

GPOD GROWLIGHT



An Arduino-based project from RossGK Tangibles

What is it?

GPOD is a wiring interface with user-controls for operating a simple grow light. It uses just one control button and a display to let you set up a day/night illumination schedule for your plants or seedlings.

What are the parts?

The GPOD consists of:

- Circuit board
- Arduino* Nano (5V version)
- Power Supply
- Protective Case
- SPLEDs in strings, rings or panels

What are SPLEDs?

Often available under the name NeoPixel* but also available under many other names, they are **serially-programmable light-emitting diodes** that have individual colour and brightness control.

What Can I Do?

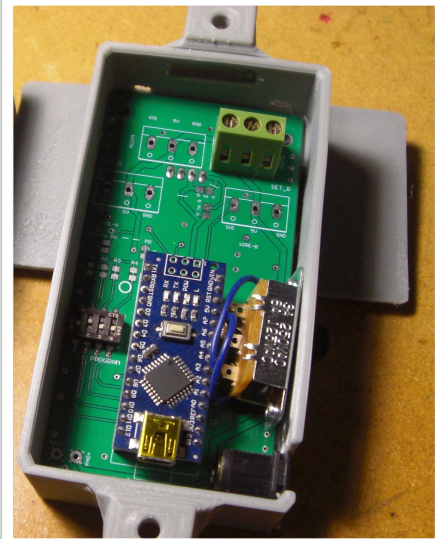
Run a plant grow light or use the platform to explore your own lighting control project based around Arduino Nano.

Get Started

If you have a fully assembled GPOD device, you can simply plug in the power, and turn the unit on to get started.

To set the time, button presses will advance the clock – hold down to go faster. The time you set determines your point in the cycle, and illumination for sunrise, day, or sunset, but off for night. While not necessary to align your GPOD timing with your actual local day cycle, it may be beneficial if your grow light aids existing daylight rather than creating the lighting all by itself.

If you have a GPOD kit, you will need a soldering iron to assemble the various pieces onto the circuit board, and then you can connect your SPLEDs to the wiring blocks and plug in the Arduino* Nano. In the simplest usage, you can run the system as a grow light, and it will keep your plants perpetually happy in an otherwise dim location (don't forget to water them!)



Beyond use as a grow light, the GPOD system is a simple platform from which you can create new fun and interesting NeoPixel* projects. The open source code allows you to revise and generate other lighting displays. The display allows you to control your own creations with the press of a button.



Caution - Be Careful

Using many SPLEDs illuminated together can draw a high level of current. The GPOD board has a master power switch and has tracks to route the power to individual LED banks. Only on/off and colour instructions come from the Arduino Nano device.

The device's power switch can probably handle about 2.0 amperes, and the board tracks and connectors something similar. If you try to go beyond 40 total SPLEDs across all the channels, and at high brightness, you can find yourself sinking a lot of current. That can cause heating, and in extreme situations, create a fire risk. More likely you can burn out a power supply, or melt connectors.

The control software is coded to drive four channels with a maximum of 40 elements each. If you wish to use more on any selected channel, you'll have to alter the code. That requires an understanding of the Arduino ecosystem. You may have those skills already, or you can learn how through many diverse and helpful tutorials available online.

So proceed with caution and have fun!

Choosing SPLED Arrays and GPOD Settings

A NeoPixel string is something many hobbyists and makers may have around home already. These are driven with WS281x chip and may be in an RGB or RGBW format. You may also have other arrays of SPLEDs driven from online sellers which are driven by the same or similar WS2811, -12 or -13 chips, so may vary in data signal timing or format. GPOD helps to manage two common options, with a set of DIP switches on the circuit board. These tiny switches can be set with the tip of a pencil.

In normal operation, the GPOD sets up a warm-orange sunrise lighting mode first thing in the morning, an intense purple/white colour that plants like for the long expanse of daytime, and then an orange sunset colour heading into the dark night. If you find that you are getting random rainbow colours on your string (or ring or array) of SPLEDs you probably need alternative control settings. Try to flip the DIP switch and restart the GPOD and see if that helps your bank of SPLEDs.

There are many control options beyond the two set up on the DIP switch. Your array may not be comprised of RGBW but rather only RGB ones, or perhaps a different setup all together. If your SPLEDs do not perform as expected, and the DIP switches do not help, you may need to adjust the software on the Arduino to call a different configuration. You can read more about other settings in the Adafruit* library documentation.

See more about changing the configurations assigned by DIP switches in the code section below.

Customizing Your GPOD

For advanced users, the GPOD device can be altered to achieve other lighting effects beyond the growl light function. The inner system software (shared on GITHUB) offers broad flexibility.

Electronics and computer programming skills are required for these other uses. You can learned much of that from various online resources. Search the internet for "Arduino."

Customizing your GPOD requires specific skills and abilities. The information below is offered for those with suitable knowledge and no guarantees or promises are made with regards to its accuracy.

Basic Operation

The primary goal of the GPOD system is to operate as a grow light, helping your plants and seedlings along in the dim winter months, or in a home without a bright southern window.

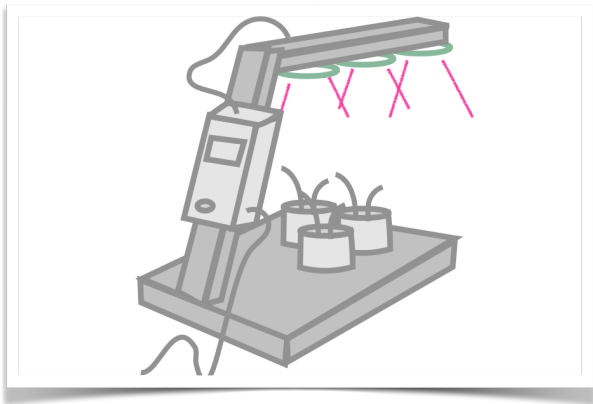
With a full GPOD system on hand you can proceed with the setup below. If you have a kit, you'll need to do some simple soldering to get the pieces in place. The assembly sheet provided with your kit will get you through those steps and ready for setup and operation.

GPOD Setup for a Grow Light - Make a Stand

Besides connecting the power and turning the unit on, you will also need a simple support structure of some kind for your SPLEDs.

The easiest setup is using a set of shelves. The illumination rings or arrays can be mounted under the shelf above your plants. With adjustable shelves, this is effective for adjusting with the growth of your plants or sprouting seedlings. A simple cardboard box as another option. Make a large hole in one side to allow viewing and watering access. The GPOD control box can simply sit beside the box or on top of it.

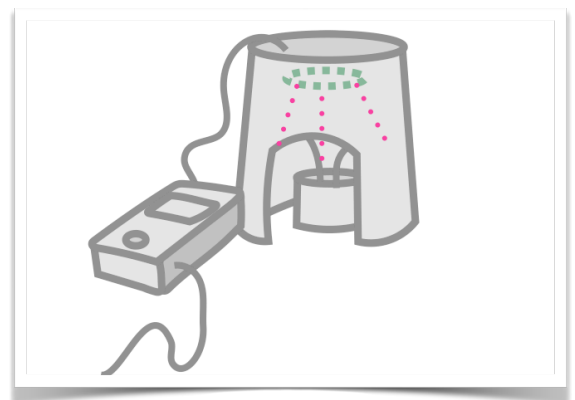
You'll want enough space below your illumination to allow for a growing plant. Begin with your seedlings or young plant on a block or two of wood to elevate them closer to the light. As they grow, you can remove a block and give them more vertical space. The light intensity is most beneficial to the plants when it's not too far away.



A simple wooden frame works well for a setup with many SPLEDs. The GPOD control box can be mounted to the upright supports of the structure for easy viewing and access.

An L-shaped support structure with a bolt and a wing-nut could work very well for adjusting the position and angle of the illumination. RGBW light arrays as ring or matrix or even linear sections would work well as the SPLEDs and would attach well to the horizontal structure.

For a very small, simple setup with just a single ring of maybe 12 or 16 SPLEDs you can use an inverted yoghurt container or coffee can as your light stand. Cutaway the sides for easy viewing and watering access. A hole punched, drilled or snipped in the bottom of the container can become your wiring access. You might want a few more to attach your SPLED ring with zip ties or twist ties, or even just string.



Power Up and Set the Time

Regardless of the setup you choose, you'll need to remove the snap-on back of the GPOD case by pulling it gently with your fingernails. Inside you'll see the green wiring blocks. Choose one to receive the wires for your connection. You should strip a few mm worth of wire and twist, maybe tin the bare wire with your soldering iron and some solder. Use a small screwdriver to attach the wires to the wiring block of your choice. Let's say the one labeled "SET_R" for example. Note the markings of SIG, 5V and GND on the board to align your wires. Ensure those connect to the same signals on your SPLED ring.

If you have more than one ring of SPLEDs, daisy-chain them together by connecting the same connections to the next set. You'll use the "Data Out" or "DO" pad on one ring or array to connect to the "Data In" or "DI" pad on the next one.

Now you can power up your GPOD. After the 'Hello' leaf shape, and the GPOD GrowLight message, you'll see the time set screen. Set the time to your desired 'current time' or if you're working at night, choose a midday time so you can ensure the illumination is working.

Once you've chosen the time, the system will proceed on its own. After a minute you'll see the current time of day, and the illumination should be activated. If you have all the pixels glowing with the same purple colour (for daytime, orange for sunset or sunrise) you're all set.

DIP Switches for SPLED Type

If you have a rainbow of colours, you'll want to try the DIP switch for your selected bank of LEDs (or just flip them all). To access the DIP switch, just turn off the power, pop open the back panel, and use a pencil tip to flip the desired switch. After adjusting, power up again and see if the process works better.

If you still have colour problems you'll need to either get a different type of SPLED or get into the system firmware to change the settings (see section below).

You can check the status of the DIP switches without removing the back by holding down the user button while turning on the power. The DIP switch positions (and firmware version) will be displayed for a few seconds before going on to the time-setting input screen again.

Time Adjustment

The specific timing of the on-off lighting isn't so important to your plants, but if you find yourself wanting to adjust for a drifting time of day, you can adjust the time at any point on the running system by pressing the button again. This returns you to the time-set screen where you can adjust the time. After adjusted, it will pop back into running state after a few seconds without any input.

Hardware Overview

The project hardware is comprised of the following elements:

| | |
|-----------------------------------|---|
| Arduino Nano (5.0V version) | System Power Switch SPDT |
| Printed Circuit board | Miniature 128 x 64 OLED Display & standoff |
| Wiring Terminal Blocks | Push-button switch & plastic key-cap |
| Power Supply Barrel Jack | 3D printed Plastic Enclosure & Lid |
| 4-position DIP Switch | 12-element SPLED ring (or user-provided ones) |
| 5Volt wall-plug power adapter | Mounting Screws (2pcs) |
| 3-conductor wire for SPLED hookup | Single-strand wire for switch hookup |

The Arduino Nano device needs to be programmed with the project software, which is (currently) shared on Github.

Wiring Channels and SPLED Arrays

The wiring terminal blocks allow you to connect SPLED arrays, strings, rings etc in four different banks. When the GPOD operates, all the channels or banks are activated, up to a total of 40 elements or pixels per channel.

The banks are marked on the circuit board as SET_S, SET_R, AUX_A, and AUX_B. For convenience we can refer to these as channels R, S, A and B.

Tip: if you have a strip and a ring that are of different signalling styles, using channel S for the strip and R for the ring helps you remember which channel is which.

Since current draw from many individual SPLED elements or ‘pixels’ can get very high, it’s recommended to keep the total usage below 40 pixels for the whole project. If you are not knowledgeable in measuring current consumption do not attempt to employ more pixels than this in total, across all channels.

You can achieve 40 elements on one channel, or maybe 20 or two, or 10 on each of the four, what ever you wish. If you are altering the firmware, and monitoring the current you can explore other configurations.

Check Mode on Start-up

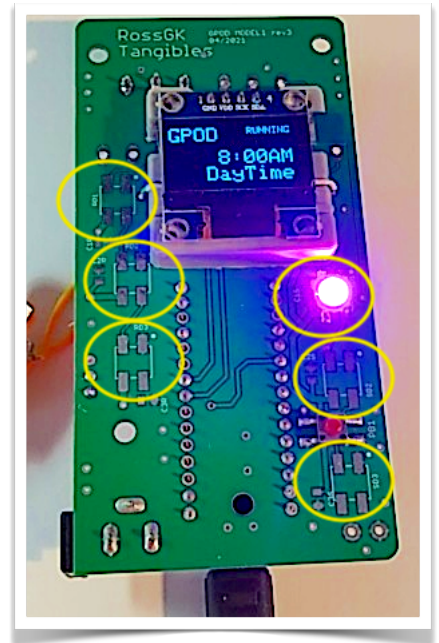
The DIP switches allow you to mix and match various types of NeoPixel or generic RGBW arrays on different channels. You can check the status of your DIP switches by holding the input button during power up, and you’ll see a DIP overview screen which times out after a few seconds.

OnBoard SPLED Testing - for Makers.

An added bonus on the PCB is that there are six sites for mounting an SPLED directly on the circuit board. While not practical for a grow-light application, or any other uses that come to mind, it is helpful for makers who are using the platform to learn about SPLEDs or developers trying to explore some additional features.

With these locations, you can solder one or several SPLEDs into place and work through your user interface with the display and button both accessible while you watch what your code changes do to the SPLED. There is also a spot in parallel to each pixel for a capacitor – some older SPLEDs that required additional decoupling for bug-free signalling.

In the picture at the right, the locations of the six SPLED mounting sites are shown. The three on the left are marked RD1 RD2 and RD3, and connected to channel SET_R. The right-side ones are SD1, SD2 and SD3 and connected to channel SET_S. There are no SPLEDs on the AUXA or AUXB channels.



Since there are three on each channel, you can do sequential effects and pixel indexing in your explorations.

Again though, these are not very useful with the board mounted in the enclosure, although you can see the glow through the plastic, if that helps.

A WARNING - Power Connections

As noted above, the external barrel-jack connector feeds 5Volts to the SPLEDs which may need a lot of power. Note that in normal operation this 5V line also powers the Arduino. Though not numbered on the Nano board, let's call it Nano pin 27 (it's beside pin A7).

The Arduino Nano's VIN un-regulated input (pin 30) is not used in this application, but if it were, the Nano's own regulator would *also* produce 5V on pin 27. Similarly, when power is supplied through a USB connector (e.g. when changing the firmware on the Nano) a 5v level is produced on pin 27 as well.

You should avoid using both the barrel jack and the USB connector at the same time, as an imprecise barrel adapter supply that is slightly high will sink power into the Nano which it might not like (though it may not be a problem in some situations).

While the GPOD board is in the enclosure, there's no risk of that, as the USB cannot be accessed (the case is in the way). However, when disassembled it is possible and you may risk damaging your Nano.

CAUTION: Hardware alterations or exceeding recommended configurations require an understanding of electronics and electricity. While this project is a helpful one for exploring those topics, beware that a novice can damage the hardware or cause fire risk by ignoring safety precautions or connecting components without understanding their functions and limitations.

Firmware Overview

The Arduino Nano has software or ‘firmware’ code which is loaded onto that device with the Interactive Arduino Development Environment (IDE). Editing and loading of the software requires familiarity with the C++ programming language, and the IDE setup and operation. While easily used by makers of all ages, there is a learning curve which requires some effort. Beginners are encouraged to visit the Arduino website to explore that ecosystem.

The GPOD firmware follows the standard Arduino project setup process of setup block and a loop block wherein all the functions of the device. There are a few differences relative to other projects a novice Arduino user might notice, such as the multi-file approach and the non-global scope of the key libraries.

The software uses a multi-file structure for ease of understanding and evolution. A user can simply open the code in the Arduino IDE and all the files will appear as tabs. The project files are as follows, where ‘x’ is the current version of the firmware.

| | |
|-------------------|---|
| GPOD_VxPROD | Project header, release notes and library calls |
| defines_GPODxP | Constants, globals, definitions and other setups |
| dispFuncs_GPODxP | Functions relating to the OLED display |
| lightFuncs_GPODxP | Functions relating to the operation of SPLED arrays and strings |
| mainloop_GPODxP | Main operating loop() |
| setups_GPODxP | Initialization setup() code and invocations |

Online tutorials for the Adafruit GFX library for SPLEDs and the SSD1306 OLED display driver library typically demonstrate their usage as global instantiations. Thus they are in a global scope such tutorials retaining memory space for the duration of operation. In the GPOD code they are used in a local scope, thus they are disposable local instances and associated resources are freed after usage to preserve memory for our other operations. This approach is necessary for operating both libraries in the limited memory footprint of the Arduino Nano.

While support and guidance for altering the GPOD firmware is beyond the resources we can provide, the following information will be helpful for a few tasks that might bring a user to wish to alter the firmware.

Number of SPLEDs per Channel Block

The firmware is set up for a maximum of 40 SPLEDs per channel. This is intended to let one set up a grow light with enough illumination on any one channel to make their plants happy. It is not recommended to use multiple channels loaded with many SPLEDs simultaneously without closely monitoring the current usage of the circuit. Many LEDs illuminated simultaneously could draw more power than the power supply can provide. This may cause overheating and damage to the device. Proceed

with caution and if you are not knowledgeable in measuring current consumption do not attempt to employ more than a total of 40 SPLEDs in your entire project without measuring the impact.

| Control / Function | Firmware File | Associated Variables or Functions |
|--------------------|----------------|--|
| SPLEDs per channel | defines_GPODxP | RBANK_PIXELS , SBANK_PIXELS , AUXA_PIXELS , and AUXB_PIXELS All are initially set to 40 in the code. Change only with caution, due to high-current risk with many pixels. |

Brightness of the SPLEDs

The SPLEDs are illuminated to full brightness for your plants, but the brightness is controlled by the signals that the Arduino Nano sends to the individual pixels. In initial power up there is a scanning through the pixels and as initial illumination begins there is a two step brightening. The intensity is controlled by some constants.

| Control / Function | Firmware File | Associated Variables or Functions |
|------------------------------|----------------|--|
| Maximum brightness of SPLEDs | defines_GPODxP | HIGH_BRITE - values 0 to 255 (coded to 240) See also MED_BRITE , and LO_BRITE . |

Timing of the Sunrise, Daytime, Sunset Illumination Cycles

The GPOD device uses a rudimentary system clock to determine on and off times for the illumination. The time is set by the user upon power up of the circuit. With each press of the button, the clock advances 15 minutes or 0.25 of an hour. Note that the clock is not a precise function and will drift a little over the space of days and weeks.

While aligning the timing with your actual sunrise and sunset isn't necessary, many will use the growlight to supplement their natural lighting, so having it roughly aligned is perhaps of value.

The sunrise timing begins at 5:00 am, the daytime phase begins at 7:00 am. Sunset begins at 7:00 pm (19:00 hrs) and it's full darkness for night time at 9:00pm (21:00hrs). These variables are expressed as floating point numbers, as follows:

| Control / Function | Firmware File | Associated Variables or Functions |
|-----------------------------|----------------|--|
| Button press time increment | defines_GPODxP | <code>TINCREM</code> - set to 0.25 for 15min advance per press |
| Time-of-day phases | defines_GPODxP | <code>SUN_UP</code> - 5.0 aka 5:00 am <code>LITE_TM</code> - 7.0 <code>SUN_DN</code> - 19.0 <code>DARK_TM</code> - 21.0 |

Configuration of SPLED Types

The SPLEDs may be of a wide variety of types, depending on where you purchased yours. They may be RGB or RGBW or more exotic ones will have GRB colour signalling, or different frequencies of communications. The Adafruit library is beyond the scope of this project, and is not associated with the GPOD project. Users are encouraged to visit Adafruit and explore the library at their leisure. There's a lot of complexity there, so the GPOD is just setup to handle two situations. If one of the DIP settings works for your SPLEDs then you need not explore changes to the configuration.

If you do wish to explore these, you can find the alternative settings for the DIP switch ON and OFF positions as follows:

| Control / Function | Firmware File | Associated Variables or Functions |
|--|-------------------|--|
| DIP switch selections for SPLED signalling | lightFuncs_GPODxP | <p>In the <code>makeFlags()</code> routine</p> <p>Set the <code>pixelFmt</code> variable for <code>PGMDIP_ON</code> or <code>PGMDIP_OFF</code> with the control keywords that are recognized by the Adafruit GFX library.</p> <p>Initial setup:</p> <pre>PGMDIP_ON = NEO_KHZ800 + NEO_RGB; PGMDIP_OFF = NEO_KHZ800 + NEO_RGBW;</pre> |

Configuration of SPLED Illumination colour

Colours for the grow light application are chosen pseudo-scientifically. The bulk of the time, the illumination is a purple colour that was chosen to be near the optimal frequency for making plants happy. One sees this colour when viewing grow lights on the International Space Station or laboratory conditions. It is likely however that some plants want a different illumination spectrum.

It's also not clear if plants need a day-night cycle, but some sources suggest it is a good idea, and it seems intuitively like they might like it, so that's how the GPOD is set up.

Finally sunrise and sunset colour may not have any value to plants, but it seems like a fun thing to do for them. It does emulate natural conditions for plants in the wild, so perhaps it is useful? Maybe you can

experiment to find out if that is the case. If nothing else, it tells the humans around that the phases of the illumination program are changing and adds a bit of mood lighting to the operation.

With these ideas in mind, an advanced user may wish to play with the colour spectrum. Remember that the colour signalling is affected by the configurations indicated in the `makeFlags()` routine described above, so some fiddling may be required. In the simplest cases, changes to the RGBW values in the table below will be sufficient.

| Control / Function | Firmware File | Associated Variables or Functions |
|------------------------|--------------------------------|---|
| SPLEDs colour settings | <code>defines_GPODxP</code> | <p><code>PURP_CLR</code> - the purple daytime illumination colour</p> <p><code>PGOLD_CLR</code> - a warm gold for sunrise use</p> <p><code>PRD_CLR</code> - a red sunset colour</p> <p><code>POFF_CLR</code> - not really a colour, just all off for night time</p> <p>Adjust RGB values for other colours if desired</p> |
| Application of colours | <code>lightFuncs_GPODxP</code> | The routine <code>lightMeUp()</code> applies the colours created in the <code>defines_</code> file to the SPLEDs on various channels. |

Disclaimer

The GPOD software, device and the RossGK Tangibles project are not affiliated with Adafruit nor Arduino, and we offer no guarantee about the availability of the code libraries nor their efficacy for your application, nor the functions of this device as a plant grow light.

This system and the associated information contained herein are offered purely as an experimental hobby-oriented platform for home use. The suitability of any of this for your application or conceived use is solely at the discretion of the user. No user-support or guarantees are offered or implied through the sharing of the information in the documentation. The accuracy of information provided heron is similarly not guaranteed and subject to change at any time. It's the Internet – things gonna change, be inaccurate or just plain wrong.

RossGK Tangibles

