

Computer

A MICROCOMPUTER-BASED TELETYPE CONTROL SYSTEM (1)

Nathan M. Reiss and Anthony J. Broccoli
 Department of Meteorology and Physical Oceanography
 Cook College
 Rutgers, The State University of New Jersey
 New Brunswick, NJ 08903

ABSTRACT

A computer program is developed which scans the sequence headings of incoming teletype data, and decides on the basis of these headings whether the data should be forwarded to the printer. Microcomputer requirements for implementation are given, as well as a diagram showing electrical connections.

1. INTRODUCTION

This department makes extensive use of the Medium Speed Weather Circuit (GD-90488-604) for its teleprinter (TTY) data. However, we have found that it is not necessary for us to receive all of the material that is available on the circuit. For some years we had attempted to select what we needed by connecting the TTY to a timer (Simplex model GR-83) that had a one-minute resolution. This proved to be a less than satisfactory solution in that much of the material is received either on a flexible time schedule or on an unscheduled basis. Furthermore, making temporary (or, for that matter, permanent) changes in the schedule was a cumbersome process.

We have found a satisfactory solution to this problem in the form of a microcomputer that scans the headings of incoming data sequences and decides, by comparing these to a group of headings stored in its memory, whether the data sequence should be forwarded to the printer. This method does not involve timers, and changes in the list of desired headings can be easily made. In this paper we present a program written for use with a microcomputer, together with minimum specifications for the microcomputer itself, and instructions for interfacing the microcomputer to the data line and to the TTY.

2. DATA DESCRIPTION

Data are received on the Medium Speed Weather Circuit as a serial stream of binary bits at a rate of 1200 baud (bits per second). The characters are coded according to the USASCII format, with special provisions for weather symbols. Each character consists of eleven bits, including one start

bit, seven information bits, one parity bit (set to produce odd parity), and two stop bits. This format is compatible with the protocol generally in use for transmitting data between other types of digital devices. At the users end, these data first pass through a modulator/demodulator (modem) which, among other things, senses the start and stop bits and produces a synchronization signal for the receiver.

3. MICROCOMPUTER REQUIREMENTS

The programs described here may be used with little, if any, modification with any microcomputer whose operation is based on the popular Z-80 microprocessor chip. Minor changes in the programs may be needed if they are to be used with other microprocessors.

If the programs are available on some sort of storage medium (e.g. a cassette tape), and if no further program development is contemplated by the user, the following are the minimum microcomputer requirements:

- 1 Z-80 based (8-bit) central processing unit (CPU), with a clock frequency of at least 1 Mhz
- 1 system monitor board
- 4 kilobytes of random access memory (RAM)
- 1 Mainframe
- 2 serial input/output (I/O) ports, with speed selectable to 1200 baud
- 1 cassette or disk drive (with appropriate software)

In addition to these components, some means of issuing instructions to the microcomputer and monitoring its operation will be required, for example, a cathode ray tube (CRT) terminal. However, this terminal need be connected only when the program, or changes in the desired sequence headings,

are being entered. At other times the terminal is available for other uses.

The components described above can often be found already assembled as a system. If further program development is planned, a second cassette (or disk) drive, with appropriate software, as well as 16-32 kilobytes of random access memory is recommended.

4. ELECTRICAL CONNECTIONS

The microcomputer is inserted between the modem, which serves as the interface between the telephone line and the user and the TTY. No modifications to the modem or to the TTY are required.

Received data flow from the modem to I/O port #2 of the microcomputer. Data that have been selected by the microcomputer for forwarding flow from I/O port #1 to the TTY. A diagram showing the electrical connections is given in Figure 1.

5. COMPUTER PROGRAMS

Although the actual selection of TTY data is accomplished by means of a single computer program, a total of three relevant programs are described here. All programs are written in Z-80 assembly language. The first, HDNGS, is used for entering sequence headings into the computer; the second CHKLST, is used for reviewing the current list of headings in the computer's memory; the third, SLCTNW, actually screens the incoming TTY data. Program listings are given in the Appendix section of this paper.

An 8-bit microprocessor is capable of storing any number between zero and 255 in one of its storage locations. The sequence heading numbers that one might wish to store range from 200 (New England States) to 432 (Global Satellite Prediction). Since there are, in all, fewer than 256 different numbers, it is possible to convert each sequence heading number to a number less than 256 by subtracting 200 from each heading number. SLCTNW automatically performs this conversion on data received via TTY. However, the headings entered into memory by the user via CHKLST must first be converted.

The procedure for using these programs is as follows:

1. Plug the CRT terminal into I/O port #1 of the microcomputer.
2. Load HDNGS into the memory of the microcomputer from the cassette tape or disk. When HDNGS is loaded, a "?" will appear on the screen to prompt the user to enter a sequence heading number. When the heading number is entered, the "carriage return" key

is pressed, causing another "?" to appear, and so on. After the last heading, the number 255 (hexadecimal FF) should be entered. This causes the program to terminate, leaving the desired headings stored in the memory of the microcomputer.

3. Load CHKLST into the memory of the microcomputer from the cassette tape or disk. When CHKLST is loaded, the list of stored headings will automatically be displayed at the CRT terminal. Changes may be made by rerunning HDNGS.

4. If necessary, reset the I/O speed of the microcomputer and CRT to 1200 baud.

5. Connect the TTY signal coming from the modem to I/O port #2.

6. Load SLCTNW into the memory of the microcomputer from the cassette tape or disk. When SLCTNW is loaded, TTY data will appear on the CRT screen whenever a sequence whose heading number is stored in the memory of the microcomputer is encountered.

7. Unplug the CRT from I/O port #2 and replace it with the plug from the TTY printer. SLCTNW will continue to run indefinitely (i.e., until the microcomputer is reset or until there is a power outage).

6. DISCUSSION

These programs provide a means whereby one may select precisely the material that one wishes to receive on the Medium Speed Weather Circuit. For example, a one-line "special" observation from an area of interest can be printed, while the material immediately preceding and following it can be rejected. With minor modifications, these programs could also be used to route incoming data to two or more printers, depending on the sequence headings. For example, surface observations could be routed to one printer, while upper-air reports could be routed to another. By adding a second level of logic to the SLCTNW program, individual stations within sequences could be singled out.

FOOTNOTE

(1) *Paper of the Journal Series, New Jersey Agricultural Experiment Station, Cook College, Rutgers, The State University of New Jersey, New Brunswick, NJ 08903.*

Nathan M. Reiss is associate Professor of Meteorology. His major interest is in the area of modeling physical processes of the atmosphere.

Anthony J. Broccoli is an instructor and agricultural meteorologist. His primary interest is the application of computers to agrometeorological data gathering and dissemination.

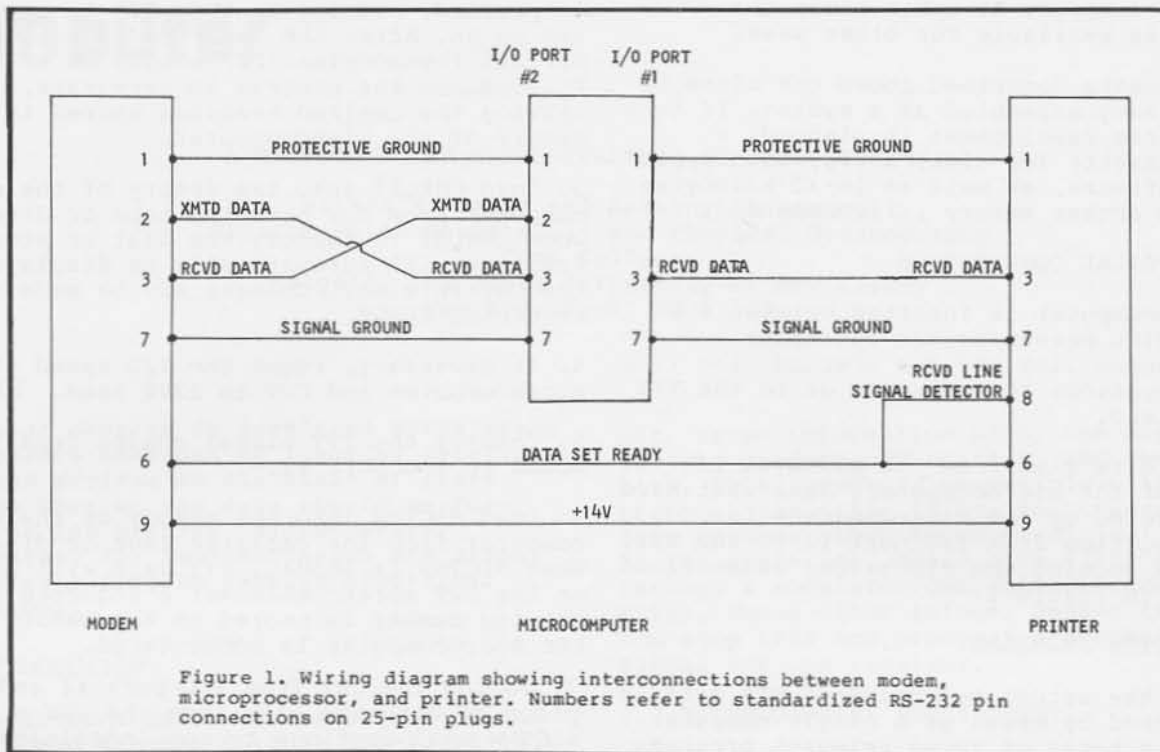


Figure 1. Wiring diagram showing interconnections between modem, microprocessor, and printer. Numbers refer to standardized RS-232 pin connections on 25-pin plugs.

HDNGS

```

0100          ORG      100H      #START THE PROGRAM AT LOCATION 100H
0100 210002    LXI      H,0200H  #SET THE MEMORY POINTER AT THE START OF THE ID #'S
0103 3E3F     START: MVI      A,3FH #SEND A QUESTION MARK
0105 CD5A01   CALL     TTYOUT  #CALL THE OUTPUT ROUTINE
0108 CD2301   CALL     HEXBIN  #READ TWO HEXADFCIMAL CHARACTERS
010B 77       MOV      M,A      #LOAD THEM INTO MEMORY
010C FEFF     CPI      OFFH    #ARE THEY OFFH?
010E D22001   JNC      STOP    #YES--SO STOP THE PROGRAM
0111 CD4F01   CALL     TTYIN   #ACCEPT A CARRIAGE RETURN
0114 CD5A01   CALL     TTYOUT  #ECHO THE CARRIAGE RETURN
0117 3E8A     MVI      A,8AH   #SEND A LINE FEED
0119 CD5A01   CALL     TTYOUT  #ECHO THE LINE FEED
011C 23       INX      H      #INCREMENT H
011D C30301   JMP      START  #READ ANOTHER ID #
0120 C30000   STOP:  JMP      0      #RETURN TO THE OPERATING SYSTEM
0123 CD3001   HEXBIN: CALL    HEXIN   #GET A HEX CHARACTER
0126 07       RLC      #ROTATE THE BINARY EQUIVALENT INTO
0127 07       RLC      # THE 4 MSB'S
0128 07       RLC
0129 07       RLC
012A 4F       MOV      C,A      #SAVE THE VALUE IN C TEMPORARILY
012B CD3001   CALL     HEXIN   #GET THE OTHER HEX CHARACTER
012E 81       ADD      C      #ADD THE TWO DIGITS TOGETHER
012F C9       RET      #RETURN WITH THE VALUE IN A
0130 CD4F01   HEXIN: CALL    TTYIN   #INPUT A CHARACTER
0133 CD5A01   CALL     TTYOUT  #ECHO THE CHARACTER
0136 FE30     CPI      30H    #IS THE CODE LESS THAN 30H?
0138 DA3001   JC      HEXIN   #YES--THEN IGNORE IT
0138 FE3A     CPI      3AH    #IS THE CODE LESS THAN THOSE FOR 0-9?
013D DA4C01   JC      NOLET   #YES--IT IS A NUMBER--JUMP TO NOLET
0140 FE41     CPI      41H    #IS THE CODE LESS THAN ASCII A?
0142 DA3001   JC      HEXIN   #YES--THEN IGNORE IT
0145 FE47     CPI      47H    #IS IT LESS THAN ASCII G?
0147 D23001   JNC     HEXIN   #NO--THEN IGNORE IT
014A C609     ADI      09H    #ADD 09H TO A THROUGH F TO CONVERT THEM
                    # TO EQUIVALENT BINARY NUMBERS
                    #NOW SAVE ONLY THE 4 LSB'S
014C E60F     NOLET: ANI      0FH    #CHECK STATUS
014E C9       RET      #ANY INCOMING DATA?
014F DB10     TTYIN: IN       10H    #NO--KEEP CHECKING
0151 1F       RAR      #A CHARACTER IS RECEIVED--READ IT
0152 D24F01   JNC     TTYIN   #MASK OUT THE PARITY BIT
0155 DB11     IN       11H
0157 E67F     ANI      7FH
0159 C9       RET
    
```

```

015A 5F      TTYOUT: MOV    E+A    #MOVE THE CHARACTER TO BE PRINTED
           # INTO THE E-REGISTER
015B DB10    TTYO:  IN     10H    #CHECK STATUS
015D E602    ANI     02H    #MASK ALL BUT THE TRANSMIT DATA READY BIT
015F FE02    CPI     02H    #IS THAT BIT A LOGIC 1?
0161 C25B01  JNZ     TTYO    #NO--SO TRY AGAIN
0164 7B      MOV     A+E    #LOAD THE CHARACTER TO BE PRINTED
           # INTO THE ACCUMULATOR
0165 D311    OUT     11H    #PRINT THE CHARACTER
0167 C9      RET
0168        END

```

CHKLST

```

0100        ORG     100H
0100 21FF01  LXI     H,01FFH #SET THE MEMORY POINTER TO THE LOCATION JUST
           # AHEAD OF THE DESIRED ID #'S
0103 23      NXT:  INX     H     #INCREMENT THE MEMORY POINTER
0104 7E      MOV     A+M    #MOVE THE ID # INTO THE ACCUMULATOR
0105 CD2401  CALL    BINHEX  #CALL THE HEXADECEMAL OUTPUT SUBROUTINE
0108 3E8B    MVI     A,80H    #SEND A CARRIAGE RETURN
010A CD5101  CALL    TTYOUT   #SEND A LINE FEED
010D 3E8A    MVI     A,8AH    #SEND A LINE FEED
010F CD5101  CALL    TTYOUT   #SEND A LINE FEED
0112 CD3E01  CALL    ONESEC   #GENERATE A ONE-SECOND TIME DELAY
0115 3E32    MVI     A,50D    #SET 50D AS THE MAXIMUM NUMBER OF MEMORY LOCATIONS LISTED
0117 B0      CMP     L
0118 CA2101  JZ      STOP
011B 7E      MOV     A+M    #MOVE THE ID # BACK INTO THE ACCUMULATOR
011C FEFF    CPI     0FFH   #IS IT OFFH?
011E DA0301  JC      NXT     #NO--GO GET THE NEXT ID # FROM THE LIST
0121 C30000  STOP:  JMP     0
0124 4F      BINHEX: MOV    C+A    #SAVE THE BINARY NUMBER IN C
0125 E6F0    ANI     0F0H    #SAVE THE FOUR MSB'S IN THE A-REGISTER BY ANDING
           # IT WITH 0F0H
           # ROTATE THE FOUR MSB'S INTO
           # THE FOUR LSB'S
0127 0F      RRC
0128 0F      RRC
0129 0F      RRC
012A 0F      RRC
012B CD3101  CALL    PHEX    #PRINT THE HEXADECEMAL CHARACTER EQUIVALENT OF
           # THE CONTENT OF THE A-REGISTER
           # GET THE ORIGINAL BINARY NUMBER INTO THE A-REGISTER
012E 79      MOV     A+C
012F E60F    ANI     0FH     #SAVE JUST THE FOUR LSB'S
0131 FE0A    PHEX:  CPI     0AH     #SHOULD A NUMBER OR A LETTER BE PRINTED?
0133 DA3B01  JC      NMBOK   #YES, IT SHOULD BE A NUMBER
0136 E607    ADI     07H     #NO, IT'S A LETTER--SO ADD 07H
0138 C6B0    NMBOK: ADI     0B0H  #ADD 0B0H TO CONVERT IT TO ASCII
013A CD5101  CALL    TTYOUT  #CALL THE PRINTING SUBROUTINE
013D C9      RET
013E 0E06    ONESEC: MVI     C,06H  #LOAD THE C-REGISTER WITH 06H
0140 1668    GENDLY: MVI     D,68H  #LOAD THE D-REGISTER WITH 68H
0142 1E00    MVI     E,00H    #LOAD THE E-REGISTER WITH 00H
0144 1D      WAIT:  DCR     E
0145 C24401  JNZ     WAIT    #JUMP TO THE DECREMENT E INSTRUCTION
0148 15      DCR     D
           #E IS ZERO SO DECREMENT D
0149 C24401  JNZ     WAIT    #IF D IS NONZERO, JUMP TO WAIT
014C 0D      DCR     C
           #TIME IS UP!!!
014D C24401  JNZ     WAIT    #JUMP BACK TO WAIT
0150 C9      RET
0151 5F      TTYOUT: MOV    E+A    #MOVE THE CHARACTER TO BE PRINTED INTO THE
           # E-REGISTER
0152 DB10    TTYO:  IN     10H    #CHECK THE STATUS WORD
0154 E602    ANI     02H    #MASK ALL BUT THE TRANSMIT DATA READY BIT
0156 FE02    CPI     02H    #IS THAT BIT A LOGIC 1?
0158 C25201  JNZ     TTYO    #NO--SO GO BACK
015B DA5201  JC      TTYO    #AND TRY AGAIN
015E 7B      MOV     A+E    #LOAD THE CHARACTER TO BE PRINTED IN THE
           # ACCUMULATOR
015F D311    OUT     11H    #SEND IT TO THE TERMINAL
0161 C9      RET

```

SLCTNW

```

0100        ORG     100H
0100 310003  LXI     SP,300H #SET UP THE STACK POINTER
01FF =      HDGS  EQU     01FFH #THE DESIRED ID #'S WILL FOLLOW IMMEDIATELY
           # MEMORY LOCATION 01FFH
0012 =      P1   EQU     12H  #STATUS/CONTROL PORT 2 - DATA RECEIVED
0013 =      P2   EQU     13H  #INPUT PORT 2 - DATA RECEIVED
0010 =      P3   EQU     10H  #STATUS/CONTROL PORT 2 - TO PRINTER
0011 =      P4   EQU     11H  #OUTPUT PORT 1 - TO PRINTER
0103 3E76    MVI     A,76H    #SET THE DATA RATE FOR PORT 2 AT 1200 BAUD
0105 D317    OUT     17H
0107 3E68    MVI     A,68H
0109 D315    OUT     15H
010B 3E00    MVI     A,00H
010D D315    OUT     15H
010F 3E12    MVI     A,12H    #SET THE STATUS WORD FOR PORT 1
0111 D310    OUT     F3
0113 CDAB01  SEARCH: CALL   TTYIN   #INPUT A CHARACTER
0114 4F      MOV     C+A    #MOVE IT INTO THE C-REGISTER
0117 3EFF    MVI     A,0FFH  #LOAD THE 'PRINT-ON' WORD
0119 B8      CMP     B
           #CHECK THE PRINT CONTROL REGISTER
011A DA4501  JC      50H     #PRINT CONTROL IS OFF, SO
           # CHECK FOR AN 'SOH'
011D C24501  JNZ     50H     #PRINT CONTROL IS ON--CHECK FOR AN 'ETX'
0120 3E03    MVI     A,03H   #LOAD THE 'ETX' CHARACTER
0122 B9      CMP     C
           #IS THE CHARACTER JUST RECEIVED AN 'ETX'?
0123 DA2B01  JC      PRINT  #IT IS NOT AN 'ETX'--JUMP
0126 C22B01  JNZ     PRINT  #TO THE PRINTING SEGMENT
0129 0600    MVI     B,00H   #IT IS AN 'ETX'--TURN PRINT CONTROL OFF
012B 3EFF    MVI     A,0FFH  #LOAD THE 'PRINT-ON' WORD
012D B8      CMP     B
           #CHECK PRINT CONTROL--IS THE PRINTER ON?
012E DA1301  JC      SEARCH  #PRINT CONTROL IS OFF--NOTHING

```

```

0170 E60F      ANI    0FH    ;MASK OUT THE FOUR HIGH-ORDER BITS
0172 57        MOV    D,A    ;TAKE THAT DIGIT AND MOVE IT TO THE D-REGISTER
0173 1E0A      MVI    E,10D  ;LOAD 10D INTO THE E-REGISTER
0175 CD9201    CALL  MP88   ;MULTIPLY THE TWO
0178 D1        POP    D    ;POP THE RESULT OF THE FIRST MULTIPLICATION
                ; BACK FROM THE STACK
0179 7B        MOV    A,E    ;MOVE THE RESULT OF THE FIRST MULTIPLICATION
                ; INTO THE ACCUMULATOR
017A 85        ADD    L    ;ADD THE RESULT OF THE SECOND MULTIPLICATION
                ; TO THE RESULT OF THE FIRST
017B 57        MOV    D,A    ;MOVE IT BACK TO THE D-REGISTER
017C CDAB01    CALL  TTYIN  ;READ THE THIRD CHARACTER OF THE ID # (THE UNITS'
                ; DIGIT)
017F E60F      ANI    0FH    ;MASK OUT THE FOUR HIGH-ORDER BITS
0181 82        ADD    D    ;THIS COMPLETES THE CONVERSION--THE ID #
                ; IS PACKED INTO AN 8-BIT WORD'
                ;
                ;THIS SECTION CHECKS THE RECEIVED ID #'S AGAINST
                ;THE LIST OF DESIRED ID #'S--THE DESIRED ID #'S ARE
                ;IN ASCENDING ORDER--THE SEARCH IS
                ;RECEIVED ID # IS EQUALLED OR EXCEEDED BY ONE, OF THE
                ;DESIRED ID #'S
                ;
0182 21FF01    LXI    H,HDGS ;SET THE MEMORY POINTER TO THE LOCATION THAT
                ; PRECEEDS THE BEGINNING OF THE DESIRED ID #'S
0185 23        COMPAR: INX   H    ;INCREMENT THE MEMORY POINTER
0186 8E        CMP    M    ;COMPARE THE ID # WITH THE DESIRED ID # LIST
0187 DA1301    JC     SEARCH ;THE ID # IS TOO LOW--LOOK FOR MORE DATA
018A C28501    JNZ   COMPAR ;THE ID # IS TOO HIGH--TRY ANOTHER
018B 06FF      MVI    B,OFFH ;THE ID # IS FOUND IN THE DESIRED ID # LIST
                ; THE PRINTER ON
018F C31301    JMP    SEARCH ;GET THE SELECTED DATA
0192 210000    MP88: LXI    H,00H
0195 0E08      MVI    C,08H
0197 7A        NXTBIT: MOV   A,D
0198 1F        RAR
0199 57        MOV    D,A
019A D2A001    JNC   NOADD
019D 7C        MOV    A,H
019E 83        ADD    E
019F 67        MOV    H,A
01A0 7C        NOADD: MOV   A,H
01A1 1F        RAR
01A2 67        MOV    H,A
01A3 7D        MOV    A,L
01A4 1F        RAR
01A5 6F        MOV    L,A
01A6 0B        DCR    C
01A7 C29701    JNZ   NXTBIT
01AA C9        RET
01AB DB12      TTYIN: IN    P1    ;CHECK STATUS OF INPUT PORT 2
01AD 1F        RAR
01AE D2AB01    JNC   TTYIN ;ANY INCOMING DATA?
                ;NO--KEEP CHECKING
01B1 DB13      IN    P2    ;A CHARACTER IS RECEIVED--READ IT
01B3 E67F      ANI    7FH   ;MASK OUT THE PARITY BIT
01B5 C9        RET
                ;RETURN
                ;
                ;THIS SUBROUTINE OUTPUTS THE DESIRED
                ;CHARACTERS ON THE PRINTER
                ;
01B6 5F        TTYOUT: MOV   E,A ;MOVE THE CHARACTER TO BE PRINTED
                ; INTO THE E-REGISTER
01B7 DB10      TTYO:  IN    P3    ;CHECK THE PRINTER STATUS--OUTPUT PORT 1
01B9 E602      ANI    02H  ;MASK ALL BUT THE TRANSMIT DATA READY BIT
01BB FE02      CPI    02H  ;IS THAT BIT A LOGIC 1?
01BD C2B701    JNZ   TTYO  ;NO--SO
01C0 DBB701    JC     TTYO  ;TRY AGAIN
01C3 7B        MOV    A,E ;LOAD THE CHARACTER TO BE PRINTED
                ; INTO THE ACCUMULATOR
01C4 D311      OUT    P4    ;PRINT THE CHARACTER
01C6 C9        RET
                ;RETURN
0131 C21301    JNZ   SEARCH ;IS PRINTED--JUMP TO WAIT FOR ANOTHER CHARACTER
0134 79        MOV    A,C ;MOVE THE LATEST CHARACTER RECEIVED INTO
                ; THE ACCUMULATOR
                ;
0135 FE1E      CPI    1EH   ;IS IT AN 'RS'?
0137 DA3F01    JC     GO    ;IT IS NOT AN 'RS'--JUMP
013A C23F01    JNZ   GO    ;TO THE OUTPUT SUBROUTINE
013D 3E00      MVI    A,00H ;IT IS AN 'RS'--CHANGE IT TO A NULL CHARACTER
013F CDB601    GO:   CALL  TTYOUT ;PRINT THE CHARACTER
0142 C31301    SEARCH: JMP   SEARCH ;JUMP TO SEARCH FOR ANOTHER CHARACTER
0145 3E01      SOH:  MVI    A,01H ;LOAD THE 'SOH' CHARACTER
0147 89        CMP    C    ;IS THE CHARACTER RECEIVED AN 'SOH'?
0148 DA1301    JC     SEARCH ;IT IS NOT AN 'SOH'--JUMP BACK
014B C21301    SEARCH: JMP   SEARCH ;AND SEARCH FOR ANOTHER CHARACTER
                ;
                ;AN 'SOH' HAS BEEN RECEIVED--SKIP THE NEXT 5 CHARACTERS
                ;
014E CDAB01    CALL  TTYIN
0151 CDAB01    CALL  TTYIN
0154 CDAB01    CALL  TTYIN
0157 CDAB01    CALL  TTYIN
015A CDAB01    CALL  TTYIN
                ;
                ;THIS SECTION OF THE PROGRAM READS EACH OF THE THREE
                ;IDENTIFYING CHARACTERS, CONVERTS THEM TO NUMERIC
                ;FORMAT. THIS SECTION ALSO SUBTRACTS 200D FROM THE
                ;ID #'S SO THAT THEY CAN BE PACKED INTO AN 8-BIT WORD.
015D CDAB01    CALL  TTYIN ;READ THE FIRST CHARACTER OF THE ID # (THE
                ; HUNDREDS' DIGIT)
0160 E60F      ANI    0FH   ;MASK OUT THE FOUR HIGH-ORDER BITS
0162 DE02      SBI    02H  ;THIS SUBTRACTS 200D
0164 57        MOV    D,A ;TAKE THAT DIGIT AND MOVE IT INTO THE D-REGISTER
0165 1E64      MVI    E,100D ;LOAD 100D INTO THE E-REGISTER
0167 CD9201    CALL  MP88  ;MULTIPLY THE TWO
016A 54        MOV    D,H ;MOVE THE RESULT FROM THE
016B 5D        MOV    E,L ;H-PAIR TO THE D-PAIR
016C DS        PUSH  D    ;AND PUSH IT ON TO THE STACK
016D CDAB01    CALL  TTYIN ;READ THE SECOND CHARACTER OF THE ID # (THE TENS'
                ; DIGIT)

```