63-3-2

TM-891/005/00

TECHNICAL MEMORANDUM

(TM Series)

ASTIA AVAILABILITY NOTICE

Qualified requesters may obtain copies of this report from ASTIA.

This document was produced by SDC in performance of contract AF 19(628)-1648, Space Systems Division Program, for Space Systems Division, AFSC.

Utility System Program Specifications

Master Tape Control II (MTCII), Mod AG

By

B. R. Pruett

1 April 1963

Approved

J. B. Munson

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA

This document is one of a series of TM-891 volumes established for Utility System program specifications.

Comments on this document must be received by 25 April 1963 to be reflected in the final design criteria. It is anticipated that this feature will be available in the AF/CPL on 1 May 1963 . The publication of a volume in the TM-705 (Systems Manual) series will officially announce the completion of this project.

Master Tape Control II (MTCII), Mod AG

The purpose of this document is to describe the new features which will be available in the next mod of the COPII control program, MTCII mod AG.

The anticipated release date for the checked out program and the documentation is 1 May 1963. The documentation for mod AG will consist of a supplement to TM-745 and a milestone 7 (operating procedures).

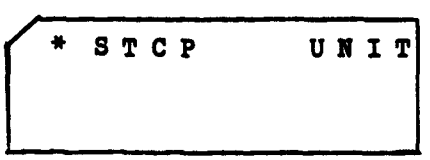
The features which warrant a new mod of MTCII are as follows:

A. Parameter Test Interface

The Parameter Test System consists of three separate programs:

1. Reference Pool Simulator (SRPS)
2. Test Control Program (STCP)
3. Data Reduction Program (SDRP)

The SRPS and SDRP programs are called via a normal function request card, but STCP must be called via the *STCP pseudo function. The format of this pseudo is:



where: STCP is the name of the parameter test pseudo, and

UNIT is the logical unit number on which the traces, recordings, etc. will be made by the Test Control Program (STCP).

Upon encountering the STCP pseudo, MTCII will store the logical unit number from the request into the cell called STCPUNIT which is then referenced symbolically by STCP. A flag, STCPFLAG, internal to MTCII, will be set so that the next function request to be processed will result in the loading of the Test Control Program and execution of it.

The deck set up for initializing the parameter test mode is:

```
* STCP  UNIT
*      FUNCTION
STCP  Control Cards
```

where FUNCTION is the element to be tested.

The restrictions imposed by MTCII on the functions that may utilize the parameter test system are:

1. The function to be tested and its environment must not exceed 60000B cells in length.

2. The ADDROF or the Successor Function (CALL) feature of MTCII can not be utilized either by the function to be tested or its environment since the Test Control Program occupies the area above 70000B and the MTCII code which processes the above features is not in core. The error message output by MTCII, if this restriction is violated, is one of the following:

   a. ADDROF REQUESTED DURING PTS MODE
   b. SUCCFUN REQUESTED DURING PTS MODE

B. Load and Go

The "Load and Go" feature provides a method of defining a program from the binary punch tape output by LARII or from a prestored binary tape. By utilizing this feature a program can be assembled and tested in one run. In addition, there will be no need for punching binary cards since the binary tape can be reserved.

The usage of this feature is actuated through the DEFINES card as illustrated below:

```
* CARDS
DEFINES  P1  P2  P3  P4  P5
Correctors if P5 = C (optional)
* P1
```

where: P_1 - name of function to be defined.

P_2 - (optional) is equal to "F" if the function is in absolute binary format; otherwise, P_2 is not present and the function is assumed to be relocatable.

P_3 - is the range in octal of a relocatable function or the octal starting location of an absolute function. P_3 is assumed to be octal. A "B" may follow P_3 since all other octal numbers are suffixed by B. Prior to mod AG of MTCII a B after P_3 was illegal.

P_4 - (optional) is the logical unit number on which the binary deck of the function is contained. If P_4 does not exist MTCII will assume the binary deck to be from the standard input source (cards or prestored function request tape).

P_5 - (optional) is equal to "C" if correctors are to be input from the standard input source (cards or prestored function request tape). The correctors must be terminated

by a term card. The card following the term card or the next card if P_5 does not exist, must be either another DEFINES, a CORRECT or a function request. P_5 is not interrogated if P_4 is missing.

If P_4 exists, MTCII will rewind the specified tape and then interrogate the third word of the first record of each file (IDENT record). This word of the IDENT card must be identical to the BCD name of the function (P_1) taken from the DEFINES card. If the function is not found the following message will be printed:

FUNCTION NOT ON LARII BINARY TAPE

If the function is found on the tape, it will be loaded. If correctors are specified by P_5, they will be read from the standard input source.

The "Load and Go" feature is not recommended for initial assemblies. An illustration of the usage of "Load and Go" is as follows:

*LARII	Assemble program. The binary output is on Unit 6.
*HALT READY UNIT 6	Instructions to the operator to ready the binary punch tape. (Optional)

*CARDS

DEFINES FUNCTION 7000 6 C

Correctors followed by Term Card

*FUNCTION

If there are no correctors for the function, "C" will not be on the DEFINES card.

C. BCD Sentence Count

A new type of field was added to free field correctors, free field data cards and function requests in mod AF of MTCII. This field provides for hollerith information which either begins with a number, blank or arithmetic symbol, has embedded blanks, or consists of more than eight characters. All information enclosed in parentheses is considered to be a BCD sentence.

Although a sentence is considered to be one parameter, it may consist of more than one word of hollerith information. A parameter type table is output by FFCONV when converting a free field data card and by MTCII when interpreting a function request. To keep the number of words in the parameter type table equivalent to the number of words of converted data, there is one word in the parameter type table for each word of a BCD sentence. To indicate the number of words in this type of field, a new entry has been added to the parameter type table, the BCD sentence count.

To illustrate this type of field, assume the following request:

* STCP (TEST∧CONTROL∧(STCP)) 1. 01

where ∧ indicates a space.

The information on this request will be interpreted as follows:

	Parameters	Parameter Type Table
Word 1	TEST∧CON	000 00003 000 00004
2	TROL∧(ST	000 00000 000 00004
3	CP)∧∧∧∧∧	000 00000 000 00004
4	2001400000000000	000 00000 000 00001
5	0000000000000001	000 00000 000 00002

The upper address of the first word of the parameter type table gives the number of words in the sentence (3). The lower address of each word is the parameter type code which is 4 for a BCD sentence, 3 for an octal number, 2 for fixed point, 1 for floating point and 0 for BCD.

A closed parenthesis,), followed by a blank terminates the BCD sentence; therefore, parentheses may be enclosed in parentheses provided a blank does not follow a). The enclosed parentheses will be converted to their Hollerith codes.

D. MTCII Correctors

Previously correctors for MTCII were stored dynamically but were not stored in the corrector table. Upon a re-load of MTCII, the correctors were not preserved. In mod AG, MTCII correctors are stored in the corrector table as well as being stored dynamically. Whenever MTCII is re-loaded the corrector table is interrogated for any correctors that are necessary and they are stored at this time.

E. MTCII Mod Number

Whenever auto-load is depressed or MTCII is re-loaded a message indicative of the mod of MTCII will be output on the printer and the system output unit. This message will serve to identify the mod of the control program being used and will be useful in trouble shooting when numerous versions of it are being used.

The messages are:

MTCII XX HAS BEEN RELOADED

or MTCII XX HAS BEEN AUTO LOADED

where XX is the mod number of the program.

F. Intercommunication

The definition of elements has been expanded to allow intercommunication between elements whose names appear in the CODES table but do not exist in the directory. Previously the message, "ROUTINE REQUESTED WHICH IS NOT IN THE DIRECTORY", was output when elements in the CODES table tried to communicate when one or both of them did not appear in the directory.

In mod AG the CODES table will be loaded whenever elements are defined. In this way this table will be searched prior to searching the directory. Additional information will be kept in EQUIVS for the defined element, i.e., its CODES number.

G. Typewriter I/O Routine

A new typewriter I/O routine will be incorporated in MTCII, AG. This routine will use the communication informational words, IOSTATUS, EXITSTAT, and IOBUFWD as described in FN-6950, Proposal for New I/O Routines in MTCII.

1. Input

The input routine for the typewriter will either convert the typewriter code to hollerith and pack the characters 8 per word, pack the typewriter code 8 per word with the upper and lower cases inserted or output the typewriter code one character per word with upper and lower cases inserted depending upon the setting of IOSTATUS. The typewriter routine will automatically carriage return and shift to lower case. A carriage return or number of words specified will terminate input. If an error is made in typing, a colon or semi-colon will result in the clearing of the output image, a carriage return and a shift to lower case. The input image will be filled up with blanks or spaces if less words are input than are requested.

2. Output

The typewriter will be carriage returned and shifted to lower case. Depending upon the IOSTATUS word the data will be either converted from BCD code to typewriter code, typed in assembly mode or character mode. Illegal BCD characters will be replaced by the typewriter code for colon or semi-colon depending upon the case.

H. Interrupt

The 160A interrupt will be processed in a slightly different fashion. Whenever MTCII cannot be interrupted, the 1607 communication flag 1 will be set. The 160A will be modified so that it will not initiate an interrupt when the flag is set. MTCII will clear flag 1 and delay at least 4 seconds whenever interrupt is feasible. Interrupt will be permitted only between the execution of two functions. When interrupted by the 160A, MTCII will determine whether SBRDTLK is already in core. If it is not, it is loaded. The message, "THE TRANSMIT MODE IS BEGINNING", will be typed and then SBRDTLK will be operated. After its execution MTCII will type "THE TRANSMIT MODE IS TERMINATED". 1607 communication flag 1 is still used to acknowledge the 160A interrupt. All previous messages and usage of jump key two will be deleted.

I. Timing Routine

The purpose of the timing routine will be to time a portion of a function, an entire function, a function including the loading time, or a group of functions. The timing element, which will be an integral part of MTCII, may be called either by a function request or by utilizing it as a subroutine.

1. Function Request Usage

The format of the timing pseudo function will be:

* CLOK P_1 P_2 P_3

where: P_1 = ON clears the clock counter and activates the clock depending upon the P_3 parameter.

P_1 = OFF stops the clock and results in the output of the total elapsed time for the period when the clock was activated until the CLOK OFF request was encountered.

P_2 = (optional) the unit number of the output of elapsed times. If no unit is specified, output will be on the 1612 printer.

P_3 = (optional) If P_3 is omitted the elapsed time for the ON-OFF period will be output.

P_3 = F indicates that the elapsed time for each function will be output in addition to the total elapsed time. The time for the function will be its execution time.

= L indicates that the elapsed time for each function including the time to load and process the function will be output in addition to the total elapsed time.

2. CLOK Subroutine Usage

The usage of CLOK as a subroutine will permit the timing of a portion of a function. The calling sequence will be:

```
B     RTJ    CLOK
B+1   P1
B+2   P2
B+3   Normal Return
```

where: P_1 = ON or OFF as explained in the function request usage of CLOK

P_2 = logical unit number of the output of elapsed time

CLOK is a symbol in the RST table; therefore, it may be referenced.

3. Output

There are four types of output: Elapsed time of individual functions, total elapsed time for a sequence, time for a function including load time and the time required for a portion of a function. The output will be on the specified unit:

2-12 and 16-19 - magnetic tape units
13 - printer
14 - typewriter

The messages will be:

a. E. T. FOR FUNCTION = HR:MIN::SEC.SEC/100

b. E. T. FOR FUNCTION INCL. L. T. = HR MIN SEC.SEC/100

c. TOTAL E. T. FOR SEQUENCE = HR MIN SEC.SEC/100

d. E. T. FOR PART OF FUNCTION = HR MIN SEC.SEC/100

where: E. T. - elapsed time
L. T. - is time required to process function request and load the elements from the Master Tape.

Messages c or d will be output when CLOK OFF is encountered. Messages a and b will be output after the execution of each individual function, if so specified.

4. Restrictions

a. Functions presently using the clock for timing or interrupt purposes should be modified to eliminate the usage of the clock. There are two reasons for this request; one is that it limits the usage of the timing pseudo and two, the inactivity feature of the Parameter Test System can not be utilized.

b. If CLOK ON is encountered when the clock has already been activated, the previous selection will be discontinued and the clock counter will be cleared.

J. Usage of Clank

The usage of CLANK will be permitted as an N_3 parameter on a function request, and as a parameter on a CLR or CARDS card.

1. Function Request

When CLANK is encountered as an N_3 parameter on a function request, the system will not be initialized, the last function processed will not be erased from core, the Special Operating Mode will not be terminated and the environment of the next function will start at the location specified by CLANK (current location counter). An example of the usage of CLANK on a function request is as follows:

```
* CLANK DUMP 3 10000B 20000B
```

This request will enable the dumping of core without initializing core or wiping out the area of interest. This capability will be useful when operating in the Special Operating Mode.

2. CLR

CLANK can be used to specify the starting location of the area to be cleared. Everything previously in core will be retained and the area

from CLANK to the end of usable core (70000B in case of CLR) will be set to zeros. The Special Operating Mode will not be terminated. An example is:

```
* CLR    C    CLANK    65000B
```

3. CARDS

When CLANK is input on the CARDS pseudo request, the first defined function will be loaded following the last element in core. The system will not be initialized in this case; however, it is the user's responsibility to save enough room in core for the defined element, otherwise an error halt will occur (defined elements can not go above 70000B in core). The following illustrates the use of CLANK:

```
* CARDS  CLANK
DEFINES  A  10000
Deck of A
* A
```

If already in the Special Operating Mode, the mode will not be terminated.

K. Length of Redefined Element

Whenever a redefined element is loaded, its new length from the DEFINES card will replace its length in the directory. As long as the element remains in core, its new length will be retained. This will result in the correct length being used by SYMDUMP since it used tables built by MTCII to obtain an element's length and starting location.

Whenever core is effectively cleared (that is, the tables acknowledging the contents of core are cleared), the directory will be re-loaded, so that if the element is requested from tape, the correct length will be used.

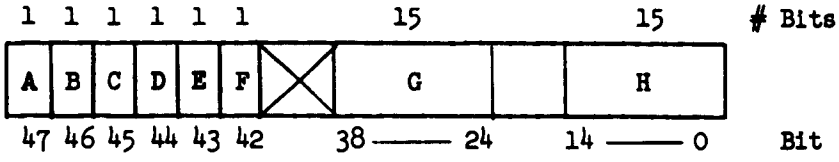
L. Successor Call

To provide for interface between SCHOPS and MTCII, the successor call capability has been expanded to allow pseudo functions to be requested, to offer the option of not initializing the core allocation tables, and to provide a "Do not Operate" option.

The expanded successor call feature has the following calling sequence:

Step 1 - Enter the Accumulator with the BCI name for the desired function.

Step 2 - Enter Q with the control information. The format of Q is:



where: A (bit 47) = 0 Return control to MTCII
= 1 Return control to object program

B (bit 46) = 0 No N_1 parameter
= 1 N_1 is present in calling sequence. N_1 is the storage analysis unit.

C (bit 45) = 0 No N_2 parameter
= 1 N_2 is present. N_2 is the number of arithmetic interrupt messages to output. If N_2 is zero, interrupt on arithmetic fault will not be selected.

D (bit 44) = 0 No N_3 parameter
= 1 N_3 is present and is the starting location desired for the function requested.

E (bit 43) = 0 No N_4 parameter
= 1 N_4 is present and is equal to "X" to indicate to load the requested function but do not operate it.

F (bit 42) = 0 Initialize core allocation tables
= 1 Do not initialize

G (bits 24-38) Number of parameters in calling sequence.

H (bits 0-14) Number of mods in calling sequence.

Step 3 - Execute the following calling sequence:

```
RTJ   CALL
P1
P2
 .
 .
 .
Pp
M1
M2
 .
 .
 .
MM
N1        Only those control parameters specified
N2        in Q can follow the call.  If A is set
N3        to 1, MTCII will return control to the
N4        next location following the calling se-
          quence (normal return).
Normal Return
```

All programs currently using successor call are not affected unless the operation code of the upper instruction contains 1 bits instead of zero.

To call pseudos via successor call, the parameters for the pseudo will have to be included in the call for it.

DISTRIBUTION LIST

External

Space Systems Division
(Contracting Agency)
Major C. R. Bond (SSOCD)

6594th Aerospace Test Wing
(Contracting Agency)
Col. A. W. Dill (TWRD)
Lt. Col. M. S. McDowell (TWRU) (4)
TWACS
V. Thomas

PIR-E1 (Lockheed)
N. N. Epstein
C. H. Finnie
H. F. Grover
H. R. Miller
W. E. Moorman (5)
461 Program Office
698BK Program Office

PIR-E2 (Philco)
J. A. Bean
J. A. Isaacs
R. Morrison
S. M. Stanley

PIR-E3 (LFE)
D. F. Criley
K. B. Williams (5)

PIR-E3 (Mellonics)
F. Druding

PIR-E5 (Aerospace)
F. M. Adair
R. V. Bigelow
R. D. Brandsberg
L. H. Garcia
G. J. Hansen
C. S. Hoff
L. J. Kreisberg
T. R. Parkin
E. E. Retzlaff
H. M. Reynolds
D. Saadeh
R. G. Stephenson
V. White

PIR-E7 (STL)
A. J. Carlson (3)

PIR-E4 (GE-Sunnyvale)
J. Farrentine
N. Kirby

PIR-E4 (GE-Santa Clara)
D. Alexander

PIR-E4 (GE - Box 8555)
J. S. Brainard
R. J. Katucki
J. D. Selby

PIR-E4 (GE-3198 Chestnut)
J. F. Butler
H. D. Gilman

PIR-E4 (GE-Bethesda)
W. L. Massey

PIR-E4 (GE-Box 8661)
J. D. Rogers

VORHAUS, A. H.	24076A	WINSOR, M. E.	22156
WAGNER, I. I.	24093	WINTER, J. F.	24117
WARSHAWSKY, S. B.	24097	WISE, R. C.	22085
WEST, G. D.	SUNNYVALE	WONG, J. P.	SUNNYVALE
WEST, G. P.	22116A	ZUBRIS, C. J.	24075
WILSON, G. D.	24124		
BUSCH, R. E.	22088B		

1 April 1963

DISTRIBUTION LIST

Internal

TM-891/005/00

AFCPL	(5)	KEY, C. D.	23013
ALLFREE, D.	24083	KEYES, R. A.	24073
ALPERIN, N. I.	22153	KINKEAD, R. L.	22093
ARMSTRONG, E.	24123	KNEEMEYER, J. A.	22088A
BERNARDS, R. M.	SUNNYVALE	KNIGHT, R. D.	22119
BIGGAR, D.	24118A	KOLBO, L. A.	22155
BILEK, R. W.	23007	KOSTINER, M.	14056B
BLACK, H.	14039	KRALIAN, R. P.	14039
BRENTON, L. R.	24103B	KRISTENSEN, K.	SUNNYVALE
BURKE, B. F.	24086	LACHAPELLE, F.	22093
CHAMPAIGN, M. E.	22152	LAUGHLIN, J. L.	24073
CHIODINI, C. M.	24091	LAVINE, J.	24093
CIACCIA, B. G.	24082A	LITTLE, J. L.	24088B
CLINE, B. J.	24127	LONG, F.	22156
COGLEY, J. L.	22156	MADRID, G. A.	22081
CONGER, L.	24088A	MAHON, G. A.	24089
COOLEY, P. R.	24081	MARIONI, J. D.	24076B
COURT, T. D.	24086B	MARTIN, W. P.	24127B
CRUM, D. W.	24105	MCKEOWN, J.	23013
DANT, G. B.	24086B	MICHAELSON, S. A.	14039
DECUIR, L. E.	24053A	MILANESE, J. J.	22155
DERANGO, W. C.	24082B	MUNSON, J. B.	22087A
DEXTER, G. W.	25016	MYERS, G. L.	14056A
DISSE, R. J.	23014	NELSON, P. A.	24075
DOBBS, G. H.	22116B	NG, J.	22077
DOBRUSKY, W. B.	24065A	NGOU, L.	24127
ELLIS, R. C.	22131A	PADGETT, L. A.	24110A
EMIGH, G. A.	14039	PATIN, O. E.	SUNNYVALE
ERICKSEN, S. R.	22113	POLK, T. W.	24113
FELKINS, J.	24097	PRUETT, B. R.	22084
FOSTER, G. A.	14039	RAYBIN, M.	14039
FRANKS, M. A.	24122	REILLY, D. F.	24121
FREY, C. R.	22078	REMSTAD, C. L.	25026
FRIEDEN, H. J.	22082	RUSSELL, R. S.	14054
GARDNER, S. A.	25026	SCHOLZ, J. W.	14039
GREENWALD, I. D.	22094A	SCOTT, R. J.	24110
GRIFFITH, E. L.	22081	SEACAT, C. M.	SUNNYVALE
HAAKE, J. W.	22153	SEIDEN, H. R.	22126B
HARRIS, E. D.	24081	SHAPIRO, R. S.	24110B
HENLEY, D. E.	22094B	SKELTON, R. H.	22148
HILL, C. L.	22101	SOLOMON, J.	22076
HILLHOUSE, J.	22078	SPEER, N. J.	24086A
HOLMES, M. A.	24103	STONE, E. S.	24058B
HOLZMAN, H. J.	24065B	SWEENEY, M. J.	25026
HOUGHTON, W. H.	24103B	TABER, W. E.	22101
HOYT, R. L.	14039	TENNANT, T. C.	27029
IMEL, L. E.	14039	TESTERMAN, W. D.	14039
KASTAMA, P. T.	22076	THOMPSON, J. W.	24088
KAYSER, F. M.	24109	THORNTON, R. L.	14050
KEDDY, J. R.	24105	TOTSCHEK, R. A.	24120

UNCLASSIFIED

System Development Corporation,
Santa Monica, California
UTILITY SYSTEM PROGRAM
SPECIFICATIONS MASTER TAPE
CONTROL II (MTCII), MOD AG.
Scientific rept., TM-891/005/00,
by B. R. Pruett. 1 April 1963, 14p.
(Contract AF 19(628)-1648, Space
Systems Division Program, for Space
Systems Division, AFSC)
Unclassified report

DESCRIPTORS: Programming (Computers).
Satellite Networks.

UNCLASSIFIED

UNCLASSIFIED

Describes the new features available
in the next modification of the
COPII control program, MTCII
(Master Tape Control II), mod AG.

UNCLASSIFIED