

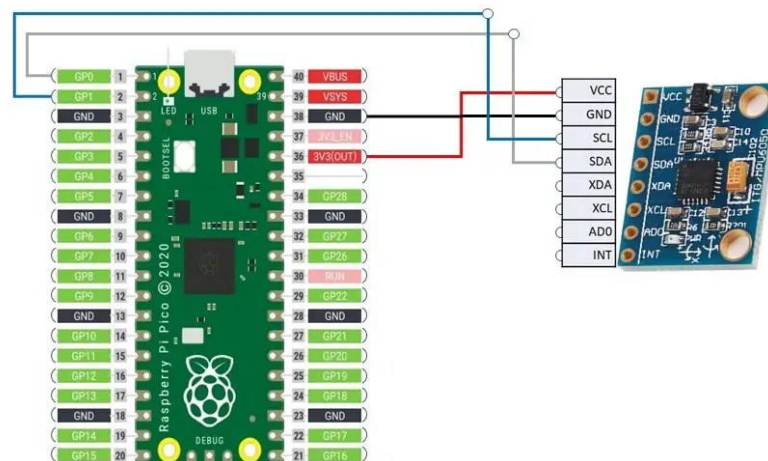
# DAY -5 LAB REPORT 01

**BATCH NO :** 05

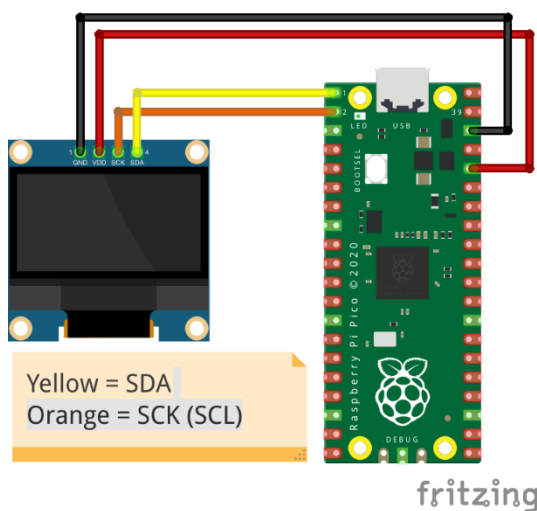
**TITLE OF THE EXPERIMENT :** POSITION SENSING USING RASPBERRY PI PICO

**OBJECTIVE :** To sense the position using GY521 MPU 6050 sensor and display the output on Oled using Raspberry Pi.

**COMPONENTS USED :** Raspberry Pi Pico W Microcontroller , GY521 MPU 6050, OLED, Breadboard



**BLOCK DIAGRAM :**



**DEBUGS IN CODE RECTIFIED:** No debugs as of now

**CODE UPDATES TESTED IF ANY:**

```
from machine import Pin, I2C
from ssd1306 import SSD1306_I2C
from imu import MPU6050
import time

# Initialize I2C for MPU6050 (GP5 = SDA, GP4 = SCL)
i2c_mpu = I2C(0, sda=Pin(0), scl=Pin(1), freq=400000)
# Initialize MPU6050 sensor
imu = MPU6050(i2c_mpu)

# Initialize I2C for OLED (GP0 = SDA, GP1 = SCL)
i2c_oled = I2C(0, sda=Pin(0), scl=Pin(1), freq=400000)

# Initialize the SSD1306 OLED display
oled_width = 128
oled_height = 64
oled = SSD1306_I2C(oled_width, oled_height, i2c_oled)

# Main loop
while True:
    # Read sensor data
    ax = round(imu.accel.x, 2)
    ay = round(imu.accel.y, 2)
    az = round(imu.accel.z, 2)

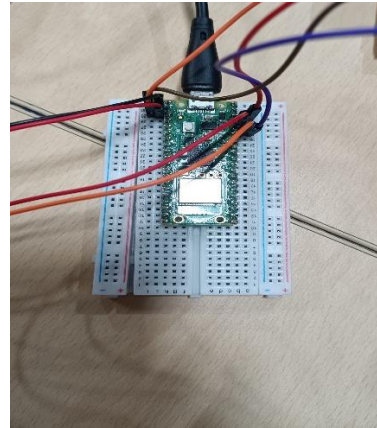
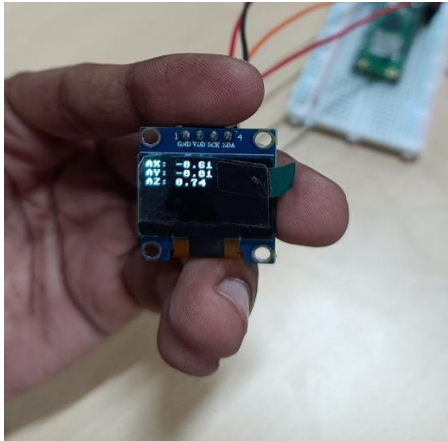
    # Clear OLED display
    oled.fill(0)

    # Display sensor data on OLED
    oled.text("AX: {:.2f}".format(ax), 0, 0)
    oled.text("AY: {:.2f}".format(ay), 0, 10)
    oled.text("AZ: {:.2f}".format(az), 0, 20)

    # Show the data on the OLED
    oled.show()

    # Print data to the console
    print(ax, "\t", ay, "\t", az, "\t", end="\r")

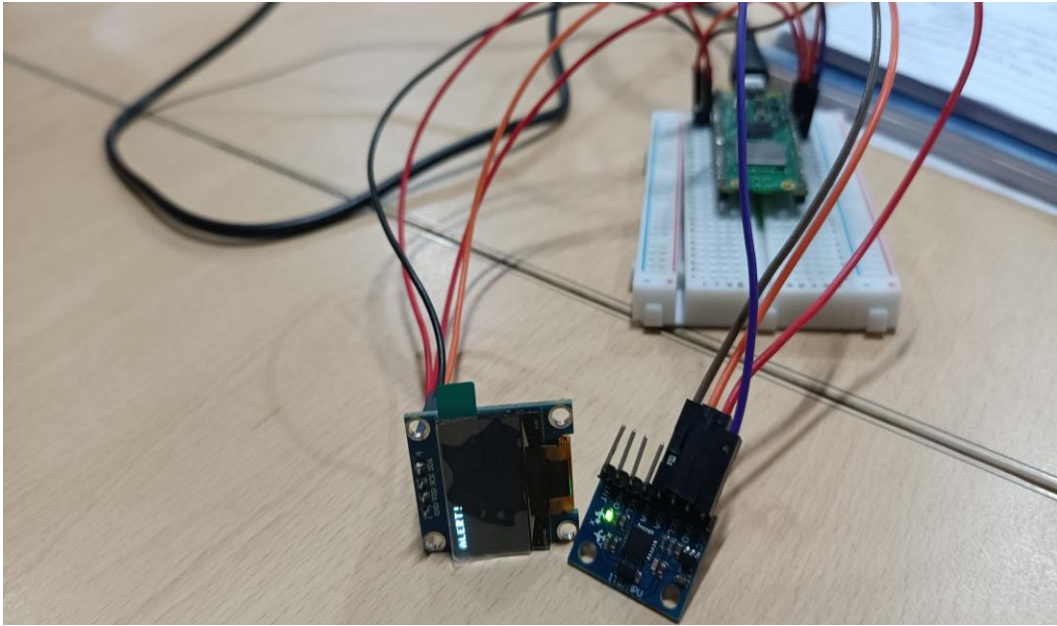
    # Sleep for a while to stabilize the values
    time.sleep(0.2)
```



**APPLICATIONS:** Indoor object tracking, Control of Drones, car navigation systems, robots.

**FURTHER ENHANCEMENT IN OUR WORK IDENTIFIED:**

If the position of the drone is tilted beyond the point, the alert message will be displayed on the OLED.



### Code:

```
from machine import Pin, I2C
from ssd1306 import SSD1306_I2C
from imu import MPU6050
import time

# Initialize I2C for MPU6050 (GP5 = SDA, GP4 = SCL)
i2c_mpu = I2C(0, sda=Pin(0), scl=Pin(1), freq=400000)

# Initialize MPU6050 sensor
imu = MPU6050(i2c_mpu)

# Initialize I2C for OLED (GP0 = SDA, GP1 = SCL)
i2c_oled = I2C(0, sda=Pin(0), scl=Pin(1), freq=400000)

# Initialize the SSD1306 OLED display
oled_width = 128
```

```

oled_height = 64
oled = SSD1306_I2C(oled_width, oled_height, i2c_oled)

# Function to display the alert message on the OLED
def display_alert_message():
    oled.fill(0) # Clear the display
    oled.text("ALERT!", 0, 0)
#    oled.text("X Coordinate", 0, 10)
#    oled.text("is below -20", 0, 20)
    oled.show()

# Main loop
while True:
    # Read sensor data
    ax = round(imu.accel.x, 2)
    ay = round(imu.accel.y, 2)
    az = round(imu.accel.z, 2)

    # Clear OLED display
    oled.fill(0)

    # Display sensor data on OLED
    oled.text("AX: {:.2f}".format(ax), 0, 0)
    oled.text("AY: {:.2f}".format(ay), 0, 10)
    oled.text("AZ: {:.2f}".format(az), 0, 20)

    # Show the data on the OLED
    oled.show()

    # Print data to the console
    print(ax, "\t", ay, "\t", az, "\t", end="\r")

    # Check if z coordinate is below -20 and display alert message if true
    if ax < -0.5:

```

```
display_alert_message()
```

```
# Sleep for a while to stabilize the values
```

```
time.sleep(1)
```