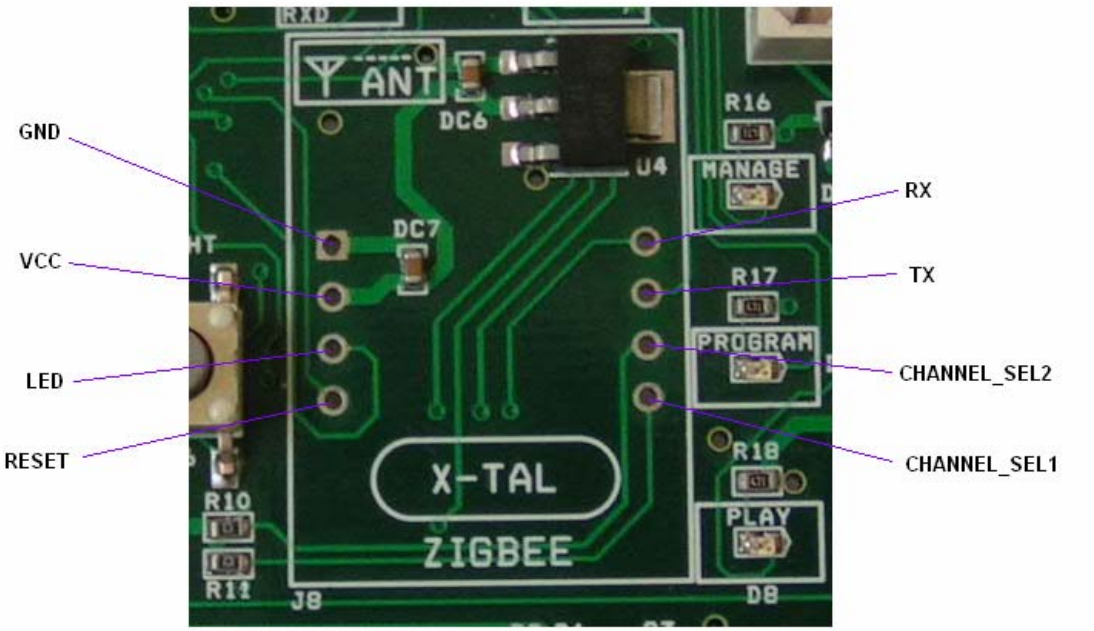
Bioloid BlueTooth Interface Hack

Warning

WARNING : This hack is not warranted in anyway. You do this at your own risk. I am fairly sure that Robotis would see this as voiding any warranty they provide on their product and if you get this wrong you could easily damage your Bioloid CM-5 processor module.

The Bioloid Zig-100 Interface

The Bioloid CM-5 processor module contains an interface on the processor board for a ZigBee based wireless link. This interface is documented in the Zig-100(English).pdf file provided by Robotis and the document details the pin-out for the 'port' on the CPU pcb. The pin-out is as follows :



For the purposes of attaching a BlueTooth module we are only interested in GND, VCC, TX and RX. The rest are redundant for this exercise.

Note : VCC is a regulated voltage of 2.7 to 3.3 volts (although I have not seen it at less than 3.2) and as such the bluetooth module you select must be able to run at around 3volts.

These details allow us to connect a serial device to the port and power it. However, there are still a number of unanswered questions.

Firstly, the serial communications speed. I determined from the Zig-100 documentation that the port runs at 57600 baud, 8 data bits, 1 stop bit, No Parity and no flow control. With a Zig-100 module you can drop into a configuration mode and change this but in our BlueTooth example we can't. So we are stuck at 57600 baud, but this should be more than adequate for a control system.

To test the principles I connected a TTL to RS232 level shifter between the CM-5 Zig-100 port and a PC. Using the Behavior Control Program I set the CM-5 to send out a value of 65 via the TX remocon data command. When viewed in hex on a PC terminal program this produced the following output:

FF 55 41 BE 00 FF

The hex 41 is the value decimal 65 and the other data appeared to be some sort of packet information. Unfortunately I don't think in hex or packets very well so was at a bit of a loss.

Anyway an email to the Robotis people provided the info needed. (Thanks John, very much appreciated)

The Zig-100 Packet Format

Basically the serial protocol used by Robotis is packet based. The packet format is as follows:

Packet Format :

0xFF 0x55 LSB ~LSB MSB ~MSB

Where ~LSB is the inverse or compliment of LSB.

To put that another way if you add up LSB and ~LSB you should end up with 255 decimal or 0xFF (hex)

Lets take our example of the value 65 again. If you want to send 65 via the Zig100 protocol you need to send:

255 85 65 190 0 255

Or

0xFF 0x55 0x41 0xBE 0x00 0xFF

If you add the LSB and ~LSB in the example above you get 255 or 0xFF.

To send larger number such as 0x1234 the LSB is sent first and the MSB second.

0xFF 0x55 0x34 0xCB 0x12 0xED

Remember even if you have a value that only has a LSB you must send a zero value MSB.

As long as you adhere to the packet format the CM-5 will recognize your incoming data and decode it to the value you sent. When the CM-5 sends data it is in this packet format so your receiving program will need to decode it.

The Hardware Hack

Okay, we now know enough to get this working. We will need a BlueTooth module that can support the following:

Operational Voltage : 3.3 volts
Baud : 57600
Data Bits : 8
Stop Bits : 1
Parity : None
Flow Control : None

Once you have a module that can support these requirements you'll need to start hacking the CM-5.

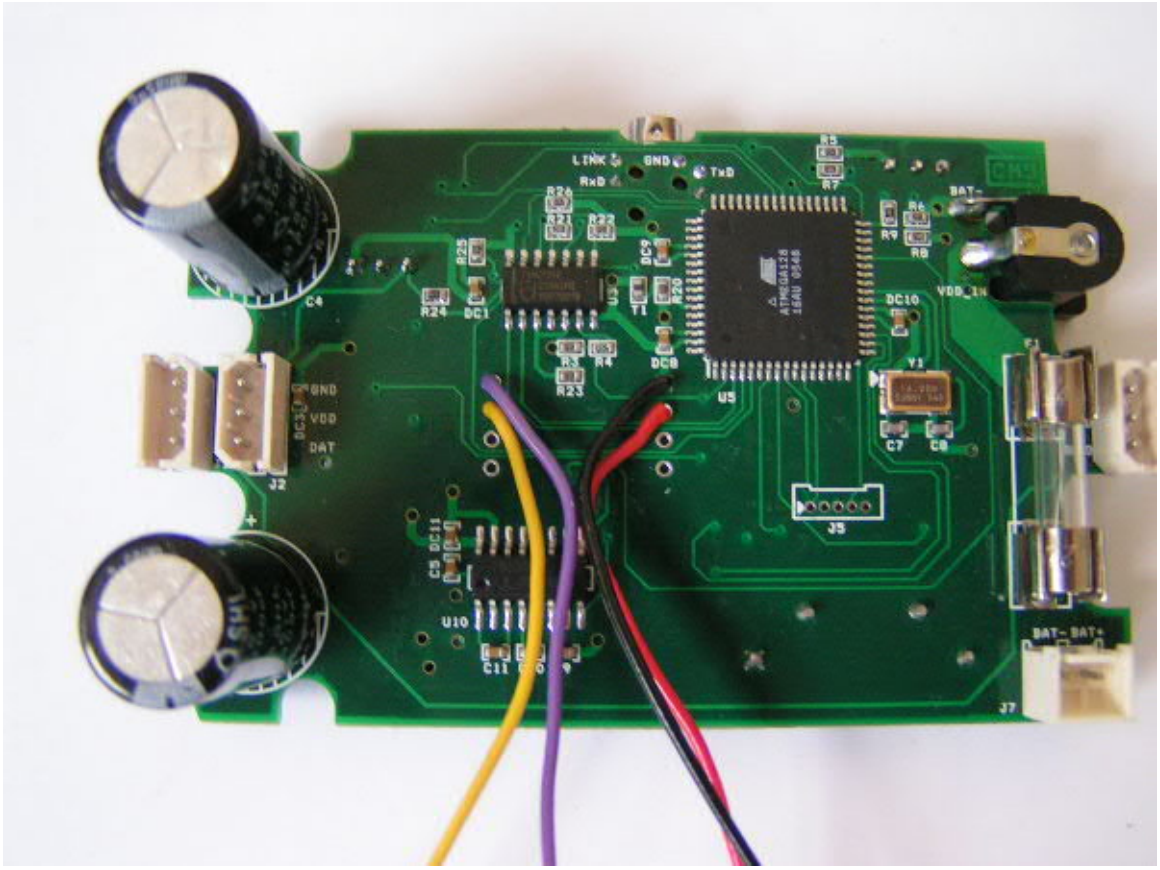
Step 1 – Hooking into the Zig-100 Port

Remove the Processor PCB from the CM-5. Remember your static precautions. If you are not sure how to get it out Have a look at the 'Changing the Fuse' help on the latest Bioloid Software CD. If you need it a link is posted on the www.robosavvy.com forums, in the Bioloid section. Look for 'August 06 update'.

Once you have the Processor PCB removed from the CM-5 case you need to solder 4 wires to the 4 ports detailed above. That's:

GND – Pin 1
VCC – Pin 2
TX – Pin7
RX – Pin 8

Once it's done it should look something like this:



Processor PCB view from the back – Pin-out Reversed

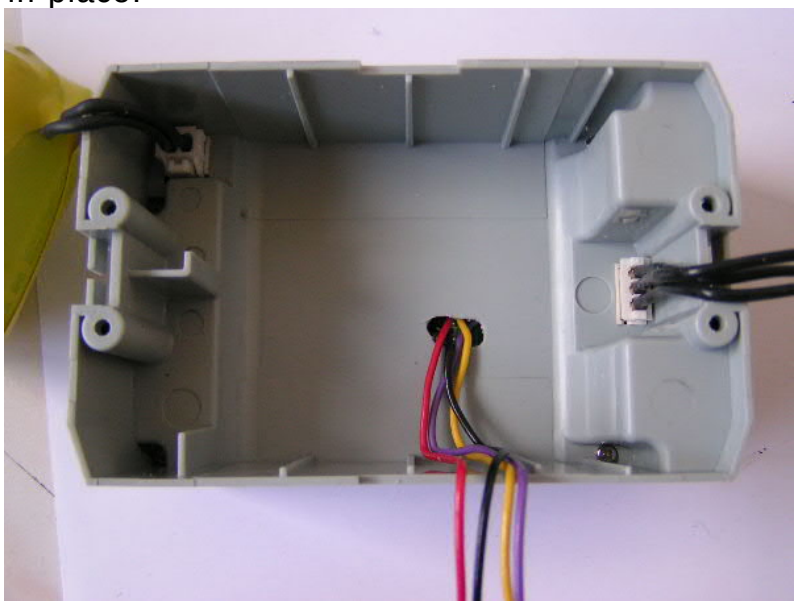
As you can see I passed the wires through from the ‘back’ of the PCB and soldered them on the front. This was just to allow me to route the wires back into the battery box on the CM-5 case without having to pass them round the pcb. By turning the PCB over you will reverse the pin/hole positions from the diagram at the start of this document.

Step 2 – Routing the Wiring

Next you need to cut a small hole to pass the wires through. I hope you do a better job than I did, mines a bit ragged. This hole needs to the cut in the plastic wall between the processor and the battery compartment.

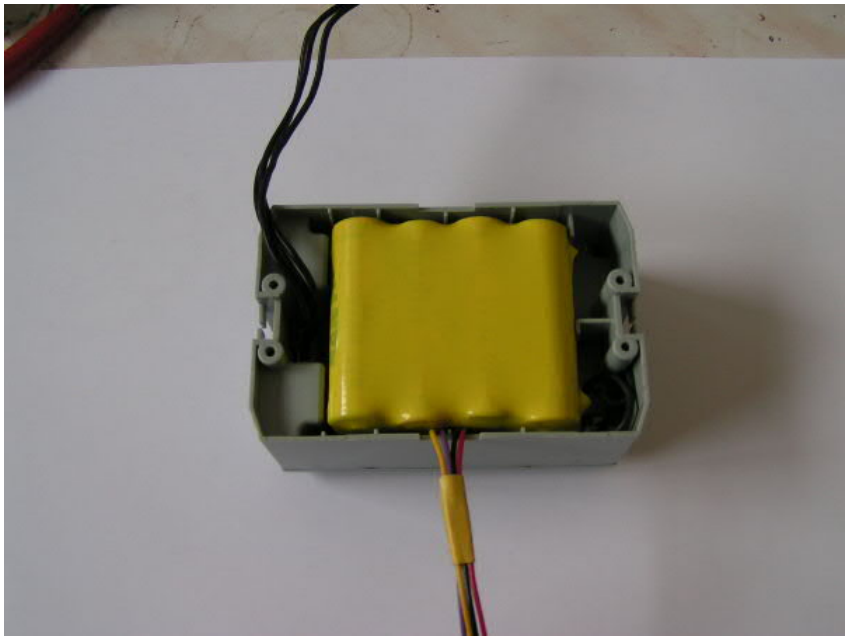
Hole Drilled in Plastic Battery Box of CM-5

Then you need will need to route your wires through the hole and re-install the Processor PCB. The back panel with all the buttons can be re-secured in place.



Processor Re-Installed and Wires Routed

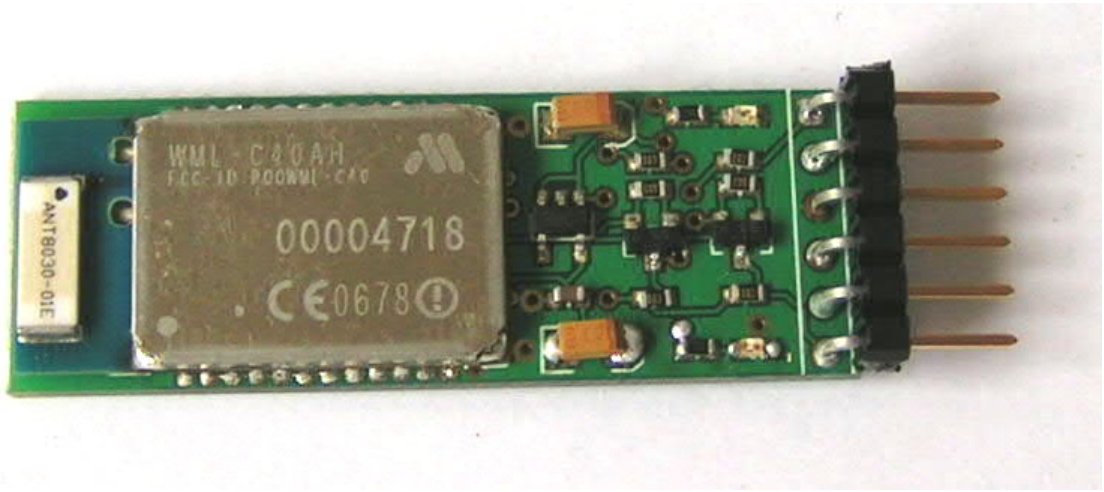
Finally check that the battery still fits correctly with the wire routing.



Battery Re-Installed

Step 3 – Attaching the Bluetooth Module

Now is a good time to attach the BlueTooth module and test everything. As I said I used a BlueSmirf module that is somewhat larger than the Zig-100 and subsequently needs to be mounted outside the CM-5 unit.



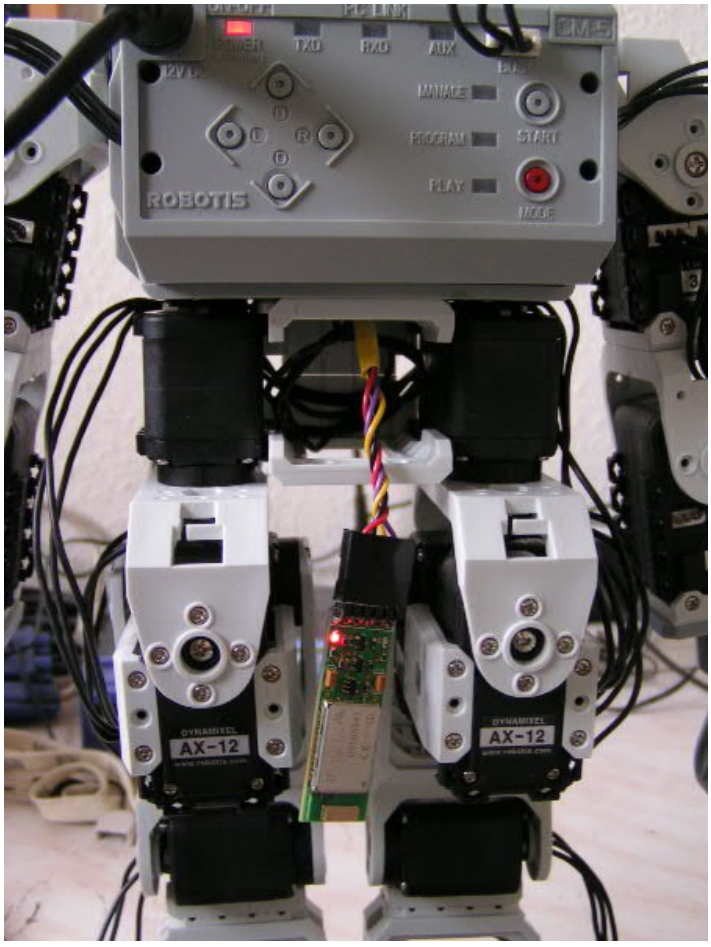
BlueSmirf BlueTooth Modem Module – Approx 50mm long

The BlueSmirf unit has 6 pins, all labeled on the reverse of the board. There is also a solder jumper to disable flow control. I, personally, always make the solder jumper and cut off the CTS pin. That way I can make a non-reversible connector for the other connections.

Anyway, connect your bluetooth module as follows:

Zig-100 Pin	Bluetooth Pin
Pin 1 - Ground	Ground
Pin 2 - VCC	VCC
Pin 7 – TX	TX (well on a BlueSmirf anyway)
Pin 8 – RX	RX

I do have a bit of an issue with manufacturers using TX and RX. It's easy for this to get confusing. Is it transmit from the unit or where you should plug your transmit connection into. It caused some real confusion in the early RN-1 days. If you have an issue when you test the first thing I would do is reverse the TX and RX wires on the BlueTooth module.



BlueSmirf Connected to Bioloid Humanoid for Testing

Now configure your BlueTooth module for 57600 baud, 8 data bits, 1 stop bit and no parity. Then setup your bluetooth connection on a PC

to the same configuration and connect. I tend to use Hyper Terminal for the test but any terminal emulator should work. This is all documented in the BlueSmirf manual so I'll assume it is for other modules.

Once you have a connection, put the CM-5 in play mode and press start. You should see the CM-5 banner on your terminal emulator screen. Something like this:

[CM-5 Version 1.14]

<->PC:57142 BPS, <->Dynamixel:1000000 BPS

ID:001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 100

019(0X13) Dynamixels Found.

My screen shot had 18 AX-12s and an AX-S1 attached so yours may not be identical.

Congratulations you have a working link. Well on the transmit from CM-5 side anyway.

Step 4 - Testing the Ability to Send to the CM-5

You will need something to send data to the CM-5 for this test. I knocked up a quick VB.NET 2005 program to send a single packet then expanded it from there.

The VB.NET program needs to setup a serial port to 57600 baud,8 databits,1 stopbit, no parity and then send a packet. I sent my value 65 packet and the VB.NET 'send' code looked like this:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    Dim Mybytes() As Byte = {255, 85, 65, 190, 0, 255}

    SerialPort1.Write(Mybytes, 0, 1)
    SerialPort1.Write(Mybytes, 1, 1)
    SerialPort1.Write(Mybytes, 2, 1)
    SerialPort1.Write(Mybytes, 3, 1)
    SerialPort1.Write(Mybytes, 4, 1)
    SerialPort1.Write(Mybytes, 5, 1)

End Sub
```

Now you will need a Behavior Control Program at the Bioloid end to receive the data and do something. My test program was very simple and I hope this is clear, as BCP programs are difficult to write out sometimes:

```
START
LABEL LOOP
IF (CM-5/RX REMOCON DATA ARRIVE) != 0 then LOAD (CM-5/AUX LED = 1)
JUMP LOOP
END
```

This BCP program will turn on the AUX LED if a valid packet is received over the BlueTooth link. You could expand this and get it to toggle the LED on and off with different values by looking at the CM-5/REMOCON DATA value when the CM-5/REMOCON DATA ARRIVED flag is triggered.

Hopefully your LED turned on and you have a fully working link.

In reality that is it. The BlueTooth module can now be used to send and receive data in a BCP program. There are a couple of things I haven't managed to get to work over it though:

1. I cannot run the motion editor over this link
2. I cannot upload new BCP programs over this link

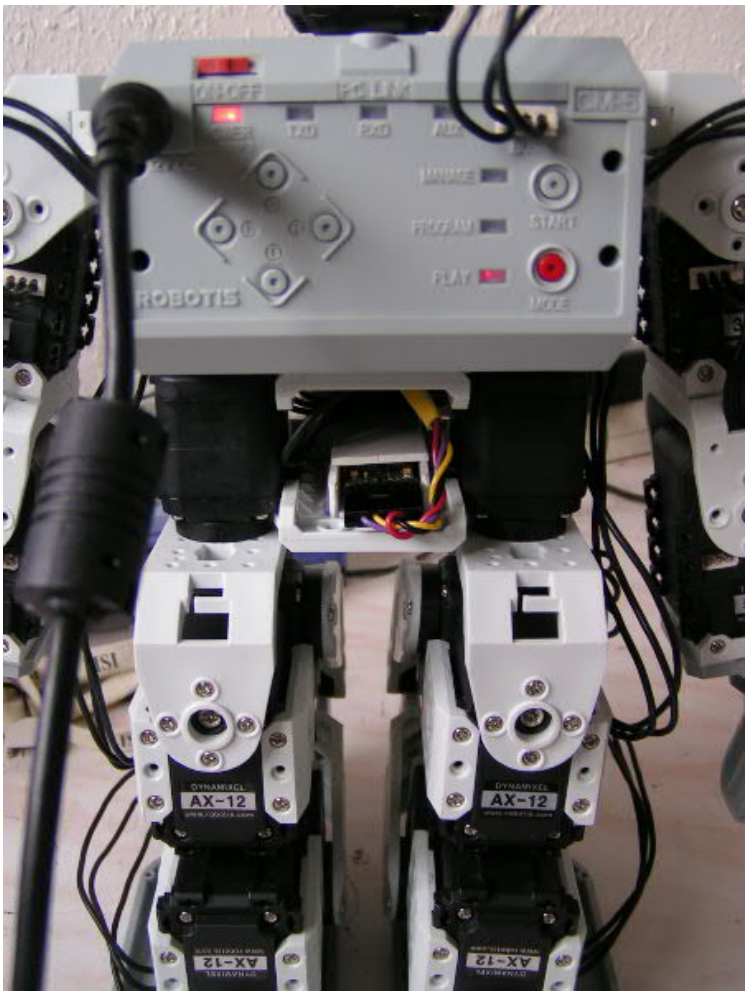
However, I am not to worried about this as I wanted this for control purposes rather than programming ones.

Step 5 – Mount the BlueTooth Module

This one is entirely up to you. I will just show you what I have done.

I have mounted my unit between the two hip rotation Dynamixels of the 18 DOF Humanoid and brought the wires down into that area. I also fitted a non-reversible plug to the wiring as I am overly cautious about plugging units in incorrectly. Watch out for the plug thing though. I added mine after I routed the wires and will now need to remove the 4 connections from the plug housing to remove my CM-5 from the robot. OOPS!!!! Only realized that as I was writing this.

I also built a small plasticard box to put my module in to protect it. That has then been held onto the robot with double sided tape.



BlueSmirf Module Mounted on Bioloid 18 DOF Humanoid

What Next

So hopefully that has enabled you to add a BlueTooth module to your Bioloid CM-5.

For me I now need to sit down and start coding some control applications. Initially this will be a Pocket PC app to act as a remote control and allow me to call motions as I wish. Then I'll think about offloading the intelligence(?) of the robot to a more powerful processor like a laptop or something.

Anyway I hope this has been helpful. **I do stress again that this is a hack that you do at your own risk**. I would hate to be the cause of anyone's robot going up in smoke but the responsibility will be yours. Good luck and feel free to contact me on the Robosavvy Forums. (www.robosavvy.com)

Regards

PEV

Perpetually Evolving Vehicle