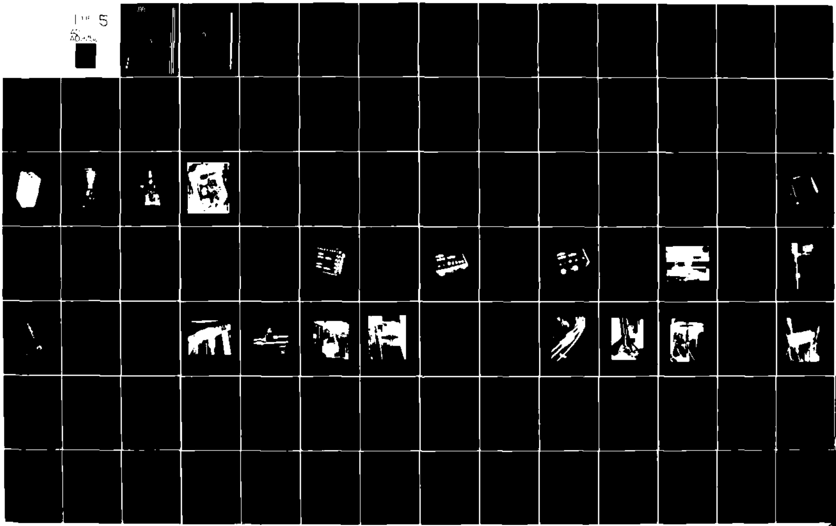
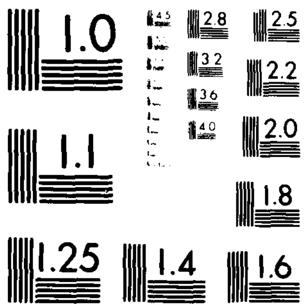
AD-A097 521 HONEYWELL INC HOPKINS MN DEFENSE SYSTEMS DIV F/G 19/6
AUTOMATED GUN LAYING SYSTEM FOR SELF-PROPELLED ARTILLERY WEAPON--ETC(U)
MAY 80 E E LEHTOLA, K A HERZING DAAA09-76-C-0284

UNCLASSIFIED NL





MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL III

(12)

DEFENSE SYSTEMS DIVISION

AUTOMATED GUN LAYING SYSTEM

FOR SELF-PROPELLED ARTILLERY WEAPONS

Honeywell Defense System Division
600 Second Street Northeast
Hopkins, Minnesota 55343

30 May 1980

Final Technical Report for Period
25 June 1976 - 30 June 1978
14 Sept 1978 - 30 Sept 1979

Prepared for

U.S. Army Research and
Development Command
Dover, New Jersey 07801

Honeywell

12

AUTOMATED GUN LAYING SYSTEM

FOR SELF-PROPELLED ARTILLERY WEAPONS

Honeywell Defense Systems Division
600 Second Street Northeast
Hopkins, Minnesota 55343

May 30, 1980

Final Technical Report for Period
25 June 1976 - 30 June 1978
14 Sept 1978 - 30 Sept 1979

Prepared by:

Project Engineer

Approved by:

Program Manager

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOV'T ACCESSION NUMBER: AD-A097 521

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (AND SUBTITLE): Automated Gun Laying System for Self-Propelled Artillery Weapons.

5. TYPE OF REPORT/PERIOD COVERED: Final Technical Report. 6 June 1976 — 30 September 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(S): W.E. Lehtola, K.A. Herzing

8. CONTRACT OR GRANT NUMBER(S): DAAA09-76-C-2084

9. PERFORMING ORGANIZATIONS NAME/ADDRESS: Honeywell Defense Systems Division, 600 Second Street N.E., Hopkins, Minnesota 55343

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME/ADDRESS

12. REPORT DATE: May 30, 1980

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME/ADDRESS (IF DIFFERENT FROM CONT. OFF.)

15. SECURITY CLASSIFICATION (OF THIS REPORT): Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (OF THIS REPORT)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

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)

Artillery	HELBAT
Automation	Howitzer
Fire Control	M109
Gun Laying	Self-Propelled

20. ABSTRACT (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)

Current procedures for laying field artillery can cause gross aiming errors and can cause time delays because these operations must be performed in sequence rather than simultaneously, and involve verbal transmission of numerical data.

To obtain more data on the benefits of automation, Contract DAA09-76-C-2084 was issued by ARMCOM (and later transferred to ARRADCOM) to design, develop, fabricate and install one prototype Automated Gun Laying System (AGLS) in a government furnished M109A1 self-propelled howitzer.

HD-168 REV 11/74

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 55 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

20. ABSTRACT (Cont.)

The primary objective of the AGLS Program was to develop a test bed to evaluate, on an incremental basis, various options for automation at the battery level. The system being developed would automate all of the on-carriage weapon positioning and fire control operations, while retaining insofar as practical, the existing weapon control and fire control equipment, and keeping the gun crew operations compatible with currently used procedures.

A system was fabricated, installed in an M109A1, and tested by the U.S. Army Field Artillery Board at Ft. Sill, Oklahoma over the period 20 March through 26 April 1978.

A contract add-on was issued on 9 September 1978 to integrate an advanced digital data communication system into AGLS. This system was designed to enhance the communication capabilities of the AGLS and to make the reconfigured vehicle compatible with the advanced fire direction center concepts employed for HELBAT VII. Additional capability in the form of an improved reference unit processor and interfaces to a projectile velocimeter, propellant temperature monitor and electronic fuze setter were also incorporated.

The system was designed, fabricated and installed in the AGLS equipped M109A1 howitzer and designated Howitzer Test Bed I. Field testing was performed by the Army during HELBAT VII (20 February 1979 - 30 March 1979) and by the Human Engineering Laboratory at Aberdeen Proving Ground (30 July 1979 - 25 August 1979). No data from these tests is included in this report; it was retained by the test agencies.

TABLE OF CONTENTS

(Continued Next Page)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

I. INTRODUCTION

Before an artillery weapon can engage a target, the weapon must be oriented on the correct gun position so that the artillery pieces can aim at the target. The current procedure to accomplish this is called "laying" the weapon. It requires considerable time and manpower and reduces the responsiveness of the weapon. The advantage of mobility inherent in the howitzer is diminished by this procedure and a new method to enable rapid deployment is desirable.

Current procedures for laying field artillery involve verbal transmission of data and an iterative sequence of manual procedures involving three or more gun crew members. Previous Human Engineer Laboratory Battalion Artillery Tests (HELBAT) have shown that these procedures can cause gross aiming errors due to transposition of digits and can cause time delays because these operations must be performed sequentially rather than simultaneously.

The Gun Alignment and Control System (GACS) offers one remedy to these problems; increasing the responsiveness of the howitzer by orienting all weapons on the gun position within seconds. Error-free displays of bearing and elevation are provided for members of the howitzer crew. The problem causing concern lies in the Fire Orders Data Section of the GACS, which consists of the Command Post Unit (CPU) and the Gun Unit (GU). Hardware breakdowns of these components have caused catastrophic system failure. The unavailability of engineering drawings has required that the components be returned to the manufacturer for repair for extended time periods and at considerable expense.

To obtain more data on the benefits of automation, Contract DAAA09-76-C-2084 was issued by ARMCOM (and later transferred to ARRADCOM) to design, develop, fabricate and install one prototype Automated Gun Laying System (AGLS) in a government furnished M109A1 self-propelled howitzer.

The primary objective of the AGLS Program was to develop a test bed to evaluate, on an incremental basis, various options for automation at the battery level.

The system being developed would automate all of the on-carriage weapon positioning and fire control operations, and improve weapon system effectiveness by reducing human errors and overall reaction time.

A secondary objective was to retain, insofar as practical, the existing weapon control and fire control equipment, and to keep the gun crew operations compatible with currently used procedures. This would retain a degree of commonality, thus facilitating the tasks of crew training and weapon maintenance, and also enabling direct cost/benefit comparisons when the various levels of automation are tested.

An additional task was added to the original workscope for the Automated Gun Laying System (AGLS) program. It provided for the replacement of several components of the GACS which had become unreliable and caused system failure. The primary goal of the added effort was to substitute components which would be more reliable and maintainable than the existing parts by providing a better design, complete with accurate engineering drawings.

Additional interfaces at the howitzer and FDC were fabricated which provided both radio and wire communication links. These were supplied to attain system compatibility for HELBAT VII testing. This howitzer was redesignated the Howitzer Test Bed #1.

II. SUMMARY

The Automated Gun Laying System developed under Contract DAAA09-76-C-2084 is a prototype or engineering model designed specifically to the requirements of a test bed system. The system configuration and characteristics were specified through a series of meetings and design reviews with the contractor's design personnel and the Contracting Officer's Technical Representatives (COTR). Following the design definition, the system components were fabricated and installed in the M109 at the contractor's facilities. The AGLS program consisted of the following phases:

1. Design Study

A detailed design study was conducted to establish the system configuration, predict performance characteristics, and to identify major error sources. This study was conducted during the first three months of the program. During this study, the system originally proposed by the contractor was further defined, utilizing the M109 component information provided by the COTR. Math models were developed to predict system performance and preliminary mechanical layout drawings were prepared to determine mechanical design feasibility of the proposed system. The design study validated the proposed method of leveling the M-15 quadrant and the M-145 mount, and indicated that the weapon could be driven by add-on stabilization system (AOS) hydraulic components.

The study was also directed to the operating characteristics of the Gun Alignment Control System (GACS), which had been proposed as the method of obtaining the azimuth reference for the AGLS. The COTR provided more data on the GACS characteristics, and the study determined that the GACS was suitable as the azimuth reference and as the input port for the fire control commands. However, questions were raised as to the light power output of the GACS reference unit XENON lamp, and the ability of a proposed Charge Coupled Device (CCD) solid state camera to detect the short pulses. Further testing when the GACS reference unit was delivered confirmed these doubts, and eventually led to the development of an IR tracker using a lateral-effect photodiode.

During the design study on the Automatic Gun Laying System (AGLS) program (Contract DAAA09-76-C-2084), it became apparent that effort beyond the scope of the original AGLS program would be required to interface with the Automated FDC planned for use on HELBAT VII. This effort was required to analyze, design, fabricate and test the electronic interfaces between the Fire Detection Center (FDC) PDP-11/34 computer and the AGLS onboard the howitzer. The system was required to support additional onboard data gathering from the AGLS, a projectile velocimeter and propellant temperature system. In addition to relaying gun orders from the FDC to the howitzer, the charge and fuze time were also to be transmitted and the latter relayed to a GFE electronic time fuze setter. The primary data link between the FDC and vehicle employed by AN/VRC-46 Military FM-VHF command radio set with backup furnished by a WD-1 and land line link. The effort was proposed to be accomplished by the contractor under Amend/Modification No. P00011 to the basic AGLS contract.

2. System Development and Fabrication

Following the design study, detail design of the AGLS components was initiated. One of the major tasks involved the modifications of the instrument servo components to provide servo drive capability. Government drawings were used as the basis for detailed layout drawings, from which wood mock-ups were fabricated. These mock-ups were then installed in the M109 and the fire control instruments were placed at the mechanical limit to determine worst case mechanical interferences. The interfering material was then removed from the wood mock-ups, and the layouts were modified to accommodate the available space. Several iterations were necessary before acceptable layouts were generated. Then, detail drawings were drawn, and used by design technicians to build the prototype hardware.

Installation of the AOS hydraulic components was accomplished using special tubes and brackets fabricated specifically for the M109. An Electronic Controller Unit (ECU) as used in the M60A1 was used except that azimuth and elevation modules characterized to the M109 were developed, fabricated and installed.

Further definition of the data display requirements indicated the desirability of three data display panels and a separate digital controller unit. A contract modification was negotiated to incorporate three display panels into the AGLS. The data display requirements, system operating mode selection, and various sequencing operations required to satisfy system performance requirements all pointed to the desirability of using a microprocessor to provide the digital data processing. Since a microprocessor was already being used by the contractor on another program, it was decided to utilize the same processor, a Motorola 6800, and to add the peripheral boards needed by the AGLS system.

The tracker, instrument controller and system power supply were all designed for the AGLS and involved initial design, breadboard test and prototype fabrication. Layout drawings were generated and details were developed sufficient to facilitate fabrication by design technicians. Functional tests were performed on each of the completed units prior to installation in the M109 to assure proper system performance.

The digital subsystem development and fabrication consisted primarily of design of the display panels and controller housing. To assist in the panel design, a human factors specialist reviewed the system requirements, participated in a contract meeting and live fire demonstration at Ft. Sill, and developed the panel arrangement for all three display panels. The remainder of the digital development task involved packaging of previously used circuits for the data interface circuit boards, assembly of previously designed processor boards, and writing of the system software.

After the system components were assembled and installed in the M109, preliminary functional and performance tests were conducted. During these tests, several changes in system sequence control and operating procedures became necessary. These were readily implemented by software changes, most of which could be implemented in less than one day elapsed time, by use of the microprocessor and the contractor's microprocessor development system.

A requirements analysis was performed on the following devices to establish subsystem compatibility and interface requirements for the HELBAT VII effort.

- GACS (CPU, GU and power supply - interface adapters)

- AN/VRC-46 Military radio

- Digital Equipment Corporation PDP11/34 minicomputer, its UNIBUS structure and DL11-E, DR11-L, DR11-M and DG11 I/O interfaces

- Lear Siegler MVR DR-810 velocimeter

- Electronic Fuze Setter and interfaces: HELBAT VI/XM587E2

- Propellant Temperature System

- M109 AGLS Power Conditioning System

Following completion of the requirements analysis, a detailed design was completed on the FDC Communication Processor and the Vehicle Communication Processor. This effort included the design of microprocessor based communication, control and interface hardware and the associated software formats, protocols and logical instruction sequences to interface between the hardware and the serial (radio and line) data link. In addition, a separate processor was designed to acquire and decode output from the GACS reference unit. This processor became part of the Vehicle Communication Subsystem.

After detailed design, the FDC Communication Processor, Vehicle Communication Processor and Reference Unit Processor were fabricated. The FDC Communication Processor was fabricated and packaged, using open card frame construction, to fit into a 6 inch relay rack chassis. Interface to the FDC computer was accomplished via a multi conductor cable and mating DEC connectors. The Vehicle Communication and Reference Unit Processors were fabricated and pacakged in a custom chassis designed to utilize the space envelope and to contain the communications and reference unit subsystems as well as the displays and controls required to supplement the AGLS Chief of Section panel.

3. System Integration & Test

Separate in-house and field test programs were conducted for the AGLS and HELBAT VII development tasks. This section presents the details of each of these separate activities in the chronological order they were performed.

a. AGLS

A test plan was prepared and submitted to ARRADCOM for approval prior to AGLS system acceptance testing. The acceptance tests were conducted at the Honeywell Proving Ground, with the assistance of the Contracting Officer's technical representatives. A test report was prepared and has been separately submitted to the Contracting Officer.

The completed AGLS, installed in the M109, was shipped to Ft. Sill, Oklahoma for tests by the U.S. Army Field Artillery Board. These tests consisted of twelve planned days of dry fire testing, and one day of live firing. A contractor representative was present to provide training of the U.S. Army Field Artillery test crews, and to assist in technical support during the test period.

The test plan was prepared by ARRADCOM, and the progress of the tests was monitored by both a contractor's representative and by a representative from ARRADCOM. The tests were conducted over the period 20 March through 26 April 1978. On 26 April, testing was concluded with the firing of twenty M107 projectiles using the M119 propelling charge (Zone 8). With the exception of two display boards being displayed out of their connectors, no AGLS components were affected by the firing shock. A separate report on the Ft. Sill tests will be issued by ARRADCOM.

b. HELBAT VII Communication

Laboratory debug and checkout of the AGLS Communications consisted of simulating the FDC computer, the gun laying subsystem and the serial line interfaces. The existing AGLS simulator was used to checkout that interface and appropriate thumb wheels, switches and displays provided stimulus for the remaining functions.

The reference unit processor was tested separately using the GACS Reference unit and GACS IR Receiver to provide stimulation.

The vehicle communication subsystem along with the available supporting subsystems were installed in the AGLS equipped M109A1 vehicle. Inputs to the system were provided through the vehicle communications processor serial line simulator. Test, diagnostic and simulation techniques developed during this testing were documented for subsequent use by the customer.

The vehicle was moved outside the Honeywell Defense Systems Laboratory for a total system checkout using both radio and land line serial data links. FDC computer inputs to the FDC comunication processor were simulated in this test. This test constituted the acceptance test.

Field testing was performed at Ft. Sill, Oklahoma as part of HELBAT VII during the period 20 February 1979 through 30 March 1979. In addition, a Human Engineering Laboratory evaluation test program was performed from 30 July 1979 to 25 August 1979. The specific test results from both these programs are to be published by HEL.

4. Documentation

Drawings were prepared and delivered. This report, with its appendix, is submitted as the final activity on this contract.

III. CONCLUSIONS AND RECOMMENDATIONS

The Automated Gun Laying System developed under contract DAAA09-76-C-2084 satisfied the previously stated objectives. The system automated the leveling, data offset, azimuth reference, elevation reference, and weapon azimuth and elevation functions of the M109A1 howitzer. Automation of various functions could be selected on an incremental basis, and all manual operations were retained. All automatic functions utilized the manual inputs (knobs) and feedback sensors (spirit levels or sight picture) to retain commonality, thus enabling the U.S. Army Field Artillery test crews to operate the AGLS/M109 in all levels of automation with a minimum of training. During contractor tests and demonstrations at Ft. Sill, it became apparent that the gun laying function could be accomplished by one crew member. In the case of momentary obscuration between the howitzer and the reference unit, it proved desirable to have a second crew member to assist in recognizing operational faults and to resume laying operations. The second crew member also provides verification of final acquisition as shown in the sight picture, and provides a safety back-up by visually checking the pantel counter and spirit levels.

The M109 operated for over five weeks at Ft. Sill (March-April 1978) with no failures, except for malfunctions precipitated by out-of-specification performance of the M109 electrical systems, and two printed circuit data display boards that became disengaged from their connectors as a result of gun fire shock. In addition, some operational problems were experienced in the use of the government furnished Gun Alignment Control System (GACS).

During the HELBAT VII testing several vehicle failures were experienced which seriously degraded the ability of the AGLS-COMM system to function effectively.

Problems experienced with TB-1 fall into three categories, namely: vehicle automotive, radio data communication, and gun laying. As a result of various <u>vehicle automotive problems</u> with the recoil, electrical, fuel cell and hydraulic hardware, TB-1 was not available for firing until March 18th; the beginning of the fifth week of the six week test program. While some limited communication system checkout and crew training were accomplished in the interim no complete exercise of the FDC/vehicle interaction in the fully automatic mode was performed.

The lack of a scheduled preparation and crew training phase also resulted in several unanticipated radio data communication problems. The most significant of these was the absence of integrated fully reliable voice communication between the FDC and TB-1. Without this voice radio capability the crew operation of the TB-1 system and range safety management were impacted. Through subsequent efforts (after March 18) of Helbat control and Automated FDC crew a voice radio link was established. Also contributing to the data communication problem were various protocol and timing differences between the TB-1 FDC interface and the FDC computer. These differences were, to a large extent, only revealed when the crew attempted to interact with the FDC in a firing scenario. Because of a lack of scheduled dry firing exercise with a dedicated FDC, many of the interface problems were not discovered until the vehicle was on line. When problems were discovered, software changes were made off-hours and were checked out by Honeywell with the excellent cooperation of the Automated FDC crew. Unfortunately, the military FDC controllers and crew were not present for those after hours training opportunities.

The digital data communication problems experienced with TB-1 were largely the result of the transmission scheme and protocols being different and more sophisticated than that used for the other three vehicles. In addition, the TB-1 system was fabricated, to specification, by Honeywell and was not as familiar to the Automated FDC technicians as the communication system that they designed and built for the remaining vehicles. We feel that the techniques used for the TB-1 system represent future self propelled howitzer digital communication configurations. Because the TB-1 system was different and required in-the-field adjustments to interface with the Helbat VII mission sequences, more FDC/crew dry fire training should have been scheduled, with the military crews of the FDC and TB-1 each using procedures identical to those used in live fire missions.

The third significant problem area, gun laying resulted mostly from the same lack of scheduled crew training and preparation. Significant software changes were accomplished in the field as a result of the crews interface with the TB-1 digital data system and displays. This man-machine interface input was most valuable to our understanding of automated fire control design, but unfortunately was not revealed until the system was on line and scheduled for firing. As an example, it was only after a fire order has been sent to TB-1 at the start of a live fire mission that FDC stated that they could not process a NORMAL angle as had been designed into the Reference Unit Processor section of the AGLS communication unit. Fortunately, the processor could be operated in the Distant Aiming Point mode to provide 3200 based azimuth data back to the FDC. Honeywell design personnel then modified the AGLS vehicle software to provide 3200 based commands and feedback, thus satisfying the belatedly recognized needs of the FDC and gun crew. Other gun laying related problems, involving the GACS Reference Unit alignment, laying the battery and fire order/check fire sequencing also were impacted by the lack of scheduled opportunity for adequate crew training with the dedicated FDC.

The Helbat VII test program was a revealing experience for Honeywell and contributed significantly to the maturation of the TB-1 system. In spite of the problems experienced in the aforementioned areas, very encouraging results were obtained. The crew's acceptance of the system (once they had adequate training) contributed to the excellent results achieved during the last week of the program. While fully automated operation was not achieved on all missions, the ability of the TB-1 system to function reliably in degraded modes was very encouraging. The digital data transmission system consistently transferred valid gun order data into vehicle in spite of severe radio skip interference and conflicting use of assigned radio frequencies; the simplex radio data link reliability was proven. In addition, the flexibility of the microprocessor approach to the onboard fire control scheme was amply demonstrated in that six significant changes to system operational software were implemented in the field.

Some conclusions/recommendations that result from our observations of the system operating in a "field" environment follow:

a. In the fully automatic mode of operation, the gunner and assistant gunner are not needed. Their tasks are essentially taken over by the chief of section, who operates the power, servo, weapon, load, and reference unit (RU) search switches to operate the AGLS. Thus, the chief of section has been transferred from a supervisory role in the present M109 to a single operator role in the AGLS. In addition, the chief of section was required to monitor the reference angle from the GACS, to mentally test for reasonableness, and to initiate a recovery plan if erroneous commands were handed off from the GACS to the AGLS. All of these new tasks represent a significant increase in the chief of sections work load.

b. It may be desirable to implement certain or all of the features of the Automated Gun Laying System into either a program to retrofit M109's or to design a new self-propelled howitzer. If this is to be accomplished, and an engineering development program is initiated, the following improvements to the AGLS should definitely be considered:

1. Instrument Servos

The M109 fire control instruments (quadrant, pantel, and mount) should be redesigned to incorporate the AGLS features into these instruments to obtain integral assemblies.

2. Instrument Controller Unit

An investigation should be directed toward the feasibility of a common controller for the five servo channels. This controller should be designed as a functionally complete, plug-in assembly, to satisfy the Reliability, Availability, and Maintainability (RAM) requirements as stated by the Ft. Sill maintenance evaluation.

3. System Power Supply

Operating power consumption measurements should be conducted to determine the maximum power requirements of the AGLS instrument servos and digital components. Test data thus obtained may permit reduction of the peak power capability, and, therefore, the physical size of the system power supply. Thermal characteristics should also be measured, to determine the feasibility of reducing the internal heat sink structure, and thus reduce the power supply size and weight. Plug-in assemblies, error monitor circuits, and test points should be implemented to enhance maintainability.

4. Digital Controller Unit

After system software has been finalized, the digital controller unit should be repartitioned to yield the minimum necessary digital system. Functionally complete assemblies should be utilized, with a minimum of inter-board connecting harnesses. Second or third generation microcomputer chip sets would permit a reduced number of components and interconnections, with attendant reductions of power consumption and enhanced system reliability.

c. Testing downtime could be minimized and more representative results obtained if more attention was directed in certain areas. These areas include:

1) Vehicular and Equipment "Shake-Down" Prior to Test -- More than one third of the test period was consumed by repair operations.

2) Training and Orientation Exercises -- Much more representative data would be available if the gun crew had a thorough understanding of the system operation. More importantly, the safety of the test program could have been improved if pretest training had been run to identify protocol and communication problems.

3) Testing Procedure -- The advantages of the system would be obvious if the test procedure could have included scenarios designed to depict the level of equipment sophistication.

IV. SYSTEM DESCRIPTION

The AGLS consists of six major subsystems as follows:

- o Fire Control Instrument Servo Subsystem
- o Digital Control Subsystem
- o Gun Alignment Control System
- o Infrared Receiver
- o Weapon Control System
- o System Power Supply

The block diagram showing the major interfaces between subsystems and components is provided in Figure 1. The cable connections between the system components are provided in Appendix A.

The configuration and basic operation of each of the AGLS subsystems is described in the following subsections.

A. Fire Control Instrument Servo Subsystem

The instrument servo subsystem consists of an instrument controller unit (see Figure 2) and the M-109 fire control instruments (M-117 telescope, M-145 mount and M-15 quadrant). The fire control instruments have been modified to provide automatic operation of the basic fire control functions in the AGLS/M-109. The modification includes the addition of electric drive motors, gears, and sensors, which have been attached to the fire control instruments. The modified fire control instruments are shown in Figures 3, 4 and 5. All existing features such as knobs, level vials and mechanical counters have been retained.

The fire control instrument servo subsystem consists of five separate servo channels as follows:

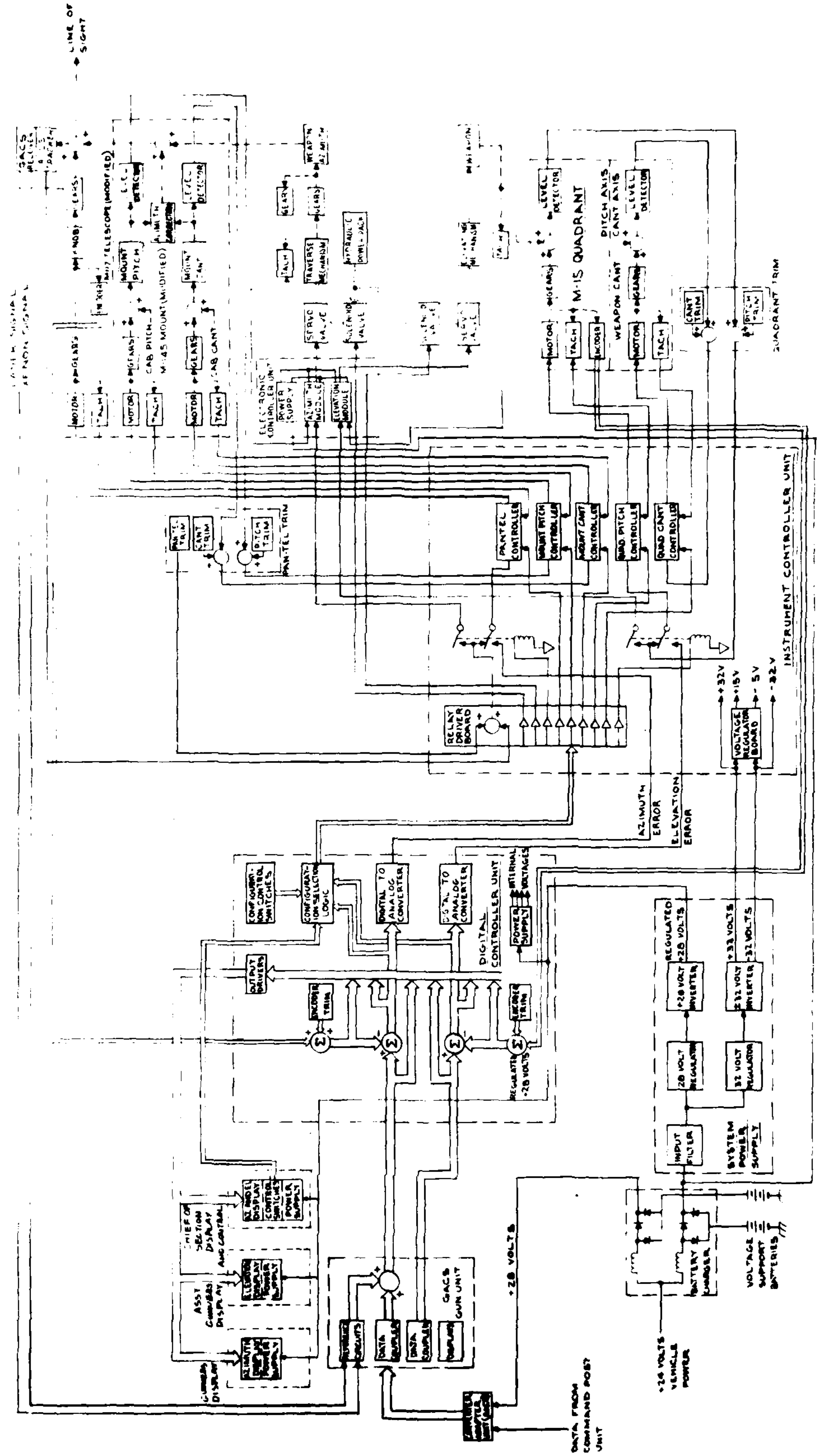


Figure 1. Block Diagram, Automatic Gun Laying System.

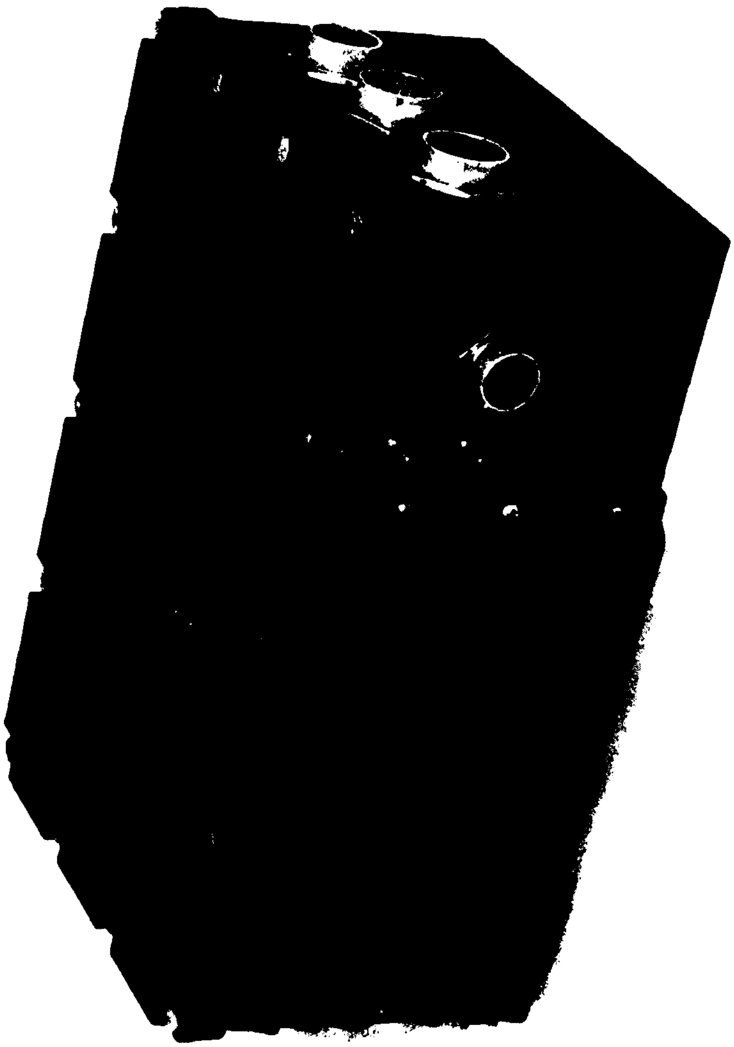


Figure 2. Instrument Controller Unit

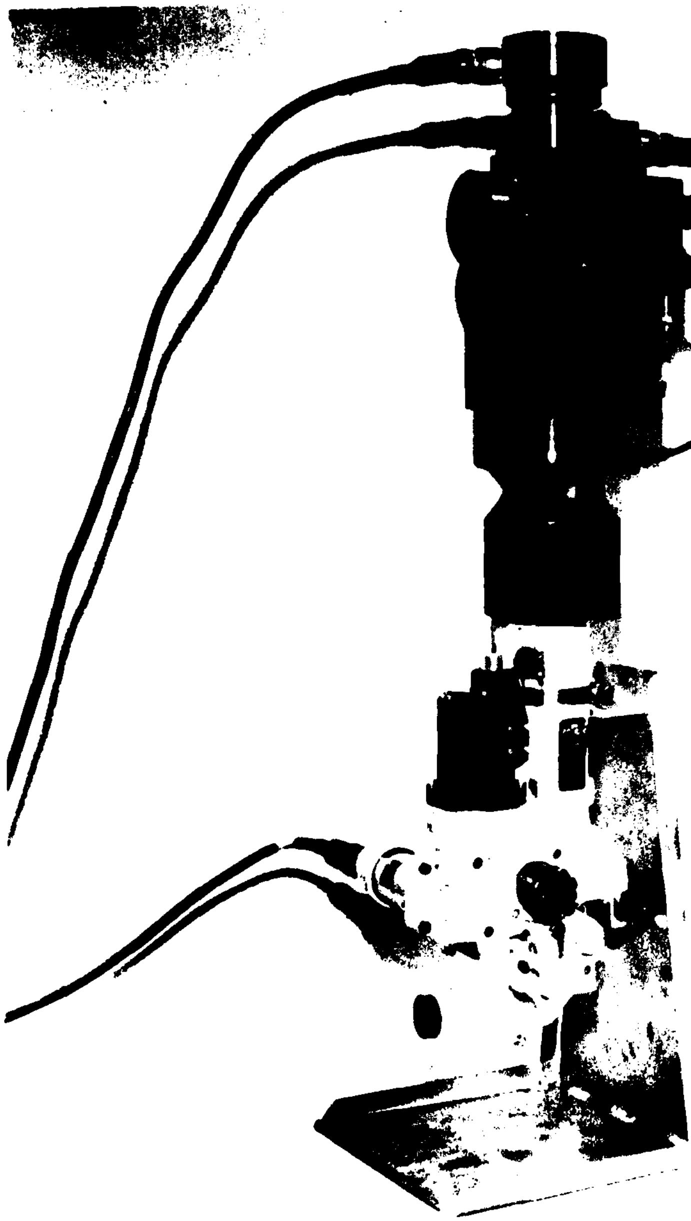


Figure 3. Modified M-117 Telescope

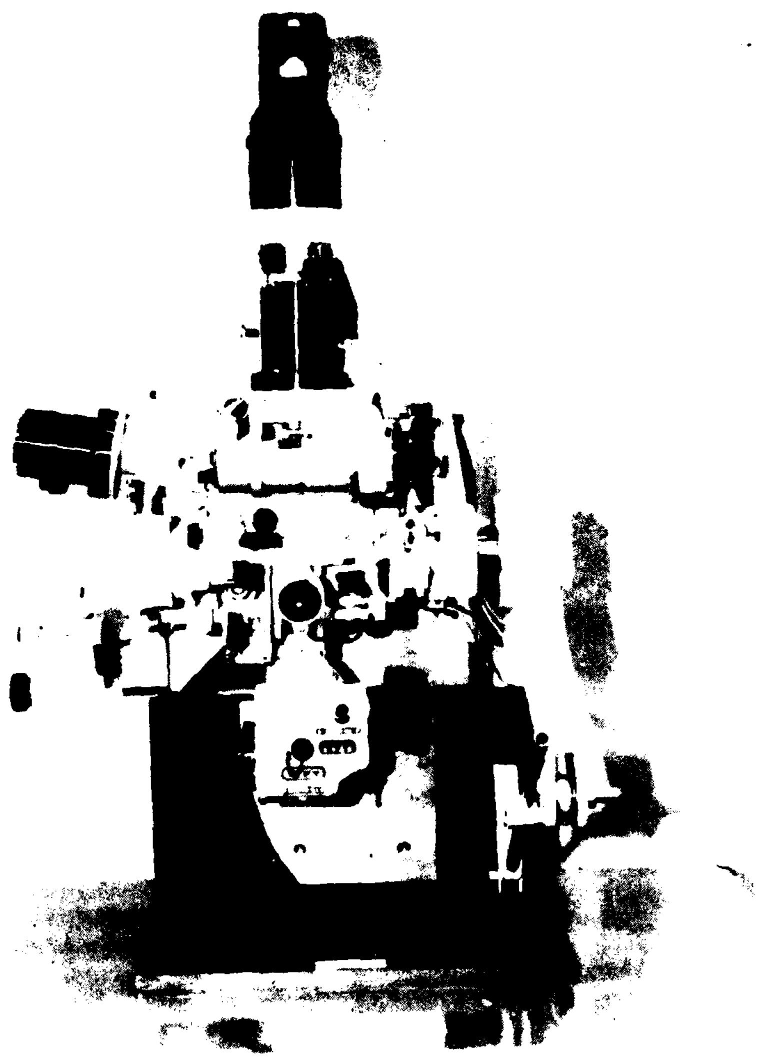


Figure 4. Modified M-145 Mount with Telescope

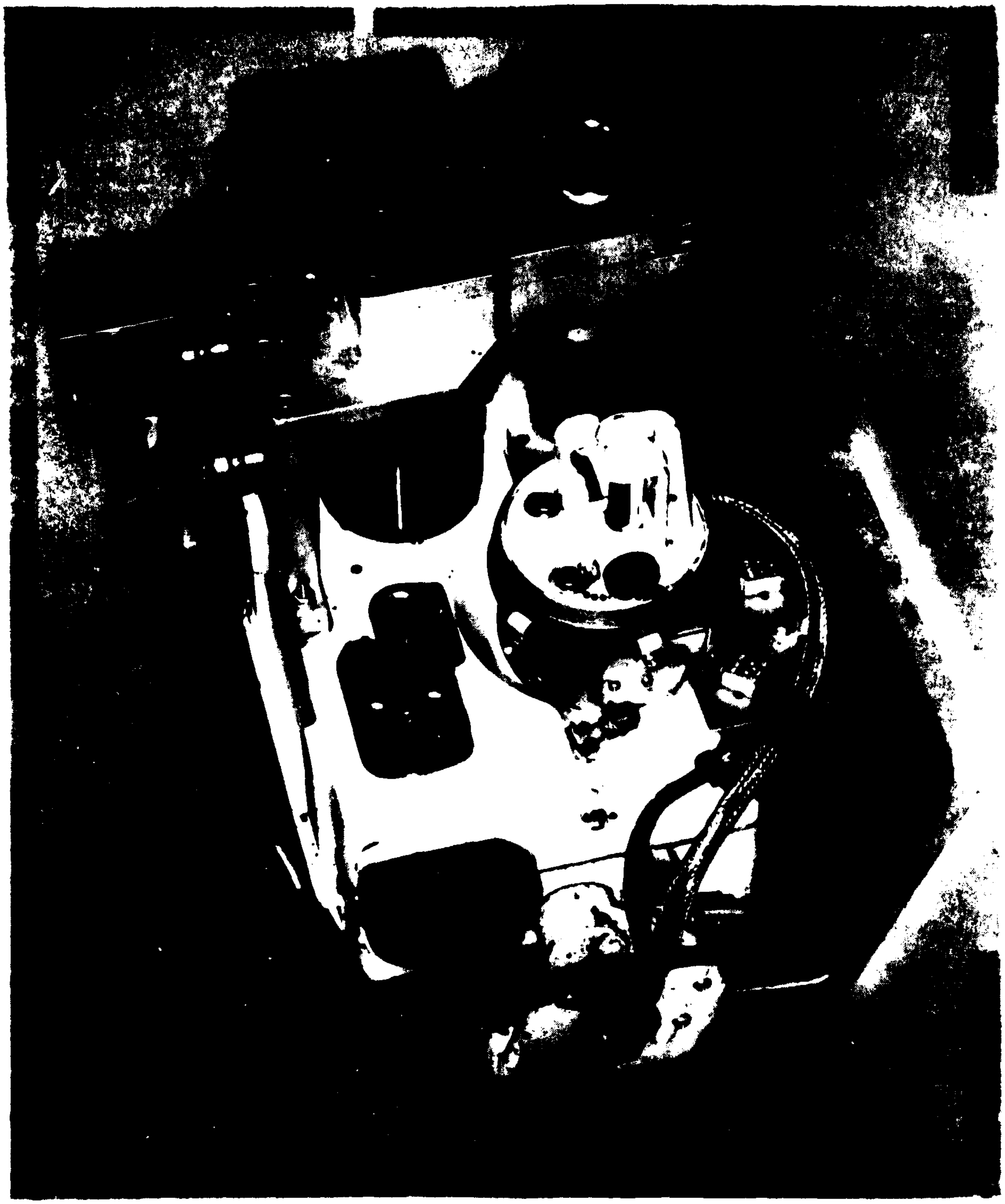


Figure 5. Modified M-15 Quadrant

1. M-15 quadrant cant
2. M-15 quadrant pitch
3. M-145 mount cant
4. M-145 mount pitch
5. M-117 telescope azimuth

Each of the servo channels consist of an electric drive motor, an amplifier, and one or more output sensors. All servo channels are similar in operation and are described in the following paragraphs.

1. Quadrant Cant Servo

The cant axis of the M-15 quadrant can be leveled by the quadrant cant servo which is shown in the block diagram of Figure 6. The servo consists of an integral motor/tachometer, coupled through precision gears to the cross level knob on the quadrant. A sensor mounted on the level vial platform detects an out-of-level condition and generates a positive or negative signal which is applied to the controller amplifier contained in the instrument controller unit. The amplifier processes the signal and generates an electric current to provide power to the servo motor, which then rotates the cross-level knob to bring the quadrant back to a level position.

The tachometer section of the motor/tachometer unit provides a direct current signal proportional to the rotating speed of the servo motor. This rate signal is used to control the maximum speed of the servo, and to provide a prediction signal to more accurately control the motor rotation. Since the tachometer is closely coupled to the motor, it is not influenced by the backlash of the quadrant mechanism, and will provide an accurate indication of servo motor motion. The same type of motor is used in one of two different housings for each of the five instrument servos.

While the tachometer provides a rate signal when the motor is rotating, the final or null position is determined by the signal from the level sensor. The level sensor is an accelerometer, which senses local gravity and generates a positive or negative signal proportional to the angle of the accelerometer with respect to

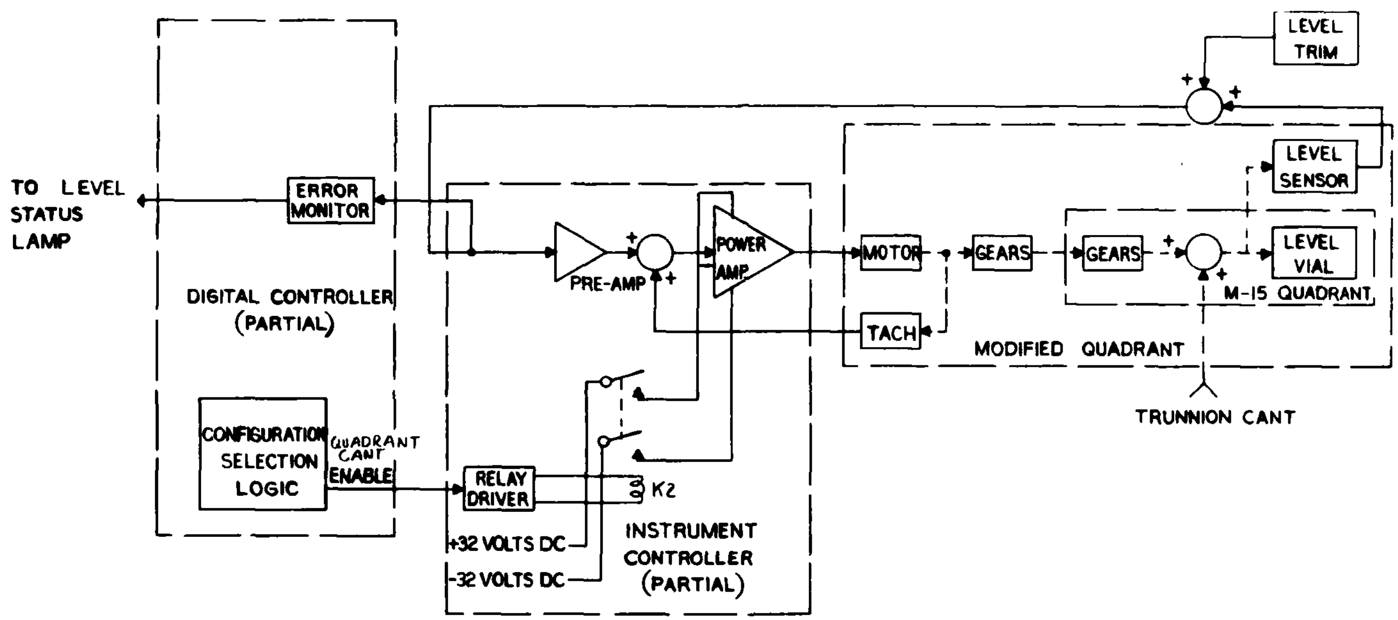


Figure 6. Block Diagram, Quadrant Cant Servo

level. For an ideal sensor with no null error, the level sensor output will be nulled when the level sensor is level.

The null position of the quadrant cant servo can be adjusted by means of the cant knob on the quadrant trim unit. This control generates a positive or negative signal which is added to the level sensor signal before it is supplied to the controller amplifier.

Trimming the level sensor permits more precise leveling of the quadrant by compensating for the changes in null signal of the level sensor. Adjustment of the trim is accomplished by observing the level vial while adjusting the cant trim knob, with the quadrant cant leveling servo engaged. The cant trim knob is then rotated clockwise or counterclockwise until the bubble is centered in the level vial.

2. Quadrant Pitch Servo

The pitch axis of the M-15 quadrant is shown in the block diagram of Figure 7. The servo consists of a motor/tachometer, gears, level sensor, and controller amplifier similar to those in the cant axis and can be controlled in either of two modes.

In the level mode, the servo functions exactly as described in the preceding discussion of the quadrant cant axis. The quadrant pitch level position can be adjusted by using the pitch trim control knob on the quadrant trim unit.

A digital encoder has been added to the quadrant pitch axis to measure the pitch angle of the level vial platform. This encoder permits operation of the quadrant pitch servo in the automatic offset mode. In this mode, the quadrant level vial platform (and mechanical counter) can be automatically driven to a commanded position, thus displacing the pitch level vial and the level sensor. This mode is used in the automatic offset configuration and also in the fully automatic elevation configuration.

The encoder is geared to the pitch input knob, and utilizes the internal mechanism of the M-15 quadrant to couple the encoder shaft to the level vial

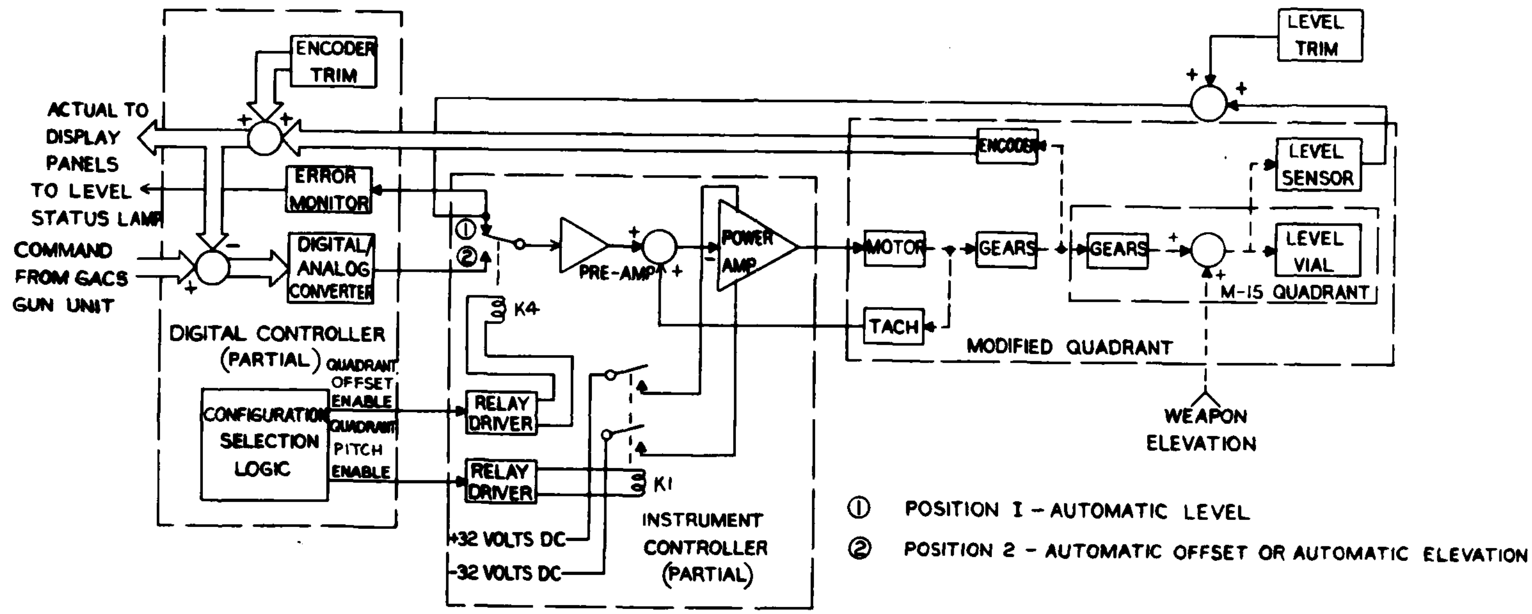


Figure 7. Block Diagram, Quadrant Pitch Servo

platform. The encoder consists of a high resolution section which resolves the knob position to the nearest 0.1 mil, and a low resolution section which counts the number of turns of the knob. The encoder thus measures actual quadrant pitch to the nearest 0.1 mil for the full range of 0 to 6399.9 mils. Since the quadrant range is limited to from negative 228 mils to postive 1383 mils, the encoder will read a negative angle Ø as (6400-Ø).

The output of the encoder consists of 19 lines of parallel digital information. Each line has either a 5.0 volt output or a zero output. The output, in binary coded decimal form, is transmitted to the digital controller unit by a separate wiring harness. The digital controller unit accepts the encoder data, the commanded data from the GACS gun unit, and the encoder trim data. It then subtracts the actual data from the commanded data to generate a correction digital signal. This digital signal is converted to a positive or negative direct current signal, and applied to the signal selector relay in the instrument controller unit. The signal selector, on command from the digital controller unit, will connect the position error signal derived from the encoder, and disconnect the level sensor signal. The signal is then applied to the quadrant pitch amplifier, to drive the pitch servo motor. This action will continue until the error signal achieves a null, indicating that the encoder output is equal to the commanded input. The quadrant has thus been driven, or offset, to a commanded position by the quadrant pitch servo.

3. Telescope Mount Cant Servo

The cant axis of the M-145 mount can be driven to level by the mount cant servo, which is identical in block diagram form to the quadrant cant servo as shown in Figure 6. The motor/tachometer is coupled through an attached drive mechanism to the cant correction knob. A level sensor mounted to measure telescope cant generates a positive or negative signal in response to the cant position of the telescope mounting seat. This signal is applied to the mount cant amplifier in the instrument controller unit, and the amplifier provides a drive current to the mount cant servo motor to drive the mount to a level condition. As in the two quadrant level axes, the mount cant level null position can be adjusted to a precise level position by use of the cant control knob on the azimuth trim unit.

4. Telescope Mount Pitch Servo

The pitch axis of the M-145 mount can be driven to level by the mount pitch servo. This servo consists of a motor/tachometer, drive mechanism, controller amplifier, and a level sensor located to measure telescope mounting seat pitch attitude. A trim control knob located on the azimuth trim unit is provided to adjust the null position for precise level.

5. Telescope Azimuth Servo

The azimuth line-of-sight of the M-117 panoramic telescope can be deflected by the telescope azimuth servo, shown in the block diagram of Figure 8. Drive is provided by a motor/tachometer coupled through gears to the azimuth knob shaft. A digital encoder is also geared to this shaft, and adjusted to measure the telescope deflection, as displayed in the azimuth counter. A controller amplifier in the instrument controller unit provides power to drive the telescope azimuth motor.

The telescope head has been modified to accommodate three added components; the GACS infrared receiver, the AGLS tracker, and a slip ring assembly. The GACS receiver will be described in a later section. The AGLS tracker is a passive device which detects the XENON lamp output from the GACS reference unit, and generates a positive or negative direct current signal proportional to the deflection of the reference unit from the telescope line-of-sight. The slip ring assembly is used to transfer the GACS receiver and the AGLS tracker signals from the rotating telescope head through a wiring harness to the telescope trim unit, and then to the instrument controller unit.

The telescope azimuth servo can be operated in two modes; Automatic Offset and Reference Unit Acquisition. Selection of mode is accomplished by program control and by the chief of section controls. In the Automatic Offset mode, the encoder output and the azimuth commanded deflection from the GACS gun unit are accepted by the digital controller unit, which calculates the digital difference signal. The digital controller unit generates a converted positive or negative azimuth error signal which is connected by the error signal selector to the telescope azimuth controller amplifier. The amplifier output current is then applied to

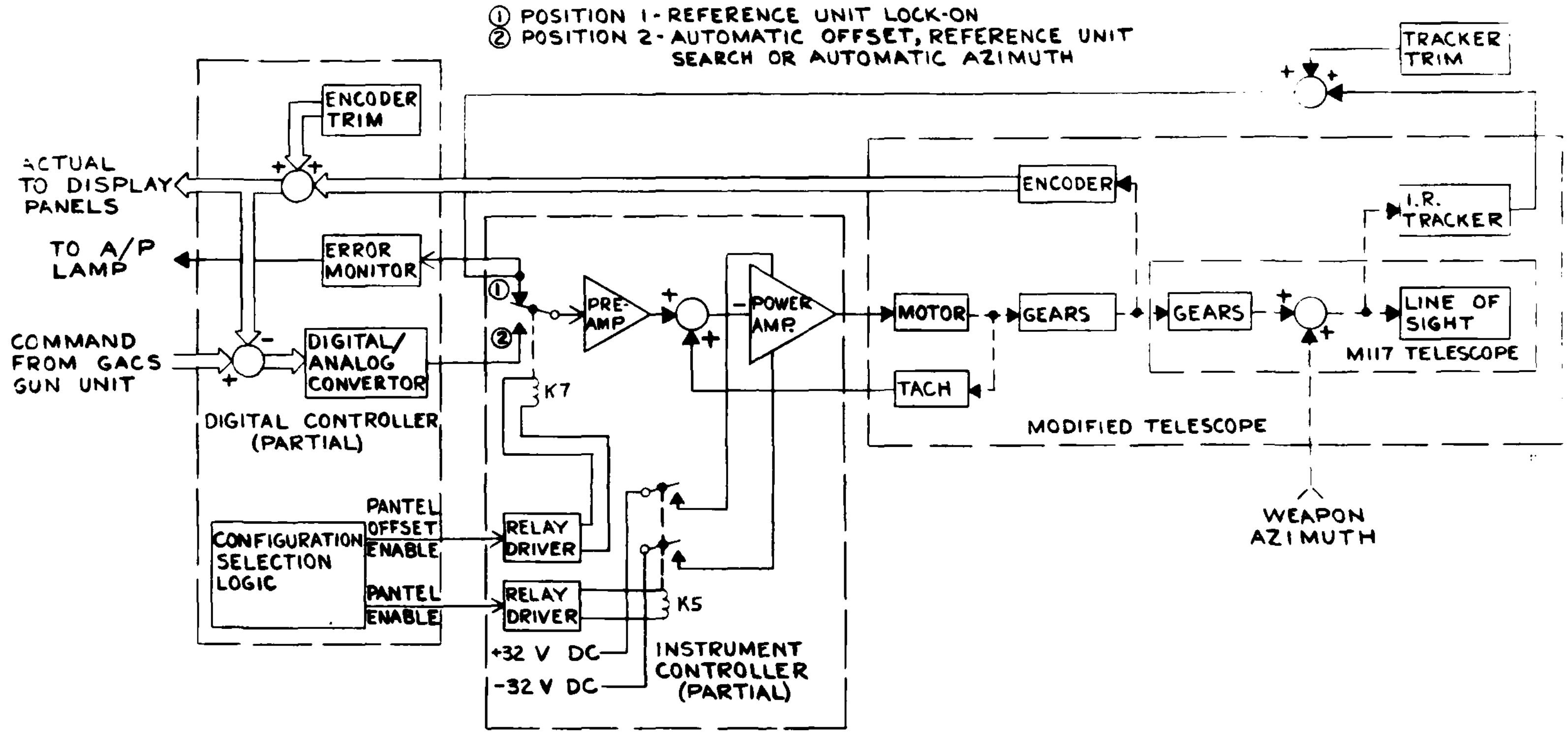


Figure 8. Block Diagram, Telescope Azimuth Servo

the telescope azimuth motor to drive the azimuth knob until the telescope deflection, as displayed in the azimuth counter, is equal to the commanded azimuth value. This mode is in principle, exactly like the automatic offset of the quadrant pitch axis.

In the Reference Acquisition mode, the telescope azimuth axis is commanded by the error signal from the tracker mounted on the telescope head. The position error signal from the tracker is applied through the error signal selector to the azimuth controller amplifier. The amplifier output drives the azimuth motor in a direction to reduce the error, until the tracker output achieves a null, thus indicating that the line-of-sight is in alignment with the reference unit. A trim control located on the azimuth trim unit is provided to adjust the final null to center the line-of-sight exactly on the reference unit.

If, prior to servo engagement, the telescope is positioned such that the reference unit is within the tracker field of view of plus or minus 100 mils, the telescope will automatically lock-on to the reference unit when the servo switch is activated. However, if the reference unit is outside the tracker field of view, the telescope servo must be commanded to acquire the reference unit. This command is provided as a steady positive or negative command from the digital controller unit and is initiated by the Reference Unit (RU) search control on the Chief of Section Panel. The digital controller unit also provides an enable signal to energize the servo and a signal select signal to activate the error signal selector to connect the command signal to the azimuth controller amplifier. The RU search command causes the telescope to drive at constant rate until the reference unit comes into the tracker field of view. As the tracker senses the reference unit, it generates a digital signal which is recognized by the digital controller unit. The digital controller unit then transfers control to the tracker by removing the signal select enable signal, and the tracker will then cause the telescope to lock onto the reference unit by the procedure described previously.

B. Digital Control Subsystem

The digital control subsystem serves as the interface between the gun crew, the fire direction center, and the servo control subsystems of the AGLS. The digital control subsystem consists of the following assemblies:

1. Digital Controller Unit
2. Chief of Section Panel
3. Gunner's Display Panel
4. Assistant Gunner's Display Panel

Each of the above assemblies is described below.

1. Digital Controller Unit

The AGLS digital controller unit (DCU) provides the system logic and control necessary to perform the following functions:

- o Receive commanded azimuth and elevation data from the GACS gun unit
- o Monitor weapon azimuth and elevation data from the panoramic telescope and M-15 quadrant
- o Calculate position errors and generate correction signals to drive the fire control instrument servos
- o Generate enable signals for the analog servos
- o Provide data to the display panels
- o Monitor analog sensor null signals

The digital controller unit is shown in Figure 9.

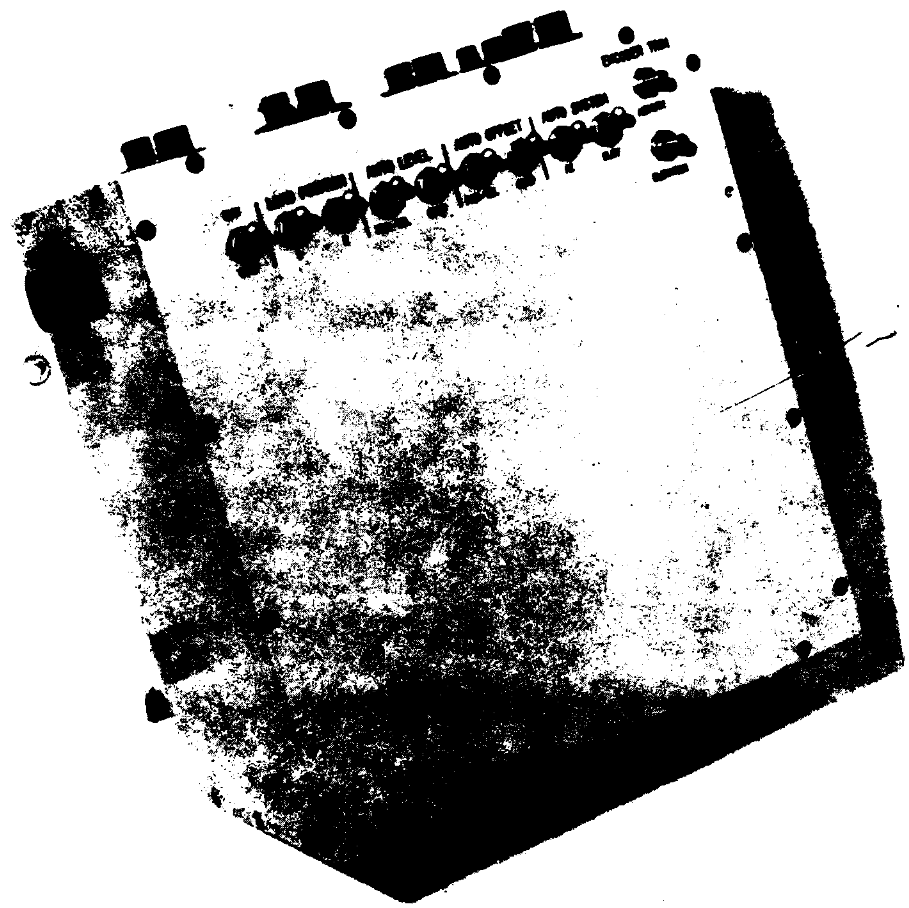


Figure 9. Digital Controller Unit

The digital controller unit processes signals from and to three separate systems: GACS, AGLS analog and AGLS digital. Since each system has its own separate ground point, ground isolation must be provided beweeen systems to prevent ground currents and common mode noise signals. Optically coupled isolators have been included at the GACS/AGLS digital interface as well as the AGLS digital/analog interface, thus permitting each system to be grounded at its optimum point while providing data flow between the systems. A block diagram of the instrument controller unit is shown in Figure 10.

The DCU consists of seven printed circuit boards as follows:

a. Central Processor Unit (CPU)

b. Parallel Interface Adaptor (PIA)

c. GACS Interface

d. Dual Analog to Digital Converter

e. Multiplexed Analog to Digital Converter

f. Dual Digital to Analog Converter

g. Power Supply

Each of these elements is described in the following paragraphs.

a. Central Processor Unit -- The CPU board contains all the components for a complete microcomputer system, requiring only power and an input/output device to provide a working digital system. The board is a general purpose computer board, containing a Motorola M6800 CPU, 4096 bytes of program memory (PROM), 4096 bytes of random access memory (RAM), two serial asynchronous interfaces (ACIA), one parallel interface adaptor (PIA), a programmable timer, and address bus drivers to interface the CPU to the remainder of the digital system.

The firmware, which determines the operating characteristics of the digital system, is stored in four electrically programmable memory (EPROM) 2708 integrated circuits. These circuits are mounted in sockets on the CPU board to facilitate program changes during development. Temporary memory, used to store

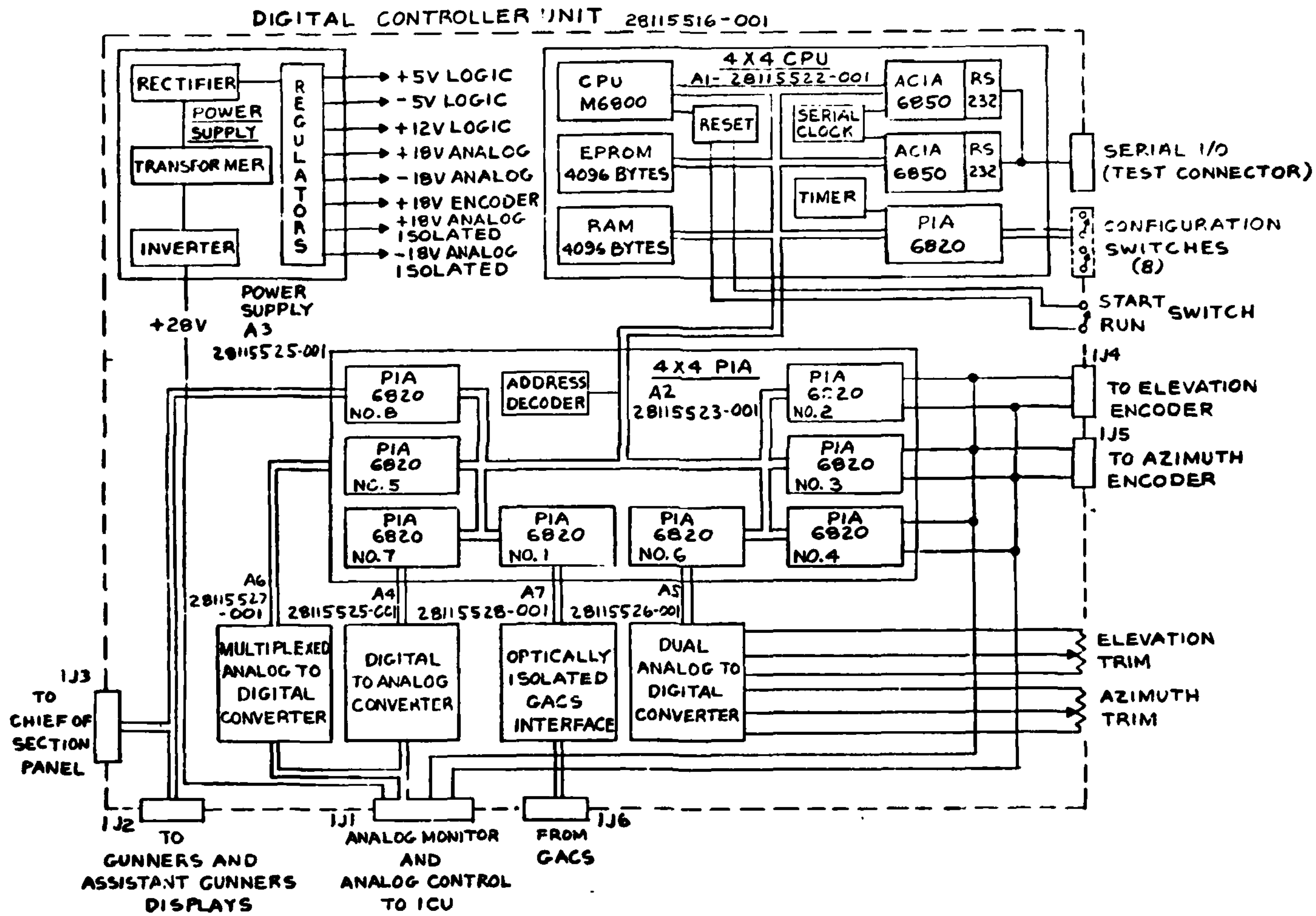


FIGURE 10. BLOCK DIAGRAM, DIGITAL CONTROLLER UNIT

intermediate data while the program is operating, is provided by the random access memory (RAM).

The ACIAs permit internal access to the CPU by keyboard or phone line for troubleshooting. They are coupled through a cable to the external test connector on the DCU. The timer is used to measure elapsed time for those program tests with a time and magnitude requirement.

In the AGLS application, the CPU board is directly connected to the configuration switch register by a separate cable and connector. The switch register permits selection of program to select the level of automation under the control of the test director. The switches are coupled to the CPU by the on-board PIA. The remainder of the digital components are accessed through the PIA board.

b. Parallel Interface Adaptor -- The PIA board contains eight identical Motorola 6820 PIA circuits, each accessing two 8 bit ports, or 16 lines of input or output data, coupled through a ribbon cable to another interface board. An address decoder is included on the PIA board, to indicate which of the PIA circuits should be connected to the CPU data bus at any given time. The PIA board essentially expands the 8-line CPU data bus to 128 lines of input or output data. The PIA board drives the display panels data bus directly through PIA circuit number 8 (Figure 10).

c. GACS Interface -- The GACS interface board connects the output of the two GACS 16-line command channels to the PIA board, using optical isolators to separate the GACS and AGLS ground connections. The GACS output circuit permits corresponding lines of the two channels to be connected to a single wire, as long as only one channel is active at any given moment. Two optically coupled isolators are also provided to activate the GACS azimuth or elevation output, under program control. The GACS data is coupled to PIA circuit number 1 (Figure 10).

d. Dual Analog to Digital Converter -- The dual channel analog to digital converter board is used to interface the azimuth and elevation encoder trim potentiometers into the digital system. A reference voltage of 10 volts is supplied to each potentiometer. The potentiometer output is routed to a buffer operational amplifier, a sample and hold amplifier, and then to the analog to digital converter. The output of the eight-bit converter is connected to PIA circuit number 6 (Figure 10). The circuitry is adjusted to yield a full eight bit change in the output code thus permitting a trim range of $\pm$ 12.8 mils for ten turns on the potentiometer.

e. Multiplexed Analog to Digital Converter -- The multiplexed analog to digital converter board accepts the analog error signals from the leveling servos and the IR tracker, and sequentially converts each of these to a digital signal. The digital signal is then transmitted through optically coupled isolators to PIA circuit number 5 (Figure 10) on the PIA board. The CPU compares the digitized errors to an acceptance level, to determine which status lamps should be illuminated.

f. Dual Digital to Analog Converter -- The dual digital to analog converter accepts azimuth and elevation errors calculated by the CPU, and converts them to analog correction signals to be applied to the pantel and quadrant pitch servos. The digital errors are provided by PIA circuit number 7 (Figure 10), optically isolated, and stored in either the azimuth or elevation latch, under control of commands from the CPU. The stored data from each latch is applied to its own D/A converter, which generates an analog signal of up to $\pm$ 10 volts full scale, proportional to the input digital error.

g. Power Supply -- The power supply accepts +28 volt regulated power from the system power supply, and converts it to the following dc voltages:

+5 volts - Logic supply

-5 volts - Logic supply

+12 volts - Logic supply

+18 volts - Encoder supply

+20 volts - Analog supply

-20 volts - Analog supply

+20 volts - Isolated analog supply

-20 volts - Isolated analog supply

The input direct current power is converted to alternating current by the inverter and then applied to a transformer with multiple secondary windings. The output voltages are obtained by rectifying the various transformer voltages, and then regulating the +5 volt, -5 volt, +12 volt, and +18 volt outputs. The +20 volt and -20 volt supplies are regulated to +15 volts and -15 volts on the individual A to D and D to A boards, to minimize the effects of system noise and provide more accurate reference voltages at each board.

2. Chief of Section Panel (COS)

The chief of section panel contains the operating controls for the AGLS, as well as numerical displays of the commanded azimuth and elevation data from the GACS gun unit, actual data corresponding to the counter readings of the M-117 telescope and M-15 quadrant, and the respective errors between commanded and actual values. The panel also contains status lamps to indicate acceptable leveling of the M-15 quadrant and M-145 mount, acceptable tracker to GACS RU lock-on (A/P), and presence of the RU in the tracker field of view (XENON lamp). If any of the above lamps extinguish, the No-Go lamp will illuminate. The chief of section panel is shown on Figure 11.

This panel contains a control to adjust the display brightness, and a test button to check proper function of all the display elements.

The following switches are located on the chief of section panel:

Power -- Activates the system power supply, digital controller unit, all data displays, and certain other electronic assemblies.

Servos -- Activates those fire control instrument servos that have been previously selected by the system configuration switches.

Weapon -- Activates the weapon azimuth and elevation servos if they have been selected by the configuration switches, and if certain check conditions have been satisfied.

Figure 11. Chief of Section Panel

Load Position -- Selects either GACS elevation (down) or previously selected load position (up) to be the command to the weapon elevation servo.

RU Search -- Causes the panoramic telescope to slew clockwise (right) or counterclockwise (left) to locate the GACS Reference Unit, if certain conditions have been satisfied.

Data displays on the chief of section panel consist of the following:

Elevation Commanded Data -- The commanded elevation from the GACS gun unit or the preselected load position.

Elevation Actual -- The elevation value displayed on the M-15 quadrant.

Elevation Error -- The difference between the two above values.

Azimuth Commanded Data -- The commanded azimuth from the GACS gun unit.

Azimuth Actual -- The deflection displayed in the upper counter of the M-117 telescope.

Azimuth Error -- The difference between the two above values.

The panel accepts the above data in Binary Coded Decimal (BCD) format, transmitted bit parallel, character serial from the digital controller unit. The panel also contains a power supply which converts the regulated 28 vdc power to +5 vdc required by the display electronics.

3. Gunner's Display Panel

The gunner's display panel accepts and displays the same azimuth data as is displayed on the COS panel. This panel also contains a 28 volt to 5 volt converter to energize the internal electronics. The gunner's display panel is shown in Figure 12.

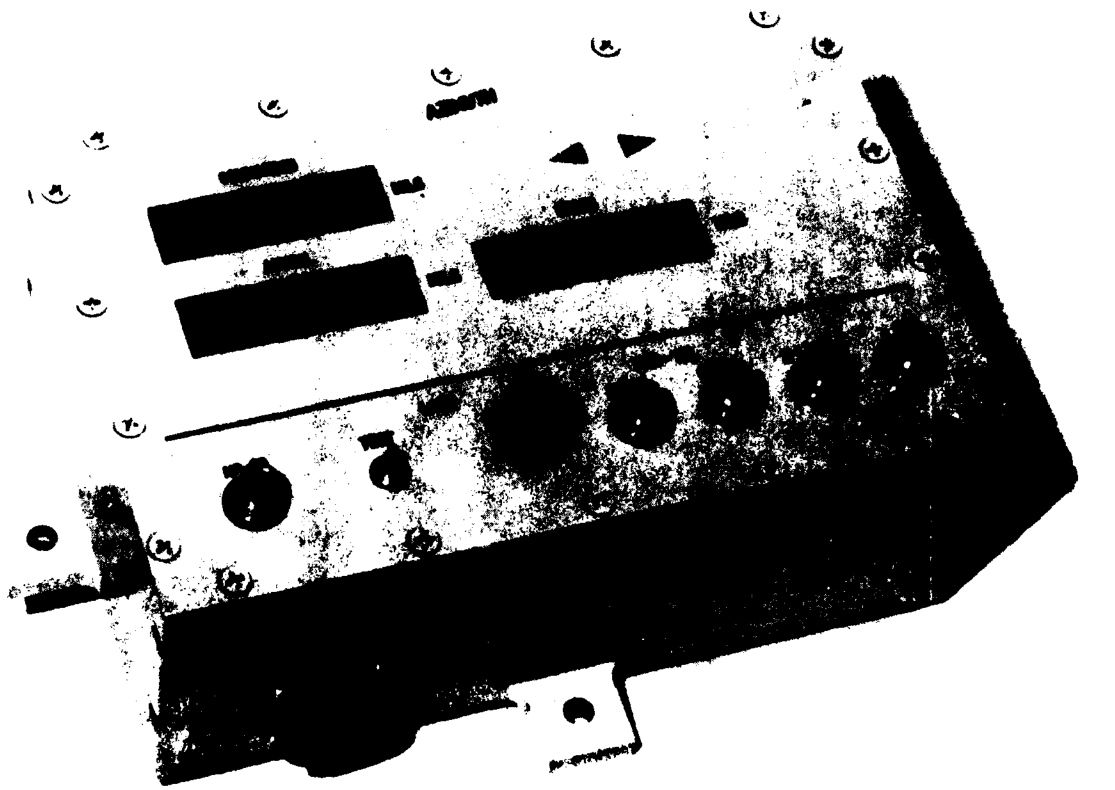


Figure 12. Gunner's Display Panel

4. Assistant Gunner's Display Panel

The assistant gunner's display panel accepts and displays the same elevation data as is displayed on the COS panel and contains a power supply identical to that used by the gunner's display panel. The assistant gunner's display panel is shown in Figure 13.

C. Gun Alignment Control System

The Gun Alignment Control System (GACS), developed and manufactured by Aviation Electric Limited, is used by the AGLS to provide an azimuth reference. The GACS consists of six assemblies:

1. Command Post Unit (CPU)
2. Command Post Adaptor Unit
3. Converter/Adaptor Unit
4. IR Receiver
5. GACS Gun Unit
6. GACS Reference Unit

The GACS establishes an azimuth reference by using a rotating laser beam synchronized to a flashing XENON lamp. Any GACS equipped gun can determine its azimuth reference by directing its IR receiver, mounted on the panoramic telescope, toward the reference unit. The GACS gun unit will count the pulses from the XENON lamp and observe the rotating laser to measure the reference angle. The command post unit will transmit, on manual command from the Fire Direction Center (FDC) fire orders to the GACS gun unit. The GACS gun unit will then compute the required deflection by adding the reference angle to the commanded angle. The resulting commanded deflection is automatically transmitted to the AGLS digital subsystem. The GACS also provides a means of transmitting elevation data to the AGLS, and fuse setter data to the GACS gun unit display. The GACS components are described in the following paragraphs.

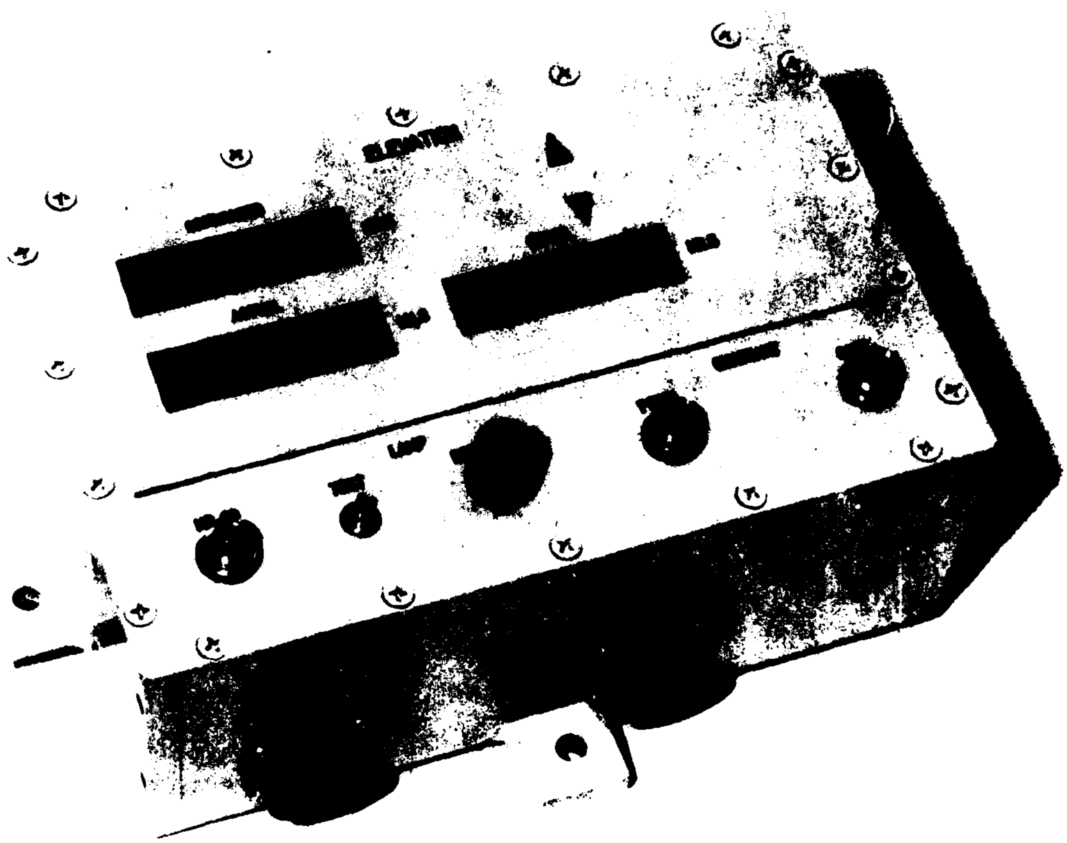


Figure 13. Assistant Gunner's Display Panel

1. Command Post Unit

The command post unit accepts input data by manually set rotary switches. Data to be transmitted consists of deflection, elevation, and fuse setting. After data has been set in, it is transmitted by manually activating a pushbutton. A flashing lamp indicates that data is being transmitted, and a steady lamp indicates that the gun unit has accepted the transmitted data.

2. Command Post Adaptor Unit

The command post adaptor unit provides a means of coupling the command post unit to either a phone line pair or a radio receiver-transmitter.

3. Converter/Adaptor Unit

The converter/adaptor unit, installed in the M-109, accepts the commanded data from the phone lines or radio and couples the data to the GACS gun unit. The converter/adaptor unit also contains a power supply to provide regulated voltages to the gun unit and infrared receiver.

4. IR Receiver

The infrared receiver detects the flashing XENON lamp and the laser beam from the reference unit, and transmits real-time electrical pulse signals as these events occur. The IR receiver is mounted with the AGLS tracker on the panoramic telescope, as shown in Figure 14.

5. GACS Gun Unit

The GACS gun unit accepts the pulses from the GACS infrared receiver to determine the reference angle. It has the capability of adding the reference angle to the commanded angle to compute the commanded deflection. It also has three data display clusters, to display azimuth, elevation, and fuse setting. The azimuth display can exhibit either commanded, reference, or normal angle as selected by a three position switch. Also on the gun unit are two lamps, one to indicate detection of the XENON pulses and, one, the presence of the laser beam.

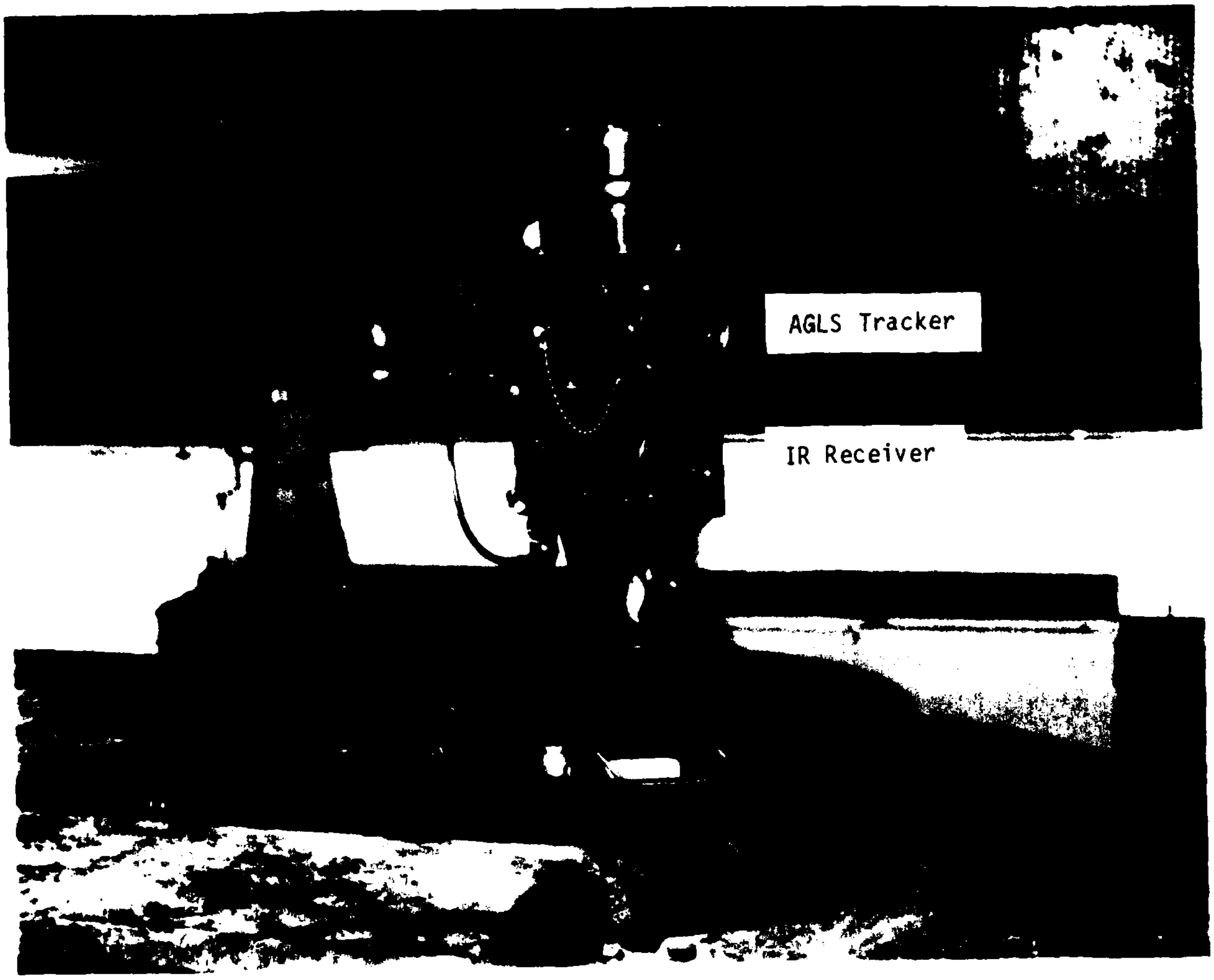


Figure 14. GACS IR Receiver and AGLS Tracker
Installed on M-117 Telescope

6. GACS Reference Unit

The GACS reference unit contains a XENON lamp and a laser diode. The laser rotates one revolution per second, and the XENON lamp flashes once for every 40 mils of laser beam rotation, and flashes twice as the laser rotates through South. The reference unit can be energized by a 24 volt storage battery. Initial alignment of the reference unit is accomplished manually by using either a magnetic compass, or a monocular sight if a survey line is available. The GACS reference unit emplaced in a field situation showing the relationship to the vehicle is shown in Figure 15.

D. Infrared Receiver

The AGLS infrared receiver detects the flashing XENON lamp of the GACS reference unit, and provides a direct current positive or negative signal proportional to the horizontal angular position of the XENON lamp in the tracker field of view. The tracker is sensitive to lamp position in the horizontal axis for displacements of 100 mils to the left and right of center, and will detect the lamp within a $\pm$ 100 mil vertical field of view. The tracker includes direct current rejection circuits and an optical filter to reject ambient light, and contains an automatic gain control to compensate for changes in range from tracker to reference unit. A one-bit digital output is also provided which indicates to the digital controller that the tracker is detecting the GACS reference unit.

E. Weapon Control System

The weapon control subsystem consists of two channels, each consisting of an electrically-operated proportional control servo valve, pressure operated engage valves, an electrically-operated solenoid valve, a tachometer, and a controller module. The two controller modules and their power supply are contained in the Weapon Azimuth and Elevation Controller Unit. See Figure 16.

1. Azimuth Control Subsystem

The azimuth control subsystem is shown in the block diagram of Figure 17. The position error is detected by the infrared tracker mounted on the panoramic

Figure 15. GACS Reference Unit Field Installation

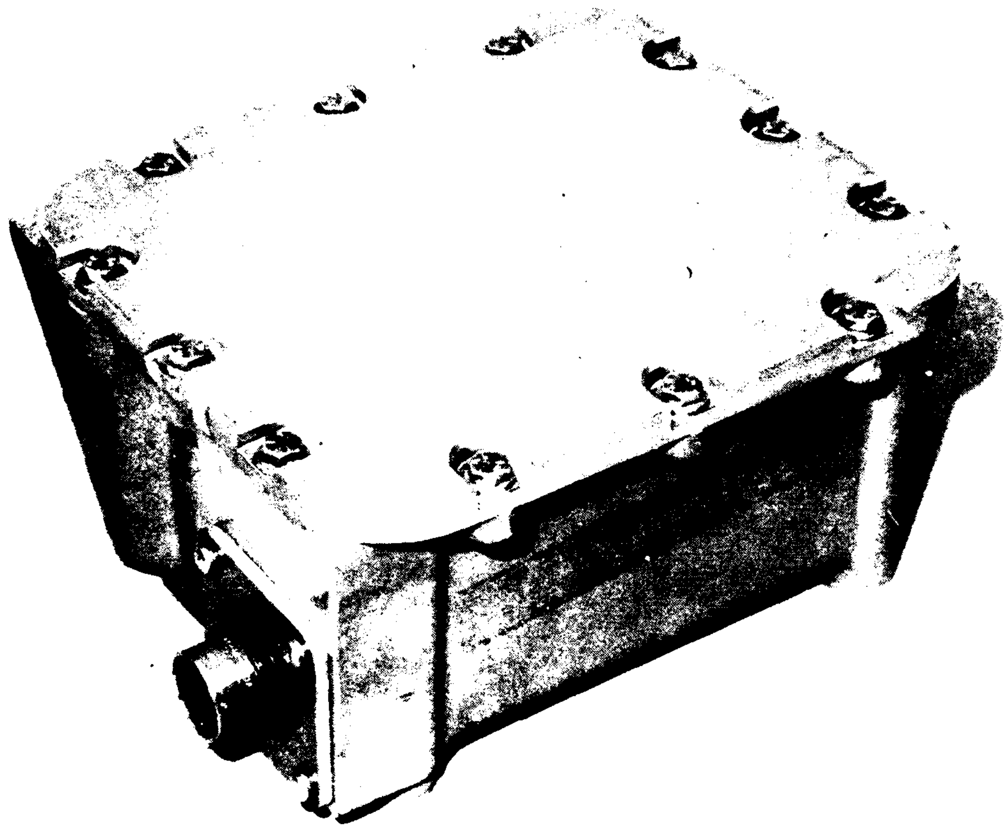


Figure 16. Weapon Azimuth and Elevation Controller Unit

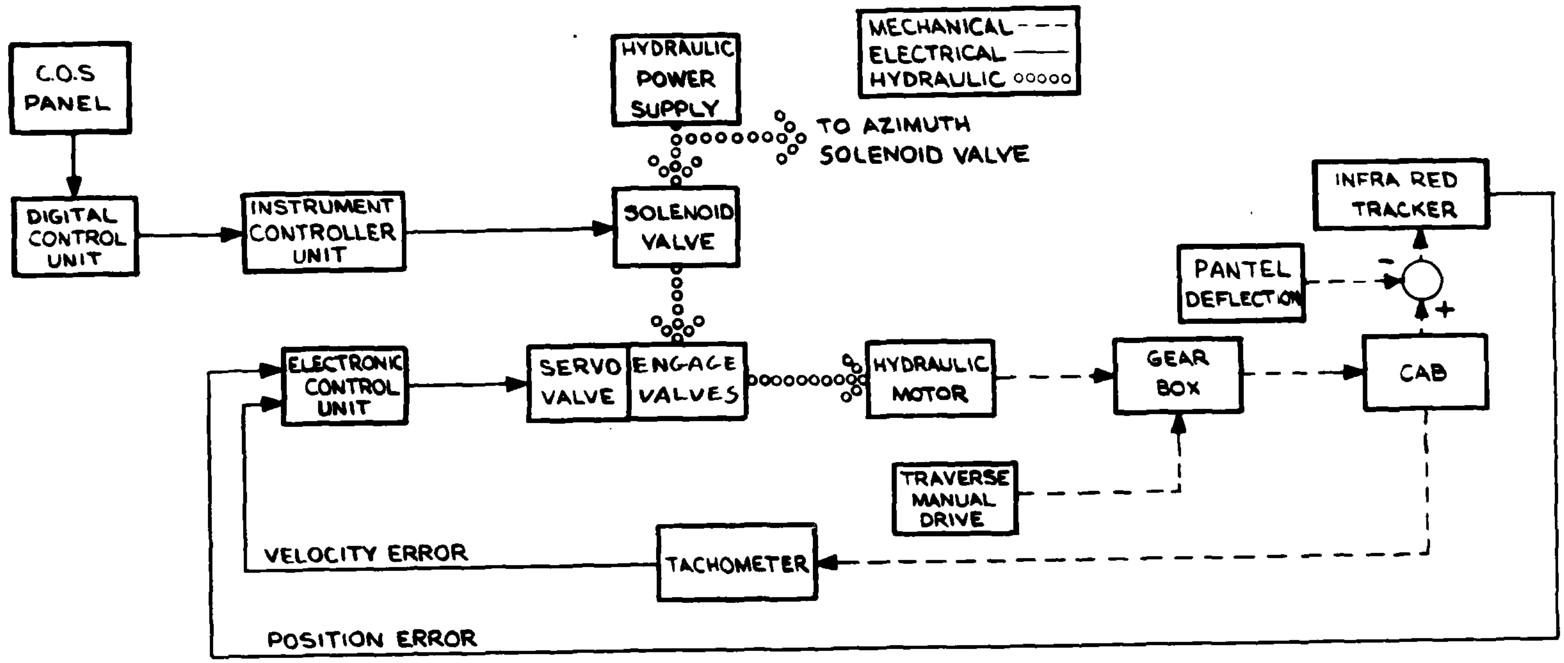


Figure 17. Block Diagram, Weapon Control System - Azimuth

telescope head, and supplied to the azimuth controller module. The module filters the error signal to obtain the desired frequency characteristic, combines the position signal with the tachometer velocity signal, and generates an output error signal to operate the azimuth servo valve.

Hydraulic fluid from the M-109 power pack is filtered and then applied through the azimuth solenoid shut-off valve to the servo valves, and also to the pilot ports of the pressure-operated engage valves. The engage valves will close upon removal of supply pressure, to disconnect the servo valve and permit normal azimuth control with the gunner's control handle. With hydraulic supply pressure applied, the servo valve will apply hydraulic flow to the azimuth hydraulic motor in proportion to the electrical current from the controller module. Direction of hydraulic flow is determined by the polarity of the control current. Pictures of the filter, solenoid shut-off valve and servo valve assembly are shown in Figures 18, 19 and 20 respectively.

The hydraulic motor rotates in response to the servo valve flow, thus rotating the cab to control weapon azimuth. If the panoramic telescope is also being driven, as is the case with the automatic azimuth configuration, the tracker will be driven away from the GACS reference unit, thereby generating a position error which continues to drive the weapon in azimuth until the telescope has reached its commanded deflection. As the telescope comes to rest, the cab will continue to rotate until the final position error, as measured by the tracker, has been reduced to zero. As the weapon approaches its commanded position, the telesocpe mount will be automatically leveled and thus the mount will insert an azimuth correction which compensates for weapon cant by deflecting the telescope line-of-sight. This correction then is automatically inserted as the weapon comes to rest.

A tachometer is utilized to provide a signal proportional to azimuth velocity. This velocity error signal is needed to provide an indication of azimuth velocity, so that the cab will rotate at the proper speed, as the cab and telescope both are driven in the Automatic Offset mode. The azimuth velocity signal is also used as a prediction signal to improve azimuth stability and provide for smooth deceleration as the weapon approaches the final position after a large change in azimuth. A picture of the tachometer is shown in Figure 21.

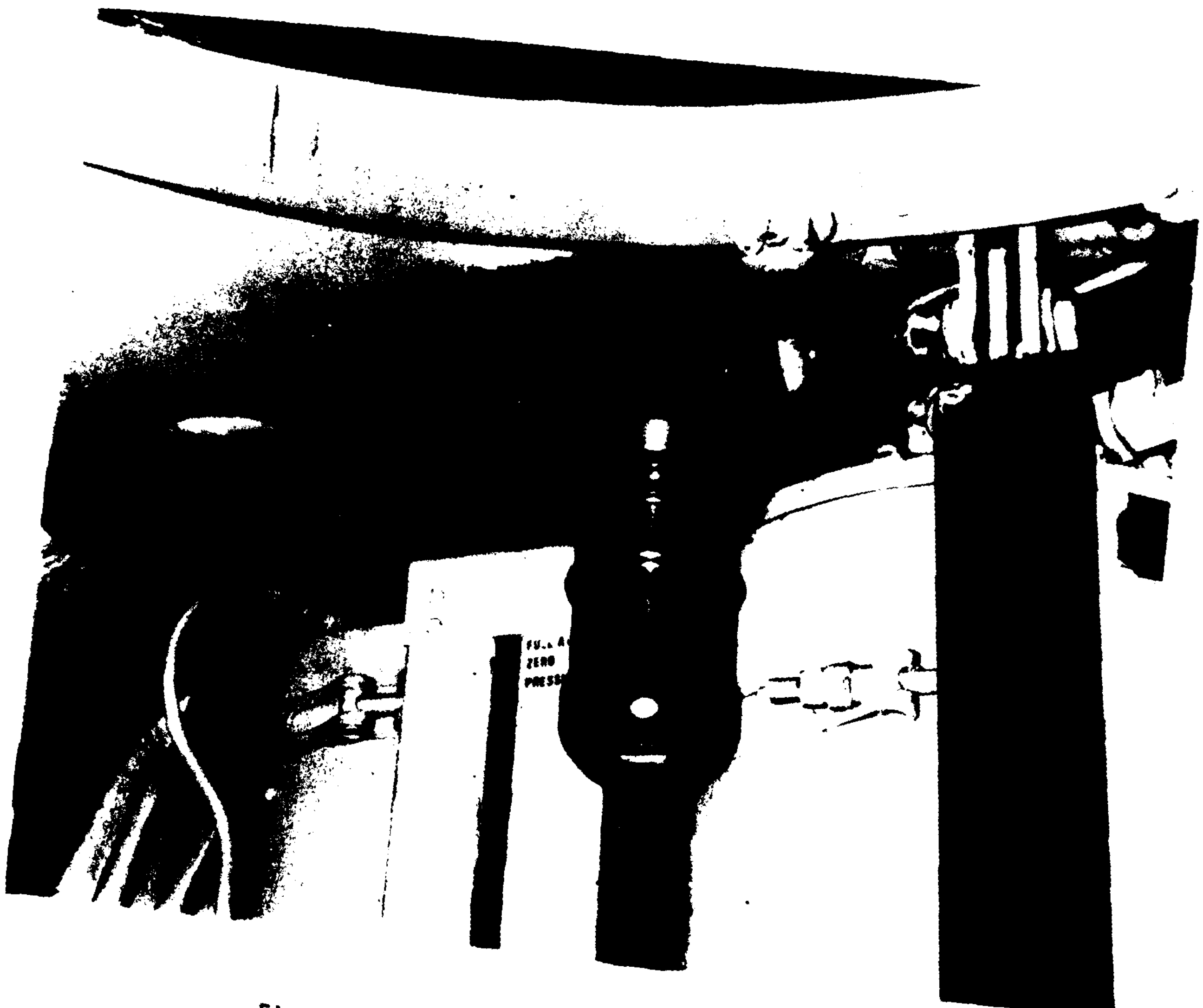


Figure 18. Hydraulic Filter Assembly

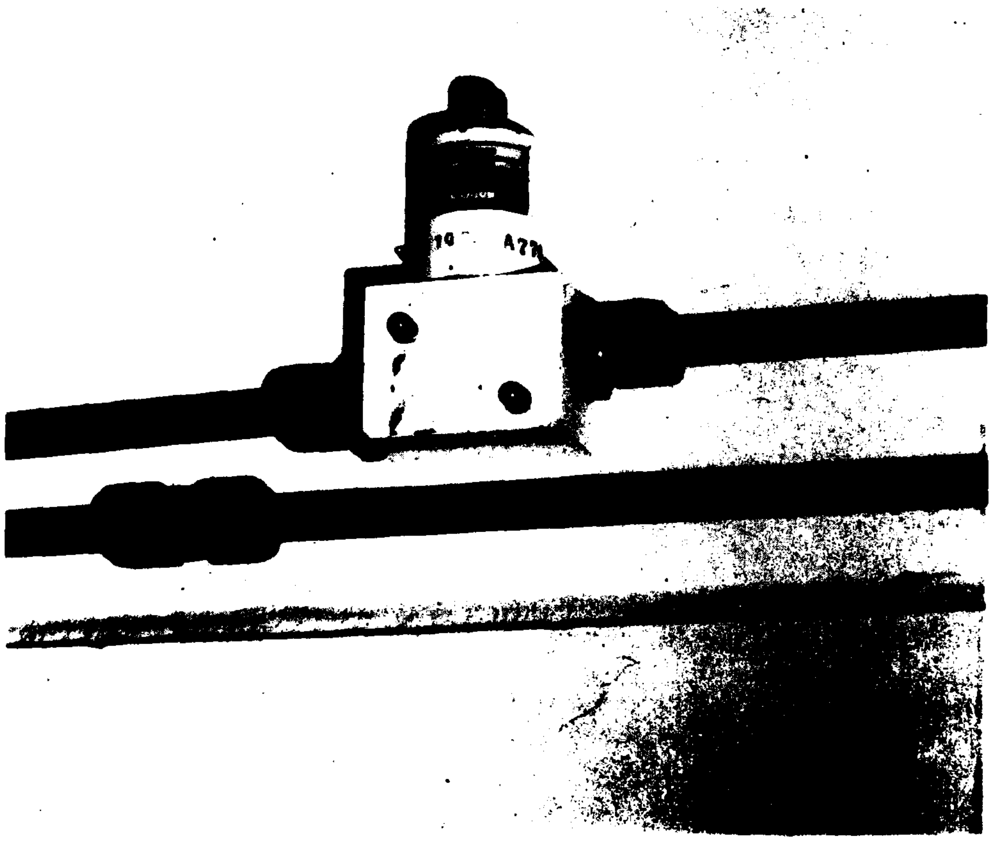


Figure 19. Azimuth Solenoid Shut-Off Valve

Figure 20. Azimuth Servo Valve Assembly

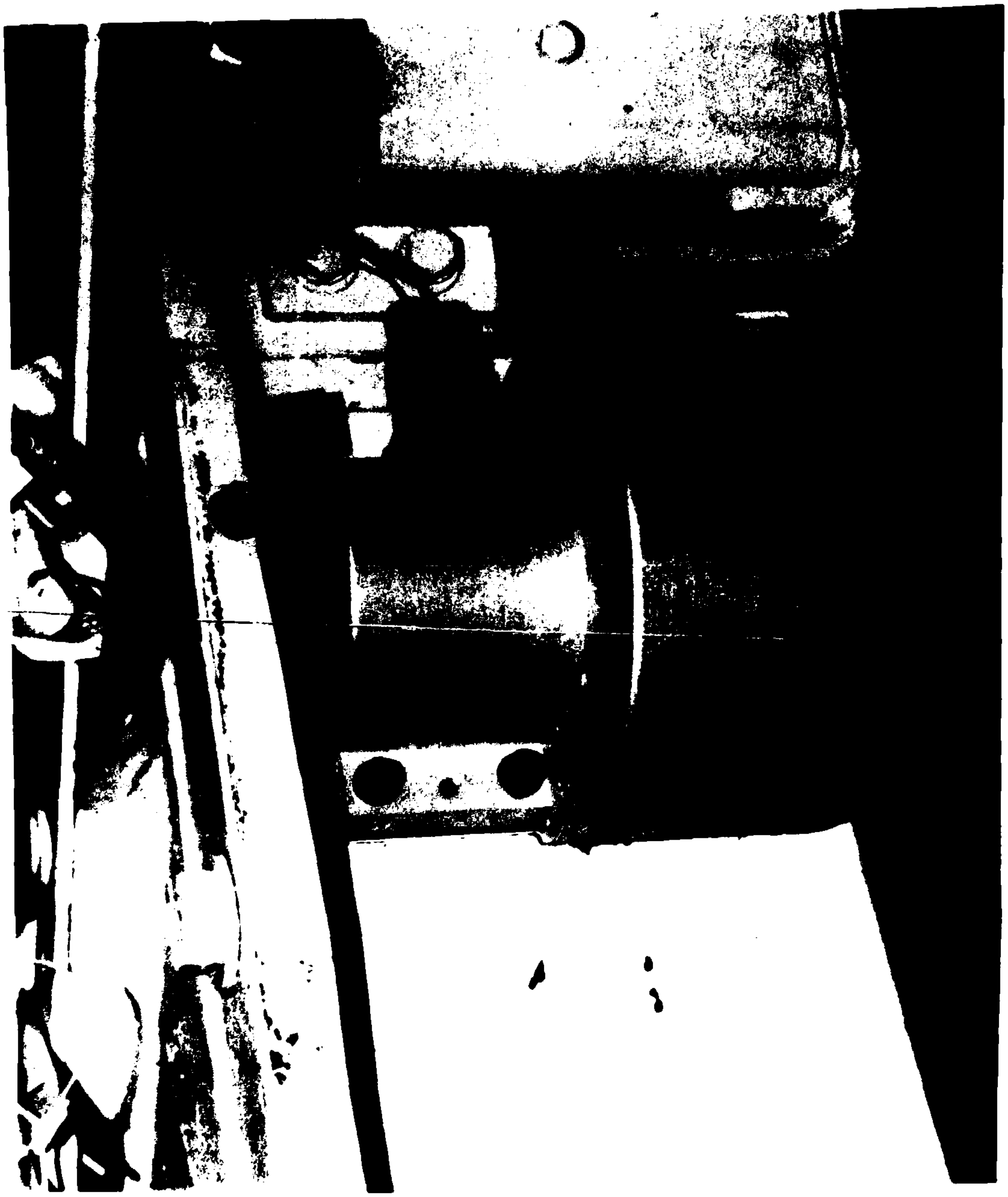


Figure 21. Weapon Azimuth Tachometer-Installed at Ring Gear

2. Elevation Control Subsystem

The elevation control subsystem, shown in the block diagram of Figure 22, is similar in operation to the azimuth control subsystem. The position error is detected by the level sensor mounted on the M-15 quadrant pitch axis and is supplied to the elevation controller module. Hydraulic pressure to the elevation engage valves is applied or removed by the elevation solenoid shut-off valve. The elevation engage valves will close on removal of supply pressure, to disconnect the servo valve and permit control of weapon elevation by either the power control handle or by the manual hand pump. With hydraulic supply pressure applied, the elevation servo valve will control pressure to the elevating mechanism in proportion to the electrical current from the elevation controller module, and polarity of the pressure is determined by polarity of the control current. Pictures of the solenoid shut-off valve and servo valve assembly are shown in Figures 23 and 24 respectively.

As the weapon elevates, the quadrant may also be driven away from level, thus generating a position error which continues to drive the weapon until the quadrant has reached its commanded elevation. After the quadrant reaches the comanded elevation, the weapon will continue to elevate until the position error measured by the level sensor approaches a null, thus indicating that the weapon has reached the proper elevation. As the weapon approaches its final position, the quadrant cant servo is also leveling the quadrant in cant, so that the cant correction is already implemented when the weapon comes to rest at the commanded quadrant elevation.

An elevation tachometer is also provided to generate an elevation velocity error signal. This signal is needed to limit the elevation velocity to a controlled value during large changes in elevation, by providing additional feedback which essentially reduces the influence of the position error signal. The velocity error signal also provides for smooth deceleration and enhanced stability as the weapon comes to rest. A picture of the tachometer in the installed position is shown in Figure 25.

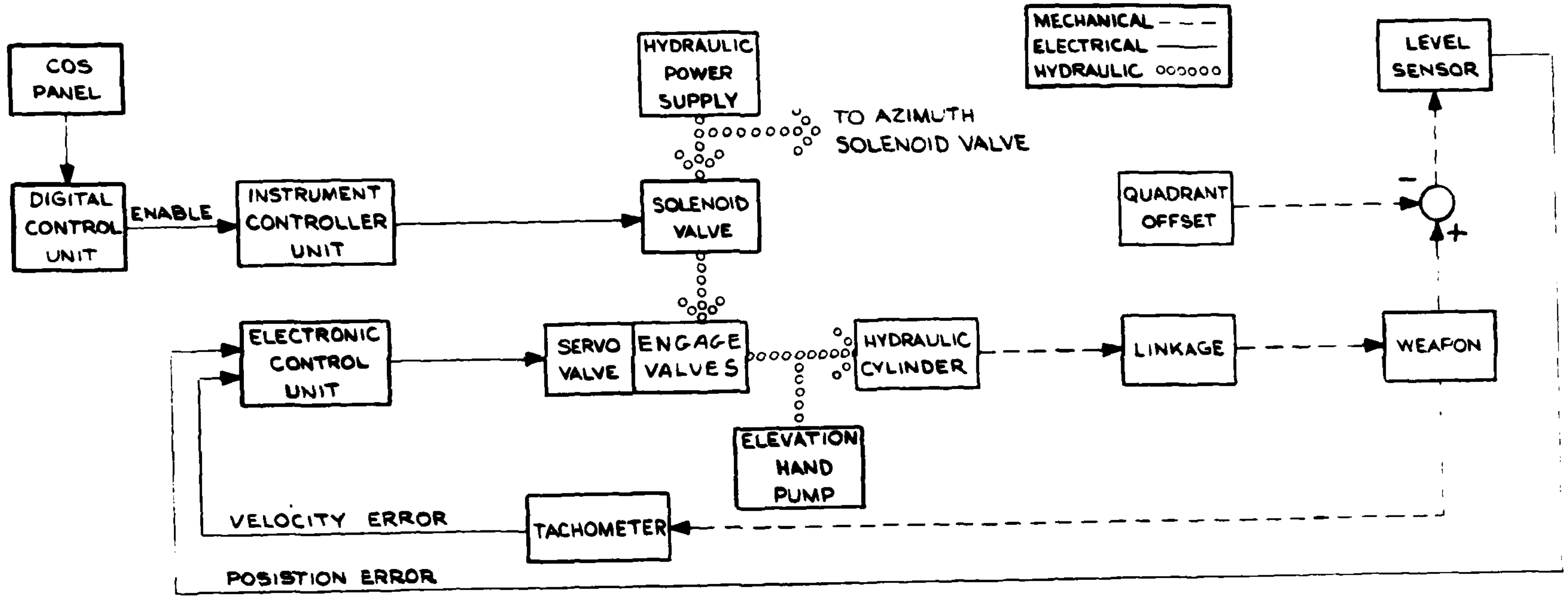


Figure 22. Block Diagram, Weapon Control System-Elevation

Figure 23. Elevation Solenoid Shut-Off Valve

Figure 24. Elevation Servo Valve Assembly-Installed on Elevating Mechanism

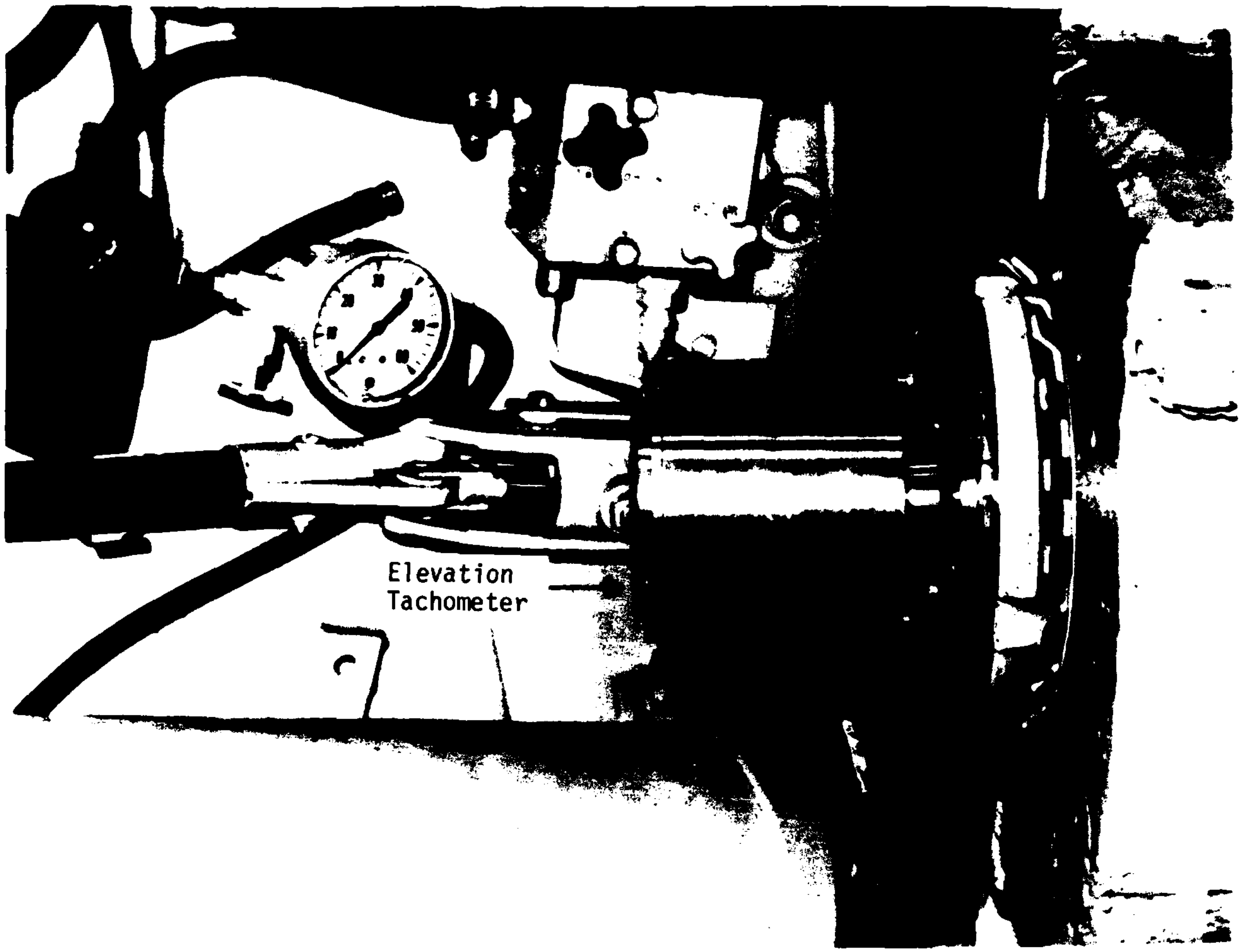


Figure 25. Elevation Tachometer-Installed on Weapon Trunnion

F. System Power Supply

The AGLS system power supply receives +24 volt power from the voltage support battery, and generates the following regulated power:

- +32 volts dc
- -32 volts dc
- +28 volts dc

The positive and negative 32 volt supply is capable of delivering a total of 10 amperes from either or both outputs. These voltages serve as the power source for the five servo amplifiers in the instrument servo controller unit. The positive 28 volt dc supply is capable of delivering 5 amperes and is the power source for the digital controller unit, the chief of section panel, and the gunner's and assistant gunner's display panels. A picture of the power supply is shown in Figure 26.

The power supply, shown in the block diagram of Figure 27, consists of two switching regulators, each controlling power to an inverter, with a common frequency source. Input power from the voltage support battery is applied through a manually resettable circuit breaker to a power relay. The power relay, controlled by the power switch on the chief of section panel, applies power to the two switching regulators and serves as the means of energizing or de-energizing the AGLS subsystems.

The +28 volt regulated output power is controlled by two semiconductor power switches on assembly A1 (Figure 27). The input power is filtered and applied to the power switches. Each switch is either completely on or off. For example, when power switch U10 is on, current flows from the input filter through U10, the inductor L1 and the current monitor resistors to the 28 volt inverter A4. When the switch U10 shuts off, the current flows through diode D10 through L1 to the load. The switching regulator controls the output voltage to the inverter A4 by adjusting the percentage of time that U10 is conducting. Switches U10 and U11 are essentially in parallel, and the current in each switch is monitored by the current shunt resistors R124 and R125. The pulse width modulator adjusts the conduction times to equalize the current in each switch.

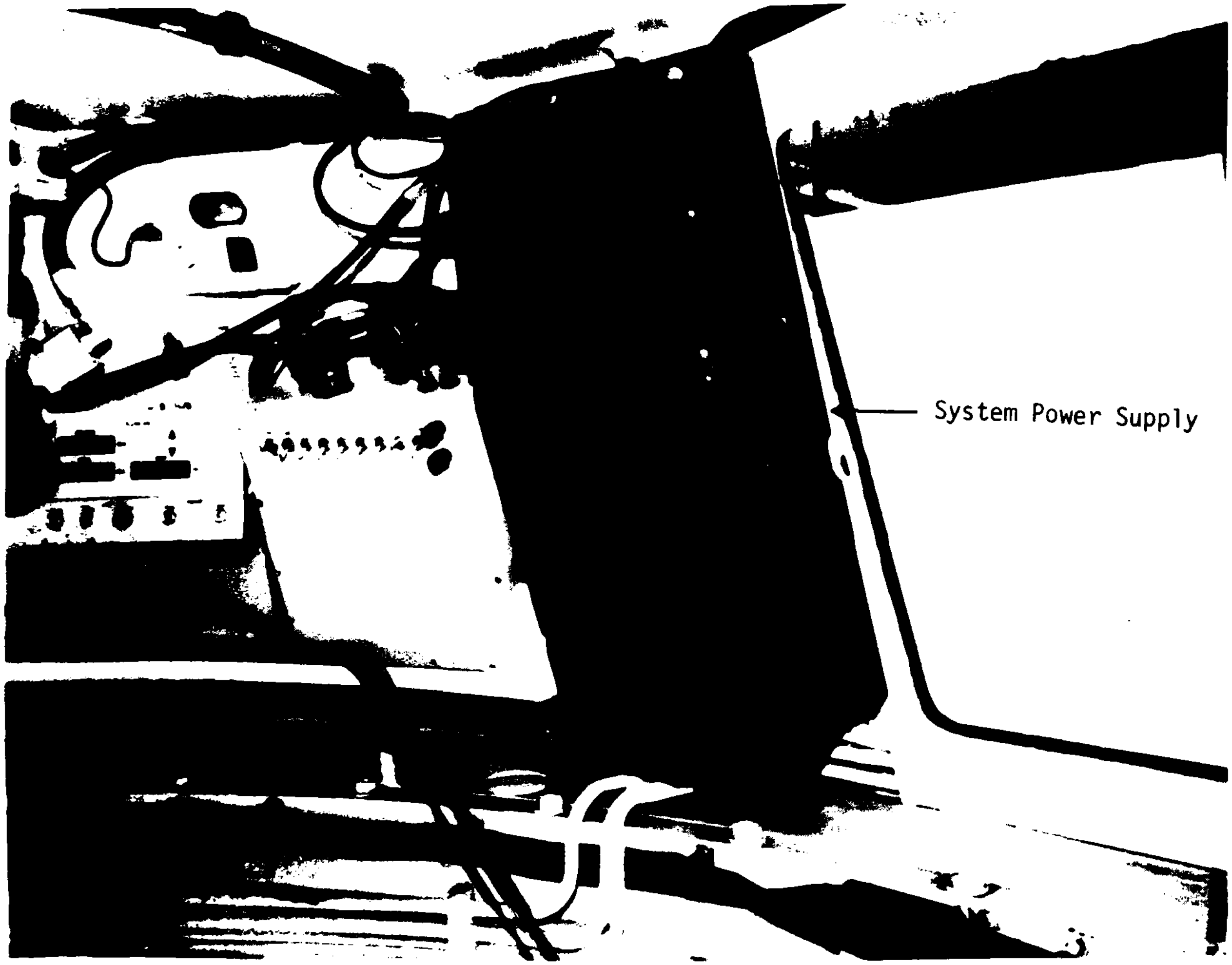


Figure 26. System Power Supply-Installed Position

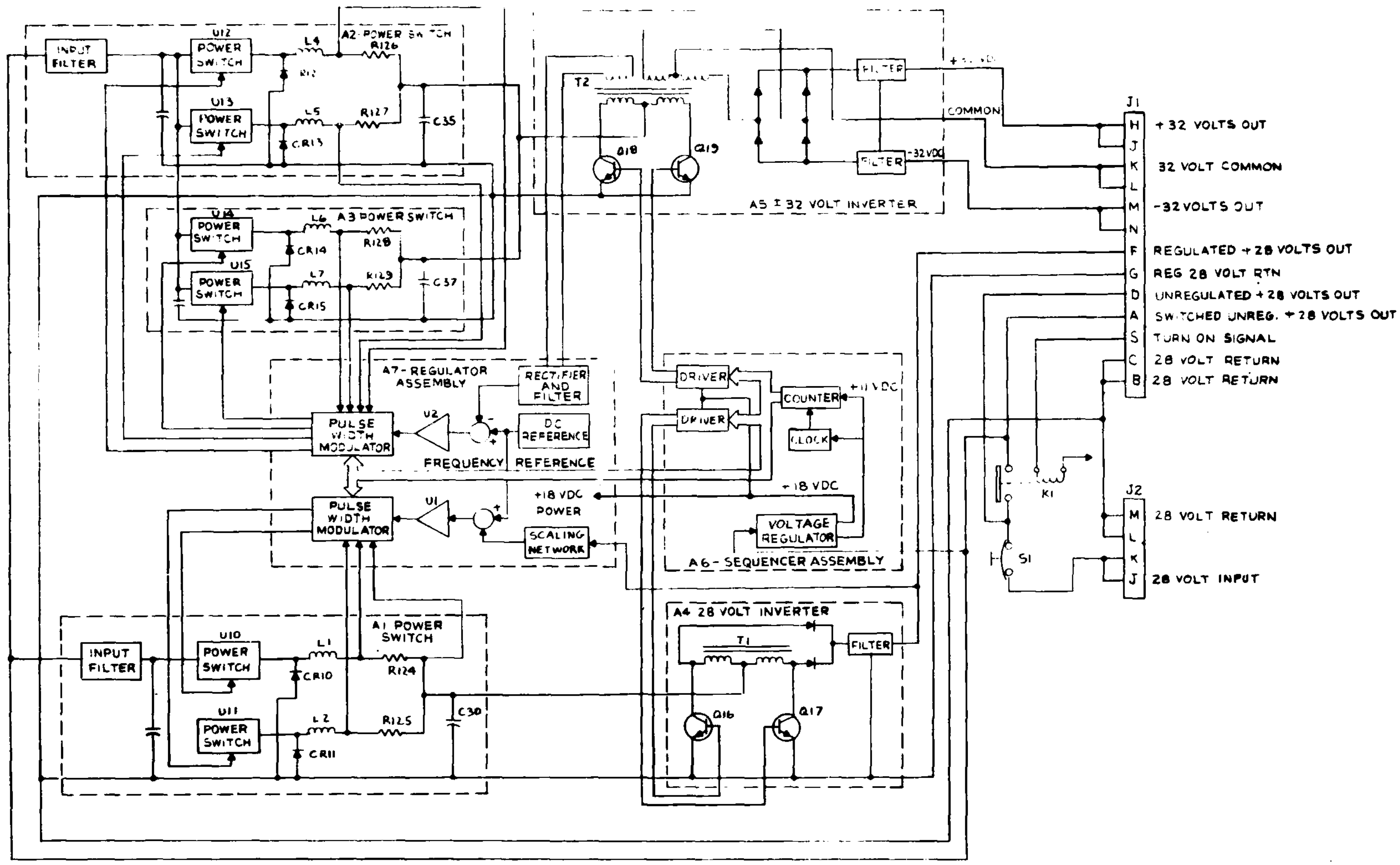


Figure 27. Block Diagram, System Power Supply

The output currents from switches U10 and U11 are combined, and then applied to inverter A4 consisting of transistors Q16 and Q17 and transformer T1. Transistors Q16 and Q17 are alternately driven on and off each for slightly less than 50 percent conduction ratio. Since the two windings on transformer T1 are equal, the inverter essentially doubles the switching regulator output to obtain +28 volts output for +15 volts switching regulator output. The regulated +28 volt output voltage is attentuated by the scaling network, and then compared with a reference voltage. The difference is amplified by U1 and applied as the input to the pulse width modulator. Thus, the switching regulator conduction time is automatically adjusted to maintain a constant output voltage as input voltage and load change.

The current monitor inputs will override the error voltage from U1 if either switch current exceeds 6.0 amps, and will then limit the switch currents to 6.0 amps each, regardless of load resistance. This will limit the short circuit current to 6.0 amps in the event of a regulated 28 volt overload, and prevent further system damage.

The $\pm$ 32 volt power supply is similar to the +28 volt supply, except that four power switches, U12, U13, U14 and U15 on assemblies A2 and A3 are connected in parallel to provide the current to inverter A5. Inverter A5, consisting of transformer T2 and transistors Q18 and Q19, multiplies the switching regulator output by a factor of 2.5, and provides isolation of the analog system ground with respect to the M-109 power ground. Feedback voltage is taken from the transformer, rectified, filtered, and compared with the reference voltage. The difference is amplified and applied to the 32 volt pulse width modulator to control the conduction ratio of the power switches. As in the +28 volt regulator, current is measured by the current monitors R126, R127, R128 and R129 to balance the load in each switch. The monitor signals also limit each switch current to 6.0 amps, thereby providing a limit of 10.0 amps on the total +32 and −32 volt supplies to prevent power supply damage in the event of a system overload or short circuit.

All control logic for the switches and inverters is provided by the two circuit boards A6 and A7. The sequencer (A6) contains the master clock for the switching regulators, and the drive amplifiers for both inverters. The regulator assembly

A7 contains the control amplifiers for both basic power supplies, and the modulators and drivers for all six power switches. Input and output power filters are included on the individual assemblies to minimize the electrical noise from the power switches.

V. DESIGN STUDY

A. Theory of Operation

1. Methodology

The Automated Gun Laying System, shown in block diagram form in Figure 1, is configured to perform the same fire control functions as are now performed manually. These functions are enumerated below:

1. Level telescope pitch
2. Cross level telescope
3. Level or offset quadrant cant
4. Cross level quadrant
5. Offset telescope
6. Drive weapon azimuth
7. Drive weapon elevation

Basically, the methodology employed in automating the M-109 was to retain the existing fire control geometry, to add sensors in parallel with the existing sensors and to add actuators in parallel with the existing manual controls. As an example, the cant level axis of the M-15 quadrant is shown in the block diagram of Figure 6. The basic quadrant is cross leveled by the assistant gunner, who rotates the cross level knob while observing the spirit vial which tells him in which direction to turn the knob. In automating this axis, a level sensor is attached to the level vial, an electric servo motor is coupled to the knob through gearing, and a power amplifier converts the level sensor output voltage to a current sufficient to drive the motor.

2. Fire Control Servos

All of the fire control servos can be represented by the basic block diagram of Figure 28. The basic servo system utilizes the concept of inner loop velocity feedback, with a tachometer closely coupled to the actuator to accurately measure

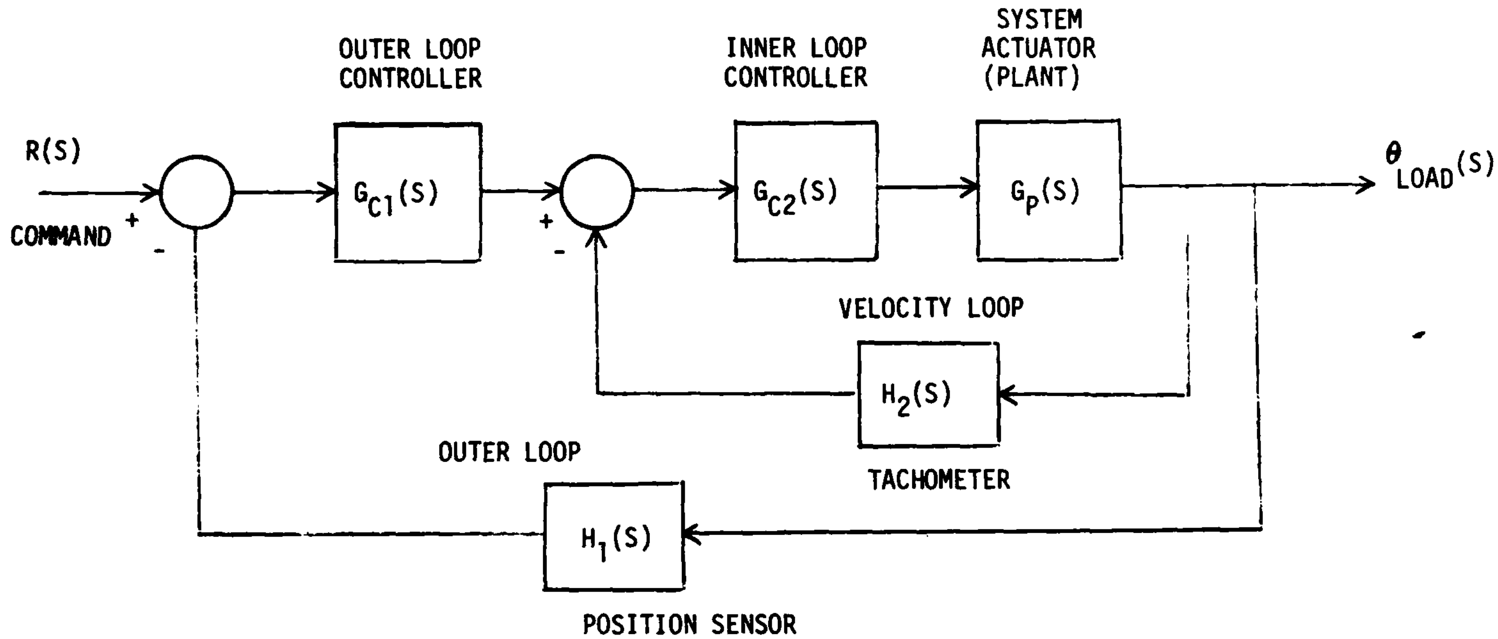


FIGURE 28. BASIC BLOCK DIAGRAM OF AGLS SERVOS

actuator movement while minimizing the effects of backlash and mechanical compliance.

The drive torque for each axis is provided by a motor-tachometer, shown in Figure 29, consisting of a direct current torque motor, with a dc tachometer closely coupled on the same shaft. A motor was selected which had sufficient power to drive each of the instrument servos, as shown in the listed requirements of Table I. In the two quadrant and two telescope mount axes, a gear ratio of 20 to 1 was used, and a ratio of 10 to 1 was used in the telescope azimuth axis. Although the gearing did increase the mechanical complexity of the servo drivers, it did permit use of a much smaller and lighter motor, thus resulting in less total actuator weight.

The motor and tachometer are coupled by a steel shaft with no linkages, thus the only dynamic element separating the motor and tachometer is a torsional resonance, estimated to be in excess of 10,000 Hertz. The remaining dynamic effect is the simple first order expression for a dc motor, with a time constant determined by motor inertia, torque constant, and armature circuit resistance. Actual motor-tachometer data from the quadrant cant axis is shown in Figure 30, and does exhibit the predicted dynamic performance. With an actuator and feedback sensor exhibiting dynamic characteristics approaching the ideal, it is possible to utilize high gain in the inner servo loop. The inner loop controllers, identical for all five fire control servos, can then control motor shaft rotation to achieve very low residual error in response to the outer loop sensor.

While the tachometer provides the short term corrections for actuator control, the outer feedback loop is used with a position sensor to drive the system to the desired null position. The outer loop then can be considered as a trimming control, which monitors the at-rest position, compares it with the commanded position, and applies a correction signal to drive the inner loop and ultimately correct the load position.

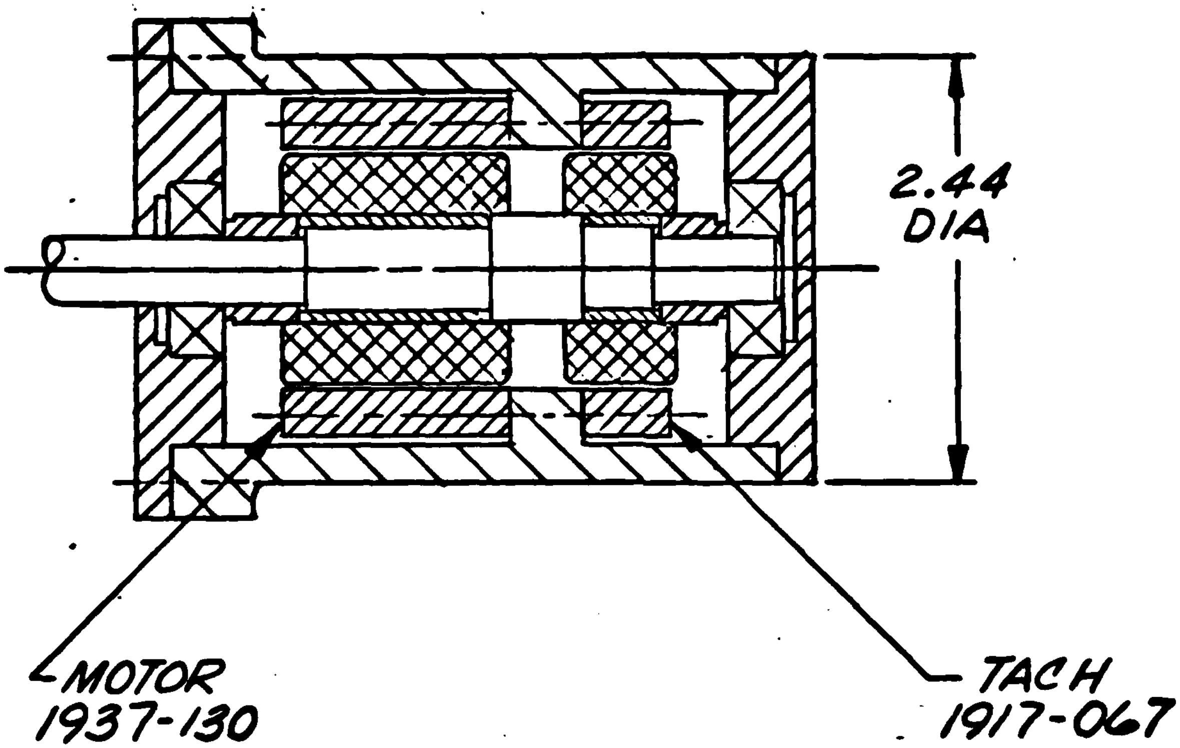


Figure 29. Servo Motor-Tachometer Cross Section Diagram

TABLE I

Fire Control Instrument Servo Performance Parameters

Servo Axis	Required Load Torque (Maximum)	Required Load Speed	Gear Ratio	Calculated Knob Torque @ Gear Ratio (Maximum)	Calculated Knob Speed @ Load Torque (Maximum)
Telescope Azimuth	9.0 lb-in	150 RPM 125 mils/sec	10:1	46.9 lb-in	161.6 RPM 134.7 mils/sec
Mount Pitch	25.0 lb-in	57 RPM 40 mils/sec	20:1	93.8 lb-in	73.3 RPM 51 mils/sec
Mount Cant	12.0 lb-in	71 RPM 40 mils/sec	20:1	93.8 lb-in	87.2 RPM 49 mils/sec
Quadrant Pitch	20.0 lb-in	60 RPM 100 mils/sec	20:1	93.8 lb-in	78.7 RPM 131 mils/sec
Quadrant Cant	18.0 lb-in	60 RPM 100 mils/sec	20:1	93.8 lb-in	80.8 RPM 135 mils/sec

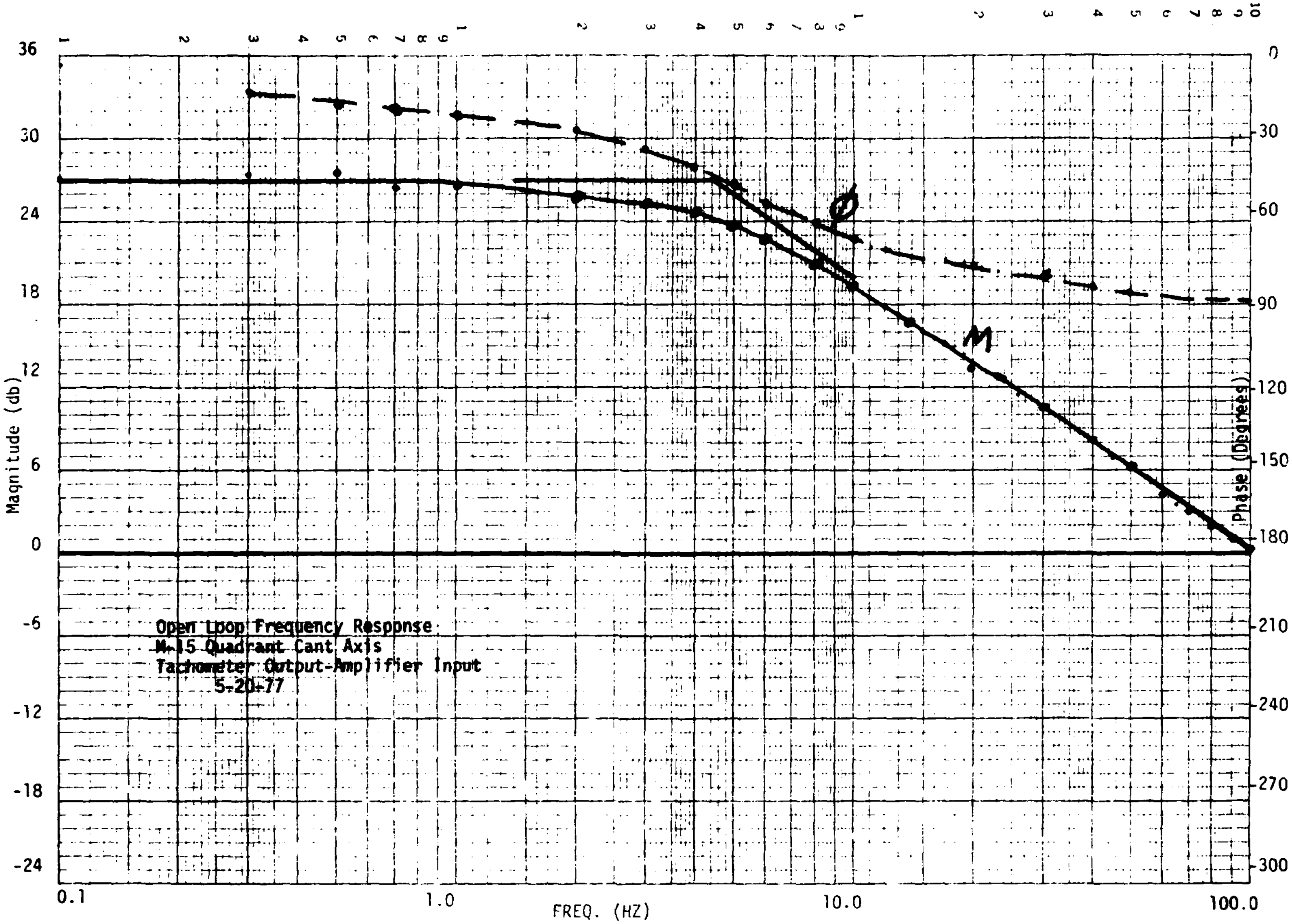


Figure 30. Open Loop Frequency Response M-15 Quadrant Cant Axis - Tachometer Output-Amplifier Input

3. Position Sensors

Three different types of position sensors are used in the AGLS fire control servos, depending on the system requirements of the particular servo. Level sensing is accomplished with accelerometers, shaft position is measured with digital encoders, and the position error of the panoramic telescope is measured with an infrared tracker; each of these is described in the following paragraphs.

a. Level Sensor

Level sensing is accomplished by a GG326 accelerometer built by Honeywell Avionics Division. This accelerometer is mounted with its sensitive axis in the horizontal plane, parallel to the level vial of the axis to be leveled. In this orientation, the accelerometer senses local gravity, and will provide a positive or negative dc signal proportional to the angular displacement from level.

Figure 31 is a schematic view of the GG326 Accelerometer.

The pendulum and torsional suspension is fabricated from quartz fiber. A thin coating of metal is vapor-deposited over the length of the suspension and pendulum, providing a conducting surface. The base of the pendulum is positioned in a permanent magnet field so that current flowing in the pendulum circuit acts as a one-turn torque generator.

The optical pickoff consists of a miniature tungsten filament lamp and a silicon pn junction photodiode. The p-layer of the photodiode is divided into equal parts with a 0.003 inch separation. At the null position, the base of the pendulum coincides with the slot in the photodiode.

An acceleration input will cause the pendulum to deflect from the null position, increasing the amount of light incident on one-half of the photodiode while decreasing the light on the other half. The light unbalance produces a differential voltage signal at the output of the photodiodes. The photodiode signal is amplified and fed back to the torque generator in the proper phase to restore the pendulum to the null position. The rebalance current is directly proportional to the input acceleration and is converted to a voltage by a series resistor.

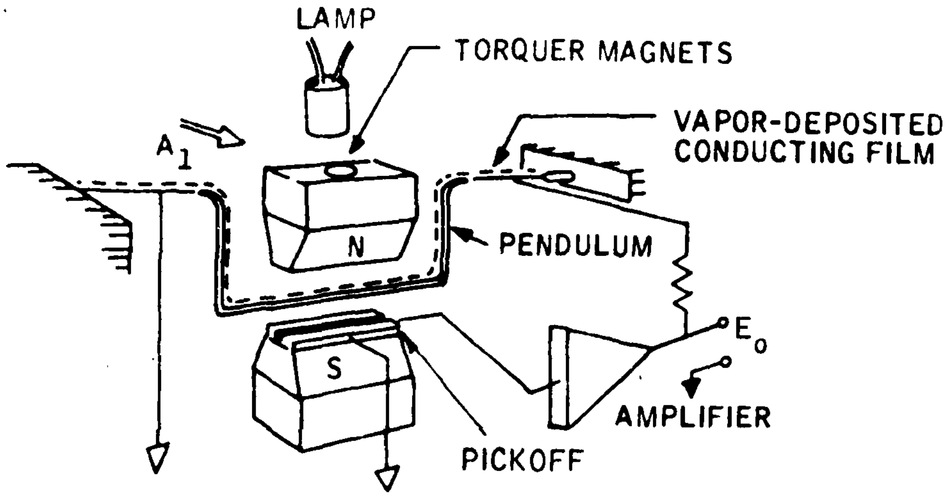


Figure 31. Schematic Diagram, GG326 Accelerometer

A significant feature of the GG326 accelerometer is the low elastic restraint of the quartz fiber suspension. This restraint is 0.5 g/radian (g/rad), compared to 2 to 3 g's/rad for metal flexure pivots or torsional suspension. Since the dual photodiode is fabricated from a single silicon chip, the output voltages of the two halves closely track with temperature. Any photodiode deviations have a small effect on accelerometer null bias due to the low elastic restraint.

The pendulum assembly is mounted in a hermetically sealed aluminum housing. The housing is filled with a low viscosity silicon fluid to control the dynamic characteristics of the accelerometer. The electronics assembly is designed to accept the terminals from the sensor and is bonded directly to the sensor housing. A hermetically sealed cover over the electronics assembly provides the electrical terminations for the accelerometer, as well as the protection against humidity and other destructive environments.

Since the accelerometer will sense lateral acceleration as well as the acceleration of gravity, its dynamic characteristics must be considered. The accelerometer is considered to be off the axis of rotation for a given leveling application, by a distance R. This displacement includes physical distance due to design constraints of the fire control instrument, internal displacement of the sensitive element of the accelerometer from its mounting face, as well as unknowns in the actual location of the rotational center. The relationship of accelerometer location with respect to axis rotation can take four forms, as shown in Figure 32. In all cases, the rotation θ is considered positive when rotation is in the counterclockwise direction.

In form (a), the accelerometer output V_a is equal to

$$V_a = K_a \left(g \sin \theta + R \frac{d^2\theta}{dt^2}\right)$$

where the first term is due to the angular deviation from level when the accelerometer is at rest, and the second term is due to lateral displacement when the axis in question is being rotated. There will also be a centripetal acceleration applied to the accelerometer in a direction toward the center of rotation, but this term will not cause an accelerometer output since it is not in the direction of the sensitive axis.

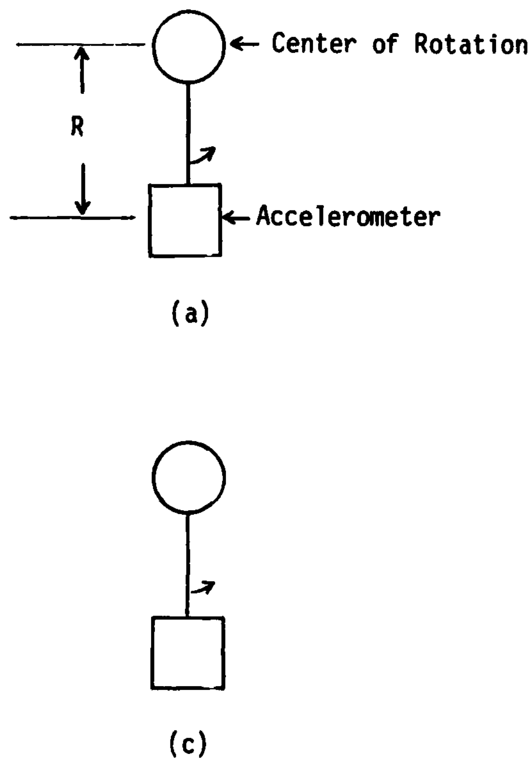


Figure 32. Accelerometer Locations

The accelerometer will be used in a nulling application where the angle θ is small, thus permitting the approximation $\sin\theta = \theta$. Then the expression for the output V_a can be further reduced to

$$V_a = K_a \left(g\,\theta + R\frac{d^2\theta}{dt^2}\right)$$

In Laplace transform form, the above expression becomes

$$\frac{V_a(S)}{\theta(S)} = K_a g\theta\left(1 + \frac{R}{g}\,S^2\right)$$

For $S = j$, it can be seen that the above transfer function is equal to $K_a g$ for much less than $\frac{g}{R}$, and is equal to $-K_a R$ ^{2} for much greater than $\frac{g}{R}$. At $= \pm \frac{g}{R}$, the transfer function goes to zero. This transfer function thus contains a pair of complex zeros at $= \pm \frac{g}{R}$. The leveling loop frequency response measured during the accelerometer placement study, shown in Figure 33, is representative of the complex zeros as derived above.

Form (c) accelerometer placement will result in a similar expression, with both signs negative.

The transfer function of a form (b) accelerometer placement is:

$$V_a = K_a \left(g\sin\theta - R\frac{d^2\theta}{dt^2}\right)$$

which when reduced becomes:

$$\frac{V_a(S)}{\theta(S)} = k_a g\left(1 - \frac{R}{g}\,S^2\right)$$

This transfer function, for much less than $\frac{g}{R}$, is equal to $K_a g$, as in form (a). However, for much larger than $\frac{g}{R}$, the transfer function becomes $+K_a R$ ^{2} and does not change sign as form (a) did. The roots of this expression are real, at $= \pm \frac{g}{R}$. The frequency response of Figure 34 is an example of this configuration.

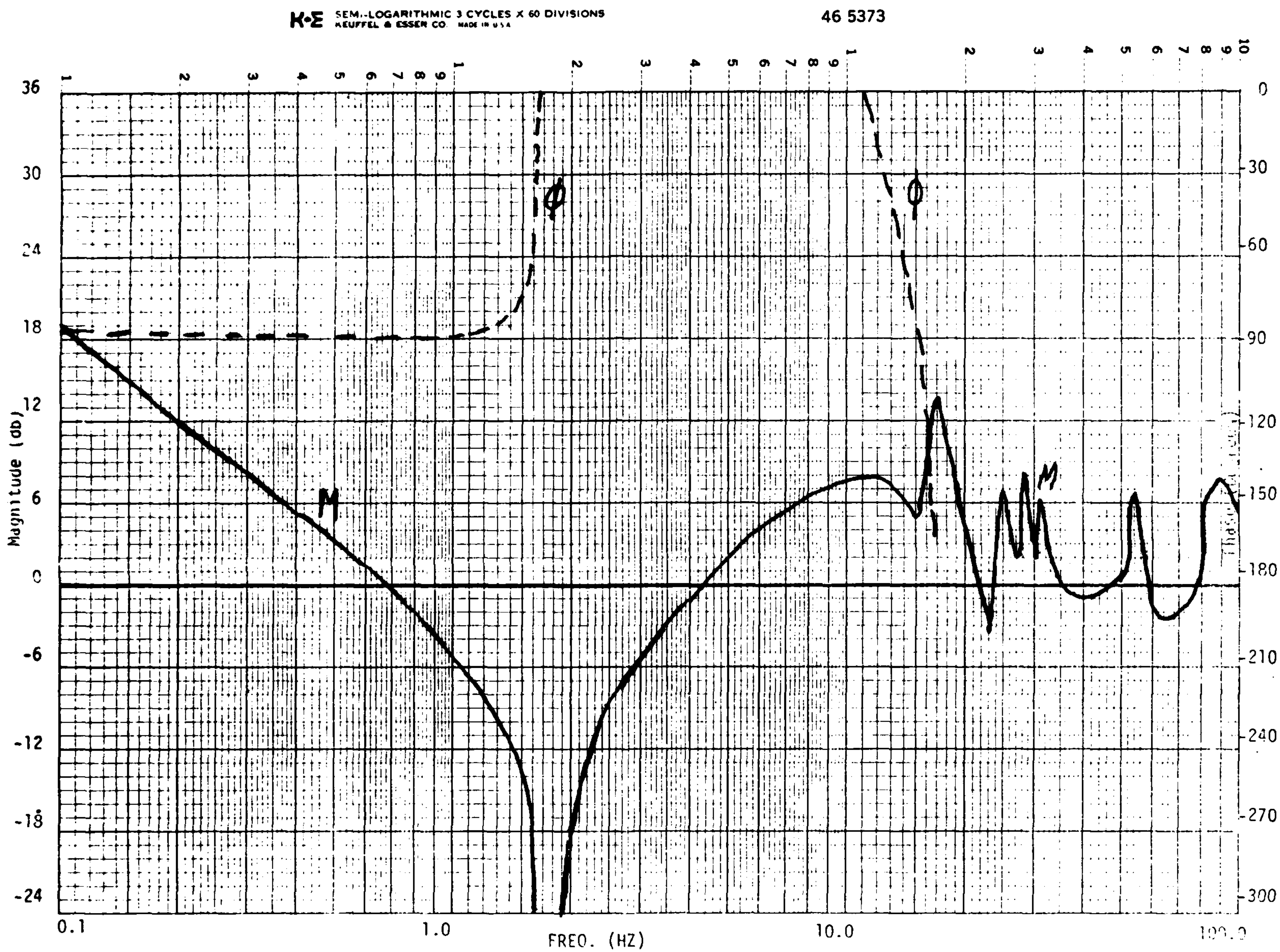


FIGURE 33.M15 QUADRANT CANT AXIS ACCELEROMETER OPEN LOOP FREQUENCY RESPONSE. ACCELEROMETER LOCATED ON A HORIZONTAL PLANE WITH CANT TRUNNION AXIS

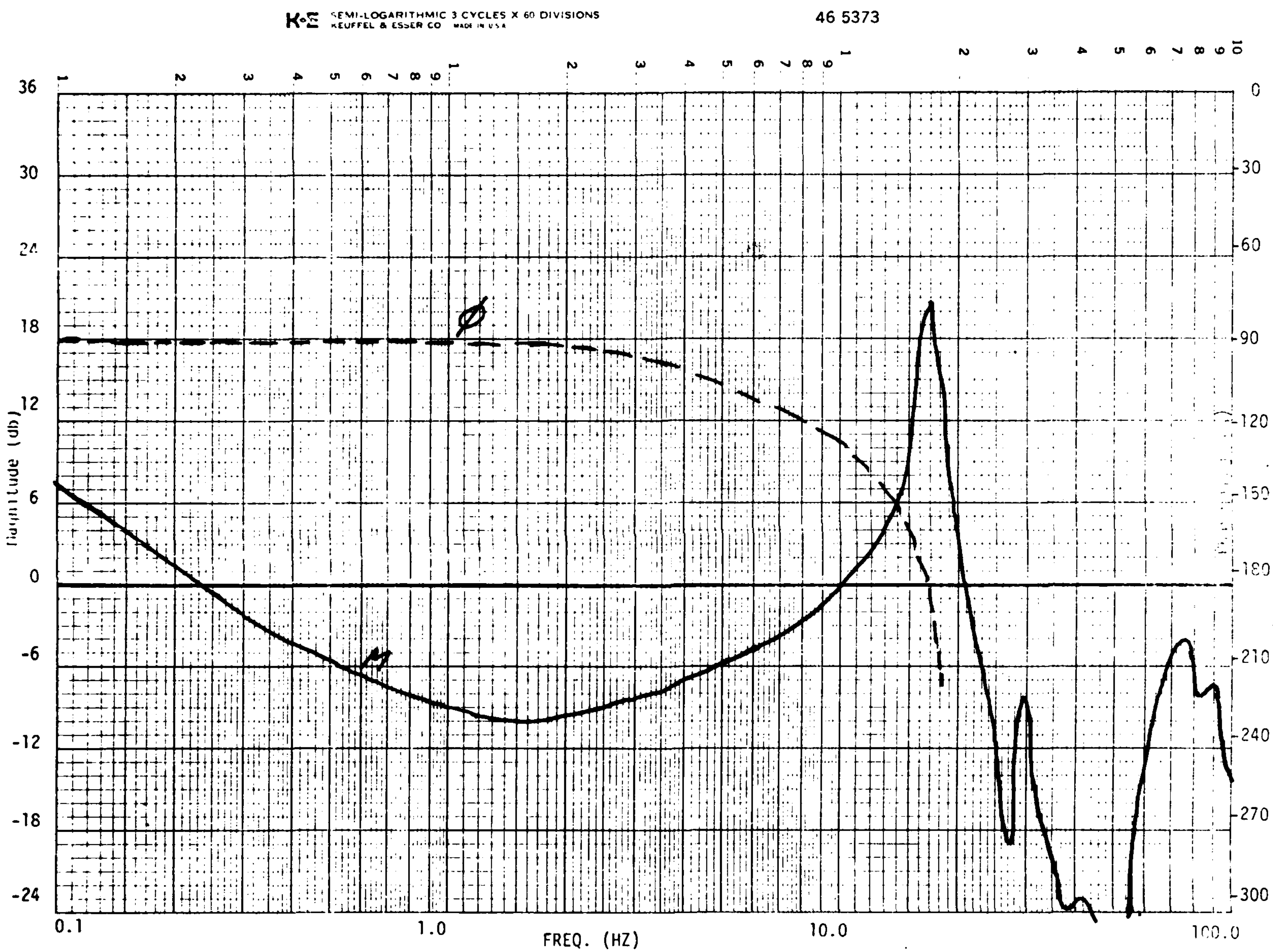


FIGURE 34. M15 QUADRANT CANT AXIS ACCELEROMETER OPEN LOOP FREQUENCY RESPONSE. ACCELEROMETER MOUNTED ON TOP SURFACE OF THE CANT AXIS DRIVE

Form (d) accelerometer placement will result in a similar expression, except with reversed polarity.

Thus, it can be seen that the dynamic response of each servo mount can be affected by the location of the accelerometer. Additional factors affecting servo response include the inertia and compliance of each of the individual servo axes. Thus, a different set of compensation networks is needed to accommodate the dynamic characteristics of each of the five instrument servos.

b. Encoder

In the automatic offset mode, the telescope azimuth servo and the quadrant pitch servo must be driven to a given deflection, as indicated by their respective mechanical counters. To sense the actual deflection of these instruments, digital encoders are coupled to the input shafts through precision gears. It was determined that the input knob scale factor will be 100 mils per turn for the quadrant, and 50 mils per turn for telescope.

The required range of operation is zero to 6399 mils, and it was decided to measure in 0.1 mil increments to achieve good system accuracy, resolution, and stability. Thus, an encoder was needed that had a full count capacity of 0 to 63,999. Vendor surveys revealed that the best method of achieving the required count range in an acceptable size was to use a two-disk encoder.

The encoder used in the AGLS has one disk driven directly by the encoder shaft, and a second disk driven by gears. The direct or high speed disk measures one complete shaft revolution as 1000 counts, while the slow speed disk advances one count for every input shaft revolution from zero to 63. After reaching a count of 63, the slow speed disk advances to zero. The data output from the slow speed disk is synchronized to the data from the high speed disk so that all numbers change state at the same time. As an example, when changing from 599.9 to 600.0, the hundreds digit (5 or 6) will be generated by the slow speed disk, while the other digits (99.9 or 00.0) will be generated by the high speed disk. The synchronizing circuitry will prevent the number 699.9 from being output during the transition from 599.9 to 600.0 if backlash or mechanical errors should exist within the encoder.

The encoder data is accepted by the digital controller unit, and used to determine magnitude and polarity of the correction signal to be applied to the instrument servo. The encoder data is also applied to the digital displays, to indicate actual quadrant and telescope counter readings.

The digital controller unit uses simple arithmetic to subtract the encoder value from the input commanded value, except that the error is checked to determine whether it is larger than 3200 mils. If larger, the error is subtracted from 6400 mils, so that the weapon is always driven to null by the shortest path. The digital value is then converted to an analog signal. All servo compensation and control manipulation is performed by the analog servo subsystems.

c. Infrared Tracker

The infrared tracker detects the flashing XENON light from the GACS reference unit, and provides an analog output voltage proportional to the displacment of the XENON lamp from the center of the tracker field of view. The AGLS tracker shown in Figure 35, consists of three major subsystems; the optics, the sensors and the electronics.

Optics

The optical system utilizes a 50.8 millimeter focal length, 50.8 millimeter diameter Fresnel lens to gather the XENON energy and focus it on to the detector. An optical filter with a passband from 820 to 893 nanometers at 50 percent transmission is placed between the lens and the sensor to reduce the ambient light level while permitting the infrared energy to pass through. Reducing the ambient light level will correspondingly reduce the ambient current through the sensor, and will help to minimize the noise output.

Sensor

The sensing element of the IR tracker is a lateral effect photodiode, shown in a cross section in Figure 36. The scene, including the XENON lamp, is focused on the sensor. For purposes of information, assume that the scene is not present and that only the lamp is visible. When the XENON lamp flashes, the photons from

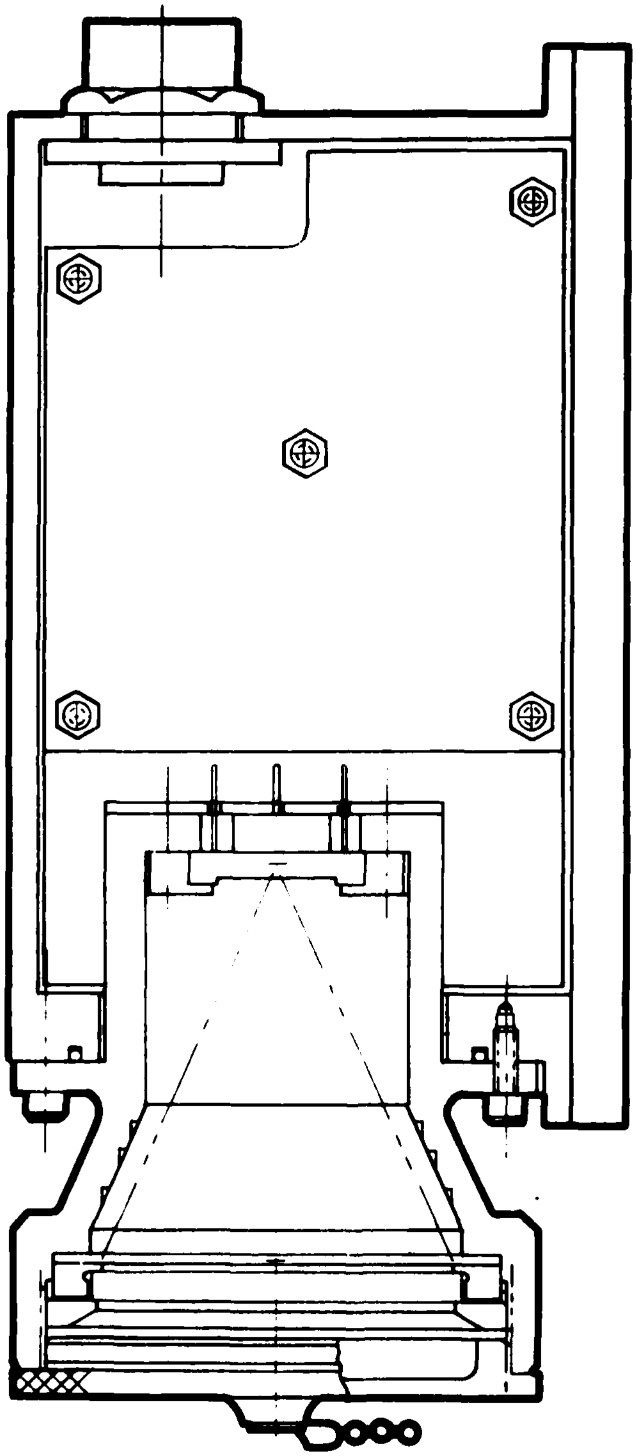


Figure 35. AGLS IR Tracker

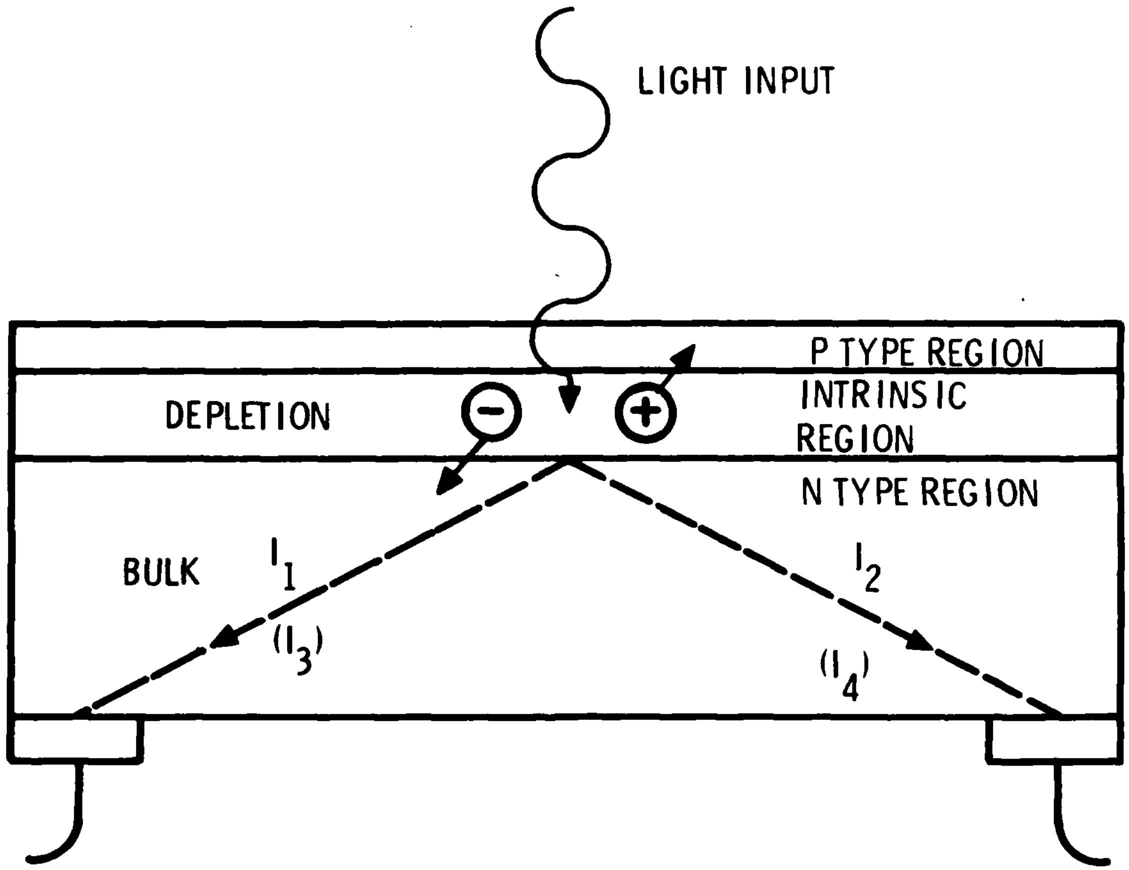


Figure 36. Lateral Effect, Diode

the lamp image generate electron hole pairs in the depletion region. Holes are attracted to the P-region and annihilated in the gold film, while the electrons are injected into the high resistance bulk silicon N-region.

The electrons travel to the two back contacts as a function of the distance to those contacts. For a point image located L distance from one edge of the sensor L distance wide, the current out of the reference contact will be (1 -) I, and the current out of the other contact will be I, where I is the total current. The difference signal will be

$$I_S = I(1-2\ \ )$$

The sum signal is equal to I, the total current, which is equal to the total current generated due to the XENON lamp energy. If the difference signal is divided by the sum signal, the resulting signal

$$\frac{IS}{I} = 1-2$$

is independent of XENON energy.

If the scene is now focused onto the detector, the difference signal will represent the centroid of brightness. The signal from the scene will be present as a steady or slowly changing bias upon which the XENON pulses are riding. Since the scene energy can be orders of magnitude larger than the XENON energy, this bias must be removed.

The electrons generated by the photon energy are generated by a linear process. That is, there is a fixed ratio of optical energy to current flow, and this relationship holds for many orders of magnitude. The gain of the Schottky sensor used in the AGLS tracker is approximately 0.4 amps per watts at the XENON IR wavelength. The linear responsivity of the detector permits detection by removal of the ambient signal.

As discussed above, the difference signal should be divided by the sum signal to normalize the tracker output. However, divider circuits usually exhibit problems in linearity, offset and frequency stability. The Automatic Gain

Control (AGC) technique used in the AGLS tracker is to process both the difference and the sum signals, and to multiply both signals by the same gain. The resulting sum signal is then compared with a reference voltage and the difference is used to adjust the gain of both channels. The sum signal is thereby kept constant over varying range and XENON light output. The difference signal output then becomes

$$Is = Io\ (1-2\ \ )$$

where Io is a constant.

Electronics

The tracker electronic circuitry shown in the block diagram of Figure 37, accepts the two sensor signals, corrects for ambient light and changes in XENON energy, and generates a positive or negative direct current signal proportional to the XENON position. The major circuit elements consist of the preamplifier, the switched AGC, the integrator, the sample and hold amplifier (S/H), the variable AGC, and the output amplifier, as described in the following sections.

Preamplifier

The basic tracker performance limiting characteristic is the fundamental noise of the preamplifier. Discussions with the sensor manufacturer and contractor engineers who have had experience in low noise amplifiers have verified this conclusion. Because of the low power level of the XENON lamp, the effect of the noise will be an increased tracker output noise, or jitter, as range from tracker to XENON lamp is increased. The preamplifier consists of a high gain, high frequency amplifier, and a means of removing the dc and low frequency components of the sensor output. The dc component results from the ambient light of the scene, caused primarily by sunlight. Some of this energy can be reduced by use of the optical bandpass filter. However, there will be steady state light energy in the wavelength of the XENON flashes, so dc removal circuits must still be included.

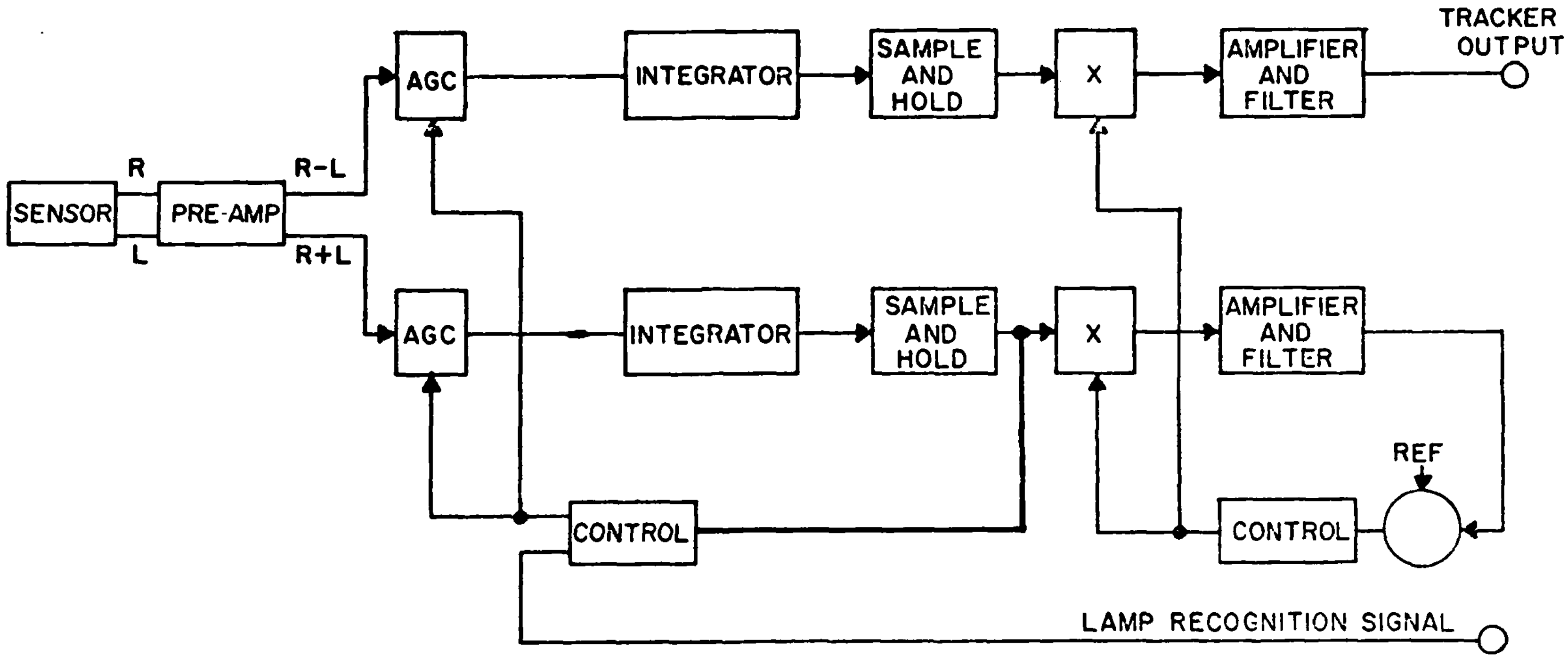


Figure 37. AGLS IR Tracker Block Diagram

By choosing the frequency at which the dc removal circuit is not effective, low frequency rejection is also achieved, thus reducing sensitivity to moving bright spots in the scene. Low frequency rejection also reduced the bandwidth which reduces the overall noise of the preamplifier.

The preamplifier shown in the schematic diagram as Figure 38, consists of two transconductance amplifiers U1 and U2, a difference amplifier U3, and a sum amplifier U4. The transconductance amplifiers change the current outputs of the sensor to voltages which can be further processed. The two outputs are subtracted by U3 to obtain a difference signal, while the two outputs are added by U4 to obtain a sum signal.

The transconductance amplifiers each consist of a second order low-pass active filter. While it might appear that a short pulse would be "lost" in a low pass amplifier, it must be noted that the low pass amplifier will output a pulse equal in volt-time integral to that of the input pulse, times the gain of the circuit. Since the integrator will determine the integral, nothing in the signal is lost by going through a low pass filter. However, reducing the magnitude of the pulse will permit more gain to be used in the preamplifier, thereby, reducing the noise effects of the remaining circuitry. The integrator gain can be reduced to maintain the same overall gain. But the most important improvement is the reduction in bandwidth, which will reduce the value of the root mean square (rms) noise of the circuit.

To accomplish dc rejection, the output of transconductance amplifier U1 is low-pass filtered to remove the XENON pulse signal and detect the remaining steady state or slowly varying signals. The signal is then amplified by the dc rejection amplifier U5 and converted by R1 to a current which essentially cancels the steady state input current from the sensor. The same process is used by amplifier U6 and resistor R21 to remove the ambient signal from the output of amplifier U2.

The outputs of amplifiers U1 and U2 are then subtracted and amplified by difference amplifier U3. Amplifier U4 accepts the U1 and U2 outputs, and amplifies the sum of the two. The sum and difference outputs are then applied to the switched AGC amplifiers.

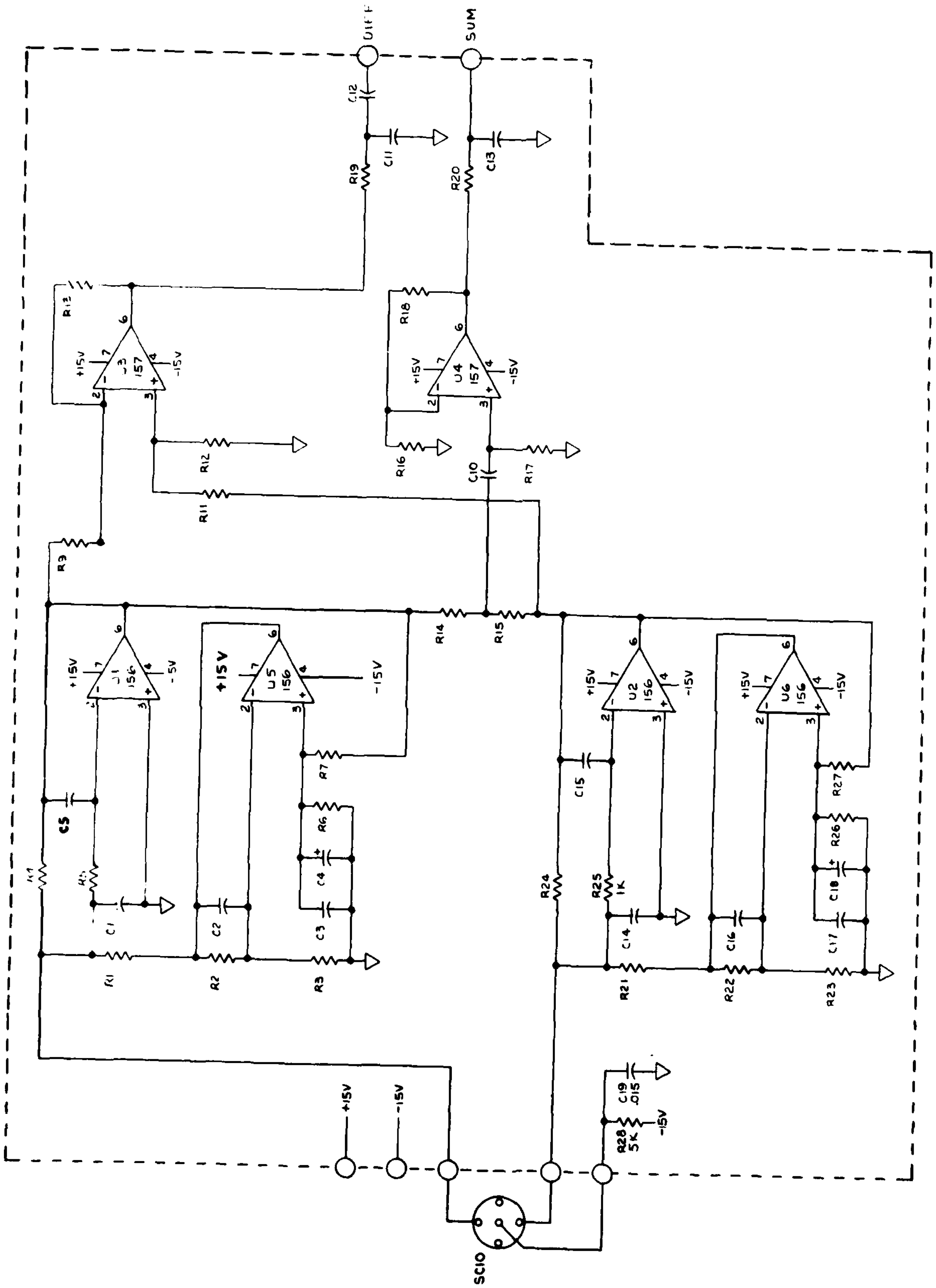


Figure 38. Schematic Diagram Sensor and Preamplifier

AGC Amplifier

Two concepts for AGC were considered. A digitally selected gain control which would change gain by selecting sets of input and feedback resistors, or a solid state multiplier which would multiply the pulse by a dc control voltage.

The digital AGC is more accurate, but would not provide sufficient resolution. For example, a gain range of at least 100 to 1 is needed. If performed in geometrically uniform steps, two amplifier sets with four gain steps per set could achieve the gain range with a step ratio R, where:

$$R = \frac{40 \text{ db}}{16 \text{ steps}} = 2.5 \text{ db}$$

or

$$R = 1.333$$

Thus, an AGC resolution error of $\pm 16\%$ could be expected using the completely digital AGC. The solid state multiplier approach would be a less complex circuit and would have essentially infinite resolution. However, the basic accuracy of the multiplier over a dynamic range of 100 to 1 may affect the tracking of the two AGC amplifiers. It was then decided to utilize both switched and continuously variable AGC, to benefit from the advantages of each approach. A total gain span of 1X to 100X is needed to a 50 to 500 meter range. To accommodate such a large span, a switched gain amplifier having gains of 1, 4, 16 and 64 is the primary gain control amplifiers, and an analog divider will be used to achieve better resolution over a limited span of approximately 6 to 1.

The switched amplifier, shown in Figure 39, uses a programmable amplifier (PRAM) consisting of four preamplifiers which are selected by digital control of the two address lines. Each preamplifier is connected to a resistor network to provide a specific gain. A given digital value on the two address lines will select a certain preamplifier, and thus provide the desired gain. The digital values are established by a separate control circuit through interaction with the linear AGC amplifier.

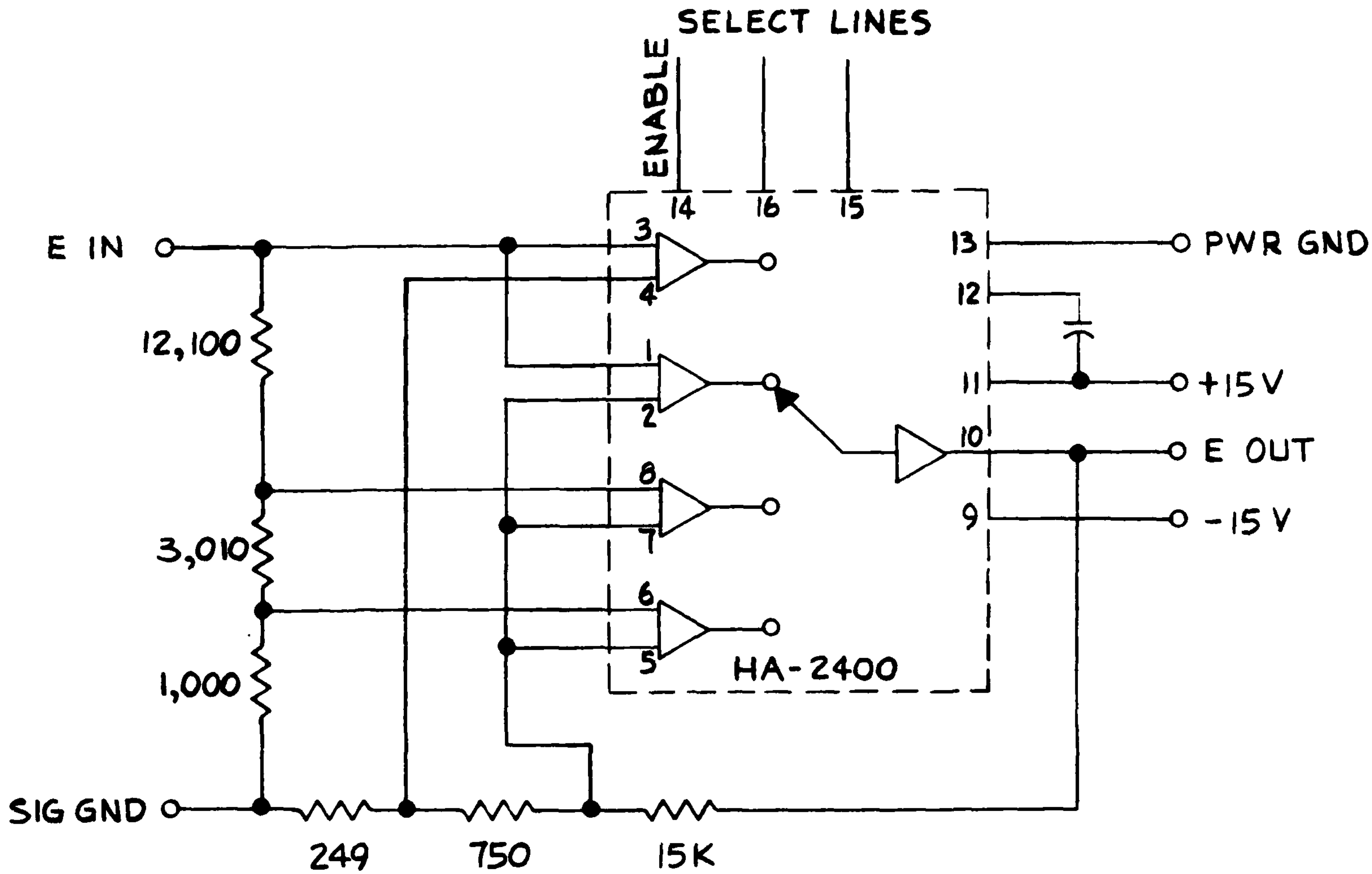


Figure 39. Gain Control Amplifier

Integrator

The following description of the integrator for the sum channel, shown in Figure 40, will also apply to the difference channel, since the two circuits are identical.

The sum signal is applied through capacitor C1 to the quad switch S1. Prior to the start of the XENON flash, terminals 1 and 2 are shorted to connect the output side of C1 to the ground. This causes C1 to charge to the direct current (dc) level of the sum signal. This voltage will then be subtracted from the video signal, effectively removing any remaining dc bias from the input signal.

To ensure that the integrator, U1, begins its integration at zero volts, terminals 8 and 9 are shorted, providing feedback around the integrating capacitor, C2.

When a XENON flash occurs, the sum signal initiates the timing and logic necessary to operate the switch drivers. The first step is to open contact 1 to 2, and 8 to 9, and to close contact 3 to 4. The signal then flows through to the integrator input resistor R1, and the signal is integrated for the selected time duration. The second step is to open switch contacts 3 to 4, and close contacts 10 to 11, thus applying a zero input to the integrator for a hold period. During this time, the integrator output is sampled by the sample-and-hold circuit.

After the sample has been stored, the contacts 10 to 11 open, contacts 8 to 9 close to reset the integrator, and contacts 3 to 4 close to restore the dc input level. The circuit is now ready to accept another input pulse.

Sample and Hold Amplifier

A sample and hold amplifier is used to accept the integrator output and to store this signal until the signal from the next pulse has been measured. The sample and hold output is then a continuous dc signal, with step transitions if the tracker should be moving with respect to the reference unit.

AD-A097 521   HONEYWELL INC HOPKINS MN DEFENSE SYSTEMS DIV   F/G 19/6

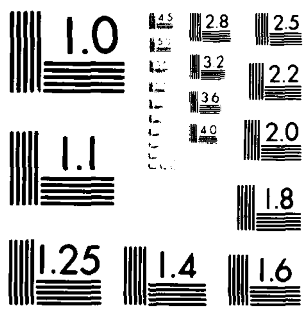
AUTOMATED GUN LAYING SYSTEM FOR SELF-PROPELLED ARTILLERY WEAPON--ETC(U)

MAY 80   E E LEHTOLA, K A HERZING   DAAA09-76-C-0284

UNCLASSIFIED   NL

2 OF 5

AD A097521



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

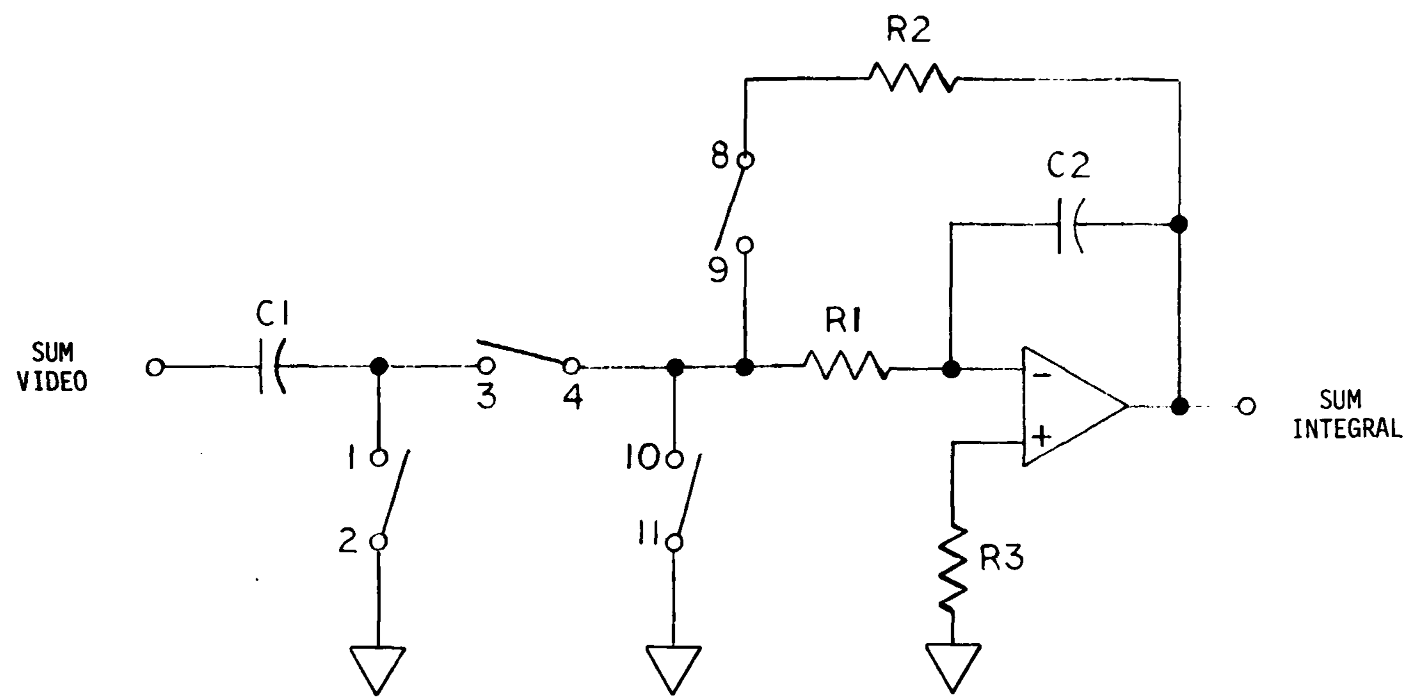


Figure 40. Agls IR Tracker Integrator

Continuous AGC and Output Filter

The sum sample and hold output is then applied to a pulse-modulated switch which either passes the signal when ON, or blocks the signal when OFF. The ON to total period ratio is under control of a pulse width modulator. The average output of the switch is then

$$e_{sw} = \frac{ton}{T} x\ e_{S/H}$$

where $e_{s/w}$ = average switch output voltage

$e_{S/H}$ = sum sample and hold output

ton = on time of the modulated switch
T = period of switch frequency

The switch output is filtered and subtracted from a reference voltage, and the difference is amplified and applied to a pulse width modulator. Since the pulse width modulator controls the switch conduction time ratio, a closed loop exists to maintain the sum signal at a constant value. The same conduction time ratio is then used to modulate the difference sample and hold amplifier, to adjust the difference signal for changes in XENON intensity. The switched multiplier controls over a dynamic range of about 6.0 to 1.0.

The control scheme is to amplify both the sum signal and the difference signal by the same gain, and to integrate both signals by identical circuits. The sum sample and hold signal is then monitored if it is lower than 1.0 volt, the digital address is advanced one count increasing the gain by a factor of 4.0. After a short time delay, the sum signal is rechecked. If it is still less than 1.0 volt, the count will be advanced again, and rechecked. The process will continue until the sum signal is in the acceptable region, or until maximum gain has been reached. The reverse process is applied if the sum sample and hold signal is too high. In this case, if the signal is more than 6.0 volts, the gain address is reduced one count, which causes the switch AGC amplifier to reduce the gain by a factor of 4.0. The range of acceptable input voltage has been established as greater than 4 to 1 to avoid oscillations which might otherwise occur if the sample and hold output should be close to the threshold value, and thus alternately advancing and retracting the gain by one count.

The difference sample and hold signal, after being multiplied by the AGC conduction ratio, is then amplified and filtered by a 5 Hertz active filter. The resulting tracker output is then transmitted by wiring harness to the Instrument Controller Unit.

Sequencer

A sequencer circuit is included to trigger the integrators and sample and hold amplifiers in response to the leading edge of the XENON pulse. A digital output from the sequencer also indicates when no pulses are being detected. This one-bit output is used by the Digital Controller Unit to determine what control mode should be permitted or implemented.

A timing diagram for the tracker sequencer is shown in Figure 41. Since the tracker, mounted in the M109, does not have any electrical connection to the GACS reference unit, the tracker must synchronize itself by detecting the leading edge of the XENON pulse. This is accomplished by applying the sum amplifier pulse output to the comparator and then to a series of six monostable multivibrators.

Time T_0 is triggered by the sum amplifier output pulse, which then triggers T_1. During time T_0 the integrator reset switch 8-9 and the input dc restoration switch 1-2 are opened, and they are both closed during the remaining time. During time T_1, which envelopes the XENON pulse, the switch 3-4 is closed to apply the sum and difference pulses to their respective integrators. At the end of time T_1, timer T_2 is triggered to actuate the switch 10-11 which holds the integrator input to zero, and to enable the sample and hold amplifier. After T_2 goes low, the sample and hold switch opens, but the integrator output is held to assure no S/H loss while the switch is opening. When T_0 goes low, the integrator resets, and the dc restoration of the input coupling capacitor is initiated. Timer T_S, also triggered by the leading edge of T_0, is used to block further trigger inputs to timer T_0 for approximately 4 milliseconds. This will prevent the tracker from being triggered by bright flashes that are not synchronous with the 160 Hertz XENON lamp.

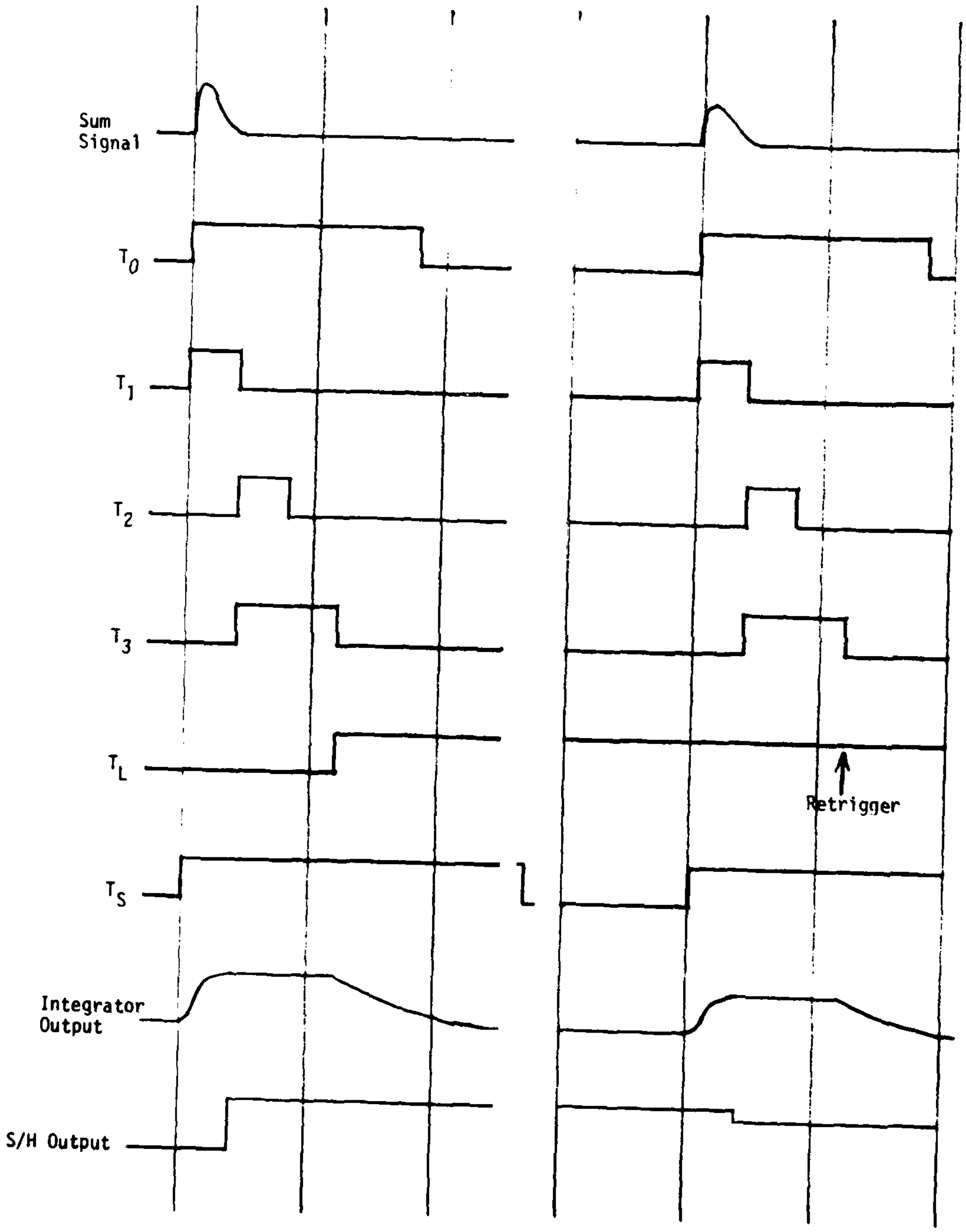


Figure 41. Integrator Timing Diagram

Timer T_3 is triggered from the trailing edge of T_1, and is used to clock the address counter for the switched AGC amplifier. The trailing edge of the T_3 pulse triggers a timer T_L, which is 10 milliseconds long. Since timer T_L is retriggerable, its output will stay high if it is triggered before 10 milliseconds have elapsed. Since the GACS reference units (RU) pulses are 6.25 milliseconds apart, T_L will stay high as long as XENON pulses continue to be present. If XENON pulses should cease, timers T_3 and T_L will form a free-running clock to permit changes in AGC address, and ultimately to drive the switched AGC amplifier to maximum gain.

System Power Distribution

Electrical power for the AGLS components is provided by the vehicle +28 volt dc system through a filter inductor as shown in Figure 42. A battery with charger and disconnect circuitry is provided to maintain input voltage during electrical transients caused by hydraulic pump cycles, slip ring noise and engine starting. Another battery provides steady power for the GACS power supply.

The system power supply accepts power from the voltage support battery, and employs a switching regulator inverter and rectifiers to provide regulated +28 volt dc power for the digital components. A second regulator and inverter provides +32 volt and -32 volt power for the instrument controller unit. Operation of the system power supply is described in Section IV.

5. Gun Alignment Control System Theory of Operation

In laying indirect fire artillery weapons systems we are concerned with the azimuth angle from an arbitrary aiming point to the target. As shown in Figure 43, this angle, defined as the NORMAL angle, has two components; the target grid bearing or azimuth angle and the reverse grid bearing or reference angle, from the weapon sight to the aiming point. The azimuth angle is specified by the Fire Direction Center while the reference angle is determined by the weapon laying equipment.

Gun Alignment Control System (GACS) provides a simplified method of communicating the weapon position-dependent reference angle to the weapon. The GACS

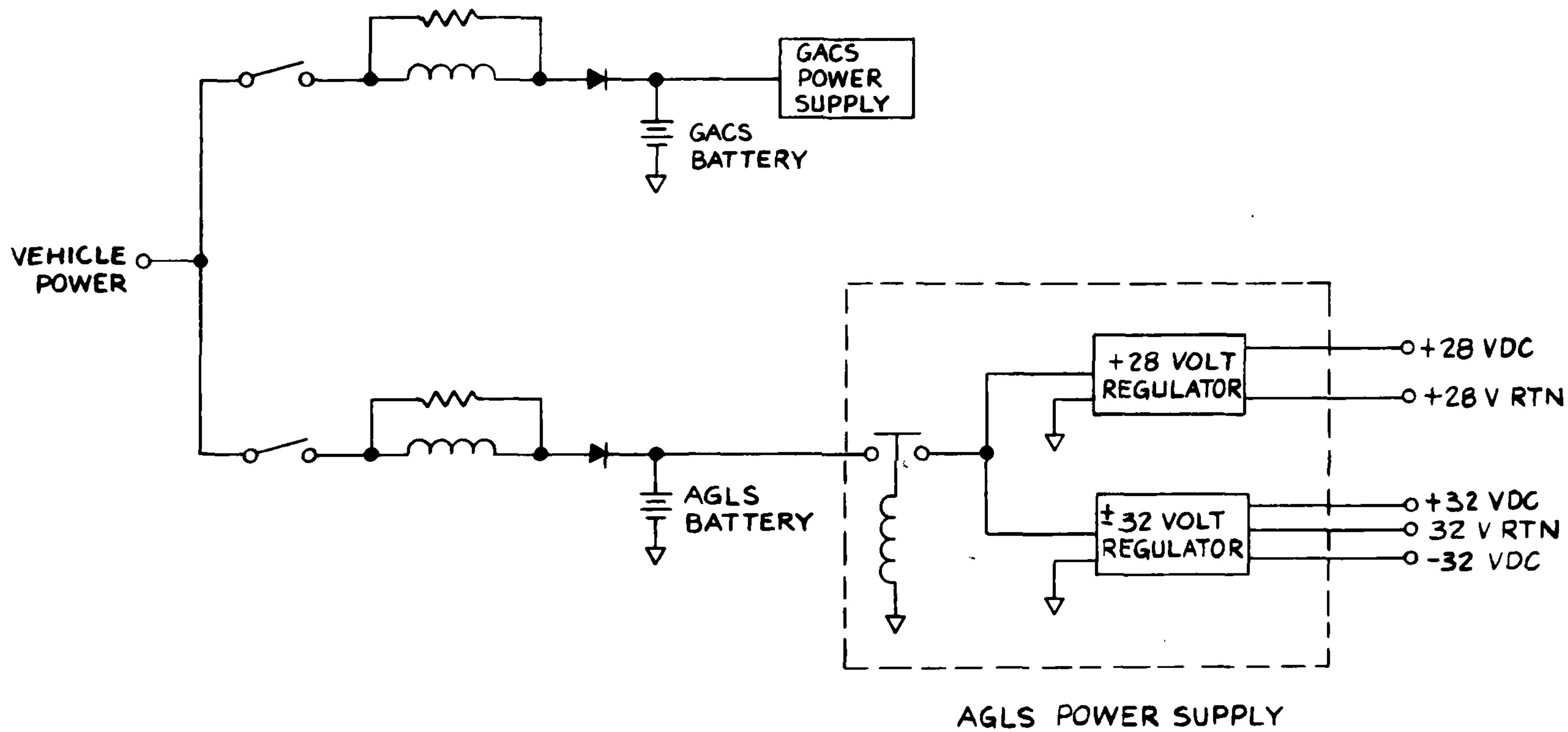


Figure 42. AGLS/GACS Power Distribution

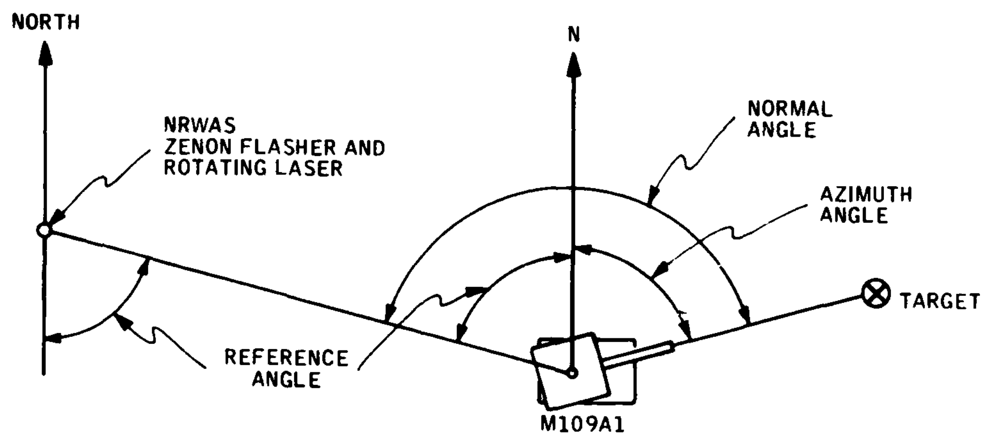


Figure 43. Fire Control Geometry

reference unit contains a solid state laser telescope which emits a narrow beam of infra-red radiation approximately 0.3 milliradians in width and 300 milliradians in height and an XENON lamp which radiates infra-red energy omni-directionally in the horizontal direction and through 300 milliradians in the vertical direction. The laser telescope rotates counterclockwise at a rate of approximately one revolution per second. At each 40 mil increment of laser rotation a pulse flashes the XENON lamp. In addition, each time the rotating laser passes through the South direction, the XENON tube emits a pair of closely spaced pulses of radiation, thus providing identification of this direction. The GACS receiver is able to receive and identify the XENON and laser pulses, and through the GACS gun unit electronic logic circuits they are processed along with the azimuth angle specified by the Fire Direction Center to yield the NORMAL angle.

In manual operation, the GACS receiver is manually aligned to the GACS reference unit. An electronic counter in the GACS gun unit, located at the weapon, remains inhibited until the pulse-pair indicating that the laser is passing through the South direction is received by the GACS receiver, whereupon it commences to count the regularly spaced XENON flashes. This process continues until the narrow laser beam is intercepted by the GACS receiver which immediately stops the count. An interpolating circuit in the GACS gun unit then calculates the angular position at which the laser pulse was received, to the nearest mil between the 40 mil spaced XENON pulses. This angular position is the reference angle. The GACS gun unit then sums this reference angle with the azimuth angle specified by the Fire Direction Center and displays this sum as the NORMAL angle.

This NORMAL angle is also available at the GACS gun unit as an electrical output in parallel binary coded decimal form. In the original AGLS/GACS implementation, the GACS normal angle was utilized as the commanded input to the M117 panoramic telescope, and a digital shaft encoder reading the azimuth counter value from the pantel provides the feedback or actual value for the telescope servo.

B. GACS Interface

The Gun Alignment and Control System as previously integrated with the AGLS, provides the following functions.

1. Azimuth reference, through an off-board IR/LASER Reference Unit and an on-board IR receiver mounted on the Automated M-117 panoramic telescope.

2. One-way commanded data transmission, by means of manually-operated switches on the Command Post Unit, through radio or field phone lines to the Gun Unit.

3. Addition of the Reference Angle to the Commanded (azimuth) angle to obtain a Normal Angle in local coordinates.

4. Data display of azimuth, elevation, and fuze setting commands.

The existing GACS system proved to be adequate for purposes of demonstrating and evaluating the Automated Gun Laying System; however, several operational deficiencies were uncovered during acceptance tests at Honeywell and U.S. Army Field Artillery Board tests at Ft. Sill. In the order of frequency of occurrence, these were:

1. The GACS power supply would periodically fail when subjected to transient supply voltage conditions during M-109 system operation. This failure would cause loss of all GACS functions.

2. When the line of sight from Reference Unit to IR Receivers was interrupted, the GACS would provide an angle of either 0 or 80 mils, thus causing erroneous gun laying. After the sight line was reestablished, the GACS was also to provide the correct reference angle.

3. The data communications from FDC to howitzer would intermittently fail, with no indication of the source of the fault.

4. Presentation of azimuth data in NORMAL angle form proved to be confusing to the gun crews, since they were accustomed to numbers based on 3200 being the azimuth of lay.

The Amendment to the Scope of Work required additional data communications capability as listed below:

1. An electronics interface at the Fire Direction Center (FDC) with its PDP-11/34 computer.

2. An electronics interface in the howitzer, with the capability to transmit back to the FDC computer all data from the AGLS controller data bus.

3. Additional electronics interfaces to the DR-810 Muzzle Velocity Radar, Electronic fuze setter, and propellant temperature measuring system.

Since these data communications requirements were beyond the capabilities of the existing GACS components, new communications units were needed at the FDC and at the howitzer. It was decided to provide a new power supply at each location, because the existing GACS supplies were subject to breakdown, and because their output current capability and voltage regulation were not known to either the contractor or the ARRADCOM project personnel.

Replacement of the GACS Gun Unit required that the Reference Angle computation feature be provided in the new Vehicle Communications Unit. The GACS IR Receiver was retained, since it had not appeared to cause any performance problems in the AGLS test phase. The GACS Reference Unit was also used in its existing configuration.

The modified AGLS/COMM System block diagram shown in Figure 44 includes only one interface to the GACS; this is the electrical interface between the Vehicle Communications Unit and the GACS IR Receiver. Electrical power is provided to the IR Receiver, and two pulse signals lines are output from the IR Receiver.

IR Receiver Interfaces

The IR Receiver provides two pulse signals, triggered by the XENON and LASER emissions of the Reference Unit, which are used to determine the Reference Angle from the weapon to the Reference Unit. The XENON pulse signal consists of a

AGLS-COMM BLOCK DIAGRAM

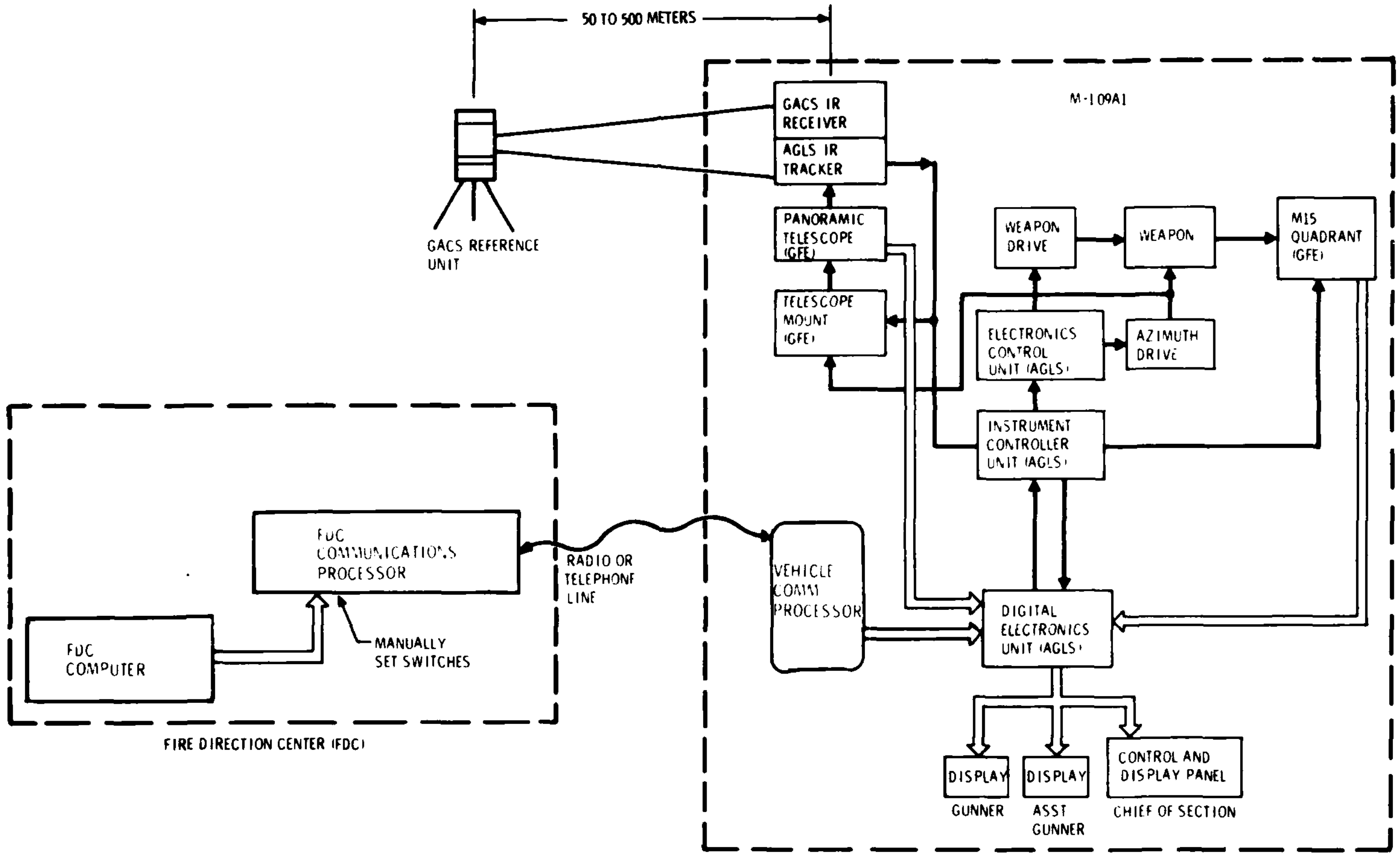


Figure 44

string uniformly spaced pulses with a 6.25 millisecond spacing. There is an additional pulse, occurring once for every 160 of the uniformly spaced pulses, relating to south. The extra pulse is spaced 2.5 milliseconds before the normal pulse, indicating that the next pulse should be counted as ZERO.

The pulse from the LASER detector appears as a separate signal, and occurs at any timing with respect to the XENON pulses. The only constraint on the LASER pulse is that only one can occur per revolution of the reference unit. Both pulses are at TTL levels, and are output by open collector drivers in the IR receiver. Thus pull-up resistors are needed at the reference processor input.

The power required by the receiver was determined to be as follows:

Pin E	+12.6 volts	90 milliamperes
Pin G	-15.5 volts	38 milliamperes
Pin D	+80 volts	0 to 2 milliamperes

This power was supplied by the power supply in the Vehicle Communications Unit.

C. FDC Computer Interface

Data supplied by ARRADCOM and Digital Equipment Computation were used to define the data interface between the PDP-11/34 and the FDC Communication Processor (FDCOM). The purpose of FDCOM was to relieve the PDP-11/34 from having to perform routine communication tasks assoicated with data transfers between the vehicle and FDC. These routine tasks include formating the message, control of the radios, executing message exchange rules (protocol) and performing error detection and correction via a retransmit sequence. Insofar as the PDP-11/34 is concerned all it expects to do is deliver gun orders and commands to FDCOM and receive accurate vehicle-originated data in return. The hardware interface between the processors used the PDP-11/34 DR11-L and DR11-M general purpose UNIBUS interface connected to the FDCOM M6820 parallel interface adapter. These devices were configured to exchange data in a bit parallel, character serial mode using ASCCII formatted characters and the DEC recommended handshake protocol. (Reference DEC Users Manual EK-DR11L-OP-001.) The messages expected from the PDP-11/34 included:

a) Gun Orders in the form:

$\$ * \$ \ F \ O_{\Delta} C_{\Delta} DDDD_{\Delta} FFF_{\Delta} \ \emptyset EEE \ \overset{E}{\underset{T}{(0)}}$

where: C = Charge

D = Deflection

F = Fuze Time

E = Elevation

b) Fire Command in the form:

$\$ * \$ \ FC \ \overset{E}{\underset{T}{(0)}}$

c) Check Fire Command in the form:

$\$ * \$ \ CF \ \overset{E}{\underset{T}{(0)}}$

d) Data Request Command in the form:

$\$ * \$ \ DR \ \overset{E}{\underset{T}{(0)}}$

e) End of Mission Command in the form:

$\$ * \$ \ EM \ \overset{E}{\underset{T}{(0)}}$

Messages returned from FDCOM to the PDP-11/34 include:

a) Gun Order acknowledge in the form:

$1110 \ DDDDEEEE \ FFFC \ \overset{E}{\underset{T}{(0)}}$

where: D = Echo-back of deflection

E = Echo-back of elevation

F = Echo-back of fuze time

D = Echo-back of charge

b) Check Fire acknowledgement in the form:

0111 DDDDEEEE FFFC $\overset{E}{\underset{T}{(0)}}$

c) Fire Command acknowledgement in the form:

1101VVVVVTTTTTTEC(5)EA(5)EE(5)AC(5)AA(5)AE(5)LLM $\overset{E}{\underset{T}{(0)}}$

where: V = Velocimeter Reading
T = Propellant Temperature
EC = AGLS Elevation, Command
EA = AGLS Elevation, Actual
EE = AGLS Elevation Error
AC = AGLS Azimuth, Command
AA = AGLS Azimuth, Actual
AE = AGLS Azimuth Errors
L = AGLS Level Status
M = AGLS Mode

d) Ready response acknowledgement in the form:

1011VVVVVTTTTTTEC(5)EA(5)EE(5)AC(5)AA(5)AE(5)LLM $\overset{E}{\underset{T}{(0)}}$

e) Data response output in the form:

0000VVVVVTTTTTTEC(5)EA(5)EE(5)AC(5)AA(5)AE(5)LLM $\overset{E}{\underset{T}{(0)}}$

Several procedural rules were established to control the sequence of operations, namely:

a) When multiple gun orders are transferred to FDCOM, the latest one should be retained for transmission to the vehicle and previous ones discarded.

b) If a gun order update is transferred to FDCOM prior to generation of the "ready request" the gun order should be transmitted to the vehicle upon receipt of the previous gun order acknowledgement.

c) If a check fire command is transferred to FDCOM prior to generation of the "ready request" the check fire should be transmitted to the vehicle upon receipt of the previous gun order acknowledgement.

D. Vehicle System Interfaces

The original interface between the GACS Gun Unit and the AGLC processor was via a character serial, bit parallel port which used a BCD data format (Figure 45). The gun unit provided both a data source for FDC commands and a processor which adjusted the azimuth angle by the measured reference angle to produce normal angle commands to AGLS. Because of reliability problems with the GACS system and the desire to provide bi-directional communication between the howitzer and FDC for HELBAT VII, a replacement to the GACS gun unit was required. This replacement system had to provide communication control, reference angle processing, additional Chief of Section (COS) controls and interface to a projectile velocimeter, propellant temperature monitor and electronic fuze setter. The system was dubbed the vehicle communication processor (VECOM) and interfaced with the AGLS as shown in Figure 46. An analysis of the interface characteristics of all subsystems connected to VECOM was performed. Data supplied by other contractors and cognizant government agencies was used to develop the I/O configuration of VECOM. In addition, our own analysis of the AGLS and reference unit processor (RUP) needs defined those interfaces; as shown in the following list:

- Propellant Temperature -- The initial intent was to interface with a real time electronic thermometer system furnished by Don Lince at Human Engineering Laboratory. The interface was subsequently redefined to be a temperature entry set of thumbwheels which allowed entry of $\pm$ temperatures of 5 digits with resolution to $0.1^{o}F$. The data format was parallel BCD with a multiplexer used to provide character serial transmission (and reduce the number of wires to the portable entry box).

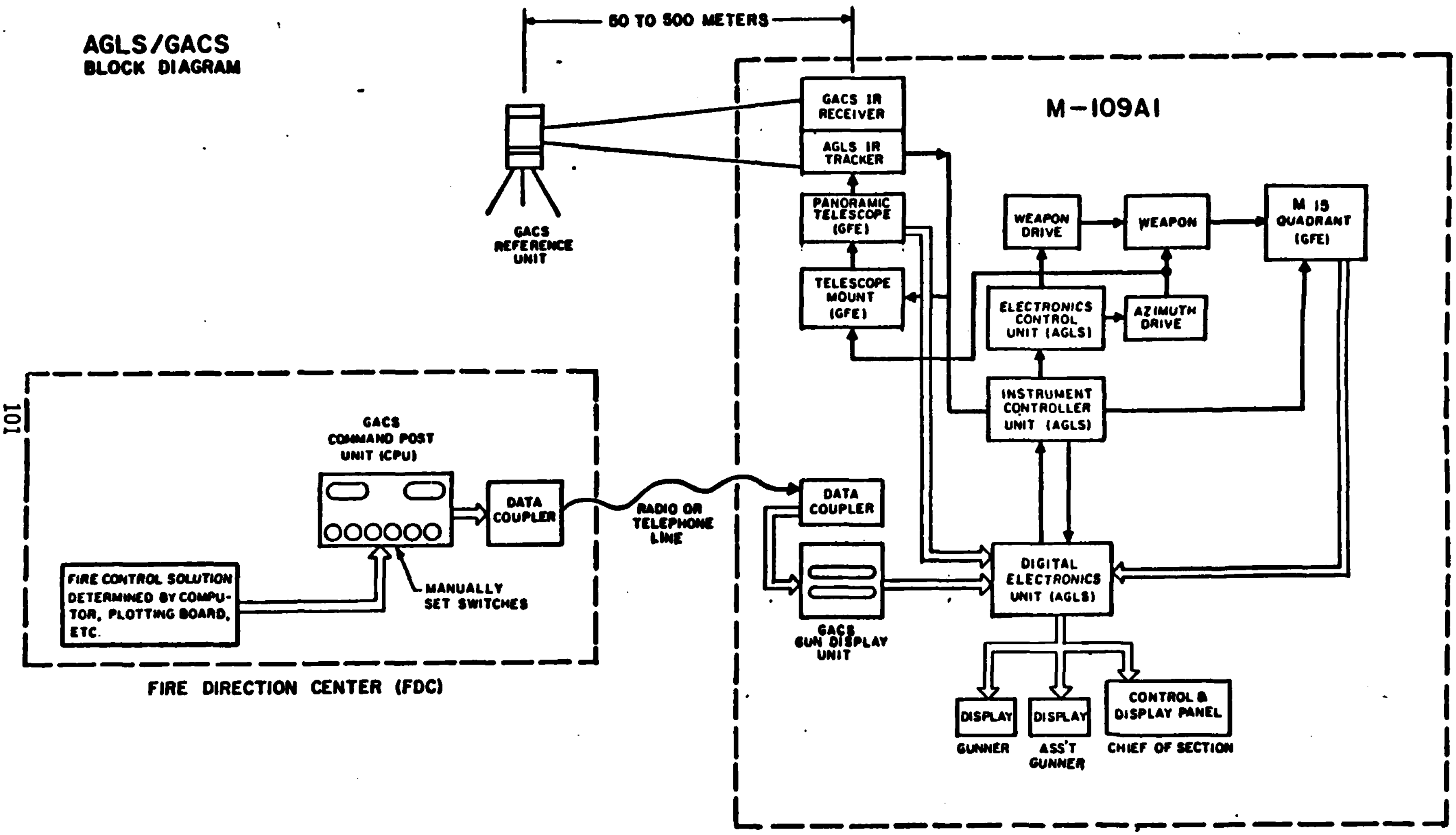

AGLS/GACS
BLOCK DIAGRAM
50 TO 500 METERS
GACS REFERENCE UNIT
M-109A1
GACS IR RECEIVER
AGLS IR TRACKER
PANORAMIC TELESCOPE (GFE)
TELESCOPE MOUNT (GFE)
WEAPON DRIVE
WEAPON
M 15 QUADRANT (GFE)
ELECTRONICS CONTROL UNIT (AGLS)
AZIMUTH DRIVE
INSTRUMENT CONTROLLER UNIT (AGLS)
DATA COUPLER
GACS GUN DISPLAY UNIT
DIGITAL ELECTRONICS UNIT (AGLS)
DISPLAY
GUNNER
DISPLAY
ASS'T GUNNER
CONTROL & DISPLAY PANEL
CHIEF OF SECTION
RADIO OR TELEPHONE LINE
GACS COMMAND POST UNIT (CPU)
DATA COUPLER
MANUALLY SET SWITCHES
FIRE CONTROL SOLUTION DETERMINED BY COMPUTOR, PLOTTING BOARD, ETC.
FIRE DIRECTION CENTER (FDC)

Figure 45

AGLS — COMMUNICATION SYSTEM

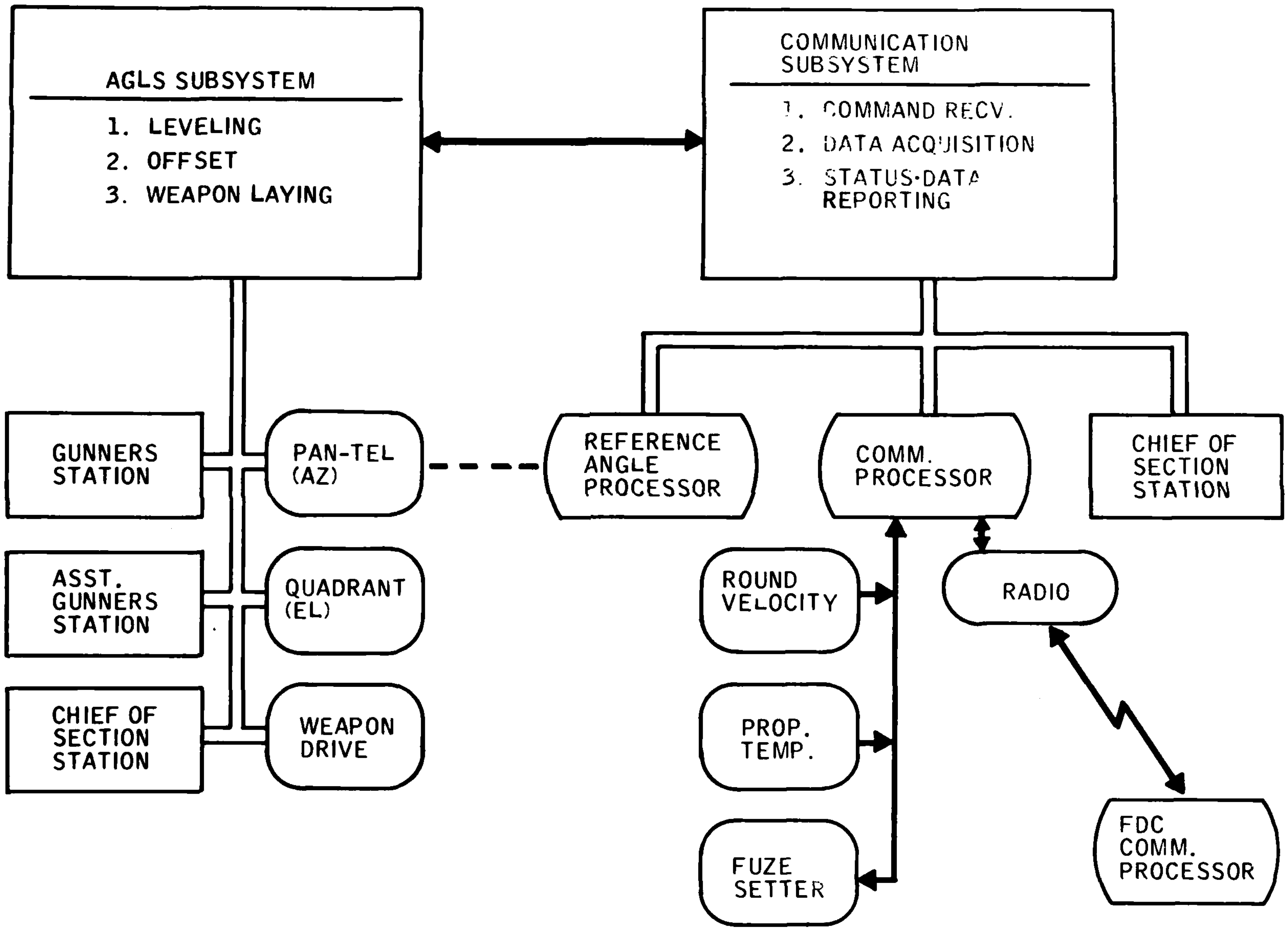


Figure 46

- o Projectile Velocimeter -- The Lear Siegler XM90 Muzzle Velocity Radar (MVR) was used to acquire real time projectile velocity data. An RS-232 compatible serial data output, operating at 300 baud was provided, with data output in the following sequence:

  LF, SW, DEF, MSD, MSD_2, MSD_3, MSD_4, LSD, CR

  where: LF = ASCII line feed character
  SW = Front panel switch position code
  DEF = Display Error Flag (E = Error, F = Good)
  MSD = Most Significant Digit
  LSD = Least Significant Digit
  CR = ASCII carriage return character

  In addition to the received data line (R_x) a data set ready signal (DSR) was provided from the MVR. This signal was tied to the data carrier detect (DCD) input of the velocimeter input serial port VECOM and was used to signal the presence (or absence) of the velocimeter. The MVR, which was used in the signal shot capture mode, required a memory clear reset signal prior to acquiring new data. The model used for the TB-I program did not have the capability for a remote reset function, hence it was necessary to instruct the crew (loader) to reset the velocimeter before the round could be measured. Since the data was output continuously from the MVR it was necessary to design the software to:

  a) Test to see if velocimeter present; if not zero fill the buffer.

  b) Acquire the data "on the fly" by seeking start and stop synchronization from the LF and CR characters respectively.

  c) Test to see if velocimeter has been reset prior to "ready acknowledge".

- o Electronic Fuze Setter -- The specification for this HDL furnished device required fuze data in the form of a frequency shift keyed (FSK)

(2225 Hz mark, 2025 Hz space) 16 character ASCII message. The message format was:

$C_{\Delta}\ DDDD_{\Delta}\ \ TTT_{\Delta}\ \ EEE\ \ \#1$

where: C = Charge
D = Deflection
T = Fuze Time in 0.1 secs
E = Elevation
#1 = Gun Number (TB-1)

This data was to be output to the setter upon receipt of a valid gun order to VECOM from the FDC.

o Reference Unit Processor -- This processor was configured to operate asynchronously from VECOM and provided an updated reference angle to the AGLS system. In order to minimize the latency introduced into either the RUP or VECOM processor, a BCD character serial bit parallel communication format was utilized betweeen parallel interface adapters (PIAS) in the two systems. A foreground communication package was used to exchange data and the control was provided by a PIA-PIA handshake routine. The data format consisted of:

a) A VECOM request for data $\overset{S}{\underset{X}{(T)}}\ \ \overset{E}{\underset{Q}{(N)}}\ \overset{E}{\underset{X}{(T)}}$

where: $\overset{S}{\underset{X}{(T)}}$ = ASCII STX character

$\overset{E}{\underset{Q}{(N)}}$ = ASCII ENQ character

$\overset{E}{\underset{X}{(T)}}$ = ASCII ETX character

b) A RUP data return message:

```
 S             E
(T) RRRR DDDD (T)
 X             X
```

where: R = Reference angle computed by RUP in ASCII character
D = Value to be displayed as function of "mode" switch

Mode	D
Normal	Azimuth from FDC
Boresight	3200
FDC	Azimuth from FDC
Base Deflection	Azimuth from FDC

c) A VECOM gun order data update

```
 S        E
(T) DDDD A (T)
 X        X
```

where: A = AGLS Mode Code

o AGLS Processor -- The primary data to be exchanged with this processor include:

a) Gun Order inputs relayed from FDC via VECOM

b) Operational status from the VECOM control panel

c) Reference angle relayed from RUP via VECOM

d) Elevation command, actual and error data from AGLS displays

e) Azimuth command, actual and error data from AGLS displays

f) Active elevation and azimuth commands from AGLS

g) Command Mode Status from AGLS, i.e., Normal, Base Deflection, Boresight, Base Deflection Set, Base Deflection Clear

h) Level status from AGLS displays

i) AGLS Mode, i.e., Auto Level, Auto Offset, Full Auto

j) Local Mode, i.e., Base Deflection Preset, Auto update enable

Data were transferred via a bidirectional, RS-232 serial link which employed ASCII formatted data and operated asynchronously at 1200 band.

The protocol employed a request to send (character "T") from VECOM which initiated transfer of the data buffer from AGLS. Parity, overrun and framing were checked upon receipt and a retransmission requested in case of error. A character "R" was sent from VECOM to signal transfer of data to AGLS. Again the data was tested on the receiver and to verify accuracy and a character "X" sent if the tests failed; if the test passed the entire message was echoed-back for verification.

- RT-524/VRC Command Radio Interface -- This VECOM interface analysis consisted of (1) developing a message format and protocol between the vehicle and FDC processors and (2) defining the electrical characteristics of the line/radio interface. The former task was accomplished as part of an ongoing independently funded effort addressing SPH digital communication techniques.

  This activity involved a review of current digital data communication schemes and a consideration of their compatibility with the objectives of the SPH fire control problem. While most high-speed computer-to-computer schemes employ synchronous code transmission because of its efficiency, we determined that the compatibility advantages of the asynchronous technique had more to offer in the relatively short-term application for the AGLS-Communication (AGLS-Comm) task. The compatibility of asynchronous code transmission, ASCII character formating and 300-baud FSK modulation with both the RT-524/VRC command radio link and the use of field-wire backup communication made it an ideal choice for AGLS-Comm.

To preserve the longer-range option of synchronous code transmission, a Binary Synchronous Communications (BISYNC) character-oriented protocol was chosen. This protocol uses special characters to delineate the various fields of a message and to control the necessary protocol functions. The communication format designed for AGLS-Comm is presented in Figure 47. Header data is designed to provide control information necessary to steer message traffic (address code), identify message purpose (format code), provide message verification (operation code), and attach special significance (identification code).

To detect transmission errors, BISYNC uses vertical/longitudinal redundancy checks (VRC/LRC). For the ASCII characters a parity check (VRC) is performed on each character (even parity), and an LRC is performed on the whole message. In this case, the block check in the postamble field of the record is a single eight-bit character. If the block check character transmitted does not agree with the block check calculated by the receiver, or if there is a VRC error, then a negative acknowledgement (NAK) is sent to the data source. To correct errors, BISYNC requires the retransmission of a record when an error occurs. Retransmission will typically be attempted several times before it is assumed that the transmission medium (radio or line) is in an unrecoverable state.

When a transmitted record block check character does match the receiver's calculated BCC, the receiver sends a positive acknowledgement (ACK). In addition, alternating sequence code characters (Figure 47) are used to detect duplicated or missing records.

Message formats have been designed to respond to the unique needs of the howitzer fire control problem. The command/request message format (Figure 48) is used for the transmission of gun order data from the FDC to the howitzer. The status field controls the command/request in accordance with the following code:

0001 = New Fire Order
0010 = Fire Command
0100 = Ready Request
1000 = Check Fire

SØH	FC	SC	AC	ØC	IC	STX	MESSAGE (UP TO 40 CHAR.)	ETX	BCC

PREAMBLE (7) POSTAMBLE (2)

SØH		START OF HEADER (01_{16})
FC	=	FORMAT CODE (CONTROL RECORD 44_{16} INFO RECORD $4F_{16}$)
SC		SEQUENCE CODE ($41/4?_{16}$)
AC	=	ADDRESS CODE (40_{16} ALL, 41 → $4F_{16}$)
ØC	=	OPERATION CODE (NO REQ, XMIT DATA, WAIT. ACK/NAK)
IC	=	IDENT. CODE ($40\text{-}4F_{16}$)
STX	=	START OF TEXT (02_{16})
ETX	=	END OF TEXT (03_{16})
BCC	=	BLOCK CHECK CHAR (EXØR OF SØH - ETX (INCL)

Figure 47 SPH Fire Control BISYNC Record Format

o FDC → VEHICLE COMMAND/REQUEST

SSSS DDDD EEEE FFF C	(16)

D DEFLECTION
E ELEVATION
F FUZE
C CHARGE
S STATUS (CMND/REQ)

o VEH → FDC VERIFY

SSSS DDDD EEEE FFF C	(16)

S STATUS (VEHICLE)

o VEH → FDC DATA REPORT

SSSS TTTTTT EC(5) EA(5) EE(5) AC(5) AA(5) AE(5) LL M	(48)

S	STATUS (VEHICLE)	EA	ELEV. ACT. (5)	LL	LEVEL (1)
V	VELOCITY (5)	EE	ELEV. ERR. (5)	M	MODE (1)
T	PROP. TEMP. (6)	AC	AZIM. CMND. (5)		
EC	ELEV. CMND. (5)	AA	AZIM. ACT. (5)		
		AE	AZIM. ERR. (5)		

Figure 48 SPH Fire Control Message Formats

The verify message format, from the howitzer to the FDC, provides acknowledgement that the new fire order has been received in the howitzer. In addition, by sending back the data received in the vehicle, a one-for-one comparison with the transmitted fire order can be made in the FDC computer for further data validation.

The data report message format, from the howitzer to the FDC, provides acknowledgement that a fire command, ready response or check fire command has been received in the vehicle. Gun laying and support system parameters are presented in the data field. The status field codes are complemented in the vehicle to identify the command/request that the data report message is responding to in accordance with the following code:

0000 = Data Request Acknowledgement
1110 = New Fire Order Acknowledgement
1101 = Fire Command Acknowledgement
1011 = Ready Response Acknowledgement
0111 = Check Fire Acknowledgement

Having chosen a protocol (BISYNC, ASCII, 300 baud FSK) and developed a fire control record and message format the inter-vehicle (FDC Howitzer) communication flow (Figure 49) was designed. Critical to the scheme was the necessity to operate the radio (line) in the simplex mode, i.e., the medium (radio channel or line) is unidirectional at any given time, and the transmit/receive mode is under the control of the master processor. The role (master or slave) of the communication processors in the vehicle and fire direction center is dynamic; that is, the roles change as the system executes the connect and disconnect sequence. As the communication systems are initialized, the FDC processor assumes the slave role, polling the communication channel for a request to connect (SELECT) from any vehicle. Once a valid select is received, the FDC processor becomes the master and the vehicle responds. This process continues throughout the communication sequence with the media (radio channel or line) alternately assuming receive and transmit roles. Termination of communication, via the issuance of an end-of-mission to the FDC processor again reverses the roles of the processors.

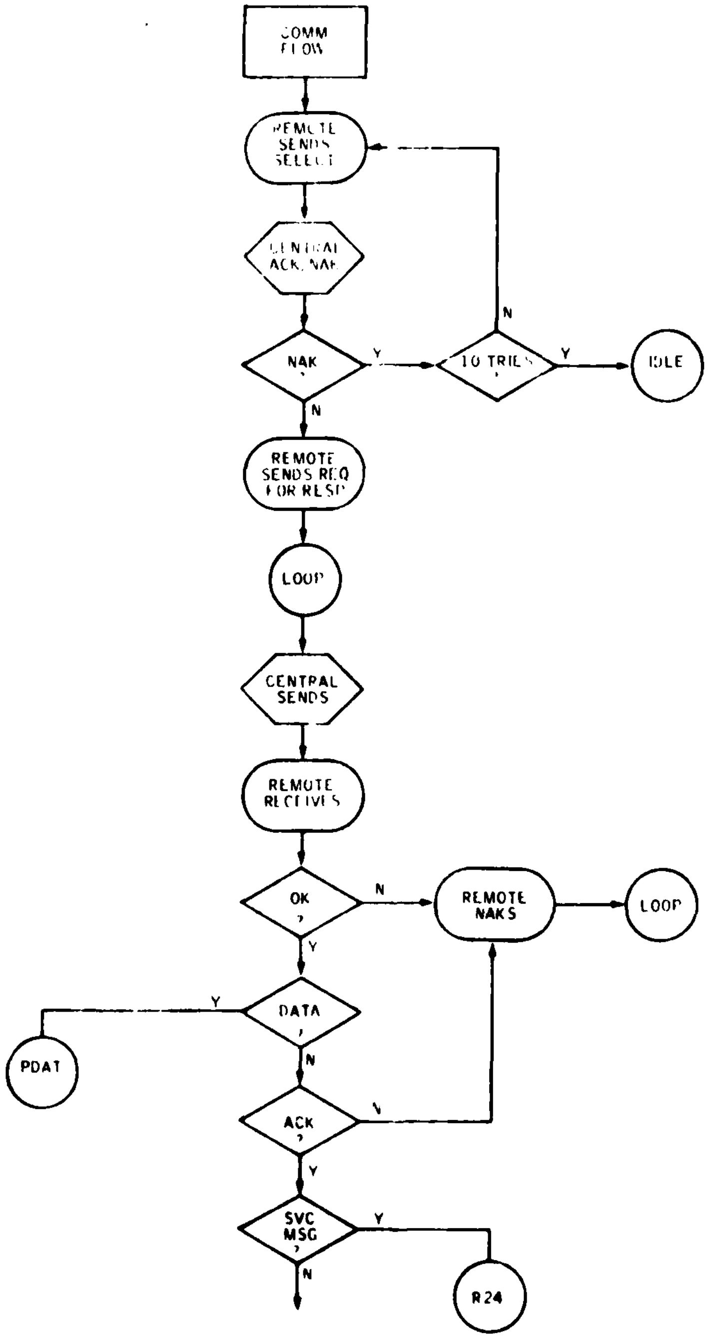


Figure 49 Intervehicle Communication Flow

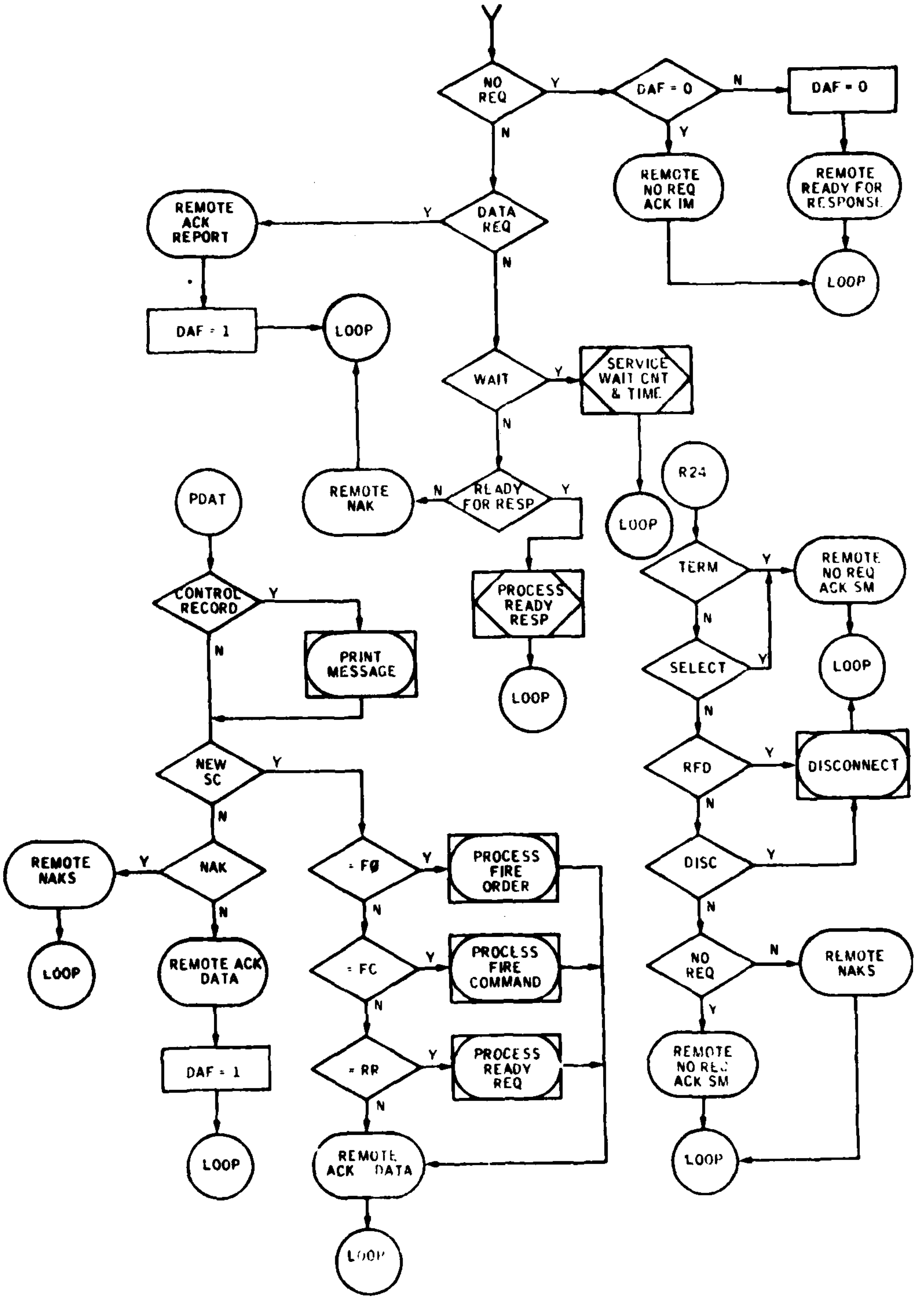


Figure 49 Intervehicle Communication Flow (Concluded)

This communication scheme was applied in the design of the AGLS-Communication system. The inherent reliability of the self-testing/self-correcting BISYNC protocol has proved especially useful when operating in the radio communication link mode.

Definition of the transceiver electrical characteristics revealed that optimum digital signal-to-noise ratio could be obtained by using the X-mode radio interface which has the following characteristics:

Transmit Input Z = 600
Transmit Input Level (Full Dev) = 0b (775 mv)
Receiver Output Z = 600
Receiver Output Level = up to 6 volts pk

Further analysis revealed that while the transmitter input characteristics were ideal the receiver output did not go through the squelch circuit and therefore possessed a continuous noise output. In order to take advantage of the squelch circuit the receiver output was tapped off the R/T connector on the rear of the unit. This output was squelched but the level was constant and not effected by the front panel volume control.

B. Error Analysis

The Automated Gun Laying System was designed to utilize the existing fire control instruments, and to essentially add a servo actuator and a sensor to each axis being automated. As a result, the mechanical errors of the existing fire control instruments will be present in the automated system, in addition to the errors due to the AGLS components.

Because the geometric complexity of the M109 fire control configuration, the following error analyses were performed on one axis of control at a time, except the weapon azimuth and elevation axes. Dynamic errors were not included, since the weapon is not fired on the move, and all servos will have come to rest before firing. Errors were calculated for each of the AGLS levels of automation.

1. Automatic Leveling

In the automatic leveling mode, the error sources include the following:

a) Accelerometer null error $\pm$ 0.75 mil.

b) Accelerometer null error due to scale factor change of 1%, for initial bias of 10 mils = 0.01 x 10 mils = $\pm$ 0.1 mil.

c) Servo amplifier input null error of 1.0 millivolt, for an accelerometer scale factor of 5.0 millivolts per mil = 1 millivolt $\div$ 5 mv/mil = $\pm$ 0.2 mil.

d) Servo loop input required to overcome load friction of 2 lb-ft, with a gear ratio of 20:1, a motor constant of 0.1 lb-ft/amp, motor circuit resistance of 10 ohms, amplifier gain of 100,000, and an accelerometer scale factor of 5.0 millivolts per mil

$$= 2 \text{ lb-ft} \times \frac{1.0 \text{ amp}}{0.1 \text{ lb-ft}} \times 1/20 \times \frac{10 \text{ volts}}{\text{amp}} \times 1/100{,}000 \times \frac{1.0 \text{ mil}}{0.005 \text{ volts}}$$

$$= 0.02 \text{ mil}$$

Maximum untrimmed root sum of squares (RSS) error of AGLS components

$$= \quad 0.75^2 + 0.1^2 + 0.2^2 + 0.2^2 \; = 0.783 \text{ mil}$$

Assuming the accelerometer and amplifier null error with respect to temperature are linear, and that a maximum short term change would be 0.25 mil, the trimmed RSS error would be

$$= \quad 0.25^2 + 0.033^2 + 0.067^2 + 0.2^2 = 0.262 \text{ mil}$$

Using the above errors, and the errors from the applicable specification, the combined leveling errors for the M15 quadrant and the M145 mount are shown in Table 2. In combining errors, the instrument error (excluding backlash) is combined by RSS with the AGLS component RSS

TABLE 2

AUTOMATIC LEVELING ERRORS

Axis (Specification)	Error (Section)	Backlash	Maximum Overall (RSS) + (Backlash ÷ 2)	Trimmed Overall (RSS) + (Backlash ÷ 2)
M-145 Mount (MIL-M-46314B) Pitch or Cant	N/R	1.0 mil	1.28 mil	0.76 mil
M-15 Quadrant (MIL-Q-46315C) Cant	0.75 mil (3.6.2)	0.30 mil (3.6.4)	1.23 mil	0.94 mil
Pitch	0.25 mil (3.6.1)	0.30 mil (3.6.4)	0.97 mil	0.51 mil

value, and added to one-half of the backlash. As an example, the trimmed overall error for the quadrant pitch axis is

$$= \quad 0.25^2 + 0.262^2 + 1/2 \times 0.3 = 0.51 \text{ mil}$$

2. Automatic Offset

In the automatic offset mode, the following error sources are considered:

a) Digital/analog converter error = $\pm$ 4 mv

b) Servo amplifier input null error = $\pm$ 1 mv

c) Servo loop input to overcome load friction of 2.0 lb-ft (quadrant), or load of 1.0 lb-ft at 10:1 gear ratio (pantel) = $\pm$ 0.1 mv.

d) Total encoder error due to digital round-off plus mechanical error = $\pm$ 1 count x 2.5 mv/count = $\pm$ 2.5 mv.

$$\text{Total Error} = \quad 4^2 + 1^2 + 2.5^2 = 4.82 \text{ mv}$$

$$\text{Equivalent Error (Untrimmed)} = 4.82 \text{ mv} \times \frac{1 \text{ count}}{2.5 \text{ mv}} \times \frac{1 \text{ mil}}{10 \text{ counts}} = \pm 0.2 \text{ mil}$$

Assuming that one-half of the digital/analog converter error has been removed by initial trim, the trimmed error is then

$2^2 + 1^2 + 0.1^2 + 2.5^2 = 3.36$ mv = 1.34 counts, or slightly more than $\pm$ 0.1 mil.

3. Automatic Azimuth

a. Reference Unit Acquisition

The reference unit acquisition error is the error that will exist when the AGLS tracker controls the telescope azimuth axis and causes the telescope to point to the GACS reference unit. In this mode, the telescope servo will be driving both clockwise and counterclockwise to maintain the tracker on the reference unit. The telescope backlash will now be inside the servo loop, and the servo will tend to stay in the middle of the backlash region, whereas the automatic offset mode or the manual procedures will cause the telescope to be set to one side of the backlash. This change will always be in the same direction, since the backlash is removed by approaching the commanded value from a lower number. The backlash measured on the modified telescope is approximately 3.0 mils.

The IR tracker error consists of two components; an offset which is a repeatable function of range, and a random noise which increases as the square of the range from tracker to reference unit. The offset can be trimmed for a given range; that is, after the howitzer has been emplaced. Further offset adjustment will not be needed unless the howitzer or the reference unit is moved. The magnitude of this offset is $\pm$ 2.0 mils for a range change of $\pm$ 50 meters from a nominal range of 100 meters. The random noise will not cause a shift, but causes the telescope servo to move through the backlash. At ranges of 250 meters and beyond, the noise increases to a level which prevents tracker lock-on.

The error on initial lock-on, if an extreme change in range has occurred since the last adjustment, could be

$$\text{Error} = \frac{3.0 \text{ mils backlash}}{2} \pm 4 \text{ mils} = -2.5 \text{ to } + 5.5 \text{ mils.}$$

All of this error can be trimmed out after initial acquisition. Subsequent acquisitions will result in an error of 1/2 of the backlash, or 1.5 mils to the right of the reference unit, since the tracker null adjustment is to be performed with backlash removed.

The remaining error, the input needed by the telescope servo to overcome friction torque, is equivalent to 0.02 mil, and as such is negligible in comparison to the backlash. The remaining tracker error due to noise, range change, and scene change is estimated to be 0.25 mil, assuming that the tracker has been properly adjusted at a given range.

b. Telescope

The error in the telescope, excluding backlash, is specified as $\pm$ 1.0 mil. If it is assumed that 20 percent of the backlash might remain or return after backlash has been removed, the overall telescope error becomes

$$(0.2 \times 3)^2 + 1^2 = 1.17 \text{ mils.}$$

c. Telescope Mount

The telescope mount can be the source of large errors of orientation between the telescope and the weapon, because of its mechanical arrangement. The mount specification permits backlash of up to 3.5 mils at maximum quadrant elevation and zero cant. Preliminary checks on the backlash of the GFE mount indicated backlash of up to 20 mils. In addition to the backlash, static errors of up to 2.0 mils can exist at high quadrant elevations. For a quadrant elevation of 800 mils, the backlash is specified as 0.75 mil, and the mechanical error is approximately 1.2 mils. Combining these errors results in an RSS error, on an in-spec mount, of

$$\frac{0.75}{2} + 1.2 = 1.58 \text{ mils.}$$

However, the backlash of the mount had more of an impact on system accuracy than the specified backlash. The mount backlash, between the tracker (feedback sensor) and weapon (load) caused system instabilities at high weapon elevation. Since the AGLS was required to operate at all weapon elevations, it became necessary to implement a drastic reduction in weapon azimuth controller gain to prevent oscillations. As a result of this gain reduction, other errors became significant, as described in the following paragraphs.

d. Weapon Azimuth

The error in the weapon azimuth channel is primarily due to the dc gain of the weapon controller and the error due to friction. Other error sources include turret backlash, controller offset, and servo valve errors.

The dc gain of the weapon azimuth controller is 2.00 volts out per volt in. For a load friction of 10 percent of rated torque, a valve current of 0.03 milliampere is required, or a valve voltage of 15 millivolts. This would require an input voltage from the tracker of 7.5 millivolts, or, for a tracker scale factor of 30 millivolts per mil, a tracker angular error of 0.25 mil.

Because of the low dc gain of the azimuth controller, other errors due to the servo valve must now be considered. These include hysteresis, null bias, and threshold. Hysteresis is reduced by the application of a high frequency excitation or dither. For the valve being used in the AGLS, the remaining hysteresis would be approximately 15 millivolts. The valve null error is 100 millivolts, and the threshold is 25 millivolts.

If the threshold is combined linearly with the load friction effect, the total system threshold is 15 + 25 = 40 millivolts, requiring a tracker input of 20 millivolts, for an equivalent input angular error of 0.67 mil.

Combining the threshold, hysteresis, and the null errors, the RSS error due to valve and load friction is equal to

$$\sqrt{40^2 + 15^2 + 100^2} = 109 \text{ millivolts}$$

or an equivalent input error of 109 ÷ 2 = 54 millivolts, or 1.8 mils angular error.

If the servo valve is properly trimmed by the weapon azimuth trim control, the 100 millivolt valve error can be reduced to 25 millivolts short term error. The valve error would then be equal to

$$\sqrt{40^2 + 15^2 + 25^2} = 49.5 \text{ millivolts}$$

or 0.82 mil equivalent error.

The azimuth system gain was reduced to prevent oscillations due to telescope mount backlash when the weapon was elevated to high quadrant elevations. If the controller dc gain were to remain at 14.0, the value prior to the gain change, the above calculated error attributed to servo valve and load friction effects would be 109 ÷ 14 volts out per volt = 7.8 millivolts, or 0.26 mil angular error.

e. <u>Combined Azimuth System</u>

With an azimuth controller gain of 2.0 the total system error would be, combining the following errors

0.25 mil tracker offset
1.17 mil telescope
1.58 mil mount
0.82 mil azimuth servo

The resulting azimuth RSS error would be

$$0.25^2 + 1.17^2 + 1.58^2 + 0.82^2 = 2.14 \text{ mils.}$$

This error prediction is based on the assumption that the tracker and weapon azimuth servo have been properly trimmed.

If the AGLS components are not considered, the combined error of the M109A1 components would be

1.17 mil telescope error
1.58 mil mount error

or a combined error of

$$1.17^2 + 1.58^2 = 1.97 \text{ mils.}$$

Thus, it can be seen that the added AGLS components in a properly adjusted system add less than 0.2 mil azimuth error, when RSS errors are compared.

4. Automatic Elevation

In the automatic elevation mode, the quadrant is driven by the digital controller to the commanded quadrant elevation, and the level error, as sensed by the quadrant pitch accelerometer, is used to drive the weapon, thus reducing the accelerometer level error to zero. The error sources to be considered then consist of:

M-15 Quadrant Mechanical Error
Quadrant Automatic Offset Servo Error
Accelerometer Offset Error
Weapon Elevation Controller Error

The impact of each of these error sources on the total elevation error is described in the following sections.

a. Quadrant Mechanical Error

The specified error for the quadrant elevation reading is 0.5 mil. Since the AGLS encoder is coupled to the elevation knob, and the accelerometer is coupled to the level vial, the error between elevation knob and level vial will also be present in automatic elevation. It will be assumed that all of the 0.5 mil error is between elevation knob and the level vial.

b. Quadrant Automatic Offset Error

As discussed previously, the error of the automatic offset mode is 0.13 mil. Since this mode determines actual quadrant setting the same error will be one of the error components in the automatic elevation mode.

c. Accelerometer Offset Error

The trimmed accelerometer output voltage is used to control the elevation drive. Thus, if the quadrant pitch servo was previously trimmed, the residual accelerometer error will be reduced to approximately 0.25 mil short term error.

d. Weapon Elevation Servo

The elevation servo error consists of the error needed to overcome imperfect weapon equilibration, load friction, and servo valve errors. The dc gain of the elevation controller is 225 volts out per volt in at the minimum gain setting. The adjusted gain is estimated as 300 volts per volt.

It is estimated that the equilibration mismatch is $\pm$ 10% of supply pressure for different weapon elevations. The load friction is estimated as an additional $\pm$ 10% of supply pressure. Since the elevation servo valve, a pressure control valve, can develop full supply pressure with 3.0 volts applied, the valve voltage needed to overcome load friction and unbalance is combined by RSS to be

$$\sqrt{(0.1 \times 3)^2 + (0.1 \times 3)^2} = 0.42 \text{ volts.}$$

The valve hysteresis, reduced by dither, is 1.0 percent of full scale. The valve threshold is 3.3 percent of full scale, and the null bias is 5.0 percent of full scale. The combined RSS valve error then becomes

$$\sqrt{(0.01 \times 3)^2 + (0.033 \times 3)^2 + (0.05 \times 3)^2} = 0.18 \text{ volt}$$

and the combined load plus valve error is

$$\sqrt{0.42^2 + 0.18^2} = 0.46 \text{ volt.}$$

The equivalent input required to obtain 0.46 volts out is 0.46 ÷ 300 = 1.5 millivolts. Since the accelerometer scale factor is 5 millivolts per mil, the equivalent angular error due to valve and load errors is 1.5 ÷ 5 = 0.3 mil.

e. The dc input offset of the elevation controller module is 1 millivolt, equivalent to 0.2 mil angular error.

f. Combined Error

The RSS combined elevation error consists of

0.5 mil quadrant error
0.13 mil automatic offset error
0.25 mil trimmed accelerometer null error
0.30 mil elevation servo error
0.20 mil controller offset

and is calculated to be

$$\sqrt{0.5^2 + 0.13^2 + 0.25^2 + 0.30^2 + 0.2^2} = 0.68 \text{ mil.}$$

The elevation error without AGLS would be due to quadrant error alone, and would be 0.5 mil assuming no operator errors. Thus, errors due to the automatic elevation components would increase the RSS error by 0.18 mil, if the components are trimmed using the given procedures.

VI. SYSTEM DEVELOPMENT AND FABRICATION

Following the three month design study, development of the AGLS components was initiated. Initial emphasis was placed on the long lead time components, primarily the instrument servo actuators and the infrared tracker, primarily because these components were recognized as the major technical problem areas, but also because the other components (controller amplifiers, power supply and controls and displays) would depend on the characteristics of these components. The weapon drive system was assembled almost entirely from M16A1 Add-On Stabilization System components. The development plan for all of the other AGLS components was to generate layout drawings in sufficient detail to verify their mechanical suitability, prepare drawings of the subassemblies which would permit fabrication by design technicians, and mark up the drawings during fabrication as the need for design changes became evident. The development of the major AGLS components is described in more detail in the following paragraphs.

1. Fire Control Instrument Servos

Early in the development phase, the gearing, drive motors, and other servo components required to provide automatic operation were designed into the M15 quadrant, M145 mount and M117 telescope. Space available for these components had been estimated from the available government drawings, and appeared to be adequate at the time the design review was conducted.

Later, when the M109 was delivered to the contractor, wooden mock-ups of the servos were fabricated and installed on the fire control instruments. It was then found that the added drive components caused interference when the vehicle was canted beyond 5 degrees. Since it was decided that full cant performance capability must be retained, additional effort was exerted to redesign the servo housings to provide full 10 degree cant capability. This redesign activity consisted primarily of installing the wooden mock-ups of the servo components, displacing the fire control instruments to their extreme positions, removing the interfering material, and rearranging the internal mechanical components to fit within the available space. As a part of this redesign effort, a different motor and tachometer were selected. The new motor has less torque at a higher speed, so 20 to 1 gear ratios were needed to meet the torque requirements of the two mount axes and the two quadrant axes. The calculated speeds for maximum load friction for each servo are shown in Table I.

Inspection of the worm gears in the quadrant and mount revealed that the existing mechanical stop, consisting of a screw in the sector gear, was not adequate for the higher torque available from the servo motor. Limit switches were then installed, to interrupt the motor current when the servo approached the mechanical limit. Each switch is bypassed with a diode which permits reverse motor current to automatically provide drive away from the limit. In addition, resilient stops were installed beyond the limit switch settings to cushion the impact in the event of a hard-over failure. The instrument servo drives, modified as described, were then capable of operating to within 0.5 degree of $\pm$ 10 degree vehicle cant and pitch.

Following completion of the fabrication of modified servo units, open loop frequency response data was taken for each of the fire instrument servo units as shown in Figures 50 through 54. In each of these tests, the tachometer feedback loop was closed, a sinusoidal forcing function applied and the position sensor (accelerometer or IR tracker) output was measured in magnitude, and phase. After developing the controllers for each axis, the closed loop frequency response data representing final servo performance were taken and are plotted in Figures 55 through 59.

2. IR Tracker

The proposed method of detecting and locking-on to the GACS reference unit was to use two solid state video cameras, each utilizing a charge-coupled device (CCD) array. However, about five months into the program concern was raised over the projected size of the optical elements needed by these two cameras. Since the camera could not resolve readily to any smaller image distance than that of one sensor element, it would have been necessary to use a large focal length (10 inch) lens to sense 0.5 mil.

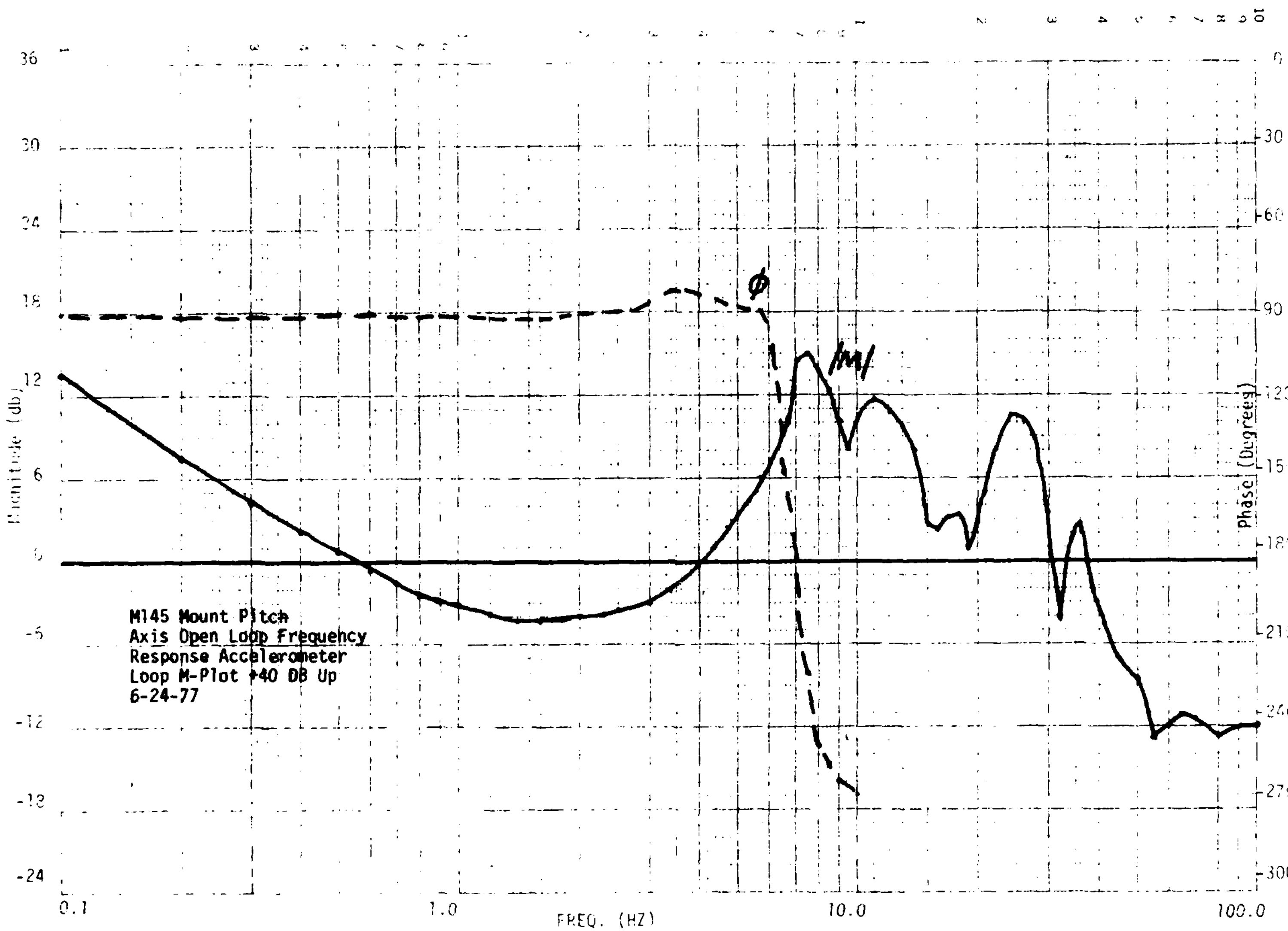


Figure 50. M-145 Mount Pitch Axis Open Loop Frequency Response Accelerometer Loop M-Plot +40DB Up.

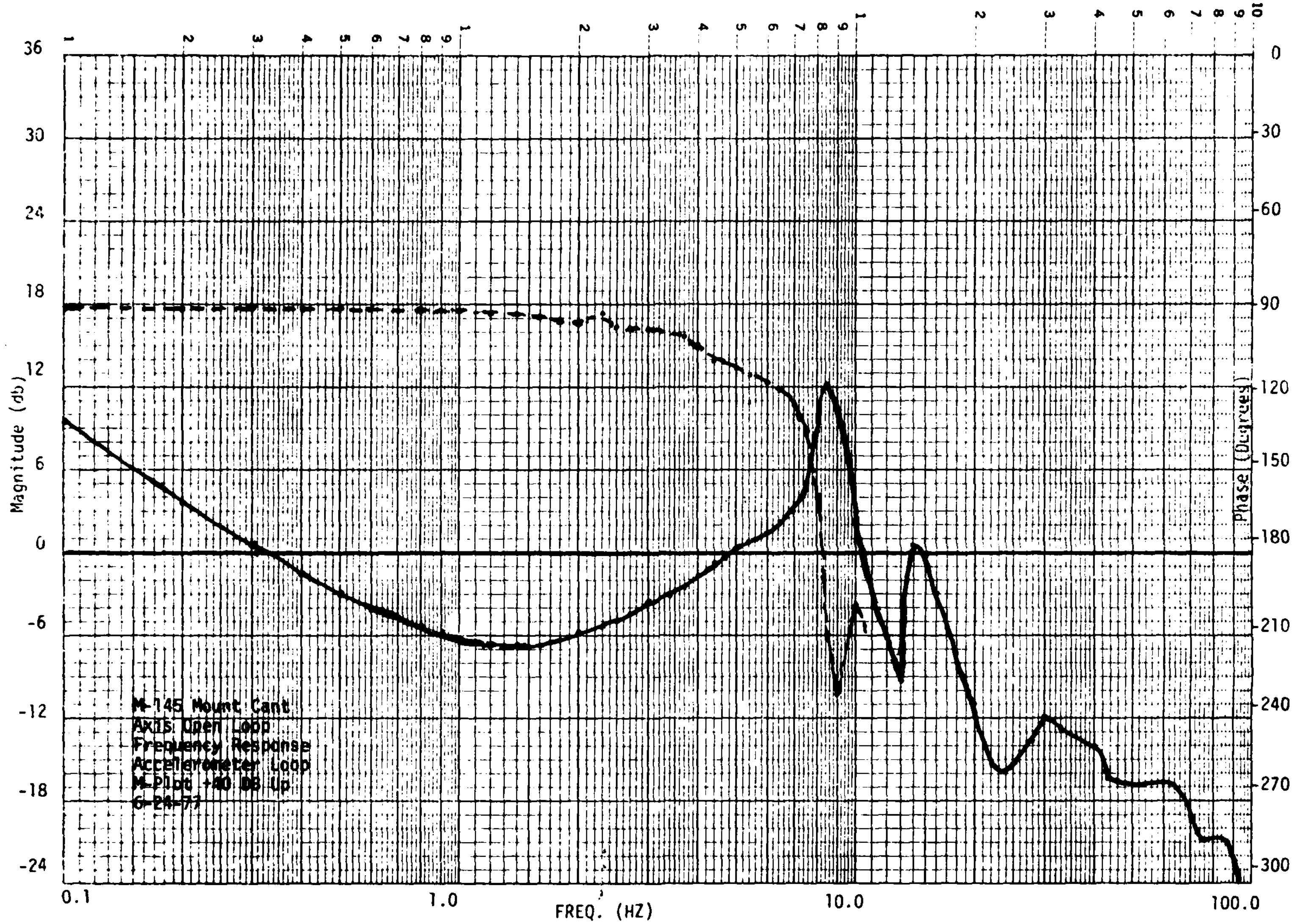


Figure 51. M-145 Mount Cant Axis Open Loop Frequency Response Accelerometer Loop M-Plot +40 DB Up.

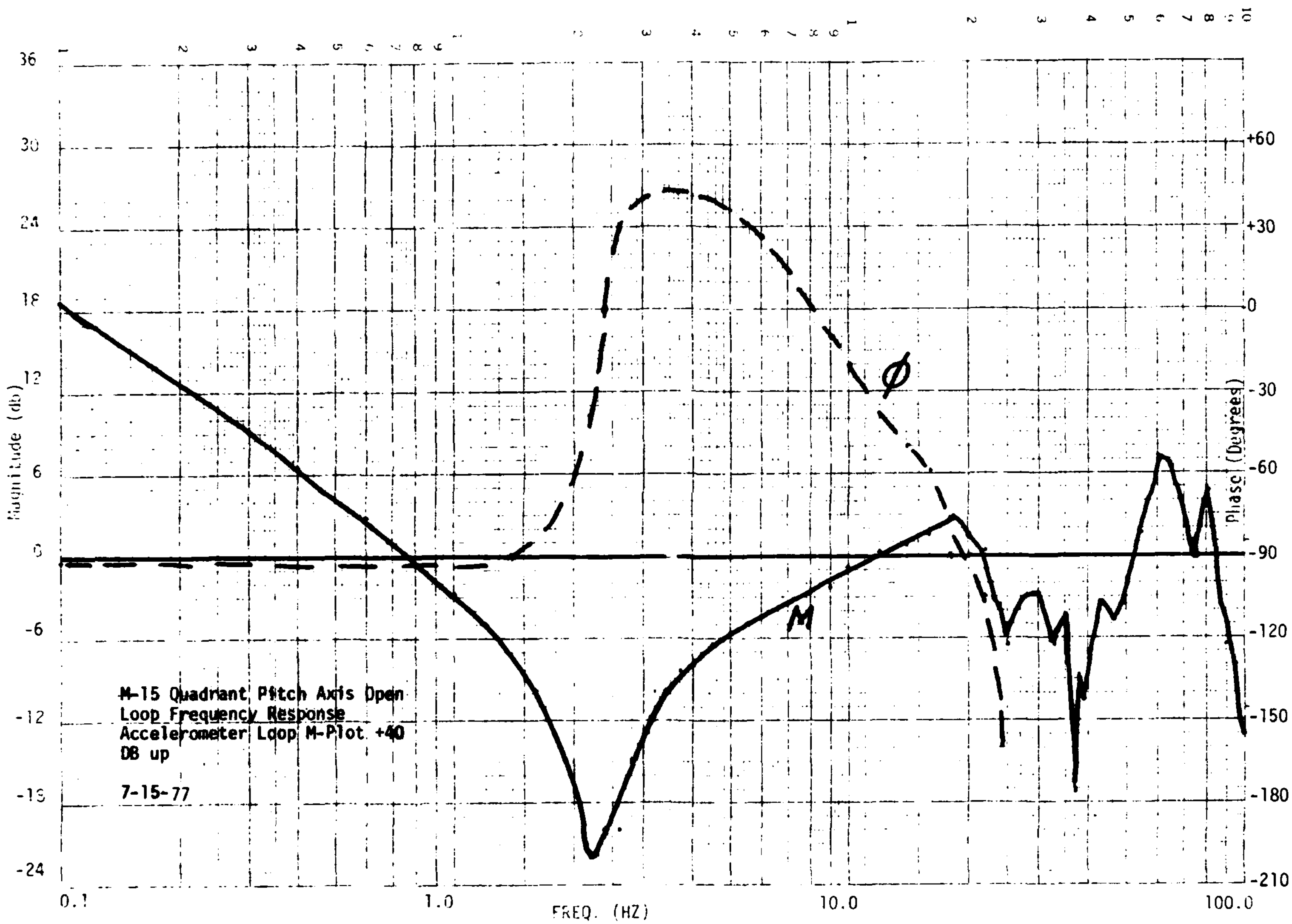


Figure 52. M-15 Quadrant Pitch Axis Open Loop Frequency Response Accelerometer Loop M-Plot +40 DB Up.

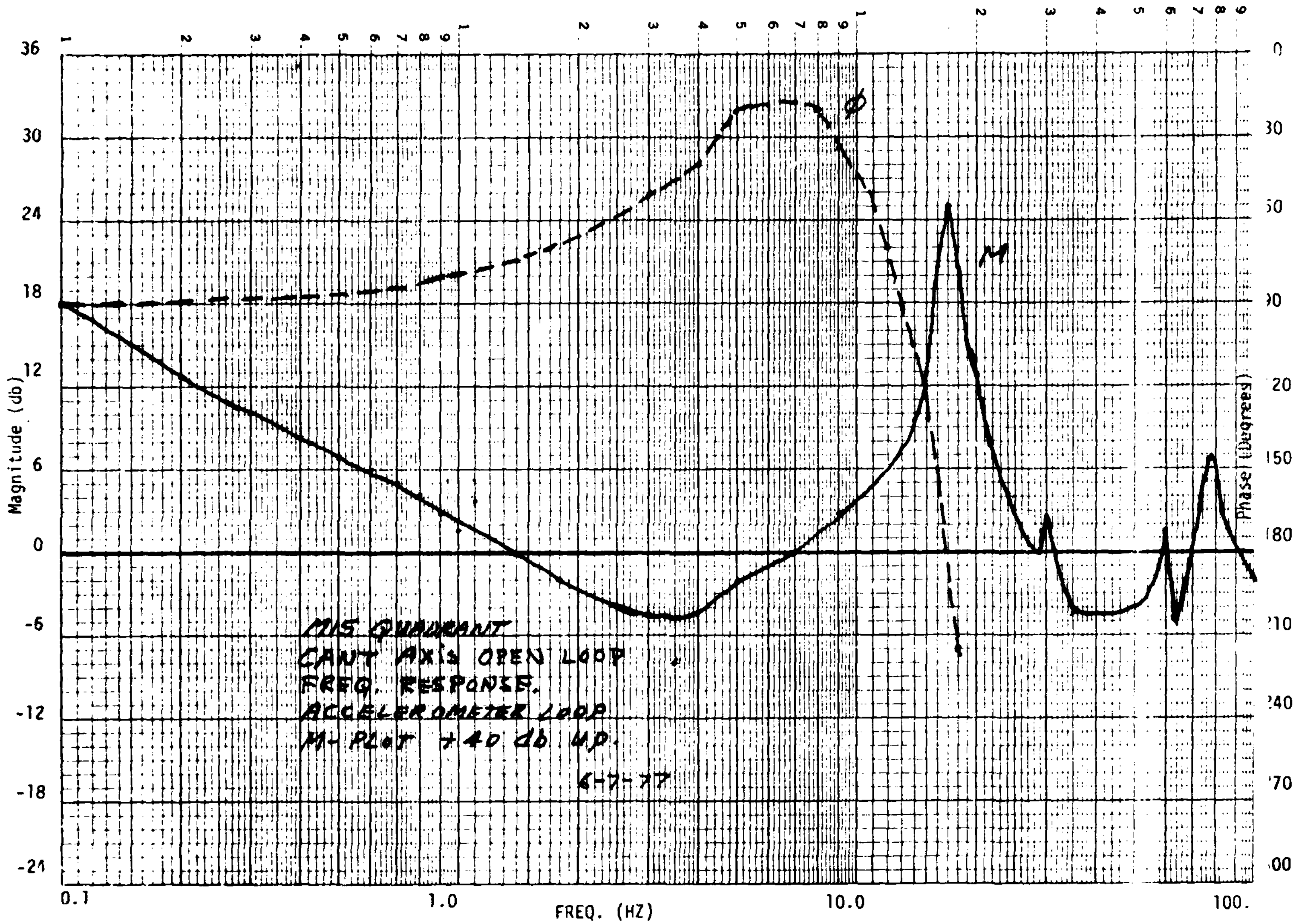


Figure 53. M-15 Quadrant Cant Axis Open Loop Frequency Response Accelerometer Loop M-Plot +40 DB Up.

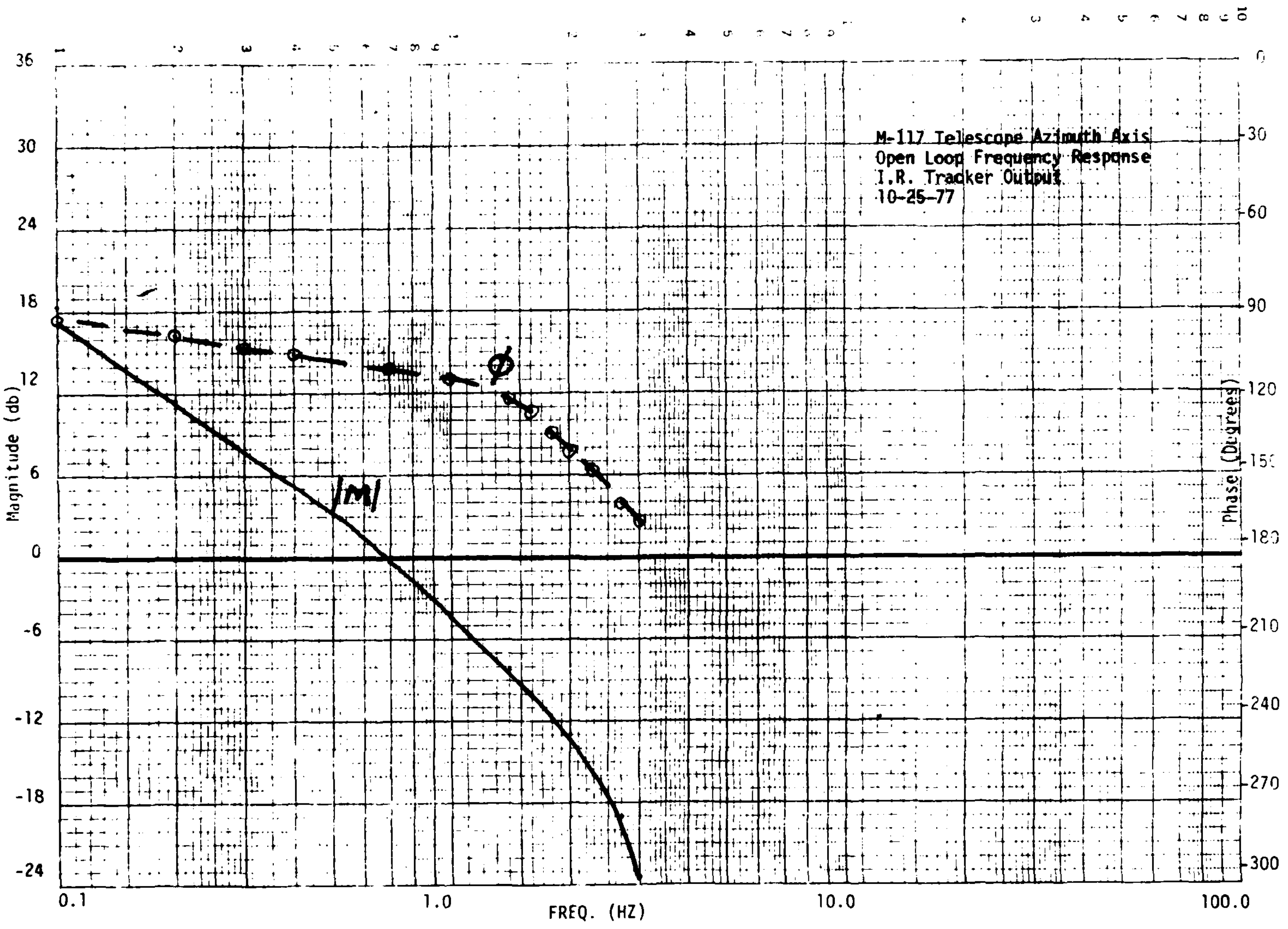


Figure 54. M-117 Telescope Azimuth Axis Open Loop Frequency Response I.R. Tracker Output.

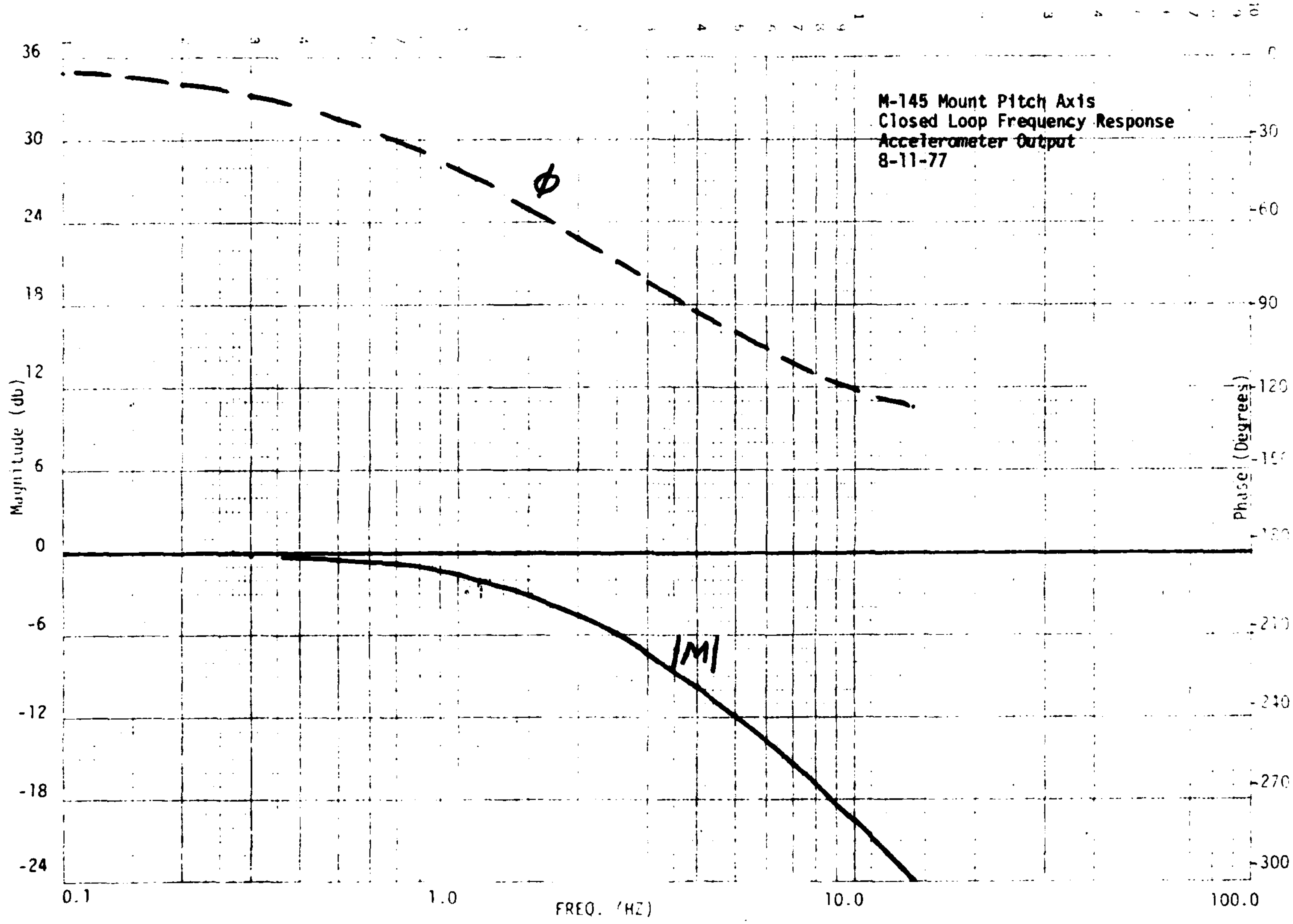


Figure 55. M-145 Mount Pitch Axis Closed Loop Frequency Response Accelerometer Output.

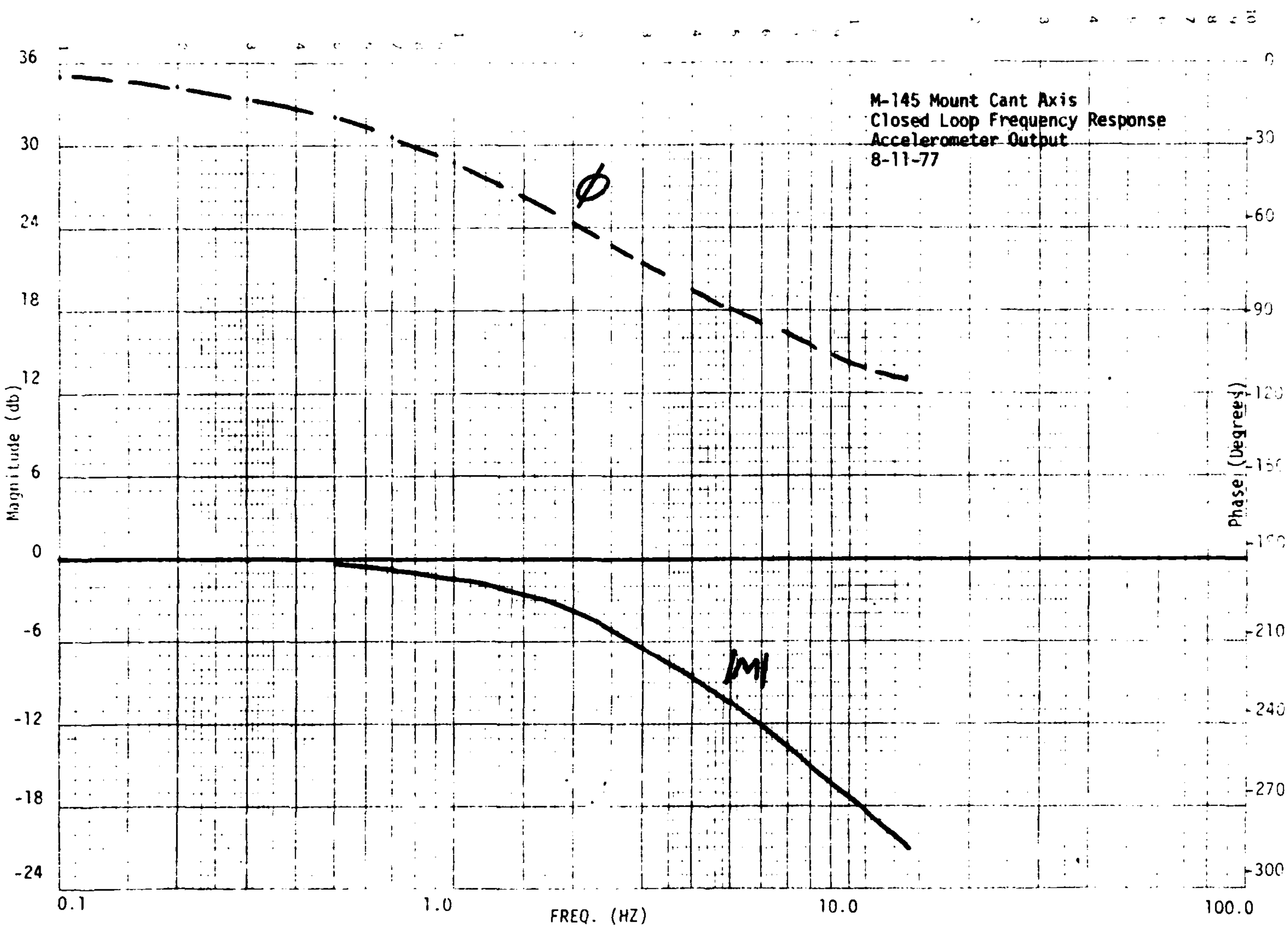


Figure 56. M-145 Mount Cant Axis Closed Loop Frequency Response Accelerometer Output.

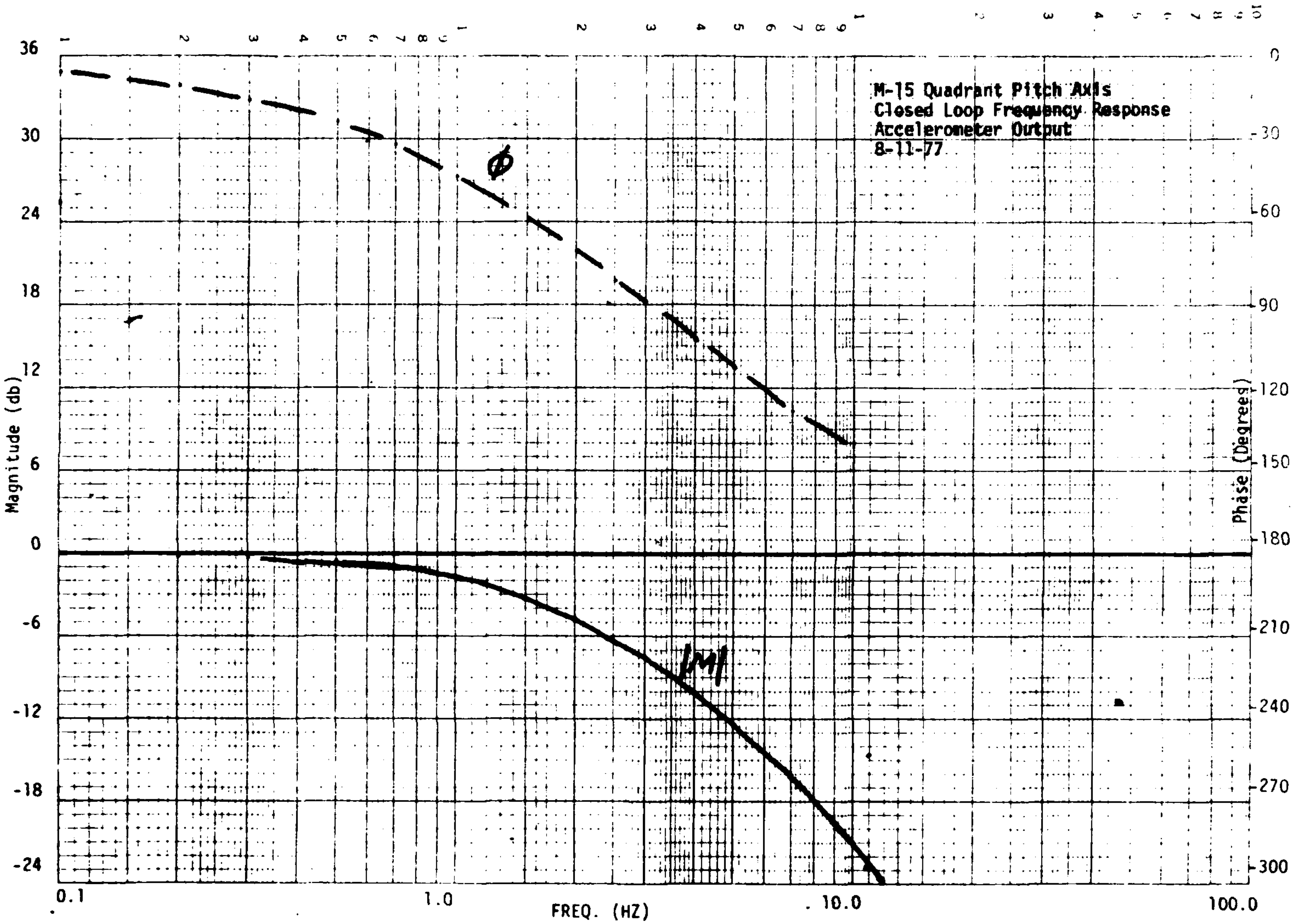


Figure 57. M-15 Quadrant Pitch Axis Closed Loop Frequency Response Accelerometer Output.

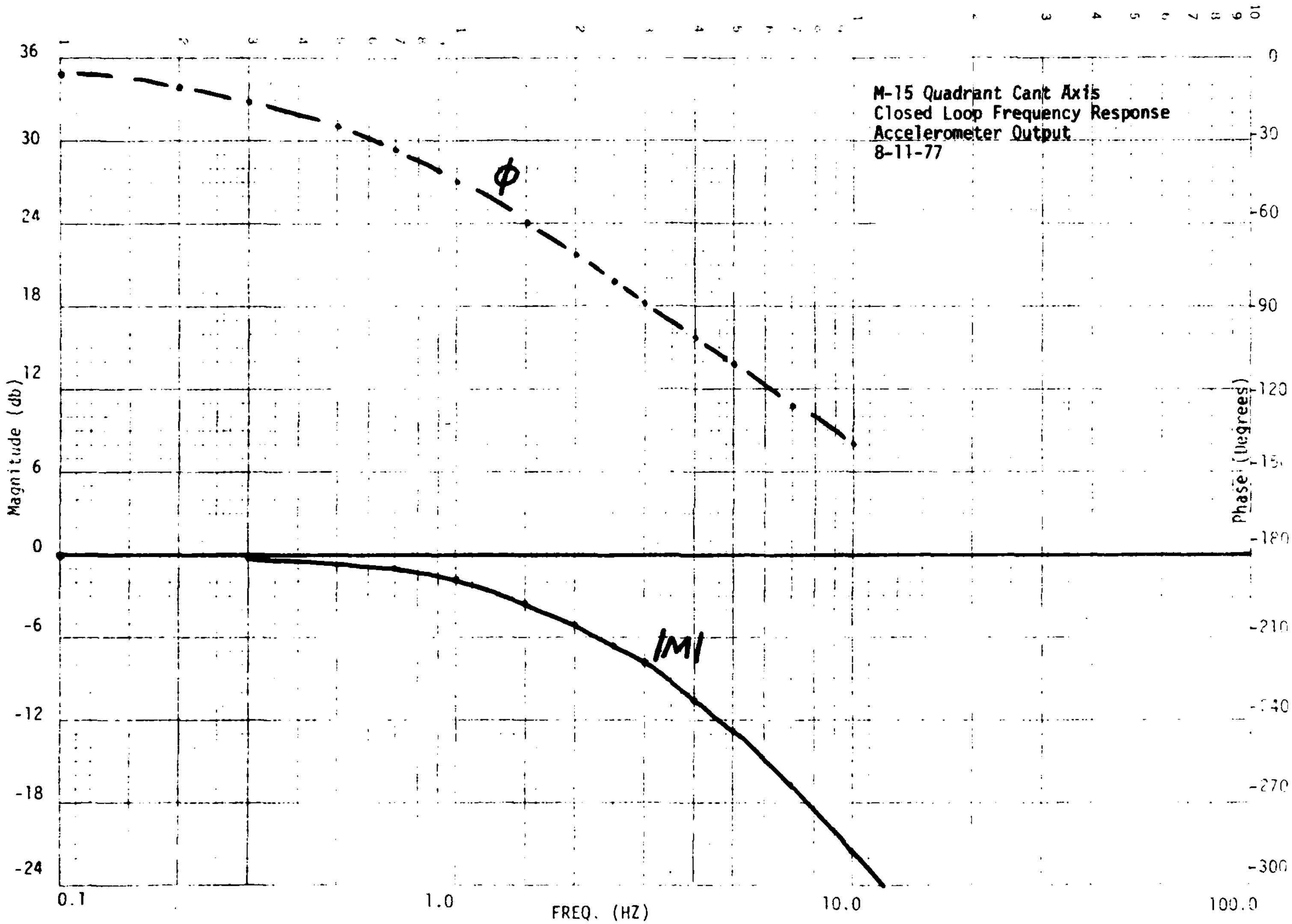


Figure 58. M-15 Quadrant Cant Axis Closed Loop Frequency Response Accelerometer Output.

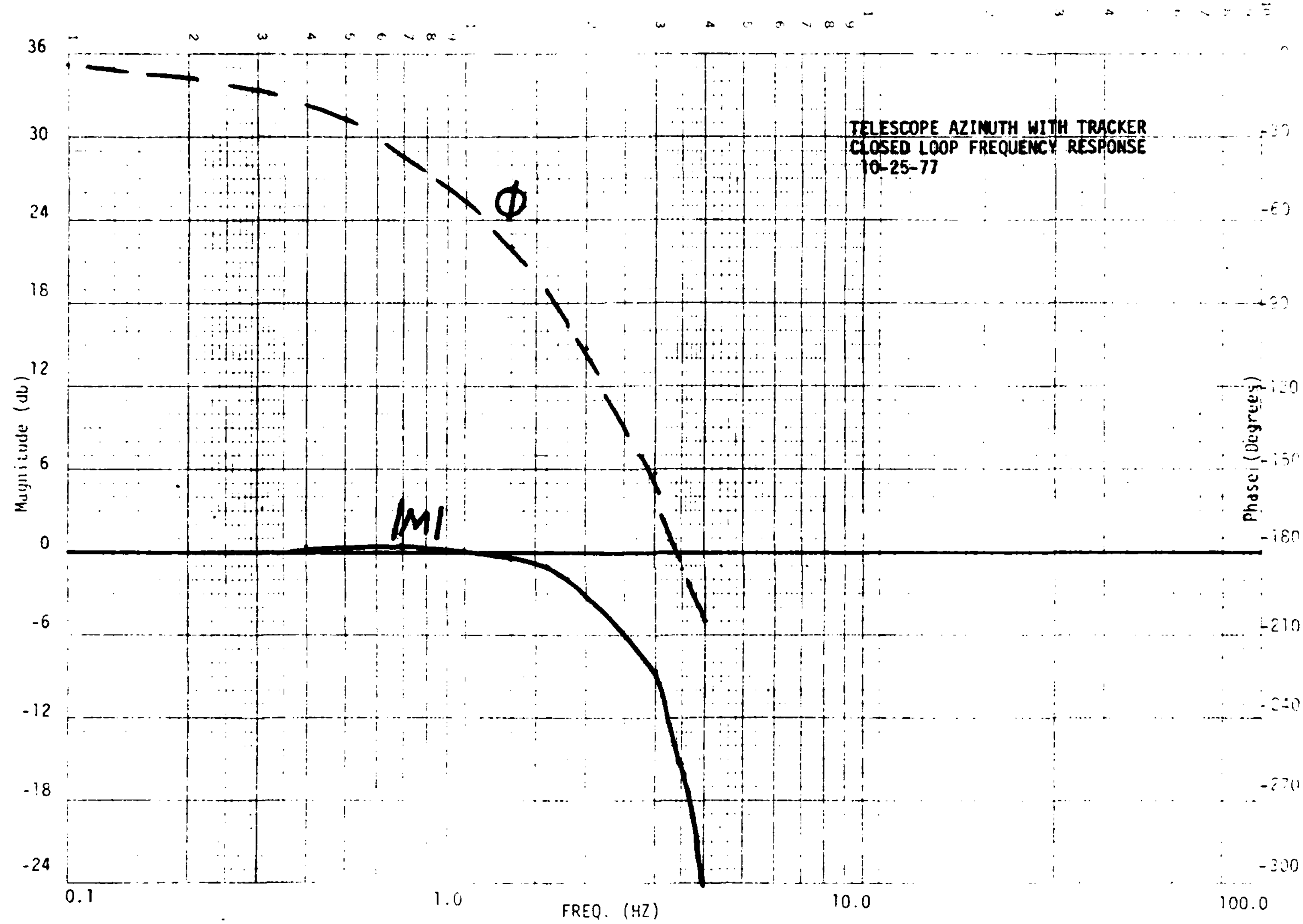


Figure 59. Telescope Azimuth with Tracker Closed Loop Frequency Response.

One solution considered was to integrate the tracker into the pantel optical system, thus utilizing the pantel magnification to reduce the added magnification needed by the AGLS tracker. However, it was apparent that space was limited in the vicinity of the pantel eyepiece, thus potentially creating packaging problems. In addition, the optical accuracy with respect to the lower housing would be degraded.

Analysis of the XENON sterradian lamp energy, based on a Frankford Arsenal measurement of 2500 watts per sterradan, suggested that a CCD array might not be able to detect the XENON lamp in the GACS reference unit. This concern was later determined to be a major problem, when the GACS reference unit was delivered to Honeywell in January, 1977. The reference unit was first set up in a 50 meter indoor ballistic range, and viewed with a commercially available GE camera with a CCD array. The camera could detect the XENON lamp, but the camera lens was opened to a low f-number. The camera output, as read on a television monitor, also exhibited a vertical row of spots rather than a single image. This ghosting would not be acceptable for the AGLS application.

The reference unit was then taken outdoors for further imaging tests. The camera could not detect the XENON flashes, either with a video monitor or with tracking electronics. The problem was more severe outdoors, because the bright scene saturated the CCD sensor when the lens was opened to the number needed for the indoor tests. At higher f-numbers which took the sensor out of saturation, the tracker would not be able to detect the XENON lamp under any ambient light conditions.

The problem is caused because, although the light output of the XENON lamp is high, the time duration is short (1 to 2 microseconds) and the repetition rate is low (160 pulses per second). This results in a low level of average energy. Since each element of the CCD array integrates all light that falls on it during one scan period, the pulse energy is a small fraction of the total energy seen by the array element. Two viable solutions to this problem were considered:

1. Place an electronic (PLZT) shutter, synchronized to the XENON flash, in the optical path to reduce the amount of ambient light while admitting all of the XENON light.

2. Utilize a photodiode sensor which would respond in real time to the high energy pulses, thus producing a high output pulse over a low background ambient signal.

Electronic Shutter

The application of a PLZT shutter to the CCD array was evaluated by several technically qualified personnel. It was determined that although the PLZT shutter would meet the GACS interface requirements (20 to 200 microsecond transmission time at a 160 Hertz rate), there were no off-the-shelf drivers in existence to control the shutter. Thus, in order to evaluate the PLZT shutter in conjunction with a solid state camera, it would be necessary to either build a PLZT shutter specifically for this application or design and build a special driver amplifier.

The PLZT shutter has a capacitance of 0.01 to 0.05 microfarad and must be driven by a 600 volt pulse. For a 20 microsecond wide pulse, the peak current to the PLZT would be from 1.0 to 5.0 amps, assuming a current with a half sine wave time function. The combination of high current and high voltage could cause Electromagnetic Interference (EMI) compatibility problems with the CCD array, since the CCD depends on the generation and transfer of small amounts of charges.

Segmented Photodiode Detector

A segmented photodiode detector system, designed by Honeywell for aircraft fire control systems, was tested to determine its ability to detect the GACS reference unit XENON lamp. This detector was able to readily detect the lamp to ranges beyond 500 meters even in bright sunlight. This test proved that a photodiode sensor detecting in real time was the best solution.

A problem with the segmented detector is that present manufacturing technology cannot provide separation between the elements of less than 0.001 inch (25 micrometers). If a 50 millimeter focal length lens were used, this would result in an angular dead space of 0.5 mil in which the lamp, as a point source, could not be detected.

To circumvent the problem of a finite dead space, a sensor was investigated which provides proportional information on the position of a light spot. The sensor is a lateral-effect photodiode with two output leads. The output current from the leads can be combined by the sum-and-difference amplifier to obtain a signal proportional to the distance of the light spot from the center of the photodiode. A sample of this device was ordered and evaluated for use as a detector.

The spot continuous diode was found to be ideally suited to the GACS tracking problem, for the following reasons.

- o It responds to the peak of the XENON flash, thus providing a comparatively large signal to noise ratio.
- o The diode responds linearly to light level, so that steady state background illumination can be readily removed from the total sensor signal thus enabling detection of low energy pulses.
- o The diode provides a continuous measure of image position, without the descrete steps and dead spaces of a CCD or segmented detector.
- o A short focal length simple optical system can be used.
- o Only one detector and lens system is needed.
- o The sensor is commercially available off-the-shelf at a reasonable cost.

The spot continuous diode was further tested, and electronic circuits were developed to best utilize the signal from the diode. A complete description of the tracker is included in Section III and IV of this report.

3. IR Tracker Noise

A fundamental problem in tracking the XENON lamp is that at far ranges the energy level of the light pulses decreases as an inverse function of the square of the range. The XENON lamp does have enough energy to permit detection. However, the

AGLS tracker is intended to resolve displacement of the XENON lamp, down to 0.25 mil, and has a field of view of ± 100 mils. Thus, the tracker must be able to resolve a signal corresponding to 0.25 percent of the total energy applied to the sensor.

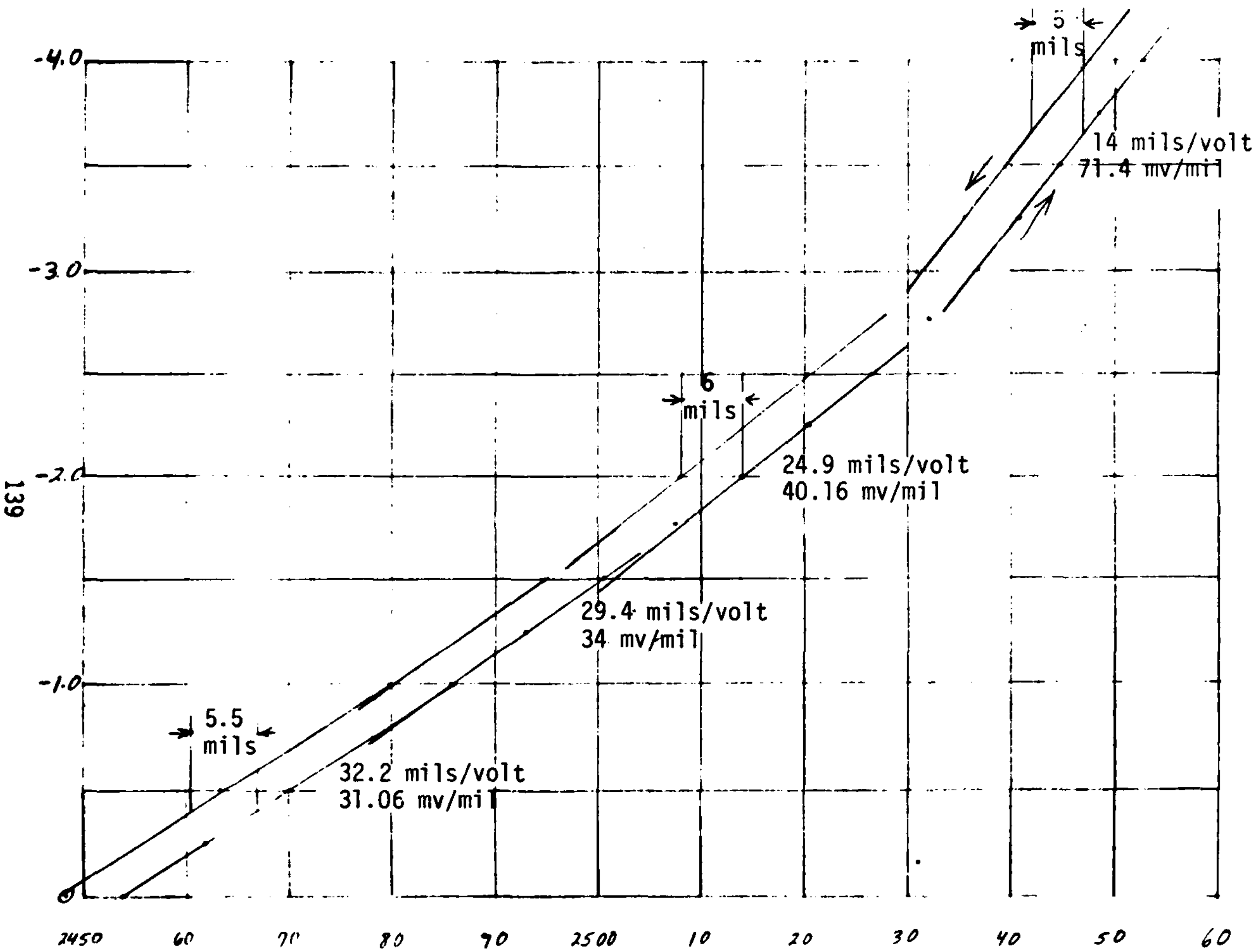
The tracker was tested with the GACS reference unit at an indoor range of 17 meters. To simulate longer ranges, aperatures were fabricated to reduce the optical entrance area. These tests demonstrated that at 70 meters distance from tracker to reference unit, the tracker output noise is equivalent to a displacement of 0.5 mil peak to peak, increasing to 2.0 mils at 140 meters and 8.0 mils at 280 meters.

The observed noise appeared to be essentially uniform with respect to frequency, and could be reduced by filtering. However, the servo system driven by the tracker cannot tolerate additional filtering since excess phase shift will result.

The tracker noise, therefore, will not permit reference unit lock-on at extended ranges. This was demonstrated in the AGLS acceptance tests, during which lock-on could not be achieved at ranges greater than 200 meters. To extend the lock-on range would require either more power from the XENON lamp or a larger entrance pupil area on the tracker. Since the reference unit is provided as GFE, it cannot be modified. In addition, it might be undesirable to increase the light output from a countermeasures viewpoint. Increasing the tracker pupil area would add to the tracker size and weight, further compounding the problem of the mechanical loading on the pantel and mount.

This deficiency was avoided during the acceptance tests and the Ft. Sill tests by keeping the reference unit within 50 to 100 meters of the howitzer. At these ranges, no problems were encountered in either acquiring or maintaining lock on the reference unit.

The tracker was installed on the M117 telescope, and data taken on output voltage as a function of telescope deflection when viewing the GACS reference unit. The results are shown in Figures 60 and 61. The two curves on each figure represent clockwise and counterclockwise rotation of the telescope. The difference



Tracker Output Test
1/2" Iris 17 m Range
Indoors
Data pg. 76 10-25-77

Figure 60. Tracker Output Test 1/2" Iris 17 m Range Indoors

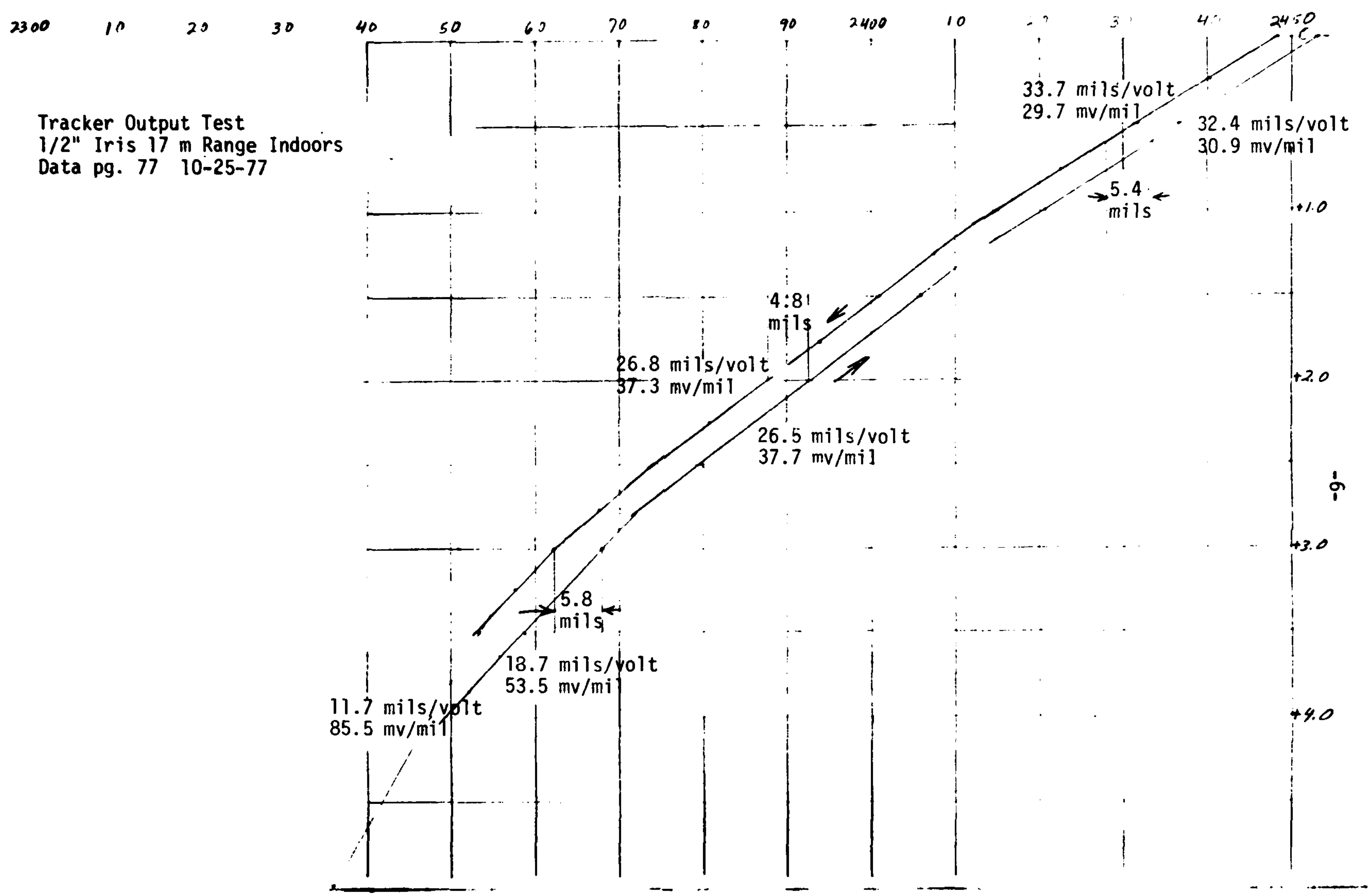


Figure 61. Tracker Output Test 1/2" Iris 17 m Range Indoors

between the two curves is due to telescope backlash. The increasing slope of the curves as the telescope moves away from center is believed to be caused by reflections placing multiple images in the tracker field of view.

4. Telescope Backlash

When the IR tracker was mounted on the M117 panoramic telescope, the backlash measured either optically or by the tracker was 5 to 6 mils compared to 1.0 mil required in the telescope specification. The AGLS controller is configured to always drive the telescope from a lower number to the desired final number and thus stay on one side of the backlash, using the same procedure as is now used by the gunner. However, this procedure cannot be expected to remove all backlash, and some small amount will remain. This residual backlash is probably proportional to the total backlash, so it is desirable to reduce this backlash to improve the overall accuracy of the telescope in the manual and automatic offset modes.

A second reason for reducing the backlash is that, during reference unit lock-on, the tracker is continually driving the telescope and reversing directions as the telescope servo approaches a null. The backlash will then cause the servo motor to continually hunt through the backlash region, and cause a servo instability. In addition, the telescope/tracker servo will tend to null in the middle of the backlash, but the telescope, when driven by the gunner or under command of the digital controller, will null at one side of the backlash. Thus, the telescope will appear to be in error by one-half of the backlash when the tracker is used to control the telescope.

The backlash was reduced to 2 to 3 mils by Ft. Sill personnel during the system acceptance tests. This was accomplished by turning-in the worm shaft spring adjustment which is external to the telescope. The spring did not require any additional adjustments through the acceptance tests or the Ft. Sill dry firing test sequences. However, during live firing the backlash increased. Again, it could be reduced by turning-in the screw. Since the actual screw position before firing was not marked, it is not known whether the backlash increase was due to the screw moving or if changes occurred internal to the telescope during gun firing.

The telescope was later disassembled and inspected by Ft. Sill field maintenance personnel and a contractor's representative. No sign of mechanical wear could be found. The servo drive was then removed from the telescope and the backlash was rechecked. The telescope backlash was less than 1 mil, as required by the pantel specification.

It appears that the springs which axially and radially load the pantel worm shaft are not adequate to hold the shaft against the motor friction and the motor torque. It would be desirable to incorporate stronger springs to keep the worm shaft in place. However, the increased loading could possibly increase the friction on the pantel.

A design study is needed to review this problem area and to determine what changes can be implemented.

5. <u>Control and Display System Configuration Definition</u>

During the Design Study phase, the need was recognized for data displays distributed among the several gun crew members. Specifically, the chief of section requires full display of all data since he is responsible for proper laying of the weapon. In addition, the gunner must have available at his station a display of all azimuth data, and the assistant gunner requires the elevation data.

The Design Study also pointed to the need to easily modify system configuration, and thus change certain fire control functions from manual to automatic. Since a certain configuration might require proper function of other subsystems, it became apparent that a high level of logic would be required. To provide the additional data displays for all crew members, the contract was modified to require three separate display panels as well as a separate digital controller unit. The three display panels would all receive their display data by means of a data bus, thereby reducing the number of display wires between units from an estimated 480 wires down to less than 60 total conductors.

To satisfy the need for an easily modified system configuration, the contractor selected a microprocessor based digital control subsystem. Such a system can easily be altered by external configuration selector switches, and can be interfaced with the other AGLS subsystems to provide a high degree of operator safety. The digital system is compatible with the data bus selected to service the data display panels. Another advantage of the microprocessor is that it will permit changes in system operation to be implemented by a simple software modification with no changes in components or wires. Since Honeywell has a complete microprocessor development facility, including semiconductor memory programming equipment, system operating characteristics can be modified by programming a new memory array in a matter of hours.

The block diagrams and descriptions of the digital controller unit are given in Section III. Detailed schematics of the controller unit circuits are shown in Figures 62 through 67.

AGLS Software Development

The AGLS Controls and Display software has been designed to:

- Sequence control operations as a function of the mode selected.
- Acquire fire order data from the GACS gun unit.
- Monitor leveling servos and GACS reference unit detector.
- Compute pointing errors as the difference between commanded and measured angular data.
- Display commanded, measured and error quantities as well as level and reference unit detector status.
- Provide overall system performance checks to insure safe operation.
- Interface with chief of section controls to enable interactive operation.

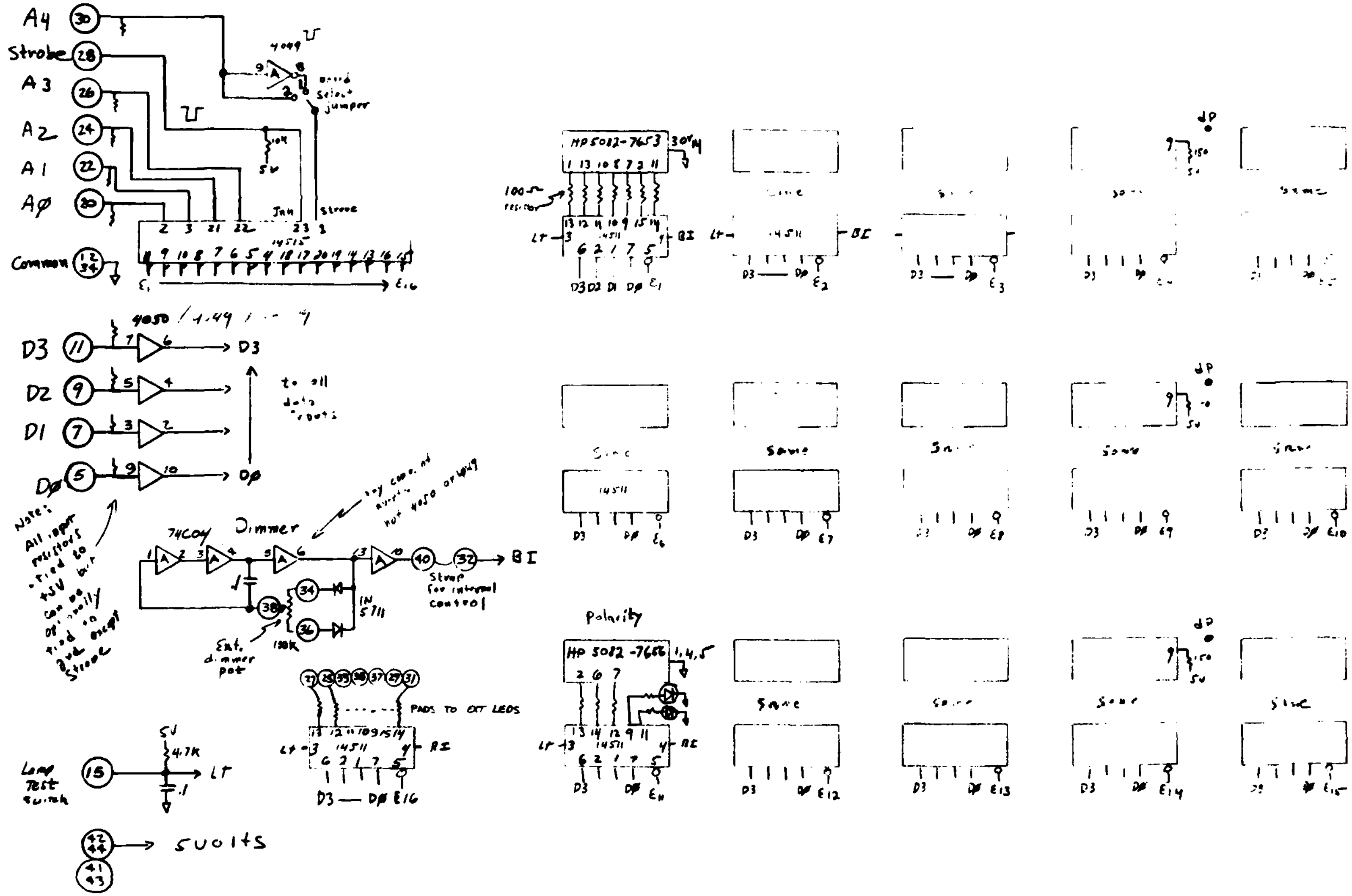
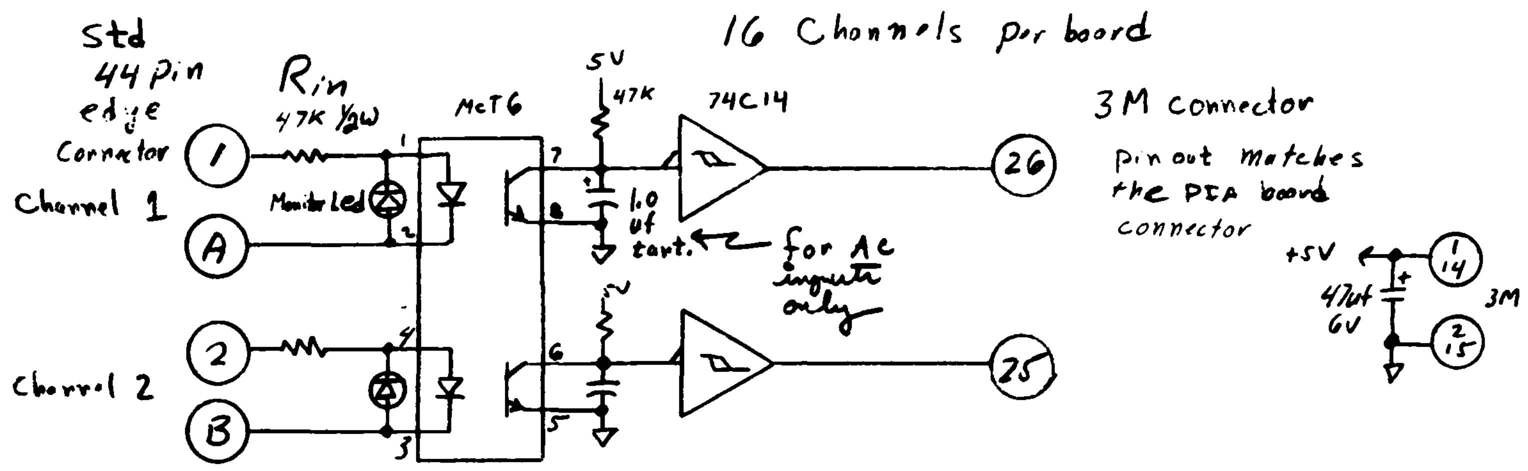


Figure 62. Display Driver



Channel No	3M pin No	22 pin No S
1	26	1, A
2	25	2, B
3	24	3, C
4	23	4, D
5	22	5, E
6	21	6, F
7	20	7, H
8	19	8, J
9	12	9, K
10	11	10, L
11	10	11, M
12	9	12, N
13	8	13, P
14	7	14, R
15	6	15, S
16	5	16, T

Figure 63. Parallel Input Signal Conditioner

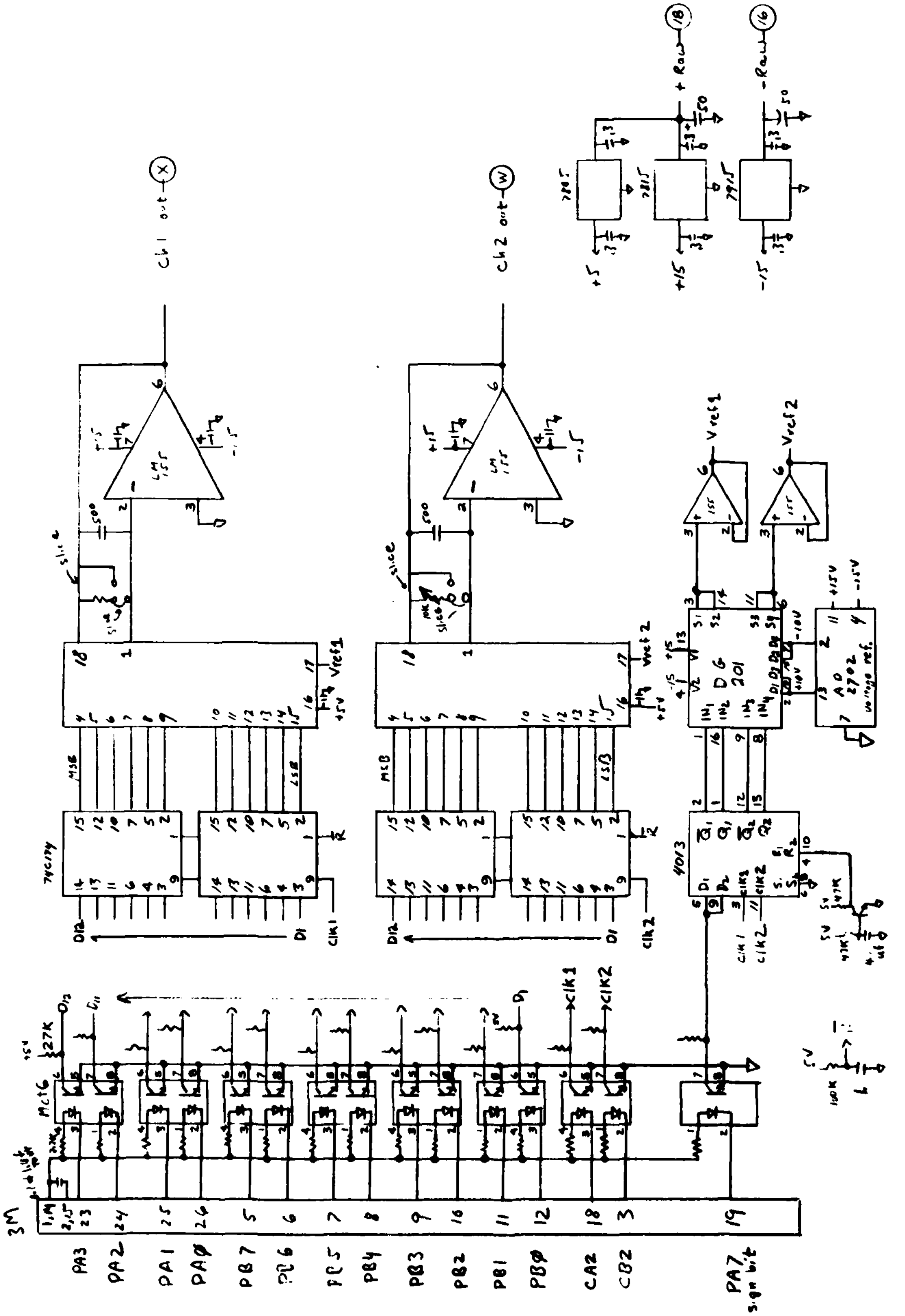


Figure 64. Dual Digital to Analog Converter

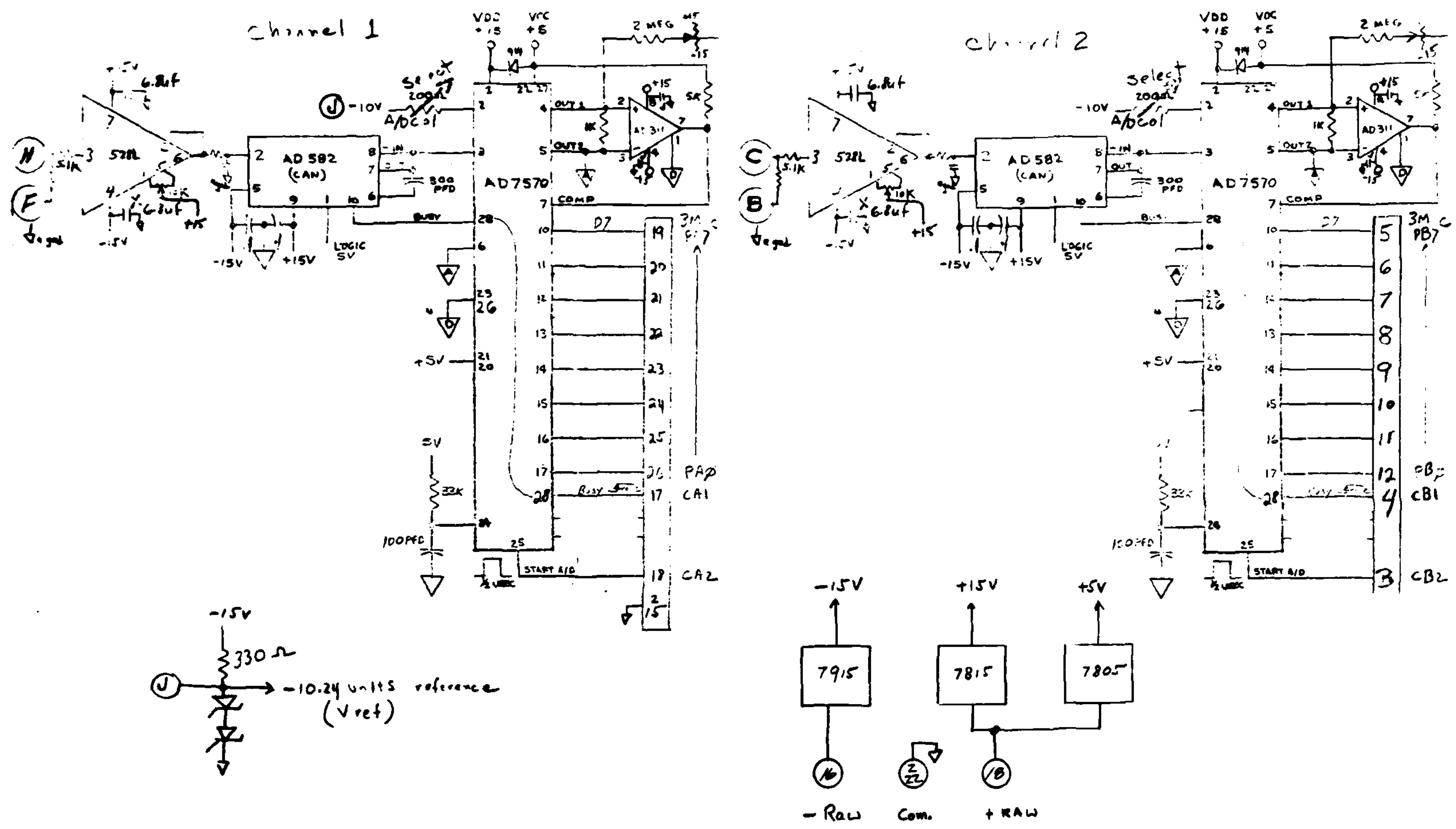


FIGURE 65. DUAL A/D CONVERTER

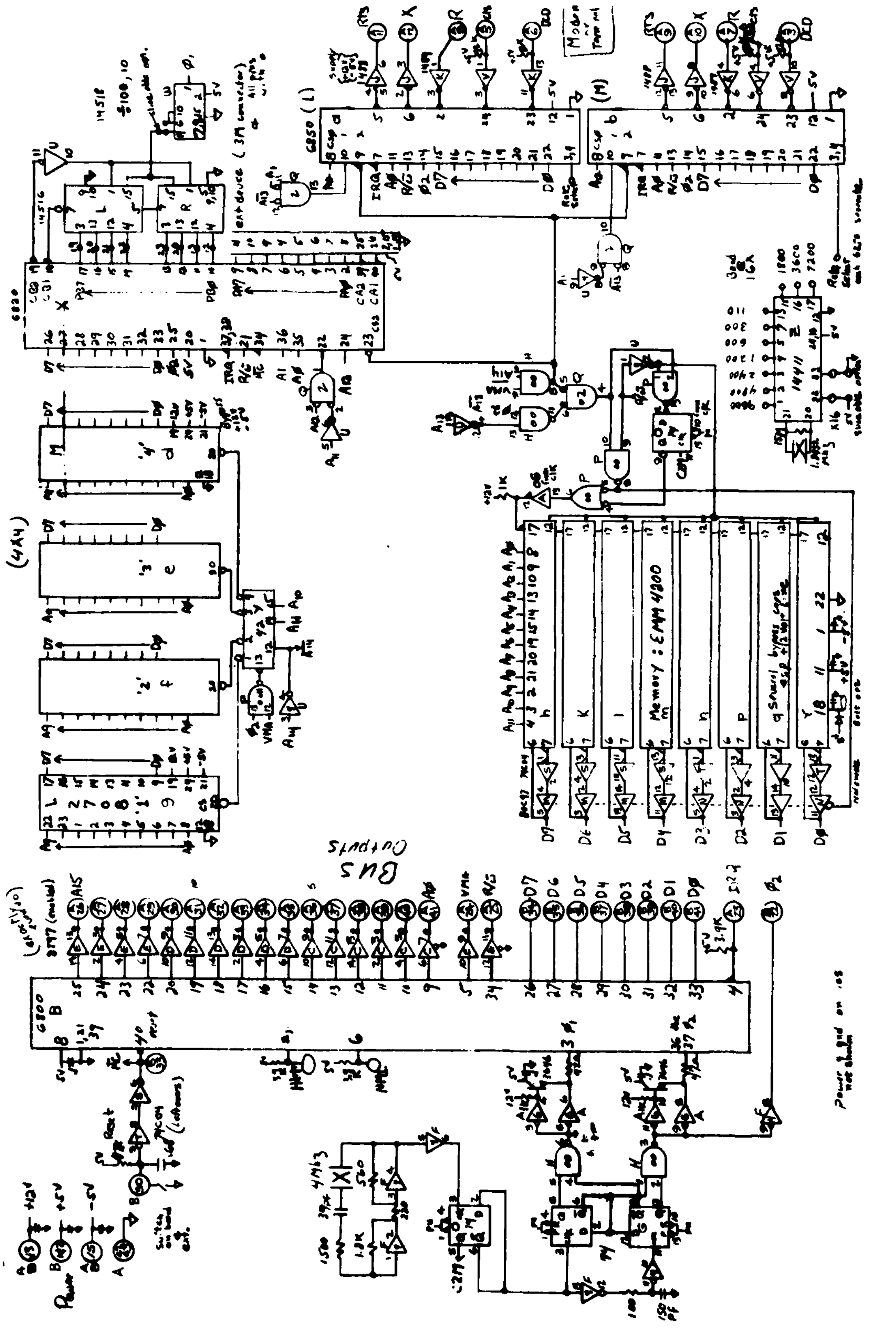


Figure 66. Central Processor Unit

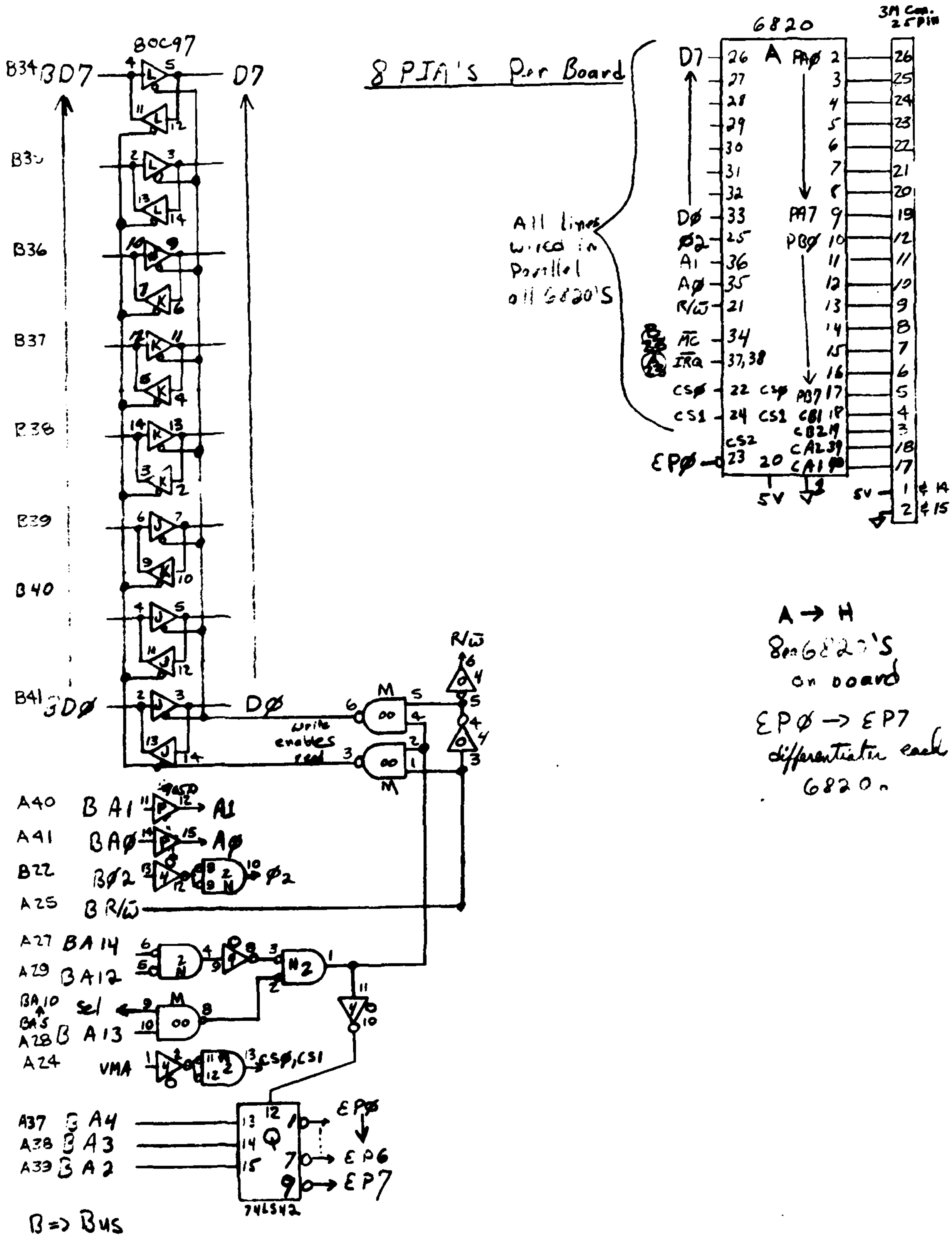


Figure 67. Parallel Interface

The overall structure of the software employs a main program which sequentially executes a variety of subroutines. This main program and supporting subroutines are written in Motorola Assembly Language (Version 1.00) which is documented in Motorola M6800 Microprocessor Programming Manual (M68 PRM(D).

The main program is supported by an interrupt driven background package which performs the functions of:

- o Providing five double precision (16 bit) programmable timers with 1 millisecond resolution.

- o Servicing the Digital to Analog converters for azimuth and elevation each 20 milliseconds.

- o Testing for the presence (or absence) of the XENON lamp signal each 20 milliseconds.

    - If the XENON lamp signal is present, and has been present for 25 consecutive tests (500 msec) the XENON recognition flag (XRECF) is set.

    - If the XENON lamp signal is absent, and has been absent for 1 second or more the XENON recognition flag is cleared.

- o Determining whether the weapon switch is being toggled, each 20 milliseconds, and setting the weapon flag (WPNF) if the switch is enabled.

- o Updating the displays and status indicators each 200 milliseconds. Display update is inhibited if new GACS data are being acquired or if a computation is in progress.

The background package is also used to service a program activity monitor which allows examination of the data and flag buffer contents through the use of a CRT on serial port $3002.

The main program is structured to run continuously through either the first, second or third loops (Appendix B) depending upon the position of the servo and weapon switches.

The first loop (idle loop) is the only loop active when the servo switch is off. The routines in this loop read commanded (GACS) data, encoder trim, encoder data and servo system status. The difference computation is performed and results are displayed. Manual adjustment of pantel mount and quadrant leveling will be monitored and the appropriate indicators will be illuminated when the values are within established bounds. The system must be operating in this mode to latch in new configuration data or to read information from the GACS system.

The second loop is active anytime the servo switch is on. This loop services the pantel search and auto offset mode of operation and performs the preliminary tests for the automatic modes. Auto-offset mode selection, in either azimuth or elevation, drives the appropriate fire control instrument to the angle commanded via the GACS link. In the azimuth axis a special control algorithm is employed to ensure that the pantel is always driven to the specified angle from the left to right (as observed from the eye piece by an operator). Azimuth search is initiated in the auto-azimuth mode by toggling the switch on the chief-of-section control panel. This operation mode can only be disabled by tracker acquisition of the GACS reference unit or by disabling the servo switch.

The third loop is enabled by actuation of the weapon switch. Before the switch signal will be recognized, the first loop conditions of XENON signal stable and quadrant offset null must be satisfied. Once recognized further testing of the second loop in inhibited. Third loop processing proceeds (if the auto mode has been selected) three continuous checks of the XENON stable signal, by commanding the appropriate weapon servos to position the turret and tube. In the elevation axis a one second delay is employed to ensure that quadrant accelerometer tangential influences have been minimized prior to initiation of tube movement. If load position was selected, and falls within prescribed limits, the tube is driven to the elevation selected by the configuration switches. When QE is selected (load deselected) the servo has to be disabled in order to latch in the commanded elevation angle. System stabilization, as determined by tracker and quadrant staying within specified null limits for a timeout period, is signalled by flashing of the numerical displays. When this happens, the program goes to idle mode (Loop #1) and the weapon flag is inhibited. To reset the system, the servo switch must be toggled.

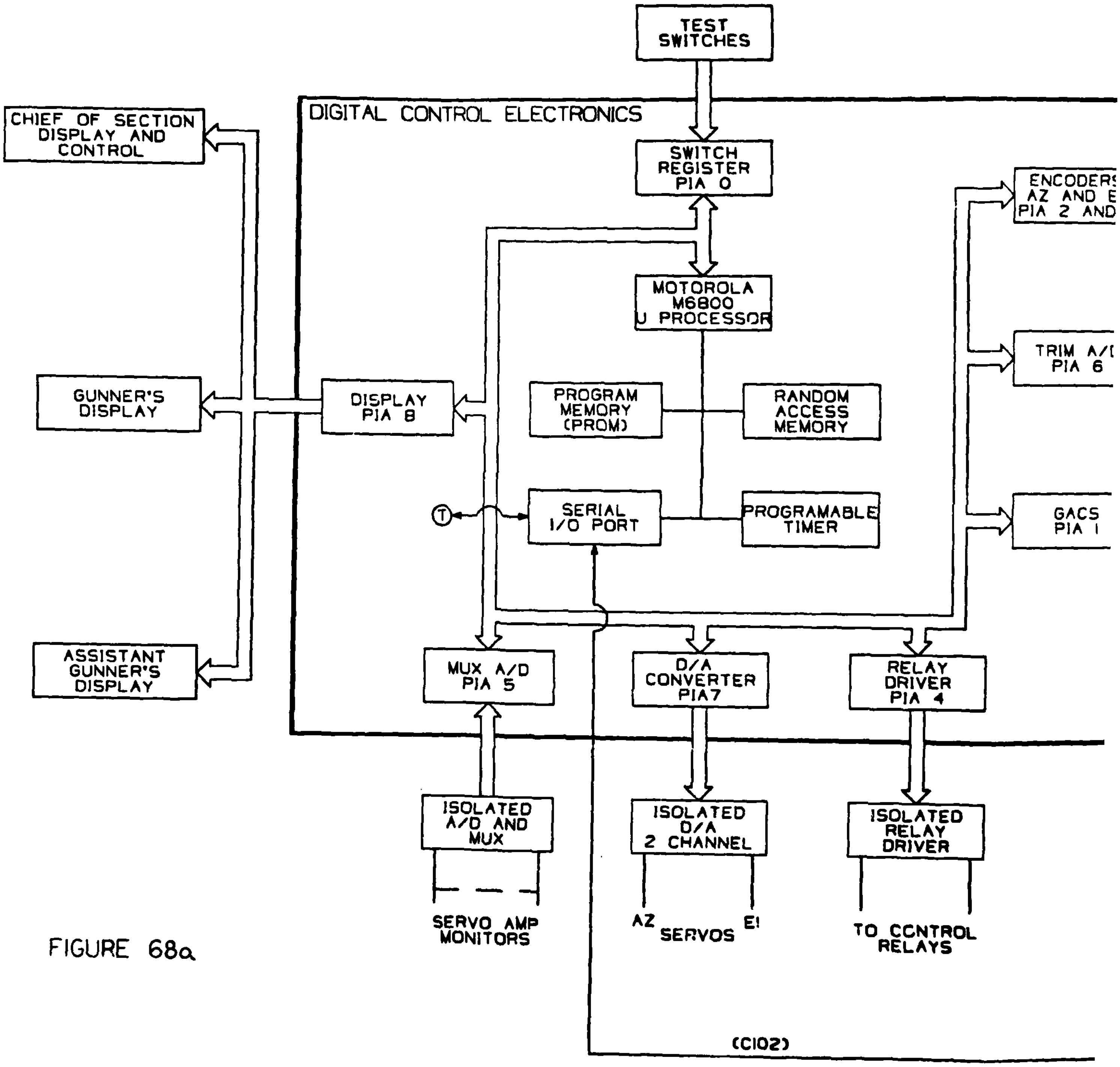
A detailed flow diagram of the main program is presented in Appendix C and the annotated source listing of the entire program is included as Appendix D.

6. Vehicle Communication Processor (VECOM)

The vehicle-located functions of the add on scope of work consist of the Vehicle Communications Processor, which provides two-way data communications with the FDC, and the Reference Unit Processor, which determines the Reference Angle for the GACS subsystem. To minimize the component placement problem, it was decided to house these two processors in a single assembly, mounted at the Chief of Section work station; as shown functionally in Figure 68. Interfaces to the AGLS, AN/VRC-46 Radio, DR-810 velocimeter, electronic fuze setter, and propellant temperature measurement system, were implemented by a start interconnect scheme as shown in the cabling diagram 28116114, of Figure 69.

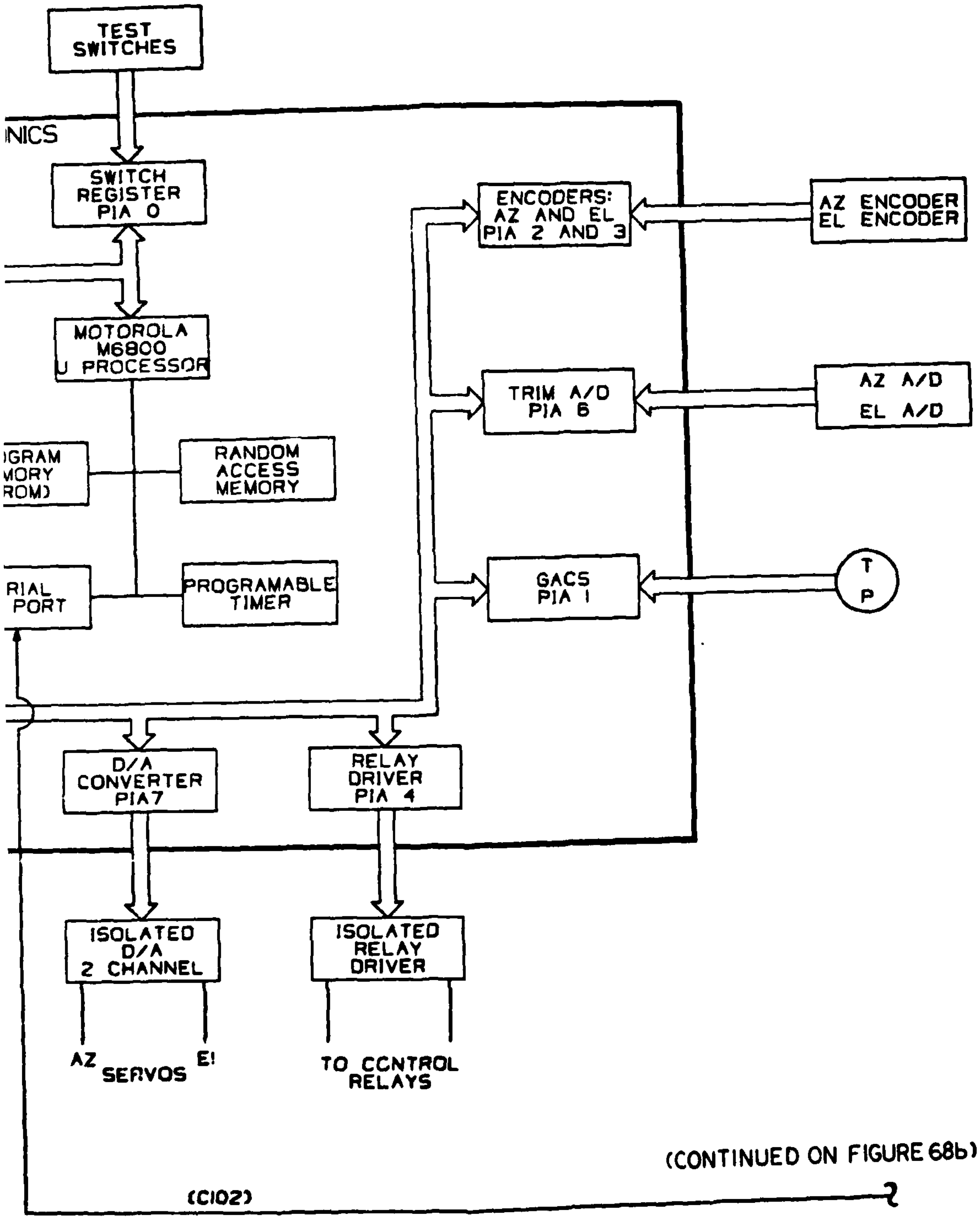
Both the VECOM and RUP were implemented using the Honeywell H10 microprocessor. This MPU and its supporting board set is shown schematically in Appendix A. Throughout the AGLS-COMM add-on system the respective elements of the H10 (MPU, Memory, PIA, etc.) are interchangeable, differing only in the instruction sets stored in the socket-mounted EPROMS. Two unique circuit cards were fabricated for VECOM; the reference unit phase locked loop and the processor DC power supply.

The pulses from the GACS IR receiver are preconditioned with the phase locked loop, which establishes time windows to accept the XENON pulse (see Figure 70). It was found that the S pulse was approximately 2.5 milliseconds before the next X pulse. The phase locked loop is synchronized with the X pulses, and a time window equal to $\pm$ 10% of the pulse period is opened to accept pulses for transmission to the digital processor. Any pulse occurring in this window is assumed to be an X (and not S) pulse. A second window is opened from 3.2 to 4.5 milliseconds after the X pulse to accept an S pulse.



TEST SWITCHES
DIGITAL CONTROL ELECTRONICS
CHIEF OF SECTION DISPLAY AND CONTROL
SWITCH REGISTER PIA 0
ENCODERS AZ AND E PIA 2 AND
MOTOROLA M6800 U PROCESSOR
GUNNER'S DISPLAY
DISPLAY PIA 8
PROGRAM MEMORY (PROM)
RANDOM ACCESS MEMORY
TRIM A/D PIA 6
T
SERIAL I/O PORT
PROGRAMABLE TIMER
GACS PIA 1
ASSISTANT GUNNER'S DISPLAY
MUX A/D PIA 5
D/A CONVERTER PIA7
RELAY DRIVER PIA 4
ISOLATED A/D AND MUX
ISOLATED D/A 2 CHANNEL
ISOLATED RELAY DRIVER
SERVO AMP MONITORS
AZ
SERVOS
El
TO CONTROL RELAYS
(C102)

FIGURE 68a



(CONTINUED ON FIGURE 68b)

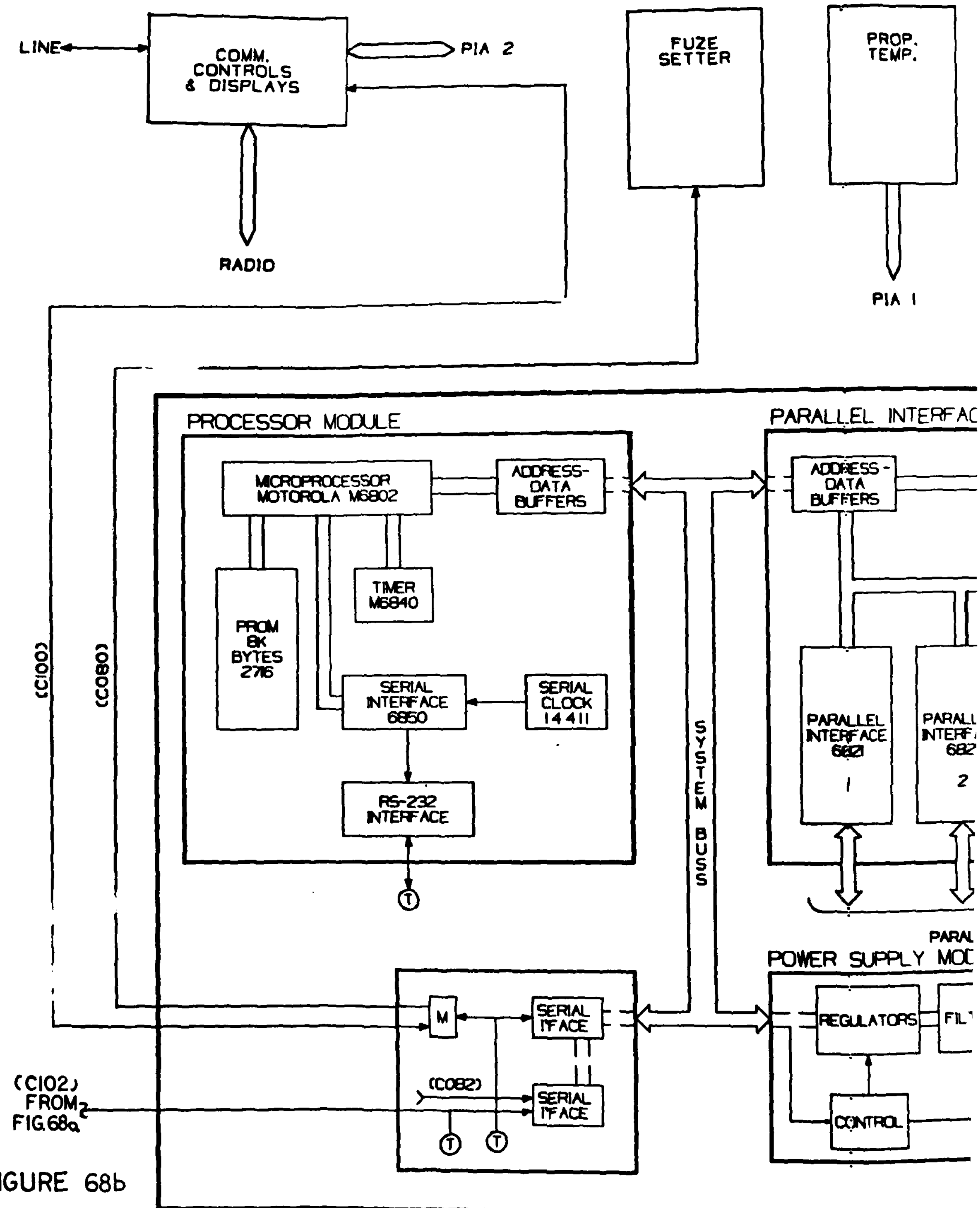
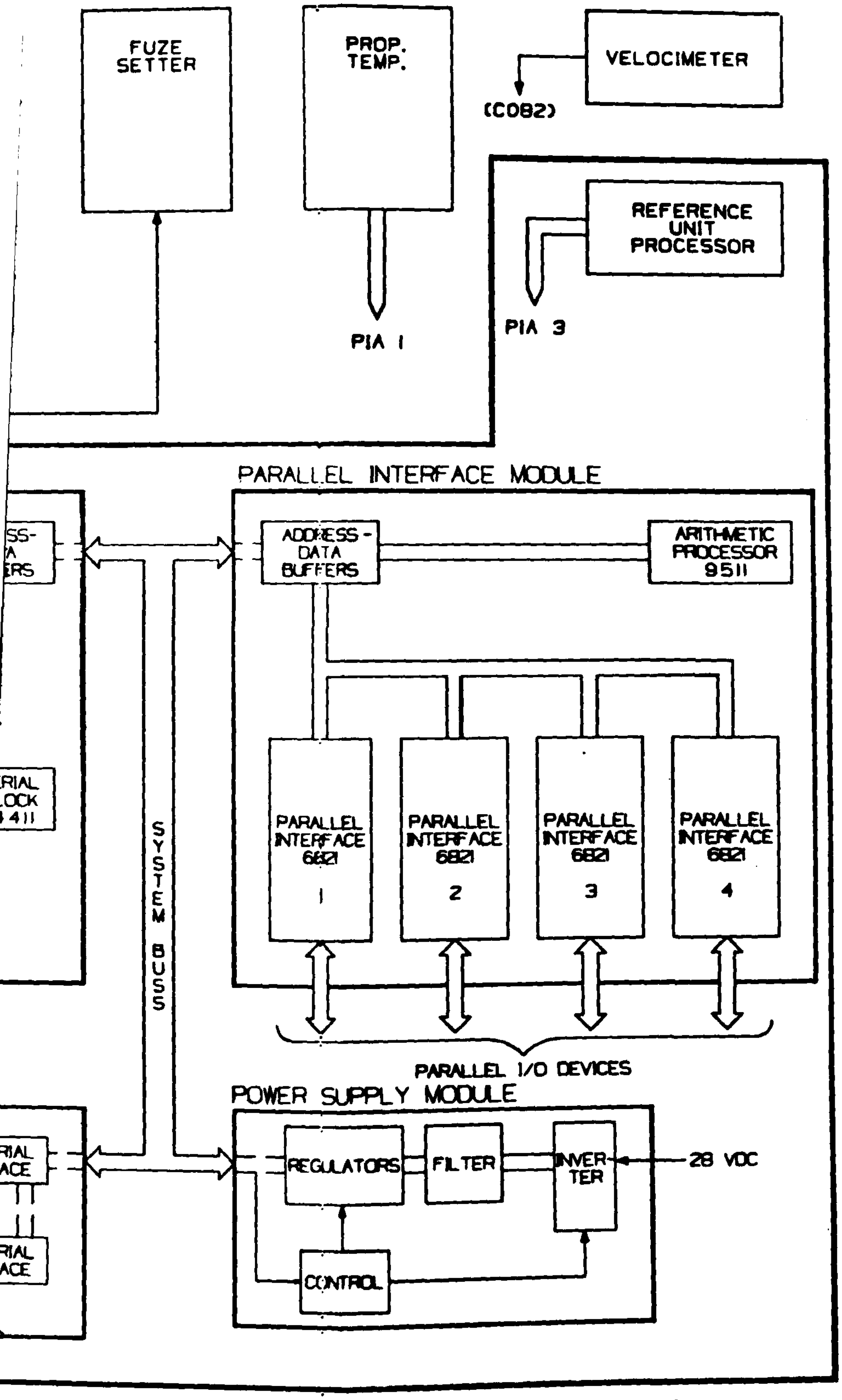
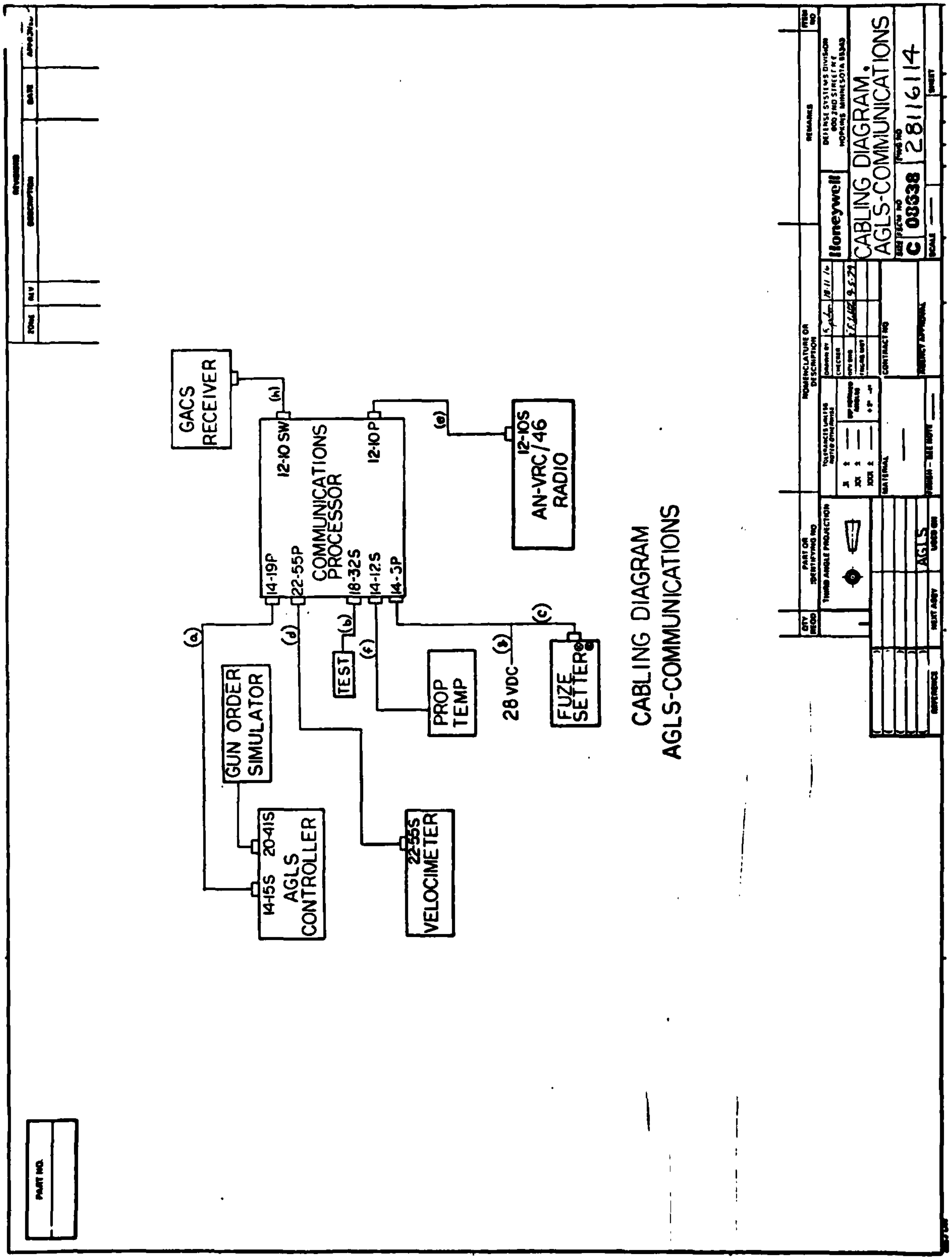

LINE
COMM. CONTROLS & DISPLAYS
PIA 2
RADIO
FUZE SETTER
PROP. TEMP.
PIA 1
IC610
(C100)
(C080)
PROCESSOR MODULE
MICROPROCESSOR MOTOROLA M6802
ADDRESS-DATA BUFFERS
PROM 8K BYTES 2716
TIMER M6840
SERIAL INTERFACE 6850
SERIAL CLOCK 14411
RS-232 INTERFACE
T
SYSTEM BUSS
PARALLEL INTERFAC
ADDRESS-DATA BUFFERS
PARALLEL INTERFACE 6821
1
2
PARAL
POWER SUPPLY MOD
REGULATORS
CONTROL
M
SERIAL I'FACE
(C082)
(C102) FROM FIG.68a

FIGURE 68b


FUZE SETTER
PROP. TEMP.
VELOCIMETER
(C082)
REFERENCE UNIT PROCESSOR
PIA 1
PIA 3
PARALLEL INTERFACE MODULE
ADDRESS - DATA BUFFERS
ARITHMETIC PROCESSOR 9511
PARALLEL INTERFACE 6821 1
PARALLEL INTERFACE 6821 2
PARALLEL INTERFACE 6821 3
PARALLEL INTERFACE 6821 4
SYSTEM BUSS
PARALLEL I/O DEVICES
POWER SUPPLY MODULE
REGULATORS
FILTER
INVERTER
28 VDC
CONTROL



GACS RECEIVER
12-10 SW
COMMUNICATIONS PROCESSOR
12-10P
12-10S
AN-VRC/46 RADIO
14-19P
22-55P
18-32S
14-12S
14-3P
GUN ORDER SIMULATOR
TEST
PROP TEMP
28 VDC
FUZE SETTER
20-41S
14-15S
AGLS CONTROLLER
22-55S
VELOCIMETER
CABLING DIAGRAM
AGLS-COMMUNICATIONS
Honeywell
CABLING DIAGRAM,
AGLS-COMMUNICATIONS
C 08338
28116114
AGLS
PART NO.

Figure 69

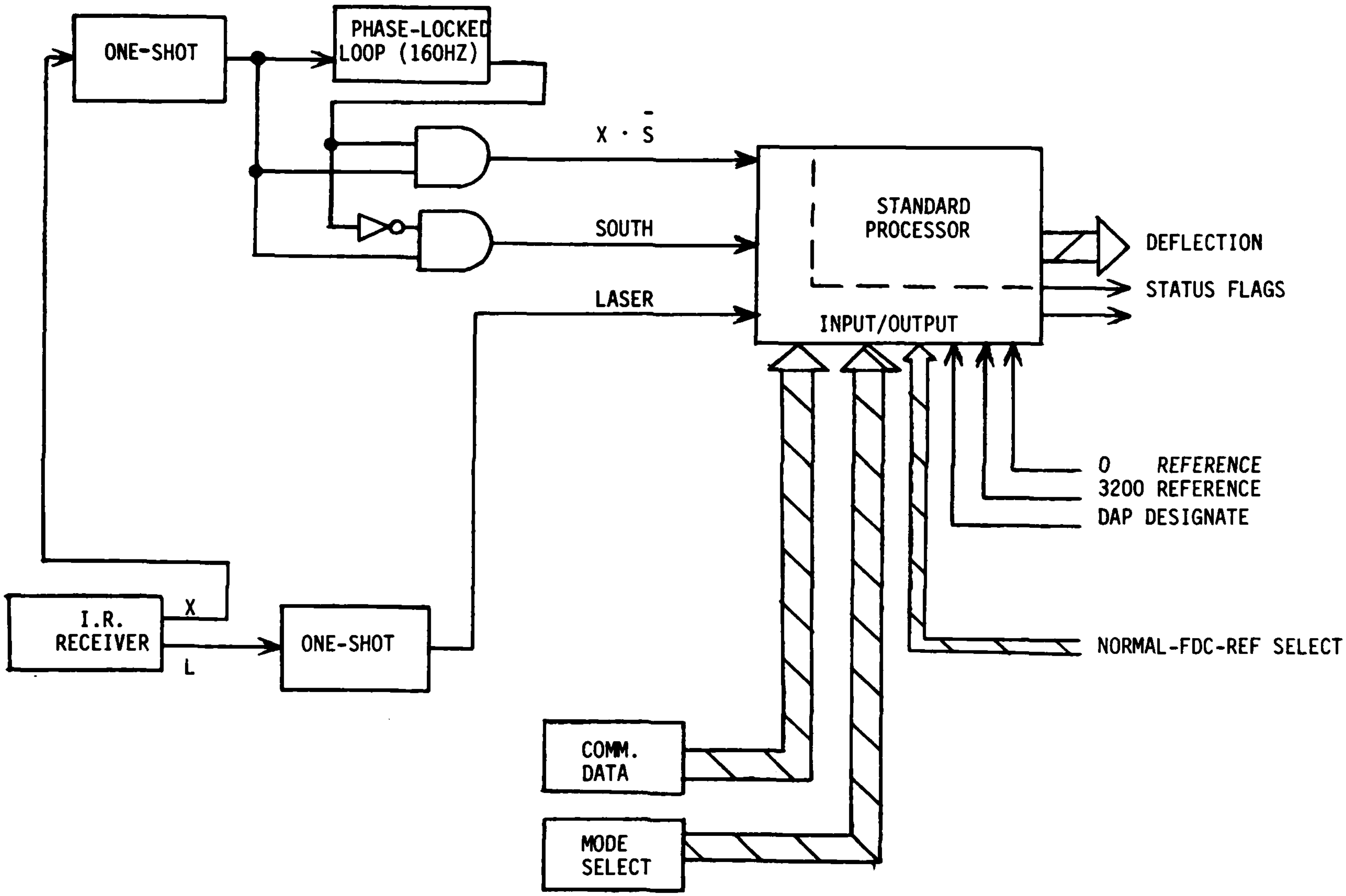


FIGURE 70 BLOCK DIAGRAM - REFERENCE UNIT PROCESSOR

The LASER pulse is passed directly to the processor, since it is not synchronous with the XENON channel.

The time gates established by the phase locked loop eliminate approximately 80% of any spurious pulses which might be detected. Further filtering is performed in the processor software.

The processor DC power supply was designed to convert raw vehicle power to the variety of voltages required to support the microprocessor system. In addition voltage outputs were provided to power analog supporting subsystems such as the GACS detector.

Operator interface to VECOM was via the controls and displays subsystem. This design used a front panel layout as shown in Figure 71. The operation of VECOM via this panel is described below:

On system power up (using the master power switch on AGLS) VECOM is in the STANDBY mode as indicated by the pilot light. The ELEVATION, DEFLECTION, FUZE and CHARGE displays will indicate 0 since no gun orders have been transmitted to the vehicle. If the GACS reference unit has been acquired and good data are being received the GACS lamp will be lit and the X and L monitors will be flashing at an approximate rate of 80/sec and 1/2 sec respectively. The reference angle can be read by switching the MODE switch to the REF position.

The operator initiates a connect to the FDC by moving the momentary COMM/STBY switch to the COMM position. A select is sent by VECOM, acknowledged by FDCOM and a turn around code sent by VECOM. At this point VECOM is in the slave mode (listening) while FDCOM is in the master mode. The COMM lamp will be lit when the system is connected and GACS/RUP operation continues as before. During the exchange of data the monitors 0-5 will flash on and off; the meaning of each is:

Monitor 0 = VECOM is transmitting
Monitor 1 = VECOM is receiving
Monitor 2 = VECOM is processing the message
Monitor 3 = Data Carrier Detect
Monitor 4 = AGLS Comm Link Busy

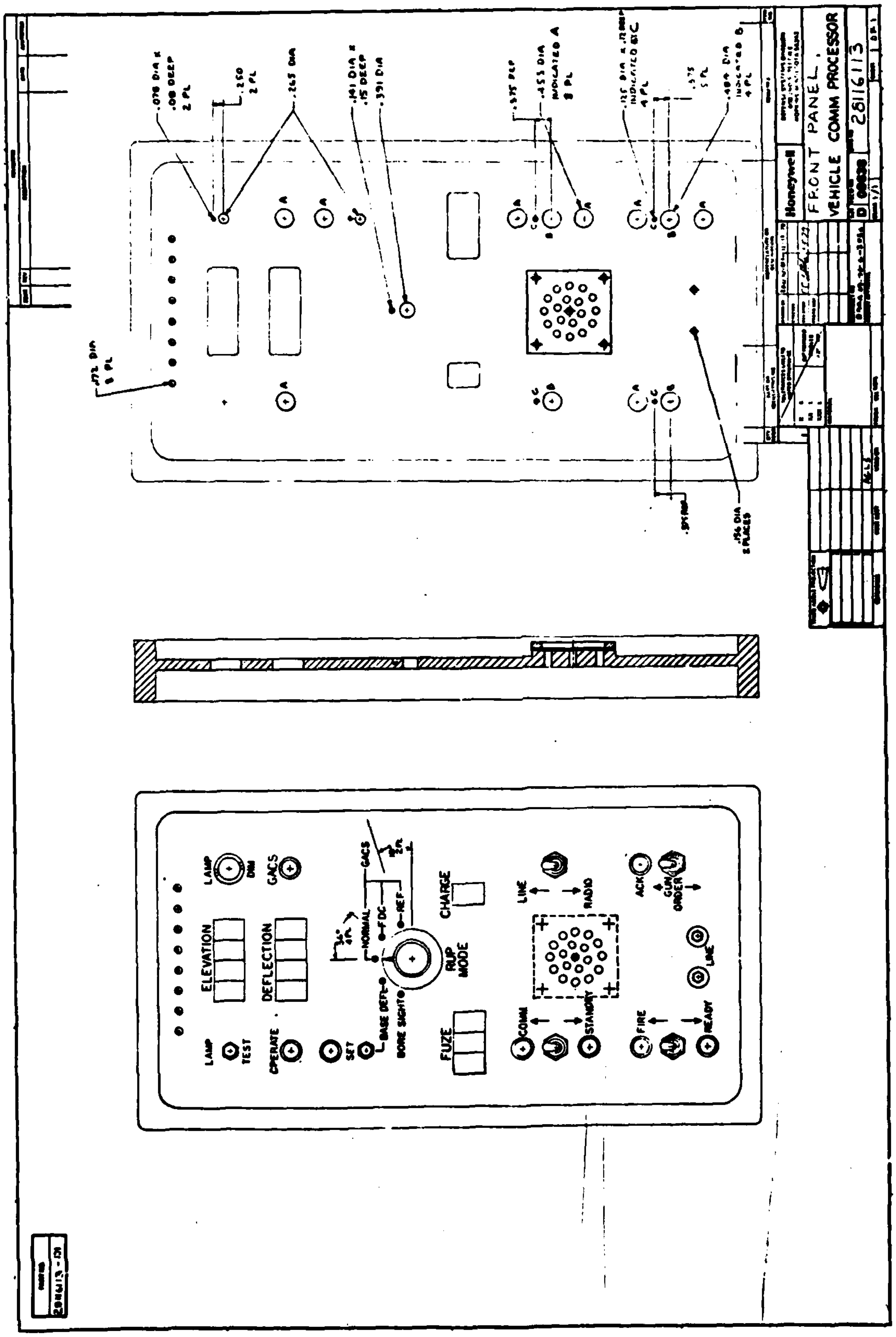

Honeywell
FRONT PANEL,
VEHICLE COMM PROCESSOR
ELEVATION
DEFLECTION
FUZE
CHARGE
RUP MODE
LAMP
TEST
OPERATE
SET
GACS
NORMAL
FDC
REF
BASE DEFL
BORE SIGHT
LINE
RADIO
COMM
STANDBY
ACK
GUN ORDER
FIRE
READY

Figure 71

Monitor 5 = Message NAK

When a gun order is sent from the FDC to VECOM the GUN ORDER ACK lamp is illuminated and the beeper sounds. The respective elevation, deflection, fuze and charge values appear in the displays and are relayed to AGLS. The operator acknowledges receipt of the gun order by moving the momentary switch in either direction. This acknowledgement sends the received gun order back to the FDC for validation and generates an automatic ready request. The ready request lights the READY monitor in VECOM and initiates an automatic sequence of updating the reference angle to AGLS. When the weapon has been laid the READY is acknowledged by moving the momentary switch toward the ready lamp. This action terminates the automatic reference angle update to AGLS and sends the data report (consisting of all AGLS status and numerical data) back to the FDC. When the FDC sends the fire command the FIRE lamp illuminates and the horn sounds. As soon as the shot is fired it is signalled to the FDC by moving the momentary switch toward the FIRE lamp. This action sends data report to the FDC. Upon receipt and acknowledgement of this data report the FDC requests an additional report. This latter report contains the measured projectile velocity from the MVR. Messages need not be received in the aforementioned sequence. Gun orders can be sent sequentially to update other gun orders. Check fires (denoted by a flashing display of 9's and acknowledged by the READY switch) can be issued at any time to halt a mission.

Operation of the system, insofar as communications are concerned, is identical in either the GACS or Base Deflection (BD) modes. In the latter mode, base deflection initialization is required. This is accomplished by selecting Base Deflection Mode, adjusting the AGLS pantel to acquire the distant aiming point (DAP) and depressing the BD SET button. When the BD setting is locked into the AGLS processor the SET lamp will be lit. One can either operate in this mode or switch back to GACS mode; the BD value remains locked into the computer unless it is powered down. In operation, in BD mode, the preset reading obtained from the pantel and stored during the set operation, is subtracted from all subsequent absolute encoder readings such that if the pantel is directed at the DAP the ACTUAL azimuth reading would be 3200.0. In the semi-automatic mode of operation (automatic azimuth offset) azimuth gun orders from the FDC are directed to the pantel to drive it to the specified angle. The cab must then be rotated by the gunner through the power handle to acquire the DAP sight picture; the weapon is then laid in azimuth.

The front panel VECOM displays and controls were designed to minimize Chief of Section (COS) workload. In the fully automatic mode of operation only one control is required from the AGLS COS panel and that is the weapon lay enable (WPN).

The software designed for VECOM is shown in flow form in Figure 72 and the associated assembly level source code is contained in Appendix B.

The reference unit processor, being sufficiently different from any other processor designed until now, was programmed using the methods of top down software designs. The circuit elements, shown in the block diagram, consist of the GACS IR Receiver, a phase-locked loop, and a standard microprocessor with input/output, random access memory (RAM), program memory and a central processing unit (CPU).

The inputs to the RUP are three pulses:

X = A XENON pulse occurring every 40 mils of LASER rotation.

S = A pulse occurring once for every 160 valid X pulses, spaced between two X pulses.

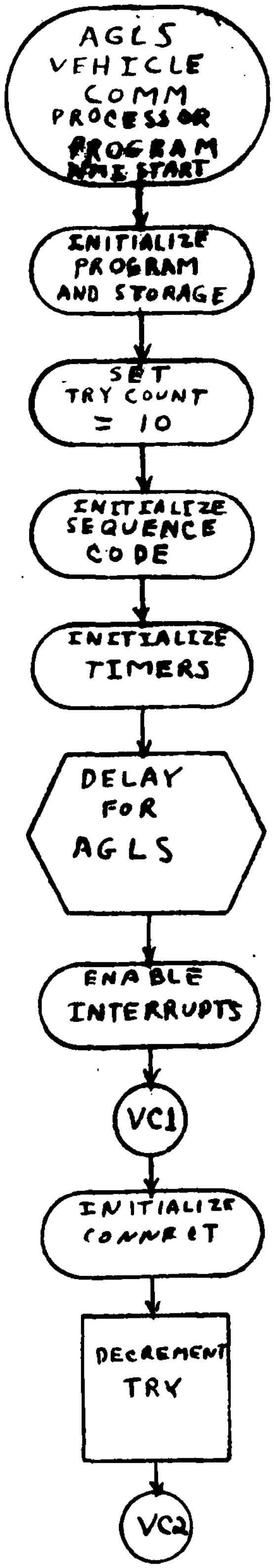
L = A pulse occurring once for every S pulse, at any timing including coincidence with an S pulse or an X pulse.

The outputs of the RUP are to be:

- The reference angle from 0 to 6399 in binary coded decimal, and
- A status flag showing that the currently computed reference angle is valid.

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

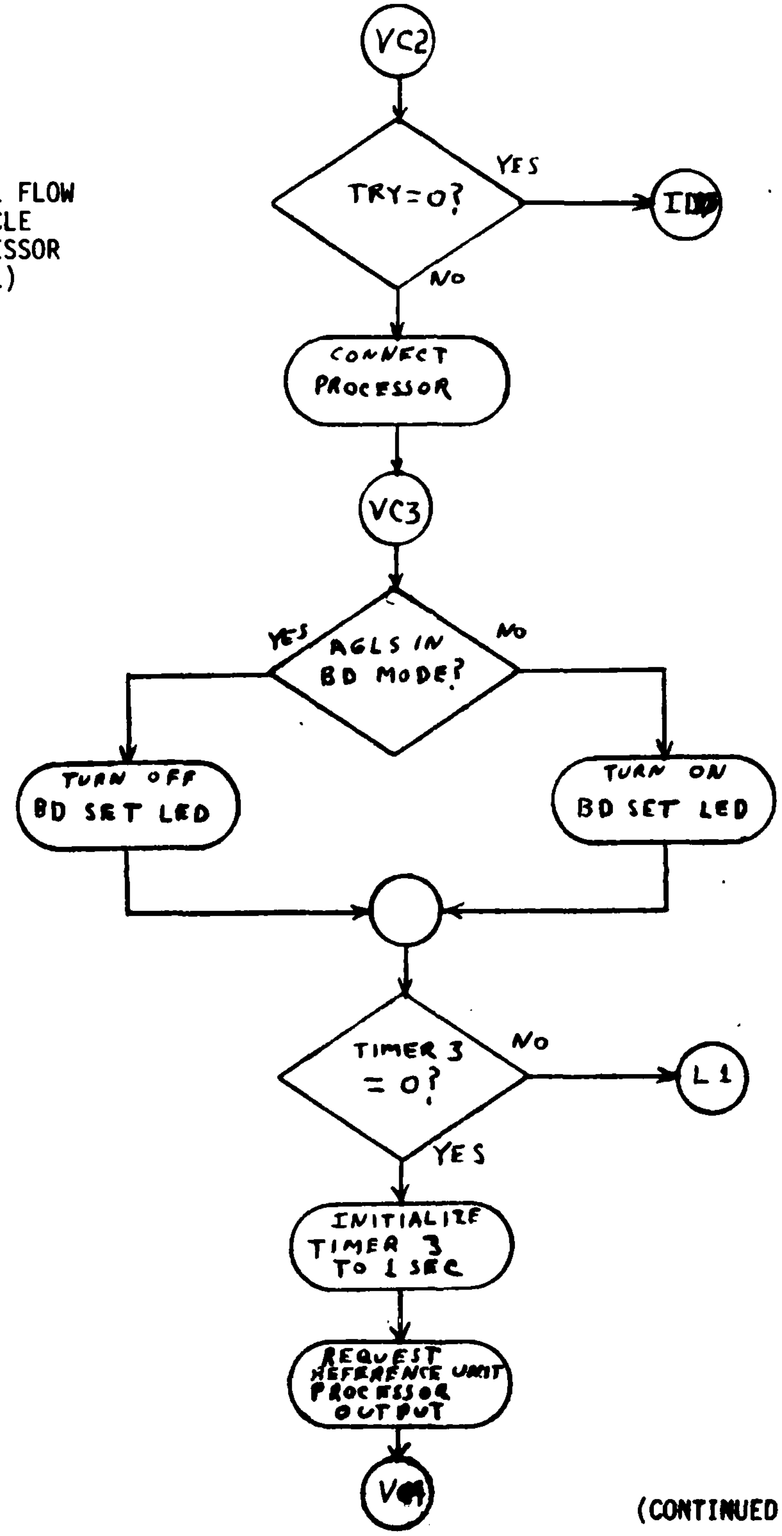
FIGURE 72a



(CONTINUED ON FIGURE 72b)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

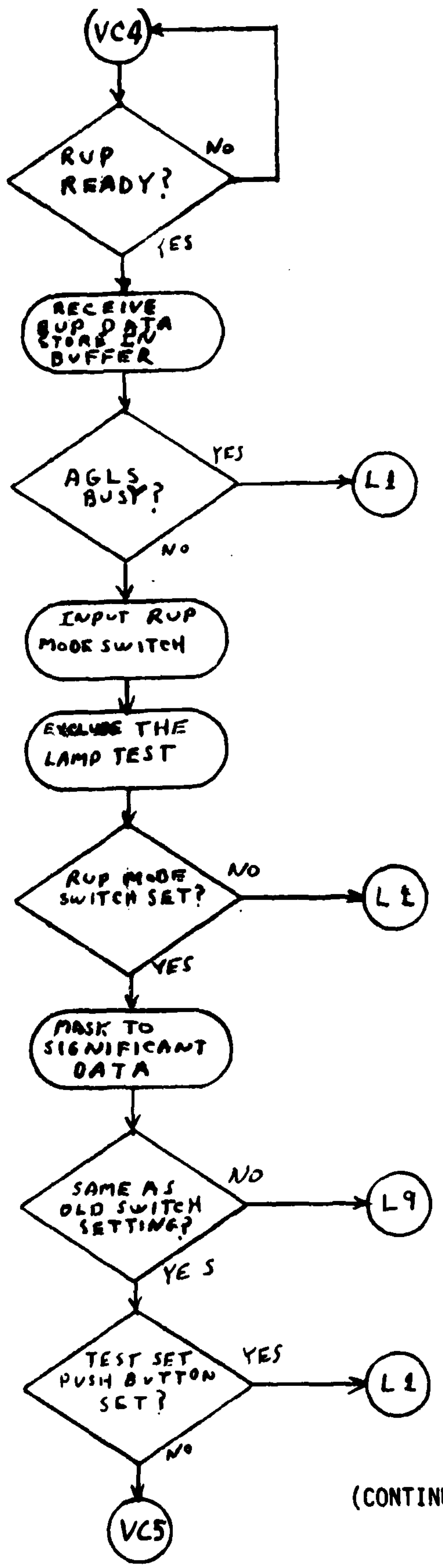
FIGURE 72b



(CONTINUED ON FIGURE 72c)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

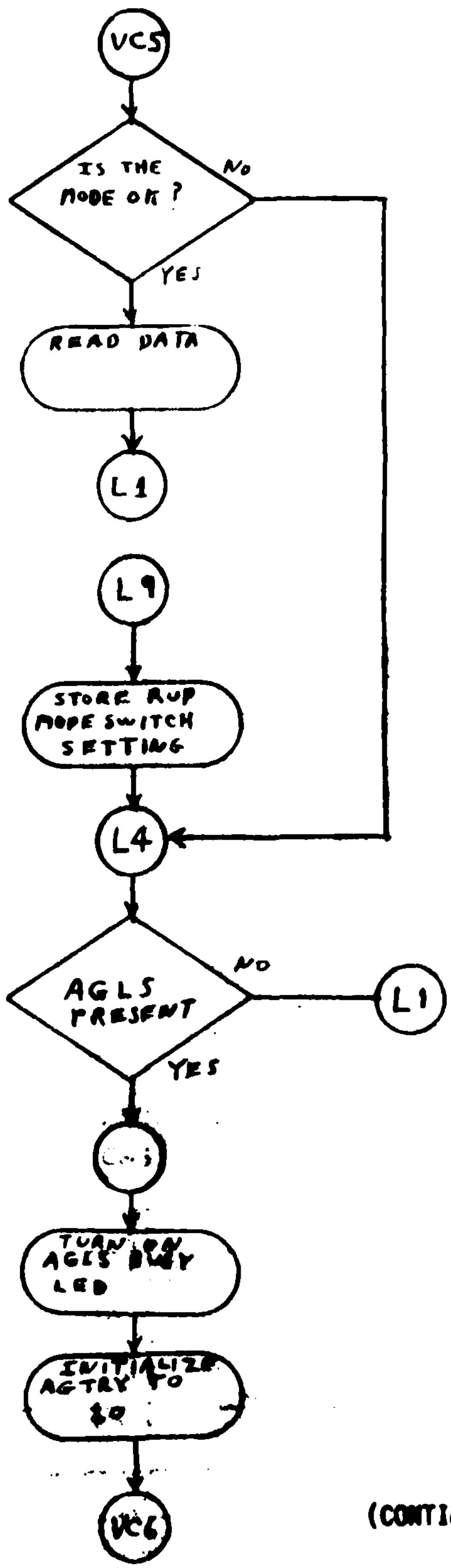
FIGURE 72c



(CONTINUED ON FIGURE 72d)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

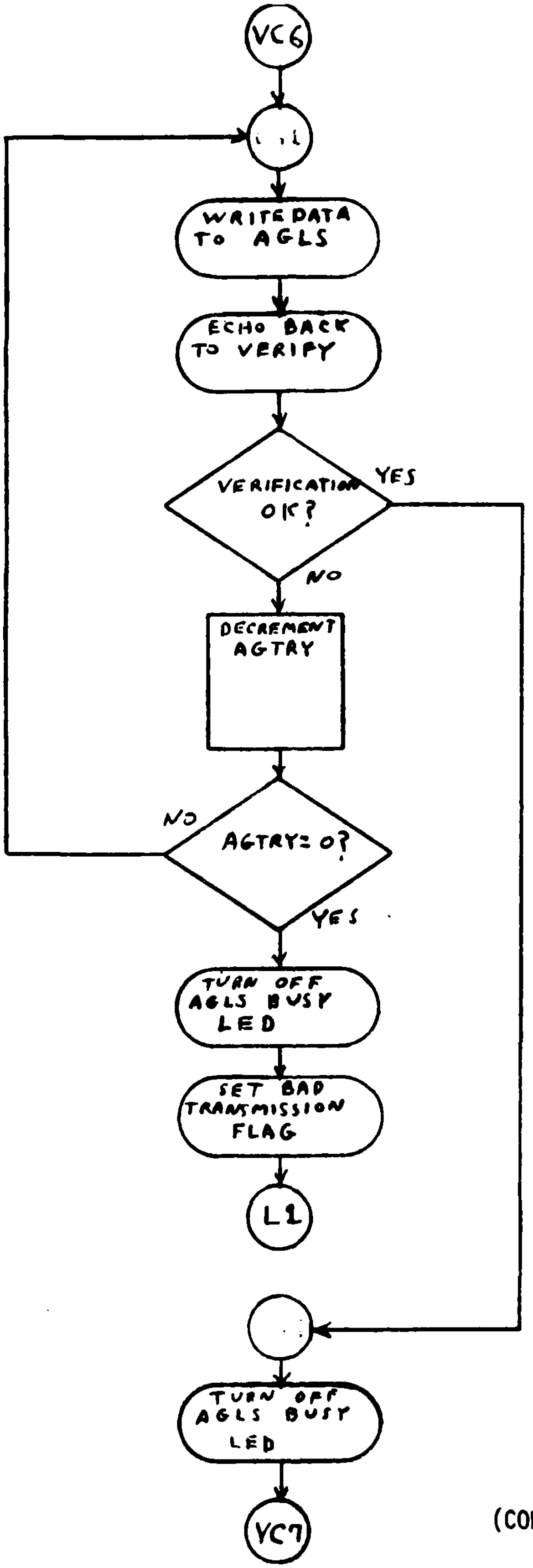
FIGURE 72d



(CONTINUED ON FIGURE 72e)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

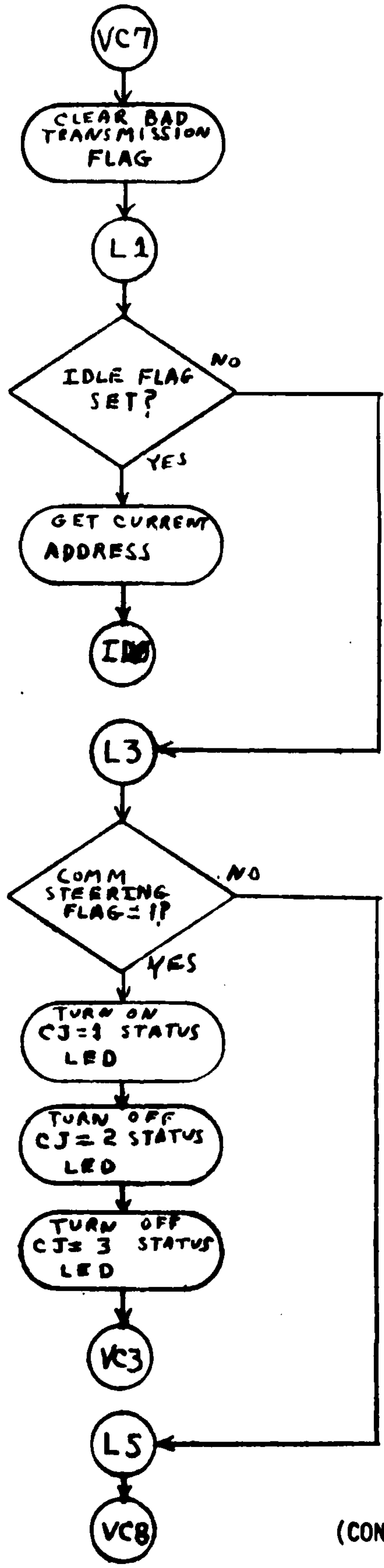
FIGURE 72e



(CONTINUED ON FIGURE 72f)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

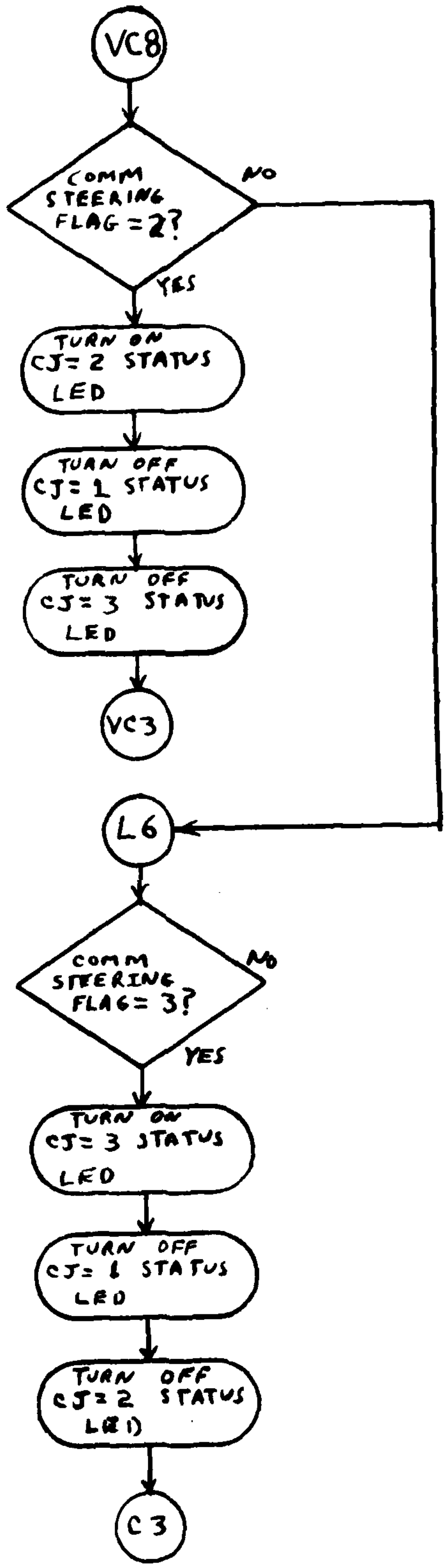
FIGURE 72f



(CONTINUED ON FIGURE 72g)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

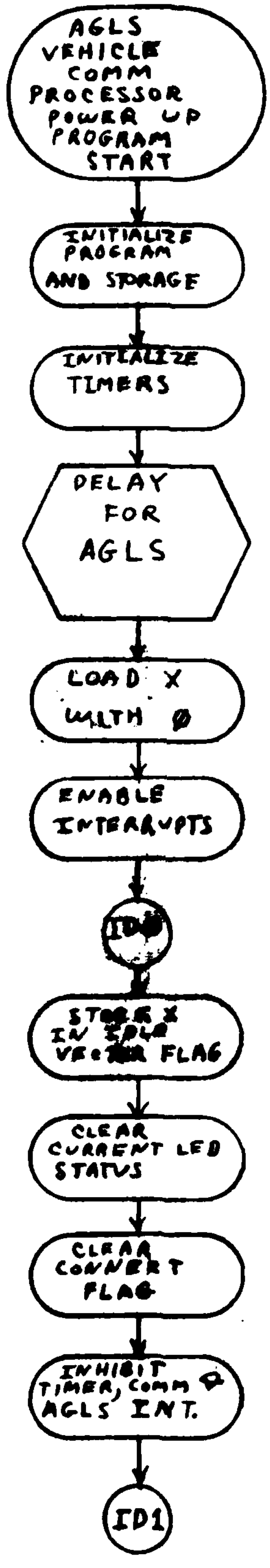
FIGURE 72g



(CONTINUED ON FIGURE 72h)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

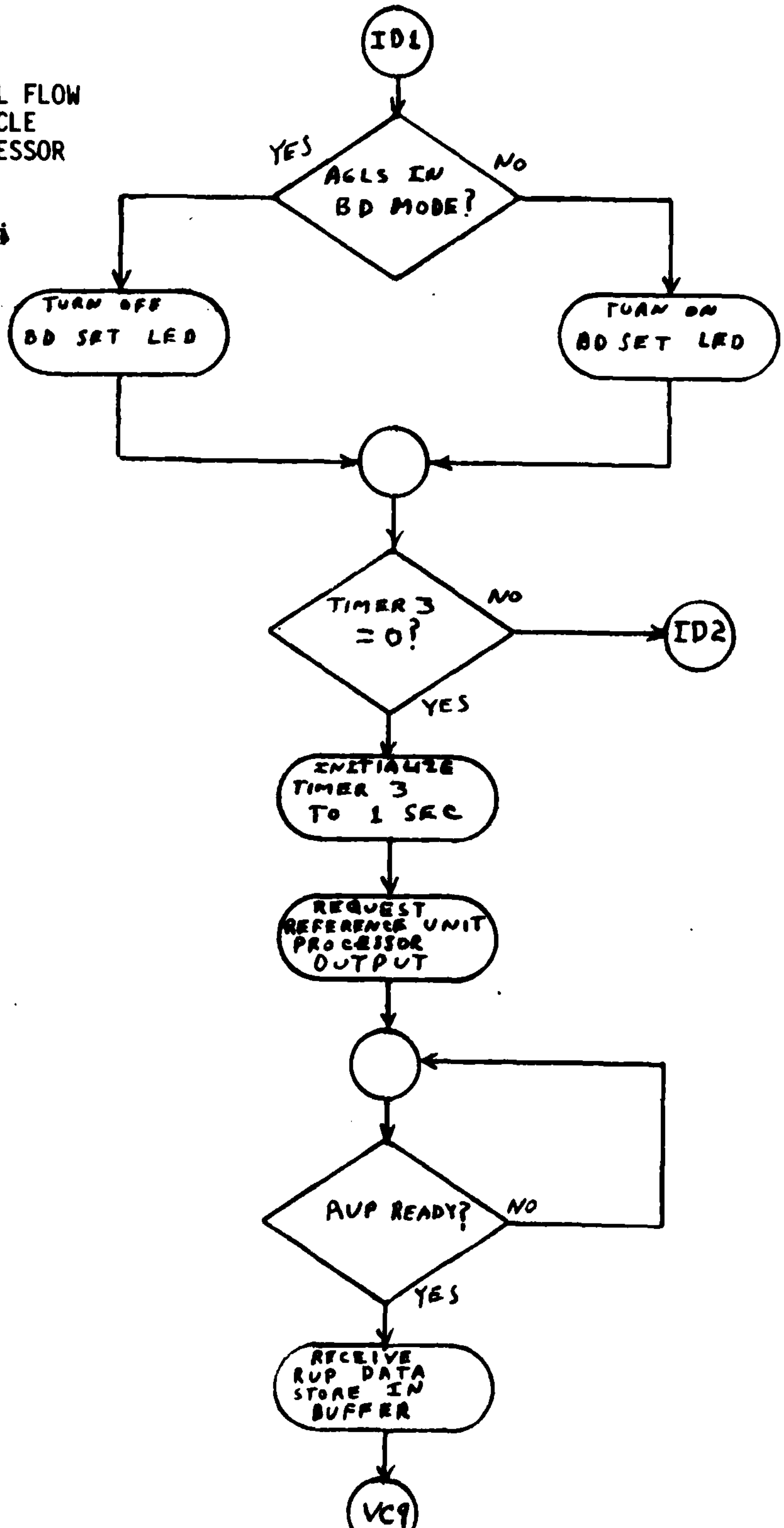
FIGURE 72h



(CONTINUED ON FIGURE 72i)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

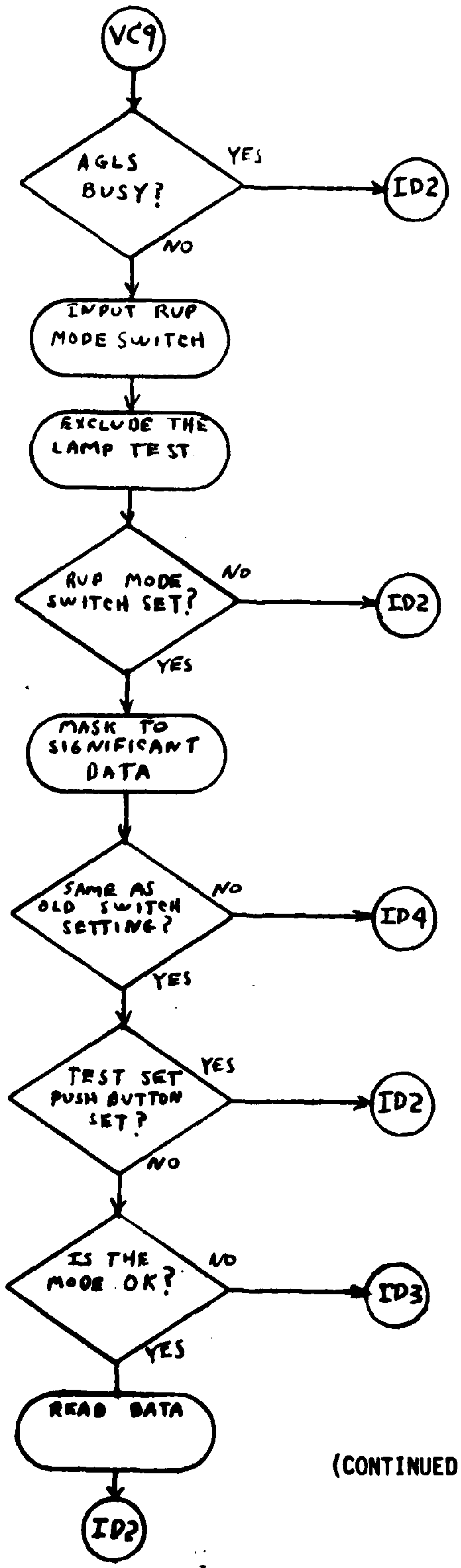
FIGURE 72i



(CONTINUED ON FIGURE 72j)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

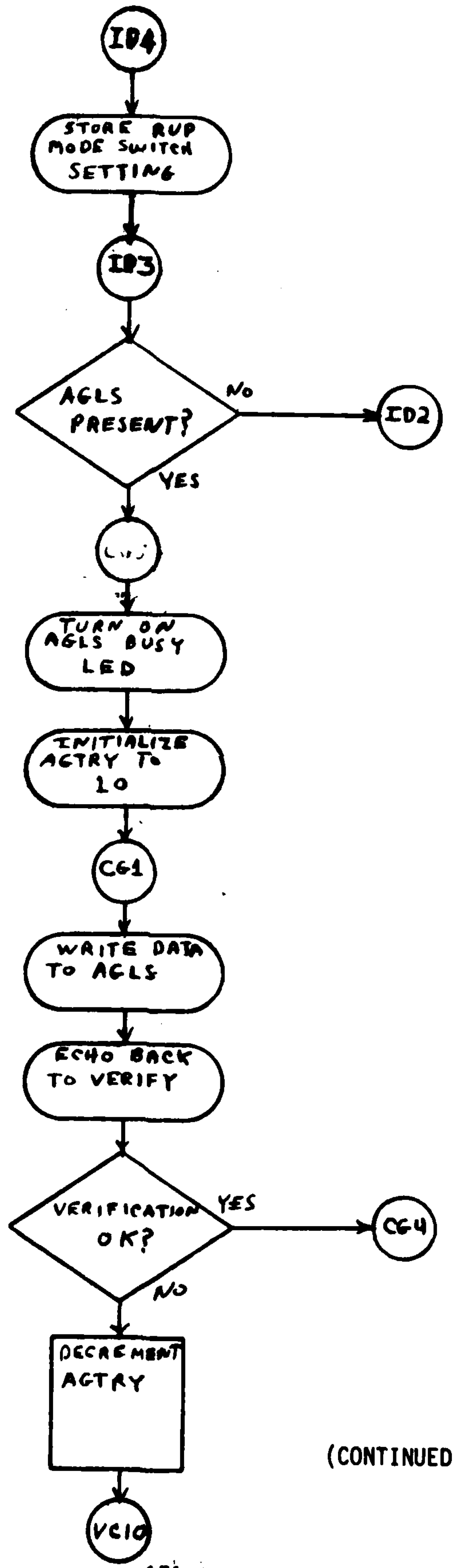
FIGURE 72j



(CONTINUED ON FIGURE 72k)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

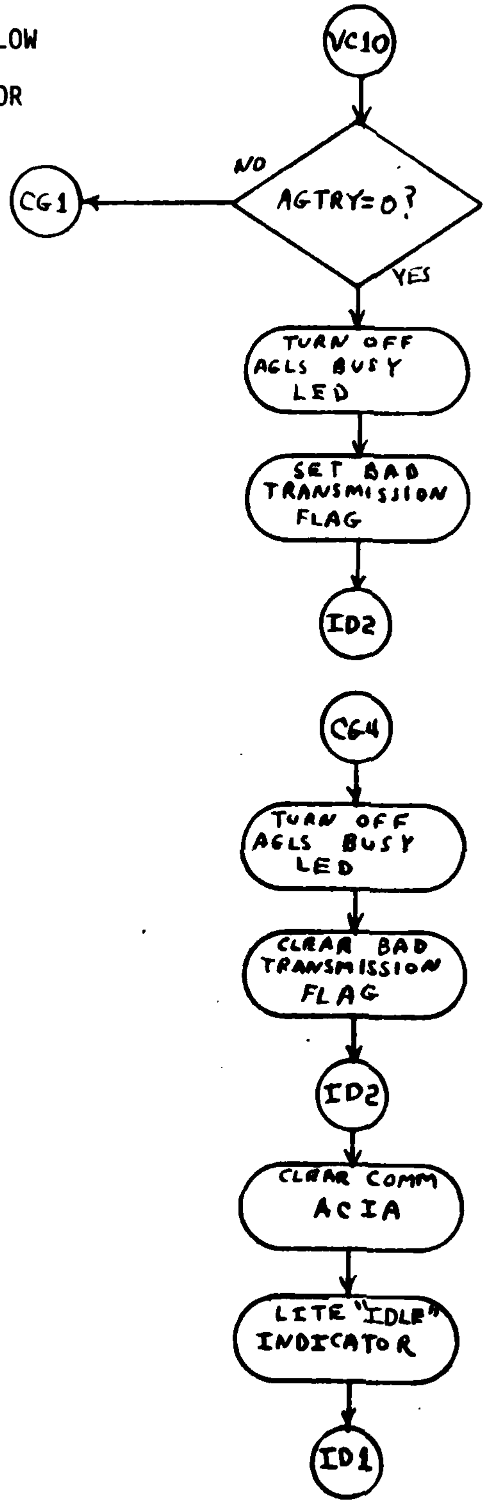
FIGURE 72k



(CONTINUED ON FIGURE 72l)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

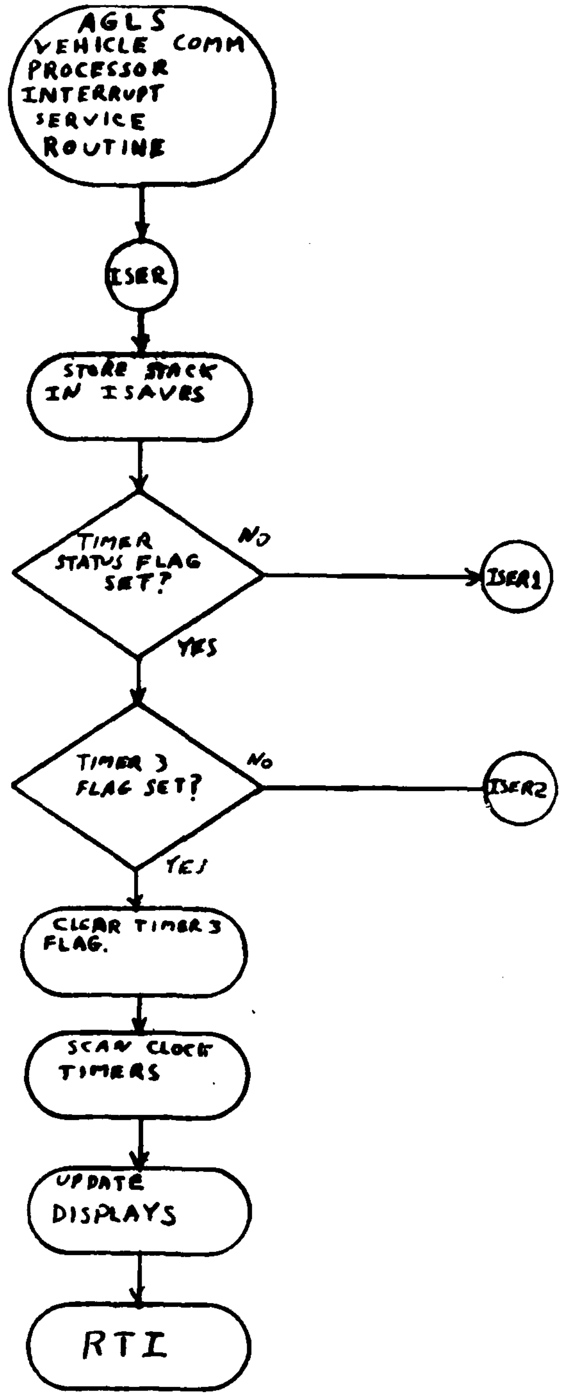
FIGURE 72l



(CONTINUED ON FIGURE 72m)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

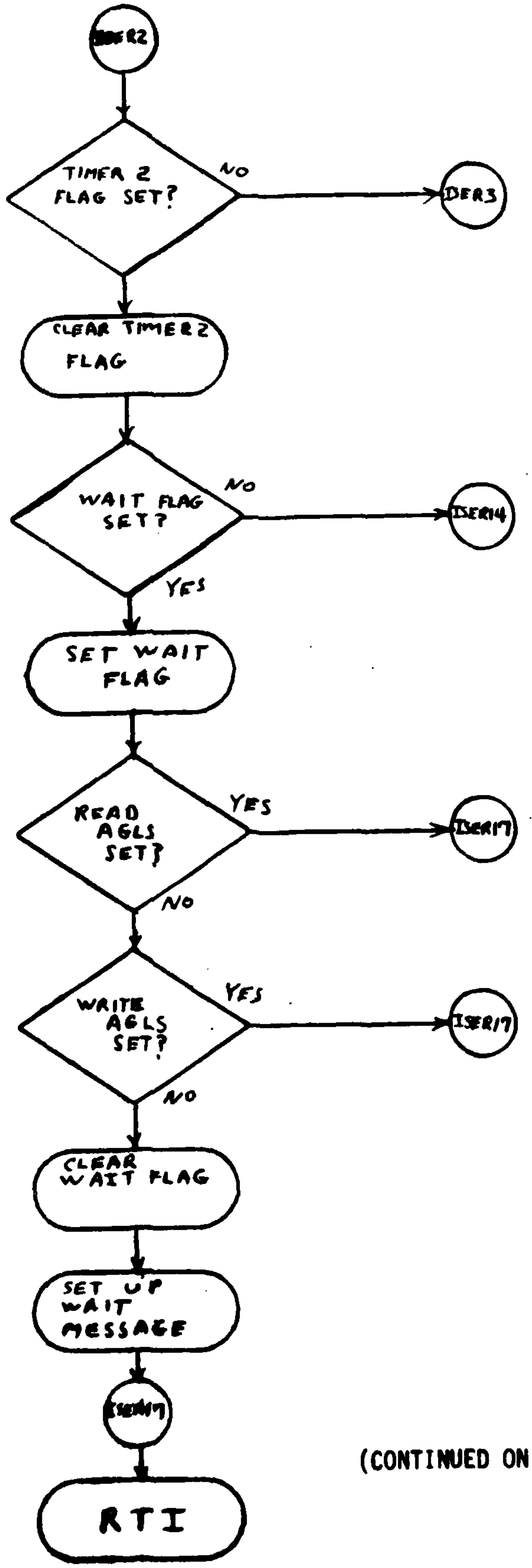
FIGURE 72m



(CONTINUED ON FIGURE 72n)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

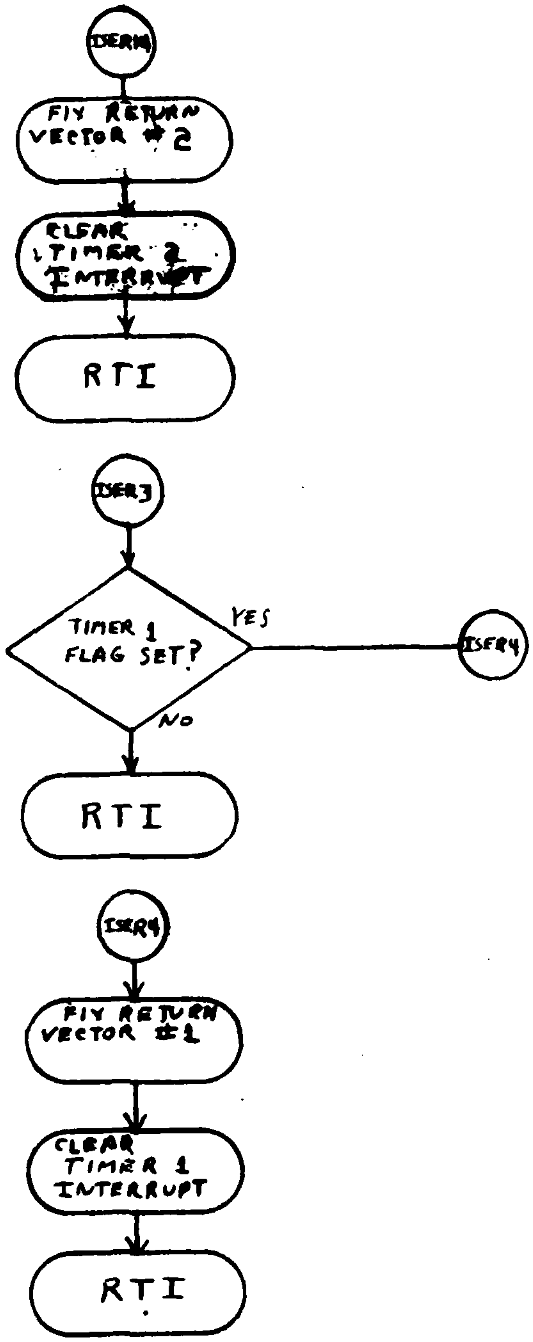
FIGURE 72n



(CONTINUED ON FIGURE 72o)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

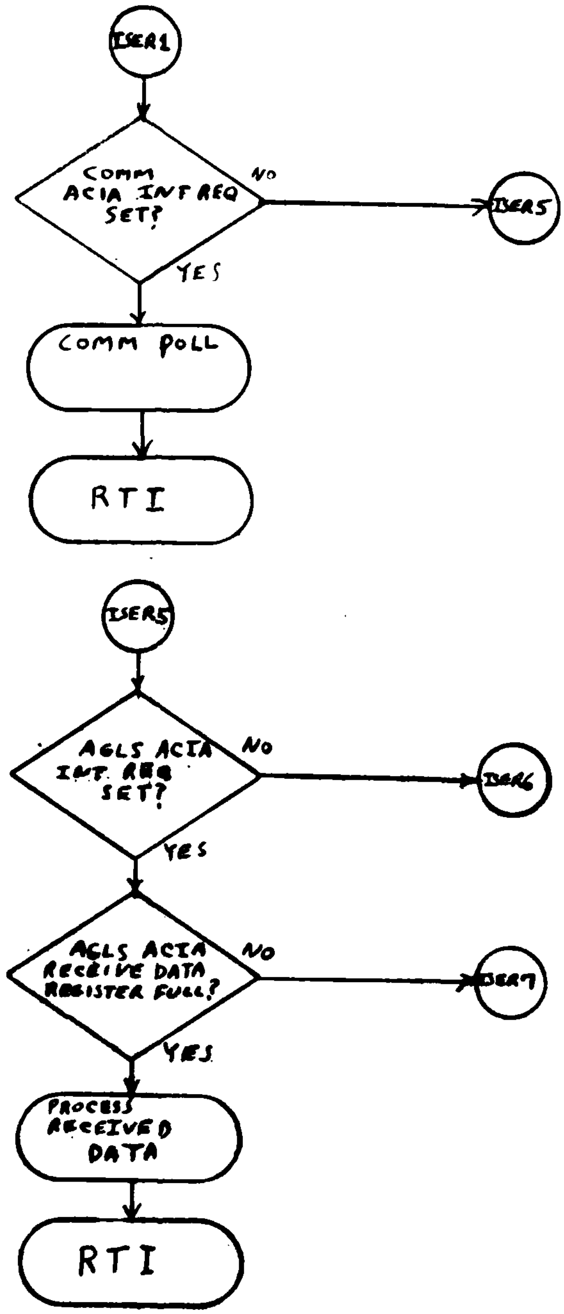
FIGURE 72o



(CONTINUED ON FIGURE 72p)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

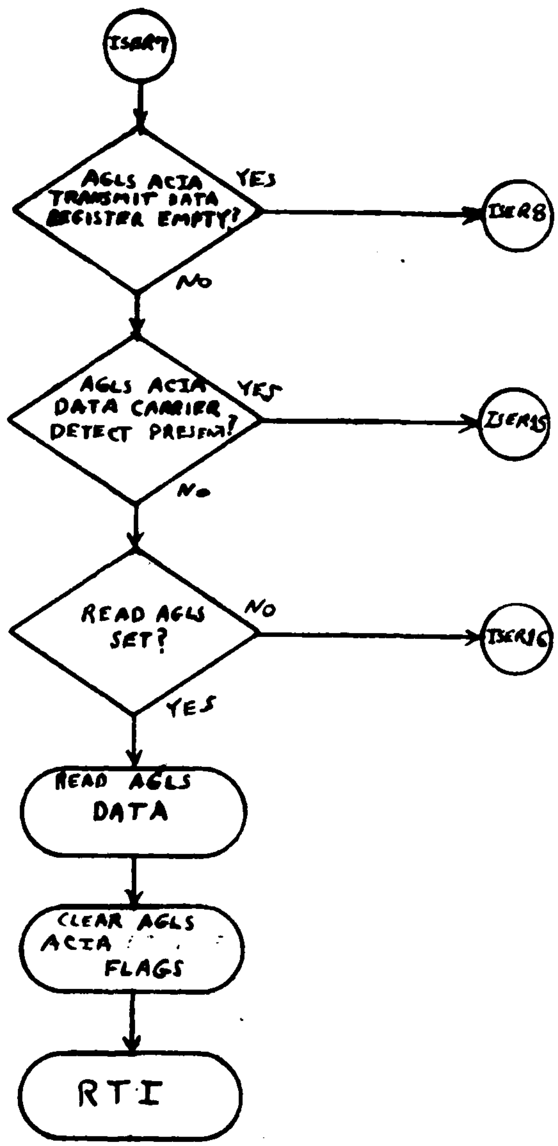
FIGURE 72p



(CONTINUED ON FIGURE 72q)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

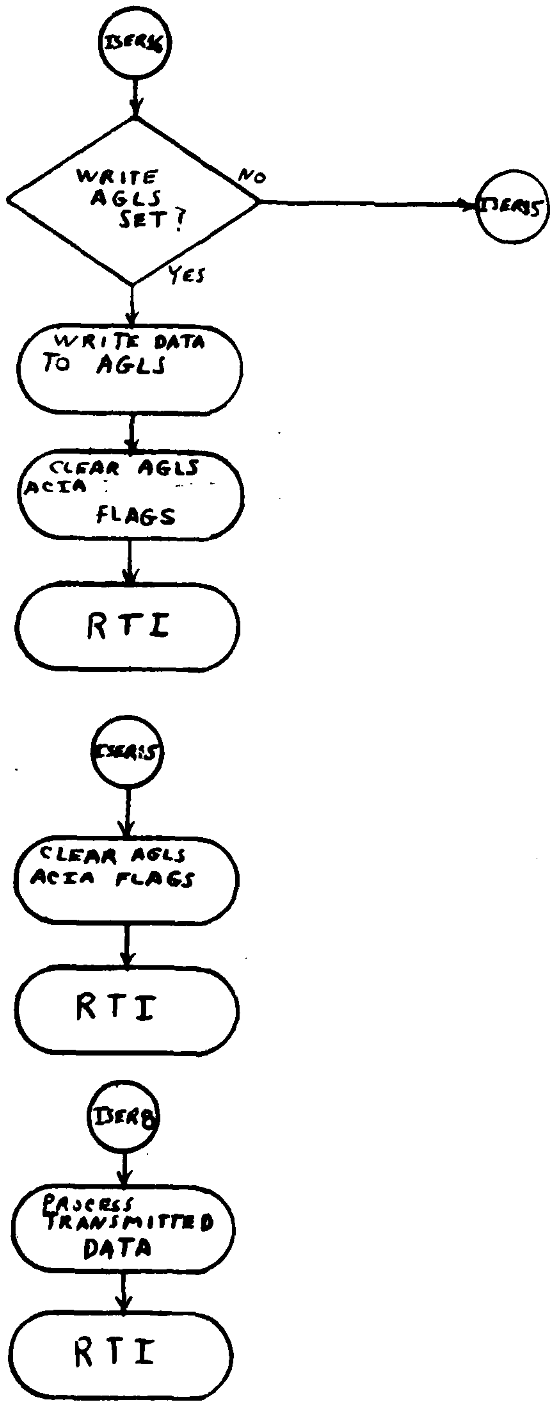
FIGURE 72q



(CONTINUED ON FIGURE 72r)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

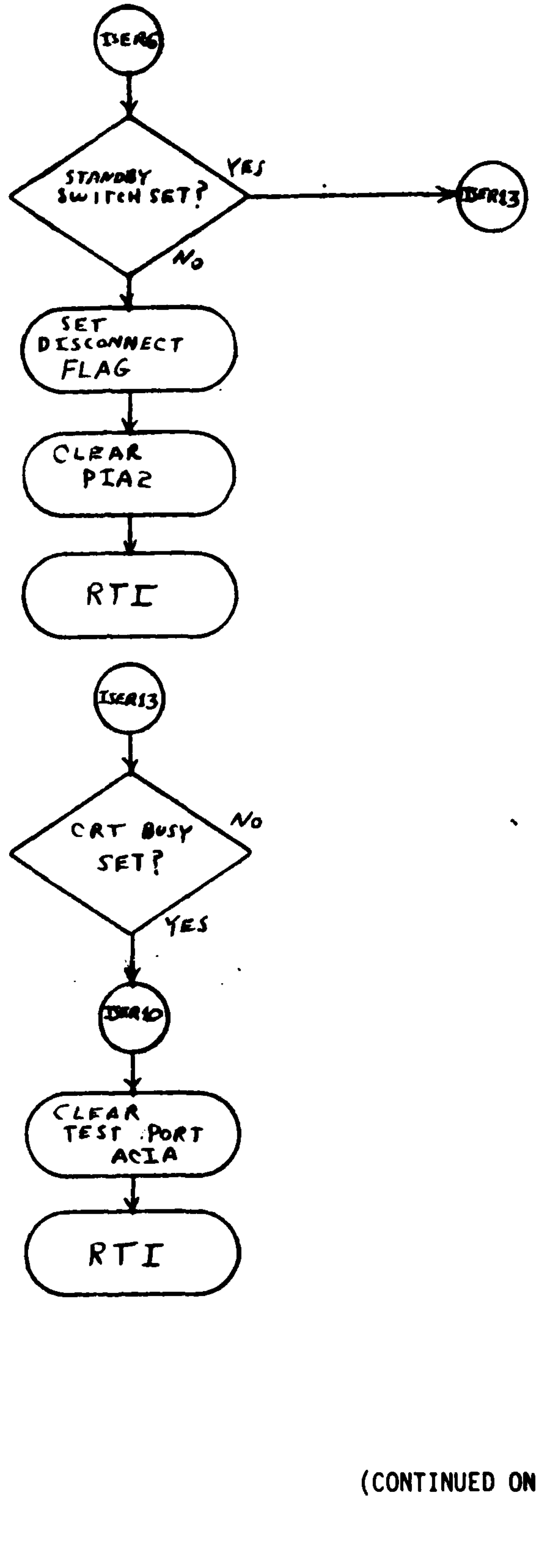
FIGURE 72r



(CONTINUED ON FIGURE 72r)

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

FIGURE 72s



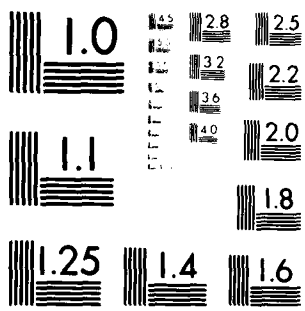
(CONTINUED ON FIGURE 72s)

AD-A097 521 HONEYWELL INC HOPKINS MN DEFENSE SYSTEMS DIV F/G 19/6
AUTOMATED GUN LAYING SYSTEM FOR SELF-PROPELLED ARTILLERY WEAPON--ETC(U)
MAY 80 E E LEHTOLA, K A HERZING DAAA09-76-C-0284

UNCLASSIFIED NL

3 of 5

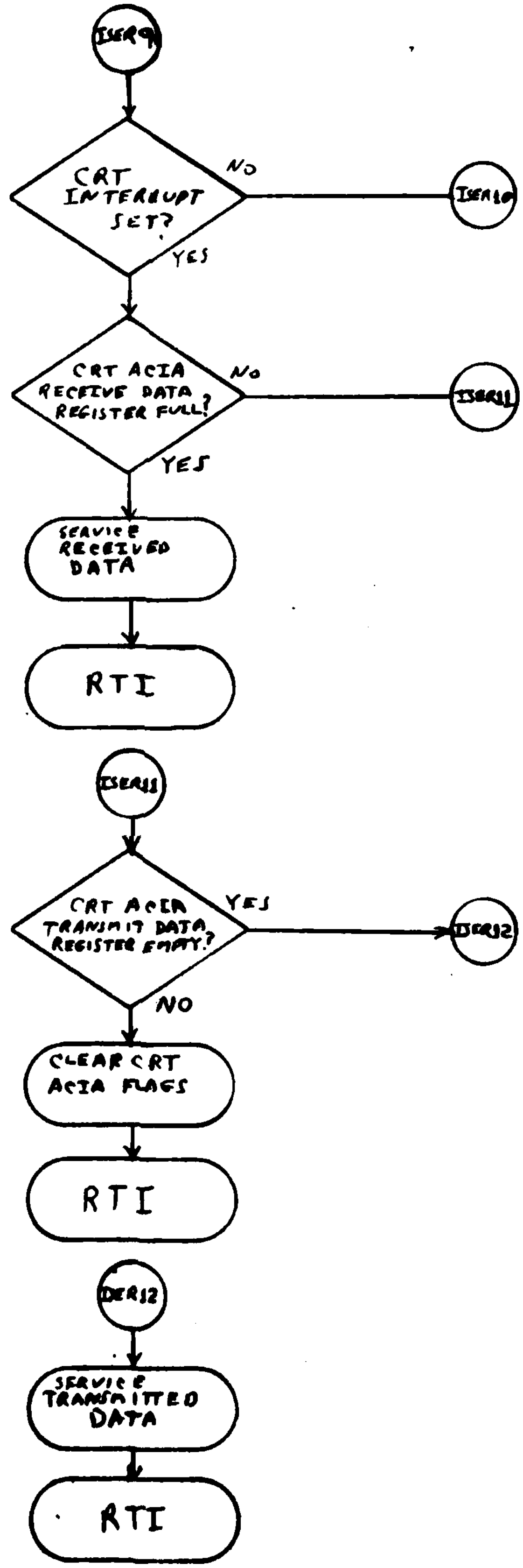
AD A097521



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

FUNCTIONAL FLOW
AGLS VEHICLE
COMM PROCESSOR
(PARTIAL)

FIGURE 72t



After the configuration was defined, a software specification was generated, briefly as follows:

- Three software timers counting from 0 to 159 shall be provided each with a status bit and a cycle counter counting complete cycles from 0 to 7.

- A timer will be started on the first "X" pulse immediately following "S" pulse, if no other timer is at count 159 and active.

- If a timer is at count 159 and active, and an "S" pulse arrives, that timer stays active and rolls over to count 0 on the "X" pulse immediately following the "S" pulse. Its cycle counter will be incremented if not at the maximum value of 7.

- If a timer is at count 159 and active, and an "X" pulse arrives without a preceding "S" pulse, that timer goes inactive, and its cycle count is cleared to zero.

This software configuration ensures that an erroneous reference angle can not be computed; and the correct count can be determined by examining the cycle counter and status bit of the three counts.

One additional check is to verify that either the "S" and the "L" pulse are alternating, or they are coincident. If true, and if at least one counter is active, the processor determines that a good GACS measurement exists.

The total reference angle is computed by the above counter, plus an interpolation which resolves the time interval during which the LASER is detected. The reference angle then is determined to be:

$$\text{Output} = 40 \times (\text{count}) + \frac{40\ T_L}{T_X}$$

where: Count = Value of counter with highest value in cycle counter
T_L = System clock cycles from last "X" pulse to "L" pulse
T_X = System clock cycles from ("X" pulse preceding "L" pulse) to ("X" pulse following "L" pulse).

The software flow diagram for the RU program is shown in Figure 73 and the associated assembly level source code in Appendix C.

7. Fire Direction Center Communication Processor (FDCOM)

The fire direction center located function of the add on scope of work consisted of the FDC communication processor interfaced to the FDC PDP-11/34 computer. This processor provided two-way communication between the FDC and VECOM. The processor was mounted in a 6" high rack cage assembly to facilitate installation in ARRADCOM's Automated FDC trailer. A functional block diagram of FDCOM is shown in Figure 74 and the corresponding schematics are contained in Appendix D. This processor used the common board set H-10 microprocessor system which allowed complete interchangeability between it and its vehicle mounted counterpart. One unique circuit card was fabricated for FDCOM that being the PDP-11/34 interface board. This board was required to make the respective system I/O port electrical characteristics compatible.

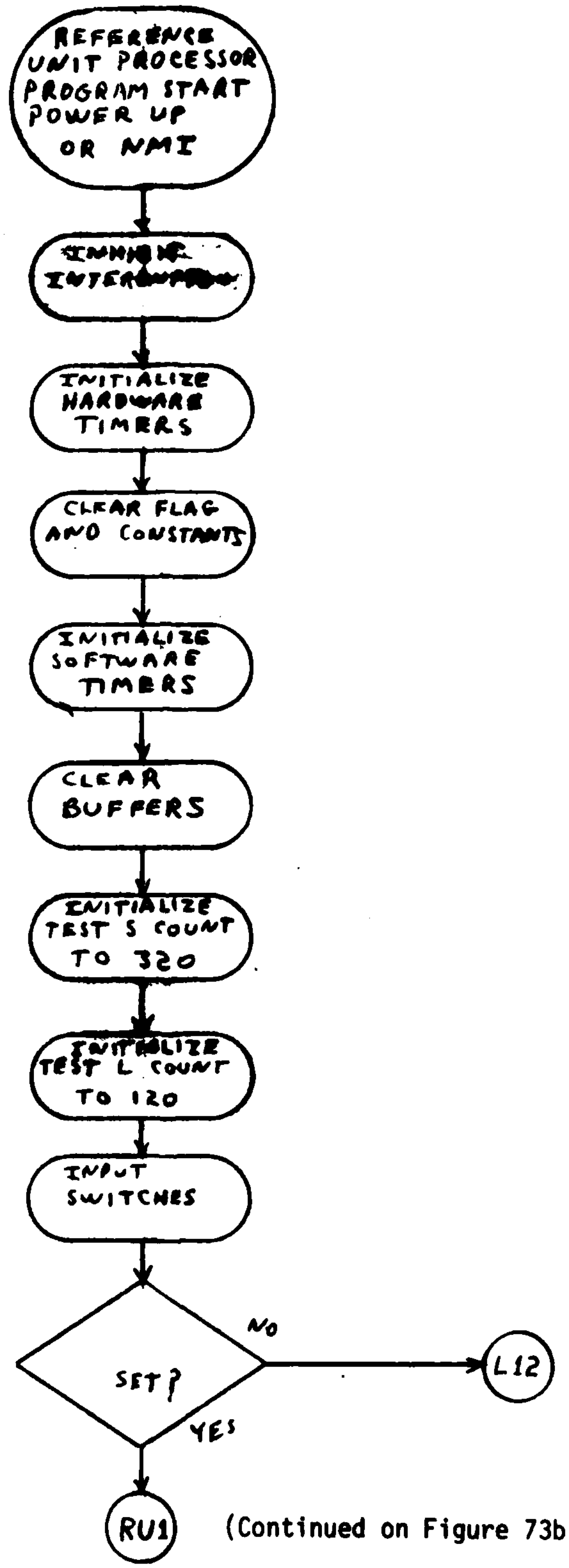
On system power up (or RESET) FDCOM goes into the STANDBY mode. In this mode the receiver circuits are enabled waiting a SELECT or sign on message from a vehicle-borne VECOM. Upon receipt of a SELECT the message is checked for validity and an acknowledgement is sent to the vehicle. The line-turn around message is then sent to place the vehicle processor in the slave mode and FDCOM in the master mode.

When a message is received from the FDC computer the transmit sequence is initiated. The first message out is usually the gun order, which is returned upon acknowledgement by the Chief of Section. If another gun order or check fire has not been received by FDCOM from the FDC computer in the interim, a ready request is sent to the vehicle. If either the gun order update or check fire was received from the FDC computer, FDCOM sends that next. All acknowledgements from the vehicle and the data messages that go with them are made available to the FDC computer via an interrupt driven output buffer.

For checkout purposes, or in case of FDC computer failure, a background package was written for FDCOM that allowed entry of gun orders and display of vehicle responses via a terminal.

REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

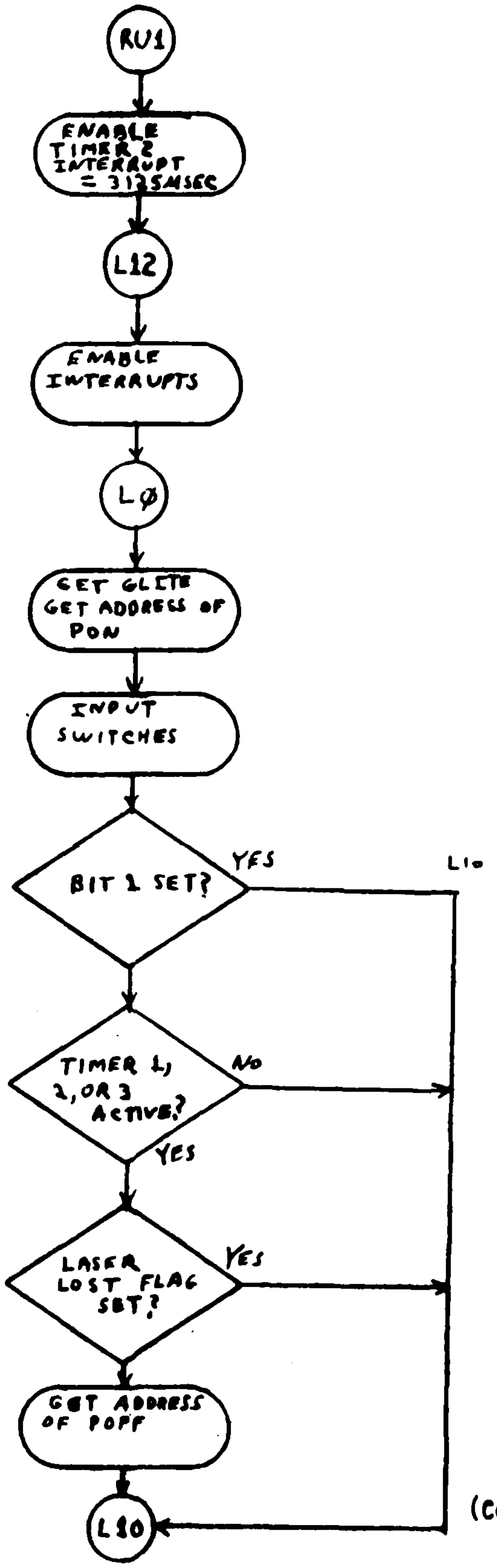
Figure 73a



(Continued on Figure 73b)

REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

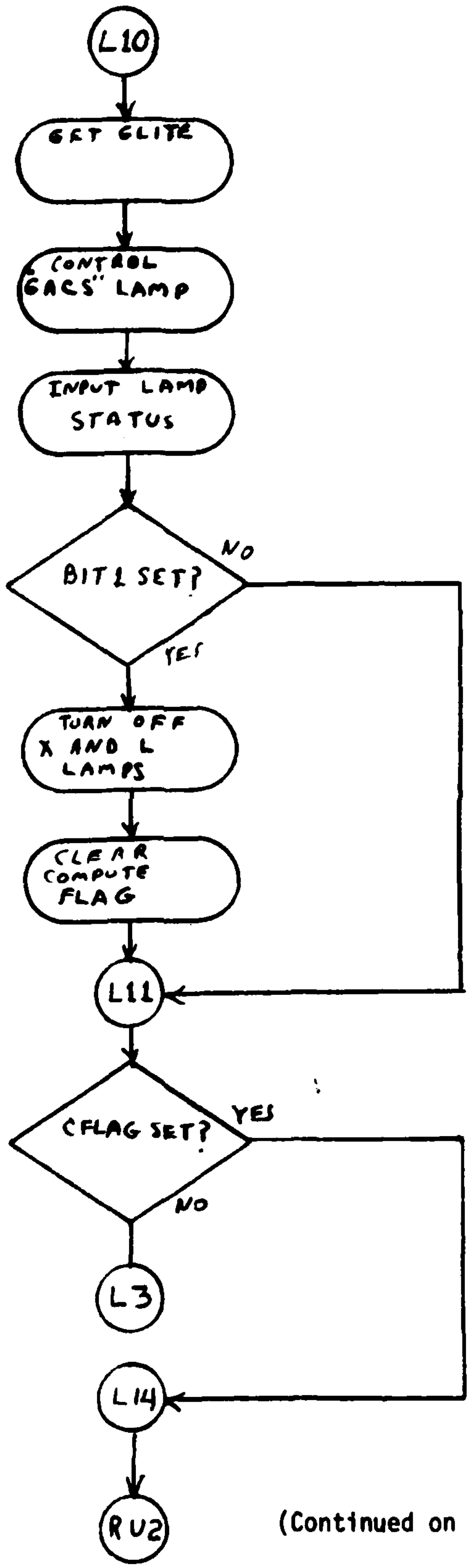
Figure 73b



(Continued on Figure 73c)

REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

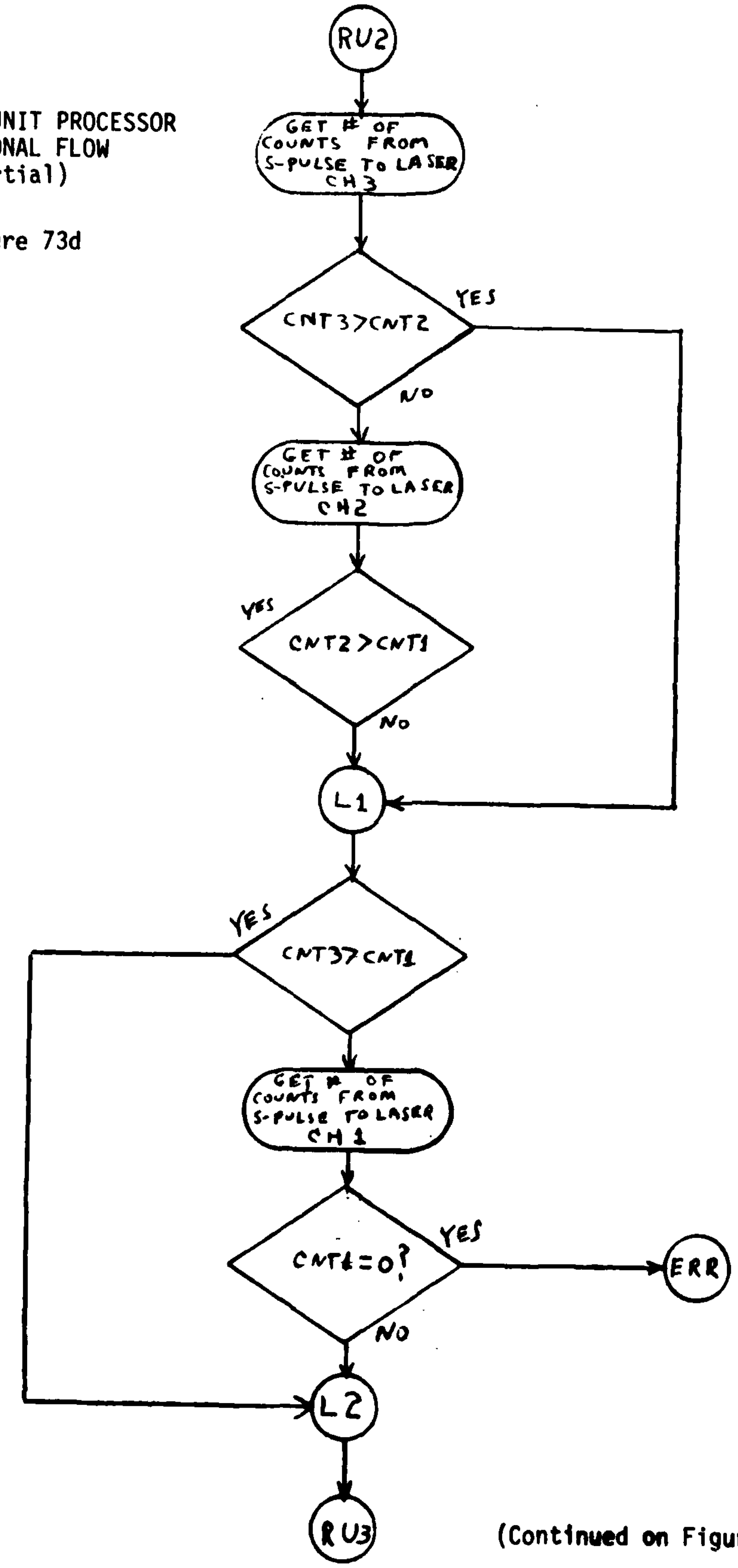
Figure 73c



(Continued on Figure 73d)

REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

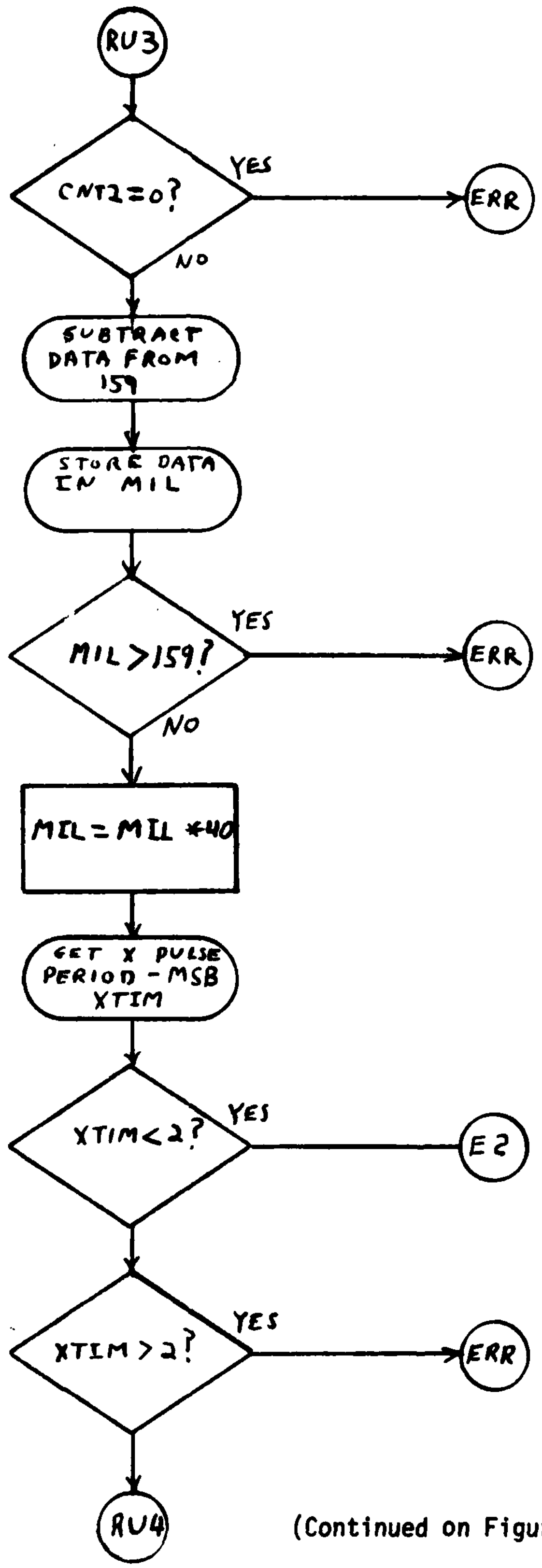
Figure 73d



(Continued on Figure 73e)

REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

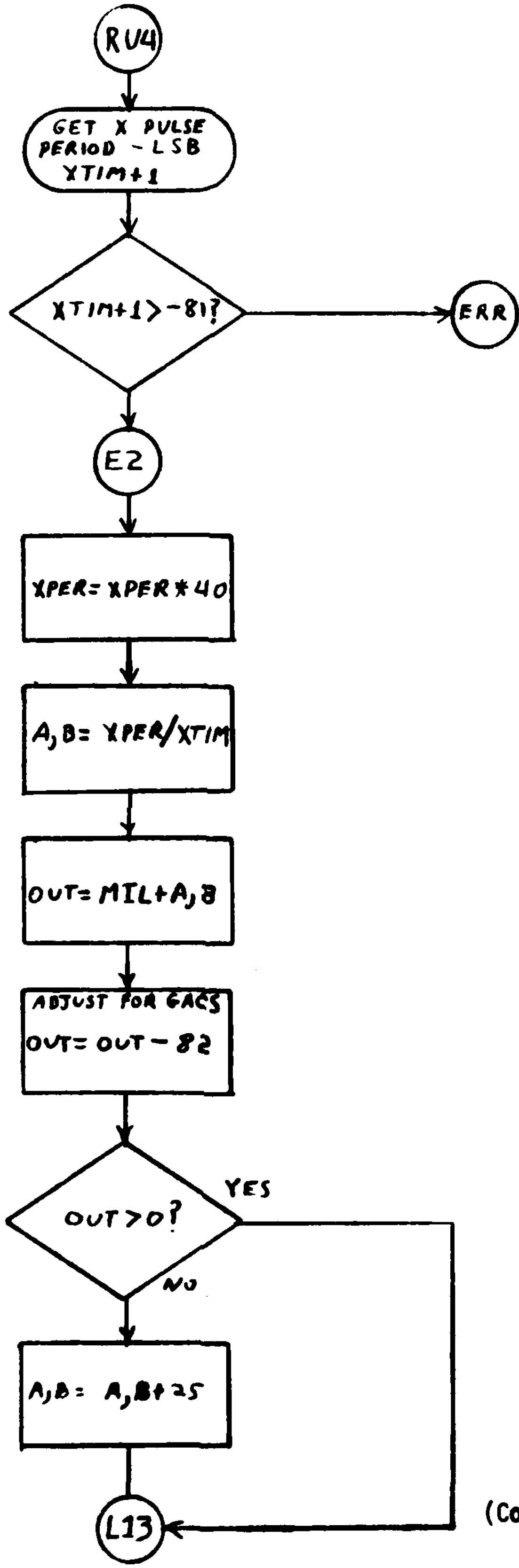
Figure 73e



(Continued on Figure 73f)

REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

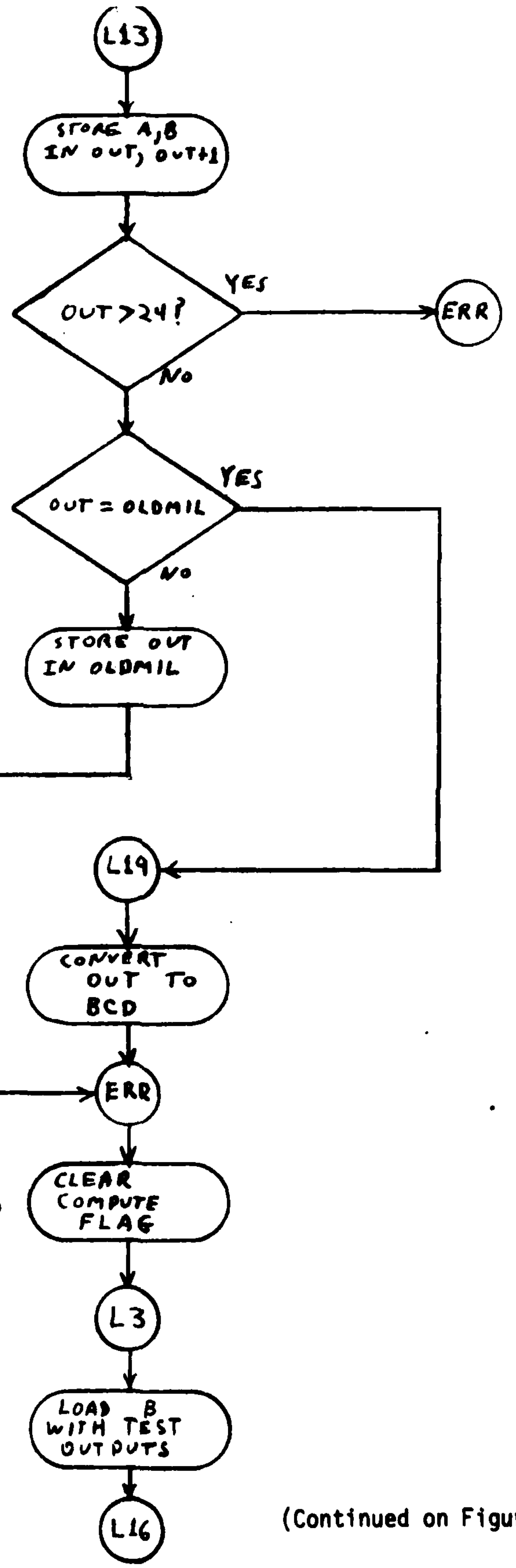
Figure 73f



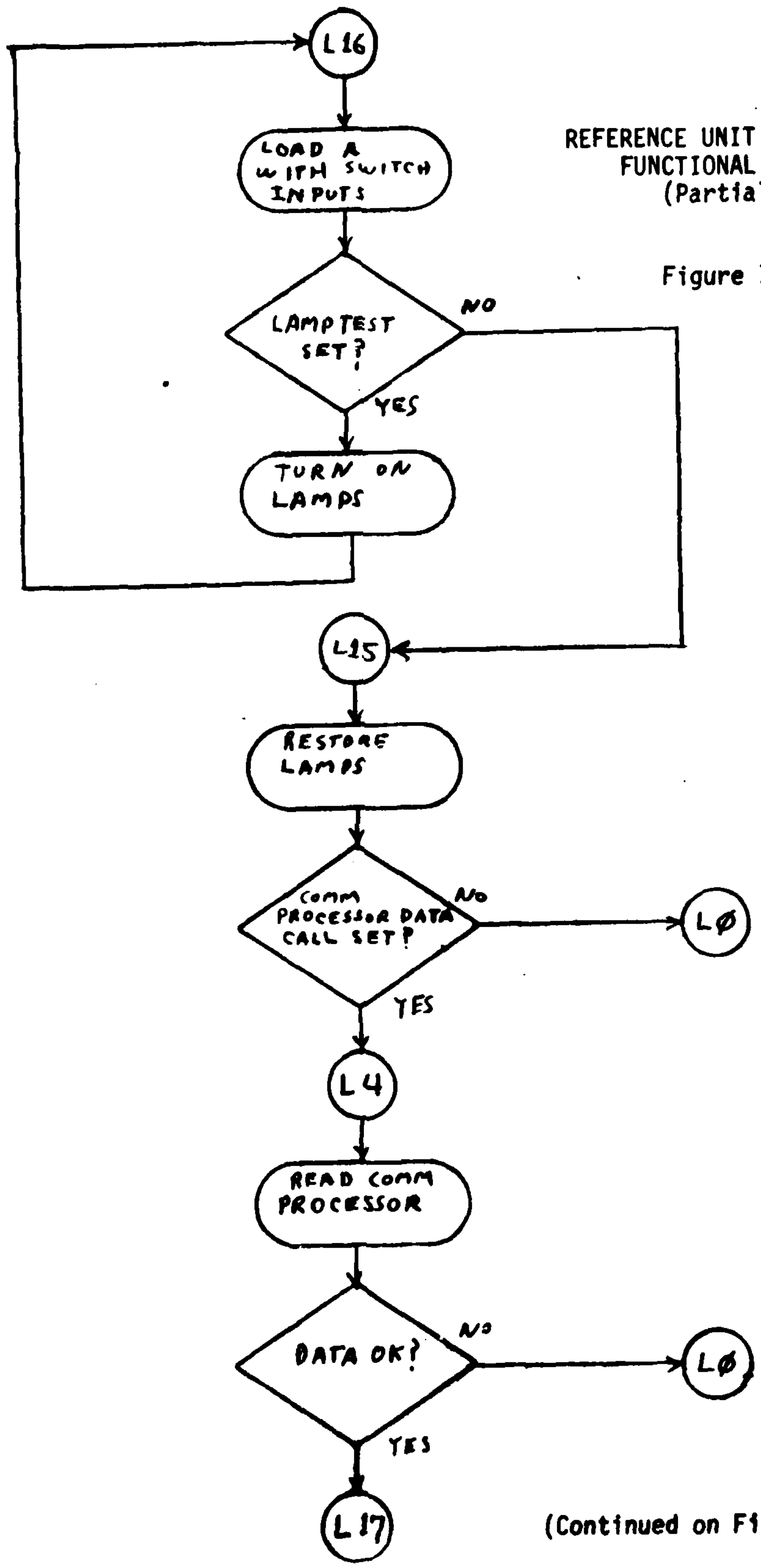
(Continued on Figure 73g)

REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73g



(Continued on Figure 73h)



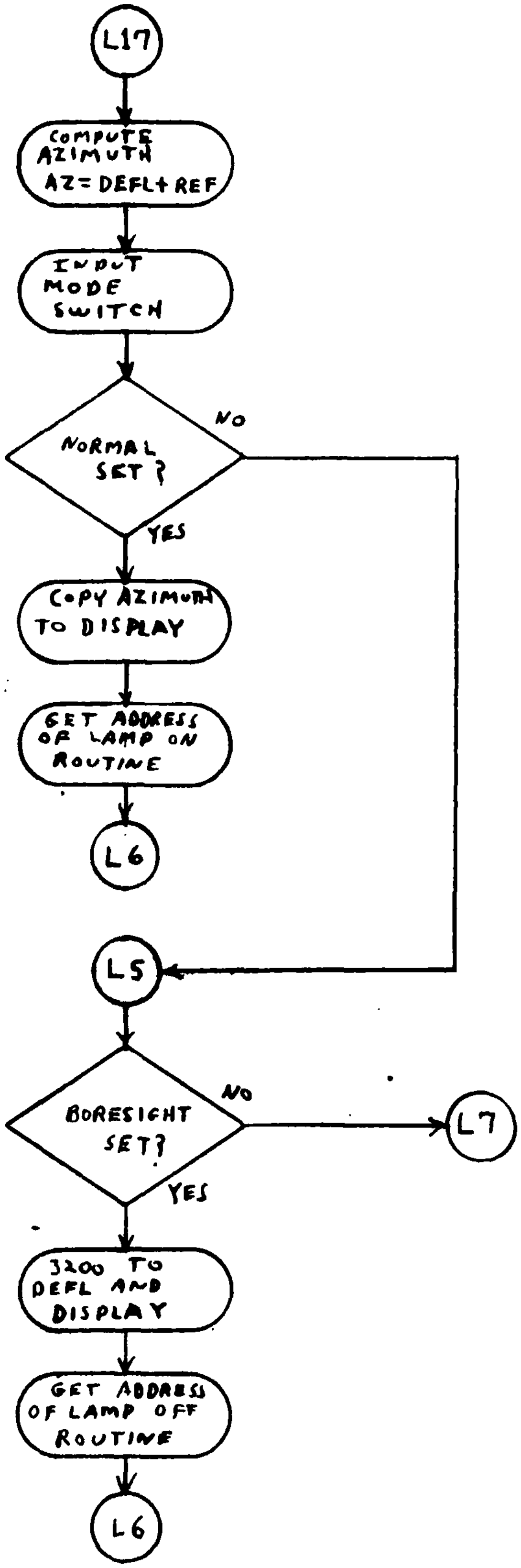
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73h

(Continued on Figure 73i)

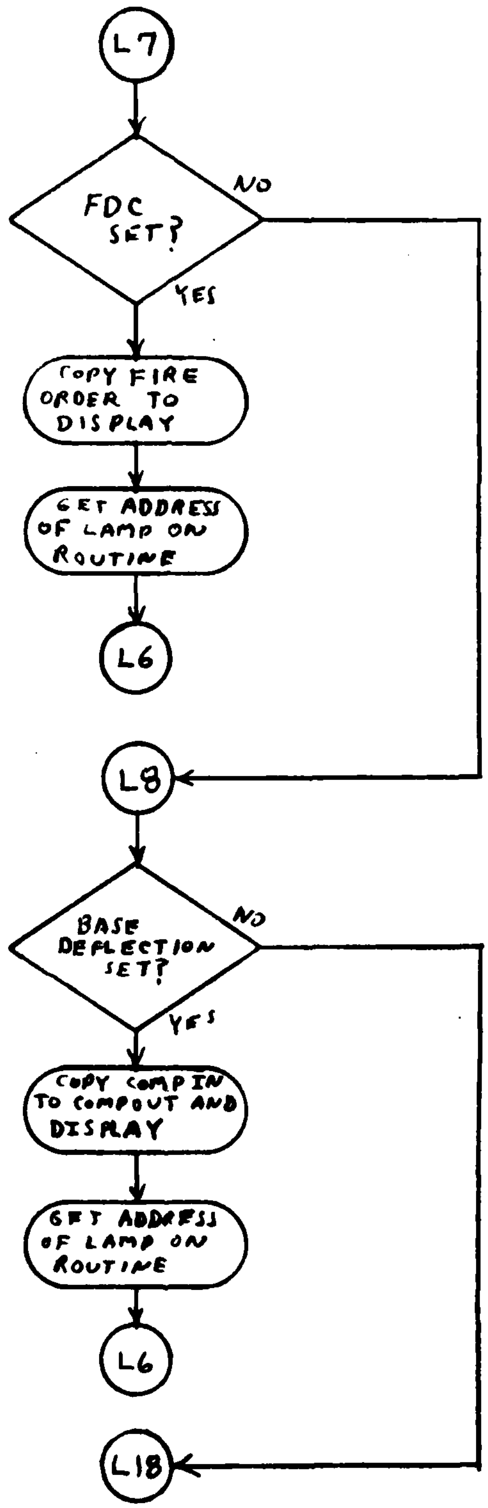
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73i



REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

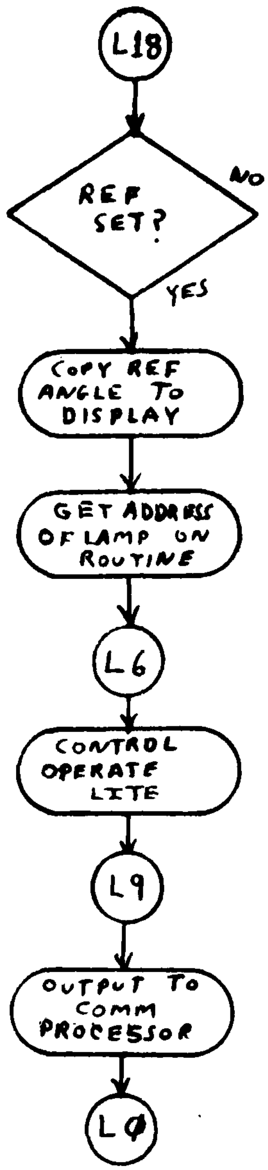
Figure 73j



(Continued on Figure 73k)

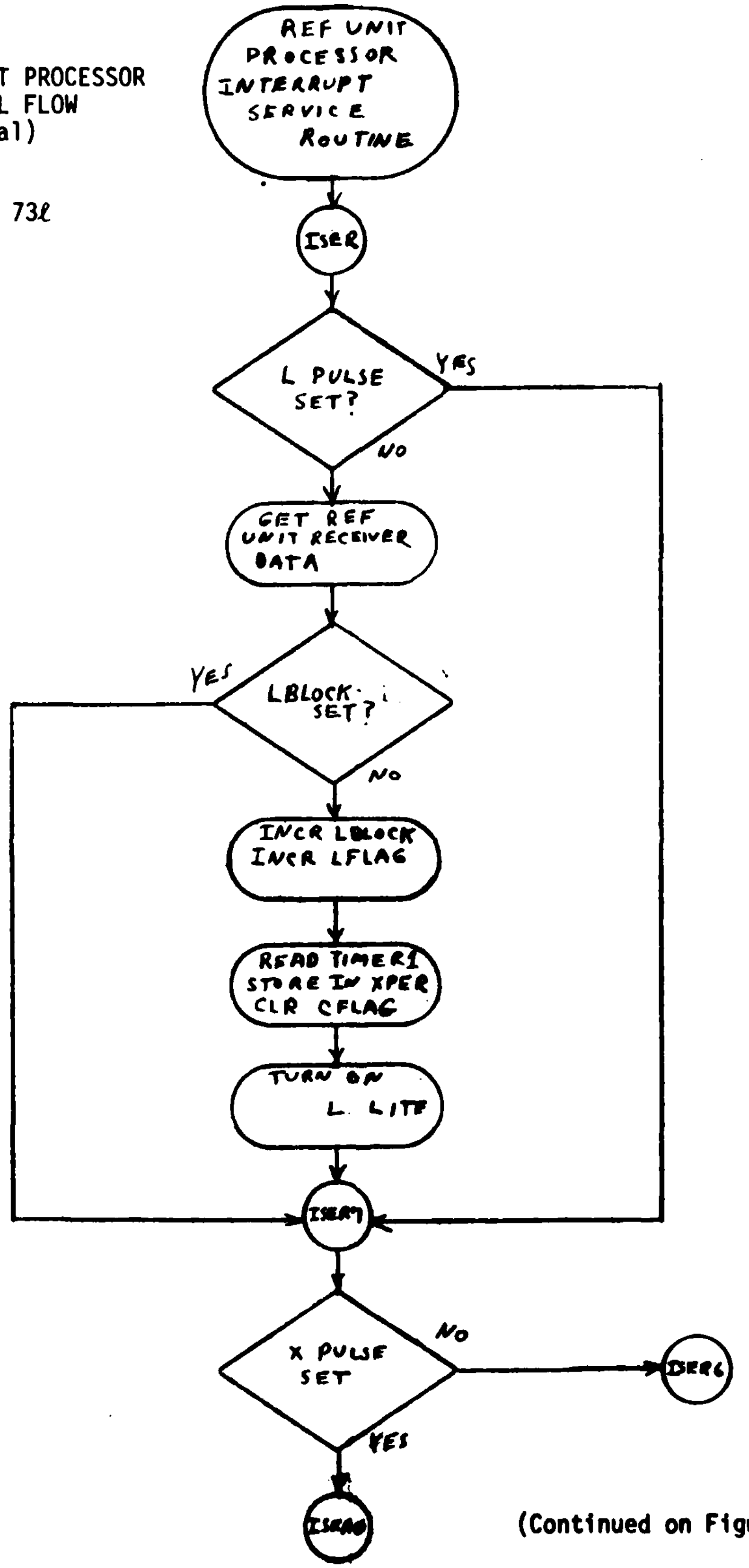
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73k



REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

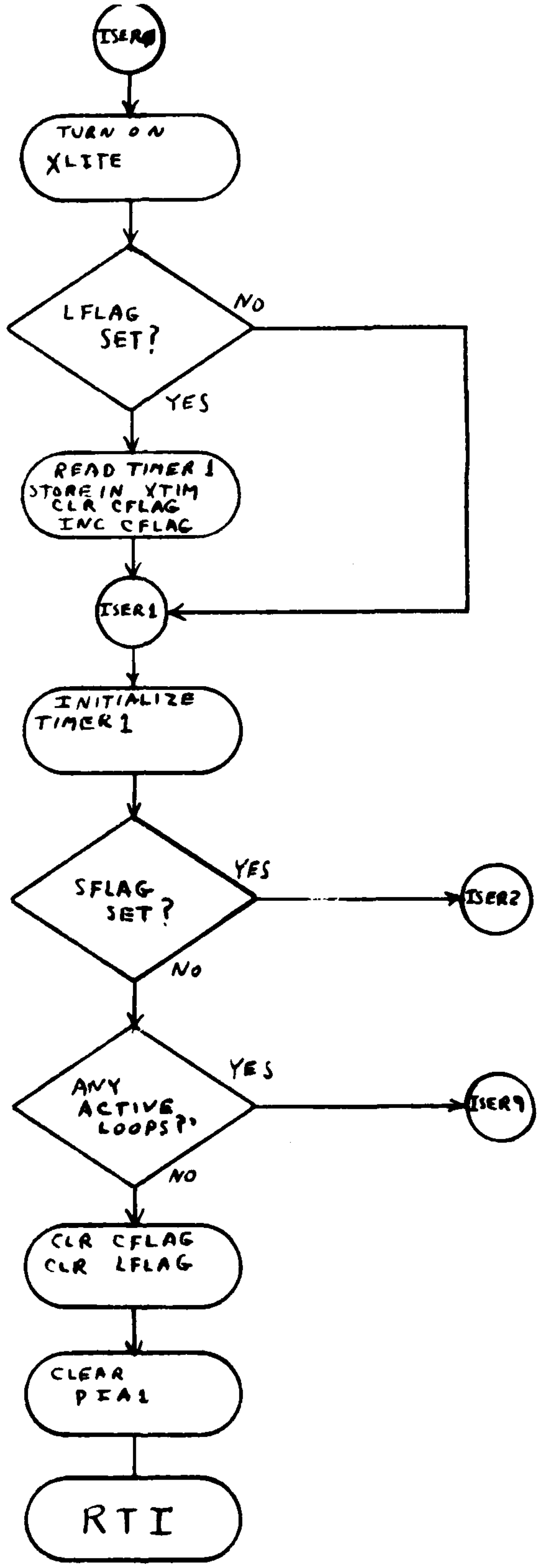
Figure 73ℓ



(Continued on Figure 73m)

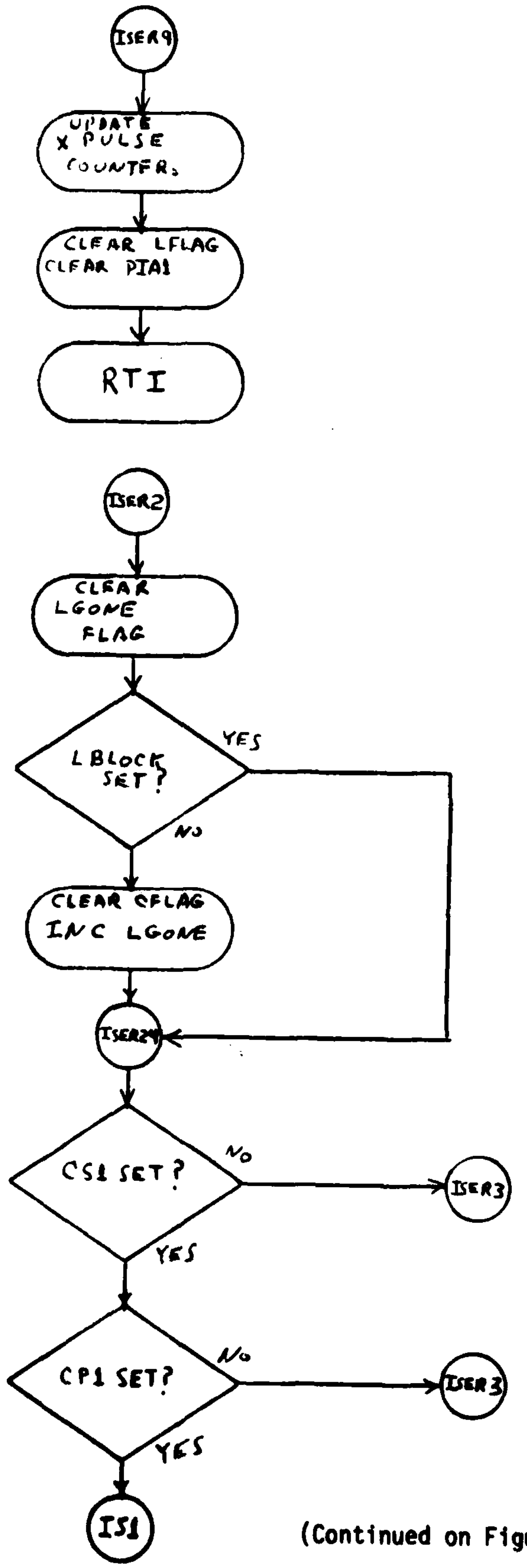
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73m



REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

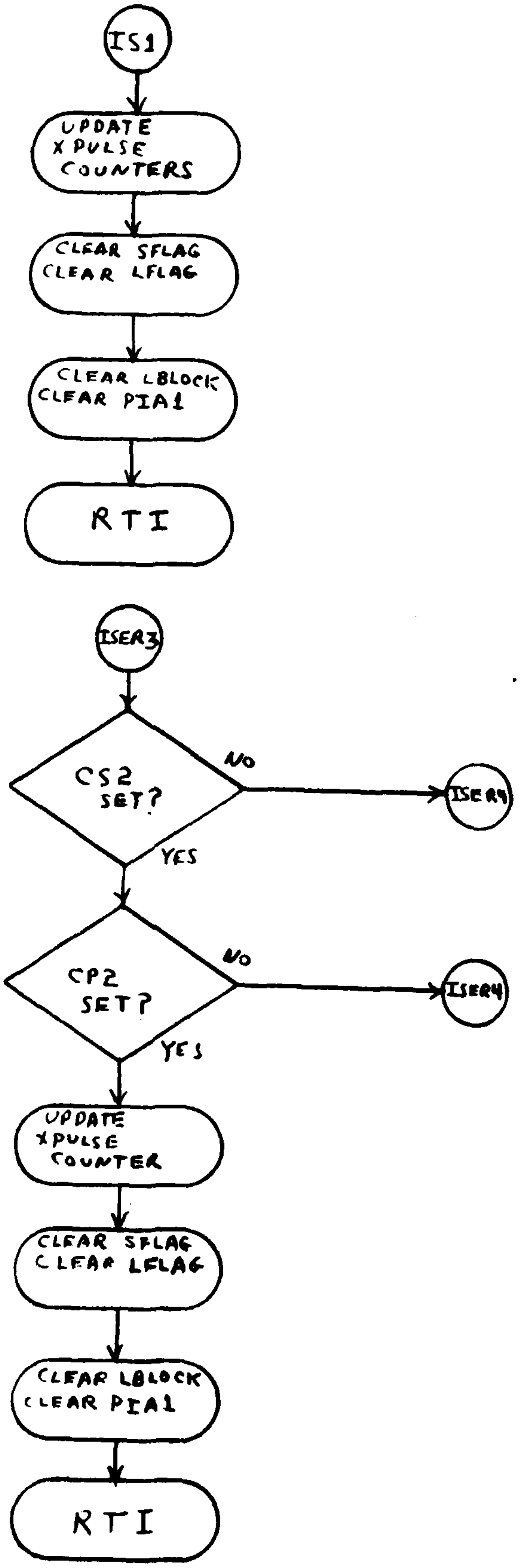
Figure 13n



(Continued on Figure 73o)

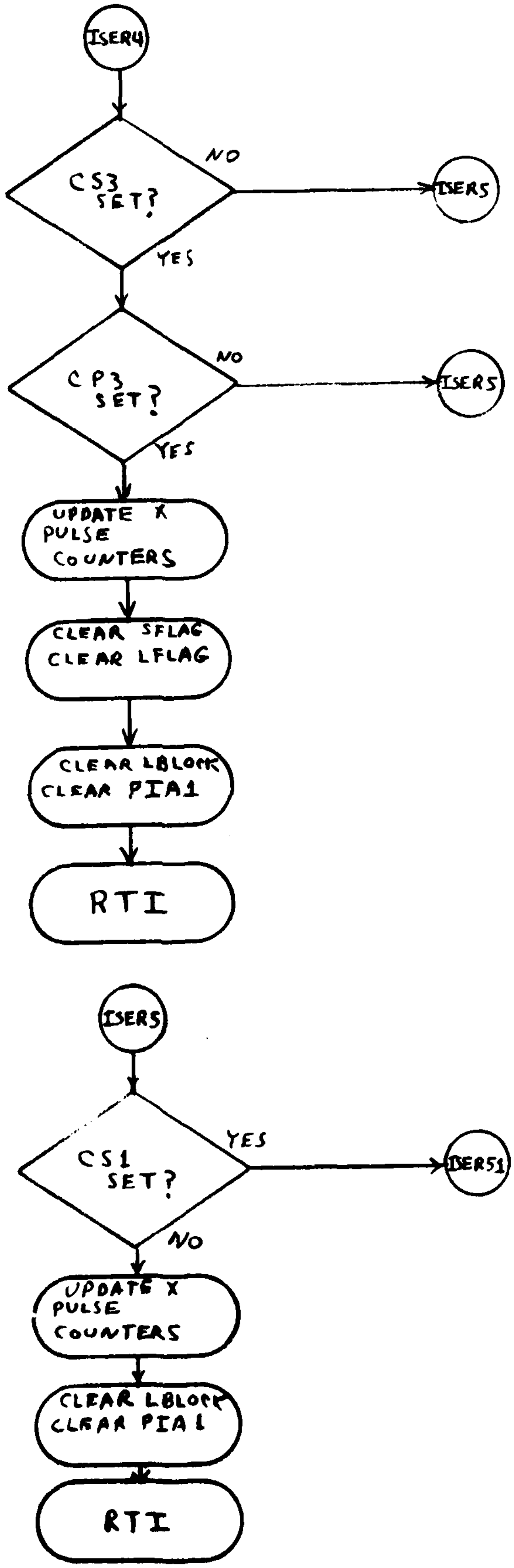
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73o



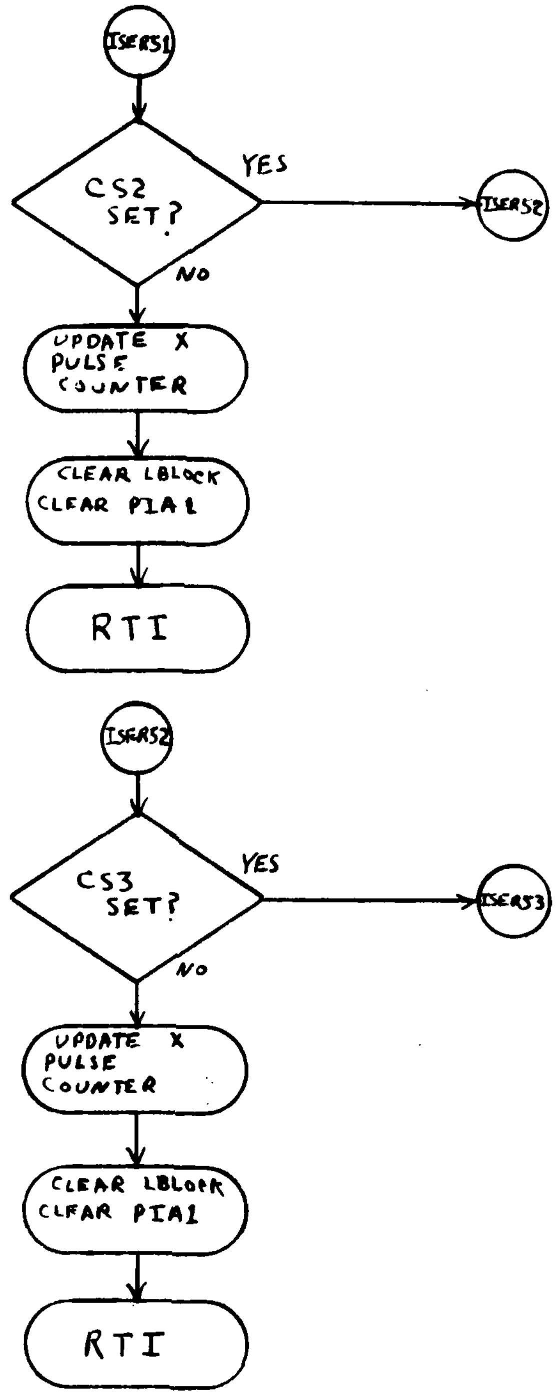
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73p



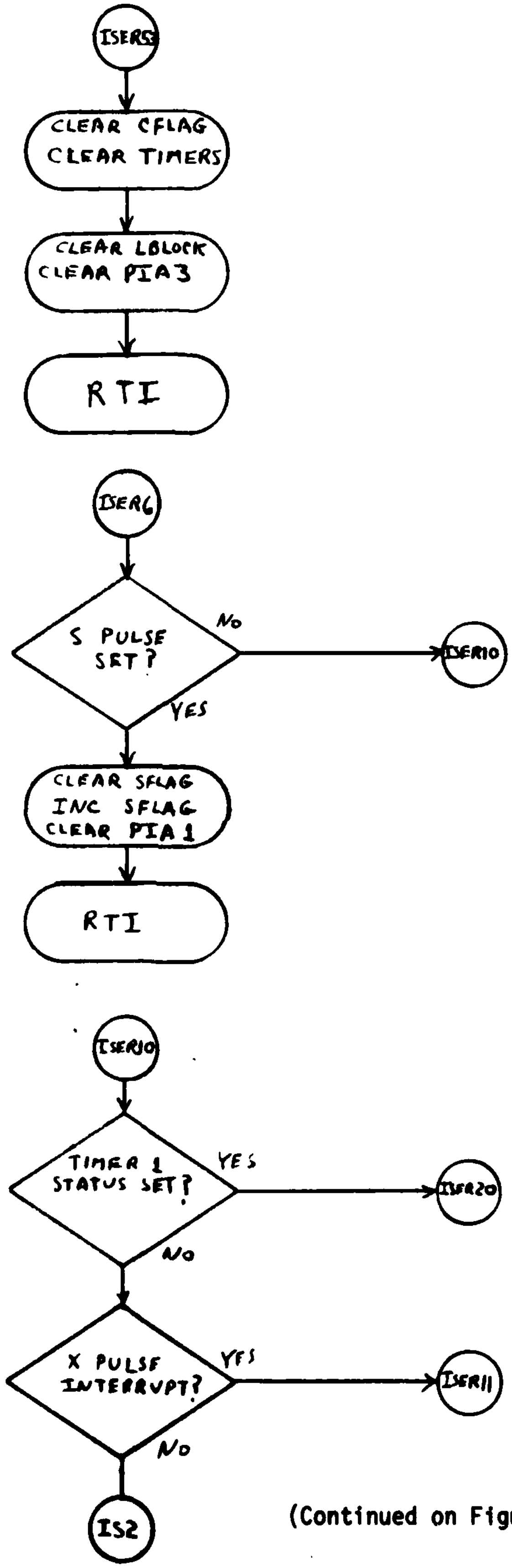
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73q



REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

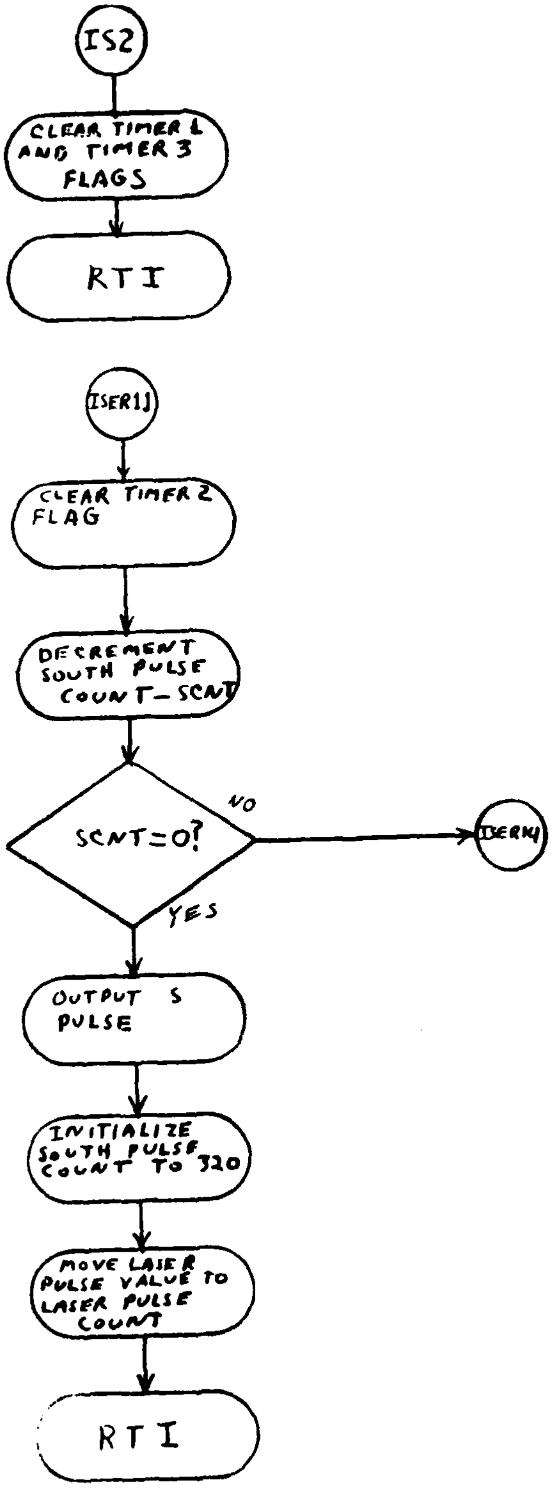
Figure 73r



(Continued on Figure 73s)

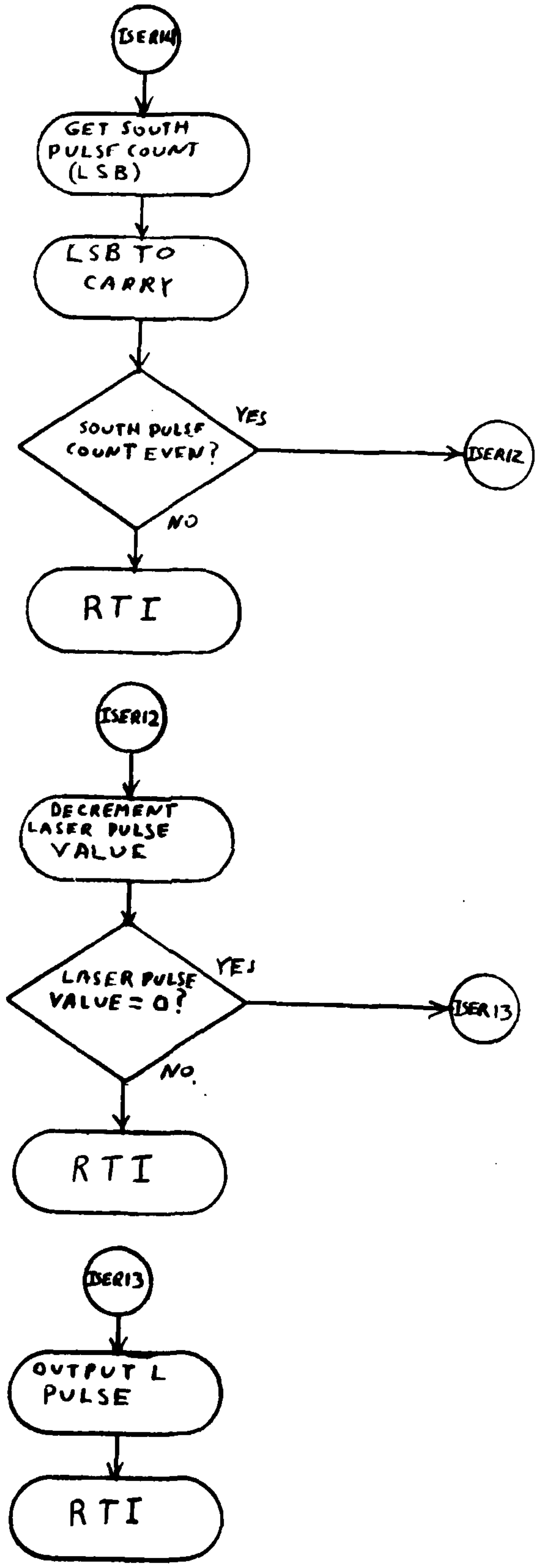
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73s



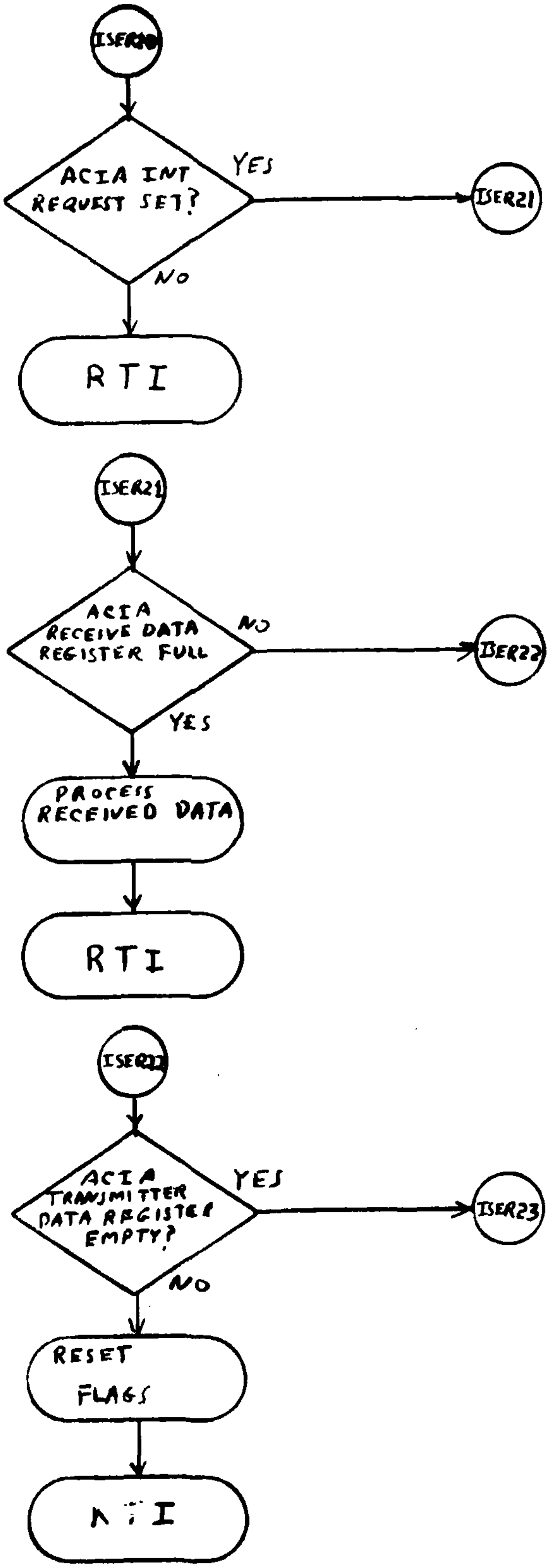
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

Figure 73t



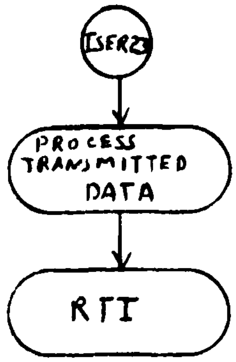
REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

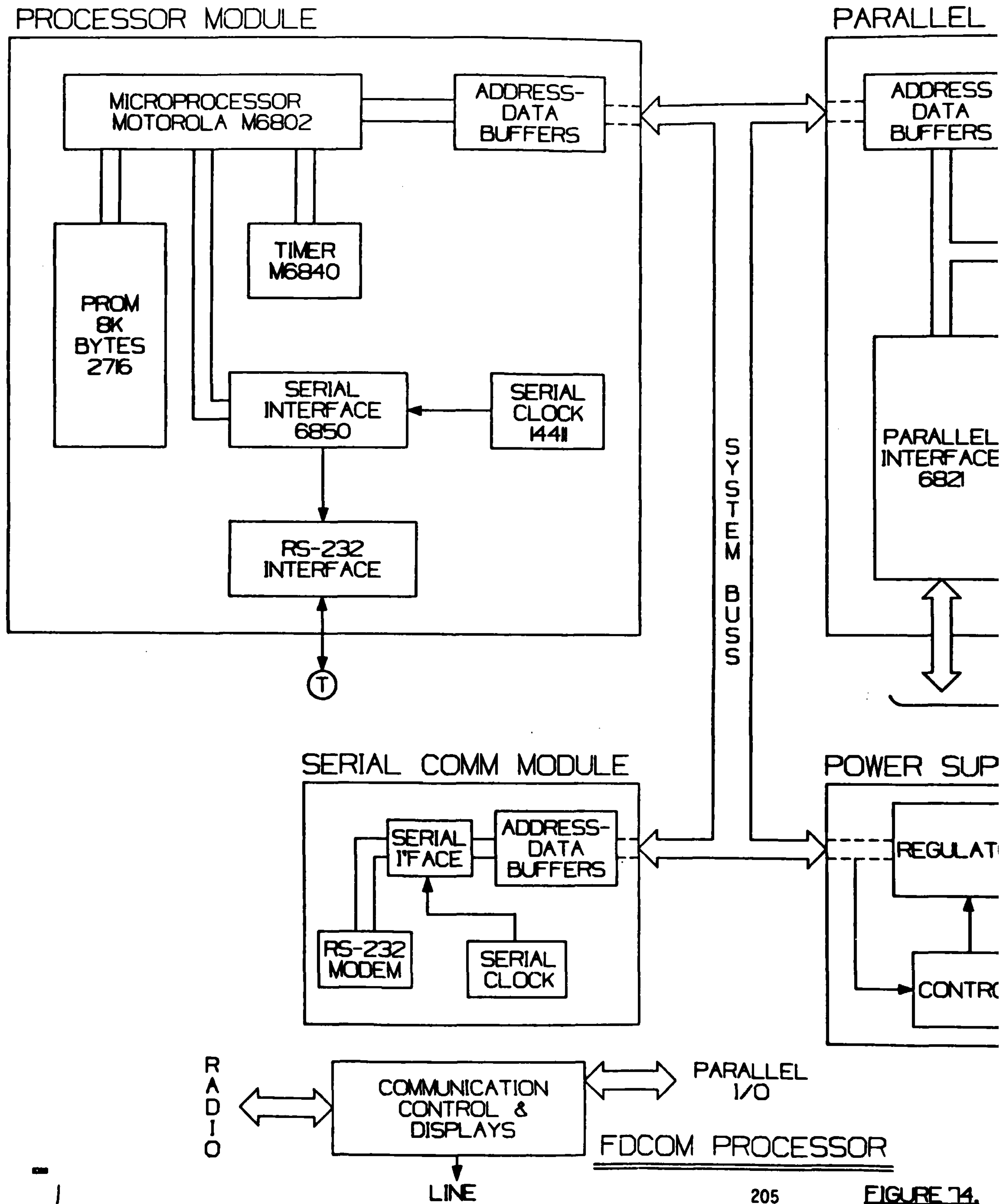
Figure 73u



REFERENCE UNIT PROCESSOR
FUNCTIONAL FLOW
(Partial)

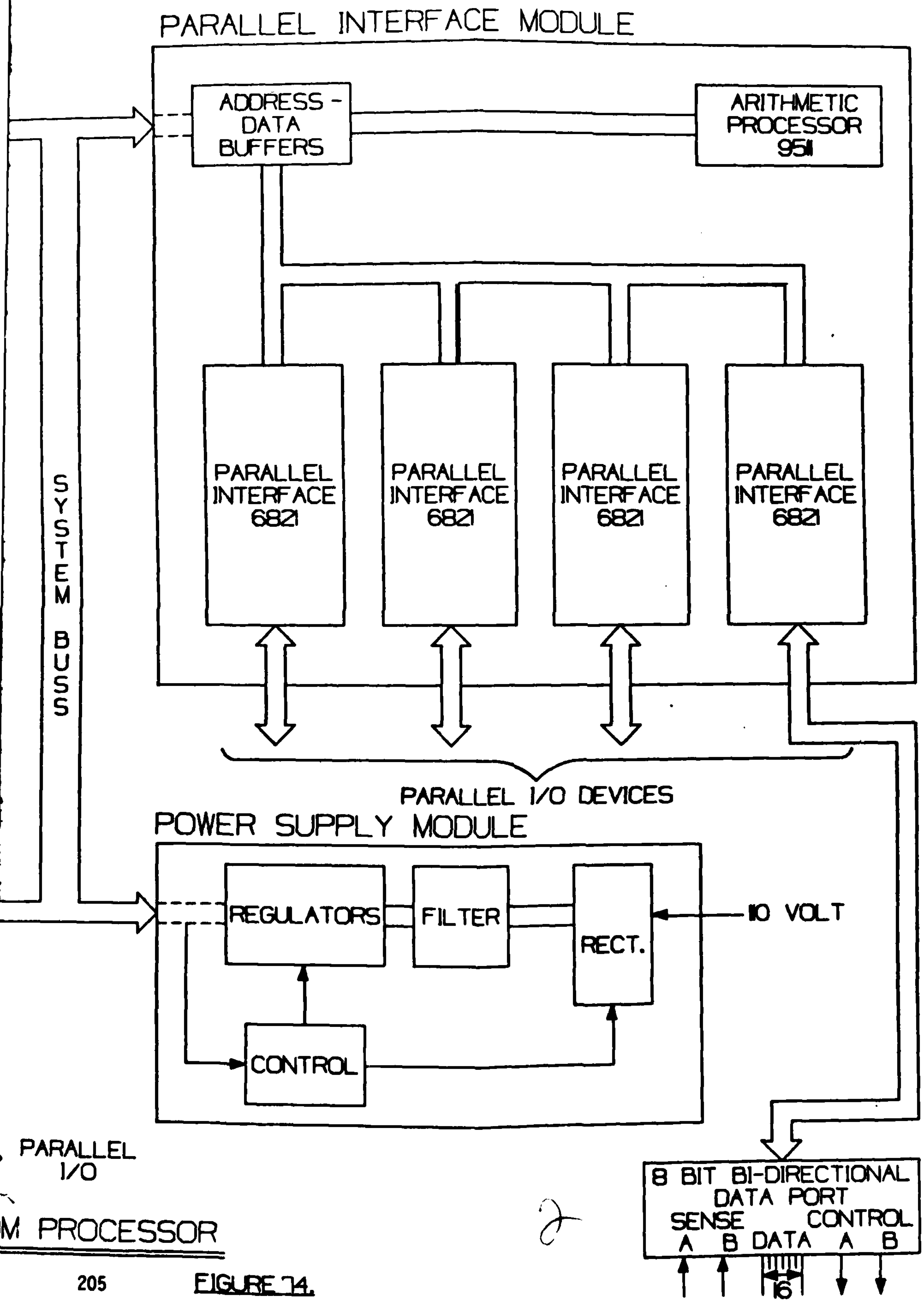
Figure 73v





FDCOM PROCESSOR

FIGURE 14.



M PROCESSOR

FIGURE 14.

Status displays are provided on FDCOM to allow monitoring of the communication sequence; the meaning of each is:

NAK = Negative acknowledgement of message received, retransmit requested.

DCD = Receive data carrier detect

FDIN = FDCOM is receiving message from either FDC computer or terminal (background)

FDOUT = Data received from vehicle is available for output

CJ=3 = FDCOM is unpacking received message

CJ=0 = FDCOM communication link is idle

COMM = Communication system is connected between vehicle and FDC

STBY = Communication system not connected and FDCOM is ready to accept a select

In the connect sequence we have shown how VECOM initiates a data exchange sequence with FDCOM. During this sequence the roles of the respective processors are interchanged; that is, VECOM goes from an active to passive state and FDCOM from a passive to active state. The disconnect sequence again reverses the roles and can be initiated from either end of the link. The disconnect is initiated at VECOM by momentarily switching to the STANDBY mode. This action sets a flag in the processor and when the next message is sent from FDCOM a request for disconnect (RFD) is returned.

The RFD results in a disconnect command (DIS) being sent back to VECOM whose response is to switch to the STANDBY mode.

The disconnect can be initiated at FDCOM by issuing an end of mission command via the FDC computer interface on background package. This message results in a disconnect being sent to VECOM and a switch to STBY mode by FDCOM once the message has been received by VECOM.

The software designed for FDCOM is shown in flow form in Figure 75 and the associated assembly level source code is contained in Appendix E.

B. Automatic Gun Laying System Processor (AGLS)

As indicated in Figure 8 communication between VECOM and AGLS is via a bidirectional RS-232 serial link. Since the original source of gun order data was via a parallel BCD interface to the GACS Gun Unit, changes were required to the AGLS program.

The operating sequence of the AGLS was modified as follows:

- All gun order data is received and operating data reported via a bi-directional serial data port.

- Back up gun order data entry via thumb wheels will be provided using the existing GACS data port.

- AGLS Status word added to the COMM buffer (mode).

- Separate error bound test routines are used for level and tracker status and tracker/quad pitch null tests.

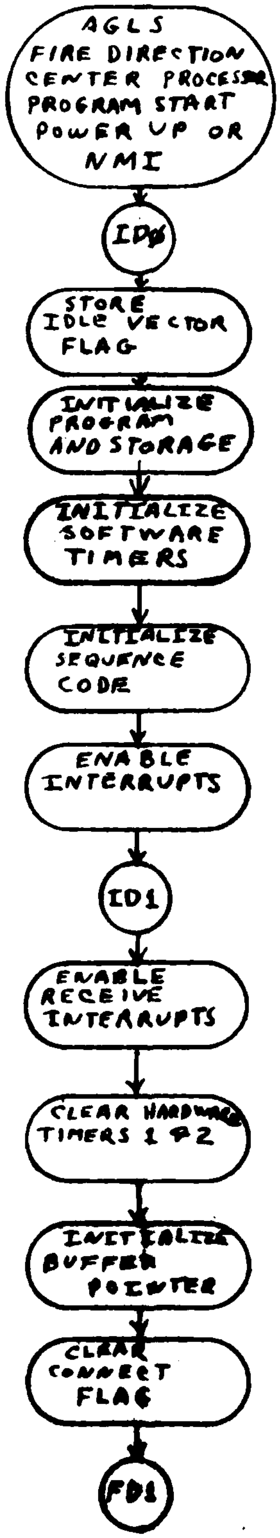
- The system now has the ability to operate on one set of gun orders and display the new (command values).

- New gun orders are latched in with a single switch movement. If in AUTO Az or El mode, the new gun order is latched in via WPN switch; if not AUTO Az or El, the gun order is latched in via SERVO switch.

- Load position is selected at any time in AUTO El mode without any other switch motion. Return to QE is also automatic; i.e., no other siwtches involved.

AGLS COMMUNICATIONS
FIRE DIRECTION CENTER
COMM PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

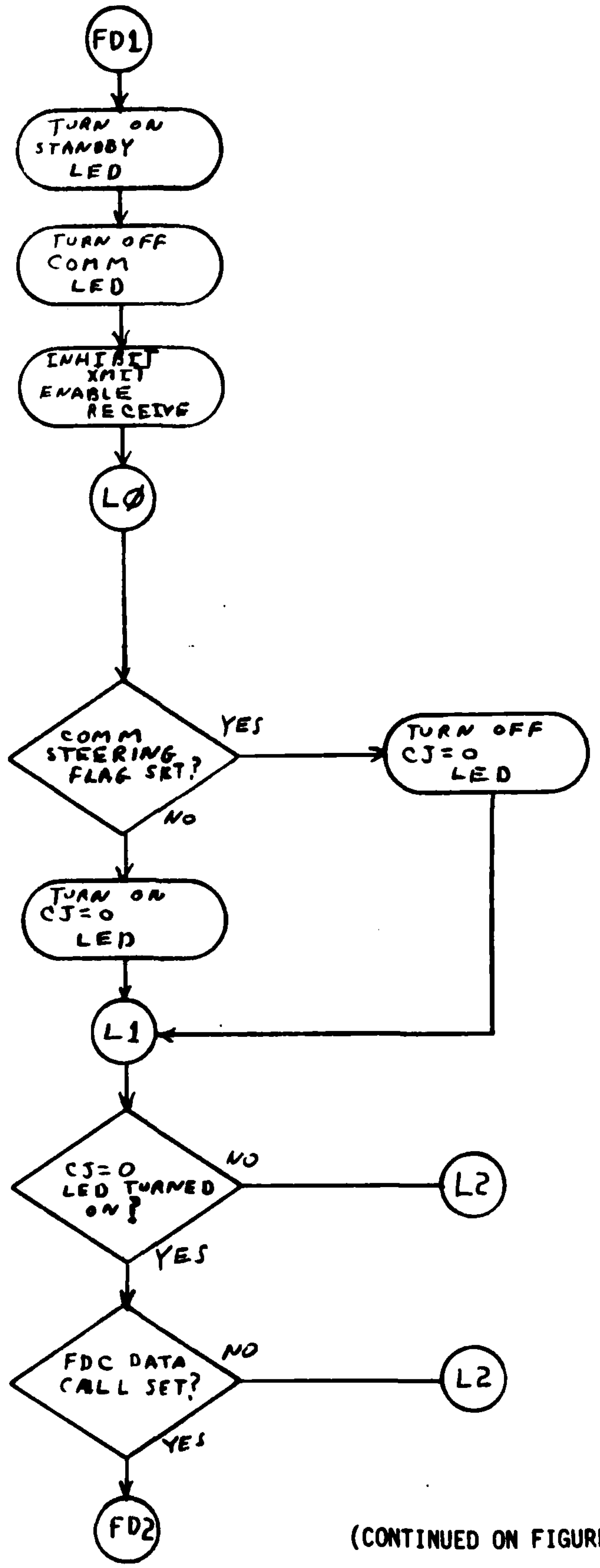
FIGURE 75 a



(CONTINUED ON FIGURE 75b)

AGLS COMMUNICATIONS
FIRE DIRECTION CENTER
COMM PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

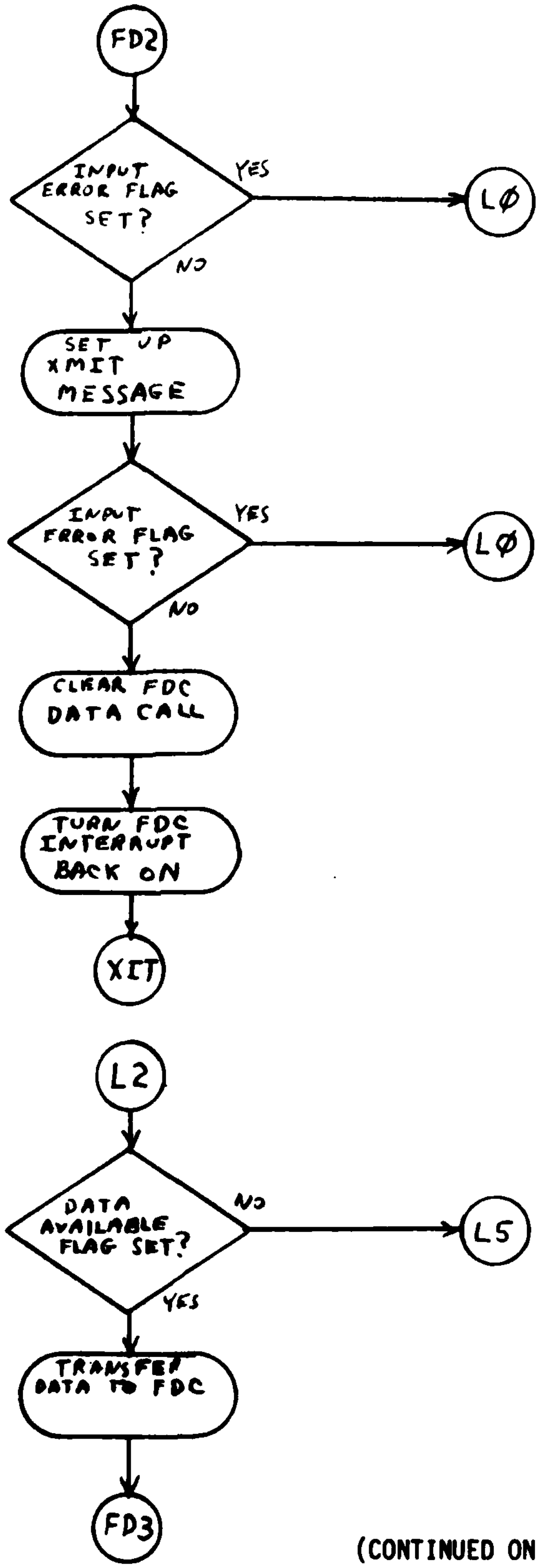
FIGURE 75 b



(CONTINUED ON FIGURE 75c)

AGLS COMMUNICATIONS
FIRE DIRECTION CENTER
COMM PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

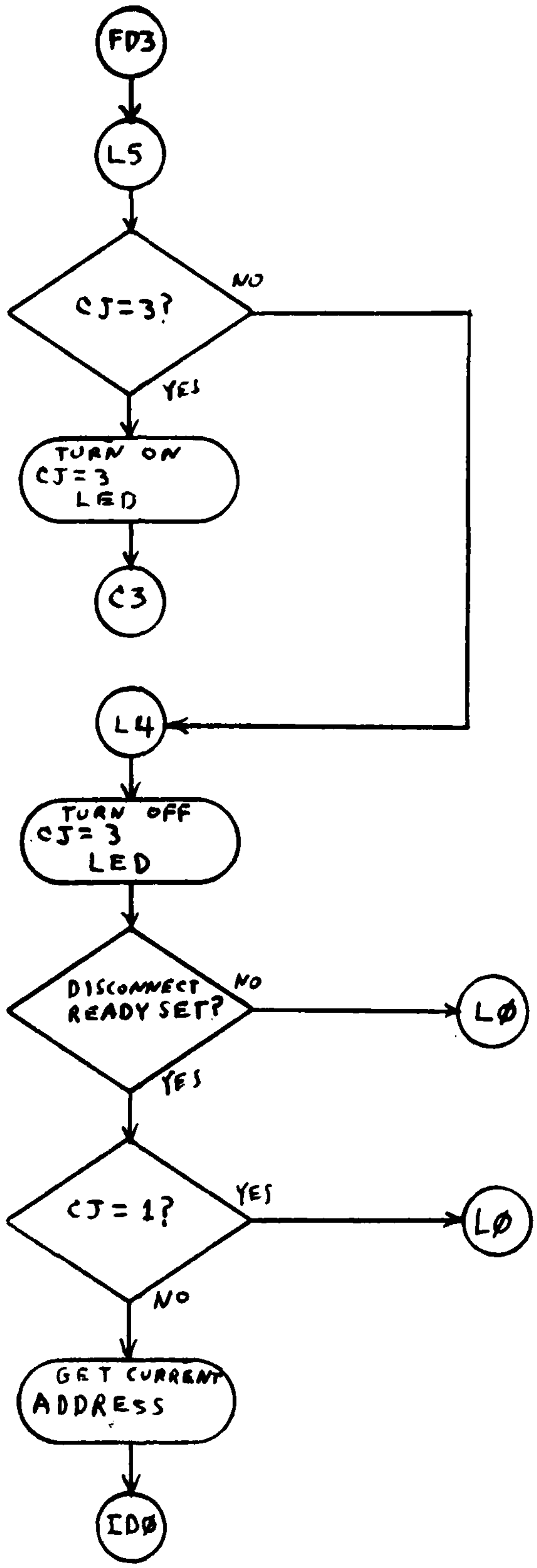
FIGURE 75 c



(CONTINUED ON FIGURE 75d)

AFLS COMMUNICATIONS
FIRE DIRECTION CENTER
COMM PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

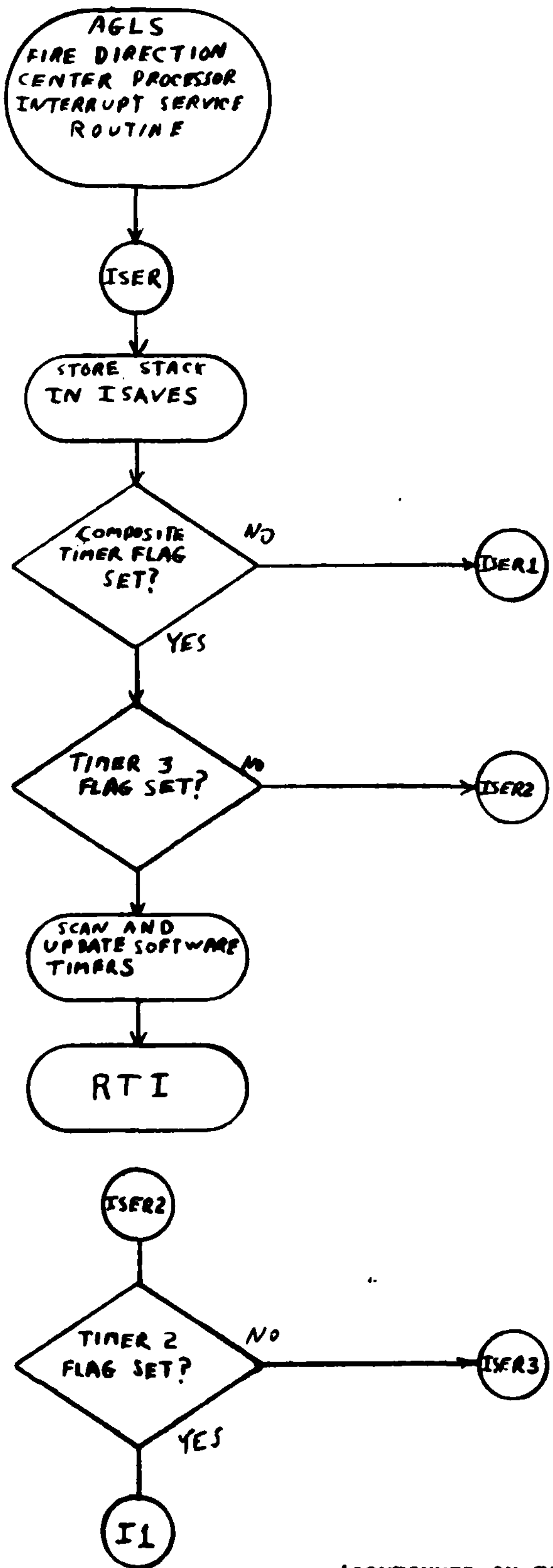
FIGURE 75 d



(CONTINUED ON FIGURE 75e)

AGLS COMMUNICATIONS
FIRE DIRECTION CENTER
COMM PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

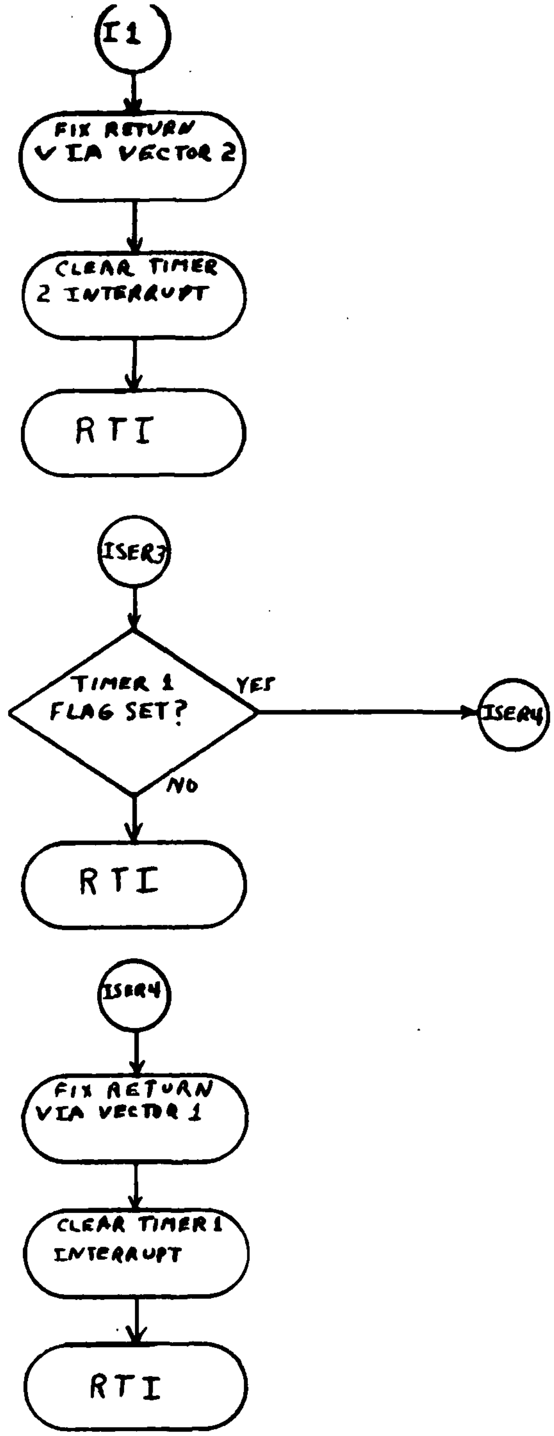
FIGURE 75 e



(CONTINUED ON FIGURE 75f)

AGLS COMMUNICATIONS
FIRE DIRECTION CENTER
COMM PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

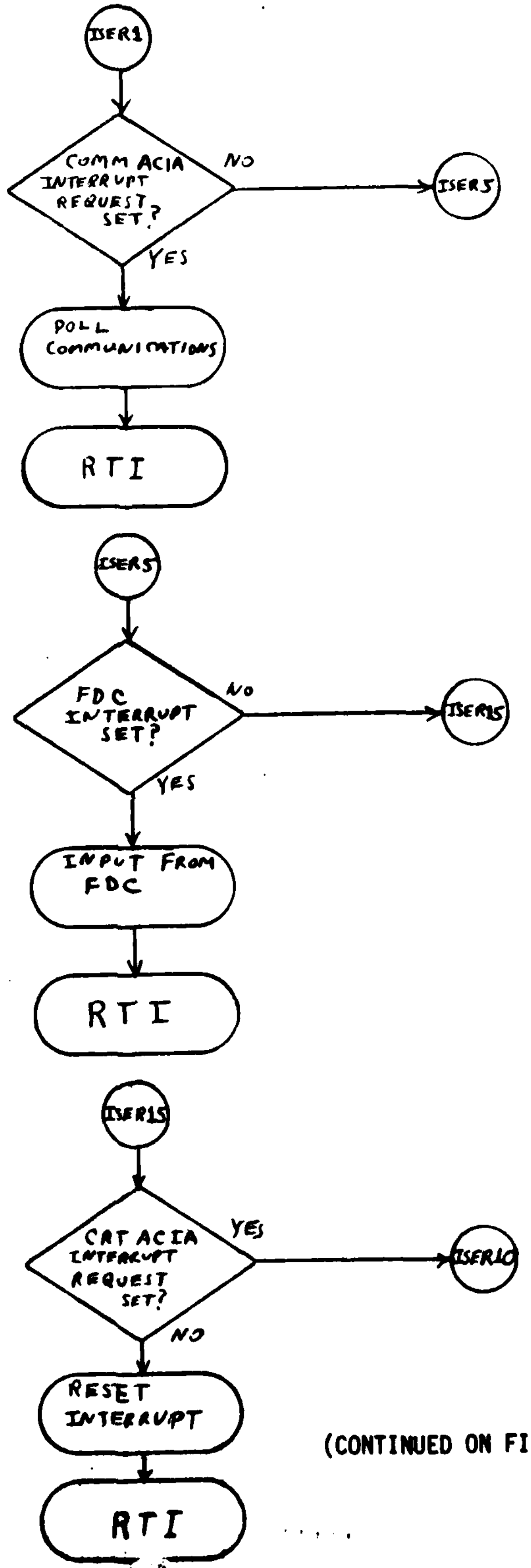
FIGURE 75 f



(CONTINUED ON FIGURE 75g)

AGLS COMMUNICATIONS
FIRE DIRECTION CENTER
COMM PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

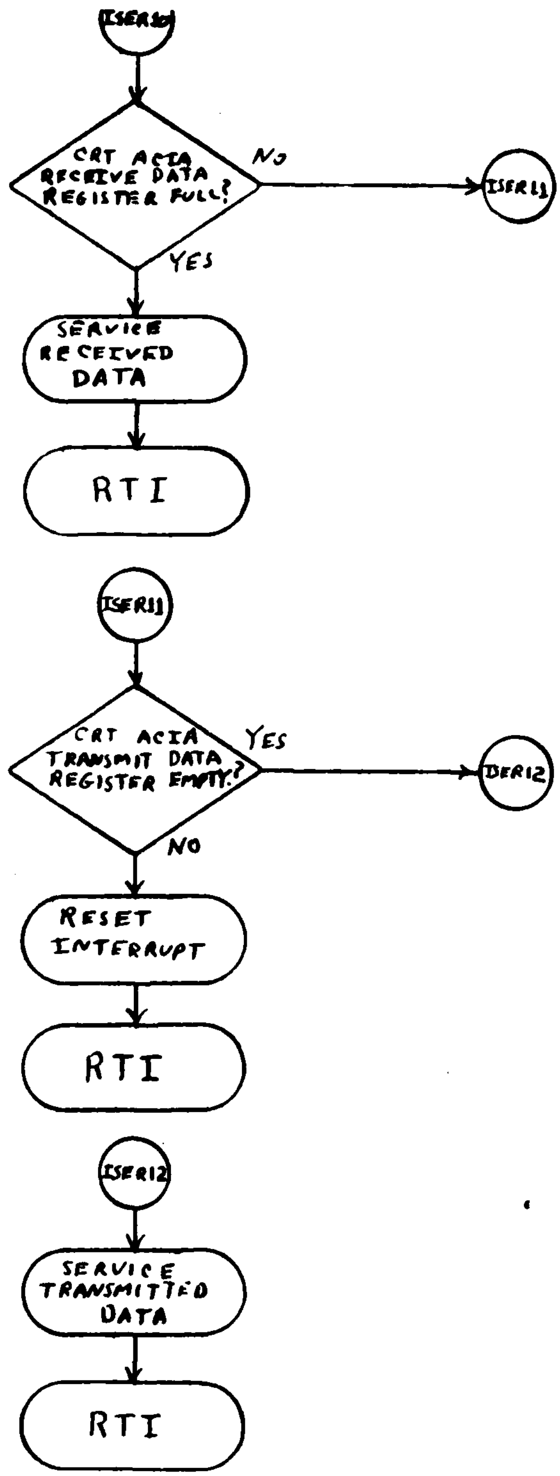
FIGURE 75 g



(CONTINUED ON FIGURE 75h)

AGLS COMMUNICATIONS
FIRE DIRECTION CENTER
COMM PROCESSOR
FUNCTIONAL FLOW
(PARTIAL)

FIGURE 75 h

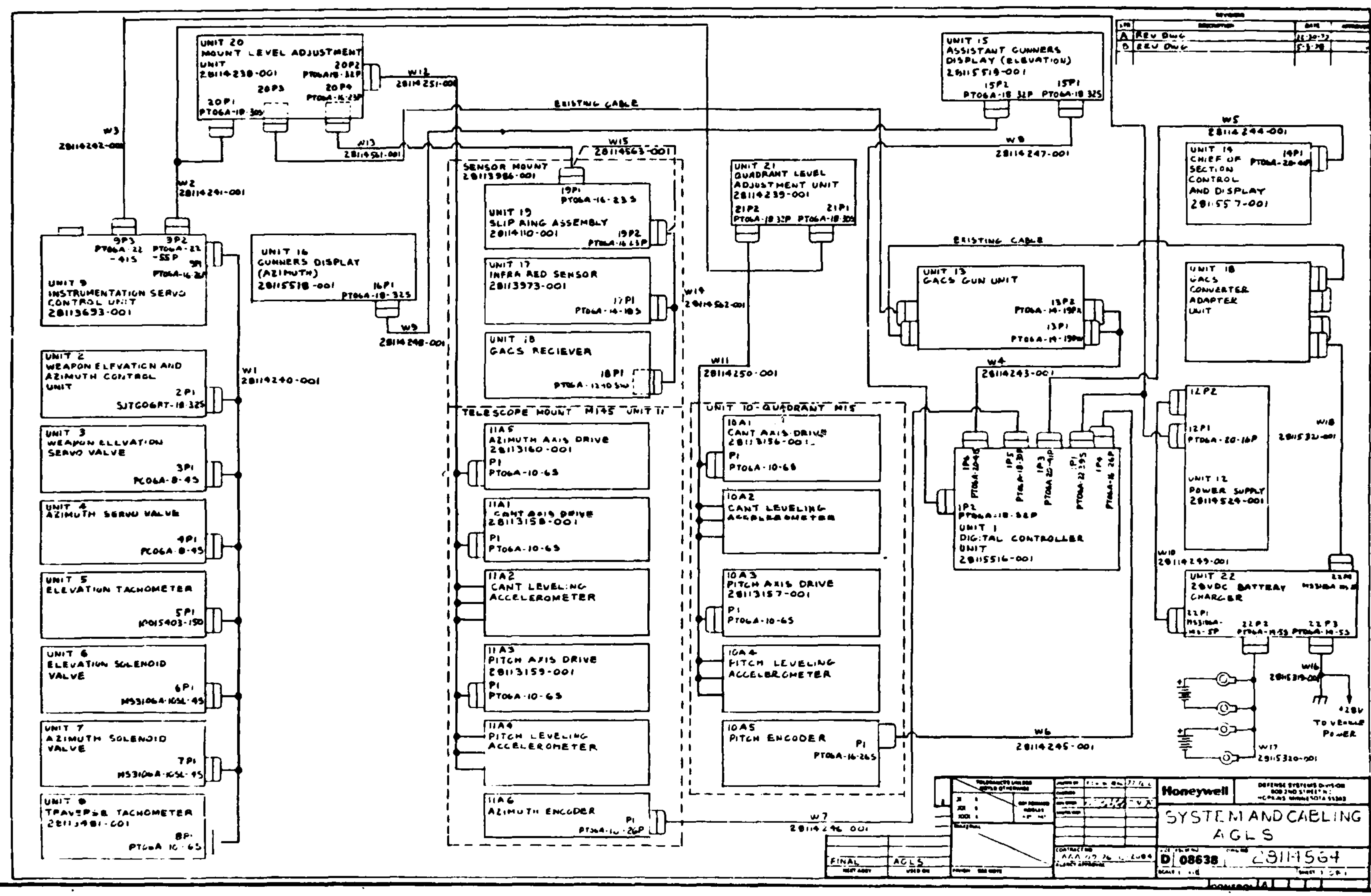


In addition to these changes, the reference angle computation algorithm was changed. Rather than accepting a COMMAND azimuth that is the sum of the FDC gun order deflection and the reference angle, COMMANDED azimuth is accepted and displayed directly. The reference angle data from VECOM is subtracted from the absolute encoded pantel reading to make the ACTUAL azimuth display compatible with the COMMANDED value.

The new assembly level source code listings reflecting these changes are contained in Appendix F.

APPENDIX A

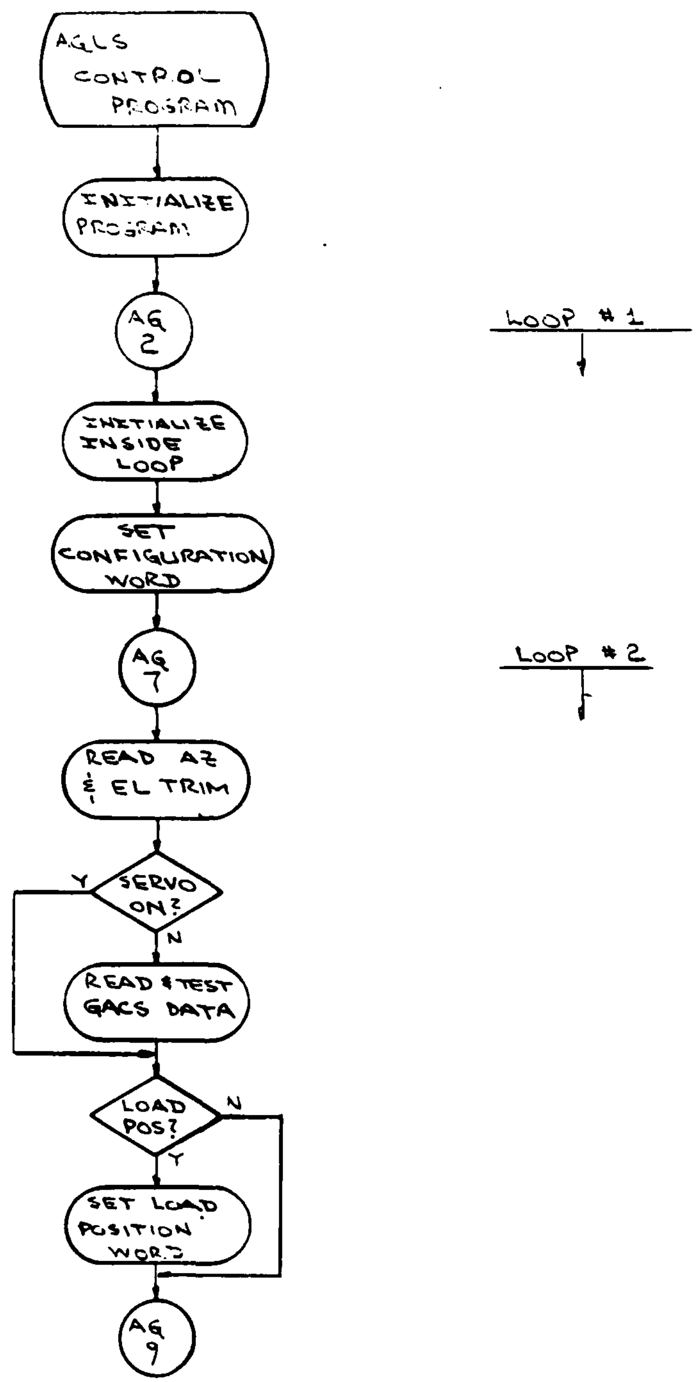
SYSTEM AND CABLING-AGLS

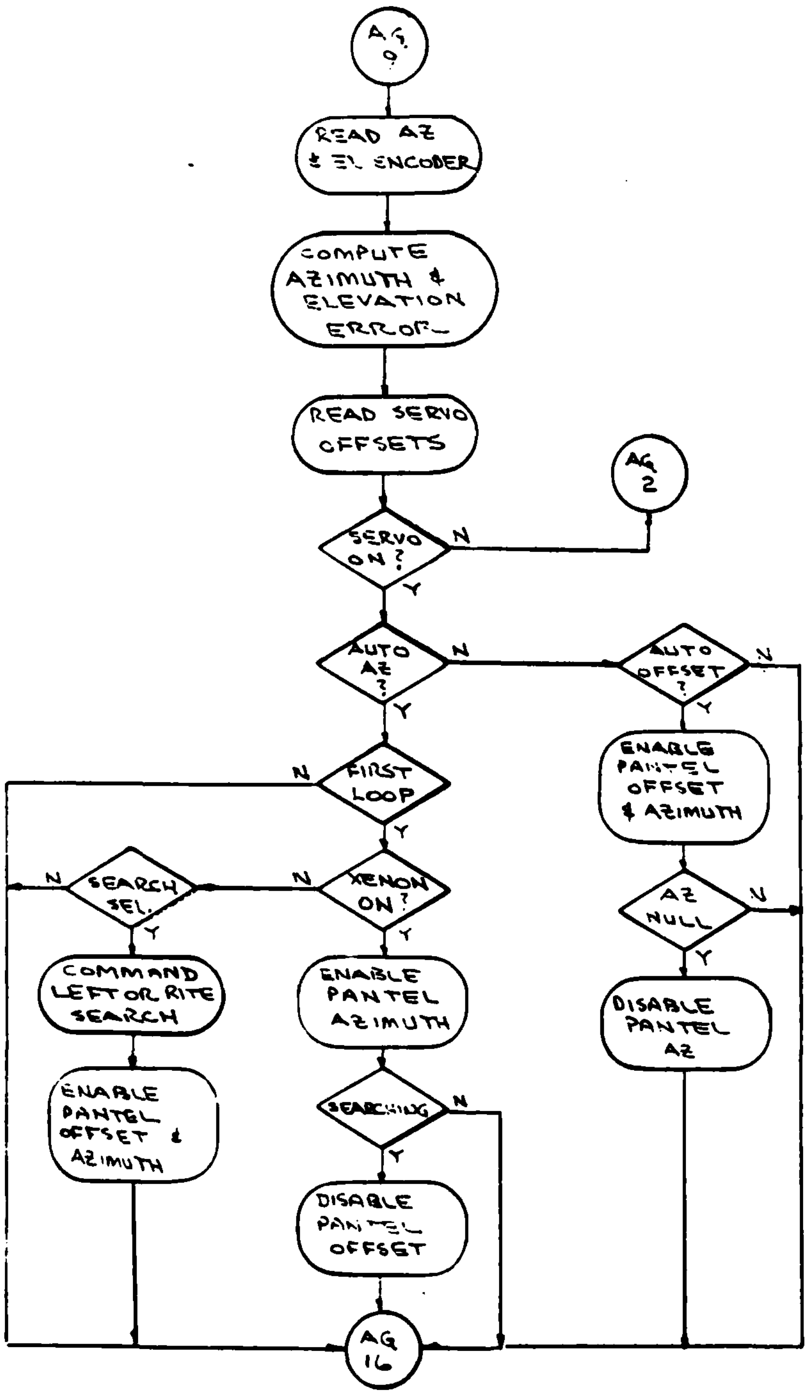

UNIT 20 MOUNT LEVEL ADJUSTMENT UNIT 28114238-001
20P1 PT06A-18-30S
20P2 PT06A-18-32P
20P3
20P4 PT06A-16-23P
W12 28114251-001
W13 28114561-001
EXISTING CABLE
W3 28114242-001
W2 28114241-001
UNIT 9 INSTRUMENTATION SERVO CONTROL UNIT 28113653-001
9P3 PT06A-22-41S
3P2 PT06A-22-55P
UNIT 16 GUNNERS DISPLAY (AZIMUTH) 28115518-001
16P1 PT06A-18-32S
W9 28114248-001
W1 28114240-001
UNIT 2 WEAPON ELEVATION AND AZIMUTH CONTROL UNIT
2P1 SJTG06RT-18-32S
UNIT 3 WEAPON ELEVATION SERVO VALVE
3P1 PC06A-8-4S
UNIT 4 AZIMUTH SERVO VALVE
4P1 PC06A-8-4S
UNIT 5 ELEVATION TACHOMETER
5P1 10015403-150
UNIT 6 ELEVATION SOLENOID VALVE
6P1 MS3106A-10SL-4S
UNIT 7 AZIMUTH SOLENOID VALVE
7P1 MS3106A-10SL-4S
UNIT 8 TRAVERSE TACHOMETER 28113481-001
8P1 PT06A-10-6S
W15 28114563-001
SENSOR MOUNT 28113986-001
UNIT 19 SLIP RING ASSEMBLY 28114110-001
19P1 PT06A-16-23S
19P2 PT06A-16-23P
UNIT 17 INFRA RED SENSOR 28113973-001
17P1 PT06A-14-18S
UNIT 18 GACS RECIEVER
18P1 PT06A-12-10SW
W14 28114562-001
TELESCOPE MOUNT M145 UNIT 11
11A5 AZIMUTH AXIS DRIVE 28113160-001
P1 PT06A-10-6S
11A1 CANT AXIS DRIVE 28113158-001
P1 PT06A-10-6S
11A2 CANT LEVELING ACCELEROMETER
11A3 PITCH AXIS DRIVE 28113159-001
P1 PT06A-10-6S
11A4 PITCH LEVELING ACCELEROMETER
11A6 AZIMUTH ENCODER
P1 PT06A-16-26P
UNIT 21 QUADRANT LEVEL ADJUSTMENT UNIT 28114239-001
21P2 PT06A-18-32P
21P1 PT06A-18-30S
W11 28114250-001
UNIT 10 QUADRANT M15
10A1 CANT AXIS DRIVE 28113156-001
P1 PT06A-10-6S
10A2 CANT LEVELING ACCELEROMETER
10A3 PITCH AXIS DRIVE 28113157-001
P1 PT06A-10-6S
10A4 PITCH LEVELING ACCELEROMETER
10A5 PITCH ENCODER
P1 PT06A-16-26S
W7 28114246-001
UNIT 15 ASSISTANT GUNNERS DISPLAY (ELEVATION) 28115519-001
15P2 PT06A-18-32P
15P1 PT06A-18-32S
W8 28114247-001
EXISTING CABLE
UNIT 13 GACS GUN UNIT
13P2 PT06A-14-19PX
13P1 PT06A-14-19PW
W4 28114243-001
UNIT 1 DIGITAL CONTROLLER UNIT 28115516-001
1P1 PT06A-22-55S
1P2 PT06A-18-32P
1P3 PT06A-22-41P
1P4 PT06A-16-26P
1P5 PT06A-18-32P
1P6 PT06A-20-41S
W6 28114245-001
W5 28114244-001
UNIT 14 CHIEF OF SECTION CONTROL AND DISPLAY 28115557-001
14P1 PT06A-20-41P
UNIT 18 GACS CONVERTER ADAPTER UNIT
UNIT 12 POWER SUPPLY 28114529-001
12P1 PT06A-20-16P
12P2
W18 28115321-001
W10 28114249-001
UNIT 22 28VDC BATTERY CHARGER
22P1 MS3106A-14S-5P
22P2 PT06A-14-5S
22P3 PT06A-14-5S
22P4
W16 28115319-001
+28V
TO VEHICLE POWER
W17 28115320-001
REVISIONS
A REV DWG
B REV DWG
FINAL
AGLS
Honeywell
DEFENSE SYSTEMS DIVISION
SYSTEM AND CABLING AGLS
D 08638 28114564

APPENDIX B

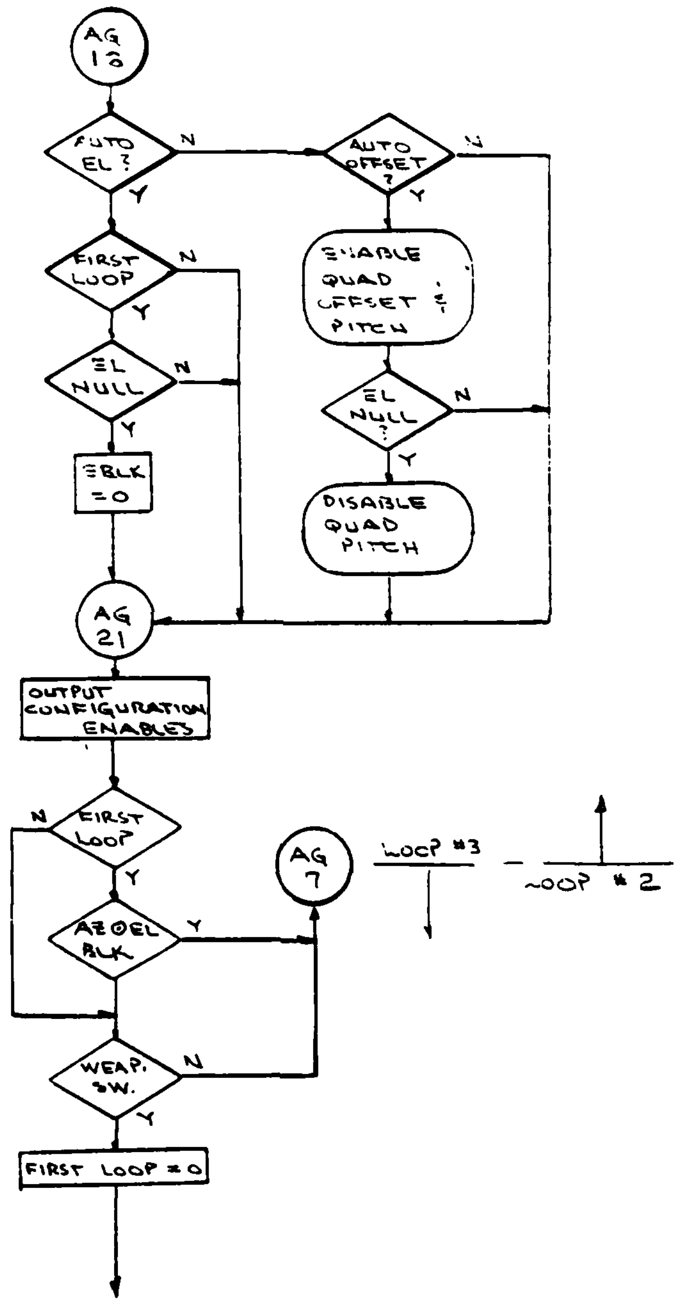
AGLS FUNCTIONAL FLOW

AGLS FUNCTIONAL FLOW

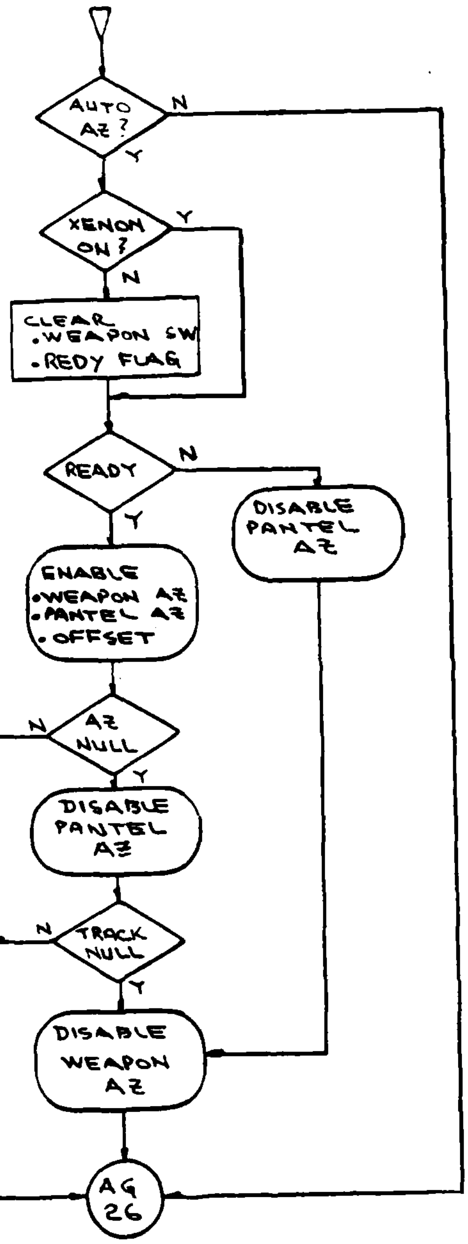


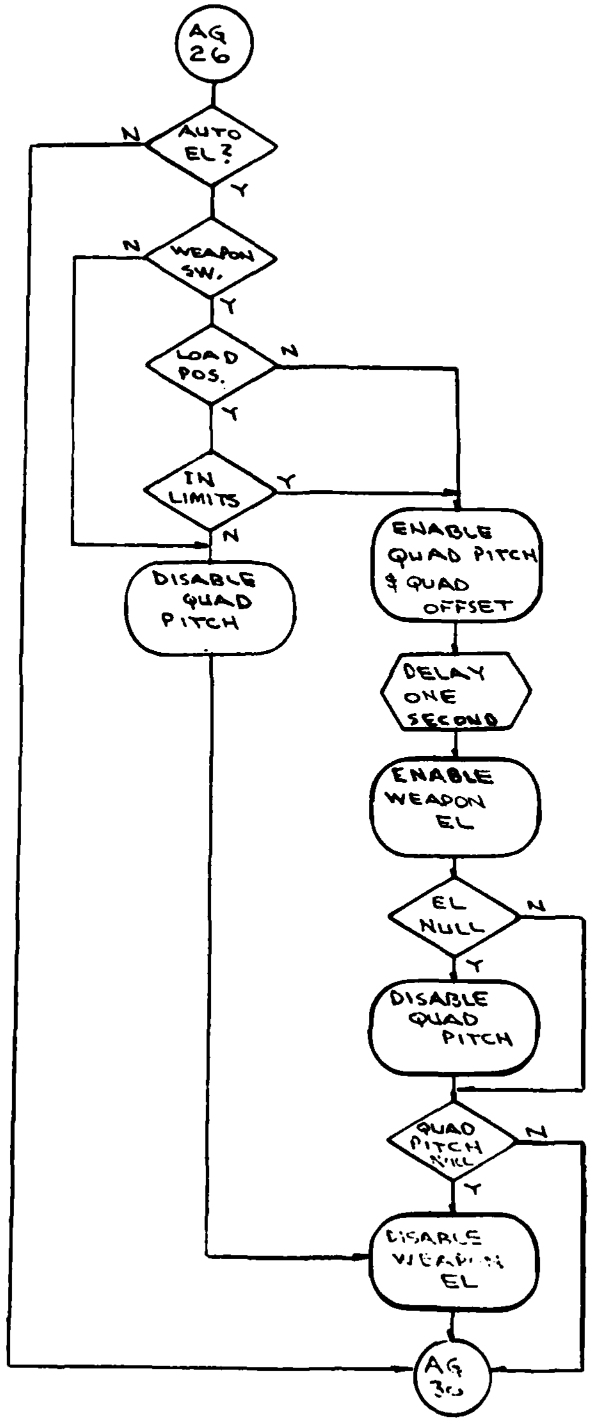

AG 9
READ AZ & EL ENCODER
COMPUTE AZIMUTH & ELEVATION ERROR
READ SERVO OFFSETS
SERVO ON?
AG 2
AUTO AZ?
AUTO OFFSET?
FIRST LOOP
ENABLE PANTEL OFFSET & AZIMUTH
SEARCH SEL.
XENON ON?
AZ NULL
COMMAND LEFT OR RITE SEARCH
ENABLE PANTEL AZIMUTH
DISABLE PANTEL AZ
ENABLE PANTEL OFFSET & AZIMUTH
SEARCHING
DISABLE PANTEL OFFSET
AG 16
N
Y

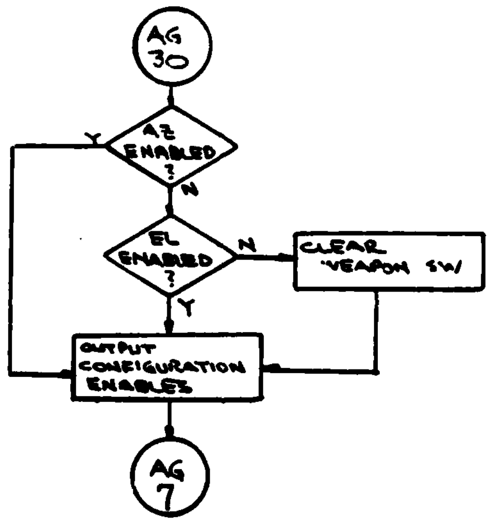
~OCP # 1



AG 18
AUTO EL ?
N
Y
AUTO OFFSET ?
N
Y
FIRST LOOP
N
Y
ENABLE QUAD OFFSET & PITCH
EL NULL
N
Y
EL NULL ?
N
Y
EBLK = 0
DISABLE QUAD PITCH
AG 21
OUTPUT CONFIGURATION ENABLES
N
FIRST LOOP
Y
AZ OEL BLK
Y
AG 7
LOOP #3
LOOP # 2
WEAP. SW.
N
Y
FIRST LOOP = 0

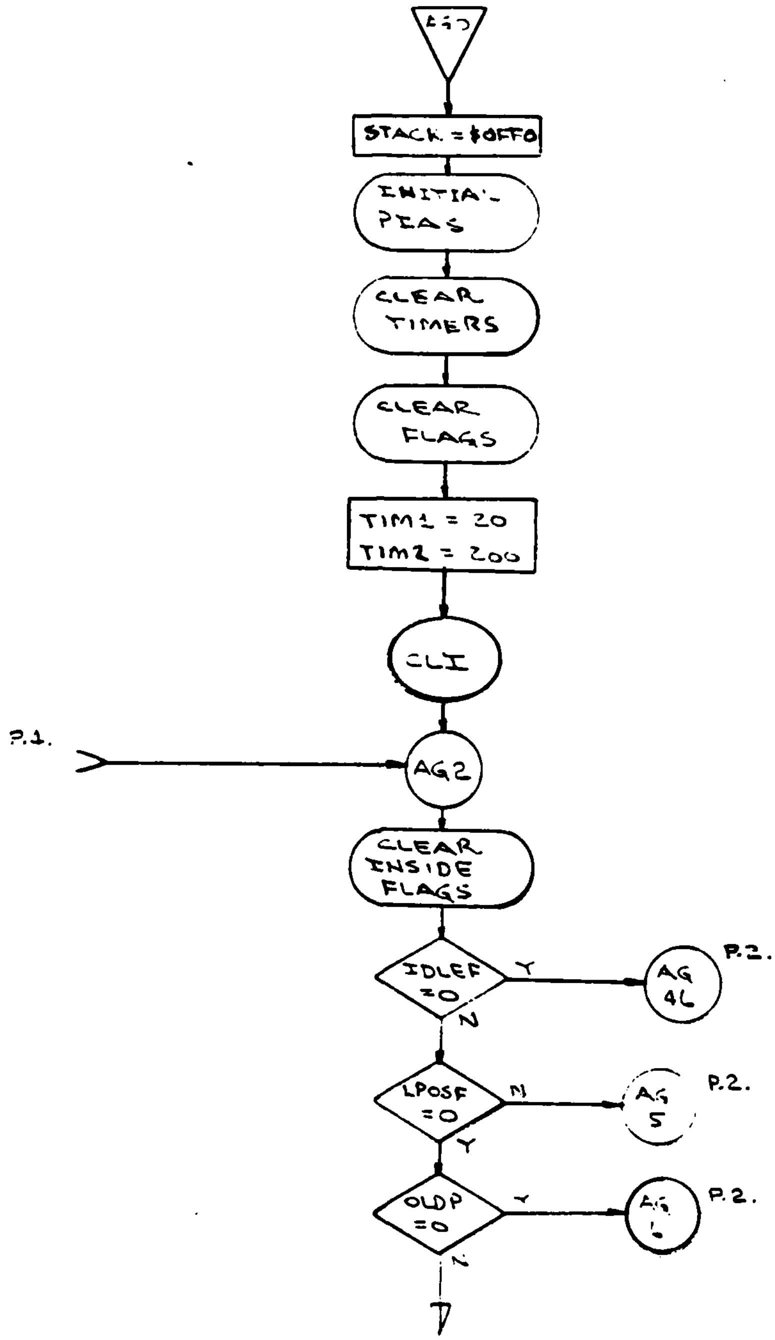

AUTO AZ?
N
Y
XENON ON?
Y
N
CLEAR
• WEAPON SW
• REDY FLAG
READY
N
Y
DISABLE PANTEL AZ
ENABLE
• WEAPON AZ
• PANTEL AZ
• OFFSET
AZ NULL
N
Y
DISABLE PANTEL AZ
TRACK NULL
N
Y
DISABLE WEAPON AZ
AG 26


AG 26
AUTO EL?
WEAPON SW.
LOAD POS.
IN LIMITS
DISABLE QUAD PITCH
ENABLE QUAD PITCH $ QUAD OFFSET
DELAY ONE SECOND
ENABLE WEAPON EL
EL NULL
DISABLE QUAD PITCH
QUAD PITCH NULL
DISABLE WEAPON EL
AG 30
Y
N

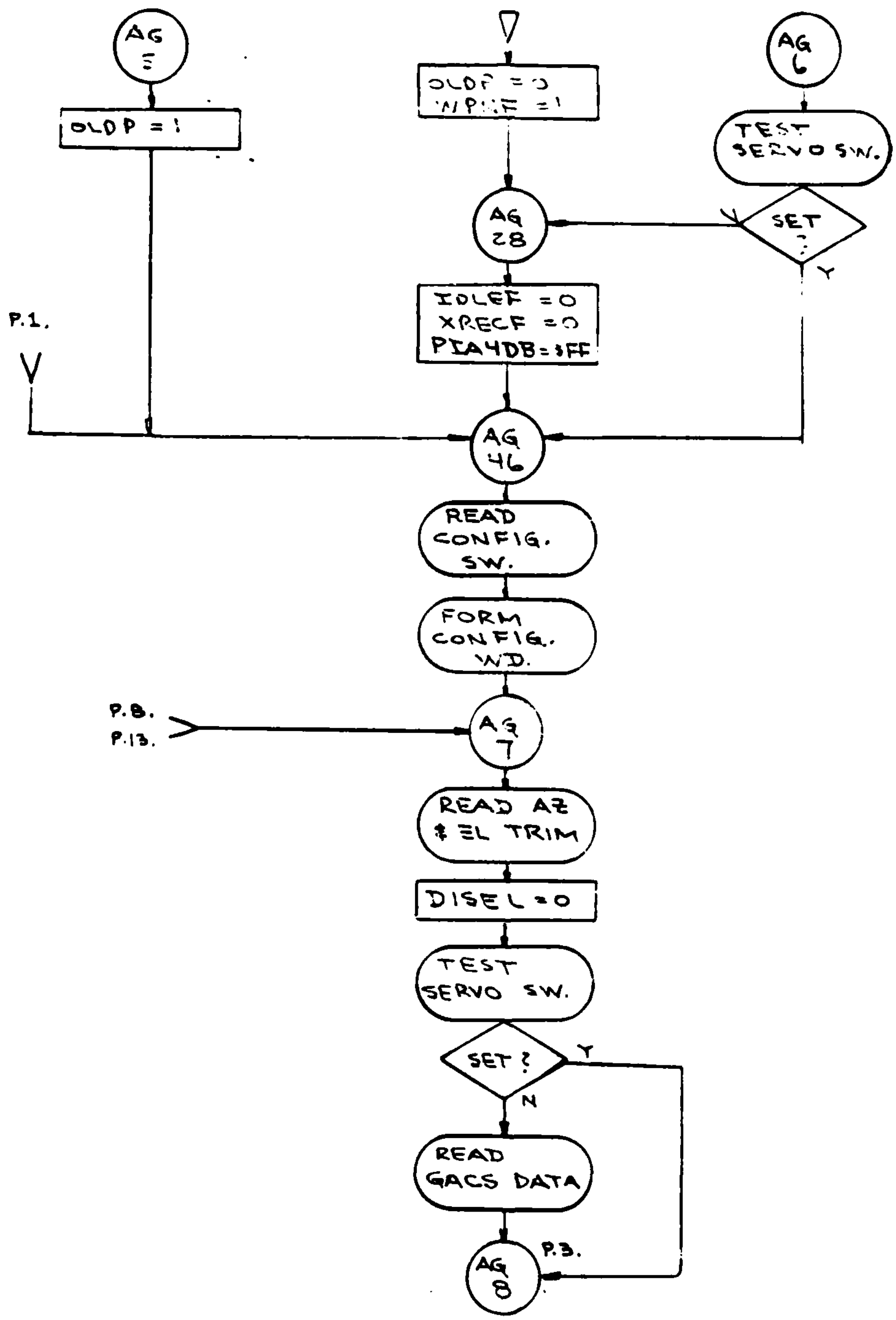

AG
30
AZ
ENABLED
?
Y
N
EL
ENABLED
?
N
CLEAR
WEAPON SW
Y
OUTPUT
CONFIGURATION
ENABLES
AG
7

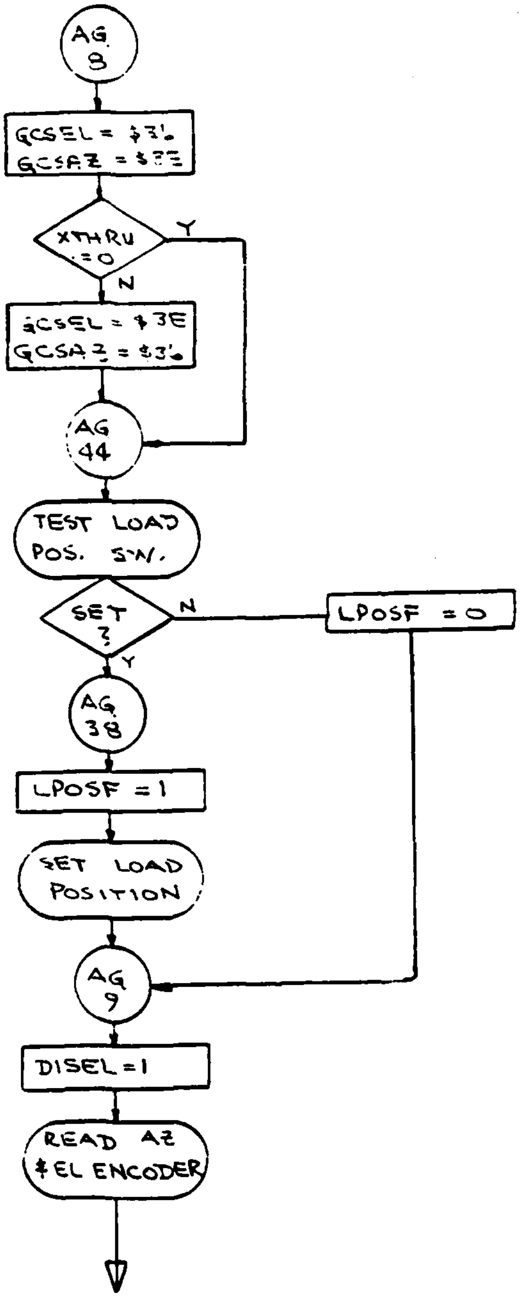
APPENDIX C

AGLS CONTROL PROGRAM FLOW DETAIL,
KEYED TO LISTING OF APPENDIX B

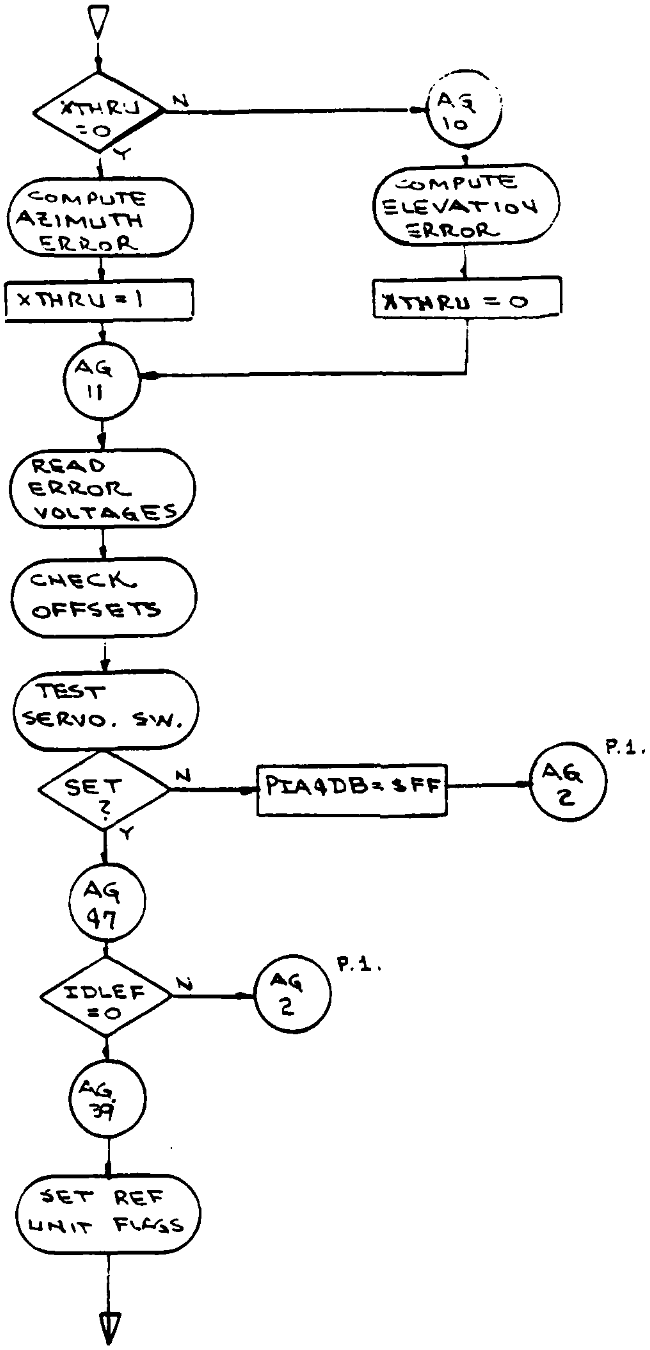


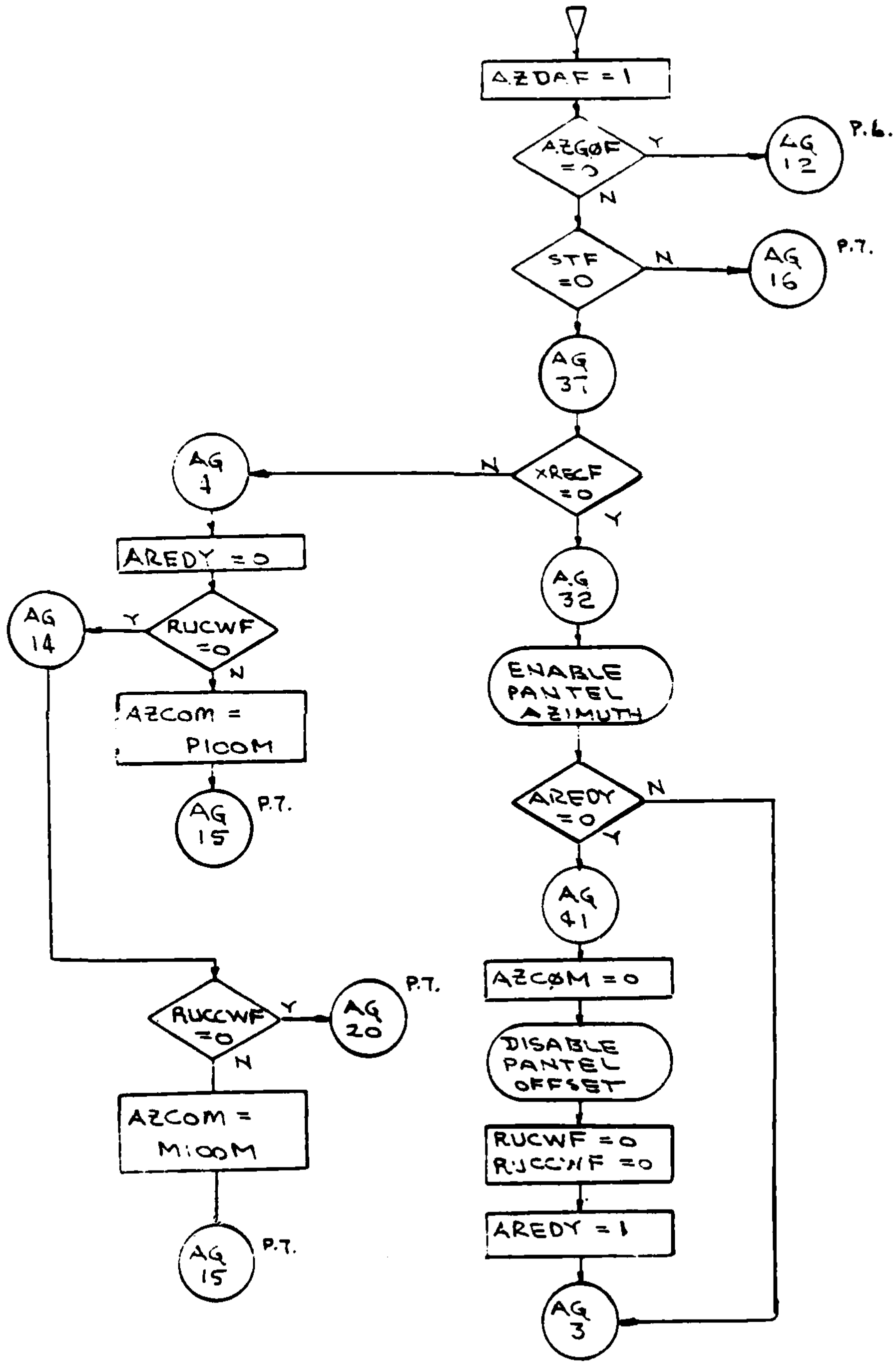
STACK = $0FF0
INITIAL PIAS
CLEAR TIMERS
CLEAR FLAGS
TIM1 = 20
TIM2 = 200
CLI
P.1.
AG2
CLEAR INSIDE FLAGS
IDLEF =0
Y
N
AG 4L
P.2.
LPOSF =0
N
Y
AG 5
P.2.
OLDP =0
Y
N
P.2.


AG 5
OLDP = 1
OLDP = 0
WPNF = 1
AG 6
TEST SERVO SW.
SET ?
Y
AG 28
IDLEF = 0
XRECF = 0
PIA4DB = $FF
P.1.
AG 46
READ CONFIG. SW.
FORM CONFIG. WD.
P.8.
P.13.
AG 7
READ AZ & EL TRIM
DISEL = 0
TEST SERVO SW.
SET ?
Y
N
READ GACS DATA
P.3.
AG 8

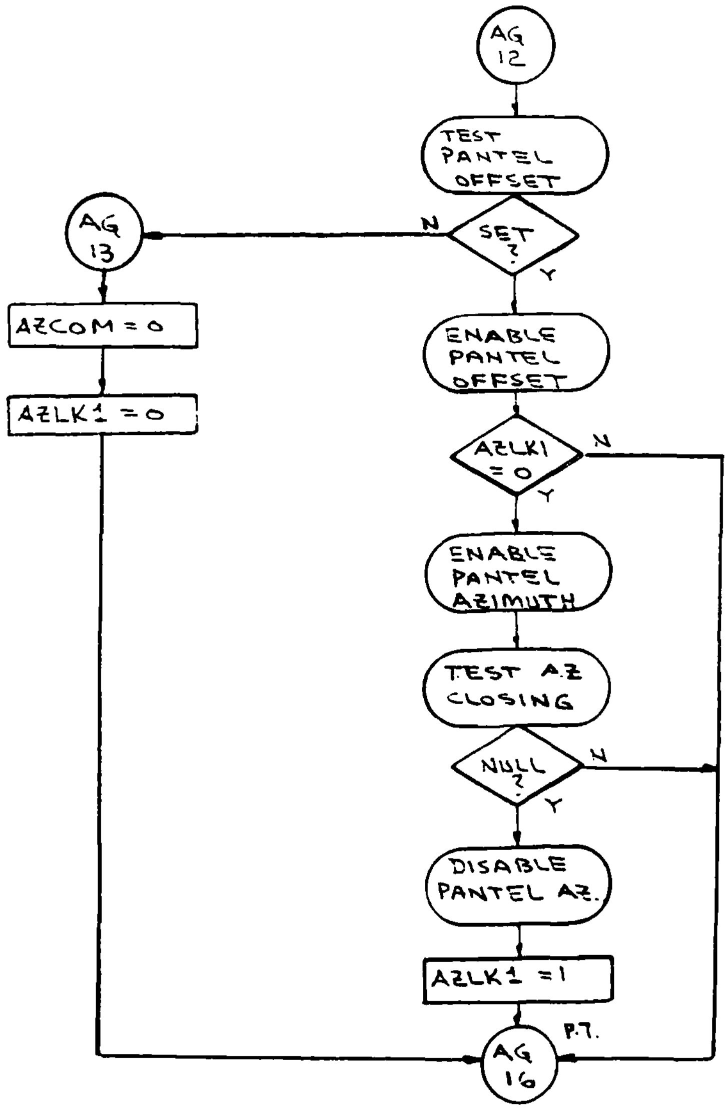


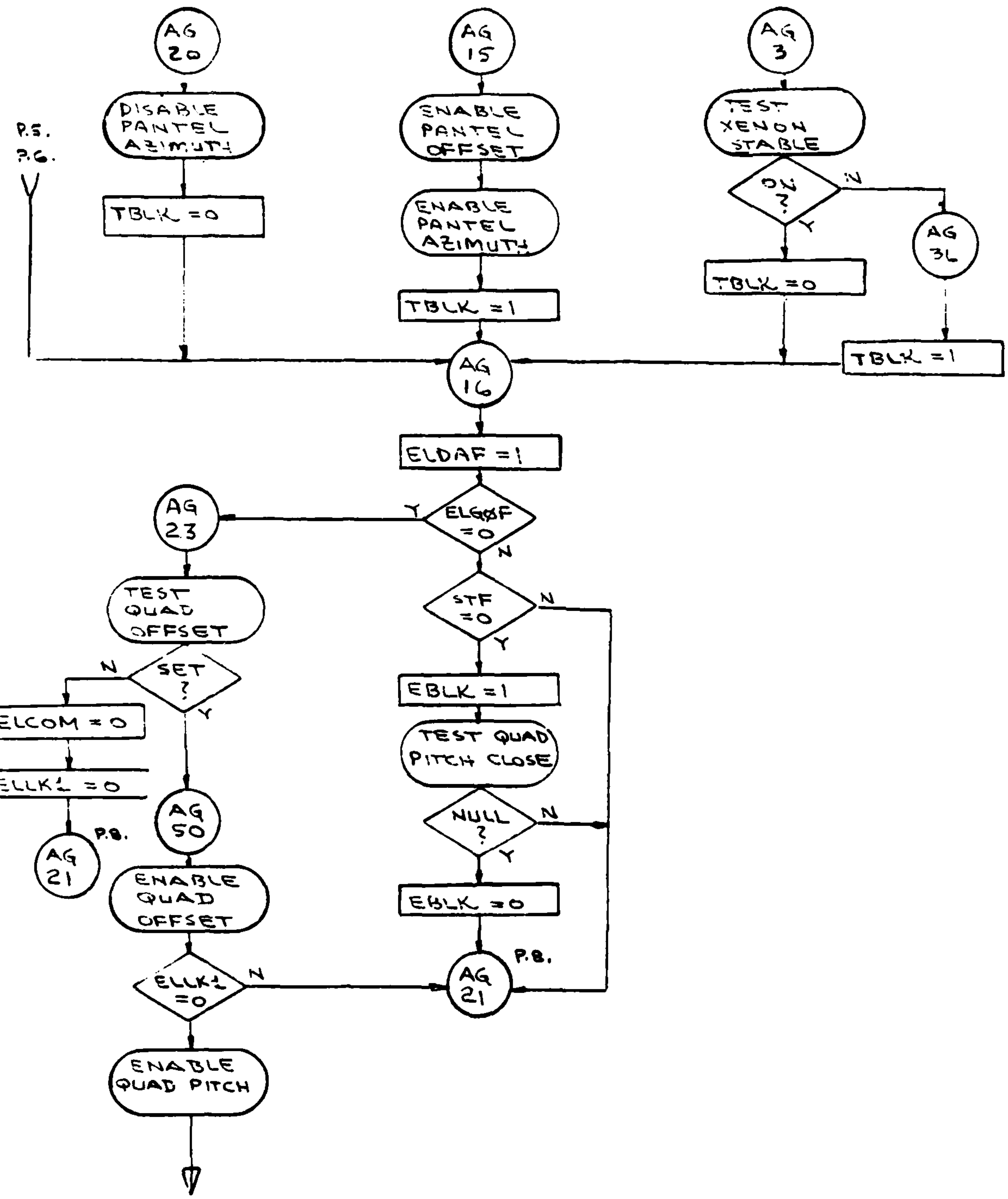
AG 8
GCSEL = $36
GCSAZ = $3E
XTHRU =0
Y
N
GCSEL = $3E
GCSAZ = $36
AG 44
TEST LOAD POS. S/W.
SET ?
N
LPOSF =0
Y
AG 38
LPOSF =1
SET LOAD POSITION
AG 9
DISEL=1
READ AZ & EL ENCODER

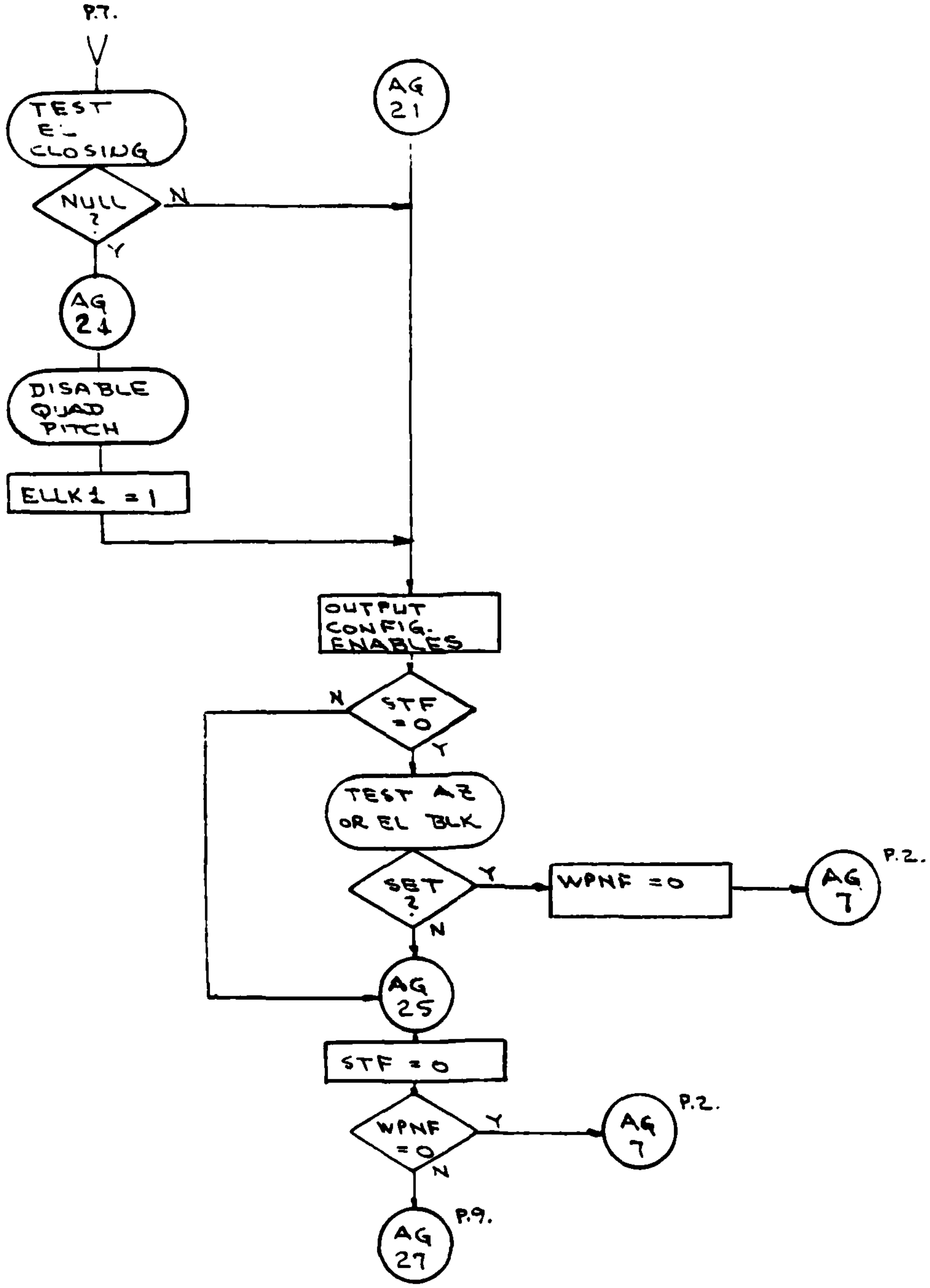

XTHRU =0
N
Y
AG 10
COMPUTE AZIMUTH ERROR
COMPUTE ELEVATION ERROR
XTHRU = 1
XTHRU = 0
AG 11
READ ERROR VOLTAGES
CHECK OFFSETS
TEST SERVO. SW.
SET ?
PIA4DB = $FF
AG 2
P. 1.
AG 47
IDLEF =0
AG 39
SET REF UNIT FLAGS

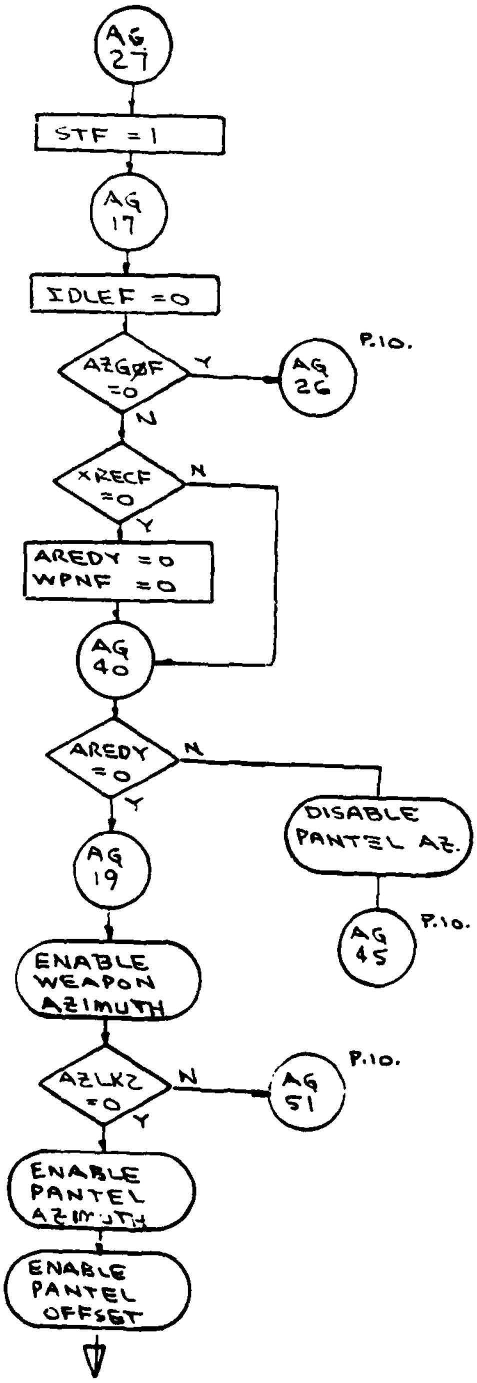


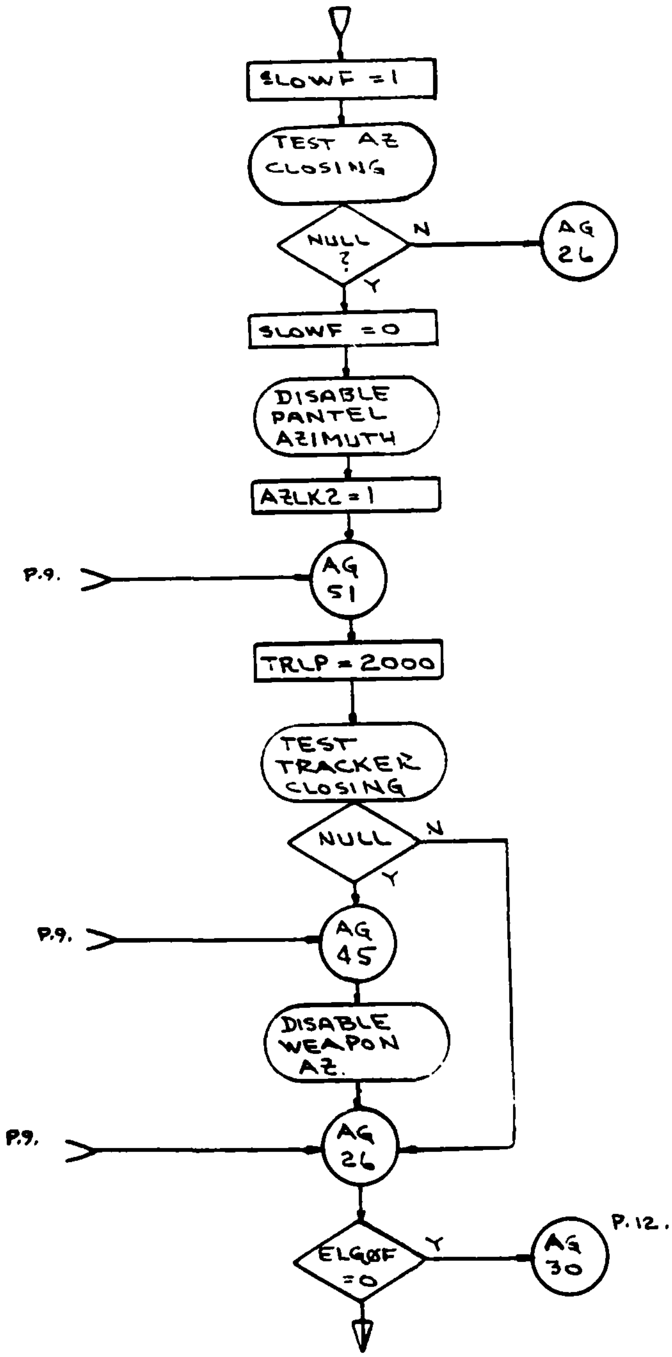
AZDAF = 1
AZGØF =0
Y
N
AG 12
P.6.
STF =0
N
AG 16
P.7.
AG 37
XRECF =0
N
Y
AG 7
AREDY =0
RUCWF =0
Y
N
AG 14
AZCOM = P100M
AG 15
P.7.
AG 32
ENABLE PANTEL AZIMUTH
AREDY =0
N
Y
AG 41
AZCØM =0
DISABLE PANTEL OFFSET
RUCWF =0
RUCCWF =0
AREDY = 1
AG 3
RUCCWF =0
Y
N
AG 20
P.7.
AZCOM = M100M
AG 15
P.7.

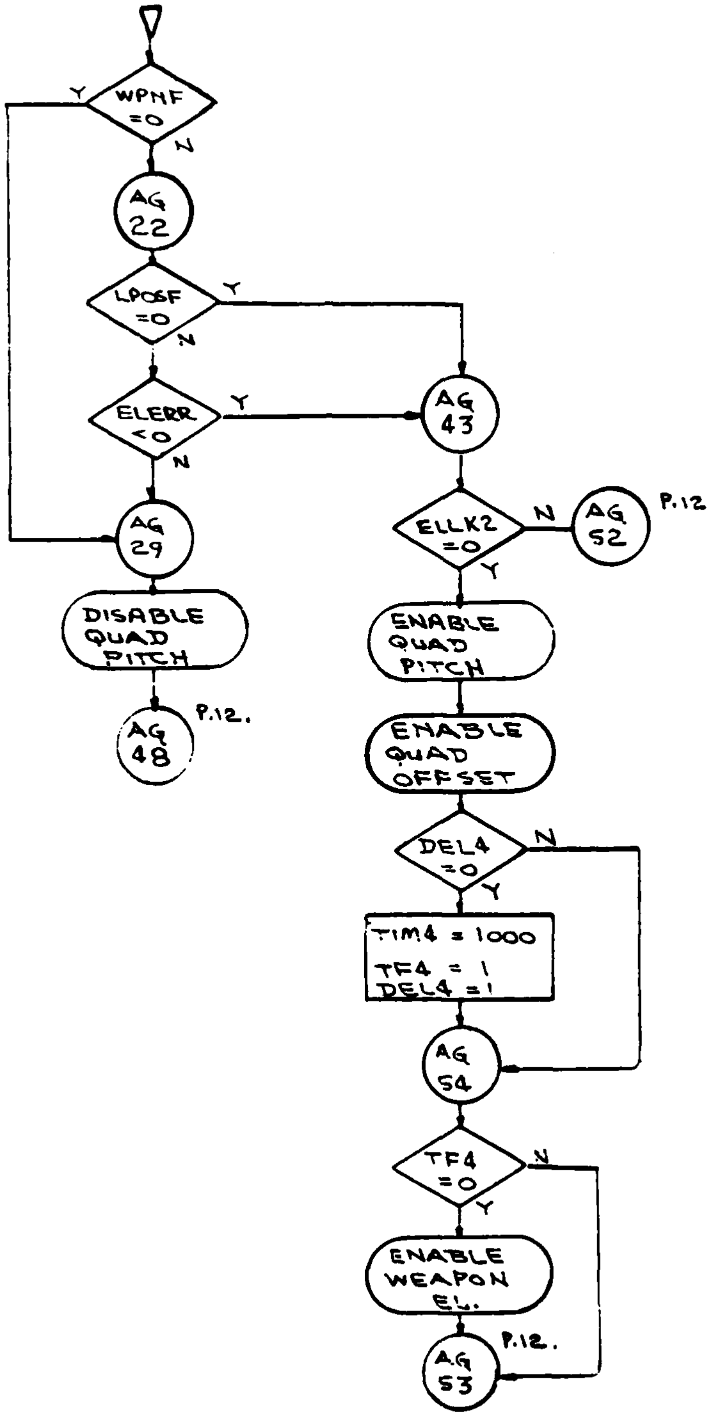

AG 12
TEST PANTEL OFFSET
SET ?
N
Y
AG 13
AZCOM = 0
AZLK1 = 0
ENABLE PANTEL OFFSET
AZLK1 = 0
N
Y
ENABLE PANTEL AZIMUTH
TEST AZ CLOSING
NULL ?
N
Y
DISABLE PANTEL AZ.
AZLK1 = 1
P.7.
AG 16


AG 20
DISABLE PANTEL AZIMUTH
TBLK = 0
AG 15
ENABLE PANTEL OFFSET
ENABLE PANTEL AZIMUTH
TBLK = 1
AG 3
TEST XENON STABLE
ON ?
N
Y
AG 36
TBLK = 0
TBLK = 1
P.5.
P.6.
AG 16
ELDAF = 1
ELGØF = 0
Y
N
AG 23
TEST QUAD OFFSET
SET ?
N
Y
ELCOM = 0
ELLK2 = 0
P.8.
AG 21
AG 50
ENABLE QUAD OFFSET
ELLK2 = 0
N
ENABLE QUAD PITCH
STF = 0
N
Y
EBLK = 1
TEST QUAD PITCH CLOSE
NULL ?
N
Y
EBLK = 0
P.8.
AG 21

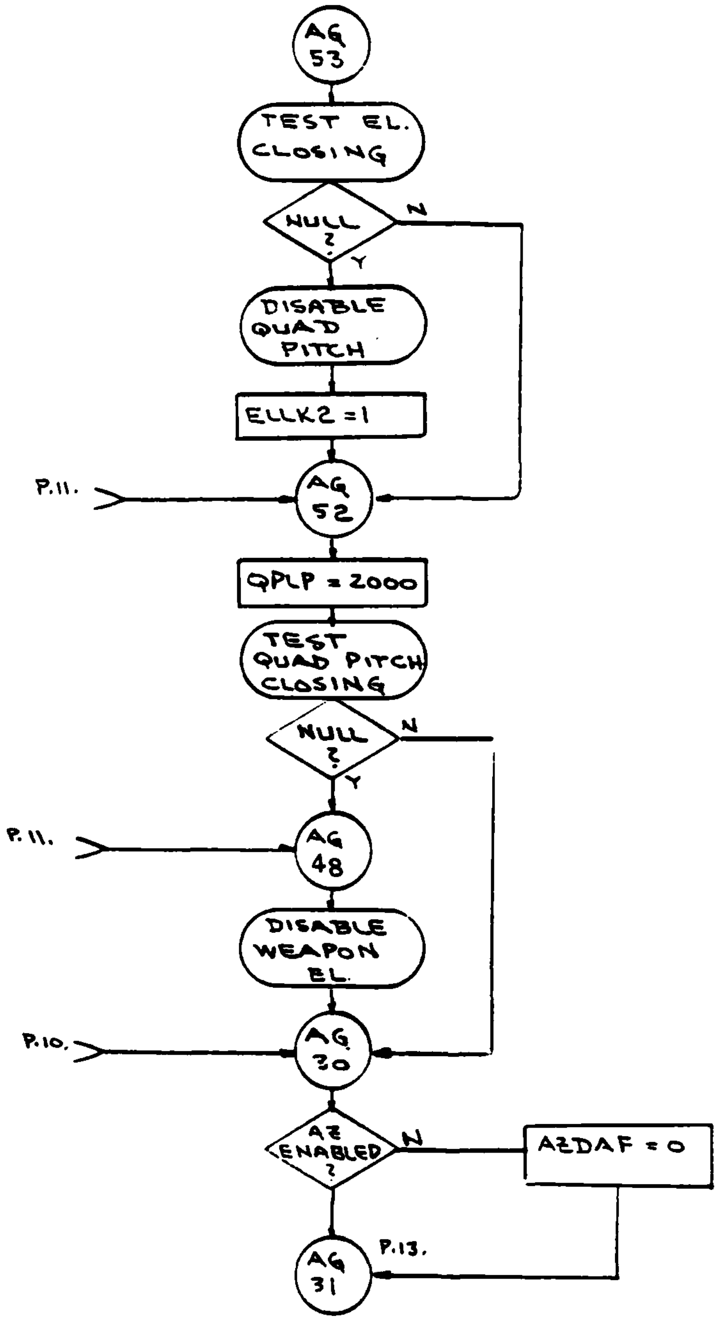

P.7.
TEST
EL
CLOSING
NULL
?
N
Y
AG
21
AG
21
DISABLE
QUAD
PITCH
ELLK1 = 1
OUTPUT
CONFIG.
ENABLES
N
STF
= 0
Y
TEST AZ
OR EL BLK
SET
?
Y
N
WPNF = 0
AG
7
P.2.
AG
25
STF = 0
WPNF
= 0
Y
N
AG
7
P.2.
AG
27
P.9.

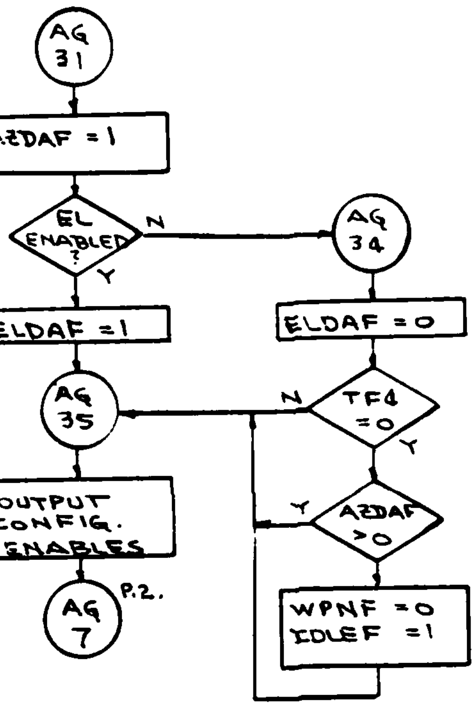

AG 27
STF = 1
AG 17
IDLEF = 0
AZGØF = 0
Y
AG 26
P.10.
N
XRECF = 0
N
Y
AREDY = 0
WPNF = 0
AG 40
AREDY = 0
N
Y
DISABLE PANTEL AZ.
AG 45
P.10.
AG 19
ENABLE WEAPON AZIMUTH
AZLKZ = 0
N
AG 51
P.10.
Y
ENABLE PANTEL AZIMUTH
ENABLE PANTEL OFFSET


SLOWF =1
TEST AZ CLOSING
NULL ?
N
AG 26
Y
SLOWF =0
DISABLE PANTEL AZIMUTH
AZLK2 = 1
P.9.
AG 51
TRLP = 2000
TEST TRACKER CLOSING
NULL
N
Y
P.9.
AG 45
DISABLE WEAPON AZ.
P.9.
AG 26
ELGØF =0
Y
AG 30
P. 12.

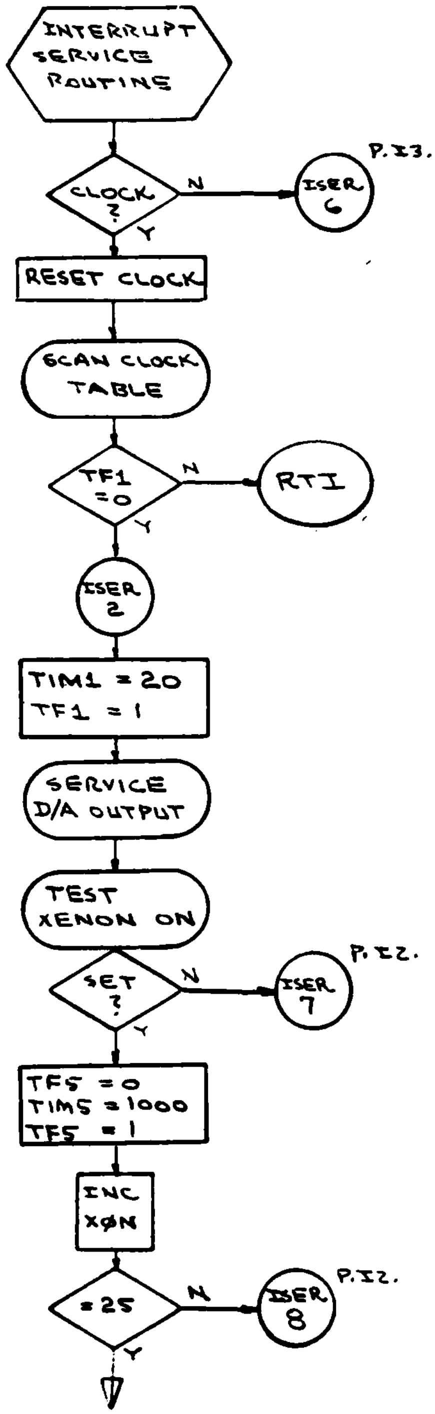


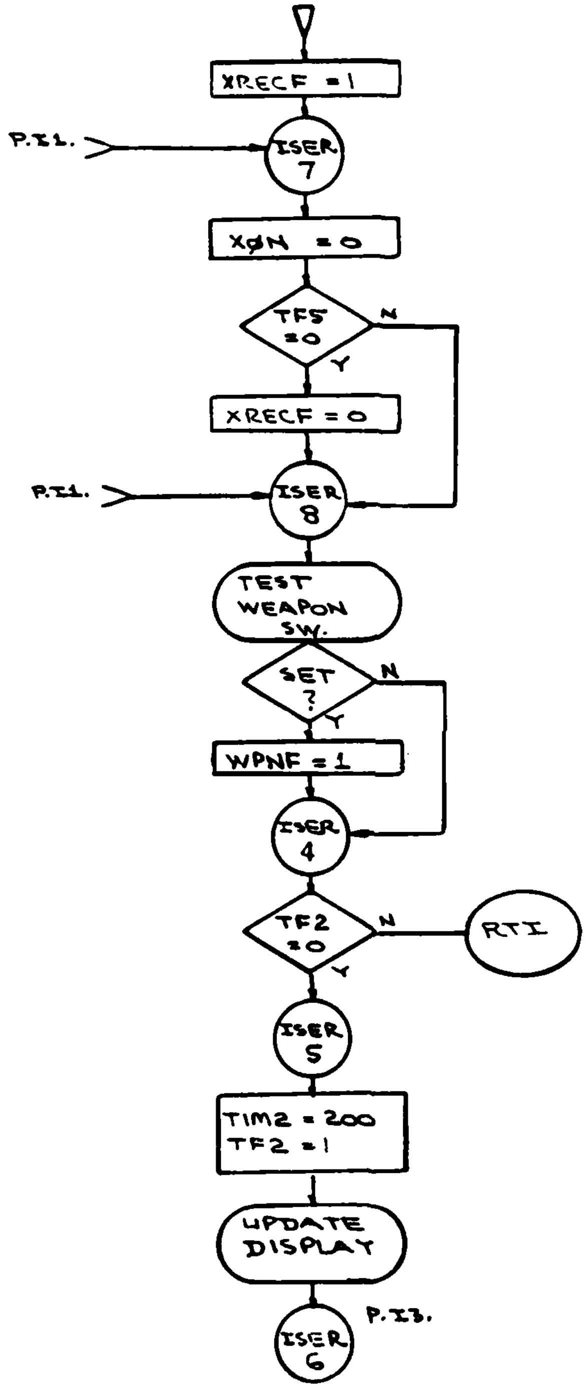
WPNF =0
Y
N
AG 22
LPOSF =0
Y
N
ELERR <0
Y
N
AG 29
DISABLE QUAD PITCH
AG 48
P.12.
AG 43
ELLK2 =0
N
Y
AG 52
P.12
ENABLE QUAD PITCH
ENABLE QUAD OFFSET
DEL4 =0
N
Y
TIM4 = 1000
TF4 = 1
DEL4 = 1
AG 54
TF4 =0
N
Y
ENABLE WEAPON EL.
AG 53
P.12.

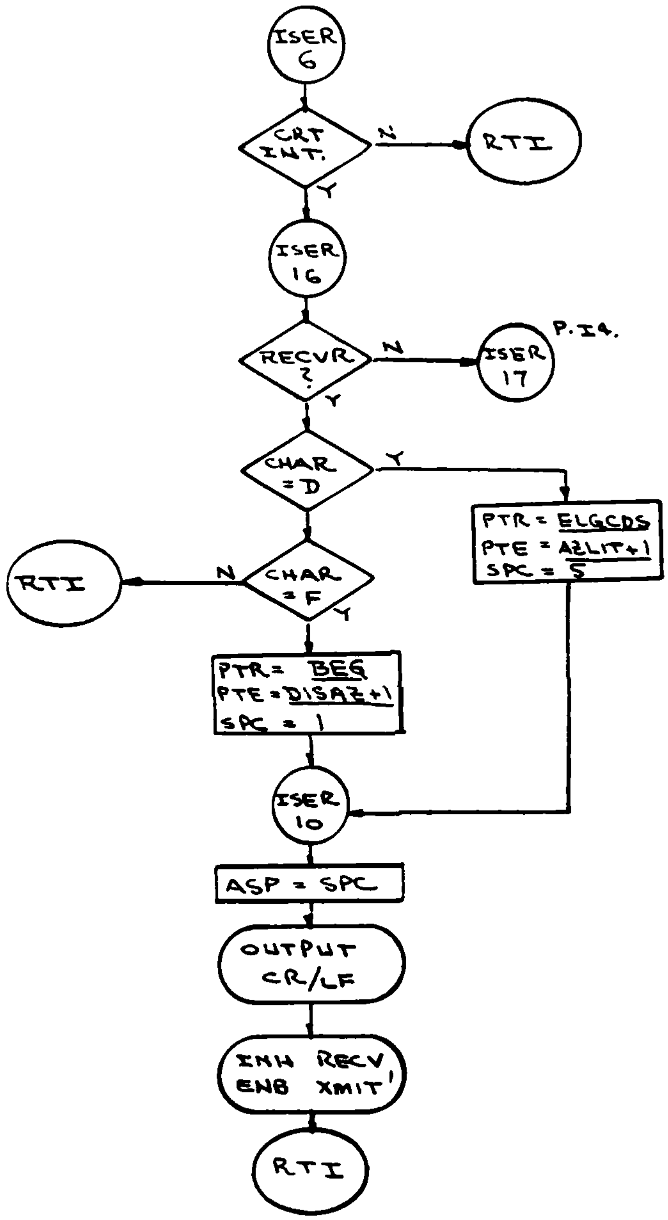

AG 53
TEST EL. CLOSING
NULL ?
N
Y
DISABLE QUAD PITCH
ELLK2 = 1
P.11.
AG 52
QPLP = 2000
TEST QUAD PITCH CLOSING
NULL ?
N
Y
P.11.
AG 48
DISABLE WEAPON EL.
P.10.
AG 30
AZ ENABLED ?
N
AZDAF = 0
P.13.
AG 31

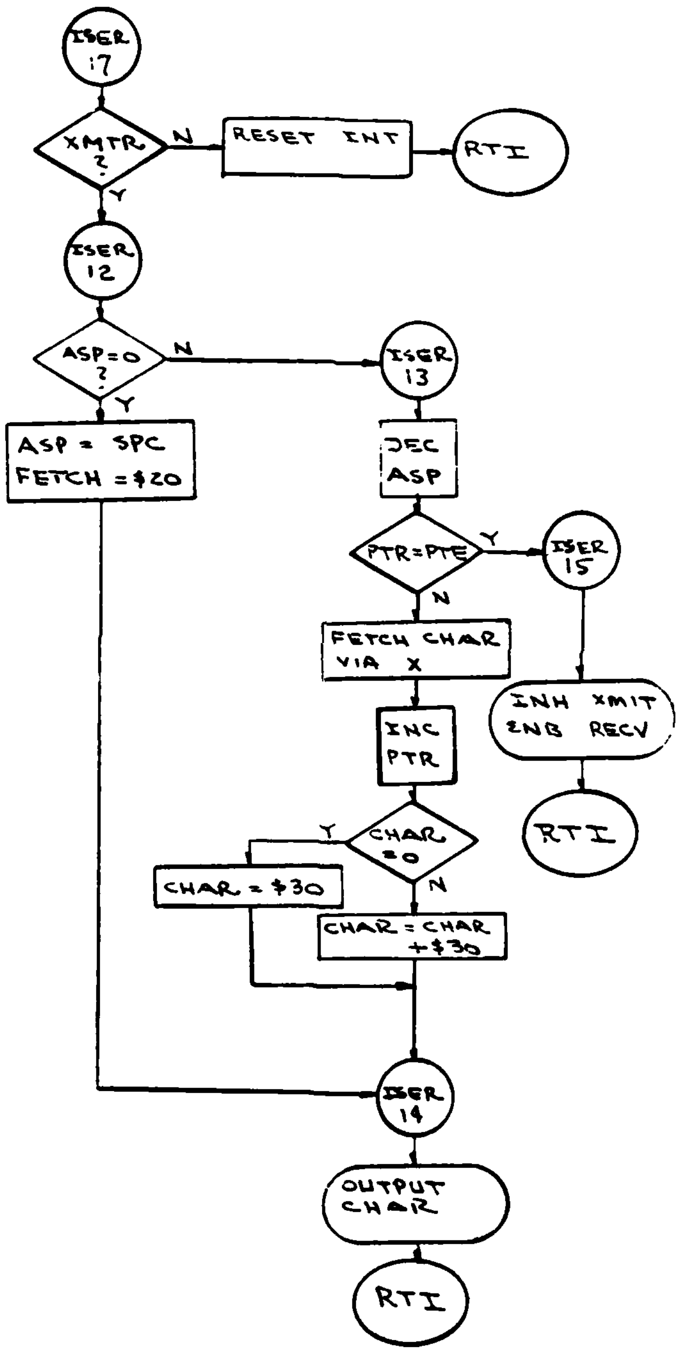


AG 31
AZDAF = 1
EL ENABLED ?
N
Y
AG 34
ELDAF = 1
ELDAF = 0
AG 35
N
TF4 = 0
Y
OUTPUT CONFIG. ENABLES
Y
AZDAF > 0
AG 7
P.2.
WPNF = 0
IDLEF = 1


INTERRUPT SERVICE ROUTINE
CLOCK ?
N
Y
ISER 6
P. I3.
RESET CLOCK
SCAN CLOCK TABLE
TF1 = 0
N
Y
RTI
ISER 2
TIM1 = 20
TF1 = 1
SERVICE D/A OUTPUT
TEST XENON ON
SET ?
N
Y
ISER 7
P. I2.
TF5 = 0
TIM5 = 1000
TF5 = 1
INC XØN
= 25
N
Y
ISER 8
P. I2.


XRECF = 1
P.I1.
ISER 7
XØN = 0
TF5 = 0
N
Y
XRECF = 0
P.I1.
ISER 8
TEST WEAPON SW.
SET ?
N
Y
WPNF = 1
ISER 4
TF2 = 0
N
Y
RTI
ISER 5
TIM2 = 200
TF2 = 1
UPDATE DISPLAY
P.I3.
ISER 6


ISER 6
CRT INT.
N
Y
RTI
ISER 16
RECVR ?
N
Y
ISER 17
P. I4.
CHAR = D
Y
PTR = ELGCDS
PTE = AZLIT+1
SPC = 5
CHAR = F
N
Y
RTI
PTR = BEG
PTE = DISAZ+1
SPC = 1
ISER 10
ASP = SPC
OUTPUT CR/LF
INH RECV
ENB XMIT
RTI


ISER 17
XMTR ?
N
RESET INT
RTI
Y
ISER 12
ASP=0 ?
N
ISER 13
Y
ASP = SPC
FETCH = $20
DEC ASP
PTR=PTE
Y
ISER 15
N
FETCH CHAR VIA X
INH XMIT
ENB RECV
INC PTR
RTI
CHAR =0
Y
CHAR = $30
N
CHAR = CHAR +$30
ISER 14
OUTPUT CHAR
RTI

APPENDIX D

AGLS CONTROL PROGRAM SOURCE LISTING

```
                        NAM     AGLS
                        OPT     O
               * REVISED 3/28/78
               * CRT DISPLAY OPTION
               *
               **********AGLS1***********
               * PIA EQUATES
               *
               * PIA0=AUTO SWITCHES(A),CLOCK RATE(B)
               *
2800           PIA0DA EQU     $2800
2801           PIA0DB EQU     PIA0DA+1
2802           PIA0CA EQU     PIA0DA+2
2803           PIA0CB EQU     PIA0DA+3
               *
               * PIA1=GACS LSB(A),GACS MSB(B)
               *
2400           PIA1DA EQU     $2400
2401           PIA1DB EQU     PIA1DA+1
2402           PIA1CA EQU     PIA1DA+2
2403           PIA1CB EQU     PIA1DA+3
               *
               * PIA2=QUAD EL ENCODER;MSB=A,LSB=B
               *
2404           PIA2DA EQU     $2404
2405           PIA2DB EQU     PIA2DA+1
2406           PIA2CA EQU     PIA2DA+2
2407           PIA2CB EQU     PIA2DA+3
               *
               * PIA3=PANTEL AZ ENCODER;MSB=A,LSB=B
               *
2408           PIA3DA EQU     $2408
2409           PIA3DB EQU     PIA3DA+1
240A           PIA3CA EQU     PIA3DA+2
240B           PIA3CB EQU     PIA3DA+3
               *
               * PIA4=1/10 ENCODER OUTPUTS(A),ENABLE OUTPUTS(B)
               *
240C           PIA4DA EQU     $240C
240D           PIA4DB EQU     PIA4DA+1
240E           PIA4CA EQU     PIA4DA+2
240F           PIA4CB EQU     PIA4DA+3
               *
               * PIA5=MUX A/D DATA (A),MUX ADDR(B)
               *
2410           PIA5DA EQU     $2410
2411           PIA5DB EQU     PIA5DA+1
2412           PIA5CA EQU     PIA5DA+2
2413           PIA5CB EQU     PIA5DA+3
               *
               * PIA6=EL TRIM A/D(A),AZ TRIM A/D(B)
               *
2414           PIA6DA EQU     $2414
2415           PIA6DB EQU     PIA6DA+1
```



```
2416           PIA6CA EQU     PIA6DA+2
2417           PIA6CB EQU     PIA6DA+3
               *
               * PIA7=D/A CONVERTER
               *
2418           PIA7DA EQU     $2418
2419           PIA7DB EQU     PIA7DA+1
241A           PIA7CA EQU     PIA7DA+2
241B           PIA7CB EQU     PIA7DA+3
               *
               * PIA8=DISPLAY(4) AND SWITCHES(4),(A)
               *       MISC INPUTS(3) AND DISP ADDR(5),(B)
               *
241C           PIA8DA EQU     $241C
241D           PIA8DB EQU     PIA8DA+1
241E           PIA8CA EQU     PIA8DA+2
241F           PIA8CB EQU     PIA8DA+3
               *
               *
               * PIA 0 SWITCH MASKS
               *
0080           LDP1M   EQU     %10000000
0040           LDP2M   EQU     %1000000
0020           PLRQM   EQU     %100000
0010           QLRQM   EQU     %10000
0008           PORQM   EQU     %1000
0004           QORQM   EQU     %100
0002           AZRQM   EQU     %10
0001           ELRQM   EQU     %1
               *
               * PIA 0 STROBE
2802           RWDOG   EQU     PIA0CA
               *
               * PIA1 ENABLE ADDRESSES
               *
2402           GCSEL   EQU     PIA1CA
2403           GCSAZ   EQU     PIA1CB
               *
               * PIA4 (A) INPUT MASKS
               *
000F           ELTM    EQU     %1111
00F0           AZTM    EQU     %11110000
               *
               * PIA4 (B) ENABLE BITS
               * 0=TRUE
               *
00FE           PLGO    EQU     %11111110
00FD           OCGO    EQU     %11111101
00FB           PAGO    EQU     %11111011
00F7           QPGO    EQU     %11110111
00EF           POGO    EQU     %11101111
00DF           QOGO    EQU     %11011111
00BF           AZGO    EQU     %10111111
007F           ELGO    EQU     %01111111
```

```
                *
                * ERROR VOLTAGE MASKS
     0001       OPMA    EQU     1
     0002       QCMA    EQU     2
     0003       MPMA    EQU     3
     0004       MCMA    EQU     4
     0005       PAMA    EQU     5
                *
                * PIA5 ENABLES AND FLAG ADDRESSES
                *
     2412       SAD5    EQU     PIA5CA
     2413       EMUX    EQU     PIA5CB
     2412       CCM5    EQU     PIA5CA
                *
                * PIA6 ENABLES AND FLAG ADDRESSES
                *
     2416       SAD6A   EQU     PIA6CA
     2417       SAD6B   EQU     PIA6CB
     2416       CCM6A   EQU     PIA6CA
     2417       CCM6B   EQU     PIA6CB
                *
                * PIA7 ENABLE ADDRESSES
                *
     241A       DAEL    EQU     PIA7CA
     241B       DAAZ    EQU     PIA7CB
                *
                * PIA8 (A) MASKS
                *
     000F       DOM     EQU     %1111
     0010       RUCWM   EQU     %10000
     0020       RUCCWM  EQU     %100000
     0040       LPOSM   EQU     %1000000
     0080       WPNM    EQU     %10000000
                *
                * PIA8(B) MASKS
                *
     001F       DAOM    EQU     %11111
     0040       XRECM   EQU     %1000000
     0080       SRVOM   EQU     %10000000
                * PIA8 ENABLE ADDRESSES
                *
     241E       DS8     EQU     PIA8CA
     241F       XCF     EQU     PIA8CB
     241F       LAMP    EQU     PIA8CB
                *
                * ACIA EQUATES
                *
     3002       AC2C    EQU     $3002
     3002       AC2S    EQU     AC2C
     3003       AC2T    EQU     $3003
     3003       AC2R    EQU     AC2T
                *
                * 300 BAUD
                * DIVIDE BY 16 ACIA
```



```
                *
          002A  XIE     EQU     %00101010
          008A  RIE     EQU     %10001010
                *
                * MISC EQUATES
                *
          0200  LOAD1   EQU     $0200
          0300  LOAD2   EQU     $0300
          0400  LOAD3   EQU     $0400
          0500  LOAD4   EQU     $0500
                *
          0A00  P100M   EQU     $0A00
          8A00  M100M   EQU     $8A00
          85FF  HAFBAK  EQU     $85FF
          8FFF  FULBAK  EQU     $8FFF
                *
                * OFFSET ERROR TABLE
                *
          000A  EQP     EQU     10
          0005  EQC     EQU     5
          0005  EMP     EQU     5
          0005  EMC     EQU     5
          0001  EPA     EQU     1
          0001  AZLIM   EQU     1
          0001  ELLIM   EQU     1
     0000               ORG     0
     0000 0001          RMB     1
                * EXEC RAM
     0001 0001  MSBY    RMB     1              DEC-BIN ROUTINE
     0002 0001  LSBY    RMB     1
     0003 0001  TMP     RMB     1
                *
                * INTERRUPT DRIVEN TIMERS
                *
                * TIMER TABLE (DECREMENT)
          0004  TMTB    EQU     *
     0004 0001  TF1     RMB     1
     0005 0002  TIM1    RMB     2
                *
     0007 0001  TF2     RMB     1
     0008 0002  TIM2    RMB     2
                *
     000A 0001  TF3     RMB     1
     000B 0002  TIM3    RMB     2
                *
     000D 0001  TF4     RMB     1
     000E 0002  TIM4    RMB     2
                *
     0010 0001  TF5     RMB     1
     0011 0002  TIM5    RMB     2
                *
                * FLAG BUFFER
                *
                *
```

```
      0013  BEG     EQU     *
0013  0001  OLP     RMB     1
0014  0001  AZGOF   RMB     1
0015  0001  ELGOF   RMB     1
0016  0001  WPNF    RMB     1
0017  0001  XRECF   RMB     1
0018  0001  LPOSF   RMB     1
0019  0001  XTHRU   RMB     1
001A  0001  IDLEF   RMB     1
001B  0001  RUCWF   RMB     1
001C  0001  STF     RMB     1
001D  0001  TBLK    RMB     1
001E  0001  EBLK    RMB     1
001F  0001  ELLK1   RMB     1
0020  0001  AZLK1   RMB     1
0021  0001  ELLK2   RMB     1
0022  0001  AZLK2   RMB     1
0023  0001  DEL4    RMB     1
0024  0001  RUCCWF  RMB     1
0025  0001  AREDY   RMB     1
0026  0001  SIG     RMB     1
0027  0001  CKOF    RMB     1
0028  0001  NEGF    RMB     1
0029  0001  SLOWF   RMB     1
002A  0001  DTHRU   RMB     1
002B  0001  DISEL   RMB     1
002C  0001  DISAZ   RMB     1
                    *
002D  0001  CONGO   RMB     1
002E  0001  CONTEM  RMB     1
002F  0001  AZTRM   RMB     1
0030  0001  ELTRM   RMB     1
0031  0002  ELGCS   RMB     2
0033  0002  AZGCS   RMB     2
                    * ERROR VOLTAGE BUFFER
      0035  ERRBUF  EQU     *
0035  0007          RMB     7
003C  0001  MUXADD  RMB     1
003D  0001  NUMRED  RMB     1
003E  0001  PREVAL  RMB     1
003F  0001  LITE    RMB     1
0040  0001  FFLAG   RMB     1
                    * INTERRUPT SERVICE ROUTINE FLAGS
0041  0001  ELDAF   RMB     1
0042  0001  AZDAF   RMB     1
0043  0001  DISADR  RMB     1
0044  0002  ACT     RMB     2
0046  0001  PASSAZ  RMB     1
0047  0001  PASSEL  RMB     1
0048  0002  PTR     RMB     2
004A  0002  PTE     RMB     2
004C  0001  SPC     RMB     1
004D  0001  ASP     RMB     1
004E  0001  XON     RMB     1
```



```
                    *
004F  0002  AZCOM   RMB     2
0051  0002  ELCOM   RMB     2
                    *
0053  0002  AZERR   RMB     2
0055  0002  ELERR   RMB     2
                    *
0057  0001  AZCNT   RMB     1
0058  0001  ELCNT   RMB     1
0059  0002  OPCNT   RMB     2
005B  0002  TRCNT   RMB     2
005D  0002  TRLP    RMB     2
005F  0001  XTIME   RMB     1
0060  0002  OPLP    RMB     2
                    *
0062  0002  A1      RMB     2
0064  0002  A2      RMB     2
0066  0002  S1      RMB     2
0068  0002  S2      RMB     2
006A  0002  X1      RMB     2
006C  0002  X2      RMB     2
                    *
006E  0002  OUTX    RMB     2
0070  0002  HOLDX   RMB     2
0072  0002  TIZX    RMB     2
0074  0002  STORX   RMB     2
0076  0002  ADDX1   RMB     2
0078  0002  SUBX1   RMB     2
007A  0002  SUBX2   RMB     2
007C  0002  TX      RMB     2
                    *
007E  0001  CNX     RMB     1
007F  0001  OLDAX   RMB     1
0080  0001  OLDBX   RMB     1
0081  0001  THI     RMB     1
0082  0001  TLO     RMB     1
                    *
0083  0001  HOLDB   RMB     1
0084  0002  KEEP    RMB     2
0086  0001  TF      RMB     1
0087  0002  GACTEM  RMB     2
                    *
0089  0001  SAVA    RMB     1
008A  0001  SAVB    RMB     1
                    * DISPLAYS BUFFER
                    * ELEVATION
                    * GACS
008B  0005  ELGCDS  RMB     5
                    * ENCODER
0090  0005  ELDISP  RMB     5
                    * ERROR
0095  0005  ELERD   RMB     5
                    *
                    *
```

```
03250                    * AZIMUTH
03260                    * GACS
03270 009A 0005          AZGCDS RMB     5
03280                    * ENCODER
03290 009F 0005          AZDISP RMB     5
03300                    * ERROR
03310 00A4 0005          AZERD  RMB     5
03320                    *
03330                    * OFFSET ERROR WORDS
03340 00A9 0001          ELLIT  RMB     1
03350 00AA 0001          AZLIT  RMB     1
03360                    * TEMPORARY BUFFERS
03370 00AB 0005          TEMSUB RMB     5
03380 00B0 0005          TEMBCD RMB     5
03390 00B5 0005          RESULT RMB     5
03400 00BA 0005          ELTEMP RMB     5
03410 00BF 0005          AZTEMP RMB     5
03420 00C4 0005          ADJX   RMB     5
03430      00C9          END    EQU     *
```



```
03450                     *
03460                     * AGLS CONTROL PROGRAM
03470                     *
03480                     * ENTER HERE ON RESET
03490 4000                        ORG    $4000
03500      4000           AG0     EQU    *
03510 4000 8E OFF0                LDS    #$0FF0
03520 4003 CE 4A36                LDX    #ISER
03530 4006 FF 0FF8                STX    $0FF8
03540                     * INITIAL PIAS
03550 4009 BD 4968                JSR    PIAS
03560                     * CLEAR TIMERS
03570 400C BD 4C27                JSR    CLTM
03580                     * CLEAR FLAGS
03590 400F BD 495C                JSR    CLFG
03600                     * SET DISPLAY INTERVALS
03610 4012 CE 0014                LDX    #20
03620 4015 DF 05                  STX    TIM1
03630 4017 CE 00C8                LDX    #200
03640 401A DF 08                  STX    TIM2
03650 401C 7C 0004                INC    TF1
03660 401F 7C 0007                INC    TF2
03670 4022 0E                     CLI
03680                     * TEST IDLE BREAKOUT
03690 4023 7F 0042 AG2            CLR    AZDAF
03700 4026 7F 0041                CLR    ELDAF
03710                     *
03720 4029 7F 001E                CLR    EBLK
03730 402C 7F 001D                CLR    T3LK
03740 402F 7F 0059                CLR    OPCNT
03745 4032 7F 005A                CLR    OPCNT+1
03750 4035 7F 005B                CLR    TRCNT
03755 4038 7F 005C                CLR    TRCNT+1
03760 403B 7F 005F                CLR    XTIME
03770 403E 7F 001C                CLR    STF
03780 4041 7F 001F                CLR    ELLK1
03790 4044 7F 0020                CLR    AZLK1
03800 4047 7F 0021                CLR    ELLK2
03810 404A 7F 0022                CLR    AZLK2
03820 404D 7F 0023                CLR    DEL4
03830 4050 7F 0046                CLR    PASSAZ
03840 4053 7F 0029                CLR    SLOWF
03850 4056 7F 0016                CLR    WPNF
03860                     *
03870 4059 7F 001B                CLR    RUCWF
03880 405C 7F 0024                CLR    RUCCWF
03890                     *
03900 405F 7D 001A                TST    IDLEF
03910 4062 27 2C                  BEQ    AG46
03920                     * COME FROM LOAD POS?
03930 4064 7D 0018                TST    LPOSF
03940 4067 26 0D                  BNE    AG5
03950 4069 7D 0013                TST    OLP
03960 406C 27 10                  BEQ    AG6
```

```
406E 7F 0013          CLR     OLP
4071 7C 0016          INC     WPNF
4074 20 0F            BRA     AG28
                 *
4076 7F 0013 AG5      CLR     OLP
4079 7C 0013          INC     OLP
407C 20 12            BRA     AG46
                 * RESET ON SERVO SW.
407E 86 80   AG6      LDA A   #SRVOM
4080 BD 44C5          JSR     TSTS8B
4083 25 0B            BCS     AG46
4085 7F 001A AG28     CLR     IDLEF
4088 7F 0017          CLR     XRECF
408B 86 FF            LDA A   #$FF
408D B7 240D          STA A   PIA4DB
                 * READ CONFIG SWITCHES
4090 BD 4390 AG46     JSR     CONRQ
                 * FORM CONFIG WORD
4093 BD 431F          JSR     OCE
                 * READ AZ AND EL TRIM
4096 BD 43A3 AG7      JSR     RTRM
                 * READ AZ AND EL GACS
4099 7F 002B          CLR     DISEL
409C 86 80            LDA A   #SRVOM
409E BD 44C5          JSR     TSTS8B
40A1 25 03            BCS     AG8
40A3 BD 4419          JSR     RGACS
                 * SWITCH GAX SELECT
40A6 86 36   AG8      LDA A   #$36
40A8 C6 3E            LDA B   #$3E
40AA B7 2402          STA A   GCSEL
40AD F7 2403          STA B   GCSAZ
40B0 7D 0019          TST     XTHRU
40B3 27 06            BEQ     AG44
40B5 B7 2403          STA A   GCSAZ
40B8 F7 2402          STA B   GCSEL
                 * TEST LOAD POS.
40BB 86 40   AG44     LDA A   #LPOSM
40BD BD 44CE          JSR     TSTS8A
40C0 25 05            BCS     AG38
40C2 7F 0018          CLR     LPOSF
40C5 20 07            BRA     AG9
                 * SET LOAD POSITION
40C7 86 01   AG38     LDA A   #1
40C9 97 18            STA A   LPOSF
40CB BD 4513          JSR     SLPOS
                 *
                 * READ AZ AND EL ENCODERS
40CE 7C 002B AG9      INC     DISEL
40D1 BD 43D8          JSR     RENCS
                 * TEST XTHRU FLAG
                 * =1 EL COMPUTE,=0 AZ COMPUTE
                 *
40D4 7D 0019          TST     XTHRU
```



```
40D7 26 08            BNE     AG10
40D9 BD 4650          JSR     COMPAZ
40DC 7C 0019          INC     XTHRU
40DF 20 06            BRA     AG11
                 *
40E1 BD 46E4 AG10     JSR     COMPEL
40E4 7F 0019          CLR     XTHRU
                 * READ AND STORE ERROR VOLTAGES
40E7 BD 4482 AG11     JSR     RAEV
                 * CHECK OFFSETS
40EA BD 4579          JSR     CKO
                 * TEST SERVO SW
40ED 86 80            LDA A   #SRVOM
40EF BD 44C5          JSR     TSTS8B
40F2 25 08            BCS     AG47
40F4 86 FF            LDA A   #$FF
40F6 B7 240D          STA A   PIA4DB
40F9 7E 4023          JMP     AG2
                 * TEST IDLE FLAG
40FC 7D 001A AG47     TST     IDLEF
40FF 27 03            BEQ     AG39
4101 7E 4023          JMP     AG2
                 * SET RU FLAGS
4104 BD 44D7 AG39     JSR     SRUF
                 * WEAPON AZ ENABLED?
4107 7F 0042          CLR     AZDAF
410A 7C 0042          INC     AZDAF
410D 7D 0014          TST     AZGOF
4110 27 35            BEQ     AG12
                 * TEST START THRU FLAG
4112 7D 001C          TST     STF
4115 27 03            BEQ     AG37
4117 7E 41BC          JMP     AG16
                 * TEST XENON ON
411A 7D 0017 AG37     TST     XRECF
411D 26 03            BNE     AG32
411F 7E 4176          JMP     AG4
                 * ENABLE PANTEL AZ
4122 86 FB   AG32     LDA A   #PAGO
4124 BD 439E          JSR     FIXENB
                 * TEST READY
4127 7D 0025          TST     AREDY
412A 27 03            BEQ     AG41
412C 7E 41AC          JMP     AG3
                 * AZ D/A OUTPUT=0
412F CE 0000 AG41     LDX     #0
4132 DF 4F            STX     AZCOM
                 * DISABLE PANTEL OFFSET
4134 86 EF            LDA A   #POGO
4136 BD 4548          JSR     FIXDIS
                 * DISABLE RU SEARCH FLAGS
4139 7F 001B          CLR     RUCWF
413C 7F 0024          CLR     RUCCWF
                 * SET READY
```

```
05040 413F 7F 0025          CLR     AREDY
05050 4142 7C 0025          INC     AREDY
05060 4145 20 65            BRA     AG3
05070                 * WEAPON AZ DISABLED
05080                 * TEST PANTEL OFFSET
05090 4147 86 08      AG12  LDA A   #PORQM
05100 4149 BD 4396          JSR     TSTSW
05110 414C 24 1E            BCC     AG13
05120                 * ENABLE PANTEL OFFSET
05130 414E 86 EF            LDA A   #POGO
05140 4150 BD 439E          JSR     FIXENB
05150                 * TEST AZ LOCK
05160 4153 7D 0020          TST     AZLKI
05170 4156 26 64            BNE     AG16
05180                 * ENABLE PANTEL AZ
05190 4158 86 FB            LDA A   #PAGO
05200 415A BD 439E          JSR     FIXENB
05210                 * TEST AZ CLOSING
05220 415D BD 4C62          JSR     CLAZ
05230 4160 24 5A            BCC     AG16
05240                 * DISABLE PANTEL AZ
05250 4162 86 FB            LDA A   #PAGO
05260 4164 BD 4548          JSR     FIXDIS
05270 4167 7C 0020          INC     AZLKI
05280 416A 20 50            BRA     AG16
05290                 * AZ D/A OUTPUT=0
05300 416C CE 0000    AG13  LDX     #0
05310 416F DF 4F            STX     AZCOM
05320 4171 7F 0020          CLR     AZLKI
05330 4174 20 46            BRA     AG16
05340 4176 7F 0025    AG4   CLR     AREDY
05350                 * RU SEARCH CW?
05360 4179 96 1B            LDA A   RUCWF
05370 417B 27 07            BEQ     AG14
05380 417D CE 0A00    AG33  LDX     #P100M
05390 4180 DF 4F            STX     AZCOM
05400 4182 20 09            BRA     AG15
05410                 * RU SEARCH CCW?
05420 4184 96 24      AG14  LDA A   RUCCWF
05430 4186 27 17            BEQ     AG20
05440 4188 CE 8A00          LDX     #M100M
05450 418B DF 4F            STX     AZCOM
05460                 * ENABLE PANTEL OFFSET
05470 418D 86 EF      AG15  LDA A   #POGO
05480 418F BD 439E          JSR     FIXENB
05490                 * ENABLE PANTEL AZ
05500 4192 86 FB            LDA A   #PAGO
05510 4194 BD 439E          JSR     FIXENB
05520                 * SET TRAV BLOCK
05530 4197 7F 001D          CLR     TBLK
05540 419A 7C 001D          INC     TBLK
05550 419D 20 1D            BRA     AG16
05560                 * DISABLE PANTEL AZIMUTH
05570 419F 86 FB      AG20  LDA A   #PAGO
```



```
05580 41A1 BD 4548          JSR     FIXDIS
05590 41A4 7F 001D          CLR     TBLK
05600 41A7 7C 001D          INC     TBLK
05610 41AA 20 10            BRA     AG16
05620                 * TEST XENON STABILITY
05630 41AC BD 4DB7    AG3   JSR     XSTAB
05640 41AF 24 05            BCC     AG36
05650                 * CLEAR TRAV BLOCK
05660 41B1 7F 001D          CLR     TBLK
05670 41B4 20 06            BRA     AG16
05680 41B6 7F 001D    AG36  CLR     TBLK
05690 41B9 7C 001D          INC     TBLK
05700 41BC 7F 0041    AG16  CLR     ELDAF
05710 41BF 7C 0041          INC     ELDAF
05720                 * AUTO EL SELECTED?
05730 41C2 7D 0015          TST     ELGOF
05740 41C5 27 1A            BEQ     AG23
05750                 * START THRU SET?
05760 41C7 7D 001C          TST     STF
05770 41CA 26 44            BNE     AG21
05780                 * TEST QUAD PITCH CLOSING
05790 41CC 7F 001E          CLR     EBLK
05800 41CF 7C 001E          INC     EBLK
05810 41D2 CE 000A          LDX     #10
05820 41D5 DF 60            STX     QPLP
05830 41D7 BD 4D42          JSR     CLQP
05840 41DA 24 34            BCC     AG21
05850                 * NULL ACHIEVED
05860 41DC 7F 001E          CLR     EBLK
05870 41DF 20 2F            BRA     AG21
05880                 * TEST QUAD OFFSET SELECT
05890 41E1 86 04      AG23  LDA A   #QORQM
05900 41E3 BD 4396          JSR     TSTSW
05910 41E6 25 0A            BCS     AG50
05920 41E8 CE 0000          LDX     #0
05930 41EB DF 51            STX     ELCOM
05940 41ED 7F 001F          CLR     ELLKI
05950 41F0 20 1E            BRA     AG21
05960                 * ENABLE QUAD OFFSET
05970 41F2 86 DF      AG50  LDA A   #QOGO
05980 41F4 BD 439E          JSR     FIXENB
05990                 * TEST EL LOCK
06000 41F7 7D 001F          TST     ELLKI
06010 41FA 26 14            BNE     AG21
06020                 * ENABLE QUAD PITCH
06030 41FC 86 F7            LDA A   #QPGO
06040 41FE BD 439E          JSR     FIXENB
06050                 * TEST EL CLOSING
06060 4201 BD 4CF7          JSR     CLEL
06070 4204 25 02            BCS     AG24
06080 4206 20 08            BRA     AG21
06090                 * DISABLE QUAD PITCH
06100 4208 86 F7      AG24  LDA A   #QPGO
06110 420A BD 4548          JSR     FIXDIS
```

```
420D 7C 001F              INC    ELLK1
                 * OUTPUT CONFIG ENABLES
4210 96 2D      AG21    LDA A  CONGO
4212 B7 240D            STA A  PIA4DB
                 * START THRU SET?
4215 7D 001C            TST    STF
4218 26 0B              BNE    AG25
                 * TEST TRAV OR EL BLOCK
421A BD 40A9            JSR    ORBLK
421D 24 06              BCC    AG25
421F 7F 0016            CLR    WPNF
4222 7E 4096            JMP    AG7
                 * TEST WEAPON SW
4225 7F 001C AG25       CLR    STF
4228 7D 0016            TST    WPNF
422B 26 03              BNE    AG27
422D 7E 4096            JMP    AG7
                 * SET START THRU
4230 7C 001C AG27       INC    STF
                 * AZ ENABLED?
4233 7F 001A AG17       CLR    IDLEF
4236 7D 0014            TST    AZGOF
4239 27 50              BEQ    AG26
                 * TEST XENON ON
423B 7D 0017            TST    XRECF
423E 26 06              BNE    AG40
4240 7F 0025            CLR    AREDY
4243 7F 0016            CLR    WPNF
                 * TEST READY
4246 7D 0025 AG40       TST    AREDY
4249 26 07              BNE    AG19
                 * DISABLE PANTEL AZ
424B 86 FB              LDA A  #PAGO
424D BD 4548            JSR    FIXDIS
4250 20 34              BRA    AG45
                 * ENABLE WEAPON AZ
4252 86 BF      AG19    LDA A  #AZGO
4254 BD 439E            JSR    FIXENB
                 * TEST AZ LOCK
4257 7D 0022            TST    AZLK2
425A 26 20              BNE    AG51
                 * ENABLE PANTEL AZ
425C 86 FB              LDA A  #PAGO
425E BD 439E            JSR    FIXENB
                 * ENABLE PANTEL OFFSET
4261 86 EF              LDA A  #POGO
4263 BD 439E            JSR    FIXENB
                 * TEST AZIMUTH CLOSING
4266 7F 0029            CLR    SLOWF
4269 7C 0029            INC    SLOWF
426C BD 4C62            JSR    CLAZ
426F 24 1A              BCC    AG26
                 * DISABLE PANTEL AZ
4271 7F 0029            CLR    SLOWF
```



```
4274 86 FB              LDA A  #PAGO
4276 BD 4548            JSR    FIXDIS
4279 7C 0022            INC    AZLK2
                 * TEST TRACKER NULL
427C CE 07D0 AG51       LDX    #2000
427F DF 5D              STX    TRLP
4281 BD 4D26            JSR    CLTR
4284 24 05              BCC    AG26
                 * DISABLE AZ
4286 86 BF      AG45    LDA A  #AZGO
4288 BD 4548            JSR    FIXDIS
                 * EL ENABLED?
428B 7D 0015 AG26       TST    ELGOF
428E 27 5B              BEQ    AG30
                 * TEST WEAPON FLAG
4290 7D 0016            TST    WPNF
4293 27 0A              BEQ    AG29
                 * TEST LOAD POSITION
4295 7D 0018 AG22       TST    LPOSF
4298 27 0C              BEQ    AG43
                 * TEST ELEVATION ERROR(D/A FORMAT)
429A 7D 0055            TST    ELERR
429D 2B 07              BMI    AG43
                 * DISABLE QUAD PITCH
429F 86 F7      AG29    LDA A  #QPGO
42A1 BD 4548            JSR    FIXDIS
42A4 20 40              BRA    AG48
                 * TEST EL  LOCK
42A6 7D 0021 AG43       TST    ELLK2
42A9 26 31              BNE    AG52
                 * ENABLE QUAD PITCH
42AB 86 F7              LDA A  #QPGO
42AD BD 439E            JSR    FIXENB
                 * ENABLE QUAD OFFSET
42B0 86 DF              LDA A  #QOGO
42B2 BD 439E            JSR    FIXENB
                 * DELAY WEAPON EL
42B5 7D 0023            TST    DEL4
42B8 26 0B              BNE    AG54
42BA CE 03E8            LDX    #1000
42BD DF 0E              STX    TIM4
42BF 7C 000D            INC    TF4
42C2 7C 0023            INC    DEL4
                 *
42C5 7D 000D AG54       TST    TF4
42C8 26 05              BNE    AG53
                 * ENABLE WEAPON EL
42CA 86 7F              LDA A  #ELGO
42CC BD 439E            JSR    FIXENB
                 * TEST EL CLOSING
42CF BD 4CF7 AG53       JSR    CLEL
42D2 24 17              BCC    AG30
                 * DISABLE QUAD PITCH
42D4 86 F7              LDA A  #QPGO
```

PAGE 015 AGLS

```
07200 42D6 BD 4548          JSR     FIXDIS
07210 42D9 7C 0021          INC     ELLK2
07220                 * TEST QUAD PITCH
07230 42DC CE 07D0  AG52    LDX     #2000
07240 42DF DF 60            STX     QPLP
07250 42E1 BD 4D42          JSR     CLQP
07260 42E4 24 05            BCC     AG30
07270                 * DISABLE ELEVATION
07280 42E6 86 7F    AG48    LDA A   #ELGO
07290 42E8 BD 4548          JSR     FIXDIS
07300                 * TEST ENABLES
07310 42EB 5F       AG30    CLR B
07320 42EC 86 BF            LDA A   #AZGO
07330 42EE 43               COM A
07340 42EF 94 2D            AND A   CONGO
07350 42F1 26 02            BNE     AG31
07360 42F3 C6 01            LDA B   #1
07370 42F5 D7 42    AG31    STA B   AZDAF
07380 42F7 5F               CLR B
07390 42F8 86 7F            LDA A   #ELGO
07400 42FA 43               COM A
07410 42FB 94 2D            AND A   CONGO
07420 42FD 26 06            BNE     AG34
07430 42FF C6 01            LDA B   #1
07440 4301 D7 41    AG49    STA B   ELDAF
07450 4303 20 12            BRA     AG35
07460                 *
07470 4305 D7 41    AG34    STA B   ELDAF
07480 4307 7D 000D          TST     TF4
07490 430A 26 08            BNE     AG35
07500 430C 7D 0042          TST     AZDAF
07510 430F 2E 06            BGT     AG35
07520 4311 7F 0016          CLR     WPNF
07530 4314 7C 001A          INC     IDLEF
07540                 * OUTPUT ENABLES AND DRIVE
07550 4317 96 2D    AG35    LDA A   CONGO
07560 4319 B7 240D          STA A   PIA4DB
07570 431C 7E 4096          JMP     AG7
```

PAGE 016 AGLS

```
00010                 *
00020                 **********AGLS2*******
00030                 *
00040                 *
00050                 * FORM CONFIGURATION WORD
00060                 *
00070 431F 86 FF    OCE     LDA A   #$FF
00080 4321 97 2D            STA A   CONGO
00100                 * TEST AUTO AZ
00110 4323 86 02            LDA A   #AZROM
00120 4325 BD 4396          JSR     TSTSW
00130 4328 24 0D            BCC     OCE1
00140 432A 86 FE            LDA A   #PLGO
00150 432C BD 439E          JSR     FIXENB
00160 432F 7F 0014          CLR     AZGOF
00170 4332 7C 0014          INC     AZGOF
00180 4335 20 1B            BRA     OCE3
00190                 * TEST PANTEL LEVEL
00200 4337 7F 0014  OCE1    CLR     AZGOF
00210 433A 86 20            LDA A   #PLROM
00220 433C BD 4396          JSR     TSTSW
00230 433F 24 05            BCC     OCE2
00240 4341 86 FE            LDA A   #PLGO
00250 4343 BD 439E          JSR     FIXENB
00260                 * TEST PANTEL OFFSET
00270 4346 86 08    OCE2    LDA A   #POROM
00280 4348 BD 4396          JSR     TSTSW
00290 434B 24 05            BCC     OCE3
00300 434D 86 EF            LDA A   #POGO
00310 434F BD 439E          JSR     FIXENB
00320                 * TEST AUTO EL
00330 4352 86 01    OCE3    LDA A   #ELROM
00340 4354 BD 4396          JSR     TSTSW
00350 4357 25 26            BCS     OCE6
00360 4359 7F 0015          CLR     ELGOF
00370                 * TEST QUAD LEVEL
00380 435C 86 10            LDA A   #QLROM
00390 435E BD 4396          JSR     TSTSW
00400 4361 24 0A            BCC     OCE4
00410 4363 86 FD            LDA A   #QCGO
00420 4365 BD 439E          JSR     FIXENB
00430 4368 86 F7            LDA A   #QPGO
00440 436A BD 439E          JSR     FIXENB
00450                 * TEST QUAD OFFSET
00460 436D 86 04    OCE4    LDA A   #QOROM
00470 436F BD 4396          JSR     TSTSW
00480 4372 24 0A            BCC     OCE5
00490 4374 86 F7            LDA A   #QPGO
00500 4376 BD 439E          JSR     FIXENB
00510 4379 86 DF            LDA A   #QOGO
00520 437B BD 439E          JSR     FIXENB
00530 437E 39       OCE5    RTS
00540                 * ENABLE AUTO EL
00550 437F 7F 0015  OCE6    CLR     ELGOF
```

```
4382 7C 0015          INC     ELGOF
4385 86 FD            LDA A   #QCGO
4387 BD 439E          JSR     FIXENB
438A 86 F7            LDA A   #QPGO
438C BD 439E          JSR     FIXENB
438F 39               RTS
                *
                * READ CONFIGURATION SWITCH REGISTER
                *
4390 B6 2800    CONRQ LDA A   PIAODA
4393 97 2E            STA A   CONTEM
4395 39               RTS
                *
                * TEST CONFIG. SWITCH WORD
                * C=SET IF SW ON$C=0 IF OFF
                *
4396 94 2E      TSTSW AND A   CONTEM
4398 27 02            BEQ     TSTS1
439A 0C               CLC
439B 39               RTS
439C 0D         TSTS1 SEC
439D 39               RTS
                *
                * FIX ENABLE WORD
                *
439E 94 2D      FIXENB AND A  CONGO
43A0 97 2D            STA A   CONGO
43A2 39               RTS
                *
                * READ TRIM ROUTINE
                * AZIMUTH
43A3 B6 2415    RTRM  LDA A   PIA6DB
43A6 CE 2417          LDX     #SAD6B
43A9 BD 43C6          JSR     SCON
43AC CE 2415          LDX     #PIA6DB
43AF BD 43CF          JSR     GET
43B2 97 2F            STA A   AZTRM
                *
                * ELEVATION
43B4 B6 2414          LDA A   PIA6DA
43B7 CE 2416          LDX     #SAD6A
43BA BD 43C6          JSR     SCON
43BD CE 2414          LDX     #PIA6DA
43C0 BD 43CF          JSR     GET
43C3 97 30            STA A   ELTRM
43C5 39               RTS
                *
                * STROBE CONTROL PULSE(B REG)
                *
43C6 C6 3E      SCON  LDA B   #$3E
43C8 E7 00            STA B   0,X
43CA C6 36            LDA B   #$36
43CC E7 00            STA B   0,X
43CE 39               RTS
```



```
                *
                * GET A/D DATA ROUTINE
                * X=DATA REG. ADDRESS
                *
43CF A6 02      GET   LDA A   2,X
43D1 01               NOP
43D2 01               NOP
43D3 A6 00            LDA A   0,X
43D5 80 7F            SUB A   #$7F
43D7 39               RTS
                *
                * READ ENCODERS ROUTINE
                * ELEVATION AXIS
43D8 CE 00BA    RENCS LDX     #ELTEMP
43DB B6 2405          LDA A   PIA2DB
43DE BD 4403          JSR     STBF
43E1 B6 2404          LDA A   PIA2DA
43E4 BD 4403          JSR     STBF
43E7 F6 240C          LDA B   PIA4DA
43EA BD 440D          JSR     STBF1
                *
                * AZIMUTH AXIS
43ED CE 00BF          LDX     #AZTEMP
43F0 B6 2409          LDA A   PIA3DB
43F3 BD 4403          JSR     STBF
43F6 B6 2408          LDA A   PIA3DA
43F9 BD 4403          JSR     STBF
43FC B6 240C          LDA A   PIA4DA
43FF BD 4412          JSR     STBF2
4402 39               RTS
                *
                * STORE DISPLAY BUFFER ROUTINE
                *
4403 16         STBF  TAB
4404 BD 4412          JSR     STBF2
4407 08               INX
4408 BD 440D          JSR     STBF1
440B 08               INX
440C 39               RTS
                *
                * STORE DISPLAY BUFFER-1
440D C4 0F      STBF1 AND B   #$0F
440F E7 00            STA B   0,X
4411 39               RTS
                *
                * STORE DISPLAY BUFFER-2
4412 44         STBF2 LSR A
4413 44               LSR A
4414 44               LSR A
4415 44               LSR A
4416 A7 00            STA A   0,X
4418 39               RTS
                *
                * READ GACS ROUTINE
```

```
                        *
4419 01         RGACS   NOP
                        * ELEVATION
441A BD 45E3            JSR     RELGAC
                        * AZIMUTH
441D B6 2403            LDA A   GCSAZ
4420 84 3F              AND A   #$3F
4422 81 36              CMP A   #$36
4424 27 01              BEQ     RGAC1
4426 39                 RTS
4427 B6 2401    RGAC1   LDA A   PIA1DB
442A 43                 COM A
442B CE 0603            LDX     #$0603
442E DF 81              STX     THI
4430 BD 445B            JSR     TVAL
4433 25 16              BCS     RGAC2
4435 97 87              STA A   GACTEM
                        *
4437 B6 2400            LDA A   PIA1DA
443A 43                 COM A
443B CE 0909            LDX     #$0909
443E DF 81              STX     THI
4440 BD 445B            JSR     TVAL
4443 25 06              BCS     RGAC2
4445 97 88              STA A   GACTEM+1
                        * FIX DISPLAY BUFFER
                        * AZIMUTH
4447 DE 87              LDX     GACTEM
4449 DF 33              STX     AZGCS
444B CE 009A    RGAC2   LDX     #AZGCDS
444E 96 33              LDA A   AZGCS
4450 BD 4403            JSR     STBF
4453 96 34              LDA A   AZGCS+1
4455 BD 4403            JSR     STBF
4458 6F 00              CLR     0,X
445A 39                 RTS
                        *
                        * TEST GACS VALUES
                        *
445B 7F 0086    TVAL    CLR     TF
445E 16                 TAB
445F C4 0F              AND B   #$F
4461 C1 09              CMP B   #9
4463 2F 02              BLE     TVAL3
4465 0D                 SEC
4466 39                 RTS
4467 D1 82      TVAL3   CMP B   TLO
4469 2F 03              BLE     TVAL1
446B 7C 0086            INC     TF
                        *
446E 16         TVAL1   TAB
446F 54                 LSR B
4470 54                 LSR B
4471 54                 LSR B
```



```
4472 54                 LSR B
4473 D1 81              CMP B   THI
4475 2D 09              BLT     TVAL2
4477 2E 05              BGT     TVAL4
4479 7D 0086            TST     TF
447C 27 02              BEQ     TVAL2
447E 0D         TVAL4   SEC
447F 39                 RTS
                        *
4480 0C         TVAL2   CLC
4481 39                 RTS
                        *
                        * READ ANALOG ERROR VOLTAGES
                        *
4482 4F         RAEV    CLR A
4483 97 3C              STA A   MUXADD
4485 C6 05              LDA B   #5
4487 CE 0035            LDX     #ERRBUF
                        * SETUP FOR REPEATED TRY'S
448A 97 3E      RAEV5   STA A   PREVAL
448C 86 05              LDA A   #5
448E 97 3D              STA A   NUMRED
                        * LOOP ON A/D CHANNELS
4490 86 34      RAEV2   LDA A   #$34
4492 B7 2412            STA A   SAD5
4495 96 3C              LDA A   MUXADD
4497 B7 2411            STA A   PIA5DB
                        * 100 USEC DELAY
449A 86 10              LDA A   #16
449C 4A         RAEV4   DEC A
449D 26 FD              BNE     RAEV4
449F 86 3C              LDA A   #$3C
44A1 B7 2412            STA A   SAD5
                        * WAIT EOC
44A4 86 20              LDA A   #32
44A6 4A         RAEV3   DEC A
44A7 26 FD              BNE     RAEV3
                        * READ AND STORE DATA
44A9 B6 2410            LDA A   PIA5DA
                        * TEST FOR CONSEC. READINGS
44AC 91 3E              CMP A   PREVAL
44AE 26 DA              BNE     RAEV5
44B0 7A 003D            DEC     NUMRED
44B3 26 DB              BNE     RAEV2
44B5 A7 00              STA A   0,X
44B7 08                 INX
44B8 96 3C              LDA A   MUXADD
44BA 8B 10              ADD A   #$10
44BC 97 3C              STA A   MUXADD
44BE 5A                 DEC B
44BF 26 C9              BNE     RAEV5
44C1 7F 2411            CLR     PIA5DB
44C4 39                 RTS
                        *
```

```
                  * TEST PIA8 SWITCHES (B SIDE)
                  * C SET IF SW ON:C=0 IF OFF
                  *
44C5 B4 241D TSTS8B AND A  PIA8DB
44C8 27 02          BEQ    TST81
44CA 0C             CLC
44CB 39             RTS
44CC 0D      TST81  SEC
44CD 39             RTS
                  *
                  * TEST PIA8 SWITCHES (A SIDE)
                  * C SET IF SW ON:C=0 IF OFF
                  *
44CE B4 241C TSTS8A AND A  PIA8DA
44D1 27 02          BEQ    TST82
44D3 0C             CLC
44D4 39             RTS
44D5 0D      TST82  SEC
44D6 39             RTS
                  *
                  * SET RU FLAGS
                  *
44D7 7D 001B SRUF   TST    RUCWF
44DA 26 1A          BNE    SRUF3
44DC 86 10          LDA A  #RUCWM
44DE BD 44CE        JSR    TSTS8A
44E1 24 03          BCC    SRUF1
44E3 7C 001B        INC    RUCWF
                  *
44E6 7D 0024 SRUF1  TST    RUCCWF
44E9 26 1F          BNE    SRUF4
44EB 86 20          LDA A  #RUCCWM
44ED BD 44CE        JSR    TSTS8A
44F0 24 03          BCC    SRUF2
44F2 7C 0024        INC    RUCCWF
44F5 39      SRUF2  RTS
44F6 7F 0024 SRUF3  CLR    RUCCWF
44F9 86 20          LDA A  #RUCCWM
44FB BD 44CE        JSR    TSTS8A
44FE 24 F5          BCC    SRUF2
4500 7F 001B        CLR    RUCWF
4503 7F 0024        CLR    RUCCWF
4506 7C 0024        INC    RUCCWF
4509 39             RTS
450A 7D 001B SRUF4  TST    RUCWF
450D 27 E6          BEQ    SRUF2
450F 7F 0024        CLR    RUCCWF
4512 39             RTS
                  *
                  * SET LOAD POSITION
                  *
4513 CE 0200 SLPOS  LDX    #LOAD1
4516 86 80          LDA A  #LDP1M
4518 BD 4396        JSR    TSTSW
```



```
451B 24 0F          BCC    SLP01
451D CE 0300        LDX    #LOAD2
                  *
4520 86 40          LDA A  #LDP2M
4522 BD 4396        JSR    TSTSW
4525 24 0F          BCC    SLP03
4527 CE 0500        LDX    #LOAD4
452A 20 0A          BRA    SLP03
                  *
452C 86 40   SLP01  LDA A  #LDP2M
452E BD 4396        JSR    TSTSW
4531 24 03          BCC    SLP03
4533 CE 0400        LDX    #LOAD3
                  *
4536 DF 31   SLP03  STX    ELGCS
                  * FIX DISPLAY BUFFER
4538 CE 008B        LDX    #ELGCDS
453B 96 31          LDA A  ELGCS
453D BD 4403        JSR    STBF
4540 96 32          LDA A  ELGCS+1
4542 BD 4403        JSR    STBF
4545 6F 00          CLR    0,X
4547 39             RTS
                  *
                  *
                  *
                  * FIX DISABLE ROUTINE
                  *
4548 43      FIXDIS COM A
4549 9A 2D          ORA A  CONGO
454B 97 2D          STA A  CONGO
454D 39             RTS
                  *
                  * TEST OFFSET ERRORS
                  * C SET=OFFSET>ALLOWED
                  * C=0 OFFSET OK
                  *
454E CE 0035 TERR   LDX    #ERRBUF
4551 5A             DEC B
4552 D7 83          STA B  HOLDB
4554 27 04          BEQ    TERR1
                  *
4556 08      TERR2  INX
4557 5A             DEC B
4558 26 FC          BNE    TERR2
455A A6 00   TERR1  LDA A  0,X
455C CE 4574        LDX    #ERRVAL
455F D6 83          LDA B  HOLDB
4561 27 04          BEQ    TERR4
4563 08      TERR3  INX
4564 5A             DEC B
4565 26 FC          BNE    TERR3
4567 E6 00   TERR4  LDA B  0,X
4569 4D             TST A
```

```
4566A 2A 01         BPL    TERR4
                 *
456C 40             NEG A
456D 11     TERR4   CBA
456E 2A 02          BPL    TERR5
4570 0C             CLC
4571 39             RTS
                 *
4572 0D     TERR5   SEC
4573 39             RTS
                 *
                 *
      4574  ERRVAL  EQU    *
4574 0A             FCB    EOP
4575 05             FCB    EOC
4576 05             FCB    EMP
4577 05             FCB    EMC
4578 01             FCB    EPA
                 *
                 * CHECK OFFSETS AND CONTROL GO/NO-GO
                 *
4579 7F 0027 CKO    CLR    CKOF
457C C6 05          LDA B  #5
457E D7 8A          STA B  SAVB
4580 7F 0040        CLR    EFLAG
4583 7F 00AA        CLR    AZLIT
4586 7F 00A9        CLR    ELLIT
4589 D6 8A  CK02    LDA B  SAVB
                 * CALL TEST ERROR
458B BD 454E        JSR    TERR
                 * SETUP ERROR WORD FOR DISPLAY
458E 24 37          BCC    CK08
4590 7C 0040        INC    EFLAG
                 *
4593 D6 8A          LDA B  SAVB
4595 C1 05          CMP B  #5
4597 26 06          BNE    CK04
4599 86 04          LDA A  #4
459B 9A AA          ORA A  AZLIT
459D 97 AA          STA A  AZLIT
                 *
459F C1 04  CK04    CMP B  #4
45A1 26 06          BNE    CK05
45A3 86 02          LDA A  #2
45A5 9A AA          ORA A  AZLIT
45A7 97 AA          STA A  AZLIT
                 *
45A9 C1 03  CK05    CMP B  #3
45AB 26 06          BNE    CK06
45AD 86 01          LDA A  #1
45AF 9A AA          ORA A  AZLIT
45B1 97 AA          STA A  AZLIT
                 *
45B3 C1 02  CK06    CMP B  #2
```



```
45B5 26 06          BNE    CK07
45B7 86 02          LDA A  #2
45B9 9A A9          ORA A  ELLIT
45BB 97 A9          STA A  ELLIT
                 *
45BD C1 01  CK07    CMP B  #1
45BF 26 06          BNE    CK08
45C1 86 01          LDA A  #1
45C3 9A A9          ORA A  ELLIT
45C5 97 A9          STA A  ELLIT
                 *
45C7 7A 008A CK08   DEC    SAVB
45CA 26 BD          BNE    CK02
                 * TEST GO/NO-GO
45CC 7D 0040        TST    EFLAG
45CF 26 09          BNE    CK01
                 * ENABLE GO
45D1 86 36          LDA A  #$36
45D3 B7 241F        STA A  LAMP
45D6 7C 0027        INC    CKOF
45D9 39             RTS
                 * ENABLE NO-GO
45DA 86 3E  CK01    LDA A  #$3E
45DC B7 241F        STA A  LAMP
45DF 7C 0027        INC    CKOF
45E2 39             RTS
                 *
                 * READ ELEVATION GACS
                 *
45E3 B6 2402 RELGAC LDA A  GCSEL
45E6 84 3F          AND A  #$3F
45E8 81 36          CMP A  #$36
45EA 27 01          BEQ    REL1
45EC 39             RTS
                 * READ GACS
45ED B6 2401 REL1   LDA A  PIAIDB
45F0 43             COM A
45F1 CE 0103        LDX    #$0103
45F4 DF 81          STX    THI
45F6 BD 445B        JSR    TVAL
45F9 25 16          BCS    REL2
45FB 97 87          STA A  GACTEM
                 *
45FD B6 2400        LDA A  PIAIDA
4600 43             COM A
4601 CE 0909        LDX    #$0909
4604 DF 81          STX    THI
4606 BD 445B        JSR    TVAL
4609 25 06          BCS    REL2
460B 97 88          STA A  GACTEM+1
                 * FIX DISPLAY BUFFER
460D DE 87          LDX    GACTEM
460F DF 31          STX    ELGCS
4611 CE 008B REL2   LDX    #ELGCDS
```

```
4614 96 31          LDA A  ELGCS
4616 BD 4403        JSR    STBF
4619 96 32          LDA A  ELGCS+1
461B BD 4403        JSR    STBF
461E 6F 00          CLR    0,X
4620 39             RTS
             *
             * TEST AZ ERR RANGE
             * (D/A FORMAT)
             *
4621 DE 53   TAZERR LDX    AZERR
4623 C6 01          LDA B  #AZLIM
4625 BD 4631        JSR    TSTIT
4628 39             RTS
             *
             * TEST EL ERR RANGE
             *
4629 DE 55   TELERR LDX    ELERR
462B C6 01          LDA B  #ELLIM
462D BD 4631        JSR    TSTIT
4630 39             RTS
             *
             *
             * TEST D/A FORMAT
             *
4631 DF 89   TSTIT  STX    SAVA
             * FIX FOR SIGN
4633 7D 0089        TST    SAVA
4636 2A 06          BPL    TSTI3
4638 96 89          LDA A  SAVA
463A 84 7F          AND A  #$7F
463C 97 89          STA A  SAVA
             *
463E 96 89   TSTI3  LDA A  SAVA
4640 84 0F          AND A  #$F
4642 27 02          BEQ    TSTI1
4644 0D             SEC
4645 39             RTS
             * CHECK LSB
4646 96 8A   TSTI1  LDA A  SAVB
4648 11             CBA
4649 01             NOP
464A 22 02          BHI    TSTI2
464C 0C             CLC
464D 39             RTS
464E 0D      TSTI2  SEC
464F 39             RTS
             *
             * COMPUTE AZ ERROR
             *
4650 7F 0026 COMPAZ CLR    SIG
4653 7F 002C        CLR    DISAZ
4656 CE 00B0        LDX    #TEMBCD
             * CONVERT TRIM TO BCD
```



```
4659 4F             CLR A
465A D6 2F          LDA B  AZTRM
465C 2A 04          BPL    COPAZ1
465E 7C 0026        INC    SIG
4661 50             NEG B
4662 01      COPAZ1 NOP
4663 01             NOP
4664 BD 476E        JSR    BINBCD
             * ADD TRIM TO ENCODER READING
4667 CE 00B0        LDX    #TEMBCD
466A DF 62          STX    A1
466C CE 00BF        LDX    #AZTEMP
466F DF 64          STX    A2
4671 CE 00B9        LDX    #RESULT+4
4674 BD 47FE        JSR    BCDADD
             * ADJUST FOR ROLLOVER
4677 CE 009F        LDX    #AZDISP
467A BD 48A3        JSR    ADJ
             * SUBTRACT RESULT FROM GACS
             *
467D CE 009F        LDX    #AZDISP
4680 DF 68          STX    S2
4682 CE 009A        LDX    #AZGCDS
4685 DF 66          STX    S1
4687 CE 00B5        LDX    #RESULT
468A BD 4855        JSR    BCDSUB
             * ADJUST FOR ROLLOVER
468D CE 00A4        LDX    #AZERD
4690 BD 48A3        JSR    ADJ
             * FIX FOR + OR - 32000
4693 7F 0026        CLR    SIG
4696 CE 00A4        LDX    #AZERD
4699 BD 48CD        JSR    TEST32
469C 24 1F          BCC    COPAZ3
             * FIX IF > 32000
469E CE 00A4        LDX    #AZERD
46A1 DF 62          STX    A1
46A3 BD 47D0        JSR    NINCOM
46A6 CE 46DA        LDX    #CONST
46A9 DF 64          STX    A2
46AB CE 00B9        LDX    #RESULT+4
46AE BD 47FE        JSR    BCDADD
             *
46B1 CE 00A4        LDX    #AZERD
46B4 BD 4902        JSR    XFER
             *
46B7 7F 0026        CLR    SIG
46BA 7C 0026        INC    SIG
             * TEST MAGNITUDE OF DISPLAY VALUE
46BD CE 00A4 COPAZ3 LDX    #AZERD
46C0 BD 48DF        JSR    TESTM
             * FIX SIGN OF DISPLAY
46C3 86 02          LDA A  #2
46C5 7D 0026        TST    SIG
```

```
46C8 27 02            BEQ    COPAZ2
46CA 86 01            LDA A  #1
46CC 97 A4     COPAZ2 STA A  AZERD
               * CONVERT ERROR TO BINARY
46CE CE 00A4          LDX    #AZERD
46D1 BD 491D          JSR    BCDBIN
46D4 DF 53            STX    AZERR
46D6 7C 002C          INC    DISAZ
46D9 39               RTS
46DA 06        CONST  FCB    6,4,0,0,0
46DB 04
46DC 00
46DD 00
46DE 00
46DF 03        CONST2 FCB    3,6,0,0,0
46E0 06
46E1 00
46E2 00
46E3 00
               *
               * COMPUTE ELEVATION ERROR
               *
46E4 7F 0026   COMPEL CLR    SIG
46E7 7F 002B          CLR    DISEL
46EA CE 0080          LDX    #TEMBCD
               * CONVERT TRIM TO BCD
46ED 4F               CLR A
46EE D6 30            LDA B  ELTRM
46F0 2A 04            BPL    COPEL1
46F2 7C 0026          INC    SIG
46F5 50               NEG B
46F6 01        COPEL1 NOP
46F7 01               NOP
46F8 BD 476E          JSR    BINBCD
               * ADD TRIM TO ENCODER READING
46FB CE 00B0          LDX    #TEMBCD
46FE DF 62            STX    A1
4700 CE 00BA          LDX    #ELTEMP
4703 DF 64            STX    A2
4705 CE 00B9          LDX    #RESULT+4
4708 BD 47FE          JSR    BCDADD
               * ADJUST FOR ROLLOVER
470B CE 0090          LDX    #ELDISP
470E BD 48A3          JSR    ADJ
               * SUBTRACT RESULT FROM GACS
4711 CE 0090          LDX    #ELDISP
4714 DF 68            STX    S2
4716 CE 008B          LDX    #ELGCDS
4719 DF 66            STX    S1
471B CE 00B5          LDX    #RESULT
471E BD 4855          JSR    BCDSUB
               * ADJUST FOR ROLLOVER
4721 CE 0095          LDX    #ELERD
4724 BD 48A3          JSR    ADJ
```



```
               * FIX FOR + OR - 32000
4727 7F 0026          CLR    SIG
472A CE 0095          LDX    #ELERD
472D BD 48CD          JSR    TEST32
4730 24 1F            BCC    COPEL3
               * FIX IF > 32000
4732 CE 0095          LDX    #ELERD
4735 DF 62            STX    A1
4737 BD 47D0          JSR    NINCOM
473A CE 46DA          LDX    #CONST
473D DF 64            STX    A2
473F CE 00B9          LDX    #RESULT+4
4742 BD 47FE          JSR    BCDADD
               *
4745 CE 0095          LDX    #ELERD
4748 BD 4902          JSR    XFER
               *
474B 7F 0026          CLR    SIG
474E 7C 0026          INC    SIG
               * TEST MAGNITUDE OF DISPLAY VALUE
4751 CE 0095   COPEL3 LDX    #ELERD
4754 BD 48DF          JSR    TESTM
               * FIX SIGN OF DISPLAY
4757 86 02            LDA A  #2
4759 7D 0026          TST    SIG
475C 27 02            BEQ    COPEL2
475E 86 01            LDA A  #1
4760 97 95     COPEL2 STA A  ELERD
               * CONVERT ERROR TO BINARY
4762 CE 0095          LDX    #ELERD
4765 BD 491D          JSR    BCDBIN
4768 DF 55            STX    ELERR
476A 7C 002B          INC    DISEL
476D 39               RTS
               *
               * BINARY-BCD CONVERSION
               * X=ADDRESS OF RESULT(5)
               * A,B= BINARY VALUE
               *
476E DF 70     BINBCD STX    HOLDX
4770 DF 6E            STX    OUTX
4772 DF 74            STX    STORX
4774 DF 62            STX    A1
               *
4776 CE 2710          LDX    #10000
4779 BD 47AE          JSR    TIZ
477C CE 03E8          LDX    #1000
477F BD 47AE          JSR    TIZ
4782 CE 0064          LDX    #100
4785 BD 47AE          JSR    TIZ
4788 CE 000A          LDX    #10
478B BD 47AE          JSR    TIZ
478E DE 6E            LDX    OUTX
4790 E7 00            STA B  0,X
```

```
                    * TEST SIGN FLAG
4792 7D 0026                TST     SIG
4795 26 01                  BNE     BIN1
4797 39                     RTS
                    * COMPLEMENT RESULT (MOD 64000)
4798 DE 70          BIN1    LDX     HOLDX
479A BD 47D0                JSR     NINCOM
479D CE 46DA                LDX     #CONST
47A0 DF 64                  STX     A2
47A2 CE 00B9                LDX     #RESULT+4
47A5 BD 47FE                JSR     BCDADD
47A8 DE 74                  LDX     STORX
47AA BD 4902                JSR     XFER
47AD 39                     RTS
                    *
                    * TEST NUMBER OF TIMES VALUE DIVISIBLE
                    *
47AE DF 72          TIZ     STX     TIZX
47B0 7F 007E                CLR     CNX
47B3 97 7F          TIZ2    STA A   OLDAX
47B5 D7 80                  STA B   OLDBX
                    * TRIAL SUBTRACT
47B7 D0 73                  SUB B   TIZX+1
47B9 92 72                  SBC A   TIZX
47BB 25 05                  BCS     TIZ1
47BD 7C 007E                INC     CNX
47C0 20 F1                  BRA     TIZ2
                    * FAIL SUBTRACT
47C2 DE 6E          TIZ1    LDX     OUTX
47C4 96 7E                  LDA A   CNX
47C6 A7 00                  STA A   0,X
47C8 08                     INX
47C9 DF 6E                  STX     OUTX
47CB 96 7F                  LDA A   OLDAX
47CD D6 80                  LDA B   OLDBX
47CF 39                     RTS
                    *
                    * NINES COMPLEMENT 5 DIGIT BCD #
                    * X= ADDRESS OF BCD MSB(BEFORE AND AFTER)
                    *
47D0 C6 04          NINCOM LDA B  #4
47D2 BD 47F9                JSR     FIXX
47D5 DF 84                  STX     KEEP
                    * COMPLEMENT EACH DIGIT
47D7 C6 05                  LDA B   #5
47D9 86 09          NIN1    LDA A   #09
47DB A0 00                  SUB A   0,X
47DD A7 00                  STA A   0,X
47DF 09                     DEX
47E0 5A                     DEC B
47E1 26 F6                  BNE     NIN1
                    * ADD ONE TO RESULT
47E3 DE 84                  LDX     KEEP
47E5 C6 05                  LDA B   #5
```



```
47E7 4F                     CLR A
47E8 0D                     SEC
                    *
47E9 A9 00          NIN2    ADC A   0,X
47EB BD 4848                JSR     JOCK
47EE 84 0F                  AND A   #$F
47F0 A7 00                  STA A   0,X
                    *
47F2 86 00                  LDA A   #0
47F4 09                     DEX
47F5 5A                     DEC B
47F6 26 F1                  BNE     NIN2
47F8 39                     RTS
                    *
                    * FIX X REG. POINTER
                    *
47F9 08             FIXX    INX
47FA 5A                     DEC B
47FB 26 FC                  BNE     FIXX
47FD 39                     RTS
                    *
                    * ADD 2-5 DIGIT BCD VALUES
                    * A1=ADDRESS OF VALUE 1 MSB
                    * A2=ADDRESS OF VALUE 2 MSB
                    * X=ADDRESS OF RESULT
                    *
47FE DF 76          BCDADD STX    ADDX1
4800 7F 0026                CLR     SIG
                    * FIX A1
4803 DE 62                  LDX     A1
4805 C6 04                  LDA B   #4
4807 BD 47F9                JSR     FIXX
480A DF 62                  STX     A1
                    * FIX A2
480C DE 64                  LDX     A2
480E C6 04                  LDA B   #4
4810 BD 47F9                JSR     FIXX
4813 DF 64                  STX     A2
4815 0C                     CLC
                    * GET FIRST VALUE
4816 C6 04                  LDA B   #4
4818 DE 62          BCA1    LDX     A1
481A A6 00                  LDA A   0,X
481C 09                     DEX
481D DF 62                  STX     A1
                    * ADD SECOND
481F DE 64                  LDX     A2
4821 A9 00                  ADC A   0,X
4823 BD 4848                JSR     JOCK
4826 09                     DEX
4827 DF 64                  STX     A2
                    * STORE IN OUTPUT
4829 DE 76                  LDX     ADDX1
482B A7 00                  STA A   0,X
```

```
482D 09                 DEX
482E DF 76              STX     ADDX1
                *
4830 5A                 DEC B
4831 26 E5              BNE     BCA1
4833 DE 62              LDX     A1
4835 A6 00              LDA A   0,X
4837 DE 64              LDX     A2
4839 A9 00              ADC A   0,X
483B BD 4848            JSR     JOCK
483E 24 03              BCC     BCA2
4840 7C 0026            INC     SIG
4843 DE 76      BCA2    LDX     ADDX1
4845 A7 00              STA A   0,X
4847 39                 RTS
4848 01         JOCK    NOP
4849 81 09              CMP A   #9
484B 2E 02              BGT     JOCK1
484D 0C                 CLC
484E 39                 RTS
484F 8B 06      JOCK1   ADD A   #6
4851 84 0F              AND A   #$F
4853 0D                 SEC
4854 39                 RTS
                *
                * SUBTRACT 2-5 DIGIT BCD VALUES
                * S1=ADDRESS OF MINUEND
                * S2=ADDRESS OF SUBTRAHEND
                * X=ADDRESS OF RESULT
                *
4855 DF 78      BCDSUB STX      SUBX1
4857 DF 7A              STX     SUBX2
                * FIX SUBX1
4859 C6 04              LDA B   #4
485B BD 47F9            JSR     FIXX
485E DF 78              STX     SUBX1
                * COMPLEMENT SUBTRAHAND
                * TRANSFER SUBTRAHAND
4860 CE 00AB            LDX     #TEMSUB
4863 DF 7C              STX     TX
4865 C6 05              LDA B   #5
4867 DE 68      TX1     LDX     S2
4869 A6 00              LDA A   0,X
486B 08                 INX
486C DF 68              STX     S2
                *
486E DE 7C              LDX     TX
4870 A7 00              STA A   0,X
4872 08                 INX
4873 DF 7C              STX     TX
                *
4875 5A                 DEC B
4876 26 EF              BNE     TX1
                *
```



```
4878 CE 00AB            LDX     #TEMSUB
487B BD 488D            JSR     COMP64
                * ADD MINUEND AND FIX SIGN OF RESULT
487E DE 66              LDX     S1
4880 DF 62              STX     A1
4882 CE 00AB            LDX     #TEMSUB
4885 DF 64              STX     A2
4887 DE 78              LDX     SUBX1
4889 BD 47FE            JSR     BCDADD
488C 39                 RTS
                *
                * 64'S COMPLEMENT ROUTINE
                *
488D DF 62      COMP64  STX     A1
488F BD 47D0            JSR     NINCOM
4892 CE 46DA            LDX     #CONST
4895 DF 64              STX     A2
4897 CE 0089            LDX     #RESULT+4
489A BD 47FE            JSR     BCDADD
489D DE 62              LDX     A1
489F BD 4902            JSR     XFER
48A2 39                 RTS
                *
                *
                * ADJUST FOR > 99999 & > 64000 ROLLOVER
                *
48A3 DF C4      ADJ     STX     ADJX
48A5 7D 0026            TST     SIG
48A8 26 08              BNE     ADJ1
                * TEST > 64000
48AA CE 0085            LDX     #RESULT
48AD BD 48F0            JSR     TEST64
48B0 24 15              BCC     ADJ2
                * FIX > 64000 ROLLOVER
48B2 CE 0085    ADJ1    LDX     #RESULT
48B5 DF 62              STX     A1
48B7 CE 46DF            LDX     #CONST2
48BA DF 64              STX     A2
48BC DE C4              LDX     ADJX
48BE C6 04              LDA B   #4
48C0 BD 47F9            JSR     FIXX
48C3 BD 47FE            JSR     BCDADD
48C6 39                 RTS
                * FIX < 64000 VALUE
48C7 DE C4      ADJ2    LDX     ADJX
48C9 BD 4902            JSR     XFER
48CC 39                 RTS
                *
                * TEST IF BCD ARRAY > OR = 32000 (C SET IF TRUE)
                *
48CD A6 00      TEST32  LDA A   0,X
48CF 81 03              CMP A   #3
48D1 2D 0A              BLT     T321
48D3 2E 06              BGT     T322
```

```
4805 A6 01          LDA A  1,X
48D7 81 02          CMP A  #2
48D9 2D 02          BLT    T321
48DB 0D      T322   SEC
48DC 39             RTS
48DD 0C      T321   CLC
48DE 39             RTS
             *
             *
             **********AGLS3**********
             *
             *
             * TEST BCD MAG LIMITS FOR DISPLAY
             *
             * X=BUFFER ADDRESS
             *
             * SETUP DIRECTION SIGN
             * FIX IF >9999
48DF A6 00   TESTM  LDA A  0,X
48E1 84 0F          AND A  #$F
48E3 27 0A          BEQ    TSTM1
48E5 86 09          LDA A  #09
48E7 A7 01          STA A  1,X
48E9 A7 02          STA A  2,X
48EB A7 03          STA A  3,X
48ED A7 04          STA A  4,X
48EF 39      TSTM1  RTS
             * TEST IF BCD ARRAY > OR = 64000
             * C SET IF TRUE;C=0 IF FALSE
             *
48F0 A6 00   TEST64 LDA A  0,X
48F2 81 06          CMP A  #6
48F4 2E 0A          BGT    T641
48F6 2D 06          BLT    T642
48F8 A6 01          LDA A  1,X
48FA 81 04          CMP A  #4
48FC 2C 02          BGE    T641
48FE 0C      T642   CLC
48FF 39             RTS
4900 0D      T641   SEC
4901 39             RTS
             *
             *
             * TRANSFER FROM BCD ARRAY "RESULT" TO
             * ARRAY SPECIFIED BY X
             *
4902 DF 6A   XFER   STX    X1
4904 CE 00B5        LDX    #RESULT
4907 DF 6C          STX    X2
4909 C6 05          LDA B  #5
490B DE 6C   XFER1  LDX    X2
490D A6 00          LDA A  0,X
490F 08             INX
4910 DF 6C          STX    X2
```



```
4912 DE 6A          LDX    X1
4914 A7 00          STA A  0,X
4916 08             INX
4917 DF 6A          STX    X1
4919 5A             DEC B
491A 26 EF          BNE    XFER1
491C 39             RTS
             * COMPUTE BINARY DATA FROM BCD(5 DIGIT) VALUE
             * X=ADDRESS OF BUFFER(ENTRY);D/A VALUE(EXIT)
             *
491D 08      BCDBIN INX
491E C6 04          LDA B  #4
4920 D7 8A          STA B  SAVB
4922 7F 0001        CLR    MSBY
4925 7F 0002        CLR    LSBY
             *
4928 A6 00   BCDB1  LDA A  0,X
492A 97 03          STA A  TMP
492C BD 4BE1        JSR    MIOX
492F 08             INX
4930 7A 008A        DEC    SAVB
4933 26 F3          BNE    BCDB1
             * DIVIDE X2
4935 74 0001        LSR    MSBY
4938 76 0002        ROR    LSBY
             * TEST SIGN
493B 96 01          LDA A  MSBY
493D 7D 0026        TST    SIG
4940 27 04          BEQ    BCDB2
4942 8A 80          ORA A  #$80
4944 97 01          STA A  MSBY
             * TEST OVERFLOW
4946 84 70   BCDB2  AND A  #$70
4948 27 0F          BEQ    BCDB3
             *
494A 96 01          LDA A  MSBY
494C 84 80          AND A  #$80
494E 8A 0F          ORA A  #$0F
4950 97 01          STA A  MSBY
4952 86 FF          LDA A  #$FF
4954 97 02          STA A  LSBY
4956 B7 241D        STA A  PIA8DB
4959 DE 01   BCDB3  LDX    MSBY
495B 39             RTS
             *
             * CLEAR FLAG TABLES
             *
495C CE 0013 CLFG   LDX    #BEG
495F 6F 00   CLFG1  CLR    0,X
4961 08             INX
4962 8C 00C9        CPX    #END
4965 26 F8          BNE    CLFG1
4967 39             RTS
             *
```

```
                 * INITIAL PIAS ROUTINE
                 *
      4968       PIAS    EQU    *
                 * PIA 0 (A)-SW INPUTS
4968 7F 2802             CLR    PIA0CA
496B 7F 2800             CLR    PIA0DA
496E 86 3E               LDA A  #$3E
4970 B7 2802             STA A  PIA0CA
4973 86 3E               LDA A  #$3E
4975 B7 2802             STA A  PIA0CA
                 * PIA 0 (B)-PROG CLOCK
4978 7F 2803             CLR    PIA0CB
497B 86 FF               LDA A  #$FF
497D B7 2801             STA A  PIA0DB
4980 86 2C               LDA A  #$2C
4982 B7 2803             STA A  PIA0CB
4985 86 0A               LDA A  #10
4987 B7 2801             STA A  PIA0DB
498A B6 2801             LDA A  PIA0DB
498D 86 2D               LDA A  #$2D
498F B7 2803             STA A  PIA0CB
4992 0F                  SEI
4993 01                  NOP
                 * PIA 1 -GACS INPUT
4994 86 3E               LDA A  #$3E
4996 CE 2400             LDX    #PIA1DA
4999 BD 4A29             JSR    SETUP
                 * PIA 2 -ELEVATION ENCODER
499C 86 3E               LDA A  #$3E
499E CE 2404             LDX    #PIA2DA
49A1 BD 4A29             JSR    SETUP
                 * PIA3 -AZIMUTH ENCODER
49A4 86 3E               LDA A  #$3E
49A6 CE 2408             LDX    #PIA3DA
49A9 BD 4A29             JSR    SETUP
                 *
                 * PIA 4 -EL AND AZ ENCODER
49AC 7F 240E             CLR    PIA4CA
49AF 7F 240C             CLR    PIA4DA
49B2 86 3E               LDA A  #$3E
49B4 B7 240E             STA A  PIA4CA
                 * PIA 4 -ENABLES
49B7 86 04               LDA A  #4
49B9 B7 240F             STA A  PIA4CB
49BC 86 FF               LDA A  #$FF
49BE B7 240D             STA A  PIA4DB
49C1 7F 240F             CLR    PIA4CB
49C4 86 FF               LDA A  #$FF
49C6 B7 240D             STA A  PIA4DB
49C9 86 04               LDA A  #4
49CB B7 240F             STA A  PIA4CB
                 *
                 * PIA 5 -MUX A/D
49CE 7F 2412             CLR    PIA5CA
```



```
49D1 7F 2413             CLR    PIA5CB
49D4 7F 2410             CLR    PIA5DA
49D7 86 FF               LDA A  #$FF
49D9 B7 2411             STA A  PIA5DB
49DC 86 3C               LDA A  #$3C
49DE B7 2412             STA A  PIA5CA
49E1 86 34               LDA A  #$34
49E3 B7 2413             STA A  PIA5CB
                 *
                 * PIA 6 -TRIM A/D
49E6 86 36               LDA A  #$36
49E8 CE 2414             LDX    #PIA6DA
49EB BD 4A29             JSR    SETUP
                 *
                 * PIA 7 -D/A
49EE 7F 241A             CLR    PIA7CA
49F1 7F 241B             CLR    PIA7CB
49F4 86 FF               LDA A  #$FF
49F6 B7 2418             STA A  PIA7DA
49F9 B7 2419             STA A  PIA7DB
49FC 86 3E               LDA A  #$3E
49FE B7 241A             STA A  PIA7CA
4A01 B7 241B             STA A  PIA7CB
                 *
                 * PIA 8 -DISPLAY
4A04 7F 241E             CLR    PIA8CA
4A07 7F 241F             CLR    PIA8CB
4A0A 86 0F               LDA A  #$F
4A0C B7 241C             STA A  PIA8DA
4A0F 86 1F               LDA A  #$1F
4A11 B7 241D             STA A  PIA8DB
4A14 86 3E               LDA A  #$3E
4A16 B7 241E             STA A  PIA8CA
4A19 86 3E               LDA A  #$3E
4A1B B7 241F             STA A  PIA8CB
                 *
                 * ACIA SETUP
4A1E 86 03               LDA A  #3
4A20 B7 3002             STA A  AC2C
4A23 BD 4B19             JSR    DISXMT
4A26 01                  NOP
4A27 01                  NOP
4A28 39                  RTS
                 *
                 * SETUP PIA USING X AND A REG.
                 *
4A29 6F 02       SETUP   CLR    2,X
4A2B 6F 03               CLR    3,X
4A2D 6F 00               CLR    0,X
4A2F 6F 01               CLR    1,X
4A31 A7 02               STA A  2,X
4A33 A7 03               STA A  3,X
4A35 39                  RTS
                 *
```

```
                  * INTERRUPT SERVICE ROUTINE
                  *
                  * TEST CLOCK
4A36 7D 2803 ISER   TST    PIAOCB
4A39 2A 6C          BPL    ISER6
4A3B B6 2801        LDA A  PIAODB
4A3E B7 2801        STA A  PIAODB
                  * SCAN CLOCKS
4A41 CE 0004        LDX    #TMTB
4A44 C6 05          LDA B  #5
4A46 BD 4C0A        JSR    SCAT
                  * TEST INSIDE LOOP(20 MSEC)
4A49 7D 0004        TST    TF1
4A4C 27 01          BEQ    ISER2
4A4E 3B             RTI
                  * SERVICE INSIDE LOOP
4A4F CE 0014 ISER2  LDX    #20
4A52 DF 05          STX    TIM1
4A54 7C 0004        INC    TF1
                  * D/A READY?
4A57 BD 4B22        JSR    DAOUT
                  * TEST XENON ON
4A5A 86 40          LDA A  #XRECM
4A5C BD 44C5        JSR    TSTS8B
4A5F 24 1D          BCC    ISER7
                  * START DROPOUT CLOCK
4A61 7F 0010        CLR    TF5
4A64 CE 03E8        LDX    #1000
4A67 DF 11          STX    TIM5
4A69 7C 0010        INC    TF5
                  * TEST ON FOR 25 TIMES
4A6C 7C 004E        INC    XON
4A6F 96 4E          LDA A  XON
4A71 81 19          CMP A  #25
4A73 2D 14          BLT    ISER8
                  *
4A75 7A 004E        DEC    XON
4A78 7F 0017        CLR    XRECF
4A7B 7C 0017        INC    XRECF
                  * TEST TIMEOUT AFTER OFF
4A7E 7F 004E ISER7  CLR    XON
4A81 7D 0010        TST    TF5
4A84 26 03          BNE    ISER8
4A86 7F 0017        CLR    XRECF
                  * TEST WEAPON SW
4A89 86 80   ISER8  LDA A  #WPNM
4A8B BD 44CE        JSR    TSTS8A
4A8E 24 06          BCC    ISER4
4A90 7F 0016        CLR    WPNF
4A93 7C 0016        INC    WPNF
                  * OUTSIDE LOOP
4A96 7D 0007 ISER4  TST    TF2
4A99 27 01          BEQ    ISER5
4A9B 3B             RTI
```



```
                  * SERVICE OUTSIDE LOOP
4A9C CE 00C8 ISER5  LDX    #200
4A9F DF 08          STX    TIM2
4AA1 7C 0007        INC    TF2
                  * UPDATE DISPLAY
4AA4 BD 4856        JSR    UDIS
                  * TEST CRT
4AA7 7D 3002 ISER6  TST    AC2S
4AAA 2B 01          BMI    ISER16
4AAC 3B             RTI
                  * TEST RECV. INT.
4AAD BD 4C33 ISER16 JSR    AOI
4AB0 25 2E          BCS    ISER17
                  *  =D ?
4AB2 81 44          CMP A  #'D
4AB4 26 0E          BNE    ISER9
4AB6 CE 008B        LDX    #ELGCDS
4AB9 DF 48          STX    PTR
4ABB CE 00AB        LDX    #AZLIT+1
4ABE DF 4A          STX    PTE
4AC0 86 05          LDA A  #5
4AC2 20 11          BRA    ISER10
                  *  =F ?
4AC4 81 46   ISER9  CMP A  #'F
4AC6 27 01          BEQ    ISER11
4AC8 3B             RTI
4AC9 CE 0013 ISER11 LDX    #BEG
4ACC DF 48          STX    PTR
4ACE CE 002D        LDX    #DISAZ+1
4AD1 DF 4A          STX    PTE
4AD3 86 01          LDA A  #1
4AD5 97 4C   ISER10 STA A  SPC
4AD7 97 4D          STA A  ASP
                  * OUTPUT CR/LF
4AD9 BD 4C51        JSR    CRLF
                  * INH REC/ENB XMIT
4ADC BD 4B13        JSR    DISREC
4ADF 3B             RTI
                  *
                  * TEST XMIT. INT.
4AE0 B6 3002 ISER17 LDA A  AC2S
4AE3 85 02          BIT A  #2
4AE5 26 04          BNE    ISER12
4AE7 B6 3003        LDA A  AC2R
4AEA 3B             RTI
                  * CHECK SPACE COUNT
4AEB 7D 004D ISER12 TST    ASP
4AEE 26 08          BNE    ISER13
4AF0 96 4C          LDA A  SPC
4AF2 97 4D          STA A  ASP
4AF4 86 20          LDA A  #$20
4AF6 20 13          BRA    ISER14
                  * PROCESS CHAR
4AF8 7A 004D ISER13 DEC    ASP
```

```
4AFB DE 48                LDX      PTR
4AFD 9C 4A                CPX      PTE
4AFF 27 0E                BEQ      ISER15
4B01 A6 00                LDA A    0,X
4B03 08                   INX
4B04 DF 48                STX      PTR
4B06 2A 01                BPL      ISER18
4B08 4F                   CLR A
4B09 8B 30      ISER18    ADD A    #$30
                * OUTPUT CHAR
4B0B BD 4C41    ISER14    JSR      A(X)
4B0E 3B                   RTI
                * WRAP UP XMIT
4B0F BD 4B19    ISER15    JSR      DISXMT
4B12 3B                   RTI
                *
                * DISABLE REC INT ROUTINE
4B13 86 2A      DISREC    LDA A    #XIE
4B15 B7 3002              STA A    AC2C
4B18 39                   RTS
                *
                * DISABLE XMIT INT ROUTINE
4B19 86 8A      DISXMT    LDA A    #RIE
4B1B B7 3002              STA A    AC2C
4B1E B6 3003              LDA A    AC2R
4B21 39                   RTS
                *
                * OUTPUT D/A ROUTINE
                *
                * ELEVATION
4B22 7D 0041    DAOUT     TST      ELDAF
4B25 27 10                BEQ      DAOUT1
4B27 26 03                BNE      DAOUT3
4B29 7A 0041              DEC      ELDAF
4B2C DE 51      DAOUT3    LDX      ELCOM
4B2E FF 2418              STX      PIA7DA
4B31 CE 241A              LDX      #PIA7CA
4B34 BD 4B4D              JSR      USCON
                *
                * AZIMUTH
4B37 7D 0042    DAOUT1    TST      AZDAF
4B3A 2D 10                BLT      DAOUT4
4B3C 26 03                BNE      DAOUT2
4B3E 7A 0042              DEC      AZDAF
4B41 DE 4F      DAOUT2    LDX      AZCOM
4B43 FF 2418              STX      PIA7DA
4B46 CE 241B              LDX      #PIA7CB
4B49 BD 4B4D              JSR      USCON
4B4C 39         DAOUT4    RTS
                *
                * INVERTED STROBE CONTROL PULSE
                *
4B4D C6 36      USCON     LDA B    #$36
4B4F E7 00                STA B    0,X
```



```
4B51 C6 3E                LDA B    #$3E
4B53 E7 00                STA B    0,X
4B55 39                   RTS
                *
                * UPDATE DISPLAY ROUTINE
                * ELEVATION
                *
                * FLASH SIGN IF IDLE
4B56 7D 001A    UDIS      TST      IDLEF
4B59 27 13                BEQ      UDIS3
4B5B 7D 002A              TST      DTHRU
4B5E 27 05                BEQ      UDIS4
4B60 7F 002A              CLR      DTHRU
4B63 20 09                BRA      UDIS3
4B65 7C 002A    UDIS4     INC      DTHRU
4B68 86 0F                LDA A    #$F
4B6A 97 95                STA A    ELERD
4B6C 97 A4                STA A    AZERD
4B6E 7F 0043    UDIS3     CLR      DISADR
4B71 7D 002B              TST      DISEL
4B74 27 0C                BEQ      UDIS1
4B76 CE 008B              LDX      #ELGCDS
4B79 DF 44                STX      ACT
4B7B D6 A9                LDA B    ELLIT
4B7D D7 3F                STA B    LITE
4B7F BD 4B98              JSR      DISIT
                * AZIMUTH
4B82 7D 002C    UDIS1     TST      DISAZ
4B85 27 10                BEQ      UDIS2
4B87 86 10                LDA A    #$10
4B89 97 43                STA A    DISADR
4B8B CE 009A              LDX      #AZGCDS
4B8E DF 44                STX      ACT
4B90 D6 AA                LDA B    AZLIT
4B92 D7 3F                STA B    LITE
4B94 BD 4B98              JSR      DISIT
4B97 39         UDIS2     RTS
                *
                * ROUTINE TO DISPLAY IT
                *
4B98 96 43      DISIT     LDA A    DISADR
4B9A B7 241D              STA A    PIA8DB
4B9D DE 44                LDX      ACT
4B9F A6 00                LDA A    0,X
4BA1 08                   INX
4BA2 DF 44                STX      ACT
4BA4 B7 241C              STA A    PIA8DA
4BA7 CE 241E              LDX      #PIA8CA
4BAA BD 4B4D              JSR      USCON
4BAD 7C 0043              INC      DISADR
4BB0 96 43                LDA A    DISADR
4BB2 84 0F                AND A    #$0F
4BB4 81 0F                CMP A    #$0F
4BB6 26 E0                BNE      DISIT
```

```
                         * OUTPUT SPECIAL DISPLAYS
4BB8 7D 0027                    TST     CKOF
4BBB 26 01                      BNE     DISIT2
4BBD 39                         RTS
4BBE 96 43               DISIT2 LDA A   DISADR
4BC0 B7 241D                    STA A   PIA8DB
4BC3 CE 4BD9                    LDX     #CODE
4BC6 4F                         CLR A
4BC7 D6 3F                      LDA B   LITE
4BC9 27 05                      BEQ     DISIT1
4BCB BD 47F9                    JSR     FIXX
4BCE A6 00                      LDA A   0,X
4BD0 B7 241C             DISIT1 STA A   PIA8DA
4BD3 CE 241E                    LDX     #PIA8CA
4BD6 BD 4B4D                    JSR     USCON
4BD9 39                         RTS
                         * DISPLAY CODE TABLE
        4BD9             CODE   EQU     *-1
4BDA 05                         FCB     5
4BDB 02                         FCB     2
4BDC 0F                         FCB     $F
4BDD 04                         FCB     4
4BDE 06                         FCB     6
4BDF 01                         FCB     1
4BE0 0F                         FCB     $F
                         *
                         ***********AGLS4************
                         *
                         * EXEC SUBROUTINES
                         * MULTIPLY MSBY/LSBY X 10+TMP
4BE1 96 01               M10X   LDA A   MSBY
4BE3 D6 02                      LDA B   LSBY
4BE5 0C                         CLC
4BE6 48                         ASL A
4BE7 58                         ASL B
4BE8 BD 4C04                    JSR     CKC
4BEB 48                         ASL A
4BEC 58                         ASL B
4BED BD 4C04                    JSR     CKC
4BF0 DB 02                      ADD B   LSBY
4BF2 BD 4C04                    JSR     CKC
4BF5 48                         ASL A
4BF6 58                         ASL B
4BF7 BD 4C04                    JSR     CKC
4BFA DB 03                      ADD B   TMP
4BFC BD 4C04                    JSR     CKC
4BFF 97 01                      STA A   MSBY
4C01 D7 02                      STA B   LSBY
4C03 39                         RTS
                         * CHECK C BIT AND FIX
4C04 25 01               CKC    BCS     CKC1
4C06 39                         RTS
4C07 4C                  CKC1   INC A
4C08 0C                         CLC
```



```
4C09 39                         RTS
                         *
                         * SCAN TIMERS ROUTINE
                         *
4C0A 6D 00               SCAT   TST     0,X
4C0C 27 12                      BEQ     RT
4C0E A6 02                      LDA A   2,X
4C10 A0 00                      SUB A   0,X
4C12 A7 02                      STA A   2,X
4C14 26 0A                      BNE     RT
4C16 6D 01                      TST     1,X
4C18 27 04                      BEQ     ST3
4C1A 6A 01                      DEC     1,X
4C1C 20 02                      BRA     RT
4C1E 6F 00               ST3    CLR     0,X
4C20 08                  RT     INX
4C21 08                         INX
4C22 08                         INX
4C23 5A                         DEC B
4C24 26 E4                      BNE     SCAT
4C26 39                         RTS
                         * CLEAR TIMERS ROUTINE
4C27 C6 0F               CLTM   LDA B   #15
4C29 CE 0004                    LDX     #TMTB
4C2C 6F 00               CLT1   CLR     0,X
4C2E 08                         INX
4C2F 5A                         DEC B
4C30 26 FA                      BNE     CLT1
4C32 39                         RTS
                         * INPUT FROM ACIA
4C33 B6 3002             AOI    LDA A   AC2S
4C36 85 01                      BIT A   #1
4C38 27 05                      BEQ     AOI1
4C3A B6 3003                    LDA A   AC2R
4C3D 0C                         CLC
4C3E 39                         RTS
4C3F 0D                  AOI1   SEC
4C40 39                         RTS
                         * OUTPUT TO ACIA
4C41 36                  AOO    PSH A
4C42 B6 3002                    LDA A   AC2S
4C45 85 02                      BIT A   #2
4C47 32                         PUL A
4C48 27 05                      BEQ     AOO1
4C4A B7 3003                    STA A   AC2T
4C4D 0C                         CLC
4C4E 39                         RTS
4C4F 0D                  AOO1   SEC
4C50 39                         RTS
                         * CR/LF ROUTINE
4C51 86 0D               CRLF   LDA A   #$D
4C53 BD 4C5C                    JSR     AOOL
4C56 86 0A                      LDA A   #$A
4C58 BD 4C5C                    JSR     AOOL
```

```
4C5B 39                  RTS
                * LOOP ON OUTPUT
4C5C BD 4C41 AOOL        JSR      AOO
4C5F 25 FB               BCS      AOOL
4C61 39                  RTS
                *
                **************AGLS5************
                *
                * TEST AZ CLOSING
                *
                * TEST PASS FLAG
4C62 7D 0046 CLAZ        TST      PASSAZ
4C65 26 1E               BNE      CLAZ1
                * TEST AZERR
4C67 96 53               LDA A    AZERR
4C69 84 80               AND A    #$80
4C6B 27 48               BEQ      CLAZ6
                * AZERR < 0
4C6D CE 0096             LDX      #150
4C70 DF 0B               STX      TIM3
4C72 86 01               LDA A    #1
4C74 97 46               STA A    PASSAZ
4C76 CE 8FFF             LDX      #FULBAK
4C79 7D 0029             TST      SLOWF
4C7C 27 03               BEQ      CLAZ8
4C7E CE 85FF             LDX      #HAFBAK
4C81 DF 4F   CLAZ8       STX      AZCOM
4C83 0C                  CLC
4C84 39                  RTS
                * PASS = 1
4C85 96 46   CLAZ1       LDA A    PASSAZ
4C87 4A                  DEC A
4C88 26 21               BNE      CLAZ3
                * TEST FOR FIRST NULL
4C8A CE 8FFF             LDX      #FULBAK
4C8D 7D 0029             TST      SLOWF
4C90 27 03               BEQ      CLAZ9
4C92 CE 85FF             LDX      #HAFBAK
4C95 DF 4F   CLAZ9       STX      AZCOM
4C97 7D 0053             TST      AZERR
4C9A 2A 02               BPL      CLAZ4
4C9C 0C                  CLC
4C9D 39                  RTS
                * NULL ACHIEVED
4C9E CE 85FF CLAZ4       LDX      #HAFBAK
4CA1 DF 4F               STX      AZCOM
4CA3 7C 000A             INC      TF3
4CA6 7C 0046             INC      PASSAZ
4CA9 0C                  CLC
4CAA 39                  RTS
                * PASS = 2
4CAB 4A      CLAZ3       DEC A
4CAC 26 12               BNE      CLAZ2
                * TEST TIMER
```



```
4CAE 7D 000A             TST      TF3
4CB1 27 02               BEQ      CLAZ6
4CB3 0C                  CLC
4CB4 39                  RTS
                * DISABLE D/A
4CB5 CE 0000 CLAZ6       LDX      #0
4CB8 DF 4F               STX      AZCOM
4CBA 86 03               LDA A    #3
4CBC 97 46               STA A    PASSAZ
4CBE 0C                  CLC
4CBF 39                  RTS
                * PASS=3
4CC0 4A      CLAZ2       DEC A
4CC1 26 F2               BNE      CLAZ6
4CC3 BD 4CDF             JSR      AZNULL
4CC6 25 0D               BCS      CLAZ7
4CC8 96 53               LDA A    AZERR
4CCA D6 54               LDA B    AZERR+1
4CCC BD 4D5E             JSR      TMAGA
4CCF 97 4F               STA A    AZCOM
4CD1 D7 50               STA B    AZCOM+1
4CD3 0C                  CLC
4CD4 39                  RTS
4CD5 CE 0000 CLAZ7       LDX      #0
4CD8 DF 4F               STX      AZCOM
4CDA 7F 0046             CLR      PASSAZ
4CDD 0D                  SEC
4CDE 39                  RTS
                *
                * TEST AZ NULL
                *
                * AZERR TEST
4CDF BD 4621 AZNULL      JSR      TAZERR
4CE2 25 0E               BCS      AZN1
4CE4 7C 0057             INC      AZCNT
4CE7 96 57               LDA A    AZCNT
4CE9 81 19               CMP A    #25
4CEB 2F 08               BLE      AZN2
4CED 7A 0057             DEC      AZCNT
4CF0 0D                  SEC
4CF1 39                  RTS
4CF2 7F 0057 AZN1        CLR      AZCNT
4CF5 0C      AZN2        CLC
4CF6 39                  RTS
                *
                * TEST EL CLOZING
                *
                * TEST ELERR
4CF7 DE 55   CLEL        LDX      ELERR
4CF9 DF 51               STX      ELCOM
4CFB 8C 0000             CPX      #0
4CFE 26 01               BNE      CLEL1
4D00 39                  RTS
                * TEST NULL
```

```
4D01 BD 4D0E CLEL1   JSR     ELNULL
4D04 25 C1           BCS     CLEL2
4D06 39              RTS
4D07 CE 0000 CLEL2   LDX     #0
4D0A DF 51           STX     ELCOM
4D0C 0D              SEC
4D0D 39              RTS
                     *
                     *
                     * TEST EL NULL
                     *
                     * ELERR TEST
4D0E BD 4629 ELNULL  JSR     TELERR
4D11 25 0E           BCS     ELN1
4D13 7C 0058         INC     ELCNT
4D16 96 58           LDA A   ELCNT
4D18 81 19           CMP A   #25
4D1A 2F 08           BLE     ELN2
4D1C 7A 0058         DEC     ELCNT
4D1F 0D              SEC
4D20 39              RTS
4D21 7F 0058 ELN1    CLR     ELCNT
4D24 0C      ELN2    CLC
4D25 39              RTS
                     *
                     * TEST TRACKER NULL
                     *
4D26 C6 05   CLTR    LDA B   #PAMA
4D28 BD 454E         JSR     TERR
4D2B 25 0E           BCS     CLTR2
4D2D DE 5B           LDX     TRCNT
4D2F 08              INX
4D30 DF 5B           STX     TRCNT
4D32 9C 5D           CPX     TRLP
4D34 26 0A           BNE     CLTR3
4D36 09              DEX
4D37 DF 5B           STX     TRCNT
4D39 0D              SEC
4D3A 39              RTS
                     *
4D3B CE 0000 CLTR2   LDX     #0
4D3E DF 5B           STX     TRCNT
                     *
4D40 0C      CLTR3   CLC
4D41 39              RTS
                     *
                     * TEST QUAD PITCH NULL
                     *
4D42 C6 01   CLOP    LDA B   #QPMA
4D44 BD 454F         JSR     TERR
4D47 25 0E           BCS     CLQP2
4D49 DE 59           LDX     QPCNT
4D4B 08              INX
4D4C DF 59           STX     QPCNT
```



```
4D4E 9C 60           CPX     QPLP
4D50 26 0A           BNE     CLQP3
4D52 09              DEX
4D53 DF 59           STX     QPCNT
4D55 0D              SEC
4D56 39              RTS
                     *
4D57 CE 0000 CLQP2   LDX     #0
4D5A DF 59           STX     QPCNT
                     *
4D5C 0C      CLQP3   CLC
4D5D 39              RTS
                     * TEST & FIX A/D MAGNITUDE
4D5E 7F 0028 TMAGA   CLR     NEGF
4D61 4D              TST A
4D62 2A 03           BPL     TMAG3
4D64 7C 0028         INC     NEGF
4D67 84 0F   TMAG3   AND A   #$F
4D69 7D 0029         TST     SLOWF
4D6C 26 13           BNE     TMAG1
                     * FULL SPEED
4D6E 81 02           CMP A   #2
4D70 2C 03           BGE     TMAG2
4D72 8D 23           BSR     M8X
4D74 39              RTS
                     *
4D75 86 0F   TMAG2   LDA A   #$0F
4D77 7D 0028         TST     NEGF
4D7A 27 02           BEQ     TMAG4
4D7C 8A 80           ORA A   #$80
4D7E C6 FF   TMAG4   LDA B   #$FF
4D80 39              RTS
                     * HALF SPEED
4D81 4D      TMAG1   TST A
4D82 26 07           BNE     TMAG5
4D84 C1 6F           CMP B   #$6F
4D86 22 03           BHI     TMAG5
4D88 8D 0D           BSR     M8X
4D8A 39              RTS
                     *
4D8B 86 05   TMAG5   LDA A   #5
4D8D 7D 0028         TST     NEGF
4D90 27 02           BEQ     TMAG6
4D92 8A 80           ORA A   #$80
4D94 C6 FF   TMAG6   LDA B   #$FF
4D96 39              RTS
                     *
                     * MULTIPLY X 8
                     *
4D97 CE 0003 M8X     LDX     #3
4D9A 58      M81     ASL B
4D9B 49              ROL A
4D9C 09              DEX
4D9D 26 FB           BNE     M81
```

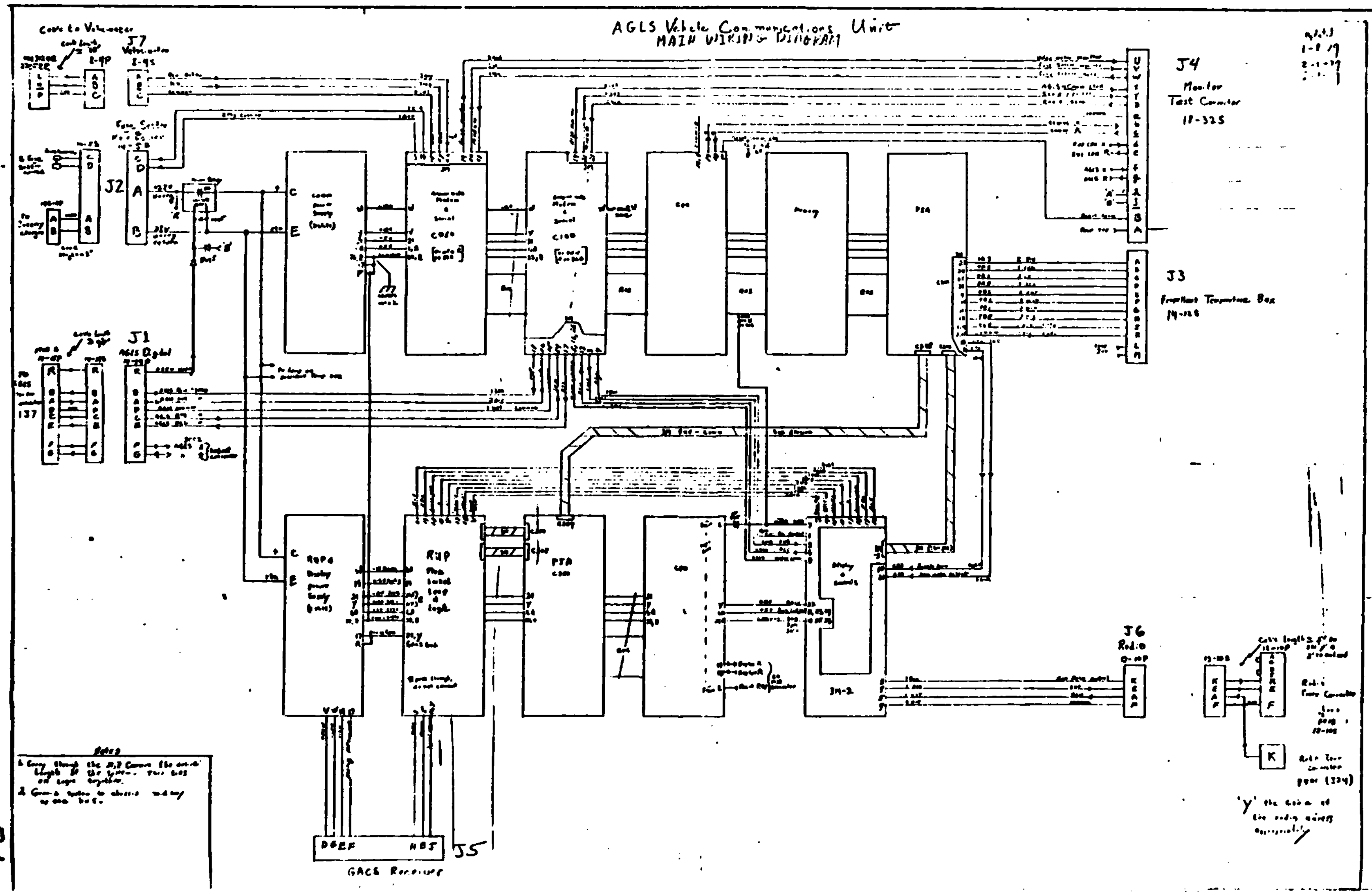
```
02120 4D9F 7D 0028          TST     NEGF
02130 4DA2 27 02            BEQ     M82
02140 4DA4 8A 80            ORA A   #$80
02150 4DA6 39         M82   RTS
02160 4DA7 26 BE            BNE     TMAG3
02170                 *
02180                 *
02190                 * OR BLOCK FLAGS
02200                 *
02210 4DA9 7D 001D ORBLK    TST     TBLK
02220 4DAC 26 07            BNE     ORB1
02230 4DAE 7D 001E          TST     EBLK
02240 4DB1 26 02            BNE     ORB1
02250 4DB3 0C               CLC
02260 4DB4 39               RTS
02270 4DB5 0D         ORB1  SEC
02280 4DB6 39               RTS
02290                 * TEST XENON STABILITY
02300 4DB7 7D 0017 XSTAB    TST     XRECF
02310 4DBA 27 0E            BEQ     XS1
02320 4DBC 7C 005F          INC     XTIME
02330 4DBF 96 5F            LDA A   XTIME
02340 4DC1 81 46            CMP A   #70
02350 4DC3 23 08            BLS     XS2
02360 4DC5 7A 005F          DEC     XTIME
02370 4DC8 0D               SEC
02380 4DC9 39               RTS
02390                 *
02400 4DCA 7F 005F XS1      CLR     XTIME
02410 4DCD 0C      XS2      CLC
02420 4DCE 39               RTS
02430 4FF8                  ORG     $4FF8
02440 4FF8 4A36             FDB     ISER
02450 4FFA 4000             FDB     AGO
02460 4FFC 4A36             FDB     ISER
02470 4FFE 4000             FDB     AGO
09999                       END

TOTAL ERRORS 00000

ENTER PASS : 1P,2P,2L,2T
```

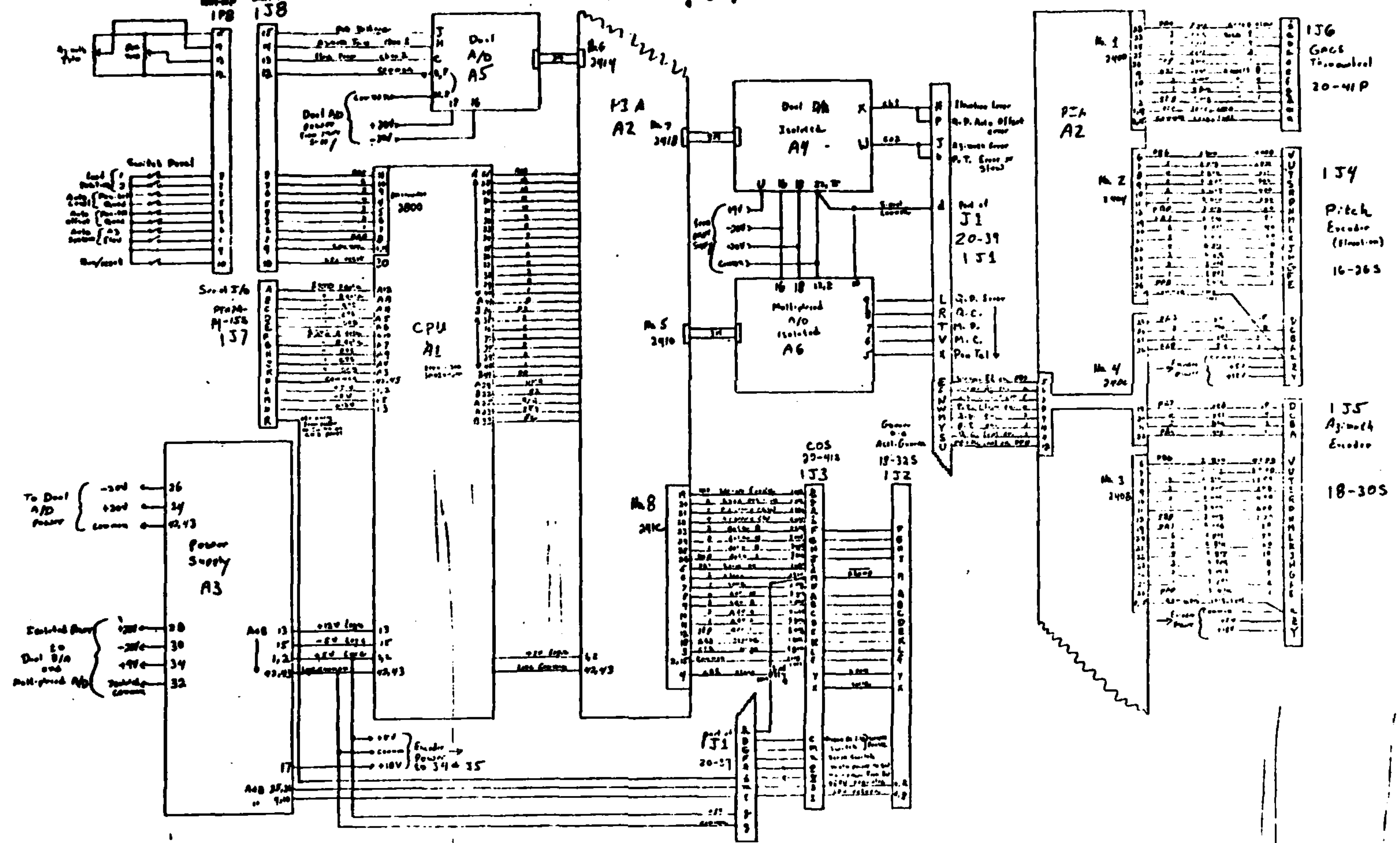
APPENDIX E

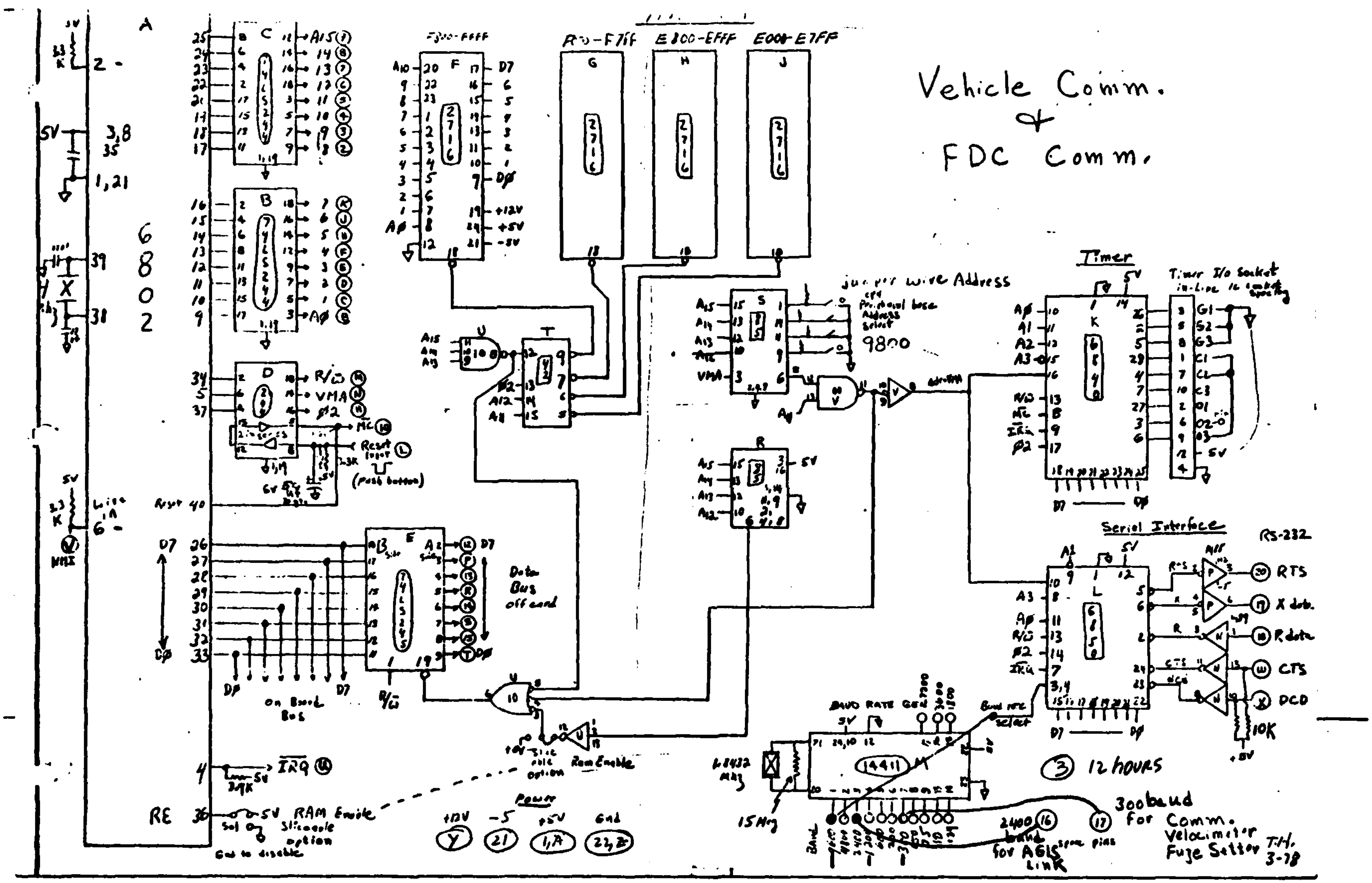
VECOM/RUP SCHEMATICS


AGLS Vehicle Communications Unit
MAIN WIRING DIAGRAM
J4
Monitor
Test Connector
J3
J1
J2
J5
J6
Radio
J7
Cable to Voltmeter
GACS Receiver
K
Notes
E1

AGLS Digital Controller Unit

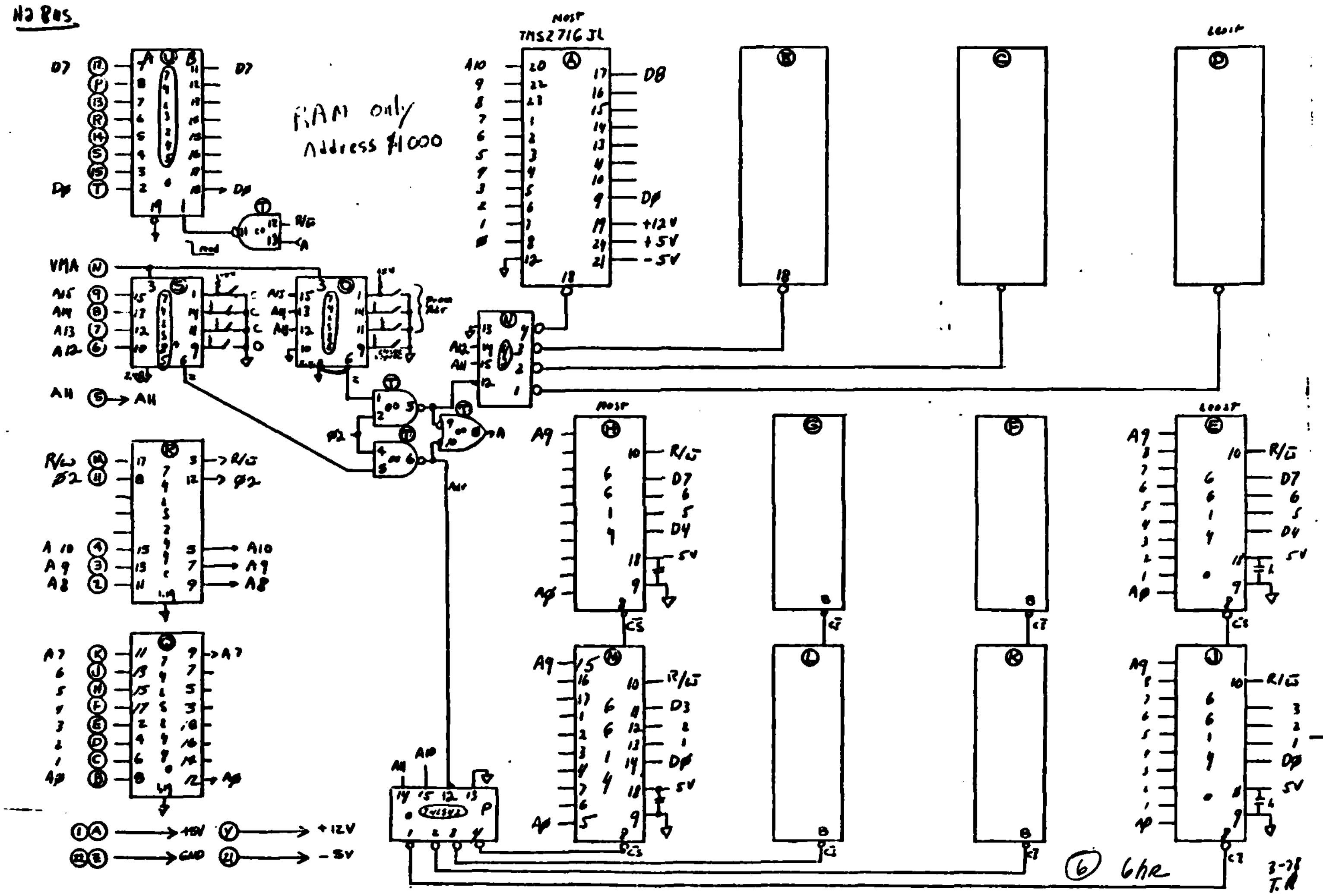
Main Wiring Diagram



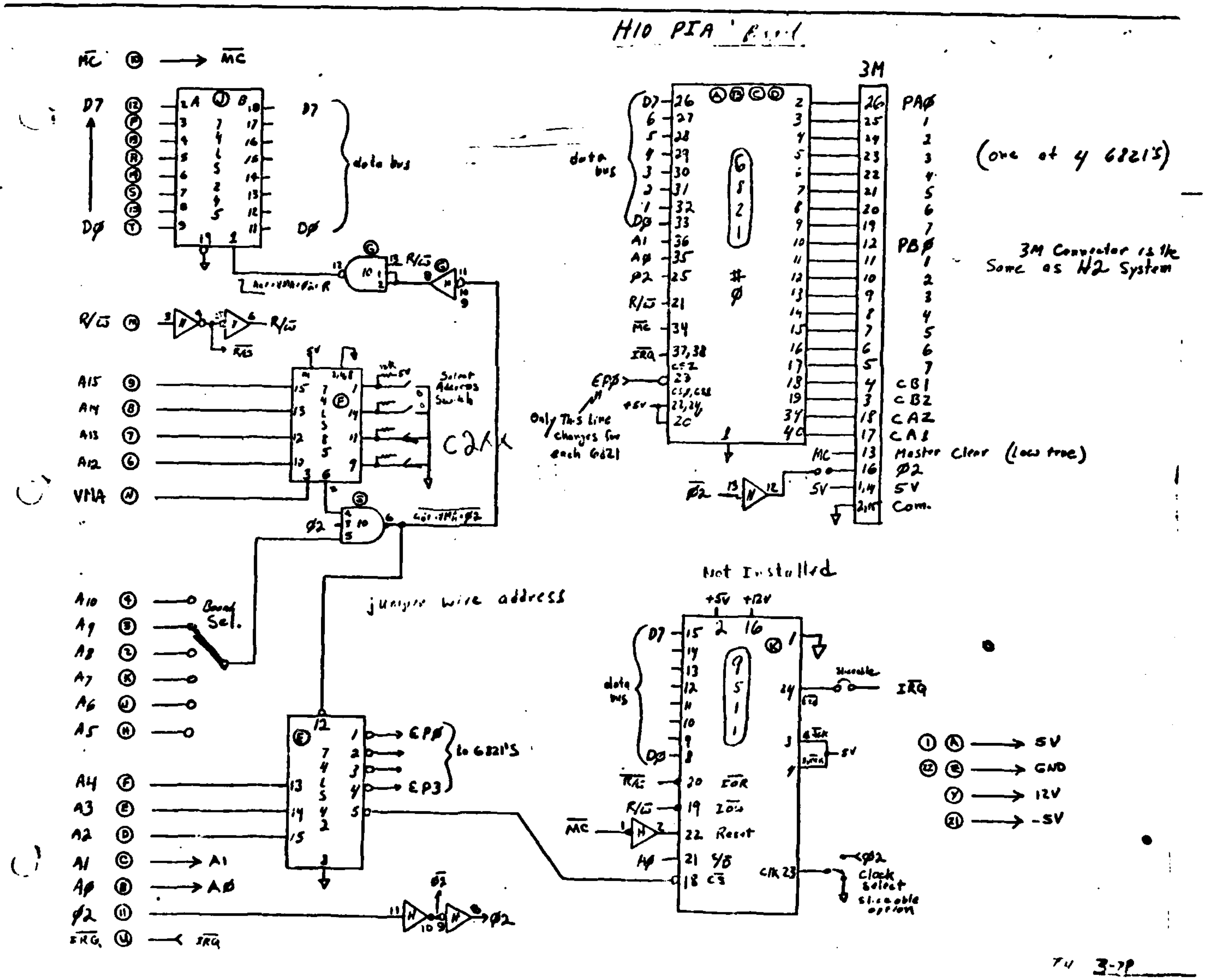


Vehicle Comm.
&
FDC Comm.
6802
Timer
Serial Interface
RS-232
RTS
X data
R data
CTS
DCD
Jumper wire Address
9800
Baud Rate Gen
15 MHz
3 12 hours
300 baud for Comm. Velocimeter Fuze Setter
2400 baud for AGLS Link
T.H. 3-78
Data Bus off card
On Board Bus
Power
+12V
-5
+5V
Gnd
RAM Enable
IRQ
Reset input (push button)

E3



E4

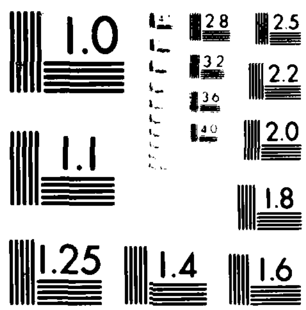


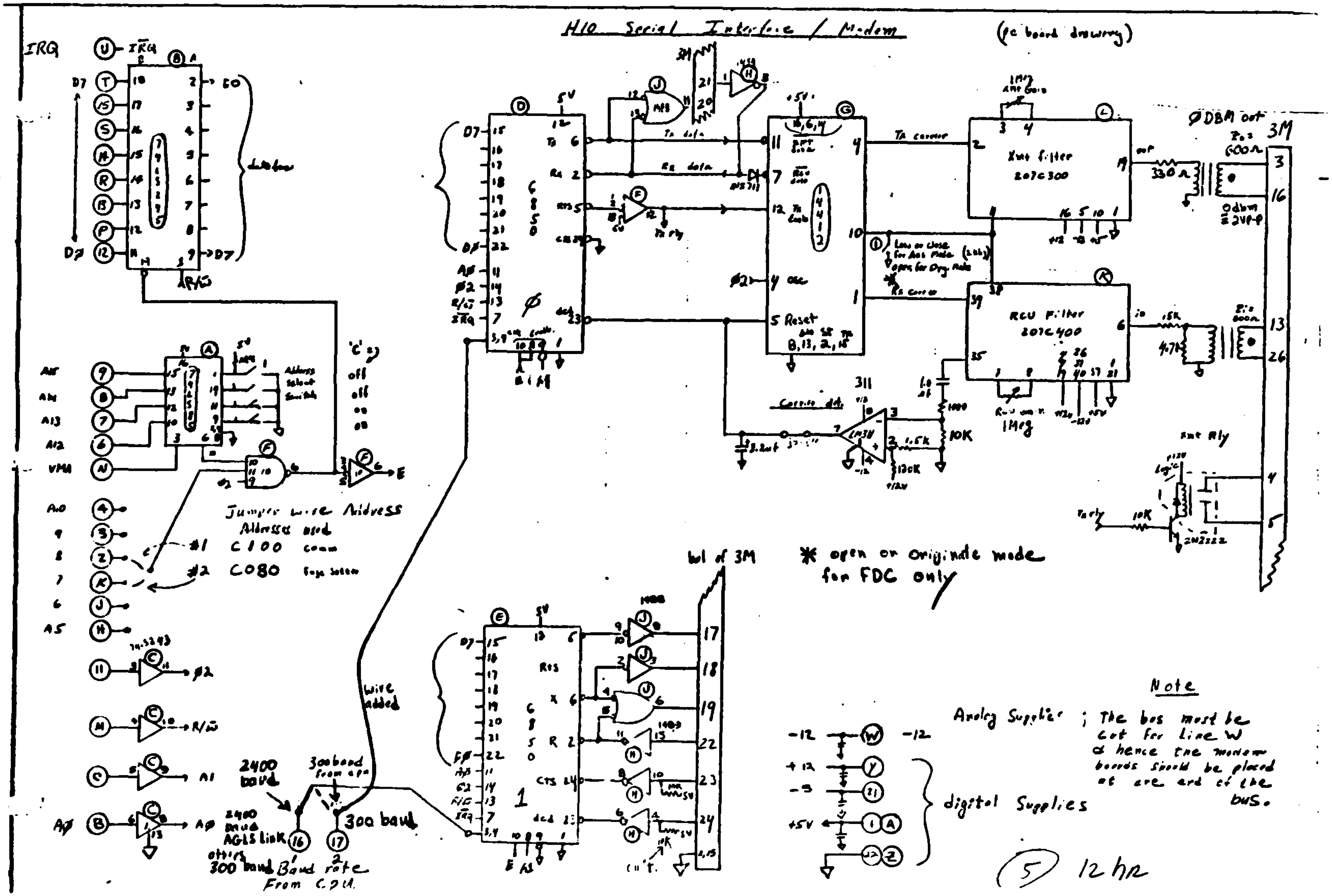
H10 PIA
(one of 4 6821'S)
3M Connector is the Same as H2 System
3M
data bus
Only This Line changes for each 6821
Master Clear (low true)
C2AX
Select Address Switch
jumper wire address
Board Sel.
Not Installed
to 6821'S
Clock Select
5V
GND
12V
-5V

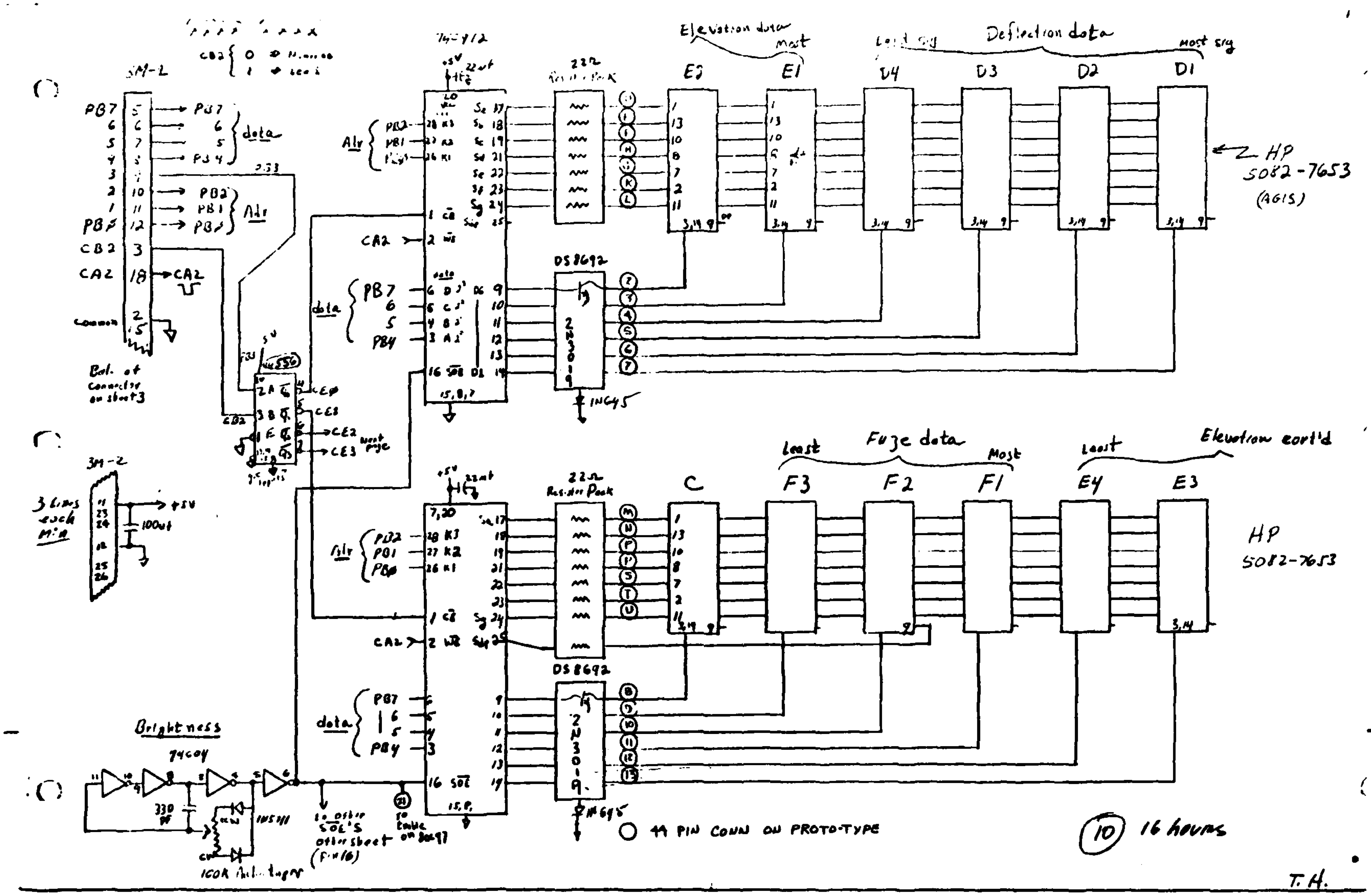
AD-A097 521 HONEYWELL INC HOPKINS MN DEFENSE SYSTEMS DIV F/G 19/6
AUTOMATED GUN LAYING SYSTEM FOR SELF-PROPELLED ARTILLERY WEAPON--ETC(U)
MAY 80 E E LEHTOLA, K A HERZING DAAA09-76-C-0284

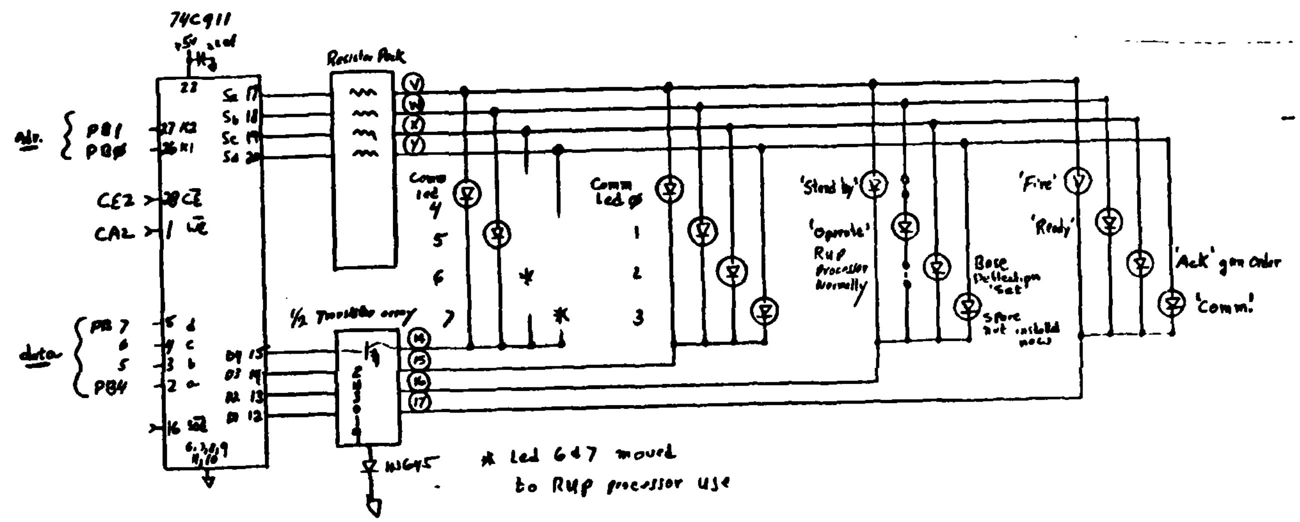
UNCLASSIFIED NL

4 OF 5
AD
A097521

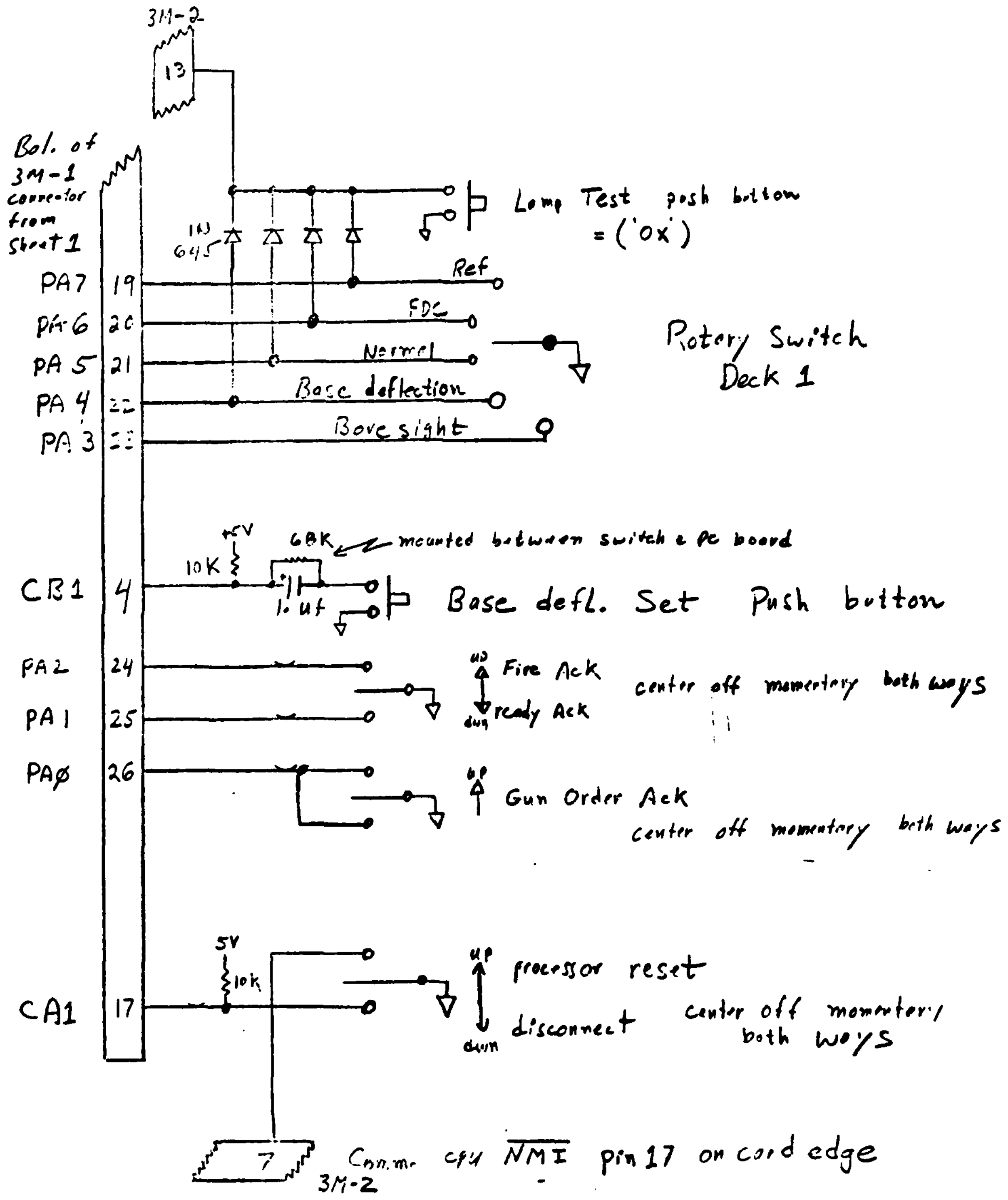

1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

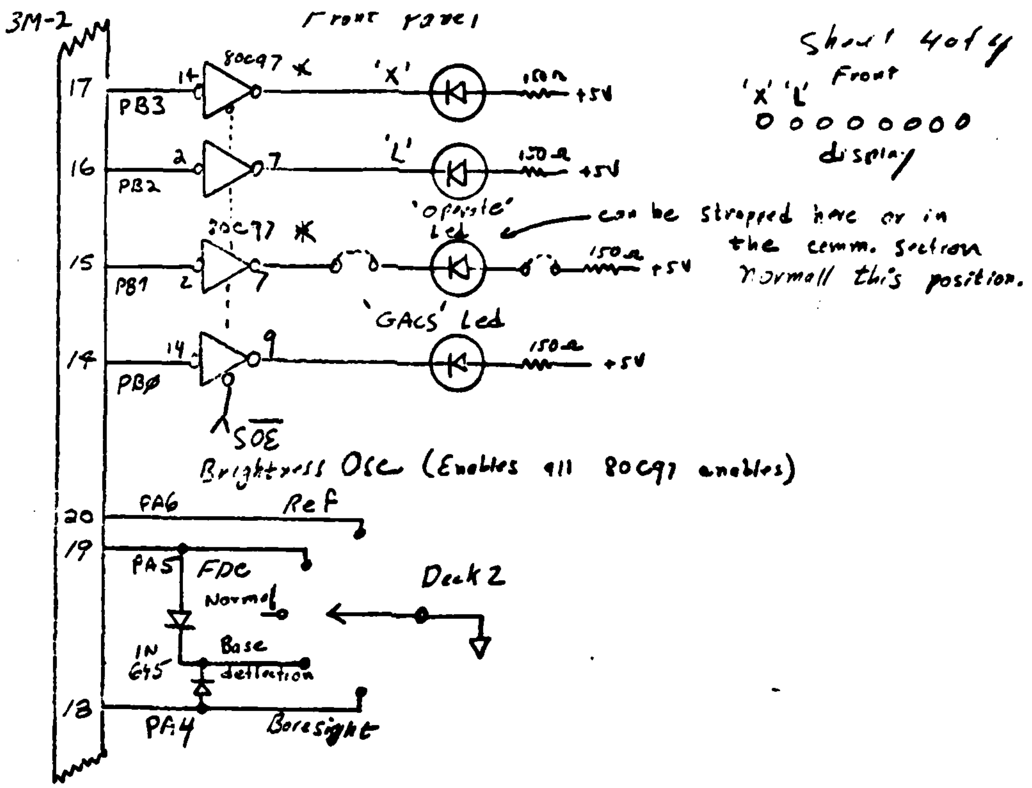

H10 Serial Interface / Modem
(pc board drawing)
Jumper wire Address
Address used
#1 C100 comm
#2 C080
wire added
2400 baud
300 baud from cpu
300 baud
Baud rate From CPU
Xmt filter 2076300
RCV Filter 2076400
ØDBM out
3M
Carrier det
Xmt Rly
* open or originate mode for FDC only
Note
Analog Supplies ; The bus must be cut for line W & hence the modem boards should be placed at one end of the bus.
digital Supplies
5 12 hr

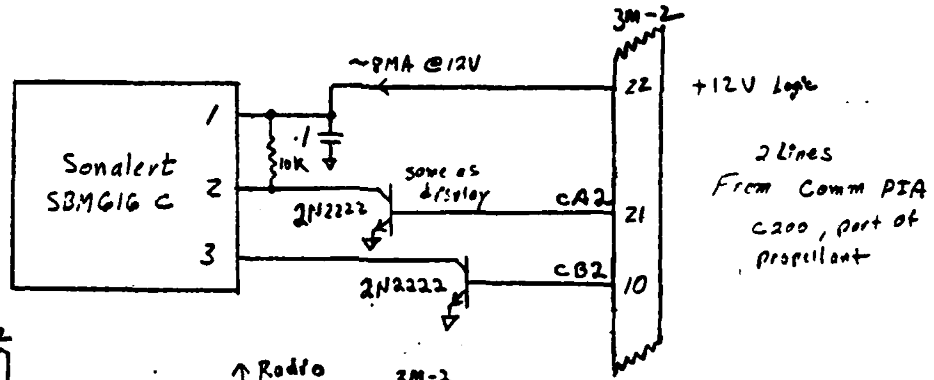

3M-2
Bal. of Connector on sheet 3
74-912
Alr
data
CA2
22Ω Resistor Pack
DS8692
1N645
Elevation data
Most
Least sig
Deflection data
Most sig
E2
E1
D4
D3
D2
D1
HP 5082-7653 (AGIS)
C
F3
F2
F1
E4
E3
Least
Fuze data
Elevation cont'd
HP 5082-7653
Brightness
74C04
330 pF
1N5711
100K
to other SOL'S other sheet (Pin 16)
3 lines each min
100uf
+5V
44 PIN CONN ON PROTO-TYPE
16 hours
T.H.

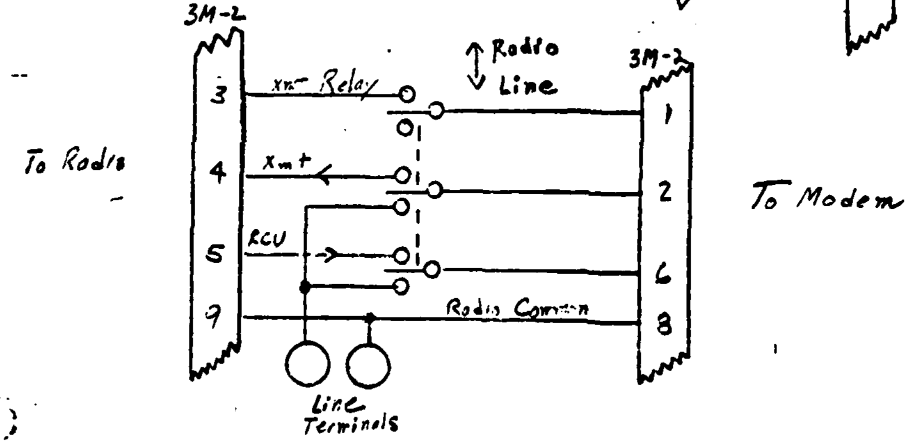

74C911
Resistor Pack
Adr.
PB1
PB∅
CE2
CA2
data
PB7
6
5
PB4
1/2 Transistor array
Comm led 4
5
6
7
Comm led ∅
1
2
3
'Stand by'
'Operate' RUP Processor Normally
Base Deflection 'Set'
Spare not installed now
'Fire'
'Ready'
'Ack' gun order
'Comm.'
* led 6 & 7 moved to RUP processor use

Front Panel Switches



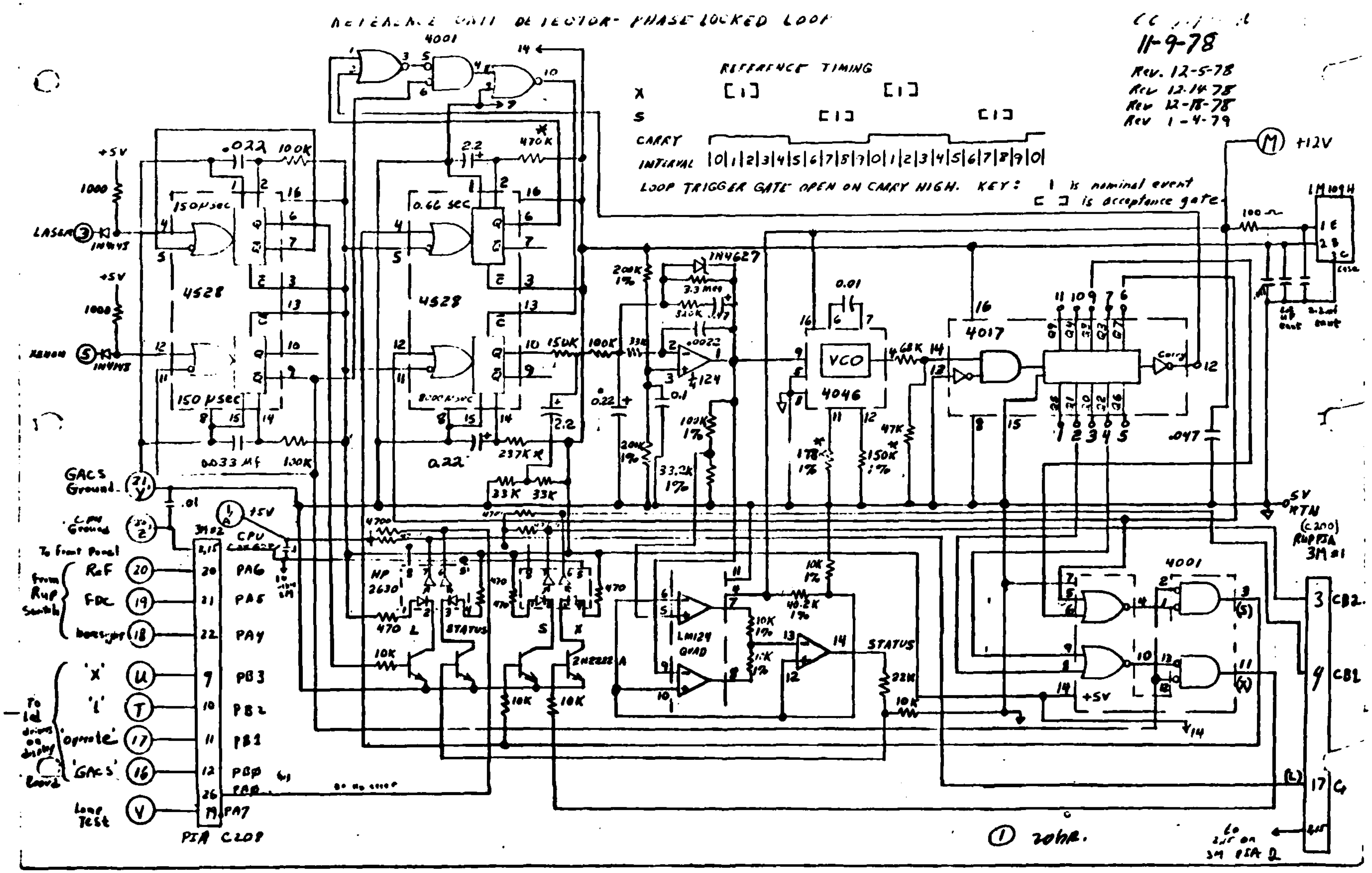






* 3 Sections each

E10



REFERENCE TIMING
X
S
CARRY
INTERVAL |0|1|2|3|4|5|6|7|8|9|0|1|2|3|4|5|6|7|8|9|0|
LOOP TRIGGER GATE OPEN ON CARRY HIGH. KEY: I is nominal event
[ ] is acceptance gate
11-9-78
Rev. 12-5-78
Rev 12-14-78
Rev 12-18-78
Rev 1-4-79
+12V
4001
4528
4017
4046
VCO
LM124 QUAD
STATUS
LASER
XENON
+5V
IN4148
IN4627
2N2222A
GACS Ground
CPU Ground
To Front Panel
From Rup Switch
RcF
FDC
To led drivers on display Board
'X'
'L'
'GACS'
Loop Test
PA6
PA5
PA4
PB3
PB2
PB1
PBØ
PA7
PIA C208
5V RTN
CB2
CB1
(1) 20hr.

E11

RUP PIA – Comm PIA

3M Cable Detail

RUP — Comm

RUP (C204)	RUP Side 3M		Comm Side	Comm (C208)
7	19	→	19	7
6	20	→	20	6
5	21	→	21	5
4	22	→	22	4
3	23	→	23	3
2	24	→	24	2
1	25	→	25	1
PAØ	26	→	26	PAØ
7	5	←	5	7
6	6	←	6	6
5	7	←	7	5
4	8	←	8	4
3	9	←	9	3
2	10	←	10	2
1	11	←	11	1
PBØ	12	←	12	PBØ
Data Accepted In * CA1	17	←	18	CA2 * Data Accepted Out
Data Ready Out * CA2	18	→	17	CA1 * Data Ready In
Data Ready In * CB1	4	←	3	CB2 * Data Ready Out
Data Accepted Out * CB2	3	→	4	CB1 * Data Accepted In
Com	2	—	2	com
Com	15	—	15	com
NC (not used RUP) +5V	1	Brown X	1	+5V
NC (not used RUP) +5V	14	Red X	14	+5V
NC	13	green X	13	NC

4¢/6' /line
= .07 $/ft

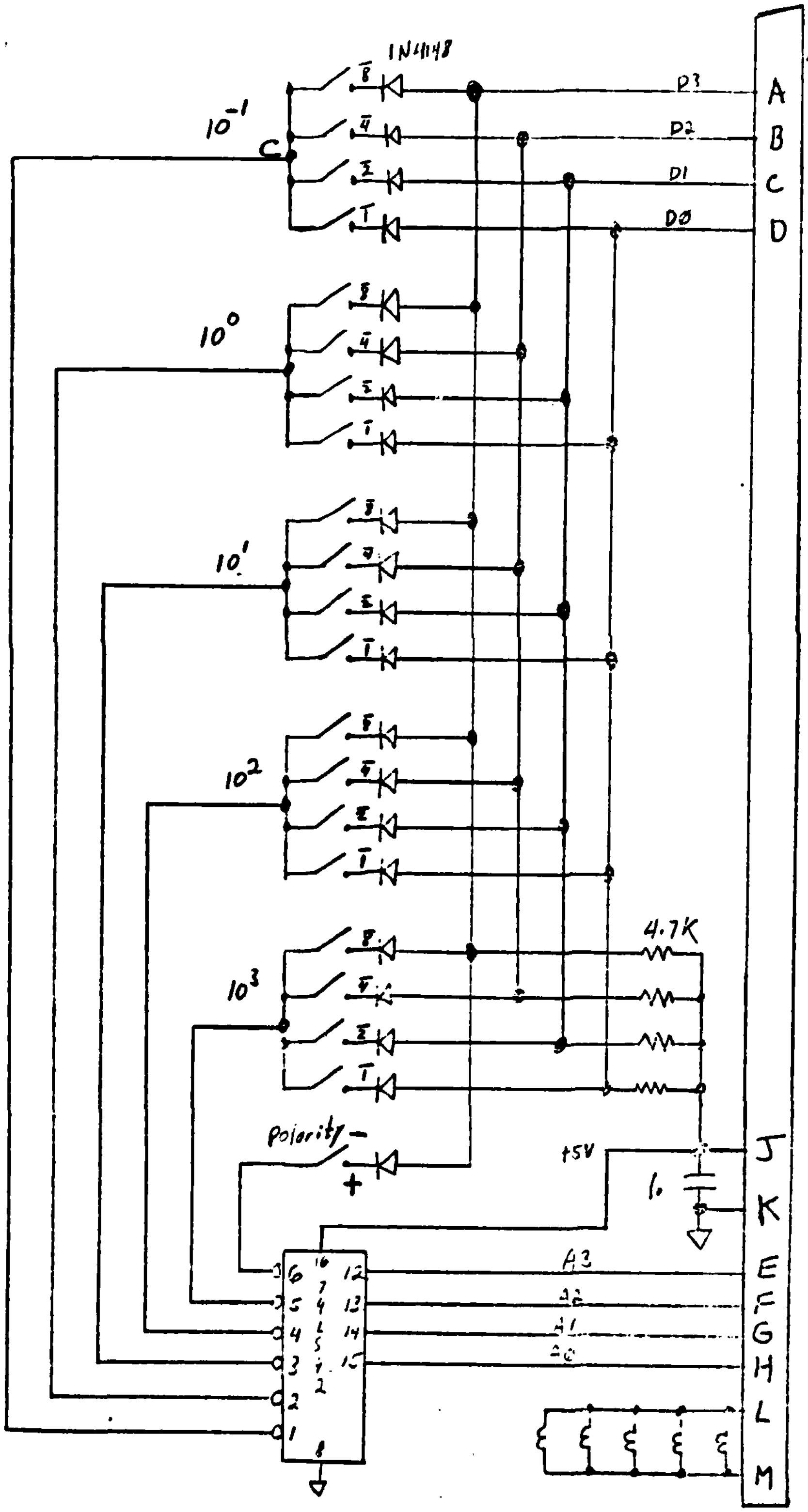
* A change exists in the cable, otherwise all other lines are straight through

x => break these lines

7 2½ hr

Temperature Thumbwheel Interface

Deliverable Unit



PA3
2
1
PAØ

Layout

Pol.	10^3	10^2	10^1	10^0	10^{-1}
+ −	☐	☐	☐	☐ .	☐

Addressing Code

PA3	PA2	PA1	PAØ	digit Se
0	0	0	0	10
0	0	0	1	10
0	0	1	0	10
0	0	1	1	10
0	1	0	0	10
0	1	0	1	po

All data in is pos. logic

+5V

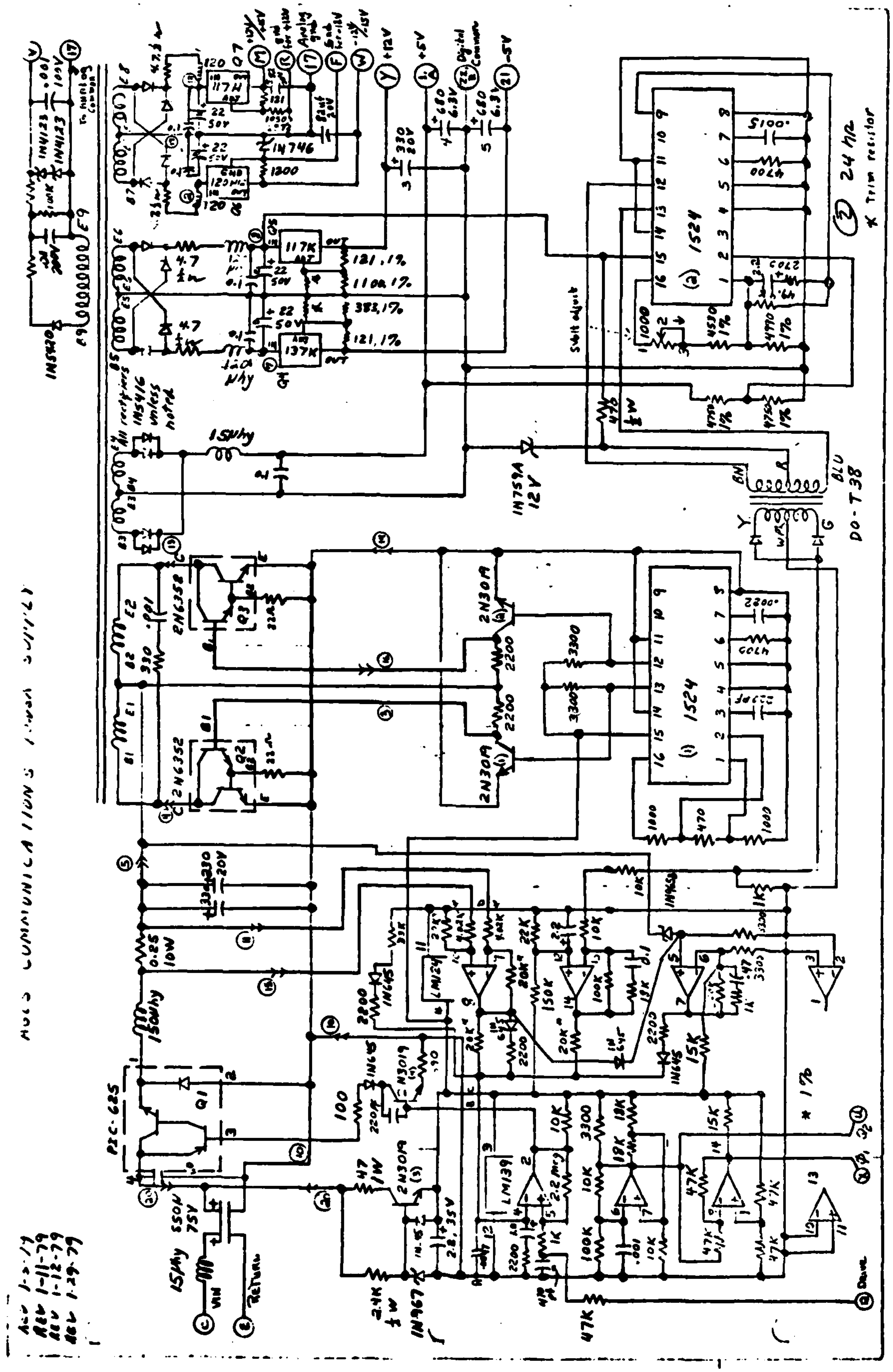
common

PB3
2
1
PBØ

Address is positive logic

24V, lamp

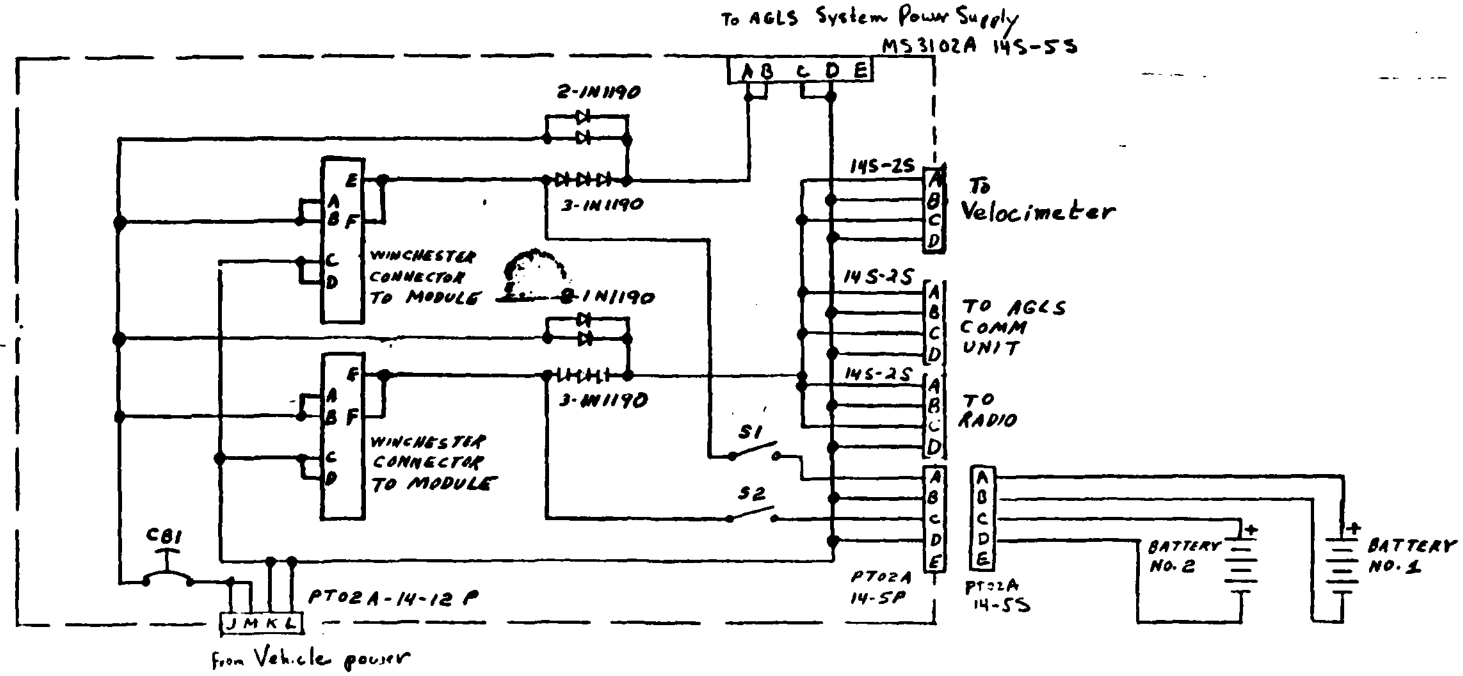
E 13
T.H.

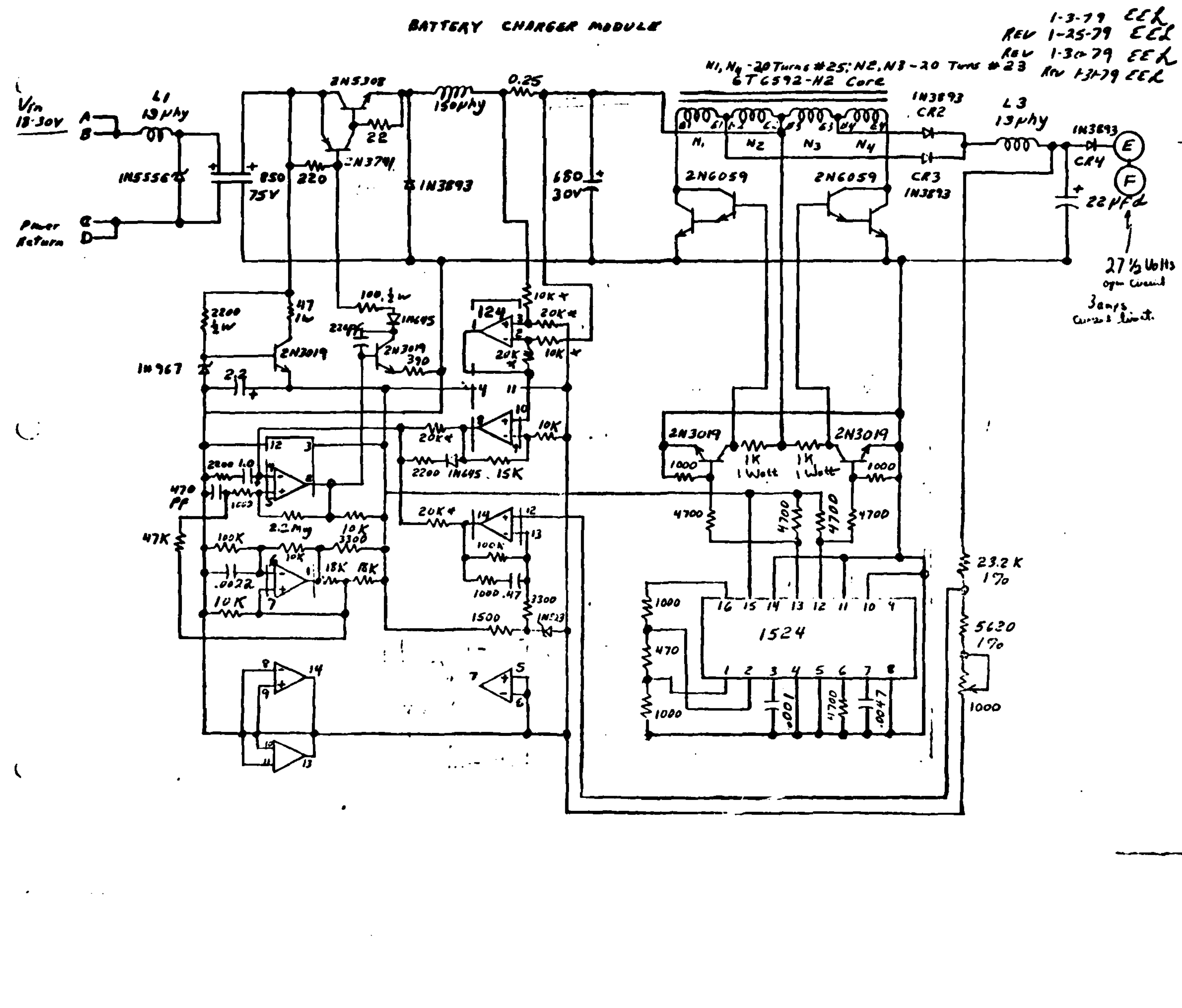


B14

AGLS BATTERY CHARGER

1-3-79
REV 1-30-79




BATTERY CHARGER MODULE
1-3-79 EEL
REV 1-25-79 EEL
REV 1-30-79 EEL
Rev 1-31-79 EEL
N1, N4 - 20 Turns #25; N2, N3 - 20 Turns #23
6TC592-H2 Core
Vin 18-30V
A
B
Power Return
C
D
L1 13μhy
1N5556
850 75V
2N5308
22
220
2N3791
150μhy
1N3893
0.25
680 30V
N1
N2
N3
N4
2N6059
2N6059
1N3893 CR2
CR3 1N3893
L3 13μhy
1N3893 CR4
E
F
22μFd
27½ Volts open circuit
3 amps current limit
2200 ½W
47 1W
2N3019
1N967
2.2
100, ½W
1N645
2N3019
390
124
10K
20K
15K
2200
2N3019
1K 1 Watt
1000
4700
1524
23.2K 1%
5620 1%
1000
470
.001
.0047
.0022
100K
2.2Meg
3300
18K
47K
470 PF
1N523

APPENDIX F

VECOM CONTROL PROGRAM
SOURCE LISTING

```
IAM VECOM

 AGLS COMMUNICATIONS
 VEHICLE
 REVISED 3/3/79 2000


LAGC EQU 1


 PIA EQUATES
 PIA1=PROP. TEMPERATURE
 A SIDE=INPUTS
 B SIDE =OUTPUTS(ADDR)
IA1DA EQU $C200
IA1DB EQU PIA1DA+1
IA1CA EQU PIA1DA+2
IA1CB EQU PIA1DA+3
N1 EQU PIA1DA
T1 EQU PIA1DB
ORNON EQU $3E
ORNOF EQU $36
*
*
* PIA2=CONTROLS AND DISPLAYS
* A SIDE=SW. INPUTS
* B SIDE=DATA/ADDR. OUTPUTS
*
IA2DA EQU $C204
IA2DB EQU PIA2DA+1
IA2CA EQU PIA2DA+2
IA2CB EQU PIA2DA+3
N2 EQU PIA2DA
T2 EQU PIA2DB
*
* PIA3=REF. UNIT PROC.
* A SIDE-INPUT
* B SIDE=OUTPUT
*
IA3DA EQU $C208
IA3DB EQU PIA3DA+1
IA3CA EQU PIA3DA+2
IA3CB EQU PIA3DA+3
N3 EQU PIA3DA
T3 EQU PIA3DB

* PIA4=NOT ASSIGNED
*
IA4DA EQU $C20C
IA4DB EQU PIA4DA+1
IA4CA EQU PIA4DA+2
IA4CB EQU PIA4DA+3
N4 EQU PIA4DA
T4 EQU PIA4DB
*
* SWITCH MASKS (PIA2A)
FOSM EQU %00000001 FIRE ORDER
```



```
RRSM EQU %00000010 READY RESPONSE
FASM EQU %00000100 FIRE ACK
BSTM EQU %00001000 BORESIGHT
BDFM EQU %00010000 BASE DEFL.
NMLM EQU %00100000 NORMAL
FDCM EQU %01000000 FDC
REFM EQU %10000000 REF ANGLE
LMTM EQU %11110000 LAMP TEST
*
*
* SWITCH WORDS (PIA2B)
FOAWD EQU 1 FIRE ORDER ACK
FCAWD EQU 2 FIRE COMMAND ACK
RRAWD EQU 4 READY REQUEST ACK
CFAWD EQU 8 CHECK FIRE ACK
*
* LED WORDS (PIA2)
* ADDRESS-DATA
*
COMM EQU $0008 COMM.
GUACK EQU $0004 GUN ORDER
REDY EQU $0002 READY
FIRE EQU $0001 FIRE
BDSET EQU $0040 BASE DEF. SET
OPRAT EQU $0020 OPERATE
STBY EQU $0010 STANDBY
CLED3 EQU $0088 CARRIER DET.
CLED2 EQU $0084 CJ=3 STATUS
CLED1 EQU $0082 CJ=2 STATUS
CLED0 EQU $0081 CJ=1 STATUS
CLED5 EQU $00A0 NAK
CLED4 EQU $0090 AGLS BUSY
*
*
* PERHIPERAL EQUATES
*
* COMM ACIA
COMC EQU $C100
COMS EQU COMC
COMX EQU COMC+1
COMR EQU COMC+1
*
* AGLS ACIA
AGC EQU $C102
AGS EQU AGC
AGX EQU AGC+1
AGR EQU AGC+1
*
* TEST PORT ACIA
TPC EQU $9808
TPS EQU TPC
TPX EQU TPC+1
TPR EQU TPC+1
*
* FUZE SETTER ACIA
*
FSC EQU $C080
FSS EQU FSC
```

```
PAGE 003  PVECOM  .SA:1

FSX EQU FSC+1
FSR EQU FSC+1
*
* VELOCIMETER ACIA
*
VLC EQU $C082
VLS EQU VLC
VLX EQU VLC+1
VLR EQU VLC+1
*
* TIMER
TCR13 EQU $9800
TSTS EQU $9801
TCR2 EQU $9801
T1D EQU $9802
T2D EQU $9804
T3D EQU $9806
* TIMER CONSTANTS
T1IE EQU %01000000
T1II EQU %00000000
*
T2IE EQU %01000001
T2II EQU %00000001
*
T3IE EQU %11000011
T3II EQU %10000011
*
* TIMEOUT CONSTANTS (.1 SECONDS)
CONTT EQU 0100 CONNECT TRY
CUD EQU 0007 CARRIER UP DELAY
CDD EQU 0005 CARRIER DN DELAY
CDT EQU 36000 CARRIER DETECT
WT EQU 0300 WAIT
* COMM EQUATES
*
SOH EQU 1 START OF HEADER
SMFC EQU $42 SERVICE MSG FORMAT CODE
IMFC EQU $48 INFO. MSG FORMAT CODE
SLFC EQU $43 SELECT MSG. FORMAT CODE
SC EQU $41 SEQUENCE CODE
AC EQU $40 ADDRESS CODE
IC EQU $40 IDENT CODE
STX EQU 2 START OF TEXT
ETX EQU 3 END OF TEXT
NOC EQU $40 NO REQ OPERATION CODE
SMTY EQU $40 SERVICE MSG. TYPE
SLTY EQU $42 SELECT MSG TYPE
DOC EQU $42 DATA REQ OPER CODE
* ACIA INTERRUPT CONSTANTS
XIE EQU %00101001 XMIT INT ENB
RIE EQU %10001001 RECV INT ENB
NIE EQU %00001001 INT. OFF
*
*
 IFNE FLAGC
*
*
* COMM INTERRUPT CONSTANTS
```

```
PAGE 004  PVECOM  .SA:1

CNIE EQU %01001001
CRIE EQU %11001001
CXIE EQU %00101001
RTS EQU %00001001
 ENDC
*
*
 IFEQ FLAGC
*
*
CNIE EQU %00001001
CRIE EQU %10001001
CXIE EQU %00101001
RTS EQU %00001001
 ENDC
 PAGE
 ORG $1000
* COMM RECEIVE BUFFER
RBUF RMB 60
REND EQU *
RDATA EQU RBUF+7
* COMM TRANSMIT BUFFER
XBUF RMB 60
XEND EQU *
XDATA EQU XBUF+7
* AGLS "FROM" BUFFER
AGLF EQU *
 RMB 5 ELEV. COMMAND
 RMB 5 ELEV. ACTUAL
 RMB 5 ELEV. ERROR
 RMB 5 AZ. COMMAND
 RMB 5 AZ. ACTUAL
 RMB 5 AZ. ERROR
 RMB 5 ACTIVE EL CMND
 RMB 5 ACTIVE AZ CMND
 RMB 1 COMM MODE
 RMB 2 LEVEL STATUS
 RMB 1 AGLS MODE
 RMB 1 LOCAL MODE
AGFE EQU *
* AGLS "TO" BUFFER
AGLT RMB 5 ELEVATION
 RMB 5 AZIMUTH
 RMB 1 MODE
AGTE EQU *
*
* DISPLAY BUFFER
DISBUF EQU *
 RMB 4 DEFLECTION
 RMB 2 ELEVATION
 RMB 2 DUMMY
 RMB 2 ELEVATION
 RMB 3 FUZE
 RMB 1 CHARGE
DISEND EQU *
*
* VELOCITY BUFFER
*
```

F2

```
VELBUF RMB 10
*
*
XIDLE RMB 2 IDLE VECTOR FLAG
*
*
* FLAGS
FLAG EQU *
ETXF RMB 1 END TEXT FLAG
CJ RMB 1 COMM STEERING
ZRFD RMB 1 REQ. DISC. FLAG
ZIDM RMB 1 ID MESSAGE FLAG
XPASS RMB 1 XMIT FIRST PASS
RJ RMB 1 RECD DATA FLAG
RLSC RMB 1 LAST RECD SEQ CODE
IDLEF RMB 1 IDLE FLAG
OBJ RMB 1 WAIT FLAG
CBSY RMB 1 CRT BUSY
RAG RMB 1 READ AGLS
WAG RMB 1 WRITE AGLS
ZRRF RMB 1 READY FOR RESP
VDF RMB 1 VALID DATA FLAG
CONN RMB 1 CONNECT FLAG
DAF RMB 1 DATA AVAIL FLAG
OUTF RMB 1 WRITE FDC GET OUT
PASS RMB 1 READ FDC FIRST PASS
FILLF RMB 1 FILL CHAR FLAG(PXMT)
VERF RMB 1 AGLS VERIFY
CTSUBY RMB 1 CLR TO SEND UP BUSY
CTSDBY RMB 1 CLR TO SEND DN BUSY
TDCDBY RMB 1 CARRIER DET. BUSY
WAITF RMB 1 WAIT DUE FLAG
FEND EQU *
* CONSTANTS
BEND RMB 2 BUFFER END POINTER
* BUFFER POINTERS
BRI RMB 2 RECEIVE
BXI RMB 2 TRANSMIT
AGXX RMB 2 AGLS TRANSMIT
AGRR RMB 2 AGLS RECEIVE
XBCC RMB 1 BCC XMIT
RBCC RMB 1 BCC RECEIVE
RERR RMB 1 RECV ERROR CODE
RSTAT RMB 1 RECV STATUS WORD
TMPX RMB 1 RJ TEMP INDEX
RXFC RMB 1 RECD FORMAT CODE
RXOC RMB 1 RECD OPER CODE
OLDSC RMB 1 SEQUENCE CODE SAVE
SAVES RMB 2 X REG SAVE (INT)
ISAVES RMB 2 INT STACK SAVE
SAVEX RMB 2 X REG SAVE
SAVA RMB 1 SAVE A REG
SAVB RMB 1 SAVE B REG
OLDCR2 RMB 1 TIMER CR#2 WORD
VECT1 RMB 2 TIMER INT VECTOR 1
VECT2 RMB 2 TIMER INT VECTOR 2
APTR RMB 2 AGLS BUFFER POINTER
TPPTR RMB 2 TEST PORT POINTER
```



```
TPPTE RMB 2 TEST PORT END
SPC RMB 1 SPACE COUNT
ASP RMB 1 SPACE COUNT
RSWD RMB 1 RECD STATUS WORD (4CHAR)
ASWD RMB 1 ACK STATUS WORD
TRY RMB 1 CONNECT TRIES
DISADD RMB 1 DISPLAY ADDRESS
DEST RMB 2 CHAR XFER DESTINATION
AGTRY RMB 1 AGLS DATA TRIES
OLDSW RMB 1 MODE SWITCH SAVE
LEDWD RMB 2 CURRENT LED STATUS
CEND EQU *
*
* INTERRUPT DRIVEN TIMERS (100 MSEC)
*
* TIMER TABLE (DECREMENT)
TMTB EQU *
TF1 RMB 1
TIM1 RMB 2
*
TF2 RMB 1
TIM2 RMB 2
*
TF3 RMB 1
TIM3 RMB 2
*
TF4 RMB 1
TIM4 RMB 2
*
TF5 RMB 1
TIM5 RMB 2
 PAGE
*
* AGLS VEHICLE COMM PROCESSOR
*
* START VECTOR FOR POWER UP OR RESET
*
 ORG $E800
STRT EQU *
* H-10 DEBUG
 LDS #$7F
* SETUP PIAS
 JSR PIAS
* CLEAR BUFFERS
* DATA BUFFERS
 LDX #DISEND
 STX BEND
 LDX #RBUF
 JSR CLBF
* FLAG BUFFER
 LDX #FEND
 STX BEND
 LDX #FLAG
 JSR CLBF
* CONSTANT BUFFER
 LDX #CEND
 STX BEND
 LDX #BRI
```

```
 JSR CLBF
* SET TRY COUNT
 LDA A #10
 STA A TRY
* SEED SEQ. CODE
 LDA A #SC
 STA A OLDSC
* CLEAR TIMERS
 JSR CLTM
* MAKE SURE AGLS THERE
 JSR RUTHER
* ENABLE INTERRUPTS
 CLI
* INITIAL CONNECT
ICONX JSR ICON
* CONNECT PROCESSOR
 JSR CONP
*
*
* SYSTEM ACTIVE LOOP
*
LOOP EQU *
* TEST BASE DEFLECTION
 JSR TBD
* RUP REQUEST TIMER (TF3)
 TST TF3
 BNE LOOP1
*
 LDX #10
 STX TIM3
 INC TF3
* REQUEST RUP OUTPUT
 JSR REQRUP
* READ RUP
 JSR RRUP
* AGLS BUSY?
 TST RAG
 BNE LOOP1
* TEST RUP MODE SW
 LDA A IN2
* EXCLUDE LAMP TEST
 BIT A #$F0
 BEQ LOOP1
* MASK TO SIGNIFICANT DATA
 AND A #%00011000
 CMP A OLDSW
 BNE LOOP9
* TEST SET/CLR PB
 TST PIA2CB
 BPL LOOP1
* MODE OK?
 EOR A #%00011000
 BNE LOOP4
 LDA A PIA2DB
 BRA LOOP1
LOOP9 STA A OLDSW
* WRITE TO AGLS
LOOP4 JSR COMAGL
```



```
* TEST IDLE FLAG
LOOP1 TST IDLEF
 BEQ LOOP3
* IDLE FLAG SET
 LDX #*
 JMP IDLE
* TEST PROCESS COMM
LOOP3 LDA A CJ
 CMP A #1
 BNE LOOP5
 LDA A #CLED0
 JSR LEDON
 JSR CLRL12
 BRA LOOP
LOOP5 CMP A #2
 BNE LOOP6
 LDA A #CLED1
 JSR LEDON
 JSR CLRL02
 BRA LOOP
LOOP6 CMP A #3
 BNE LOOP
* PROCESS RECD DATA
 LDA A #CLED2
 JSR LEDON
 JSR CLRL01
 JMP C3
*
* POWER UP ENTRY
*
PWRUP EQU *
 LDS #$7F
* CLEAR BUFFERS AND FLAGS
 LDX #FEND
 STX BEND
 LDX #RBUF
 JSR CLBF
 JSR CLTM
* INITIAL PIAS
 JSR PIAS
* MAKE SURE AGLS THERE
 JSR RUTHER
 LDX #0
* ENABLE FOR TEST PORT
 CLI
*
* COMM IDLE LOOP
IDLE EQU *
 STX XIDLE
 LDX #0
 STX LEDWD
 CLR CONN
* INHIBIT TIMER, COMM & AGLS INT
 LDA A #NIE
 STA A AGC
*
 JSR CLR1
 JSR CLR2
```

F4

```
IDLE1 EQU *
* TEST BASE DEFLECTION
 JSR TBD
* RUP REQUEST TIMER (TF3)
 TST TF3
 BNE IDLE2
*
 LDX #10
 STX TIM3
 INC TF3
*
* REQUEST RUP OUTPUT
 JSR REQRUP
* READ RUP
 JSR RRUP
* AGLS BUSY?
 TST RAG
 BNE IDLE2
* TEST RUP MODE SW
 LDA A IN2
* EXCLUDE LAMP TEST
 BIT A #$F0
 BEQ IDLE2
* MASK TO SIGNIFICANT DATA (BD,BS,NOR)
 AND A #%00011000
 CMP A OLDSW
 BNE IDLE4
* TEST CLR/SET PB
 TST PIA2CB
 BPL IDLE2
* MODE OK?
 EOR A #%00011000
 BNE IDLE3
 LDA A PIA2DB
 BRA IDLE2
IDLE4 STA A OLDSW
* WRITE TO AGLS
IDLE3 JSR COMAGL
* CLEAR COMM ACIA
IDLE2 LDA A #$43
 STA A COMC
* LITE "IDLE"IND.
 JSR IDLL
 JMP IDLE1
 PAGE
* SUBROUTINES
*
*
* CLEAR BUFFER ROUTINE
* X= BUFFER START
* BEND=BUFFER END
*
*
CLBF CLR 0,X
 INX
 CPX BEND
 BNE CLBF
 RTS
```



```
*
* PROCESS TRANSMIT
*
PXMT LDX BX1
 TST FILLF
 BGT PXM4
 BMI PXM5
 LDA A 0,X
 BEQ PXM1
 INX
 STX BX1
*=SOH?
 CMP A #SOH
 BNE PXM2
* YES, =SOH
 CLR XBCC
 BRA PXM3
* = DATA
PXM2 TAB
 EOR B XBCC
 STA B XBCC
* TRANSMIT CHAR
 JSR XMIT
PXM3 CLC
 RTS
* LAST CHAR
PXM1 LDA A XBCC
 JSR XMIT
 INC FILLF
 CLC
 RTS
* TRANSMIT FILL CHAR
PXM4 LDA A #$20
 JSR XMIT
 NEG FILLF
 CLC
 RTS
* SECOND TIME AROUND
PXM5 LDA A #$20
 JSR XMIT
 CLR FILLF
 LDX #XBUF
 STX BX1
 SEC
 RTS
*
* PROCESS RECEIVE
*
PREC JSR RECV
 BCS PRE6
* CLEAR XMIT FLAG
 CLC
 RTS
PRE6 LDX BR1
* CHAR = BCC?
 INC ETXF
 BEQ PRE1
* NO, = DATA
```

```
 CLR ETXF
 STA A 0,X
 INX
 STX BR1
* TEST RECIEVE STATUS
 JSR JSTS
* TEST BUFFER OVERRUN
 JSR BOVR
* CHAR = SOH?
 CMP A #SOH
 BEQ PRE2
* CHAR = ETX?
 CMP A #ETX
 BNE PRE3
 DEC ETXF
PRE3 EOR A RBCC
 STA A RBCC
 CLC
 RTS
* FIRST CHAR
PRE2 CLR RBCC
 CLR RERR
 LDX #RBUF
 STA A 0,X
 INX
 STX BR1
 INC VDF
 CLC
 RTS
* LAST CHAR
PRE1 EOR A RBCC
 BNE PRE4
PRE5 LDX #RBUF
 STX BR1
 CLR VDF
 SEC
 RTS
* BCC ERROR
PRE4 LDA A #$10
 EOR A RERR
 STA A RERR
 BRA PRE5
* RECEIVE CHAR ROUTINE
*
RECV LDA A COMS
 BIT A #1
 BNE RECV1
* NO DATA CALL
 LDA A COMR
 CLC
 RTS
* DATA CALL
RECV1 STA A RSTAT
* READ CHAR
 LDA A COMR
 SEC
 RTS
*
```



```
* TEST RECEIVE STATUS
*
JSTS PSH A
 LDA A RSTAT
* PARITY ERROR?
 BIT A #$40
 BEQ JSTS1
 LDA B #$18
 EOR B RERR
 STA B RERR
* OVER RUN ERROR?
JSTS1 BIT A #$20
 BEQ JSTS2
 LDA B #8
 EOR B RERR
 STA B RERR
* FRAMING ERROR
JSTS2 BIT A #$10
 BNE JSTS3
 PUL A
 RTS
JSTS3 LDA B #$20
 EOR B RERR
 STA B RERR
 PUL A
 RTS
*
* TRANSMIT CHAR.
*
XMIT LDA B COMS
 BIT B #2
 BNE XMIT1
* NO DATA CALL, RESET RECVR
 LDA A COMR
 RTS
* DATA CALL
XMIT1 STA A COMX
 RTS
*
* INHIBIT XMIT/ENB RECV
*
COMIX LDA A #CRIE RECV INT ENB
 STA A COMC
 LDA A COMR RESET
 RTS
*
* INHIBIT COMM INT
COMOFF LDA A #CNIE
 STA A COMC
 LDA A COMR
 RTS
*
* INHIBIT RECV/ENB XMIT
*
* PREP COMM
COMIR EQU *
* MAKE SURE RECD CAR DN
COMI3 LDA A COMS
```

F6

```
 BIT A #4
 BEQ COMI3
 LDA A #RTS
 STA A COMC
* WAIT CARRIER UP
COMI1 JSR CTSU
 BCC COMI1
 LDA A #SOH
COMI2 LDA B COMS
 BIT B #2
 BEQ COMI2
 STA A COMX
 LDA A #CXIE XMIT INT ENABLE
 STA A COMC
 RTS
*
* TEST BUFFER OVERRUN
*
BOVR CPX #REND
 BEQ BOVR1
 RTS
* OVERRUN HAS OCCURRED
BOVR1 LDA B #8
 EOR B RERR
 STA B RERR
 DEX
 STX BR1
 RTS
* COMMUNICATION POLL
*
CPOLL LDA A CJ
 CMP A #1
 BEQ CPOL1
 CMP A #2
 BEQ CPOL2
* RESET INTERRUPT
 LDA A COMR
 RTS
* TRANSMIT LOOP
CPOL1 EQU *
* FIRST PASS?
 INC XPASS
 BNE CPOL6
* SETUP MESSAGE HEADERS
CPOL11 JSR SETUM
* SWAP SEQ. CODES
 JSR SSC
* PRCESS TRANSMIT
CPOL6 CLR XPASS
 JSR PXMT
 BCS CPOL5
 RTS
* SWITCH TO RECEIVE
CPOL5 EQU *
* CLEAR RECEIVE BUFF
 LDX #RBUF
 JSR CLRB
 LDX #RBUF
```



```
 STX BR1
* SWITCH COMM. INT
 JSR COMIX
 LDA A #2
 STA A CJ
 RTS
*
* RECEIVE LOOP
CPOL2 EQU *
*
*
 IFNE FLAGC
* TEST CARRIER
 LDA A #CLED3
 JSR LEDOFF
 JSR TDCD
 BCS CPOL10
 LDA A COMR
 RTS
*
 ENDC
*
* PROCESS RECEIVE
CPOL10 LDA A #CLED3
 JSR LEDON
 JSR PREC
 BCS CPOL7
 RTS
* SWITCH TO UNPACK
CPOL7 LDA A #3
 STA A CJ
 JSR COMOFF
 LDA A #CLED3
 JSR LEDOFF
 RTS
*
* SETUP MESSAGE ROUTINE
*
SETUM EQU *
* TEST RFD FLAG
 TST ZRFD
 BEQ SETUM1
* SETUP SERVICE MESSAGE
 LDA B #$44
 JSR SSM
* TEST I.D. MESSSAGE
SETUM1 INC ZIDM
 BNE SETUM2
* SETUP SELECT
 LDA B #$42
 JSR SSLM
SETUM2 CLR ZIDM
* TEST READY FOR RESPONSE
 INC ZRRF
 BNE SETUM3
* SETUP READY FOR RESPONSE
 LDA B #$46
 JSR SIM
```

```
* SETUP XMIT POINTER
SETUM3 CLR ZRHF
 LDX #XBUF
 STX BX1
 RTS
*
* CARRIER UP DELAY
*
CTSU EQU *
 TST CTSUBY
 BNE CTSU1
* SETUP TIMEOUT VECTOR
 LDX #CUD
 STX TIM2
 INC TF2
 INC CTSUBY
* TEST CTS-UP
CTSU1 TST TF2
 BEQ CTSU2
 CLC
 RTS
* INHIBIT TIMER
CTSU2 CLR CTSUBY
 SEC
 RTS
*
* CARRIER DOWN DELAY
CTSD EQU *
 TST CTSDBY
 BNE CTSD1
* SETUP TIMEOUT VECTOR
 LDX #CDD
 STX TIM2
 INC TF2
 INC CTSDBY
* TEST CTS DOWN
CTSD1 TST TF2
 BEQ CTSD2
 CLC
 RTS
* INHIBIT TIMER
CTSD2 CLR CTSDBY
 SEC
 RTS
*
* TEST CARRIER DETECT
*
TDCD EQU *
 TST TDCDBY
 BNE TDCD2
* SETUP TIMEOUT VECTOR
 LDX #IDLE1
 STX VECT1
 LDX #*
 STX XIDLE
 LDX #(CDT/2-1)
 JSR SET1 CARRIER DETECT TIME
 INC TDCDBY
```



```
* TEST DCD UP
TDCD2 LDA A COMS
 BIT A #$4
 BEQ TDCD1
 LDA A COMR
 CLC
 RTS
* INHIBIT TIMER
TDCD1 JSR CLR1
 CLR TDCDBY
 SEC
 RTS
*
* UNPACK ROUTINE (CJ=3)
*
C3 EQU *
* TEST ERROR FLAG
 TST RERR
 BNE C36
* UNPACK RECEIVED DATA
 LDX #RBUF
 LDA A 0,X SOH
 ADD A 6,X STX
* SOH + STX OK?
 CMP A #3
 BEQ C31
* NO
 LDA B #$28
 EOR B RERR
 STA B RERR
C36 JMP R0
* TEST STX+1 (DATA/NO-DATA)
C31 CLR B
 LDA A 7,X
 CMP A #3
 BEQ C32
 LDA B #4
C32 STA B RJ
* TEST OPER. CODE
 CLR B
 LDA A 4,X
 AND A #$38 MASK ACK/NAK
 BNE C33
 LDA B #2
C33 EORB RJ
 STA B RJ
* TEST SEQUENCE CODE
 LDA A 2,X
 CMPA RLSC
 BEQ C34
 LDA B #1
 EOR B RJ
 STA B RJ
C34 STA A RLSC
* FIX RJ TO INDEX
 LDA A RJ
 AND A #7
 STA A TMPX
```

```
* BRANCH IF DATA
 CMP A #4
 BLT C35
 JSR PDAT
* BRANCH TO PROCESS ACK/NAK
C35 LDA B TMPX
 LDX #RTBL
 JSR FIXX
 JMP 0,X
*
* RECEIVE RESPONSE TABLE
* RJ=0 NO-DATA NAK OLD-SC
* RJ=1 NO-DATA NAK NEW-SC
* RJ=2 NO-DATA ACK OLD-SC
* RJ=3 NO-DATA ACK NEW-SC
* RJ=4 DATA NAK OLD-SC
* RJ=5 DATA NAK NEW-SC
* RJ=6 DATA ACK OLD-SC
* RJ=7 DATA ACK NEW-SC
*
* ACK/NAK TABLE
RTBL JMP R0 RI=0 NAK
 JMP R0 1 NAK
 JMP R2 2 ACK
 JMP R2 3 ACK
 JMP R0 4 NAK
 JMP R5 5 PROCESS
 JMP R6 6 RESPOND ONLY
 JMP R5 7 PROCESS
*
* FIX POINTER BY INDEX
FIXX TST B
 BNE FIXX1
 RTS
FIXX1 INX
 INX
 INX
 DEC B
 BNE FIXX1
 RTS
*
* PROCESS DATA MESSAGE
*
PDAT LDX #RBUF
 LDA A 1,X
 CMP A #$44
 BNE PDAT1
* DATA = MESSAGE FOR CRT
 JSR XFRMSG
 LDA A #6
 STA A TMPX
 RTS
* DATA TO BE PROCESSED
PDAT1 LDA A RJ
 CMP A #5
 BEQ PDAT2
 CMP A #7
 BEQ PDAT2
```



```
 RTS
* TRANSFER DATA FROM REC BUFFER
PDAT2 JSR TRAN
* SET DATA FLAG
 CLR DAF
 INC DAF
 RTS
*
* ACK/NAK PROCESSING, RJ= 0,1,4
*
R0 LDA A #CLED5
 JSR LEDOFF
 TST RERR
 BEQ R01
* SETUP NAK RESPONSE
 LDA A #CLED5
 JSR LEDON
 JSR SNR
* SWAP SEQ CODES
R01 JSR SSC
* RESET FOR TRANSMIT
XIT EQU *
* TEST CONNECT
 TST CONN
 BNE XIT1
 JMP CONT
XIT1 LDA A #1
 STA A CJ
 LDA A #$FF
 STA A XPASS
* ENB XMIT/INH RECV
 JSR COMIR
 JMP LOOP
* RJ = 2,3
R2 LDA A #CLED5
 JSR LEDOFF
 LDX #RBUF
* STRIP FORMAT CODE
 LDA A 1,X
 STA A RXFC
* STRIP OPER CODE
 LDA A 4,X
 AND A #7
 STA A RXOC
** TEST SERVICE MSG
 LDA A RXFC
 CMP A #SMFC
 BEQ R24
* TEST RANGE OF FORMAT CODE
 BLS R2ER
 CMP A #$4D
 BHI R2ER
* TEST OPER CODE
* NO REQ OC?
 LDA A RXOC
 CMP A #0
 BNE R21
* TEST IF DATA ACK
```

```
 TST DAF
 BNE R22
* SET READY RESP.
 CLR ZRRF
 DEC ZRRF
R22 CLR DAF
* SETUP INFO MESSAGE
 LDA B #$40
 JSR SIM
 JMP XIT
* TEST IF DATA REQUEST
R21 CMP A #2
 BNE R2ER
* SETUP DR ACK WORD
 LDA A #$0F
 STA A ASWD
* SETUP DATA MESSAGE HEADER
 LDA B #$40
 JSR SDMH HEADER
* SETUP DATA REPORT
 JSR SDR
* SETUP DATA REPORT TRAILER
 JSR SDRT
* SET DATA FLAG
 CLR DAF
 INC DAF
 JMP XIT
*
* PROCESS SERVICE MESSAGES
* TERMINATE
R24 LDA A RXOC
 CMP A #$3
 BNE R25
* SET IDLE FLAG
 CLR IDLEF
 INC IDLEF
R26 LDA B #$40
 JSR SSM
 JMP XIT
* SELECT?
R25 CMP A #$2
 BEQ R26
* REQUEST FOR DISCONNECT?
 CMP A #$4
 BNE R27
* SET FLAG FOR NEXT PASS
 LDA A #2
 STA A ZRFD
 JMP XIT
* DISCONNECT?
R27 CMP A #$6
 BNE R28
 JSR DISCON
 JMP LOOP
* NO INSTRUCTION?
R28 CMP A #$0
 BEQ R26
* ERROR PROCESSOR
```



```
R2ER LDA A #$28
 STA A RERR
 JMP R0
*
* RJ = 5,7
R5 EQU *
 LDA A #CLED5
 JSR LEDOFF
 JMP XIT
* RESPOND ONLY,ALREADY PROCESSED THIS SC
R6 LDA A #CLED5
 JSR LEDOFF
 LDA B #$40
 JSR SIM
 JMP XIT
*
* SETUP WAIT MESSAGE
*
WAIT EQU *
* BRING UP CARRIER
 LDA A #RTS
 STA A COMC
* HOLD TO DELAY
 LDX #0
WA1 DEX
 BNE WA1
* GET MESSAGE
 LDX #SWM
 STX SAVEX
WA2 LDX SAVEX
 LDA A 0,X
 INX
 STX SAVEX
 LDX #COMC
 JSR AOOL
 TST A
 BNE WA2
* TURN OFF CARRIER
 LDA A #CNIE
 STA A COMC
 CLR WAITF
 RTS
*
SWM FCB SOH,IMFC,SC,AC,$44,IC,STX,ETX,0
*
*
* SETUP SERVICE MESSAGE
SSM LDX #SMH
 JSR XFER
 RTS
SMH FCB SOH,SMFC,SC,AC,NOC,IC,STX,ETX,0
*
* TRANSFER DATA FROM STACK ARRAY
* TO X ARRAY
*
XFER STX SAVEX
 LDX #XBUF
 STX DEST
```

```
XFER2 LDX SAVEX
 LDA A 0,X
 INX
 STX SAVEX
*
 LDX DEST
 STA A 0,X
 INX
 STX DEST
 TST A
 BNE XFER2
* CLEAR REST OF BUFFER
 JSR CLXB
* SET OPER CODE
 LDX #XBUF
 STA B 4,X
 RTS
*
* SETUP INFO MESSAGE
*
SIM LDX #IMH
 JSR XFER
 RTS
*
IMH FCB SOH,IMFC,SC,AC,NOC,IC,STX,ETX,0
*
* SETUP DATA MESSAGE HEADER
*
SDMH LDX #OMH
 JSR XFER
 RTS
OMH FCB SOH,IMFC,SC,AC,NOC,IC,STX,0
*
* SETUP SELECT MESSAGE
*
SSLM LDX #SSH
 JSR XFER
 RTS
*
SSH FCB SOH,SMFC,SC,AC,NOC,IC,STX
 FCB ETX,0
*
* STUP NAK RESPONSE
*
SNR LDA A RERR
* EXTRACT NAK BITS
 AND A #$38
 STA A RERR
* RECOVER HEADER
 LDX #XBUF
 LDA A 4,X
* REMOVE NAK BITS
 AND A #$47
* INSERT RERR MESSAGE
 EOR A RERR
 STA A 4,X
 RTS
*
```



```
* SWAP SEQUENCE CODES (41-42)
*
SSC LDX #XBUF
 LDA A OLDSC
 EOR A #3
 STA A 2,X
 STA A OLDSC
 RTS
*
* SETUP TIMER #1 (INTERRUPT)
*
SET1 LDA A OLDCR2
 ORA A #1
 STA A OLDCR2
 STA A TCR2
* STORE TIME & START
 LDA A TSTS
 LDA A #T1IE TIMER 1
 STX T1D
 STA A TCR13
 RTS
*
* CLEAR TIMER #1 (INTERRUPT)
*
CLR1 LDA A OLDCR2
 ORA A #1
 STA A OLDCR2
 STA A TCR2
* DISABLE INTERRUPT
 LDA A #T1II
 STA A TCR13
 RTS
*
* SETUP TIMER #2 (INTERRUPT)
*
SET2 LDA A #T2IE
 STA A OLDCR2
 LDA B TSTS
* STORE TIME & START
 STX T2D
 STA A TCR2
 RTS
*
* CLEAR TIMER #2 (INTERRUPT)
*
CLR2 LDA A #T2II
 STA A OLDCR2
 STA A TCR2
 RTS
*
* SETUP TIMER #3 (INTERRUPT)
SET3 LDA A OLDCR2
 AND A #%11111110
 STA A OLDCR2
 STA A TCR2
* SET TIME & START
 LDA A TSTS
 LDA A #T3IE
```

```
 STX T3D
 STA A TCR13
 RTS
*
* CLEAR TIMER #3 (INTERRUPT)
*
CLR3 LDA A OLDCR2
 AND A #%11111110
 STA A OLDCR2
 STA A TCR2
* DISABLE INTERRUPT
 LDA A #T3II
 STA A TCR13
 RTS
*
* INTERRUPT SERVICE ROUTINE
*
ISER EQU *
* TEST TIME
 STS ISAVES
 LDA A TSTS
 BPL ISER1
* TEST CLOCK (TIMER #3 - 100 MSEC)
 BIT A #4
 BEQ ISER2
 LDX T3D
* SCAN CLOCK TABLE
 LDX #TMTB
 LDA B #5
 JSR SCAT
* UPDATE DISPLAYS
 LDX #DISBUF
 JSR SDIS
 RTI
* TEST TIMERS
ISER2 BIT A #2
 BEQ ISER3
* FIX RETURN VIA VECTOR2
 LDX T2D
* TEST IF WAIT TIMER INT
 TST UBJ
 BEQ ISER14
 INC WAITF
* TEST IF AGLS BUSY
 TST RAG
 BNE ISER17
 TST WAG
 BNE ISER17
 CLR WAITF
* SET WAIT RESPONSE
 JSR WAIT
ISER17 RTI
ISER14 JSR FRET2
 JSR CLR2
 RTI
ISER3 BIT A #1
 BNE ISER4
 RTI
```



```
* FIX RETURN VIA VECTOR #1
ISER4 JSR FRET1
 JSR CLR1
 RTI
*
* TEST COMM INTERRUPT
ISER1 TST COMS
 BPL ISER5
* COMM POLL
 JSR CPOLL
 RTI
*
* TEST AGLS COMM
ISER5 TST AGS
 BPL ISER6
* TEST RECEIVE
 LDA A AGS
 BIT A #1
 BEQ ISER7
* PROCESS RECEIVE
 JSR AREC
 RTI
* PROCESS TRANSMIT
ISER7 BIT A #2
 BNE ISER8
* TEST DCD INT.
 BIT A #4
 BEQ ISER15
* WAS I READING?
 TST RAG
 BEQ ISER16
 JSR RAGL
 LDA A AGR
 RTI
* WAS I WRITING?
ISER16 TST WAG
 BEQ ISER15
 JSR WAGL
 LDA A AGR
 RTI
* RESET INT
ISER15 LDA A AGR
 RTI
* PROCESS TRANSMT
ISER8 JSR AXMT
 RTI
*
* TEST STANDBY SWITCH
ISER6 TST PIA2CA
 BPL ISER13
* SET DISCONNECT FLAG
 LDA A #1
 STA A ZRFD
 LDA A PIA2DA
 RTI
* TEST CRT BUSY LOCKOUT
ISER13 TST CBSY
 BEQ ISER9
```

```
* RESET INT.
ISER10 LDA A TPR
 RTI
* TEST CRT INT.
ISER9 TST TPS
 BPL ISER10
* TEST RECEIVE
 LDA A TPS
 BIT A #1
 BEQ ISER11
* SERVICE RECEIVE
 JSR TPREC
 RTI
* TEST TRANSMIT
ISER11 BIT A #2
 BNE ISER12
* RESET INT.
 LDA A TPR
 RTI
* SERVICE TRANSMIT
ISER12 JSR TPXMT
 RTI
*
*
* SCAN TIMER TABLE
*
SCAT TST 0,X
 BEQ RT
 LDA A 2,X
 SUB A 0,X
 STA A 2,X
 BNE RT
 TST 1,X
 BEQ ST3
 DEC 1,X
 BRA RT
ST3 CLR 0,X
RT INX
 INX
 INX
 DEC B
 BNE SCAT
 RTS
*
* CLEAR TIMERS
*
CLTM LDA B #15
 LDX #TMTB
CLT1 CLR 0,X
 INX
 DEC B
 BNE CLT1
 RTS
*
* FIX RETURN VECTOR #2
*
FRET2 LDX ISAVES
 LDA A VECT2
```



```
 LDA B VECT2+1
 STA A 6,X
 STA B 7,X
 RTS
*
* FIX RETURN VECTOR #1
*
FRET1 LDX ISAVES
 LDA A VECT1
 LDA B VECT1+1
 STA A 6,X
 STA B 7,X
 RTS
*
* AGLS RECEIVE
*
AREC EQU *
* VERIFY MODE?
 TST VERF
 BEQ AREC2
* GET CHAR
 LDX #AGC
 JSR AOI
 BCC AREC3
* SKIP CRLF
 JSR TCRLF
 BCC AREC5
 RTS
* =X?
AREC5 CMP A #'X
 BEQ AREC3
* DATA OK
 CLR VERF
 BRA AREC4
* BAD XMISSION, DO AGAIN
AREC3 CLR RAG
 LDA A #NIE
 STA A AGC
 RTS
* RECEIVE DATA MODE
AREC2 LDX #AGC
 JSR AOI
 BCC AREC3
 CMP A #'X
 BEQ AREC3
* SKIP CR/LF
 JSR TCRLF
 BCC AREC4
 RTS
AREC4 LDX APTR
 SUB A #$30
 STA A 0,X
 INX
 STX APTR
 CPX #AGFE
 BEQ AREC1
* TEST STATUS
 LDA A AGS
```

```
 AND A #%01110000
 BNE AREC3
 RTS
AREC1 CLR CBSY
 CLR RAG
* INHIBIT INTERRUPT
 LDA A #NIE
 STA A AGC
 RTS
*
* INPUT FROM ACIA
*
AOI LDA A 0,X
 BIT A #1
 BEQ AOI1
 LDA A 1,X
 SEC
 RTS
AOI1 CLC
 RTS
* TEST CR/LF
*
TCRLF CMP A #$0D
 BNE TCR1
 SEC
 RTS
*
TCR1 CMP A #$0A
 BEQ TCR8
 CLC
 RTS
*
TCR8 SEC
 RTS
*
* AGLS TRANSMIT
*
AXMT CLR VERF
 LDX APTR
 LDA A 0,X
 ADD A #$30
 INX
 STX APTR
 LDX #AGC
 JSR AOO
 BCC AXMT1
 LDX APTR
 CPX #AGTE
 BEQ AXMT1
 RTS
AXMT1 CLR CBSY
 CLR RAG
 INC VERF
* INHIBIT INTERRUPT
 LDA A #NIE
 STA A AGC
 RTS
*
```



```
* TEST PORT TRANSMIT
TPXMT TST ASP
 BEQ TPXM1
 DEC ASP
TPXM4 LDX TPPTR
 CPX TPPTE
 BEQ TPXM2
 LDA A 0,X
 INX
 STX TPPTR
 TST SPC
 BEQ TPXM3
 ADD A #$30
TPXM3 LDX #TPC
 JSR AOO
 RTS
* FIX SPACE
TPXM1 LDA A SPC
 BEQ TPXM4
 STA A ASP
 LDA A #$20
 BRA TPXM3
* LAST CHAR-INH XMT /ENB RECV
TPXM2 LDA A #RIE
 STA A TPC
 LDA A TPR
 RTS
*
* TEST PORT RECEIVE
*
TPREC LDX #TPC
 JSR CRLF
 JSR AOI
 BCC TPRE1
 JSR AOOL
* - R? (RECV BUFFER)
TPRE1 CMP A #'R
 BNE TPRE2
 LDX #RBUF
 STX TPPTR
 LDX #REND
 STX TPPTE
 CLR A
 BRA TPRE3
* = X? (XMIT BUFFER)
TPRE2 CMP A #'X
 BNE TPRE4
 LDX #XBUF
 STX TPPTR
 LDX #XEND
 STX TPPTE
 CLR A
 BRA TPRE3
* = A? (AGLS BUFFER)
TPRE4 CMP A #'A
 BNE TPRE5
 LDX #AGLF
 STX TPPTR
```

F14

```
 LDX #AGTE
 STX TPPTE
 LDA A #5
 BRA TPRE3
* = F? (FLAG BUFFER)
TPRE5 CMP A #'F
 BNE TPRE6
 LDX #FLAG
 STX TPPTR
 LDX #FEND
 STX TPPTE
 LDA A #1
* SETUP SPACE COUNT
TPRE3 STA A SPC
 STA A ASP
* OUTPUT CR/LF
 LDX #TPC
 JSR CRLF
* INH REC/ENB XMIT
 LDA A #XIE
 STA A TPC
TPRE6 RTS
*
* OUTPUT TO ACIA
*
AOO PSH A
 LDA A 0,X
 BIT A #2
 PUL A
 BEQ AOO1
 STA A 1,X
 SEC
 RTS
AOO1 CLC
 RTS
*
* CR/LF ROUTINE
*
CRLF LDA A #$D
 JSR AOOL
 LDA A #$A
 JSR AOOL
 RTS
*
* LOOP ON OUTPUT
*
AOOL JSR AOO
 BCC AOOL
 RTS
*
* TRANSFER RECD MESSAGE TO CRT
*
XFRMSG LDX #RDATA
 STX TPPTR
* FIND END OF MESSAGE
XRMS1 LDA A 0,X
 CMP A #ETX
 BEQ XRMS2
```



```
 INX
 BRA XRMS1
* FIX END ADDRESS
XRMS2 STX TPPTE
 CLR SPC
 CLR ASP
* OUTPUT CR/LF
 LDX #TPC
 JSR CRLF
* INH REC/ENB XMIT
 LDA A #XIE
 STA A TPC
 RTS
*
* SET WAIT TIMER
*
SWAIT LDX #WAIT
 STX VECT2
 LDX #(WT/2-1)
 JSR SET2
 CLR UBJ
 INC UBJ
 RTS
*
* CLEAR WAIT TIMER
*
CWAIT JSR CLR2
 CLR UBJ
 RTS
*
* TRANSFER RECEIVED DATA
*
TRAN EQU *
* SET WAIT TIMER
 JSR SWAIT
* STRIP STATUS
 JSR STS
* TEST FIRE ORDER
 CMP A #1
 BNE TRAN1
 JSR PFO
 BRA TRAN2
* TEST CHECK FIRE
TRAN1 CMP A #8
 BNE TRAN3
* PROCESS CHECK FIRE
 JSR PCF
 BRA TRAN5
* ECHO-BACK RECIEVED DATA
* SETUP DATA MSG HEADER
TRAN2 LDA B #$40
 JSR SDMH
* SETUP DATA MESSAGE
 JSR SDM
* SETUP DATA MESSAGE TRAILER
 JSR SDMT
* CLEAR WAIT
 JSR CWAIT
```

```
 RTS
* TEST FIRE COMMAND
TRAN3 CMP A #2
 BNE TRAN4
 JSR PFC
 BRA TRAN5
* TEST READY REQUEST
TRAN4 CMP A #4
 BNE TRAN6
 JSR PRR
* SEND DATA REPORT
* SETUP DATA MESSAGE HEADER
TRAN5 LDA B #$40
 JSR SDMH
* SETUP DATA REPORT
 JSR SDR
* SETUP DATA REPORT TRAILER
 JSR SDRT
* CLEAR WAIT
 JSR CWAIT
 RTS
* CAN'T DECODE (NAK)
TRAN6 JSR CWAIT
 CLR TMPX
 INC RERR
 RTS
*
* STRIP STATUS
*
STS LDX #RDATA
 LDA B #4
 CLR RSWD
STS2 LDA A 0,X
 SUB A #$30
 EOR A RSWD
 STA A RSWD
 DEC B
 BNE STS1
 RTS
STS1 INX
 ASL RSWD
 BRA STS2
*
* RESET STATUS
*
RSTS LDA A ASWD FROM PROC ROUTINE ACK STATUS
 COM A
 STA A ASWD
* XFER TO XMIT BUFFER
 LDX #XDATA+3
 LDA B #4
RSTS2 CLR A
 ASR ASWD
 BCC RSTS1
 INC A
RSTS1 ADD A #$30
 STA A 0,X
 DEX
```



```
 DEC B
 BNE RSTS2
 RTS
*
* PROCESS FIRE ORDER
*
PFO EQU *
* STRIP ASCII ON DATA
 LDA B #12
 LDX #RDATA+4
 JSR SASC
* TRANSFER DEFLECTION TO REF. UNIT F
 JSR XRUP
* SET TIMEOUT
 LDX #30
 STX TIM1
 INC TF1
* TEST VALID RUP DATA (CB1)
PF01 TST PIA3CA
 BMI PF02 DATA VALID
* TEST TIME UP
 TST TF1
 BNE PF01
* TIMEOUT - TRANSFER DEFLECTION TO A
 JSR TBA
 BRA PF07
* READ & TRANSFER RUP DATA
PF02 CLR TF1
 JSR RRUP
* TRANSFER RECD DATA
* BUFFER
PF07 JSR XRDB
* WRITE DATA TO AGLS
 JSR COMAGL
 BCC PF08
* NAK FDC
 JSR CWAIT
 CLR TMPX
 INC RERR
 RTS
* OUTPUT DATA TO FUZE SETTER
PF08 JSR OFD
* TURN ON HORN & LITE
 LDA A #GUACK FO LITE ENB
 JSR LEDON
 JSR HORNIN
* WAIT FOR ACK
PF03 LDA A #FOSM FO ACK SW MASK
 JSR TSTSW
 BCS PF04
* TEST RUP UPDATE
 TST TF3
 BNE PF03
*
 LDX #10
 STX TIM3
 INC TF3
* REQUEST RUP UPDATE
```

```
 JSR REQRUP
 JSR RRUP
 BRA PF03
* TURN OFF HORN & LITE
PF04 LDA A #GUACK FO LITE DISAB
 JSR LEDOFF
 JSR HORNIF
* SET ACK STATUS WORD
 LDA A #FOAWD
 STA A ASWD
 RTS
*
* TRANSFER TO REF UNIT PROC.
*
XRUP EQU *
* MESSAGE COUNT
 LDA B #4
* SEND STX
 LDA A #1
 STA A OT3
 JSR XRUX
* SEND MESSAGE
 LDX #RDATA+4
XRUP4 LDA A O,X
 ADD A #$30
 STA A OT3
 JSR XRUX
 INX
 DEC B
 BNE XRUP4
* SEND AGLS MODE
 LDA A AGFE-2
 ADD A #$30
 STA A OT3
 JSR XRUX
* SEND ETX
 LDA A #3
 STA A OT3
 JSR XRUX
 RTS
*
* STROBE AND WAIT DATA ACCEPT
*
XRUX EQU *
* SEND DATA READY (CB2)
 LDA A PIA3DB
 JSR STROB3
* WAIT FOR DATA ACCEPTED (CB1)
XRUX1 TST PIA3CB
 BPL XRUX1
 RTS
*
* REQUEST RUP DATA
* SEND EQU(=5)
*
REQRUP EQU *
* SEND STX
 LDA A #1
```



```
 STA A OT3
 JSR XRUX
* SEND ENQ
 LDA A #5
 STA A OT3
 JSR XRUX
* SEND ETX
 LDA A #3
 STA A OT3
 JSR XRUX
 RTS
*
* TRANSFER CHAR STRING
* X=SOURCE,DEST=DESTINATION,B=CHAR CNT
*
TCS STX SAVEX
TCS1 LDX SAVEX
 LDA A O,X
 INX
 STX SAVEX
*
 LDX DEST
 STA A O,X
 INX
 STX DEST
 DEC B
 BNE TCS1
 RTS
*
* TRANSFER DEFLECTION TO AGLS
*
TBA LDX #AGLT+5
 STX DEST
 LDX #RDATA+4
 LDA B #4
 JSR TCS
 CLR O,X
 RTS
*
* SET AGLS STATUS WORD
* O=NORMAL
* 1=BD
* 2=BS
* 3=BD SET
* 4=BD CLR
*
ASTS EQU *
* READ RUP MODE
 CLR B
 LDA A IN2
 COM A
 AND A #%11100000
 BNE ASTS1
* TEST BASE DEFLECTION
 INC B
 LDA A IN2
 COM A
 BIT A #BDFM
```

```
PAGE 035   PVECOM   .SA:1
 BEQ ASTS2
* TEST BASE DEF. SET
 TST PIA2CB
 BPL ASTS1
 LDA A PIA2DB
 INC B
 INC B
 BRA ASTS1
* TEST BORESIGHT
ASTS2 INC B
 BIT A #BSTM
 BEQ ASTS3
* TEST BASE DEF CLEAR
 TST PIA2CB
 BPL ASTS1
 LDA A PIA2DB
 INC B
 INC B
* SET STATUS WORD
ASTS1 LDX #AGLT+10
 STA B 0,X
ASTS3 RTS
*
* READ & TRANSFER RUP DATA
*
RRUP EQU *
* TEST DATA REQ (CA1)
RRUP1 TST PIA3CA
 BPL RRUP1
* RECEIVE DATA
RRUP2 LDA A IN3
* TEST STX
 CMP A #1
 BNE RRUP2
 LDA B #4
 LDX #AGLT+5
* SEND DATA ACCEPTED (CA2)
RRUP6 LDA A IN3
 JSR STROA3
* TEST FOR DATA CALL (CA1)
RRUP4 TST PIA3CA
 BPL RRUP4
* GET DATA
RRUP5 CLR A
 LDA A IN3
 BEQ RRUP5
* - ETX
 CMP A #3
 BEQ RRUP3
* GET DATA
 SUB A #$30
 STA A 0,X
 INX
 DEC B
 BNE RRUP6
* SETUP TO GET DISPLAY DATA
 LDX #DISBUF
 CLR B
```

```
PAGE 036   PVECOM   .SA:1
 BRA RRUP6
* SEND DATA ACCEPTED (CA2)
RRUP3 JSR STROA3
 LDA A IN3
 RTS
*
**
* TRANSFER RECD DATA BUFFER
*
XRDB EQU *
* XFER ELEVATION TO AGLS
 LDX #AGLT
 STX DEST
 LDX #RDATA+8
 LDA B #4
 JSR TCS
 CLR 0,X
* SET AGLS MODE
 JSR ASTS
* SETUP DISPLAY BUFFER
* DEFLECTION
 LDX #DISBUF
 STX DEST
 LDX #AGLT+5
 LDA B #4
 JSR TCS
* ELEVATION,FUZE,CHARGE
 LDX #DISBUF+4
 STX DEST
 LDX #RDATA+8
 LDA B #2
 JSR TCS
*
 LDX #DISBUF+8
 STX DEST
 LDX #RDATA+10
 LDA B #6
 JSR TCS
 RTS
*
* WRITE DATA TO AGLS
*
WAGL JSR ASTS
 CLR CBSY
 INC CBSY
* SEND "R"
 LDA A #$20
 LDX #AGC
 JSR AOOL
 LDA A #'R
 JSR AOOL
* INITIALIZE POINTERS
 LDX #AGLT
 STX APTR
* SET FLAGS
 CLR A
 STA A RAG
 INC A
```

F/8

```
 STA A WAG
* ENB XMIT/INH RECV
 LDA A #XIE
 STA A AGC
 LDA A AGR
 RTS
*
*
*
LEDON EQU *
 LDX #LEDUP
 JSR LEDFIX
 RTS
*
*
LEDOFF EQU *
 LDX #LEDDN
 JSR LEDFIX
 RTS
*
*
LEDUP EQU *
 TST B
 BNE LUP1
*
 ORA A LEDWD
 STA A LEDWD
 RTS
*
LUP1 ORA A LEDWD+1
 STA A LEDWD+1
 RTS
*
*
LEDDN EQU *
 TST B
 BNE LDN1
*
 COM A
 AND A LEDWD
 STA A LEDWD
 RTS
*
LDN1 COM A
 AND A LEDWD+1
 STA A LEDWD+1
 RTS
*
* FIX LED WORD
*
LEDFIX EQU *
 CLR B
 TST A
 BMI LON1
* LOW ORDER
 JSR 0,X
 RTS
* HI ORDER
```



```
LON1 AND A #$7F
 INC B
 JSR 0,X
 RTS
*
* SERVICE LEDS (DATA-ADDR)
*
SLED EQU *
* ENABLE CHIP SELECT CB2 HI
 LDA B #%00111110
 STA B PIA2CB
*
 STX SAVES
 LDA B #1
* OUTPUT TO LEDS
 LDA A SAVES
 AND A #$F0
 ABA
 STA A OT2
 JSR STROB2
*
 LDA B #0
 ASL SAVES
 ASL SAVES
 ASL SAVES
 ASL SAVES
 LDA A SAVES
 ABA
 STA A OT2
 JSR STROB2
*
 LDA B #3
 LDA A SAVES+1
 AND A #$F0
 ABA
 STA A OT2
 JSR STROB2
*
 LDA B #2
 ASL SAVES+1
 ASL SAVES+1
 ASL SAVES+1
 ASL SAVES+1
 LDA A SAVES+1
 ABA
 STA A OT2
 JSR STROB2
 RTS
*
* TEST SWITCH STATUS
* C=SET IF SW ON; C=0 IF OFF
*
TSTSW AND A IN2
 BEQ TSTS1
 CLC
 RTS
TSTS1 SEC
 RTS
```

```
*
* STROBE OUTPUT PULSE
* PIA2A
*
STROB2 LDA A #%00110101
 STA A PIA2CA
 NOP
 LDA A #%00111101
 STA A PIA2CA
 RTS
*
* STROBE OUTPUT PULSE
* PIA3B
*
STROB3 LDA A #%00110100
 STA A PIA3CB
 NOP
 LDA A #%00111100
 STA A PIA3CB
 RTS
*
* STROBE OUTPUT PULSE
* PIA3A
*
STROA3 LDA A #%00110100
 STA A PIA3CA
 NOP
 LDA A #%00111100
 STA A PIA3CA
 RTS
*
* PROCESS FIRE COMMAND
*
PFC EQU *
* TURN ON FIRE INDICATOR
 LDA A #FIRE FIRE LITE ENB
 JSR LEDON
 JSR HORN2N
* WAIT FOR ACK
PFC1 LDA A #FASM FC ACK SW MASK
 JSR TSTSW
 BCC PFC1
* TURN OFF FIRE INDICATOR
 LDA A #FIRE FIRE LITE DIS
 JSR LEDOFF
 JSR HORN2F
* SET ACK STATUS WORD
 LDA A #FCAWD
 STA A ASWD
 RTS
*
* READ AGLS DATA
*
RAGL CLR CBSY
 INC CBSY
* SEND "T"
 LDA A #'T
 LDX #AGC
```



```
 JSR AOOL
* INITIALIZE POINTERS
 LDX #AGLF
 STX APTR
* SET FLAGS
 CLR A
 STA A WAG
 INC A
 STA A RAG
* ENB RECV/INH XMIT
 LDA A #RIE
 STA A AGC
 LDA A AGR
 RTS
*
* TRANSFER RECD BUFFER TO XMIT
*
XRX LDX #XDATA+4
 STX DEST
 LDX #RDATA+4
 LDA B #12
 JSR TCS
* ADD ASCII TO DATA
 LDA B #12
 LDX #XDATA+4
 JSR AASC
 RTS
* TRANSFER AGLS 'FROM' TO XMIT
*
AGTX LDX #XDATA+15
 STX DEST
 LDX #AGLF
 LDA B #30
 JSR TCS
 LDX #XDATA+45
 STX DEST
 LDX #AGLF+41
 LDA B #3
 JSR TCS
* ADD ASCII TO DATA
 LDA B #33
 LDX #XDATA+15
 JSR AASC
* FIX AGLS ERR SIGNS
 LDX #XDATA+25
AGTX2 LDA B #$20
 LDA A 0,X
 CMP A #$31
 BNE AGTX3
 LDA B #'-
AGTX3 STA B 0,X
*
 CPX #XDATA+25
 BEQ AGTX1
 RTS
*
AGTX1 LDX #XDATA+40
 BRA AGTX2
```

F20

```
 RTS
*
* STRIP ASCII FROM CHARS
*
SASC LDA A 0,X
 SUB A #$30
 STA A 0,X
 INX
 DEC B
 BNE SASC
 RTS
*
*
* ADD ASCII TO CHARS
*
AASC LDA A 0,X
 ADD A #$30
 STA A 0,X
 INX
 DEC B
 BNE AASC
 RTS
*
* PROCESS READY REQUEST
*
PRR EQU *
* TURN ON READY INDICATOR
 LDA A #REDY READY LITE ENB
 JSR LEDON
* TEST VELOCIMETER READY
 LDA A VLR
 LDA A VLS
 BIT A #4
 BNE PRR1
* READ DATA
 JSR RVEL
* TEST DATA
 JSR TVEL
 BCC PRR
* WAIT FOR ACK
PRR1 LDA A #RRSM RR ACK SW MASK
 JSR TSTSW
 BCS PRR2
* TEST RUP UPDATE
 TST TF3
 BNE PRR1
*
 LDX #10
 STX TIM3
 INC TF3
* REQUEST RUP UPDATE
 JSR REQRUP
 JSR RRUP
* READ AGLS
* PRESENT?
 LDA A AGR
 LDA A AGS
 BIT A #4
```



```
 BNE PRR1
*
 JSR RAGL
PRR4 TST RAG
 BNE PRR4
* TEST AUTO UPDATE
 LDA A AGFE-1
 CMP A #2
 BGE PRR3
* TEST DEFERRED WAIT
 TST WAITF
 BEQ PRR1
 JSR WAIT
 BRA PRR1
* AUTO UPDATE,XFER NEW DATA
PRR3 JSR COMAGL
* TEST DEFERRED WAIT
 TST WAITF
 BEQ PRR1
 JSR WAIT
 BRA PRR1
* TURN OFF READY INDICATOR
PRR2 LDA A #REDY READY LITE DIS
 JSR LEDOFF
* SET ACK STATUS WORD
 LDA A #RRAWD
 STA A ASWD
 RTS
* SETUP DATA MESSAGE
*
SDM EQU *
* TRANSFER RECD DATA TO XMIT
 JSR XRX
* TRANSFER STATUS
 JSR RSTS
 RTS
*
* SETUP DATA REPORT
*
SDR EQU *
* TEST AGLS PRESENT
 LDA A AGR
 LDA A AGS
 BIT A #4
 BNE SDM2
 LDA A #'E
 STA A AGLF
* WAIT TILL NOT BUSY
SDM3 TST RAG
 BNE SDM3
* REQUEST AGLS DATA READ
 JSR RAGL
* WAIT COMPLETE
SDM1 TST RAG
 BNE SDM1
* VERIFY GOOD READ
 LDA A AGLF
 CMP A #'E
```

```
 BNE SDM2
* RECD NAK FROM AGLS-RECOVER
 JSR COMAGL
* TRANSFER STATUS
SDM2 JSR RSTS
* TRANSFER AGLS TO XMIT
 JSR AGTX
* TEST VELOCIMETER
* DCD INDICATES
*
* FILL VEL BUFFER
 JSR FVEL
* TEST FIRE CMND RESP
* SKIP VEL READ IF=
 LDA A XDATA+2
 CMP A #$30
 BEQ SDM6
*
 LDA A VLR
 LDA A VLS
 BIT A #4
 BNE SDM6
* READ VELOCIMETER
 JSR RVEL
* TRANSFER VELOCITY
SDM6 JSR XVEL
* READ TEMPERATURE
 JSR RTEMP
 RTS
*
* SETUP DATA MSG TRAILER
*
SDMT LDX #XDATA+16
 LDA A #ETX
 STA A 0,X
 CLR 1,X
 RTS
*
* SETUP DATA REPORT TRAILER
*
SDRT LDX #XDATA+48
 LDA A #ETX
 STA A 0,X
 CLR 1,X
 RTS
*
* SETUP PIAS
*
* PIA1-PROP TEMP
PIAS LDX #PIA1DA
 LDA A #$36
 JSR SETUP
* PIA2-DISPLAYS & CONTROLS
 LDX #PIA2DA
 LDA A #$3C
 JSR SETUP
* SETUP CA1 TO INTERRUPT
 LDA A #%00111101
```



```
 STA A PIA2CA
* ZERO DISPLAYS
 JSR CDIS
*
* PIA3 - REF UNIT PROC
* A=INPUT
* B=OUTPUT
* DE-GLITCH CA2,CB2
 LDX #PIA3DA
 LDA A #%00111000
 STA A 2,X
 STA A 3,X
* A=INPUT
 CLR 0,X
* B=OUTPUT
 LDA A #$FF
 STA A 1,X
* CONTROL
 LDA A #%00111100
 STA A 2,X
 STA A 3,X
* CLEAR CA1,CB1
 LDA A 0,X
 LDA A 1,X
*
*
* TIMER
 LDA A #T311
 STA A TCR13
 LDA A #T211
 STA A TCR2
 STA A OLDCR2
 LDA A #T111
 STA A TCR13
* SET TIMER #3 PERIOD (100 MSEC)
 LDX #12500
 JSR SET3
*
*
* COMM ACIA
 LDA A #$43
 STA A COMC
 LDA A #CNIE
 STA A COMC
*
* AGLS ACIA
 LDA A #3
 STA A AGC
 LDA A #NIE
 STA A AGC
*
* TEST PORT ACIA
 LDA A #3
 STA A TPC
 LDA A #RIE
 STA A TPC
*
* FUZE SETTER ACIA
```

F27

```
 LDA A #$43
 STA A FSC
 LDA A #NIE
 STA A FSC
*
* VELOCIMETER ACIA
 LDA A #3
 STA A VLC
 LDA A #%00001101
 STA A VLC
* DISABLE INTERRUPTS
 SEI
 NOP
 RTS
*
* SETUP PIAS
*
SETUP CLR 2,X CA DD SELECT
 CLR 3,X CB DD SELECT
 LDA B #$FF B SIDE-OUTPUT
 STA B 1,X B SIDE OUTPUT
 CLR 0,X A SIDE=INPUT
 STA A 2,X CA OUPUT & CONTROL SELECT
 STA A 3,X CB OUTPUT & CONTROL SELECT
 CLR 1,X ZERO OUTPUT
 LDA A 0,X RESET
 LDA A 1,X RESET
 RTS
*
* INITIALIZE CONNECT
*
ICON EQU *
* SELECT MESSAGE FLAG
 JSR CLR1
 CLR ZIDM
 DEC ZIDM
* CLEAR CONNECT FLAG
 CLR CONN
* TIMEOUT RETURN VECTOR
 LDX #ICONX
 STX VECT2
* TRANSMIT BUFFER POINTER
 LDX #XBUF
 STX BX1
* TEST 10 TRIES
 DEC TRY
 BEQ ICON1
 RTS
ICON1 LDX #*
 JMP IDLE
*
* CONNECT PROCESS
*
CONP EQU *
* SET PASS FLAG
 CLR XPASS
 DEC XPASS
* START CLOCK
```



```
 LDX #(CONTT/2-1)
 JSR SET2
* SET FLAG CJ
 CLR CJ
 INC CJ
* ENB XMIT/INH RECV
 JSR COMIR
 RTS
*
* CONNECT TEST
*
CONT EQU *
* TEST ACK/NAK
* STOP TIMER
 JSR CLR2
 LDA A RJ
 CMP A #2
 BGE CONT1
* NAK IF 1 OR 0
 JMP ICONX
* ACK IF 2 OR 3
CONT1 INC CONN
* TURN ON COMM/OFF STANDBY
 LDA A #COMM
 JSR LEDON
 LDA A #STBY
 JSR LEDOFF
* FLAG READY FOR RESPONSE
 CLR ZRRF
 DEC ZRRF
 JMP XIT
*
*
* READ VELOCIMETER
*
RVEL EQU *
* WAIT ON CHAR
RVEL1 LDX #VLS
 JSR AOI
 BCC RVEL1
* WAIT ON LEADING LF
 CMP A #$OA
 BNE RVEL1
* GET DATA
 LDX #VELBUF
 STX SAVEX
RVEL2 LDX #VLS
 JSR AOI
 BCC RVEL2
* STORE CHAR
 LDX SAVEX
 STA A 0,X
 INX
 STX SAVEX
* GET OUT ON CR
 CMP A #$OD
 BNE RVEL2
 RTS
```

```
*
* TEST VELOCIMETER FOR 0
* C=CLR, NOT RESET
* C=SET, RESET
*
TVEL LDX #VELBUF+2
 LDA B #5
TVEL1 LDA A 0,X
 CMP A #$30
 BNE TVEL2
 INX
 DEC B
 BNE TVEL1
 SEC
 RTS
*
TVEL2 CLC
 RTS
*
* TRANSFER VELOCITY
*
XVEL LDX #XDATA+4
 STX DEST
 LDX #VELBUF+2
 LDA B #5
 JSR TCS
 RTS
*
* FILL VELOCITY BUFFER (=$30)
*
FVEL LDX #VELBUF
 LDA B #10
 LDA A #$30
FVEL1 STA A 0,X
 INX
 DEC B
 BNE FVEL1
 RTS
*
* READ TEMPERATURE
*
RTEMP EQU *
* SETUP BUFFER PTRS
 LDX #XDATA+14
 CLR B
* PUSH ADDR AND READ
RTM1 STA B OT1
 LDA A IN1
 AND A #$0F
 ADD A #$30
 STA A 0,X
 DEX
 INC B
 CMP B #5
 BNE RTM1
* GET POLARITY
 STA B OT1
 LDA B #'- MINUS
```



```
 LDA A IN1
 AND A #$0F
 CMP A #7
 BNE RTM2
 LDA B #'+ PLUS
RTM2 STA B 0,X
 RTS
*
* SERVICE DISPLAY
* X=ADDRESS OF BUFFER
*
SDIS EQU *
* LAMP TEST?
 LDA A IN2
 AND A #$F0
 BEQ SDIS1
* CLEAR?
 CPX #0
 BEQ SDIS2
* NORMAL DISPLAY
 JSR DIS
 LDX LEDWD
 JSR SLED
 RTS
*
* LAMP TEST
SDIS1 LDA A #$80
 LDX #0
 JSR DIS
* ALL LEDS ON
 LDX #$FFFF
 JSR SLED
 RTS
*
* CLEAR
SDIS2 LDX #0
 CLR A
 JSR DIS
 RTS
*
* DISPLAY ROUTINE
*
* X=ADDRESS
*
DIS EQU *
* DISABLE CHIP SELECT
 LDA B #%00110110 CB2 LO
 STA B PIA2CB
*
 LDA B #16
 CLR DISADD
*
DIS3 CPX #0
 BEQ DIS1
 LDA A 0,X
 ASL A
 ASL A
 ASL A
```

```
 ASL A
DIS1 AND A #$F0
 EOR A DISADD
 STA A OT2
 INC DISADD
 PSH A
 JSR STROB2
 PUL A
 CPX #0
 BEQ DIS2
 INX
DIS2 DEC B
 BNE DIS3
 RTS
*
* CLEAR DISPLAYS
*
CDIS LDX #0
 JSR SDIS
 LDX #0
 JSR SLED
 RTS
*
*
* OUTPUT FUZE DATA
* C DDDD FFF EEE #1
*
OFD LDX #RDATA+15
 LDA B #1
 JSR XFUZE
*
 LDX #SP
 LDA B #1
 JSR XFUZE
*
 LDX #RDATA+4
 LDA B #4
 JSR XFUZE
*
 LDX #SP
 LDA B #1
 JSR XFUZE
*
 LDX #RDATA+12
 LDA B #3
 JSR XFUZE
*
 LDX #SP
 LDA B #1
 JSR XFUZE
*
 LDX #RDATA+9
 LDA B #3
 JSR XFUZE
*
 LDX #GN
 LDA B #3
 JSR XFUZE
```



```
*
 RTS
*
SP FCB $F0
GN FCB $F0,$F3,01
*
* TRANSMIT FUZE DATA
*
XFUZE STX SAVEX
XFZ1 LDX SAVEX
 LDA A 0,X
 INX
 STX SAVEX
 ADD A #$30
 LDX #FSC
* LOOP OUTPUT
 JSR AOOL
 DEC B
 BNE XFZ1
 RTS
*
* SET IDLE LITE
*
IDLL LDA A #COMM
 JSR LEDOFF
 LDA A #STBY
 JSR LEDON
 RTS
*
* DISCONNECT COMM
*
DISCON CLR CONN
 LDA A #1
 STA A IDLEF
 RTS
* PROCESS CHECK FIRE
*
PCF EQU *
* DISPLAY 9'S
PCF6 LDA A #9
 JSR FIXDIS
* SOUND HORN
 JSR HORNIN
* LIGHT READY ACK
 LDA A #REDY
 JSR LEDON
* TEST READY ACK
 LDA A #RRSW
 JSR TSTSW
 BCS PCF1
* WAIT A LITTLE
 LDX #0
PCF2 DEX
 BNE PCF2
* DISPLAY BLANK
 LDA A #$F
 JSR FIXDIS
* EXTINGUISH HORN
```

```
 STA A PIAICB
 LDA A #HORNOF
 STA A PIAICA
 RTS
*
* HORN 1 OFF (CONT)
*
HORN1F EQU *
* GET PIA CONTROL
 LDA A #HORNOF
 STA A PIAICB
 STA A PIAICA
 RTS
*
* HORN 2 ON (PULSE)
*
HORN2N EQU *
* GET PIA CONTROL
 LDA A #HORNON
 STA A PIAICB
 STA A PIAICA
 RTS
*
* HORN 2 OFF (PULSE)
*
HORN2F EQU *
* GET PIA CONTROL
 LDA A #HORNOF
 STA A PIAICA
 STA A PIAICB
 RTS
*
* CLEAR COMM STATUS LEDS
*
CLRL01 LDA A #CLEDO
 JSR LEDOFF
 LDA A #CLED1
 JSR LEDOFF
 RTS
*
CLRL02 LDA A #CLEDO
 JSR LEDOFF
 LDA A #CLED2
 JSR LEDOFF
 RTS
*
CLRL12 LDA A #CLED1
 JSR LEDOFF
 LDA A #CLED2
 JSR LEDOFF
 RTS
*
* COMMUNICATE TO AGLS
* C=SET,WONT TAKE IT
* C=CLR,XMISSION OK
*
COMAGL EQU *
* TEST AGLS PRESENT (DCD LOW)
```



```
 JSR HORN1F
* TEST READY ACK
 LDA A #RRSM
 JSR TSTSW
 BCS PCF1
* WAIT A LITTLE
 LDX #0
PCF4 DEX
 BNE PCF4
 BRA PCF6
*
* READY ACK RECIEVED
PCF1 LDA A #REDY
 JSR LEDOFF
* SHUT OFF HORN
 JSR HORN1F
 LDA A #$F
 JSR FIXDIS
* SETUP ACK WORD
 LDA A #CFAWD
 STA A ASWD
 RTS
*
* FIX DISPLAY BUFFER
*
FIXDIS LDA B #14
 LDX #DISBUF
FXD1 STA A 0,X
 INX
 DEC B
 BNE FXD1
 RTS
*
* CLEAR REC BUFFER
*
* X=START ADDR
*
CLRB CLR 0,X
 INX
 CPX #REND
 BNE CLRB
 RTS
*
* CLEAR XMIT BUFFER
*
* X=START ADDR
*
CLXB CLR 0,X
 INX
 CPX #XEND
 BNE CLXB
 RTS
*
* HORN 1 ON (CONT)
*
HORN1N EQU *
* GET PIA CONTROL
 LDA A #HORNON
```

```
 LDA A AGR
 LDA A AGS
 BIT A #4
 BEQ COMAG5
 CLC
 RTS
*
COMAG5 LDA A #CLED4
 JSR LEDON
 LDA B #10
 STA B AGTRY
COMAG1 JSR WAGL
* WAIT TILL DONE
COMAG2 TST WAG
 BNE COMAG2
* ECHO BACK TO VERIFY
 JSR RAGL
* WAIT TILL DONE
COMAG3 TST RAG
 BNE COMAG3
* OK ?
 TST VERF
 BEQ COMAG4
* COUNT TRYS
 DEC AGTRY
 BNE COMAG1
 LDA A #CLED4
 JSR LEDOFF
 SEC
 RTS
* GOOD RETURN
COMAG4 LDA A #CLED4
 JSR LEDOFF
 CLC
 RTS
* TEST AGLS PRESENT
* DCD LOW IND.
*
RUTHER CLR B
 LDX #60000
RUT2 LDA A AGR
 LDA A AGS
 BIT A #4
 BEQ RUT1
* FIRST PASS ?
 CPX #60000
 BEQ RUT3
 TST B
 BEQ RUTHER
RUT3 CLR B
 INC B
 DEX
 BNE RUT2
 RTS
*
RUT1 TST B
 BNE RUTHER
 DEX
```



```
 BNE RUT2
 RTS
*
* TEST BASE DEFL
*
TBD EQU *
 LDX #LEDON
 LDA A #BDSET
 LDA B AGFE-1
 AND B #1
 BNE TBD1
* TURN OFF
 LDX #LEDOFF
* TURN ON
TBD1 JSR 0,X
 RTS
* SYSTEM VECTORS
 ORG $FFF8
 FDB ISER
 FDB ISER
 FDB STRT
 FDB PWRUP
 END
```

F27

APPENDIX G

RUP CONTROL PROGRAM
SOURCE LISTING

```
 NAM RUP
*
* REFERENCE UNIT PROCESSOR
*
* REVISED 3/2/79 1630
*
*
* PIA EQUATES
* PIA1-REF UNIT RECEIVER
*
*
PIA1DA EQU $C200
PIA1DB EQU PIA1DA+1
PIA1CA EQU PIA1DA+2
PIA1CB EQU PIA1DA+3
*
* PIA2-COMM. PROCESSOR
* A=OUTPUT
* B=INPUT
PIA2DA EQU $C204
PIA2DB EQU PIA2DA+1
PIA2CA EQU PIA2DA+2
PIA2CB EQU PIA2DA+3
*
* PIA3-SW INPUTS/TEST OUTPUTS
*
* A=INPUT
* B=OUTPUT
PIA3DA EQU $C208
PIA3DB EQU PIA3DA+1
PIA3CA EQU PIA3DA+2
PIA3CB EQU PIA3DA+3
*
* TIMER EQUATES
*
TCR13 EQU $9800
TSTS EQU $9801
TCR2 EQU $9801
T1D EQU $9802
T2D EQU $9804
T3D EQU $9806
*
* TIMER CONSTANTS
*
T1II EQU %00000000 C=EXT,0=OFF
T1IE EQU %01000000
*
T2II EQU %10000001 C=EXT,0=ON
T2IE EQU %11000001
*
T3II EQU %10000010 C=INT,0=ON,NO PRE
T3IE EQU %11000010
*
* ACIA EQUATES
*
TPC EQU $9808
TPS EQU TPC
TPX EQU TPC+1
```



```
TPR EQU TPC+1
*
* ACIA CONSTANTS
*
XIE EQU %00101001
RIE EQU %10001001
NIE EQU %00001001
*
* PIA3 CONSTANTS
*
GLITE EQU 1
OLITE EQU 2
LLITE EQU 4
XLITE EQU 8
SOUT EQU $10
LOUT EQU $20
 PAGE
* PROGRAM RAM
 ORG $0
*
FLAG EQU *
LFLAG RMB 1 LASER PULSE
LBLOCK RMB 1 LASER BLOCK
LGONE RMB 1 LASER LOST
CFLAG RMB 1 COMPUTE
SFLAG RMB 1 SOUTH PULSE
AUTOF RMB 1 AUTO RESPONSE
INF RMB 1 INPUT FLAG
FEND EQU *
*
* CONSTANTS
*
XPER RMB 2 X PULSE-L PULSE PERIOD
XTIM RMB 2 X PULSE-X PULSE PERIOD
MIL RMB 2 COARSE MILS COUNT
OUT RMB 2 BINARY OUTPUT VALUE
OLDMIL RMB 2 2 PASS SAVE VALUE
OUTX RMB 2 BUFFER PTR (BINBCD)
DEST RMB 2 TRANSFER CHAR PTR
SCNT RMB 2 SOUTH PULSE COUNT (TEST)
LVAL RMB 2 LASER PULSE VALUE (TEST)
LCNT RMB 2 LASER PULSE COUNT (TEST)
SAVES RMB 2 STACK SAVE
A1 RMB 2 BCD ADD
A2 RMB 2 BCD ADD
RESULT RMB 4 TEMP CHAR BUFFER
SAVEX RMB 2 X REG SAVE
SIG RMB 1 SIGN OF ADD
OLDCR2 RMB 1 TIMER 2 CONTROL
TPPTR RMB 2 TP CHAR PTR
TPPTE RMB 2 TP CHAR END
TBUF RMB 5 CHAR BUFFER
SPC RMB 1 SPACE CNT
ASP RMB 1 ACTIVE SPACE
TMP RMB 1 DGN TEMPORARY STORAGE
MSBY RMB 1 DGN MS BYTE
LSBY RMB 1 DGN LS BYTE
CEND EQU *
```

```
BEND RMB 2 END BUFFER (CLBF)
*
* TIMER TABLE
* COUNTER #1
TMTB EQU *
*
TMT1 EQU *
CS1 RMB 1 STATUS
C1 RMB 1 COUNTER (DEC)
CNT1 RMB 1 CYCLE
CP1 RMB 1 LOOK-AHEAD
CH1 RMB 1 HOLD COUNT
*
TMT2 EQU *
CS2 RMB 1 STATUS
C2 RMB 1 COUNTER (DEC)
CNT2 RMB 1 CYCLE
CP2 RMB 1 LOOK-AHEAD
CH2 RMB 1 HOLD COUNT
*
TMT3 EQU *
CS3 RMB 1 STATUS
C3 RMB 1 COUNTER (DEC)
CNT3 RMB 1 CYCLE
CP3 RMB 1 LOOK-AHEAD
CH3 RMB 1 HOLD COUNT
*
TEND EQU *
* REFERENCE ANGLE BUFFER
*
REFBUF RMB 4
*
* COMM PROC. BUFFER
*
CMPOUT EQU *
AZ RMB 4
DIS RMB 4
*
CMPIN EQU *
DEFL RMB 4
AMODE RMB 1
BUFEND EQU *
 PAGE
*
*
* REFERENCE UNIT PROCESSOR PROGRAM
*
*
 ORG $F800
STRT EQU *
* H-10 DEBUG
 LDS #$7F
* SETUP PIAS AND TIMERS
 SEI
 JSR PIAS
 JSR TIMI
* CLEAR FLAGS/CONSTANTS
 LDX #CEND
```



```
 STX BEND
 LDX #FLAG
 JSR CLBF
* CLEAR TIMER TABLE AND INITIALIZE
 JSR CLTI
* CLEAR BUFFERS
 LDX #BUFEND
 STX BEND
 LDX #REFBUF
 JSR CLBF
* INITIAL TEST S COUNT
 LDX #(160*2)
 STX SCNT
* INITIAL TEST L COUNT
 LDX #120
 STX LVAL
* ENABLE TEST MODE ?
 LDA A PIA3DA
 BIT A #4
 BNE L12
* ENABLE TIMER #2 INT (3125 USEC)
 LDX #(312/2-1)
 JSR SET2
* ENABLE INTERRUPTS
L12 CLI
*
*
* PROGRAM ACTIVE LOOP
*
*
LOOP EQU *
*
 LDA B #GLITE
 LDX #PON
* LOCK FLAG?
 LDA A PIA3DA
 BIT A #1
 BNE L10
* ACTIVE COUNTER ?
 LDA A CS1
 ORA A CS2
 ORA A CS3
 BEQ L10
 TST LOONE
 BNE L10
 LDX #POFF
* CONTROL "GACS" LAMP
L10 TBA
 JSR 0,X
* IF GLITE OFF, CFLAG=0
 LDA A PIA3DB
 BIT A #GLITE
 BEQ L11
 LDA A #$0C
 JSR PON
 CLR CFLAG
* COMPUTE FLAG ?
L11 TST CFLAG
```

```
 BNE L14
 JMP L3
* FIND MAX LIKELY VALUE
L14 LDA B CH3
 LDA A CNT3
 CMP A CNT2
 BGT L1
*
 LDA B CH2
 LDA A CNT2
*
L1 CMP A CNT1
 BGT L2
 LDA B CH1
 LDA A CNT1
*
L2 TST A
 BEQ ERR
 CLR MIL
 LDA A #159
 SBA
 STA A MIL+1
* TEST COARSE RANGE
 CMP A #159
 BHI ERR
*
* COMPUTE: OUT=40*MIL+40(XPER/XTIM)
*
* COMPUTE: MIL=MIL*40
 LDX #MIL
 JSR MUX40
 STA A MIL
 STA B MIL+1
* FETCH XTIM AND TEST LIMITS
 LDA A XTIM
* TEST RANGE (<110%)
 CMP A #2
 BLT E2
 BGT ERR
*
 LDA A XTIM+1
*
 CMP A #$AF
 BGT ERR
* COMPUTE: XPER=XPER*40
E2 LDX #XPER
 JSR MUX40
 STA A XPER
 STA B XPER+1
* COMPUTE: A,B=XPER(A,B)/XTIM
 LDX #XTIM
 JSR DIV16
* COMPUTE: OUT=MIL+A,B
 ADD B MIL+1
 ADC A MIL
* ADJUST FOR GACS
 SUB B #82
 SBC A #0
```



```
* TEST ROLLOVER
 BPL L13
* FIX ROLLOVER (ADD 6400)
 ADD A #$19
L13 STA A OUT
 STA B OUT+1
* TEST RANGE (0-6399)
 CMP A #$18
 BGT ERR
* MAKE SURE SAME VALUE 2X
 LDX OUT
 CPX OLDMIL
 BEQ L19
 STX OLDMIL
 BRA ERR
* CONVERT OUT TO BCD
L19 LDX #REFBUF
 JSR BINBCD
ERR CLR CFLAG
* LAMP TEST ?
L3 LDA B PIA3DB
L16 LDA A PIA3DA
 COM A
 BIT A #$80
 BEQ L15
* TEST LAMPS
 LDA A #$0F
 JSR POFF
 BRA L16
* RESTORE
L15 STA B PIA3DB
* COMM PROCESSOR DATA CALL ?
 LDA A PIA2CB
 BMI L4
 JMP LOOP
*
* DATA CALL-ACQUIRE
*
L4 JSR RCMP
 BCC L17
 JMP LOOP
* COMPUTE AZIMUTH (DEFL+REF)
L17 JSR CAZ
* TEST MODE SWITCH
 LDA A PIA3DA
 COM A
 AND A #$F0
 BNE L5
* NORMAL
* COPY AZ TO DISPLAY
 LDX #DIS
 STX DEST
 LDX #AZ
 LDA B #4
 JSR TCS
 LDX #POFF
 BRA L6
*
```

```
.5 CMP A #$10
 BNE L7
* BORESIGHT, 3200 TO DEFL AND DIS
 LDX #AZ
 STX DEST
 LDX #CON32
 LDA B #8
 JSR TCS
 LDX #PON
 BRA L6
*
.7 CMP A #$20
 BNE L8
* FDC, COPY FIRE ORDER TO DISP
 LDX #DIS
 STX DEST
 LDX #DEFL
 LDA B #4
 JSR TCS
 LDX #POFF
 BRA L6
.8 CMP A #$30
 BNE L18
* BASE DEFL, COPY COMPIN TO COMPOUT AND DISP
 LDX #AZ
 STX DEST
 LDX #DEFL
 LDA B #4
 JSR TCS

 LDX #DEFL
 LDA B #4
 JSR TCS
 LDX #POFF
 BRA L6

18 CMP A #$40
 BNE L9
* REF, COPY REF ANGLE TO DISP
 LDX #DIS
 STX DEST
 LDX #REFBUF
 LDA B #4
 JSR TCS
 LDX #POFF
 BRA L6
* CONTROL "OPERATE" LITE
5 LDA A #OLITE
 JSR 0,X
* OUTPUT TO COMM PROC.
) JSR SCMP
 JMP LOOP


N32 FCB 3,2,0,0,3,2,0,0
N99 FCB 9,9,9,9
PAGE
```



```
* INTERRUPT SERVICE ROUTINE
*
*
ISER EQU *
* TEST L PULSE (CA1)
 TST PIAICA
 BPL ISER7
* PROCESS L PULSE
 LDA A PIAIDA
* TEST FOR MULTIPLE L'S
 TST LBLOCK
 BNE ISER7
 INC LBLOCK
 INC LFLAG
 LDX TID
 STX XPER
 COM XPER
 COM XPER+1
 CLR CFLAG
* SERVICE LITE
 LDA A #LLITE
 JSR ALTLIT
* TEST X OR S PULSE (X=CB1)
ISER7 LDA A PIAICB
 BMI ISER0
 JMP ISER6
* SERVICE LITE
ISER0 LDA A #XLITE
 JSR ALTLIT
* PULSE FOLLOWING LASER ?
 TST LFLAG
 BEQ ISER1
* READ TIME AND RESET
 LDX TID
 STX XTIM
 COM XTIM
 COM XTIM+1
 CLR CFLAG
 INC CFLAG
* RESTART TIMER
ISER1 LDX #$FFFF
 STX TID
* PULSE FOLLOWING SOUTH ?
 TST SFLAG
 BNE ISER2
* TEST ANY ACTIVE LOOPS ?
 LDA A CS1
 ORA A CS2
 ORA A CS3
 BNE ISER9
 CLR CFLAG
 CLR LFLAG
 LDA A PIAIDB
 RTI
* UPDATE X PULSE COUNTERS
ISER9 LDX #TMT1
 JSR XPULSE
 LDX #TMT2
```

G4

```
JSR XPULSE
LDX #TMT3
JSR XPULSE
CLR LFLAG
LDA A PIAIDB
RTI
 SOUTH PULSE PROCESSING
SER2 EQU *
 ANY L PULSES THIS PASS?
CLR LGONE
TST LBLOCK
BNE ISER24
CLR CFLAG
INC LGONE
SER24 TST CS1
BEQ ISER3
TST CP1
BEQ ISER3
LDX #TMT1
JSR XPULSE
CLR SFLAG
CLR LFLAG
CLR LBLOCK
LDA A PIAIDB
RTI

SER3 TST CS2
BEQ ISER4
TST CP2
BEQ ISER4
LDX #TMT2
JSR XPULSE
CLR SFLAG
CLR LFLAG
CLR LBLOCK
LDA A PIAIDB
RTI

ER4 TST CS3
EQ ISER5
ST CP3
EQ ISER5
DX #TMT3
SR XPULSE
LR SFLAG
LR LFLAG
LR LBLOCK
DA A PIAIDB
TI

TEST WHICH TO START
ER5 TST CS1
NE ISER51
DX #TMT1
SR XPULSE
LR LBLOCK
DA A PIAIDB
TI
```



```
*
ISER51 TST CS2
 BNE ISER52
 LDX #TMT2
 JSR XPULSE
 CLR LBLOCK
 LDA A PIAIDB
 RTI
*
ISER52 TST CS3
 BNE ISER53
 LDX #TMT3
 JSR XPULSE
 CLR LBLOCK
 LDA A PIAIDB
 RTI
*
ISER53 CLR CFLAG
 JSR CLTI
 CLR LBLOCK
 LDA A PIAIDB
 RTI
* PROCESS S PULSE (S=CB2)
ISER6 BIT A #$40
 BEQ ISER10
*
 CLR SFLAG
 INC SFLAG
 LDA A PIAIDB
 RTI
* TEST OUTPUT PROCESSING
*
ISER10 LDA A TSTS
 BPL ISER20
* X PULSE INTERRUPT ?
 BIT A #2
 BNE ISER11
 LDX T1D
 LDX T3D
 RTI
* TEST S PULSE
ISER11 LDX T2D
 LDX SCNT
 DEX
 STX SCNT
 BNE ISER14
* OUTPUT S PULSE
 LDA A #SOUT
 JSR PULSE3
* RESET COUNT
 LDX #(160*2)
 STX SCNT
 LDX LVAL
 STX LCNT
 RTI
* ODD-EVEN PASS
ISER14 LDA A SCNT+1
 LSR A
```

G5

```
 BCC ISER12
 RTI
* TEST L PULSE
ISER12 LDX LCNT
 DEX
 STX LCNT
 BEQ ISER13
 RTI
* OUTPUT L PULSE
ISER13 LDA A #LOUT
 JSR PULSE3
* STROBE INT.
 RTI
*
* CRT PROCESSING
ISER20 TST TPS
 BMI ISER21
 RTI
* TEST RECEIVE
ISER21 LDA A TPS
 BIT A #1
 BEQ ISER22
* PROCESS RECEIVE
 JSR TPREC
 RTI
*TEST TRANSMIT
ISER22 BIT A #2
 BNE ISER23
* RESET TP INT.
 LDA A TPR
 RTI
* PROCESS TRANSMIT
ISER23 JSR TPXMT
 RTI
*
*
* SUBROUTINES
*
*
* X PULSE PROCESSOR
* OFFSET
*          +0=STATUS
*          +1=COUNT
*          +2=CYCLE
*          +3=LOOK-AHEAD
*          +4=HOLD
*
*
XPULSE EQU *
* TEST STATUS
 TST 0,X
 BNE XP1
* INACTIVE, TEST START
 TST SFLAG
 BNE XP2
* INACTIVE, NO START
 RTS
* START
```



```
XP2 LDA A #160
 STA A 1,X
 INC 0,X
 CLR 2,X
 CLR SFLAG
 BRA XP3
* ACTIVE, DECREMENT COUNT
XP1 DEC 1,X
XP3 CLR 3,X
* HOLD DATA ?
 TST LFLAG
 BEQ XP7
* YES, TRANSFER
 LDA A 1,X
 STA A 4,X
* LOOK AHEAD
XP7 DEC 1,X
 BNE XP4
 INC 3,X
XP4 INC 1,X
* LAST COUNT ?
 BEQ XP5
* NO
 RTS
*
* PROCESS LAST COUNT
XP5 TST SFLAG
 BNE XP6
* NO, SOUTH RESET
 CLR 0,X
 CLR 2,X
 INC 3,X
 CLR SFLAG
 RTS
* YES, SOUTH RESET
XP6 INC 2,X
 LDA A 2,X
 CMP A #8
 BEQ XP8
 LDA A #160
 STA A 1,X
 RTS
* HOLD COUNT AT 7
XP8 DEC 2,X
 LDA A #160
 STA A 1,X
 RTS
 PAGE
* MULTIPLY 40 X ROUTINE
*
* X=ADDRESS OF VALUE ON ENTRY
* A,B=ANSWER
*
MUX40 LDA A #2
 PSH A
 LDA A 0,X
 LDA B 1,X
 PSH B
```

```
 PSH A
 TSX
*
* STACK
*        +0=MSB
*        +1=LSB
*        +2=COUNT
*
* MULTIPLY A,B X 10
 ASL 1,X
 ROL 0,X
*
 LDA B 1,X
 LDA A 0,X
*
MUX41 ASL 1,X
 ROL 0,X
 DEC 2,X
 BNE MUX41
*
 ADD B 1,X
 ADC A 0,X
* MULTIPLY RESULT X 4
 ASL B
 ROL A
 ASL B
 ROL A
* CLEAN UP STACK
 INS
 INS
 INS
 RTS *
* BINARY-BCD CONVERSION
* X=ADDRESS OF RESULT(4)
* A,B=BINARY VALUE
*
BINBCD EQU *
 STX OUTX
*
 LDX #CONS
 JSR TIZ
 LDX #CONS+2
 JSR TIZ
 LDX #CONS+4
 JSR TIZ
 LDX OUTX
 STA B 0,X
 RTS
*
CONS FDB 1000,100,10
*
* TEST VALUE DIVISIBLE
*
TIZ PSH B
 PSH A
 LDA A 0,X
 LDA B 1,X
 PSH B
```



```
 PSH A
 CLR A
 PSH A
*
* STACK
*        +0=COUNT
*        +1=MS CONST
*        +2=LS CONST
*        +3=MS VALUE
*        +4=LS VALUE
*
 TSX
 LDA B 4,X
 LDA A 3,X
TIZ2 SUB B 2,X
 SBC A 1,X
 BCS TIZ1
 INC 0,X
 BRA TIZ2
* FAIL SUBTRACT,RESTORE
TIZ1 ADD B 2,X
 ADC A 1,X
 STA B 4,X
 STA A 3,X
* RECOVER COUNT
 LDA A 0,X
* STORE COUNT
 LDX OUTX
 STA A 0,X
 INX
 STX OUTX
* RESTORE STACK
 INS
 INS
 INS
 PUL A
 PUL B
 RTS
*
* TIMER INITIALIZATION
*
TIMI LDA A #T3II
 STA A TCR13
 LDA A #T2II
 STA A TCR2
 STA A OLDCR2
 LDA A #T1II
 STA A TCR13
* SETUP TIMER #3 FOR 10 USEC OUTPUT
 LDX #9
 STX T3D
* SETUP TIMER #1 FOR COUNTDOWN
 LDX #$FFFF
 STX T1D
 RTS
*
* SETUP TIMER #2 (INT)
*
```

```
SET2 LDA A #T2IE
 STA A OLDCR2
 LDA B TSTS
* STORE TIME AND START
 STX T2D
 STA A TCR2
 RTS
*
*
* 16 BIT UNSIGNED DIVIDE
* A,B DIVIDED BY (X),(X+1)
* RESULT IN A,B
*
DIV16 PSH B
 PSH A
 LDA A 0,X
 LDA B 1,X
 PSH B
 PSH A
 DES
 TSX
 LDA A #1
 TST 1,X
 BMI DIV153
DIV151 INC A
 ASL 2,X
 ROL 1,X
 BMI DIV153
 CMP A #17
 BNE DIV151
DIV153 STA A 0,X
 LDA A 3,X
 LDA B 4,X
 CLR 3,X
 CLR 4,X
*
* STACK
*
*       +0=COUNT
*       +1=MS DIVISOR
*       +2=LS DIVISOR
*       +3=MS DIVIDEND
*       +4=LS DIVIDEND
*
DIV163 SUB B 2,X
 SBC A 1,X
 BCC DIV165
 ADD B 2,X
 ADC A 1,X
 CLC
 BRA DIV167
DIV165 SEC
DIV167 ROL 4,X
 ROL 3,X
 LSR 1,X
 ROR 2,X
 DEC 0,X
 BNE DIV163
```



```
* CLEAN UP STACK
 INS
 INS
 INS
 PUL A
 PUL B
 RTS
*
* READ COMM PROCESSOR
*
RCMP EQU *
 CLR AUTOF
* TEST DATA REQUEST (CB1)
RCM1 LDA A PIA2CB
 BPL RCM1
* RECEIVE DATA
RCM2 LDA A PIA2DB
* TEST STX
 CMP A #1
 BNE RCM8
 LDX #CMPIN
* SEND DATA ACCEPTED (CB2)
RCM7 LDA A PIA2DB
 JSR STROB2
* TEST DATA REQUEST (CB1)
RCM4 TST PIA2CB
 BPL RCM4
* GET DATA
RCM5 CLR A
 LDA A PIA2DB
 BEQ RCM5
 CMP A #3
 BEQ RCM3
* TEST AUTO FLAG (ENQ)
 CMP A #5
 BNE RCM6
 INC AUTOF
 BRA RCM7
* DATA
RCM6 SUB A #$30
 STA A 0,X
 INX
 BRA RCM7
* SEND DATA ACCEPTED (CB2)
RCM3 JSR STROB2
 LDA A PIA2DB
 CLC
 RTS
RCM8 SEC
 RTS
*
* OUTPUT STROBE PIA2CB
*
STROB2 LDA A #%00110100
 STA A PIA2CB
 NOP
 LDA A #%00111100
 STA A PIA2CB
```

```
RTS

 OUTPUT STROBE PIA2CA

TROA2 LDA A #%00110100
STA A PIA2CA
NOP
LDA A #%00111100
STA A PIA2CA
RTS

 TRANSFER CHAR STRING
 DEST=DESTINATION,X=SOURCE
 B=CHAR CNT

CS STX SAVEX
CS1 LDX SAVEX
LDA A 0,X
INX
STX SAVEX

LDX DEST
STA A 0,X
INX
STX DEST
DEC B
BNE TCS1
RTS
*
* COMPUTE AZIMUTH
* AZ=DEFL+REF
*
:AZ EQU *
* FETCH DEFLECTION
 LDX #DEFL
 STX A1
* FETCH REFERENCE
 LDX #REFBUF
 STX A2
*
 LDX #RESULT+3
 JSR BCDADD
* ADJUST FOR ROLLOVER
 LDX #AZ
 JSR ADJ
 RTS
*
*
* ADD 2-4 DIGIT BCD VALUES
* A1=ADDRESS OF VALUE 1 MSB
* A2=ADDRESS OF VALUE 2 MSB
* X=ADDRESS OF SUM
*
BCDADD STX SAVEX
 CLR SIG
* FIX A1
 LDX A1
 JSR FIXX
```



```
 STX A1
* FIX A2
 LDX A2
 JSR FIXX
 STX A2
 CLC
* GET FIRST VALUE
 LDA B #3
BCA1 LDX A1
 LDA A 0,X
 DEX
 STX A1
* ADD SECOND
 LDX A2
 ADC A 0,X
 JSR JOCK
 DEX
 STX A2
* STORE SUM
 LDX SAVEX
 STA A 0,X
 DEX
 STX SAVEX
*
 DEC B
 BNE BCA1
*
 LDX A1
 LDA A 0,X
 LDX A2
 ADC A 0,X
 JSR JOCK
 BCC BCA2
 INC SIG
BCA2 LDX SAVEX
 STA A 0,X
 RTS
*
* FIX VALUE OF X (3X)
*
FIXX LDA B #3
FIX1 INX
 DEC B
 BNE FIX1
 RTS
*
* ADJUST FOR BCD CARRY
*
JOCK CMP A #9
 BGT JOCK1
 CLC
 RTS
JOCK1 ADD A #6
 AND A #$F
 SEC
 RTS
*
* ADJUST FOR ROLLOVER
```

```
*
ADJ STX SAVEX
 TST SIG
 BNE ADJ1
* TEST FOR >6400 ROLLOVER
 LDX #RESULT
 JSR TEST64
 BCC ADJ2
* FIX > 6400 ROLLOVER
ADJ1 LDX #RESULT
 STX A1
 LDX #CON64
 STX A2
 LDX SAVEX
 JSR FIXX
 JSR BCDADD
 RTS
* FIX <6400 VALUE
ADJ2 LDX SAVEX
 JSR XFER
 RTS
*
CON64 FCB 3,6,0,0
*
*
* TEST IF >6400
*
TEST64 LDA A 0,X
 CMP A #6
 BGT T641
 BLT T642
 LDA A 1,X
 CMP A #4
 BGE T641
T642 CLC
 RTS
T641 SEC
 RTS
*
* TRANSFER ARRAY FROM "RESULT" TO (X)
*
XFER STX DEST
 LDX #RESULT
 LDA B #4
 JSR TCS
 RTS
*
* OUTPUT TO COMM PROCESSOR
*
SCMP EQU *
* SET MESSAGE COUNT
 LDA B #8
* SEND STX
 LDA A #1
 STA A PIA2DA
 JSR XRUX
* SEND MESSAGE
 LDX #CMPOUT
```



```
SCM2 LDA A 0,X
 ADD A #$30
 STA A PIA2DA
 JSR XRUX
 INX
 DEC B
 BNE SCM2
* SEND ETX
 LDA A #3
 STA A PIA2DA
 JSR XRUX
 LDA A PIA2DA
 RTS
*
* STROBE AND TEST OUTPUT
*
XRUX EQU *
* OUTPUT READY TO SEND (CA2)
 LDA A PIA2DA
 JSR STROA2
* WAIT DATA ACCEPTED (CA1)
XRUX1 TST PIA2CA
 BPL XRUX1
 RTS
*
* TEST PULSE ON
* A=DATA WORD
*
PON ORA A PIA3DB
 STA A PIA3DB
 RTS
*
* TEST PULSE OFF
* A=DATA WORD
*
POFF COM A
 AND A PIA3DB
 STA A PIA3DB
 RTS
*
* TEST PULSE STROBE
*
PULSE3 PSH A
 ORA A PIA3DB
 STA A PIA3DB
 PUL A
 COM A
 AND A PIA3DB
 STA A PIA3DB
 RTS
*
* INITIAL PIAS
*
PIAS EQU *
* PIA1-A INPUT,B OUTPUT
 CLR PIA1CA SELECT DD
 CLR PIA1CB SELECT DD
 CLR PIA1DA SET A INPUT
```

```
 LDA A #$FF
 STA A PIA1DB SET B OUTPUT
* ENABLE CA1 INTERRUPT AND CB2 (INPUT)
 LDA A #%00000101
 STA A PIA1CA
* ENABLE CB1 AND CB2 INTERRUPT
 LDA A #%00001101
 STA A PIA1CB
* CLEAR FLAGS
 LDA A PIA1DA
 LDA A PIA1DB
*
* PIA2-A INPUT/OUTPUT FOR COMM PROC
* A=OUTPUT
* B=INPUT
* SET DD
 LDX #PIA2DA
 CLR 2,X
 CLR 3,X
* INPUT-B
 CLR 1,X
* OUTPUT-A
 LDA A #$FF
 STA A 0,X
* CONTROL
 LDA A #%00111100
 STA A 2,X
 STA A 3,X
* CLEAR FLAGS
 LDA A 0,X
 LDA A 1,X
*
* PIA3 SWITCH / OUTPUT
*
 LDX #PIA3DA
* SELECT DD
 CLR 2,X
 CLR 3,X
* A=INPUT
 CLR 0,X
* B=OUTPUT
 LDA A #$FF
 STA A 1,X
* CONTROLS
 LDA A #$36
 STA A 2,X
 STA A 3,X
* RESETS
 LDA A 0,X
 LDA A 1,X
* CLEAR LAMPS
 LDA A #$0F
 JSR PON
*
*
* SETUP ACIA
* ENABLE RECV INT.
*
```



```
 LDA A #3
 STA A TPC
 LDA A #RIE
 STA A TPC
 RTS
*
* CLEAR BUFFER ROUTINE
*
CLBF CLR 0,X
 INX
 CPX BEND
 BNE CLBF
 RTS
*
* CLEAR AND INITIAL TIMERS
*
CLTI LDX #TEND
 STX BEND
 LDX #TMTB
 JSR CLBF
*
 LDA A #1
 STA A CP1
 STA A CP2
 STA A CP3
 RTS
*
* TEST PORT RECEIVE
*
TPREC LDX #TPC
 JSR CRLF
 JSR AO1
 BCC TPRE1
 JSR AOOL
* TEST DATA INPUT ?
TPRE1 TST INF
 BNE TPRE7
* =T (TIMER TABLE)
 CMP A #'T
 BNE TPRE2
 LDX #TMTB
 STX TPPTR
 LDX #TEND
 STX TPPTE
 LDA A #5
 BRA TPRE3
* =F (FLAG BUFFER)
TPRE2 CMP A #'F
 BNE TPRE4
 LDX #FLAG
 STX TPPTR
 LDX #FEND
 STX TPPTE
 LDA A #1
 BRA TPRE3
* =D (DATA BUFFERS)
TPRE4 CMP A #'D
 BNE TPRE5
```

```
 LDX #REFBUF
 STX TPPTR
 LDX #BUFEND
 STX TPPTE
 LDA A #4
 BRA TPRE3
* =L (L VAL ENTRY)
TPRE5 CMP A #'L
 BNE TPRE6
* SET INPUT FLAG
 CLR INF
 INC INF
* SET INPUT POINTERS
 JSR CLTB
 LDX #TBUF
 STX TPPTR
 LDX #TBUF+4
 STX TPPTE
 RTS
* GET INPUT
TPRE7 LDX TPPTR
* TEST EOT
 STA A 0,X
 CMP A #4
 BEQ TPRE8
 INX
 STX TPPTR
 CPX TPPTE
 BEQ TPRE8
 RTS
* LAST CHAR
TPRE8 CLR INF
 JSR DGN
 BCS TPRE9
 STA A LVAL
 STA B LVAL+1
TPRE9 RTS
*
* SETUP SPACE COUNT
TPRE3 STA A SPC
 STA A ASP
* OUTPUT CR/LF
 LDX #TPC
 JSR CRLF
* INH REC/ENB XMIT
 LDA A #XIE
 STA A TPC
TPRE6 LDA A TPR
 RTS
*
* TEST PORT TRANSMIT
*
TPXMT TST ASP
 BEQ TPXM1
 DEC ASP
TPXM4 LDX TPPTR
 CPX TPPTE
 BEQ TPXM2
```



```
 LDA A 0,X
 INX
 STX TPPTR
 ADD A #$30
TPXM3 LDX #TPC
 JSR AOOL
 RTS
* FIX SPACE
TPXM1 LDA A SPC
 STA A ASP
 LDA A #$20
 BRA TPXM3
* LAST CHAR-INH XMIT/ENB RECV
TPXM2 LDA A #RIE
 STA A TPC
 LDA A TPR
 RTS
*
* EXEC SUBROUTINES
*
*
* CLEAR CHAR BUFFER
*
CLTB LDX #TBUF
 LDA B #4
 LDA A #$20
CLTB1 STA A 0,X
 INX
 DEC B
 BNE CLTB1
 RTS
*
* INPUT FROM ACIA
*
AOI LDA A 0,X
 BIT A #1
 BEQ AOI1
 LDA A 1,X
 SEC
 RTS
AOI1 CLC
 RTS
*
*
* OUTPUT TO ACIA
*
AOO PSH A
 LDA A 0,X
 BIT A #2
 PUL A
 BEQ AOO1
 STA A 1,X
 SEC
 RTS
AOO1 CLC
 RTS
*
* CR/LF ROUTINE
```

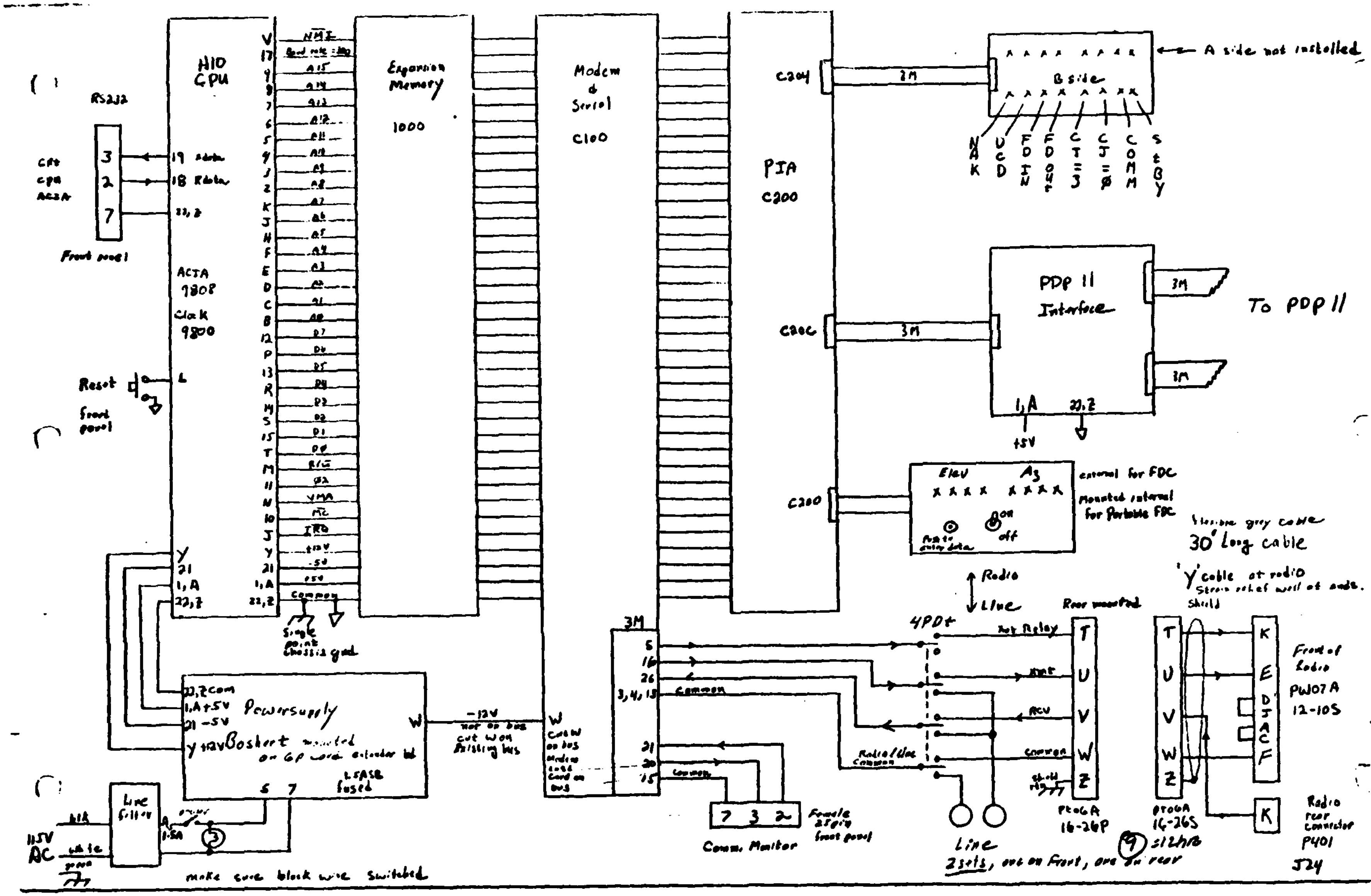
```
*
CRLF LDA A #$D
 JSR AOOL
 LDA A #$A
 JSR AOOL
 RTS
*
* LOOP ON OUTPUT
*
AOOL JSR AOO
 BCC AOOL
 RTS
*
* CONVERT STRING TO DECIMAL NO.
*
* STRING IN TBUF
* ON RETURN A=MSBY,B=LSBY
*
DGN LDX #TBUF
 LDA B #4
 CLR MSBY
 CLR LSBY
DGN2 LDA A O,X
 CMP A #$20
 BNE DGN1
 INX
 DEC B
 BNE DGN2
*
DGNE SEC
 RTS
*
DGN1 CMP A #4
 BEQ DGNE
 SUB A #$30
 BLT DGNE
 CMP A #9
 BGT DGNE
 STA A LSBY
 INX
 DEC B
 BEQ DGNE
DGN4 LDA A O,X
 CMP A #4
 BNE DGN3
 LDA A MSBY
 LDA B LSBY
 CLC
 RTS
DGN3 SUB A #$30
 BLT DGNE
 CMP A #9
 BGT DGNE
 STA A TMP
 JSR MIOX
 INX
 DEC B
 BEQ DGNE
```



```
 BRA DGN4
*
* MULTIPLY MSBY,LSBY X 10 + TMP
*
MIOX LDA A MSBY
 LDA B LSBY
 CLC
 ASL B
 ROL A
 ASL B
 ROL A
 ADD B LSBY
 ADC A #0
 ASL B
 ROL A
*
 ADD B TMP
 ADC A #0
 STA A MSBY
 STA B LSBY
 RTS
*
* ALTERNATE LIGHT (ON-OFF)
*
ALTLIT EOR A PIA3DB
 STA A PIA3DB
 RTS
* SYSTEM VECTORS
 ORG $FFF8
 FDB ISER
 FDB STRT
 FDB ISER
 FDB STRT
 END
```

APPENDIX H

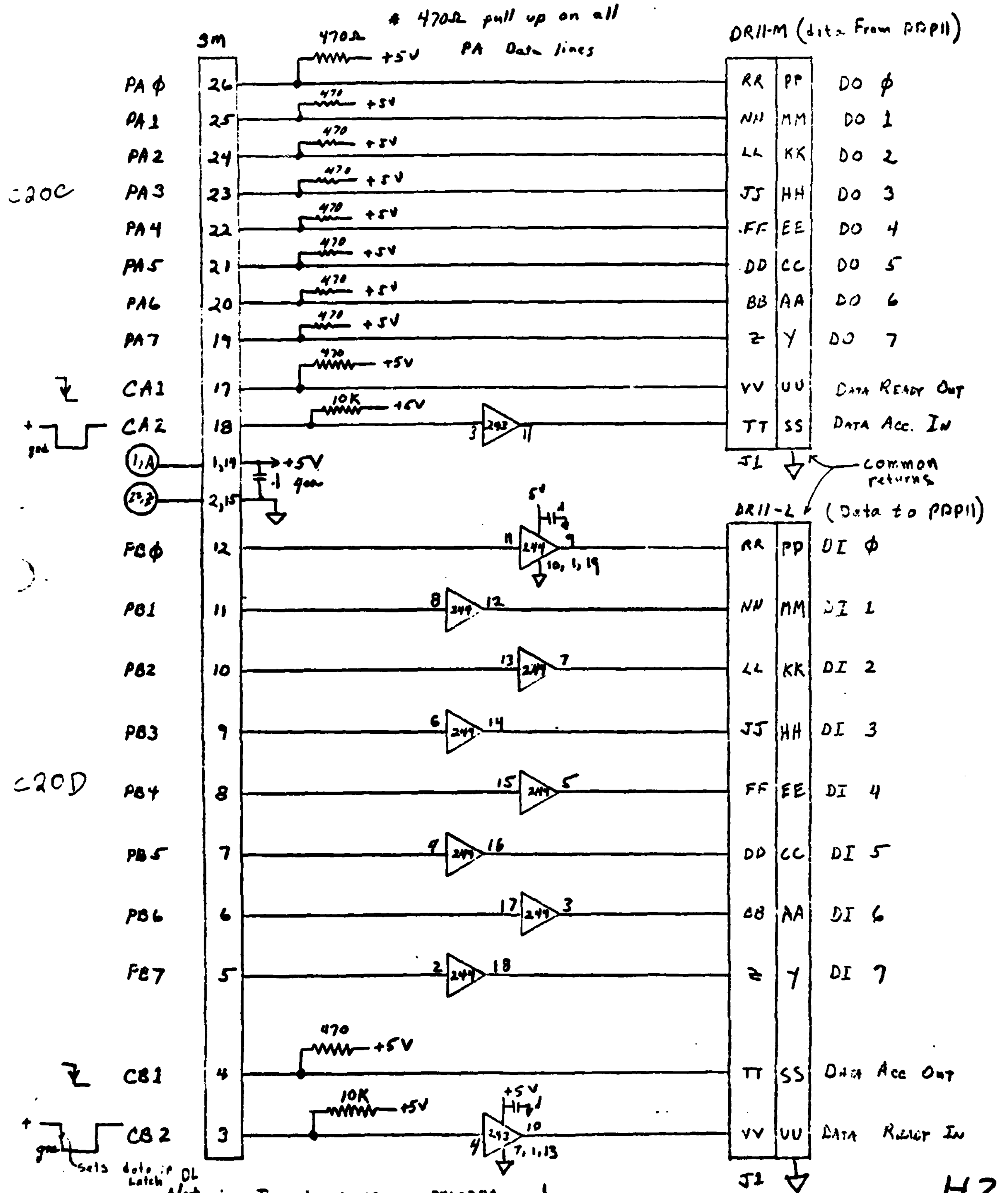
FDCOM SCHEMATICS


A side not installed
B side
To PDP 11
PDP 11 Interface
PIA
C200
Modem & Serial
C100
Expansion Memory
1000
H10 CPU
ACIA 7808
Clock 9800
RS232
Front panel
Reset
Powersupply
Line filter
115V AC
Comm. Monitor
30' long cable
Radio
Line
4PDT
Rear mounted
Front of Radio
PW07A
12-10S
make sure black wire switched

H1

) => Card Edge connector

H10 PIA To PDP 11 Interface



Note: Tie pins 1+19 on 74LS244 and pins 1+13 on 74LS243 to gnd.

Vcc + GND on 243+244 as per norm.

H2

APPENDIX I

FDCOM CONTROL PROGRAM
SOURCE LISTING

```
 NAM FDCOM
*
* AGLS COMMUNICATIONS
* FIRE DIRECTION CENTER
* REVISED 2/28/79 1500
*
*
FLAGC EQU 1
*
* PIA EQUATES
* PIA1=UNASSIGNED
PIA1DA EQU $C200
PIA1DB EQU PIA1DA+1
PIA1CA EQU PIA1DA+2
PIA1CB EQU PIA1DA+3
IN1 EQU PIA1DA
OT1 EQU PIA1DB
*
*
* PIA2=CONTROLS AND DISPLAYS
* A SIDE=SW. INPUTS
* B SIDE=LED OUTPUTS
*
PIA2DA EQU $C204
PIA2DB EQU PIA2DA+1
PIA2CA EQU PIA2DA+2
PIA2CB EQU PIA2DA+3
IN2 EQU PIA2DA
OT2 EQU PIA2DB
*
* PIA3=UNASSIGNED
*
PIA3DA EQU $C208
PIA3DB EQU PIA3DA+1
PIA3CA EQU PIA3DA+2
PIA3CB EQU PIA3DA+3
IN3 EQU PIA3DB
OT3 EQU PIA3DB
*
* PIA4=FDC PDP11/34
* A SIDE=DATA INPUT
* B SIDE=DATA OUTPUT
*
PIA4DA EQU $C20C
PIA4DB EQU PIA4DA+1
PIA4CA EQU PIA4DA+2
PIA4CB EQU PIA4DA+3
IN4 EQU PIA4DA
OT4 EQU PIA4DB
EOT EQU 04
*
*
* LED WORDS (PIA2)
*
CLED1 EQU %00000001 (STBY)
CLED2 EQU %00000010 (COMM)
CLED3 EQU %00000100 (CJ=0)
CLED4 EQU %00001000 (CJ=3)
```



```
CLED5 EQU %00010000 (FD OUT)
CLED6 EQU %00100000 (FD IN)
CLED7 EQU %01000000 (CARRIER DET.)
CLED8 EQU %10000000 (NAK)
*
*
* PERHIPERAL EQUATES
*
* COMM ACIA
COMC EQU $C100
COMS EQU COMC
COMX EQU COMC+1
COMR EQU COMC+1
*
*
* TEST PORT ACIA
TPC EQU $9808
TPS EQU TPC
TPX EQU TPC+1
TPR EQU TPC+1
*
*
* TIMER
TCR13 EQU $9800
TSTS EQU $9801
TCR2 EQU $9801
T1D EQU $9802
T2D EQU $9804
T3D EQU $9806
* TIMER CONSTANTS
T1IE EQU %01000000
T1II EQU %00000000
*
T2IE EQU %01000001
T2II EQU %00000001
*
T3IE EQU %11000011
T3II EQU %10000011
*
* TIMEOUT CONSTANTS (.1 SECONDS)
CONTT EQU 0200 CONNECT TRY
CUD EQU 0007 CARRIER UP DELAY
CDD EQU 0005 CARRIER DN DELAY
CDT EQU 36000 CARRIER DETECT
MSGT EQU 0600 MSG WAIT TIME
* COMM EQUATES
*
SOH EQU 1 START OF HEADER
SMFC EQU $42 SERVICE MSG FORMAT CODE
IMFC EQU $48 INFO. MSG FORMAT CODE
SLFC EQU $43 SELECT MSG. FORMAT CODE
CMFC EQU $44 CONTROL MSG FORMAT CODE
SC EQU $41 SEQUENCE CODE
AC EQU $40 ADDRESS CODE
IC EQU $40 IDENT CODE
STX EQU 2 START OF TEXT
ETX EQU 3 END OF TEXT
NOC EQU $40 NO REQ OPERATION CODE
```

```
SMTY EQU $40 SERVICE MSG. TYPE
SLTY EQU $42 SELECT MSG TYPE
DOC EQU $42 DATA REQ OPER CODE
RROC EQU $46 READY RESP. OPER. CODE
WOC EQU $44 WAIT OPER. CODE
* ACIA INTERRUPT CONSTANTS
XIE EQU %00101001 XMIT INT ENB
RIE EQU %10001001 RECV INT ENB
NIE EQU %00001001 INT. OFF
*
*
 IFNE FLAGC
*
*
* COMM INTERRUPT CONSTANTS
CNIE EQU %01001001
CRIE EQU %11001001
CXIE EQU %00101001
RTS EQU %00001001
 ENDC
*
*
 IFEQ FLAGC
*
*
CNIE EQU %00001001
CRIE EQU %10001001
CXIE EQU %00101001
RTS EQU %00001001
*
 ENDC
 PAGE
 ORG $1000
* COMM RECEIVE BUFFER
RBUF RMB 60
REND EQU *
RDATA EQU RBUF+7
* COMM TRANSMIT BUFFER
XBUF RMB 60
XEND EQU *
XDATA EQU XBUF+7
*
* PDP 11 "FROM" BUFFER
*
FDCF EQU *
KEYS RMB 3
CODE RMB 2
DATA RMB 17
FDCFE EQU *
*
* PDP 11 "TO" BUFFER
*
FDCT EQU *
STAT RMB 4
MSG RMB 45
FDCTE EQU *
*
*
```



```
* OLD FIRE ORDER BUFFER
*
OLDFO RMB 12
*
*
XIDLE RMB 2 IDLE VECTOR FLAG
*
*
* FLAGS
FLAG EQU *
ETXF RMB 1 END TEXT FLAG
CJ RMB 1 COMM STEERING
ZRFD RMB 1 REQ. DISC. FLAG
ZIDM RMB 1 ID MESSAGE FLAG
XPASS RMB 1 XMIT FIRST PASS
RJ RMB 1 RECD DATA FLAG
RLSC RMB 1 LAST RECD SEQ CODE
ZRRF RMB 1 READY FOR RESP
VDF RMB 1 VALID DATA FLAG
CONN RMB 1 CONNECT FLAG
FDCA RMB 1 DATA AVAIL FLAG
FDCC RMB 1 FDC DATA CALL
OUTF RMB 1 WRITE FDC GET OUT
PASS RMB 1 READ FDC FIRST PASS
FILLF RMB 1 FILL CHAR FLAG(PXMT)
CTSUBY RMB 1 CLR TO SEND UP BUSY
CTSDBY RMB 1 CLR TO SEND DN BUSY
TDCDBY RMB 1 CARRIER DET. BUSY
DISF RMB 1 DISCONNECT READY
INF RMB 1 CRT INPUT MODE
ATORR RMB 1 AUTO READY REQUEST
ATODR RMB 1 AUTO DATA REQUEST
INEF RMB 1 INPUT ERROR (FDC)
FIBSY RMB 1 FDC INPUT BUSY
FOBSY RMB 1 FDC OUTPUT BUSY
CFFF RMB 1 CHECK FIRE PENDING
FEND EQU *
* CONSTANTS
BEND RMB 2 BUFFER END POINTER
* BUFFER POINTERS
BR1 RMB 2 RECEIVE
BX1 RMB 2 TRANSMIT
XBCC RMB 1 BCC XMIT
RBCC RMB 1 BCC RECEIVE
RERR RMB 1 RECV ERROR CODE
RSTAT RMB 1 RECV STATUS WORD
TMPX RMB 1 RJ TEMP INDEX
RXFC RMB 1 RECD FORMAT CODE
RXOC RMB 1 RECD OPER CODE
OLDSC RMB 1 SEQUENCE CODE SAVE
SAVES RMB 2 STACK SAVE
ISAVES RMB 2 INT STACK SAVE
SAVEX RMB 2 X REG SAVE
SAVA RMB 1 SAVE A REG
SAVB RMB 1 SAVE B REG
OLDCR2 RMB 1 TIMER CR#2 WORD
VECT1 RMB 2 TIMER INT VECTOR 1
VECT2 RMB 2 TIMER INT VECTOR 2
```

PAGE 005 PFDCOM .SA:1

```
TPPTR RMB 2 TEST PORT POINTER
TPPTE RMB 2 TEST PORT END
SPC RMB 1 SPACE COUNT
ASP RMB 1 SPACE COUNT
RSWD RMB 1 RECD STATUS WORD (4CHAR)
ASWD RMB 1 ACK STATUS WORD
WAITC RMB 1 WAIT COUNTER
TABP RMB 2 TABLE POINTER
DEST RMB 2 CHAR XFER DESTINAION
FPTR RMB 2 FDC ADDR PTR (RECV)
FPTT RMB 2 FDC ADDR PTR (TRAN)
CEND EQU *-1
*
* INTERRUPT DRIVEN TIMERS (100 MSEC)
*
* TIMER TABLE (DECREMENT)
TMTB EQU *
TF1 RMB 1
TIM1 RMB 2
*
TF2 RMB 1
TIM2 RMB 2
*
TF3 RMB 1
TIM3 RMB 2
*
TF4 RMB 1
TIM4 RMB 2
*
TF5 RMB 1
TIM5 RMB 2
 PAGE
*
* AGLS FIRE DIR. CNTR. PROCESSOR
*
* START VECTOR FOR POWER UP OR RESET
*
 ORG $F000
STRT EQU *
* H-10 DEBUG
 LDS #$7F
* COMM IDLE VECTOR
IDLE EQU *
 STX XIDLE
* SETUP PIAS
 JSR PIAS
* CLEAR BUFFERS
* DATA BUFFERS
 LDX #FDCTE
 STX BEND
 LDX #RBUF
 JSR CLBF
* FLAG BUFFER
 LDX #FEND
 STX BEND
 LDX #FLAG
 JSR CLBF
* CONSTANT BUFFER
```

PAGE 006 PFDCOM .SA:1

```
 LDX #CEND
 STX BEND
 LDX #BR1
 JSR CLBF
* CLEAR TIMERS
 JSR CLTM
* SEED SEQ. CODE
 LDA A #SC
 STA A OLDSC
* ENABLE INTERRUPTS
 CLI
*
*
* ENABLE RECV. INTERRUPTS
IDLE1 LDA A #2
 STA A CJ
* CLEAR TIMERS
 JSR CLR1
 JSR CLR2
* INITIAL BUFFER POINTER
 LDX #RBUF
 STX BR1
*
*
 CLR CONN
* TURN ON "STBY",OFF "COMM"
 LDA A #CLED1
 JSR LEDON
 LDA A #CLED2
 JSR LEDOFF
*
 JSR COMIX
*
*
* SYSTEM ACTIVE LOOP
*
*
LOOP EQU *
 LDA A #CLED3
 LDX #LEDOFF
 TST CJ
 BNE LOOP1
* TURN ON "CJ=0"
 LDX #LEDON
LOOP1 JSR 0,X
 CPX #LEDON
 BNE LOOP2
* FDC DATA CALL?
 TST FDCC
 BEQ LOOP2
* SETUP XMIT MSG.
 TST INEF
 BNE LOOP
 JSR SETXMT
 TST INEF
 BNE LOOP
 CLR FDCC
* TRUN FDC INT BACK ON (CA1)
```

I 3

```
 LDA A #%00111101
 STA A PIA4CA
 JMP XIT
* FDC DATA AVAILABLE?
LOOP2 TST FDCA
 BEQ LOOP5
* TRANSFER DATA TO FDC
 JSR FDOUT
LOOP5 LDA A CJ
 CMP A #3
 BNE LOOP4
* TURN ON "CJ=3"
 LDA A #CLED4
 JSR LEDON
 JMP C3
* DISCONNECT PENDING?
LOOP4 EQU *
* TURN OFF "CJ=3"
 LDA A #CLED4
 JSR LEDOFF
 TST DISF
 BEQ LOOP
* YES,WAIT TILL SENT
 LDA A CJ
 CMP A #1
 BEQ LOOP
*
 LDX #*
 JMP IDLE
 PAGE
* SUBROUTINES
*
*
* CLEAR BUFFER ROUTINE
* X= BUFFER START
* BEND=BUFFER END
*
*
CLBF CLR 0,X
 INX
 CPX BEND
 BNE CLBF
 RTS
*
* PROCESS TRANSMIT
*
PXMT LDX BXI
 TST FILLF
 BGT PXM4
 BMI PXM5
 LDA A 0,X
 BEQ PXM1
 INX
 STX BXI
*=SOH?
 CMP A #SOH
 BNE PXM2
* YES, =SOH
```



```
 CLR XBCC
 BRA PXM3
* = DATA
PXM2 TAB
 EOR B XBCC
 STA B XBCC
* TRANSMIT CHAR
 JSR XMIT
PXM3 CLC
 RTS
* LAST CHAR
PXM1 LDA A XBCC
 JSR XMIT
 INC FILLF
 CLC
 RTS
* TRANSMIT FILL CHAR
PXM4 LDA A #$20
 JSR XMIT
 NEG FILLF
 CLC
 RTS
* SECOND TIME AROUND
PXM5 LDA A #$20
 JSR XMIT
 CLR FILLF
 LDX #XBUF
 STX BXI
 SEC
 RTS
*
* PROCESS RECEIVE
*
PREC JSR RECV
 BCS PRE6
* CLEAR XMIT FLAG
 CLC
 RTS
PRE6 LDX BRI
* CHAR = BCC?
 INC ETXF
 BEQ PREI
* NO, = DATA
 CLR ETXF
 STA A 0,X
 INX
 STX BRI
* TEST RECIEVE STATUS
 JSR JSTS
* TEST BUFFER OVERRUN
 JSR BOVR
* CHAR = SOH?
 CMP A #SOH
 BEQ PRE2
* CHAR = ETX?
 CMP A #ETX
 BNE PRE3
 DEC ETXF
```

```
PRE3 EOR A RBCC
 STA A RBCC
 CLC
 RTS
* FIRST CHAR
PRE2 CLR RBCC
 CLR RERR
 LDX #RBUF
 STA A 0,X
 INX
 STX BRI
 INC VDF
 CLC
 RTS
* LAST CHAR
PREI EOR A RBCC
 BNE PRE4
PRE5 LDX #RBUF
 STX BRI
 CLR VDF
 SEC
 RTS
* BCC ERROR
PRE4 LDA A #$10
 EOR A RERR
 STA A RERR
 BRA PRE5
* RECEIVE CHAR ROUTINE
*
RECV LDA A COMS
 BIT A #1
 BNE RECVI
* NO DATA CALL
 LDA A COMR
 CLC
 RTS
* DATA CALL
RECVI STA A RSTAT
* READ CHAR
 LDA A COMR
 SEC
 RTS
*
* TEST RECEIVE STATUS
*
JSTS PSH A
 LDA A RSTAT
* PARITY ERROR?
 BIT A #$40
 BEQ JSTSI
 LDA B #$18
 EOR B RERR
 STA B RERR
* OVER RUN ERROR?
JSTSI BIT A #$20
 BEQ JSTS2
 LDA B #8
 EOR B RERR
```



```
 STA B RERR
* FRAMING ERROR
JSTS2 BIT A #$10
 BNE JSTS3
 PUL A
 RTS
JSTS3 LDA B #$20
 EOR B RERR
 STA B RERR
 PUL A
 RTS
*
* TRANSMIT CHAR.
*
XMIT LDA B COMS
 BIT B #2
 BNE XMITI
* NO DATA CALL, RESET RECVR
 LDA A COMR
 RTS
* DATA CALL
XMITI STA A COMX
 RTS
*
* INHIBIT XMIT/ENB RECV
*
COMIX LDA A #CRIE RECV INT ENB
 STA A COMC
 LDA A COMR RESET
 RTS
*
* INHIBIT COMM INT
COMOFF LDA A #CNIE
 STA A COMC
 LDA A COMR
 RTS
*
* INHIBIT RECV/ENB XMIT
*
* PREP COMM
COMIR EQU *
* MAKE SURE RECD CAR ON
COMI3 LDA A COMS
 BIT A #4
 BEQ COMI3
 LDA A #RTS
 STA A COMC
* WAIT FOR CARRIER UP
COMII JSR CTSU
 BCC COMII
 LDA A #SOH
COMI2 LDA B COMS
 BIT B #2
 BEQ COMI2
 STA A COMX
 LDA A #CXIE XMIT INT ENABLE
 STA A COMC
 RTS
```

```
*
* TEST BUFFER OVERRUN
*
BOVR CPX #REND
 BEQ BOVR1
 RTS
* OVERRUN HAS OCCURRED
BOVR1 LDA B #8
 EOR B RERR
 STA B RERR
 DEX
 STX BR1
 RTS
* COMMUNICATION POLL
*
CPOLL LDA A CJ
 CMP A #1
 BEQ CPOL1
 CMP A #2
 BEQ CPOL2
* RESET INTERRUPT
 LDA A COMR
 RTS
* TRANSMIT LOOP
CPOL1 EQU *
* FIRST PASS?
 INC XPASS
 BNE CPOL6
* SETUP MESSAGE HEADERS
CPOL11 JSR SETUM
* SWAP SEQ. CODES
 JSR SSC
* PRCESS TRANSMIT
CPOL6 CLR XPASS
 JSR PXMT
 BCS CPOL5
 RTS
* SWITCH TO RECEIVE
CPOL5 EQU *
* CLEAR RECEIVE BUFF
 LDX #RBUF
 JSR CLRB
 LDX #RBUF
 STX BR1
* SWITCH COMM INT
 JSR COMIX
 LDA A #2
 STA A CJ
 RTS
*
* RECEIVE LOOP
CPOL2 EQU *
*
*
 IFNE FLAGC
* TEST CARRIER
 LDA A #CLED7
 JSR LEDOFF
```



```
 JSR TDCD
 BCS CPOL10
 LDA A COMR
 RTS
*
 ENDC
*
* PROCESS RECEIVE
CPOL10 LDA A #CLED7
 JSR LEDON
 JSR PREC
 BCS CPOL7
 RTS
* SWITCH TO UNPACK
CPOL7 LDA A #3
 STA A CJ
 JSR COMOFF
 LDA A #CLED7
 JSR LEDOFF
 RTS
*
* SETUP MESSAGE ROUTINE
*
SETUM EQU *
* TEST RFD FLAG
 TST ZRFD
 BEQ SETUM1
* SETUP RFD SERVICE MESSAGE
 LDA B #$44
 JSR SSM
* TEST I.D. MESSSAGE
SETUM1 INC ZIDM
 BNE SETUM2
* SETUP SELECT
 LDA B #$42
 JSR SSLM
SETUM2 CLR ZIDM
* TEST READY FOR RESPONSE
 INC ZRRF
 BNE SETUM3
* SETUP READY FOR RESPONSE
 LDA B #$46
 JSR SIM
* SETUP XMIT POINTER
SETUM3 CLR ZRRF
 LDX #XBUF
 STX BX1
 RTS
*
* CARRIER UP DELAY
*
CTSU EQU *
 TST CTSUBY
 BNE CTSU1
* SETUP TIMEOUT VECTOR
 LDX #CUD
 STX TIM2
 INC TF2
```

```
 INC CTSUBY
* TEST CTS-UP
CTSU1 TST TF2
 BEQ CTSU2
 CLC
 RTS
* INHIBIT TIMER
CTSU2 CLR CTSUBY
 SEC
 RTS
*
* CARRIER DOWN DELAY
*
CTSD EQU *
 TST CTSDBY
 BNE CTSD1
* SETUP TIMEOUT VECTOR
 LDX #CDD
 STX TIM2
 INC TF2
 INC CTSDBY
* TEST CTS-DOWN
CTSD1 TST TF2
 BEQ CTSD2
 CLC
 RTS
* INHIBIT TIMER
CTSD2 CLR CTSDBY
 SEC
 RTS
*
* TEST CARRIER DETECT
*
TDCD EQU *
 TST TDCDBY
 BNE TDCD2
* SETUP TIMEOUT VECTOR
* TEST MODE
 TST CONN
 BEQ TDCD3
* SETUP MSG TIMEOUT
 LDX #XIT
 STX VECT1
 LDX #(MSGT/2-1)
 BRA TDCD4
TDCD3 LDX #IDLE1
 STX VECT1
 LDX #*
 STX XIDLE
 LDX #(CDT/2-1)
TDCD4 JSR SET1 CARRIER DETECT TIME
 INC TDCDBY
* TEST DCD UP
TDCD2 LDA A COMS
 BIT A #$4
 BEQ TDCD1
 LDA A COMR
 CLC
```



```
 RTS
* INHIBIT TIMER
TDCD1 JSR CLR1
 CLR TDCDBY
 SEC
 RTS
*
* UNPACK ROUTINE (CJ=3)
*
C3 EQU *
* TEST ERROR FLAG
 TST RERR
 BNE C36
* UNPACK RECEIVED DATA
 LDX #RBUF
 LDA A 0,X SOH
 ADD A 6,X STX
* SOH + STX OK?
 CMP A #3
 BEQ C31
* NO
 LDA B #$28
 EOR B RERR
 STA B RERR
C36 JMP R0
* TEST STX+1 (DATA/NO-DATA)
C31 CLR B
 LDA A 7,X
 CMP A #3
 BEQ C32
 LDA B #4
C32 STA B RJ
* TEST OPER. CODE
 CLR B
 LDA A 4,X
 AND A #$38 MASK ACK/NAK
 BNE C33
 LDA B #2
C33 EORB RJ
 STA B RJ
* TEST SEQUENCE CODE
 LDA A 2,X
 CMPA RLSC
 BEQ C34
 LDA B #1
 EOR B RJ
 STA B RJ
C34 STA A RLSC
* FIX RJ TO INDEX
 LDA A RJ
 AND A #7
 STA A TMPX
* BRANCH IF DATA
 CMP A #4
 BLT C35
 JSR PDAT
* BRANCH TO PROCESS ACK/NAK
C35 LDA B TMPX
```

```
 LDX #RTBL
 JSR FIXX
 JMP O,X
*
* RECEIVE RESPONSE TABLE
* RJ=0 NO-DATA NAK OLD-SC
* RJ=1 NO-DATA NAK NEW-SC
* RJ=2 NO-DATA ACK OLD-SC
* RJ=3 NO-DATA ACK NEW-SC
* RJ=4 DATA NAK OLD-SC
* RJ=5 DATA NAK NEW-SC
* RJ=6 DATA ACK OLD-SC
* RJ=7 DATA ACK NEW-SC
*
* ACK/NAK TABLE
RTBL JMP RO RI=0 NAK
 JMP RO 1 NAK
 JMP R2 2 ACK
 JMP R2 3 ACK
 JMP RO 4 NAK
 JMP R5 5 PROCESS
 JMP R6 6 RESPOND ONLY
 JMP R5 7 PROCESS
*
* FIX POINTER BY INDEX
FIXX TST B
 BNE FIXX1
 RTS
FIXX1 INX
 INX
 INX
 DEC B
 BNE FIXX1
 RTS
*
* PROCESS DATA MESSAGE
*
PDAT LDX #RBUF
 LDA A 1,X
 CMP A #$44
 BNE PDAT1
* DATA = MESSAGE FOR CRT
 JSR XFRMSG
 RTS
* DATA TO BE PROCESSED
PDAT1 LDA A RJ
 CMP A #5
 BEQ PDAT2
 CMP A #7
 BEQ PDAT2
 RTS
* TRANSFER DATA FROM REC BUFFER
PDAT2 JSR TRAN
 RTS
*
* SETUP XMIT MESSAGE
*
SETXMT EQU *
```



```
 CLR CFFF
* SET WAIT COUNT
 LDA A #10
 STA A WAITC
* DECODE CODE
 JSR DCODE
 BCC SETX1
 STA B ASWD
* STATUS DETERMINED
* SETUP DATA MSG HEADER
 LDA B #NOC
 JSR SDMH
* SETUP STATUS BITS
 JSR RSTS
* TEST IF FIRE ORDER (0001)
 LDA A XDATA+3
 CMP A #$31
 BEQ SETX2
* RECOVER F.O. DATA
 LDX #XDATA+4
 STX DEST
 LDX #OLDFO
 LDA B #12
 JSR TCS
* TAG WITH TRAILER
 JSR SDMT
 RTS
*
* TRANSFER F. O. DATA
*
SETX2 LDX #XDATA+4
 JSR XFDCR
* TAG WITH TRAILER
 JSR SDMT
* MOVE F.O. DATA TO HOLD BUFFER
 LDX #OLDFO
 JSR XFDCR
 RTS
*
* SPECIAL MESSAGES
SETX1 JSR SSSM
 RTS
*
* CODE TABLE
*
TAB FCC /FO/ FIRE ORDER
 FCC /FC/ FIRE COMMAND
 FCC /RR/ READY RESPONSE
 FCC /CF/ CHECK FIRE
 FCC /DR/ DATA REQUEST
 FCC /EM/ END OF MISSION
 FCC /MG/ MESSAGE
*
* DECODE CODE
*
DCODE EQU *
 LDX #TAB
 STX TABP
```

```
 LDA B #1
*
DCODE2 LDX TABP
 LDX 0,X
 CPX CODE
 BEQ DCODE1
*
 LDX TABP
 INX
 INX
 STX TABP
*
 ASL B
 CMP B #$80
 BNE DCODE2
* TEST IF "STATUS" CODE
DCODE1 CMP B #8
 BLE DCODE3
 CLC
 RTS
*
DCODE3 SEC
 RTS
*
* TRANSFER FDC RECD DATA
* X=DATA DESTINATION ADDR.
*
XFDCR EQU *
 STX DEST
* DEFLECTION
 LDX #DATA+3
 LDA B #4
 JSR TCS
* ELEVATION
 LDX #DATA+12
 LDA B #4
 JSR TCS
* FUZE
 LDX #DATA+8
 LDA B #3
 JSR TCS
* CHARGE
 LDX #DATA+1
 LDA B #1
 JSR TCS
 RTS
*
* SETUP SPECIAL MESSAGES
*
SSSM EQU *
* DATA REQUEST?
 CMP B #$10
 BNE SSSM1
 LDA B #DOC
 JSR SIM
 RTS
*
* END OF MISSION?
```



```
*
SSSM1 CMP B #$20
 BNE SSSM2
 CLR ZRFD
 INC ZRFD
 RTS
*
* CRT MESSAGE (CONTROL RECORD)
*
SSSM2 CMP B #$40
 BNE SSSM5
* SETUP CONTROL RECORD HEADER
 LDA B #NOC
 JSR SCRH
* FIND END OF MESSAGE
 CLR B
 LDX #DATA
SSSM4 LDA A 0,X
 CMP A #EOT
 BEQ SSSM3
 INX
 INC B
 BRA SSSM4
* TRANSFER MSG TO XMIT BUFFER
SSSM3 LDX #XDATA
 STX DEST
 LDX #DATA
 JSR TCS
* SETUP MSG TRAILER
 LDA A #ETX
 STA A 0,X
 CLR 1,X
 RTS
SSSM5 INC INEF
 CLR FDCC
 RTS
*
* SETUP CONTROL RECORD HEADER
*
SCRH LDX #CRH
 JSR XFER
 RTS
*
CRH FCB SOH,CMFC,SC,AC,NOC,IC,STX,0
*
* TRANSFER DATA FROM REC BUFFER
* TO FDC "TO" BUFFER
*
TRAN EQU *
* STRIP STATUS
 JSR STS
* =FIRE ORDER ACK?
 CMP A #%00001110
 BNE TRAN1
* YES,TRANSFER TO FDC "TO" BUFFER
 JSR XFDCT
* TEST CHECK FIRE PENDING
 TST CFFF
```

```
BEQ TRAN3
RTS
 SETUP DATA MSG HEADER
RAN3 LDA B #NOC
JSR SOMH
 FIX STATUS
LDA A #4
STA A ASWD
JSR RSTS
 TRANSFER F.O. DATA
LDX #XDATA+4
JSR XFDCR
 SETUP DATA MSG TRAILER
JSR SOMT
 SET AUTO RR FLAG
CLR ATORR
INC ATORR
RTS

 * READY REQ,CHECK FIRE,FIRE COMMAND, DATA REQ ACK?
RAN1 EQU *
 FIRE COMMAND ACK?
CLR ATODR
CMP A #%00001101
BNE TRAN2
 YES, SETUP DATA REQUEST FLAG
INC ATODR
 TRANSFER TO FDC "TO" BUFFER
JSR XFDCT
 SETUP DR MESSAGE
LDA B #DOC
JSR SIM
RTS
 TRANSFER TO FDC "TO" BUFFER
RAN2 JSR XFDCT
CLR ATORR
CLR ATODR
RTS

 TRANSFER FROM RECV BUFFER TO FDC "TO" BUFFER

FDCT EQU *
 CLEAR FDC BUFFER
LDX #FDCTE
STX BEND
LDX #FDCT
JSR CLBF
 FIND ETX
CLR B
LDX #RDATA
FDCT1 LDA A O,X
CMP A #ETX
BEQ XFDCT2
INX
INC B
CMP B #48
BLS XFDCT1
 TRANSFER DATA
```



```
XFDCT2 LDX #FDCT
 STX DEST
 LDX #RDATA
 JSR TCS
* TAG WITH EOT
 LDA A #EOT
 STA A O,X
* SET DATA AVAIL FLAG
 CLR FDCA
 INC FDCA
 RTS
*
* TEST CONNECT SEQUENCE
*
CONT EQU *
* READY RESPONSE FLAG SET?
 INC ZRRF
 BEQ CONT1
 CLR ZRRF
 JMP XIT1
* FLAG SET, CONNECT
CONT1 CLR CONN
 INC CONN
* TURN ON "CONN" LED,OFF "STB
 LDA A #CLED1
 JSR LEDOFF
 LDA A #CLED2
 JSR LEDON
*
 CLR CJ
 JMP LOOP
*
* ACK/NAK PROCESSING, RJ= 0,1
*
R0 LDA A #CLED8
 JSR LEDOFF
 TST RERR
 BEQ R01
* SETUP NAK RESPONSE
 JSR SNR
 LDA A #CLED8
 JSR LEDON
* SWAP SEQ CODES
R01 JSR SSC
*
*
*
* RESET FOR TRANSMIT
XIT EQU *
* TEST CONNECT
 TST CONN
 BNE XIT1
 JMP CONT
XIT1 LDA A #1
 STA A CJ
 LDA A #$FF
 STA A XPASS
 CLR ATORR
```

I10

```
 CLR ATODR
* IN CASE WAIT RE-XMIT
 JSR CLRI
 CLR TDCDBY
* ENB XMIT/INH RECV
 JSR COMIR
 JMP LOOP
*
*
*
* RJ = 2,3
R2 LDA A #CLED8
 JSR LEDOFF
 LDX #RBUF
* STRIP FORMAT CODE
 LDA A 1,X
 STA A RXFC
* STRIP OPER CODE
 LDA A 4,X
 AND A #7
 STA A RXOC
** TEST SERVICE MSG
 LDA A RXFC
 CMP A #SMFC
 BEQ R24
* TEST RANGE OF FORMAT CODE
 BLS R2ER
 CMP A #$4D
 BHI R2ER
* TEST OPER CODE
* NO REQ OC?
 LDA A RXOC
 CMP A #0
 BNE R21
 CLR CJ
 JMP LOOP
* TEST IF READY RESPONSE
R21 CMP A #6
 BNE R22
* READY RESPONSE,SET CONNECT FLAG
 CLR ZRRF
 DEC ZRRF
 JMP XIT
* TEST IF WAIT
R22 CMP A #4
 BNE R2ER
* PROCESS WAIT
 JSR PWAIT
 BCS R23
 LDA A #2
 STA A CJ
* RESET RECEIVE
 LDX #RBUF
 STX BRI
* CLEAR BUFFER
 JSR CLRB
* INIT INT
 JSR COMIX
```



```
 JMP LOOP
* TOO MANY WAITS
* SET RFD
R23 CLR ZRFD
 INC ZRFD
 JMP XIT
*
* PROCESS SERVICE MESSAGES
* SELECT?
R24 LDA A RXOC
 CMP A #$2
 BNE R26
* SETUP INFO MSG
R28 LDA B #$40
 JSR SIM
 JMP XIT
* REQUEST FOR DISCONNECT?
R26 CMP A #$4
 BNE R27
* SETUP DISC SVS MSG
 LDA B #$46
 JSR SSM
 CLR DISF
 INC DISF
 CLR ZRFD
 JMP XIT
* NO INSTRUCTION?
R27 CMP A #$0
 BNE R2ER
*
 CLR CJ
 JMP LOOP
* ERROR PROCESSOR
R2ER LDA A #$28
 STA A RERR
 JMP R0
*
* RJ = 5,7
R5 EQU *
R6 EQU *
 LDA A #CLED8
 JSR LEDOFF
* AUTO RR FLAG SET?
 TST ATORR
 BNE R52
* AUTO DR FLAG SET?
 TST ATODR
 BEQ R51
* TEST CHECK FIRE PENDING
R52 TST CFFF
 BNE R51
 JMP XIT
* WAIT NEXT MESSAGE
R51 CLR CJ
 JMP LOOP
* SETUP SERVICE MESSAGE
SSM LDX #SMH
 JSR XFER
```

```
 RTS
SMH FCB SOH,SMFC,SC,AC,NOC,IC,STX,ETX,0
*
* TRANSFER DATA FROM STACK ARRAY
* TO X ARRAY
*
XFER STX SAVEX
 LDX #XBUF
 STX DEST
XFER2 LDX SAVEX
 LDA A 0,X
 INX
 STX SAVEX
*
 LDX DEST
 STA A 0,X
 INX
 STX DEST
 TST A
 BNE XFER2
* CLEAR REST OF BUFFER
 JSR CLXB
* SET OPER CODE
 LDX #XBUF
 STA B 4,X
 RTS
*
* PROCESS WAIT MESSAGE
*
PWAIT DEC WAITC
 BEQ PWTI
 CLC
 RTS
*
PWTI LDA A #10
 STA A WAITC
 SEC
 RTS
*
* SETUP INFO MESSAGE
*
SIM LDX #IMH
 JSR XFER
 RTS
*
IMH FCB SOH,IMFC,SC,AC,NOC,IC,STX,ETX,0
*
* SETUP DATA MESSAGE HEADER
*
SDMH LDX #DMH
 JSR XFER
 RTS
DMH FCB SOH,IMFC,SC,AC,NOC,IC,STX,0
*
* SETUP DATA MSG TRAILER
*
SDMT LDX #XDATA+16
 LDA A #ETX
```



```
 STA A 0,X
 CLR 1,X
 RTS
*
* SETUP SELECT MESSAGE
*
SSLM LDX #SSH
 JSR XFER
 RTS
*
SSH FCB SOH,SMFC,SC,AC,NOC,IC,STX
 FCB ETX,0
*
* STUP NAK RESPONSE
*
SNR LDA A RERR
* EXTRACT NAK BITS
 AND A #$38
 STA A RERR
* RECOVER HEADER
 LDX #XBUF
 LDA A 4,X
* REMOVE NAK BITS
 AND A #$47
* INSERT RERR MESSAGE
 EOR A RERR
 STA A 4,X
 RTS
*
* SWAP SEQUENCE CODES (41-42)
*
SSC LDX #XBUF
 LDA A OLDSC
 EOR A #3
 STA A 2,X
 STA A OLDSC
 RTS
*
* SETUP TIMER #1 (INTERRUPT)
*
SETI LDA A OLDCR2
 ORA A #1
 STA A OLDCR2
 STA A TCR2
* STORE TIME & START
 LDA A TSTS
 LDA A #TIIE TIMER 1
 STX TID
 STA A TCR13
 RTS
*
* CLEAR TIMER #1 (INTERRUPT)
*
CLRI LDA A OLDCR2
 ORA A #1
 STA A OLDCR2
 STA A TCR2
* DISABLE INTERRUPT
```

I12

```
 LDA A #T1II
 STA A TCR13
 RTS
*
* SETUP TIMER #2 (INTERRUPT)
*
SET2 LDA A #T2IE
 STA A OLDCR2
 LDA B TSTS
* STORE TIME & START
 STX T2D
 STA A TCR2
 RTS
*
* CLEAR TIMER #2 (INTERRUPT)
*
CLR2 LDA A #T2II
 STA A OLDCR2
 STA A TCR2
 RTS
*
* SETUP TIMER #3 (INTERRUPT)
SET3 LDA A OLDCR2
 AND A #%11111110
 STA A OLDCR2
 STA A TCR2
* SET TIME & START
 LDA A TSTS
 LDA A #T3IE
 STX T3D
 STA A TCR13
 RTS
*
* CLEAR TIMER #3 (INTERRUPT)
*
CLR3 LDA A OLDCR2
 AND A #%11111110
 STA A OLDCR2
 STA A TCR2
* DISABLE INTERRUPT
 LDA A #T3II
 STA A TCR13
 RTS
*
* INTERRUPT SERVICE ROUTINE
*
ISER EQU *
* TEST TIME
 STS ISAVES
 LDA A TSTS
 BPL ISER1
* TEST CLOCK (TIMER #3 = 100 MSEC)
 BIT A #4
 BEQ ISER2
 LDX T3D
* SCAN CLOCK TABLE
 LDX #TMTB
 LDA B #5
```



```
 JSR SCAT
 RTI
* TEST TIMERS
ISER2 BIT A #2
 BEQ ISER3
* FIX RETURN VIA VECTOR2
 JSR FRET2
 JSR CLR2
 RTI
ISER3 BIT A #1
 BNE ISER4
 RTI
* FIX RETURN VIA VECTOR #1
ISER4 JSR FRET1
 JSR CLR1
 RTI
*
* TEST COMM INTERRUPT
ISER1 TST COMS
 BPL ISER5
* COMM POLL
 JSR CPOLL
 RTI
*
* TEST FDC INTERRUPT
*
ISER5 TST PIA4CA
 BPL ISER15
 JSR FDIN
 RTI
*
* TEST CRT INT.
ISER15 TST TPS
 BMI ISER10
* RESET INT.
 LDA A TPR
 RTI
* TEST RECEIVE
ISER10 LDA A TPS
 BIT A #1
 BEQ ISER11
* SERVICE RECEIVE
 JSR TPREC
 RTI
* TEST TRANSMIT
ISER11 BIT A #2
 BNE ISER12
* RESET INT.
 LDA A TPR
 RTI
* SERVICE TRANSMIT
ISER12 JSR TPXMT
 RTI
*
*
* SCAN TIMER TABLE
*
SCAT TST 0,X
```

```
 BEQ RT
 LDA A 2,X
 SUB A 0,X
 STA A 2,X
 BNE RT
 TST 1,X
 BEQ ST3
 DEC 1,X
 BRA RT
ST3 CLR 0,X
RT INX
 INX
 INX
 DEC B
 BNE SCAT
 RTS
*
* CLEAR TIMERS
*
CLTM LDA B #15
 LDX #TMTB
CLT1 CLR 0,X
 INX
 DEC B
 BNE CLT1
 RTS
*
* FIX RETURN VECTOR #2
*
FRET2 LDX ISAVES
 LDA A VECT2
 LDA B VECT2+1
 STA A 6,X
 STA B 7,X
 RTS
*
* FIX RETURN VECTOR #1
*
FRET1 LDX ISAVES
 LDA A VECT1
 LDA B VECT1+1
 STA A 6,X
 STA B 7,X
 RTS
*
*
* INPUT FROM ACIA
*
AOI LDA A 0,X
 BIT A #1
 BEQ AOI1
 LDA A 1,X
 SEC
 RTS
AOI1 CLC
 RTS
*
* TEST PORT TRANSMIT
```



```
TPXMT TST ASP
 BEQ TPXM1
 DEC ASP
TPXM4 LDX TPPTR
 CPX TPPTE
 BEQ TPXM2
 LDA A 0,X
 INX
 STX TPPTR
 TST SPC
 BEQ TPXM3
 ADD A #$30
TPXM3 LDX #TPC
 JSR AOO
 RTS
* FIX SPACE
TPXM1 LDA A SPC
 BEQ TPXM4
 STA A ASP
 LDA A #$20
 BRA TPXM3
* LAST CHAR-INH XMT /ENB RECV
TPXM2 LDA A #RIE
 STA A TPC
 LDA A TPR
 RTS
*
* TEST PORT RECEIVE
*
TPREC LDX #TPC
 JSR CRLF
 JSR AOI
 BCC TPRE1
 JSR AOOL
* =* TEST DATA INPUT
TPRE1 TST INF
 BEQ TPRE14
 JMP TPRE9
*  =R? (RECV BUFFER)
TPRE14 CMP A #'R
 BNE TPRE2
 LDX #RBUF
 STX TPPTR
 LDX #REND
 STX TPPTE
 CLR A
 JMP TPRE3
* = X? (XMIT BUFFER)
TPRE2 CMP A #'X
 BNE TPRE4
 LDX #XBUF
 STX TPPTR
 LDX #XEND
 STX TPPTE
 CLR A
 JMP TPRE3
* =D? (FDC BUFFER)
TPRE4 CMP A #'D
```

I14

```
 BNE TPRE5
 LDX #FDCT
 STX TPPTR
 LDX #FDCTE
 STX TPPTE
 CLR FDBSY
 CLR FDCA
 LDA A #CLED5
 JSR LEDOFF
 CLR A
 JMP TPRE3
* = F? (FLAG BUFFER)
TPRE5 CMP A #'F
 BNE TPRE6
 LDX #FLAG
 STX TPPTR
 LDX #FEND
 STX TPPTE
 LDA A #1
 JMP TPRE3
* TEST DATA INPUT MODE
TPRE6 CMP A #'$
 BNE TPRE7
* SET INPUT FLAG
 CLR INF
 INC INF
 LDA A #CLED6
 JSR LEDON
* SET POINTERS
 LDX #FDCF
 STX TPPTR
* CLEAR BUFFER
TPRE10 CLR 0,X
 INX
 CPX #FDCFE
 BNE TPRE10
 CLR CFFF
 LDX #FDCFE
 STX TPPTE
* START INPUT
TPRE9 LDX TPPTR
 STA A 0,X
* TEST CANCEL (XC)
 CMP A #$18
 BNE TPRE12
* RESTART
 CLR INF
 LDA A #CLED6
 JSR LEDOFF
 CLR FDCC
 RTS
TPRE12 INX
 STX TPPTR
 CPX TPPTE
 BNE TPRE11
 DEX
 LDA A #EOT
 STA A 0,X
```



```
TPRE11 CMP A #EOT
 BEQ TPRE8
 RTS
* LAST CHAR
TPRE8 CLR INF
* TEST IF CHECK FIRE
 LDA A CODE
 CMP A #'C
 BNE TPRE13
 LDA A CODE+1
 CMP A #'F
 BNE TPRE13
 INC CFFF
TPRE13 LDA A #CLED6
 JSR LEDOFF
 CLR FDCC
 INC FDCC
 RTS
* SETUP SPACE COUNT
TPRE3 STA A SPC
 STA A ASP
* OUTPUT CR/LF
 LDX #TPC
 JSR CRLF
* INH REC/ENB XMIT
 LDA A #XIE
 STA A TPC
TPRE7 LDA A TPR
 RTS
*
* TRANSFER FDC "TO" BUFFER TO FDC
*
*
* OUTPUT TO ACIA
*
AOO PSH A
 LDA A 0,X
 BIT A #2
 PUL A
 BEQ AOO1
 STA A 1,X
 SEC
 RTS
AOO1 CLC
 RTS
*
* CR/LF ROUTINE
*
CRLF LDA A #$D
 JSR AOOL
 LDA A #$A
 JSR AOOL
 RTS
*
* LOOP ON OUTPUT
*
AOOL JSR AOO
 BCC AOOL
```

```
 RTS
 *
 * TRANSFER RECD MESSAGE TO CRT
 *
XFRMSG LDX #RDATA
 STX TPPTR
* FIND END OF MESSAGE
XRMS1 LDA A O,X
 CMP A #ETX
 BEQ XRMS2
 INX
 BRA XRMS1
* FIX END ADDRESS
XRMS2 DEX
 STX TPPTE
 CLR SPC
 CLR ASP
* OUTPUT CR/LF
 LDX #TPC
 JSR CRLF
* INH REC/ENB XMIT
 LDA A #XIE
 STA A TPC
 RTS
*
* STRIP STATUS
*
STS LDX #RDATA
 LDA B #4
 CLR RSWD
STS2 LDA A O,X
 AND A #1
 EOR A RSWD
 STA A RSWD
 DEC B
 BNE STS1
 RTS
STS1 INX
 ASL RSWD
 BRA STS2
*
* RESET STATUS
*
RSTS EQU *
* XFER TO XMIT BUFFER
 LDX #XDATA+3
 LDA B #4
RSTS2 CLR A
 LSR ASWD
 BCC RSTS1
 INC A
RSTS1 ADD A #$30
 STA A O,X
 DEX
 DEC B
 BNE RSTS2
 RTS
 *
```



```
* TRANSFER CHAR STRING
* X=SOURCE,DEST=DESTINATION,B=CHAR CNT
*
TCS STX SAVEX
TCS1 LDX SAVEX
 LDA A O,X
 INX
 STX SAVEX
*
 LDX DEST
 STA A O,X
 INX
 STX DEST
 DEC B
 BNE TCS1
 RTS
*
* SETUP PIAS
*
* PIA1-NOT USED
PIAS EQU *
* PIA2-DISPLAYS & CONTROLS
 LDX #PIA2DA
 LDA A #$36
 JSR SETUP
*
* PIA3 - NOT USED
*
*
* PIA4 - FDC PDP11/34
* SETUP B SIDE FOR OUTPUT
* SETUP A SIDE FOR INPUT
*
 LDX #PIA4DA
 LDA A #%00111100
 JSR SETUP
* ENABLE CA1
 LDA A #%00111101
 STA A PIA4CA
*
* TIMER
 LDA A #T3II
 STA A TCR13
 LDA A #T2II
 STA A TCR2
 STA A OLDCR2
 LDA A #T1II
 STA A TCR13
* SET TIMER #3 PERIOD (100 MSEC)
 LDX #12500
 JSR SET3
*
*
* COMM ACIA
 LDA A #$43
 STA A COMC
 STA A $C102
 LDA A #CNIE
```

I16

```
 STA A COMC
*
*
* TEST PORT ACIA
 LDA A #3
 STA A TPC
 LDA A #RIE
 STA A TPC
*
* DISABLE INTERRUPTS
 SEI
 NOP
 RTS
*
* SETUP PIAS
*
SETUP CLR 2,X CA DD SELECT
 CLR 3,X CB DD SELECT
 LDA B #$FF B SIDE-OUTPUT
 STA B 1,X B SIDE OUTPUT
 CLR 0,X A SIDE=INPUT
 STA A 2,X CA OUPUT & CONTROL SELECT
 STA A 3,X CB OUTPUT & CONTROL SELECT
 CLR 1,X ZERO OUTPUT
 LDA A 0,X RESET
 LDA A 1,X RESET
 RTS
*
* COMM LED ON
*
LEDON ORA A OT2
 STA A OT2
 RTS
*
* COMM LED OFF
*
LEDOFF COM A
 AND A OT2
 STA A OT2
 RTS
*
* CLEAR REC BUFFER
*
* X=START ADDR
*
CLRB CLR 0,X
 INX
 CPX #REND
 BNE CLRB
 RTS
*
* CLEAR XMIT BUFFER
*
* X=START ADDR
*
CLXB CLR 0,X
 INX
 CPX #XEND
```



```
 BNE CLXB
 RTS
*
* INPUT FROM FDC
*
FDIN EQU *
 CLR CFFF
* TEST PASS FLAG
 LDA A #CLED6
 JSR LEDON
 TST FIBSY
 BNE FDI3
* INITIAL POINTER
 LDX #FDCF
 STX FPTR
 INC FIBSY
* GET AND STORE CHAR
FDI3 LDX FPTR
 LDA A PIA4DA
 AND A #$7F
* CONVERT SPACES TO 0
 CMP A #$20
 BNE FDI5
 LDA A #$30
FDI5 STA A 0,X
 CLR INEF
 CMP A #EOT
 BEQ FDI2
* STROBE DATA ACCEPT
 JSR STRB4A
 INX
 STX FPTR
 CPX #FDCFE
 BEQ FDI4
 RTS
* WRAP UP
FDI4 INC INEF
* OUTPUT FINAL STROBE
FDI2 JSR STRB4A
 CLR FIBSY
 INC FDCC
* TEST IF CHECK FIRE
 LDA A CODE
 CMP A #'C
 BNE FDI6
 LDA A CODE+1
 CMP A #'F
 BNE FDI6
 INC CFFF
FDI6 LDA A #CLED6
 JSR LEDOFF
* DISABLE INT. (CA1)
 TST INEF
 BEQ FDI1
 RTS
FDI1 LDA A #%00111100
 STA A PIA4CA
 RTS
```

```
*
* STROBE OUTPUT-PIA4A
*
STRB4A LDA A #%00110101
 STA A PIA4CA
 NOP
 LDA A #%00111101
 STA A PIA4CA
 RTS
*
* READ OUTPUT CHAR AND GENERATE PARITY
*
FREAD LDA A 0,X
 JSR PAR
 BCS FRD1
* RECOVER DATA,EVEN
 LDA A 0,X
 AND A #$7F
 RTS
* TAG TO MAKE EVEN
FRD1 LDA A 0,X
 ORA A #$80
 RTS
*
* TEST EVEN PARITY
* C SET=ODD, C CLR=EVEN
*
PAR CLR B
PAR2 LSR A
 BCC PAR1
 INC B
PAR1 TST A
 BNE PAR2
* CHECK B EVEN OR ODD
 LSR B
 RTS
*
* OUTPUT TO FDC
*
FDOUT EQU *
 LDA A #CLED5
 JSR LEDON
* OUTPUT BUSY ?
 TST FOBSY
 BNE FD01
* INITIAL BUFFER PTR
 LDX #FDCT
 STX FPTT
* CLEAR DATA ACCEPT (CB1)
 INC FOBSY
FD01 LDA A PIA4DB
* SETUP WAIT TIME (=.5 SEC)
 LDX #5
 STX TIM1
* FETCH CHAR AND OUTPUT
 LDX FPTT
 JSR FREAD
 STA A PIA4DB
```



```
* STROBE DATA READY
 JSR STRB4B
* WAIT FOR ACCEPPT
 CLR TF1
 INC TF1
FD03 TST PIA4CB
 BMI FD02
 TST TF1
 BEQ FD04
 BRA FD03
* DATA ACCEPTED,RECOVER CHAR
FD02 LDA A 0,X
* INC POINTER
 INX
 STX FPTT
* TEST LAST CHAR
 CMP A #EOT
 BEQ FD04
 RTS
* LAST CHAR CLEAR FLAGS
FD04 CLR FOBSY
 CLR FDCA
 LDA A #CLED5
 JSR LEDOFF
 RTS
*
* STROBE OUTPUT-PIA4B
*
STRB4B LDA A #%00110100
 STA A PIA4CB
 NOP
 LDA A #%00111100
 STA A PIA4CB
 RTS
* SYSTEM VECTORS
 ORG $FFF8
 FDB ISER
 FDB ISER
 FDB ISER
 FDB STRT
 END
```

I 18

APPENDIX J

AGLS CONTROL PROGRAM
SOURCE LISTING

```
 NAM AGLS
* REVISED 7/15/79 1000

* SERIAL COMM GUN ORDERS
*
**********AGLS1***********
* PIA EQUATES
*
* PIA0=AUTO SWITCHES(A),CLOCK RATE(B)
*
PIA0DA EQU $2800
PIA0DB EQU PIA0DA+1
PIA0CA EQU PIA0DA+2
PIA0CB EQU PIA0DA+3
*
*
* MANUAL G.D. INPUT
* A=DATA, B=ADDR.
PIA1DA EQU $2400
PIA1DB EQU PIA1DA+1
PIA1CA EQU PIA1DA+2
PIA1CB EQU PIA1DA+3
*
* PIA2=QUAD EL ENCODER:MSB=A,LSB=B
*
PIA2DA EQU $2404
PIA2DB EQU PIA2DA+1
PIA2CA EQU PIA2DA+2
PIA2CB EQU PIA2DA+3
*
* PIA3=PANTEL AZ ENCODER:MSB=A,LSB=B
*
PIA3DA EQU $2408
PIA3DB EQU PIA3DA+1
PIA3CA EQU PIA3DA+2
PIA3CB EQU PIA3DA+3
*
* PIA4=1/10 ENCODER OUTPUTS(A),ENABLE OUTPUTS(B)
*
PIA4DA EQU $240C
PIA4DB EQU PIA4DA+1
PIA4CA EQU PIA4DA+2
PIA4CB EQU PIA4DA+3
*
* PIA5=MUX A/D DATA (A),MUX ADDR(B)
*
PIA5DA EQU $2410
PIA5DB EQU PIA5DA+1
PIA5CA EQU PIA5DA+2
PIA5CB EQU PIA5DA+3
*
* PIA6=EL TRIM A/D(A),AZ TRIM A/D(B)
*
PIA6DA EQU $2414
PIA6DB EQU PIA6DA+1
PIA6CA EQU PIA6DA+2
PIA6CB EQU PIA6DA+3
*
```

J1

```
PAGE 002  AGLS1   .SA:1

* PIA7=D/A CONVERTER
*
PIA7DA EQU $2418
PIA7DB EQU PIA7DA+1
PIA7CA EQU PIA7DA+2
PIA7CB EQU PIA7DA+3
*
* PIA8=DISPLAY(4) AND SWITCHES(4),(A)
*       MISC INPUTS(3) AND DISP ADDR(5),(B)
*
PIA8DA EQU $241C
PIA8DB EQU PIA8DA+1
PIA8CA EQU PIA8DA+2
PIA8CB EQU PIA8DA+3
*
*
* PIA 0 SWITCH MASKS
*
LDP1M EQU %10000000
LDP2M EQU %1000000
PLROM EQU %100000
QLROM EQU %10000
POROM EQU %1000
QOROM EQU %100
AZROM EQU %10
ELROM EQU %1
*
* PIA 0 STROBE
RWDOG EQU PIA0CA
*
*
* PIA4 (A) INPUT MASKS
*
ELTM EQU %1111
AZTM EQU %11110000
*
* PIA4 (B) ENABLE BITS
* 0=TRUE
*
PLGO EQU %11111110
QCGO EQU %11111101
PAGO EQU %11111011
QPGO EQU %11110111
POGO EQU %11101111
QOGO EQU %11011111
AZGO EQU %10111111
ELGO EQU %01111111
*
* ERROR VOLTAGE MASKS
QPMA EQU 1
QCMA EQU 2
MPMA EQU 3
MCMA EQU 4
PAMA EQU 5
*
* PIA5 ENABLES AND FLAG ADDRESSES
*
SAD5 EQU PIA5CA
```

J2

```
PAGE 003  AGLS1   .SA:1

EMUX EQU PIA5CB
CCM5 EQU PIA5CA
*
* PIA6 ENABLES AND FLAG ADDRESSES
*
SAD6A EQU PIA6CA
SAD6B EQU PIA6CB
CCM6A EQU PIA6CA
CCM6B EQU PIA6CB
*
* PIA7 ENABLE ADDRESSES
*
DAEL EQU PIA7CA
DAAZ EQU PIA7CB
*
* PIA8 (A) MASKS
*
DOM EQU %1111
RUCWM EQU %10000
RUCCWM EQU %100000
LPOSM EQU %1000000
WPNM EQU %10000000
*
* PIA8(B) MASKS
*
DAOM EQU %11111
XRECM EQU %1000000
SRVOM EQU %10000000
* PIA8 ENABLE ADDRESSES
*
DSB EQU PIA8CA
XCF EQU PIA8CB
LAMP EQU PIA8CB
*
* ACIA EQUATES
ACIC EQU $3000
ACIS EQU ACIC
ACIT EQU $3001
ACIR EQU ACIT
AC2C EQU $3002
AC2S EQU AC2C
AC2T EQU $3003
AC2R EQU AC2T
*
* 300 BAUD
* DIVIDE BY 16 ACIA
*
AIE EQU %00101001
RIE EQU %10001001
NIE EQU %00001001
*
* MISC EQUATES
*
LOAD1 EQU $0200
LOAD2 EQU $0300
LOAD3 EQU $0400
LOAD4 EQU $0500
*
```

```
PAGE 004  AGLS1   .SA:1

P100M EQU $0A00
M100M EQU $8A00
HAFBAK EQU $85FF
FULBAK EQU $8FFF
*
* OFFSET ERROR TABLE
*
EQP EQU 10
EQC EQU 5
EMP EQU 5
EMC EQU 5
EPA EQU 2
AZLIM EQU 1
ELLIM EQU 1
OPLIM EQU 10
 ORG 0
 RMB 1
* EXEC RAM
MSBY RMB 1 DEC-BIN ROUTINE
LSBY RMB 1
TMP RMB 1
*
* INTERRUPT DRIVEN TIMERS
*
* TIMER TABLE (DECREMENT)
* 1=INSIDE LOOP (ISER)
* 2=OUTSIDE LOOP (ISER)
* 3=AZ CLOSING (PANTEL)
* 4=EL WPN DELAY (1 SEC)
* 5=XENON DROPOUT
* 6=AZ UPDATE DELAY
TMTB EQU *
TF1 RMB 1
TIM1 RMB 2
*
TF2 RMB 1
TIM2 RMB 2
*
TF3 RMB 1
TIM3 RMB 2
*
TF4 RMB 1
TIM4 RMB 2
*
TF5 RMB 1
TIM5 RMB 2
*
TF6 RMB 1
TIM6 RMB 2
*
* FLAG BUFFER
*
*
BEG EQU *
AZDAF RMB 1 AZ D/A INHIBIT
ELDAF RMB 1 EL D/A INHIBIT
AZGOF RMB 1 AUTO AZ SELECT
ELGOF RMB 1 AUTO EL SELECT
```

J3

PAGE 005 AGLSI .SA:1

```
XRECF RMB 1 XENON RECOGNITION
LPOSF RMB 1 LOAD POSITION SELECT
LOADF RMB 1 LOAD POSITION ENABLED
XTHRU RMB 1 ALTERNATING COMPUTATION PASS
NEWFO RMB 1 NEW FIRE ORDER
WPNS RMB 1 WEAPON SW. SELECTED
CBSY RMB 1 COMM. BUSY
AREDY RMB 1 LOOP 2 READY
SIG RMB 1 COMPUTATION SIGN
CKOF RMB 1 STATUS UPDATE DISPLAY LOCKOUT
NEGF RMB 1 A/D DIRECTION FLAG
DTHRU RMB 1 ALTERNATING DISPLAY PASS
DISEL RMB 1 EL DISPLAY LOCKOUT
DISAZ RMB 1 AZ DISPLAY LOCKOUT
BDF RMB 1 BASE DEFLECTION FLAG
COMERR RMB 1 COMM. ERROR
WPNF RMB 1 WEAPON SWITCH FLAG
MANIN RMB 1 MANUAL INPUT FLAG
ENDF EQU *+1
*
* VARIABLE STORAGE
*
LOADX RMB 2 LOAD VALUE
CONTEM RMB 1 MODE SW WORD
LITE RMB 1 DISPLAY LIGHT WORD
CONGO RMB 1 MODE SELECT WORD
AZTRM RMB 1 AZ TRIM VALUE
ELTRM RMB 1 EL TRIM VALUE
* ERROR VOLTAGE BUFFER
ERRBUF EQU *
 RMB 7
MUXADD RMB 1 MULTIPLEXER ADDRESS
NUMRED RMB 1 CURRENT VALUE
PREVAL RMB 1 PREVIOUS VALUE
EFLAG RMB 1 GO/NO-GO FLAG
* INTERRUPT SERVICE ROUTINE FLAGS
DISADR RMB 1 DISPLAY ADDRESS(MUX)
ACT RMB 2 DISPLAY ACTIVE MSG ADDR.
PTR RMB 2 CRT ACTIVE MSG ADDR.
PTE RMB 2 CRT MSG END ADDR.
PTRC RMB 2 COMM ACTIVE MSG ADDR
PTRE RMB 2 COMM MSG END ADDR
SPC RMB 1 ACTIVE SPACE CNT
ASP RMB 1 FIXED SPACE CNT
XON RMB 1 XENON CONSECUTIVE ON COUNTER
*
AZCOM RMB 2 AZ D/A OUTPUT
ELCOM RMB 2 EL D/A OUTPUT
*
AZERR RMB 2 AZ BINARY ERROR
ELERR RMB 2 EL BINARY ERROR
*
AZCNT RMB 1 AZ ERROR IN-BOUNDS COUNTER
ELCNT RMB 1 EL ERROR IN-BOUNDS COUNTER
*
A1 RMB 2 BCD ARITH ARG. BUFFERS
A2 RMB 2
S1 RMB 2
```

PAGE 006 AGLSI .SA:1

```
S2 RMB 2
X1 RMB 2
X2 RMB 2
*
OUTX RMB 2 BCD ARITH ADDR BUFFER
HOLDX RMB 2
TIZX RMB 2 BIN-BCD VALUE BUFFER
STORX RMB 2 BCD ARITH ADDR BUFFER
ADDX1 RMB 2 BCD ARITH ARG. BUFFERS
SUBX1 RMB 2
SUBX2 RMB 2
TX RMB 2 BCD SUBT. ADDR. BUFFER
*
CNX RMB 1 BIN-BCD CONVERSION COUNTERS
OLDAX RMB 1
OLDBX RMB 1
*
HOLDB RMB 1 TEMPORARY B STORE--TERR
KEEP RMB 2 NINES COMP ADDR BUFFER
*
SAVA RMB 1 WORK SPACE
SAVB RMB 1
SAVEX RMB 2
SAVES RMB 2
ISAVES RMB 2
ISAVEX RMB 2
********
********
* DISPLAYS BUFFER
* ELEVATION
* FIRE ORDER
ELFO RMB 5 CURRENT COMMAND
* ENCODER
ELDISP RMB 5 ACTUAL
* ERROR
ELERD RMB 5 ERROR
*
*
* AZIMUTH
* FIRE ORDER
AZFO RMB 5 CURRENT COMMAND
* ENCODER
AZDISP RMB 5 ACTUAL
* ERROR
AZERD RMB 5 ERROR
*
* ACTIVE COMMAND VALUES
ELGCDS RMB 5
AZGCDS RMB 5
* COMM SYSTEM MODES
*
*   0=NORMAL
*   1=BASE DEFLECTION
*   2=BORESIGHT
*   3=BASE DEFL. SET
*   4=CLEAR BASE DEFL. SET
*
CMODE RMB 1
```

J4

```
*
* OFFSET ERROR WORDS
* ELLIT: QP=1
*        QC=2
*        QP+QC=3
* AZLIT: MP=1
*        MC=2
*        MP+MC=3
*        AP=4
ELLIT RMB 1
AZLIT RMB 1
* MODE SWITCH WORD
* AGLS MODE SELECT
*  AUTO EL =1
*  AUTO AZ =2
*  QUAD OFFSET =3
*  PAN OFFSET =4
AMODE RMB 1
* LOCAL MODE WORD
*  =0 NO B.D. PRESET
*  =1 B.D. PRESET
*  =2 AUTO UPDATE F.O.
LMODE RMB 1
********
********
* TEMPORARY BUFFERS
TEMSUB RMB 5
TEMBCD RMB 5
RESULT RMB 5
ADJX RMB 5
ELTEMP RMB 5
AZTEMP RMB 5
TRMENC RMB 5
* REF ANGLE HOLD BUFFER
RHOLD RMB 5
* REF ANGLE ACTIVE BUFFER
REF RMB 5
* BASE DEFLECTION BUFFER
BDBUF RMB 5
* COMM BUFFER
CBUF RMB 16
* RESTART FLAGS HERE
REST EQU *
*
EBLK RMB 1 LOOP 2 BLOCK
TBLK RMB 1
XTIME RMB 1 XENON STABILITY
STF RMB 1 FIRST PASS FLAG
ELLK1 RMB 1 LOOP 2 LOCK
AZLK1 RMB 1
ELLK2 RMB 1 LOOP 3 LOCK
AZLK2 RMB 1
DEL4 RMB 1 WPN EL START DELAY
DEL6 RMB 1 AZ UPDATE DELAY
PASSAZ RMB 1 AZ CLOSING PASS
PASSEL RMB 1 EL CLOSING PASS
SLOWF RMB 1 AZ SLOW SPEED
RUCWF RMB 1 CW SEARCH
```



```
RUCCWF RMB 1 CCW SEARCH
OPLP RMB 1 QUAD PITCH ERROR CNT
OPCNT RMB 1 QUAD PITCH ERROR CNT
ATOUP RMB 1 AUTO UPDATE
WPPASS RMB 1 WPN SW PASS
END EQU *
 PAGE
*
* AGLS CONTROL PROGRAM
*
* ENTER HERE ON RESET
 ORG $4800
AGO EQU *
 LDS #$OFFO
* INITIAL PIAS
 JSR PIAS
* CLEAR TIMERS
 JSR CLTM
* CLEAR FLAGS
 LDX #BEG
 JSR CLFG
* SET DISPLAY INTERVALS
 LDX #2
 STX TIM1
 LDX #20
 STX TIM2
 INC TF1
 INC TF2
* SETUP AZ FOR 3200
 LDX #AZGCDS
 STX ISAVES
 LDX #CONST3
 LDA B #5
 JSR TCS
*
 LDX #AZFO
 STX ISAVES
 LDX #CONST3
 LDA B #5
 JSR TCS
* TEST MANUAL F.O. INPUT
 CLR MANIN
 LDA A PIA1DA
 BIT A #$10
 BNE AG1
* DISABLE ACIA INT.
 LDA A #NIE
 STA A ACIC
 LDA A ACIR
 INC MANIN
 CLI
 BRA AG2
* PULL COMM PROC RTS HI
AG1 CLI
 LDA A #$89
 STA A ACIC
 LDA A ACIR
* CLEAR RESTART FLAGS
```

J5

```
AG2 LDX #REST
 JSR CLFG
* CLEAR WEAPON FLAGS
 CLR WPNS
 CLR WPNF
* DISABLE AUTO UPDATE ON RESTART
 LDA A LMODE
 AND A #$FD
 STA A LMODE
 CLR ATOUP
* READ CONFIG SWITCHES
 JSR CONRO
* FORM CONFIG WORD
 JSR OCE
* READ AZ AND EL TRIM
AG7 JSR RTRM
* SHUT OFF DISPLAYS
 CLR DISAZ
 CLR DISEL
* IF MANUAL READ IT
 TST MANIN
 BEQ AG31
 JSR RMAN
* TEST IF COME FROM LOAD POS.
AG31 JSR TLOAD
 TST LOADF
 BNE AG33
* TEST FIRE ORDER UPDATE
 TST NEWFO
 BEQ AG8
* TEST AUTO UPDATE
 TST ATOUP
 BEQ AG33
* AUTO UPDATE SEQ.
* TEST RANGE OF UPDATE
 JSR TRANG
 BCC AG34
* WITHIN RANGE
 SEI
 JSR MVCD
 CLI
 JSR UFO
* RESTART TIMER
 CLR TF6
 LDX #300
 STX TIM6
 INC TF6
AG34 CLR NEWFO
 BRA AG8
* TEST AUTO MODE
AG33 JSR AUTO
 BCC AG8
* TEST WEAPON SWITCH
 LDA A WPNS
 CMP A #1
 BNE AG8
* RESTART SYSTEM
AG5 LDX #REST
```



```
 JSR CLFG
* UPDATE F.O.
 JSR UFO
 CLR NEWFO
 CLR WPNS
 INC WPPASS
* LOAD POS. CHANGE?
 TST LOADF
 BEQ AG8
 INC STF
 CLR LOADF
* TEST COMM MODE
* NORMAL ?
AG8 CLR BDF
 LDA A CMODE
 BEQ AG18
* BORESIGHT ?
 CMP A #2
 BEQ AG18
* CLEAR B.D.?
 CMP A #4
 BLT AG17
* FIX CLEAR
 LDA A LMODE
 AND A #$FE
 STA A LMODE
 CLR BDF
 JSR CLRBD
 BRA AG18
* FIX BASE DEF.
AG17 INC BDF
* TEST BD SET
 CMP A #3
 BNE AG18
 SUB A #2
 STA A CMODE
 LDA A LMODE
 ORA A #1
 STA A LMODE
* READ AZ ENCODER AND STORE
 JSR FIXBD
*
* READ AZ AND EL ENCODERS
AG18 INC DISEL
 INC DISAZ
 JSR RENCS
* TEST XTHRU FLAG
* =1 EL COMPUTE,=0 AZ COMPUTE
*
 TST XTHRU
 BNE AG10
 JSR COMPAZ
 INC XTHRU
 BRA AG11
*
AG10 JSR COMPEL
 CLR XTHRU
* READ AND STORE ERROR VOLTAGES
```

J6

```
PAGE 011  AGLSI    .SA:1

AG11 JSR RAEV
* CHECK OFFSETS
 JSR CKO
* TEST SERVO SW
 LDA A #SRVOM
 JSR TSTS8B
 BCS AG39
 LDA A #$FF
 STA A PIA4DB
 JMP AG2
*
* LOOP 2 START
*
*
* SET RU FLAGS
AG39 JSR SRUF
* WEAPON AZ ENABLED?
 CLR AZDAF
 TST AZGOF
 BEQ AG12
* AUTO AZ SELECTED
* TEST START THRU FLAG
AG44 TST STF
 BEQ AG37
 JMP AG16
* FIRST TIME THRU
* TEST XENON ON
AG37 TST XRECF
 BNE AG32
 JMP AG4
* ENABLE PANTEL AZ
AG32 LDAA #PAGO
 JSR FIXENB
* TEST READY
 TST AREDY
 BEQ AG41
 JMP AG3
* AZ D/A OUTPUT=0
AG41 LDX #0
 STX AZCOM
* DISABLE PANTEL OFFSET
 LDA A #POGO
 JSR FIXDIS
* DISABLE RU SEARCH FLAGS
 CLR RUCWF
 CLR RUCCWF
* SET READY
 CLR AREDY
 INC AREDY
 BRA AG3
* WEAPON AZ DISABLED
* TEST PANTEL OFFSET
AG12 LDA A #POROM
 JSR TSTSW
 BCC AG13
* ENABLE PANTEL OFFSET
 LDA A #POGO
 JSR FIXENB
```

```
PAGE 012  AGLSI    .SA:1

* TEST AZ LOCK
 TST AZLKI
 BNE AG16
* ENABLE PANTEL AZ
 LDA A #PAGO
 JSR FIXENB
* TEST AZ CLOSING
 JSR CLAZ
 BCC AG16
* DISABLE PANTEL AZ
 LDA A #PAGO
 JSR FIXDIS
 INC AZLKI
 BRA AG16
* NO AUTO WEAPON OR AUTO OFFSET
* AZ D/A OUTPUT=0
AG13 LDX #0
 STX AZCOM
 CLR AZLKI
 BRA AG16
* NO LITE.TEST SEARCH
AG4 CLR AREDY
* RU SEARCH CW?
 LDAA RUCWF
 BEQ AG14
 LDX #P100M
 STX AZCOM
 BRA AG15
* RU SEARCH CCW?
AG14 LDA A RUCCWF
 BEQ AG20
 LDX #M100M
 STX AZCOM
* ENABLE PANTEL OFFSET
AG15 LDA A #POGO
 JSR FIXENB
* ENABLE PANTEL AZ
 LDA A #PAGO
 JSR FIXENB
* SET TRAV BLOCK
 CLR TBLK
 INC TBLK
 BRA AG16
* DISABLE PANTEL AZIMUTH
AG20 LDA A #PAGO
 JSR FIXDIS
 CLR TBLK
 INC TBLK
 BRA AG16
* TEST XENON STABILITY
AG3 CLR TBLK
 JSR XSTAB
 BCS AG16
* SET TRAV BLOCK
 INC TBLK
AG16 CLR ELDAF
* AUTO EL SELECTED?
 TST ELGOF
```

```
 BEQ AG23
* START THRU SET?
 TST STF
 BNE AG21
* TEST QUAD PITCH CLOSING
 CLR EBLK
 INC EBLK
 LDA A #10
 STA A QPLP
 JSR CLQP
 BCC AG21
* NULL ACHIEVED
 CLR EBLK
 BRA AG21
* TEST QUAD OFFSET SELECT
AG23 LDAA #QORQM
 JSR TSTSW
 BCS AG50
 LDX #0
 STX ELCOM
 CLR ELLK1
 BRA AG21
* ENABLE QUAD OFFSET
AG50 LDA A #QOGO
 JSR FIXENB
* TEST EL LOCK
 TST ELLK1
 BNE AG21
* ENABLE QUAD PITCH
 LDA A #QPGO
 JSR FIXENB
* TEST EL CLOSING
 JSR CLEL
 BCC AG21
* DISABLE QUAD PITCH
 LDA A #QPGO
 JSR FIXDIS
 INC ELLK1
* OUTPUT CONFIG ENABLES
AG21 INC ELDAF
 INC AZDAF
 LDA A CONGO
 STA A PIA4DB
* START THRU SET?
 TST STF
 BNE AG25
* TEST TRAV OR EL BLOCK
 JSR ORBLK
 BCC AG25
 TST WPPASS
 BNE AG38
 CLR WPNF
AG38 JMP AG7
*
* LOOP 3 START
*
*
* TEST WEAPON SW
```



```
AG25 CLR STF
 TST WPNF
 BNE AG27
 JMP AG7
* SET START THRU
AG27 INC STF
*
* AUTO AZ LOOP
*
* AZ ENABLED?
 TST AZGOF
 BNE AG45
 JMP AG26
* TEST XENON ON
AG45 TST XRECF
 BNE AG40
 CLR AREDY
 CLR WPNF
* TEST READY
AG40 TST AREDY
 BNE AG19
* DISABLE PANTEL AZ
 LDA A #PAGO
 JSR FIXDIS
* DISABLE WEAPON AZ
 LDA A #AZGO
 JSR FIXDIS
* DISABLE WEAPON EL
 LDA A #ELGO
 JSR FIXDIS
 JMP AG29
* ENABLE WEAPON AZ
AG19 LDA A #AZGO
 JSR FIXENB
* TEST AZ LOCK
 TST AZLK2
 BNE AG26
* ENABLE PANTEL AZ
 LDAA #PAGO
 JSR FIXENB
* ENABLE PANTEL OFFSET
 LDAA #POGO
 JSR FIXENB
* TEST AZIMUTH CLOSING
 CLR SLOWF
 INC SLOWF
 JSR CLAZ
 BCC AG26
* START AUTO UPDATE
 TST DEL6
 BNE AG28
*
 LDA A LMODE
 ORA A #2
 STA A LMODE
 INC ATOUP
 INC DEL6
 BRA AG26
```

J8

```
*
AG28 TST TF6
 BNE AG26
* DISABLE PANTEL AZ
AG35 CLR SLOWF
 LDAA #PAGO
 JSR FIXDIS
 INC AZLK2
* FIX LMODE
 LDA A LMODE
 AND A #$FD
 STA A LMODE
 CLR DEL6
 CLR TF6
 CLR ATOUP
*
* AUTO EL LOOP
*
* EL ENABLED?
AG26 TST ELGOF
 BEQ AG30
* TEST EL  LOCK
AG43 TST ELLK2
 BNE AG30
* ENABLE QUAD PITCH
 LDA A #QPGO
 JSR FIXENB
* ENABLE QUAD OFFSET
 LDAA #QOGO
 JSR FIXENB
* DELAY WEAPON EL
 TST DEL4
 BNE AG54
 LDX #100
 STX TIM4
 INC TF4
 INC DEL4
*
AG54 TST TF4
 BNE AG53
* ENABLE WEAPON EL
 LDAA #ELGO
 JSR FIXENB
* TEST EL CLOSING
AG53 JSR CLEL
 BCC AG30
* DISABLE QUAD PITCH
 INC ELLK2
AG29 LDAA #OPGO
 JSR FIXDIS
* OUTPUT ENABLES AND DRIVE
AG30 LDA A #1
 STA A AZDAF
 STA A ELDAF
 LDA A CONGO
 STA A PIA4DB
 JMP AG7
 PAGE
```



```
*********AGLS2*******
*
*
* FORM CONFIGURATION WORD
*
OCE LDA A #$FF
 STA A CONGO
* TEST AUTO AZ
 LDAA #AZROM
 JSR TSTSW
 BCC OCE1
 LDA A #PLGO
 JSR FIXENB
 CLR AZGOF
 INC AZGOF
 BRA OCE3
* TEST PANTEL LEVEL
OCE1 CLR AZGOF
 LDA A #PLROM
 JSR TSTSW
 BCC OCE2
 LDA A #PLGO
 JSR FIXENB
* TEST PANTEL OFFSET
OCE2 LDA A #POROM
 JSR TSTSW
 BCC OCE3
 LDA A #POGO
 JSR FIXENB
* TEST AUTO EL
OCE3 LDA A #ELROM
 JSR TSTSW
 BCS OCE6
 CLR ELGOF
* TEST QUAD LEVEL
 LDA A #QLROM
 JSR TSTSW
 BCC OCE4
 LDA A #QCGO
 JSR FIXENB
 LDA A #QPGO
 JSR FIXENB
* TEST QUAD OFFSET
OCE4 LDA A #QOROM
 JSR TSTSW
 BCC OCE5
 LDA A #QPGO
 JSR FIXENB
 LDA A #QOGO
 JSR FIXENB
OCE5 RTS
* ENABLE AUTO EL
OCE6 CLR ELGOF
 INC ELGOF
 LDAA #QCGO
 JSR FIXENB
 LDAA #QPGO
 JSR FIXENB
```

J9

```
 RTS
*
* READ CONFIGURATION SWITCH REGISTER
*
CONRQ LDA A PIAODA
 STA A CONTEM
 COM A
 AND A #$F
 STA A AMODE
 RTS
*
* TEST CONFIG. SWITCH WORD
* C=SET IF SW ON:C=0 IF OFF
*
TSTSW AND A CONTEM
 BEQ TSTS1
 CLC
 RTS
TSTS1 SEC
 RTS
*
* FIX ENABLE WORD
*
FIXENB AND A CONGO
 STA A CONGO
 RTS
*
* READ TRIM ROUTINE
* AZIMUTH
RTRM LDA A PIA6DB
 LDX #SAD6B
 JSR SCON
 LDX #PIA6DB
 JSR GET
 STA A AZTRM
*
* ELEVATION
 LDA A PIA6DA
 LDX #SAD6A
 JSR SCON
 LDX #PIA6DA
 JSR GET
 STA A ELTRM
 RTS
*
* STROBE CONTROL PULSE(B REG)
*
SCON LDA B #$3E
 STA B O,X
 LDA B #$36
 STA B O,X
 RTS
*
* GET A/D DATA ROUTINE
* X=DATA REG. ADDRESS
*
GET LDA A 2,X
 NOP
```



```
 NOP
 LDA A O,X
 SUB A #$7F
 RTS
*
* READ ENCODERS ROUTINE
* ELEVATION AXIS
RENCS LDX #ELTEMP
 LDA A PIA2DB
 AND A #$7F
 JSR STBF
 LDA A PIA2DA
 JSR STBF
 LDA B PIA4DA
 JSR STBF1
*
* AZIMUTH AXIS
 LDX #AZTEMP
 LDA A PIA3DB
 AND A #$7F
 JSR STBF
 LDA A PIA3DA
 JSR STBF
 LDA A PIA4DA
 JSR STBF2
 RTS
*
* STORE DISPLAY BUFFER ROUTINE
*
STBF TAB
 JSR STBF2
 INX
 JSR STBF1
 INX
 RTS
*
* STORE DISPLAY BUFFER-1
STBF1 AND B #$OF
 STA B O,X
 RTS
*
* STORE DISPLAY BUFFER-2
STBF2 LSR A
 LSR A
 LSR A
 LSR A
 STA A O,X
 RTS
*
* READ ANALOG ERROR VOLTAGES
*
RAEV CLRA
 STA A MUXADD
 LDA B #5
 LDX #ERRBUF
* SETUP FOR REPEATED TRY'S
RAEV5 STA A PREVAL
 LDA A #5
```

J10

```
 STA A NUMRED
* LOOP ON A/D CHANNELS
RAEV2 LDA A #$34
 STA A SAD5
 LDA A MUXADD
 STA A PIA5DB
* 100 USEC DELAY
 LDA A #16
RAEV4 DEC A
 BNE RAEV4
 LDA A #$3C
 STA A SAD5
* WAIT EOC
 LDA A #32
RAEV3 DEC A
 BNE RAEV3
* READ AND STORE DATA
 LDA A PIA5DA
* TEST FOR CONSEC. READINGS
 CMP A PREVAL
 BNE RAEV5
 DEC NUMRED
 BNE RAEV2
 STA A 0,X
 INX
 LDA A MUXADD
 ADD A #$10
 STA A MUXADD
 DEC B
 BNE RAEV5
 CLR PIA5DB
 RTS
*
* TEST PIA8 SWITCHES (B SIDE)
* C SET IF SW ON:C=0 IF OFF
*
TSTS8B AND A PIA8DB
 BEQ TST81
 CLC
 RTS
TST81 SEC
 RTS
*
* TEST PIA8 SWITCHES (A SIDE)
* C SET IF SW ON:C=0 IF OFF
*
TSTS8A AND A PIA8DA
 BEQ TST82
 CLC
 RTS
TST82 SEC
 RTS
*
* SET RU FLAGS
*
SRUF TST RUCWF
 BNE SRUF3
 LDA A #RUCWM
```



```
 JSR TSTS8A
 BCC SRUF1
 INC RUCWF
*
SRUF1 TST RUCCWF
 BNE SRUF4
 LDA A #RUCCWM
 JSR TSTS8A
 BCC SRUF2
 INC RUCCWF
SRUF2 RTS
SRUF3 CLR RUCCWF
 LDA A #RUCCWM
 JSR TSTS8A
 BCC SRUF2
 CLR RUCWF
 CLR RUCCWF
 INC RUCCWF
 RTS
SRUF4 TST RUCWF
 BEQ SRUF2
 CLR RUCCWF
 RTS
*
* TEST LOAD POSITON TRANSITION
*
TLOAD EQU *
* TEST LOAD POSITION
 LDA A #LPOSM
 JSR TSTS8A
 BCC TL01
* SW. SET (LP)
* TEST LAST POSITION
 TST LPOSF
 BEQ TL02
 RTS
* WENT OE-LP
TL02 INC LPOSF
 JSR SLPOS
 BRA TL04
* SW NOT SET (OE)
* TEST LAST POSITION
TL01 TST LPOSF
 BNE TL03
 RTS
* WENT LP-OE
* COMM BUSY?
TL03 CLR LPOSF
 TST CBSY
 BNE TL03
* MOVE COMM BUFF TO DISP
 SEI
 JSR MVCD
 CLI
* MOVE DISP TO CONTROL
 JSR UFO
* RESTART ELEVATION
TL04 CLR WPNF
```

J11

```
 LDA A #$FF
 STA A WPNS
 INC LOADF
* DISABLE QUAD OFFSET
 LDA A #QOGO
 JSR FIXDIS
* DISABLE WPN ELEVATION
 LDA A #ELGO
 JSR FIXDIS
 RTS
*
* SET LOAD POSITION
*
SLPOS LDX #LOAD1
 LDA A #LDP1M
 JSR TSTSW
 BCC SLPO1
 LDX #LOAD2
*
 LDA A #LDP2M
 JSR TSTSW
 BCC SLPO3
 LDX #LOAD4
 BRA SLPO3
*
SLPO1 LDA A #LDP2M
 JSR TSTSW
 BCC SLPO3
 LDX #LOAD3
*
SLPO3 STX LOADX
* FIX CONTROL BUFFER
 LDX #ELGCDS
 LDA A LOADX
 JSR STBF
 LDA A LOADX+1
 JSR STBF
 CLR O,X
* FIX DISPLAY BUFFER
 LDX #ELFO
 LDA A LOADX
 JSR STBF
 LDA A LOADX+1
 JSR STBF
 CLR O,X
 RTS
*
*
*
* FIX DISABLE ROUTINE
*
FIXDIS COM A
 ORA A CONGO
 STA A CONGO
 RTS
*
* TEST OFFSET ERRORS
* C SET=OFFSET>ALLOWED
```



```
* C=O OFFSET OK
*
TERR LDX #ERRBUF
 DECB
 STA B HOLDB
 BEQ TERR1
*
TERR2 INX
 DEC B
 BNE TERR2
TERR1 LDA A O,X
 LDX #ERRVAL
 LDA B HOLDB
 BEQ TERR4
TERR3 INX
 DEC B
 BNE TERR3
TERR4 LDA B O,X
 TST A
 BPL TERR6
*
 NEG A
TERR6 CBA
 BPL TERR5
 CLC
 RTS
*
TERR5 SEC
 RTS
*
*
ERRVAL EQU *
 FCB EOP
 FCB EOC
 FCB EMP
 FCB EMC
 FCB EPA
*
* CHECK OFFSETS AND CONTROL GO/NO-GO
*
CKO CLR CKOF
 LDA B #5
 STA B SAVB
 CLR EFLAG
 CLR AZLIT
 CLR ELLIT
CKO2 LDA B SAVB
* CALL TEST ERROR
 JSR TERR
* SETUP ERROR WORD FOR DISPLAY
 BCC CKO8
 INC EFLAG
*
 LDAB SAVB
 CMPB #5
 BNE CKO4
* BYPASS A/P IF BD MODE
 TST BDF
```

J 12

```
 BNE CK04
 LDAA #4
 ORAA AZLIT
 STAA AZLIT
*
CK04 CMPB #4
 BNE CK05
 LDAA #2
 ORAA AZLIT
 STAA AZLIT
*
CK05 CMPB #3
 BNE CK06
 LDAA #1
 ORAA AZLIT
 STAA AZLIT
*
CK06 CMPB #2
 BNE CK07
 LDAA #2
 ORAA ELLIT
 STAA ELLIT
*
CK07 CMPB #1
 BNE CK08
 LDAA #1
 ORAA ELLIT
 STAA ELLIT
*
CK08 DEC SAVB
 BNE CK02
* TEST GO/NO-GO
 TST EFLAG
 BNE CK01
* TEST DIGITAL ERR (<1 MIL)
 LDA A #$F
 CMP A AZERD
 BNE CK01
*
 CMP A ELERD
 BNE CK01
* ENABLE GO
 LDA A #$3E
 STA A LAMP
 INC CKOF
 RTS
* ENABLE NO-GO
CK01 LDA A #$36
 STA A LAMP
 INC CKOF
 RTS
*
* TEST AZ ERR RANGE
* (D/A FORMAT)
*
TAZERR LDX AZERR
 LDA B #AZLIM
 JSR TSTIT
```



```
 RTS
*
* TEST EL ERR RANGE
*
TELERR LDX ELERR
 LDA B #ELLIM
 JSR TSTIT
 RTS
*
*
* TEST D/A FORMAT
*
TSTIT STX SAVA
* FIX FOR SIGN
 TST SAVA
 BPL TSTI3
 LDAA SAVA
 ANDA #$7F
 STAA SAVA
*
TSTI3 LDAA SAVA
 AND A #$F
 BEQ TSTI1
 SEC
 RTS
* CHECK LSB
TSTI1 LDA A SAVB
 CBA
 NOP
 BHI TSTI2
 CLC
 RTS
TSTI2 SEC
 RTS
*
* COMPUTE AZ ERROR
*
COMPAZ CLR SIG
 CLR DISAZ
 LDX #TEMBCD
* CONVERT TRIM TO BCD
 CLR A
 LDA B AZTRM
 BPL COPAZ1
 INC SIG
 NEG B
COPAZ1 NOP
 NOP
 JSR BINBCD
* ADD TRIM TO ENCODER READING
 LDX #TEMBCD
 STX A1
 LDX #AZTEMP
 STX A2
 LDX #RESULT+4
 JSR BCDADD
* ADJUST FOR ROLLOVER
 LDX #AZDISP
```

J13

```
 JSR ADJ
* SAVE TRIMMED ENC. VALUE
 LDX #TRMENC
 STX ISAVES
 LDX #AZDISP
 LDA B #5
 JSR TCS
* BASE DEFL. SELECT ?
 LDX #REF
 TST BDF
 BEQ COPAZ5
 LDX #BDBUF
* SUBTRACT REF. OR BD ANGLE
COPAZ5 STX S2
 LDX #AZDISP
 STX S1
 LDX #RESULT
 JSR BCDSUB
* ADJUST FOR ROLLOVER
 LDX #AZDISP
 JSR ADJ
*
* SUBTRACT RESULT FROM GACS
*
 LDX #AZDISP
 STX S2
 LDX #AZGCDS
 STX S1
 LDX #RESULT
 JSR BCDSUB
* ADJUST FOR ROLLOVER
 LDX #AZERD
 JSR ADJ
* FIX FOR + OR - 32000
 CLR SIG
 LDX #AZERD
 JSR TEST32
 BCC COPAZ3
* FIX IF > 32000
 LDX #AZERD
 STX A1
 JSR NINCOM
 LDX #CONST
 STX A2
 LDX #RESULT+4
 JSR BCDADD
*
 LDX #AZERD
 JSR XFER
*
 CLR SIG
 INC SIG
* TEST MAGNITUDE OF DISPLAY VALUE
COPAZ3 LDX #AZERD
 JSR TESTM
* FIX SIGN OF DISPLAY
 BCC COPAZ4
* BLANK SIGN IF < 1 MIL
```



```
 LDA A #$F
 BRA COPAZ2
COPAZ4 LDA A #1
 TST SIG
 BEQ COPAZ2
 LDA A #2
COPAZ2 STA A AZERD
* CONVERT ERROR TO BINARY
 LDX #AZERD
 JSR BCDBIN
 STX AZERR
 INC DISAZ
 RTS
CONST FCB 6,4,0,0,0
CONST2 FCB 3,6,0,0,0
*
* COMPUTE ELEVATION ERROR
*
COMPEL CLR SIG
 CLR DISEL
 LDX #TEMBCD
* CONVERT TRIM TO BCD
 CLR A
 LDA B ELTRM
 BPL COPEL1
 INC SIG
 NEG B
COPEL1 NOP
 NOP
 JSR BINBCD
* ADD TRIM TO ENCODER READING
 LDX #TEMBCD
 STX A1
 LDX #ELTEMP
 STX A2
 LDX #RESULT+4
 JSR BCDADD
* ADJUST FOR ROLLOVER
 LDX #ELDISP
 JSR ADJ
* SUBTRACT RESULT FROM GACS
 LDX #ELDISP
 STX S2
 LDX #ELGCDS
 STX S1
 LDX #RESULT
 JSR BCDSUB
* ADJUST FOR ROLLOVER
 LDX #ELERD
 JSR ADJ
* FIX FOR + OR - 32000
 CLR SIG
 LDX #ELERD
 JSR TEST32
 BCC COPEL3
* FIX IF > 32000
 LDX #ELERD
 STX A1
```

J14

```
 JSR NINCOM
 LDX #CONST
 STX A2
 LDX #RESULT+4
 JSR BCDADD
*
 LDX #ELERD
 JSR XFER
*
 CLR SIG
 INC SIG
* TEST MAGNITUDE OF DISPLAY VALUE
COPEL3 LDX #ELERD
 JSR TESTM
* FIX SIGN OF DISPLAY
 BCC COPEL4
* BLANK SIGN IF < 1 MIL
 LDA A #$F
 BRA COPEL2
COPEL4 LDA A #1
 TST SIG
 BEQ COPEL2
 LDA A #2
COPEL2 STA A ELERD
* CONVERT ERROR TO BINARY
 LDX #ELERD
 JSR BCDBIN
 STX ELERR
 INC DISEL
 RTS
*
* BINARY-BCD CONVERSION
* X=ADDRESS OF RESULT(5)
* A,B= BINARY VALUE
*
BINBCD STX HOLDX
 STX OUTX
 STX STORX
 STX A1
*
 LDX #10000
 JSR TIZ
 LDX #1000
 JSR TIZ
 LDX #100
 JSR TIZ
 LDX #10
 JSR TIZ
 LDX OUTX
 STA B 0,X
* TEST SIGN FLAG
 TST SIG
 BNE BIN1
 RTS
* COMPLEMENT RESULT (MOD 64000)
BIN1 LDX HOLDX
 JSR NINCOM
 LDX #CONST
```



```
 STX A2
 LDX #RESULT+4
 JSR BCDADD
 LDX STORX
 JSR XFER
 RTS
*
* TEST NUMBER OF TIMES VALUE DIVISIBLE
*
TIZ STX TIZX
 CLR CNX
TIZ2 STA A OLDAX
 STA B OLDBX
* TRIAL SUBTRACT
 SUB B TIZX+1
 SBC A TIZX
 BCS TIZ1
 INC CNX
 BRA TIZ2
* FAIL SUBTRACT
TIZ1 LDX OUTX
 LDA A CNX
 STA A 0,X
 INX
 STX OUTX
 LDAA OLDAX
 LDA B OLDBX
 RTS
*
* NINES COMPLEMENT 5 DIGIT BCD #
* X= ADDRESS OF BCD MSB(BEFORE AND AFTER)
*
NINCOM LDA B #4
 JSR FIXX
 STX KEEP
* COMPLEMENT EACH DIGIT
 LDA B #5
NIN1 LDA A #09
 SUB A 0,X
 STA A 0,X
 DEX
 DEC B
 BNE NIN1
* ADD ONE TO RESULT
 LDX KEEP
 LDA B #5
 CLR A
 SEC
*
NIN2 ADC A 0,X
 JSR JOCK
 AND A #$F
 STA A 0,X
*
 LDA A #0
 DEX
 DEC B
 BNE NIN2
```

J 15

```
 RTS
*
* FIX X REG. POINTER
*
FIXX INX
 DEC B
 BNE FIXX
 RTS
*
* ADD 2-5 DIGIT BCD VALUES
* A1=ADDRESS OF VALUE 1 MSB
* A2=ADDRESS OF VALUE 2 MSB
* X=ADDRESS OF RESULT
*
BCDADD STX ADDX1
 CLR SIG
* FIX A1
 LDX A1
 LDA B #4
 JSR FIXX
 STX A1
* FIX A2
 LDX A2
 LDA B #4
 JSR FIXX
 STX A2
 CLC
* GET FIRST VALUE
 LDAB #4
BCA1 LDX A1
 LDA A 0,X
 DEX
 STX A1
* ADD SECOND
 LDX A2
 ADC A 0,X
 JSR JOCK
 DEX
 STX A2
* STORE IN OUTPUT
 LDX ADDX1
 STA A 0,X
 DEX
 STX ADDX1
*
 DEC B
 BNE BCA1
 LDX A1
 LDAA 0,X
 LDX A2
 ADCA 0,X
 JSR JOCK
 BCC BCA2
 INC SIG
BCA2 LDX ADDX1
 STAA 0,X
 RTS
JOCK NOP
```



```
 CMPA #9
 BGT JOCK1
 CLC
 RTS
JOCK1 ADDA #6
 ANDA #$F
 SEC
 RTS
*
* SUBTRACT 2-5 DIGIT BCD VALUES
* S1=ADDRESS OF MINUEND
* S2=ADDRESS OF SUBTRAHEND
* X=ADDRESS OF RESULT
*
BCDSUB STX SUBX1
 STX SUBX2
* FIX SUBX1
 LDAB #4
 JSR FIXX
 STX SUBX1
* COMPLEMENT SUBTRAHAND
* TRANSFER SUBTRAHAND
 LDX #TEMSUB
 STX TX
 LDAB #5
TX1 LDX S2
 LDAA 0,X
 INX
 STX S2
*
 LDX TX
 STAA 0,X
 INX
 STX TX
*
 DECB
 BNE TX1
*
 LDX #TEMSUB
 JSR COMP64
* ADD MINUEND AND FIX SIGN OF RESULT
 LDX S1
 STX A1
 LDX #TEMSUB
 STX A2
 LDX SUBX1
 JSR BCDADD
 RTS
*
* 64'S COMPLEMENT ROUTINE
*
COMP64 STX A1
 JSR NINCOM
 LDX #CONST
 STX A2
 LDX #RESULT+4
 JSR BCDADD
 LDX A1
```

J16

```
ISR XFER
RTS

 ADJUST FOR > 99999 & > 64000 ROLLOVER

IJ STX ADJX
ST SIG
NE ADJ1
TEST > 64000
.DX #RESULT
SR TEST64
CC ADJ2
FIX > 64000 ROLLOVER
J1 LDX #RESULT
X A1
DX #CONST2
TX A2
DX ADJX
.DAB #4
SR FIXX
SR BCDADD
TS
FIX < 64000 VALUE
J2 LDX ADJX
SR XFER
S
TEST IF BCD ARRAY > OR = 32000 (C SET IF TRUE)

ST32 LDAA 0,X
PA #3
LT T321
GT T322
.DAA 1,X
PA #2
.T T321
22 SEC
TS
21 CLC
S

TEST IF BCD ARRAY >OR= 14000
 SET IF TRUE

ST14 LDA A 0,X
MP A #1
.T T141
T T142
DA A 1,X
MP A #4
LT T141
42 SEC
S
41 CLC
S

********AGLS3*********
```



```
*
*
* TEST BCD MAG LIMITS FOR DISPLAY
*
* X=BUFFER ADDRESS
* C=SET IF BLANK SIGN
* SETUP DIRECTION SIGN
* FIX IF >9999
TESTM LDA A 0,X
 AND A #$F
 BEQ TSTM1
 LDA A #09
 STA A 1,X
 STA A 2,X
 STA A 3,X
 STA A 4,X
 BRA TSTM3
* BLANK SIGN IF < 1 MIL
TSTM1 LDA B #4
TSTM2 LDA A 0,X
 BNE TSTM3
 INX
 DEC B
 BNE TSTM2
 SEC
 RTS
TSTM3 CLC
 RTS
* TEST IF BCD ARRAY > OR = 64000
* C SET IF TRUE:C=0 IF FALSE
*
TEST64 LDA A 0,X
 CMP A #6
 BGT T641
 BLT T642
 LDA A 1,X
 CMP A #4
 BGE T641
T642 CLC
 RTS
T641 SEC
 RTS
*
*
* TRANSFER FROM BCD ARRAY "RESULT"
* ARRAY SPECIFIED BY X
*
XFER STX X1
 LDX #RESULT
 STX X2
 LDA B #5
XFER1 LDX X2
 LDA A 0,X
 INX
 STX X2
 LDX X1
 STA A 0,X
 INX
```

J17

```
STX X1
DEC B
BNE XFER1
RTS
 COMPUTE BINARY DATA FROM BCD(5 DIGIT) VALUE
 X=ADDRESS OF BUFFER(ENTRY);D/A VALUE(EXIT)

BCDBIN INX
LDA B #4
STA B SAVB
CLR MSBY
CLR LSBY

BCDB1 LDA A 0,X
STA A TMP
JSR M1OX
INX
DEC SAVB
BNE BCDB1
* DIVIDE X2
LSR MSBY
ROR LSBY
* TEST SIGN
LDA A MSBY
TST SIG
BEQ BCDB2
ORA A #$80
STA A MSBY
* TEST OVERFLOW
BCDB2 AND A #$70
BEQ BCDB3
*
LDA A MSBY
AND A #$80
ORA A #$0F
STA A MSBY
LDA A #$FF
STA A LSBY
STA A PIABDB
BCDB3 LDX MSBY
RTS
*
* CLEAR FLAG TABLES
*
CLFG CLR 0,X
INX
CPX #END
BNE CLFG
RTS
*
* CLEAR BASE DEF. BUFFER
*
CLRBD LDA B #5
LDX #BDBUF
CLRDB1 CLR 0,X
INX
DEC B
BNE CLRDB1
```



```
RTS
*
* READ AND STORE BASE DEFLECTION
*
FIXBD EQU *
* SUBTRACT 3200 FROM CURRENT READI?
LDX #TRMENC
STX S1
LDX #CONST3
STX S2
LDX #RESULT
JSR BCDSUB
* ADJUST FOR ROLLOVER
LDX #BDBUF
JSR ADJ
RTS
*
CONST3 FCB 3,2,0,0,0
*
* INITIAL PIAS ROUTINE
*
PIAS EQU *
* PIA 0 (A)-SW INPUTS
CLR PIA0CA
CLR PIA0DA
LDA A #$3E
STA A PIA0CA
LDA A #$3E
STA A PIA0CA
* PIA 0 (B)-PROG CLOCK
CLR PIA0CB
LDA A #$FF
STA A PIA0DB
LDA A #$2C
STA A PIA0CB
LDA A #100
STA A PIA0DB
LDA A PIA0DB
LDA A #$2D
STA A PIA0CB
SEI
* PIA1 -MANUAL G.O. INPUTS
LDA A #$36
LDX #PIA1DA
CLR 2,X
CLR 3,X
CLR 0,X
LDA B #$FF
STA B 1,X
STA A 2,X
STA A 3,X
LDA A 0,X
LDA A 1,X
* PIA 2 -ELEVATION ENCODER
LDA A #$3E
LDX #PIA2DA
JSR SETUP
* PIA3 -AZIMUTH ENCODER
```

J18

PAGE 035 AGLS1 .SA:1

```
 LDA A #$3E
 LDX #PIA3DA
 JSR SETUP
*
* PIA 4 -EL AND AZ ENCODER
 CLR PIA4CA
 CLR PIA4DA
 LDA A #$3E
 STA A PIA4CA
* PIA 4 -ENABLES
 LDA A #4
 STA A PIA4CB
 LDA A #$FF
 STA A PIA4DB
 CLR PIA4CB
 LDA A #$FF
 STA A PIA4DB
 LDA A #4
 STA A PIA4CB
*
* PIA 5 -MUX A/D
 CLR PIA5CA
 CLR PIA5CB
 CLR PIA5DA
 LDA A #$FF
 STA A PIA5DB
 LDA A #$3C
 STA A PIA5CA
 LDA A #$34
 STA A PIA5CB
*
* PIA 6 -TRIM A/D
 LDA A #$36
 LDX #PIA6DA
 JSR SETUP
*
* PIA 7 -D/A
 CLR PIA7CA
 CLR PIA7CB
 LDA A #$FF
 STA A PIA7DA
 STA A PIA7DB
 LDA A #$3E
 STA A PIA7CA
 STA A PIA7CB
*
* PIA 8 -DISPLAY
 CLR PIA8CA
 CLR PIA8CB
 LDA A #$F
 STA A PIA8DA
 LDA A #$1F
 STA A PIA8DB
 LDA A #$3E
 STA A PIA8CA
 LDA A #$3E
 STA A PIA8CB
*
```

PAGE 036 AGLS1 .SA:1

```
* ACIA SETUP
 LDAA #$43
 STA A AC1C
 STAA AC2C
 LDX #AC2C
 JSR DISXMT
 LDX #AC1C
 LDA A #$49
 STA A 0,X
 NOP
 NOP
 RTS
*
* SETUP PIA USING X AND A REG.
*
SETUP CLR 2,X
 CLR 3,X
 CLR 0,X
 CLR 1,X
 STA A 2,X
 STA A 3,X
 RTS
*
* INTERRUPT SERVICE ROUTINE
*
* TEST CLOCK
ISER TST PIA0CB
 BPL ISER6
 LDA A PIA0DB
 STA A PIA0DB
* SCAN CLOCKS
 LDX #TMTB
 LDA B #6
 JSR SCAT
* TEST INSIDE LOOP(20 MSEC)
 TST TF1
 BEQ ISER2
 RTI
* SERVICE INSIDE LOOP
ISER2 LDX #2
 STX TIM1
 INC TF1
* D/A READY?
 JSR DAOUT
* TEST XENON ON
 LDAA #XRECM
 JSR TSTS8B
 BCC ISER7
* START DROPOUT CLOCK
 CLR TF5
 LDX #100
 STX TIM5
 INC TF5
* TEST ON FOR 25 TIMES
 INC XON
 LDAA XON
 CMPA #25
 BLT ISER8
```

```
*
 DEC XON
 CLR XRECF
 INC XRECF
* TEST TIMEOUT AFTER OFF
ISER7 CLR XON
 TST TF5
 BNE ISER8
 CLR XRECF
* TEST WEAPON SW
ISER8 LDAA #WPNM
 JSR TSTS8A
 BCC ISER4
 CLR WPNF
 INC WPNF
* FIX WPN SW WORD
 LDA A WPNS
 CMP A #$FF
 BNE ISER4
 CLR WPNS
 INC WPNS
* OUTSIDE LOOP
* NO SERVICE IF COMM BUSY
ISER4 TST CBSY
 BEQ ISER1
 RTI
ISER1 TST TF2
 BEQ ISER5
 RTI
* SERVICE OUTSIDE LOOP
ISER5 LDX #20
 STX TIM2
 INC TF2
* UPDATE DISPLAY
 JSR UDIS
 RTI
*TEST COMM.
*
*
* COMM ROUTINE
*
*
ISER6 TST ACIS
 BMI ISER31
 JMP ISER20
ISER31 CLR CBSY
 INC CBSY
* TEST RECV INT
 LDX #ACIS
 JSR AOI
 BCS ISER26
 JMP ISER32
*TEST CHAR =R?
ISER26 CMP A #'R
 BNE ISER21
* INITIALIZE F. O. RECIEVE
 LDX #CBUF
 STX PTRC
```

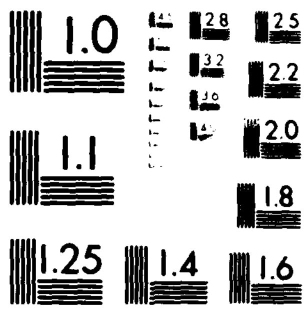


```
 LDX #CBUF+15
 STX PTRE
 CLR COMERR
 RTI
*TEST CHAR =T?
ISER21 CMP A #'T
 BNE ISER22
* INITIALIZE F.O. TRANSMIT
 LDX #ELFO
 STX PTRC
 LDX #LMODE
 STX PTRE
* OUTPUT STARTER
 LDX #ACIS
 JSR CRLF
* INH RECV/ENB XMIT
 LDX #ACIC
 JSR DISREC
 RTI
* RECIEVED DATA
ISER22 SUB A #$30
 BPL ISER23
 INC COMERR
ISER23 CMP A #9
 BLE ISER24
 INC COMERR
ISER24 LDX PTRC
 CPX #0
 BEQ ISER29
* STORE DATA
 STA A 0,X
 CPX PTRE
 BNE ISER33
 TST COMERR
 BNE ISER29
* TEST RANGE OF DATA
 INC COMERR
 LDX #CBUF
 JSR TEST14
 BCS ISER29
 LDX #CBUF+5
 JSR TEST64
 BCS ISER29
* IF AUTO UPDATE
 CLR COMERR
 TST ATOUP
 BNE ISER34
* MOVE FROM COM BUFF TO DISP
 JSR MVCD
ISER34 INC NEWFO
* PREP SW WORD
 LDA A #$FF
 STA A WPNS
ISER29 LDX #0
 STX PTRC
 CLR CBSY
 JSR DISXMT
 RTI
```

AD-A097 521 HONEYWELL INC HOPKINS MN DEFENSE SYSTEMS DIV F/G 19/6
AUTOMATED GUN LAYING SYSTEM FOR SELF-PROPELLED ARTILLERY WEAPON--ETC(U)
MAY 80 E E LEHTOLA, K A HERZING DAAA09-76-C-0284
UNCLASSIFIED NL

5 OF 5

END
DATE FILMED
5 81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

```
* TEST STATUS REGISTER
ISER33 LDA A ACIS
 AND A #%01110000
 BEQ ISER25
 INC COMERR
ISER25 INX
 STX PTRC
 RTI
*TRANSMIT INTERRUPT?
ISER32 LDA A ACIS
 BIT A #2
 BNE ISER19
 LDA A ACIR
 RTI
*TEST RECIEVE ERROR
ISER19 TST COMERR
 BEQ ISER27
 LDA A #'X
ISER28 LDX #ACIS
 JSR A(X)
 CLR CBSY
 LDX #0
 STX PTRC
* INH XMIT/ENB RECV
 LDX #ACIC
 JSR DISXMT
 RTI
* TRANSMIT CHARACTER
ISER27 LDX PTRC
 CPX #0
 BEQ ISER28
 LDA A 0,X
 ADD A #$30
 CPX PTRE
 BEQ ISER28
 INX
 STX PTRC
 LDX #ACIS
 JSR A(X)
 RTI
*
*
*
* TEST CRT
ISER20 TST CBSY
 BEQ ISER30
* RESET INT
 LDA A AC2R
 RTI
ISER30 TST AC2S
 BMI ISER16
 RTI
* TEST RECV. INT.
ISER16 LDX #AC2S
 JSR AOI
 BCC ISER17
*  =D ?
 CMPA #'D
```



```
 BNE ISER9
 LDX #ELFO
 STX PTR
 LDX #LMODE+1
 STX PTE
 LDAA #5
 BRA ISER10
*  =F ?
ISER9 CMPA #'F
 BEQ ISER11
 RTI
ISER11 LDX #BEG
 STX PTR
 LDX #ENDF
 STX PTE
 LDAA #1
ISER10 STAA SPC
 STAA ASP
* OUTPUT CR/LF
 LDX #AC2S
 JSR CRLF
* INH REC/ENB XMIT
 LDX #AC2S
 JSR DISREC
 RTI
*
* TEST XMIT. INT.
ISER17 LDAA AC2S
 BITA #2
 BNE ISER12
 LDAA AC2R
 RTI
* CHECK SPACE COUNT
ISER12 TST ASP
 BNE ISER13
 LDAA SPC
 STAA ASP
 LDAA #$20
 BRA ISER14
* PROCESS CHAR
ISER13 DEC ASP
 LDX PTR
 CPX PTE
 BEQ ISER15
 LDAA 0,X
 INX
 STX PTR
 BPL ISER18
 CLRA
ISER18 ADDA #$30
* OUTPUT CHAR
ISER14 LDX #AC2S
 JSR A(X)
 RTI
* WRAP UP XMIT
ISER15 LDX #AC2C
 JSR DISXMT
 RTI
```

J21

```
*
* MOVE FROM COMM BUFFER TO DISPLAY
* EL(5),AZ(5),REF(5),CMODE
*
MVCD EQU *
* TEST LOAD POS.
 LDA A #LPOSM
 JSR TSTSBA
 BCS MV3
 LDX #ELFO
 STX ISAVES
 LDX #CBUF
*
 LDA B #5
 JSR TCS
*
MV3 LDX #AZFO
 STX ISAVES
 LDX #CBUF+5
 LDA B #5
*
 JSR TCS
*
* MOVE REF ANGLE
 LDX #RHOLD
 STX ISAVES
 LDX #CBUF+10
 LDA B #5
 JSR TCS
 LDA A CBUF+15
 STA A CMODE
*
 RTS
*
* TRANSFER CHAR STRING
* ISAVES=DEST,X=SOURCE,B=CNT
*
TCS STX ISAVEX
TCS1 LDX ISAVEX
 LDA A O,X
 INX
 STX ISAVEX
*
 LDX ISAVES
 STA A O,X
 INX
 STX ISAVES
 DEC B
 BNE TCS1
 RTS
*
* DISABLE REC INT ROUTINE
DISREC LDAA #XIE
 STAA O,X
 RTS
*
* DISABLE XMIT INT ROUTINE
DISXMT LDAA #RIE
```



```
 STAA O,X
 LDAA 1,X
 RTS
*
* OUTPUT D/A ROUTINE
*
* ELEVATION
DAOUT TST ELDAF
 BEQ DAOUT1
 LDX ELCOM
 STX PIA7DA
 LDX #PIA7CA
 JSR USCON
*
* AZIMUTH
DAOUT1 TST AZDAF
 BEQ DAOUT4
 LDX AZCOM
 STX PIA7DA
 LDX #PIA7CB
 JSR USCON
DAOUT4 RTS
*
* INVERTED STROBE CONTROL PULSE
*
USCON LDA B #$36
 STA B O,X
 NOP
 LDA B #$3E
 STA B O,X
 RTS
*
* UPDATE DISPLAY ROUTINE
* ELEVATION
*
UDIS EQU *
* TEST DISPLAY INHIBIT
* O=NO DISPLAY
 TST DISEL
 BEQ UDIS1
 CLR DISADR
 LDX #ELFO
 STX ACT
 LDAB ELLIT
 STA B LITE
 JSR DISIT
* AZIMUTH
* TEST DISPLAY INHIBIT
* O=NO DISPLAY
UDIS1 TST DISAZ
 BEQ UDIS2
 LDA A #$10
 STA A DISADR
 LDX #AZFO
 STX ACT
 LDAB AZLIT
 STA B LITE
 JSR DISIT
```

J22

```
UDIS2 TST DTHRU
 BEQ UDIS3
 CLR DTHRU
 RTS
*
UDIS3 INC DTHRU
 RTS
*
* ROUTINE TO DISPLAY IT
*
DISIT LDA A DISADR
 STAA PIA8DB
 LDX ACT
 JSR TFS
 BCS DISIT3
 LDA A 0,X
DISIT3 INX
 STX ACT
 STA A PIA8DA
 LDX #PIA8CA
 JSR USCON
 INC DISADR
 LDA A DISADR
 AND A #$0F
 CMPA #$0F
 BNE DISIT
* OUTPUT SPECIAL DISPLAYS
 TST CKOF
 BNE DISIT2
 RTS
DISIT2 LDA A DISADR
 STAA PIA8DB
 LDX #CODE
 CLR A
 LDA B LITE
 BEQ DISIT1
 JSR FIXX
 LDAA 0,X
DISIT1 STAA PIA8DA
 LDX #PIA8CA
 JSR USCON
 RTS
*
* TEST FLASH SEQ (DISPLAYS)
*
TFS EQU *
* TEST ADDR RANGE
 AND A #$0F
 CMP A #4
 BGT TFS3
*
 TST ATOUP
 BNE TFS3
 TST NEWFO
 BNE TFS1
TFS3 CLC
 RTS
*
```



```
TFS1 TST DTHRU
 BEQ TFS3
 LDA A #$F
 SEC
 RTS
* DISPLAY CODE TABLE
CODE EQU *-1
 FCB 5
 FCB 2
 FCB $F
 FCB 4
 FCB 6
 FCB 1
 FCB $F
**********AGLS4***********
*
* EXEC SUBROUTINES
* MULTIPLY MSBY/LSBY X 10+TMP
M10X LDA A MSBY
 LDA B LSBY
 CLC
 ASL A
 ASL B
 JSR CKC
 ASL A
 ASL B
 JSR CKC
 ADD B LSBY
 JSR CKC
 ASL A
 ASL B
 JSR CKC
 ADD B TMP
 JSR CKC
 STA A MSBY
 STA B LSBY
 RTS
* CHECK C BIT AND FIX
CKC BCS CKC1
 RTS
CKC1 INC A
 CLC
 RTS
*
* SCAN TIMERS ROUTINE
*
SCAT TST 0,X
 BEQ RT
 LDA A 2,X
 SUB A 0,X
 STA A 2,X
 BNE RT
 TST 1,X
 BEQ ST3
 DEC 1,X
 BRA RT
ST3 CLR 0,X
RT INX
```

```
 INX
 INX
 DEC B
 BNE SCAT
 RTS
* CLEAR TIMERS ROUTINE
CLTM LDA B #18
 LDX #TMTB
CLT1 CLR 0,X
 INX
 DEC B
 BNE CLT1
 RTS
* INPUT FROM ACIA
AOI LDAA 0,X
 BITA #1
 BEQ AOI1
 LDAA 1,X
 SEC
 RTS
AOI1 CLC
 RTS

* OUTPUT TO ACIA
AOO PSHA
 LDAA 0,X
 BITA #2
 PULA
 BEQ AOO1
 STAA 1,X
 SEC
 RTS
AOO1 CLC
 RTS
* CR/LF ROUTINE
CRLF LDAA #$D
 JSR AOOL
 LDAA #$A
 JSR AOOL
 RTS
* LOOP ON OUTPUT
AOOL JSR AOO
 BCC AOOL
 RTS
*************AGLS5***********
*
* TEST AZ CLOSING
*
* TEST PASS FLAG
CLAZ TST PASSAZ
 BNE CLAZ1
* TEST AZERR
 LDA A AZERR
 AND A #$80
 BEQ CLAZ6
* AZERR < 0
 LDX #15
 STX TIM3
```



```
 LDAA #1
 STAA PASSAZ
 LDX #FULBAK
 TST SLOWF
 BEQ CLAZ8
 LDX #HAFBAK
CLAZ8 STX AZCOM
 CLC
 RTS
* PASS = 1
CLAZ1 LDAA PASSAZ
 DECA
 BNE CLAZ3
* TEST FOR FIRST NULL
 LDX #FULBAK
 TST SLOWF
 BEQ CLAZ9
 LDX #HAFBAK
CLAZ9 STX AZCOM
 TST AZERR
 BPL CLAZ4
 CLC
 RTS
* NULL ACHIEVED
CLAZ4 LDX #HAFBAK
 STX AZCOM
 INC TF3
 INC PASSAZ
 CLC
 RTS
* PASS = 2
CLAZ3 DECA
 BNE CLAZ2
* TEST TIMER
 TST TF3
 BEQ CLAZ6
 CLC
 RTS
* DISABLE D/A
CLAZ6 LDX #0
 STX AZCOM
 LDA A #3
 STA A PASSAZ
 CLC
 RTS
* PASS=3
CLAZ2 DEC A
 BNE CLAZ6
 JSR AZNULL
 BCS CLAZ7
 LDAA AZERR
 LDAB AZERR+1
 JSR IMAGA
 STAA AZCOM
 STAB AZCOM+1
 CLC
 RTS
CLAZ7 LDX #0
```

J24

```
 STX AZCOM
 CLR PASSAZ
 SEC
 RTS
*
* TEST AZ NULL
*
* AZERR TEST
AZNULL JSR TAZERR
 BCS AZN1
 INC AZCNT
 LDAA AZCNT
 CMP A #25
 BLE AZN2
 DEC AZCNT
 SEC
 RTS
AZN1 CLR AZCNT
AZN2 CLC
 RTS
*
* TEST EL CLOZING
*
* TEST ELERR
CLEL LDX ELERR
 STX ELCOM
 CPX #0
 BNE CLEL1
 RTS
* TEST NULL
CLEL1 JSR ELNULL
 BCS CLEL2
 RTS
CLEL2 LDX #0
 STX ELCOM
 SEC
 RTS
*
*
* TEST EL NULL
*
* ELERR TEST
ELNULL JSR TELERR
 BCS ELN1
 INC ELCNT
 LDAA ELCNT
 CMPA #25
 BLE ELN2
 DEC ELCNT
 SEC
 RTS
ELN1 CLR ELCNT
ELN2 CLC
 RTS
*
* TEST QUAD PITCH NULL
*
CLOP LDA B #QPMA
```



```
 JSR TERRQ
 BCS CLQP2
 INC QPCNT
 LDA A QPCNT
 CMP A QPLP
 BNE CLQP3
 DEC QPCNT
 SEC
 RTS
*
CLQP2 CLR QPCNT
*
CLQP3 CLC
 RTS
* TEST & FIX A/D MAGNITUDE
TMAGA CLR NEGF
 TSTA
 BPL TMAG3
 INC NEGF
TMAG3 ANDA #$F
 TST SLOWF
 BNE TMAG1
* FULL SPEED
 CMPA #2
 BGE TMAG2
 BSR M8X
 RTS
*
TMAG2 LDAA #$0F
 TST NEGF
 BEQ TMAG4
 ORAA #$80
TMAG4 LDAB #$FF
 RTS
* HALF SPEED
TMAG1 TSTA
 BNE TMAG5
 CMPB #$6F
 BHI TMAG5
 BSR M8X
 RTS
*
TMAG5 LDAA #5
 TST NEGF
 BEQ TMAG6
 ORAA #$80
TMAG6 LDAB #$FF
 RTS
*
* MULTIPLY X 8
*
M8X LDX #3
M81 ASLB
 ROLA
 DEX
 BNE M81
 TST NEGF
 BEQ M82
```

```
 ORAA #$80
MB2 RTS
*
*
* OR BLOCK FLAGS
*
ORBLK TST TBLK
 BNE ORB1
 TST EBLK
 BNE ORB1
 CLC
 RTS
ORB1 SEC
 RTS
*
*
*
* TEST XENON STABILITY
XSTAB TST XRECF
 BEQ XS1
 INC XTIME
 LDAA XTIME
 CMPA #70
 BLS XS2
 DEC XTIME
 SEC
 RTS *
*
XS1 CLR XTIME
XS2 CLC
 RTS
*
* UPDATE FIRE ORDER
*
UFO SEI
 STS SAVES
* TEST LOAD POS.
 LDA A #LPOSM
 JSR TSTSBA
 BCS UFO3
 LDS #ELFO-1
 LDX #ELOCDS
 LDA B #5
UFO1 PUL A
 STA A O,X
 INX
 DEC B
 BNE UFO1
UFO3 LDS #AZFO-1
 LDX #AZOCDS
 LDA B #5
UFO2 PUL A
 STA A O,X
 INX
 DEC B
 BNE UFO2
 LDS SAVES
* MOVE REF ANGLE
```



```
 LDX #REF
 STX ISAVES
 LDX #RHOLD
 LDA B #5
 JSR TCS
 CLI
 RTS
*
* TEST AUTO MODE
*
AUTO LDA A CONGO
 COM A
 AND A #%11000000
 BNE AUTO1
 CLC
 RTS
AUTO1 SEC
 RTS
* TEST QUAD PITCH ERROR
*
TERRQ LDA A ERRBUF
 BPL TERQ1
 NEG A
TERQ1 CMP A #QPLIM
 BPL TERQ2
 CLC
 RTS
TERQ2 SEC
 RTS
*
* READ MANUAL INPUT
*
RMAN EQU *
* TEST IF NEW VALUE
 LDX #CBUF
 CLR B
RMN2 JSR RTHM
 CMP A O,X
 BNE RMN1
 INX
 CMP B #10
 BNE RMN2
 RTS
* UPDATE BUFFER
RMN1 LDX #CBUF
 CLR B
RMN3 JSR RTHM
  STA A O,X
 INX
 CMP B #10
 BNE RMN3
* FIX REF ANGLE (=0)
 LDA B #6
 CLR A
RMN6 STA A O,X
 INX
 DEC B
 BNE RMN6
```

```
* TEST RANGE OF DATA
 CLR COMERR
 INC COMERR
 LDX #CBUF
 JSR TEST14
 BCS RMN4
 LDX #CBUF+5
 JSR TEST64
 BCS RMN4
* IF AUTO UPDATE
 CLR COMERR
 TST ATOUP
 BEQ RMN5
 INC NEWFO
 RTS
* MOVE FROM COMM TO DISP BUFFER
RMN5 SEI
 JSR MVCD
 CLI
 INC NEWFO
* PREP SW WORD
 LDA A #$FF
 STA A WPNS
RMN4 RTS
*
* READ THUMBWHEELS
* B=PASS COUNT
*
RTHW EQU *
* TEST PASS
 TST B
 BNE RTH3
 LDA A #7
 STA A PIA1DB
*
RTH3 CMP B #4
 BEQ RTH1
 CMP B #9
 BNE RTH2
*
RTH1 INC B
 CLR A
 RTS
*
RTH2 LDA A PIA1DA
 AND A #$0F
 INC B
 DEC PIA1DB
 RTS
*
* TEST RANGE OF NEW F.O.
* C=SET, NEW WITHIN +- 40 MILS
* C=CLR, OUTSIDE RANGE
*
TRANG EQU *
* AZFO = MINUEND
 LDX #AZFO
 STX S1
```



```
* CBUF=SUBTRAHEND
 LDX #CBUF+5
 STX S2
* ANSWER=RESULT
 LDX #RESULT
* SUBTRACT
 JSR BCDSUB
* FIX ROLLOVER
 LDX #TEMBCD
 JSR ADJ
* TEST RANGE OF RESULT +-400
 LDX #TEMBCD
 LDA A 0,X
 CMP A #6
 BEQ TRAG3
*
 TST 0,X
 BNE TRAG2
 TST 1,X
 BNE TRAG2
 LDA A 2,X
 CMP A #3
 BGT TRAG2
 SEC
 RTS
*
TRAG3 LDA A 1,X
 CMP A #3
 BLT TRAG2
 LDA A 2,X
 CMP A #6
 BLT TRAG2
 SEC
 RTS
*
TRAG2 CLC
 RTS
* INTERRUPT VECTORS
 ORG $FFF8
 FDB ISER
 FDB AGO
 FDB ISER
 FDB AGO
 END
```

END

DATE
FILMED
5 81

DTIC