VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELGAUM-590018

A Project Report On

"CONNECTED BUSSES AND SMART BUS SHELTERS"

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

Bachelor of Engineering in Computer Science and Engineering

Submitted by

A SANTHOSH	**(1OX16CS001)**
KIRAN S	**(1OX16CS035)**
KISHAN SHETTY	**(1OX16CS037)**
VIKAS VINUGNA	**(1OX16CS104)**

Under the guidance of

Dr. E SARAVANA KUMAR

Project Guide

Department of Computer Science and Engineering

The Oxford College of Engineering

Bommanahalli, Bangalore-68

2019-2020

THE OXFORD COLLEGE OF ENGINEERING

Bommanahalli, Hosur Road, Bangalore – 560068

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

DECLARATION

We, the students of seventh semester B.E, in the Department of Computer Science and Engineering, **The Oxford College of Engineering,** Bengaluru declare that the project work entitled **"CONNECTED BUSES AND SMART BUS SHELTER"** has been carried out by us and submitted in partial fulfillment of the course requirements for the award of the Degree of **Bachelor of Engineering in Computer Science and Engineering** discipline of **Visvesvaraya Technological University, Belagavi** during the year 2019-2020. Further, the matter embodied in the dissertation has not been submitted previously by anybody for the award of anydegree or diploma to any other university.

A SANTHOSH

KIRAN S

KISHAN SHETTY

VIKAS VINUGNA

Place: Bengaluru
Date:

THE OXFORD COLLEGE OF ENGINEERING

Bommanahalli, Hosur Road, Bangalore – 560068

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

Certified that the Project work entitled **"CONNECTED BUSSES AND SMART BUS SHELTERS"** carried out by **A Santhosh Kumar(1OX16CS001), Kiran S (1OX16CS035)**, **Kishan Shetty(1OX16CS037), Vikas Vinugna(1OX16CS104)** bonafide students of The Oxford College of Engineering, Bangalore in partial fulfillment for the award of the Degree of **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during the year 2019-2020. The Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Dr. E Saravana Kumar	**Dr. R. Ch. A Naidu**	**Dr. A S Aravind**
Project Guide	**Prof & H.O.D, Dept. of CSE**	**Principal, TOCE**

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

ABSTRACT

A smart city is where physical infrastructure is combined with digital infrastructure, in order to improve the quality of economy while reducing environmental impact. Bus shelters are public facilities that are highly used by people while commuting. With IoT, bus shelters can provide a better experience to commuters, and also help the local businessmen to advertise their business. The society is highly benefited by implementation of ZigBee and PIR tracking sensors to predict the frequency of busses and crowd respectively for reducing the commuter density in bus shelters by applying machine learning algorithms. Logistic Regression is a machine learning algorithm which is used in this project, it is a predictive analysis algorithm and based on the concept of probability for predicting the crowd density in the shelter. The tracking of buses is done by the ZigBee module for which an app is created to display the next bus and the crowd density in the shelter. As many people travel through public transport the crowd in the shelter increases day by day, to predict the future crowd density a cloud can be used to store the previous data. The GoDaddy cloud is used to store and retrieve the data for processing or displaying.

ACKNOWLEDGEMENT

A project is a job of great enormity and it can't be accomplished by an individual all by them. Eventually, we are grateful to a number of individuals whose professional guidance, assistance and encouragement have made it a pleasant endeavor to undertake this project.

It gives us great pleasure in expressing our deep sense of gratitude to our respected Founder Chairman **Shri S. Narasa Raju** and to the respected Chairman **Shri S.N.V.L Narasimha Raju** for having provided us with great infrastructure and well-furnished labs.

We take this opportunity to express our profound gratitude to our respected Principal **Dr. A S Aravind** for his support.

We are graceful to the Head of the Department **Dr. R. Ch. A Naidu** for his unfailing encouragement and suggestion given to us in the course of our project work.

Guidance and deadlines play a very important role in successful completion of the project on time. We convey our gratitude to **Dr. E Saravana Kumar,** Project Guide for having constantly guided and monitored the development of the project.

A note of thanks to the Department of Computer Science Engineering, both teaching and non-teaching staff for their co-operation extended to us.

We thank our parents for their constant support and encouragement. Last, but not the least, we would like to thank our peers and friends.

A SANTHOSH

KIRAN S

KISHAN SHETTY

VIKAS VINUGNA

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Internet of Things (IoT) paradigm is becoming more prevalent as the need for connected devices across different environments grows. The ubiquity of connected devices can be observed in virtually every environment; from transportation and logistics domain, healthcare domain, smart environments, personal and social domain.

The Internet of Things definition is usually derived differently among various research in academia and practice, creating fuzziness around this term. However, from a user – centric perspective, it can be described as interconnection of sensing and actuating devices capable of information sharing across platforms through a unified framework. Enabling technologies with characteristics such as the reduction in size (complexity), energy consumption and costs, provide more potential application possibilities for IoT integration into our everyday life. Current trends emphasize particular interests in the application of IoT to urban context by governments. Smart city initiatives aim to address and improve economic prospects, environmental aspects and the quality of life for the public.

The application of IoT in cities and urban areas is gaining momentum where a number of institutions and governments are pushing the adoption of smart solutions to manage public affairs. The Smart bus shelter will be a way to connect various businesses and institutions to public transport commuters. Through considering alternative design approaches, bus shelters can relate to the wider urban environment and everyday practices, which includes complex relationships between multiple stakeholders and public transport services, functions, design, and management. The use of IoT enables advertisers determine how effectively information is conveyed by identifying the potential number people that view the message in a public space such as the bus shelter.

In the smart city domain, IoT is widely adopted to provide solutions that make use of public resources to increase the quality of services offered to people while reducing operational costs.

Although there is no clear definition of a “smart city”, the objective towards smart city initiatives is usually aimed to deploy a seamless integration of communication infrastructure,

sensing and actuating technologies. Applications that allow access to different public services are used, thus unleashing potential synergies and increasing transparency to the citizens. Cities in North America, Europe and India are leading efforts in implementing smart technologies to address urban issues such as energy usage in street lamps, traffic congestion

In a world of increasing demand for connectivity, the connected bus shelter demonstrates how street furniture and city infrastructure such as ubiquitous bus shelters, traffic lights, public information panels, kiosks, rubbish bins and power poles, can become Smart City connectivity hubs and enablers for the Internet of Things (IOT) – a world where millions of devices and machines are connected.

The Connected Bus Shelter opens up new business model opportunities by providing a broadband connection to the shelter.

1.2 Problem Statement

The traditional bus shelters that we see in our day to day life are very subtle and lack development of any sort. To be more specific, they just have roof over head and chairs to sit. In this rapid developing world where technology is taking over everywhere, we believe the bus shelters must also not lag behind. In the present scenario, they don’t have any sensors that will estimate and predict the crowd density and also the electric facilities that are available in the shelter must be economically utilized i.e., as and when there are no commuters in the shelter the electric facilities must turn off and vice versa. There are also no display panels that will display necessary information like advertisements or maps for local tourists or breaking news to update the commuters. Instead there are umpteen number of posters and banners that are pasted on the walls of shelters and hung on top of it which looks very clumsy and not attractive. In addition, there are no tracking facilities for the buses so that the user can know the estimated arrival time of the next bus. Also there is no cloud to store all the data which can be used later for predicting future crowd density.

1.3 Proposed Solution

Our research is associated with Emergent Configurations of Connected Systems (ECOS), which is a part of the research projects in the Internet of Things and People (Iota). Our proposed system encompasses the Smart city domain, where we investigate how IoT can be applied in smart bus shelters to add value to stakeholders such as businesses, institutions, transport providers as well as the commuters.

The focus of the project is about utilizing digital display of information in public spaces such as bus shelters. The information or content that we focus on in the research concerns digital advertisements or announcements from businesses, institutions and other points of interest around the city. We try to investigate a smart approach towards connecting businesses with the target audience in a particular bus shelter. This focus is expected to contribute to new smart advertising opportunities that create value for stakeholders. The integration of IoT in the smart bus shelter involves the use of sensors that collect contextual data from the commuters as well as the environment. Such data can be useful in the system's decisions making, when it comes to what content is displayed on the screens. This "smart behavior" results in a context aware system that optimizes the relevance of information displayed, at a given time, location and for specific commuters in a bus shelter. Another aspect that utilizes the sensors concerns automation of the bus shelter. This reduces unnecessary energy consumption by different appliances such as lights, display screens or heating in the bus shelter. Motion sensors for instance can, influence the dimming of lights whenever the bus shelter is not in use for a set period of time. An increased number of mediated messages delivered through billboards, posters, direct mail, e-mail spam, etc., tend to be intrusive to the viewer. New interactive advertising methods are being adopted in public spaces to capture audiences and engage them, tending to make cities "livelier", and becoming more effective for attracting customers. Smart bus shelters can be interactive in the sense that the screens activate with motion detection, giving a certain experience to the commuter which is more likely to capture their attention while waiting for a bus.

Perhaps one of the common values adding aspect of the smart bus shelter is in the sense that it is an enabler of effective communication of the city to the public. Information is usually of value if communicated and utilized appropriately. The information displayed in a bus shelter for instance, can create awareness of the happenings around the city which creates interest for the commuter.

The value added can be looked at from two perspectives:

A. User perspective: The user refers to the public transport commuter or a pedestrian that interacts with the smart bus shelter. Relevant information about the city can be of value to the user in terms of saving costs. From the user perspective, another key aspect that comes into play is the user experience in a bus shelter while waiting for the bus,

B. Stakeholder perspective: The stakeholders are the public transport provider, businesses and institutions that rely on smart bus shelters to convey information to the commuters. Public transport providers consider services that convey traffic information such as a real time update of the bus position or the estimated time of arrival. Businesses and institutions on the other hand invest in public announcements of events, or advertisement of local sales or offers.

The focus of our proposed system is on the fact of reducing commuter density as well as by doing so benefiting the stakeholder and commuters.

1. Estimating Crowd Density

The density of pedestrian crowds in public places has often been determined by use of cameras that have the capability to identify distinct objects in the field of view. However, it raises an issue of privacy in various areas. In the case of the bus shelters, using cameras to determine the density of an occupied bus shelter may be expensive to implement for every bus shelter, resulting to less viability. A number of researchers have investigated the possibilities of using sensors instead, to map the density of an occupied space. Binary Proximity Sensors (BPS) provide a low cost and privacy preserving solution to track mobile objects in a smart environment. A BPS is a low-cost sensor that outputs a “1” when motion is detected within a given range, and “0” otherwise. The research suggests that sensors can be used to estimate the density of a crowd in an occupied space, without compromising the user’s privacy as it would with cameras. It is relevant to our research since we try to strategically estimate the crowd density in a bus shelter. We suggest using dynamic infrared distance sensors which are an improvement to standard PIR’s. A distance sensor has a varying output that is proportional to the proximity of the detected object. A standard infrared distance sensor, which can detect close range objects from a distance of 10cm – 100cm. Using a similar setting as mentioned in [20], the values obtained from the sensors are dynamic, indicating how close or far a person is to the sensor.

2. Machine Learning

Machine learning is a method where the system analyses data and uses algorithms to iteratively learn from the data, increasing its performance from experience.

The use of Machine learning (ML) has been spreading rapidly with the increase of data collected and availability of devices that can collect contextual data from various sources. Machine learning systems continuously improve the performance of the executing program by learning from past examples. The basic idea of machine learning is to take data from a sufficiently vast data set and identify patterns that exists within the data. The patterns are used to predict and determine the future behavior of the machine without having to program it more.

Logistic regression: Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability. We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the '*Sigmoid function*' or also known as the 'logistic function' instead of a linear function.

The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

CHAPTER 2

ANALYSIS AND LITERATURE SURVEY

2.1 Literature Survey

2.1.1 Public Transportation based Literature Survey

Title: Urban Space Design Concepts

Author: Kevin Tingyi Zhang

Because public spaces such as bus stops affect the daily lives of many, the styles of these elements are important (Lillebye, 1996). From survey results, it is shown that most people prefer public spaces that are backdrop-like, that blend in with the surroundings and do not seek attention (Gjerde, 2011). The general public likes patterns of conformity and order more than design professionals (Gjerde, 2011). These preferences speak to the desire of people for a more integrated travel experience that benefits from the latest advances in design but also has a relatively conservative look that maintains its attractiveness over time (Fung, 2012; Thomsen, 2011). In addition, it is important to not be carried away by over urbanizing spaces, for there are many factors of lower density developments, such as long views, that are conducive towards active transportation (Susilo et al., 2012).

While there is usually support for public space improvement projects as they are often on city owned land (Giddings, Charlton, & Horne, 2011) and for environmentally sustainable agendas (Susilo et al., 2012), there are still major road blocks. Common difficulties of public space projects include issues of public versus private realm, constrained roadway widths, aligning multiple funding sources and stakeholders, and coordination of utilities, streetscaping and infrastructure (Reconnecting America, 2011). Various design elements may become points of contention. Therefore, it is recommended that the public be involved at all stages of the process (Borst, Miedema, de Vries, Graham, & van Dongen, 2008; J. Jacobs, 1961). Due diligence in this aspect will ensure there is adequate support for the project and that the final design reflects the needs of the neighborhood and commuters alike (Chrisomallidou, Chrisomallidis, & Theodosiou, 2004). Great designs are often the result of planners, architects, and engineers coming together to realize an integrative urban vision (Kashef, 2008). Other non-design parties that are significant to the process include area businesses, local employers, development

professionals, local associations and law enforcement personnel (Reconnecting America, 2011). [1]

2.1.2 Connected Vehicles based Literature Survey

Title: Inter Vehicle Connectivity Concepts

Author: Ning Lu

It is widely believed that the advances of inter-vehicle communications will reshape the future of road transportation systems, where inter-connected vehicles are no longer information-isolated islands. By means of inter-vehicle communications or V2V communications, information generated by the vehicle-borne computer, control system, on-board sensors, or passengers can be effectively disseminated among vehicles in proximity, or to vehicles multiple hops away in a vehicular ad hoc network (VANET). Without the assistance of any built infrastructure, a variety of active road safety applications (e.g., collision detection, lane changing warning, and cooperative merging) [52] and infotainment applications (e.g., interactive gaming, and file and other valuable information sharing) [53] are enabled by inter-vehicle wireless links.

Moreover, the installation and maintenance of aftermarket sensors (providing add-on functions) are inconvenient using cable connection. Recent advances in wireless sensor communication and networking technologies have paved the way for an intriguing alternative, where ECU and sensors are composed of an intra-vehicle wireless sensor network, leading to a significant reduction of deployment cost and complexity. There exist multiple candidate wireless technologies to build intra-vehicle wireless sensor networks, and the feasibility of different wireless options to in-vehicle environments has been a research focus.

CVs will operate in a complex ecosystem that connects vehicles between each other and the traffic infrastructure, and also opens up new forms of connectivity and relationships to cloud-based services and smart home, smart cities, etc. The ecosystem is becoming a very complex System-of-Systems in which security must be considered and accounted for at all levels. For a safe and secure transportation system, the community must take a fresh look at the larger picture, and develop the policies, designs, and operations needed to incorporate security throughout the design. Use of disruptive technologies such as big data, machine learning and AI can help build a better, safer and more secure CV ecosystem. [3]

2.1.3 IoT based Literature Survey

Title: IoT Architectural Technologies

Author: Kelvin Wachira

Several efforts have been put forward to digitize bus shelters with the aim of providing better service to commuters. Ericsson unveiled the connected bus shelter in 2015 that incorporates a 3G, Long-Term Evolution (LTE) and Wi-Fi small cell technology for communication [15]. Small cells are low-powered access nodes that can operate in a licensed spectrum or unlicensed carrier–grade Wi-Fi. They typically have a range of 10 meters to several hundred meters. Due to the low power consumption [16], they are a promising candidate for backhauling Wireless Sensor Networks (WSNs).

WSNs are spatially distributed autonomous sensors that monitor certain environmental conditions such as temperature, motion, humidity, etc.

Radio Frequency Identification (RFID) technology was a major breakthrough that enabled automatic digital identification of various things that were otherwise digitally unidentifiable. [4]

Title: IoT in public transportation

Author: Kelvin Wachira

Boja proposes an IoT system for an Intelligent Transport System (ITS) that includes a GPS system to track the location of the bus, NFC (Near Field Communication) device in the bus, and sensors to monitor the ambient environment in the bus i.e. temperature, humidity and air quality.

The system architecture of the ITS consists of 3 subsystems:

The location subsystem consists of a Global System for Mobile (GSM) module with General Packet Radio Service (GPRS) modem for communication with other devices.

The commuter subsystem consists of an NFC reader which enables payment by the commuter.

The ambient subsystem consists of temperature sensors, humidity sensors and air quality sensors. It is responsible for monitoring the ambient environment in the bus.

The ITS system proposed in the research relates the smart bus shelter in terms of the goals intended to be achieved. The ITS aims to improve the quality of service provided to the commuter by presenting useful information about the oncoming bus. It was deployed in India and seemed useful for the public transportation system. Similarly, the information system in a smart bus shelter provides information concerning surrounding businesses or other points of interest around the city. This improves the commuter's experience in a bus shelter while they wait for the next bus. [4]

Title: PIR Sensors for crowd density

Author: Kelvin Wachira

A number of researchers have investigated the possibilities of using sensors instead, to map the density of an occupied space. Binary Proximity Sensors (BPS) provide a low cost and privacy preserving solution to track mobile objects in a smart environment [20]. A BPS is a low-cost sensor that outputs a "1" when motion is detected within a given range, and "0" otherwise. A more common BPS used in prototyping is the Passive Infrared Sensor (PIR).

The research suggests that sensors can be used to estimate the density of a crowd in an occupied space, without compromising the user's privacy as it would with cameras. It is relevant to our research since we try to strategically estimate the crowd density in a bus shelter. We suggest using dynamic infrared distance sensors which are an improvement to standard PIR"s. [4]

2.1.4 Smart Cities based Literature Survey

Title: Smart Bus Shelter

Author: Palle Joel Karthik

M Kopielski investigate the use cases of SMARTIE, a smart city initiative, which include smart public transport specifically regarding smart bus shelters. Users (commuters) of the public bus transport interact with the system using their smart phones, through associated augmented reality markers (AR markers), available on specified bus stops. A dedicated mobile application indicates information on the time of arrival to that bus shelter and suggest alternative routes to the commuter. Figure 2 shows an illustration of the solution.

In relation to our work, the research investigates the importance of having interactive installations in different public spaces. Additionally, the aspect of connectivity consequently leads to linking the business to the advertisement in the bus shelters. The commuter can interact

with the advertisement displayed through integrating QR codes that can be scanned. Once scanned, it can reveal more information concerning the subject and display it on a commuter's mobile device.

The focus of our research is on the business side and how local businesses, institutions or outdoor media companies can communicate information and connect to specific commuters. [5]

2.1.5 Cloud based Literature Survey

Title: Bus to cloud communication traffic analysis

Author: Johannes Pillmann, Benjamin Silwa

In this work, a simulative analysis of car-to-cloud data traffic was presented. The simulation model based on a detailed environment model of the area of the German city Dortmund including a precise road map and cell locations of a large German mobile network provider. The mobility of the vehicles was simulated for different traffic states using the SUMO simulator. The car-to-cloud communication model, leveraging LTE uplink channels, was founded on a measurement-based empirical channel model. CVIM Data Packages were sent to the cloud leveraging the CVIM format. The simulation resulted in an average data rate of 482.1 kbps in the case of the Free Flow traffic state and was significantly reduced by a factor of seven when traffic jams occurred. This effect occurred contrary to the number of in-vehicle generated CVIM Data Packages per LTE cell that increased due to the slower vehicle speed during traffic jams. The authors of this work therefore propose an automotive traffic aware communication scheduler reducing the amount of data sent in accordance with the current vehicular traffic situation. The simulation provided an estimate for the upper and lower data rates. The results serve as a reference for network resource planning in terms of RB and resource scheduling for car-to-cloud type services.

We use cloud computing rather than distributed computing because the goal of Distributed Computing is to provide collaborative resource sharing by connecting users and resources. Distributed Computing strives to provide administrative scalability (number of domains in administration), size scalability (number of processes and users), and geographical scalability (maximum distance between the nodes in the distributed system).

Whereas Cloud Computing is all about delivering services or applications in on demand environment with targeted goals of achieving increased scalability and transparency, security, monitoring and management. In cloud computing systems, services are delivered with transparency not considering the physical implementation within the Cloud. [6]

2.1.6 Arduino based Literature Survey

Title: Integrated Arduino Architectural Concepts

Author: Shaik Thaslim , V. Raghavendra Reddy

The Arduino UNO (ATmega328) is a microcontroller has 14 digital input and output pins, 6 analog pins, 16 MHZ ceramic based resonator, USB port, power port, ICSP along with reset port. This Arduino acts as simple computer that run one program at a time. In our work, the Arduino is operated at 5V to detect the bus stop number using the other integrated component RFID reader.

Tracking and management system utilizing the various electronic components such as Arduino UNO, Wi-Fi module, Router and GPS. This system is designed to the waiting time of bus passengers and track the bus location. All the data is stored in the server and is retrieved to the passengers through mobile application. The IOT based smart local bus transport management system is presented in. This system helps the users to reach the destination with short time. Moreover, the proposed also provides the current location of bus. Based on these literatures, we implemented the smart public transport system for blind people to identify the information bus.

There are many types of Arduino like Arduino Uno, Arduino Leonardo, Arduino Due, Arduino Yun etc. We use Arduino Uno because its cost efficient and serves its purpose better. [7]

2.1.7 RASPBERRY PI based Literature Survey

Title: A Survey on GPS enabled City Bus Tracking System using Raspberry Pi.

Author: Shaik Thaslim

Raspberry Pi is the name of series of single board computers made by Raspberry Pi Foundation. The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO pins that allows you to control electronic component for physical computing and explore the IOT.

The Raspberry Pi board used in the proposed system which having following features: 5V 1A maximum power from an adaptor, 700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set), 1GHz operating speed,4 USB ports for Key board mouse or accessing external memory,40 GPIO pins, Ethernet port for internet connectivity, VGA connector and HDMI connector, 3.5mm stereo jack for audio out to amplifier, MicroSD card interface slot to carry the OS, 512MB of SDRAM.

In this project IR sensor are connected directly to microcontroller (port pin). The GPS is connected via serial cable and data is sent to server by means of wireless communication. The data is received at server side which is operating on Linux based system. Which will support the MYSQL based data base which will contain details as: - Location, Vacant seat. The server runs on it. To full fill all the above requirement we have used the raspberry pi. Which can easily run all commands on server and support all sever function. The data form server can have made available to end user using mobile application or web page. These things can be controlled dynamically from both sides i.e. user command can be answered from server side which make it complete IOT based system.

Other alternative boards are Pine H64 Model B, Orange Pi3 etc.

i. H64 has slower Wi-Fi and Bluetooth and doesn't support dual display whereas Pi has faster speed as it has newer arm architecture. [8]
ii. Orange Pi3 has very flaky performance in testing compared to Raspberry Pi.

2.1.8 Machine Learning based Literature Survey

Title: Logistic Regression for Machine Learning

Author: Lian Niu

Machine learning is a method where the system analyses data and uses algorithms to iteratively learn from the data, increasing its performance from experience. Machine learning systems continuously improve the performance of the executing program by learning from past examples. The basic idea of machine learning is to take data from a sufficiently vast data set and identify patterns that exists within the data. The patterns are used to predict and determine the future behavior of the machine without having to program it more.

There are many types of ML that exists; however, the more common methods are:

Classification: It includes a Classifier, which is a system that inputs a set of discrete/ continuous values called features, and outputs a single discrete value called a class. P. Domingo's, describes ML as consisting of a combination of three key components: Representation, Evaluation and Optimization.

Logistic regression: Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability. We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function; this cost function can be defined as the '*Sigmoid function*' or also known as the 'logistic function' instead of a linear function.
The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

Advantage of using Logistic regression:

It is a widely used technique because it is very efficient, does not require too many computational resources, it's highly interpretable, it doesn't require input features to be scaled, it doesn't require any tuning, it's easy to regularize, and it outputs well-calibrated predicted probabilities.

Like linear regression, logistic regression does work better when you remove attributes that are unrelated to the output variable as well as attributes that are very similar (correlated) to each other. Therefore, Feature Engineering plays an important role in regards to the performance of Logistic and also Linear Regression. Another advantage of Logistic Regression is that it is incredibly easy to implement and very efficient to train. I typically start with a Logistic Regression model as a benchmark and try using more complex algorithms from there on.

Because of its simplicity and the fact that it can be implemented relatively easy and quick, Logistic Regression is also a good baseline that you can use to measure the performance of other more complex Algorithms.

Comparison with other algorithms:

There are other algorithms like linear regression, decision tree, KNN (k nearest neighbor) etc.

Linear regression technique is used to predict continuous values. It is a well-known training method for ML that requires a sufficiently long enough human labeled dataset to predict values based on the patterns of the dataset.

KNN is slow in real time as it has to keep track of all training data and find nearest neighbor nodes whereas LR can easily extract output from the tuned co-efficient. [9]

2.1.9 ZigBee based Literature Survey

Title: STUDY ON ZIGBEE TECHNOLOGY

Author: Muthu Ramya.C, Shanmugaraj.M, Prabakaran.R

Wireless Technology is being developed rapidly nowadays. Advancement in microelectromechanical systems brings integration of sensing, signal processing and RF capability on very small devices. All kind of portable applications tend to be able to communicate without the use of any wires. Aim of wireless communication is to gather information or perform certain task in the environment. A typical sensor node contains three C's, are Collection, Computation and Communication units. Based on the request of sink, gathered information will be transmitted wirelessly. The collection unit has series of sensors. Computation unit contains microcontroller and memory. Finally, the communication unit contains transceiver to transmit and receive data; various transceivers (such as RFM TR1000 family, Hardware accelerators, Chipcon CC1000 and CC2420 family, Infineon TDA 525x family, IEEE 802.15.4/Ember EM2420 RF transceiver, Conexant RDSSS9M) used for this purpose.

The reasons [1] for using ZigBee are,

i. Reliable and self-healing

ii. Supports large number of nodes.
iii. Easy to deploy
iv. Very long battery life
v. Secure
vi. Low cost
vii. Can be used globally
viii. Vibrant industry supports with thirty or more vendors supplying products and services
ix. Open Standards protocol with no or negligible licensing fees
x. Chipsets available from multiple sources
xi. Remotely upgradeable firmware
xii. No new wires
xiii. Low power (ability to operate on batteries measured in years)
xiv. Low maintenance (meshing, self-organizing)
xv. Standards based security [AES128]
xvi. Ability to read gas meters

ZigBee is one of the most widely utilized Wireless Sensor Network standards with low power, low data rate, low cost and short time delay characteristics, simple to develop and deploy and provides robust security and high data reliability. Name of the ZigBee came from zigzagging patterns of honey bees between flowers, represents the communication between nodes in a mesh network [1].

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

3.1 Hardware and Software Requirements

Hardware Requirements:

1. Raspberry Pi
2. Arduino Uno
3. ESP8266
4. Proximity IR Sensors
5. Motion Sensors
6. TFT Display
7. GPS Module
8. Pollution Sensors

Software Requirements:

1. Arduino IDE
2. Python IDE
3. GoDaddy Cloud
4. Twitter API
5. Google Maps API

3.2 System Requirements

3.2.1 Functional Requirements

The system functional requirements define the specific functions expected from the system in order to operate successfully according to the case scenario.

FR1- Activation / Deactivation

The system is automatically activated when motion is detected in the bus shelter through motion sensors. It should deactivate (or enter sleep mode) if the bus shelter is left unused for a period of time. This is done in order to conserve the energy whenever the bus shelter is idle, for instance during the night.

A Proximity Infra-Red sensor (**PIR**) is used in the prototype due to its efficiency to detect motion.

FR2 – Determine density of occupants in the bus shelter

The density of occupants refers to the approximate number of people occupying a bus shelter. The reason as to why the exact number is not specified is due to the lack of sensors that can determine this number. Current devices that are used to achieve such a result are cameras that detect heat signatures from distinct objects. However, they are expensive and unavailable for the small-scale prototype. Therefore, small scale infrared proximity sensors are installed strategically at different points in the bus shelter model.

FR3 – Connect to cloud

The connectivity aspect of the Smart bus shelter system is one of its underlying significance towards presenting relevant content. This means that the advertisements or information displayed should be presently significant, rather than having outdated advertisements. The prototype should connect to twitter and retrieve tweets having a specific hashtag for instance "*#Lunchoffer*", from specific twitter accounts. Twitter accounts were created that represent different local businesses where to simulate a lunch offer, afterwork offers, gallery opening, sales, etc.

FR4 - Time – based information display

Time – based display refers to the capability of the system to determine the right time to display certain information. The aim is to demonstrate a smart system that can display at the right time when the information it is more likely to be relevant to the target audience. For instance, an advertisement for a lunch offer is likely to be irrelevant after lunch. Therefore, in this case, another type of advertisement is prioritized to be displayed for example, after work offer.

FR5 – Interaction

Interactivity in the system refers to the commuter's ability to interact with an advertisement. This is advantageous in the case that the commuter prefers to have more information about the offer on their phone. A QR code is used that point to the URL of a particular advertisement for instance. The commuter can use this URL to redeem an offer at the business premise or to get more information about the location of the offer.

3.2.2 Non-Functional Requirements

NFR's elaborate the performance characteristics of the proposed system.

NFR1 - Reliability

A measure of how relevant the information displayed in the bus shelter is to the commuter depending upon his interests and the route he is taking. The information is meant to be targeted to a specific set of audience. Hence, a highly reliable system should capture a substantial number of the targeted group.

NFR2 – Accuracy

A measure of how accurate information is displayed based on the time and location attributes. A lunch offer should not be displayed three hours after lunch for instance.

NFR3 - Availability

When motion is detected in the bus shelter, the screen activates and displays the information. The information displayed should be available through efficient connectivity for the bus shelter.

CHAPTER 4

SYSTEM DESIGN AND MODELLING

4.1 Preliminary Design

The one of the important features of the project is designing the system. The design part provides the different elements of the system such as, architecture and components. System design solves the problem by splitting the components of the complex system into smaller components and will perform and operate on each individual component. The main primary component in our system is a raspberry pi board. This board contains different individual modules like pollution sensor, motion sensor, a TFT LCD display, and an emergency button. The board communicates with the cloud to get the time and location of the bus continuously using the integrated GPS tracking modules. The motion sensors are used to calculate the crowd density and these values are uploaded to cloud. The combination of these values can be used to predict the future crowd density in the shelter. We use an Arduino IDE to program the Arduino microcontroller and the ESP8266 Wi-Fi module. The ESP8266 is used to provide internet connectivity to the Arduino microcontroller and the GPS module for uploading data to the cloud. The GPS co-ordinates can be fetched by using google API. We use a GoDaddy cloud to store and use the data. GoDaddy is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud.

4.1.1 System Architecture

The Figure 3.1 shows the system architecture and the functions included in the model of the bus shelter. A Raspberry pi board is used to implement the functions like controlling the TFT LCD, pollution and motion sensors and also accessing the cloud. This board is connected to a TFT LCD screen to display all the information that is useful to the commuter. The PIR and the motion sensors are used to detect motion in the shelter and predict crowd density. Pollution sensors are used to provide information on the surrounding weather.

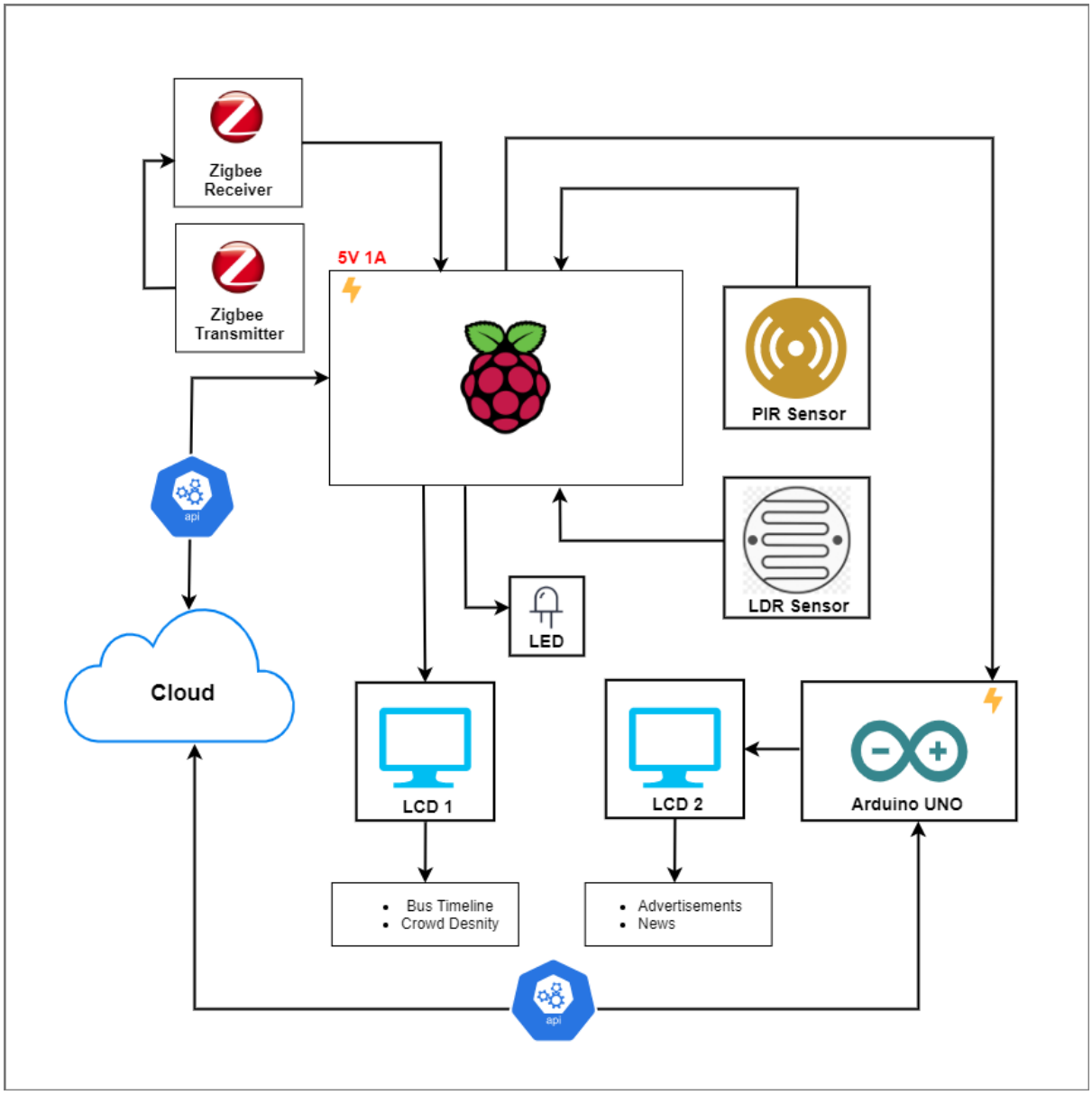


Fig 4.1: System Architecture

Every bus that enters the bus shelter will have an integrated GPS sensor which is controlled by the Arduino board. This Zigbee Transceiver communicates through the cloud to the raspberry pi board telling which bus has entered or left the shelter. All these operations are carried out in the cloud and the output is given to the LCD panel. This LCD screen can display the bus schedule of that particular day and time, advertisements by different businesses, latest global and local news and also the weather information. The smart lighting system of the shelter will turn on the lights only during night time and if there is some commuter in the shelter.

4.1.2 Flow-Diagram of Shelter Display System

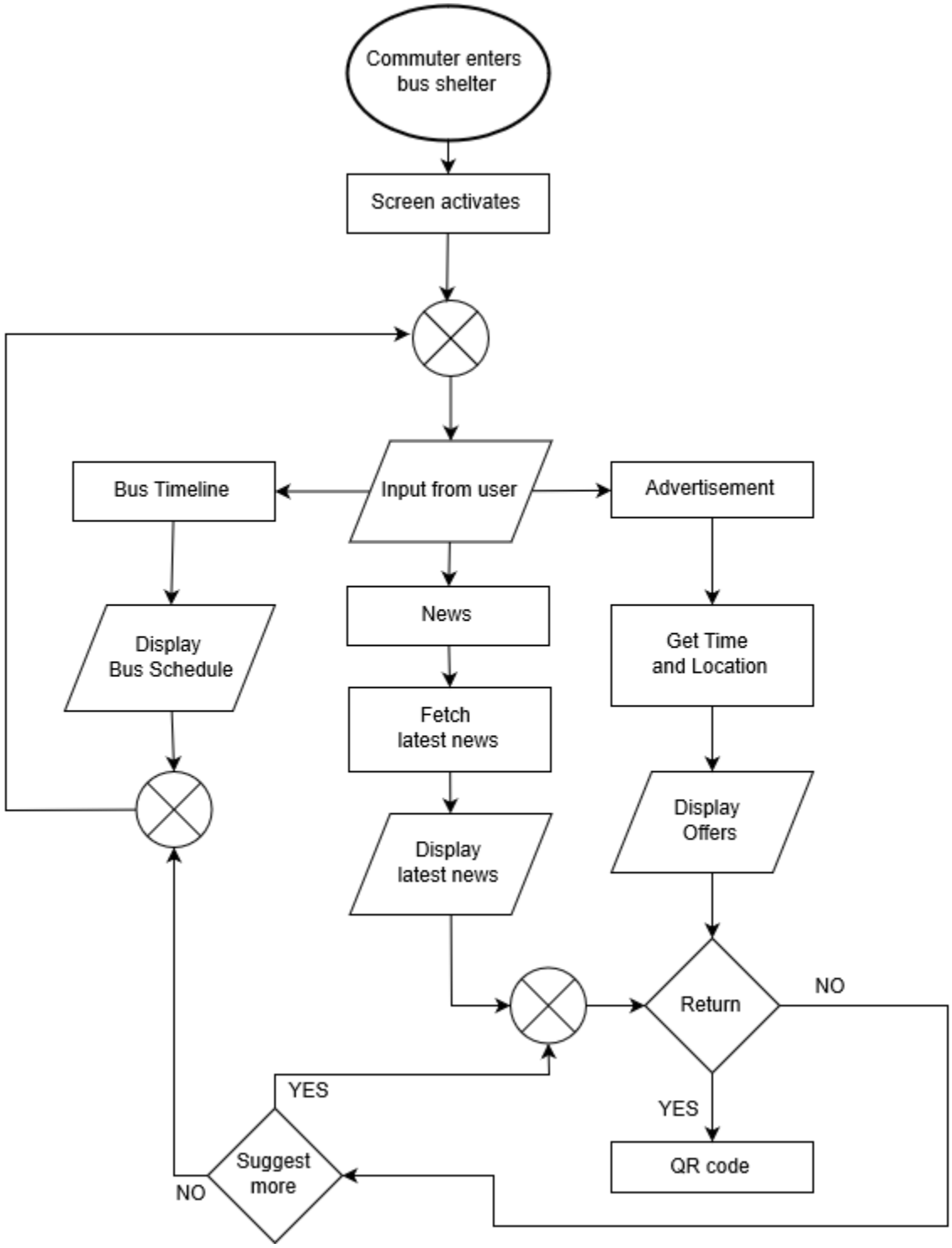


Fig 4.2: Flowchart for Shelter Display System

The figure 4.2 shows the flowchart of the interaction between the commuter and the LCD screen. The LCD screen is awakened as soon as someone walks into the bus shelter. The

movement in the bus shelter is detected by a motion sensor. The screen displays many windows or options for the commuter. This includes bus schedules, news and advertisements when selected displays more information about it. The bus timeline gives information of the next bus that is to arrive with its ETA and the following busses for the day.

4.1.3 SSD for Fetching Advertisements and QR Code

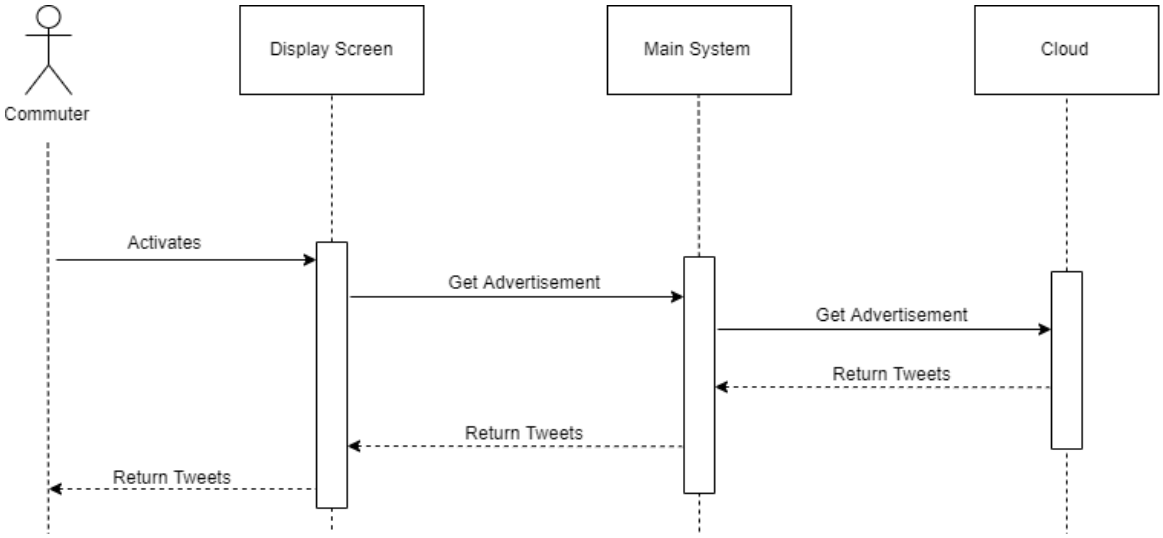


Fig 4.3: SSD for fetching advertisement and QR code

The figure 4.3 is a system sequence diagram for fetching any advertisements and qr code and display it to the user.The screen gets activated when commuter enters the shelter.Input is fetched and sent to main system,then ads are read from repository which acts as buffer.Cloud returns tweets to buffer which reads it and sends it to the main system then acknowledges with tweets to screen for display. Commuter scans the qr code from display screen and gets acknowledged with details about the advertisement in return.

4.1.4 Flow-Diagram of Crowd Density Measure in Shelter

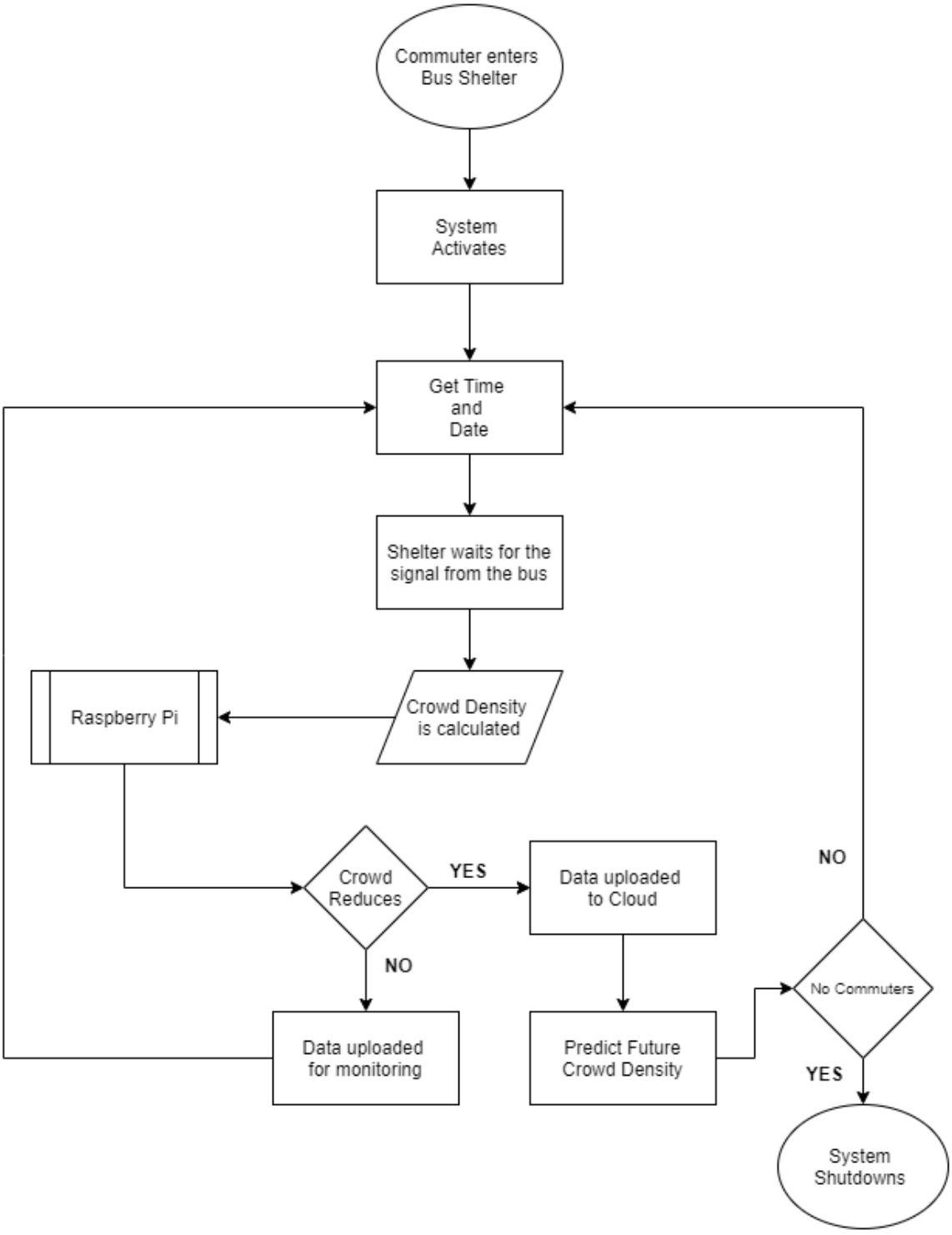


Fig 4.4: Flowchart for crowd density measure

The figure 4.4 shows the flowchart of how the crowd density is calculated in the shelter. When the commuter enters the shelter the PIR sensor detects motion and takes the date and time of

detection and calculates the crowd density. This is done by the Raspberry Pi board and the crowd is continuous monitored for changes. When a bus arrives at the shelter the crowd density is updated, if the crowd density increases then the density is calculated again. If the crowd density decrease the updated value is uploaded to the cloud. If no motion is detected by the sensor, then the system stays idle till motion is detected.

CHAPTER 5

SYSTEM IMPLEMENTATION

Implementation is the phase of the undertaking when the hypothetical configuration is transformed out into a working framework. Implementation should be perfect mapping of the design document in a suitable programming language in order to achieve the necessary final product. Often the product is ruined due to incorrect programming language chosen for implementation or unsuitable method of programming. It is better for the coding phase to be directly linked to the design phase in the sense if the design is in terms of object oriented terms then implementation should be preferably carried out in an object oriented way.

The implementation involves:

1. Careful planning.

2. Investigation of the current system and the constraints on implementation.

3. Training of staff in the newly developed system.

In this stage, the design or design changes are introduced and made operational in a specific situation. Implementation of any software is always preceded by important decisions regarding selection of the platform, the language used, etc. these decisions are often influenced by several factors such as real environment in which the system works, the speed that is required, the security concerns, and other implementation specific details. There are three major implementation decisions that have been made before the implementation of this project.

They are as follows:

1. Selection of the platform (Operating System).

2. Selection of the programming language for development of the application.

3. Coding guideline to be followed.

Implementation of any product is constantly continued by critical choices with respect to choice of the stage, the dialect utilized, and so forth these choices are frequently affected by a few elements, for example, genuine environment in which the framework meets expectations, the

velocity that is needed, the security concerns, and other execution choices that have been made before the usage of this venture.

5.1 Software Description

This section includes all the programming languages and IDE used for building an application. It briefly describes about Python IDE and C++.

5.1.1 Python Description

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, Bit Torrent, etc.

Python is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Characteristics of Python

i. It provides rich data types and easier to read syntax than any other programming languages
ii. It is a platform independent scripted language with full access to operating system API's
iii. Compared to other programming languages, it allows more run-time flexibility
iv. It includes the basic text manipulation facilities of Perl and Awk
v. A module in Python may have one or more classes and free functions
vi. Libraries in Pythons are cross-platform compatible with Linux, MacIntosh, and Windows.

Thonny IDE

Thonny is an integrated development environment for Python that is designed for beginners. It supports different ways of stepping through the code, step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap.

The following are some of the primary features of Thonny:

i. It autocompletes code.

ii. It inspects code to provide bracket matching and highlight errors.

iii. It is easy to start with as its installer also installs Python 3.7.

iv. Its debugger is simple to use as no knowledge of breakpoints is required.

v. It enables users to step into a function call by providing details about local variables and displaying the code pointer.

vi. It has an easy interface to install packages. This makes it very suitable for beginners.

Thonny is an integrated development environment (IDE) for Python bundled with the latest version of the Raspbian with BUSTER operating system. It supports different ways of stepping through the cod step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap. Using Th01m y, it's now much easier to learn to code. Thonny comes with Python 3.6 built in, so you don' t need to install anything. It offers a lot of advanced features not currently available in the Python 3 (IDLE) program, which is still included with Raspbian. Thonny has a range of additional features that are perfect for learning programming. One of the best features is a powerful, but easy-to-us, debug mode. Instead of running your program, it steps through the code line by line. You can see the variables and objects being created, and values being passed into functions or assessed by comparators.

Pros

i. It has an easy-to-use user interface.

ii. The user interface does not contain any distractions for beginners.

Cons

i. It offers basic functionality as opposed to other advanced IDEs (i.e., PyCharm).

ii. Users may encounter some issues for which a quick fix is not available.

5.1.2 C++

C++ is a middle-level programming language developed by Bjarne Stroustrup starting in 1979 at Bell Labs. **C++** runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. This **C++** tutorial adopts a simple and practical approach to describe the concepts of **C++** for beginners to advanced software engineers.

The **Arduino Integrated Development Environment (IDE)** is a cross- platform application (for Windows, macOS Linux) that is written m functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

Arduino code is written in **C++** with an addition of special methods and functions. **C++** is a human-readable programming language. When you create a 'sketch' (the name given to Arduino code files), it is processed and compiled to machine language.

Features of C++

i. Rich Library

ii. Platform Dependent

iii. Object Oriented

iv. Case Sensitive

v. Compiler Based

vi. Syntax based language

5.2 Hardware Description

5.2.1 Raspberry PI

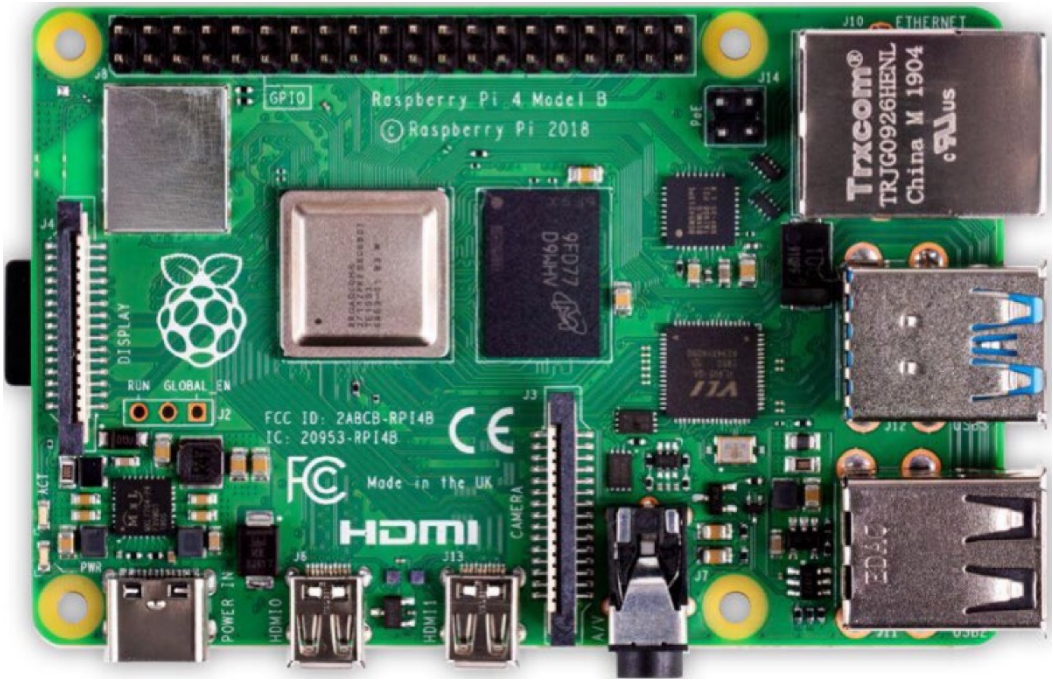


Fig 5.1: Motherboard of RaspberryPi

The Raspberry Pi is a low cost **credit-card sized computer** that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enable les people of all ages to explore computing, and to learn how to program in languages like scratch and Python. It’s capable of doing everything you' d expect a desktop computer to do, from browsing the ins met and playing high-definition video, to making spreadsheets, word-processing and playing games.

The Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects from music machines and parent detectors to weather stations and tweeting birdhouses with infrared cameras.

5.2.2 PIR Sensor

PIR sensors allow you to s ns motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason, they are commonly found in appliances and gadgets used in homes

or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

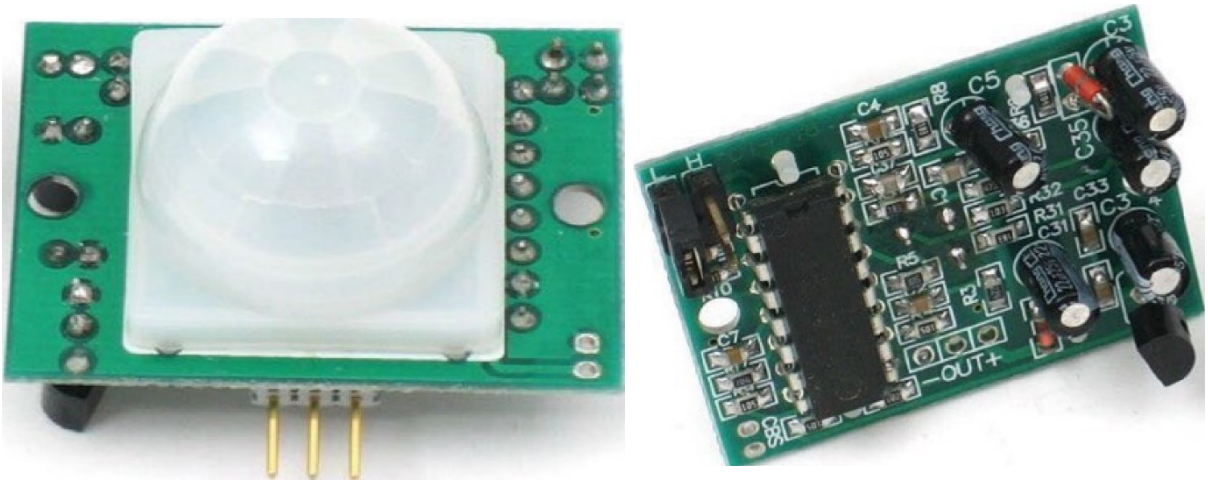


Fig 5.2: PIR sensor

PIRs are basically made of a pyroelectric sensor, which can detect levels of infrared radiation. Everything emits some low-level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

5.2.3 LDR Sensor

An LDR or light dependent resistor is also known as photo resistor, photocell, photoconductor. It is a one type of resistor whose resistance varies depending on the amount of light falling on its surface. When the light falls on the resistor, then the resistance changes. These resistors are often used in many circuits where it is required to sense the presence of light. These resistors have a variety of functions and resistance. For instance, when the LDR is in darkness, then it can be us d to turn O a light or to turn OFF a light when it is in the light. A typical light d pendent resistor has a resistance in the darkness of 1MOhm, and in the brightness a resistance of a couple of K Ohm.

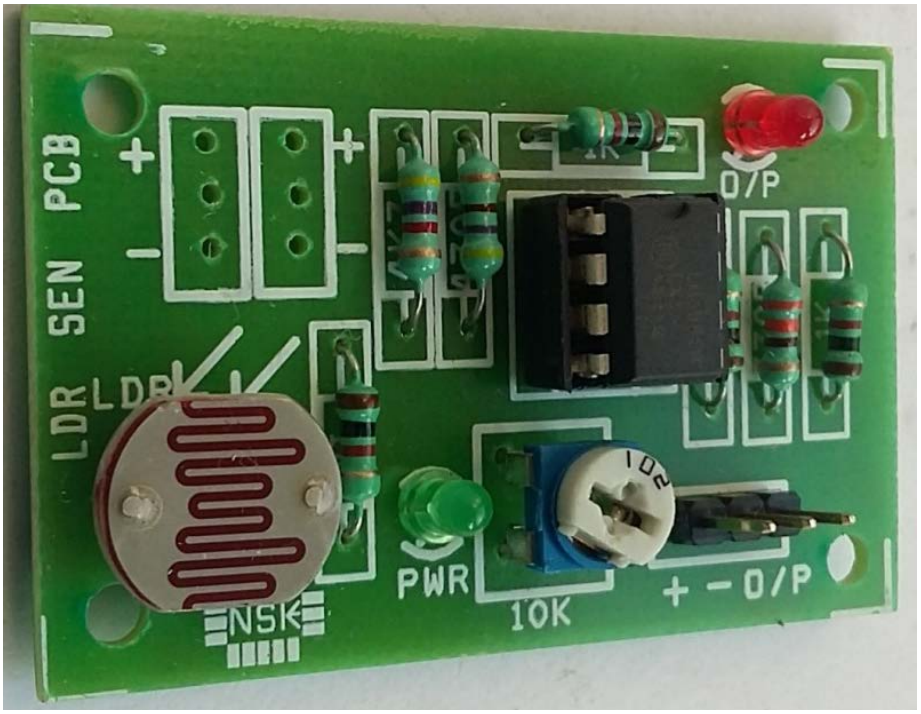


Fig 5.3: LDR sensor

This resistor works on the principle of photo conductivity. These devices depend on the light, when light falls on the LDR then the resistance decreases, and increases in the dark. When an LDR is kept in the dark place, its resistance is high and, when the LDR is kept in the light its resistance will decrease.

5.2.4 LED Output

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

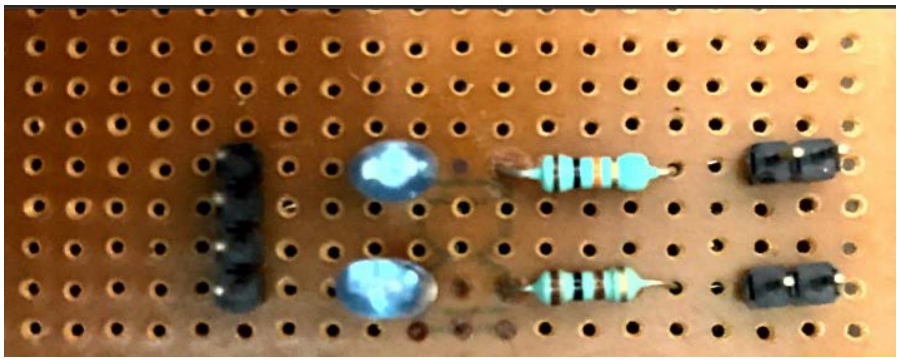


Fig 5.4: LED output

The circuit is implemented by connecting 1K Ohm resistor to the LED and it has 2 setups of each LED circuitry. There are 2 connector pins; one end is for providing the input voltage to the LED and other end is for grounding.

5.2.5 Arduino Uno

Fig 5.5: Motherboard of Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53 -R0), 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 cab! or power it with a AC-to-DC adapter or battery to get started. "Uno"

means on 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.

5.2.6 LCD Panel

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

Fig 5.6: LCD Panel

16x 2 LCD is named so because; it has 16 Columns m1d 2 Rows. There me a lot of combinations available like, 8>< 1, 8x2, 1Ox 2, 16x1, etc. but the most used one is the 16>< 2 LCD. So, it will have (16><2=32) 32 characters in total and each character will be made of 5x 8 Pixel Dots. ow, we know that each character has (S x8= 40) 40 Pixels and for 32Characters we will have (32x40) 1280 Pixels. Further, the LCD should also be instructed about the Position of the Pixels. Hence it will be a hectic task to handle everything with the help of MCU, hen ce an Interface IC like HD44780is used, which is mounted on the backside of the LCD Module itself. The function of this IC is to get the Commands and Data from the MCU and process them to display meaningful information onto our LCD Screen.

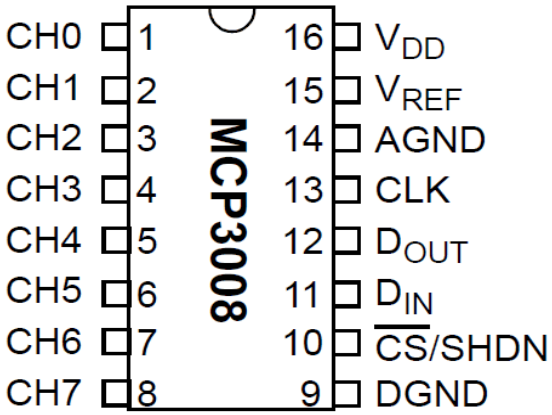
5.2.7 MCP 3008

The MCP3008 is a low cost 8-channel 10-bit analog to digital converter. The precision of this ADC is similar to that of an Arduino Uno, and with 8 channels you can read quite a few analog

signals from the Pi. This chip is a great option if you just need to read simple analog signals, like from a temperature or light sensor. If you need more precision or features, check out the ADS1x115 series on the next page.

Before you use the MCP3008 it will help to skim this older Raspberry Pi MCP3008 guide for more information about using it with the Raspberry Pi. However, don't use the code from the older guide as it's deprecated. This guide will show you an easier way to install and use new Python code to talk to the MCP3008 ADC.

The MCP3008 datasheet is also an important resource to skim and have handy.



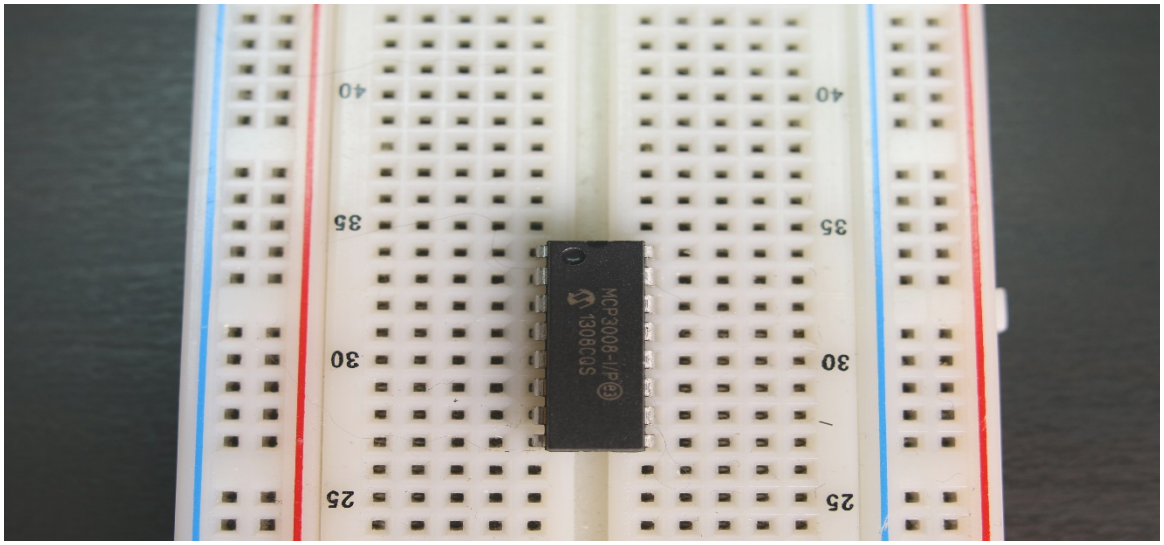


Fig 5.7: MCP 3008 chip

5.2.8 ZigBee Transmitter and Receiver

Transceiver Module provides bi-directional communication between two-way enabled RTI remote controls and control processors utilizing ZigBee wireless communication. Capable of being hard-wired directly to a control processor or be used as a wireless repeater device to create an ultra-reliable, self-healing ZigBee network.

The ZM-24 does not directly control A/V components. It must be used in conjunction with a ZigBee enabled processor.

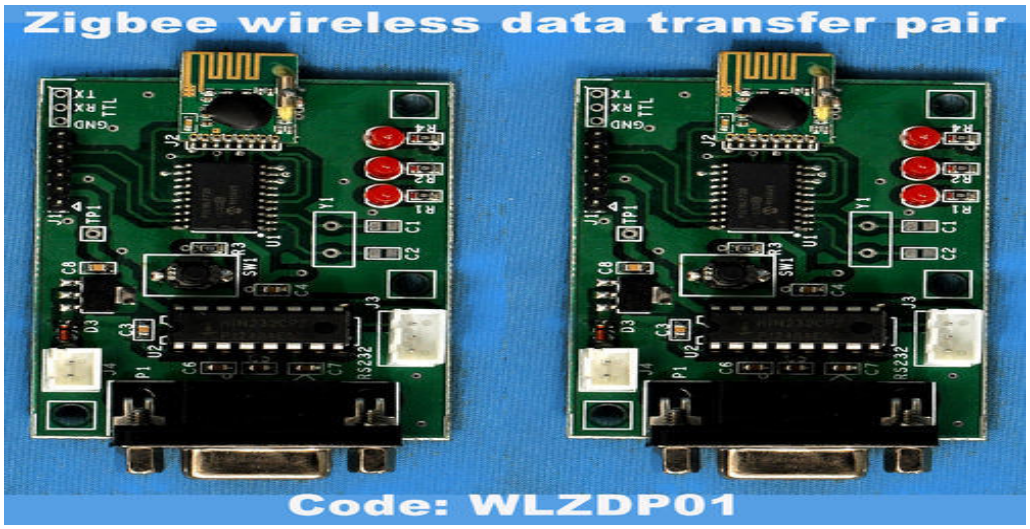


Fig 5.8: ZigBee Transmitter and Receiver

5.3 Methodology

5.3.1 Raspberry Pi – PIR – MCP Interfacing

The figure 5.9 shows the Raspberry Pi motherboard interfaced with the MCP 3008 board to intern connect to the Digital to Analog converter (DAC). This DAC is used to convert digital signals to analog signals. The DAC is used converter the binary output given by the PIR sensor into analog signal for the motherboard to calculate the density of the crowd.

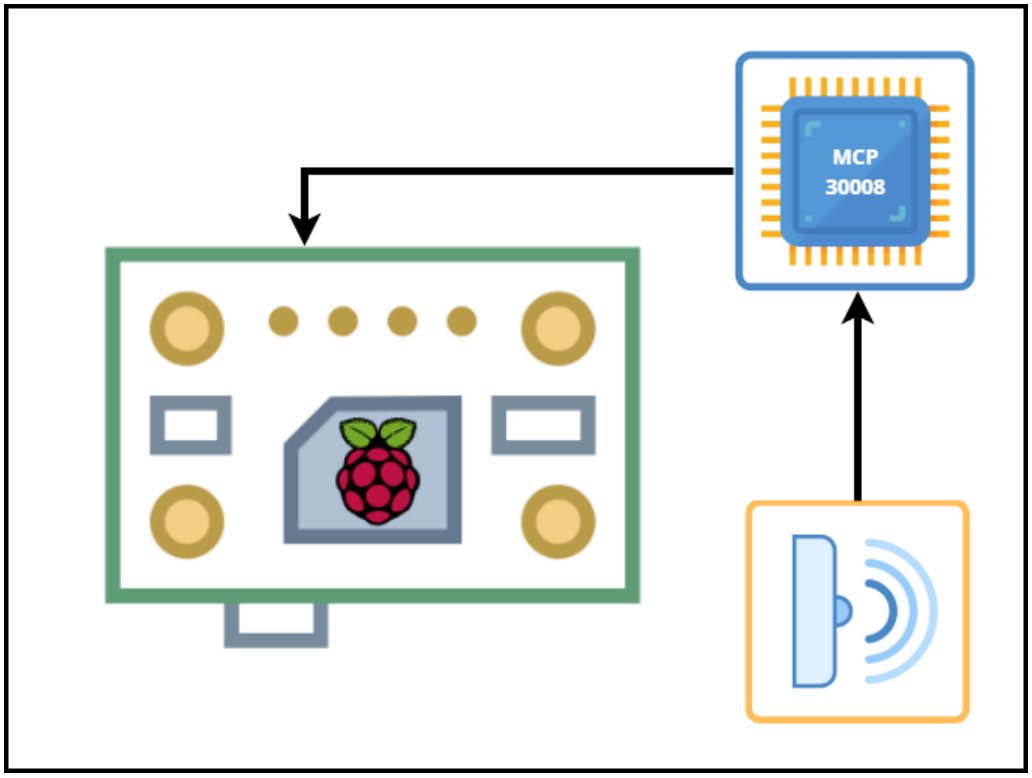


Fig 5.9: Raspberry Pi – PIR – MCP Interfacing

5.3.2 Raspberry Pi – LDR-LED Interfacing

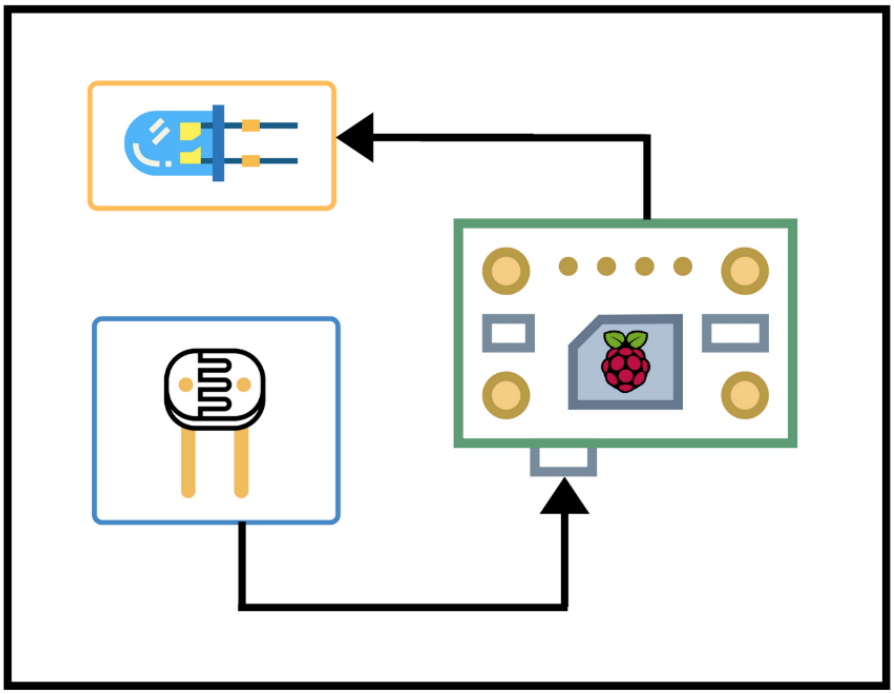


Fig 5.10: Raspberry Pi – LDR-LED Interfacing

The figure 5.10 shows the LDR-LED interfacing for light detection. When its day time the LDR detects light and the LED is not turned on. When night time the LDR senses no light and the LED is turned on to light the shelter.

5.3.3 Raspberry Pi – LCD Interfacing

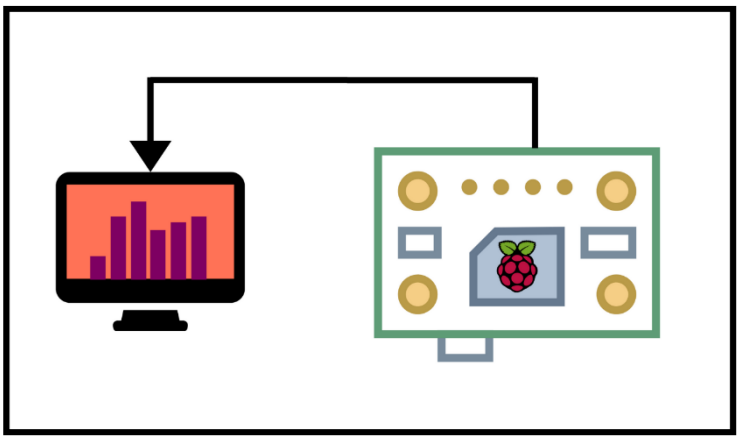


Fig 5.11: Raspberry Pi – LCD Interfacing

The figure 5.11 shows the Raspberry Pi motherboard interfaced with LCD. This LCD is used to display the bus information and also display the estimated crowd density in the shelter. All the data are stored and fetched from the cloud.

5.3.4 Arduino UNO – LCD Interfacing

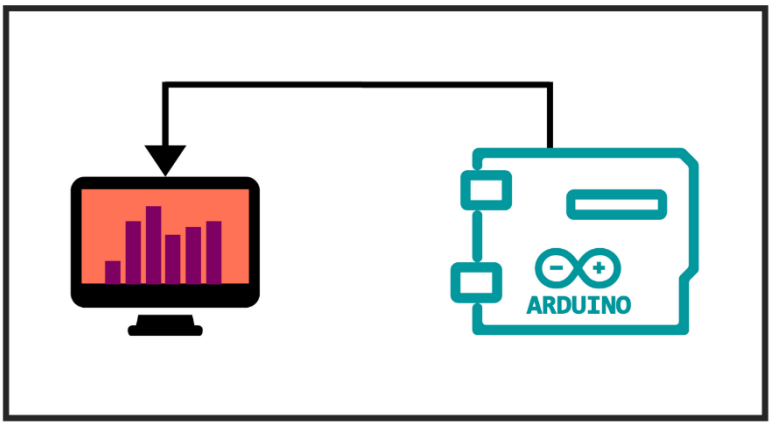


Fig 5.12 Arduino UNO – LCD Interfacing

The figure 5.12 shows Arduino UNO board interfaced with another LCD. This LCD is used to display advertisements, offers, business advertisements, etc. All the data are stored and fetched from the cloud.

5.3.5 Zigbee Transceiver Interfacing

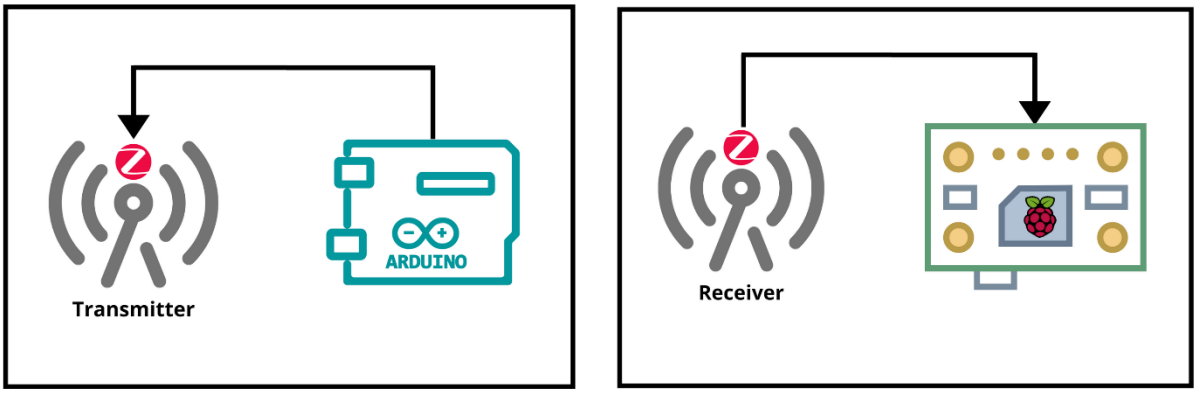


Fig 5.13: Zigbee Transceiver Interfacing

The figure 5.12 shows the zigbee transceiver. The zigbee transmitter is interfaced with the Arduino board and is integrated in the buses. The zigbee receiver is interfaced with the Raspberry Pi board and is integrated in the shelter. When the bus arrives at the shelter the receiver at the shelter receives a signal from the zigbee transmitter module in the bus which tells us the information of a particular bus.

CHAPTER 6

TESTING

Testing is an essential stage in the advancement life cycle of the item. This is the stage, where the remaining lapses, if any, from all the stages are identified. Thus testing performs an extremely discriminating part for quality certification and guaranteeing the dependability of the product. Amid the testing, the system to be tried was executed with a situated of experiments and the yield of the project for the experiment was assessed to figure out if the project was executing of course. Slips were discovered and adjusted by utilizing the beneath expressed testing steps and remedy was recorded for future references.

Consequently, a progression of testing was performed on the framework, before it was prepared for usage. It is the procedure used to help recognize the accuracy, fulfilment, security, and nature of created PC programming. Testing is a procedure of specialized examination, performed for the benefit of partners, i.e. proposed to uncover the quality- related data about the item as for connection in which it is planned to work. This incorporates, however is not restricted to, the procedure of executing a project or application with the goal of discovering lapses.

There are numerous ways to deal with programming testing, yet viable testing of complex items is basically a procedure of examination not only a matter of making and taking after routine method. Albeit a large portion of the scholarly procedures of testing are almost indistinguishable to that of audit or investigation, the word testing is indicated to mean the dynamic examination of the item putting the item through its paces. A portion of the normal quality traits incorporate capacity, unwavering quality, productivity, versatility, viability, similarity and ease of use.

6.1 Software Testing Introduction

Software testing is a process used to help identify the correctness, completeness and quality of developed computer software. Software testing is the process used to measure the quality of developed software. Testing is the process of executing a program with the intent of finding errors. Software testing is often referred to as verification & validation.

6.2 Explanation for SDLC & STLC

SDLC: The software development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

6.3 Phases of Software Development

- Requirement Analysis
- Software design
- Development or Coding
- Testing
- Maintenance

6.3.1 Requirement Analysis

The requirements of a desired software product are extracted. Based the business scenario the SRS (Software Requirement Specification) document is prepared in this phase.

6.3.2 Design

Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications, and security issues for the software. Design phase is concerned with making sure the software system will meet the requirements of the product. There are 2 stages in design,

- HLD – High Level Design
- LLD – Low Level Design

HLD – gives the architecture of the software product to be developed and is done by architects and senior developers.

LLD – done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code.

6.3.3 Testing

Testing is evaluating the software to check for the user requirements. Here the software is evaluated with intent of finding defects.

6.3.4 Maintenance

Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up-to date concerning the latest modifications and procedures.

6.4 STLC (Software Testing Life Cycle)

Testing itself has many phases i.e. is called as STLC. STLC is part of SDLC

- Test Plan
- Test Development
- Test Execution
- Analyze Results
- Defect Tracking
- Summaries Report

6.4.1 Test Plan

It is a document which describes the testing environment, purpose, scope, objectives, test strategy, schedules, mile stones, testing tool, roles and responsibilities, risks, training, staffing and who is going to test the application, what type of tests should be performed and how it will track the defects.

6.4.2 Test Development

Preparing test cases, test data, preparing test procedure, preparing test scenario, Writing test script.

6.4.3 Test Execution

In this phase, execute the documents those are prepared in test development phase.

6.4.4 Analyze Result

Once executed documents will get results either pass or fail. Need to analyze the results during this phase.

6.4.5 Defect Tracking

When we get defect on the application then need to prepare the bug report file and forwards to Test Team Lead and Dev Team. The Dev Team will fix the bug. Again have to test the application. This cycle repeats untill getting the software without defects.

6.5 Types of Testing

- White Box Testing
- Black Box Testing
- Grey box testing

6.5.1 White Box Testing

White box testing as the name suggests gives the internal view of the software. This type of testing is also known as structural testing or glass box testing as well, as the interest lies in what lies inside the box.

6.5.2 Black Box Testing

It's also called as behavioral testing. It focuses on the functional requirements of the software. Testing either functional or non-functional without reference to the internal structure of the component or system is called black box testing.

6.5.3 Grey Box Testing

Grey box testing is the combination n of black box and white box testing. Intention of this testing is to find out defects related to bad design or bad implementation of the system.

6.6 Level of Testing Used in Project

6.6.1 Unit Testing

Initialization testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable.

6.6.2 Integration Testing

All modules which make application are tested. Integration testing is to make sure that the interaction of two or more components produces results that satisfy functional requirement.

6.6.3 System Testing

To test the complete system in terms of functionality and non-functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

6.6.4 Functional Testing

The outgoing links from all the pages from specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields.

Wrong inputs to the fields in the forms.

6.6.5 Alpha Testing

Alpha testing is final testing before the software is released to the general public. This testing is conducted at the developer site and in a controlled environment by the end user of the software.

6.6.6 Beta Testing

The beta test is conducted at one or more customer sites by the end user of the software. The beta test is conducted at one or more customer sites by the end user of the software.

6.7 Unit Testing Cases

Initialization testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable.

Testing with Test Cases

Each program components are tested for errors to discover defects in the testing stage. The components may be any of the program functions, objects or modules. The integrated components are used in system testing to form the complete system. In this stage testing must be focused to establish the system that meets functional requirements and must be ensured that system does not behave in an unexpected way.

Test data are inputs that are been devised and trained to test the system whereas test cases are inputs used to test the system and if the system works as specified with the given input the output is specified for the given input, the behavior is examined in a cohesive system. The test cases are opted for ensuring that the system behavior is examined in all the possible combinations of conditions considered.

Accordingly, system behavior that is expected under various combinations of conditions is given. Therefore, test cases are selected which have inputs and the outputs on expected lines, inputs that are not valid and for which suitable messages must be given and inputs that do not occur frequently which can be regarded as special cases. In this chapter, several test cases have been explained with the underlying the proposed techniques.

Testing Strategy, the strategy that is used to perform unit testing is described below:

• Features to be tested:

The features to be tested, most importantly include the operation of individual component for the proper execution of the entire program.

• Items to be tested:

The items to be tested include all the individual units or functions, which collectively form the whole system. In case of unit testing the items to be tested, are the main graphical user interface, deploying the sensor nodes and handling the events in the sensor network. Purpose of testing: The purpose of the testing is to check the unit functionality of the main source of the project.

• Pass/Fail Criteria:

The pass or fail criteria are designed with the basis of appropriate compliance of main source file.

TEST CASES

Test Case 1	
Name of Test	Lighting System.
Input	Commuter enters the bus shelter
Expected output	Lights in the shelter turn on.
Actual result	Same as expected.
Remarks	Successful
Table 6.1 Test case 1 for lighting system	
Test Case 2	
Name of Test	LCD System.
Input	Commuter enters the shelter.
Expected Result	LCDs turn on.
Actual output	Same as expected.
Remarks	Successful
Table 6.2 Test case 2 for LCD	

Test Case 3	
Name of Test	Bus Transmitting
Input	Transmitter on the bus sends signals.
Expected Result	Shelter detects the arrival of bus.
Actual output	Same as expected.
Remarks	Successful

Table 6.3 Test case 3 for Bus Transmitter

Test Case 4	
Name of Test	Shelter Receiving
Input	Signals from the bus.
Expected Result	Shelters receiver detects the signal of bus.
Actual output	Same as expected.
Remarks	Successful

Table 6.4 Test case 4 for Shelter receiving

Test Case 5	
Name of Test	Cloud Uploading
Input	Density of the shelter.
Expected Result	Data is getting stored in the cloud.
Actual output	Same as expected.
Remarks	Successful

Table 6.5 Test case 5 for Cloud uploading

Test Case 6	
Name of Test	ML Algorithm
Input	Time and Density values.
Expected Result	Predicting the density for future in shelter
Actual output	Same as expected.
Remarks	Successful

Table 6.6 Test case 6 for ML Algorithm

Test Case 7	
Name of Test	Dashboard.
Input	Density and Time from the shelter.
Expected Result	Predicted density is being shown.
Actual output	Same as expected.
Remarks	Successful

Table 6.7 Test case 7 for Dashboard

CHAPTER 7

MACHINE LEARNING, CLOUD and CONNECTED BUSES

7.1 Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning algorithms are often categorized as supervised or unsupervised.

1. Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
2. In contrast, unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
3. Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy.

Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

4. Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

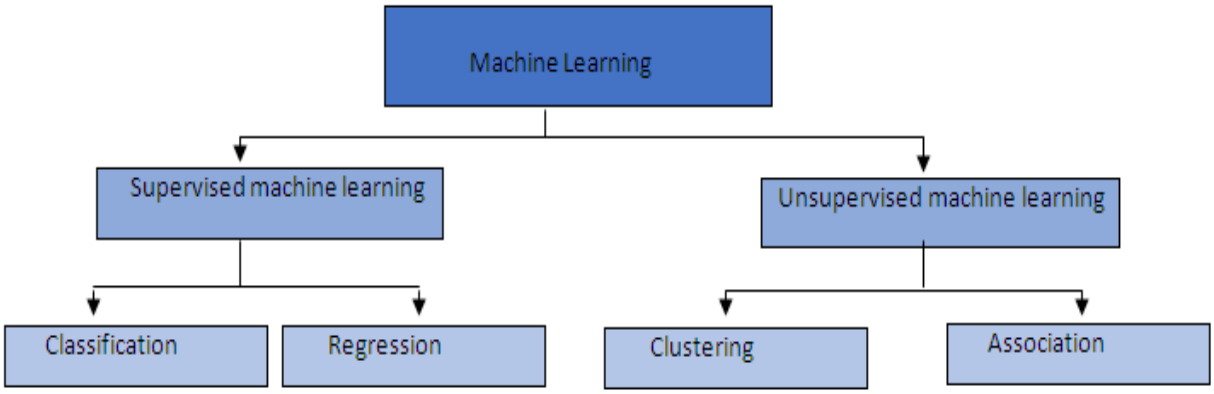


Fig 7.1 Types of ML

7.1.1 Supervised Learning

Regression and classification are categorized under the same umbrella of supervised machine learning. Both share the same concept of utilizing known datasets (referred to as training datasets) to make predictions. In supervised learning, an algorithm is employed to learn the mapping function from the input variable (x) to the output variable (y); that is y = f(X).

The objective of such a problem is to approximate the mapping function (f) as accurately as possible such that whenever there is a new input data (x), the output variable (y) for the dataset can be predicted.

The difference between regression machine learning algorithms and classification machine learning algorithms sometimes confuse most data scientists, which make them to implement wrong methodologies in solving their prediction problems.

The main difference between them is that the output variable in regression is numerical (or continuous) while that for classification is categorical (or discrete).

7.1.2 Regression Analysis

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression. Regression analysis is an important tool for modelling and analyzing data. Here, we fit a curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized. I'll explain this in more details in coming sections. Let's say, you want to estimate growth in sales of a company based on current economic conditions. You have the recent company data which indicates that the growth in sales is around two and a half times the growth in the economy. Using this insight, we can predict future sales of the company based on current & past information.

There are multiple benefits of using regression analysis. They are as follows:

1. It indicates the significant relationships between dependent variable and independent variable.
2. It indicates the strength of impact of multiple independent variables on a dependent variable.

Regression analysis also allows us to compare the effects of variables measured on different scales, such as the effect of price changes and the number of promotional activities. These benefits help market researcher's / data analyst's / data scientists to eliminate and evaluate the best set of variables to be used for building predictive models.

7.1.3 Logistic Regression

Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

$$1 / (1 + e^{-value})$$

Where e is the base of the natural logarithms (Euler's number or the EXP () function in your spreadsheet) and value is the actual numerical value that you want to transform. Below is a plot of the numbers between -5 and 5 transformed into the range 0 and 1 using the logistic function.

Logistic regression uses an equation as the representation, very much like linear regression.

Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary value (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)})$$

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

7.1.4 Implementation of the Algorithm

1. From the shelter the time in hours and minutes are continuously being measured are sent to cloud.

2. The sensor values of the density are also being stored in the cloud.

3. The Logistic Regression algorithm uses these 3 attributes as input for generating a model.

4. It takes hours, minutes and density values and then generates the predicted values of the density.

5. This predicted value is being displayed on the mobile app which any commuter can see.

6. The predicted data is also stored in the cloud and hence that can be used by the authorities to estimate the bus demand and hence take necessary actions to increase the bus frequency.

7. By doing so, the density in the bus shelter and buses will be reduced.

The efficiency of the algorithm is approximately 86%.

7.2 Cloud Storage

Cloud storage is a service model in which data is transmitted and stored on remote storage systems, where it is maintained, managed, backed up and made available to users over a network. Users generally pay for their cloud data storage on a per-consumption, monthly rate. Although the per-gigabyte cost has been radically driven down, cloud storage providers have added operating expenses that can make the technology considerably more expensive to use. The security of cloud storage services continues to be a concern among users. Service providers have tried to allay those fears by enhancing their security capabilities by incorporating data encryption, multi-factor authentication and improved physical security into their services

7.2.1 Types of cloud storage

There are three main cloud-based storage access models: public, private and hybrid.

7.2.2 Public cloud storage

Public cloud storage services provide a Multi-tenant storage environment that is most suited for unstructured data on a subscription basis. Data is stored in the service provider’s data centers with storage data spread across multiple regions or continents. Customers generally pay on a per-use basis similar to the utility payment model; in many cases, there are also transaction charges based on frequency and the volume of data being accessed.

7.2.3 Private cloud storage

Private cloud storage service is provided by in-house storage resources deployed as a dedicated environment protected behind an organization's firewall. Internally hosted private cloud storage implementations emulate some of the features of commercially available public cloud

services, providing easy access and allocation of storage resources for business users, as well as object storage protocols. Private clouds are appropriate for users who need customization and more control over their data, or who have stringent data security or regulatory requirements.

7.2.3.1 Implemented private cloud storage

GoDaddy: GoDaddy is the web hosting platform which is used for hosting website. It has hosted more than nineteen million websites since its birth and is doing great with it. With mixed reviews posted, it is considered the best domain hosting service providers as being a lead company, it is not possible to satisfy everyone. GoDaddy provides great customer support and helps business owners to manage the business website.

7.2.4 Hybrid cloud storage

Hybrid cloud storage is a mix of private cloud storage and third-party public cloud storage services with a layer of orchestration management to integrate operationally the two platforms. The model offers businesses flexibility and more data deployment options. An organization might, for example, store actively used and structured data in an on-premises cloud, and unstructured and archival data in a public cloud. A hybrid environment also makes it easier to handle seasonal or unanticipated spikes in data creation or access by "cloud bursting" to the external storage service and avoiding having to add in-house storage resources. In recent years, there has been increased adoption of the hybrid cloud model.

7.2.5 Implemented cloud architecture

We have used private cloud storage in this project security of the data in the smart bus shelter system. The entire data is securely uploaded to the private cloud and stored in the database. Various objects like mobile application and raspberry pi are connected to the private cloud using a PHP as a scripting language for uploading and fetching the data.

The raspberry pi simultaneously sends the sensor values to the cloud and stores it in the database for providing data to the ML algorithm. The ML algorithm uses all the input data from the database provided by the sensors and predicts the future density in the shelter. The mobile application is simply connected to the private cloud through a PHP as a scripting language and fetches the predicted density in the bus shelter and displays appropriate messages.

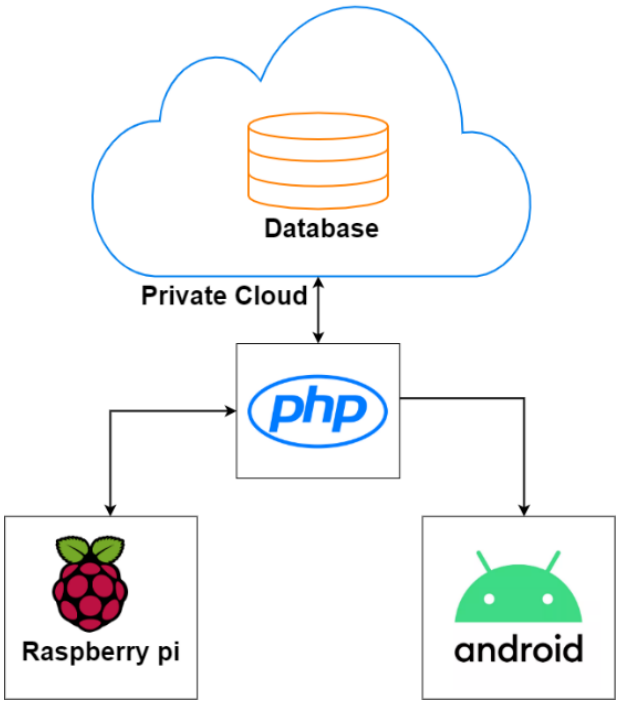


Fig 7.2 Implementation of cloud architecture

7.2.5.1 Working of cloud architecture

1. The pretrained dataset is stored in the database on a private cloud.
2. PHP scripts are used for establishing a connection between the raspberry pi and android mobile application to the cloud.
3. The raspberry pi sends the continuous sensor values to the cloud through a URL and the data is stored in the database.
4. The android mobile application uses PHP to establish a connection to the cloud and fetches the data using a URL and displays appropriate messages.

7.3 Connected Buses

The core idea of the connected buses is based on the fundamental laws of a transmitter and receiver. The notion of connected buses here in the project means that the buses will be connected to the shelter via T-R system and not the inter bus connection itself.

7.3.1 Transmitter

Transmitters are devices that are used to send out data as radio waves in a specific band of the electromagnetic spectrum in order to fulfill a specific communication need, be it for voice or for general data. In order to do this, a transmitter takes energy from a power source and transforms this into a radio frequency alternating current that changes direction millions to

billions of times per second depending on the band that the transmitter needs to send in. When this rapidly changing energy is directed through a conductor, in this case an antenna, electromagnetic or radio waves are radiated outwards to be received by another antenna that is connected to a receiver that reverses the process to come up with the actual message or data.

7.3.2 Receiver

A receiver mostly refers to that part of a device that receives signals; often, the device acts as both a transmitter and a receiver (transceiver) such as in the case of cell phones (cellular radio) and antennas used for data communication. If both the transmitter and the receiver are in the same locality, the transmission medium would usually be cables or wire, but wireless signals are also viable to allow for a broadcast method of transmission to multiple receivers. An example of a receiver is the transceiver module, which also serves as a transmitter for bi-directional communication of a terrestrial radio installation or cellular tower. It uses its transceiver to send signals to a cell phone such as voice, text messages and data, and, in return, it receives the same kinds of signals from a phone to be retransmitted and received by other towers until they reach their final destination. The same applies to the communication between a Wi-Fi router and a laptop or mobile device; signals are transmitted and received bi-directionally.

7.3.3 Zigbee Transceivers

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Hence, Zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network. The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or more general wireless networking such as Wi-Fi. Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that requires short-range low-rate wireless data transfer.

Its low power consumption limits transmission distances to 10–100 meters' line-of-sight, depending on power output and environmental characteristics. [2] Zigbee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to

reach more distant ones. Zigbee is typically used in low data rate applications that require long battery life and secure networking (Zigbee networks are secured by 128-bit symmetric encryption keys.) Zigbee has a defined rate of 250 kbit/s, best suited for intermittent data transmissions from a sensor or input device.

7.3.4 The Hypothetical Module

The figure below shows the diagrammatical module of how the bus and shelter looks once they are configured with their respective components (transmitter and receiver) so send and receive signals respectively.

Fig 7.3 Connected Bus Module

CHAPTER 8

RESULT AND ANALYSIS

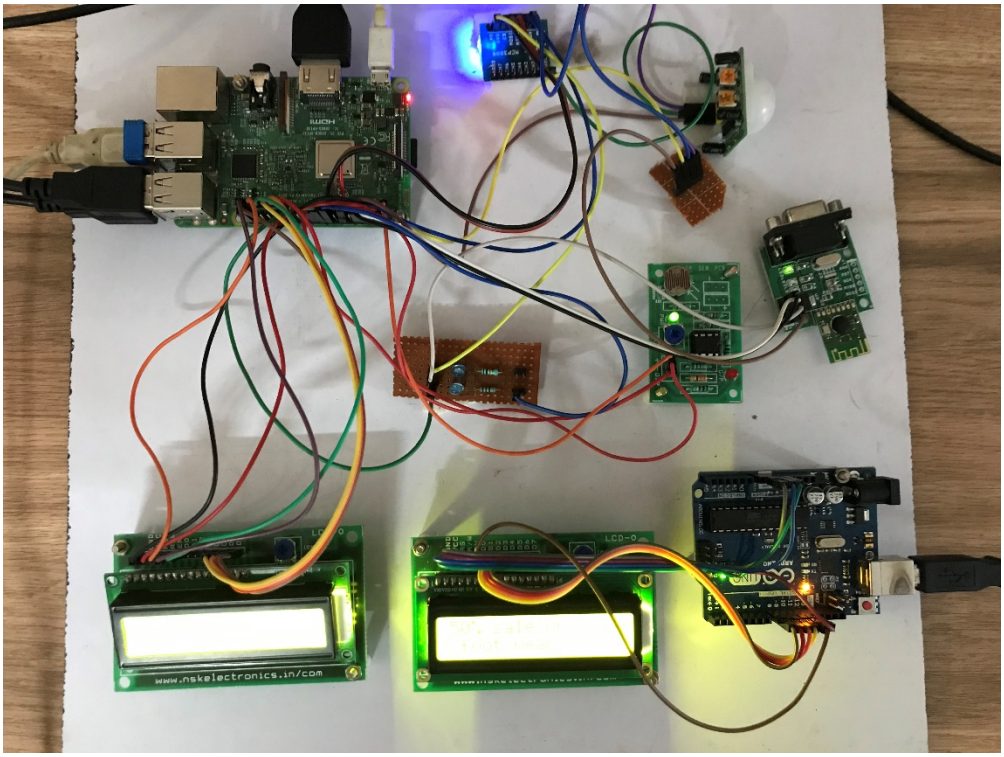


Fig 8.1 Bus Shelter Implemented Architecture

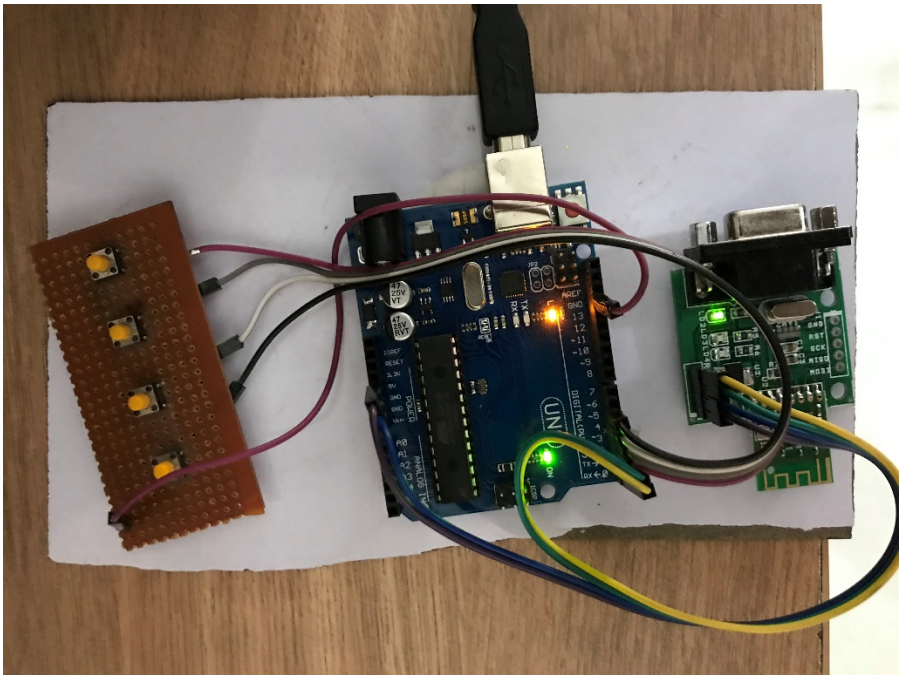


Fig 8.2 Bus Implemented Architecture

Fig 8.3 LCD to display offers

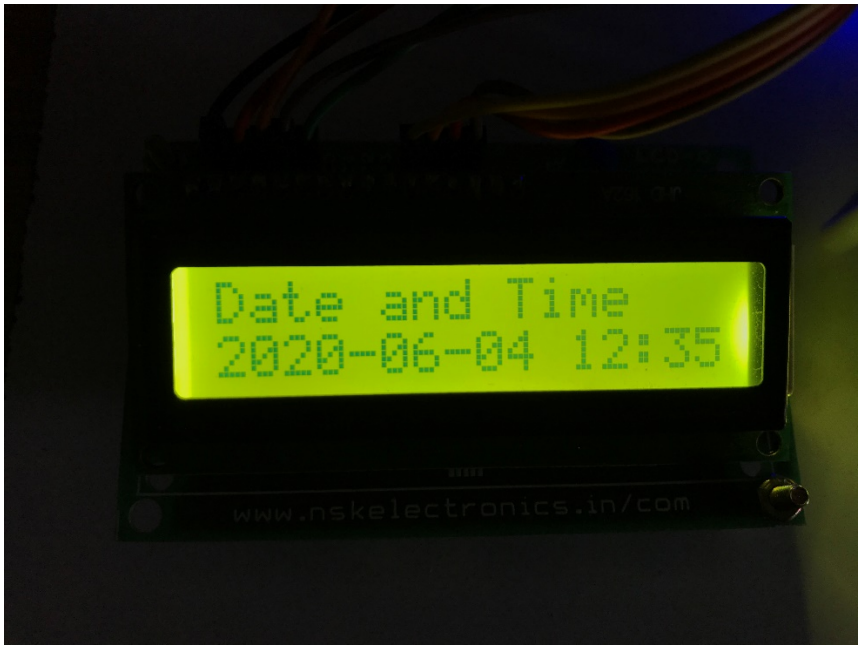


Fig 8.4 LCD to display date and time

Fig 8.5 LCD to display the bus timeline

Fig 8.6 LCD to display the density of the bus

Fig 8.7 LCD to display detect the arrival of one bus and prompt the wait for another

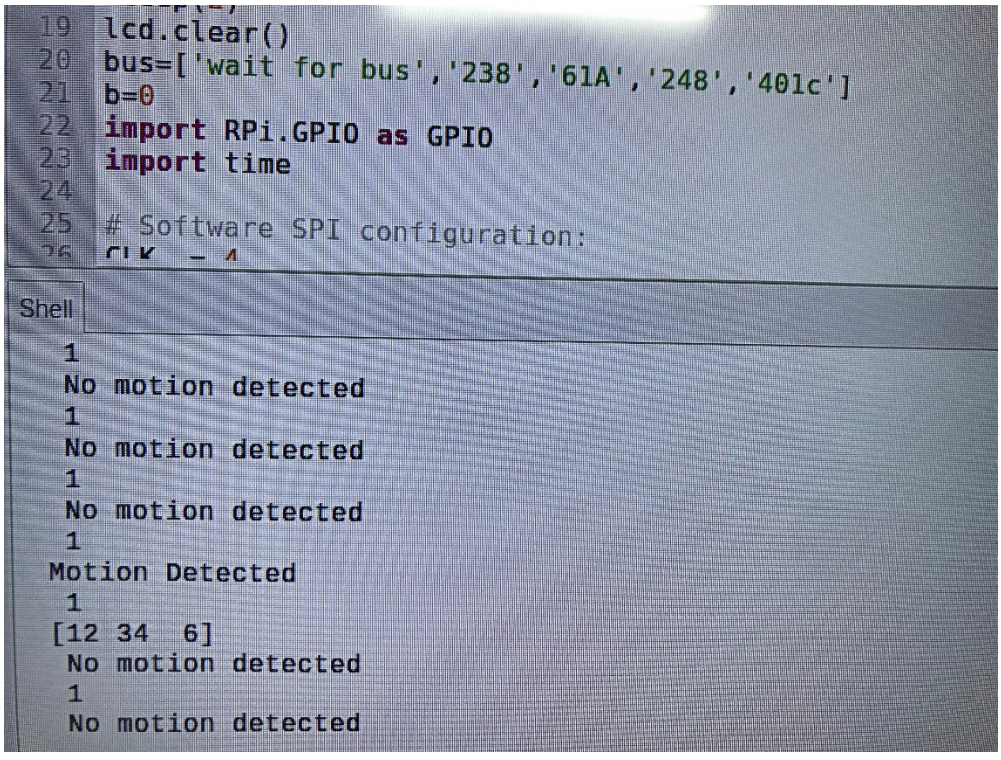


Fig 8.8 Output screen to show no motion during day

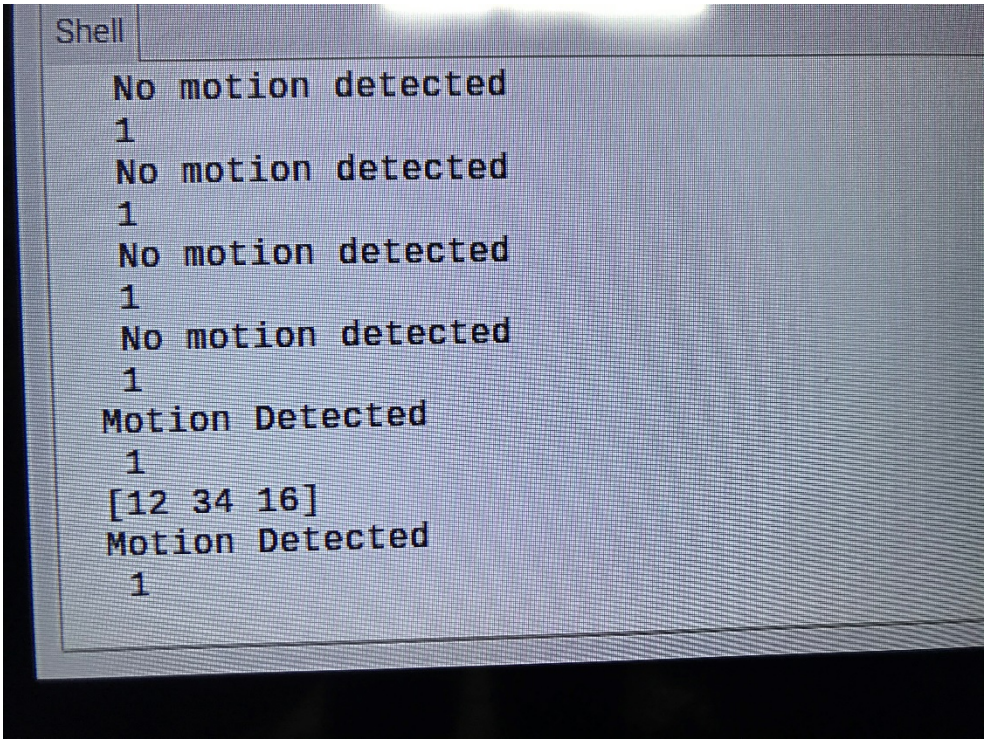


Fig 8.9 Output screen to show motion detected at day

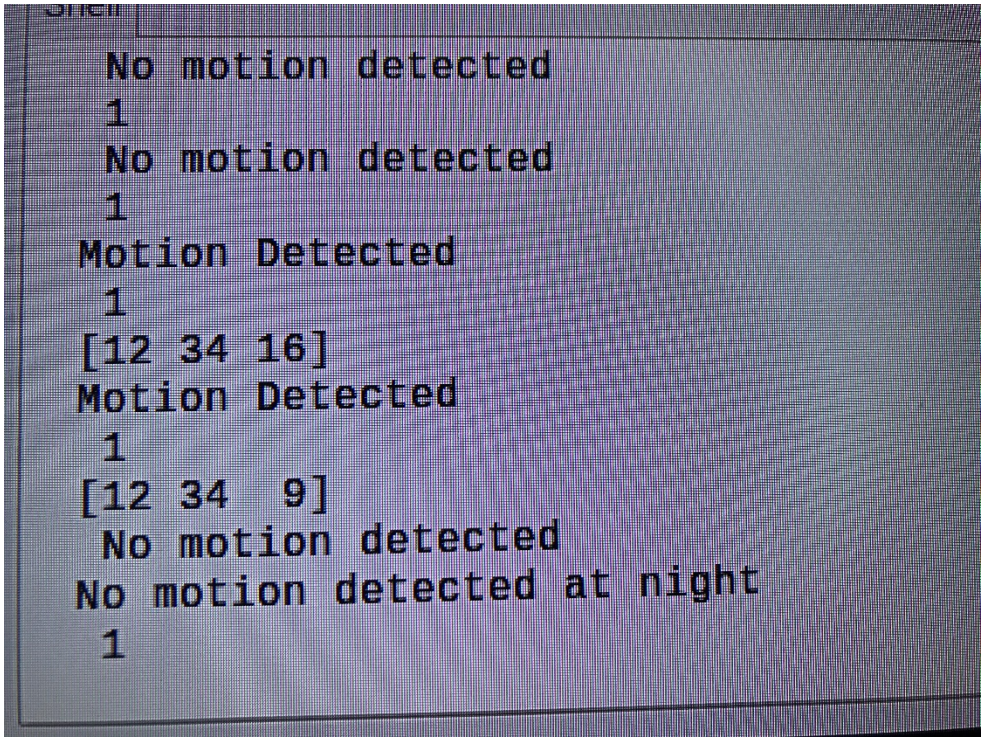


Fig 8.10 Output screen to show no motion detected at night

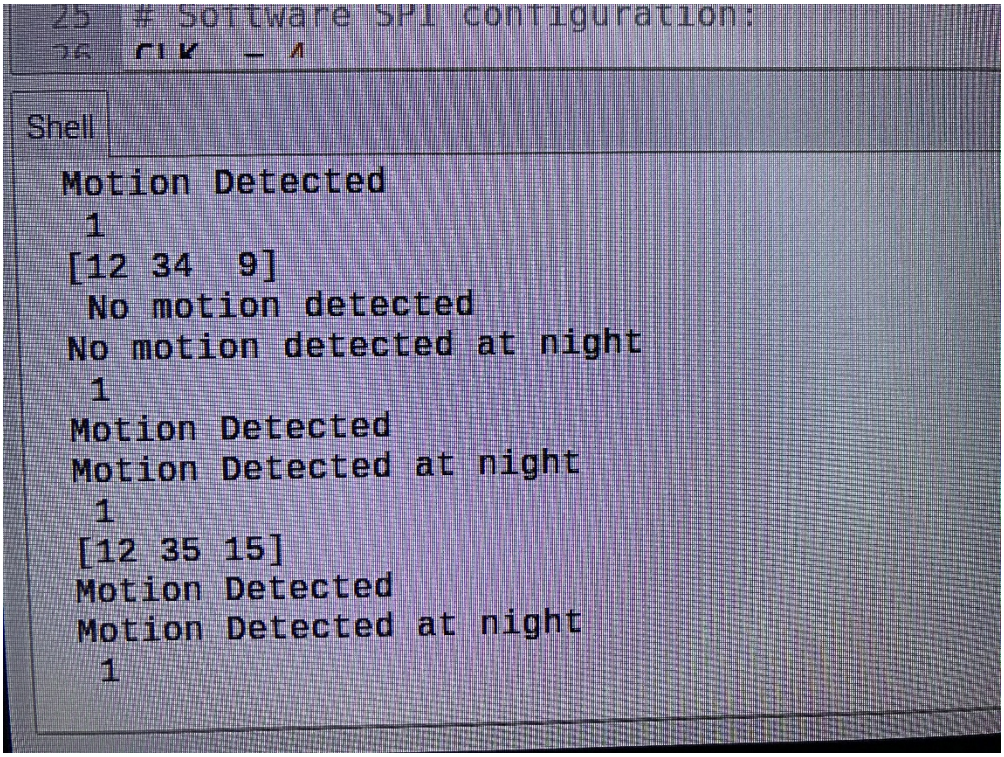


Fig 8.11 Output screen to show motion detected at night

Fig 8.12 App Dashboard

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 Conclusion

In the research, various useful ways in which IoT can be applicable in bus shelters are established, including particular value adding aspects to stakeholders. The stakeholders identified are, i.e. public transport provider, outdoor media advertisement companies, businesses and commuters. In most cases, the advertisement companies are the major stakeholders in the sense that they build and maintain bus shelters. Integration of IoT to connect bus shelters facilitates the possibilities of data collection through sensors, that is useful to various stakeholders such as local businesses, advertising companies and public transport providers. Currently, the main source of revenue generated in bus shelters comes from advertisements. Thus, to investigate the value adding aspects of smart bus shelters, we focused on the use case on smart advertising. The concept of dynamic and interactive advertising in bus shelters is being adopted widely since it has a positive influence on the target market. With the prototype we sought to demonstrate one way in which dynamic advertising can be achieved through connecting to social media, as well as location and time-based advertising. Vast amounts of sensor data collected over time introduce the possibility machine learning for the smart bus shelter, whereby the system *learns* the movement of people at different times of day resulting to better response and performance. This application would be useful in optimizing the bus routes, as well as the information displayed at certain time of the day, week or month. Moreover, integrating the buses with GPS modules also helps a lot by aiding us with techniques to reduce crowd density. More work can be done to implement live tracking of buses and traffic analysis and management. In addition, commuters can also know where exactly the bus is and how long will it take to reach the shelter.

8.2 Future Enhancements

Deployment of the smart bus shelter in a real world scenario should be done and tested to evaluate the user interaction as well as the effective advertising from local businesses. Further research should be employed to investigate the applications of machine learning in the smart bus shelters. This implies to the system's capability to learn different patterns from an array of factors, and help identify or predict the number, type and interests of people in a given bus shelter a given time. Further development presents an opportunity for data analytics that can be used to optimize to bus allocation in different routes.

i. Safety features can be added like CCTV
ii. Pollution sensors can be also added
iii. Emergency services can be an additional service to alert the nearest police stations and hospitals
iv. Touch Screen User Intractable Display system can be implemented too

REFERENCE

[1] Kevin Jingyi Zhang “Bus Stop Urban Design”. (2017)

[2] Kajeet “The Connected Bus and Technologies for Next Generation Bus”-. (2018)

[3] Ning Lu “Connected Vehicles: Solutions and Challenges”- IEEE Internet of Things Journal. (2016)

[4] Kelvin Wachira “Enhancing Public Information System in Bus Shelters by Integrating Smart IoT Solutions”- Master Thesis Project Report. (2016)

[5] Palle Joel Karthik “Smart Bus Shelter “-Master Thesis Project Report (2016)

[6] Johannes Pillmann, Benjamin Silwa,” Car to Cloud Communication Traffic Analysis based on Common Vehicle Information Model “-IEEE 85th Vehicular Technology Conference (VTC Spring) (2017)

[7] Dr. V. Raghavendra Reddy “Integrated Arduino Architectural Concepts”-International Journal of Automation, Control and Communication. Vol-1 No.-2 Page No.3 (2018)

[8] Shaik Tashlim “A Survey on GPS enabled City Bus Tracking System using Raspberry Pi.”-International Journal of Automation, Control and Communication. Vol-2 No.-3 Page No.2. (2017)

[9] Jason Brownlee “Linear Regression for Machine Learning”-MATIC Web of Conference 176. (2018)