VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

"IOT Based Anti-Poaching of Trees and Protection of Forest"

Thesis submitted in partial fulfillment of the curriculum prescribed for the award of the degree of Bachelor of Engineering in Computer Science & Engineering by

1CR14CS161	Vishal S
1CR14TE059	Roshini Sultana
1CR14CS062	Keerthi Reddy
1CR14CS084	Monica Raju

Under the Guidance of

Shivaraj V B
Assistant Professor
Department of CSE, CMRIT, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
#132, AECS LAYOUT, IT PARK ROAD, BENGALURU - 560037

2017-18

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

Certificate

This is to certify that the project entitled **"IOT Based Anti-Poaching of Trees and Protection of Forest"** is a bonafide work carried out by **Vishal S** bearing **USN:1CR14CS161**, **Roshini Sultana** bearing **USN:1CR14TE059**, **Keerthi Reddy** bearing **USN:1CR14CS062**, and **Monica Raju** bearing **USN:1CR14CS084** in partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2017-18. It is certified that all corrections / suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

-------------------------	-------------------------	-------------------------
Signature of Guide	Signature of HoD	Signature of Principal
Shivaraj V B	**Dr. Jhansi Rani P**	**Dr. Sanjay Jain**
Assistant Professor	Professor & Head	Principal
Department of CSE	Department of CSE	CMRIT,
CMRIT, Bengaluru - 37	CMRIT, Bengaluru - 37	Bengaluru - 37

External Viva

	Name of the Examiners	Institution	Signature with Date
1.	-------------------------	----------------------------	----------------------------
2.	-------------------------	----------------------------	----------------------------

Acknowledgement

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So with gratitude I acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

I would like to thank **Dr.Jhansi Rani P**, Professor and HOD, Department of Computer Science and Engineering, who shared her opinion and experience through which I received the required information crucial for the project.

I consider it a privilege and honour to express my sincere gratitude to my guide **Shivaraj V B**, Assistant Professor, Department of Computer Science and Engineering, for his valuable guidance throughout the tenure of this review.

I consider it a privilege and honour to express my sincere gratitude to Project coordinator **Sudhakar K N**, Associate Professor, Department of Computer Science Engineering for his valuable guidance throughout the tenure of this review.

Finally I would like to thank all my family members and friends whose encouragement and support was invaluable.

Vishal S(1CR14CS161)
Roshini Sultana(1CR14TE059)
Keerthi Reddy(1CR14CS062)
Monica Raju(1CR14CS084)

Table of Contents

List of Figures

List of Tables

Abstract

Smuggling/theft of most important trees such as sandal wood in forests, poses a serious threat to forest resources, causes significant economic damage and ultimately has quite a devastating effect on the environment all over the world. This paper proposes a microcontroller based anti-poaching system employing WSN technology, which is capable of detecting theft by monitoring the vibrations produced by the cutting of trees/branches using a 3 axis MEMS accelerometer.

A microcontroller is used along with PC so that the information can be uploaded. WSN is widely used technology in remote monitoring applications. The embedded system architecture and the hardware/software designs are described in detail. Vibration data collected by various tests on wood and simulated using Arduino IDE.

Chapter 1

PREAMBLE

1.1 INTRODUCTION

In recent years poaching or smuggling of environmentally and economically important species of trees in forested areas- such as Sandalwood, Teakwood, Pine and Rosewood has been tremendously increased. There have been several initiatives undertaken by different stakeholders– and in particular by the Govt. of India, to mitigate these problems. These include the recruitment, training and deployment of anti-poaching watchers and/or private/govt. security guards across forests. Strict punishments for convicted offenders, as well as giving special incentives for anti-poaching activities (Twelfth Five Year Plan 2012-2017) were aimed for eradicating the menace.

The main idea presented in this paper is to design a portable wireless sensor node which is a part of wireless sensor Network. It will be mounted on trunk of each tree, capable of detecting theft as well as automatically initiate send alarm signals if any to remote terminal through wireless media.

1.2 EXISTING SYSTEM

The existing system consists of hiring security personals for monitoring the entire area for suspicious activity , However due to physical limitations in human it is hard to monitor the entire area continuously ,thus hiring of guards proves unreliable and inadequate. Another existing system is the installation of CCTV cameras for covering large area proves very costly and is hard to implement. Also ,the latest trend for protection of trees is to tag an RF-ID to trees just like tagging an animal for knowing the whereabouts of a particular tree. However ,this technology does not give the real time information while the activity is happening. Activity is detected only when the tree leaves its initial position

1.3 PROPOSED SYSTEM

The main idea presented in this paper is to design a portable wireless sensor node which will be a part of a Wireless Sensor Network. The suggested system will consist of two modules one involving sensors and controller module which will be at tree spot and another one is Android phone. We built an application which will continuously receive sensor data. This is an IOT based project where we upload sensor data continuously to cloud.Tilt sensor is used to determine whether the tree is cut down or not similarly temperature sensor is used to determine whether the forest is on fire or not. Using built application we can turn on the water pump in case of forest fire and we can turn on alarm other devices in smuggling case. This system is like from anywhere we can monitor and control by using GSM Technology. Which will convey the message from monitor station to the control station vise versa using wireless technology.

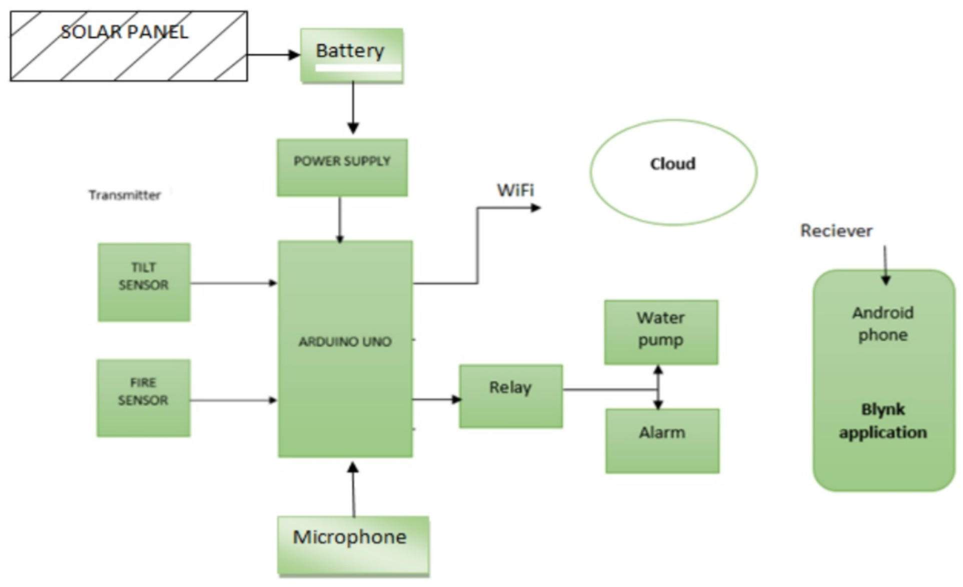


Figure 1.1: Anti Poaching Architecture

1.4 PROBLEM STATEMENT

- Illegal logging refers to what in forestry might be called timber theft by the timber mafia.
- It refers to the harvesting , transportation, purchase or sale of timber in violation of laws.
- Selective logging almost often diseased or malformed trees.
- As a preventive measure to the above problem , we have come up with a system based on IoT that can be used to avoid the smuggling of the trees which would stop the deforestation

Chapter 2

LITERATURE SURVEY

2.1 INTRODUCTION

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

2.2 LITERATURE SURVEY

Literature survey is the documentation of a comprehensive review of the published and unpublished work from secondary sources data in the areas of specific interest to the researcher. The library is a rich storage base for secondary data and researchers used to spend several weeks and sometimes months going through books, journals, newspapers, magazines, conference proceedings, doctoral dissertations, masters theses, government publications and financial reports to find information on their research topic. Reviewing the literature on the topic area at this time helps the researcher to focus further interviews more meaningfully on certain aspects found to be important is the published studies even if these had not surfaced during the earlier questioning .So the literature survey is important for gathering the secondary data for the research which might be proved very helpful in the research. The literature survey can be conducted for several reasons. The literature review can be in any area of the business.

2.3 PAPER 1

Title: Anil Kulkarni, Ajay Khandare, Mandar Malve, "Wireless Sensor Network (WSN) for protection high cost trees in remote jungles from fire and poaching", International Seminar on Sandalwood: Current Trends and Future Prospects
Context:
Wildlife prevention has become an important practice due to negative effects of human activities such as cutting of trees on large scale and unregulated hunting which causes major threat to wildlife. So we are going to introduce the project on prevention of trees and wildlife in forest. This article presents the design of a system for detection of vibration for prevention of cutting of trees, detection of temperature for prevention of forest fires also detection of pulses of animal for prevention wildlife using wireless sensor networks to prevent a disaster (forest) that could lead to loss of a significant number of natural resources. In this project, The sensing device can sense the

vibration, pulse, and temperature, and then sent them over zig-bee networks to forest office. To save the transmission cost, we also sent the GPS location information simultaneously. Here we use Wireless Sensor Networks (WSNs).In this network numerous sensors are usually deployed on remote places, the deployment and maintenance must be easy and scalable. Wireless sensor network is the network which consists of large number of small nodes. Sensor nodes are great for deployment in hostile environments or over large geographical areas.

2.4 PAPER 2

Title: Sridevi Veerasingam, Saurabh Karodi, Sapna Shukla, "Design of Wireless Sensor Network node on Zigbee for Temperature Monitoring"
Context:
In this paper a portable wireless data logging system for temperature monitoring in real time process dynamics. Process variables (like temperature, pressure, flow, level) vary with time in certain applications and these variations should be recorded so that a control action can take place at a defined set point. This paper proposes a 8-bit embedded platform for a temperature sensor node having a network interface using the 802.15.4 ZigBee protocol, that is a wireless technology developed as open global standard to address the low-cost, low-power wireless sensor networks. The wireless temperature sensor node senses and transmits the variations in the local temperature to the central computing unit placed within the range. The central base station receives the data and stores it in the file and plotting the variations simultaneously.

2.5 PAPER 3

Title: Manish Y. Upadhye, P. B. Borole, Ashok K. Sharma, "Real-Time Wireless Vibration Monitoring System Using LabVIEW"
Context:
Vibration analysis provides relevant information about abnormal working condition of machine parts. Vibration measurement is prerequisite for vibration analysis which is used for condition monitoring of machinery. Also, wireless vibration monitoring has many advantages over wired monitoring. This Paper presents, implementation of a reliable and low cost wireless vibration monitoring system. Vibration measurement has been done using 3-Axis digital output MEMS Accelerometer sensor. This sensor can sense vibrations in the range 0.0156g to 8g where, 1g is 9.81m/s2. Accelerometer Sensor is interfaced with Arduino-derived microcontroller board having Atmel's AT-mega328p microcontroller. The implemented system uses ZigBee communication

protocol i.e. standard IEEE 802.15.4, for wireless communication between Sensor Unit and Vibration Monitoring Unit. The wireless communication has been done using XBee RF modules. National Instruments's LabVIEW software has been used for development of graphical user interface, data-logging and alarm indication on the PC. Experimental results show continuous real-time monitoring of machine's vibrations on charts. These results, along with data-log file have been used for vibration analysis. This analysis is used to ensure safe working condition of machinery and used in predictive maintenance.

Chapter 3

THEORETICAL BACKGROUND

3.1 INTRODUCTION

3.2 ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Arduino is a computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

The project's board designs use a variety of microprocessors and controllers. These systems provide sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. The microcontrollers are mainly programmed using a dialect of features from the programming languages C and C++.

In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy,[2] aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worring too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

”Uno” means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

3.2.1 Why Are We Using Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia’s BX-24, Phidgets, MIT’s Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than 50 dollars
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming software** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

3.2.2 Technical Specification

Microcontroller	ATMega328P
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage(limit)	6-20V
Digital I/O Pins	14(of which 6 provide PWN output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pins	20 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by boot-loader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 3.1: Specification of Arduino

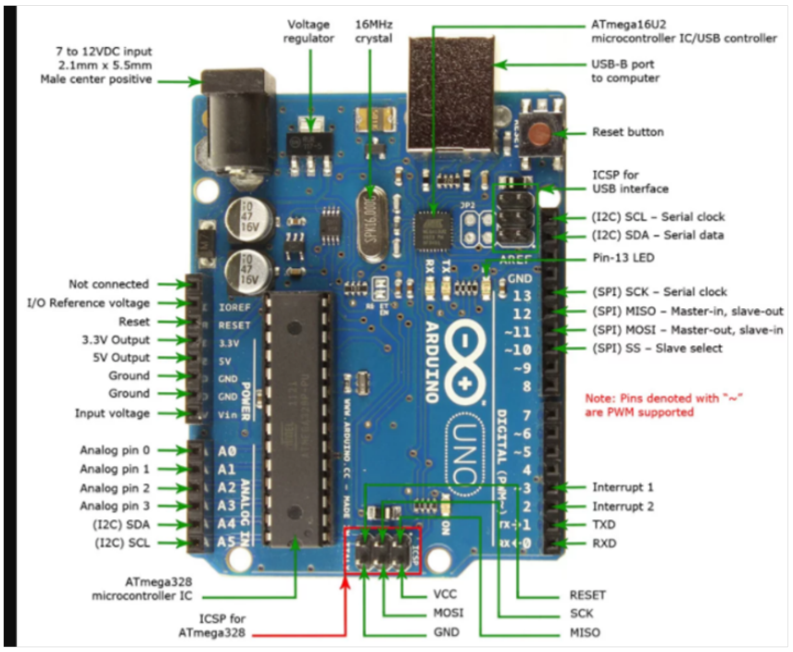


Figure 3.1: ATMega328P

3.3 INTERNET OF THINGS

IoT (Internet of Things) is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.

IoT systems have applications across industries through their unique flexibility and ability to be suitable in any environment. They enhance data collection, automation, operations, and much more through smart devices and powerful enabling technology.

IoT systems allow users to achieve deeper automation, analysis, and integration within a system. They improve the reach of these areas and their accuracy. IoT utilizes existing and emerging technology for sensing, networking, and robotics.

IoT exploits recent advances in software, falling hardware prices, and modern attitudes towards technology. Its new and advanced elements bring major changes in the delivery of products, goods, and services; and the social, economic, and political impact of those changes.

3.3.1 Features of IOT

The most important features of IoT include artificial intelligence, connectivity, sensors, active engagement, and small device use. A brief review of these features is given below

- AI IoT essentially makes virtually anything smart, meaning it enhances every aspect of life with the power of data collection, artificial intelligence algorithms, and networks. This can mean something as simple as enhancing your refrigerator and cabinets to detect when milk and your favorite cereal run low, and to then place an order with your preferred grocer.
- Connectivity New enabling technologies for networking, and specifically IoT networking, mean networks are no longer exclusively tied to major providers. Networks can exist on a much smaller and cheaper scale while still being practical. IoT creates these small networks between its system devices.
- Sensors IoT loses its distinction without sensors. They act as defining instruments which transform IoT from a standard passive network of devices into an active system capable of real-world integration.
- Active Engagement Much of today's interaction with connected technology happens through passive engagement. IoT introduces a new paradigm for active content, product, or service engagement.

- Small Devices Devices, as predicted, have become smaller, cheaper, and more powerful over time. IoT exploits purpose-built small devices to deliver its precision, scalability, and versatility.

3.3.2 IOT Advantages

The advantages of IoT span across every area of lifestyle and business. Here is a list of some of the advantages that IoT has to offer

- Improved Customer Engagement Current analytics suffer from blind-spots and significant flaws in accuracy; and as noted, engagement remains passive. IoT completely transforms this to achieve richer and more effective engagement with audiences.

- Technology Optimization The same technologies and data which improve the customer experience also improve device use, and aid in more potent improvements to technology. IoT unlocks a world of critical functional and field data.

- Reduced Waste IoT makes areas of improvement clear. Current analytics give us superficial insight, but IoT provides real-world information leading to more effective management of resources.

- Enhanced Data Collection Modern data collection suffers from its limitations and its design for passive use. IoT breaks it out of those spaces, and places it exactly where humans really want to go to analyze our world. It allows an accurate picture of everything.

3.3.3 IOT Software

IoT software addresses its key areas of networking and action through platforms, embedded systems, partner systems, and middleware. These individual and master applications are responsible for data collection, device integration, real-time analytics, and application and process extension within the IoT network. They exploit integration with critical business systems (e.g., ordering systems, robotics, scheduling, and more) in the execution of related tasks.

- Data Collection

  This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine-to-machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also works in reverse by distributing data over devices. The system eventually transmits all collected data to a central server.

- Device Integration

  Software supporting integration binds (dependent relationships) all system devices to create the body of the IoT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IoT network because without them, it is not an IoT system. They manage the various applications, protocols, and limitations of each device to allow communication.

- Real-Time Analytics

  These applications take data or input from various devices and convert it into viable actions or clear patterns for human analysis. They analyze information based on various settings and designs in order to perform automation-related tasks or provide the data required by industry.

- Application and Process Extension

  These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection.

3.3.4 IOT Technology and Protocols

IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low-energy Bluetooth, low-energy wireless, low-energy radio protocols, LTE-A, and WiFi-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems.

- NFC and RFID

  RFID (radio-frequency identification) and NFC (near-field communication) provide simple, lowenergy, and versatile options for identity and access tokens, connection bootstrapping, and payments.

  - RFID technology employs 2-way radio transmitter-receivers to identify and track tags associated with objects.
  - NFC consists of communication protocols for electronic devices, typically a mobile device and a standard device.

  - Low-Energy Bluetooth

    This technology supports the low-power, long-use need of IoT function while exploiting a standard technology with native support across systems.

- Radio Protocols

  ZigBee, Z-Wave, and Thread are radio protocols for creating low-rate private area networks. These technologies are low-power, but offer high throughput unlike many similar options. This increases the power of small local device networks without the typical costs.

- LTE-A

  LTE-A, or LTE Advanced, delivers an important upgrade to LTE technology by increasing not only its coverage, but also reducing its latency and raising its throughput. It gives IoT a tremendous power through expanding its range, with its most significant applications being vehicle, UAV, and similar communication.

- WiFi-Direct

  WiFi-Direct eliminates the need for an access point. It allows P2P (peer-to-peer) connections with the speed of WiFi, but with lower latency. WiFi-Direct eliminates an element of a network that often bogs it down, and it does not compromise on speed or throughput.

3.3.5 ARCHITECTURE OF IOT

IoT is a three layer architecture. The layers include:

1. Perception Layer
2. Network Layer
3. Application Layer

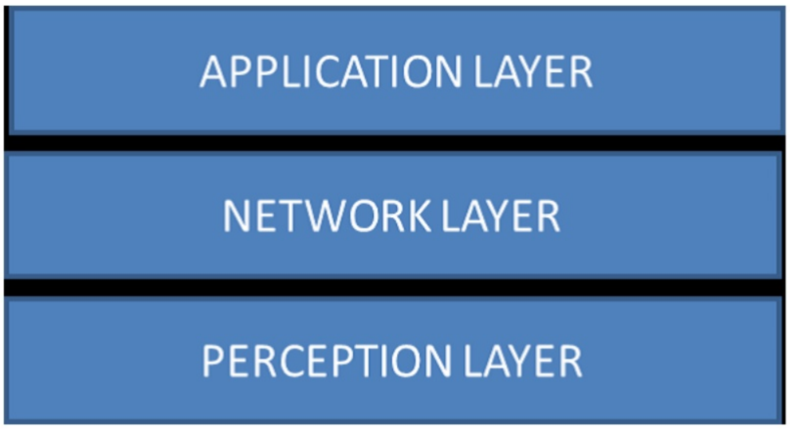


Figure 3.2: Internet of Things Architecture

3.3.6 Perception Layer

It is the first layer of IoT architecture.It is mainly used for identifying objects and collecting information. It is tied to the hardware device like a GPS, sensor, RFID tags, or sensor network and linked to any intelligent system. It also called physical layer as the information from the physical devices is changed into a digital signal that is suitable for network transmission. The primary work of this layer is to gather information from the sensing technology

3.3.7 Network Layer

It is a second layer of the IOT architecture .Its main function is to conduct and obtain data or information. It is a network management center for IoT. It gains data or information from the perception layer that has been collected and transferred to different networks via wired or wireless network. It also transfers huge amount of data between dissimilar networks.

3.3.8 Application Layer

It is a third layer of the IoT architecture. It ties the application to the network. The application layer uses the processed data sent by the network Layer. In fact, this layer constitutes the front end of the whole IoT architecture through which IoT potential will be exploited.

3.4 SENSORS

A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display. These sensors send their output to micro-controllers

3.5 TEMPERATURE SENSOR

An analog temperature sensor is pretty easy to explain, it's a chip that tells you what the ambient temperature is!

These sensors use a solid-state technique to determine the temperature. That is to say, they don't use mercury (like old thermometers), bimetallic strips (like in some home thermometers or stoves), nor do they use thermistors (temperature sensitive resistors). Instead, they use the fact as temperature increases, the voltage across a diode increases at a known rate. (Technically, this is actually the voltage drop between the base and emitter - the Vbe - of a transistor. By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature. There have been some improvements on the technique but, essentially that is how temperature is measured.

Because these sensors have no moving parts, they are precise, never wear out, don't need calibration, work under many environmental conditions, and are consistent between sensors and readings. Moreover they are very inexpensive and quite easy to use.

3.5.1 Types of Temperature Sensor

There are two temperature sensing methods:

- Contact
- Non-Contact

3.5.1.1 Contact Temperature Sensor

These types of temperature sensor are required to be in physical contact with the object being sensed and use conduction to monitor changes in temperature. They can be used to detect solids, liquids or gases over a wide range of temperatures.

3.5.1.2 Non-Contact Temperature Sensor

These types of temperature sensor use convection and radiation to monitor changes in temperature. They can be used to detect liquids and gases that emit radiant energy as heat rises and cold settles to the bottom in convection currents or detect the radiant energy being transmitted from an object in the form of infra-red radiation (the sun).

3.5.2 LM35 Sensor Specifications

The LM35 series are precision integrated-circuit LM35 temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 sensor thus has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 sensor does not require any external calibration or trimming to provide typical accuracies of C at room temperature and C over a full -55 to +150C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 A from its supply, it has very low self-heating, less than 0.1C in still air. The LM35 is rated to operate over a -55 to +150C temperature range, while the LM35C sensor is rated for a -40 to +110C range (-10 with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D sensor is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

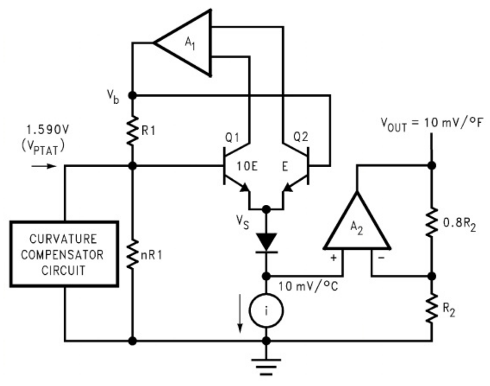


Figure 3.3: LM35 Sensor Circuit Schematic

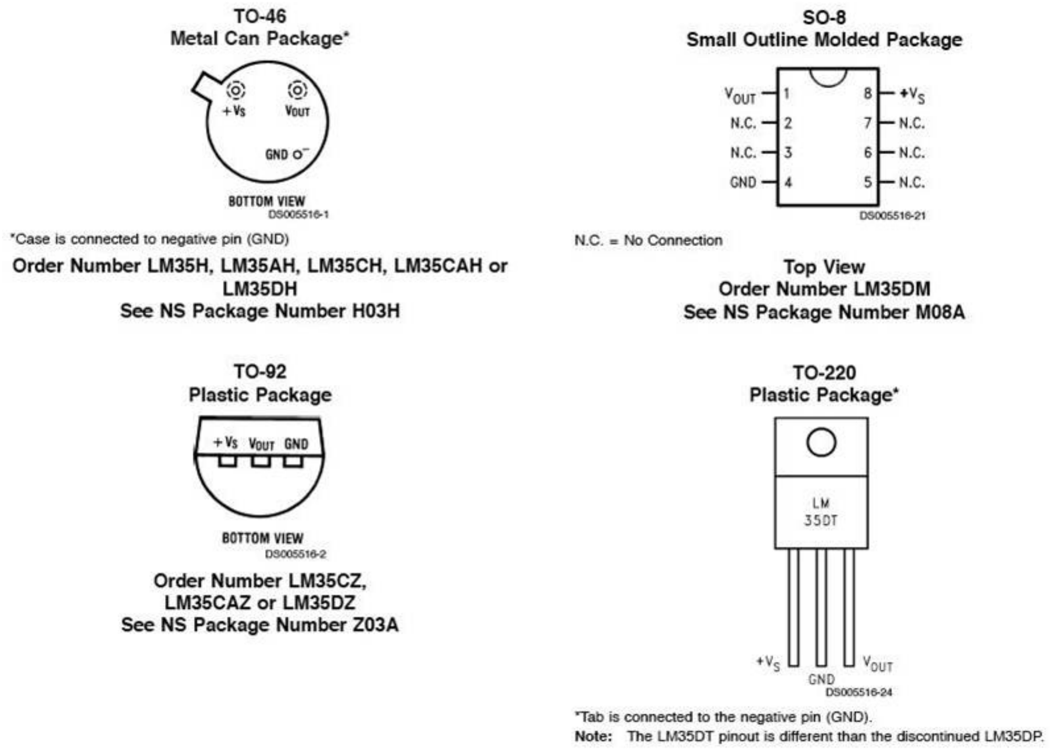

TO-46
Metal Can Package*
+Vs
VOUT
GND
BOTTOM VIEW
DS005516-1
*Case is connected to negative pin (GND)
Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH
See NS Package Number H03H
SO-8
Small Outline Molded Package
VOUT 1
N.C. 2
N.C. 3
GND 4
8 +VS
7 N.C.
6 N.C.
5 N.C.
DS005516-21
N.C. = No Connection
Top View
Order Number LM35DM
See NS Package Number M08A
TO-92
Plastic Package
+Vs VOUT GND
BOTTOM VIEW
DS005516-2
Order Number LM35CZ, LM35CAZ or LM35DZ
See NS Package Number Z03A
TO-220
Plastic Package*
LM 35DT
+VS
GND
VOUT
DS005516-24
*Tab is connected to the negative pin (GND).
Note: The LM35DT pinout is different than the discontinued LM35DP.

Figure 3.4: LM35 Sensor Pinouts and Packaging

3.5.3 LM35 Sensor Background and Applications

Most commonly-used electrical temperature sensors are difficult to apply. For example, thermocouples have low output levels and require cold junction compensation. Thermistors are nonlinear. In addition, the outputs of these sensors are not linearly proportional to any temperature scale. Early monolithic sensors, such as the LM3911, LM134 and LM135, overcame many of these difficulties, but their outputs are related to the Kelvin temperature scale rather than the more popular Celsius and Fahrenheit scales. Fortunately, in 1983 two I.C.s, the LM34 Precision Fahrenheit Temperature Sensor and the LM35 Precision Celsius Temperature Sensor, were introduced. This application note will discuss the LM34, but with the proper scaling factors can easily be adapted to the LM35.

The LM35/LM34 has an output of 10 mV/F with a typical nonlinearity of only 0.35F over a 50 to +300F temperature range, and is accurate to within 0.4F typically at room temperature (77F). The LM34s low output impedance and linear output characteristic make interfacing with readout or control circuitry easy. An inherent strength of the LM34 sensor over other currently available temperature sensors is that it is not as susceptible to large errors in its output from low level leakage currents. For instance, many monolithic temperature sensors have an output of only 1 A/K. This leads to a 1K error for only 1 -Ampere of leakage current. On the other hand, the LM34 sensor may be operated as a current mode device providing 20 A/F of output current. The same 1 A of leakage current will cause an error in the LM34s output of only 0.05F (or 0.03K after scaling).

Low cost and high accuracy are maintained by performing trimming and calibration procedures at the wafer level. The device may be operated with either single or dual supplies. With less than 70 A of current drain, the LM34 sensor has very little self-heating (less than 0.2F in still air), and comes in a TO-46 metal can package, a SO-8 small outline package and a TO-92 plastic package.

The LM35/LM34 is a versatile device which may be used for a wide variety of applications, including oven controllers and remote temperature sensing. The device is easy to use (there are only three terminals) and will be within 0.02F of a surface to which it is either glued or cemented. The TO-46 package allows the user to solder the sensor to a metal surface, but in doing so; the GND pin will be at the same potential as that metal. For applications where a steady reading is desired despite small changes in temperature, the user can solder the TO-46 package to a thermal mass. Conversely, the thermal time constant may be decreased to speed up response time by soldering the sensor to a small heat fin.

3.6 TILT SENSOR

The Paris Air show is one of the biggest platforms for various airplane manufacturers to showcase the swiftness of their planes. Some of the valiant pilots of the world perform amazing stunts. These pilots are supported by the power of fast computing machinery in their airplanes. One of the critical parts of this computer assisted circuitry is the tilt sensor. A type of transducer, tilt sensor aids in giving information about the vertical as well as horizontal inclination of the airplane so that the pilot can understand how can he tackle the obstacles during the flight and perform the stunts. Keeping the pilots informed about the current orientation of the plane, the angle at which they are inclined to earths surface, tilt sensors play a very important role in decision making for the pilots. These types of transducer produce an electric signal proportional to the degree of inclination with respect one or multiple axes. This article will detail more about tilt sensor, the types of tilt sensors, need and their applications. Lets dig in deep to know about tilt sensors.

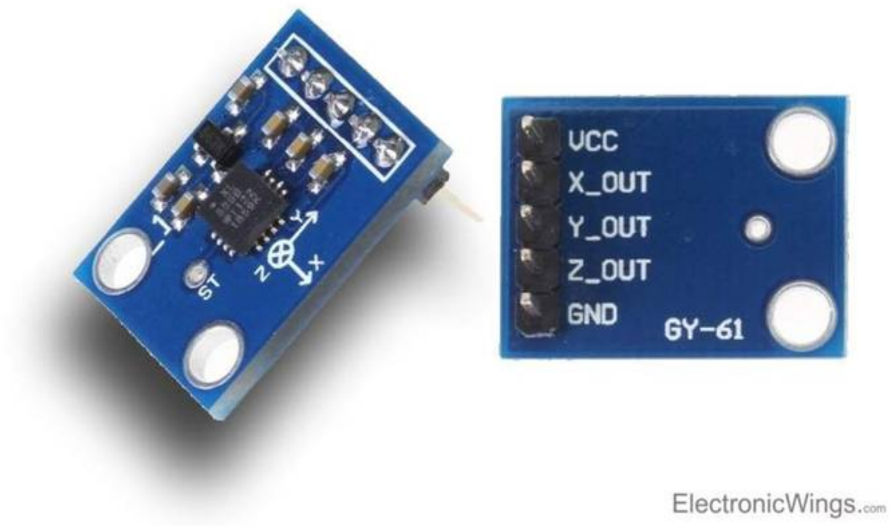


Figure 3.5: Tilt Sensor

3.6.1 Need and Applications of Tilt Sensor

Tilt sensors play a vital role in numerous applications.Diversely popular in multiple fields, tilt sensors are needed in:

- Portable Computers: A popular advertisement about a portable computer says There is no wrong way or right way you hold the device. This phrase means that the display of that computer aligns itself in the manner user is holding it and he can tiltit the way he wants. Tilt sensors are required to keep the device display in the correct position with the way user holds it. Often, motion based gaming devices use this application of tilt sensor or tilt sensor derived accelerometer.
- Vehicular Security Systems: Several vehicle security alarm systems are based on tilt sensors. In the cases of unauthorized vehicle towing or movement of steering, tilt sensor engages the alarm. This in turn produces an electric signal and the alarm starts sounding.

- Aviation: In airplanes and helicopters, tilt sensors, along with inclinometers form an altitude monitoring system through which the pilot can monitor the inclinations of the flying machine.

- Robotics: For any type of robot, balance is one of the most important criteria to be taken care of. Whenever a robot inclines to any direction, tilt sensor aids in giving details about corresponding incline in form of electrical signals. It thus makes the robot judge on its own that whether it is supposed to get aligned or not.

3.6.2 Specification of a Tilt Sensor

A few structural and working configurations form the basis of manufacturing and use of tilt sensors. A few distinct specifications are mentioned below:

1. Number of Axes: The total number of axes upon which the sensor can be expected to respond when actuated is an important factor. While some applications require just a single axis, certain applications such as robotics do require a dual axes tilt sensor. Video game controllers and joysticks are often 3 axes tilt sensor based so that easy movement of the character is there.

2. Resolution: The minimum inclination that can be detected by the sensor is termed as resolution. A good resolution sensor should be able to measure as small inclinations as possible. Electrolytic tilt sensors have a high resolution than the MEMS based ones. For instance, a sensor that can respond to 0.5degree resolution would have a higher resolution that the one which requires a tilt of 1.0degrees.

3. Sensitivity: Another important term is the ability of the sensor to respond to small changes which can be termed as sensitivity. Sensitivity plays a crucial role in those areas where a degree more or less of inclination can make big changes such as in airplanes or high speed trains.

4. Measuring Range: The range of inclination that the sensor can respond to is also a crucial factor. While some sensors are used for a few degree measurements like +/-10, others are required to work at a versatile range of over +/-60.

5. Noise Tolerance: Noise can produce harmonic distortions in the sensor working that might result in output variation and reduce the system efficiency. Noise levels are mentioned by the manufacturer so that the system is accordingly set up.

6. Output: Depending upon the system requirements, the electrical outputs of these sensors can be utilized by the sensors in several manners such as frequency change or even simple current or voltage based indication purposes. Also, output ports can vary as per the system they are used into.

7. Vibration: Similar to noise, vibration is also an undesired factor that has adverse effects on the sensor working. Vibration resistance measures are thus required especially when the sensors are used in harsh conditions like off-road vehicles or construction sites.

8. Other Features: Environmental factors such as humidity and temperature effects, mechanical factors of casing etc also form an important while designing the tilt sensors.

3.7 RELAYS

A relay is usually an electro-mechanical device that is actuated by an electrical current. The current flowing in one circuit causes the opening or closing of another circuit. Relays are like remote control switches and are used in many applications because of their relative simplicity, long life, and proven high reliability.Relays are used where it is necessary to control a circuit by a separate low-power signal,or where several circuits must be controlled by one signal

Working of Relays

All relays contain a sensing unit, the electric coil, which is powered by AC or DC current. When the applied current or voltage exceeds a threshold value, the coil activates the armature, which operates either to close the open contacts or to open the closed contacts. When a power is supplied to the coil, it generates a magnetic force that actuates the switch mechanism. The magnetic force is, in effect, relaying the action from one circuit to another. The first circuit is called the control circuit; the second is called the load circuit.

Functions of Relays

There are three basic functions of a relay:

- On/Off Control
- Limit Control
- Logic Operation

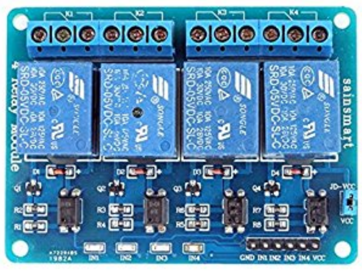


Figure 3.6: Relay Board

3.8 Arduino IDE

Arduino programs may be written in any programming language with a compiler that produces binary machine code. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio, which can be used for programming Arduino.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It was created for people with no profound knowledge of electronics. It includes a code editor with features such as syntax highlighting, brace matching, cutting-pasting and searching-replacing text, and automatic indenting, and provides simple one-click mechanism to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a series of menus.

A program written with the IDE for Arduino is called a ”sketch”.[40] Sketches are saved on the development computer as files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

The Arduino IDE supports the languages C and C++ using special rules to organize code. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two functions, for starting the sketch and the main programs loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions

- setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

- loop(): After setup() is called, this function is called repeatedly by a program loop in the main program. It controls the board until it is powered off or is reset.

3.9 BLYNK APPLICATION

Blynk is a toolset for all makers, inventors, designers, teachers, nerds and geeks who would love to use their smartphones to control electronics like Arduino, RaspberryPi and similar ones. Weve done all the hard work of establishing internet connection, building an app and writing hardware code.

With Blynk, you simply snap together an amazing interface from various widgets we provide, upload the example code to your hardware and enjoy seeing first results in under 5 minutes! It works perfectly for newbie makers and saves tons of time for evil geniuses.

Blynk will work with all popular boards and shields. We wanted to give you full freedom when deciding how to plug Blynk into your existing or new project. You will also enjoy the convenience of Blynk Cloud. Which is, by the way is free and open-source.

Imagine a prototyping board on your smartphone where you drag and drop buttons, sliders, displays, graphs and other functional widgets. And in a matter of minutes these widgets can control Arduino and get data from it.

Blynk is not an app that works only with a particular shield. Instead, it's been designed to support the boards and shields you are already using. And it works on iOs and Android.Blynk also works over USB. This means you can tinker with the app by connecting it to your laptop or desktop while waiting for some internet shield to arrive.

Blynk works over the Internet. So the one and only requirement is that your hardware can talk to the Internet.

No matter what type of connection you choose - Ethernet, Wi-Fi or maybe this new ESP8266 everyone is talking about Blynk libraries and example sketches will get you online, connect to Blynk Server and pair up with your smartphone.

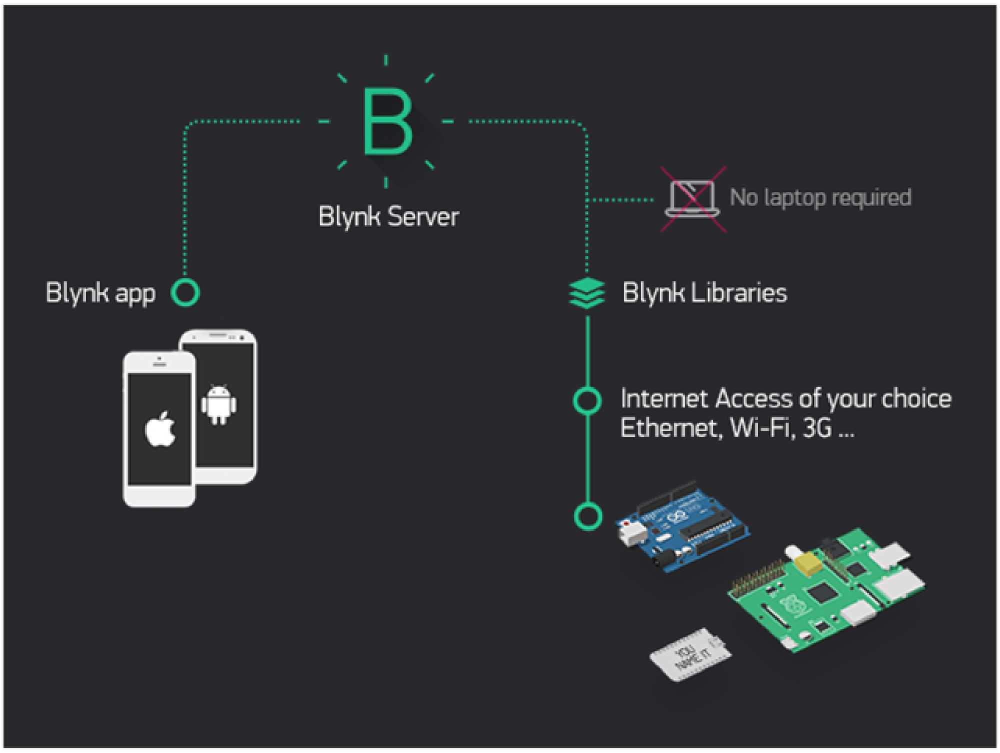


Figure 3.7: Blynk Architecture

Currently, Blynk libraries work with:

- USB
- Ethernet shield
- WiFi shield
- Arduino with Ethernet
- Arduino YN (testing in progress)
- ESP8266
- Raspberry Pi (Blynk will communicate with Pi's GPIOs)
- More Arduino compatible shields and boards

It's not that easy to take Arduino out of your home network, so we've built a Blynk server. It handles all the authentication and communication, and also keeps an eye on your board while the smartphone is offline. Blynk server runs on Java and is open-source. You will be able to run it locally if you really need to. Messaging between mobile apps , Blynk Server and Arduino is based on a simple, lightweight and fast binary protocol over TCP/IP sockets.

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

This chapter describes about the requirements. It specifies the hardware and software requirements that are required in order to run the application properly. The Software Requirement Specication (SRS) is explained in detail, which includes overview of dissertation as well as the functional and non-functional requirement of this dissertation.

A SRS document describes all data, functional and behavioral requirements of the software under production or development. SRS is a fundamental document, which forms the foundation of the software development process. Its the complete description of the behavior of a system to be developed. It not only lists the requirements of a system but also has a description of its major feature. Requirement Analysis in system engineering and software engineering encompasses those tasks that go into determining the need or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirement Analysis is critical to the success to a development project. Requirement must be documented, measurable, testable, related to in identified business needs or opportunities, and defined to a level of detail sufficient for system design.

The SRS functions as a blueprint for completing a project. The SRS is often referred to as the parent document because all subsequent project management documents, such as design specifications, statements of work, software architecture specification, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and non-functional requirements only.

Thus the goal of preparing the SRS document is to:

- To facilitate communication between the customer, analyst, system developers, maintainers.
- To serve as a contrast between purchaser and supplier.
- To firm foundation for the design phase.
- Support system testing facilities.
- Support project management and control.
- Controlling the evolution of the system.

4.2 FUNCTIONAL REQUIREMENTS

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

- Accurately measuring the environments temperature, tilt angle.
- The micro-controller must update the relay to change the status of each component
- Sending the data from the sensors to the application on detection.
- The application must not stop working when kept running for even a long time.
- The application should generate on-demand services.

4.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy.

Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

- Product Requirements
- Organizational Requirements
- User Requirements
- Basic Operational Requirements

In systems engineering and requirements engineering, 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. This should be contrasted with functional requirements that define specific behaviour or functions. The plan for implementing non-functional requirements is detailed in the system architecture. Broadly, functional requirements define what a system is supposed to do and non- functional requirements define how a system is supposed to be. Functional requirements are usually in the form of system shall do requirement, an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of system shall be requirement, an overall property of the system as a whole or of a particular aspect and not a specific function. The systems overall properties commonly mark the difference between whether the development project has succeeded or failed.

The Non Functional and Product Requirements for our project include:

- Response Time

  the time taken by the system to load and the time for responses on any action that happens to the trees

- Reliability

  It is the capability of the software to maintain its level of performance when used under specified conditions. Data accuracy is checked to make sure sensors do not send wrong data. Time accuracy is achieved by programming the hardware and sensor.

- Portability

  It is the capability of software to be transferred from one environment to another.The code can work on any platform efficiently as Arduino can run on any platform.

- Performance

  Power is an important factor when it comes to measurement of performance. The power for NodeMCU is via the USB.The power for the control measures implemented(fan,pump,light,sprinkler) is given via the battery.The battery must be connected to main supply for system to work.

- Robustness

  This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness.

- Ease of Use

  The bylnk app is designed in such a way that the user can understand in a easy manner

4.3.1 Organizational Requirements

Process Standards: IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world. Design Methods: Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

4.3.2 User Requirements

The user requirements document (URD) or user requirements specification is a document usually used to software engineering that specifies the requirements the user expects from software to be constructed in a software project. Once the required infomation is completely gathered it is documented in a URD, which is meant to spell out exactly what the software must do and becomes part of the contractual agreement. A customer cannot demand feature not in the URD, whilst the developer cannot claim the product is ready if it does not meet an item of the URD. The URD can be used as a guide to planning cost, timetables, milestones, testing etc. The explicit nature of the URD allows customers to show it to various stakeholders to make sure all necessary features are described. Formulating a URD requires negotiation to determine what is technically and economically feasible. Preparing a URD is one of those skills that lies between a science and economically feasible. Preparing a URD is one of those skills that lies between a science and an art, requiring both software technical skills and interpersonal skills.

4.3.3 Basic Operational Requirements

Operational requirement is the process of linking strategic goals and objectives to tactic goals and objectives. It describes milestones, conditions for success and explains how, or what portion of, a strategic plan will be put into operation during a given operational period, in the case of, a strategic plan will be put into operation during a given operational period, in the case of commercial application, a fiscal year or another given budgetary term. An operational plan is the basis for, and justification of an annual operating budget request.

Therefore, a five-year strategic plan would typically require five operational plans funded by five operating budgets. Operational plans should establish the activities and budgets for each part of the organization for the next 1-3 years. They link the strategic plan with the activities the organization will deliver and the resources required to deliver them

An operational plan draws directly from agency and program strategic plans to describe agency and program missions and goals, program objectives, and program activities. Like a strategic plan, an operational plan addresses four questions:

- Where are we now?
- Where do we want to be?
- How do we get there?

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points:

- Mission profile or scenario: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or effciency of the system.
- Performance and related parameters: It points out the critical system parameters to accomplish the mission
- Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.
- Operational life cycle: It defines the system lifetime

4.4 HARDWARE REQUIREMENTS

- **Micro-Controllers** : ATMega328P
- **Motors** : Water pump motor
- **Storage** : 32 KB (of which 0.5 KB used by bootloader) SRAM 2 KB and EEPROM 1 KB
- **Sensors** : LM35 Temperature sensor and ADXL335 Tilt sensor

For Mobile

- **Operating System** : Android or IOS
- **RAM** : 512 MB minimum
- **Internal Storage** : 100 MB
- **Internet access** : Yes

4.5 SOFTWARE REQUIREMENTS

- **Operating System** : Windows 10
- **Coding Language** : Embedded C
- **Tools** : Arduino IDE and Blynk App

Chapter 5

SYSTEM ANALYSIS

5.1 INTRODUCTION

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

5.2 FEASIBILITY REQUIREMENT

The study begins by classifying the problem definition. Smuggling/theft of most important trees such as sandal wood in forests, poses a serious threat to forest resources, causes significant economic damage and ultimately has quite a devastating effect on the environment all over the world.

This paper proposes a microcontroller based anti-poaching system employing WSN technology, which is capable of detecting theft. A microcontroller is used along with PC so that the information can be uploaded. WSN is widely used technology in remote monitoring applications.

Four key considerations involved in the feasibility analysis are:

- Operational Feasibility
- Economical Feasibility
- Technical Feasibility
- Social Feasibility

5.3 OPERATIONAL FEASIBILITY

This system will be used widely by various land owners who want to protect their trees, and also can be used by the government to protect the natural resources of the trees which are going extinct.

This system when implemented would reduce the worry of the owners of the trees with all the sensors embedded, it would give information about the tree like temperature of the area(in case of fire) and use of tilt sensor (if the tree is cut and it falls down) and an SOS button which could be used by the owners to call the police.

The need for the system was obtained by analyzing the growing need of these natural resources by most cosmetics and medical companies and to prevent the illegal smuggling of trees.

Since the implementation is done using an Arduino chip with the IoT technology, data observed from the sensors can be transmitted to the owner without distance being a concern and be observed by the app installed in the owners phone.

5.4 ECONOMICAL FEASIBILITY

The system developed will be a good investment for the owners of their lands and the government buying this for the protection of trees, as it wont need any continuous payment for a security guard which is not secure due to human incapability and is a onetime investment of the product.

The sensors provided by the system proposed here can guarantee the maximum security and safety of the trees. As the information about their trees is just a click away on their widely used smart phones, it provides a highly positive business venture on the system proposed.

5.5 TECHNICAL FEASIBILITY

The proposed technology used here is IoT(Internet of Things) which tells us that anything/device which is connected to the Internet can be controlled or transmit data to any other device which is connected to the Internet irrespective of the distance of the devices from each other. This is done by connecting to a server and the devices to be in connection with each other are connected to the server through the internet.

The Arduino Uno used in the system can be easily programmed to connect to the server, provides various ports to connect to the computer with an internet connection or on using a WIFI module on the Arduino internet connectivity can be provided by various service providers as a built in service. Thus, the IoT technology can be implemented on the Arduino chip with any issues.

With the growing speed of the network and availability of internet to all, therefore the proposed system can be implemented using the IoT technology with guarantees of accuracy, reliability and data security.

5.6 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Chapter 6

SYSTEM DESIGN

6.1 INTRODUCTION

Design is a meaningful engineering representation of something that is to be built .It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirements in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design. It consists of:

- System Development Methodology
- Data Flow Diagram
- Class Diagram
- Use Case Diagram
- Sequence Diagram

6.2 SYSTEM DEVELOPMENT METHODOLOGY

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress.

6.2.1 Model Phases

The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete

Unlike the more traditional waterfall model, which focuses on a stringent stepbystep process of development stages, the iterative model is best thought of as a cyclical process. After an initial planning phase, a small handful of stages are repeated over and over, with each completion of the cycle incrementally improving and iterating on the software. Enhancements can quickly be recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last.

Planning and Requirements

As with most any development project, the first step is go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.

Analysis and Design

Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc) that will be utilized in order to meet the needs of the analysis stage

Implementation

With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

Testing

Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have have cropped up

Evaluation

Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on

6.2.2 Reasons for choosing Iterative Model as Development Model

- Requirements of the complete system are clearly defined and understood.
- Major Requirements must be defined;however some functionalities or requested enhancements may evolve with time.
- Resources are planned to be used on contract basis for specific iterations.
- There are some goals which may change in future.
- With every increment,operational product will be delivered.

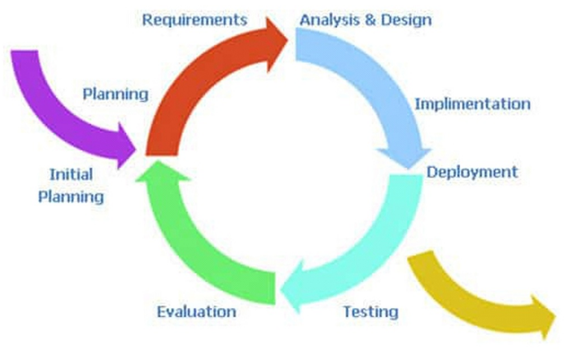


Figure 6.1: Iterative Model

6.3 DESIGN USING UML

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

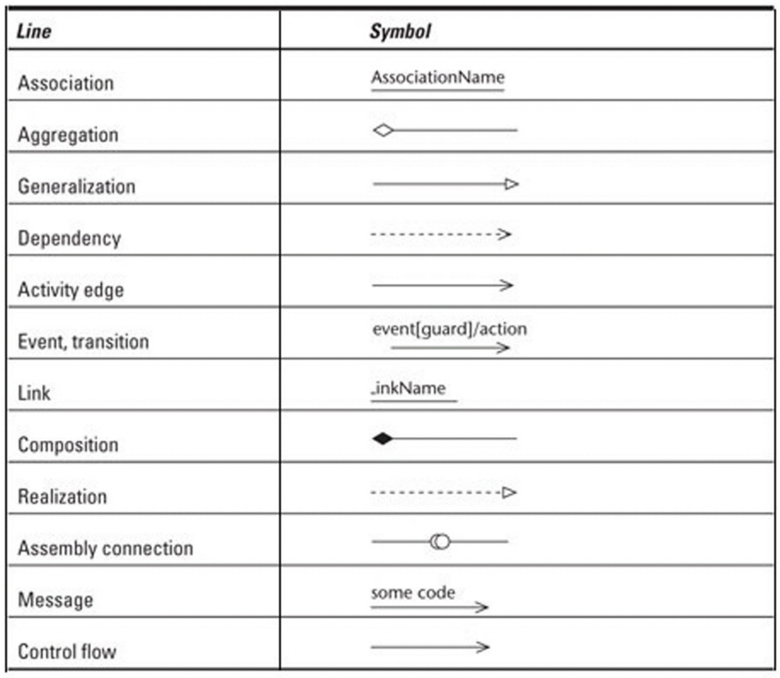
- The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.
- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models
- Provide extendibility and specialization mechanisms to extend the core concepts
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.

- Support higher level development concepts such as collaborations, frameworks, patterns and components.



Line	*Symbol*
Association	AssociationName
Aggregation	◇——————
Generalization	——————▷
Dependency	- - - - - - - - - - - ->
Activity edge	——————>
Event, transition	event[guard]/action ——————>
Link	.inkName
Composition	◆——————
Realization	- - - - - - - - - - - - -▷
Assembly connection	———○———
Message	some code ——————>
Control flow	——————>

Figure 6.2: Symbols used in UML

6.4 DATA FLOW DIAGRAMS

A data flow diagram (DFD) is graphic representation of the flow of data through an information system. A data flow diagram can also be used for the visualization of data-processing (structured design).

It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities.DFDs show the flow of data from external entities into the system, how the data moves from one process to another, as well as its logical storage. There are only four symbols:

1. Squares representing external entities, which are sources and destinations of information entering and leaving the system

2. Rounded rectangles representing processes, in other methodologies, may be called Activities, Actions, Procedures, Subsystems etc. which take data as input, do processing to it, and output it.

3. Arrows representing the data flows, which can either, be electronic data or physical items. It is impossible for data to flow from data store to data store

except via a process, and external entities are not allowed to access data stores directly.

4. The flat three-sided rectangle is representing data stores should both receive information for storing and provide it for further processing.

6.4.1 DFD for Tilt Sensor

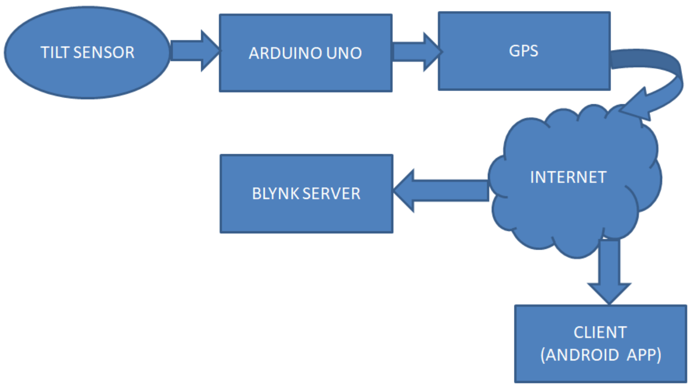


Figure 6.3: DFD for Tilt Sensor

6.4.2 DFD for Temperature Sensor

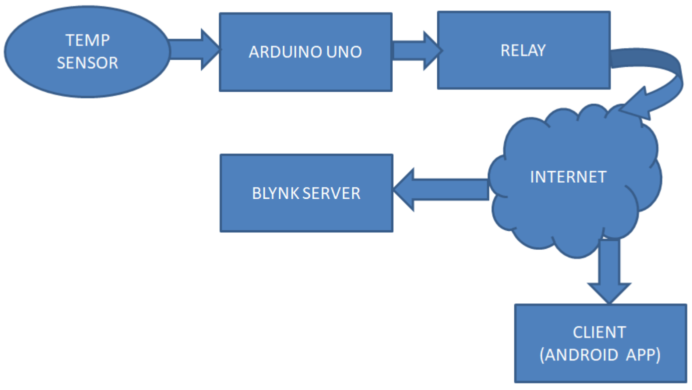


Figure 6.4: DFD for Temperature Sensor

6.5 CLASS DIAGRAM

UML class diagram shows the static structure of the model. The class diagram is a collection of static modeling elements, such as classes and their relationships, connected as a graph to each other and to their contents. The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed.

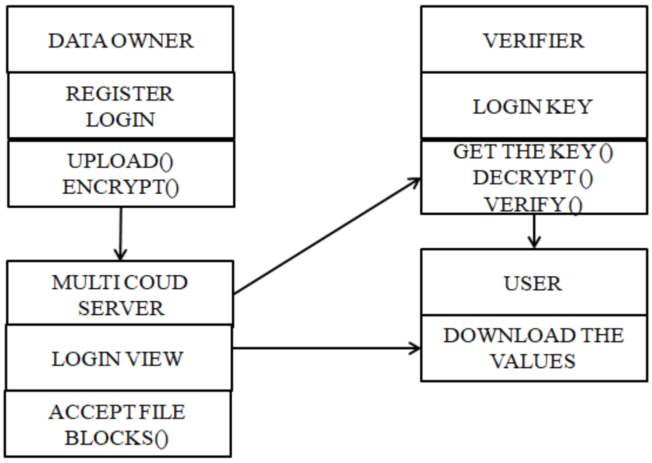


Figure 6.5: Class Diagram

6.6 USE-CASE DIAGRAM

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities which interact with the system are its actors. A set of use cases describe the complete functionality of the system at a particular level of detail and it can be graphically denoted by the use case diagram.

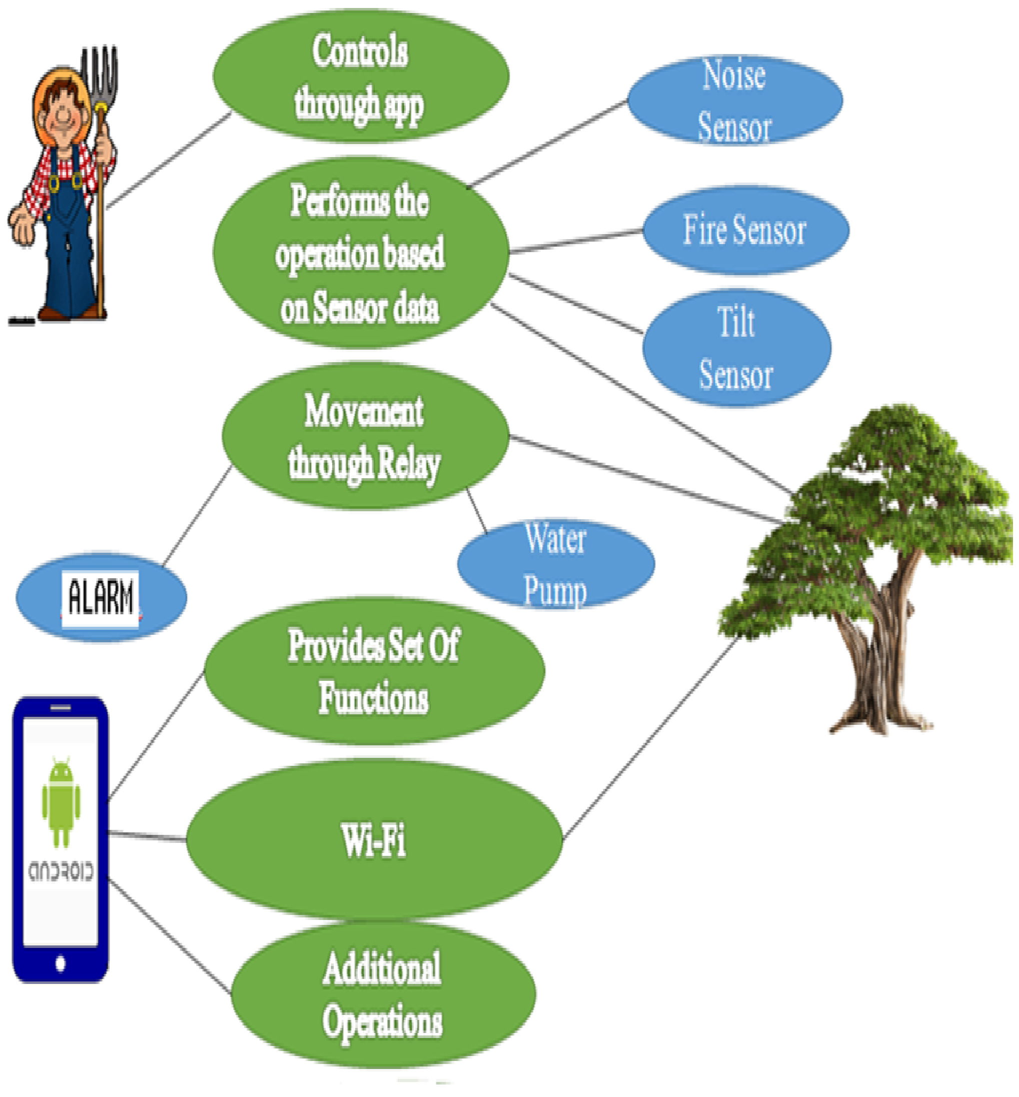


Figure 6.6: Use-Case Diagram

6.7 SEQUENCE DIAGRAM

Sequence diagram are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and the environment. A sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time, the horizontal dimension represents the objects existence during the interaction. Basic elements:

- Vertical rectangle: Represent the object is active (method is being performed).
- Vertical dashed line: Represent the life of the object.
- X: represent the life end of an object. (Being destroyed from memory)

- Horizontal line with arrows: Messages from one object to another.

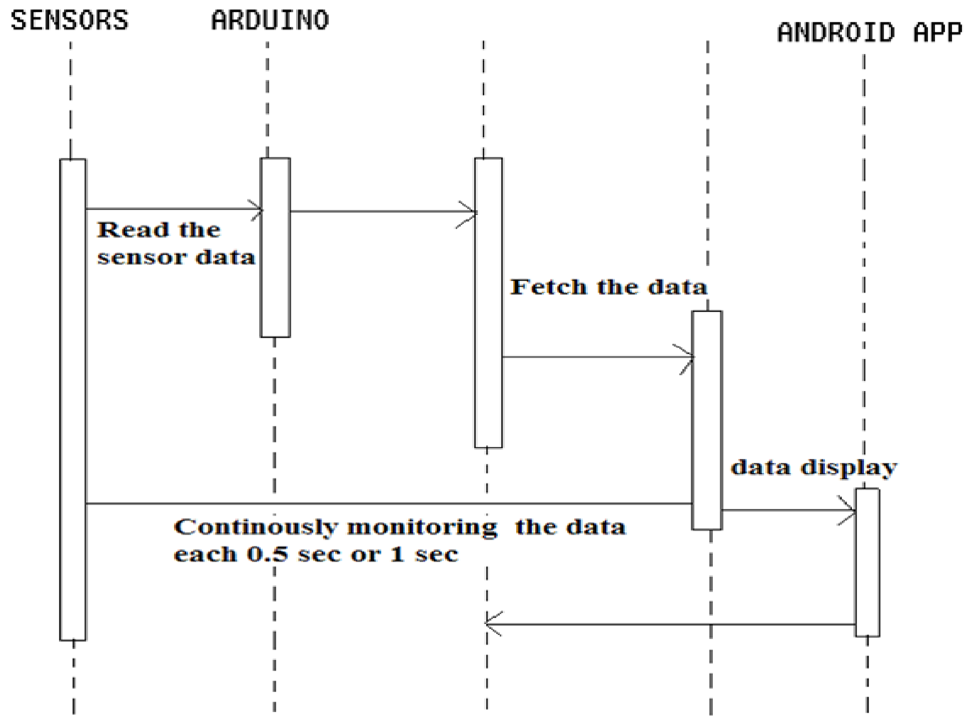


Figure 6.7: Sequence Diagram

Chapter 7

IMPLEMENTATION

7.1 INTRODUCTION

Implementation is a stage of the project where the theoretical design is turned into a working system. This phase involves the actual materialization of the ideas, which are expressed in the analysis document and developed in the design phase. Implementation should be perfect mapping of the design document in a suitable programming language in order to achieve the necessary final product. If the implementation is not carefully planned and controlled, it can cause chaos and confusion.

The implementation stage requires the following tasks:

- Careful Planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Evaluation of the changeover method.
- Correct decisions regarding selection of the platform.
- Appropriate selection of the language for application development.

7.2 ARDUINO CODE

```
#include <Blynk.h>
#include<SoftwareSerial.h>
#include<BlynkSimpleStream.h>
#include<SimpleTimer.h>
#define BLYNK_MAX_SENDBYTES 1200
//SoftwareSerial DebugSerial(2,3);
#define BLYNK_PRINT DebugSerial

#include <TinyGPS.h>  //GPS libraries
TinyGPS gps;
 float flat, flon; // gps long,latti variables

 SoftwareSerial ss(2,3); //Software serial for RX,TX

 static void smartdelay(unsigned long ms);//delay function for gps
```

```
int tilt=A3;
int tilt2=A4;
int temp=A0;
int sound=6;

//int sounddetect=HIGH;
//unsigned long lastSoundDetectTime;//Record the time that
                                    we measured a sound
boolean bAlarm = false;
//int soundAlarmTime = 500;
int rPin1=8;
int rPin2=9;

char auth[]="e94291ef75b545258347b8d1c17d01d9";
SimpleTimer timer;

void setup()
{
 //DebugSerial.begin(9600);
 Serial.begin(9600);
 Blynk.begin(auth,Serial);
 //interruptSetup();
pinMode(tilt,INPUT);
pinMode(tilt2,INPUT);
 pinMode(temp,INPUT);
  pinMode(sound,INPUT);
 pinMode(rPin1,OUTPUT);
 pinMode(rPin2,OUTPUT);
 digitalWrite(rPin1,HIGH);
 digitalWrite(rPin2,HIGH);
 ss.begin(9600);
//timer.setInterval(0,readdata);
}

BLYNK_WRITE(V0)
{
  int pinstate1=param.asInt();
  digitalWrite(rPin1,pinstate1);
}
```

```
BLYNK_WRITE(V1)
{
  int pinstate2=param.asInt();
  digitalWrite(rPin2,pinstate2);
}


void loop()
{
  Blynk.run();
  timer.run();

gps.f_get_position(&flat, &flon); //get the gps location function

 int sounddetect = digitalRead(6);
 if(sounddetect == 1) // If we hear a sound
  {
    Blynk.virtualWrite(V3,"LOW");
  //digitalWrite(8,LOW);
  //delay(500);
  }
 if(sounddetect == 0) // If we hear a sound
 {
  Blynk.virtualWrite(V3,"HIGH");

 }



 int val=analogRead(tilt);
 int val2=analogRead(tilt2);
 delay(500);
 if(((val>280) && (val<400)) && ((val2>280) && (val2<400)))
 {
//   Blynk.virtualWrite(V2,val);
 Blynk.virtualWrite(V2,"NORMAL");
 delay(100);
 digitalWrite(rPin1,HIGH);
 digitalWrite(rPin2,HIGH);
 }
 else
 {
  Blynk.virtualWrite(2,"FALL DETECTED");
  Blynk.notify("Tree has fallen");
  Blynk.email("email id", "ACTION REQUIRED","the tree has fallen");
  digitalWrite(rPin2,LOW);


//gps.f_get_position(&flat, &flon); //get the gps location function
smartdelay(1000);
float flat, flon;
gps.f_get_position(&flat, &flon);
//delay(500);
Blynk.virtualWrite(V6,flat);//print lattitude range in blynk
Blynk.virtualWrite(V7,flon);//print logitude range in blynk

 }

int tempPin=analogRead(temp);
float mv=(tempPin/1024.0)*5000;
float tempc=mv/10;
//int tempPin=analogRead(temp);
//float cel=(tempPin*0.48828125);
Blynk.virtualWrite(V4,tempc);
if(tempc>35)
{
  Blynk.notify("Alert!.Forest on fire");
  digitalWrite(rPin1,LOW);
  digitalWrite(rPin2,LOW);
}
if(tempc<35)
{
//  Blynk.virtualWrite(V1,cel);
  digitalWrite(rPin1,HIGH);
  digitalWrite(rPin2,HIGH);
}
```

```
}
static void smartdelay(unsigned long ms) //smart delay main function
{
  unsigned long start = millis();
  do
  {
    while (ss.available())
      gps.encode(ss.read());
  } while (millis() - start < ms);
}
```

7.3 ESP8266 WIFI MODULE

The ESP8266 can communicate over WiFi in two different modes. It can connect to an existing wireless hot spot, or access point, similar to the way you connect your phone or computer to the Internet. This is called station mode. In station mode the ESP8266 board can reach out to the internet and may also communicate with other devices connected to the same network, including other ESP8266 modules. The values for the ssid and password variables with the name and password for the wireless network that we wll be using.

7.4 CREATING A PROJECT BLYNK APP

After downloading the app, we need to create an account and log in. Welcome to Blynk!

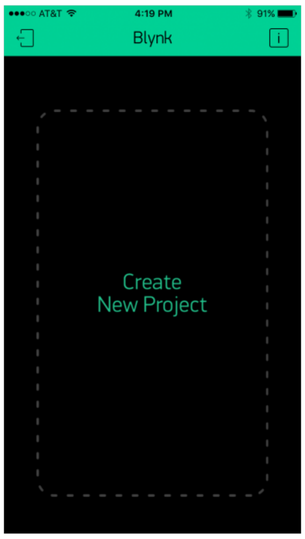
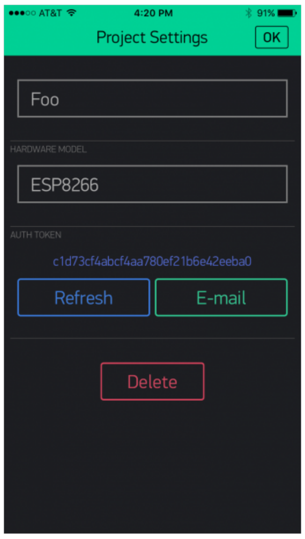


Figure 7.1: Blynk first screen

Well also need to install the Blynk Arduino Library, which helps generate the firmware running on your ESP8266. Download the latest release from Blynks GitHub repo, and follow along with the directions there to install the required libraries.

Create a Blynk Project

Next, click the Create New Project in the app to create a new Blynk app. Give it any name you please, just make sure the Hardware Model is set to ESP8266.



Foo

HARDWARE MODEL

ESP8266

AUTH TOKEN

c1d73cf4abcf4aa780ef21b6e42eeba0

Figure 7.2: Creating a Project

The Auth Token is very important youll need to stick it into your ESP8266s firmware. For now, copy it down or use the E-mail button to send it to yourself.

Add Widgets to the Project

Then youll be presented with a blank new project. To open the widget box, click in the project window to open.

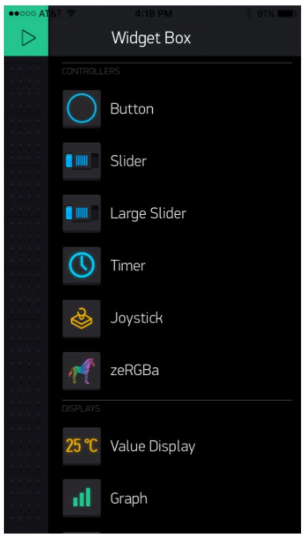
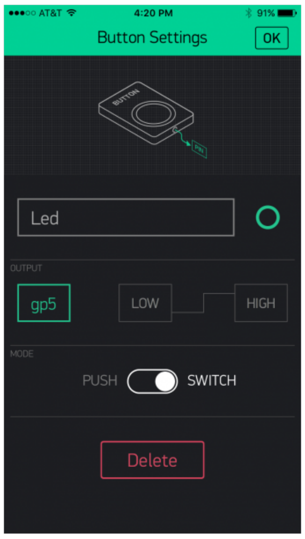


Figure 7.3: Adding widgets

Add a Button, then click on it to change its settings. Buttons can toggle outputs on the ESP8266. Set the buttons output to gp5, which is tied to an LED on the Thing Dev Board. You may also want to change the action to Switch.



Led

OUTPUT
gp5 LOW HIGH

MODE
PUSH SWITCH

Figure 7.4: Adding a button

Upload the Blynk Firmware

Now that your Blynk project is set up, open Arduino and navigate to the ESP8266 Standalone example in the File ¿ Examples ¿ Blynk ¿ BoardsAndShields menu.

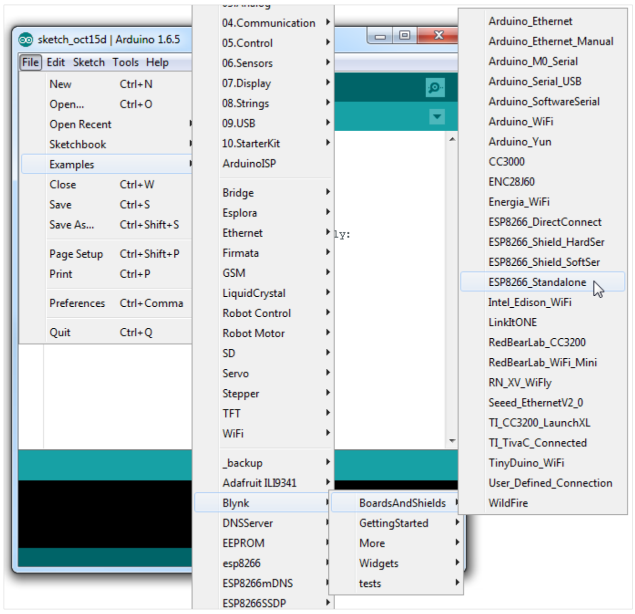


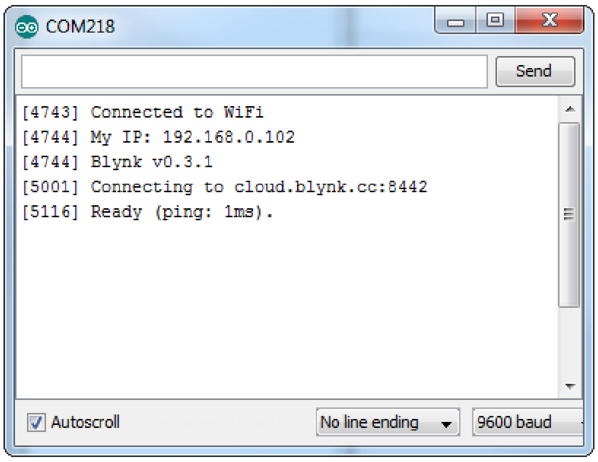
Figure 7.5: Uploading into ESP8266 firmware

Before uploading, make sure to paste your authorization token into the auth[] variable. Also make sure to load your WiFi network settings into the Blynk.begin(auth, ”ssid”, ”pass”) function.

Then upload!

Run the Project

After the app has uploaded, open the serial monitor, setting the baud rate to 9600. Wait for the Ready (ping: xms). message.



```
[4743] Connected to WiFi
[4744] My IP: 192.168.0.102
[4744] Blynk v0.3.1
[5001] Connecting to cloud.blynk.cc:8442
[5116] Ready (ping: 1ms).
```

Autoscroll [✓] No line ending 9600 baud

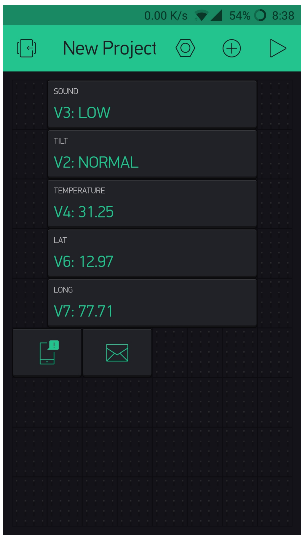
Figure 7.6: Running the Firmware

Then click the Run button in the top right corner of the Blynk app. Press the button and watch the LED!

Then add more widgets to the project. They should immediately work on the ESP8266 without uploading any new firmware.You can add analog output sliders, digital input monitors, and analog input gauges.

The following snaps are for the all the widgets and the sensors and for our project:

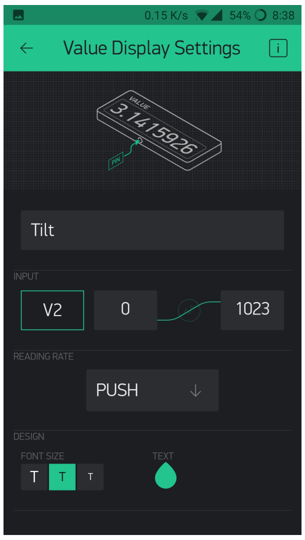
Project Overview



SOUND
V3: LOW
TILT
V2: NORMAL
TEMPERATURE
V4: 31.25
LAT
V6: 12.97
LONG
V7: 77.71

Figure 7.7: Project Overview

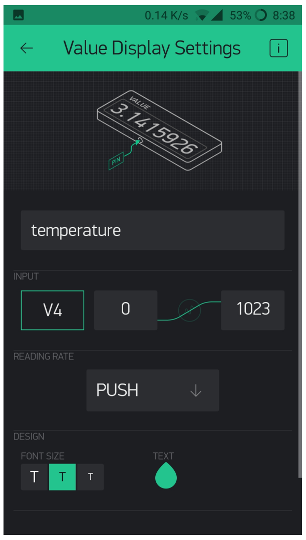
Tilt Widget settings



Tilt
INPUT
V2 0 1023
READING RATE
PUSH
DESIGN
FONT SIZE
T T T
TEXT

Figure 7.8: Widget settings for Tilt Sensor

Temperature Widget Settings


0.14 K/s 53% 8:38

Value Display Settings

temperature

INPUT

V4 0 1023

READING RATE

PUSH

DESIGN

FONT SIZE T T T

TEXT

Figure 7.9: Widget Settings for Temperature Sensor

Chapter 8

TESTING AND RESULTS

8.1 INTRODUCTION

Testing is an important phase in the development life cycle of the product this was the phase where the error remaining from all the phases was detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. Once the implementation is done, a test plan should be developed and run on a given set of test data

Each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually csarried out to make sure that the product exactly does the same thing what is suppose to do. Testing is the final verification and validation activity within the organization itself. In the testing stage following goals are tried to achieve:-

- To affirm the quality of the project
- To find and eliminate any residual errors from previous stages.
- To validate the software as the solution to the original problem.
- To provide operational reliability of the system.

During testing the major activities are concentrated on the examination and modification of the source code.

8.2 TESTING METHODOLOGIES

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important types of testing are:

8.2.1 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level. Using white box testing we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.

- Execute internal data structure to assure their validity.

8.2.2 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot see into it. The test provides inputs and responds to outputs without considering how the software works. It uncovers a different class of errors in the following categories:-

- Incorrect or missing function.
- Interface errors.
- Performance errors.
- Initialization and termination errors.
- Errors in objects

Advantages

- The test is unbiased as the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

8.3 UNIT TESTING

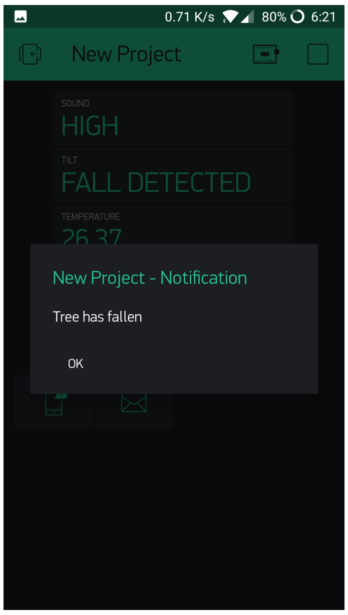
During this implementation of the system each module of the system was tested separately to uncover errors within its boundaries. User interface is used as a guide in the process.

In computer programming, unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure. In object-oriented programming a unit is usually a method. Unit tests are created by programmers or occasionally by white box testers during the development process.

Ideally, each test case is independent from the others: substitutes like method stubs, mock objects, fakes and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

8.4 TEST CASE 1

Function : To detect Tilt Angle threshold.
Purpose : To detect if the tree has fallen.
Preconditions : The tree should be fallen.
Inputs : Readings from the tilt sensor.
Expected Outputs : The notification should come to the Blynk application.
Postconditions : Fall has been detected.



HIGH

FALL DETECTED

26.37

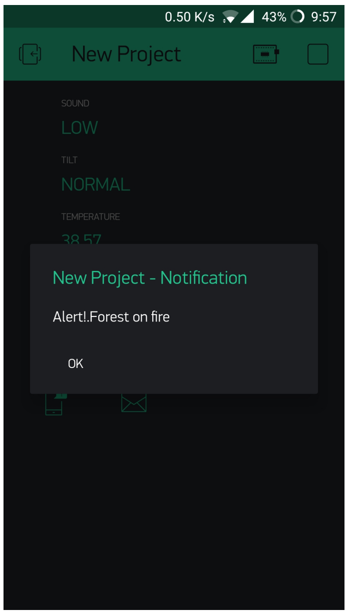
New Project - Notification

Tree has fallen

Figure 8.1: Notification received when fall detected

8.5 TEST CASE 2

Function : To detect Temperature value threshold.
Purpose : To detect if the forest is on fire.
Preconditions : The temperature should be more than the threshold value.
Inputs : Readings from temperature sensor.
Expected Outputs : The notification should come to the Blynk application.
Postconditions : Forest fire has been detected.


0.50 K/s 43% 9:57

New Project

SOUND
LOW

TILT
NORMAL

TEMPERATURE
38.57

New Project - Notification

Alert!.Forest on fire

OK

Figure 8.2: Notification received when fire is detected

8.6 IMAGE OF THE PROTOTYPE

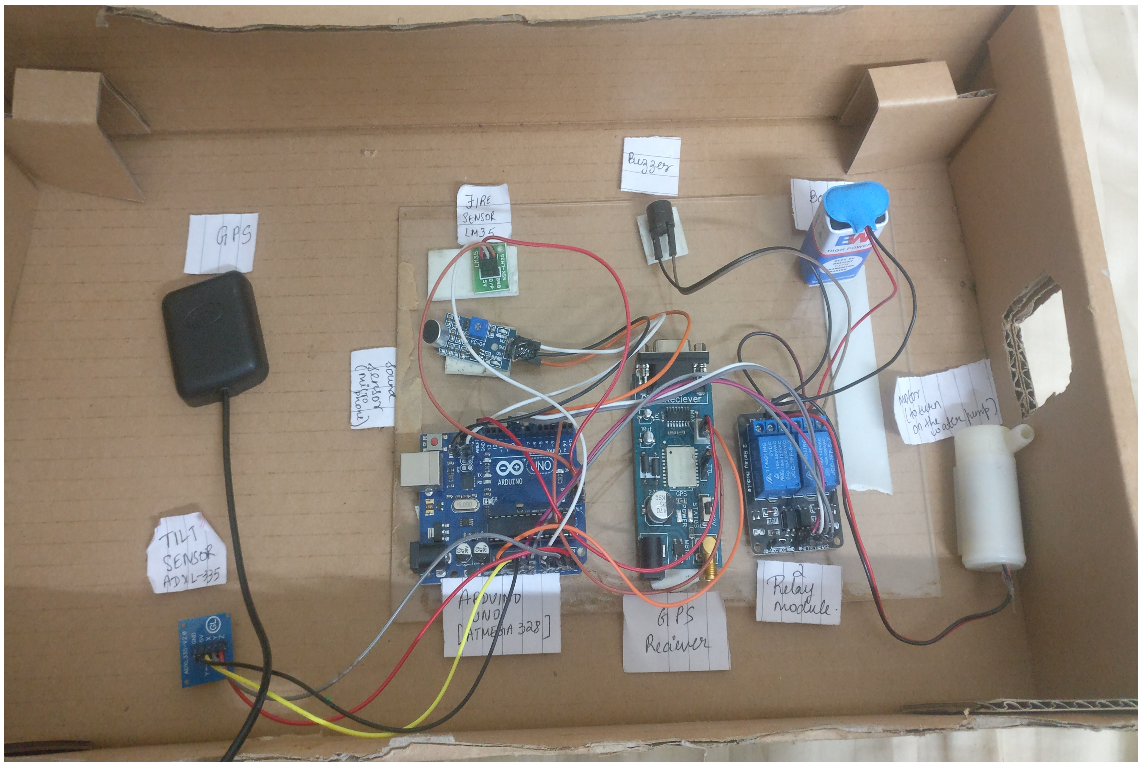


Figure 8.3: Prototype

8.7 FOREST FOR DISPLAYING PURPOSE

Figure 8.4: Forest

Chapter 9

CONCLUSION & FUTURE SCOPE

9.1 CONCLUSION

We identified from the beginning that producing a complete result would be impossible within the given time frame. We viewed the project as a journey where we learnt many lessons and gained insights to the subject which we tried to share in this report and summarized in this chapter. We tried to look at the problem from many points of view which generated some new ideas that could be explored in future. We suggested formal approaches for modelling and analyzing the system which are by no means complete but could become the initiation for further research. We also created a working system and algorithms which we claim to be useful and extensible. However, as we have seen in these chapter, all these achievements are only partially successful. Personally, we would consider this project a success if the ideas described in the report can be a useful reference for future work on the subject.

9.2 FUTURE SCOPE

Although the design was successful there are improvements that could be made in future adaptations of this project.The future scope of work is implementation of Multi-node network and incorporation of microphone motion detector sensor to make systems more effective to acquire data such human or animal interference.

References

[1] Anil Kulkarni, Ajay Khandare, Mandar Malve, "Wireless Sensor Network (WSN) for protection high cost trees in remote jungles from fire and poaching", International Seminar on Sandalwood: Current Trends and Future Prospects , Feb 2014,pp.68-73.

[2] Digital Output MEMS Accelerometer-ADXL345, Analog Devices, 2009, datasheet available at `www.analog.com`

[3] Sridevi Veerasingam, Saurabh Karodi, Sapna Shukla,Design of Wireless Sensor Network node on Zigbee for Temperature Monitoring",2009 International Conference on Advances in Computing, Control and Telecommunication Technologies, IEEE Journals 978-0-7695-3915-7/09,2009.

[4] Manish Y. Upadhye, P. B. Borole, Ashok K. Sharma, Real-Time Wireless Vibration Monitoring System Using LabVIEW, 2015 International Conference on Industrial Instrumentation and Control Pune, India. May 28-30, 2015, pp. 925-928.

[5] Pedro Cheong, Student Member, IEEE, Ka-Fai Chang, Member, IEEE, Ying-Hoi Lai, Sut-KamHo,Iam-Keong Sou, and Kam-Weng Tam, Senior Member, IEEE,"A ZigBee-Based Wireless Sensor Network Node for Ultraviolet Detection of Flame",IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 58, NO. 11, NOVEMBER 2011.

[6] Jamali Firmat Banzi,"A Sensor Based Anti-Poaching System in Tanzania National Parks", International Journal of Scientific and Research Publications, Volume 4, Issue 4, April 2014.

[7] Ravi Bagree,Vishwas Raj Jain, Aman Kumar and Prabhat Ranjan,"TigerSENSE :Wireless Image Sensor Network to Monitor Tiger", P.J.Marron et al : Realwsn 2010,LNCS 6511,pp 13-24,Springer Verlag Berlin Heidelberg 2010.

[8] XBee/XBee-PRO RF Module, Digi International,Inc., Sept 2009.

[9] X-CTU Configuration and Test Utility Software User Guide, Digi International, Inc., August 2008.

[10] Information about MSP430F5529, Users Guide for MSP430 series.pdf available on `www.ti.com`

[11] Shih-Lun Chen, Ho-Yin Lee, Chiung-An Chen, Hong-Yi Huang, Member, IEEE, and Ching-Hsing Luo, Member, IEEE ,"Wireless Body Sensor Network With Adaptive Low-Power Design for Biometrics and Healthcare Applications",IEEE SYSTEMS JOURNAL, VOL. 3, NO. 4