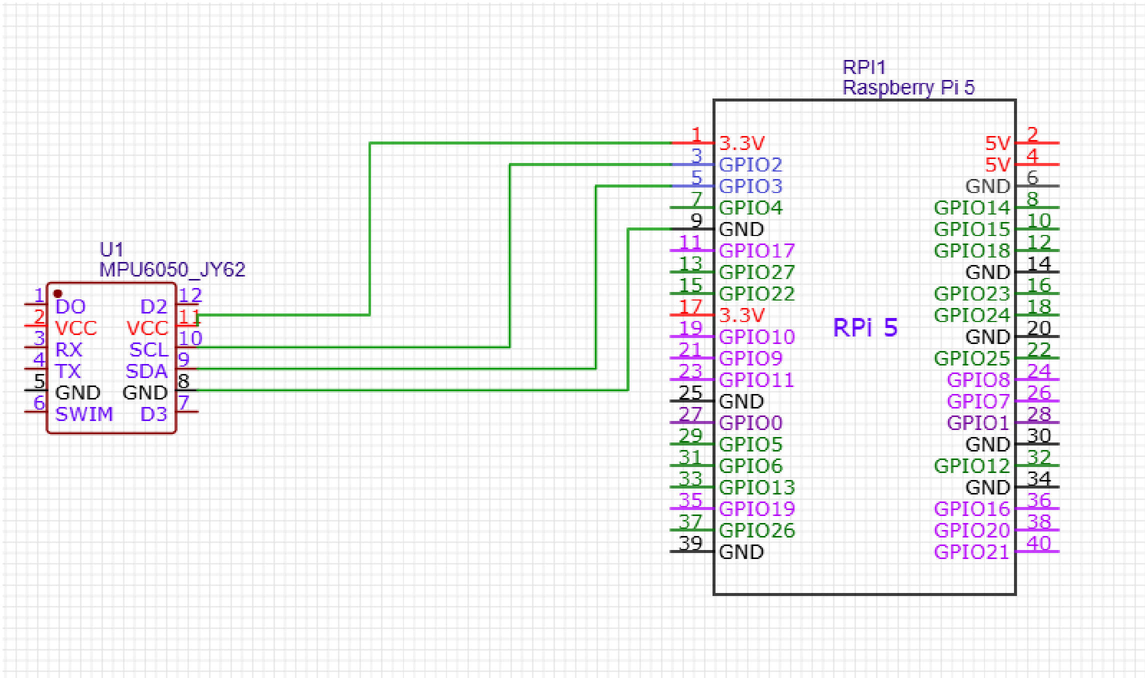
Report

Wiring



Description

1. Introduction

MPU6050 is a device which includes 3 axis gyroscope and 3 axis accelerometer.
We have used ‘smbus’ library in python for I2C communication

2. Imports and Initial Setup

```python
#!/usr/bin/env python3
import smbus
import time
import os

# MPU6050 registers
PWR_MGMT_1 = 0x6B
MPU_ADDR = 0x68

# Initialize I2C bus
bus = smbus.SMBus(1)
```

‘smbus.SMBus(1)’ is used for I2C bus interface.
‘MPU_ADDR’ for device I2C address i.e 0x68.
‘PWR_MGMT_1’ is used to wake up the sensor from sleep.

3. Reading Sensor Data

‘read_word()’ will give high and low bytes and ‘read_word_2c()’ will do the two complement conversion.

```python
def read_word(adr):
    high = bus.read_byte_data(MPU_ADDR, adr)
    low = bus.read_byte_data(MPU_ADDR, adr + 1)
    return (high << 8) + low

def read_word_2c(adr):
    val = read_word(adr)
    if val >= 0x8000:
        return -((65535 - val) + 1)
    else:
        return val
```

The MPU6050 outputs data as 16-bit integers.

4. Waking the Sensor

```
# Wake up MPU6050
bus.write_byte_data(MPU_ADDR, PWR_MGMT_1, 0)
```

This will write ‘0’ to the power management register and wake up the sensor from sleep.

5. Data Collection Loop

This part collects accelerometer and gyro data over 0.1sec duration and for 1 minute. Acceleration is in g and angular velocity is in degrees per second.

```
# Settings
duration = 60.0  # seconds
interval = 0.1   # seconds
samples = int(duration / interval)

# Data lists
time_data = []
accel_x_data, accel_y_data, accel_z_data = [], [], []
gyro_x_data, gyro_y_data, gyro_z_data = [], [], []

print(f"⏳ Collecting {samples} samples over {duration} seconds...")

start_time = time.time()
```

```python
for i in range(samples):
    current_time = time.time() - start_time

    # Read and convert data
    accel_x = read_word_2c(0x3B) / 16384.0
    accel_y = read_word_2c(0x3D) / 16384.0
    accel_z = read_word_2c(0x3F) / 16384.0

    gyro_x = read_word_2c(0x43) / 131.0
    gyro_y = read_word_2c(0x45) / 131.0
    gyro_z = read_word_2c(0x47) / 131.0

    # Debug print
    print(f"[{i+1:03}] Time: {current_time:.2f}s | "
          f"A: ({accel_x:.4f}, {accel_y:.4f}, {accel_z:.4f}) | "
          f"G: ({gyro_x:.2f}, {gyro_y:.2f}, {gyro_z:.2f})")

    # Store data
    time_data.append(current_time)
    accel_x_data.append(accel_x)
    accel_y_data.append(accel_y)
    accel_z_data.append(accel_z)
    gyro_x_data.append(gyro_x)
    gyro_y_data.append(gyro_y)
    gyro_z_data.append(gyro_z)

    time.sleep(interval)
```

This loop runs for 60 seconds, taking readings every 0.1 seconds (totaling 600 samples). The accelerometer data is divided by 16384 (sensitivity scale factor for ±2g range), and gyroscope data by 131 (for ±250°/s range).

6. Data Storage

```python
# Save to file
file_path = "data.txt"
with open(file_path, "w") as f:
    f.write("Time,Accel_X,Accel_Y,Accel_Z,Gyro_X,Gyro_Y,Gyro_Z\n")  # Header
    for i in range(samples):
        f.write(f"{time_data[i]:.2f},{accel_x_data[i]:.4f},{accel_y_data[i]:.4f},{accel_z_data[i]:.4f},"
                f"{gyro_x_data[i]:.2f},{gyro_y_data[i]:.2f},{gyro_z_data[i]:.2f}\n")

print(f"📁 Data saved to {os.path.abspath(file_path)}")
```

After the data collection, results are saved in the data.txt file.