Wensleydale Software

# Lights

APA102 144LED strips connected to form a single loop. Power is fed into each end of each individual strip, as the signal does not travel down the strip correctly if the power is not fed in at regular intervals (from observations more than 5 strips). 3 touch pad sensors control the colour, speed and pattern on the LEDs.

Each LED is individually addressable and its brightness and RBG colour can be set.

### Existing GitHub code.

HardwareControl/Lights/Physical/apa102.py – Modified to fix bug where last few LEDs where not getting the correct signal.

Internals/Lights/colourschemes.py

Internals/Lights/colourcycletemplate.py

Scripts/run\_light\_patterns.py

### Folders

Main – Enums, global variables,

Internals –

Lights – code that creates the patterns on the LEDs

Utils – Utility classes, timer, logger and stoppable thread.

HardwareControl –

Lights – Code which sends the patterns to the LEDs

Environment – Code which controls the global variables and listens to the touch pads.

Logs – Log files are created here.

Scripts – Routines to run and test patterns and pattern combinations.

Development – Throw away experimental code.

Testing – Tests for logger and other utils.

Class wblock –

* Each block of LEDs that want to be given its own pattern.
* Start and End index of the block.
* Colour, Speed and Pattern to be displayed on block. Note some of the patterns ignore colour/speed.
* Stores a “pattern index” which stores information about where in the pattern cycle the current block is. This is used when changing speed so the LEDs do not jump.

Class colourcycletemplate –

Patterns.py

This file contains all the patterns that can be displayed on a set of blocks. Patterns here each have a corresponding enum and also need to be added to the BlockLightPattern update block method.

BlockLightPatterns.py

Class BlockLightPattern

Subclass of colourcycletemplate that works for blocks of LEDs each block set to the its local colour speed and pattern.

Class ChangingBlockLightPattern

Subclass of BlockLightPattern that listens to the buttons and sets the LEDs based on global variables.

Class GlobalPattern

Subclass of BlockLightPattern that sets every block to the same pattern.

Various other classes that run specific combinations of patterns.

# Eyes

4 servo motors control the eyes. These have a canned routine that they repeat.

### Folders

Main – Enums, global variables,

Internals –

Eyes – code that pulls together different movements into a rountine.

Utils – Utility classes, timer, logger and stoppable thread.

HardwareControl –

Eyes – Code which sends required movement to the eyes.

Logs – Log files are created here.

Scripts – Routines to run and test movements and movement combinations.

Development – Throw away experimental code.

Testing – Tests for logger and other utils.

# Tail

# SSH into the pi using Putty and TightVNCServer.

TightVNCServer is already been installed on the pi but for future reference:

**sudo apt-get install tightvncserver**

**ifconfig –** this lists the ip address of the rpi. As we are connecting via ethernet cable. Use the inet number in the eth0: section. These numbers are listed below for our pis. If SD cards are swapped in and out then these may change.

**IP addressed for the pi when using ethernet direct connection. (believe this is pi specific not sd card specific)**

LED pi 169.254.210.239

Eye pi 169.254.125.42

If these don’t work, your computer might be providing the IP address (Ubuntu)

Type **cat /var/lib/misc/dnsmasq.leases** to get the IP address.

**Connecting via putty.exe**

Open putty.exe

Type IP address into the box and click open – used saved session if available.

User pi

Pw: wensleydale

**Connecting via TightVNCViewer**

**vncserver –** enables vnc server so we can use tightvncserver (first time will ask for pw and possibly cause issues – see fixes below.

Take note of the port number the vnc server opens with.

Close putty window.

Open tightvncveiwer

Ip address : port number click connect.

**Issues**

**Pi gets stuck on login screen.**

I think I may have a fix that doesn't require a card reflash. Please bear with me, I'm new to linux.  
  
I installed TightVnc and set it up to auto start (which I never got working satisfactorily), then on reboot started getting the login screen that would only accept Root (once activated). SSH still worked and I could login as pi.  
  
I've since removed TightVnc to try to help and removed all the autostart changes I made. Still getting the same login screen.  
  
Here's the fix that got me back to booting straight into Lxde as user Pi with no login screen:  
  
1. Boot your R-Pi  
  
2. SSH log in with your user account (normally user: pi, password: raspberry)  
  
3 Enter your home directory by typing

**Code:**[**Select all**](https://www.raspberrypi.org/forums/viewtopic.php?t=27905)

cd ~

4. Type

**Code:**[**Select all**](https://www.raspberrypi.org/forums/viewtopic.php?t=27905)

ls -A

5. You will now see which files are in your home directory. Look if you see a file called .Xauthority  
  
6. Type

**Code:**[**Select all**](https://www.raspberrypi.org/forums/viewtopic.php?t=27905)

mv .Xauthority .Xauthority.backup

7. Type

**Code:**[**Select all**](https://www.raspberrypi.org/forums/viewtopic.php?t=27905)

sudo chmod 777 /home/pi

8. Type

**Code:**[**Select all**](https://www.raspberrypi.org/forums/viewtopic.php?t=27905)

sudo reboot

That did the trick for me, let me know if it works for you.  
I think I'll have a go with x11vnc and see if that's better.