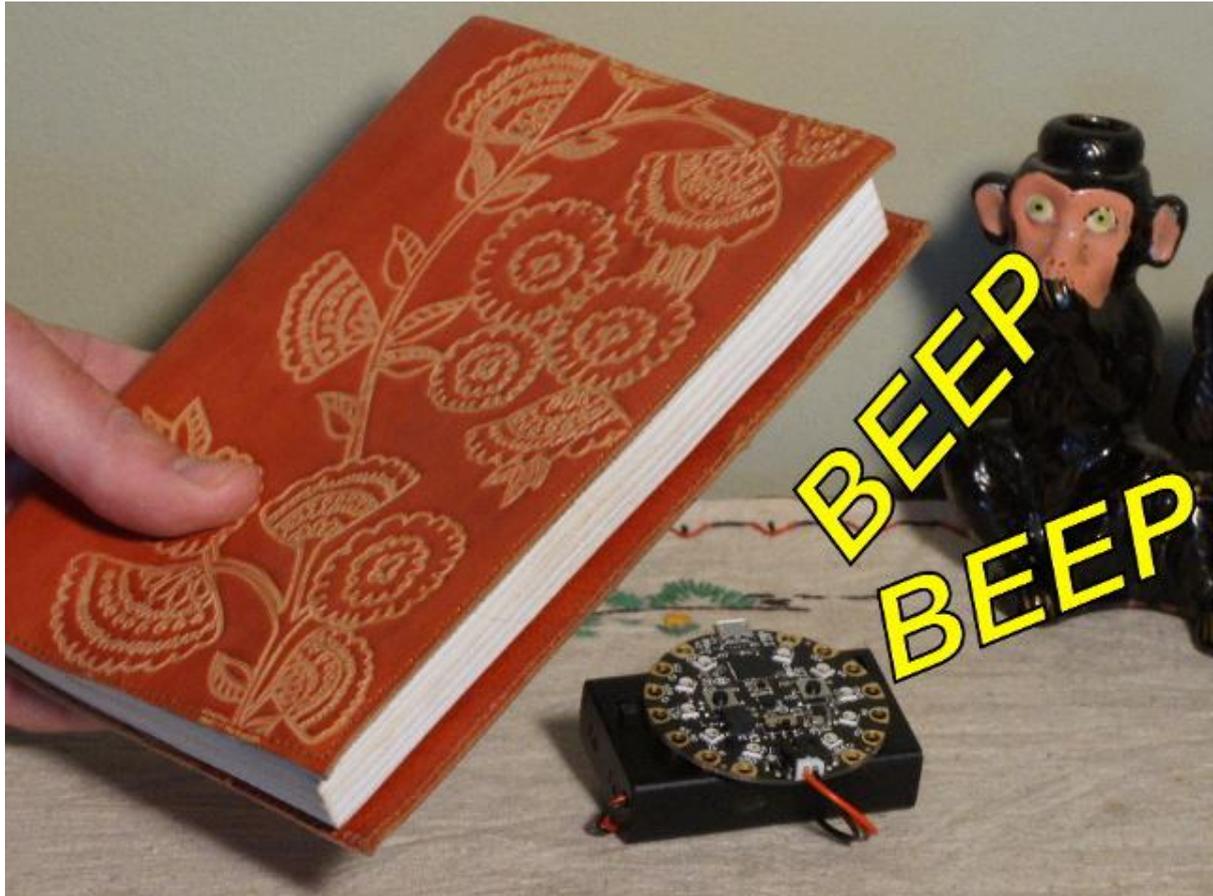




Dear Diary Alarm

Created by Carter Nelson



<https://learn.adafruit.com/dear-diary-alarm>

Last updated on 2022-12-01 02:49:13 PM EST

Table of Contents

Overview	3
<ul style="list-style-type: none">• Required Parts• Before Starting• Circuit Playground Classic• Circuit Playground Express	
Hello Light Sensor	4
<ul style="list-style-type: none">• Using Arduino Serial Monitor• Testing the Light Sensor in Arduino• Using CircuitPython REPL	
Alarm Code	8
Arduino	8
CircuitPython	9
Using the Alarm	10
Questions and Code Challenges	12
<ul style="list-style-type: none">• Questions• Code Challenges	

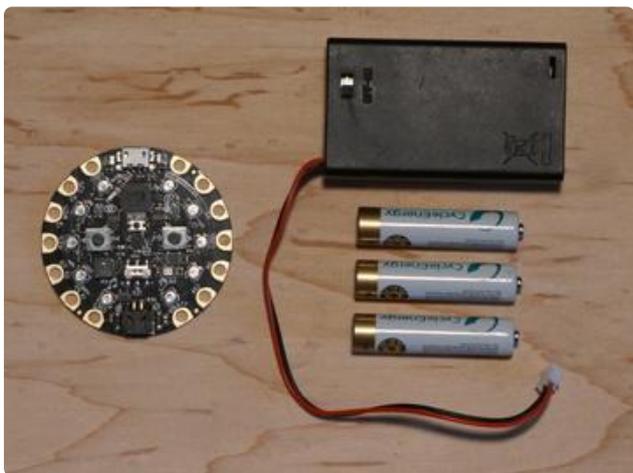
Overview



This is a fun and simple beginner project that uses the Circuit Playground to create a light activated alarm. A diary, or other object, can be placed on top of the Circuit Playground and an alarm will sound when the object is removed.

Required Parts

This project uses the sensors already included on the Circuit Playground, either a [Classic](http://adafru.it/3000) or an [Express](http://adafru.it/3333). The only additional items needed are batteries for power and a holder for the batteries.



Circuit Playground
[Classic](http://adafru.it/3000)
[Express](http://adafru.it/3333)
[3 x AAA Battery Holder](http://adafru.it/727)
3 x AAA Batteries (NiMH work great!)

Before Starting

If you are new to the Circuit Playground, you may want to first read these overview guides.

Circuit Playground Classic

- [Overview \(\)](#)
- [Lesson #0 \(\)](#)

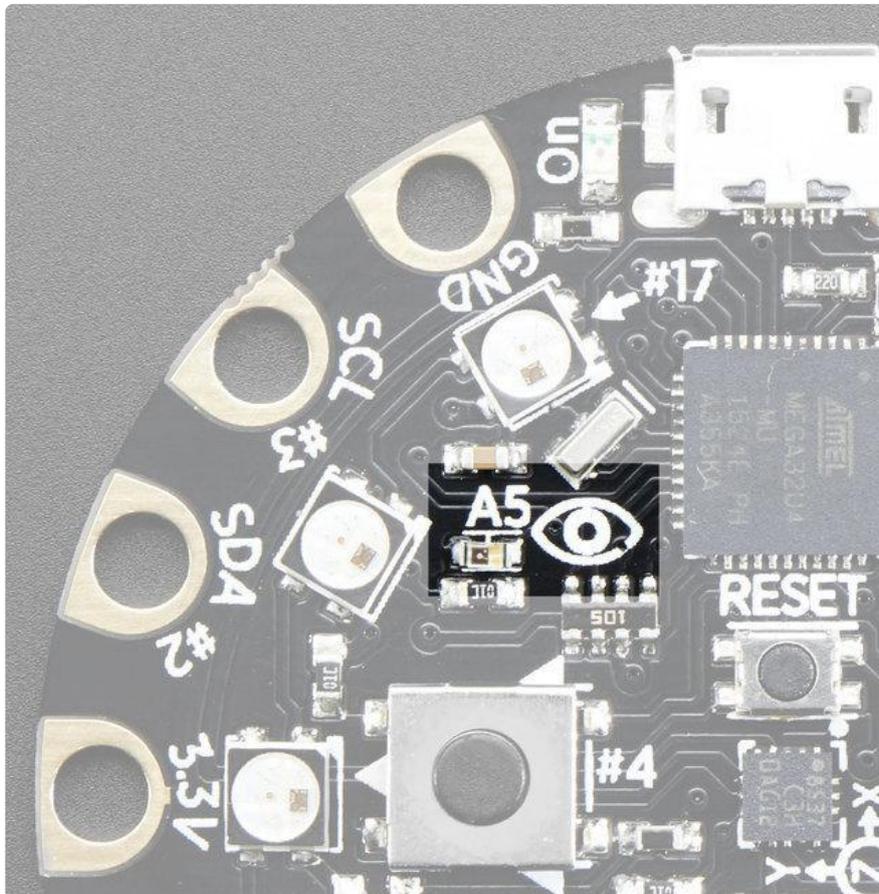
Circuit Playground Express

- [Overview \(\)](#)

Hello Light Sensor

Using Arduino Serial Monitor

To get an idea of how the alarm works, let's start by exploring the light sensor itself. You can read some technical details in the [Lesson #0 Guide \(\)](#). And for reference, here is where the light sensor is located on the Circuit Playground.



Testing the Light Sensor in Arduino

There is an example sketch included with the Circuit Playground library that we can use to play around with the light sensor. It can be found in the following location:

File -> Examples -> Adafruit Circuit Playground -> Hello_CircuitPlayground -> Hello_LightSensor

With this sketch loaded and running on the Circuit Playground, open the Serial Monitor.

Tools -> Serial Monitor

The current value from the light sensor will be printed once a second.



Play around with shining different lights on the light sensor and changing the distance. Also try covering the sensor with your hand, a sheet of paper, or anything else lying around (like a diary). The value can range from 0 to 1023, with higher numbers meaning more light is getting to the sensor. When you cover the light sensor, you should see the value go down.

Another way to watch the value of the light sensor is to use the Serial Plotter. To do this, let's first modify the code slightly. The code below removes the text from the output and increases the rate at which the value is displayed.

```
#include <Adafruit_CircuitPlayground.h>;
#include <Wire.h>;
#include <SPI.h>;

int value;

void setup() {
  Serial.begin(9600);
```

```

    CircuitPlayground.begin();
  }

  void loop() {
    value = CircuitPlayground.lightSensor();

    Serial.println(value);

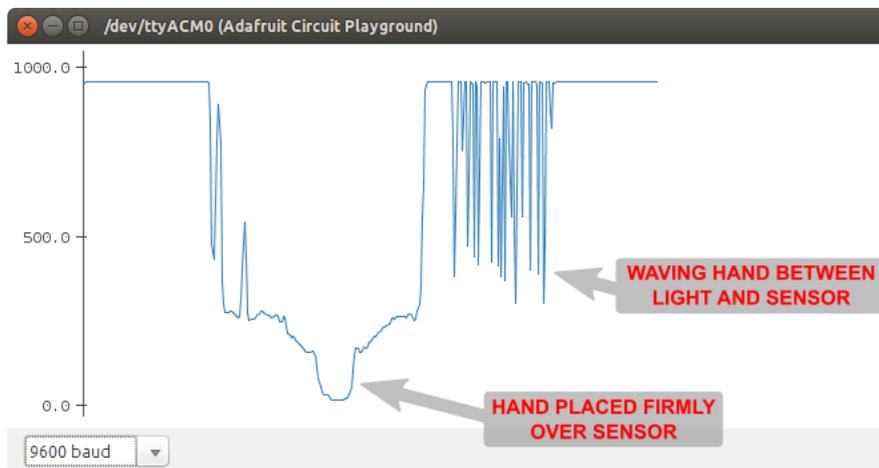
    delay(100);
  }

```

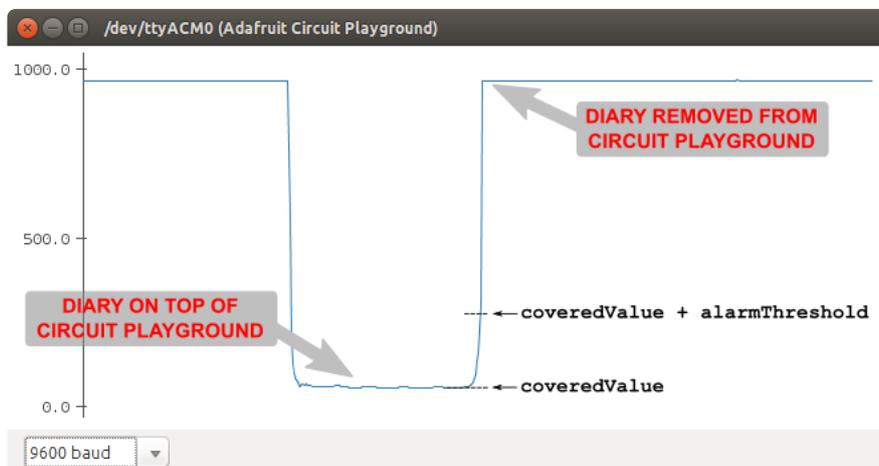
With this sketch loaded and running on the Circuit Playground, open the Serial Plotter.

Tools -> Serial Plotter

The light sensor value will now be plotted like a strip chart as shown below. Again, play around with different lighting conditions and watch the value change.



Now we can explain how the alarm is going to work. With a diary placed on top of the Circuit Playground, the light sensor will have a low reading. If the diary is removed, more light will get to the sensor and the reading will increase. An example of what this would look like is shown below.



So if we simply monitor the value of the light sensor and watch for this increase, we can detect when the diary has been removed and sound an alarm.

Using CircuitPython REPL

If you want to do the same thing using CircuitPython, here is a simple program to do that:

CircuitPython only works on the Circuit Playground Express.

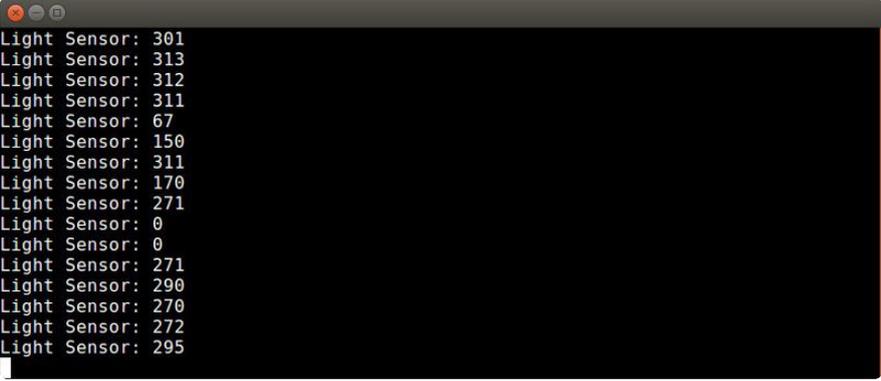
```
# Circuit Playground Express Hello Light Sensor
import time
from adafruit_circuitplayground.express import cpx

# Loop forever
while True:
    value = cpx.light
    print("Light Sensor: {}".format(value))
    time.sleep(1)
```

Save the above to a file named main.py on your Circuit Playground Express. Then read the following guide section for how to setup and access the REPL:

Serial Console (REPL)

With the above program running on the Circuit Playground Express, you should see output like this in your REPL:



```
Light Sensor: 301
Light Sensor: 313
Light Sensor: 312
Light Sensor: 311
Light Sensor: 67
Light Sensor: 150
Light Sensor: 311
Light Sensor: 170
Light Sensor: 271
Light Sensor: 0
Light Sensor: 0
Light Sensor: 271
Light Sensor: 290
Light Sensor: 270
Light Sensor: 272
Light Sensor: 295
```

Try playing around with shining lights on the sensor and placing your hand over it to see the value change.

Alarm Code

The alarm code is available in both Arduino and [CircuitPython \(\)](#) version. They both work in the same general way, which is:

1. Wait for a button to be pressed.
2. Provide a countdown timer with the NeoPixels. (the diary should be placed on Circuit Playground during countdown)
3. Read and store the value of the light sensor.
4. Loop forever reading the light sensor value.
5. If value exceeds the stored value from step 3, sound the alarm.

Arduino

Here is the full code listing for the Arduino version. You can copy and paste it into the Arduino IDE.

This code has been tested and verified to work on either a Circuit Playground Classic or Express.

```
// Circuit Playground Dear Diary Alarm
//
// Don't let sister/brother read your secrets! Place the "armed" Circuit
// Playground under your diary. If someone tries to remove the diary, an alarm
// will sound.
//
// Author: Carter Nelson
// MIT License (https://opensource.org/licenses/MIT)

#include <Adafruit_CircuitPlayground.h>;
#include <Wire.h>;
#include <SPI.h>;

uint16_t lightValue;           // current light sensor reading
uint16_t coveredValue;        // the light sensor value with diary on top
uint16_t alarmThreshold = 20; // adjust to change alarm sensitivity
bool triggered = false;       // becomes true when diary removed

////////////////////////////////////
void setup()
{
  // Initialize Circuit Playground library.
  CircuitPlayground.begin();

  // Turn on all the NeoPixels.
  for (int p=0; p<10; p=p+1) {
    CircuitPlayground.setPixelColor(p, 255, 0, 0);
  }

  // Wait for button press.
  while ( (CircuitPlayground.leftButton() == false) &&
          (CircuitPlayground.rightButton() == false) ) {
    // Do nothing.
  }
}
```

```

}

// Countdown timer.
for (int p=0; p<10; p=p+1) {
  CircuitPlayground.setPixelColor(p, 0, 0, 0);
  delay(500);
}

// Compute covered light sensor value as average of 5 readings.
coveredValue = 0;
for (int i=0; i<5; i=i+1) {
  lightValue = CircuitPlayground.lightSensor();
  coveredValue = coveredValue + lightValue;
}
coveredValue = coveredValue / 5;

// Beep to indicate ARMED.
CircuitPlayground.playTone(1000, 1000);
delay(1000);
}

////////////////////////////////////
void loop()
{
  // Check to see if alarm has been triggered.
  if (triggered) {
    // Sound the alarm.
    CircuitPlayground.playTone(2000, 500);
    delay(500);
    CircuitPlayground.playTone(3000, 500);
    delay(500);
  } else {
    // Get current light sensor value.
    lightValue = CircuitPlayground.lightSensor();

    // Check light sensor value to determine if alarm should be triggered.
    if (lightValue > coveredValue + alarmThreshold) {
      triggered = true; // Trigger the alarm.
    }
  }
}
}
}

```

CircuitPython

Here is the full code listing for the CircuitPython version. Name the file main.py and save it to your Circuit Playground Express.

CircuitPython only works on the Circuit Playground Express.

```

# Circuit Playground Express Dear Diary Alarm
#
# Don't let sister/brother read your secrets! Place the "armed" Circuit
# Playground under your diary. If someone tries to remove the diary, an alarm
# will sound.
#
# Author: Carter Nelson
# MIT License (https://opensource.org/licenses/MIT)
import time
from adafruit_circuitplayground.express import cpx

# Set alarm threshold

```

```
ALARM_THRESHOLD = 20

# Turn on all the NeoPixels
cpx.pixels.fill((255,0,0))

# Wait for button press
while (cpx.button_a == False) and (cpx.button_b == False):
    # Do nothing
    pass

# Countdown timer
for p in range(10):
    cpx.pixels[p] = 0
    time.sleep(0.5)

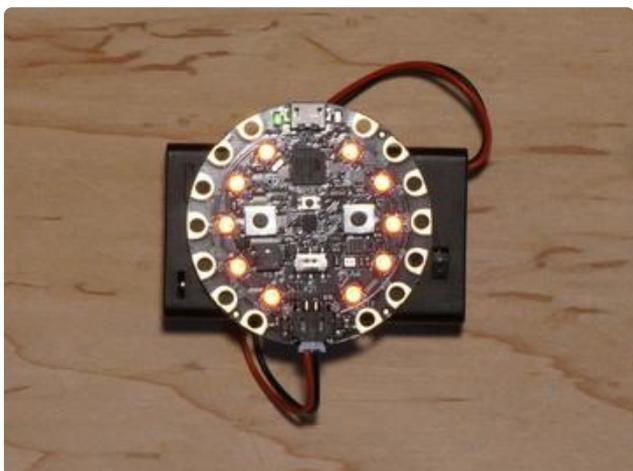
# Compute covered light sensor value
covered_value = 0
for count in range(5):
    covered_value = covered_value + cpx.light
covered_value = covered_value / 5

# Beep to indicate armed
cpx.play_tone(1000, 1)

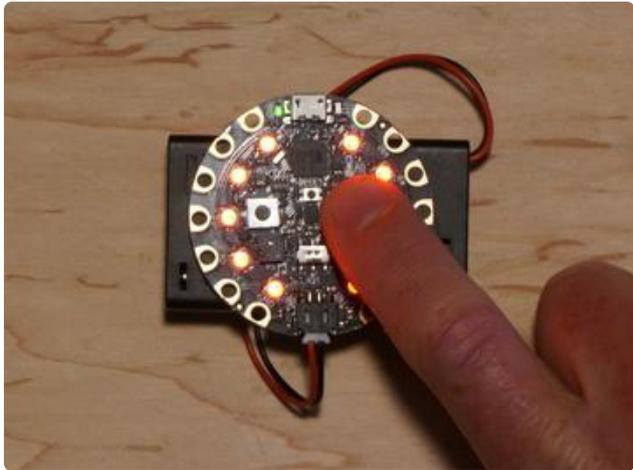
# Begin alarm
triggered = False
while True:
    if triggered:
        cpx.play_tone(2000, 0.5)
        time.sleep(0.5)
        cpx.play_tone(3000, 0.5)
        time.sleep(0.5)
    if cpx.light > covered_value + ALARM_THRESHOLD:
        triggered = True
```

Using the Alarm

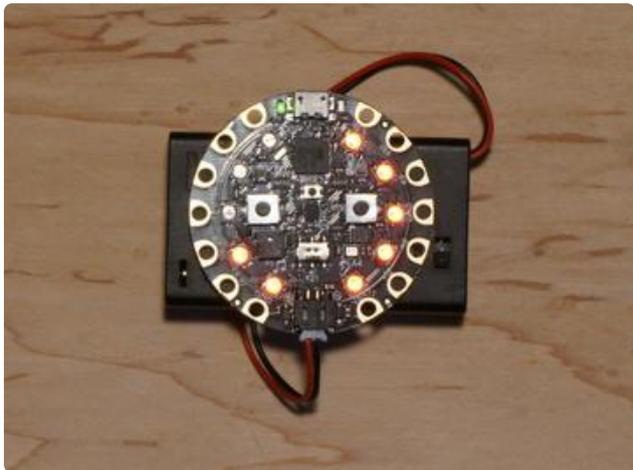
Once the sketch has been loaded on the Circuit Playground, it is ready for use. Install the batteries, plug in the power cable, and turn the switch on. You should see all of the NeoPixels come on and turn red.



All the lights are on. Ready to use.



Press either button to start the countdown timer.



The NeoPixels will turn off one at a time.



BEFORE ALL THE LIGHTS GO OUT, PLACE THE DIARY ON THE CIRCUIT PLAYGROUND.

Make sure the diary covers the light sensor before the countdown ends.

When the countdown ends a short BEEP will sound.

THE ALARM IS NOW ARMED

If you now remove the diary from the Circuit Playground, the alarm should sound. You can press the RESET button to stop the alarm and start again. Yes, this is an easy way

to defeat the alarm, but hopefully you will hear the alarm before someone figures that out.

Questions and Code Challenges

The following are some questions related to this project along with some suggested code challenges. The idea is to provoke thought, test your understanding, and get you coding!

While the alarm sketch provided works, there is room for improvement and additional features. Have fun playing with the provided code to see what you can do with it.

Questions

- Would the alarm work at night or in a dark room?
- Why is computing a covered value for the light sensor necessary?
- What would happen if the alarm threshold were set too low or too high?

Code Challenges

- Increase the time of the countdown timer.
- Change the alarm sound.
- Use the NeoPixels to add some lights to the alarm.