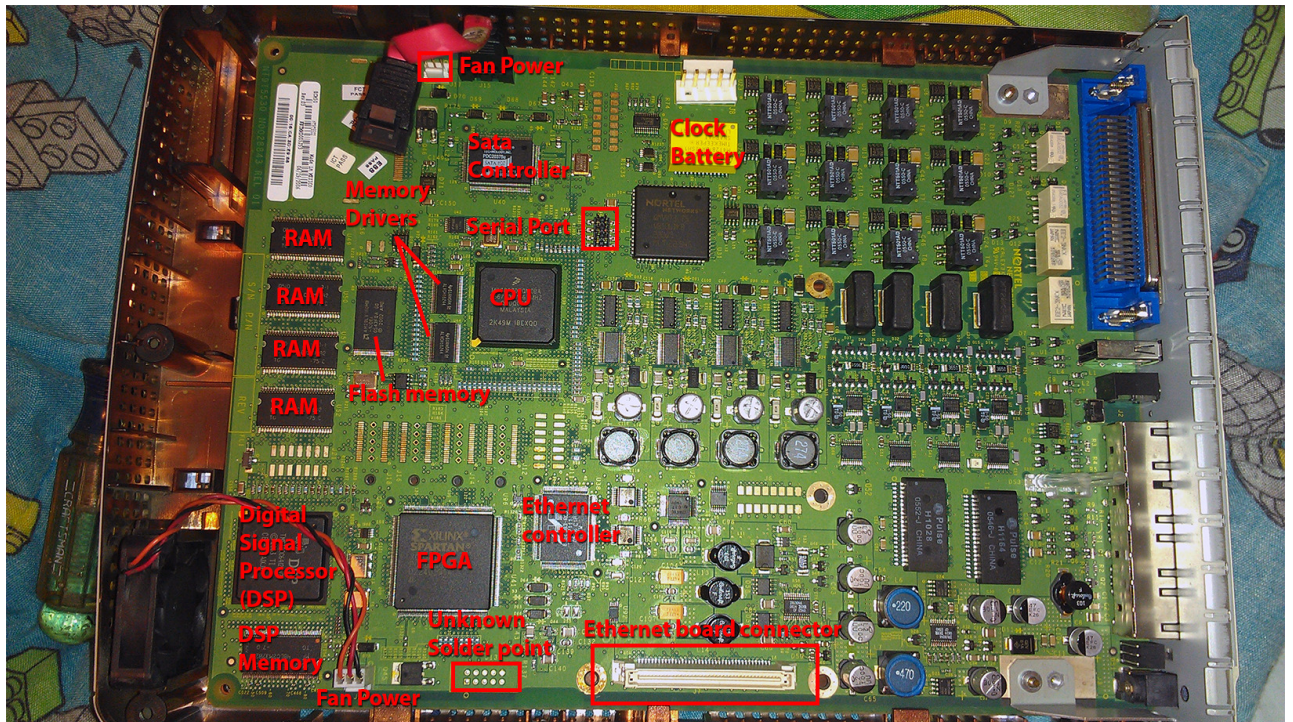


## BCM 50 Serial Port and Motherboard components layout

This guide also explains how to install a serial port to USB cable at 115200 baud onto the BCM 50.



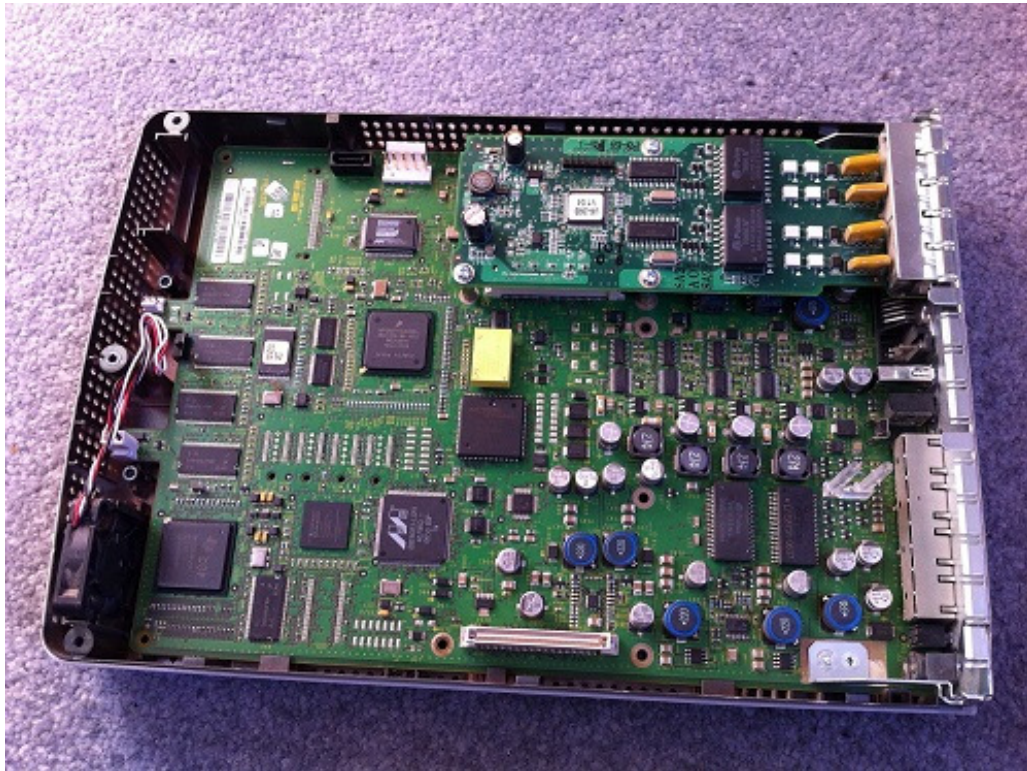
The picture above shows you where the main components are located on a standard BCM 50 system. The Ethernet Board connector is used for the BCM 50 “a” and “e” versions as it supports another board that is attached. It also uses some of the mounting posts that are screwed onto the main Motherboard.

An example of a BCM 50a is shown below with the ADSL 1 card plugged in at the bottom

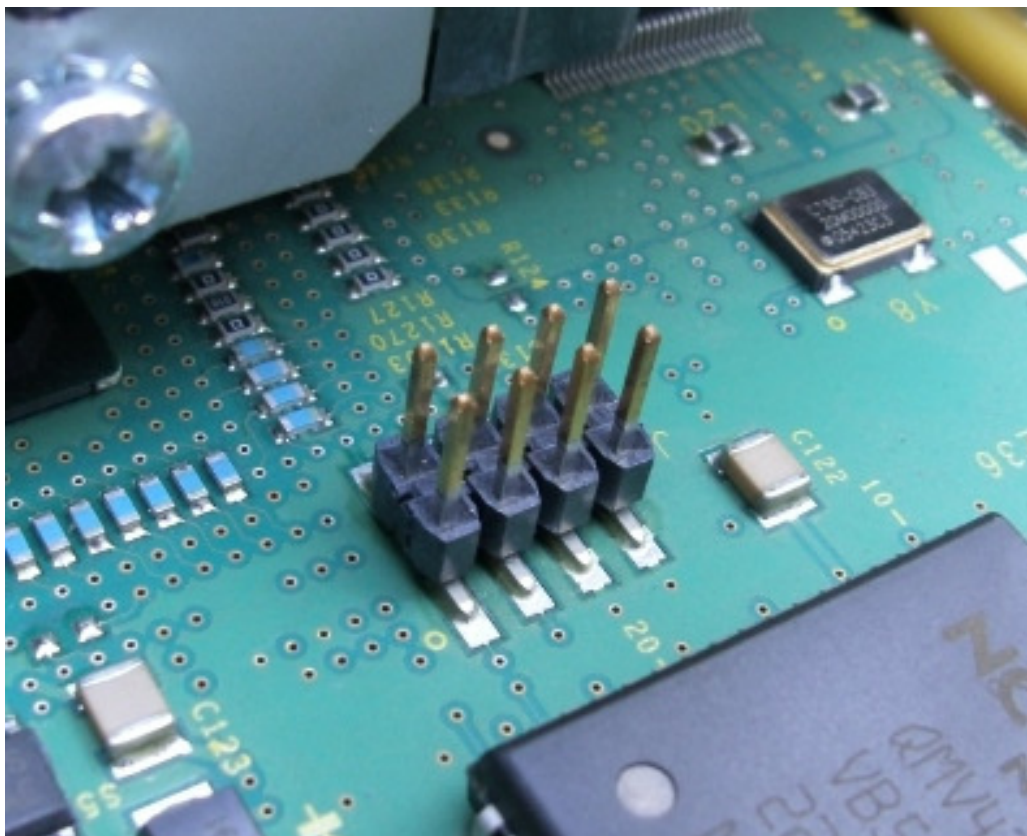




The BCM 50 "b" has a different (Type 2) Motherboard and it uses a connector that exists near to the top because the board that sits over the serial port pins has two BRI ports that go over the 25 pair Viking connector. An example of a BCM 50b is shown below



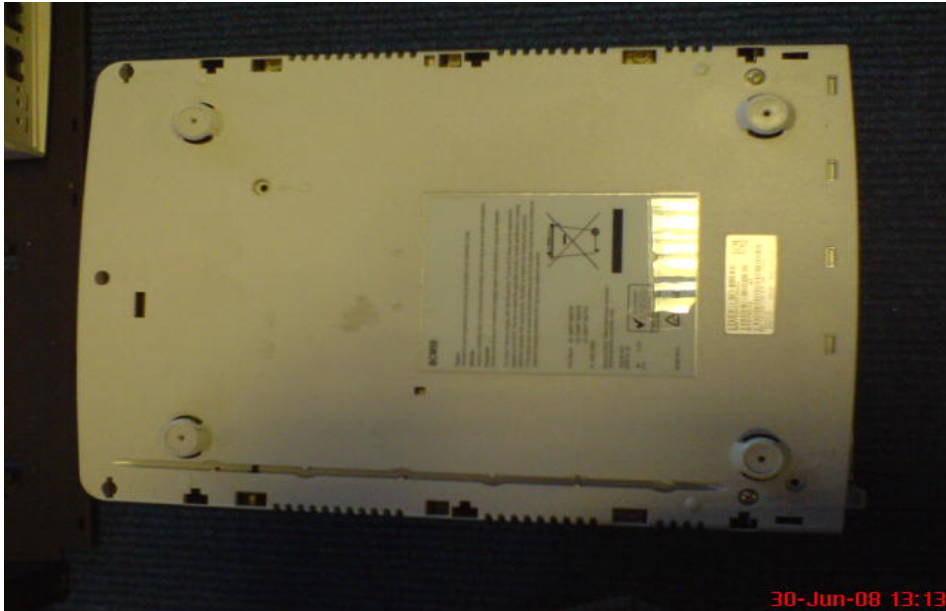
A close up of the serial port on the BCM 50 (Type 1) Motherboard is shown below



## How to access the Motherboard or Hard Disk on a BCM 50 system

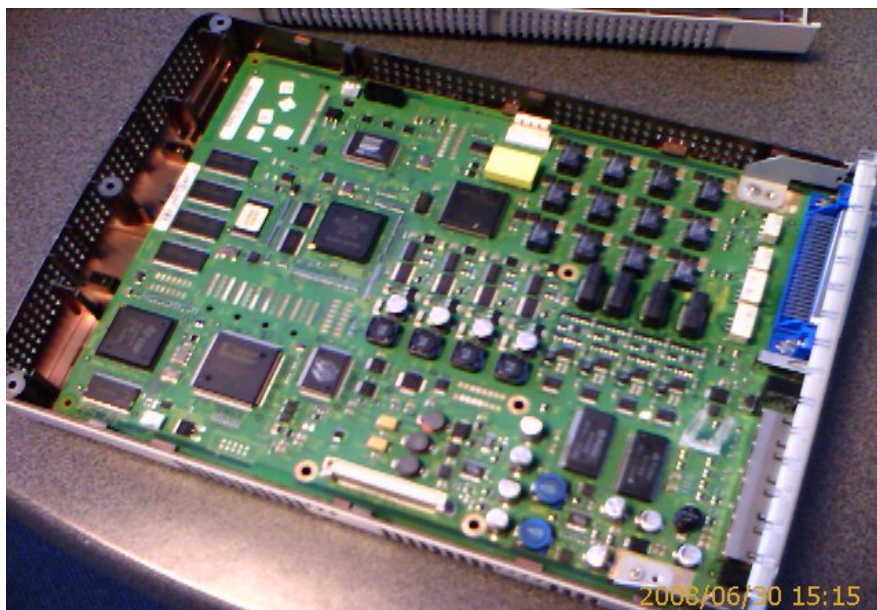
The details listed below are intended to guide you through the process of accessing the Motherboard or Hard disk to connect up a serial port cable.

The best way to open up the BCM 50, is to unplug (mark all of the cables first!) the Ethernet cables etc. Turn over the unit to view the 5 screws on the back. There are 3 at the top & 2 at the bottom.



Another useful thing to note is that at the bottom of the base is a Nortel or Avaya label that lists out the systems site ID & MAC address. This is very important as you will need either of these to obtain the keycode lic file for that site ID if it is missing.

Remove the top 3 screws, marked with an A & an arrow pointing upwards!. Turn the unit to slide the top cover towards the end with the Nortel symbol on. The picture below shows you what it looks like inside. In this shot, I have already removed the SATA hard disk which sits on the left hand side.





## How to configure and install the BCM 50 Serial Port to USB cable

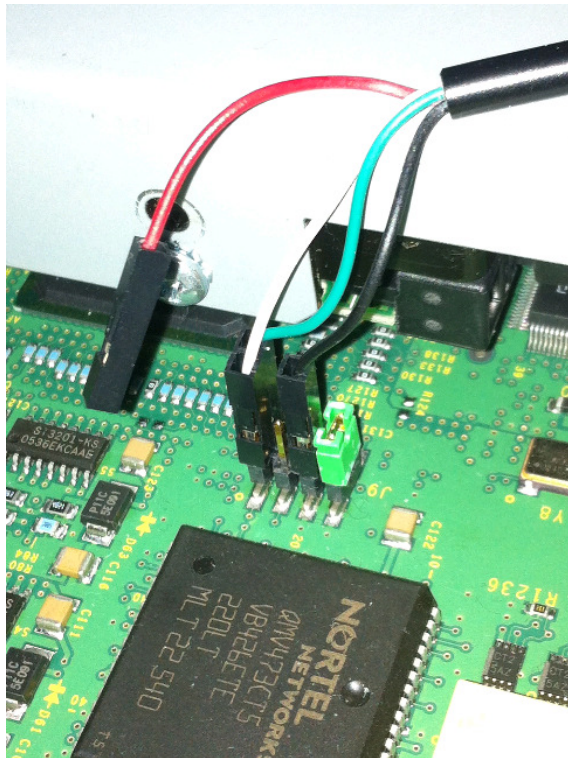
You can use a serial port cable for a BCM 50 system and this is useful to try and fix faults in a similar way to the BCM 400 and 450 Linux versions. Access onto the Linux programming is also possible via accounts such as nnsupport etc. An example of a BCM 50 connected to a laptop via a modified Serial to USB cable is shown below using serial port programs such as Procomm Plus or Hyper Terminal etc.



The serial port connector on a BCM 50b is located below the BRI module and you will probably need to remove the board after powering off the BCM first to insert the serial cable and then replace the module, before powering it up again as shown below with the BRI module attached. Be aware that the **bridging pins are on the right** towards the hard disk power connector.



A close up of the serial cable and coloured wires is shown here attached onto a standard BCM 50. Please be aware that only three wires are needed and the Red 3.3 volts lead isn't used!. Also a green IDE jumper was used to bridge the pins needed for the u-boot console to work.



On the BCM 50b Motherboard, the pins are mounted with the green IDE jumper used for the u-boot console positioned at the top end towards the hard disk & power connector as shown below. The BCM 50b in the photo below uses a Solid State Drive and the full details are an [Intel X25-M SSDSA2M080G2GC 80GB 2.5" MLC SATA SSD](#). This was housed in a [red 1Pc 2.5 to 3.5 adaptor Bracket SSD HDD Notebook Mounting Holder Tray Caddy Bay](#). The only moving part left is the fan!.





The pinout for the BCM50 serial port is done using a USB to 3.3 volt TTL serial converter lead. They can be found online (eBay) for a few dollars / Pounds. The full details are listed below.

<http://www.ebay.co.uk/itm/USB-to-TTL-serial-Cable-UART-Converter-PL2303HX-1-0M-For-Arduino-UNO-mini-Black-/181808575770?hash=item2a54a2b51a>



### Description

This is a new USB to TTL Serial Cable chipset PL2303HX 1.0M for Arduino UNO mini. The TTL-PL2303HX cable is a family USB to TTL serial UART converter cables which incorporate FTD's PL2303HXQ USB to Serial UART interface IC device which handles all the USB signalling and protocols. The cables provide a fast, simple way to connect devices with a TTL level serial interface to USB. The cables are available and has a 6-way SIL, a 0.1" connector, a 3.5mm Audio Jack, an 8 way, keyed 2mm pitch connector (intended for use with VMUSIC2 or VDRIVE2) or bare and tinned wire ended connections. Each cable is 1.0 m long and supports a data transfer rate up to 3 MB?

### Features:

100% Brand New and High Quality with compact portable design.

FTDI based USB to TTL Serial Cable are designed using the standard PL2303HX chipset.

The cables provide a fast, simple way to connect devices with a TTL level serial interface to USB.

RX and TX LEDs indicator.

The cables are available with a 6-way SIL, 0.1" connector.

TTL level is 3.3V.

The TTL-PL2303HX cables are a family of USB to TTL serial UART converter cables incorporating FTD's PL2303HXQ USB to Serial.

UART interface IC device which handles all the USB signalling and protocols.

Each TTL-PL2303HX cable contains a small internal electronic circuit board, utilizing the PL2303HX which is encapsulated into the USB connector end of the cable. The other end of the cable comes with a selection of different connectors supporting various applications.

Cables are FCC, CE, and RoHS compliant and are available.

### Specification:

TTL level is 3.3V

Size: The length of the cable is 1.0m

Colour: Black

Package weight: 1.0 Oz/29g

### Compatible Mode:

The USB side of the cable is USB powered and USB 2.0 full speed compatible. The device will work on most computers that are Windows and Linux etc, however quite a number of the PL2303 USB adapters don't seem to work on Windows 8 onwards although the 64bit PC's tend to be unaffected. After referring to the Prolific website, I was told that counterfeit were being produced in China and the drivers won't work as a "code 10" error is displayed in Windows Device Manager.

After doing some more research, I found a link on the web to download the older drivers and after installing them I was able to roll back the latest driver to the one that I had found and I was then able to connect OK. I'm not sure if this link went to the correct driver that worked?

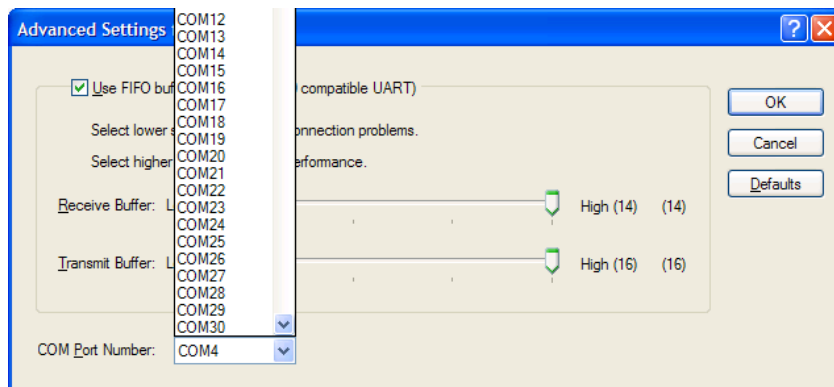
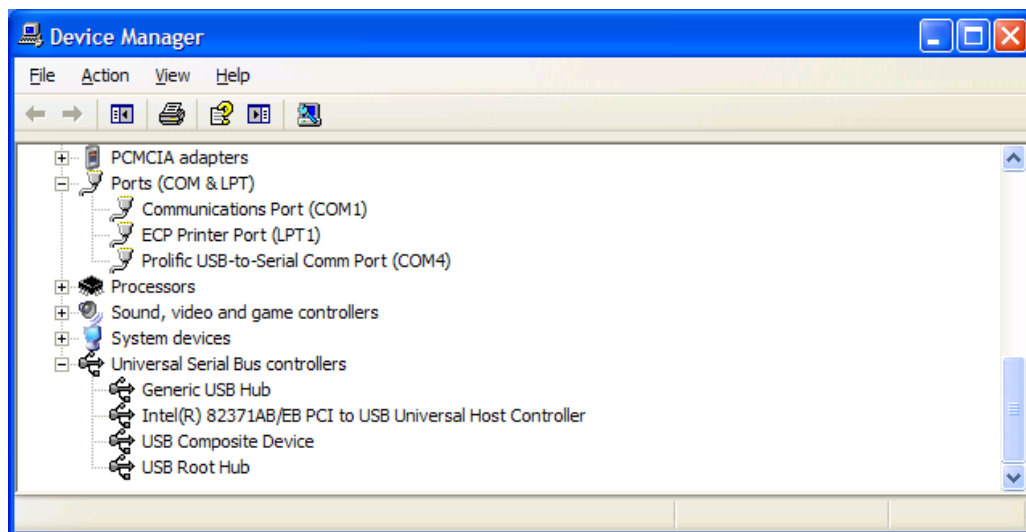
[http://www.prolific.com.tw/UserFiles/files/PL2303\\_Prolific\\_DriverInstaller\\_v1\\_12\\_0.zip](http://www.prolific.com.tw/UserFiles/files/PL2303_Prolific_DriverInstaller_v1_12_0.zip)

The actual file I found was called "[PL2303\\_Prolific\\_GPS\\_1013\\_20090319.exe](#)"

## Important note before connecting up the wires onto the BCM 50 system for the first time

The first thing to do is to plug in the USB cable into the computer that you want to link up with the BCM 50 and make sure that you can access the internet before connecting up the lead. By doing this, the USB Drivers are loaded up and therefore it will assign a USB Port that you can check on after it is done as some communication tools such as Procomm Plus will only be able to use Communication ports 1 to 4.

You might need to go into the device settings area to change it. Doing this now, will reduce the troubleshooting time later on when you connect to the BCM 50. In my case, I was using the Prolific USB-to-Serial Comm Port 4 (COM4) as shown below on my Windows computer. Go into the advance Settings section to change it by right clicking it if required.

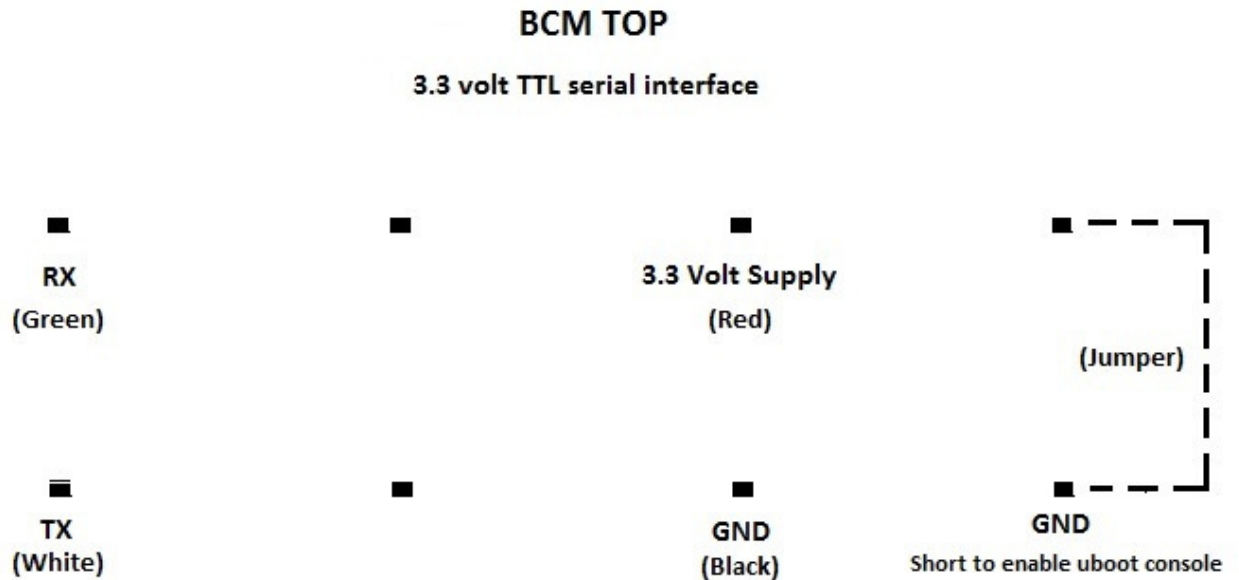


Take care when connecting up the cables and follow the chart below. Some of the USB to Serial Port adapters have been supplied with different colours.

It would be worthwhile at this stage to obtain a volt meter so that the 3.3 volt **+Vcc** wire can be found. In my example, it was the **Red** wire.

Because some of the USB adaptors don't follow any standard colour code, then you should **under no circumstances**, plug in the **+Vcc (Red)** wire from the USB to TTL adaptor into any of the 8 pins on the Motherboard, except where mentioned below in the diagram on the next page, although it can just be left unplugged as shown in my original close up photo a few pages earlier as it isn't used.

Please be aware that “**BCM TOP**” below refers on the standard BCM 50 Motherboard as the top end where the hard disk bay is positioned and that the “**Jumper**” end is towards the hard disk power connector on the right hand side. For the BCM 50b, it’s 90° anticlockwise and the Jumper end still points towards the hard disk power connector.



The (Jumper) pins on the far right only need to be shorted to access the U-Boot console before the Linux operating system is loaded. They can remain shorted and you can still access the Linux console. If you have them joined, then you can watch the entire BCM 50 boot process all the way from the beginning of the bootloader until Linux has finished loading and allows you get to the nnsupport login screen.

I used a spare IDE connector that I had, to link up the u-boot console. You can use other leads instead to create a short to join them together.

Once the cable has been made up, check that the USB plug is plugged into your computer. Use a communications software package such as [Hyper Terminal](#) , [Procomm Plus 4.8](#) or [ZOC](#) for Windows and [PuTTY](#) is available for Windows and Linux etc. Check that you are on the correct communications port and that you have **115200 baud** set. You should have the **cable plugged in** and ready before you power up the BCM 50 system!. Standard 8 data bits, No Parity and 1 stop bit settings were used with VT100 set, but I you can use other parameters.

You can make a more robust lead onto the BCM 50 by obtaining extra connectors to make an 8 pin connector and by sticking them all together with very strong glue to make it easier to plug in & out.

If after turning on the BCM system, you have a blank screen, try and swap around the leads at the BCM end, but being careful not to pull the wires out of the connector. It doesn’t matter if the cable is plugged in the wrong way as long as the 3.3 volt lead is disconnected. It might be easier to mark up one side of the connector so that you know which way it plugs in onto the Motherboard.

Another useful point is that the cable will also work with older USB 1.0 computers.



Once the BCM Linux operating system has loaded, you get the same interface on the serial port as you do over putty! This could be very useful if the Ethernet somehow gets messed up as happened to one of my BCM's recently.

Having access to the serial port enables you to carry out many tests that aren't available via the Ethernet Port. A number of BCM tests can be carried out to check that the hardware is OK. Also Linux access is available if the BCM boots up OK, when the Ethernet ports aren't accessible. A list of some of the main useful commands that can be done by keying in **help** at the => prompt after you stop the auto boot within 5 seconds by pressing any key are shown on the next page.

A typical working BCM 50 version 2.0 boot up sequence is shown below

CSC U-Boot 2.04.01 (Jul 12 2005 - 18:25:35)

MPC8260 Reset Status: External Hard, External Soft

MPC8260 Clock Configuration

- Bus-to-Core Mult 4x, VCO Div 2, 60x Bus Freq 25-75 , Core Freq 100-300
- dfbrg 0, corecnf 0x1a, busdf 5, cpmdf 1, plldf 0, pllmf 5
- vco\_out 399960000, scc\_clk 99990000, brg\_clk 99990000
- cpu\_clk 266640000, cpm\_clk 199980000, bus\_clk 66660000

CPU: MPC8260 (HiP7 Rev 00, Mask unknown [immr=0x0a10,k=0x0071]) at 266.640 MHz

Board: BCM CSC

I2C: ready

DRAM: Total size: 256 MB

Memory cleared

Testing SDRAM ...

sdram data path test passed

macho address test passed

walking ones test passed

Now running in RAM - U-Boot at: 0x0ff55000 .data 0x0ffb35b0 .bss 0x0ffc1000

FLASH: 2 MB

Reboot Reason: Power Recycle

Program FPGA

Image Name: FPGA csc\_200 rel 20

Created: 2005-07-12 22:25:26 UTC

Image Type: PowerPC FPGA Firmware (gzip compressed)

Data Size: 46374 Bytes = 45.3 kB

Load Address: 00010000

Entry Point: 00000000

Verifying Checksum ... OK

Uncompressing from ffe70040 fpga Firmware ... OK

Configuring The FPGA Device .....

.....wrote 130952 bytes...

test passed

PCI INIT

00 16 105a 3375 0180 0e

sub\_bus:0

```
DEVICE INIT
JUMPTABLE INIT
CONSOLE INIT R
In: serial
Out: serial
Err: serial
enable Interrupts init
get environmnet information
Net: FCC1 ETHERNET [PRIME], FCC2 ETHERNET, FCC3 ETHERNET
IDE: Bus 0: OK Bus 1: not available Status = 0x7F
  Device 0: Model: WDC WD2500JS-55MHB1 Firm: 10.02E01 Ser#: WD-WCANKH356025
    Type: Hard Disk
    Capacity: 131071.9 MB = 127.9 GB (268435455 x 512)
Powersaving FF
Powersaving FF
Powersaving FF
BEDBUG:ready
dbg_mode_l
Hit any key to stop autoboot: 0 ##### (Press any key within 5 seconds!)
```

### Useful commands to try out

```
=> help          ##### (Prompt to print out the commands. Also try e.g. help date etc)
=> base          ##### (Print out the BCM Site ID or MAC address against ethaddr)
=> csc_sata      ##### (Command to check out the hard disk interface is in an active state)
=> date          ##### (Print out the date and time from the Motherboard)
=> printenv      ##### (Print out the boot up status and main Avaya ID against SerialESS)
=> test all      ##### (Command to look for individual tests to try out from 1 to 100)
=> test 10 to 13 ##### (Test that the LED's on the BCM 50 work correctly)
=> test 40       ##### (Identify the hard disk and other details. Also use ide info)
=> test 61       ##### (Ethernet Test, external loopback tests)
=> boot          ##### (Boots up the BCM without turning the power off)
=> reset         ##### (Resets the BCM as a reboot without pulling out the power lead)
```

### Troubleshooting tips

The following examples were solved on BCM 50 systems that would have been thrown away as being hardware errors.

A number of faults and tests can also be looked at on the following BCM systems below

The BCM 200, 400, 1000 version 4.0 (Linux) systems has a 9 way RS232 serial port hidden behind a panel on the front on the left which can be used at 9600 baud by removing the two screws. You might need to power down the BCM before you remove it as this panel also covers the monitor, keyboard and mouse connections which can be used to get to the BIOS during the power up process.

The BCM 450 system (All versions) has a 9 way RS232 serial port connector on the front to the right of the LAN Ethernet ports and you need to have it as 115200 baud.

You might also require a null modem adapter and use standard settings such as 8 1 None etc.



## Problem 1

All four Ethernet ports not working and the Green LED's are constant all the time. This is due to a corrupted configuration and the symptoms are that the FPGA boot up sequence is missing. The example below was taken from a BCM 50 at release 6.0.

```
CSC U-Boot 2.06.6 (Aug 17 2009 - 17:07:17)
MPC8260 Reset Status: External Hard, External Soft
CPU: MPC8260 (HiP7 Rev 00, Mask unknown [immr=0x0a10,k=0x0071]) at 266.640 MHz
Board: BCM CSC Rel 1 Rev 0
I2C: ready
DRAM: Total size: 256 MB
Memory cleared
Testing SDRAM ...
sdram data path test passed
macho address test passed
walking ones test passed
Reboot Reason: Power Recycle
Now running in RAM - U-Boot at: 0x0ff52000 .data 0x0ffb5650 .bss 0x0ffc2000
FLASH: 2 MB
PCI INIT
    00 16 105a 3375 0180 0e
sub_bus:0
DEVICE INIT
JUMPTABLE INIT
CONSOLE INIT R
In: serial
Out: serial
Err: serial
enable Interrupts init
get environmnet information
Net: FCC1 ETHERNET [PRIME], FCC2 ETHERNET, FCC3 ETHERNET
IDE: Bus 0: OK Bus 1: not available Status = 0x7F
    Device 0: Model: Maxtor 6N040T0 Firm: NAN51680 Ser#: N10DXKDG
        Type: Hard Disk
        Capacity: 39205.6 MB = 38.2 GB (80293248 x 512)
BEDBUG:ready
dbg_mode_l
Hit any key to stop autoboot: 0 ##### (Make sure that you press a key!) #####
```

## Solution

Carry out the following tests using a serial port lead at the => prompt to see if it is corruption.

```
=> fpgactrl enable      (You can either use disable or enable!. It should be set to enable)
=> printenv           (Look for the line of fpga_config. It should be fpga_config=y)
=> setenv fpga_config y (Use this command if after using printenv, it was set to fpga_config=n)
=> saveenv            (This saves the environment settings upon a reboot!)
=> reset              (Resets the BCM as a reboot without pulling out the power lead)
```

If it works, then you will now see the missing lines of FPGA Device data during the initial power up. The 4 Green LED's for the Ethernet ports will go out when you see the Linux boot up data appear.

## Problem 2

BCM 50 system that has both flashing Red LED's after powering it up and they stay constant.

## Solution

This was due to an upgrade failure. By having a serial lead connected, you can watch the BCM boot up and see where the last command line was.

Opening the console

INIT: version 2.84 booting

Welcome to NCGL NCGL 2005 (cjdavola)

Press 'l' to enter interactive startup.

Mounting proc filesystem: [ OK ]

Setting hostname bcm\_m50r1: [ OK ]

Activating swap partitions: [ OK ]

Finding module dependencies: [ OK ]

Mounting local filesystems: [ OK ]

Enabling swap space: [ OK ]

INIT: Entering runlevel: 3

Entering non-interactive startup

Starting nn\_modules: [ OK ]

Starting fesConfig: [ OK ]

Starting Fans\_control: [ OK ]

Setting network parameters: [ OK ]

Bringing up loopback interface: [ OK ]

Bringing up interface eth0: [ OK ]

Bringing up interface eth2: [ OK ]

Starting portmapper: [ OK ]

Starting system logger: [ OK ]

Starting kernel logger: [ OK ]

Initializing random number generator: [ OK ]

Starting xinetd: [ OK ]

**Starting ResetButton: ##### (Press the Reset Button on the front panel 2 times) #####**

CSC U-Boot 2.04.01 (Jul 12 2005 - 18:25:35)

MPC8260 Reset Status: External Hard, External Soft

MPC8260 Clock Configuration

- Bus-to-Core Mult 4x, VCO Div 2, 60x Bus Freq 25-75 , Core Freq 100-300
- dfbrg 0, corecnf 0x1a, busdf 5, cpmdf 1, plldf 0, pllmf 5
- vco\_out 399960000, scc\_clk 99990000, brg\_clk 99990000
- cpu\_clk 266640000, cpm\_clk 199980000, bus\_clk 66660000

In my case, the BCM 50 just stopped at this point. All I did was to press the small reset button on the front of the BCM a couple of times at "**Starting ResetButton:**" and allowed it to reboot again. This time the prompt didn't appear and the BCM 50 powered up OK.

If you don't have a serial cable attached to the BCM 50 but you had both flashing Red LED's after powering it up and they stay constant, then you could just wait 5 minutes and then press the small reset button on the front of the BCM a couple of times and see if it reboots OK.



### Problem 3

BCM 50 system that had constant Red LED's after powering it up and they stay constant. Also all four Ethernet ports not working and the Green LED's are constant all the time.

### Solution

This was due to a patch upgrade failure. By having a serial lead connected, you can watch the BCM boot up and see where the last command line was. Please refer to Problem 2 as the fix was to just press the reset button for around 5 to 10 seconds twice and then reset the power lead. There wasn't a requirement to use the serial cable on this fault.

### Problem 4

Unable to access an unknown BCM system that had unknown IP addresses and password etc. The nnsupport password was failing when trying using a known IP address. This might have also happened if the Yellow Lithium battery was removed from the BCM motherboard.

### Solution

This was due to the date on the BCM being incorrect. It is possible to use the serial port to check the date and change it to allow the nnsupport account to work. Two methods are by using the date listed after "**date**" is keyed in to obtain the passcode, or by changing the date to the current time and date using the method below.

The example shows the date was changed from the 11<sup>th</sup> December 1913 to the 4<sup>th</sup> September 2015.

=> **date**

Date: 1913-12-11 (Tuesday) Time: 1:09:05

=> **help date**

date [MMDDhhmm[[CC]YY][.ss]]

date reset

- without arguments: print date & time
- with numeric argument: set the system date & time
- with 'reset' argument: reset the RTC

=> **date 090415152015.50**

Date: 2015-09-04 (Friday) Time: 15:15:51

Remember to add in the seconds with a "." In front at the end!