Gator:starter ProtoSnap Hookup Guide

Introduction

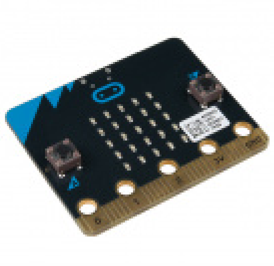
The gator:starter is one of a series of gator-clippable boards called gator:boards that SparkFun has created to interface with the micro:bit and gator:bit expansion for micro:bit. The gator:starter is a nice contained package of a temperature sensor, an RGB LED, and light sensor. In this hookup guide we'll go over how to hook up each of the individual boards, along with some examples involving all of the boards together.

SparkFun gator:starter ProtoSnap

SEN-14891

Required Materials

For this activity, you’ll of course a micro:bit. You’ll also need some alligator clips to connect everything together, and a micro-B USB cable to program your micro:bit. All of these things are shown below, so grab them if you haven’t already. The gator:bit is recommended if you’d like to add some more input and output to your micro:bit, but you’ll be able to get along fine with just a micro:bit.



micro:bit Board

○ DEV-14208

SparkFun gator:bit

⊖ DEV-14484

Alligator Test Leads - Multicolored (10 Pack)

◉ PRT-12978

USB Micro-B Cable - 6"

◉ CAB-13244

Suggested Reading

If you decide to use the gator:bit and it's your first time using the board, check out the gator:bit Hookup Guide.

Gator:bit Hookup Guide

MAY 10, 2018

Gator:bit is a rugged development breakout board for the BBC micro:bit. Gator:bit exposes almost every pin on the micro:bit to an alligator clippable pad. Gator:bit features over-voltage, over-current, and short circuit protection circuitry on every IO pin as well as built-in addressable LEDs, built-in buzzer, and a power management solution to give you sensible powering options.

Also, if you're starting out with electronics and aren't familiar with the following concepts, we recommend checking out these tutorials before continuing.

What is a Circuit?

Every electrical project starts with a circuit. Don't know what a circuit is? We're here to help.

Voltage, Current, Resistance, and Ohm's Law

Learn about Ohm's Law, one of the most fundamental equations in all electrical engineering.

What is Electricity?

We can see electricity in action on our computers, lighting our houses, as lightning strikes in thunderstorms, but what is it? This is not an easy question, but this tutorial will shed some light on it!

Light-Emitting Diodes (LEDs)

Learn the basics about LEDs as well as some more advanced topics to help you calculate requirements for projects containing many LEDs.

Analog vs. Digital

This tutorial covers the concept of analog and digital signals, as they relate to electronics.

Getting Started with the micro:bit

The micro:bit is a compact, powerful programming tool that requires no software installation. Read on to learn how to use it YOUR way!

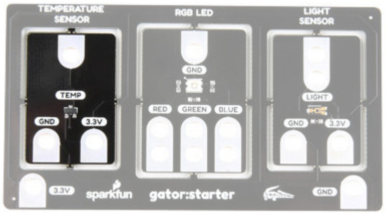
Hardware Overview

The gator:starter contains 3x boards that can be snapped out of the main assembly simply by twisting back and forth repeatedly. The available boards are:

- temperature sensor (MCP9700)
- common cathode RGB LED
- light sensor (TEMT6000)

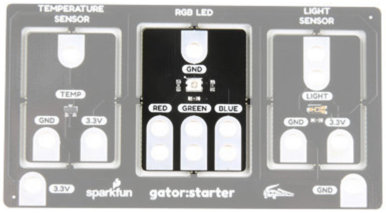
Temperature

The temperature sensor (MCP9700) is an analog sensor. Simply provide power to the sensor on 3.3V and GND. To read, you would connect the TEMP pin to the micro:bit's analog input.



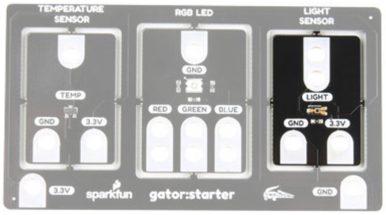
Common Cathode RGB LED

The center component is a common cathode RGB LED. Each individual LED is connect to the GND pin. You would need one I/O pin per color to light up.



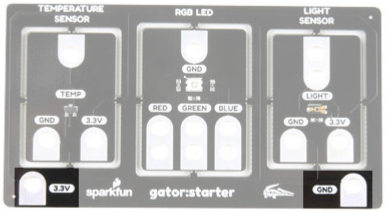
Light Sensor

The light sensor is also an analog sensor. Simply provide power to the sensor on the 3.3V and GND pin. To read, you would connect the LIGHT pin to a micro:bit's analog input.



Power Tabs

We've broken out the power (3.3V and GND) so it can be clipped to on the edge of the board for power without breaking each board away from the main board. These power pins are outlined in the image below.



Power Tabs

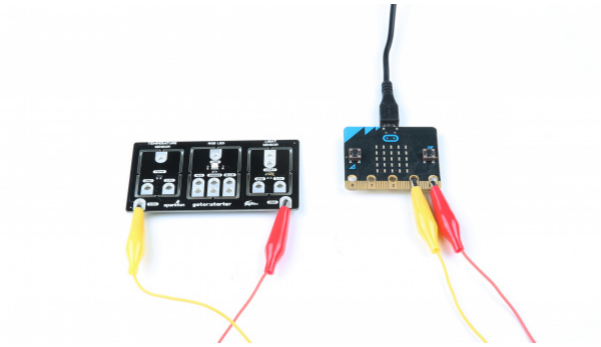
There are also power tabs available on each board if you choose to break the gator:starter into its basic components. Try not to hook power up backwards! If you don't break the board apart however, you'll still be able to interface with all of the sensor and light pins.

Hardware Assembly

Connecting the gator:starter to the gator:bit is a relatively simple process that doesn't even involve any soldering, all you'll have to do is clip onto the tabs to make your connection. Let's start by powering the gator:starter.

Power

When connecting power there are a few practices you should get used to doing. The first of these is to always connect ground (GND) first. Second, you want to make sure that you aren't powering your device with the wrong voltage, as powering a 3.3V device with 5V could damage the device. Keeping these things in mind, go ahead and clip the ground of your gator:board to ground on the gator:bit. Once this is done, go ahead and do the same with 3.3V. Now that power is connected, we can go ahead and start connecting to our sensors.



Connected Power

Signals

Temperature Sensor

The temperature sensor onboard is an MCP9700 analog temperature sensor with a temperature range of -40°C to 125°C. This sensor outputs an analog signal so you'll need to connect it to a pin on the gator:bit that is capable of reading an analog signal. The analog capable pins are pins **0**, **1** and **2**. Go ahead and clip the **TEMP** pin to any of those three pins.

RGB LED

The RGB LED is three LED's (red, green and blue) all combined into a single package. It's a common cathode LED which means that all three of the LED's share the same ground pin. To activate any color, simply make sure ground is connected, then connect 3.3V up to the color you'd like to turn on. Colors can be added and mixed simply by connecting more than one color. If you want to create even more colors, you can connect the LED to any of pins **0**, **1** and **2** to allow for things like dimming and full color control.

> ⚡ **Using with the gator:bit:** If you are using the gator:bit, keep in mind that you'll have to use pins P20, P19, P15, P14 or P13 to drive your LEDs, as the protection circuitry on the rest of the gator:bit pins does not create a high enough voltage to drive some colors of LED.

Light Sensor

The light sensor onboard is a TEMT6000 ambient light sensor. This sensor is also an analog sensor, so once again we'll only be able to use it with pins **0**, **1** and **2**. To connect this sensor, just clip the **LIGHT** tab to any of the three analog pins.

Using MakeCode

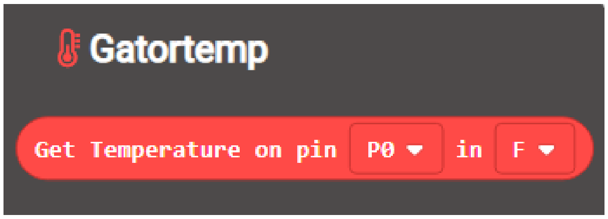
So you've connected your sensors and LED to the gator:bit, but how do we get useful information from these sensors? Luckily, SparkFun has written a few MakeCode packages to allow the gator:bit to make sense of the values that come in from the MCP9700 temperature sensor and TEMT6000 light Sensor.

Installing the Gatortemp Package for Microsoft MakeCode

To download the MakeCode package for the light sensor, go to **Advanced** -> **Add Package**. Then copy and paste the following link into the search bar:

```
https://github.com/sparkfun/pxt-gator-temp
```

Clicking on the **gator:temp** will add it to your list of usable packages in **Advanced**. Go ahead and open the package and you'll see a block that allows you to get the temperature on the pin that the gator:temp is connected to.

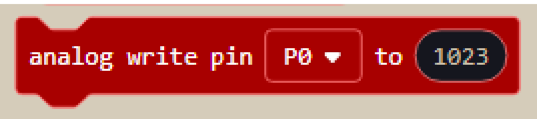


```
Get Temperature on pin P0 ▾ in F ▾
```

gator:temp Block

RGB LED

With the three primary colors, we're able to mix colors to create any color we like. LED's are either on or off, with no brightness in between, so we control brightness by turning the light on and off very fast in a process known as pulse width modulation (PWM). The common cathode LED does not require any additional MakeCode packages. To use PWM, go to `Advanced` -> `Pins` and select the **analog write** block, shown below.



```
analog write pin P0 ▾ to 1023
```

Analog Write Block

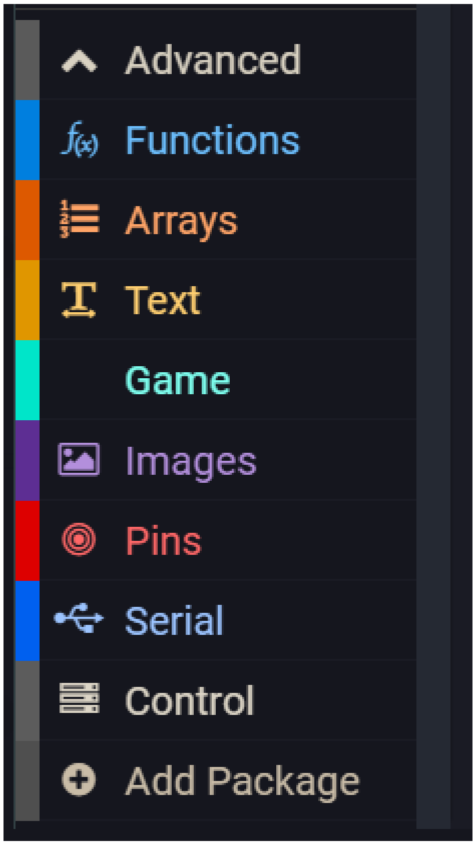
You'll need three of these blocks, one to control each color of the LED. Make sure the pin selected in the block is connected to the proper color (`P0` = Red, `P1` = Green, `P2` = Blue). The analog write block defaults to a value of `1023` , or 100%. If you wanted to turn the light down to 50% brightness, you'd plug a value of 511 into your analog write function. This allows for really any color to be created. For instance orange would be created by writing 100% (1023) to red and 10% (102) to green.

Installing the Gatorlight Package for Microsoft MakeCode

To download the Makecode package for the light sensor, go to **Advanced**->**Add Package**. Then copy and paste the following link into the search bar:

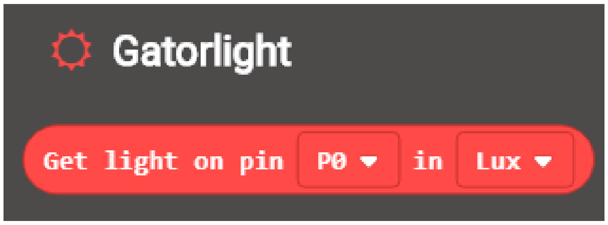
```
https://github.com/sparkfun/pxt-gator-light
```

Clicking on the **gator:light** will add it to your list of usable packages in **Advanced**.



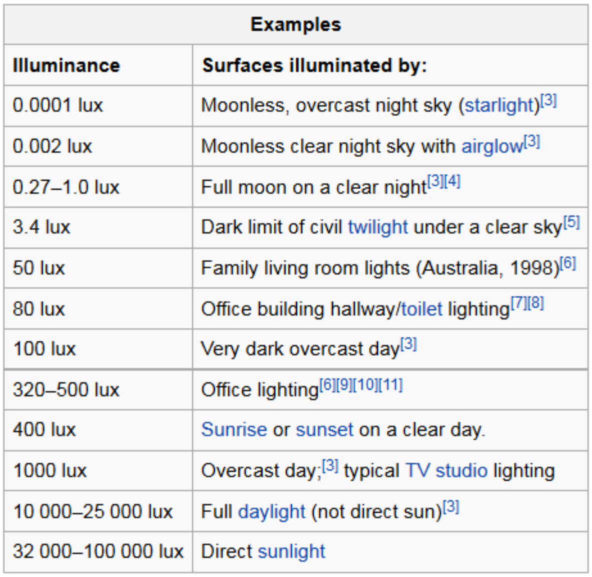
Adding a Package

Go ahead and open the package to take a look at the variable block we are now able to use. There are two drop down menus that you can use to change which pin the light sensor is attached to as well as the units of the measurement.



Get Light Block

The light sensor measures light in terms of illuminance, or lux. If you're unfamiliar with illuminance, it's basically just a measure of the amount of light reaching a certain spot. The units can either be the raw value from the analog-to-digital converter (ADC) or in units of lux. Check out the chart below to get an idea for what this lux value corresponds to in terms of some common lighting situations.



Examples	
Illuminance	**Surfaces illuminated by:**
0.0001 lux	Moonless, overcast night sky (starlight)[3]
0.002 lux	Moonless clear night sky with airglow[3]
0.27–1.0 lux	Full moon on a clear night[3][4]
3.4 lux	Dark limit of civil twilight under a clear sky[5]
50 lux	Family living room lights (Australia, 1998)[6]
80 lux	Office building hallway/toilet lighting[7][8]
100 lux	Very dark overcast day[3]
320–500 lux	Office lighting[6][9][10][11]
400 lux	Sunrise or sunset on a clear day.
1000 lux	Overcast day;[3] typical TV studio lighting
10 000–25 000 lux	Full daylight (not direct sun)[3]
32 000–100 000 lux	Direct sunlight

Table of Lux Values Courtesy of Wikipedia: Lux

This block can be used to modify the value of a variable in MakeCode.

Examples

Simple Thermometer

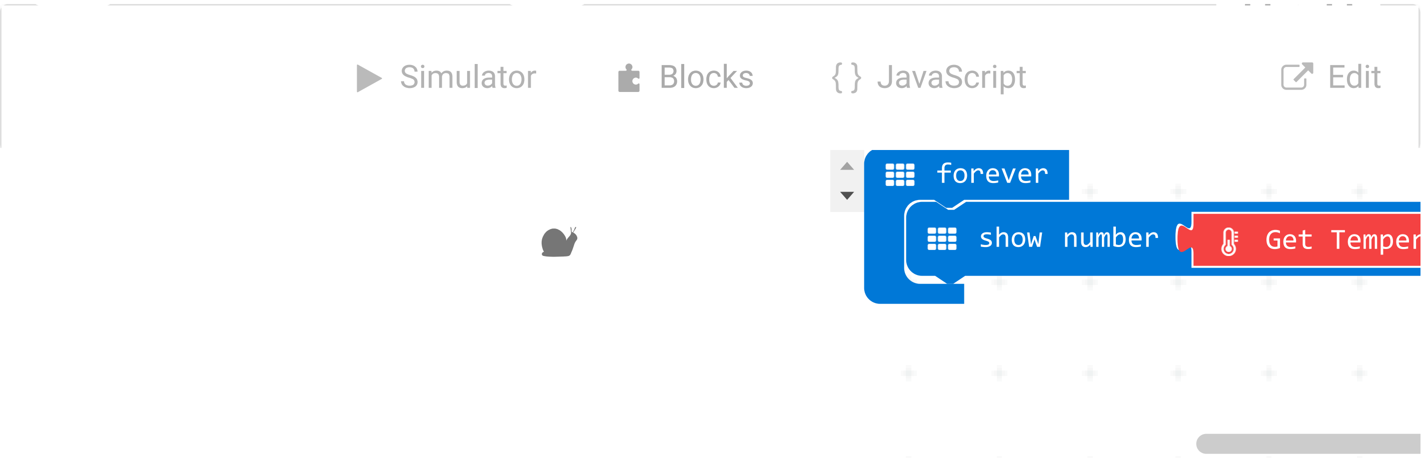
Hardware Hookup

In this example, we'll use Makecode and the gator:temp to create a simple thermometer. The goal is to read the temperature from the sensor, then have the micro:bit display the number as scrolling text. To start, go ahead and power the gator:temp by clipping to the micro:bit, then connect the `temp` pin to pin `0` on the micro:bit (It's also possible to connect to pins `1` or `2`) This will allow the micro:bit to read the analog value on pin `0` .

MakeCode

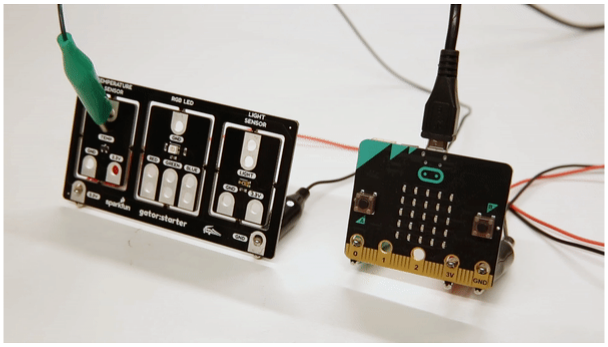
Now we need to program our micro:bit using Makecode to do what we want. In this example, we only need to display the value of the temperature once, so we won't need to create a variable to store the value. In fact, all we'll need to do is use the `Show Number` block under `Basic` . What we'll need to do is replace the `0` in the

`Show Number` block with the `Get Temperature` block in our gator:temp MakeCode package. This will make the micro:bit read the temperature from pin `0` and show the temperature on the LED screen. Re-create the following code into your MakeCode editor or download the example by clicking the download button to test it out!



```
forever
    show number ( Get Temper
```

Uploading the above code to your micro:bit will result in something like the following gif.



Temperature Display

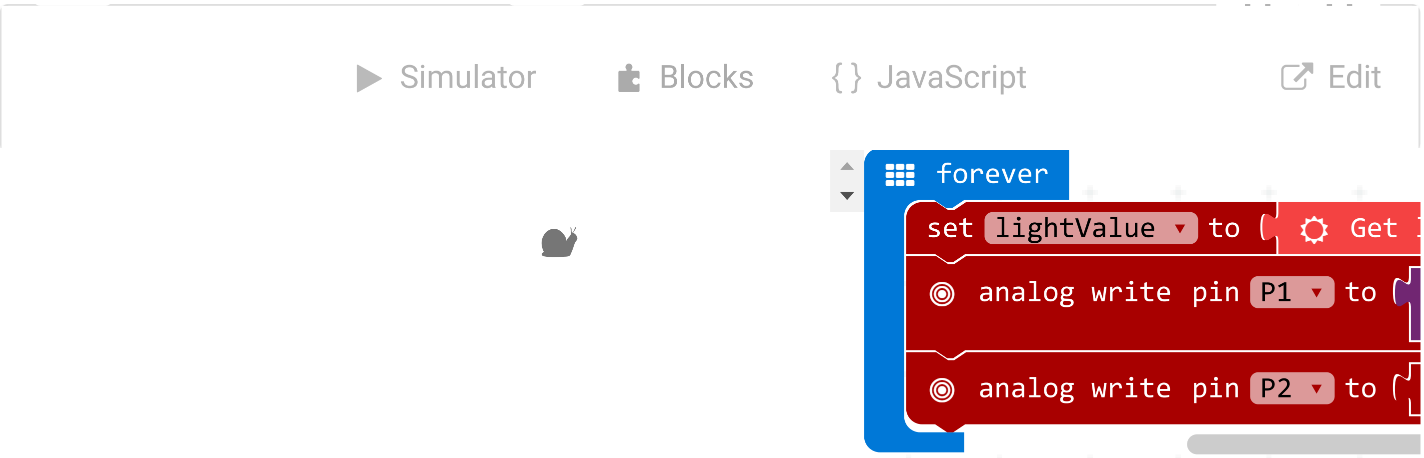
Light Indicator

Hardware Hookup

In this example, we'll go through how to use the gator:light in conjunction with the gator:RGB to create a simple light indicator. Our goal for this example is to create a light indicator to show blue when it's dark, red when it's bright, and shades of purple for everything in between. To begin let's connect all of our components up to one another before we upload code. Begin by powering the gator:temp and connecting the `light` pin to pin `0`. We will then ground the gator:RGB, and connect red and blue to pins `1` and `2`, respectively. This will allow the gator:bit to control red and blue on the gator:RGB.

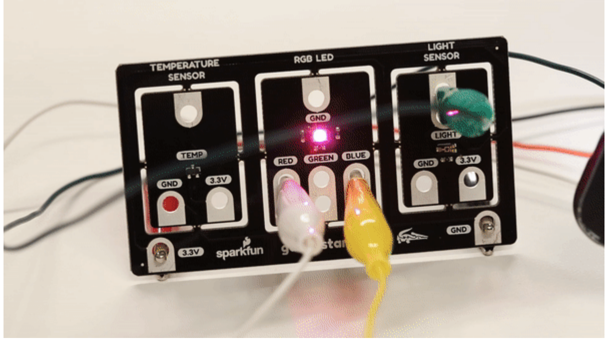
MakeCode

Now let's move on to the Makecode portion, which will tell our micro:bit how to behave. First we need to create a variable to store the value of our light in from the **Variables** category. Let's call it `lightValue` for sake of simplicity. Then let's set the `lightValue` variable to the value that we are reading from our light sensor on pin `0`. Use the block in `Gatorlight` to read the pin `P0`. Since we're going to be writing this value directly to our LED, we can select units of `adcVal` from the second dropdown menu. As an ADC (Analog to digital conversion) value has the same range of possible values as the values that we can write to our LED (both range from 0 to 1023). Re-create the following code into your MakeCode editor or download the example by clicking the download button to test it out!



```
forever
    set lightValue ▾ to ( Get
    analog write pin P1 ▾ to (
    analog write pin P2 ▾ to (
```

If things are pretty bright outside, the value will be closer to `1023`. So you'll end up writing a value that's close to 100% percent brightness to the LED on pin 2 (Blue, if you hooked things up correctly) and a value that's closer to `0` on pin 1 (Red). Moving the light sensor into darkness should give you more red, while bright light should yield blue. Experiment and see how something like a flashlight affects the sensor.



Light Indicator

📌 **Tip:** Looking for more examples using the common cathode RGB LED? Try editing the example used in the micro:bit breakout board to cyle through the colors with the common cathode RGB LED. In this case, you'll be connecting to pin 0, 1, and 2 to control the colors. Or try creating tertiary colors by combining one color at 100% with another at 50%. The LED will be very bright so try covering it with a translucent material to diffuse the light or turning the intensity down.

Resources & Going Further

Now that you've successfully got your gator:starter up and running, it's time to incorporate it into your own project! For more information about the board, check out the resources below:

- Schematic (PDF)
- Eagle Files (ZIP)
- About micro:bit Page
- micro:bit Programmer
- GitHub Product Repo
- SFE Product Showcase

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

micro:bit Educator Lab Pack Experiment Guide

A quickstart guide for the micro:bit educator lab pack.

micro:bot Kit Experiment Guide

Get started with the moto:bit, a carrier board for the micro:bit that allows you to control motors, and create your own robot using this experiment guide for the micro:bot kit.

New!

Getting Started with MicroPython and the SparkFun Inventor's Kit for micro:bit

Learn MicroPython with the micro:bit.

Gator:control ProtoSnap Hookup Guide

Buttons and switches and inputs oh my! Start adding more control to your gator:bit with the SparkFun gator:control.