Home Project #7

RP PICO and MPU 6050 inclinometer

Victor Mendoza

ID: 6348908

EML480 Introduction to Mechatronics

Date of Submission: 02-11-2025

Fall, 2025

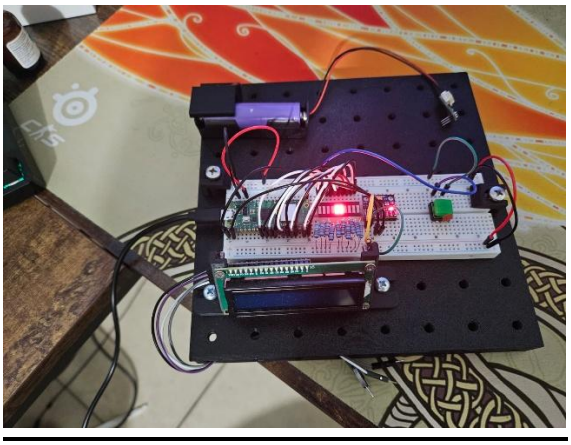
Introduction:

This project focuses on the implementation of the MPU-6050 sensor with a Raspberry Pi Pico to measure and analyze motion through acceleration and angular velocity. Using I²C communication, the MPU-6050's 3-axis accelerometer and 3-axis gyroscope data were processed to calculate roll and pitch angles. The system incorporated an LCD display to present calculated values and an LED bar graph to visually represent inclination. A digital low-pass filter was applied to stabilize the readings, and a calibration function was added to enhance accuracy by compensating for offset errors.

Materials used:

1. LED bar graph
2. 10 x 220-ohm Resistors
3. MPU 6050 Module
4. RP PICO board
5. Button
6. Beadboard
7. Wires

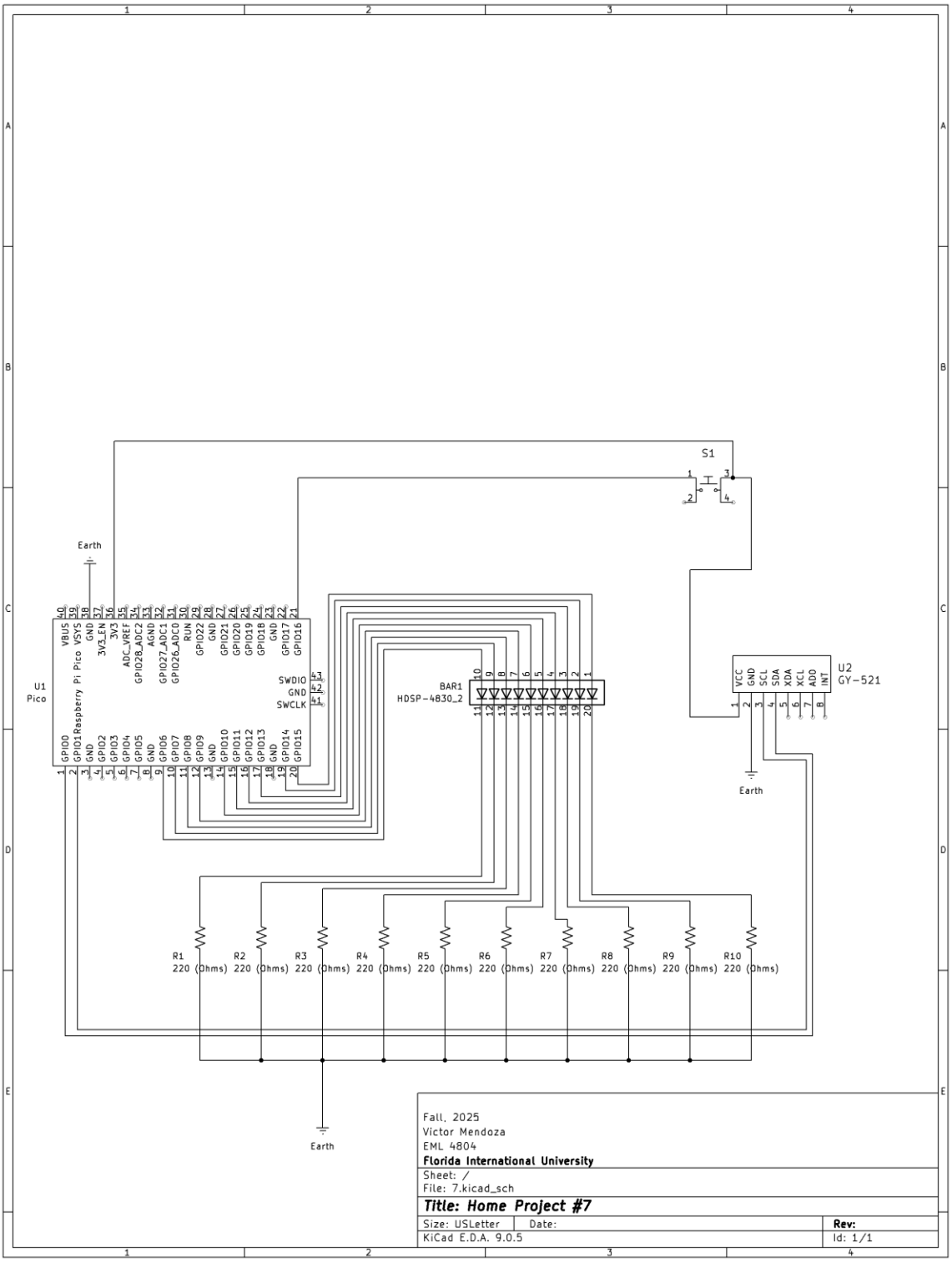
Picture:



Video link:

https://youtu.be/CVRZNNwieRU

Diagram:


S1
Earth
U1
Pico
Raspberry Pi Pico
VBUS
VSYS
GND
3V3_EN
3V3
ADC_VREF
GPIO28_ADC2
AGND
GPIO27_ADC1
GPIO26_ADC0
RUN
GPIO22
GND
GPIO21
GPIO20
GPIO19
GPIO18
GND
GPIO17
GPIO16
GPIO0
GPIO1
GND
GPIO2
GPIO3
GPIO4
GPIO5
GND
GPIO6
GPIO7
GPIO8
GPIO9
GND
GPIO10
GPIO11
GPIO12
GPIO13
GND
GPIO14
GPIO15
SWDIO
GND
SWCLK
BAR1
HDSP-4830_2
U2
GY-521
VCC
GND
SCL
SDA
XDA
XCL
ADO
INT
Earth
Earth
R1
220 (Ohms)
R2
220 (Ohms)
R3
220 (Ohms)
R4
220 (Ohms)
R5
220 (Ohms)
R6
220 (Ohms)
R7
220 (Ohms)
R8
220 (Ohms)
R9
220 (Ohms)
R10
220 (Ohms)
Fall, 2025
Victor Mendoza
EML 4804
Florida International University
Sheet: /
File: 7.kicad_sch
Title: Home Project #7
Size: USLetter
Date:
Rev:
KiCad E.D.A. 9.0.5
Id: 1/1

Code:

```
from imu import MPU6050
from machine import I2C, Pin
import time
import math
import utime

#LED Definition
pin = [6,7,8,9,10,11,12,13,14,15]
led= []
for i in range(10):
    led.append(None)
    led[i] = machine.Pin(pin[i],  machine.Pin.OUT)

butg = Pin(16, mode=Pin.IN, pull=Pin.PULL_DOWN)
alpha = 0.3 #Correction factor
i2c = I2C(0, sda=Pin(0), scl=Pin(1), freq=400000)
mpu = MPU6050(i2c)

def ewma_filter(new_value, prev_value, alpha):
    return alpha * new_value + (1 - alpha) * prev_value

def update_leds(croll):
    [l.off() for l in led]
    on_pairs = [[8,9],[7,8],[6,7],[5,6],[4,5],[3,4],[2,3],[1,2],[0,1]]
    index = int(round(croll)) + 4
    index = max(0, min(8, index))
    for i in on_pairs[index]:
        led[i].on()

#Starts the program at 0
a_x = mpu.accel.x
a_y = mpu.accel.y
a_z = mpu.accel.z
roll = 180 * (math.atan2(a_y, math.sqrt(a_x**2 + a_z**2)) / math.pi)
offset = roll

while True:
        a_x = mpu.accel.x
        a_y = mpu.accel.y
```

```
a_z = mpu.accel.z

roll = 180 * (math.atan2(a_y, math.sqrt(a_x**2 + a_z**2)) / math.pi)

if butg.value() == 1:
    [l.off() for l in led]
    for _ in range(1):
        led[9].toggle()
        utime.sleep(0.2)
        for i in range(9):
            led[9-i].toggle()
            led[8-i].toggle()
            utime.sleep(0.25)
        led[0].toggle()
        offset = roll

croll = roll - offset
filtered_value = croll
filtered_value = ewma_filter(croll, filtered_value, alpha)

update_leds(croll)
print("Filtered:", filtered_value)
time.sleep(0.1)
```

Conclusions:

The MPU-6050 module was successfully interfaced with the Raspberry Pi Pico, accurately measuring and displaying roll and pitch angles derived from accelerometer data. The integration of filtering reduced noise, providing smoother and more stable outputs. The LED bar graph correctly reflected inclination changes within the expected ±4° range, and the calibration button effectively stored and retrieved offset values, ensuring consistent results after restarts. Overall, the project achieved reliable real-time motion detection and visualization, demonstrating effective sensor integration, data processing, and calibration control.

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
- 6-axis Motion Tracking https://docs.sunfounder.com/projects/kepler-kit/en/latest/pyproject/py_mpu6050.html
- Display the Level https://docs.sunfounder.com/projects/kepler-kit/en/latest/pyproject/py_led_bar.html