Collecting Data and Visualizing of Sensors of an Android Device.

A PROJECT REPORT

Submitted by
Ritwik Parija(170301150003)
Ayush Kumar (170301150022)
Ambikesh Parida(170301141021)
Hritesh Mallik(170301120070)

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT (CUTM)
BHUBANESWAR, ODISHA

BONAFIED CERTIFICATE

Certified that this project report "Hand Written Digit Recognition Based Learning Android Application" is the work of **Ritwik Parija,Ayush Kumar,Ambikesh Parida,Hritesh Mallik** who carried out the project work under my supervision. This is to further certify to the best of my knowledge that this project has not been carried out earlier in this institute and the university.

Signature of External Examiner   Signature of Professor

Certified that the above-mentioned project has been duly carried out as per the norms of the college and statutes of the university.

SIGNATURE

Mr. Harish Chandra Mohanta

Head of Department
Electronics and Communications Engineering

DECLARATION

We hereby declare that the work entitled "Collecting Data and Visualizing of Sensors of an Android Device" submitted to "**Mr. Subrat Kumar Pradhan**" is a record of an original work done by us under the guidance of "**Mr. Subrat Kumar Pradhan**", Professor from Department of Electronics & Communication Engineering, "**Centurion University of Technology and Management**".

Place: Jatni , Khurda

Ritwik Parija(170301150003)
Ayush Kumar (170301150022)
Ambikesh Parida(170301141021)
Hritesh Mallik(170301120070)

Date:

ACKNOWLEDGEMENT

We wish to express our profound and sincere gratitude to Mr. Subrat Kumar Pradhan, Department of Electronics and Communication Engineering, CUTM Bhubaneswar, who guided us into the intricacies of this project nonchalantly with matchless magnanimity.

We thank Mrs. Swarna Prabha Jena, Dept. of Electronics and Communication Engineering, and CIT Bhubaneswar for extending their support during course of this investigation.

We would like to thank all our friends and especially our classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious.

Ritwik Parija(170301150003)
Ayush Kumar (170301150022)
Ambikesh Parida(170301141021)
Hritesh Mallik(170301120070)

CONTENTS

Abstract

Now a days, smartphones integrate high quality sensors and communication systems, easily managed by simple APPs under Android. On the contrary, managing remote sensors requires a big effort in programming, both on the sensor and on the Smartphone side. In this work, a solution for filling this gap is proposed, providing the instruments to easily handle remote sensors, exactly as if they were embedded in Android environment

CHAPTER 1
INTRODUCTION

When starting any IoT project, the most important materials needed for data collection are sensors. However, buying each individual sensor and finding places to store them can become a hassle, especially as you start making more and more projects. Luckily, there are at least ten high-tech sensors in a device that we use almost everyday: our smart phones. From accelerometers, gyroscopes, pressure and light sensors, to GPS, these devices can be utilized to gather the necessary data to start numerous IoT projects.Now a days, smartphones integrate high quality sensors and communication systems, easily managed by simple APPs under Android. On the contrary, managing remote sensors requires a big effort in programming, both on the sensor and on the Smartphone side. In this work, a solution for filling this gap is proposed, providing the instruments to easily handle remote sensors, exactly as if they were embedded in Android environment..

1.1. Proposed System:

One of the best and easiest ways to collect, visualize, and analyze that smart phone data is with uBeac. uBeac is a versatile IoT platform for centralized digital transformation, data integration, and visualization. uBeac's IoT hub allows you to connect, process, and visualize real-time data in a secure way. In conjunction with the Data Collector app, you will be working on your IoT projects in no time.There are at least ten high-tech sensors in a device that we use almost everyday: our smart phones. From accelerometers, gyroscopes, pressure and light sensors, to GPS, these devices can be utilized to gather the necessary data to start numerous IoT projects. Take advantage of the sensors in your Android device and visualize the data in an IoT dashboard from uBeac

1.2. Aim and Objectives:

Starting any IoT project, the most important materials needed for data collection are sensors. However, buying each individual sensor and finding places to store When them can become a hassle, especially as you start making more and more projects. Luckily, there are at least ten high-tech sensors in a device that we use almost everyday: our smart phones. From accelerometers, gyroscopes, pressure and light sensors, to GPS, these devices can be utilized to gather the necessary data to start numerous IoT projects

1.3. Features:

Smartphone sensor data provides organizations with a chance to learn about their employees' work patterns and behavior. IT could apply this data to the user authorization process to more accurately identify the users logging in to the device.

Organizations can also use sensor data to perform sentiment analysis to determine how satisfied employees are with their jobs. IT can set up this data collection to avoid tracking users individually and instead aggregate the data for AI analysis.

This data can help management understand group morale of their mobile users and what sort of events might cause stress. For example, sensor data could monitor employee engagement and attitudes after management introduces new technologies, modifies existing workflows or makes other organizational changes.

Organizations can use the data to streamline day-to-day operations as well. Organizations can analyze how employees move through their environments and interact with each other on a day-to-day basis, for example. The organization could then adjust its floor plans and workflows to optimize efficiency and productivity. This approach can also help organizations address accessibility issues for employees with disabilities.

CHAPTER 2
LITRETURE REVIEW

Serial Number	Title Of the Paper	Author & Publisher	Problem Statement	Methodology	Outcomes
1	**Integrating remote sensors in a smartphone: The project "sensors for ANDROID in embedded systems"**	**1.Stefano Rinaldi 2.Alessandro Depari 3.Alessandra Flammini 4.Angelo Vezzoli Published by- (Department of Information Engineering, University of Brescia,April 2016)**	**To find out the accurate Data from a android Device and analysis the data.**	**They used the sensors that are attached to an android device and connected to a physical alarm system through Bluetooth**	**To create the network between sensor devices and smart device**

Serial Number	Title Of the Paper	Author & Publisher	Problem Statement	Methodology	Outcomes
1	Integration of Sensors and Mobile Phones	1- Awais Munwar Quareshi 2- Amar Maseed Publisher- National University of Technology 2018	Propose of an Open source Platform based on open system Architecture Design for the integration of mobile phones and sensors	Using Software tools 1-Ni Labview 2-Carbide C/C++ 3-Symbian SDK 4-Sybian C++	Got the Sensor data

CHAPTER 3
REQUIREMENT AND DETAILS

3.1. Software Requirement:

3.1.1 uBeac IOT Platform:

uBeac is a versatile IoT platform for centralized digital transformation, data integration, and visualization. uBeac's IoT hub allows you to connect, process, and visualize real-time data in a secure way. Build an enterprise IoT solution that securely scales to millions of devices.

3.1.1 Features:

A. **3.1.1. Cloud Platform**:

i. Uses our freeware hosted cloud-based service, focus on your IoT solution.

ii. UBeac contains high availability, reliability, scalability, and security.

iii. Connects our gateways and devices with defined URIs, IPs, certificates, custom headers, and usernames/passwords.

3.1.1 Integration:

i. Combine your data flow with external services using custom scripts and powerful API.

ii. Route device data back to existing services such as AWS, IBM, or Microsoft Power BI.

iii. Integrate with external messaging services such as IFTTT, Twilio, WhatsApp, Telegram, SMS, and Email to enable alerts and triggers.

3.1.1 Data Insights:

i. Utilize charts, indicators, gauges, maps, and graphs for seamless visualizations and create dashboards using these widgets to deliver insights.

ii. Real-time and historical time-series IoT data are available for dynamic analytics.

3.1.1 Supported Devices:

i. Connects to ESP32, ESP8266, Arduino, Raspberry Pi, smart phones, and other industrial IoT Devices.

ii. Processes gateway data and extract sensor data using edge computing.

3.1.1 Sensors Types:

i. Support location (GPS), temperature, humidity, acceleration, rotational motion, current, voltage, magnetic field, pressure, and illuminance.

ii. You can define your own sensor type and unit.

3.1.1 Connectivity:

i. Connect to any device over Wi-Fi, LoRa, GPRS, LTE, NB-IoT, BLE, Zigbee, Thread, and Z-Wave.

ii. The platform supports HTTP, WebSocket, MQTT, and CoAP messaging protocols to connect devices.

3.1.2 Data Collector App

Android-based application that helps to collect data from many sensors and send it in JSON format to unlimited number of servers using HTTP, HTTPS, Web Socket, Web Socket Secure protocols with specified sample interval or at time periods.

3.1.2 Features:

i. The Application design is responsive both for small smartphone screens and big tablet screens.

ii. The Application interface is understandable and succinct, that allows to focus on data received from Bluetooth devices.

iii. All operations start after one or two clicks. No need to do multiple click to start operation.

iv. You are provided with a full control over data collection and transfer, you can set any interaction parameter based on your needs.

v. It has comparatively small size, that allows you to install and run it on devices with different capability.

vi. It is supported by Android 4.3 and later devices.

vii. It also contains four Modules

- Sensors
- Servers
- Task
- Settings

3.1.3 Sensors:

An access to all modules is available through Menu icon in upper-left corner of the App. The screen of Sensors, Servers, Tasks, Settings modules are arranged in the same way as the lists of module elements and with Add button, which is used for new elements adding. Pressing Sensors list element opens a screen for review and editing this module as well as opens dialog windows for editing and deleting elements in Server and Tasks modules.An access to all modules is available through icon in upper-left corner of the App. The screen of Sensors, Servers, Tasks, Settings modules are arranged in the same way as the lists of module elements and with button, which is used for new elements adding. Pressing Sensors list element opens a screen for review and editing this module as well as opens dialog windows for editing and deleting elements in Server and Tasks modules.

3.1.4 Sensors Module:

Initially the list of this module contains internal sensors, which are detected by Data Collector at mobile device. Upon choosing such sensor at the screen, a screen will appear where a new sensor name can be assigned or sensor can be switched on/off. For all hardware sensors current values and their plots are indicated, except location sensor, for which a map with current position data is shown. Using a Time/Division slider bar you can specify time sweep for a plot. GPS, network, passive provider data source can be assigned for location sensor.
Adding of guest sensors is initiated by pressing Add button, which activates dialog window. In this dialog window sensor type and name are supposed to be indicated. Upon saving the sensor, it will appear in the list of elements of Sensors module. Pressing such element will open a screen where a new name of the sensor can be assigned or the sensor can be switched on/off as well as required values of the sensor can be assigned. In the top left corner of guest sensors there is a menu with Clear Sensor, Delete Sensor, Settings commands. In Settings setting of current sensor data mode is available (updating of sensor value in tasks):

Manually (data is updated upon pressing Send button).Upon changes (data is updated upon any change of sensor value).Periodic (data is updated in assigned period in seconds. Period can be assigned in the text field below)

In the current version the following guest sensors are performed:
Text Sensor, that transfers any text Mosaic Sensor presents arbitrary size-matrix, where you can assign any color to any cell. On sensor's screen the matrix size and current color for cells filling can be assign. Matrix scale is manually changeable and scrollable. To fill the cell with a color, make a single click on it. Double click will clear it. For continual cells filling first make a long press on a start cell, and then move your finger on filled cells.
XY sensor, that transfers the value of the assigned Y(X) function. X value interval and Y(X) function formula are to be assigned by user. X value is changeable in tree ways: entering the

number, decreasing or increasing the current X value for a set step, using cyclic slider bar. In Y(X) function formula the following can be used: operators, constants and functions described here.

3.1.5 Servers Module:

In this module servers are assigned, which will be used in task settings for sensors data transmittance. Servers number is unlimited. Each list element includes the following: server name, URL to send sensors data through, server connection status, as well as WS protocols for Web Socket and Web Socket Secure servers. Connection status is updated upon each attempt of data transmittance. If server is switched off in Settings, the status will be "disable". Before the first attempt of data sending the status of server will be indicated as "unknow".

New server adding will be available after pressing Add button, that opens dialog window for entering server parameters. In this window the following can be indicated: the name and URL of the server, login and password is authorization is required, set/ reset of "Enable" flag, as well as WS protocols separated by commas for Web Socket and Web Socket Secure.

The following URL scheme is applied <protocol>://<host>[:<port>][</script>], where
<protocol> - http, https, ws, wss,
<host> - IP address or domain name of the server,
<port> - the number of TCP port, which listens the server for request receipt,
</script> - server program, that processes data from sensors.
Parameters enclosed in square brackets are optional.

Pressing list element in this module opens the same dialog window as at new server adding, where any above mentioned server parameter can be changed as well as server deleted by pressing "Delete" button.

Servers with set "Enable" flags and correct URLs are used in tasks. If servers status is "unknow" for a long time, then first check these two parameters correctness.

- **Task Module:**
  In Data Collector two types of tasks are presented - a simple task, that sends data in assigned period in milliseconds, and a task, that send data in timepoints, assigned in cron-format. All data are sent in JSON format.

  Each list element of this module included the following:

- Task name.
- The list of sensors, the data of which is sent to server.
- The name of the server, the data is sent for.
- Task execution status (running, disable, invalid cron-pattern).
- Schedule parameters. Simple schedule requires indication of sampling period from 1 to 60000 ms. For cron-format schedule a cron-sample shall be assigned, for example */1 * * * *, number of samples and sampling period in ms.

New task adding is will be available upon pressing Add button, which opens dialog window for task parameters entering. Task name, flag set/ reset are to be indicated in this window. In Sensors list required sensors can be indicated. The server to which data is to be sent can be chosen from Server list. If a simple schedule is chosen in this window, sampling period from 1 to 60000 ms is to be assigned. If a cron-format schedule is chosen, then a correct cron-pattern, samples number and sampling period shall be assigned. In the letter case the task will run at the moments indicated in the pattern and make indicated number of samples with assigned sampling period for each sensor. If Enable flag is set and Save button is pressed, then the task will run immediately.

Pressing the list element in this module will open the same dialog window as at task adding, where any above mentioned task parameter can be changed as well as task can be deleted by pressing Delete button. If Enable flag was reset and Save button pressed or a task was deleted, then the task immediately would stop execution.

- **Setting Module:**

In Settings a general parameters are indicated, such as:
Servers timeout from 1 to 5 sec., after which server will be considered unavailable.
Data format, where in dialog window you may indicate whether to switch or not to switch UUID transmittance of mobile device, task name, sensors names, units of measurement of sensors data.
Background Service flag. If the flag is set, then tasks will continue to run even after Data Collector closure.
Home screen enables to set chosen module running upon application start.
Screen orientation (automatic, portrait, landscape) enables to set screen behavior during mobile device orientation change.
UUID assigns a unique identifier of mobile device by which it's data packages can be identified from other mobile devices, if they all send data to the same.

3.2. Hardware Requirement:

3.2.1 An Android device

An Android device is a device that runs on the Android operating system. Android is an array of software intended for mobile devices that features an operating system, core applications and middleware.

An Android device may be a smartphone, tablet PC, e-book reader or any type of mobile device that requires an OS.

Android is developed by the Open Handset Alliance, which is led by Google. Some of the well-known Android device manufacturers include Acer, HTC, Samsung, LG, Sony Ericsson and Motorola.

3.2.2 Features:

i. **Messaging-** SMS and MMS are available forms of messaging, including threaded text messaging and Android Cloud To Device Messaging (C2DM) and now enhanced version of C2DM, Android Google Cloud Messaging (GCM) is also a part of Android Push Messaging services.

ii. **Auto Correction and Dictionary**-Android Operating System has an interesting feature called Auto Correction. When any word is misspelled, then Android recommends the meaningful and correct words matching the words that are available in Dictionary.

iii. **Web browser**-The web browser available in Android is based on the open-source Blink (previously WebKit) layout engine, coupled with Chromium's V8 JavaScript engine. Then the WebKit-using Android Browser scored 100/100 on the Acid3 test on Android 4.0 ICS; the Blink-based browser currently has better standards support. Google has begun licensing Google Chrome (a proprietary software) separately from Android.

iv. **Voice-based features-** Google search through voice has been available since initial release.[5] Voice actions for calling, texting, navigation, etc. are supported on Android 2.2 onwards, Android 4.1, Google has expanded Voice Actions with ability to talk back and read answers from Google's Knowledge Graph when queried with specific commands.

v. **Multi-touch-** Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. The feature was originally disabled at the kernel level Google has since released an update for the Nexus One and the Motorola Droid which enables multi-touch natively.

vi. **Multitasking-** Multitasking of applications, with unique handling of memory allocation.

vii. **Multiple language support-**Android supports multiple languages.

viii. **Accessibility-**Built-in text-to-speech is provided by *TalkBack* for people with low or no vision. Enhancements for people with hearing difficulties are available, as are other aids.

3.2.3 Connectivity-

i. Android supports connectivity Technologies including GSM/EDGE, Bluetooth,LTE,CDMA,EV,DP,UMTS,NFC,IDEN and WiMAX.

ii. Android supports tethering, which allows a phone to be used as a wireless/wired Wi-Fi hotspot. Before Android 2.2, this was supported by third-party applications or manufacturer customizations**.**

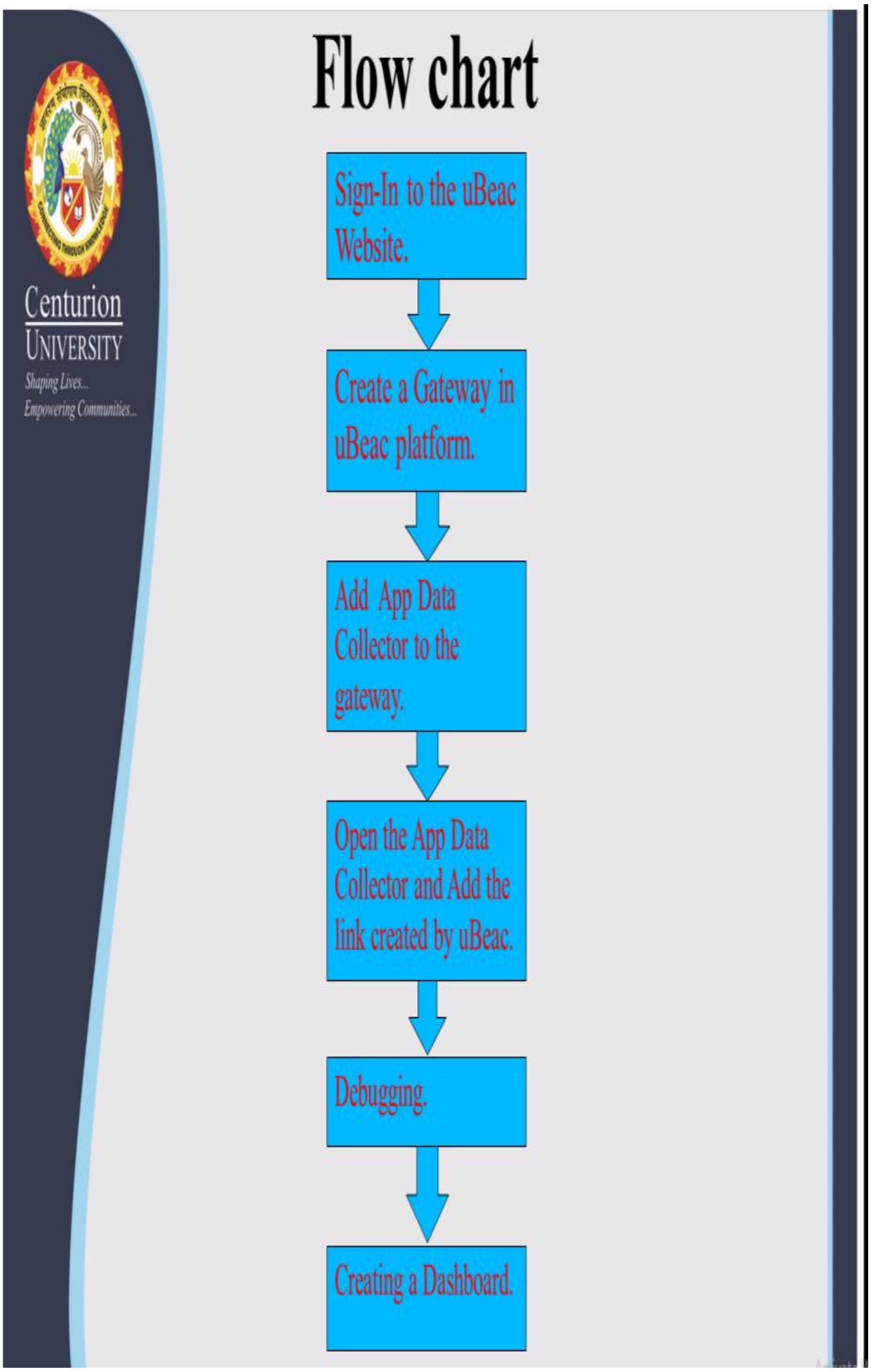
3.2.4 Hardware support-

I. Android devices can include still/video cameras, touchscreens, GPS, accelerometers, gyroscopes, barometers, magneto meters, dedicated gaming controls, proximity and pressure sensors, thermometers, accelerated 2D bit blits (with hardware orientation, scaling, pixel format conversion) and accelerated 3D graphics.

Chapter 4

Methodology

Fig.4.3 (Block Diagram of connecting To Ubeac)



4.1. Software Installation Process:

4.1.2 Signing up with Ubeac:

i. Signing up with uBeac is easy. You just need to add your email and create a password to get started.

ii. You must create a team. Involving just a name for the team, a code name, and an address.

iii. Once you have finished this, you will arrive at a blank homepage for your team.

iv. We need to create a gateway to connect all of your devices. From the ubeac homepage.

v. We need to create a gateway to connect all of your devices. From the ubeac homepage.

vi. Homepage,click on the **Gateways** module and add a new gateway. Under the **General** tab, create a UID and name for your gateway.

vii. Under the HTTP tab, there will be two gateway URLs: one HTTP and one HTTPS. These will be used to connect to your phone. Click submit to add the gateway.

viii. This type of blank page you will find after you click submit button.

4.2.1. Install DATA Collector app:

i. First go to your Android Device and install Data Collector App.

ii. After installation Open the app

iii. After opening the app you will find Four module in the app at left corner

iv. The Sensors module displays a list of all sensors connected to the device.

    Clicking on each sensor will display the current data that is being read from that specific sensor.

v. The Servers module allows the user to add an unlimited number of servers to read the sensor data.

vi. The Tasks module lets users assign what, where, when and how data will be sent from the Android device to the servers.

4.2.2 Connecting DATA collector to Ubeac

i. On the Data Collector app, select the Servers module and add a new server.

ii. Provide a name for this server and add one of the Gateway URLs.

iii. We created a task to send to the server.

iv. The Tasks module and add a new task.

v. Then choose the sensors that will be used from your phone.

vi. Select all the sensors.

vii. Then choose the sensors that will be used from your phone.

viii. Select the server you had just created, and add a sampling time for how often you want Data Collector to send data to uBeac.

ix. Finally, in Settings, select Data Format and select all options except Strict JSON

4.2.3 Debugging:

i. Select the Gateways module to see that a device has been added to the gateway.

ii. If you click your gateway, you can see all the HTTP POST requests.

iii. If you select the Devices module and click on the newly added device, which is the Android device, you can find all the data that each sensor is sending to uBeac

4.2.4 Creating a dashboard

i. You can select widgets such as indicators, charts, and device tracker to help you visualize your data.

ii. you would first drag and drop the indicator widget onto the dashboard. Next, you would click the “connect to data” button to edit the widget's settings.

iii. Once you are satisfied with your widget, save your progress. You can continue doing this for as many widgets as you would like.

.

Chapter 5

Sensor and their Types

5.3.1 Introduction:

A sensor is a component used in mobile devices, the purpose of which is to detect changes in the environment (such as changes in brightness, magnetic fields, temperature, and gravity) and movement (such as the device being moved, flipped, or picked up), and convert them into electronic signals that can be processed by the device.

5.3.2 Types of Sensors:

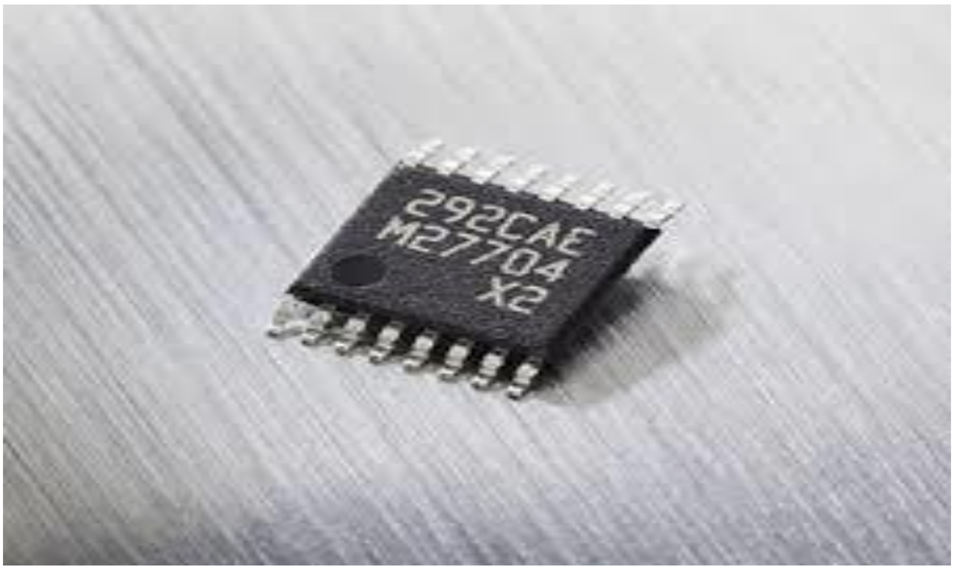
i. Location Sensors
   - Wifi
   - Cell-ID
   - GPS

ii. Position Sensors
   - Orientation Sensor
   - Proximity Sensor

iii. Motion Sensors
   - Accelerometer
   - Gyroscope

iv. Environmental Sensors
   - Ambient Temperature
   - Ambient Light
   - Atmospheric Pressure
   - Realtive Humidity
   - Magnetic Field
   - Miscellaneous Sensors

5.3.3 Location sensors:



IC Name:{ 292CAEM27704X2}

Image of a Location Sensor

The LocationSensor is used to communicate with the global positioning satellite receiver (GPS) in your phone/tablet. When the LocationSensor communicates with the built-in GPS receiver, the GPS determines the location of your device. The sensor can also work with network/wifi location services. Finding a location through the network uses very different techniques than with a GPS. Location means the device's present latitude and longitude, or it can mean your street address. The measuring units employed in the LocationSensor for distance are meters.

When the sensor reports distance information or you set a distance into the component, the units are in meters. If your app must deal in English units, use the Math blocks to convert units at the time you display them. Calculate everything in meters, then convert to report the result in feet or miles on your display.

A nautical mile is the distance subtended by a minute of angle (1/60 of a degree) across the earth's radius. This means that the distance between a degree of latitude and the next whole degree is sixty nautical miles. A degree of longitude is sixty nautical miles at the equator, but the separation between adjacent whole degrees of longitude diminishes as you change latitude toward the poles. The spacings between degrees of latitude are constant; the spacing between degrees of longitude are variable.

5.3.3 Wireless Sensor Network:

A Wireless sensor network can be defined as a network of devices that can communicate the information gathered from a monitored field through wireless links. The data is forwarded through multiple nodes, and with a gateway, the data is connected to other networks like wireless Ethernet**.**
WSN is a wireless network that consists of base stations and numbers of nodes (wireless sensors). These networks are used to monitor physical or environmental conditions like sound, pressure, temperature, and co-operatively pass data through the network to the main location.

5.3.3.1 Star Topologies

Star topology is a communication topology, where each node connects directly to a gateway. A single gateway can send or receive a message to several remote nodes. Instar topologies, the nodes are not permitted to send messages to each other. This allows low-latency communications between the remote node and the gateway (base station).

5.3.3.2 Mesh Topologies

The Mesh topologies allow transmission of data from one node to another, which is within its radio transmission range. If a node wants to send a message to another node, which is out of the radio communication range, it needs an intermediate node to forward the message to the desired node. The advantage of this mesh topology includes easy isolation and detection of faults in the network. The disadvantage is that the network is large and requires huge investment.

5.3.3.3 Tree Toplogies

Tree topology is also called as a cascaded star topology. In tree topologies, each node connects to a node that is placed higher in the tree, and then to the gateway. The main advantage of the tree topology is that the expansion of a network can be easily possible, and also error detection becomes easy. The disadvantage with this network is that it relies heavily on the bus cable; if it breaks, all the network will collapse**.**

5.3.4 GPS(Global Positioning System)

GPS receivers are generally used in smartphones, fleet management system, military etc. for tracking or finding location. Global Positioning System (GPS) is a satellite-based system that uses satellites and ground stations to measure and compute its position on Earth. GPS is also known as Navigation System with Time and Ranging (NAVSTAR) GPS. GPS receiver needs to receive data from at least 4 satellites for accuracy purpose. GPS receiver does not transmit any information to the satellites. This GPS receiver is used in many applications like smartphones, Cabs, Fleet management etc. A GPS receiver has to have a clear line of sight to the satellite to operate, so dense tree cover and buildings can keep it from getting a fix on your location.GPS receivers and cell phones have a lot in common, and both are very popular. In the next section, we'll look at some of the features of GPS-enabled cell phones A GPS receiver has to have a clear line of sight to the satellite to operate, so dense tree cover and buildings can keep it from getting a fix on your location.

GPS receivers and cell phones have a lot in common, and both are very popular. In the next section, we'll look at some of the features of GPS-enabled cell phones However, some phones have a complete GPS receiver located in the phone or can connect to one with wires or through a Bluetooth connection. These GPS-enabled phones can understand programming languages like Java and can provide turn-by-turn directions or information about nearby businesses and attractions. Others can work like a tracking device.

5.3.4.1.1 How GPS Works

GPS receiver uses a constellation of satellites and ground stations to calculate accurate location wherever it is located.

These GPS satellites transmit information signal over radio frequency (1.1 to 1.5 GHz) to the receiver. With the help of this received information, a ground station or GPS module can compute its position and time.

GPS receiver receives information signals from GPS satellites and calculates its distance from satellites. This is done by measuring the time required for the signal to travel from satellite to the receiver.

Where,
Speed = Speed of Radio signal which is approximately equal to the speed of light i.e.3*10^{8}

Time = Time required for a signal to travel from the satellite to the receiver.

By subtracting the sent time from the received time, we can determine the travel time.

To determine distance, both the satellite and GPS receiver generate the same pseudocode signal at the same time.

The satellite transmits the pseudocode; which is received by the GPS receiver.

These two signals are compared and the difference between the signals is the travel time.

Now, if the receiver knows the distance from 3 or more satellites and their location (which is sent by the satellites), then it can calculate its location by using Trilateration method.

5.3.5 Position Sensors

Position Sensors/Detectors/Transducers are electronic devices used to sense the positions of valves, doors, throttles, etc. and supply signals to the inputs of control or display devices. Key specifications include sensor type, sensor function, measurement range, and features that are specific to the sensor type. Position sensors are used wherever positional information is needed in a myriad of control applications. A common position transducer is a so-called string-pot, or string potentiometer.

5.3.5.1.1 Types of Postion Sensors:

- **Proximity Sensors**
- **Orientation Sensors**

i. Proximity Sensors:

Reporting-mode: On-change

Usually defined as a wake-up sensor

GetDefaultSensor(SENSOR_TYPE_PROXIMITY) returns a wake-up sensor

A proximity sensor reports the distance from the sensor to the closest visible surface.Up to Android 4.4, the proximity sensors were always wake-up sensors, waking up the SoC when detecting a change in proximity. After Android 4.4, we advise to implement the wake-up version of this sensor first, as it's the one that is used to turn the screen on and off while making phone calls.The measurement is reported in centimeters

A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance), and looks for changes in the field or return signal. The object being sensed is often referred to as the proximity sensor's target. Different proximity sensor targets demand different sensors. For example, a capacitive proximity sensor or photoelectric sensor might be suitable for a plastic target; an inductive proximity sensor always requires a metal target.Proximity sensors can have a high reliability and long functional life because of the absence of mechanical parts and lack of physical contact between the sensor and the sensed object.

Proximity sensors are also used in machine vibration monitoring to measure the variation in distance between a shaft and its support bearing. This is common in large steam turbines, compressors, and motors that use sleeve-type bearings.A proximity sensor adjusted to a very short range is often used as a touch switch.

ii. **Orientation Sensor:**

An orientation sensor reports the attitude of the device. The measurements are reported in degrees in the x, y, and z fields. the roll angle is positive in the clockwise direction. (Mathematically speaking, it should be positive in the counter-clockwise direction).

This definition is different from yaw, pitch, and roll used in aviation where the X axis is along the long side of the plane (tail to nose). The orientation sensor derives its data by processing the raw sensor data from the accelerometer and the geomagnetic field sensor. Because of the heavy processing that is involved, the accuracy and precision of the orientation sensor is diminished. Specifically, this sensor is reliable only when the roll angle is 0. As a result, the orientation sensor was deprecated in Android 2.2 (API level 8), and the orientation sensor type was deprecated in Android 4.4W (API level 20). Instead of using raw data from the orientation sensor,

- **Azimuth (degrees of rotation about the -z axis).** This is the angle between the device's current compass direction and magnetic north. If the top edge of the device faces magnetic north, the azimuth is 0 degrees; if the top edge faces south, the azimuth is 180 degrees. Similarly, if the top edge faces east, the azimuth is 90 degrees, and if the top edge faces west, the azimuth is 270 degrees.

- **Pitch (degrees of rotation about the x axis).** This is the angle between a plane parallel to the device's screen and a plane parallel to the ground. If you hold the device parallel to the ground with the bottom edge closest to you and tilt the top edge of the device toward the ground, the pitch angle becomes positive. Tilting in the opposite direction—moving the top edge of the device away from the ground—causes the pitch angle to become negative. The range of values is -180 degrees to 180 degrees.

- **Roll (degrees of rotation about the y axis)**. This is the angle between a plane perpendicular to the device's screen and a plane perpendicular to the ground. If you hold the device parallel to the ground with the bottom edge closest to you and tilt the left edge of the device toward the ground, the roll angle becomes positive. Tilting in the opposite direction—moving the right edge of the device toward the ground— causes the roll angle to become negative. The range of values is -90 degrees to 90 degrees.

5.3.6 Motion Sensors:

The Android platform provides several sensors that let you monitor the motion of a device. The gravity, linear acceleration, rotation vector, significant motion, step counter, and step detector sensors are either hardware-based or software-based.
The accelerometer and gyroscope sensors are always hardware-based. Most Android-powered devices have an accelerometer, and many now include a gyroscope. The availability of the software-based sensors is more variable because they often rely on one or more hardware sensors to derive their data. Depending on the device, these software-based sensors can derive their data either from the accelerometer and magnetometer or from the gyroscope.

Motion sensors are useful for monitoring device movement, such as tilt, shake, rotation, or swing. The movement is usually a reflection of direct user input (for example, a user steering a car in a game or a user controlling a ball in a game), but it can also be a reflection of the physical environment in which the device is sitting (for example, moving with you while you drive your car). In the first case, you are monitoring motion relative to the device's frame of reference or your application's frame of reference; in the second case you are monitoring motion relative to the world's frame of reference. Motion sensors by themselves are not typically used to monitor device position, but they can be used with other sensors, such as the geomagnetic field sensor, to determine a device's position relative to the world's frame of reference.

All of the motion sensors return multi-dimensional arrays of sensor values for each SensorEvent. For example, during a single sensor event the accelerometer returns acceleration force data for the three coordinate axes, and the gyroscope returns rate of rotation data for the three coordinate axes. These data values are returned in a float array (values) along with other SensorEvent parameters.

The rotation vector sensor and the gravity sensor are the most frequently used sensors for motion detection and monitoring. The rotational vector sensor is particularly versatile and can be used for a wide range of motion-related tasks, such as detecting gestures, monitoring angular change, and monitoring relative orientation changes. For example, the rotational vector sensor is ideal if you are developing a game, an augmented reality application, a 2-dimensional or 3-dimensional compass, or a camera stabilization app. In most cases, using these sensors is a better choice than using the accelerometer and geomagnetic field sensor or the orientation sensor.

5.3.7 Accelerometer:

An accelerometer sensor reports the acceleration of the device along the three sensor axes. The measured acceleration includes both the physical acceleration (change of velocity) and the gravity. The measurement is reported in the x, y, and z fields of sensors_event_t.acceleration.
All values are in SI units (m/s^2) and measure the acceleration of the device minus the force of gravity along the three sensor axes.
The bias and scale calibration must only be updated while the sensor is deactivated, so as to avoid causing jumps in values during streaming.
The accelerometer also reports how accurate it expects its readings to be through sensors_event_t.acceleration.status. See the SensorManager's SENSOR_STATUS_* constants for more information on possible values for this field.

Accelerometers have many uses in industry and science. Highly sensitive accelerometers are used in inertial navigation systems for aircraft and missiles. Vibration in rotating machines is monitored by accelerometers. They are used in tablet computers and digital cameras so that images on screens are always displayed upright. In unmanned aerial vehicles, accelerometers help to stabilise flight.

When two or more accelerometers are coordinated with one another, they can measure differences in proper acceleration, particularly gravity, over their separation in space—that is, the gradient of the gravitational field. Gravity gradiometry is useful because absolute gravity is a weak effect and depends on the local density of the Earth, which is quite variable.
Single- and multi-axis accelerometers can detect both the magnitude and the direction of the proper acceleration, as a vector quantity, and can be used to sense orientation (because the direction of weight changes), coordinate acceleration, vibration, shock, and falling in a resistive medium (a case in which the proper acceleration changes, increasing from zero).
Micromachined microelectromechanical systems (MEMS) accelerometers are increasingly present in portable electronic devices and video-game controllers, to detect changes in the positions of these devices.

An accelerometer at rest relative to the Earth's surface will indicate approximately 1 g upwards because the Earth's surface exerts a normal force upwards relative to the local inertial frame (the frame of a freely falling object near the surface). To obtain the acceleration due to motion with respect to the Earth, this "gravity offset" must be subtracted and corrections made for effects caused by the Earth's rotation relative to the inertial frame.
Some smartphones, digital audio players and personal digital assistants contain accelerometers for user interface control; often the accelerometer is used to present landscape or portrait views of the device's screen, based on the way the device is being held.

5.3.8 Gyroscope:

A gyroscope sensor reports the rate of rotation of the device around the three sensor axes. Rotation is positive in the counterclockwise direction (right-hand rule). That is, an observer looking from some positive location on the x, y, or z axis at a device positioned on the origin would report positive rotation if the device appeared to be rotating counter clockwise. Note that this is the standard mathematical definition of positive rotation and does not agree with the aerospace definition of roll.

The readings are calibrated using:

- Temperature compensation
- Factory (or online) scale compensation
- Online bias calibration (to remove drift)

  The gyroscope can't be emulated based on magnetometers and accelerometers, as this would cause it to have reduced local consistency and responsiveness. It must be based on a usual gyroscope chip.

The measurement is reported in the x, y, and z fields of sensors_event_t.gyro and all values are in radians per second (rad/s).

A Gyroscope can be understood as a device that is used to maintain a reference direction or provide stability in navigation, stabilizers, etc. Similarly, a gyroscope or a Gyro sensor is present in your smartphone to sense angular rotational velocity and acceleration.

A Gyro sensor in your phone provides the ability to answer your phone, or open a website by present commands such as rotating, gently shaking the phone 2 to 3 times, etc.

Implementations of Gyroscope Sensor in a Mobile App.

- As discussed earlier, a Gyroscope Sensor can enable a number of actions to take place basis different set of motions done by a user, such as shaking the phone to undo written content.
- Gyroscope sensor is responsible for the autorotation of the screen and view on the screen whenever a phone is rotated.
- One of the biggest implementations of a Gyroscope is that it enables smooth rotations and execution of multiple commands in games by 3D motions.
- Gyroscope is capable of providing precision motion inside the App functionality. This allows the user to execute majority of the tasks with the motion of the device itself.
- Gyroscope captures a 6-dimensional angular motion. This simply means, the mobile apps that are developed using Gyroscope sensor are much likely to provide an alluring user experience than the one without the sensor.

5.3.9 Enviornmental Sensors:

The Android platform provides four sensors that let you monitor various environmental properties. You can use these sensors to monitor relative ambient humidity, illuminance, ambient pressure, and ambient temperature near an Android-powered device. All four environment sensors are hardware-based and are available only if a device manufacturer has built them into a device. With the exception of the light sensor, which most device manufacturers use to control screen brightness, environment sensors are not always available on devices. Because of this, it's particularly important that you verify at runtime whether an environment sensor exists before you attempt to acquire data from it.

Unlike most motion sensors and position sensors, which return a multi-dimensional array of sensor values for each SensorEvent, environment sensors return a single sensor value for each data event. For example, the temperature in °C or the pressure in hPa. Also, unlike motion sensors and position sensors, which often require high-pass or low-pass filtering, environment sensors do not typically require any data filtering or data processing.

Environmental sensors are connected objects capable of providing various types of information: location, position, the individual’s movements and contextual elements which can be compared to data collected via sensors embedded on or implanted in the individual, including the validation of alarms, like in the case of falls. They pose particular ethical problems as they are a type of surveillance which can affect the individual’s private life, even their privacy, depending on where they are placed. This point is particularly sensitive in the case of video capture. This type of sensor can also be associated with robotic devices with which they interact to allow them to adapt to the context or the need of the person with whom they are meant to interact. The monitoring of the individual and their health is not the first objective for some sensors as air quality sensors, light sensors, smoke detectors, etc. exist. However, data they collect can, by cross-referencing with data from other sources, contribute to the production of potentially personalized health information, and eventually the generation of alarms.

The raw data you acquire from the light, pressure, and temperature sensors usually requires no calibration, filtering, or modification, which makes them some of the easiest sensors to use.

5.3.10 Ambient Temperature:

This sensor provides the ambient (room) temperature in degrees Celsius**.** Measuring air temperature at a high spatial resolution is very important for many applications including detection of urban heat islands. However, air temperature is currently measured by weather stations those are very sparse spatially.

In this paper, we propose a new approach to estimate air temperature using smartphones in different contexts. Most of the smartphones are not equipped with air temperature sensors but they are all equipped with battery temperature sensors. When a smartphone is in idle state, its battery temperature is stable and correlated with ambient air temperature. Furthermore, it is often carried close to human body, e.g. in pockets of coats, trousers and in hand. Therefore we developed a new approach of using two linear regression models to estimate air temperature from the idle smartphones battery temperature given their in-pocket or out-of-pocket positions.

Temperature is a physical quantity that is very important to human health. In urban areas due to different concentration of roads, buildings and population, there exists urban heat islands. An UHI is an urban area that is significantly warmer than its surroundings. The UHI has bad consequences such as decreasing air quality, water quality and directly influencing human health. Thus a number of UHI mitigation attempts have been made to improve Climate change

Crowdsourcing for environment data collection is a new approach due to availability of sensor-equipped smartphones. Some models of smartphones, for example Samsung Galaxy S4, Samsung Galaxy Note 3, Motorola Moto X, Huawei Ascend P6 and Xiaomi Mi3, are equipped with environment sensors including temperature ones.

However, in an area with a small number of smartphones, the estimated temperature of the app is not very accurate. Furthermore, this approach provides aggregated temperature of an area from a large number of smartphone battery temperature readings rather than its temperature distribution.

When a smartphone is in idle state, its battery temperature is stable and is correlated with ambient air temperature. An idle state of a smartphone is the one that is achieved when the smartphone is unplugged, its screen is off and its battery temperature variance in a temporal window is small enough. When the smartphone is not idle, its battery temperature fluctuates and depends highly on many factors, for example CPU load, screen brightness level, 3G/WiFi and GPS status. Battery temperature of an idle smartphone is considered data and that of a non-idle one is noise. Air temperature prediction models will be built based on the data only.

5.3.11 Ambient Light Sensor:

An ambient light sensor is a component in smartphones, notebooks, other mobile devices, automotive displays and LCD TVs. It is a photodetector that is used to sense the amount of ambient light present, and appropriately dim the device's screen to match it. This avoids having the screen be too bright when the user's pupils are adapted for vision in a dark room, or too dim when the device is used outdoors in the daytime. Dimming the screen on a mobile device also prolongs the lifetime of the battery.

The standard international unit for the illuminance of ambient light is the lux. The typical performance of an ambient light sensor is from less than 50 lux in dim light to over 10,000 lux at noon.

There are three common types of ambient light sensor: phototransistors, photodiodes, and photonic integrated circuits, which integrate a photodetector and an amplifier in one device.

The sensor is a light-dependent resistor (LDR). which has a resistance that varies with ambient light intensity. The ALS provides measurements of ambient light intensity which match the human eye's response to light under a variety of lighting conditions, and through a variety of attenuation materials.

The light sensor detects the lighting levels in the vicinity to adjust the display brightness accordingly. It is used in Automatic Brightness Adjuster to decrease or increase the brightness of the smartphone screen based on the availability of light.

The Ambient Light Sensor (ALS) products from ams provide measurements of ambient light intensity which match the human eye's response to light under a variety of lighting conditions. Each device has a specific operating range of performance, from very low light up to bright sunlight.

Light sensors have a lot of uses. The most common use in our daily lives is in cell phones and tablets. Most portable personal electronics now have ambient light sensors used to adjust brightness.

5.3.12 Atmospheric Pressure Sensor:

A barometric pressure sensor for smartphones, wearable and hearable devices.
The BMP390 delivers altitude tracking in smartphones as well as wearable and hearable devices.The sensor can measure height changes below 10 cm.

This is accomplished by applying barometric pressure sensors in the phone, like the Bosch BMP390, and combining them with NextNav's Metropolitan Beacon System (MBS) z-axis service to determine three-dimensional location and positioning.

Not limited to emergency applications, the sensor also enables improved indoor navigation in general, for example in combination with the Position Tracking Smart Sensor BHI160BP.These navigation solutions compensate for traditional localization technologies such as GPS that do not work efficiently in shielded environments.
This will help users to save time and avoid the hassle of getting lost, for example when searching for their car in an underground garage.

The device supports enhanced GPS applications for outdoor navigation and calorie expenditure estimation tasks.

The use of advanced barometric pressure sensing can determine whether a user is walking up or down an incline, stairs or lifting weights during a fitness training session.

This helps to increase the precision of calorie tracking by up to 15 percent2. Thanks to the improved accuracy of altitude measurements, fitness trackers are able to show exactly how far a user has run, walked or cycled.

The device provides a typical relative accuracy of ±0.03 hPa, which is superior to any other comparable product on the market today.

Typical absolute accuracy is ±0.5 hPa. The excellent accuracy is the result of significant improvements in temperature stability, drift behavior and noise.

The sensor offers high temperature stability across its entire operating temperature and pressure range of 0 to 65 °C and 700 to 1100 hPa respectively, with an average temperature coefficient offset (TCO) of just ±0.6 Pa/K. Noise is also low, at only 0.9 Pa typical, an improvement of 25 percent relative to the predecessor BMP380. The device also provides high long-term stability, and low short- and long-term drift.
Its measures 2.0 mm x 2.0 mm x 0.75 mm, which makes the sensor easy to integrate into portable devices.

Power consumption is kept at 3.2 µA at 1 Hz (typical) to maximize battery life on portable devices.

5.3.13 Magnetic Field sensor:

A MEMS magnetic field sensor is a small-scale microelectromechanical systems (MEMS) device for detecting and measuring magnetic fields (Magnetometer). Many of these operate by detecting effects of the Lorentz force: a change in voltage or resonant frequency may be measured electronically, or a mechanical displacement may be measured optically. Compensation for temperature effects is necessary. Its use as a miniaturized compass
The Magnetic Field Sensor can be used to study the field around permanent magnets, coils, and electrical devices. It features a rotating sensor tip to measure both transverse and longitudinal magnetic fields.

- **Magnetic Field Senseing**:

  Magnetometers can be categorized into four general types[1] depending on the magnitude of the measured field. If the targeted B-field is larger than the earth magnetic field (maximum value around 60 µT), the sensor does not need to be very sensitive. To measure the earth field larger than the geomagnetic noise(around 0.1 nT), better sensors are required. For the application of magnetic anomaly detection, sensors at different locations have to be used to cancel the spatial-correlated noise in order to achieve a better spatial resolution. To measure the field below the geomagnetic noise, much more sensitive magnetic field sensors have to be employed. These sensors are mainly used in medical and biomedical applications, such as MRI and molecule tagging.

- **Advantages of MEMS-based sensors:**

  A MEMS-based magnetic field sensor is small, so it can be placed close to the measurement location and thereby achieve higher spatial resolution than other magnetic field sensors. Additionally, constructing a MEMS magnetic field sensor does not require the microfabrication of magnetic material. Therefore, the cost of the sensor can be greatly reduced. Integration of MEMS sensor and microelectronics can further reduce the size of the entire magnetic field sensing system.

- **Application:**

  Detect flaws of electrically conductive material:
  Magnetometers based on piezoelectric resonators can be applied to finding flaws in safety-critical metal structures, such as airplane propellers, engines, fuselage and wing structures, or high pressure oil or gas pipelines.

  Monitoring health of organs of thoracic cavity:

  When we breathe, the nerves and muscles of our thoracic cavity create a weak magnetic field. Magnetometers based on piezoelectric resonators have high resolution (in the range of nT), allowing solid-state sensing of our respiratory system.

5.3.14 Miscellanous Sensor:

- **Relative humidity**

  Reporting-mode: On-change

  A relative humidity sensor measures relative ambient air humidity and returns a value in percent.

- **Linear acceleration**

  Reporting-mode: Continuous

  Underlying physical sensors: Accelerometer and (if present) gyroscope (or magnetometer if gyroscope not present)
  A linear acceleration sensor reports the linear acceleration of the device in the sensor frame, not including gravity. Readings on all axes should be close to 0 when the device is immobile.

  If the device possesses a gyroscope, the linear acceleration sensor must use the gyroscope and accelerometer as input.

  If the device doesn't possess a gyroscope, the linear acceleration sensor must use the accelerometer and the magnetometer as input

  The output is conceptually: output of the accelerometer minus the output of the gravity sensor. It's reported in m/s^2 in the x, y, and z fields.

- **Significant Motion:**

  Reporting-mode: One-shot

  A significant motion detector triggers when detecting a significant motion: a motion that might lead to a change in the user location.

  Examples of such significant motions are:

- Walking or biking
- Sitting in a moving car, coach, or train

  Examples of situations that don't trigger significant motion:

- Phone in pocket and person isn't moving
- Phone is on a table and the table shakes a bit due to nearby traffic or washing machine

At the high level, the significant motion detector is used to reduce the power consumption of location determination. When the localization algorithms detect that the device is static, they can switch to a low-power mode, where they rely on significant motion to wake the device up when the user is changing location.

This sensor must be low power. It makes a tradeoff for power consumption that may result in a small amount of false negatives. This is done for a few reasons:

The goal of this sensor is to save power.
Triggering an event when the user isn't moving (false positive) is costly in terms of power, so it should be avoided.
Not triggering an event when the user is moving (false negative) is acceptable as long as it isn't done repeatedly. If the user has been walking for 10 seconds, not triggering an event within those 10 seconds isn't acceptable.

- **Step Detector:**

  Reporting-mode: Special

  A step detector generates an event each time a step is taken by the user.
  The timestamp of the event timestamp corresponds to when the foot hit the ground, generating a high variation in acceleration.

  Compared to the step counter, the step detector should have a lower latency (less than two seconds). Both the step detector and the step counter detect when the user is walking, running, and walking up the stairs. They shouldn't trigger when the user is biking, driving, or in other vehicles.

  This sensor must be low power. That is, if the step detection cannot be done in hardware, this sensor shouldn't be defined. In particular, when the step detector is activated and the accelerometer isn't, only steps should trigger interrupts (not every accelerometer reading).Each sensor event reports 1.

- **Step Counter :**

  Reporting-mode: On-change
  A step counter reports the number of steps taken by the user since the last reboot while activated. The timestamp of the event is set to the time when the last step for that event was taken. Compared to the step detector, the step counter can have a higher latency (up to 10 seconds). Thanks to this latency, this sensor has a high accuracy; the step count after a full day of measures should be within 10% of the actual step count. Both the step detector and the step counter detect when the user is walking, running, and walking up the stairs. They shouldn't trigger when the user is biking, driving, or in other vehicles.

The hardware must ensure the internal step count never overflows. The minimum size of the hardware's internal counter shall be 16 bits. In case of imminent overflow (at most every ~2^16 steps), the SoC can be woken up so the driver can do the counter maintenance. As stated in Interaction, while this sensor operates, it shall not disrupt any other sensors, in particular, the accelerometer, which might very well be in use.

If a particular device can't support these modes of operation, then this sensor type must not be reported by the HAL. That is, it isn't acceptable to "emulate" this sensor in the HAL.

This sensor must be low power. That is, if the step detection can't be done in hardware, this sensor shouldn't be defined. In particular, when the step counter is activated and the accelerometer isn't, only steps should trigger interrupts (not accelerometer data).

- **Tilt Detector:**

  Reporting-mode: Special

  A tilt detector generates an event each time a tilt event is detected.

  A tilt event is defined by the direction of the 2-seconds window average gravity changing by at least 35 degrees since the activation or the last event generated by the sensor.

  Large accelerations without a change in phone orientation shouldn't trigger a tilt event. For example, a sharp turn or strong acceleration while driving a car shouldn't trigger a tilt event, even though the angle of the average acceleration might vary by more than 35 degrees. Typically, this sensor is implemented with the help of only an accelerometer. Other sensors can be used as well if they do not increase the power consumption significantly. This is a low-power sensor that should allow the SoC to go into suspend mode. Do not emulate this sensor in the HAL. Each sensor event reportTrigger when angle(reference_estimated_gravity, current_estimated_gravity) > 35 degreest

- **Reference_estimated_gravity** = average of accelerometer measurements over the first second after activation or the estimated gravity when the last tilt event was generated
- **current_estimated_gravity** = average of accelerometer measurements over the last 2 seconds.

- **Rotation vector**

Underlying physical sensors: Accelerometer, magnetometer, and gyroscope

Reporting-mode: Continuous

A rotation vector sensor reports the orientation of the device relative to the East-North-Up coordinates frame. It's usually obtained by integration of accelerometer, gyroscope, and magnetometer readings. The East-North-Up coordinate system is defined as a direct orthonormal basis where:

- X points east and is tangential to the ground.
- Y points north and is tangential to the ground.
- Z points towards the sky and is perpendicular to the ground.

The orientation of the phone is represented by the rotation necessary to align the East-North-Up coordinates with the phone's coordinates. That is, applying the rotation to the world frame (X,Y,Z) would align them with the phone coordinates (x,y,z).
The rotation can be seen as rotating the phone by an angle theta around an axis rot_axis to go from the reference (East-North-Up aligned) device orientation to the current device orientation.

The rotation is encoded as the four unit-less x, y, z, w components of a unit quaternion:

- **sensors_event_t.data =** rot_axis.x*sin(theta/2)
- **sensors_event_t.data** = rot_axis.y*sin(theta/2**)**
- **sensors_event_t.data =** rot_axis.z*sin(theta/2)
- **sensors_event_t.data =** cos(theta/2)

Where. The x, y, and z fields of rot_axis are the East-North-Up coordinates of a unit length vector representing the rotation axis theta is the rotation angle

The quaternion is a unit quaternion: It must be of norm 1. Failure to ensure this will cause erratic client behavior.
In addition, this sensor reports an estimated heading accuracy:

The heading error must be less than estimated_accuracy 95% of the time. This sensor must use a gyroscope as the main orientation change input.
This sensor also uses accelerometer and magnetometer input to make up for gyroscope drift, and it can't be implemented using only the accelerometer and magnetometer.

- **Game rotation vector**

  Underlying physical sensors: Accelerometer and gyroscope (no magnetometer)

  Reporting-mode: Continuous

  A game rotation vector sensor is similar to a rotation vector sensor but not using the geomagnetic field. Therefore the Y axis doesn't point north but instead to some other reference. That reference is allowed to drift by the same order of magnitude as the gyroscope drifts around the Z axis.

  In an ideal case, a phone rotated and returned to the same real-world orientation should report the same game rotation vector.
  This sensor must be based on a gyroscope and an accelerometer. It can't use magnetometer as an input, besides, indirectly, through estimation of the gyroscope bias.

- **Gravity**

  Underlying physical sensors: Accelerometer and (if present) gyroscope (or magnetometer if gyroscope not present)

  Reporting-mode: Continuous

  A gravity sensor reports the direction and magnitude of gravity in the device's coordinates.
  The gravity vector components are reported in m/s^2 in the x, y, and z fields

  When the device is at rest, the output of the gravity sensor should be identical to that of the accelerometer.
  On Earth, the magnitude is around 9.8 m/s^2.

  If the device possesses a gyroscope, the gravity sensor must use the gyroscope and accelerometer as input.
  If the device doesn’t possess a gyroscope, the gravity sensor must use the accelerometer and the magnetometer as input.

- **Geomagnetic rotation vector**

  Underlying physical sensors: Accelerometer and magnetometer (no gyroscope)

  Reporting-mode: Continuous

  A geomagnetic rotation vector is similar to a rotation vector sensor but using a magnetometer and no gyroscope.

  This sensor must be based on a magnetometer. It can't be implemented using a gyroscope, and gyroscope input can't be used by this sensor.
  Just like for the rotation vector sensor, the heading error must be less than the estimated accuracy 95% of the time.
  This sensor must be low power, so it has to be implemented in hardware.

- **Orientation (deprecated)**

  Underlying physical sensors: Accelerometer, magnetometer and (if present) gyroscope

  Reporting-mode: Continuous

  This is an older sensor type that has been deprecated in the Android SDK. It has been replaced by the rotation vector sensor, which is more clearly defined. Use the rotation vector sensor over the orientation sensor whenever possible.
  An orientation sensor reports the attitude of the device. The measurements are reported in degrees in the x, y, and z fields of sensors event orientation:
  • **Sensors_event_t.orientation.x:**
  Azimuth, the angle between the magnetic north direction and the Y axis, around the Z axis (0<=azimuth<360). 0=North, 90=East, 180=South, 270=West.

  •**Sensors_event_t.orientation.y:**
  pitch, rotation around X axis (-180<=pitch<=180), with positive values when the Z axis moves toward the Y axis.

  •**Sensors_event_t.orientation.z**: roll, rotation around Y axis (-90<=roll<=90), with positive values when the X axis moves towards the Z axis.

  This definition is different from yaw, pitch, and roll used in aviation where the X axis is along the long side of the plane (tail to nose).
  The orientation sensor also reports how accurate it expects its readings to be through sensors_event_t.orientation.status

- **Uncalibrated sensors**

  Uncalibrated sensors:

  Provide more raw results and may include some bias but also contain fewer "jumps" from corrections applied through calibration.
  uncalibrated results as smoother and more reliable. For instance, if an app is attempting to conduct its own sensor fusion, introducing calibrations can actually distort results.

- **Accelerometer uncalibrated:**

  Underlying physical sensor: Accelerometer

  Reporting-mode: Continuous

  An uncalibrated accelerometer sensor reports the acceleration of the device along the three sensor axes without any bias correction (factory bias and temperature compensation are applied to uncalibrated measurements), along with a bias estimate.

  All values are in SI units (m/s^2) and are reported in the fields of sensors_event_t.uncalibrated_accelerometer:

  •**X_uncalib**: Acceleration (without bias compensation) along the X axis

  •**Y_uncalib**: Aceleration (without bias compensation) along the Y axis

  •**Z_uncalib**: Acceleration (without bias compensation) along the Z axis

  •**X_bias**: Estimated bias along X axis

  •**Y_bias**: Estimated bias along Y axis

  •**Z_bias**: Estimated bias along Z axis

- **Gyroscope uncalibrated**

Underlying physical sensor: Gyroscope

Reporting-mode: Continuous

An uncalibrated gyroscope reports the rate of rotation around the sensor axes without applying bias compensation to them, along with a bias estimate. All values are in radians/second and are reported in the fields of sensors_event_t.uncalibrated_gyro:

• **x_uncalib**: angular speed (without drift compensation) around the X axis

• **y_uncalib**: angular speed (without drift compensation) around the Y axis

• **z_uncalib**: angular speed (without drift compensation) around the Z axis

• **x_bias**: estimated drift around X axis

• **y_bias**: estimated drift around Y axis

• **z_bias**: estimated drift around Z axis

Conceptually, the uncalibrated measurement is the sum of the calibrated measurement and the bias estimate: _uncalibrated = _calibrated + _bias.

The x_bias, y_bias and z_bias values are expected to jump as soon as the estimate of the bias changes, and they should be stable the rest of the time.

Factory calibration and temperature compensation must be applied to the measurements. Also, gyroscope drift estimation must be implemented so that reasonable estimates can be reported in x_bias, y_bias and z_bias. If the implementation isn't able to estimate the drift, then this sensor must not be implemented.

If this sensor is present, then the corresponding Gyroscope sensor must also be present and both sensors must share the same sensor_t.name and sensor_t.vendor values.

- **Magnetic field uncalibrated**

  Underlying physical sensor: Magnetometer

  Reporting-mode: Continuous

  An uncalibrated magnetic field sensor reports the ambient magnetic field together with a hard iron calibration estimate. All values are in micro-Tesla (uT) and are reported in the fields of sensors_event_t.uncalibrated_magnetic:
  • **x_uncalib**: Magnetic field (without hard-iron compensation) along the X axis
  • **y_uncalib**: Magnetic field (without hard-iron compensation) along the Y axis
  • **z_uncalib**: Magnetic field (without hard-iron compensation) along the Z axis
  • **x_bias**: Estimated hard-iron bias along the X axis
  • **y_bias**: Estimated hard-iron bias along the Y axis
  • **z_bias**: Estimated hard-iron bias along the Z axis

  Conceptually, the uncalibrated measurement is the sum of the calibrated measurement and the bias estimate: _uncalibrated = _calibrated + _bias.
  The uncalibrated magnetometer allows higher level algorithms to handle bad hard iron estimation.

  The x_bias, y_bias and z_bias values are expected to jump as soon as the estimate of the hard-iron changes, and they should be stable the rest of the time.
  Soft-iron calibration and temperature compensation must be applied to the measurements. Also, hard-iron estimation must be implemented so that reasonable estimates can be reported in x_bias, y_bias and z_bias. If the implementation isn't able to estimate the bias, then this sensor must not be implemented.

  If this sensor is present, then the corresponding magnetic field sensor must be present and both sensors must share the same sensor_t.name and sensor_t.vendor values.

- **Hinge angle:**

  Reporting-mode: On-change

  A hinge angle sensor measures the angle, in degrees, between two integral parts of the device. Movement of a hinge measured by this sensor type is expected to alter the ways in which the user can interact with the device, for example, by unfolding or revealing a display.

  Some sensors are mostly used to detect interactions with the user. We don't define how those sensors must be implemented, but they must be low power and it's the responsibility of the device manufacturer to verify their quality in terms of user experience.

- **Wake up gesture:**

  Underlying physical sensors: Undefined (anything low power)

  Reporting-mode: One-shot

  Low-power
  Implement only the wake-up version of this sensor.

  A wake up gesture sensor enables waking up the device based on a device specific motion. When this sensor triggers, the device behaves as if the power button was pressed, turning the screen on. This behavior (turning on the screen when this sensor triggers) might be deactivated by the user in the device settings. Changes in settings don't impact the behavior of the sensor: only whether the framework turns the screen on when it triggers. The actual gesture to be detected isn't specified, and can be chosen by the manufacturer of the device.
  This sensor must be low power, as it's likely to be activated 24/7.
  Each sensor event reports 1

- **Pick up gesture:**

  Underlying physical sensors: Undefined (anything low power)

  Reporting-mode: One-shot

  Low-power
  Implement only the wake-up version of this sensor.

  A pick-up gesture sensor triggers when the device is picked up regardless of wherever it was before (desk, pocket, bag).
  Each sensor event reports 1 in sensors_event_t.data

- **Glance gesture:**

  Underlying physical sensors: Undefined (anything low power)

  Reporting-mode: One-shot

  Implement only the wake-up version of this sensor.
  A glance gesture sensor enables briefly turning the screen on to enable the user to glance content on screen based on a specific motion. When this sensor triggers, the device will turn the screen on momentarily to allow the user to glance notifications or other content while the device remains locked in a non-interactive state (dozing), then the screen will turn off again. This behavior (briefly turning on the screen when this sensor triggers) might be deactivated by the user in the device settings. Changes in settings do not impact the behavior of the sensor: only whether the framework briefly turns the screen on when it triggers. The actual gesture to be detected isn't specified, and can be chosen by the manufacturer of the device.

Chapter 6

Output Images And Analysis

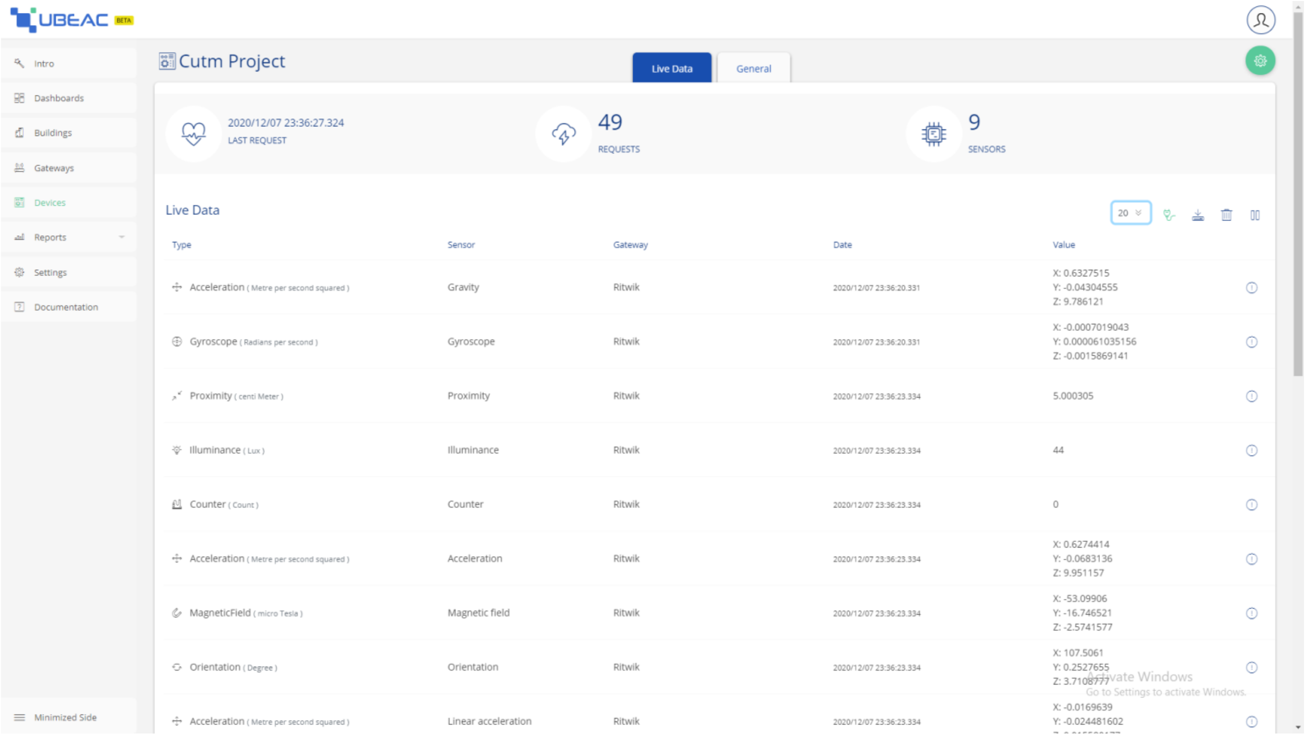


Figure:1(Device Report 1)

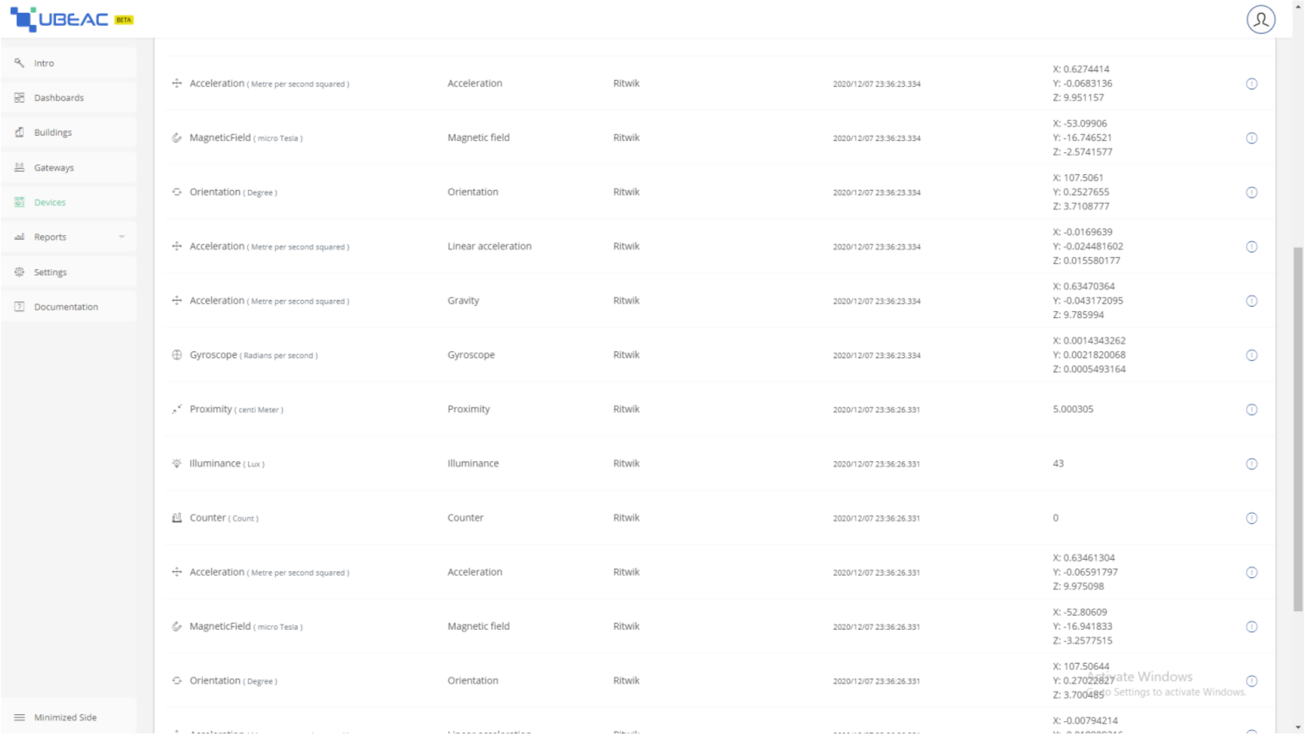


Figure:2 (Device Report 2)

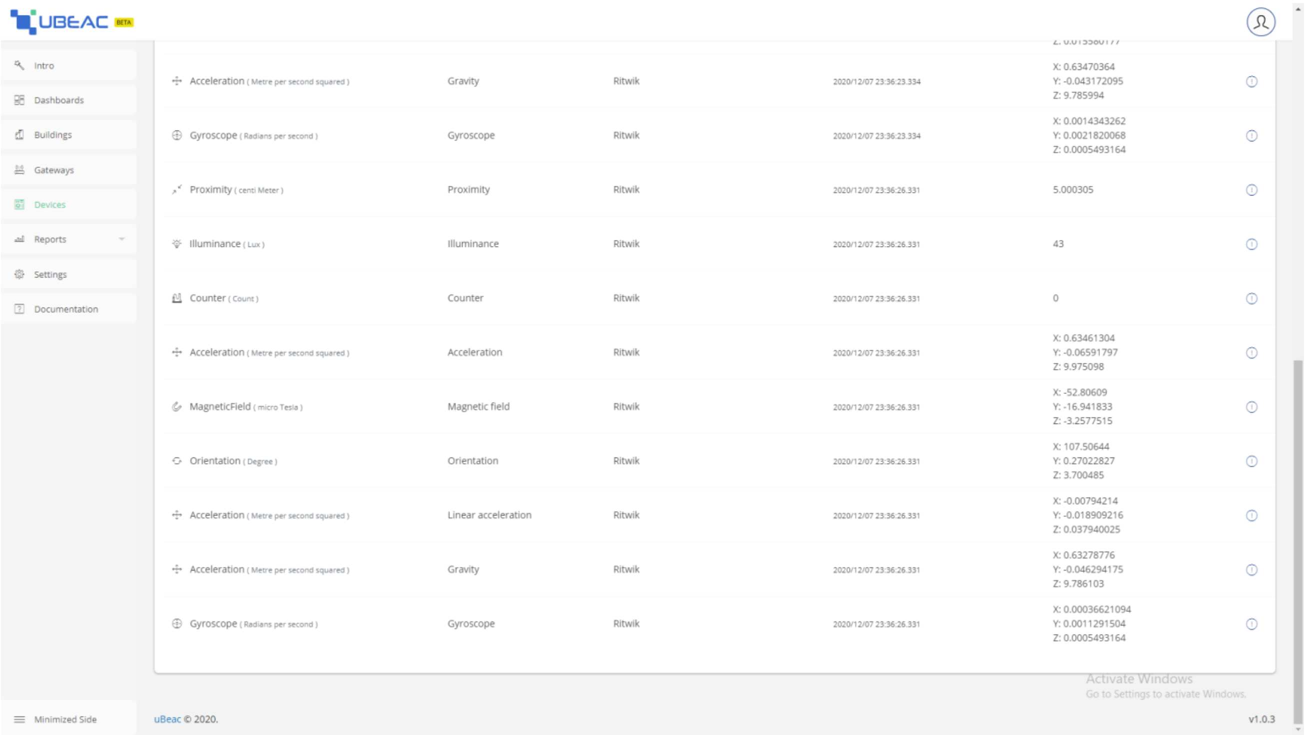


Figure:3(Device Report 3)

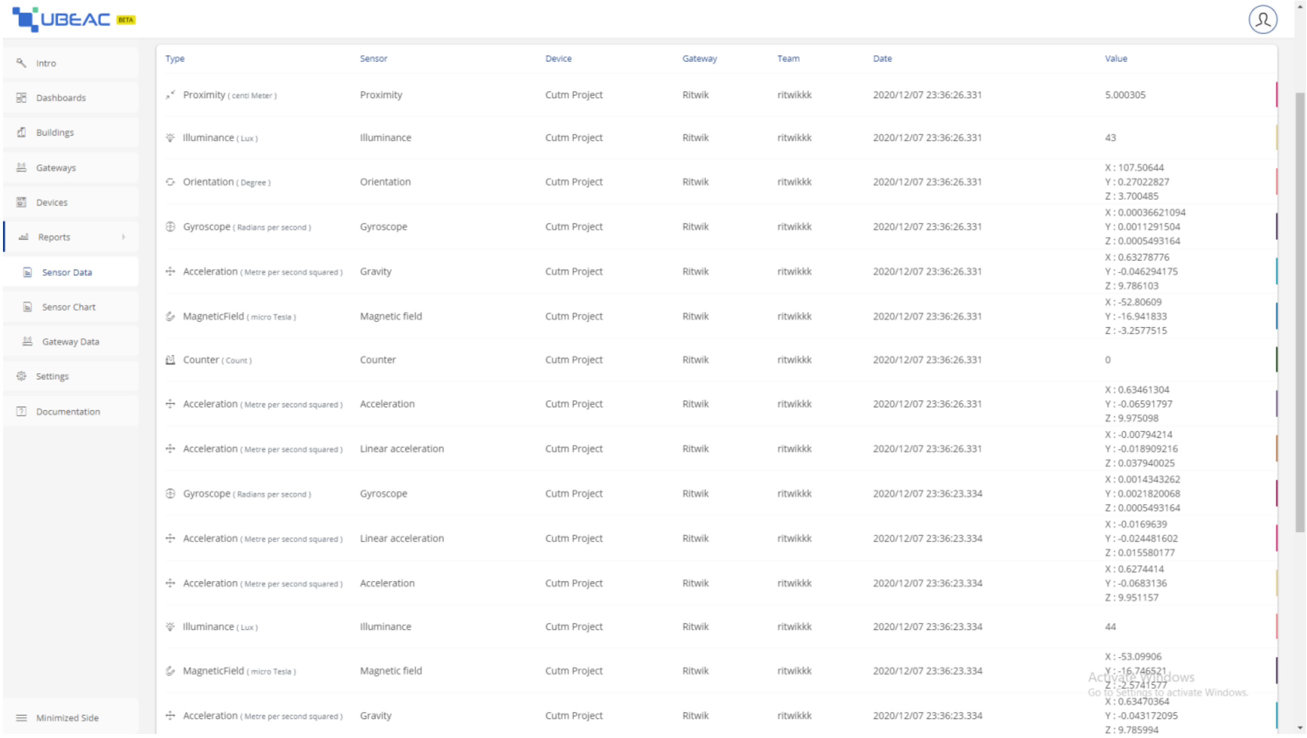


Figure:4(Device Report 4)

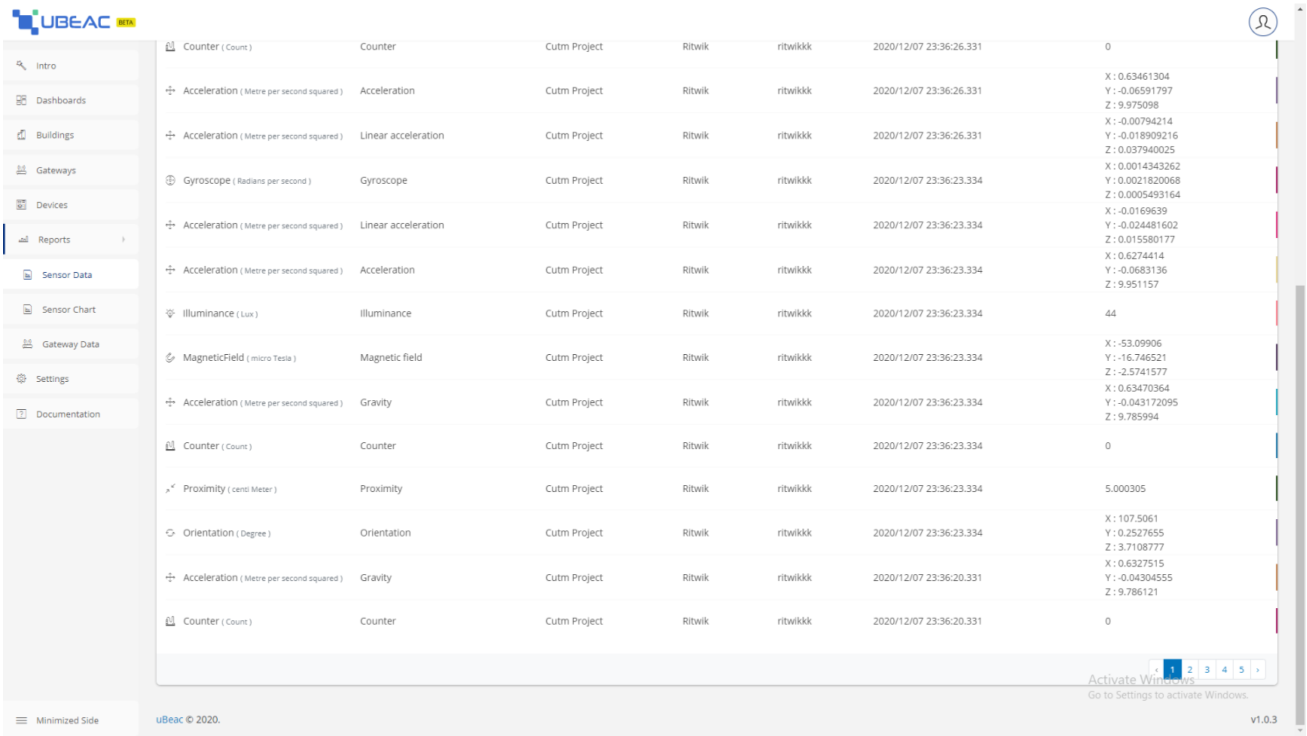


Figure6(Device Report 6)

Figur

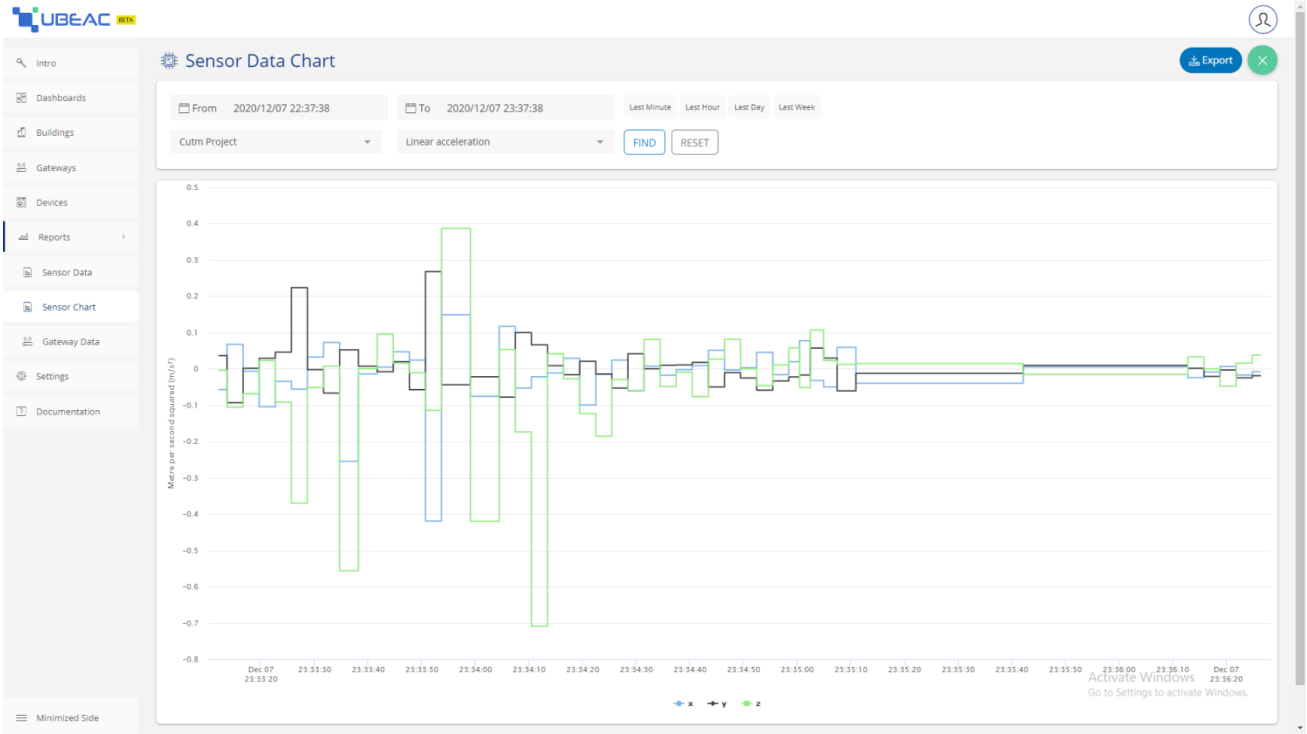


Figure:5(Device Report 5)

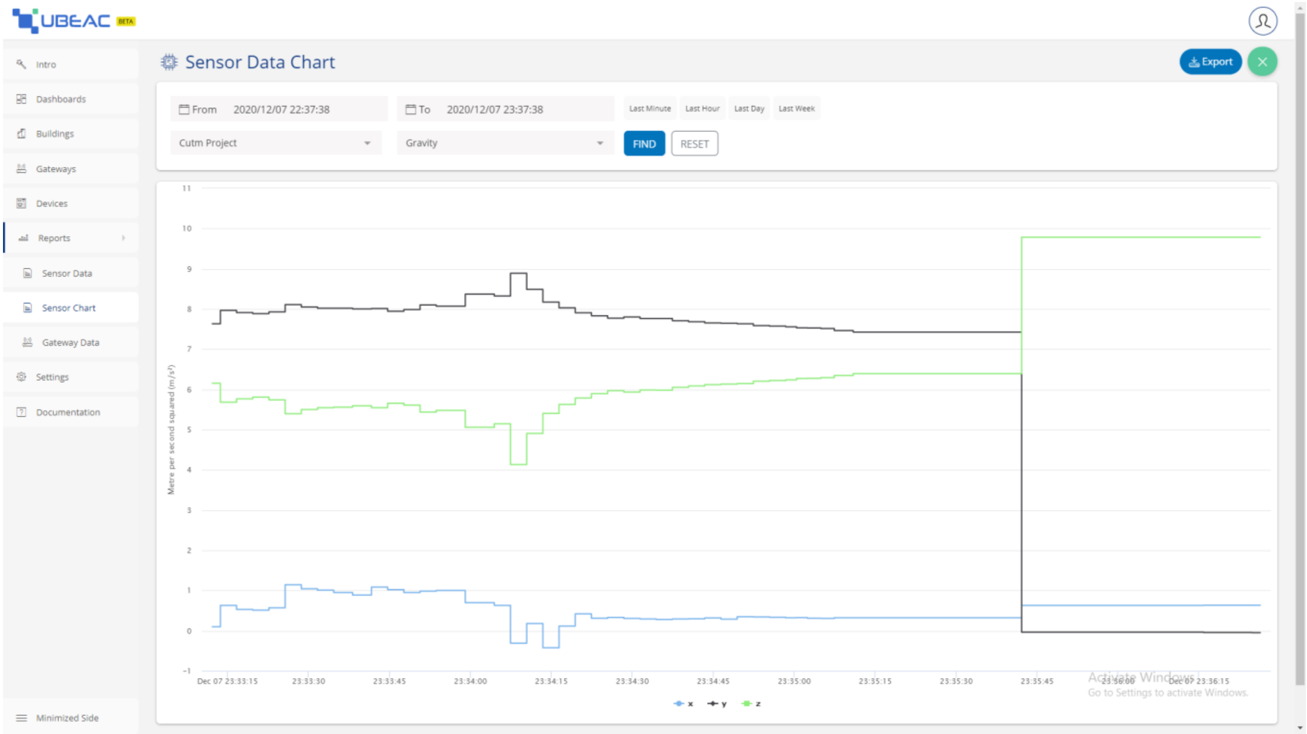


Figure7:(Device Report 7)

Figure 8(Acceleration)

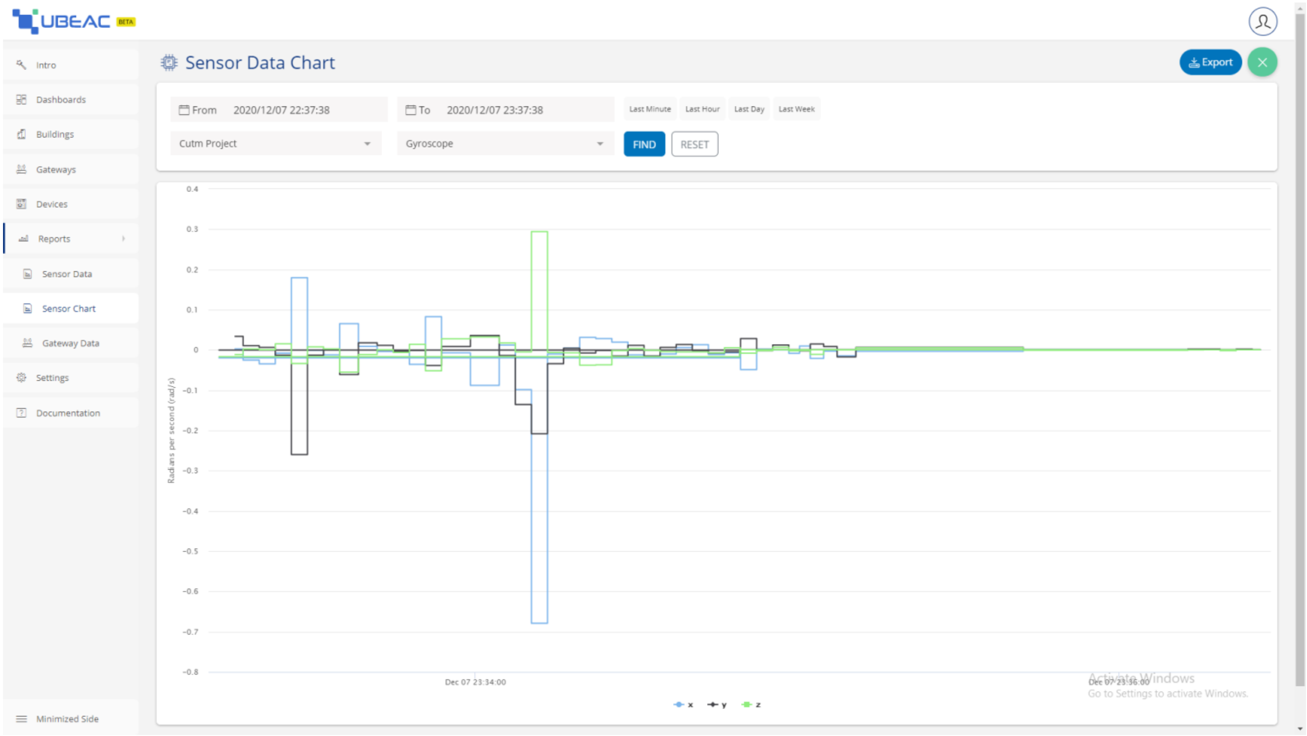


Figure 9(Gyroscope)

fbdb

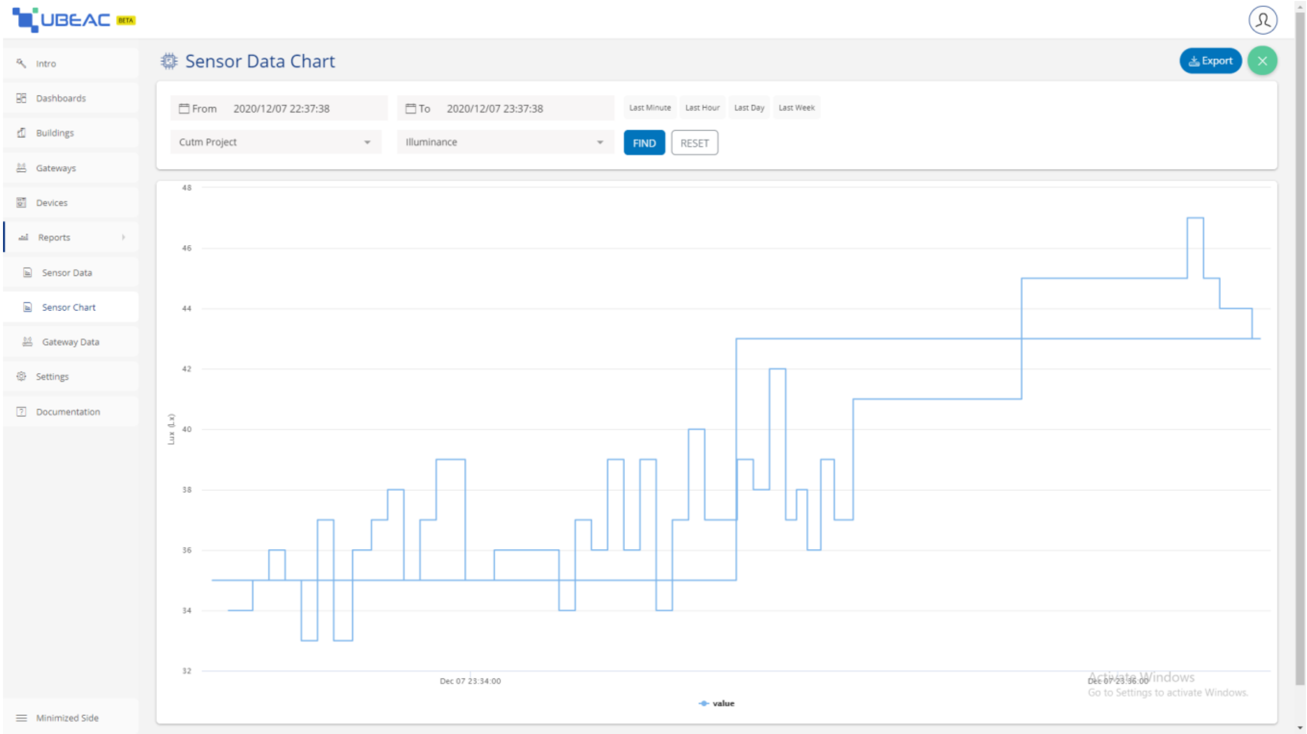


Figure 10:(Illuminace)

Figure 11 (Magnectic Field)

Figure 12 : (Orientation)

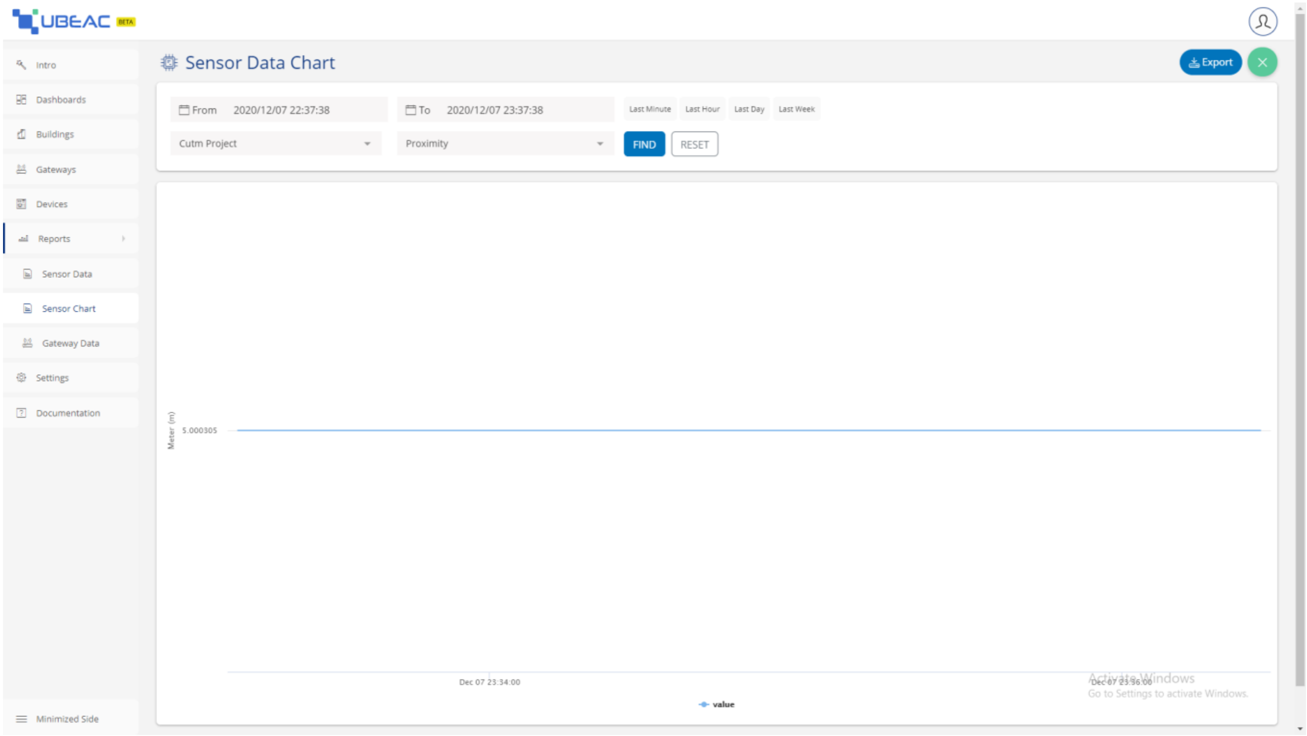


Figure 13 : (Proximity)

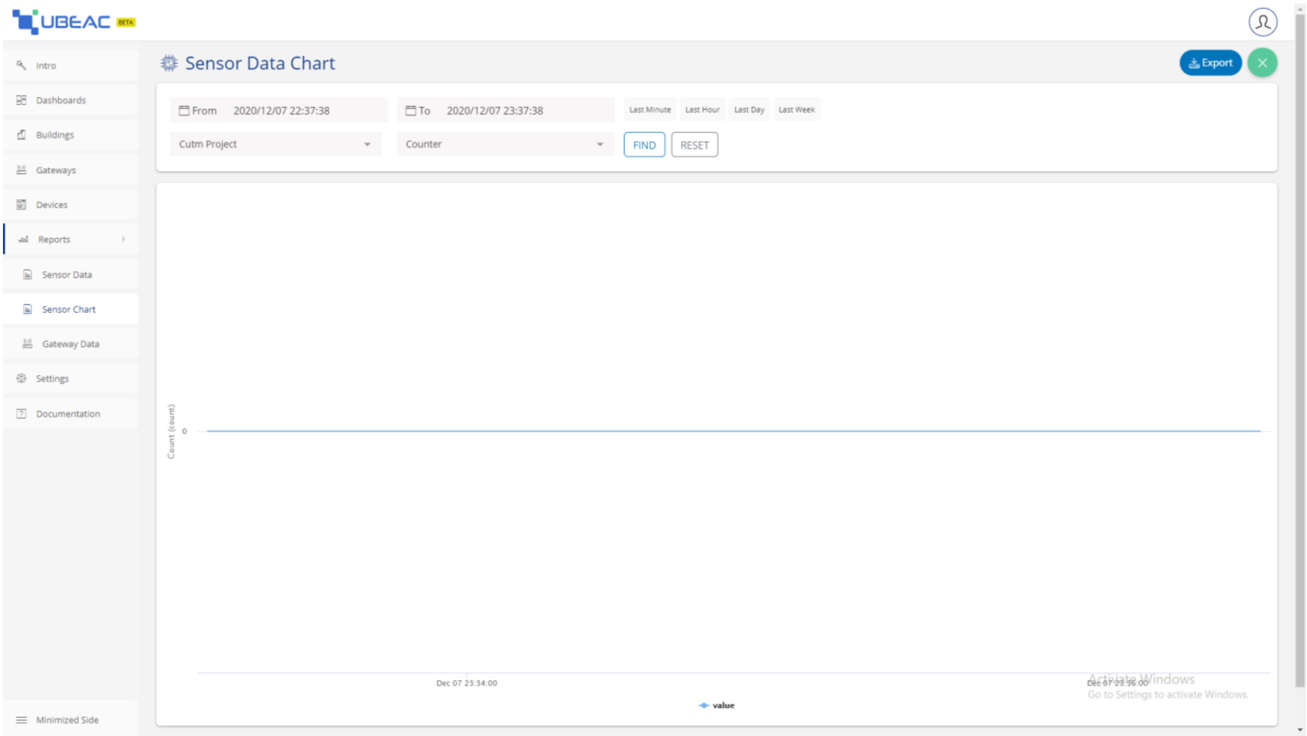
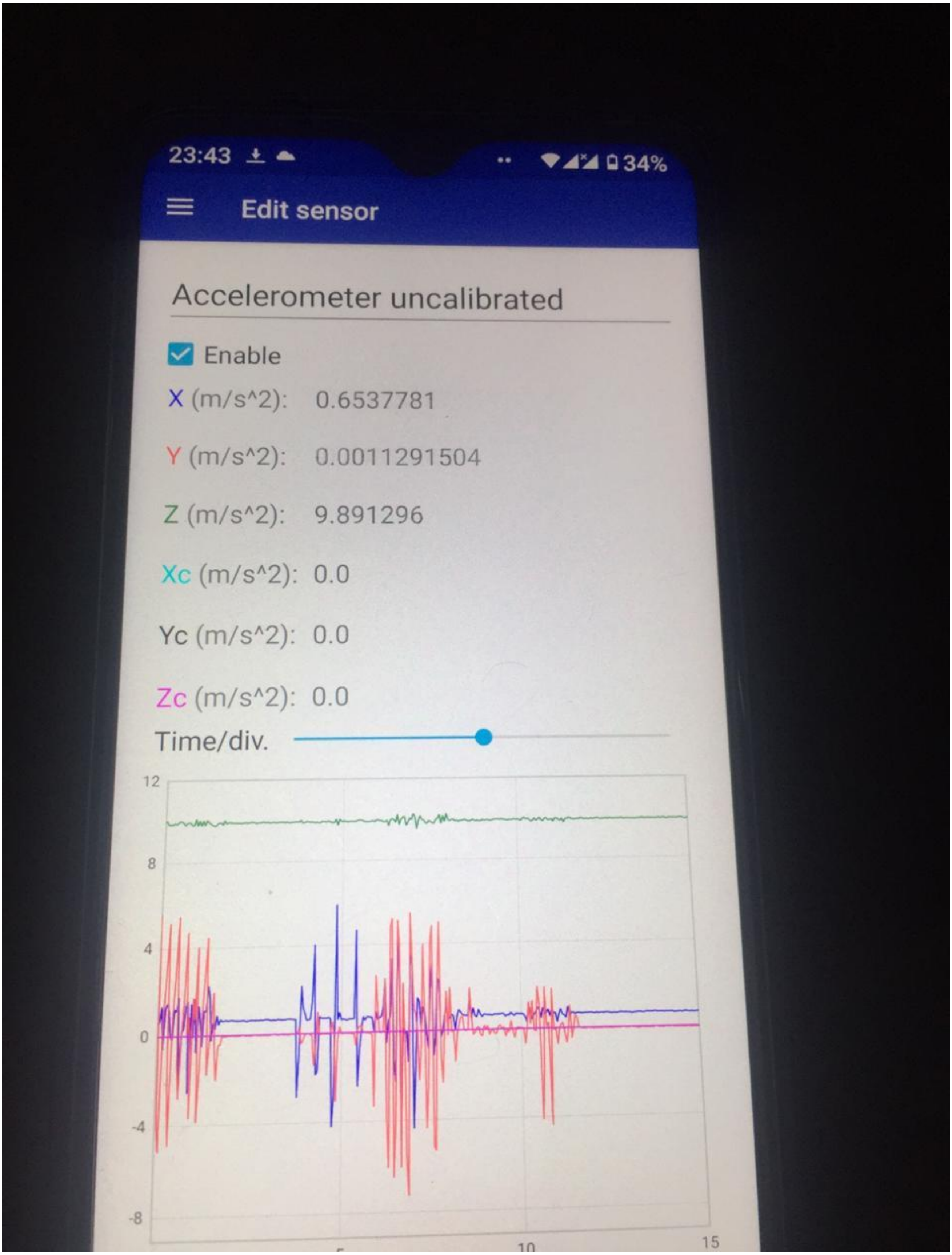
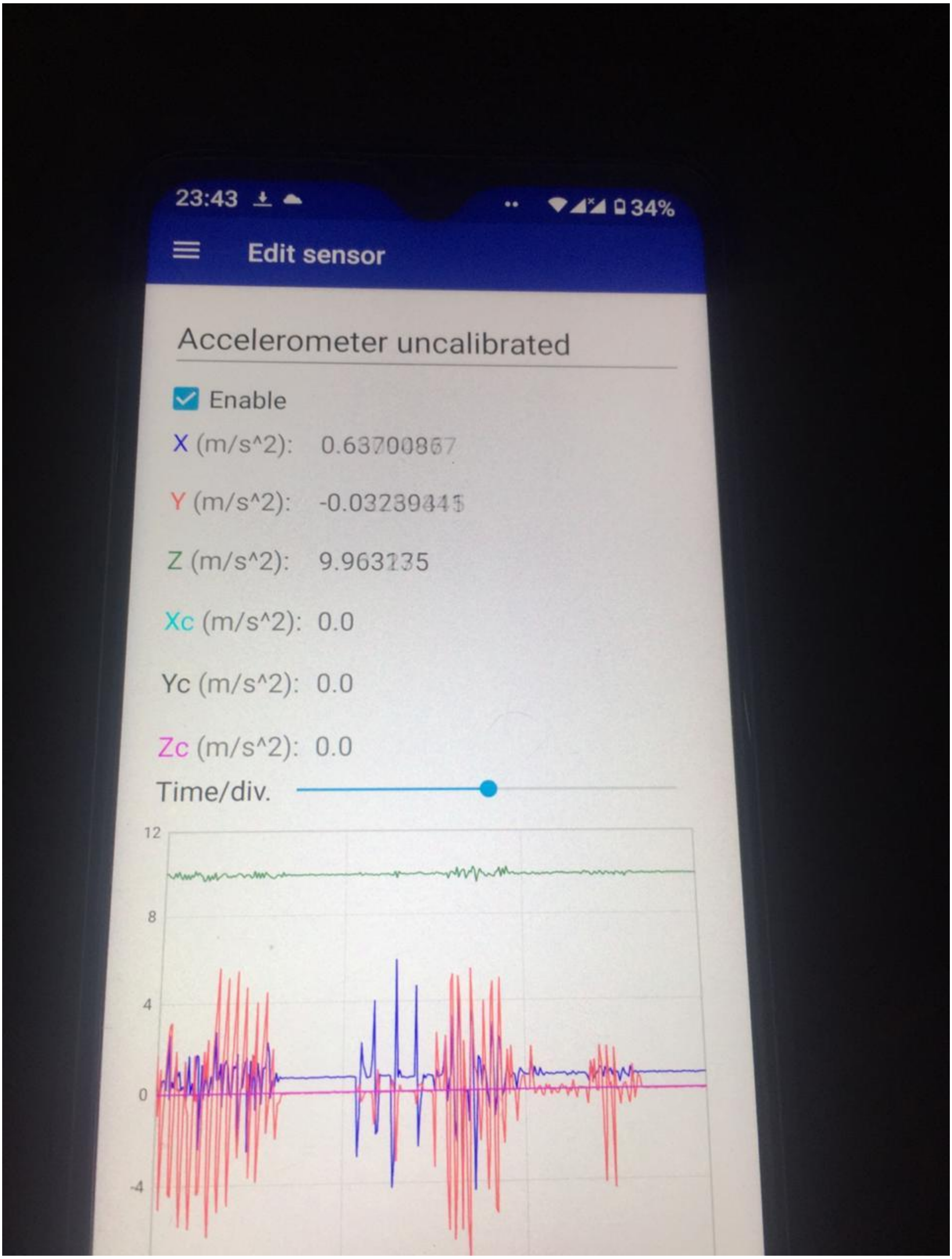


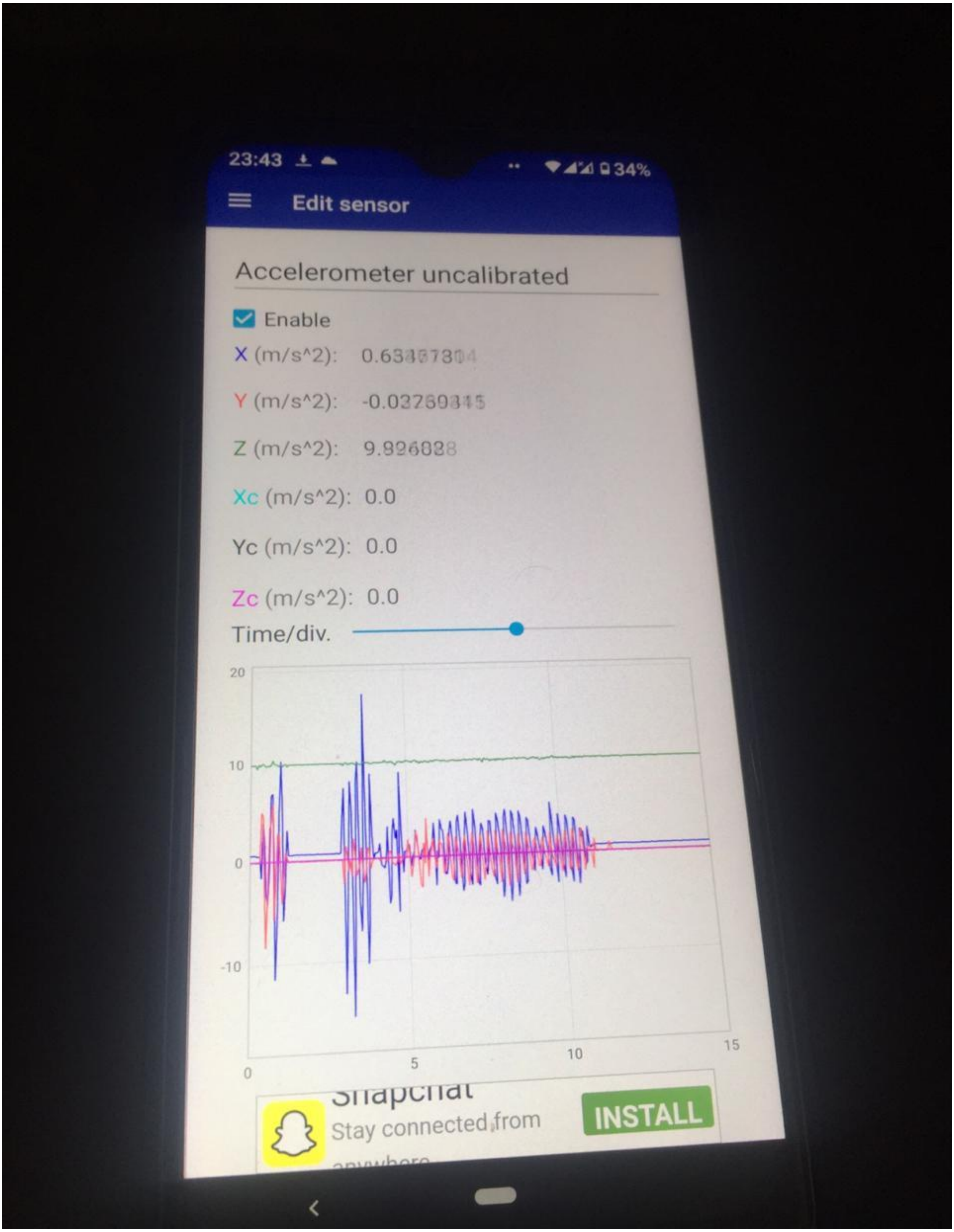
Figure 14 : (Step Counter)



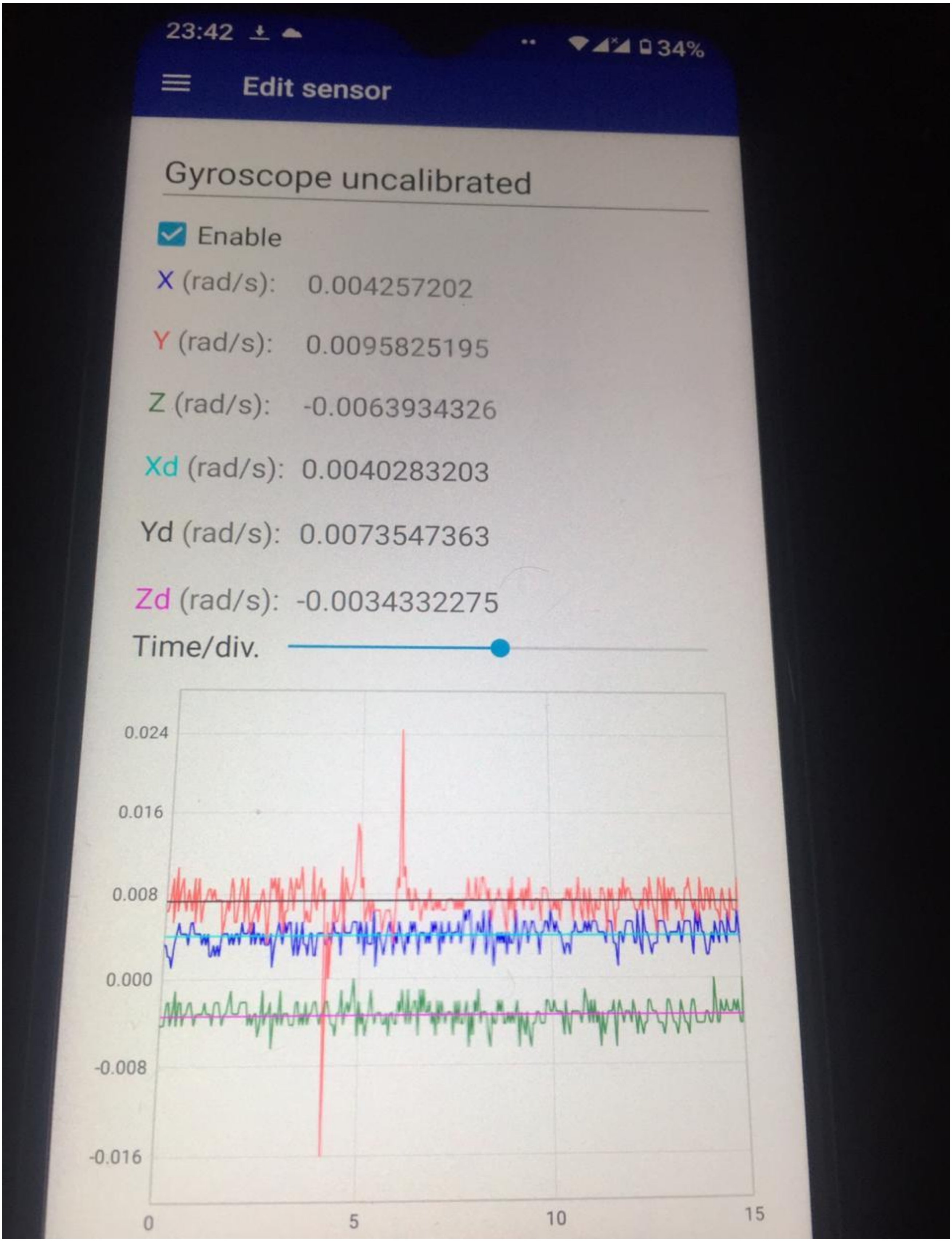
Accelerometer Uncalibrated



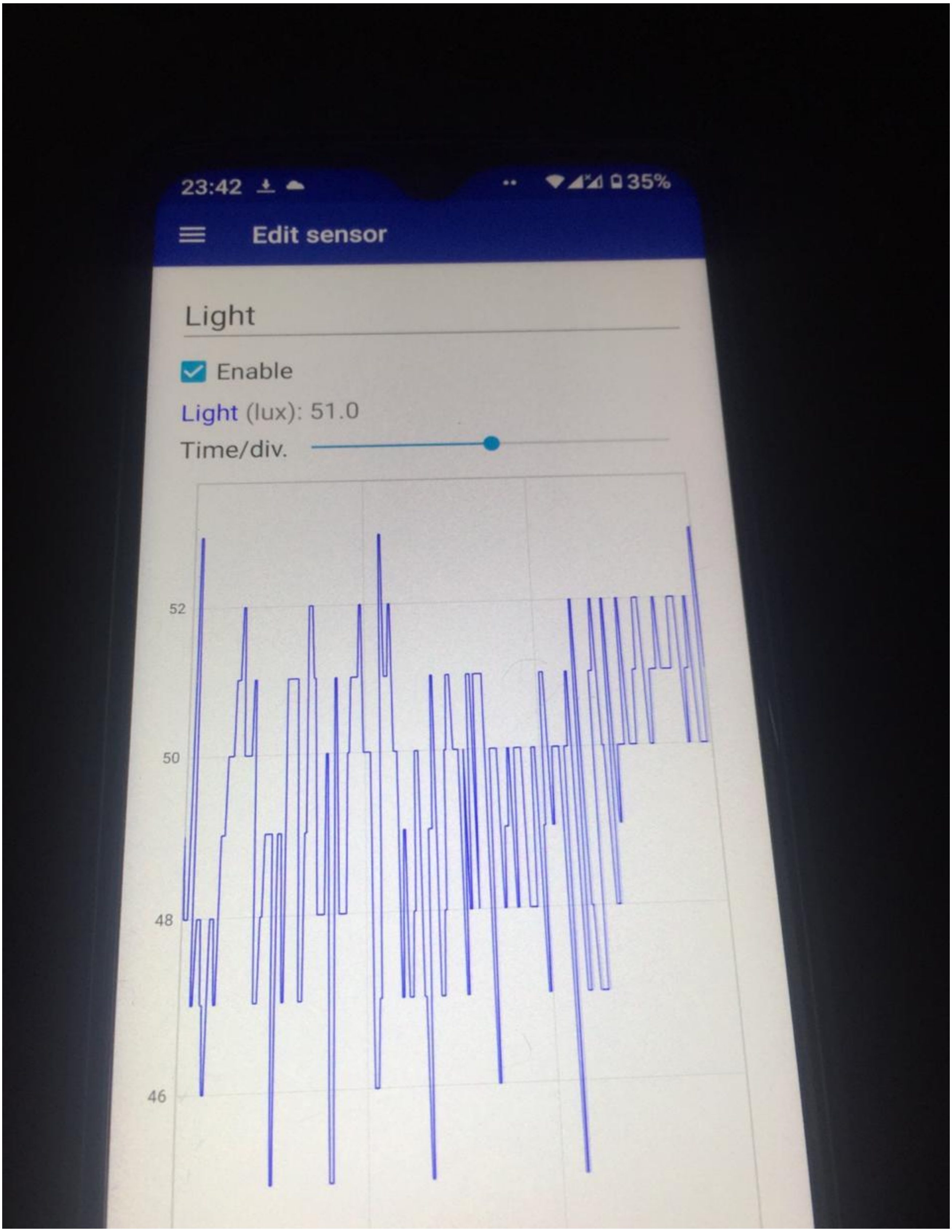
Accelerometer Uncalibrated After 20 Sec



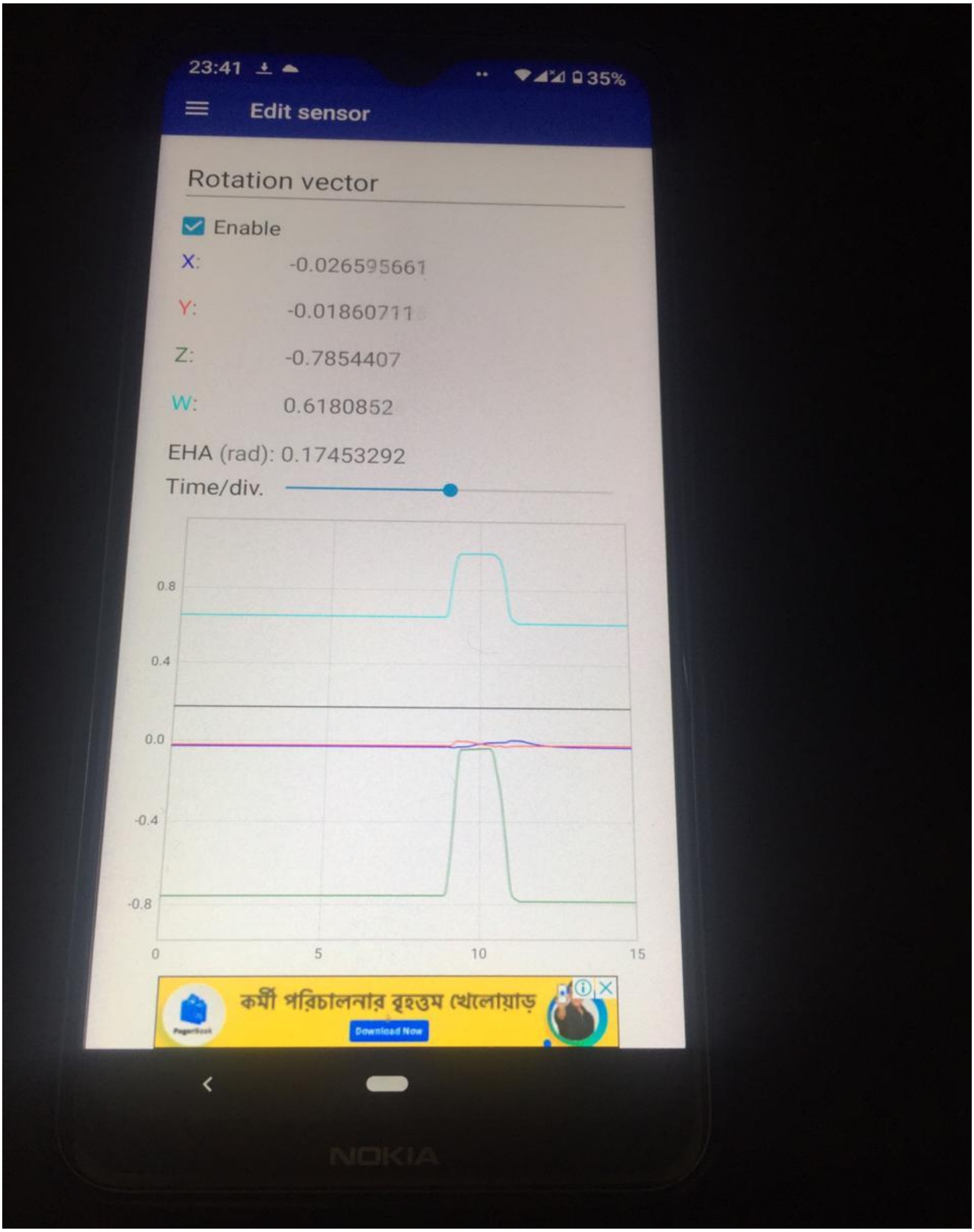
Accelerometer Uncalibrated After 20 Sec



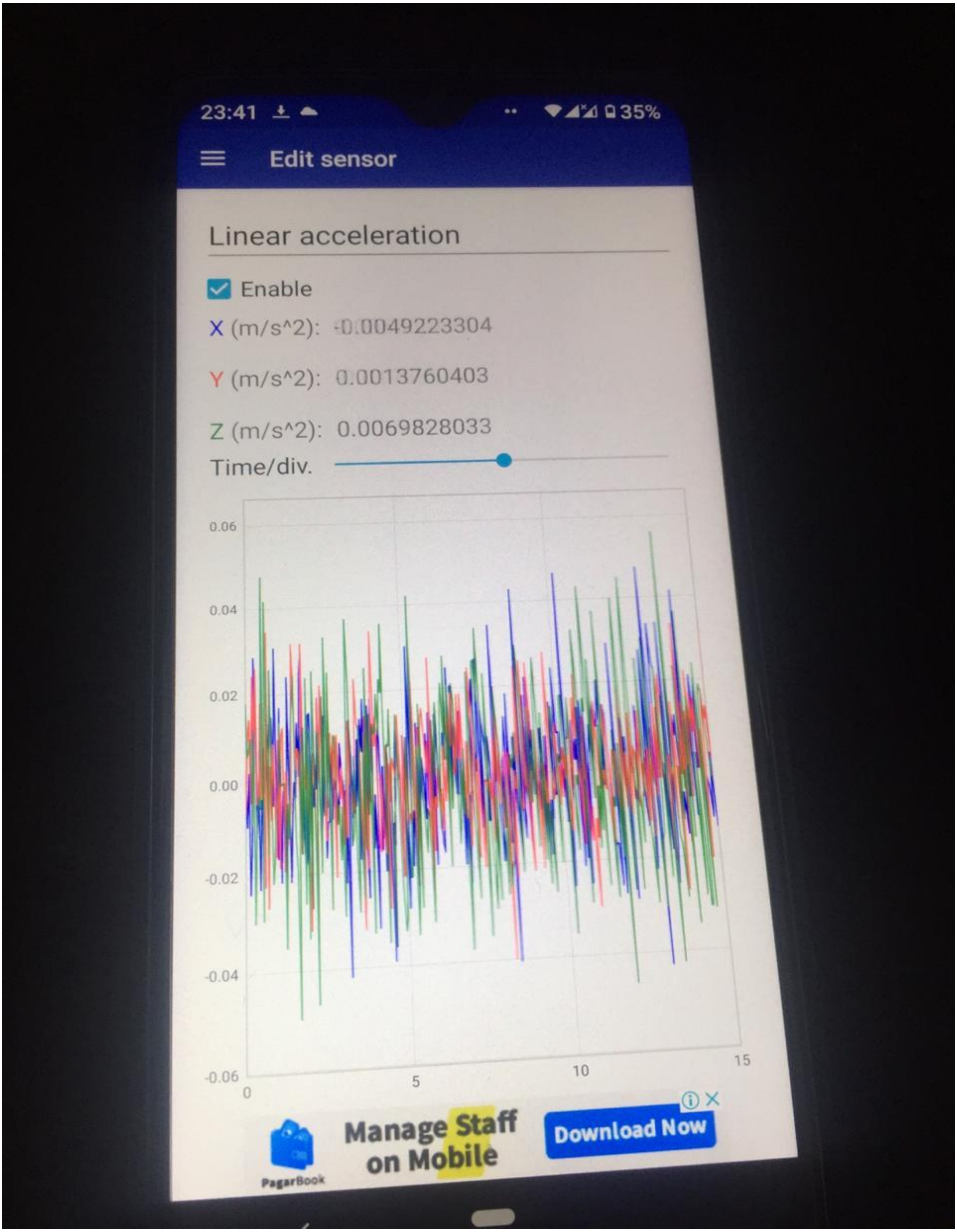
Gyroscope Uncalibrated



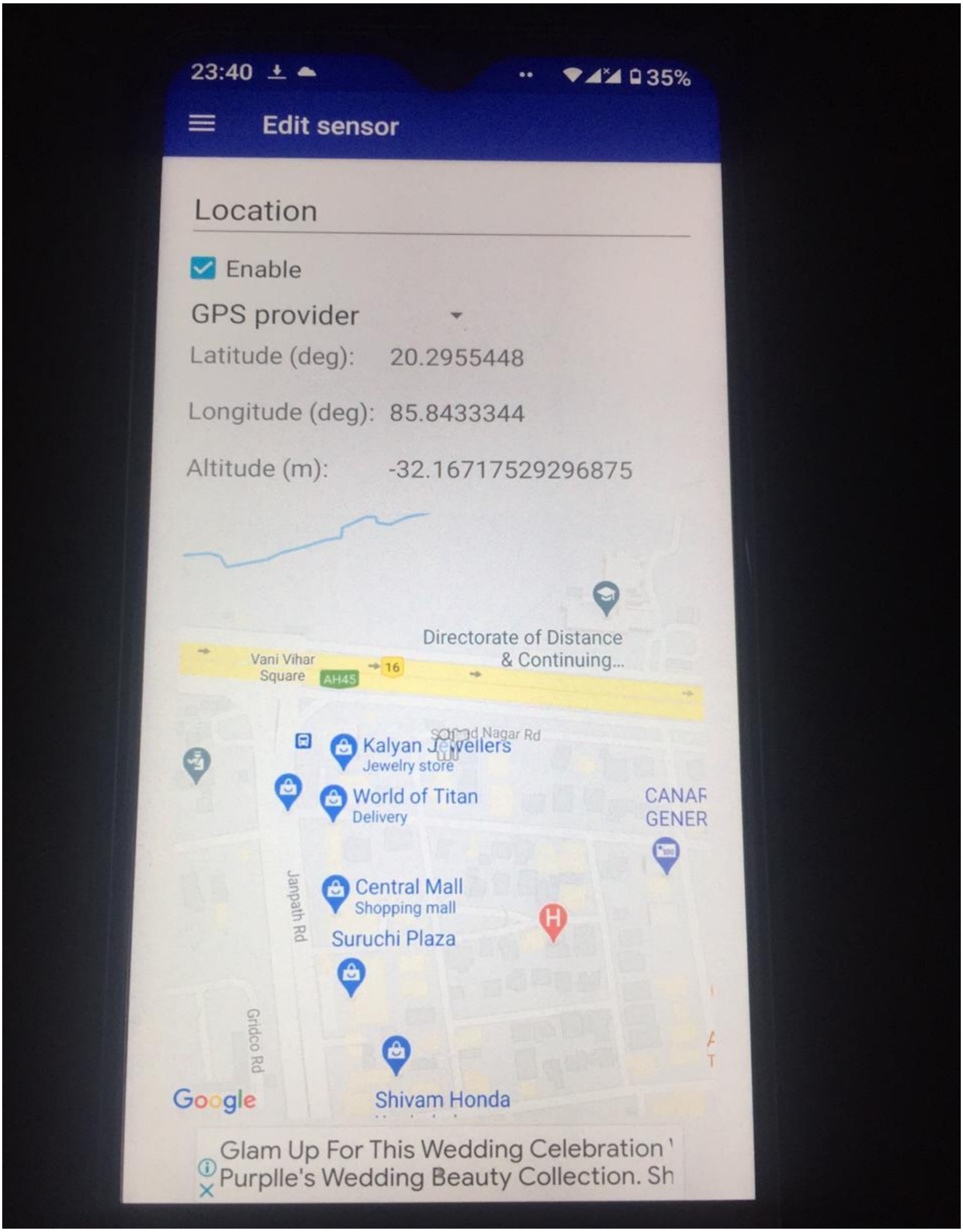
Light or Illuminace



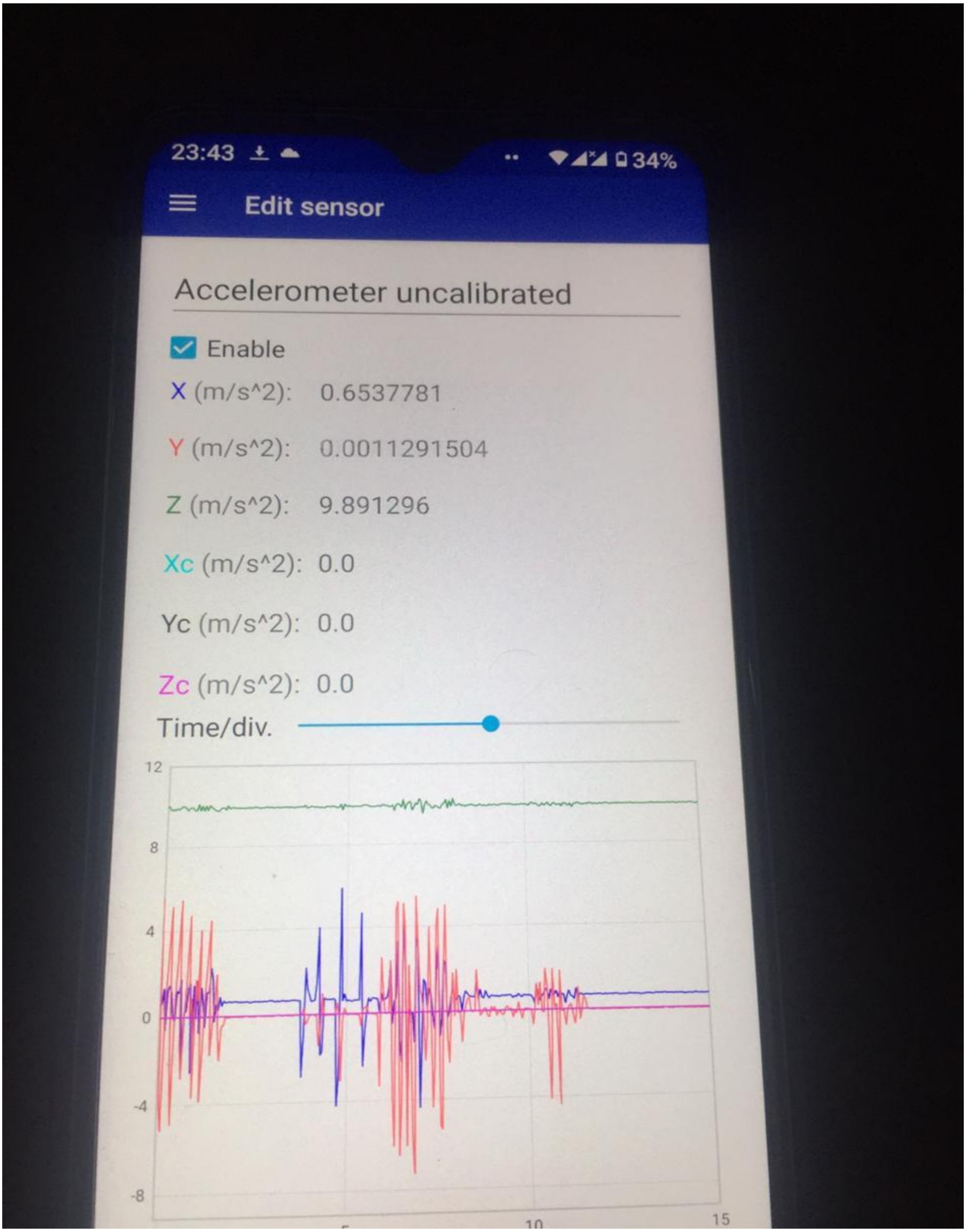
Rotation Vector



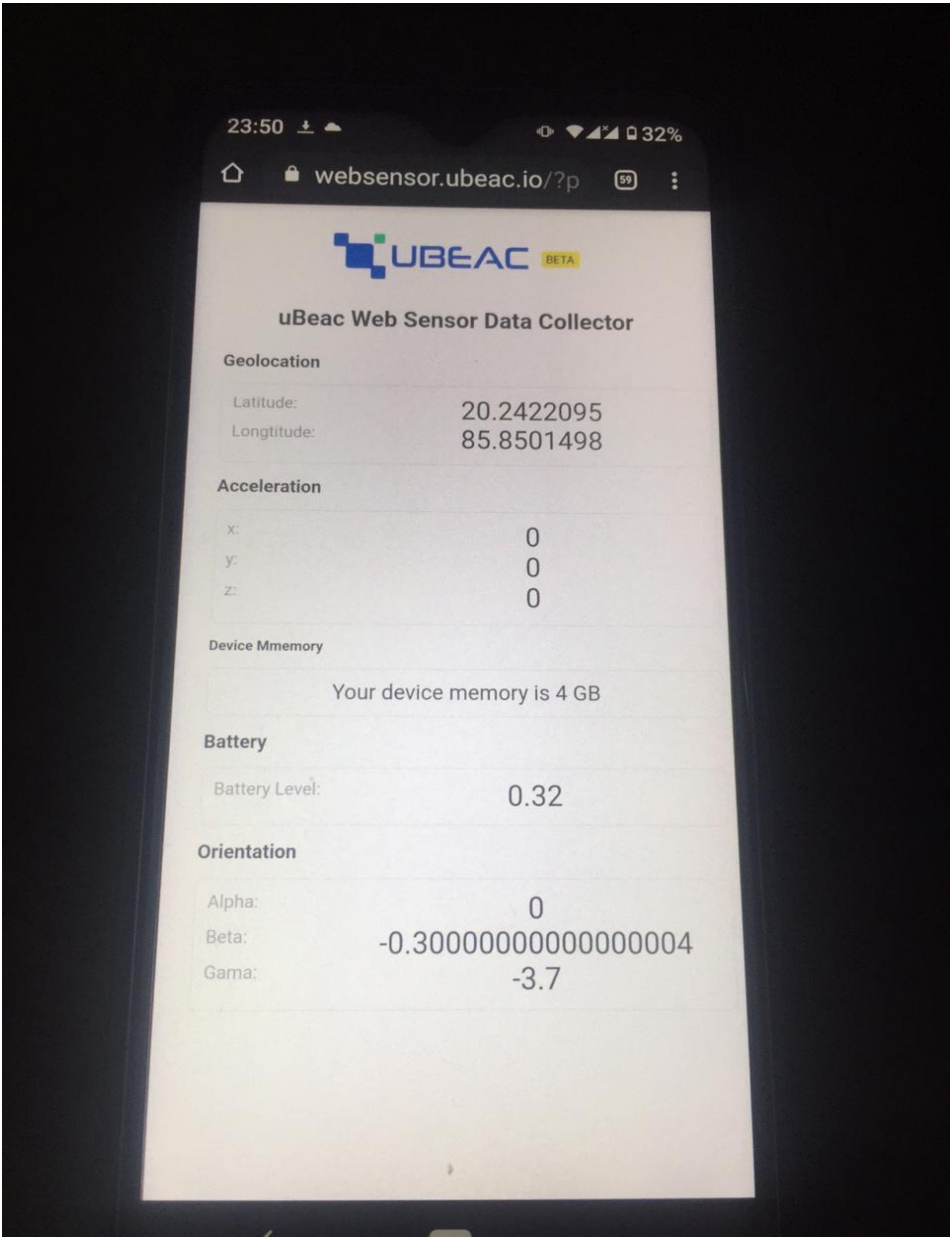
Linear Acceleration



GPS(location Sensor)



Accelrometer Uncalibrated after 20 Sec



Live Data of Your Sensor present in your Phone through Web site

Conclusion

- This is how you can use uBeac to create a dashboard from your phone's sensors. Now you can use it for numerous projects, such as keeping track of where and how you have traveled to and from work, and what you can do to optimize your trip.

References

- https://app.ubeac.io/gatewaydata
- https://www.hackster.io/amir-pournasserian/iot-dashboards-for-sensors-on-android-device-using-ubeac-b207a5
- https://app.ubeac.io/devices/new
- https://play.google.com/store/apps/details?id=tech.unismart.dc&hl=en&gl=US
- https://source.android.com/devices/sensors/report-modes#on-change
- https://source.android.com/devices/sensors/sensor-types#gyroscope
- https://www.phonegg.com/list/302-Cell-Phones-with-Humidity-Sensor
- https://www.google.com/search?q=ambient+humidity+sensor&oq=ambient+Humidity+Sensor&aqs=chrome.0.0j0i22i30l4.6065j0j7&sourceid=chrome&ie=UTF-8
- https://en.wikipedia.org/wiki/MEMS_magnetic_field_sensor
- https://developer.android.com/guide/topics/sensors/sensors_position
- https://www.newscientist.com/article/dn25844-magnetic-messages-let-smartphones-receive-secret-data/#:~:text=A%20system%20called%20Pulse%20uses,produced%20by%20a%20nearby%20electromagnet.
- https://www.google.com/search?sxsrf=ALeKk024ow5POVzUa5rg7imeT4jEEx4UYw%3A1607348575075&ei=XzHOX6yNBNyG4-EP_ZmZoAM&q=atmospheric+pressure+sensor+in++phones&oq=atmospheric+pressure+sensor+in++phones&gs_lcp=CgZwc3ktYWIQAzIGCAAQFhAeOgQIABBHOgQIABBDOgIIADoICCEQFhAdEB5QlGpYgJEBYJ2SAWgAcAJ4AIAByQGIAY4OkgEFMC45LjGYAQCgAQGqAQdnd3Mtd2l6yAEIwAEB&sclient=psy-ab&ved=0ahUKEwjsuOOGgLztAhVcwzgGHf1MBjQQ4dUDCA0&uact=5
- https://www.google.com/search?sxsrf=ALeKk03p1P3QlxTpD2nBkNQmMVp9_uP6yA%3A1607347348022&ei=lCzOX5Zzy6_IA4X5sTg&q=atmospheric+pressure+sensor&oq=Atmosperic+Pressure&gs_lcp=CgZwc3ktYWIQAxgAMgQIABBDMgcIABCxAxBDMgQIABAKMgQIABAKMgQIABAKMgQIABAKMgcIABCxAxAKMgQIABBDMgQIABAKMgcIABCxAxAKOgQIABBHOgcIIxDqAhAnOgcILhDqAhAnOgQIIxAnOgUIABCRAjoECC4QQzoHCC4QJxCTAjoKCAAQsQMQFBCHAjoKCAAQsQMQyQMQQzoICAAQkgMQiwM6CggAELEDEEMQiwM6DgguELEDEIMBEMcBEK8BOgIIADoQCC4QsQMQgwEQxwEQrwEQClCu3B1YlZceYO-hHmgBcAJ4AIAB3wGIAY8akgEGMC4xNi4zmAEAoAEBqgEHZ3dzLXdperABCsgBCLgBAsABAQ&sclient=psy-ab
- https://www.google.com/search?q=Ambient+light+sensor+in+phones&oq=Ambient+light+sensor+in+phones&aqs=chrome..69i57j0i22i30l4.6939j0j7&sourceid=chrome&ie=UTF-8

Thank You

Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities!

School of Applied Sciences
Department of Zoology

Ref.No.: CUTM//Dean/SoAS/10/258 **Dated: 22.02.2021**

To
Dr. Manas Kumar Sinha
Senior Executive cum Officer
National Fresh Water Fish Brood Bank, Kausalyaganga, Bhubaneswar
Odisha-751002

Sub: Internship of fifteen M.Sc Zoology students of Centurion University of Technology and Management, Bhubaneswar.

Respected Sir,
Greetings from the Centurion University of Technology and Management!

Centurion University of Technology & Management (CUTM), the first multi sector State University of Odisha, was established through an act of State Legislative Assembly in 2010. Currently about 300 Students are Pursuing their career in M.Sc. As per the academic regulations, all M.Sc, students have to undergo training/internship/Apprenticeship training in Government organizations, industries and allied business organizations.

Your esteemed organization "National Fish Brood Bank, Bhubaneswar, Odisha" is emerged as a leading centre for fisheries research and serves as a catalyst for enhanced productivity and sustainability of agriculture and fisheries. fifteen M.Sc Zoology students from School of Applied Sciences are interested to learn and acquire knowledge on the different aspects of fisheries activities for duration of 4 weeks of joining your institute which will be helpful for their understanding of the subject.

For Further Queries you may contact Dr. Yashaswi Nayak (HOD and Dean) (Associate Professor, School of Applied Sciences, Centurion University of Technology & Management) Mobile No: 9861522222; Mail: yashaswi.nayak@cutm.ac.in.

Hence your consent and convenient schedule may kindly be communicated at the earliest.

Thanking you.
With warm regards,

Y. Nayak
22.02.2021

Dean
School of Applied Sciences
Centurion University of Technology and Management

DEAN
School of Applied Sciences
CENTURION UNIVERSITY OF TECHNOLOGY & MANAGEMENT
Bhubaneswar, Odisha

FORENSIC LAB

ISO 9001:2015 & 10002:2018 CERTIFIED

The undersigned do hereby confer this

Certificate of Internship

(Virtual)

to

Ms. BIJAYLAXMI JENA

On completing one month

Internship cum Training

In the Domain

Questioned document &

Fingerprint

at Clue4 Evidence Forensic Lab

between

1st Jun 2021 and 30th Jun 2021

Granted on this day

1st July 2021

Certificate No. : C4E/INT/JUN-10

Phaneendar B.N.

Director- Clue4 Evidence

Reg. No.-ALFA/EDU/2020/124

Certificate No.-FP-024

ALPHA LEGAL & FORENSIC ACCESS (P) LTD.

Certificate of Training

Proudly presented to

KRISHNA NAYAK

for successfully completion of one week online training on

"Fingerprint Science"

with an outstanding performance (Grade-A^+)

held from 07-04-2020 to 13-04-2020

The Training included various aspects of Fingerprint Examination, Fingerprint Identification, Fingerprint Development and Comparison, Report writing & Analytical Views on real cases

Granted on this day

16th Day of April 2020

DIRECTOR

Reg. No.-ALFA/EDU/2020/125

Certificate No.-FP-025

ALPHA LEGAL & FORENSIC ACCESS (P) LTD.

ALPHA LEGAL & FORENSIC ACCESS
FORENSIC ORGANIZATION

Certificate of Training

Proudly presented to

LITTLE JENA

for successfully completion of one week online training on

"Fingerprint Science"

with an outstanding performance (Grade A^{+})

held from 07-04-2020 to 13-04-2020

The Training included various aspects of Fingerprint Examination, Fingerprint Identification, Fingerprint Development and Comparison, Report writing & Analytical Views on real cases

Granted on this day

16th Day of April 2020

CERTIFIED SEAL

DIRECTOR

CLUE 4 EVIDENCE

FORENSIC LAB

ISO 9001:2015 & 10002:2018 CERTIFIED

The undersigned do hereby confer this

Certificate of Internship

(Virtual)

to

Ms. RASHMI PRAVA SAHOO

On completing one month

Internship cum Training

In the Domain

Questioned document & Fingerprint

at Clue4 Evidence Forensic Lab

between

1st Jun 2021 and 30th Jun 2021

Granted on this day

1st July 2021

Certificate No. : C4E/INT/JUN-09

Phaneendar B.N.

Phaneendar B.N.

Director- Clue4 Evidence

FORENSIC LAB
ISO 9001:2015 & 10002:2018 CERTIFIED

The undersigned do hereby confer this

Certificate of Internship

(Virtual)

to

Ms. SHRUTI MISHRA

On completing one month

Internship cum Training

In the Domain

Questioned document &

Fingerprint

at Clue4 Evidence Forensic Lab

between

1st Jun 2021 and 30th Jun 2021

Granted on this day

1st July 2021

Certificate No. : C4E/INT/JUN-08

Phaneendar B.N.
Director- Clue4 Evidence

Reg. No.-ALFA/EDU/2020/126

Certificate No.-FP-026

ALPHA LEGAL & FORENSIC ACCESS (P) LTD.

Certificate of Training

Proudly presented to

SHRUTI MISHRA

for successfully completion of one week online training on

"Fingerprint Science"

with an outstanding performance (Grade-A^{+})

held from 07-04-2020 to 13-04-2020

The Training included various aspects of Fingerprint Examination, Fingerprint Identification, Fingerprint Development and Comparison, Report writing & Analytical Views on real cases

Granted on this day

16th Day of April 2020

DIRECTOR

Reg. No.-ALFA/EDU/2020/103

Certificate No.-CSI-003

ALPHA LEGAL & FORENSIC ACCESS (P) LTD.

Certificate of Training

Proudly presented to

TANUSHREE MOHANTY

for successfully completion of one week online training on

"Crime Scene Investigation"

with an outstanding performance (Grade-A+)

held from 07-04-2020 to 13-04-2020

The Training included various aspects of Crime Scene Investigation, Role of Crime Scene Investigators at Different Scenes of Occurrence, Report writing & Analytical Views on real cases

Granted on this day

16th Day of April 2020

CERTIFIED SEAL

DIRECTOR

CERTIFICATE OF INTERNSHIP

Certificate No.: IIP-Kolkata/HIT/Internship/2021-22/B1-073

This is to confirm that

MALEPATI VISHNU VARDHAN CHOUDARY

has participated in the following internship course and passed the internal and external assessments necessary for successful completion.

INDIAN INSTITUTE OF PACKAGING (IIP) APPROVED INTERNSHIP TRAINING ON:

EMERGING TRENDS & INNOVATIVE FOOD PACKAGING TECHNOLOGIES IN FOOD INDUSTRIAL PRACTICES

Organized in collaboration with the Department of Food Technology, Haldia Institute of Technology and supported by Food Safety and Standards Authority of India (FSSAI) from 19 June – 11 July 2021 (Every Saturday & Sunday – total 24 hrs.)

Tutors: Dr. Tanweer Alam, Mr. Bidhan Das and Mr. Shubhabrata Basu

Course description/Focus areas:

- Definition and Functions of Packaging
- Mechanical Strength of Different Packaging Materials, Printing and Barcodes
- Quality Control Testing of Packaging Materials
- The Physical and Chemical Properties of the Packaging Materials
- Packaging & Labelling Regulations in Food Industry
- Advances in Emerging Packaging Technologies
- Packaging Requirements of Different Categories of Food and Beverages
- Safety Considerations in Food Packaging and Shelf-life Evaluation

Place and date: **Kolkata, August 2, 2021**

Delivery Type: Virtual Classroom

Signed on behalf of
Indian Institute of Packaging (IIP)

Bidhan Das

Bidhan Das
Deputy Director & Regional Head

INDIAN INSTITUTE OF PACKAGING
KOLKATA

BIDHAN DAS
Deputy Director & Branch Head
INDIAN INSTITUTE OF PACKAGING
Kolkata Centre

CERTIFICATE OF INTERNSHIP

Certificate No.: IIP-Kolkata/HIT/Internship/2021-22/B1-102

This is to confirm that

RADHAKRISHNAN DASH

has participated in the following internship course and passed the internal and external assessments necessary for successful completion.

INDIAN INSTITUTE OF PACKAGING (IIP) APPROVED INTERNSHIP TRAINING ON:

EMERGING TRENDS & INNOVATIVE FOOD PACKAGING TECHNOLOGIES IN FOOD INDUSTRIAL PRACTICES

Organized in collaboration with the Department of Food Technology, Haldia Institute of Technology and supported by Food Safety and Standards Authority of India (FSSAI) from 19 June – 11 July 2021 (Every Saturday & Sunday – total 24 hrs.)

Tutors: Dr. Tanweer Alam, Mr. Bidhan Das and Mr. Shubhabrata Basu

Course description/Focus areas:

- Definition and Functions of Packaging
- Mechanical Strength of Different Packaging Materials, Printing and Barcodes
- Quality Control Testing of Packaging Materials
- The Physical and Chemical Properties of the Packaging Materials
- Packaging & Labelling Regulations in Food Industry
- Advances in Emerging Packaging Technologies
- Packaging Requirements of Different Categories of Food and Beverages
- Safety Considerations in Food Packaging and Shelf-life Evaluation

Place and date: **Kolkata, August 2, 2021**

Delivery Type: Virtual Classroom

Signed on behalf of
Indian Institute of Packaging (IIP)

Bidhan Das

Bidhan Das
Deputy Director & Regional Head

INDIAN INSTITUTE OF PACKAGING KOLKATA

BIDHAN DAS
Deputy Director & Branch Head
INDIAN INSTITUTE OF PACKAGING
Kolkata Centre