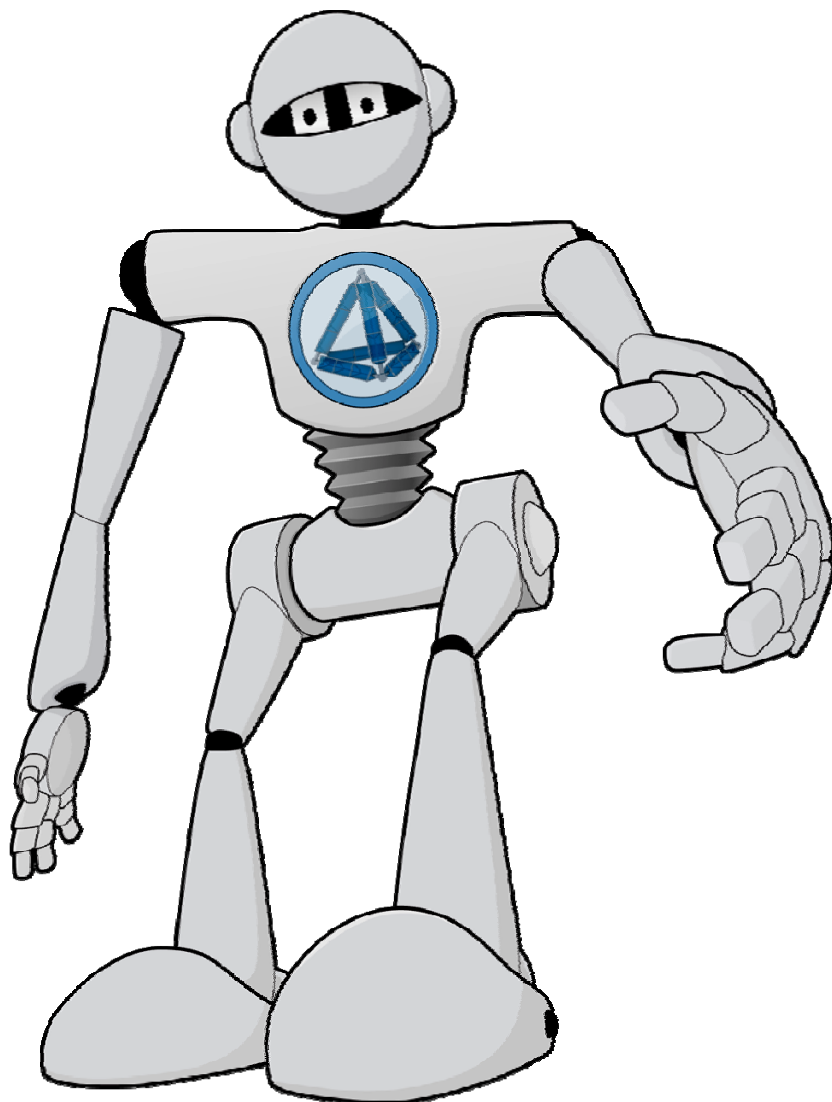


# Tribotix

## Adding Bluetooth to the Bioid's CM5





## 1. Disclaimer

This document/procedure has been documented to allow people wanting to add Bluetooth functionality to their Robotis CM-5 control board.

If the reader attempts these modifications and either the modification fails or the CM-5 control board is damaged then the author accepts no responsibility for replacing or repairing the faulty/damaged CM-5 control board.

This procedure/modification should only be attempted by users with relevant electronic experience, including (but not limited to) soldering, cutting pcb tracks and adding additional components.

The modifications documented here are not difficult for a trained electronic technician, please be aware though that the users performs these modifications at their own risk.

Also, all the information contained within this document/procedure was obtained manually – i.e. no information was provided by Robotis to achieve this modification. All schematics were generated by visually tracing the tracks on the pcb then verification via a multimeter. As of the date that this document/procedure was written Robotis still haven't made the schematic for the CM-5 available to distributors or the general public.



## 2. Background

The original motivation behind this modification was seeing that an 8-pin DIP package labelled 'ZigBee' on the CM-5 control module. This DIP package looked very similar to the DIP package on Robotis's CM-2 that was used to add Bluetooth functionality to the Cycloid range of humanoids. After initial investigations it was obvious that the I/O allocation on the 'ZigBee' module was pretty close to that of the Sena ESD200 [1,2], shown below in Figure 2.1

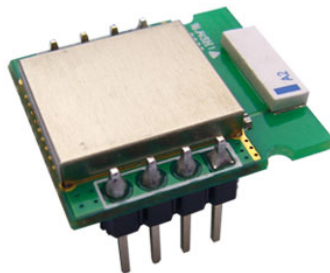


Figure 2.1 Sena Parani-ESD200 OEM Bluetooth-Serial Module-Class 2

The initial problems faced with adding this to the CM-5 where:

1. **5V & 3.3V TTL logic**, The CM-5 uses 5V logic but the ESD200 uses 3.3V. When I started with this modification I was having trouble getting this working and I thought this was an issue but it appears that the 5V & 3.3V thresholds are the pretty close and it works without a logic level translator. Just to be on the safe side, I originally had a logic level translator in my circuit and when it started working I successfully removed this.
2. **Reset**, the reset pin at the 'ZigBee' 8 pin pcb location is connected to the mcu reset, which is in turn driven by the MODE pushbutton (pushing the MODE pushbutton on the CM-5 causes the mcu to undergo a hardware reset). This means if you change modes then this will reset the ESD200, i.e. it disconnects the Bluetooth link, which is not very practical. So I used a simple RC to hold the Reset pin high, therefore it only resets when the power is cycled to the CM-5.
3. The **serial communications on the CM5 is a little tricky**, there are separate enable pins for the serial when it goes to the 'ZigBee' or wired serial port. This means you can have wireless communications between CM5's (Zig100 enable) or programming via serial lead (wired serial enabled). I traced these circuits out and did some modifications and I was able to get it so the CM-5 could be programmed via the



Bluetooth as well as talk to other CM5's BUT not at the same time.  
What I mean by this is the ESD200 is programmed to either:

1. connect to the same device as last time (I use this for CM5-CM5 communications) or,
2. a device capable of detecting and connecting (I use a D-Link DBT122 [3] dongle).

As there are 2 different ways to use, the ESD must be reprogrammed when you change usage. This is a problem considering you must solder the ESD200 into the CM-5 board. For my testing and debugging I used 2 x 4pin SIP headers, but with the headers soldered into place you can't put the plastic back over the CM-5.

So, you really have to choose how you want to use the Bluetooth on the CM-5 and then you'll be stuck with this (unless you are good with a desoldering tool). Because of this I have decided to have a separate complete section for both options – pick the one that best suits your needs.



### 3. CM5 Communications

The complete schematic of the serial communications circuits of the CM5 can be accessed with the associated zip file, *CM5 Comms.pdf*.

#### 3.1 AX TTL Serial Communications

*(ATMega128 UART0)*

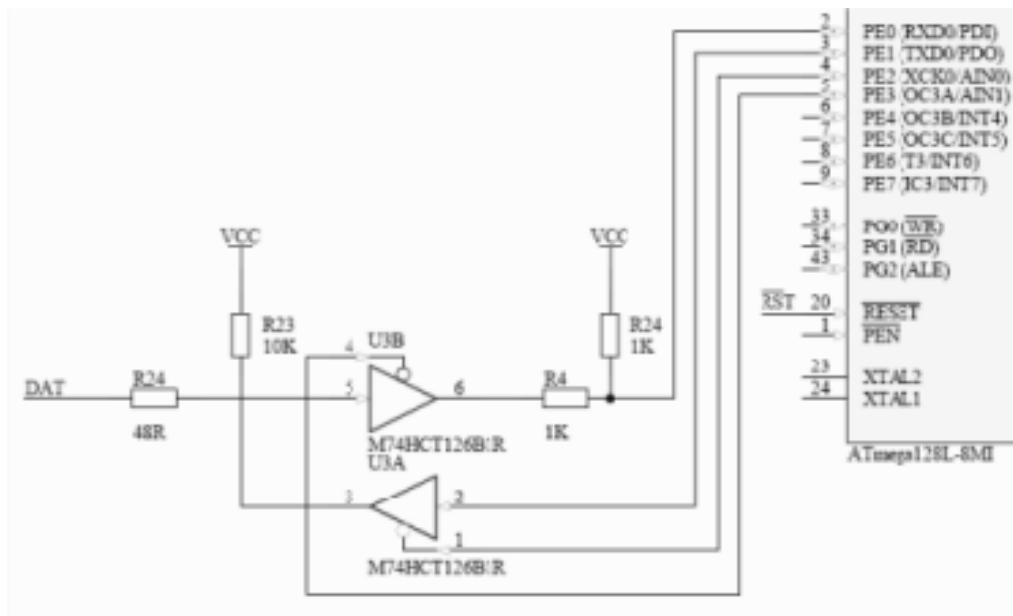


Figure 3.0: Schematic of AX TTL Serial Communication

Figure 3.0 shows the configuration of the AX TTL Serial communication circuitry. This is pretty straight forward, the *DAT* signal is switched to either the *RXD0* or *TXD0* signals, depending on the status of the *PE3* and *PE2* signals respectively. As the CM5 & AX modules act in a half-duplex Master/Slave (CM5/AX module) configuration, using a single wire as both Tx and Rx data is workable.



### 3.2 RS232/ZigBee Serial Communications (ATMega128 UART1)

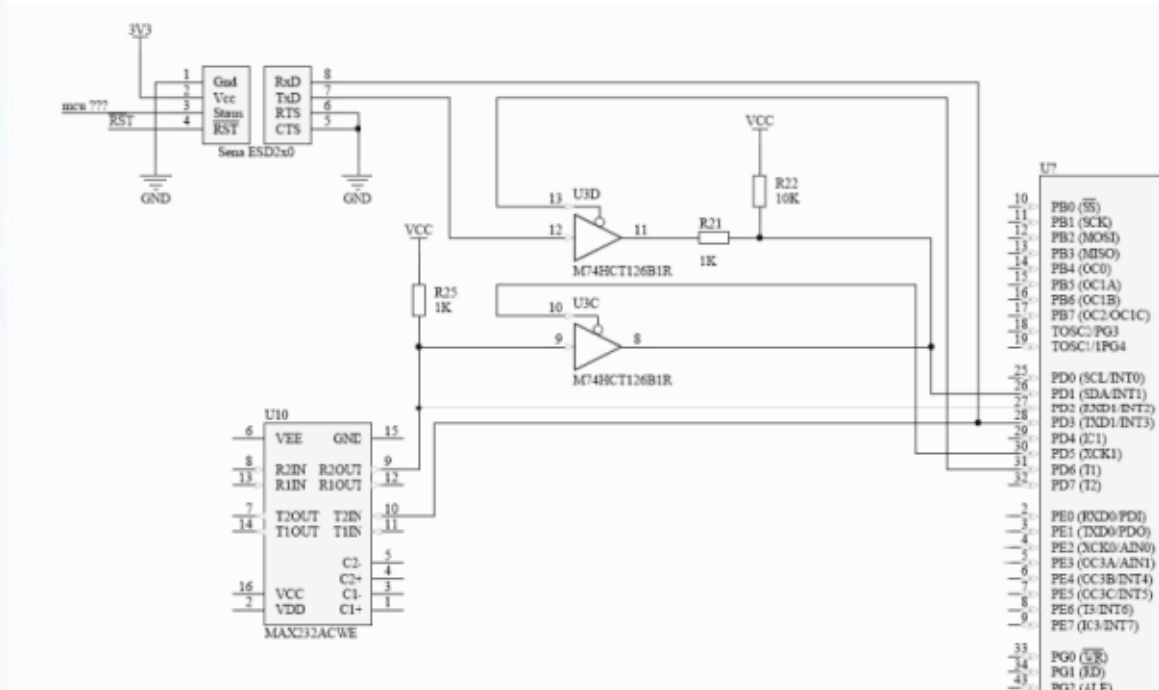


Figure 3.1: Schematic of AX TTL Serial Communication

Figure 3.1 shows the configuration of the RS232/ZigBee Serial communication circuitry. This is a little more complicated than that described previously in section 3.1.

#### **TXD1**

The *TXD1* signal is quite straightforward. Basically the *TXD1* output from the ATmega has enough drive to drive both the MAX232 style ic and the ESD200.

#### **RXD1**

The *TXD1* signal was a case of 1 output port having enough drive capacity to drive 2 inputs. The *RXD1* is the opposite case where there are 2 outputs and 1 input port, so the outputs must be switched – hopefully so that both are never on simultaneously. This is shown in figure 3.1, here the data from the ESD200 is switched through a 74HCT126 and is enabled via *PD6*, whilst the data from the MAX232 style ic is switched through a 74HCT126 and is enabled via *PD5*.



## 4. CM5 – CM5 Communications

CM5 to CM5 communications is the easiest to setup from a hardware modifications perspective, i.e. you don't have to cut as many tracks and add as many components.

### 4.1 ESD setup

Programming the ESD200 modules requires an ESD Parani Starter Kit board, an image of this is shown below:

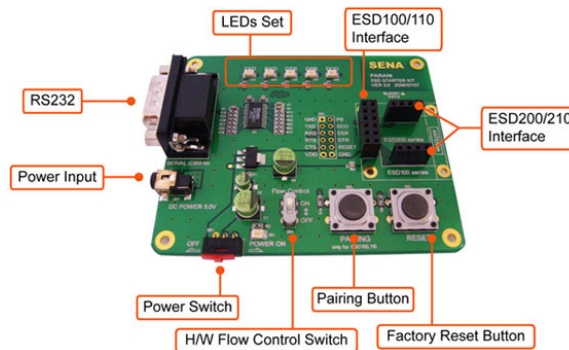


Figure 4.1.1 ESD Parani Starter Kit board

When you have one of these, it's simply a matter of installing the software [4] which is provided and connecting the Starter Kit board to a COM port. Providing you have done both of these successfully, you can follow the following procedure to 'pair' 2 ESD200 so on power-up they will always establish a Bluetooth connection between 2 CM-5's.

NB. When the ESD200 are 'paired' they then establish a point-to-point communication link. You can't setup anything fancy like a piconet, just a simple point-to-point communication link is all that will be available.

The ESD200's factory setting are shown below in figure 4.1.2:

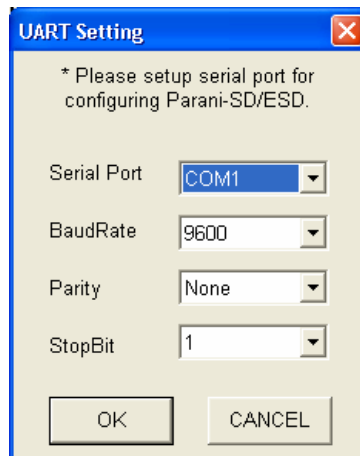


Figure 4.1.2 ESD200 Default Settings

You'll need to specify these, if they aren't done for you automatically, to be able to configure the ESD200 module.

In establishing a CM5 to CM5 communication link, you'll need to assign one ESD200 module as a Master and the other as a Slave. On power-up, the Master will attempt to re-establish the connection with the Slave automatically. The Parani-SD/ESD Configuration software [4] allows you to setup the ESD200's in 4 modes – MODE0 to MODE3. What needs to be done is:

Master: MODE1 (The Parani-SD shall connect to the last connected device only), and  
Slave: MODE2 (The Parani-SD shall be connected from the last connected device only).

Before discussing the procedure, there are a couple of peculiarities of the software that should be mentioned:

1. The software appears to be originating from Korea and all the messages that pop-up are meaningless to the user.
2. One of the options available is to select *H/W Flow Control*, i.e. RTS/CTS, BUT if you select no hardware flow control you get an error message that I think says you can't do this. BUT, having this set doesn't seem to matter – the ESD200 modules still work even with the RTS/CTS signals NOT connected (this is strange, I don't understand this but it works).

To make this easier, the programming software provides a connection Wizard which steps the user through the procedure of setting up and pairing the ESD200 modules. This procedure is very simple to follow, so I won't repeat this here, BUT **you'll need to make sure that the ESD200's are set so that the Baud Rate is 57600kbps, No Parity and 1 Stop Bit.**



Figure 4.1.3 shows the initial screen before starting the 'pairing' procedure.

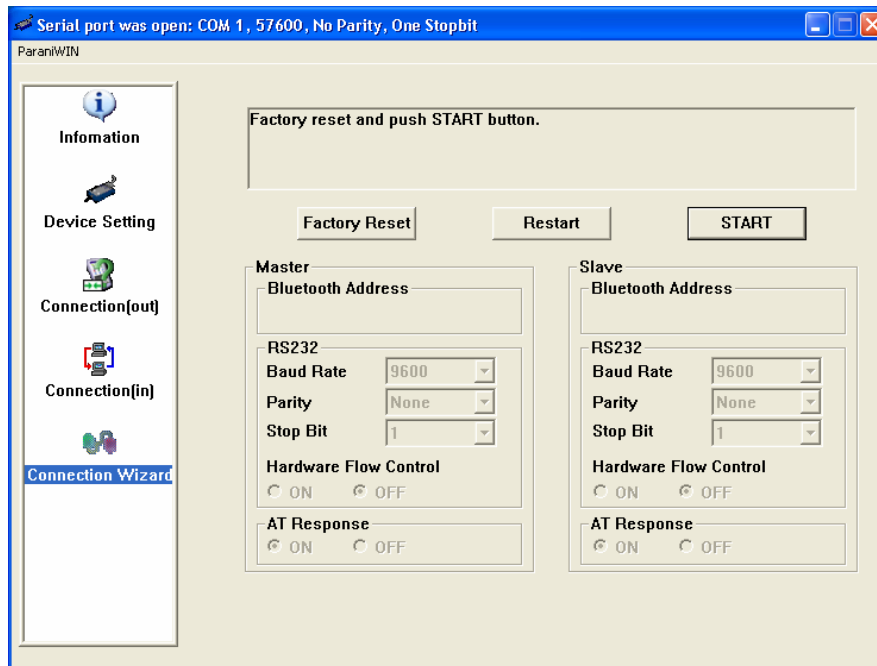


Figure 4.1.3: Screen capture of Connection Wizard before 'pairing' has started.

Figures 4.1.4 and 4.1.5 show the settings of the two modules after the 'pairing' procedure has been successfully completed. If you select the Information screen and press the REFRESH button there parameters should be seen.

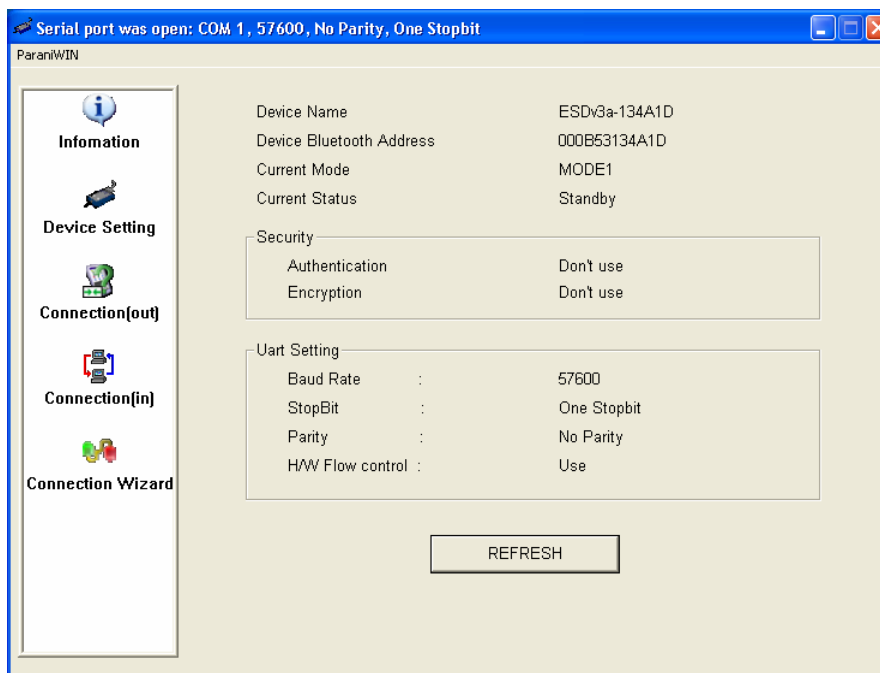


Figure 4.1.4: Screen capture of Master settings after 'pairing' procedure.

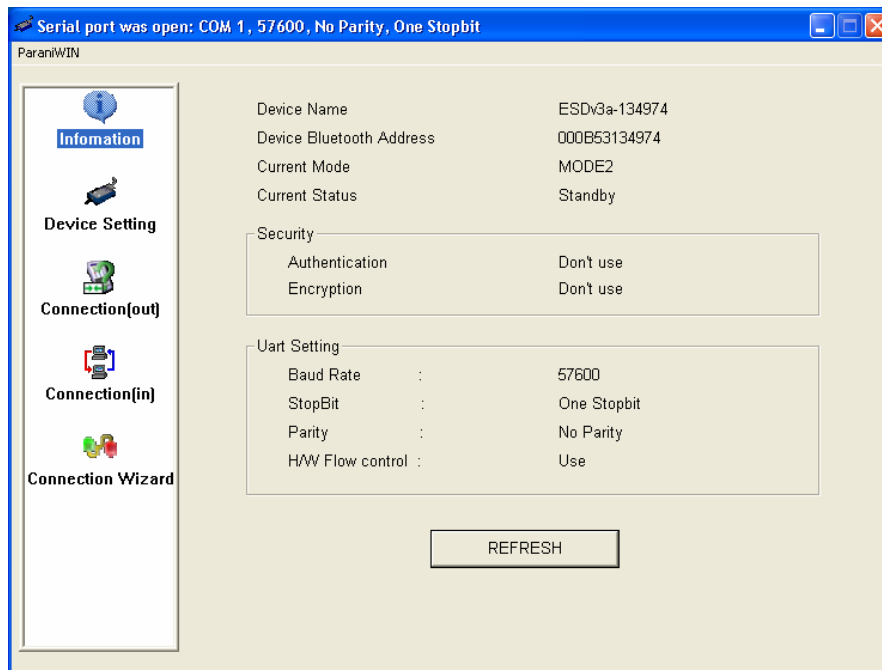


Figure 4.1.5: Screen capture of Slave settings after 'pairing' procedure.

As far as setting up the ESD200 modules via software, that's all that is required. Next is the harder task of the hardware modifications.



## 4.2 Hardware modifications

The complete schematic of the modified serial communications circuits of the CM5 be accessed with the associated zip file, *CM5 to CM5 Comms.pdf*.

The hardware modifications that need to be performed are:

1. Cut Tracks:
  - (i) Reset signal – stop mcu RESET from resetting the ESD200, and
  - (ii) RTS & CTS – disconnect the tracks that connect RTS & CTS to Gnd.

These modifications are shown below:

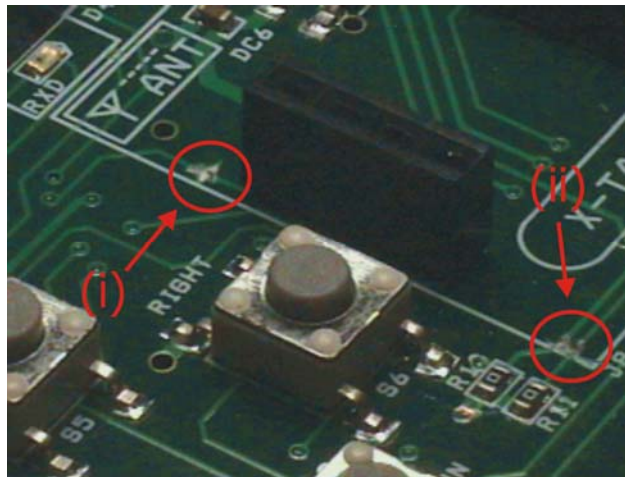


Figure 4.2.1: View of TOP of CM5 showing tracks cut.

2. Solder on an R & C for the *RESET* signal.

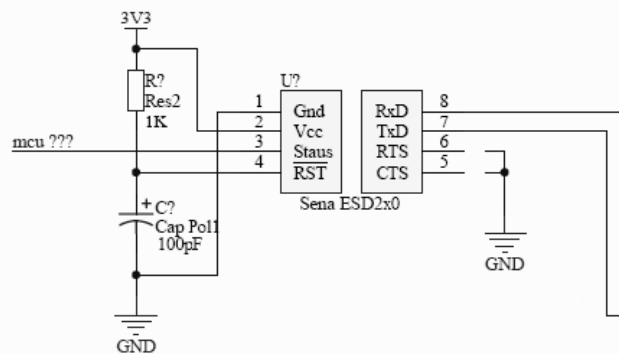


Figure 4.2.2: Schematic showing how R & C are connected.  
**NB values for R and C are 10kΩ and 1a TAG uF**

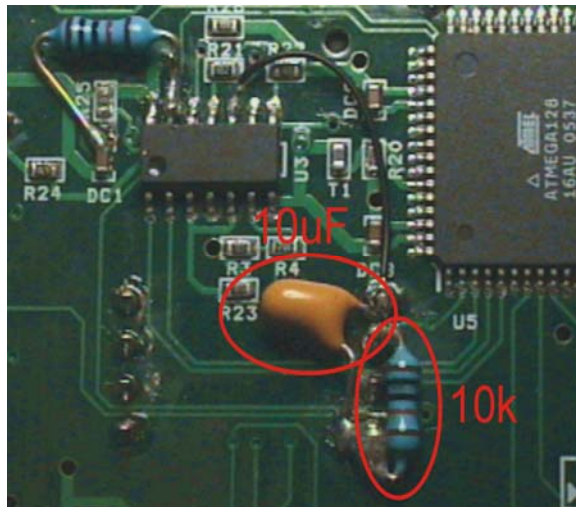


Figure 4.2.3: View of BOTTOM of CM5 showing tracks cut.  
**NB the polarity of C must be considered.**

As I was continually plugging/unplugging the ESD modules I soldered in 2 x 4 pin SIP sockets, BUT if you do this the CM-5 won't fit back into the original casing so you'll need to solder straight to the pcb.

That's all you need to do, cut 3 tracks and add a Resistor and a Capacitor.

### 4.3 Example

It doesn't matter which CM-5 the Master of Slave ESD200's are installed in for this (or any other example). But when you turn the CM-5's on it's best to turn the one with the Slave ESD200 fitted first, so when the Master ES200 is turned on the Salve is already active and waiting to be connected to.

There are 2 example programs supplied on the Bioloid CD that you can use to test if this has worked, these are:

- Example(RF transmitter).bpg
- Example(RF receiver).bpg

Obviously, downloading each of these programs to two modified CM-5's will allow you to turn ON/OFF the Aux LED on the receiver by pressing the UP/DOWN pushbuttons on the transmitter. This is pretty easy to do using the Behaviour Control Program's GUI.



## 4.4 Protocol

When you use the Behaviour Control Program (BCP) to send/receive data it uses a very primitive packet structure. You really don't need to know this unless:

- you are writing some code on the other 'paired' mcu and this code is not being written in the BCP (e.g. C), or
- you are writing some code on a PC that has a dongle which is 'paired' to the ESD200 on a CM-5 control module.

The protocol for the CM-5 ZigBee communications has been previously documented [5] as being:

**0xFF 0x00 LSB ~ LSB MSB ~ MSB**

The ~ operator indicates the inverse (or NOT) operator. This is best shown in an example, if your BCP program to send '1' from one CM-5 to another via the '*Tx remocon data*' command, as shown below in figure 4.4.1,

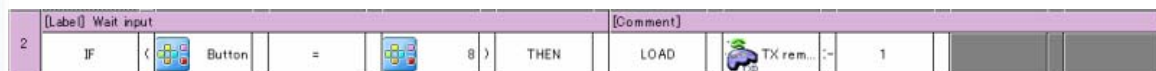


Figure 4.4.1: Behaviour Control Program command to send data via ZigBee/Bluetooth.

Our packet in this case would be:

**0xFF 0x00 0x01 0xFE 0x00 0xFF**

NB. sending 1 is equivalent to send 0x0001 in a 16bit word format, this is why the last two bytes (0x00 and 0xFF) are required.

## 4.5 Advantages & Disadvantages

### Advantages:

- Good for setting up 2 autonomous Bioloids to communicate together.
- Good for have a 2<sup>nd</sup> CM-5 control module to act as a Remote Control for the CM-5 within the Bioloid.

### Disadvantages:

- Using this method you can only connect to the same CM-5 or ESD200 – this may limit what you'd like to do in the future.



## 5. CM5 – PC Communications

CM5 to PC communications is a little bit more difficult from a hardware modifications perspective, i.e. you need to cut a few more tracks and add a couple more components.

### 5.1 ESD setup

Programming the ESD200 modules requires an ESD Parani Starter Kit board, an image of this is shown below:

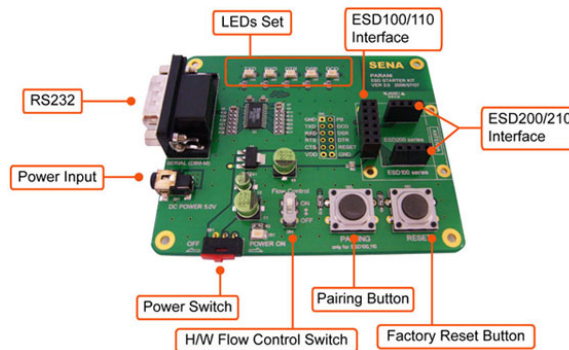


Figure 5.1.1 ESD Parani Starter Kit board

When you have one of these, it's simply a matter of installing the software [4] which is provided and connecting the Starter Kit board to a COM port. Providing you have done both of these successfully, you can follow the following procedure to configure an ESD200 so on power-up you will be able to see the device and establish a connection with it from a PC.

The ESD200's factory setting are shown below in figure 5.1.2:

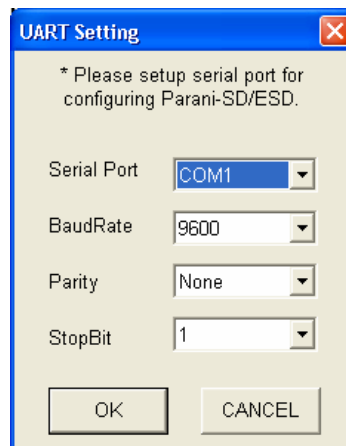


Figure 5.1.2 ESD200 Default Settings



You'll need to specify these, if they aren't done for you automatically, to be able to configure the ESD200 module.

In establishing a CM5 to PC communication link, you'll need to configure the ESD200 module in MODE3 (Allow any Bluetooth devices discover/connect to this Parani-SD). On power-up, the ESD200 will wait for a request to establish a communication link, which in this case will be initiated by the PC's Bluetooth Dongle.

Before discussing the procedure, there are a couple of peculiarities of the software that should be mentioned:

1. The software appears to be originating from Korea and all the messages that pop-up are meaningless to the user.
2. One of the options available is to select *H/W Flow Control*, i.e. RTS/CTS, BUT if you select no hardware flow control you get an error message that I think says you can't do this. BUT, having this set doesn't seem to matter – the ESD200 modules still work even with the RTS/CTS signals NOT connected (this is strange, I don't understand this but it works).

Figure 5.1.3 shows the settings that are required for CM5 to PC communication, obviously the Device Name can be anything you like.

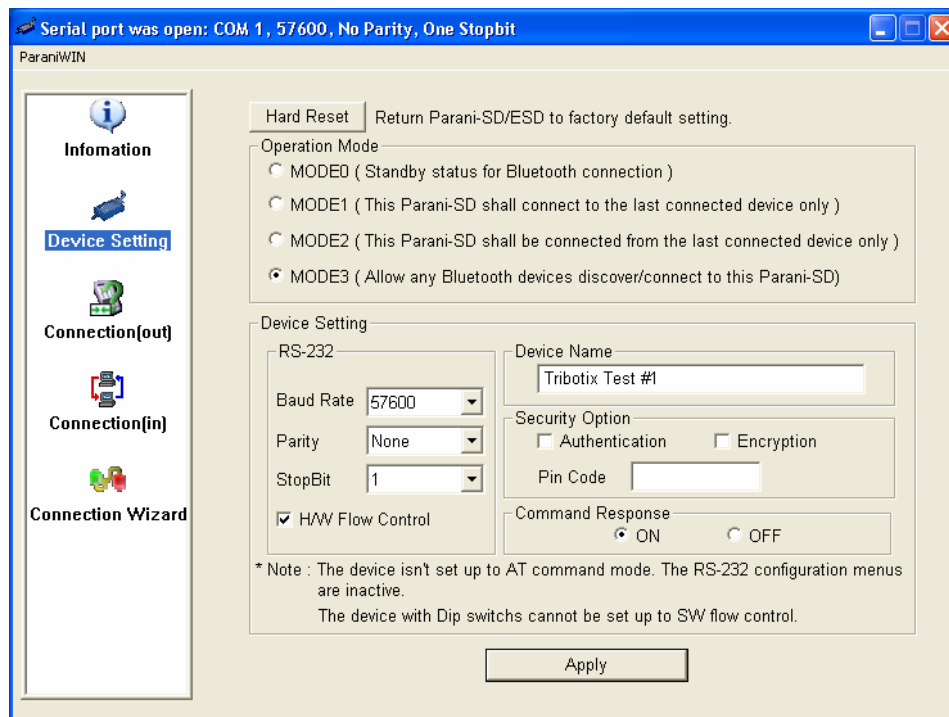


Figure 5.1.3 ESD200 settings for CM5 to PC communication



**NB. You'll need to make sure that the ESD200 is set so that the Baud Rate is 57600kbps, No Parity and 1 Stop Bit.**

After hitting the APPLY button, you can select the Information page which should be the same as that shown in figure 5.1.4.

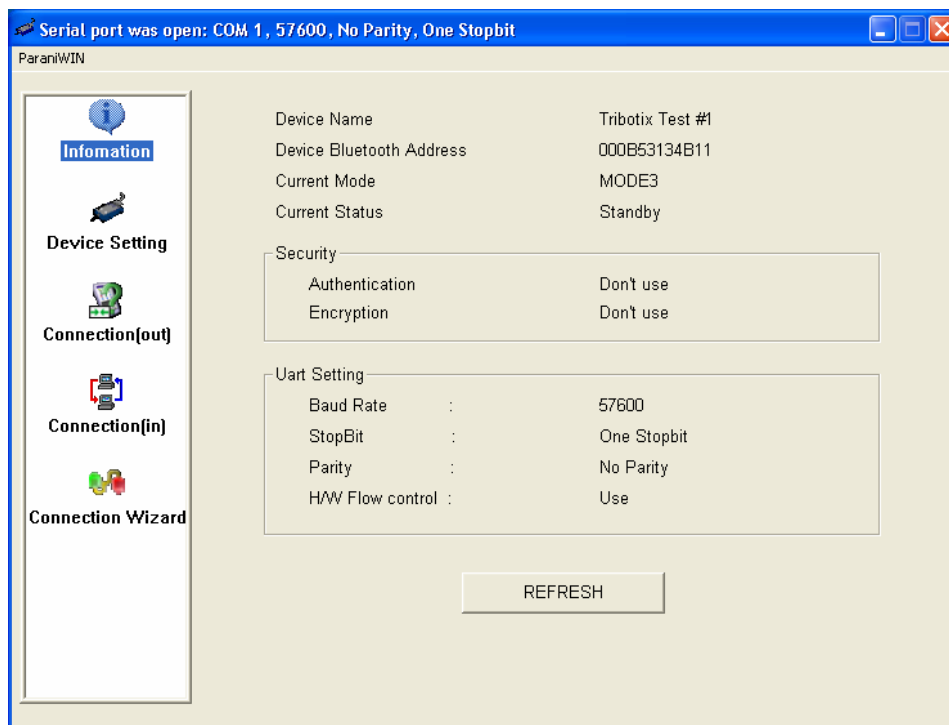


Figure 5.1.3 Screen capture of ESD200 settings configuration.

That's all that needs to be done to setup the ESD200. Next is the harder task of the hardware modifications



## 5.2 Hardware modifications

The complete schematic of the modified serial communications circuits of the CM5 can be accessed with the associated zip file, *CM5 to PC Comms.pdf*.

The hardware modifications that need to be performed are:

1. Cut Tracks:
  - (i) Reset signal – stop mcu RESET from resetting the ESD200, and
  - (ii) RTS & CTS – disconnect the tracks that connect RTS & CTS to Gnd.

These modifications are shown below:

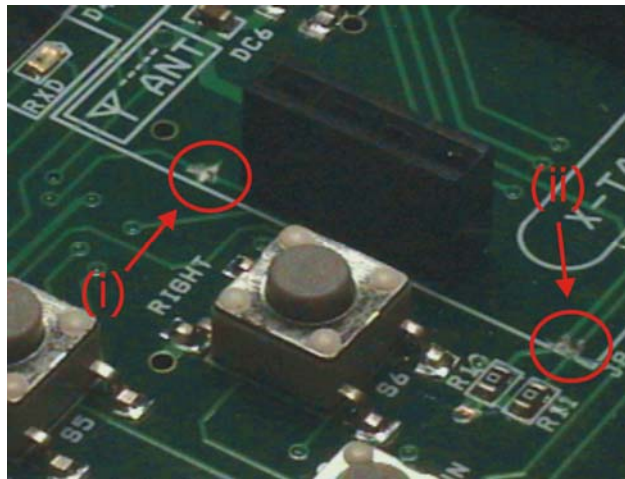


Figure 5.2.1: View of TOP of CM5 showing tracks cut.

2. Solder on an R & C for the *RESET* signal.

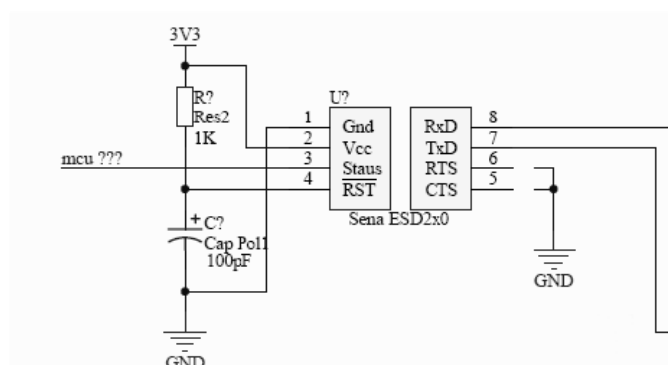


Figure 5.2.2: showing how R & C are connected.  
**NB values for R and C are 10k $\Omega$  and 1a TAG  $\mu$ F**

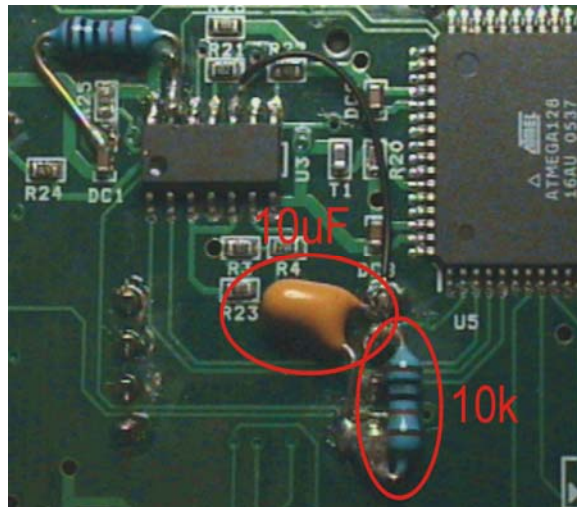


Figure 5.2.3: View of BOTTOM of CM5 showing tracks cut.  
**NB the polarity of C must be considered.**

3. Cut Tracks:
  - (i) U3 p10, and
  - (ii) U3 p13.

This disconnects the mcu's control signals to the 74HC126.

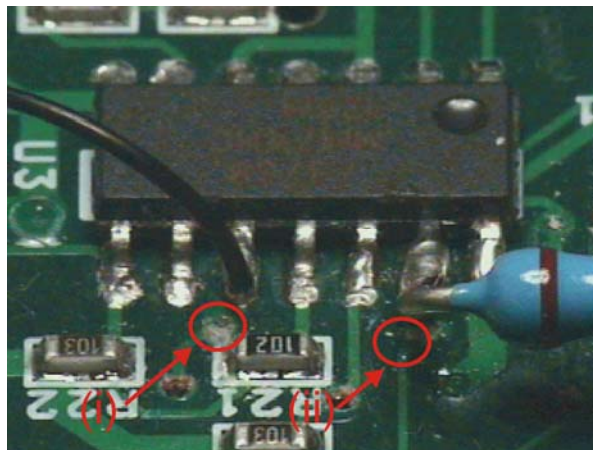


Figure 5.2.4: View of BOTTOM of CM5 showing tracks cut.

4. Connect a 10k pull-up resistor and a 0V link to U3 pins 13 and 10 respectively.

This allows the ESD200 to always be activated, it is this mod that gives you the ability to program and monitor the CM5 via the Bluetooth link.

To get access to 5V I used the top of DC1 and to get access to a 0V I used the 0V from the ESD200 (as shown in figure 5.2.5).

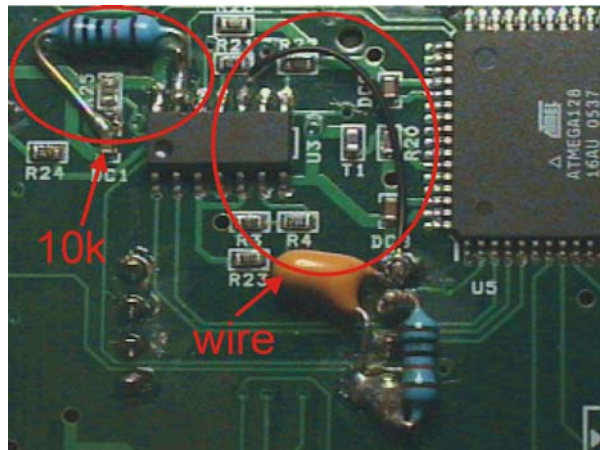


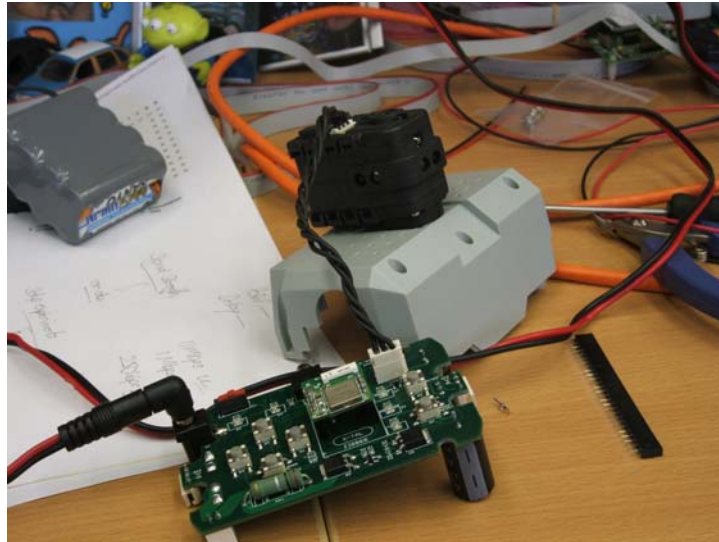
Figure 5.2.5: View of BOTTOM of CM5 showing tracks cut.

As I was continually plugging/unplugging the ESD modules I soldered in 2 x 4 pin SIP sockets, BUT if you do this the CM-5 won't fit back into the original casing so you'll need to solder straight to the pcb.

That's all you need to do, cut 5 tracks and add 2 Resistors, a Capacitor and a link.



### 5.3 Example



The best way to show how this works is with an example. In this example I connected an AX-S1 to the CM-5 and wrote a very simple program to read the status of all 3 IR's in the AX-S1 and display these on the PC.

I use a D-Link DBT122 Bluetooth Dongle, which this example is based around. The driver for the DBT-122 allows you to set the USB link up as a pseudo COM port, doing this allows you to use the Bluetooth Dongle as a COM port.

If you open the *My Bluetooth Places* folder and select the *Search for devices in range* option. After 10 seconds or so you should see a new Bluetooth device, in my case this is 'Test Tribotix #1', as shown below in figure 5.3.1.

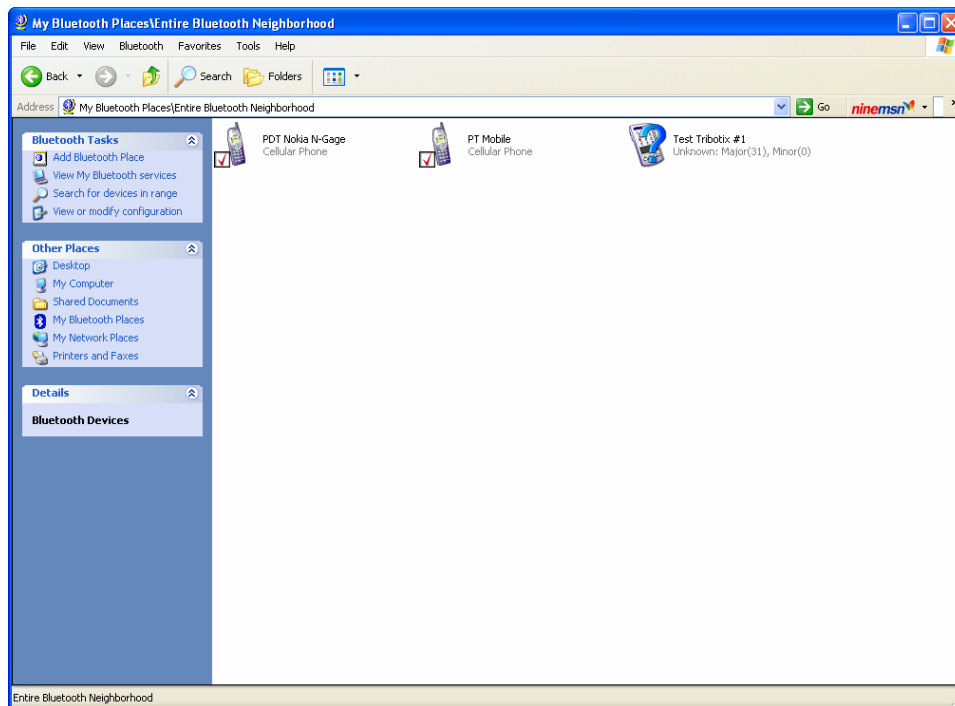


Figure 5.3.1

To establish a generic serial selection, simply highlight the icon for the new device and select the Connect Generic Serial option, as shown below in figure 5.3.2.

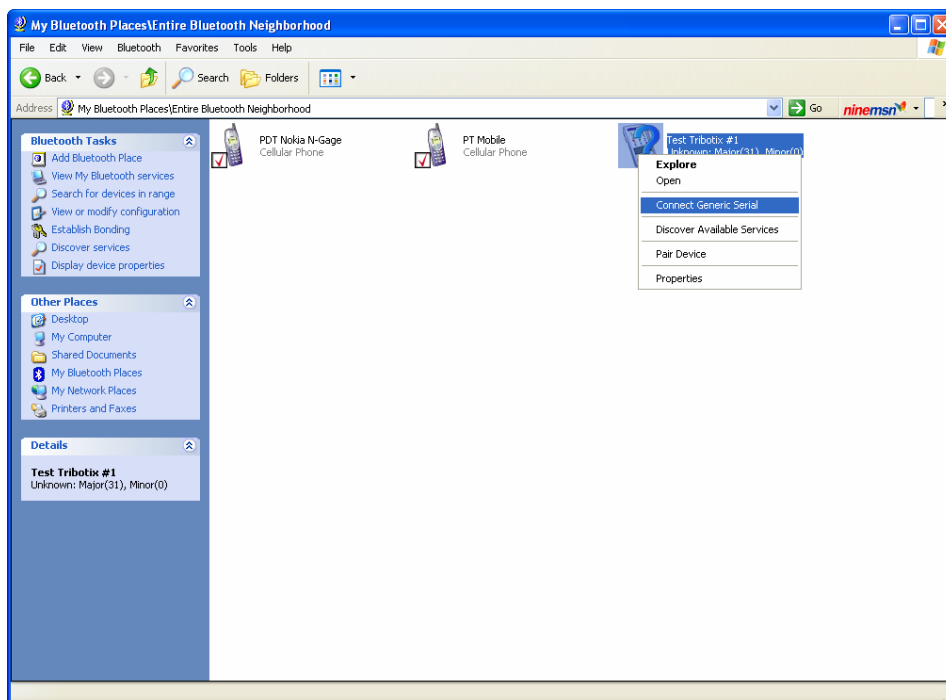


Figure 5.3.2



If this is successful, the icon will change as shown below in figure 5.3.3.

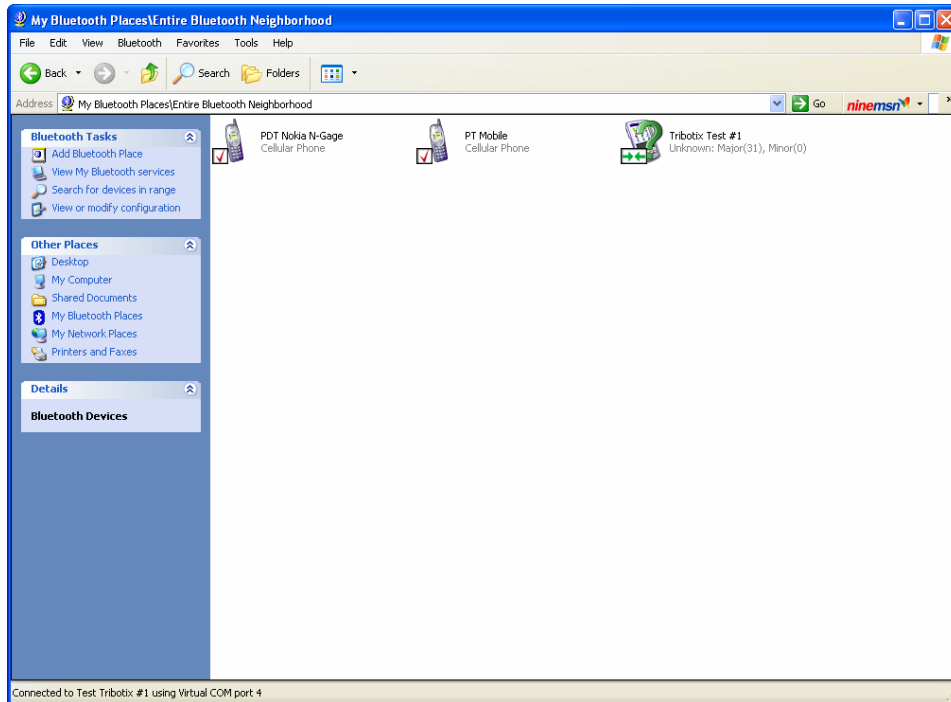


Figure 5.3.3

If you then double click on this icon you will see that a serial connection has been completed between your device and the ESD200 in the CM5, this is shown below in figure 5.3.4

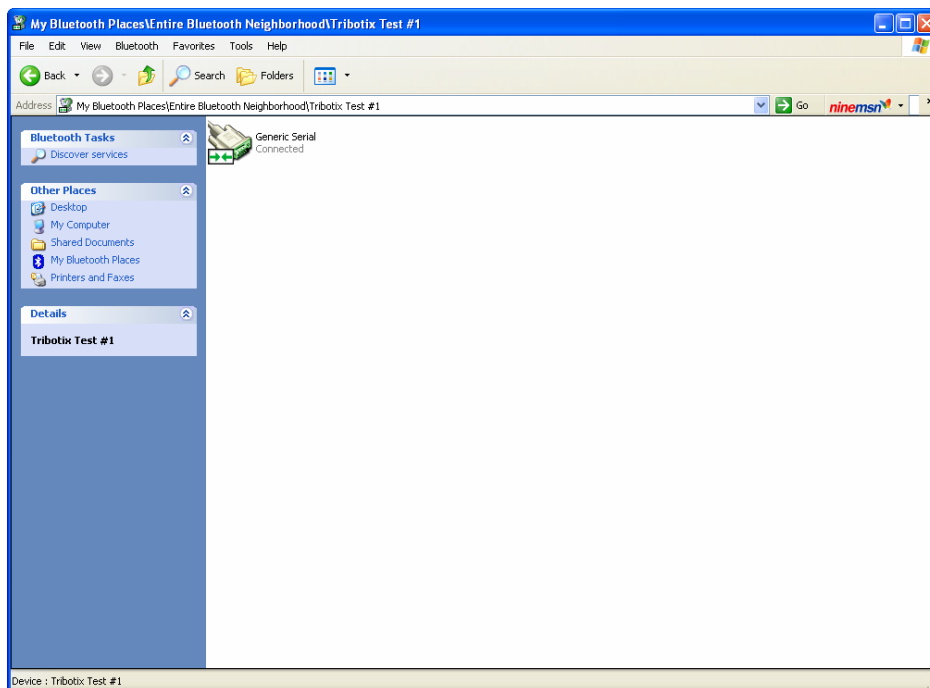


Figure 5.3.4



Once the connection is established, you can then look at the status of the connection by selecting the icon, right clicking the mouse and selecting the *Status* option.

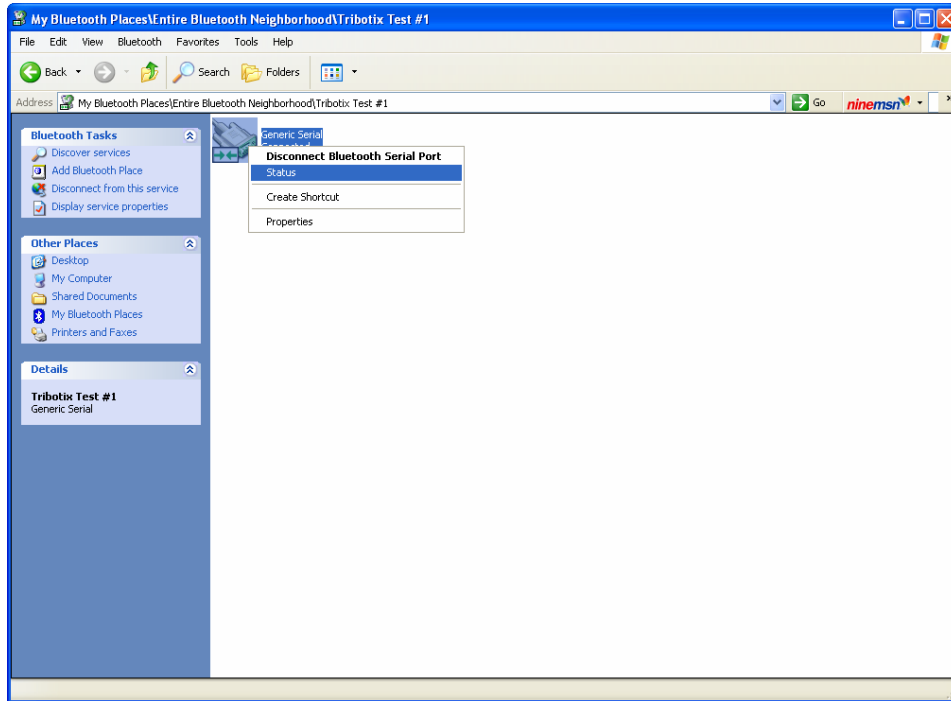


Figure 5.3.5

The window that then opens is shown below in figure 5.3.6 – you'll notice it gives an indication as to the signal strength.

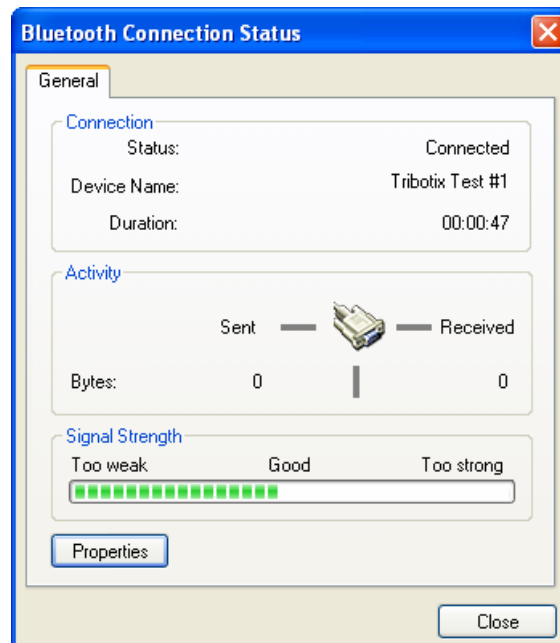


Figure 5.3.6



This *Good* signal was received in my office, with there being line of sight between the 2 devices and a distance of 2 meters. This was not as good as I had hoped for.

Below, figure 5.3.7, shows a screen capture of my simple program. When the Bluetooth Dongle driver was installed it automatically configured COM4 as the COM port that it would use for it's generic serial connection. So all I need to do is select *Program / Download/Play*, select COM4 and I can download my code via the Bluetooth link.

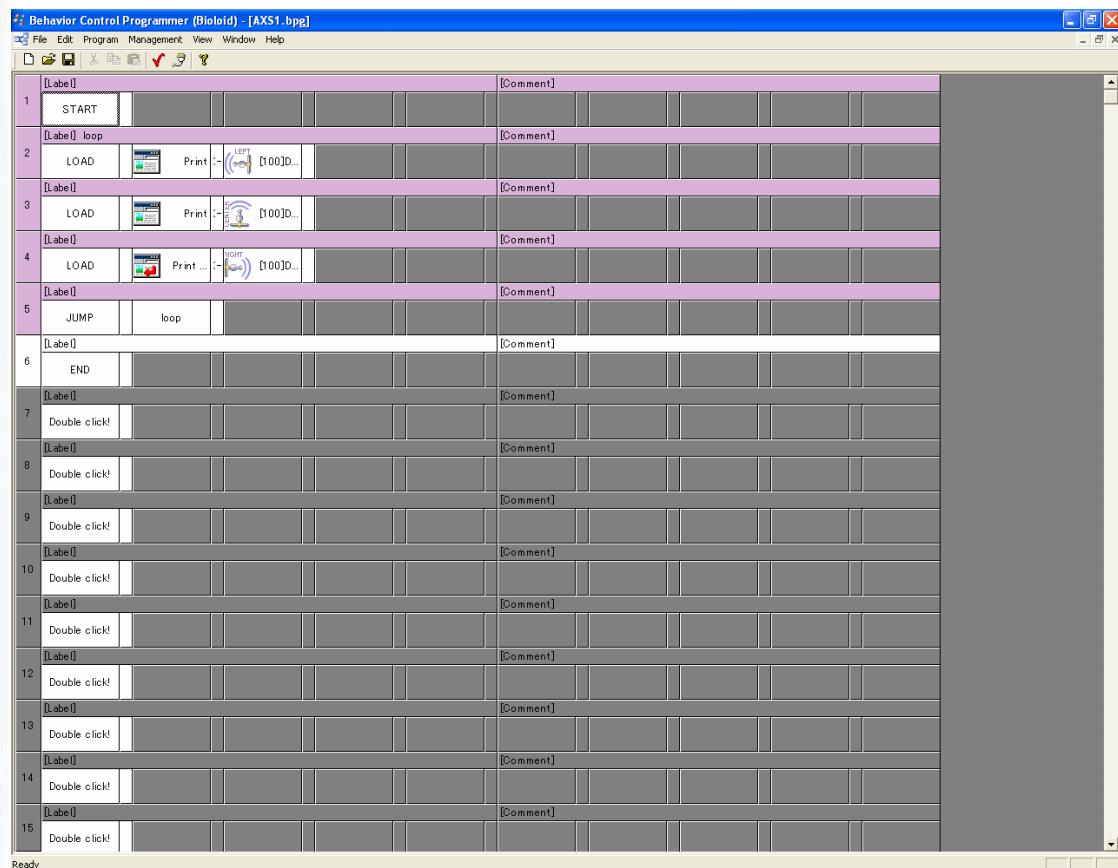


Figure 5.3.7

The Screen Capture below, figure 5.3.8, shows the status just after the download via Bluetooth (COM4 in my case) has just been completed and the *Play* button has been selected. Note that the status of the AX-S1's IR detectors are being displayed via the Bluetooth link, this allows you to get info on the status of your robots sensors in real-time without having a serial cable trailing behind the robot.

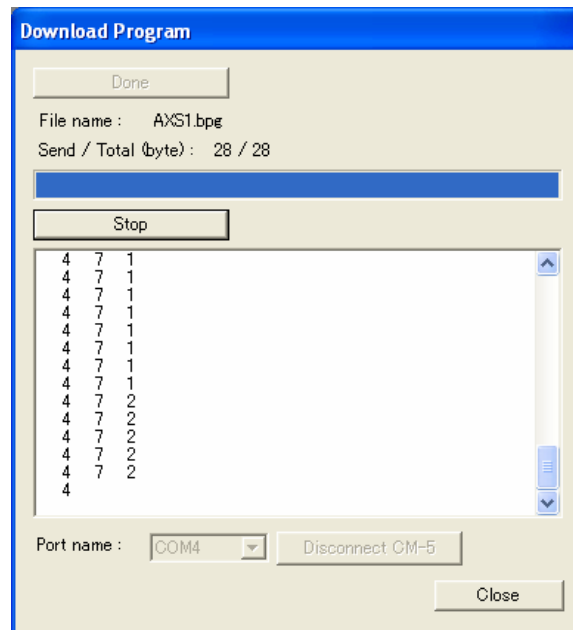


Figure 5.3.8

When you hit the *Close* button on the *Program Download* window, this not only closes the window it also shuts down the Bluetooth communication link. **THIS IS ANNOYING**, but I think this is just Windows releasing resources, i.e. if it were a physical COM port Windows would prevent a single application claiming this resource – so I assume this is what's happening here. All you need to do is go through the connect Generic Serial device again, but this does get annoying.

Another nice thing you can do is run Robot Terminal across the Bluetooth link. Figure 5.3.9 shows a screen capture of this being done, all you need to do is set the COM port for Robot Terminal to COM4.



```
Robot Terminal(Bioloid) v1.01
Setup Files

[CM-5 Version 1.14]
<->PC:57142 BPS, <->Dynamixel:1000000 BPS
ID:100
001(0X01) Dynamixel Found.
[ CID:019(0X13) ]
[ CID:019(0X13) ]
[ CID:019(0X13) ] scan 10
[000:-][001:-][002:-][003:-][004:-][005:-][006:-][007:-][008:-][009:-]
000(0X00) Dynamixel Found.
[ CID:019(0X13) ] cid 1
[ CID:001(0X01) ]

No recent file COM4-57600
```

Figure 5.3.9

## 5.4 Advantages & Disadvantages

### Advantages:

- Good for programming CM5 without serial lead
- Good for getting status of sensors without having a trailing serial cable

### Disadvantages:

- The need to have to reconnect the link each time Windows decides the Behaviour Control Program should hand back resources to the O/S.



## 6. References

1. ESD200 Datasheet,  
[http://www.sena.com/download/datasheet/ds\\_parani\\_esd.pdf](http://www.sena.com/download/datasheet/ds_parani_esd.pdf)
2. ESD200 User Guide  
[http://www.sena.com/download/manual/manual\\_parani\\_esd-v1.0.4.pdf](http://www.sena.com/download/manual/manual_parani_esd-v1.0.4.pdf)
3. D-Link DBT122  
<http://www.dlink.com.au/Products.aspx?Sec=1&Sub1=18&Sub2=39&PID=211>
4. ESD Configuration Software  
[http://www.sena.com/download/software/setup\\_esd\\_conf-v1.0.1.exe](http://www.sena.com/download/software/setup_esd_conf-v1.0.1.exe)
5. Bioloid Bluetooth Interface Hack  
<http://robosavvy.com/Builders/Pev/Bioloid%20BlueTooth%20Hack.pdf>