# FSK/RTTY DECODER FOR PCs

**The circuit and the software described here are aimed at those of you who have so far dreaded the complexity of a full-blown FSK decoder and the well-presented objections of the Miss or Missus about the weight, size and noise of a good-as-new teleprinter machine (no matter how cheaply you may have acquired this wonderful equipment), when all you want to do is receive RTTY (telex) transmissions in the short-wave bands.**

**by Roger Collins**

KEEPING the peace at home and still be able to intercept FSK (frequency-shift keying) transmissions requires some hardware to be built or purchased that changes the output of a short-wave receiver into a form that is suitable for processing by a personal computer (PC). The decoder presented here does everything to achieve just that. In combination with a simple BASIC program (Fig. 2), it turns your IBM PC (or compatible) into an RTTY decoder capable of handling different types of FSK, different baud rates, and different mark/space tone conventions. The method used to accomplish this is fairly rudimentary, and intended as a guide for further experimenting.

## FSK techniques

Much transmission of data, whether news broadcasts, weather information or amateur traffic, over a radio network employs the principle of frequency shift keying (FSK). The data to be sent is in the form of logic 1s and 0s. This stream of data is used to shift the frequency of the transmitter, resulting in two discrete frequencies being radiated just like an SSB signal modulated by two (alternate) tones. A high transmit frequency denotes a 1 (or mark), and a low transmit frequency a 0 (or space). The two tones and the difference in frequency shift vary depending on the standard used.

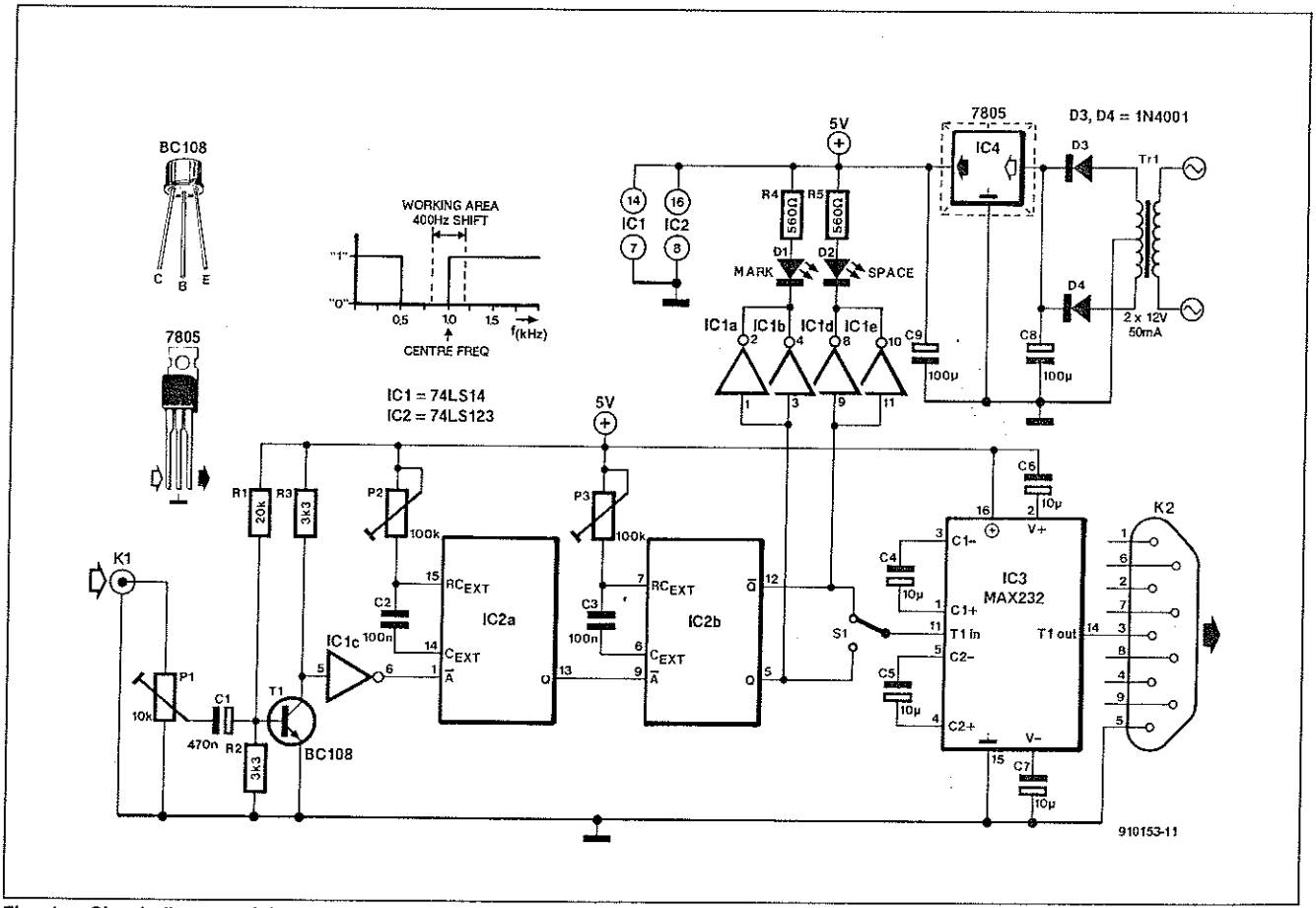


Fig. 1. Circuit diagram of the FSK decoder. A MAX232 is used to ensure the correct signal levels for the PC's serial port, COM2:.

A databyte can be sent asynchronously if it is preceded by a start bit to enable the terminal equipment to get ready to receive it. Likewise, one or two stop bits are used to enable the terminal equipment to shift the newly received databyte out, and prepare for the next start bit, which signals the arrival of a new databyte.

Since a byte is eight bits, a complete dataword would produce a packet of 10 or 11 bits. To reduce the number of bits, and with it the bandwidth occupied by the transmitter, the length of the dataword is reduced to seven and a half bits — five databits, one start bit, and a one and a half stop bit. However, five bits of data will only produce 32 ($2^5$) combinations. Assuming that plain language is used for the transmission, the 32 codes available allow the complete alphabet to be sent. In the RTTY (radioteletype) system, one of the codes is reserved to indicate 'figure shift', which offers another set of 32 codes that may be used for numbers and punctuation marks. In this set, there is a 'letter shift' code that returns the equipment to the alphabet. The code used is based on the Murry, or more frequently, the Baudot, convention.

To keep the bandwidth of the transmission in the short-wave band as small as possible, the transmission (data-) rate must be kept within limits. The normal speed in terms of bits transmitted per second (baud rate) is 45 to 75. At 50 baud, one bit of data has a length of 20 ms. Using a tone of 1 kHz to indicate a 1 would mean 20 cycles of the tone being transmitted.

At the receiver, a BFO (beat frequency oscillator) enables the two tones to be converted to any frequency within the audio pass-band, i.e., they may not necessarily be the exact original two tones. In the FSK demodulator, filters and phase-locked loop techniques are used to convert these two tones into the marks and spaces (1s and 0s) of the original transmitted data.

## FSK decoding on a PC

MS-DOS as well as most communication and terminal emulation programs developed for PCs will allow the baud rate of a serial port (COM1: to COM4:) to be set only as low as 110 baud, and the data format to 7 or 8 bits, with 1 or 2 stop bits. For the reception of FSK data we require to set the baud rate as low as 45, with 5 databits and 1½ stop bit. Quite an unusual format for the average PC user!

Most PCs use a 8250 UART or similar IC in the serial interfaces COM1: and COM2:. The COM1: and COM2: base addresses are 03FB and 02FB respectively. The register functions of the 8250 are listed in Table 1.

For the present application, COM2: is used, and the BASIC program has been written to use this port for the RTTY decoder. The baud rate is sent to the UART as two bytes (high and low). Testing LSR bit 0 will indicate if data has been received in the RDR. Next, the RDR is read, the content is converted to ASCII, a check is made on letter shift or figure shift, and the converted char-

**Table 1. 8250 UART register functions**

| address | LCR bit 7 | Function |
|---|---|---|
| base + 0 | 0 | Tx holding reg. (THR) (write)<br>Rx data reg. (RDR) (read) |
| base + 0 | 1 | baud rate divisor low (BRDL) |
| base + 1 | 1 | baud rate divisor high (BRDH) |
| base + 1 | 0 | interrupt enable reg. (IER) |
| base + 2 | x | interupt ID reg. (IIR) |
| base + 3 | x | line control reg. (LCR) |
| base + 4 | x | modem control reg. (MCR) |
| base + 5 | x | line status reg. (LSR) |
| base + 6 | x | modem status reg. (MSR) |

x = don't care.

**Table 2. Teletypewriter codes**

| CHARACTER | | | CODE SIGNALS | | | | | | | ASCII | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LTRS | FIGS | START | 1 | 2 | 3 | 4 | 5 | STOP | CODE | LTRS | FIGS |
| A | – | | 1 | 1 | 0 | 0 | 0 | | 3 | 65 | 45 |
| B | ? | | 1 | 0 | 0 | 1 | 1 | | 25 | 66 | 63 |
| C | : | | 0 | 1 | 1 | 1 | 0 | | 14 | 67 | 58 |
| D | $ | | 1 | 0 | 0 | 1 | 0 | | 9 | 68 | 36 |
| E | 3 | | 1 | 0 | 0 | 0 | 0 | | 1 | 69 | 51 |
| F | > | | 1 | 0 | 1 | 1 | 0 | | 13 | 70 | 62 |
| G | * | | 0 | 1 | 0 | 1 | 1 | | 26 | 71 | 42 |
| H | < | | 0 | 0 | 1 | 0 | 1 | | 20 | 72 | 60 |
| I | 8 | | 0 | 1 | 1 | 0 | 0 | | 6 | 73 | 56 |
| J | bell | | 1 | 1 | 0 | 1 | 0 | | 11 | 74 | 07 |
| K | ( | | 1 | 1 | 1 | 1 | 0 | | 15 | 75 | 40 |
| L | ) | | 0 | 1 | 0 | 0 | 1 | | 18 | 76 | 41 |
| M | . | | 0 | 0 | 1 | 1 | 1 | | 28 | 77 | 46 |
| N | , | | 0 | 0 | 1 | 1 | 0 | | 12 | 78 | 44 |
| O | 9 | | 0 | 0 | 0 | 1 | 1 | | 24 | 79 | 57 |
| P | 0 | | 0 | 1 | 1 | 0 | 1 | | 22 | 80 | 48 |
| Q | 1 | | 1 | 1 | 1 | 0 | 1 | | 23 | 81 | 49 |
| R | 4 | | 0 | 1 | 0 | 1 | 0 | | 10 | 82 | 52 |
| S | ' | | 1 | 0 | 1 | 0 | 0 | | 5 | 83 | 39 |
| T | 5 | | 0 | 0 | 0 | 0 | 1 | | 16 | 84 | 53 |
| U | 7 | | 1 | 1 | 1 | 0 | 0 | | 7 | 85 | 55 |
| V | = | | 0 | 1 | 1 | 1 | 1 | | 30 | 86 | 61 |
| W | 2 | | 1 | 1 | 0 | 0 | 1 | | 19 | 87 | 50 |
| X | / | | 1 | 0 | 1 | 1 | 1 | | 29 | 88 | 47 |
| Y | 6 | | 1 | 0 | 1 | 0 | 1 | | 21 | 89 | 54 |
| Z | + | | 1 | 0 | 0 | 0 | 1 | | 17 | 90 | 43 |
| BLANK | | | 0 | 0 | 0 | 0 | 0 | | 0 | 00 | 00 |
| SPACE | | | 0 | 0 | 1 | 0 | 0 | | 4 | 32 | 32 |
| CR | | | 0 | 0 | 0 | 1 | 0 | | 8 | 13 | 13 |
| LF | | | 0 | 1 | 0 | 0 | 0 | | 2 | 10 | 10 |
| FIGS | | | 1 | 1 | 0 | 1 | 1 | | 27 | 00 | 00 |
| LTRS | | | 1 | 1 | 1 | 1 | 1 | | 31 | 00 | 00 |

```
10  '   ********************************************
20  '   *                                          *
30  '   *              FSK DECODER PROGRAM         *
40  '   *                                          *
50  '   *          L - LTRS ARRAY                  *
60  '   *          F - FIGS ARRAY                  *
70  '   *          BRD - BAUD RATE DIVISOR         *
80  '   *          G - 1/0 FIGS/LTRS               *
90  '   *          &H2F8 - COMM 2 RDR/BRDL         *
100 '   *          &H2F9 - COMM 2 IER/BRDH         *
110 '   *          &H2FB - COMM 2 LCR              *
120 '   *          &H2FD - COMM 2 LSR              *
130 '   ********************************************
140 '
150 CLS
160 PRINT
170 PRINT "   FREQ            BAUD RATE "
180 PRINT "   ----            --------- "
190 PRINT
195 PRINT "  20.967             50          TASS "
200 PRINT "  18.404             50          TASS "
210 PRINT "  18.194             50          TASS "
220 PRINT "  18.439             50              "
230 PRINT "  18.049             50          FRENCH "
240 PRINT "  15.937             50              "
250 PRINT "  13.490             50          FRENCH "
260 PRINT "   8.064             50          FRENCH "
270 PRINT "   7.594             50          BELGRADE "
280 PRINT
290 PRINT
300       INPUT "Enter baud rate  ",B
310       BRD=1843200!/(16*B)
320       BDRL=BRD AND &HFF
330       BRDH=(BRD AND &HFF00)/256
340 CLS
350 DIM L(31),F(31)
360 FOR J=0 TO 31
370 READ L(J)
380 NEXT J
390 FOR J=0 TO 31
400 READ F(J)
410 NEXT J
420 OUT &H2FB,&H84
430 OUT &H2F8,BRDL
440 OUT &H2F9,BRDH
450 OUT &H2FB,&H4
460 S=INP (&H2FD)
470   K=S AND 1
480   IF K=0 THEN GOTO 460
490 A=INP (&H2F8)
500 IF A=27 THEN G=1
510 IF A=31 THEN G=0
520 IF G=0 THEN PRINT CHR$(L(A)); ELSE PRINT CHR$(F(A));
530         OUT &H2FD,0
540 GOTO 460
550 DATA 00,69,10,65,32,83,73,85,13,68,82,74,78,70,67,75,84,90,76,87,72,89,
80,81,79,66,71,00,77,88,86,00
560 DATA 00,51,10,45,32,39,56,55,13,36,52,07,44,62,58,40,53,43,41,50,60,54,
48,49,57,63,42,00,46,47,61,00
                                                    910153-12
```

**Fig. 2. Listing of the control program written in BASIC.**

acter is sent to the video adapter. LSR bit 0 is then cleared, and the software waits for the next character by testing this bit.

## A simple FSK decoder

The circuit diagram of the FSK decoder hardware is given in Fig. 1. The audio output of the receiver is applied to connector K1, and the preset level of P1 is adjusted to give a squared-up signal at the collector of T1. Provided the receiver has been tuned correctly to the FSK signal, the rectangular signal supplied by T1 will be the digital version of the two tones.

The two signal frequencies are fed to IC2a via a Schmitt trigger, IC1c. IC2a is a retriggerable monostable multivibrator set to a mono time of about 1 ms with preset P2. This means that signals higher than 1 kHz will cause this mono to be permanently set. Any frequency lower than 1 kHz, for instance, the lower FSK tone, will cause the mono to set and clear. This signal is fed to a second monostable, IC2b, set to 2 ms with P3. This monostable will be set for the periods of the signals, supplied by IC2a, shorter than 2 ms, i.e., for frequencies higher than 500 Hz. The result is that frequencies between 500 Hz and 1 kHz will produce a logic 0, and frequencies higher than 1 kHz a logic 1, with a sharp transition at this centre frequency.

If the receiver is tuned so that the high tone and the low tone (mark and space) are equidistant around the centre frequency, the mark and space signals will produce 1s and 0s, depending on the received transmission.

The Q and $\overline{Q}$ signals are used to drive the mark and space LEDs to assist tuning to the FSK transmission. As there are as many marks as spaces, the two LEDs will flash at the same rate when the tuning is correct. These signals are also fed to the phase reversing switch, S1, and from there to a Type MAX-232 RS232 driver, IC3. This IC has an on-board DC-DC converter providing +10 V and −10 V rails to ensure the correct swing of the RS232 signal required by the serial port of the PC.

Correct setting-up of the serial port, the baud rate, the number of databits and stop bits will result in the interception of the RTTY transmission, which will de displayed on your computer monitor. Converting the 5-bit code into ASCII will enable the characters to be displayed.

Presets are sufficient here since the final adjustment of the audio signal can be done at the receiver. The phase switch, S1, will be called upon occasionally when the marks and spaces are swapped as a result of tuning the receiver above or below the tones (USB or LSB).

Finally, the decoder is powered by a simple 5-V supply connected to the mains. Although its current drain is small, no attempt should be made at powering the decoder from the modem signals on the PC serial port lines. ■

**Table 3. Decoder to PC connections**

| Decoder (D9) | PC (D9) | PC (D25) | Signal |
|---|---|---|---|
| 5 | 5 | 7 | ground |
| 3 | 2 | 3 | RxD |