	0710	ORG EXEC	
1B64	0711	EXEANG DS	3
1B67	0712	EXEDSC DS	3
1B6A	0713	↓	
1B6A	0714	↓ DEFINE COMMANDS	
1B6A	0715	↓	
1B6A	0716	ORG DIGIT/4+CMDTAB	
007C 81 1E	0717	DW DIGCMD	
007E	0718	↓	
007E	0719	ORG CSUM/4+CMDTAB	
0072 77 1E	0720	DW CSCMD	
0074	0721	↓	
0074	0722	ORG RESET/4+CMDTAB	
005C EA 1E	0723	DW SOFTRESET	

8000: C3 6D 18 C3 E0 1C 00 00 36 00 23 0D C2 08 00 C8
8010: 1A 77 13 23 0D C3 0F 00 B5 6F D0 24 B7 C9 00 00
8020: C3 C1 1D 00 C3 00 00 00 00 00 00 00 F5 C3 2D 1E
8030: C3 61 1E 00 F5 C3 32 1D C9 00 00 00 F5 C3 42 1D
8040: 63 01 80 01 76 01 98 02 AE 02 DD 13 E7 13 DE 1B
8050: 2E 14 ED 13 13 12 46 13 18 18 F7 13 EA 1E 00 00
8060: 00 00 00 00 C9 1B 00 00 2A 10 24 10 44 10 44 10
8070: 00 00 77 1E DC 16 B5 0E 00 00 87 16 B1 1E 7D 0D
8080: 21 01 00 E6 0F C8 29 3D C3 85 00 7C 2F 67 7D 2F
8090: 6F 23 C9 7C D3 01 7D D3 00 C9 3E 20 32 00 20 CD
80A0: 85 03 3E 40 30 00 00 00 97 CD 18 03 3E 01 CD 18
80B0: 03 3E 3F D3 F3 32 14 20 3E 01 32 01 50 CD FA 02
80C0: 97 32 15 20 32 16 20 32 17 20 C3 08 02 20 C9 30
80D0: C9 2A FF AF C9 22 FF AF C9 DB 80 E6 80 21 02 20
80E0: BE 77 C9 DB 90 C9 6F 3E 0D 30 7D F6 80 D3 E0 3E
80F0: 19 3D C2 F1 00 7D D3 E0 3E 90 D3 F3 3E FF D3 50
8100: 76 3E 10 D3 F3 3E 08 30 7D 2A 00 50 FE 07 CA 16
8110: 01 FE 0D C2 20 01 7D E6 F0 6F DB 80 E6 0F B5 6F
8120: 7C E6 0F 67 3E F9 D3 E0 FB C9 57 FE 07 CA 5E 01
8130: FE 0D CA 5E 01 FE 10 DA 4C 01 FE 2E DA 5B 01 11
8140: 30 0E CA 48 01 11 40 0F 7B C3 4F 01 F6 10 5F CD
8150: E6 00 7C FE 07 C8 FE 08 C8 53 C9 E6 EF 57 CD E6
8160: 00 97 C9 1E 05 3E 03 CD AF 01 FE 02 2E 0E DA 87
8170: 01 2E 07 C3 B7 01 1E 08 CD AD 01 2E 08 C3 B7 01
8180: 1E 09 CD AD 01 2E 0D F5 CD 3D 02 E6 F0 B5 6F 7C
8190: EE 7F 67 F1 CD B1 02 2E 14 0F 7D D2 A0 01 F6 20
81A0: F3 D3 F3 E6 EF D3 F3 F6 10 D3 F3 FB C9 3E 01 25
81B0: A4 F5 B3 5F 16 20 7D 12 F1 65 C9 F5 CD 4E 02 F1
81C0: 17 FE 04 DA 02 02 FE 0E DA FD 01 FE 24 DA EF 01
81D0: 2E 01 FE 2B DA DB 01 2D F5 1F 7D CD BB 01 F1 D6
81E0: 20 1F 37 F5 CD BB 01 CD A7 03 F1 B7 C3 BB 01 F5
81F0: 1F 3E 01 CD BB 01 F1 CD 02 02 C3 08 02 0F 07 D2
8200: 08 02 CD 19 02 C3 08 02 CD 3D 02 67 F6 0F 6F CD
8210: B1 02 3E 14 D3 F3 C3 A7 03 B7 1F D6 02 DA 29 02
8220: E6 0F C6 18 E6 27 C3 2C 02 3A 01 20 E6 3F 32 01
8230: 20 5F 3A 15 20 E6 03 0F 0F EE C0 B3 C9 CD 45 02
8240: 3E 10 C3 2C 02 3A 15 20 E6 B3 32 15 20 C9 F5 FE
8250: 02 D4 45 02 F1 11 15 20 F5 FE 08 DA 64 02 13 D6
8260: 08 C3 59 02 CD 80 00 EB F1 7B DA 71 02 2F A6 77
8270: C9 B6 77 C9 21 03 20 B5 6F 7E C9 C5 0E 08 C3 84
8280: 02 C5 0E 10 F5 29 3E 08 17 F3 D3 F3 F6 02 D3 F3
8290: FB 0D C2 85 02 F1 C1 C9 1E 03 CD AD 01 F5 EB 7A
82A0: EE 7F 57 2E 01 CD 00 03 F1 2E 18 C3 99 01 1E 0D
82B0: 3E 07 CD AF 01 54 3C E6 07 CD 80 00 CD 00 03 3E
82C0: 18 CD A0 01 3E 38 CD A0 01 C3 FA 02 C5 5D 21 F2
82D0: 02 E6 07 4F 06 00 09 3A 14 20 CD C1 03 32 14 20
82E0: C1 C3 FA 02 E6 07 5F 16 00 21 F2 02 19 3A 14 20
82F0: A6 C9 20 01 06 08 21 27 2F 10 21 00 00 11 00 00
8300: E5 3A 14 20 67 CD 7B 02 E1 EB CD 7B 02 7B 2F 67

8310: CD 7B 02 3E 18 D3 F3 C9 8F CD 80 00 CD 49 03 3E
8320: 0A CD A7 03 3D C2 21 03 C3 49 03 2E 80 C3 46 03
8330: E5 2E 40 CD 49 03 E1 7C 2F D3 D0 7D 2F D3 90 2E
8340: 40 C3 49 03 2E 20 CD 49 03 3A 00 20 AD D3 C0 32
8350: 00 20 C9 20 07 DC 7E 03 E5 C5 01 10 00 3E C0 30
8360: 05 37 CA 77 03 20 07 D2 60 03 00 3E 4D 30 3E 80
8370: 29 1F 30 0D C2 6E 03 3E 46 30 FB C1 E1 C9 E5 CD
8380: 84 03 E1 C9 20 E6 80 C8 C5 EB 01 10 00 00 3E CD
8390: 30 20 04 A8 FA 91 03 0A 20 07 18 0D C2 97 03 EB
83A0: 3E 48 30 B7 C1 FB C9 F5 D5 11 FE 01 18 7B 82 C2
83B0: AC 03 D1 F1 C9 3A 00 20 F6 10 D3 C0 EE 16 D3 C0
83C0: C9 4F 7E 2F A1 4F EB 1A 0F DA D0 03 29 C3 C8 03
83D0: 1A A5 B1 C9 4E 23 56 23 5E C9 71 23 72 23 73 C9
83E0: 7A B7 C8 46 23 7E B7 CA 70 05 23 6E 67 78 A9 F2
83F0: FD 03 CD B1 05 CD FD 03 79 EE 90 4F C9 78 81 D6
8400: 40 FA 76 05 4F CD F0 05 C3 5C 05 7A B7 C8 46 23
8410: 7E B7 CA 7B 05 23 6E 67 78 A9 F2 26 04 CD 81 05
8420: CD 26 04 C3 F8 03 79 90 C6 40 FA 76 05 4F C5 7C
8430: 2F 47 7D 2F 4F 03 62 6B 09 0A 56 04 EB 3E 10 CD
8440: 6F 04 29 DA 4C 04 09 DA 4C 04 C1 C9 C1 1C C0 14
8450: C0 11 00 80 0C C9 3E 10 11 FF FF CD 6F 04 C1 0C
8460: 37 7A 1F 57 7B 1F 5F D0 C3 40 04 33 33 3D C8 29
8470: DA 81 04 EB 29 E8 1C E5 09 DA 6B 04 E1 1D C3 6D
8480: 04 EB 29 EB 09 1C C3 6D 04 C5 D5 CD 98 04 7A B7
8490: CA 95 04 79 07 D1 C1 C9 7E EE 80 47 C3 A0 04 46
84A0: 23 7E 23 6E 67 97 BC C8 DA CA 0C 05 79 90 87 F2
84B0: B8 04 78 41 4F EB 90 87 CA C2 04 0F FE 10 D0 CD
84C0: F4 04 78 A9 FA D2 04 19 EB D0 7A 1F 57 7B 1F 5F
84D0: 0C C9 78 95 7A 9C DA E7 04 57 7B 95 5F 21 00 00
84E0: B2 C2 21 05 0E 00 C9 7D 93 5F 7C 9A 57 48 21 00
84F0: 00 C3 21 05 D6 08 DA FF 04 6C 26 00 C8 D6 08 C5
8500: 47 97 29 8F 04 C2 02 05 C1 6C 67 C9 EB 48 C9 7A
8510: B7 0E 60 F2 21 05 CD 69 06 CD 21 05 79 F6 80 4F
8520: C9 79 B7 F2 2F 05 E6 7F 4F CD 2F 05 C3 F8 03 7A
8530: B7 C2 59 05 B3 C2 4E 05 B4 C2 48 05 B5 C2 42 05
8540: 4A C9 55 06 18 C3 53 05 EB 06 10 C3 53 05 53 5C
8550: 65 06 08 79 90 4F DA 70 05 7A 6C 63 B7 FA 66 05
8560: 0D 29 9F F2 60 05 57 5C 7D 07 DC 4D 04 79 87 F0
8570: 0E 00 11 00 00 C9 FE C0 D2 70 05 0E 7F 11 FF FF
8580: C9 78 E6 7F 47 79 E6 7F 4F C9 79 EE 80 FA 97 05
8590: 4F CD 97 05 C3 69 06 E6 7F FE 41 DA C5 05 FE 60
85A0: D2 CC 05 21 00 00 D6 50 CB EB D2 BC 05 C6 10 CD
85B0: BC 05 EB 11 00 00 C9 29 2C EB 3D C8 29 EB DA B7
85C0: 05 29 C3 B9 05 11 00 00 21 00 00 C9 11 FF 7F 21
85D0: FF FF C9 7A B7 C8 41 62 6B C3 ED 03 7A B7 C8 79
85E0: E6 01 C4 CA 04 C5 CD 44 06 C1 79 0F C6 20 4F C9
85F0: 97 8B CA 08 06 B5 CA 09 06 E5 CD 0A 06 6C 67 E3
8600: 7C CD 0A 06 D1 19 88 C9 EB 7C 21 00 00 44 87 D2
8610: 14 06 19 88 29 8F D2 1B 06 19 88 29 8F D2 22 06
8620: 19 88 29 8F D2 29 06 19 88 29 8F D2 30 06 19 88

8630: 29 8F D2 37 06 19 88 29 8F D2 3E 06 19 88 29 8F
8640: D0 19 88 C9 01 00 80 CD 54 06 78 0F 47 D2 47 06
8650: 51 1E 00 C9 D5 78 81 5F 16 00 C5 CD 0A 06 C1 D1
8660: 78 95 7A 9C D8 78 81 4F C9 CD 77 06 E8 CD 77 06
8670: EB 23 7C 85 C0 13 C9 7C 2F 67 7D 2F 6F C9 22 92
8680: 24 EB 22 94 24 32 90 24 3C 32 91 24 C9 3E 01 32
8690: 97 24 CD FF 06 D8 CD 5D 07 21 97 24 34 3A 90 24
86A0: BE C2 92 06 3A 90 24 32 9A 24 67 3A 91 24 6F CD
86B0: D7 07 3A 9A 24 6F 67 CD E3 07 2A 94 24 CD BC 07
86C0: CD DA 03 3A 9A 24 3D C8 32 96 24 67 3A 9A 24 6F
86D0: CD D7 07 2A 94 24 CD BC 07 CD E0 03 79 EE 80 4F
86E0: 3A 96 24 67 3A 91 24 6F E5 CD E9 07 E1 CD DD 07
86F0: 3A 96 24 3E C2 C8 06 3A 9A 24 3D C2 A7 06 C9 3A
8700: 97 24 32 96 24 67 3A 97 24 6F CD EF 07 C2 1D 07
8710: 3A 96 24 3C 21 91 24 BE C2 02 07 37 C9 3A 96 24
8720: 21 97 24 BE C8 7E 32 98 24 6F 3A 96 24 67 CD C8
8730: 07 EB 3A 97 24 67 3A 98 24 6F CD C8 07 CD 4F 07
8740: 3A 98 24 3C 21 91 24 BE DA 26 07 CA 26 07 C9 0E
8750: 03 1A 46 EB 12 70 23 13 0D C2 51 07 C9 3A 97 24
8760: 3C 32 98 24 67 3A 97 24 6F CD D7 07 3A 97 24 67
8770: 6F CD E3 07 21 9C 24 CD DA 03 3A 97 24 32 99 24
8780: 6F 3A 97 24 67 CD D7 07 21 9C 24 CD E0 03 79 EE
8790: 80 4F 3A 98 24 67 3A 99 24 6F E5 CD E9 07 E1 CD
87A0: DD 07 3A 99 24 3C 21 91 24 BE DA 7D 07 CA 7D 07
87B0: 3A 98 24 3C 21 91 24 BE C2 61 07 C9 2A 94 24 3A
87C0: 9A 24 3D 47 87 80 DF C9 25 2D 7C 87 87 84 85 6F
87D0: 87 85 2A 92 24 DF C9 CD C8 07 C3 D4 03 CD C8 07
87E0: C3 DA 03 CD C8 07 C3 08 04 CD C8 07 C3 9F 04 CD
87F0: D7 07 7A B7 C8 79 E6 7F FE 37 D0 97 C9 21 91 08
8800: FE 33 DA 07 08 D6 30 DF C9 CD 16 08 C3 12 08 CD
8810: 1D 08 3E 33 DF C9 D6 18 D2 1D 08 C6 60 21 2B 08
8820: FE 33 DA 29 08 2F 3C C6 60 DF C9 41 80 00 40 FB
8830: 15 40 EC 83 40 D4 D8 40 B5 04 40 BE 3A 3F C3 EE
8840: 3E C7 C5 00 00 00 BE C7 C5 BF C3 EE C0 8E 3A C0
8850: 85 04 C0 D4 D8 C0 EC 83 C0 FB 15 C1 80 00 41 80
8860: 00 40 F6 43 40 DA 83 40 B0 FC 40 80 00 3F 9E 08
8870: 3E 95 F6 3C 9B E5 00 00 00 3C 98 E5 3E 95 F6 3F
8880: 9E 08 40 80 00 40 B0 FC 40 DA 83 40 F6 43 41 80
8890: 00 00 00 00 3E C3 EF 3F 85 04 3F EC 83 40 80 00
88A0: 3F EC 83 3F 85 04 3E C3 EF 00 00 00 BE C3 EF BF
88B0: 85 04 BF EC 83 C0 80 00 BF EC 83 BF 85 04 8E C3
88C0: EF 00 00 00 22 20 23 EB 22 22 23 C5 3E C3 32 28
88D0: 23 CD E3 08 CD 0E 0C CD FB 08 D1 21 35 23 0E 10
88E0: C3 88 0C 3E 04 11 35 23 CD 13 09 CD 24 09 CD AE
88F0: 09 CD 85 0B CD E1 09 C2 EE 08 C9 3E 02 11 3B 23
8900: CD 13 09 CD 62 09 CD D3 09 CD 85 08 CD ED 09 C2
8910: 06 09 C9 21 48 24 CD 7E 06 3E 03 CD DB 0C 11 46
8920: 23 C3 E6 0C CD 6D 0A 32 44 23 32 45 23 32 2D 23
8930: 21 39 09 CD 10 0D C3 6A 0B CD C1 0C 5E 23 56 E5
8940: CD 9F 0C E1 7E E6 10 C2 53 09 CD D0 0C CD DA 03

8950: C3 E9 0A 2A 22 23 CD E0 03 CD D0 0C CD DA 03 C3
8960: C2 0A 3E 80 32 2D 23 01 18 24 11 3F 24 21 35 23
8970: CD 94 09 01 18 24 11 30 24 21 38 23 CD 94 09 01
8980: 27 24 11 42 24 21 35 23 CD 94 09 01 27 24 11 33
8990: 24 21 38 23 C5 D5 CD D4 03 7A B7 CA AB 09 FE 40
89A0: DA AB 09 E1 CD E0 03 E1 C3 7E 0C E1 E1 C9 11 48
89B0: 24 21 0C 24 0E 3C CD DB 09 3A 45 23 B7 C2 C5 09
89C0: 3E 40 32 36 23 3A 44 23 21 45 23 96 C0 3E 40 32
89D0: 39 23 C9 11 48 24 21 12 24 0E 18 CD 88 0C C3 8D
89E0: 06 21 19 0A CD 0F 0A CD 0A 08 C3 F3 09 21 29 0A
89F0: CD 0F 0A 21 46 23 CD D4 03 3E 96 CD DB 0C CD 9F
8A00: 04 21 46 23 CD DA 03 3A 2E 23 21 44 23 96 C9 3A
8A10: 44 23 32 2E 23 3E 80 32 2D 23 C3 10 0D CD 33 0A
8A20: D0 E6 10 DA E9 0A C3 C2 0A CD 33 0A D0 CD 4D 0B
8A30: C3 7D 0A CD C1 0C 23 7E FE 20 D0 E6 10 21 45 23
8A40: 7E C2 48 0A 3A 44 23 96 FE 03 DA 63 0A CD CA 0C
8A50: CD D4 03 7A B7 C8 79 E6 7F 4F 21 49 23 BE CC 89
8A60: 04 3F D0 CD C1 0C 23 7E F6 20 77 37 C9 21 0B 24
8A70: 0E 14 11 03 00 97 77 19 0D C2 76 0A C9 21 44 23
8A80: CD 88 0C CD F5 0C 21 12 24 CD 7E 0C CD 07 0D 21
8A90: 15 24 CD 7E 0C 21 21 24 CD DA 03 CD FE 0C 21 24
8AA0: 24 CD 7E 0C CD E3 0C CD D0 0C CD E0 03 21 18 24
8AB0: CD 7E 0C CD EC 0C CD D0 0C CD E0 03 21 27 24 C3
8AC0: 7E 0C 21 45 23 CD 88 0C CD E3 0C 21 3F 24 CD 7E
8AD0: 0C CD EC 0C 21 42 24 CD 7E 0C CD B0 0C CD D4 03
8AE0: 21 45 24 CD 7E 0C C3 7D 0A CD E3 0C 21 30 24 CD
8AF0: 7E 0C CD EC 0C 21 33 24 CD 7E 0C CD D0 0C CD D4
8B00: 03 21 36 24 CD 7E 0C C3 7D 0A 21 3F 24 11 0C 24
8B10: CD B6 0C 11 1B 24 CD 86 0C 21 30 24 11 0F 24 CD
8B20: E6 0C 11 1E 24 CD 86 0C 3A 45 23 B7 C2 30 0B 3C
8B30: CD 96 0C 21 39 24 CD DA 03 3A 44 23 21 45 23 96
8B40: C2 44 0B 3C CD 96 0C 21 2D 24 C3 DA 03 21 36 23
8B50: E6 10 C2 5B 0B 21 39 23 7E B7 C8 FE 40 C8 2B E5
8B60: CD D4 03 3A 2B 23 CD 1D 0B CD E0 03 21 18 24 CD
8B70: 7E 0C E1 CD D4 03 3A 2B 23 CD 16 0B CD E0 03 21
8B80: 27 24 C3 7E 0C 97 32 42 23 3A 44 23 3D CD 96 0C
8B90: 21 2F 23 CD DA 03 21 8D 0B CD 10 0D 21 41 23 CD
8BA0: D4 03 21 2F 23 CD 0B 04 CD DC 05 21 41 23 CD DA
8BB0: 03 21 46 23 CD E0 03 21 49 23 C3 DA 03 CD C1 0C
8BC0: 23 7E FE 20 D0 F5 CD E3 0C 21 38 23 CD E0 03 21
8BD0: 32 23 CD DA 03 CD EC 0C 21 3E 23 CD E0 03 21 32
8BE0: 23 CD 9F 04 F1 E6 10 21 38 23 CA F0 0B 21 35 23
8BF0: CD 9F 04 CD D0 0C CD 98 04 CD CA 0C CD DA 03 CD
8C00: D3 05 21 41 23 CD 9F 04 21 41 23 C3 DA 03 3E 09
8C10: CD BB 0C EB 21 06 00 19 0E 12 CD 88 0C 21 35 23
8C20: 0E 06 CD 88 0C 3E 09 21 35 23 CD 32 0C 3E 0C 21
8C30: 3B 23 E5 0E 00 11 09 00 21 2C 23 36 00 F5 CD DB
8C40: 0C CD 70 0C F1 C6 06 FE 21 DA 3D 0C CD 53 0C E1
8C50: C3 DA 03 3A 2C 23 B7 CA 6D 0C C5 05 CD 96 0C 21
8C60: 32 23 CD DA 03 D1 C1 21 32 23 C3 0B 04 16 40 C9

8C70: 23 7E FE 40 C8 2B CD 9F 04 21 2C 23 34 C9 E5 3A
8C80: 2D 23 A9 4F CD 9F 04 E1 C3 DA 03 3A 2D 23 87 FA
8C90: 94 0C 34 C9 35 C9 6F 26 00 11 00 00 C3 0F 05 21
8CA0: 00 00 7A E6 0F 57 FE 08 DA AE 0C F6 F0 57 CD 0F
8CB0: 05 79 D6 1B 4F C9 0E 03 7E 12 23 13 0D C2 88 0C
8CC0: C9 2A 20 23 3A 2C 23 C3 D6 0C 21 AC 23 C3 D3 0C
8CD0: 21 4C 23 3A 28 23 85 6F D0 24 C9 2A 22 23 85 6F
8CE0: D0 24 C9 3A 2B 23 CD 10 08 C3 D4 03 3A 2B 23 CD
8CF0: 16 08 C3 D4 03 3A 2B 23 CD 0F 08 C3 D4 03 3A 2B
8D00: 23 CD 09 08 C3 D4 03 3A 2B 23 CD FD 07 C3 D4 03
8D10: 22 29 23 97 32 2C 23 32 2B 23 CD 28 23 3A 2C 23
8D20: C6 20 32 2C 23 3A 2B 23 C6 30 32 2B 23 CD 28 23
8D30: 3A 2C 23 D6 1E 32 2C 23 3A 2B 23 D6 2D FE 30 DA
8D40: 17 0D C9 00 00 00 00 00 00 00 00 00 00 00 00
8D50: C3 62 0D C3 96 0D C3 A7 0D C3 EA 0D C3 F7 0D C3
8D60: 85 0D 21 00 FE CD 7D 0D 21 26 29 22 A6 24 97 32
8D70: 2F 21 3E FD 32 2C 21 3E 07 32 2D 21 C9 32 33 21
8D80: 3E 07 C3 CC 02 B7 C4 8F 0D 53 CD 8F 0D EB C9 21
8D90: 2E 21 34 6E 5E C9 E6 03 CA A1 0D FE 02 CA 6E 0D
8DA0: C9 3E FF 32 2E 21 C9 01 E7 0D 11 37 21 CD 84 0E
8DB0: 11 E7 0D 21 34 21 0E 03 D7 01 34 21 2A 37 21 CD
8DC0: CB 0D 2A 39 21 CD C8 0D 2A 38 21 EB 2A A6 24 7A
8DD0: 87 CA DC 0D D6 0F C2 E5 0D 93 5F 65 78 8C D2 E5
8DE0: 0D 0A F6 10 02 03 C9 02 01 00 21 2F 21 34 01 34
8DF0: 21 11 3D 21 C3 84 0E 3A 2F 21 FE 01 CC 43 0E 2A
8E00: 3D 21 CD 58 0E 2A 3F 21 CD 58 0E 2A 41 21 CD 58
8E10: 0E 21 30 21 11 34 21 CD 77 0E CD 77 0E CD 77 0E
8E20: 3A 2F 21 FE 01 CA 38 0E FE 08 C0 2A 30 21 22 29
8E30: 21 3A 32 21 32 2B 21 C9 21 04 21 3A 33 21 E6 0F
8E40: B6 77 C9 2A 37 21 CD 52 0E 2A 39 21 CD 52 0E 2A
8E50: 3B 21 11 2C 21 C3 58 0E 11 2D 21 1A C6 03 12 87
8E60: 1F 5F DA 6E 0E 29 29 29 29 EB 72 23 73 C9 EB 7A
8E70: E6 0F 86 77 23 73 C9 1A E6 10 CA 7E 0E 37 7E 17
8E80: 77 13 23 C9 CD 8A 0E CD 8A 0E 0A 03 CD E6 00 EB
8E90: 73 23 72 23 EB C9 00 00 C3 A8 0E C3 C0 0E 3A 58
8EA0: 21 E6 3F 21 59 21 86 C9 21 50 21 0E 0C CF 11 F3
8EB0: 0F 0E 10 D7 C9 32 58 21 07 D0 E6 80 32 59 21 C9
8EC0: 3A 59 21 E6 80 C0 C5 21 58 21 34 7E 0F D2 F4 0E
8ED0: CD 24 0F 22 50 21 32 56 21 CD 3C 0F 22 54 21 32
8EE0: 57 21 4F 3E 0B EB CD 0A 06 6C 67 79 CD D5 0F 22
8EF0: 52 21 C1 C9 CD 30 0F CD D5 0F EB 2A 52 21 19 CD
8F00: D4 0F 97 32 59 21 C1 29 D8 27 D8 29 D8 3A 50 21
8F10: 95 3A 51 21 9C D8 C5 CD A4 0F CD 30 03 C1 3E 40
8F20: 32 59 21 C9 3A 34 21 2A 3D 21 01 5C 21 C3 5F 0F
8F30: 3A 36 21 2A 41 21 01 68 21 C3 5F 0F 3E 07 CD E4
8F40: 02 CA 56 0F 01 64 21 CD 59 0F 4F 3E 27 EB CD 0A
8F50: 06 6C 67 79 2F C9 01 60 21 3A 35 21 2A 3F 21 E6
8F60: 10 3E 00 C2 6A 0F 3E 33 03 03 F5 7C FE 08 DA 74
8F70: 0F F6 F0 67 0A 5F 03 0A 57 19 7C 32 5A 21 CD D4
8F80: 0F F1 54 5D CC 0A 06 CD 93 0F 87 3A 5A 21 C8 21

BF90: FF 7F C9 B7 1F 23 CD 9B 0F B7 1F 4F 7C 1F 67 7D
9FA0: 1F 6F 79 C9 2A 50 21 EB 2A 54 21 B7 CD 9B 0F CD
BFB0: DA 0F 3C B7 1F FE 40 DA BE 0F E6 3F F6 B0 5F 3A
BFC0: 56 21 21 57 21 AE 7B F2 CC 0F F6 40 6F 3A 5B 21
BFD0: F6 B0 57 C9 7C B7 F0 C3 B8 00 7A 2F 47 7B 2F 4F
BFE0: 03 3E 01 C3 E9 0F 17 B8 29 54 5D 09 DA E6 0F EB
BFF0: C3 E6 0F A5 FF FC FF F4 FF FE FF 60 00 01 00 EA
9000: FF FE FF 00 00 00 00 00 C3 17 10 C3 65 10 C3 38
9010: 11 21 70 21 0F 7E C9 97 32 70 21 21 10 00 22 72
9020: 21 C3 44 03 11 9e 21 C3 2D 10 11 71 21 7C E6 03
9030: EB DF 97 32 71 21 73 3A 71 21 E6 07 C8 3E 40 CD
9040: AE 11 B7 C9 97 32 71 21 CD 4D 10 B7 C9 7C FE B9
9050: CA 17 10 FE B4 CA 3D 10 FE B5 CA E9 10 FE B8 CA
9060: 1F 11 C3 53 03 CD A6 11 0F 0F 0F E6 0E 21 73 10
9070: C3 B5 10 4C 10 CE 10 01 11 16 11 B0 10 A0 10 B9
9080: 10 B3 10 21 79 21 35 C0 3E 40 C3 AE 11 21 00 B4
9090: CD 53 03 97 32 79 21 3A 71 21 32 B6 21 C3 C1 11
90A0: CD B4 03 C8 3A 70 21 E6 F0 B5 32 70 21 CD B4 03
90B0: CA AD 10 22 B0 21 C3 C1 11 CD B4 03 C8 22 B2 21
90C0: 21 B7 21 CD E0 11 3E 10 C3 AE 11 3A 71 21 E6 07
90D0: C8 B7 21 D9 10 DF 7E 23 66 6F E9 E9 10 73 11 BF
90E0: 11 9F 11 9F 11 9F 11 9F 11 21 BC 21 CD E0 11 2A
90F0: 72 21 22 7B 21 97 32 7A 21 32 7D 21 3E 20 C3 AE
9100: 11 CD C9 11 D0 21 60 B5 CD 53 03 21 91 21 CD E0
9110: 11 3E 30 C3 AE 11 CD B4 03 C8 3E 60 CD AE 11 21
9120: 00 B8 CD 53 03 3E B1 32 B5 21 3E B5 32 B4 21 3A
9130: 70 21 F6 B0 32 70 21 C9 21 B4 21 7E FE 9A DA 6D
9140: 11 CD B4 03 C0 3A 70 21 E6 B0 CA 69 11 3A 70 21
9150: E6 7F 32 70 21 3E 10 32 79 21 3A 71 21 E6 B0 3E
9160: 70 C2 66 11 3E 00 CD AE 11 21 00 00 C9 34 6F 6E
9170: 26 00 C9 3A 96 21 CD E6 00 29 29 29 29 7C CD BA
9180: 11 67 3A 97 21 BC DC E9 10 C9 B7 F0 2F 3C C9 97
9190: CD 9E 0E E6 40 C8 21 79 21 BE C8 77 C3 E9 10 3A
91A0: 75 21 FE AA CA 76 21 C9 3A 70 21 E6 70 C9 E6 70
91B0: E5 6F 3A 70 21 E6 6F B5 32 70 21 6F 26 DB F7 E1
91C0: C9 21 70 21 7E C6 10 77 C9 11 7A 21 21 DC 11 B7
91D0: CD D3 11 CD D6 11 1A 9E 12 13 23 C9 20 00 00 00
91E0: 11 1C 20 0E 05 D7 C9 00 C3 F4 11 C3 29 12 C3 A3
91F0: 12 C3 D0 12 21 B0 22 0E 07 CF 21 D7 22 CD 03 12
9200: 21 F8 22 11 20 13 0E 09 D7 11 1B 40 72 23 1D C2
9210: 0C 12 C9 5F 29 A5 57 7B EE 22 21 B6 22 A6 B2 77
9220: E6 02 C0 21 E0 22 C3 09 12 D6 A0 5F E6 07 C0 3A
9230: B6 22 E6 20 C0 7B CD 6B 12 3E 0F CD 90 12 3A B2
9240: 22 11 30 22 CD 9D 12 21 B2 22 CD 64 12 3A B6 22
9250: E6 02 C0 3E 08 CD 90 12 3A B0 22 11 B0 21 CD 9D
9260: 12 21 B0 22 7E C6 02 E6 7F 77 C9 FE B0 21 B2 22
9270: CA 7E 12 B7 C0 3A B6 22 E6 02 C0 21 B0 22 7E E6
9280: 3F CC B9 12 7E E6 C0 77 C9 7E EE 40 2C 77 2D C9
9290: CD 2A 01 7A CD E6 00 7A E6 10 B4 67 C9 EB DF 73
92A0: 23 72 C9 21 B4 22 7E B7 CA B3 12 35 21 B5 22 34

92B0: 6E 7E C9 21 B1 22 16 A1 1E B6 7E FE AA CA D0 12
92C0: 21 B3 22 16 A3 1E C6 7E FE AA CA D0 12 3E 00 C9
92D0: 36 FF 7B 32 B5 22 3E 10 32 B4 22 7A C9 3A B1 22
92E0: B7 F2 FE 12 3A B3 22 B7 F8 21 30 22 CD 13 13 01
92F0: C7 22 11 F8 22 CD C4 08 3E AA 32 B3 22 C9 21 B6
9300: 21 CD 13 13 01 B7 22 11 D7 22 CD C4 08 3E AA 32
9310: B1 22 C9 DF 3A B6 22 E6 08 C8 11 20 00 C3 CF 13
9320: 38 A0 F2 41 E6 66 3F CC CD 00 00 00 00 00 00
9330: C3 3C 13 C3 5F 13 21 A4 21 DF 7E C9 11 B5 13 21
9340: A4 21 0E 06 D7 C9 7C E6 07 5D 21 A4 21 DF 73 C9
9350: 21 A0 21 11 A4 21 0E 04 D7 3E 01 32 A3 21 C9 3A
9360: A5 21 B7 C8 3A 1D 20 FE FE C2 74 13 3A 1C 20 FE
9370: E0 B4 50 13 3A B8 21 0F D0 21 A3 21 35 C0 3A A7
9380: 21 77 3A B8 21 E6 02 C4 A5 13 21 A2 21 35 CA 9A
9390: 13 3A A1 21 21 A0 21 B6 77 C9 3A A6 21 77 21 A1
93A0: 21 97 96 77 C9 2A A0 21 3A A9 21 67 E5 CD 63 01
93B0: E1 24 C3 63 01 00 02 40 02 00 01 00 00 00 00
93C0: C3 B5 13 C3 B8 14 C3 6A 15 C3 31 15 C3 42 15 C3
93D0: 2E 16 3A 77 20 C9 21 50 20 0E A5 CF C9 7C E6 07
93E0: FE 07 3F D8 C3 CC 02 7C 0F 7D C3 B8 01 7C E6 07
93F0: F6 20 67 22 50 20 C9 C6 3E 32 94 20 0F 11 0A 14
9400: D2 06 14 11 1C 14 3E 01 E7 C9 12 31 13 31 07 30
9410: 08 30 09 30 10 30 01 28 01 29 FF FF 07 31 08 31
9420: 09 31 10 31 12 30 13 30 00 28 00 29 FF FF 7C E6
9430: 04 C2 70 14 3E 01 11 52 20 CD 41 14 3E 02 11 54
9440: 20 A4 C8 7D 0F 0F 0F 0F CD 50 14 7B C6 04 5F 7D
9450: E5 CD 5D 14 EB 73 23 72 2B EB E1 B7 C9 21 3D 16
9460: E6 0F FE 0A DA 6A 14 21 F5 1F 87 87 87 87 DF C9
9470: 7C 0F 7D D2 B3 14 5D 3A 5B 20 21 95 20 DF 73 3A
9480: 5B 20 3C FE 60 D0 32 5B 20 B7 C9 21 7E 20 11 6A
9490: 20 0F DA A5 14 EB CD E0 14 CD EE 11 32 B8 20 3E
94A0: FF 32 5A 20 C9 CD E0 14 3A 1D 20 3C E6 07 CC 09
94B0: 15 CD 13 15 97 CD 11 10 E6 B0 5F CD EA 17 E6 01
94C0: 6F 3E 02 E5 E7 E1 29 DF 3A 94 20 E6 41 29 DF B3
94D0: 5F 2A 5E 20 7C B5 CA DB 14 3E B8 B3 32 77 20 C9
94E0: 22 68 20 EB 22 64 20 11 0E 00 19 22 66 20 2A 68
94F0: 20 11 0C 00 19 3A 92 20 77 11 07 00 19 7E E6 F0
9500: 5F 3A 93 20 E6 0F B3 77 C9 21 56 20 CD 1D 15 22
9510: 62 20 C9 21 52 20 CD 1D 15 22 60 20 C9 3A 77 20
9520: E6 01 1E 02 C2 28 15 5F 16 00 19 7E 23 66 6F 2B
9530: C9 2A 68 20 5E 23 B7 CA 3D 15 53 5E 23 22 68 20
9540: EB C9 FE 10 21 52 20 DA 54 15 D6 10 FE 20 21 56
9550: 20 D2 62 15 F5 3A 1E 20 E6 01 CD 22 15 F1 23 C3
9560: 67 15 D6 20 21 50 20 DF 7E C9 21 5A 20 34 7E 0F
9570: DA 92 15 2A 60 20 23 22 60 20 CD DA 15 21 92 20
9580: CD B9 15 2A 64 20 3A 5A 20 E6 02 CD C4 15 22 64
9590: 20 C9 3A 5A 20 E6 02 CA 0B 10 2A 62 20 23 22 62
95A0: 20 CD DA 15 21 93 20 CD B9 15 2A 66 20 3A 5A 20
95B0: E6 04 CD C4 15 22 66 20 C9 7A E6 10 CA C0 15 37
95C0: 7E 17 77 C9 C2 D1 15 EB 29 29 29 29 EB 72 23 73

95D0: C9 7A E6 0F B6 77 23 73 23 C9 3A 77 20 A6 FA 0A
95E0: 16 7E E6 40 C2 FF 15 7E E6 3F CD 2A 01 1E 10 C2
95F0: F4 15 1E 00 7A CD E6 00 7B E6 10 B4 57 5D C9 7E
9600: E6 3F 2A 50 20 DF 5E 23 56 C9 3A 77 20 E6 08 C2
9610: 17 16 CD 0E 10 EB C9 2A 5E 20 7C B5 EB C8 1B EB
9620: 22 5E 20 2A 5C 20 5E 23 56 23 22 5C 20 C9 3A 77
9630: 20 E6 08 37 C0 22 5C 20 EB 22 5E 20 C9 08 0F 88
9640: 8F 08 0F 88 8F 08 0F 88 8F 08 0F 88 8F 86 83 87
9650: 08 84 8C 87 09 86 83 87 0A 84 8C 87 09 86 83 87
9660: 08 84 8C 87 09 86 83 87 0A 84 8C 87 09 00 00 00
9670: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9680: 21 30 29 22 20 29 C9 7C 06 EB CA 9B 16 3D CA A0
9690: 16 3D CA A5 16 3D CA AF 16 37 C9 7D 32 20 29 C9
96A0: 7D 32 21 29 C9 EB 2A 20 29 73 23 22 20 29 C9 3A
96B0: 30 29 85 FE AA C0 97 32 30 29 C3 31 29 00 00 00
96C0: 00 00 00 00 C3 D0 16 C3 0B 17 21 44 20 DF 7E C9
96D0: CD E3 00 32 45 20 21 00 D0 CD 02 17 7C FE D4 CA
96E0: 02 17 E6 FC 06 D0 37 C0 7D 32 48 20 7C E6 03 21
96F0: 07 17 DF 7E 32 44 20 97 32 4A 20 32 46 20 32 47
9700: 20 C9 7D 32 49 20 C9 22 21 12 11 D5 21 4A 20 34
9710: CD 68 17 CD 18 17 D1 C9 CD 93 17 FE 00 C8 FE 01
9720: CA 36 17 FE 02 CA 55 17 CD AB 17 D2 55 17 CD C8
9730: 17 D8 FE 03 D2 5A 17 C9 CD AB 17 3E 00 D2 82 17
9740: CD C8 17 DA 47 17 C0 3E 03 CD 82 17 3A 4A 20 E6
9750: 01 37 C3 18 03 3E 00 C3 5C 17 3E 01 CD 82 17 3A
9760: 4A 20 E6 01 87 C3 18 03 21 45 20 5E CD E3 00 77
9770: AB A6 5F 3E 02 21 46 20 CD 7E 17 3E 20 23 A3 C8
9780: 34 C9 6F 3A 44 20 CD 9F 17 E6 30 B5 CD 9F 17 32
9790: 44 20 C9 3A 44 20 CD 9F 17 E6 03 C9 3A 45 20 5F
97A0: 3A 4A 20 0F 7B D0 0F 0F 0F 0F C9 CD 9C 17 21 49
97B0: 20 B6 6F E6 01 C8 7D E6 04 C8 7D E6 88 EE 88 C0
97C0: CD D6 17 21 48 20 BE C9 3A 44 20 E6 11 FE 11 37
97D0: C0 CD D6 17 95 C9 2A 46 20 3A 4A 20 0F 7C D8 7D
97E0: 6C C9 00 00 C3 04 18 C3 31 18 C3 FD 17 C3 A2 18
97F0: FE 06 DA F7 17 C6 09 21 C3 24 DF 7E C9 3A E2 24
9800: 32 E3 24 C9 21 C1 24 11 2D 1B 0E 11 D7 11 2F 1B
9810: 0E 0F D7 36 01 3E 98 32 D3 24 C9 7C FE 62 CA 7F
9820: 18 EB 21 C0 24 0F D2 2F 18 34 7E FE 22 D0 DF 73
9830: C9 5F 87 C0 67 18 01 C1 24 0A E6 03 CA 6F 18 D6
9840: 02 C0 CD 02 1B 23 7E 93 C0 02 0E 10 CD E2 1A 2A
9850: E1 24 C4 85 18 21 E1 24 35 36 03 CA 60 18 36 01
9860: 3A C2 24 32 E4 24 C9 21 E4 24 35 F0 36 00 C9 3A
9870: E4 24 87 C0 3E 01 CD FC 1A BB C0 3E 01 02 C9 3A
9880: C1 24 FE 03 C6 55 0E 10 CD DC 1A 3E 03 CA 92 18
9890: 3E 04 6A CD F3 1A DF 6E 62 E5 CD 63 01 E1 24 C3
98A0: 63 01 3A C1 24 3D C0 21 10 25 22 EE 24 3E 04 CD
98B0: 36 13 32 14 29 01 01 07 CD DA 18 01 04 09 CD DA
98C0: 18 97 32 E2 24 2A 14 29 26 5C F7 0E 01 CD DC 1A
98D0: C4 87 19 21 C1 24 34 C3 A5 1A CD DC 1A C8 21 E2
98E0: 24 36 01 23 7E 30 C2 E4 18 CD E2 1A 32 15 29 78

9BF0: CD FC 1A CD 1D 1B 22 E9 24 CD 02 1B 79 3C DF EB
9900: 21 EB 24 0E 03 D7 21 01 5C F7 CD 25 19 01 B0 04
9910: CD 4C 19 CD 3D 19 CD 5B 19 06 01 0D C2 10 19 2A
9920: EB 24 CD B5 1B 3A 15 29 B7 CB CD D2 13 3C E6 01
9930: 6F 26 6B F7 C9 3A 1C 20 E6 07 C2 35 19 3A 1C 20
9940: D6 02 E6 07 C2 3D 19 05 C2 35 19 C9 2A EA 24 CD
9950: 99 1B 3A E9 24 21 EA 24 B6 77 C9 3A EC 24 CD 64
9960: 19 3A ED 24 CD 7B 19 29 29 29 29 10 10 10 10 EB
9970: 2A EE 24 73 23 72 23 22 EE 24 C9 B7 F2 E6 00 2F
9980: 3C CD E6 00 C3 BB 00 3A E5 24 FE AA CA E6 24 3E
9990: 02 CD FC 1A 36 00 3E 0B CD FC 1A 36 00 21 10 25
99A0: CD B6 19 F5 21 12 25 CD B6 19 5F F1 B3 1F 5F 3E
99B0: 04 CD FC 1A 73 C9 22 10 29 3E 06 CD FC 1A CD E1
99C0: 19 3E 05 CD FC 1A CD 4C 1A 5F FE FF 3E 03 CA FC
99D0: 1A 3E 07 CD FC 1A 16 00 E5 CD 0A 06 D1 13 1A B5
99E0: C9 32 12 29 3C CB 06 7E CD F3 19 05 7B FE 02 D2
99F0: EB 19 C9 7B CD 0E 1B CB 3A 12 29 DF EB 7B 3D CD
9A00: 0E 1B CB CD 8B 00 19 7C B7 F0 7B CD 0E 1B EB 7B
9A10: 3D CD 0E 1B 19 CD BB 00 EB 7B 3C CD 0E 1B C8 19
9A20: EB 7B D6 02 CD 0E 1B CB 19 7C B7 7B F2 30 1A 3D
9A30: 4F 3C CD 0E 1B EB 79 3D CD 0E 1B 19 10 EB 79 CD
9A40: 22 1B 73 23 72 3E 0B CD FC 1A 34 C9 11 FF 7F 4F
9A50: 3E FF 32 13 29 3E 7E 91 47 D5 CD 74 1A D1 7D 93
9A60: 7C 9A D2 6A 1A EB 7B 32 13 29 05 7B B9 D2 59 1A
9A70: 3A 13 29 C9 7B 3C CD B7 1A DB E5 7B 91 CD B7 1A
9AB0: D1 DB CD BB 00 19 C9 11 00 00 C5 F5 CD 0E 1B CA
9A90: 9E 1A 19 EB F1 3C 0D C2 BB 1A EB C1 B7 C9 F1 C1
9AA0: 21 FF 3F 37 C9 0E 40 CD DC 1A CB CD F0 1A EB 21
9AB0: F2 24 0E 0F CD D0 1A 21 E0 0E 3A E1 24 DF 22 F0
9AC0: 24 2A EE 24 11 10 DB 19 10 EB 21 F0 24 C3 CF 13
9AD0: 1A 13 77 23 36 00 23 0D C2 D0 1A C9 CD F0 1A C3
9AE0: E5 1A CD 02 1B CD D2 13 0F 79 D2 EE 1A B7 A6 C9
9AF0: 2A E1 24 2D 21 C3 24 CB 21 D2 24 C9 CD F0 1A DF
9B00: 7E C9 2A E1 24 2D 21 3E 1B CB 21 46 1B C9 CD 22
9B10: 1B CD 1D 1B 7C FE 07 CB E6 0F FE 0B C9 7E 23 66
9B20: 6F C9 2A 10 29 B7 B7 D2 2B 1B 24 DF C9 00 0B CF
9B30: 5B 00 F0 FF 05 0B 02 B1 02 B1 00 00 00 00 16 9F
9B40: 01 06 FC 01 0B 04 34 1F 03 03 0C 01 0B 04 00 00
9B50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9B60: 00 00 00 00 C3 FD 1B 21 A0 24 DF 7E C9 31 FF 2F
9B70: 21 00 20 01 00 0F CF 05 C2 76 1B CD 9A 00 CD 03
9B80: 00 CD E4 17 CD EB 11 11 40 1C CD 2A 1C 2A AB 24
9B90: 22 AC 24 2A C7 1B 22 B0 24 3A B0 24 FE AA C2 A7
9BA0: 1B CD B1 24 C3 B5 1B 3E 21 32 A0 24 2A C5 1B 22
9BB0: AE 24 CD AE 24 CD F1 11 CD ED 17 CD 0A 1C 97 32
9BC0: B4 24 C3 99 1B 76 C9 55 C9 7C 0F 11 B2 1C D2 2A
9BD0: 1C 3A 1D 20 BD C2 D1 1B C9 3E 9B 32 A2 24 32 A1
9BE0: 24 CD B0 00 2B 22 AB 24 3A A1 24 0F 0F 0F 0F CD
9BF0: B0 00 2B 22 AA 24 21 01 00 22 AC 24 C9 21 A2 24
9C00: BE C0 2A AC 24 2B 22 AC 24 C9 2A AC 24 7C B5 C0

9C10: CD D2 13 E6 01 2A AB 24 11 2E 1C C2 24 1C 2A AA
9C20: 24 11 38 1C 7C B5 C8 22 AC 24 3E 01 E7 C9 02 50
9C30: 00 5C 00 68 01 62 FF FF 03 50 01 68 03 5C FF FF
9C40: 01 43 F0 01 F0 02 F0 03 F0 04 FF 09 FF 0A FF 11
9C50: FF 12 FF 21 FF 22 FF 23 FF 24 FF 25 FF 26 FF 27
9C60: FF 19 FF 1A 0A 31 0B 30 0D 30 0E 30 11 30 14 31
9C70: 15 31 00 30 01 2A 01 28 00 48 00 68 77 38 FF C8
9C80: FF FF 00 60 03 61 C0 58 02 5C 03 68 01 29 00 80
9C90: 11 31 01 30 05 B1 00 91 9C 03 64 04 06 31 20 91
9CA0: 64 03 9C 04 06 31 40 91 03 5C 00 B4 80 91 00 85
9CB0: 00 5C 00 68 00 60 00 61 01 62 03 62 11 30 FF FF
9CC0: 4D 41 49 4E 32 2E 31 2D 48 41 52 56 45 59 00 00
9CD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9CE0: 21 C0 FF 22 1D 20 21 FF FF 22 1F 20 CD C4 16 CD
9CF0: 80 16 CD 50 0D CD 98 0E CD C0 13 CD 08 10 CD 30
9D00: 13 21 00 00 22 3C 20 CD F8 1D 3E 0D CD CF 00 FB
9D10: 76 3E 18 CD CF 00 97 32 21 20 32 22 20 32 2D 20
9D20: D3 00 D3 01 21 53 07 CD A3 1D 3E 50 32 32 20 C3
9D30: 28 03 3A 1D 20 F6 1F 32 1D 20 3E FC 32 1C 20 F1
9D40: F8 C9 E5 D5 C5 FB 3A 1C 20 C6 02 32 1C 20 B7 1F
9D50: CD 59 1D C1 D1 E1 F1 FB C9 0F DA 43 1F 6F 26 1F
9D60: 6E E9 21 1C 20 3E 04 CD 80 1D CD B5 03 21 84 24
9D70: 34 CA 24 00 3A 1D 20 CD 53 0D 3A 1D 20 C3 C3 13
9D80: 23 34 C0 3D C2 80 1D C9 2A 2E 20 23 22 2E 20 2A
9D90: 30 20 11 00 FF 19 D4 84 1D 22 30 20 CD D9 00 C8
9DA0: 2A 2E 20 22 25 20 2B 22 30 20 21 00 00 22 23 20
9DB0: 22 2E 20 C9 E8 2A 23 20 23 22 23 20 2A 25 20 19
9DC0: C9 FE 01 CA 15 1E FE 02 CA 0A 1E FE 03 CA F8 1D
9DD0: FE 04 C0 E5 CD E1 1D E1 C0 22 3A 20 E8 22 3C 20
9DE0: C9 2A 3C 20 7C B5 C9 CD E1 1D C8 28 22 3C 20 2A
9DF0: 3A 20 7E 23 22 3A 20 C9 7C B5 3E AA C2 03 1E 97
9E00: 21 4D 1F 22 36 20 32 38 20 C9 21 21 20 3A 35 20
9E10: 96 C8 3E 01 C9 1A 6F 13 1A 67 13 3C C8 3A 2D 20
9E20: FE AA CA 1D 1E D5 CD 61 1E D1 C3 15 1E 3E AA 32
9E30: 2D 20 3E 09 CD CF 00 F1 F8 C9 3A 2D 20 FE AA C0
9E40: CD CD 00 E6 10 C0 97 32 2D 20 CD D1 00 CD 93 00
9E50: 3E 08 CD CF 00 CD 61 1E 21 21 20 D2 5F 1E 23 34
9E60: C9 E8 7A E6 F8 0F 0F 21 40 00 DF 7E 23 66 6F 84
9E70: 37 C8 E5 E8 7D 87 C9 32 35 20 21 00 00 22 21 20
9E80: C9 FE 10 CA 95 1E 2A 33 20 29 29 29 29 E6 0F 85
9E90: 6F 22 33 20 C9 2A 33 20 97 32 33 20 32 34 20 C3
9EA0: 61 1E 21 32 20 35 CC 2A 1D CD C7 16 3A 1D 20 E6
9EB0: 1F C0 3A 1D 20 07 07 07 E6 07 FE 06 D0 F5 C6 22
9EC0: CD E6 00 29 29 29 29 5C F1 21 27 20 DF 73 C9 32
9ED0: CC 13 08 67 18 15 74 02 0C F0 17 04 36 13 04 11
9EE0: 10 05 CA 16 01 9E 0E 64 F3 1F 7D FE 07 CA 00 00
9EF0: 37 C9 00 00 00 00 00 00 00 00 00 00 00 00 00
9F00: 40 4E 5E 56 86 86 8F AA 8C 4E 78 56 86 86 8F AA
9F10: 89 80 78 86 86 86 8F AA 8C 4E 78 56 86 86 8F AA
9F20: 8C 4E 4D 4D 86 86 8F AA 8C 4E 4D 4D 86 86 8F AA

9F30: 89 B0 4D B9 B6 B6 BF AA BC 4E 4D 4D B6 B6 BF AA
9F40: C3 62 1D 3A 38 20 FE AA C0 2A 36 20 E9 C9 3E 01
9F50: CD C9 13 C3 D5 00 3E 01 CD 5F 0D C3 D5 00 97 CD
9F60: C9 13 E5 3A 1D 20 0F DA 71 1F CD BF 1F 6F C3 81
9F70: 1F CD E7 1D 6F C3 81 1F 97 CD C9 13 E5 97 CD 5F
9F80: 0D D1 63 C3 D5 00 C3 C6 13 C3 56 0D C3 9B 0E CD
9F90: 88 1D 3A 23 20 21 39 20 BE 77 C6 CD EB 11 3A 23
9FA0: 20 CD 64 1B 3A 23 20 C3 E7 17 CD 33 13 C3 3A 1E
9FB0: CD 4E 1F C3 59 0D CD 56 1F C3 5C 0D C3 A2 1E 21
9FC0: 1D 20 7E E6 1F C2 D3 1F 66 3A 1E 20 29 17 29 17
9FD0: 29 17 C9 7E E6 E0 0F 0F 0F 0F 2F B6 0F E6 7F 21
9FE0: CF 1E BE DA ED 1F 96 23 23 23 C3 E2 1F 23 5E 23
9FF0: 56 EB E9 21 1E 20 DF 7E C9 00 00 00 00 00 00

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE---BURST EXECUTIVE SECTION
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 * FILE BEXEC.A
0000          0005 *
0000          0006 * 8085 SPECIFIC INFORMATION
0000          0007 *
0000          0008 PSW   EQU   6
0000          0009 SP   EQU   6
0000          0010 *
0000          0011 * ROM ALLOCATIONS OF THE 4K AREA
0000          0012 *
0000          0013      ORG   40H
0040          0014 BID   DS   340H
0380          0015 BEXEC DS   0C0H
0440          0016 BLD   DS   100H
0540          0017 BSMP  DS   380H
08C0          0018 BFMT  DS   180H
0A40          0019 BCMP  DS   100H
0B40          0020      DS   0C00H-$
0C00          0021 FFP   DS   300H
0F00          0022 *
0F00          0023 * RAM ALLOCATIONS/EQUATES OF THE 2K AREA
0F00          0024 *
0F00          0025 RAM1  EQU   1000H  START ADDRESS
0F00          0026 RAMSIZE EQU   2048  SIZE IN BYTES
0F00          0027 MEM   EQU   8000H  BURST MEMORY BANK
0F00          0028 *
0F00          0029      ORG   RAM1
1000          0030 B1DRAM DS   20H
1020          0031 BEXRAM DS   10H
1030          0032 BFMTRAM DS   60H
1090          0033 BCMPRAM DS   30H
10C0          0034 BSMPRAM DS   140H
1200          0035 B1DRAM EQU   $
1200          0036 STACK EQU   RAM1+RAMSIZE
1200          0037 *
1200          0038 * DEFINE THESE GLOBALLY
1200          0039 *
1200          0040      COM   BID
1200          0041      COM   BEXEC
1200          0042      COM   BLD
1200          0043      COM   BSMP
1200          0044      COM   BFMT
1200          0045      COM   BCMP
1200          0046      COM   FFP
1200          0047      COM   B1DRAM
1200          0048      COM   BEXRAM
1200          0049      COM   B1DRAM
  
```

1200	0050	COM	BSMPRAM	
1200	0051	COM	BFMTRAM	
1200	0052	COM	BCMPRAM	
1200	0053	‡		
1200	0054	‡	RESET VECTOR	
1200	0055	‡		
1200	0056	ORG	0	
0000 C3 80 03	0057	JMP	BEXEC	
0003	0058	‡		
0003	0059	‡	BURST PROCESSOR EXECUTIVE CONTROL	
0003	0060	‡		
0003	0061	ORG	BEXEC	
0380 31 00 18	0062	LXI	SP,STACK	INIT STACK POINTER
0383 21 11 11	0063	LXI	H,1111H	
0386 CD FA 03	0064	CALL	MARK	
0389 CD 40 00	0065	CALL	BIOINIT	INITIALIZE THE DRIVERS
038C CD 67 00	0066	CALL	D5MS	
038F CD 40 04	0067	CALL	BLDINIT	INIT THE USER PROGRAM LDR
0392 CD 40 05	0068	CALL	BSMPINIT	INIT THE SAMPLING ROUTINES
0395 97	0069	SUB	A	
0396 32 20 10	0070	STA	CMDERR	
0399 21 00 00	0071	LXI	H,0	
039C CD FA 03	0072	CALL	MARK	
039F	0073	‡		
039F CD 4C 00	0074	BURST CALL	RECEIVE	RECEIVE A COMMAND
03A2 C2 A8 03	0075	JNZ	BEX	IF READY, EXECUTE
03A5 CD C0 03	0076	CALL	WAIT	POWER DOWN-WAIT
03A8 C3 9F 03	0077	JMP	BURST	
03AB	0078	‡		
03AB 22 21 10	0079	BEX SHLD	CMDCPY	
03AE CD FA 03	0080	CALL	MARK	
03B1 CD 43 05	0081	CALL	BSMPCMD	IF A SAMPLE CMD
03B4 D2 9F 03	0082	JNC	BURST	OK.
03B7 CD 43 04	0083	CALL	BLDCMD	IF A USER PROGRAM LOAD
03BA DC CB 03	0084	CC	ERROR	OK ELSE ERROR
03BD C3 9F 03	0085	JMP	BURST	
03C0	0086	‡		
03C0 2A C9 03	0087	WAIT LHLD	PWRDN	
03C3 22 23 10	0088	SHLD	PROG	
03C6 C3 23 10	0089	JMP	PROG	
03C9 76	0090	PWRDN	HLT	
03CA C9	0091	RET		
03CB	0092	‡		
03CB	0093	‡	DIAGNOSTIC OUTPUT ROUTINES	
03CB	0094	‡		
03CB 21 20 10	0095	ERROR LXI	H,CMDERR	UPDATE COMMAND ERROR
03CE 34	0096	INR	M	
03CF C9	0097	RET		
03D0	0098	‡		
03D0 11 00 10	0099	MENTEST LXI	D,RAM1	START AT RAM START

```

03D3 0E 08      0100      MVI   C,B
03D5 2E 00      0101      MVI   L,0
03D7 CD E6 03   0102 MTN   CALL  MTPAG
03DA 29         0103      DAD   H
03DB CA DF 03   0104      JZ    MTK
03DE 2C         0105      INR   L
03DF 14         0106 MTK    INR   D
03E0 7A         0107      MOV   A,D
03E1 0D         0108      DCR   C
03E2 C2 D7 03   0109      JNZ   MTN
03E5 C9         0110      RET
03E6           0111 *
03E6 1E 00      0112 MTPAG  MVI   E,0
03E8 CD F1 03   0113 MTP1   CALL  MTLOC
03EB C0         0114      RNZ
03EC 1C         0115      INR   E
03ED C2 EB 03   0116      JNZ   MTP1
03F0 C9         0117      RET
03F1           0118 *
03F1 1A         0119 MTLOC  LDAX  D      SAVE OLD VALUE
03F2 47         0120      MOV   B,A
03F3 2F         0121      CMA   .      FLIP ALL BITS
03F4 12         0122      STAX  D
03F5 1A         0123      LDAX  D
03F6 2F         0124      CMA
03F7 12         0125      STAX  D
03F8 BB         0126      CMP   B
03F9 C9         0127      RET
03FA           0128 *
03FA 7D         0129 MARK  MOV   A,L    OUTPUT TO DIAGNOSTIC LEDES
03FB D3 00      0130      OUT   0
03FD 7C         0131      MOV   A,H
03FE D3 01      0132      OUT   1
0400 C9         0133      RET
0401           0134 *
0401 42 55 52 53 0135      ASC   'BURST 2-1-85 PR HARVEY'
      54 20 32 2D
      31 2D 38 35
      20 50 52 20
      48 41 52 56
      45 59
0417 00      V 0136      DB   256    END OF BEXEC
0418           0137 *
0418           0138 * VARIABLES
0418           0139 *
0418           0140      ORG   BEXRAM
1020           0141 CMDERR DS 1      #COMMAND ERRORS FOUND
1021           0142 CMDCPY DS 2
1023           0143 PROG  DS 2      POWER DOWN PROGRAM
1025           0144 *

```

1025	0145 * EXTERNAL MODULE DEFINITIONS
1025	0146 *
1025	0147 ORG BIO
0040	0148 BIOINIT DS 3
0043	0149 GETMASK DS 3
0046	0150 SETMASK DS 3
0049	0151 RECSTAT DS 3
004C	0152 RECEIVE DS 3
004F	0153 SEND DS 3
0052	0154 ADPWR DS 3
0055	0155 SAMPLE DS 3
0058	0156 MEMPWR DS 3
005B	0157 MARSET DS 3
005E	0158 BANKSET DS 3
0061	0159 MODESET DS 3
0064	0160 SECOND DS 3
0067	0161 DSMS DS 3
006A	0162 READ DS 3
006D	0163 WRITE DS 3
0070	0164 *
0070	0165 ORG BLD
0440	0166 BLDINIT DS 3
0443	0167 BLDCMD DS 3
0446	0168 *
0446	0169 ORG BSMP
0540	0170 BSMPINIT DS 3
0543	0171 BSMPCMD DS 3

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE---BURST I/O DRIVER SECTION
0000          0003 * WRITTEN BY PETER P HARVEY
0000          0004 *
0000          0005 * FILE BIO.A - COMPATIBLE WITH BREADBOARD, PROTOTYPE AND
0000          0006 *             FLIGHT HARDWARE SYSTEMS.
0000          0007 *
0000          0008 * SYSTEM OPTIONS
0000          0009 *
0000          0010 COMLEN EQU   16      SERIAL COMMUNICATION BIT LENGTH
0000          0011 *
0000          0012 * SYSTEM DESCRIPTION
0000          0013 *
0000          0014 ROM1 EQU    0        4K ROM
0000          0015 RAM1 EQU   1000H    2K RAM
0000          0016 *
0000          0017 * A/D INTERFACING FOR HIGH SPEED DIGITIZATIONS.
0000          0018 * TO MUX ADDRESS XXX, READ ADC+(XXX*2)
0000          0019 *
0000          0020 ADC EQU    3000H    A/D DATA
0000          0021 ADCTL EQU   3001H    A/D CONTROL
0000          0022 MUXAD EQU   0FEH    MUX BITS
0000          0023 HIGH EQU   10H     HIGH GAIN OVERRIDE
0000          0024 LOW EQU    20H     LOW GAIN OVERRIDE
0000          0025 BV34AC EQU   06H    STUB QTY
0000          0026 GND EQU   LOW+BV34AC*2 SET DC QTY ON MUX
0000          0027 *
0000          0028 KGAINS EQU   23H    KELLEY AUTO GAIN INPUT BITS
0000          0029 KLYQTY EQU    13    KELLEY AUTO GAIN QTY NUMBER
0000          0030 *
0000          0031 FLIGHT EQU    1      FLIGHT BOARD == YES
0000          0032 *
0000          0033 * BURST MEMORY ADDRESSING
0000          0034 *
0000          0035 MEM EQU    3000H    BURST MEMORY BANK --WRITE ADDRESS
0000          0036 MEMRD EQU   MEM+7000H BURST MEMORY BANK -- READ ADDRESS
0000          0037 HIMAR EQU    70H    HIGH MEM ADDRESS BITS
0000          0038 AUTOWRITE EQU  80H    AUTOWRITE TO MEMORY
0000          0039 LOADADR EQU    40H    LOAD MAR
0000          0040 UPPER6 EQU    3FH    UPPER 6 MAR BITS
0000          0041 RELAYS EQU    20H    RELAY CONTROL BITS
0000          0042 ENDADR EQU    63H    END ADDR FOR WRAP-AROUND
0000          0043 RLYCTL EQU   ENDADR RELAY CONTROL TOO
0000          0044 SRRLY EQU    20H    SET/RESET RELAY
0000          0045 LOMAR EQU   0CFFFH  LOW 12 BITS OF MAR
0000          0046 *
0000          0047 * BREADBOARD I/O PORTS WHICH ARE DIFFERENT FROM ABOVE
0000          0048 *
0000          0049 AB155 EQU    20H    PROGRAMMABLE I/O CONTROL PORTS
  
```

0000	0050	BB155	EQU	60H	
0000	0051	BBHIMAR	EQU	BB155+1	BREADBOARD HIGH MAR REGISTER
0000	0052	BBRELAYS	EQU	BB155+2	BREADBOARD RELAY CONTROLS
0000	0053	‡			
0000	0054	PSW	EQU	6	B0B5 INFORMATION
0000	0055	SP	EQU	6	
0000	0056	SHLX	EQU	10H	STORE HL AT [DE]
0000	0057	SLDE	EQU	18H	SHIFT LEFT [DE]
0000	0058	RIM	EQU	20H	READ INT MASK
0000	0059	SIM	EQU	30H	SET INT MASK
0000	0060	SDO	EQU	80H	SERIAL DATA OUTPUT
0000	0061	SDOEN	EQU	40H	SERIAL ENABLE
0000	0062	RES75	EQU	10H	RESET 7.5 FLOP
0000	0063	MSE	EQU	8	INTERRUPT MASK SET ENABLE
0000	0064	DIS75	EQU	4	DISABLES FOR THE INTERRUPTS
0000	0065	DIS65	EQU	2	
0000	0066	DIS55	EQU	1	
0000	0067	‡			
0000	0068	ADUT	EQU	1	B155 [SA3001] PROGRAMMING
0000	0069	POUT	EQU	2	
0000	0070	COUT	EQU	12	
0000	0071	NOTM	EQU	040H	TIMER OFF
0000	0072	TM	EQU	0C0H	TIMER ON
0000	0073	‡			
0000	0074	‡	INTERRUPTS		
0000	0075	‡			
0000	0076		ORG	618+4	2KHZ CLOCK
0034 E5	0077	INT65	PUSH	H	
0035 2A 04 10	0078	LHLD	TIMIVECT		
0038 E9	0079	PCHL	.		
0039	0080	‡			
0039	0081		ORG	718+4	INT 7.5
003C C3 C5 00	0082	JMP	RST75		
003F	0083	‡			
003F	0084	‡	BIO ENTRY POINTS		
003F	0085	‡			
003F	0086		ORG	BIO	
0040 C3 7C 00	0087	JMP	BIOINIT		
0043 C3 AF 00	0088	JMP	GETMASK		
0046 C3 AA 00	0089	JMP	SETMASK		
0049 C3 FE 00	0090	JMP	RECSTAT		
004C C3 04 01	0091	JMP	RECEIVE		
004F C3 0E 01	0092	JMP	SEND		
0052	0093	‡			
0052 C3 68 01	0094	JMP	ADPWR		
0055 C3 3A 01	0095	JMP	SAMPLE		
0058	0096	‡			
005B C3 2A 02	0097	JMP	MEMPWR		
005B C3 7C 01	0098	JMP	MARSET		
005E C3 AF 01	0099	JMP	BANKSET		

0061 C3 CF 01	0100	JMP	MODESET
0064	0101	↓	
0064 C3 67 02	0102	JMP	SECOND
0067 C3 71 02	0103	JMP	B5MS
006A	0104	↓	
006A C3 DC 01	0105	JMP	READ
006D C3 E2 01	0106	JMP	WRITE
0070 C3 91 02	0107	JMP	REWIND
0073 C3 17 02	0108	JMP	MARGET
0076 C3 B1 00	0109	JMP	SETVECT
0079 C3 C1 00	0110	JMP	SETIO
007C	0111	↓	
007C	0112	↓	INITIALIZE THE BURST HARDWARE
007C	0113	↓	
007C 3E 4C	0114	BIOSINIT MVI	A,NOTM+COUT CONFIGURE I/O IN
007E D3 20	0115	OUT	B8155 CASE THIS IS THE BREADBOARD
0080 3E 4F	0116	MVI	A,NOTM+ACOUT+ROUT+COUT
0082 D3 60	0117	OUT	B8155
0084	0118	↓	
0084 CD 43 02	0119	CALL	RLYINIT INIT RELAY CONTROL
0087 3E 50	0120	MVI	A,SDGEN+RES75 SDO=0
0089 CD AA 00	0121	CALL	SETMASK AND 7.5 INITIALIZED
008C 97	0122	SUB	A COMMAND STATUS=NOT READY
008D 32 03 10	0123	STA	CMDSTAT
0090 32 08 10	0124	STA	IOMODE CMD-PENDING MODE OFF
0093 21 00 00	0125	LXI	H,0
0096 CD B1 00	0126	CALL	SETVECT TURN OFF INTERRUPTS
0099	0127	↓	
0099 11 34 01	0128	LXI	D,LPW1 PUT LOW POWER WAIT
009C 21 0E 10	0129	LXI	H,LPWAIT INTO MEMORY
009F 0E 06	0130	MVI	C,LPWLEN
00A1	0131	↓	
00A1 1A	0132	COPY LDAX	D
00A2 77	0133	MOV	M,A
00A3 13	0134	INX	D
00A4 23	0135	INX	H
00A5 0D	0136	DCR	C
00A6 C2 A1 00	0137	JNZ	COPY
00A9 C9	0138	RET	
00AA	0139	↓	
00AA	0140	↓	SET/READ PROCESSOR MASK
00AA	0141	↓	
00AA 32 00 10	0142	SETMASK STA	SDCOPY
00AD 30	0143	DB	SIM
00AE C9	0144	RET	.
00AF	0145	↓	
00AF 20	0146	GETMASK DB	RIM
00B0 C9	0147	RET	.
00B1	0148	↓	
00B1 F3	0149	SETVECT DI	. DISABLE INTERRUPTS

00B2 22 04 10	0150	SHLD	TIMIVECT	SET THE VECTOR
00B5 7C	0151	MOV	A,H	IF HL=0, DISABLE INTS
00B6 B5	0152	ORA	L	
00B7 3E 0B	0153	MVI	A,MSE+DIS55+DIS65	
00B9 CA BE 00	0154	JZ	SV1	
00BC 3E 09	0155	MVI	A,MSE+DIS55	
00BE 30	0156 SV1	DB	SIM	
00BF FB	0157	E1	.	REENABLE INTERRUPTS
00C0 C9	0158	RET	.	
00C1	0159 #			
00C1 32 0B 10	0160 SET10	STA	IOMODE	SET COMMAND-PENDING MODE ON/OFF
00C4 C9	0161	RET	.	
00C5	0162 #			
00C5	0163 #			SERIAL REQUEST FROM THE MAIN PROCESSOR
00C5	0164 #			
00C5 F5	0165 RST75	PUSH	PSW	SAVE REGISTERS USED
00C6 C5	0166	PUSH	B	
00C7 D5	0167	PUSH	D	
00CB 0E 10	0168	MVI	C,COMLEN	NUMBER OF BITS TO SHIFT
00CA 3E C0	0169	MVI	A,SDO+SDOEN	SDO<-1 (READY)
00CC 30	0170	DB	SIM	
00CD 20	0171 WTTM	DB	RIM	WAIT TILL S1D=0 (START)
00CE 07	0172	RLC		
00CF DA CD 00	0173	JC	WTTM	
00D2 00	0174	NOP	.	GIVE TM PROC SOME TIME
00D3	0175 #			
00D3 3E 00	0176 SH1	MVI	A,0	NOP OF 7 CYCLES
00D5 20	0177	DB	RIM	GET A BIT
00D6 07	0178	RLC		
00D7 18	0179	DB	SLDE	SHIFT INTO DE
00D8 0D	0180	DCR	C	#BITS--
00D9 C2 D3 00	0181	JNZ	SH1	
00DC	0182 #			
00DC EB	0183	XCHG	.	CMDREG=[DE]
00DD 22 01 10	0184	SHLD	CMDREG	
00E0 EB	0185	XCHG		
00E1 3E 01	0186	MVI	A,1	CMDSTAT=READY
00E3 32 03 10	0187	STA	CMDSTAT	
00E6	0188 #			
00E6 3A 00 10	0189	LDA	SDCOPY	RESET SDO TO STATE PRIOR
00E9 E6 80	0190	ANI	SDO	TO INTERRUPT AND RESET THE
00EB F6 50	0191	ORI	SDOEN+RES75	7.5 INTERRUPT FLOP TOO.
00ED 30	0192	DB	SIM	
00EE D1	0193	POP	D	RESTORE REGISTERS
00EF C1	0194	POP	B	
00F0	0195 #			
00F0 3A 06 10	0196	LDA	IOMODE	IN CMD PENDING MODE, SET CARRY
00F3 0F	0197	RRC	.	WHEN RETURNING
00F4 DA FA 00	0198	JC	CPMODE	
00F7 F1	0199	POP	PSW	NORMAL MODE

00FB FB	0200	EI
00F9 C9	0201	RET
00FA	0202	↓
00FA F1	0203	CPMODE POP PSM
00FB 37	0204	STC
00FC FB	0205	EI .
00FD C9	0206	RET .
00FE	0207	↓
00FE	0208	↓ RETURN STATUS OF COMMUNICATIONS WITH TM PROCESSOR
00FE	0209	↓
00FE 21 03 10	0210	RECSTAT LXI H,CMDSTAT RETURN ZERO IF NOT READY
0101 7E	0211	MOV A,M AS WELL AS THE ADDRESS OF STAT
0102 B7	0212	ORA A
0103 C9	0213	RET
0104	0214	↓
0104	0215	↓ RECEIVE DATA FROM THE TELEMETRY PROCESSOR
0104	0216	↓ ON EXIT: [HL]=DATA IF NOT ZERO, ELSE NOT READY
0104	0217	↓
0104 CD FE 00	0218	RECEIVE CALL RECSTAT CHECK FOR STAT=READY
0107 C8	0219	RZ .
0108 36 00	0220	MVI M,0 STAT=NOT READY
010A 24 01 10	0221	LHLD CMDREG PICK UP THE COMMAND REGISTER
010D C9	0222	RET .
010E	0223	↓
010E	0224	↓ SEND [HL] TO THE TELEMETRY PROCESSOR
010E	0225	↓
010E E5	0226	SEND PUSH H SAVE MESSAGE OUT
010F C5	0227	PUSH B
0110 0E 10	0228	MVI C,COMLEN C=#BITS TO SHIFT
0112 3E C0	0229	MVI A,SDO+SDOEN SDO=1 (REQUEST)
0114 32 00 10	0230	STA SDCOPY
0117 30	0231	DB SIM
0118 F3	0232	DI . IF NO INT, WE'VE GOT THE LINE.
0119	0233	↓
0119 CD 0E 10	0234	CALL LPWAIT
011C 3E 40	0235	MVI A,SDOEN SDO=0 (START)
011E 30	0236	DB SIM
011F	0237	↓
011F 3E 80	0238	SENDERBIT MVI A,SDOEN*2
0121 29	0239	DAD H
0122 1F	0240	RAR
0123 30	0241	DB SIM
0124 0D	0242	DCR C
0125 C2 1F 01	0243	JNZ SENDERBIT
0128	0244	↓
0128 C1	0245	POP B
0129 E1	0246	POP H RESTORE HL AND DELAY
012A 3E 40	0247	MVI A,SDOEN SDO=0 (NO MORE REQUEST)
012C 32 00 10	0248	STA SDCOPY
012F F6 10	0249	ORI RES75 AND RESET 7.5 INTERRUPT

0131 30	0250	DB	SIM	
0132 FB	0251	EI	.	
0133 C9	0252	RET	.	
0134	0253	*		
0134	0254	*	LOW POWER WAIT ROUTINE	
0134	0255	*	(RUNS IN RAM AT LPWAIT)	
0134	0256	*		
0134 20	0257	LPWI	DB	RIM
0135 07	0258	RLC		
0136 D2 0E 10	0259	JNC	LPWAIT	
0139 C9	0260	RET		
013A	0261	LPWLEN	EQU	*-LPWI
013A	0262	*		
013A	0263	*	SAMPLE A QTY IN THE NORMAL FASHION	
013A	0264	*	ON ENTRY: [A]=QTY TO SAMPLE	
013A	0265	*	ON EXIT : [HL]=13 BIT SAMPLE (GAIN INCLUDED)	
013A	0266	*		
013A F5	0267	SAMPLE	PUSH	PSW SAVE THE SAMPLE *
013B B7	0268	ADD	A	MOVE INTO THE LOW ADDRESS BITS
013C E6 FE	0269	ANI	MUXAD	
013E 6F	0270	MOV	L,A	
013F 26 30	0271	MVI	H,ADC/256	
0141	0272	*		
0141 3E 03	0273	MVI	A,3	CONFIGURE A/D FOR AUTO START MODE
0143 32 01 30	0274	STA	ADCTL	
0146	0275	*		
0146 7E	0276	MOV	A,M	READ THE A/D TO SET THE
0147 CD 61 01	0277	CALL	SMPDLA	GAIN DECISION AND SAMPLE/HOLD
014A	0278	*		
014A 7E	0279	MOV	A,M	READ THE A/D TO CONVERT THE
014B CD 61 01	0280	CALL	SMPDLA	VALUE HELD BY THE SAMPLE/HOLD
014E	0281	*		
014E 5E	0282	MOV	E,M	READ THE VALUE FROM THE A/D REGISTER
014F 2C	0283	INR	L	
0150 66	0284	MOV	H,M	
0151 6B	0285	MOV	L,E	
0152	0286	*		
0152 FI	0287	POP	PSW	RESTORE THE ORIGINAL QTY NUMBER
0153 FE 0D	0288	CPI	KLYQTY	
0155 C0	0289	RNZ	.	
0156 29	0290	DAD	H	IF KELLEY, USE ONLY MS 8 BITS
0157 29	0291	DAD	H	
0158 29	0292	DAD	H	
0159 29	0293	DAD	H	
015A 6C	0294	MOV	L,H	
015B D0 23	0295	IN	KGAINS	THEN APPLY FOUR BITS
015D E6 0F	0296	ANI	0FH	FROM THE KELLEY PORT
015F 67	0297	MOV	H,A	
0160 C9	0298	RET		
0161	0299	*		

0161 3E 02	0300 SMPDLA MVI A,2 DELAY A WHILE
0163 3D	0301 SDLA DCR A
0164 C2 63 01	0302 JNZ SDLA
0167 C9	0303 RET
0168	0304 *
0168	0305 * A/D POWER CONTROL
0168	0306 * ON ENTRY: CARRY=1 TO TURN ON, 0 TO TURN OFF.
0168	0307 *
0168 DC 3A 02	0308 ADPWR CC SRLY 1F ON, SET S/R RELAY
016B 3E 40	0309 MVI A,40H PULSE THE 5V RELAY COIL
016D CD 58 02	0310 CALL PULSE
0170 3E 80	0311 MVI A,80H AND THEN THE 12V RELAY
0172 CD 58 02	0312 CALL PULSE
0175 CD 48 02	0313 CALL RRLY
0178 3A 4C 30	0314 LDA ADC+6ND ADDRESS GROUND INPUT
017B C9	0315 RET
017C	0316 *
017C	0317 * SET THE MEMORY ADDRESS REGISTER
017C	0318 * ON ENTRY: [AHL]=16 BIT ADDRESS TO SET
017C	0319 *
017C C5	0320 MARSET PUSH B
017D E6 03	0321 ANI 3 MASK TO 2 BITS
017F 47	0322 MOV B,A
0180	0323 *
0180 7C	0324 MOV A,H PICK UP 4 BITS FROM HL
0181 E6 F0	0325 ANI 0F0H
0183 B0	0326 ORA B PUT EM TOGETHER
0184 07	0327 RLC . AND SHIFT TO 6 LSB'S
0185 07	0328 RLC
0186 07	0329 RLC
0187 07	0330 RLC
0188 F6 40	0331 ORI LOADADR SET LOAD FLAG
018A CD A3 01	0332 CALL OUTHM AND HIGH BITS
018D 2D	0333 DCR L SET LOW 12 BUT CORRECT
018E 22 FF CF	0334 SHLD LOMAR FOR THE INCREMENT
0191 2C	0335 INR L TO THE LOW BYTE.
0192 3A 09 10	0336 LDA STBANK RESTORE THE START BANK
0195 CD C0 01	0337 CALL SETSB TO THE HIMAR LATCH.
0198	0338 *
0198 29	0339 DAD H NOW RECORD THE MAR SETTING
0199 78	0340 MOV A,B IN A SHIFTED CONFIGURATION
019A 17	0341 RAL . (*2)
019B 32 0D 10	0342 STA MARREG+2
019E 22 08 10	0343 SHLD MARREG
01A1 C1	0344 POP B
01A2 C9	0345 RET
01A3	0346 *
01A3 D5	0347 OUTHM PUSH D
01A4 32 07 10	0348 STA MEMMODE RECORD ALL HIMAR OUTPUTS
01A7 11 61 70	0349 LXI D,HIMAR#256+BBHIMAR [DE]=BB+FLIGHT PORT #'S

01AA CD 86 02	0350	CALL	00UT	
01AD 01	0351	POP	D	
01AE C9	0352	RET		
01AF	0353	*		
01AF	0354	*	SET BANK START AND END ADDRESSING	
01AF	0355	*	ON ENTRY: [B]=START BANK # (0..5)	
01AF	0356	*	[C]=END BANK # (0..5)	
01AF	0357	*		
01AF 79	0358	BANKSET MOV	A,C	OUTPUT THE END BANK
01B0 E6 07	0359	ANI	7	MASKED TO PROPER BITS
01B2 32 0A 10	0360	STA	ENBANK	
01B5 EE 07	0361	XRI	7	AND COMPLEMENTED
01B7 D3 63	0362	OUT	ENDAD	
01B9 32 06 10	0363	STA	ENDCOPY	
01BC	0364	*		
01BC 78	0365	MOV	A,8	MASK THE START ADDRESS INFO
01BD 32 09 10	0366	STA	STBANK	
01C0 87	0367	SETSB	ADD A	PUT IN THE 3 ZERO BITS
01C1 87	0368	ADD	A	
01C2 87	0369	ADD	A	
01C3 E6 3F	0370	SETHM	ANI	UPPER6
01C5 4F	0371	MOV	C,A	
01C6 3A 07 10	0372	LDA	MEMMODE	DON'T EFFECT THE MEMMODE
01C9 E6 80	0373	ANI	AUTOWRITE	
01CB B1	0374	ORA	C	
01CC C3 A3 01	0375	JMP	OUTHM	
01CF	0376	*		
01CF	0377	*	SET/RESET THE MEMORY AUTOWRITE MODE	
01CF	0378	*	ON ENTRY: [A]=1 FOR AUTOWRITE ELSE NO AUTOWRITE	
01CF	0379	*		
01CF E6 01	0380	MODESET ANI	1	USE ONLY THE LSB
0101 0F	0381	RRC	.	
01D2 47	0382	MOV	B,A	SAVE THE BIT
0103 3A 07 10	0383	LOA	MEMMODE	PICK UP THE CURRENT START
01D6 E6 3F	0384	ANI	UPPER6	BANK
01D8 B0	0385	ORA	8	
01D9 C3 A3 01	0386	JMP	OUTHM	
01DC	0387	*		
01DC	0388	*	MEMORY FUNCTIONS	
01DC	0389	*		
01DC 2A 00 F0	0390	READ LHLO	MEMRO	
01DF C3 E5 01	0391	JMP	INCMAR	
01E2	0392	*		
01E2 22 00 80	0393	WRITE SHLD	MEM	
01E5 E5	0394	INCMAR	PUSH H	
01E6 2A 0B 10	0395	LHLO	MARREG	INCREMENT THE MAR COPY
01E9 11 04 00	0396	LXI	D,4	BY DOUBLE THE AMOUNT TO MAKE
01EC 19	0397	DAD	0	THE BANK # IN 1 BYTE.
01ED 22 0B 10	0398	SHLO	MARREG	
01F0 D2 FF 01	0399	JNC	INDEX	

01F3 21 0D 10	0400	LX1	H,MARREG+2	IF MARBANK++ ==ENDBANK+1	VALUE
01F6 34	0401	INR	M		
01F7 3A 0A 10	0402	LDA	ENBANK	THEN	REWIND
01FA 3C	0403	INR	A		
01FB BE	0404	CMP	M		
01FC CC 01 02	0405	CZ	REWIND		
01FF E1	0406	INCEX	FOP	H	
0200 C9	0407	RET			
0201	0408	‡			
0201 3A 09 10	0409	REWIND	LDA	STRANK	SET MAR TO BEGINNING OF BANK
0204 CD 0A 02	0410	CALL	ADRBANK		
0207 C3 7C 01	0411	JMP	MARSET		
020A	0412	‡			
020A 0F	0413	ADRBANK	RRC	.	CONVERT BANK INTO ADDRESS
020B C5	0414	PUSH	B		
020C 47	0415	MOV	B,A		
020D E6 E0	0416	ANI	B0H		
020F 67	0417	MOV	H,A		
0210 2E 00	0418	MVI	L,0		
0212 78	0419	MOV	A,B		
0213 E6 03	0420	ANI	3		
0215 C1	0421	POP	B		
0216 C9	0422	RET			
0217	0423	‡			
0217	0424	‡	READ-BACK	MEMORY ADDRESS REGISTER	
0217	0425	‡			
0217 3A 0D 10	0426	MARGET	LDA	MARREG+2	GET THE COPY
021A C5	0427	PUSH	B		
021B B7	0428	ORA	A		
021C 1F	0429	RAR	.	FROM THE RAM AND	
021D 47	0430	MOV	B,A	DIVIDE BY 2	
021E 2A 0B 10	0431	LHLD	MARREG		
0221 7C	0432	MOV	A,H		
0222 1F	0433	RAR			
0223 67	0434	MOV	H,A		
0224 7D	0435	MOV	A,L		
0225 1F	0436	RAR			
0226 6F	0437	MOV	L,A		
0227 7B	0438	MOV	A,B		
0228 C1	0439	POP	B		
0229 C9	0440	RET			
022A	0441	‡			
022A	0442	‡	MEMORY POWER CONTROL SECTION		
022A	0443	‡	ON ENTRY: [A]= BANK NUMBER TO TURN ON OR OFF		
022A	0444	‡	CARRY=1 FOR ON, 0 FOR OFF		
022A	0445	‡			
022A DC 3A 02	0446	MEMPWR	CC	SRLY	IF CRY, PREPARE SET/RESET RELAY
022D FE 06	0447	CFI	6	IF ERRONEOUS BANK, SIMPLY RETURN	
022F D0	0448	RNC	.		
0230 CD 7B 02	0449	CALL	UNARY	CONVERT TO UNARY	

0233 70	0450	MOV	A,L	SO THAT WE PULSE ONLY 1
0234 C0 58 02	0451	CALL	PULSE	DO IT.
0237 C3 48 02	0452	JMP	RRLY	AND GO TO LOW POWER AFTERWARD
023A	0453	*		
023A	0454	*	RELAY CONTROL SECTION	
023A	0455	*		
023A F5	0456	SRLY	PUSH	PSW
023B 3A 06 10	0457	LDA	ENDCOPY	SET-RESET TO SET POSITION
023E F6 20	0458	ORI	SRRLY	
0240 C3 4E 02	0459	JMP	SRSET	
0243	0460	*		
0243 3E FF	0461	RLYINIT	MVI	A,OFFH RESET ALL RELAY COILS
0245 CD 61 02	0462	CALL	OUTRELAYS	
0248	0463	*		
0248 F5	0464	RRLY	PUSH	PSW
0249 3A 06 10	0465	LDA	ENDCOPY	RESET POSITION
024C E6 DF	0466	ANI	-1-SRRLY	
024E D3 63	0467	SRSET	OUT	RLYCTL
0250 32 06 10	0468	STA	ENDCOPY	
0253 CD 71 02	0469	CALL	D5MS	DELAY FOR IT TO SETTLE OUT
0256 F1	0470	POP	PSW	
0257 C9	0471	RET		
0258	0472	*		
0258 2F	0473	PULSE	CMA	. OUTPUT IN COMPLEMENT FORM
0259 CD 61 02	0474	CALL	OUTRELAYS	
025C CD 71 02	0475	CALL	D5MS	AND DELAY FOR THE FLIP
025F	0476	*		
025F 3E FF	0477	MVI	A,OFFH	RESET ALL COILS
0261 11 62 20	0478	OUTRELAYS	LXI	D,RELAYS*256+BBRELAYS
0264 C3 86 02	0479	JMP	DOUT	
0267	0480	*		
0267 06 C8	0481	SECOND	MVI	B,200 DELAY 1 SECOND
0269 CD 71 02	0482	SEC1	CALL	D5MS
026C 05	0483	DCR	B	
026D C2 69 02	0484	JNZ	SEC1	
0270 C9	0485	RET		
0271	0486	*		
0271 11 71 02	0487	D5MS	LXI	D,5*3*1000/24 DELAY 5 MILLISECS AT 3MHZ
0274 1B	0488	DELAY	DCX	D
0275 78	0489	MOV	A,E	
0276 B2	0490	ORA	D	
0277 C2 74 02	0491	JNZ	DELAY	
027A C9	0492	RET		
027B	0493	*		
027B	0494	*	CONVERT NUMBER IN A INTO UNARY IN HL	
027B	0495	*		
027B 21 01 00	0496	UNARY	LXI	H,1
027E E6 0F	0497	ANI	15	
0280 C8	0498	UNA1	RZ	.
0281 29	0499	DAD	H	


```

0282 3D          0500      DCR   A
0283 C3 80 02   0501      JMP   UNAI
0286           0502      #
0286           0503      # D OUTPUT. THIS ROUTINE MAINTAINS COMPATIBILITY BETWEEN
0286           0504      # THE BREADBOARD AND FLIGHT UNITS BY OUTPUTTING
0286           0505      # THE ACCUM TO PORT(D) IF FLIGHT AND PORT(E) IF
0286           0506      # THE BREADBOARD.
0286           0507      #
0286 F5         0508 DOUT  PUSH  PSW   SAVE THE ACCUM
0297 3E 01      0509      MVI   A,FLIGHT THE FLIGHT UNIT=1
0289 B7         0510      ORA   A
028A C2 8E 02   0511      JNZ   DOFLGHT
028D 53         0512      MOV   D,E
028E F1         0513 DOFLGHT POP  PSW   STORE THE DATA AT PORT(D)
028F 12         0514      STAX  D
0290 C9         0515      RET
0291 00         V 0516      DB   256  END-OF-BIO MODULE
0292           0517      #
0292           0518      # VARIABLES FOR THE BURST DRIVERS
0292           0519      #
0292           0520      ORG   BIORAM
1000           0521 SDCOPY DS   1    SERIAL DATA COPY
1001           0522 CMDREG DS   2    COMMAND REGISTER
1003           0523 CMDSTAT DS  1    COMMAND STATUS (1=READY)
1004           0524 TIMIVECT DS  2    TIMER 1 VECTOR
1006           0525 ENDCOPY DS  1    COPY OF END ADDRESS
1007           0526 MEMMODE DS  1    MEMORY MODE REGISTER
1008           0527 IOMODE DS  1    CMO PENDING MODE
1009           0528 STBANK DS  1    START BANK OF MEMORY
100A           0529 ENBANK DS  1    END BANK
100B           0530 MARREG DS  3    RAM COPY OF MEMORY ADDRESS REG
100E           0531 LPWAIT DS  LPWLEN LOW POWER WAIT ROUTINE
  
```

```

0000          0001 #
0000          0002 # CRRES FLIGHT SOFTWARE---BURST SAMPLING ROUTINES
0000          0003 # FILE : BSMP.A
0000          0004 #
0000          0005 PSW EQU 6 PROCESSOR EQUATES
0000          0006 #
0000          0007 # ENTRY POINTS
0000          0008 #
0000          0009 ORG BSMP
0540 C3 46 05 0010 JMP BSMPINIT
0543 C3 63 05 0011 JMP BSMPCMO
0546          0012 #
0546          0013 # INITIALIZE THE SAMPLING ROUTINES.
0546          0014 #
0546 C0 C0 0B 0015 BSMPINIT CALL INIFMT RESET ALL FORMATS
0549 97          0016 SUB A SET TO FORMAT(0)
054A C0 C3 0B 0017 CALL SETFMT
054D 3E 0F 0018 MVI A,15 SET TO HIGHEST RATE
054F C0 94 05 0019 CALL SETRATE
0552 B7 0020 ORA A TURN OFF THE A/D POWER
0553 C0 52 00 0021 CALL AOPWR
0556 97 0022 SUB A ZERO THE MODE
0557 32 D5 10 0023 STA BMOOE
055A 21 00 00 0024 LXI H,0 ZERO THE MEMORY ADDRESS
055D C0 5B 00 0025 CALL MARSET
0560 21 45 B3 0026 LXI H,0B345H TURN ON BANKS 4 AND 5
0563          0027 #
0563 7C 0028 BSMPCMO MOV A,H 1F COMMAND BEGINS WITH
0564 E6 F0 0029 ANI OFOH OBXXXH, THEN WE'RE INTERESTED
0566 FE B0 0030 CPI OBOH
0568 37 0031 STC .
0569 C0 0032 RNZ .
056A          0033 #
056A 7C 0034 MOV A,H CALCULATE WHICH OF THE SMP COMMANDS
056B D6 B0 0035 SUI OBOH THIS ONE IS.
056D FE 0B 0036 CPI NCMDS 1F GREATER THAN KNOWN CMO
056F 3F 0037 CMC . RETURN(CARRY)
0570 D8 0038 RC .
0571          0039 #
0571 EB 0040 XCHG . SAVE THE ORIGINAL COMMAND VALUE
0572 B7 0041 ADD A
0573 21 7E 05 0042 LXI H,CMDTB AND REFERENCE THE TABLE
0576 C0 51 0B 0043 CALL REF FOR THE ROUTINES ADDRESS
0579 C0 56 0B 0044 CALL LOOHL
057C 7B 0045 MOV A,E A=LOW VALUE OF COMMAND
057D E9 0046 PCHL
057E          0047 #
057E 94 05 0048 CMDTB DW SETRATE BFREQ
0580 C3 0B 0049 DW SETFMT BFMT

```

0582 C6 08	0050	DW	ADDFMT	BQTY
0584 9A 05	0051	DW	BSELECT	BANKS
0586 E0 05	0052	DW	GO	BGO
0588 2E 06	0053	DW	STOP	BSTOP
058A 26 06	0054	DW	PAUSE	BPAUSE
058C 19 06	0055	DW	CONT	RCONTINUE
058E 46 06	0056	DW	PLAY	BPLAY
0590 00 00	0057	DW	0	BRESET
0592 D9 05	0058	DW	MODSET	BMODE
0594	0059	NCMDS	EQU	‡-CMDTR/2 NUMBER OF COMMANDS
0594	0060	‡		
0594	0061	‡	SET THE RATE OF THE BURST	
0594	0062	‡		
0594 E6 0F	0063	SETRATE	ANI	15
0596 32 C4 10	0064	STA	FREQ	SET NEW DESIRED FREQ
0599 C9	0065	RET		
059A	0066	‡		
059A	0067	‡	SELECT THE MEMORY BANKS TO USE	
059A	0068	‡		
059A 5F	0069	BSELECT	MOV	E,A
059B 0F	0070	RRC	.	B=START BANK
059C 0F	0071	RRC		
059D 0F	0072	RRC		
059E 0F	0073	RRC		
059F E6 07	0074	ANI	7	
05A1 47	0075	MOV	B,A	
05A2 32 C6 10	0076	STA	STBANK	
05A5	0077	‡		
05A5 7B	0078	MOV	A,E	C=END BANK
05A6 E6 07	0079	ANI	7	
05A8 4F	0080	MOV	C,A	
05A9 32 C7 10	0081	STA	ENBANK	
05AC CD 5E 00	0082	CALL	BANKSET	
05AF	0083	‡		
05AF	0084	‡	MEMORY POWER CONTROLLER. TURNS ON STBANK TO ENBANK	
05AF	0085	‡	AND TURNS OFF ANY OTHERS.	
05AF	0086	‡		
05AF 21 05 00	0087	MEMPC	LXI	H,005H TURN OFF BANKS 0 TO 5
05B2 06 00	0088	MVI	B,0	
05B4 CD C8 05	0089	CALL	ONOFF	
05B7	0090	‡		
05B7 3A C6 10	0091	LDA	STBANK	TURN ON FROM START TO END
05BA E6 07	0092	ANI	7	
05BC 67	0093	MOV	H,A	
05BD 3A C7 10	0094	LDA	ENBANK	
05C0 6F	0095	MOV	L,A	
05C1 06 01	0096	MVI	B,1	
05C3 CD C8 05	0097	CALL	ONOFF	
05C6 B7	0098	CRA	A	RETURN(NO CARRY)
05C7 C9	0099	RET		

05CB	0100 †		
05CB	0101 DNDFF	EQU	‡
05CB 7D	0102	MDV	A,L IF START > END, RETURN
05C9 8C	0103	CMP	H
05CA D8	0104	RC	.
05CB 78	0105	MDV	A,B SET FDR DN DR DFF
05CC 0F	0106	RRC	
05CD 7C	0107	MDV	A,H SELECT BANK
05CE E5	0108	PUSH	H
05CF C5	0109	PUSH	B
05D0 CD 58 00	0110	CALL	MEMPWR
05D3 C1	0111	PDP	B
05D4 E1	0112	POP	H
05D5 24	0113	INR	H
05D6 C3 C8 05	0114	JMP	ONOFF
05D9	0115 †		
05D9 32 D5 10	0116 MDDSET	STA	BMDDE
05DC 0F	0117	RRC	.
05DD C3 52 00	0118	JMP	ADPWR
05E0	0119 †		
05E0	0120 †	BURST START/STOP/PAUSE/CONT CONTRDLS	
05E0	0121 †		
05E0	0122 BDGUS	EQU	1800H BOGUS SAMPLE (SATURATED HIGH GAIN)
05E0	0123 SMARK	EQU	BOGUS+0A00AH START MARKER
05E0	0124 PMARK	EQU	BOGUS+0C00CH PAUSE MARKER
05E0	0125 EMARK	EQU	BOGUS+0E00EH STDP MARKER
05E0	0126 †		
05E0 3A D5 10	0127 BD	LDA	BMDDE TURN DN THE A/D IN
05E3 0F	0128	RRC	.
05E4 3F	0129	CMC	
05E5 DC 52 00	0130	CC	ADPWR CASE IT IS DFF
05E8 CD E4 06	0131	CALL	CMPROG COMPILE THE SAMPLE PRDGRAM
05E8 3A C5 10	0132	LDA	REALF SEND THE REAL FREQ BACK
05EE CD CF 06	0133	CALL	SENDA
05F1 CD 65 07	0134	CALL	GETDUR CALCULATE DURATION DF BURST
05F4 22 C0 10	0135	SHLD	DURATION
05F7 D5	0136	PUSH	D
05F8 CD 4F 00	0137	CALL	SEND
05F8 E1	0138	POP	H
05FC 22 C2 10	0139	SHLD	DURATION+2
05FF CD 4F 00	0140	CALL	SEND
0602	0141 †		
0602 CD 70 00	0142 WRTST	CALL	REWIND REPOSITION THE DIGITAL TAPE
0605 21 0A 88	0143	LXI	H,SMARK WRITE('START-MARK') INTD MEMDRY
0608 CD 6D 00	0144	CALL	WRITE IN CASE OF SHORT BURST
060B	0145 †		
060B CD 70 00	0146	CALL	REWIND VERIFY THE MARK
060E CD 6A 00	0147	CALL	READ
0611 11 0A 88	0148	LXI	D,SMARK
0614 CD 58 08	0149	CALL	EQ16

0617 37	0150	STC	
0618 C0	0151	RNZ	.
0619	0152	*	
0619 3E 01	0153	CONT	MVI A,1 SHOW THE FILE OPEN
0618 32 C8 10	0154	STA	FILESTAT
061E CD 1C 07	0155	CALL	GETDLA C=DELAY VALUE
0621 CD D6 10	0156	CALL	SMPAREA AND BEGIN
0624 97	0157	SUB	A RETURN(0)
0625 C9	0158	RET	
0626	0159	*	
0626 21 0C D8	0160	PAUSE	LXI H,PMARK WRITE('PAUSE-MARK')
0629 CD 6D 00	0161	CALL	WRITE
062C 97	0162	SUB	A RETURN(0)
062D C9	0163	RET	
062E	0164	*	
062E 3A D5 10	0165	STOP	LDA BMODE TURN OFF THE A/D CONVERTER
0631 0F	0166	RRC	. UNLESS COMMANDED TO STAY ON
0632 D4 52 00	0167	CNC	ADPWR
0635 CD 9F 07	0168	CALL	FCLOSE
0638 21 4B 4F	0169	LXI	H,'OK'
063B D2 41 06	0170	JNC	STOP1
063E 21 4F 4E	0171	LXI	H,'NO'
0641 CD 4F 00	0172	STOP1	CALL SEND
0644 B7	0173	DRA	A
0645 C9	0174	RET	
0646	0175	*	
0646	0176	*	BEGIN PLAYBACK TO THE MASTER COMPUTER
0646	0177	*	
0646 3A C8 10	0178	PLAY	LDA FILESTAT IF FILE OPEN, CLOSE IT
0649 FE 01	0179	CP1	1 TO GET THE FILE PARAMETERS
064B CC 9F 07	0180	CZ	FCLOSE
064E	0181	*	
064E CD BA 06	0182	CALL	PHEAD PLAY HEADER
0651 3A C8 10	0183	LDA	FILESTAT IF AN ERROR IN FILE, QUIT
0654 FE 02	0184	CP1	2
0656 C8	0185	RZ	.
0657 2A C9 10	0186	LHLD	STADR POSITION HEAD TO START MARK
065A 3A C8 10	0187	LDA	STADR+2 OR THE FIRST VALID RECORD
065D CD 5B 00	0188	CALL	MARSET
0660	0189	*	
0660 CD 49 00	0190	PLOOP	CALL RECSTAT IF COMMAND READY, QUIT THIS
0663 C2 81 06	0191	JNZ	PLAYX
0666 CD 6A 00	0192	CALL	READ ELSE PLAY OUT DATA UNTIL END-MARK
0669 CD 4F 00	0193	CALL	SEND
066C	0194	*	
066C CD 73 00	0195	CALL	MARGET IF THE ADDRESS MATCHES
066F EB	0196	XCHG	. WHERE WE FOUND THE ENDMARK
0670 47	0197	MOV	B,A
0671 2A CC 10	0198	LHLD	ENADR QUIT
0674 CD 5B 08	0199	CALL	EQ16

0677 C2 60 06	0200	JNZ	PLDOP	
067A 3A CE 10	0201	LDA	ENADR+2	
067D BB	0202	CMP	B	
067E C2 60 06	0203	JNZ	PLDOP	
0681	0204	*		
0681 97	0205	PLAYX	SUB A	MAR = 0
0682 21 00 00	0206	LXI	H,0	
0685 CD 5B 00	0207	CALL	MARSET	
068B B7	0208	DRA	A	RETURN(0)
0689 C9	0209	RET		
068A	0210	*		
068A	0211	*	PLAY HEADER	
068A	0212	*		
068A 3E B2	0213	PHEAD	MVI A,0B2H	START HEADER PLAYBACK
068C CD CF 06	0214	CALL	SENDA	
068F 3E A1	0215	MVI	A,0A1H	FORMAT 1
0691 CD CF 06	0216	CALL	SENDA	
0694 3A C5 10	0217	LDA	REALF	REAL FREQUENCY CODE
0697 CD CF 06	0218	CALL	SENDA	
069A	0219	*		
069A 2A C9 10	0220	LHLD	STADR	START ADDRESS
069D CD 4F 00	0221	CALL	SEND	
06A0 2A CA 10	0222	LHLD	STADR+1	
06A3 CD DB 06	0223	CALL	DIV16	
06A6 CD 4F 00	0224	CALL	SEND	
06A9	0225	*		
06A9 2A CC 10	0226	LHLD	ENADR	PAGE OF END ADDRESS
06AC CD 4F 00	0227	CALL	SEND	
06AF 2A CD 10	0228	LHLD	ENADR+1	
06B2 CD DB 06	0229	CALL	DIV16	
06B5 CD 4F 00	0230	CALL	SEND	
06BB	0231	*		
06BB CD C9 0B	0232	CALL	ADRFMT [DE]->LIST	
06BB EB	0233	XCHG	.	
06BC CD CC 0B	0234	CALL	LNGFMT C=LENGTH OF FMT	
06BF 4F	0235	MOV	C,A	
06C0 CD CF 06	0236	CALL	SENDA	SEND LENGTH
06C3 79	0237	SQTYLP	MOV A,C	IF NO QTYS LEFT, QUIT
06C4 B7	0238	DRA	A	
06C5 CB	0239	RZ	.	
06C6 1A	0240	LDAX	D	SEND NEXT QTY DESCRIPTOR
06C7 CD CF 06	0241	CALL	SENDA	
06CA 13	0242	INX	D	
06CB 0D	0243	DCR	C	
06CC C3 C3 06	0244	JMP	SQTYLP	
06CF	0245	*		
06CF 6F	0246	SENDA	MOV L,A	
06D0 26 00	0247	MVI	H,0	
06D2 C3 4F 00	0248	JMP	SEND	
06D5 C3 CB 05	0249	JMP	ONOFF	

06D8	0250 †			
06D8 06 04	0251 DIV16	MVI	B,4	SHIFT HL RIGHT 4 BITS
06DA 97	0252	SUB	A	
06DB 29	0253 DVI	DAD	H	
06DC 17	0254	RAL	.	
06DD 05	0255	DCR	B	
06DE C2 DB 06	0256	JNZ	DVI	
06E1 6C	0257	MOV	L,H	
06E2 67	0258	MOV	H,A	
06E3 C9	0259	RET		
06E4	0260 †			
06E4	0261 †	COMPILE THE SAMPLING LIST INTO A PROGRAM		
06E4	0262 †			
06E4 CD 0B 07	0263 CMPROG	CALL	CHKFREQ	CHECK FREQUENCY
06E7 41	0264	MOV	B,C	SAVE THE DELAY CODE
06E8 CD CC 08	0265	CALL	LNGFMT	C=LENGTH OF THE FORMAT
06EB 4F	0266	MOV	C,A	
06EC CD C9 08	0267	CALL	ADRFMT	[DEJ]->LIST OF QTYS
06EF EB	0268	XCHG	.	
06F0 21 D6 10	0269	LXI	H,SMPAREA	[HL]->AREA TO PUT PROGRAM
06F3	0270 †			
06F3 3A C5 10	0271	LDA	REALF	IF THE FREQUENCY IS IN THE
06F6 FE 08	0272	CPI	INTYPE	INTERRUPT TIMES, GO
06F8 D2 00 07	0273	JNC	INTCMP	
06FB 06 02	0274	MVI	B,2	
06FD C3 40 0A	0275	JMP	COMPILE	
0700	0276 †			
0700 78	0277 INTCMP	MOV	A,B	GET THE DELAY CODE
0701 B7	0278	ORA	A	IF ZERO DELAY, REQUEST THIS
0702 06 00	0279	MVI	B,0	FROM THE COMPILER
0704 CA 40 0A	0280	JZ	COMPILE	
0707 04	0281	INR	B	IF SOME DELAY, ASK FOR IT
0708 C3 40 0A	0282	JMP	COMPILE	
0708	0283 †			
0708	0284 †	CHECK THAT THE REQUESTED FREQUENCY IS POSSIBLE		
0708	0285 †	SET REALF AT WHATEVER IS THE REAL FREQUENCY		
0708	0286 †			
0708 3A C4 10	0287 CHKFREQ	LDA	FREQ	ASSUME THE REAL RATE IS
070E 32 C5 10	0288	STA	REALF	THE REQUESTED RATE.
0711	0289 †			
0711 CD 1C 07	0290 CHKRT	CALL	GETDLA	C=DELAY VALUE
0714 D0	0291	RNC	.	IF VALID, RETURN
0715 21 C5 10	0292	LXI	H,REALF	ELSE DECREASE THE FREQUENCY CODE
0718 35	0293	DCR	M	AND TRY AGAIN
0719 C3 11 07	0294	JMP	CHKRT	
071C	0295 †			
071C 3A C5 10	0296 GETDLA	LDA	REALF	GET THE PROPER DELAY VALUE
071F 21 35 07	0297	LXI	H,DLATBL	TO REGULATE SPEED
0722 C9 51 08	0298	CALL	REF	
0725 4E	0299	MOV	C,M	

```

0726          0300 *
0726 JA C5 10 0301      LDA   REALF IF < INTERRUPT TYPE CODE
0729 FE 08     0302      CPI   INTYPE THEN RETURN(NO CRY)
072B 3F       0303      CMC
072C D0       0304      RNC
072D          0305 *
072D CD CC 08 0306      CALL  LNGFMT ELSE SUBTRACT THE LENGTH
0730 47       0307      MOV   B,A   OF THE LIST FROM THE DELAY
0731 79       0308      MOV   A,C   SINCE SAMPLING ALWAYS DELAYS
0732 90       0309      SUB   B     ITSELF.
0733 4F       0310      MOV   C,A
0734 C9       0311      RET   .
0735          0312 *
0735 CB       0313  DLATBL DB   200   10 HZ  CODES 0-7
0736 64       0314      DB   100   20 (ALL INTERRUPT TIME-BASED)
0737 28       0315      DB   40    50
0738 14       0316      DB   20    100
0739 0A       0317      DB   10    200
073A 04       0318      DB   4     500
073B 02       0319      DB   2     1000
073C 01       0320      DB   1     2000
073D          0321 *
073D          0322 INTYPE EQU  $-DLATBL ALL CODES BELOW ARE INTERRUPTS
073D 14       0323      DB   20    CODES 8-15 (CPU CYCLE COUNTING)
073E 0A       0324      DB   10
073F 06       0325      DB   6
0740 04       0326      DB   4
0741 03       0327      DB   3
0742 02       0328      DB   2
0743 01       0329      DB   1
0744 00       0330      DB   0
0745          0331 FRMAX EQU  $-DLATBL-2
0745          0332 *
0745 0A 00     0333 FRQTB DW   10    FREQUENCIES
0747 14 00     0334      DW   20
0749 32 00     0335      DW   50
074B 64 00     0336      DW  100
074D CB 00     0337      DW  200
074F F4 01     0338      DW  500
0751 EB 03     0339      DW 1000
0753 D0 07     0340      DW 2000
0755 A0 08     0341      DW 2976
0757 40 17     0342      DW 5952
0759 C1 26     0343      DW 9921
075B 21 3A     0344      DW 14881
075D B1 4D     0345      DW 19841
075F 42 74     0346      DW 29762
0761 B4 EB     0347      DW 59524
0763 B4 EB     0348      DW 59524
0765          0349 *

```


0765	0350	+	CALCULATE THE LENGTH OF TIME TO FILL MEMORY
0765	0351	+	
0765 3A C5 10	0352	GETDUR LDA	REALF REFERENCE FROM TO GET
0768 87	0353	ADD A	THE #SAMPLES SETS/SECND
0769 21 45 07	0354	LX1 H,FRQTB	
076C CD 51 08	0355	CALL REF	
076F CD 56 08	0356	CALL LODHL	
0772 E5	0357	PUSH H	
0773 CD CC 08	0358	CALL LNGFMT	GET THE #QTY IN THE LIST
0776 D1	0359	PDP D	
0777 CD 24 0C	0360	CALL MU21	AND MULTIPLY TO GET TDAL
077A 16 00	0361	MVI D,0	INPUT RATE TO MEMORY.
077C 5F	0362	MOV E,A	
077D CD 18 0C	0363	CALL FLT32	CONVERT TO FLOATING POINT
0780	0364	+	
0780 21 CF 10	0365	LX1 H,TRATE	AND SAVE THE TDAL RATE
0783 CD 03 0C	0366	CALL STOPP	
0786	0367	+	
0786 3A C7 10	0368	LDA ENBANK	CALCULATE # BANKS
0789 21 C6 10	0369	LX1 H,STBANK	TIMES 16K WORDS + 1000
078C 96	0370	SUB M	WHICH IS NBANKS + 24 BITS
078D 3C	0371	INR A	
078E 57	0372	MOV D,A	
078F 1E 00	0373	MVI E,0	
0791 26 00	0374	MVI H,0	
0793 CD 18 0C	0375	CALL FLT32	
0796	0376	+	
0796 21 CF 10	0377	LX1 H,TRATE	NOW DIVIDE BY THE INPUT RATE
0799 CD 09 0C	0378	CALL FDIV	AND RETURN (DEHL) MILLISECOND
079C C3 18 0C	0379	JMP FIX32	
079F	0380	+	
079F	0381	+	CLOSE THE FILE IN THE BURST MEMORY
079F	0382	+	
079F 21 0E F8	0383	FCLDSE LX1	H,ENARK WRITE ("END-MARK")
07A2 CD 6D 00	0384	CALL WRITE	
07A5	0385	+	
07A5 CD 70 00	0386	CALL REWIND	REPOSITION TAPE
07A8 11 0E F8	0387	LX1 D,ENARK	AND FIND ITS ADDRESS
07AB CD 16 08	0388	CALL SEARCH	
07AE D8	0389	RC .	IF ERROR, RETURN NOW.
07AF	0390	+	
07AF CD 73 00	0391	CALL MARGET	ELSE FOUND IT.
07B2 22 CC 10	0392	SHLD ENADR	SAVE ITS ADDRESS
07B5 32 CE 10	0393	STA ENADR+2	
07B8	0394	+	
07B8 CD 70 00	0395	CALL REWIND	REPOSITION TAPE
07BB CD 6A 00	0396	CALL READ	READ THE FIRST ENTRY
07BE 11 0A B8	0397	LX1 D,SMARK	LOOKING FOR THE START MARK
07C1 CD 5B 08	0398	CALL EQ16	IF THERE, IT WAS A SHORT BURST
07C4 C2 D5 07	0399	JNZ FSLDNG	

07C7	0400 #
07C7 CD 73 00	0401 FENDSET CALL MARGET GET THE TAPE POSITION
07CA 22 C9 10	0402 SHLD STADR AND SAVE FOR PLAYBACK
07CD 32 CB 10	0403 STA STADR+2
07D0 97	0404 SUB A SHOW THE BURST FILE CLDSED
07D1 32 CB 10	0405 STA FILESTAT
07D4 C9	0406 RET .
07D5	0407 #
07D5 2A CC 10	0408 FSLONG LHLD ENADR START AT THE ENDMARKER
07DB 3A CE 10	0409 LDA ENADR+2
07DB CD 5B 00	0410 CALL MARSET
07DE	0411 #
07DE CD CC 08	0412 CALL LNGFMT C=LENGTH DF RECDRD
07E1 4F	0413 MOV C,A
07E2	0414 #
07E2 CD 49 00	0415 FSNREC CALL RECSTAT IF COMMAND READY, QUIT
07E5 C0	0416 RNZ .
07E6 41	0417 MOV B,C START AT RECDRD CDUNT
07E7 CD 6A 00	0418 FS1REC CALL READ READ NEXT VALUE
07EA 11 0C DB	0419 LX1 D,PMARK IF A PAUSE MARK
07ED CD 5B 08	0420 CALL EQ16 DR AN END MARK, QUIT.
07F0 CA 03 08	0421 JZ FSFND
07F3 11 0E FB	0422 LX1 D,EMARK
07F6 CD 5B 08	0423 CALL EQ16
07F9 CA 03 08	0424 JZ FSFND
07FC 05	0425 DCR B ELSE CONTINUE WITHIN RECDRD
07FD C2 E7 07	0426 JNZ FSIREC
0800 C3 E2 07	0427 JMP FSNREC DR GET A NEW RECORD
0803	0428 #
0803 79	0429 FSFND MOV A,C COMPUTE PARTIAL RECORD
0804 90	0430 SUB B LENGTH IN ORDER TO SKIP IT
0805 F5	0431 PUSH PSW
0806 2A CC 10	0432 LHLD ENADR POSITION TAPE TD ENDMARK
0809 3A CE 10	0433 LDA ENADR+2
080C CD 5B 00	0434 CALL MARSET
080F F1	0435 POP PSW
0810 C4 47 08	0436 CNZ SKIP SKIP(A) WRDSD IF A!=0.
0813 C3 C7 07	0437 JMP FENDSET SET FILE END TD CLDSE
0816	0438 #
0816	0439 # TAPE-LIKE FUNCTIONSD
0816	0440 #
0816 3E 03	0441 SEARCH MVI A,3 MAXCNT = 30000H
0818 32 D4 10	0442 STA MAXCNT+2 (#WRDSD CASE SEARCH LENGTH)
0818 21 00 00	0443 LXI H,0
081E 22 D2 10	0444 SHLD MAXCNT
0821	0445 #
0821 CD 49 00	0446 SEAR1 CALL RECSTAT IF COMMAND RECEIVED, QUIT
0824 37	0447 STC .
0825 C0	0448 RNZ .
0826 D5	0449 SEAR2 PUSH D

0827 CD 6A 00	0450	CALL	READ	READ A VALUE THERE
082A D1	0451	POP	D	
082B CD 5B 08	0452	CALL	EQ16	IF SAME AS WE'RE LOOKING FOR
082E C8	0453	RZ	.	QUIT
082F	0454 *			
082F 21 D2 10	0455	LXI	H,MAXCNT	STOP COUNTING AFTER
0832 35	0456	DCR	M	THE MAX COUNT
0833 C2 26 06	0457	JNZ	SEAR2	
0836 23	0458	INX	H	
0837 35	0459	DCR	M	
0838 C2 21 08	0460	JNZ	SEAR1	
083B 23	0461	INX	H	
083C 35	0462	DCR	M	
083D C2 21 08	0463	JNZ	SEAR1	
0840 3E 02	0464	MVI	A,2	ERROR IN FILE
0842 32 C8 10	0465	STA	FILESTAT	
0845 37	0466	STC		
0846 C9	0467	RET		
0847	0468 *			
0847 F5	0469 SKIP	PUSH	PSW	SKIP # RECORDS
0848 CD 6A 00	0470	CALL	READ	
084B F1	0471	POP	PSW	
084C 3D	0472	DCR	A	
084D C2 47 08	0473	JNZ	SKIP	
0850 C9	0474	RET	.	
0851	0475 *			
0851 85	0476 REF	ADD	L	[HL]=[HL]+A
0852 6F	0477	MOV	L,A	
0853 D0	0478	RNC		
0854 24	0479	INR	H	
0855 C9	0480	RET		
0856	0481 *			
0856 7E	0482 LOGHL	MOV	A,M	[HL]=MEM[HL]
0857 23	0483	INX	H	
0858 66	0484	MOV	H,M	
0859 6F	0485	MOV	L,A	
085A C9	0486	RET		
085B	0487 *			
085B 7D	0488 EQ16	MOV	A,L	COMPARE HL AND DE FOR =
085C 88	0489	CMP	E	
085D C0	0490	RNZ		
085E 7C	0491	MOV	A,H	
085F BA	0492	CMP	D	
0860 C9	0493	RET	.	
0861	0494 *			
0861 7D	0495 MARK	MOV	A,L	
0862 D3 00	0496	OUT	0	
0864 7C	0497	MOV	A,H	
0865 D3 01	0498	OUT	I	
0867 C9	0499	RET		

```

0868 00      V 0500      OB   256  END-OF-MODULE
0869          0501 #
0869          0502 # VARIABLES
0869          0503 #
0869          0504      ORG   BSMPRAM
10C0          0505 DURATION DS 4    DURATION OF BURST IN MSEC
10C4          0506 FREQ   DS 1    DESIRED FREQUENCY
10C5          0507 REALF  DS 1    REAL FREQUENCY OF BURST USED
10C6          0508 STBANK DS 1    START BANK #
10C7          0509 ENBANK DS 1    END BANK #
10C8          0510 #
10C8          0511 FILESTAT DS 1    FILE STATUS (1=OPEN)
10C9          0512 STAOR  DS 3    START OF BURST
10CC          0513 ENADR  DS 3    END OF BURST
10CF          0514 TRATE  DS 3    TOTAL INPUT RATE TO MEMORY
10D2          0515 MAXCNT DS 3    MAXIMUM # SAMPLES TO SEARCH
10D5          0516 BMODE  DS 1    INTERNAL MODE (1=KEEP A/D ON)
10D6          0517 SMPAREA EQU #    SAMPLE LIST AREA
10D6          0518 #
10D6          0519 # EXTERNAL MODULE DEFINITIONS
10D6          0520 #
10D6          0521      ORG   BIO
0040          0522 BIOINIT DS 3
0043          0523 GETMASK DS 3
0046          0524 SETMASK DS 3
0049          0525 RECSTAT DS 3
004C          0526 RECEIVE DS 3
004F          0527 SEND   DS 3
0052          0528 ADPWR  DS 3
0055          0529 SAMPLE DS 3
0058          0530 MEMPWR DS 3
005B          0531 MARSET DS 3
005E          0532 BANKSET DS 3
0061          0533 MDDESET DS 3
0064          0534 SECOND DS 3
0067          0535 OSMS   DS 3
006A          0536 #
006A          0537 READ   DS 3
006D          0538 WRITE  DS 3
0070          0539 REWIND DS 3
0073          0540 MARGET DS 3
0076          0541 SETVECT DS 3
0079          0542 #
0079          0543      ORG   BCMP
0A40          0544 COMPILE DS 3
0A43          0545 #
0A43          0546      ORG   BFMT
08C0          0547 INIFMT DS 3
08C3          0548 SETFMT DS 3
08C6          0549 ADOFMT DS 3
  
```

08C9	0550	ADRFMT	DS	3
08CC	0551	LN6FMT	DS	3
08CF	0552	ENDFMT	DS	3
08D2	0553	*		
08D2	0554	ORG	FFP	
0C00	0555	LODFP	DS	3
0C03	0556	STOFFP	DS	3
0C06	0557	FMUL	DS	3
0C09	0558	FDIV	DS	3
0C0C	0559	FADD	DS	3
0C0F	0560	FSUB	DS	3
0C12	0561	FCMP	DS	3
0C15	0562	FNEG	DS	3
0C18	0563	FLT32	DS	3
0C1B	0564	FIX32	DS	3
0C1E	0565	FSQUA	DS	3
0C21	0566	FSQRT	DS	3
0C24	0567	MU21	DS	3

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE---BURST PROGRAM LOADER
0000          0003 * FILE : BLD.A
0000          0004 *
0000          0005 BLDCODE EQU   0BCH
0000          0006 *
0000          0007          ORG   BLD
0440 C3 46 04 0008          JMP   BLDINIT
0443 C3 4D 04 0009          JMP   BLD CMD
0446          0010 *
0446 21 02 12 0011 BLDINIT LX1  H,USER  POINT THE ADR REGISTER
0449 22 00 12 0012          SHLD  ADR    TO THE USER LOADING AREA
044C C9          0013          RET   .
044D          0014 *
044D 7C          0015 BLD CMD MOV  A,H    CHECK FOR WHICH MEM LOAD CMD
044E E8 FC      0016          ANI   0FCH
0450 FE BC      0017          CPI   BLD CODE
0452 37          0018          STC
0453 C0          0019          RNZ   .
0454 EB          0020          XCHG
0455 7A          0021          MOV   A,E   GET THE COMMAND AGAIN
0456 D6 BC      0022          SUI   BLD CODE REMOVE THE BIAS
0458 CA 69 04  0023          JZ    SADR L  AND COUNT OFF EACH NUMBER
045B 3D          0024          DCR   A
045C CA 6E 04  0025          JZ    SADR H
045F 3D          0026          DCR   A
0460 CA 73 04  0027          JZ    LOAD
0463 3D          0028          DCR   A
0464 CA 7C 04  0029          JZ    JUMP
0467 37          0030          STC   . IF UNKNOWN, RETURN(CRY)
0468 C9          0031          RET
0469          0032 *
0469 7E          0033 SADR L  MOV  A,E   SET LOW ADDRESS
046A 32 00 12  0034          STA   ADR
046D C9          0035          RET
046E          0036 *
046E 7E          0037 SADR H  MOV  A,E   SET HIGH ADDRESS
046F 32 01 12  0038          STA   ADR+1
0472 C9          0039          RET   .
0473          0040 *
0473 2A 00 12  0041 LOAD  LHLD  ADR   MEM[ADR++] = VALUE
0476 73          0042          MOV   M,E
0477 23          0043          INX   H
047B 22 00 12  0044          SHLD  ADR
047B C9          0045          RET   .
047C          0046 *
047C 3A 02 12  0047 JUMP  LDA   USER  EXECUTE USER PROGRAM
047F FE AA      0048          CPI   0AAH  CHECK CODE TO VERIFY PROGRAM THERE
0481 C0          0049          RNZ   .    IF NOT RIGHT, SIGNAL ERROR
  
```

0482 97	0050	SUB	A	RESET THE CODE
0483 32 02 12	0051	STA	USER	
0486 C3 03 12	0052	JMP	USER+1	
0489	0053 #			
0489	0054 #	VARIABLES		
0489	0055 #			
0489	0056	ORG	2LDRAM	
1200	0057	ADR	DS 2	USER LOAD ADDRESS
1202	0058	USER	DS 1024	USER PROGRAM LOADING AREA

```

0000          0001 *
0000          0002 * CRRES FLIGHT SOFTWARE---BURST FORMAT ROUTINES
0000          0003 * WRITTEN BY PETER R HARVEY
0000          0004 * FILE : BFMT.A
0000          0005 *
0000          0006 PSW   EQU   6
0000          0007 EOL   EQU   OFOH  END OF LIST INDICATOR
0000          0008 *
0000          0009 * MULTIPLEXOR QUANTITIES
0000          0010 *
0000          0011 BZFAST EQU   0
0000          0012 BXFAST EQU   1
0000          0013 BYFAST EQU   2
0000          0014 BV3   EQU   3
0000          0015 BV4   EQU   4
0000          0016 BV34  EQU   5
0000          0017 BV34AC EQU   6
0000          0018 BVISC EQU   7
0000          0019 BV12AC EQU   8
0000          0020 BV2   EQU   9
0000          0021 BV1   EQU  10
0000          0022 BV12  EQU  11
0000          0023 BDIRECT EQU  12
0000          0024 BAGCU  EQU  13
0000          0025 BGUARD EQU  14
0000          0026 BSTUB  EQU  15
0000          0027 TESTQTY EQU 03FH
0000          0028 *
0000          0029          ORG   BFMT
08C0 C3 DD 08          0030          JMP   INIFMT
08C3 C3 D2 08          0031          JMP   SETFMT
08C6 C3 24 09          0032          JMP   ADDFMT
08C9 C3 F3 08          0033          JMP   ADRFMT
08CC C3 1A 09          0034          JMP   LNGFMT
08CF C3 0D 09          0035          JMP   ENDFMT
08D2          0036 *
08D2          0037 * SET THE CURRENT FORMAT NUMBER
08D2          0038 * ON ENTRY: A IS THE FORMAT TO USE
08D2          0039 *
08D2 E6 0F          0040 SETFMT ANI   IS
08D4 32 30 10          0041          STA   CURFMT
08D7 3E FF          0042          MVI   A,-I  SET TO DELETE ON 1ST ADDFMT
08D9 32 31 10          0043          STA   INSFLAG
08DC C9          0044          RET
08DD          0045 *
08DD          0046 * INIT ALL RAM FORMATS
08DD          0047 *
08DD 21 33 10          0048 INIFMT LXI   H,RAMLIST
08E0 11 A4 09          0049          LXI   D,RAMDEF
  
```


08E3 0E 06	0050	MV1	C,RAMD _X -RAMDEF+1
08E5	0051	*	
08E5 1A	0052	1L1 LDAX	D PUT THE END MARKER INTD MEM
08E6 13	0053	1NX	D
08E7 77	0054	MDV	M,A
08E8 23	0055	1NX	H
08E9 0D	0056	DCR	C
08EA C2 E5 08	0057	JNZ	1L1
08ED 2D	0058	DCR	L
08EE 7D	0059	MDV	A,L RECORD THE END DF MEMDRY USED
08EF 32 32 10	0060	STA	RLEND
08F2 C9	0061	RET	
08F3	0062	*	
08F3	0063	*	ADDRESS THE CURRENT FDRMAT
08F3	0064	*	RETURNS [HL]=ADDRESS DF CURRENT FDRMAT
08F3	0065	*	
08F3 3A 30 10	0066	ADRFMT LDA	CURFMT IF LIST# <10, THEN ITS A
08F6 FE 0A	0067	CPI	10 LIST CDNTAINED IN RDM
08F8 21 7C 09	0068	LXI	H,RDMLIST
08FB DA 03 09	0069	JC	AL1
08FE 21 33 10	0070	LX1	H,RAMLIST ELSE ITS A RAM LIST
0901 D6 0A	0071	SU1	10
0903 B7	0072	AL1 DRA	A
0904 CB	0073	RZ	
0905 CD 0D 09	0074	CALL	ENDFMT FIND THE END DF THE LIST
0908 23	0075	1NX	H STEP DVER THE END MARKER
0909 3D	0076	DCR	A AND REPEAT TILL DONE
090A C3 03 09	0077	JMP	AL1
090D	0078	*	
090D F5	0079	ENDFMT PUSH	PSW
090E 7E	0080	EL1 MDV	A,M SEARCH TILL THE END MARKER
090F FE F0	0081	CPI	EDL
0911 CA 1B 09	0082	JZ	ELX
0914 23	0083	1NX	H
0915 C3 0E 09	0084	JMP	EL1
0918 F1	0085	ELX POP	PSW
0919 C9	0086	RET	
091A	0087	*	
091A	0088	*	OBTAIN THE LENGTH DF THE CURRENT LIST
091A	0089	*	
091A CD F3 08	0090	LNGFMT CALL	ADRFMT GET THE START ADDRESS
091D 5D	0091	MDV	E,L
091E CD 0D 09	0092	CALL	ENDFMT AND THE ADDRESS OF THE ENDMARK
0921 7D	0093	MDV	A,L
0922 93	0094	SUB	E RETURN LENGTH DF LIST
0923 C9	0095	RET	
0924	0096	*	
0924	0097	*	ADD A DTY TD THE CURRENT FORMAT
0924	0098	*	DN ENTRY: A HDLDS THE QTY
0924	0099	*	

0924 4F	0100	ADDFMT	MOV	C,A	ADD QTY TO CURRENT LIST
0925 3A 30 10	0101	LDA		CURFMT	IF ROM FORMAT, QUIT
0928 FE 0A	0102	CPI		I0	
092A D8	0103	RC			
092B	0104	*			
092B 21 31 10	0105	LXI		H,INSFLAG	IF 1ST ADD, DELETE ALL THE REST
092E 34	0106	INR		M	
092F 02 38 09	0107	JNZ		ADEND	
0932 CD F3 08	0108	CALL		ADRFMT	HL->START OF THIS FORMAT
0935 CD 60 09	0109	ADCLR	CALL	DELETE	DELETE 1 AT A TIME
0938 CA 35 09	0110	JZ		ADCLR	
093B	0111	*			
093B CB F3 08	0112	ADEND	CALL	ADRFMT	HL->END MARK FOR THIS FMT
093E CD 0D 09	0113	CALL		ENDFMT	
0941 79	0114	MOV		A,C	
0942	0115	*			
0942	0116	*		INSERT A VALUE AT [HL]	BY MOVING EVERYTHING ELSE OVER
0942	0117	*			
0942 F5	0118	INSERT	PUSH	PSW	SAVE VALUE AND ADDRESS
0943 E5	0119	PUSH		H	
0944 3A 32 10	0120	LDA		RLEND	GET THE CURRENT RAMLIST END
0947 FE 78	0121	CPI		RLMAX#256/256	IF AT MAXIMUM QUIT
0949 D2 5C 09	0122	JNC		INSX	
094C 5F	0123	MOV		E,A	[DE]->LAST BYTE OF RAMLIST
094D 54	0124	MOV		D,H	
094E	0125	*			
094E 3C	0126	INR		A	RLEND++
094F 32 32 10	0127	STA		RLEND	
0952	0128	*			
0952 1A	0129	INS1	LDAX	D	MOVE LAST ELEMENT UP
0953 13	0130	INX		D	
0954 12	0131	STAX		D	
0955 18	0132	DCX		D	
0956 18	0133	DCX		D	
0957 78	0134	MOV		A,E	
0958 B0	0135	CMP		L	
0959 02 52 09	0136	JNC		INS1	
095C	0137	*			
095C E1	0138	INSX	POP	H	
095D F1	0139	POP		PSW	
095E 77	0140	MOV		M,A	PUT VALUE IN MEMORY
095F C9	0141	RET			
0960	0142	*			
0960	0143	*		DELETE THE VALUE AT [HL]	
0960	0144	*		IF MEM[HL] IS AN END-OF-LIST,	RETURN NOT ZERO
0960	0145	*			
0960 7E	0146	DELETE	MOV	A,M	NEVER DELETE AN END OF LIST
0961 FE F0	0147	CPI		EOL	
0963 CA 7A 09	0148	JZ		RETNZ	
0966	0149	*			

0946 ES	0150	PUSH	H	SAVE THE START ADDRESS
0947 23	0151 DEL1	INX	H	MOVE THE VALUE FROM 1 BEYOND
0948 7E	0152	MOV	A,M	
0949 2B	0153	DCX	H	
094A 77	0154	MOV	M,A	
094B	0155 ↓			
094B 23	0156	INX	H	
094C 3A 32 10	0157	LDA	RLEND	IF RLEND WAS JUST MOVED
094E 8D	0158	CMF	L	QUIT, ELSE LOOP AROUND
0970 C2 67 09	0159	JNZ	DEL1	
0973	0160 ↓			
0973 21 32 10	0161	LYI	H,RLEND	RLEND--
0976 35	0162	DCR	M	
0977 E1	0163	POP	H	RESTORE THE DELETE ADDRESS
0978 97	0164	SUB	A	RETURN ZERO
0979 09	0165	RET		
097A 3C	0166 RETNZ	INR	A	
097B 09	0167	RET		
097C	0168 ↓			
097C 0B	0169 ROMLIST	DB	BV12	FMT#0
097D F0	0170	DB	EOL	
097E	0171 ↓			
097E 0B	0172	DB	BV12	FMT#1
097F 07	0173	DB	BV15C	
0980 F0	0174	DB	EOL	
0981	0175 ↓			
0981 0B	0176	DB	BV12	FMT#2
0982 05	0177	DB	BV34	
0983 F0	0178	DB	EOL	
0984	0179 ↓			
0984 0B	0180	DB	BV12	FMT#3
0985 05	0181	DB	BV34	
0986 07	0182	DB	BV15C	
0987 F0	0183	DB	EOL	
0988	0184 ↓			
0988 0B	0185	DB	BV12	FMT#4
0989 05	0186	DB	BV34	
098A 01	0187	DB	BXFAST	
098B 02	0188	DB	BYFAST	
098C 00	0189	DB	BIFAST	
098D F0	0190	DB	EOL	
098E	0191 ↓			
098E 00	0192	DB	BIFAST	FMT#5
098F 01	0193	DB	BXFAST	
0990 02	0194	DB	BYFAST	
0991 03	0195	DB	BV3	
0992 04	0196	DB	BV4	
0993 05	0197	DB	BV34	
0994 06 00	0198	DW	BV34AC	
0996 07	0199	DB	BV15C	

0997 08	0200	DB	BV12AC	
0998 09	0201	DB	BV2	
0999 0A	0202	DB	BV1	
099A 0B	0203	DB	BV12	
099E 0C	0204	DB	BDIRECT	
099C 0D	0205	DB	BAGCU	
099D 0E	0206	DB	BGUARD	
099E 0F	0207	DB	BSTUB	
099F F0	0208	DB	EOL	
09A0 F0	0209	DB	EOL	FMT#6
09A1 F0	0210	DB	EOL	FMT#7
09A2 F0	0211	DB	EOL	FMT#8
09A3 F0	0212	DB	EOL	FMT#9
09A4	0213 *			
09A4 F0	0214 RAMDEF	DB	EOL	FMT#10 RAM DEFAULT FORMATS 10-15
09A5 F0	0215	DB	EOL	FMT#11
09A6 F0	0216	DB	EOL	FMT#12
09A7 F0	0217	DB	EOL	FMT#13
09A8 F0	0218	DB	EOL	FMT#14
09A9 F0	0219	DB	EOL	FMT#15
09AA	0220 RAMDX	EQU	\$-1	
09AA 00	v 0221	DB	256	END-OF-FMT
09AB	0222 *			
09AB	0223 * VARIABLES			
09AB	0224 *			
09AB	0225	ORG	BFMTRAM	
1030	0226 CURFMT	DS	1	CURRENT LIST NUMBER
1031	0227 INSFLAG	DS	1	INSERT FLAG
1032	0228 RLEND	DS	1	RAM LIST END-OF-MEMORY
1033	0229 RAMLIST	DS	64+6	
1079	0230 RLMAX	EQU	\$-1	

0A60	0050 †			
0A60 CD EE 0A	0051	CMFEND	CALL	GENFIN GENERATE LOOP CODE
0A63 11 53 0B	0052		LX1	D,R0MRTL
0A66 21 94 10	0053		LX1	H,R0MRTL
0A69 0E 24	0054		MV1	C,RTLEN
0A6B CD 27 0B	0055		CALL	COPY
0A6E 2A 90 10	0056		LHLO	SLPTR RETURN(NEXT AVAIL MEMORY ADDR)
0A71 B7	0057		ORA	A
0A72 C9	0058		RET	
0A73	0059 †			
0A73	0060 †	COMPILE	LIST	03 THRU 0N
0A73	0061 †			
0A73 E5	0062	CMPLIST	PUSH	H SAVE THE LIST ADDRESS
0A74 C5	0063		PUSH	B AND ITS LENGTH
0A75 23	0064		INX	H STEP PAST THE 1ST TWO ELEMENTS
0A76 23	0065		INX	H
0A77 0D	0066		OCR	C
0A78 00	0067		OCR	C
0A79	0068 †			
0A79 CD A6 0A	0069	CL1	CALL	GENSMP GENERATE A SAMPLE
0A7C CD CB 0A	0070		CALL	GENSD AND A SAMPLE-DELAY
0A7F 23	0071		INX	H STEP TO NEXT LIST ELEMENT
0A80 00	0072		DCR	C
0A81 C2 79 0A	0073		JNZ	CL1
0A84 C1	0074		POP	B
0A85 E1	0075		POP	H
0A86 C9	0076		RET	
0A87	0077 †			
0A87	0078 †	GENSMP	--	CODE GENERATION ROUTINES
0A87	0079 †			
0A87	0080	EOC	EQU	-1 END-OF-CODE
0A87	0081 †			
0A87 3A A0 0A	0082	GENBEGIN	LDA	8C1 START BY LOADING THE
0A8A CD 1C 0B	0083		CALL	PUT FREQUENCY CODING.
0A8D 78	0084		MOV	A,B
0A8E CD 1C 0B	0085		CALL	PUT
0A91	0086 †			
0A91 11 A2 0A	0087		LX1	D,BEGCODE INSERT THE INIT ROUTINE
0A94 CD 00 0B	0088		CALL	PUTN
0A97 E5	0089		PUSH	H
0A98 2A 90 10	0090		LHLO	SLPTR THEN RECORD WHERE THE LOOP GOES
0A9B 22 92 10	0091		SHLD	LPPTR
0A9E E1	0092		POP	H
0A9F C9	0093		RET	
0AA0	0094 †			
0AA0 06 00	0095	BC1	MV1	B,0
0AA2 CD 30 0B	0096	BEGCODE	CALL	STARBURST
0AA5 FF	0097		DB	EOC
0AA6	0098 †			
0AA6 7E	0099	GENSMP	MOV	A,M IF TEST QTY, USE SPECIAL CODE

0AA7 FE 3F	0100	CPI	TESTQTY
0AA9 11 C3 0A	0101	LXI	D,TSTCODE
0AAC CA 0D 0B	0102	JZ	PUTN
0AAF 3A C0 0A	0103	LDA	SMPCODE PUT THE OPCODE
0AB2 CD 1C 0B	0104	CALL	PUT
0AB5 7E	0105	MOV	A,M THEN THE QTY#2
0AB6 87	0106	ADD	A
0AB7 CD 1C 0B	0107	CALL	PUT
0ABA 3A C2 0A	0108	LDA	SMPCODE+2
0ABD C3 1C 0B	0109	JMP	PUT
0AC0 2A 00 B0	0110	SMPCODE LHL	MEM+ADC+0 EXAMPLE OF A/D SAMPLE
0AC3 2A 90 90	0111	TSTCODE LHL	MEM+TEST EXAMPLE OF TEST SAMPLE
0AC6 2C	0112	INR	L
0AC7 22 90 10	0113	SHLD	TEST
0ACA FF	0114	DB	EOC
0ACB	0115	*	
0ACB	0116	*	GENERATE SAMPLE DELAY (BETWEEN SAMPLES)
0ACB	0117	*	
0ACB 11 01 0A	0118	GENSD LXI	D,S0CODE COPY SOME CODE
0ACE C3 0D 0B	0119	JMP	PUTN
0AD1 DB 7F	0120	S0CODE IN	7FH DELAY 10 CYCLES
0AD3 FF	0121	DB	EOC
0AD4	0122	*	
0AD4	0123	*	GENERATE TIMING
0AD4	0124	*	
0AD4 78	0125	SENTIM MOV	A,B GET THE TIMING INFO
0AD5 87	0126	ORA	A IF NO DELAY, NO CODE.
0AD6 C8	0127	RZ	.
0AD7 3D	0128	DCR	A IF B=1, SOFTWARE DELAY
0AD8 11 E9 0A	0129	LXI	D,SOFTIM
0ADB CA 0D 0B	0130	JZ	PUTN
0ADE 11 E4 0A	0131	LXI	D,INTTIM ELSE INTERRUPT DELAY
0AE1 C3 0D 0B	0132	JMP	PUTN
0AE4	0133	*	
0AE4 69	0134	INTTIM MOV	L,C COUNT THE INTERRUPTS
0AE5 CD 9F 10	0135	CALL	INTDLA
0AE8 FF	0136	DB	EOC
0AE9 69	0137	SOFTIM MOV	L,C COUNT CYCLES
0AEA CD 94 10	0138	CALL	SOFDLA
0AED FF	0139	DB	EOC
0AEE	0140	*	
0AEE	0141	*	GENERATE LOOP END CODE
0AEE	0142	*	
0AEE 11 06 0B	0143	GENFIN LXI	D,LPCODE
0AF1 CD 0D 0B	0144	CALL	PUTN
0AF4 3A 92 10	0145	LDA	LPPTR PUT THE LOOP ADDRESS IN
0AF7 CD 1C 0B	0146	CALL	PUT
0AFA 3A 93 10	0147	LDA	LPPTR+1
0AFD CD 1C 0B	0148	CALL	FUT
0B00 11 09 0B	0149	LXI	D,FINCODE

0803 C3 0D 08	0150	JMP	PUTN	
0806	0151	*		
0806	0152	LPCODE EQU		* THE CODE WHICH IMPLEMENTS A LOOP
0806 D2 FF FF	0153	JNC	EDC	(COPY UP TO THE JNC OPCODE)
0809 C3 48 08	0154	FINCODE JMP	ENDBURST	
080C FF	0155	DB	EDC	
080D	0156	*		
080D 1A	0157	PUTN LDAX	D	COPY CODE FROM MEM[DE]
080E FE FF	0158	CP1	EDC	
0810 C8	0159	RZ		
0811 CD IC 08	0160	CALL	PUT	
0814 13	0161	INX	D	
0815 C3 0D 08	0162	JMP	PUTN	
0818	0163	*		
0818 22 90 10	0164	STPUT SHLD	SLPTR	SET SAMPLE LIST POINTER
081B C9	0165	RET		
081C	0166	*		
081C E5	0167	PUT PUSH	H	SAVE [HL]
081D 2A 90 10	0168	LHLD	SLPTR	PUT ACCUM INTO MEM AT SLPTR++
0820 77	0169	MOV	M,A	
0821 23	0170	INX	H	
0822 22 90 10	0171	SHLD	SLPTR	
0825 E1	0172	PDF	H	
0826 C9	0173	KET		
0827	0174	*		
0827 1A	0175	COPY LOAX	D	
0828 77	0176	MOV	M,A	
0829 13	0177	INX	D	
082A 23	0178	INX	H	
082B 0D	0179	DCR	C	
082C C2 27 08	0180	JNZ	COPY	
082F C9	0181	RET		
0830	0182	*		
0830	0183	*		THE S COMPILER'S RUN TIME LIBRARY
0830	0184	*		(SEE ABOVE FOR RESTART 1 AND 2)
0830	0185	*		
0830	0186	*		STARBURST INITIALIZES THE SYSTEM SO THAT THE COMPILED
0830	0187	*		SAMPLING ROUTINE ALWAYS WORKS.
0830	0188	*		
0830	0189	STARBURST EQU	*	
0830 3E 01	0190	MVI	A,I	SET MEMORY FOR AUTOWRITE
0832 CD 61 00	0191	CALL	MODESET	
0835 3E 03	0192	MVI	A,3	SET A/O FOR AUTO-CONVERT
0837 32 01 30	0193	STA	ADCTL	
083A 21 00 00	0194	LXI	H,0	START TEST COUNTER AT 0
083D 22 90 10	0195	SHLD	TEST	
0840 3E 01	0196	MVI	A,I	SET I/O MODE TO SET CARRY
0842 CD 79 00	0197	CALL	IOMODE	WHEN COMMAND PENDING
0845 FB	0198	EI	.	ALWAYS KEEP COMMAND INPUT (7.5) ENABLED
0846 3A 90 10	0199	LOA	8CMPRAM	REFERENCE RAM FOR LOW POWER

0B49 B7	0200	ORA	A	CLEAR CARRY
0B4A C9	0201	RET		
0B4B	0202	‡		
0B4B 97	0203	ENDBURST SUB	A	REMOVE COMMAND-PENDING
0B4C C8 79 00	0204	CALL	ICMODE	STATUS RETURN
0B4F 97	0205	SUB	A	CLEAR AUTOWRITE MODE
0B50 C3 61 00	0206	JMP	MODESET	
0B53	0207	‡		
0B53	0208	‡	THE DELAY ROUTINES	
0B53	0209	‡		
0B53	0210	ROMRTL EQU	‡	
0B53	0211	SOFDLA EQU	RAMRTL	
0B53 FB	0212	EI		
0B54 2D	0213	DCR	L	
0B55 0B	0214	RZ		
0B56 0B 7F	0215	IN	7FH	
0B5B 00	0216	NOP		
0B59 00	0217	NOP		
0B5A D2 94 10	0218	JNC	SOFDLA	
0B5D C7	0219	RET		
0B5E	0220	‡		
0B5E	0221	INTDLA EQU	‡-ROMRTL+RAMRTL	
0B5E D4 A7 10	0222	CNC	INTWAIT	
0B61 2D	0223	DCR	L	
0B62 C2 9F 10	0224	JNZ	INTDLA	
0B65 C9	0225	RET		
0B66	0226	‡		
0B66	0227	INTWAIT EQU	‡-ROMRTL+RAMRTL	
0B66 F3	0228	DI		
0B67 F5	0229	FUSH	PSW	
0B6B 06 20	0230	MVI	B, 20H	
0B6A	0231	IWH EQU	‡-ROMRTL+RAMRTL	
0B6A 20	0232	DB	RIM	
0B6B A0	0233	ANA	B	
0B6C C2 4B 10	0234	JNZ	IWH	
0B6F	0235	‡		
0B6F	0236	IWL EQU	‡-ROMRTL+RAMRTL	
0B6F 20	0237	DB	RIM	
0B70 A0	0238	ANA	B	
0B71 CA 80 10	0239	JZ	IWL	
0B74 F1	0240	POP	PSW	
0B75 FB	0241	EI		
0B76 C9	0242	RET		
0B77	0243	RTLEN EQU	‡-ROMRTL	
0B77 00	V 0244	DB	256	END-OF-BCMP MODULE
0B7B	0245	‡		
0B7B	0246	‡	EXTERNALS	
0B7B	0247	‡		
0B7B	0248	ORG	B10	
0B40	0249	BIDINIT DS	3	

0043	0250 GETMASK DS	3
0046	0251 SETMASK DS	3
0049	0252 RECSTAT DS	3
004C	0253 RECEIVE DS	3
004F	0254 SEND DS	3
0052	0255 ADPWR DS	3
0055	0256 SAMPLE DS	3
0058	0257 MEMPWR DS	3
005B	0258 MARSET DS	3
005E	0259 BANKSET DS	3
0061	0260 MODESET DS	3
0064	0261 SECOND DS	3
0067	0262 D5MS DS	3
006A	0263 READ DS	3
006D	0264 WRITE DS	3
0070	0265 REWIND DS	3
0073	0266 MARGET DS	3
0076	0267 SETVECT DS	3
0079	0268 IOMODE DS	3

0000		0001 :	
0000		0002 :	CARES FLIGHT SOFTWARE---FAST FLOATING POINT
0000		0003 :	WRITTEN BY PETER HARVEY
0000		0004 :	FILE : BFFP.A
0000		0005 :	
0000		0006 :	F.P. REGISTER IS CDE.
0000		0007 :	FORMAT IS SIGN(1)+EXP(7)+MANTISSA(16)
0000		0008 :	NO HIDDEN BIT
0000		0009 :	
0000		0010 FSW	EQU 6
0000		0011 SP	EQU 6
0000		0012 :	
0000		0013	ORG FFP
0000 C3 27 0C		0014	JMP LODFFP
0003 C3 2D 0C		0015	JMP STOFFP
0006 C3 33 0C		0016	JMP FMUL
0009 C3 5E 0C		0017	JMP FDIV
000C C3 F2 0C		0018	JMP FADD
000F C3 EB 0C		0019	JMP FSUB
0012 C3 DC 0C		0020	JMP FCMF
0015 C3 48 0C		0021	JMP FNEG
0018 C3 62 0D		0022	JMP FLT32
001B C3 0D 0D		0023	JMP FIX32
001E C3 26 0E		0024	JMP FSQRA
0021 C3 2F 0E		0025	JMP FSORT
0024 C3 5D 0E		0026	JMP MU21
0027		0027 :	
0027 4E		0028 LODFFP	MOV C,M
0028 23		0029	INX H
0029 56		0030	MOV D,M
002A 23		0031	INX H
002B 5E		0032	MOV E,M
002C 09		0033	RET .
002D		0034 :	
002D 71		0035 STOFFP	MOV M,C
002E 23		0036	INX H
002F 72		0037	MOV M,D
0030 23		0038	INX H
0031 73		0039	MOV M,E
0032 09		0040	RET .
0033		0041 :	
0033		0042 :	F.P. MULTIPLY ROUTINE
0033		0043 :	
0033 7A		0044 FMUL	MOV A,D IF X=0. QUIT NOW
0034 87		0045	ORA A
0035 09		0046	RZ .
0036 46		0047	MOV B,M LOAD PARAM FROM MEM
0037 23		0048	INX H INTO BHL FORMAT
0038 7E		0049	MOV A,M

0C39 B7	0050	DRA	A	IF ZERO THEN SET TO 0
0C3A CA C3 0D	0051	JZ	RET0	
0C3D 23	0052	INX	H	ELSE LOAD THE REST
0C3E 6E	0053	MOV	L,M	
0C3F 67	0054	MOV	H,A	
0C40	0055	*		
0C40 78	0056 FMS33	MOV	A,B	IF SAME SIGN, GO
0C41 A9	0057	XRA	C	
0C42 F2 50 0C	0058	JP	FMS33	
0C45 CD D4 0D	0059	CALL	STRIP	REMOVE SIGNS FROM B&C
0C48 CD 50 0C	0060	CALL	FMS33	MULTIPLY THEN NEGATE
0C48	0061	*		
0C4B	0062 FNEG	EQU	*	
0C4B 79	0063 NEGFP	MOV	A,C	AND NEGATE F.P
0C4C EE B0	0064	XRI	80H	
0C4E 4F	0065	MOV	C,A	
0C4F C9	0066	RET	.	
0C50	0067	*		
0C50	0068	*	F.P.	MULTIPLY POSITIVES ONLY
0C50	0069	*		
0C50 78	0070 FMS33	MOV	A,B	ADD EXPONENTS
0C51 B1	0071	ADD	C	
0C52 D6 40	0072	SUI	40H	ADJUST BACK TO EXCESS 64
0C54 FA C9 0D	0073	JM	ERCHK	IF MINUS, CHECK THE ERROR
0C57 4F	0074	MOV	C,A	
0C58 CD 43 0E	0075	CALL	MU22F	[AHL.]= DE X HL
0C5B C3 AF 0D	0076	JMP	NCHK	SHIFT UNTIL AHL NORMED,ROUND OFF
0C5E	0077	*		
0C5E	0078	*	F.P.	DIVIDE
0C5E	0079	*		
0C5E 7A	0080 FDIV	MOV	A,D	IF ZERO DIVIDEND, QUIT
0C5F B7	0081	DRA	A	
0C60 CB	0082	RZ	.	
0C61 46	0083	MOV	B,M	PICK UP DIVISOR
0C62 23	0084	INX	H	
0C63 7E	0085	MOV	A,M	
0C64 B7	0086	DRA	A	IF DIVISOR 0, OVERFLOW
0C65 CA CE 0D	0087	JZ	OVERFLOW	
0C68 23	0088	INX	H	
0C69 6E	0089	MOV	L,M	
0C6A 67	0090	MOV	H,A	
0C6B	0091	*		
0C6B 78	0092	MOV	A,B	IF SAME SIGN, DO
0C6C A9	0093	XRA	C	SAME SIGNED VERSION
0C6D F2 79 0C	0094	JP	FMS33	
0C70 CD D4 0D	0095	CALL	STRIP	REMOVE SIGNS
0C73 CD 79 0C	0096	CALL	FMS33	DIVIDE OUT
0C76 C3 4B 0C	0097	JMP	NEGFP	AND NEGATE
0C79	0098	*		
0C79 79	0099 FMS33	MOV	A,C	EXP=C-B+40H

0C7A 90	0100	SUB	B	
0C7B C6 40	0101	ADI	40H	
0C7D FA C9 0D	0102	JM	ERCHK	
0CB0 4F	0103	MOV	C,A	
0CB1 C5	0104	FUSH	B	SAVE EXPONENT
0C32	0105	*		
0C32 7C	0106	MOV	A,H	BC=-DIVISOR
0CB3 2F	0107	CMA		
0CB4 47	0108	MOV	B,A	
0CB5 7D	0109	MOV	A,L	
0CB6 2F	0110	CMA		
0CB7 4F	0111	MOV	C,A	
0CB8 03	0112	INX	B	
0CB9	0113	*		
0CB9	0114	*		IF THE REMAINDER STARTS AS LARGE AS
0CB9	0115	*		THE DIVISOR, THE FIRST BIT IS 1
0CB9	0116	*		
0CB9 62	0117	MOV	H,D	HL=REMAINDER
0CBA 6B	0118	MOV	L,E	
0CB8 09	0119	DAD	B	HL=REMAINDER-DIVISOR
0CB8 DA A9 0C	0120	JC	FBITI	
0CBF	0121	*		
0CBF	0122	*		IF REMAINDER LESS THAN DIVISOR, THE FIRST
0CBF	0123	*		BIT (INTEGER PART) IS ZERO. DIVIDE FOR
0CBF	0124	*		FRACTIONAL PART WHICH WILL BE AUTOMATICALLY
0CBF	0125	*		NORMALIZED.
0CBF	0126	*		
0CBF EB	0127	XCHG	.	HL=REMAINDER AGAIN
0C90 3E 10	0128	MVI	A,16	
0C92 C0 C2 0C	0129	CALL	FDSHF	[DE]=[HL]*2/[BC]
0C95 29	0130	DAD	H	IF REMAINDER>B000H
0C96 DA 9F 0C	0131	JC	DVRND	THEN ROUND UP
0C99 09	0132	DAD	B	IF NEXT BIT WOULD BE 1
0C9A DA 9F 0C	0133	JC	DVRND	THEN ROUND UP
0C9D C1	0134	POP	B	RESTORE EXPONENT
0C9E C9	0135	RET	.	NO NORMALIZATION REQD
0C9F	0136	*		
0C9F C1	0137	DVRND	POP	B C=EXPONENT
0CA0 1C	0138	RND	INR	E ROUND OFF DE
0CA1 C0	0139	RNZ	.	BUT DON'T PRODUCE
0CA2 14	0140	INR	D	A ZERO
0CA3 C0	0141	RNZ	.	
0CA4 11 00 B0	0142	LXI	D,B000H	IF ZERO, THEN
0CA7 0C	0143	INR	C	UP THE EXPONENT
0CAB C9	0144	RET	.	
0CA9	0145	*		
0CA9	0146	*		FIRST BIT=1. DIVIDE OUT 16 MORE BITS
0CA9	0147	*		USING WHAT'S LEFT OF THE REMAINDER IN HL
0CA9	0148	*		
0CA9 3E 10	0149	FBIT1	MVI	A,16

0CAB 11 FF FF	0150	LXI	D,-1	
0CAE CD C2 0C	0151	CALL	FDSHF	[DEJ]=[HLJ]/[RC]
0CB1	0152	*		
0CB1 C1	0153	POP	B	RESTORE THE EXPONENT
0CB2 0C	0154	INR	C	ADJUST SINCE 1ST BIT=1
0CB3 37	0155	STC	.	RIGHT SHIFT A 1 INTO DE
0CB4 7A	0156	MOV	A,D	
0CB5 1F	0157	RAR		
0CB6 57	0158	MOV	D,A	
0CB7 7B	0159	MOV	A,E	
0CB8 1F	0160	RAR		
0CB9 5F	0161	MOV	E,A	
0CBA D0	0162	RNC	.	IF 17TH BIT WAS 0, STOP
0CBB C3 A0 0C	0163	JMP	ROND	ELSE ROUND OFF
0CBE	0164	*		
0CBE	0165	*		DIVIDE NORMALIZED INTEGERS FOR F.P.
0CBE	0166	*		
0CBE 33	0167	FDSTK	INX	SP REMOVE PARTIAL REMAINDER
0CBF 33	0168	INX	SP	FROM STACK
0CC0 3D	0169	FDTST	DCR	A DECR BIT COUNTER
0CC1 C8	0170	RZ	.	
0CC2 29	0171	FDSHF	DAD	H BRING DOWN A BIT INTO REM
0CC3 DA D4 0C	0172	JC	SUBIT	IF >=10000, THEN SUBTRACT
0CC6 EB	0173	XCHG	.	
0CC7 29	0174	DAD	H	AND SHIFT RESULT REG
0CC8 EB	0175	XCHG		
0CC9	0176	*		
0CC9 1C	0177	INR	E	ASSUME RESULT=1
0CCA E5	0178	FDV22	FUSH	H SAVE REMAINDER ON STK
0CCB 09	0179	DAD	B	IF REM<DIVISOR, LEAVE REM ALONE
0CCD DA BE 0C	0180	JC	FDSTK	
0CCF E1	0181	POP	H	ELSE RESTORE REMAINDER
0CD0 1D	0182	DCR	E	SET RESULT BIT=0
0CD1 C3 C0 0C	0183	JMP	FDTST	
0CD4	0184	*		
0CD4 EB	0185	SUBIT	XCHG	.
0CD5 29	0186	DAD	H	FINISH THE SHIFT
0CD6 EB	0187	XCHG	.	
0CD7 09	0188	DAD	B	SUBTRACT DIVISOR
0CD8 1C	0189	INR	E	SET RESULT BIT
0CD9 C3 C0 0C	0190	JMP	FDTST	
0CDC	0191	*		
0CDC	0192	*		F.P. COMPARE
0CDC	0193	*		ON EXIT: ZERO SET IF EQUAL, CARRY IF LESS THAN
0CDC	0194	*		CDE UNTOUCHED
0CDC	0195	*		
0CDC C5	0196	FCMP	FUSH	B SAVE CDE
0CDD D5	0197	FUSH	D	
0CDE CD EB 0C	0198	CALL	FSUB	SUBTRACT THE TWO
0CE1 7A	0199	MOV	A,D	IF RESULT=0, RET

0CE2 B7	0200	ORA	A	
0CE3 CA E8 0C	0201	JZ	FCMPX	
0CE6 79	0202	MOV	A,C	IF NEGATIVE, THEN
0CE7 07	0203	RLC	.	SET CARRY, ELSE NO CARRY
0CE8 D1	0204 FCMPX	PDP	D	RESTORE CDE
0CE9 D1	0205	FDP	B	
0CEA C9	0206	RET	.	
0CEB	0207 *			
0CEB	0208 *	F.P.	SUB	
0CEB	0209 *			
0CEB 7E	0210 FSUB	MOV	A,M	INVERT SIGN OF 2ND
0CEC EE 86	0211	XRI	B,H	PARAMETER
0CEE 47	0212	MOV	B,A	
0CEF C3 F3 0C	0213	JMP	FAD1	
0CF2	0214 *			
0CF2	0215 *	F.P.	ADD	
0CF2	0216 *			
0CF2 46	0217 FADD	MOV	B,M	LOAD UP
0CF3 23	0218 FAD1	INX	H	
0CF4 7E	0219	MOV	A,M	
0CF5 23	0220	INX	H	
0CF6 6E	0221	MOV	L,M	
0CF7 67	0222	MOV	H,A	
0CF8 97	0223	SUB	A	
0CF9 BC	0224	CMP	H	IF BHL=0, QUIT
0CFA C8	0225	RZ	.	
0CFB BA	0226	CMP	D	IF CDE=0, QUIT
0CFC DA 5F 0D	0227	JZ	SWITCH	
0CFF	0228 *			
0CFF 79	0229	MOV	A,C	COMPUTE EXP DIFFERENCE
0D00 90	0230	SUB	B	
0D01 87	0231	ADD	A	
0D02 F2 0B 0D	0232	JP	POSDX	
0D05 78	0233	MOV	A,B	SWAP CDE FOR BHL
0D06 41	0234	MOV	B,C	
0D07 4F	0235	MOV	C,A	
0D08 EB	0236	XCHG	.	
0D09 90	0237	SUB	B	COMPUTE EXP DIFFERENCE
0D0A B7	0238	ADD	A	AGAIN
0D0B CA 15 0D	0239 POSDX	JZ	ADSUB	
0D0E 0F	0240	RRC	.	DIV BY 2
0D0F FE 10	0241	CPI	16	IF CDE>>BHL, QUIT
0D11 D0	0242	RNC		
0D12 CD 47 0D	0243	CALL	SHFHL	REDUCE HL A TIMES
0D15	0244 *			
0D15 78	0245 ADSUB	MOV	A,B	IF SIGNS DIFFER, GO
0D16 A9	0246	XRA	C	
0D17 FA 25 0D	0247	JM	DIFFER	
0D1A 19	0248	DAD	D	ADD DE TO HL
0D1B EB	0249	XCHG	.	IF NO CARRY,

0D1C D0	0250	RNC	.	RETURN CDE
0D1D	0251 *			
0D1D 7A	0252 RITE1	MOV	A,D	ELSE SHIFT RIGHT ONE
0D1E 1F	0253	RAR	.	INCLUDING THE CARRY
0D1F 57	0254	MOV	D,A	
0D20 7B	0255	MOV	A,E	
0D21 1F	0256	RAR		
0D22 5F	0257	MOV	E,A	
0D23 0C	0258	INR	C	ADJUST EXPONENT
0D24 C9	0259	RET	.	
0D25	0260 *			
0D25 7B	0261 DIFFER	MOV	A,E	IF DE<HL,
0D26 95	0262	SUB	L	THEN NORM(B:HL-DE)
0D27 7A	0263	MOV	A,D	
0D28 9C	0264	SBB	H	
0D29 DA 3A 0D	0265	JC	SUBD	
0D2C	0266 *			
0D2C 57	0267	MOV	D,A	ELSE NORM(C:DE-HL)
0D2D 7B	0268	MOV	A,E	
0D2E 95	0269	SUB	L	
0D2F 5F	0270	MOV	E,A	
0D30 21 00 00	0271	LX1	H,0	
0D33 B2	0272	ORA	D	IF DE<>0,
0D34 C2 74 0D	0273	JNZ	NORM	NORMALIZE WITH C
0D37 0E 00	0274	MVI	C,0	IF HL-DE=0, RETURN
0D39 C9	0275	RET	.	
0D3A	0276 *			
0D3A 7D	0277 SUBD	MOV	A,L	DE = DE - HL
0D3B 93	0278	SUB	E	
0D3C 5F	0279	MOV	E,A	
0D3D 7C	0280	MOV	A,H	
0D3E 9A	0281	SBB	D	
0D3F 57	0282	MOV	D,A	
0D40 4B	0283	MOV	C,B	USE BHL'S EXPONENT
0D41 21 00 00	0284	LX1	H,0	AND SHIFT IN ZERDES
0D44 C3 74 0D	0285	JMP	NORM	NORMALIZE
0D47	0286 *			
0D47	0287 *			SHIFT HL RIGHT A TIMES
0D47	0288 *			
0D47 D6 0B	0289 SHFHL	SUI	B	IF LT B, GO NOW
0D49 DA 52 0D	0290	JC	LTB	
0D4C 6C	0291	MOV	L,H	SHIFT B
0D4D 26 00	0292	MVI	H,0	
0D4F C8	0293	RZ	.	IF EXACTLY B, RETURN
0D50 D6 0B	0294	SUI	B	ELSE DO SECOND B
0D52 C5	0295 LTB	PUSH	B	SAVE EXPONENTS
0D53 47	0296	MOV	B,A	SAVE INVERTED COUNTER
0D54 97	0297	SUB	A	CLEAR ACCUM
0D55 29	0298 SHF1	DAD	H	SHIFT LEFT
0D56 8F	0299	ADC	A	INTO A FROM HL

0D57 04	0300	INR	B	CDUNT UP TO 0
0D58 C2 55 0D	0301	JNZ	SHF1	
0D5B C1	0302	PDP	B	RESTDRE EXPS
0D5C 6C	0303	MOV	L,H	
0D5D 67	0304	MDV	H,A	
0D5E C9	0305	RET	.	
0D5F	0306 †			
0D5F EB	0307 SWITCH XCHG	.		CDE= BHL
0D60 4B	0308	MOV	C,B	
0D61 C9	0309	RET	.	
0D62	0310 †			
0D62	0311 †	CONVERT 32 BIT DATA TO F.P. FORMAT		
0D62	0312 †			
0D62 7A	0313 FLT32	MOV	A,D	IF POSITIVE, JUST NDRM
0D63 B7	0314	DRA	A	
0D64 0E 60	0315	MVI	C,64+32	WITH LSB=2**0 TO BEGIN
0D66 F2 74 0D	0316	JP		NDRM
0D69 CD BC 0E	0317	CALL	NEG32	NEGATE DEHL
0D6C CD 74 0D	0318	CALL	NDRM	NDM NDRMALIZE
0D6F 79	0319	MOV	A,C	AND NEGATE FP
0D70 F6 B0	0320	DRI	80H	
0D72 4F	0321	MDV	C,A	
0D73 C9	0322	RET	.	
0D74	0323 †			
0D74	0324 †	NDRMALIZE C:DEHL TO F.P. NORMAL FORM		
0D74	0325 †			
0D74 79	0326 NDRM	MDV	A,C	IF C NEGATIVE, TRAP IT
0D75 B7	0327	DRA	A	
0D76 F2 B2 0D	0328	JP		NORMF
0D79 E6 7F	0329	ANI	7FH	
0D7B 4F	0330	MDV	C,A	
0D7C CD B2 0D	0331	CALL	NDRMP	
0D7F C3 4B 0C	0332	JMP	NEGFP	AND NEG LATER
0DB2	0333 †			
0DB2 7A	0334 NDRMP	MOV	A,D	IF WITHIN 8 BITS, 6D NDW
0DB3 B7	0335	DRA	A	
0DB4 C2 AC 0D	0336	JNZ	NDRM1	
0DB7 B3	0337	DRA	E	IF WITHIN 16, USE EHL
0DB8 C2 A1 0D	0338	JNZ	NRMEHL	
0DBB B4	0339	DRA	H	IF WITHIN 24, USE HL
0DBC C2 9B 0D	0340	JNZ	NRMHL	
0DBF B5	0341	DRA	L	IF JUST L, USE IT
0D90 C2 95 0D	0342	JNZ	NRML	
0D93 4A	0343	MOV	C,D	ELSE CDE=0
0D94 C9	0344	RET	.	
0D95	0345 †			
0D95 55	0346 NRML	MDV	D,L	LOO FOR 3 BYTES
0D96 06 1B	0347	MVI	B,24	ADJUST EXP BY 24 BITS
0D98 C3 A6 0D	0348	JMP	AJEXP	
0D9B EB	0349 NRML	XCHG	.	HLO FOR 3BYTES

```
0D9C 06 10      0350      MVI    B,16  ADJUST EXP 16
0D9E C3 A6 0D   0351      JMP    AJEXP
0DA1 53         0352 NRMEHL MOV    D,E   SHIFT EHL TO DEH
0DA2 5C         0353      MOV    E,H
0DA3 65         0354      MOV    H,L
0DA4 06 08     0355      MVI    B,B   ADJUST B BITS
0DA6 79         0356 AJEXP  MOV    A,C   EXP=EXP-B
0DA7 90         0357      SUB    B
0DAB 4F         0358      MOV    C,A   IF PROBLEM, THEN UNDER
0DA9 DA C3 0D   0359      JC    UNDERFLOW
0DAC           0360 *
0DAC           0361 * BIT BY BIT NORMALIZATION
0DAC           0362 *
0DAC 7A         0363 NORM1 MOV    A,D   AHL=DEH
0DAD 6C         0364      MOV    L,H
0DAE 63         0365      MOV    H,E
0DAF B7         0366 NCHK  ORA    A    SHIFT AHL TILL NORMED
0DB0 FA B9 0D   0367      JM    NRMFIN
0DB3 0D         0368 NCHK1 DCR    C    EXP<-EXP-1
0DB4 29         0369      DAD    H
0DB5 8F         0370      ADC    A
0DB6 F2 B3 0D   0371      JP    NCHK1
0DB9 57         0372 NRMFIN MOV    D,A   DE=AH
0DBA 5C         0373      MOV    E,H
0DBB 7D         0374      MOV    A,L   IF MSB(L)=1, ROUND OFF DE
0DBC 07         0375      RLC    .
0DBD DC A0 0C   0376      CC    ROND
0DC0 79         0377      MOV    A,C   IF EXP POSITIVE, OK
0DC1 87         0378      ORA    A
0DC2 F0         0379      RP    .
0DC3           0380 *
0DC3           0381 * ERRORS : UNDERFLOW AND OVERFLOW
0DC3           0382 *
0DC3           0383 UNDERFLOW EQU *
0DC3 0E 00     0384 RET0  MVI    C,0   RETURN CDE=0
0DC5 11 00 00   0385      LXI    D,0
0DC8 C9         0386      RET    .
0DC9           0387 *
0DC9 FE C0     0388 ERCHK CPI    0C0H  IF BETWEEN 0B0H AND 0BFH
0DC8 D2 C3 0D   0389      JNC   UNDERFLOW THEN UNDERFLOW, ELSE OVER
0DCE           0390 *
0DCE 0E 7F     0391 OVERFLOW MVI  C,7FH RETURN CDE=MAXIMUM
0DD0 11 FF FF   0392      LXI    D,-1
0DD3 C9         0393      RET    .
0DD4           0394 *
0DD4 7B         0395 STRIP MOV    A,B   REMOVE SIGNS FROM B
0DD5 E6 7F     0396      ANI    7FH
0DD7 47         0397      MOV    B,A
0DD8 79         0398      MOV    A,C
0DD9 E6 7F     0399      ANI    7FH
```

```

ODDB 4F          0400      MOV   C,A
ODDC C9          0401      RET   .
ODDD             0402 *
ODDD             0403 * FIX32:  FLT TO FIX CONVERSION
ODDD             0404 *
ODDD 79          0405 FIX32 MOV   A,C   IF NEGATIVE. INVERT
ODDE EE 80       0406      XRI   80H   RESULTS
ODE0 FA EA 0D    0407      JM    FIXPOS
ODE3 4F          0408      MOV   C,A
ODE4 CD EA 0D    0409      CALL  FIXPOS
ODE7 C3 BC 0E    0410      JMP   NEG32
ODEA             0411 *
ODEA E6 7F       0412 FIXPOS ANI   7FH   IF CDE<1, RETURN(0)
ODEC FE 41       0413      CPI   41H
ODEE DA 1B 0E    0414      JC    ZERDZ
ODF1 FE 60       0415      CPI   60H   IF >2**31, MAX IT
ODF3 D2 1F 0E    0416      JNC   MAXDH
ODF6             0417 *
ODF6 21 00 00    0418      LXI   H,0   ELSE SHIFT MANTISSA
ODF9 D6 50       0419      SUI   40H+16 IF 2**16, QUIT
ODFB CB          0420      RZ    .
ODFC EB          0421      XCHG .    DEHL=00XX, READY TO SHIFT
ODFD D2 0F 0E    0422      JNC   SHDH  IF EXP WAS 5I TO 5F, 6D
OE00 C6 10       0423      ADI   16   ELSE 41-4F, SHIFT THEN
OE02 CD 0F 0E    0424      CALL  SHDH  DIVIDE BY 2**16
OE05 EB          0425      XCHG
OE06 II 00 00    0426      LXI   D,0
OE09 C9          0427      RET   .
OE0A             0428 *
OE0A 29          0429 SHCAR DAD   H    SHIFT DE PART
OE0B 2C          0430      INR   L    AND PUT IN CARRY
OE0C EB          0431 DECRA XCHG .    SWAP BACK HL
OE0D 3D          0432      DCR   A    IF COUNT=0, QUIT
OE0E CB          0433      RZ    .
OE0F 29          0434 SHDH  DAD   H    SHIFT HL ONE BIT
OE10 EB          0435      XCHG .    IF CARRY, THEN
OE11 DA 0A 0E    0436      JC    SHCAR UPDATE DE WITH CARRY
OE14 29          0437      DAD   H    ELSE WITHOUT CARRY
OE15 C3 0C 0E    0438      JMP   DECRA
OE1B             0439 *
OE1B II 00 00    0440 ZERDZ LXI   D,0  DEHL=0
OE1B 2I 00 00    0441      LXI   H,0
OE1E C9          0442      RET   .
OE1F 1I FF 7F    0443 MAXDH LXI   D,7FFFH DEHL=MAXIMUM
OE22 2I FF FF    0444      LXI   H,-I
OE25 C9          0445      RET   .
OE26             0446 *
OE26             0447 * SQUARE (CDE)
OE26             0448 *
OE26 7A          0449 FSQUA MOV   A,D   CHECK FOR 0

```

0E27 B7	0450	DRA	A	
0E28 CB	0451	RZ	.	
0E29 41	0452	MDV	B,C	BHL=CDE
0E2A 62	0453	MOV	H,D	
0E2B 6B	0454	MDV	L,E	
0E2C C3 40 0C	0455	JMP	FMS33	
0E2F	0456 †			
0E2F 7A	0457 FSQRT	MDV	A,D	IF ZERD, QUIT
0E30 B7	0458	DRA	A	
0E31 CB	0459	RZ	.	
0E32 79	0460	MDV	A,C	IF DDD EXPDNENT, SHIFT
0E33 E6 01	0461	ANI	1	
0E35 C4 1D 0D	0462	CNZ	RITE1	
0E3B C5	0463	PUSH	B	SAVE EXPDNENT
0E39 CD 97 0E	0464	CALL	SQR2	DE=DE*1/2
0E3C C1	0465	PDP	B	
0E3D 79	0466	MOV	A,C	DIVIDE EXP BY 2
0E3E 0F	0467	RRC		
0E3F C6 20	0468	ADI	20H	IN EXCESS 64
0E41 4F	0469	MDV	C,A	
0E42 C9	0470	RET	.	
0E43	0471 †			
0E43	0472 †	16 X 16 MULTIFLY UNSIGNED. OPTIMIZED FOR F.P.		
0E43	0473 †	[AHL] = [HL] † [DE] TDP 3 BYTES		
0E43	0474 †			
0E43 97	0475 MU22F	SUB	A	IF E=0, DO SHDRT MULT
0E44 BB	0476	CMP	E	
0E45 CA 5B 0E	0477	JZ	SHDRD	
0E4B B5	0478	DRA	L	IF L=0, DO SHDRT WITH H
0E49 CA 5C 0E	0479	JZ	SHDRH	
0E4C E5	0480	PUSH	H	AHL= L*DE
0E4D CD 5D 0E	0481	CALL	MU21	
0E50 6C	0482	MOV	L,H	THROW AWAY LS BYTE
0E51 67	0483	MDV	H,A	SAVE UPPER BYTES
0E52 E3	0484	XTHL	.	SAVE EM, GET MS BYTE DF 1ST
0E53	0485 †			
0E53 7C	0486	MOV	A,H	AHL=MSB*DE
0E54 CD 5D 0E	0487	CALL	MU21	
0E57 D1	0488	POP	D	GRAB THE TWD STORED
0E58 19	0489	DAD	D	ADD PARTIAL RESULTS
0E59 BB	0490	ADC	B	FOR THREE BYTES (AHL)
0E5A C9	0491	RET	.	
0E5B	0492 †			
0E5B EB	0493 SHDRD	XCHG	.	SHORT MULT
0E5C 7C	0494 SHDRH	MOV	A,H	JUST MULT H*DE
0E5D	0495 †			
0E5D	0496 †	16 X 8 MULTIPLY UNSIGNED		
0E5D	0497 †	[AHL] <- A † [DE]		
0E5D	0498 †	TAKES 198 TD 297 CYCLES		
0E5D	0499 †			

0E5D 21 00 00	0500 MU21 LX1 H,0	ZERO RESULT REG
0E60 44	0501 MOV B,H	B<-0
0E61	0502 †	
0E61 87	0503 MULTX ADD A	SHIFT MSB TO CARRY
0E62 D2 67 0E	0504 JNC X2	
0E65 19	0505 DAD D	IF C=1, THEN ADD [DE]
0E66 88	0506 ADC B	IF OVERFLOW,BUMP MSBYTE
0E67 29	0507 X2 DAD H	SHIFT FOR NEXT TEST
0E68	0508 †	
0E68 8F	0509 ADC A	AND SO ON
0E69 D2 6E 0E	0510 JNC X4	
0E6C 19	0511 DAD D	
0E6D 88	0512 ADC B	
0E6E 29	0513 X4 DAD H	
0E6F	0514 †	
0E6F 8F	0515 ADC A	
0E70 D2 75 0E	0516 JNC X8	
0E73 19	0517 DAD D	
0E74 88	0518 ADC B	
0E75 29	0519 X8 DAD H	
0E76	0520 †	
0E76 8F	0521 ADC A	
0E77 D2 7C 0E	0522 JNC X10	
0E7A 19	0523 DAD D	
0E7B 88	0524 ADC B	
0E7C 29	0525 X10 DAD H	
0E7D	0526 †	
0E7D 8F	0527 ADC A	
0E7E D2 83 0E	0528 JNC X20	
0E81 19	0529 DAD D	
0E82 88	0530 ADC B	
0E83 29	0531 X20 DAD H	
0E84	0532 †	
0E84 8F	0533 ADC A	
0E85 D2 8A 0E	0534 JNC X40	
0E88 19	0535 DAD D	
0E89 88	0536 ADC B	
0E8A 29	0537 X40 DAD H	
0E8B	0538 †	
0E8B 8F	0539 ADC A	
0E8C D2 91 0E	0540 JNC X80	
0E8F 19	0541 DAD D	
0E90 88	0542 ADC B	
0E91 29	0543 X80 DAD H	
0E92	0544 †	
0E92 8F	0545 ADC A	
0E93 D0	0546 RNC	
0E94 19	0547 DAD D	
0E95 88	0548 ADC B	
0E96 C9	0549 RET	

```

0E97          0550 *
0E97          0551 * INTEGER SQUARE ROOT OF DE
0E97          0552 *  [\]
0E97 01 00 80 0553 SQR2 LXI  B,8000H GUESS=80, ROOT0=0
0E9A CD A7 0E 0554 SQRA1 CALL APPX CHECK APPROXIMATION
0E9D 78       0555     MOV  A,B   AND SHIFT APPX BIT
0E9E 0F       0556     RRC
0E9F 47       0557     MOV  B,A
0EA0 D2 9A 0E 0558     JNC  SQRA1
0EA3 51       0559     MOV  D,C   DE=RESULT
0EA4 1E 00    0560     MVI  E,0
0EA6 C9       0561     RET  .
0EA7          0562 *
0EA7 D5       0563 APPX PUSH  D   SAVE X
0EA8 78       0564     MOV  A,B   TRY NEW TEST BIT
0EA9 81       0565     ADD  C
0EAA 5F       0566     MOV  E,A
0EAB 16 00    0567     MVI  D,0
0EAD C5       0568     PUSH B   SAVE BC
0EAE CD 5D 0E 0569     CALL MU21 AHL=A*DE
0EB1 C1       0570     POP  B
0EB2 D1       0571     POP  D   COMPARE TO X
0EB3 78       0572     MOV  A,E   IF X <HL THEN TOO BIG
0EB4 95       0573     SUB  L
0EB5 7A       0574     MOV  A,D
0EB6 9C       0575     SBB  H
0EB7 DB       0576     RC   .
0EB8 78       0577     MOV  A,B   ELSE ADD TEST BIT TO C
0EB9 81       0578     ADD  C
0EBA 4F       0579     MOV  C,A
0EBB C9       0580     RET  .
0EBC          0581 *
0EBC CD CA 0E 0582 NEG32 CALL INV16 INVERT DEHL
0EBF EB       0583     XCHG
0EC0 CD CA 0E 0584     CALL INV16
0EC3 EB       0585     XCHG  .
0EC4 23       0586     INX  H   AND ADD 1
0ECS          0587 *
0ECS 7C       0588     MOV  A,H   IF HL=0, INCR DE
0EC6 85       0589     ORA  L
0EC7 C0       0590     RNZ  .
0EC8 13       0591     INX  D
0EC9 C9       0592     RET  .
0ECA          0593 *
0ECA 7C       0594 INV16 MOV  A,H   INVERT HL
0ECB 2F       0595     CMA
0ECC 67       0596     MOV  H,A
0ECD 7D       0597     MOV  A,L
0ECE 2F       0598     CMA
0ECF 6F       0599     MOV  L,A

```

OED0 C9	0600	RET	.	
OED1 00	V 0601	DB	256	END OF FFP

8000: C3 80 03 00 00 00 00 00 00 00 00 00 00 00 00
8010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8030: 00 00 00 00 E5 2A 04 10 E9 00 00 00 C3 C5 00 00
8040: C3 7C 00 C3 AF 00 C3 AA 00 C3 FE 00 C3 04 01 C3
8050: 0E 01 C3 68 01 C3 3A 01 C3 2A 02 C3 7C 01 C3 AF
8060: 01 C3 CF 01 C3 67 02 C3 71 02 C3 DC 01 C3 E2 01
8070: C3 01 02 C3 17 02 C3 B1 00 C3 C1 00 3E 4C D3 20
8080: 3E 4F D3 60 CD 43 02 3E 50 CD AA 00 97 32 03 10
8090: 32 08 10 21 00 00 CD 81 00 11 34 01 21 0E 10 0E
80A0: 06 1A 77 13 23 0D C2 A1 00 C9 32 00 10 30 C9 20
80B0: C9 F3 22 04 10 7C B5 3E 0B CA 8E 00 3E 09 30 F8
80C0: C9 32 08 10 C9 F5 C5 D5 0E 10 3E C0 30 20 07 DA
80D0: CD 00 00 3E 00 20 07 18 0D C2 D3 00 EB 22 01 10
80E0: EB 3E 01 32 03 10 3A 00 10 E6 80 F6 50 30 D1 C1
80F0: 3A 08 10 0F DA FA 00 F1 FB C9 F1 37 F8 C9 21 03
8100: 10 7E 87 C9 CD FE 00 CB 36 00 2A 01 10 C9 E5 C5
8110: 0E 10 3E C0 32 00 10 30 F3 CD 0E 10 3E 40 30 3E
8120: 80 29 1F 30 0D C2 1F 01 C1 E1 3E 40 32 00 10 F6
8130: 10 30 FB C9 20 07 D2 0E 10 C9 F5 B7 E6 FE 6F 26
8140: 30 3E 03 32 01 30 7E CD 61 01 7E CD 61 01 5E 2C
8150: 66 68 F1 FE 0D C0 29 29 29 29 6C D8 23 E6 0F 67
8160: C9 3E 02 3D C2 63 01 C9 DC 3A 02 3E 40 CD 58 02
8170: 3E 80 CD 58 02 CD 48 02 3A 4C 30 C9 C5 E6 03 47
8180: 7C E6 F0 80 07 07 07 07 F6 40 CD A3 01 2D 22 FF
8190: CF 2C 3A 09 10 CD C0 01 29 78 17 32 0D 10 22 08
81A0: 10 C1 C9 D5 32 07 10 11 61 70 CD 86 02 D1 C9 79
81B0: E6 07 32 0A 10 EE 07 D3 63 32 06 10 78 32 09 10
81C0: 87 87 87 E6 3F 4F 3A 07 10 E6 80 81 C3 A3 01 E6
81D0: 01 0F 47 3A 07 10 E6 3F 80 C3 A3 01 2A 00 F0 C3
81E0: E5 01 22 00 80 E5 2A 08 10 11 04 00 19 22 08 10
81F0: D2 FF 01 21 0D 10 34 3A 0A 10 3C 8E CC 01 02 E1
8200: C9 3A 09 10 CD 0A 02 C3 7C 01 0F C5 47 E6 80 67
8210: 2E 00 78 E6 03 C1 C9 3A 0D 10 C5 87 1F 47 2A 08
8220: 10 7C 1F 67 7D 1F 6F 78 C1 C9 DC 3A 02 FE 06 D0
8230: CD 78 02 7D CD 58 02 C3 48 02 F5 3A 06 10 F6 20
8240: C3 4E 02 3E FF CD 61 02 F5 3A 06 10 E6 DF D3 63
8250: 32 06 10 CD 71 02 F1 C9 2F CD 61 02 CD 71 02 3E
8260: FF 11 62 20 C3 86 02 06 C8 CD 71 02 05 C2 69 02
8270: C9 11 71 02 18 78 82 C2 74 02 C9 21 01 00 E6 0F
8280: C8 29 3D C3 80 02 F5 3E 01 87 C2 8E 02 53 F1 12
8290: C9 00 00 00 00 00 00 00 00 00 00 00 00 00 00
82A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
82B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
82C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
82D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
82E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
82F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8300: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

8310: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8330: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8340: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8350: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8360: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8380: 31 00 18 21 11 11 CD FA 03 CD 40 00 CD 67 00 CD
8390: 40 04 CD 40 05 97 32 20 10 21 00 00 CD FA 03 CD
83A0: 4C 00 C2 AB 03 CD C0 03 C3 9F 03 22 21 10 CD FA
83B0: 03 CD 43 05 D2 9F 03 CD 43 04 DC CB 03 C3 9F 03
83C0: 2A C9 03 22 23 10 C3 23 10 76 C9 21 20 10 34 C9
83D0: 11 00 10 0E 08 2E 00 CD E6 03 29 CA DF 03 2C 14
83E0: 7A 0B C2 D7 03 C9 1E 00 CD F1 03 C0 1C C2 E8 03
83F0: C9 1A 47 2F 12 1A 2F 12 88 C9 7D D3 00 7C D3 01
8400: C9 42 55 52 53 54 20 32 2D 31 2D 38 35 20 50 52
8410: 20 48 41 52 56 45 59 00 00 00 00 00 00 00 00
8420: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8430: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8440: C3 46 04 C3 4D 04 21 02 12 22 00 12 C9 7C E6 FC
8450: FE 8C 37 C0 E8 7A D6 8C CA 69 04 3D CA 6E 04 3D
8460: CA 73 04 3D CA 7C 04 37 C9 7B 32 00 12 C9 7B 32
8470: 01 12 C9 2A 00 12 73 23 22 00 12 C9 3A 02 12 FE
8480: AA C0 97 32 02 12 C3 03 12 00 00 00 00 00 00
8490: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
84A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
84B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
84C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
84D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
84E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
84F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8510: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8520: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8530: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8540: C3 46 05 C3 63 05 CD C0 08 97 CD C3 08 3E 0F CD
8550: 94 05 B7 CD 52 00 97 32 D5 10 21 00 00 CD 5B 00
8560: 21 45 83 7C E6 F0 FE 80 37 C0 7C D6 B0 FE 08 3F
8570: D8 E8 87 21 7E 05 CD 51 08 CD 56 08 78 E9 94 05
8580: C3 08 C6 08 9A 05 E0 05 2E 06 26 06 19 06 46 06
8590: 00 00 D9 05 E6 0F 32 C4 10 C9 5F 0F 0F 0F 0F E6
85A0: 07 47 32 C6 10 78 E6 07 4F 32 C7 10 CD 5E 00 21
85B0: 05 00 06 00 CD C8 05 3A C6 10 E6 07 67 3A C7 10
85C0: 6F 06 01 CD C8 05 87 C9 7D BC D8 78 0F 7C E5 C5
85D0: CD 58 00 C1 E1 24 C3 C8 05 32 D5 10 0F C3 52 00
85E0: 3A D5 10 0F 3F DC 52 00 CD E4 06 3A C5 10 CD CF
85F0: 06 CD 65 07 22 C0 10 D5 CD 4F 00 E1 22 C2 10 CD
8600: 4F 00 CD 70 00 21 0A 88 CD 6D 00 CD 70 00 CD 6A
8610: 00 11 0A B8 CD 58 08 37 C0 3E 01 32 C8 10 CD 1C
8620: 07 CD D6 10 97 C9 21 0C D8 CD 6D 00 97 C9 3A D5

8630: 10 0F D4 52 00 CD 9F 07 21 48 4F D2 41 06 21 4F
8640: 4E CD 4F 00 87 C9 3A C8 10 FE 01 CC 9F 07 CD 8A
8650: 06 3A C8 10 FE 02 C8 2A C9 10 3A C8 10 CD 58 00
8660: CD 49 00 C2 81 06 CD 6A 00 CD 4F 00 CD 73 00 E8
8670: 47 2A CC 10 CD 58 08 C2 60 06 3A CE 10 88 C2 60
8680: 06 97 21 00 00 CD 58 00 87 C9 3E B2 CD CF 06 3E
8690: A1 CD CF 06 3A C5 10 CD CF 06 2A C9 10 CD 4F 00
86A0: 2A CA 10 CD D8 06 CD 4F 00 2A CC 10 CD 4F 00 2A
86B0: CD 10 CD D8 06 CD 4F 00 CD C9 08 EB CD CC 08 4F
86C0: CD CF 06 79 87 C8 1A CD CF 06 13 0D C3 C3 06 6F
86D0: 26 00 C3 4F 00 C3 C8 05 06 04 97 29 17 05 C2 D8
86E0: 06 6C 67 C9 CD 08 07 41 CD CC 08 4F CD C9 08 E8
86F0: 21 D6 10 3A C5 10 FE 08 D2 00 07 06 02 C3 40 0A
8700: 78 B7 06 00 CA 40 0A 04 C3 40 0A 3A C4 10 32 C5
8710: 10 CD 1C 07 D0 21 C5 10 35 C3 11 07 3A C5 10 21
8720: 35 07 CD 51 08 4E 3A C5 10 FE 08 3F D0 CD CC 08
8730: 47 79 90 4F C9 C8 64 28 14 0A 04 02 01 14 0A 06
8740: 04 03 02 01 00 0A 00 14 00 32 00 64 00 C8 00 F4
8750: 01 E8 03 D0 07 A0 08 40 17 C1 26 21 3A 81 4D 42
8760: 74 84 E8 84 E8 3A C5 10 87 21 45 07 CD 51 08 CD
8770: 56 08 E5 CD CC 08 D1 CD 24 0C 16 00 5F CD 18 0C
8780: 21 CF 10 CD 03 0C 3A C7 10 21 C6 10 96 3C 57 1E
8790: 00 26 00 CD 18 0C 21 CF 10 CD 09 0C C3 18 0C 21
87A0: 0E F8 CD 6D 00 CD 70 00 11 0E F8 CD 16 08 D8 CD
87B0: 73 00 22 CC 10 32 CE 10 CD 70 00 CD 6A 00 11 0A
87C0: 88 CD 58 08 C2 D5 07 CD 73 00 22 C9 10 32 C8 10
87D0: 97 32 C8 10 C9 2A CC 10 3A CE 10 CD 58 00 CD CC
87E0: 08 4F CD 49 00 C0 41 CD 6A 00 11 0C D8 CD 58 08
87F0: CA 03 08 11 0E F8 CD 5B 08 CA 03 08 05 C2 E7 07
8800: C3 E2 07 79 90 F5 2A CC 10 3A CE 10 CD 58 00 F1
8810: C4 47 08 C3 C7 07 3E 03 32 D4 10 21 00 00 22 D2
8820: 10 CD 49 00 37 C0 D5 CD 6A 00 D1 CD 5B 08 C8 21
8830: D2 10 35 C2 26 08 23 35 C2 21 08 23 35 C2 21 08
8840: 3E 02 32 C8 10 37 C9 F5 CD 6A 00 F1 3D C2 47 08
8850: C9 85 6F D0 24 C9 7E 23 66 6F C9 7D 88 C0 7C 8A
8860: C9 7D D3 00 7C D3 01 C9 00 00 00 00 00 00 00
8870: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8880: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8890: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
88A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
88B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
88C0: C3 DD 08 C3 D2 08 C3 24 09 C3 F3 08 C3 1A 09 C3
88D0: 0D 09 E6 0F 32 30 10 3E FF 32 31 10 C9 21 33 10
88E0: 11 A4 09 0E 06 1A 13 77 23 0D C2 E5 08 2D 7D 32
88F0: 32 10 C9 3A 30 10 FE 0A 21 7C 09 DA 03 09 21 33
8900: 10 D6 0A 87 C8 CD 0D 09 23 3D C3 03 09 F5 7E FE
8910: F0 CA 18 09 23 C3 0E 09 F1 C9 CD F3 08 5D CD 0D
8920: 09 7D 93 C9 4F 3A 30 10 FE 0A D8 21 31 10 34 C2
8930: 38 09 CD F3 08 CD 60 09 CA 35 09 CD F3 08 CD 0D
8940: 09 79 F5 E5 3A 32 10 FE 78 D2 5C 09 5F 54 3C 32

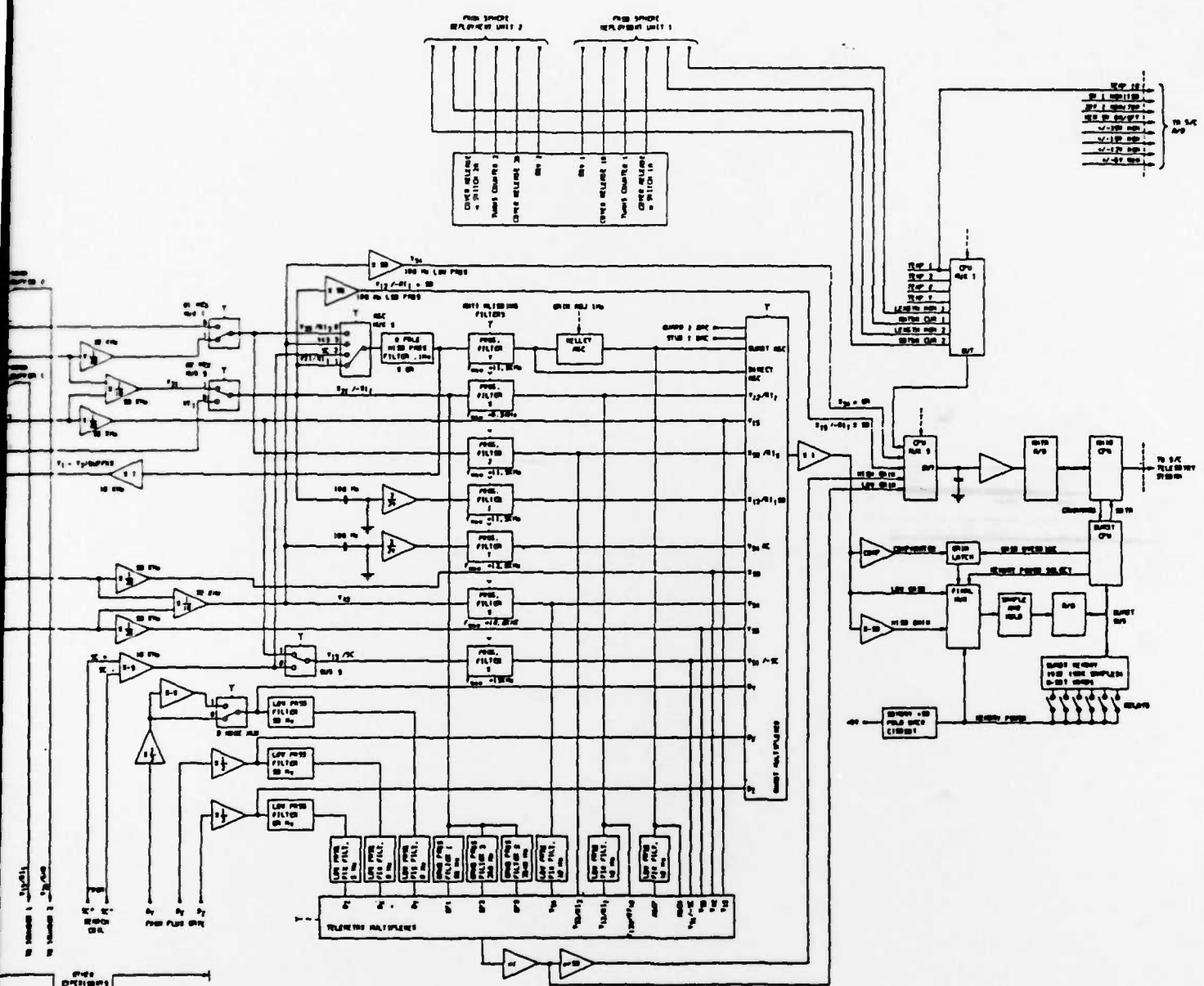
8950: 32 10 1A 13 12 18 18 78 BD D2 52 09 E1 F1 77 C9
8960: 7E FE F0 CA 7A 09 E5 23 7E 28 77 23 3A 32 10 BD
8970: C2 67 09 21 32 10 35 E1 97 C9 3C C9 0B F0 08 07
8980: F0 08 05 F0 0B 05 07 F0 0B 05 01 02 00 F0 00 01
8990: 02 03 04 05 06 00 07 08 09 0A 0B 0C 0D 0E 0F F0
89A0: F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 00 00 00 00 00
89B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
89C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
89D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
89E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
89F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8A00: 00 00 06 00 00 00 00 00 00 00 00 00 00 00 00
8A10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8A20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8A30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8A40: CD 18 0B EB CD 87 0A CD D4 0A 79 FE 03 D4 73 0A
8A50: CD A6 0A 23 79 FE 02 DA 60 0A CD CB 0A CD A6 0A
8A60: CD EE 0A 11 53 0B 21 94 10 0E 24 CD 27 0B 2A 90
8A70: 10 B7 C9 E5 C5 23 23 0D 0D CD A6 0A CD CB 0A 23
8A80: 0D C2 79 0A C1 E1 C9 3A A0 0A CD 1C 0B 78 CD 1C
8A90: 0B 11 A2 0A CD 0D 0B E5 2A 90 10 22 92 10 E1 C9
8AA0: 06 00 CD 30 0B FF 7E FE 3F 11 C3 0A CA 0D 0B 3A
8AB0: C0 0A CD 1C 0B 7E B7 CD 1C 0B 3A C2 0A C3 1C 0B
8AC0: 2A 00 B0 2A 90 90 2C 22 90 10 FF 11 D1 0A C3 0D
8AD0: 0B D8 7F FF 7B 87 C8 3D 11 E9 0A CA 0D 0B 11 E4
8AE0: 0A C3 0D 0B 69 CD 9F 10 FF 69 CD 94 10 FF 11 06
8AF0: 0B CD 0D 0B 3A 92 10 CD 1C 0B 3A 93 10 CD 1C 0B
8B00: 11 09 0B C3 0D 0B D2 FF FF C3 4B 0B FF 1A FE FF
8B10: C8 CD 1C 0B 13 C3 0D 0B 22 90 10 C9 E5 2A 90 10
8B20: 77 23 22 90 10 E1 C9 1A 77 13 23 0D C2 27 0B C9
8B30: 3E 01 CD 61 00 3E 03 32 01 30 21 00 00 22 90 10
8B40: 3E 01 CD 79 00 F8 3A 90 10 87 C9 97 CD 79 00 97
8B50: C3 61 00 FB 20 C8 D8 7F 00 00 D2 94 10 C9 D4 A7
8B60: 10 2D C2 9F 10 C9 F3 F5 06 20 20 A0 C2 A8 10 20
8B70: A0 CA B0 10 F1 F8 C9 00 00 00 00 00 00 00 00
8B80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8B90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8BA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8BB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8BC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8BD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8BE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8BF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8C00: C3 27 0C C3 2D 0C C3 33 0C C3 5E 0C C3 F2 0C C3
8C10: E8 0C C3 DC 0C C3 4B 0C C3 62 0D C3 DD 0D C3 26
8C20: 0E C3 2F 0E C3 5D 0E 4E 23 56 23 5E C9 71 23 72
8C30: 23 73 C9 7A 87 C8 46 23 7E 87 CA C3 0D 23 6E 67
8C40: 78 A9 F2 50 0C CD D4 0D CD 50 0C 79 EE 80 4F C9
8C50: 78 81 D6 40 FA C9 0D 4F CD 43 0E C3 AF 0D 7A B7
8C60: C8 46 23 7E B7 CA CE 0D 23 6E 67 78 A9 F2 79 0C

8C70: CD D4 0D CD 79 0C C3 48 0C 79 90 C6 40 FA C9 0D
8C80: 4F C5 7C 2F 47 7D 2F 4F 03 62 68 09 DA A9 0C E8
8C90: 3E 10 CD C2 0C 29 DA 9F 0C 09 DA 9F 0C C1 C9 C1
8CA0: 1C C0 14 C0 11 00 80 0C C9 3E 10 11 FF FF CD C2
8CB0: 0C C1 0C 37 7A 1F 57 78 1F 5F D0 C3 A0 0C 33 33
8CC0: 3D C8 29 DA D4 0C E8 29 E8 1C E5 09 DA 8E 0C E1
8CD0: 1D C3 C0 0C E8 29 E8 09 1C C3 C0 0C C5 D5 CD EB
8CE0: 0C 7A 87 CA E8 0C 79 07 D1 C1 C9 7E EE 80 47 C3
8CF0: F3 0C 46 23 7E 23 6E 67 97 8C C8 8A CA 5F 0D 79
8D00: 90 87 F2 0B 0D 78 41 4F E8 90 87 CA 15 0D 0F FE
8D10: 10 D0 CD 47 0D 78 A9 FA 25 0D 19 E8 D0 7A 1F 57
8D20: 7B 1F 5F 0C C9 78 95 7A 9C DA 3A 0D 57 78 95 5F
8D30: 21 00 00 82 C2 74 0D 0E 00 C9 7D 93 5F 7C 9A 57
8D40: 4B 21 00 00 C3 74 0D D6 08 DA 52 0D 6C 26 00 C8
8D50: D6 08 C5 47 97 29 8F 04 C2 55 0D C1 6C 67 C9 E8
8D60: 4B C9 7A 87 0E 60 F2 74 0D CD BC 0E CD 74 0D 79
8D70: F6 80 4F C9 79 87 F2 82 0D E6 7F 4F CD 82 0D C3
8D80: 4B 0C 7A 87 C2 AC 0D 83 C2 A1 0D B4 C2 98 0D B5
8D90: C2 95 0D 4A C9 55 06 18 C3 A6 0D E8 06 10 C3 A6
8DA0: 0D 53 5C 65 06 08 79 90 4F DA C3 0D 7A 6C 63 87
8DB0: FA 89 0D 0D 29 8F F2 83 0D 57 5C 7D 07 DC A0 0C
8DC0: 79 87 F0 0E 00 11 00 00 C9 FE C0 D2 C3 0D 0E 7F
8DD0: 11 FF FF C9 78 E6 7F 47 79 E6 7F 4F C9 79 EE 80
8DE0: FA EA 0D 4F CD EA 0D C3 8C 0E E6 7F FE 41 DA 18
8DF0: 0E FE 60 D2 1F 0E 21 00 00 D6 50 C8 E8 D2 0F 0E
8E00: C6 10 CD 0F 0E E8 11 00 00 C9 29 2C E8 3D C8 29
8E10: EB DA 0A 0E 29 C3 0C 0E 11 00 00 21 00 00 C9 11
8E20: FF 7F 21 FF FF C9 7A B7 C8 41 62 68 C3 40 0C 7A
8E30: 87 C8 79 E6 01 C4 1D 0D C5 CD 97 0E C1 79 0F C6
8E40: 20 4F C9 97 B2 CA 5B 0E 85 CA 5C 0E E5 CD 5D 9E
8E50: 6C 67 E3 7C CD 5D 0E D1 19 88 C9 E8 7C 21 00 00
8E60: 44 87 D2 67 0E 19 88 29 8F D2 6E 0E 19 88 29 8F
8E70: D2 75 0E 19 88 29 8F D2 7C 0E 19 88 29 8F D2 83
8E80: 0E 19 88 29 8F D2 8A 0E 19 88 29 8F D2 91 0E 19
8E90: 88 29 8F D0 19 88 C9 01 00 80 CD A7 0E 78 0F 47
8EA0: D2 9A 0E 51 1E 00 C9 D5 78 81 5F 16 00 C5 CD 5D
8EB0: 0E C1 D1 78 95 7A 9C D8 78 81 4F C9 CD CA 0E EB
8EC0: CD CA 0E E8 23 7C 85 C0 13 C9 7C 2F 67 7D 2F 6F
8ED0: C9 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8EE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8EF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8F80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

BF90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BFA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BF80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BFC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BFD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BFE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BFF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Appendix B.
Block Diagrams

REV	DESCRIPTION	DATE
1	REVISION FROM V-13, 000000000000	1/10/67
2	REVISION FROM V-13, 000000000000	1/10/67



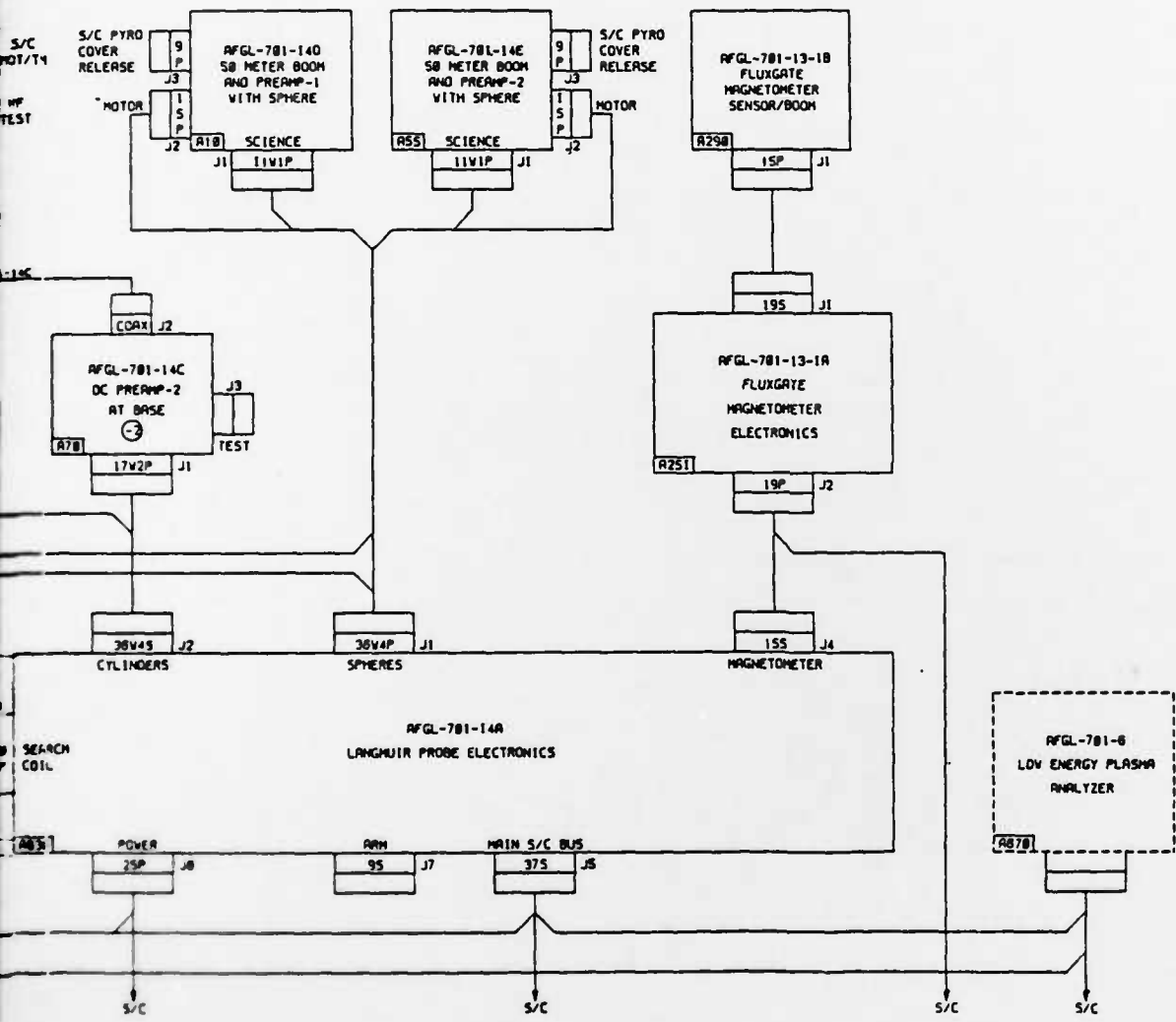
REV	DESCRIPTION	DATE
1	REVISION FROM V-13, 000000000000	1/10/67
2	REVISION FROM V-13, 000000000000	1/10/67

REV	DESCRIPTION	DATE
1	REVISION FROM V-13, 000000000000	1/10/67
2	REVISION FROM V-13, 000000000000	1/10/67

2 OF 2

CARES
 AFCL-781-14A
 LANGMUIR PROBE
 ANALOG BLOCK
 DIAGRAM
 C 1358

REV	DESCRIPTION	DATE	INIT
H	REDR.WN. ADD COAX T'S. DEL -14A J6, -15D J4, -15E J4.	14 FEB 85	J.O.
I	ADDED R291 AND S/C NO. 'S	01 JUL 85	J.O.



DESIGNED BY J. DILLING	DATE 2/8/85	CRRES AFGL-781-13.-14.-15 INTERCONNECTION BLOCK DIAGRAM	AIR FORCE GEOPHYSICS LABORATORY PHG WASHINGTON AIR FORCE BASE WASHINGTON, D.C. 20331	
CHECKED BY			SHEET	REV
USED ON			01	1
APPLICATION			01299	
FILED IN				

2 OF 2