Alvin, ART385, ESP32 to Processing, 4/16/20

Statement: For this assignment, we will work with sending data from your ESP32 hardware to Processing.

Based on our in-class work and the GitHub repositories, we will develop physical controls that change aspects of something on the screen. There will be two physical controls, a button (on/off, 0 or 1) and a potentiometer (analog value, 0-255) that control shapes, colors, image tints or some other aspect of graphics on the screen.

Your repository should be called **ESP32ToProcessing** and have two subdirectories: Processing, which contains your .pde file and any related images and ESP32, which contains your .ino sketch.

** You will have unique code, no reusing your self-portrait or slide shows.

** No state machine, just a single screen of at least four interactive elements.

** One of the elements should involve text somehow

** in the README.md file, describe what you are doing in 4 sentences or less

** Potentiometer is on **pin A2**

** Switch is on **pin 12**

** LED for startup or power indication on **pin 15**

Please include a simple **[ART385 Design Document](), which should be saved as a PDF and include the following sections:**
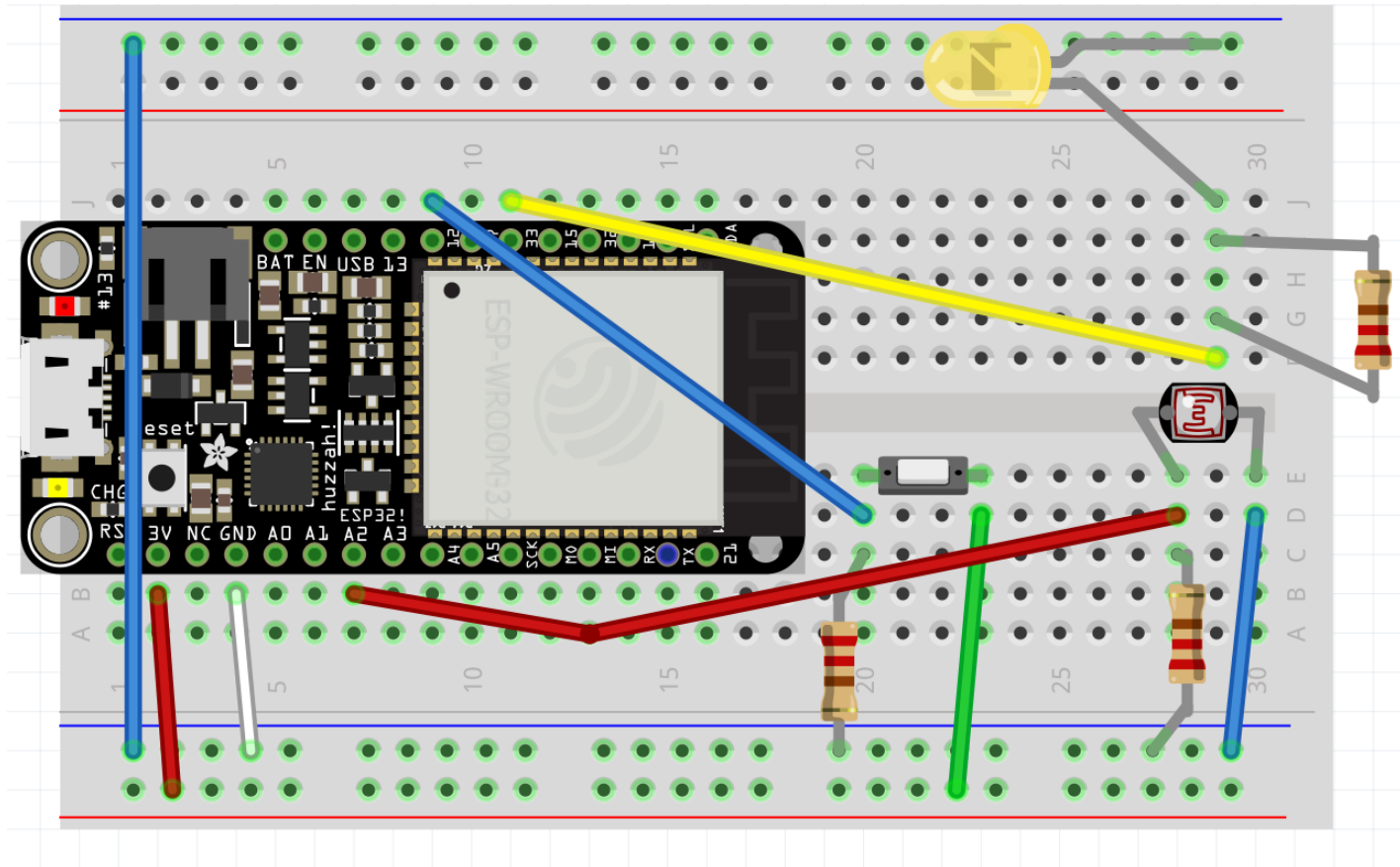
Document Info

Restate the Assignment

Reflections

You will be assessed on code legibility (specifically the comments), working code (we will all use the same hardware pins), GitHub management, and following all the instructions.
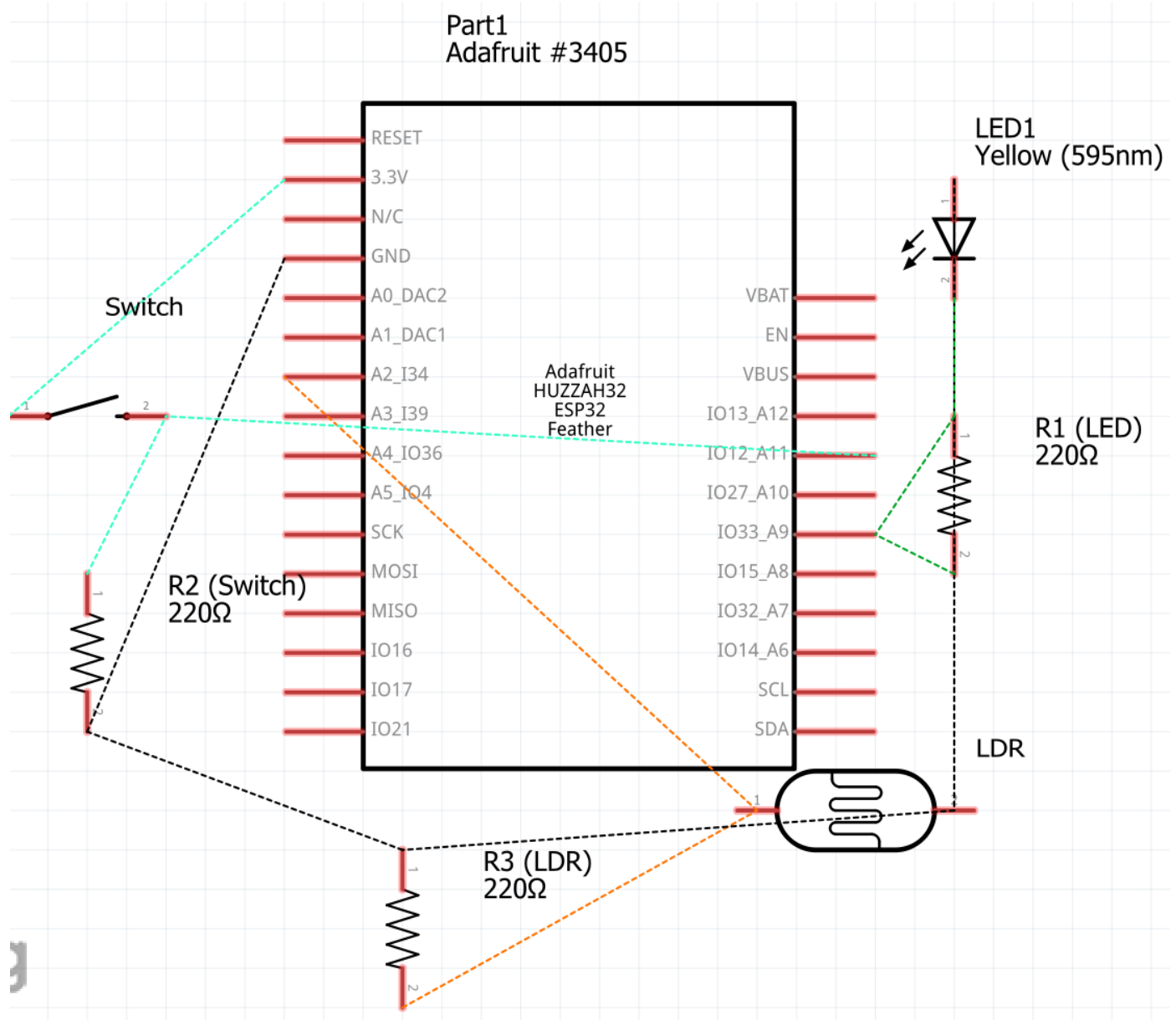
The GitHub repo should be Called **"ESP32ToProcessing"**. And should include a **README.md** and the Art385 Design Document (called **DesignDocument.pdf**). Finally, please include a video of the circuit in action, you pressing the button and turning the potentiometer and the results on the screen. This can either be as a .mp4 file in your repository or Vimeo/Youtube link in the ReadMe file.

Note: no need to do the LICENSE file for any of the Weekly assignments from here on out.

Sketch:

Schematic:

Part1
Adafruit #3405

RESET
3.3V
N/C
GND
A0_DAC2
A1_DAC1
A2_I34
A3_I39
A4_IO36
A5_IO4
SCK
MOSI
MISO
IO16
IO17
IO21

Adafruit
HUZZAH32
ESP32
Feather

VBAT
EN
VBUS
IO13_A12
IO12_A11
IO27_A10
IO33_A9
IO15_A8
IO32_A7
IO14_A6
SCL
SDA

LED1
Yellow (595nm)

R1 (LED)
220Ω

Switch

R2 (Switch)
220Ω

R3 (LDR)
220Ω

LDR

Reflections: My biggest initial problem was trying to get my millis() function to work. I had to go back to my Project 1 and Project 2 to see how I had used my millis() function. However, I had no way to have it revert to its original until I just wrote in some if and if else clauses. One thing I will have to reveal is the use of the millis() function. Just because I used it in both of my previous projects does not mean that it will work the same time on a different sketch. This assignment is definite proof of that. Another issue I will have to work out is how to not have to hold that the button in order to have the light stay on. While I chose not to implement because I still need to work out how to keep track of global variables and how to increment them inside a function, I believe that a button push tracker similar to the one used in my Project 2 should do work in a future assignment.