

Understanding Light Sensor using Circuit Playground Express (CPX)

What You're Learning

In this activity, you will:

- Learn what a **light sensor** is and how it works.
- Understand how to use the **CPX built-in light sensor** to measure light intensity.
- Learn to use **if-else conditions** to respond to light level changes.
- Understand how Python reads sensor data in **real time**.

What is a Light Sensor?

A **light sensor** is an **electronic component** that detects and measures the amount of light (brightness) present in the environment.

It converts the **light energy** into an **electrical signal** that can be read by a microcontroller such as the **Circuit Playground Express (CPX)**.

How It Works

- The light sensor on the CPX is a **phototransistor**.
- It behaves like a switch that allows more current to pass when there is **more light** and less current when there is **less light**.
- The CPX measures this current as a **numeric value** (light level).
-

The sensor values generally range from:

- **0 to 50** → Very dark
- **50 to 150** → Dim or indoor light
- **150 to 300+** → Bright sunlight or torch light

The value keeps **changing continuously** as the brightness around the board changes.

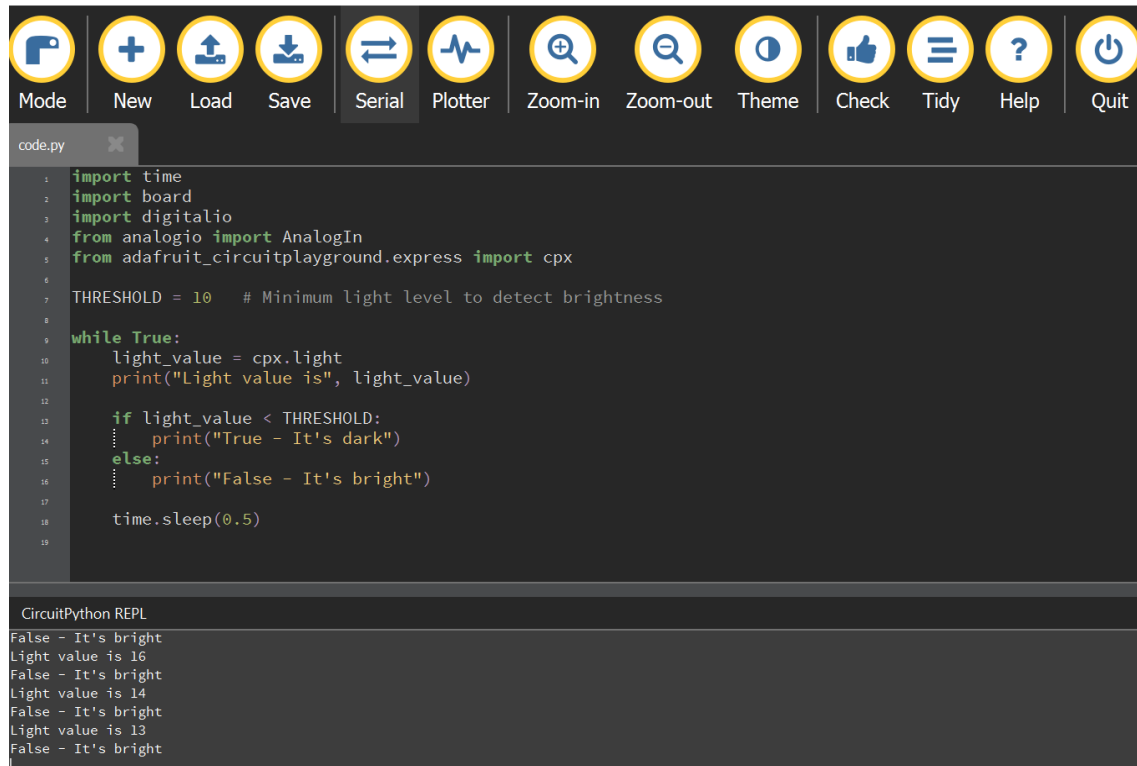
Where It Is Used

Light sensors are used in many **real-world applications**, for example:

- **Automatic street lights** (turn ON in dark, OFF in daylight)
- **Mobile brightness control** (adjusts screen based on surrounding light)

- **Smart homes** (detects when to turn lights on/off)
- **Solar trackers** (track sunlight direction)

Code



The screenshot shows the CircuitPython IDE interface. At the top is a toolbar with icons for Mode, New, Load, Save, Serial, Plotter, Zoom-in, Zoom-out, Theme, Check, Tidy, Help, and Quit. Below the toolbar is a code editor window titled 'code.py' containing the following Python code:

```

1 import time
2 import board
3 import digitalio
4 from analogio import AnalogIn
5 from adafruit_circuitplayground.express import cpx
6
7 THRESHOLD = 10 # Minimum light level to detect brightness
8
9 while True:
10     light_value = cpx.light
11     print("Light value is", light_value)
12
13     if light_value < THRESHOLD:
14         print("True - It's dark")
15     else:
16         print("False - It's bright")
17
18     time.sleep(0.5)
19

```

Below the code editor is a REPL window titled 'CircuitPython REPL' showing the output of the program:

```

False - It's bright
Light value is 16
False - It's bright
Light value is 14
False - It's bright
Light value is 13
False - It's bright

```

Code Explanation (Line-by-Line)

```

import time
import board
import digitalio
from analogio import AnalogIn
from adafruit_circuitplayground.express import cpx

```

- These lines **import libraries** required to control the CPX board.
 - `time` → allows delays.
 - `board`, `digitalio`, `analogio` → used to access hardware pins.
 - `cpx` → helps control CPX features like sensors, LEDs, buttons, etc.

```
THRESHOLD = 10
```

- Defines a **limit value** to compare light readings.
- If the light level is less than 10, the program considers it **dark**.

```
while True:
```

- Creates an **infinite loop** so the program runs continuously.

```
light_value = cpx.light
```

- Reads the **current light sensor value** from the CPX board and stores it in the variable `light_value`.

```
print("Light value is", light_value)
```

- Displays the **measured light value** in the serial monitor, helping you observe the sensor's response to brightness changes.

```
if light_value < THRESHOLD:  
    print("True")  
else:  
    print("False")
```

- Checks if the light value is **below** the threshold (dark).
 - If **dark**, it prints "True".
 - Otherwise (bright), it prints "False".

```
time.sleep(0.5)
```

- Waits for half a second before repeating the loop — this prevents too-fast updates.

Output

When you run this code on your CPX board:

- You will see messages printed in the **serial monitor** like:

```
Light value is 85  
False  
Light value is 5  
True
```

- When you cover the light sensor or move to a dark place → prints **“True”** (light value is low).
- When you shine light or move to bright area → prints **“False”** (light value is high).

Learning Outcome

After completing this, you will:

- Understand how a **light sensor** works.
- Know how to use **threshold-based decisions** in Python.
- Be able to build projects that respond to brightness — e.g., automatic night lights or alarms.