Twin Peaks Light Reactive Picture Frame

Created by Liz Clark

https://learn.adafruit.com/twin-peaks-light-reactive-pyportal-picture-frame

Last updated on 2023-12-05 05:22:45 PM EST

Table of Contents

Overview

As you probably know by now, the PyPortal has quite a few built-in hardware features. In this project, we're going to take advantage of the light sensor, which allows you to have a built-in, light sensitive analog input. We're going to use that light sensor to change between two pictures on the PyPortal. If there isn't enough light hitting the sensor, then the picture will change and play an audio file; otherwise the original picture will remain displayed.

This particular instance of this project concept is based on the cult classic TV series Twin Peaks. Twin Peaks is about the mythical town of Twin Peaks in Washington State and the many mysterious, creepy and bizarre happenings there that all seem to center around Laura Palmer.

One of the classic images from Twin Peaks is in fact a physical photograph: Laura Palmer's homecoming photo, displayed in a silver metal frame. This framed photograph is displayed during the end credits of every episode of Twin Peaks and also has some featured moments in the series. This is why the picture displayed on the PyPortal when enough light is available is the same iconic homecoming photo and the housing for this project is a modified silver picture frame.

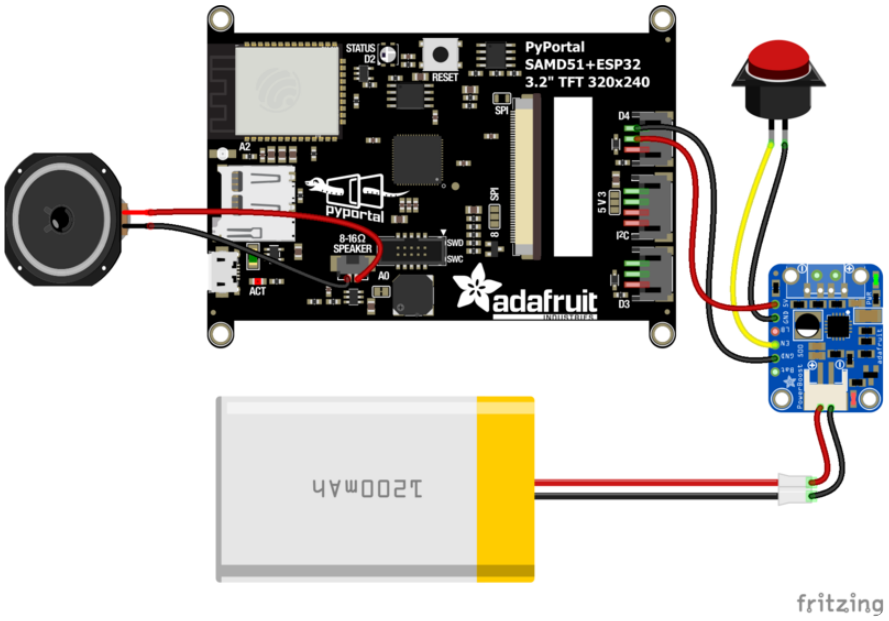
When darkness falls though, a more sinister Twin Peaks reference is revealed: the Woodsmen. The Woodsmen are a mysterious group featured in The Return season of Twin Peaks, which ran 25 years (as promised by Laura Palmer herself) after the original two seasons. The Woodsmen are very cryptic in both their actions and their words. One of their catchphrases, "Gotta light?", is played alongside their photo when darkness falls.

If you aren't into Twin Peaks though, don't worry. You can use any picture or audio files with this project concept. Let's take a look at how to get everything setup with the code and circuit.

1 x Adafruit PyPortal - CircuitPython Powered Internet Display PyPortal	https://www.adafruit.com/product/4116
1 x PowerBoost 500 Basic - 5V USB Boost @ 500mA from 1.8V+ PowerBoost	https://www.adafruit.com/product/1903
1 x Mini On/Off Push-Button Switch On/off button	https://www.adafruit.com/product/3870
1 x Mini Oval Speaker - 8 Ohm 1 Watt Speaker	https://www.adafruit.com/product/3923
1 x JST PH 3-Pin to Male Header Cable - 200mm JST cable for power	https://www.adafruit.com/product/3893
1 x Lithium Ion Polymer Battery - 3.7v 1200mAh Lipo battery	https://www.adafruit.com/product/258
1 x Silver Picture Frame 3.5" x 2.5" picture frame - Amazon	https://www.amazon.com/gp/product/B001LOQPU8/ref=ppx_yo_dt_b_asin_title_o07_s01?ie=UTF8&psc=1

Wiring

The PyPortal has taken care of a lot of the circuitry for this project for us. For audio, we're going to plug-in a speaker to the JST speaker connection on the PyPortal. For power, we can use a PowerBoost board to allow for a rechargeable lipo battery. The PowerBoost board also allows for an on/off switch to be included in the circuit.



Following the wiring diagram, we connect the switch terminals to EN and GND on the PowerBoost. Then take a 3-wire JST cable and connect the red wire to 5V and the bla ck wire to GND. You can snip off the third wire on the JST cable since it won't need to connect to anything.

This particular JST connector type can plug right into the D4 plug on the PyPortal allowing for simple and removable wiring.

Of course if you don't want this project to utilize a lipo battery, you can also use the USB port for power with a power supply or USB battery bank.

The PyPortal Case tutorial has a more detailed walkthrough of how to build this circuit

Picture and Audio Files

Now we can move on to the software side of things. First, let's prepare our pictures. The two photos needed for this project are available for download from GitHub at the link below along with the audio file. For future projects though, you'll want to keep in mind that for photos to be properly displayed on the PyPortal, they need to be converted to 320 x 240 pixel RGB 16-bit bitmap files. You can do this with a few different free pieces of software, including MS Paint on Windows.

To have the pictures be in the correct orientation, you also need to rotate them 90 degrees, as shown in the screen grab below. This way they'll be in a vertical orientation for the PyPortal and you won't need to rotate the images in the code.

For audio, you'll need the file to be in a .wav format. See this guide () on how to prepare audio files into the 16-bit, 22,050 Hz, mono wav format easily used on microcontrollers.

Image and audio files on GitHub

CircuitPython Code

The code is written in CircuitPython and is fairly brief. If you're just getting started with CircuitPython and/or the PyPortal it's recommended to check out this Learn guide to help get you started:

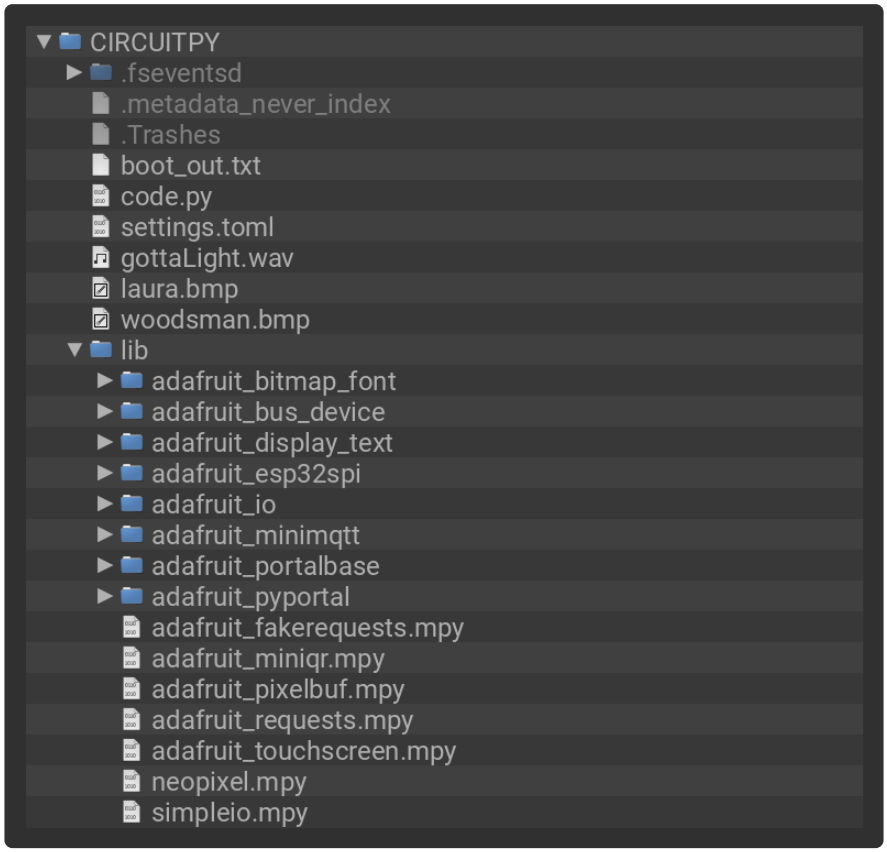
Adafruit PyPortal Intro Learn Guide

Click the Download Project Bundle button and save the zip file on your computer. Plug your PyPortal into your computer via a known good USB data+power cable. In your operating system file explorer app, there should be a new drive named CIRCUIT PY. Copy the contents of the zip file to the CIRCUITPY drive.

```
# SPDX-FileCopyrightText: 2019 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# Adafruit PyPortal display of Twin Peaks
# Liz (BlitzCityDIY) for Adafruit Industries  MIT License
# Tutorial: https://learn.adafruit.com/twin-peaks-light-reactive-pyportal-picture-
frame
import time
import board
from analogio import AnalogIn
from adafruit_pyportal import PyPortal

analogin = AnalogIn(board.LIGHT)

cwd = ("/"+__file__).rsplit('/', 1)[0]

laura = (cwd+"/laura.bmp")

woodsman = (cwd+"/woodsman.bmp")

gottaLight = (cwd+"/gottaLight.wav")

pyportal = PyPortal(default_bg=laura)

def getVoltage(pin):  # helper
    return (pin.value * 3.3) / 65536

while True:

    if getVoltage(analogin) > 0.175:
        pyportal.set_background(laura)
        time.sleep(1)
    else:
        pyportal.set_background(woodsman)
```

```
        pyportal.play_file(gottaLight)
        time.sleep(1)
```

At this point, everything should be copied over to the CIRCUITPY drive. It should look something like this:



First, we're importing the necessary libraries: `time`, `board`, `displayio`, `analogio` and `adafruit_pyportal`. We begin with declaring the built-in light sensor as an analog input with `analogin = AnalogIn(board.LIGHT)`. Utilizing `board.LIGHT` for the light sensor is possible thanks to the `adafruit_pyportal` library.

Next, we're going to take care of our audio and picture files. We declare `cwd` as the current working directory, which is where our files will be saved on the PyPortal. We're then going to declare our files to be variables so that we can use them later in the code. The PyPortal is then setup to have the PyPortal settings from the library along with the "laura" image as the default background so that on first boot, the laura.bmp file will be displayed.

The final step before the loop is the function that allows us to read the analog input data from the light sensor.

In the loop, there is an if/else statement. It begins with if the analog input from the light sensor is greater than 0.175, then the laura.bmp file will be displayed. Else, the w

oodsman.bmp file will be displayed and the gottaLight.wav file will also be played. Each portion also has a one second delay.

Assembly

Technically you could be done with the project now, however sometimes coming up with a housing for your project can be half the fun (or half the work - Ed.). As mentioned in the Overview portion of this guide, the inspiration for this project comes from the picture of Laura Palmer that is framed in a silver metal picture frame in the series. As a result, we're going to go thru how to modify a silver metal picture frame that fits the PyPortal perfectly.

If you don't want to modify a picture frame though, there are a few other options. The only requirement is that the light sensor has to be visible for it to work. There are a few 3D printed case options that have this or you could even utilize the acrylic case specially made for the PyPortal.

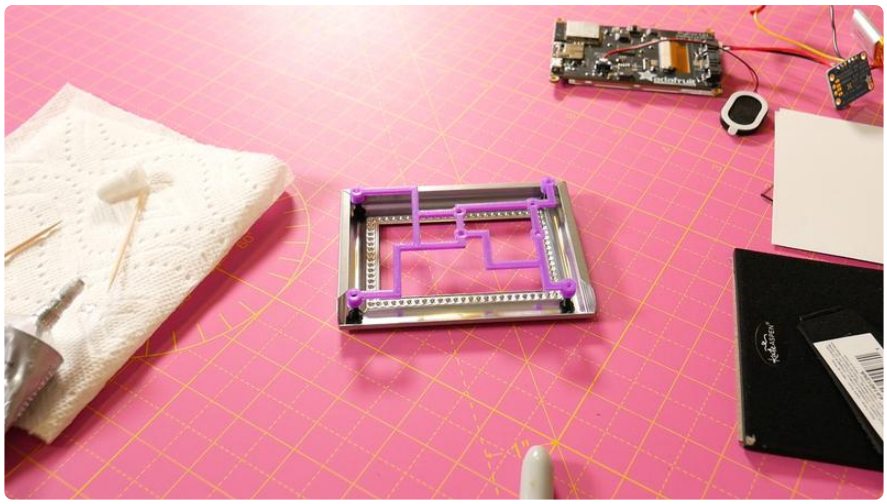
1 x Adafruit PyPortal Desktop Stand Enclosure Kit https://www.adafruit.com/product/4146
Acrylic case for the PyPortal

PyPortal YouTube Bezel on Thingiverse

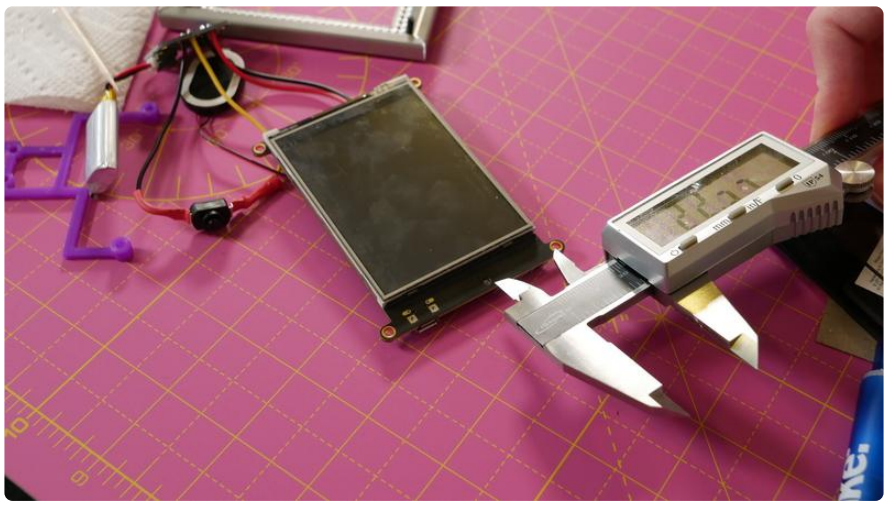
For those daring to modify a store frame, you'll need one that is 3.5" x 2.5". The one used in the project is from Amazon. It is very affordable and is often used (apparently) for weddings. The frame housing also utilizes the 3D printed bracket portion from the PyPortal case modeled by the Ruiz brothers and available on Thingiverse:

PyPortal Case on Thingiverse

Once you have your frame, remove the backing and any inner packaging. Carefully remove and discard the glass as well; it won't be necessary going forward. Take the 3D printed bracket and use it to mark where the mounting holes for the PyPortal are located, making sure to center it in the frame. Then, glue some 2.5mm screws in those spots. E6000 glue works well for this.

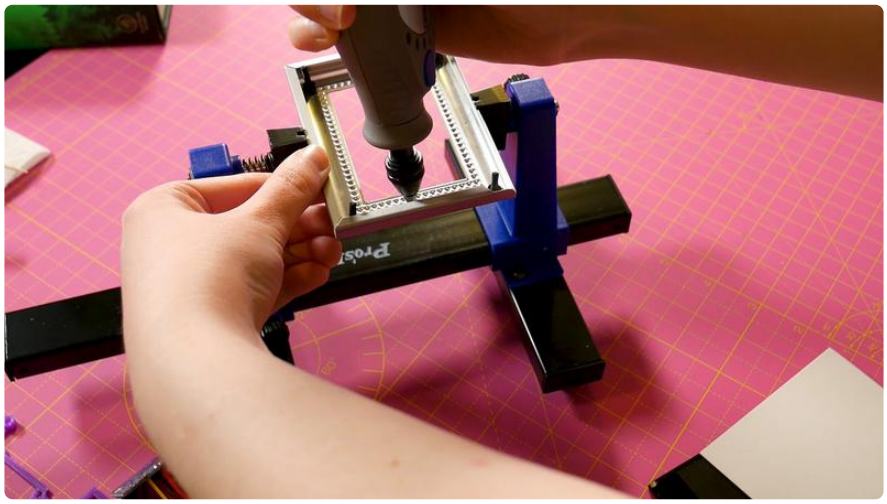


After the screws have setup in the glue, measure on the PyPortal the distance that the light sensor is from each mounting hole. Then mark on the frame approximately where the light sensor will be located once the PyPortal is inserted in the frame.

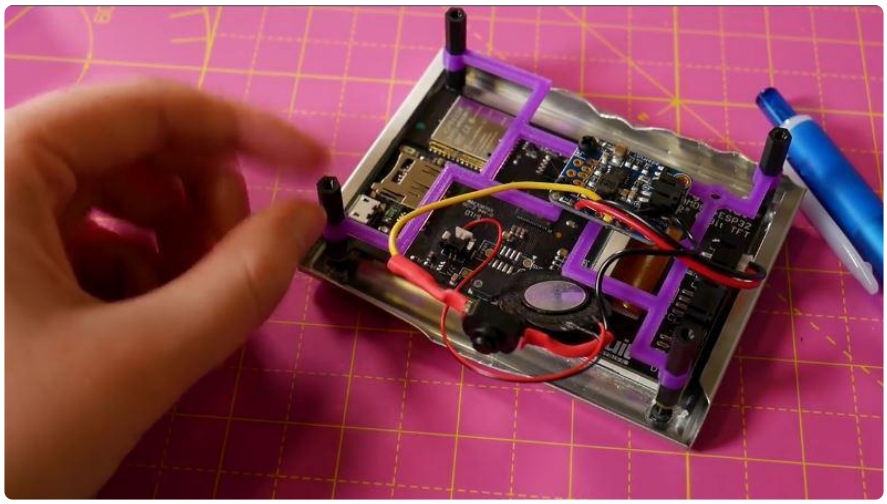


Once the spot is marked, take a Dremel or similar tool and carefully drill out a hole for the light sensor to be exposed. You'll want to make it slightly bigger than the sensor so that you won't get any unnecessary shadows that will affect the code.

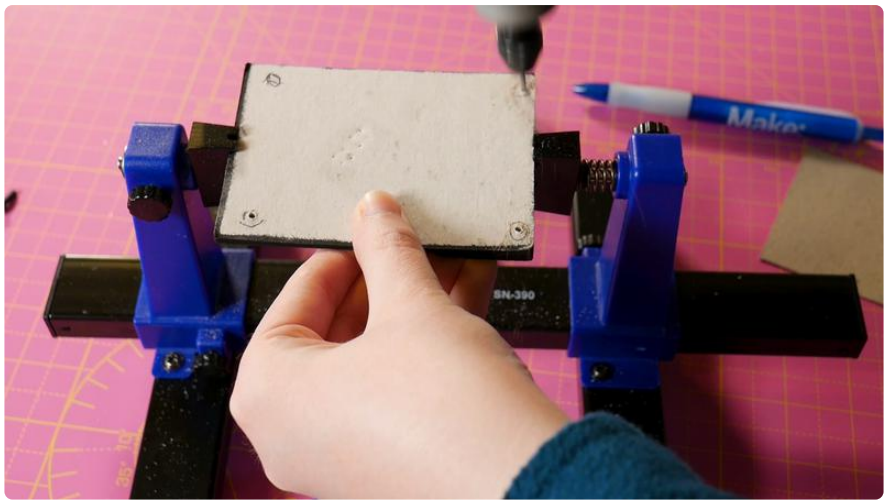
Always be careful and take your time when using power tools!



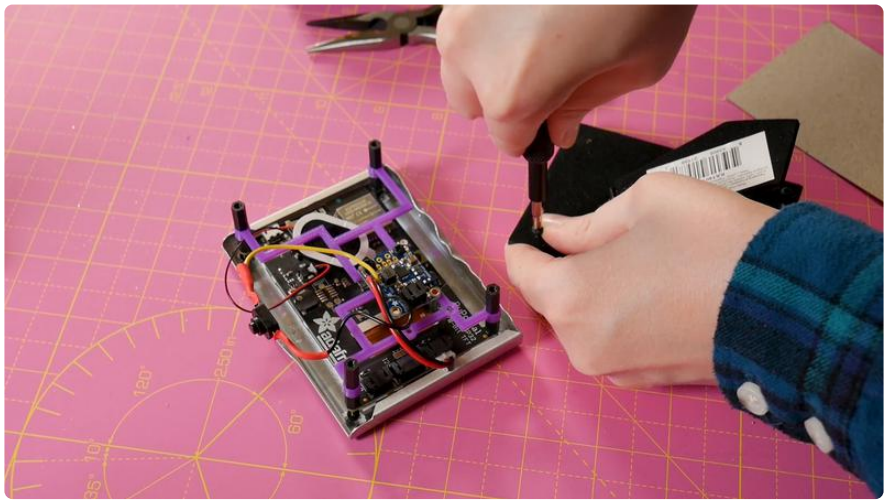
Now the PyPortal can be inserted into the frame, sliding the mounting holes onto the glued in screws. Then take a spacer stand-off and screw it onto the screw. This can be followed by the 3D printed bracket. First though, make sure that your PowerBoost board is mounted onto the bracket. This is followed by a regular stand-off, as shown below.



Take your almost completed PyPortal frame assembly and hold it against the back of the frame. Trace the stand-off's locations on the back of the frame and then using a Dremel or similar tool drill holes that will fit a 2.5mm screw.



After the holes are all set on the back of the frame, screw in 2.5mm screws to connect with the stand-offs on the 3D printed bracket. This completes the frame modification. Since there's so much space between the PyPortal and the back of the frame, you can tuck the speaker and power button behind the PyPortal without having to worry too much about wire management or space.



You now have a creepy, Twin Peaks-themed project. Agent Dale Cooper would be so proud. To reward yourself, why don't you have some coffee, doughnuts and a slice of cherry pie?