

IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DIKLA-B-D
PRODUCT NAME: KLB-JA & KLB-KA/KB/KC/KD
LOOP BACK TEST
DATE CREATED: AUGUST, 1974
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR(S): BRUCE HANSEN & R. MOORE

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 ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973 & 1974, DIGITAL EQUIPMENT CORP., MAYNARD, MASS;

TABLE OF CONTENTS

1. ABSTRACT

2. REQUIREMENTS

2.1 HARDWARE

2.2 STORAGE

2.3 PREREQUISITE SOFTWARE

3. LOADING PROCEDURE

3.1 METHOD

4. STANDARD TEST PROCEDURE

4.1 CONNECTIONS FOR TESTING

4.1.1 EIA LOOP BACK CONNECTIONS

4.1.2 20MA LOOP BACK CONNECTIONS

4.2 RUN CONTROL/DATA TEST

4.2.1 TELETYPE INTERROGATION

4.3 RESTARTING THE PROGRAM

4.4 BAUD RATE TIMING TEST

5. OPERATING PROCEDURES

5.1 STARTING ADDRESSES

5.2 SWITCH REGISTER CONTROL

6. PROGRAM AND/OR OPERATOR ACTION

6.1 NORMAL HALTS

7. ERRORS

7.1 CONTROL/DATA TEST ERRORS

7.1.1 CONTROL/DATA TEST ERROR RECOVERY

7.2 BAUD RATE TIMING TEST ERRORS

7.2.1 BAUD RATE TIMING TEST ERROR RECOVERY

8. PROGRAM DESCRIPTION

8.1 CONTROL/DATA TEST

8.2 BAUD RATE TIMING TEST

9. LISTING

1. ABSTRACT

KLB-JA & KLB-KA/KB/KC/KD LOOP BACK TEST IS A PROGRAM TO CHECK OUT THE TERMINAL CONTROL/DATA INTERFACE OPTION (M8655); THE BOARD IS TESTED IN LOOP BACK MODE BY CONNECTING EITHER EIA OUTPUT TO EIA INPUT OR CONNECTING 20 MA CURRENT OUTPUT TO 20 MA CURRENT INPUT;

2. REQUIREMENTS

2.1 HARDWARE

PDP-8, 8I, 8L WITH A DW8E-P OR DW8E-N BUS CONVERTER

PDP-8E, F OR M

KLB-JA TERMINAL CONTROL/DATA INTERFACE (M8655 9 BAUDS RATES) OR A KLB-KA (SAME AS THE KLB-JA) OR A

KLB-KB TERMINAL CONTROL/DATA INTERFACE (M8655-YA 1050 BAUD) OR A

KLB-KC TERMINAL CONTROL/DATA INTERFACE (M8655-YB 66,7 BAUD) OR A

KLB-KD TERMINAL CONTROL/DATA INTERFACE (M8655-YC 56,8 BAUD)

2.2 STORAGE

THE PROGRAM OCCUPIES MEMORY LOCATIONS 0000 TO 5400

2.3 PREREQUISITE SOFTWARE

THE SYSTEM MUST BE CAPABLE OF RUNNING ALL BASIC PROCESSOR DIAGNOSTICS;

3.0 LOADING PROCEDURE

3.1 METHOD

THE PROGRAM IS LOADED USING THE STANDARD BINARY LOADER TECHNIQUE, AND THE PROGRAM MUST RESIDE IN FIELD 0;

4.0 STANDARD TEST PROCEDURE

4.1 CONNECTIONS FOR TESTING

4.1.1 EIA LOOP BACK CONNECTIONS

CONNECT PIN F TO PIN J AND PIN E TO PIN M ON THE BERG CONNECTOR

4.1.2 20MA LOOP BACK CONNECTIONS

CONNECT PIN E TO PIN H, PIN K TO PIN KK, AND PIN S TO PIN AA ON THE BERG CONNECTOR

```

////////////////////////////////////
/ WARNING! 20MA LOOP CONNECTIONS CAN ONLY BE CONNECTED /
/ THIS WAY FOR TESTING IN LOOP BACK MODE. DO NOT ATTEMPT /
/ TO CONNECT 2 M8655'S TOGETHER AT ANY TIME WITH 20MA LOOPS /
////////////////////////////////////

```

4.2 RUN CONTROL/DATA TEST

- A. DO EITHER STEP 4.1.1 OR 4.1.2 FOR EIA OR 20MA LOOP BACK CONNECTIONS
- B. THE PROGRAM WHEN FIRST LOADED IS INITIALIZED FOR 110 BAUD, 2 STOP BITS, 8 DATA BITS, RECEIVE IOT OF 03 AND A TRANSMIT IOT OF 04. IF THIS IS THE CONFIGURATION DESIRED GO TO PARAGRAPH 4.3 (RESTARTING THE PROGRAM) OTHERWISE GO TO STEP C
- C. THE PROGRAM CAN BE INITIALIZED EITHER OF TWO WAYS:
 1. BY WAY OF THE SWITCH REGISTER OR
 2. BY WAY OF AN OPTIONAL TELETYPE WITH DEVICE CODE OF 03 AND 04. THESE DEVICE CODES CANNOT BE CHANGED.
- D. SET SWITCH REGISTER TO 0200 AND PRESS "LOAD ADDRESS".
- E. SET SR11=0 FOR INITIALIZING THE PROGRAM WITH THE SR OR SET SR11=1 FOR INITIALIZING THE PROGRAM WITH THE TELETYPE AND PRESS "CLEAR" AND THEN "CONTINUE".
- F. IF SR11=0 GO TO G, IF SR11=1 GO TO 4.2.1 FOR TELETYPE INTERROGATION.

- G. THE PROGRAM WILL HALT AT LOCATION 0207.
- H. SET SR 0-5 TO THE RECEIVE IOT AND SR 6-11 TO THE TRANSMIT IOT AND PRESS "CONTINUE".
- I. THE PROGRAM WILL HALT AT LOCATION 0212.
- J. SETUP THE SWITCH REGISTER FOR THE FOLLOWING CONDITION AND PRESS "CONTINUE".

NOTE: A 1 FOLLOWING A JUMPER OR SWITCH MEANS THAT THE JUMPER IS INSERTED OR A SWITCH IS IN THE ON POSITION.

- SR0=1 IF PARITY JUMPER IS INSTALLED NP=1
- SR1=1 IF STATUS ENABLE JUMPER IS INSTALLED SWD=1
- SR2=1 IF FILLER CHARACTER JUMPER INSTALLED FIL=1

SR5	SR6	SR7	SR8	BAUD RATE	ROCKER SWITCHES	JUMPERS	VARIATION
0	0	0	0	110	BAUD B1=0 B2=0	B3=0 W2=1 W3=0	KL8=JA & KA
0	0	0	1	150	BAUD B1=0 B2=0	B3=1 W2=1 W3=0	KL8=JA & KA
0	0	1	0	300	BAUD B1=0 B2=1	B3=0 W2=1 W3=0	KL8=JA & KA
0	0	1	1	600	BAUD B1=0 B2=1	B3=1 W2=1 W3=0	KL8=JA & KA
0	1	0	0	1200	BAUD B1=1 B2=0	B3=0 W2=1 W3=0	KL8=JA & KA
0	1	0	1	2400	BAUD B1=1 B2=0	B3=1 W2=1 W3=0	KL8=JA & KA
0	1	1	0	4800	BAUD B1=1 B2=1	B3=0 W2=1 W3=0	KL8=JA & KA
0	1	1	1	9600	BAUD B1=1 B2=1	B3=1 W2=1 W3=0	KL8=JA & KA
1	0	0	0	19.2K	BAUD B1=1 B2=1	B3=1 W2=0 W3=1	KL8=JA & KA
1	0	0	1	56.8	BAUD B1=0 B2=0	B3=0 W2=1 W3=0	KL8=KD (M8655=YC)
1	0	1	0	66.7	BAUD B1=0 B2=0	B3=0 W2=1 W3=0	KL8=KC (M8655=YB)
1	0	1	1	1050	BAUD B1=1 B2=0	B3=0 W2=1 W3=0	KL8=KB (M8655=YA)

*19.2 KILO BAUD IS ONLY OBTAINABLE WITH A SPECIAL UART.

- SR9=1 IF TWO STOP BITS=JUMPER NOT INSTALLED SB=0

SR10	SR11	#	DATA BITS/CHARACTER	JUMPERS
0	0	5	DATA BITS/CHARACTER	NB1=1 NB2=1
0	1	6	DATA BITS/CHARACTER	NB1=0 NB2=1
1	0	7	DATA BITS/CHARACTER	NB1=1 NB2=0
1	1	8	DATA BITS/CHARACTER	NB1=0 NB2=0

- K. THE PROGRAM WILL HALT AT LOCATION 0245.
- L. SET SWITCH REGISTER TO 0000 IF PROCESSOR IS ONE OF THE PDP-8E FAMILY OTHERWISE SET IT TO 0002 AND PRESS "CONTINUE".
- M. SETTING THE SWITCH REGISTER TO 0200 WHILE RUNNING WILL HALT THE PROCESSOR AT THE COMPLETION OF A PROGRAM PASS AT LOCATION 2327
- N. THE PROGRAM WILL NOW RUN UNTIL AN ERROR IS ENCOUNTERED OR THE PROGRAM IS STOPPED BY THE OPERATOR OR SR4=1.

NOTE: THIS SECTION OF PROGRAM WAS ENTERED FROM STEPS
D, E AND F OF PARAGRAPH 4.2.

- A. THE PROGRAM WILL TYPE RECEIVE IOT?
THE OPERATOR TYPES IN THE DEVICE CODE OF THE RECEIVER(2 NUMBERS)
- B. THE PROGRAM WILL TYPE TRANSMIT IOT?
THE OPERATOR TYPES IN THE DEVICE CODE OF THE TRANSMITTER(2 NUMBERS)
- C. THE PROGRAM WILL TYPE PARITY(Y OR N)?
IF NP JUMPER IS INSTALLED TYPE Y IF IT ISN'T TYPE N,
THE PROGRAM WILL THEN TYPE NP=1? IF ANSWER WAS YES, AND NP=0?
IF ANSWER WAS NO, NP=THE PARITY JUMPER 1=INSTALLED 0=NOT INSTALLED,
THE PROGRAM WILL THEN TYPE EVEN PARITY EVN=0? ODD PARITY EVN=1?
EVN= ODD OR EVEN PARITY JUMPER, 1= JUMPER INSTALLED 0= NOT INSTALLED.
- D. THE PROGRAM WILL THEN TYPE STATUS ENABLED(Y OR N)?
IF SWD JUMPER IS INSTALLED TYPE Y IF NOT TYPE N
THE PROGRAM WILL THEN TYPE SWD=1? IF ANSWER WAS YES, AND SWD=0? IF
ANSWER WAS NO, SWD=STATUS WORD ENABLE JUMPER, 1=JUMPER
INSTALLED, 0=JUMPER NOT INSTALLED.
- E. THE PROGRAM WILL THEN TYPE FILLER CHARACTERS(Y OR N)?
IF FIL JUMPER IS INSTALLED TYPE Y IF NOT TYPE N,
THE PROGRAM WILL THEN TYPE FIL=1? IF ANSWER WAS YES, AND
FIL=0? IF ANSWER WAS NO, FIL=FILLER CHARACTER JUMPER,
1= JUMPER INSTALLED AND 0= JUMPER NOT INSTALLED.
- F. THE PROGRAM WILL NOW TYPE OUT THE FOLLOWING MESSAGE
BAUD RATE(00=15)? 00=110 01=150 02=300 03=600 04=1200 05=2400
06=4800 07=9600 10=19,200 11=36.8 12=66.7 13=1050
THE OPERATOR WILL NOW TYPE IN TWO NUMBERS AND THE PROGRAM
WILL RESPOND WITH THE FOLLOWING MESSAGE:

XXXX BAUD = B1=Y? B2=Y? B3=Y W2=Z W5=Z

(XXXX IS THE BAUD RATE FROM 36.8 TO 19,200 BAUD
Y=0 OR 1 0=SWITCH IN OFF POSITION 1= SWITCH IN ON POSITION
Z=0 OR 1 0=JUMPER NOT INSTALLED 1=JUMPER INSTALLED).
- G. THE PROGRAM WILL NOW TYPE TWO STOP BITS(Y OR N)?
IF SB JUMPER IS NOT INSTALLED TYPE Y IF IT IS TYPE N,
THE PROGRAM WILL THEN TYPE SB=0? IF ANSWER WAS YES,
AND SB=1 IF ANSWER WAS NO, SB=STOP BIT JUMPER,
1=JUMPER INSTALLED 0=JUMPER NOT INSTALLED
- H. THE PROGRAM WILL THEN TYPE DATA BITS/CHARACTER? 0=5 1=6 2=7 3=8
THE OPERATOR WILL NOW TYPE IN ONE NUMBER AND THE PROGRAM
WILL RESPOND WITH THE FOLLOWING MESSAGE:
X DATA BITS=N1=Y? N2=Y?
X=THE NUMBER OF DATA BITS SELECTED 5,6,7 OR 8
Y=0 OR 1 0=JUMPER NOT INSTALLED 1=JUMPER INSTALLED

- I. THE PROGRAM WILL HALT AT LOCATION 0245.
- J. SET SWITCH REGISTER TO 0000 IF PROCESSOR IS ONE OF THE PDP-8E FAMILY OTHERWISE SET IT TO 0002 AND PRESS "CONTINUE".
- K. SETTING THE SWITCH REGISTER TO 0200 WHILE RUNNING WILL HALT THE PROCESSOR AT THE COMPLETION OF A PROGRAM PASS AT LOCATION 2327
- L. THE PROGRAM WILL NOW RUN UNTIL AN ERROR IS ENCOUNTERED OR THE PROGRAM IS STOPPED BY OPERATOR OR SR4=1.

4.3

RESTARTING THE PROGRAM

- A. SET SR TO 0201 AND PRESS LOAD ADDRESS
- B. SET SR TO ALL ZEROES IF PROCESSOR IS ONE OF THE PDP-8E FAMILY OTHERWISE SET SR TO 0002 AND PRESS "CLEAR" AND THEN "CONTINUE"
- C. SETTING SR4 TO A ONE WILL HALT THE PROGRAM AFTER ONE COMPLETE PROGRAM PASS AT LOCATION 2327
- D. THE PROGRAM WILL NOW RUN UNTIL AN ERROR IS ENCOUNTERED OR THE PROGRAM IS HALTED BY THE OPERATOR OR SR4=1.

4.4

RUN BAUD RATE TIMING TEST

- A. DO STEP A-K OF PARAGRAPH 4.2 IF NOT ALREADY DONE
- B. THIS TEST IS A 30 SECOND STOP WATCH TIMING TEST
- C. SET SR TO 0202 AND PRESS "LOAD ADDRESS" THEN "CLEAR".
- D. CHECK STOP WATCH AND PRESS "CONTINUE".
- E. THE PROGRAM SHOULD HALT IN APPROXIMATELY 30 SECONDS AT LOCATION 2517 (SEE NOTE FOR EXCEPTION) IF THE BAUD RATE WAS SETUP CORRECTLY.

NOTE: THE PROGRAM WILL HALT IN APPROXIMATELY 28 SECONDS FOR THE FOLLOWING CONDITIONS:

5 DATA BITS, 2 STOP BITS, AND NO PARITY

5. OPERATING PROCEDURES
-----5.1 STARTING ADDRESSES

200 WITH SR11=0 - INITIALIZE THE PROGRAM BY THE SWITCH REGISTER
 200 WITH SR11=1 - INITIALIZE THE PROGRAM BY THE TELETYPE
 201 RESTART ADDRESS; NO INITIALIZATION NEEDED
 202 BAUD RATE TIMING TEST

5.2 SWITCH REGISTER CONTROL

SR	STATE	ACTION
0	1	DO NOT HALT ON ERROR
1	1	LOOP ON ERROR OR ON A CONSTANT DATA PATTERN
2	1	LOOP ON TEST SEQUENCE
4	1	HALT PROGRAM AFTER A COMPLETE PROGRAM PASS
10	1	PROCESSOR NOT OF THE PDP-8E FAMILY
11	0	INITIALIZE THE PROGRAM WITH SR (STARTING ADDRESS 200 ONLY)
11	1	INITIALIZE THE PROGRAM WITH TELETYPE (STARTING ADDRESS 200 ON

6. PROGRAM AND/OR OPERATOR ACTION
-----6.1 NORMAL HALTS

0207 INITIALIZATION OF PROGRAM HALT - SET DEVICE CODES IN THE SR.
 0212 INITIALIZATION OF PROGRAM HALT - SETUP THE FOLLOWING CONDITIONS
 OF JUMPERS AND SWITCHES IN THE SWITCH REGISTER: PARITY
 STATUS ENABLE, FILLER CHARACTERS, BAUD RATE, NUMBER OF
 STOP BITS, AND NUMBER OF DATA BITS/CHARACTER
 0245 SETUP THE SR OPTIONS FOR RUNNING THE PROGRAM
 2327 END OF CONTROL/DATA TEST - SR4=1
 2517 END OF BAUD RATE TIMING TEST HALT

7. ERRORS

7.1 CONTROL/DATA TEST ERRORS

ALL ERRORS DETECTED BY THE PROGRAM WILL RESULT IN AN ERROR HALT, REFER TO THE PROGRAM LISTING FOR THE CAUSE OF THE ERROR,

7.1.1 CONTROL/DATA TEST ERROR RECOVERY

SET SWITCH REGISTER 0,1 AND 2 TO A 1 AND PRESS "CONTINUE", THERE MAY BE 1 OR 2 MORE ERROR HALTS, IF THE ERROR WAS A DATA ERROR, THE PROGRAM IS NOW IN A SCOPE LOOP,

7.2 BAUD RATE TIMING TEST ERRORS

THE OPERATOR MUST DETECT ANY ERRORS IN THE BAUD RATE TIMING TEST, ONCE STARTED THE PROGRAM SHOULD HALT IN APPROXIMATELY 30 SECONDS EXCEPT WHEN THE MODULE IS SET UP FOR 9 DATA BITS, 2 STOP BITS AND NO PARITY, THE PROGRAM WILL THEN HALT IN APPROXIMATELY 28 SECONDS, ANY DEVIATIONS OF MORE THAN A 1/2 SECOND IS AN ERROR,

7.2.1 BAUD RATE TIMING TEST ERROR RECOVERY

AFTER CHECKING THE MODULE TO BE SET UP CORRECTLY, RESTART THE TEST BY SETTING SR2=1 AND PRESSING "CONTINUE",

IF ERROR STILL EXISTS GO TO PARAGRAPH 4.4 AND DO EACH AND EVERY STEP AGAIN,

IF ERROR STILL EXISTS CHECK THE BAUD RATE WITH A SCOPE,

8. PROGRAM DESCRIPTION

8.1 CONTROL/DATA TEST

THE FIRST TEST (CLRBRD) ISSUES A CAF INSTRUCTION TO GENERATE AN INITIALIZE PULSE. THE PROGRAM THEN CHECKS THAT THE TRANSMIT AND RECEIVE FLAGS ARE NOT STUCK ON AND THAT KSF,TSF, AND SPI DON'T SKIP. THE PROGRAM ALSO CHECKS THAT INTERRUPT REQUEST LINE IS NOT PULLED LOW. ALL ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406. THE CONTENTS OF THE AC WILL CONTAIN THE ADDRESS WHERE THE ERROR WAS DETECTED.

THE NEXT TEST (SOXMIT) CHECKS THAT THE TRANSMIT FLAG CAN BE SET AND CLEARED BY TFL,TSF AND TCF. THE RECEIVE FLAG IS ALSO CHECKED TO BE 0. KCF,TFL,TCF,KSF ARE CHECKED NOT TO SKIP. TSF IS CHECKED TO SKIP AND NOT TO SKIP. ALL ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406 WITH THE AC EQUAL TO THE PC WHERE THE ERROR WAS DETECTED.

THE NEXT TEST (CAFXTM) CHECKS THAT THE TRANSMIT FLAG CAN BE CLEARED BY CAF AND THAT THE RECEIVE FLAG IS STILL 0. ALL ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406 WITH THE AC EQUAL TO THE PC WHERE THE ERROR WAS DETECTED.

THE NEXT TEST (INTXMT) USES THE TRANSMIT FLAG TO CHECK THAT INTERRUPT ENABLE CAN BE SET AND CLEARED AND THAT THE PROGRAM CAN INTERRUPT. INTERRUPT ENABLE IS SET AND CLEARED BY DATA BIT 11 AND THE KIE COMMAND. SPI IS CHECKED TO SKIP AND NOT TO SKIP AND THE PROGRAM ALSO CHECKS THE MODULE TO INTERRUPT AND NOT TO INTERRUPT. AT THE END OF THE TEST THE RECEIVE FLAG IS CHECKED TO BE A 0. ALL ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406 WITH THE CONTENTS OF THE AC EQUAL TO THE PC WHERE THE ERROR WAS DETECTED.

THE NEXT TEST (CAFINT) CHECKS THAT CAF WILL SET INTERRUPT ENABLE BY USING THE TRANSMIT FLAG TO SKIP AND INTERRUPT ON. ALL ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406 WITH THE AC EQUAL TO THE PC WHERE THE ERROR WAS DETECTED.

THE NEXT TEST (ACNSKP) CHECKS THE EFFECT OF THE IOT ON THE AC AND ALSO CHECKS THAT THE IOT'S DO NOT SKIP; TPC AND TFS ARE NOT TESTED; AN ERROR HALT AT LOCATION 2423 INDICATES THAT AN IOT SKIPPED THAT SHOULDN'T, THE AC CONTAINS THE PC WHERE THE ERROR WAS DETECTED; AN ERROR HALT AT LOCATION 2442 INDICATES THAT THE IOT AFFECTED THE CONTENTS OF THE AC; THE CONTENTS OF THE AC EQUALS THE PC WHERE THE ERROR WAS DETECTED; PRESSING CONTINUE WILL RESULT IN AN ERROR HALT AT LOCATION 2448 WITH THE AC EQUAL TO THE BITS THAT WERE EFFECTED BY THE IOT;

THE NEXT TEST (STFLGS) CHECKS THAT THE TRANSMIT FLAG CAN BE SET BY TPC AND THAT SOMETIME AFTER THE TRANSMIT FLAG IS SET THE RECEIVE FLAG WILL GET SET BY DATA AVAILABLE; THE PROGRAM CHECKS THAT FLAGS CAN CAUSE AN INTERRUPT AND NOT TO INTERRUPT BY SETTING AND CLEARING INTERRUPT ENABLE; THE PROGRAM CHECKS THAT TCF AND KCC WILL CLEAR THE FLAGS; ALL ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406 WITH THE AC EQUAL TO THE PC WHERE THE ERROR WAS DETECTED; WHEN LOOPING ON THE ERROR, THE PROGRAM WILL DELAY APPROXIMATELY 200MS AT THE BEGINNING OF EACH LOOP TO ALLOW TIME FOR THE FLAGS TO SETTLE;

THE NEXT TEST (XMTREC) CHECKS THAT A TPC COMMAND WILL SET THE TRANSMIT FLAG AND THAT A TFS COMMAND WILL CLEAR THE FLAG AND THEN RESET IT; THE TEST ALSO CHECKS THAT THE RECEIVE FLAG WILL GET SET FROM THE RESULT OF A TPC AND TFS COMMAND AND THAT THE RECEIVE FLAG CAN BE CLEARED BY A KRB AND KCC OR KCF COMMAND; ALL ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406 WITH THE AC EQUAL TO THE PC WHERE THE ERROR WAS DETECTED; IF SCOPE LOOPING, THERE WILL BE A 200MS DELAY AT THE BEGINNING OF EACH LOOP TO ALLOW THE FLAGS TO SETTLE;

THE NEXT 7 TESTS (SDTST1 TO 7) ARE SIMPLE DATA TESTS; THE PROGRAM TRANSMITS ONE WORD AND THEN WAITS IN A LOOP FOR THE TRANSMIT FLAG OR RECEIVE FLAG TO SET; WHEN THE TRANSMIT FLAG IS SET THE PROGRAM CLEARS IT AND THEN WAITS FOR THE RECEIVE FLAG; WHEN THE RECEIVE FLAG GETS SET, THE PROGRAM COMPARES THE WORD TRANSMITTED WITH THE WORD RECEIVED AND IF THEY DON'T COMPARE THE PROGRAM HALTS AT LOCATION 1560 WITH THE WORD TRANSMITTED IN THE AC; PRESSING "CONTINUE" WILL RESULT WITH AN ERROR HALT AT LOCATION 1563 WITH THE AC EQUAL TO THE WORD READ; ALL OTHER ERRORS WILL RESULT WITH A HALT AT LOCATION 2406 WITH THE AC CONTAINING THE PC WHERE THE ERROR WAS DETECTED

THE NEXT TEST (FDATAT) IS A FASTER DATA TEST USING RANDOM DATA. THE PROGRAM TRANSMITS THE FIRST WORD AND THEN WAITS IN A LOOP FOR THE TRANSMIT OR RECEIVE FLAG TO SET. WHEN THE TRANSMIT FLAG GETS SET A NEW WORD IS THEN GENERATED AND TRANSMITTED. THE PROGRAM THEN WAITS IN THE LOOP AGAIN FOR THE RECEIVE FLAG TO SET AND THEN DATA IS COMPARED WITH THE FIRST WORD TRANSMITTED. THE DIFFERENCE BETWEEN THIS TEST AND SOTST IS THAT THE PROGRAM IS TRANSMITTING 1 WORD AHEAD OF WHAT IT IS READING. IF AN ERROR OCCURS THE PROGRAM WILL HALT AT LOCATION 1653 WITH THE AC EQUAL TO THE WORD EXPECTED. PRESS "CONTINUE" AND THE PROGRAM WILL HALT AT LOCATION 1656 WITH THE WORD RECEIVED IN THE AC. PRESS "CONTINUE" AGAIN AND THE PROGRAM WILL HALT AT LOCATION 1661 WITH THE AC EQUAL TO THE NEW WORD TRANSMITTED. THIS WORD MAY BE THE SAME AS THE WORD EXPECTED DEPENDING WHERE THE ERROR WAS DETECTED. WHEN SCOPE LOOPING ON THIS ERROR, THE FIRST AND THIRD ERROR HALT WORDS WILL BE THE WORDS USED TO TRANSMIT. WHEN AN ERROR IS ENCOUNTERED DURING THIS SCOPE LOOP, THE PROGRAM DELAYS 200MS TO ALLOW FLAGS TO SETTLE BEFORE TRANSMITTING AGAIN. THERE ARE NO ERROR HALTS IN THE SCOPE LOOP. ALL OTHER ERRORS WILL RESULT WITH AN ERROR HALT AT LOCATION 2406 WITH THE AC CONTAINING THE PC WHERE THE ERROR WAS DETECTED.

THE NEXT TEST (CHARLG) CHECKS THAT THE OPERATOR HAS SELECTED THE CORRECT NUMBER OF DATA BITS. THE PROGRAM TRANSMITS A 377 AND THEN TAKES THE 1'S COMPLEMENT OF THE NUMBER OF DATA BITS THE OPERATOR HAD SET UP THE PROGRAM WITH AND COMPARES IT TO READ. IF THE AC EQUATED ZERO AFTER THE COMPARISON, THE NUMBER OF DATA BITS WERE SELECTED CORRECTLY. OTHERWISE, THE PROGRAM WILL HALT AT LOCATION 2027 WITH THE AC CONTAINING THE BITS THAT WEREN'T SUPPOSED TO BE SELECTED. PRESS "CONTINUE" AND THE PROGRAM WILL HALT AT LOCATION 2032 WITH THE AC EQUAL TO THE BITS THE OPERATOR HAD INITIALIZED THE PROGRAM WITH. ALL OTHER ERRORS WILL RESULT WITH AN ERROR HALT AT LOCATION 2406 WITH THE AC CONTAINING THE PC WHERE THE ERROR WAS DETECTED.

THE NEXT TEST (FILERT) IS A FILLER CHARACTER TEST AND WILL ONLY BE DONE IF THE OPERATOR HAS INITIALIZED THE PROGRAM FOR FILLER CHARACTERS. THE PROGRAM TRANSMITS A LINE FEED AND CHECKS THAT 4 RECEIVE FLAGS GET SET BEFORE THE TRANSMIT FLAG AND THAT THE 5TH RECEIVE FLAG GETS SET AFTER THE TRANSMIT FLAG. THE DATA RECEIVED SHOULD BE 1 WORD OF LINE FEED AND 4 WORDS OF FILLER CHARACTERS. IF THE WORD EXPECTED DOESN'T EQUAL THE WORD RECEIVED THE PROGRAM WILL HALT AT LOCATION 2121 WITH THE AC CONTAINING THE WORD EXPECTED. PRESS "CONTINUE" AND THE PROGRAM WILL HALT AT LOCATION 2124 WITH THE AC CONTAINING THE WORD RECEIVED. SCOPE LOOPING ON THIS ERROR WILL RESULT IN A 200MS DELAY AT THE BEGINNING OF EACH ERROR TO ALLOW TIME FOR THE FLAGS TO SETTLE. ALL OTHER ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406 WITH THE AC EQUAL TO THE PC WHERE ERROR WAS DETECTED.

THE LAST TEST (STENAB) IS A STATUS ENABLE TEST AND WILL ONLY BE EXECUTED IF THE OPERATOR HAD SET THE STATUS ENABLE BIT TO A ONE WHEN HE INITIALIZED THE PROGRAM. THIS TEST WILL CHECK THAT THE ERROR BIT AND THE OVERRUN BIT CAN BE SET AND CLEARED IN THE STATUS REGISTER. THE TEST WILL CHECK THAT STATUS ENABLE F/F CAN BE SET AND CLEARED. THE RECEIVE BUFFER WILL BE CHECKED TO CONTAIN THE CORRECT WORD. THREE WORDS (1,2+3) WILL BE TRANSMITTED AND THEN THE RECEIVE BUFFER WILL BE CHECKED. IF THERE WAS AN ERROR DURING COMPARISON OF DATA THE PROGRAM WILL HALT AT LOCATION 2307 WITH THE AC EQUAL TO THE PC WHERE THE ERROR WAS DETECTED. PRESS "CONTINUE" AND THE PROGRAM WILL HALT AT LOCATION 2312 WITH THE WORD EXPECTED IN THE AC. PRESS "CONTINUE" AGAIN AND THE PROGRAM WILL HALT AT LOCATION 2315 WITH THE WORD RECEIVED IN THE AC. ALL OTHER ERRORS WILL RESULT IN AN ERROR HALT AT LOCATION 2406 WITH THE PC WHERE THE ERROR WAS DETECTED IN THE AC. SCOPE LOOPING ON THIS ERROR CAUSES THE PROGRAM TO DELAY 200MS BEFORE STARTING TEST OVER TO ALLOW FLAGS TIME TO SETTLE.

END OF TEST-START TEST OVER AT CLRBRD IF SR4=0 OTHERWISE HALT AT LOCATION 2327.

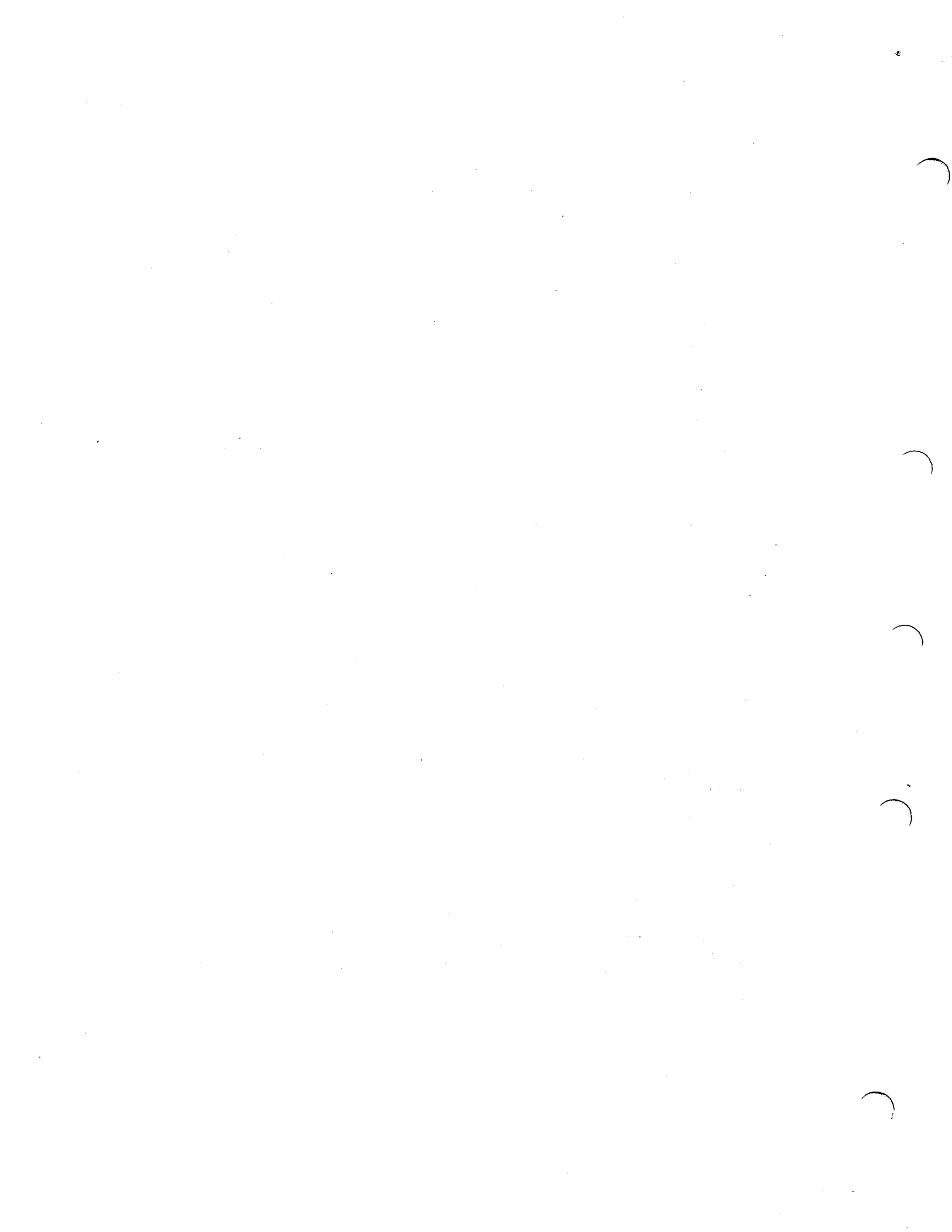
8.2

BAUD RATE TIMING TEST

BAUD RATE TIMING TEST IS A STOP WATCH TIMING TEST. ONCE THIS TEST (BAUDTM) HAS BEEN STARTED, THE PROGRAM TURNS THE INTERRUPT ON AND TRANSMITS A CALCULATED NUMBER OF CHARACTERS (DETERMINED FROM THE BAUD RATE, PARITY, NUMBER OF DATA BITS/CHARACTER AND NUMBER OF STOP BITS). THE PROGRAM SHOULD HALT AT LOCATION 2517 IN 30 SECONDS. EXCEPTIONS TO THIS ARE: ANY BAUD RATE, NO PARITY, 5 DATA BITS/CHARACTER AND 2 STOP BITS. THE PROGRAM IN THIS CASE SHOULD HALT IN 28 SECONDS.

9.

LISTING



/KL8=JA & KL8=KA/KB/KC/KD LOOP BACK TEST;
/MAINDEC=08=0IKLA=B=L
/COPYRIGHT (C) 1973,1974 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS., 01754
/PROGRAMMER: BRUCE HANSEN (KL8=K MODIFICATIONS = R, MOORE)
/

5000 BASEA=5000
6007 CAF=6007
7402 HLT=7402

/RECEIVE IOTS FOR KL8=JA,KA,KB,KC & KD

6030 KCF=6030 /CLEAR RECEIVE FLAG,DON'T SET READER RUN F/P
6031 KSF=6031 /SKIP ON RECEIVE FLAG
6032 KCO=6032 /CLEAR AC AND RECEIVE FLAG AND SET READER RUN
6034 KRS=6034 /READ RECEIVE BUFFER AND STATUS
6035 KIE=6035 /AC 11=1 SET INTERRUPT ENABLE
/AC 11 = 0 CLEAR INTERRUPT ENABLE F/P
6035 KSE=KIE /AC10=1 SET STATUS WORD ENABLE
/AC10=0 CLEAR STATUS WORD ENABLE
6036 KR3=6036 /CLEAR AC AND RECIEVE FLAG,SET READER RUN AND
/READ RECEIVE BUFFER AND STATUS

/TRANSMIT IOTS FOR KL8=JA,KA,KB,KC & KD

6040 TFL=6040 /SET THE TRANSMIT FLAG
6041 TSP=6041 /SKIP ON TRANSMIT FLAG
6042 TCF=6042 /CLEAR THE TRANSMIT FLAG
6044 TPC=6044 /LOAD TRANSMIT BUFFER AND TRANSMIT
6045 SPI=6045 /SKIP IF TRANSMIT OR RECEIVE FLAG IS SET AND
/INTERRUPT ENABLE FLIP/FLOP IS SET
6046 TIS=6046 /LOAD TRANSMIT BUFFER,TRANSMIT AND CLEAR TRANSMIT FLAG

/SWITCH REGISTER SETTINGS

/SR0=1 DON'T HALT ON ERROR
/SR1=1 LOOP ON ERROR OR DATA PATTERN
/SR2=1 LOOP ON TEST SEQUENCE
/SR10=1 PROCESSOR NOT A PDP-8E
/SR11=0 STARTING ADDRESS 200 ONLY=SETUP PROGRAM LIMITS
/ BY WAY OF THE SWITCH REGISTER
/SR11=1 STARTING ADDRESS 200 ONLY=SETUP PROGRAM LIMITS
/ BY WAY OF THE TELETYPE (DEVICE CODES OF 03&04)

/SUBROUTINE CALLS

4577 DELAY=JMS I CXDELAY
4576 STLPPC=JMS I CXPORET
4575 LOOP=JMS I CXSR2
4574 EHLLP=JMS I EHLTLP
4573 SH10NE=JMS I CNOT8E
4572 TSFSKP=JMS I CWATSF
4571 KSFSKP=JMS I CWATKSF
4570 LISN=JMS I EXLISN
4567 MESSAGE=JMS I CMESAGX
4566 ONEOCT=JMS I CONEOCK
4565 THOOCT=JMS I CTHOOCK
4564 FOROCT=JMS I CFOROCK
4563 PRNT1=JMS I CXPRNT1
4562 PRNT2=JMS I CXPRNT2
4561 PRNT4=JMS I CXPRNT4
4560 SPACE2=JMS I CSPACX2
4557 TYPE=JMS I CXTYPE
4556 CRLF=JMS I CXCRLF
4555 MIOT=JMS I CXMIOT
4554 XOR=JMS I CXORS
4553 YESRNO=JMS I CYESRNK
4552 RANDOM=JMS I CXRAND
4551 SAVGEN=JMS I CXSAVGN
4550 RESGEN=JMS I CXRESGN
4547 BSWAP=JMS I CXBSW

0000 *0
0000 0000 0
0001 5001 JMP 1
0002 0002 2
0003 0003 3

0020 *20
0020 0037 K37, 37
0021 0077 C77, 77
0022 0177 C177, 177
0023 0377 K377, 377
0024 0304 DEVCOD, 0304
0025 0007 SAVBTS, 0007
0026 0004 BITNO, 0004
0027 0377 DATBIT, 0377
0030 0000 BAUDNO, 0000
0031 0000 XMTDAT, 0
0032 0000 XMTDT1, 0
0033 0000 ERRFLG, 0
0034 0000 RECDAT, 0
0035 0000 LOPPC, 0
0036 0000 NOCLAY, 0
0037 0000 CNT1, 0
0040 0000 CNT2, 0

```

0041 0000 TSTCNT, 0
      0200 *200
0200 0203 BGNINT, JMP, +3 /BEGIN INTERROGATION FOR SETUP
0201 0246 NOINTR, JMP, START /GO TO START OF TEST NO INTERROGATION REQUIRED
0202 0777, JMP, BAUDTM /BAUD RATE TIMING TEST (THE PROGRAM SHOULD HALT IN 30 SECONDS
0203 7604 LAS /LOOK AT SR11 FOR DESIRED TYPE OF INTERROGATION
0204 7012 RTR /PUT BIT 11 INTO ACC
0205 7710 SPA CLA /IF AC11=1 USE TELETYPE FOR INPUT OTHERWISE USE THE SR
0206 0776, JMP, TVINTR /GO TO TELETYPE FOR INTERROGATION
0207 7402 HLT /SET BITS 0-5 TO THE RECEIVE IOT AND BITS 6-11 TO TRANSMIT IOT
0210 7604 LAS /GET DEVICE CODE FROM SWITCH REGISTER
0211 0024 DCA DEVCOD /SAVE IT FOR IOT MODIFICATION
0212 7402 HLT /SET PARITY=STATUS=FILLER=BAUD RATE=STOP BITS=AND # OF DATA BITS IN SR
0213 7604 LAS /GET THE SR AND CALCULATE THE RESULTS
0214 0025 DCA SAVBTS /SAVE THEM
0215 1025 SETUP, TAD, SAVBTS /SETUP THE NUMBER OF DATA BITS
0216 0375 AND, (3
0217 0026 DCA BITNO
0220 1374 TAD, (TAD K37
0221 1026 TAD, BITNO
0222 0223 DCA, +1
0223 7402 HLT/TAD, K37+(X)
0224 0027 DCA DATBIT /THIS NUMBER=37,77,177,377 FOR 5,6,7 OR 8 DATA BITS
0225 1025 TAD, SAVBTS /SET UP LENGTH OF CHARACTER FROM # OF STOP BITS
0226 0373 AND, (4 /IF BIT 9=1 2 STOP BITS,IF 0 ONLY 1 STOP BIT
0227 7640 SEA CLA
0230 0026 ISB BITNO /ADD 1 MORE TO CHARACTER LENGTH
0231 1025 TAD, SAVBTS /DOES IT HAVE PARITY
0232 07710 SPA CLA
0233 0026 ISB BITNO /YES BUMP THE CHARACTER LENGTH BY 1
0234 1025 TAD, SAVBTS /SET UP FOR BAUD RATE
0235 7012 RTR,
0236 0710 RAR,
0237 0372 AND, (17
0240 0030 DCA BAUDNO /SAVE THE BAUD NUMBER POINTER
0241 1030 TAD, BAUDNO /IS THE BAUD NUMBER WITHIN LIMITS
0242 1371 TAD, (=13
0243 07740 SMA SEA CLA
0244 0212 JMP, SETUP=3 /NO,BAUD NUMBER OUT OF BOUNDS GO BACK TO STATUS SETUP
0245 7402 HLT /SET SR TO DESIRED SWITCH SETTINGS
0246 1024 START, TAD, DEVCOD /GET THE DEVICE CODE
0247 7012 RTR /PUT THE RECEIVE DEVICE CODE IN BITS 3-8
0250 7010 RAR,
0251 4555 MIOT /GO MODIFY THE IOTS
0252 4370 REPNT /RECEIVE IOT TABLE POINTER
0253 1024 TAD, DEVCOD /GET THE DEVICE CODES
0254 7006 RTL, /PUT THE TRANSMIT DEVICE CODE IN BITS 3-8
0255 7004 RAL,
0256 4555 MIOT /GO MODIFY THEM
0257 4501 XMTIOT /POINTER TO TRANSMIT IOT TABLE
0260 1370 TAD, (JMP I 2 /SET UP INTERRUPT RETURN LOCATIONS
0261 0001 DCA, 1
0262 1367 TAD, (INTRET

```

```

0263 0002 DCA, 2
0264 0766, JMP, CLRBRD /GO START TEST
0265 7240 INTRET, CLA, CHA
0266 0400 JMP, I, 0
0267 0000 XPCRET, 0
0270 7240 CLA, CHA
0271 1267 TAD, XPCRET
0272 0035 DCA, LOOPPC
0273 1276 TAD, H10
0274 0041 DCA, TSTCNT
0275 0667 JMP, I, XPCRET
0276 7770 H10, =10
      /LOOP ON TEST IF SR2=1
0277 0000 XSR2, 0
0300 7604 LAS
0301 7006 RTL,
0302 7710 SPA, CLA
0303 0435 JMP, I, LOOPPC
0304 0677 JMP, I, XSR2
0305 0000 NOT8E, 0 /IF SR10=1 PROCESSOR NOT PDP-8E FAMILY
0306 7604 LAS
0307 7012 RTR
0310 7010 RAR,
0311 7710 SPA, CLA
0312 0315 JMP, +3
0313 0305 ISB, NOT8E
0314 0705 JMP, I, NOT8E
0315 1705 TAD, I, NOT8E
0316 0305 DCA, NOT8E
0317 0705 JMP, I, NOT8E
0366 0400
0367 0265
0370 0402
0371 7765
0372 0017
0373 0004
0374 1020
0375 0003
0376 0200
0377 2453
      *400

```

```

/INITIALIZE THE MODULE WITH A CAF INSTRUCTION AND CHECK THAT THE
/RECEIVE AND TRANSMIT PLUGS ARE NOT STUCK ON AND THAT KSP,TSP
/AND SPI DONT SKIP AND THAT THE INTERRUPT REQUEST LINE
/IS NOT PULLED LOW,

```



```

0400 4576 CLRBRD, STLPCC /SET LOOPING PC FOR TEST AND ERROR LOOPING
0401 4573 SW1ONE /CHECK SR10 TO SEE IF PROCESSOR A PDP-8E
0402 0645 ACNSKP /PROCESSOR NOT A PDP-8E
0403 6007 CAF /CLEAR THE BOARD-CHECK THE SKIP 10T/S NOT TO SKIP
0404 6031 KSF0, KSF /SKIP ON RECEIVE FLAG
0405 7410 SKP
0406 4574 EHLTLP /ERROR,RECEIVE FLAG SET OR KSF SKIPPED
0407 6041 TSF0, TSF /SKIP ON TRANSMIT FLAG
0410 7410 SKP
0411 4574 EHLTLP /ERROR,TRANSMIT FLAG SET OR TSF SKIPPED
0412 6045 SPI0, SPI /SKIP IF XMIT/RECEIVE FLAG SET WITH INT ENB,
0413 7410 SKP
0414 4574 EHLTLP /SPI SKIPPED OR XMIT/RECEIVE FLAG SET WITH INT ENB
0415 6001 IOV /CHECK THAT INT REQ IS NOT PULLED BY INT, ENB,
0416 7000 NOP /AND TRANSMIT/RECEIVE FLAG BEING SET
0417 6002 IOF
0420 7710 SPA
0421 4574 EHLTLP CLA /INT REQ LINE PULLED LOW OR TRANSMIT/RECEIVE FLAG SET
0422 2041 ISE TSTCNT /DO TEST 10 TIMES
0423 5201 JMP CLRBRD+1
0424 4575 LOOP /LOOP IF SR2=1

/ CHECK THAT THE TRANSMIT FLAG CAN BE SET AND CLEARED BY TFL,TSF AND TCF

0425 4576 SCXMIT, STLPCC /SET LOOPING PC FOR TEST AND ERROR LOOPING
0426 6030 KCF0, KCF /CLEAR RECEIVE FLAG FOR SCOPE LOOPING
0427 7410 SKP /SAFETY SKIP IN CASE KCF SKIPPED
0430 4574 EHLTLP /KCF SKIPPED
0431 6040 TFL0, TFL /SET THE TRANSMIT FLAG
0432 7410 SKP /SAFETY SKIP TO CHECK TFL NOT TO SKIP
0433 4574 EHLTLP /ERROR,TFL SKIPPED
0434 6041 TSF1, TSF /SKIP IF TRANSMIT FLAG=1
0435 4574 EHLTLP /TRANSMIT FLAG NOT SET OR TFL FAILED
0436 6042 TCF0, TCF /CLEAR THE TRANSMIT FLAG
0437 7410 SKP /SAFETY SKIP TO CHECK TCF NOT TO SKIP
0440 4574 EHLTLP /ERROR, TCF SKIPPED
0441 6041 TSF2, TSF /SKIP ON TRANSMIT FLAG
0442 7410 SKP
0443 4574 EHLTLP /ERROR,TCF FAILED TO CLEAR TRANSMIT FLAG
0444 6031 KSF1, KSF /CHECK TO SEE IF RECEIVE FLAG GOT SET
0445 7610 SKP CLA
0446 4574 EHLTLP /RECEIVE FLAG SET BY SETTING TRANSMIT FLAG
0447 2041 ISE TSTCNT
0450 5226 JMP SCXMIT+1
0451 4575 LOOP /LOOP IF SR2=1

/ CHECK THAT TRANSMIT FLAG CAN BE CLEARED BY CAF

0452 4576 CAFXMT, STLPCC /SET LOOPING PC FOR TEST AND ERROR LOOPING
0453 6040 TFL1, TFL /SET THE TRANSMIT FLAG
0454 6041 TSF3, TSF /SKIP ON TRANSMIT FLAG
0455 4574 EHLTLP /ERROR,TRANSMIT FLAG FAILED TO SET
0456 6007 CAF /CLEAR ALL FLAGS
0457 6041 TSF4, TSF /SKIP ON TRANSMIT FLAG
0460 7410 SKP /OK FLAG NOT SET

```

```

0461 4574 KSF2, EHLTLP /ERROR,CAF FAILED TO CLEAR TRANSMIT FLAG
0462 6031 KSF /CHECK TO SEE IF RECEIVE FLAG GOT SET
0463 7610 SKP CLA /NO, IT DIDN'T
0464 4574 EHLTLP /ERROR,RECEIVE FLAG GOT SET
0465 2041 ISE TSTCNT /DO TEST 10 TIMES
0466 5253 JMP CAFXMT+1
0467 4575 LOOP /LOOP IF SR2=1

/ USING THE TRANSMIT FLAG-CHECK THAT INTERRUPT ENABLE CAN BE SET
/ AND CLEARED AND THAT THE PROGRAM CAN INTERRUPT, INTERRUPT
/ ENABLE IS SET AND CLEARED BY DATA BIT 11 AND KIE COMMAND,

0470 4576 INTXMT, STLPCC /SET LOOPING PC FOR TEST AND ERROR LOOPING
0471 6030 KCF1, KCF /CLEAR RECEIVE FLAG FOR SCOPE LOOPING
0472 7300 CLA
0473 6035 KIE0, KIE CLL /CLEAR INTERRUPT ENABLE
0474 7410 SKP /SAFETY SKIP TO CHECK KIE NOT TO SKIP
0475 4574 EHLTLP /ERROR, KIE SKIPPED
0476 6040 TFL2, TFL /SET THE TRANSMIT FLAG
0477 6041 TSF5, TSF /SKIP ON TRANSMIT FLAG
0500 4574 EHLTLP /TRANSMIT FLAG FAILED TO SET
0501 6045 SPI1, SPI /SKIP IF INTERRUPT ENABLE AND XMIT FLAG SET,
0502 7410 SKP /OK, INTERRUPT ENABLE NOT SET
0503 4574 EHLTLP /ERROR,INT ENB, SET,FAILED TO BE CLEARED BY KIE
0504 6001 IOV /CHECK THAT INTERRUPT REQUEST IS NOT PULLED
0505 7000 NOP /INTERRUPT HERE IF SET
0506 6002 IOF /TURN THE INTERRUPT OFF
0507 7710 SPA CLA
0510 4574 EHLTLP /PROGRAM INTERRUPTED-CHECK INT, ENB,
0511 7001 IAC
0512 6035 KIE1, KIE CLA /SET INT, ENB, WITH DATA BIT 11 AND KIE
0513 7610 SKP
0514 4574 EHLTLP /ERROR, KIE SKIPPED
0515 6041 TSF6, TSF /SKIP ON TRANSMIT FLAG
0516 4574 EHLTLP /ERROR,TRANSMIT FLAG GOT CLEARED
0517 6045 SPI2, SPI /SKIP ON INT ENB AND TRANSMIT FLAG
0520 4574 EHLTLP /SPI FAILED TO SKIP OR INT ENB NOT SET
0521 6001 IOV /CHECK THAT INTERRUPT REQUEST IS PULLED
0522 7000 NOP /SHOULD INTERRUPT HERE
0523 6002 IOF /TURN IT OFF
0524 7700 SHA CLA /DID IT INTERRUPT?
0525 4574 EHLTLP /FAILED TO INTERRUPT-CHECK XMIT AND INT ENB
0526 6042 TCF1, TCF /CLEAR THE TRANSMIT FLAG
0527 6041 TSF7, TSF /CHECK TO SEE IF IT CLEARED
0530 7410 SKP
0531 4574 EHLTLP /IT FAILED TO CLEAR
0532 6045 SPI3, SPI /SKIP ON INT ENB AND TRANSMIT FLAG
0533 7410 SKP
0534 4574 EHLTLP /TRANSMIT FLAG IS GONE IT SHOULDN'T SKIP
0535 6001 IOV /CHECK THAT IT DOESN'T INTERRUPT
0536 7000 NOP /
0537 6002 IOF /
0540 7710 SPA CLA
0541 4574 EHLTLP /PROGRAM INTERRUPTED WITHOUT TRANSMIT FLAG
0542 6040 TFL3, TFL /SET THE FLAG AGAIN

```

```

0543 6041 TSF0, TSF /SKIP ON THE TRANSMIT FLAG
0544 4574 EHLTLP /FLAG FAILED TO SET
0545 6045 SPI4, SPI /SKIP ON XMIT AND INT, ENB,
0546 4574 EHLTLP /FAILED TO SKIP ON INT ENB AND XMIT FLAG
0547 6035 KIE2, KIE /CLEAR INTERRUPT ENABLE WITH KIE AND DATA BIT 11
0550 6045 SPI5, SPI /SKIP IF INT ENB=1 WITH XMIT FLAG
0551 7410 SKP
0552 4574 EHLTLP /KIE FAILED TO CLEAR INTERRUPT ENABLE
0553 6001 IOV /CHECK THAT THE PROGRAM DOESN'T INTERRUPT
0554 7000 NOP
0555 6002 IOF
0556 7710 SPA
0557 4574 EHLTLP CLA /PROGRAM INTERRUPTED WITHOUT INT ENB
0560 6042 TCF2, TCF /CLEAR TRANSMIT FLAG
0561 6031 KSF3, KSF /CHECK TO SEE IF RECEIVE FLAG GOT SET
0562 7410 SKP
0563 4574 EHLTLP CLA /RECEIVE FLAG SET BY ABOVE CODE
0564 2041 ISE TSTCNT
0565 5271 JMP INTXMT+1
0566 4575 LOOP /LOOP ON TEST IF SR2=1
0567 3777 JMF CAFINT

```

0577 0600
0600 *400

/CHECK THAT CAF WILL SET INTERRUPT ENABLE USING THE TRANSMIT
/FLAG TO SKIP AND INTERRUPT ON,

```

0600 4576 CAFINT, STLPFC /SET THE LOOPING PC FOR TEST AND ERROR LOOPING,
0601 6030 KCF2, KCF /CLEAR RECEIVE FLAG
0602 6035 KIE3, KIE /CLEAR INTERRUPT ENABLE
0603 6040 TFL4, TFL /SET THE TRANSMIT FLAG
0604 6041 TSP9, TSP /SKIP ON THE TRANSMIT FLAG
0605 4574 EHLTLP /FLAG FAILED TO SET
0606 6045 SPI6, SPI /SKIP ON TRANSMIT FLAG AND INT ENB,
0607 7410 SKP
0610 4574 EHLTLP /SPI SKIPPED OR INT ENB IS SET
0611 6007 CAF /CLEAR TRANSMIT FLAG AND SET INT ENB,
0612 6041 TSP10, TSP /SKIP IF TRANSMIT FLAG = 1
0613 7410 SKP
0614 4574 EHLTLP /CAF FAILED TO CLEAR XMIT FLAG
0615 6045 SPI7, SPI /SKIP ON TRANSMIT FLAG AND INT ENB,
0616 7410 SKP
0617 4574 EHLTLP /SPI SKIPPED WITHOUT TRANSMIT FLAG
0620 6040 TFL5, TFL /SET THE TRANSMIT FLAG
0621 6041 TSP11, TSP /SKIP IF XMIT FLAG IS SET
0622 4574 EHLTLP /TRANSMIT FLAG FAILED TO SET
0623 6045 SPI8, SPI /SKIP ON INTERRUPT ENABLE AND TRANSMIT FLAG
0624 4574 EHLTLP /CAF FAILED TO SET INTERRUPT ENABLE
0625 6001 IOV /CHECK THAT THE PROGRAM WILL INTERRUPT
0626 7000 NOP /GO AND INTERRUPT
0627 6002 IOF /TURN THE INTERRUPT OFF IF IT DIDN'T
0630 7700 SMA CLA
0631 4574 EHLTLP /PROGRAM FAILED TO INTERRUPT WITH XMIT AND INT ENB

```

```

0632 6042 TCF3, TCF /CLEAR THE TRANSMIT FLAG
0633 6041 TSP12, TSP /SKIP IF TRANSMIT FLAG IS SET
0634 7610 SKP CLA /IS FLAG SET
0635 4574 EHLTLP /FLAG FAILED TO CLEAR
0636 6035 KIE4, KIE /CLEAR INTERRUPT ENABLE
0637 6031 KSF4, KSF /CHECK TO SEE IF RECEIVE FLAG IS SET
0640 7610 SKP CLA
0641 4574 EHLTLP /RECEIVE FLAG GOT SET DURING TEST
0642 2041 ISE TSTCNT
0643 5201 JMP CAFINT+1
0644 4575 LOOP /LOOP ON TEST IF SR2=1

```

/THE FOLLOWING TEST CHECKS THE EFFECT OF THE IOT ON THE AC
/AND ALSO CHECKS THAT THE IOTS DON'T SKIP, IPC AND TCS ARE NOT TESTED.

```

0645 4576 ACNSKP, STLPFC /SET THE LOOPING PC FOR TEST AND ERROR LOOPING
0646 4575 SHONE /CHECK TO SEE IF PROCESSOR A PDP8E
0647 0657 KSF5=1 /NOT A PDP=8E GO TO NEXT SUBTEST
0650 7240 CLA CMA /SET AC TO ALL ONE'S
0651 6030 KCF3, KCF /CLEAR THE RECEIVE FLAG
0652 7410 SKP
0653 4777 JMS HLTL1 /ERROR, KCF SKIPPED
0654 7040 CMA /SET THE AC BACK TO ZEROES
0655 7440 SEA
0656 4776 JMS HLTL2 /ERROR, KCF CHANGED THE AC
0657 7240 CLA CMA /SET THE AC BACK TO 1'S
0660 6031 KSF5, KSF /SKIP ON RECEIVE FLAG
0661 7410 SKP
0662 4777 JMS HLTL1 /ERROR, RECEIVE FLAG SHOULD NOT BE SET
0663 7040 CMA /SET THE AC BACK TO ZEROES
0664 7440 SEA
0665 4776 JMS HLTL2 /ERROR, KSF CHANGED THE AC
0666 7240 CLA CMA /SET THE AC TO ALL 1'S
0667 6032 KCO0, KCO /CLEAR AC AND RECEIVE FLAG AND SET READER RUN
0670 7410 SKP
0671 4777 JMS HLTL1 /ERROR, KCO SKIPPED
0672 7440 SEA
0673 4776 JMS HLTL2 /ERROR, KCO FAILED TO CLEAR AC
0674 1375 TAO (#4 /SET AC TO ALL 1'S EXCEPT BITS 10 AND 11
0675 6035 KIE5, KIE /CLEAR INTERRUPT AND STATUS ENABLE
0676 7410 SKP
0677 4777 JMS HLTL1 /ERROR, KIE SKIPPED
0700 1374 TAO (3 /ADD 3 TO AC AND THEN COMPLEMENT IT
0701 7040 CMA
0702 7440 SEA
0703 4776 JMS HLTL2 /ERROR, KIE CHANGED THE AC
0704 7240 CLA CMA /SET THE AC = TO ALL 1'S
0705 6034 KR00, KRS /READ RECEIVE BUFFER STATIC AND STATUS
0706 7410 SKP
0707 4777 JMS HLTL1 /ERROR, KRS SKIPPED
0710 7040 CMA /SET THE AC BACK TO ZEROES
0711 7440 SEA
0712 4776 JMS HLTL2 /ERROR, KRS CHANGED THE AC
0713 1146 TAO (7400 /SET AC BITS 0-3
0714 6036 KR00, KRS /CLEAR AC AND RECEIVE FLAG AND READ RECEIVE BUF

```

```

0715 7410 SKP
0716 4777 JMS HLTL1 /ERROR, KRB SKIPPED
0717 0146 AND [7400
0720 7440 SZA
0721 4776 JMS HLTL2 /KRB FAILED TO CLEAR THE AC
0722 4573 SW1ONE /CHECK TO SEE IF POP-BE
0723 0733 TCF4=1 /PROCESSOR NOT A POP-BE GO DO NEXT SUBTEST
0724 7240 CLA CHA /SET AC EQUAL TO ALL ONES
0725 6040 TFL6, TFL /SET THE TRANSMIT FLAG
0726 7410 SKP
0727 4777 JMS HLTL1 /ERROR, TFL SKIPPED
0730 7040 CHA /SET THE AC BACK TO 0
0731 7440 SZA
0732 4776 JMS HLTL2 /TFL CHANGED THE AC
0733 7240 CLA CHA /SET THE AC TO 1/S
0734 6042 TCF4, TCF /CLEAR THE TRANSMIT FLAG
0735 7410 SKP
0736 4777 JMS HLTL1 /ERROR, TCF SKIPPED
0737 7040 CHA /SET THE AC BACK TO 0
0740 7440 SZA
0741 4776 JMS HLTL2 /TCF CHANGED THE AC
0742 7240 CLA CHA /SET THE AC TO ALL 1/S
0743 6041 TSF13, TSF /SKIP ON TRANSMIT FLAG
0744 7410 SKP
0745 4777 JMS HLTL1 /TRANSMIT FLAG IS SET
0746 7040 CHA /SET THE AC BACK TO 0
0747 7440 SZA
0750 4776 JMS HLTL2 /ERROR, TSF CHANGED THE AC
0751 7240 CLA CHA /SET THE AC TO ALL 1/S
0752 6045 SPI9, SPI /SKIP IF XMT/REC + INT ENB =1
0753 7410 SKP
0754 4777 JMS HLTL1 /ERROR, SPI SKIPPED OR XMT/REC AND INT ENB =1
0755 7040 CHA /SET THE AC BACK TO ZERO
0756 7440 SZA
0757 4776 JMS HLTL2 /ERROR, SPI CHANGED THE AC
0760 2041 ISZ TSTCNT
0761 0246 JMP ACNSKP+1
0762 4575 LOOP /LOOP ON TEST IF SR2=1
0763 3773 JMP STPLGS

0773 1000
0774 0003
0775 7774
0776 2433
0777 2414
1000 1000 *1000
    
```

/START OF LOOP BACK TEST
 /CHECK THAT THE TRANSMIT FLAG CAN BE SET BY TPC AND THAT DATA
 /AVAILABLE WILL SET THE RECEIVE FLAG, CHECK THAT THE FLAGS CAN
 /BE CLEARED BY TCF AND KCG, CHECK THAT THE FLAGS CAN CAUSE AN
 /INTERRUPT BY MANIPULATING INTERRUPT ENABLE,

1000 4576 STPLGS, STLPPC /STORE LOOPING PC FOR TEST AND SCOPE LOOPING

```

1001 1145 TAD [=3720 /SET UP A DELAY OF 200MS TO ALLOW FLAGS TO SETTLE
1002 3036 DCA NDELAY
1003 4577 DELAY /WAIT FOR 200MS
1004 6035 KIE6, KIE /CLEAR INTERRUPT ENABLE
1005 7410 SKP CLA /SAFETY SKIP TO CHECK KIE NOT TO SKIP
1006 4574 EHLTLP /ERROR, KIE SKIPPED
1007 6032 KCG1, KCG /CLEAR AC AND RECEIVE FLAG AND SET READER RUN
1010 7410 SKP CLA /SAFETY SKIP TO CHECK KCG NOT TO SKIP
1011 4574 EHLTLP /ERROR, KCG SKIPPED
1012 6042 TCF5, TCF /CLEAR TRANSMIT FLAG
1013 7410 SKP CLA /SAFETY SKIP TO CHECK TCF NOT TO SKIP
1014 4574 EHLTLP /ERROR, TCF SKIPPED
1015 6031 KSF6, KSF /CHECK THE RECEIVE FLAG TO BE ZERO
1016 7410 SKP CLA
1017 4574 EHLTLP /ERROR, RECEIVE FLAG =1 OR KSF SKIPPED
1020 6041 TSF14, TSF /SKIP IF TRANSMIT FLAG =1
1021 7410 SKP CLA
1022 4574 EHLTLP /ERROR, TRANSMIT FLAG=1 OR TSF SKIPPED
1023 6044 TPC0, TPC /LOAD TRANSMIT BUFFER AND TRANSMIT
1024 7410 SKP CLA /SAFETY SKIP TO CHECK TPC NOT TO SKIP
1025 4574 EHLTLP /ERROR, TPC SKIPPED
1026 4572 TSFXP /WAIT FOR A SECOND FOR TRANSMIT FLAG TO SET
1027 4574 EHLTLP /ERROR, TPC FAILED TO SET XMIT FLAG OR TSF FAILED
1030 6031 KSF7, KSF CLA /CHECK THE RECEIVE FLAG TO STILL BE A 0
1031 7410 SKP CLA
1032 4574 EHLTLP /RECEIVE FLAG GOT SET TO SOON
1033 6045 SPI10, SPI /SKIP IF XMIT/RECEIVE FLAG=1 AND INT ENB SET
1034 7410 SKP CLA
1035 4574 EHLTLP /ERROR, SPI SKIPPED OR INTERRUPT ENABLE SET
1036 6001 ION /CHECK THE PROGRAM NOT TO INTERRUPT
1037 7000 NOP
1040 6002 IOP
1041 7710 SPA CLA
1042 4574 EHLTLP /ERROR, INT ENB SET OR INT REQ PULLED LOW
1043 7301 CLA OLL IAC /SET INTERRUPT ENABLE TO A 1
1044 6035 KIE7, KIE /AC11=1 AND KIE SET INTERRUPT ENABLE
1045 6045 SPI11, SPI /SKIP IF XMIT/RECEIVE FLAG=1 WITH INT ENABLE
1046 4574 EHLTLP /INTERRUPT ENABLE FAILED TO SET OR KIE FAILED
1047 6001 ION /CHECK THE PROGRAM TO INTERRUPT
1050 7000 NOP /IT SHOULD INTERRUPT HERE
1051 6002 IOP /TURN IT OFF
1052 7700 SMA CLA
1053 4574 EHLTLP /ERROR PROGRAM FAILED TO INTERRUPT WITH XMIT AND INT ENABLE
1054 6035 KIE8, KIE /SET INT ENB=0 WITH AC11=0 AND KIE COMMAND
1055 6045 SPI12, SPI /CHECK THAT INT ENB CLEARED BY KIE
1056 7410 SKP CLA
1057 4574 EHLTLP /ERROR, INT ENB FAILED TO CLEAR OR SPI SKIPPED
1060 6001 ION /CHECK THE PROGRAM NOT TO INTERRUPT
1061 7000 NOP
1062 6002 IOP
1063 7710 SPA CLA
1064 4574 EHLTLP /ERROR PROGRAM INTERRUPTED WITHOUT INT ENABLE
1065 6041 TSF16, TSF /CHECK THE TRANSMIT FLAG TO STILL = 1
1066 4574 EHLTLP /SOMETHING CLEARED THE TRANSMIT FLAG
1067 6042 TCF6, TCF /CLEAR THE TRANSMIT FLAG
    
```

```

1070 6041 TSF17, TSF: /SKIP IF TRANSMIT FLAG #1
1071 7610 SKP CLA
1072 4574 EHLLTP /ERROR, TCF FAILED TO CLEAR XMIT FLAG
1073 4571 KSF1, KSF: /WAIT FOR ABOUT A SECOND FOR RECEIVE FLAG TO SET
1074 4574 EHLLTP /ERROR, RECEIVE FLAG #0 OR DATA AVAILABLE FAILED TO SET RECEIVE FLAG
1075 6034 KRS1, KRS: /CHECK THAT KRS DOESN'T CLEAR RECEIVE FLAG
1076 7610 SKP CLA /SAFETY SKIP TO CHECK KRS NOT TO SKIP
1077 4574 EHLLTP /ERROR, KRS SKIPPED
1100 6031 KSF9, KSF: /SKIP ON RECEIVE FLAG
1101 4574 EHLLTP /KRS CLEARED RECEIVE FLAG
1102 6045 SPI13, SPI: /SKIP IF XMIT/RECEIVE FLAG AND INT ENABLE#1
1103 7610 SKP CLA
1104 4574 EHLLTP /ERROR SPI SKIPPED OR INT ENABLE#1
1105 6001 ION /CHECK THE PROGRAM NOT TO INTERRUPT
1106 7000 NOP:
1107 6002 IOF:
1110 7710 SPA CLA
1111 4574 EHLLTP /PROGRAM INTERRUPT WITHOUT INTERRUPT ENABLE
1112 7301 CLA CLL IAC /SET INT ENB F/F#1
1113 6035 KIE9, KIE:
1114 6045 SPI14, SPI: /SKIP IF RECEIVE AND INT ENB#1
1115 4574 EHLLTP /ERROR, SPI FAILED OR RECEIVE/INT ENB NOT# TO A 1
1116 6001 ION /CHECK THE PROGRAM TO INTERRUPT
1117 7000 NOP:
1120 6002 IOF:
1121 7700 SMA CLA
1122 4574 EHLLTP /ERROR, FAILED TO INTERRUPT WITH INT ENB AND RECEIVE FLAG = A.1
1123 6032 KCC2, KCC: /CLEAR AC AND RECEIVE FLAG AND SET READER RUN
1124 6031 KSF10, KSF: /SKIP IF RECEIVE FLAG #1
1125 7610 SKP CLA
1126 4574 EHLLTP /ERROR, KCC FAILED TO CLEAR RECEIVE FLAG
1127 6045 SPI15, SPI: /SKIP IF INT ENB AND RECEIVE FLAG #1
1130 7610 SKP CLA
1131 4574 EHLLTP /ERROR, SPI SKIPPED WITHOUT RECEIVE FLAG = 1
1132 6001 ION /CHECK THE PROGRAM NOT TO INTERRUPT
1133 7000 NOP:
1134 6002 IOF:
1135 7710 SPA CLA
1136 4574 EHLLTP /INTERRUPTED WITHOUT RECEIVE FLAG SET
1137 6035 KIE10, KIE: /CLEAR INTERRUPT ENABLE
1140 7300 CLA CLL
1141 2041 ISE TSTCNT
1142 5204 JMP STPLGS#4
1143 4575 LOOP
1144 5777/ JMP XMTREC

1177 1200
1200 *1200

```

THE FOLLOWING TEST CHECKS THAT A TPC COMMAND WILL SET THE TRANSMIT FLAG AND THAT A TFS WILL CLEAR THE FLAG AND THEN RESET IT, CHECK THAT THE RECEIVE FLAG WILL GET SET FROM A TPC AND TFS COMMAND AND THAT IT CAN BE CLEARED BY A KRB AND KCC OR KCF COMMAND

```

1200 4576 XMTREC, STLPPC /STORE LOOPING PC FOR TEST AND ERROR LOOPING
1201 1145 TAD [3720 /DELAY 200MS FOR SCOPE LOOPING TO LET
1202 3036 DCA NDELAY /FLAGS SETTLE
1203 4577 DELAY /GO DELAY 200MS
1204 6032 KCC3, KCC: /CLEAR AC AND RECEIVE FLAG
1205 6042 TCF7, TCF: /CLEAR THE TRANSMIT FLAG
1206 6031 KSF11, KSF: /CHECK THE RECEIVE FLAG TO BE 0
1207 7610 SKP CLA
1210 4574 EHLLTP /RECEIVE FLAG STILL # 1 AFTER A KCC COMMAND
1211 6041 TSF18, TSF: /SKIP IF TRANSMIT FLAG # 1
1212 7610 SKP CLA
1213 4574 EHLLTP /TRANSMIT FLAG STILL A 1 AFTER A TCF COMMAND
1214 6044 TPC1, TPC: /LOAD TRANSMIT BUFFER AND TRANSMIT
1215 4572 TSF5, TSF: /WAIT FOR THE FLAG TO SET
1216 4574 EHLLTP /ERROR XMIT FLAG FAILED TO SET BY TPC
1217 6031 KSF12, KSF: /CHECK THE RECEIVE FLAG TO STILL BE 0
1220 7610 SKP CLA
1221 4574 EHLLTP /RECEIVE FLAG SET TO SOON
1222 6046 TFS, TFS: /LOAD TRANSMIT BUFFER AND TRANSMIT AND CLEAR FLAG
1223 7610 SKP CLA /SAFETY SKIP TO CHECK TFS NOT TO SKIP
1224 4574 EHLLTP /ERROR, TFS SKIPPED
1225 6041 TSF20, TSF: /SKIP IF TRANSMIT FLAG # 1
1226 7610 SKP CLA
1227 4574 EHLLTP /ERROR, TFS FAILED TO CLEAR TRANSMIT FLAG
1230 4571 KSF5, KSF: /WAIT FOR RECEIVE FLAG TO SET FROM FIRST XMIT
1231 4574 EHLLTP /ERROR, REC FLAG FAILED TO SET FROM FIRST XMIT
1232 6036 KRB1, KRB: /CLEAR AC AND RECEIVE FLAG AND READ RECEIVE BUFF
1233 7610 SKP CLA /SAFETY SKIP TO CHECK KRB NOT TO SKIP
1234 4574 EHLLTP /ERROR, KRB SKIPPED
1235 6031 KSF14, KSF: /SKIP ON RECEIVE FLAG
1236 7610 SKP CLA
1237 4574 EHLLTP /ERROR, KRB FAILED TO CLEAR RECEIVE FLAG
1240 4572 TSF5, TSF: /WAIT FOR TRANSMIT FLAG TO SET FROM 2ND XMIT
1241 4574 EHLLTP /TRANSMIT FLAG FAILED TO SET FROM TFS COMMAND
1242 6042 TCF8, TCF: /CLEAR THE TRANSMIT FLAG
1243 6041 TSF22, TSF: /SKIP IF TRANSMIT FLAG SET
1244 7610 SKP CLA
1245 4574 EHLLTP /ERROR, TCF FAILED TO CLEAR FLAG
1246 4571 KSF5, KSF: /WAIT FOR RECEIVE FLAG TO SET FROM TFS COMMAND
1247 4574 EHLLTP /ERROR, RECEIVE FLAG FAILED TO SET FROM 2ND TRANSMIT
1250 7604 LAS /CHECK TO SEE IF PDP-8E
1251 7010 RAR:
1252 7012 RTN:
1253 7700 SHA CLA
1254 5262 JMP .06 /POPBE
1255 6032 KCC4, KCC: /CLEAR THE AC AND RECEIVE FLAG
1256 6031 KSF16, KSF: /SKIP IF RECEIVE FLAG # 1
1257 7610 SKP CLA
1260 4574 EHLLTP /ERROR, KCC FAILED TO CLEAR RECEIVE FLAG
1261 5266 JMP .05
1262 6030 KCF4, KCF: /CLEAR THE RECEIVE FLAG
1263 6031 KSF17, KSF: /SKIP IF RECEIVE FLAG SET
1264 7610 SKP CLA
1265 4574 EHLLTP /ERROR, KCF FAILED TO CLEAR FLAG
1266 2041 ISE TSTCNT

```

```

1267 5204 JMP XMTREC+4
1270 4575 LOOP /LOOP IF SR2=1

1271 5777' JMP SDTST1 /GO TEST THE DATA PATH

1377 1400
1400 1400 *1400
    
```

/START OF DATA TEST=TRANSMIT 1 WORD AND THEN WAIT FOR THE /RECEIVE FLAG TO SET

/DATA TEST 1 - TRANSMIT 0'S AND CHECK THAT 0'S CAME BACK

```

1400 4576 SDTST1, STLPPC /STORE LOOPING PC FOR TEST LOOPING
1401 3031 DCA XMTDAT /SET TRANSMIT WORD EQUAL TO ZERO
1402 4327 JMS SLWDAT /GO TRANSMIT AND RECEIVE 1 WORD
1403 2041 ISZ TSTCNT /DO 8 TIMES
1404 5202 JMP ,=2
1405 4575 LOOP /LOOP IF SR2=1
    
```

/DATA TEST 2 - TRANSMIT ALL ONES AND CHECK THAT 1'S CAME BACK

```

1406 4576 SDTST2, STLPPC /STORE LOOPING PC FOR TEST LOOPING
1407 7240 CLA CMA
1410 0027 AND DATBIT /MASK OUT FOR THE NUMBER OF DATA BITS
1411 3031 DCA XMTDAT /SAVE THE WORD FOR TRANSMITTING
1412 4327 JMS SLWDAT /GO TRANSMIT AND RECEIVE ONE WORD
1413 2041 ISZ TSTCNT /DO TEST 8 TIMES
1414 5212 JMP ,=2
1415 4575 LOOP /LOOP ON TEST IF SR2=1
    
```

/DATA TEST 3 - TRANSMIT ONES AND ZEROES

```

1416 4576 SDTST3, STLPPC /STORE LOOPING PC FOR TEST LOOPING
1417 3031 DCA XMTDAT /SET TRANSMIT WORD EQUAL TO 0
1420 4327 JMS SLWDAT /GO TRANSMIT AND RECEIVE ONE WORD
1421 7240 CLA CMA
1422 0027 AND DATBIT
1423 3031 DCA XMTDAT /SET TRANSMIT WORD - TO ALL ONES
1424 4327 JMS SLWDAT /GO TRANSMIT AND CHECK IT
1425 2041 ISZ TSTCNT /DO TEST 8 TIMES
1426 5217 JMP SDTST3+1
1427 4575 LOOP /LOOP ON TEST IF SR 2=1
    
```

/DATA TEST 4 - TRANSMIT AND RECEIVE A BINARY COUNT PATTERN

```

1430 4576 SDTST4, STLPPC /STORE LOOPING PC FOR TEST LOOPING
1431 1027 TAD DATBIT /SET UP WORD COUNTER FROM THE # OF DATA BITS
1432 7040 CMA
1433 3041 DCA TSTCNT /SAVE IT
1434 3031 DCA XMTDAT /CLEAR THE TRANSMIT WORD
1435 4327 JMS SLWDAT /GO TRANSMIT AND RECEIVE ONE WORD
1436 2031 ISZ XMTDAT /INCREMENT THE TRANSMIT WORD
1437 1031 TAD XMTDAT
1440 4777' JMS FILCHK /GO CHECK FILLER CHAR FOR LF IF SELECTED
    
```

```

1441 5236 JMP ,=3 /FIL IS SELECTED AND ITS A LF GO GET NEW WORD
1442 2041 ISZ TSTCNT /DONE YET?
1443 5235 JMP ,=6 /NO GO DO NEXT WORD
1444 4575 LOOP /LOOP ON TEST IF SR2=1
    
```

/DATA TEST 5 - TRANSMIT A COMPLEMENTING BINARY COUNT PATTERN

```

1445 4576 SDTST5, STLPPC /STORE LOOPING PC FOR TEST LOOPING
1446 1027 TAD DATBIT /SET UP WC FROM NUMBER OF DATA BITS
1447 7040 CMA
1450 3041 DCA TSTCNT /SAVE IT
1451 3031 DCA XMTDAT /CLEAR THE TRANSMIT WORD
1452 4327 JMS SLWDAT /GO TRANSMIT AND RECEIVE ONE WORD
1453 1031 TAD XMTDAT /SET THE TRANSMIT WORD TO ITS COMPLEMENT
1454 7040 CMA
1455 0027 AND DATBIT
1456 3031 DCA XMTDAT /SAVE THE NEW WORD
1457 1031 TAD XMTDAT
1460 4777' JMS FILCHK
1461 7410 SK*
1462 4327 JMS SLWDAT /GO TRANSMIT AND RECEIVE ONE WORD
1463 1031 TAD XMTDAT /RESET THE WORD BACK AND ADD ONE
1464 7041 CIA
1465 0027 AND DATBIT
1466 3031 DCA XMTDAT
1467 1031 TAD XMTDAT
1470 4777' JMS FILCHK
1471 5253 JMP SDTST5+6
1472 2041 ISZ TSTCNT
1473 5252 JMP SDTST5+5
1474 4575 LOOP /LOOP ON TEST IF SR2=1
    
```

/DATA TEST 6 - TRANSMIT A COMPLEMENTING ONE'S AND ZEROES PATTERN

```

1475 4576 SDTST6, STLPPC /STORE THE LOOPING PC FOR TEST LOOPING
1476 1376 TAD (2B2
1477 0027 AND DATBIT /MASK OUT FOR NUMBER OF BITS
1500 3031 DCA XMTDAT /SAVE IT
1501 4327 JMS SLWDAT /GO TRANSMIT AND CHECK THE WORD
1502 1375 TAD (125 /GET THE COMPLEMENTING WORD
1503 0027 AND DATBIT /MASK OUT FOR NUMBER OF BITS
1504 3031 DCA XMTDAT /SAVE IT
1505 4327 JMS SLWDAT /GO TRANSMIT AND CHECK IT
1506 2041 ISZ TSTCNT /DO TEST 8 TIMES
1507 5276 JMP SDTST6+1 /DO AGAIN
1510 4575 LOOP /LOOP ON TEST IF SR2=1
    
```

/DATA TEST 7 - RANDOM DATA TEST

```

1511 4576 SDTST7, STLPPC /STORE LOOPING PC FOR TEST LOOPING
1512 1374 TAD (=1000
1513 3041 DCA TSTCNT
1514 4592 RANDOM
1515 0027 AND DATBIT
1516 3031 DCA XMTDAT
    
```

```

1517 1031 TAD XMTDAT
1520 4777/ JMS FILCHK /CHECK FOR FILLER CHARACTERS
1521 5314 JMP ,=5
1522 4327 JMS SLWDAT /GO TRANSMIT THE RANDOM NUMBER AND CHECK IT
1523 2041 ISE TSTCNT
1524 5314 JMP ,=10
1525 4575 LOOP /LOOP ON TEST IF SR2=1
1526 5773/ JMP FDATAT

/SLOW DATA TEST ROUTINE

1527 0000 SLWDAT, 0
1530 7326 CLA OLL OML RTL /SET STATUS WORD ENABLE
1531 6035 KSE0, KSE
1532 7200 CLA
1533 1031 TAD XMTDAT /GET WORD TO BE TRANSMITTED
1534 6046 SLWTL5, TLS /LOAD AND TRANSMIT AND CLEAR
1535 6031 KSF24, KSF /SKIP IF THE RECEIVE FLAG=1
1536 7410 SKP
1537 5344 JMP SLWREC /GO GET THE RECEIVE WORD
1540 6041 TSF32, TSF /SKIP IF TRANSMIT FLAG SET
1541 5335 JMP ,=4
1542 6042 TCF14, TCF /CLEAR THE TRANSMIT FLAG
1543 5335 JMP SLWTL5+1/GO WAIT FOR THE RECEIVE FLAG

1544 7240 SLWREC, CLA CMA
1545 6036 KRB11, KRB /READ RECEIVE BUFFER AND CLEAR RECEIVE FLAG
1546 3034 DCA RECDAT /SAVE THE WORD RECEIVED
1547 1034 TAD RECDAT /COMPARE THE WORD WITH THE WORD TRANSMITTED
1548 7041 CIA
1549 1031 TAD XMTDAT
1550 7050 CLA SNA /GO GET THE NEXT WORD
1551 5364 JMP ,+11
1552 7004 LAS
1553 7110 CLA /IS SR0 SET
1554 5364 JMP ,+6 /YES,DON'T HALT ON ERROR,GO CHECK LOOP SWITCH
1555 1031 TAD XMTDAT
1556 7402 HLT /GOOD DATA WORD IN AC=WORD TRANSMITTED
1557 7200 CLA
1558 1034 TAD RECDAT
1559 7402 HLT /AB=BAD DATA WORD=WORD RECEIVED
1560 7004 LAS /CHECK SR1 FOR LOOP ON ERROR
1561 7004 RAL /PUT BIT 1 IN AC0
1562 7110 CLA /LOOP?
1563 5333 JMP SLWTL5-1/YES,GO TRANSMIT AND RECEIVE SAME WORD
1564 5727 JMP I SLWDAT /RETURN FOR THE NEXT WORD

1573 1600
1574 7000
1575 0125
1576 0252
1577 3046
1580 1600 *1600

```

```

/FASTER DATA TEST = TRANSMIT-TRANSMIT=RECEIVE=TRANSMIT=RECEIVE=ETC

1600 4576 FDATAT, STLPPC /STORE LOOPING PC FOR TEST LOOPING
1601 1145 TAD (=3720
1602 3036 DCA NDELAY /SETUP A DELAY OF 200 MS FOR SCOPE LOOPING
1603 1377 TAD (=1000
1604 3041 DCA TSTCNT /SETUP A TEST LOOP OF 1000 WORDS
1605 3033 DCA ERRFLG /CLEAR THE PROGRAM ERROR FLAG
1606 4577 DELAY /DELAY 200MS
1607 6036 KRB2, KRB /ISSUE A KRB TO CLEAR ANY RECEIVE FLAG SET
1610 7200 CLA
1611 6042 TCF9, TCF /CLEAR THE TRANSMIT FLAG IF SET
1612 4552 RANDOM /GENERATE A RANDOM NUMBER
1613 0027 AND DATBIT /MASK OUT FOR NUMBER OF DATA BITS
1614 3031 DCA XMTDAT /SAVE THE FIRST WORD
1615 1031 TAD XMTDAT
1616 4776/ JMS FILCHK /CHECK TO SEE IF FILL CHAR OPTION SELECTED
1617 3212 JMP ,=5 /GO GET NEW WORD,IT WAS A LF AND FILL WAS SELECTED
1620 1031 TAD XMTDAT /GET THE WORD TO TRANSMIT
1621 6046 XMIT, TLS /TRANSMIT THE WORD
1622 6031 KSF10, KSF /IS THE RECEIVE FLAG SET
1623 7410 SKP /NOPE
1624 5237 JMP RECEVE /YES,GO GET THE WORD
1625 6041 TSF23, TSF /SKIP IF TRANSMIT FLAG SET
1626 5222 JMP ,=4
1627 4552 RANDOM /GO GENERATE ANOTHER WORD
1630 0027 AND DATBIT /MASK OUT FOR NUMBER OF DATA BITS
1631 3032 DCA XMTDT1 /SAVE THE SECOND WORD
1632 1032 TAD XMTDT1
1633 4776/ JMS FILCHK /CHECK TO SEE IF FILL=1 AND THAT IT WAS A LF
1634 3227 JMP ,=5 /YES IT WAS,GO GENERATE A NEW WORD
1635 1032 TAD XMTDT1 /GET THE WORD AND PRINT IT
1636 3221 JMP XMIT /
1637 7240 RECEVE, CLA CMA
1640 6036 KRB3, KRB /READ THE BUFFER AND CLEAR THE FLAGS
1641 3034 DCA RECDAT /SAVE THE WORD
1642 1034 TAD RECDAT /GET THE WORD AND COMPARE IT TO THE WORD
1643 7041 CIA /TRANSMITTED
1644 1031 TAD XMTDAT
1645 7050 CLA /ARE THEY EQUAL?
1646 3264 JMP UPDATE /YES,GO CHECK LOOP SWITCH
1647 7004 LAS /ERROR CHECK SR0
1650 7110 CLA
1651 5262 JMP UPDATE=2
1652 1031 TAD XMTDAT /GET THE FIRST WORD TRANSMITTED
1653 7402 HLT /AC=THE 1ST WORD TRANSMITTED
1654 7200 CLA
1655 1034 TAD RECDAT /
1656 7402 HLT /AC=WORD RECEIVED
1657 7200 CLA
1660 1032 TAD XMTDT1
1661 7402 HLT /AC=2ND WORD TRANSMITTED IF PROGRAM GOT THAT FAR
1662 7240 CLA CMA
1663 3033 DCA ERRFLG /SET ERROR FLAG FOR SCOPE LOOPING
1664 7004 UPDATE, LAS /IS SR1=1

```

```

1665 7004 RA, CLA
1666 7710 SPA SPA
1667 5307 JMP ERRLOP /LOOP ON DATA PATTERN
1670 1032 TAD XMTDT1 /PUT SECOND WORD IN FIRST WORD FOR COMPARISON
1671 3031 DCA XMTDAT /OF NEXT READ
1672 2041 TSTCNT /BUMP THE TEST COUNTER
1673 7410 SKP SKP
1674 5301 JMP END /END OF TEST
1675 1033 TAD ERRFLG /CHECK THE ERROR FLAG FOR RETURN POINTER
1676 7640 SEA CLA
1677 5205 JMP FDATAT+5 /ERROR GO START TEST OVER
1700 5222 JMP XMT+1 /GO TRANSMIT NEXT CHARACTER AND WAIT FOR RECEIVE
1701 4571 END, KSFSPK
1702 4574 EHLTLP /LAST FLAG FAILED TO SET
1703 6036 KRB4, KR3 /CLEAR THE FLAG
1704 6042 TCF10, TCF /CLEAR THE TRANSMIT FLAG
1705 4575 LOOP /LOOP ON TEST IF SR1=1
1706 5775 JMP CHARLG

1707 4577 ERRLOP, DELAY /DELAY 200MS TO ALLOW FLAGS TO SETTLE
1710 6036 KRB5, KR3 /CLEAR RECEIVE FLAG IF SET
1711 6042 TCF11, TCF /CLEAR TRANSMIT FLAG IF SET
1712 7200 CLA CLA
1713 1031 TAD XMTDAT /GET THE FIRST WORD TO TRANSMIT
1714 6046 TLS1, TLS /LOAD AND TRANSMIT IT
1715 6041 TSF24, TSF
1716 5315 JMP .=1 /WAIT FOR THE FIRST TRANSMIT FLAG
1717 7200 CLA CLA
1720 1032 TAD XMTDT1 /GET THE SECOND WORD TO TRANSMIT
1721 6046 TLS2, TLS /LOAD AND TRANSMIT IT
1722 6031 FDTLOP, KSF /WAIT FOR THE FIRST RECEIVE FLAG
1723 5322 JMP .=1
1724 7240 CMA CMA
1725 6036 KRB6, KR3 /READ THE FIRST WORD
1726 3034 DCA RECDAT /SAVE THE FIRST WORD RECEIVED
1727 1034 TAD RECDAT /COMPARE IT TO THE FIRST WORD TRANSMITTED
1730 7041 CIA CIA
1731 1031 TAD XMTDAT
1732 7640 SEA CLA
1733 5363 JMP XSR1LP /ERROR GO CHECK SR1
1734 6041 TSF25, TSF
1735 5334 JMP .=1 /WAIT FOR 2ND TRANSMIT FLAG
1736 1031 TAD XMTDAT /GET THE FIRST WORD AGAIN
1737 6046 TLS3, TLS /LOAD AND TRANSMIT IT
1740 6031 KSF19, KSF
1741 5340 JMP .=1 /WAIT FOR SECOND RECEIVE FLAG
1742 7240 CLA CMA
1743 6036 KRB7, KR3 /READ THE SECOND WORD
1744 3034 DCA RECDAT /IS IT EQUAL TO SECOND TRANSMIT
1745 1034 TAD RECDAT
1746 7041 CIA CIA
1747 1032 TAD XMTDT1
1750 7640 SEA CLA
1751 5363 JMP XSR1LP /ERROR,GO CHECK SR1=1
    
```

```

1752 6041 TSF26, TSF
1753 5352 JMP .=1 /WAIT FOR THE TRANSMIT FLAG
1754 1032 TAD XMTDT1 /GET 2ND WORD AND TRANSMIT IT
1755 6046 TLS4, TLS /LOAD AND TRANSMIT
1756 7604 LAS /CHECK SR1=1 TO LOOP ON TRANSMIT RECEIVE
1757 7004 RA, CLA
1760 7710 SPA SPA
1761 5322 JMP FDTLOP
1762 5205 JMP FDATAT+5

1763 7604 XSR1LP, LAS
1764 7004 RA, RA
1765 7710 SPA SPA
1766 5307 JMP ERRLOP
1767 5270 JMP UPDATE+4

1775 2000
1776 3046
1777 7000
2000 *2000

/ THE FOLLOWING TEST CHECKS THAT THE NUMBER OF DATA BITS WERE
/ SETUP CORRECTLY, TRANSMIT 377 AND TAKE THE 1'S COMPLEMENT
/ OF THE DATA BIT MASK WORD AND CHECK THAT THE AC CAME BACK
/ AS ZEROES;

2000 4576 CHARLG, ST:PPC /STORE THE LOOPING PC FOR ERROR AND TEST LOOPING
2001 1027 TAD DATBIT /SETUP 1'S COMPLEMENT OF SELECTED DATA
2002 7040 CMA CMA /BIT CHARACTER LENGTH
2003 3031 DCA XMTDAT /SAVE IT FOR COMPARISON
2004 1023 TAD K377
2005 6046 TLS5, TLS /TRANSMIT 8 BITS OF ONES
2006 4572 TSFSPK /WAIT FOR THE TRANSMIT FLAG TO SET
2007 4574 EHLTLP /TRANSMIT FLAG FAILED TO SET
2010 6042 TCF12, TCF /CLEAR THE FLAG
2011 4571 KSFSPK /WAIT FOR THE RECEIVE FLAG TO SET
2012 4574 EHLTLP /ERROR,RECEIVE FLAG FAILED TO SET
2013 7240 CLA CMA
2014 6036 KRB8, KR3 /READ THE WORD AND SAVE IT
2015 3034 DCA RECDAT
2016 1034 TAD RECDAT
2017 0031 AND XMTDAT
2020 7450 SNA SNA
2021 5237 JMP .=16
2022 3032 DCA XMTDT1 /SAVE THE BITS THAT WEREN'T SUPPOSED TO BE SELECTED
2023 7604 LAS /HALT ON ERROR?
2024 7710 SPA CLA
2025 5233 JMP .=6 /NO CHECK LOOP SWITCH
2026 1032 TAD XMTDT1
2027 7402 HLT /AC=BITS THAT WEREN'T SUPPOSED TO BE SELECTED
2030 7200 CLA CLA
2031 1027 TAD DATBIT /
2032 7402 HLT /AC=DATA BITS THAT OPERATOR HAD TOLD THE PROGRAM
2033 7604 LAS /THAT WERE SELECTED
2034 7004 RAL RAL
    
```

```

2035 7710 SPA CLA
2036 5204 JMP @MARLG+4 /LOOP ON ERROR
2037 2041 ISE YSTCNT
2040 5204 JMP @MARLG+4
2041 4575 LOOP /LOOP ON TEST IF SR2=1

/FILLER CHARACTER TEST=00 THIS TEST IF OPERATOR HAS SELECTED
/THE FILLER CHARACTER OPTION, THE PROGRAM TRANSMITS A LINE
/FEED AND CHECKS THAT 5 RECEIVE FLAGS COME BACK, THE DATA RECEIVED
/SHOULD BE 1 WORD OF LINE FEED AND 4 WORDS OF FILLER CHARACTERS,
/THE PROGRAM ALSO CHECKS THAT THE TRANSMIT FLAG DOES NOT GET SET
/UNTIL ALL THE RECEIVE FLAGS ARE IN,

2042 1025 FILEXT, TAD @VBTS /CHECK TO SEE IF FILLER CHARACTERS SELECTED
2043 7006 RTL
2044 7700 SMA CLA /HAS IT SELECTED BY THE OPERATOR
2045 5777 JMP @STENAB /NO, GO CHECK FOR STATUS ENABLE
2046 4576 STLPPC /STORE THE LOOPING PC
2047 3036 DCA NDELAY /SETUP PROGRAM DELAY OF 409MS,
2050 4577 DELAY /DELAY TO ALLOW FLAGS TO SETTLE
2051 6036 KRB9, KRB /CLEAR THE RECEIVE FLAG IF SET
2052 7200 CLA
2053 6042 TCF13, TCF /CLEAR THE TRANSMIT FLAG IF SET
2054 1376 TAD (=4 /SETUP A COUNTER TO RECEIVE FOUR FLAGS
2055 3032 DCA XMTDT1 /SAVE IT
2056 1375 TAD (212 /GET LINE FEED
2057 0027 AND OATBIT /MASK OUT TO WORD LENGTH
2060 3031 DCA XMTDAT /SAVE IT FOR COMPARISON OF FIRST WORD
2061 1031 TAD XMTDAT /GET THE LINE FEED CHARACTER
2062 6046 TLS6, TLS /LOAD AND TRANSMIT IT
2063 7200 CLA
2064 1144 FILL0P, TAD (=100 /LOOP TO WAIT FOR THE RECEIVE FLAG AND CHECK XMIT
2065 3040 DCA CNT2
2066 3037 DCA CNT1
2067 6041 TSF20, TSF /SKIP IF TRANSMIT FLAG=1
2070 7610 SKP CLA
2071 4574 EHLTLP /ERROR, TRANSMIT FLAG SET SHOULD GET 4 RECEIVE FLAGS FIRST
2072 6031 KSF21, KSF /SKIP IF RECEIVE FLAG SET
2073 7610 SKP CLA
2074 5302 JMP ,+6
2075 2037 ISE CNT1
2076 5247 JMP ,+7
2077 2040 ISE CNT2
2100 5247 JMP ,+11
2101 4574 EHLTLP /ERROR, RECEIVE FLAG NOT SET OR MISSING SOME
2102 1032 TAD XMTDT1 /SETUP TO COMPARE FOR EITHER A L.F. OR FILLER
2103 1374 TAD (4 /WAS IT THE FIRST CHARACTER
2104 7640 SZA CLA
2105 3031 DCA XMTDAT /NO, THEN CLEAR COMPARE WORD FOR FILLER CHAR
2106 6036 KRB10, KRB /READ THE WORD AND CLEAR THE FLAG
2107 3034 DCA RECDAT /SAVE IT
2110 1034 TAD RECDAT /COMPARE THE WORD RECEIVED WITH WORD EXPECTED
2111 7041 CIA
2112 1031 TAD XMTDAT
2113 7650 SNA CLA

```

```

2114 5332 JMP CNTREC /WORD COMPARES, GO BUMP RECEIVE COUNTER
2115 7604 LAS /ERROR, L.F. OR FILLER CHAR, CAME BACK WRONG
2116 7710 SPA CLA /HALT ON ERROR
2117 5325 JMP CNTREC-5 /NO GO CHECK LOOP SWITCH
2120 1031 TAD XMTDAT /PRESS "CONTINUE" FOR EXPECTED CHARACTER
2121 7402 HLT /AG=WORD EXPECTED X12=L.F OR 0000=FIL CHAR,
2122 7200 CLA
2123 1034 TAD RECDAT /GET THE WORD RECEIVED
2124 7402 HLT /AG=WORD RECEIVED SHOULD BE 212 OR 12 OR 000
2125 7604 LAS /CHECK SR1 TO LOOP ON ERROR
2126 7004 RAL
2127 7710 SPA CLA
2130 5246 JMP FILEXT+4 /LOOP ON THE ERROR
2131 5370 JMP FILEXT /EXIT THE TEST
2132 2032 CNTREC, ISE @BTDY1 /BUMP THE RECEIVE COUNTER
2133 5244 JMP FILL0P /GO GET THE NEXT RECEIVE FLAG
2134 1144 TAD (=100
2135 3040 DCA CNT2
2136 3037 DCA CNT1
2137 6031 KSF22, KSF /CHECK THAT THE RECEIVE FLAG DOESN'T GET SET BEFORE XMIT
2140 7610 SKP CLA
2141 4574 EHLTLP /LAST RECEIVE FLAG SHOULDN'T GET SET UNTIL SOME
/TIME AFTER THE TRANSMIT FLAG
/WAIT FOR THE TRANSMIT FLAG TO GET SET

2142 6041 TSF29, TSF
2143 7610 SKP CLA
2144 5352 JMP ,+6
2145 2037 ISE CNT1
2146 5337 JMP ,+7
2147 2040 ISE CNT2
2150 5337 JMP ,+11
2151 4574 EHLTLP /TRANSMIT FLAG FAILED TO SET AFTER 5 RECEIVE FLAGS
2152 6032 KCC5, KCC /CLEAR THE FLAG
2153 1144 TAD (=100
2154 3040 DCA CNT2
2155 6031 KSF23, KSF
2156 7610 SKP CLA
2157 5365 JMP ,+6
2160 2037 ISE CNT1
2161 5355 JMP ,+4
2162 2040 ISE CNT2
2163 5355 JMP ,+6
2164 4574 EHLTLP /LAST RECEIVE FLAG FAILED TO SET
2165 6032 KCC6, KCC /CLEAR THE RECEIVE FLAG
2166 2041 ISE YSTCNT /BUMP THE TEST COUNTER
2167 5254 JMP FILL0P-10 /GO TRANSMIT ANOTHER FILLER CHARACTER
2170 4575 FILEXT, LOOP /LOOP ON TEST IF SR2=1
2171 5777 JMP @STENAB

2174 0004
2175 0212
2176 7774
2177 2200
2200 *2200

```

/THE FOLLOWING TEST WILL BE EXECUTED ONLY IF THE OPERATOR HAS SET

/THE STATUS ENABLE BIT TO A ONE WHEN HE INITIALIZED THE PROGRAM,
 /THIS TEST WILL CHECK THAT THE ERROR BIT AND THE OVERRUN BIT CAN
 /BE SET AND CLEARED IN THE STATUS REGISTER, THE TEST WILL ALSO
 /CHECK THAT STATUS ENABLE F/F CAN BE SET AND CLEARED, THE RECEIVE
 /BUFFER WILL ALSO BE CHECKED TO CONTAIN THE CORRECT WORD, THREE
 /WORDS WILL BE TRANSMITTED AND THEN THE STATUS AND THE RECEIVE BUFFER
 /WILL BE CHECKED.

2200	1025	STENAB,	TAJ	SAVBTS	/CHECK TO SEE IF STATUS ENABLE WAS SELECTED
2201	7004		RAL		
2202	7700		SMA	CLA	/HAS IT SELECTED BY THE OPERATOR
2203	5323		JMP	SR4HLT	/NO,GO CHECK END OF PROGRAM HALT
2204	4576		STLPPC		/STORE THE LOOPING PC FOR TEST AND SCOPE LOOPING
2205	3036		DCA	NDELAY	/SETUP A DELAY OF 409MS
2206	4577		DELAY		/DELAY TO ALLOW FLAGS TO SETTLE FOR SCOPE LOOPING
2207	6036	KRB12,	KR3		/CLEAR THE RECEIVE FLAG IF SET
2210	6042	TCF15,	TCF		/CLEAR THE TRANSMIT FLAG IF SET
2211	7346		CLA CLL	CMA RTL	/SETUP A COUNT OF 3 TO TRANSMIT 3 TIMES
2212	3032		DCA	XMTDT1	/SAVE IT
2213	7001		IAC		
2214	3031		DCA	XMTDAT	/SET THE FIRST WORD TO BE TRANSMITTED=1
2215	7326		CLA CLL	CML RTL	/SET AC BIT 10 TO A 1
2216	6035	KSE1,	KSE		/AND TRY TO SET STATUS ENABLE TO A 1
2217	7200		CLA		
2220	1031		TAJ	XMTDAT	/GET THE WORD (1,2 OR 3)
2221	6046	TLS9,	TLS		/TRANSMIT IT
2222	7200		CLA		
2223	4572		TSFSKP		/WAIT FOR THE TRANSMIT FLAG
2224	4574		EH_TLP		/ERROR, TRANSMIT FLAG FAILED TO SET
2225	2031		ISE	XMTDAT	/BUMP THE WORD TO A 2 THEN 3
2226	2032		ISE	XMTDT1	/TRANSMITTED 3 WORDS YET
2227	5220		JMP	,=7	/NO,GO TRANSMIT NEXT WORD
2230	6042	TCF16,	TCF		/CLEAR THE TRANSMIT FLAG
2231	6031	KSF25,	KSF		/SKIP ON THE RECEIVE FLAG
2232	4574		EH_TLP		/ERROR, RECEIVE FLAG FAILED TO SET AFTER 3 XMTS
2233	6034	KRS2,	KRS		/DO A STATIC READ OF STATUS AND RECEIVE BUFFER
2234	3034		DCA	RECDAT	/AND SAVE IT
2235	1377		TAJ	(4402	/GET EXPECTED WORD(ERROR-OVERRUN=DATA OF 2)
2236	3031		DCA	XMTDAT	/AND SAVE IT FOR COMPARISON
2237	4274		JMS	STERR	/GO CHECK THE WORDS FOR ERRORS
2240	6035	KSE2,	KSE		/CLEAR STATUS ENABLE AND CHECK BUFFER FOR A 2
2241	6034	KRS3,	KRS		/DO A STATIC READ OF THE RECEIVE BUFFER
2242	3034		DCA	RECDAT	/SAVE THE WORD
2243	7326		CLA CLL	CML RTL	/SETUP FOR WORD EXPECTED
2244	3031		DCA	XMTDAT	/SAVE IT FOR COMPARISON
2245	4274		JMS	STERR	/GO CHECK FOR ERRORS
2246	7326		CLA CLL	CML RTL	/SET AC BIT 10 TO A 1
2247	6035	KSE3,	KSE		/RESET STATUS ENABLE
2250	7200		CLA		
2251	1377		TAJ	(4402	/RESET EXPECTED DATA TO(ERROR-OVERRUN=DATA OF 2)
2252	3031		DCA	XMTDAT	/DO A DYNAMIC READ OF STATUS AND RECEIVE BUFFER
2253	6036	KRB13,	KR3		/SAVE THE WORD READ FOR COMPARISON
2254	3034		DCA	RECDAT	/GO CHECK FOR ERROR CONDITIONS
2255	4274		JMS	STERR	/CHECK THAT RECEIVE FLAG = 0
2256	6031	KSF26,	KSF		

2257	7610		SKP	CLA	
2260	4574		EH_TLP		/ERROR, RECEIVE FLAG SHOULD NOT BE SET YET
2261	4571		KSF25		/GO WAIT FOR THE LAST TRANSMITTED CHARACTER
2262	4574		EH_TLP		/THE THIRD TRANSMIT DID NOT SET RECEIVE FLAG
2263	6036	KRB14,	KR3		/READ THE CHARACTER
2264	3034		DCA	RECDAT	/SAVE IT FOR COMPARISON
2265	1376		TAJ	(3	
2266	3031		DCA	XMTDAT	/SETUP FOR WORD EXPECTED
2267	4274		JMS	STERR	/GO CHECK FOR ERRORS
2270	2041		ISE	STCNT	/IS COUNT EQUAL TO ZERO?
2271	5205		JMP	STENAB=5	/NO GO DO TEST AGAIN
2272	4575		LOOP		/LOOP ON TEST IF SR2=1
2273	5323		JMP	SR4HLT	/GO CHECK END OF PROGRAM HALT
2274	0000	STERR,	0		/CHECK TO SEE IF ERROR EXIST IN STATUS REGISTER AND RECEIVE BUFFER
2275	1034		TAJ	RECDAT	/GET THE WORD RECEIVED AND COMPARE IT WITH
2276	7041		CIA		/THE WORD EXPECTED
2277	1031		TAJ	XMTDAT	/
2300	7650		SNA	CLA	/ARE THEY EQUAL?
2301	5674		JMP	I STERR	/YES, CONTINUE TESTING
2302	7604		LAS		/HALT ON ERROR
2303	7710		SPA	CLA	
2304	5316		JMP	,*12	/NO,GO CHECK LOOP SWITCH
2305	7240		CLA	CHA	
2306	1274		TAJ	STERR	
2307	7402		HLT		/AC=PC WERE ERROR WAS DETECTED AT
2310	7200		CLA		
2311	1031		TAJ	XMTDAT	/GET THE WORD EXPECTED
2312	7402		HLT		/AC=WORD EXPECTED
2313	7200		CLA		
2314	1034		TAJ	RECDAT	/GET THE WORD RECEIVED
2315	7402		HLT		/AC=WORD RECEIVED
2316	7604		LAS		/LOOP ON ERROR?
2317	7004		RAL		
2320	7710		SPA		
2321	5204		JMP	STENAB+4	/YES GO LOOP
2322	5674		JMP	I STERR	/NO,GO GET NEXT ERROR
/HALT AT END OF PROGRAM IF SWITCH REGISTER 4 EQUALS A ONE					
2323	7604	SR4HLT,	LAS		
2324	0375		AND	(200	
2325	7650		SNA	CLA	
2326	5774		JMP	CLRBRD	
2327	7402		HLT		/END OF THE PROGRAM SR4=1
2330	5774		JMP	CLRBRD	/PRESS CONTINUE TO GO ON
2374	0400				
2375	0200				
2376	0003				
2377	4402				
	2400	*2400			

```

/INHIBIT ERROR HALT IF SR0=1 AND LOOP ON ERROR IF SR1=1
2400 0000 HLTLOP, 0
2401 7004 LAS /HALT ON ERROR?
2402 7710 SPA CLA
2403 5207 JMP ,+4 /NO,CHECK LOOP SWITCH
2404 7240 CLA CMA
2405 1200 TAD HLTLOP /GET THE FAILING PC WHERE THE ERROR WAS DETECTED
2406 7402 HLT /AC=FAILING PC WHERE ERROR WAS DETECTED
2407 7004 LAS /LOOP ON THE ERROR?
2410 7004 RAL
2411 7710 SPA CLA
2412 5435 JMP I LOOPPC
2413 5000 JMP I HLTLOP

2414 0000 HLTLP1, 0 /ROUTINE USED IN AGNSKP TEST TO SAVE AC FOR FALSE SKIPPING
2415 3232 DCA SAVAC /SAVE THE AC FOR NON LOOPING PURPOSES
2416 7004 LAS /HALT ON ERROR?
2417 7710 SPA CLA
2420 5224 JMP ,+4 /NO GO CHECK LOOP SWITCH
2421 7240 CLA CMA
2422 1214 TAD HLTLP1 /GET THE FAILING PC WHERE ERROR WAS DETECTED
2423 7402 HLT /AC=FAILING PC WHERE ERROR WAS DETECTED
2424 7004 LAS /LOOP ON THE ERROR?
2425 7004 RAL
2426 7710 SPA CLA
2427 5435 JMP I LOOPPC /YES,LOOP ON THE ERROR
2430 1232 TAD SAVAC /NO,RESET THE AC AND CONTINUE
2431 5614 JMP I HLTLP1 /RETURN AND CHECK THE EFFECT OF THE IOT ON AC

2432 0000 SAVAC, 0

2433 0000 HLTLP2, 0 /THIS ROUTINE USED ONLY WHEN IOT EFFECTS CONTENTS OF AC
2434 3232 DCA SAVAC /SAVE THE AC FOR ERROR INDICATION
2435 7004 LAS /HALT ON THE ERROR?
2436 7710 SPA CLA
2437 5246 JMP ,+7 /NO,GO CHECK LOOP SWITCH
2440 7240 CLA CMA /GET THE FAILING PC WHERE ERROR WAS DETECTED
2441 1233 TAD HLTLP2 /
2442 7402 HLT /AC=FAILING PC WHERE ERROR WAS DETECTED
2443 7200 CLA /PRESS "CONT" TO GET CONTENTS OF AC AFTER EXECUTION
/AND COMPARISON OF THE IOT AND AC

2444 1232 TAD SAVAC
2445 7402 HLT /AC=BITS THAT WERE EFFECTED AFTER EXECUTION OF IOT
2446 7004 LAS /LOOP ON THE ERROR?
2447 7004 RAL
2450 7710 SPA CLA
2451 5435 JMP I LOOPPC /YES,GO LOOP ON THE ERROR
2452 5633 JMP I HLTLP2 /RETURN AND CONTINUE THE TEST

```

/BAUD RATE TIMING TEST-LOAD AND START 202 AND WATCH YOUR WATCH
/FOR 30 SECONDS, THE PROCESSOR SHOULD HALT IN 30 SECONDS, IF IT
/DNEST CHECK THE BAUD RATE WITH A SCOPE OR CHECK THE BAUD RATE
/SWITCHES

```

2453 1377 BAUDTM, TAD (JMP I 2
2454 3001 DCA 1
2455 1376 TAD (RETINT /SETUP RETURN POINTER FOR THE INTERRUPT
2456 3002 DCA 2
2457 1030 TAD BAUDNO /GET THE BAUD RATE
2460 1375 TAD (BAUDTB /GET THE ADDRESS OF THE BAUD RATE TABLE
2461 3322 DCA BDPNTR /SAVE THE POINTER TO THE BAUD RATE TABLE
2462 1026 TAD BITNO /GET THE CHARACTER LENGTH
2463 7104 CLL RAL /MULTIPLY IT BY 2
2464 1722 TAD I BDPNTR /ADD IN BAUD RATE ADDRESS
2465 3322 DCA BDPNTR /ADDRESS OF BAUD RATE CONSTANTS ARE READY TO BE SETUP
2466 4976 STLPPC /STORE THE LOOPING PC FOR TEST LOOPING
2467 1322 TAD BDPNTR /GET POINTER ADDRESS TO THE CONSTANTS
2470 3041 DCA TSTCNT /SAVE IT IN TEST COUNT
2471 1441 TAD I TSTCNT /GET THE FIRST COUNSTANT
2472 3037 DCA CNT1
2473 2041 ISZ TSTCNT /BUMB THE ADDRESS BY 1 FOR NEXT CONSTANT
2474 1441 TAD I TSTCNT
2475 3040 DCA CNT2
2476 6046 TLS7, TLS /LOAD AND TRANSMIT FIRST CHARACTER=FLAG
2477 6041 TSF30, TSF /COMES UP ALMOST IMMEDIATELY
2500 5277 JMP ,+1
2501 5304 JMP ,+3
2502 6036 INTON, KR3 /CLEAR THE RECEIVE FLAG
2503 7010 SKP CLA
2504 6046 TLS8, TLS /LOAD AND TRANSHIT NEXT CHARACTER AND CLEAR FLAG
2505 6001 ION /TURN THE INTERRUPT ON
2506 6031 RETINT, KSP /SKIP IF RECEIVE FLAG SET
2507 7010 SKP /FLAG NOT SET,CHECK TRANSMIT FLAG
2510 5302 JMP INTON /RECEIVE FLAG SET,GO CLEAR IT
2511 6041 TSF31, TSF /SKIP IF TRANSMIT FLAG SET
2512 5306 JMP INTON+4 /NOT SET YET GO WAIT FOR A FLAG
2513 2037 ISZ CNT1 /BUMP THE FIRST COUNTER
2514 5304 JMP INTON+2 /GO TRANSMIT ANOTHER CHARACTER
2515 2040 ISZ CNT2 /FIRST COUNTER OVERFLOWED
2516 5304 JMP INTON+2 /GO DO ANOTHER 4095 INTERRUPTS
2517 7402 HLT /TRANSHITTED FOR 30 SECONDS????
2520 4575 LOOP /LOOP ON TEST IF SR2=1
2521 5317 JMP ,+2 /END OF THE TEST
2522 0000 BDPNTR, 0

/POINTERS TO BAUD RATE TABLE
2523 2600 BAUDTB, BR110
2524 2614 BR150
2525 2630 BR300
2526 2644 BR600
2527 2660 BR1200
2530 2674 BR2400
2531 2710 BR4800
2532 2724 BR9600
2533 2740 BR19200
2534 2754 BR360
2535 2770 BR667
2536 3004 BR1050

```

```

/POINTERS TO BAUD RATE TYPEOUTS
2537 3622 MESTAB, MESS6A
2540 3647 MESS6B
2541 3674 MESS6C
2542 3721 MESS6D
2543 3746 MESS6E
2544 3774 MESS6F
2545 4022 MESS6G
2546 4090 MESS6H
2547 4076 MESS6I
2550 4125 MESS6J
2551 4153 MESS6K
2552 4201 MESS6L

/POINTERS TO DATA BIT TYPEOUTS
2553 4274 MESTB1, MES10A
2554 4313 MES10B
2555 4332 MES10C
2556 4351 MES10D

2575 2523
2576 2506
2577 5402
2600 *2600

/BAUD RATE CONSTANTS FOR 110 BAUD
2600 7051 BR110, -727 /7 BITS AT 15,71 CHAR/SEC=471 CHAR/30 SEC
2601 7777 -1
2602 7143 -635 /8 BITS AT 13,75 CHAR/SEC=413 CHAR/30 SEC
2603 7777 -1
2604 7221 -557 /9 BITS AT 12,22 CHAR/SEC=367 CHAR/30 SEC
2605 7777 -1
2606 7266 -512 /10 BITS AT 11 CHAR/SEC=330 CHAR/30 SEC
2607 7777 -1
2610 7324 -424 /11 BITS AT 10 CHAR/SEC=300 CHAR/30 SEC
2611 7777 -1
2612 7355 -423 /12 BITS AT 9,17 CHAR/SEC=275 CHAR/30 SEC
2613 7777 -1

/BAUD RATE CONSTANTS FOR 150 BAUD
2614 6575 BR150, -1203 /7 BITS AT 21,43 CHAR/SEC=643 CHAR/30 SEC
2615 7777 -1
2616 6715 -1363 /8 BITS AT 18,75 CHAR/SEC=563 CHAR/30 SEC
2617 7777 -1
2620 7014 -764 /9 BITS AT 16,67 CHAR/SEC=500 CHAR/30 SEC
2621 7777 -1
2622 7076 -702 /10 BITS AT 15 CHAR/SEC=450 CHAR/30 SEC
2623 7777 -1
2624 7147 -631 /11 BITS AT 13,64 CHAR/SEC=409 CHAR/30 SEC
2625 7777 -1
2626 7211 -567 /12 BITS AT 12,50 CHAR/SEC=375 CHAR/30 SEC
2627 7777 -1
    
```

```

/BAUD RATE CONSTANTS FOR 300 BAUD
2630 5372 BR300, -2406 /7 BITS AT 42,86 CHAR/SEC=1286 CHAR/30 SEC
2631 7777 -1
2632 5633 -2145 /8 BITS AT 37,50 CHAR/SEC=1125 CHAR/30 SEC
2633 7777 -1
2634 6030 -1750 /9 BITS AT 33,33 CHAR/SEC=1000 CHAR/30 SEC
2635 7777 -1
2636 6174 -1604 /10 BITS AT 30,00 CHAR/SEC=900 CHAR/30 SEC
2637 7777 -1
2640 6316 -1462 /11 BITS AT 27,27 CHAR/SEC=818 CHAR/30 SEC
2641 7777 -1
2642 6422 -1356 /12 BITS AT 25,00 CHAR/SEC=750 CHAR/30 SEC
2643 7777 -1

/BAUD RATE CONSTANTS FOR 600 BAUD
2644 2765 BR600, -5313 /7 BITS AT 85,71 CHAR/SEC=2571 CHAR/30 SEC
2645 7777 -1
2646 3466 -4312 /8 BITS AT 75,00 CHAR/SEC=2250 CHAR/30 SEC
2647 7777 -1
2650 4060 -3720 /9 BITS AT 66,67 CHAR/SEC=2000 CHAR/30 SEC
2651 7777 -1
2652 4370 -3410 /10 BITS AT 60,00 CHAR/SEC=1800 CHAR/30 SEC
2653 7777 -1
2654 4633 -3145 /11 BITS AT 54,55 CHAR/SEC=1637 CHAR/30 SEC
2655 7777 -1
2656 5044 -2734 /12 BITS AT 50,00 CHAR/SEC=1500 CHAR/30 SEC
2657 7777 -1

/BAUD RATE CONSTANTS FOR 1200 BAUD
2660 5750 BR1200, -2330 /7 BITS AT 171,43 CHAR/SEC=5143 CHAR/30 SEC
2661 7776 -2
2662 7153 -625 /8 BITS AT 150 CHAR/SEC=4500 CHAR/30 SEC
2663 7776 -2
2664 8140 -7640 /9 BITS AT 133,33 CHAR/SEC=4000 CHAR/30 SEC
2665 7777 -1
2666 8760 -7020 /10 BITS AT 120 CHAR/SEC=3600 CHAR/30 SEC
2667 7777 -1
2670 1467 -6311 /11 BITS AT 109,09 CHAR/SEC=3273 CHAR/30 SEC
2671 7777 -1
2672 2110 -5670 /12 BITS AT 100 CHAR/SEC=3000 CHAR/30 SEC
2673 7777 -1

/BAUD RATE CONSTANTS FOR 2400 BAUD
2674 3720 BR2400, -4360 /7 BITS AT 342,86 CHAR/SEC=10,286 CHAR/30 SEC
2675 7775 -3
2676 6326 -1452 /8 BITS AT 300 CHAR/SEC=9000 CHAR/30 SEC
2677 7775 -3
2700 8277 -7901 /9 BITS AT 266,67 CHAR/SEC=8000 CHAR/30 SEC
2701 7776 -2
2702 1737 -6341 /10 BITS AT 240 CHAR/SEC=7200 CHAR/30 SEC
    
```

2703 7776 =2
 2704 3156 =4822 /11 BITS AT 218,18 CHAR/SEC=6545 CHAR/30 SEC
 2705 7776 =2
 2706 4217 =3561 /12 BITS AT 200 CHAR/SEC=6000 CHAR/30 SEC
 2707 7776 =2

/BAUD RATE CONSTANTS FOR 4800 BAUD

2710 7640 BR4800, =140 /7 BITS AT 689,71 CHAR/SEC=20,571 CHAR/30 SEC
 2711 7772 =6
 2712 4654 =3124 /8 BITS AT 600 CHAR/SEC=18,000 CHAR/30 SEC
 2713 7773 =5
 2714 0575 =7203 /9 BITS AT 333,33 CHAR/SEC=16,000 CHAR/30 SEC
 2715 7774 =4
 2716 3675 =4103 /10 BITS AT 400 CHAR/SEC=14,400 CHAR/30 SEC
 2717 7774 =4
 2720 6332 =1446 /11 BITS AT 436,36 CHAR/SEC=13,091 CHAR/30 SEC
 2721 7774 =4
 2722 0436 =7342 /12 BITS AT 400 CHAR/SEC=12000 CHAR/30 SEC
 2723 7775 =3

/BAUD RATE CONSTANTS FOR 9600 BAUD

2724 7477 BR9600, =301 /7 BITS AT 1371,43 CHAR/SEC=41,143 CHAR/30 SEC
 2725 7765 =13
 2726 1530 =6250 /8 BITS AT 1200 CHAR/SEC=36,000 CHAR/30 SEC
 2727 7767 =11
 2730 1371 =6407 /9 BITS AT 1066,67 CHAR/SEC=32000 CHAR/30 SEC
 2731 7770 =10
 2732 7571 =207 /10 BITS AT 960 CHAR/SEC=28,800 CHAR/30 SEC
 2733 7770 =10
 2734 4664 =3114 /11 BITS AT 872,73 CHAR/SEC=26,182 CHAR/30 SEC
 2735 7771 =7
 2736 1073 =6705 /12 BITS AT 800 CHAR/SEC=24,000 CHAR/30 SEC
 2737 7772 =6

/BAUD RATE CONSTANTS FOR 19,2 KILO BAUD

2740 7176 B19200, =602 /7 BITS AT 2742,86 CHAR/SEC=82,286 CHAR/30 SEC
 2741 7753 =25
 2742 3297 =4521 /8 BITS AT 2400 CHAR/SEC=72,000 CHAR/30 SEC
 2743 7756 =22
 2744 2761 =5017 /9 BITS AT 2133,33 CHAR/SEC=64,000 CHAR/30 SEC
 2745 7760 =20
 2746 7302 =416 /10 BITS AT 1920 CHAR/SEC=57,600 CHAR/30 SEC
 2747 7761 =17
 2750 1590 =6230 /11 BITS AT 1745,45 CHAR/SEC=52,364 CHAR/30 SEC
 2751 7763 =15
 2752 2165 =5613 /12 BITS AT 1600 CHAR/SEC=48,000 CHAR/30 SEC
 2753 7764 =14

/BAUD RATE CONSTANTS FOR 56,8 BAUD

2754 7415 BR568, =353 /7 BITS AT 8,11 CHAR/SEC =243 CHAR/30 SEC

2755 7777 =1
 2756 7453 =325 /8 BITS AT 7,10 CHAR/SEC =213 CHAR/30 SEC
 2757 7777 =1
 2760 7503 =275 /9 BITS AT 6,31 CHAR/SEC =189 CHAR/30 SEC
 2761 7777 =1
 2762 7526 =252 /10 BITS AT 5,68 CHAR/SEC =170 CHAR/30 SEC
 2763 7777 =1
 2764 7545 =233 /11 BITS AT 5,16 CHAR/SEC =155 CHAR/30 SEC
 2765 7777 =1
 2766 7542 =216 /12 BITS AT 4,73 CHAR/SEC =142 CHAR/30 SEC
 2767 7777 =1

/BAUD RATE CONSTANTS FOR 66,7 BAUD

2770 7342 BR667, =436 /7 BITS AT 9,53 CHAR/SEC =286/30 SEC
 2771 7777 =1
 2772 7406 =372 /8 BITS AT 8,35 CHAR/SEC =250/30 SEC
 2773 7777 =1
 2774 7442 =336 /9 BITS AT 7,41 CHAR/SEC =222/30 SEC
 2775 7777 =1
 2776 7470 =310 /10 BITS AT 6,67 CHAR/SEC =200/30 SEC
 2777 7777 =1
 3000 7512 =266 /11 BITS AT 6,06 CHAR/SEC =182/30 SEC
 3001 7777 =1
 3002 7531 =247 /12 BITS AT 5,56 CHAR/SEC =167/30 SEC
 3003 7777 =1

/BAUD RATE CONSTANTS FOR 1050 BAUD

3004 7154 BR1050, =624 /7 BITS AT 150 CHAR/SEC =4500/30 SEC
 3005 7776 =2
 3006 0236 =7342 /8 BITS AT 131,25 CHAR/SEC =3938/30 SEC
 3007 7777 =1
 3010 1124 =6054 /9 BITS AT 116,66 CHAR/SEC =3500/30 SEC
 3011 7777 =1
 3012 1642 =6116 /10 BITS AT 105 CHAR/SEC =3150/30 SEC
 3013 7777 =1
 3014 2320 =5460 /11 BITS AT 99,45 CHAR/SEC =2864/30 SEC
 3015 7777 =1
 3016 2677 =5101 /12 BITS AT 87,5 CHAR/SEC =2625/30 SEC
 3017 7777 =1

3020 0000 XDELAY, 0
 3021 7300 CLA CLL
 3022 1036 TAD NDELAY
 3023 3243 DCA DELAYN
 3024 1244 DELLOP, TAD CON100
 3025 3245 DCA US100
 3026 2245 ISE US100
 3027 5226 JMP ,=1
 3030 7200 CLA
 3031 7200 CLA
 3032 7200 CLA
 3033 7200 CLA
 3034 2243 ISE DELAYN

```

3035 7610 SKP CLA
3036 5620 JHP I XDELAY
3037 0620 AND I XDELAY
3040 0620 AND I XDELAY
3041 0220 AND XDELAY
3042 5224 JHP DELLOP

3043 0000 DELAYN, 0
3044 7754 CON100, =24
3045 0000 US100, 0
    
```

/IF FILLER CHARACTER OPTION IS SELECTED-DO NOT TRANSMIT A L,F;
/FILLER CHARACTERS WILL BE CHECKED LATER,

```

3046 0000 FILCHK, 0
3047 3270 DCA CHKFIL
3050 1025 TAD SAVBTS /HAS FILLER CHARACTER OPTION SELECTED
3051 7006 RTL /BY THE OPERATOR
3052 7700 CLA
3053 5253 JHP ,+10 /NO,EXIT AND TRANSMIT THE CHARACTER
3054 1270 TAD CHKFIL /CHECK TO SEE IF THE WORD TO BE TRANSMITTED IS A
3055 1377 TAD (=12 /LINE FEED
3056 7450 SNA /HAS IT A 12
3057 5255 JHP ,+6
3060 1376 TAD (=200 /HAS IT A 212
3061 7650 SNA CLA
3062 5255 JHP ,+3 /YES IT WAS A LINE FEED
3063 2246 ISZ FILCHK /WORD IS OK,GO TRANSMIT IT
3064 5646 JHP I FILCHK
3065 2041 ISZ TSTCNT /BUMP TEST COUNTER TO GET RID OF WORD
3066 5646 JHP I FILCHK /GO GET ANOTHER WORD
3067 5646 JHP I FILCHK /GO DO TEST OVER IF LAST WORD IS A LINE FEED

3070 0000 CHKFIL, 0
3071 0000 WATTSF, 0 /ROUTINE TO WAIT FOR TRANSMIT FLAG,
3072 1144 TAD C=100 /IF IT DOESN'T SET IN A SECOND OR SO
3073 3040 DCA CNT2 /TIMEOUT AND HALT
3074 3037 DCA CNT1
3075 4041 TSF15, TSF
3076 7610 SKP CLA
3077 5305 JHP ,+6
3100 2037 ISZ CNT1
3101 5275 JHP ,+4
3102 2040 ISZ CNT2
3103 5275 JHP ,+6
3104 5671 JHP I WATTSF /TSF FAILED TO SKIP
3105 2271 ISZ WATTSF
3106 5671 JHP I WATTSF /OK,TSF SKIPPED RETURN

3107 0000 WATKSF, 0 /ROUTINE TO WAIT FOR THE RECEIVE FLAG
3110 1144 TAD C=100 /IF IT DOESN'T SET IN A SECOND OR SO
3111 3040 DCA CNT2 /TIMEOUT AND HALT,
3112 3037 DCA CNT1
    
```

```

3113 6031 KSF8, KSF
3114 7610 SKP CLA
3115 5323 JHP ,+6
3116 2037 ISZ CNT1
3117 5343 JHP ,+4
3120 2040 ISZ CNT2
3121 5343 JHP ,+6
3122 5707 JHP I WATKSF /ERROR,KSF FAILED TO SKIP
3123 2307 ISZ WATKSF /
3124 5707 JHP I WATKSF /OK,RECEIVE FLAG SKIPPED

3176 7600
3177 7766
3200 3200 *3200
    
```

```

3200 4567 TYINTR, /INTERAGATION SETUP FOR THE TELETYPE
3201 3400 MESSAGE /TYPE RECEIVE IOT?
3202 4565 MESS1
3203 5200 TWC0CT /GET RECEIVE DEVICE CODE
3204 4547 JHP ,=3 /INPUT ERROR
3205 3024 BSWAP /SWAP IT AROUND TO BITS 0-5
3206 4567 DCA DEVCOD /SAVE THE RECEIVE DEVICE CODE
3207 3410 MESSAGE /TYPE TRANSMIT IOT
3210 4565 MESS2
3211 5206 TWC0CT /GET TRANSMIT IOT
3212 1024 JHP ,=3 /INPUT ERROR
3213 3024 TAD DEVCOD /ADD TRANSMIT IOT TO RECEIVE IOT
3214 4567 DCA DEVCOD /SAVE THE IOTS
3215 3421 MESSAGE /TYPE PARITY(Y OR N)?
3216 4593 YESRNO
3217 5214 JHP ,=3 /WAIT FOR A YES OR NO
3220 7610 SKP CLA /NOT A YES OR N
3221 7330 CLA CLL CML RAR /SET NO PARITY BIT
3222 3025 DCA SAVBTS /SET THE PARITY BIT TO A 1
3223 4567 MESSAGE /SAVE THE PARITY BIT IN STATUS WORD
3224 3433 MESS3A
3225 1025 TAD SAVBTS
3226 7710 SPA CLA
3227 7001 IAC
3230 4563 PRNT1
3231 4567 MESSAGE /TYPE STATUS REGISTER(Y OR N)
3232 3436 MESS4
3233 4593 YESRNO
3234 5231 JHP ,=3 /WAIT FOR A YES OR NO
3235 7610 SKP CLA /NOT A YES OR NO
3236 7332 CLA CLL CML RTR /NO STATUS REGISTER SELECTED
3237 1025 TAD SAVBTS /STATUS REGISTER IS SELECTED
3240 3025 DCA SAVBTS /ADD STATUS REGISTER BIT TO WORD
3241 4567 MESSAGE /AND SAVE IT
3242 3477 MESS4A
3243 1025 TAD SAVBTS
3244 7004 RAL
3245 7710 SPA CLA
3246 7001 IAC
    
```

```

3247 4563 PRNT1
3250 4567 MESSAGE /TYPE FILLER CHARACTERS(Y OR N)?
3251 3503 MESS5
3252 4553 YESRNO /WAIT FOR A YES OR NO
3253 5250 JMP ,=3 /NOT A YES OR NO
3254 7610 SKP CLA /NO FILLER CHARACTERS
3255 1377 TAD (1000 /YES,FILLER CHARACTERS
3256 1025 TAD SAVBTS /ADD THE FILLER CHARACTER BIT TO STATUS WORD
3257 3025 DCA SAVBTS /
3260 4567 MESSAGE
3261 3523 MESS5A
3262 1025 TAD SAVBTS
3263 7006 RTL
3264 7710 SPA CLA
3265 7001 IAC
3266 4563 PRNT1
3267 4567 MESSAGE /TYPE BAUD RATE(00=13)?
3270 3527 MESS6
3271 4567 MESSAGE
3272 3547 CMESS6
3273 4565 THDOCT /INPUT A NUMBER FROM 00=13
3274 5247 JMP ,=5 /INPUT ERROR
3275 3364 DCA SAVIT /SAVE THE NUMBER TYPED BY OPERATOR
3276 1364 TAD SAVIT /HAS THE NUMBER WITHIN BAUD RATE LIMITS
3277 1376 TAD (=13
3300 7740 SMA SZA CLA
3301 5247 JMP ,=12 /NOT WITHIN LIMITS GO TYPE MESSAGE OVER
3302 1364 TAD SAVIT /GET THE NUMBER AND PUT IN SAVBTS
3303 7106 CLL RTL
3304 7004 RAL /PUT NUMBER IN BITS 5 6 7 AND 8
3305 1025 TAD SAVBTS /ADD IT TO THE STATUS WORD
3306 3025 DCA SAVBTS /AND SAVE IT
3307 1025 TAD SAVBTS
3310 7012 RTR
3311 7010 RAR
3312 0375 AND (17
3313 1374 TAD (MESTAB
3314 3320 DCA CHGMES
3315 1720 TAD I CHGMES
3316 3320 DCA CHGMES
3317 4567 MESSAGE
CHGMES, MESS6A
3320 3022 MESSAGE /TYPE TWO STOP BITS?
3321 4567 MESS7
3322 4227 MESS7
3323 4553 YESRNO /WAIT FOR A Y OR N
3324 5321 JMP ,=3 /INPUT ERROR
3325 7610 SKP CLA /ONLY 1 STOP BIT
3326 1373 TAD (4 /2 STOP BITS
3327 1025 TAD SAVBTS /ADD THE NUMBER OF STOP BITS
3330 3025 DCA SAVBTS /TO THE STATUS WORD
3331 4567 MESSAGE
3332 4244 MESS7A
3333 1025 TAD SAVBTS
3334 0373 AND (4
3335 7650 SMA CLA
    
```

```

3336 7001 IAC
3337 4563 PRNT1
3340 4567 MESSAGE /TYPE # OF DATA BITS (0,1,2,3)
3341 4247 MESS10
3342 4566 ONEOCT /WAIT FOR A NUMBER
3343 5340 JMP ,=3 /NOT A NUMBER
3344 3320 DCA CHGMES /SAVE THE NUMBER
3345 1372 TAD (=3 /IS IT GREATER THAN 3
3346 1320 TAD CHGMES
3347 7740 SMA SZA CLA
3350 5340 JMP ,=10 /YES,TRY AGAIN
3351 1320 TAD CHGMES /
3352 1025 TAD SAVBTS /ADD IT TO STATUS WORD
3353 3025 DCA SAVBTS /AND SAVE IT
3354 1371 TAD (MESTB1
3355 1320 TAD CHGMES
3356 3362 DCA ,=4
3357 1762 TAD I ,=3
3360 3362 DCA ,=2
3361 4567 MESSAGE
3362 4274 MESS10A
3363 5770 JMP SETUP /GO AND CALCULATE IT

3364 0000 SAVIT, 0

3370 0215
3371 2553
3372 7775
3373 0004
3374 2537
3375 0017
3376 7765
3377 1000
3400 3400 *3400

3400 4322 MESS1, TEXT "#RECEIVE IOT? "
3401 0503
3402 0511
3403 2605
3404 4011
3405 1724
3406 7740
3407 4000
3410 4324 MESS2, TEXT "#TRANSMIT IOT? "
3411 2201
3412 1023
3413 1511
3414 2440
3415 1117
3416 2477
3417 4040
3420 0000
3421 4320 MESS3, TEXT "#PARITY(Y OR N)? "
3422 0122
3423 1124
    
```

3424	3150		
3425	3140		
3426	1722		
3427	4016		
3430	5177		
3431	4040		
3432	0000		
3433	4040	MESS3A, TEXT	" NP="
3434	1625		
3435	7500		
3436	7743	MESS4, TEXT	"?#EVEN PARITY EVN=0? ODD PARITY EVN=1?#STATUS ENABLED(Y OR N)? "
3437	0526		
3440	0516		
3441	4020		
3442	0122		
3443	1124		
3444	3140		
3445	0526		
3446	1675		
3447	4077		
3450	4040		
3451	1704		
3452	0440		
3453	2001		
3454	2211		
3455	2431		
3456	4005		
3457	2616		
3460	7561		
3461	7743		
3462	2324		
3463	0124		
3464	2523		
3465	4005		
3466	1601		
3467	0214		
3470	0504		
3471	5031		
3472	4017		
3473	2240		
3474	1651		
3475	7740		
3476	4000		
3477	4040	MESS4A, TEXT	" SHD="
3500	2327		
3501	0475		
3502	0000		
3503	7743	MESS5, TEXT	"?#FILLER CHARACTERS(Y OR N)? "
3504	0611		
3505	1414		
3506	0522		
3507	4003		
3510	1001		
3511	2201		
3512	0324		

3513	0522		
3514	2350		
3515	3140		
3516	1722		
3517	4016		
3520	5177		
3521	4040		
3522	0000		
3523	4040	MESS5A, TEXT	" FIL="
3524	0611		
3525	1475		
3526	0000		
3527	7743	MESS6, TEXT	"?#BAUD RATE(00=13)? 00=110 01=150 02=300 03=600 04=1200 05=2400"
3530	0201		
3531	2504		
3532	4022		
3533	0124		
3534	0550		
3535	0060		
3536	5561		
3537	6351		
3540	7740		
3541	6060		
3542	7561		
3543	6160		
3544	4060		
3545	6175		
3546	6165		
3547	6040		
3550	6062		
3551	7563		
3552	6060		
3553	4060		
3554	6375		
3555	6660		
3556	6040		
3557	6064		
3560	7561		
3561	6260		
3562	6040		
3563	6065		
3564	7562		
3565	6460		
3566	6000		
3567	4360	MESS6, TEXT	"#06=4800 07=9600 10=19,200 11=56,8 12=66,7 13=1050 "
3570	6675		
3571	6470		
3572	6060		
3573	4060		
3574	6775		
3575	7166		
3576	6060		
3577	4061		
3600	6075		

3601 6171
 3602 5462
 3603 6060
 3604 4061
 3605 6175
 3606 6566
 3607 5670
 3610 4061
 3611 6275
 3612 6666
 3613 5667
 3614 4061
 3615 6375
 3616 6160
 3617 6560
 3620 4040
 3621 0000

MESS6A, TEXT "#110 BAUD = B1=0? B2=0? B3=0? W2=1? W5=0?"

3622 4361
 3623 6160
 3624 4002
 3625 0125
 3626 0440
 3627 5540
 3630 0261
 3631 7560
 3632 7740
 3633 0262
 3634 7560
 3635 7740
 3636 0263
 3637 7560
 3640 7740
 3641 2762
 3642 7561
 3643 7740
 3644 2765
 3645 7560
 3646 7700
 3647 4361
 3650 6560
 3651 4002
 3652 0125
 3653 0440
 3654 5540
 3655 0261
 3656 7560
 3657 7740
 3660 0262
 3661 7560
 3662 7740
 3663 0263
 3664 7561
 3665 7740
 3666 2762

MESS6B, TEXT "#150 BAUD = B1=0? B2=0? B3=1? W2=1? W5=0?"

3667 7561
 3670 7740
 3671 2765
 3672 7560
 3673 7700
 3674 4363
 3675 6060
 3676 4002
 3677 0125
 3700 0440
 3701 5540
 3702 0261
 3703 7560
 3704 7740
 3705 0262
 3706 7561
 3707 7740
 3710 0263
 3711 7560
 3712 7740
 3713 2762
 3714 7561
 3715 7740
 3716 2765
 3717 7560
 3720 7700
 3721 4366
 3722 6060
 3723 4002
 3724 0125
 3725 0440
 3726 5540
 3727 0261
 3730 7560
 3731 7740
 3732 0262
 3733 7561
 3734 7740
 3735 0263
 3736 7561
 3737 7740
 3740 2762
 3741 7561
 3742 7740
 3743 2765
 3744 7560
 3745 7700
 3746 4361
 3747 6260
 3750 6040
 3751 0201
 3752 2504
 3753 4055
 3754 4002
 3755 6175

MESS6C, TEXT "#300 BAUD = B1=0? B2=1? B3=0? W2=1? W5=0?"

MESS6D, TEXT "#600 BAUD = B1=0? B2=1? B3=1? W2=1? W5=0?"

MESS6E, TEXT "#1200 BAUD = B1=1? B2=0? B3=0? W2=1? W5=0?"

3756 6177
 3757 4002
 3760 6275
 3761 6077
 3762 4002
 3763 6375
 3764 6077
 3765 4027
 3766 6275
 3767 6177
 3770 4027
 3771 6575
 3772 6077
 3773 0000
 3774 4362
 3775 6460
 3776 6040
 3777 0201
 4000 2504
 4001 4055
 4002 4002
 4003 6175
 4004 6177
 4005 4002
 4006 6275
 4007 6077
 4010 4002
 4011 6375
 4012 6177
 4013 4027
 4014 6275
 4015 6177
 4016 4027
 4017 6575
 4020 6077
 4021 0000
 4022 4364
 4023 7060
 4024 6040
 4025 0201
 4026 2504
 4027 4055
 4030 4002
 4031 6175
 4032 6177
 4033 4002
 4034 6275
 4035 6177
 4036 4002
 4037 6375
 4040 6077
 4041 4027
 4042 6275
 4043 6177
 4044 4027

MESS6F, TEXT "#2400 BAUD - B1=1? B2=0? B3=1? W2=1? W5=0?"

MESS6G, TEXT "#4800 BAUD - B1=1? B2=1? B3=0? W2=1? W5=0?"

4045 6575
 4046 6077
 4047 0000
 4050 4371
 4051 6660
 4052 6040
 4053 0201
 4054 2504
 4055 4055
 4056 4002
 4057 6175
 4060 6177
 4061 4002
 4062 6275
 4063 6177
 4064 4002
 4065 6375
 4066 6177
 4067 4027
 4070 6275
 4071 6177
 4072 4027
 4073 6575
 4074 6077
 4075 0000
 4076 4361
 4077 7154
 4100 6260
 4101 6040
 4102 0201
 4103 2504
 4104 4055
 4105 4002
 4106 6175
 4107 6177
 4110 4002
 4111 6275
 4112 6177
 4113 4002
 4114 6375
 4115 6177
 4116 4027
 4117 6275
 4120 6077
 4121 4027
 4122 6575
 4123 6177
 4124 0000
 4125 4365
 4126 6656
 4127 7040
 4130 0201
 4131 2504
 4132 4055

MESS6H, TEXT "#9600 BAUD - B1=1? B2=1? B3=1? W2=1? W5=0?"

MESS6I, TEXT "#19,200 BAUD - B1=1? B2=1? B3=1? W2=0? W5=1?"

MESS6J, TEXT "#56,8 BAUD - B1=0? B2=0? B3=0? W2=1? W5=0?"

4133 4002
4134 6175
4135 6077
4136 4002
4137 6275
4140 6077
4141 4002
4142 6375
4143 6077
4144 4027
4145 6275
4146 6177
4147 4027
4150 6575
4151 6077
4152 0000
4153 4366 MESS6K, TEXT "#66,7 BAUD = B1=0? B2=0? B3=0? W2=1? W5=0?"
4154 6656
4155 6740
4156 0201
4157 2504
4160 4055
4161 4002
4162 6175
4163 6077
4164 4002
4165 6275
4166 6077
4167 4002
4170 6375
4171 6077
4172 4027
4173 6275
4174 6177
4175 4027
4176 6575
4177 6077
4200 0000
4201 4361 MESS6L, TEXT "#1050 BAUD = B1=1? B2=0? B3=0? W2=1? W5=0?"
4202 6065
4203 6040
4204 0201
4205 2504
4206 4055
4207 4002
4210 6175
4211 6177
4212 4002
4213 6275
4214 6077
4215 4002
4216 6375
4217 6077
4220 4027
4221 6275

4222 6177
4223 4027
4224 6575
4225 6077
4226 0000
4227 4324 MESS7, TEXT "#TWO STOP BITS(Y OR N)? "
4230 2717
4231 4023
4232 2417
4233 2040
4234 0211
4235 2423
4236 5031
4237 4017
4240 2240
4241 1651
4242 7740
4243 4000
4244 4040 MESS7A, TEXT " SB="

4245 2302
4246 7500
4247 7743 MESS10, TEXT "?#DATA BITS/CHARACTER? 0=5 1=6 2=7 3=8 "
4250 0401
4251 2401
4252 4002
4253 1124
4254 2357
4255 0310
4256 0122
4257 0103
4260 2405
4261 2277
4262 4060
4263 7565
4264 4061
4265 7566
4266 4062
4267 7567
4270 4063
4271 7570
4272 4040
4273 0000

4274 4365 MESS10A, TEXT "#5 DATA BITS = NB1=1? NB2=1?#"
4275 4004
4276 0124
4277 0140
4300 0211
4301 2423
4302 4055
4303 4016
4304 0261
4305 7561
4306 7740
4307 1602

4310	6275		
4311	6177		
4312	4300		
4313	4366	MES10B, TEXT	"#6 DATA BITS = NB1=07 NB2=17#"
4314	4004		
4315	0124		
4316	0140		
4317	0211		
4320	2423		
4321	4095		
4322	4016		
4323	0261		
4324	7560		
4325	7740		
4326	1602		
4327	6275		
4330	6177		
4331	4300		
4332	4367	MES10C, TEXT	"#7 DATA BITS = NB1=17 NB2=07#"
4333	4004		
4334	0124		
4335	0140		
4336	0211		
4337	2423		
4340	4095		
4341	4016		
4342	0261		
4343	7561		
4344	7740		
4345	1602		
4346	6275		
4347	6077		
4350	4300	MES10D, TEXT	"#8 DATA BITS = NB1=07 NB2=07#"
4351	4370		
4352	4004		
4353	0124		
4354	0140		
4355	0211		
4356	2423		
4357	4095		
4360	4016		
4361	0261		
4362	7560		
4363	7740		
4364	1602		
4365	6275		
4366	6077		
4367	4300		
4370	0426	RECPT, KCF0	
4371	0471	KCF1	
4372	0601	KCF2	
4373	0691	KCF3	
4374	1262	KCF4	
4375	0404	KSF0	
4376	0444	KSF1	

4377	0462	KSF2	
4400	0561	KSF3	
4401	0637	KSF4	
4402	0660	KSF5	
4403	1015	KSF6	
4404	1030	KSF7	
4405	3113	KSF8	
4406	1100	KSF9	
4407	1124	KSF10	
4410	1206	KSF11	
4411	1217	KSF12	
4412	1235	KSF14	
4413	1256	KSF16	
4414	1263	KSF17	
4415	1622	KSF18	
4416	1722	FDTL0P	
4417	1740	KSF19	
4420	2072	KSF21	
4421	2137	KSF22	
4422	2155	KSF23	
4423	1935	KSF24	
4424	2231	KSF29	
4425	2256	KSF26	
4426	0667	KCC0	
4427	1007	KCC1	
4430	1123	KCC2	
4431	1204	KCC3	
4432	1255	KCC4	
4433	2152	KCC5	
4434	2165	KCC6	
4435	0705	KRS0	
4436	1075	KRS1	
4437	2233	KRS2	
4440	2241	KRS3	
4441	1531	KSE0	
4442	2216	KSE1	
4443	2240	KSE2	
4444	2247	KSE3	
4445	0473	KIE0	
4446	0512	KIE1	
4447	0547	KIE2	
4450	0602	KIE3	
4451	0636	KIE4	
4452	0675	KIE5	
4453	1004	KIE6	
4454	1044	KIE7	
4455	1054	KIE8	
4456	1113	KIE9	
4457	1137	KIE10	
4460	0714	KRS0	
4461	1232	KRS1	
4462	1607	KRS2	
4463	1640	KRS3	
4464	1703	KRS4	
4465	1710	KRS5	

4466	1725	KR36
4467	1743	KR37
4470	2044	KR38
4471	2051	KR39
4472	2106	KR310
4473	2502	INTON
4474	1545	KR311
4475	2207	KR312
4476	2253	KR313
4477	2263	KR314
4500	0000	0000

4501	0431	XMTIOT, TFL0
4502	0453	TFL1
4503	0476	TFL2
4504	0542	TFL3
4505	0603	TFL4
4506	0620	TFL5
4507	0725	TFL6
4510	0407	TSF0
4511	0434	TSF1
4512	0441	TSF2
4513	0454	TSF3
4514	0457	TSF4
4515	0477	TSF5
4516	0515	TSF6
4517	0527	TSF7
4520	0543	TSF8
4521	0604	TSF9
4522	0612	TSF10
4523	0621	TSF11
4524	0633	TSF12
4525	0743	TSF13
4526	1020	TSF14
4527	3075	TSF15
4530	1065	TSF16
4531	1070	TSF17
4532	1211	TSF18
4533	1225	TSF20
4534	1243	TSF22
4535	1625	TSF23
4536	1715	TSF24
4537	1734	TSF25
4540	1752	TSF26
4541	2067	TSF28
4542	2142	TSF29
4543	2477	TSF30
4544	2511	TSF31
4545	1540	TSF32
4546	0436	TCF0
4547	0526	TCF1
4550	0560	TCF2
4551	0632	TCF3
4552	0734	TCF4

4553	1012	TCF5
4554	1067	TCF6
4555	1205	TCF7
4556	1242	TCF8
4557	1611	TCF9
4560	1704	TCF10
4561	1711	TCF11
4562	2010	TCF12
4563	2053	TCF13
4564	1542	TCF14
4565	2210	TCF15
4566	2230	TCF16
4567	1023	TF00
4570	1214	TF01
4571	0412	SP10
4572	0501	SP11
4573	0517	SP12
4574	0532	SP13
4575	0545	SP14
4576	0550	SP15
4577	0606	SP16
4600	0615	SP17
4601	0623	SP18
4602	0752	SP19
4603	1033	SP110
4604	1045	SP111
4605	1055	SP112
4606	1102	SP113
4607	1114	SP114
4610	1127	SP115
4611	1222	TLS0
4612	1621	XMIT
4613	1714	TLS1
4614	1721	TLS2
4615	1737	TLS3
4616	1755	TLS4
4617	2005	TLS5
4620	2062	TLS6
4621	2476	TLS7
4622	2504	TLS8
4623	2221	ILS9
4624	1534	SLWTL5
4625	0000	0000

5000 *BASEA /MUST BE THE FIRST ADDRESS OF A PAGE

5000 0000 FILLER, 0 /SET TO NUMBER OF FILLERS REQUIRED

/INPUT ONE OCTAL NUMBER TO AC 9 THRU 11

/GOOD RETURN IS JMS+2

5001 0000 ONEOCK, 0 /CALL BY "ONEOCT"

```

5002 4570 LISN
5003 0001 1
5004 5007 ,+3
5005 0000 0
5006 5010 ,+2
5007 2201 ISZ ONEOCK
5010 5601 JMP I ONEOCK

/INPUT TWO OCTAL NUMBERS TO AC 6 THRU 11
/GOOD RETURN IS JMS+2

5011 0000 THOOCK, 0 /CALL BY "THOOCK"
5012 4201 JMS ONEOCK
5013 5611 JMP I THOOCK
5014 7104 CLL RAL
5015 7006 RTL
5016 3224 DCA XPRNT2
5017 4201 JMS ONEOCK
5020 5611 JMP I THOOCK
5021 1224 TAD XPRNT2
5022 2211 ISZ THOOCK
5023 5611 JMP I THOOCK

/PRINT THE TWO OCTAL NUMBERS IN THE AC 6 THRU 11

5024 0000 XPRNT2, 0 /CALL BY "PRNT2"
5025 3211 DCA THOOCK
5026 1211 TAD THOOCK
5027 7012 RTR
5030 7010 RAR
5031 4563 PRNT1
5032 1211 TAD THOOCK
5033 4563 PRNT1
5034 5624 JMP I XPRNT2

/TYPE THE ASCII CHARACTER IN THE AC

5035 0000 XTYPE, 0 /CALL BY "TYPE"
5036 6046 TLS
5037 7200 CLA
5040 6041 TSP
5041 5240 JMP ,=1
5042 6042 TCF
5043 5635 JMP I XTYPE

/TYPE A CR AND LF WITH NUMBER OF FILLERS
/AS DETERMINED BY LOCATION "FILLER"

5044 0000 XCRLF, 0 /CALL BY "CRLF"
5045 7200 CLA
5046 1260 TAD K215
5047 4557 TYPE
5050 1200 TAD FILLER
    
```

```

5051 7040 CHA
5052 3262 DCA XORS
5053 1201 TAD K212
5054 4557 TYPE
5055 2262 ISZ XORS
5056 5254 JMP ,=2
5057 5644 JMP I XCRLF
5060 0215 K215, 0215
5061 0212 K212, 0212

/PERFORM THE XOR OF THE AC AND THE CALL+1
/RETURN TO CALL+2

5062 0000 XORS, 0 /CALL BY "XOR"
5063 3274 DCA YESRNX
5064 1274 TAD YESRNX
5065 0662 AND I XORS /IN BRIEF, TAD THE TWO
5066 7041 CIA /NUMBERS THEN SUBTRACT
5067 7104 CLL RAL /THE CARRIES TO PRODUCE
5070 1274 TAD YESRNX /A HALF ADD (XOR)
5071 1662 TAD I XORS
5072 2262 ISZ XORS
5073 5662 JMP I XORS

/LOOK FOR "Y" OR "N" INPUT

5074 0000 YESRNX, 0 /CALL BY "YESRNX"
5075 4570 LISN /INPUT ONE CHARACTER IF AC=0
5076 7447 =NY
5077 5104 ,+3 /RETURN TO CALL+3 IF "Y"
5100 7462 =NY
5101 5105 ,+4 /RETURN TO CALL+2 IF "N"
5102 0000 0
5103 5106 ,+3 /RETURN TO CALL+1 IF NEITHER
5104 2274 ISZ YESRNX
5105 2274 ISZ YESRNX
5106 5674 JMP I YESRNX

/PRINT 2 SPACES

5107 0000 SPACX2, 0 /CALL BY "SPACE2"
5110 4567 YESAGE
5111 5113 ,+2
5112 5707 JMP I SPACX2
5113 4040 4040
5114 0010 K10, 0010 /USED BY LISN

/COMPARE INPUT TO LIST FOLLOWING CALL
/INPUT ONE CHARACTER IF AC=0
/USE LAST INPUT IF AC NON ZERO

5115 0000 XLISN, 0 /CALL BY "LISN"
5116 7640 SZA CLA
5117 5341 JMP LISN1 /USE LAST INPUT SINCE AC NOT ZERO
    
```

```

5120 6031 KSF
5121 5320 JMP ,=1
5122 6036 KRB
5123 0373 AND K177
5124 1374 TAD K200
5125 3707 DCA I LISNT1
5126 1707 TAD I LISNT1
5127 1375 TAD M212
5130 7450 SNA /IS IT A LF?
5131 5335 JMP ,+4 /YES
5132 1376 TAD M3
5133 7640 SZA CLA /IS IT A CR?
5134 5337 JMP ,+3 /NO
5135 4556 CRLF
5136 5341 JMP LISN1
5137 1707 TAD I LISNT1
5140 4557 TYPE /PRINT THE CHARACTER
5141 1715 LISN1, TAD I XLISN /GET COMPARE VALUE
5142 2315 ISE XLISN
5143 7450 SNA /EXIT?
5144 5352 JMP LISN3 /YES
5145 7500 SNA
5146 5362 JMP LISNUM /LOOK FOR OCTAL NUMBER
5147 1707 TAD I LISNT1 /COMPARE
5150 7640 SZA CLA /EQUAL?
5151 5357 JMP LISN2 /NO
5152 3563 LISN3, DCA I EXPRNT1
5153 1715 TAD I XLISN
5154 3315 DCA XLISN
5155 1563 TAD I EXPRNT1
5156 5715 JMP I XLISN /AC IS ZERO UNLESS OCTAL NUMBER
5157 7200 LISN2, CLA
5160 2315 ISE XLISN
5161 5341 JMP LISN1
5162 7200 LISNUM, CLA /LOOK FOR OCTAL NUMBER
5163 1707 TAD I LISNT1
5164 1370 TAD M270
5165 7500 SNA /IS IT LESS THAN 8?
5166 5357 JMP LISN2 /NO, SO NOT AN OCTAL NUMBER
5167 1314 TAD K10
5170 7510 M270, SPA /IS IT GREATER THAN ZERO?
5171 5357 JMP LISN2 /NO, SO NOT A NUMBER
5172 5352 JMP LISN3
5173 0177 K177, 0177
5174 0200 K200, 0200
5175 7566 M212, 7566
5176 7775 M3, 7775
LISNT1=ISPACX2
PAGE

```

```

/PRINT PACKED ASCII TEXT TERMINATED BY
/SIX-BIT 00

```

```

5200 0000 MESAGX, 0 /CALL BY "MESAGE"
5201 7200 CLA
5202 1600 TAD I MESAGX
5203 3264 DCA FOROCK
5204 2200 ISE MESAGX /SET UP RETURN
5205 1664 TAD I FOROCK
5206 7012 RTN
5207 7012 RTN
5210 7012 RTN
5211 4216 JMS MESAGF
5212 1664 TAD I FOROCK
5213 4216 JMS MESAGF
5214 2264 ISE FOROCK
5215 5205 JMP ,=10
5216 0000 MESAGF, 0
5217 0235 AND K77
5220 7450 SNA /TERMINATOR (00)?
5221 5600 JMP I MESAGX /YES
5222 1236 TAD M43
5223 7450 SNA /CRLF?
5224 5233 JMP ,+7 /YES
5225 1237 TAD K3
5226 7510 SPA /200 OR 300
5227 1240 TAD K100 /300
5230 1241 TAD K240 /200
5231 4557 TYPE
5232 5616 JMP I MESAGF
5233 4556 CRLF
5234 5616 JMP I MESAGF
5235 0077 K77, 0077
5236 7735 M43, 7735
5237 0003 K3, 0003
5240 0100 K100, 0100
5241 0240 K240, 0240

/MODIFY DEVICE CODE FOR A LIST OF IOT ADDRESSES
XH10T, 0 /CALL BY "HIOT"
5242 0000
5243 0262 AND K770
5244 3200 DCA MESAGX
5245 1642 TAD I XH10T
5246 2242 ISE XH10T
5247 3264 DCA FOROCK
5250 1664 TAD I FOROCK /GET NEXT ADDRESS
5251 7450 SNA /END OF LIST? (ZERO)
5252 5642 JMP I XH10T /YES
5253 3300 DCA XPRNT4
5254 1700 TAD I XPRNT4 /GET IOT
5255 0263 AND K7007 /REMOVE OLD DEVGIE CODE
5256 1200 TAD MESAGX /ADD NEW DEVICE CODE
5257 3700 DCA I XPRNT4 /PUT BACK IOT
5260 2264 ISE FOROCK
5261 5250 JMP ,=11

```

```

5262 0770 K770, 0770
5263 7007 K7007, 7007

/INPUT 4 OCTAL NUMBERS TO AC
/GOOD RETURN IS CALL+2

5264 0000 FOROCK, 0 /CALL BY "FOROCK"
5265 4565 TWOCT
5266 5664 JMP I FOROCK
5267 7106 CLL RTL
5270 7006 RT_
5271 7006 RTL
5272 3300 DCA XPRNT4
5273 4565 TWOCT
5274 5664 JMP I FOROCK
5275 1300 TAJ XPRNT4
5276 2244 ISE FOROCK
5277 5664 JMP I FOROCK

/PRINT FOUR OCTAL NUMBERS IN AC 11 THRU 0 FOLLOWED
/BY TWO SPACES

5300 0000 XPRNT4, 0 /CALL BY "PRNT4"
5301 3244 DCA FOROCK
5302 1244 TAJ FOROCK
5303 7012 RTR
5304 7012 RTR
5305 7012 RTR
5306 4562 PRNT2
5307 1264 TAJ FOROCK
5310 4562 PRNT2
5311 4560 SPACE2
5312 5700 JMP I XPRNT4

/PRINT THE OCTAL NUMBER IN AC 9 THRU 11
/CALL BY "PRNT1"
5313 0000 XPRNT1, 0
5314 0320 AND K7
5315 1321 TAJ K260
5316 4557 TYPE
5317 5713 JMP I XPRNT1
5320 0007 K7,
5321 0260 K260,

/SWAP BITES IN THE AC, PRESERVE THE LINK
/
5322 0000 XBSH, 0 /CALL BY "BSWAP"
5323 3337 DCA XRAND
5324 7012 RTR
5325 7012 RTR
5326 7012 RTR
5327 1337 TAJ XRAND
5330 0336 AND K7700
5331 1337 TAJ XRAND
5332 7006 RT_
5333 7006 RT_
    
```

```

5334 7006 RT_
5335 5722 JMP I XBSH
5336 7700 K7700, 7700

/GENERATE RANDOM NUMBER
/EXIT WITH NUMBER IN AC

5337 0000 XRAND, 0 /CALL BY "RANDOM"
5340 7301 CLA CLL IAC
5341 1371 TAJ RAN1
5342 1372 TAJ RAN2
5343 7106 CLL RTL
5344 3371 DCA RAN1
5345 1372 TAJ RAN2
5346 7012 RTR
5347 1371 TAJ RAN1
5350 3372 DCA RAN2
5351 1372 TAJ RAN2
5352 5737 JMP I XRAND

/SAVE RANDOM
/GENERATOR PRIMES

5353 0000 XSAVGN, 0 /CALL BY "SAVGEN"
5354 7200 CLA
5355 1371 TAJ RAN1
5356 3373 DCA SAV1
5357 1372 TAJ RAN2
5360 3374 DCA SAV2
5361 5753 JMP I XSAVGN

/RESTORE RANDOM
/GENERATOR PRIMES

5362 0000 XRESGN, 0 /CALL BY "RESGEN"
5363 7200 CLA
5364 1373 TAJ SAV1
5365 3371 DCA RAN1
5366 1374 TAJ SAV2
5367 3372 DCA RAN2
5370 5762 JMP I XRESGN
5371 1234 RAN1,
5372 5670 RAN2,
5373 0000 SAV1,
5374 0000 SAV2,
    
```

```

S
0144 7700
0145 4060
0146 7400
0147 5322
0150 5362
0151 5393
0152 5337
    
```

0153 5074
0154 5042
0155 5242
0156 5044
0157 5035
0160 5107
0161 5300
0162 5024
0163 5313
0164 5244
0165 5011
0166 5001
0167 5200
0170 5115
0171 3107
0172 3071
0173 0305
0174 2400
0175 0277
0176 0267
0177 3020

0000 11110000 00000000 11111111 11111111 11000000 00000000 00000000 00000000
0100 00000000 00000000 00000000 00000000 00001111 11111111 11111111 11111111

0200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0300 11111111 11111111 00000000 00000000 00000000 00000000 00000011 11111111

0400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 00000001

0600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0700 11111111 11111111 11111111 11111111 11111111 11111111 11110000 00011111

1000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1100 11111111 11111111 11111111 11111111 11110000 00000000 00000000 00000001

1200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11000000
1300 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000001

1400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 10011111

1600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 00000111

2000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11001111

2200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2300 11111111 11111111 11111111 10000000 00000000 00000000 00000000 00001111

2400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2500 11111111 11111111 11111111 11111111 11111111 11111110 00000000 00000111

2600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

3000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3100 11111111 11111111 11111000 00000000 00000000 00000000 00000000 00000001

3200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3300 11111111 11111111 11111111 11111111 11111111 11111111 11111000 11111111

3400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

3600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111


```

4000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

4200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

4400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4500 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

4600 11111111 11111111 11111100 00000000 00000000 00000000 00000000 00000000
4700 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

5000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
5100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

5200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
5300 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111000

5400
5500

5600
5702

6000
6100

6200
6302

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700
    
```

```

ACNSKP 0645 FOROCK 5264 KR00 0714 LISN1 5141
B17200 2740 FOROCT 4564 KR01 1232 LISN2 5197
BASEA 5000 HLT 7402 KR02 2106 LISN3 5192
BAUDNO 0030 MLTLOP 2400 KR03 1545 LISNT1 0100
BAUDTB 2523 MLTLP1 2414 KR04 2207 LISNUM 5162
BAUDTH 2453 MLTLP2 2433 KR05 2253 LOOP 4575
SDPNTR 2522 INTON 2502 KR06 2263 LOOPPC 0035
BGNINT 0200 INTRET 0265 KR07 1607 M10 0276
BITNO 0026 INTXMT 0470 KR08 1640 M212 5175
BR1050 3004 K10 5114 KR09 1703 M270 5170
BR110 2600 K100 5240 KR10 1710 M3 5176
BR1200 2660 K177 5173 KR11 1725 M43 5236
BR150 2614 K200 5174 KR12 1743 MES10A 4274
BR2400 2674 K212 5061 KR13 2014 MES10B 4313
BR300 2630 K215 5060 KR14 2051 MES10C 4332
BR4000 2710 K240 5241 KR15 6034 MES10D 4351
BR500 2754 K250 5321 KR16 0705 MES10E 4567
BR600 2644 K3 5237 KR17 1075 MES10F 5216
BR667 2770 K37 0020 KR18 2233 MES10G 5200
BR9600 2724 K377 0023 KR19 2241 MESS1 3400
BSWAP 4547 K7 5320 KSF 6035 MESS10 4247
C177 0022 K7007 5263 KSE0 1531 MESS2 3410
C7 0021 K7 5235 KSE1 2216 MESS3 3421
CAF 6007 K770 5262 KSE2 2240 MESS3A 3433
CAFINT 0600 K7700 5336 KSE3 2247 MESS4 3436
CAFMT 0452 KCC 6032 KSF 6031 MESS4A 3477
CHARLG 2000 KCC0 0667 KSF0 0404 MESS5 3503
CHGWES 3320 KCC1 1007 KSF1 0444 MESS5A 3523
CHKFIL 3070 KCC2 1123 KSF10 1124 MESS6 3527
CLBRDR 0400 KCC3 1204 KSF11 1206 MESS6A 3622
CHES6 3567 KCC4 1255 KSF12 1217 MESS6B 3647
CNT1 0037 KCC5 2192 KSF14 1235 MESS6C 3674
CNT2 0040 KCC6 2165 KSF16 1256 MESS6D 3721
CNTREC 2132 KCF 6030 KSF17 1263 MESS6E 3746
CON100 3044 KCF0 0426 KSF18 1622 MESS6F 3774
CRLF 4556 KCF1 0471 KSF19 1740 MESS6G 4022
DATBIT 0027 KCF2 0601 KSF2 0462 MESS6H 4050
DELAY 4577 KCF3 0651 KSF21 2072 MESS6I 4076
DELAYN 3043 KCF4 1262 KSF22 2137 MESS6J 4125
DELLOP 3024 KIE 6035 KSF23 2155 MESS6K 4153
DEVOD 0024 KIE0 0473 KSF24 1535 MESS6L 4201
EHLTLP 4574 KIE1 0512 KSF25 2231 MESS7 4227
END 1701 KIE10 1137 KSF26 2256 MESS7A 4244
ERRFLG 0033 KIE2 0947 KSF3 0561 MESTAB 2537
ERRLOP 1707 KIE3 0602 KSF4 0637 MESTB1 2553
FDATAT 1600 KIE4 0636 KSF5 0660 M10T 4555
FDTLOP 1722 KIE5 0675 KSF6 1015 NODELAY 0036
FILGHK 3046 KIE6 1004 KSF7 1030 NOINTR 0201
FILERT 2042 KIE7 1044 KSF8 3113 NOT0E 0305
FILEXT 2170 KIE8 1054 KSF9 1100 ONEOCK 5001
FILLER 5000 KIE9 1113 KSPSKP 4571 ONEOCT 4566
FILL0P 2064 KR3 6036 LISN 4570 PRNT1 4563
    
```

PRNT2	4562	STLPPC	4576	TSP17	1070	XTYPE	5035
PRNT4	4561	SW10NE	4573	TSP18	1211	YESRNO	4553
RAN1	5371	TCF	0042	TSP2	0441	YESRNX	5074
RAN2	5372	TCF0	0436	TSP20	1225		
RANDOM	4592	TCF1	0526	TSP22	1243		
RECDAT	0034	TCF10	1704	TSP23	1625		
RECEVE	1637	TCF11	1711	TSP24	1715		
RECPNT	4370	TCF12	2010	TSP25	1734		
RESGEN	4590	TCF13	2053	TSP26	1752		
RETINT	2506	TCF14	1542	TSP28	2067		
SAV1	5373	TCF15	2210	TSP29	2142		
SAV2	5374	TCF16	2230	TSP3	0454		
SAVAC	2432	TCF2	0560	TSP30	2477		
SAVBTS	0025	TCF3	0632	TSP31	2511		
SAVGEN	4591	TCF4	0734	TSP32	1540		
SAVIT	3364	TCF5	1012	TSP4	0457		
SCXN1T	0425	TCF6	1067	TSP5	0477		
SDTST1	1400	TCF7	1205	TSP6	0515		
SDTST2	1406	TCF8	1242	TSP7	0527		
SDTST3	1416	TCF9	1611	TSP8	0543		
SDTST4	1430	TFL	0040	TSP9	0604		
SDTST5	1445	TFL0	0431	TSPSKP	4572		
SDTST6	1475	TFL1	0453	TSYCNT	0041		
SDTST7	1511	TFL2	0476	TWOOCK	5011		
SETUP	0215	TFL3	0542	TWOOCK	4505		
SLWDAT	1527	TFL4	0603	TYINTH	3200		
SLWREC	1544	TFL5	0620	TYPE	4557		
SLWFLS	1534	TFL6	0725	UPDATE	1664		
SPACE2	4560	TLS	0046	US100	3045		
SPACX2	5107	TLS0	1222	WATKSF	3107		
SPI	6045	TLS1	1714	WATTSF	3071		
SPI0	0412	TLS2	1721	XBSW	5322		
SPI1	0501	TLS3	1737	XGRLF	5044		
SPI10	1033	TLS4	1755	XDELAY	3020		
SPI11	1045	TLS5	2005	XLISN	5115		
SPI12	1055	TLS6	2062	XMIOT	5242		
SPI13	1102	TLS7	2476	XMIT	1621		
SPI14	1114	TLS8	2504	XMTDAT	0031		
SPI15	1127	TLS9	2221	XMTDT1	0032		
SPI2	0547	TPC	0044	XMTIOT	4501		
SPI3	0532	TPC0	1023	XMYREC	1200		
SPI4	0545	TPC1	1214	XOR	4554		
SPI5	0550	TSF	0041	XORS	5062		
SPI6	0606	TSF0	0407	XPCRET	0267		
SPI7	0615	TSF1	0434	XPRNT1	5313		
SPI8	0623	TSF10	0612	XPRNT2	5024		
SPI9	0752	TSF11	0621	XPRNT4	5300		
SR4HLT	2323	TSF12	0633	XRAND	5337		
START	0246	TSF13	0743	XRESOIN	5362		
STENAB	2200	TSF14	1020	XSAVGN	5353		
STERR	2274	TSF15	3075	XSR1LP	1763		
STFLGS	1000	TSF16	1065	XSR2	0277		

ERRORS DETECTED: 0
 LINKS GENERATED: 40
 RUN-TIME: 12 SECONDS
 3K CORE USED



