Flowcode 7

Displays in Flowcode

FLOWCODE7
Smarter programming

Working with displays

www.flowcode.co.uk/circuitcellar

By Ben Rowland,
Embedded Engineer

Last month, we introduced Flowcode 7, a flow-chart driven, electronic integrated development environment where users can produce hex code for more than 1300 different microcontrollers including PIC8, PIC16, PIC32, AVR, Arduino and ARM.

This month, we are going to look in more detail at one of the great features of Flowcode over a pure coding environment such as Arduino; the ability to simulate the various component libraries within the IDE. Whilst simulation may be seen as an "glory" feature to many, what must not be underestimated is the ability that simulation gives users to confirm code is working as you intend before compiling the code and programming it to the black box that is a microcontroller (**Figure 1**).

DISPLAYS IN FLOWCODE

Flowcode includes hundreds of individual components, from simple input/output components to complex communications such as Bluetooth, GPS,TCPIP, mechatronic components and more. One set of components that has been expanded for Flowcode 7 are graphical LCDs, both monochrome and colour. The electronics market is full of an overwhelming selection of graphical displays all of which may have their own unique driver IC and additional
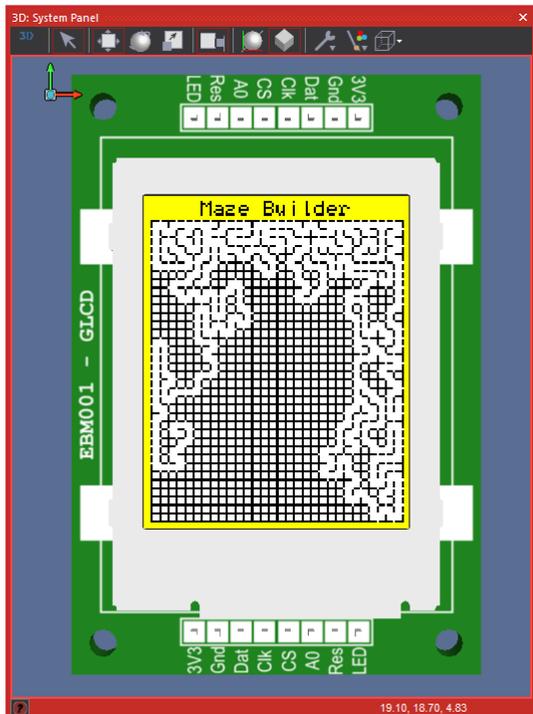
Figure 1 - *A maze generation algorithm being tested using a graphical LCD and the Flowcode simulation.*

configuration settings. The Flowcode environment has been steadily building up the list of graphical LCDs that the IDE will work with to ensure that driving a display is as easy as possible.

Flowcode 7 out of the box supports the following display driver IC's and many more: ILI9341, KS0108, ST7036, ST7565R, ST7567, SSD1305, SSD1306, SSD1289, SSD1322, SSD1351, SSD1963, 4D Systems Displays.

Displays feature in just about all electronic engineering developments and are the perfect output method to display feedback from your system. With this in mind, Flowcode is one of the easiest ways, users can interface with a whole host of different display modules, without the need to worry about developing complex or time-consuming code, which simply allows for the initial communication to be established.

Flowcode works by wrapping components (including the displays listed in **Figure 2**) into component macros. These component macros include all of the functions that can be called upon by users: PrintNumber, PrintString, etc. meaning the ability to interface to the display is quicker than ever before.

COMMUNICATING WITH OUR OWN DISPLAYS

The list in Figure 2, is far from complete and so there are two other routes by which users can get a new display type into the Flowcode environment should your chosen display not already feature.

1 – Using Flowcode's In-Built Component Creator

All of the components available in Flowcode are Flowcode projects that have been exported as a component. Flowcode provides the source for a few different graphical LCD components and by following a few of the component creation tutorials available through the Flowcode support resources, it is possible to tailor the examples to drive a new kind of display. The main thing that is likely to change are the pixel resolutions, pin connections, initialisation sequence and
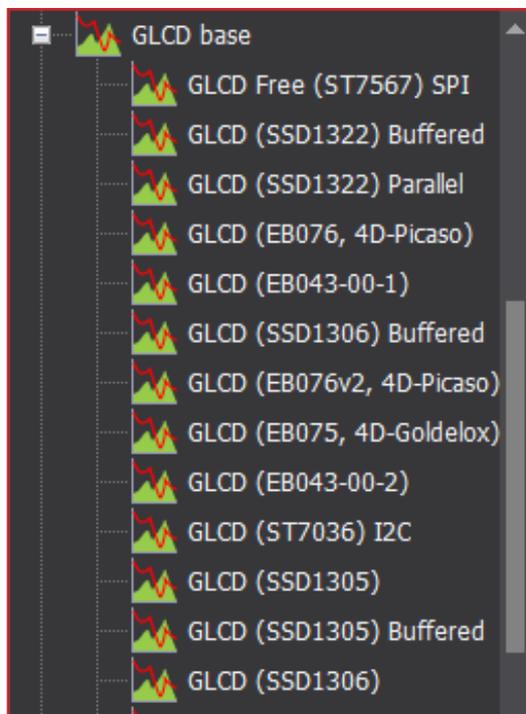

Figure 2 - *A partial list of the graphical display components available in Flowcode*
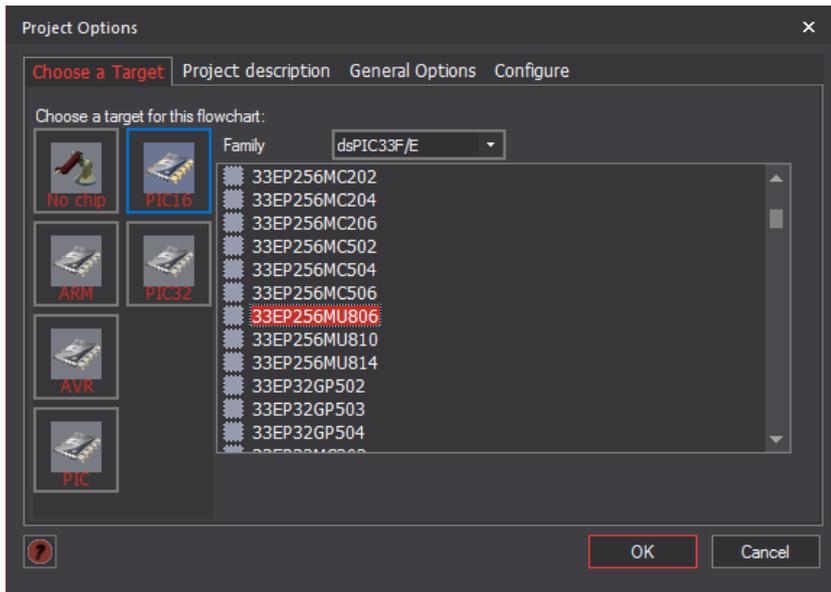
Figure 3 - Changing the target microcontroller is easy, even half way through your project
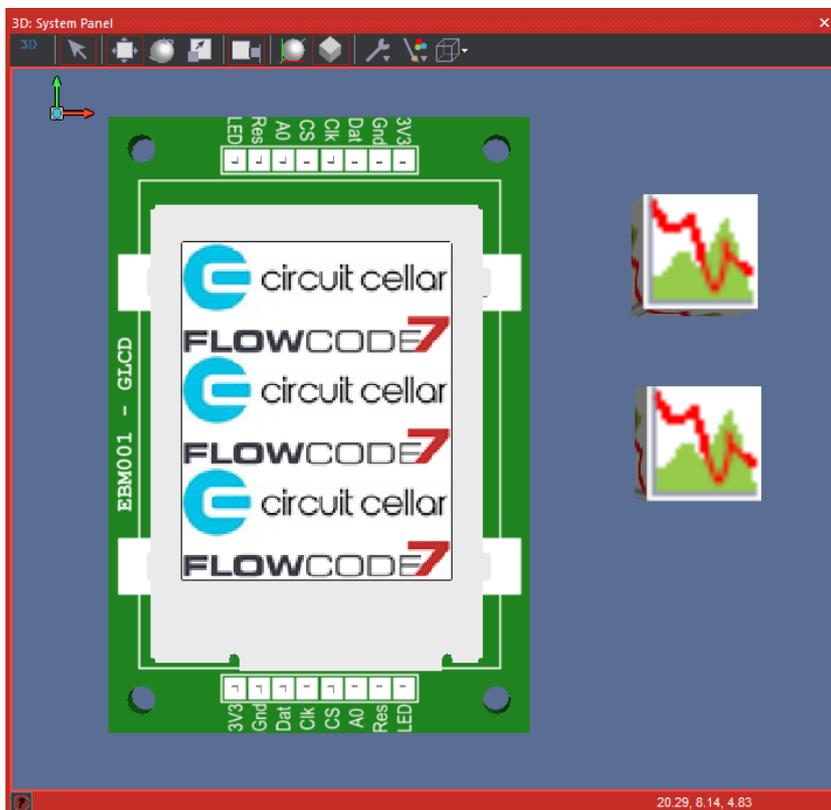

Figure 4 - Bitmap drawer component in action allowing for easy graphics and sprites

the commands to modify a pixel. Once you can control a pixel all the other functions should automatically fall into line.

2 – Sending a Sample Display to Matrix TSL

If you have a display you would like to see in Flowcode but don't have the time or resources to generate the component yourself then there might be something the development team at Matrix TSL can do to help. Send your ideas via the Flowcode user forums for consideration from the Flowcode development team on their ability to create the component for you. If this goes ahead, the component will then be included automatically in future releases of the Flowcode software.

CODE AND DISPLAY PORTING

Another great aspect of Flowcode is the ability to change microcontroller half way through a project. This is known commonly as code-porting. Changing the microcontroller is as easy as choosing a new micro from the project options at the start of the whole process, and users are able to switch easily from a 16 bit to a 32 bit PIC device, or from a PIC to an Arduino, without any need to spend hours re-writing major parts of code from your project.

This method of programming gives you freedom to make fairly significant changes to your designs, half way through development, including increasing memory on your design, without the worry you experience using textual based languages such as C (**Figure 3**).

The same thing also applies to the graphical LCD components. For example, if the user finds a significantly cheaper display half way through the product design cycle, rather than having to go through code line by line (or icon by icon as the case may be) it is simply a case of using the "Selection > Replace" tool available by right clicking the component on the panel to instantly switch between display types. This level of flexibility is what gives Flowcode its appeal, not only to the Maker community but also

FLOWCODE7
Smarter programming

to product development engineers at some of the World's most advanced blue-chip companies.

BITMAP DRAWER COMPONENT

As there is already an impressive library of graphical LCDs, the Flowcode development team have also started to create new components that will leverage the available displays. One of these components is the Bitmap Drawer component which allows the user to load a bitmap image directly from your PC to the ROM of the microcontroller. The component also features a macro to automatically draw the image to the display of your choosing. You can see an example of this in practice in **Figure 4**, using the Flowcode and Circuit Cellar logos.

LED MESSAGE BOARD

As we're on the subject of graphical displays another recent addition to Flowcode is the ability to create 1D, 2D and 3D displays and light effects using the WS2812B intelligent LEDs. These LEDs have an inbuilt controller IC which allows them to be driven via a single serial signal. What's even more impressive is that the serial signal can be chained together from LED to LED to control a

Figure 5 - Large LED message board displaying ROM bitmap image

large number of LEDs. Again the Flowcode component offers full simulation and also compatibility with the Bitmap Drawer component allowing us to easily display and animate the Matrix logo as seen in **Figure 5**.

It is worth noting for those interested in developing a project like this for yourself, that you need a relatively fast microcontroller to output the high speed timings to drive the WS2812B LEDs. For example, an Arduino would do for this but instead we used the dsPIC ECIO40P16 which has roughly 9 times more processing power than an AVR based Arduino and is available from www.matrixtsl.com.

Ben Rowland is a graduate of the University of Huddersfield in the UK, where he earned a Master's degree in Embedded Systems & Cybernetics before going on to work in the Flowcode development team at Matrix TSL where he has worked for more than 10 years.

SPONSORED CONTENT