Development and implementation of an acceleration and GPS coordinate sensing box with automatic data transfer (2023)

Manoj Mohan, Nishanth Sundaran, Sushanth Mane, Seyedeh Farzaneh Alavi

***Abstract*— This case study presents a unique development and implementation of an acceleration and GPS sensing system focused on overspeed detection. The system effectively combines hardware components and data transfer technologies to accurately identify overspeed events, promptly alert relevant parties, and record crucial data for further analysis. The versatility of this solution extends its potential applications to various domains, particularly in transportation safety, where it facilitates data-driven decision-making and enhances overall safety measures.**

***Index Terms*— accelerometer, ADXL345 sensor, ThingSpeak, ThinkView, Raspberry Pi zero microcontroller, GSM/GPRS module.**

I. Introduction

THESE days, using sensors to measure and monitor various environmental factors has become inevitable. One of these efficient and common sensors is the acceleration sensor. Accelerometers are used in various fields to measure and control movement and force and provide information about acceleration, vibration, and speed changes of objects. For example, in the construction industry, it is used to evaluate the health of buildings by monitoring vibrations. In this way, abnormal vibrations are identified and measures are taken to prevent damage. This module is also used in vehicles and for travel safety. After detecting a sudden speed change or an immediate stop, car safety systems such as safety belts are activated. In the field of sports, measuring acceleration allows athletes to evaluate and improve their performance. The need to measure movement has also been seen in robotics and medicine. In robotics, it is used to control the movement of robots and their orientation, and in medicine, it is used to detect involuntary movements of patients or to detect them falling to the ground, and as a result, take timely action. These mentioned cases are a small example of the need for this project in different fields.

In this project, we are going to use the ADXL345 sensor. It can measure acceleration in three axes X, Y, and Z. The accelerometer box can be used to continuously measure acceleration and geographic location and is also possible to send SMS automatically using fast2sms in emergencies.

A. Methodology

Initially, it was started by collecting research papers regarding the acceleration sensor, GPS module Raspberry Pi and web interface and gathering information about the requirements for the project. In the meantime, while collecting the details, it started doing hardware design and concept as per the collected data. Once the design was completed, the coding was started as per the requirement.

B. Background

In this world, it requires the continuous monitoring of many technical parameters. The periodic mechanical vibration parameters are the most important mechanical parameters that are mainly used for the automobile industry, railway sectors, and even the power industry. Moreover, our project will provide a lot of solutions coupled with the vibration in Machinery and automobiles by calculating the acceleration. This paper it is proposed an accelerometer sensing box that can sense the acceleration of the moving object and update the continuous data into a webpage. The ADXL345 accelerometer is used as a sensor for measuring acceleration.

C. Purpose of study

Monitoring key components in mechanical or moving equipment is important to reduce the maintenance cost, breakage, and tearing of equipment and increase the lifespan. For example, nowadays vehicles with high acceleration are facing a lot of damage to engines and powertrains, suspension systems, and braking systems. Sometimes it can lead to human death. Through our literature research, it was understood that these types of problems can be minimized by continuously monitoring the acceleration and giving an alert to the user.

D. Scope of the project

By optimally calculating the acceleration of a moving object it can calculate the speed of the object at a particular time. In addition to that, many companies can use this data to avoid warranty claims and loss of parts and also can even reduce maintenance by improving the existing component using IOT technologies, In the future, we can control the speed and acceleration of many equipment like wind turbines, CNC machines, and in power distribution. We can also use this technology to switch ON or OFF equipment in the remote area.

II. LITERATURE REVIEW

A lot of work and effort has been put into this case study for developing a system that is capable of implementing an acceleration sensor box with automatic data transfer. Most of the research papers that were chosen for creating and developing the system were used for different applications. The first process includes designing hardware that can be capable of sensing the acceleration data and location. In the second approach, a web interface is created for accessing the accelerated data to the server. In addition to that it was added an alert message notification if the speed exceeds. Raspberry pi zero is a small and compactable low-board computer that is easy to integrate acceleration sensors into small devices. The processing time, GPIO pinas, and community support make the microcontroller excellent for acceleration sensing and web interfacing.

III. HARDWARE REQUIREMENT

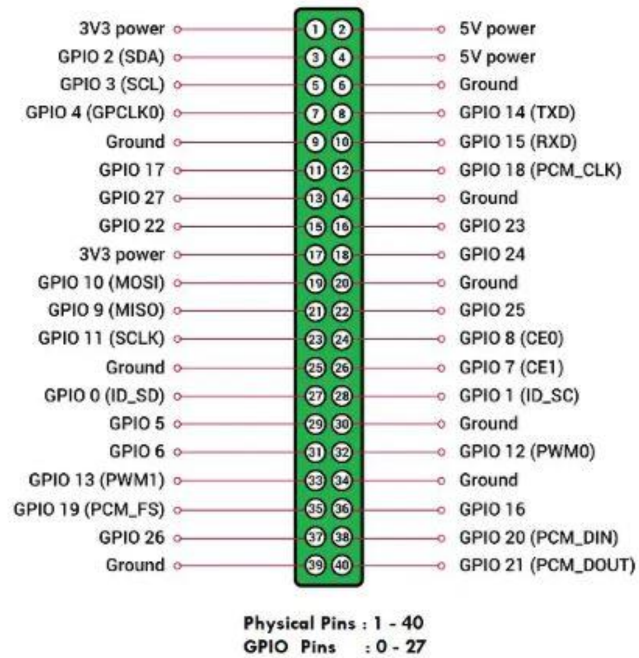
To build the accelerometer box, several components are used that play an essential role in the performance and implementation of this project, which is explained in this section.

A. Raspberry Pi Zero

Raspberry Pi Zero is a powerful single-board computer with high processing power that is used for data collection and processing in this project. Raspberry Pi's compact designs and small dimensions and affordable cost make it a suitable option for various projects. See Fig.1. This microcontroller can process heavy programs because of its features and also 40 GPIO pins on its board, making it suitable for connecting external devices such as sensors to it. These pins equip the micro with digital input/output, analog input, I2C, SPI, and other communication protocols which are shown in Fig.2.

Fig. 1. Raspberry pi zero wh.

Fig. 2. Pin details of Raspberry pi zero



The specifications of this micro are listed below [1]:

- Bundles or kits based on the Raspberry Pi Zero WH (pre-soldered with headers)
- Processor: BCM 2835 SOC, Single Core, 32-bit ARMv6Z
- Clock speed: 1GHz
- RAM: 512MB
- Built-in Wireless: BCM43143, WIFI + Bluetooth 4.1
- BLE (Bluetooth Low Energy), the same as Raspberry Pi 3.
- Memory: micro-SD
- Display and Audio: mini–HDMI USB Port: 1 x Micro-B USB for data (with power too)
- Power input: 1 x Micro-B USB for power (no data)
- Camera interface: CSI camera connector (needs adaptor cable, 0.5mm pitch to pitch CSI)
- GPIO: Pre-soldered 40-pin GPIO connector, compatible with Raspberry Pi Zero v1.3 and Raspberry Pi 3.
- Compatibility of GPIO: Compatible with existing HAT add-ons

To do different projects with the Raspberry Pi board, other tools offer other useful features. some of them are as bellow [3]:

- OTG cable: This cable is used as an interface between different devices and has many uses. In this way, it connects devices with a micro USB or Type C port to a device equipped with a USB port, and data transfer can be done easily. It is shown in Fig.3 You can use this cable to power your Raspberry Pi from a powerful bank or any other power source [3].

- PowerHub: The second most important component is the PowerHub. It is a device powered by a USB or external power source and produces four or more USB ports.
- GPIO header: GPIO module is beneficial because Raspberry Pi comes without soldered pins, it is useful to connect GPIO pins over and breadboard.
- MicroSD card and card adapter: A Microsoft card adapter may be required as not every computer has a MicroSD card slot for reading and writing data. It is cheap and saves you time.
- HDMI to mini-HDMI cable: Raspberry Pi Zero does not have a normal HDMI port. It is a bit smaller, and you need the HDMI to mini-HDMI cable to expand the port to a normal HDMI. then you are free to connect the port with any HDMI-compatible device.
- HDMI to VGA cable: In situations where you want to view something on the monitor and the monitor is not HDMI compatible, this cable allows you to connect the HDMI to mini-HDMI cable with an external monitor or TV.
- PCA Jack: With this module, you can connect the Raspberry Pi to the RCA jack on your Tv by connecting the two weirs of signal and the ground of the module.
- Raspberry Pi Zero case: The case will keep your Pi Zero safe and secure while also looking super sleek. And all ports and slots on the board are easily accessible.

In the Raspberry Pi board, small ports are used to save space, and there is also a row of 40 pins, which is considered one of the powerful features of this board because it provides great flexibility for use in different projects. Especially since by writing different programs we can see different functions of pins. Some of the popular GPIO pin functions are given in Table I [2].

The board consists of two 5V pins, two 3V3 pins, and 9 ground pins (0V), which are unconfigurable.

5V: The 5v pins directly deliver the 5v supply coming from the mains adaptor. This pin can use to power up the Raspberry Pi Zero, and it can also use to power up other 5v devices.

3.3V: The 3v pin is there to offer a stable 3.3v supply to power components and to test LEDs.

PWM (pulse-width modulation) Pins:

- Software PWM available on all pins
- Hardware PWM available on these pins: GPIO12, GPIO13, GPIO18, GPIO19

TABLE I
PIN DETAILS

Pin Type	GPIO pins	Description
PWM pins	GPIO12, GPIO13, GPIO18, GPIO19	For Pulse Width Modulation
SPI pins	**SPI0:** GPIO9 (MISO), GPIO10 (MOSI), GPIO11 (SCLK), GPIO8(CE0)/ GPIO7(CE1) **SPI1:** GPIO19 (MISO), GPIO20 (MOSI), GPIO21 (SCLK), GPIO18 (CE0)/ GPIO17(CE1)/GPIO17(CE2)	Used for Serial Peripheral Interface
I2C Pins	**Data:** (GPIO2), **Clock:** (GPIO3) **EEPROM Data:** (GPIO0), **EEPROM Clock**: (GPIO1)	Used for Inter-Integrated Circuit communication
UART Pins	**TX**: (GPIO14) **RX**: (GPIO15)	To communicate between two microcontrollers or computers.

SPI PINS on R-Pi Zero:

This is a communication protocol between the microcontroller and the electrical parts. It makes communication and data transfer very easy because no hardware parts are needed but it is not recommended in cases where we need high speed [2].

- **SPI0:** GPIO9 (MISO), GPIO10 (MOSI), GPIO11 (SCLK), GPIO8 (CE0), GPIO7 (CE1)
- **SPI1:** GPIO19 (MISO), GPIO20 (MOSI), GPIO21 (SCLK), GPIO18 (CE0), GPIO17 (CE1), GPIO16 (CE2)

I2C Pins on R-Pi Zero:

It is a protocol to communicate between microcontrollers and electrical components. This protocol includes two lines, which are used to synchronize signals and receive data for data transmission [2].

Data: (GPIO2), Clock (GPIO3)

- EEPROM Data: (GPIO0), EEPROM Clock (GPIO1)

UART Pins on R-Pi Zero:

Through UART (Universal Asynchronous Receiver / Transmitter) pins, two microcontrollers can be connected, and for this purpose TX and RX pins are used to transmit and receive serial data from different serial devices respectively.

- TX (GPIO14)
- RX (GPIO15)

Different Programming Guide for R-Pi Zero (Learning Material):

You can control the GPIO of Raspberry Pi Zero using many programming languages. Some of the popular languages along with learning material are given below:

- GPIO Programming using Python
- GPIO Programming using Scratch 1.4
- GPIO Programming using Scratch 2
- Programming GPIO with C/C++ using standard kernel interface via libgpiod
- Programming GPIO with C/C++ using 3rd party library pigpio
- GPIO Programming using Processing3

B. ADXL345 Sensor

ADXL345 sensor is a small, low-power, three-axis accelerometer that measures acceleration with high resolution. It is also capable of measuring acceleration due to gravity in tilt sensing applications as well as acceleration due to movement. Its high resolution allows it to detect the smallest changes in angle or movement.

The ADXL345 sensor has the following pinout as shown in Fig 3.
In Fig.4 a functional block diagram of the ADXL345 is shown.

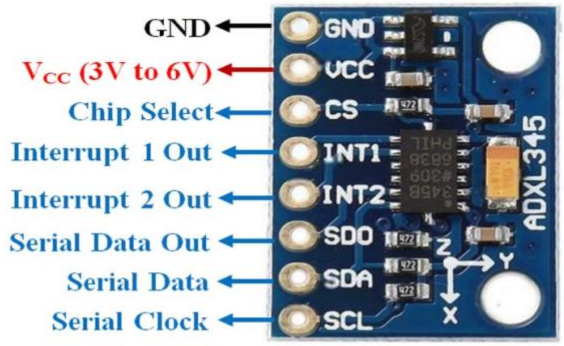


Fig. 3 ADXL pin details

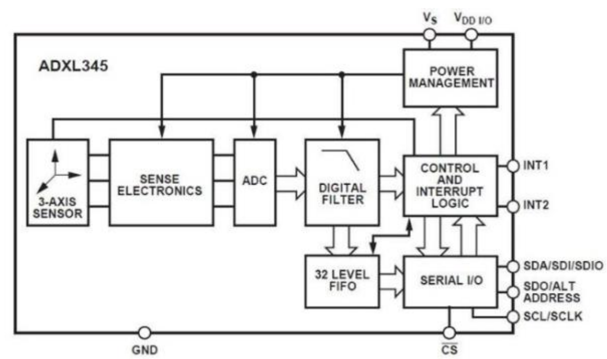


Fig. 4 ADXL block diagram

Some of the ADXL345 features are:

- This sensor is capable of measuring acceleration in three directions: X, Y, and Z. It has high accuracy and detects small changes with 13-bit accuracy.
- It has a wide measurement range.
- It has different power modes that allow the user to control power consumption by switching from full operation mode to standby mode.
- It can reduce unnecessary noises and thus obtain more accurate information on acceleration.
- various software libraries for this sensor allow the user to program easily.

Below you can see the specifications of this sensor:

- 3V-6V DC Supply Voltage
- On-board LDO Voltage regulator
- Built-in Voltage level convertor (MOSFET based)
- It can be interfaced with a 3.3V or 5V Microcontroller.
- Ultra-Low Power: 40uA in measurement mode, 0.1uA in standby@ 2.5V

- Tap/Double Tap Detection
- Free-Fall Detection
- SPI and I2C interfaces
- Measuring Range: ±16g
- Measuring Values (-16g to +16g):
- X: -235 to +270
- Y: -240 to +260
- Z: -240 to +270

C. Raspberry Pi GSM/GPRS/GPS Module

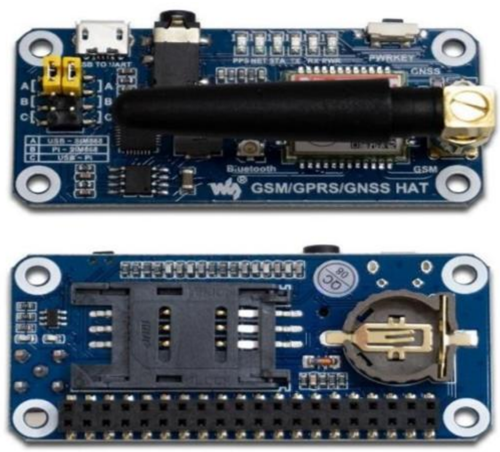


Fig. 5. GSM/GPRS/GPS module.

GSM/GPRS module is a communication module that is mostly used in applications that require positioning. This module also allows your Pi to easily make a telephone call, send messages, connect to wireless Internet, global position, transfer data via Bluetooth, and so on. Then by adding its GPS antenna, precise navigation can be done. It has many applications in the fields of tracking, monitoring, navigation, and IoT [6]. See Fi.6.

The specifications of this module are as follows:

- Raspberry Pi connectivity, compatible with Raspberry Pi 2B/3B/3B+/Zero/Zero W
- Supports SMS, phone calls, GPRS, DTMF, HTTP, FTP, MMS, email, etc.
- Support GPS, COMPASS, LBS base station positioning, omni-positioning
- Bluetooth 3.0, supports data transferring through Bluetooth
- Onboard USB TO UART converter CP2102 for UART debugging
- 6x LEDs for indicating the module’s working status
- SIM card slot for 1.8V/3V SIM card
- RTC with backup battery holder
- Baud rate auto-detection (1200bps ~115200 bps)
- Control via AT commands (3GPP TS 27.007, 27.005, and SIMCOM-enhanced AT Commands)
- Supports SIM application toolkit: GSM 11.14 Release 99
- Comes with development resources and manual (examples for Raspberry Pi/Arduino/STM32)

IV. Hardware configuration

Several important components make up the hardware setup for implementing the acceleration sensing box with automated data transfer. The Raspberry Pi Zero microcontroller acts as the project's central processing unit. The ADXL345 sensor is connected to the Raspberry Pi Zero and is an accelerometer capable of detecting acceleration in three axes: x, y, and z. This sensor collects acceleration data from its surroundings. A GSM/GPRS module facilitates communication and data transmission by connecting to the server over a network. ThingSpeak, a web-based platform that functions as a centralized storage and processing hub for the collected data, is the server utilized in this project.[7]

A. System Overview

The system overview in Figure 7 depicts the relationship of each component to the microcontroller. The prototype development process is divided into two stages, the first of which is the selection and construction of hardware. A sensor, microprocessor, and GSM/GPRS modules make up the hardware. The second section is about software development; software is the coding that is programmed into the microcontroller for the control part. A microcontroller is an integrated circuit that has a processor, RAM, ROM, and input/output (I/O) peripherals. The Raspberry Pi Zero is popular because it is open-source and offers free material that can be downloaded from the internet. It was chosen for this project because it is perfect for a vehicle security system with easy hidden installation.

The Raspberry Pi zero microcontroller runs at a clock speed of 7000 MHz. This clock speed indicates the number of cycles the processor can execute in one second and provides 40 digital input/output (I/O) ports. It provides the serial peripheral interface (SPI) and I2C protocol for connecting sensors. In this project, that is shown in Figure 7 Raspberry Pi zero is used to process sensor reading from the ADXL345 accelerometer, Raspberry Pi GSM/GPRS Module, and GSM modem by python programming using Raspberry Pi Integrated Development Environment (IDE) software. The ADXL345 accelerometer sensor comes with a small and thin integrated circuit and the module used in this case study. This module features dual voltage supply options of 5 V and 3.3 V for user convenience.

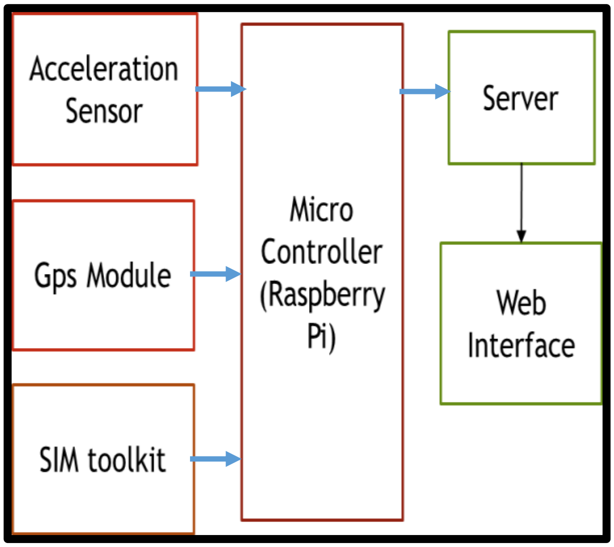


Fig. 6. overview.

ADXL345 accelerometer sensor can measure acceleration in the X, Y, and Z-axis with a high resolution of 13-bits and user configurable-measurement range of ± 2g, ± 4g, ± 8g, and ± 16g, which is more than enough for this project. Besides, the ADXL345 also features SPI (3 or 4 wire) and I2C digital interface for interfacing with microcontrollers. The ADXL345 provides the measurement of tilt angle by measuring the static acceleration of gravity and dynamic acceleration in the 3 axes, which is suitable for this project to detect any vehicle movement in all axes and transmit it to the microcontroller. As part of the hardware configuration for the accelerometer box project, a Raspberry Pi Zero microcontroller, an ADXL345 accelerometer sensor, and a GSM/GPRS module will be used. Providing computational capabilities for the system is the Raspberry Pi Zero, which acts as the central processing unit. This accelerometer sensor is capable of measuring acceleration in three axes, so that movement and position can be accurately tracked. As a result of the GSM/GPRS module enabling communication with the mobile network and Fast2SMS, SMS alerts can be sent in emergencies in response to the emergency. The internet connection, established either via Wi-Fi or Ethernet, is an essential part of transmitting acceleration information and position information to a server through a web interface. As well as this, a reliable power supply is essential to ensure that the Raspberry Pi Zero and other components are powered on continuously. By exporting the collected data through the ThingSpeak web platform, and by using the ThinksView mobile app for users to access, analyze, and visualize the collected data, the system enables effective monitoring of environmental conditions and informed decision-making based on the data gathered.

B. Acceleration Measurement

1) **Method**
Fig8 shows the flowchart of the acceleration measurement part when the alarm is turned ON by the user, the sensor will be initialized with the first measurement reading as a reference point. After initializing, the sensor will measure the dynamic acceleration continuously to detect changes in acceleration in X, Y, and Z-axis. If a measured acceleration value is higher than the threshold value, the siren alarm is triggered and followed by the activation of the engine ignition control relay, which will turn off the power of the ignition system. SMS will be sent to alert the user. The output of the ADXL345 sensor through the I2C protocol needs to be filtered by taking average values to minimize the effect of noise. This is achieved by taking an average of five readings continuously and calculating the average value in the 3-axis. The value is then converted to unit G based on the formula stated in the ADXL345 datasheet as shown below. Accelerometer calibration and offset values:

2) **Accelerometer sensor calibration and offset values**:
The accelerometer or the accelerometer sensor is attached to a mechanical structure that can be freely moved in any direction. These elements are more sensitive to mechanical stresses like shock or shaking than electronics. The offset value is more important to the acceleration because it is significant to measure the accurate acceleration. The most common and accurate calibration method is to use two points per axis. Since the ADXL345 has a three- axis, it must be measured in six points. By calculating the maximum and minimum readings of the sensor, the offset value can be calculated. The new values of offset and gain will give you the calibrated acceleration value. The accelerometer should be mounted with the z-axis parallel upward before taking the measurement. Each time to measure a different axis, the block on which the ADXL345 breakout is mounted, is turned. An X-Y-Z axis symbol on the breakout board helps with orienting in each direction.[7]

3) **Flowchart**
The figure 8 initializing the sensor to read acceleration data is the first step in the procedure. The system then uses the ADXL345 sensor to detect the acceleration before updating the information to the online platform ThingSpeak for data storage and presentation. The observed acceleration value is then checked to see if it exceeds a predetermined

threshold. In the absence of such, the system keeps recording acceleration. A warning message is, however, delivered to the customer through email or the Fast2SMS service if the threshold is surpassed. The flowchart ends at this point, signaling the end of the system's execution. This condensed explanation emphasizes the crucial elements in the workflow of the system and illustrates the sequential flow from sensor initialization through warning messages based on acceleration values.

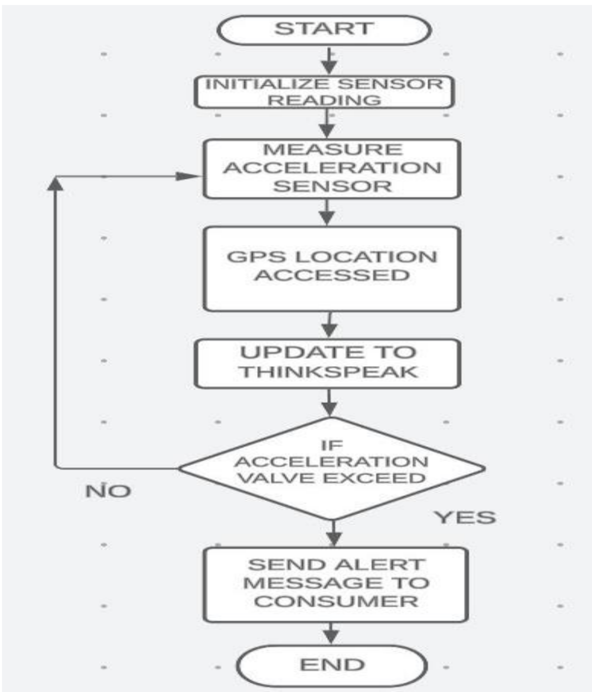


Fig. 7. System flowchart.

The following steps make up the operation of an accelerometer box:

a) **START**: The flowchart begins with the start point, indicating the initiation of the system's operation.
b) **INITIALIZE SENSOR READING:** This step involves initializing the ADXL345 sensor to prepare for acceleration measurement. It ensures that the sensor is ready to detect and measure acceleration.
c) **MEASURE ACCELERATION SENSOR:** The system proceeds to measure the acceleration using the ADXL345 sensor. This step captures the current acceleration values from the sensor
d) **GPS LOCATION ACCESSED:** Updated GPS position data from the ThingSpeak platform on a regular basis, ensuring that the displayed location is always up to date.
e) **UPDATE TO THINKSPEAK:** After measuring the acceleration, the system updates the acquired acceleration data to ThingSpeak, a web platform for storing and visualizing data. This step allows for remote access and monitoring of the acceleration data.
f) **IF ACCELERATION VALUE EXCEED:** In this conditional statement, the system checks whether the measured acceleration value exceeds a predefined threshold or limit. If the acceleration value does not exceed the threshold, the system returns to the "MEASURE ACCELERATION SENSOR" step to continue monitoring.
g) **SEND ALERT MESSAGE TO CONSUMER (E-MAIL, FAST2SMS):** If the acceleration value exceeds the specified threshold, the system proceeds to send an alert message to the consumer. The alert can be sent through different communication channels, such as email or the Fast2SMS service. This step aims to notify the consumer of an emergency or critical situation based on the detected acceleration.
h) **END:** The flowchart terminates here, signifying that the system has completed its operation.

V. DESIGN

A. Why Raspberry Pi is used over Arduino?

Single-board computers or well known as microcontrollers have a wide range of applications in day-to-day life. It gives a vast opportunity in the development and learning process for a beginner. This single circuit board consists of a microprocessor, memory, and input or output pins which is similar to a computer in small size Both Arduino and Raspberry Pi have different characteristics and capabilities. A comparison of these two is provided in Table II.

The following features give more advantages to this project for Raspberry Pi over Arduino:

- **Learning curve:** The Pi is much more similar to a mini computer and provides more doorways in the world of programming. This processor can be programmed using Python which is one of the simplest programming languages in today's world. In addition to that, the Linux computation can be used on any Linux machine, once it's installed.

	Arduino	Raspberry pi zero
Price	5 $	8
Size	7.6x1.9x6.4	8.6x5.4x1.7
Memory	0.002 MB	512MB
Clock speed	16 MHz	700MHz
On Board Network	None	10/100 wired Ethernet RJ45
Multitasking	No	Yes
Input Voltage	7 to 12V	5V
Flash	32KB	SD card
USB	One, Input only	Two, peripherals OK
Operating System	None	Linux distributions
Integrated Development Environment	Arduino	Scratch, IDLE
I/O pins	28	40
Family	AVR	ARM
Languages used	C/C++	PYTHON; HTML; JAVA SCRIPT

TABLE II
COMPARISON OVER ARDUINO

- **Simplicity:** The Pi provides a user-friendly interface for IOT-based projects when compared with the other processor. Also, it requires simple languages to transfer data to a server when compared to Arduino.
- **Available programming languages**: The Pi adopts several programming languages like Scratch, Python, HTML 5, JavaScript, JQuery, Java, C, C++, Perl, and Erlang which gives a beginner to design their idea better.
- **Network capabilities**: PI has more network capabilities when compared to Arduino. It has 3 Bluetooth and wireless connection capabilities. On the other hand, it can be connected to the internet by using Ethernet. It has 1 HDMI port, 4 USB ports, one camera port, 1 Micro USB port, 1 LCD port, and 1 DSI display port, which makes it compatible with various applications.
- **Process speed:** The process speed difference between Arduino and Raspberry Pi is vast. Because Pi is similar to a computer and Arduino is just simply a processor. The comparison of clock speed between Arduino and Pi has the values 16MHz and 700 MHz respectively. This gives a clear image that Raspberry Pi is 40 times faster than Arduino. Moreover, the Pi board has 128000 times more RAM when compared to Arduino.

B. Web Interface

ThingSpeak is an open-source Internet of Things (IoT) platform developed by one of the software companies which is mainly concentrating on the mathematical computation of Real-time data and for collecting, processing, and visualizing sensor data from various IoT devices. It works as a communication platform between the physical and digital worlds, which helps users to collect or monitor real-time data from sensors and actuators and analyze them for meaningful insights. It also offers many features and functionalities that can be useful for the users to apply in various sectors and applications. The most common key aspects of ThingSpeak are:

- **Data Collection**: ThingSpeak allows the user to collect and monitor data from sensors by using HTTP or MQTT APIs integrate with a wide range of devices and protocols. Using this technique, users can send data to ThingSpeak in real-time or even in batch mode. Also, it provided flexible data acquisition.
- **Data Processing and Analysis**: Once the data is fed or collected, ThingSpeak has a built-in feature that is capable of supporting the data process and analysis. Also, with the help of MATLAB can perform complex mathematical computations, statistical analysis, or machine learning algorithms.
- **Visualization and Dashboarding**: ThingSpeak provides a customizable dashboard that can be used by the users to create a visualization of the data. These IOT data can be used to make charts, graphs, and maps to present it in a meaningful way and make it easy to monitor.
- **Integration with External Services**
It offers a wide range of communication or integration between social networks like Twitter, YouTube, MATLAB, and other external services.
- **Open-Source and Community Support:** ThingSpeak is an open source, the user can access the code and modify it as per the suitable requirements. This platform is developed by an active community of users and developers and provides support and shares their projects and ideas.

C. File Formats for Graphics

ThingView is an IOT-based dashboard and a visualization tool that helps to access, monitor and control the IOT device from the updated data in the ThingSpeak. The main key features of ThingsView are as follows:

- **Real-time Monitoring:** It displays the updated or monitored data in the form of graphs, charts, and gauges. This will help the user to monitor or control various parameters and devices.
- **Remote Control**: The ThingsView dashboard offers remote control access. This feature helps users to communicate remotely with the IOT devices and can execute various tasks like ON/OFF devices, and changing the setting with the help of the internet.
- **Alerts and Notifications:** By creating threshold values or conditions, it can send alert messages and Notifications to the users. This feature will help to monitor critical situations.
- **User access control**

ThingSpeak provides a user interface that can be accessed and monitored by a group of p people. This will help multiple users to do their roles in a project.

D. HAT

The main advantages for using HAT in this project are as follows.

- **Purpose**: It is developed for the processors to achieve certain requirements and to increase their abilities and performance. By using a HAT, the user can access GPS, motor control, display outputs and can be used as sensor interface and some other wireless communication.
- **Compatibility:** These are compatible with all microprocessors and some microcontrollers. it can be connected directly to the input and output pins and can access data to and from the controller pins.
- **Physical form:** The shape of the pins in the HAT is just similar to the controller pin. So it can be easily plugged on to the micro controller. It is pretty much secure and aligned properly for easy communication.
- **Electrical communication**: Hats which we used to expand the microcontroller are capable of transferring data and power.

Moreover, the hats offer community support, Stack capability, Software integration and it also provides the user to expand the project structure and expandability

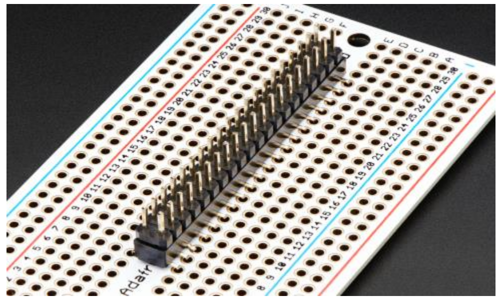


Fig. 7. HAT.

E. Advantages of using HAT as a connector between Raspberry pi and GPS in this project

The GPS is connected directly to the Raspberry Pi for simplicity and efficiency. Sometimes as per the requirement of the project and the structure of the project, it can also be connected through a breadboard. Preliminarily, in this project the GPS was connected to a breadboard and found many problems in delaying data transfer and performance.

By using a HAT to communicate with GPS, it offers the following features.

- **Easy and Compact connectivity**: It offers easy and compact connectivity by using HAT as the expandable module or add-on module to avoid the messy wiring and also improve the accuracy in the correction.
- **GPIOS safety guar**d: The electrical surge happened because of the spike in input voltage or due to other issues to Pi can be reduced and it act as an electrical protector such as voltage level shifters and protectors. Thereby, the GPIO pins are protected to a knowledge.
- **Plug and play**: The HAT configuration is simple and cost efficient. The system can simply understand the connection instantly and offers the drives and configurations instantly.
- **Reliability:** HATS are really tested and designed and approved by the Community. IT ensures reliability and Compatibility. It provides the chances of hardware conflict and improves the integration with Pi.
- **Performance:** IT offers a better performance than any other connections. The manufactures designed the HAT with high signal integrity, best performance and with minimal interference to the data transfer and signal exchange.
- **Community support:** The HATS have a strong community and they offer all the documentation, guidance and tutorial for the new projects. It helps the user to know about the severe connections in a large project and the usage of HAT.

- **Space limiter:** The HAT acts as a space limiter and aesthetics. This will help the user to reduce the size of the project and understand quickly without any messy wires and inaccurate connections.
- **Expansion:** Further expansion of the project is also possible using a HAT. It helps in easy expansion and additional devices.

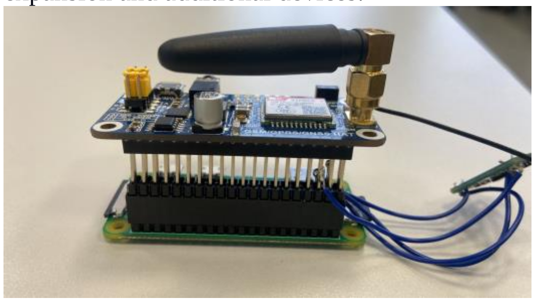


Fig. 8. Hat connection.

F. Why is ADXL345 used in this project?

Having small, thin dimensions, low power requirements, and voltage outputs that are signal-conditioned, the ADXL335 is a full 3-axis accelerometer. In order to measure acceleration, the system requires a full-scale range of at least 3 g. It can evaluate both the gravitational acceleration that is static and the dynamic acceleration that is brought on by motion, shock, or vibration in tilt sensing applications.

Across a range of professions and industries, the ADXL345 is becoming more common and well-liked. You can rely on its dependability and accuracy because a sizable user base has thoroughly examined and verified its performance. The communication protocol options available with the ADXL345 are flexible. You can select the interface based on your project's needs, your microcontroller's or development board's capabilities, and the fact that it supports both I2C and SPI. This compatibility facilitates the sensor's integration into current systems and ensures platform diversity.

The ADXL345 is an excellent option for applications that require power optimization because it is made to operate well in low-power modes. Thanks to the device's assortment of power-saving features, which include selectable measurement ranges and variable data output rates, you can modify the power consumption to suit your unique needs. This is particularly advantageous for projects requiring high levels of energy efficiency, wearable technology, or battery-powered applications. The ADXL345 offers three-axis acceleration sensing, allowing acceleration in the X, Y, and Z directions to be measured. Additionally, you can choose the sensitivity that best suits your application from a variety of selectable measurement ranges. Depending on the needs of your project, you can use this flexibility to measure a broad range of accelerations, from small movements to strong forces. The ADXL345 is offered in SMT (surface-mount technology) packages, which are small. It is appropriate for projects requiring little space and/or low weight or size due to its small size. Physically mounting the sensor onto your circuit board or device is also made simpler by the integrated package. The ADXL345 device operates without issue in challenging conditions or under mechanical strain. Its dependability and durability are ensured by built-in security measures. The sensor's durability qualifies it for industrial applications as well as other tasks that call for durability and robustness.

The ADXL345 enjoys the support of a strong user and developer community because it is a well-liked sensor. This suggests that there is a wealth of information available, including manuals, code examples, and libraries that show how to use the sensor effectively. When resolving problems, learning about challenging features, or seeking guidance from knowledgeable users, community support can be incredibly helpful. The prices of the ADXL345 are frequently comparable to those of other high-performance accelerometer sensors due to its popularity and wide availability. Due to its accessibility, it is a desirable choice for projects requiring a small budget or where cost containment is a top priority.

G. 3D Design

3D modeling is called simulating or designing a model, space, image, etc. in 3D space. This operation is done with the help of computer graphics software. 3D modeling and rendering also allow you to virtually inspect and examine a project from different angles before it is built and produced. This feature helps a team to show their ideas to the employer in the best way and inform him of their goals. 3D modeling allows engineers to visualize their machines and processes in greater detail. If a machine has hundreds of small parts, it can examine each 3D part individually. One of its applications is in the electronics industry and for making enclosures for electronic components. The delicacy of sensitive parts and wiring in electrical systems makes the need for a case for the system vital, as a result of which the performance of the system is not affected by external factors, and in addition to the significant increase in appearance, the performance of the device is also more reliable and resistant. It also increases the system against unwanted impacts. This technology allows making boxes and bags suitable for the type of product. One of the 3D design software is AutoCAD. This software uses computer processing capabilities to draw, edit, analyze, and optimize various designs with greater speed and efficiency. Using AutoCAD software and considering the hardware design of the project and how to connect the modules, a box has been designed which is shown in Fig.10.

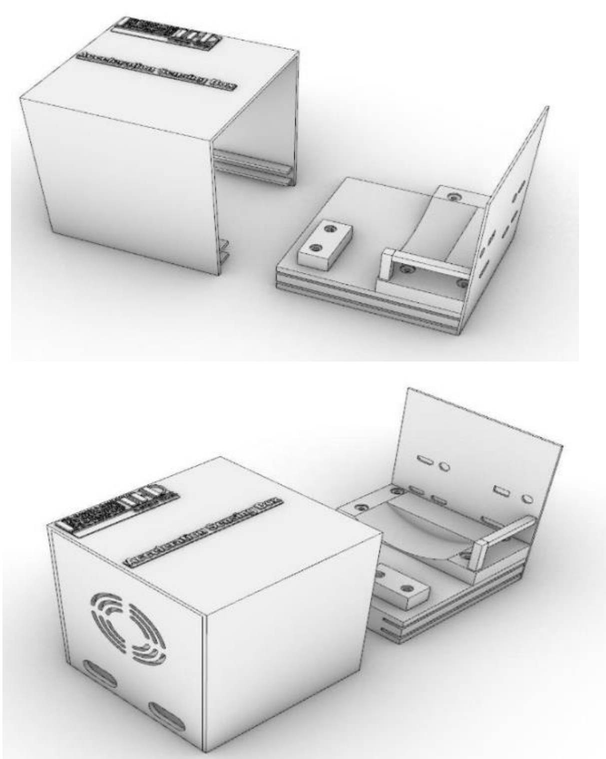


Fig. 9. CAD design

H. Software Design

We are using python programming language for writing the script and it is handy for reverse engineering. The software design of our case study plays a critical part in processing the data from our ADXL345 sensor. The data from the ADXL345 sensor plays a pillar role where we set a threshold for the acceleration data, with respect to the threshold, we are facilitating communication alert messages via SMS/E-mail. The alert message contains GPS data, acceleration data, time of the incident etc.

The first phase of the code is designed to begin with the acquisition of data from our ADXL345 sensor. The communication of ADXL345 and raspberry pi is through the appropriate communication protocol, that is I2C & SPI. So, we initialized the I2C bus in our code. It also involves the initialization of the GNSS module for serial communication, threshold values, configuring the Email credentials, Thingspeak API key, and the gravitational acceleration. We expected the gravitational acceleration on the Z-axis and we were required to calibrate.

The Email and SMS is planned to send through API requests. For optimization purposes, SMTP connection is made ready on a priority basis.

Once it is done, we must extract values from our ADXL345 sensor, where the sensor will give acceleration measurement values along the X, Y, Z axis. For calculating the magnitude of the acceleration value, we need to know how the sensor works, the law behind it and what exact value sensor we get. ADXL345 sensor works based on Newton's law of motion, particularly the second law, states that the force acting on an object is equal to its mass multiplied by its acceleration (F=ma). ADXL345 measures the values along three axes using a microelectromechanical system (MEMS) structure. It detects the changes in capacitance caused by acceleration and this value is converted to digital acceleration values. We use this formula for calculating the magnitude from the acceleration vector.

Magnitude = sqrt(X^2 + Y^2 + Z^2)

This combines the individual acceleration value along each axis into a single value by using Pythagoras theorem.

We are subtracting the gravitation component that is 9.8(m/s^2) to get the net acceleration. The net acceleration is used for all further condition statements in the code.

The alert message condition is carried depending on a threshold value which is kept on 10 (m/s^2). Once the acceleration value exceeds the threshold value, a condition is written to send alert messages over SMS and emails.

Parallelly, Sensor data is transferred to the web interface which we are using Thingspeak in our case. This transfer is also done over API requests.

Once the over speed condition is triggered, the alert message must contain location data, for that we are collecting location information from our GNSS which comes with our module. We get latitude and longitude values from the GNSS module, but the end user can't easily access locations with latitude and longitude values. so we are converting it to google map link where it is more convenient to access the location.

Google map url,

https://maps.google.com/?q={latitude},{longitude}

looks like this and we are writing our code to replace it with the actual coordinates.

The alert message includes, date and time, Gmap link, magnitude of the acceleration, GPS coordinates and a slowdown message.

Acceleration sensor value is sent to the web interface thingspeak where we can visualize the data, store data and if we need CSV data, we can download it from thingspeak.

VI. Development

A. Prerequisites

1) **Raspberry Pi setup**

We can call Raspberrry pi zero board as low powered mini-single board computer, where it needs a operating system to work. The popular operating system it uses is Raspbian OS which is based on

Linux. For installation we need a MicroSD card (at least 8GB) with a microSD card reader. We can use Raspberry pi OS imager application to download and install raspbian OS to our MicroSD card. Raspberry pi Imager can be downloaded from raspberry pi official page or with a Linux based terminal, use command.

sudo apt-install rpi-imager

While installing, setup WIFI SSID and password. Also, we made sure SSH connection is enabled before flashing the OS into the SD card. SSH connection helps to access the terminal of Raspberry pi to enable VNC viewer. We can access the display of raspberry pi from our desktop with the VNC app.

To access the terminal of Raspberry pi, we made a SSH connection through our desktop terminal. Make sure the raspberry pi and our desktop are connected to the same WIFI network. We have to use the IP address of raspberry pi and the password which is given at the time of OS installation. After successful SSH connection, we can run the following command to open the configuration utility,

sudo raspi-config

using the arrow keys, we can navigate into 'interfacing options. Inside interfacing options, scroll down and select VNC (Virtual Network Computing). Choose 'yes' to enable VNC.

We can use the IP address of raspberry pi to connect through VNC viewer. Once we are connected, we will get the graphical user interface of raspberry pi into our desktops display. [12] [14]

2) **Enable I2C protocol and SPI serial communication**

Enable I2C (inter-integrated communication) and serial communication on raspberry pi from the 'interfacing options' itself. Turning on I2C will help to communicate between a master device and a slave device. Here it is raspberry pi and ADXL345 sensor. Turning on serial communication will help us to communicate with the GPS module which is connected over the hat with raspberry pi.

3) **Installation of required libraries**

Adafruit ADXL345 -We are using ADXL345 acceleration sensor, so for the communication between raspberry pi and sensor in python environment, we need python libraries to do the task. Adafruit-adxl345 library helps to do that. For installation we can run the following command in the terminal of raspberry pi using pip.

pip install adafruit-adxl345

Serial - Serial library is a popular library in python which will provide support to communicate through serial communication, allowing us to interact with devices connected through serial ports. we can install the python library using

pip install pyserial

Requests - Requests library helps to enable HTTP requests and communication in our python code. We are using this library for Email, SMS, and web interface. The following command will install the requests library.

pip install requests

All other libraries are standard libraries, so no need to install it.

4) **Setup of mails via SMTP**

We created a personal Gmail account (accelerationsensorthdcham@gmail.com) and used it for sending email alerts. To use personal Gmail for raspberry pi or any other dev boards or devices, as dev boards or devices are considered as insecure or less secure devices, we need to generate app passwords for sign in with google. For generating app passwords, we need to turn on Two-step verification on our Gmail account and follow the below steps which we followed as in google instructions.

- go to your google account
- select security
- under 'signing in to google', select 2-step verification.
- at the bottom of the page, select App password.
- select Generate
- We will get a 16-character code that can be used as our app password and select done.

We can use this for signing purposes in our python code using SMTP (Simple Mail Transfer Protocol). [13]

5) **Creation of account in Fast2sms**

As mentioned in the software design part, we are sending SMS via SMS gateway platform fast2sms though API. Once we create an account in it, we will get the API keys for accessing the account from our raspberry pi board. The documentation for sending SMS through API is available in the platform itself. We used documentation for reference purposes. [11]

6) **Creation of account in thingspeak**

Thingspeak is an IOT platform that enables us to collect, analyze and visualize the data from connected devices. It provides an API for sending and retrieving data and provides a visualization. Once we create an account, we will get the Write API key. We also created a channel and named it 'acceleration sensor value'. The channel ID can be shared to anyone for visualization purposes.

B. Development of script

The development of code is done in such a way that we wrote individual code for acceleration sensor, gps, for sending Email and sms. We tested and validated each code once it is written and moved to other components, alerts and web interfaces.

In beginning of the script, we imported all the necessary modules, these include time, board, busio, adafruit_adxl34x, smtplib, requests, datetime, serial. The time library allows for time-related operations, while board and busio handle the I2C bus communication. The adafruit_adxl34x library provides support for the ADXL345 accelerometer sensor. Additionally, the smtplib library enables email sending, requests library facilitates HTTP requests, datetime handles timestamp operations, and serial supports communication with the GPS module.

The I2C bus is initialized using the busio library, specifying the SCL and SDA pins from the board library. This initialization allows the script to establish communication with the ADXL345 accelerometer sensor effectively. The code then configures the serial port settings for the GPS module using the serial library. The specified serial port and baud rate establish the communication channel for retrieving GPS data.

Email and SMS Configuration: The script proceeds to set up the configuration details for sending email alerts. This includes specifying the SMTP server, SMTP port, email address from which the alerts will be sent, password for authentication, and the recipient's email address. These details will be used later in the script to send email alerts when overspeed conditions are detected.

Additionally, the script configures the Fast2SMS API by providing the API key and the recipient's phone number. This configuration enables the script to send SMS alerts using the Fast2SMS service.

Thresholds and API Key Configuration: To detect overspeed conditions accurately, the script defines the overspeed threshold. This threshold represents the point at which an acceleration value is considered as an overspeed event. In this code, the threshold is set to 5.0 m/s^2, but it can be adjusted according to specific requirements.

We specify the ThingSpeak API key for sending the acceleration data to the ThingSpeak platform. ThingSpeak is an IoT platform that allows for data storage, visualization, and analysis. With the provided API key, the script can seamlessly integrate with ThingSpeak and transmit the recorded acceleration data.

Furthermore, the script defines the value of gravitational acceleration, which is subtracted from the magnitude of acceleration to obtain the net acceleration. By excluding the gravitational component, the net acceleration provides a more accurate representation of the acceleration caused by movement or external forces.

Alerting Function and Integration Setup: we included the send_alert() function, which is responsible for sending email and SMS alerts when overspeed conditions are detected. This function takes a subject and message as input parameters. Inside the function, it constructs an email message using the MIMEMultipart and MIMEText classes from the email.mime.text and email.mime.multipart modules, respectively.

To send the email, the function establishes a connection with the SMTP server, starts TLS encryption, and logs in using the provided email credentials. Then sends the email to the specified recipient using the sendmail() function. If the email sending process is successful, the function prints a confirmation message. However, if any errors occur during the email sending process, the function catches the exceptions and prints an error message, ensuring robustness and error handling.

For SMS alerts, the function constructs the necessary parameters, including the API key, message, language, route, and recipient's phone number. It then sends an HTTP GET request to the Fast2SMS API endpoint to trigger the SMS alert. The response is evaluated, and appropriate messages are printed based on the success or failure of the SMS sending process.

The integration with ThingSpeak is implemented by constructing the URL for sending the acceleration data using the ThingSpeak API key. An HTTP GET request is sent to the specified URL using the requests library. This enables the script to send the magnitude of acceleration to ThingSpeak for recording and visualization.

Main Loop Execution: The script enters an infinite loop, which represents the main execution flow of the system. Inside the loop, the code reads acceleration values from the accelerometer sensor for the X, Y, and Z axes using the acceleration attribute of the accelerometer object. These values represent the raw acceleration values in m/s^2. The code then calculates the magnitude of the acceleration by squaring each axis's value, summing them, and taking the square root. This provides the overall magnitude of the acceleration vector.

To obtain the net acceleration, the code subtracts the value of gravitational acceleration (9.8 m/s^2) from the calculated magnitude. This adjustment effectively removes the gravitational component from the readings, allowing for the detection of acceleration caused by movement or external forces.

Then we check if the net acceleration exceeds the overspeed threshold defined earlier. If the net acceleration value is higher than the threshold, it indicates the occurrence of an overspeed event. When an overspeed condition is detected, the script proceeds to read GPS data from the GPS module. It uses the readline() method of the gps_serial object to read a line of GPS data, decodes it into a string, and splits it into individual values. The assumption is made that the GPS data is in the format "latitude,longitude".The latitude and longitude values are then extracted from the GPS data as floats. Using these coordinates, a Google Maps link is constructed, providing the exact location where the overspeed event occurred. This link can be included in the email and SMS alerts, enabling the recipient to visualize the location on a map.

After retrieving the GPS data, the send_alert() function is called, providing the subject and message for the email and SMS alerts. The message includes information about the overspeed event, such as the magnitude of acceleration, GPS coordinates, and the Google Maps link. This ensures that the recipient receives detailed and actionable information in the alerts, facilitating prompt action and response.

Furthermore, the script sends the magnitude of acceleration to the ThingSpeak platform using an HTTP GET request. The script constructs a URL with the ThingSpeak API key and the acceleration value. This URL is used to send the data to the ThingSpeak API endpoint. The response from the request is captured and printed to the console, allowing for the monitoring of the data transmission to ThingSpeak.

Finally, to avoid excessive resource consumption and allow for proper timing, a short delay of 0.1 seconds is included using the time.sleep() function before starting the next iteration of the loop. This delay ensures that the loop does not execute too rapidly, providing a brief pause between consecutive readings and reducing the computational load on the system.

VII. Testing and Validation

A. Testing

The testing of the system is carried out in a closed environment where we created a scenario of acceleration of 5m/s^2. For testing purposes, we reduced the threshold value to 5m/s^2, So when it exceeds, the system will give an alert message. Once we tested, we were successfully able to give alert messages over SMS and EMAIL. The alert message includes, date and time, an overspeed message, magnitude of the acceleration, GPS coordinates, and a google map link.

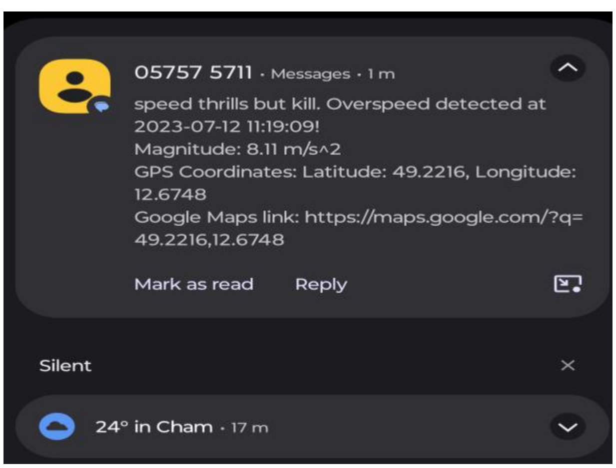


Fig. 10. SMS alert

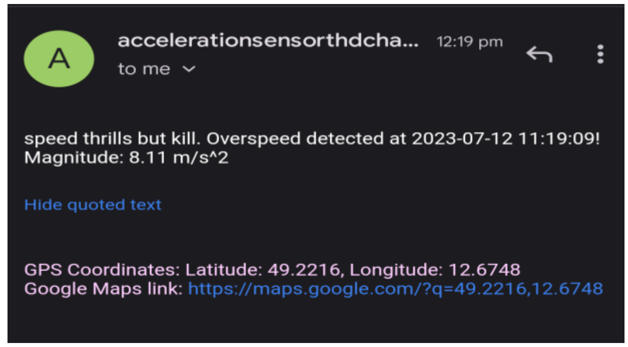


Fig. 11. Email alert

Since the starting stage of the system, it is continuously sending acceleration data to thingspeak.

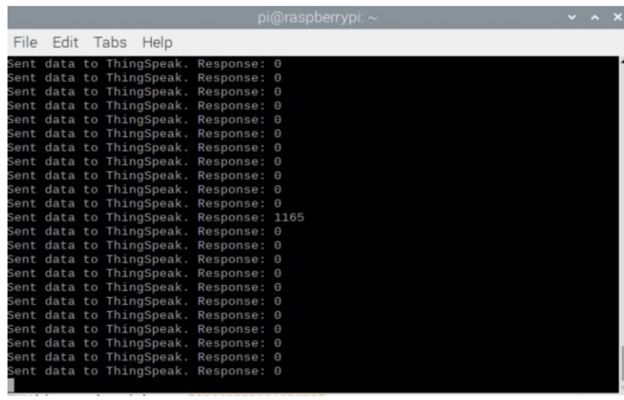


Fig. 12. Thingspeak update

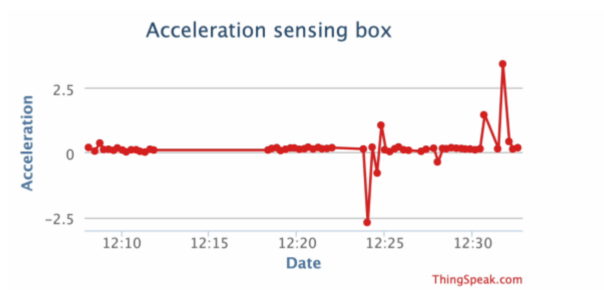


Fig. 13. This is a sample of a figure caption

The fig shows thingspeak is receiving acceleration data successfully and able to plot a graph.

Further we had gone through boundary and stress tests.

1) **Boundary test**

Overspeed Threshold: We tested the system with different overspeed thresholds. We set a very low threshold of 1.0 m/s^2 to verify that the system does not generate false positives. The system performed accurately, only triggering alerts when the acceleration exceeded the defined threshold. We also tested with a high threshold of 20.0 m/s^2 to ensure that the system can accurately detect extremely high acceleration values. Again, the system responded correctly, generating alerts when the acceleration surpassed the threshold.

Acceleration Range: To assess the system's behavior at the edges of the acceleration range, we tested with values close to zero and the sensor's maximum limit. The system handled both cases effectively, detecting overspeed events accurately. It did not generate false positives for low acceleration values and triggered alerts when the acceleration approached the maximum limit.

GPS Coordinates: We tested the system with minimum and maximum latitude and longitude values to verify its behavior with boundary coordinates. The system correctly processed and displayed these extreme values in the overspeed alerts. We also tested with invalid or out-of-range coordinates to ensure that the system handled such cases gracefully, displaying appropriate error messages when required.

Email and SMS: The system was tested with extreme configurations for email and SMS. We sent alerts to invalid email addresses and phone numbers to evaluate the system's error handling. The system properly reported errors and provided meaningful error messages. Additionally, we tested with long email subjects and message contents, and the system managed to handle the large payloads without any issues.

2) **Stress test**

Acceleration Load: We applied a high load on the accelerometer by generating a continuous stream of acceleration data with varying magnitudes. The system demonstrated robustness and stability, handling the increased data volume without any slowdowns, data loss, or crashes. It consistently processed and reacted to the high influx of acceleration data, ensuring accurate overspeed detection.

Concurrent Overspeed Events: We simulated multiple overspeed events occurring simultaneously to test the system's concurrency handling. The system successfully detected and processed all the concurrent events, generating alerts for each event without delays or interference between them. This highlighted the system's ability to handle and report multiple overspeed conditions concurrently.

Continuous Operation: We ran the system continuously for an extended period to assess its stability and long-term performance. Throughout the test, the system remained stable without any memory leaks, resource exhaustion, or performance degradation. It reliably processed acceleration data, detected overspeed events, and delivered alerts consistently over the extended duration.

Load Balancing: To evaluate the system's scalability, we ran multiple instances concurrently and distributed the load across them. Simulating a high number of overspeed events simultaneously, the system effectively distributed and handled the workload across multiple instances. It demonstrated optimal performance and response times, ensuring timely alerts and accurate data recording.

Fault Tolerance and Recovery: We intentionally introduced unexpected failures during operation, such as disconnecting and reconnecting hardware components or interrupting network connections. The system exhibited fault tolerance and recovery capabilities, seamlessly handling these failures. It recovered gracefully without losing critical data or functionality, resuming normal operation once the failures were resolved.

B. Validation

The validation part is mainly validating the overall system functionality. The system is giving alerts with different threshold values, we validated the result with 5,10,15 m/s^2 values. Also, we check the thingspeak iterations and is functioning well. The CSV data is recorded in the thingsview itself and we are able to download it whenever we want. Once this is done, we validated the GPS location we got from the GPS data is accurate or not, we tested from our cham campus, we open the Gmap link from the alert message and located, as we can see it from the fig that, location is precise.

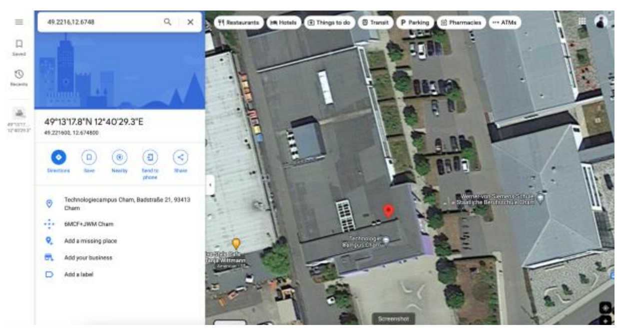


Fig. 14. Gmap link when opened.

The location symbol is the location given from our GPS module and we can see our Technology Campus, Cham in the satellite view. This comes to validate that; system is precise to give accurate results.

We validate that the device is not heating up even with 3hr of continuous use. The proper airflow design of the casing provides proper airflow to the device and keeps the device at atmospheric temperature.

VIII. Performance analysis

The performance analysis of the implemented acceleration sensor box with automated data transfer entails analyzing several components of the system to determine its efficiency, dependability, and effectiveness in meeting the specified objectives. This study seeks to offer a thorough knowledge of the system's performance characteristics and to identify any areas that need to be improved. The system's ability to convey data justifies scrutiny. A crucial component is the automatic sending of acceleration and position data to a server via a web interface. Assessing the dependability and speed of data transfer is the main goal of the performance analysis. The Raspberry Pi zero microcontroller is in charge of carrying out the data transmission procedure, and it is essential that it establishes a reliable connection and ensures correct data transfer. Potential bottlenecks or inefficiencies can be found and fixed by measuring the success rate and delay of data transfer.

The functioning of the web interface, specifically the ThingSpeak platform, is evaluated during the performance analysis. The web interface is a means of gaining access to the acquired data. It is critical to assess the responsiveness and usability of the interface to maintain a consistent user experience. The analysis looks at web page loading times, interface stability under changing network conditions, and data retrieval ease. Any delays or irregularities in data accessibility may impair system performance, necessitating optimization efforts. The performance analysis takes into account the ThinksView mobile application in addition to the web interface. This program provides an additional method of data access. The evaluation of the application's performance on various mobile devices and operating systems is part of the analysis. The responsiveness, stability, and user-friendliness of the system are all evaluated. To improve user pleasure and usability, any flaws or constraints influencing the application's performance are recognized and remedied.

Another critical component of the performance analysis is determining the accuracy and reliability of the ADXL345 sensor's acceleration and position measurements. The sensor's accuracy in gathering acceleration data and consistency in providing accurate location information are critical to the overall performance of the system. The study can establish the sensor's dependability and identify any potential inaccuracies or deviations by comparing the sensor results to known values or benchmark measurements.

The performance evaluation includes an evaluation of the emergency SMS alert capabilities using the Fast2SMS. This feature is crucial for quickly warning users in emergency situations. The investigation focuses on the dependability of SMS notifications, ensuring that they are dispatched on time and reach the target recipients without delay. Evaluating the SMS delivery success rate and the Fast2SMS response time assists in identifying any potential flaws or areas for improvement in the emergency alert system.

IX. Risk and challenges

The potential risks and challenges of implementing an acceleration detecting box with automatic data transfer are as follows:

A. Technical risks and challenges

1) **Sensor Integration**
   Compatibility difficulties and communication protocol settings may develop when integrating the ADXL345 sensor with the Raspberry Pi zero microcontroller.
2) **Data Acquisition and Processing**
   Accurate and dependable data acquisition from sensors, as well as the implementation of precise algorithms for data processing, can be technically difficult.
3) **Communication Protocols**
   To circumvent potential network connectivity and data transmission issues, effective communication protocols must be established between the Raspberry Pi zero, GSM/GPRS module, and the server.
4) **Web Interface Development**
   Creating a user-friendly online interface that presents obtained acceleration and location data efficiently necessitates knowledge of web development and visualization techniques.
5) **Mobile Application Integration**
   Integrating the ThinksView mobile application with the acceleration sensor box involves taking into account platform-specific needs and enabling seamless data synchronization.

B. Logical risks and challenges

1) **Component Availability**
   From supply chain disruptions or compatibility issues, ensuring the availability of needed components such as the Raspberry Pi zero, ADXL345 sensor, GSM/GPRS module, and extension modules (HAT) may cause challenges.
2) **Component Sourcing**
   It might be difficult to identify reliable suppliers, compare pricing, and manage component sourcing logistics within budget and time constraints.
3) **Assembly and Enclosure Design**
   Careful planning is required for properly assembling components, incorporating extension modules (HAT), and constructing a

suitable enclosure for protection and heat dissipation.

4) **Power Management**
Creating an efficient power management system that meets the system's requirements while also dealing with power fluctuations presents logistical issues.

C. Operational risks and challenges

1) **System Reliability**
The total dependability and stability of the acceleration sensing box, including its hardware elements, software configuration, and communication channels, affects the accuracy of data monitoring.
2) **Security and Privacy**
Implementing strong security measures to secure transmitted data while also protecting user privacy necessitates careful study and adherence to encryption techniques and authentication mechanisms.
3) **Scalability and Flexibility**
A flexible system can adapt to changing requirements and expand to accommodate future expansions and technological advancements.
4) **Maintenance and Support**
In order for the acceleration sensing box to function properly, periodic system updates, bug fixes, and maintenance should be established. The acceleration sensing box must receive ongoing technical support and issue resolution in order to satisfy the users.

X. Conclusion

In this project, the objective is to create a box that will transmit acceleration and position information to a server via a web interface, as well as send an accelerometer box that will transmit acceleration and position data to the server. In addition, the Fast2SMS provides SMS alert functionality for emergencies. The hardware configuration of the system consists of a Raspberry Pi Zero microcontroller, an ADXL345 accelerometer sensor, and a GSM/GPRS module. With the Raspberry Pi Zero acting as the central processing unit and providing the computational power, the system can be tracked. This system also features an ADXL345 accelerometer sensor that measures acceleration on three axes (X, Y, and Z), providing data on movement and positioning. SMS alerts can be sent using the GSM/GPRS module, which allows the system to communicate with the mobile network. As a result of the Internet connectivity, the acceleration and position data collected during the experiment need to be transmitted via the web interface to a server. The Raspberry Pi Zero can be connected using either Wi-Fi or Ethernet connections. To ensure that the Raspberry Pi Zero and the associated components remain adequately powered, power supply considerations are required. It is possible to transmit data using the web page ThingSpeak, which allows for data storage and accessibility. You can also use ThinksView on your mobile phone to visualize the data. Ultimately, this project is designed to create an accelerometer box that uses a Raspberry Pi Zero microcontroller, an ADXL345 accelerometer sensor, and a GSM/GPRS module that sends SMS messages based on the acceleration measured by the device. By using a web interface, the system transmits data about accelerations and positions to a server. ThingSpeak allows users to access and analyze data both on the web, as well as on mobile devices.

XI. Result

The focus of this project to web interface the accelerometer reading of a moving object and the alert message for the user was successful. Also, Using GPS and a web interface called ThingSpeak, the user or the customized person can know not only the exact location of the moving object, but also the acceleration. With a 3D designed box the components were fixed accurately, and we have implemented and tested them in an Automobile.

XII. Future scope

For the future development and enhancement of this project, several potential tasks can be considered for consideration in the future:

- Real-time Data Visualization: A real-time data visualization feature will allow you to display accelerometer data visually on a web interface or mobile application to facilitate a better understanding and analysis of the data. In addition, graphs, charts, and other visual representations can also be used to facilitate the analysis and understanding of data.
- Autonomous Vehicle Integration: Integrate the accelerometer box with autonomous vehicles to improve navigation, collision avoidance, and overall safety. The accelerometer data can be used as part of the vehicle's sensor suite to improve safety.
- Vehicle Performance Optimization: Optimizing vehicle performance and efficiency by using accelerometer data. This system identifies opportunities for improving fuel economy, reducing emissions, and improving vehicle performance based on analysis of acceleration patterns and vibrations.
- Driver Monitoring and Assistance: Using accelerometer data, monitor driver behavior and

provide real-time assistance, such as detecting driver fatigue, analyzing driving patterns, and providing alerts and interventions to improve road safety.

- Spacecraft Health Monitoring: Monitoring the health and performance of spacecraft in real-time using accelerometer data. To ensure the safety and reliability of a spacecraft, vibrations, and accelerations experienced during launch, orbit, and reentry can be analyzed early on to detect potential anomalies or structural issues.
- Spacecraft Navigation and Guidance: Ensure that the accelerometer box is integrated into the navigation systems of spacecraft to enhance guidance, navigation, and control capabilities. This system can enhance trajectory calculations, attitude control, and docking operations by accurately measuring acceleration and position changes.

REFERENCES

[1] "BerryBase," Raspberry Pi, [Online]. Available: https://www.berrybase.de/Pixelpdfdata/Articlepdf/id/2089/onumber/RPIZ-WH.
[2] "etechnophiles," Raspberry Pi, [Online]. Available: https://www.etechnophiles.com/raspberry-pi-zero-gpio-pinout-specifications-programming-language/?utm_content=cmp-true.
[3] P. P, Raspberry Pi Zero W Wireless Projects, 2017.
[4] "Analog Devices," ADXL345, [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf.
[5] "Components101," ADXL345, [Online]. Available: https://components101.com/modules/adxl345-accelerometer-module.
[6] "Copperhill Technologies," [Online]. Available: https://copperhilltech.com/content/GSM_GPRS_GNSS_HAT_User_Manual_EN.pdf.
[7]A.M. Kassim[1], A. K. R. A. Jaya[2], A. H. Azahar[3], H. I. Jaafar[4], S Sivarao[5], F. A. Jafar[6], M. S. M. Aras[7] “Performance Analysis of Acceleration Sensor for Movement Detection in Vehicle Security System”Centre of Excellence for Robotic, and Industrial Automation (CERIA) Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya Durian Tunggal, 76100 Melaka, MALAYSIA
[8] Subimal Bikash Chaudary. Vibration Monitoring of Rotating Machines Using MEMS Accelerometer. – Intern. Journal of Scientific Engineering and Research. – Vol. 2 Issue 9, September 2014 – pp. 7.
[9] 3-Axis, ±2 g/±4 g/±8 g/±16 g Digital Accelerometer Data Sheet ADXL345. http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf.
[10]Zhao Xiangxin. Research on Fall Monitoring Based on 3D Acceleration Sensor [D]. Zhejiang University, 2008.
[11]fast2sms documentation
https://docs.fast2sms.com/#overview
[12] adxl345 interfacing https://pimylifeup.com/raspberry-pi-accelerometer-adxl345/
[13] google app passwords
https://support.google.com/accounts/answer/185833?hl=en
[14] adafruit library https://learn.adafruit.com/adxl345-digital-accelerometer/python-circuitpython
[15]serial communication in python on raspberry pi
https://www.raspberrypi.org/forums/viewtopic.php?t=12345