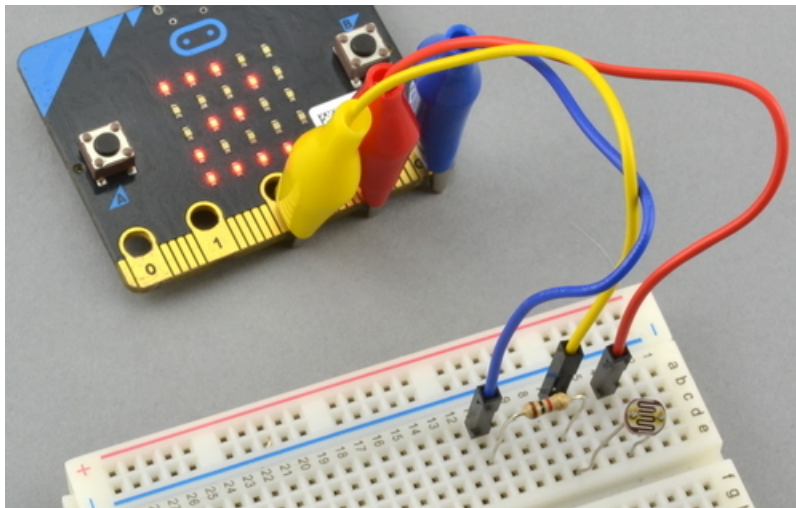


## micro:bit Lesson 4. Sensing Light

Created by Simon Monk



Last updated on 2018-08-22 04:05:53 PM UTC

## Guide Contents

Guide Contents	2
Overview	3
Parts	4
BBC micro:bit	4
Small Alligator Clip to Male Jumper Wire Bundle - 12 Pieces	4
Half-size breadboard	4
Photo cell (CdS photoresistor)	5
Through-Hole Resistors - 2.2K ohm 5% 1/4W - Pack of 25	5
Built-in Light Sensor	6
Breadboard Layout	7
Software	9
JavaScript Block Code	9
MicroPython	9
Arduino	9
Other Things to Do	11
About the Author	11

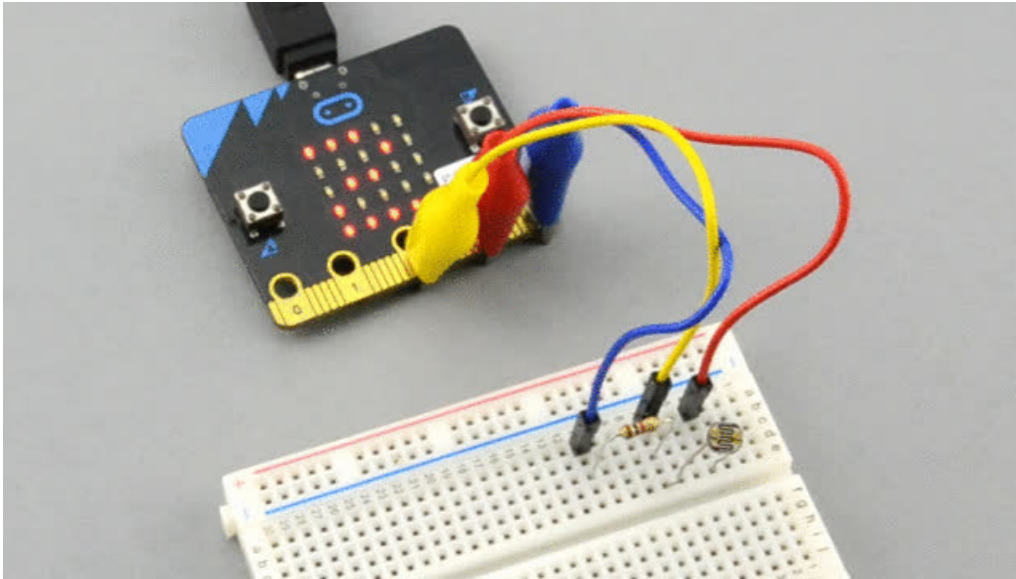
## Overview

---

In this guide, you will learn how to sense the light level using a micro:bit.

By use of a clever bit of coding the micro:bit's display LEDs can be used to measure the light level, without the need for any extra hardware. However, this feature is only available when using the JavaScript blocks editor.

You can however attach a resistor and photoresistor to the micro:bit and use an analog input of the micro:bit to measure the light level. This has the advantage that once you know how to do this, you can attach other analog sensors to your micro:bit and make measurements with them.

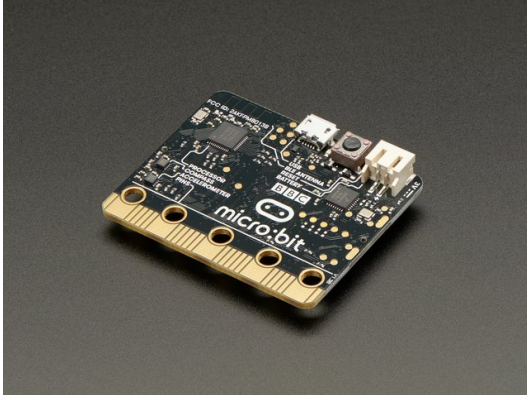


## Parts

---

If you just want to measure light using the built-in facility then you just need a micro:bit.

When it comes to the choice of resistor to use, I have used a 2.2k $\Omega$  resistor here, but a 1k $\Omega$  or 10k $\Omega$  resistor will also work, but will change the range of readings.



BBC micro:bit

\$14.95  
OUT OF STOCK

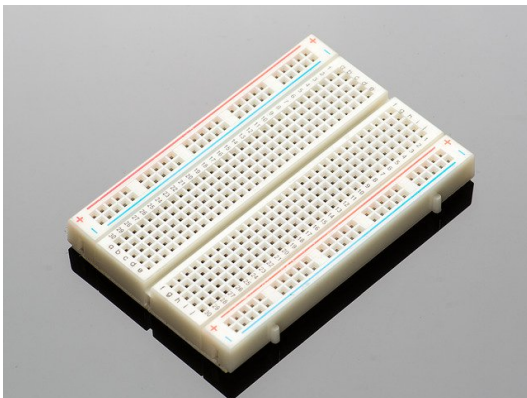
OUT OF STOCK



Small Alligator Clip to Male Jumper Wire Bundle - 12 Pieces

\$7.95  
IN STOCK

ADD TO CART



Half-size breadboard

\$5.00  
OUT OF STOCK

OUT OF STOCK

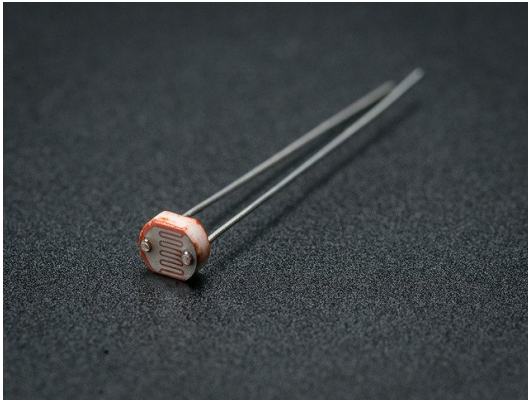
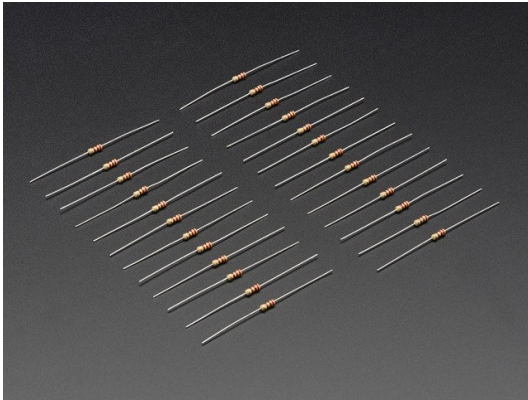


Photo cell (CdS photoresistor)

\$0.95  
IN STOCK

ADD TO CART



Through-Hole Resistors - 2.2K ohm 5% 1/4W - Pack of 25

\$0.75  
OUT OF STOCK

OUT OF STOCK

## Built-in Light Sensor

---

To try out the built-in light sensing capability of the micro:bit, run the JavaScript Blocks code below.

From the editor, you can click on Download button (bottom right) and then copy the downloaded file onto your micro:bit. Alternatively, you can [Click here \(https://adafru.it/CjM\)](https://adafru.it/CjM) to open the editor in a separate browser tab.

The *light level* block returns a value between 0 and 255 where 0 is dark and 255 is very bright. To fit this into a single digit range between 0 and 9 this number needs to be divided by 28.

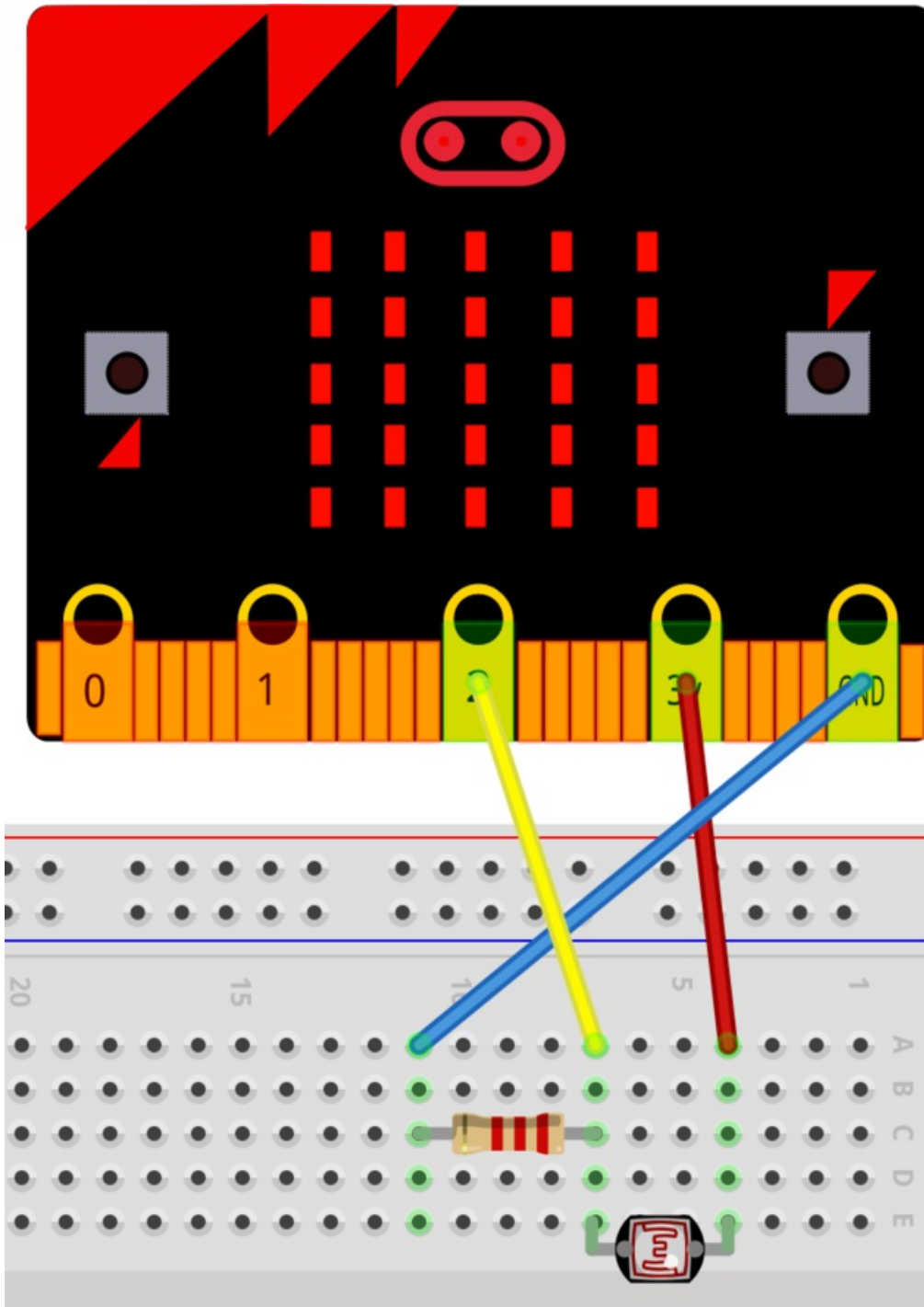
You may be looking on the front of the micro:bit for a light sensor, but you won't find one. The micro:bit actually uses the LEDs of the display as a light sensor.

The technique for this relies on the fact that when an LED is reverse-biased (the wrong polarity for producing light) it can be filled with a small amount of charge. If you then let one of the pins for the LED act as an input and the other as a low output, the charge on the LED will leak away until the digital input stops being HIGH and becomes LOW. The time taken for this to happen depends on the amount of light falling on the LED. You can read more about [this technique here \(https://adafru.it/CjQ\)](https://adafru.it/CjQ)

## Breadboard Layout

Push the wires of the photoresistor and resistor into the breadboard in the positions shown below. Note, it doesn't matter which way around the leads go for either the resistor or photoresistor.

Then connect the breadboard to your micro:bit using the alligator clip to male header pin leads.



The photoresistor and fixed value resistor ( $2.2\text{k}\Omega$ ) are in an arrangement called a voltage divider. The point where they are connected (called the voltage divider's output) together will have a voltage that depends on the ratio of the two resistors. Since the fixed resistor never changes value, then the photoresistor (whose resistance varies with the

amount of light falling on it) will control the voltage at this point. This voltage is connected by the yellow lead to pin 2 of the micro:bit which is used as an analog input to measure this voltage.

If the photoresistor is very brightly lit then it will have a low resistance and the output voltage will be 'pulled up' towards 3V. If on the other hand the photoresistor is in the dark, it will have a very high resistance and the fixed value resistor will dominate, pulling the voltage down towards 0V.



## Software

---

Whatever language you use to measure the light level, the program will follow the same basic pattern. The voltage will be read at pin 2 and then scaled to produce a number between 0 and 9 to be displayed on the micro:bit's LED display.

## JavaScript Block Code

---

The JavaScript Blocks Code editor is embedded directly on this page below. From the editor, you can click on Download button (bottom right) and then copy the downloaded file onto your micro:bit. Alternatively, you can [Click here \(https://adafru.it/CjR\)](https://adafru.it/CjR) to open the editor in a separate browser tab.

The result of the *analog read pin* block will be a number between 0 and 1023 where 0 represents 0V and 1023 represents 3V. This value is stored in the variable *reading*. This is then scaled to be a single digit number by dividing by 50. You can change the number 50 to make the light readings more or less sensitive depending on the range of light you are interested in. For instance the light level indoors is considerably less then outdoors on a sunny day.

## MicroPython

---

To run the MicroPython version of the code, open up the [online Python editor here \(https://adafru.it/CiK\)](https://adafru.it/CiK) and paste the following code into the editor window.

```
from microbit import *

while True:
    reading = pin2.read_analog()
    number = int(reading / 50)
    display.show(str(number))
```

So that the display only shows a single digit, the number is converted to the integer part of the number (no decimal point) using the *int* function.

The *str* function must then be used inside the *show* method, to convert the integer value into the *string* that *show* expects.

## Arduino

---

Make sure that you have your Arduino environment set up for micro:bit by following [this guide \(https://adafru.it/CiL\)](https://adafru.it/CiL).

Now start a new Sketch by clicking on the *File* menu and *New*. Then paste the following code into the editor window.

```
#include <Adafruit_Microbit.h>

const int readingPin = 2;

Adafruit_Microbit_Matrix microbit;

void setup() {
  microbit.begin();
}

void loop() {
  int reading = analogRead(readingPin);
  int number = reading / 50;
  microbit.print(number);
}
```

## Other Things to Do

You might like to try and turn your micro:bit into an automatic night-light that sets the screen to all LEDs on when the light level falls below a certain level.

## About the Author

If you have found this lesson useful, and want to learn Python, then you might like my book Programming the BBC micro:bit: Getting Started with MicroPython.

