Home Project #8

DC Motor Control and RPM Meter

Victor Mendoza

ID: 6348908

EML480 Introduction to Mechatronics

Date of Submission: 23-11-2025

Fall, 2025

Introduction:

This project integrates a DC motor, photoresistor, and LCD display with the Raspberry Pi Pico to measure and control motor speed through analog signal processing. Motor speed is regulated using a potentiometer, with the control signal read via an ADC and output to a TA6586 motor driver chip. A photoresistor and interrupter disc are used to detect light variations caused by motor rotation, allowing RPM measurement through rising-edge detection. The system also utilizes capacitors to stabilize voltage, and the LCD displays real-time RPM and power output, demonstrating closed-loop speed control and sensor-based monitoring in a compact mechatronic system.

Materials used:

1. PICO board
2. Breadboard
3. Small DC motor
4. LED
5. 2x 220 Ohm resistor
6. 10k Ohm's resistor
7. 3x 10μF capacitors
8. TA6586 – Motor Driver Chip
9. LCD
10. Improvised Mechanical Assembly
11. Wires

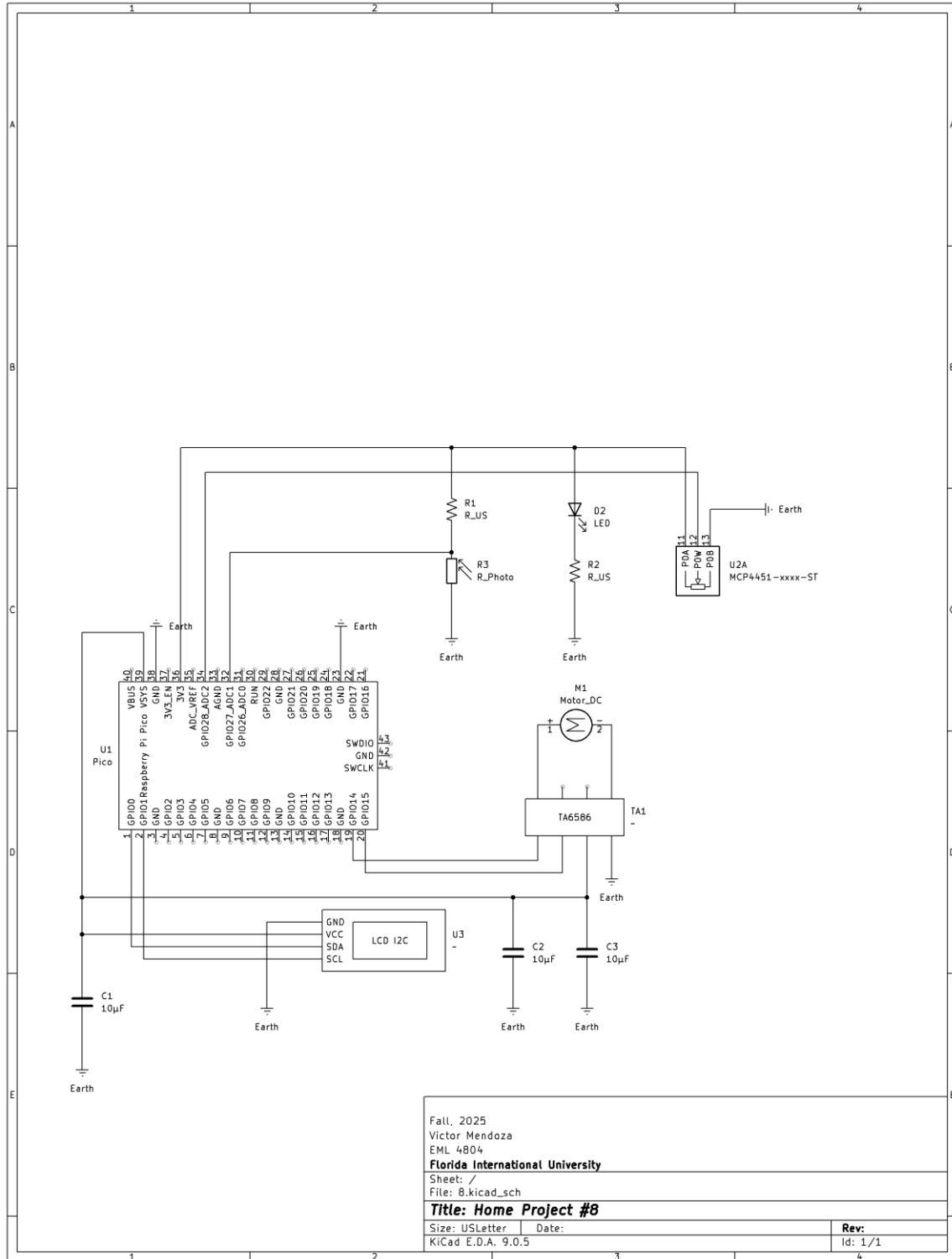## Picture:
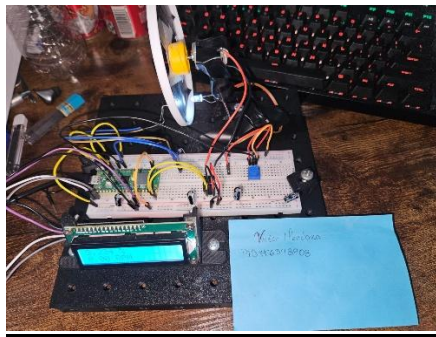
Video link:

https://www.youtube.com/shorts/od79_5wuQas

Diagram:

4

Code:

```
from machine import Pin,PWM
import time
from lcd1602 import LCD

icPin1=Pin(14,Pin.OUT)
icPin2=Pin(15,Pin.OUT)
fPWM=PWM(icPin1)
rPWM=PWM(icPin2)
fPWM.freq(1000)
rPWM.freq(1000)
analog_value = machine.ADC(28)
analog_valuep = machine.ADC(27)

lcd = LCD()
lcd.clear()

i = 0
p = 0
c = 0
v = 0

per = "%"
lcd.write(6, 0, per)

try:
    while True:
        x = analog_value.read_u16()
        z = analog_valuep.read_u16()
        v = (3.3/65535) * z
        rPWM.duty_u16(0)
        fPWM.duty_u16(int(100/100*x))
        a = x / 65535 * 100
        percent = f"{a:.2f}"
        lcd.write(0, 0, percent)

        if c - v > 0.10:
            p = p + 1
            while v - c > 0.5:
                x = analog_value.read_u16()
```

```python
            z = analog_valuep.read_u16()
            v = (3.3/65535) * z
            rPWM.duty_u16(0)
            fPWM.duty_u16(int(100/100*x))
            a = x / 65535 * 100
            percent = f"{a:.2f}"
            lcd.write(0, 0, percent)

            if i >= 1:
                r = p*60/2
                i = 0
                p = 0
                lcd.write(0, 1, " " * 16)
                rpm = f"{r:.2f} RPM"
                lcd.write(0, 1, rpm)

            i = i + 0.01
            utime.sleep(0.01)
            print(v)
            c = v

        if i >= 1:
            r = p*60/2
            i = 0
            p = 0
            lcd.write(0, 1, " " * 16)
            rpm = f"{r:.2f} RPM"
            lcd.write(0, 1, rpm)

        i = i + 0.01
        time.sleep(0.01)
        print(v)
        c = v

except KeyboardInterrupt:
    print("Program Stopped by User")
    rPWM.duty_u16(0)
    fPWM.duty_u16(0)
    time.sleep(.15)
    rPWM.deinit()
```

```
fPWM.deinit()
icPin1.off()
icPin2.off()
```

Conclusions:

The project successfully demonstrated real-time DC motor speed measurement and control using the Raspberry Pi Pico. The analog input from the potentiometer effectively adjusted the motor's RPM, while the photoresistor and interrupter disc combination accurately detected rotational speed through light intensity changes. The RPM values displayed on the LCD matched expected behavior, and the use of capacitors ensured stable motor operation without voltage fluctuations. The system achieved consistent readings up to the motor's measurable limit, validating the design's reliability and efficiency in integrating motor control, analog sensing, and data display.

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
- TA6586 Motor Driver Chip https://docs.sunfounder.com/projects/pico-2w-kit/en/latest/component/component_ta6585.html
- Control DC Motor with Raspberry Pi Pico W and TA6586 https://toptechboy.com/control-dc-motor-with-raspberry-pi-pico-w-and-ta6586/