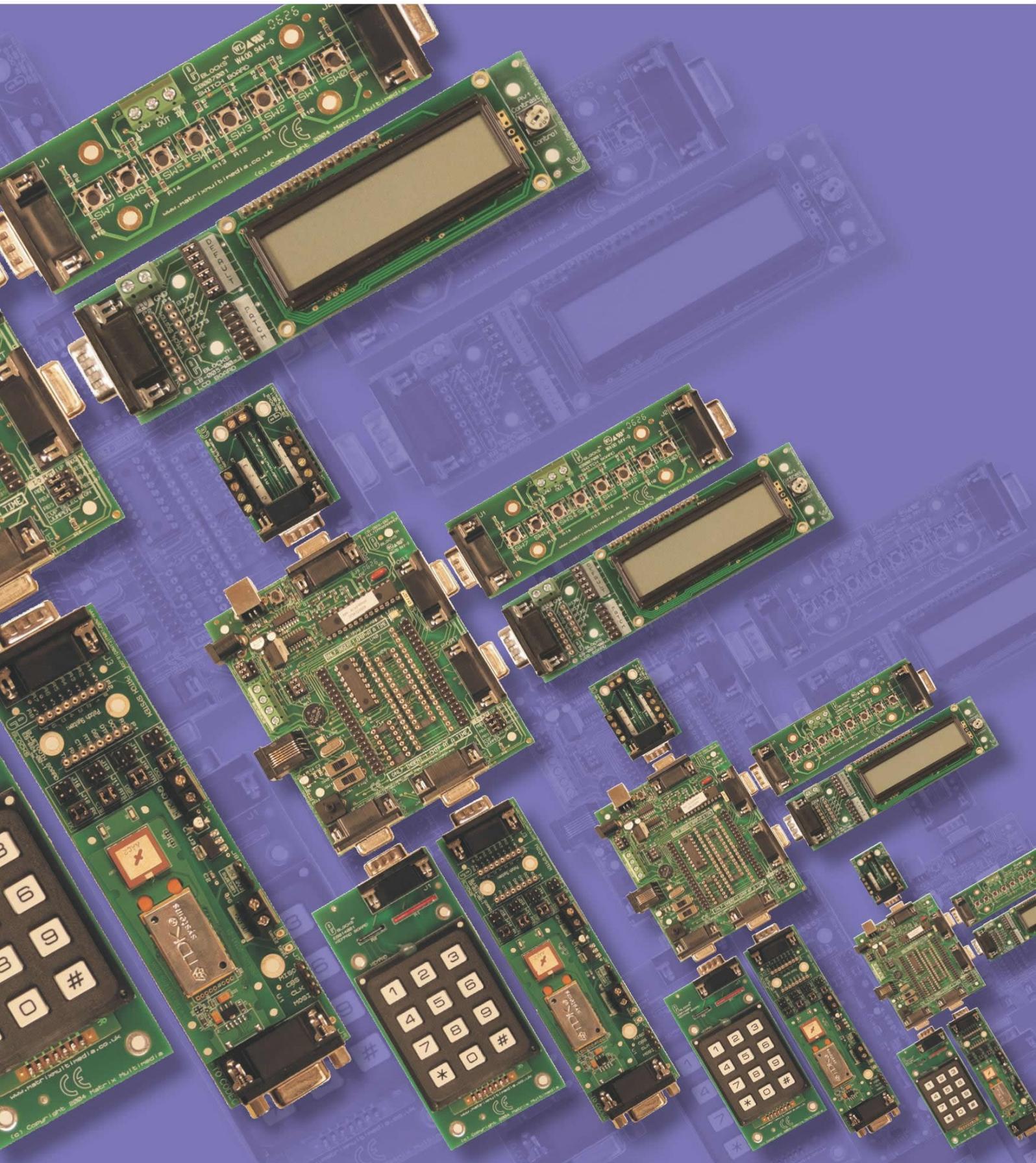


# MATRIX

getting started guide

## E-Blocks Build Your Own PLC Bundle



# Flowcode

EB867-80-1

## E-Blocks PLC Bundle

### Installing Flowcode

Instruction for installing Flowcode can be found inside the installation booklet located inside the Flowcode DVD case.

Before starting with the course it is recommended to update your version of Flowcode to the latest released version. This allows for the latest bug fixes and components to run on your machine. The latest version of Flowcode can be found by visiting the Matrix TSL website and clicking on the Flowcode page.

<http://www.matrixtsl.com/Flowcode3a-X.php>

### Getting Started with Flowcode

There is a free online course available for helping with getting started with learning Flowcode. This course covers basic principals through to designing your own programs and programming the devices. It is recommended that you take time to go through this course before proceeding with the bundle exercises to give you a better grasp of what the Flowcode program is doing.

The online course is available from the learning centre on our website or by visiting the following address:

[http://www.matrixtsl.com/lc\\_microcontroller.php](http://www.matrixtsl.com/lc_microcontroller.php)

### Flowcode Examples

A number of pre-made example files are available for download from the main Flowcode page on the Matrix TSL website. These files are also located on the Flowcode CD. Before the example files can be used you must first copy them into a folder on your hard drive.

The example programs referenced in this bundle can be found at the following web address:  
[http://www.matrixtsl.com/lc\\_bundle\\_manuals.php](http://www.matrixtsl.com/lc_bundle_manuals.php)

Or by clicking the bundle manuals link from the Learning Centre area of our website.

### Flowcode Help

There is a help file available that covers all the main features of Flowcode. This help file can be accessed by clicking the question mark icon in the main Flowcode toolbar or alternatively clicking the help menu and selecting contents.

There are also help files available for each and every component in Flowcode which explain the functionality of the component and the component macros. The component help files can be found by selecting the component on the panel and the clicking the Help button in the properties toolbar.

### General Support

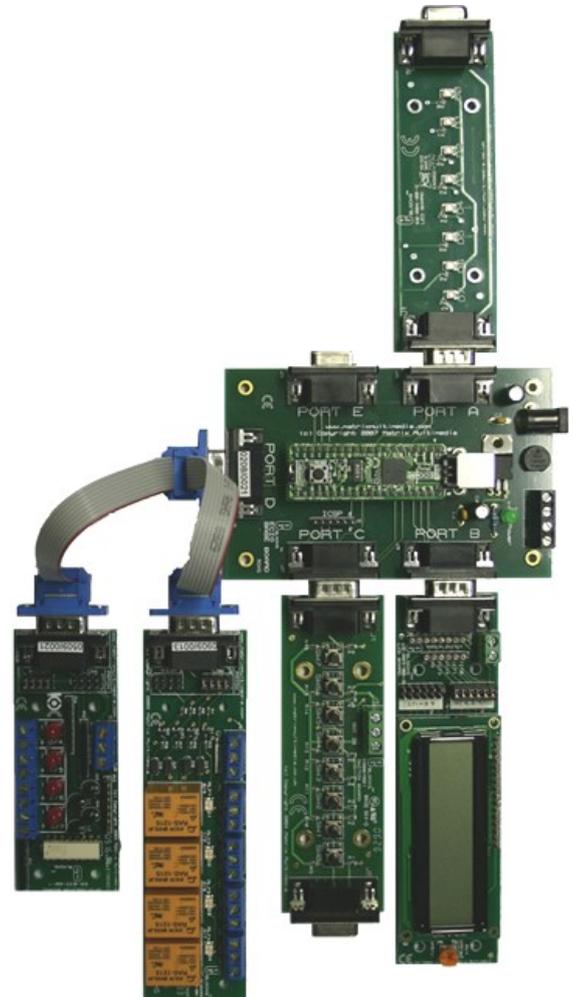
Support for frequently encountered problems can be found online on our FAQ's site. Our online forums can also be used as a general discussion area or for help or advice.

# Wiring & Testing

## EB867-80-1 E-Blocks PLC Bundle

To setup your E-Blocks for use with the example programs you must perform the following actions:

- Insert the ECIO into the EB061;
- Connect up the E-Blocks as shown on the right.
- The Opto Isolator, Relay, LCD and Switch E-Blocks all need to be connected to the VCC terminal on the EB061 via single core wire.
- The jumper on the Relay E-Block should be set to Low.
- The jumper on the Opto Isolator E-Block should be set to High.



Before you can begin you must install the driver for the ECIO using the ELSAM CD or by visiting the Matrix TSL website: <http://www.matrixtsl.com>

The system can be tested by compiling and sending one of the example programs to the hardware. This is done by opening one of the example files in Flowcode and then clicking the compile to chip button.

The example Flowcode files contain a correct configuration so you will not have to modify the configuration to allow them to run on the hardware.

Any program you create from scratch will have to be configured to use the ECIO device as the target rather than using the raw PIC 18F4455 device. This ensures that the ECIO programming tool is used and that the ECIO configuration is automatically added to your Flowcode program.

Example 1 is a good test file as it is very basic and simply checks each of the E-Blocks functionality.

# Flowcode Components

## EB867-80-1 E-Blocks PLC Bundle

The Flowcode components can be added to your program by finding the component you desire in the Common section of the Flowcode component toolbar and clicking it to add it to the Flowcode panel.



**LCD Component Icon**



**LED Component Icon**



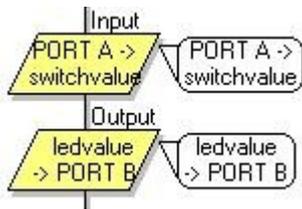
**LED Array Component Icon**



**Switch Component Icon**



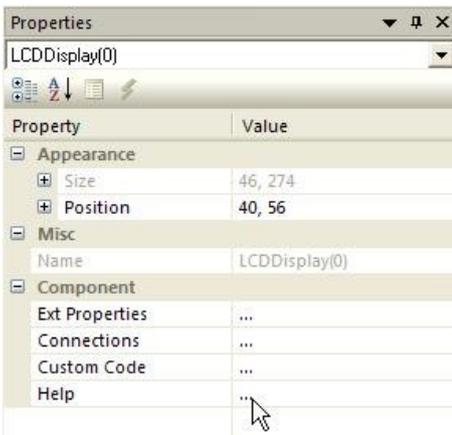
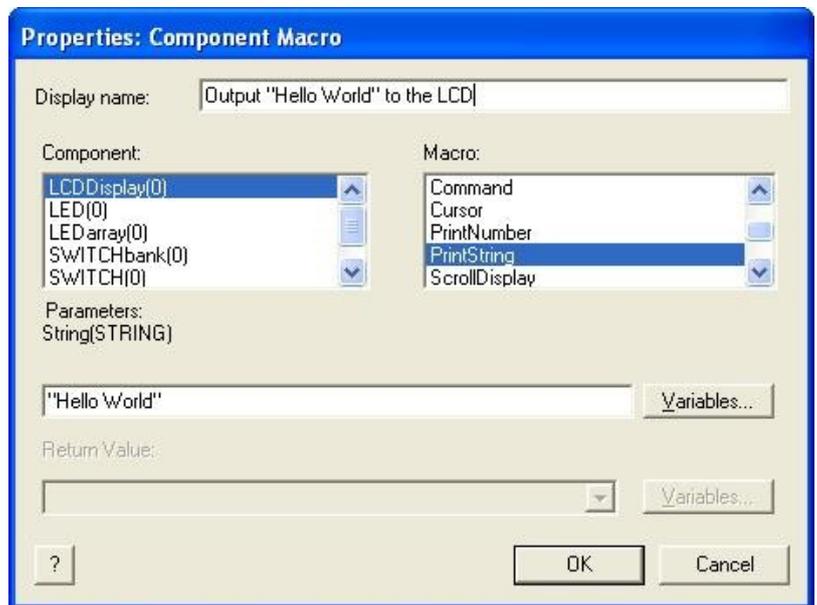
**Switch Bank Component Icon**



Digital inputs or outputs can be read or controlled by using the functions associated with the switch and led components or by using the input and output flowchart icons directly.

For each component there are a set of predefined component macros available. To see the available functions you can simply add a component macro flowchart icon to your program and have a look at the list.

Parameters are variables or constants that are passed to the function whereas return values are normally used for data collection or error checking.

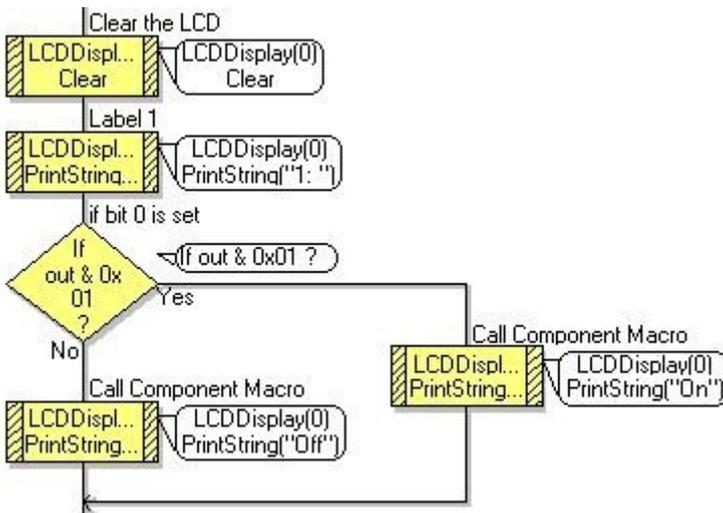
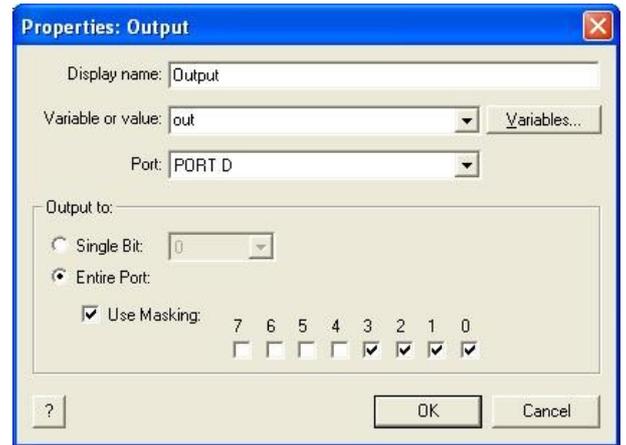
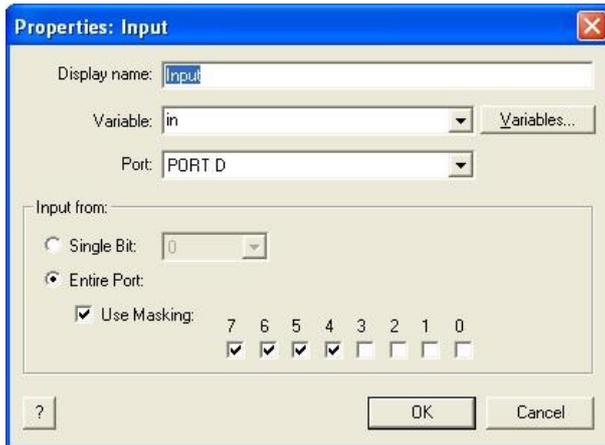


Help files are available for all of the components in Flowcode by selecting the component on the panel and then clicking the Help button in the properties side bar. For Vista and Windows 7 machines you will have to download the Microsoft Help32 runtime to allow you to open and view the help files.

# Example 1

## EB867-80-1 E-Blocks PLC Bundle

The input and output icons use masking when reading and writing the data. This allows the port's data direction to remain untouched. For example if you read the entire port using an input icon then you would convert the entire port to an input which would have the effect of switching off the relay outputs.



Next we break down the value into the 4-bits and print out to the LCD the logical state of each bit.

We start with a clear display, then print a label to state the bit number.

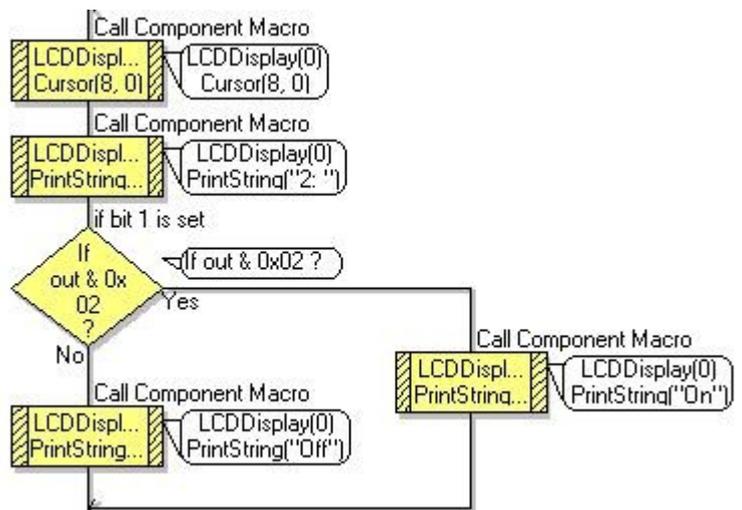
After that we use a logical AND command to mask off the bit from the 8-bit variable. If the single bit is set then the decision will return yes, otherwise the decision will return no.

0x01 is the hexadecimal representation of the binary value 0b00000001.

For the next bit we have to position the cursor on the LCD to specify where we are going to be printing the next characters.

Once we are in the correct position on the display we again print out a label for the bit number and then perform the decision again.

This time we are using 0x02 for the bit mask. The binary representation of this is 0b00000010.

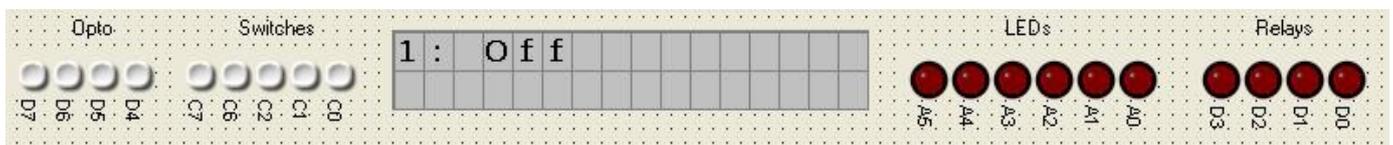


# Example 1

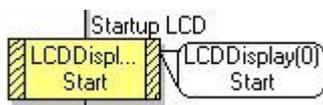
## EB867-80-1 E-Blocks PLC Bundle

Example 1 demonstrates how to read a value from the opto isolated inputs, manipulate the input value and then output the value back to the relays. The LCD also monitors the status of the inputs allowing their state to be shown on the screen.

The examples all come with a pre made panel to represent all of the components used by the PLC bundle.

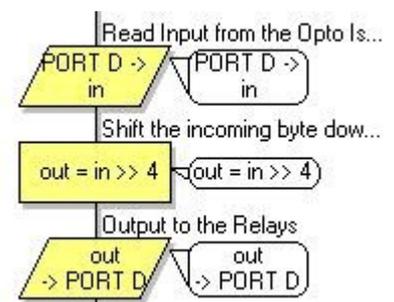


LEDs and Switches have been used to represent the opto isolated inputs and the relay outputs. This is because the inputs and outputs are basically digital signals that are equivalent to reading a switch or outputting a value to a LED.



Component macro to start up the LCD and get it ready for communications.

Here we read the 8-bit binary value from Port D and store into a variable named in. As the opto-isolators are connected to the upper 4-bits of Port D we have to shift the bits down to the lower 4-bits before we can send the value back out to the relays.



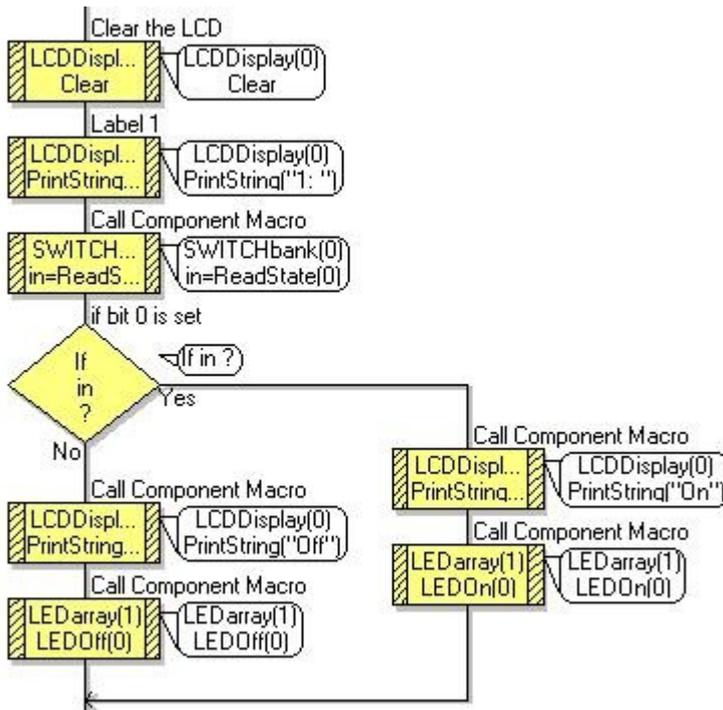
When testing the example on the hardware you will be forwarding the value from the 4 opto isolator inputs to the Relay outputs. The Opto isolators allow for inputs free from noise and any nasty signals that may damage your microcontroller if connected directly. The Relay outputs allow for a high voltage, high current or both depending on your output requirements. These can be used for switching mains or other signals which you would not want to connect directly to your microcontroller.

The Opto isolators can be made to input a logic 1 by connecting a power wire from ground to -ve and +V to +ve on one of the Opto input channels.

# Example 2

## EB867-80-1 E-Blocks PLC Bundle

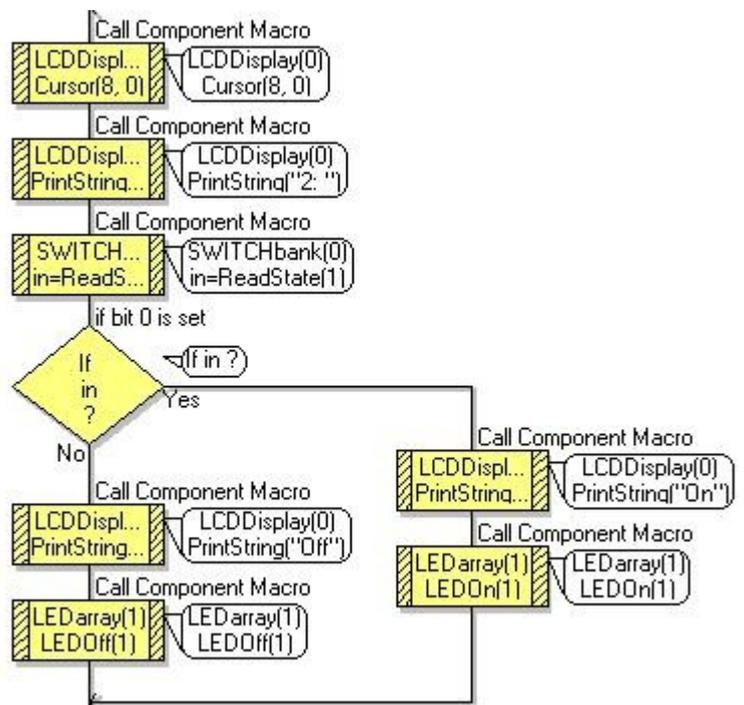
Exercise 2 performs the same functionality as exercise 1 but this time uses component macro calls rather than the direct I/O port calls.



We now read a single bit at a time rather than reading the entire 8-bit port at once. Reading the switch at position 0 will return the value of the opto isolator connected to pin D4.

If the bit is set then the decision will return yes and we can output a 1 to the relay at position 0 which relates to pin D0.

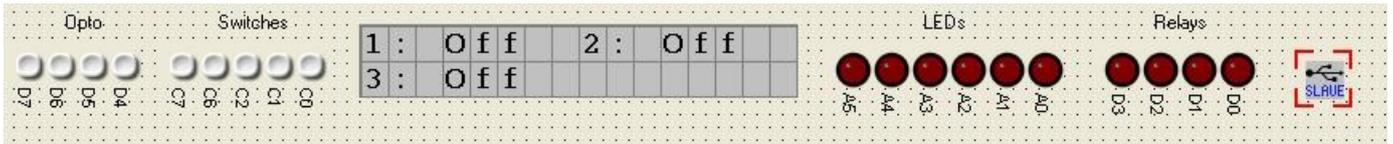
Again we use the component macro method to read the value from the opto isolator input and pass the value onto the relay using another component macro function.



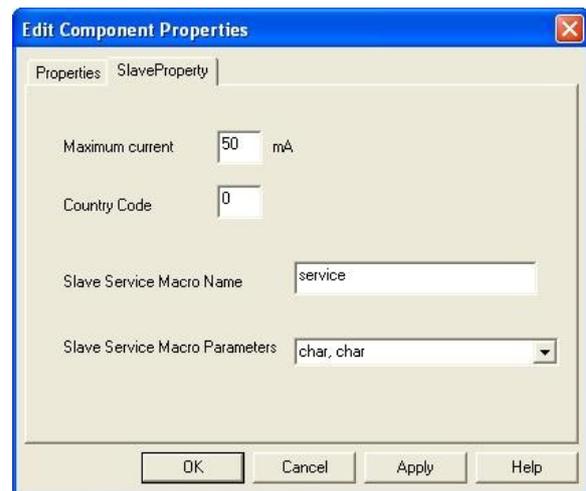
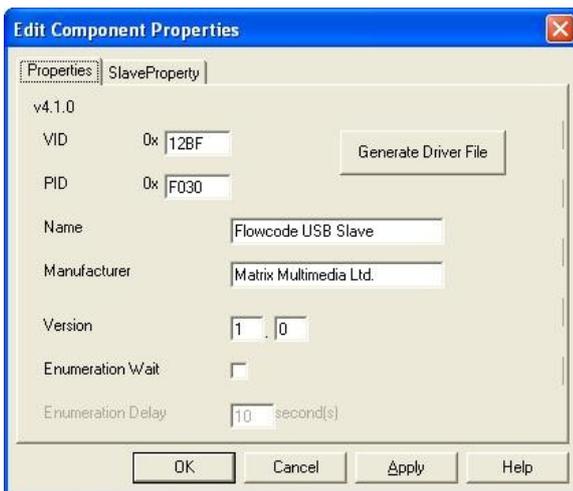
# Example 3

## EB867-80-1 E-Blocks PLC Bundle

Exercise 3 allows access to the relays and opto isolators via the USB connection to the ECIO device. A visual basic application is used to control the relays and read back the values of the opto isolators. The visual basic application source code can be modified using the free version of Microsoft VB 2008 Express.

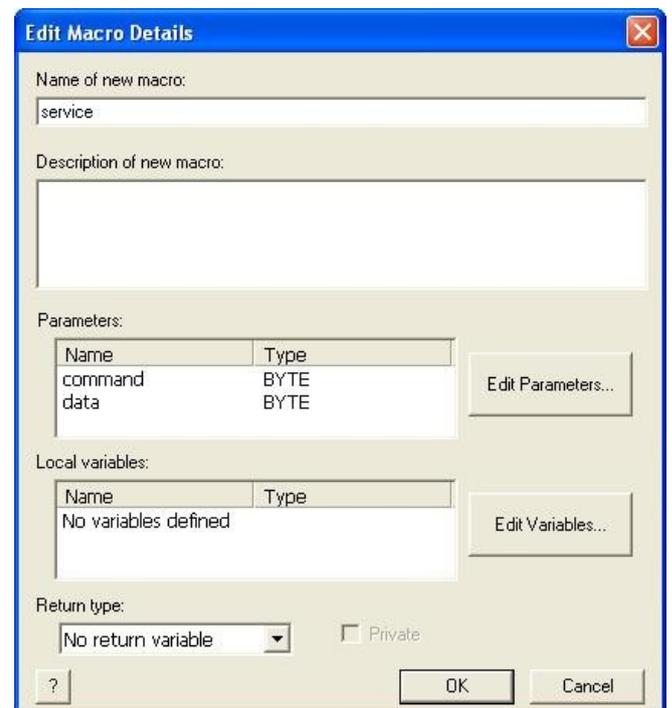


The USB slave component allows you to expose portions of your program to a computer. The component properties must be set up as shown to reference the service macro that will be ran when ever there is communications on the USB.



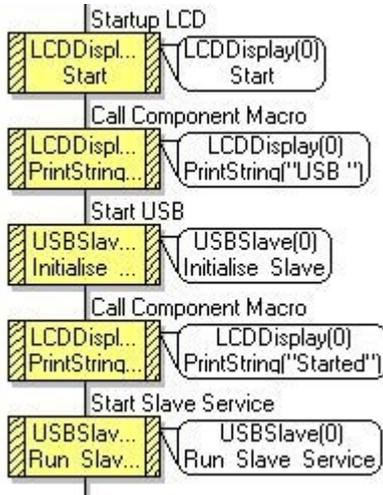
The service macro is given two byte parameters to match up with the settings specified in the USB component properties. A driver file for the USB device can be generated by clicking the Generate driver file button located in the USB component properties.

The names command and data are given to the parameters of the service macro. These will be used to tell the microcontroller which command to process and if necessary which data value to use.

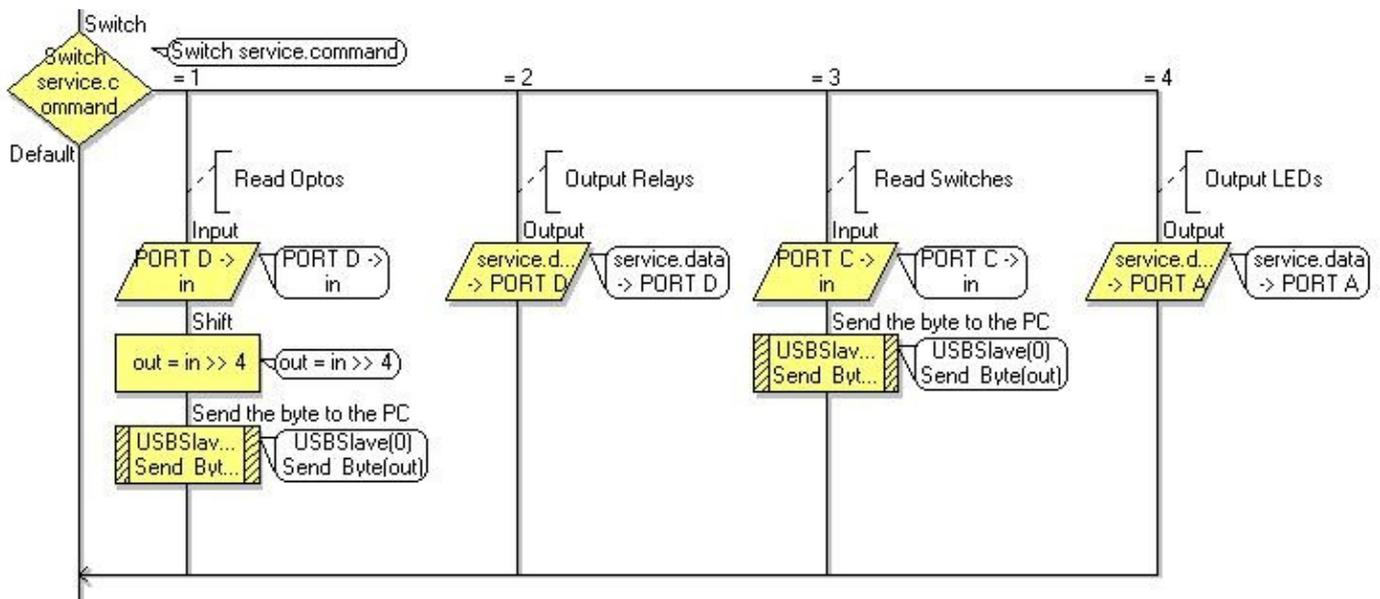


# Example 3

## EB867-80-1 E-Blocks PLC Bundle



The main macro simply initialises the LCD, starts up the USB component and then starts the USB slave service running. The USB slave service is what is known as a blocking function. This means that the code will cease running when this function is called. When a USB transaction is received the slave service macro will be ran. If you need to come back to your main macro at any time then you can call the stop slave service macro from inside the service routine. This is useful if you want to be able to connect a device up to USB mid operation to change some parameters etc.



The above picture shown the contents of the slave service macro. As we can see there is a switch case that will decide what to do based on the first parameter. This is the first byte from the VB send and receive function. There is also a non mandatory second parameter that is used to control the output data that is sent to the relays and LEDs.

The commands are all numeric and are triggered by clicking the appropriate buttons from the visual basic code. More parameters can be added as required and integer values can also be used. Please remember to update the USB component properties if you update the service macro parameters otherwise you will receive a compilation error when you try to download the program to your ECIO.

# Example 3

## EB867-80-1 E-Blocks PLC Bundle

```
'Set the identifier
```

```
sID(0) = Asc("v")
sID(1) = Asc("i")
sID(2) = Asc("d")
sID(3) = Asc("_")
sID(4) = Asc("1")
sID(5) = Asc("2")
sID(6) = Asc("b")
sID(7) = Asc("f")
sID(8) = Asc("s")
sID(9) = Asc("p")
sID(10) = Asc("i")
sID(11) = Asc("d")
sID(12) = Asc("_")
sID(13) = Asc("f")
sID(14) = Asc("0")
sID(15) = Asc("3")
sID(16) = Asc("1")
sID(17) = 0
```

```
Dim lRetVal As Integer
lRetVal = ECIO_Open(0, sID(0))
```

```
'Declare the transaction variables
```

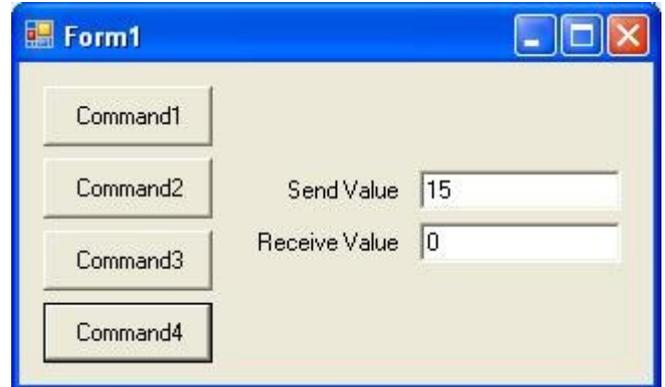
```
Dim Outgoing(0 To 2) As Byte
Dim Incoming(0 To 2) As Byte
Dim SentCnt As Long
Dim ReceivedCnt As Long
```

```
'Command 2 - Write to relays
```

```
Outgoing(0) = 2
Outgoing(1) = Send.Text
```

```
'Perform send and receive transaction
```

```
ECIO_Transmit(Outgoing(0), 2, SentCnt, Incoming(0), 1, ReceivedCnt, 10)
```



The Visual basic code was created using the free version of Microsoft VB 2008 Express. The on load function controls the start-up operation of the device. By default this assigns the values for the default USB PID and VID descriptors and then attempts to connect to the device.

If the start-up connection was successful then we can use the buttons to send various commands to the ECIO device. Above is the example of sending out the command complete with data whereas below is the example of sending out a command and receiving back data.

```
'Declare the transaction variables
```

```
Dim Outgoing(0 To 2) As Byte
Dim Incoming(0 To 2) As Byte
Dim SentCnt As Long
Dim ReceivedCnt As Long
```

```
'Command 1 read Optos
```

```
Outgoing(0) = 1
```

```
'Perform send and receive transaction
```

```
ECIO_Transmit(Outgoing(0), 2, SentCnt, Incoming(0), 1, ReceivedCnt, 10)
```

```
Receive.Text = Incoming(0)
```

# Troubleshooting

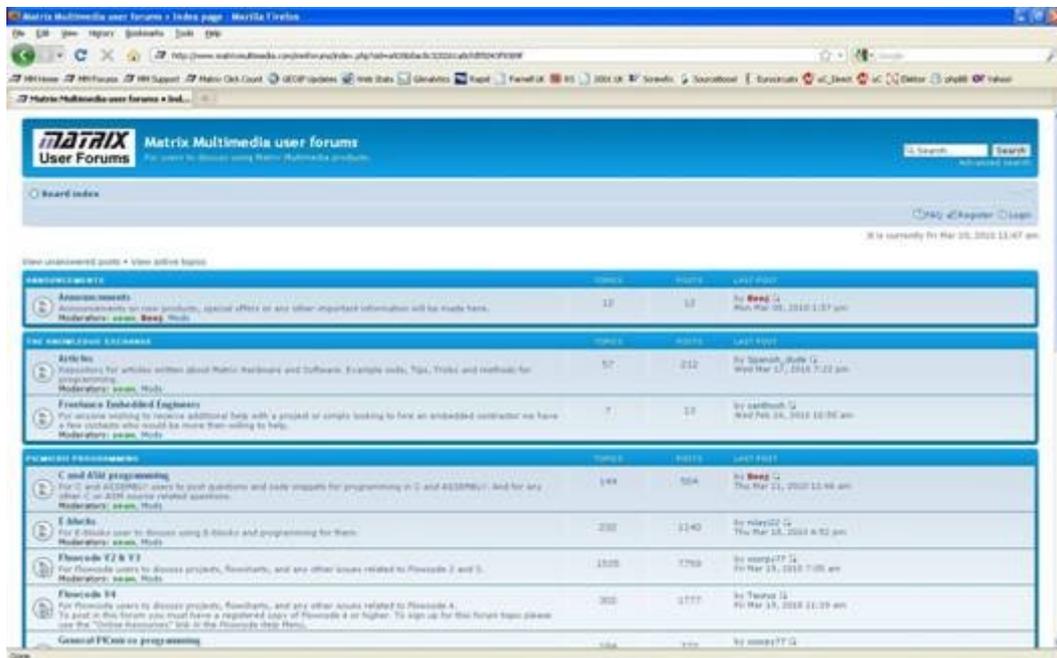
## EB867-80-1 E-Blocks PLC Bundle

If you are having any problems getting up and running with any of the examples or any of the Flowcode components then the first port of call is to ensure you have your boards plugged together and wired correctly. As a rule of thumb any board with a screw terminal and a +V marking should be connected via a single core wire to the +V screw terminal on the corresponding Multiprogrammer.

If you do run into any problems then there is help and advice available from our online user forums located here:

<http://www.matrixsl.com/mmforums/>

The Articles section contains quite a few examples, as well as hints and tips to aid in your applications



There is also an online video demonstrating Flowcode, available from the videos section of our website: [http://www.matrixsl.com/lc\\_videos.php](http://www.matrixsl.com/lc_videos.php)

# Other Products

## EB867-80-1 E-Blocks PLC Bundle

Matrix TSL is a leading global technology company. Over the years we have developed a portfolio of award-winning products which have applications in Education, Industry and in the home.

Learning is at the heart of much of what our company does, and the philosophy of all Matrix learning products is based on 'learning by doing'. Each year Matrix spends around 25% of turnover on research and development to ensure that our learning and development resources are world class.

### MIAC PLC

MIAC (Matrix Industrial Automotive Controller) is an industrial grade control unit which can be used to control a wide range of different electronic systems including sensing, monitoring and automotive. It has a number of applications in industry and learning.



### Formula Flowcode

Formula Flowcode is a robot vehicle which is used to teach robotics, and to provide a platform for competing in robotics events.



### ECIO



ECIO devices are powerful USB programmable microcontrollers with either 28 or 40 pin standard DIL (0.6") footprints. They are perfect for student use at home, project work and building fully integrated embedded systems.

### FlowKit

The FlowKit allows for in circuit debugging directly from within Flowcode. This is the same ICD debugging feature that is included with our version 7 EB006 Multiprogrammer boards.



### Flowcode + E-Block Technology bundles



Matrix TSL technology bundles are based on a combination of two of our most popular products, E-Blocks and Flowcode.



### Other bundles in the range

- Easy Mobile Communications Pack
- Easy Zigbee Pack
- Easy RFID
- Easy Internet
- Easy CAN Bus
- Easy GPS
- Easy USB
- Build your own PC Interface
- Build your own PLC
- Build your own Data-logger