SERIAL ROUTINES FOR THE TEC

RECEIVER

This is the routine I use when I wish to down-load a file from the IBM. It's a simple routine that converts a serial stream into bytes and stores them in RAM starting at the address provided at 0898. The routine also has an end address to allow a maximum file length. This is in case something goes wrong with the data transfer. Anything important can be protected by placing it above the end address.

No hand-shaking is needed as the TEC can cope with the speed of the data stream. It is up to you to ensure the TEC is ready before you send the data. The serial input is bit 0 of PORT 3. The DAT BOARD has provision for 2 diodes and a resistor at this input to clip an incoming RS232 signal. In the RS232 format, a logic 1 is represented by a negative voltage while a logic 0 is a positive voltage. The clipper on the DAT BOARD changes an RS232 logic 0 (positive voltage) into a digital logic 1 while an RS232 logic 1 is clipped to zero volts and becomes a digital logic 0.

This means that the inputted data must be inverted back into its true form. This is done with the CPL instruction at 092C. The format of the data is as follows: 2400 BAUD, NO PARITY, 8 BITS, STOP BITS OPTIONAL, TEC SPEED: 3.58/2

SERIAL INPUT ROUTINE

```
0900  2A 98 08     LD HL,(0898)   START ADDRESS IN HL
0903  CD 12 09     CALL 0912      GET BYTE
0906  ED 4B 9A 08  LD BC,(089A)   PUT END ADDR IN BC
090A  B7           OR A           CLEAR CARRY FLAG
090B  E5           PUSH HL        SAVE HL
090C  ED 42        SBC HL,BC      SUB CURRENT ADDR FROM END
090E  E1           POP HL         RECOVER HL
090F  38 F2        JR C 0903      JUMP IF NOT DONE
0911  C9           RET            ELSE RETURN TO JMON
0912  DB 03        IN A,03        LOOK FOR START BIT
0914  07           RLCA           PUT IN CARRY
0915  30 FB        JR NC 0912     LOOP UNTIL START BIT FOUND
0917  06 40        LD B,40        DELAY TO HALF WAY IN
0919  10 FE        DJNZ 0919      FIRST CELL
091B  1E 00        LD E,00        E IS RECEIVER BYTE
091D  0E 08        LD C,08        C IS COUNT SET FOR 8 BITS
091F  DB 07        IN A,07        INPUT BIT
0921  07           RLCA           INTO CARRY FLAG
0922  CB 1B        RR E           THEN STORE IN E
0924  06 39        LD B,39        B=HALF CELL DELAY
0926  10 FE        DJNZ 0926      LOOP TO NEXT CELL ARRIVES
0928  0D           DEC C          DEC LOOP COUNTER
0929  20 F4        JR NZ 091F     JUMP IF 8 BITS NOT DONE
092B  7B           LD A,E         PUT INPUTTED BYTE IN A
092C  2F           CPL            AND INVERT TO TRUE FORM
092D  77           LD (HL),A      STORE IN MEMORY
092E  23           INC HL         INCREASE MEMORY POINTERS
092F  C9           RET            RETURN FROM SUBROUTINE
```

SERIAL OUTPUT ROUTINE

This is the complement routine of the serial receiver. It will send serial data through the TEC speaker bit. The data is taken from the latch side of the base resistor of the transistor inverter and inputted directly to an RS232 Rx input or the DAT BOARD serial input.

Strictly speaking the data stream is not RS232 compatible but in practice it works ok, although the occasional error may creep in.

Oh yes, before sending data, the key press beep must be turned off. To do this, place FF at 0822 and put AA at 08FF.

The serial sender uses the same start and end buffers as the receive described above with the same speed etc. Two stop bits are sent as this provides compatibility with all serial systems.

SERIAL OUTPUT ROUTINE

```
0A00  2A 98 08     LD HL,(0898)   PUT START IN HL
0A03  CD 12 0A     CALL 0A12      OUTPUT BYTE
0A06  ED 4B 9A 08  LD BC (089A)   END IN BC
0A0A  B7           OR A           CLEAR CARRY
0A0B  E5           PUSH HL        SAVE START
0A0C  ED 42        SBC HL,BC      SUB END FROM START
0A0E  E1           POP HL         RECOVER START
0A0F  38 F2        JR C,0A03      JUMP IF END << THAN
0A11  C9           RET            START ELSE RETURN
0A12  3E 80        LD A,80        SET START BIT
0A14  D3 01        OUT (01),A     OUT START BIT
0A16  CD 2D 0A     CALL 0A2D      CALL DELAY
0A19  7E           LD A,(HL)      GET BYTE TO OUTPUT
0A1A  23           INC HL         POINT TO NEXT BYTE
0A1B  06 08        LD B,08        SET COUNT FOR 8 BITS
0A1D  0F           RRCA           PUT BIT INTO BIT 7
0A1E  EE 80        XOR 80         COMPLEMENT BIT
0A20  D3 01        OUT (01),A     OUTPUT IT
0A22  CD 2D 0A     CALL 0A2D      CALL DELAY
0A25  10 F6        DJNZ,0A1D      DO FOR 8 BITS
0A27  AF           XOR A          CLEAR FOR STOP BIT
0A28  D3 01        OUT (01),A     OUT STOP BIT x2
0A2A  CD 2D 0A     CALL 0A2D      FIRST STOP DELAY
0A2D  C5           PUSH BC        SAVE BIT COUNT
0A2E  06 36        LD B,36        LOAD B WITH DELAY
0A30  10 FE        DJNZ,0A30      DO DELAY
0A32  C1           POP BC         RECOVER BIT COUNT
0A33  C9           RET            DONE
```

IBM SOFTWARE

The software I used for receiving the serial is PROCOMM. It is a public domain program and can be purchased from the Talking Electronics Shop. Cat S-449.

The protocol to use is ASCII.

The sending software poses a few difficulties. One big problem is that some packages won't send the 1A character. Actually, I believe the problem is in the DOS serial interrupt and if the software uses it then it won't send the 1A character.

It is rare that I send anything back to the TEC and when I do, it's with a serial routine Craig wrote and probably won't work with all computers as it directly manipulates the hardware; not a recommended practice.

It is up to you to experiment around and find something that works.

I would like to hear from anyone who has found or written a good sending routine that doesn't have the 1A character problem.

Hardware wise, the CTS must be taken high before the IBM will send the data. This means that the IBM to TEC link consists of three wires: the ground, the serial data line and +5v.

Only ground and the serial data are required for the TEC to IBM link.