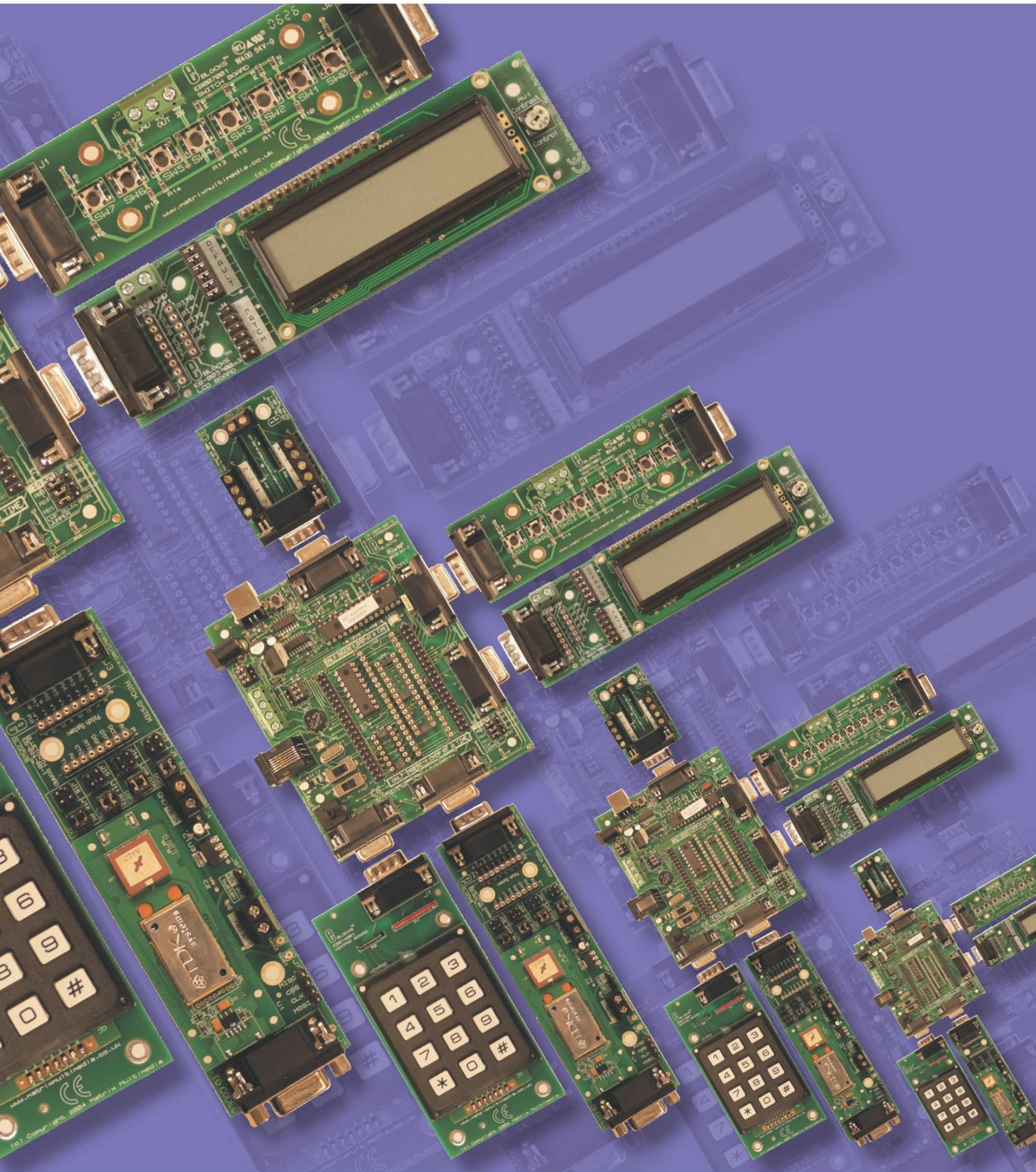
MATRIX

getting started guide

E-Blocks Easy Zigbee Bundle



Flowcode

EB998-80-1
E-Blocks Easy Zigbee Pack

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

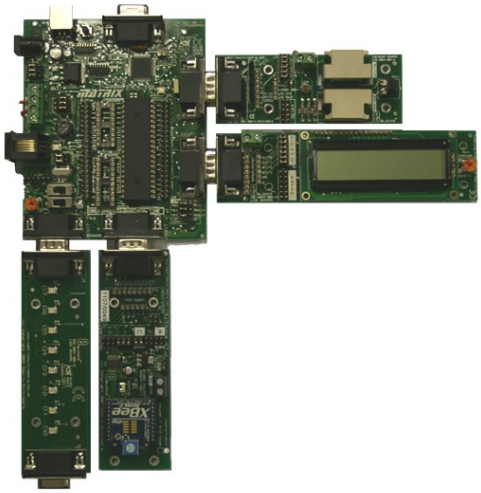
Support for frequently encountered problems can be found online on our FAQ's site.
Our online forums can also be used as a general discussion area or for help or advice.

Wiring & Testing

EB998-80-1

E-Blocks Easy Zigbee Pack

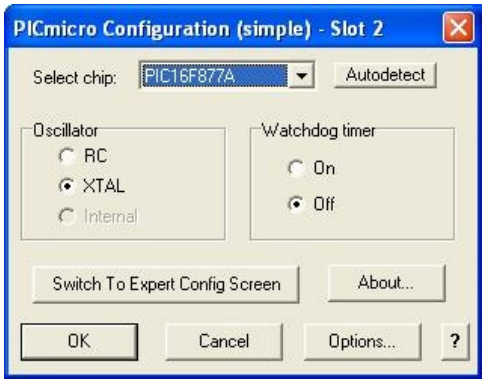
The Zigbee pack consists of three Zigbee E-Blocks. One is configured as a coordinator and the others are routers.

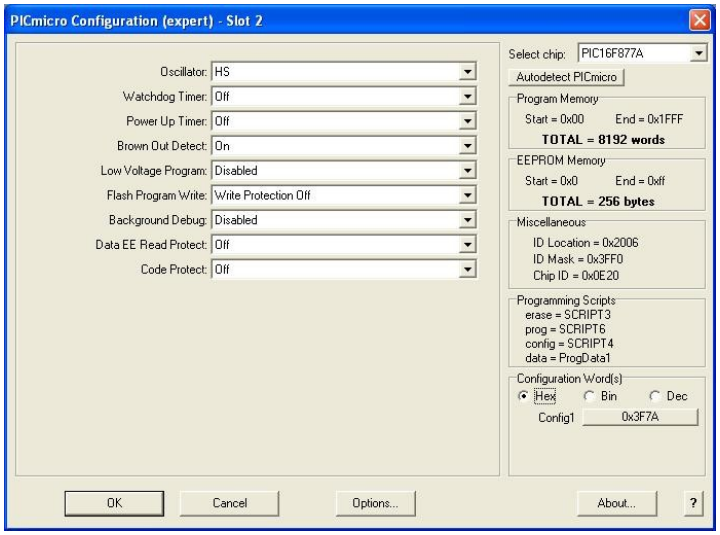


To setup your E-Blocks for use with the example programs you must perform the following actions:

- Remove the 16F88 Devices from the EB006's;
- Insert the 16F877A's into the EB006's;
- The coordinator is set up as shown on the right;
- The router boards have the Zigbee E-Block connected to Port C.
- One of the router boards has the sensors E-Block connected to Port A.
- The other router board has the switch E-Block connected to Port D.
- The zigbee, sensor, LCD and switches E-Blocks all need to be connected to the +V on their associated EB006 via single core wire.

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 icon Flowcode and then if you need more options click the switch to expert config screen button.

Example file 2 is a good test file as it scans for the nodes and displays the connected nodes on the LCD. This allows for a quick and easy check to ensure that all of the nodes are working as they should.

Flowcode

EB998-80-1

E-Blocks Easy Zigbee Pack

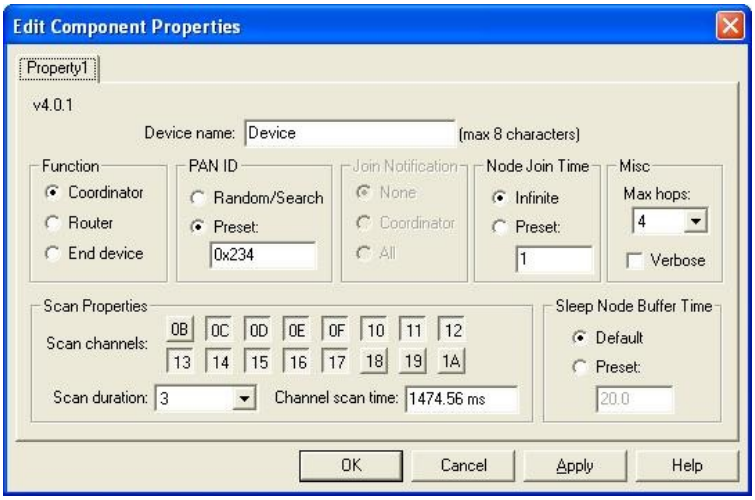
The Flowcode Zigbee component can be added to your program by finding the component in the wireless section of the Flowcode component toolbar.

Zigbee component icon.

Once the component has been added you will find that the component has been added to your panel.

Component Properties

The Zigbee component properties looks like the images shown on the right. Here there are several configurable options that define how the component behaves. When selecting the function of the Zigbee module make sure that the coordinator function is only given to the Zigbee E-Block that is marked as being a Coordinator. All other Zigbee E-Blocks should be configured as routers or end devices.



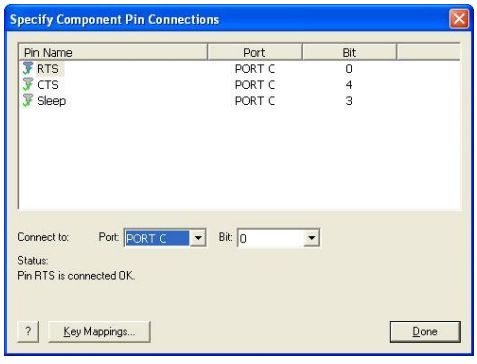
Macro Functions

Init / Restart / Scan — These functions control the entire network as a whole and allow you to initialise, reset or scan the network for devices.

Connect To... — These functions select which device we are talking to. We can either specify a ID, a device name, the system coordinator or the entire network.

Send / Receive — These functions are responsible for sending or receiving data directly to or from the network. These are used once you have used the above functions to configure the network and to connect to a specific node or set of nodes.

Component Connections



The connections for the Zigbee component should be set to Port C to correspond with the way the E-Blocks are connected. When using other microcontrollers you should always connect the Zigbee E-Block to the port which contains the UART hardware.

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

Example 1

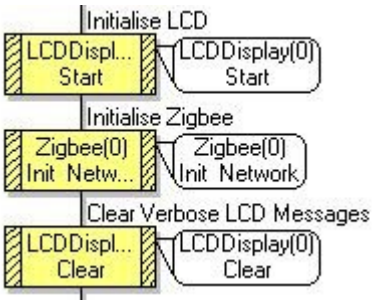
EB998-80-1

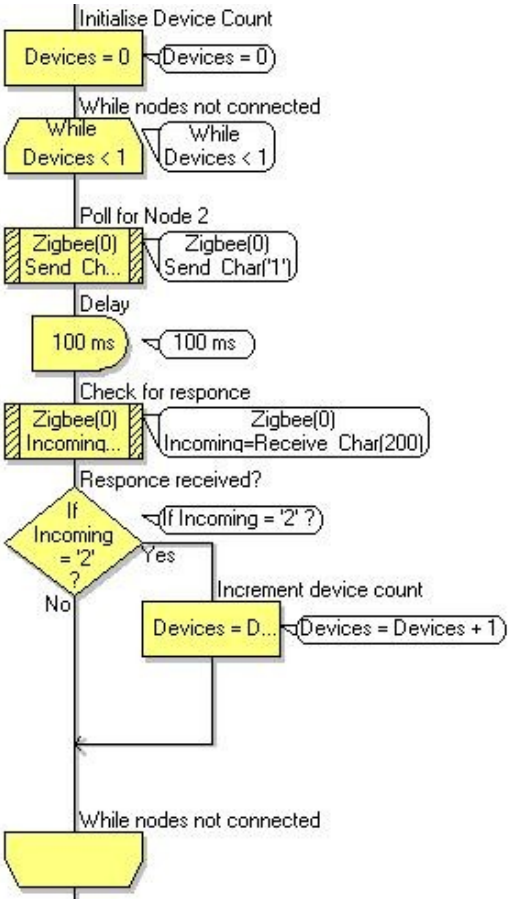
E-Blocks Easy Zigbee Pack

Example 1 is a simple 2 node system. The Coordinator node starts up the network and then checks to see if node 2 is available. Once node 2 has been detected any incoming data from node 2 is displayed on the LCD display. The data on node 2 is specified by pressing a switch on the switch E-Block connected to Port B.

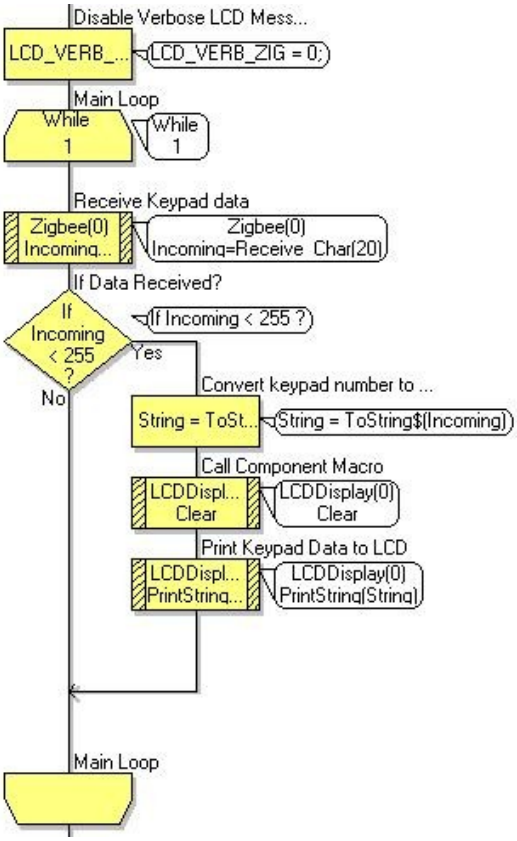
Node 1 Coordinator Program

Node 1 of this example should be programmed into the Coordinator node. The program starts by initialising the LCD and then calls the Zigbee initialise function. If the Verbose mode is enabled in the Zigbee properties then you will get a print out of the AT commands as they are fired into the Zigbee module. When the initialise is finished we clear the display to remove and verbose messages before continuing.





This portion of the code simply outputs a 1 to any connected devices on the zigbee network. If we get a return value of 2 from over the network then we know that node 2 is connected and is running correctly. Therefore we can drop out the detection loop and continue with the program.

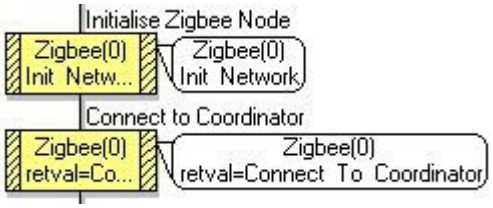


The C code icon simply overrides the verbose mode allowing us to switch it on and off mid program. We then enter the main program loop which scans for any incoming data and then outputs the data to the LCD.

Example 1

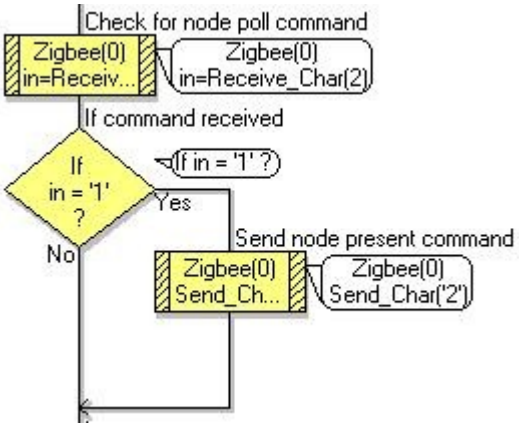
EB998-80-1
E-Blocks Easy Zigbee Pack

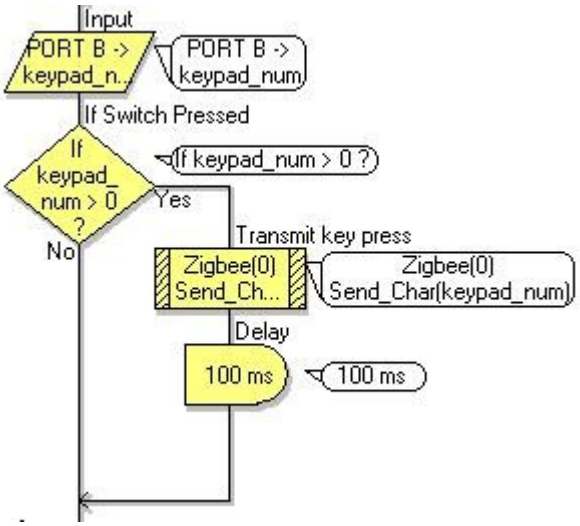
Node 2 End Device Program



For the end device we are first initialising the Zigbee device, this lets the device join the network on the PAN ID specified in the component properties. We then use the connect to coordinator macro to route the outgoing data to the coordinator module.

Once we have the end device connected to the network we check to see if there is any incoming data. If we receive a 1 then this is the node request so we know to reply with a 2.





Inside the main loop we also check for a switch press by reading the port connected to the Switch E-Block. If any of the switches are pressed then this will result in a return value greater then 0. If this is the case then we simply send that return value and that then gets automatically forwarded to the end point which we have already defined as the Coordinator node.

Please note that the verbose mode should only be used with a program that has a standard LCD or graphical LCD component. Otherwise you will get errors when trying to compile to the device.

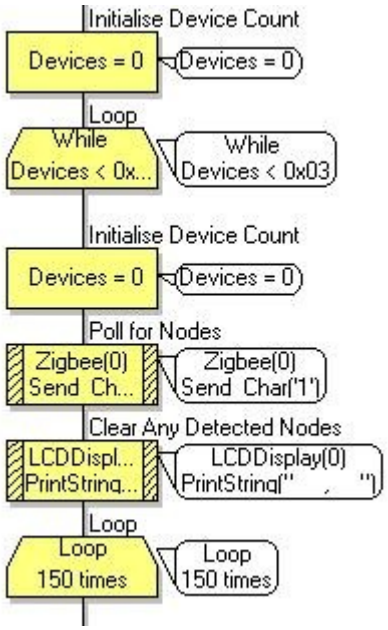
Example 2

EB998-80-1

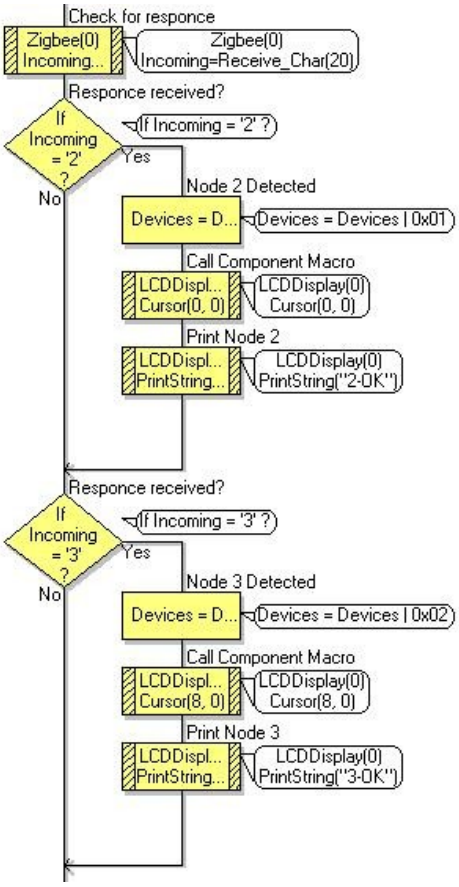
E-Blocks Easy Zigbee Pack

In example 2 we will connect all 3 of the multiprogrammer boards to the Zigbee network. Firstly move the LED E-Block from the coordinator node to PortB of Node 3. The Coordinator node will start up the network and then check that all nodes are present. Once all of the nodes have been successfully discovered the program then sends out a 0 which triggers the end nodes to start their main functionality. That is to send the switch data from node 2 to the LEDs connected to node 3.

Node 1 Coordinator Program

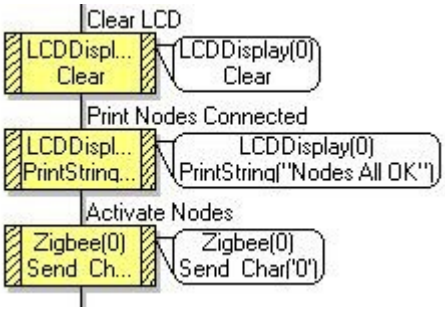


Here the coordinator now waits for the two other nodes to connect to the network and respond at the same time. After sending out the detection byte the program allows 150 loop cycles to detect the received acknowledge bytes before starting the scan process over again.



When listening for the receive acknowledge bytes we simply check for an incoming byte and then see if it is equal to 2 or 3 to correspond to the node that has responded. When both nodes 2 and 3 have responded the Devices variable has both of its lower bits set which allows us to drop out of the scan for nodes loop.

Once all of the nodes have been detected by the Coordinator node the command 0 is sent out to start all of the end nodes in their main functionality. This allows for a clear distinction between establishing all the nodes on the network and then performing a task.

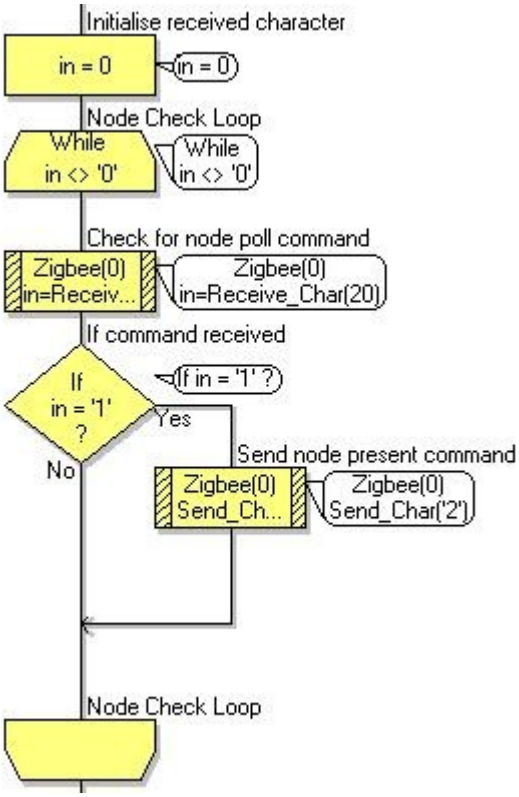


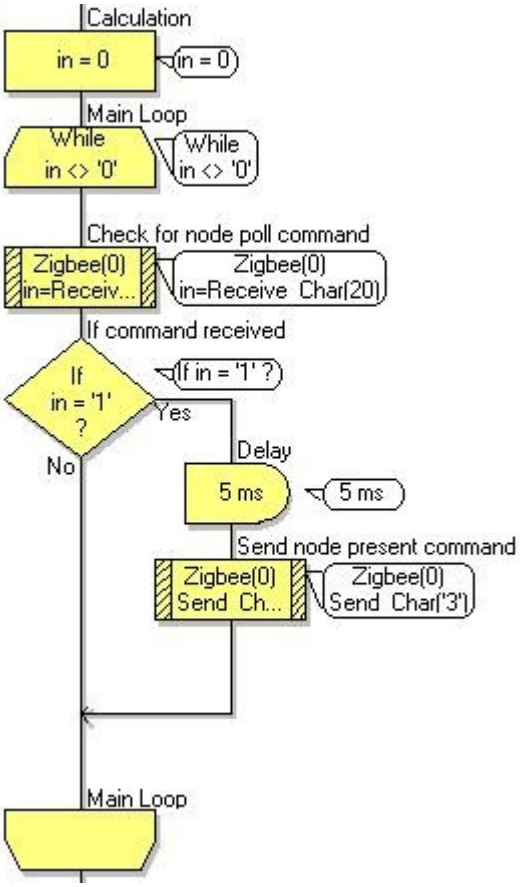
Example 2

EB998-80-1
E-Blocks Easy Zigbee Pack

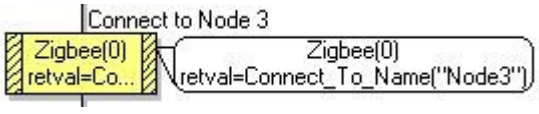
Nodes 2 and 3 End Device Programs

As nodes 2 and 3 are both waiting for command 1 to be received when they start up we have added a small delay to the decision in node 3. This allows both nodes 2 and 3 to respond to the node detection message without any problems incurred with two devices both trying to transmit at exactly the same time. As zigbee has a fairly low data throughput it is wise to try and avoid such conflicts if possible to ensure your control data makes it to the destination in the fastest possible time.





When the value of 0 is received the end nodes both drop out of their detection routines and instead go to their main function. This is to simply forward the value from the switches connected to node 2 to the LEDs connected to node 3.



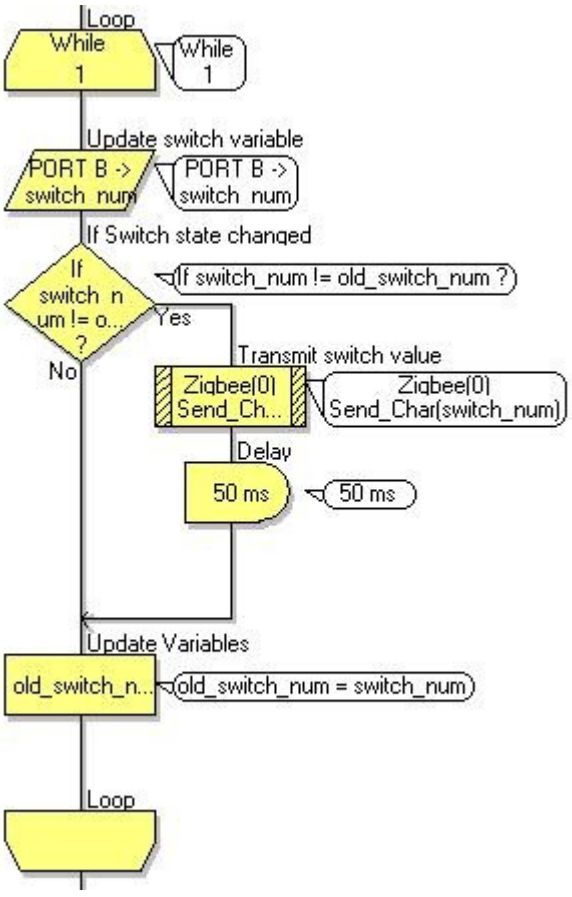
The node 2 program addresses node 3 as the data endpoint by using the Connect_To_Name macro which uses the name specified in the Node 3 Zigbee component properties page.

Example 3

EB998-80-1
E-Blocks Easy Zigbee Pack

In example 3 we will repeat the functionality shown in example 2 but now we will allow end node 3 to go to sleep to save power. This functionality is useful for devices that are battery powered or require high power efficiency. When a node is asleep its parent node stores the data for it. The Zigbee protocol states that either the coordinator or router devices are capable of storing data for end points which are asleep. Once the endpoint needs to wakes up it uses the node wake macro to wake up the Zigbee module and then the data is automatically streamed to the end device from the parent node. This is therefore another reason why Zigbee is not useful for high data throughput but very useful for control applications.

Nodes 2 End Device Program



Shown on the left is the routine used for node 2 in examples and 3. If the switch value that is read from the port differs from the value that is stored in memory from the last sample then we know that at least one of the switches has been pressed or released. Note that we do not have to make any distinction between sending data normally and sending data to a device that is asleep.

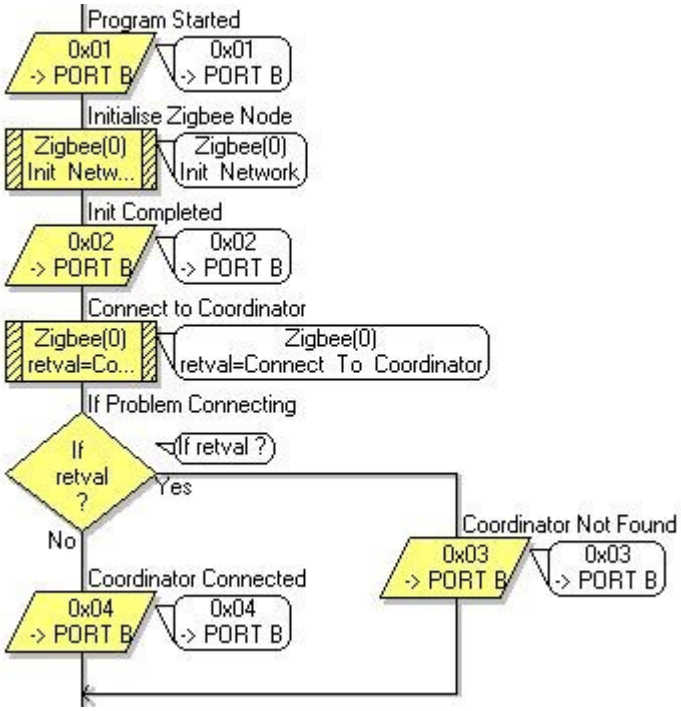
The coordinator program for node 1 has the standard start up and node check routines followed by the 0 command to trigger the nodes into action. This is just a protocol we have adopted here at Matrix. In your own programs you can in fact create your own protocol to do any functionality you choose.

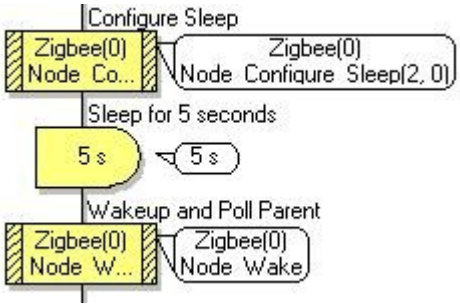
Example 3

EB998-80-1
E-Blocks Easy Zigbee Pack

Nodes 3 End Device Program

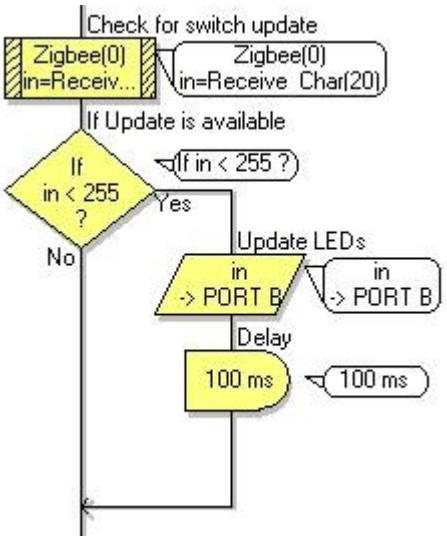
In our node 3 program we have used the LED E-Block to show the status of the Zigbee component initialisation. If we get a output of 3 on the LEDs then we know that the start up routine has failed. This kind of troubleshooting is very helpful when it comes to finding out why a program is not running as expected.





The Configure_Sleep macro is used to put the Zigbee device into a powered down state. This allows the device to save power but also means that the device is not reachable over the network.

Calling the Node_Wake macro returns the Zigbee device to its normal operation and if we are already joined to a network then the node will automatically rejoin. If any data was sent to the node while it was asleep then it will be buffered by the parent node. This could be the coordinator or the other end node. Data is read into the device as normal so we simply call the Receive_Char macro to retrieve any stored data.



Troubleshooting

EB998-80-1

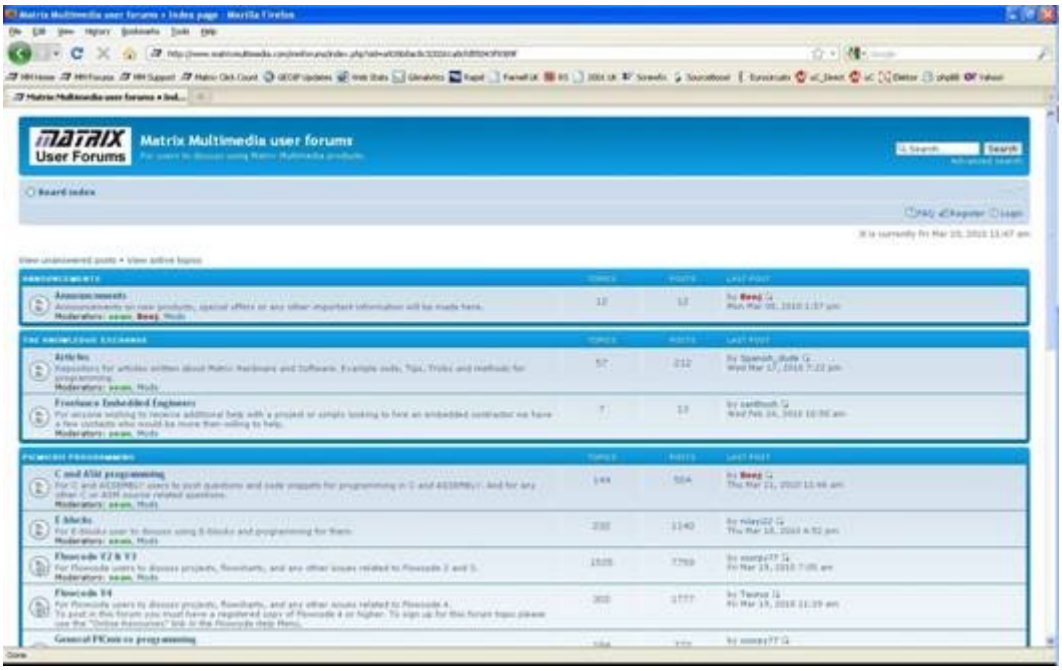
E-Blocks Easy Zigbee Pack

If you are having 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 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.matrixtsl.com/mmforums/

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



There is also a set of online videos demonstrating Flowcode, available from the videos section of our website: http://www.matrixtsl.com/lc_videos.php

Other Products

EB998-80-1

E-Blocks Easy Zigbee Pack

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