Smart Machine State Informer

CSE 321 (Real Time and Embedded Operating Systems)

University at Buffalo

Redwan Ibne Seraj Khan

rikhan@buffalo.edu

Table of Contents

Executive Summary

Smart Machine State Informer is a device that checks the state in which a machine is at and informs the user when a machine has finished completed its task. It senses the vibration of a machine and based on that it tells whether a machine is in the on state, pause state or off state. It can be useful in helping people with their laundry in their daily life. Many people will forget to collect their laundry after the washing machine has finished washing clothes. As a result, the clothes will become smelly. Moreover, if more people are using one washing machine it is difficult to manage time of washing if one forgets to collect one's clothes. As for me, I faced this problem when I first arrived in USA and had to share a washing machine at the basement with other people. To solve this problem, I came up with the idea of building an embedded device that would notify me in real time when the washing machine or the dryer has finished washing/drying the clothes so that everything can be managed efficiently in time. As I was working on this device, I noticed the fundamental aspect of vibration, which every machine does when it operates. I decided to use this aspect to build the device that I was planning to make. I used a Raspberry Pi and integrated an accelerometer sensor with it to detect the vibrations to the minute fractions of seconds. The accelerometer used is a MPU6050 gyro accelerometer which is a 6-axis motion tracking device designed for low power, low cost and high performance requirements of smartphone, tablets and wearable sensors. I configured the Raspberry Pi with codes to detect the WIFI network that is around the place and connect to it. It was found out that the device could be used with not only washing machines but also any other machines. The algorithm used in this device checks the vibrations and makes an average of the number of values given by the accelerometer sensor. Afterwards it compares those values against the threshold values calibrated for each particular machine. Based on the readings it determines whether the machine is on or off and sends messages accordingly.

Project Objectives

1. Making an embedded system designed for sensing the state of different appliances.
2. Sending Real time feedback to the user based on calculations done with the help of pre-defined algorithms.
3. Automating the process of receiving information about the state of a machine to make the lives of general people easier and thus having a vast impact on the large mass of people.
4. Ability to effectively notify the user over the internet through email which will add to the creative aspect of the project.
5. Providing step wise calculations and denotation of stages at the console level and finally sending the end results to the user through email.

Project Approach

First step of the project was to choose the problem to solve. Likewise, this device was built with the intention of informing a user about the exact time when his laundry will be finished so that he would be able to collect it in time. Later on it was found that this device could be used to know the state of any electrical appliance like ovens which vibrate while it operates. After identifying the problem, it was necessary to know which hardware to use. I chose Raspberry Pi to device the system. The application was developed in phases in order to ensure I could prototype the device and test it before working on its extensions. At first I set up the Raspbian operating system after buying the kits. Afterwards I wrote small programs to test whether everything is working properly before starting the project. The programming language for development was C. However, afterwards Python was combined with the C language to provide the feature of sending email notifications to the user. main steps that lead to the completion of the project are given below. I tested whether the Gyro Accelerometer was getting readings. It was then necessary to configure the internal settings of the Raspberry Pi so that I could connect it to the home WIFI without the Ethernet cable. After checking that everything is working properly I gradually started building the algorithm bit by bit. I tested each portion of my code with print statements to check whether each part is working properly. After getting everything done I made a demo video to present my project.

Project Description

This device was made comprising a Raspberry Pi and an accelerometer sensor. At first the device is calibrated according to the vibration readings of the machine to be operated upon. The MPU6050 Gyro Acceleration sensor reads out the values of acceleration of the machine along the different axes based upon the vibration. The function read_word_2c reads the values from the appropriate port of the accelerometer. The get_x_rotation and get_y_rotation finds the amount of rotation in the x and y axes. The readSensor function gets these values using those functions and converts those tiny values into analyzable data using an algorithm. The function uses those data to calculate the average of numerous reading within fractions of seconds to determine whether is vibrating or not. If the machine is vibrating the mode is calculated as 1 and if it is not vibrating the mode is calculated as 0. The readSensor functions sends those values to a Finite State Machine which figures out whether the machine is in off state, on state, pause state or end state based on the values of mode at different times. If the machine goes to the end state, the program calls the sendingEmail function to send an email to the user over the internet informing about the completion of the task.

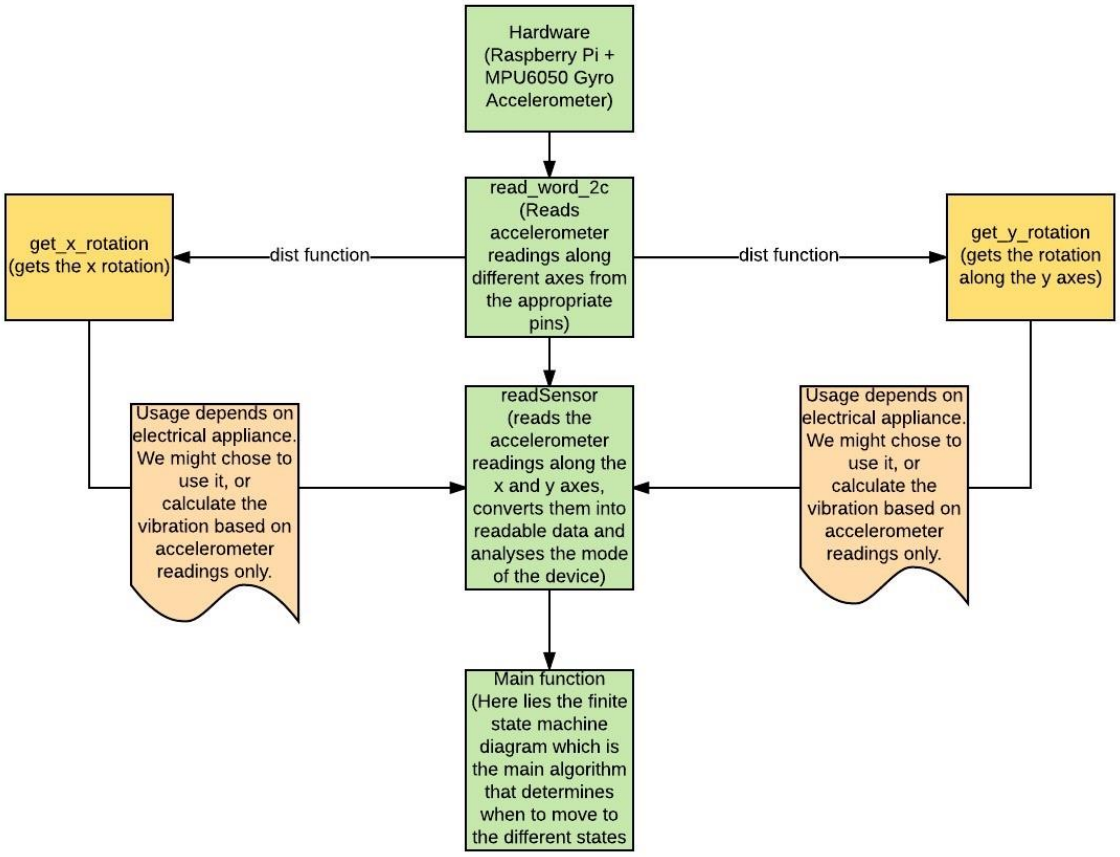

Figure: A flowchart of how the system operates using the different functions

Design Details

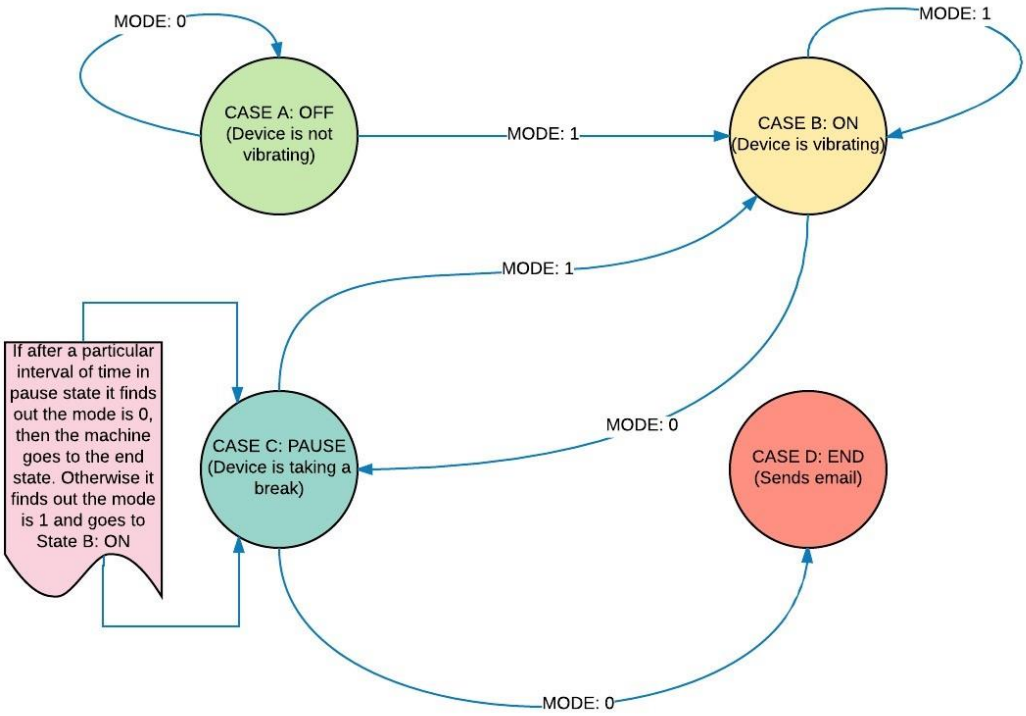

Figure: A finite State Machine describing how the system moves through the different states.

User's Manual

The user at first has to calibrate the device to suit the machine he is using the device for and set the threshold values for vibrations accordingly. Afterwards he has to log into the Raspberry Pi server. He can find out the IP address using a Network Scanner. He has to configure the sendingemail.py and change the email and password field with his email and password. He can then run the program and wait for his notification via email.

Programmer's Manual

The main steps that led to the completion of the project are given step by step below.

1. It was first necessary to collect the following parts.
    - Raspberry Pi 3
    - 5V micro-USB power supply
    - USB keyboard

- USB mouse
- microSD card
- microSD USB card Reader
- A monitor or TV that supports HDMI or composite video
- An HDMI cable or composite video cable
- An Ethernet cable
- MPU6050 Gyro Accelerometer

Figure: Components necessary for starting the project in one frame

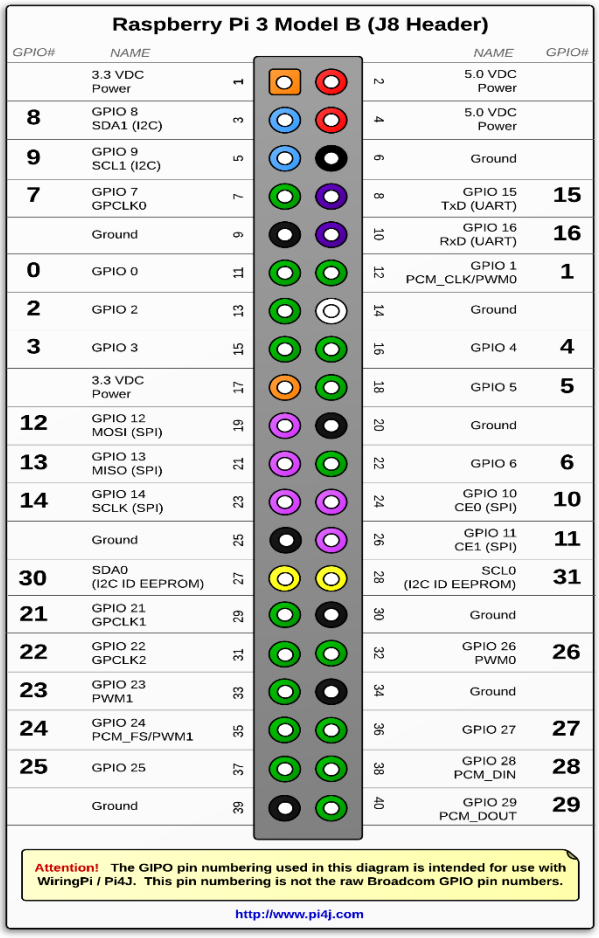

Figure: Raspberry Pi Pin Number Information

Figure: MPU6050 Gyro Acceleration Sensor

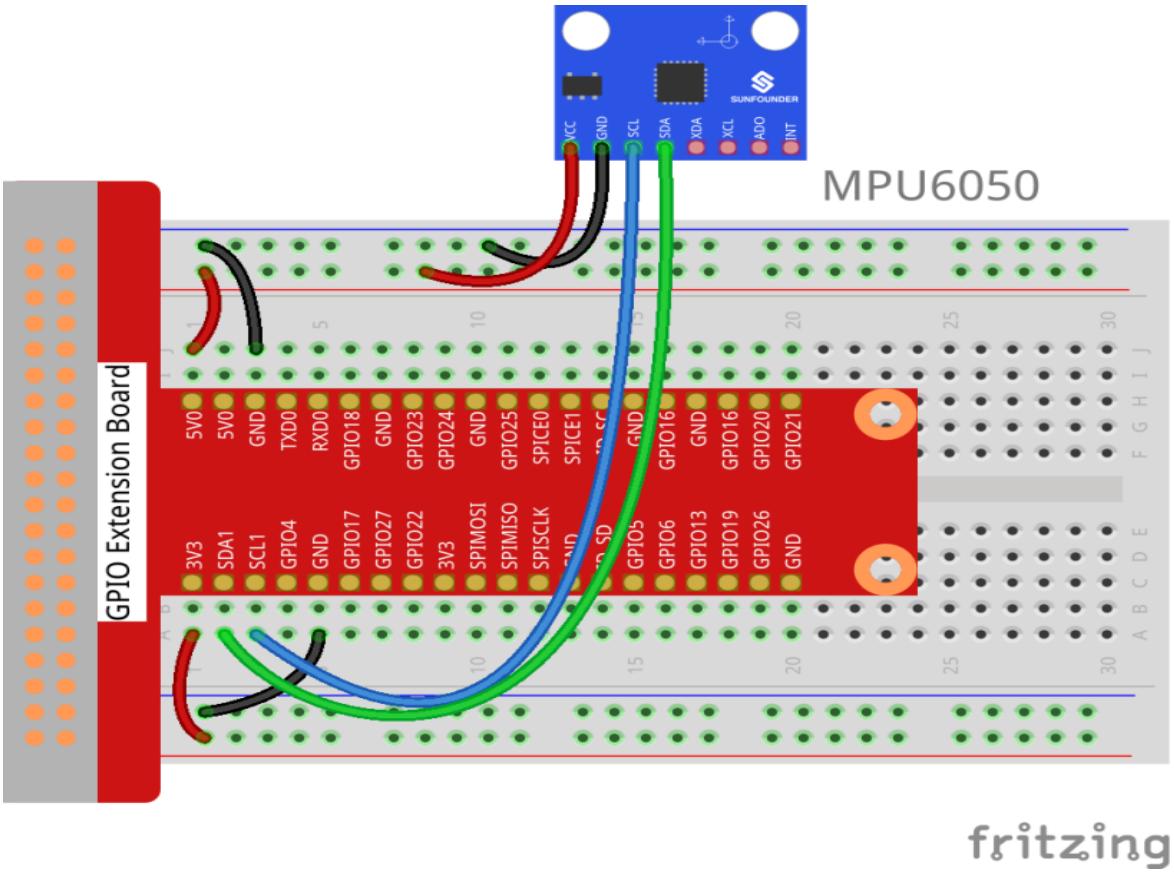

Figure: Connection of the Raspberry Pi with MPU6050 Acceleration Sensor

2. Reformatting the Micro SD card

3. Downloading NOOBS into Micro SD card

4. Connecting the Micro SD card into the Raspberry Pi and installing the Raspbian Operating system

5. Configuring the Raspberry Pi according to our needs.

6. To make the process easier, it was decided to operate the Raspberry Pi using the terminal. For that I decided to use “Network Scanner” application to know the server address of the Raspberry Pi and then access it.

7. To ensure that the Raspberry Pi can run without the help of Ethernet cable, I configured the internal system so that it can get connected to the WIFI network easily.

8. I began chalking out a design on the basis of which the device will operate and then drew a diagram.

9. I started to develop the algorithm into code.

10. After developing the initial algorithm using C code, I integrated python code into the C code to ensure the process of sending messages via email. This ensured the automation of the device and thus ensuring a truly remarkable embedded system capable of transferring information in real time.

11. I tested the device at first using a chair as a representative of any kind of machine. I vibrated the chair to see if the device was operating efficiently.

12. After a successful attempt with the testing of the device, I tested it with a washing machine and it was found to give correct results.

References

1. https://github.com/kootenpv/yagmail
2. https://www.sunfounder.com/learn/sensor-kit-v2-0-for-raspberry-pi-b-plus/lesson-32-mpu6050-gyro-acceleration-sensor-sensor-kit-v2-0-for-b-plus.html
3. http://pi4j.com/images/j8header-3b-large.png
4. https://www.imore.com/how-get-started-using-raspberry-pi
5. https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

Project Presentation Details

Part 1: https://www.youtube.com/watch?v=cZyUJT_VmFo&t=3s

Part 2: https://www.youtube.com/watch?v=ubVwlOaqJZo

Part 3: https://www.youtube.com/watch?v=cLcTbGTDRII