…and couldn't find anywhere, you will find in this Elektor article (and where else would you find it?). It's a natural idea to use a PC keyboard for developing microcomputer applications, for example, in order to send commands or respond to specific actions. Why go to the trouble of building your own keyboard when you can use a ready-made (and inexpensive) PC keyboard? The only problem is that you first have to know exactly what signals a PC keyboard supplies.

By F. Wohlrabe

# PC keyboard encoding

## Everything you ever wanted to know about the signals from a PC keyboard...

In addition to the ready availability, low cost and accustomed manner of use of a PC keyboard, connecting a PC keyboard directly to a microcontroller system has the advantage that it makes valuable port pins available that otherwise would be used for polling a keyboard built from individual components. A PC keyboard, by contrast, produces a serial signal, and is thus an ideal complement to a microcontroller project. Of course, the manner in which the signal from the PC keyboard is constructed has a few special features. Two lines are used for the serial data transfer. One of these, labelled data, transfers the data, while the second one transfers the clock. The serial data transfer protocol, which is fairly complex, is explained below.

## Key codes

The most widely-used type of keyboard is the MF2 model ('multi-functional version 2'). It was originally developed by IBM for computers in the XT, AT and PS/2 series. This model has become an industry standard in the meantime, and almost all PCs are equipped with it. The keyboard itself contains a 'keyboard controller', which generates the key

codes and provides for communication with the keyboard interface of the PC. The keyboard controller is usually a mask-programmed microcontroller. Data are sent and received according to the IBM protocol. Commands can be used to control the LEDs, specify the delay and rate for key repetition, and select the scan code set. The MF2 keyboard has three different scan code sets. Set 1 is used by XT/PC and PS/2-30 compatible computers, while set 2 is used by AT computers and all other PS/2-compatible computers. Set 3 supports workstations and terminal emulation on the PC. Country-specific keyboard drivers in the operating system translate each key press into the desired character.
When a key is pressed, the keyboard produces a Make code. This code corresponds to the scan code for that key. The repeat function causes the Make code to be continuously repeated if the key is held down long enough. The delay time and repetition rate of the repeat function are programmable.
When a key is released, the keyboard produces a Break code. However, if scan code set 3 is selected, no Break code is generated and the repeat function is disabled. After a reset, the keyboard selects scan code set 2 as a default.

You should bear in mind that a PC/XT keyboard cannot be programmed, since its internal controller cannot accept data. Only with the introduction of the AT computer did the keyboard become more user-friendly, since the behaviour of an AT keyboard can be adapted to the needs of the user via software. The following information relates to a keyboard that is set to operate in the AT mode.

## Sending and receiving

**Figure 1** shows the pin assignments of the two commonly-used keyboard plugs (the 5-pin DIN plug and the 6-pin PS/2 plug). The keyboard is powered with 5 V from the PC. Its maximum current consumption is around 200 mA.
In general, the clock rate is set by the keyboard. It lies in the range of 10 to 16.7 kHz. Data are sent using a start bit (always 0), eight data bits with bit 0 first, an odd-parity bit and a stop bit (always 1). **Figure 2** shows the data transfer timing diagram.
If an external device or system (which is normally a PC) wants to send data to the keyboard, the keyboard recognizes this by the fact that the data line is pulled to earth by the external device

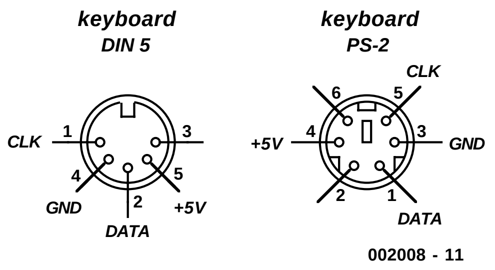

Figure 1. Pin assignments of the standard PC keyboard connectors (viewed from the front).

(the PC). The keyboard responds by sending the clock signal, and it expects the data to be sent synchronized to the clock signal. After the data transfer, the data line must exhibit a High level, which acts as the stop bit. The keyboard will continue to send the clock until this condition is satisfied. Following this, the command $FE_H$ is sent to request a new data packet. Data are accepted on the rising edge of the clock signal. After the stop bit has been detected, the keyboard controller holds the data line Low for the duration of one bit period. The keyboard answers every command that it receives, within at most 20 ms, by sending the byte $FA_H$ (ACK), except for the ECHO and RESET commands.
The keyboard sends data in the AT format to an external device using the following process.
Before sending data, the keyboard controller first tests whether the clock or data line is at earth level. Communications can be blocked by holding the clock line Low. In this case, the keyboard holds the data to be sent in an internal buffer. The keyboard can send data only if both the clock and data lines are at a High level. It then sets the data line Low (for the start bit) and generates the clock signal. The data are valid on the falling edge of the clock signal and change after the rising edge.

In order to implement a MF2 keyboard connection in a microcomputer system, you will also need to know certain information regarding the most important commands and return codes for an AT keyboard. These are described in the following section. The key codes, which are the codes that the keyboard produces according to the selected scan-code set when keys are pressed, are listed in **Table 1**.

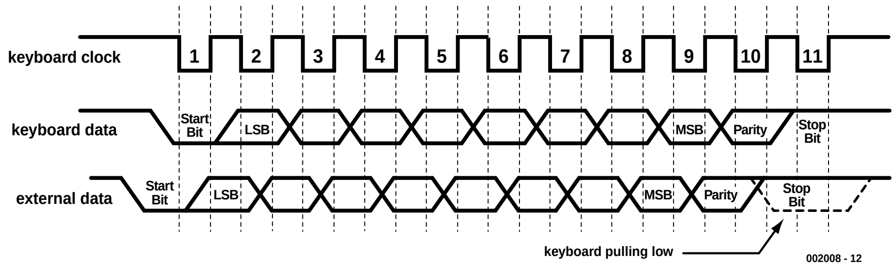

Figure 2. Timing diagram for serial data transfers between the keyboard and a PC.

### The most important commands

**SET/RESET MODE INDICATORS → code $ED_H$**
This two-byte command controls the behaviour of the LEDs.
Command: $ED_H$
Command: 0000 0xxx
Bit 0: Scroll lock
Bit 1: Num lock
Bit 2: Caps lock
1 = LED on, 0 = LED off

**ECHO → code $EE_H$**
The keyboard answers this command with $EE_H$. It can be used to confirm the presence of a keyboard.

**SCAN CODES SELECT → code $F0_H$**
This two-byte command selects the scan code set. Scan code set 2 is selected by default after a reset. However, scan code set 3 recommends itself for microcontroller applications, due to its simplicity. With scan code set 3, no Break code is sent for almost all keys and the repeat function is disabled.
Command: $FO_H$
Command: 0000 00xx
01 = scan code set 1
10 = scan code set 2
11 = scan code set 3

**READING ID CODE → code $F2_H$**
In response to this command, the keyboard sends three bytes which contain a manufacturer-specific code.
1st byte = $FA_H$ (ACK)
2nd byte = xxxx xxxx
3rd byte = xxxx xxxx

**SET TYPEMATIC RATE/DELAY → code $F3_H$**
This two-byte command controls the key repeat rate and the delay for starting key repetition.
Command: $F3_H$
Command: 0xxx xxxx
Bits 5 and 6 control the delay, which ranges from 150 ms to 1 s.

| Bit 6 | Bit 5 | Delay (± 20%) |
|---|---|---|
| 0 | 0 | 150 ms |
| 0 | 1 | 500 ms |
| 1 | 0 | 750 ms |
| 1 | 1 | 1 s |

Bits 0 through 4 control the repetition rate, which ranges from 2 to 30 Hz. In the following table, only three values are shown as examples.

| Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Frequency (± 20%) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 30 Hz |
| 0 | 1 | 1 | 1 | 1 | 8 Hz |
| 1 | 1 | 1 | 1 | 1 | 2 Hz |

**SET ALL KEYS → codes $F7_H$, $F8_H$, $F9_H$, $FA_H$**
These commands assign attributes to the keys, as follows:
- $F7_H$: all keys have the repeat function
- $F8_H$: all keys produce Make and Break codes
- $F9_H$: all keys produce only a Make code
- $FA_H$: all keys have the repeat function and produce Make and Break codes

**RESET → code $FF_H$**
This command restores all keyboard settings to their default values.

## The most important return codes

**BAT COMPLETION → code $AA_H$**
This byte is sent to the external system after the supply voltage has been applied or a reset command ($FF_H$) has been recognized. In indicates correct execution of the keyboard self-test.

**RESEND NAK → code $FE_H$**
This byte is sent in response to a data transfer error.

**ACK → code $FA_H$**
This byte is sent to the external device in response to each received command.

**OVERRUN → code $00_H$/$FF_H$**
All key presses are stored internally in the keyboard until their codes can be serially transferred to the external device. If the storage buffer overflows, the byte $00_H$ is sent for scan code sets 2 and 3, while the byte $FF_H$ is sent for scan code set 1.

**BREAK CODE PREFIX → code $F0_H$**
With scan code set 2, the byte $F0_H$ is sent before the Break code.

## Programming example

In conclusion, the manner in which a (microcontroller) system should address an AT keyboard can be illustrated with a simple programming example.
When the 5-V supply voltage is switched on, +5 V is applied to the keyboard. The keyboard controller in the keyboard then executes a self-test. If this is completed successfully, the keyboard sends the byte $AA_H$.

Next comes the selection of the scan code set. In this example, scan code set 3 is selected using the SCAN CODES SELECT command, as follows:

1 Pull the data line to earth.
2 Send the command code $F0_H$, synchronous to the clock.
3 The keyboard sends the code $FA_H$ (ACK) as confirmation.
4 Pull the data line to earth.
5 Send the command code $03_H$, synchronous to the clock.
6 The keyboard sends the code $FA_H$ (ACK) as confirmation.

Now a key can be pressed, and the key code from the scan code 3 set (see Table 1) will be received:

7 Press ‘G’ on the keyboard.
8 The keyboard sends the code $34_H$.

A practical application example of the use of a PC keyboard with a microcontroller system is “Text Running Line Display” elsewhere in this Supplement. In this example, a common or variety PC keyboard is used for entering text to be displayed on an LED running-line display. Keyboard decoding is handled by a COP-8 microcontroller, and the data transfer to the running-line display uses an infrared link. The photo at the head of this article shows a small circuit board holding the keyboard decoder and IR transmitter.

(002008-1)

**Table 1.**
**The key codes produced when keys are pressed, for each of the three scan code sets.**

| Symbol | Scan code set 1 | | Scan code set 2 | | Scan code set 3 | |
|---|---|---|---|---|---|---|
| | Make | Break | Make | Break | Code | Type |
| ^ | 29 | A9 | 0E | F0-0E | 0E | T |
| 1 | 02 | 82 | 16 | F0-16 | 16 | T |
| 2 | 03 | 83 | 1E | F0-1E | 1E | T |
| 3 | 04 | 84 | 26 | F0-26 | 26 | T |
| 4 | 05 | 85 | 25 | F0-25 | 25 | T |
| 5 | 06 | 86 | 2E | F0-2E | 2E | T |
| 6 | 07 | 87 | 36 | F0-36 | 36 | T |
| 7 | 08 | 88 | 3D | F0-3D | 3D | T |
| 8 | 09 | 89 | 3E | F0-3E | 3E | T |
| 9 | 0A | 8A | 46 | F0-46 | 46 | T |
| 0 | 0B | 8B | 45 | F0-45 | 45 | T |
| ß | 0C | 8C | 4E | F0-4E | 4E | T |
| ‘ | 0D | 8D | 55 | F0-55 | 55 | T |
| ← Back | 0E | 8E | 66 | F0-66 | 66 | T |
| \|← →\| Tab | 0F | 8F | 0D | F0-0D | 0D | T |
| Q | 10 | 90 | 15 | F0-15 | 15 | T |
| W | 11 | 91 | 1D | F0-1D | 1D | T |
| E | 12 | 92 | 24 | F0-24 | 24 | T |
| R | 13 | 93 | 2D | F0-2D | 2D | T |
| T | 14 | 94 | 2C | F0-2C | 2C | T |
| Z | 15 | 95 | 35 | F0-35 | 35 | T |
| U | 16 | 96 | 3C | F0-3C | 3C | T |
| I | 17 | 97 | 43 | F0-43 | 43 | T |
| O | 18 | 98 | 44 | F0-44 | 44 | T |
| P | 19 | 99 | 4D | F0-4D | 4D | T |
| Ü | 1A | 9A | 54 | F0-54 | 54 | T |
| + | 1B | 9B | 5B | F0-5B | 5B | T |
| CapsLock | 3A | BA | 58 | F0-58 | 14 | M,B |
| A | 1E | 9E | 1C | F0-1C | 1C | T |
| S | 1F | 9F | 1B | F0-1B | 1B | T |
| D | 20 | A0 | 23 | F0-23 | 23 | T |
| F | 21 | A1 | 2B | F0-2B | 2B | T |
| G | 22 | A2 | 34 | F0-34 | 34 | T |
| H | 23 | A3 | 33 | F0-33 | 33 | T |
| J | 24 | A4 | 3B | F0-3B | 3B | T |
| K | 25 | A5 | 42 | F0-42 | 42 | T |
| L | 26 | A6 | 4B | F0-4B | 4B | T |
| Ö | 27 | A7 | 4C | F0-4C | 4C | T |
| Ä | 28 | A8 | 52 | F0-52 | 52 | T |
| # | 2B | AB | 5D | F0-5D | 53 | T |
| Return | 1C | 9C | 5A | F0-5A | 5A | T |
| Shift l. | 2A | AA | 12 | F0-12 | 12 | M,B |
| < | 56 | D6 | 61 | F0-61 | 13 | T |
| Y | 2C | AC | 1A | F0-1A | 1A | T |
| X | 2D | AD | 22 | F0-22 | 22 | T |
| C | 2E | AE | 21 | F0-21 | 21 | T |
| V | 2F | AF | 2A | F0-2A | 2A | T |
| B | 30 | B0 | 32 | F0-32 | 32 | T |
| N | 31 | B1 | 31 | F0-31 | 31 | T |
| M | 32 | B2 | 3A | F0-3A | 3A | T |
| , | 33 | B3 | 41 | F0-41 | 41 | T |
| . | 34 | B4 | 49 | F0-49 | 49 | T |
| - | 35 | B5 | 4A | F0-4A | 4A | T |
| Shift r. | 36 | B6 | 59 | F0-59 | 59 | M,B |
| Ctrl l. | 1D | 9D | 14 | F0-14 | 11 | M,B |
| Alt l. | 38 | B8 | 11 | F0-11 | 19 | M,B |
| Space | 39 | B9 | 29 | F0-29 | 29 | T |
| Num | 45 | C5 | 77 | F0-77 | 76 | M |
| 7 Nb | 47 | C7 | 6C | F0-6C | 6C | M |
| 4 Nb | 4B | CB | 6B | F0-6B | 6B | M |
| 1 Nb | 4F | CF | 69 | F0-69 | 69 | M |
| / Nb | E0-35 | E0-B5 | E0-4A | E0-F0-4A | 77 | M |
| 8 Nb | 48 | C8 | 75 | F0-75 | 75 | M |
| 5 Nb | 4C | CC | 73 | F0-73 | 73 | M |
| 2 Nb | 50 | D0 | 72 | F0-72 | 72 | M |
| 0 Nb | 52 | D2 | 70 | F0-70 | 70 | M |
| * Nb | 37 | B7 | 7C | F0-7C | 7E | M |
| 9 Nb | 49 | C9 | 7D | F0-7D | 7D | M |
| 6 Nb | 4D | CD | 74 | F0-74 | 74 | M |
| 3 Nb | 51 | D1 | 7A | F0-7A | 7A | M |
| Del Nb | 53 | D3 | 71 | F0-71 | 71 | M |
| - Nb | 4A | CA | 7B | F0-7B | 84 | M |
| + Nb | 4E | CE | 79 | F0-79 | 7C | M |
| Enter | E0-1C | E0-9C | E0-5A | E0-F0-5A | 79 | T |
| Esc | 01 | 01 | 76 | F0-76 | 08 | M |
| F1 | 3B | BB | 05 | F0-05 | 07 | M |
| F2 | 3C | BC | 06 | F0-06 | 0F | M |
| F3 | 3D | BD | 04 | F0-04 | 17 | M |
| F4 | 3E | BE | 0C | F0-0C | 1F | M |
| F5 | 3F | BF | 03 | F0-03 | 27 | M |
| F6 | 40 | C0 | 0B | F0-0B | 2F | M |
| F7 | 41 | C1 | 83 | F0-83 | 37 | M |
| F8 | 42 | C2 | 0A | F0-0A | 3F | M |
| F9 | 43 | C3 | 01 | F0-01 | 47 | M |
| F10 | 44 | C4 | 09 | F0-09 | AF | M |
| F11 | 57 | D7 | 78 | F0-78 | 56 | M |
| F12 | 58 | D8 | 07 | F0-07 | 5E | M |
| PrtSc | E0-2A-E0-37 | E0-B7-E0-AA | E0-12-E0-7C | E0-F0-7C-E0-F0-12 | 57 | M |
| Scroll Lock | 46 | C6 | 7E | F0-7E | 5F | M |
| Pause | E1-1D-45-E1-9D-C5 | no break code | E1-12-77-E1-F0-14-F0-77 | no break code | 62 | M |
| Insert | E0-52 | E0-D2 | E0-70 | E0-F0-70 | 67 | M |
| Del | E0-53 | E0-D3 | E0-71 | E0-F0-71 | 64 | T |
| ← | E0-4B | E0-CB | E0-6B | E0-F0-6B | 61 | T |
| Home | E0-47 | E0-C7 | E0-6C | E0-F0-6C | 6E | M |
| End | E0-4F | E0-CF | E0-69 | E0-F0-69 | 65 | M |
| ↑ | E0-48 | E0-C8 | E0-75 | E0-F0-75 | 63 | T |
| ↓ | E0-50 | E0-D0 | E0-72 | E0-F0-72 | 60 | T |
| PgUp | E0-49 | E0-C9 | E0-7D | E0-F0-7D | 6F | M |
| PgDn | E0-51 | E0-D1 | E0-7A | E0-F0-7A | 6D | M |
| → | E0-4D | E0-CD | E0-74 | E0-F0-74 | 6A | T |

Nb = numeric block
M = Make code when key pressed
B = Break code when key released
T = Typematic repeat function with delay & make