

# PROGRAMMING AN ARDUINO ON OR VIA CISECO HARDWARE FROM THE RASPBERRY PI

In this advanced user guide we will show you how to set up the Raspberry Pi so you can

- 1.program the Arduino ATmega on the [Pi-Lite](#) to run your own or a third party program
- 2.program another Arduino ATmega wirelessly via a GPIO connected radio (e.g. [Slice of Radio](#))

BEFORE YOU START, please make sure your Raspberry Pi has been set up to work with Cisecco hardware and that you have followed the steps in [Setting up the Raspberry Pi for Cisecco hardware](#).

If you are at all unsure about any of the steps below, do **NOT** proceed. In any case 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 and apply the changes detailed in the guide: [Setting up the Raspberry Pi for Cisecco Hardware](#) before following this guide.

These instructions are specific for Cisecco hardware ([Slice of Radio](#) and [Pi-Lite](#) in particular).

*NOTE: If you have purchased Cisecco's [4G Raspberry Wheezy SD Card, configured for Cisecco products](#), or download the [Cisecco Wheezy image](#) to create an SD card yourself, then all the steps in this document have been done for you.*

## Setting up for programming an Arduino from the Raspberry Pi

For the Raspberry Pi to correctly program an Arduino processor on or via Cisecco hardware, we will need to:

- 1.Install the Arduino IDE on the Raspberry Pi (step 1 below);
- 2.Arrange for the Arduino IDE to correctly issue a reset command to the Arduino at the end of uploading the program (step 2 below);

None of these steps are complicated, but you must carry them out very carefully. Mistakes are always more difficult to fix than doing it slowly and right first time. .

### Step 1: Installing the Arduino IDE

To install the Arduino IDE on your Raspberry Pi start LXTerminal and type:

```
sudo apt-get update
```

This will update the package lists and make sure that the Arduino package can be found when you run the next command:

```
sudo apt-get install arduino
```

You need to answer yes to the questions you get. The process may take a while, so be patient. Once done, you should see "a new category called "Electronics" in your start menu. You should see "Ardiono IDE" in this category, as well as in the "Programming" category in your start menu.

The serial port on the GPIO that we want to use to communicate with the Arduino on the Pi-Lite is /dev/ttyAMA0. Unfortunately, the Arduino IDE does not recognise this port. It prefers to use /dev/ttyS0. To get round this we link /dev/ttyS0 to /dev/ttyAMA0 and make sure this link is permanent. To do this, we need to create a file called [/etc/udev/rules.d/99-ty.rules](#) using a text editor.

In LXTerminal type

```
sudo leafpad
```

In the text editor type:

```
KERNEL=="ttyAMA0?,SYMLINK+="ttyS0? GROUP="dialout"
KERNEL=="ttyACM0?,SYMLINK+="ttyS1? GROUP="dialout"
```

Then select **File > Save As** and type [/etc/udev/rules.d/99-ty.rules](#)

Close leafpad and don't worry about the Gik Warnings that will appear in the LXTerminal window (they are to do with the fact that Leafpad cannot store temporary and backup files in /etc).

Then reboot using:

```
sudo reboot
```

### Step 2: Updating AVRdude

AVRdude is the program that, amongst other things, uploads the compiled Arduino sketch to the Arduino on the Pi-Lite. For that programming step to be successful, it needs to send a "reset" signal to the Arduino at the end of the upload. Normally, this reset is sent via the DTR line of the serial port, but the GPIO connector on the Raspberry Pi does not support DTR. Instead Cisecco route DTR out via GPIO22 (Pin 15). For this signal to be correctly conveyed to the Arduino, we need a wrapper round AVRdude, which will use GPIO22 instead of DTR.

The wrapper Cisecco provide is a Python script, and it in turn uses the Python RPi.GPIO library to make access to the GPIO easier. So, first we must check if this library is installed, and if not, install it. Either at a text console or using LXTerminal enter the following:

```
dpkg -s python-rpi.gpio
```

and

```
dpkg -s python3-rpi.gpio
```

You will need to see a version of python-rpi.gpio of 0.5 or greater. If it is not installed already, or if the version number is below 0.5, you need to get and install two packages: [python-rpi.gpio](#) and [python3-rp.gpio](#).

Now, there is quite a bit of development taking place with these packages, so you will need to go and search for the latest versions of the packages if the instructions below don't work. The easiest way at time of writing is to use a browser and go to <http://code.google.com/p/raspberry-gpio-python/>

Then select the latest packages and download them. (In our case we downloaded version 0.5.2a. If you download another version, adjust the commands below with the your version number.)

Again, using LXTerminal and in the directory to where you downloaded the packages, enter the following, waiting for each command to complete:

```
sudo dpkg -i python-rpi.gpio_0.5.2a-1_armhf.deb
```

```
sudo dpkg -i python3-rpi.gpio_0.5.2a-1_armhf.deb
```

# Articles by list

Last Updated on Wednesday, 10 July 2013 16:03

---

Now we can proceed and install AVRdude. Here is how to do it:

Get the AVRdude for the Raspberry Pi from: <https://github.com/CisecoPlc/avrdude-rpi>

Click on the ZIP button and save the file. It is called **avrdude-rpi-master.zip**

Double click on the file to open Xarchiver and click **Action>Extract**. You should see a folder called **avrdude-rpi-master** appear. In it are two files: **avrdude-autoreset** and **autoreset**, as well as a readme file, which you can ignore.

You need to copy both files into your `/usr/bin` directory, then rename the original avrdude to avrdude-original and symlink avrdude-autoreset to become avrdude. You will need `sys admin` privileges, so open LXTerminal, go to the directory in which you saved **avrdude-rpi-master**, and type the following commands, waiting for each one to complete:

```
cd /avrdude-rpi-master/
```

```
sudo cp autoreset /usr/bin
```

```
sudo cp avrdude-autoreset /usr/bin
```

```
sudo mv /usr/bin/avrdude /usr/bin/avrdude-original
```

```
sudo ln -s /usr/bin/avrdude-autoreset /usr/bin/avrdude
```

Just to be sure everything is as it should be open the file `/usr/bin/autoreset` and look at line 15. It should look like this:

```
pin = 22
```

If it shows that a different pin number is specified, then set it to 22. It needs to reflect the pin that we have wired up to be the reset pin via the Pi-Lite.

Close all windows and editors, ready for the next step.

We hope these instructions were useful. Please send comments for improvements to the Forum.