Jameco ELECTRONICS

JE665 RS-232C COMPUTER INTERFACE
OPERATION MANUAL

JE665 RS-232C INTERFACE OPTION

1. INTRODUCTION

The JE665 option is a circuit-board assembly added to the JE664 Programmer. It implements exchange of data between the JE664's RAM and an external computer over the standardized RS-232C data link. The format used: 9600 baud rate, 8-bit data word, odd parity, two stop bits.

JAMECO provides software in "Micro-Soft Basic" written for the TRS-80 computer, Mod I, Level II. For other computers, the program must be modified appropriately by the user since port addresses, status-bit positions, protected memory area, etc. will likely not be the same. Also, the subroutine for saving and loading to tape or disk will differ. The program may be entered manually from the written text supplied.

With any computer of 16K or less memory capacity, the program's remarks should not be entered since they will require too much memory capacity.

2. COMMANDS USED

1) **SAVE TO CASSETTE.**
Loads data from memory locations 24100-32291 to cassette (if optional subroutine is used).

2) **LOAD FROM CASSETTE.**
Loads data from cassette into memory locations 24100-32291 (if optional subroutine is used).

3) **LOAD FROM PROGRAMMER.**
Loads data from the JE664's RAM into memory locations 24100-32291.

4) **SAVE TO PROGRAMMER.**
Loads data from memory locations 24100-32291 into JE664's RAM.

5) **MEMORY DUMP (HEX).**
Prints out the contents of memory locations 24100-32291 in hexadecimal.

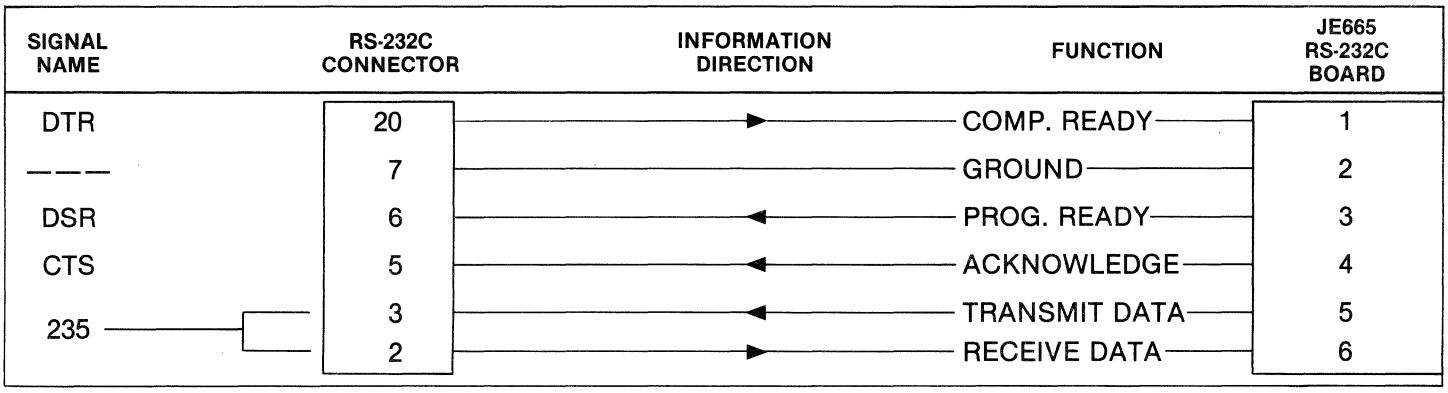
6) **MEMORY DUMP (ASCII).**
Prints out the contents of memory locations 24100-32291 in ASCII-equivalent characters.

7) **KEY.**
Enables data entry from the computer's keyboard. Data may be entered into the memory block 24100-32291 as well as edited at will.

8) **EXIT.**
Halts execution of the program.

3. DESCRIPTION OF OPERATION.

A. **The RS-232C Link.**
In the JE664, setting the pulse width/RS-232 switch to RS-232C enables the function (and disables the JE664's keyboard). The RS-232C interface functions are as follows:



The "235" signals, having but one "name" at the computer end, are identified by their function names at the JE664 end (i.e. Transmit and Receive).

RS-232C voltage levels are +8 to 12V = "High" = Logic "0" and -8 to -12V = "Low" = Logic "1." These levels are appropriately converted to 0V, +5V levels on the RS option board.

B. **FLOW CHARTS.**
Following are flow charts for the ***Load from Programmer*** and ***Save to Programmer*** operations. These should be referred to when modifying the supplied program for other than a TRS-80 computer.

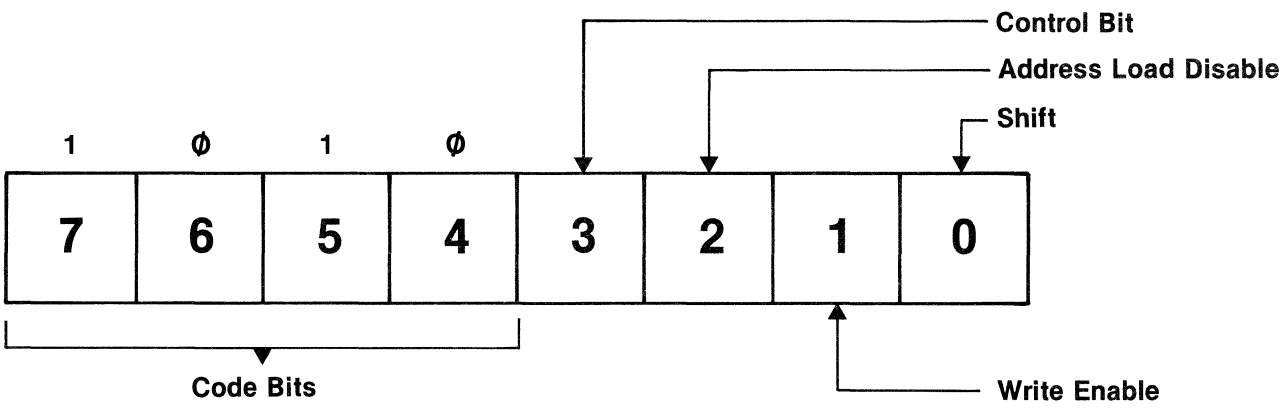
FLOW CHART NOTES

LOAD FROM PROGRAMMER CHART. *(See Charts)*

1. Control words are in hex notation with binary notation shown in parentheses.
2. A1 (1010 0001) sets SHIFT, disables WRITE, zeros upper 32K ADDRESS and signifies that another control word is next.
3. A8 (1010 1000) clears SHIFT, disables WRITE, zeros lower 32K ADDRESS and signifies that data will follow.
4. A0 (1010 0000) clears SHIFT, disables WRITE, zeros lower 32K ADDRESS and signifies that another control word is next.
5. A9 (1010 1001) sets SHIFT, disables WRITE, zeros upper 32K ADDRESS and signifies that data will follow.
6. AD (1010 1101) sets SHIFT, disables WRITE, disables ADDRESS clear and signifies that data will follow.

SAVE TO PROGRAMMER CHART. *(See Charts)*

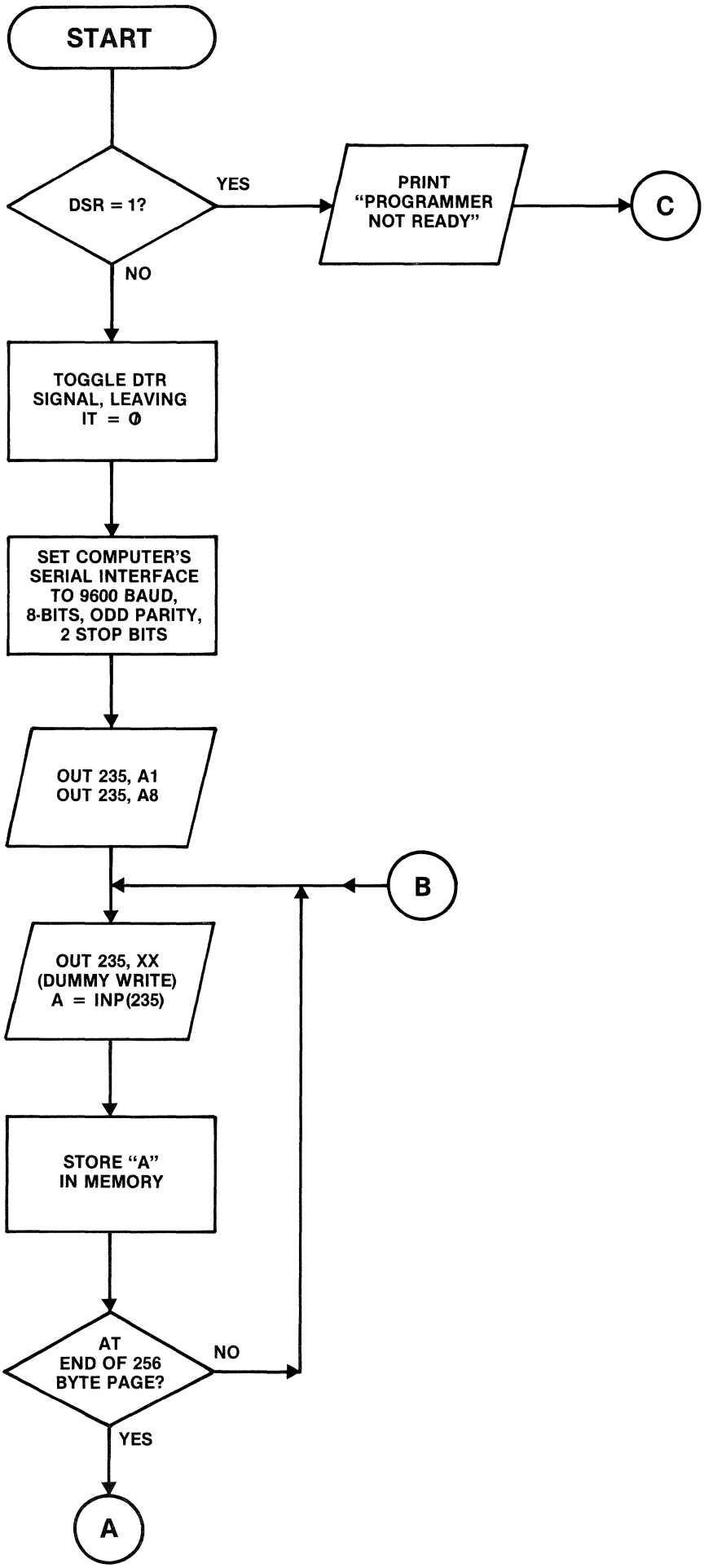
1. Control words are in hex notation with binary notation shown in parentheses.
2. A1 (1010 0001) sets SHIFT, disables WRITE, zeros upper 32K ADDRESS and signifies that another control word is next.
3. AA (1010 1010) clears SHIFT, enables WRITE, clears lower 32K ADDRESS and signifies that data will follow.
4. A0 (1010 0000) clears SHIFT, disables WRITE, zeros lower 32K ADDRESS and signifies that another control word is next.
5. AB (1010 1011) sets SHIFT, enables WRITE, zeros upper 32K ADDRESS and signifies that data will follow.
6. AF (1010 1111) sets SHIFT, enables WRITE, disables ADDRESS clear and signifies that data will follow.



CONTROL WORD

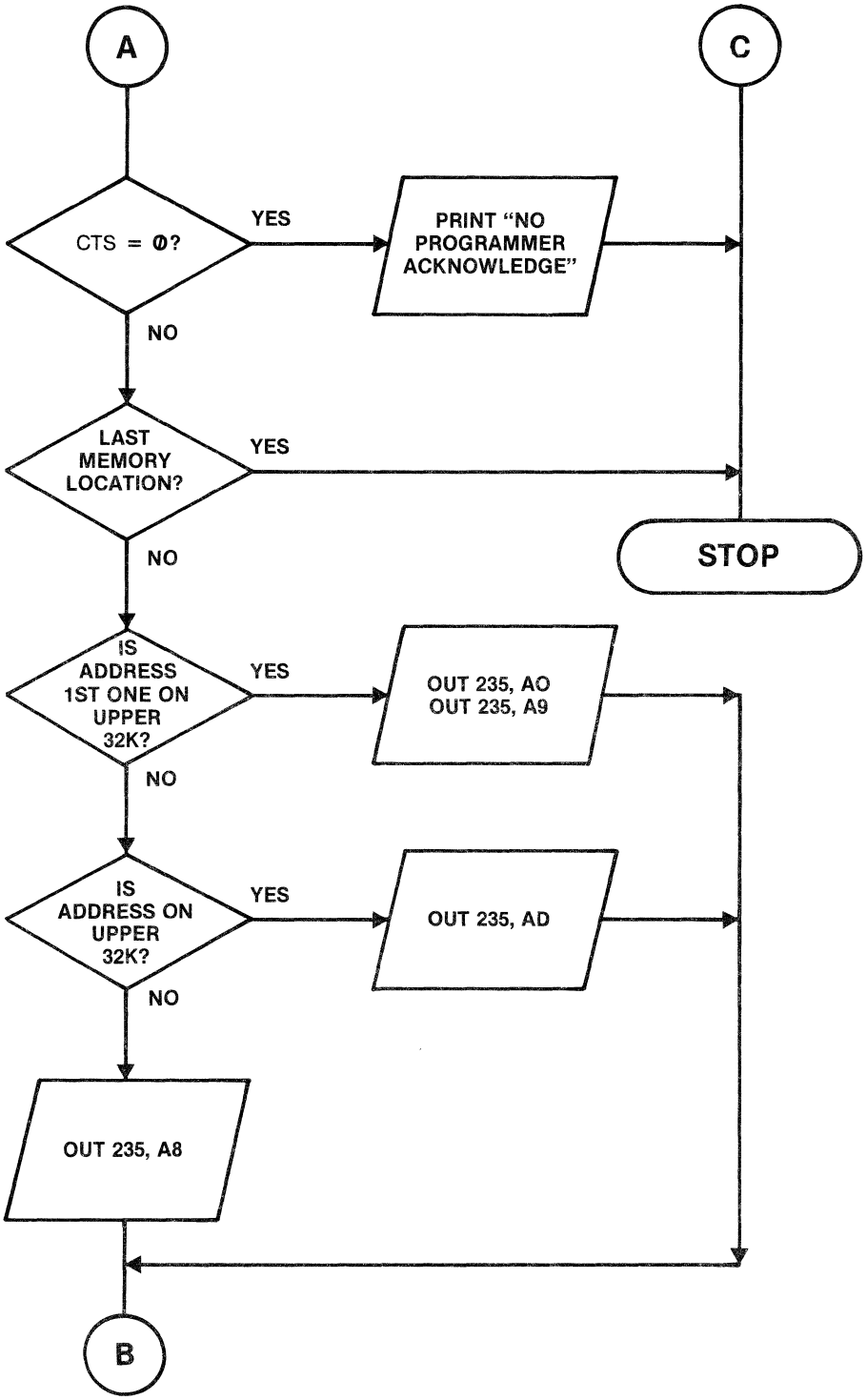
LOAD FROM PROGRAMMER

1 OF 2



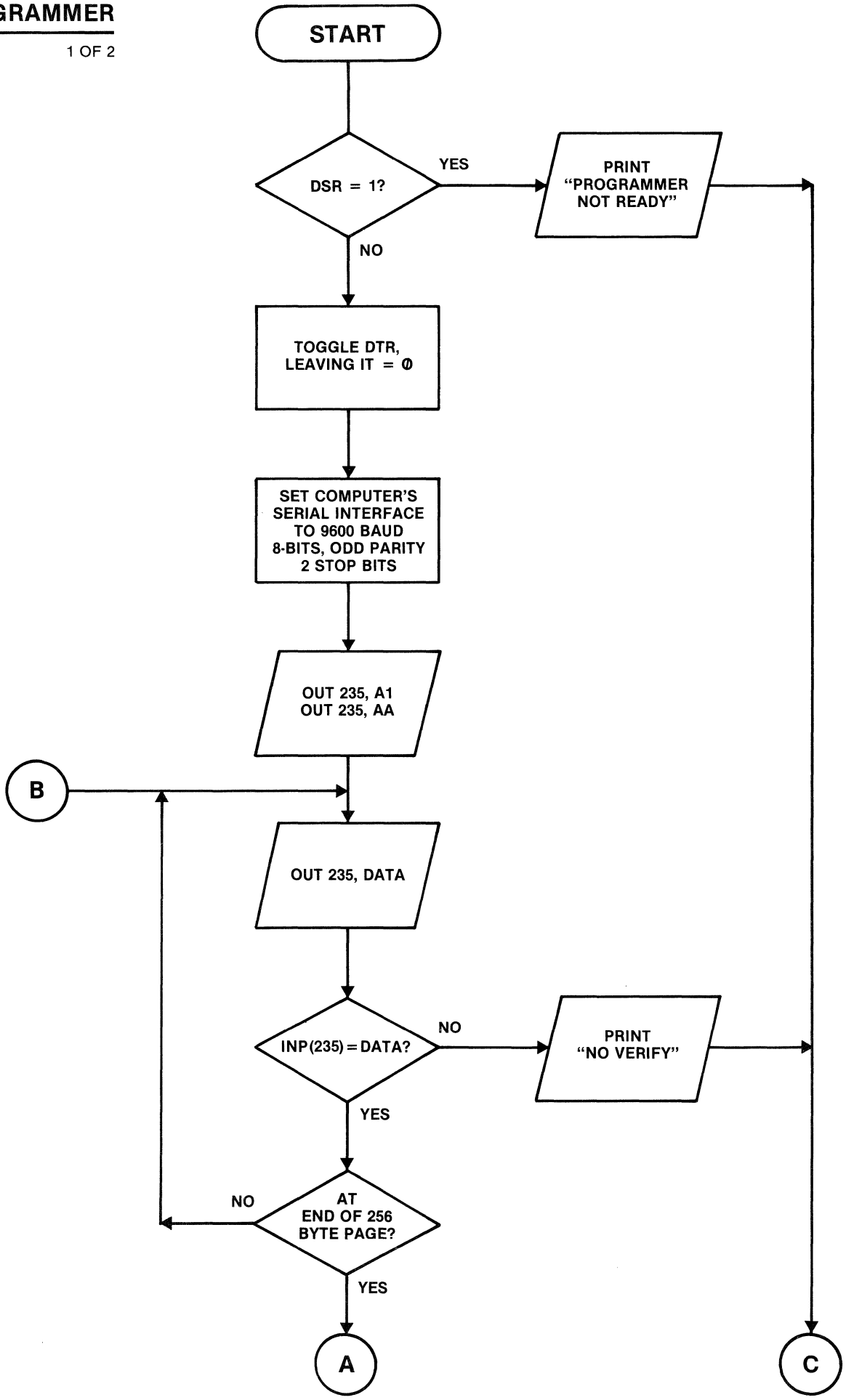
LOAD FROM PROGRAMMER

2 OF 2



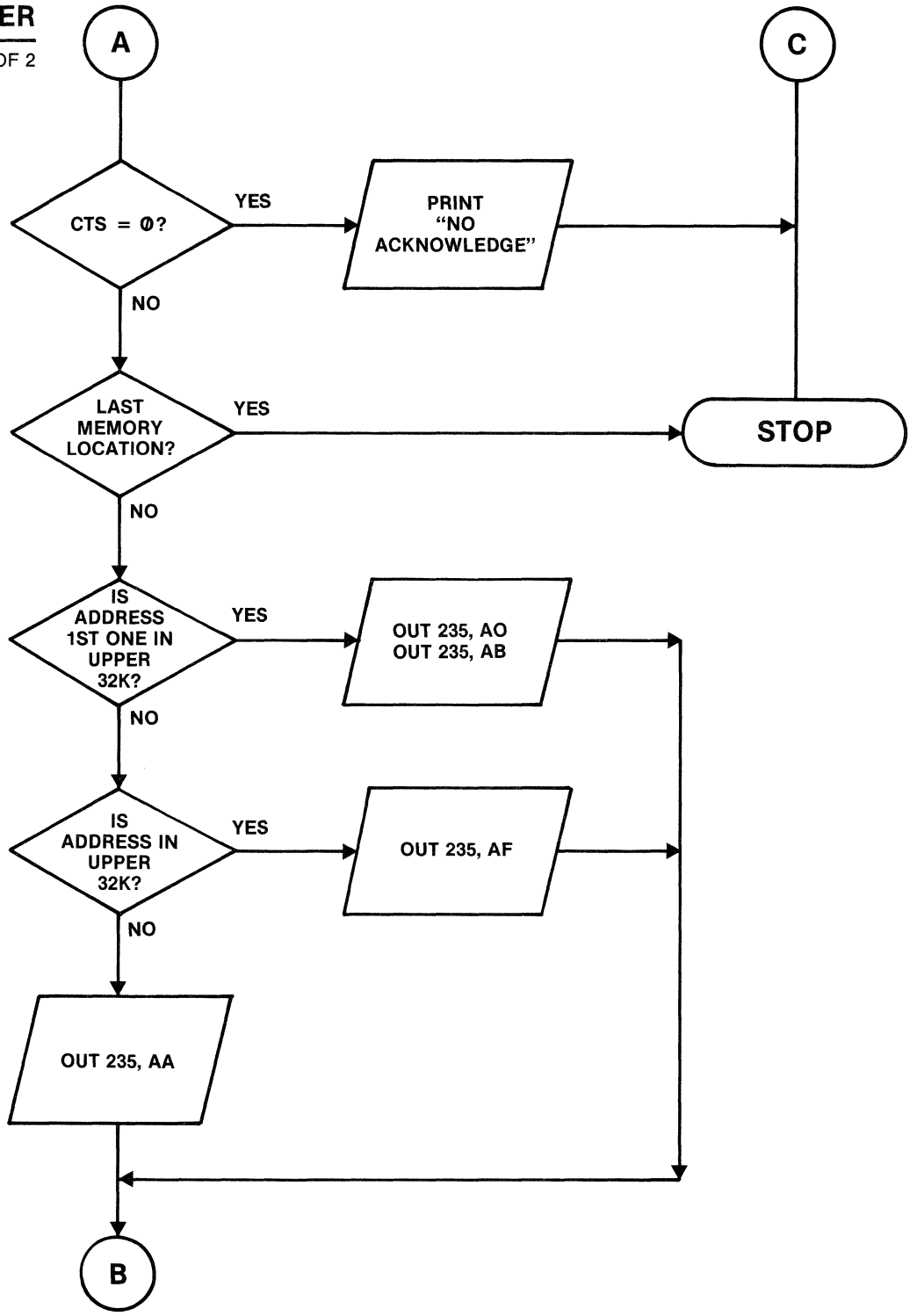
SAVE FROM PROGRAMMER

1 OF 2



SAVE FROM PROGRAMMER

2 OF 2



The program that implements the RS-232C function for the referenced TRS-80 computer is as follows:

SAMPLE PROGRAM FOR USE WITH THE JAMECO JE665 RS-232C INTERFACE BOARD

```
10 ' SERIAL C0MMUNICATI0N PR0GRAM
          BY B0B FAULK
            7/15/82
20 G0SUB 4000:' USED 0NLY F0R TRS-80 MACH. LANG. CASSETTE I/0
40 0UT 232,0:' RESET TRS-80 SERIAL INTERFACE B0ARD
60 CLEAR 100:0N ERR0R G0T0 3420
80 ' BE SURE T0 SET T0P 0F ME10RY T0 24099 (VARIABLE BA -1)
100 DEFINT A-Z:DIM ED(8):NT=256:NR=235:K=99:ST=5:BC=1:I=0:J=0
120 I$=" ":J$=" ":BK=0:0I=1:I=1:KC=1:KF=1:CT=1
140 BA=24100:N0=0:N1=1:N2=2:N3=3:N4=4:N5=5:N6=6:N7=7:N8=8:N9=9
160 ED(1)=1024:ED(2)=2048:ED(3)=4096:ED(4)=8192
180 F0R T=5 T0 7:ED(T)=ED(T-4):NEXT T
200 F0R T=1 T0 7:ED(T)=ED(T)+BA:NEXT T:DA$="0123456789ABCDEF"
220 CLS
240 PRINT "
1) SAVE T0 CASSETTE
2) L0AD FR0M CASSETTE"
260 PRINT "3) L0AD FR0M PR0GRAMMER
4) SAVE T0 PR0GRAMMER"
280 PRINT "5) ME10RY DUMP  (HEX)
6) MEM0RY DUMP  (ASCII)"
300 PRINT "7) KEY
8) EXIT PR0GRAM
"
320 PRINT "C0MMAND ? ";
340 LN=N1:G0SUB 3500
360 A=ASC(A$):IF A<49 0R A>56 THEN PRINT CHR$(N8);:G0T0 340
380 A=VAL(A$)
400 F0R T=N1 T0 50:NEXT T
420 0N A G0SUB 460,660,880,1540,1940,2220,2520,3860
440 G0T0 240
460 CLS:PRINT "SAVE DATA T0 CASSETTE":PRINT:PRINT
480 PRINT "PRESS '1' WHEN READY (0=RETURN) ";:LN=N1:G0SUB 3500
500 IF A$="0" THEN RETURN
520 ' USE THIS 0NLY WITH THE TRS-80 CASSETTE I/0
540 ' IF N0T USING TRS-80, THEN A SUBR0UTINE MUST
560 ' BE WRITTEN THAT WILL DUMP MEM0RY L0CATI0NS
580 ' 24100 T0 32291 T0 DISK 0R CASSETTE.
600 P0KE 16526,36:P0KE 16527,126
620 X=USR(0)
640 RETURN
660 CLS:PRINT "
L0AD DATA FR0M CASSETTE
"
680 PRINT "PRESS '1' T0 BEGIN (0=RETURN) ";:LN=N1:G0SUB 3500
700 IF A$="0" THEN RETURN
720 ' USE THIS R0UTINE 0NLY WITH TRS-80.
740 ' IF N0T USING TRS-80, THEN A SUBR0UTINE MUST
760 ' BE WRITTEN THAT WILL L0AD MEM0RY L0CATI0NS
780 ' 24100-32291 FR0M DISK 0R CASSETTE.
800 P0KE 16526,71:P0KE 16527,126
820 X=USR(0)
840 RETURN
860 ' CHECK F0R PR0GRAMMER READY SIGNAL
880 0UT 232,N0:A=INP(232):IF (A AND 64) THEN PRINT "
PR0GRAMMER N0T READY.":RETURN
900 CLS:PRINT "
```

```
1) ØNE 8K CHIP
2) ØNE 16K CHIP
3) ØNE 32K CHIP
4) ØNE 64K CHIP"
920 PRINT "5) TWØ 8K CHIPS
6) TWØ 16K CHIPS
7) TWØ 32K CHIPS
"
940 PRINT "ENTER YØUR SELECTIØN : ";
960 LN=N1:GØSUB 3500
980 SL=VAL(A$):IF SL<N1 ØR SL>N7 THEN PRINT CHR$(N8);:GØTØ 960
1000 EN=ED(SL):ST=BA:BC=NO
1020 ' PØRTS 232-234 ARE CØNTRØL PØRTS FØR THE SERIAL
1040 ' INTERFACE BØARD. PØRT 235 IS THE SERIAL LINK
1060 ØUT 232,0:' RESET RS-232-C BØARD
1080 ØUT 233,238:' SET BAUD RATE TØ 9600
1100 ØUT 234,118:ØUT 234,116:' SET RS-232-C CØNTRØL REGISTER TØ
1120 ' ØDD PARITY, 8 BIT WØRD, ALSØ THE DATA T. READY IS TØGGLED
1140 '   PRØGRAMMER CØNTRØL WØRD  (SERIAL LINK--PØRT 235) ---
1160 ' D7 = 1
1180 ' D6 = 0
1200 ' D5 = 1
1220 ' D4 = 0
1240 ' D3 : 1-NEXT WØRD IS DATA, 0-NEXT IS A CØNTRØL WØRD
1260 ' D2 : 0-ADDRESS CAN BE ZERØED DURING CTRL WØRD
1280 ' D1 : 1-WRITE IS ENABLED ØN PRØGRAMMER
1300 ' D0 : 1-UPPER ADDRES/MEMØRY BANK, 0-LØWER
1320 ØUT 235,161:' SET CY. AND ZERØ UPPER ADDR.
1340 ØUT 235,168:' ZERØ LØWER ADDRESS AND MAKE NEXT WØRD DATA
1360 ØUT 235,NO:PØKE ST,INP(235)
1380 ST=ST+N1:BC=BC+N1:IF BC<>256 THEN 1360
1400 IF (INP(232) AND 128)<>128 THEN PRINT "
NØ ACKNØWLEDGE IN ADDRESS ";:J=ST-BA:GØSUB 3200:PRINT J$:RETURN
1420 IF ST=EN AND SL>4 THEN ST=4096+BA
1440 BC=NO
1460 IF ST=4096+BA THEN ØUT 235,160:ØUT 235,169:GØTØ 1360
1480 IF (ST=EN AND SL<5) ØR ST=EN+4096 THEN PRINT "
DATA LØAD CØMPLETE.":RETURN
1500 IF ST>4096+BA THEN ØUT 235,173 ELSE ØUT 235,168
1520 GØTØ 1360
1540 ØUT 232,NO:A=INP(232):IF (A AND 64) THEN PRINT"
PRØGRAMMER NØT READY.":RETURN
1560 CLS:PRINT"1) ØNE 8K CHIP
2) ØNE 16K CHIP
3) ØNE 32K CHIP
4) ØNE 64K CHIP"
1580 PRINT "5) TWØ 8K CHIPS
6) TWØ 16K CHIPS
7) TWØ 32K CHIPS
"
1600 PRINT "ENTER YØUR SELECTIØN : ";
1620 LN=N1:GØSUB 3500
1640 SL=VAL(A$):IF SL<N1 ØR SL>N7 THEN PRINT CHR$(N8);:GØTØ 1620
1660 EN=ED(SL):ST=BA:BC=NO
1680 ØUT 232,NO:' RESET RS-232C BØARD
1700 ØUT 233,238:' SET BAUD RATE
1720 ØUT 234,118:ØUT 234,116:' SET RS-232C CØNTRØL REGISTER
1740 ØUT 235,161:' SET CØNTRØL BØARD TØ ZERØ UPPER ADDRESS
1760 ØUT 235,170:' ZERØ LØWER ADDRESS AND MAKE NEXT WØRD DATA
1780 K=PEEK(ST):ØUT NR,K:BK=INP(NR):IF BK<>K THEN PRINT "
NØ VERIFY IN ADDRESS ";:J=ST-BA:GØSUB 3200:PRINT J$:RETURN
1800 ST=ST+N1:BC=BC+N1:IF BC<>NT THEN 1780
```

```
1820 K=INP(232):IF (K AND 128)<>128 THEN PRINT "
NØ ACKNØWLEDGE IN ADDRESS ";:J=ST-BA:GØSUB 3200:PRINT J$:RETURN
1840 IF ST=EN AND SL>4 THEN ST=4096+BA
1860 BC=0:IF ST=4096+BA THEN ØUT 235,160:ØUT 235,171:GØTØ 1780:'
--- IF AT FIRST ADDRESS ØN UPPER PAGE THEN ZERØ UPPER ADDRESS
AND SET SHIFT, THEN GØTØ 1600
1880 IF (ST=EN AND SL<N5) ØR ST=EN+4096 THEN PRINT "
DATA SAVE CØMPLETE.":RETURN
1900 IF ST>4096+BA THEN ØUT 235,175 ELSE ØUT 235,170:'--------
-------IF ØN UPPER PAGE, THEN ØUTPUT CTRL WØRD FØR UPPER PAGE,
ELSE ØUTPUT CØNTRØL WØRD FØR LØWER PAGE
1920 GØTØ 1780
1940 CLS:PRINT "MEMØRY DUMP
"
1960 PRINT "WHAT IS THE START ADDRESS ? ";
1980 LN=N4:GØSUB 3340:PRINT
2000 FØR C=N1 TØ 15
2020 GØSUB 3200:PRINT "    ";J$;" ";
2040 FØR ØI=N1 TØ 16
2060 I=PEEK(J+BA)
2080 GØSUB 2880:PRINT I$;" ";
2100 J=J+N1
2120 NEXT ØI:PRINT:NEXT C
2140 PRINT "PRESS '1' TØ CØNTINUE, '0' TØ RETURN : ";
2160 LN=1:GØSUB 3500
2180 PRINT:IF A$="1" THEN 2000
2200 RETURN
2220 CLS:PRINT "ASCII MEMØRY DUMP
"
2240 PRINT "WHAT IS THE START ADDRESS ? ";
2260 LN=N4:GØSUB 3340:PRINT
2280 FØR C=N1 TØ 15
2300 GØSUB 3200:PRINT J$;" ";
2320 FØR ØI=N1 TØ 16
2340 P=PEEK(J+BA)
2360 IF P>31 AND P<192 THEN PRINTCHR$(P); ELSE PRINT ".";
2380 PRINT " ";
2400 J=J+N1
2420 NEXT ØI
2440 PRINT:NEXT C
2460 PRINT "PRESS '1' TØ CØNTINUE, '0' TØ RETURN : ";
2480 LN=1:GØSUB 3500
2500 PRINT:IF A$="1" THEN 2280ELSE RETURN
2520 CLS:PRINT "
KEY     (XX=RETURN, PRESS ENTER TØ AVØID WRITING IN.)
"
2540 PRINT "ADDRESS ? ";
2560 LN=N4:GØSUB 3340:PRINT
2580 GØSUB 3200:PRINT J$,
2600 I=PEEK(J+BA):GØSUB 2880:PRINT I$,
2620 A$="":INPUT A$:IF A$="" THEN J=J+N1:GØTØ 2580
2640 IF A$="XX" THEN RETURN
2660 I$=A$:GØSUB 2740
2680 PØKE J+BA,I:J=J+N1
2700 GØTØ 2580
2720 ' CØNVERT HEX TØ DECIMAL
2740 I1=ASC(LEFT$(I$,N1))
2760 I2=ASC(RIGHT$(I$,N1))
2780 IF I1>64 THEN I1=I1-55 ELSE I1=I1-48
2800 IF I2>64 THEN I2=I2-55 ELSE I2=I2-48
2820 I=I1*16+I2
2840 RETURN
2860 ' CØNVERT DECIMAL TØ HEX
```

```
2880 I1=INT(I/16)
2900 I2=I-I1*16
2920 I$=MID$(DA$,I1+N1,N1)+MID$(DA$,I2+N1,N1)
2940 RETURN
2960 ' CØNVERT HEX
2980 J1=ASC(LEFT$(J$,N1))
3000 J2=ASC(MID$(J$,N2,N1))
3020 J3=ASC(MID$(J$,N3,N1))
3040 J4=ASC(RIGHT$(J$,N1))
3060 IF J1>64 THEN J1=J1-55 ELSE J1=J1-48
3080 IF J2>64 THEN J2=J2-55 ELSE J2=J2-48
3100 IF J3>64 THEN J3=J3-55 ELSE J3=J3-48
3120 IF J4>64 THEN J4=J4-55 ELSE J4=J4-48
3140 J=J1*4096+J2*256+J3*16+J4
3160 RETURN
3180 ' CØNVERT DECIMAL TØ HEX
3200 J1=INT(J/4096)
3220 J2=INT(J/256)-J1*16
3240 J3=INT(J/16)-J1*256-J2*16
3260 J4=J-J1*4096-J2*256-J3*16
3280 J$=MID$(DA$,J1+N1,N1)+MID$(DA$,J2+N1,N1)
3300 J$=J$+MID$(DA$,J3+N1,N1)+MID$(DA$,J4+N1,N1)
3320 RETURN
3340 LN=N4
3360 GØSUB 3500
3380 J$=A$
3400 GØTØ 2980
3420 LN=N2
3440 GØSUB 3500
3460 I$=A$
3480 GØTØ 2740
3500 ' KEYBØARD INPUT RØUTINE.
 IN: LN=DESIRED LENGTH ØF A$ ØN ØUTPUT
 ØUT: A$=STRING ØF CHARS TYPED IN
3520 A$=INKEY$:A$=""
3540 KC=N0:KF=N0:CT=N0
3560 K$=INKEY$:IF K$="" THEN 3740
3580 K=ASC(K$):IF K$="X" THEN 3680
3600 IF K=N8 AND KC>N0 THEN PRINT CHR$(N8);:KC=KC-N1:
A$=LEFT$(A$,LEN(A$)-N1):GØTØ 3560
3620 IF K=13 AND KC=LN THEN RETURN
3640 IF K<47 ØR (K>57 AND K<65) ØR K>70 THEN 3740
3660 IF KC=LN THEN 3560
3680 KC=KC+N1
3700 PRINT K$;
3720 A$=A$+K$
3740 CT=CT+N1
3760 IF CT<N7 THEN 3560
3780 CT=0
```

```
3800 IF KF=NO THEN KF=N1:PRINT CHR$(14); ELSE KF=NO:
PRINT CHR$(15);
3820 GØTØ 3560
3840 PRINT "
ADDRESS ERRØR.
":RESUME 240
3860 END
3880 ' USE THE SUBRØUTINE BELØW TØ PØKE THE TRS-80
3900 ' MACHINE LANGUAGE CASSETTE I/Ø PRØGRAM INTØ MEMØRY
3920 ' BE SURE THE TØP ØF MEMØRY IS SET TØ 24090.
3940 ' IF USING ANØTHER MACHINE, THEN A RØUTINE SHØULD
3960 ' BE WRITTEN THAT WILL READ AND WRITE ME1ØRY LØCATIØNS
3980 ' 24100 (VARIABLE BA) TØ 32291 (BA+8192) TØ TAPE ØR DISK
4000 FØR A=32292 TØ 32361:READ B:PØKE A,B:NEXT A:RETURN
4020 DATA 62,1,205,18,2,205,132,2,33,36,94,126,205,100,2
4040 DATA 35,17,36,126,122,188,194,47,126,123,189,194,47
4060 DATA 126,62,1,205,248,1,201,62,1,205,18,2,205,150,2
4080 DATA 33,36,94,205,53,2,119,35,17,36,126,122,188,194
4100 DATA 82,126,123,189,194,82,126,62,1,205,248,1,201
```

ASSEMBLY LANGUAGE CASSETTE I/O PROGRAM

The portion of the program listed below in Assembly Language will produce the Machine Code in lines 4020 through 4100 of the main program (above). This listing is not required to communicate with the JE664 Programmer; it is supplied only as an example for experienced assembly-language programmers who desire cassette or disk I/O on a computer other than the TRS-80.

```
7E24           00100          ØRG     32292D
7E24 3E01      00200 SAVE     LD      A,1
7E26 CD1202    00300          CALL    212H     ; TURN ØN CASSETTE
7E29 CD8402    00400          CALL    284H     ; WRITE LEADER AND SYNC BYTE
7E2C 21245E    00500          LD      HL,24100D ; HL=BASE ADDRESS
7E2F 7E        00600 LØØPS    LD      A,(HL)
7E30 CD6402    00700          CALL    264H     ; WRITE BYTE
7E33 23        00800          INC     HL
7E34 11247E    00900          LD      DE,SAVE
7E37 7A        01000          LD      A,D
7E38 BC        01100          CP      H
7E39 C22F7E    01200          JP      NZ,LØØPS ; LØØP IF NØT LAST ADDRESS
7E3C 7B        01300          LD      A,E
7E3D BD        01400          CP      L
```

```
7E3E C22F7E    01500          JP      NZ,L00PS
7E41 3E01      01600          LD      A,1
7E43 CDF801    01700          CALL    1F8H    ; TURN 0FF CASSETTE
7E46 C9        01800          RET
               01900
7E47 3E01      02000 L0AD     LD      A,1
7E49 CD1202    02100          CALL    212H    ; TURN 0N CASSETTE 1
7E4C CD9602    02200          CALL    296H    ; READ LEADER AND SYNC BYTE
7E4F 21245E    02300          LD      HL,24100D ; HL=BASE ADDRESS
7E52 CD3502    02400 L00PL    CALL    235H    ; READ BYTE
7E55 77        02500          LD      (HL),A  ; ST0RE BYTE
7E56 23        02600          INC     HL
7E57 11247E    02700          LD      DE,SAVE
7E5A 7A        02800          LD      A,D
7E5B BC        02900          CP      H
7E5C C2527E    03000          JP      NZ,L00PL ; L00P IF N0T LAST ADDRESS
7E5F 7B        03100          LD      A,E     ;
7E60 BD        03200          CP      L       ;
7E61 C2527E    03300          JP      NZ,L00PL;
7E64 3E01      03400          LD      A,1
7E66 CDF801    03500          CALL    1F8H    ; TURN 0FF CASSETTE
7E69 C9        03600          RET
7E24           03700          END     SAVE
00000 T0TAL ERR0RS
L00PL   7E52
L0AD    7E47
L00PS   7E2F
SAVE    7E24
```

CIRCUIT DESCRIPTION

Referring to the JE665 (option) schematic, incoming serial data at RS-232C connector pin 2 are converted, in UART IC U8, to parallel data on the 8-line data bus connecting to J1. Conversely, when transmitting serial data on the RS- bus, the 8-bit parallel data feed into U8 and is converted therein to serial data leading to RS-connector pin 3.

ICU11, a 4-bit latch, stores control-word data as follows:
Bit 0 ("SHIFT") selects upper or lower 32K ("1" selects upper);
Bit 1 enables/disables WRITE ("1" enables);
Bit 2 enables/disables ADDRESS clearing ("0" enables);
Bit 3 signifies type of next word ("0" signifies a control word, "1" signifies that data will follow).

The sequence of events starts with a READY pulse (DTR) from the computer at RS-connector pin 20 and proceeds according to the outline following. To enable RS-232C operation, signal DSR is set High, via 5C, by a Low signal at J2 pin 4 that also enables the several other logic circuits connected to that J2 signal.

1. **Receive DTR low pulse via 2A.**
   - A. Reset control-bit latch U11.
     - (1) Hold latch 6A High via "0" at U11 pin 6, 3D, CR4.
       - (a) Hold CTS Low via 5B.
       - (b) Enable W & DIR/READ gates at J2 via 6B.
       - (c) Enable MS hex A detector 3A, 3B & 4B.

2. **Receive word [A(control)] via 2B and U8.**
   - A. Detect hex A in 3A, 3B & 4B.
     - (1) Trigger U1B.
       - (a) Enable W & DIR/READ gates at J2 via 6B.
       - (b) Trigger programmer board clock via CR2, 9E, CR6 & J1 pin 11.
       - (c) Enable ADDRESS-CLEAR at 6C.
     - (2) Latch (control) bits in U11.
       - (a) Bit 0: Select upper/lower 32K (U11 pin 5 via 10 & 12, J1 to main board J4).
       - (b) Bit 1: "1" enables WRITE at 3C.
       - (c) Bit 2: "0" enables ADDRESS-CLEAR via U11 pin 4, CR3 & 6D.
       - (d) Bit 3: A "1" latched at U11 pin 6 removes the latch-hold from DTR latch 6A (via 3D & CR4). The "1" signifies last control word.
   - B. Generate delayed word-receipt pulses at UIA and 9F from U8's outputs to 4A.
     - (1) Apply word 00 to address counters (UIA pin 9 via 6C to gates 7 & 10).
     - (2) Reset DTR latch 6A ***IF*** control-word bit 3 is "1" (see [d] above). (UIA pin 9 via 6C & D1).
       - (a) Set CTS High via 5B.
     - (3) Load address counters with 00 (6C, 6D, 2D & R11 & R20 to J1 and thence to programmer board J4).

3. **Receive 256-word data block, after last control word (see [2] above), via 2B and U8.**
   - A. Write each 8-bit word into RAM.
     - (1) Serial data via 2B to U8.
     - (2) Parallel data from U8 to J1.
     - (3) Write-pulse from U8 via 4A, 3C and gate 7.
     - (4) Address-counter clock from U8 via 9E, CR6 & J1 to programmer board J4.
   - B. Read each newly-written word from RAM and return it to external computer.
     - (1) Parallel data from J1 to U8.
     - (2) Serial data from U8 via 5A.
     - (3) DIR/READ signal from UIA via gate 7 to J2.

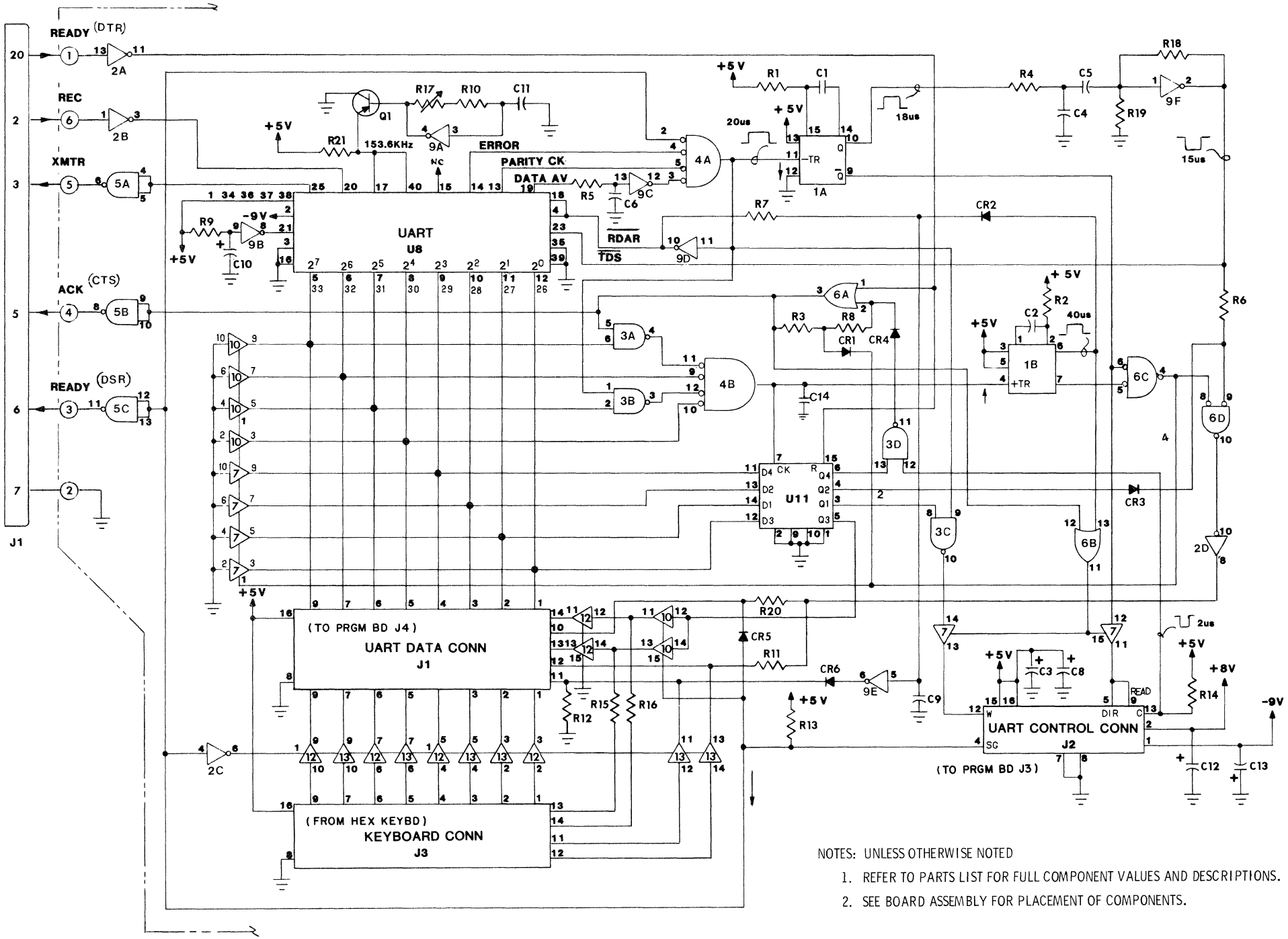
4. **Sense address LSDs reaching count FF.**
   - A. Set DTR latch 6A High (by the Low carry from J2 pin 13 inverted through 3D and CR4).
     - (1) Set CTS Low (logic "1") via 5B.
       - (a) Computer senses CTS = 1 and delivers a new control word.

5. **Go to 2. above unless 256-word data block is the last one.**
   - A. If so, the RS-232C interchange will stop.

Several features should be noted in addition:

1. Gates 2A, 2B, 5A, 5B and 5C convert between the RS-232C logic levels and the TTL levels of 0V and +5V.
2. UART U8's baud rate is set at 9600 by oscillator 9A, buffered by Q1, set to 153.6KHz ± 2%.
3. Transmission gates 12 and 13 implement disconnection of the keyboard outputs during RS-232C operation. They are controlled via 2C from J2's RS-232C enable signal. Gates 10 and diode CR5 also implement this function.

JE665 RS-232 INTERFACE SCHEMATIC DIAGRAM

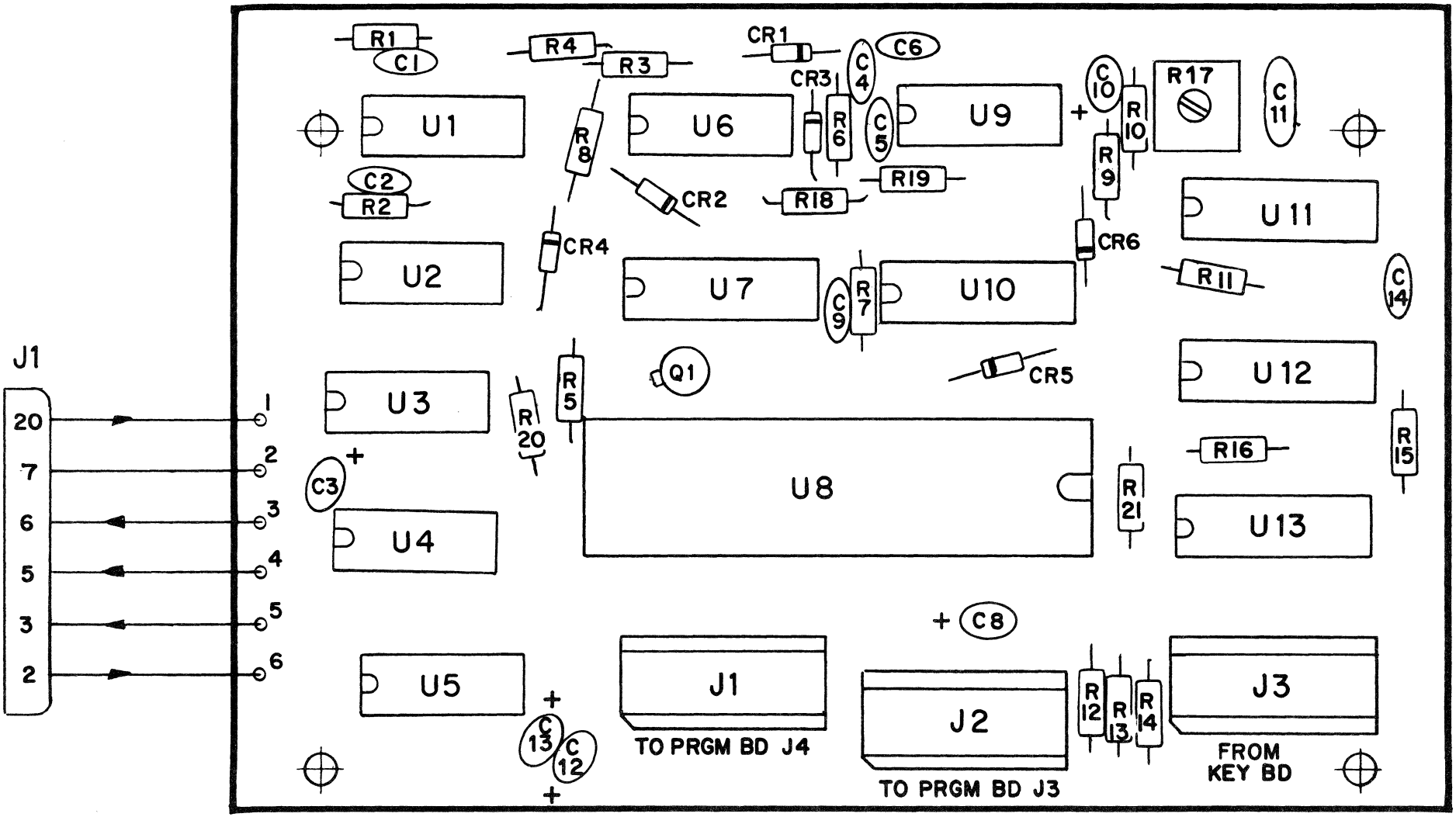


NOTES: UNLESS OTHERWISE NOTED

1. REFER TO PARTS LIST FOR FULL COMPONENT VALUES AND DESCRIPTIONS.
2. SEE BOARD ASSEMBLY FOR PLACEMENT OF COMPONENTS.

JE665 RS-232C INTERFACE BOARD ASSEMBLY

REFER TO PARTS LIST FOR FULL COMPONENT VALUES AND DESCRIPTIONS.



JE665 RS-232 INTERFACE BOARD COMPONENT MATERIAL LIST

SCHEM REF	DESCRIPTION	QTY
U1	IC—CD4098, DUAL MULTI VIBRATOR	1
U2	IC-LM1489, QUAD LINE RECVR	1
U3	IC-CD4011, QUAD NAND GATE	1
U4	IC-CD4002, DUAL NOR GATE	1
U5	IC-LM1488, QUAD LINE DRIVER	1
U6	IC-CD4017, DECADE CNT/DIV	1
U7,10,12,13	IC-CD4503, TRI HEX BUFFER	3
U8	IC-AY-5-1013A, UART IC	1
U9	IC-74C14, HEX SCH TRIGGER	1
U11	IC-CD4076, QUAD REGIST	1
C1,4,9	CAPACITOR, DISC, 220PF, 50V	3
C2,C6	CAPACITOR, DISC, .001MF, 50V	2
C3,8,10,12,13	CAPACITOR, TANT, 2.2MF, 35V	5
C5,C14	CAPACITOR, DISC, 100PF, 50V	2
C11	CAPACITOR, MICA, 240PF, 5%, 500V	1
CR1-CR6	DIODE, 1N4148	6
J1	CONNECTOR, RS-232, 25-PIN, DB25S	1
R1	RESISTOR, ¼W, 5%, 68K	1
R2	RESISTOR, ¼W, 5%, 47K	1
R3,8,14	RESISTOR, ¼W, 5%, 27K	3
R4,7	RESISTOR, ¼W, 5%, 4.7K	2
R5	RESISTOR, ¼W, 5%, 18K	1
R6,11,15,16,20,21	RESISTOR, ¼W, 5%, 10K	6
R9,12,19	RESISTOR, ¼W, 5%, 100K	3
R10	RESISTOR, METAL FILM 10K, 1%	1
R13	RESISTOR, ¼W, 5%, 1K	1
R17	RESISTOR, TRIM POT, 1K (840P-1K)	1
R18	RESISTOR, ¼W, 5%, 200K	1
Q1	TRANSISTOR, 2N2907A	1
	RS232 PC BOARD JE665-1	1

JAMECO ELECTRONICS • 1355 SHOREWAY ROAD • BELMONT, CALIFORNIA 940002 • (415) 592-8097