GE107 - Tinkering Lab

Digital Air Marker

A Smart Device, that uses the capabilities of an Inertial Measurement Unit to track the motion of a marker to identify the text written by the user.

Group: 31_Thurs_2

Instructor: Dr. Basant Subba

Teaching Assistant: Mr. Lavi Jain

Introduction

In this project, a device was developed to recognise digits (0 to 9) written in the air using hand movements. An IMU sensor is used to collect motion data when a person presses a button and draws a digit in the air. This data is then sent to a computer, where it is labelled and used to train a deep learning model. The trained model can then identify which digit was written based on the movement. This project shows how motion sensors can be used to create new ways of interacting with machines without using a keyboard or touchscreen.

Parts Used

ESP32 Development Board

The ESP32 is a powerful microcontroller with built-in Wi-Fi and Bluetooth. In this project, it acts as the central unit that reads data from the motion sensor and sends it to the computer via serial communication. It is responsible for control and communication.

MPU6050 IMU Sensor

The MPU6050 is a 6-axis Inertial Measurement Unit that combines a 3-axis accelerometer and a 3-axis gyroscope. It detects the motion and orientation of the hand when a digit is drawn. The sensor provides the raw data used for gesture recognition.

Push Button (2-pin)

A simple tactile push button is used to signal the start of a gesture. The user presses the button while drawing a digit, and data recording begins as soon as the button is pressed.

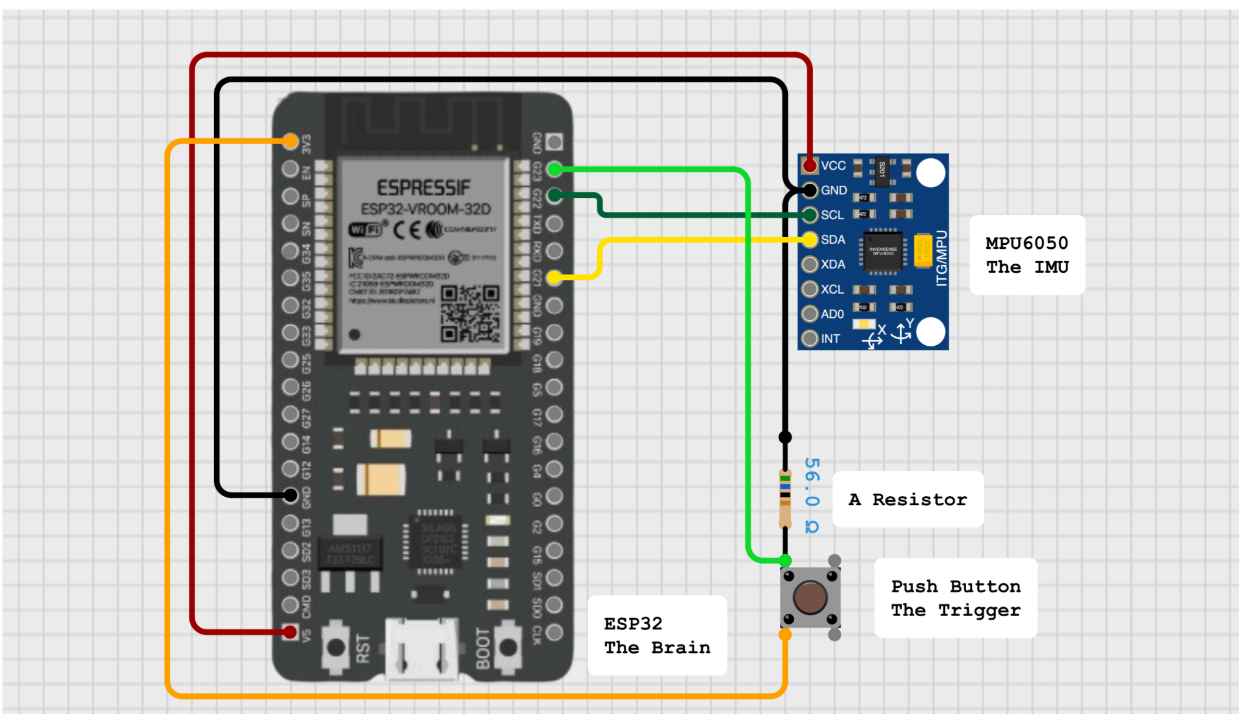
Jumper Wires

These are used to make connections between the ESP32, the MPU6050, and the button. They enable communication and power delivery between components.

USB Cable

The USB cable connects the ESP32 to the computer. It provides power to the ESP32 and also allows serial communication for sending sensor data to the Python script running on the computer.

Circuit Diagram



Challenges We Faced

Data Acquisition Delays and Inefficiency

The ESP32 experienced delays in processing and transmitting sensor data, causing missed samples during high-frequency capture. We optimized the code by reducing unnecessary operations and improving serial communication to ensure timely data transmission.

Calibration Delays

Real-time calibration (using calcOffsets function) was slow and affected usability, so we had to find alternatives to maintain consistent reference frames without recalibrating each time.

Inaccuracy of the model

In the initial stages of developing the machine learning model, we encountered several issues, including a very low accuracy. We could minimise this problem by increasing the amount and quality of training data, as well as by making changes to the architecture of the model and shifting to the LSTM-based approach.

Submitted By:

Nalin Angrish	(2023MEB1360)
Shubham	(2023MEB1380)
Shreshth Shuka	(2023MEB1379)
Shubham Singh	(2023MEB1381)
Nikunj Mahajan	(2023MEB1363)
Priyansh Gupta	(2023MEB1369)