

# DM11

DATA TESTS  
MD-11-DZDMB-C

EP-DZDMB-C-DL  
COPYRIGHT ©72-77  
FICHE 1 OF 1

JAN 1978  
**digital**  
MADE IN USA



REM %

IDENTIFICATION  
-----

PRODUCT CODE    MAINCEC-11-DZDMB-C-D  
PRODUCT NAME    DM11 DATA TESTS  
DATE RELEASED    NOVEMBER, 1977  
MAINTAINER       DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1972-1977 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDF	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1 ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR TESTING THE DM11 (ASYNCHRONOUS DATA MULTIPLEXER), MAINDEC-11-DZDMA (DM11 LOGIC TESTS), AND MAINDEC-11-DZDMB (DM11 MULTIPLE LINE DATA TESTS). THE LOGIC TESTS INDIVIDUALLY TEST EACH OF THE 16 DM11 LINES AND ALL COMMON LOGIC. THE MULTIPLE LINE DATA TESTS RUN SEVERAL LINES CONCURRENTLY AND ARE USED TO TEST LINE INTERACTION AND DATA TRANSMISSION/RECEPTION RELIABILITY. THIS DOCUMENT DESCRIBES THE MULTIPLE LINE DATA TESTS. THE AVAILABLE TESTS ARE.

PRG0 - DATA TESTS  
PRG1 - DATA TEST (ALL LINES SIMULTANEOUSLY)  
PRG2 - TRANSMIT TO TERMINALS  
PRG3 - ECHO RECEIVED DATA

2 REQUIREMENTS

2.1 EQUIPMENT

A PDP 11 FAMILY PROCESSOR  
B DM11  
C JUMPERS CONNECTING 16 TRANSMITTERS TO THEIR RESPECTIVE RECEIVERS  
D TERMINALS (IF AVAILABLE)  
E DM11 DISTRIBUTION PANEL

2.2 STORAGE

THIS PROGRAM USES ALL OF COPE (4K) EXCEPT THAT AREA RESERVED FOR THE LOADERS.

2.3 PREREQUISITE PROGRAMS  
MAINDEC-11-DZDMA (DM11 LOGIC TESTS)

3 LOADING PROCEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.

- 4 USE PROCEDURE
- 4.1 STARTING PROCEDURE

BEFORE STARTING MAKE SURE THAT THE TTY IS IN REMOTE MODE  
THREE STARTING ADDRESSES WRE PROVIDED

0200 - THIS STARTING ADDRESS REQUESTS DM11 PARAMETERS, AND MUST  
BE USED TO INITIALLY START THE PROGRAM, AND WHENEVER ANY  
OF THE PARAMETERS LISTED BELOW IS CHANGED

A VECTOR ADDRESS ?  
RESPONSE TYPE IN THE VECTOR ADDRESS OF THE DM11 RECEIVER  
UNDER TEST CARRIAGE RETURN SELECTS 0300

B UNIT #(8)?  
RESPONSE THE DM11 UNIT NUMBER CORRESPONDS TO THE  
ADDRESS TO WHICH THE CLOCK STATUS REGISTER (CSR) RESPONDS

CSR ADDRESS	DM11 UNIT #	CSR ADDRESS	DM11 UNIT #
175000	0	175100	10
175010	1	175110	11
175020	2	175120	12
175030	3	175130	13
175040	4	175140	14
175050	5	175150	15
175060	6	175160	16
175070	7	175170	17

CARRIAGE RETURN SELECTS UNIT # 0

C PRG #  
RESPONSE TYPE PROGRAM NUMBER OF PROGRAM YOU WISH TO  
RUN CARRIAGE RETURN SELECTS PROGRAM # 0.

CARRIAGE RETURN TERMINATES ALL RESPONSES  
ANY UNACCEPTABLE RESPONSE WILL RESULT IN A ? TYPEOUT AND  
THE PARAMETER WILL AGAIN BE REQUESTED.

0204 - THIS STARTING ADDRESS USES PREVIOUSLY DEFINED DM11  
PARAMETERS AND REQUESTS THE PROGRAM NUMBER OF THE  
PROGRAM YOU WISH TO RUN

0210 - THIS STARTING ADDRESS STARTS THE PREVIOUSLY SELECTED  
PROGRAM USING PREVIOUSLY SELECTED PPARAMETERS

4 2 SWITCH SETTINGS

THE FOLLOWING SWITCH SETTINGS APPLY TO PROGRAM #0

SR 0-6	ROUTINE TO BE RUN (IF ENABLED BY SR-9)
SR 9	LOOP SELECTED ROUTINE
SR 11	INHIBIT ITERATION (DO EACH POUT NE ONCE)
SR 13	INHIBIT PRINTOUT
SR 14	SCOPE (LOOP ROUTINE)
SR 15	HALT ON ERROR

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC 176 ) IS DEFAULTED TO IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE

1	<CR>	IF NO CHANGES ARE TO BE MADE
2	6 DIGITS 0-7	TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ,LAST DIGIT FOLLOWED BY <CR>
3	U	TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALLE

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION BY STRIKING G (CNTRL G ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR POUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS

5 0 PROGRAM DESCRIPTIONS

5 1 PRGO - LOGIC TESTS

PRGO CONSISTS OF 34(8) INDEPENDENT ROUTINES WHICH TRANSMIT VARIOUS DATA PATTERNS ON ALL LINES WITH A DECREASING DELAY BEFORE STARTING SUCCESSIVE LINES THE DATA IS CHECKED WHEN ALL TRANSMITTERS HAVE COMPLETED TRANSMITTING IF A DATA ERROR OCCURS THE ERROR TYPEOUT WILL SHOW THE DATA FAILURE AND THE LINE NUMBER

5 2 PRG1 - DATA TEST (ALL LINES SIMULTANEOUSLY)  
PROGRAM 1 TRANSMITS ' A QUICK BROWN FOX JUMPED OVER THE LAZY DOGS  
BACK 1234567890' ON ALL LINES SIMULTANEOUSLY WHEN ALL LINES HAVE  
FINISHED RECEIVED DATA IS VERIFIED AN ERROR TYPEOUT IS AS IN PRGO

5 3 PRG2 - TRANSMIT TO TERMINALS  
PROGRAM 2 IS THE SAME AS PROGRAM 1 EXCEPT THAT THE RECEIVED  
DATA IS NOT CHECKED

5 4 PRG3 - ECHO RECEIVED DATA  
NOTE THIS PROGRAM MAY ONLY BE RUN IF USING AN  
ASR 33 NOT MODIFIED BY DEC  
PROGRAM 3 ECHOES BACK DATA RECEIVED FROM A TERMINAL  
NOTE PROGRAM 3 SHOULD BE RUN AND DATA TYPED AT ALL AVAILABLE  
DM11 TERMINALS IT IS THE ONLY TEST THAT INSURES CORRECT OPERATION  
OF THE DM11 DISTRIBUTION PANEL LOGIC IF THE TERMINALS ARE ASR-33  
WITH A PAPER TAPE READER/PUNCH I SUGGEST THAT INDIVIDUAL TAPES BE  
MADE UP FOR EACH LINE THIS CAN BE DONE BY RUNNING PROGRAM 2 WITH  
THE PUNCH TURNED ON PROGRAM 2 WILL THEN PUNCH A TAPE ON EACH TERMINAL  
WITH THE LINE NUMBER IDENTIFIER AT THE BEGINNING OF EACH TAPE  
PROGRAM 3 CAN BE RUN WITH THESE TAPES IN THE PAPER TAPE READERS

%

TITLE MAINDEC-11-DZDMB DM11 DATA TESTS  
NLIST MC,MD  
L ST ME  
ENABLE ABS,AMA

,DM11 DATA TESTS DIAGNOSTIC MAINDEC-11-DZDMB-B (FORMERLY D9GB)  
,COPYRIGHT 1972, 1977 DIGITAL EQUIPMENT CORP., MAYNARD, MASS 01754  
,PRGO- DATA TESTS  
,PRG1- DATA TEST (ALL LINES SIMULTANEOUSLY)  
,PRG2- TRANSMIT TO TERMINALS  
,PRG3- ECHO RECEIVED DATA

,STANDARD SR SWITCH OPTIONS SWITCH SET TO A 1 (UP)  
,SR15- HALT ON ERROR  
,SR14- SCOPE  
,SR13- INHIBIT PRINTOUT  
,SR12- INHIBIT TRACE  
,SR11- INHIBIT ITERATION  
,SR9- LOOP ROUTINE  
,SR6 THROUGH SR0 - NUMBER OF ROUTINE TO BE LOOPEC

EQUATE STATEMENTS

177776  
177776  
000004  
000240  
000000  
100000  
100000  
040000  
020000  
010000  
004000  
002000  
001000  
000400  
000200  
000100  
000040  
000020  
000010  
000004  
000002  
000001  
005726  
022626  
000340  
000300  
000240  
000200  
000140  
000100  
000040

CC=177776  
PSW=177776  
ERRVEC=4  
NOP=240  
OPEN=0  
MANUAL=BIT15  
BIT15=100000  
BIT14=40000  
BIT13=20000  
BIT12=10000  
BIT11=4000  
BIT10=2000  
BIT9=1000  
BIT8=400  
BIT7=200  
BIT6=100  
BIT5=40  
BIT4=20  
BIT3=10  
BIT2=4  
BIT1=2  
BIT0=1  
POPSP=5726  
POPSP2=022626  
PRTY7=340  
PRTY6=300  
PRTY5=240  
PRTY4=200  
PRTY3=140  
PRTY2=100  
PRTY1=40

,POP THE STACK SAME AS TST (6)+  
,POP STACK TWICE SAME AS CMP (6)+, (6)+  
,PRIORITY LEVEL DEFINITIONS

000000	000000	PRTY0=0	
000001	000000	RO=%0	
000002	000001	R1=%1	
000003	000002	R2=%2	
000004	000003	R3=%3	
000005	000004	R4=%4	
000006	000005	R5=%5	
000007	000006	SP=%6	
	000007	PC=%7	
		EMT CALLS	
104000		TYPE=EMT+0	
104001		TYPES=EMT+1	
104002		STALL=EMT+2	
104003		ERROR=EMT+3	
104004		DATCHK=EMT+4	
104006		STRXV=EMT+6	
104007		STTXV=EMT+7	
104010		EHALT=EMT+10	
104012		SCOPE=EMT+12	
104013		SAVREG=EMT+13	
104014		RSTREG=EMT+14	
104015		ERROR1=EMT+15	
104016		SUSWR=EMT+16	
104017		KBDIN=EMT+17	
104020		CNTLU=EMT+20	
	000007	BELL=007	
	177777	ATLAST=-1	
	125252	ALTO=125252	.ALTERNATING 0'S PATTERN
	052525	ALT1=052525	.ALTERNATING 1'S PATTERN
	000000	Y=0	
	177777	X=-1	
	000000	A=0	
	000000	=0	
000000	000000	HALT	
000002	000000	HALT	
000004	000006	+2	.SP OVERFLOW, BUS ERROR TRAP
000006	000000	HALT	
000010	000012	+2	.RESERVED INSTRUCTION TRAP
000012	000000	HALT	
000014	000016	+2	.TRACE TRAP
000016	000000	HALT	
000020	000022	+2	.TRAP TO CALL 104
000022	000000	HALT	
000024	000026	+2	.POWER FAIL TRAP
000026	000000	HALT	
000030	002572	EMT INT	.EMT TRAP
000032	000340	PRTY?	
000034	000036	+2	
000036	000000	HALT	
000040	000042	+2	.TRAPPED TO PREVIOUS ADDRESS
000042	000000	HALT	
000044	000046	+2	.TRAPPED TO PREVIOUS ADDRESS
000046	000000	HALT	
000050	000052	+2	.TRAPPED TO PREVIOUS ADDRESS
000052	000000	HALT	



000054	000056	+2	
000056	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000060	000062	+2	
000062	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000064	000066	+2	
000066	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000070	000072	+2	
000072	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000074	000076	+2	
000076	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000100	000102	+2	
000102	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000104	000106	+2	
000106	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000110	000112	+2	
000112	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000114	000116	+2	
000116	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000120	000122	+2	
000122	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000124	000126	+2	
000126	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000130	000132	+2	
000132	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000134	000136	+2	
000136	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000140	000142	+2	
000142	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000144	000146	+2	
000146	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000150	000152	+2	
000152	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000154	000156	+2	
000156	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000160	000162	+2	
000162	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000164	000166	+2	
000166	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000170	000172	+2	
000172	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000174	000176	+2	
000176	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000200	000202	+2	
000202	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000204	000206	+2	
000206	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000210	000212	+2	
000212	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000214	000216	+2	
000216	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000220	000222	+2	
000222	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000224	000226	+2	
000226	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000230	000232	+2	
000232	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS

000234	000236	+2	
000236	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000240	000242	+2	
000242	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000244	000246	+2	
000246	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000250	000252	+2	
000252	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000254	000256	+2	
000256	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000260	000262	+2	
000262	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000264	000266	+2	
000266	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000270	000272	+2	
000272	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000274	000276	+2	
000276	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000300	000302	+2	
000302	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000304	000306	+2	
000306	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000310	000312	+2	
000312	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000314	000316	+2	
000316	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000320	000322	+2	
000322	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000324	000326	+2	
000326	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000330	000332	+2	
000332	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000334	000336	+2	
000336	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000340	000342	+2	
000342	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000344	000346	+2	
000346	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000350	000352	+2	
000352	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000354	000356	+2	
000356	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000360	000362	+2	
000362	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000364	000366	+2	
000366	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000370	000372	+2	
000372	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS
000374	000376	+2	
000376	000000	HALT	, TRAPPED TO PREVIOUS ADDRESS

000046	000046		=46		.ACT:1 HOOKS
000046	002530		SENDAD		
000052	000052		=52		
000052	020000		020000		
000174	000174		=174		
000174	000000	DISPREG	0		
000176	000000	SWREG	0		
000200	000200		=200		
000204	000137	002112	JMP	@#START	.GO TO START OF DIAGNOSTIC
000204	000137	002130	JMP	@#RSTAT1	.GO GET PROGRAM # & RESTART PROGRAM
000210	000137	002200	JMP	@#PSTAT2	.USING PREVIOUS DM11 PARAMETERS
					.RESTART PREVIOUS PROGRAM USING
					.PREVIOUS DM11 PARAMETERS
	001100		=1100		
001100	000000	SPBOT	0		
001102	177570	SWR	177570		
001104	177570	DISPLAY	177570		
001106	000000	CAT	OPEN		.STARTING ADDRESS OF
	001146		=CAT+32		.CURRENT ADDRESS TABLE
001146	000000	WCT	OPEN		.STARTING ADDRESS OF
	001206		=WCT+32		.WORD COUNT TABLE
001206	000000	BAT	OPEN		.STARTING ADDRESS OF
	001246		=BAT+32		.BIT ASSEMBLY TABLE
001246	000000	VAC	OPEN		.32 SPARE WORDS
001250	175000	CSR	175000		.ADDRESS OF CLOCK STATUS REGISTER
001252	175002	BAR	175002		.ADDRESS OF BUFFER ACTIVE REGISTER
001254	175004	BKCSR	175004		.ADDRESS OF BREAK STATUS REGISTER
001256	175006	BASREG	175006		.ADDRESS OF BASE REGISTER
001260	000000	CLKINT	OPEN		.DM11 VECTOR ADDRESS (RECEIVER)
001262	000240	CLKLVL	PRTY5		.PRIORITY LEVEL
001264	000000	XMTINT	OPEN		.DM11 VECTOR ADDRESS (TRANSMITTER)
001266	000240	XMTLVL	PRTY5		.TRANSMITTER PRIORITY LEVEL
001270	000000	BARIM	OPEN		.PROGRAM BAR IMAGE
001272	000000	TTDAT	OPEN		.TUMBLE TABLE DATA
001274	000000	LINBIT	OPEN		.LINE BIT (FOR BAR)
001276	000000	BARDAT	OPEN		.BAR DATA
001300	000000	TTPTR	OPEN		.PROGRAM TUMBLE TABLE POINTER
	001306		=VAC+32		
001306	000000	TUMTAB	OPEN		.STARTING ADDRESS OF
	001506		=TUMTAB+100		.TUMBLE TABLE
001506	000060	TKVTR	60		.LSR INTERRUPT VECTOR
001510	000200	TKLVL	PRTY4		.LSR PRIORITY LEVEL
001512	000064	TPVTR	64		.LSP INTERRUPT VECTOR
001514	000200	TPLVL	PRTY4		.LSP PRIORITY LEVEL
001516	000000	KSTART	OPEN		.CURRENT PROGRAM START ADDRESS
001520	000000	CURTST	OPEN		.CONTAINS ADDR OF CURRENT TEST
001522	000000	RTNNO	OPEN		.CONTAINS CURRENT TEST #
001524	000000	NXTST	OPEN		.CONTAINS ADDR OF NEXT TEST
001526	000000	ICTR	OPEN		.CONTAINS CURRENT ITERATION COUNT
001530	000000	SCOPTR	OPEN		.CONTAINS CURRENT SCOPE PCINTER
001532	177774	PRGLIM	-4		

001534 005366  
001536 006346  
001540 006532  
001542 006542  
001544 005410  
001546 006364  
001550 006536  
001552 006560  
001554 002726  
001556 000000  
001560 000000  
001562 001732  
001564 001664  
001566 000000  
001570 000000  
001572 000000  
001574 000000  
001576 000000  
001600 002356  
001602 002624  
001604 002664  
001606 001750  
001610 004120  
001612 003764  
001614 004040

001616 000000  
001620 177560  
001622 177562  
001624 177564  
001626 177566  
001630 000000  
001632 000000  
001634 000000  
001636 000000  
001640 000000  
001642 000000  
001644 000000  
001646 000000  
001650 000000  
001652 000000  
001654 000000

PRGTAB PRGO  
PRG1  
PRG2  
PRG3  
RSTART PRGOR  
PRG1R  
PRG2R  
PRG3R  
EMTTAB TYP  
OPEN  
OPEN  
ERR  
DTCHK  
OPEN  
OPEN  
OPEN  
OPEN  
OPEN  
OPEN  
ESCOPE  
SAVRG  
RSTRG  
ERR1  
SUSWRR  
KBDINTT  
CNTLUU

SRT OPEN  
TKCSR 177560  
TKDBR 177562  
TPCSR 177564  
TPDBR 177566  
RCV DAT OPEN  
XMT DAT OPEN  
CARMSK OPEN  
TEMP OPEN  
PCADD OPEN  
APCADD OPEN  
PRVCNT- OPEN  
LINE OPEN  
L NBUF OPEN  
PASS OPEN  
COUNT OPEN

, PRGO START ADDRESS  
, PRG1 START ADDRESS  
, PRG2 START ADDRESS  
, PRG3 START ADDRESS  
, PRGO RESTART ADDRESS  
, PRG1 " "  
, PRG2 " "  
, PRG3 " "  
, POINTER TO TYPEOUT ROUTINE  
, POINTER TO CHAINED MESSAGES ROUTINE  
, POINTER TO RANDOM STALL ROUTINE  
, POINTER TO ERROR ROUTINE



```

001656 104000      INCRTN TYPE
001660 013006      M1
001662 000207      RTS      %7      ,TYPE INCORRECT ROUTINE SELECTED
                                     ,EXIT.

, DATA CHECK ROUTINE
001664 123737 001630 001632 DTCHK  CMPB  RCVDAT,XMTDAT ,COMPARE EXPECTED AND RECEIVED
001672 001416      BEQ    15      ,CHARS BRANCH IF SAME
001674 004737 002064      JSR    7,CNVDAT ,CONVERT RCVDAT & XMTDAT TO ASCII
001700 032777 020000 177174 BIT    #BIT13,@SWP ,ERROR TYPEOUT DESIRED?
001706 001010      BNE    15      ,BRANCH IF NO TYPEOUT DESIRED
001710 004537 004620      JSR    5,@#OACNV ,CONVERT LINE
001714 001646      LINE
001716 012762      ALINE
001720 000002      2
001722 104015      ERROR1
001724 104000      TYPE
001726 012753      LINEM
001730 000002      15      RTI      ,TYPE LINE # AS PART
                                     ,OF ERROR MESSAGE
                                     ,EXIT

, ERROR SERVICE ROUTINE CALLED BY TRAP (HLT)
001732 012737 000402 002032 ERR    MOV    #402,ERRB ,MOV BR +6 TO ERRB
001740 013737 001640 001642      MOV    @#PCADD,@#APCADD ,GET PC WHERE ERROR OCCURRED
001746 000410      BR    ERRA
001750 012737 000240 002032 EPR1  MOV    #240,ERRB ,MOVE NOP TO ERRB
001756 013737 001640 001642      MOV    @#PCADD,@#APCADD ,GET PC WHERE ERROR OCCURRED
001764 004737 002064      JSR    7,@#CNVDAT ,CONVERT RCVDAT & XMT DAT TO ASCII
001770 104017      EPRA  KBDIN
001772 032777 020000 177102 BIT    #BIT13,@SWP ,CHECK FOR G
002000 001017      BNE    ERRC ,ERROR PRINTOUT DESIRED
002002 004537 004620      JSR    5,@#OACNV ,BRANCH IF NO PRINTOUT
002006 001642      APCADD
002010 013442      APC
002012 000006      6
002014 004537 004620      JSR    5,@#OACNV ,CONVERT
002020 001522      RTNNO ,DATA
002022 013432      ATNUMB ,TO
002024 000003      3 ,ASCII
002026 104000      TYPE ,FOR
002030 013427      EMO ,PRINTOUT
002032 000000      EPRB  OPEN ,TYPE ERROR
002034 104000      TYPE ,MESSAGE
002036 012706      ERDAT ,NOP IF ERROR1, BR +6 IF ERROR
002040 023737 000042 000046 ERRC  CMP    @#42,@#46 ,TYPE ANOTHER MESSAGE
002046 001403      BEQ    ERPHLT ,IF ERROR 1
002050 005777 177026      TST    @SWP ,ACT11?
002054 100001      BPL    ERREX ,BR IF YES
002056 000000      ERRHLT HALT ,HALT ON ERROR
002060 104017      ERPEX KBDIN ,GO TO EXIT IF NO HALT ON ERROR
002062 000002      RTI    ,HALT
                                     ,CHECK FOR G
                                     ,RETURN

SUBROUTINE TO CONVERT RCVDAT AND XMTDAT TO ASCII AND PLACE
IN MESSAGE
002064 004537 004620 CNVDAT JSR    5,OACNV
002070 001632      XMTDAT
002072 012726      AASB
  
```

002074 000006  
002076 004537 004620  
002102 001630  
002104 312743  
002106 000006  
002110 000207

6  
JSR 5.0ACNV  
RCV DAT  
AWAS  
6  
RTS 7 .EXIT



002412	001403			BEQ	SCOPEB			, BRANCH IF SCOPE SW NOT SET
002414	013716	001530		SCOPEA	MOV	SCOPTR, @%		, SET UP TO RETURN TO ROUTINE
002420	000002				RTI			, RETURN TO ROUTINE
002422	032777	004000	176452	SCOPEB	BIT	#BIT11, @SWR		, TEST INHIBIT ITERATION SWITCH
002430	001012				BNE	SCOPEC		, BRANCH IF INHIBIT ITERATION SW SET
002432	023737	000042	000046		CMP	@#42, @#46		; ACT11?
002440	001003				BNE	1\$		, BR IF NO
002442	005737	001652			TST	@#PASS		, 1ST PASS?
002446	001403				BEQ	SCOPEC		, BR IF YES
002450	005337	001526		1\$	DEC	ICTR		, DECREMENT ITERATION COUNT
002454	001357				BNE	SCOPEA		, BRANCH IF COUNT NOT 0
002456	022626			SCOPEC	POPSP2			, POP STACK TWICE
002460	032777	001000	176414	SCOPEB	BIT	#BIT9, @SWR		, CHECK SELECT ROUTINE SWITCH
002466	001267				BNE	GETRDY		, BRANCH IF SELECT RTN SW SET
002470	022737	177777	001524		CMP	#-1, NXTST		, LAST TEST?
002476	001266				BNE	GTRDYX		, BRANCH IF NOT LAST TEST
002500	005237	001652			INC	@#PASS		, END PASS
002504	104000				TYPE			, TYPE
002506	013011				M2			, 'END'
002510	013702	000042			MOV	@#42 %2		, CHECK DDP/ACT11 MONITOR HOOK
002514	001654				BEQ	GETFDY		
002516	000005				RESET			
002520	000240				NOP			
002522	000240				NOP			
002524	000240				NOP			
002526	000240				NOP			
002530	004712			SENCAL	JSP	7 (2)		RETURN TO DDP/ACT11 MONITOR
002532	000240				NOP			
002534	000240				NOP			
002536	000240				NOP			
002540	011705	001524		FCPWD	MOV	NXTST, %5		, ADDR OF NEXT ROUTINE TO P5
002544	012507	001522			MOV	(5)+, RTNNO		, GET NEXT ROUTINE NUMBER
002550	012537	001524			MOV	(5)+, NXTST		, GET ADDR OF NEXT 'NEXT' ROUTINE
002554	012537	001526			MOV	(5)+, ICTR		, GET ITERATION COUNT
002560	012537	001530			MOV	(5)+, SCOPTR		, GET SCOPE LOOP ENTRY POINTER
002564	010537	001520			MOV	%5, CURTST		, ADDR OF NOW CURRENT TEST TO CURTST
002570	000207				RTS	%7		, EXIT FORWD SUBROUTINE
EMT TRAP INTERPRETER								
002572	011646			EMTINT	MOV	(6), -(6)		, GET PC OF NEXT INSTRUCTION
002574	162716	000002			SUB	#2, (6)		, POINT SP TO PC OF EMT
002600	011637	001640			MOV	(6), PCADD		, GET PC OF EMT CALL
002604	011616	000000			MOV	@(6), (6)		, GET EMT CALL
002610	105066	000001			CLRB	1(6)		, STRIP EMT & SAVE IDENTIFIER
002614	006316				ASL	(6)		, SHIFT IDENTIFIER LEFT
002616	062716	001554			ADD	#EMTTAB, (6)		
002622	013607				MOV	@(6)+, %7		GO TO PROPER EMT
. SAVE REGS 0 TO 4 SUBROUTINE								
002624	012637	002660		SAVRG	MOV	(6)+, 1\$		, SAVE PC AND PSW
002630	012637	002662			MOV	(6)+, 2\$		
002634	010446				MOV	%4, -(6)		, SAVE REGS 0 - 4
002636	010346				MOV	%3, -(6)		, IN STACK
002640	010246				MOV	%2, -(6)		
002642	010146				MOV	%1, -(6)		



002644	010046			MOV	%0, -(6)		
002646	013746	002662		MOV	2\$, -(6)	, RESTORE PC AND PSW	
002652	013746	002660		MOV	1\$, -(6)		
002656	000002			RTI		, EXIT	
002660	000000		1\$	OPEN		, CONTAINS SAVED PC	
002662	000000		2\$	OPEN		, CONTAINS SAVED PSW	
, RESTORE REGS 0 TO 4 SUBROUTINE							
002664	000240			RSTRG	NOP		
002666	012637	002722		MOV	(6)+, 1\$	, SAVE PC AND PSW	
002672	012637	002724		MOV	(6)+, 2\$		
002676	012600			MOV	(6)+, %0	, RESTORE REGS 0 - 4	
002700	012601			MOV	(6)+, %1	, FROM STACK	
002702	012602			MOV	(6)+, %2		
002704	012603			MOV	(6)+, %3		
002706	012604			MOV	(6)+, %4		
002710	013746	002724		MOV	2\$, -(6)	, RESTORE PC AND PSW	
002714	013746	002722		MOV	1\$, -(6)		
002720	000002			RTI		, EXIT	
002722	000000		1\$	OPEN		, CONTAINS SAVED PC	
002724	000000		2\$	OPEN		, CONTAINS SAVED PSW	
, SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER							
002726	000240			TYP	NOP		
002730	011600			MOV	(SP), %0	, GET ADDRESS THAT CONTAINS MESSAGE ADDRESS	
002732	062716	000002		ADD	#2, (SP)	, SET UP EXIT	
002736	011000			MOV	@%0, %0	, ADDRESS OF MESSAGE TO RD	
002740	112037	003040	1\$	MOVB	(0)+, 5\$	, GET CHARACTER	
002744	122737	000100	003040	CMPB	#100, 5\$	, CHECK FOR "@" CHARACTER	
002752	001001			BNE	2\$	, BRANCH IF NOT "@"	
002754	000002			RTI		, TERMINATOR CHAR DONE EXIT	
002756	122737	000045	003040	2\$	CMPB	#45, 5\$	
002764	001412			BEQ	4\$	, CHECK FOR "?"	
002766	004737	002774		JSR	%7, 3\$	, TYPE CHAR IN 5\$	
002772	000762			BR	1\$		
002774	113777	003040	176624	3\$	MOVB	5\$, @TPDBP	, OUTPUT CHARACTER TO PRINTER
003002	105777	176616		TSTB	@TPCSR	, WAIT FOR DONE FLAG	
003006	100375			BPL	-4		
003010	000207			RTS	%7	, EXIT	
003012	112737	000015	003040	4\$	MOVB	#15, 5\$	, MOVE CARRIAGE RETURN CODE TO 5\$
003020	004737	002774		JSR	%7, @#3\$	, GO TYPE CHAR	
003024	112737	000012	003040		MOVB	#12, 5\$	, MOVE LF CODE TO 5\$
003032	004737	002774		JSP	%7, 3\$	, GO TYPE CHAR	
003036	000740			BR	1\$		
003040	000000		5\$	OPEN			
, SUBROUTINE TO GET DM11 PARAMETERS							
, VECTOR ADDRESS							
003042	000240			DMPAR	NOP	, BEGIN	
003044	023737	000042	000046	CMP	@#42, @#46	, ACT11?	
003052	001060			BNE	6\$	, BR IF NO	
, SIZE FOR INTERRUPT VECTOR IN AUTO MODE							
003054	012700	000302		MOV	#302, R0	SET UP FLOATING VECT AREA	
003060	010060	177776	4\$	MOV	R0, -2(R0)		
003064	012720	000003		MOV	#3, (R0)+		
003070	005720			TST	(R0) +		



003406 012701 001250  
 003412 042711 000370  
 003416 063721 003346  
 003422 005302  
 003424 001372

25 MOV #CSR,%1  
 B C #370,(1)  
 ADD UNIT,(1)+  
 DEC %2  
 BNE 25

.CALCULATE CHARACTER LENGTH

003426 012777 001106 175622  
 003434 005077 175610  
 003440 012737 177777 007070  
 003446 012737 177777 001146  
 003454 012737 177777 001634  
 003462 012737 007070 001106  
 003470 012777 003526 175562  
 003476 012777 000340 175556  
 003504 005037 001306  
 003510 012777 000001 175534  
 003516 012777 000105 175524  
 003524 000001  
 003526 005077 175516  
 003532 143737 001306 001634  
 003540 005037 177776  
 003544 022626  
 003546 000207

MOV #CAT,@BASREG  
 CLR @CSR  
 MOV #-1,OUTBUF ;LOAD OUTBUF WITH CHAR TO BE TRANSMITTED  
 MOV #-1,WCT ;SET UP TO TRANSMIT 1 CHAR  
 MOV #-1,@#CARMSK ;PRE SET THE CHARACTER MASK  
 MOV #OUTBUF,CAT ;1 CHARACTER ON LINE 0  
 MOV #35,@CLKINT ;LOAD RECEIVER INTERRUPT  
 MOV #340,@CLKLVL ;AND PRIORITY LEVEL  
 CLR TUMTAB  
 MOV #1,@BAR ;START TRANSMITTING  
 MOV #BIT6+BIT2+BIT0 @CSR ;SET IE,MAINT AND GO BITS  
 WAIT ;WAIT FOR RECEIVER INTERRUPT  
 35 CLR @CSR  
 BICB TUMTAB,CARMSK ;LOAD CHARACTER LENGTH MASK  
 CLR PSW ;RESTORE PROCESSOR TO PRIORITY 0  
 POPSP2 ;RESTORE THE STACK POINTER  
 RTS 7 ;EXIT PARAMETERS ROUTINE

.ROUTINE TO LOAD TRAP/INTERRUPT VECTOR AREA WITH HALT. +2 HALTS PROGRAM  
 .AT ADDRESS OF TRAP/INTERPUPT VECTOR +2

003550 012701 000300  
 003554 012702 000302  
 003560 010221  
 003562 005021  
 003564 020227 000776  
 003570 001403  
 003572 062702 000004  
 003576 000770  
 003600 000240  
 003602 000207

OVRLAY MOV #300,%1  
 MOV #302,%2  
 15 MOV %2,(1)+  
 CLR (1)+  
 CMP %2,#776  
 BEQ 25  
 ADD #4,%2  
 BR 15  
 25 NOP  
 RTS 7 ;EXIT

.SUBROUTINE TO RECEIVE DATA  
 .THIS SUBROUTINE RECEIVES DATA FROM THE KEYBOARD (UP TO SIX OCTAL  
 DIGITS AND PLACES THEM INTO THE ADDRESS FOLLOWING THE SUBROUTINE  
 CALL (JSR 5,RECD) NO REGISTER CONTENTS ARE DISTURBED

.SUBROUTINE TO INPUT DATA FROM TT:

003604 010046  
 003606 005015  
 003610 012737 000007 003762  
 003616 105777 175776  
 003622 100375  
 003624 117700 175772  
 003630 142700 000200  
 003634 110077 175766  
 003640 122700 000025  
 003644 001443

RECD MOV RD,-(SP)  
 15 CLR (5) ;CLEAR OLD DATA  
 MOV #7,CNT ;SET CHAR COUNT  
 25 TSTB @TKCSR ;WAIT FOR CHAR  
 BPL 25  
 MOVB @TKOBR,RO  
 BICB #200,RO ;STRIP OFF PARITY  
 MOVB RO,@TPDBR ;ECHO CHARACTER  
 CMPB #25,RO ;IS IT A U  
 BEQ 55 ;BRANCH IF YES

003646	122700	000015		CMPB	#15, RO		, IS IT A <CR>
003652	001415			BEQ	65		, BRANCH IF YES
003654	142700	000060		B CB	#60, RO		
003660	132700	000110		B TB	#110, RO		, CHECK FOR 0-7 (8)
003664	001031			BNE	75		, BRANCH IF NOT
003666	006315			ASL	(5)		
003670	006315			ASL	(5)		
003672	006315			ASL	(5)		, SHIFT DATA
003674	150015			BISB	RO, (5)		INSET NEW CHAR
003676	005337	003762		DEC	CNT		
003702	001422			BEQ	75		, ONLY 6 CHAR'S PLEASE
003704	000744			BR	25		, NEXT CHARACTER
003706	105777	175712		TSTB	@TPCSR		
003712	100375			BPL	65		, WAIT FOR READY
003714	012777	000012	175704	MOV	#12, @TPDBR		, TYPE <LF>
003722	105777	175676		TSTB	@TPCSR		
003726	100375			BPL	85		, WAIT FOR READY
003730	005077	175672		CLR	@TPDBR		, NEXT CHARACTER
003734	105777	175664		TSTB	@TPCSR		
003740	100375			BPL	95		, WAIT FOR READY
003742	005725			TST	(F5)+		, ADJUST R5
003744	012600			MOV	(SP)+, RO		, RESTORE RO
003746	000205			RTS	R5		
003750	104000			TYPE			
003752	013006			M1			
003754	104000			TYPE			
003756	012646			SCTLU			
003760	000712			BR	15		, START OVER
003762	000000			CNT	0		

. ROUTINE TO CHECK FOR G BEING TYPED

003764	022737	000176	001102	KBDINTT	CMP	#SWREG, SWP	
003772	001021			BNE	15		
003774	023737	000042	000046	CMP	@#42, @#46		, ACT11?
004002	001415			BEQ	15		BR IF YES
004004	005037	004076		CLR	TMP1		, CLEAR TEMP AREA
004010	117737	175606	004076	MOV	@TADBR, TMP1		, FETCH THE BUFFER
004016	142737	000200	004076	BICB	#200, TMP1		STRIP OFF PARITY
004024	122737	000007	004076	CMPB	#7, TMP1		, WAS IT G
004032	001001			BNE	15		, NOP
004034	104020			CNTLU			, GO CHANGE T
004036	000002			RTI			EXIT

. ROUTINE TO CHANGE CONTENTS OF SWREG LOC 176

004040	022737	000176	001102	CNTLUU	CMP	#SWREG, SWP	
004046	001023			BNE	FAJAG		
004050	104000			TYPE			
004052	012617			SSWREG			
004054	004537	004620		JSR	R5, OACNV		CONVERT TO ASCII
004060	000176			SWREG			
004062	012626			SVALUE			
004064	000006				6		



Address	Op1	Op2	Op3	Op4	Label	Type	Value	Comment
004066	104000					TYPE		
004070	012626					SVALUE		
004072	004537	003604				JSR	S, RECD	, GET THE TMP1 &
004076	000000				TMP1	0		, PUT IT HERE
004100	022737	000007	003762			CMP	#7, CNT	
004106	001403					BEQ	FAJAG	
004110	013777	004076	174764			MOV	TMP1, @SWR	, CHANGE CONTENTS OF SWREG
004116	000002				FAJAG	RTI		
004120	013746	000006			SUSWRR.	MOV	@#6, -(SP)	, SAVE VECTORS
004124	013746	000004				MOV	@#4, -(SP)	
004130	012737	004150	000004			MOV	#15, @#4	, SET UP FOR TIMEOUT
004136	022777	177777	174736			CMP	#-1, @SWR	, REFERENCE HARDWARE SWITCH REGISTER
004144	001402					BEQ	2\$	
004146	000407					BR	3\$	
004150	022626				1\$	CMP	(SP)+, (SP)+	, ADJUST STACK
004152	012737	003176	001102		2\$	MOV	#SWREG, SWR	, POINT TO SOFTWARE SWITCH REG
004160	012737	000174	001104			MOV	#DISPREG, DISPLAY	, POINT TO SOFT DISPLAY REG
004166	012637	000004			3\$	MOV	(SP)+, @#4	, RESTORE VECTORS
004172	012637	000006				MOV	(SP)+, @#6	
004176	000002					RTI		

, SUBROUTINE TO TRANSMIT ON ALL LINES WITH A DELAY BETWEEN TRANSMITTING  
 , ON SUCCESSIVE LINES THE DELAY FOR THE TEST IS SUPPL ED BY THE  
 , CALLING JSR INSTRUCTION DATA IS CHECKED AFTER ALL  
 , LINES HAVE FINISHED TRANSMITTING

004200	000240			DLYXMT	NOP		, BEGIN TEST
004202	012777	001106	175046		MOV	#CAT, @BASREG	, SET UP BASE REGISTER
004210	004737	004532			JSR	7, @#IDENT	, TRANSMIT LINE # ON EACH LINE
004214	000240				NOP		, NOP
004216	005077	175026			CLR	@CSR	
004222	012537	004232			MOV	(5)+, 105	, GET MESSAGE ADDRESS
004226	004537	004706			JSR	5, @#BMOVE	, LOAD OUTPUT BUFFER
004232	000000			105	OPEN		, WITH DATA TO
004234	007070				OUTBUF		, BE TRANSMITTED
004236	000100				64,		
004240	005037	001306			CLR	@#TUMTAB	, CLEAR TUMBLE
004244	004537	004706			JSR	5 @#BMOVE	, TABLE (200
004250	001306				TUMTAB		, BYTES)
004252	001307				TUMTAB+1		
004254	000177				177		
004256	004537	004706			JSR	5 @#BMOVE	, CLEAR CHARACTER COUNT TABLE
004262	001306				TUMTAB		
004264	012374				CNTTAB		
004266	000020				16		
004270	005037	007234			CLR	@#LNOBUF	
004274	004537	004706			JSP	5 @#BMOVE	CLEAR ALL
004300	007234				LNOBUF		, LINE'S INPUT
004302	007235				LNOBUF+1		, BUFFERS
004304	003077				1599		(16 BUFFERS OF 100 CHARS EACH)
004306	022737	000006	002154		CMP	#6 PRGNLM	
004314	001002				BNE	+6	
004316	000137	006570			JMP	PRG3A	
004322	012504				MOV	(5)+, %4	GET # OF CHARACTERS TO TRANSMIT BEFORE
							TRANSMITTING ON NEXT LINE
004324	012737	001306	001300		MOV	#TUMTAB, @#TTPTR	, INITIALIZE TUMBLE TABLE POINTER
004332	013701	001260			MOV	@#CLKINT, %1	, GET RECEIVER VECTOR ADDRESS
004336	012721	005134			MOV	#RINT, (1)+	, LOAD RECEIVER VECTOR
004342	013721	001262			MOV	@#CLKLVL, (1)+	, AND PRIORITY LEVEL
004346	012721	005326			MOV	#TINT, (1)+	, LOAD TRANSMITTER VECTOR
004352	013721	001266			MOV	@#XMTLVL, (1)+	, AND PRIORITY LEVEL
004356	005737	002154			TST	PRGNUM	, RUNNING PROGRAM 0?
004362	001402				BEG	+6	
004364	000137	006374			JMP	PRG1A	, RETURN TO PROGRAM 1 CODE
004370	012777	010101	174652		MOV	#BIT12+BIT6+BIT0, @CSP	, SET IE & GC BITS
004376	012737	000001	001274		MOV	#1, @#LINBIT	
004404	005037	001646			CLR	@#LINE	
004410	013700	001646		15	MOV	LINE, %0	, LINE # X2 TO PO
004414	000240				NOP		, NOP
004416	004537	004730			JSR	5, @#XMITC	, TRANSMIT 64 CHARACTERS
004422	177700				-64		, ON LINE # AS SPECIFIED IN ADDRESS LINE
004424	020460	001146		25	CMP	%4, WCT(0)	, WAIT FOR THE WORD COUNT TO DEC TO THE
004430	001375				BNE	25	, CORRECT VALUE BEFORE STARTING NEXT LINE
004432	062737	000002	001646		ADD	#2, LINE	, FORM NEXT LINE NUMBER
004440	006337	001274			ASL	LINBIT	, SHIFT LINE BIT
004444	103361				BCC	15	, START NEXT LINE
004446	005760	001146		35	TST	WCT(0)	, WAIT FOR LAST LINE TO FINISH

```

004452 001375          BNE      3$
004454 042777 177400 174566 31$ BIC      #177400,@CSR ,CLEAR ODD BYTE OF CSR
004462 062700 000001          ADD      #1,R0 ,WAIT FOR RECEIVER TO RECEIVE
004466 001375          BNE      31$ ,ALL TRANSMITTED DATA
004470 017737 174556 001630 MOV      @BAR,RCV DAT ,GET AND TEST BAR CONTENTS
004476 001410          BEQ      4$ ,BRANCH IF IS CLEAR

004500 005037 001632          CLR      XMT DAT
004504 005077 174540          CLR      @CSR
004510 005077 174536          CLR      @BAR
004514 104015          ERROR1
004516 000403          BR       5$ ,ERROR BAR DID NOT CLEAR IN SUFFICIENT TIME
004520 000240          NOP
004522 004737 005026 4$ JSR      7,@#CHK DAT ,GO TEST DATA
004526 022626          CMP      (6)+,(6)+ ,RESET THE STACK
004530 104012          SCOPE ,SCOPE
  
```

```

                                SUBROUTINE TO TRANSMIT ON EACH LINE ITS LINE NUMBER (CRLF XX CRLF)
004532 005037 001646 IDENT CLR      @#LINE ,GET LINE NUMBER 0
004536 012737 000001 001274 MOV      #1,@#LIN BIT ,GET LINE BIT
004544 013702 001646 1$ MOV      LINE,%2
004550 016262 012414 001106 MOV      10(2),CAT(2) ,LOAD CAT
004556 012762 177772 001146 MOV      #-6,WCT(2) ,LOAD WORD COUNT
004564 053777 001274 174460 BIS      LIN BIT,@BAR ,SET BAR BIT
004572 062737 000002 001646 ADD      #2,LINE ,FORM NEXT LINE NUMBER
004600 006337 001274 ASL      LIN BIT ,FORM NEXT LINE BIT
004604 103357          BCC      1$ ,BRANCH IF NOT DONE
004606 005777 174440 2$ TST      @BAR ,WAIT FOR BAR TO CLEAR
004612 001375          BNE      2$
004614 000240          NOP
004616 000207          RTS      ? ,EXIT SUBROUTINE
  
```

```

                                .OCTAL TO ASCII CONVERT ROUTINE
004620 104013 OACNV SAVREG ,SAVE REGISTERS ON THE STACK
004622 013537 004704 MOV      @5)+,2$ ,GET OCTAL VALUE
004626 012501 MOV      (5)+,%1 ,GET DESTINATION ADDR
004630 012502 MOV      (5)+,%2 ,GET CONVERT COUNT
004632 060201 ADD      %2,%1 ,DEVELOP ADDR TO STORE 1ST CHAR
004634 013703 004704 1$ MOV      2$,%3
004640 042703 177770 BIC      #177770,%3 ,ISOLATE LEAST SIGNIFICANT DIGIT
004644 062703 000060 ADD      #60,%3 ,CONVERT DIGIT TO ASCII
004650 110341 MOV      %3,-(1) ,STORE ASCII CHARACTER
004652 042737 000007 004704 BIC      #7,2$
004660 006037 004704 ROR      2$
004664 006037 004704 ROR      2$
004670 006037 004704 ROR      2$
004674 065302 DEC      %2 ,DONE ALL DIGITS?
004676 001356 BNE      1$ ,BRANCH IF NOT DONE
004700 104014 RSTREG ,RESTORE THE REGISTERS
004702 000205 RTS      1$ ,DONE EXIT
004704 000000 2$ OPEN
  
```

SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES

004706 104013  
004710 012501  
004712 012502  
004714 012503  
004716 112122  
004720 005303  
004722 001375  
004724 104014  
004726 000205

BMOVE	SAVREG		, SAVE REGS
	MOV	(5)+,%1	, GET"FROM"ADDRESS
	MOV	(5)+,%2	, GET"TO"ADDRESS
	MOV	(5)+,%3	, GET COUNT
15	MOVB	(1)+,(2)+	, MOVE BYTE
	DEC	%3	, DECREMENT COUNT
	BNE	15	, BRANCH IF NOT DONE
	PSTREG		, RESTORE REGS
	RTS	%5	, DONE EXIT



```

, SUBROUTINE TO TRANSMIT DATA      SUBROUTINE CALLED BY
, JSR 5, XMITD
XMITD  NOP
      MOV    %0, -(SP)      , SAVE RO ON THE STACK
      MOV    @#LINE, %0    , GET LINE
      MOV    #OUTBUF, CAT(0) , LOAD FIRST CHAR ADDRESS IN CAT
      MOV    (5)+, WCT(0)   , LOAD WORD COUNT INTO LINE'S TABLE ADDRESS
      BIS    @#LINBIT, @#BARIM, LOAD LINE POSITION INTO BAR IMAGE
      BIS    LINBIT, @BAR   , START TRANSMITTING ON LINE SPECIFIED
                          , IN LINBIT
      MOV    (SP)+, %0      , RESTORE RO
      NOP
      RTS    5            , EXIT

, SUBROUTINE TO FORM LINE BIT POSITION WITH THE LINE # IN LINE
GTLINB MOV    %0, -(SP)    , SAVE RO ON THE STACK
      CLR    @#LINBIT
      MOV    @#LINE, %0    , GET LINE
      SEC
      SEC    , SET CARRY
      ROL    LINBIT       , SHIFT LINE BIT
      SUB    #2, %0       , SUBTRACT 2 FROM LINE NUMBER
      BPL    15
      MOV    (SP)+, %0    , RESTORE RO
      RTS    7            , EXIT

SUBROUTINE TO CHECK TRANSMITTED DATA
CHKDAT SAVREG      , SAVE THE REGISTERS ON THE STACK
      NOP
      CLR    %1         , CLEAR CHARACTER COUNT
      MOV    #INTAB, %2  , GET ADDRESS OF LINE'S INPUT BUFFER
      CLR    %3         , ADDRESS, GET LINE COUNT
      MOV    %3, @#LINE  , MOVE LINE # TO LINE
      MOV    (2)+, @#LINBUF , GET LINE'S INPUT BUFFER ADDRESS
      DEC    LINBUF     , SUBTRACT 1 FROM LINE'S INPUT BUFFER ADDRESS
      INC    LINBUF     , INCREMENT LINE'S INPUT BUFFER ADDRESS
      MOVB  @LINBUF, @#RCVDAT , GET RECEIVED CHARACTER
      MOVB  OUTBUF(1), XMTDAT, GET TRANSMITTED CHARACTER
      BIC   @#CARMSK, XMTDAT , CLEAR UNTRANSMITTED BITS
      DATCHK
      INC    %1         , INCREMENT CHARACTER COUNT
      CMP    %1, #64    , ALL CHARACTERS BEEN COMPARED
      BNE   25         , GO CHECK NEXT CHAR IF NOT
      CLR    %1         , CLEAR CHARACTER COUNT
      INC    %3         , INCREMENT LINE COUNT
      CMP    %3, #16    , ALL LINES CHECKED?
      BLT   15         , BRANCH IF ALL LINES NOT CHECKED
      RSTREG
      RTS    7            , EXIT SUBROUTINE

RECEIVER INTERRUPT SERVICE ROUTINE
PINT   NOP
      SAVREG      , SAVE THE REGISTERS ON THE STACK
  
```

005134 000240  
 005136 104013

005140	013701	001300		MOV	@#TTPTR,%1	,GET TUMBLE TABLE POINTER
005144	011137	001272		MOV	(1),TTDAT	,GET TUMBLE TABLE ENTRY
005150	100410			BMI	25	,BRANCH IF VALID DATA ENTRY
005152	104003			ERROR		,ERROR! FALSE INTERRUPT
005154	000454			BR	65	,EXIT
005156	011137	001272	15	MOV	(1),@#TTDAT	,GET TUMBLE TABLE ENTRY
005162	001451			BEQ	65	,GO TO EXIT IF NO DATA ENTRY
005164	100402			BMI	25	,BRANCH IF VALID DATA ENTRY
005166	104003			ERROR		,ERROR! NO VALID DATA ENTRY INDICATOR
005170	000425			BR	35	
005172	005011		25	CLR	(1)	,CLEAR TUMBLE TABLE ENTRY
005174	042737	160400	001272	BIC	#160400,@#TTDAT	,CLEAR ALL BUT CHAR & LINE #
005202	113702	001273		MOVB	TTDAT+1,%2	,PUT LINE # IN R2 (LINE WILL BE IN LSH)
005206	010204			MOV	%2,%4	
005210	016237	012334	001650	MOV	INTAB(2),@#LINBUF	,GET LINE'S INPUT BUFFER ADDRESS
005216	006202			ASR	%2	,SHIFT LINE #
005220	005003			CLR	%3	
005222	116203	012374		MOVB	CNTTAB(2),%3	,GET LINE'S RECEIVED CHAR COUNT
005226	105262	012374		INCB	CNTTAB(2)	,INCREMENT CHARACTER COUNT
005232	060337	001650		ADD	%3,LINBUF	,FORM ADDRESS WHERE CHAR IS TO BE STORED
005236	113777	001272	174404	MOVB	TTDAT,@LINBUF	,STORE CHAR IN LINE'S INPUT BUFFER
005244	000240		35	NOP		,NOP
005246	016437	001146	001630	MOV	WCT(4),RCV DAT	,GET TRANSMITTERS WORD COUNT
005254	003405			BLE	45	,BRANCH IF WORD COUNT IS 0 OR NEGATIVE
005256	010437	001630		MOV	%4,XMTDAT	,GET LINE # OF FAILING LINE
005262	104015			ERROR1		,ERROR! INCORRECT WORD COUNT N
				TYPE OUT SHOWS	FAILING LINE #,	AND FAILING LINE'S WORD COUNT
005264	000005			RESET		,STOP THE DM11
005266	104012			SCOPE		,EXIT TEST
005270	022701	001504	45	CMP	#TUMTAB+176,%1	,IS THE TUMBLE TABLE POINTER AT THE
005274	001002			BNE	55	,THE END OF THE TABLE
005276	012701	001304		MOV	#TUMTAB-2,%1	,RESET POINTER
005302	005721		55	TST	(1)+	,INCREMENT POINTER
005304	000724			BR	15	,GO CHECK NEXT ENTRY
005306	042777	000200	173734	BIC	#BIT7,@CSR	,CLEAR RECEIVER DONE FLAG
005314	010137	001300		MOV	%1,TTPTR	,SAVE POINTER
005320	104014			RSTREG		,RESTORE THE REGISTERS
005322	000240			NOP		
005324	000002			RTI		,EXIT SERVICE ROUTINE
				,TRANSMITTER INTERRUPT SERVICE ROUTINE		
005326	000240			TINT	NOP	,BEGIN
005330	032777	060000	173712	BIT	#BIT14+BIT13,@CSR	,TEST ERROR FLAGS
005336	001404			BEQ	15	,BRANCH IF NO ERROR FLAGS
005340	104003			ERROR		,ERROR! ERROR FLAG IS SET
005342	042777	060000	173700	BIC	#BIT14+BIT13,@CSR	,CLEAR ERROR FLAGS
005350	005777	173674	15	TST	@CSR	,TEST READY FLAG
005354	100003			BPL	25	,BRANCH IF READY IS CLEAR
005356	042777	100000	173664	BIC	#BIT15,@CSR	,CLEAR READY FLAG
005364	000002		25	RTI		

```
005366 104000 PRGO TYPE
005370 013127 PRGOM
005372 012737 005426 001516 PRGOA MOV #RTO,KSTART ,GET ADDRESS OF FIRST TEST
005400 005037 001522 CLR RTNNO ,CLEAR ROUTINE #
005404 000137 002214 JMP SRSET
005410 012737 005426 001516 PRGOR MOV #RTO,KSTART ,GET ADDRESS OF FIRST TEST
005416 005037 001522 CLR RTNNO ,CLEAR ROUTINE NUMBER
005422 000137 002246 JMP GETROY ,GO AND START PROGRAM
,*****
005426 000000 RTO 0 ,ROUTINE # 0 *
005430 005446 RT1 ,ADDR OF NEXT ROUTINE *
005432 000002 2 ,ITERATION COUNT *
005434 005436 RTOA ,SCOPE ENTRY POINT *
000000 X=X+1
,*****
,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
,NEXT LINE
005436 004537 004200 RTOA JSR 5,DLYXMT ,GO DO TEST.
005442 013451 MSG1 ,TRANSMIT THIS MESSAGE &
005444 000000 0 ,DELAY THIS MUCH BETWEEN LINES
,*****
005446 000001 RT1 1 ,ROUTINE # 1 *
005450 005466 RT2 ,ADDR OF NEXT ROUTINE *
005452 000002 2 ,ITERATION COUNT *
005454 005456 RT1A ,SCOPE ENTRY POINT *
000001 X=X+1
,*****
,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
,NEXT LINE
005456 004537 004200 RT1A JSR 5,DLYXMT ,GO DO TEST
005462 013451 MSG1 ,TRANSMIT THIS MESSAGE &
005464 177740 -32 ,DELAY THIS MUCH BETWEEN LINES
,*****
005466 000002 RT2 2 ,ROUTINE # 2 *
005470 005506 RT3 ,ADDR OF NEXT ROUTINE *
005472 000002 2 ,ITERATION COUNT *
005474 005476 RT2A ,SCOPE ENTRY POINT *
000002 X=X+1
,*****
,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
,NEXT LINE
005476 004537 004200 RT2A JSR 5,DLYXMT ,GO DO TEST
005502 013 51 MSG1 ,TRANSMIT THIS MESSAGE &
005504 177720 -48 ,DELAY THIS MUCH BETWEEN LINES
,*****
005506 000003 RT3 3 ,ROUTINE # 3 *
005510 005526 RT4 ,ADDR OF NEXT ROUTINE *
005512 000002 2 ,ITERATION COUNT *
005514 005516 RT3A ,SCOPE ENTRY POINT *
000003 X=X+1
,*****
,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
,NEXT LINE
005516 004537 004200 RT3A JSR 5,DLYXMT ,GO DO TEST
005522 013451 MSG1 ,TRANSMIT THIS MESSAGE &
```

```
005524 177710 -56 , DELAY THIS MUCH BETWEEN L NES
, *****
005526 000004 RT4 4 , ROUTINE # 4 *
005530 005546 RT5 , ADDR OF NEXT ROUTINE *
005532 000002 2 , ITERATION COUNT *
005534 005536 RT4A , SCOPE ENTRY POINT *
000004 X=X+1
, *****
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
, NEXT LINE
005536 004537 004200 RT4A JSR 5, DLYXMT , GO DO TEST
005542 013451 MSG1 , TRANSMIT THIS MESSAGE &
005544 177704 -60 , DELAY THIS MUCH BETWEEN LINES
, *****
005546 000005 RT5 5 , ROUTINE # 5 *
005550 005566 RT6 , ADDR OF NEXT ROUTINE *
005552 000002 2 , ITERATION COUNT *
005554 005556 RT5A , SCOPE ENTRY POINT *
000005 X=X+1
, *****
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STAT NG THE
, NEXT LINE
005556 004537 004200 RT5A JSR 5, DLYXMT , GO DO TEST
005562 013451 MSG1 , TRANSMIT THIS MESSAGE &
005564 177702 -62 , DELAY THIS MUCH BETWEEN LINES
, *****
005566 000006 PT6 6 , ROUTINE # 6 *
005570 005606 PT7 , ADDR OF NEXT ROUTINE *
005572 000002 2 , ITERATION COUNT *
005574 005576 RT6A , SCOPE ENTRY POINT *
000006 X=X+1
, *****
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
, NEXT LINE
005576 004537 004200 RT6A JSR 5, DLYXMT , GO DO TEST
005602 013451 MSG1 , TRANSMIT THIS MESSAGE &
005604 177701 -63 , DELAY THIS MUCH BETWEEN LINES
, *****
005606 000007 PT7 7 , ROUTINE # 7 *
005610 005626 RT10 , ADDR OF NEXT ROUTINE *
005612 000002 2 , ITERATION COUNT *
005614 005616 RT7A , SCOPE ENTRY POINT *
000007 X=X+1
, *****
, TEST TO TPANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
, NEXT LINE
005616 004537 004200 PT7A JSR 5, DLYXMT , GO DO TEST
005622 013451 MSG1 , TRANSMIT THIS MESSAGE &
005624 177700 -64 , DELAY THIS MUCH BETWEEN LINES
, *****
005626 000010 PT10 10 , ROUTINE # 10 *
005630 005646 PT11 , ADDR OF NEXT ROUTINE *
005632 000002 2 , ITERATION COUNT *
005634 005636 RT10A , SCOPE ENTRY POINT *
000010 X=X+1
, *****
```

```

,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STAT NG THE
,NEXT LINE
005636 004537 004200 RT10A JSR 5,DLYXMT ,GO DO TEST
005642 013552 MSG2 ,TRANSMIT THIS MESSAGE &
005644 177740 -32 ,DELAY THIS MUCH BETWEEN LINES
,*****
005646 000011 RT11 11 ,ROUTINE # 11 *
005650 005666 RT12 ,ADDR OF NEXT ROUTINE *
005652 000002 2 ,ITERATION COUNT *
005654 005656 RT11A ,SCOPE ENTRY POINT *
000011 X=X+1
,*****
,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
,NEXT LINE
005656 004537 004200 RT11A JSR 5,DLYXMT ,GO DO TEST
005662 013552 MSG2 ,TRANSMIT THIS MESSAGE &
005664 177720 -48 ,DELAY THIS MUCH BETWEEN LINES
,*****
005666 000012 RT12 12 ,ROUTINE # 12 *
005670 005706 RT13 ,ADDR OF NEXT ROUTINE *
005672 000002 2 ,ITERATION COUNT *
005674 005676 RT12A ,SCOPE ENTRY POINT *
000012 X=X+1
,*****
,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
,NEXT LINE
005676 004537 004200 RT12A JSR 5,DLYXMT ,GO DO TEST
005702 013552 MSG2 ,TRANSMIT THIS MESSAGE &
005704 177710 -56 ,DELAY THIS MUCH BETWEEN LINES
,*****
005706 000013 RT13 13 ,ROUTINE # 13 *
005710 005726 RT14 ,ADDR OF NEXT ROUTINE *
005712 000002 2 ,ITERATION COUNT *
005714 005716 RT13A ,SCOPE ENTRY POINT *
000013 X=Y+1
,*****
,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
,NEXT LINE
005716 004537 004200 RT13A JSP 5,DLYXMT ,GO DO TEST
005722 013552 MSG2 ,TRANSMIT THIS MESSAGE &
005724 177704 -60 ,DELAY THIS MUCH BETWEEN LINES
,*****
005726 000014 RT14 14 ,ROUTINE # 14 *
005730 005746 RT15 ,ADDR OF NEXT ROUTINE *
005732 000002 2 ,ITERATION COUNT *
005734 005736 RT14A ,SCOPE ENTRY POINT *
000014 X=X+1
,*****
,TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
,NEXT LINE
005736 004537 004200 RT14A JSR 5,DLYXMT ,GO DO TEST
005742 013552 MSG2 ,TRANSMIT THIS MESSAGE &
005744 177702 -62 ,DELAY THIS MUCH BETWEEN LINES
,*****
005746 000015 RT15 15 ,ROUTINE # 15 *
005750 005766 RT16 ,ADDR OF NEXT ROUTINE *
```



005752 000002  
005754 005756  
000015

005756 004537 004200  
005762 013552  
005764 177701

005766 000016  
005770 006006  
005772 000002  
005774 005776  
000016

```
      2          , ITERATION COUNT          *  
      RT15A      , SCOPE ENTRY POINT        *  
      X=X+1  
      , *****  
      , TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
      , NEXT LINE  
RT15A JSR      5, DLYXMT      , GO DO TEST  
      MSG2      , TRANSMIT THIS MESSAGE &  
      -63      , DELAY THIS MUCH BETWEEN LINES  
      , *****  
RT16   16      , ROUTINE # 16                *  
      RT17      , ADDR OF NEXT ROUTINE        *  
      2          , ITERATION COUNT          *  
      RT16A     , SCOPE ENTRY POINT          *  
      X=X+1
```

005776 004537 004200  
006002 013552  
006004 177700

006006 000017  
006010 006026  
006012 000002  
006014 006016  
000017

```
      , *****  
      , TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
      , NEXT LINE  
RT16A JSR      5, DLYXMT      , GO DO TEST  
      MSG2      , TRANSMIT THIS MESSAGE &  
      -64      , DELAY THIS MUCH BETWEEN LINES  
      , *****  
RT17   17      , ROUTINE # 17                *  
      RT20      , ADDR OF NEXT ROUTINE        *  
      2          , ITERATION COUNT          *  
      RT17A     , SCOPE ENTRY POINT          *  
      X=X+1
```

006016 004537 004200  
006022 013652  
006024 177720

006026 000020  
006030 006046  
006032 000002  
006034 006036  
000020

```
      , *****  
      , TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
      , NEXT LINE  
RT17A JSR      5, DLYXMT      , GO DO TEST  
      MSG3      , TRANSMIT THIS MESSAGE &  
      -48      , DELAY THIS MUCH BETWEEN LINES  
      , *****  
RT20   20      , ROUTINE # 20                *  
      RT21      , ADDR OF NEXT ROUTINE        *  
      2          , ITERATION COUNT          *  
      RT20A     , SCOPE ENTRY POINT          *  
      X=X+1
```

006036 004537 004200  
006042 013652  
006044 177704

006046 000021  
006050 006066  
006052 000002  
006054 006056  
000021

```
      , *****  
      , TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
      , NEXT LINE  
RT20A JSR      5, DLYXMT      , GO DO TEST  
      MSG3      , TRANSMIT THIS MESSAGE &  
      -60      , DELAY THIS MUCH BETWEEN LINES  
      , *****  
RT21   21      , ROUTINE # 21                *  
      RT22      , ADDR OF NEXT ROUTINE        *  
      2          , ITERATION COUNT          *  
      RT21A     , SCOPE ENTRY POINT          *  
      X=X+1
```

006056 004537 004200  
006062 013652

```
      , *****  
      , TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
      , NEXT LINE  
RT21A JSR      5, DLYXMT      , GO DO TEST  
      MSG3      , TRANSMIT THIS MESSAGE &
```

006064 177701  
006066 000022  
006070 006106  
006072 000002  
006074 006076  
006074 000022

-63 , DELAY THIS MUCH BETWEEN L NES  
\*\*\*\*\*  
RT22 22 , ROUTINE # 22 \*  
RT23 , ADDR OF NEXT ROUTINE \*  
2 , ITERATION COUNT \*  
RT22A , SCOPE ENTRY POINT \*  
X=X+1

006076 004537 004200  
006102 013652  
006104 177700

\*\*\*\*\*  
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
, NEXT LINE  
RT22A JSR 5, DLYXMT , GO DO TEST  
MSG3 , TRANSMIT THIS MESSAGE &  
-64 , DELAY THIS MUCH BETWEEN L NES

006106 000023  
006110 006126  
006112 000002  
006114 006116  
006114 000023

\*\*\*\*\*  
RT23 23 , ROUTINE # 23 \*  
RT24 , ADDR OF NEXT ROUTINE \*  
2 , ITERATION COUNT \*  
RT23A , SCOPE ENTRY POINT \*  
X=X+1

006116 004537 004200  
006122 013752  
006124 177740

\*\*\*\*\*  
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
, NEXT LINE  
RT23A JSR 5, DLYXMT , GO DO TEST  
MSG4 , TRANSMIT THIS MESSAGE &  
-32 , DELAY THIS MUCH BETWEEN L NES

006126 000024  
006130 006146  
006132 000002  
006134 006136  
006134 000024

\*\*\*\*\*  
RT24 24 , ROUTINE # 24 \*  
RT25 , ADDR OF NEXT ROUTINE \*  
2 , ITERATION COUNT \*  
RT24A , SCOPE ENTRY POINT \*  
X=X+1

006136 004537 004200  
006142 013752  
006144 177710

\*\*\*\*\*  
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STAT NG THE  
, NEXT LINE  
RT24A JSR 5, DLYXMT , GO DO TEST  
MSG4 , TRANSMIT THIS MESSAGE &  
-56 , DELAY THIS MUCH BETWEEN L NES

006146 000025  
006150 006166  
006152 000002  
006154 006156  
006154 000025

\*\*\*\*\*  
RT25 25 , ROUTINE # 25 \*  
RT26 , ADDR OF NEXT ROUTINE \*  
2 , ITERATION COUNT \*  
RT25A , SCOPE ENTRY POINT \*  
X=X+1

006156 004537 004200  
006162 013752  
006164 177702

\*\*\*\*\*  
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
, NEXT LINE  
RT25A JSR 5, DLYXMT , GO DO TEST  
MSG4 , TRANSMIT THIS MESSAGE &  
-62 , DELAY THIS MUCH BETWEEN L NES

006166 000026  
006170 006206  
006172 000002  
006174 006176  
006174 000026

\*\*\*\*\*  
RT26 26 , ROUTINE # 26 \*  
RT27 , ADDR OF NEXT ROUTINE \*  
2 , ITERATION COUNT \*  
RT26A , SCOPE ENTRY POINT \*  
X=X+1  
\*\*\*\*\*

006176 004537 004200  
006202 013752  
006204 177700

```
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STAT NG THE  
, NEXT LINE  
RT26A JSR 5. DLYXMT , GO DO TEST  
MSG4 , TRANSMIT THIS MESSAGE &  
-64 , DELAY THIS MUCH BETWEEN L NES  
*****  
RT27 27 , ROUTINE # 27 *  
RT30 , ADDR OF NEXT ROUTINE *  
2 , ITERATION COUNT *  
RT27A , SCOPE ENTRY POINT *  
X=X+1
```

006206 000027  
006210 006226  
006212 000002  
006214 006216  
000027

006216 004537 004200  
006222 014052  
006224 177720

```
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STAT NG THE  
, NEXT LINE  
RT27A JSR 5. DLYXMT , GO DO TEST  
MSG5 , TRANSMIT THIS MESSAGE &  
-48 , DELAY THIS MUCH BETWEEN L NES  
*****  
RT30 30 , ROUTINE # 30 *  
RT31 , ADDR OF NEXT ROUTINE *  
2 , ITERATION COUNT *  
RT30A , SCOPE ENTRY POINT *  
X=X+1
```

006226 000030  
006230 006246  
006232 000002  
006234 006236  
000030

006236 004537 004200  
006242 014052  
006244 177710

```
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
, NEXT LINE  
RT30A JSR 5. DLYXMT , GO DO TEST  
MSG5 , TRANSMIT THIS MESSAGE &  
-56 , DELAY THIS MUCH BETWEEN L NES  
*****  
RT31 31 , ROUTINE # 31 *  
RT32 , ADDR OF NEXT ROUTINE *  
2 , ITERATION COUNT *  
RT31A , SCOPE ENTRY POINT *  
X=X+1
```

006246 000031  
006250 006266  
006252 000002  
006254 006256  
000031

006256 004537 004200  
006262 014052  
006264 177704

```
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
, NEXT LINE  
RT31A JSR 5. DLYXMT , GO DO TEST  
MSG5 , TRANSMIT THIS MESSAGE &  
-60 , DELAY THIS MUCH BETWEEN L NES  
*****  
RT32 32 , ROUTINE # 32 *  
RT33 , ADDR OF NEXT ROUTINE *  
2 , ITERATION COUNT *  
RT32A , SCOPE ENTRY POINT *  
X=X+1
```

006266 000032  
006270 006306  
006272 000002  
006274 006276  
000032

006276 004537 004200  
006302 014052  
006304 177702

```
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
, NEXT LINE  
RT32A JSR 5. DLYXMT , GO DO TEST  
MSG5 , TRANSMIT THIS MESSAGE &  
-62 , DELAY THIS MUCH BETWEEN L NES  
*****  
RT33 33 , ROUTINE # 33 *  
RT34 , ADDR OF NEXT ROUTINE *
```

006306 000033  
006310 006306

006312 000002  
006314 006316  
000033

2  
RT33A  
X=X+1

, ITERATION COUNT \*  
, SCOPE ENTRY POINT \*

\*\*\*\*\*  
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
, NEXT LINE

006316 004537 004200  
006322 014052  
006324 177701

RT33A JSR 5, DLYXMT  
MSG5  
-63

, GO DO TEST \*  
, TRANSMIT THIS MESSAGE & \*  
, DELAY THIS MUCH BETWEEN LINES \*

006326 000034  
006330 177777  
006332 000002  
006334 006336  
000034

RT34 34  
RT35  
2  
RT34A  
X=X+1

\*\*\*\*\*  
, ROUTINE # 34 \*  
, ADDR OF NEXT ROUTINE \*  
, ITERATION COUNT \*  
, SCOPE ENTRY POINT \*

\*\*\*\*\*  
, TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE  
, NEXT LINE

006336 004537 004200  
006342 014052  
006344 177700  
177777

RT34A JSR 5, DLYXMT  
MSG5  
-64  
RT35=-1

, GO DO TEST \*  
, TRANSMIT THIS MESSAGE & \*  
, DELAY THIS MUCH BETWEEN LINES \*

```

,PRG1- DATA TESTS ALL LINES SIMULTANEOUSLY DATA TRANSMITTED IS 'THE
,QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
006346 104000 PRG1 TYPE ,TYPE
006350 013155 PRG1M ,PROGRAM TITLE
006352 022737 00C176 001102 PRGX CMP #SWREG,SWR ,SEE IF SWITCH-LESS
006360 001001 PRGX BNE PRG1R ,BRANCH IF NOT
006362 104020 PRGX CNTLU ,GET SWREG SETTINGS
006364 004537 004200 PRG1R JSR 5,DLYXMT ,GO TO DLYXMT TO SET UP DM11
006370 013451 PRG1R MSG1 ,MSG1 WILL BE THE DATA TRANSMITTED
006372 177700 PRG1R -64 ,DO NOT DELAY
006374 012737 007070 001106 PRG1A MOV #OUTBUF,CAT ,LOAD CURRENT
006402 004537 004706 PRG1A JSR 5,BMOVE ,ADDRESS TABLE
006406 001106 PRG1A CAT ,TO POINT TO
006410 001110 PRG1A CAT+2 ,OUTBUF
006412 000040 PRG1A 32
006414 012737 177700 001146 PRG1A MOV #-64,WCT ,LOAD WORD COUNT
006422 004537 004706 PRG1A JSR 5,BMOVE ,TO -64
006426 001146 PRG1A WCT
006430 001150 PRG1A WCT+2
006432 000040 PRG1A 32
006434 012777 010100 172606 PRG1A MOV #BIT12+BIT6,DCSR ,SET TRANSMITTER & RECIEVER LE BITS
006442 023727 002154 000004 PRG1A CMP PRGNUM,#4 ,RUNNING PROGRAM #2?
006450 001403 PRG1A BEQ +10
006452 052777 000001 172570 PRG1A BIS #BIT0,DCSR ,SET THE GO BIT
006460 012777 172570 172564 PRG1A MOV #-1,ABAP ,START TRANSMITTING ON ALL LINES
006466 005777 172560 PRG1A TST ABAP ,WAIT FOR ALL LINES TO COMPLETE
006472 001375 PRG1A BNE -4
006474 005205 PRG1A INC %5
006476 001376 PRG1A BNE -2
006500 005077 172544 PRG1A CLP DCSR
006504 023727 002154 000004 PRG1C CMP PRGNUM #4 ,DO NOT CHECK DATA IF RUNNING
006512 001402 PRG1C BEQ PRG1D ,PROGRAM # 2
006514 004737 005026 PRG1C JSR 7,CHEKDAT ,GO CHECK RECEIVED DATA
006520 104000 PRG1C TYPE ,TYPE
006522 013011 PRG1C M2 ,'PRGEND'
006524 012706 001100 PRG1E+ MOV #SPBOT,SP ,RESET THE STACK POINTER
006530 000715 PRG1E+ BR PRG1P ,GO RESTART TEST

```

.PRG2-PROGRAM 2 RUNS PROGRAM 1 EXCEPT FOR THE DATA CHECKING  
 .WHEN ALL LINES ARE FINISHED TRANSMITTING THIS ALLOWS THE DATA  
 .TRANSMITTED TO BE SENT TO TERMINALS BEFORE STARTING THIS PROGRAM  
 .REMOVE THE JUMPERS CONNECTING THE TRANSMITTERS TO THE RECEIVERS

```

006532 104000 PRG2 TYPE ,TYPE
006534 013235 PRG2M ,PROGRAM TITLE
006536 000137 006752 PRG2P IMP PRG2 ,GO RUN PRG1

```

, PRG3-ECHO TEST THIS PROGRAM ECHOS BACK DATA RECEIVED FROM ANY DM11  
 , TERMINAL(S)  
 , NOTE THIS TEST IS THE ONLY TEST THAT INSURES PROPER OPERATION  
 , OF THE DM11 DISTRIBUTION PANEL LOGIC

ADDRESS	DATA	ADDRESS	DATA	PRG3	TYPE	OPERATION	DESCRIPTION
006542	104000			PRG3	TYPE		. TYPE PROGRAM
006544	013264				PRG3M		. TITLE
006546	022737	000176	001102		CMP	#SWREG, SWR	. SEE IF SWITCH-LESS
006554	001001				BNE	PRG3R	. BRANCH IF NOT
006556	104020				CNT, U		. GET SWREG SETTINGS
006560	004537	004200		PRG3R	JSR	5. DLYXMT	. USE PART OF THE
006564	013451				MSG1		. DLYXMT ROUTINE TO
006566	000240				NOP		. SET UP DM11
006570	012737	001306	001300	PRG3A	MOV	#TUMTAB, TTPTP	. INITIALIZE SOFTWARE POINTER
006576	013701	001260			MOV	CLKINT, %1	. LOAD RECEIVER
006602	012721	006640			MOV	#RINT3, (1)+	. AND TRANSMITTER
006606	013721	001262			MOV	CLKLVL, (1)+	. VECTORS AND PRIORITY
006612	012721	007044			MOV	#TINT3, (1)+	. LEVELS
006616	013721	001266			MOV	XMTLVL, (1)+	
006622	012777	010101	172420		MOV	#BIT12+BIT6+BIT0, @CSR	. SET IE AND GC BITS
006630	012700	000001			MOV	#1, %0	
006634	005200				INC	%0	
006636	000776				BR	-C	
006640	000240			PINT3	NOP		
006642	000240				NOP		
006644	013701	001300			MOV	TTPTP, %1	. GET SOFTWARE POINTER
006650	011137	001272		PINT3A	MOV	(1), TTDAT	. GET TUMBLE TABLE ENTRY
006654	001463				BEQ	RINT3X	. EXIT IF NO ENTRY
006656	005011				CLP	(1)	. CLEAR ENTRY
006660	032737	040000	001272		BIT	#BIT14, TTDAT	. WAS BREAK RECEIVED
006666	001047				BNE	RINT3B	. DO NOTHING ABOUT IT
006670	042737	160400	001272		BIC	#160400, TTDAT	. CLEAR ALL BUT LINE # AND DATA
006676	113702	001273			MOVB	TTDAT+1, %2	. GET LINE NUMBER
006702	010237	001646			MOV	%2, LINE	. FETCH LINE NUMBER
006706	034737	004774			JSP	7, GTLINB	. FORM LINE BIT FOR BAP
006712	033777	001274	172332		BIT	LINBIT, @BAP	. IS THIS LINE ACTIVE
006720	001414				BEQ	NONACT	. LINE NOT ACTIVE
006722	033777	001274	172322		BIT	LINBIT, @BAP	. WAIT FOR LINE
006730	001374				BNE	-6	
006732	032777	060000	172310		BIT	#BIT14+BIT13, @CSR	
006740	001401				BEQ	+4	. BRANCH IF NO ERRORS
006742	104003				ERROR		
006744	042777	100000	172276		BIC	#BIT15, @CSR	. CLEAR TRANSMIT DONE
006752	113762	001272	007070	NONACT	MOVB	TTDAT, OUTBUF(2)	. STORE RECEIVED CHARACTER
006760	012762	177777	001146		MOV	#-1, WCT(2)	. LOAD LINE'S WORD COUNT
006766	010203				MOV	%2, %3	
006770	062703	007070			ADD	#OUTBUF, %3	
006774	010362	001106			MOV	%3, CAT(2)	. AND CURRENT ADDRESS
007060	053777	001274	172244		BIS	LINBIT, @BAP	. ECHO RECEIVED CHARACTER
007006	022701	001504		PINT3B	CMP	#TUMTAB+176, %1	. CHECK TUMBLE
007012	001002				BNE	+6	. TABLE POINTER
007014	012701	001304			MOI	#TUMTAB-2, %1	
007020	005721				TST	(1)+	

```
007022 000712 BR RINT3A  
007024 042777 0002C 172216 RINT3X BIC #BIT7,@CSR ,CLEAR CHARACTER DONE FLAG  
007032 010137 001300 MOV %1,TTPTR ,RESTORE POINTER  
007036 000240 NOP  
007040 000240 NOP  
007042 000002 RTI ,EXIT  
  
007044 000240 TINT3 NOP  
007046 032777 060000 172174 BIT #BIT14+BIT13,@CSR ,ANY ERROR FLAGS SET  
007054 001401 BEQ +4  
007056 104003 EROR  
007060 042777 160000 172162 BIC #BIT15+BIT14+BIT13,@CSR ,CLEAR ALL FLAGS  
007066 000002 RTI ,EXIT
```



007070 000000  
007234 007234  
007234 000000  
007400 007400  
007400 000000  
007544 007544  
007544 000000  
007710 007710  
007710 000000  
010054 010054  
010054 000000  
010220 010220  
010220 000000  
010364 010364  
010364 000000  
010530 010530  
010530 000000  
010674 010674  
010674 000000  
011040 011040  
011040 000000  
011204 011204  
011204 000000  
011350 011350  
011350 000000  
011514 011514  
011514 000000  
011660 011660  
011660 000000  
012024 012024  
012024 000000  
012170 012170  
012170 000000  
012334 012334  
012334 007234  
012336 007400  
012340 007544  
012342 007710  
012344 010054  
012346 010220  
012350 010364  
012352 010530  
012354 010674  
012356 011040  
012360 011204  
012362 011350  
012364 011514  
012366 011660  
012370 012024  
012372 012170  
012374 000000  
012414 012414  
  
012414 012454  
012416 012460  
012420 012464

OUTBUF 0  
=OUTBUF+100  
LN0BUF 0  
=LN0BUF+100  
LN1BUF 0  
=LN1BUF+100  
LN2BUF 0  
=LN2BUF+100  
LN3BUF 0  
=LN3BUF+100  
LN4BUF 0  
=LN4BUF+100  
LN5BUF 0  
=LN5BUF+100  
LN6BUF 0  
=LN6BUF+100  
LN7BUF 0  
=LN7BUF+100  
LN10BF 0  
=LN10BF+100  
LN11BF 0  
=LN11BF+100  
LN12BF 0  
=LN12BF+100  
LN13BF 0  
=LN13BF+100  
LN14BF 0  
=LN14BF+100  
LN15BF 0  
=LN15BF+100  
LN16BF 0  
=LN16BF+100  
LN17BF 0  
=LN17BF+100  
NTAB LN0BUF  
LN1BUF  
LN2BUF  
LN3BUF  
LN4BUF  
LN5BUF  
LN6BUF  
LN7BUF  
LN10BF  
LN11BF  
LN12BF  
LN13BF  
LN14BF  
LN15BF  
LN16BF  
LN17BF  
CNTTAB 0  
=CNTTAB+16  
  
C IDENT0  
IDENT1  
CENT2

012422	012470		DENT3
012424	012474		IDENT4
012426	012500		IDENT5
012430	012504		IDENT6
012432	012510		IDENT7
012434	012514		IDNT10
012436	012520		IDNT11
012440	012524		IDNT12
012442	012530		IDNT13
012444	012534		IDNT14
012446	012540		IDNT15
012450	012544		IDNT16
012452	012550		IDNT17
012454	105215	IDENT0	CRLF
012456	030060		"00
012460	105215	IDENT1	CRLF
012462	030460		"01
012464	105215	DENT2	CRLF
012466	031060		"02
012470	105215	IDENT3	CRLF
012472	031460		"03
012474	105215	IDENT4	CRLF
012476	032060		"04
012500	105215	IDENT5	CRLF
012502	032460		"05
012504	105215	IDENT6	CRLF
012506	033060		"06
012510	105215	DENT7	CRLF
012512	033460		"07
012514	105215	IDENT10	CRLF
012516	030061		"10
012520	105215	DNT11	CRLF
012522	030461		"11
012524	105215	IDNT12	CRLF
012526	031061		"12
012530	105215	IDNT13	CRLF
012532	031461		"13
012534	105215	IDNT14	CRLF
012536	032061		"14
012540	105215	IDNT15	CRLF
012542	032461		"15
012544	105215	IDNT16	CRLF
012546	033061		"16
012550	105215	DNT17	CRLF
012552	033461		"17
012554	105215		CRLF
	105215	CRLF=105215	

				, MESSAGES	
012556	042045	030515	020061	WHERE	ASCII '%DM11 RECEIVER VECTOR ADDRESS = @'
012564	042522	042503	053111		
012572	051105	053040	041505		
012600	047524	020122	042101		
012606	051104	051505	020123		
012614	020075	100			
012617	045	053523	036522	\$SWREG	ASCII '%SWR= @'
012624	040040				
012626	020040	020040	020040	SVALUE	ASCII 'NEW= @'
012634	020040	020040	042516		
012642	036527	040040			
012646	036445	040		\$CTLU	ASCII '%= '
012651	045	044127	041511	WHICH	ASCII '%WHICH DM11 ARE YOU TESTING @'
012656	020110	046504	030461		
012664	040440	042522	054440		
012672	052517	052040	051505		
012700	044524	043516	040040		
012706	042045	052101	020101	ERDAT	ASCII '%DATA ERR S/B'
012714	051105	020122	051440		
012722	041057	020072			
012726	020040	020040	020040	AASB	ASCII 'WAS '
012734	020040	040527	035123		
012742	040				
012743	040	020040	020040	AWAS	ASCII '@'
012750	020040	100			
012753	114	047111	020105	LINEM	ASCII 'LINE # '
012760	020043				
012762	020040	040040		ALINE	ASCII '@'
012766	052045	050131	020105	MO	ASCII '%TYPE PROGRAM #@'
012774	051120	043517	040522		
013002	020115	040043			
013006	037445	100		M1	ASCII '%?@'
013011	045	042524	052123	M2	ASCII '%TEST DZOMB COMPLETE@'
013016	042040	042132	041115		
013024	041440	046517	046120		
013032	052105	040105			
013036	051445	052105	051440	M3	ASCII '%SET SP OPTIONS NORMAL OPERATION'
013044	020122	050117	044524		
013052	047117	027123	047040		
013060	051117	040515	020114		
013066	050117	051105	052101		
013074	047511	116			
013077	123	020122	020075		ASCII '%SP = 00000 PRESS CONT @'
013104	030060	030060	030060		
013112	050040	042522	051523		
013120	041440	047117	027124		
013126	100				
013127	045	040504	040524	PRGOM	ASCII '%DATA TEST ALL LINES @'
013134	052040	051505	020124		
013142	046101	020114	044514		
013150	042516	020123	100		
013155	045	040504	040524	PRGIM	ASCII '%DATA TEST TRANSMIT ON ALL LINES SIMULTANEOUSLY@'
013162	052040	051505	020124		
013170	051124	047101	046523		

013176	052111	047440	020116			
013204	046101	020114	044514			
013212	042516	020123	044523			
013220	052515	052114	047101			
013226	047505	051525	054514			
013234	100					
013235	045	051124	047101	PRG2M	ASCII	'%TRANSMIT TO TERM NALS@'
013242	046523	052111	052040			
013250	020117	042524	046522			
013256	047111	046101	040123			
013264	042445	044103	020117	PRG3M	ASCII	'%ECHO TEST@'
013272	042524	052123	100			
013277	045	052520	020124	PRG1	ASCII	'%PUT CHAR IN SR(0-7), DELAY IN SR(8-15)@'
013304	044103	051101	044440			
013312	020116	051123	030050			
013320	033455	026051	042504			
013326	040514	020131	047111			
013334	051440	024122	026470			
013342	032461	040051				
013346	052045	050131	020105	POPPAP	ASCII	'%TYPE PARITY OPTION (N=NOT DESIRED O=ODD, E=EVEN)@'
013354	040520	044522	054524			
013362	047440	052120	047511			
013370	020116	047050	047075			
013376	052117	042040	051505			
013404	051111	042105	047440			
013412	047475	042104	020054			
013420	036505	053105	047105			
013426	100					
013427	045	020122		EMO	ASCII	'%R '
013432	020040	020040	041520	ATNUMB	ASCII	'% PC= '
013440	020075					
013442	020040	020040	020040	APC	ASCII	'% @'
013450	100					
013451	015	012		MSG1	BYTE	15 12
013453	040	044124	020105		ASCII	'% THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
013460	052521	041511	020113			
013466	051102	053517	020116			
013474	047506	020130	052512			
013502	050115	042105	047440			
013510	042526	020122	044124			
013516	020105	040514	054532			
013524	042040	043517	020123			
013532	040502	045503	030440			
013540	031462	032464	033466			
013546	034470	060				
	013552					
013552	015	012		MSG2	EVEN BYTE	15 12
013554	177400				177400	
013556	177400				177400	
013560	177400				177400	
013562	177400				177400	
013564	177400				177400	
013566	177400				177400	
013570	177400				177400	
013572	177400				177400	

013574	177400			177400	
013576	177400			177400	
013600	177400			177400	
013602	177400			177400	
013604	177400			177400	
013606	177400			177400	
013610	177400			177400	
013612	177400			177400	
013614	177400			177400	
013616	177400			177400	
013620	177400			177400	
013622	177400			177400	
013624	177400			177400	
013626	177400			177400	
013630	177400			177400	
013632	177400			177400	
013634	177400			177400	
013636	177400			177400	
013640	177400			177400	
013642	177400			177400	
013644	177400			177400	
013646	177400			177400	
013650	177400			177400	
013652	015	012	MSG3	BYTE	15 12
013654	125252			ALTO	
013656	125252			ALTO	
013660	125252			ALTO	
013662	125252			ALTO	
013664	125252			ALTO	
013666	125252			ALTO	
013670	125252			ALTO	
013672	125252			ALTO	
013674	125252			ALTO	
013676	125252			ALTO	
013700	125252			ALTO	
013702	125252			ALTO	
013704	125252			ALTO	
013706	125252			ALTO	
013710	125252			ALTO	
013712	125252			ALTO	
013714	125252			ALTO	
013716	125252			ALTO	
013720	125252			ALTO	
013722	125252			ALTO	
013724	125252			ALTO	
013726	125252			ALTO	
013730	125252			ALTO	
013732	125252			ALTO	
013734	125252			ALTO	
013736	125252			ALTO	
013740	125252			ALTO	
013742	125252			ALTO	
013744	125252			ALTO	
013746	125252			ALTO	
013750	125252			ALTO	
013752	015	012	MSG4	BYTE	15 12

013754	052525			ALT1	
013756	052525			ALT1	
013760	052525			ALT1	
013762	052525			ALT1	
013764	052525			ALT1	
013766	052525			ALT1	
013770	052525			ALT1	
013772	052525			ALT1	
013774	052525			ALT1	
013776	052525			ALT1	
014000	052525			ALT1	
014002	052525			ALT1	
014004	052525			ALT1	
014006	052525			ALT1	
014010	052525			ALT1	
014012	052525			ALT1	
014014	052525			ALT1	
014016	052525			ALT1	
014020	052525			ALT1	
014022	052525			ALT1	
014024	052525			ALT1	
014026	052525			ALT1	
014030	052525			ALT1	
014032	052525			ALT1	
014034	052525			ALT1	
014036	052525			ALT1	
014040	052525			ALT1	
014042	052525			ALT1	
014044	052525			ALT1	
014046	052525			ALT1	
014050	052525			ALT1	
014052	015	012	MEGS	BYTE	15 12
014054	000400			400	
014056	002002			2002	
014060	010010			10010	
014062	040040			40040	
014064	000200			200	
014066	177377			177377	
014070	175775			175775	
014072	167767			167767	
014074	137737			137737	
014076	177500			177500	
014100	000400			400	
014102	002002			2002	
014104	010010			10010	
014106	040040			40040	
014110	000200			200	
014112	177377			177377	
014114	175775			175775	
014116	167767			167767	
014120	137737			137737	
014122	177500			177500	
014124	000400			400	
014126	002002			2002	
014130	010010			10010	
014132	040040			40040	

014134	000200		200	
014136	177377		177377	
014140	175775		175775	
014142	167767		167767	
014144	137737		137737	
014146	177500		177500	
014150	015	012	BYTE	15.12
014152	015	012	BYTE	15.12
	000001		END	



A = 000000	DMPAR 003042	LINE 001646	PRG1EX 006524	RT2 005466
AASB 012726	DMPARB 003326	LINEM 012753	PRG1M 013155	RT2A 005476
ALINE 012762	DTCHK 001664	LN08UF 007234	PRG1R 006364	RT20 006026
ALTO = 125252	EHALT = 104010	LN18UF 007400	PRG2 006532	RT20A 006036
ALT1 = 052525	EMO 013427	LN10BF 010674	PRG2M 013235	RT21 006046
APC 013442	EMTINT 002572	LN11BF 011040	PRG2R 006536	RT21A 006056
APCADD 001642	EMTTAB 001554	LN12BF 011204	PRG3 006542	RT22 006066
ATLAST= 177777	ERDAT 012706	LN13BF 011350	PRG3A 006570	RT22A 006076
ATNUMB 013432	ERR 001732	LN14BF 011514	PRG3M 013264	RT23 006106
AHAS 012743	ERRA 001770	LN15BF 011660	PRG3R 006560	RT23A 006116
BAP 001252	ERRB 002032	LN16BF 012024	PRTY0 = 000000	RT24 006126
BARDAT 001276	ERRC 002040	LN17BF 012170	PRTY1 = 000040	RT24A 006136
BAPIM 001270	ERREX 002060	LN28UF 007544	PRTY2 = 000100	RT25 006146
BASREG 001256	ERRHLT 002056	LN38UF 007710	PRTY3 = 000140	RT25A 006156
BAT 001206	ERROR = 104003	LN48UF 010054	PRTY4 = 000200	RT26 006166
BELL = 000007	ERROR1= 104015	LN58UF 010220	PRTY5 = 000240	RT26A 006176
BIT0 = 000001	ERRVEC= 000004	LN68UF 010364	PRTY6 = 000300	RT27 006206
BIT1 = 000002	ERR1 001750	LN78UF 010530	PRTY7 = 000340	RT27A 006216
BIT10 = 002000	ESCOPE 002356	MANUAL= 100000	PRVCNT 001644	RT3 005506
BIT11 = 004000	FAJAG 004116	MSG1 013451	PSW = 177776	RT3A 005516
BIT12 = 010000	FORWD 002540	MSG2 013552	RCV DAT 001630	RT30 006226
BIT13 = 020000	GETRDY 002246	MSG3 013652	RECD 003604	RT30A 006236
BIT14 = 040000	GTLINB 004774	MSG4 013752	RINT 005134	RT31 006246
BIT15 = 100000	GTPDYA 002274	MSG5 014052	RINT3 006640	RT31A 006256
BIT2 = 000004	GTRDYC 002316	MO 012766	RINT3A 006650	RT32 006266
BIT3 = 000010	GTRDYD 002340	M1 013006	RINT3B 007006	RT32A 006276
BIT4 = 000020	GTPCYX 002254	M2 013011	RINT3X 007024	PT33 006306
BIT5 = 000040	ICTR 001526	M3 013036	RSTART 001544	RT33A 006316
BIT6 = 000100	ID 012414	NONACT 006752	RSTAT1 002130	RT34 006326
BIT7 = 000200	IDENT 004532	NOP = 000240	RSTAT2 002200	RT34A 006336
BIT8 = 000400	IDENTO 012454	NXTST 001524	RSTREG= 104014	RT35 = 177777
BIT9 = 001000	IDENT1 012460	OALNV 004620	RSTPG 002664	RT4 005526
BKCSR 001254	IDENT2 012464	OPEN = 000000	RTNNO 001522	PT4A 005536
BMOVE 004706	IDENT3 012470	OUTBUF 007070	RTD 005426	RT5 005546
CARMSK 001634	IDENT4 012474	OVLAY 003550	RTDA 005436	RT5A 005556
CAT 001106	IDENT5 012500	PASS 001652	RT1 005446	RT6 005566
CC = 177776	IDENT6 012504	PCADD 001640	RT1A 005456	RT6A 005576
CH DAT 005026	IDENT7 012510	POPPAR 013346	RT10 005626	RT7 005606
CLPINT 001260	IDNT10 012514	POPSP = 005726	RT10A 005636	RT7A 005616
CLPLVL 001262	IDNT11 012520	POPSP2= 022626	RT11 005646	SAVREG= 104013
CNT 003762	IDNT12 012524	PRGI 013277	RT11A 005656	SAVRG 002624
CNTLU = 104020	IDNT13 012530	PRGLIM 001532	RT12 005666	SCOPE = 104012
CNTLUU 004040	IDNT14 012534	PRGNUM 002154	RT12A 005676	SCOPEA 002414
CNTTAB 012374	IDNT15 012540	PRGTAB 001534	RT13 005706	SCOPEB 002422
CNTDAT 002064	IDNT16 012544	PRGX 006352	RT13A 005716	SCOPEC 002456
COUNT 001654	IDNT17 012550	PRGO 005366	RT14 005726	SCOPEE 002460
CRLF = 105215	INCRN 001656	PRGDA 005372	RT14A 005736	SCOPEF 001530
CSR 001250	INTAB 012334	PRGDM 013127	RT15 005746	SPBOT 001100
CURTST 001520	IBDIN = 104017	PRGOR 005410	RT15A 005756	SRSET 002214
DATCHK= 104004	IBDINT 003764	PRGI 006346	RT16 005766	SPT 001616
DISPLA 001104	IBSTART 001516	PRGIA 006374	RT16A 005776	STALL = 104002
DISPRE 000174	LINBIT 001274	PRGIC 006504	RT17 006006	START 002112
DLXMT 004200	LINBUF 001650	PRGID 006520	RT17A 006016	STPRM = 104006

STTV = 104007	TKDBR 001622	TTPTR 001300	VECOKB 003264	XMTLVL 001266
SUSWR = 104016	TKLVL 001510	TUMTAB 001306	VECTOR 003230	Y = 000000
SUSWR 004120	TKVTR 001506	TYP 002726	WCT 001146	SCTLU 012646
SWR 001102	TMP1 004076	TYPE = 104000	WHERE 012556	SENDAD 002530
SWREG 000176	TPCSP 001624	TYPES = 104001	WHICH 012651	SSWREG 012617
TEMP 001636	TPDBR 001626	UNIT 003346	X = 000034	SVALUE 012626
TINT 005326	TPLVL 001514	VAC 001246	XMITD 004730	= 014154
TINT3 007044	TPVTR 001512	VECOK 003246	XMTDAT 001632	
TRISP 001620	TTDAT 001272	VECOKA 003256	XMTINT 001264	

AB 014154 000

ERRORS DETECTED 0

DDMMDD DZOMB SEQ NL SEQ=DZOMB P11  
FIN TIME 13 0 SECONDS  
FIN TIME RATIO 418 5=70 0  
CORE USED 51 9 PAGES

