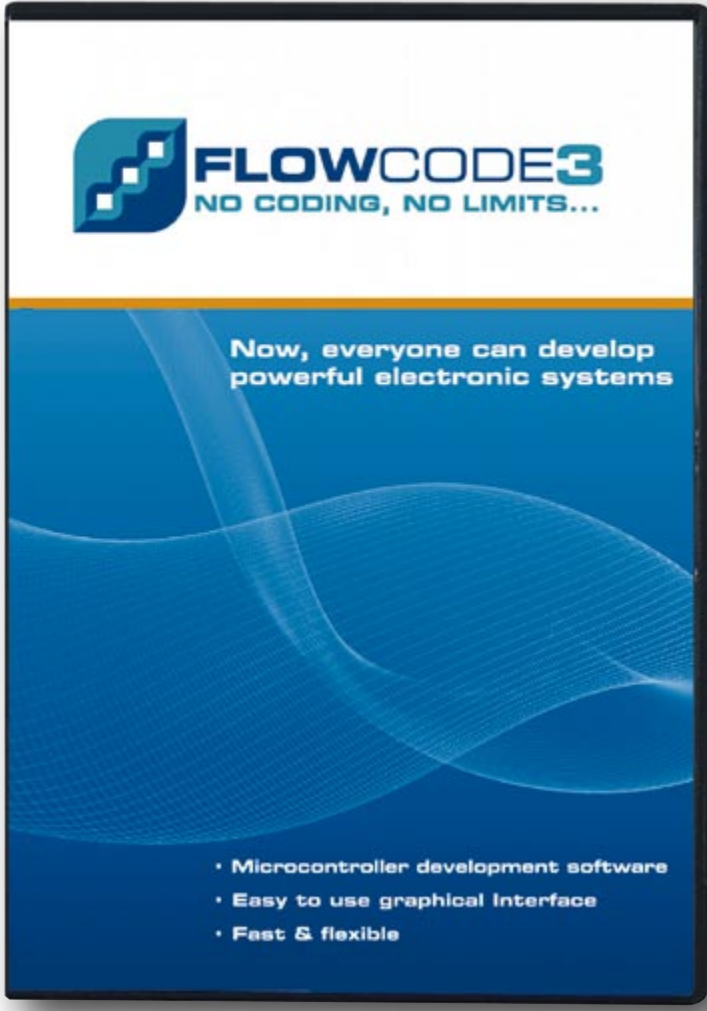


# A New Flowcode

## Putting Flowcode 3 through its paces



Jan Buiting

**A new version of Flowcode for E-blocks has just been released — version 3. This is more than a simple upgrade: Flowcode has matured into a nice if not impressive development tool.**

Many of our readers are now quite familiar with Flowcode, the software brains behind all E-blocks projects, with some having actually bought the product. A number of E-blocks tutorials, reports and application examples were published earlier this year.

Originally, Flowcode was designed to help College students develop electronic projects based on the popular PIC microcontroller. As time has passed, more and more features have been added to Flowcode, and the latest version 3 has actually turned out to be quite a nice development tool not just for the programmers starting out but also for professional engineers.

In this respect, my own position is that I have been an electronics enthusiast for 30-odd years and have occasionally worked with microcontrollers, specifically the CDP1802 (see *Retronics* in the October 2006 issue). Fortunately, thanks to contributing authors and colleagues in the Elektor lab I have never been out of touch with microcontrollers and related stuff like (E)PROMs and Flash devices. Although I can read PIC, 8051 and AVR assembly code and the odd piece of C and Visual Basic, I must admit that I am not *au fait* with the latest in microcontroller land when it comes to programming. So, when the new

Flowcode landed on my desk I thought I would see for myself whether it lived up to my expectation: microcontroller programming should be *easier* than 20 years ago.

### On the desk and on the PC

To undertake the review I got hold of the 30-day demo copy of Flowcode 3, which is available as a free download from our website. At the time of writing I got Beta 5 — the final version will be on our website. I also got out a set of E-blocks modules comprising a PICmicro Multiprogrammer board, an LED board, a Switch board and an LCD board. The lot was connected up as shown in

### Figure 1.

My colleagues in the lab having estimated my proficiency (“zero”) and general chances of success (“we’ll see”), gave me an 18-pin PIC16F88 device (worth £2.75) to insert in the Multiprogrammer. Somehow I surmised that this must be a low-end device but having looked up the datasheet on the Microchip website I discovered the device has 7 kB ROM, 368 bytes of RAM, seven 10-bit A to D channels, timers, a USART communication port and a host of other things I had never heard of (BOR? LVD?).

# for E-blocks

Looks like things have moved on a little since I last used a microprocessor! I was happy with my 'low-end' 'F88 chip because its simple architecture would prevent me from attempting the impossible (yet).

Having installed Flowcode 3 and the associated PPP driver (which was a breeze) I thought I would see if I was in control of the hardware and software setup by getting a single LED to flash. I know the example is corny, but you have to walk before you run — writing that I<sup>2</sup>C bus driver or 8-level DMA demultiplexer will come later!

## The sequence

Right. In Flowcode I started a new flowchart, dragged an Output icon onto the chart and clicked on it to get the properties up. I set bit 1 of PIC port A to go On and pressed the 'PLAY' button. The software simulated the 'program' and pin A0 on the PIC graphic faithfully went red to indicate a logic 1 had been placed on port A bit 0. After a little trial and error I very quickly had a program that flashed A0 on and off on the screen. No time wasted on editing assembly code, re-assembling the code, burning a new EPROM, stirring the coffee and finally relaunching the application, fingers crossed and praying for success. That was 20 years ago.

I learned a few things here: you need to slow Flowcode's simulator down so you can see what is going on in your program. Correspondingly I deduced you need to put delays in the program otherwise what I had hoped to be a slowly flashing light will just look like a slightly dim LED. You can see my first program in **Figure 2**. Not bad for 10 minutes work.

Flushed with success I then decided it was time to see this in action on a real PICmicro device. Simulation is great but I am only convinced by real-life electronics. To download a program to a microcontroller, in my case the lowly 16F88, you simply click on the small chip icon on the menu and off it goes.

Off it went — but nothing happened on my hardware. Time to read the Help file and the Getting Started guide, both of which can only be described as crisply produced, easy-going yet comprehensive documents with great educational value. Having gone through this I concluded that I needed to load the CHIP...CONFIGURE screen to set up the parameters for the real-life PIC: set the clock oscillator for a Crystal, turn off the watchdog timer, Brown Out Detection and Disable Low Voltage Programming. These settings, by the way, are now the number-1 problem experienced by readers having built an Elektor project based on a PIC, having burned their own PIC device and finding that it doesn't work. Check those PIC config words!

Then I pressed the Download to Chip icon again and I had my first program running on the real-life PIC. I looked at my watch — I started 30 minutes ago. For a relative newcomer who occasionally still gets nightmares in assembly code I did not think that was too bad.

## Learning curve

Having read the Help file I discovered that Flowcode 3 is shipped with around 30 example files that demonstrate

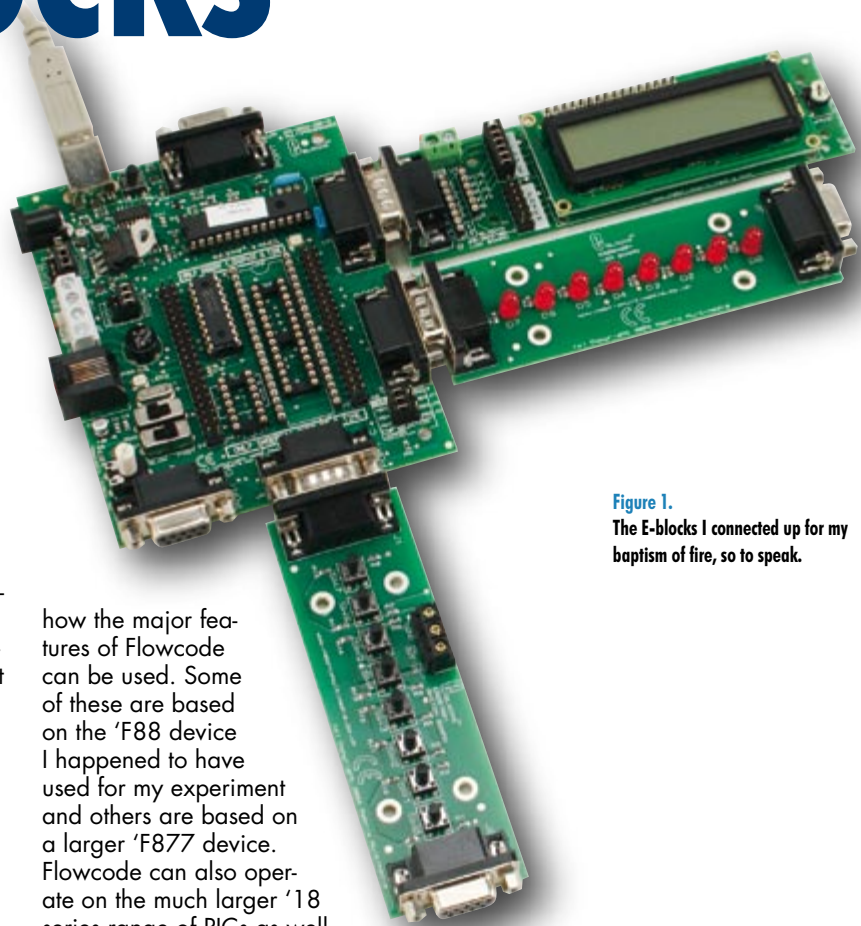


Figure 1.  
The E-blocks I connected up for my baptism of fire, so to speak.

how the major features of Flowcode can be used. Some of these are based on the 'F88 device I happened to have used for my experiment and others are based on a larger 'F877 device. Flowcode can also operate on the much larger '18 series range of PICs as well as the smaller '12 series. I then spent the next hour or so looking through the example files for the 'F88, downloading a number of them to my hardware and launching the applications.

Many only require the use of simple LEDs, switches and LEDs, but some call for more advanced components

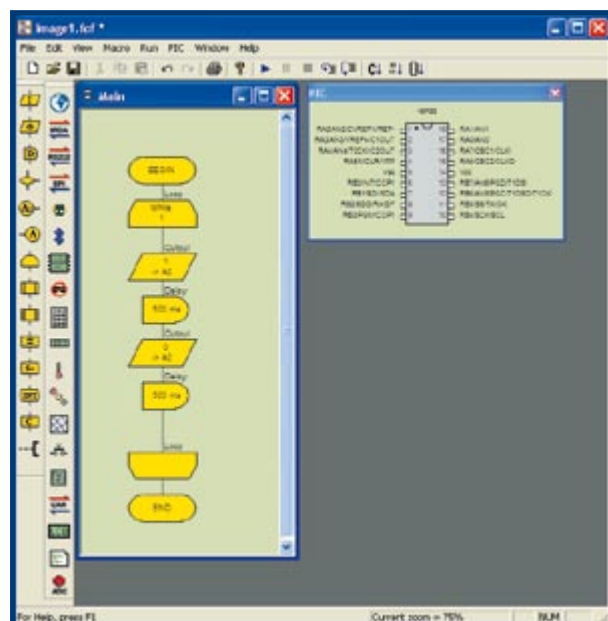


Figure 2.  
My first program.

# What's new & better in Flowcode Version 3

## Graphical User Interface improvements

- Zoom: Multiple zoom levels, zoom to fit
- Tiling: horizontal tile, vertical tile
- Smaller PICmicro MCU on-screen device
- Screen icons: new graphics, description now inside icons, better comments, more icons per screen viewable
- Screen appearance: user selected icon shading, and background colour
- Hardware and software macros now have separate icon graphics now known as: 'macro' and 'hardware macro'
- New: Print Preview and print to screen zoom setting
- Flowcharts can be exported to JPEG or BMP for incorporation into documents
- Tile horizontal and vertical and auto-arrange for multiple macro viewing

## Improvements for migration to C

- All icons have bubble help to display icon function
- Icons can also produce equivalent C code of each icon as bubbles
- Students can view the C code equivalent of the whole program
- Students can view the Assembly code equivalent of the whole program
- Screen layout is preserved on save to allow educators to build more relevant examples
- Tutorial file descriptions now included
- Tutorial files exploit features such as labelled components to add context

## Multilingual support

- Main program and Help file: English, French, Dutch, Finnish, German, Spanish
- Main program but not Help file: Chinese, Italian, Greek

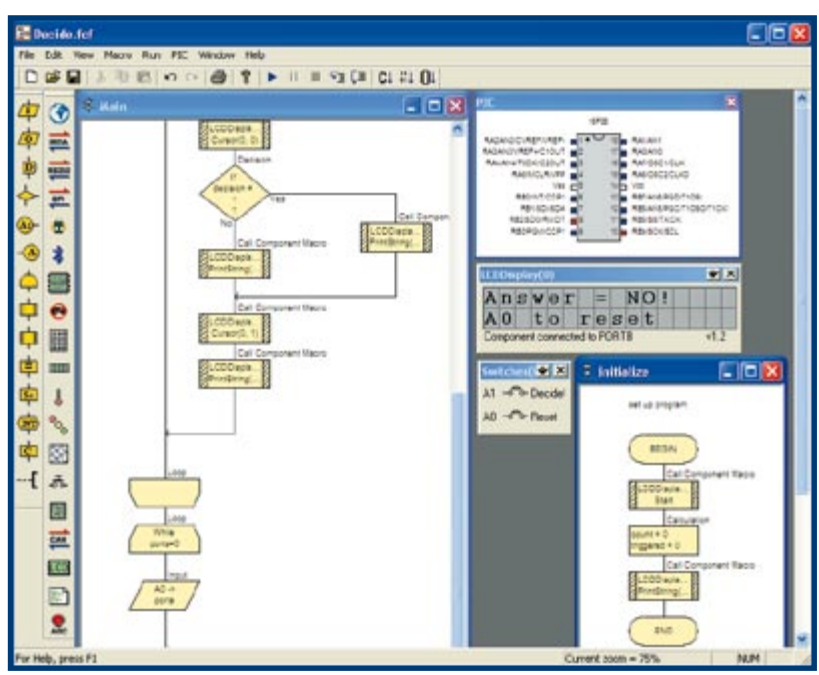


Figure 3. My final Decision Maker program.

(i.e. E-blocks) like keypads and sensors. When you look through these examples, Flowcode's two main strengths become clear. Firstly, it is a very good way for those that are not front line programmers, like myself, to understand how programs work and can be created in a short time. This in turn means that the learning curve for Flowcode itself is actually very steep. Secondly, it is not a toy. Whilst it is great for those starting out to program, it is also quite powerful, with fully supported interrupts and a range of communication protocols like SPI, RS232, CAN, Bluetooth and even TCP/IP supported — all of which I will grind my teeth on — in due course!

# Earlier in this series

- *Electronic Building Blocks*, November 2005.
- *E-blocks and Flowcode*, December 2005.
- *E-blocks in Cyberspace*, January 2006.
- *E-blocks – now you CAN*, February 2006.
- *E-blocks Making Waves*, March 2006.
- *E-blocks Making Waves at C*, April 2006.

## From 2 to 3

Those of you who are currently using Flowcode version 2 will want to know what improvements have been made to version 3. I have no experience of version 2 but looking in the Help file it states that major improvements have been made to the graphical user interface, there are new features to help learners, lots of functionality improvements such as 16-bit numbers, support for strings, better interrupt handling, improved macros, more components, and so on. A full list is given in the **inset**.

## Flowcode flips the coin

So, having trained myself up it was time to get down to developing an application that, although serious in intention, is the tongue-in-cheek equivalent of flipping a coin: a PIC-assisted Yes/No decision maker. Some people, mostly managers, attribute great authority and powers of decision to PCs and microcontrollers, arguing that they are 100% digital hence have no 'grey areas' or room for 'discussion'. What I need to confirm these people in their beliefs (as well as making them forget the simple coin in their wallet) is a program where they could press a button and get a 'YES' or 'NO' decision to display on the LCD. So I set about making it — as an exercise, of course. You can see the final program in **Figure 3**. It took me about an hour in the end. I have put up the file 'Decide.

### Software functionality improvements

- Better range of simulation speeds to check working program before downloading it to the PICmicro microcontroller
- Alter variables whilst simulation is paused
- Support for 16-bit numbers and arithmetic, choice of types includes CHAR, INT and STRING
- Support for hexadecimal and binary numbers in all dialogue boxes
- Full support for strings including string manipulation commands like ADD, LEFT, RIGHT
- Variables are now case sensitive
- New string process icon supports string manipulation
- Interrupt icon supports a larger range of interrupts as well as custom interrupt definition. Each interrupts run a macro of your choice.
- Improved Delay icon with a much greater range of delays
- While icon can operate for a defined number of times
- Subroutines can now have parameters passed to them, and returned
- Larger range of supported devices now includes 18 series PICmicro microcontrollers (technical specification for full list).

- Undo and Redo commands
- Improved C compiler

### Component improvements

- LCD: range supported now includes 40-character 2-line; 20-character 4-line etc.
- Full LCD functions now supported with scroll and other features.
- Switches can now be labelled, options for display vertical and horizontal, left to right or right to left.
- LEDs can now be labelled, options for display vertical and horizontal, left to right or right to left.
- New PWM (Pulse Width modulation) component for motor control.
- Analogue components available now include thermometer, dial, or slider
- Target communications components now include RS232, I2C, Internet web server, Internet TCP/IP, Bluetooth, CAN bus and LIN bus

E-blocks for Prototyping, May 2006.

E-blocks and X-10, June 2006.

E-blocks Easy ARM Pack, September 2006.

Articles may be downloaded individually from [www.elektor.com](http://www.elektor.com).

An overview of available E-blocks and software is available on the [SHOP](http://www.elektor.com) pages at [www.elektor.com](http://www.elektor.com)

fcf' for free downloading with this article — the file number is **065096-11.zip**.

Unfortunately Flowcode does not have a random number generator so I created a simple counter and then used the MOD feature in a calculation icon to detect whether the count was odd or even. This outcome was used to answer Yes or No to whatever weighty decision the user had in mind.

What impressed me is just how easy this was. The LCD was very easy to incorporate in the program and it really adds a lot of functionality to an electronic system.

The hardest part of the job was deciding how I was going to structure the program itself in terms of the logical flow and the variables I needed to track the status of the program. Once I had done this on paper, transferring it to Flowcode was quite painless. The thing that most impressed me was that once I had got the program working using Flowcode's simulation mode, when I transferred it to the hardware (this time having loaded the PIC config bits!) it worked just like the simulation did.

### Conclusion

To a newcomer, Flowcode 3 'does what it says on the tin'. It was easy to get started and make a program, was intuitive to use, and produced code that worked. There are a few minor disappointments such as the lack of a random number generator, which I happened to stum-

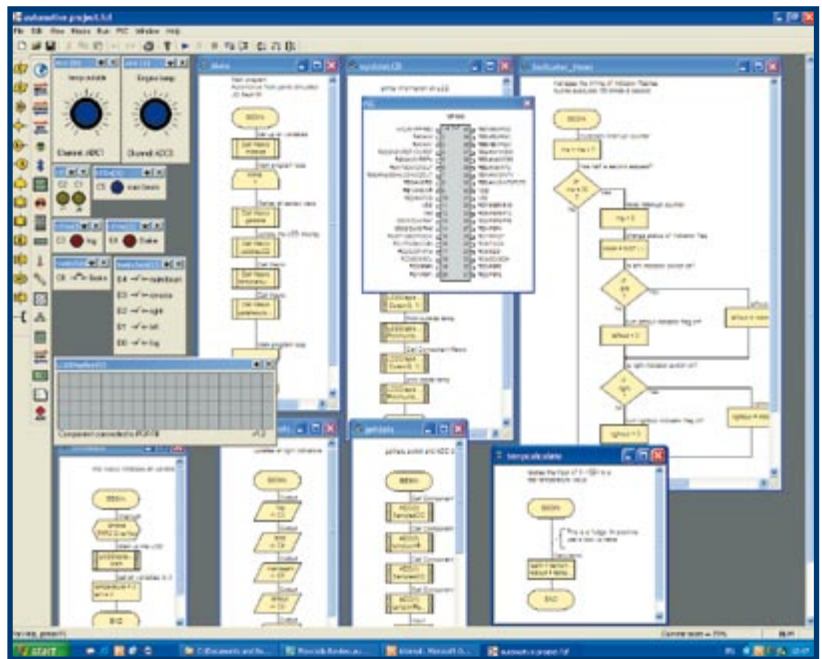


Figure 4. An example of a more advanced Flowcode program.

ble on for the *Decision Maker* program, and the fact that floating-point numbers are not supported as variables. It does however seem to have more features and capabilities than you could shake a stick at. I was also struck by the thought that it was more fun than the day job. Then I realized that this was my day job!

(065096-1)

A full featured but 30-day limited demo version of Flowcode 3 can be downloaded free of charge from [www.elektor.com](http://www.elektor.com).

Follow: magazine → volumes → 2006 → december → A New Flowcode for E-blocks