FIU Engineering & Computing

FLORIDA INTERNATIONAL UNIVERSITY

Home Project #6

Using interrupts, digital output and PWM output

Victor Mendoza

ID: 6348908

EML480 Introduction to Mechatronics

Date of Submission: 19-10-2025

Fall, 2025

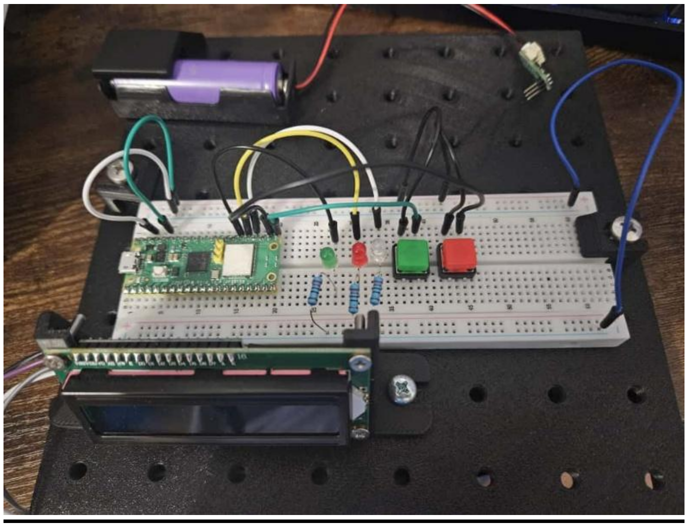
Introduction:

This project explores the use of digital inputs, interruptions, and pulse-width modulation (PWM) on the Raspberry Pi Pico to control LED brightness. Three LEDs and two buttons were connected to the board to implement a brightness control system, where one button increased and the other decreased the brightness of a white LED through 16 defined levels. Additional LEDs were used to indicate maximum and minimum brightness levels.

Materials used:

1. Pico Board
2. Red LED
3. Green LED
4. White LED
5. 3x 220-Ohm resistor
6. 2x Buttons
7. Hook-up wires
8. Breadboard

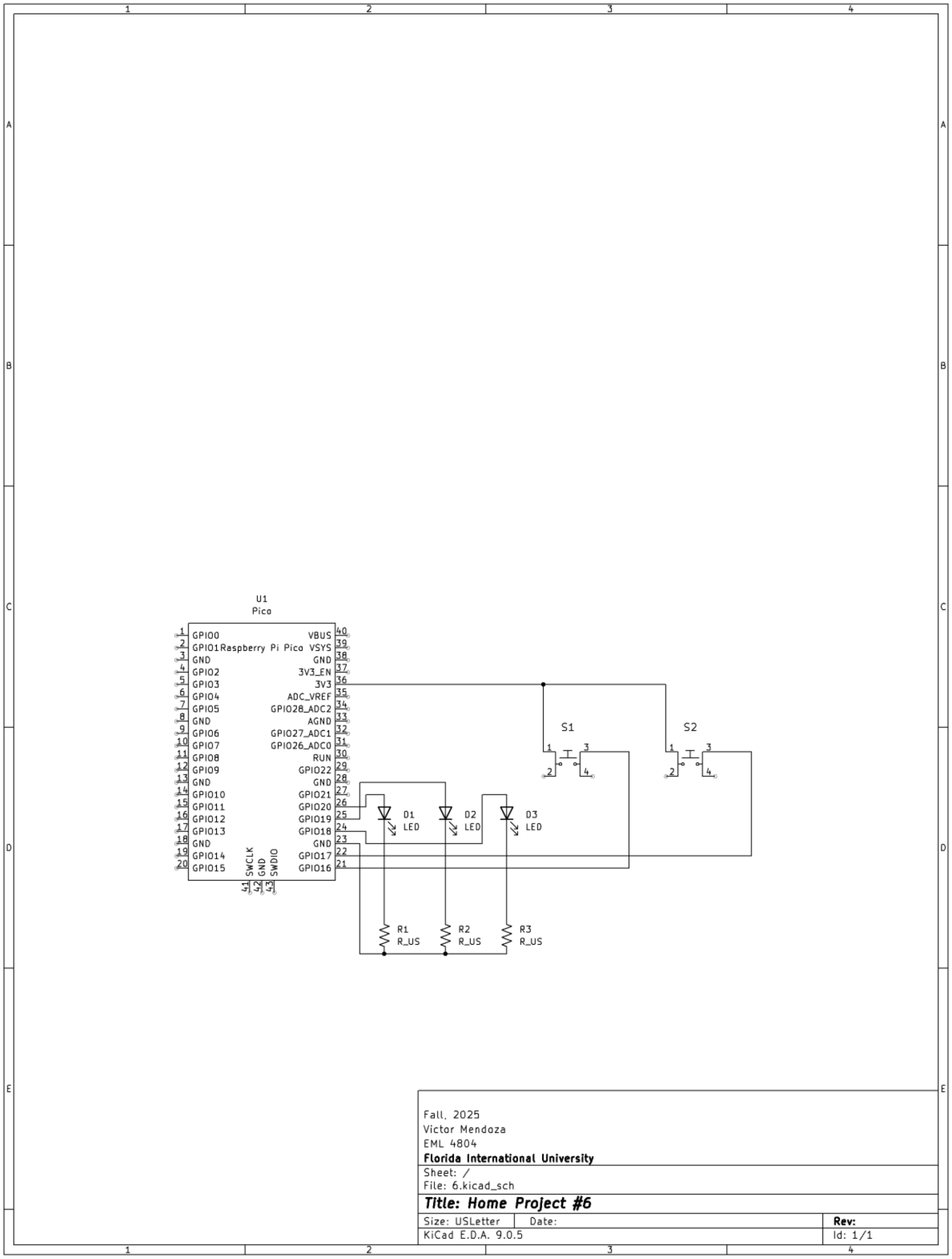
Picture:



Video link:

https://youtu.be/YUSG6jL-RO8

Diagram:



U1
Pico
Raspberry Pi Pico
1 GPIO0
2 GPIO1
3 GND
4 GPIO2
5 GPIO3
6 GPIO4
7 GPIO5
8 GND
9 GPIO6
10 GPIO7
11 GPIO8
12 GPIO9
13 GND
14 GPIO10
15 GPIO11
16 GPIO12
17 GPIO13
18 GND
19 GPIO14
20 GPIO15
40 VBUS
39 VSYS
38 GND
37 3V3_EN
36 3V3
35 ADC_VREF
34 GPIO28_ADC2
33 AGND
32 GPIO27_ADC1
31 GPIO26_ADC0
30 RUN
29 GPIO22
28 GND
27 GPIO21
26 GPIO20
25 GPIO19
24 GPIO18
23 GND
22 GPIO17
21 GPIO16
41 SWCLK
42 GND
43 SWDIO
S1
S2
1
2
3
4
D1 LED
D2 LED
D3 LED
R1 R_US
R2 R_US
R3 R_US
1
2
3
4
A
B
C
D
E
Fall, 2025
Victor Mendoza
EML 4804
Florida International University
Sheet: /
File: 6.kicad_sch
Title: Home Project #6
Size: USLetter
Date:
Rev:
KiCad E.D.A. 9.0.5
Id: 1/1

Code:

```
from picozero import LED, Button
from time import sleep
from machine import Pin

#Button definition
butg = Pin(16, mode=Pin.IN, pull=Pin.PULL_DOWN)
butr = Pin(17, mode=Pin.IN, pull=Pin.PULL_DOWN)

#Led definition
ledw = LED(18)
ledr = LED(19)
ledg = LED(20)

i = 0

ledg.on()
sleep(1)
ledg.off()
sleep(1)
ledr.on()
sleep(1)
ledr.off()
sleep(1)
ledw.on()
sleep(1)
ledw.off()
sleep(1)

while True:
    if butr.value() == 1:
        i = i + 0.0625
        if i > 1:
            i = 1
        while butr.value() == 1:
            ledw.brightness = i
            print(i)

    if butg.value() == 1:
        i = i - 0.0625
        if i < 0:
            i = 0
        while butg.value() == 1:
            ledw.brightness = i
            print(i)

    if i >= 1:
        ledr.on()

    if i <=0:
        ledg.on()

    if i < 1 and i >0:
        ledg.off()
        ledr.off()
```

Conclusions:

The circuit and program were successfully implemented to adjust LED brightness using hardware interrupts. The white LED responded smoothly across all 16 brightness levels, while the red and green LEDs correctly indicated maximum and minimum brightness conditions. Button inputs were detected reliably without delays or interference, confirming proper interrupt configuration. The project effectively demonstrated the integration of PWM control and interrupts on the Raspberry Pi Pico, achieving stable and responsive output regulation.

References:

- Raspberri Pi Pico Library: https://github.com/ncarandini/KiCad-RP-Pico/tree/main
- How to use a breadboard https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all
- Raspberry Pi Pico usage https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html
- LCD wiring instructions https://docs.sunfounder.com/projects/thales-kit/en/latest/micropython/liquid_crystal_display.html
- Using I/O with MicroPython on the Pi Pico https://www.upesy.com/blogs/tutorials/micropython-raspberry-pi-pico-gpio-pins-usage#google_vignette

- Measure an analog voltage with the Pi Pico ADC in MicroPython https://www.upesy.com/blogs/tutorials/micropython-raspberry-pi-pico-adc-usage-measure-voltage
- Control LED brightness with PWM https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico/7