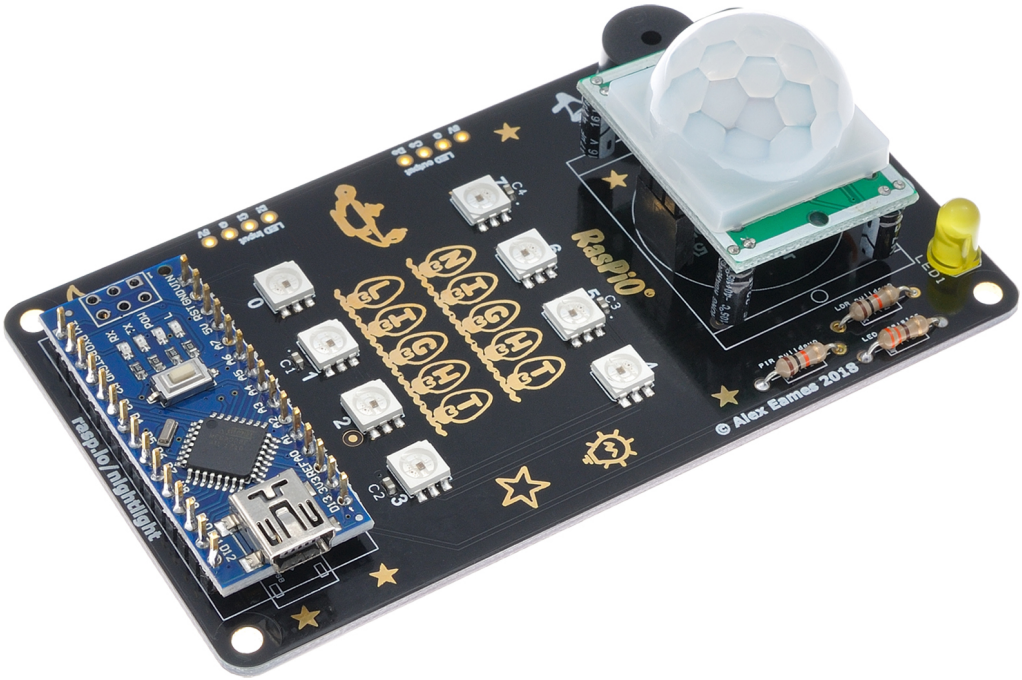
Using Your RasPiO®

Night Light



by Alex Eames

Introduction

The RasPiO Night Light is a motion-activated RGB light that lights your way in the dark. It's a lovely build-it-yourself kit designed to be gorgeous to look at and fun to build, use and tweak.
It's great for providing a bit of automatic nocturnal illumination in places like your...

- hall
- garage
- landing
- workshop
- shed

RasPiO Night Light has a passive infra-red (PIR) sensor to detect motion, a light-dependent resistor (LDR) to detect light levels and a buzzer for alerts.

It uses an Arduino nano clone to control everything and the open source FastLED library to drive eight SK9822 LEDs (APA102 compatible).

An indicator LED lights when the PIR detects motion.

RasPiO Night Light has a frosted Perspex case to diffuse the light and house the electronics. The back has 4-way mounting holes enabling mounting in any orientation.

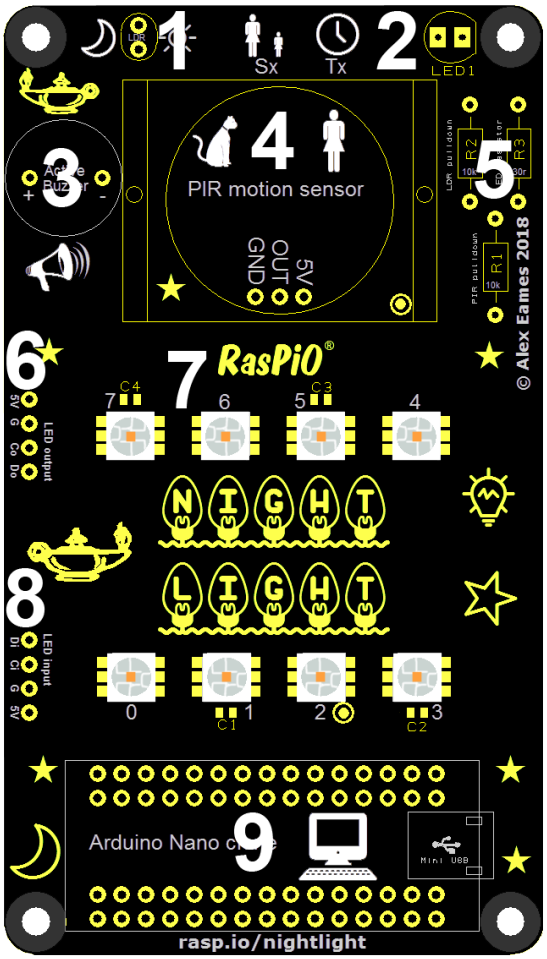
The PIR sensitivity and duration can be adjusted in hardware using the onboard potentiometers. LDR sensitivity can be adjusted in software (it uses an analog port on the nano to read the LDR). The buzzer usually beeps once on startup, but can be controlled or disabled in the software sketch.

Some simple through-hole soldering is required to assemble the board.

RasPiO Night Light LEDs are incredibly bright when used at full power with no diffuser. Please be sensible (or wear shades).

RasPiO Night Light Instructions

Know Your Night Light Board



1 The Light sensor (LDR) connected to A7 lets the Arduino nano tell whether it's dark enough to switch on the LEDs when someone is detected (light threshold is programmable).

2 The yellow LED connected to D13 shows when the PIR sensor detects someone (even in daylight). Its function can be reprogrammed.

3 The Piezo buzzer connected to D2 emits a piercing sound when D2 is HIGH. Its function can be disabled or reprogrammed.

4 The PIR motion sensor connected to D12 detects changes in infra-red (heat) passing in front of it. The sensitivity and duration are controlled by two potentiometers (Sx & Tx)

5 Three resistors are used: R1 & R2 10kΩ, R3 330Ω

6 LED output holes can be used to connect additional LEDs SK9822/APA102 *etc*. to lengthen the 'chain'.

7 The onboard RGB LEDs are SK9822

8 LED input holes are used mostly for testing the boards at the time of manufacturing, but are there for use if required.

9 An Arduino nano clone is the 'brains' of this device.

Assembly Instructions

Night Light Board Assembly

There is a comprehensive soldering and case assembly video here...

https://youtu.be/pUnwkKNYH8k

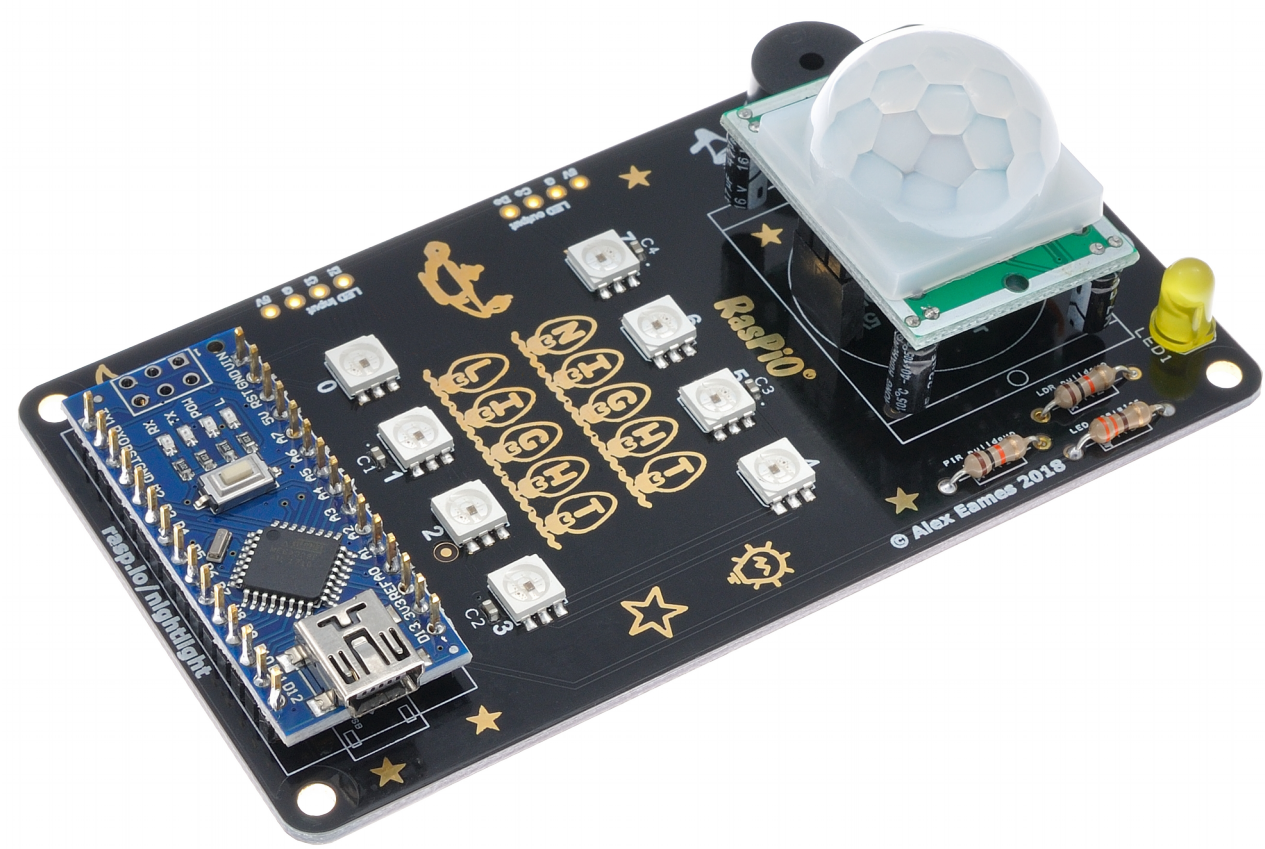
Suggested Soldering Order

Generally it's easier to solder the 'lowest' components first and build up the board in increasing component height.

1. Three resistors. R1 & R2 are 10kΩ (Brown, Black, Orange, Gold), R3 is 330Ω (Orange, Orange, Brown, Gold)
2. Buzzer - ensure correct polarity. (Longer leg is positive.)
3. LED - ensure correct polarity. (Longer leg is positive.)
4. LDR (make sure to push the legs in far enough so the LDR only sticks up <1cm from the board as the legs are quite weak and bendy)
5. 3-way female header for PIR (must be perpendicular to the PCB)
6. Arduino nano headers (whichever way up you prefer. If you choose to put them 'long side down' you'll need to trim all the pins on the underside of the board, but the topside will look neater. You can use the main PCB to hold them in place to start with.)
7. Arduino nano to PCB (sand the nano board edge smooth at each end to avoid fouling the spacers)

8. Plug in the PIR motion sensor so it covers the footprint on the PCB. Sometimes the small capacitors on the PIR sensor get slightly bent in transit. If necessary they can be gently bent back into alignment.

Here's what the assembled PCB should look like...

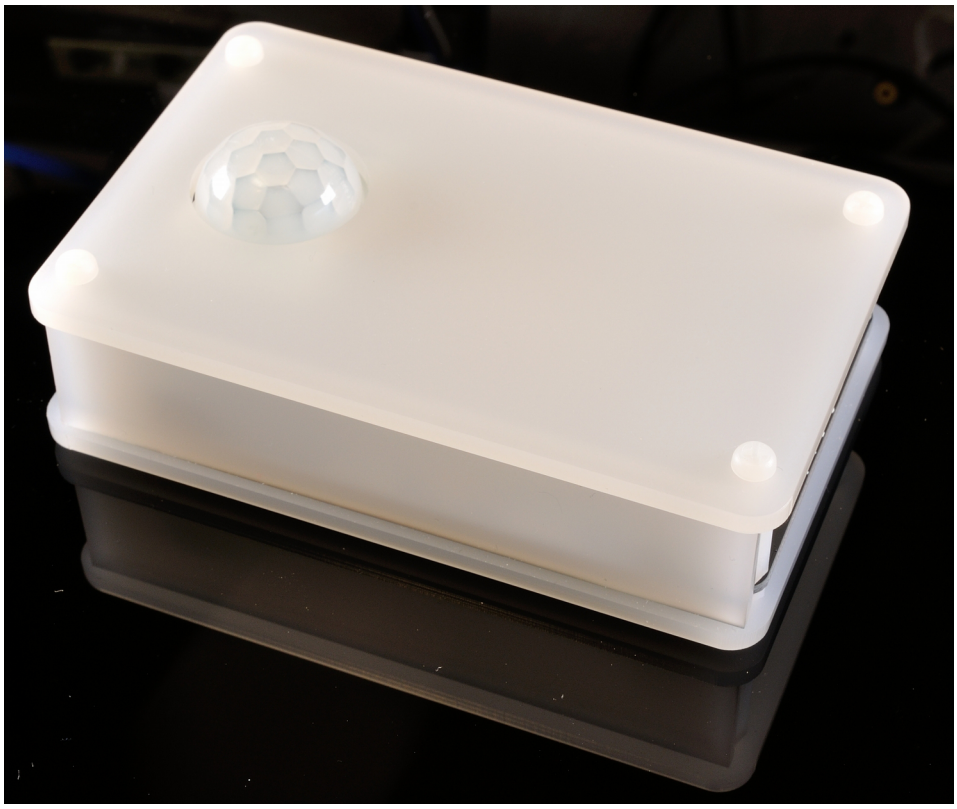


Night Light Case Assembly

Case assembly is covered in the video. In case you prefer written instructions, those are here...

1. Put a bolt through each of four corner holes on the base and a nut loosely at the top for the PCB to rest on
2. Place the PCB on top. Don't fully tighten the nuts yet.
3. Thread a spacer onto each of the protruding bolts (can be a bit tight at the Arduino end).
4. Put the case top on and loosely fit the top bolts.
5. Fully tighten the bottom bolts.
6. Tighten the spacers.
7. Install the sides and tighten the top bolts until they are held in place. Do not over-tighten.

When you've finished it should look like this...

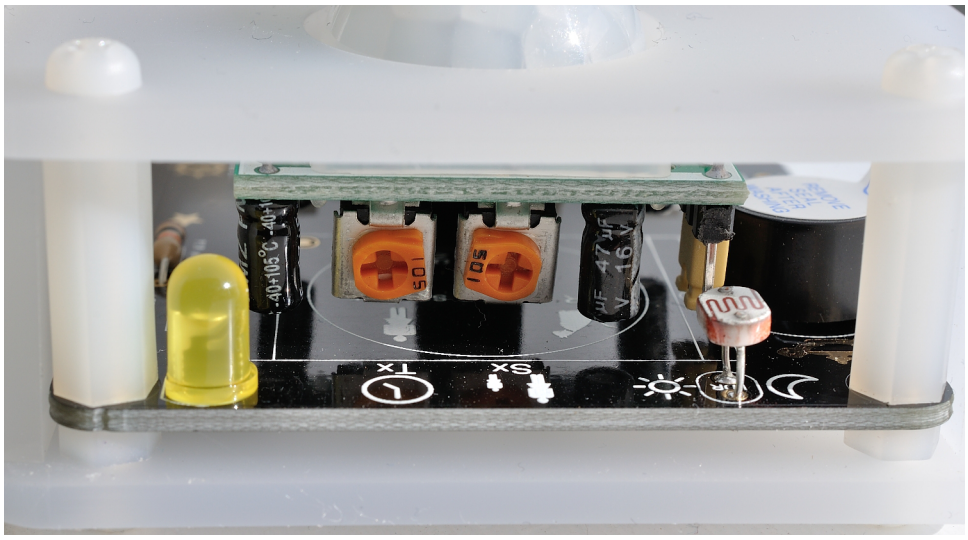


Adjusting the PIR Sensor

The sensitivity and duration settings on the PIR sensor are both adjusted using its onboard potentiometers (by rotating the orange objects in the photo). You will want to tweak these to fit your own requirements, but a good starting point is shown in the photograph.

The potentiometer marked Tx sets the duration that its signal pin remains HIGH when movement is detected. It's better to keep this duration short and set the 'lights on' duration in software.

The potentiometer marked Sx sets motion detection sensitivity. Your circumstances may vary, but I've found that adjusting this to near its maximum setting works best.



The PIR jumper can be left in its default position.

Arduino IDE & Driver Installation

Install Arduino IDE

If you don't already have the Arduino IDE (Integrated Development Environment) installed, install it according to the instructions here...
https://www.arduino.cc/en/Guide/HomePage

For instructions, go to the "Install the Arduino Desktop IDE" section and click the appropriate link for your computer's operating system (OS).

You will also need to download it from here...
https://www.arduino.cc/en/Main/Software

Install Driver

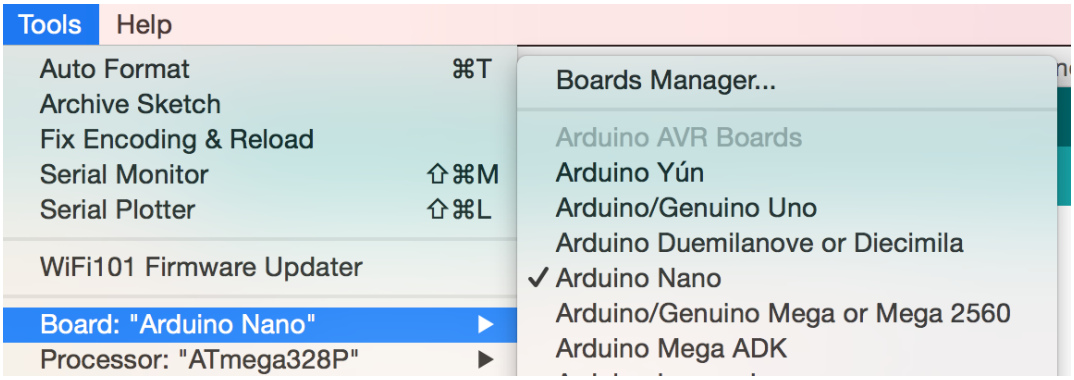
If using Windows or Mac, install the driver for the CH340G serial chip. Download the appropriate file from here...

https://wiki.wemos.cc/downloads

Unzip it and follow the instructions in the ReadMe PDF file.

Setting up the Arduino IDE for nano

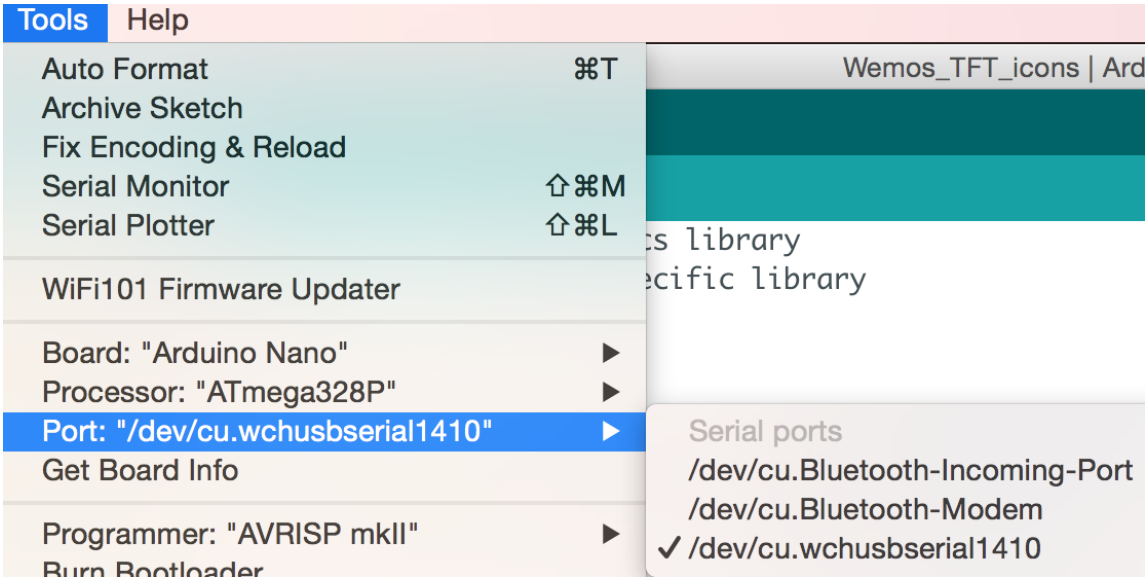
Run the Arduino IDE and select Tools > Board > Arduino nano



This will set up the IDE to compile code correctly for the nano clone.

Testing Your System

Now plug your nano clone into your computer using the mini-USB cable provided. Choose Tools > Port **and you should be shown the available USB options. In my case (on a Mac), the CH340 chip shows up as /dev/cu/wchusbserial1410**

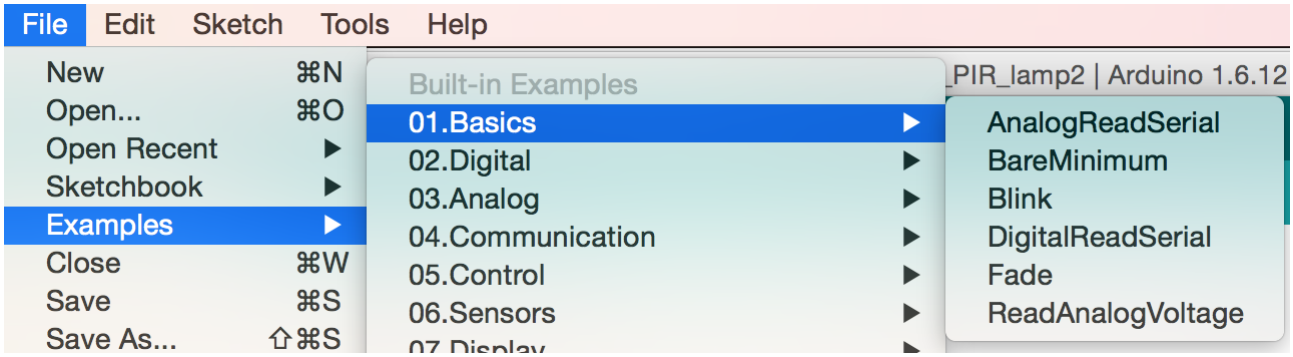


If you have a recent version of the Arduino IDE, it may be necessary for you to select Processor: "ATmega328P Old Bootloader" **to be able to flash sketches to your nano.**

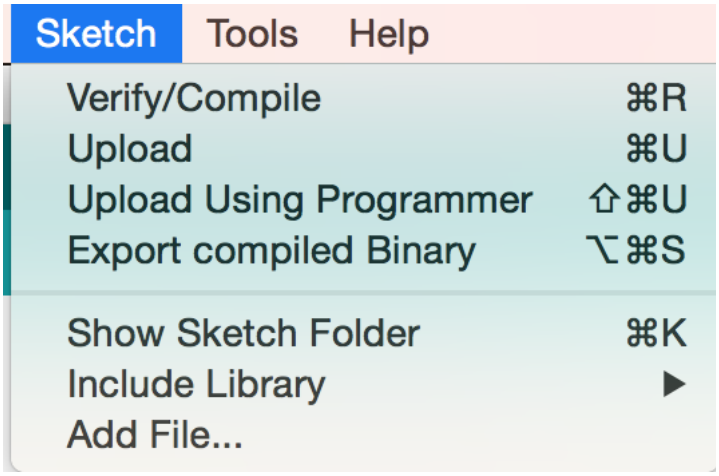
Flash the Blink Sketch

To test that communication is working, let's flash the blink sketch to the Arduino nano clone. This will make the onboard LED and the PIR LED both flash on and off at 1s intervals.

Load the blink sketch File > Examples > Basics > Blink



Now change the delay times from 1000 to 100 and upload the sketch to the nano clone with Sketch > Upload



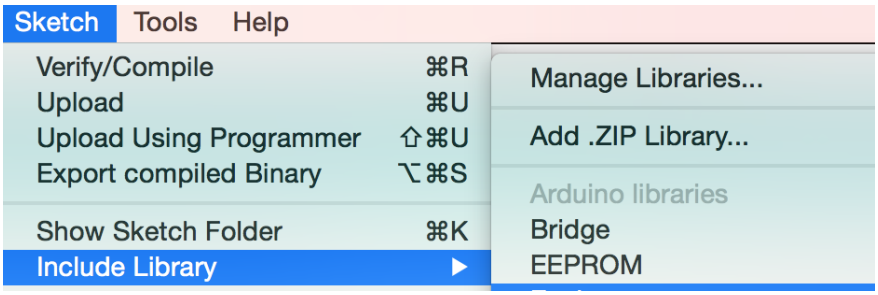
If this works properly, the onboard LED and the PIR LED both flash on and off at 0.1s intervals. If not, check your USB connection and try again.

Install the FastLED Library

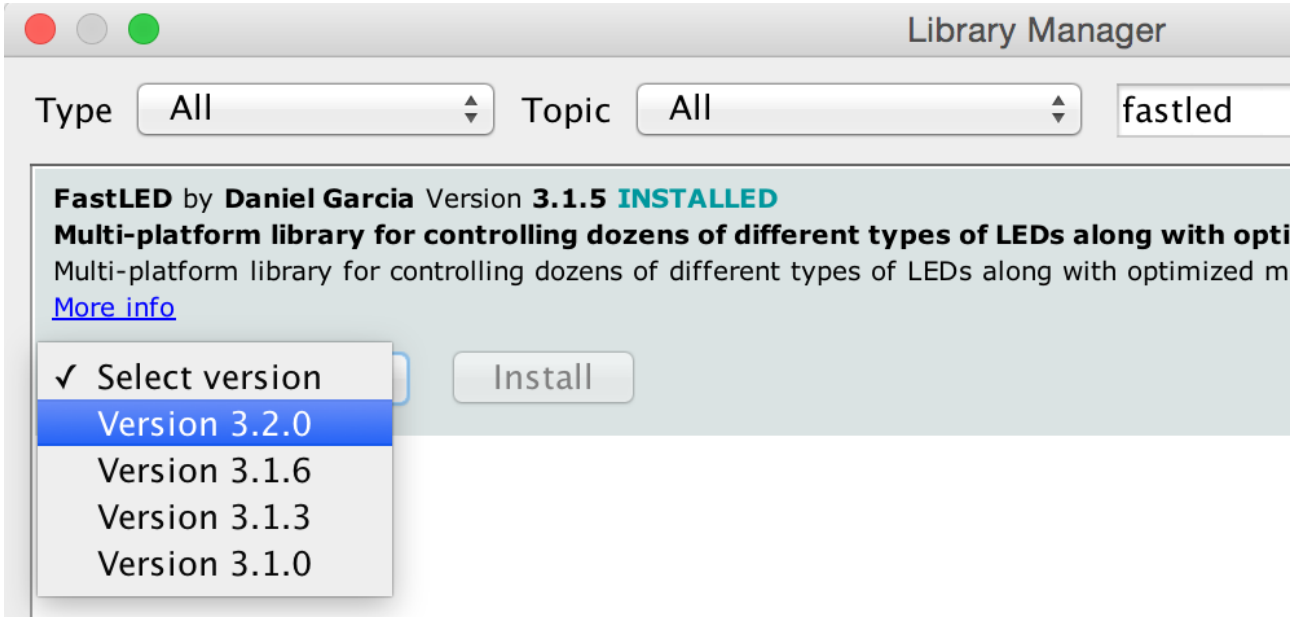
To drive the SK9822 LEDs, we need to install a software library called FastLED.

Open the Arduino Library Manager with...

Sketch > Include Library > Manage Libraries



In the search box, type 'fastled', choose a version and click 'install'



Now everything is ready to start programming your RasPiO Night Light

Obtain and Flash the Demo Sketch

The RasPiO Night Light demo sketch can be found on GitHub...

https://github.com/raspitv/nl

Start a new sketch with File > New

Delete the few lines of default code that appear, so the sketch is empty.

Then, in your web browser, find the raw code for the demo sketch here...

https://raw.githubusercontent.com/raspitv/nl/master/nl.ino

Edit > Select All
Edit > Copy

Then in the Arduino IDE

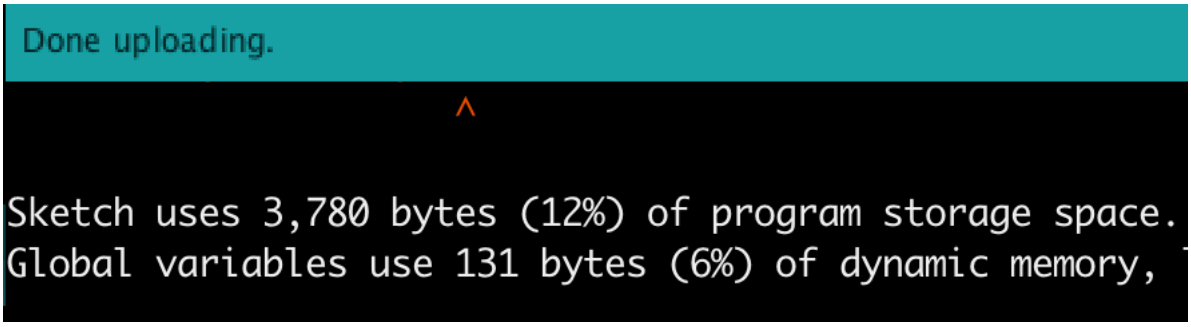
Edit > Paste

This should copy the demo sketch to your computer. Now save it with...
File > Save As **and give it a name.**

Now it's time to compile and flash (upload) the sketch to the Night Light. With the Night Light connected to USB, click...

Sketch > Upload **as you did before.**

If it works properly, you should see a message like this...



...and the buzzer should beep once. This means your Night Light is programmed and ready for use.

General Usage Notes

PIR Warm-up time

PIR sensors require about one minute of warm-up time before they will start detecting.

Eliminating the Arduino LEDs

If you find the Arduino's onboard LEDs annoying, you can cover them with a blob of Blu-tack or plasticine. This should not impede functioning of the device.

Disabling the Buzzer

You can disable the buzzer on line 8 of the demo sketch by changing 0 to 1.

```
int buzz_disable = 0;
```

In the demo sketch, the buzzer will only ever sound briefly on startup.

Tweaking the Darkness Threshold

To adjust the darkness threshold for when the LEDs will switch on, tweak line 54...

```
if (pir == 1 && ldr <= 50 ) {
```

Reducing the 50 to nearer 1 will mean it needs to be darker before the LEDs will switch on.
Increasing it to nearer 1023 (maximum) will mean it needs to be lighter.
In practice, 50 to 150 seems to work quite well, but your environment and needs may vary.

Powering the Board

RasPiO Night Light requires a 5V, 0.5A power source (0.5A is minimum – it can be greater). You can use any USB power supply with a mini-USB cable plugged into the Arduino nano clone.

Alternatively, you can connect a well regulated 5V supply directly to any of the through-holes labelled 5V on the PCB (negative to GND). There's one in the Arduino nano area of the board, one in LED inputs and one in LED outputs. They are all connected.

Finally, you can also use the V_{in} hole in the Arduino nano area to connect a supply between 6-12V. The 5V regulator on the nano clone will provide 5V to the rest of the board. The Arduino nano clone's ports are labelled on the underside of the board.

Specifications

The eight onboard RGB LEDs are SK9822 (click here for the datasheet) spaced 1cm apart.

SK9822 Power Usage

Single SK9822 LED	Current Consumption
At rest (224,0,0,0)	0.28 mA
At full RGB brightness (255,255,255,255)	51 mA
Full brightness, full Red (255,0,0,255)	17.7 mA
Full brightness, full Green (255,0,255,0)	17.7 mA
Full brightness, full Blue (255,255,0,0)	17.5 mA

If you want maximum brightness white, budget for 60 mA per LED. For most applications, you can likely manage with about half of that as you don't usually have all the LEDs at full brightness at the same time.

Final Word

You should now have a thorough overview of how to set up and use your RasPiO Night Light.

I hope you have a lot of fun with it. There is always more to learn and further to go.

And if you haven't yet got yourself a RasPiO Night Light kit, or need another, you can get that from here...

http://rasp.io/nightlight