

Project Report
on
PEDOMETER DESIGNING USING MPU6050 GYROSCOPE
SENSOR AND NODE MCU

CPS Lab

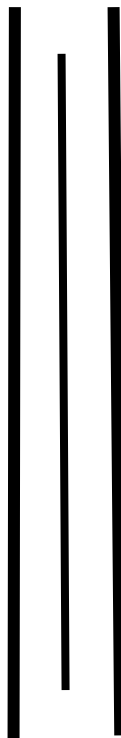
(CH5425)

M. Tech in Cyber-Physical Systems, IIT Madras
Chennai, India, 600036

Submitted By:

Group-2

(KRISHNA SH TELI- GE22M018, BHUVANESH P- MM19B027, SAI KIRAN- ME19B132)



2nd Semester, 2023

Aim

In this project, we targeted to develop a pedometer system which counts the steps as accurate as possible by using gyroscope sensor (MPU6050) integrating with Node MCU microcontroller. We also implemented a classification algorithm to find the type of movements (such as walking, jogging and running) based on the measurement of acceleration vector. Eventually, the step count is pushed onto the cloud (Thing Speak) and displayed in graphical form.

Introduction

A pedometer is a device used for counting the number of steps taken by a person during walking or running. The MPU6050 gyroscope sensor is a popular choice for building pedometers because it can measure changes in acceleration and rotation, which are both important factors in determining steps taken.

The MPU6050 sensor contains a 3-axis accelerometer and a 3-axis gyroscope. The accelerometer measures changes in acceleration in three directions (X, Y, and Z), while the gyroscope measures changes in rotational velocity around each of these three axes. By combining the data from these two sensors, it is possible to accurately determine the number of steps taken by a person.

To use the MPU6050 sensor as a pedometer, the sensor data must be processed using an algorithm that identifies step patterns. One common algorithm is the peak detection algorithm, which identifies peaks in the acceleration data that correspond to steps. The algorithm must also filter out noise and other non-step movements to avoid falsely counting steps.

Once the algorithm identifies steps, it can keep a running count of the total number of steps taken. This count can be displayed on a screen or transmitted wirelessly to a smartphone or other device.

Further movements can be classified into walking, jogging and running based on the acceleration vector by using threshold condition. The number of steps a person takes per minute while walking, jogging, or running can vary based on factors such as age, gender, fitness level, and stride length. However, here are some general estimates for the average number of steps per minute for each activity:

- Walking: The average number of steps per minute while walking is around 100 to 120 steps per minute. This equates to a pace of 2.5 to 3 miles per hour.
- Jogging: The average number of steps per minute while jogging is around 150 to 170 steps per minute. This equates to a pace of 4 to 6 miles per hour.
- Running: The average number of steps per minute while running is around 180 to 200 steps per minute. This equates to a pace of 6 to 9 miles per hour.

It's important to note that these are just general estimates and the actual number of steps per minute can vary based on individual factors.

Hardware Requirements:

- **Node MCU with USB cable**

NodeMCU is an open-source firmware and development kit that is based on the ESP8266 WiFi module. It is widely used in IoT (Internet of Things) projects due to its low cost, small size, and powerful capabilities. The NodeMCU can be programmed using the Arduino IDE, which makes it easy to develop and deploy custom applications.

In the application of a pedometer, the NodeMCU can be used to collect sensor data from a gyroscope or accelerometer, such as the MPU6050, and process that data to count steps taken by a person. The NodeMCU can then transmit this data wirelessly to a cloud server or a smartphone app for further analysis.

The NodeMCU can also be used to interface with other sensors, such as GPS, to provide more accurate location data for fitness tracking. Additionally, the NodeMCU's WiFi capabilities allow it to connect to the internet, making it possible to download and display weather information, news updates, or other relevant data on a user's device.

Overall, using the NodeMCU in a pedometer application can provide a low-cost and flexible solution for step tracking and fitness monitoring. Its small size and WiFi capabilities make it an ideal choice for wearable devices, while its ease of programming and compatibility with Arduino IDE make it accessible to developers and hobbyists alike.

- **MPU6050 gyroscope sensor**

A gyroscope sensor is a type of sensor that measures angular velocity or rotation rate around one or more axes. It is commonly used in a wide range of applications, such as navigation systems, robotics, and gaming controllers, to detect and measure rotational motion.

There are several types of gyroscope sensors, including mechanical gyroscopes, ring laser gyroscopes, and fibre optic gyroscopes. However, one of the most common types used in consumer electronics and robotics is the micro-electro-mechanical system (MEMS) gyroscope.

MEMS gyroscope sensors use tiny vibrating structures or masses, called proof masses, that are suspended on springs inside a sealed chamber. When the sensor is rotated, the proof masses experience Coriolis forces, which cause them to move in a direction perpendicular to the rotation. These movements are detected by sensing electrodes, which measure changes in capacitance or resistance as the proof masses move.

Gyroscope sensors can measure rotational motion around one, two, or three axes, depending on the number of proof masses and sensing electrodes. Three-axis gyroscope sensors, such as the MPU6050, are commonly used in consumer electronics and robotics applications to detect rotation around the X, Y, and Z axes.

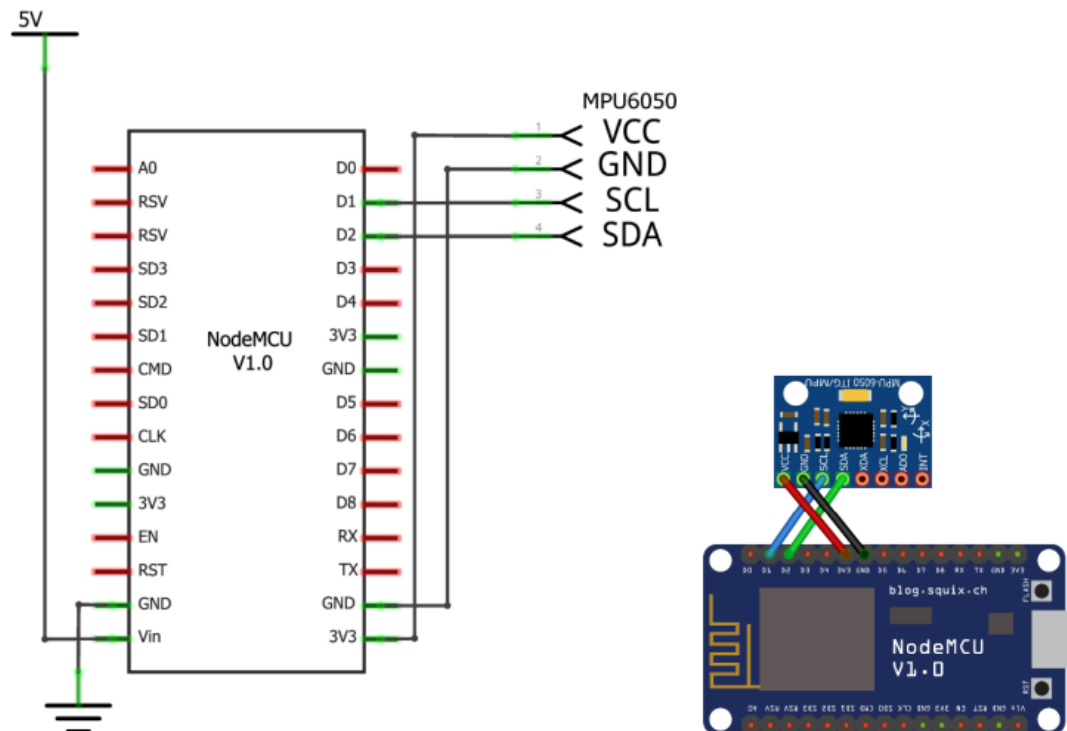
Gyroscope sensors can provide both rate and angle measurements, with the rate measurement indicating the rotation speed, and the angle measurement indicating the rotation angle. However, gyroscope sensors are prone to drift, which means that over time, small errors can accumulate and cause the angle measurement to become inaccurate. To

mitigate this drift, gyroscope sensors are often used in combination with other sensors, such as accelerometers and magnetometers, in a technique called sensor fusion.

Overall, gyroscope sensors are a valuable tool for measuring rotational motion and are used in a wide range of applications. MEMS gyroscope sensors are particularly popular due to their small size, low cost, and low power consumption, making them ideal for use in portable devices and embedded systems.

- **Laptop running either windows, Linux or Mac OS with Arduino IDE installed**

Connection Diagram



Underlying Principle behind the code for counting steps

The overall underlying principle involves detecting peaks. The first trouble encountered while writing the code is making it to detect the movements and not count while being idle. This is done by measuring the acceleration vector every 100 ms and comparing it with the previous acceleration vector in terms of norm. If the norm is greater than a certain threshold, we consider it to be moving. The threshold set was around 2.8 for the acceleration vector. The next part involves detecting steps. At first, we tried to detect acceleration using the certain thresholds in terms of the

acceleration vector i.e., if the acceleration vector is past certain value, we count steps. But this didn't work out accurately as it detected other involuntary movements as steps as well.

So, we tried to capture the rhythmic movement of walking or jogging in terms of acceleration vector. When we walk or jogging there is a rhythmic acceleration and deceleration, so using this knowledge we started counting steps if there is an acceleration followed by deceleration and then acceleration again. This reduced the noise by a big margin. The way we implemented it is by constantly monitoring the acceleration vector every loop and having a marker in the code which capture this increase and decrease of acceleration value. However, it is also not dead accurate and the overall error for every 30 steps taken, is around ± 5 steps.

Next trouble lies in detecting the mode. While we had three modes, only the modes walking and running worked accurately. The idea behind detecting the mode basically involves detecting the speeds of steps. If no of speed of the steps is between 2 and 8 every 3 seconds then we consider walking. If between 8 and 12 we consider jogging and more than that running. However, it is important to note that these are just average numbers. If you walk really fast then it will be classified as jogging but is a bug which isn't taken care of. One more thing to note is that all these values are optimized for movement along the hip region hence keeping the device close to the hip region is necessary for it to be more accurate.

Cloud update

In the beginning we updated the step count to a think speak account via a channel created every 20 seconds but it didn't feel good as the update rate was very low. So, we utilized another model wherein we try to update the data to a local computer system. This involves using the python module called socket which reads data from the IP address specified. Here the update can be as low as 1 seconds letting the data to flow faster. The downside to this is that both the computer and the pedometer device needs to be connected to the same network connection. The data received by this local cloud is the number of steps and mode of activity i.e., walking or jogging.

Procedure

For the working of device, we use power bank as the power supply for powering up the device. Ensure that the Arduino code is uploaded to the device before powering up. Simultaneously run the python script in the computer for the local cloud reception of data. As soon as the device is powered up, it starts counting the steps and sends the data to the cloud. To reset the device, switch off the power supply and restart it again. Ensure that the both your local PC and the device is connected to the same WIFI connection and change the IP address accordingly in the Arduino code. Make sure the device is aligned along your hip for more accuracy and the device is good to go.

Algorithm Flowchart

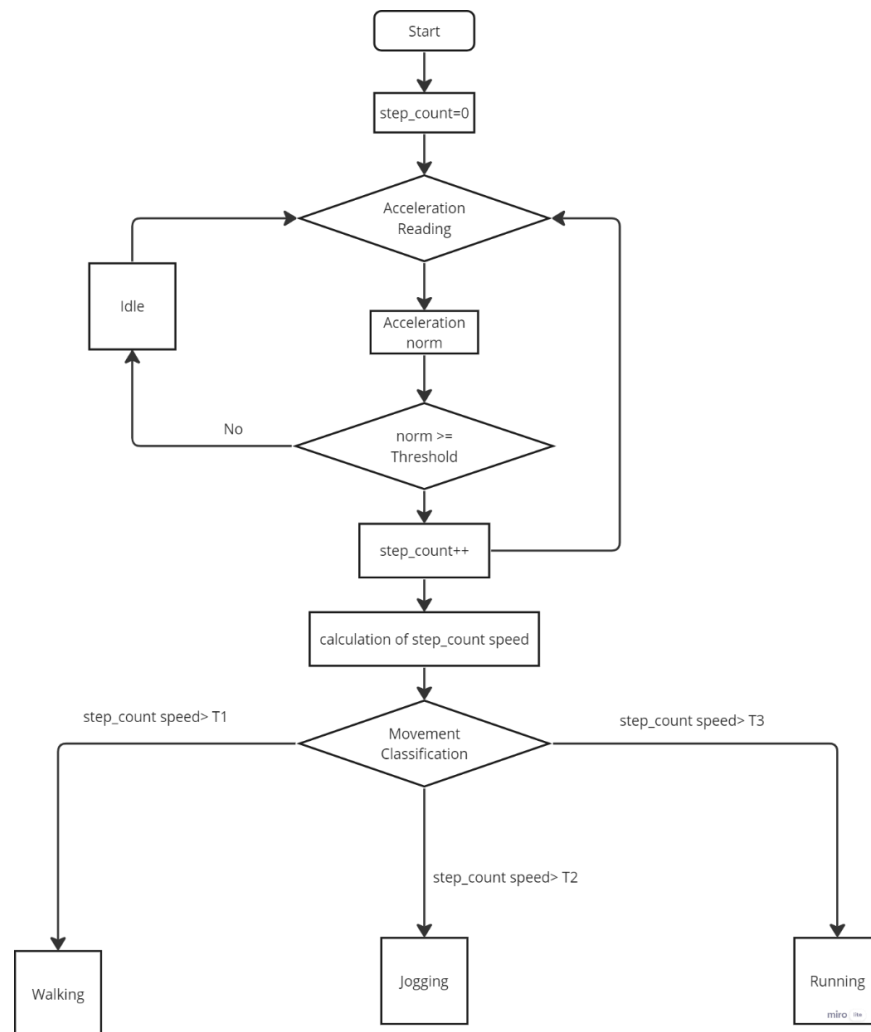


Fig. Workflow of Algorithm

Enclosure Design

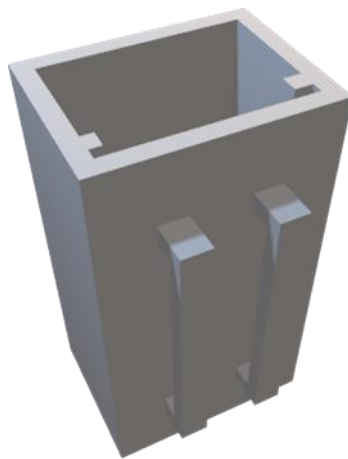


Fig. Enclosure design

We designed an enclosure for MPU using fusion 360. For that, we took appropriate dimensions. The length and breadth of enclosure are 34 and 28mm respectively. We also gave 2-3 mm clearance. We designed a slot for the smooth passage of MCU into the enclosure. Height of the slot is 4 mm. After that we also designed hook for belt, so that we can wear the device. And dimension of the hook is 46*7 mm. Finally, we 3D printed the enclosure.

Observation and Results

The device is fairly accurate. For every 30 steps taken, the accuracy is about ± 5 steps. For walking the accuracy is slightly better compared to jogging. When running, you have to run really fast to detect it otherwise it takes it as jogging. When you walk at a faster pace too, it classifies it as jogging. The data is uploaded to the cloud every 3 seconds. Giving the device considerably good amount to detect the motion felt necessary and hence we couldn't go any faster although it is possible to update it once in every second as well.

Conclusion

In this project, a pedometer is developed using NodeMCU and Gyroscope sensor and demonstrated in real time. Threshold based approach has been implemented in the algorithm. Its accuracy is found to be ± 5 in 30 steps which is fairly good. Based on the speed of steps, the movement has also been classified into walking, jogging and running. The steps count data is sent to local device using WIFI IP address for further analysis. There is future scope in this project regarding improving the step count using FFT and identifying the peak frequency.