**E-AUTOMOBS A BATTERY MANAGEMENT SYSTEM FOR ELECTRIC VEHICLES**

A PROJECT REPORT

Submitted by:

**ARUN KUMAR A P**

**LLMC17MCA1014**

*to*

*The APJ Abdul Kalam TechnologicalUniversity*

*in partial fulfillment of the requirements for the award of the Degree*

*of*

*Master of Computer Applications*

**

**Department of Computer Applications**

LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY

KUTTICHAL, THIRUVANANTHAPURAM 695574

JULY 2020

**E-AUTOMOBS A BATTERY MANAGEMENT SYSTEM FOR ELECTRIC VEHICLES**

A PROJECT REPORT

Submitted by:

**ARUN KUMAR A P**

**LLMC17MCA1014**

*to*

*The APJ Abdul Kalam Technological University*

*in partial fulfillment of the requirements for the award of the Degree*

*of*

*Master of Computer Applications*

**

**Department of Computer Applications**

LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY

KUTTICHAL, THIRUVANANTHAPURAM 695574

JULY 2020

**Lourdes Matha College of Science and Technology**

**(Managed By Archdiocese Of Changanacherry)**

**(Affiliated To APJ Abdul Kalam Technological University, Kerala)**

**Kuttichal, Thiruvananthapuram-695574**

**Department of Computer Applications**

****

**Certificate**

**This is to certify that the project work entitled “E-AUTOMOBS A” is a bonafide record of the work done by Mr ARUNKUMAR.A.P, Reg No LMC17MCA014, student of Department of Computer Applications, Lourdes Matha College of Science & Technology, Kuttichal, Thiruvananthapuram, affiliated to APJ Abdul Kalam Technological University, Kerala during the academic year 2019-2020 from January 2020 to July 2020 in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications from APJ Abdul Kalam Technological University, Kerala.**

**Prof. Justin G Russel**

**(Internal Supervisor)**

**Prof. Justin G Russel**

**(Project Co-Ordinator) Prof. Selma Joseph**

(**Head of the Dept.)**

**DECLARATION**

I undersigned hereby declare that the project report “E-AUTOMOBS A BATTERY MANAGEMENT SYSTEM FOR ELECTRIC VEHICLES”, submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Application of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Justin G Russel. This submission represents my ideas in my own words and, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University

Thiruvananthapuram

06/07/2020 ARUN KUMAR A P

**CONTENTS**

Contents PageNo.

ACKNOWLEDGEMENT i

ABSTRACT ii

Chapter 1. Introduction

1.1 General Background 01

1.2 Objective 02

Chapter 2. Literature Survey

2.1 Study of similar work 03

2.1.1 Existing System 09

Chapter 3. Overall Description

3.1 Proposed System 11

3.2 Features of Proposed System 11

3.3 Functions of Proposed System 12

3.4 Requirements Specification 12

3.5 Feasibility Analysis 13

3.5.1 Technical Feasibility 13

3.5.2 Operational Feasibility 14

3.5.3 Economical Feasibility

3.5.4 Behavioral Feasibility

Chapter 4. Operating Environment

4.1 Hardware Requirements 15

4.2 Software Requirements 21

4.3 Tools and Platforms

Chapter 5. Design

5.1 System Design 33

5.1.1 Activity Diagram and Flow chart 34

5.1.2 Project Diagram

5.2. Database Design 34

5.3. Input Design 42

5.4 Output Design 43

5.5 Program Design 47

Chapter 6. Functional and Non-Functional Requirements

6.1 Functional requirements 50

6.2 Non-Functional Requirements 50

Chapter 7. Testing

7.1 Testing Strategies 52

7.2 Unit Testing

7.3 Integration Testing

7.4 System Testing

7.5 Testing Results

Chapter 8. Results and Discussion

8.1 Results 54

8.2 Screen Shots 55

Chapter 9. Conclusion

9.1 System Implementation 57

9.2 Conclusion 58

9.3 Future Enhancement 58

References

1. Books 59

2. Websites

3. Journals and Publications

Appendices I 60

1. SCRUM Board

2. List of Tables

3. List of Figures

4. Abbreviations and Notation 63

5. Coding 64

**ACKNOWLEDGEMENT**

An endeavour over a long can be successful only with advice and support of many well wishers. I wish to place on record my profound indebtedness and gratitude to all those who have contributed directly or indirectly to make this project work a success. At the very onset, I express my gratitude to God Almighty, who sheltered me under his protective wings and showered on innumerable blessings throughout the period of this Master of Computer Application Course.

It is a great pleasure to express my sincere gratitude to **Rev. Dr.Tomy Joseph Padinjareveettil,** Director and **Prof. Dr. P.P Mohanlal**, Principal, Lourdes Matha College of Science and Technology for permitting to do this project with the fullest spirit.

I am highly obliged to **Prof. Selma Joseph,** Head of the Department of Computer Applications, Lourdes Matha College of Science and Technology, for being the source of inspiration throughout the course and for her valuable guidance.

With heart full of thanks, I would like to take up this opportunity to wish my internal guide **Prof Justin G Russel**, Assistant Professor and all staffs of department of computer applications for their endless support, encouragements and suggestions in various stages of the development of this project.

I thank my parents and friends for their moral support and encouragement for the successful completion of this project.

**ABSTRACT**

With The Internet of Things (IoT) technology have a wide variety of application in improvement and development of Battery Management System in Electric Vehicles. The need of energy storage system is one of the key factors in an Electric Vehicle. Electric Vehicles (EVs) are relatively standard due to their excellent electrical properties and flexibility, but the possibility of getting damage to their batteries is quite high, in case of overcharging or deep discharging. To prevent the possibility of damage, EVs’ batteries need a precise state of charge estimation to increase their lifespan and to protect the equipment they power this is already implemented in existing system but there is no option for monitoring individual cell inside the battery pack. An intelligent battery cell management system with power efficiency monitoring feature is most likely to optimize the performance and lifespan of the vehicle’s battery. This help in performing the tasks to help validate the safety of the battery. Thorough and frequent monitoring of vital operating parameters of each battery cell we can diagnose each cell individually and provide accurate monitoring system with proper data history using latest IOT platforms and API to store and retrieve data using HTTP and MQTT protocol over internet or via a Local Area Network.. Proposed project E-Automobs is a battery cell monitoring and management system in electric vehicles in order to diagnose each cell states in the battery to improve the efficiency. Battery is collection of individual cell, if there is any fault in a single cell, it can affect the whole battery pack, eventually affecting the efficiency and life of Electric Vehicle. E-Automobs can manage and monitor all the individual cell in a battery pack. So we are able to monitor each and every cell in a battery pack and its efficiency via using an open source platform known as ThingSpeak and hence we are able to monitor the charge of individual cell. An electric vehicle user can monitor the battery condition and state on the display provided, at the same time notifications are send over an android app, The data also get recorded on open source platform ThingSpeak where the service person or the EV authorities can monitor the battery condition in a graphical format. Also for servicing there is no need of any plugging of wires, a Bluetooth module has provided for ease of access and can monitor using a mobile phone and an android Arduino monitoring application

**CHAPTER 1**

**INTRODUCTION**

**1.1GENERAL BACKGROUND**

The proposed IOT based project is a smart battery management and monitoring embedded system for Electric Vehicles, where we are able to monitor and manage each and every individual cell in the battery pack. Battery means collection of multiple cells, and the efficiency of the battery pack depends upon each and every individual cell enclosed in it. Major drawback of the existing BMS system (Battery Management System) is that we can monitor and manage the whole battery pack but we cannot check or manage the efficiency of the individual cell enclosed in it. So our proposed system E-AUTOMOBS can manage and monitor each and every cell inside the battery pack by using a cheap and affordable Voltage Divider Circuit. An electric vehicle user can monitor the battery condition and state on the display provided at the same time the data also get recorded on open source platform ThingSpeak where the service person or the EV authorities can monitor the battery condition in a graphical format. Also for servicing there is no need of any plugging of wires, a Bluetooth module has provided for ease of access and can monitor using a mobile phone and an android free Arduino monitoring application

All the values are registered and acquired from several sensor and voltage divider circuit are sending to an Android application via Bluetooth communication and the data are monitored and calculated. If any of the cell inside the battery pack are not performing well, we can easily identify the location of the damaged cell easily. Hence it enable us to check and manage it in real time we are easily get notified when to change and service the system, by doing so it will ensure as long life for the battery as well as the Electric Vehicles

**1.2 OBJECTIVE AND SCOPE**

The Electric vehicles (EVs) are growing are rapidly because of the fact that they are cleaner and more efficient and even fun (think Tesla). Their growth, however, is still considered just a market problem: The end user should choose on the basis of what it costs to buy and run, or how it performs, etc.

Indians are famously value conscious. This is why consumers love diesel cars, despite their higher MRP and pollution relative to petrol counterparts. Even at today’s low oil prices, running a diesel sedan can cost about Rs3.8 per kilometre versus petrol’s Rs5.5. In contrast, CNG costs roughly Rs1.9/km, but it’s not widely available. The cost of EVs depends on electricity price, which varies significantly. At Rs7/kWh (kilowatt hour) of power, they cost only about Rs1.1/km This saves consumers driving 5,000km per year over Rs20,000 annually, and taxis much more as they drive 10-15 times as much. The catch is the upfront cost. EVs are expensive, primarily because of the battery. A single kWh of electricity is enough to go about 6km, so a 200km “full tank" range requires about 35 kWh of battery. Today’s prices for lithium ion batteries are about $250/kWh globally, which comes to Rs5.7lakh in battery costs, excluding import duties. Even with an eight-year lifespan and a 12% interest rate, justifying the battery costs on per kilo metre savings alone means one would have to drive over 25,000km per year. However, when battery prices fall to $100/kWh, as projected a few years out, EVs can become a game changer.

Range turns out to be key: 5,000km per year is only about 15km per day on average, while an urban taxi may do 300km daily. Higher range means not only more battery cost but weight as well. In an ideal world, we would have a smaller battery pack and simply recharge periodically. In practice, taxi and fleet vehicles can only charge overnight, and even private users may have limits on charging options. Without fast-charging infrastructure—fast-charging an EV requires much more power than household 15 amp sockets, which can only offer about 3 kW of power, so 35 kWh takes almost 12 hours to charge—one inevitably has “range anxiety". Unlike the US, most Indians don’t have a personal garage. Hence, widespread and company-agnostic public charging infrastructure becomes a key policy choice.

Despite of having so many advantages like low pollution, noise, high mileage and low maintenance for EV’s, one of the major drawbacks is that, the cost of battery and its maintenance. There is the scope of our project E-AUTOMOBS where we can manage and monitor the whole battery system regularly and precisely at very cheap cost in real time

**CHAPTER 2**

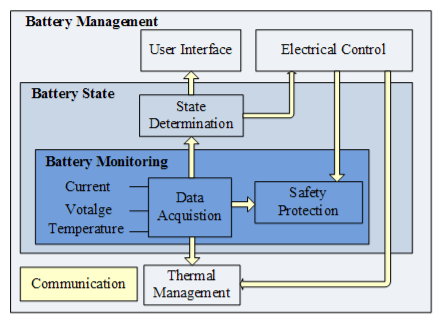
**LITERATURE SURVEY**

**2.1** **STUDY OF SIMILAR WORKS**

The Lithium-ion batteries have proved to be the battery of interest for Electric Vehicle manufacturers because of its high charge density and low weight. Even though these batteries pack in a lot of punch for its size they are highly unstable in nature. It is very important that these batteries should never be over charged or under discharge at any circumstance which brings in the need to monitor its voltage and current. This process gets a bit tougher since there are a lot of cells put together to form a battery pack in EV and every cell should be individually monitored for its safety and efficient operation which requires a special dedicated system called the Battery Management System. Existing BMI have the capability of monitoring the entire battery pack but it doesn’t have ability to monitor individual cell in the battery pack.

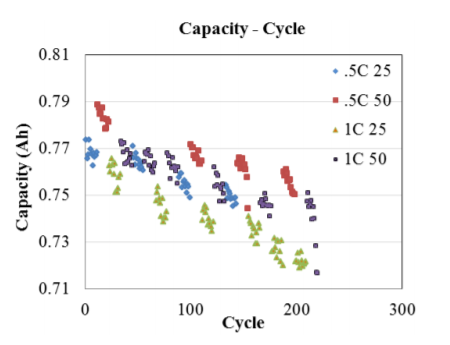
From portable electronics to electric vehicles (EVs), batteries are widely used as a main energy source in many applications. Interest in batteries for EVs can be traced back to the mid-19th century when the first EV came into existence. Today, since EVs can reduce gasoline consumption up to 75% , EV batteries have gained renewed attention in the vehicle market. Boston Consulting Group has reported that, by 2020, the global market for advanced batteries for electric vehicles is expected to reach US $25 billion, which is three times the size of today’s entire lithium-ion battery market for consumer electronics. The U.S. Council for Automotive Research (USCAR) and the U.S. Advanced Battery Consortium (USABC) have set minimum goals for battery characteristics for the long-term commercialization of advanced batteries in EVs and hybrid electric vehicles (HEVs) . To enlarge the market share of EVs and HEVs, safety and reliability are the top concerns of users. However, both of them are subject to not only the battery technology but also the management system for the battery. Therefore, a battery management system (BMS), as the connector between the battery and the vehicle, plays a vital role in improving battery performance and optimizing vehicle operation in a safe and reliable manner. In view of the rapid growth of the EV and HEV market, it is urgent to develop a comprehensive and mature BMS. Similar to the engine management system in a gasoline car, a gauge meter should be provided by the BMS in EVs and HEVs. BMS indicators should show the state of the safety, usage, performance, and longevity of the battery. Due to volatility, flammability and entropy changes, a lithium-ion battery could ignite if overcharged. This is a serious problem, especially in EV and HEV applications, because an explosion could cause a fatal accident [5]. Moreover, over-discharge causes reduced cell capacity due to irreversible chemical reactions. Therefore, a BMS needs to

monitor and control the battery based on the safety circuitry incorporated within the battery packs. Whenever any abnormal conditions, such as over-voltage or overheating, are detected, the BMS should notify the user and execute the preset correction procedure. In addition to these functions, the BMS also monitors the system temperature to provide a better power consumption scheme, and communicates with individual components and operators.



BATTERY DISCHARGE

Capacity Estimation under Varying Loads and Environmental Temperatures Battery degradation models are based on specific materials, environmental conditions, and charge-discharge cycling. Battery status is estimated when discharging at constant current and constant temperature. Ng [25] illustrated the SOC matrix related to the discharge voltage and different discharge rates of lithium-ion batteries (CGR 18650 from Panasonic Co.). However, when it comes to the combined factors, the degradation model based on the single influence factor is subject to query. Figure shows the capacity profile that was tested under two discharge rates and temperatures alternatively. Figure . Discharge capacity alternating at the different discharge rates with different temperatures. The objective of our experiment was to simulate the application under the combined factors. The conditional parameters are listed in Table 2. As Figure 4 shows, the capacity profile moved up with a higher temperature under the same discharge rate, while it went down with a higher discharge rate at the Energies 2011, 4 1852 same temperature. Thus, the combined factors increased the complexity of the capacity estimation more than by considering temperature fluctuation or varying discharge current alone.

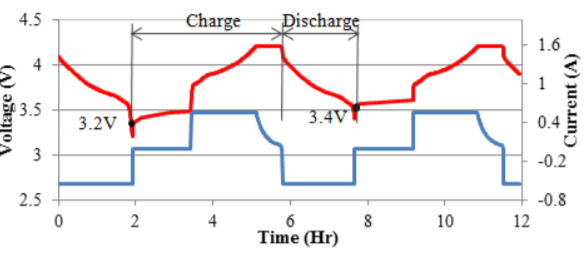
****

ESTIMATION OF MAXIMUM CAPACITY

The maximum capacity of the battery determines the performance and future life of the battery. The current method for estimation is largely based on the full discharge test. The online capacity is calculated by:

∫ Idt =Capacity

The longer the integration time, the higher the capacity will be. When the battery is fully discharged at a constant current, the maximum capacity is equal to the result obtained by Equation (11). However, the battery will not be always discharged at a constant discharge current and will not be depleted to the cut-off voltage every time. Figure 5 shows a battery discharged at different depths of discharge. The battery was charged to the full charge, 4.2 V, and stopped discharging at random cut-off voltages. This simulates operation in the real world. Thus, determining how to assess the maximum performance of a battery with partial discharge and variant current loads is a key challenge.

****

**2.1.1 EXISTING SYSTEM**

In the existing system we can control and manage the input charging voltage and discharging voltage. Automatic cut off facility if overcharging happens we are able to monitor and manage the entire battery pack but not the individual cell enclosed in the battery pack. Over heating indication

**2.1.2 DRAWBACK OF EXISTING SYSTEM**

Major drawbacks of the existing system are

* We cannot manage and monitor the battery pack in real time
* There is no option to monitor each and every cell in the battery pack and to find the damaged cell in the battery pack
* Current system is so expensive
* Existing system cannot send alerts based on individual cell

**CHAPTER 3**

**OVERALL DESCRIPTION**

**3. OVERALL DESCRIPTION**

The main intention of this project is to design an advanced battery monitoring and managing system using an embedded system and an android application for operation. It is monitored through wireless communication. For the desired operation, Arduino microcontroller and a Node microcontroller are used.

In the proposed system, an open source application is used to receive commands from the transmitter end to the receiver end for monitoring the cells in the battery pack. At the receiver side, a voltage divider circuit is interfaced with the Arduino microcontroller and a NodeMCU is used to monitor the overall voltage of the battery pack and sends the value over network in real time

**3.1 PROPOSED SYSTEM**

In our proposed system E-AUTOMOBS enables the electric vehicle to monitor and manage the overall functionality of an electric vehicle mainly the battery pack. Despite of having so many advantages like low pollution, noise, high mileage and low maintenance for EV’s one of the major drawbacks is that, the cost of battery and its maintenance. There is the scope of our project E-AUTOMOBS where we can manage and monitor the whole battery system regularly and precisely at very cheap cost in real time. The proposed project is a smart battery management and monitoring embedded system for Electric Vehicles, where we are able to monitor and manage each and every individual cell in the battery pack. Battery means collection of multiple cells, and the efficiency of the battery pack depends upon each and every individual cell enclosed in it. Major drawback of the existing BMS system (Battery Management System) is that we can monitor and manage the whole battery pack but we cannot check or manage the efficiency of the individual cell enclosed in it. So our proposed system E-AUTOMOBS can manage and monitor each and every cell inside the battery pack by using a cheap and affordable Voltage Divider Circuit

**3.2 FUNCTIONS OF PROPOSED SYSTEM**

The proposed project is a smart battery management and monitoring embedded system for Electric Vehicles, where we are able to monitor and manage each and every individual cell in the battery pack. Battery means collection of multiple cells, and the efficiency of the battery pack depends upon each and every individual cell enclosed in it. Major drawback of the existing BMS system (Battery Management System) is that we can monitor and manage the whole battery pack but we cannot check or manage the efficiency of the individual cell enclosed in it. So our proposed system E-AUTOMOBS can manage and monitor each and every cell inside the battery pack by using a cheap and affordable Voltage Divider Circuit, it can also monitor overall voltage of the battery pack on Electric Vehicle. All the values are registered and acquired from several sensor and voltage divider circuit are sending to an Android application via Bluetooth communication and the data are monitored and calculated. If any of the cell inside the battery pack are not performing well, we can easily identify the location of the damaged cell easily. Hence it enable us to check and manage it in real time we are easily get notified when to change and service the system, by doing so it will ensure as long life for the battery as well as the Electric Vehicle. An electric vehicle user can monitor the battery condition and state on the display provided, at the same time the data also get recorded on open source platform ThingSpeak where the service person or the EV authorities can monitor the battery condition in a graphical format. Also for servicing there is no need of any plugging of wires, a Bluetooth module has provided for ease of access and can monitor using a mobile phone and an android free Arduino monitoring application

**3.3 FEATURES OF PROPOSED SYSTEM**

* Efficient
* Cost effective
* User friendly
* Real time monitoring
* Increased efficiency
* Long life of battery pack

**3.4 REQUIREMENTS SPECIFICATION**

System analyst talks to a variety of persons to gather details about the business process and their opinions of why things happen as they do and their ideas for changing the process. These can be done through questionnaire’, detailed investigation, observation, collection of samples etc. As the details are collected, the analyst study the requirements data to identify features the new system must have, including both the information the system should produce and operational features such as processing controls, response times and input-output methods.

Requirements specification simply means, “Figuring out what is to be made before making it.” It determines what people need before starting to develop a product for them. Requirement definition is the activity of translating the information gathered in to a document that defines a set of requirements. These should reflect what consumer wants.

The requirements for an effective battery management system for an electric vehicle are as follows:

* A real-time monitoring system, to ensure the successful monitoring of the cell and overall function of the Battery pack
* A system that will work and monitor whenever we are using it
* Additionally, the other features that are implemented on the system must work efficiently.

The above requirements are subsequently the aims of this project. The project will consist of a concept level system that will meet all the above requirements.

**3.5 FEASIBILITY ANALYSIS**

The initial investigation points to be question whether the project is feasible. The feasibility study concerns with the considerations made to verify whether the system fit to be developed in all terms. Once the idea to develop software is put forward, the question that rises first will pertain to be the feasibility aspects. The prime objective of feasibility study is to ensure that the problem is worth to be solved. At the stage a cost benefit analysis is performed to assertion that the benefit from the system will over rule the cost association with the whole analysis, design and development of the new system. An important outcome of the preliminary investigation determining whether the system required is feasible.

The proposed system is tested in all four aspects of feasibility.

* Technical Feasibility study
* Operational Feasibility study
* Economic Feasibility study
* Behavioural Feasibility study

**3.5.1 TECHNICAL FEASIBILITY**

The main objective of feasibility study is to test the technical, social and economic feasibility of developing a system. Investing the existing system in the area under investigation and generating ideas about the new system does this. Feasibility study has been done to gather required information. Training, experience and common sense are required for collection of the information. Data was gathered and checked for completeness and accuracy. Analysing the data involved identification of the components of the system and their interrelationship and identified the strength and weakness of the system. My system is developed by using front end as Python and Android .The front end is technically feasible and it has lot of features as well as its secure too. So the technical part of this project is very secure. So my system is technically feasible

**3.5.2 OPERATIONAL FEASIBILITY**

There is no difficulty in implementing the system. The proposed system is effective, user friendly and functionally. The user of the system must be completely unaware of the internal working of the system so that the users will not face any problem running the system. The system thus reduces the responsive time of computer thereby, the system is found to be operationally feasible. Design is the only ways that can accurately translate the user needs into finished software or system. Without software design, the risk of building an unstable system exists. System design provides the procedural details necessary for implementing the system recommended in the feasibility study

**3.5.3 ECONOMIC FEASIBILITY**

There Economic and Financial analysis is used for evaluating the effectiveness of the system. the project is technically and operationally feasible.

The software used for developing android application is Android Studio and. The hardware consists of a voltage divider circuit, temperature sensors, humidity sensors, speed sensor Arduino microcontroller and NodeMCU. Also a Smartphone is needed for controlling and communicate with the system. The overall cost for making this prototype is considered as Rs.10000/-

**3.5.4 BEHAVIOURAL FEASIBILITY**

The behavioural feasibility depends upon whether the system performed in the expected way or not. Feasibility study is a test of system proposal according to it workability, impact on organization, ability to meet the user’s need and effective use of resources. However, a feasibility study provides a useful starting point for full analysis.

My system is behaviourally feasible because of the effective use of the resources and also the system satisfied the user needs and the system is user friendly.

**CHAPTER 4**

**OPERATING ENVIRONMENT**

**4.1 HARDWARE CONFIGURATIONS**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

**PROCESSOR**

SYSTEM : I3 PROCESSOR

**MEMORY**

TOTAL RAM :1GB

**STORAGE**

HARD DISK :250GB

**INPUT DEVICES**

KEYBOARD :WIRED OR WIRELESS

MOUSE : OPTICAL MOUSE

**OUTPUT DEVICES**

MONITOR :STANDARD MONITORS

**4.2 HARDWARE REQUIREMENTS**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

* Microcontroller : Arduino Uno
* Microcontroller : Node MCU
* Voltage Divider Circuit – Circuit to monitor individual cell
* Power Source – Lithium-ion Battery

**4.3 SOFTWARE REQUIREMENTS**

* Operating System : Windows 10, Android
* Software : Arduino IDE

: Embedded C

**:** Thing Speak

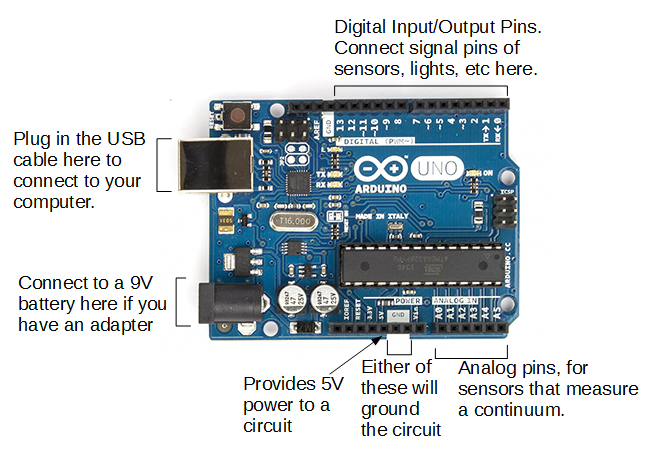
**4.3 TOOLS AND PLATFORMS**

* Thing Speak
* Arduino IDE

**4.2 HARDWARE REQUIREMENTS**

**4.2.1 Arduino UNO**

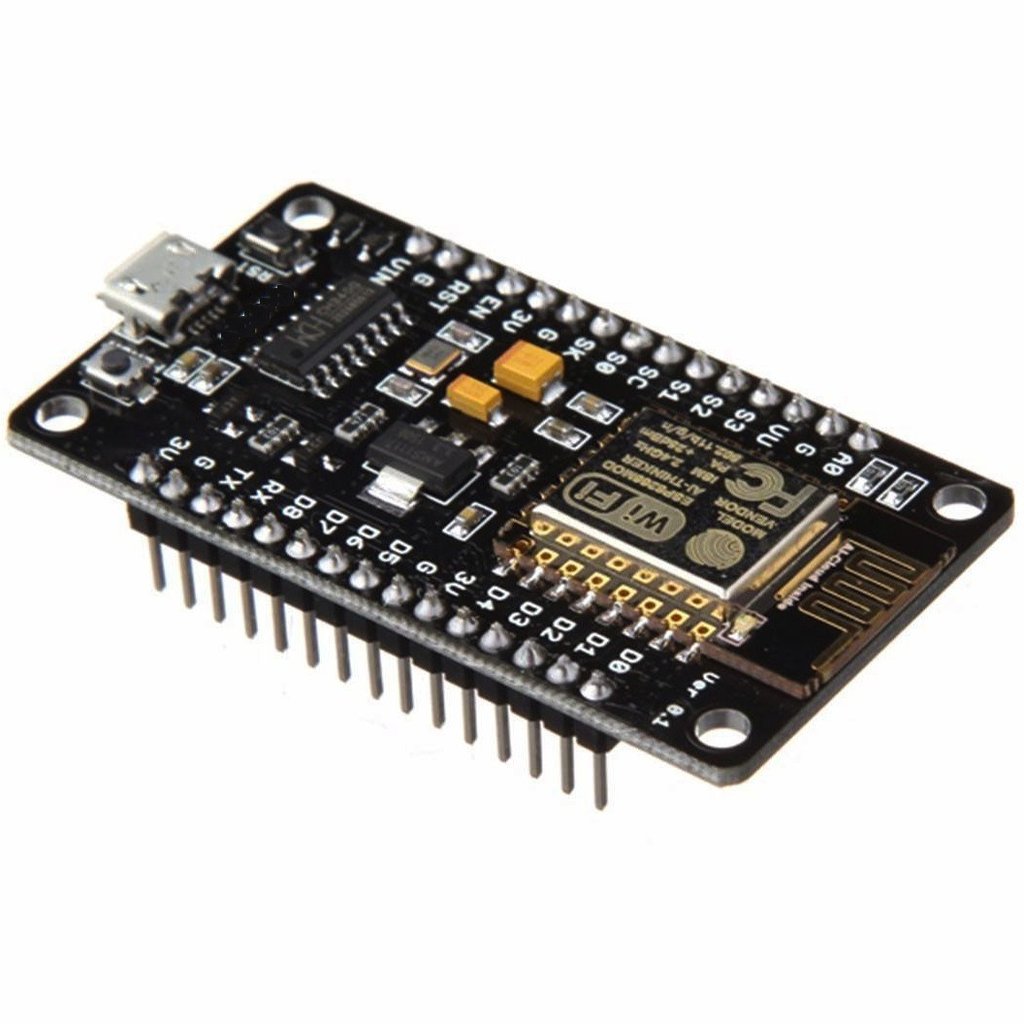
The arduino board is an open source programmable circuit board. The Arduino Uno is the microcontroller board based on the ATmega 328. It is a programmable microcontroller for prototyping electromechanical devices. It has 14 digital inputs/output pins (of which 6 can be used as PWM output),6 analog inputs , a 16 MHz ceramic resonators the arduino differs from all preceding board is that it does not use the FTDI USB to serial driver chip. Arduino that can be integrated into a wide variety of maker space projects both simple and complex. This board contains a microcontroller which is able to be programmed to sense and control objects in the physical world. By responding to sensors and inputs, the Arduino is able to interact with a large array of outputs such as LEDs, motors and displays. Because of its flexibility and low cost, Arduino has become a very popular choice for makers and maker spaces looking to create interactive hardware projects.



**4.2.2 NodeMCU**

NodeMCU is an open-source firmware and development kit that helps you to prototype or builds IoT products. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The firmware uses the Luascripting language. It is based on the eLua project and built on the Espressif Non-OS SDK for ESP8266.

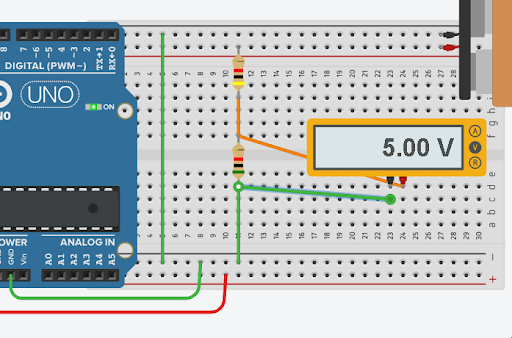
The firmware is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines node and "MCU". MCU stands for Microcontroller Unit - which really means it is a computer on a single chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. They are used to automate automobile engine control, implantable medical devices, remote controls, office machines, appliances, power tools, toys etc. The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs are open source..The firmware uses the Luascripting language. The firmware is based on the eLua project, and built on the Espress if Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32has also been implemented.



**4.2.3 Voltage Divider Circuit**

Voltage dividers are also known as potential dividers, because the unit of voltage, the “Volt” represents the amount of potential difference between two points. A voltage or potential divider is a simple passive circuit that takes advantage of the effect of voltages being dropped across components which are connected in series. The potentiometer, which is a variable resistor with a sliding contact, is the most basic example of a voltage divider as we can apply a voltage across its terminals and produce an output voltage in proportion to the mechanical position of its sliding contact. But we can also make voltage dividers using individual resistors, capacitors and inductors as they are two-terminal components which can be connected together in series.

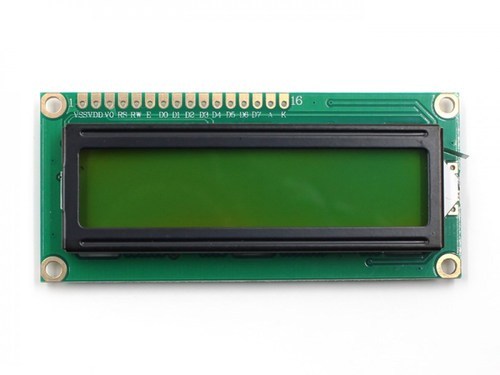
Arduino analog inputs can be used to measure DC voltage between 0 and 5V (on 5V Arduino such as the Arduino Uno when using the standard 5V analog reference voltage). The range over which the Arduino can measure voltage can be increased by using two resistors to create a voltage divider. The voltage divider decreases the voltage being measured to within the range of the Arduino analog inputs. Code in the Arduino sketch is then used to calculate the actual voltage being measured.

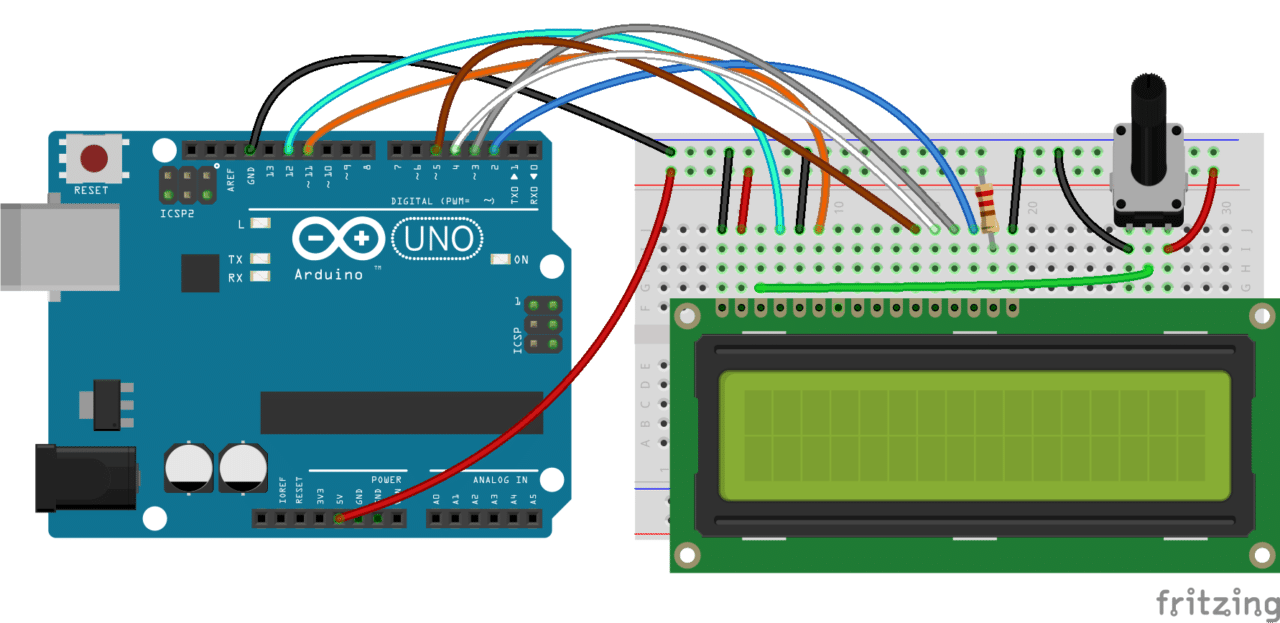


**4.2.4 16\*2 LED Display**

LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO’s or calculators. The appearance and the pin outs have already been visualized above now let us get a bit technical.

16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots.

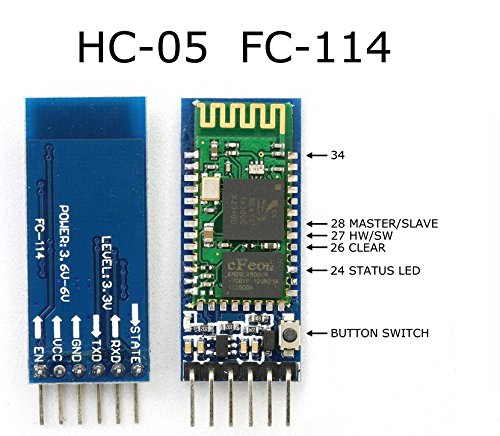
****

****

**4.2.5 Bluetooth Module**

HC‐05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04 External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

The Bluetooth module HC-05 is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc.

****

**4.2.5 Lithium Battery**

Lithium batteries are [primary](https://en.wikipedia.org/wiki/Primary_Cell) [batteries](https://en.wikipedia.org/wiki/Battery_(electricity)) that have metallic [lithium](https://en.wikipedia.org/wiki/Lithium) as an [anode](https://en.wikipedia.org/wiki/Anode). These types of batteries are also referred to as lithium-metal batteries. They stand apart from other batteries in their high [charge density](https://en.wikipedia.org/wiki/Charge_density) (long life) and high cost per unit. Depending on the design and chemical compounds used, lithium cells can produce voltages from 1.5 V (comparable to a [zinc–carbon](https://en.wikipedia.org/wiki/Zinc%E2%80%93carbon_battery) or [alkaline battery](https://en.wikipedia.org/wiki/Alkaline_battery)) to about 3.7 V. Disposable primary lithium batteries must be distinguished from secondary [lithium-ion](https://en.wikipedia.org/wiki/Lithium-ion_battery) or a [lithium-polymer](https://en.wikipedia.org/wiki/Lithium_polymer_battery),[[1]](https://en.wikipedia.org/wiki/Lithium_battery#cite_note-1) which are [rechargeable](https://en.wikipedia.org/wiki/Rechargeable_battery) batteries. Lithium is especially useful, because its ions can be arranged to move between the anode and the [cathode](https://en.wikipedia.org/wiki/Cathode), using an [intercalated](https://en.wikipedia.org/wiki/Intercalation_(chemistry)) lithium [compound](https://en.wikipedia.org/wiki/Chemical_compound) as the cathode material but without using lithium metal as the anode material. Pure lithium will instantly react with water, or even moisture in the air; the lithium in lithium ion batteries is in a less reactive compound.

Lithium batteries are widely used in portable consumer electronic devices, and in electric vehicles ranging from full sized vehicles to radio controlled toys. The term "lithium battery" refers to a family of different lithium-metal chemistries, comprising many types of cathodes and [electrolytes](https://en.wikipedia.org/wiki/Electrolytes) but all with metallic lithium as the anode. The battery requires from 0.15 to 0.3 kg of lithium per kWh. As designed these primary systems use a charged cathode, that being an electro-active material with crystallographic vacancies that are filled gradually during discharge.

The most common type of lithium cell used in consumer applications uses metallic lithium as anode and [manganese dioxide](https://en.wikipedia.org/wiki/Manganese_dioxide) as cathode, with a salt of lithium dissolved in an organic [solvent](https://en.wikipedia.org/wiki/Solvent).

****

**4.2 SOFTWARE REQUIREMENTS**

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

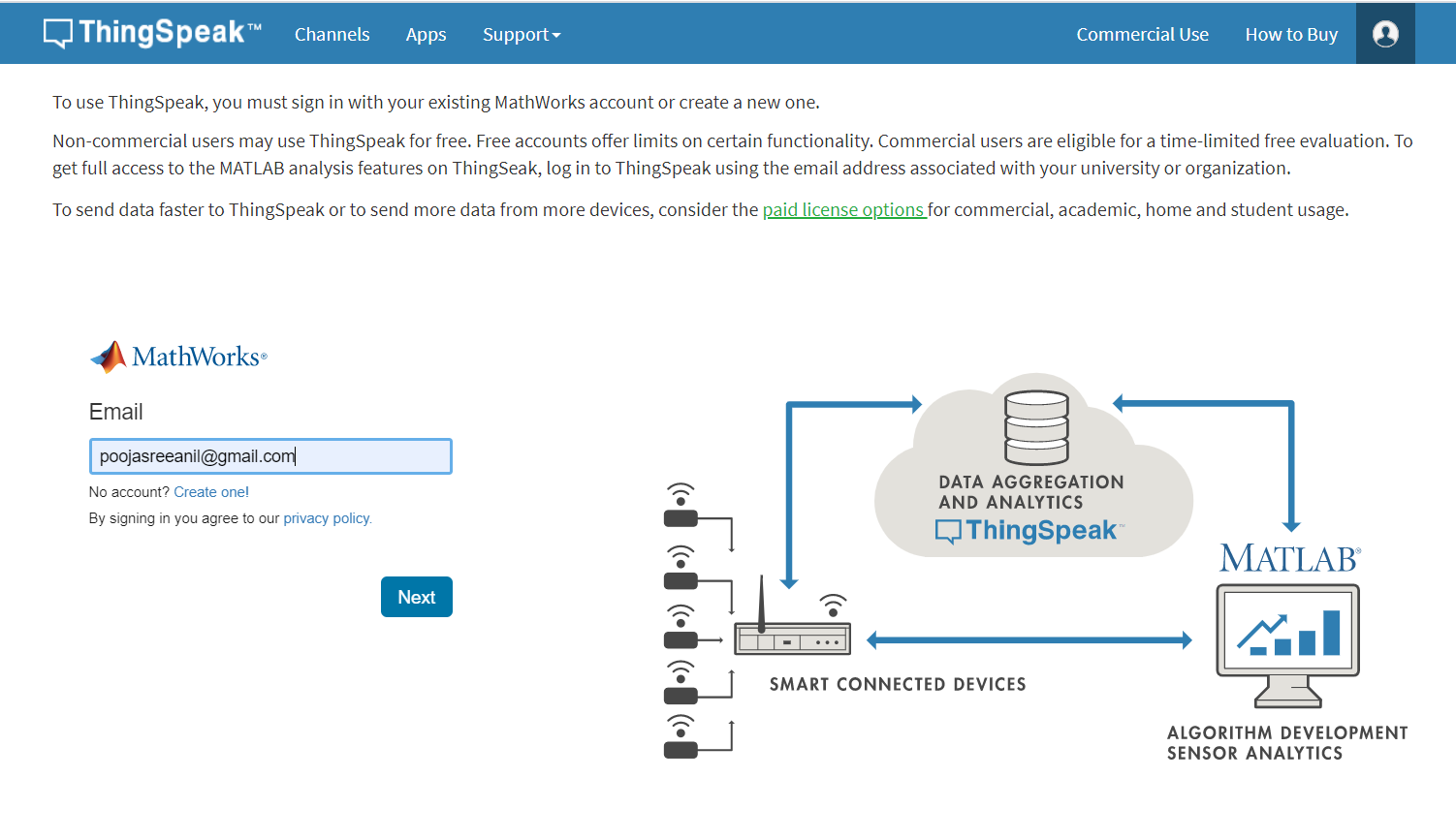
|  |  |  |
| --- | --- | --- |
| SOFTWARE | THING SPEAK,  ARDUNIO IDE | Collect Agricultural Data over The Things Network - MATLAB & SimulinkWhat is Arduino IDE? And its different functions. - Bot Solvers |
| OPERATING SYSTEM | **WINDOWS 10** | How to use PGP encryption on Windows for free using Gpg4win |
| LANGUAGE | **EMBEDDED C** | Advanced Linux Programming - Embedded Systems | edjiO |
| DATABASE | **THINGSPEAK** | Collect Agricultural Data over The Things Network - MATLAB & Simulink |

**4.3 TOOLS AND PLATFORMS**

**4.3.1 THING SPEAK**

Thing Speak is a platform providing various services exclusively targeted for building applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plug-in and apps for collaborating with web services, social network and other APIs..The core element of Thing Speak is a ‘Thing Speak Channel’. To use Thing Speak, we need to signup and create a channel. Once we have a channel, we can send the data, allow Thing Speak to process it and also retrieve the same. Thing Speak allows you to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of Thing Speak include the ability to:

* Easily configure devices to send data to Thing Speak using popular IoT protocols.
* Visualize your sensor data in real-time.
* Aggregate data on-demand from third-party sources.
* Use the power of MATLAB to make sense of your IoT data.
* Run your IoT analytics automatically based on schedules or events.
* Prototype and build IoT systems without setting up servers or developing web software.
* Automatically act on your data and communicate using third-party services like Twitter.

****

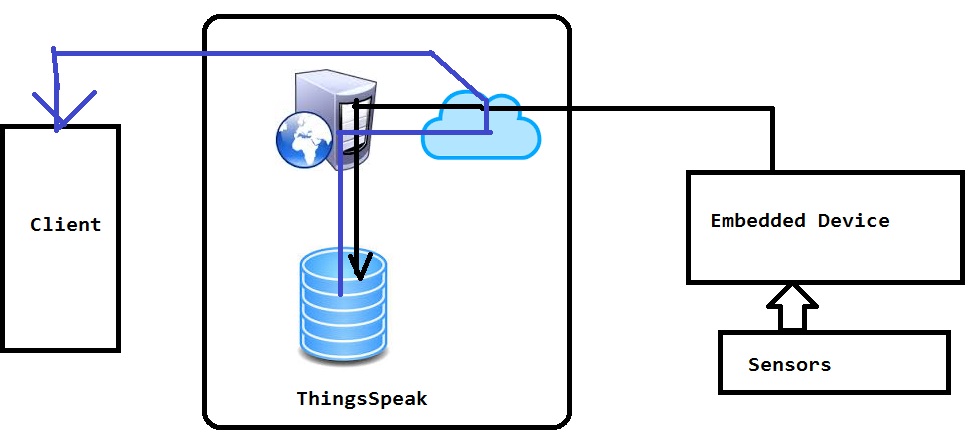
****

fig: Thing Speak Architecture

**THING SPEAK IOT PLATFORM**

* **Channel Visualisation:**

The system uses Thing Speak IoT platform for monitoring the data. The system uses channels for monitoring the data from each E-Automobs. Android app named Thing Chart is used to visualize and monitor the data through a smart phone.

* **React App**:

React app Send a tweet or trigger a Thing HTTP request when the Channel meets a certain condition. In this project when there is any voltage get dropped Thing speaks Tweet to send warning message to the EV authority

**Thing http App**:

Thing HTTP enables communication among devices, websites, and web services without having to implement the protocol on the device level. It is done using the GET, PUT, POST and DELETE methods of HTTP

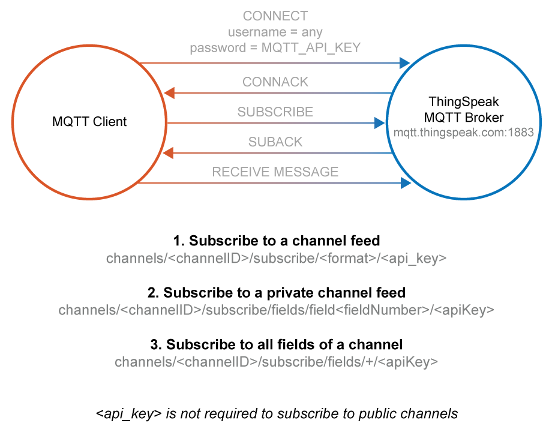
* **Talk Back APP**:

Talk Back is used to queue up commands and then allow a device to act upon these queued commands. Talk Back API is used to add, get and execute a Talk Back command, which can be accessed by Thing HTTP app. In this project Talk Back app is used to add commands for device control based on time or twitter message.

**PROTOCOLS**

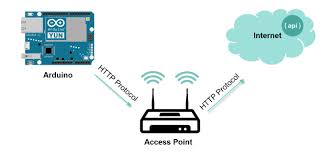
* **MQTT**

MQTT is a common protocol used in IoT systems to connect low-level devices and sensors. MQTT is used to pass short messages to and from a broker. Thing Speak has recently added an MQTT broker so devices can send messages to Thing Speak. A message might contain the current temperature in an office collected by a sensor. Thing Speak takes the message and stores its content in a Thing Speak channel. Once the data is in a channel, you can easily visualize and analyze the data with MATLAB code.



**HTTP ( Hyper Text Transfer Protocol)**

The Thing HTTP App allow a microcontroller or low level device to connect to any web service using HTTP over internet. We can create an HTTP object using Thing HTTP app and then control the object using simple API commands Thing speak supports GET,POST,PUT,DELETE method. We have a device interface with many web services and API without having to implementing on the device level.



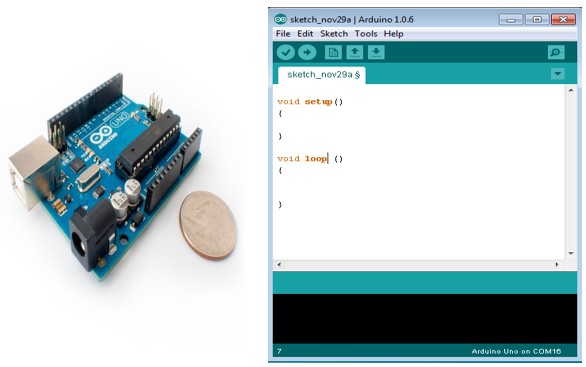
**4.3.2 Arduino IDE(The**[**Arduino**](https://en.wikipedia.org/wiki/Arduino)**Integrated Development Environment)**

The [Arduino](https://en.wikipedia.org/wiki/Arduino) Integrated Development Environment ([IDE](https://en.wikipedia.org/wiki/Integrated_development_environment)) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in functions from [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B_(programming_language)). It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The key features are −

* Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
* You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
* Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.

Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.



**4.3.8 Embedded C**

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, data type declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

**CHAPTER 5**

**DESIGN**

**5.1 SYSTEM DESIGN**

System design involves translating system requirements and conceptual design into technical specifications and general flow of processing. After the system requirements have been identified, information has been gathered to verify the problem and after evaluating the existing system, a new system is proposed.

System design is the process of planning of new system or to replace or complement an existing system. It must be thoroughly understood about the old system and determine how computers can be used to make its operations more effective.

There are two levels of system design:

• Logical design.

• Physical design.

In the logical design, the designer produces a specification of the major features of the system which meets the objectives. The delivered product of logical design includes current requirements of the following system components:

• Input design.

• Program design

• Output design

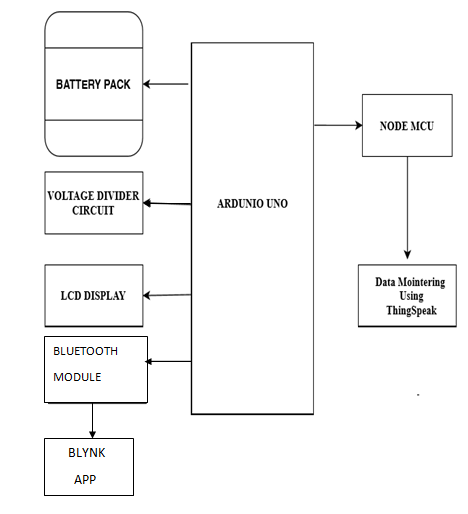
• Database design

Physical design takes this logical design blue print and produces the program software, files and a working system. Design specifications instruct programmers about what the system should do. The programmers in turn write the programs that accept input from users, process data, produce reports and store data in files.

**5.1.1 Block Diagram and Flow Chart**

**i. Block Diagram**

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams. Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction.



**ii. Flow Chart**

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields

Components of Flowchart

Start/End

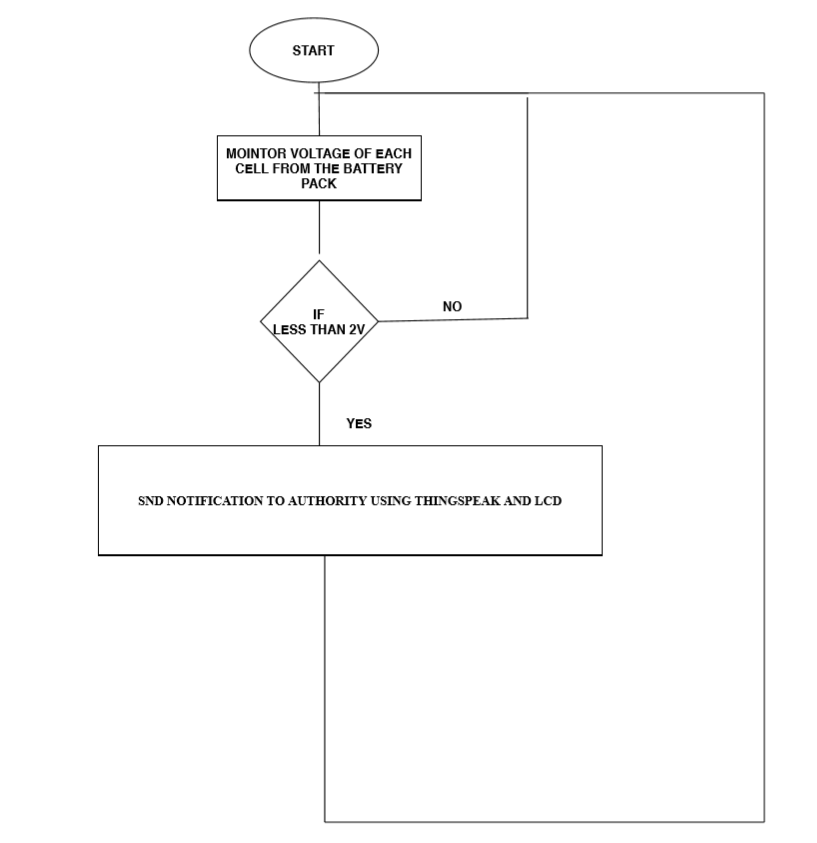
Connector

Input/output

Process

Decision

Figure: Components of flow chart



**iii. Activity Diagram**

An activity diagram is a UML behaviour diagram that represents the workflow of stepwise activities of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behaviour diagrams. An activity diagram is a behavioural diagram i.e. it depicts the behaviour of a system. An activity diagram is used by developers to understand the flow of programs on a high level.

* Components of Activity Diagram

Start Point/initial state

Activity

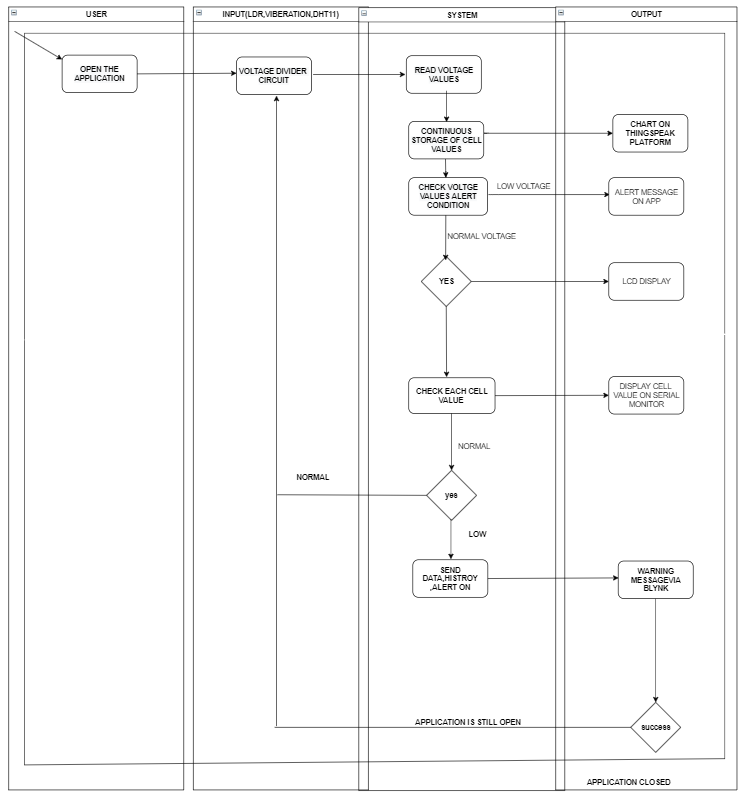
Action flow

Class/object

Decision/branching

Merge

Fig 5.1.1 Components of activity diagram



**5.2 DATABASE DESIGN**

The most important aspect of building software systems is database design. The highest level in

the hierarchy is the database. It is a set of inter-related files for real time processing. It contains the

necessary data for problem solving and can he used by several users accessing data concurrently.

The general objective of database design is to make the data access easy, inexpensive and flexible

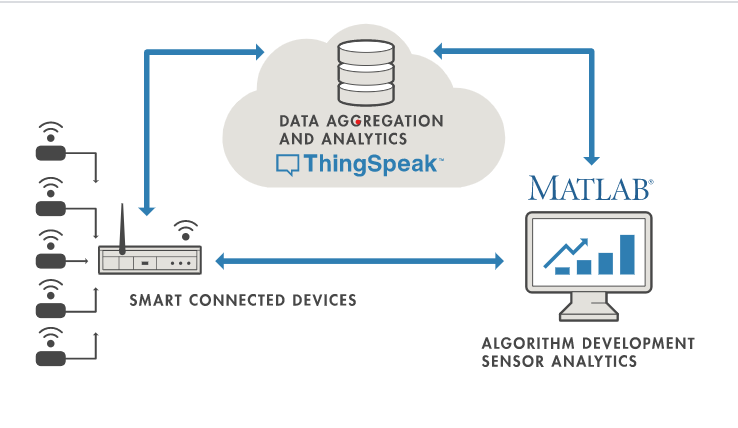
to the user. Database design is used to define and then specify the structure of business used in the

client/server system. A business object is nothing but information that is visible to the users of the

system.

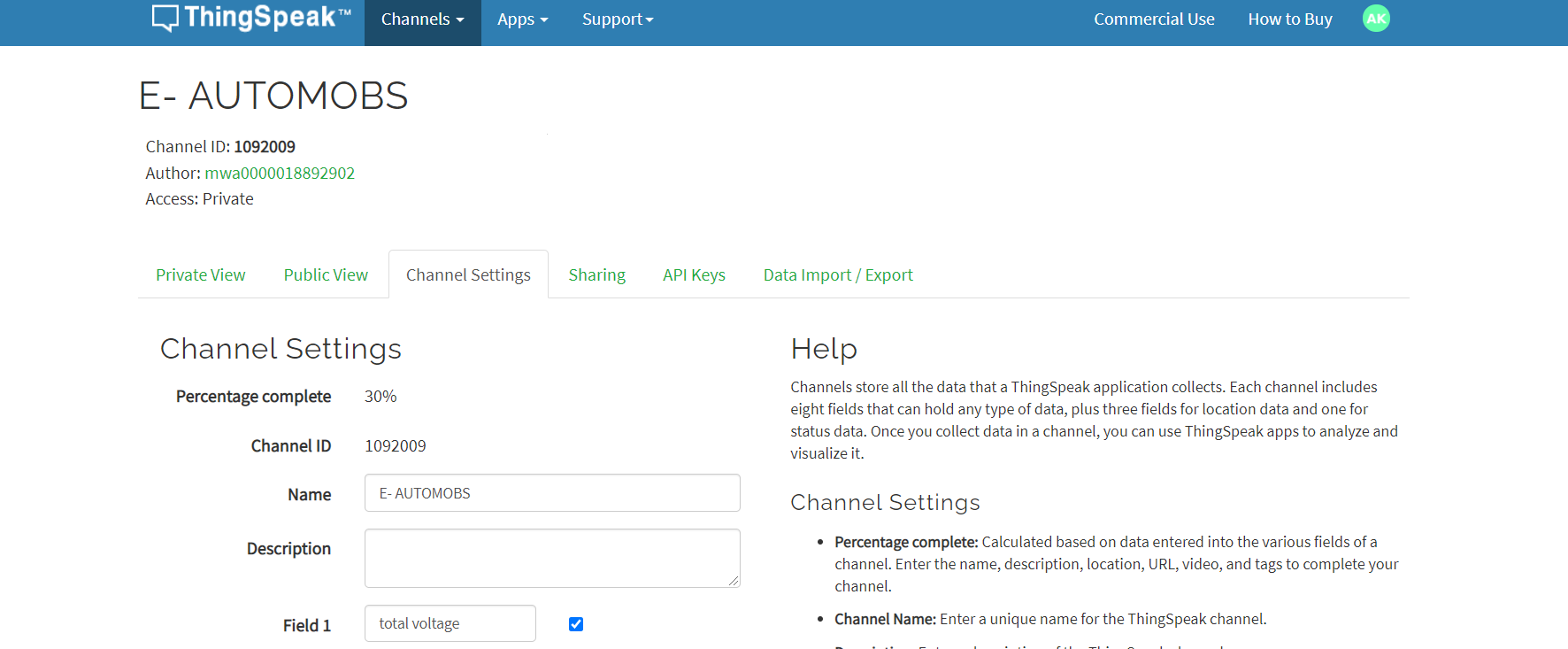
Database Name: ThingSpeak

Description : Store the values from Sensor and histroical time.



**5.3INPUT DESIGN**

The input design is the process of converting the user oriented inputs in to the computer based format. The goal of designing input data is to make automation as easy and free from errors as possible. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right message and help for the user at right time are also considered for the development of the project. The input design is the link between the information system and the user.

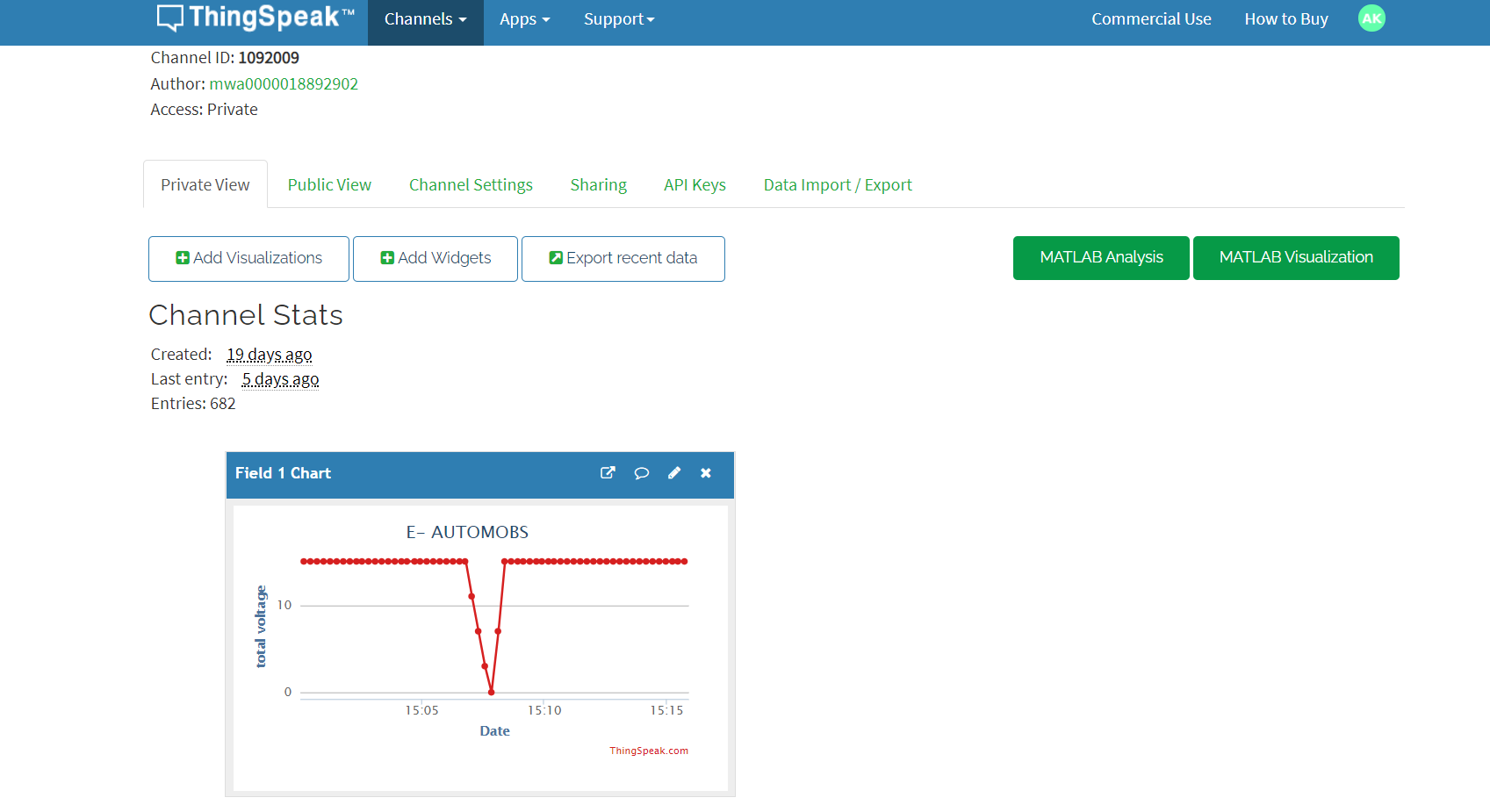


**5.4 OUTPUT DESIGN**

The output generally refers to the results and information that are generated by the system. A major

form of the output is the display of the information generated by the system and servicing the user

requests to the system. In this project the necessary outputs are;



|  |  |  |
| --- | --- | --- |
| PROCESS | INPUT DESIGN | OUTPUT DESIGN |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | . |

**5.5 PROGRAM DESIGN**

**CHAPTER 6**

**FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

**6.1 FUNCTIONAL REQUIREMENTS**

In software engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Generally, functional requirements are expressed in the form “system must do requirement”. Functional requirements for each of the uses cases described below:

• Descriptions of data to be entered into the system.

• Descriptions of operations performed by each inputs.

• Descriptions of work-flows performed by the system.

• Descriptions of system outputs.

• How the system meets applicable regulatory requirements.

**6.2 NON-FUNCTIONAL REQUIREMENTS**

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non-functional requirements are “system shall be requirement ". Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes”, “quality goals", "quality of service requirements" and "non-behavioural requirements. Some of the non-functional requirements are mentioned below:

**i. Performance requirements**

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance.

**ii. Operating constraints**

List any run-time constraints. This could include system resources, people, needed software, etc.

**iii. Platform constraints**

Discuss the target platform. Be as specific or general as the user requires. If the user doesn't care, there are still platform constraints.

**iv. Accuracy and Precision**

Requirements about the accuracy and precision of the data. Beware of 100% requirements; they often cost too much.

**v. Modifiability**

Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).

**vi. Portability**

The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

**vii. Reliability**

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

**viii. Security**

One or more requirements about protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, time) to break into the system. Do not discuss solutions (e.g. passwords) in a requirements document.

**ix. Usability**

Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

**CHAPTER 7**

**TESTING**

**7.1 TESTING STRATEGIES**

A **test strategy** is an outline that describes the testing approach of the software development cycle. The purpose of a test strategy is to provide a rational deduction from organizational, high-level objectives to actual test activities to meet those objectives from a quality assurance perspective. The creation and documentation of a test strategy should be done in a systematic way to ensure that all objectives are fully covered and understood by all stakeholders. It should also frequently be reviewed, challenged and updated as the organization and the product evolve over time. Furthermore, a test strategy should also aim to align different stakeholders of quality assurance in terms of terminology, test and integration levels, roles and responsibilities, traceability, planning of resources, etc.

**7.2 UNIT TESTING**

Unit testing focuses on verification effort on the smallest limit of software design. Using the unit test plan prepared in the design phase of the system, important control paths are tested to uncover the errors within the module. This testing was carried out during the coding itself. In this testing step each module is going to be working satisfactorily as the expected output from the module.

**7.3 INTEGRATION TESTING**

It is the systematic technique for constructing the program structure to uncover errors associated with the interface. The objective is to take unit-tested module and built the program structure that has been dictated by design. All modules are combined in this step. Then the entire program is tested as a whole. If a set of errors is encountered connection is difficult because the isolation of causes is complicated by vastness of the entire program. Using this test plan preparing the design phase of the system, the integration was carried out. All the errors found in the system were corrected for the next testing step.

**7.4 SYSTEM TESTING**

In system testing, the software and other system elements are tested as a whole. System testing is actually a series of different test whose primary purpose is to fully exercise the computer-based system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.No** | **Test Case** | **Input** | **Expected**  **Output** | **Actual**  **Output** | **Pass or**  **Fail** |
| 1 | Application is  Open | Read each cell  Value | Chart on Thing  Speak | Get Chart | Pass |
| 2 | Thing Speak is  Open | Login with  User name  And password. | Get in to the  platform | Get into  ThingSpeak | Pass |
| **233** 3 | Android app is  Open | Authorised  Bluetooth device  of E-V verified | Get ready with the  App for voltage  result | Get notification  If the total voltage  dropped below 5v | Pass |
| 4 | Checking the total  Voltage on the  Lcd screen | Read total voltage  value | Total value should  displayed on  screen | Value displayed | Pass |
| 5 | Android app is  Open | Authorised Bluetooth  Device of E-V verified | Get ready with app for  Individual voltage result | Only the total value ge  t displayed | fail |
| 6 | Low battery notification | Read total voltage | “un safe” Message on  screen | “un safe” Message  Displayed on Screen | Pass |

Table 7.1 Test Result

**CHAPTER 8**

**RESULT AND DISCUSSION**

Results of our experiments show that our embedded system E-AUTOMOBS is a battery monitoring system based on Iot technology mainly focused on electric vehicles, that has an on screen LCD display which shows the current status of the battery pack, and warning notification can also be displayed another speciality is that there is an android app which shows the current status of the total voltage of the battery pack . For ease of servicing the battery, a Bluetooth device has been embedded into the device so, At service point there is no need to even touch the vehicle we can check using an android application designed based on a free platform, where we can monitor the individual cell.

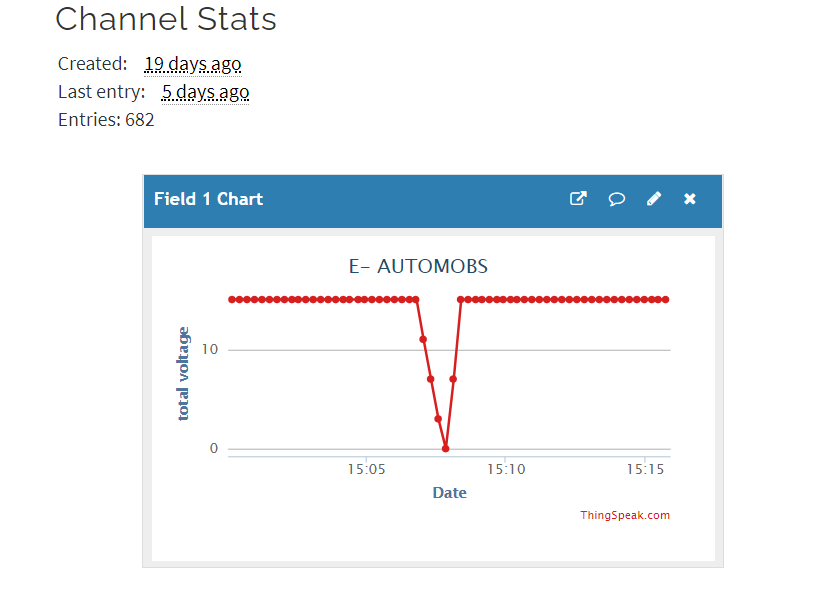
A real time monitoring web platform has also been designed using ThingSpeak for the vehicle manufacturer to monitor the real time graphical data of the battery pack. The project is successfully tested for all the commands and it also ensure precise monitoring of battery pack by implementing as a central processing unit. Once the variation is detected, The system send notification to user as well as the vehicle manufacturer.

**8.1 RESULTS**

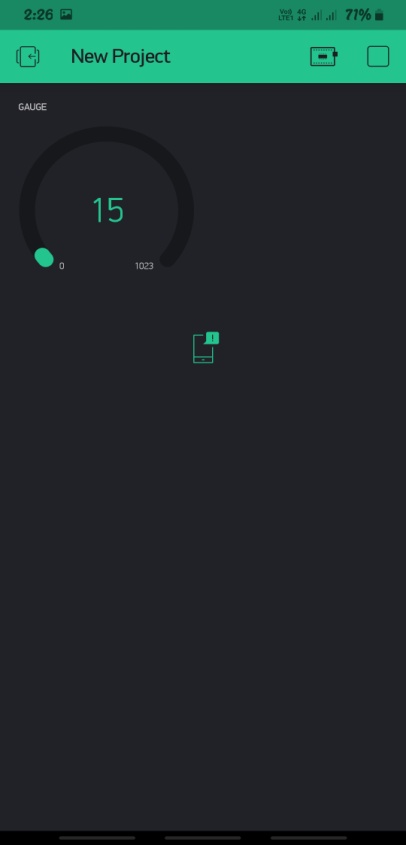
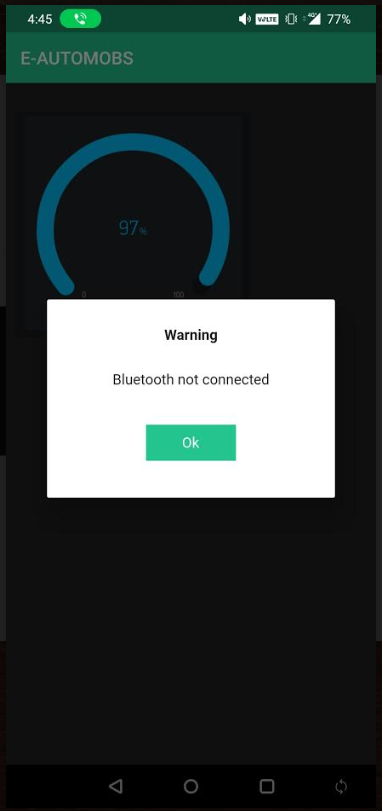
The proposed system incorporated with the following features.

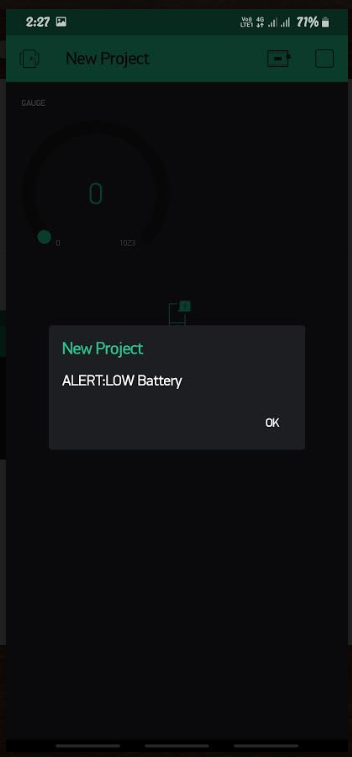
* Quick and appropriate action can be taken easily.
* Human effort can be reduced.
* Improved efficiency.
* Precise monitoring of battery pack.
* Can monitor remotely from anywhere.
* Give proper data history for all monitoring.
* Provide maintenance and controlling system

**8.2 SCREENSHOTS**

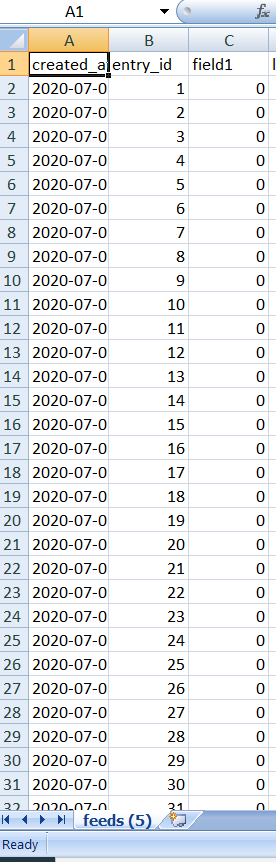


BLYNK

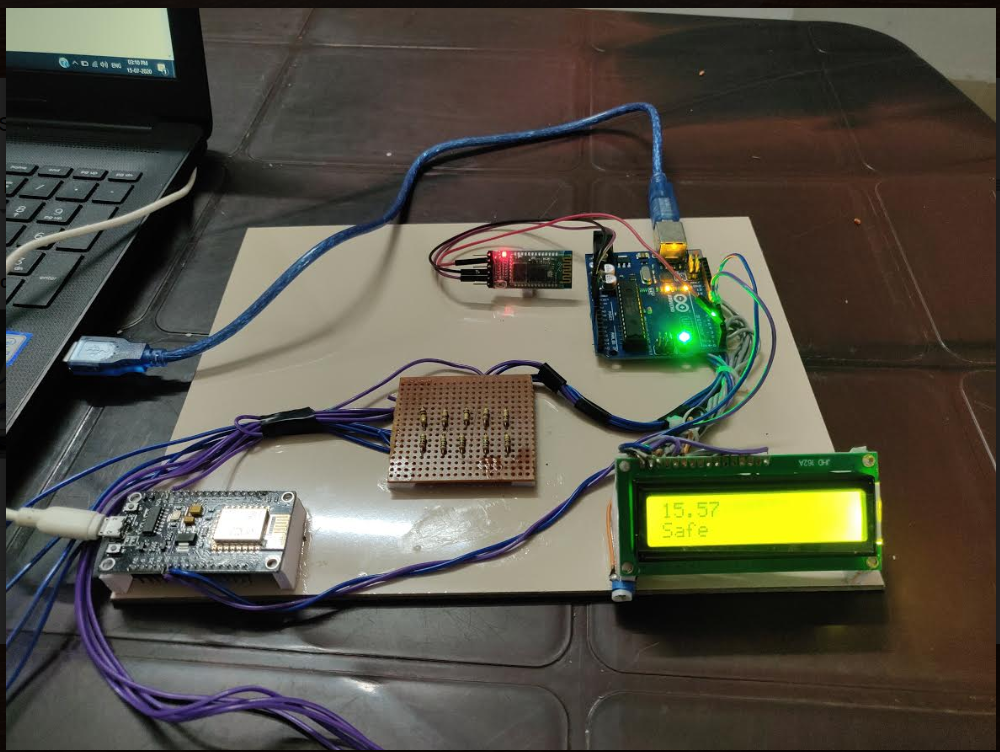
 

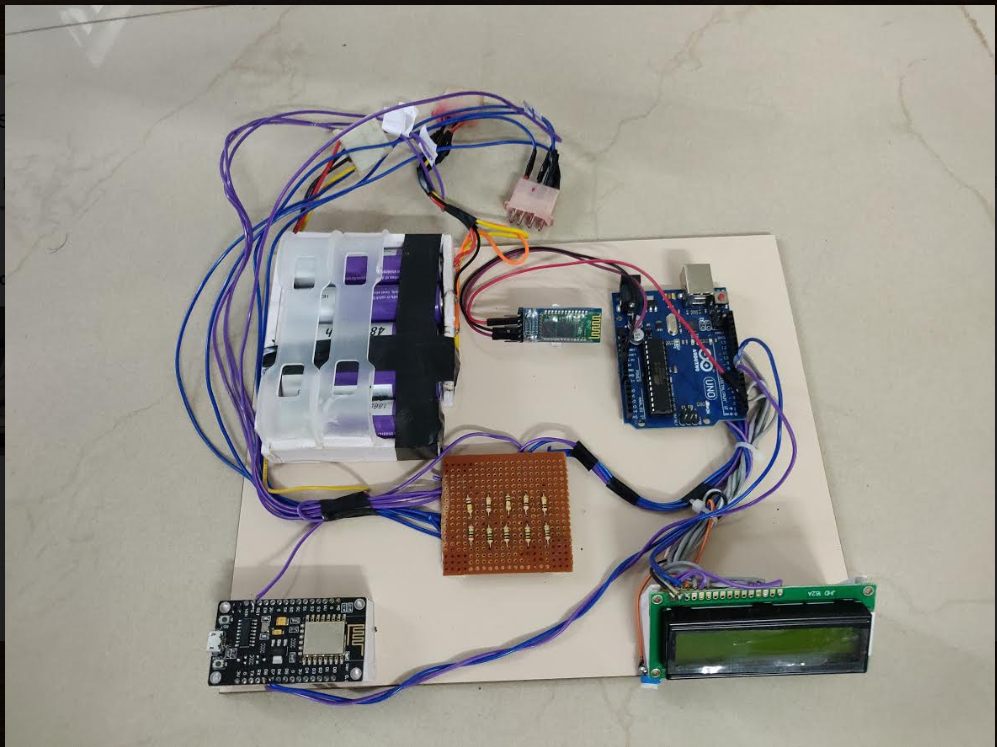
 

DATA HISTORY



**PROTOTYPE**

****

****

**CHAPTER 9**

**CONCLUSION**

**9.1 SYSTEM IMPLEMENTATION**

Implementation means converting a new design into iteration .During implementation there should be a strong interaction between the developer of the software and the users. Implementation involves installing hardware terminals and training the operating staff. In this phase, user training is critical for minimizing reluctance to change and giving the new system a chance to prove its worth. The new system may be totally new replacing the existing system, or it may be the modifications of existing system. In either case proper implementation is essential to provide a reliable system to meet organizational requirements.

**Major steps in the implementation of the system are as follows**:

Installation of the hardware required for “E-AUTOMOBS”, is required special hardware module which is to be installed for the working of both software and hardware. This can be worked on any system that support Ardunio IDE,ThingSpeak and a mobile device which support Blynk., Since the application is being developed with a server configured machine with a machine language. This case study is comparatively easy to implement. There are some technical risks where we can monitor Only one vehicle now since it is a. As a prerequisite you have to study about ThingSpeak,Basic of Ardunio, Ardunio coding,basic of electronics and working different sensors. other commonly used functionality will reduce our work and make the case study a beautiful one.

**9.2 CONCLUSION**

The implementation of system E-AUTOMOBS is a battery monitoring system based on Iot technology mainly focused on electric vehicles, that has an on screen LCD display which shows the current status of the battery pack, and warning notification can also be displayed another speciality is that there is an android app which shows the current status of the total voltage of the battery pack . For ease of servicing the battery, a Bluetooth device has been embedded into the device so, At service point there is no need to even touch the vehicle we can check using an android application designed based on a free platform, where we can monitor the individual cell.

A real time monitoring web platform has also been designed using ThingSpeak for the vehicle manufacturer to monitor the real time graphical data of the battery pack. The project is successfully tested for all the commands and it also ensure precise monitoring of battery pack by implementing as a central processing unit. Once the variation is detected, The system send notification to user as well as the vehicle manufacturer.

**9.3 FUTURE ENHANCEMENT**

* This project can be implemented in the form of small device which is easy to insert in electric vehicle
* This project can be integrated with additional features like various riding modes battery discharge control module
* Can implement dedicated Android app

**REFERENCE / BIBLIOGRAPHY**

**1. BOOKS**

* Fundamentals of software engineering

**2. IEEE PAPER**

2009 International Conference on Test and Measurement The Smart Battery Management System Xiao-feng WAN College ofInformation Engineering Nanchang University Jiangxi, China, Jian-ping WU, Hai-lin HU College ofInformation Engineering Nanchang University Jiangxi, China

**3.JOURNELS**

IoT-Based Battery Monitoring System for Electric Vehicle Mohd Helmy Abd Wahab\*,1,5 , Nur Imanina Mohamad Anuar1 , Radzi Ambar1 , Aslina Baharum2 , Shanoor Shanta1 , Mohd Suffian Sulaiman3 , Shukor Sanim Mohd Fauzi3 , Hafizul Fahri Hanafi4**, International Journal of Engineering & Technology, 7 (4.31) (2018) 505-510 International Journal of Engineering & Technology**

* Robotic Automation for Electric Vehicle Battery Assembly: Digital Factory Design and Simulation for the Electric Future of Mobility. AJIT SHARMA1 , PHILIP ZANOTTI2 , LAXMI P. MUSUNUR2 . 1Wayne State University, Detroit, MI 48104 2 Fanuc America Corporation, Rochester Hills, MI 48309
* Robotic Automation for Electric Vehicle Battery Assembly: Digital Factory Design and Simulation for the Electric Future of Mobility. AJIT SHARMA1 , PHILIP ZANOTTI2 , LAXMI P. MUSUNUR2 . 1Wayne State University, Detroit, MI 48104 2 Fanuc America Corporation, Rochester Hills, MI 48309

**4. WEBSITES**

<https://thingspeak.com/channels?tag=database>

<http://help.blynk.cc/en/articles/512105-how-to-install-blynk-library-for-arduino-ide>

**APPENDICES**

**1.SCRUM BOARD**

**i. Git**

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision-control system, it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

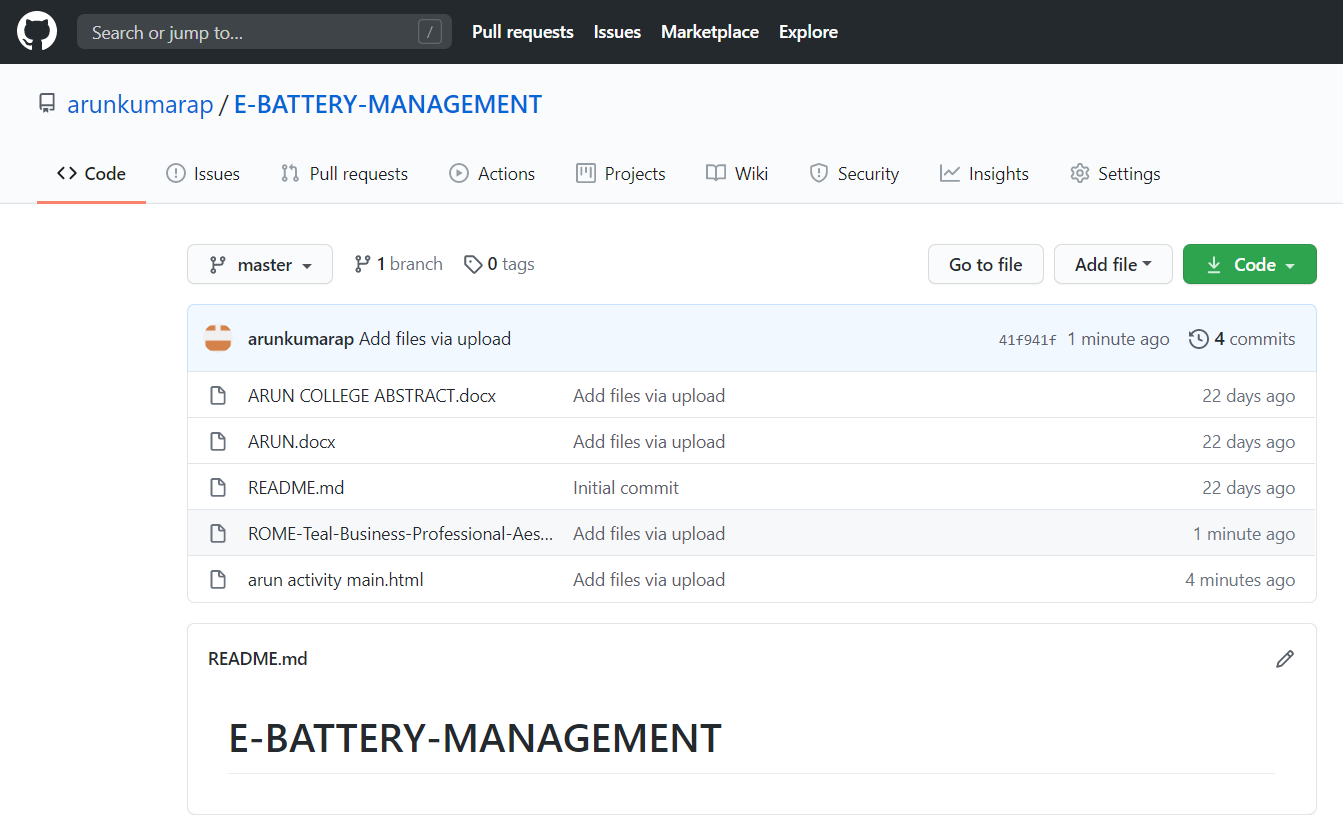
**ii. Git Repositories**

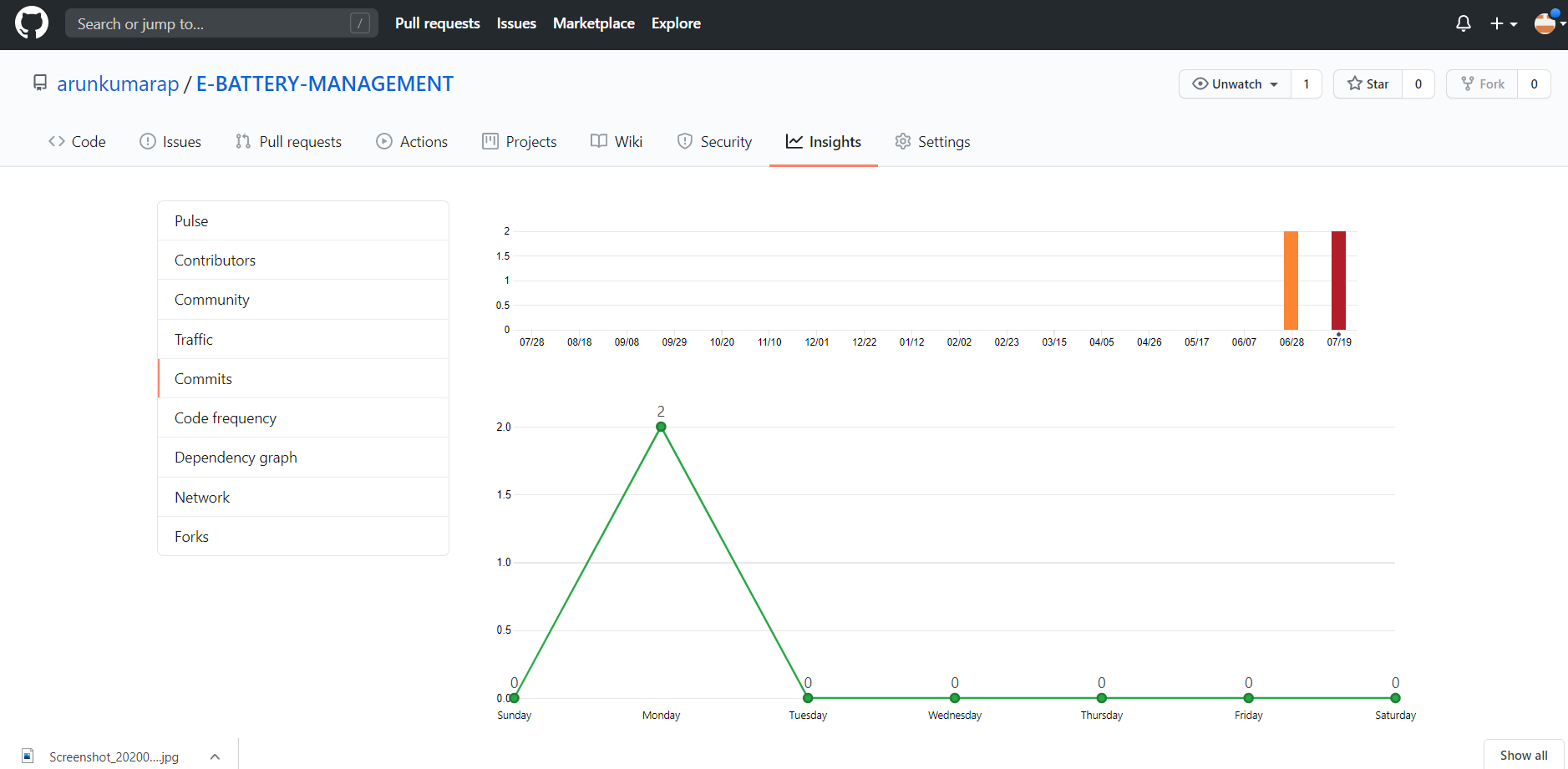
A Git repository contains the history of a collection of files starting from a certain directory. The process of copying an existing Git repository via the Git tooling is called cloning. After cloning a repository the user has the complete repository with its history on his local machine. Of course, Git also supports the creation of new repositories. If you want to delete a Git repository, you can simply delete the folder which contains the repository. If you clone a Git repository, by default, Git assumes that you want to work in this repository as a user

**iii. Scrum**

Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. Within agile development, Scrum teams are supported by two specific roles. The first is a Scrum Master, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level. The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

**iv. Git History**

****

****

**2. LIST OF TABLES**

|  |  |
| --- | --- |
| **Table** | **Description** |
| 5.1 | Input / Output Design |
| 7.1 | Test Result |

**3. LIST OF FIGURES**

|  |  |
| --- | --- |
| **Figures** | **Description** |
| 4.1 | NODE MCU |
| 4.2 |  |
| 4.3 |  |
| 4.4 |  |
| 5.1 | Block Diagram |

**4. ABBREVIATIONS AND NOTATION**

|  |  |
| --- | --- |
| **Notation** | **Description** |
| Node MCU | Node Microcontroller |
| IOT | Internet of Things |

**CODE**

**Think speak and blynk activation**

#include <ESP8266WiFi.h> // ESP8266WiFi.h library

 #include <BlynkSimpleEsp8266.h>

const char\* ssid     = "OnePlus 6T";// replace subscribe with your WiFi SSID(Name)

const char\* password = "arkgroups";//replace with Your Wifi Password name

const char\* host = "[api.thingspeak.com](http://api.thingspeak.com/)";

const char\* writeAPIKey = "I9DF9C70AV3KO6A9"; //copy yout ThingSpeak channel API Key.

#include <SoftwareSerial.h>

SoftwareSerial s(D7,D8);//rx tx

float data;

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;

char auth[] = "kQnVap4P8GvzH4snp1NuhfS8lQJbLzSA";

void setup() {

  Blynk.begin(auth, ssid, password);

  s.begin(9600);

// Initialize sensor

 Serial.begin(115200);

delay(1000);

 Serial.println("Connecting to ");

       Serial.println(ssid);

//  Connect to WiFi network

  WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

    Serial.print(".");

  }

   Serial.println("");

   Serial.println("WiFi connected");

}

void loop() {

  Blynk.run();

  timer.run();

s.write("s");

  if (s.available()>0)

  {

    data=s.read();

    Serial.println(data);

    Blynk.virtualWrite(V1,data);

  }

  if(data<5)

  {

    Blynk.notify("ALERT:LOW Battery");

  }

// make TCP connections

  WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

return;

  }

  String url = "/update?key=";

  url+=writeAPIKey;

  url+="&field1=";

  url+=String(data);

  url+="\r\n";

// Request to the server

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

Serial.println(data);

  Serial.println("Send to ThingSpeak.\n");

client.stop();

  Serial.println("Wait for 15 sec to update next datapack in thingSpeak");

delay(1000);

}

sLCD DISPLAY program

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#define NUM\_SAMPLESbattery1 10

int sumbattery1 = 0;                    // sum of samples taken BT1

unsigned char sample\_countbattery1 = 0; // current sample number

float voltagebattery1 = 0.0;            // calculated voltage

#define NUM\_SAMPLESbattery2 10

int sumbattery2 = 0;                    // sum of samples taken BT2

unsigned char sample\_countbattery2 = 0; // current sample number

float voltagebattery2 = 0.0;            // calculated voltage

#define NUM\_SAMPLESbattery3 10

int sumbattery3 = 0;                    // sum of samples taken BT3

unsigned char sample\_countbattery3 = 0; // current sample number

float voltagebattery3 = 0.0;            // calculated voltage

#define NUM\_SAMPLESbattery4 10

int sumbattery4 = 0;                    // sum of samples taken BT3

unsigned char sample\_countbattery4 = 0; // current sample number

float voltagebattery4 = 0.0;            // calculated voltage

void setup()

{ lcd.begin(16, 2);

    Serial.begin(9600);

}

void loop()

{

    while (sample\_countbattery1 < NUM\_SAMPLESbattery1) {

        sumbattery1 += analogRead(A0);//B1

        sample\_countbattery1++;

        delay(10);

    }

    voltagebattery1 = ((float)sumbattery1 / (float)NUM\_SAMPLESbattery1 \* 5.015) / 1024.0;

    sample\_countbattery1 = 0;

    sumbattery1 = 0;

 while (sample\_countbattery2< NUM\_SAMPLESbattery2) {

        sumbattery2 += analogRead(A1);//B2

        sample\_countbattery2++;

        delay(10);

    }

    voltagebattery2 = ((float)sumbattery2 / (float)NUM\_SAMPLESbattery2 \* 5.015) / 1024.0;

    sample\_countbattery2 = 0;

    sumbattery2 = 0;

 while (sample\_countbattery3 < NUM\_SAMPLESbattery3) {

        sumbattery3 += analogRead(A2);//B3

        sample\_countbattery3++;

        delay(10);

    }

    voltagebattery3 = ((float)sumbattery3 / (float)NUM\_SAMPLESbattery3 \* 5.015) / 1024.0;

    sample\_countbattery3 = 0;

    sumbattery3 = 0;

 while (sample\_countbattery4 < NUM\_SAMPLESbattery4) {

        sumbattery4 += analogRead(A3);//B4

        sample\_countbattery4++;

        delay(10);

    }

    voltagebattery4 = ((float)sumbattery4 / (float)NUM\_SAMPLESbattery4 \* 5.015) / 1024.0;

    sample\_countbattery4 = 0;

    sumbattery4 = 0;

 Serial.print(voltagebattery1 \* 11.132);

   Serial.println ("Battery 1 V");

 Serial.print(voltagebattery2 \* 11.132);

   Serial.println ("Battery 2 V");

 Serial.print(voltagebattery3 \* 11.132);

   Serial.println ("Battery 3 V");

 Serial.print(voltagebattery4 \* 11.132);

   Serial.println ("Battery 4 V");

  float total=(voltagebattery1 \* 11.132+voltagebattery2 \* 11.132+voltagebattery3 \* 11.132+voltagebattery4 \* 11.132);

Serial.print(total);

   Serial.println ("Total Battery V");

  lcd.setCursor(0,0);

   lcd.print(total);

Serial.println ("\n");

delay(1000);

if(total<=5)

{

  lcd.setCursor(0,1);

   lcd.print("Unsafe         ");

   Serial.println ("Unsafe Service....      ");

}

else

{

 lcd.setCursor(0,1);

   lcd.print("Safe       ");

   Serial.println ("Safe");

}

}