

Microcontroller Basics Course FAQs

the most commonly asked questions

By K. Walraven and B. Kainka

We have received many questions from readers regarding our microcontroller course. The most interesting questions are summarised in this list of FAQs (frequently asked questions).

? *Where can I order the printed circuit board for the 89S8252 Flash Board? Can I also order it complete with the components?*

Only the printed circuit board can be ordered from *Elektor Electronics*. All other components must be purchased from regular electronics traders. You can order the circuit board via Internet or by telephoning us at *Elektor Electronics*. Some traders like C-I Electronics (www.dil.nl) can provide a complete construction kit. Elektor also sell a floppy disk (order number **010208-19**) containing all the sample software and necessary utility programs, such as TASM and Microflash. You can also download this software free of charge from our website at www.elektor-electronics.co.uk. Select Free Downloads, then June 2002, then file 010208-19. For some parts of the course, you will also have to download additional programs from the Internet.



? *With some of the course files, I can't find Tasm.exe, and also not the editor.*

The editor is called **TASMedit.exe**, and the exact name of the assembler is **Tasm.exe**. Both programs are stored in Tasm.zip. These programs

must thus be present on the diskette for the course and in the zip file that you can download from the *Elektor Electronics* site. You can see which files are supposed to be on the diskette or in the zip file by looking at the 'context.txt' file.

The assembler manual be separately unpacked using the command 'booz x tasmdoc.zoo'.

? *Where can I find the source code for Microflash, so I can see how this program works?*

All the source code that has been released is on the course diskette. If it is not there, it is not available.

? *There is a RAM IC on the AT89S8252 Flash Board. I wonder if this is really necessary. Can the board also be used without the RAM?*

If the software does not use external RAM, it can be omitted without any problems.

? *What do I have to do to use the microcontroller without the external RAM? If I omit the RAM IC, it looks like the microcontroller is still searching for it (activity on Port 0) and the program hangs. The same program code runs with no problems as soon as the RAM IC is back in its socket.*

Actually, everything should work OK without the external RAM. That is in any case true for the assembler. However, BASIC52 does need external RAM and first checks extensively to see if it is present. Rigel C also does not work without external RAM, but Bascom can easily manage without it.

? *Can I use a MAX202 instead of the MAX232?*

We think so, since it is pin-compatible with the MAX232. However, we have not tried this ourselves.

? *Is the 74HC126 actually necessary?*

Only for programming.

? *I've closely examined the schematic diagram of the Flash Board, but I can't find any indication of what sort of power supply the board needs. It's clear to me that it must be a mains adaptor that can deliver 12 V dc, but how much current must it supply? Maybe that's obvious to an experienced electronics technician, but a hobbyist cannot figure this out so easily using the schematic diagram. I think this actually should be stated in the components list.*

There we have to agree with you. The board needs around 50 mA, and 35 mA during programming. If LEDs or other devices are connected to the board, they will also draw some current. The smallest mains adapters you can buy nowadays deliver around 300 mA, so they will be perfectly satisfactory.

? *How should the board be connected to the computer?*

A female DB9 connector must be fitted to the board. The connecting cable should have a male DB9 connector on one end and a female DB9 connector on the other end. The pins of the two connectors should be wired 1:1, which means pin 1 to pin 1, pin 2 to pin 2 and so on. Pay careful attention to the pin numbers on the connectors, since the male and female connectors have mirror-image arrangements.

For programming you use connector K2, and for serial communications you use connector K1.

? *Despite many attempts, I can't manage to program the board!*

The cause is most likely that the protective Zener diodes on the programming inputs are 1.3-watt types. They have an excessively low Zener voltage at the rather low programming currents that are used, so the digital signals are not recognised.

Replacing the diodes with 0.5-watt types will almost always clear up the problem.

? *I have the following problem with the AT89S8252 board: on Port 1 there is a separate LED connected to each bit line, always between V_{cc} and the port pin. I connected the LEDs directly without any series resistors, but they still give an excellent indication of the output levels.*

? *If I now try to reprogram the AT89S8252, it only responds briefly to 'Break' and then continues to execute the old program. The new program is only sent and subsequently executed if I open the V_{cc} line during programming. I would like to know what's wrong here.*

The answer here is the same as for the previous question. The LEDs prevent the programming signal levels from being greater than 2 V. This level is not high enough to be recognised if an HC type is used for IC2 (but an HCT type will work OK). This means that you shouldn't connect the LEDs without series resistors! For more information, see the question below about connecting LEDs.

? *Whenever I retrieve a program using the TASM assembler or write my own program and then click on the 'TASM' button, the assembler immediately translates the entire program, but this is always the first test program 'flash1.asm'. This happens even if I only change a numeric value (e.g. 0Fh to 10h). If I load flash2.asm and then press 'TASM', the assembler again translates the test program 'flash1.asm'.*

You are not the first person to experience this problem. It is essential to put TASM together with TASMedit and all other associated files in the same folder on the hard disk. If you do this, everything will be OK.

? *I have recently built the Flash Board with the Atmel microcontroller and have already generated a large number of programs in assembler. They all work fine.*

Now I am busy developing C programs for the same Flash Board. However, when I tried to compile the first simple sample C program from the April issue, the Reads51 C compiler reported two linker errors, namely 'external "InitSerialPort0" defined in a01.obj is not exported by any module' and then the same

again for the function `putc` defined in `<Sio51.h>`. What might be the cause of this problem?

These messages mean that the compiler cannot find the include files (such as `sio51h`). If you put copies of the desired files in the directory where the program to be compiled (`a01c`) is located, the compiler will be happy. Most of the include files are located in `reads51/include`.

? *Is there also software available to allow the Flash Board to be programmed under Linux?*

Absolutely. Albert van der Horst of the HCC Forth Users' Group wrote us the following on this subject:

'Here is a Flash program for Linux for the *Elektor Electronics* board in the December 2001 issue. It took a while, since we had to experiment a bit, particularly with resetting the board. It has also become somewhat more luxurious than the original program; among other things it has a test option that helps in tracking down cabling errors.'

'This program is GPL'ed, which means that it can be distributed with no restrictions as long as it is accompanied by the source code.'

You can find more information about this program on the Forth Users' Group website at www.forth.hccnet.nl/.

? *Which connections are suitable for driving something, and how should I do that?*

All of the lines on the board that are not used for something else can readily be used as inputs or outputs (which means P1.0 through P1.4 and P3.2 through P3.7). With the lines that are used for programming (P1.5 through P1.7), you can only connect something that will not disturb the programming process, which in practice means using a series resistance of at least 10 k Ω . If you don't use the serial interface, P3.0 and P3.1 are also free.

Similar considerations apply to using ports P0 and P2. If you do not use the RAM and LCD, you can omit the components and the ports are then freely available; otherwise they cannot be used.

If you use a connection as an input, you can connect anything you wish to it as long as the voltage does not go below 0 V or above 5 V.

If you use a connection as an output, it works as follows: when you output a logical '1', this is in practice equivalent to a resistance of around 50 k Ω to the positive supply

line (100 μ A). If you set the output to '0', it can sink a few mA to ground. To connect an LED, for instance, you don't connect it between the output and ground (since the LED could draw 100 μ A with this arrangement), but instead between the output and +5 V. In order to limit the current to a few mA, a resistance of around 1.5 t it will be fully on at 2 mA.

If you short an output (or all of the outputs) to ground, it's not so bad since only 100 μ A can flow. If you short an output to +5 V, around 5–25 mA will flow. This will not destroy the IC, but you should avoid this situation. If several outputs are shorted to +5 V at the same time, the IC can be destroyed by overheating. You should thus always use series resistors in this situation to limit the current to a maximum of 2 mA per output.

The values given above are rules of thumb. If you measure the actual currents, you will find that they can vary markedly from chip to chip.

? *I have downloaded the Reads51 C compiler, but even though my microcontroller works OK according to the tests the Flash Board cannot be programmed using Reads51, and I have to use microflash.exe instead. Reads51 also has tutorials in which it says that the board should be programmed using a 'MON/RUN' switch, but no such switch is present on our 89S8252 board.*

Am I doing something wrong, or is this simply not possible using Reads51?

Reads51 can only be used for programming with Rigel's own boards. You must in fact always use an indirect route, such as the Microflash program.

? *I am using the demo version of Bascom 51 in combination with the 89S8252 board. In this connection, I would like to know the answers to the following questions:*

– *Which type of programmer must I select for OPTION/COMPILER/MISC/PROGRAMMER to be able to work with the Flash Board?*

– *Is a special configuration necessary to allow the Flash Board to work with Bascom 51?*

Bascom 51 has not (yet) been adapted to our Flash Board. You have to take the output from Bascom 51 (the hex file) and use it as the input to TASM in order to program the Flash Board.

If you make a special connecting cable, it is probably actually possible to send data directly from Bascom to the Flash Board. For this you should use the parallel port of the PC and select the Sample Electronics programmer.

The connecting cable should be wired as follows:

DB25 pin	μ C pin (89S8252)
2 (D0)6	(MOSI) (P1.5)
4 (D2)9	(RESET)
5 (D3)8	(SCK) (P1.7)
11 (BUSY)	7 (MISO) (P1.6)
18–25 (GND)	20 (GROUND)

P1.5–P1.7 can be found on K4 (pins 6–8), GROUND is on K7 and RESET can be best tapped off from push-button S1.

For safety, it is a good idea to insert 220 Ω series resistors in the data lines, but it will probably work fine without these resistors.

? *Why does the LCD not work in combination with Bascom 51?*

Bascomer way uses the data bus, just as with the Flash Board. However, Bascom assumes different connections to the LCD. If you swap lines A1 and A0 to the LCD, it will also work with Bascom.

? *Can I also drive the LCD using Bascom without swapping these lines?*

In order to support the standard LCD screen for the Flash Board without any soldering or swapping lines, the author of Bascom, Mark Alberts, has made a modification. It is included in the new demo version on the Bascom website (www.mcselec.com). The following simple program demonstrates this:

```
$regfile = "89s8252.dat"
Config Lcd = =16 * 2
$lcd = &H8000
$lcdrs = &H8002
```

```
Cls
Lcd "Test"
End
```

? Can I also buy the 89S8252 board fully assembled?

No, as far as we are aware there are only retailers who offer construction kits.

However, there is a German company called AK-Modul-Bus that offers a similar board. It was also developed by the author of the course (B. Kainka) and it can be used very well with the course. The board is called 'ES52-Flash' and costs 99 Euros.

More information can be found on the Modul-Bus website (www.modul-bus.de).

However, there are some differences between this board and our board. The ES52-Flash board has only one serial port that is used for both downloading and communications. The associated download software is called 'Flash.exe' and is available on the Modul-Bus website.

? Where can I find the original version of Intel's BASIC-52?

The original file, 'Basic52.hex', is one of the files present on the diskette for our course (diskette or download number **010208-19**).

? I have the impression that there is a mistake in Listing 3 of Part 2 of the course (February issue, port characteristics and operations). The counter loop is initialised using r1 and then executed using r3.

A very good observation! So why does it still work? The reason is that r3 always leaves the loop with a value of zero and also starts again at zero the next time. As a consequence, the loop is executed 256 times instead of 255.

The correct version is thus:

```
;flash4.asm port outputs
#include 8051.H
.org 0000H

main mov a,#00
next mov P1,a ;1
mov r1,#255 ;1
loop djnz r1,loop ;2 * 255
inc a ;1
sjmp next ;2
.end
```

? I would like to have more information for the course instalments, for instance more programming examples and Basic source code programs.

You can find Basic source code programs, Reads51 programming examples and even more on the author's home page

(home.t-online.de/home/b.kainka/basismi.htm). Most of these files are called mikro1.zip, mikro2.zip and so on.

? My Flash Board sometimes makes strange computational errors in Basic. When computing $\log((1))$ I sometimes obtain a result of 0 and sometimes the board says 'bad argument' or 'divide by zero'. When computing $\exp(1)$ I sometimes obtain the correct value and sometimes a value of -2.7182 . These errors occur when a program is executed in direct mode. I also have an old system with the original 8052AH BASIC and it works just fine! Is there a bug in the Basic version on your website or am I doing something wrong?

This could be a RAM error or a timing error. However, with a RAM error the consequences are usually worse. Besides this, we have to note that a whole lot of minor bugs were present in the original Intel Basic. However, that does not explain why you do obtain the proper results with your old system. Most of the errors will probably disappear if you use the upgraded version (BASIC5 2 V1.3).

? The Microflash.exe programming software freezes whenever I try to reprogram the board while the flash2 program is running (the program that loads 0Fh and F0h alternately into P1). The 153-kHz signal on P1.6 can also be found on the CTS line of the PC, and evidently there is a timing problem that prevents the DTR line from putting the board into the Reset/Programming mode. As soon as the HEX or BIN button is clicked in Microflash, the program hangs. Everything works normally for loading the flash2 program. Could this have something to do with my PC? I am using Windows XP Professional. I can work around this problem using a trick, which is to briefly disconnect R1 and D1 and forcing a reset at that point using a pushbutton switch connected to V_{cc} , after which I select the program to be programmed in Microflash. I was able to load the very first test program using this technique.

After that I restored everything to its original state. Reprogramming flash1 went well, and reprogramming with flash2 also worked, but as soon as that happened I was again faced with the same problem.

It looks like something is going wrong with the download, which leads to your timing problems. Try using the ATMElisp program, which will probably work better under Windows XP. Besides this, you should have a good look at the Zener diodes used on the circuit board (see one of the earlier questions).

? I have two questions regarding the Flash Board. I am currently programming an IDE tool for the Flash Board and I have run into a problem with the serial connection.

Although I can separately configure the DTR and RTS lines using

```
SetCommState(hport, SETDTR),
```

I do not know how I should do this with the TxD (MOSI) and CTS (MISO) lines. Do you have a tip for this?

Also, I would like to know whether I can have the microcontroller execute only one instruction and then stop, for debugging purposes. It is possible to set breakpoints?

If you use PORT.DLL, which has been used in a variety of other projects (and can be downloaded from the September 2000 page of the *Elektor Electronics* website, number **000074-11**), it is very easy to set the lines to the desired states.

For debugging you would have to use a monitor program, but we do not have any experience with this.