

Micro Fiche Scan

Name of device(s) tested:

DHU11

Test description:

DHU-11 DEC/X11 MOD

MAINDEC Number or Package Identifier (after SEP 1977):

CXDHUA0

Fiche Document Part Number:

AH-T803A-MC

Fiche preparation date unknown, using copyright year:

1984

Image resolution:

8-bit gray levels, max. quality for archiving

COPYRIGHT (C) 1983-84 by d|i|g|i|t|a|l

1 .REM_

.SBTTL DOCUMENTATION

IDENTIFICATION

PRODUCT CODE: AC-T802A-MC

PRODUCT NAME: CXDHUAO DHU-11 DEC/X11 MODULE

PRODUCT DATE: 15 DECEMBER 1983

MAINTAINER: ENE - DIAGNOSTICS, READING

AUTHOR: A.HYATT

VERSION: A.0

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

COPYRIGHT (C) 1983,1984 DIGITAL EQUIPMENT CORPORATION

C1

REVISION HISTORY

REV.	DATE	AUTHOR	COMMENTS
A.0	15-DEC-83	A.HYATT	FIRST RELEASE

1. ABSTRACT:

THIS DOCUMENT DESCRIBES THE DHU11 DEC/X11 EXERCISER MODULE. THE PROGRAM EXERCISES UP TO FOUR (CONSECUTIVELY ADDRESSED) DHU11 ASYNCHRONOUS INTERFACES. MAINTENANCE MODE IS USED TO TRANSMIT AND RECEIVE A BINARY COUNT PATTERN TRANSMITTED AND RECEIVED IN 108 CHARACTER BURSTS. ALL DEVICES SELECTED FOR TESTING ARE ACTIVATED AND RUN CONCURRENTLY. ALL SIXTEEN LINES ARE RUN ON EACH SELECTED DEVICE.

THIS PROGRAM WILL RUN ONLY ON A UNIBUS PROCESSOR. THIS PROGRAM WILL NOT SUPPORT ANY DEVICE OTHER THAN THE DHU11.

THE DHU11 IS AN ASYNCHRONOUS MULTIPLEXER THAT PROVIDES AN INTERFACE BETWEEN SIXTEEN ASYNCHRONOUS SERIAL DATA COMMUNICATIONS CHANNELS AND ANY UNIBUS PROCESSOR.

THE DHU11 PROVIDES RS232-A AND RS423-A INTERFACES WITH ENOUGH MODEM CONTROL TO PERMIT FULL DUPLEX POINT TO POINT OPERATION OR DIAL UP (AUTO ANSWER) OPERATION OVER THE PUBLIC SWITCHED TELEPHONE NETWORK.

FOR EACH LINE THE CHARACTERISTICS ARE SET UNDER PROGRAM CONTROL. THE ONLY INFORMATION THAT IS SET VIA SWITCHES ON THE CARD BEING THE UNIBUS ADDRESS AND THE VECTOR ADDRESS.

2. REQUIREMENTS:

HARDWARE: 0 A PDP SYSTEM WITH UNIBUS

0 32K BYTES OF MEMORY (TOTAL) ON THE SYSTEM.

0 1-4 DHU11 INTERFACES. NO WRAPAROUND OR MAINTENANCE CABLE IS NEEDED IF RUNNING IN LOCAL LOOPBACK MODE.

STORAGE: THIS MODULE (DHU11) REQUIRES:

1. DECIMAL WORDS: 1122
2. OCTAL WORDS: 2142
3. OCTAL BYTES: 4304

SOFTWARE: LATEST REVISION OF THE DEC/X11 MONITOR & LINKER.

3. PASS DEFINITION:

ONE PASS OF THE DHU11 MODULE CONSISTS OF A NUMBER (DEFAULT OF 8) OF INTERACTIONS.

AN ITERATION CONSISTS OF FOUR BASIC TESTS ON EACH DHU11, INCLUDING THE TRANSMITTING AND RECEIVING OF 108 CHARACTERS ON EACH LINE.

4. EXECUTION TIME:

THE EXECUTION TIME (TIME TO MAKE ONE PASS) DEPENDS ON THE NUMBER OF DHU11'S, THE NUMBER OF LINES SELECTED, THE BAUD RATE AND THE OTHER MODULES CONFIGURED IN. THE AVERAGE TIME FOR A MODULE RUNNING STAND ALONE ON A PDP 11/24 WITH BAUD RATES ETC. SET TO DEFAULT IS ONE MINUTE.

5. CONFIGURATION PARAMETERS:

MEANING OF PARAMETERS:

DVA: ADDRESS OF FIRST DHU11 CSR REG.
VCT: VECTOR ADDRESS OF FIRST DHU11
BR1: PRIORITY OF DHU11'S
BR2: SECOND PRIORITY (NOT USED)
DVC: NO. OF DHU11'S IF GREATER THAN 1
SR1: FIRST SOFTWARE REGISTER

DEFAULT PARAMETERS:

DVA: 1, VCT: 1, BR1: 5, BR2: 5, DVC: 1, SR1: 0

REQUIRED PARAMETERS:

AT CONFIGURATION TIME THE USER MUST SPECIFY:

DVA: ADDRESS OF FIRST DHU11 CSR REG.
VCT: VECTOR ADDRESS OF FIRST DHU11
DVC: NO. OF DHU11'S IF GREATER THAN 1

MODULE LOCATION DVID1 (HOLDS DVC) MAY BE MODIFIED (MOD DHUA 14) TO EXERCISE ANY COMBINATION OF DHU11'S. (SEE ITEM #8, "OPERATOR OPTIONS").

6. DEVICE OPTION SETUP:

NONE REQUIRED

7. MODULE OPERATION:

THE DHU11 DEC/X11 MODULE IS ASSEMBLED AND LINKED ACCORDING TO STANDARDS OUTLINED IN THE DEC/X11 MODULE PROGRAMMER'S GUIDE.

THE MAIN ALGORITHM:

THE TESTING IS PERFORMED WITH INTERNAL LOOPBACK SELECTED, VIA MAINTENANCE MODE. THE CHARACTERS 224 - 377 (OCTAL) ARE TRANSMITTED ON 16 LINES SIMULTANEOUSLY VIA DIRECT MEMORY ACCESS. THE RECEIVER, NOT OPERATING ON A DMA BASIS, WILL RECEIVE CHARACTERS ON A PER/CHARACTER BASIS INTO THE RECEIVE FIFO. IF SUCCESSFUL, 108 (DECIMAL) CHARACTERS PER LINE SHOULD BE RECEIVED. EACH RECEIVED LINE WILL BE CHECKED FOR PROPERLY RECEIVED DATA BY COMPARING EACH RECEIVED CHARACTER TO ITS EXPECTED VALUE.

- START: 0 SETUP NUMBER OF DHU11'S TO BE TESTED.
- RESTR1: 0 SETUP DEVICE ADDRESSES & VECTORS.
- DHUT01: 0 RESET DHU11 AND TEST THE RESET BIT CLEARS.
- DHUT02: 0 SEND ONE CHARACTER AROUND THE LOOP AND TEST THAT THE BOARD INTERRUPTS. DONE FOR EACH LINE ON ALL BOARDS.
- DHUT03: 0 RUN INTERNAL SELF TEST DIAGNOSTICS ON ALL DHU11'S SELECTED AND TEST FOR PASS OR FAIL.
- DHUT04: 0 GET LINES TO BE TESTED ON THIS DHU11.
 0 SET-UP WRITE COUNT PATTERN INTO XMIT BUFFER.
 0 SET-UP LINE PARAMETERS (9600 BAUD, 8 BIT/CHAR., ENABLE RECEIVER).

0 START ALL LINES/DHU11 SELECTED.

TXSERV:

0 CHECK FOR VALID INTERRUPT AND REPORT
ERROR AS REQUIRED.

RXSERV:

0 RECEIVES EACH CHARACTER OF EACH LINE UPON GETTING
FIFO INTERRUPTS.
0 VERIFIES CORRECT DATA REPORT ERROR AS REQUIRED.
0 CHECK FOR END OF DATA AND BOARD FINISHED.
0 WHEN ALL DONE DECLARE END OF ITERATION.

8. OPERATOR OPTIONS:

NOTE

SR1 , 2 AND 3 APPLY TO ALL DHU11'S SELECTED

1). TESTING MULTIPLE DHU11'S

MODULE LOCATION DVID1 (MOD DHU 14) MAY BE MODIFIED TO
EXERCISE ANY COMBINATION OF FOUR DHU11'S:

DEVICE TO TEST -----	DVID1 = -----
1	1
2	2
3	4
4	10
1,2	3
1,3	5
1,4	11
2,3	6
2,4	12
3,4	14
1,2,3	7
1,2,4	13
1,3,4	15
2,3,4	16
1,2,3,4	17

2). SELECTABLE BAUD RATE:

THE BAUD RATE MAY BE SELECTED BASED ON THE CODE BELOW. THE SELECTION

PROCEDURE IS VERY SIMPLE.

IF YOU WANT TO USE 9600 BITS/SEC. THEN SIMPLY LEAVE BOTH SR1 & SR2 "0". THIS IS THE DEFAULT VALUE.

IF YOU WISH TO USE OTHER BAUD RATES THEN FOLLOW THIS PROCEDURE:

- A. SELECT THE BAUD RATE YOU DESIRE FROM THE BAUD RATE TABLE. WE WILL USE 38.4K FOR EXAMPLE.

NOTE 1: THE TRANSMIT & RECEIVE BAUD RATES WILL BE THE SAME.

2: IF USING 38.4K BAUD RATE, A MAXIMUM OF 2 LINES MAY BE USED AT ONE TIME.

IF USING 19.2K BAUD RATE, A MAXIMUM OF 4 LINES MAY BE USED AT ONE TIME.

THE VARIABLE "LINESELECT" MUST BE CHANGED TO REFLECT THE MAXIMUM ALLOWABLE # OF LINES THAT MAY BE TESTED. (2 LINES @ 38.4, 4 LINES @ 19.2)

- B. PLACE THE BINARY CODE FOR THE PARTICULAR BAUD RATE INTO SR1. FOR OUR EXAMPLE 38.4K HAS A BINARY CODE VALUE OF 17.
- C. PLACE A "1" IN SR2. SR2 MUST ALWAYS CONTAIN A ONE WHENEVER A BAUD RATE (OTHER THAN THE DEFAULT) IS SELECTED.
- D. IN ORDER FOR THE PROGRAM TO RUN FOR 1 MINUTE, THE VALUE "ICONT" MUST BE MODIFIED FOR EACH SELECTED BAUD RATE. THE APPROPRIATE VALUE FOR "ICONT" MAY BE FOUND IN THE BAUD RATE TABLE. THE 1 MINUTE RUNTIME IS BASED ON RUNNING 1 DHU11. THE RUNTIME WILL BE PROPORTIONATLY LONGER FOR EACH ADDITIONAL DHU11 SELECTED FOR TESTING.

THE BAUD RATE MAY BE SELECTED AT TIME OF CONFIGURATION OR AT RUNTIME BY CHANGING THE CONTENTS OF SR1, SR2, & ICONT TO SELECT A PARTICULAR BAUD RATE.

LOCATION SR1 (MOD DHUA 16).
LOCATION SR2 (MOD DHUA 20).
LOCATION ICONT (MOD DHUA 36).

(NOTE: IF SR2 CONTAINS A "0", THE BAUD RATE SELECTED WILL BE 9600 NO MATTER WHAT IS IN SR1!!)

BAUD RATE TABLE

DATA RATE BINARY CODE ICONT -

(BITS/SEC)	(SR1 =)	
50	0	1
75	1	1
110	2	2
134.5	3	2
150	4	3
300	5	4
600	6	5
1200	7	6
1800	10	6
2000	11	7
2400	12	7
4800	13	7
7200	14	7
9600	15	7
19200	16	10
38400	17	10

- 3). MAY CHOOSE BETWEEN INTERNAL LOOPBACK OR NORMAL OPERATION (NORMAL OPERATION REQUIRES LOOPBACK CONNECTORS TO BE FITTED).
- A. INTERNAL LOOPBACK REQUIRES NO CABLES, ONLY THE MODULE IS TESTED.
 - B. NORMAL OPERATION ALLOWS THE OPERATOR TO TEST NOT ONLY THE MODULE, BUT THE CABLES TOO. A H325 LOOPBACK CONNECTOR IS REQUIRED FOR EACH LINE SET TO NORMAL, IF THIS IS PLACED ON THE DISTRIBUTION PANEL ONLY THE INTERNAL CABLES ARE TESTED. THE H325 CAN ALSO BE PLACED AT THE END OF THE BC05L CABLE (FROM THE DISTRIBUTION PANEL) AND THE BC05L CABLE IS THEN TESTED.

INTERNAL LOOPBACK:
 LEAVE SR3 "0".

NORMAL OPERATION:
 PLACE INTO SR3 THE BIT MASK OF THE LINE DESIRED FOR TESTING, (SEE BIT MASK VALUES BELOW). (THE SAME LINE NUMBER WILL BE TESTED ON ALL DEVICES SELECTED).

LOCATION SR3: (MOD DHUA 22).

BIT MASK VALUES FOR TESTING A LINE IN NORMAL OPERATION

LINE # DESIRED	SR3
-----	---
0	1

1	2
2	4
3	10
4	20
5	40
6	100
7	200
8	400
9	1000
10	2000
11	4000
12	10000
13	20000
14	40000
15	100000

4). INDIVIDUAL & MULTIPLE LINE TESTING

MODULE VARIABLE "LINESELECT" MAY BE MODIFIED TO EXERCISE ANY COMBINATION OF SIXTEEN LINES:

LINE NUMBER	LINESELECT
-----	-----
0	1
1	2
2	4
3	10
4	20
5	40
6	100
7	200
8	400
9	1000
10	2000
11	4000
12	10000
13	20000
14	40000
15	100000
1.2	3
1.3	5
1.4	11
ETC.	
ALL 16 LINES	177777

9. ERROR REPORTS:

STANDARD ERROR REPORTS ARE GENERATED EXCEPT AS NOTED UNDER THE DESCRIPTION FOR NON-STANDARD PRINTOUTS. SEE THE DEC/X11 USERS GUIDE FOR EXAMPLES.

10. NON-STANDARD PRINTOUTS:

A. HARD ERROR REPORT

ASTAT (STATC) IS THE CONTENTS OF THE RECEIVER BUFFER.

B. DATA ERROR REPORT

WRADR IS THE LINE NUMBER.

ROADR HAS NO MEANING AND WILL ALWAYS EQUAL "0".

11. MNEMONICS:

THE FOLLOWING INFORMATION SHOULD BE USEFUL IN UNDERSTANDING
NAMES GIVEN TO VARIABLES IN THIS PROGRAM:

LINESELECT: .WORD 177777 ; INDICATES LINES SELECTED. (16 LINES UNLESS
; CHANGED BY OPERATOR).
LINESDONE: .BLKW 4 ; USED IN RXSERV AS A BITMAP OF LINES FINISHED
LINECOUNT: .WORD 0 ; NUMBER OF LINES BEING TESTED.

DHUSDONE: .WORD 0 ; INDICATES HOW MANY DHU11'S ARE DONE USED IN DHUTO4
DHUSLEFT: .WORD 0 ; INDICATES DHU11'S LEFT TO DO USED IN DHUTO4
DHU.LIVE: .WORD 0 ; BIT MAP OF NUMBER OF DHU11'S TO TEST.
DHUNOW: .WORD 0 ; CURRENT DHU11 USED IN DHUTO4
DHUCSR: .WORD 0 ; DHU11 CSR.

POINT: .WORD 0 ; POINTS TO CURRENT DHU11 BEING TESTED.
OFFSET: .WORD 0 ; OFFSET TO INDICATE SPECIFIC UNIT.
BUFADR: .WORD 0 ; HOLDS CURRENT BUFFER ADDRESS FOR BINARY COUNT PATTERN.
CDHUCSR: .WORD 0 ; HOLDS CSR.
LCKOUT: .WORD 0 ; ALLOWS ONLY 1 ERROR REPORT AT A TIME
; I.E. SET TO 1 WHEN REPORTING AN ERROR
TXSPEED: .WORD 0 ; TRANSMIT SPEED (BAUD RATE).
RCVSPEED: .WORD 0 ; RECEIVE SPEED (BAUD RATE).

TXBUF: .BLKB 256. ; HOLDS TRANSMIT MESSAGE

:
:
: THE DHR BUFFERS ARE USED IN DHUTO4.
: THE FIRST 16 BYTES HOLD THE NEXT EXPECTED CHARACTER ON
: THE CORRESPONDING LINE
: THE REST OF THE BUFFER IS USED TO HOLD ERROR DATA FOR DHUTO4
:

DHROBUF: .BLKB 256. ; HOLDS THE BINARY COUNT PATTERN FOR DEVICE #1.
DHR1BUF: .BLKB 256. ; HOLDS THE BINARY COUNT PATTERN FOR DEVICE #2.
DHR2BUF: .BLKB 256. ; HOLDS THE BINARY COUNT PATTERN FOR DEVICE #3.

```

DHR3BUF:      .BLKB  256.  ; HOLDS THE BINARY COUNT PATTERN FOR DEVICE #4.
DHUVECT:      .WORD   0    ; HOLDS THE VECTOR ADDRESS OF CURRENT DHU11.

VA:           .WORD   0    ; VIRTUAL ADDRESS.
PA:           .WORD   0    ; PHYSICAL ADDRESS.
EA:           .WORD   0    ; EXTENDED ADDRESS.
PA22:         .WORD   0    ; PHYSICAL ADDRESS (22 BIT).
EA22:         .WORD   0    ; EXTENDED ADDRESS (22 BIT).

ADDR0:        .WORD   0    ; HOLDS CSR ADDRESS OF DHU11 #1.
ADDR1:        .WORD   0    ; HOLDS CSR ADDRESS OF DHU11 #2.
ADDR2:        .WORD   0    ; HOLDS CSR ADDRESS OF DHU11 #3.
ADDR3:        .WORD   0    ; HOLDS CSR ADDRESS OF DHU11 #4.

FLAG:         .WORD   0    ; INDICATES IF DEVICE CAN INTERRUPT O.K.
DHULPR:       .WORD  30    ; LPR BEFORE SETTING BAUD RATES.
LPRSELECT:    .WORD   0    ; LPR AFTER SETTING BAUD RATES.  NEEDED BECAUSE
; THE BAUD RATES ARE SET TO DEFAULTS DURING SELF TEST.

```

538
539
540

.SBTTL MODULE DEFINITIONS

542
543

```

.LIST SEQ,LOC,BIN
.TITLE DHUA DEC/X11 SYSTEM EXERCISER MODULE
; DDXCOM VERSION 6 23-MAY-78
.LIST BIN
;*****
000000 BEGIN:
000000 104 110 125 MODNAM: .ASCII /DHUA / ;MODULE NAME.
000003 101 040
000005 000
000006 000001 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000010 000001 ADDR: 1+0 ;1ST DEVICE ADDR.
000012 240 VECTOR: 1+0 ;1ST DEVICE VECTOR.
000013 240 BR1: .BYTE PRTY5+0 ;1ST BR LEVEL.
000014 000020 BR2: .BYTE PRTY5+0 ;2ND BR LEVEL.
000016 000000 DVID1: 17+1 ;DEVICE INDICATOR 1.
000020 000000 SR1: OPEN ;SWITCH REGISTER 1
000022 000000 SR2: OPEN ;SWITCH REGISTER 2
000024 000000 SR3: OPEN ;SWITCH REGISTER 3
000026 140000 SR4: OPEN ;SWITCH REGISTER 4
;*****
000030 004340' STAT: 140000 ;STATUS WORD.
000032 000222' INIT: START ;MODULE START ADDR.
000034 000000 SPOINT: MODSP ;MODULE STACK POINTER.
000036 000007 PASCNT: 0 ;PASS COUNTER.
000040 000000 ICOUNT: 7 ;# OF ITERATIONS PER PASS=7
000042 000000 SOFCNT: 0 ;LOC TO COUNT ITERATIONS
000044 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000046 000000 SOFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000050 000000 HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000052 000000 SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000054 000000 RANUM: 0 ;# OF SYS ERRORS ACCUMULATED
000056 000000 CONFIG: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000060 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000062 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000064 000000 SVR0: OPEN ;RESERVED FOR MONITOR USE
000066 000000 SVR1: OPEN ;LOC TO SAVE R0.
000070 000000 SVR2: OPEN ;LOC TO SAVE R1.
000072 000000 SVR3: OPEN ;LOC TO SAVE R2.
000074 000000 SVR4: OPEN ;LOC TO SAVE R3.
000076 000000 SVR5: OPEN ;LOC TO SAVE R4.
000100 000000 SVR6: OPEN ;LOC TO SAVE R5.
000102 000000 CSRA: OPEN ;LOC TO SAVE R6.
000104 000000 SBADR: OPEN ;ADDR OF CURRENT CSR.
000106 000000 ACSR: OPEN ;ADDR OF GOOD DATA, OR
000110 000000 WASADR: OPEN ;CONTENTS OF CSR.
000112 004374' ASTAT: OPEN ;ADDR OF BAD DATA, OR
000114 000000 ERRTYP: OPEN ;STATUS REG CONTENTS.
000116 000000 ASB: OPEN ;TYPE OF ERROR
000120 000000 AWAS: OPEN ;EXPECTED DATA.
000122 000040 RSTRT: RESTRT ;ACTUAL DATA.
MODSP: .REPT SPSIZ ;RESTART ADDRESS AFTER END OF PASS
;*****
000222 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: ;MODULE IDENTIFICATION NUMBER=
MODSP: ;MODULE STACK STARTS HERE.
;*****

```

544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

; BIT DEFINITIONS

100000	BIT15	=	100000
040000	BIT14	=	40000
020000	BIT13	=	20000
010000	BIT12	=	10000
004000	BIT11	=	4000
002000	BIT10	=	2000
001000	BIT9	=	1000
000400	BIT8	=	400
000200	BIT7	=	200
000100	BIT6	=	100
000040	BIT5	=	40
000020	BIT4	=	20
000010	BIT3	=	10
000004	BIT2	=	4
000002	BIT1	=	2
000001	BIT0	=	1

; OFFSETS FOR REGISTERS (BASE IS CSR).

000000	CSR	=	0
000002	RBUF	=	2
000004	LPR	=	4
000006	TXCHAR	=	6
000010	LNCTRL	=	10
000012	TX1BUFADR	=	12
000014	TX2BUFADR	=	14
000016	TXBUFCNTR	=	16

; REGISTER DEFINITIONS.

; CSR(R2)	=	CSR.
; RBUF(R2)	=	RECEIVE BUFFER.
; TXCHAR(R2)	=	TRANSMIT CHARACTER FIFO.
; LPR(R2)	=	LINE PARAMETER REGISTER.
; LNCTRL(R2)	=	LINE CONTROL REGISTER.
; TX1BUFADR(R2)	=	TRANSMIT BUFFER ADDRESS REGISTER 1.
; TX2BUFADR(R2)	=	TRANSMIT BUFFER ADDRESS REGISTER 2.
; TXBUFCNTR(R2)	=	TRANSMIT DMA BUFFER COUNTER.

; CSR BIT DEFINITIONS.


```

601
602      100000      TX.ACTION      = BIT15
603      040000      TXINTR.ENABLE = BIT14
604      020000      DIAG.FAIL      = BIT13
605      010000      TXDMA.ERROR    = BIT12
606      000200      RCVDATA.AVAIL  = BIT7
607      000100      RCVINTR.ENABLE = BIT6
608      000040      MASTER.RESET   = BIT5
609
610
611
612      ; RECEIVE BUFFER BIT DEFINITIONS.
613
614
615      100000      DATA.VALID     = BIT15
616      040000      OVERRUN.ERROR  = BIT14
617      020000      FRAMING.ERROR  = BIT13
618      010000      PARITY.ERROR   = BIT12
619
620
621
622      ; TRANSMIT CHARACTER USED IN DHUT02
623
624      000224      TXDATA          = 224
625
626
627
628      ; LINE CONTROL REGISTER BIT DEFINITIONS.
629
630
631      000200      LOCAL.LOOP      = BIT7
632      000004      RX.ENABLE      = BIT2
633      000020      AUTO0          = BIT4
634      000002      AUTO1          = BIT1
635
636
637
638      ; TRANSMIT BUFFER ADDRESS 2 BIT DEFINITIONS.
639
640
641      100000      TX.ENABLE       = BIT15
642      000200      DMA.START      = BIT7
643
644
645
646
647
648      ; THESE ARE THE PROGRAM PARAMETERS:
649 000222 177777  LINESELECT: .WORD 177777 ; INDICATES LINES SELECTED. (16 LINES UNLESS
650                                     ; CHANGED BY OPERATOR).
651 000224          LINESDONE: .BLKW 4      ; USED IN RXSERV AS A BITMAP OF LINES FINISHED
652 000234 000000  LINECOUNT: .WORD 0    ; NUMBER OF LINES BEING TESTED.
653
654 000236 000000  DHUSDONE: .WORD 0      ; INDICATES HOW MANY DHU11'S ARE DONE USED IN DHUT04
655 000240 000000  DHUSLEFT: .WORD 0     ; INDICATES DHU11'S LEFT TO DO USED IN DHUT04
656 000242 000000  DHU.LIVE: .WORD 0    ; BIT MAP OF NUMBER OF DHU11'S TO TEST.
657 000244 000000  DHUNOW: .WORD 0      ; CURRENT DHU11 USED IN DHUT04
    
```

```

658 000246 000000      DHUCSR:      .WORD    0      ; DHU11 CSR.
659
660 000250 000000      POINT:      .WORD    0      ; POINTS TO CURRENT DHU11 BEING TESTED.
661 000252 000000      OFFSET:    .WORD    C      ; OFFSET TO INDICATE SPECIFIC UNIT.
662 000254 000000      BUFADR:    .WORD    0      ; HOLDS CURRENT BUFFER ADDRESS FOR BINARY COUNT PATTERN.
663 000256 000000      CDHUCSR:   .WORD    0      ; HOLDS CSR.
664 000260 000000      LCKOUT:    .WORD    0      ; ALLOWS ONLY 1 ERROR REPORT AT A TIME
665                                     ; I.E. SET TO 1 WHEN REPORTING AN ERROR
666 000262 000000      TXSPEED:   .WORD    0      ; TRANSMIT SPEED (BAUD RATE).
667 000264 000000      RCVSPEED: .WORD    0      ; RECEIVE SPEED (BAUD RATE).
668
669 000266              TXBUF:      .BLKB   256.    ; HOLDS TRANSMIT MESSAGE
670
671
672                                     ; THE JMR BUFFERS ARE USED IN DHU04.
673                                     ; THE FIRST 16 BYTES HOLD THE NEXT EXPECTED CHARACTER ON
674                                     ; THE CORRESPONDING LINE
675                                     ; THE REST OF THE BUFFER IS USED TO HOLD ERROR DATA FOR DHU04
676
677 000666              DHR0BUF:    .BLKB   256.    ; HOLDS THE BINARY COUNT PATTERN FOR DEVICE #1.
678 001266              DHR1BUF:    .BLKB   256.    ; HOLDS THE BINARY COUNT PATTERN FOR DEVICE #2.
679 001666              DHR2BUF:    .BLKB   256.    ; HOLDS THE BINARY COUNT PATTERN FOR DEVICE #3.
680 002266              DHR3BUF:    .BLKB   256.    ; HOLDS THE BINARY COUNT PATTERN FOR DEVICE #4.
681 002666 000000      DHUVECT:   .WORD    0      ; HOLDS THE VECTOR ADDRESS OF CURRENT DHU11.
682
683 002670 000000      VA:        .WORD    0      ; VIRTUAL ADDRESS.
684 002672 000000      PA:        .WORD    0      ; PHYSICAL ADDRESS.
685 002674 000000      EA:        .WORD    0      ; EXTENDED ADDRESS.
686 002676 000000      PA22:     .WORD    0      ; PHYSICAL ADDRESS (22 BIT).
687 002700 000000      EA22:     .WORD    0      ; EXTENDED ADDRESS (22 BIT).
688
689 002702 000000      ADDR0:     .WORD    0      ; HOLDS CSR ADDRESS OF DHU11 #1.
690 002704 000000      ADDR1:     .WORD    0      ; HOLDS CSR ADDRESS OF DHU11 #2.
691 002706 000000      ADDR2:     .WORD    0      ; HOLDS CSR ADDRESS OF DHU11 #3.
692 002710 000000      ADDR3:     .WORD    0      ; HOLDS CSR ADDRESS OF DHU11 #4.
693
694 002712 000000      FLAG:      .WORD    0      ; INDICATES IF DEVICE CAN INTERRUPT O.K.
695 002714 000030      DHULPR:    .WORD   30      ; LPR BEFORE SETTING BAUD RATES.
696 002716 000000      LPRSELECT: .WORD    0      ; LPR AFTER SETTING BAUD RATES. NEEDED BECAUSE
697                                     ; THE BAUD RATES ARE SET TO DEFAULTS DURING SELF TEST.
698
699
700
701
702                                     ; THESE ARE THE ERROR MESSAGES
703
704 002720 0045 0106 0101  ERR.A: .ASCIZ '#FATAL ERROR!! CANNOT SELECT MORE THAN 4 LINES @19.2 K BAUD!#'
      002723 0124 0101 0114
      002726 0040 0105 0122
      002731 0122 0117 0122
      002734 0041 0041 0040
      002737 0040 0103 0101
      002742 0116 0116 0117
      002745 0124 0040 0123
      002750 0105 0114 0105
      002753 0103 0124 0040
      002756 0115 0117 0122
    
```


	002761	105	040	124	
	002764	110	101	116	
	002767	040	064	040	
	002772	114	111	116	
	002775	105	123	040	
	003000	100	061	071	
	003003	056	062	040	
	003006	113	040	102	
	003011	101	125	104	
	003014	041	041	045	
	003017	000			
705	003020	045	106	101	ERR.B: .ASCIZ 'FATAL ERROR!! CANNOT SELECT MORE THAN 2 LINES @38.4 K BAUD!!'
	003023	124	101	114	
	003026	040	105	122	
	003031	122	117	122	
	003034	041	041	040	
	003037	040	103	101	
	003042	116	116	117	
	003045	124	040	123	
	003050	105	114	105	
	003053	103	124	040	
	003056	115	117	122	
	003061	105	040	124	
	003064	110	101	116	
	003067	040	062	040	
	003072	114	111	116	
	003075	105	123	040	
	003100	100	063	070	
	003103	056	064	040	
	003106	113	040	102	
	003111	101	125	104	
	003114	041	041	045	
	003117	000			
706	003120	045	106	101	ERR.C: .ASCIZ 'FATAL ERROR!! SELF-TEST TIMED OUT!! DROP THE UNIT!!'
	003123	124	101	114	
	003126	040	105	122	
	003131	122	117	122	
	003134	041	041	040	
	003137	040	123	105	
	003142	114	106	055	
	003145	124	105	123	
	003150	124	040	124	
	003153	111	115	105	
	003156	104	040	117	
	003161	125	124	041	
	003164	041	040	040	
	003167	104	122	117	
	003172	120	040	124	
	003175	110	105	040	
	003200	125	116	111	
	003203	124	041	041	
	003206	045	000		
707	003210	045	106	101	ERR.D: .ASCIZ 'FATAL ERROR!! SELF-TEST DIAGNOSTICS FAILED!! DROP THE UNIT!!'
	003213	124	101	114	
	003216	040	105	122	
	003221	122	117	122	
	003224	041	041	040	

	003227	040	123	105	
	003232	114	106	055	
	003235	124	105	123	
	003240	124	040	104	
	003243	111	101	107	
	003246	116	117	123	
	003251	124	111	103	
	003254	123	040	106	
	003257	101	111	114	
	003262	105	104	041	
	003265	041	040	040	
	003270	104	122	117	
	003273	120	040	124	
	003276	110	105	040	
	003301	125	116	111	
	003304	124	041	041	
	003307	045	000		
708	003311	045	106	101	ERR.E: .ASCIZ 'FATAL ERROR!! DATA VALID DID NOT SET!! DROP THE UNIT!!'
	003314	124	101	114	
	003317	040	105	122	
	003322	122	117	122	
	003325	041	041	040	
	003330	040	104	101	
	003333	124	101	040	
	003336	126	101	114	
	003341	111	104	040	
	003344	104	111	104	
	003347	040	116	117	
	003352	124	040	123	
	003355	105	124	041	
	003360	041	040	040	
	003363	104	122	117	
	003366	120	040	124	
	003371	110	105	040	
	003374	125	116	111	
	003377	124	041	041	
	003402	045	000		
709	003404	045	106	101	ERR.F: .ASCIZ 'FATAL ERROR!! OVERRUN, PARITY, OR FRAMING ERROR!! DROP THE UNIT!!'
	003407	124	101	114	
	003412	040	105	122	
	003415	122	117	122	
	003420	041	041	040	
	003423	040	117	126	
	003426	105	122	122	
	003431	125	116	054	
	003434	120	101	122	
	003437	111	124	131	
	003442	054	040	117	
	003445	122	040	106	
	003450	122	101	115	
	003453	111	116	107	
	003456	040	105	122	
	003461	122	117	122	
	003464	041	041	040	
	003467	040	104	122	
	003472	117	120	040	
	003475	124	110	105	

	003500	040	125	116	
	003503	111	124	041	
	003506	041	045	000	
710	003511	045	106	101	ERR.G: .ASCIZ 'FATAL ERROR!! SELF-TEST ERROR CODE INDICATES AN ERROR!! DROP THE UNIT!!'
	003514	124	101	114	
	003517	040	105	122	
	003522	122	117	122	
	003525	041	041	040	
	003530	040	123	105	
	003533	114	106	055	
	003536	124	105	123	
	003541	124	040	105	
	003544	122	122	117	
	003547	122	040	103	
	003552	117	104	105	
	003555	040	111	116	
	003560	104	111	103	
	003563	101	124	105	
	003566	123	040	101	
	003571	116	040	105	
	003574	122	122	117	
	003577	122	041	041	
	003602	040	040	104	
	003605	122	117	120	
	003610	040	124	110	
	003613	105	040	125	
	003616	116	111	124	
	003621	041	041	045	
	003624	000			
711	003625	045	105	122	ERR.H: .ASCIZ 'ERROR!! ILLEGAL TRANSMIT INTERRUPT!!'
	003630	122	117	122	
	003633	041	041	040	
	003636	040	111	114	
	003641	114	105	107	
	003644	101	114	040	
	003647	124	122	101	
	003652	116	123	115	
	003655	111	124	040	
	003660	111	116	124	
	003663	105	122	122	
	003666	125	120	124	
	003671	041	041	045	
	003674	000			
712	003675	045	105	122	ERR.I: .ASCIZ 'ERROR!! OVERRUN, PARITY, OR FRAMING ERROR!!'
	003700	122	117	122	
	003703	041	041	040	
	003706	040	117	126	
	003711	105	122	122	
	003714	125	116	054	
	003717	120	101	122	
	003722	111	124	131	
	003725	054	040	117	
	003730	122	040	106	
	003733	122	101	115	
	003736	111	116	107	
	003741	040	105	122	
	003744	122	117	122	

MODULE DEFINITIONS

	003747	041	041	045	
	003752	000			
713	003753	045	105	122	ERR.J: .ASCIZ 'ERROR!! DATA ERROR!!'
	003756	122	117	122	
	003761	041	041	040	
	003764	040	104	101	
	003767	124	101	040	
	003772	105	122	122	
	003775	117	122	041	
	004000	041	045	000	
714	004003	045	105	122	ERR.K: .ASCIZ 'ERROR!! NO LINES SELECTED!!'
	004006	122	117	122	
	004011	041	041	040	
	004014	040	116	117	
	004017	040	114	111	
	004022	116	105	123	
	004025	040	123	105	
	004030	114	105	103	
	004033	124	105	104	
	004036	041	041	045	
	004041	000			
715	004042	045	106	101	ERR.L: .ASCIZ 'FATAL ERROR!! DEVICE DID NOT INTERRUPT PROPERLY!! DROP THE UNIT!!'
	004045	124	101	114	
	004050	040	105	122	
	004053	122	117	122	
	004056	041	041	040	
	004061	040	104	105	
	004064	126	111	103	
	004067	105	040	104	
	004072	111	104	040	
	004075	116	117	124	
	004100	040	111	116	
	004103	124	105	122	
	004106	122	125	120	
	004111	124	040	120	
	004114	122	117	120	
	004117	105	122	114	
	004122	131	041	041	
	004125	040	040	104	
	004130	122	117	120	
	004133	040	124	110	
	004136	105	040	125	
	004141	116	111	124	
	004144	041	041	045	
	004147	000			
716	004150	045	106	101	ERR.M: .ASCII 'FATAL ERROR!! DEVICE DID NOT INTERRUPT PROPERLY!!'
	004153	124	101	114	
	004156	040	105	122	
	004161	122	117	122	
	004164	041	041	040	
	004167	040	104	105	
	004172	126	111	103	
	004175	105	040	104	
	004200	111	104	040	
	004203	116	117	124	
	004206	040	111	116	
	004211	124	105	122	

	004214	122	125	120
	004217	124	040	120
	004222	122	117	120
	004225	105	122	114
	004230	131	041	041
	004233	045		
717	004234	045	103	110
	004237	105	103	113
	004242	040	105	130
	004245	124	105	122
	004250	116	101	114
	004253	040	103	101
	004256	102	114	105
	004261	040	117	122
	004264	040	124	125
	004267	122	116	101
	004272	122	117	125
	004275	116	104	040
	004300	103	117	116
	004303	116	105	103
	004306	124	111	117
	004311	116	041	041
	004314	040	040	104
	004317	122	117	120
	004322	040	124	110
	004325	105	040	125
	004330	116	111	124
	004333	041	041	045
	004336	000		

.ASCIZ 'CHECK EXTERNAL CABLE OR TURNAROUND CONNECTION!! DROP THE UNIT!!'

718
719
720
721

.EVEN
.SBTTL INITIAL SETUP

```

723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749 004340
750
751 004340 016767 173450 173674
752 004346 001012
753
754
755 004350
756 004350 012701 000600
757 004354
758 004354 104407 000000'
004360 104407 000000'
759 004364 005301
760 004366 001372
761 004370 104410 000000'
762
763
764 004374
765
766 004374 042767 177760 173640
767
768
769
770
771
772
773
774 004402 012767 000266' 176260
775 004410 104415 000000' 002670'
776 004416 016767 176250 176252
777 004424 016767 176244 176246
778 004432 006267 176242

;) **
;) NAME: START: RESTART:
;)
;) FUNCTIONAL DESCRIPTION: START AND RESTART ARE ENTRY POINTS
;) INTO THE CODE.
;) START SETS DHU.LIVE UP FROM DVID1
;) RESTART DOES THE 18 BIT CONVERSION
;) ON THE TXBUF ADDRESS FOR DMA
;)
;)
;) CALLING SEQUENCE: IN LINE.
;)
;) INPUTS REQUIRED: MODULE HEADER
;)
;) OUTPUTS GIVEN: "DHU.LIVE" AND THE 18 BIT ADDRESSES
;)
;) REGISTERS USED,UNRESTORED: NONE.
;)
;) SUBROUTINES USED: NONE.
;)
;) RESTRICTIONS: NONE
;)
;) --
;
START:
MOV DVID1,DHU.LIVE ; SETUP # OF DHU11'S TO BE TESTED.
BNE RESTRT ; IF THERE ARE NO DHU11'S TO TEST.

DROP:
MOV #600,R1
DROP1:
BREAK$,BEGIN ; TEMPORARY RETURN TO MONITOR....
BREAK$,BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
DEC R1
BNE DROP1
END$,BEGIN ;

RESTRT:
BIC #+C<17>,DHU.LIVE ; # OF DHU11'S TO BE TESTED.

; WE MUST CONVERT THE 16 BIT VIRTUAL ADDRESS (TXBUF) TO AN 18 PHYSICAL
; ADDRESS

MOV #TXBUF,VA ; GET THE TRANSMIT BUFFER ADDRESS.
GETPA$,BEGIN, VA ; GET PHYSICAL ADDRESS FROM 16-BIT VA
MOV PA,PA22 ; NOW WE HAVE THE 18 BIT PHYSICAL ADDRESS AND
MOV EA,EA22 ; THE 18 BIT EXTENDED ADDRESS.
ASR EA22 ; SHIFT THE EXTENDED ADDRESS BITS

```


J2

779 004436 006267 176236
780 004442 006267 176232
781 004446 006267 176226
782

ASR EA22
ASR EA22
ASR EA22

: SO THEY ARE CORRECT FOR THE DMU11
: NOTE GETPA PUTS THE EXTENDED ADDRESS
: BITS IN BITS 4 AND 5

```

784      ; ) **
785      ; ) NAME:          SETUP1: ( INCLUDING SETUP2: )
786      ; )
787      ; ) FUNCTIONAL DESCRIPTION:  POINT TO FIRST BOARD
788      ; ) AND SET UP THE VECTORS
789      ; )
790      ; )
791      ; ) CALLING SEQUENCE:      IN LINE.
792      ; )
793      ; ) INPUTS REQUIRED:        "DHU.LIVE" AND THE MODULE HEADER
794      ; )
795      ; ) OUTPUTS GIVEN:        "POINT"
796      ; ) R1 = ADDRSES OF POINTER TO CURRENT DHU11
797      ; ) R2 = ADDRESS OF CSR ( CURRENT DHU11 )
798      ; ) R3 = RX INTERRUPT ROUTINE ADDRESS
799      ; ) R4 = TX INTERRUPT ROUTINE ADDRESS
800      ; )
801      ; ) REGISTERS USED,UNRESTORED: R1 - 5 PUSHED ONTO STACK
802      ; )
803      ; ) SUBROUTINES USED:      NONE.
804      ; )
805      ; ) RESTRICTIONS:         NONE.
806      ; )
807      ; ) --
808      ; )
809      ; )
810      004452      SETUP1:
811
812      004452      016701      173564      MOV      DHU.LIVE,R1      ; ANY DHU11'S TO TEST?
813      004456      001734      BEQ      DROP              ; NO, DROP THE MODULE.
814
815      004460      012767      000001      173562      MOV      #1,POINT      ; YES, SETUP DEVICE POINTER TO POINT TO 1ST DEVICE.
816      004466      016767      173314      173552      MOV      ADDR,DHUCSR   ; GET THE CSR ADDRESS FOR FIRST DEVICE.
817      004474      016767      173310      176164      MOV      VECTOR,DHUVECT ; GET THE DEVICE VECTOR.
818
819      004502      012703      007566'      MOV      #DHRO,R3      ; SETUP ADDRESS OF RECEIVER INTERRUPT ROUTINE.
820      004506      012704      007260'      MOV      #DHXO,R4      ; SETUP ADDRESS OF TRANSMIT INTERRUPT ROUTINE.
821      004512      012701      002702'      MOV      #ADDR0,R1     ; SETUP DEVICE ADDRESS TABLE FOR 1ST DEVICE.
822
823
824      ; DETERMINE WHICH UNITS REQUIRE TESTING.
825
826
827      004516      SETUP2:
828
829      004516      036767      173526      173516      BIT      POINT,DHU.LIVE ; DOES OPERATOR WANT THIS DHU11 TESTED?
830      004524      001002      BNE      1$           ; YES, GO TEST IT.
831      004526      000167      001162      JMP      NEXTDHU      ; NO, GO GET ANOTHER DHU11.
832
833      004532      1$:
834      004532      016702      173510      MOV      DHUCSR,R2     ; YES, SAVE THE CSR ADDRESS OF THIS DHU11.
835
836
837
838
839
840      004536      010146      MOV      R1,-(SP)     ; SAVE REGISTERS.
    
```

L2

841	004540	010246	MOV	R2,-(SP)
842	004542	010346	MOV	R3,-(SP)
843	004544	010446	MOV	R4,-(SP)
844	004546	010546	MOV	R5,-(SP)
845				
846			.SBTTL	TEST 1

```

848
849
850      ;) **
851      ;) NAME:                DHUT01
852      ;) FUNCTIONAL DESCRIPTION: THIS SECTION RESETS THE BOARD AND TESTS
853      ;)                      THAT THE BOARD COMES OUT OF RESET
854      ;)
855      ;)
856      ;) CALLING SEQUENCE:    IN LINE.
857      ;)
858      ;) INPUTS REQUIRED:      R2 = CSR ADDRESS OF CURRENT DHU11
859      ;)
860      ;) OUTPUTS GIVEN:      "DHU.LIVE" WILL BE MODIFIED TO DROP A UNIT
861      ;)                      ON A FAIL
862      ;)
863      ;) REGISTERS USED,UNRESTORED: R1 CORRUPTED
864      ;)
865      ;) SUBROUTINES USED:    NONE.
866      ;)
867      ;) RESTRICTIONS:       NONE.
868      ;)
869      ;) --
870      ;
871
872      004550      DHUT01:                ; FIRST WE INITIALIZE THE DEVICE.
873
874      004550      012701      004000      MOV      #4000,R1                ; SET A WATCHDOG TIMER
875      004554      032762      000040      000000      5$:      BIT      #MASTER.RESET,CSR(R2) ; SEE IF DHU11 IS IN RESET
876      004562      001407      BEQ      8$                ; NO THEN WE CAN RESET IT
877
878      004564      104407      000000'      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
879      004570      104407      000000'      BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
880      004574      005301      DEC      R1                ; HAVE WE TIMED OUT
881      004576      001366      BNE      5$                ; NO TRY AGAIN
882      004600      000417      BR      15$              ; YES GO TO THE ERROR REPORT
883      004602      012701      004000      8$:      MOV      #4000,R1                ; SET-UP WATCHDOG TIMER.
884      004606      012762      000040      000000      MOV      #MASTER.RESET,CSR(R2) ; SET MASTER RESET IN CSR.
885      004614      10$:
886      004614      032762      000040      000000      BIT      #MASTER.RESET,CSR(R2) ; IS MASTER RESET STILL SET?
887      004622      001442      BEQ      20$              ; NO, SELF-TEST IS DONE.
888
889      004624      104407      000000'      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
890      004630      104407      000000'      BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
891      004634      005301      DEC      R1                ; DECREMENT WATCHDOG TIMER.
892      004636      001366      BNE      10$              ; YES, WAIT FOR MASTER RESET TO CLEAR.
893
894
895
896      ; SELF-TEST TIMED OUT!! FATAL ERROR!! DROP THE UNIT!!
897
898      004640      15$:
899      004640      016767      173402      173232      MOV      DHUCSR,CSRA          ; YES, MASTER RESET DIDN'T SET. REPORT AN ERROR.
900      004646      016267      000000      173226      MOV      CSR(R2),ACSR        ; PRINTOUT THE CSR ADDRESS (CSRA), THE CONTENTS
901      004654      016267      000002      173222      MOV      RBUF(R2),ASTAT      ; OF THE CSR (ACSR), AND RBUF (ASTAT).
902      004662      012767      000034      173216      MOV      #34,ERRTYP
    
```

```

903
904 004670 104401 000000' 003120'      MSG$,BEGIN,ERR.C      ;ASCII MESSAGE CALL
905      004676 104405 000000' 000000      ;*****
      HRDR$,BEGIN,NULL      ;
      ;*****
906
907 004704 046767 173340 173330      BIC      POINT,DHU.LIVE      ; FATAL ERROR!! DROP THIS DHU11 UNIT!!
908
909 004712 012605      MOV      (SP)+,R5      ; RESTORE REGISTERS.
910 004714 012604      MOV      (SP)+,R4
911 004716 012603      MOV      (SP)+,R3
912 004720 012602      MOV      (SP)+,R2
913 004722 012601      MOV      (SP)+,R1
914
915 004724 000167 000764      JMP      NEXTDHU      ; GO BACK FOR ANOTHER DHU11.
916
917      .SBTTL TEST 2

```



```

919
920 ;) **
921 ;)
922 ;) NAME: DMUT02
923 ;)
924 ;) FUNCTIONAL DESCRIPTION: SEND ONE CHARACTER AROUND THE LOOP
925 ;) AND TEST THE RX AND TX INTERRUPTS
926 ;)
927 ;)
928 ;) CALLING SEQUENCE: IN LINE.
929 ;)
930 ;) INPUTS REQUIRED: R2 = ADDRESS OF DMU11 CSR TO BE TESTED
931 ;) "DMUVECT" , "DMUCSR"
932 ;)
933 ;) OUTPUTS GIVEN: "DMU.LIVE" WILL BE MODIFIED TO DROP A UNIT
934 ;) ON A FAIL
935 ;)
936 ;) REGISTERS USED,UNRESTORED: R1 - 5 PULLED FROM STACK
937 ;)
938 ;) SUBROUTINES USED: NONE.
939 ;)
940 ;) RESTRICTIONS: NONE.
941 ;)
942 ;) --
943
944 004730 200:
945
946 004730 DMUT02:
947
948 004730 300:
949 004730 016200 000002 MOV RBUF(R2),R0 ; CLEAR OUT THE RECEIVER FIFO.
950 004734 100001 BPL 400
951 004736 000774 BR 300
952
953 004740 400:
954 004740 016705 175722 MOV DMUVECT,R5 ; POINT TO VECTOR ADDRESS.
955
956 004744 012725 005314' MOV @RCV,(R5). ; LOAD INTERRUPT RECEIVE ROUTINE ADDRESS.
957 004750 116725 173036 MOVB BR1,(R5). ; LOAD PRIORITY.
958 004754 105025 CLRB (R5).
959 004756 012725 005274' MOV @TX,(R5). ; LOAD INTERRUPT TRANSMIT ROUTINE ADDRESS.
960 004762 116725 173024 MOVB BR1,(R5). ; LOAD PRIORITY.
961 004766 105015 CLRB (R5)
962
963 004770 005000 CLR R0 ; START WITH LINE ZERO.
964 004772 012701 000001 MOV @1,R1 ; POINT TO IT.
965 004776 500:
966 004776 005067 175710 CLR FLAG ; INIT. INTERRUPT INDICATOR.
967 005002 030167 173214 BIT R1,LINSELECT ; IS THIS LINE ON THE DMU11 SELECTED FOR TESTING?
968 005006 001525 BEQ 1000 ; NO, DON'T SET IT UP.
969
970 005010 010062 000000 MOV R0,CSR(R2) ; SELECT LINE NUMBER IN CSR.
971 ; NOTE THE INITIAL RESET SELECTED A
972 ; XMIT AND RECEIVE BAUD OF 9600, NO PARITY,
973 ; 1 STOP BIT, NORMAL OPERATION. WE'RE
974 ; NOT GOING TO CHANGE THAT HERE.
975

```



```

976 005014 030167 173002          BIT    R1,SR3          ; DOES THE OPERATOR WANT TO TEST IN LOCAL LOOPBACK?
977 005020 001404          BEQ    60$           ; YES, SET-UP FOR LOCAL LOOPBACK.
978                                ; NO, SET-UP IN NORMAL OPERATION.
979
980 005022 012762 000026 000010    MOV    @RX.ENABLE!AUTO1!AUTO0,LNCTRL(R2)
981                                ; SET RX ENABLE & NORMAL
982                                ; OPERATION IN LNCTRL.
983 005030 000403          BR     65$
984
985 005032 012762 000226 000010    60$: MOV    @LOCAL.LOOP!RX.ENABLE!AUTO1!AUTO0,LNCTRL(R2)
986                                ; SET RX ENABLE & LOCAL
987                                ; LOOPBACK IN LNCTRL.
988
989                                65$:
990 005040 052762 100000 000014    BIS    @TX.ENABLE,TX2BUFADR(R2) ; SET TX ENABLE
991
992 005046 010004          MOV    R0,R4
993 005050 052704 040100          BIS    @TXINTR.ENABLE!RCVINTR.ENABLE,R4
994                                ; SET TRANSMIT & RECEIVE
995                                ; INTERRUPT ENABLES.
996
997 005054 010462 000000          MOV    R4,CSR(R2)
998
999
1000
1001 005060 012703 020000          MOV    @20000,R3          ; SET WATCHDOG TIMER. WE WILL
1002                                ; TIME OUT IF IT TAKES TOO LONG
1003                                ; TO INTERRUPT SUCCESSFULLY.
1004 005064 112762 000224 000006    MOVB   @TXDATA,TXCHAR(R2) ; PUT TXDATA IN THE TX FIFO
1005                                ; TO SEND A CHARACTER.
1006
1007 005072 104407 000000'          70$: BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
1008 005076 104407 000000'          BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
1009
1010 005102 005303          DEC    R3                ; UPDATE COUNTDOWN.
1011 005104 001052          BNE    90$              ; HAS THE TIMER TIMED OUT?
1012
1013
1014 005106 016767 173134 172764    MOV    DHUCSR,CSRA      ; YES, THE DEVICE DID NOT INTERRUPT PROPERLY!
1015                                ; REPORT AN ERROR.
1016 005114 016267 000000 172760    MOV    CSR(R2),ACSR     ; PRINTOUT THE CSR ADDRESS (CSRA), THE CONTENTS
1017 005122 016267 000002 172754    MOV    RBUF(R2),ASTAT   ; OF THE CSR (ACSR), AND RBUF (ASTAT).
1018
1019 005130 005762 000002          75$: TST    RBUF(R2)        ; FLUSH OUT RECEIVER SILO.
1020 005134 100775          BMI    75$
1021 005136 012762 000000 000000    MOV    @0,CSR(R2)      ; DISABLE TRANSMIT & RECEIVER INTERRUPTS.
1022
1023
1024 005144 022767 000002 175540    CMP    @BIT1,FLAG      ; DID TRANSMITTER INTERRUPT?
1025 005152 001007          BNE    80$             ; NO IT DIDN'T, REPORT INTERRUPT ERROR.
1026 005154 030167 172642          BIT    R1,SR3          ; EXTERNAL LOOPBACK MODE (NORMAL)?
1027 005160 001404          BEQ    80$             ; NO, REPORT INTERRUPT ERROR.
1028 005162 104401 000000' 004150'  MSG$,BEGIN,ERR.M      ;ASCII MESSAGE CALL
1029 005170 000403          BR     85$
1030 005172
1031 005172 104401 000000' 004042'  80$: MSG$,BEGIN,ERR.L      ;ASCII MESSAGE CALL

```


1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109

```

;) **
;)
;) NAME: CHECKLINES:
;) FUNCTIONAL DESCRIPTION: CHECKS THAT WE HAVE LINES SELECED AND
;) NOT TOO MANY LINES IF ON HIGH BAUD
;) RATES, CHECK FOR DEFAULT ( SR2 = 0 )
;) SET UP THE BAUDRATE
;)
;) CALLING SEQUENCE: IN LINE.
;)
;) INPUTS REQUIRED: R2 = CURRENT DHU11 CSR
;) "DHU.LIVE" AND THE MODULE HEADER
;)
;) OUTPUTS GIVEN: NONE.
;)
;) REGISTERS USED,UNRESTORED: ALL
;)
;) SUBROUTINES USED: BAUDRATE
;)
;) RESTRICTIONS: NONE.
;)
;) --

```

1110 005346

CHECKLINES:

```

1111
1112 005346 022767 000000 172646      CMP    #0,LINSELECT      ; MAKE SURE THERE ARE LINES SELECTED.
1113 005354 001024                    BNE    3$                ; YES, THERE ARE. GO TEST THEM.
1114
1115 005356 016767 172664 172514      MOV    DHUCSR,CSRA       ; NO, THERE ARE NO LINES SELECTED. REPORT AN ERROR
1116 005364 016267 000000 172510      MOV    CSR(R2),ACSR      ; PRINTOUT THE CSR ADDRESS (CSRA), THE CONTENTS
1117 005372 016267 000002 172504      MOV    RBUF(R2),ASTAT    ; OF THE CSR (ACSR), AND RBUF (ASTAT).
1118
1119 005400 104401 000000' 004003'    MSG$,BEGIN,ERR,K        ; ASCII MESSAGE CALL
1120 005406 104405 000000' 000000      ;*****
;HDR$,BEGIN,NULL        ;
;*****
1121
1122 005414 046767 172630 172620      BIC    POINT,DHU.LIVE    ; FATAL ERROR!! DROP THIS DHU11 'NIT!!
1123 005422 000167 000266                    JMP    NEXTDHU           ; GO BACK FOR ANOTHER DHU11.
1124
1125 005426                    3$:
1126 005426 012700 000020                    MOV    #16.,R0           ; SET-UP BIT COUNTER.
1127 005432 005067 172576                    CLR    LINECOUNT       ; INITIALIZE THE "LINES SELECTED INDICATOR".
1128 005436 022767 000000 172354      CMP    #0,SR2            ; DOES OPERATOR WANT TO RUN AT 9600 BAUD?
1129 005444 001005                    BNE    5$                ; NO, DETERMINE THE BAUD RATE DESIRED.
1130
1131 005446 012767 000015 172342      MOV    #15,SR1           ; YES, SET-UP BAUD USING 9600.
1132 005454 000167 000200                    JMP    12$               ; GO DO THE SET-UP.
1133
1134 005460                    5$:
1135                                ; FIRST, HOWEVER, WE MUST DETERMINE HOW MANY
1136                                ; LINES HAVE BEEN SELECTED FOR TESTING.
1137 005460 006167 172536      ROL    LINSELECT         ; PLACE THE LINSELECT BIT INTO THE CARRY BIT.
1138 005464 103002                    BCC    7$                ; IS THIS LINE SELECTED?

```



```

1139 005466 005267 172542          INC      LINECOUNT          ; YES, INDICATE THAT IT IS.
1140
1141
1142
1143
1144 005472          7$:
1145 005472 077006          SOB      R0,5$              ; UPDATE COUNT-DOWN UNTIL ALL BITS ARE CHECKED.
1146
1147 005474 006167 172522          ROL      LINESELECT         ; RESTORE LINESELECT.
1148 005500 022767 000016 172310          CMP      #16,SR1            ; IS SELECTED BAUD RATE 19.2K?
1149 005506 001030          BNE      10$                ; NO, SEE IF IT IS SET FOR 38.4K.
1150 005510 022767 000004 172516          CMP      #4,LINECOUNT      ; YES, ARE THERE MORE THAN 4 LINES SELECTED?
1151 005516 002060          BGE      12$                ; NO, THAT'S GOOD. GO SET-UP THE BAUD RATE.
1152
1153 005520 016767 172522 172352          MOV      DHUCSR,CSRA        ; YES, TOO MANY LINES SELECTED. REPORT AN ERROR.
1154 005526 016267 000000 172346          MOV      CSR(R2),ACSR       ; PRINTOUT THE CSR ADDRESS (CSRA), THE CONTENTS
1155 005534 016267 000002 172342          MOV      RBUF(R2),ASTAT     ; OF THE CSR (ACSR), AND RBUF (ASTAT).
1156
1157 005542 104401 000000' 002720'          MSG$,BEGIN,ERR.A          ;ASCII MESSAGE CALL
1158                                     ; RUNNING CONCURRENTLY AT 19.2K BAUD!
1159                                     ;*****
005550 104405 000000' 000000          HRDR$,BEGIN,NULL          ;
                                     ;*****
1160
1161 005556 046767 172466 172456          BIC      POINT,DMU.LIVE     ; FATAL ERROR!! DROP THIS DMU11 UNIT!!
1162 005564 000167 000124          JMP      NEXTDMU           ; GO BACK FOR ANOTHER DMU11.
1163
1164 005570          10$:
1165
1166 005570 022767 000017 172220          CMP      #17,SR1            ; IS SELECTED BAUD RATE 38.4K?
1167 005576 001030          BNE      12$                ; NO, GO SETUP ITS BAUD RATE.
1168 005600 022767 000002 172426          CMP      #2,LINECOUNT     ; YES, ARE THERE MORE THAN 2 LINES SELECTED?
1169 005606 002024          BGE      12$                ; NO, GO SET-UP THE BAUD RATE.
1170
1171 005610 016767 172432 172262          MOV      DHUCSR,CSRA        ; YES, TOO MANY LINES SELECTED. REPORT AN ERROR.
1172 005616 016267 000000 172256          MOV      CSR(R2),ACSR       ; PRINTOUT THE CSR ADDRESS (CSRA), THE CONTENTS
1173 005624 016267 000002 172252          MOV      RBUF(R2),ASTAT     ; OF THE CSR (ACSR), AND RBUF (ASTAT).
1174
1175 005632 104401 000000' 003020'          MSG$,BEGIN,ERR.B          ;ASCII MESSAGE CALL
1176                                     ; RUNNING CONCURRENTLY AT 38.4K BAUD!
1177                                     ;*****
005640 104405 000000' 000000          HRDR$,BEGIN,NULL          ;
                                     ;*****
1179
1180 005646 046767 172376 172366          BIC      POINT,DMU.LIVE     ; FATAL ERROR!! DROP THIS DMU11 UNIT!!
1181 005654 000167 000034          JMP      NEXTDMU           ; GO BACK FOR ANOTHER DMU11.
1182
1183 005660          12$:
1184 005660 004767 002704          JSR      PC,BAUDRATE        ; GO SET-UP THE BAUDRATE.
1185
1186
1187
1188

```



```

1190 ;) **
1191 ;)
1192 ;) NAME: VECT
1193 ;)
1194 ;) FUNCTIONAL DESCRIPTION: SET UP VECTORS FOR EACH BOARD
1195 ;)
1196 ;)
1197 ;) CALLING SEQUENCE: IN LINE.
1198 ;)
1199 ;) INPUTS REQUIRED: MODULE HEADER,
1200 ;) "DHUCSR" AND "DHUVECT"
1201 ;)
1202 ;) OUTPUTS GIVEN: VECTOR TABLE AT CURRENT DHU11 VECTOR ADDRESS
1203 ;)
1204 ;) REGISTERS USED,UNRESTORED: R1 , 3 , 4 , 5
1205 ;)
1206 ;) SUBROUTINES USED: NONE.
1207 ;)
1208 ;) RESTRICTIONS: NONE.
1209 ;)
1210 ;) --

```

```

1211
1212
1213
1214 005664 VECT:
1215
1216 005664 016705 174776 MOV DHUVECT,R5 ; POINT TO THE VECTOR.
1217 005670 016711 172352 MOV DHUCSR,(R1) ; MOVE THE CSR ADDRESS INTO THE ADDRESS TABLE.
1218 005674 010325 MOV R3,(R5)+ ; MOVE THE ADDRESS OF THE RECEIVER INTERRUPT
1219 ; ROUTINE & THE PRIORITY OF THIS DHU11 INTO
1220 005676 116725 172110 MOVB BR1,(R5)+ ; THE VECTOR ADDRESS FOR THIS DHU11.
1221 005702 105025 CLRB (R5)+
1222 005704 010425 MOV R4,(R5)+ ; MOVE THE ADDRESS OF THE TRANSMIT INTERRUPT
1223 ; ROUTINE & THE PRIORITY OF THIS DHU11 INTO
1224 005706 116725 172100 MOVB BR1,(R5)+ ; THE VECTOR ADDRESS FOR THIS DHU11.
1225 005712 105015 CLRB (R5)
1226

```

1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266

```

;) **
;)
;) NAME: NEXTDHU
;)
;) FUNCTIONAL DESCRIPTION: SET UP POINTERS FOR THE NEXT DHU11
;) GOTO NEXT TEST IF ALL DHU11'S DONE
;)
;) CALLING SEQUENCE: IN LINE.
;)
;) INPUTS REQUIRED: "POINT" , "DHUCSR" AND "DHUVECT"
;)
;) OUTPUTS GIVEN: "POINT" , "DHUCSR" AND "DHUVECT"
;)
;) REGISTERS USED,UNRESTORED: R1 , 3 , 4
;)
;) SUBROUTINES USED: NONE.
;)
;) RESTRICTIONS: NONE.
;)
;) --

```

NEXTDHU:

```

ASL POINT ; POINT TO NEXT DHU11.
ADD #2,R1 ; POINT TO NEXT LOC. IN ADDRESS TABLE.
ADD #DHR1-DHRO,R3 ; POINT TO NEXT DHU11'S RX INTERRUPT ROUTINE.
ADD #DHX1-DHXO,R4 ; POINT TO NEXT DHU11'S TX INTERRUPT ROUTINE.
BIT #20,POINT ; HAVE WE SETUP ALL DHU11S?
BNE DHU03 ; YES, LET'S GO TEST THEM!
ADD #20,DHUCSR ; NO, GO SETUP ANOTHER DHU11.
ADD #10,DHUVECT
JMP SETUP2

```

.SBTTL TEST 3

```

1268 ;) **
1269 ;)
1270 ;) NAME: DHUT03
1271 ;)
1272 ;) FUNCTIONAL DESCRIPTION: RESET BOARD AND TEST STATE OF
1273 ;) DIAG.FAIL BIT AND SELF TEST CODES
1274 ;)
1275 ;)
1276 ;) CALLING SEQUENCE: IN LINE.
1277 ;)
1278 ;) INPUTS REQUIRED: "DHUCSR" , "DHU.LIVE"
1279 ;)
1280 ;) OUTPUTS GIVEN: "DHU.LIVE" WILL BE MODIFIED TO DROP A UNIT
1281 ;) ON A FAIL
1282 ;)
1283 ;) REGISTERS USED,UNRESTORED: ALL
1284 ;)
1285 ;) SUBROUTINES USED: NONE.
1286 ;)
1287 ;) RESTRICTIONS: NONE.
1288 ;)
1289 ;) --
1290 ;)
1291 ;)
1292 ;)
1293 ;)

```

```

1294 005764 DHUT03:
1295
1296 005764 012767 000001 172256 MOV #1,POINT ; SETUP DEVICE POINTER TO POINT TO 1ST DEVICE.
1297 005772 016767 172010 172246 MOV ADDR,DHUCSR ; SETUP ADDRESS OF 1ST DEVICE.
1298
1299 006000 DHU.L1:
1300
1301 006000 036767 172244 172234 BIT POINT,DHU.LIVE ; DOES THE OPERATOR WANT TO TEST THIS DEVICE?
1302 006006 001002 BNE 3$ ; YES, GO TEST IT.
1303 006010 000167 000534 JMP DHU.N1 ; NO, GO LOOK AT THE NEXT DHU11.
1304 006014 3$:
1305 006014 016702 172226 MOV DHUCSR,R2 ; SETUP THIS DHU11'S DEVICE ADDRESS.
1306
1307
1308
1309
1310
1311 ; THE ONBOARD SELF-TEST DIAGNOSTIC WILL NOW BE RUN.
1312 ; THIS IS ACCOMPLISHED BY SETTING THE MASTER RESET IN THE CSR. THE
1313 ; CSR MASTER RESET BIT IS CLEARED WHEN THE SELF TEST IS COMPLETED.
1314
1315 006020 012701 004000 MOV #4000,R1 ; SET A WATCHDOG TIMER
1316 006024 032762 000040 000000 5$: BIT #MASTER.RESET,CSR(R2) ; SEE IF DHU11 IS IN RESET
1317 006032 001407 BEQ 8$ ; NO THEN WE CAN RESET IT
1318
1319 006034 104407 000000' BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
1320 006040 104407 000000' BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1321 006044 005301 DEC R1 ; HAVE WE TIMED OUT
1322 006046 001366 BNE 5$ ; NO TRY AGAIN
1323 006050 000417 BR 15$ ; YES GO TO THE ERROR REPORT

```

```

1324 006052 012701 004000      8$:  MOV    #4000,R1          ; SET-UP WATCHDOG TIMER.
1325 006056 012762 000040 000000  MOV    #MASTER.RESET,CSR(R2) ; SET MASTER RESET IN CSR.
1326 006064                                10$:
1327 006064 032762 000040 000000  BIT    #MASTER.RESET,CSR(R2) ; IS MASTER RESET STILL SET?
1328 006072 001435                                BEQ    20$                ; NO. SELF-TEST IS DONE.
1329
1330 006074 104407 000000'      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
      006100 104407 000000'      BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
1331
1332 006104 005301              DEC    R1                ; DECREMENT WATCHDOG TIMER.
1333 006106 001366              BNE    10$                ; YES, WAIT FOR MASTER RESET TO CLEAR.
1334 006110                      15$:
1335
1336
1337
1338 ; SELF-TEST TIMED OUT!! FATAL ERROR!! DROP THE UNIT!!
1339
1340
1341 006110 016767 172132 171762      MOV    DHUCSR,CSRA      ; YES, MASTER RESET DIDN'T SET. REPORT AN ERROR.
1342 006116 016267 000000 171756      MOV    CSR(R2),ACSR     ; PRINTOUT THE CSR ADDRESS (CSRA), THE CONTENTS
1343 006124 016267 000002 171752      MOV    RBUF(R2),ASTAT  ; OF THE CSR (ACSR), AND RBUF (ASTAT).
1344 006132 012767 000034 171746      MOV    #34,ERRTYP
1345
1346 006140 104401 000000' 003120'  MSG$,BEGIN,ERR.C      ;ASCII MESSAGE CALL
1347      006146 104405 000000' 000000  ;*****
      HRDR$,BEGIN,NULL  ;
      ;*****
1348
1349 006154 046767 172070 172060      BIC    POINT,DHU.LIVE  ; FATAL ERROR!! DROP THIS DHU11 UNIT!!
1350 006162 000167 000362              JMP    DHU.N1          ; GO BACK FOR ANOTHER DHU11.
1351
1352
1353
1354 ; THE SELF-TEST DID COMPLETE FOR THIS DHU11. BUT, WERE THERE ANY FAILURES?
1355
1356
1357 006166                                20$:
1358 006166 032762 020000 000000  BIT    #DIAG.FAIL,CSR(R2) ; DID SELF-TEST DIAGNOSTICS PASS?
1359 006174 001426                                BEQ    30$                ; YES, IF "DIAGNOSTICS FAILURE" BIT CLEAR.
1360
1361
1362
1363 ; SELF-TEST DIAGNOSTICS FAILED!! FATAL ERROR!! DROP THE UNIT!!
1364
1365
1366 006176 016767 172044 171674      MOV    DHUCSR,CSRA      ; REPORT CSR.
1367 006204 016267 000000 171670      MOV    CSR(R2),ACSR     ; REPORT CSR CONTENTS.
1368 006212 010567 171666              MOV    R5,ASTAT         ; REPORT CONTENTS OF RBUF.
1369 006216 012767 000034 171662      MOV    #34,ERRTYP      ; REPORT ERROR TYPE.
1370
1371 006224 104401 000000' 003210'  MSG$,BEGIN,ERR.D      ;ASCII MESSAGE CALL
1372      006232 104405 000000' 000000  ;*****
      HRDR$,BEGIN,NULL  ;
      ;*****
1373
1374 006240 046767 172004 171774      BIC    POINT,DHU.LIVE  ; REMOVE THIS FAULTY DEVICE FROM TESTING.
1375 006246 000167 000276              JMP    DHU.N1          ; GO GET NEXT DHU11.

```


1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429

```

; SELF-TEST PASSED O.K.

30$:
MOV #8.,R4 ; SETUP TO READ THE 8 CHAR. CODES FROM RBUF.
MOV #4000.,R1 ; SETUP WATCHDOG TIMER.

40$:
MOV RBUF(R2),R3 ; READ RBUF FIFO. A VALID CHAR. WILL SET BIT
BMI 50$ ; #15 (FIFO NOT EMPTY).

BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.

DEC R1 ; DECREMENT WATCHDOG TIMER.
BNE 40$ ; WAIT, WE WANT TO READ RBUF FIFO.

; DATA VALID DID NOT SET!! FATAL ERROR!! DROP THIS DHU1!!

; BIT #15 DIDN'T SET! REPORT AN ERROR!
MOV DHUCSR,CSRA ; REPORT CSR.
MOV CSR(R2),ACSR ; REPORT CSR CONTENTS.
MOV R3,ASTAT ; REPORT CONTENTS OF RBUF.
MOV #34,ERRTYP ; REPORT ERROR TYPE.

MSG$,BEGIN,ERR.E ;ASCII MESSAGE CALL
;*****
HDRR$,BEGIN,NULL ;
;*****

BIC POINT,DHU.LIVE ; REMOVE THIS FAULTY DHU11 FROM TESTING.
JMP DHU.N1 ; GO GET NEXT DHU11.

; IT IS A VALID CHARACTER. CHECK BITS 12-14 FOR A POSSIBLE ERROR.

50$:
BIT #OVERRUN.ERROR!FRAMING.ERROR!PARITY.ERROR,R3
; IS IT MODEM STATUS,
; SELF-TEST ERROR CODE,
; OR DATA?
BEQ 40$ ; IF NONE SET THEN IT IS DATA. IT DOESN'T COUNT.
ADD #BIT12,R3 ; MAKE SURE ALL THE BITS WERE SET!
BPL 60$ ; THEY WERE. THAT'S GOOD.

; OVERRUN, PARITY OR FRAMING ERROR!! FATAL ERROR!! DROP THIS UNIT!!

```

```

1430                                     ; IF OVERFLOW TO BIT 15 DID NOT OCCUR,
1431                                     ; THEN WE KNOW ALL 3 BITS WERE NOT SET!
1432                                     ; REPORT AN ERROR.
1433 006374 016767 171646 171476      MOV    DHUCSR,CSRA      ; REPORT CSR.
1434 006402 016267 000000 171472      MOV    CSR(R2),ACSR   ; REPORT CSR CONTENTS.
1435 006410 010367 171470              MOV    R3,ASTAT       ; REPORT RBUF CONTENTS.
1436 006414 012767 000034 171464      MOV    #34,ERRTYP    ; REPORT ERROR TYPE.
1437
1438 006422 104401 000000' 003404'    MSG$,BEGIN,ERR.F     ;ASCII MESSAGE CALL
1439 006430 104405 000000' 000000      ;*****
HRDR$,BEGIN,NULL      ;
;*****

1440
1441 006436 046767 171606 171576      BIC    POINT,DHU.LIVE ; REMOVE THIS FAULTY DHU11 FROM TESTING.
1442 006444 000167 000100              JMP    DHU.N1         ; GO GET NEXT DHU11.
1443
1444
1445
1446                                     ; THE "ALL ONES" CODE FOR BITS 12-14 IN RBUF INDICATES THAT MODEM
1447                                     ; STATUS OR SELF-TEST DIAGNOSTIC INFO. IS BEING HELD IN RBUF BITS 0-7.
1448                                     ; WE WANT TO LOOK AT THE SELF-TEST CODE TO ASSURE THAT THE SELF-TEST
1449                                     ; COMPLETED O.K.
1450
1451
1452 006450                                     60$:
1453 006450 032703 000001              BIT    #BIT0,R3      ; IS THIS MODEM STATUS OR SELF-TEST ERROR CODE?
1454 006454 001702                      BEQ    40$           ; IGNOR THE MODEM STATUS. WE JUST WANT
1455                                     ; SELF-TEST ERROR CODES.
1456
1457 006456 105703                      TSTB   R3            ; LOOK TO SEE IF ROM VERSION OR SELF-TEST ERROR COD
E.
1458 006460 100031                      BPL    70$           ; IF BIT7=0 THEN ROM VERSION. READ AND IGNOR.
1459 006462 120327 000201              CMPB   R3,#201       ; DID WE GET SUCCESSFUL COMPLETION CODE?
1460 006466 001426                      BEQ    70$           ; IF SO, CODE 201 RETURNED. ANY OTHER CODE IS
1461                                     ; AN ERROR!
1462
1463
1464
1465                                     ; SELF-TEST ERROR CODE INDICATES AN ERROR!! FATAL ERROR!! DROP THIS DHU11!!
1466
1467                                     ; REPORT AN ERROR.
1468 006470 016767 171552 171402      MOV    DHUCSR,CSRA   ; REPORT CSR.
1469 006476 016267 000000 171376      MOV    CSR(R2),ACSR  ; REPORT CSR CONTENTS.
1470 006504 010367 171374              MOV    R3,ASTAT       ; REPORT RBUF CONTENTS.
1471 006510 012767 000034 171370      MOV    #34,ERRTYP    ; REPORT ERROR TYPE.
1472
1473 006516 104401 000000' 003511'    MSG$,BEGIN,ERR.G     ;ASCII MESSAGE CALL
1474 006524 104405 000000' 000000      ;*****
HRDR$,BEGIN,NULL      ;
;*****

1475
1476 006532 046767 171512 171502      BIC    POINT,DHU.LIVE ; REMOVE THIS FAULTY DHU11 FROM TESTING.
1477 006540 000167 000004              JMP    DHU.N1         ; GO GET NEXT DHU11.
1478
1479
1480
1481
1482

```

```

1483      ; THESE ARE THE SELF-TEST ERROR CODES
1484      ;
1485      ;
1486      ; CODE                MEANING
1487      ;
1488      ; 201                SELF TEST NULL CODE (OK)
1489      ; 203                SELF TEST CODE SKIPPED
1490      ; 211                BASIC DATA PATH ERROR FRO PROC2
1491      ; 213                UNDEFINED UART ERROR
1492      ; 225                PROC1 TO EXTERNAL RAM ERROR
1493      ; 227                PROC2 TO EXTERNAL RAM ERROR
1494      ; 231                PROC1 TO INTERNAL RAM ERROR
1495      ; 233                PROC2 TO INTERNAL RAM ERROR
1496      ; 235                PROC1 ROM ERROR
1497      ; 237                PROC2 ROM ERROR
1498      ; 241-277           UART ACCESS OR FUNCTION
1499      ;
1500      ; BIT 7=0 AND BIT0=1 ROM VERSION
1501      ; ALL OTHER CODES ARE UNDEFINED ERRORS.

```

```

1504 006544
1505 006544 005304
1506 006546 001245
1507
1508
1509
1510
1511
1512

```

```

70$:      DEC      R4      ; NEXT CHARACTER.
          BNE      40$     ; HAVE WE DONE ALL 16 CHARACTERS?
          ; IF NOT, GO BACK & DO THEM.

```

1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549

006550
006550 005767 171466
006554 001002
006556 000167 175566
006562
006562 106367 171462
006566 032767 000020 171454
006574 001005
006576 062767 000020 171442
006604 000167 177170

```
;) **
;)
;) NAME: DHU.N1
;) FUNCTIONAL DESCRIPTION: SET UP POINTERS FOR THE NEXT DHU11
;) GOTO NEXT TEST IF ALL DHU11'S DONE
;)
;) CALLING SEQUENCE: IN LINE.
;)
;) INPUTS REQUIRED: "POINT" AND "DHUCSR"
;)
;) OUTPUTS GIVEN: "POINT" AND "DHUCSR"
;)
;) REGISTERS USED, UNRESTORED: NONE
;)
;) SUBROUTINES USED: NONE.
;)
;) RESTRICTIONS: NONE.
;)
;) --
```

```
DHU.N1:
TST DHU.LIVE ; ANY DHU11'S TO TEST?
BNE 10$
JMP DROP
10$: ASLB POINT ; LOOK AT NEXT DHU11.
BIT #20, POINT ; HAVE WE TESTED ALL DHU11'S.
BNE DHU04 ; YES, GOTO NEXT TEST SECTION.
ADD #20, DHUCSR ; NO, GET NEXT DHU11 ADDRESS.
JMP DHU.L1 ; GO BACK AND TEST.
```

.SBTTL TEST 4


```

1551 ;
1552 ;) **
1553 ;)
1554 ;) NAME: DMU04
1555 ;)
1556 ;) FUNCTIONAL DESCRIPTION: THIS SECTION SETS UP THE DMU11 TO PASS DATA IN
1557 ;) INTERNAL LOOPBACK MODE OR TO LOOPBACK DATA
1558 ;) EXTERNALLY IN NORMAL MODE.
1559 ;)
1560 ;)
1561 ;) CALLING SEQUENCE: IN LINE.
1562 ;)
1563 ;) INPUTS REQUIRED: MODULE HEADER
1564 ;)
1565 ;) OUTPUTS GIVEN: "LINESELECT" , "DHUSLEFT"
1566 ;)
1567 ;) REGISTERS USED,UNRESTORED: ALL
1568 ;)
1569 ;) SUBROUTINES USED: NONE.
1570 ;)
1571 ;) RESTRICTIONS: NONE.
1572 ;)
1573 ;) --
1574 ;)
1575 ;)
1576 006610 DMU04:
1577
1578 006610 005067 171444 CLR LCKOUT ; INIT. LOCKOUT.
1579 006614 005067 171404 CLR LINESDONE ; INIT. LINE DONE INDICATOR.
1580 006620 005067 171402 CLR LINESDONE+2 ; INIT. LINE DONE INDICATOR.
1581 006624 005067 171400 CLR LINESDONE+4 ; INIT. LINE DONE INDICATOR.
1582 006630 005067 171376 CLR LINESDONE+6 ; INIT. LINE DONE INDICATOR.
1583
1584 006634 005067 171404 CLR DHUNOW ; START WITH DEVICE #0.
1585 006640 016767 171142 171400 MOV ADDR,DHUCSR ; GET ADDRESS OF FIRST DMU11.
1586 006646 016767 171136 174012 MOV VECTOR,DHUVECT ; GET VECTOR OF FIRST DMU11.
1587
1588 006654 012767 000001 171366 MOV #1,POINT ; RECORD WORKING ON 1ST DMU11.
1589 006662 005067 171350 CLR DHUSDONE ; CLEAR THE INDICATOR OF HOW MANY DMU11'S
1590 ; ARE DONE,
1591 006666 005067 171346 CLR DHUSLEFT ; AND HOW MANY ARE LEFT TO DO.
1592
1593 006672 DMU.L2:
1594
1595 006672 036767 171352 171342 BIT POINT,DHU.LIVE ; DOES OPERATOR WANT THIS DMU11 TESTED?
1596 006700 001503 BEQ DHU.N2 ; NO, GO LOOK AT THE NEXT DMU11.
1597
1598 006702 016702 171340 MOV DHUCSR,R2 ; SAVE DMU11 ADDRESS FOR SUBROUTINES.
1599
1600
1601
1602
1603 ; SETUP THE COUNT PATTERN TO SEND VIA THE TRANSMITTER.
1604 ; THE COUNT PATTERN IS 224-377 (148-255 DECIMAL).
1605
1606
1607 006706 DMU.BUF:
    
```

```

1608
1609 006706 012700 000266'      MOV    #TXBUF,R0      ; NOW WRITE COUNT PATTERN INTO XMIT BUFFER.
1610 006712 012701 000224      MOV    #224,R1       ; START WITH 224 UP TO 377 (148-255 DECIMAL).
1611 006716                      10$:  MOVB  R1,(R0)+       ; STORE PATTERN.
1612 006716 110120              INCB  R1              ; UPDATE PATTERN UNTIL PATTERN 377 IS
1613 006720 105201              ; STORED.
1614                              ; HAVE WE STORED UP TO 377 YET?
1615 006722 001375              BNE   10$
1616
1617      ; INITIALISE THE RX BUFFERS IN USE WITH THE FIRST 16 BYTES AS THE FIRST
1618      ; EXPECTED CHARACTER AND THE REST CLEARED
1619
1620 006724                      20$:  MOV    #DHROBUF,R0    ; GET RECEIVER BUFFER ADDRESS.
1621 006724 012700 000666'      MOV    DHUNOW,R1
1622 006730 016701 171310
1623 006734                      30$:  DEC    R1              ; IS THIS DMU11 #0?
1624 006734 005301              BMI    40$           ; YES THEN R0 IS CORRECT
1625 006736 100403              ADD    #DHR1BUF-DHROBUF,R0 ; NO UPDATE R0 FOR NEXT BUFFER ADDRESS
1626 006740 062700 000400      BR    30$           ; DO MORE UNTIL CORRECT BUFFER ADDRESS OBTAINED.
1627 006744 000773
1628
1629 006746                      40$:  INC    DHUSLEFT      ; KEEP TRACK OF HOW MANY DMU11'S HAVE TO BE DONE.
1630 006746 005267 171266
1631
1632 006752 012701 000020      MOV    #16.,R1       ; SETUP COUNT FOR 16 LINES.
1633 006756                      45$:  MOVB  #224,(R0)+     ; DEPOSIT 1ST EXPECTED DATA VALUE INTO RECEIVER BIT
1634 006756 112720 000224      ; MAP FOR THIS UNIT.
1635                              ; NEXT LINE.
1636 006762 005301              BNE   45$           ; HAVE WE FINISHED 16 LINES YET?
1637 006764 001374
1638
1639 006766 012701 000170      MOV    #120.,R1     ; SETUP COUNT FOR 240 BYTE SIZE LOCATIONS.
1640 006772                      47$:  CLR    (R0)+         ; CLEAR REMAINDER OF RECEIVER BIT MAP FOR THIS UNIT
1641 006772 005020              ; HAVE WE FINISHED 240 LOCATIONS YET?
1642 006774 005301              DEC    R1
1643 006776 001375              BNE   47$
1644
1645
1646
1647
1648
1649      ; LINE PARAMETER ROUTINE.
1650
1651
1652 007000 005000              CLR    R0              ; START WITH LINE ZERO.
1653 007002 012701 000001      MOV    #1,R1         ; POINT TO IT.
1654 007006                      50$:  BIT    R1,LINSELECT  ; IS THIS LINE ON THE DMU11 SELECTED FOR TESTING?
1655 007006 030167 171210      BEQ    70$           ; NO, DON'T SET IT UP.
1656 007012 001433
1657
1658 007014 010062 000000      MOV    R0,CSR(R2)   ; SELECT LINE NUMBER IN CSR.
1659                              ; NOTE THE INITIAL RESET SELECTED A
1660                              ; XMIT AND RECEIVE BAUD OF 9600, NO PARITY.
1661                              ; 1 STOP BIT, NORMAL OPERATION. WE'RE
1662                              ; NOT GOING TO CHANGE THAT HERE.
1663
1664 007020 030167 170776      BIT    R1,SR3        ; DOES THE OPERATOR WANT TO TEST IN LOCAL LOOPBACK?

```

```

1665 007024 001404          BEQ      52$          ; YES, SET-UP FOR LOCAL LOOPBACK.
1666
1667                          ; NO, SET-UP IN NORMAL OPERATION.
1668
1669 007026 012762 000026 000010      MOV      @RX.ENABLE!AUTO1!AUTO0, LNCTRL(R2)
1670                          ; SET RX ENABLE & NORMAL
1671                          ; OPERATION IN LNCTRL.
1672 007034 000403          BR       54$
1673
1674 007036 012762 000226 000010      52$:  MOV      @LOCAL.LOOP!RX.ENABLE!AUTO1!AUTO0, LNCTRL(R2)
1675                          ; SET RX ENABLE & LOCAL
1676                          ; LOOPBACK IN LNCTRL.
1677
1678
1679 007044 016762 173626 000012      54$:  MOV      PA22, TX1BUFADR(R2)          ; PUT XMIT BUF ADDRESS IN TBUFAD1.
1680                          ; IF EXTENDED ADDRESS, MUST SET UP +14.
1681
1682 007052 116762 173622 000014      MOV      EA22, TX2BUFADR(R2)
1683 007060 012762 000154 000016      MOV      @108., TXBUFCNTR(R2)          ; SET UP BYTE COUNT.
1684
1685
1686
1687                          ; SETUP THE BAUD RATES.
1688
1689
1690 007066 016762 173624 000004      MOV      LPRSELECT, LPR(R2)          ; SETUP THE BAUD RATES.
1691
1692
1693
1694                          ; TURN ON THE TRANSMITTER
1695                          ; NOTE THE INTERRUPTS ARE NOT ENABLED AT THIS POINT BUT THE AUTO FLOW
1696                          ; WILL STOP TX WHEN THE FIFO STARTS TO GET FULL SO DATA WILL NOT BE LOST
1697
1698 007074 052762 100200 000014      BIS      @TX.ENABLE!DMA.START, TX2BUFADR(R2)
1699                          ; ENABLE DMA START AND TXMIT ENABLE.
1700
1701 007102 005200      70$:  INC      R0          ; UPDATE LINE NUMBER FOR CSR.
1702 007104 006301      ASL      R1          ; POINT TO NEXT LINE.
1703 007106 001337      BNE      50$          ; HAVE WE DONE 16 LINES?
1704
1705
1706
1707

```



```

1709 ;) **
1710 ;)
1711 ;) NAME: DHU.N2
1712 ;)
1713 ;) FUNCTIONAL DESCRIPTION: SET UP POINTERS FOR THE NEXT DHU11
1714 ;) AND INCREMENT DHUNOW
1715 ;) WHEN ALL BOARDS SET TURN ON INTERRUPT ENABLES
1716 ;)
1717 ;) CALLING SEQUENCE: IN LINE.
1718 ;)
1719 ;) INPUTS REQUIRED: "POINT" , "DHUCSR" , "DHUVECT"
1720 ;)
1721 ;) OUTPUTS GIVEN: "POINT" , "DHUNOW" , "DHUCSR" , "DHUVECT"
1722 ;)
1723 ;) REGISTERS USED,UNRESTORED: R1 , 3 , 4
1724 ;)
1725 ;) SUBROUTINES USED: NONE.
1726 ;)
1727 ;) RESTRICTIONS: NONE.
1728 ;)
1729 ;) --
1730
1731 007110 DHU.N2:
1732
1733 007110 062767 000020 171130 ADD #20,DHUCSR ; NEXT DHU11 ADDRESS.
1734 007116 062767 000010 173542 ADD #10,DHUVECT ; NEXT VECTOR ADDRESS.
1735 007124 005267 171114 INC DHUNOW ; UPDATE FOR ERROR TYPECOUNT IF ANY.
1736
1737 007130 106367 171114 ASLB POINT ; LOOK TO NEXT DHU11 BITMASK POSITION.
1738 007134 032767 000020 171106 BIT #20,POINT ; HAVE WE TESTED ALL DHU11'S.
1739 007142 001653 BEQ DHU.L2 ; NO GO BACK AND TEST IF ANY OTHERS SELECTED
1740
1741 ; YES ALL BOARDS SET SO TURN ON THE TRANSMITTER
1742 ; AND RECEIVER INTERRUPT ENABLES
1743
1744 007144 032767 000001 171070 BIT #1,DHU.LIVE ; CHECK FOR DHU11
1745 007152 001405 BEQ 10# ; IF NOT TRY THE NEXT ONE
1746 007154 016702 173522 MOV ADDR0,R2 ; SET UP R2 FOR FIRST CSR
1747
1748 007160 012762 040100 000000 ; MOV #TXINTR.ENABLE!RCVINTR.ENABLE,CSR(R2)
1749 ; SET XMIT-INTERPT-ENABLE AND
1750 ; RECEIVE DATA INTERRUPT ENABLE.
1751 007166 10#:
1752 007166 032767 000002 171046 BIT #2,DHU.LIVE ; CHECK FOR DHU11
1753 007174 001405 BEQ 20# ; IF NOT TRY THE NEXT ONE
1754 007176 016702 173502 MOV ADDR1,R2 ; SET UP R2 FOR SECOUND CSR
1755
1756 007202 012762 040100 000000 ; MOV #TXINTR.ENABLE!RCVINTR.ENABLE,CSR(R2)
1757 ; SET XMIT-INTERPT-ENABLE AND
1758 ; RECEIVE DATA INTERRUPT ENABLE.
1759 007210 20#:
1760 007210 032767 000004 171024 BIT #4,DHU.LIVE ; CHECK FOR DHU11
1761 007216 001405 BEQ 30# ; IF NOT TRY THE NEXT ONE
1762 007220 016702 173462 MOV ADDR2,R2 ; SET UP R2 FOR THIRD CSR
1763
1764 007224 012762 040100 000000 ; MOV #TXINTR.ENABLE!RCVINTR.ENABLE,CSR(R2)
1765 ; SET XMIT-INTERPT-ENABLE AND

```



```

1766                                     ; RECEIVE DATA INTERRUPT ENABLE.
1767 007232                               30$:
1768 007232 032767 000010 171002          BIT    #10,DHU.LIVE          ; CHECK FOR DHU11
1769 007240 001405                               BEQ    40$                ; IF NOT TRY THE NEXT ONE
1770 007242 016702 173442                  MOV    ADDR3,R2          ; SET UP R2 FOR FORTH CSR
1771                                     ;
1772 007246 012762 040100 000000          ;      MOV    #TXINTR.ENABLE!RCVINTR.ENABLE,CSR(R2)
1773                                     ;      ; SET XMIT-INTERPT-ENABLE AND
1774                                     ;      ; RECEIVE DATA INTERRUPT ENABLE.
1775 007254                               40$:
1776                                     ;
1777                                     ; AND NOW WE WAIT FOR THOSE INTERRUPTS
1778                                     ;
1779 007254 104400 000000'                  ;      EXIT$,BEGIN          ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
1780
1781
1782
1783
1784
1785                                     .SBTTL          DHU11 XMITTER INTERRUPT SERVICE ROUTINES.

```

```

1787 ;) **
1788 ;)
1789 ;) NAME: DHX0-X3,DHXSrv
1790 ;)
1791 ;) FUNCTIONAL DESCRIPTION: PROCESS THE INTERRUPTS OF XMITTERS.
1792 ;) CHECK FOR VALID INTERRUPT AND REPORT
1793 ;) ERROR AS REQUIRED.
1794 ;)
1795 ;) CALLING SEQUENCE: THESE ROUTINES ARE INTERRUPTED INTO.
1796 ;)
1797 ;) INPUTS REQUIRED: ADDRESS OF ROUTINES MUST BE SET UP.
1798 ;) "ADDR0" , 1 , 2 , 3 POINT TO DHU11 CSR'S
1799 ;)
1800 ;) OUTPUTS GIVEN: NONE
1801 ;)
1802 ;) REGISTERS USED,UNRESTORED: R1 , 2 ALL RESTORED
1803 ;)
1804 ;) SUBROUTINE ROUTINES USED: NONE.
1805 ;)
1806 ;) RESTRICTIONS: NONE.
1807 ;)
1808 ;)
1809 ;) --

```

; ENTRANCE POINT FOR DHU11'S DURING TRANSMITTER INTERRUPTS.

```

1814 007260          DHX0:
1815 007260 010246      MOV     R2,-(SP)           ; SAVE THE REGISTERS.
1816 007262 010146      MOV     R1,-(SP)           ;
1817 007264 012701 000766' MOV     @DHR0BUF+100,R1      ; SAVE BUFFER AREA FOR ERROR REPORTS.
1818
1819 007270 016702 173406 MOV     ADDR0,R2           ; CSR ADDRESS FOR DHU11 #1.
1820 007274 000167 000060 JMP     TXSERV             ; GOTO COMMON TX INTERRUPT ROUTINE.
1821
1822 007300          DHX1:
1823 007300 010246      MOV     R2,-(SP)           ; SAVE THE REGISTERS.
1824 007302 010146      MOV     R1,-(SP)           ;
1825 007304 012701 001366' MOV     @DHR1BUF+100,R1      ; SAVE BUFFER AREA FOR ERROR REPORTS.
1826
1827 007310 016702 173370 MOV     ADDR1,R2           ; CSR ADDRESS FOR DHU11 #2.
1828 007314 000167 000040 JMP     TXSERV             ; GOTO COMMON TX INTERRUPT ROUTINE.
1829
1830 007320          DHX2:
1831 007320 010246      MOV     R2,-(SP)           ; SAVE REGISTERS.
1832 007322 010146      MOV     R1,-(SP)           ;
1833 007324 012701 001766' MOV     @DHR2BUF+100,R1      ; SAVE BUFFER AREA FOR ERROR REPORTS.
1834
1835 007330 016702 173352 MOV     ADDR2,R2           ; CSR ADDRESS FOR DHU11 #3.
1836 007334 000167 000020 JMP     TXSERV             ; GOTO COMMON TX INTERRUPT ROUTINE.
1837
1838
1839
1840
1841 007340          DHX3:
1842 007340 010246      MOV     R2,-(SP)           ; SAVE REGISTERS.
1843 007342 010146      MOV     R1,-(SP)           ;

```

```

1844 007344 012701 002366'      MOV      @DHR3BUF+100,R1      ; SAVE BUFFER AREA FOR ERROR REPORTS.
1845
1846 007350 016702 173334      MOV      ADDR3,R2          ; CSR ADDRESS FOR DHU11 #4.
1847 007354 000167 000000      JMP      TXSERV            ; GOTO COMMON TX INTERRUPT ROUTINE.
1848
1849
1850
1851      ; COMMON TRANSMITTER SERVICE ROUTINE.
1852
1853
1854 007360      TXSERV:
1855
1856 007360 016267 000000 170670      MOV      CSR(R2),CDHUCSR    ; GET CSR.
1857 007366 100474      BMI     100$              ; REPORT ERROR IF TRANSMITTER ACTION BIT (15) NOT S
ET.
1858
1859
1860
1861      ; ILLEGAL TRANSMIT INTERRUPT!! ERROR!!
1862
1863
1864
1865 007370 010221      MOV      R2,(R1)+          ; REPORT "ILLEGAL TRANSMIT INTERRUPT ERROR."
1866 007372 016721 170660      MOV      CDHUCSR,(R1)+     ; STORE CSR REPORT.
1867 007376 016221 000002      MOV      RBUF(R2),(R1)+    ; STORE CSR CONTENTS.
1868
1869 007402 012762 000000 000000      MOV      #0,CSR(R2)        ; STORE RBUF CONTENTS.
1870
1871
1872 007410 012601      MOV      (SP)+,R1          ; DISABLE TRANSMIT &
1873 007412 012602      MOV      (SP)+,R2          ; RECEIVER INTERRUPTS.
1874
1874      ;-----
007414 000004 000000' 007422'      PIRQ$,BEGIN,10$           ; RESTORE REGISTERS.
;-----
1875
1876 007422      10$:
1877 007422 005767 170632      TST      LCKOUT            ; DO NOT TYPEOUT ERROR REPORT IF ANOTHER REPORT
1878 007426 001405      BEQ      15$              ; IS IN PROGRESS.
1879 007430 104407 000000'      BREAK$,BEGIN              ; TEMPORARY RETURN TO MONITOR....
007434 104407 000000'      BREAK$,BEGIN              ; THEN CONTINUE AT NEXT INSTRUCTION.
1880 007440 000770      BR       10$
1881 007442      15$:
1882 007442 005267 170612      INC      LCKOUT            ; DURING ERROR TYPEOUT, WE MUST NOT ALLOW ANY OTHER
1883 007446 012704 000766'      MOV      @DHR0BUF+100,R4   ; SECTION TO REPORT AN ERROR.
1884 007452      20$:
1885 007452 005714      TST      (R4)              ; DOES THIS LINE HAVE A SAVED ERROR? (R4<>0).
1886 007454 001003      BNE     30$              ; YES, REPORT IT.
1887 007456 062704 000400      ADD     @DHR1BUF-DHR0BUF,R4 ; NO, THEN LOOK AT NEXT LINE FOR ERROR.
1888 007462 000773      BR      20$
1889
1890
1891
1892
1893 007464      30$:
1894 007464 011467 170410      MOV      (R4),CSRA         ; REPORT CSR ADDRESS.
1895 007470 005024      CLR     (R4)+             ; CLEAR ERROR INDICATION.
1896
1897 007472 012467 170404      MOV      (R4)+,ACSR        ; REPORT CSR CONTENTS.

```

```

1898 007476 012467 170402      MOV    (R4)+,ASTAT      ; REPORT RBUF CONTENTS.
1899 007502 012767 000025 170376  MOV    @25,ERRTYP      ; REPORT ERROR TYPE.
1900
1901 007510 104401 000000' 003625'  MSG$,BEGIN,ERR.H      ;ASCII MESSAGE CALL
1902      007516 104405 000000' 000000  ;*****
;HRDR$,BEGIN,NULL      ;
;*****
1903
1904 007524 005067 170530      CLR    LCKOUT          ; ALLOW ANOTHER DHU11 TO RUN THIS DIAGNOSTIC.
1905 007530 005267 170502      INC    DHUSDONE        ; ANOTHER DHU11 IS DONE.
1906 007534 026767 170476 170476  CMP    DHUSDONE,DHUSLEFT ; ANY DHU11'S LEFT TO TEST?
1907 007542 001402      BEQ    35$            ; YES. GO DO THEM.
1908
1909 007544 104400 000000'      EXIT$,BEGIN          ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
1910
1911 007550      35$:
1912 007550 104413 000000'      ENDIT$,BEGIN        ;SIGNAL END OF ITERATION.
;MONITOR SHALL TEST END OF PASS
; GO START TESTING ANOTHER DHU11.
1913 007554 000167 174614      JMP    RESTRT
1914
1915 007560      100$:
1916 007560 012601      MOV    (SP)+,R1        ; RESTORE REGISTERS.
1917 007562 012602      MOV    (SP)+,R2
1918 007564 000002      RTI
1919
1920
1921
1922
1923      .SBTTL          DHU11 RECEIVER INTERRUPT SERVICE ROUTINES.

```



```

1925 ;) **
1926 ;)
1927 ;) NAME: DHR0-R3,DHRSRV
1928 ;)
1929 ;) FUNCTIONAL DESCRIPTION: PROCESS THE INTERRUPTS OF RECEIVERS.
1930 ;) CHECK DATA RECIEVED AND REPORT ANY ERRORS
1931 ;) ON GETTING A LAST CHARACTER ( 377 )
1932 ;) SET THE APPROPRIATE BIT IN LINESDONE
1933 ;) TEST FOR ALL BOARDS FINISHED AND IF
1934 ;) SO DROP TO NORMAL PRIORITY AND DECLARE
1935 ;) AN END OF ITERATION. RETURN TO RESTART
1936 ;) POINT.
1937 ;)
1938 ;)
1939 ;) CALLING SEQUENCE: THESE ROUTINES ARE INTERRUPTED INTO.
1940 ;)
1941 ;) INPUTS REQUIRED: ADDRESS OF ROUTINES MUST BE SET UP.
1942 ;) "ADDRO" , 1 , 2 AND 3 CONTAIN CSR ADDRESSES
1943 ;) "LINESELECT" CONTAINS BIT MAP OF SELECTED LINES
1944 ;) "DHUSNOW" CONTAINS NUMBER OF DMU11'S SELECTED
1945 ;)
1946 ;) OUTPUTS GIVEN: "LINESDONE" AND "DHUSDONE" ARE UPDATED
1947 ;)
1948 ;) REGISTERS USED,UNRESTORED: R1 , 2 , 3 , 4 ALL RESTORED
1949 ;)
1950 ;) SUBROUTINE ROUTINES USED: NONE.
1951 ;)
1952 ;) RESTRICTIONS: NONE.
1953 ;)
1954 ;) --
1955 ;)
1956 ;)
1957 ;)
1958 ;)
1959 ;)

```

; ENTRANCE POINT FOR DMU11'S DURING RECEIVER INTERRUPTS.

```

1960 007566 010146 DHR0: MOV R1,-(SP) ; SAVE REGISTERS ON STACK.
1961 007566 010246 MOV R2,-(SP)
1962 007570 010346 MOV R3,-(SP)
1963 007572 010446 MOV R4,-(SP)
1964 007574 010446
1965
1966 007576 012767 000000 170446 MOV #0,OFFSET ; INDICATE THIS IS DEVICE #1.
1967 007604 012767 000666' 170442 MOV #DHR0BUF,BUFADR ; GET BUFFER TO RECEIVE CHARACTERS.
1968 007612 016702 173064 MOV ADDRO,R2 ; CSR ADDRESS FOR DMU11 #1.
1969 007616 000167 000124 JMP RXSERV ; GOTO COMMON RECEIVER INTERRUPT SERVICE ROUTINE.
1970
1971 007622 DHR1: MOV R1,-(SP) ; SAVE REGISTERS ON STACK.
1972 007622 010146 MOV R2,-(SP)
1973 007624 010246 MOV R3,-(SP)
1974 007626 010346 MOV R4,-(SP)
1975 007630 010446
1976
1977 007632 012767 000002 170412 MOV #2,OFFSET ; INDICATE THIS IS DEVICE #2.
1978 007640 012767 001266' 170406 MOV #DHR1BUF,BUFADR ; GET BUFFER TO RECEIVE CHARACTERS.
1979 007646 016702 173032 MOV ADDR1,R2 ; CSR ADDRESS FOR DMU11 #2.
1980 007652 000167 000070 JMP RXSERV ; GOTO COMMON RECEIVER INTERRUPT SERVICE ROUTINE.
1981

```

```

1982
1983
1984
1985
1986 007656          DHR2:
1987 007656 010146   MOV      R1,-(SP)          ; SAVE REGISTERS ON STACK.
1988 007660 010246   MOV      R2,-(SP)
1989 007662 010346   MOV      R3,-(SP)
1990 007664 010446   MOV      R4,-(SP)
1991
1992 007666 012767 000004 170356   MOV      #4,OFFSET        ; INDICATE THIS IS DEVICE #3.
1993 007674 012767 001666' 170352   MOV      #DHR2BUF,BUFADR  ; GET BUFFER TO RECEIVE CHARACTERS.
1994 007702 016702 173000          MOV      ADDR2,R2         ; CSR ADDRESS FOR DHU11 #3.
1995 007706 000167 000034          JMP      RXSERV          ; GOTO COMMON RECEIVER INTERRUPT SERVICE ROUTINE.
1996
1997 007712          DHR3:
1998 007712 010146   MOV      R1,-(SP)          ; SAVE REGISTERS ON STACK.
1999 007714 010246   MOV      R2,-(SP)
2000 007716 010346   MOV      R3,-(SP)
2001 007720 010446   MOV      R4,-(SP)
2002
2003 007722 012767 000006 170322   MOV      #6,OFFSET        ; INDICATE THIS IS DEVICE #4.
2004 007730 012767 002266' 170316   MOV      #DHR3BUF,BUFADR  ; GET BUFFER TO RECEIVE CHARACTERS.
2005 007736 016702 172746          MOV      ADDR3,R2         ; CSR ADDRESS FOR DHU11 #4.
2006 007742 000167 000000          JMP      RXSERV          ; GOTO COMMON RECEIVER INTERRUPT SERVICE ROUTINE.
2007
2008
2009
2010          ; COMMON RECEIVER INTERRUPT SERVICE ROUTINE.
2011
2012
2013 007746          RXSERV:
2014
2015 007746          20$:
2016 007746 016201 000002   MOV      RBUF(R2),R1      ; PULL OUT RECEIVED CHARACTER.
2017 007752 100402          BMI      25$             ; ARE THERE ANY CHARACTERS?
2018 007754 000167 000576   JMP      100$            ; THERE ARE NO MORE CHARACTERS TO PROCESS.
2019 007760          25$:
2020 007760 032701 070000   BIT      #OVERRUN.ERROR!FRAMING.ERROR!PARITY.ERROR,R1
2021          ; YES. ARE THERE
2022          ; ANY ERRORS?
2023 007764 001510          BEQ      40$             ; NO.
2024
2025 007766 010103          MOV      R1,R3           ; THERE COULD BE AN ERROR.
2026 007770 062703 010000   ADD      #BIT12,R3        ; ARE BITS 12,13, & 14 SET? WE HOPE SO.
2027 007774 103764          BCS      20$            ; YES THEY ARE. NO PROBLEM.
2028
2029
2030
2031
2032
2033          ; ERROR!! OVERRUN,PARITY, OR FRAMING ERROR!!
2034
2035
2036 007776 016704 170252   MOV      BUFADR,R4        ; GET BUFFER TO STORE ERROR REPORT.
2037 010002 062704 000040   ADD      #40,R4
2038

```

```

2039 010006 010224          MOV      R2,(R4)+          ; STORE THE CSR.
2040 010010 016224 000000  MOV      CSR(R2),(R4)+    ; STORE THE CSR CONTENTS.
2041 010014 010124          MOV      R1,(R4)+          ; STORE THE RECEIVE BUFFER CONTENTS.
2042
2043 010016 012762 000000 000000  MOV      #0,CSR(R2)      ; DISABLE TX & RECEIVE
2044                                     ; INTERRUPTS.
2045
2046 010024 012604          MOV      (SP)+,R4         ; RESTORE THE REGISTERS.
2047 010026 012603          MOV      (SP)+,R3
2048 010030 012602          MOV      (SP)+,R2
2049 010032 012601          MOV      (SP)+,R1
2050
2051      010034 000004 000000' 010042'  ;-----
PIRQ$,BEGIN,26$          ; QUEUE UP TO CONTINUE AT 26$ AND RTI
;-----

2052
2053 010042          26$:
2054 010042 005767 170212      TST      LCKOUT          ; DO NOT TYPEOUT ERROR REPORT IF ANOTHER ERROR
2055 010046 001405          BEQ      27$             ; REPORT IS IN PROGRESS.
2056 010050 104407 000000'    BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
2057 010054 104407 000000'    BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
2058 010060 000770          BR       26$
2059 010062          27$:
2060 010062 005267 170172      INC      LCKOUT          ; DURING ERROR TYPEOUT, WE MUST NOT ALLOW ANY OTHER
2061 010066 012704 000726'    MOV      #DHR0BUF+40,R4 ; SECTION TO REPORT AN ERROR.
2062 010072          28$:
2063 010072 005714          TST      (R4)           ; DOES THIS LINE HAVE A SAVED ERROR? (R1<>0).
2064 010074 001003          BNE     29$             ; YES, REPORT IT.
2065 010076 062704 000400      ADD     #DHR1BUF-DHR0BUF,R4 ; NO, THEN LOOK AT NEXT LINE FOR ERROR.
2066 010102 000773          BR       28$
2067
2068 010104          29$:
2069 010104 011467 167770      MOV      (R4),CSRA      ; REPORT THE CSR ADDRESS.
2070 010110 005024          CLR      (R4)+          ; CLEAR ERROR INDICATION.
2071
2072 010112 012467 167764      MOV      (R4)+,ACSR     ; REPORT THE CSR CONTENTS.
2073 010116 012467 167762      MOV      (R4)+,ASTAT    ; REPORT THE CONTENTS OF RBUF.
2074
2075 010122 012767 000034 167756  MOV      #34,ERRTYP     ; REPORT ERROR TYPE.
2076 010130 104401 000000' 003675'  MSG$,BEGIN,ERR.I      ;ASCII MESSAGE CALL
2077 010136          30$:          ; ERROR OCCURRED.
2078      010136 104405 000000' 000000  ;*****
HRDR$,BEGIN,NULL      ;
;*****

2079
2080 010144          32$:
2081 010144 005067 170110      CLR      LCKOUT
2082 010150 005267 170062      INC      DHUSDONE
2083 010154 026767 170056 170056  CMP      DHUSDONE,DHUSLEFT ; THIS DHU11 IS DONE!
2084 010162 001405          BEQ      36$             ; ANY MORE DHU11'S TO DO?
2085
2086 010164          33$:
2087 010164 104400 000000'    EXIT$,BEGIN          ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
2088
2089
2090

```



```

2091
2092
2093 010170          35$:
2094 010170 000004 000000' 010176'  ;-----;
;PIRQ$,BEGIN,36$          ; QUEUE UP TO CONTINUE AT 36$ AND RTI
;-----;
2095 010176          36$:
2096 010176 104413 000000'          ;SIGNAL END OF ITERATION.
;MONITOR SHALL TEST END OF PASS
2097
2098 010202 000167 174166          JMP      RESTRT
2099
2100 010206          40$:
2101 010206 010103          MOV      R1,R3          ; GET LINE #.
2102 010210 000303          SWAB     R3           ; PLACE LINE # IN 1ST FOUR BITS.
2103 010212 042703 177760          BIC     #+C<17>,R3     ; CLEAR OUT EVERYTHING BUT THE LINE #.
2104
2105 010216 066703 170032          ADD     BUFADR,R3     ; SET-UP LOCATION FOR LINE #.
2106
2107 010222          60$:
2108 010222 042701 177400          BIC     #+C<377>,R1    ; CLEAR OUT EVERYTHING BUT THE CHARACTER RECEIVED.
2109 010226 120113          CMPB    R1,(R3)        ; DOES THE EXPECTED (R3) MATCH THE RECEIVED R1.
2110 010230 001504          BEQ     70$           ; BRANCH IF EQUAL. THIS IS WHAT WE WANT.
2111
2112 010232 020127 000021          CMP     R1,#21        ; XON? IF SO, WE MUST IGNORE.
2113 010236 001643          BEQ     20$
2114
2115 010240 020127 000023          CMP     R1,#23        ; XOFF? IF SO, WE MUST IGNORE. IF NOT, REPORT
2116 010244 001640          BEQ     20$          ; A DATA ERROR. IF CHARACTER DOES NOT MATCH
2117
2118
2119
2120
2121          ; ERROR!! DATA ERROR!!
2122
2123
2124 010246          62$:
2125 010246 016704 170002          MOV     BUFADR,R4
2126 010252 062704 000020          ADD     #20,R4
2127
2128 010256 010124          MOV     R1,(R4)+      ; ACTUAL DATA.
2129 010260 011324          MOV     (R3),(R4)+    ; EXPECTED DATA.
2130 010262 010224          MOV     R2,(R4)+      ; CSR ADDRESS.
2131 010264 010314          MOV     R3,(R4)       ; LINE # + BUFADR.
2132 010266 166724 167762          SUB     BUFADR,(R4)+  ; LINE #.
2133
2134 010272 012762 000000 000000    MOV     #0,CSR(R2)    ; DISABLE TX & RECEIVE
2135
2136
2137 010300 012604          MOV     (SP)+,R4      ; RESTORE REGISTERS.
2138 010302 012603          MOV     (SP)+,R3
2139 010304 012602          MOV     (SP)+,R2
2140 010306 012601          MOV     (SP)+,R1
2141
2142 010310 000004 000000' 010316'  ;-----;
;PIRQ$,BEGIN,66$          ; QUEUE UP TO CONTINUE AT 66$ AND RTI
;-----;

```



```

2143 010316          66$:
2144 010316 005767 167736      TST      LCKOUT          ; DON'T TYPEOUT ERROR REPORT IF ANOTHER ERROR
2145 010322 001405          BEQ      67$            ; REPORT IS IN PROGRESS.
2146 010324 104407 000000'    BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR,...
2146 010330 104407 000000'    BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
2147 010334 000770          BR       66$
2148
2149
2150
2151
2152 010336          67$:
2153 010336 005267 167716      INC      LCKOUT          ; DURING ERROR TYPEOUT, WE MUST NOT ALLOW ANY OTHER
2154 010342 012701 000706'    MOV     #DHR0BUF+20,R1 ; SECTION TO REPORT AN ERROR.
2155
2156 010346          68$:
2157 010346 005711          TST     (R1)            ; DOES THIS LINE HAVE A SAVED ERROR? (R1<>0)
2158 010350 001003          BNE     69$            ; YES, REPORT IT.
2159 010352 062701 000400      ADD     #DHR1BUF-DHROBUF,R1 ; NO, THEN LOOK AT NEXT LINE FOR ERROR.
2160 010356 000773          BR      68$
2161
2162
2163 010360          69$:
2164 010360 011167 167524      MOV     (R1),AWAS      ; REPORT ACTUAL DATA.
2165 010364 005021          CLR     (P1)+          ; CLEAR ERROR INDICATION.
2166
2167 010366 112167 167514      MOVB   (R1)+,ASB      ; REPORT EXPECTED DATA.
2168 010372 105021          CLRB   (R1)+
2169 010374 012167 167500      MOV     (R1)+,CSRA    ; REPORT CSR ADDRESS.
2170 010400 012167 167476      MOV     (R1)+,SBADR   ; REPORT LINE #.
2171
2172 010404 104401 000000' 003753' MSG$,BEGIN,ERR.J      ;ASCII MESSAGE CALL
2173
2174          ;*****
          010412 104404 000000' DATER$,BEGIN          ;DATA ERROR!!!
          ;*****
2175
2176 010416 005067 167636      CLR     LCKOUT          ; DONE REPORTING THIS ERROR?
2177 010422 005267 167610      INC     DHUSDONE        ; MARK THIS DHU11 AS BEING DONE.
2178 010426 026767 167604 167604 CMP     DHUSDONE,DHUSLEFT ; ARE ALL THE DHU11'S DONE?
2179 010434 001660          BEQ     36$            ; YES, THEN END PASS. ELSE NORMAL EXIT.
2180
2181 010436 104400 000000'    EXIT$,BEGIN          ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
2182
2183 010442          70$:
2184 010442 105213          INCB   (R3)            ; INCREMENT CHARACTER.
2185 010444 001402          BEQ     75$            ; SEE IF DID LAST CHARACTER (377), BR IF NOT.
2186 010446 000167 177274      JMP     20$            ; NOT THE LAST CHARACTER. GO SEE IF ANOTHER
2187 010452          75$:
2188 010452 166703 167576      SUB    BUFADR,R3      ; CHARACTER IS AVAILABLE.
2189 010456 012704 000001      MOV     #1,R4         ; R3 = LINE# + BUFFER ADDRESS. GET LINE#.
2190 010462          80$:
2191 010462 005303          DEC    R3              ; MAKE BITMASK OF LINE NUMBER.
2192 010464 100402          BMI    90$            ; DECREMENT UNTIL 0.
2193 010466 006304          ASL    R4              ; SHIFT UNTIL # GOES TO ZERO FOR BITMASK.
2194 010470 000774          BR     80$            ; BITMASK IN R4.
2195 010472          90$:
2196 010472 016701 167554      MOV    OFFSET,R1

```

```

2197 010476 050461 000224'      BIS      R4,LINESDONE(R1)      ; RECORD THIS LINE DONE.
2198
2199 010502 026761 167514 000224'  CMP      LINESELECT,LINESDONE(R1) ; SEE IF ALL LINES DONE ON THIS UNIT.
2200
2201 010510 001402                BEQ      95$
2202 010512 000167 177230          JMP      20$
2203
2204
2205 010516                95$:
2206 010516 012762 000000 000000      MOV      #0,CSR(R2)          ; DISABLE TX & RECEIVE
2207                                ; INTERRUPTS.
2208
2209 010524 012604                MOV      (SP),R4            ; RESTORE REGISTERS.
2210 010526 012603                MOV      (SP),R3
2211 010530 012602                MOV      (SP),R2
2212 010532 012601                MOV      (SP),R1
2213
2214 010534 005267 167476          INC      DHUSDONE           ; RECORD THIS DHU11 AS BEING DONE.
2215 010540 026767 167472 167472    CMP      DHUSDONE,DHUSLEFT ; ARE ALL DHU11'S DONE?
2216 010546 001002                BNE     97$
2217 010550 000167 177414          JMP      35$
2218 010554                97$:
2219 010554 000002                RTI
2220                                ; EXIT, JUST THIS DHU11 IS DONE.
2221
2222                100$:
2223 010556 012604                MOV      (SP),R4            ; RESTORE REGISTERS.
2224 010560 012603                MOV      (SP),R3
2225 010562 012602                MOV      (SP),R2
2226 010564 012601                MOV      (SP),R1
2227 010566 000002                RTI
2228
2229
2230

```

```

2232 ;) **
2233 ;)
2234 ;) NAME: BAUDRATE
2235 ;)
2236 ;) FUNCTIONAL DESCRIPTION: SUBROUTINE TO SET UP THE BAUDRATE
2237 ;) AND STORE IT IN TXSPEED AND RCVSPEED
2238 ;)
2239 ;) CALLING SEQUENCE: JSR PC,BAUDRATE
2240 ;)
2241 ;) INPUTS REQUIRED: "SR1" AND "2" IN MODULE HEADER AND "DHULPR"
2242 ;)
2243 ;) OUTPUTS GIVEN: "TXSPEED", "RCVSPEED" AND "LPRSELECT"
2244 ;)
2245 ;) REGISTERS USED,UNRESTORED: R0
2246 ;)
2247 ;) SUBROUTINE ROUTINES USED: NONE.
2248 ;)
2249 ;) RESTRICTIONS: "DHULPR" IS SET UP WHEN DECLARED.
2250 ;) IT SHOULD NOT BE CHANGED.
2251 ;) --
2252 010570 BAUDRATE:
2253
2254 010570 016767 167222 167464 MOV SR1, TXSPEED ; GET SELECTED TRANSMIT SPEED.
2255 010576 016767 167214 167460 MOV SR1, RCVSPEED ; GET SELECTED RECEIVE SPEED.
2256
2257
2258 010604 012700 000004 MOV #4, R0 ; SET-UP SHIFT COUNTER.
2259 010610 100:
2260
2261 010610 006367 167446 ASL TXSPEED ; LINE-UP TRANSMIT SPEED SELECTOR W/LPR.
2262 010614 077003 SOB R0, 100
2263
2264 010616 056767 167440 167440 BIS TXSPEED, RCVSPEED ; PUT TRANSMIT & SPEED SELECTION TOGETHER.
2265 010624 000367 167434 SWAB RCVSPEED
2266
2267
2268 010630 016767 172060 172060 MOV DHULPR, LPRSELECT ; INITIALIZE THE LINE PARAMETER SELECTION.
2269 010636 056767 167422 172052 BIS RCVSPEED, LPRSELECT ; SAVE THE TRANSMIT & RECEIVE SPEEDS.
2270
2271 010644 000207 RTS PC ; RETURN TO MAIN PROGRAM.
2272
2273 000001 .END

```


ACSR	000102R	DATCK#	104411	ERR.C	003120R	OFFSET	000252R	SETUP1	004452R
ADDR	000006R	DATER#	104404	ERR.D	003210R	OPEN	= 000000	SETUP2	004516R
ADDR0	002702R	DHR0	007566R	ERR.E	003311R	OTOA#	= 104420	SOFCNT	000042R
ADDR1	002704R	DHR0BU	000666R	ERR.F	003404R	OVERRU	= 040000	SOFER#	= 104406
ADDR2	002706R	DHR1	007622R	ERR.G	003511R	PA	002672R	SOFPAS	000046R
ADDR22	= 001000	DHR1BU	001266R	ERR.H	003625R	PARITY	= 010000	SPOINT	000032R
ADDR3	002710R	DHR2	007656R	ERR.I	003675R	PASCNT	000034R	SPSIZ	= 000040
ASB	000106R	DHR2BU	001666R	ERR.J	003753R	PA22	002676R	SR1	000016R
ASTAT	000104R	DHR3	007712R	ERR.K	004003R	PIRQ#	= 000004	SR2	000020R
AUTO0	= 000020	DHR3BU	002266R	ERR.L	004042R	POINT	000250R	SR3	000022R
AUTO1	= 000002	DHUCSR	000246R	ERR.M	004150R	POPSP	= 005726	SR4	000024R
AMAS	000110R	DHULPR	002714R	EX	005334R	POPSP2	= 022626	START	004340R
BAUDRA	010570R	DHUNOW	000244R	EXIT#	= 104400	PRTY	= 000000	STAT	000026R
BEGIN	000000R	DHUSDO	000236R	FLAG	002712R	PRTY0	= 000000	SVR0	000062R
BIT0	= 000001	DHUSLE	000240R	FRAMIN	= 020000	PRTY1	= 000040	SVR1	000064R
BIT1	= 000002	DHUT01	004550R	GETPA#	= 104415	PRTY2	= 000100	SVR2	000066R
BIT10	= 002000	DHUT02	004730R	GMBUF#	= 104414	PRTY3	= 000140	SVR3	000070R
BIT11	= 004000	DHUT03	005764R	HRDCNT	000044R	PRTY4	= 000200	SVR4	000072R
BIT12	= 010000	DHUT04	006610R	HRDER#	= 104405	PRTY5	= 000240	SVR5	000074R
BIT13	= 020000	DHUEVC	002666R	HRDPAS	000050R	PRTY6	= 000300	SVR6	000076R
BIT14	= 040000	DHU.BU	006706R	ICONT	000036R	PRTY7	= 000340	SYSCNT	000052R
BIT15	= 100000	DHU.LI	000242R	ICOUNT	000040R	PS	= 177776	TRPDFD	= 000022
BIT2	= 000004	DHU.L1	006000R	IDNUM	000122R	PSW	= 177776	TX	005274R
BIT3	= 000010	DHU.L2	006672R	INIT	000030R	PUSH	= 005746	TXBUF	000266R
BIT4	= 000020	DHU.N1	006550R	INTR	000120R	PUSH2	= 024646	TXBUFC	= 000016
BIT5	= 000040	DHU.N2	007110R	LCKOUT	000260R	PWRFLG	= 000002	TXCHAR	= 000006
BIT6	= 000100	DHX0	007260R	LINECO	000234R	RAND#	= 104417	TXDATA	= 000224
BIT7	= 000200	DHX1	007300R	LINESD	000224R	RANUM	000054R	TXDMA	= 010000
BIT8	= 000400	DHX2	007320R	LINESE	000222R	RBUF	= 000002	TXINTR	= 040000
BIT9	= 001000	DHX3	007340R	LNCTRL	= 000010	RCV	005314R	TXSERV	007360R
BREAK#	= 104407	DIAG.F	= 020000	LOCAL	= 000200	RCVDAT	= 000200	TXSPEE	000262R
BR1	000012R	DMA.ST	= 000200	LPR	= 000004	RCVINT	= 000100	TX.ACT	= 100000
BR2	000013R	DROP	004350R	LPRSEL	002716R	RCVSPE	000264R	TX.ENA	= 100000
BTOD#	= 104421	DROP1	004354R	MAP22#	= 104416	RESTRT	004374R	TX1BUF	= 000012
BUFADR	000254R	DVID1	000014R	MASTER	= 000040	RES1	000056R	TX2BUF	= 000014
CDATA#	= 104412	EA	002674R	MODNAM	000000R	RES2	000060R	VA	002670R
CDHUCS	000256R	EA22	002700R	MODSP	000222R	RSTRT	000112R	VECT	005664R
CHECKL	005346R	ENDIT#	= 104413	MSG#	= 104403	RXSERV	007746R	VECTOR	000010R
CONFIG	000056R	END#	= 104410	MSG#	= 104402	RX.ENA	= 000004	WASADR	000104R
CSR	= 000000	ERRTYP	000106R	MSG#	= 104401	R6	=#000006	WDFR	000116R
CSRA	000100R	ERR.A	002720R	NEXTDM	005714R	R7	=#000007	WDT0	000114R
DATA.V	= 100000	ERR.B	003020R	NULL	= 000000	SBADR	000102R	XFLAG	000005R

. ABS. 000000 000
010646 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 12513 WORDS (49 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:00:31
CXDHUAO.OBJ,CXDHUAO.LST/-SP=DDXCOM.MAC,CXDHUAO.P11