

MATRIX

EBLOCKS[®]

Beginners guide to Flowcode
and Arduino

www.matrixsl.com

Contents

About this document	3
Prerequisites	3
The Hardware	4
Connecting your Hardware	5
Creating your First Program	6
Example 1: LED Flash	9
Example 2: LCD Screen & Counter	12

About this document

This document will provide an overview for using the new EB093 Arduino Compatible E-block Shield and EB083 Development Board with your Arduino Uno Microcontroller and Flowcode 6.

1. Trademarks and copyright

E-blocks is a trademark of Matrix Technology Solutions Limited.

2. Disclaimer

The information provided within this document is correct at the time of going to press. Matrix Technology Solutions Limited reserves the right to change specifications from time to time.

3. Testing this product

It is advisable to test the product upon receiving it to ensure it works correctly. Matrix provides test procedures for all E-blocks, which can be found in the Support section of the website.

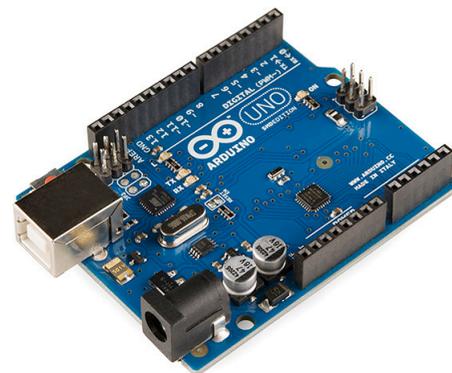
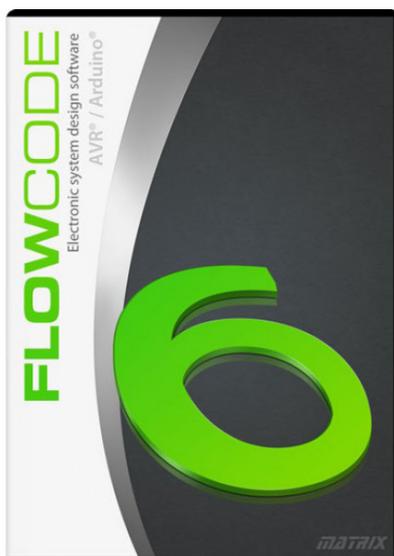
4. Product support

If you require support for this product then please visit the Matrix website, which contains many learning resources for the E-blocks series. On our website you will find:

- How to get started with E-blocks - if you are new to E-blocks and wish to learn how to use them from the beginning there are resources available to help.
- Relevant software and hardware that allow you to use your E-blocks product better.
- Example files and programs.
- Ways to get technical support for your product, either via the forums or by contacting us directly.

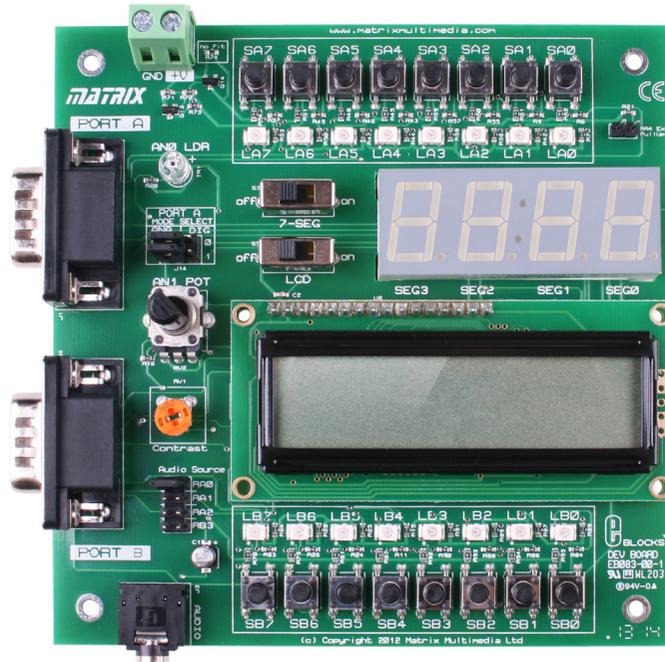
Prerequisites

1. Working installation of Flowcode 6 with a valid AVR license
2. EB083 Development Board
3. EB093 Arduino Compatible E-block Shield
4. USB Cable to program/ power the Arduino Uno
5. Arduino Uno R3 hardware (other Arduino products may be compatible, but this guide is designed for the genuine Arduino Uno R3)

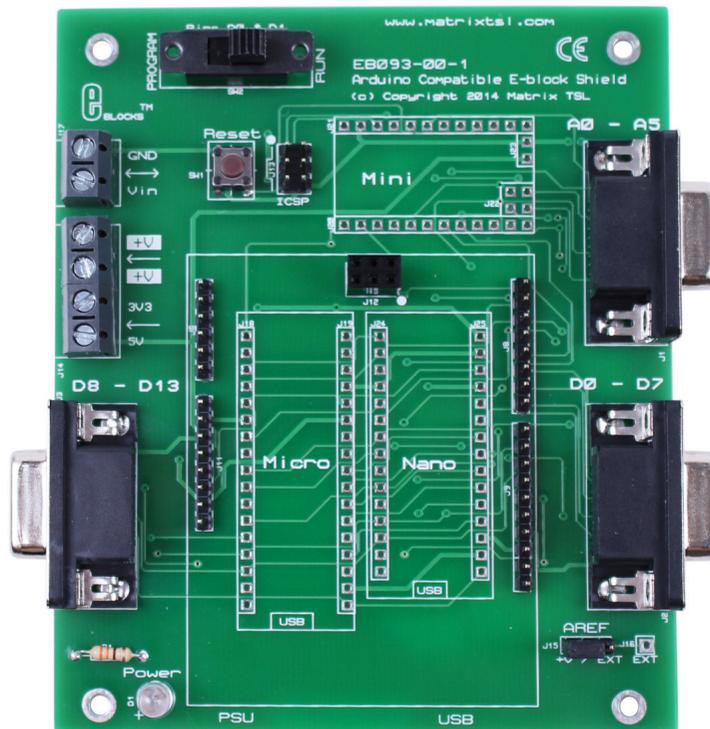


The Hardware

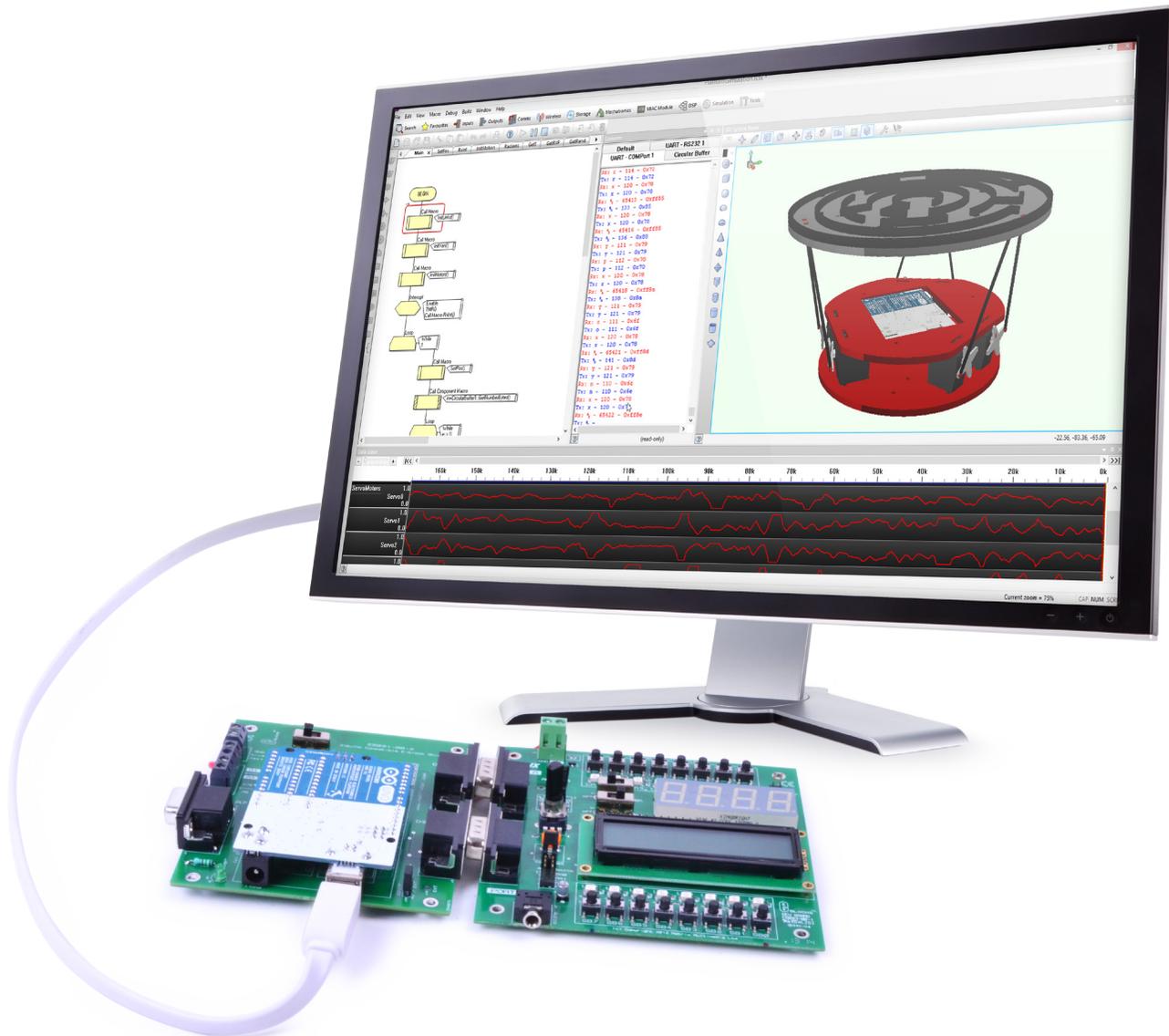
The EB083 Development Board



The EB093 Arduino E-block



Connecting Your Hardware



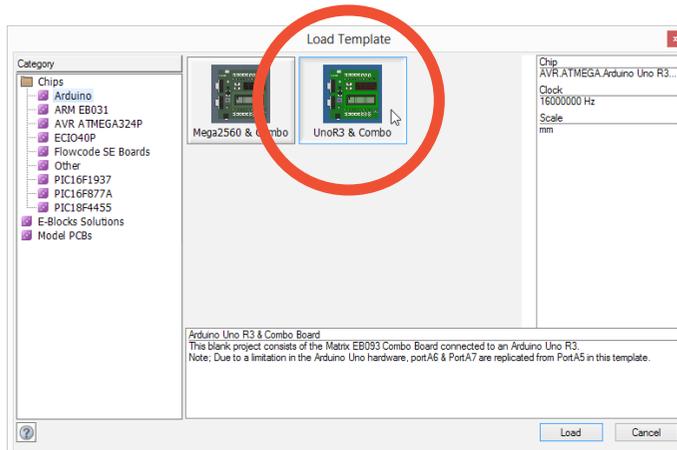
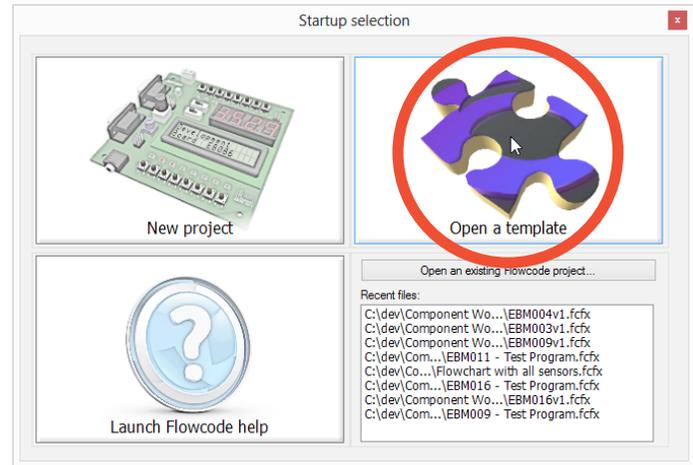
You should connect your EB083 and EB093 together as shown in the photograph above. You will also need to connect your Arduino Uno to the EB093 E-block, and provide a USB connection between the Arduino and your PC running Flowcode.

Notes: You must connect a wire between the +V screw terminals on both boards in order to provide 5V to the Development Board from the Arduino E-block. The GND signal is passed through the 9 way D-Type connectors so a wire is not needed.

For more information relating to the operation of the EB083 and EB093 boards, please visit our website.

Creating your First Program

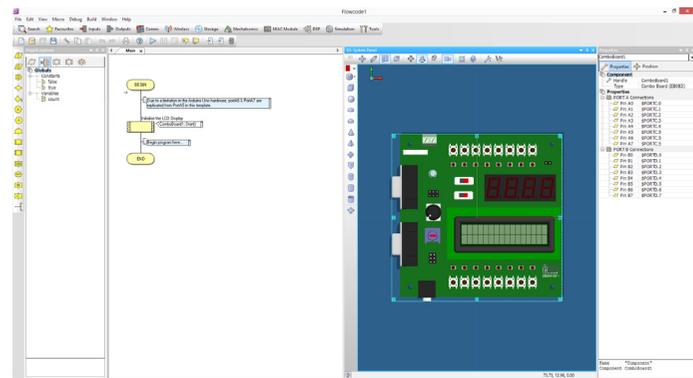
When opening Flowcode you will be presented with a splash screen as seen opposite. From here we can create a new project, open a pre-made template, launch Flowcode help, or open an existing project. Here we will open a template.



From the following screen we can select from one of many templates to suit our chosen application. Templates are pre-made files based on popular configurations of hardware. We at Matrix have developed many templates for you, or you can create your own by saving a Flowcode file into the templates directory.

From the Load Template screen we first select 'Arduino' from the left hand 'Chips' category, and then select 'UnoR3 & Combo'. Press 'Load' to open the template.

Once the template has been opened you will be presented with a Flowcode environment where the EB083 Development Board can be seen on the simulation panel. A basic flowchart will also be presented where the Development Board has been initialised. From here we can create applications that use these components on the EB083 hardware.



Creating your First Program..cont

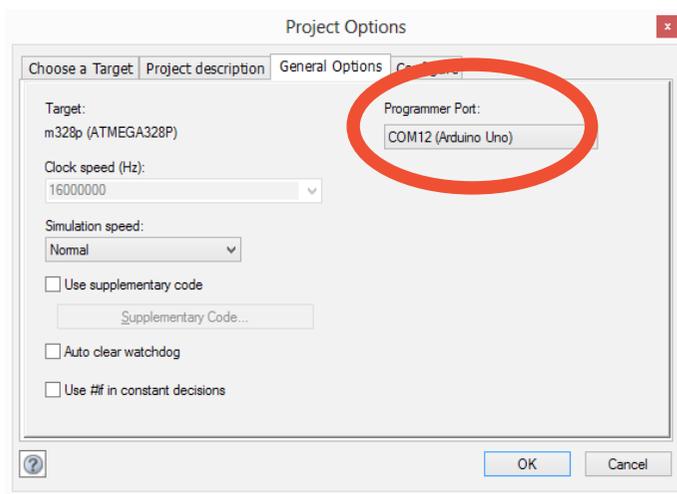
You will notice that in the right hand 'Properties' menu we have pre-configured the hardware connections to match those of an Arduino Uno. If you use a different Arduino programmer you will need to adjust these as per the Arduino datasheet specifications.

The EB083 Development Board features a flexible topology that allows the inputs and outputs to be used in a number of ways depending on user preference. It features 16 LEDs, 16 switches, 1 Light Dependent Resistor (LDR), 1 Potentiometer, a quad 7-segment display and an LCD screen.

In this document we will step through creating 2 examples with the Arduino Uno and EB083 combo board; an LED flasher and a counter which uses the LCD display to present data.

Note: Despite having two full 8bit hardware port connections between the EB083 Development board and the EB093 Shield, the Arduino Uno can only provide 6 GPIO connections on PORTA (A0-A5). Therefore, LEDs and switches LA6-LA7 and SA6-SA7 respectively, cannot be used with the Arduino Uno.

In order to program the Arduino Uno board directly from within Flowcode you must ensure the drivers are installed. The Arduino drivers are provided with the Flowcode download, but are not automatically installed.



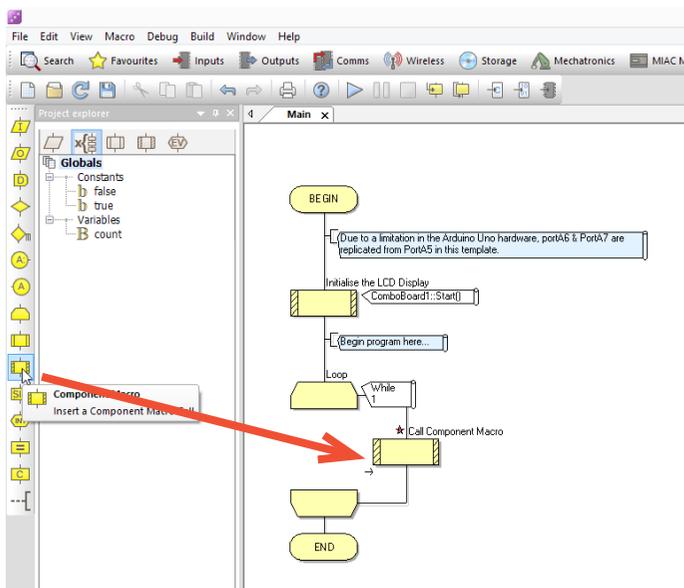
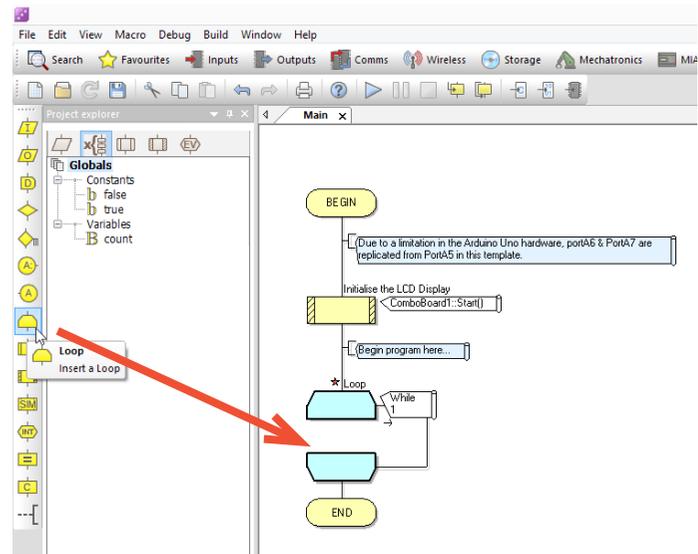
Assuming you selected the default installation path, the Arduino drivers are found in the following folder:
C:\Program Files (x86)\Flowcode 6\Drivers\Arduino

Once the hardware is installed we can make sure Flowcode can detect the Arduino Uno hardware. We check this by selecting 'Build->Project Options->General Options' from the top menu. You will be presented with the screen opposite.

We can see that Flowcode has automatically connected to the Arduino, here on COM12, and that it has recognised the device connected as an Arduino Uno. If you do not see your device selected here in the 'Programmer Port' you must ensure the drivers are correctly installed for your hardware.

Example 1: LED Flash

In the first example, we will make a simple LED flash on our EB083 Development Board. To make an LED flash repeatedly we must first insert a Loop. The While(1) loop used here ensures that the LED will flash forever, as long as the microcontroller is running. Now that the microcontroller is in an 'infinite loop' we can place code within this loop to make the LED flash.



First, we must turn an LED ON. We do this by adding a component macro to our flowchart within the While loop. This is achieved by dragging the Component Macro icon from the left tool bar as seen below.

After we have added our component macro we can configure it by double clicking on it within the flowchart. We want to first turn the LED ON, so we use the "SetLEDState" macro. This allows us to decide which LED we want to talk to, and whether we want it ON/OFF.

The image below shows we have configured to select PORTA (entering 0 in the expression box selects PORTA and 1 = PORTB). We have also selected LED0 (again, the board is zero indexed, meaning the first LED is 0, the second is 1 etc). And finally, we select the state. Entering a '1' will turn the LED ON, and a '0' will turn the LED OFF.

Example 1: LED Flash

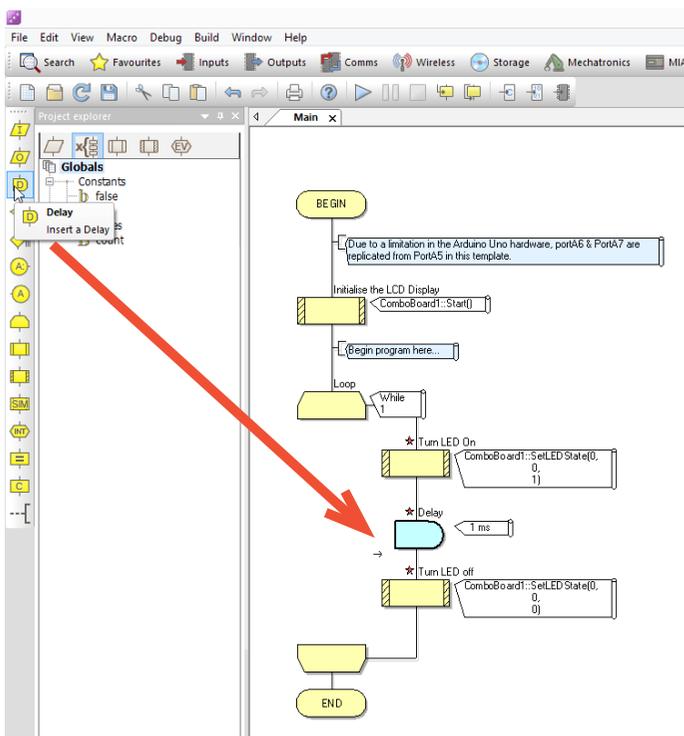
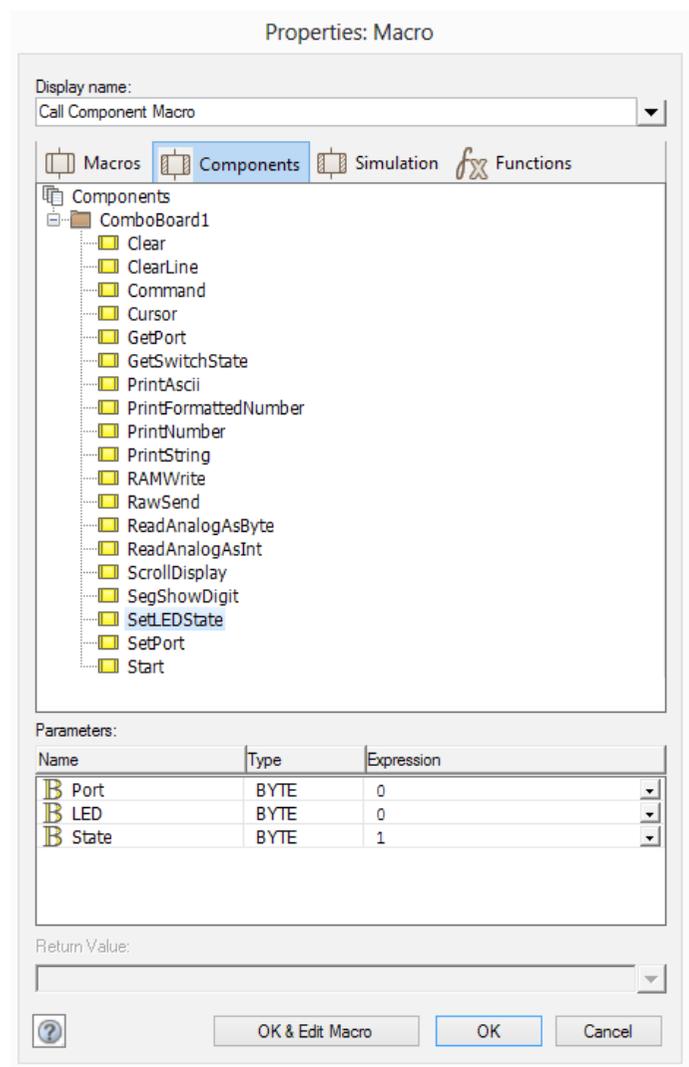
In order to make the LED flash, we must also turn it OFF. We add another component macro as before, but this time configure it to so that the state is set to '0' to turn the LED off.

Note: You must ensure that you are turning the same LED on and off. In this example, Port = 0 and LED = 0.

Our final programming icons to add in this example are delays. Our microcontroller operates at a very fast speed (16 MHz). This means it does 16 million 'things' every second! If it was to turn the LED on and off at this speed we would not be able to see the change occurring. Therefore we add delays into our system. We will add a simple 1 second delay.

To add a delay we follow the same procedure as the component macro, but instead drag a 'delay' icon onto our flowchart as below. As before, we double click our 'delay' icon to configure it for the delay we require. We can see the delay configuration screen below, where we have set a 1 second delay.

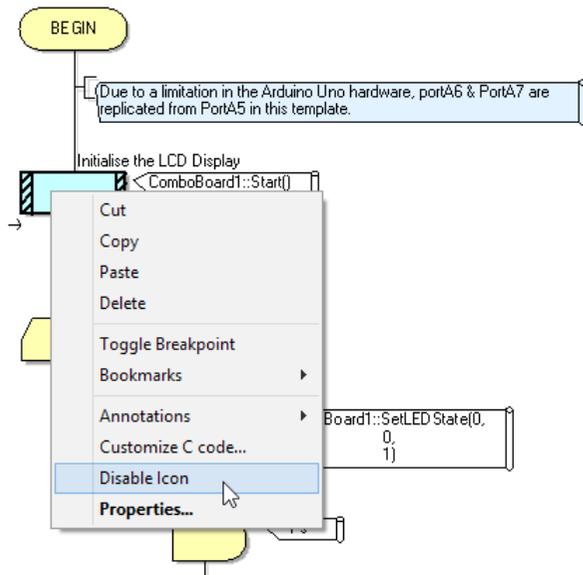
Our final programming icon we must include is another delay. Currently we turn our LED on for 1 second, then we turn it off. However, we do not hold the LED off and so it will turn on again very fast, so it would appear that the LED is always on. We must add another delay icon below where we turn the LED off, which can be seen in the final flowchart on the following page.



Despite adding our final programming icon we will now look at how to disable an icon. By default the Arduino template we opened has one programming icon present, 'Comboboard1::Start()'. This is required by default whenever we use the LCD screen on the development board and as such we included it in the template. Since we are not using this in our LED flasher, we can disable the icon which gives us a good introduction to disabling icons. Disabling icons is exactly the same principle as commenting out code in other languages you may have written in.

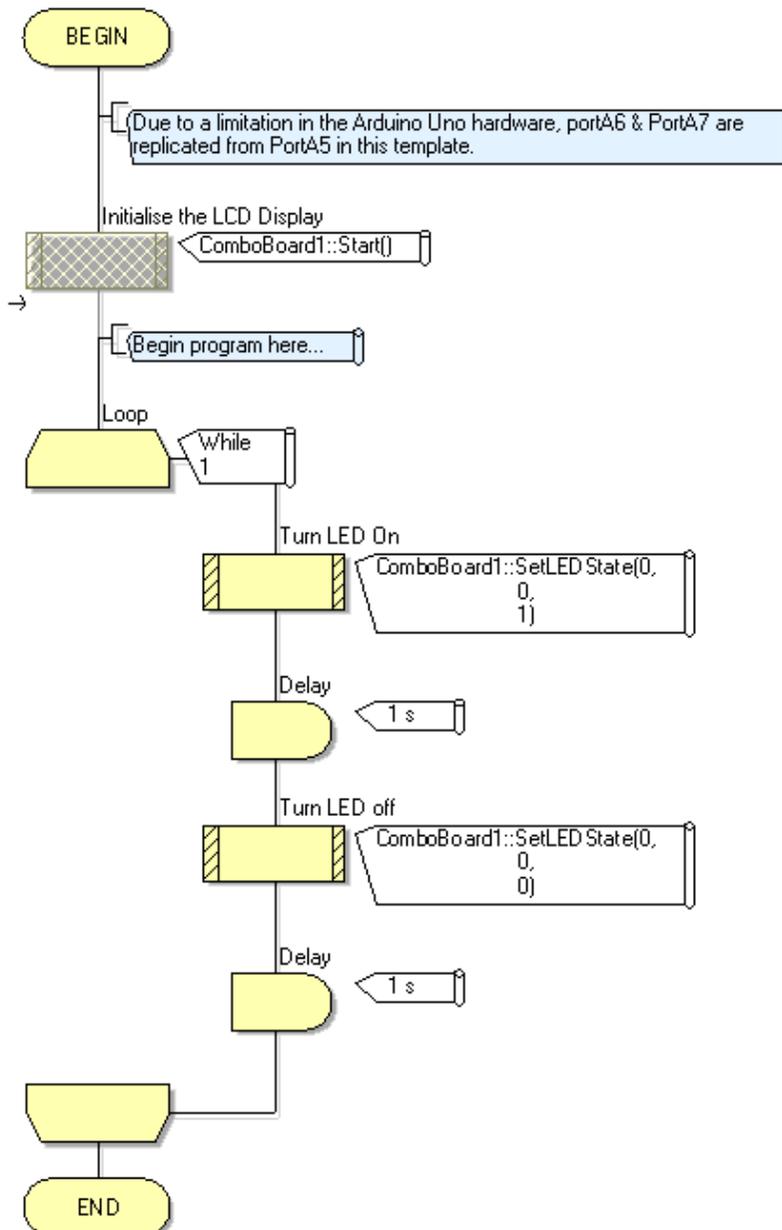
In order to disable code, we right click the icon we wish to disable and select 'Disable Icon', as seen on the following page. Once disabled, our icon will go grey and be hashed out, alerting us that it will be ignored.

Example 1: LED Flash



This is our final flowchart below, and we are now ready to simulate our LED flasher. To simulate we use the buttons seen in the image opposite. The 'Play' button begins our simulation (you can also press F5 on the keyboard).

Once in play mode we can pause or stop our simulation. The two icons on the right allow us to manually step through our program. As previously mentioned microcontrollers operate at very high speeds...faster than we can see! We often need to manually step through each icon during simulation in order to check that our program is working correctly.

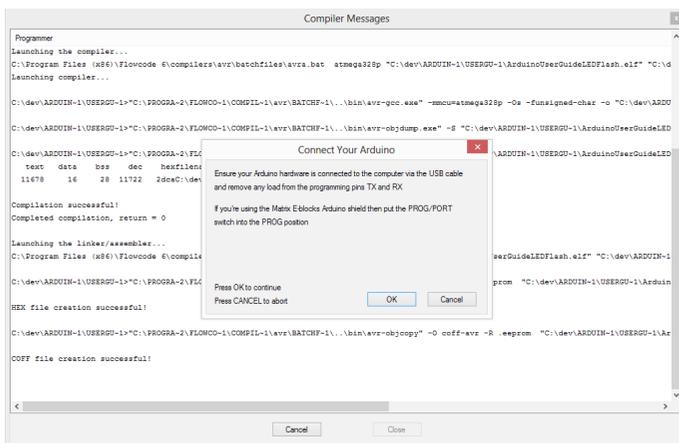
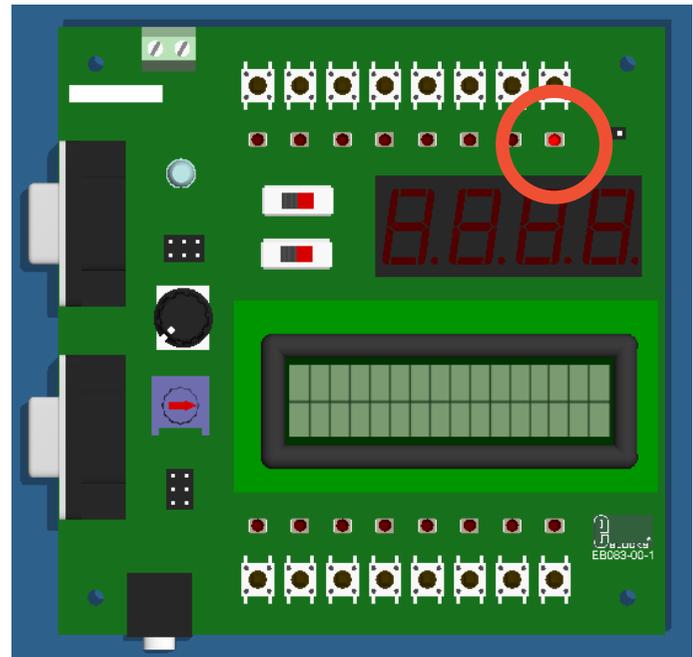


Example 1: LED Flash

If we run our simulator we will see that LED A0 flashes at 1 second intervals, in the image opposite A0 is highlighted for us.

Now that our program has been simulated and verified as operating correctly we will download our program to hardware. In Flowcode it is easy to download our program to the chosen hardware. When downloading code to our device we again have a shortcut toolbar, seen below.

There are 3 icons in the programming section of the toolbar. The left icon compiles our flowchart into the 'C Language'. The middle icon compiles the Flowchart into 'Hexadecimal'. These two options are useful if you are using Flowcode to program your system, but an external programmer to flash your hardware. We are using Flowcode to flash our hardware, and as such use the right hand icon. In order to compile our code and program our Arduino we only need to click the right hand icon.



Once we select to program the device a new window will appear on the screen detailing the compilation and programming progress. Due to limitations of the Arduino hardware, we need to manually disconnect the Development Board from the Arduino during programming each time we program the microcontroller.

To save you manually disconnecting the EB083 hardware we have placed a switch on the EB093 Arduino E-block. During compilation users will be alerted when to ensure that the switch is positioned in the 'Program' position, thus isolating the Arduino from the Development Board. After the program has completed you will need to return the switch to the 'Run' position.

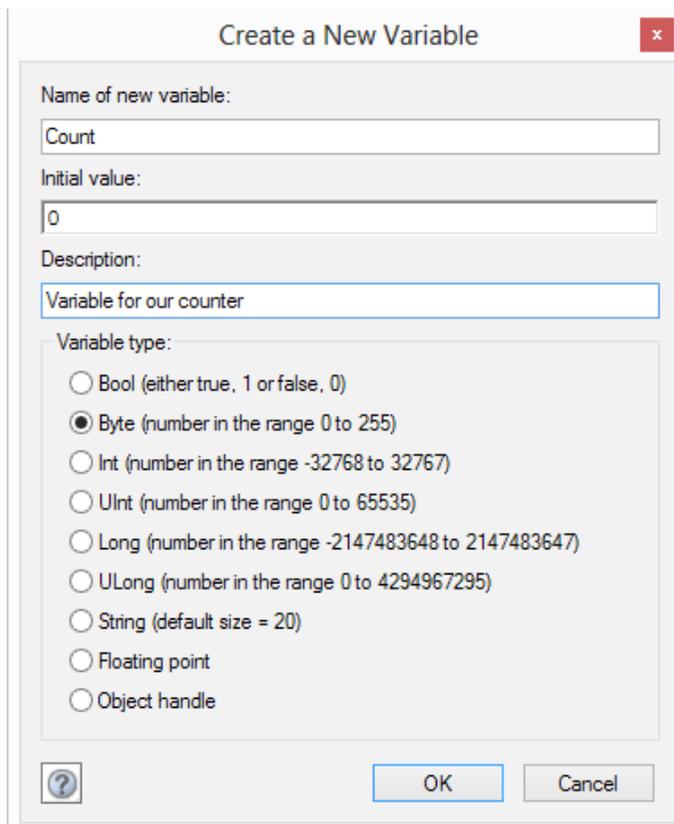
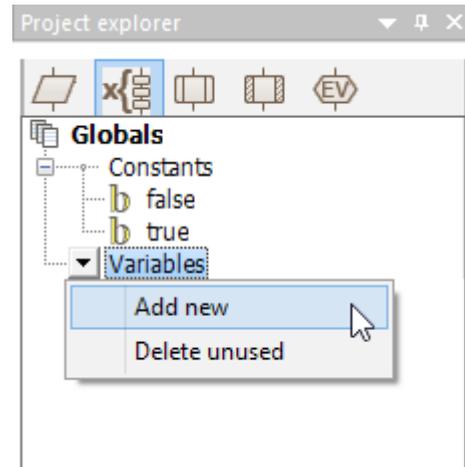
Finally, you should verify that the program is working as desired on hardware. If it is not, ensure that you have followed every step in this tutorial, and that your flowchart is correctly configured.

Example 2: LCD Screen & Counter

In this example we will demonstrate how to interface with the LCD screen on the EB083 Development Board. The LCD display is connected to the lower port, and as such is connected to D0 – D7 of our Arduino Uno.

When interfacing with an LCD screen, we soon realise the power of the component macros built in to Flowcode 6. Interfacing with an LCD is not a simple task when writing in C code, but component libraries and flowcharts make this task much simpler.

This example program will also demonstrate the use of variables to create a counter. To create a variable within Flowcode we use the Project Explorer, as seen opposite:



When we add a new variable we must configure it. We do this by giving it a name, an initialisation condition, a description and selecting a data type. In the example below we have called our variable 'Count', initialised it to '0', provided a description, and we have selected that the data type is a 'Byte'.

This means that the maximum value we can store in our variable is 255. If we count beyond this number the variable will roll back around to zero.

Example 2: LCD Screen & Counter

Now we have our variable we can create our program, which will increment a counter each time it loops through the program, and display this counter value on the LCD screen.

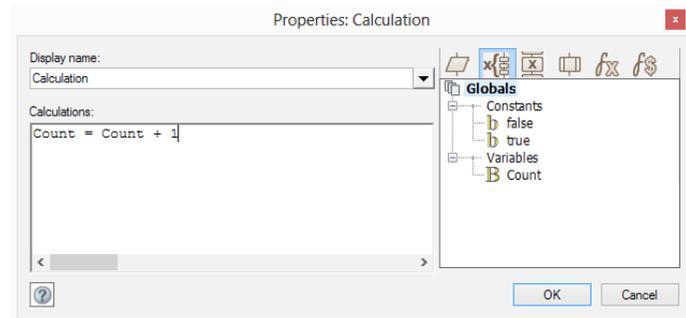
Seen as this is our second example, we will provide less detail and let you work through this yourself a little more.

The function 'ComboBoard::Start()' which is provided as default in the template flowchart is used to initialise the LCD display on the Development Board. In example 1 we disabled this icon as we did not need it. Here, we will leave it enabled. We need to initialise the display only once per program.

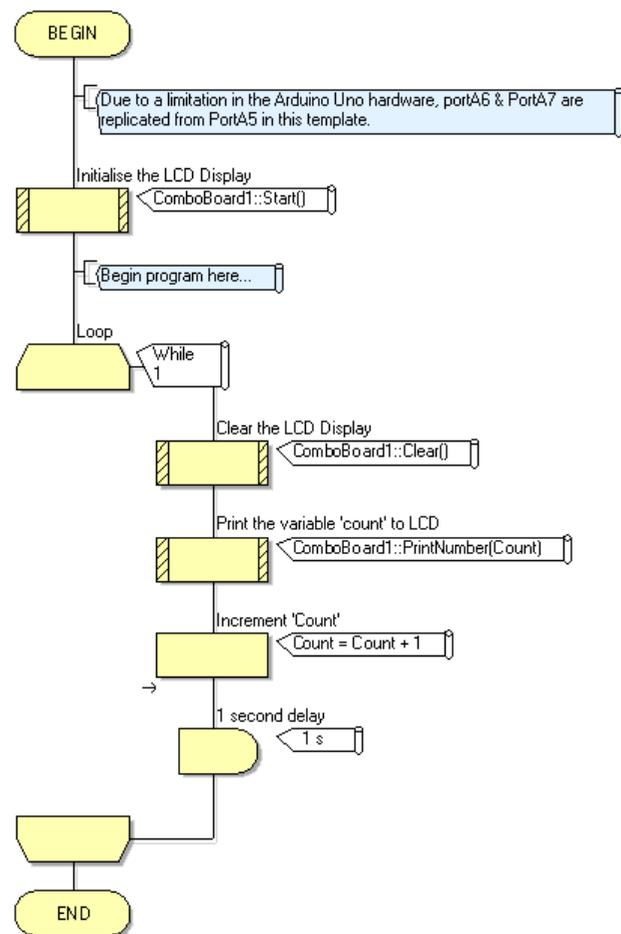
With the display initialised we can write a variable to it. Drag a component macro onto the flowchart and double click to configure it. Select the macro 'PrintNumber'. This will allow us to print a value to the LCD screen. We can print either a static value, by entering a numerical value into the 'Expression' box, or we can pass a variable to it. Here we will pass our variable 'Count'. Therefore, whenever that function is called the value of Count will be shown on the LCD display. You must also add another component macro and configure it to clear the LCD screen each time the loop executes, to remove the previous value and allow us to write the new value.

Once you have added the 'PrintNumber' and 'Calculation' macros you should be able to add the other necessary components to create the same flowchart as seen opposite.

Download the code to your Arduino microcontroller and confirm that it works as desired. If it does not, go back and check that your code is correct. We have also provided an example flowchart on our website that you may download if required.



The final icon that you have not yet used is the 'Calculation' command. This allows us to perform mathematical operations on variables. Since we are creating a counter here we want to add the value of '1' to the variable 'Count' each time the loop iterates. Drag a calculation box into the flowchart and double click to configure. Once open, insert the code as seen in the figure above.





Matrix Technology Solutions Ltd.
The Factory
33 Gibbet Street
Halifax, HX1 5BA, UK

t: +44 (0)1422 252380
e: sales@matrixtsl.com

www.matrixtsl.com