cedargrove_shadowwatcher

ShadowWatcher is a CircuitPython helper class to detect a shadow cast over an analog light sensor such as the *ALS-PT19* phototransistor used in the *Adafruit PyPortal*, *PyGamer*, *PyBadge*, *CircuitPlayground Express*, *CircuitPlayground Bluefruit*, and the *ALS-PT19* breakout board. Incorporates a low-pass filter to reduce sensitivity to flickering light levels which may be caused by power line frequency or light dimmer PWM passthrough. Useful as a simple gesture detector.

- Author(s): JG Cedar Grove Maker Studios

Implementation Notes

The concept of the ShadowWatcher was inspired by Liz Clark's gesture detector code for the Adafruit *Smart Mirror with PyPortal* project: https://learn.adafruit.com/smart-mirror-with-pyportal/coding-the-smart-mirror

Hardware:

Analog light sensor hardware such as the *ALS-PT19* phototransistor with an output value directly in proportion to the light intensity.

The ShadowWatcher was primarily built for and tested on the PyPortal, but should be able to function reliably on other microcontrollers with similar sensors. The automatic samples mode will test the microcontroller's analog acquisition latency and adjust the internal low-pass filter's sample size to maintain the ~25 Hz cutoff frequency.

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: https://circuitpython.org/downloads

class cedargrove_shadowwatcher.ShadowWatcher(_*, pin, threshold=0.9, samples=2000, decay=0.01, auto=False_)

Class representing the CedarGrove ShadowWatcher. Detects a shadow cast over an analog light sensor.

Parameters:

- **pin** – The analog input pin that connects to the light sensor.
- **threshold** — The relative brightness threshold for shadow detection. Defaults to ***0.9***, 90% of the foreground-to-background brightness ratio. Range is floating 0.0 to 1.0.
- **samples** — The number of samples needed for the internal read method's low-pass filter. Default is ***2000*** for a cut-off frequency of approximately 25Hz when using a SAMD-51 (M4) clocked at 120MHz. Range is any positive non-zero integer value.
- **decay** — The magnitude of the foreground-induced decay used to continuously adjust the background value each time the foreground value is read. The decay compensates for slowly changing background light levels. Default is ***0.01***, equivalent to a weight of 1 foreground sample per 99 background samples. Range is floating 0.0 to 1.0.
- **auto** — Enables automatic samples detection when ***True***. If enabled, the samples value is replaced with a calculated value based upon measured acquisition time. This preserves the low-pass filter's cutoff frequency regardless of variations in microcontroller ADC latency. Defaults to boolean ***False***.

background

The most recent background measurement. Range is an integer value, 0 to 65535. A value of 65535 is approximately 1100 Lux.

foreground

The most recent foreground measurement. Range is an integer value, 0 to 65535. A value of 65535 is approximately 1100 Lux.

refresh_background()

Read and update the filtered background sensor level.

detect()

The fundamental ShadowWatcher function. Compares foreground to background light levels to detect a shadow. The function uses two thresholds, a lower one that indicates a shadow and an upper threshold that when exceeded, indicates an increased background light level. Returns ***True*** when the ratio of foreground to background is less than the threshold. A non-blocking method.

Example:

```
import board
import time
from shadowwatcher import ShadowWatcher

# Instantiate detector class and establish background level
gesture = ShadowWatcher(pin=board.LIGHT, auto=True)

while True:
    if gesture.detect():
        print(f"SHADOW DETECTED")
        while gesture.detect():
            # Wait until the shadow is gone
            time.sleep(1)
        # Rebaseline the background level
        gesture.refresh_background()
        print(f"background: {gesture.background:6.0f}")
```