SEMESTER: SUMMER 2023

INTRODUCTION TO ROBOTICS

CSE-461

LAB REPORT - 05

Submitted by : Abdul Munnaf Rafsan

Student ID: 20201008

Section: 08

Group: 05

Department: Computer Science and Engineering

Experiment Number: 05

Experiment name: Controlling of DC Motors using MPU-92/65- a 9-axis motion processing unit

Objective:

In this lab, we were introduced to the concept of an Inertial Measurement Unit (IMU). This device comprises three distinct sensors: a gyroscope, an accelerometer, and a magnetometer. The gyroscope is designed to capture rapid and sharp movements, the accelerometer records angular changes over time, and the magnetometer, in conjunction with the accelerometer and gyroscope, serves to rectify prolonged sensor data drift from the gyroscope and correct its bias.

The process of interfacing the IMU with a Raspberry Pi involved connecting the IMU's pins to specific pins on the Raspberry Pi board. The Voltage (Vcc) pin of the IMU was connected to the 5V pin of the Raspberry Pi to supply power. Likewise, the Ground (GND) pin of the IMU was linked to the Raspberry Pi's ground pin to establish a common voltage reference. The IMU's Serial Data (SDA) and Serial Clock (SCL) pins, integral to the I2C protocol, were combined with corresponding GPIO pins on the Raspberry Pi that supported I2C communication.

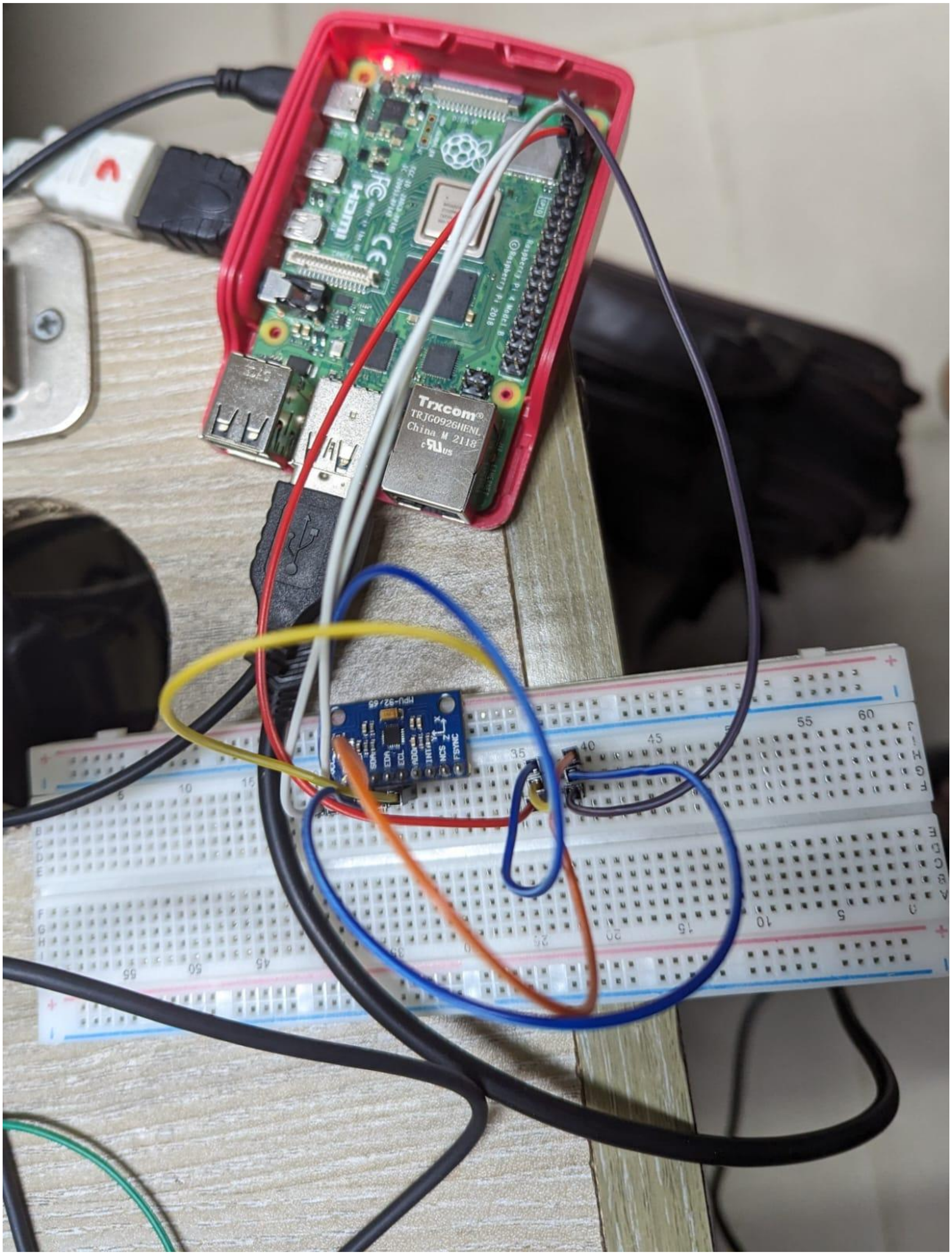
To facilitate the communication between the IMU and the Raspberry Pi, the I2C protocol was employed. This protocol employs two lines, SDA for bidirectional data transfer and SCL for synchronizing the data exchange, enabling seamless communication between devices on the same bus.For the operational aspect, the coding was undertaken in the Python programming language. The Raspberry Pi's built-in Integrated Development Environment (IDE), known as Thonny, was utilized for coding purposes. Thonny facilitated the writing, editing, and execution of the Python code used to interact with the IMU sensors.To sum up, the laboratory exercise encompassed the establishment of a connection between an IMU—comprising a gyroscope, accelerometer, and magnetometer—and a Raspberry Pi using the I2C protocol. This configuration empowered the Raspberry Pi to acquire data from the IMU sensors and process it via Python programming, executed within the Thonny IDE. Such a setup holds significance for a spectrum of projects entailing motion tracking, orientation determination, and related domains.

Components:

For controlling the DC Motors on the Raspberry Pi 4, we need the following electronic components for controlling the DC Motors on the Raspberry Pi 4:

- Raspberry Pi
- DC Motor (1 piece)
- Motor Driver L298N
- Battery 3.7V (2 pieces)
- Battery Case (1)
- Jumper Wire
- Breadboard

Experimental Setup:


Trxcom®
TRJG0926HENL
China M 2118

Code:

```
from mpu9250_i2c import *

time.sleep(1) # delay necessary to allow mpu9250 to settle

print('recording data')
while 1:
    try:
        ax,ay,az,wx,wy,wz = mpu6050_conv() # read and convert mpu6050 data
        mx,my,mz = AK8963_conv() # read and convert AK8963 magnetometer data
    except:
        continue

    print('{}'.format('-'*30))
    print('accel [g]: x = {0:2.2f}, y = {1:2.2f}, z {2:2.2f}= '.format(ax,ay,az))
    print('gyro [dps]:  x = {0:2.2f}, y = {1:2.2f}, z = {2:2.2f}'.format(wx,wy,wz))
    print('mag [uT]:   x = {0:2.2f}, y = {1:2.2f}, z = {2:2.2f}'.format(mx,my,mz))
    print('{}'.format('-'*30))
    time.sleep(1)
```

Result:

As mentioned before python code was written to operate the IMU, the code was about detecting any changes in motion and position of the IMU and giving output in the IDE terminal the x, y, and z values for each of the 3 sensors (gyroscope, accelerometers, and magnetometer) inside the IMU. The values updated constantly due to using a loop so readings for every position were shown regardless of any change in position of the IMU.

Discussion:

This laboratory report provides an in-depth examination of the successful fusion of the MPU9250 Inertial Measurement Unit (IMU) with a Raspberry Pi using the provided code and hardware arrangement. The report systematically outlines the initialization process of the MPU6050 accelerometer, gyroscope, and the AK8963 magnetometer via I2C communication. It further elucidates the process of configuring sampling rates and sensitivity parameters to ensure precise data capture. The subsequent discussion delves into the conversion mechanism, wherein raw sensor data is translated into meaningful metrics such as gravitational acceleration, angular velocity, and magnetic field intensity. A comprehensive explication of the Python script's functionality follows, elucidating its integral role in continuous data retrieval, conversion procedures, and eventual display on the console. The report duly acknowledges encountered challenges during the interfacing procedure and proffers potential remedies. Through an accentuation of practical applications such as orientation estimation and motion tracking, the report underscores the pertinence of integrating the MPU9250 with the Raspberry Pi. The conclusion encapsulates the report by underscoring the significance of this interface while also pointing towards avenues for future enhancements and progressions in this domain.