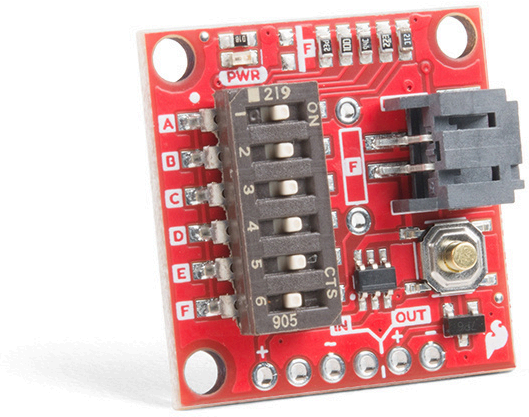
TPL5110 Nano Power Timer Hookup Guide

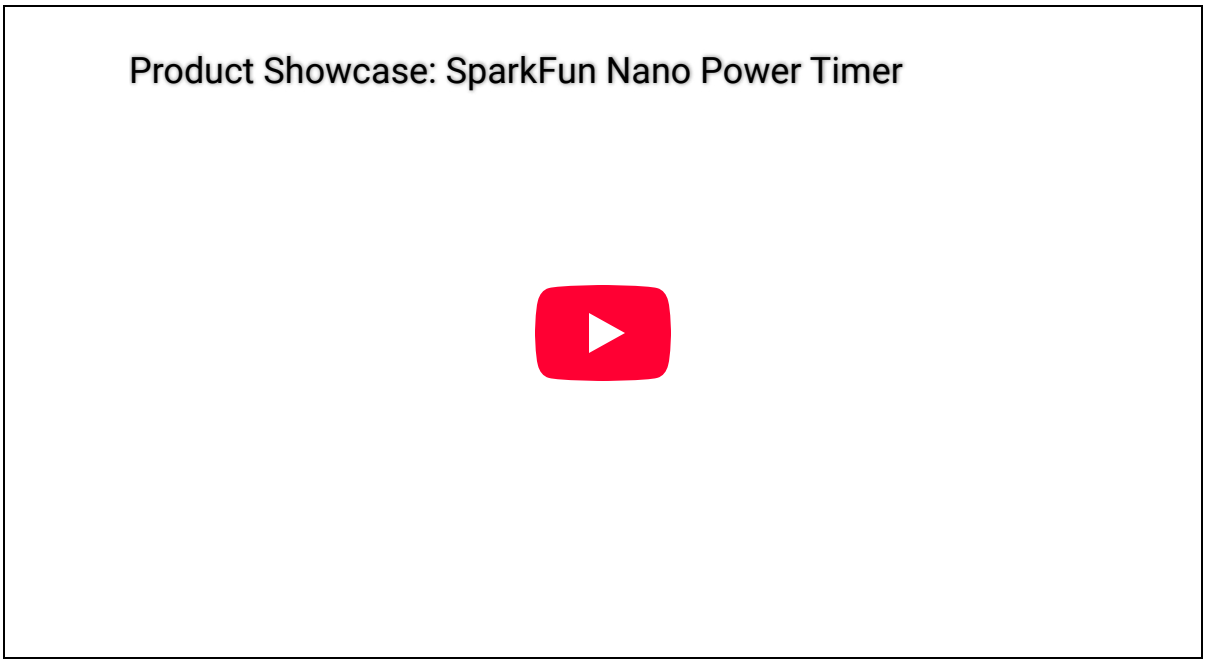
CONTRIBUTORS: *ELIAS THE SPARKIEST*

Introduction

The TPL5110 Nano Power Timer is ideal for applications that require low power, and especially those projects that are running off of a LiPo battery. The Nano Power Timer will turn on your project, most likely a microcontroller, after the set amount of time, continuously. When your microcontroller has completed whatever needs doing, sampling air quality for example, it can then signal back to the Nano Power Timer to turn it off. While the project is off, the Nano Power Timer will only consume 35nA of power until the timer turns the project back on again. In this tutorial, we'll discuss how the time is set with the on board six DIP switch and use a microcontroller to turn it off when a task is finished.



SparkFun Nano Power Timer - TPL5110
PRT-15353
$8.58



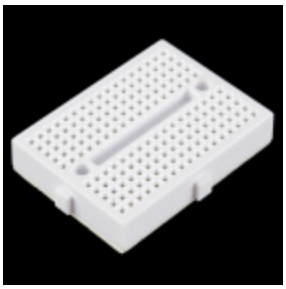
Required Materials

To follow along with this tutorial, you will need the following materials. You may not need everything though depending on what you have. For example, I chose the RedBoard as a simple demo, but you could use any microcontroller. Add it to your cart, read through the guide, and adjust the cart as necessary.

Lithium Ion Battery - 400mAh
PRT-13851
$7.98

Jumper Wires - Connected 6" (M/M, 20 pack)
PRT-12795
$2.95

Break Away Headers - Straight
PRT-00116
$1.95



Breadboard - Mini Modular (White)
PRT-12043
$5.25

SparkFun RedBoard Turbo - SAMD21 Development Board
DEV-14812
$22.50

SparkFun Traveler microB Cable - 1m
CAB-14741
Retired

Tools

You will need a soldering iron, solder, general soldering accessories, and a diagonal cutter.

Soldering Iron - 60W (Adjustable Temperature)
TOL-14456
$25.95

Solder Lead Free - 15-gram Tube
TOL-09163
$4.95

Suggested Reading

If you aren't familiar with the following concepts, we recommend checking out these tutorials before continuing.

Pull-up Resistors

A quick introduction to pull-up resistors - whey they're important, and how/when to use them.

How to Solder: Through-Hole Soldering

This tutorial covers everything you need to know about through-hole soldering.

How to Power a Project

A tutorial to help figure out the power requirements of your project.

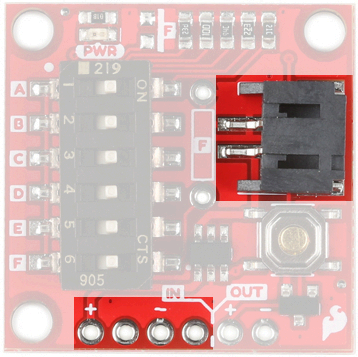
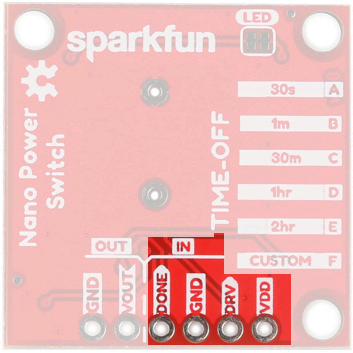
Button and Switch Basics

A tutorial on electronics' most overlooked and underappreciated component: the switch! Here we explain the difference between momentary and maintained switches and what all those acronyms (NO, NC, SPDT, SPST, ...) stand for.

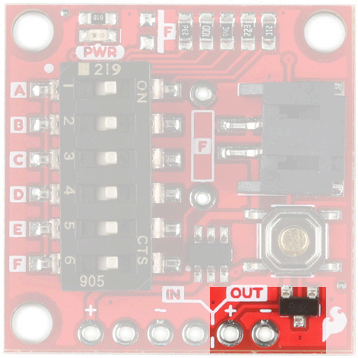
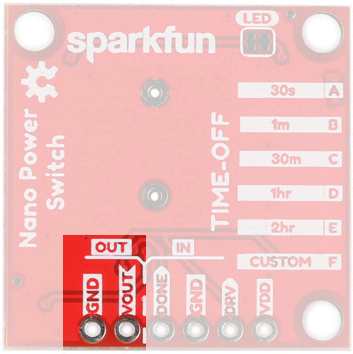
Hardware Overview

Power and LiPo Battery

The Nano Power Timer can handle voltages between **1.8V - 5.5V** and current up to **1.1 Amps**. There are two options when connecting power to the Nano Power Timer. The first and more obvious option is the on board LiPo Battery Connector. The second is the `VDD` and `GND` pins on the five pin header underneath the `IN` label (short for **INPUT**). The power you provide to the **INPUT** side will flow out the **OUTPUT** side to power your microcontroller or project. If your using a LiPo battery, then **do not** attach another power source to these pins.

	
Top	**Bottom**

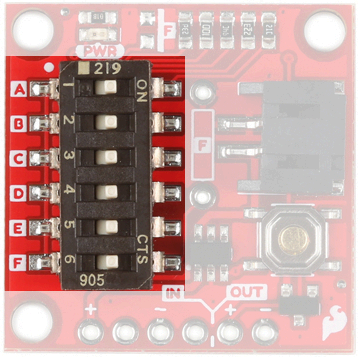
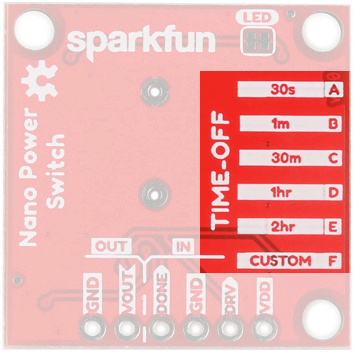
The `OUT` (**OUTPUT**) power is labeled `VDD_OUT` and `GND`. These pins will go to the board or project that you are powering.

	
Top	**Bottom**

Timer and Delay Switch

> **Note:** As reported in the datasheet, the resistance corresponding to the given time has a certain margin of error. The example given in the datasheet is a desired 600 second time which would require a resistance that falls in the window: 56.96kΩ-57.44kΩ. The resistor values on the board have a 1% tolerance as well. The tolerance of the resistors and margin of error makes it imprudent to try and use this product for **HIGH** precision applications.

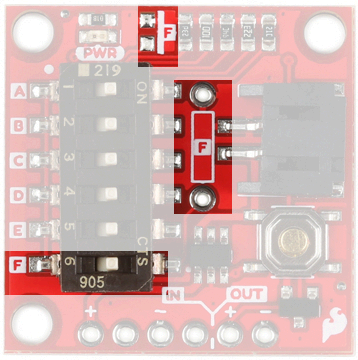
The Nano Power Timer's main function is to turn your microcontroller *on* after a set amount of time, continuously. The board has a six DIP switch that controls this time by changing resistance to the timer pin on the IC. To the left of the switch are five letters labeling each switch. On the underside of the board are these letters and the amount of time that is set when the corresponding switch is flipped.

	
Top	**Bottom**

This time is **set** when the board is **powered up**, so cycle power after you select your time. Check the table below to find a time suitable for your project.

Timer	Switch	Resistance
30 s	A	16.2 kΩ
1 min	B	22 kΩ
30 min	C	93.1 kΩ
1 hr	D	124 kΩ
2 hr	E	169 kΩ
Custom	F	Custom

If you're time is not listed then we've left two additional pads for a custom time: one SMD and the other PTH. These two spaces are labeled with F on the product and their corresponding switch is labeled the same. If you decide to use them both at the same time their resistance is in parallel so make sure to calculate accordingly.



Note: Regardless of how long your application takes to complete what it needs to do, the Nano Power Timer will *always* turn the board on at the chosen interval. For example, a two second blink of an LED will not delay a 10 second interval by two seconds (making the delay 12 seconds). Instead the LED will turn back on after 8 seconds because the two second delay cuts into the 10 second interval.

More Timer Options

You are *not* limited to the times that are represented on the silk. Ignoring the `Custom` switch option, there are **26 possible combinations** of switches aside from the five printed on the board that yield times from three seconds up to 15 minutes. The chart below gives you the combination of **ON** resistors, their combined resistance, and their approximate time. A few redundant resistances were ommited.

Timer	Switch Combo	Resistance
2-3 s	A+B+C+D+E	7.579 kΩ
3-4 s	A+B+C+D	7.933 kΩ
4 s	A+B+C	8.470 kΩ
5 s	A+B+E	8.844 kΩ
6 s	A+B	9.329 kΩ
10 s	A+C+D+E	11.563 kΩ
~12 s	A+C+D	12.407 kΩ
~13 s	A+C+E	12.742 kΩ
~15 s	A+D+E	13.225 kΩ
~18 s	A+C	13.774 kΩ
~19 s	B+C+D+E	14.243 kΩ
20 s	A+D	14.341 kΩ
~22 s	A+E	14.790 kΩ
~25 s	B+C+D	15.546 kΩ
~28 s	B+C+E	16.075 kΩ
32 s	B+D+E	16.852 kΩ
35 s	B+C	17.754 kΩ
40 s	B+D	18.707 kΩ

~45 s	B+E	19.479 kΩ
~5 min	C+D+E	40.400 kΩ
8 min	C+D	52.995 kΩ
~12 min	C+E	59.694 kΩ
15 min	D+E	72.033 kΩ

Some of these times are approximate, but I have provided a Python script in the resources below to help you calculate what your exact time is. This will differ by small margins for each board due to the tolerance of the resistors.

Additional Timer Options From Datasheet

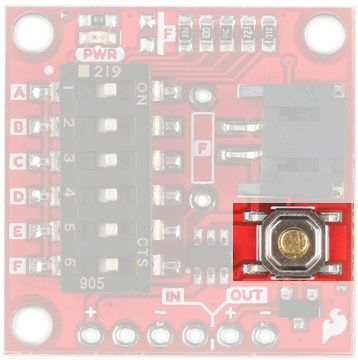
Below is the list of available times from the datasheet that are *not* included in the charts above. Of note are the timer options in the millisecond range. You can access these times by taking advantage of the two empty resistor pads on the Nano Power Timer.

Timer	Resistance
100 ms	500 Ω
200 ms	1000 Ω
300 ms	1500 Ω
400 ms	2000 Ω
500 ms	2500 Ω
600 ms	3000 Ω
700 ms	3500 Ω
800 ms	4000 Ω
900 ms	4500 Ω
1 s	5.20 kΩ
2 s	6.79 kΩ
7 s	9.71 k Ω
8 s	10.18 kΩ
9 s	10.68 kΩ

30 s	16.78 kΩ
50 s	20.047 kΩ
2 min	29.35 kΩ
3 min	34.73 kΩ
4 min	39.11 kΩ
5 min	42.90 kΩ
6 min	46.29 kΩ
7 min	49.38 kΩ
9 min	54.92 kΩ
10 min	57.44 kΩ
20 min	77.57 kΩ
40 min	104.67 kΩ
50 min	115.33 kΩ
1 hr 30 min	149.39 kΩ

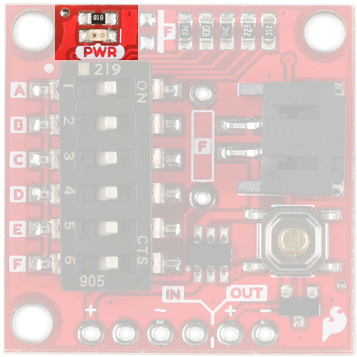
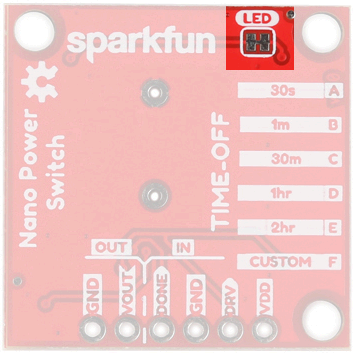
Push Button

The push button on the product will manually begin the timer. This allows you to test the timer and your project which should send an **OFF** signal to the `Done` pin.



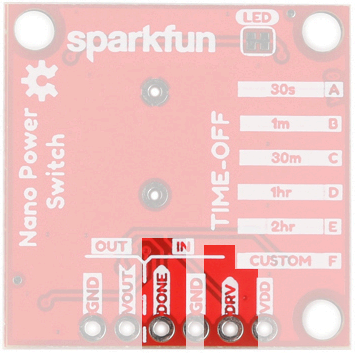
LED

There's a single red power LED on the topside of the product which indicates power is being supplied to the `OUT` pins. If you want to disconnect this LED, simply cut the trace between the jumper on the underside of the board labeled `LED`.

	
Power LED	**Power LED Jumper**

Header

Next to `GND` is the `DONE` pin that tells the product to turn your microcontroller or project off. To do this, your microcontroller or project will need to send a digital signal from **LOW** to **HIGH** to this pin. Finally, the `DRV` pin is active high and will start the timer of your project when the pin receives a **HIGH** signal. This is the same function as the on board push button. Not very many people will want to attach a different button, but if you wanted to, this would be the place.



How Do I Check the Exact Time?!

I've provided a simple Arduino sketch below to calculate the exact timing of the timer setting. You'll attach the VOUT pin to a digital I/O pin on your microcontroller and the sketch will start its timer when the power is high and end the timer when the power is off.

Nano Power Timer Checker (INO)

Hardware Hookup

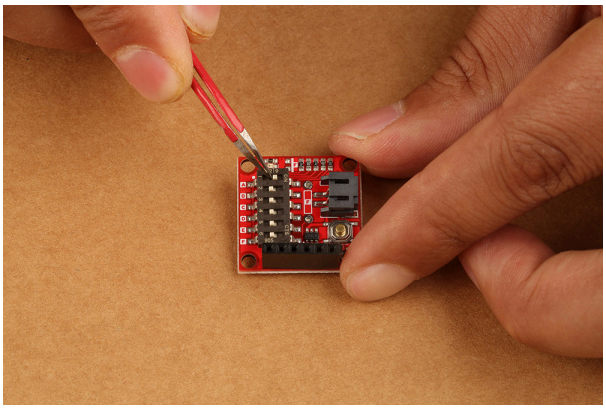
⚡ **Warning!** When powering a microcontroller to the Arduino via USB, you will need to disconnect the TPL5110 from the Arduino's power input. The conflicting power sources will damage the TPL5100. You may want to consider adding a Schottky diode betwen the TPL5110's output voltage and the microcontroller's voltage input.

Schottky Diode
COM-10926
$0.30

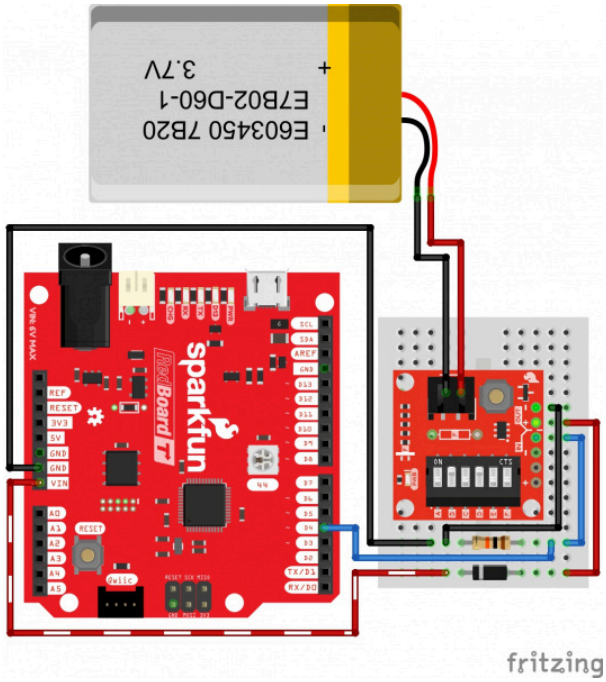
Note: Depending on your microcontroller, you may need to add a pull-down resistor on the Done pin. We found that the SAMD21 and SAMD51 boards required a pull-down resistor to trigger the TPL5110's Done pin reliably.

Resistor 10K Ohm 1/4 Watt PTH - 20 pack (Thick Leads)
PRT-14491
$1.50

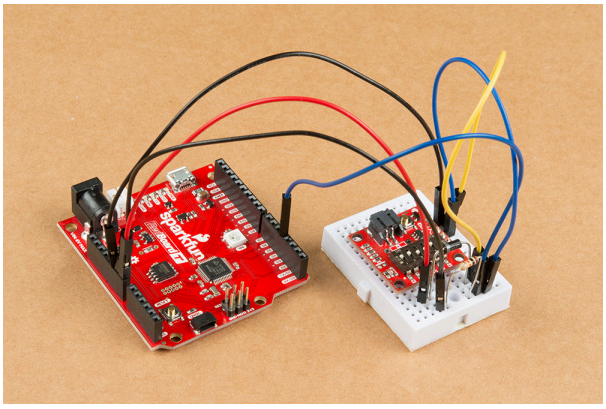
We're going to power a RedBoard Turbo with a 3.7V LiPo Battery and set it to a 14 second delay. By default, the board comes with every switch flipped to the **ON** position which is 3 second timer. To set the timer to 14 seconds, we'll turn some switches to the *ON* position and others to the *OFF* position. I used tweezers because the switches were too small for my hands.



To get a 14 second delay switches 'A' + 'D' + 'E' must be flipped **ON**, and the *other* swtiches flipped **OFF**. Next solder a 6 pin header of your choice to the Nano Power Timer. After the six pin header is soldered to the Nano Power Timer, plug in three wires into the female header as follows:

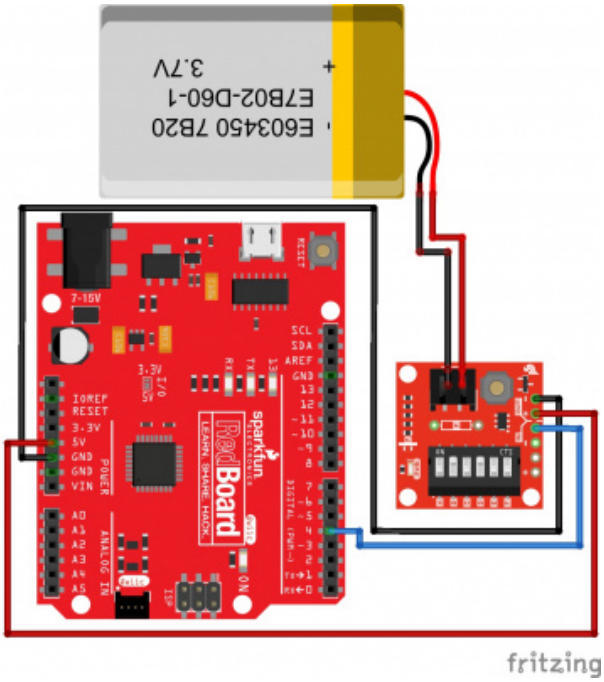


We'll use a Schottky diode to protect the TPL5110 when the RedBoard Turbo is connected to a computer when prototyping and uploading code. Additionally, we'll add a pull-down resistor on the Done pin. Depending on your microcontroller, you may not need resistor. After wiring your circuit together, you may have something similar to the image below without the battery connected.



Let's move onto the code.

Note: If you are looking to reduce the number of components used with the TPL5110 and a microcontroller, you could solder wire directly to the PTH pins, remove the pull-down resistor, and Schottky diode. Just make sure to disconnect the VOUT pin whenever you are connecting a USB cable to your microcontroller to upload.



Simple Example

Note: If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE. If you've never connected an SAMD21 device to your computer before, you will need to install the board add-on and may need to install drivers. Check out our section on UF2 Bootloader and Drivers and Setting Up Arduino for help with the installation.

With this example we'll be laying out the very basics of how the Nano Power Timer works. Copy the code and paste in the Arduino IDE. Select your board (in this case, the **RedBoard Turbo**), COM port, and hit the upload button to upload in the Arduino IDE.

The Nano Power Timer is powered by a LiPo Battery and will turn on a RedBoard Turbo every 14 seconds. The RedBoard Turbo will blink it's blue LED, and then send a *done* signal back to the Nano Power Timer, which will turn off the RedBoard Turbo.

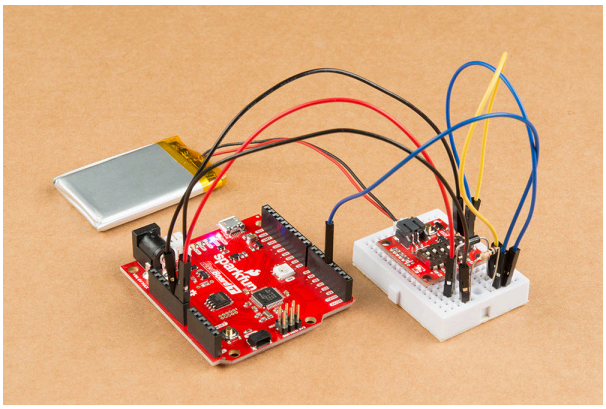
Regardless of how long your application takes to complete what it needs to do, the Nano Power Timer will *always* turn the board on at the chosen interval. For this simple example, the one second blink of the LED does **NOT** impact the 14 second interval.

```
/*
TPL5110_Blink_Demo_example.ino

  Simple Example Code for the TPL5110 Nano Power Timer Hookup Guide. Th
is code
  simply blinks the pin 13 LED and writes pin 4 (donePin pin) high. Thi
s shift from
  LOW to HIGH of the donePin pin, signals to the Nano Power Timer to tu
rn off the
  microcontroller.
  SparkFun Electronics
  Date: May, 2019
  Author: Elias Santistevan
*/

int led = 13; // Pin 13 LED
int donePin = 4; // Done pin - can be any pin.

void setup(){

  pinMode(led, OUTPUT);
  pinMode(donePin, OUTPUT);

}

void loop(){
  // Blink.
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);

  // We're done!
  // It's important that the donePin is written LOW and THEN HIGH. This
shift
  // from low to HIGH is how the Nano Power Timer knows to turn off the
  // microcontroller.
  digitalWrite(donePin, LOW);
  digitalWrite(donePin, HIGH);
```

```
  delay(10);
}
```

After uploading this code, and plugging in the LiPo battery into the *Nano Power Timer*, the RedBoard Turbo will blink once, be turned off by the Nano Power Timer, and then will turn on again 12 seconds later (14 second timer - 2 second delay in sketch).

Note that the Power LED on the Nano Power Timer turns on when it provides power to your microcontroller.

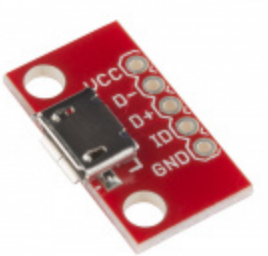


This Nano Power Timer really shines when you're doing remote projects that are running off of battery and you need to maximize the life of the battery! There's only so much deep sleeping that you can do in code, and nothing will compare to 35nA of power consumed in the off state of the Nano Power Timer. You just need one additional GPIO or some method of sending a digital signal that can go from **LOW** to **HIGH** to signal **OFF** to the Nano Power Timer.

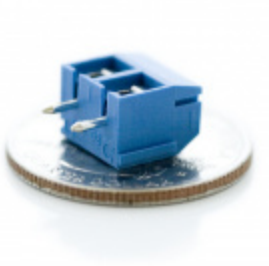
Note: Don't forget! We've also broken out **INPUT** power pins, so you're not limited to just a LiPo battery. Check out the list below for other power connectors for additional power options (not a complete list)!

SparkFun USB-C Breakout - Horizontal
BOB-15100
$4.95

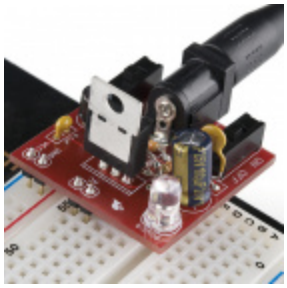
DC Barrel Jack Adapter - Female
PRT-10288
$3.75



SparkFun microB USB Breakout
BOB-12035
$2.95



Screw Terminals 5mm Pitch (2-Pin)
PRT-08432
$1.10



SparkFun Breadboard Power Supply 5V/3.3V
PRT-00114
$12.50

Resources and Going Further

Now that you've successfully got your TPL5110 Nano Power Timer up and running, it's time to incorporate it into your own project! For more information, check out the resources below:

- Schematic (PDF)
- Eagle Files (ZIP)
- Datasheet (PDF)
- GitHub Repo
    - /Firmware/Arduino
        - Nano Power Timer Checker Sketch
        - TPL5110 Blink Done Example Sketch
    - /Firmware/Python
        - Resistor Combinator Calculator
- SFE Product Showcase

Need some inspiration for your next project? Check out some of these related tutorials:

Reaction Timer

Demonstrate mental chronometry with this simple reaction timer!

Sunny Buddy Solar Charger V13 Hookup Guide

How to hookup the Sunny Buddy: a solar-powered, MPPT (peak-power tracking), LiPo battery charger.

SparkFun Inventor's Kit for Photon Experiment Guide

Dive into the world of the Internet of Things with the SparkFun Inventor's Kit for Photon.

Adding a Timed Button to a Project

This tutorial will walk you through making a timed power controller for interactive projects. You will learn how to add an on button that will provide power to your project for an amount of time and then turn off again.