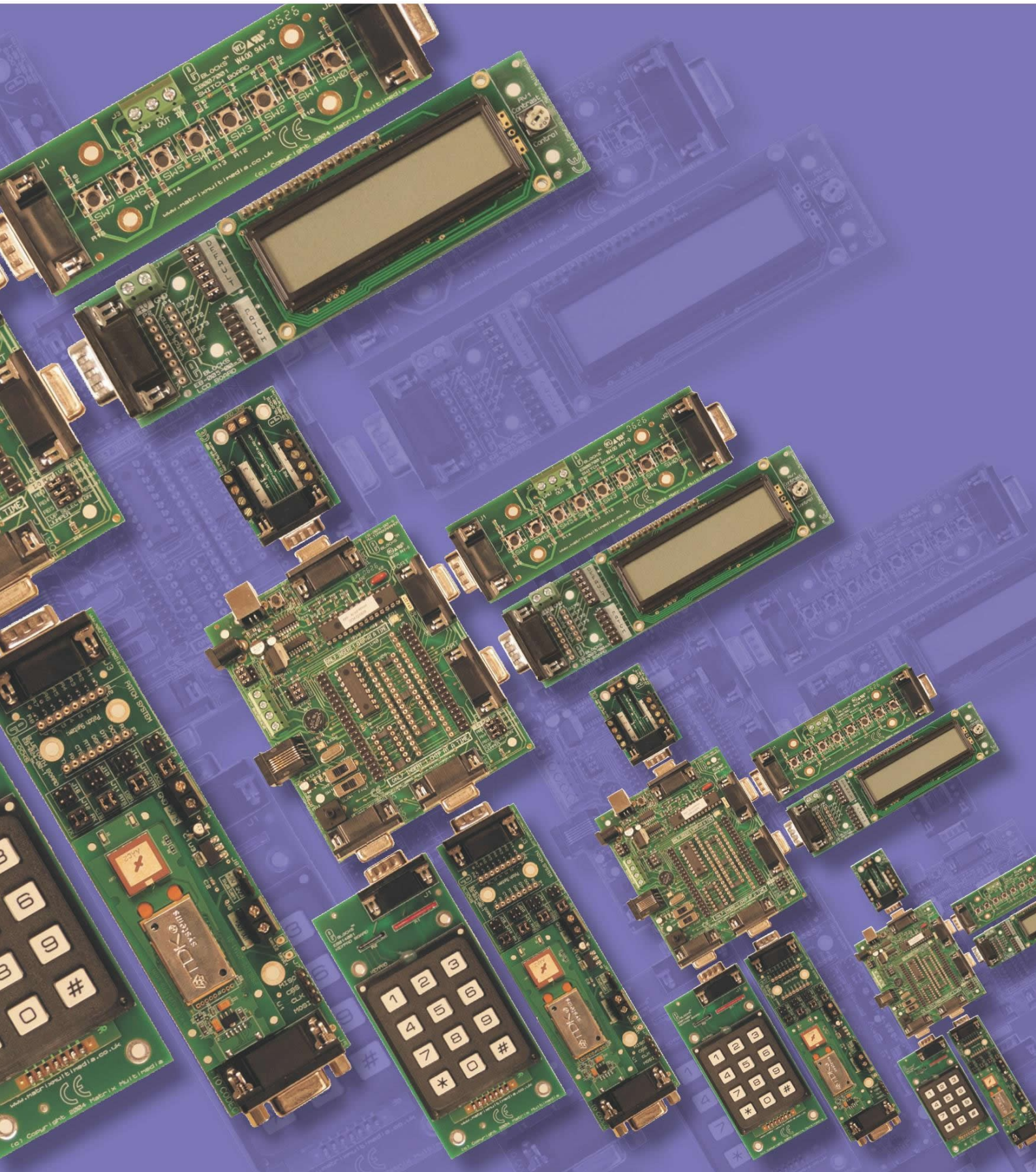
MATRIX

getting started guide

E-Blocks USB Bundle



Flowcode

EB896-80-1
E-Blocks USB 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

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

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.

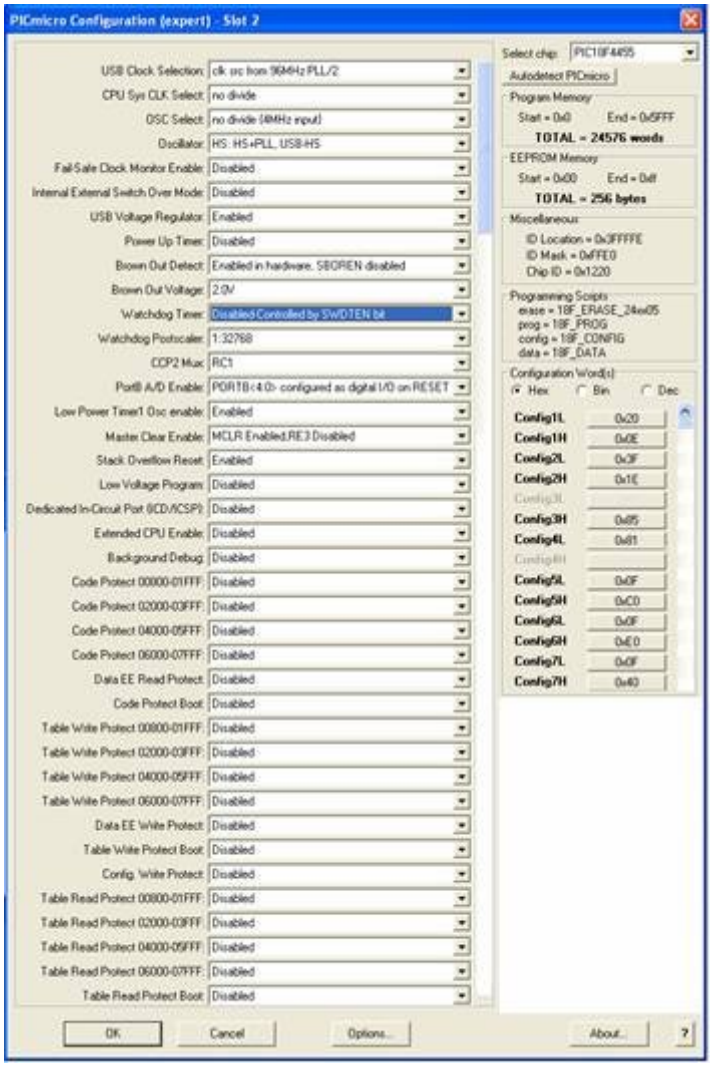
Wiring & Testing

EB896-80-1
E-Blocks USB Bundle

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

- Remove the 19.66 MHz crystal from the EB006 using a flathead screwdriver;
- Remove the 16F88 Device from the EB006;
- Insert the 4MHz Crystal into the EB006;
- Insert the 18F4455 into the EB006;
- Connect up the E-Blocks as shown on the right.
- Ensure the USB E-Block has the jumper setting A;
- Wire a link between the +V on the LCD and the +V on the EB006;

Before you can begin you must install the driver for the EB006 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 as shown on the left. The configuration is specified by clicking the Chip -> Configure menu ion Flowcode and then clicking the switch to expert config screen button.

Example file 1 is a good choice for a test file as it will connect to your computer as a HID device so you will not have to install a driver to allow this to work. You should check your computers device manager to confirm that the HID device has been added to the system. You may want to make sure that the num lock key is set to off before allowing the example 1 program to run.

USB Components

EB896-80-1
E-Blocks USB Bundle

The Flowcode USB components are as follows:

USB Serial – Emulates a communications port to replace legacy serial ports.
USB HID – Highly configurable human interface device.
USB Slave – Turns the microcontroller into a slave to the host.

USB Serial Component

This is the simplest of the USB components and you can basically treat it as you would a normal serial RS232 connection. The USB host will see the device as a communications port (once the driver has been installed) and allow programs like Hyperterminal and Labview to create a direct communications link to the device. The component allows you to read or write single bytes or full strings so the component can be used to create a true RS232 converter or replacement.

USB HID Component

This is similar to the serial component in that you can read and write data but unlike the serial component there is an additional timed event which is controlled by the computer that causes the data to be sent and received. Take for example a simple USB keyboard. A standard HID keyboard transfers 8 bytes of information to the USB host in regular intervals of say 10 milliseconds. All your program has to do therefore is to update the outgoing data buffer based on key press readings and check the incoming data buffer to see if any LEDs need to be lit or extinguished. The HID component also has another advantage in that instead of requiring a device driver like the other two USB components, you instead place a HID descriptor into the embedded device and this acts to enumerate the device and integrate it into the system for use.

USB Slave Component

The USB slave component forces the microcontroller to become a slave to the computer in that it must receive data from the USB host before trying to send data back. This is useful for creating question / answer style systems. Again Flowcode has come to the rescue by allowing a configurable slave service macro. This software macro is fired whenever data from the USB host is received allowing you to process the incoming data as soon as it arrives, make decisions and then respond with a timely appropriate answer.

Further help on the USB components can be found by selecting the component on the Flowcode Panel and then clicking the help button in the properties toolbar.

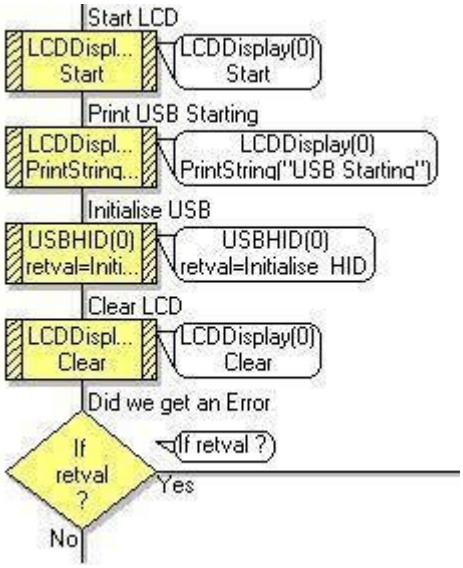
Example 1

EB896-80-1
E-Blocks USB Bundle

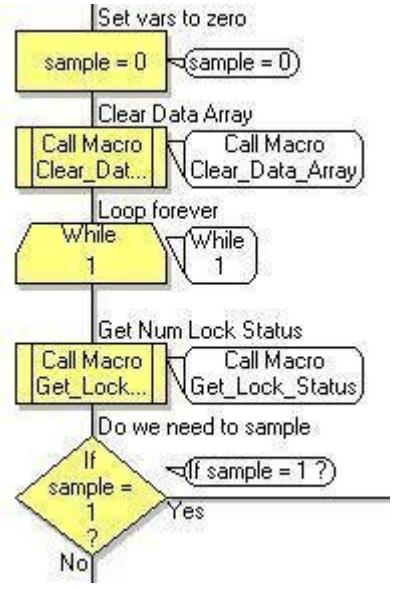
Example 1 is a simple data-logging program. The USB device connects to the computer as a HID keyboard so no driver file is required. It is recommended to ensure that the num lock key is not active when programming this example otherwise you will get a stream of text data being entered to the computer similar to if you held a key down on your keyboard.

Load Microsoft Excel and click into a cell. Then press the num lock key on your keyboard and the logging ability will being. Pressing the num lock key again will tell the USB device to stop logging data.

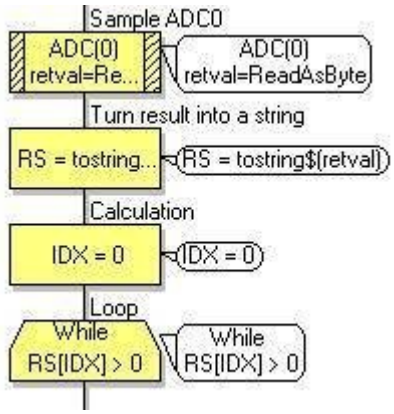
The program samples the analogue channel connected to pin RA0 of the microcontroller and returns the value as a byte value 0-255.



The program starts by initialising the USB component and waiting for the USB device to be recognised by the computer. The return value from the initialise function is used to determine if we have a successful connection or not.



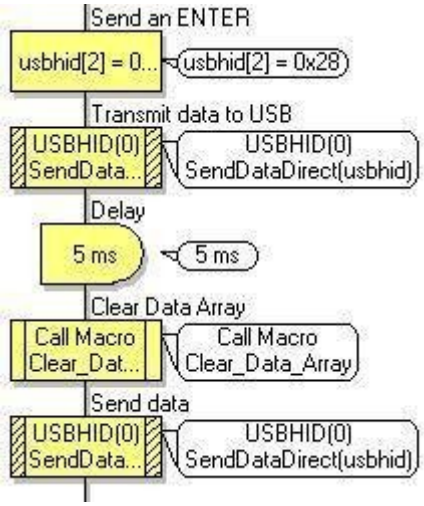
Before we enter the main program while loop we simply clear the sample control flag and clear the outgoing data buffer. The get lock status macro returns the status of the num lock modifier key which is then used to decide if we are actively sampling or not.



If we are sampling then sample the voltage connected to the ADC pin specified by the ADC component and then convert the numeric value into a string of characters. For example the number 255 would be converted into the characters '2', '5' and '5'. The program then enters a loop where each character in the string is converted into a key press and sent on to the computer. After each key press is sent the outgoing buffer is cleared and resent to represent a key release.

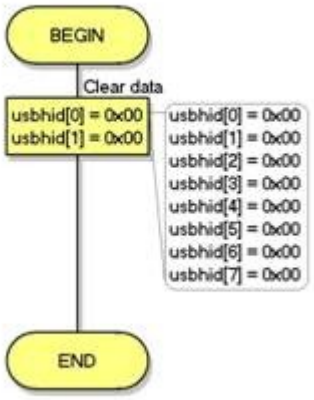
Example 1

EB896-80-1
E-Blocks USB Bundle



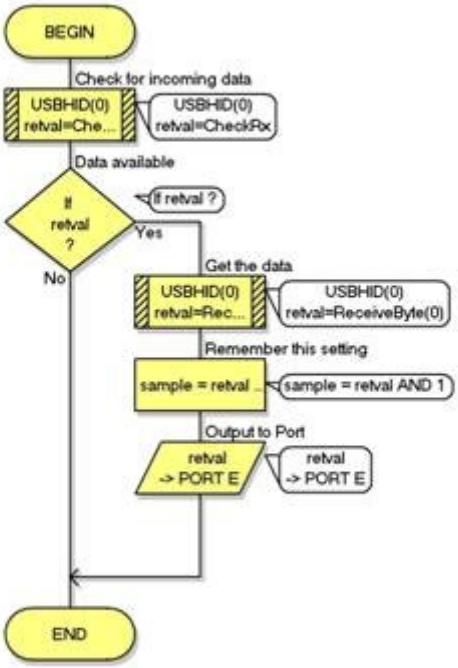
The end of the program loop assigns an enter to the outgoing data buffer to allow the numeric data to be saved into the cell in Excel and move onto the next cell.

Clear_Data_Array Macro



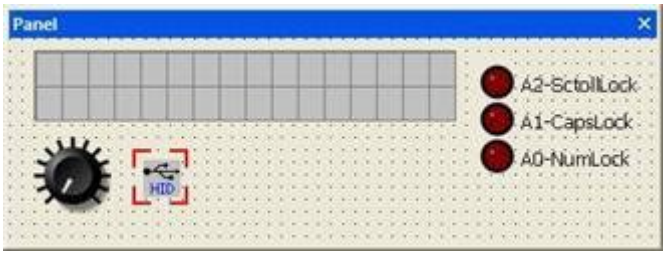
The clear data array software macro simply assigns 0 to all eight of the outgoing data bytes. This is the equivalent of no keys being pressed.

Get_Lock_Status Macro



The get lock status software macro checks for any incoming data and assigns it to the LED E-Block port. This allows you to see return values such as the num lock, caps lock and scroll lock. Inside the macro we simply check if the num lock bit is set and if it is then we set the sample flag which is used in the main program.

The panel for the program simply shows the LCD, the ADC, the USB HID and the LEDs used for displaying the keyboard lock status values.

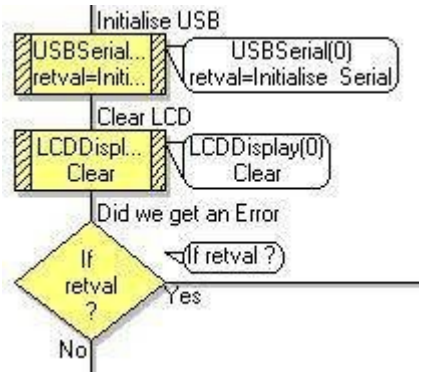


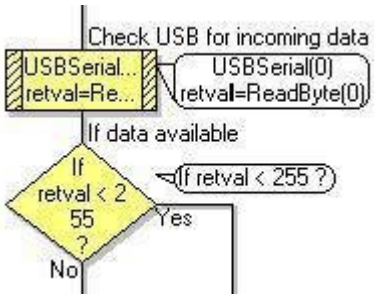
Example 2

EB896-80-1
E-Blocks USB Bundle

Example 2 is a basic terminal emulator that allows you to connect to the device using serial port software such as HyperTerminal or RealTerm. An example USB driver for the device is included along with the example files. The example allows you to enter data via the serial port software on your computer and this forwards the text to the LCD display.

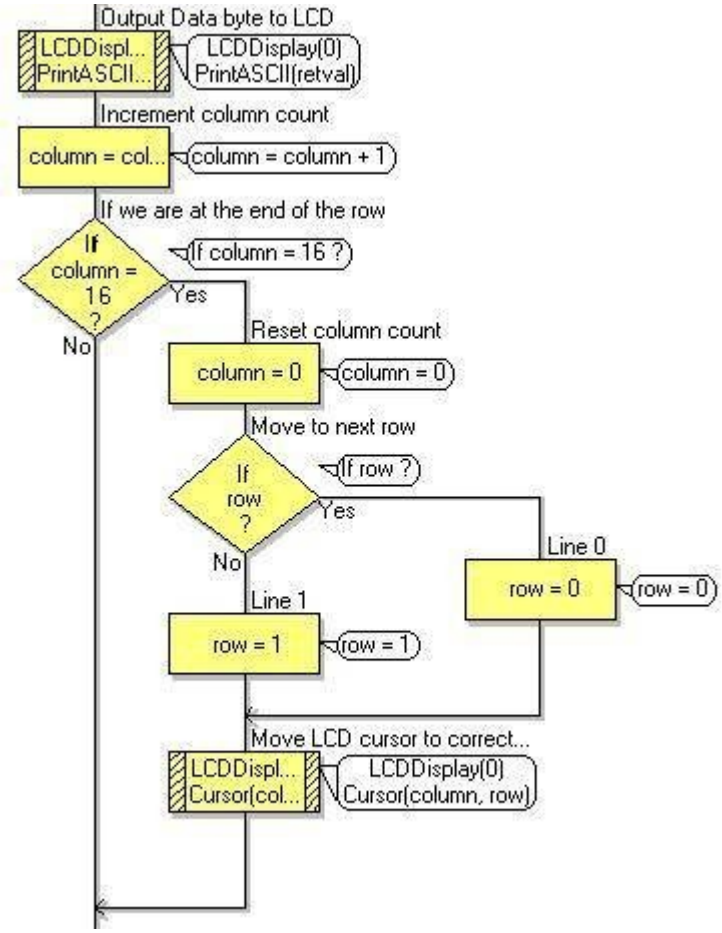
The program starts by initialising the USB component and waiting for the USB device to be recognised by the computer. The return value from the initialise function is used to determine if we have a successful connection or not.





The program checks for incoming data by calling the ReadByte hardware macro. If the return value is less then 255 then we have valid data, otherwise no data has been received.

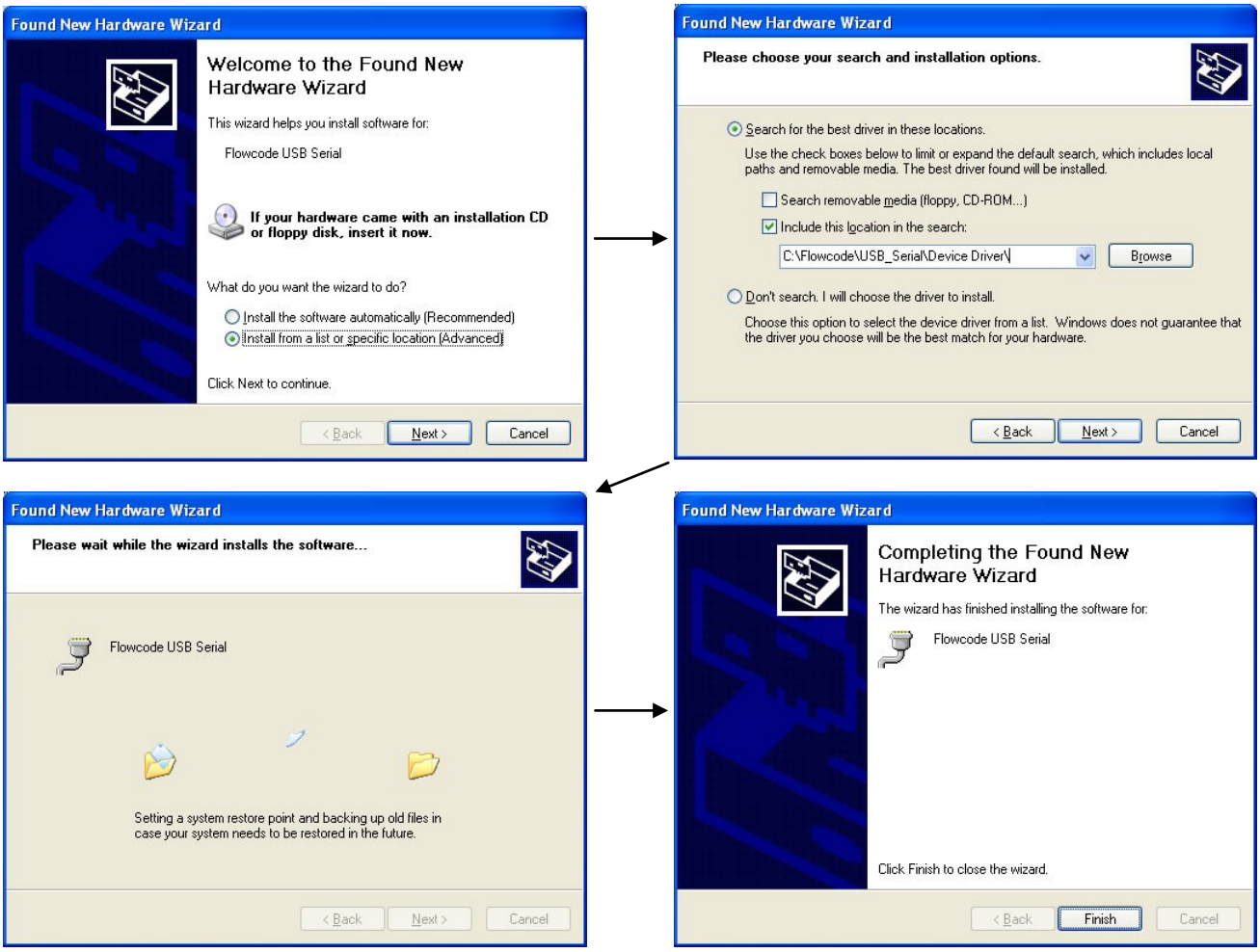
We print out the incoming character to the LCD keep track of the current cursor position to allow us to move to the next line or clear the screen if we have reached the end.



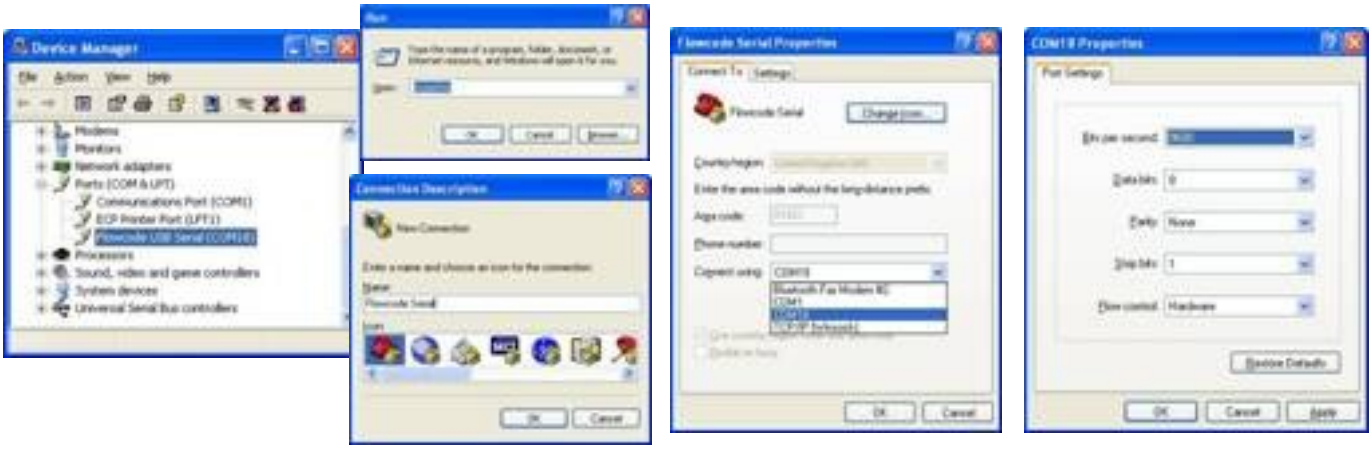
Example 2

EB896-80-1
E-Blocks USB Bundle

Shown below are the steps required for installing the USB Serial component driver. An example driver is provided as part of the example files. Alternatively you can use Flowcode to generate a new driver file.



Shown below are the steps required for getting HyperTerminal running with the USB serial component. The images show the devices port identification in device manager, Creating a new connection in HyperTerminal and assigning the correct port parameters.



Example 3

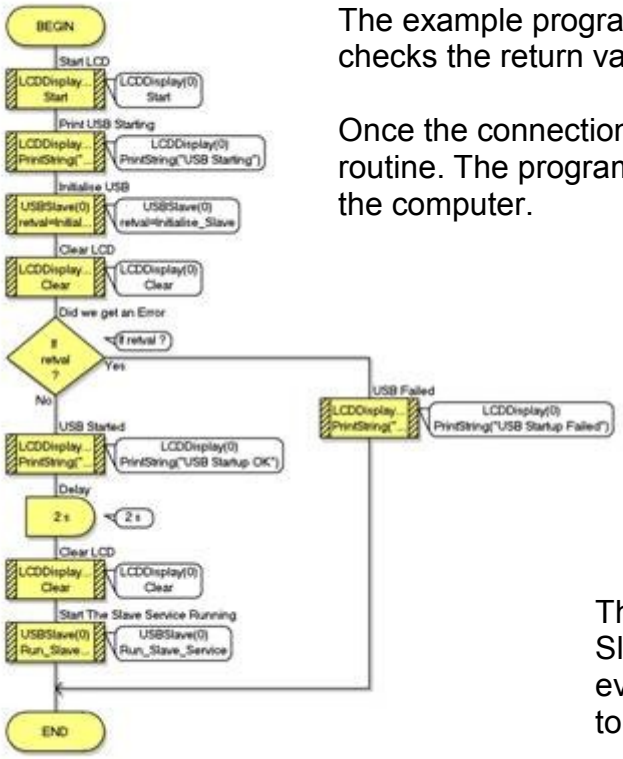
EB896-80-1

E-Blocks USB Bundle

Example 3 demonstrates how to use a computer programming language such as Microsoft Visual Basic to control a USB embedded device. Several Flowcode macros are made available to the computer by means of creating a command look up table. Every time the computer sends a packet to the slave USB device the slave service routine is triggered allowing the microcontroller to process the command and then return any data if applicable.

The Visual Basic example code can be opened and compiled using the freely downloadable version of Visual Studio 2008 Express Edition.

An example USB driver for the device is included along with the example files.

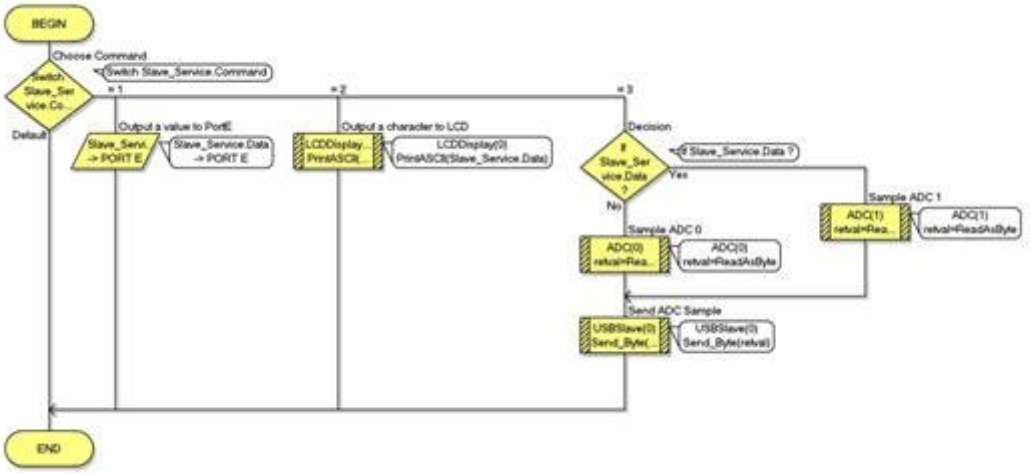


The example program does the usual USB component start up and checks the return value for a successful connection to the computer.

Once the connection is established we are starting the slave service routine. The program then pauses waiting for communication from the computer.

The Slave_Service macro is specified in the USB Slave component properties and is called whenever there is a communication from the computer to the device.

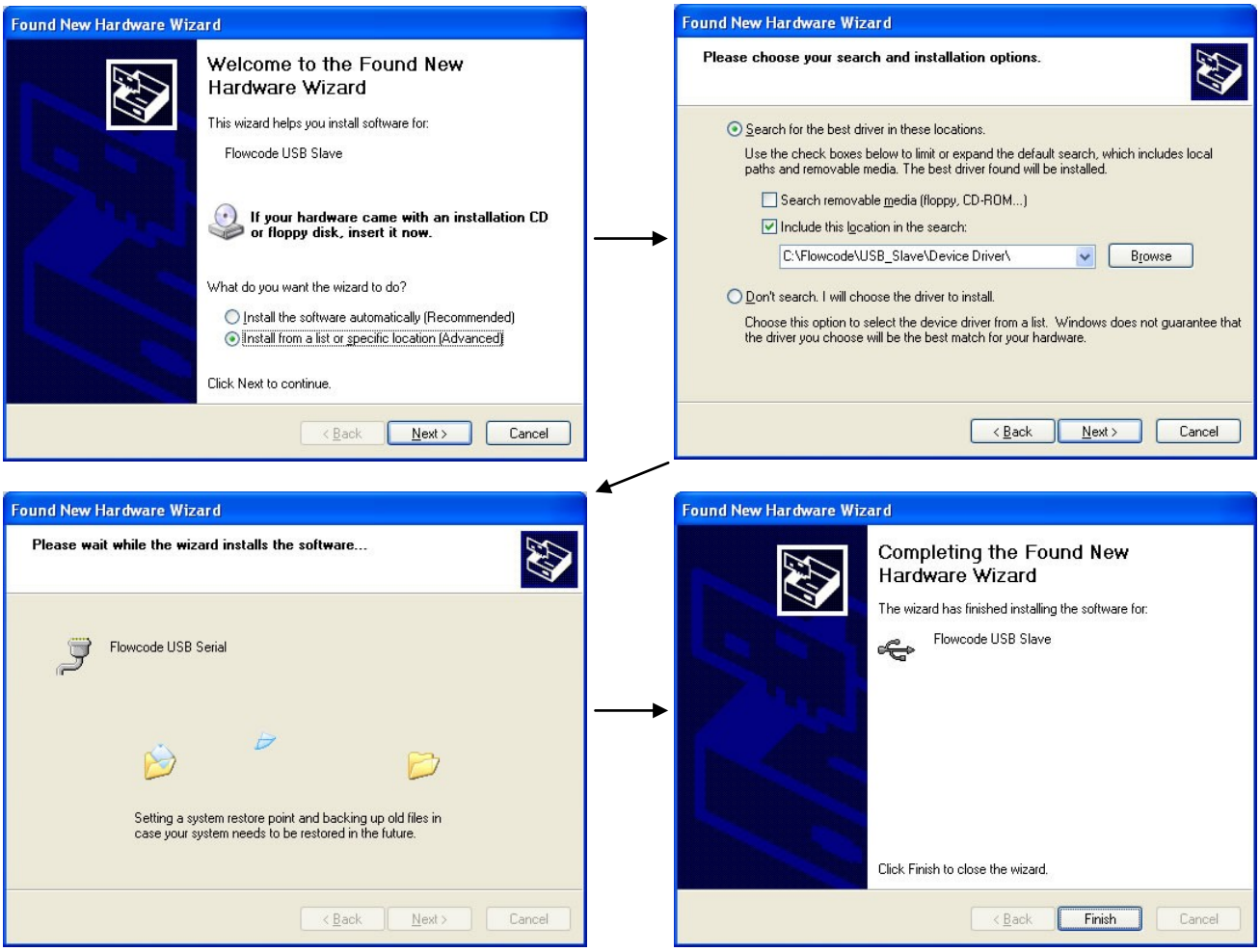
Slave_Service Macro



Example 3

EB896-80-1
E-Blocks USB Bundle

Shown below are the steps required for installing the USB Slave component driver. An example driver is provided as part of the example files. Alternatively you can use Flowcode to generate a new driver file.



The USB Slave component requires a DLL library to allow the computer to communicate. The library contains a number of functions to allow you to communicate with the device. The get count function returns the number of USB slave devices connected to the computer, the open and close functions open and close the connection to a specific USB slave device. Finally the transmit function is responsible for sending and receiving data from the USB slave device.

Here is an example VB call to the transmit function. Note that the parameters for this function are as follows: Outgoing Data Buffer, Max bytes to send, Number of bytes sent, Incoming data buffer, Max bytes to receive, Number of bytes received, Timeout in milliseconds.

```
'Perform send and receive transaction
ECIO_Transmit(Outgoing(0), MaxSend, SentCnt, Incoming(0), MaxReceive, ReceivedCnt, 10)
```

Troubleshooting

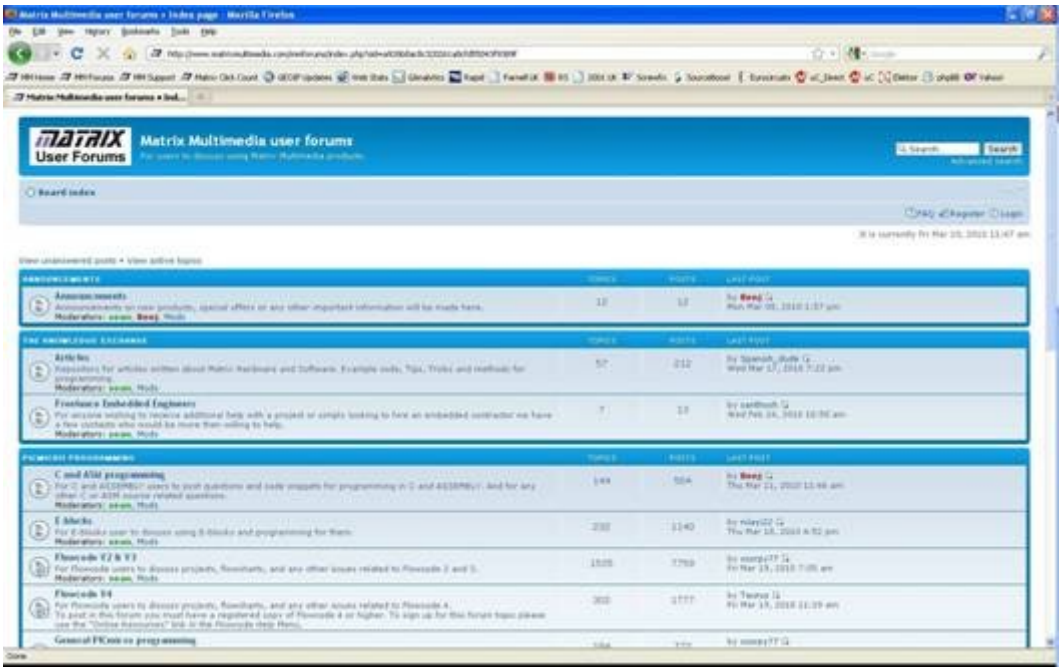
EB896-80-1
E-Blocks USB Bundle

If you are having problems getting up and running with any of the examples or any of the Flowcode USB components then the first port of call is to ensure you have your USB component properties configured correctly and also that your driver .inf file is up to date with the properties you have selected. It is also worth checking Windows device manager to see if the device is appearing correctly and if it has its driver installed correctly. Device manager can be opened by right clicking 'My Computer', selecting properties, clicking the 'Hardware' Tab and then selecting 'Device Manager'. In here you can view the details of your hardware and you can manually install and uninstall devices and drivers.

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

http://www.matrixtsl.com/mmforums/

The Articles section contains quite a few USB examples, Visual Basic source code as well as hints and tips to aid in your applications



There is also an online video demonstrating the Flowcode USB serial component available from the videos section of our website: http://www.matrixtsl.com/lc_videos.php

Other Products

EB896-80-1

E-Blocks USB 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