

Setting up the Raspberry Pi to work with Ciseco hardware

Please note: If you have purchased Ciseco's [4G Raspberry Wheezy SD Card, configured for Ciseco products](#), or used the Ciseco Wheezy image to create an SD card yourself, then all the steps in this document, except remote access (SSH) set up, which is optional anyway, have been done for you.

In this document we will show you how to:

1. Configure [access to the Raspberry Pi serial port](#). Ciseco hardware (e.g. [Slice of Pi](#), [Slice of Radio](#), and [Pi-Lite](#)) all communicate with programs on the Raspberry Pi via the serial port. Since the Raspberry Pi ships with exclusive access to this port, we need to do some reconfiguration. You **need** to carry out this task for Ciseco hardware to work properly.
2. [Install minicom](#), a terminal emulator that allows you to type characters on the serial port and receive characters from the serial port. You **should** carry out this task if you **don't** already have a program that allows you to easily access the serial port.
3. [Set up the Raspberry Pi for remote access \(SSH\)](#), useful if you use your Raspberry Pi as a stand-alone controller, without screen, keyboard and mouse. You **do not need** to carry out this task for Ciseco hardware to work properly.

None of the tasks are hard, but you must be careful to follow the instructions exactly. Take your time. Getting it wrong takes a lot longer to correct than taking time and doing it right first time.

We recommend you make a full backup of your Raspberry Pi image before making the changes suggested in this guide. Alternatively (or even better for a predictable outcome) use a new copy of the Raspberry Pi operating system.

We assume you have a standard Raspberry Pi with e.g. Wheezy and this has been set up correctly, following the [Basic Setup Instructions](#) in the Raspberry Pi user guide if you are unsure.

If you have purchased Ciseco's [4G Raspberry Wheezy SD Card, configured for Ciseco products](#), or use the [Ciseco Wheezy image](#) to create an SD card yourself, then all the steps in this document, except the optional remote access (SSH) set up have been done for you.

Access to the serial port

By default the Raspberry Pi's serial port is configured to be used for console input/output. Whilst this is useful if you want to login using the serial port, it means you can't use the Serial Port in your programs. Ciseco hardware, such as Slice of Pi, Slice of Radio and Pi-Lite are connected to this serial port, so we need to set the Raspberry Pi up such that it gives up its exclusive access to the serial port.

Needless to say, if you need some other way to login to the Raspberry Pi, we suggest doing this over the network using an SSH connection (see later).

A) - Disable Serial Port Login

To enable the serial port for your own use you need to disable login on the port. There are two files that need to be edited

The first and main one is `/etc/inittab`

To access this, open [LXTerminal](#) from your pi desktop, log into your pi as before and once `PI@raspberrypi ~ $` shows, type:

```
sudo leafpad /etc/inittab
```

This will open inittab with admin privileges and allow you to edit it.

This file has the command to enable the login prompt and needs to be disabled. At the end of the file. You will see a line similar to:

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Articles by list

Last Updated on Wednesday, 21 May 2014 11:29

Disable it by adding a # character to the beginning. Save the file.

```
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

B) - Disable Bootup Info

When the Raspberry Pi boots up, all the bootup information is sent to the serial port. Disabling this bootup information is optional and you may want to leave this enabled as it is sometimes useful to see what is happening at bootup. If you have a device connected (i.e. Arduino) at bootup, it will receive this information over the serial port, so it is up to you to decide whether this is a problem or not.

You can disable it by editing the file `/boot/cmdline.txt`

To find this, go back to the terminal screen and enter;

```
sudo leafpad /boot/cmdline.txt
```

The contents of the file looks like this;

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

You need to remove all references to ttyAMA0 (which is the name of the serial port we need access to). After your edits, the file should now look like this;

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

C) - Reboot

In order to enable the changes you have made, you will need to reboot the Raspberry Pi

```
sudo shutdown -r now
```

Installing and running Minicom

You will need this program to run a linux terminal through which you can send and receive text and commands via the serial port.

In LXTerminal, type:

```
sudo apt-get install minicom
```

Answer yes (Y) to any questions you may get. You should then have minicom installed.

Once installed, you will need to know the *port name* of your PC's serial port. On the Raspberry Pi this will be:

```
ls -l /dev/ttyAMA0
```

Run minicom with the following parameters:

```
minicom -b 9600 -o -D /dev/ttyAMA0
```

In this case the port name is `/dev/ttyAMA0`

You can exit minicom with Control-A x, when you are finished.

Please note that if you use Ciseco radio modules with LLAP, then you won't be able to type in minicom fast enough to ensure your messages are sent in a single packet. You can still use minicom, but we recommend you prepare a text file with the messages you wish to send and then copy/paste them into minicom.

Setting up for remote access (SSH)

Setting up for SSH is useful if you use Raspberry Pies as controllers, without screen, keyboard and mouse. You **do not need** to carry out this task for Ciseco hardware to work, so you can skip this step.

The notes here are heavily based on those found here: http://elinux.org/RPI_Advanced_Setup#Serial_connection

A) - Add the pi user to the dialout group

In LXTerminal, type:

sudo usermod -a -G dialout pi

Before setting up for remote access, you may want to change the default password for your Pi, especially if it starts appearing on networks that you don't fully control, such as at work, or the internet at large.

B) - Enable SSH

Enabling basic remote access is simple: just enable SSH. When you first switch on the Pi you can do it at the very first configuration screen. If you have not done it there, you can do it by the following command:

sudo mv /boot/boot_enable_ssh.rc /boot/boot.rc

Reboot your Pi and towards the bottom of the status messages you should see a line like:

Starting OpenBSD Secure Shell server: sshd

When you see that, you know that SSH is enabled. You should then be able to SSH to your Pi, using its IP address, e.g.:

ssh 192.168.1.224

Of course you need to use the IP address of your Pi, which will most likely be different from the one above. To find the IP address of your Pi, log into it and then type:

ip addr

If using the IP address works for you, **stop here**. You are done!

C) - Setting up Avahi

Most networks use DHCP, so the IP address of your Raspberry Pi will not be fixed. I do like to address my Raspberry Pi regardless of its actual IP address. And I like the solution to be robust. Here is one that works, even it is a bit elaborate to set up. It is based around zeroconf and requires the installation and configuration of a package called avahi.

First, install avahi with the following commands on the Pi:

sudo apt-get install avahi-daemon

and then initialise the daemon with:

sudo insserv avahi-daemon

Next you need to create a configfile for Avahi at **/etc/avahi/services/multiple.service**

The contents of this file should look like the following, courtesy of aXon on the Raspberry Pi

forums:

```
<?xml version="1.0" standalone='no'?>

<!DOCTYPE service-group SYSTEM "avahi-service.dtd">

<service-group>

    <name replace-wildcards="yes">%h</name>

    <service>

        <type>_device-info._tcp</type>

        <port>0</port>

        <txt-record>model=RackMac</txt-record>

    </service>

    <service>

        <type>_ssh._tcp</type>

        <port>22</port>

    </service>

</service-group>
```

Apply the new configuration by restarting the avahi daemon, with:

```
sudo /etc/init.d/avahi-daemon restart
```

The Pi should now be addressable from other machines as raspberrypi.local, for example:

```
ssh pi@raspberrypi.local
```

Since Avahi negotiates with your router, it may take a little while before your machine becomes recognised and visible on the network, so don't despair if you do not see your Raspberry Pi straight away. Just retry every few minutes.