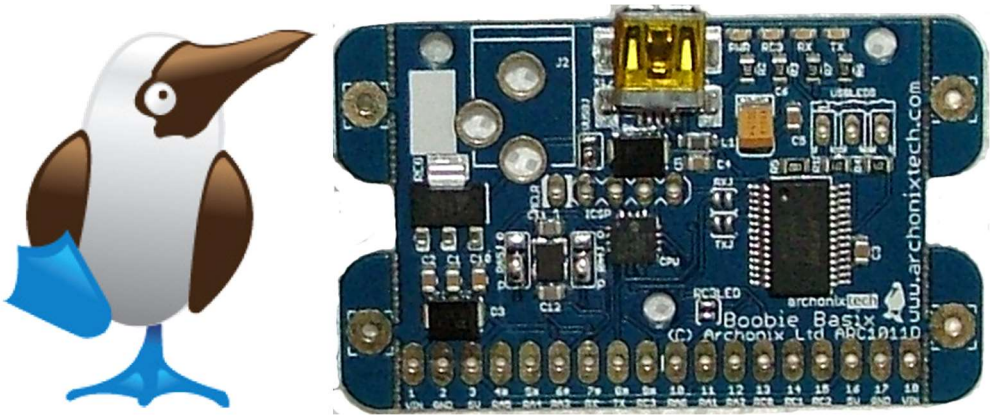
THE BACKOFFICE

Boobie Board Manual



Basix Edition

Overview:

Boobie boards are general purpose controller interfaces that are used to connect personal computers to electronic devices, leds, motors, switches and sensors using the USB interface. The best way of thinking about a Boobie board is that it is the brain of your project!

As standard Boobies come pre-programmed with a general purpose IO interface that allows you to easilly setup its ports as inputs, input with interrupts and outputs. These can be configured and operated via a simple serial terminal program such as "Hyperterminal" or "Putty" or directly via you own software applications.

There are additional features on a boobie board that make it useful for the more advanced user. They can be programmed to no require a host computer in order to complete complicated tasks, they have additional onboard hardware that can be used, e.g. 20Mhz crystal, Analogue to digital converters, interrupt timers.

Why?

Boobie board exist for various reasons including:

- They provide pre-manufactured the most critical building block for building any embedded system. System integrators and hobbiests alike no longer have to build or design this essential component.
- They are extremely durable and well made. You are using a professionally manufactured, well tested and reliable "brain" for your project.
- Basic programming is easy, you can control a Boobie board through a simple serial terminal.
- Interfacing is easy, you can hook up your boobie to your PC using USB or serial. In fact, Advanced programmers can hook Boobies together using serial!
- Advanced programming is easy, the additional hardware needed is very affordable and the PC software is easily obtainable and generally free.
- Most importantly they are extremely useful! There are lots of times when you need some electronics to perform some particular function, now all you need to learn is how to program a Boobie Board and it can be almost anything you want it to be!

Getting Started

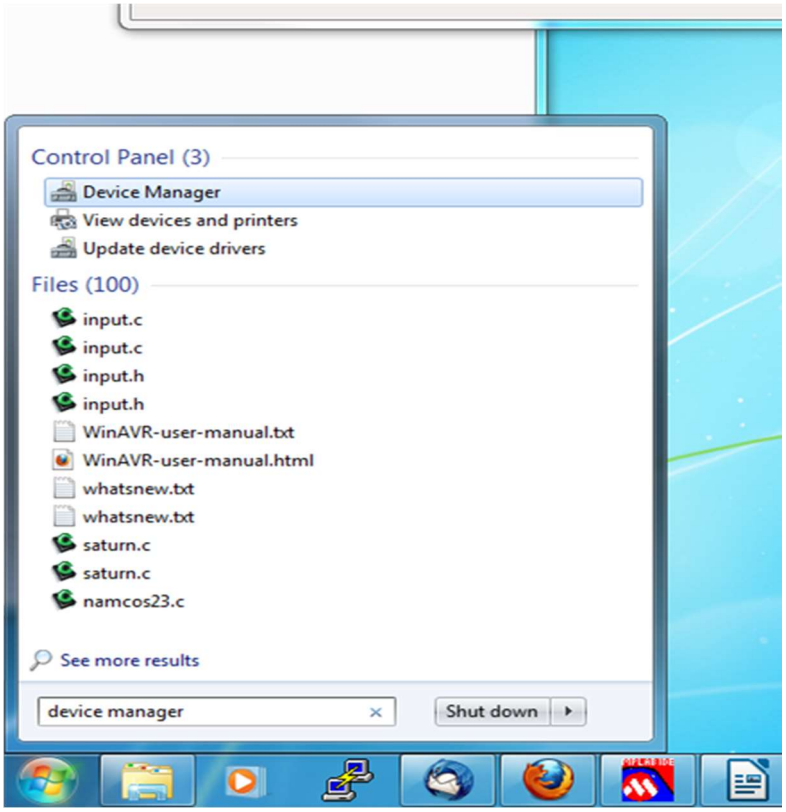
Inside the shipped packaging you will find (depending on your order of course!) the following:

- Boobie Basix Board (as shown above)
- USB Cable
- Instruction Manual
- 18 pin connectivity header or socket*
- 5 pin programming header*
- DC power jack*

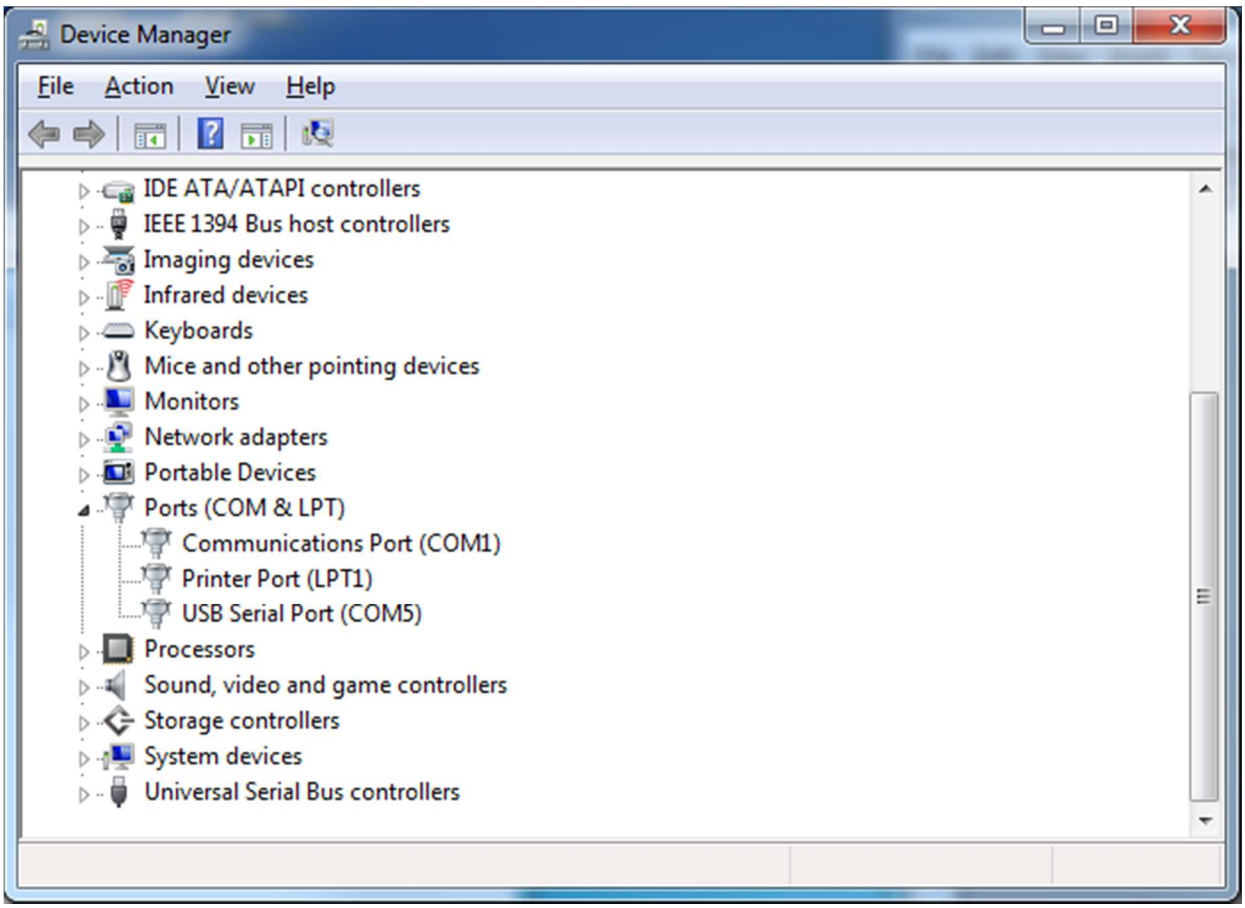
*If ordered – these may come pre-fitted if pre-fitting purchased at time of order

1. Place your Boobie board on a suitable, non-conductive (paper, wood) surface and connect the USB lead into the port on the rear of the PCB.

2. Place the other end of the USB port into your PC. Windows should automatically install the appropriate drivers. These will appear as a new COM port in device manager.
3. To check the COM port address windows has assigned the board, type in "device manager" into the search bar inside your start menu and hit return.

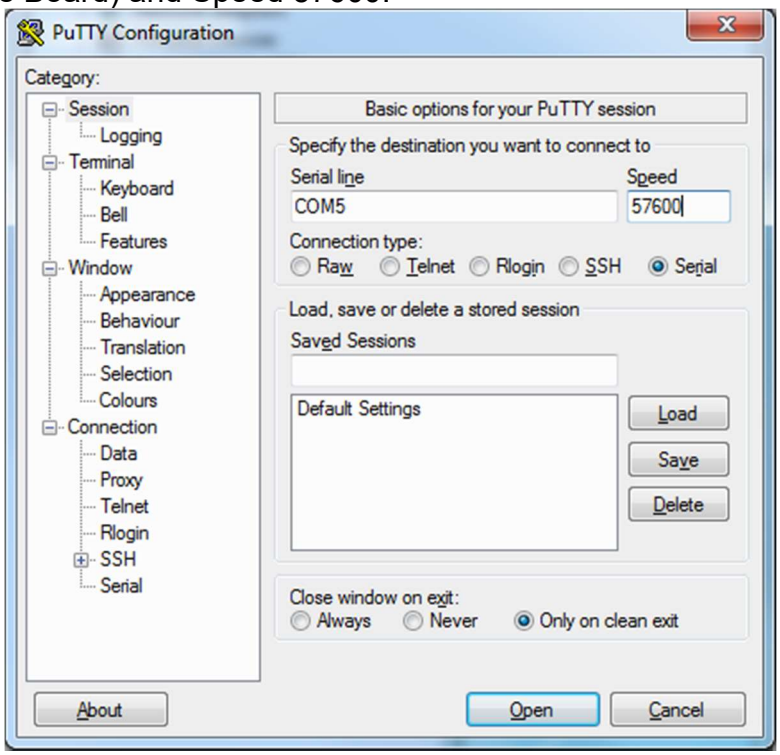


The Device Manager application should load. If you look down the list and click "ports" you will notice a new "USB Serial Port" will appear. In this case it is called COM5*:



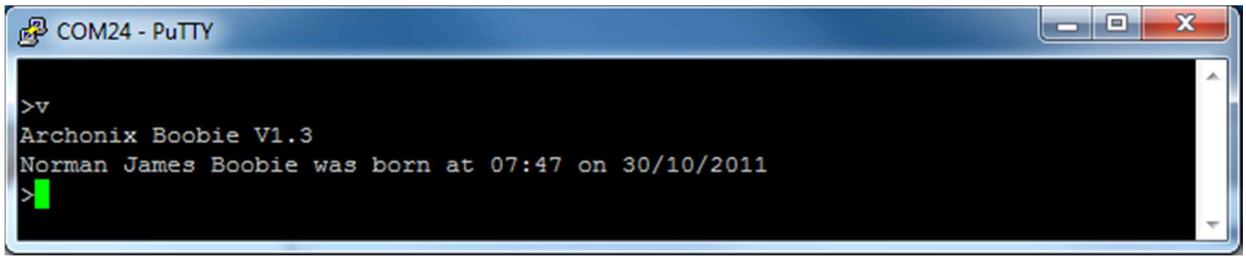
*Please note that windows can get confused if a Boobie board is removed without the user shutting down their terminal program. If this is the case windows may give your board a new COM id. It is always best to shut down any application that is using your Boobie Board prior to removing it from the USB bus.

4. Download "putty" from: http://www.chiark.greenend.org.uk/~sgtatham/putty/
5. Click on the putty executable and select Serial, COMX (where X is the COM port of your Boobie Board) and Speed 57600.



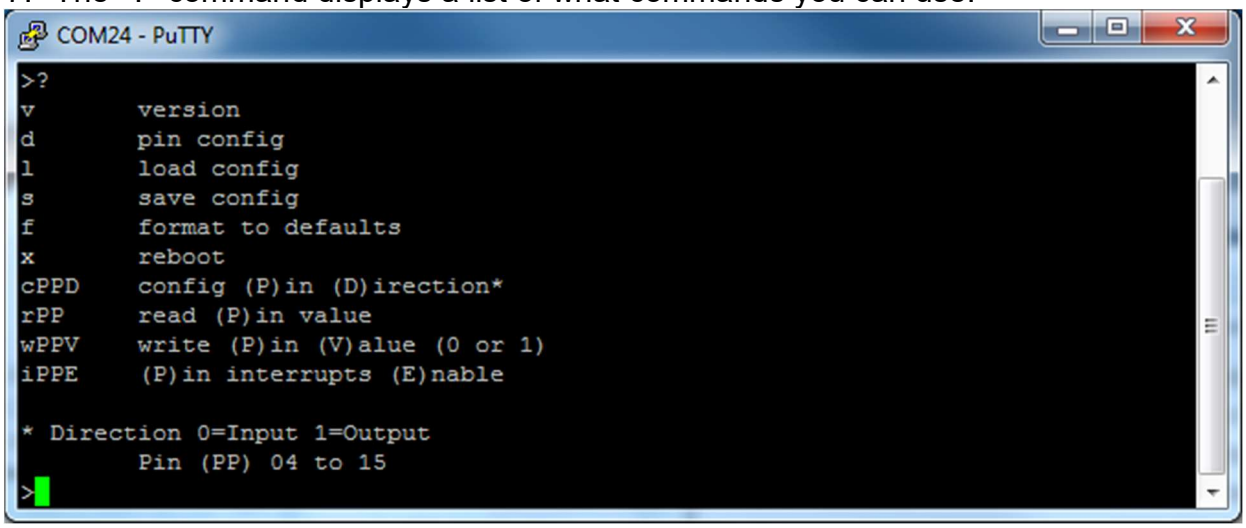
6. In the putty window type "v" on the keyboard and press the return key. You should be

presented with something similar to the following:

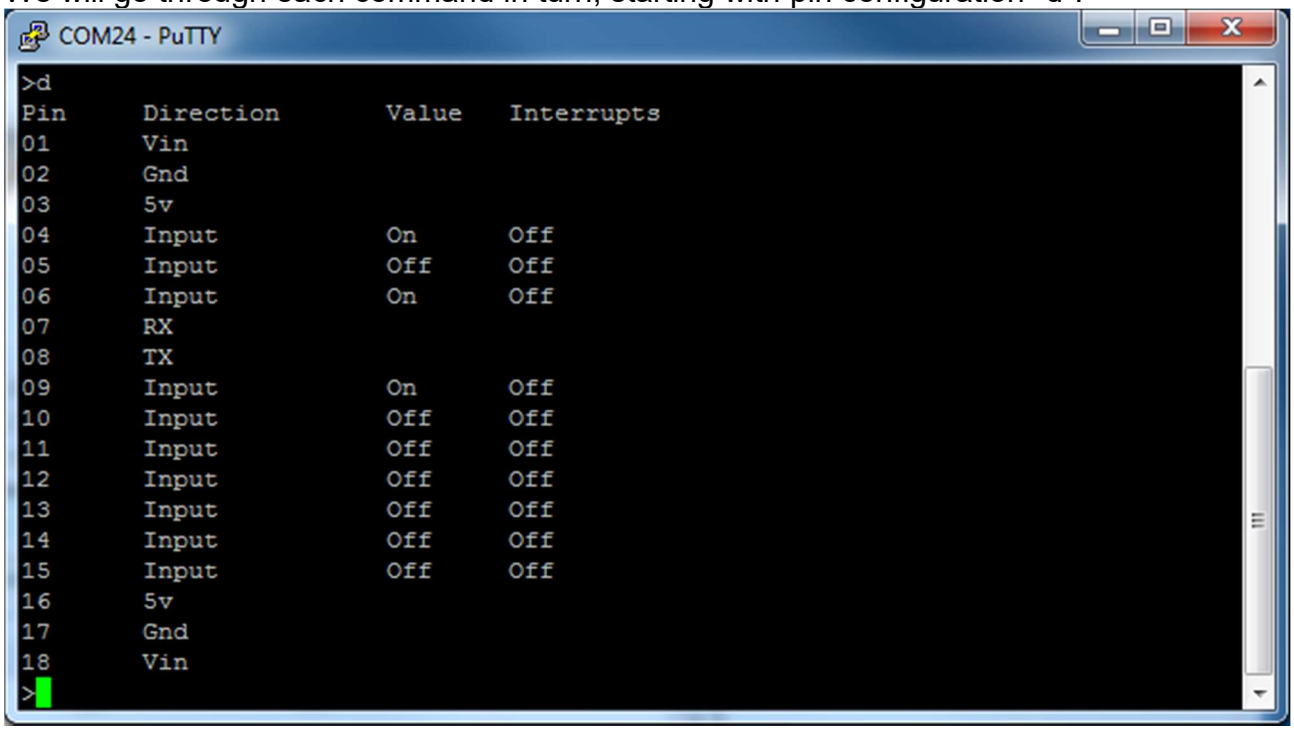


This shows that you are now connected and talking to the boobie. The "v" command displays version information. Here you can see the version of the software, the name of your board, and the time and date it was "born"!

7. The "?" command displays a list of what commands you can use:

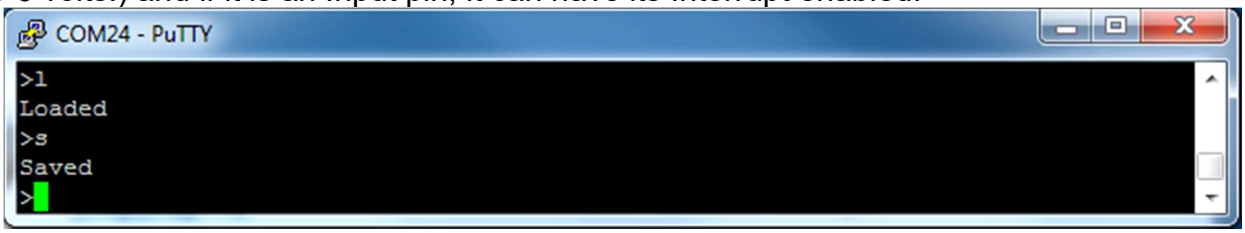


We will go through each command in turn, starting with pin configuration "d":



This shows all the pins on your Boobie Basix board, its current direction (either input or output), Its value (if it is an input "on" means it is detecting voltage on the pin, "off" means the pin is at 0 volts. If it is an output "on" means the pin is set to 5 volts, "off" means it is set

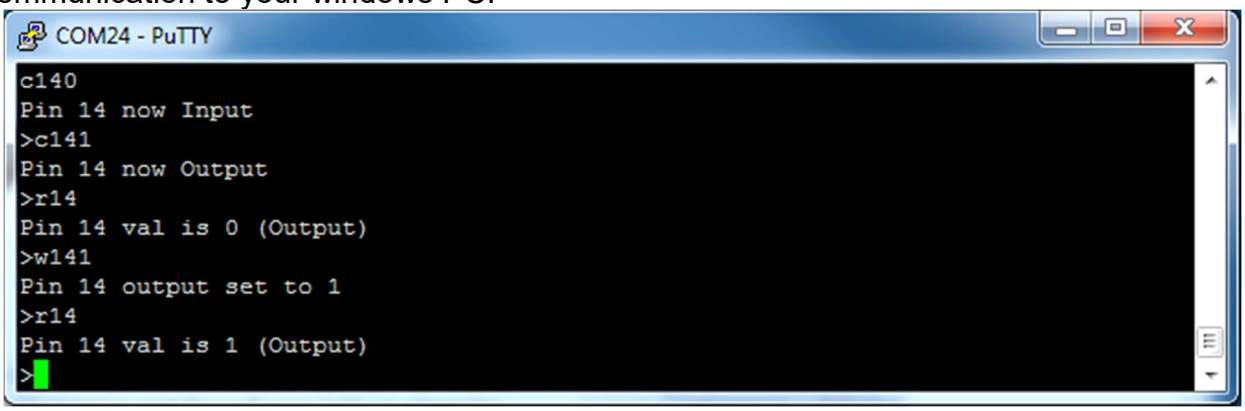
to 0 volts.) and if it is an Input pin, it can have its Interrupt enabled.



Load / Save configuration. When you load a configuration the Boobie Basix loads all port directions, values and interrupt options out of EEPROM memory and configures your board with the settings. This is done automatically at boot by the system. When you save a configuration all these settings are saved into EEPROM. Your Boobie Basix will boot up with this configuration.

Formatting the configuration sets all the ports to inputs with interrupts disabled and saves this to EEPROM.

Rebooting the system resets the board and reloads the settings from EEPROM. This is similar to removing and applying the power to the board. This does not disrupt the USB communication to your windows PC.



Here we cover several commands "c" configures the pin to either an input or an output. When you type in the pin number make sure any single digit numbers start with a leading zero e.g. For pin 4 enter 04. "r" reads the value of a pin, the value is either 1 or 0, which means 5 volts or 0 volts. If the pin is an input this shows what it is receiving, if the pin is an output this shows what the pin is outputting. "w" is used to write the output on an output pin. Writing a 1 means set the pin high – e.g. 5 volts. Writing a 0 means set the ping low – e.g. 0 volts.

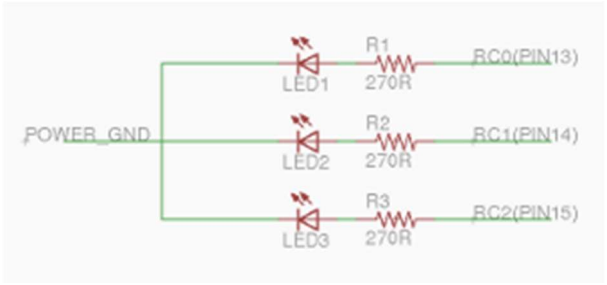
Setting an interrupt on an interrupt pin will cause an unsolicited value report to be generated when a change in pin level is detected. The number that follows the message is the system timer. This is a looping 16bit number that you can use to make high-accuracy calculations against.

PC interfacing example

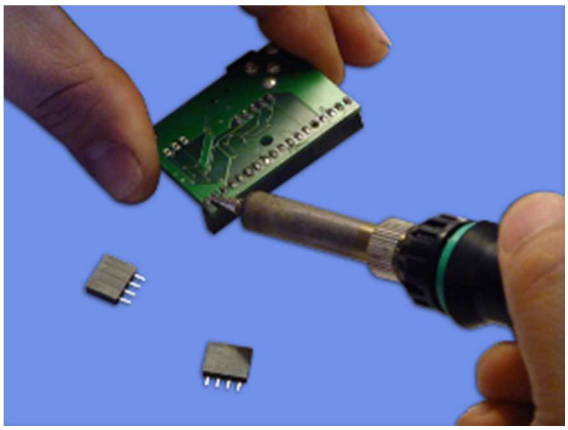
This Boobie Board twitter notifier can be made using a few off the shelf parts. All source code can be downloaded via the website or https://github.com/backofficeshow

Circuit

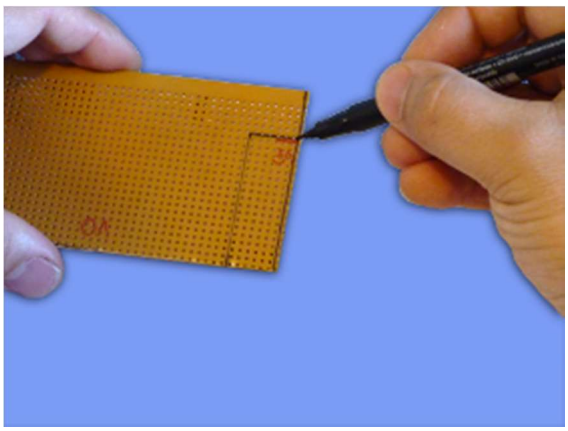
Fortunately for us, the Boobie board does all the heavy lifting. We do however need to put together a little board. Of course the on-board user controllable LED (RC3-PIN9) is factory fitted, but we want the option of several LEDs. The circuit is as follows, and basically the Boobie drives the LEDs directly via a current limiting resistor:



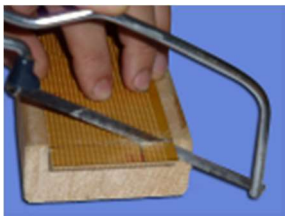
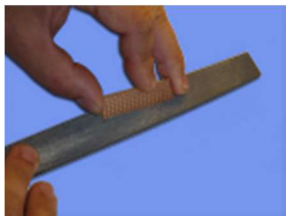
We might reuse this Boobie Board for something else, so we are soldering on pin sockets to it. If you purchased some fitted sockets these may already be fitted for you. This is an optional step, it would be more robust soldered directly using pin headers.



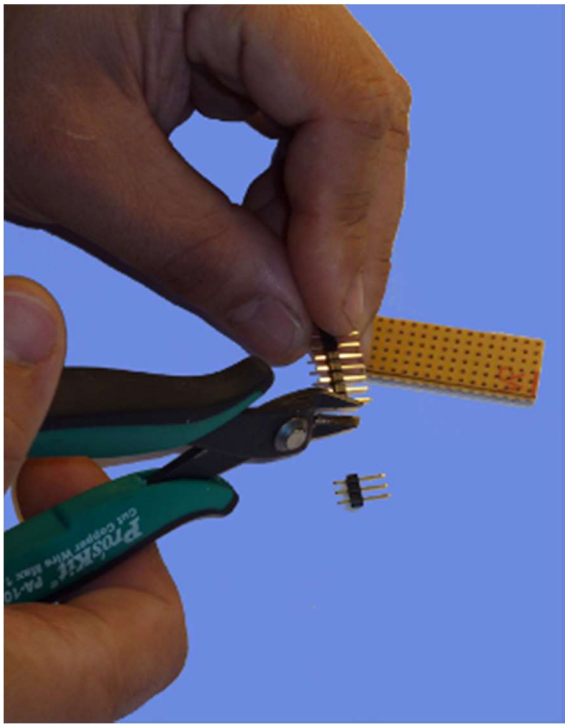
Work out how big a board you need. We want the unit to keep within the same footprint so going to keep the board quite narrow.



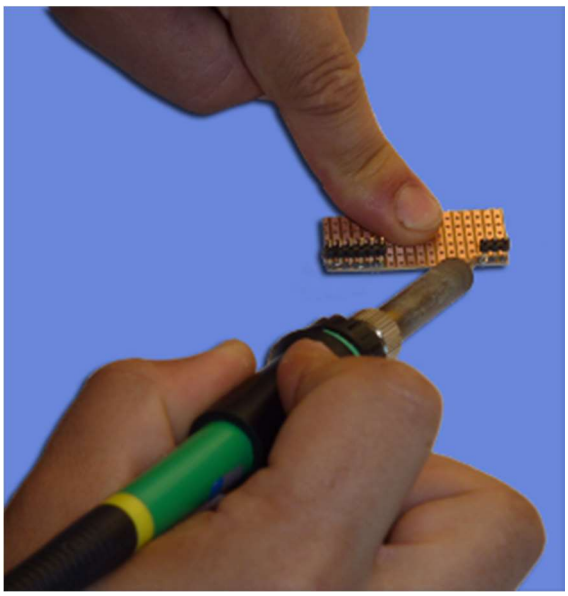
Cut out and make sure you file the edges, you want a presentable looking board!

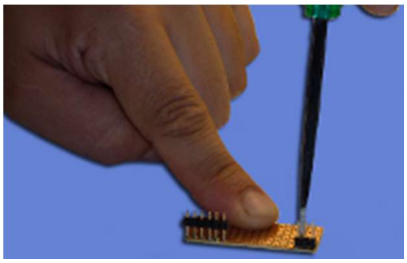



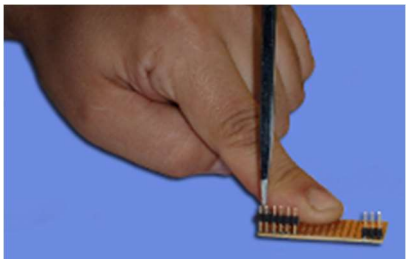
We solder on the required pin headers onto the Boobie Board, as we are only using 5 pins we dont need to fully populate the board. We added a few extra pins for stability – these are not actually connected.



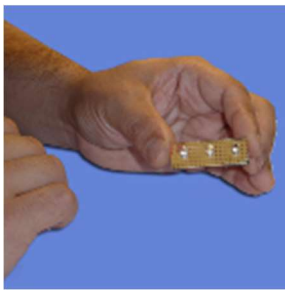
Here is a little tip, we are effectively soldering in the header onto the boobie board from the wrong side. Solder it as shown in the following image and then use a screwdriver to push the plastic stand-off down onto the board.



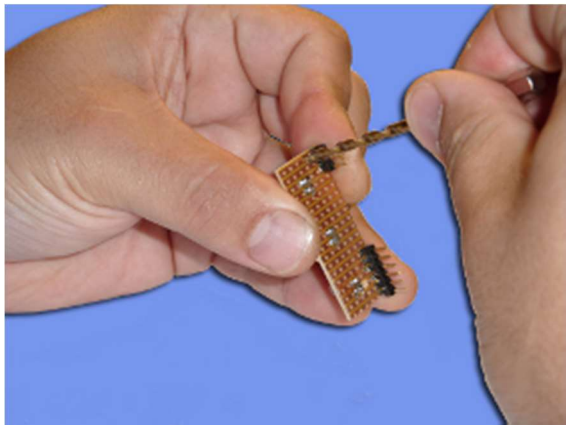




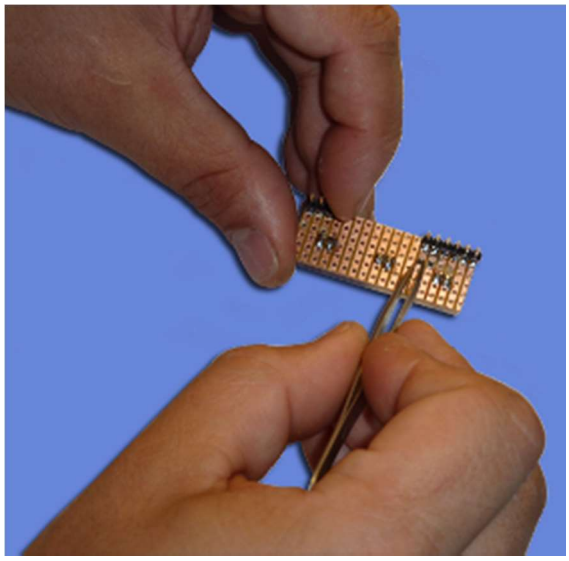
Now we add the LEDs, fortunately we have a few ultrabrite LEDs in the parts (floor sweepings) box, blue, green and white. We would have preferred red but we have to make do with what we have! Luckily enough the spacing is perfect! If you use these crystal clear ones it does not hurt to write the colour of the LED on the base as they all look the same on the bench.



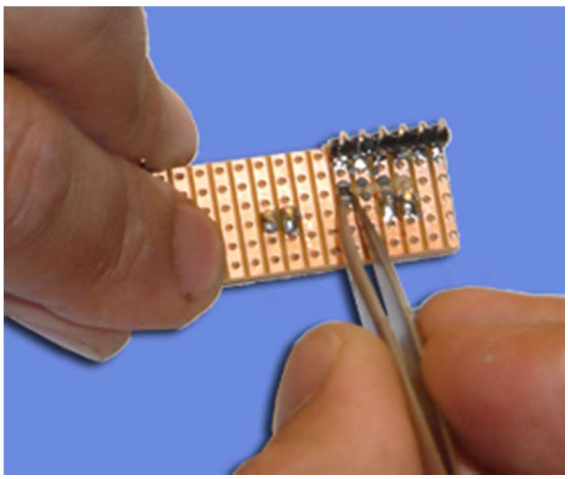

Now that the LEDs are in place we can drill out the copper strips to ensure that there are no shorts.



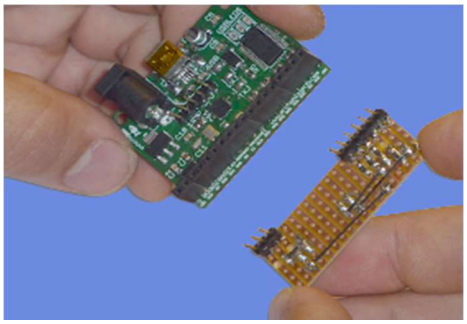
We only have surface mount resistors in the value we require (270ohm), luckily enough this happens frequently so we have developed a little technique to make it easy to use these – in fact they are easier and neater than through hole so consider it. Basically drill out the veroboard in the usual way. Put a little solder paste on either side of the hole. Solder as per usual but be careful as the resistors might try to jump before the solder cools, so keep a pin or tweezers handy to keep them in place.



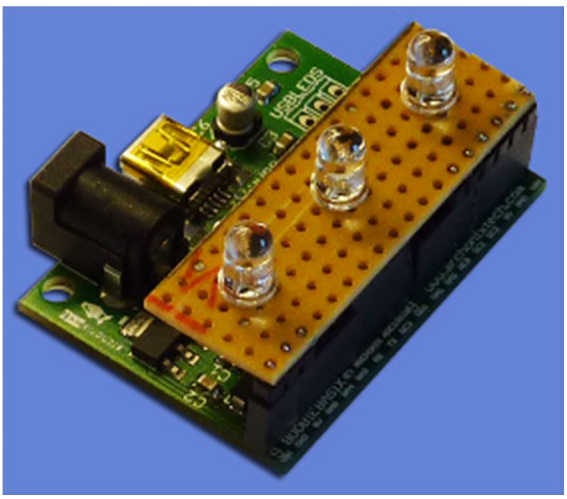
Here is a close-up of the solderpaste placement:



Seeing as we are not using through hole components on the top of the veroboard apart from the LEDs, for neatness we are using Kynar. With this we just running the power and grounds to the LEDs. You can run these any way you wish, such as using tin wire over the top of the board.

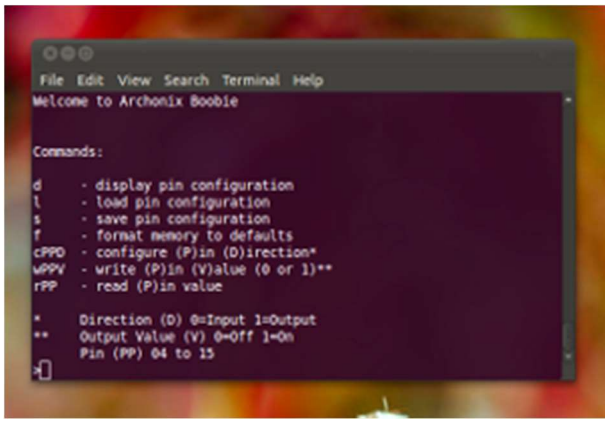


The final completed board, you have to agree, it is pretty neat! Nothing beats a professional build.



Quick test

By logging into the Boobie interface using minicom we can manually test that the LEDs are performing as expected. We can also use the opportunity to set all the port directions and initial status by setting the LEDs to the desired state and saving the settings to the Boobie's memory. This will be the configuration the board will boot into when powered up.



Software

Of course we are big Lua fans here, writing practically everything in this great language. The source code is primarily for Linux, however if anyone comes up with a version that works on a Windows machine we will be happy to host it. Lua itself is pretty portable however the system calls we have made will not be – there will be Windows equivalents. There are 3 files:

notifier.sh -> The notification script that interrogates twitter and checks for updates. Turns on an LED when there is a new tweet.

clearnotify.sh -> Run this to clear the notification LED

serialinterface.lua -> Our early version of the Boobie Lua API, alows bash scripts and WSAPI web scripts to interface to Boobie hardware.

Completed Project

We found an old tin with a plastic window in it that was just about the right size, coupled with a printed out decal we think it makes for an attractive gadget! You cannot see it in the picture, but the Boobie Board's red power LED illuminated all the white in the decal, it looks totally awesome when the light is dim.

For more examples visit the The Backoffice website at www.backofficeshow.com

Notes for Advanced Users

The microchip onboard is a Microchip PIC16F688 interfaced to an FTDI232R USB controller.

The onboard crystal is 20Mhz and in default mode disconnected. The PIC16F688 has an internal 8Mhz oscillator. The onlboard regulator is capable of 500mA current. By default USB power is connected to the 5V line directly, but can be switched through the regulator with solder jumper VUSBJ.

ICSP Pinout

1. MCLR
2. ICSPCLK
3. VDD (5V)
4. VSS (GND)
5. ICSPDAT

Boobie Pinout

1. VIN - Unregulated Voltage Rail (from DC Jack) VUSBJ controls if USB power goes via the regulator. By default it does not.
2. GND
3. 5V - Regulated 5V rail 500mA supply
4. RA5 - Disabled when Oscillator Enabled (RAJ5 set to O)
5. RA4 - Disabled when Oscillator Enabled (RAJ4 set to O)
6. RA3 - Input only as also MCLR
7. RX - Connected to FTDI USB chip, can be disconnected by cutting link RXJ
8. TX - Connected to FTDI USB chip, can be disconnected by cutting link TXJ
9. RC3 - Connected to onboard LED, can be disconnected by cutting link RC3LED
10. RA0
11. RA1
12. RA2
13. RC0
14. RC1
15. RC2
16. 5V
17. GND
18. VIN

USBLED port

1. TXLED - 270R Series resistor already connected
2. RXLED - 270R Series resistor already conencted
3. 5V - Regulated supply to the LEDs

www.backofficeshow.com