



Home Project #5
LCD, digital and analog inputs, serial communication

Victor Mendoza
ID: 6348908

EML480 Introduction to Mechatronics
Date of Submission: 12-10-2025

Fall, 2025

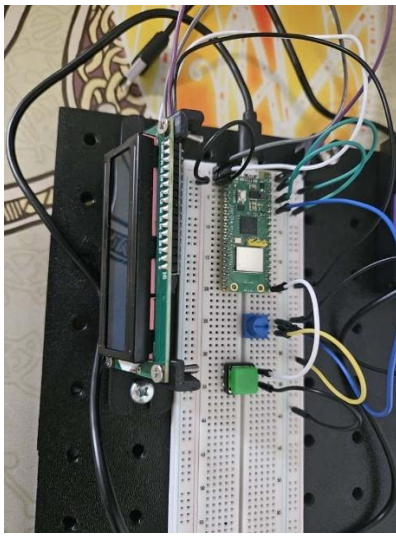
Introduction:

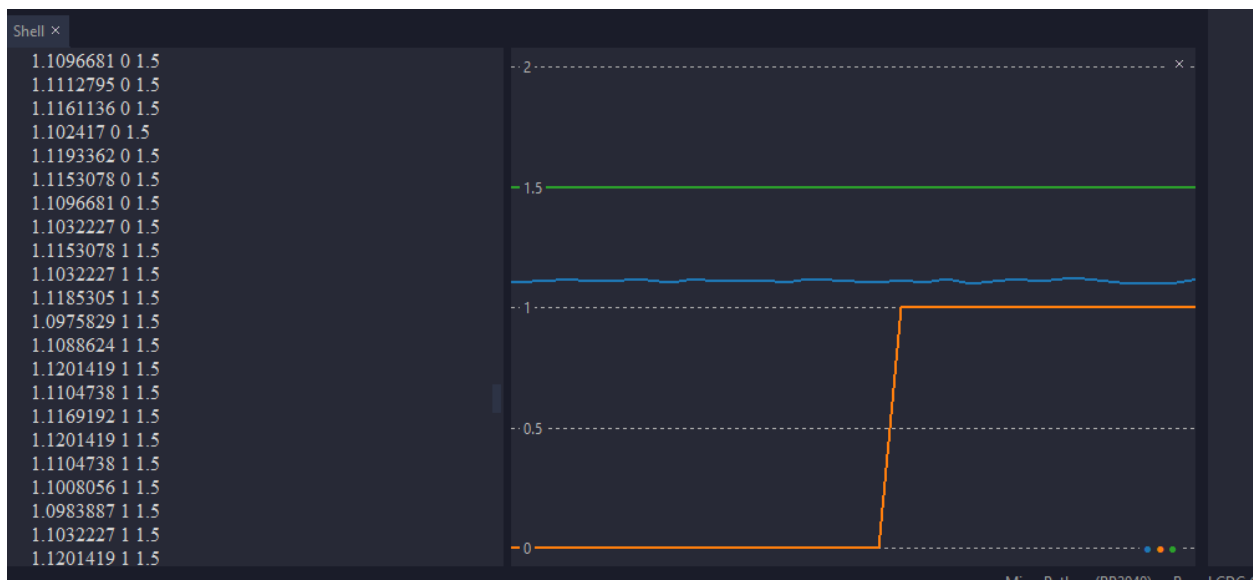
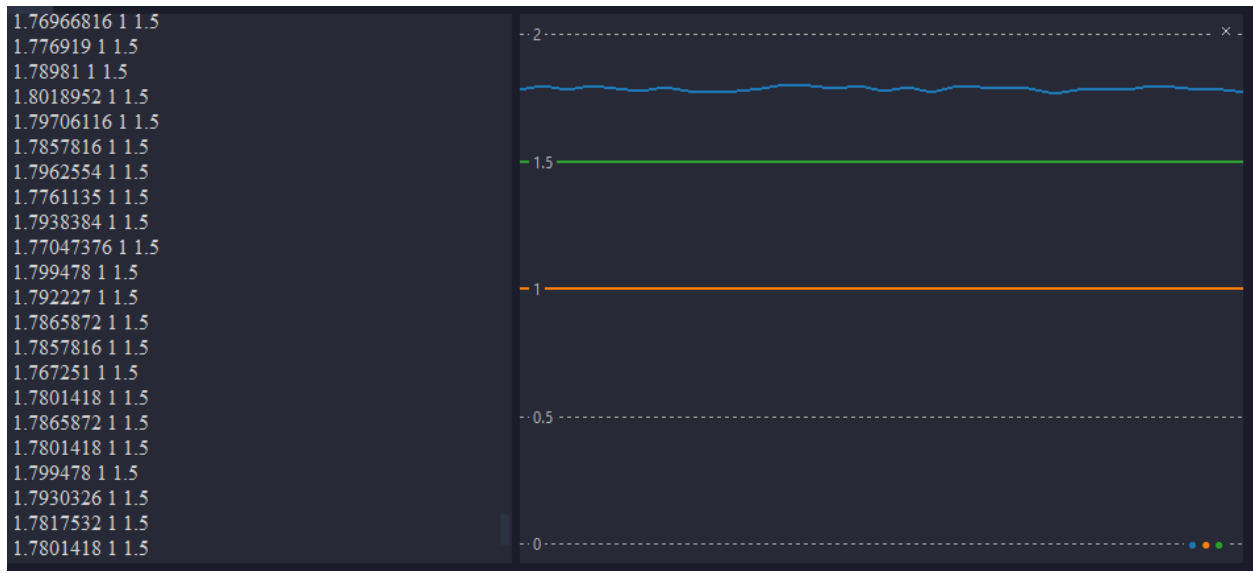
This project focuses on reading and processing digital and analog inputs using a Raspberry Pi Pico and displaying the results on an LCD screen. A tactical switch was used as a digital input to indicate ON/OFF states, while a 10 k Ω potentiometer provided a variable analog voltage read through Pico's ADC. The program converted the ADC values to volts and displayed them in real time, alongside counters for both analog threshold crossings and button presses. Serial communication was also implemented to transmit data to a computer for monitoring and plotting, integrating both hardware control and data visualization.

Materials used:

1. Pico Board
2. LCD Screen
3. 10k ohm potentiometer
4. Tactile switch
5. Hook-up wires
6. Breadboard
7. Wires
8. 3D printed mechanical board
9. 3D printed LCD stand

Picture:

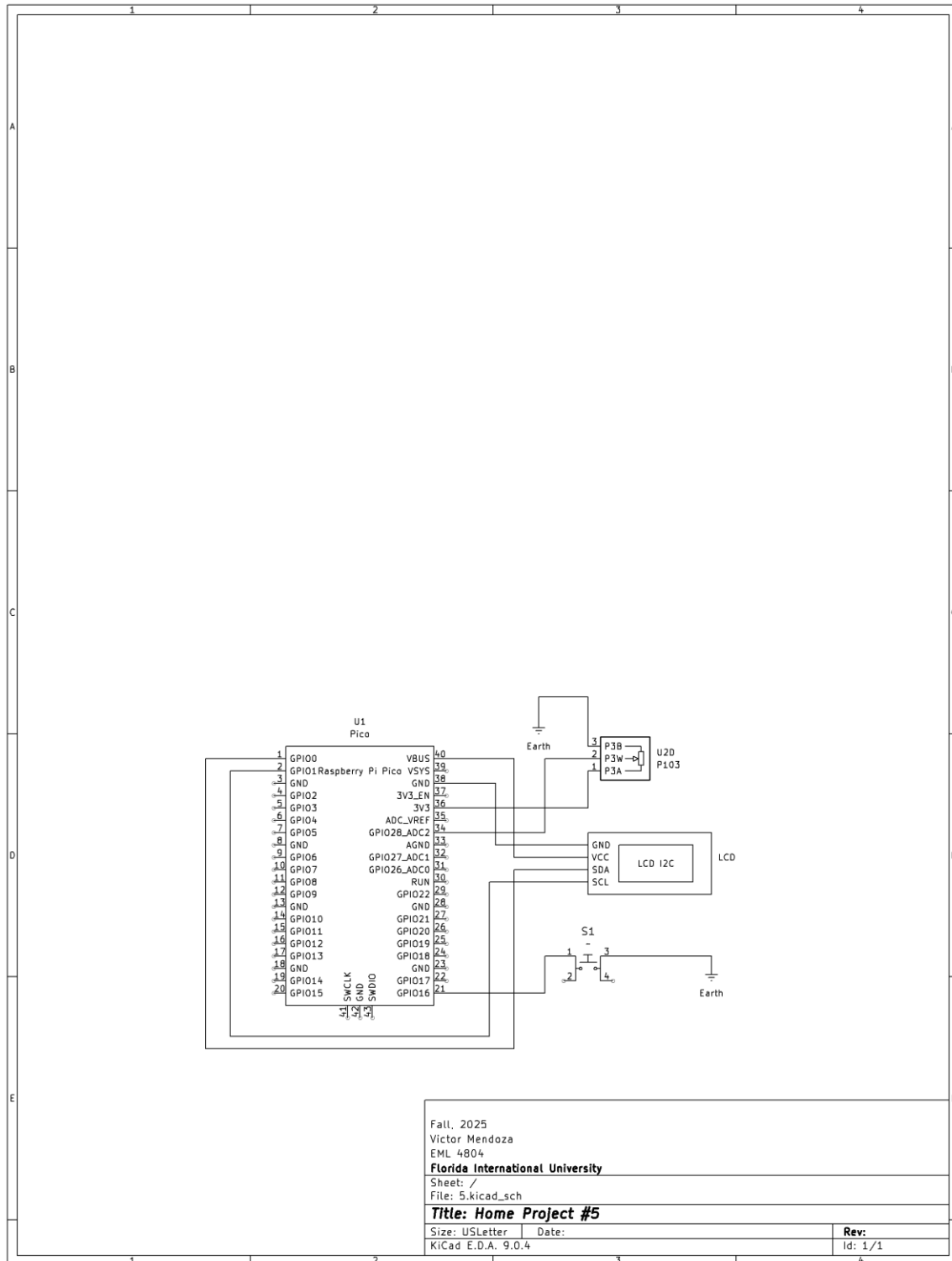




Video link:

<https://youtube.com/shorts/-XSg-YCtdH4>

Diagram:



Code:

```
from machine import Pin, ADC
from lcd1602 import LCD
import utime
import math

# Pin definitions
adc_pin = Pin(28, mode=Pin.IN)
adc = ADC(adc_pin)

#Button definition
pin_button = Pin(16, mode=Pin.IN, pull=Pin.PULL_DOWN)

lcd = LCD()
lcd.clear()

u = 0
i = 0
o = "On "
f = "Off"
m = f"{i}"
n = f"{u}"

while True:
    x = adc.read_u16()
    v = (3.3/65535) * x
    voltage = f"V={v:.2f} V"
    lcd.write(0, 1, f)
    lcd.write(10,0, n)
    lcd.write(10,1, m)
    lcd.write(0, 0, voltage)
    print(v, 1, 1.5)

    if v > 1.49:
        u = u+1
        n = f"{u}"
        lcd.write(10,0, n)
        while v > 1.49:
            x = adc.read_u16()
```

```

v = (3.3/65535) * x
voltage = f"V={v:.2f} V"
lcd.write(10,1, m)
lcd.write(0, 0, voltage)
print(v, 1, 1.5)

if pin_button.value() == 1:
    lcd.write(0, 1, o)
    lcd.write(10,1, m)
    i = i + 1
    m = f"{i}"

while pin_button.value() == 1:
    x = adc.read_u16()
    v = (3.3/65535) * x
    voltage = f"V={v:.2f} V"
    lcd.write(10,1, m)
    lcd.write(0, 0, voltage)
    print(v, 0, 1.5)

if pin_button.value() == 1:
    lcd.write(0, 1, o)
    lcd.write(10,1, m)
    i = i + 1
    m = f"{i}"

while pin_button.value() == 1:
    x = adc.read_u16()
    v = (3.3/65535) * x
    voltage = f"V={v:.2f} V"
    lcd.write(10,1, m)
    lcd.write(0, 0, voltage)
    print(v, 0, 1.5)

```

Conclusions:

The system successfully read and displayed both analog and digital signals on the LCD screen while transmitting real-time data through serial communication. The voltage readings from the potentiometer accurately reflected input variations, and the threshold counter functioned correctly when the analog signal exceeded 1.5 V. The button press counter responded reliably to

single presses, confirming proper use of pull-up input configuration. The project achieved stable LCD updates without redundant refreshes and generated clear serial outputs suitable for plotting, demonstrating full integration of input processing, display, and communication functions using the Raspberry Pi Pico.

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/tales-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>