



## Android, Bluetooth and MIAC

by Ben Rowland, June 2012



### Abstract

Discover how easy it is to use TCP network communications to link together high level systems. This article demonstrates techniques to pass data between embedded devices such as Microcontrollers, Computers, Raspberry Pi and Android.

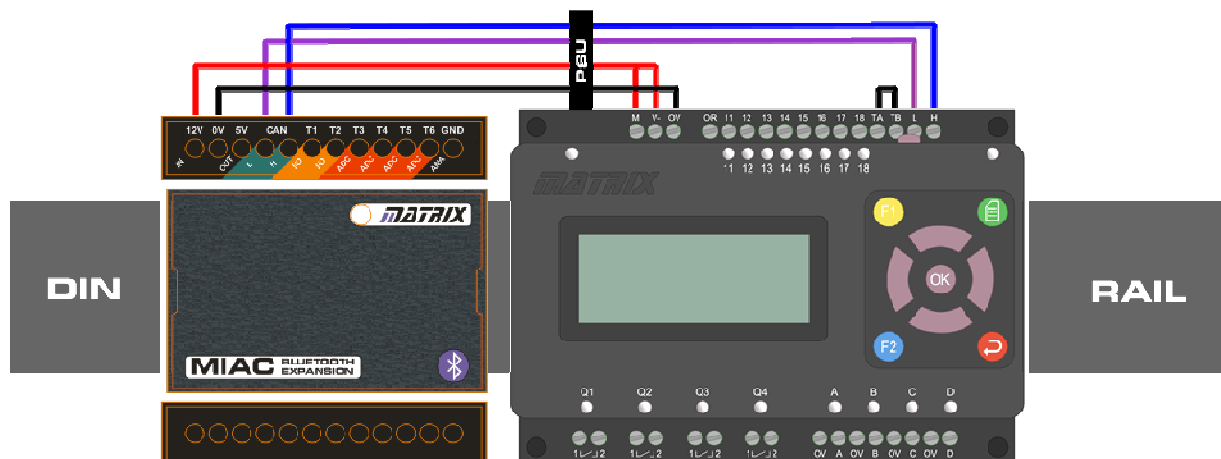
### Requirements

#### Software:

- Professional or Educational version of Flowcode V4 / V5
- Python 2.7
- Android JAVA SDK with Eclipse

#### Hardware:

- E-blocks setup with EB023 Ethernet board
- Raspberry Pi with EB080 E-blocks adapter board
- Android Phone or Tablet
- Computer



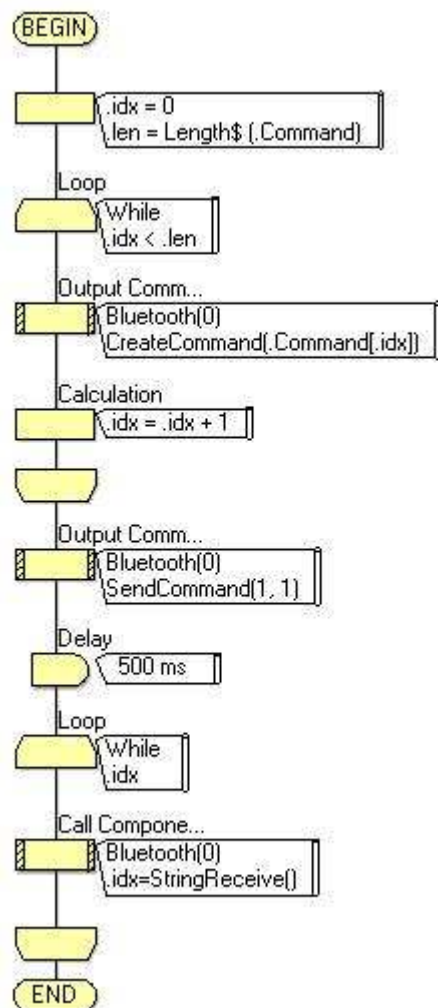
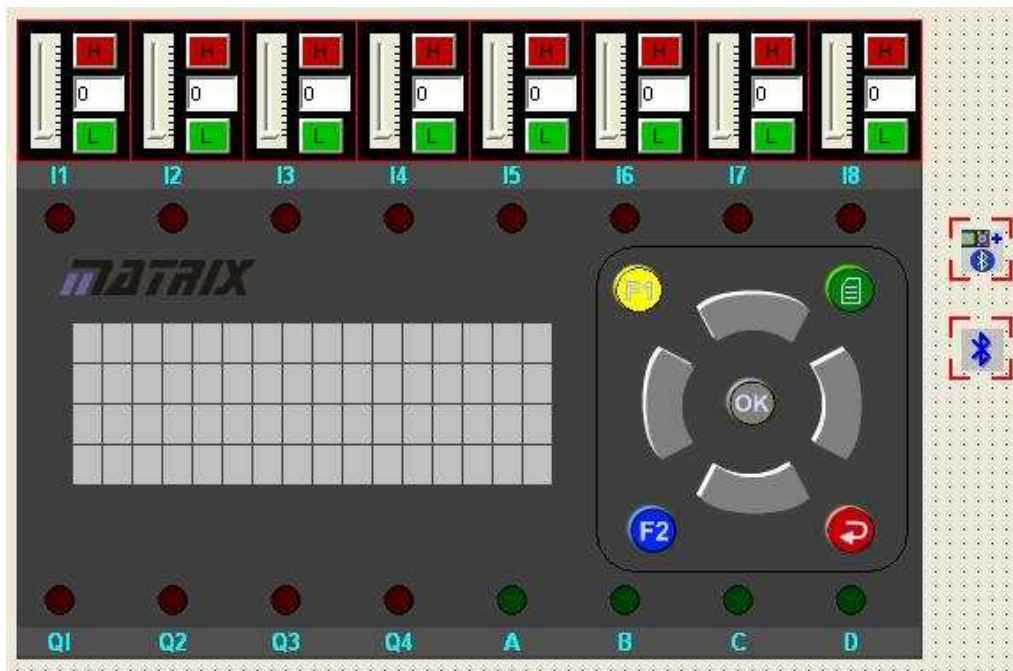
### The Flowcode Program

This demonstration application highlights a way to externally control the MIAC operation using a remote device such as an Android phone or tablet.

The setup consists of a MIAC System, comprising a Master MIAC and a Bluetooth Expansion Module.

The Bluetooth module is then used as a data link between the MIAC and the controlling device.

The Master MIAC is also used to setup the Bluetooth module ready for data pairing and then sit and wait for commands coming in via the Bluetooth. When data is received from the Bluetooth module the Master MIAC processes the incoming data and allows things like the relays to be switched on or off.

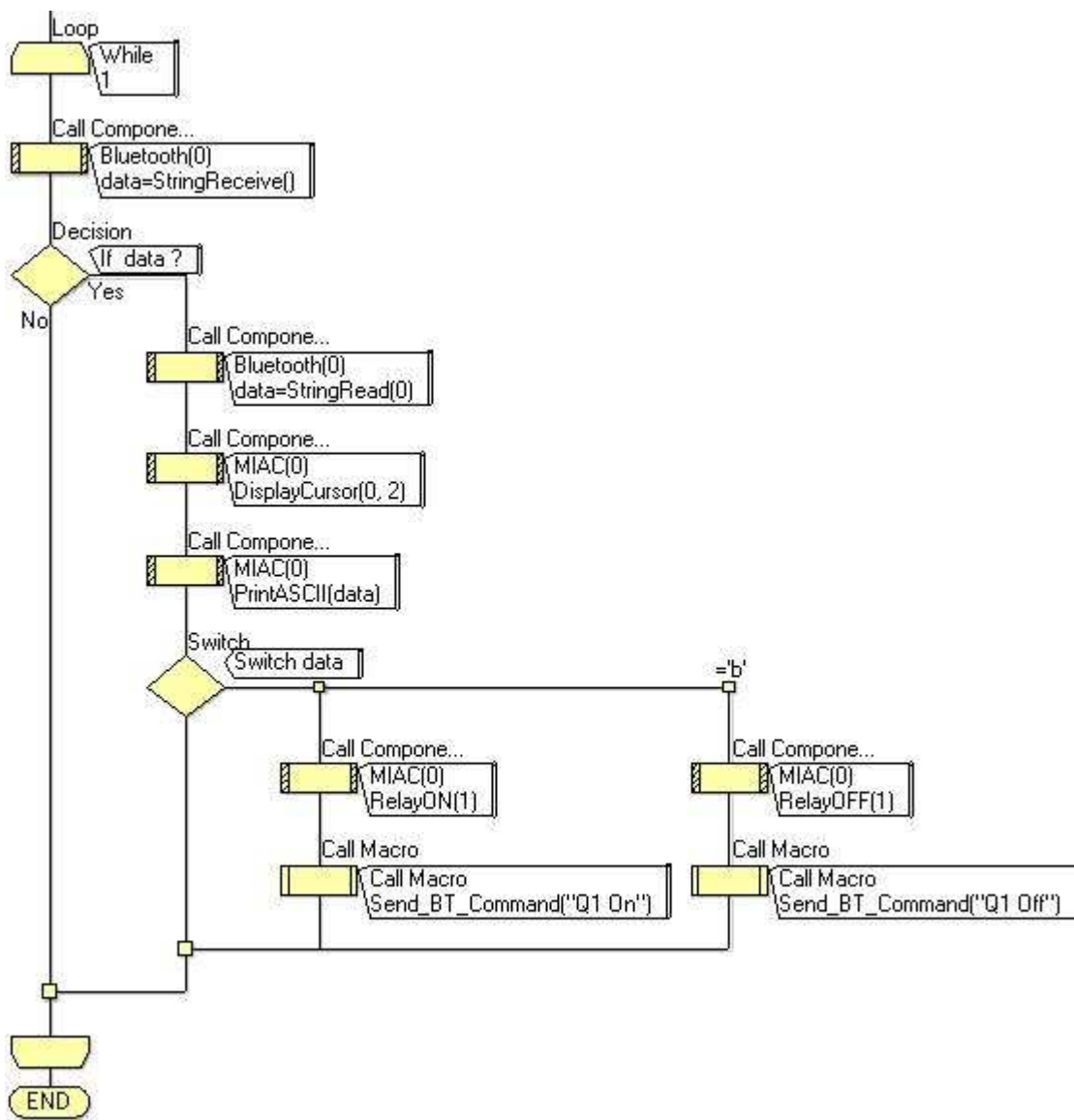


The main part of the program begins by setting up the initialization of the Bluetooth module and other components such as the MIAC LCD. A series of AT commands are then sent to the Bluetooth module which do a number of things. The send\_BT\_Command macro is then used to transmit the command string to the Bluetooth module, collect the automatic echo and then collect the module's response.

AT&F*	Factory Reset All Settings
ATS512=4 -	Not Sure
ATS0=-1	Not Sure
AT+BTK="1234"	Setup Pair Key
AT&w	Write To Non-Volatile Memory
ATZ	Module Software Reset

The main program then loops waiting for a data byte to be received from the Bluetooth module. If the ASCII character 'a' is received then relay Q1 will be switched on whereas if character 'b' is received then relay Q1 will be switched off.

The MIAC will also reply to the Android device to confirm that the state of relay Q1 has changed. This can then be adapted to create complex routines on the MIAC from a single data command byte or can be expanded to control other MIAC Add-on modules such as RS232 or PWM motor control.



## The Android App

To begin developing on a system such as Android first you need to download and install the SDK or software developers kit. For Android there are guides available on what to download and then how to plug it all together within the Eclipse environment.

Official Getting Started Tutorial - <http://developer.android.com/sdk/installing.html>

Once you have everything installed you should have all of the following tools required to develop android applications.

- Eclipse IDE (Integrated Development Environment)
- Eclipse ADT Plugin (Android Development Tools)
- Android SDK
- Android ADB (Android Debug Bridge)
- Android Emulator (PC / MAC / Linux Virtual Android Device)

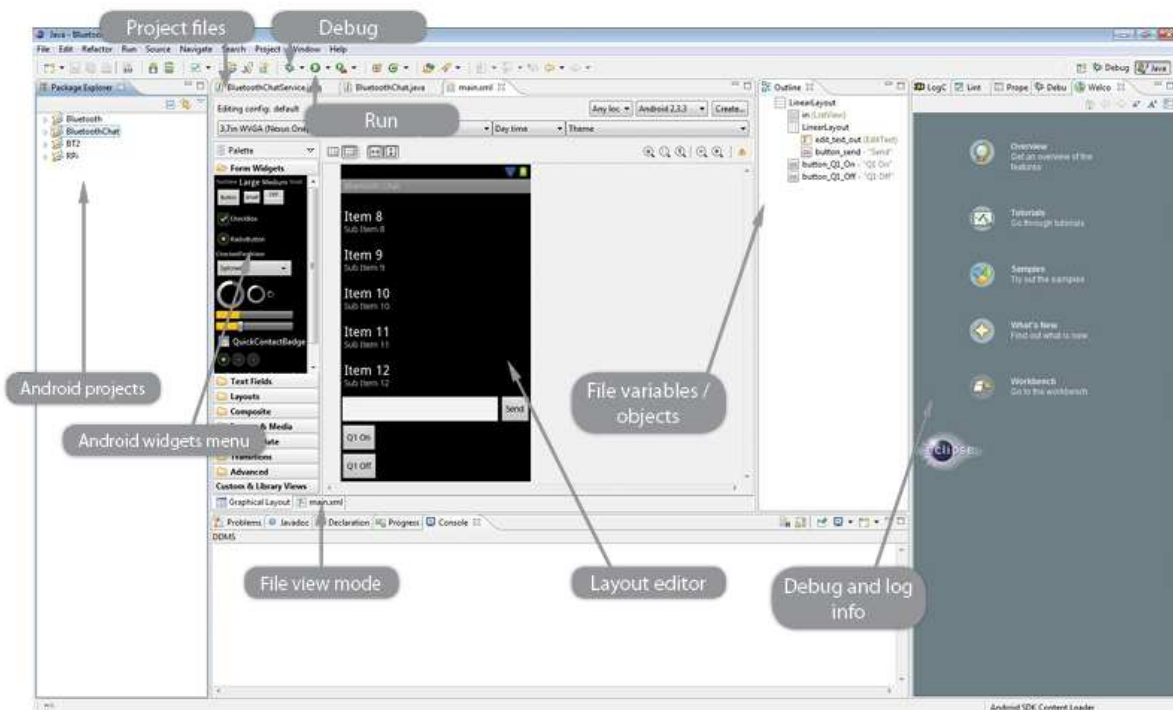
The Eclipse environment is used to edit the various code files in the android project. The ADT plugin is used like a wrapper to convert Eclipse into an Android development environment with all the android related functionality. The SDK is the compiler version responsible for deciding which version of Android you are compiling for and are compatible with. The ADB is used to debug and test your applications directly on your android device and finally the Emulator allows you to debug and test via a virtual Android. Note that the virtual Android does not currently support Bluetooth functionality but things like TCP style communications are supported.

It is also recommended to go through some of the demo applications provided such as the Hello World demo to allow you to get used to the IDE and various Android tools. There are generally only five basic files you need to be aware of when working on a basic android project.

- The Java source file – Located in the “src” folder and used to control the app’s activity
- The main.xml file – Located in the “res” folder and used to control the app’s layout
- The strings.xml file – located in the “res” folder and used to store all the dynamic string data
- The AndroidManifest.xml file – Located in the root folder and used to control app permissions such as Bluetooth/Wifi access.

The Android App file – Located in the “bin” folder and used to install the app onto a Android device. In the example Android App we have created two buttons to allow the ASCII characters ‘a’ and ‘b’ to be sent to the MIAC system to control relay Q1. There is also a third button and a text input field which allows for other messages to be sent to the MIAC system for development purposes. Any data send between the MIAC system and the Android app is shown in a non-editable text field.

**Fig4 Android IDE and Layout file Main.xml**



## Further reading

Below are some links to other resources and articles on related subjects, and technical documentation relating to the hardware used for this project...

Flowcode:	<a href="http://www.matrixmultimedia.com/flowcode.php">http://www.matrixmultimedia.com/flowcode.php</a>
MIAC :	<a href="http://www.matrixmultimedia.com/miac.php">http://www.matrixmultimedia.com/miac.php</a>
Android:	<a href="http://developer.android.com/sdk/installing.html">http://developer.android.com/sdk/installing.html</a>
Learning Centre:	<a href="http://www.matrixmultimedia.com/lc_index.php">http://www.matrixmultimedia.com/lc_index.php</a>
User Forums:	<a href="http://www.matrixmultimedia.com/mmforums">http://www.matrixmultimedia.com/mmforums</a>
Product Support:	<a href="http://www.matrixmultimedia.com/sup_menu.php">http://www.matrixmultimedia.com/sup_menu.php</a>

Copyright © Matrix Multimedia Limited 2011

Flowcode, E-blocks, ECIO, MIAC and Locktronics are trademarks of Matrix Multimedia Limited.  
PIC and PICmicro are registered trademarks of Arizona Microchip Inc.  
AVR, ATmega and ATtiny are registered trademarks of the ATMEL corporation.  
ARM is a registered trademark of ARM Ltd.