CSC 591/791 ECE 592/792: IoT Architecture, Application, and Implementation

Spring 2022

Assignment 4

G10 Members:

Chaitanya Pawar

[Pavel Koprov](mailto:pkoprov@ncsu.edu)

Rachana Kondabala

Rishi Patel

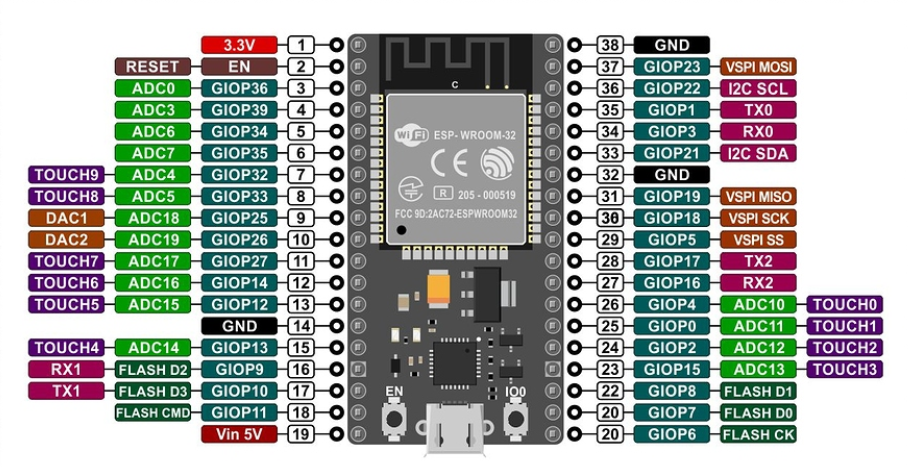
[Sajal Kaushik](mailto:skaushi2@ncsu.edu)

[Thomas Batchelder](mailto:tjbatche@ncsu.edu) 

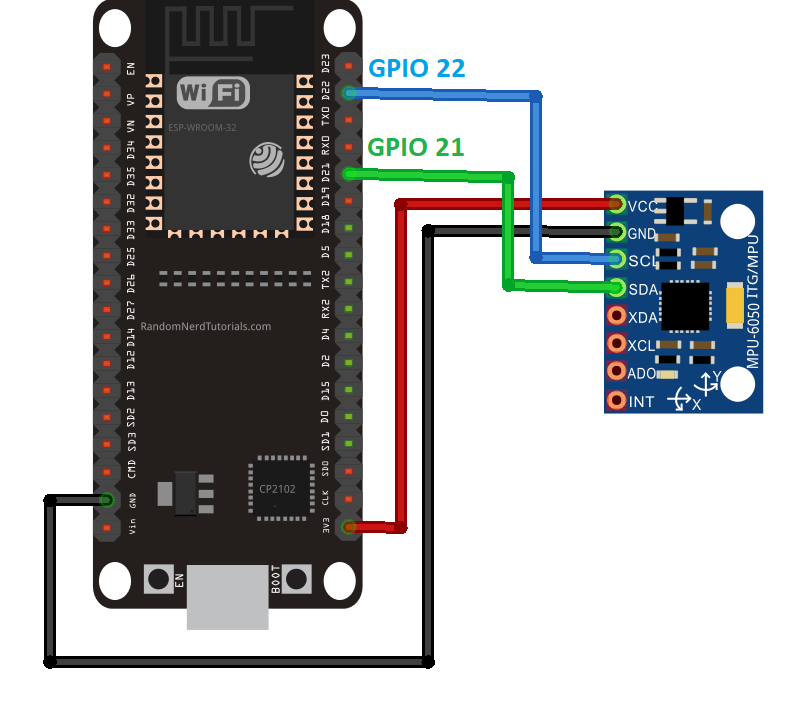
**Team Contribution**

|  |  |  |
| --- | --- | --- |
| Team | Tasks | %1/n, n=6 |
| Chai | - Wire electrical hardware  - Assist in develop IMU code to measure motion of the door  - Assist in setting up communication btw IMU and IoT device | 16.67% |
| Pasha | -Lead/guide team  - Editing scripts  - Maintain github  - Demo | 16.67% |
| Rachana | -Develop IMU code on arduino to measure motion of the door  -Set up coomunication btw MSP32/IMU and MQTT Broker | 16.67% |
| Rishi | - Data Preprocessing  - Script to read data and pass to ML inference | 16.67% |
| Sajal | training the ML model and performing inference | 16.67% |
| Thomas | Lead final project | 16.67% |
| Total | | 100.00% |

**Schematic Diagram of connecting ESP32 devkit to MPU9250**

****

**Figure 1: ESP32 Pinouts**

****

**Figure 2: Wiring Diagram**

**Capturing IMU Data and Publishing to MQTT Broker**

In this setup, when the door starts to move, the IMU measures the acceleration in x and z directions, along with the gyro values about the y axis. The firmware filters the y axis rotation values and publishes all the values to an online MQTT broker. The filter is designed by taking 1 second of y axis gyro values and finding the average and standard deviation of it. The upper limit of the filter is set by the following equation:

                                       µ + 5σ                              Equ 1

Where µ is the average of those values within 1 second and σ is the standard deviation.  The lower limit was calculated by the following equation:

 µ - 5σ                              Equ 2

The filter was used to capture values that fell outside of the upper and lower limits of gyro  values about the y axis. So, when the door moves, the values exceed filter limits and the code publishes all those values to the online MQTT broker. An ML\_model client script was created to subscribe to the topic on the online MQTT broker. This client extracts those values from the MQTT broker and appends the dataframe. Preprocessing of the dataframe is explained in the following section.

**Feature Extraction and Selection**

Feature selection was performed by first slicing the event data into ten slices. From each slice, the mean of acceleration in the X and Z as well as rotation in the Y was extracted. This created a row with 30 features. For training purposes ten samples of open events and ten samples of close events were collected. Data wrangling and preprocessing was done using pandas.

**Classifier Training Method**

We have used Support Vector Machines as a classifier since it works well when there is a clear margin of separation between the classes. We have used the pandas to divide the data for passing into the model at the time of training and testing. The parameter values of C is 1 and kernel is linear which gives the best result on SVM in our case. We have used the trained model loaded with weights to make the final inference for open and close states of the door.

The script with data parsing, feature extraction and classifier inference was implemented in AWS EC2 Cloud9.

Client for reading the door status runs with Python and subscribes to door status topic.