VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum - 590018

A project report on

“Smart Bus Tracking System”

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

INFORMATION SCIENCE & ENGINEERING

By

1CR16IS018 Aparna AjayKumar
1CR16IS024 Bhanupriya P.C
1CR16IS093 Sanuallah Shivanagi
1CR16IS114 Sweta Singh

Under the Guidance of

Mrs.Savitha Hiremath

Assistant Professor

Department of ISE,Bengaluru

CMR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

#132, AECS Layout, IT Park Road, Bengaluru -

560037 2018-19

CMR INSTITUTE OF TECHNOLOGY
BANGALORE-37

Department of Information Science & Engineering

Certificate

This is to certify that the project entitled **"Smart Bus Tracking System"** has been successfully completed by **Ms. Aparna AjayKumar, USN 1CR16IS018, Ms. Bhanupriya P.C, USN 1CR16IS024 ,Mr. Sanaullah Shivanagi, USN 1CR16IS093 and Sweta Singh, USN 1CR16IS114,** bonafide students of CMR Institute of Technology in partial fulfillment of the requirements for the award of degree of Bachelor of Engineering in **Information Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the academic year 2019-2020. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the project report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Name & Signature of Guide	Name & Signature of HOD	Signature of Principal
(Mrs. Savitha Hiremath)	(Mrs.Farida Begam)	(Dr.Sanjay Jain)

External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

CMR INSTITUTE OF TECHNOLOGY
BANGALORE-37

Department of Information Science & Engineering

DECLARATION

We, **Aparna AjayKumar, USN:1CR16IS018, Bhanupriya P.C, USN:1CR16IS024 , Sanaullah Shivanagi , USN:1CR16IS093** and **Sweta Singh, USN:1CR16IS114,** bonafide students of CMR Institute of Technology, Bangalore, hereby declare that the dissertation entitled, **"Smart Bus Tracking System"** has been carried out by us under the guidance of **Mrs. Savitha Hiremath, Asst. Professor**, CMRIT, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Information Science Engineering**, of the Visvesvaraya Technological University, Belgaum during the academic year 2019-2020. The work done in this dissertation report is original and it has not been submitted for any other degree in any university.

APARNA AJAYKUMAR (1CR16IS018)

BHANUPRIYA P.C (1CR16IS024)

SANAULLAH SHIVANAGI (1CR16IS093)

SWETA SINGH (1CR16IS114)

ABSTRACT

Student safety is a primary concern in our society. Increased rates of child abduction signify the relevance of a proper mechanism to track children. The current system involves parents calling the cab driver to ensure student has boarded the bus and to know the current location. There is always an element of uncertainty regarding student whereabouts. Proposed system involves a low cost solution by allowing parents to track child location via a mobile application.The system involves allotment of a unique identification for each student using RFID (Radio Frequency based Identification system). Live tracking of the bus is enabled using GPS (Global Positioning System). Wi-Fi module is used to transfer the details over remote server from which this will be made available to parent portal. This work involves identification of students present in the bus along with tracking of bus. The mobile application allows parent's login and displays the current status of the student as well as a link to google map showing the current location of the bus.

ACKNOWLEDGEMENT

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

So, it is with gratitude that we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We would like to thank **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for providing an excellent academic environment in the college and his never-ending support for the B.E program.

We would like to express our gratitude towards **Mrs Farida Begam,** Assoc Professor and HOD, Department of Information Science and Engineering CMRIT, Bangalore, who provided guidance and gave valuable suggestions regarding the project.

We consider it a privilege and honour to express our sincere gratitude to our internal guide **Mrs. Savitha Hiremath,** Asst. Professor, Department of Information Science & Engineering, CMRIT, Bangalore for their valuable guidance throughout the tenure of this project work.

We would also like to thank all the faculty members who have always been very Co-operative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during our project.

APARNA AJAYKUMAR

BHANUPRIYA P.C

SANAULLAH SHIVANAGI

SWETA SINGH

TABLE OF CONTENTS

Title **Page No.**

LIST OF FIGURES

LIST OF FIGURES

Chapter 1

PREAMBLE

1.1 Introduction

The general objective of the development of this system is to help people track things in more efficient way and effective way resulting in greater reliability. In real life, when a freight service provider wants to track his/ her shipment service which is been carried by a vehicle. It's really difficult to man- age all those fleet of vehicles which is in movement in whereas corners of the city.

There are buses made available for passengers travelling distances, but not many passengers have complete information about these buses. Complete information namely the number of buses that go to the required destination, bus numbers, bus timings, the routes through which the bus would pass, time taken for the bus to reach, maps that would guide the passenger with his/her route and most importantly, track the current location of the bus and give the correct time for the bus to reach its bus stop. The proposed system deals with overcoming the problems stated above. The system is an Android application that gives necessary information about all the buses travelling in Pune. This information overcomes the problems faced in the previously built application "Pune Bus Guide". The platform chosen for this kind of system is android, reason being Android Operating System has come up on a very large scale and is owned by almost every second person. Also, Android is a user friendly platform, thereby enabling ease of access for all the users. A number of applications made for the Android Operating System is increasing on a large scale ever since its advent. Android is an open source mobile software environment.

Brought up by Google, the operating system has been made Linux based and uses Java programming language. It has a virtual machine that is used to optimize memory usage as well as resources. This application has been developed using IDE (Android Studio 1.6) with ADT (Android Development Tools) and Android SDK (Software Development Kit). There are a number of constraints that need to be satisfied.

Vehicle Tracking System (VTS) is the technology used to determine the location of a vehicle using different methods like GPS and other radio navigation systems operating through satellites and ground based stations. This system is an important tool for tracking each vehicle at a given period of time and now it is becoming increasingly popular for people having expensive cars and hence as a theft prevention and retrieval device.

Vehicle Security using VTS:-

Vehicle Security is a primary concern for all vehicle owners. Owners as well as researchers are always on the lookout for new and improved security systems for their vehicles. One has to be thankful for the upcoming technologies, like GPS systems, which enables the owner to closely monitor and track his vehicle in real-time and also check the history of vehicles movements. This new technology, popularly called Vehicle Tracking Systems has done wonders in maintaining the security of the vehicle tracking system is one of the biggest technological advancements to track the activities of the vehicle. The security system uses Global Positioning System GPS, to find the location of the monitored or tracked vehicle and then uses satellite or radio systems to send to send the coordinates and the location data to the monitoring center. Due to real-time tracking facility, vehicle tracking systems are becoming increasingly popular among owners of expensive vehicles.

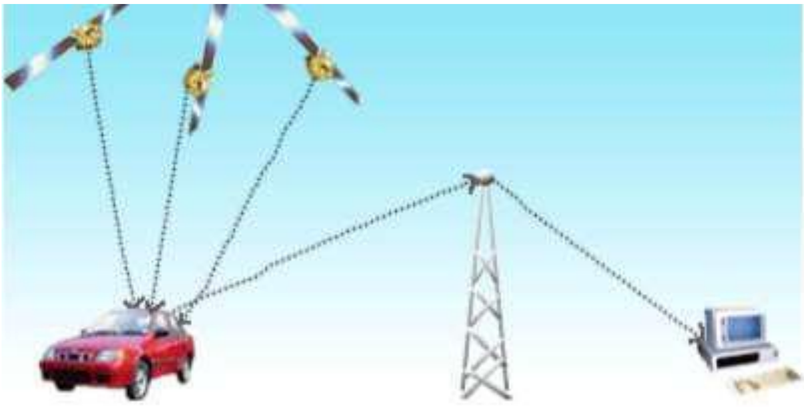


Figure 1.1 Vehicle tracking system

1.2 EXISTING SYSTEM

The existing system has many disadvantages related to cost, performance etc. The proposed system is going to reduce all the defects of existing system. The proposed system uses the Arduino Uno kit which provides power of microcontroller as well as microprocessor which is going to increase the performance of the system with less energy consumption.

1.3 PROPOSED SYSTEM

The system has two sub systems namely the school bus sub system and remote server sub system. The school bus subsystem is associated with student identification using RFID and location tracking using GPS. Each entry and exit in bus involves activation of RFID reader and acquisition of student ID. This is used to determine student status which is transmitted to server via Wi-Fi module. The location is transmitted at regular intervals to server to track the bus.

The server subsystem involves remote database that stores the student status as well as GPS co-ordinates along with student details. This is used to display relevant information to parents after login. Server subsystem is used to update the data and relay the same to application. Each student has a passive RFID tag which stores unique data for identification. When the tag is in vicinity of reader, internal inductive current produced by the tag in response to the wireless signal transmitted by reader allows the tag to provide the data to the reader. It can work without manual intervention at all making the process automatic.

1.4 PROBLEM STATEMENT

The objective of this proposed system is to develop an application which will help to provide security for school going children's. This allows parents and management to check the status of secure smart school bus by using Iota. The proposed system will provide various facilities like speed control, check drunk and drive, missing children's, accident emergencies, inappropriate drop, panic button, logistic management etc. which are helpful for child security.

CHAPTER 2

LITERATURE SURVEY

Implementation of Children Tracking System on Android Mobile Terminals

Authors: J.Saranya , J.Selvakumar

Publications: International conference on Communication and Signal Processing

Abstract:

recently, all over the world, crime against children is increasing at higher rates and it is high time to offer safety support system for the children going to schools. This paper focuses on implementing children tracking system for every child attending school. However the existing systems are not powerful enough to prevent the crime against children since these systems give information about the children group and not about each child resulting in low assurance about their child safety to parents and also does not concentrate on sensing the cry of the child and intimating the same to its parents. The proposed system includes a child module and two receiver modules for getting the information about the missed child on periodical basis. The child module includes ARM7 microcontroller (lpc 2378), Global positioning system (GPS), Global system for mobile communication (GSM), Voice playback circuit and the receiver module includes Android mobile device in parent's hand and the other as monitoring database in control room of the school. Finally, implementation results for the proposed system are provided in this paper.

A Self-Configurable New Generation Children Tracking System Based on Mobile Ad Hoc Networks Consisting of Android Mobile Terminals

Authors: Yuichiro MORI, Hideharu KOJIMA, Eitaro KOHNO,

Publications: IEEE

Abstract:

Hiroshima City Children Tracking System is a safety support system for children based on ad hoc network technologies. Field experiments have been conducted in cooperation with an elementary school in Hiroshima. In this paper, we propose a new generation children tracking system which is based on experiences and findings of the field experiments for Hiroshima City Children Tracking System. Our proposed system consists of Android terminals which has

Wireless LAN device and Bluetooth device with the ad hoc communication function. Our system manages groups of Android terminals using Autonomous Clustering technique. In this paper, we show the system requirements for our children tracking system and describe the implementation features to satisfy the system requirements. Finally, we provide some preliminary implemented results for our proposed system.

SMS Based Kids Tracking and Safety System by Using RFID and GSM

Authors: Nitin Shyam

Publications: International Journal of Innovative Science, Engineering & Technology

Abstract:

In present time due to increase in number of kidnapping and road accident cases, Parents always worry about their children. This paper proposes a SMS based solution to aid parents to track their children location in real time. The proposed system takes the advantage of the location services provided by module kit which carry by the Childs in their school bag. It allows the parent to get their child's location on a real time map by the geographical coordinates which send by the module kit. Information such as GPS coordinates and time are gathered and sent to the parent's phone that's preregistered on the module kit. The communication between the parent and the child module kit is done using Short Message Service (SMS). SMS offers the system unique features. It will allow the system to work without the need of internet connection. The system sends the location of child's smart phone to parent's smart phone when the parent wishes to check on the child.

Smart Tracking System for School Buses Using Passive RFID Technology to Enhance Child Safety

Authors: Khaled Shaaban

Publications: Journal of Traffic and Logistics Engineering

Abstract:

Millions of children need to be moved from home to school and vice versa every day. For parents, obtaining a safe transport for their children is a critical issue. Many children find themselves locked in a school bus in the bus parking lot after falling asleep on their way to school, miss the bus, step into the wrong bus, or leave at the wrong station with no method to track them. This research tested the applicability of radio frequency identification (RFID) technology in tracking and monitoring children during their trip to and from school on school busses. The child safety system developed in this research utilized the passive RFID tracking technology due to its efficient tracking capabilities, low cost, and easy maintenance. To explore the technical feasibility of the proposed system, a set of tests were performed in the lab and with the public. These experiments showed that the RFID tags were effective and stable enough to be used for successfully tracking and monitoring children using the bus. When asked to give their feedback of the solution through a questionnaire, more than 95% of the parents see that such a solution will take their anxiety and worry away and will provide them a tool to track their kids during commuting to and from their schools.

The Evolution of RFID Security and Privacy:A Research Survey

Authors: R.K. Pateriya

Publications: International Conference on Communication Systems and Network Technologies

Abstract:

This paper presents the recent technical research on the problems of privacy and security for radio frequency identification (RFID). RFID technology is already used widely and is increasingly becoming a part of daily life. However, issues regarding security and privacy with respect to RFID technology have not been resolved satisfactorily. There are huge number of challenges, which must be overcome to resolve RFID security and privacy issues. It is because of the many constraints attached to the provision of security and privacy in RFID systems. These challenges are chiefly technical and economic in nature but also include ethical and social issues. Along with meeting the security and privacy needs of RFID technology, solutions must be inexpensive, practical, reliable, scalable, flexible, inter-organizational, and long lasting. This paper reviews the approaches which had been proposed by scientists for privacy protection and integrity assurance in RFID systems, and treats the social and technical context of their work. This paper can be useful as a reference for non specialist, as well as for specialist readers.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 Overview on IOT

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

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

3.2 Key Features

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

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

passive engagement. IoT introduces a new paradigm for active content, product, or service engagement.

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

3.3 Internet of Things - Technology and Protocols

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

3.3.1 NFC and RFID

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

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

3.3.2 Low-Energy Bluetooth

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

3.3.3 Low-Energy Wireless

This technology replaces the most power hungry aspect of an IoT system. Though sensors and other elements can power down over long periods, communication links (i.e., wireless) must remain in listening mode. Low-energy wireless not only reduces consumption, but also extends the life of the device through less use.

3.3.4 Radio Protocols

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

3.3.5 LTE-A

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

3.4 Internet of Things - Hardware

The hardware utilized in IoT systems includes devices for a remote dashboard, devices for control, servers, a routing or bridge device, and sensors. These devices manage key tasks and functions such as system activation, action specifications, security, communication, and detection to support-specific goals and actions.

3.4.1 IoT − Sensors

The most important hardware in IoT might be its sensors. These devices consist of energy modules, power management modules, RF modules, and sensing modules. RF modules manage communications through their signal processing, WiFi, ZigBee, Bluetooth, radio transceiver, duplexer, and BAW.

3.4.2 Wearable Electronics

Wearable electronic devices are small devices worn on the head, neck, arms, torso, and feet.

Current smart wearable devices include −

- **Head** − Helmets, glasses
- **Neck** − Jewelry, collars
- **Arm** − Watches, wristbands, rings

- **Torso** – Clothing, backpacks
- **Feet** – Socks, shoes

3.4.3 Standard Devices

The desktop, tablet, and cellphone remain integral parts of IoT as the command center and remotes.

- The **desktop** provides the user with the highest level of control over the system and its settings.
- The **tablet** provides access to the key features of the system in a way resembling the desktop, and also acts as a remote.
- The **cellphone** allows some essential settings modification and also provides remote functionality.

Other key connected devices include standard network devices like **routers** and **switches**.

3.5 Internet of Things - Software

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

3.5.1 Data Collection

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

3.5.2 Device Integration

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

3.5.3 Real-Time Analytics

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

3.5.4 Application and Process Extension

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

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

A System Requirement Specification (SRS) is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives the brief description of the services that the system should provide and also the constraints under which, the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it.

SRS document itself states in precise and explicit language those functions and capabilities a software system must provide, as well as states any required constraints by which the system must abide. SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

4.1Hardware

- Arduino UNO
- Power Supply
- RFID
- DC Motor
- Buzzer
- Gas Sensor
- NodeMcu

4.1.1 Arduino:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

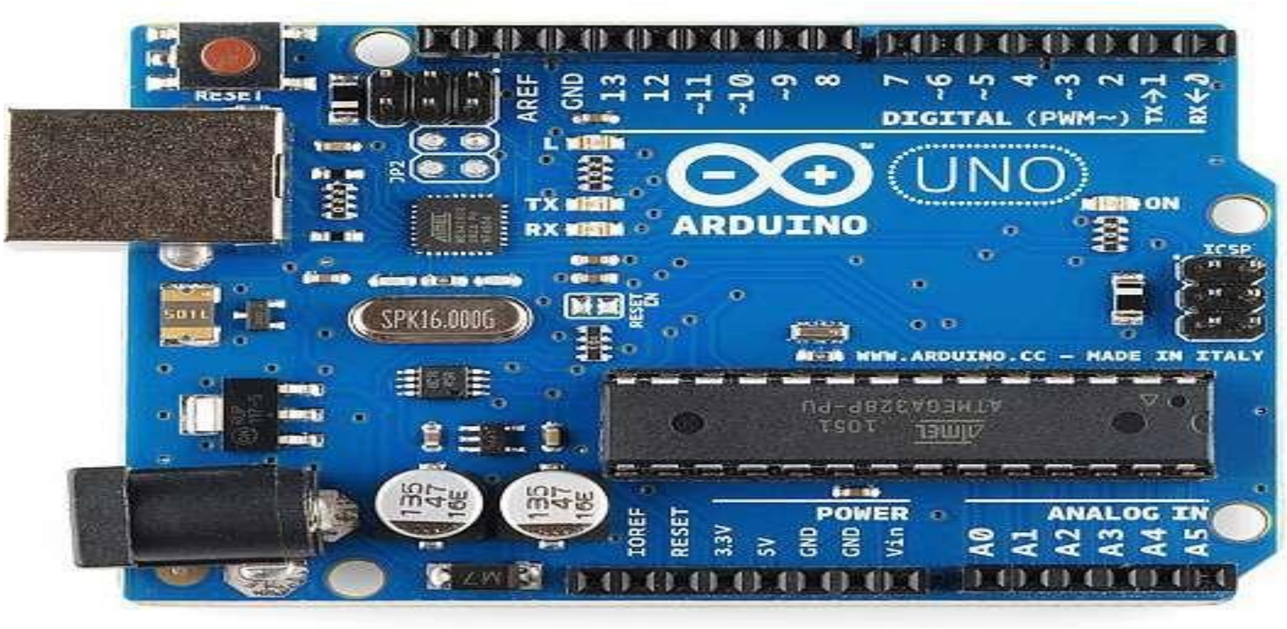


Fig:4.1 Arduino

Arduino specification:

Microcontroller	ATmega328P
Operating Voltage	5v
Input voltage	7-12v
Input voltage limit	6-20v
Digital I/O Pins	6
Analogue input Pins	6
DC current perI/O pins	20 mA
DC current for 3.3v Pin	50 mA
Flash Memory	Of which o.5KB is used
SRAM	2 KB
EEPROM	1KB
Clock Speed	16MHz
Length	68.6mm

Width	53.4nm
Weight	25g

Table 4.1

Arduino programming:

The Arduino/Genuino Uno can be programmed with the (Arduino Software (IDE)).Select "Arduino/Genuino Uno from the Tools > Board menu (according to the microcontroller on your board). The ATmega328 on the Arduino/Genuino Uno comes preprogrammed with a boot loader that allows us to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

We can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then rese ing the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. [1]

Warnings:

The Arduino/Genuino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Differences with other boards:

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Power:

The Arduino/Genuino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and VIN pin headers of the POWER connector.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN. The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). One can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

Memory:

The ATmega328 has 32 KB (with 0.5 KB occupied by the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library). [1]

Input & Output:

Each of the 14 digital pins on the Uno can be used as an input or output, using pin mode (), digital write (), and digital read () functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull- up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach interrupt () function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog write () function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog reference () function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analog Reference ().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields

which block the one on the board.

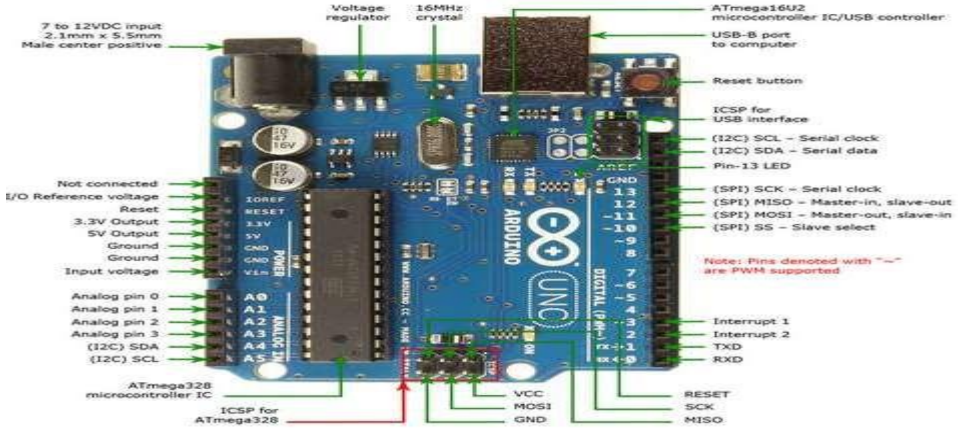


Fig 4.2: Pin Specification

Communication:

Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an

.inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software serial library allows serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communicatio n, use the SPI library.

Automatic (Software) Reset:

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.[1]

4.1.2 Regulated power supply:

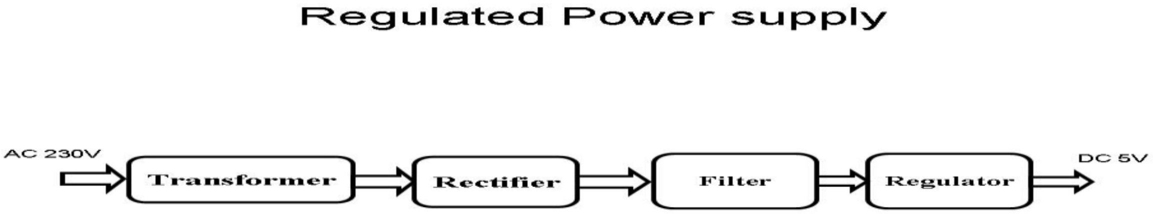


Fig 4.3

Transformer:

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors without changing its frequency. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction. If a load is connected to the secondary, an electric current will flow in the secondary winding and electrical energy will be transferred from the primary circuit through the transformer to the load. This field is made up from lines of force and has the same shape as a bar magnet. If the current is increased, the lines of force move outwards from the coil. If the current is reduced, the lines of force move inwards. If another coil is placed adjacent to the first coil then, as the field moves out or in, the moving lines of force will "cut" the turns of the second coil. As it does this, a voltage is induced in the second coil. With the 50 Hz AC mains supply, this will happen 50 times a second. This is called MUTUAL INDUCTION and forms the basis of the transformer.

Rectifier:

A rectifier is an electrical device that converts alternating current (AC) to direct current (DC), a process known as rectification. Rectifiers have many uses including as components of power supplies and as detectors of radio signals. Rectifiers may be made of solid-state diodes, vacuum tube diodes, mercury arc valves, and other components. A device that it can perform the opposite function (converting DC to AC) is known as an inverter. When only one diode is used to rectify AC (by blocking the negative or positive portion of the waveform), the difference between the term diode and the term rectifier is merely one of usage, i.e., the term rectifier describes a diode that is being used to convert AC to DC. Almost all rectifiers comprise a number of diodes in a specific arrangement for more efficiently converting AC to DC than is possible with only one diode. Before the development of silicon semiconductor rectifiers, vacuum tube diodes and copper (I) oxide or selenium rectifier stacks were used.

Filter:

The process of converting a pulsating direct current to a pure direct current using filters is called as filtration. Electronic filters are electronic circuits, which perform signal-processing functions, specifically to remove unwanted frequency components from the signal, to enhance wanted ones.

Regulator:

A voltage regulator (also called a _regulator‘) with only three terminals appears to be a simple device, but it is in fact a very complex integrated circuit. It converts a varying input voltage into a constant _regulated ‘output voltage. Voltage Regulators are available in a variety of outputs like 5V, 6V, 9V, 12V and 15V. The LM78XX series of voltage regulators are designed for positive input. For applications requiring negative input, the LM79XX series is used. Using a pair of _voltage-divider‘ resistors can increase the output voltage of a regulator circuit. It is not possible to obtain a voltage lower than the stated rating. You cannot use a 12V regulator to make a 5V power supply. Voltage regulators are very robust. These can withstand over-current draw due to short circuits and also over-heating. In both cases, the regulator will cut off before any damage occurs. The only way to destroy a regulator is to apply reverse voltage to its input. Reverse polarity destroys the regulator almost instantly.

4.1.3 LCD display

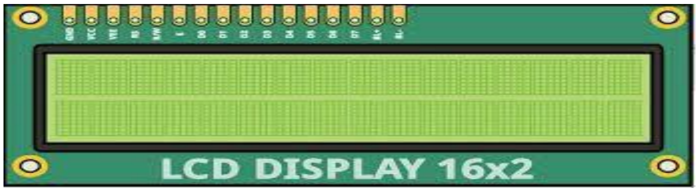


Fig 4.4 16*2 LCD display

A **liquid-crystal display** (**LCD**) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

LCD is used in wide range application including computer monitors, televisions, instrument panels, aircraft cockpit displays, and indoor and outdoor signage. Small LCD screens are common in portable consumer devices such as digital cameras, watches, calculators, and mobile telephones, including smartphones. LCD screens are also used on consumer electronics products such as DVD players, video game devices and clocks. LCD screens have replaced heavy, bulky cathode ray tube (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and plasma displays, with LCD screens available in sizes ranging from tiny digital watches to huge, big- screen television sets.

Since LCD screens do not use phosphors, they do not suffer image burn-in when a static image is displayed on a screen for a long time (e.g., the table frame for an aircraft schedule on an indoor sign). LCDs are, however, susceptible to image persistence.

Interfacing an LCD with an Arduino

The 16x2 LCD has a total of 16 pins. As shown in the table below, eight of the pins are data lines (pins 7-14), two are for power and ground (pins 1 and 16), three are used to control the operation of LCD (pins 4-6), and one is used to adjust the LCD screen brightness (pin 3). The remaining two pins (15 and 16) power the backlight.The details of the LCD terminals are as follows:

Terminal 1	GND
Terminal 2	+5V
Terminal 3	Mid terminal of potentiometer (for brightness control)
Terminal 4	Register Select (RS)
Terminal 5	Read/Write (RW)
Terminal 6	Enable (EN)
Terminal 7	DB0
Terminal 8	DB1
Terminal 9	DB2
Terminal 10	DB3
Terminal 11	DB4
Terminal 12	DB5
Terminal 13	DB6
Terminal 14	DB7
Terminal 15	+4.2-5V
Terminal 16	GND

Table 4.2 . LCD terminals

LCD Module

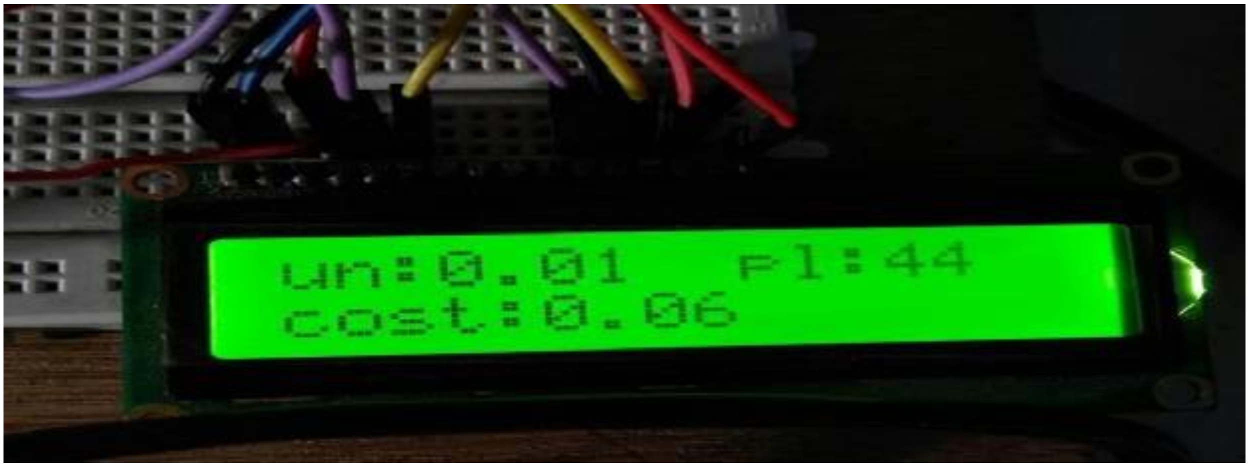


Fig4.5 Output of LCD

The name and functions of each pin of the 16×2 LCD module is given below. Pin1 (Vss): Ground pin of the LCD module.

Pin2 (Vcc): Power to LCD module (+5V supply is given to this pin)

Pin3 (VEE): Contrast adjustment pin. This is done by connecting the ends of a 10K potentimeter to +5V and ground and then connecting the slider pin to the VEE pin. The voltage at the VEE pin defines the contrast. The normal setting is between 0.4 and 0.9V.

Pin4(RS): Register select pin. Logic HIGH at RS pin selects data register and logic LOW at RS pin selects command register. If we make the RS pin HIGH and feed an input to the data lines (DB0 to DB7), this input will be treated as data to display on LCD screen. If we make the RS pin LOW and feed an input to the data lines, then this will be treated as a command (a command to be written to LCD controller – like positioning cursor or clear screen or scroll).

Pin5 (R/W): Read/Write modes. This pin is used for selecting between read and write modes. Logic HIGH at this pin activates read mode and logic LOW at this pin activates write mode.

Pin6 (E): This pin is meant for enabling the LCD module. A HIGH to LOW signal at this pin will enable the module.

Pin7 (DB0) to Pin14(DB7): These are data pins. The commands and data are fed to the LCD module though these pins.

Pin15 (LED+): Anode of the back light LED. When operated on 5V, a 560 ohm resistor should be connected in series to this pin. In Arduino based projects the back light LED can be powered from the 3.3V source on the Arduino board.

Pin16 (LED-): Cathode of the back light LED.

RS pin of the LCD module is connected to digital pin 12 of the Arduino. R/W pin of the LCD is grounded. Enable pin of the LCD module is connected to digital pin 11 of the Arduino. This method is very simple, requires less connections and we can almost utilize the full potential of the LCD module. Digital lines DB4, DB5, DB6 and DB7 are interfaced to digital pins 5, 4, 3 and 2 of the Arduino. The 10K potentiometer is used for adjusting the contrast of the display. The Arduino can be powered through the external power jack provided on the board. +5V required in some other parts of the circuit can be tapped from the 5V source on the Arduino board. The Arduino can be also powered from the PC through the USB port.

4.1.4 Jumper Wires

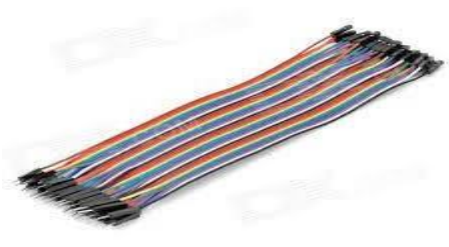


Fig 4.6. Jumper wires

A **jump wire** (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable) is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

4.1.5RFID

RFID History

In 1946, a Russian invented an espionage tool called the Covert Listening Device. This device retransmitted incident radio waves with audio information. Sound waves vibrated a diaphragm which slightly altered the shape of the resonator, which modulated the reflected radio frequency. This passive device was attributed to be the first known device and a predecessor of the RFID technology. The British invented a similar system during the World War II to identify enemy aircraft. It was called the Identification of Friend or Foe (IFF). Initial application was during World War II-The United Kingdom used RFID devices to distinguish returning English airplanes from inbound German ones. RADAR was only able to signal the presence of a plane, not the kind of plane it was. It was invented in 1948 by Harry Stockman. In 1971, an RFID device that was passive, powered by the interrogating signal, with a 16-bit memory transponder was invented. This device was the true ancestor to modern RFID and was patented in 1973 in the USA that had demonstrated its uses in:

- Transportation (automotive vehicle identification, automatic toll system, electronic license plate, electronic manifest, vehicle routing, vehicle performance monitoring)
- Banking (electronic check book, electronic credit card), security (personnel identification, automatic gates, surveillance)
- Medical (identification, patient history)

It came into commercial use only in 1990s.

Radio frequency identification (RFID) technology is a wireless communication technology that enables users to uniquely identify tagged objects or people. RFID is rapidly becoming a cost-effective technology. This is in large part due to the efforts of Wal-Mart and the Department of Defense (DoD) to incorporate RFID technology into their supply chains. Although the foundation of the Radio Frequency Identification (RFID) technology was laid by past generations, only recent advances opened an expanding application range to its practical implementation.

RFID is only one of numerous technologies grouped under the term Automatic Identification (Auto ID), such as bar code, magnetic inks, optical character recognition, voice recognition, touch memory, smart cards, biometrics etc. Auto ID technologies are a new way of controlling information and material flow, especially suitable for large production networks.

RFID Concept

The RFID technology is a means of gathering data about a certain item without the need of touching or seeing the data carrier, through the use of inductive coupling or electromagnetic waves. The data carrier is a microchip attached to an antenna (together called transponder or tag), the latter enabling the chip to transmit information to a reader (or transceiver) within a given range, which can forward the information to a host computer. The middleware (software for reading and writing tags) and the tag can be enhanced by data encryption for security-critical application at an extra cost, and anti- collision algorithms may be implemented for the tags if several of them are to be read simultaneously.

One important feature enabling RFID for tracking objects is its capability to provide unique identification. One possible approach to item identification is the EPC (Electronic Product Code), providing a standardized number in the EPC global Network, with an Object Name Service (ONS) providing the adequate Internet addresses to access or update instance-specific data. However, currently, ONS cannot be used in a global environment, and since it is a proprietary service, its use is relatively expensive, especially for participants with limited resources such as SMEs. As an alternative, researchers from the Helsinki University have proposed the notation ID@URI, where ID stands for an identity code, and URI stands for a corresponding Internet address. This allows several partners to use the system and still guarantee unique identification. The project ‘Identity-Based Tracking and Web-Services for SMEs’ (http://www.traser- project.eu) is currently working on further development of this concept.

RFID tags or radio-frequency identification tags are helping streamline distribution, logistics and asset tracking and rapidly replacing traditional barcode technology as the solution of choice for company's in nearly every industry sector globally. With the increasing success and popularity of RFID more demands are being placed on its performance.

Additional capabilities are required for RFID tag design and functionality including the ability to package and encapsulate tags and incorporate sensor based technology. RFID tags are being used increasingly in extreme environments requiring exposure to harsh chemicals, high moisture and high heat.

The FOUR CORE Components of an RFID System

An RFID system has four basic components:

- A tag which is composed of a semiconductor chip and an antenna.
- An interrogator (sometimes called a read/write device), which is composed of an Antenna, a RF electronics module, and a control electronics module.
- A controller (sometimes called a host), which most often takes the form of a PC or a workstation running database and control (often called middleware) software.
- An antenna, which converts electrical power to RF power.

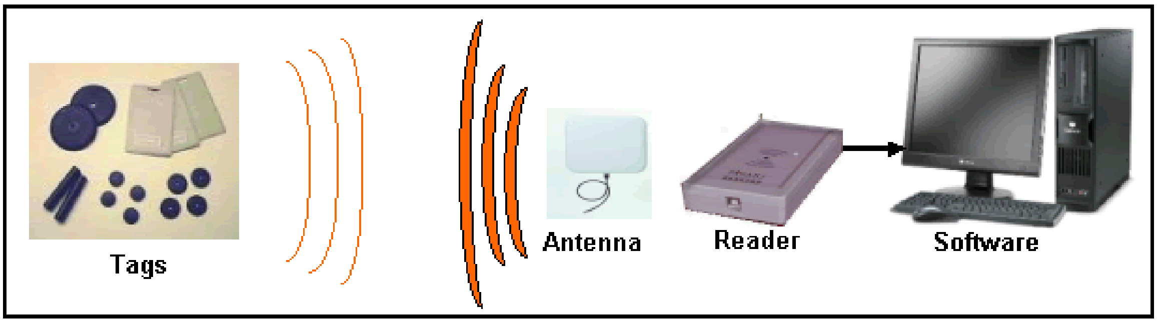


Fig4.7. Basic Building blocks of an RFID system

RFID TAGS

The basic function of an RFID tag is to store data and transmit data to the interrogator. At its most basic, a tag consists of an electronics chip and an antenna encapsulated in a package to form a usable tag, such as a packing label that might be attached to a box.

Generally, the chip contains memory where data may be stored and read from and sometimes written, too, in addition to other important circuitry. Some tags also contain batteries, and this is what differentiates active tags from passive tags. In our project we use passive tag.

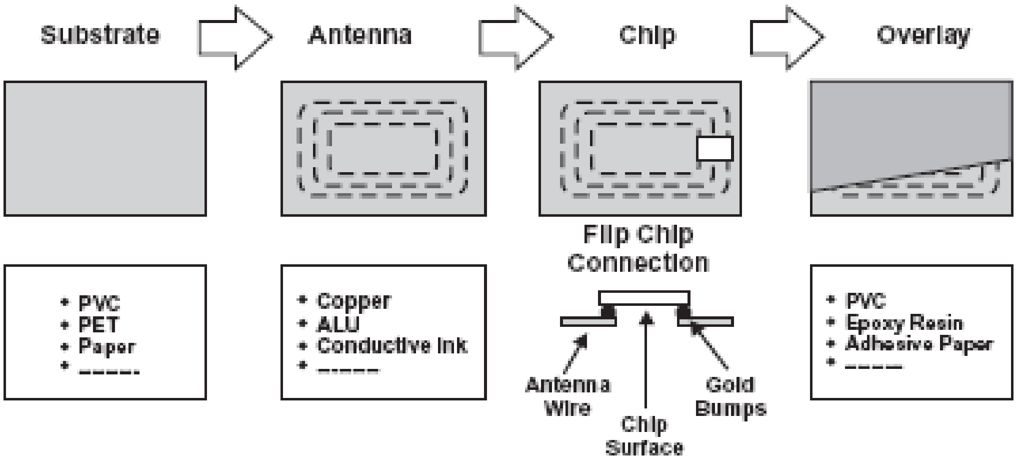


Fig 4.7: RFID Tag components

TYPES OF TAGS AND READERS

RFID tags and readers can be grouped under a number of categories. Their classification is presented.

Classification of RFID tags

Passive

- Also called 'pure passive', 'reflective' or 'beam powered'.
- Obtains operating power from the reader.

- The reader sends electromagnetic waves that induce current in the tag's antenna, the tag reflects the RF signal transmitted and adds information by modulating the reflected signal.
- Semi-passive uses a battery to maintain memory in the tag or power the electronics that enable the tag to modulate the reflected signal communicates in the same method, as the other passive tag.

Active

- Powered by an internal battery, used to run the microchip's circuitry and to broadcast a signal to the reader.
- Generally ensures a longer read range than passive tags.
- More expensive than passive tags (especial because usually are read/write)
- The batteries must be replaced periodically by the tag's memory type
- Read only - The memory is factory programmed, cannot be modified after its manufacture
- Its data is static.
- Very limited quantity of data can be stored, usually 96 bits of information can be easily integrated with data collection systems.
- Typically are cheaper than read-write tags.
- Read write - Can be read as well as written into.
- Its data can be dynamically altered.
- Can store a larger amount of data, typically ranging from 32 Kbytes to 128 Kbytes
- Being more expensive than read-only chips, is impractical for tracking inexpensive items by the method of wireless signal used for communication between the tag and reader Induction.
- Close proximity electromagnetic or inductive coupling - near field
- Generally use LF and HF frequency bands.

Tags can take a variety of forms like:

□ Smart labels

□ Keys and Key fobs

□ Watches

□ Smart Cards

□ Disks and Coins

□ Mount-on-metal

Fig 4.8: Various forms of Tag

RFID Interrogators (Reader)

An RFID interrogator acts as a bridge between the RFID tag and the controller and has a few basic functions to perform:

□ Read the data contents of an RFID tag

□ Write data to the tag (in the case of smart tags)

□ Relay data to and from the controller

□ Power-up the tag (in the case of passive tags).

RFID interrogators are composed of roughly three parts: an antenna, an RF electronics module, responsible for communicating with the RFID tag, and a controller electronics module, responsible for communicating with the controller. A number of factors can affect the distance at which a tag can be read (the read range). The frequency used for identification, the antenna gain, the orientation and polarization of the reader antenna and the transponder antenna, as well as the placement of the tag on the object to be identified will all have an impact on the RFID system's read range. The reader either continuously (in case of fixed readers) or on demand (as in handheld readers) sends out electromagnetic waves to inquire the presence of any tags in its active read field.

CLASSIFICATION OF READERS:

By design and technology used

- Read - only reads data from the tag. Usually a micro-controller-based unit with a wound output coil, Peak detector hardware, comparators, and firmware designed to transmit energy to a tag and read information back from it by detecting the backscatter modulation.

  Different types for different protocols, frequencies and standards exist.

- Read/write - reads and writes data from/on the tag by fixation of the device stationary. The device is attached in a fixed way, for example at the entrance gate, respectively at the exit gate of products mobile in this case the reader is a handy, movable device.

RFID Read/Module: DT125R Series

The DT125R series proximity OEM RFID Read modules work at the industry standard 125 kHz frequency.

- Designed to detect and read/write Hitag2 and TK5561 tags.
- Built-in antenna and pin out for external antenna.

Specifications:

Frequency	125 KHz
Reading distance	>= 50 mm
Interface	UART
Antenna	Built in / External
Supply Voltage	5V
Operating Temperature	-10°C to +50°C
Tag Types	Unique, TK 5530
Output Format	ASCII, Wiegand 26

Table 4.3: Specifications of RFID

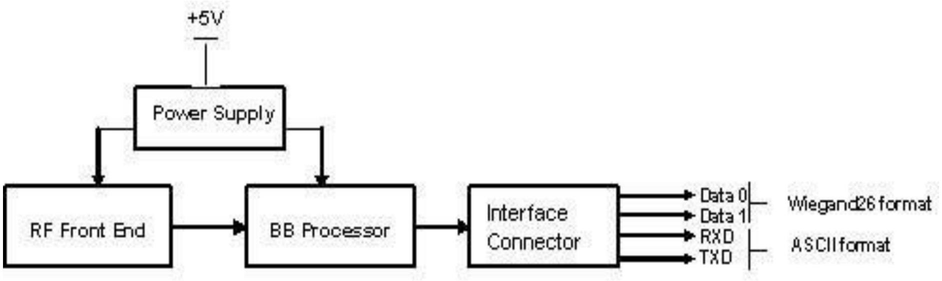


Fig 4.9: Block diagram of DT125R Series

The LF DT125R reader consists of a RF front end interfaced with the baseband processor that operates with +5V power supply. An antenna is interfaced with the RF front end, and tuned at 125 kHz to detect a tag (transponder) that comes in the vicinity of the reader field. The data read from the tag by the front end is detected and decoded by the baseband processor and is then sent to the UART interface.

Components of RFID Systems:

An RFID system is always made up of two components:

- The transponder, which is located on the object to be identified.
- The interrogator or reader, which, depending upon the design and the technology used, may be a read or write/read device.

A practical example is shown in Figure X. A reader typically contains a radio frequency module (transmitter and receiver), a control unit and a coupling element to the transponder. In addition, many readers are fitted with an additional interface (RS 232, RS 485, etc.) to enable them to forward the data received to another system (PC, robot control system, etc.). The transponder, which represents the actual data-carrying device of an RFID system, normally consists of a coupling element and an electronic microchip.

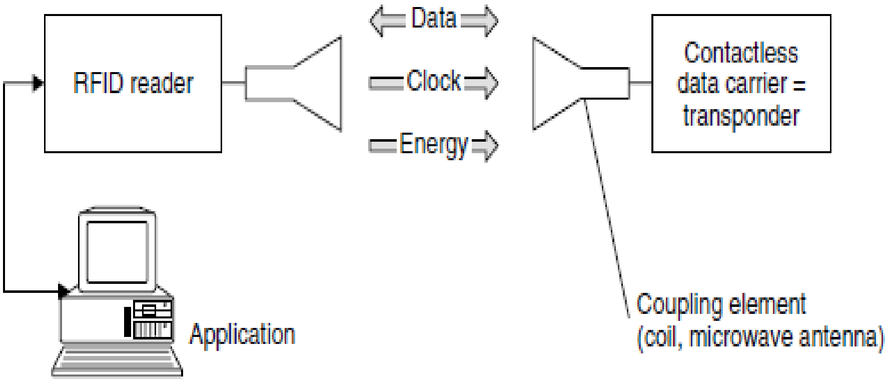

Data
Clock
Energy
RFID reader
Contactless data carrier = transponder
Application
Coupling element (coil, microwave antenna)

Fig 4.10: RFID System consisting of transponder and receiver

Fundamental Operating Principles of 1-Bit Transponder. A bit is the smallest unit of information that can be represented and has only two states: 1 and 0. This means that only two states can be represented by systems based upon a 1-bit transponder: ‘transponder in interrogation zone’ and ‘no transponder in interrogation zone’. Despite this limitation, 1-bit transponders are very widespread - their main field of application is in electronic anti-theft devices in shops (EAS, electronic article surveillance).

The radio frequency (RF) procedure is based upon LC resonant circuits adjusted to a defined resonant frequency. Early versions employed inductive resistors made of wound enameled copper wire with a soldered on capacitor in a plastic housing (hard tag). Modern systems employ coils etched between foils in the form of stick-on labels. To ensure that the damping resistance does not become too high and reduce the quality of the resonant circuit to an unacceptable level, the thickness of 1-bit transponder the aluminum conduction tracks on the 25μm thick polyethylene foil must be at least 50μm. Intermediate foils of 10μm thickness are used to manufacture the capacitor plates.

The reader (detector) generates a magnetic alternating field in the radio frequency Range. If the LC resonant circuit is moved into the vicinity of the magnetic alternating field, energy from the alternating field can be induced in the resonant circuit via its coils (Faraday’s law). If the frequency of the alternating field corresponds with the resonant frequency of the LC resonant circuit the resonant circuit produces a sympathetic oscillation. The current that flows in the resonant circuit as a result of this acts against its cause, i.e. it acts against the external magnetic alternating field. This effect is noticeable as a result of a small change in the voltage drop across the transmitter’s generator coil and ultimately leads to a weakening of the measurable magnetic field strength. A change to the induced voltage can also be detected in an optional sensor coil as soon as a resonant oscillating circuit is brought into the magnetic field of the generator coil. The relative magnitude of this dip is dependent upon the gap between the two coils (generator coil — security element, security element — sensor coil) and the quality *Q* of the induced resonant circuit (in the security element)

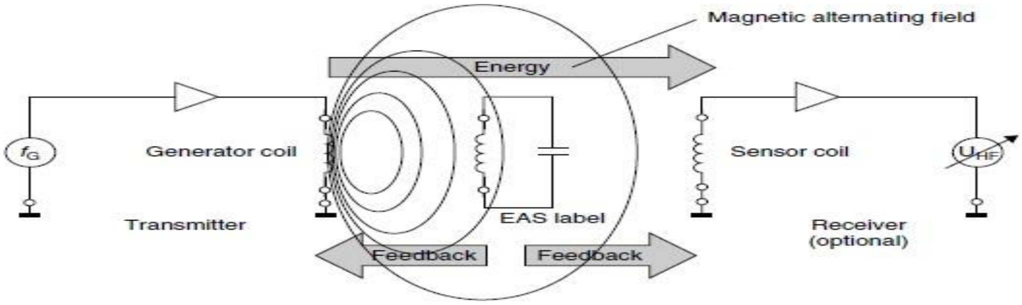


Fig 4.11: Operating principle of the EAS radio frequency procedure

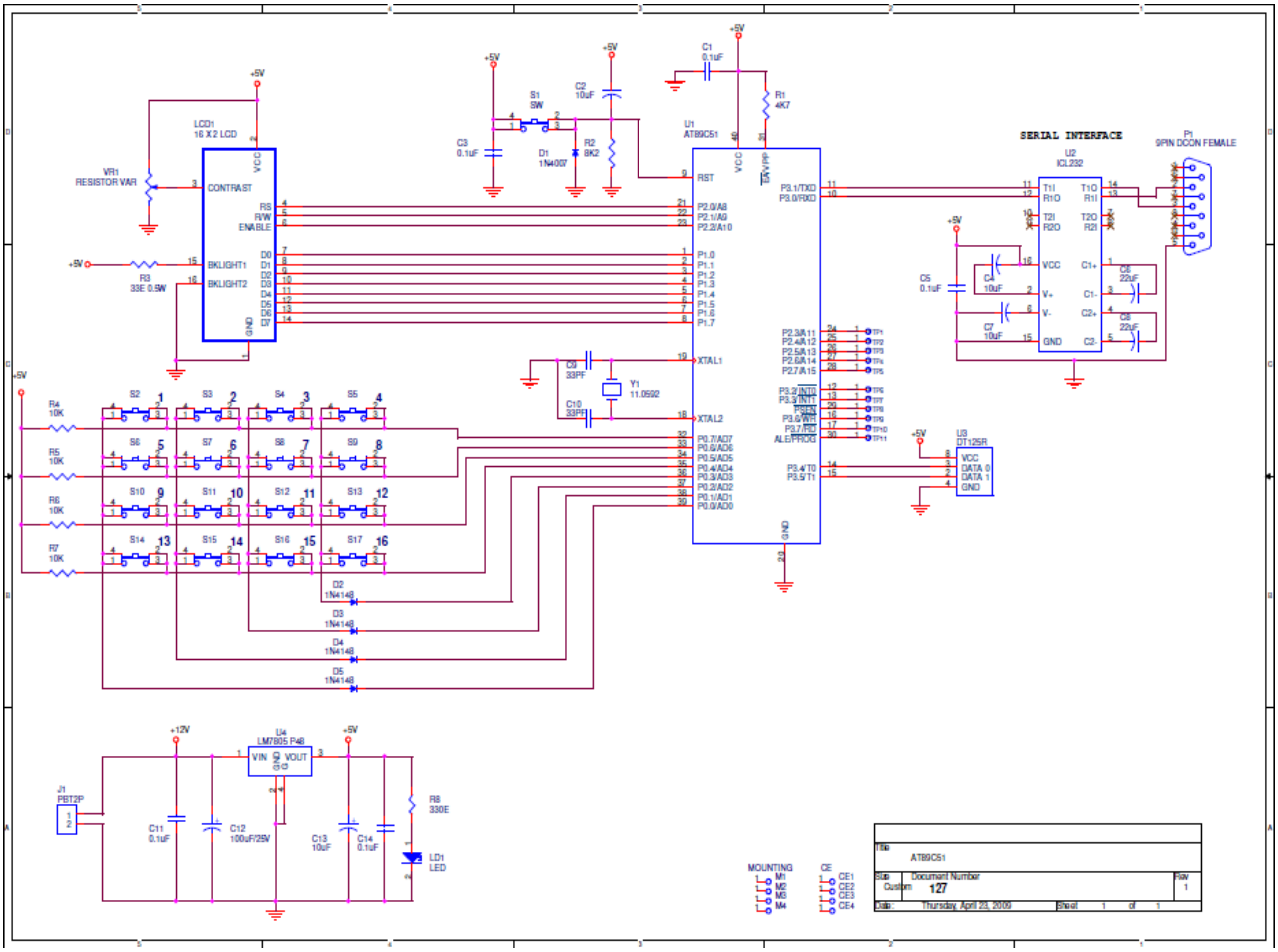


Figure 4.12: Schematic circuit diagram of RFID

4.1.6 H-BRIDGE

When the IR receiver does not receive a signal, the potential at the inverting input goes higher than that non-inverting input of the comparator IC (LM393). Thus the output of the comparator (LM 339) goes high and the LED starts glowing. Resistor R1 (100), R2 (10k) and R3 (330) are used to ensure that minimum 10 mA current passes through the IR LED Devices like Photodiode and normal LEDs respectively. Resistor VR2 (preset=5k) is used to adjust the output terminals. Resistor VR1 (preset= 10k) is used to set the sensitivity of the circuit diagram. A H-bridge is fabricated with four switches like S1, S2, S3 and S4. When the S1 and S4 switches are closed, then a positive voltage will be applied across the motor. By opening the switches S1 and S4 and closing the switches S2 and S3, this voltage is inverted, allowing invert operation of the motor.

Generally, the H-bridge motor driver circuit is used to reverse the direction of the motor. The table below gives the different operations with the four switches corresponding to the above circuit.

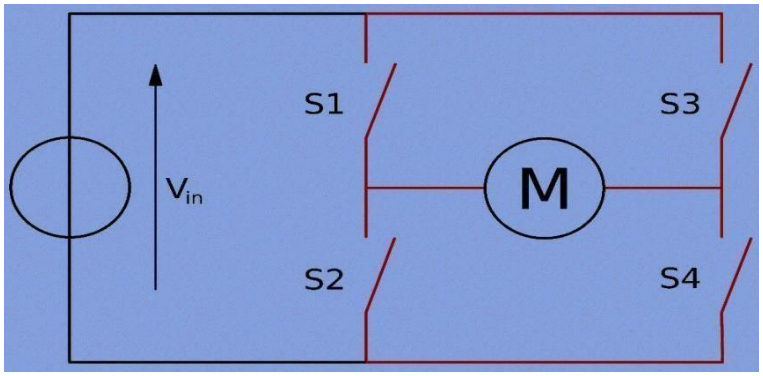
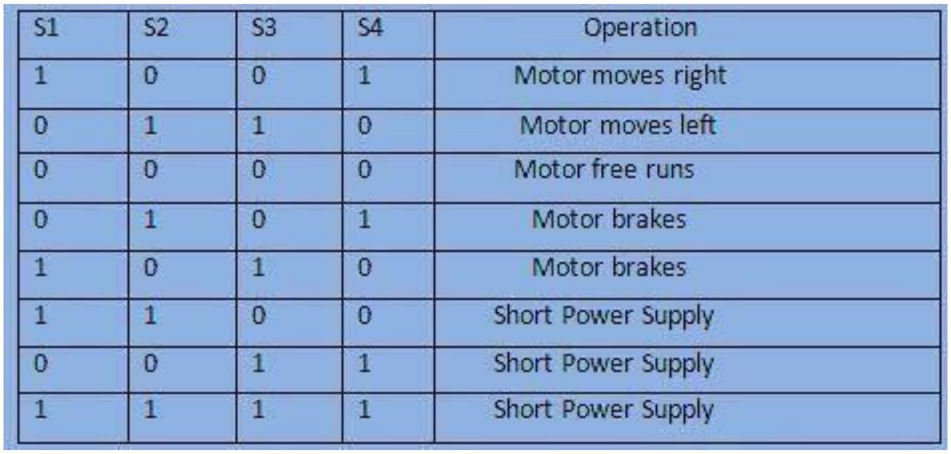


Fig.4.14:H-Bridge



S1	S2	S3	S4	Operation
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor free runs
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes
1	1	0	0	Short Power Supply
0	0	1	1	Short Power Supply
1	1	1	1	Short Power Supply

Table 4.4: H-Bridge operation

L293D MOTOR DRIVERIC

L293D IC is a typical Motor Driver IC which allows the DC motor to drive on any direction. This IC consists of 16-pins which are used to control a set of two DC motors instantaneously in any direction. It means, by using a L293D IC we can control two DC motors. As well, this IC can drive small and quiet big motors.

This L293D IC works on the basic principle of H-bridge, this motor control circuit allows the voltage to be flowing in any direction. As we know that the voltage must be change the directionofbeingabletorotatetheDCmotorinboththedirections.Hence,H-bridgecircuitusing L293D ICs are perfect for driving a motor. Single L293D IC consists of two H-bridge circuit inside which can rotate two DC motors separately. Generally, these circuits are used in robotics due to its size for controlling DC motors.

PIN DIAGRAM OF A L293D MOTOR DRIVER IC CONTROLLER

- Pin-1(Enable1-2): This pin is also called as a master control pin.
- Pin-2 (Input-1): When the input pin is high, then the flow of current will be through output1
- Pin-3 (Output-1): This output-1 pin must be connected to one of the terminals of the motor
- Pin4 &5: These pins are ground pins
- Pin-6 (Output-2): This pin must be connected to one of the terminals of the motor.
- Pin-7 (Input-2): When this pin is HIGH then the flow of current will be though output2

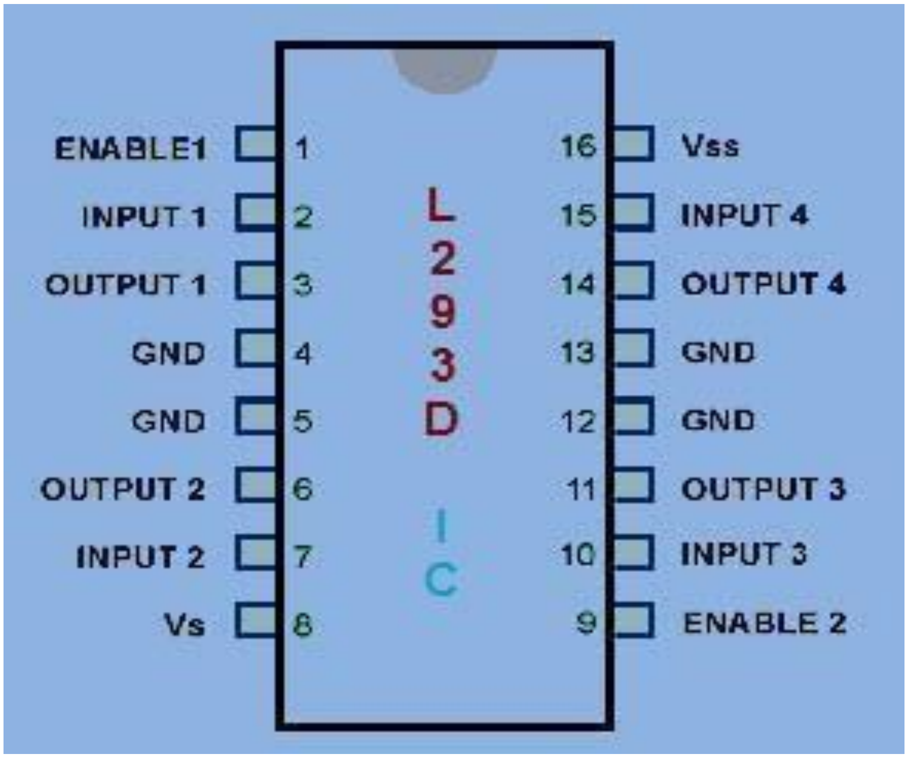
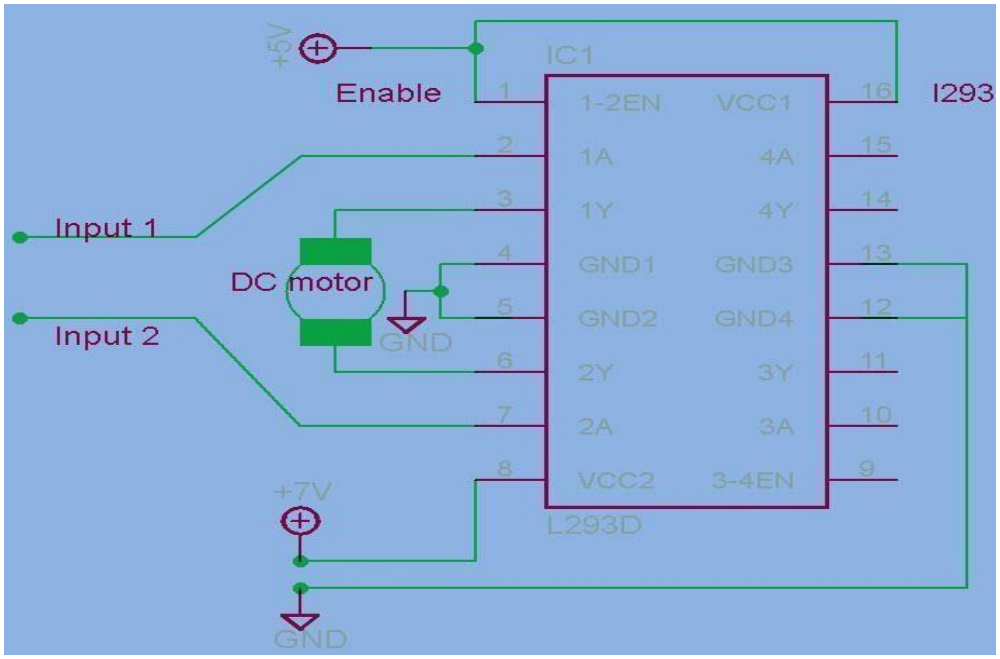


Fig. 4.15: H-Bridge Pin configuration

- Pin-8 (Vcc2): This is the voltage pin which is used to supply the voltage to themotor.
- Pin-16 (VSS): This pin is the power source to the integrated circuit.
- Pin-15 (Input-4): When this pin is high, then the flow of current will be throughoutput-4.
- Pin-14 (Output-4): This pin must be connected to one of the terminals of the motor
- Pin-12 & 13: These pins are ground pins
- Pin-11 (Output-3): This pin must be connected to one of the terminals of the motor.
- Pin-10 (Input-3): When this pin is high, then the flow of current will throughoutput-3
- Pin-9(Enable3-4):This pin is also called as a master control pin for the right part of the IC.



H-BRIDGE MOTOR CONTROL CIRCUIT USING L293DIC

Fig. 4.16: Pin connection with motor

Let‘s consider, when a motor is connected to the o/p pins 3 and 6 on the left side of the IC. For rotating of the motor in clockwise direction, then the i/p pins have to be provided with Logic 0 and Logic 1.

When Pin-2= logic 1 & pin-7 = logic 0, then it rotates in clockwise direction.

Pin-2=logic 0 & Pin7=logic1, then it rotates in anti-clock direction Pin-2= logic 0 & Pin7=logic 0, then it is idle (high impedance state) Pin-2= logic 1 & Pin7=logic 1, then it is idle In a similar way the motor can also operate across input pin-15 and pin-10 for the motor on the right hand side. The L4293D motor driver IC deals with huge currents, due to this reason, this circuit uses a heat sink to decrease the heat. Therefore, there are 4-ground pins on L293D IC. When we solder these pins on the PCB (printed circuit board), then we can get a huge metallic area between the ground pins where the heat can be produced.

4.1.7 DC MOTOR

DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills.

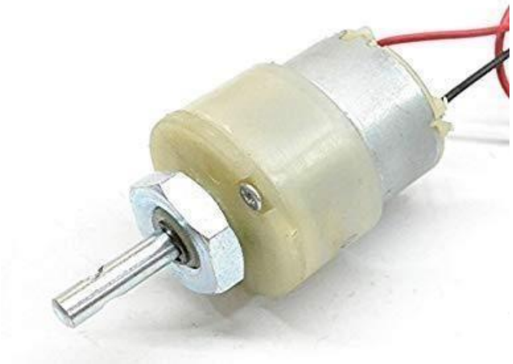


Fig. 4.17: DC motor

4.1.8 Gas Sensor

This sensor is used to sense the leakage of LPG. In normal conditions the output of this sensor is ‘high’ and it goes ‘low’, when the LPG is sensed.

Fig :4.18 Gas Sensor

The MQ series of gas sensors use a small heater inside with an electro-chemical sensor. They are sensitive for a range of gasses and are used indoors at room temperature.

They can be calibrated more or less (see the section about "Load-resistor" and "Burn-in") but a know concentration of the measured gas or gasses is needed for that.

The output is an analog signal and can be read with an analog input of the Arduino.

The preferred wiring is to connect both 'A' pins together and both 'B' pins together. It is safer and it is assumed that is has more reliable output results. Although many schematics and datasheets show otherwise, you are advised to connect both 'A' pins together and connect both 'B' pins together.

List of sensors

MQ-2

Sensitive for Methane, Butane, LPG, smoke.

This sensor is sensitive for flamable and combustible gasses. The heater uses 5V.

MQ-3

Sensitive for Alcohol, Ethanol, smoke

The heater uses 5V

Q-4

Sensitive for Methane, CNG Gas

The heater uses 5V.

MQ-5

Sensitive for Natural gas, LPG The heater uses

5V.

4.1.9 Nodemcu

The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, Espressif Systems (A Shanghai-based Semiconductor Company) has released an adorable, bite-sized WiFi enabled microcontroller – ESP8266, at an unbelievable price! For less than $3, it can monitor and control things from anywhere in the world – perfect for just about any IoT project.

4.2 Software:

- Arduino IDE
- Embedded C

4.2.1 Arduino IDE

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an

Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

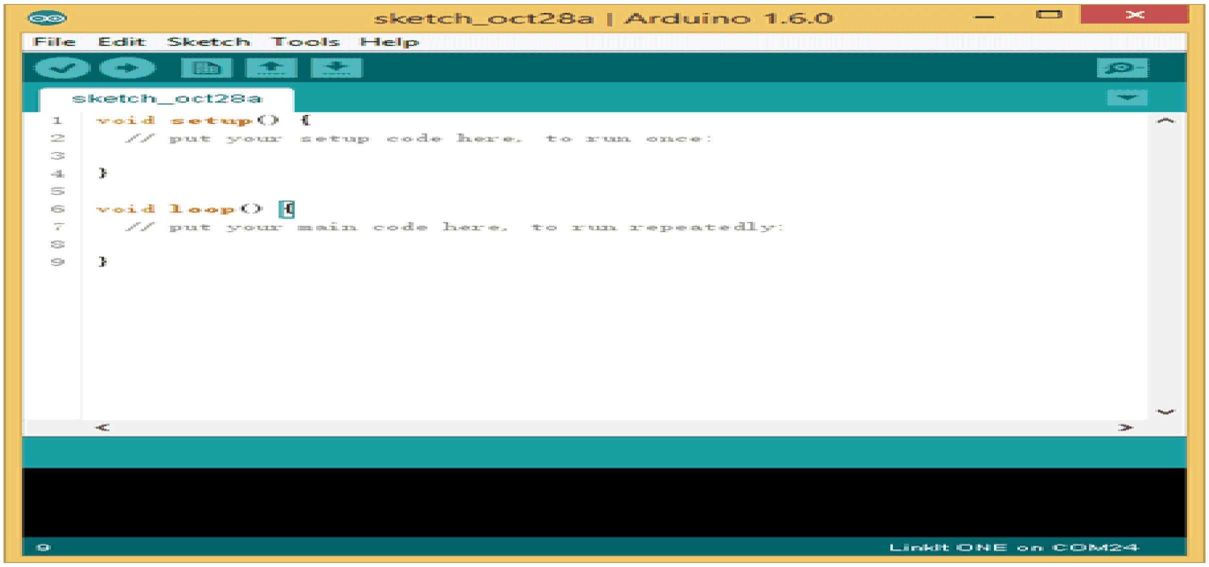
A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

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

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

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

- **loop():** After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

9 LinkIt ONE on COM24

4.2.2Embedded C:

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems.

Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

What is an Embedded System?

An Embedded System can be best described as a system which has both the hardware and software and is designed to do a specific task. A good example for an Embedded System, which many households have, is a Washing Machine.

Chapter 5

SYSTEM ANALYSIS

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

5.1 Feasibility Study

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an ill- conceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

5.1.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.1.3 Social Feasibility

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

5.2 Analysis

5.2.1 Performance Analysis

For the complete functionality of the project work, the project is run with the help of healthy networking environment. Performance analysis is done to find out whether the proposed system. It is essential that the process of performance analysis and definition must be conducted in parallel.

5.1.2 Technical Analysis

System is only beneficial only if it can be turned into information systems that will meet the organization's technical requirement. Simply stated this test of feasibility asks whether the system will work or not when developed & installed, whether there are any major barriers to implementation. Regarding all these issues in technical analysis there are several points to focus on:-

Changes to bring in the system: All changes should be in positive direction, there will be increased level of efficiency and better customer service.

Required skills: Platforms & tools used in this project are widely used. So the skilled manpower is readily available in the industry.

Acceptability: The structure of the system is kept feasible enough so that there should not be any problem from the user's point of view.

5.1.3 Economical Analysis

Economic analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we need not have any routers which are highly economical. So the system is economically feasible enough.

Chapter 6

SYSTEM DESIGN

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

6.1 System development methodology

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

6.1.1 Model phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

Requirement Analysis: This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

System Design: Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:-algorithm**,** data structure**,** software architecture etc.

Coding: In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code.

Implementation: The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation

Testing: In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

Maintenance: The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

6.1.2 Advantages of the Waterfall Model

- Clear project objectives.
- Stable project requirements.
- Progress of system is measurable.
- Strict sign-off requirements.
- Helps you to be perfect.
- Logic of software development is clearly understood.
- Production of a formal specification
- Better resource allocation.

- Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.
- Less human resources required as once one phase is finished those people can start working on to the next phase.

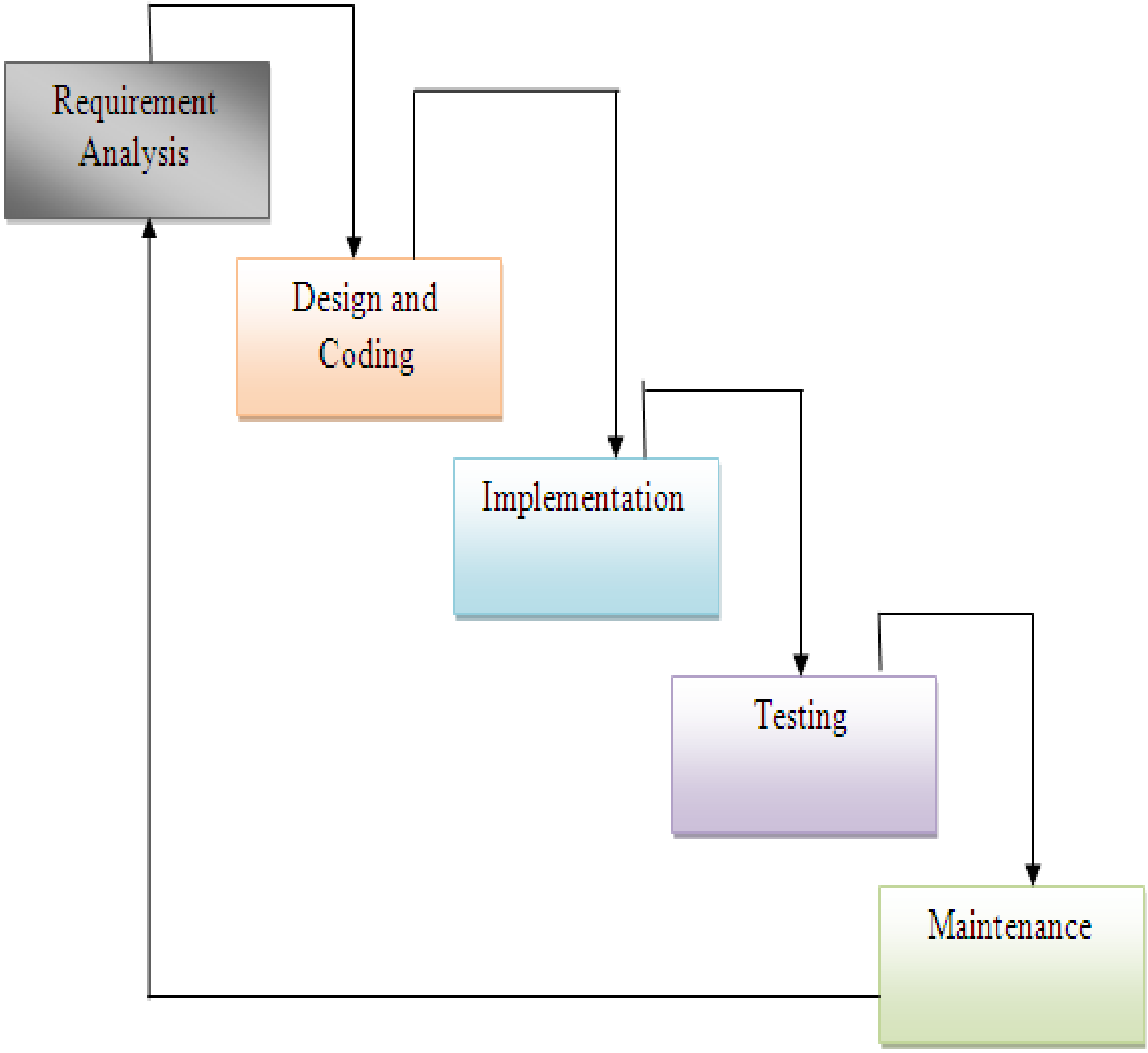


Fig 6.1: Waterfall model

6.2 System Architecture

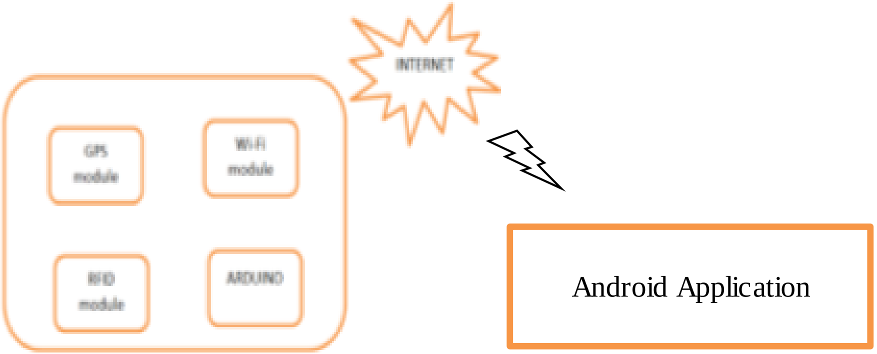


Fig 6.2: System Architecture

The proposed embedded system consists of hardware block as well as software part. The hardware components for implementation require a microcontroller, RFID reader , RFID cards, GPS module, Wi-Fi module, LCD module. Moreover the development of mobile application requires corresponding android studio for software development.

1. Microcontroller: Arduino Mega microcontroller is used as the central part of the school bus sub system. C language program saved in the memory of the controller ensures proper module functionality .The microcontroller has 16 analog inputs, 54 digital input/output pins as well as 4 UARTS. The UARTs are used to connect RFID, GPS and the Wi-Fi module. Arduino IDE is the software used to develop the code.

2. GPS Receiver: SIM 28 ML GPS receiver with low power consumption is used. NMEA format is followed by signals received by the module. It provided co-ordinates of any location on planet Earth with exact Universal Time Coordinated time (UTC). It is used for live tracking with high accuracy and reliability. GPRMC signal contains location information with co-ordinate specification.

3. ESP8266 Wi-Fi module: Wi-Fi module is used to transfer the data to remote server periodically. It follows AT commands and can be connected to the server using TCP protocol. It requires 3.3V power supply and is compact in size. It is easily configurable using Arduino.

4. RFID module: EM18 RFID reader is used along with passive tags for student identification. It operates at frequency of 125KHz. The module radiates 125KHz through its coils and when a

125KHz passive RFID tag is brought into this field it will get energized from this field. These passive RFID tags mostly consist of CMOS IC EM4102 which can get enough power for its working from the field generated by the reader. By changing the modulation current through the coils, tag will send back the information contained in the factory programmed memory array.

5. LCD Display: 16x2 LCD display is used to verify student status at the time of entrance as well as exit. It is compatible with the Hitachi HD44780 driver. It can be easily interfaced with arduino board. It forms part of school bus subsystem.

Chapter 7

IMPLEMENTATION

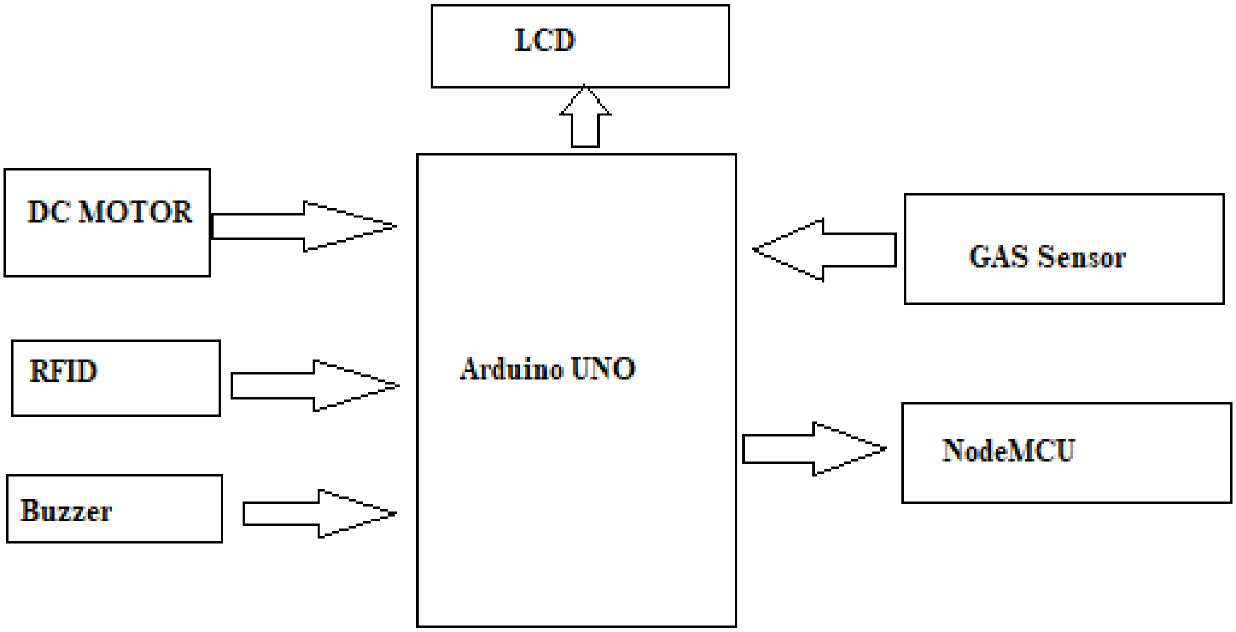


Fig 7.1 Flowchart of model

The system has two sub systems namely the school bus sub system and remote server sub system. The school bus subsystem is associated with student identification using RFID and location tracking using GPS. Each entry and exit in bus involves activation of RFID reader and acquisition of student ID. This is used to determine student status which is transmitted to server via Wi-Fi module. The location is transmitted at regular intervals to server to track the bus. The bus sub system is shown in above Figure.

The server subsystem involves remote database that stores the student status as well as GPS co-ordinates along with student details. This is used to display relevant information to parents after login. Server subsystem is used to update the data and relay the same to application. Each student has a passive RFID tag which stores unique data for identification. When the tag is in vicinity of reader, internal inductive current produced by the tag in response to the wireless signal transmitted by reader allows the tag to provide the data to the reader. It can work without manual intervention at all making the process automatic. Server sub system takes care of the data received from bus sub system. It updates the database based on the received query. It extracts relevant data to be included in the application when checked by parents.

Arduino - Installation

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

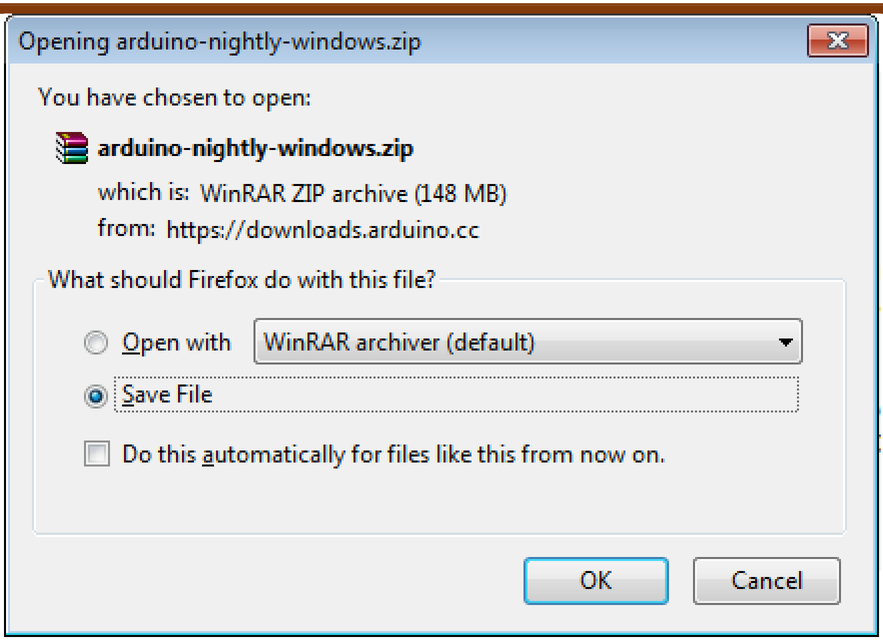
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



Opening arduino-nightly-windows.zip

You have chosen to open:

arduino-nightly-windows.zip

which is: WinRAR ZIP archive (148 MB)

from: https://downloads.arduino.cc

What should Firefox do with this file?

○ Open with WinRAR archiver (default)

● Save File

□ Do this automatically for files like this from now on.

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

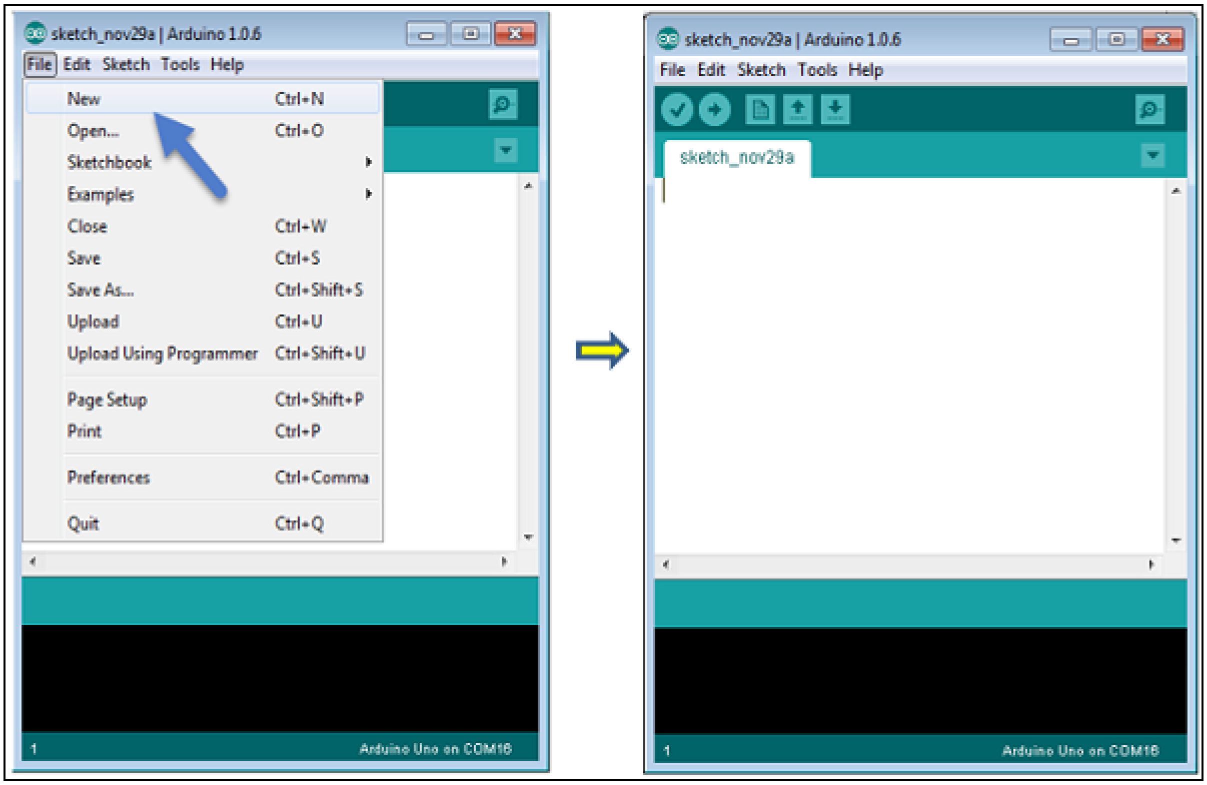
After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.



To open an existing project example, select File → Example → Basics → Blink.

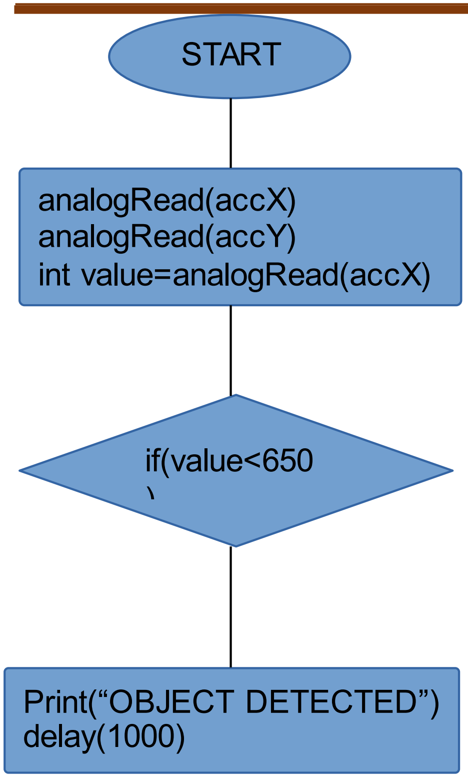


Fig 7.1 Accident Detector Working

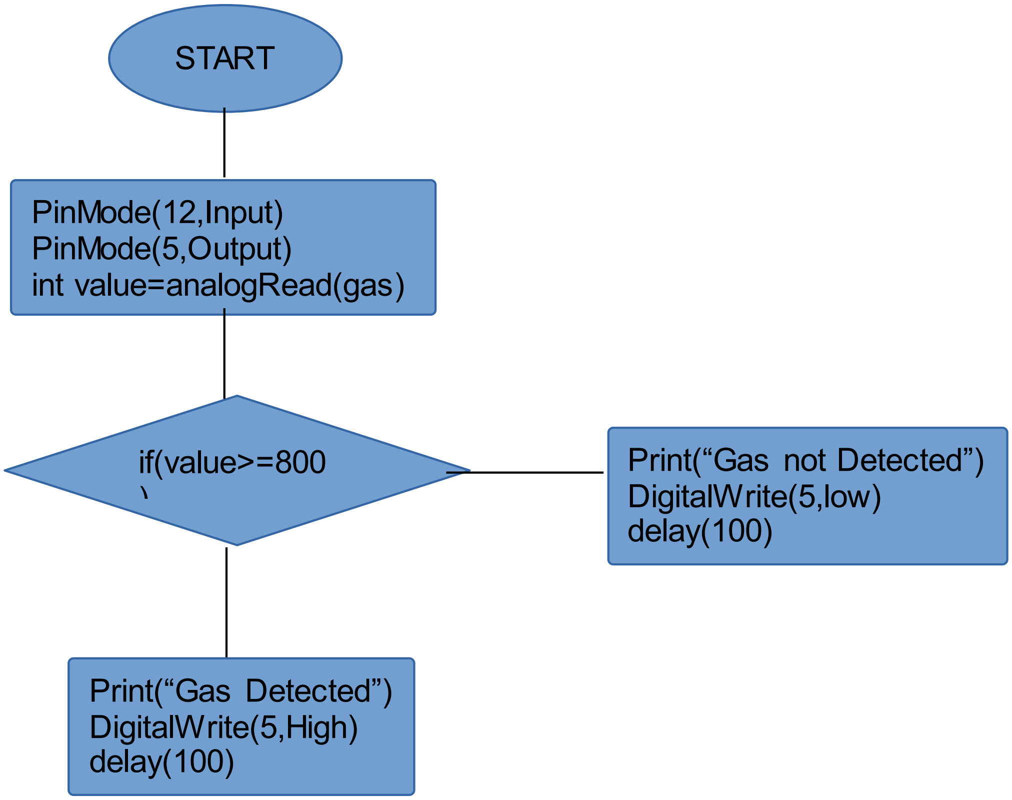


Fig 7.2 Gas Detector Working

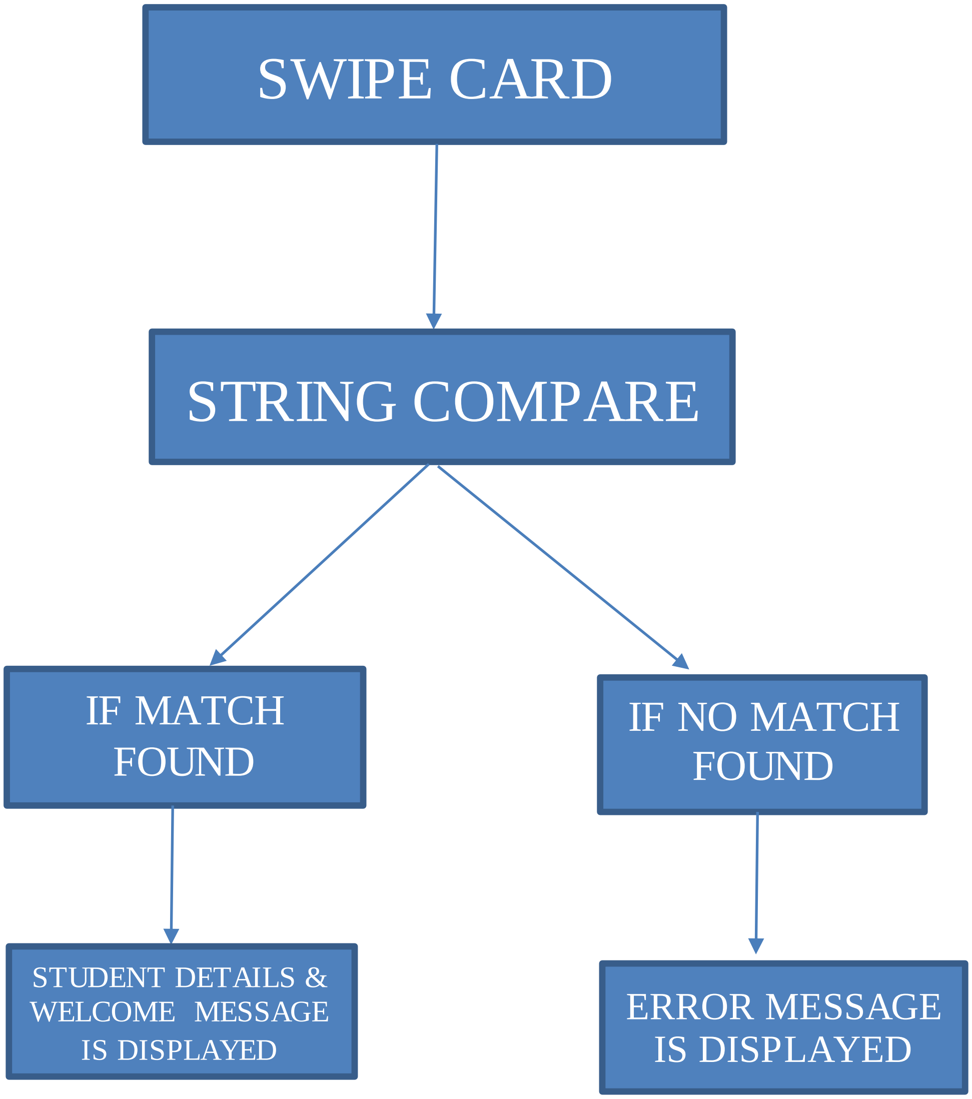

SWIPE CARD
STRING COMPARE
IF MATCH FOUND
IF NO MATCH FOUND
STUDENT DETAILS & WELCOME MESSAGE IS DISPLAYED
ERROR MESSAGE IS DISPLAYED

Fig 7.3 RFID Working

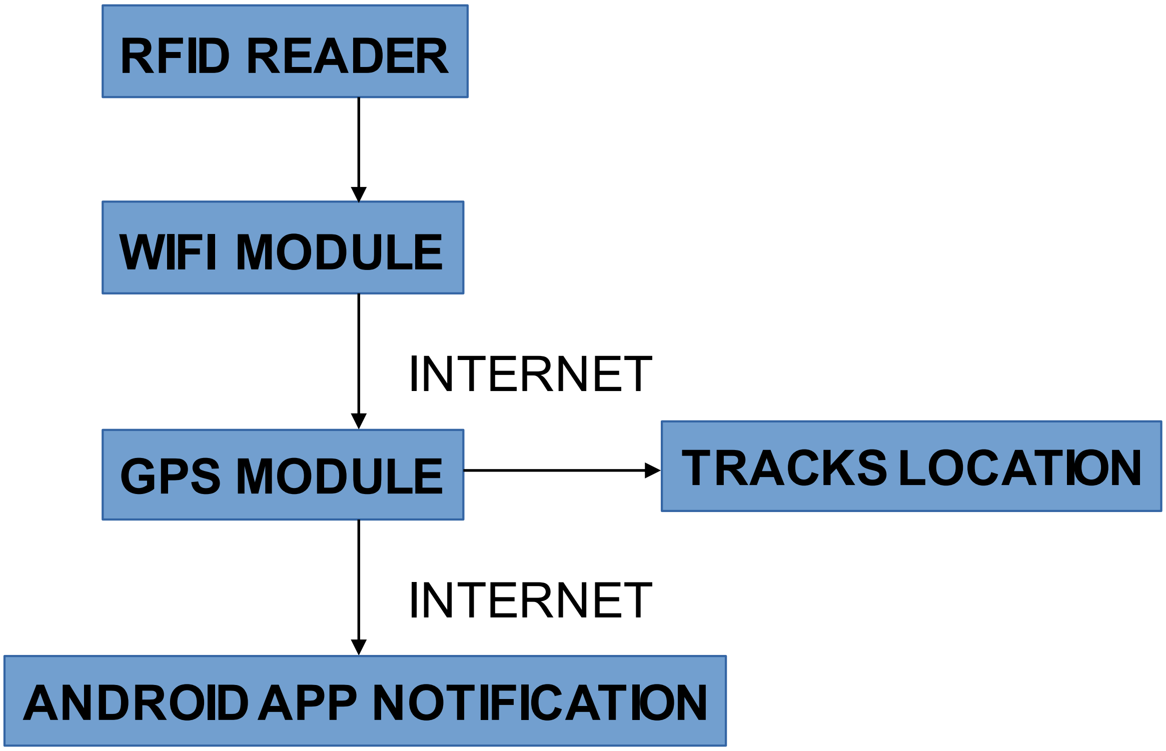


Fig 7.4 :GPS Module

BUS TRACKING CODE

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <Q2HX711.h>
#include<String.h>
#include <stdio.h>
#include <stdlib.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

SoftwareSerial rfidSerial(8,9);

int GAS=19;
int ENTRY=16;
int EXIT=17;
int BUZZER=18;

const int pingPin =7; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 6; // Echo Pin of Ultrasonic Sensor


int IN1=14;
int IN2=15;

char CARD_1[]="3A0019E7D31";
char CARD_2[]="3A001A3A5B4";
char CARD_3[]="3A001ABF079";
char CARD_4[]="3A0019F69A4";



String message="";
char ch;
long microsecondsToCentimeters(long microseconds);
int A=0;
void setup()
{
   Serial.begin(9600);
   rfidSerial.begin(9600);
   lcd.begin(16, 2);
   pinMode(IN1,OUTPUT);
   pinMode(IN2,OUTPUT);
   pinMode(BUZZER,OUTPUT);
   pinMode(GAS,INPUT);
```

```
  pinMode(ENTRY,INPUT_PULLUP);
   pinMode(EXIT,INPUT_PULLUP);

   lcd.clear();
   lcd.setCursor(0, 0);

   lcd.print("BUS TRACKING");
   lcd.setCursor(0, 1);
   lcd.print("SYSTEM...");
   Serial.println("BUS  TRACKING  SYSTEM,");
   delay(3000);

}

void loop()
{
  DISTANCE_CHECK();
 SENSOR_READING();
 MODE();

}
void VEHICLE_START()
{
 digitalWrite(IN1,HIGH);
 digitalWrite(IN2,LOW);
 delay(1000);

}
void VEHICLE_STOP()
{
 digitalWrite(IN1,LOW);
 digitalWrite(IN2,LOW);
 delay(1000);

}
void ENTRY_RFID_READING()
{
 lcd.clear();
 lcd.print("ENTRY IN BUS");
 delay(1000);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("SWIPE YOUR");
 lcd.setCursor(0,1);
 lcd.print("ID CARD...");
 Serial.println("SHOW  UR  CARD");
 delay(500);
 while(1)
 {
  while(rfidSerial.available()>0)
```

```
{

     message=rfidSerial.readString();
     int str_len = message.length() + 1;
     char textmessage[11];
     message.toCharArray(textmessage,str_len);
     rfidSerial.println(textmessage);
     textmessage[11]='\0';
     Serial.println(textmessage);
     lcd.setCursor(0,1);
     lcd.print(textmessage);
     delay(1000);
     if(!strcmp(textmessage,CARD_1))
     {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT NAME:ABC");
        lcd.setCursor(0,1);
        lcd.print("AGE-10YEAR");

        Serial.print("$ENTRY STUDENT NAME:");
        Serial.println("STUDENT1#");
//        Serial.print("AGE-: ");
//        Serial.println("10YEARS");
//        Serial.print("ENTER LOCATION: ");
//        Serial.println("VIJAYANAGER");

        delay(1000);
        VEHICLE_START();


     }
    else if(!strcmp(textmessage,CARD_2))
    {
       lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT NAME:STUDENT2");
        lcd.setCursor(0,1);
        lcd.print("AGE-20YEAR");
        Serial.print("$STUDENT NAME:");
        Serial.println("STUDENT2#");

        delay(1000);
         VEHICLE_START();

    }
    else if(!strcmp(textmessage,CARD_3))
    {
```

```
 lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT NAME:STUDENT3");
        lcd.setCursor(0,1);
        lcd.print("AGE-15YEAR");
        Serial.print("STUDENT NAME:");
        Serial.println("STUDENT3#");

        delay(1000);
         VEHICLE_START();
    }
    else if(!strcmp(textmessage,CARD_4))
    {
       lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT NAME:STUDENT4");
        lcd.setCursor(0,1);
        lcd.print("AGE-8YEAR");
        Serial.print("STUDENT NAME:");
        Serial.println("STUDENT4#");


        delay(1000);
         VEHICLE_START();

   }

   else
   {
      lcd.clear();
      lcd.print("INVALID ID..");
      delay(1000);
      break;
    }

  }
  MODE();

 }
}
void EXIT_RFID_READING()
{
 lcd.clear();
 lcd.print("EXIT FROM BUS");
 delay(1000);
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("SWIPE YOUR");
 lcd.setCursor(0,1);
 lcd.print("ID CARD...");
```

```
Serial.println("SHOW UR CARD");
  delay(500);
  while(1)
  {
    while(rfidSerial.available()>0)
   {

     message=rfidSerial.readString();
     int str_len = message.length() + 1;
     char textmessage[11];
     message.toCharArray(textmessage,str_len);
     rfidSerial.println(textmessage);
     textmessage[11]='\0';
     Serial.println(textmessage);
lcd.setCursor(0,1);
     lcd.print(textmessage);
     delay(1000);
     if(!strcmp(textmessage,CARD_1))
     {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT NAME:ABC");
        lcd.setCursor(0,1);
        lcd.print("AGE-10YEAR");
        Serial.print("$EXIT STUDENT NAME:");
        Serial.println("STUDENT1#");
//          Serial.print("STUDENT NAME: ");
//          Serial.println("ABC#");
//          Serial.print("$AGE-: ");
//          Serial.println("10YEARS#");
//          Serial.print("$EXIT LOCATION: ");
//          Serial.println("VIJAYANAGER#");

        delay(1000);
        VEHICLE_STOP();


     }
    else if(!strcmp(textmessage,CARD_2))
    {
       lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT NAME:STUDENT2");
        lcd.setCursor(0,1);
        lcd.print("AGE-20YEAR");
        Serial.print("$STUDENT NAME:");
        Serial.println("STUDENT2#");
delay(1000);
        VEHICLE_STOP();


 }
    else if(!strcmp(textmessage,CARD_3))
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT NAME:STUDENT3");
        lcd.setCursor(0,1);
        lcd.print("AGE-15YEAR");
        Serial.print("$STUDENT NAME:");
        Serial.println("STUDENT3#");

        delay(1000);
     }
     else if(!strcmp(textmessage,CARD_4))
     {
  lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT NAME:STUDENT4");
        lcd.setCursor(0,1);
        lcd.print("AGE-8YEAR");
        Serial.print("$STUDENT NAME:");
        Serial.println("STUDENT4#");

        delay(1000);
        VEHICLE_STOP();

    }

    else
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("INVALID ID..");
      delay(1000);
      break;
     }

    }

   //MODE();

  }
}
void SENSOR_READING()
{
```

```
if(digitalRead(GAS)==LOW)
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("ALCOHOL DETECTED");
    lcd.setCursor(0,1);
    lcd.print("VEHICLE STOP");
    digitalWrite(BUZZER,HIGH);
    Serial.println("$ALCOHOL  DETECTED VEHICLE STOP#");
    delay(1000);
    while(digitalRead(GAS)==LOW);
  }
  else
  {
    digitalWrite(BUZZER,LOW);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("DRIVER IS IN");
    lcd.setCursor(0,1);
    lcd.print("GOOD CONDITION");
   // Serial.println("$DRIVER IS IN GOOD CONDITION#");
    delay(1000);
//VEHICLE_START();

  }

}
void MODE()
{
  DISTANCE_CHECK();
  if(digitalRead(ENTRY)==LOW)
  {
    ENTRY_RFID_READING();
  }
  if(digitalRead(EXIT)==LOW)
  {
    EXIT_RFID_READING();
  }

}
void DISTANCE_CHECK()
{
  long duration, inches, cm;
   pinMode(pingPin, OUTPUT);
   digitalWrite(pingPin, LOW);
   delayMicroseconds(2);
   digitalWrite(pingPin, HIGH);
   delayMicroseconds(10);
```

```
 digitalWrite(pingPin,  LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

  cm = microsecondsToCentimeters(duration);
  Serial.print("Distance:");
  Serial.print(cm);
  Serial.println("cm");
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Distance:");
   lcd.print(cm);
  delay(100);
  if(cm<20)
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("OBJECT DETECTED ");
    lcd.setCursor(0,1);
    lcd.print("FRONT..");
    Serial.print("$OBJECT DETECTED FRONT#");
    delay(1000);
    VEHICLE_STOP();

  }

}
long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}


char Serial_read(void)
{
      char ch;
      while(Serial.available() == 0);
      ch = Serial.read();
      return ch;
}
```

TELEGRAM BLINK APP CODE

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "CTBot.h"        //arduino json should be 5.13.2v
//#define  msg_sender_id 884359289
#define  msg_sender_id  1263660424

CTBot myBot;
String ssid = "Redmi";      // REPLACE  mySSID WITH YOUR WIFI SSID
String pass = "dontconnect";
 // REPLACE  myPassword YOUR WIFI PASSWORD, IF ANY
String token = "1227270559:AAFdZ_D_pPgahph4Uyu-GOahbIHTToVc1UE";


uint8_t led  = D0;            // the onboard ESP8266 LED.
                         // If you have a NodeMCU you can use the BUILTIN_LED pin
                         // (replace 2 with BUILTIN_LED)
char Start_buff[70];
int i,z;
char ch;
int str_len;
 char textmessage[20];
TBMessage msg;
void MESSAGE_SEND();
void WAITING();
void SEND_SERVER();
void buffer_clear();
char auth[] = "mPRtoKKtY3ujDBHucFUgA6_nR0iNnlWh";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid1[] = "Technofly";
char pass1[] = "786786123450";
//WidgetLCD lcd(V0);
void TEST();
//char ch;
WidgetTerminal terminal(V0);
void setup()
{

  Serial.begin(9600);
//  Blynk.begin(auth, ssid1, pass1);
//  delay(2000);
  // You can also specify server:
```

```
//Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
Serial.println("Starting TelegramBot...");

    // connect the ESP8266 to the desired access point
    myBot.wifiConnect(ssid, pass);

    // set the telegram bot token
    myBot.setTelegramToken(token);
   // myBot.setTelegramToken(token1);

    // check if all things are ok
    if (myBot.testConnection())
        Serial.println("\ntestConnection OK");
    else
        Serial.println("\ntestConnection NOK");
//  Blynk.virtualWrite(V0,"FOREST  SERVER");
//   Blynk.virtualWrite(V0,"\r\n");
   TEST();
    //MESSAGE_SEND();
}
char Serial_read(void)
{
      char ch;
      while(Serial.available() == 0);
      ch = Serial.read();
      return ch;
}

void TEST()
{

String ssid = "Redmi";      // REPLACE  mySSID WITH YOUR WIFI SSID
String pass = "dontconnect";
 // REPLACE  myPassword YOUR WIFI PASSWORD, IF ANY
String token = "1227270559:AAFdZ_D_pPgahph4Uyu-GOahbIHTToVc1UE";

//char auth[] = "mPRtoKKtY3ujDBHucFUgA6_nR0iNnlWh";
//
//char ssid1[] = "Technofly";
//char pass1[] = "786786123450";


uint8_t led = D0;            // the onboard ESP8266 LED.
                            // If you have a NodeMCU you can use the BUILTIN_LED pin
                            // (replace 2 with BUILTIN_LED)
```

```
TBMessage msg;

  // initialize  the Serial
  Serial.begin(9600);
  Serial.println("Starting TelegramBot...");
  // connect the ESP8266 to the desired access point
  myBot.wifiConnect(ssid, pass);

  // set the telegram bot token
  myBot.setTelegramToken(token);

  // check if all things are ok
  if (myBot.testConnection())
    Serial.println("\ntestConnection OK");
  else
    Serial.println("\ntestConnection NOK");

  // set the pin connected to the LED to act as output pin
  pinMode(led, OUTPUT);
  digitalWrite(led, HIGH); // turn off the led (inverted logic!)
 //myBot.sendMessage(msg_sender_id, "WELCOME TO TECHNOFLY OPERATE THE DEVICES");
MESSAGE_SEND();

while(1) {
  // a variable to store telegram message data

  // if there is an incoming message...
  if (myBot.getNewMessage(msg)) {

    if (msg.text.equalsIgnoreCase("START")) {              // if the received message is "LIGHT ON"...
      digitalWrite(led, HIGH);                             // turn on the LED (inverted logic!)
      myBot.sendMessage(msg.sender.id, "BUS TRACKING SYSTEM SEVER IS ON ");  // notify the sender
     Serial.println(msg.sender.id);
      WAITING();
    }
     else {                                                // otherwise...
      // generate the message for the sender
      String reply;
      reply = (String)"Welcome " + msg.sender.username + (String)". Try START";
      myBot.sendMessage(msg.sender.id, reply);             // and send it

    }
  }
  // wait 500 milliseconds
  delay(500);
}
```

```
}

void loop()
{

//  Blynk.run();
// //serialEvent();
WAITING();

}

void MESSAGE_SEND()
{
   myBot.sendMessage(msg_sender_id, "SEND START TO CONTINUE");
  // myBot.sendMessage(msg_sender_id1, "WELCOME TO ATM");

}

void WAITING()
{
   Serial.println("WAITE");

  buffer_clear();

 while(1)
 {     // buffer_clear();
//    Blynk.run();
      if (Serial.available()  > 0)
      {
         //Serial.println("halo");

          while(Serial_read()!='$');
          i=0;
          while((ch=Serial_read())!='#')
          {
            Start_buff[i] = ch;
             i++;
          }
          Start_buff[i]='\0';
       }
     Serial.println(Start_buff);
    // delay(1000)
    if(strcmp(Start_buff,"ENTRY STUDENT NAME:STUDENT1")==0)
    {

        String one = "STUDENT NAME:STUDENT1  ENTRY AT:https://www.google.com/maps/?q=";
    //  String two = ',' ;
      String message = one +"13.0332" +','+ "77.5186";
      String str = "C";
      String all = str + message;
      // Convert String to char array
      int str_len = message.length() + 1;
      char textmessage[str_len];
      message.toCharArray(textmessage,str_len);

       myBot.sendMessage(msg_sender_id, textmessage);
        delay(1000);
//       Blynk.virtualWrite(V0,textmessage);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
    }
     else if(strcmp(Start_buff,"EXIT STUDENT NAME:STUDENT1")==0)
     {


        String one = "STUDENT NAME:STUDENT1 EXIT AT:https://www.google.com/maps/?q=";
    //  String two = ',' ;
      String message = one +"13.0332" +','+ "77.5186";
      String str = "C";
      String all = str + message;
      // Convert String to char array
      int str_len = message.length() + 1;
      char textmessage[str_len];
      message.toCharArray(textmessage,str_len);

       myBot.sendMessage(msg_sender_id, textmessage);
         delay(1000);
//       Blynk.virtualWrite(V0,textmessage);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
    }
    if(strcmp(Start_buff,"ENTRY STUDENT NAME:STUDENT2")==0)
     {

        String one = "STUDENT NAME:STUDENT2 ENTRY AT:https://www.google.com/maps/?q=";
    //  String two = ',' ;
      String message = one +"13.0332" +','+ "77.5186";
      String str = "C";
      String all = str + message;
      // Convert String to char array
      int str_len = message.length() + 1;
      char textmessage[str_len];
      message.toCharArray(textmessage,str_len);

       myBot.sendMessage(msg_sender_id, textmessage);
        delay(1000);
//       Blynk.virtualWrite(V0,textmessage);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
   }
    else if(strcmp(Start_buff,"EXIT STUDENT NAME:STUDENT2")==0)
    {


        String one = "STUDENT NAME:STUDENT2 EXIT AT:https://www.google.com/maps/?q=";
    //  String two = ',' ;
      String message = one +"13.0332" +','+ "77.5186";
      String str = "C";
String all = str + message;
      // Convert String to char array
      int str_len = message.length() + 1;
      char textmessage[str_len];
      message.toCharArray(textmessage,str_len);

       myBot.sendMessage(msg_sender_id, textmessage);
         delay(1000);
//       Blynk.virtualWrite(V0,textmessage);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
   }
   if(strcmp(Start_buff,"ENTRY STUDENT NAME:STUDENT3")==0)
    {

        String one = "STUDENT NAME:STUDENT3 ENTRY AT:https://www.google.com/maps/?q=";
    //  String two = ',' ;
      String message = one +"13.0332" +','+ "77.5186";
      String str = "C";
      String all = str + message;
      // Convert String to char array
      int str_len = message.length() + 1;
      char textmessage[str_len];
      message.toCharArray(textmessage,str_len);

       myBot.sendMessage(msg_sender_id, textmessage);
        delay(1000);
//       Blynk.virtualWrite(V0,textmessage);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
   }
    else if(strcmp(Start_buff,"EXIT STUDENT NAME:STUDENT3")==0)
    {

        String one = "STUDENT NAME:STUDENT3 EXIT AT:https://www.google.com/maps/?q=";
    //  String two = ',' ;
      String message = one +"13.0332" +','+ "77.5186";
      String str = "C";
```

```
 String all = str + message;
     // Convert String to char array
     int str_len = message.length() + 1;
     char textmessage[str_len];
     message.toCharArray(textmessage,str_len);

      myBot.sendMessage(msg_sender_id, textmessage);
        delay(1000);
//       Blynk.virtualWrite(V0,textmessage);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
    }
    if(strcmp(Start_buff,"ENTRY STUDENT NAME:STUDENT4")==0)
     {

       String one = "STUDENT NAME:STUDENT4 ENTRY AT:https://www.google.com/maps/?q=";
   //  String two = ',' ;
     String message = one +"13.0332" +','+ "77.5186";
     String str = "C";
     String all = str + message;
     // Convert String to char array
     int str_len = message.length() + 1;
     char textmessage[str_len];
     message.toCharArray(textmessage,str_len);

      myBot.sendMessage(msg_sender_id, textmessage);
       delay(1000);
//       Blynk.virtualWrite(V0,textmessage);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
    }
     else if(strcmp(Start_buff,"EXIT STUDENT NAME:STUDENT4")==0)
     {


       String one = "STUDENT NAME:STUDENT4 EXIT AT:https://www.google.com/maps/?q=";
   //  String two = ',' ;
     String message = one +"13.0332" +','+ "77.5186";
     String str = "C";
     String all = str + message;
     // Convert String to char array
     int str_len = message.length() + 1;
     char textmessage[str_len];
     message.toCharArray(textmessage,str_len);

      myBot.sendMessage(msg_sender_id, textmessage);
```

```
  delay(1000);
//       Blynk.virtualWrite(V0,textmessage);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
   }
    else if(strcmp(Start_buff,"ALCOHOL DETECTED VEHICLE  STOP")==0)
   {
       myBot.sendMessage(msg_sender_id, Start_buff);
          delay(1000);
//       Blynk.virtualWrite(V0,Start_buff);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
        buffer_clear();

   }
    else if(strcmp(Start_buff,"OBJECT DETECTED FRONT")==0)
   {
       myBot.sendMessage(msg_sender_id, Start_buff);
          delay(1000);
//       Blynk.virtualWrite(V0,Start_buff);
//       Blynk.virtualWrite(V0,"\r\n");
//        delay(1000);
        buffer_clear();

   }
 }
}
void buffer_clear()
{
  for(z=0;z<60;z++)
  {
    Start_buff[z]='\0';
//    textmessage[z]='\0';

  }
}
void buffer1_clear()
{
  for(z=0;z<5;z++)
  {textmessage[z]='\0';

  }

}
```

Chapter 8

TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve:-

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

8.1 Testing Methodologies

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

8.1.1 White box testing

White box testing (clear box testing, glass box testing, and transparent box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also test paths between units during integration, and between subsystems during a system level test.

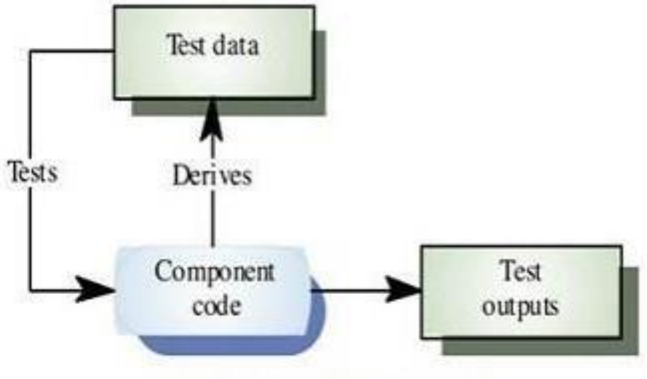


Fig 8.1: White Box Testing

Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed. Using white box testing we can derive test cases that:

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

8.1.1.1 Advantages of White Box Testing

- To start the white box testing of the desired application there is no need to wait for user face (UI) to be completed. It covers all possible paths of code which will ensure a thorough testing.
- It helps in checking coding standards.
- Tester can ask about implementation of each section, so it might be possible to remove unused/deadlines of codes helps in reducing the number of test cases to be executed during the black box testing.

- As the tester is aware of internal coding structure, then it is helpful to derive which type of input data is needed to test the software application effectively.
- White box testing allows you to help in code optimization

8.1.1.2 Disadvantages of White Box Testing

- To test the software application a highly skilled resource is required to carry out testing who has good knowledge of internal structure of the code which will increase the cost.
- Updating the test script is required if there is change in requirement too frequently.
- If the application to be tested is large in size, then exhaustive testing is impossible.
- It is not possible for testing each and every path/condition of software program, which might miss the defects in code.
- White box testing is a very expensive type of testing.
- To test each paths or conditions may require different input conditions, so in order to test full application, the tester need to create range of inputs which may be a time consuming.

8.1.2 Black box testing

Black box testing focuses on the functional requirements of the software. It is also known as functional testing. It is a software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs.

The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications. It enables us to derive sets of inputs that will fully exercise all functional requirements for a program.

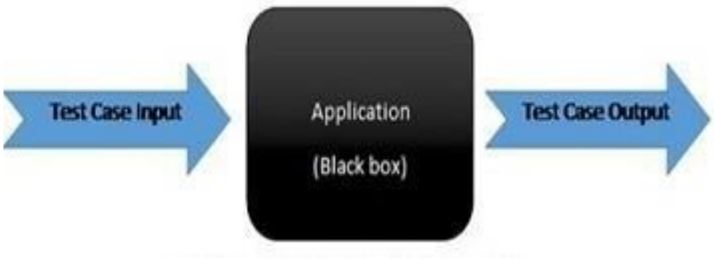


Fig 8.2: Black Box Testing

Black box testing is an alternative to white box technique. Rather it is a complementary approach that is likely to uncover a different class of errors in the following categories:-

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

8.1.2.1 Advantages of Black Box Testing

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

8.1.2.2 Disadvantages of Black Box Testing

- The test inputs need to be from large sample space. That is, from a huge set of data this will take time.

- Also it is difficult to identify all possible inputs in limited testing time. So writing test cases is slow and difficult.
- Chances are more that there will be unidentified paths during this testing.

8.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3 System Testing

This information contributes towards reducing the ambiguity about the system. For example, when deciding whether to release a product, the decision makers would need to know the state of the product including aspects such as the conformance of the product to requirements, the usability of the product, any known risks, the product's compliance to any applicable regulations,

Software testing enables making objective assessments regarding the degree of conformance of the system to stated requirements and specifications.

System testing checks complete end-end scenarios, as a user would exercise the system. The system has to be tested for correctness of the functionality by setting it up in a controlled environment. System testing includes testing of functional and nonfunctional requirements. It helps to verify and validate the system. All components of system should have been successfully unit tested and then checked for any errors after integration.

8.4 Quality Assurance

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confident that the product quality is meeting its goals. This is an “umbrella activity” that is applied throughout the engineering process**.** Software quality assurance encompasses:-

- Analysis, design, coding and testing methods and tools
- Formal technical reviews that are applied during each software engineering
- Multi-tiered testing strategy
- Control of software documentation and the change made to it.
- A procedure to ensure compliance with software development standards.
- Measurement and reporting mechanisms.

8.4.1 Quality Factors

An important objective of quality assurance is to track the software quality and assess the impact of methodological and procedural changes on improved software quality. The factors that affect the quality can be categorized into two broad groups:

- Factors that can be directly measured.
- Factors that can be indirectly measured

These factors focus on three important aspects of a software product

- Its operational characteristics
- Its ability to undergo changes
- Its adaptability to a new environment.

- Effectiveness or efficiency in performing its mission
- Duration of its use by its customer.

8.5 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

Chapter 9

RESULT AND PERFORMANCE ANALYSIS

The main goal of the proposed work is to improve the Bus system by adding the necessary additional features into the application, like accurate bus timings, correct bus numbers and moreover adding a GPS tracker into it. This study accepts input in the form of selection of the source and destination and selection of the bus travelling the distance to display the entire details about the routes and also track the location of the respective bus and give the map for the same.

The last two decades have seen growing interest in the development of Android based platform. Our review of this area shows that there have been only few approaches that provide automated tools for the functioning of the application:

1. An application has been implemented in Pune, named "Pune Bus Guide". This application gives the way to the destination correctly, but the number of drawbacks that it has is greater than the number of advantages. It does not show the passengers current location even if he/she is connected to the GPS. Also, this application has been proven useless as it does not display the bus numbers, so the passengers find it very hard to know the number and time of arrival of the respective buses. It does not have a real time bus tracking service or does not even generate maps for the users ease. This application has never been updated ever since its development. Moreover, this application has bugs which makes it all the more difficult for the user to use it.

2. Another application that was implemented in Mumbai, named "M-Indicator – Mumbai" has drawbacks like: It displays matter which is the same as what is online. Its latest updates have given issues on every Android mobile supporting even the most recent device version. The "A to B" module of buses has given problems. Whenever an option for the source to destination is selected, the field still remains blank, i.e. no bus routes are displayed.

3. The application built in Delhi named "Delhi Bus Navigator" has drawbacks like: The application works smoothly when offline, but works very badly when connected to the Internet. The application gives information about direct routes only. It does not give information about the alternate routes. This application has bugs due to which it lags all the time. Most of the time the

application crashes when requested for specific bus routes.

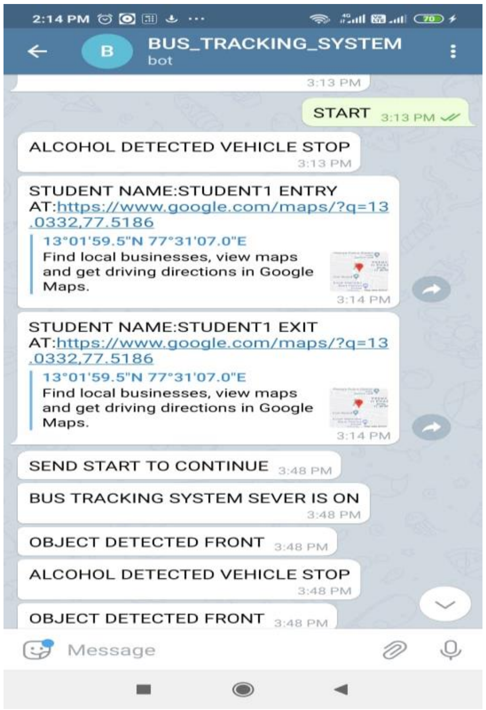
4. The application developed in Bengaluru named “Bangalore BMTC Info” has drawbacks like: The application is never in an updated condition. The application has fed in wrong routes on several buses and given no updates to fix them. After the minimization and restoration of the application, it cannot search anything. This application crashes almost always. The application is not user friendly with a complicated User Interface (UI).

5. The application developed in Chennai named “Chennai Bus Route” has the following drawbacks: The application works fine, but the bus timings have not been mentioned. Not all bus stops are updated. The application does not display maps.

Thus the proposed system will provide various facilities like check drunk and drive, children’s entry and exit information, accident emergencies, inappropriate drop, GPS tracking system, logistic management etc., which are helpful for child security.

The various advantages of the proposed system are Transportation Safety, Student Safety and Attendance of ward is a time benefit for school management.

SNAPSHOTS



3:13 PM

START 3:13 PM

ALCOHOL DETECTED VEHICLE STOP 3:13 PM

STUDENT NAME:STUDENT1 ENTRY AT:https://www.google.com/maps/?q=13.0332,77.5186
13°01'59.5"N 77°31'07.0"E
Find local businesses, view maps and get driving directions in Google Maps. 3:14 PM

STUDENT NAME:STUDENT1 EXIT AT:https://www.google.com/maps/?q=13.0332,77.5186
13°01'59.5"N 77°31'07.0"E
Find local businesses, view maps and get driving directions in Google Maps. 3:14 PM

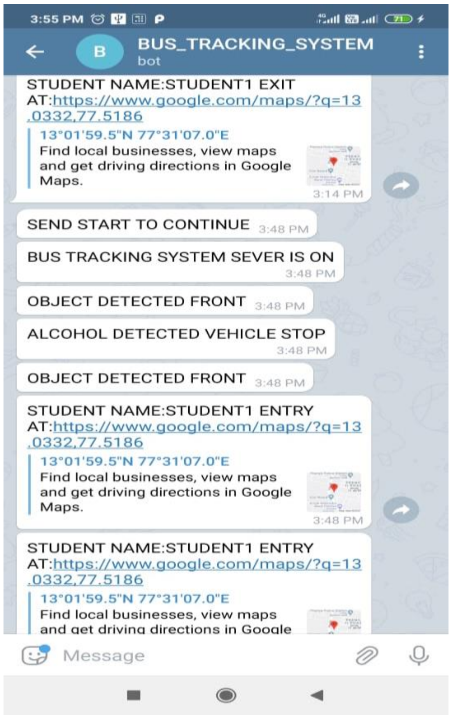
SEND START TO CONTINUE 3:48 PM

BUS TRACKING SYSTEM SEVER IS ON 3:48 PM

OBJECT DETECTED FRONT 3:48 PM

ALCOHOL DETECTED VEHICLE STOP 3:48 PM

OBJECT DETECTED FRONT 3:48 PM



STUDENT NAME:STUDENT1 EXIT AT:https://www.google.com/maps/?q=13.0332,77.5186
13°01'59.5"N 77°31'07.0"E
Find local businesses, view maps and get driving directions in Google Maps. 3:14 PM

SEND START TO CONTINUE 3:48 PM

BUS TRACKING SYSTEM SEVER IS ON 3:48 PM

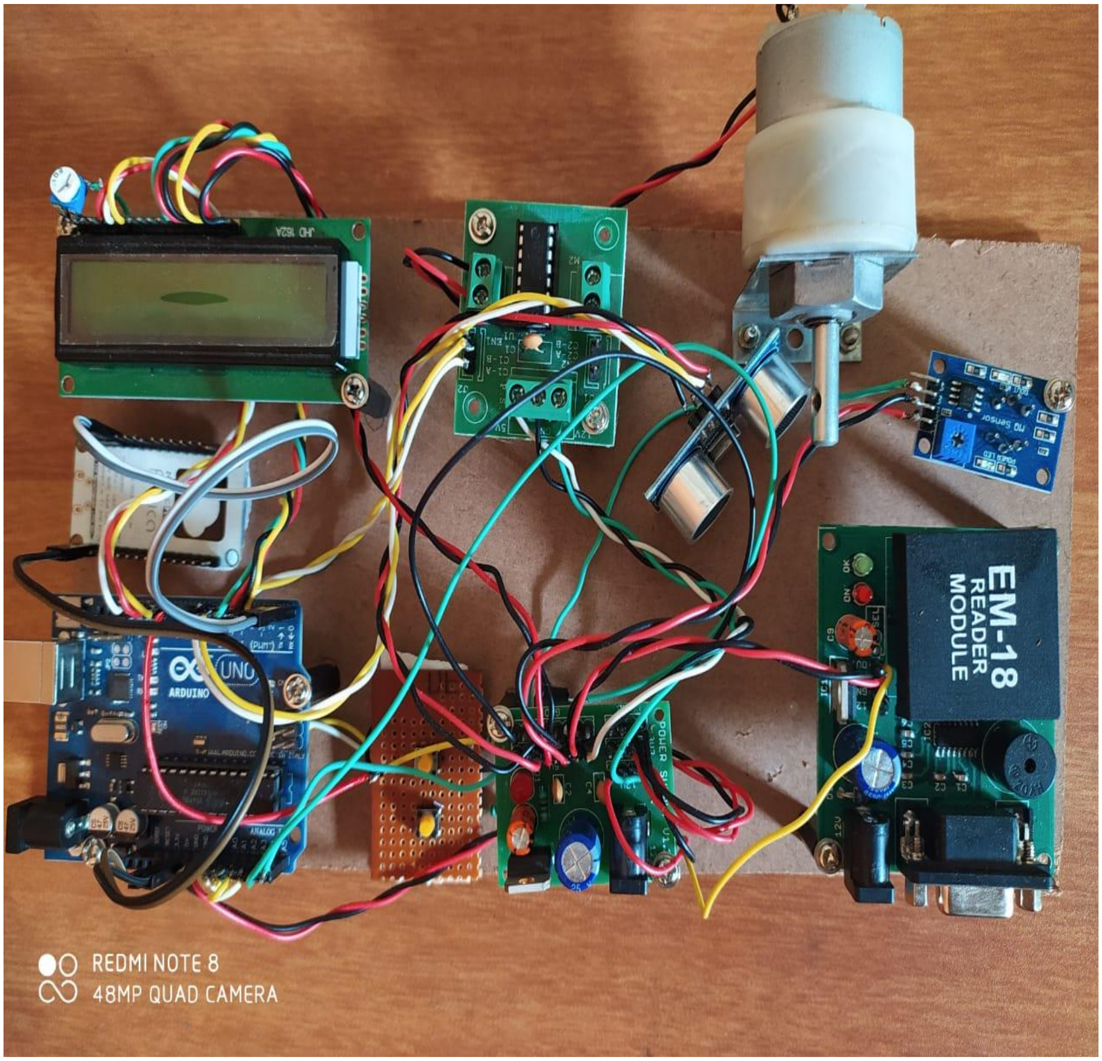
OBJECT DETECTED FRONT 3:48 PM

ALCOHOL DETECTED VEHICLE STOP 3:48 PM

OBJECT DETECTED FRONT 3:48 PM

STUDENT NAME:STUDENT1 ENTRY AT:https://www.google.com/maps/?q=13.0332,77.5186
13°01'59.5"N 77°31'07.0"E
Find local businesses, view maps and get driving directions in Google Maps. 3:48 PM

STUDENT NAME:STUDENT1 ENTRY AT:https://www.google.com/maps/?q=13.0332,77.5186
13°01'59.5"N 77°31'07.0"E
Find local businesses, view maps and get driving directions in Google


EM-18
READER
MODULE
UNO
ARDUINO
REDMI NOTE 8
48MP QUAD CAMERA

Chapter 10

CONCLUSION AND FUTURE SCOPE

The proposed system will provide various facilities like check drunk and drive, children's entry and exit information, accident emergencies, inappropriate drop, GPS tracking system, logistic management etc., which are helpful for child security.

The various advantages of the proposed system are Transportation Safety, Student Safety and Attendance of ward is a time benefit for school management.

SCOPE

Many cities have found that GPS tracking system not only improve the efficiency of city bus operation, but also encourage commuters to take the advantage of city bus system.

Many city bus system have discovered that GPS tracking system which allows to monitor the location and arrival time of their bus actually increase the number of people using city buses for routine communing.

The application is a user friendly one that anyone can access for free of cost. The basic idea for this project was to guide the bus travelers with the routes, all the possible stops that come on their way to the destination and moreover, display maps and track their locations and show the estimate remaining time required to reach. The aim is to overcome all the drawbacks faced in all the previous applications and generate fast and accurate results. The proposed system has been divided into two modules as follows. Module 1 gives information about all the routes from the source to the destination and give maps for the same. Module 2 give information about all the buses along with the bus numbers that go through the selected stops, track the location of the selected bus and send this information to the passenger giving him/her the estimate time required for the bus to reach. This is done using the Client-Server technology.

References:

[1] India's missing children, by the number. Link available at: http: // blogs.wsj.com/indiarealtime/2012/10/16/indias-missing-children-bythe-numbers/

[2] Children injured after drunk driver rams school bus into railing of bridge.See more at: http:// indianexpress.com/ article/cities/ chandigarh/children-injured-after-drunk-driver-rams-school-bus-intorailing-of bridge/#sthash.f8k6D7Vz.dpuf/

[3] Saranya. J, Selvakumar. J, "Implementation of children tracking system on android mobile terminals," Communications and Signal Processing International Conference, Vol., no., pp.961, 965, 3-5 April 2013.

[4] Mori, Y.; Kojima, H.; Kohno, E.; Inoue, S.; Ohta, T.; Kakuda, Y.; Ito, A, "A Self-Configurable New Generation Children Tracking System Based on Mobile Ad Hoc Networks Consisting of Android Mobile Terminals," Autonomous Decentralized Systems (ISADS), 2011, 10th International Symposium , vol., no., pp.339,342, 23-27 March 2011.

[5] Shu, C., "Guardian Uses Bluetooth Low Energy Tech To Keep Your Child Safe" Available at: http://techcrunch.com/2013/10/09/guardianuses-bluetooth-low-energy-tech-to-keep-your-child-safe/

[6] Akshay Shetty, Harshad Shinde, Ashwath Kumar, Ankit Verma, Popat Borse, " Proposed BLE(Bluetooth Low Energy) – Based Safety System for School Bus Network," International Journal of Technical Research and Applications, Vol. 3, Issue 5, 5th (September-October 2015), pp.272-274.

[7] Nitin Shyam, Narendra Kumar, Maya Shashi, Devesh Kumar, " SMS Based Kids Tracking and Safety System by using RFID and GSM," International Journal of Innovative Science, Engineering and Technology, Vol. 2, Issue 5, May, 2015.

[8] Khaled Shaaban, Abdelmoula Bekkali, Elyes Ben Hamida, Abdullah Kadri,"Smart Tracking System for School Buses using Passive RFID Technology to Enhance Child Safety," Journal of Traffic and Logistics Engineering,Vol,1,No.2, December, 2013.

[9] R.K. Pateriya, Sangeeta Sharma, "The Evolution of RFID Security and Privacy: A Research Survey," in IEEE International Conference on Communication Systems and Network Technologies, 2011.

[10] Abid Khan, Ravi Mishra, "GPS-GSM Based Tracking System," International Journal of

Engineering Trends and Technology, Vol, 3, Issue 2, 2012

[11] Dhivya M. and Kathiravan S., "Hybrid driver safety, vigilance and security system for vehicle," Innovations in Information, Embedded and Communication Systems, 2015 International Conference on, Coimbatore, 2015, pp. 1-6.

[12] S.S Pethakar, N.Srivastava, S.D.Suryavanshi," GPS and GSM Based Vehicle Tracking and Employee Security System," International Journal of Computer Applications (0975-8887), Vol. 62, No.6, January, 2013.