**Using Nanpy to connect your RPi to an Arduino**

**Preparing the SD card**

You a need a 4 GB class 4 card (slow). Copy the latest version of Raspbian Wheezy (9/2/2013) onto it and expand the root partition. Reboot the Pi.

**Install setuptools**

You need python **setuptools** to install **nanpy** on your card. This is not in the current distribution. You can download it from the Web.

Start Midori and type in the URL box <https://pypi.python.org/pypi/setuptools>

Scroll down to the Linux instructions and then on to the downloads. We want the file:

[setuptools-0.6c11-py2.7.egg](https://pypi.python.org/packages/2.7/s/setuptools/setuptools-0.6c11-py2.7.egg#md5=fe1f997bc722265116870bc7919059ea)

Click on it and you will be asked to open or download. Click on SAVE. It downloads very quickly.

Close Midori and you should see the egg file in the pi directory.

Open the LX Terminal and type in:

**sudo sh** [**setuptools-0.6c11-py2.7.egg**](https://pypi.python.org/packages/2.7/s/setuptools/setuptools-0.6c11-py2.7.egg#md5=fe1f997bc722265116870bc7919059ea)

This is a very quick installation.

**Install serialpy**

Using Midori go to <https://pypi.python.org/pypi/pyserial>

Download **pyserial-2.6.tar.gz**

Make a **temp** folder and move the downloaded file into it.

Using LXTerminal

**cd temp** Change to the temp directory

**gunzip pyserial-2.6.tar.gz**  to unzip it

**tar –xvf pyserial-2.6.tar** to untar it

**cd pyserial-2.6** move into the new folder

**sudo python setup.py install** to install it

Your Pi can now use serial communication

**Install the Arduino software**

Type ‘startx’ to start the GUI.

Open the LX Terminal.

Type in the the following commands:

**sudo apt-get update**

**sudo apt-get install arduino**

Answer ‘Y’ when asked if you want to continue. This installation takes some time…..

You can now program your Arduino from the Arduino IDE.

**Install Nanpy**

The next step is to download the nanpy files:

I find it much easier to do on a Windows PC and then transfer the unzipped folder to the Pi via a memory stick.

On a PC using your browser navigate to <https://github.com/nanpy/nanpy> and click on the ZIP button. This downloads the zipped directory. Unzip it and copy the **nanpy-master directory** via a USB stick to your pi directory.

Open the LX Terminal and navigate to the firmware directory in **nanpy-master**.

Connect your Arduino via a USB cable to the Pi.

**cd nanpy-master**

**cd firmware**

**export BOARD=uno** (Type ‘make boards’ for a full list)

**make**

**make upload** This also takes some time…….

**This RED section needs to be done each time you connect the Arduino to the RPi. It loads the Arduino part of nanpy into the Arduino .**

**cd ..** Move back to **nanpy-master** directory

**sudo python setup.py install**

This adds the RPi part of nanpy to Python2 and only needs to be done once. You can now use the Arduino as a I/O board for the Pi. This adds 19 extra I/O pins – 6 can be used for 10-bit analog inputs (range 0-1023) and 6 as PWM outputs (range 0-255).

**Testing**

The Arduino really needs to be connected via a powered USB hub so that it does not take too much power from the RPi.

On your Arduino connect an LED in series with a resistor (about 250 Ohms for protection) between pin 10 and ground. The longer lead of the LED, the anode, goes towards pin10 and the shorter, the cathode, towards GND. Pin 10 allows PWM (Pulse Width Modulation).

From LXED start LXTerminal and type: **Sudo idle**

**(With sudo you can run the program from the Run menu in Idle)**

Click on File,then New window; to open a new window and type in the following program:

#!/usr/bin/env python

# LED with 250 Ohm resistor on Pin 10 to GND

# Tony Goodhew - 10 May 2013

from nanpy import Arduino

from nanpy import serial\_manager

serial\_manager.connect('/dev/ttyACM0') # serial connection to Arduino

from time import sleep

LED =10 # LED on Arduino Pin 10 (with PWM)

Arduino.pinMode(LED, Arduino.OUTPUT)

print"Starting"

print"5 blinks"

for i in range(0,5):

Arduino.digitalWrite(LED, Arduino.HIGH)

sleep(0.5)

Arduino.digitalWrite(LED, Arduino.LOW)

sleep(0.5)

print"Changing brightness of LED"

bright = 128 # Mid brightness

Arduino.analogWrite(LED, bright)

Arduino.digitalWrite(LED,Arduino.HIGH) # Turn on LED

for i in range(0,200):

bright = bright + 8

if (bright > 200): # LED already full on at this point

bright = 0 # Minimum power to LED

Arduino.analogWrite(LED, bright) # Change PWM setting/brightness

sleep(0.05)

Arduino.digitalWrite(LED,Arduino.LOW) # Turn off LED

print"Finished"

Save and run the program from the idle menu.

**Problem:** If you pull the USB cable out of the Arduino while the Pi is controlling it you may need to re-boot the Pi before it will re-connect. You may also need to re-do the instructions at the bottom of page 1 if the firmware gets corrupted.

Thanks to Andrea Stagi [stagi.andrea@gmail.com](mailto:stagi.andrea@gmail.com) for the software. See MagPi – issue 8, page 12

Please let me know how you get on with this guide. Report errors or problems to: [tony.goodhew@ntlworld.com](mailto:tony.goodhew@ntlworld.com)