Berlin Clock Remake

Using the SPI bus

Prof. Dr. Bernd vom Berg and Dipl.-Ing. Peter Groppe
(Georg Agricola University of Applied Sciences, Germany)

In the world of unusual timepieces one example stands out: the Berlin clock, which has been showing the time since 1975 using an ever-changing pattern of lights. Our version imitates the original using an LED display controlled over an SPI bus.

The original Berlin clock [1] was designed by local engineer and inventor Dieter Binninger as a mathematics teaching aid and was installed in 1975 in an upmarket shopping area in the city. The construction, seven metres high, shows the time using a number of lamps, mostly arranged in groups of four. At the time new mathematics teaching methods were being introduced, giving rise to heated discussions; perhaps not coincidentally the clock rapidly became famous.

By the end of the 1970s a desktop version had become available, using a Texas Instruments TMS1000 4-bit processor (the first mass-produced microcontroller); models are still for sale with almost identical external appearance [2]. A quick trawl of the Internet for 'Berlin clock' turns up not only photographs and information, but also a number of PC-based programs to emulate the clock [3].

We published a variation on the Berlin clock in the Summer Circuits issue of 1998 [4], using a DCF-77 module and a PIC16C54. In a future issue we will also describe another variation: a binary clock.

That example, designed by Professor Beutelsbacher for the 'Mathematikum', an interactive mathematics exhibition centre in Giessen, Germa-

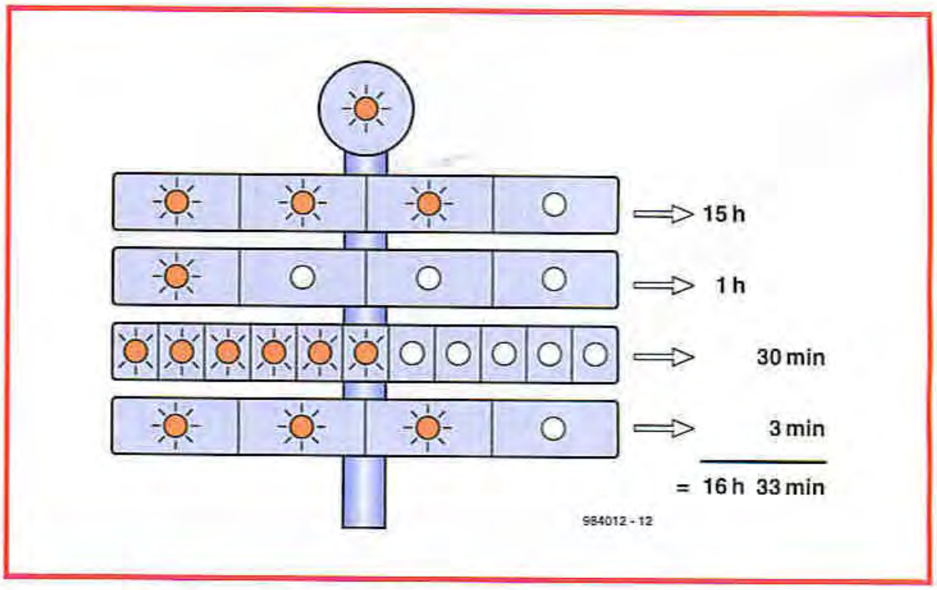

Figure 1. The top lamp of the Berlin clock flashes at 1 Hz, and the remaining lamps show the time.

Figure 2. The display of our version uses LEDs instead of lamps.

ny, was inspired by the Berlin clock. Finding the base-five system employed a little unnatural, however, he opted for a binary display.

Sign of the times

However, it is not so difficult to interpret the display of the Berlin clock, as **Figure 1** shows. The time is shown using a series of rows of lights where each lamp is assigned a 'weight'. From the top row to the bottom row each lamp represents one second, 5 hours, 1 hour, 5 minutes and 1 minute. In the version described in this article we use 24 LEDs rather than lamps, but still arranged according to the original layout (**Figure 2**). At the top centre is a green LED which flashes at 1 Hz. The second row consists of red LEDs and the third row yellow LEDs. So that it is easier to read the quarter hours from the LEDs in the third row, every third one is red rather than yellow.

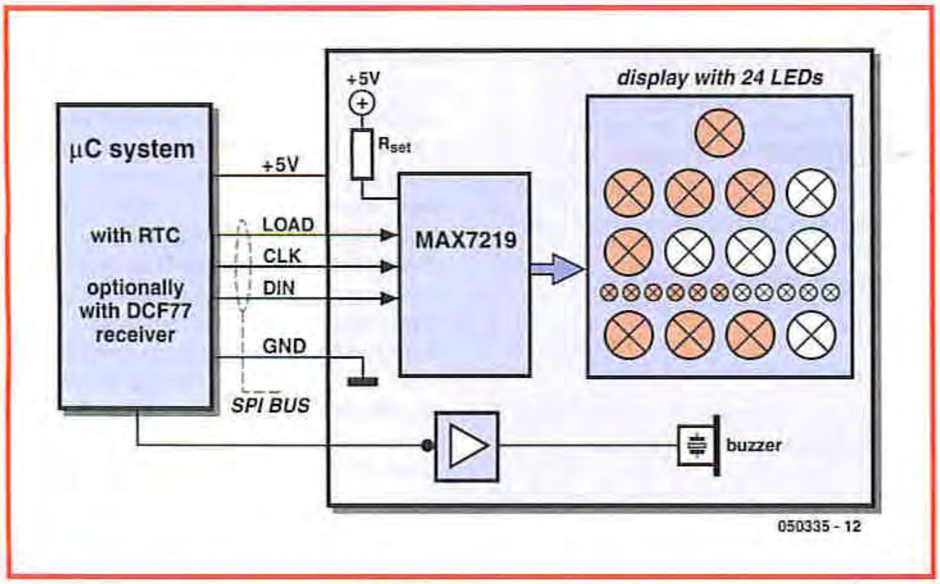

Figure 3. Block diagram of the clock. The microcontroller system drives the display using an SPI bus.

Components

There are two main parts to the design: the example clock software, written in C for an 8051 microcontroller, and the clock display. **Figure 3** shows the block diagram of the system.

Almost any microcontroller system could be used, as long as it has three or four spare digital I/O port pins and either a real-time clock circuit or a radio timecode receiver. Alternatively, a real-time clock could be implemented purely in software.

The authors used a board that they developed themselves, featuring an Atmel AT89C51CC03 microcontroller from the popular 8051 family. The board also sports a 72421 real-time clock IC, although it would also be possible to receive and process DCF-77 signals.

The software reads the time from the real-time clock and generates the necessary control signals on the SPI bus, which connects to the clock display board.

The display board carries an SPI-compatible LED driver IC type MAX7219. Driving this device over a standard SPI bus gives the advantage that almost any microcontroller system can be used as the master, whether based on 8051, R8C, ATMega or PIC. The only point to bear in mind is that it must be possible to create the LOAD, CLK and DIN control signals of the SPI bus using the hardware, or that the selected microcontroller must already provide an on-chip SPI bus interface. In the latter case the communications software required is extremely straightforward.

MAX7219

This device is capable of driving up to 64 individual 20 mA LEDs using a multiplexing scheme. The most important

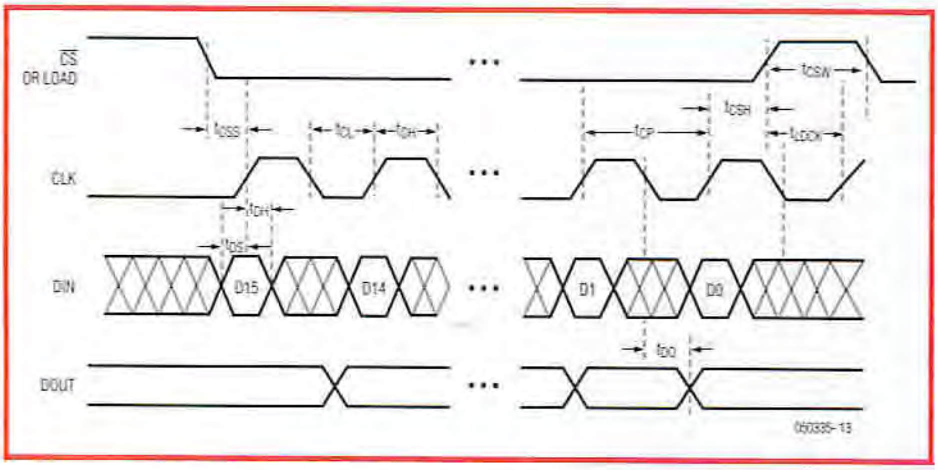

Figure 4. Timing diagram of communications with the MAX7219 over the SPI bus.

features of the MAX7219 are listed in **Table 1**, and a more thorough description of the device can be found in its datasheet [5]. We will give a brief overview here.

The basic functions of the driver IC are controlled using six registers. The LEDs are arranged in eight groups of eight LEDs each; these can either be considered as eight seven-segment displays (plus decimal points) or as 64 individual LEDs to be driven in any desired pattern. This allows for flexibility in the arrangement of the display, and the use of different-sized LEDs (although they must all be 20 mA types).

SPI bus

As we can see in Figure 3 communication between microcontroller and MAX7219 using the SPI bus requires just two power supply connections and three digital I/O port pins, namely:

- LOAD: enable/transfer signal for communicated data;
- DIN: serial data transfer;
- CLK: clock signal for data transfer.

It is very easy to control the MAX7219 from the microcontroller in software using these three signals. The basic SPI signal timing diagram is shown in **Figure 4**. Transferring data to the device proceeds in four phases:

1. Quiescent.
The LOAD and CLK control signals start in their quiescent states, LOAD being high and CLK low.

2. Initiation of data transfer.
Before the data (in the form of two-byte, or 16-bit, packets) are sent to the MAX7219 the LOAD signal is taken low.

3. Data transfer.
The microcontroller presents the first bit on the DIN data signal. The bit is written to the MAX7219 by setting the CLK signal high: on the rising edge of CLK the data bit is clocked into a shift register. The CLK signal is then taken low again. This procedure is repeated a further 15 times so that all the relevant data bits are clocked serially into the shift register. The maximum clock frequency is 10 MHz.

4. Termination of data transfer.
Immediately after the 16 data bits have been transferred the LOAD signal must be set high again. The rising edge of the LOAD signal tells the MAX7219 that the data transfer is complete and that the data bits in the receive shift register can be processed. This last step is very important: if LOAD is not returned high the data bits will remain in the shift register and not be processed.

In our application the SPI bus is driven using three spare ordinary port bits of the microcontroller using the so-called 'bit-banging' technique:

LOAD is connected to port pin P3.5,
DIN to P3.3 and
CLK to P3.4.

These port pins are available on practically every 8051-family device and so the SPI driver software can be used on most of these microcontrollers without modification. If the pins happen to be used for some other function, it is a simple matter to change the code to use other pins.

For correct operation of the MAX7219 each message must consist of two bytes (16 bits) sent consecutively on the SPI bus. The first byte contains the address of an internal control or data register in the device and the second byte contains the data to be written to it.

Full source listings and object code for the C program written for the AT89C51CC03 are available for free download from the *Elektor Electronics* website.

Table 1. Features of the MAX7219

- Drives common-cathode LED displays.
- Individual digits can be changed without the need to rewrite the data for the entire display.
- LEDs individually controlled, or via BCD-to-seven-segment decoder, selectable for each digit position.
- Programmable display size (number of active digits adjustable from one to eight).
- Display test mode, lighting all segments simultaneously.
- Digital and analogue brightness control.
- Serial SPI bus interface (up to 10 MHz).
- Low power shutdown mode (current consumption only 150 μA).
- Suitable for use in applications with:
  - multiple groups of seven- or eight-segment displays using cascade feature;
  - bargraph displays;
  - displays of any shape and size made up of individual LEDs;
  - dot matrix LED displays.

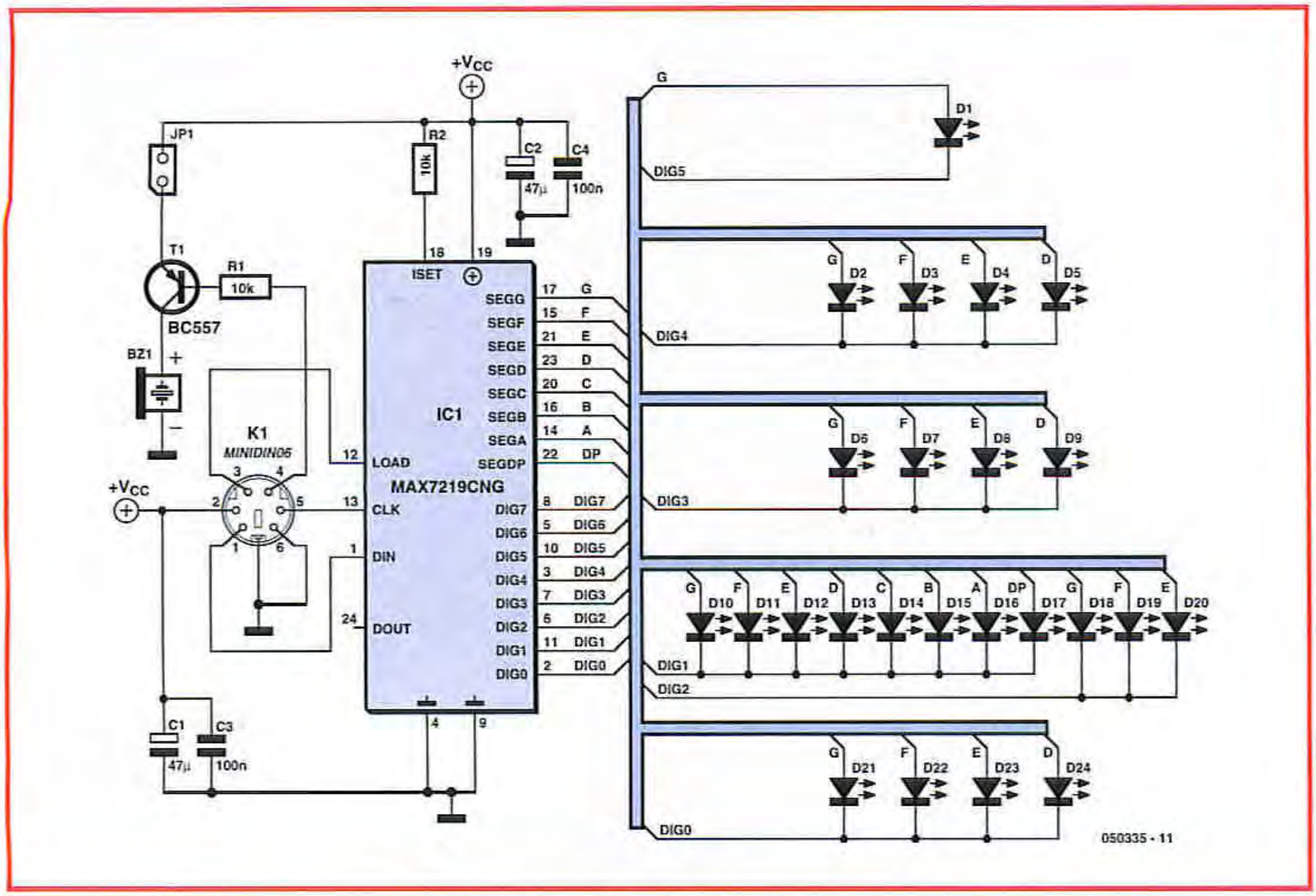

Figure 5. Circuit diagram of the display, using a MAX7219 LED driver and 24 LEDs.

Circuit diagram and printed circuit board

Thanks to the use of the SPI bus and the MAX7219 the circuit diagram (**Figure 5**) and the display printed circuit board could hardly be more simple. Besides the MAX7219 and the 24 LEDs there is also a DC piezo buzzer on the board, which allows an alarm function to be added to the clock.

A simple way to connect the display to the microcontroller master system and power supply is to use a 6-pin mini-DIN connector: this conveniently allows the use of ready-made PS/2 mouse extension cables.

The printed circuit board layouts are too large to reproduce in this article hence are available for **free download** from the *Elektor Electronics* website, www.elektor-electronics.co.uk. Note that the display board is populated on both sides, the LEDs being soldered to one side and the other components to the reverse. This has the advantage that the display can be fitted behind an attractive front panel, making the device an eye-catching and functional addition to any living room, office or desk. All you need to do now is prepare yourself to explain to every visitor how to work out what time it is!

Finally, the author can supply unpopulated circuit boards and aluminium front panels (www.palmtec.de).

(050335)

[1] Information on the original Berlin clock:
http://www.surveyor.in-berlin.de/berlin/uhr/indexe.html
http://www.europa-center-berlin.de/index.php?target=sights&page=7&lang=en

[2] Berlin clocks for sale:
http://www.berlin-uhr.com/ (in German)

[3] A Berlin clock for your PC:
http://www.tucows.com/preview/167639
http://widgets.opera.com/widget/4256
http://www.widgetgallery.com/view.php?widget=35808 (also available for Mac)

[4] Martin Raschke:
Berlin Clock, Elektor Electronics, July/August 1998, p. 87

[5] MAX7219 datasheet:
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1339