Electronic Badge

Michael Gaus & Thijs Beckers

The market for electronic gadgets thrives like never before. Can we still make something nice ourselves? Yes of course! In this month's Modding & Tweaking article we use a mobile phone display to dynamically show images. Mobile, illuminated and very eye-catching as a name badge or case mod.

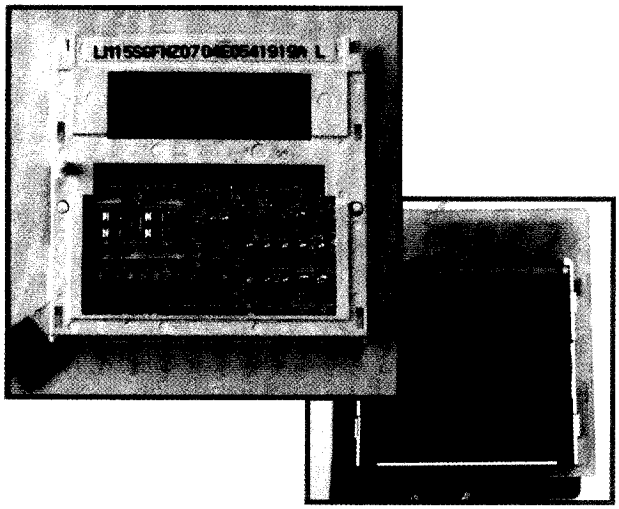


Figure 2. **The C60 display from Siemens. There is no way you could make this yourself for just over a fiver...**

The standard pin or clip-on badges are well-known by now. They are almost a necessary evil when visiting exhibitions and the like. But why not make it into something nice? That is how the following project was born. Using an LCD from an older model mobile phone this is certain to be a success.

MANY COLOURS

The displays in mobile phones are getting better all the time. In the past there was only black and white, these days the colours seem to jump off the screen. We could of course use the latest and greatest display, but that will unfortunately hurt our wallet a lot. A little bit less will work just as well, for example the display from a Siemens C60 we got from E-bay for 7 quid or so (**Figure 2**). That's more like it!
The display with the riveting part number LM15GFNZ07 has a resolution of 101 by 80 pixels and can display 4096 colours. After a little experimenting we discovered that the LCD operates with an SPI bus. Once we knew this, we realised we could use an AVR microcontroller type ATMega8, to drive the display. The only thing that remained to be provided was some sort of medium to store the pictures to be displayed. For this we decided to use an SD memory card (MMC can also be used; it has the same dimensions and pinout). The AVR micro reads the files from the memory card and shows them on the display. Multiple bitmaps are shown one after the other in a slide show. The amount of time that each image is displayed is adjustable.

HEART OF SILICON

In the schematic (**Figure 1**) we see that only eight discrete components are used. The other two components are ICs. The heart of the circuit is the AVR-microcontroller of course, its program memory of 4096 words is for 99.8% full. There are only 14 bytes spare. For a future model we keep the pin-compatible ATMega168 already in mind. This one is via the reset output also easy to debug.
To regulate and condition the power supply voltage for the microcontroller we use a low-drop voltage regulator from Analog Devices, the ADP 3303. Hint: using the samples-program from Analog Devices you can have this IC delivered to your door at no cost. The LEDs for the LCD backlight are connected directly to the 4.5 V input voltage via R1 and R2.

Mobile phone LCD with slide show

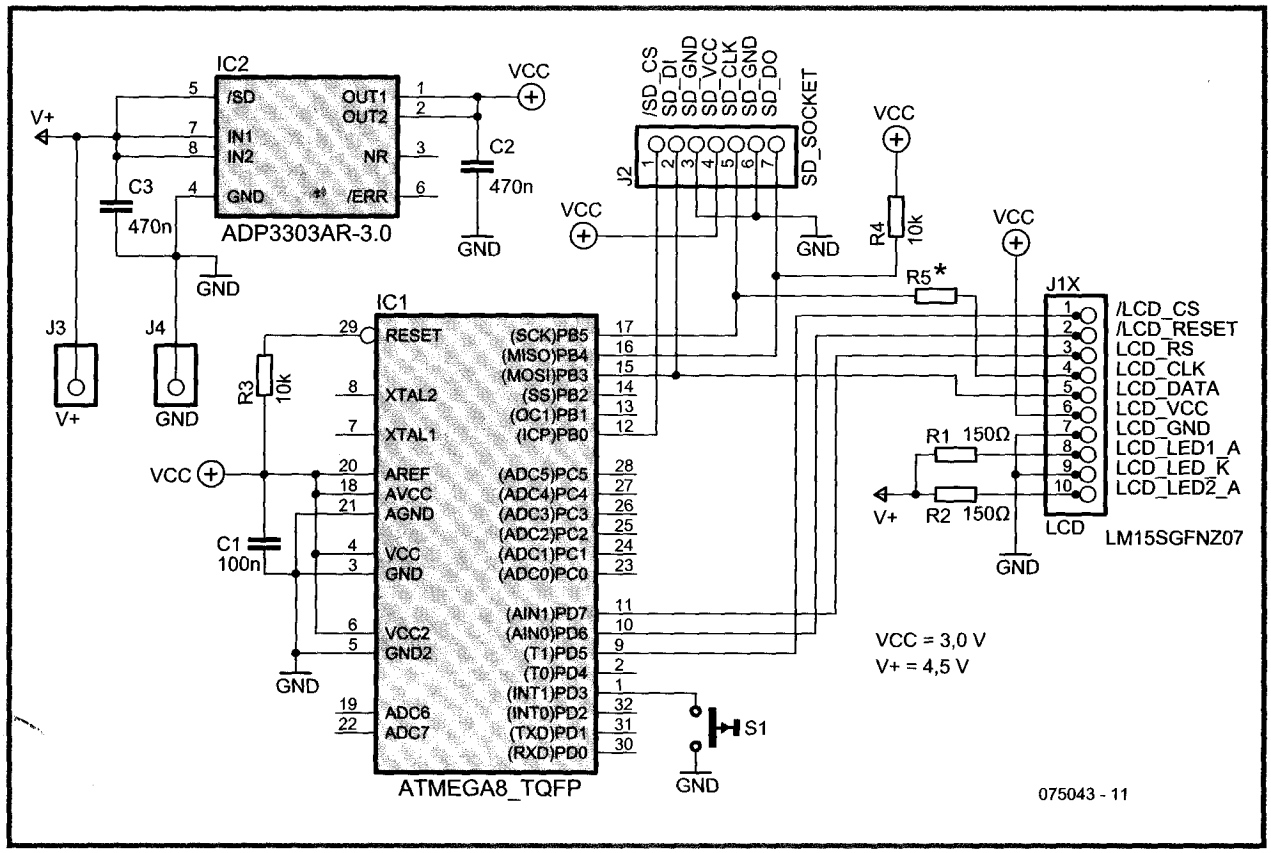


Figure 2. **The C60 display from Siemens. There is no way you could make this yourself for just over a fiver...**

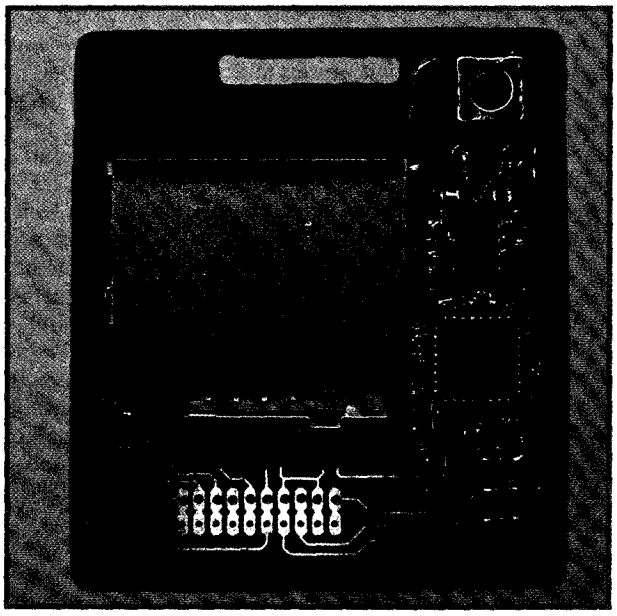


Figure 3. **Our prototype. The display is fitted on the other side.**

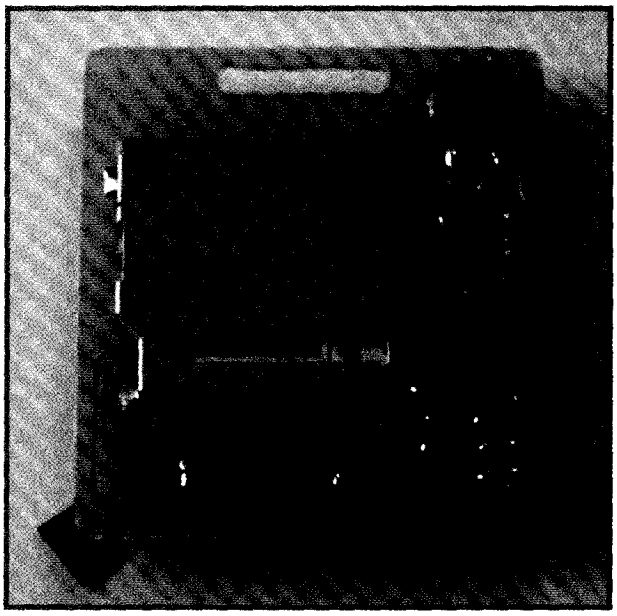
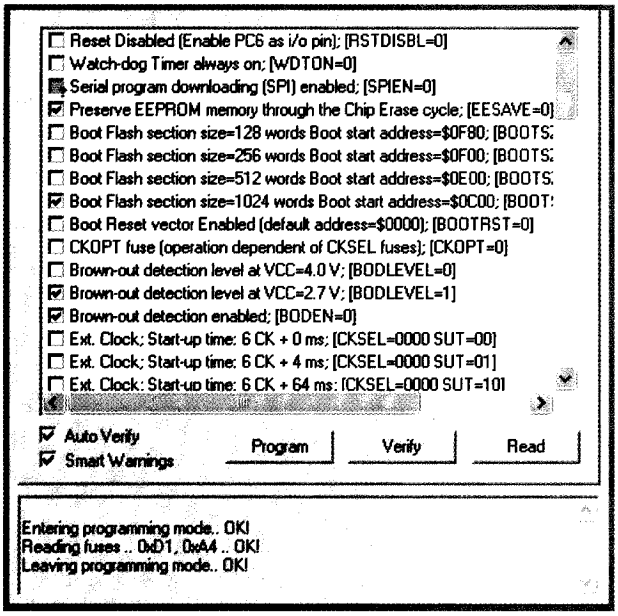


Figure 4. **Wires with reducing lengths make threading easier.**

Figure 5.
The LCD certainly works well.



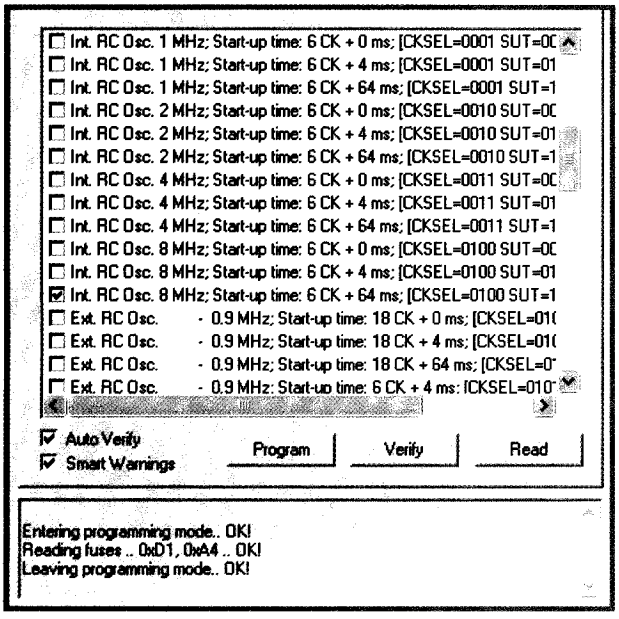


Figure 6.
The fusebits in the AVR micro must be set correctly of course.

To get the circuit to work it is of course necessary to program the AVR. The source code in HEX-format can be downloaded from the Elektor Electronics website (see month of publication). On the SD or MMC card there has to be a configuration file indicating how long each picture is to be displayed. Although the file is just a text file containing only the time in milliseconds, we have zipped it together with the HEX file for convenience.

PRELIMINARY WORK

The BMP files need to conform to a few requirements. Firstly, the dimensions: 101 pixels horizontal and 80 vertical. The colour depth has to be 16.7 million. In this way, every bitmap file, including the header, is exactly 24,374 bytes in size. The microcontroller takes the first four bits of each of the colours (red, green and blue).
The image has to be stored upside down (i.e., mirror it horizontally first). The reason for this is the way a BMP is stored, namely 'from bottom to top'. By storing the picture reversed ('normal') the task of reading the file and displaying it by the AVR has been made much easier.
The memory card has to be formatted in FAT-16 format. This can be done with a standard card reader. This format limits the number of files that can be stored in the root directory to 512. Taking into account the config file and since no directory structure is supported a maximum of 511 pictures can be stored on the memory card. This is sufficient for the time being. The files are also not allowed to be fragmented. By first formatting the card and then copying all the BMP files in one go you can prevent this from happening.
The images are displayed in the same order as they are stored on the SD card. The file name has to be in DOS 8.3 format. Long file names are not supported.

CONSTRUCTION WORK

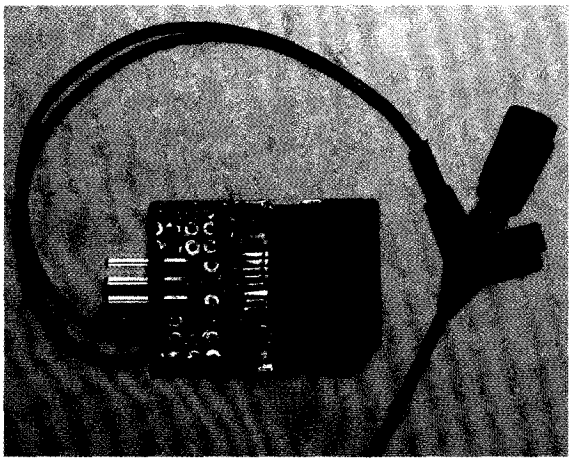
Because the badge was going to be worn by colleagues at the Embedded 2007 exhibition, we quickly designed a small PCB (**Figure 3**). The design of a proper PCB we leave up to you.
When mounting the parts it is best to start with the ATMega8. Followed by the memory card holder after which the other parts can be fitted. Note the wire link which is shown on the schematic as R5.
To attach the display it is easiest to first connect copper wires to it (**see Figure 4**). Cut the wires to different lengths so that they are easier to thread through the holes in the board. A small piece of double-sided tape holds the display in place on the front of the circuit board (**Figure 5**).
Seeing that the circuit is to be worn as a badge, the power supply consists of three batteries, which brings the power supply voltage to 4.5 V. The regulator turns that into 3 V for the processor. Once we have checked this, we can program the AVR micro.
For this we use an ISP (In System Programming). This can be done in one of two ways:

1. solder wires to the programming pins of the ATMega8 (MOSI, MISO, SCK, RESET, VCC and GND) and connect these with the corresponding pins of an AVR-ISP from Atmel;
2. make an SD adapter (see inset).

Before programming, pay careful attention to the fuse bit setting (see **Figure 6**).

SD programming adapter

A Transflash-to-SD adapter can easily be modified into an SD programming adapter. The contact pins are simply connected to the (6-way) plug of the Atmel AVR-ISP In-System Programmer, for example. To do this, carefully cut the adapter open so that the connecting pins for the Transflash card holder are accessible. Then connect the pins to a small PCB into which the ISP header is soldered. This PCB can be glued to the adapter.



The correct connections are:
SD DI → MOSI
SD DO → MISO
SD CLK → SCK
SD VCC → VCC
SD GND → GND
A test clip is connected to the reset pin. This can then be easily connected to the reset pin of the ATMega8 (or R3). You could make a small wire loop at the appropriate side of R3 to make this easier.

The end product is something to behold! It is of course not necessary to use the display just as a name tag. It could also be used as an original case mod. For the really smart guys among you who thought of making a movie with 511 pictures, each displayed for 50 ms, we have bad news. The AVR is unfortunately not fast enough for this.

(075043-I)

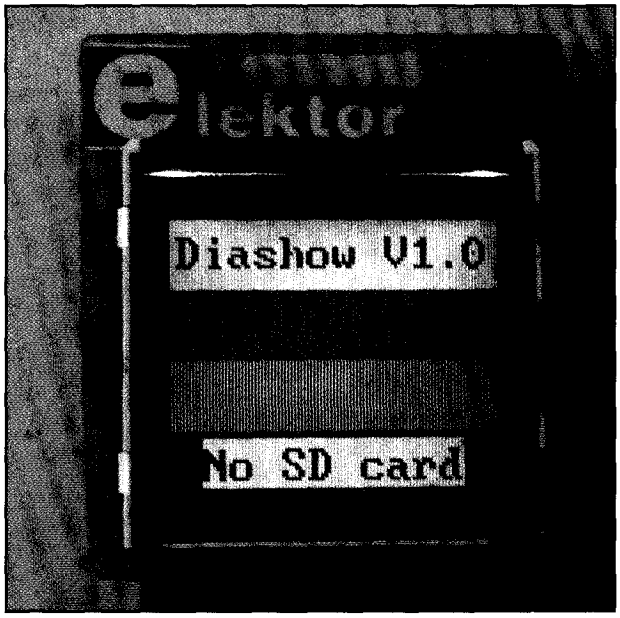


Figure 7.
The AVR micro generates a test image when the circuit is turned on for the first time.

Figure 8.
The AVR micro even controls the contrast adjustment.

PRACTICAL WORK

It's a good idea to test the circuit first without an SD card. By the way, when connecting the battery observe the correct polarity because there is no reverse-polarity protection.
After switching on, the display shows four coloured bars, white, red, green and blue, the firmware version and the text 'No SD card' (**Figure 7**). If this is all working properly then disconnect the batteries and insert the SD card (never insert or remove the SD card when the circuit is powered). Once the battery is reconnected, the four coloured bars appear again for a short time, after which the slide show starts.
To adjust the display contrast, S1 has to be held depressed while the power is turned on. A menu will appear where you can select the contrast adjustment (Figure 8). With a brief push on S1 (< 500 ms) you can scroll through the menu. To select an item S1 needs to be pushed for longer than 500 ms.