A PROJECT REPORT ON

GPS BASED TOLL COLLECTION SYSTEM

submitted in Partial fulfillment of

the requirements for the award of the degree

BACHELOR of TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the esteemed guidance of

Dr.CH.V.M.S.N.PAVAN KUMAR,M.Tech,Ph.D
Assistant Professor

Submitted by

CH.S.N.Jaya vardhan	(Y18AEC570)
A.Jagadeesh	(Y18AEC401)
E.Hemanth	(Y18AEC429)
B.Naga Chandrika	(Y18AEC409)
J.Durga Rama Krishna	(Y18AEC444)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BAPATLA ENGINEERING COLLEGE

(Affiliated to Acharya Nagarjuna University)

BAPATLA-522 102,Andhra Pradesh,India.

A.Y.2018-2022

Department of Electronics and Communication Engineering

(Affiliated to Acharya Nagarjuna University)

CERTIFICATE

This is certify that the Project entitled"**GPS BASED TOLL COLLECTION SYSTEM**"is the Bonafide work of submitted by CH.S.N.Jaya vardhan(Y18AEC570),A.Jagadeesh(Y18AEC401),E.Hemanth(Y18AEC429),B.Naga Chandrika(Y18AEC409)J.Durga Rama Krishna(Y18AEC444)in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology(B.Tech)in **Electronics and Communication Engineering**(ECE) by Acharya Nagarjuna University during the academic year 2021-2022.

Signature of HOD **Signature of Project Guide**

(Dr.N.Venkateswara Rao) **(Dr.CH.V.M.S.N.Pavan Kumar)**

ACKNOWLEDGEMENT

The Project Work entitled "**GPS BASED TOLL COLLECTION SYSTEM**" has Carried out as a part of our B.Tech in Electronics and Communication Engineering at Bapatla Engineering College,Bapatla.

We are very much thankful to **Dr.V.Damodara Naidu,Ph.D** Principal of Bapatla Engineering College,Bapatla for providing excellent academic environment in the college.

We are expressing our sincere gratitude to **Dr.N.Venkateswara Rao,Ph.D,**Head of the Department,Electronics and communication Engineering for providing constant support and stimulating environment for presenting this document.

We are extremely indebted to our guide **Dr.CH.V.M.S.N.Pavan Kumar,Ph.D**,Assistant Professor,Department of Electronics and Communication Engineering for his constant motivation and support during our project work.

We are thankful to all the faculty members and staff of the Department of Electronics and Communication Engineering is gratefully acknowledged.Finally,we thank every one who helped directly or indirectly helped us during work.

Submitted By:

CH.S.N.Jaya vardhan (Y18AEC570)

A.Jagadeesh (Y18AEC401)

E.Hemanth (Y18AEC429)

B.Naga Chandrika (Y18AEC409)

J.DurgaRamaKrishna (Y18AEC444)

INDEX

CONTENTS	PAGE NO
ABSTRACT	
Chapter 1:INTRODUCTION	**1-6**
1.1 Introduction	**1**
1.2 History	**4**
Chapter 2:LITERATURE SURVEY	**7-10**
2.1 Introduction	**7**
2.2 Literature Papers	**7**
Chapter 3:EXISTING SYSTEMS	**11-12**
3.1 Existing Styem1	**11**
3.1.2 Block Diagram	**11**
3.1.3 Description	**11**
3.1.4 Advantages	**11**
3.1.5 Disadvantages	**11**
3.2 Existing System2	**12**
3.2.1 Objective	**12**
3.2.2 Block Diagram	**12**
3.2.3 Description	**12**
3.2.3 Advantages	**12**
3.2.4 Disadvantages	**12**
Chapter 4:PROPOSED SYSTEM	**13-15**
4.1 Objective	**13**
4.2 Block Diagram	**13**
4.3 Working	**13**
4.4 Tools Required	**15**
4.5 Advantages	**15**
4.6 Applications	**15**

Chapter 5:HARDWARE IMPLEMENTATION **16-32**

5.1 Micro Controller Arduino Uno **16**

5.1.1 Arduino Uno Peripherals **17**

5.1.2 Peripheral Description **17**

5.1.3 Features **22**

5.1.4 Applications **22**

5.2 NEO-6MV2 GPS Module **23**

5.2.1 Features **23**

5.2.2 Pin Diagram **25**

5.2.3 Applications **25**

5.3 LCD Display **26**

5.3.1 what is LED 16*2? **26**

5.3.2 Pin Configuration **26**

5.3.3 Features **27**

5.3.4 Registers in LCD **28**

5.3.5 LCD Commands **29**

5.3.6 Applications **29**

5.4 ESP8266 WIFI Module **30**

5.4.1 ESP8266 History **30**

5.4.2 Features **30**

5.4.3 Pin Configuration **31**

Chapter6:SOFTWARE IMPLEMENTATION **33-50**

6.1 Arduino IDE **33**

6.1.1 Introduction to Arduino IDE **33**

6.1.2 How to Download Arduino IDE **35**

6.1.3 Libraries **40**

6.1.4 Making Pins Inputs or Outputs **40**

6.1.5 How to Select the Board **40**

6.1.6 Boot Loader **41**

6.2 Thingspeak Configuration **43**

6.3 MIT APP Inventor **45**

6.4 Source Code **48**

Chapter 7:RESULT **51**

CONCLUSION **52**

FUTURE SCOPE **53**

REFERENCES **54**

LIST OF TABLES

Table No.	Name	Page.No.
1.1	Pin Description	**25**
1.2	Pin Configuration Of ESP8266	**32**
1.3	File Option	**36**

LIST OF FIGURES

Fig.No	Figure Name	Page.No
1.1(a)	Embedded System Design Calls	**3**
1.1(b)	V Diagram	**3**
3.1	Functional Block Diagram of Existing System1	**11**
3.2	Functional Block Diagram of Existing System2	**12**
4.1	Block Diagram of Proposed System	**13**
4.2	Work Flow Diagram of GPS based Toll Collection System	**14**
5.1	Arduino Uno Configuration	**16**
5.2	NEO-6MV2 GPS Module	**23**
5.3	Pin Diagram	**25**
5.4	LCD Display	**26**
5.5	LCD Display Pin configuration	**26**
5.6	LCD Display Registers	**28**
5.7	ESP8266 Pin Configurations	**31**
6.1	Introduction to Arduino IDE	**34**
6.2	IDE Environment	**35**
6.3	Code Compilation	**37**
6.4	Menu Tab	**38**
6.5	Text Editor	**39**
6.6	Output Window	**40**
6.7	Arduino Bootloader	**42**
6.8	Choose My Channel	**43**
6.9	Create New Channel	**44**
6.10	Data Field Chart	**44**
6.11	API Key Page	**44**
6.12	Home Page Of MIT APP INVENTOR	**47**

LIST OF ABBREVIATIONS

ORT - Open Road Tolling.

GPS - Global Positioning System.

IoT - Internet of Things.

PDA - Personal Digital Assistant.

M2M - Machine To Machine.

OS - Operating System.

NOC - Network Operations Centre.

RFID - Radio Frequency Identification.

GSM - Global System for Mobile communication.

IDE - Integrated Development Environment.

MCT - Mobile Communication Technologies.

LCD - Liquid Crystal Display.

ABSTRACT

Open Road Tolling(ORT)is a type of electronic toll collection without the use of manual toll booths.The major advantage to ORT is that users are able to drive through the toll plaza at highway speeds without having to slow down to pay the toll and also reduces the traffic in the tolls.Toll collection in India differs from the practices in other countries.Selecting an optimum advanced technology system for ORT is the most crucial issue.

This project refers to some problems of worldwide applications in electronic toll collection systems for motorways and expressways.Most of these systems should use one or more of the following technologies:Global Positioning System and Internet of Things(IoT).In this project,we have analyzed the systems which meet the requirements of Indian tolls.As a result of analysis,it has turned out that only system using satellite positioning technology and IoT is the best toll solution of unique capabilities and this kind of technologically sophisticated system should be implemented in India.

This type of system has many advantages.First,it is capable working without toll booths,extra lanes,speed restrictions or complex structures along toll roads.Second is greater adaptability and accuracy to changes in charge parameters like road classes,vehicle types etc.

CHAPTER 1
INTRODUCTION

1.1 Introduction

An Embedded System is a combination of computer hardware and software,and perhaps additional mechanical or other parts,designed to perform a specific function.An Embedded System is Microontroller-based,software driven,reliable,real-time control system,autonomous,or human or network interactive,operating on diverse physical variables and in diverse environments and sold into a competitive and cost-conscious market.

An Embedded System is not a computer system that is used primarily for processing,not a software system on PC or UNIX,not a traditional business or scientific application.High-end Embedded&lower end Embedded Systems.High-end Embedded System-Generally 32,64 Bit Controllers used with OS.Examples Personal Digital Assistant and Mobile phones etc.Lower end Embedded Systems-Generally 8,16 Bit Controllers used with a minimal operating systems and hardware layout designed for the specificpurpose.

An Embedded System is a special-purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls.Unlike a general-purpose computer,such as a personal computer,an Embedded System performs one or a few pre-defined tasks,usually with very specific requirements.Since the system is dedicated to specific tasks,design engineers can optimize it,reducing the size and cost of the product.Embedded Systems are often mass-produced,benefiting from economies of scale.

Personal Digital Assistants(PDAs) or handheld computers are generally considered embedded devices because of the nature of their hardware design,even though they are more expandable in software terms.This line of definition continues to blur as devices expand.

Physically,Embedded Systems range from portable devices such as digital watches and MP3 players,to large stationary installations like traffic lights,factory controllers,or the systems controlling nuclear powerplants.

In terms of complexity Embedded Systems can range from very simple with a single microcontroller chip,to very complex with multiple units ,peripherals and networks mounted inside a large chassis or enclosure.

System Design Calls

The Embedded System is a dedicated system which performs the desired function upon power up repeatedly.Fig,1.1(a)shows the system design calls of Embedded System.

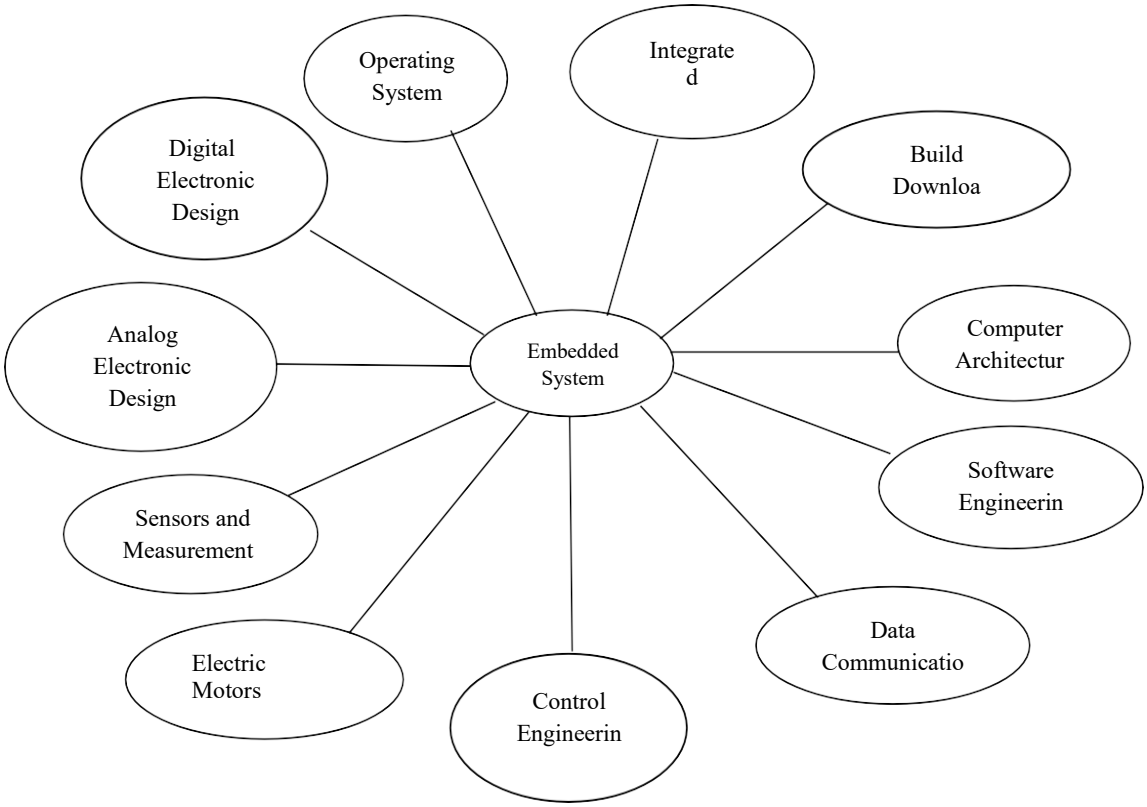


Fig.1.1(a)Embedded System Design Calls

Embedded System Design Cycle

The design of an Embedded System implies that both software and hardware are being very easy to designed in parallel.Fig.1.1 provides a schematic representation of the Embedded System design cycle.

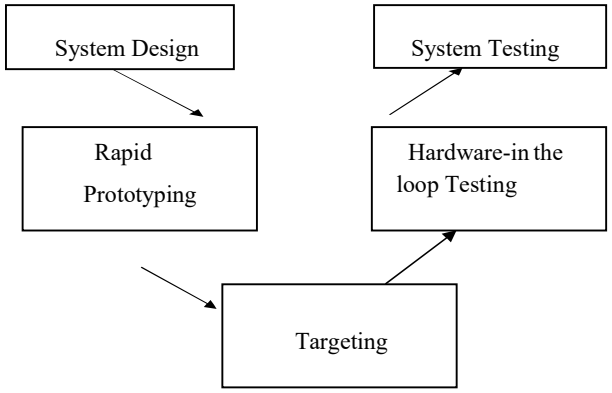


Fig.1.1(b)V Diagram

1.2 History

The Internet of Things(IoT)is rapidly evolving.There is a need to understand challenges obtained horizontal and vertical application balance and the key fundamentals required to attain the expected 50 billion connected devices in 2020.With more than 27 years in the high tech industry,Jim Chase has spent his carrer work with customers and helping them get in front of technology trends and challenges.As a trusted expert,he employs his system solutions approach to business and consumer cases world wide.

It is that methodology that has him creating solutions at Texas Instruments(TI)for the IoT and helping customers connect their products The Internet of Things(IoT)is generally thought of as connecting things to the Internet and using that connection to provide some kind of useful remote monitoring or control of those things.This definition of IoT is limited,and references only part of the IoT evolution.It is basically to reband of the exisitng Machine To Machine(M2M)Market of Today.

Internet of Things(IOT)is a concept where each device is assigned to an IP address and through that IP address anyone makes that device identifiable on internet.The mechanical and digital machines are provided with unique identifiers(UIDs)and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.Basically,it started as the“Internet of Computers.”Research studies have forecast an explosive growth in the number of“things”or devices that will be connected to the Internet.The resulting network is called the“Internet of Things”(IoT).The recent developments in technology which permit the use of wireless controlling environments like,Bluetooth and Wi-Fi that have enabled different devices to have capabilities of connecting with each other.

Using a WIFI shield to act as a Micro web server for the Arduino which eliminates the need for wired connections between the Arduino board and computer which reduces cost and enables it to work as a standalone device.The Wi-Fi shield needs connection to the internet from a wireless router or wireless hotspot and this would act as the gateway for the Arduino to communicate with the internet.Due to the

advancement of wireless technology,there are several different types of connections are introduced such as GSM,WIFI,and BT.Each of the connection has their own unique specifications and applications.Among the four popular wireless connections that often implemented in HAS project,WIFI is being chosen with its suitable capability.The capabilities of WIFI are more than enough to be implemented in the design.most of the current laptop/notebook or Smartphone come with built-in WIFI adapter.It will indirectly reduce the cost of this system.

IoT in its culmination–where we live in the data is defined as:The IoT creates an intelligent,invisible network fabric that can be sensed,controlled and programmed.IoT-enabled products employed on embedded technology that allows them to communicate,directly or indirectly,with each other or the Internet In the 1990s,Internet connectivity began to proliferate in enterprise and consumer markets,but was still limited in its use because of the low performance of the network interconnect.In the 2000s Internet connectivity became the norm for many applications and today is expected as part of many enterprise,industrial and consumer products to provide access to information.However,these devices are still primarily things on the Internet that require more human interaction and monitoring through apps and interfaces.The true promise of the IoT is just starting to be realized–when invisible technology operates behind the scenes dynamically responding to how we want“things”to act.

To date,the world has deployed about 5 billion“smart”connected things.Predictions say there will be 50 billion connected devices by 2020 and in our lifetime,we will experience life with a trillion-node network.Those are really big numbers.How things are fundamentally deployed today is a barrier to realizing those numbers.The industry will only achieve the reality of 50 billion connected devices by simplifying how things connect and communicate today Manufacturers have been connecting things to the Internet before we called it the Internet.By the mid-1990s,Web servers were being added to embedded products.Current M2M manufacturers have been integrating Internet-connected systems into high-value asset tracking,alarm systems,fleet management and the like for more than 15 years.

These M2M systems are challenging to build even though some are based on industry standard protocols.However,it is getting easier to integrate M2M systems as more powerful processors are incorporated into the end nodes.And since these processors support high-level operating systems(OS)and languages,the platform can leverage intelligent frameworks.These systems are typically tied into high-end business service layers and are managed by a Network Operations Centre(NOC).

CHAPTER 2
LITERATURE SURVEY

2.1. Introduction

The literature survey gives information about the projects that are done earlier. This literature survey gives various perspectives regarding the project.

2.2 Literature Paper

Ms.Rajkuwar patil,Ms.Kajal Thakur,Ms.Shruthi Kharat,Mr.V.V.Shetkar published a paper"Gps based toll collection system",International journal for scientific research&development.

Transportation has emerged as a dominant part of india.Toll plazas play a crucial role in maintaining the road transportation.At present,manual toll collection is widely used collection menthod in india.It significantly require a toll collector or attendant.Due to manual intervention,the processing time at toll plazas is highest.The project has been designed for the automation in toll tax payments using GPS and GSM.

Automation of toll plazas has been experienced using combination of microcontroller,RFID,Global positioning System for Mobile.Implementation of automation in toll plaza enhances the monitoring of vehicles that are travel in predestined routes.

A GPS-based highway toll collection system was developed by equipping a micro-controller with third generation (3G) and GPS connectivity. The constant acquisition of GPS coordinates acts as the basis to trace travelling vehicle and to perform all necessary toll collections. With additional works to improve the overall system accuracy and reliability, the proposed system can easily be commercialized as a future toll collection technology.

SAishwarayaRao,SahanaS,priyankaNS,IncharaG.National Conference on Image processing,computing communication,networking and Data Analytics(2018).

Road Transportation is a leading part for a development of the country.In india day-by-day number of vehicles on roads are increasing which results in traffic jam.While travelling we often visit toll booths where we see a long queue of vehicles.The toll is the system of fee collection from vehicle owners who used roadway facilities it can be done by manually using physical toll gates and manpower.In this paper,Physical toll gates are replaced by GPS based toll collection System.

Global positing system is a satellite navigation system that furnishes location and time information in all climate conditions to the user.They designed a useful and convenient application which spares people the trouble of stopping their vehicle and waiting in queues for hours at toll locations.

The operation principle of the developed GPS-based highway toll collection system was easy to follow. The system tracked the travels of a moving vehicle from the acquired GPS coordinates. The acquired coordinates were constantly compared with the predefined coordinates of the toll collection points in the database.sqlite database.

Whenever a match was detected, the toll collection for the matched destination was performed through debit transactions. User would be notified on the toll location and payment details through the LCD module. History of all toll payments were then recorded in the sql.sqlite database.Moreover, an additional feature of speed tracking was implemented. In addition, in case of the unavailability of debit transactions, these travel histories can be referred for a one-off periodic billing.

Road user charging using vehicle positioning systems. In Proceeding of IEEE International Conference on Road Transport Information and Control (pp. 126–130). London: IEEE.

Road user charging (RUC) is a key part of the British government's integrated transport policy. Whilst initial schemes in the UK may be paper-based, considerable development of electronic systems has occurred during the last 15 years. The technical development of electronic fee collection (EFC) has been largely focused on dedicated short-range communications (DSRC). The combination of an on-board positioning capability, typically using the GPS satellite system, with on-board processing and mobile communications now offers a viable alternative technology option for EFC.

Trials in Hong Kong indicated that this approach-generically referred to as based on vehicle positioning systems (VPS)-is fast becoming a viable alternative to DSRC, particularly where there are concerns over the environmental intrusion of roadside DSRC infrastructure and where integration with other intelligent transport systems (ITS) applications is an important feature.

The paper describes VPS technology for RUC, and discusses its development in Germany, Switzerland, Denmark, the UK and, in particular, the Hong Kong electronic road pricing feasibility study. It briefly describes the INITIATIVE project that is demonstrating aspects of interoperability between DSRC and VPS-based EFC, and summarises the development of standards for VPS-based EFC within ISO and CEN. Finally it compares the advantages and disadvantages of DSRC and VPS for RUC, particularly in the light of current plans in the UK.

Automated Toll Collection Using Satellite Navigation Ms.Kirti A.Lonkar , Ms. Pratibha P. Kulkarni , Ms Monalisha Dash , Mr.Abhishek Dhawan , Mr.Hemant R.Kumbhar , Mr.Monika P.Gagtap.International Journal of Computational Engineering Research.

Recently, most developments are done in the field of the Expressways Network Toll Collection. Most electronic toll collection are implemented by DSRC (Dedicated short-range communication) technology .In recent years in ETC development GPS global positioning system technique took place DSRC technique. However a new generation of electronic toll collection is rapidly developed to replace dedicated short-range communication based electronic toll collection system.

Global positioning system technology has become the new trend for road charging system, which implements electronic toll collection system based on positioning and Global System for MCT(Mobile communication technologies).

Dias, J., Matos, J. N., & Oliveira, A. S. R. (2014). The charge collector system: A new NFC and smartphone-based toll collection system. Procedia Technology, 17, 130–137.

This paper proposes a new system to collect tolls on Open Road Tolling (ORT) infrastructures. The actual Electronic Toll Collection (ETC) systems do not fulfill fundamental user requirements, such as interoperability and portability between systems and road operators (in the same or different countries), as well as advanced toll logging and reporting (capabilities ensuring user privacy, an interesting feature in car rental or sharing use cases).

The Charge Collector System (C2S) is a work in progress project that will provide some features, such as flexible payment options, recording of incurred tolls and make them available to the end user and management entities, exploring a synergy of technologies in ETC scenario, namely Dedicated Short Range Communication (DSRC), Global Navigation Satellite System (GNSS), Near Field Communication (NFC) and smartphone-based mobile applications. This system is also an approach to interoperable European ETC solutions, in a way that uses DSRC and GNSS-based solutions together.

CHAPTER 3

GPS BASED TOLL COLLECTION SYSTEM

3.1 Existing Systems 1:RFID BASED TOLL COLLECTION SYSTEM

3.1.1 Objective

The main focus of this system to collect the toll from vehicle using the RFID tags present in the vehicles.when the vehicle stop at the toll gate the RFID tag reader read the tag id and collect the data from the data base about the vehicle type and user details it will automatically collect money from wallet or bank account.

3.1.2 Block Diagram

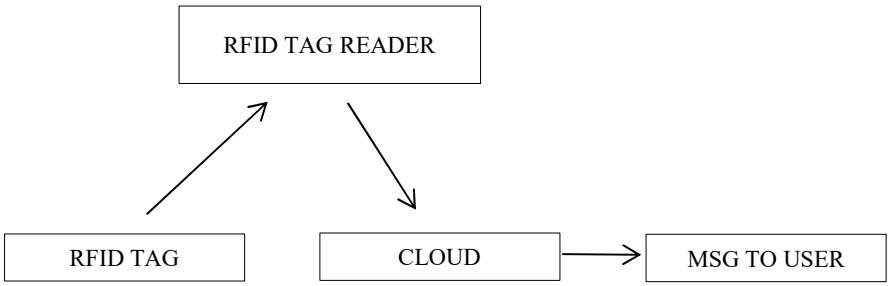


Fig.3.1 Functional Block Diagram of Existing System 1

3.1.3 Description

The main focus of this project is to scan the RFID tag present in the car and retrive the data from the data base and collect toll from according to the data of the registered vehicle.

The functional block diagram describes the monitoring of vehicle tag and detected by the RFID sensor to collect the toll from the user.

3.1.4 Advantages:-

- Easy way to collect toll
- Reduce Man Power

3.1.5 Disadvantages:-

- Radio frequencies can be disturbed easily during different weather condition.
- Sensitive Equipment.

3.2 Existing System-2

Computer Vision Based Vehicle Detection for Toll Collection System Using Embedded System and Linux.

3.2.1 Objective

The main focus of this project is to detect the vehicle Registration Number Plate using camera and computer vision algorithm.The camera scan the vehicle id and retrive the data from the cloud using the id and toll collected according to the details in the Data Base.

3.2.2 Block Diagram

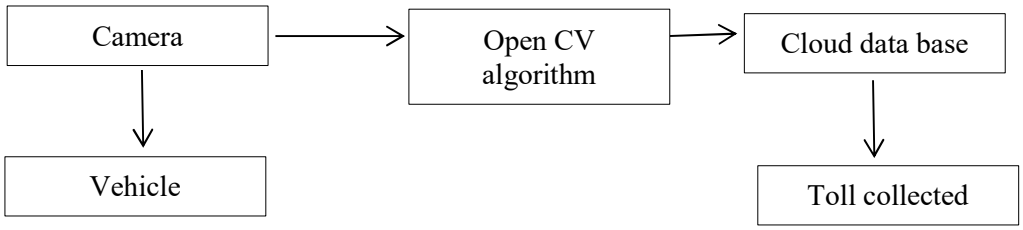


Fig.3.2 Functional Block Diagram of Existing System 2

3.2.3 Description

The main focus of this project is to scan the number plate of the vehicle and collect the data from the database about the vehicle details and other information.

According to that the toll will be collected from particular owner of the car money will be debited from the owner wallet or bank account linked by the user directly and send the conformation message to the user.

3.2.4 Advantages

- Easy way to implement.
- Reduce Man power.

3.2.5 Disadvantages

- Some times camera will not detect the number plate due to low day light.
- May be failed during rainy and foggy Days.

CHAPTER 4
PROPOSED SYSTEM
GPS BASED TOLL COLLECTION SYSTEM

4.1 Objective

The main aim of the project to collect toll from the vehicles using the GPS location of vehicle.In this system the car dashboard contains a GPS sensor which identify the location of the vehicle when its reach the determined location the toll will collect automatically from the database information and replaced physical toll gates with virtual toll gates.

4.2 Block Diagram

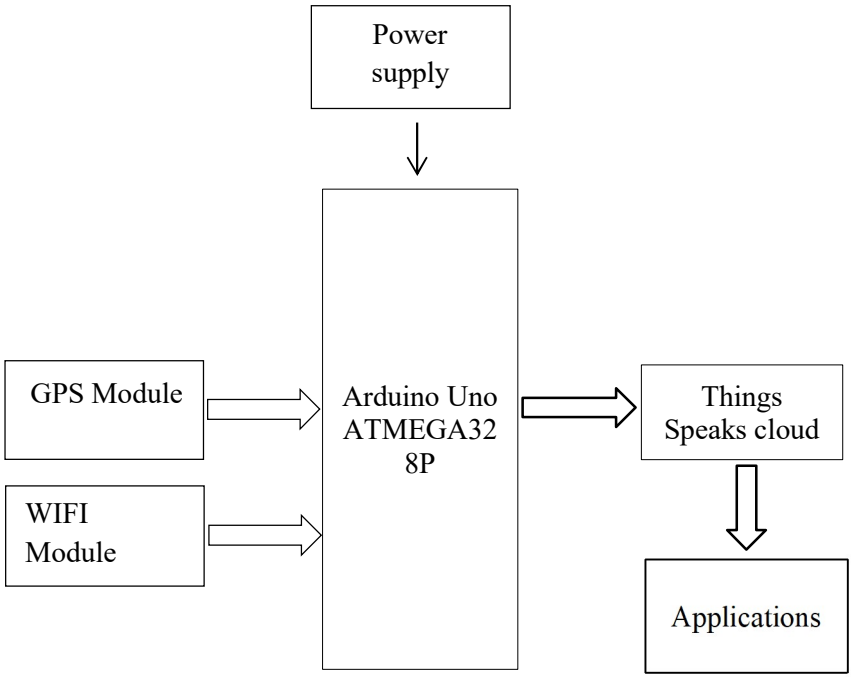


Fig.4.1 Block Diagram

4.3 Working

This system does not require infrastructure along the road,such as toll data collection bridges and toll booths.The on board unit(OBU)enables automatic usage based toll payment.The operating principle of the OBU is based on two technologies:GPS and IoT.It could be even your cell phone.

The GPS made up of at least 24 satellites at a distance of 20,200 kilometers,which continuously sends positioning signals.The system is designed to guarantee that users and receivers will have a minimum of four satellites at their disposal at any time worldwide,regardless of the weather.

This satellite system initiated by Toll Collect uses onboard units within the trucks which are compatible with GPS technology to track the distances that individual trucks travel on toll roads.This data is wirelessly transmitted to the data center(mobile application)for billing.In,if the ETC is working based on prepaid system then the cycle stops at 6th step.

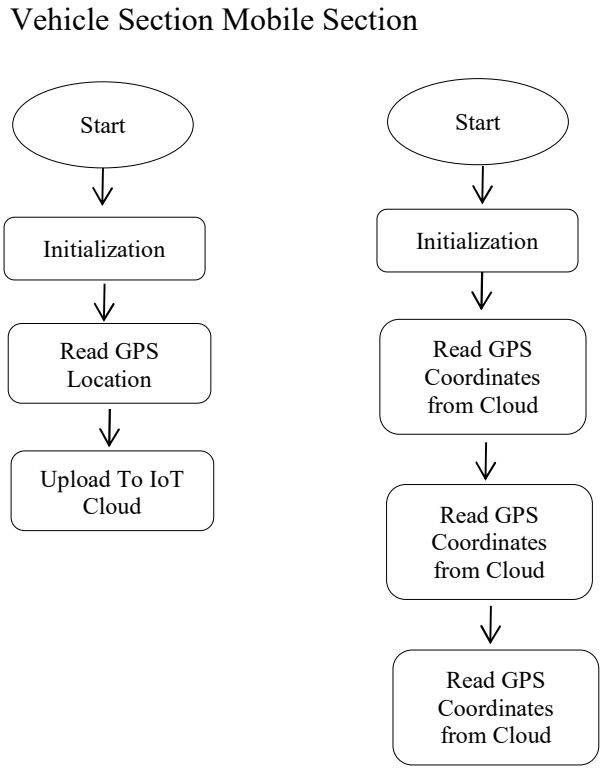


Fig.4.2 Work flow of GPS BASED TOLL COLLLECTION SYSTEM

4.4 Tools Required

Hard ware Tools

- Arduino uno.
- ESP8266 WIFI MODULE.
- GPS SENSOR.
- Power Supply.

Software Tools

- Arduino IDE.
- ThingSpeak Cloud.
- MIT APP Inventor.

4.5 Advantages

- It is low cost.
- Circuit complexity is less.

4.5 Applications

- This can be used in both indoor and outdoor environment.
- This can also be used to identify the GPS location of the vehicle

CHAPTER 5
HARDWARE IMPLEMENTATION

5.1 ARDUINO UNO

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.

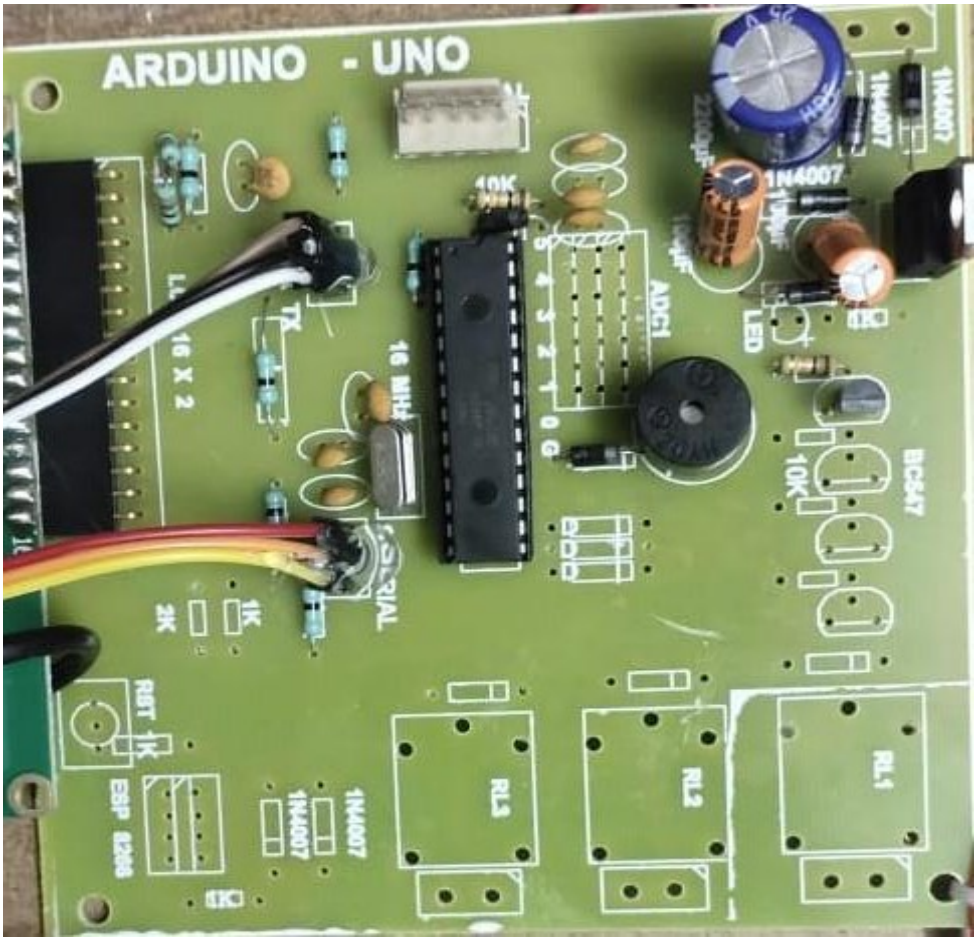


Fig.No.5.1 Arduino Uno Configuration

It contains everything needed to support the micro controller;simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

You can tinker with your UNO without working 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 ofboards.

5.1.1 ESP32 Peripherals

The ESP32 peripherals include:

- USB:The port used to transfer data and programs to the Arduino.It is also used to power theArduino.
- DC power:If you don't connect the Arduino to the computer via the USB,you can power theArduino by connecting a power supply or a battery pack to the DC power port.
- Reset:Press this button to make your program restart.
- Headers.There are four headers that expose pins.You can connect your peripherals to theArduino using those pins.
- ATMEGA328P:This is the"brain"of the Arduino Uno,the microcontroller.It sits on a socket,so if needed,you can swap it for a new one.

5.1.2 ESP32 Peripherals Description

CPU and Memory

- 3.3V(6)−Supply 3.3 output volt
- 5V(7)−Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
- GND(8)(Ground)−There are several GND pins on the Arduino,any of which can be used to groundyour circuit.

- V_{in}(9)−This pin also can be used to power the Arduino board from an external power source,likeAC mains power supply.
- 16/24-bit Instruction Set provides high code-density
- Support for Floating Point Unit
- Support for DSP instructions,such as a 32-bit multiplier,a 32-bit divider,and a 40-bit MAC
- Support for 32 interrupt vectors from about 70 interrupt sources the single-/dual-CPU interfacesinclude:
- External and internal interrupt sources.

Internal Memory

Arduino's internal memory includes:

- 448 KB of ROM for booting and core functions.
- 520 KB of on-chip SRAM for data and instructions.
- 8KB of RAM in RTC,which is called RTC FAST Memory and can be used for data storage;it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8KB of RAM in RTC,which is called RTC Slow Memory and can be used for data storage;it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 1 KB of e Fuse:256 bits are used for the system(MAC address and chip configuration)and the remaining 768 bits are reserved for customer applications,including flash-encryption and Chip-ID)

Most of the modules like ESP32 Wroom use external Flash-W25Q32**(4M Bytes!)**for storing the application code.The chip supports 4 x 16 MBytes of external QSPI flash and SRAM with hardware encryption based on AES.

ESP32 accesses the external QSPI flash and SRAM through high-speed caches.

- Up to 16 MBytes of external flash are memory-mapped onto the CPU code space,supporting 8,16 and 32-bit access.Code execution is supported.
- Up to 8 MBytes of external SRAM are memory-mapped onto the CPU data space,supporting 8,16 and 32-bit access.Data-read is supported on the flash and SRAM.Data-write is supported on the SRAM.

The two cores are named Protocol CPU(PRO_CPU)and Application CPU(APP_CPU).That basically means the PRO_CPU processor handles the WiFi,Bluetooth and other internal peripherals like SPI,I2C,ADC etc.The APP_CPU is left out for the application code.This differentiation is done in the Espressif Internet Development Framework(ESP-IDF).ESP-IDF is the official software development framework for the chip.Arduino and other implementations for the development will be based on ESP-IDF.

Analog-to-Digital Converter(ADC)

Arduino uno integrates 12-bit SAR ADCs and supports measurements on 18 channels(analog-enabled pins).The ULP-coprocessor in Arduino uno is also designed to measure voltage,while operating in the sleep mode,which enables low-power consumption.The CPU can be woken up by a threshold setting and/or via other triggers.With appropriate settings,the ADCs can be configured to measure voltage on 18 pins maximum.Table 7 describes the ADC characteristics.

Digital-to-Analog Converter(DAC)

Two 8-bit DAC channels can be used to convert two digital signals into two analog voltage signal outputs.The design structure is composed of integrated resistor strings and a buffer.This dual DAC supports power supply as input voltage reference.The two DAC channels can also support independent conversions.

Universal Asynchronous Receiver Transmitter(UART)

ESP32 has three UART interfaces,i.e.,UART0,UART1 and UART2,which provide asynchronous communication(RS232 and RS485)and IrDA support,communicating at a speed of up to 5 Mbps.UART provides hardware management of the CTS and RTS signals and software flow control(XON and XOFF).Allof the interfaces can be accessed by the DMA controller or directly by the CPU.

I²C Interface

Arduino uno has two I²C bus interfaces which can serve as I²C master or slave,depending on the user'sconfiguration.The I²C interfaces support:

- Standard mode(100 Kbit/s)
- Fast mode(400 Kbit/s)
- Up to 5 MHz,yet constrained by SDA pull-up strength
- 7-bit/10-bit addressing mode
- Dual addressing mode users can program command registers to control I²C interfaces,so thatthey have more flexibility.

Pulse Width Modulation(PWM)

The Pulse Width Modulation(PWM)controller can be used for driving digital motors and smart lights.The controller consists of PWM timers,the PWM operator and a dedicated capture sub-module.Each timer provides timing in synchronous or independent form,and each PWM operator generates a waveform for one PWM channel.The dedicated capture sub-module can accurately capture events with external timing.

LED PWM

The LED PWM controller can generate 16 independent channels of digital waveforms with configurable periods and duties.The 16 channels of digital waveforms operate with an APB clock of 80 MHz.Eight of these channels have the option of using the 8 MHz oscillator clock.Each channel can select a 20-bit timer with configurable counting range,while its accuracy of duty can be up to 16 bits within a 1 msec period.The software can change the duty immediately.Moreover,each channel automatically supports step-by-step duty increase or decrease,which is useful for the LED RGB color-gradient generator.

Serial Peripheral Interface(SPI)

Arduino features three SPIs(SPI,HSPI and VSPI)in slave and master modes in 1-line full-duplex and 1/2/4-line half-duplex communication modes.These SPIs also support the following general-purpose SPI features:

- Four modes of SPI transfer format,which depend on the polarity(CPOL)and the phase(CPHA)ofthe SPI clock
- Up to 80 MHz(The actual speed it can reach depends on the selected pads,PCB tracing,peripheralcharacteristics,etc.)
- up to 64-byte FIFO All SPIs can also be connected to the external flash/SRAM and LCD.Each SPIcan be served by DMA controllers.

5.1.3 Features

- Microcontroller:ATmega328
- Operating Voltage:5V
- Input Voltage(recommended):7-12V
- Input Voltage(limits):6-20V
- Digital I/O Pins:14(of which 6 provide PWM output)
- Analog Input Pins:6
- DC Current per I/O Pin:40 mA
- DC Current for 3.3V Pin:50 mA
- Flash Memory:32 KB of which 0.5 KB used by bootloader
- SRAM:2 KB(ATmega328)
- EEPROM:1 KB(ATmega328)
- Clock Speed:16 MHz
- On-board PCB antenna.

5.1.4 Applications

- Weighing Machines.
- Smart Socket
- Traffic Light Count Down Timer.
- Parking Lot Counter.
- Embedded systems.
- Home Automation.
- Industrial Automation.
- Medical Instrument Emergency Light for Railways.

5.2 NEO-6MV2 GPS Module

The **NEO-6MV2** is a **GPS**(Global Positioning System)module and is used for navigation.The module simply checks its location on earth and provides output data which is longitude and latitude of its position.It is from a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine.These flexible and cost effective receivers offer numerous connectivity options in a miniature(16 x 12.2 x 2.4 mm)package.The compact architecture,power and memory options make **NEO-6 modules** ideal for **battery operated mobile devices** with very strict cost and space constraints.Its Innovative design gives **NEO-6MV2** excellent navigation performance even in the most challenging environments.

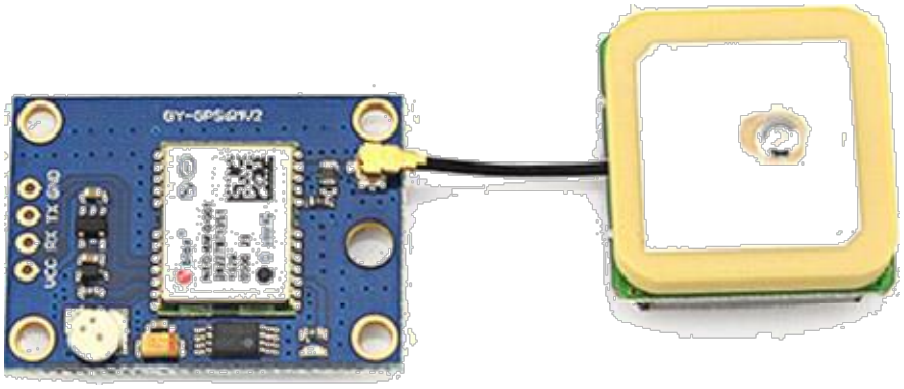


Fig.5.2 NEO-6MV2 GPS Module

5.2.1 Features

- Standalone GPS receiver
- Anti-jamming technology
- UART Interface at the output pins(Can use SPI,I2C and USB by soldering pins to the chip core)
- Under 1 second time-to-first-fix for hot and aided starts.
- Time-To-First-fix:For Cold Start 32s,For Warm Start 23s,For Hot Start<1s
- Maximum navigation update rate:5Hz
- Default baud rate:9600bps
- EEPROM with battery backup
- Sensitivity:-160dBm

- Supply voltage:3.6V
- Maximum DC current at any output:10mA
- Operation limits:Gravity-4g,Altitude-50000m,Velocity-500m/s
- Operating temperature range:-40ºC TO 85°C.

5.2.2 Pin Diagram

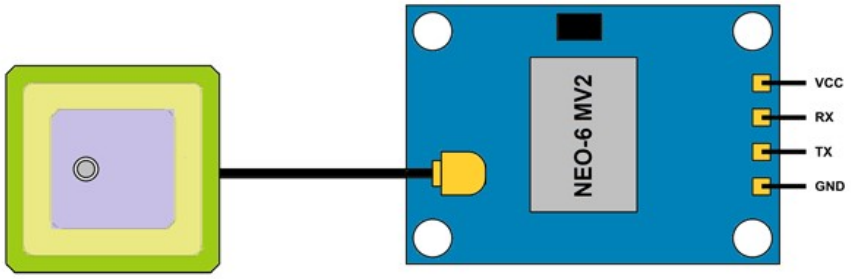


Fig.5.3 Pin Diagram

Table 1.1:Pin Description

Pin Name	Description
VCC	Positive power pin
RX	UART receive pin
TX	UART transmit pin
GND	Ground

5.2.3 Applications

- Location–determining a position.
- Navigation–getting from one location to another.
- Tracking–monitoring object or personal movement.
- Mapping–creating maps of the world.
- Timing–bringing precise timing to the world.

5.3 LCD Display

5.3.1 What is the LCD 16×2?

The term LCD stands for liquid crystal display.It is one kind of electronic display module used in an extensive range of applications like various circuits&devices like mobile phones,calculators,computers,TV sets,etc.These displays are mainly preferred for multi-segmentlight-emitting diodesand seven segments.The main benefits of using this module are inexpensive;simply programmable,animations,and there are no limitations for displaying custom characters,special and even animations,etc.

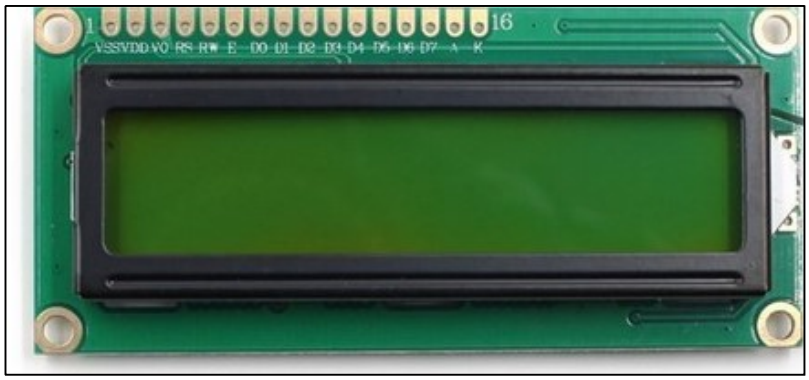


Fig.5.4 LCD Display

5.3.2 Pin configuration

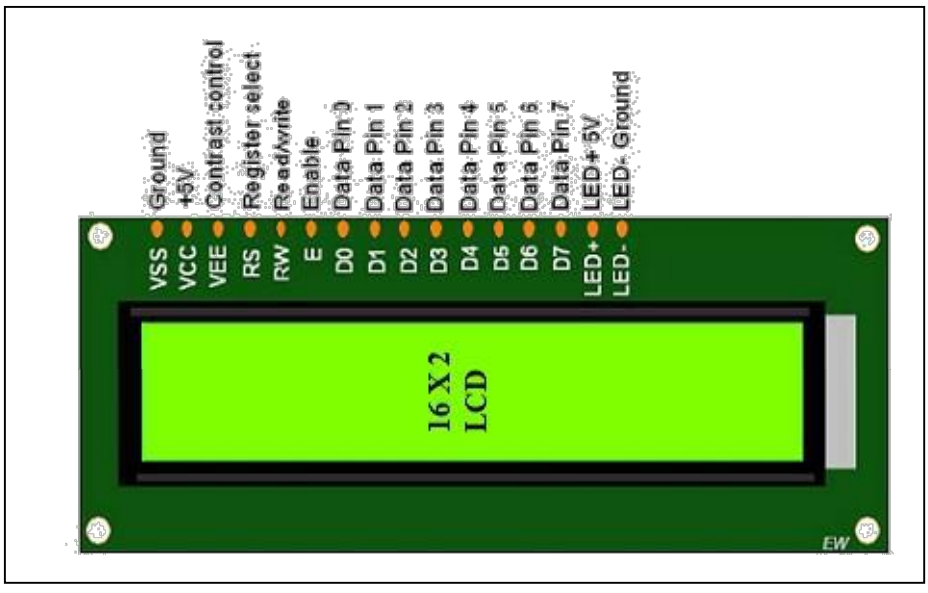


Fig.5.5 LCD Pin Configuration

- Pin1(Ground/Source Pin):This is a GND pin of display,used to connect the GND terminal of the microcontroller unit or power source.

- Pin2(V_{CC}/Source Pin):This is the voltage supply pin of the display,used to connect the supply pin of the power source.

- Pin3(V_0/V_{EE}/Control Pin):This pin regulates the difference of the display,used to connect a changeable POT that can supply 0 to 5V.

- Pin4(Register Select/Control Pin):This pin toggles among command or data register,used to connect a microcontroller unit pin and obtains either 0 or 1(0=data mode,and 1=command mode).

- Pin5(Read/Write/Control Pin):This pin toggles the display among the read or writes operation,and it is connected to a microcontroller unit pin to get either 0 or 1(0=Write Operation,and 1=Read Operation).

- Pin 6(Enable/Control Pin):This pin should be held high to execute Read/Write process,and it is connected to the microcontroller unit&constantly held high.

- Pins 7-14(Data Pins):These pins are used to send data to the display.These pins are connected in two-wire modes like 4-wire mode and 8-wire mode.In 4-wire mode,only four pins are connected to the microcontroller unit like 0 to 3,whereas in 8-wire mode,8-pins are connected to microcontroller unit like 0 to 7.

- Pin15(+ve pin of the LED):This pin is connected to+5V

- Pin 16(-ve pin of the LED):This pin is connected to GND.

5.3.3 Features

- The features of this LCD mainly include the following.
- The operating voltage of this LCD is 4.7V-5.3V.
- It includes two rows where each row can produce 16-characters.

- The utilization of current is 1mA with no backlight.
- Every character can be built with a 5×8 pixel box.
- The alphanumeric LCDs alphabets&numbers.
- Is display can work on two modes like 4-bit&8-bit.
- These are obtainable in Blue&Green Backlight.
- It displays a few custom generated characters.

5.3.4 Registers in LCD

A 16×2 LCD has tworegisterslike data register and command register.The RS(register select)is mainly used to change from one register to another.When the register set is‘0’,then it is known as command register.Similarly,when the register set is‘1’,then it is known as data register.

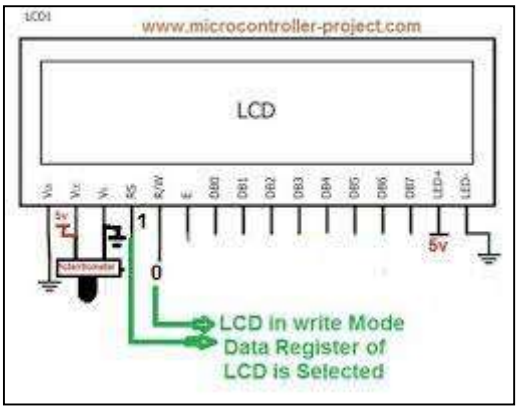


Fig.5.6 Registers

Command Register

The main function of the command register is to store the instructions of command which are given to the display.So that predefined tasks can be performed such as clearing the display,initializing,set the cursor place,and display control.Here commands processing can occur within the register.

Data Register

The main function of the data register is to store the information which is to be exhibited on the LCD screen.Here,the ASCII value of the character is the

information which is to be exhibited on the screen of LCD.Whenever we send the information to LCD,it transmits to the data register,and then the process will be starting there.When register set=1,then the data register will be selected.

5.3.5 LCD Commands

- The commands of LCD 16X2 include the following.
- For Hex Code-01,the LCD command will be the clear LCD screen.
- For Hex Code-02,the LCD command will be returning home.
- For Hex Code-04,the LCD command will be decrement cursor.
- For Hex Code-06,the LCD command will be Increment cursor.
- For Hex Code-05,the LCD command will be Shift display right.
- For Hex Code-07,the LCD command will be Shift display left.
- For Hex Code-08,the LCD command will be Display off,cursor off.
- For Hex Code-0A,the LCD command will be cursor on and display off.
- For Hex Code-0C,the LCD command will be cursor off,display on.
- For Hex Code-0E,the LCD command will be cursor blinking,Display on.
- For Hex Code-0F,the LCD command will be cursor blinking,Display on.
- For Hex Code-10,the LCD command will be Shift cursor position to left.
- For Hex Code-14,the LCD command will be Shift cursor position to the right.
- For Hex Code-18,the LCD command will be Shift the entire display to the left.
- For Hex Code-1C,the LCD command will be Shift the entire display to the right.
- For Hex Code-80,the LCD command will be Force cursor to the beginning(1st line).
- For Hex Code-C0,the LCD command will be Force cursor to the beginning(2nd line).
- For Hex Code-38,the LCD command will be 2 lines and 5×7 matrix.

5.3.6 Applications

- The liquid crystal displays(LCDs)are used in aircraft cockpit displays.
- It is used as a display screen in calculators.
- For displaying images used in digital cameras.
- Mostly the computer monitor is made up of LCDs

5.4 WIFI Module(ESP8266)

5.4.1 ESP8266 History

ESP8266 was designed by the Chinese company Espressif Systems for uses in Internet of Things(IoT)systems.ESP8266 is a complete WiFi system on chip that incorporates a 32-bit processor,some RAM and depending on the vendor between 512KB and 4MB of flash memory.This allows the chip to either function as a wireless adapter that can extend other systems with WiFi functionality,or as a standalone unit that can by itself execute simple applications.Depending on the specific module variant(ESP-1 to ESP-12 at the time of this thesis)between 0 and 7 General Purpose Input/Output(GPIO)pins are available,in addition to Rx and Tx pins of the UART,making the module very suitable for IoT applications.The Software Development Kit(SDK)provided by Espressif contains a lightweight implementation of a TCP/IP control stack(lwIP)for WiFi communication.

The modules houses libraries for optional services such as Dynamic Host Configuration Protocol(DHCP),Domain Name System(DNS),JavaScript Object Notation(JSON)and Secure Socket Layer(SSL)libraries for Application Level programming.It incorporates 802.11 MAC extensions such as 802.11b/g/n/d/e/h/i/k/r that manage signal transmission,encapsulation,encryption,collision management and roaming functionality.The chip generally comes as part of a module,soldered to a Printed Circuit Board(PCB),however it is possible to purchase only the chip itself in order to create a truly custom module.The module variants currently available on the market may include an antenna(PCB or ceramic)or a U-FL connector,a hardware component for serial communication and a myriad of other auxiliary components such as resistors,capacitors and LEDs.

5.4.2 Features

- Low cost,compact and powerful Wi-Fi Module
- Power Supply:+3.3V only
- Current Consumption:100mA
- I/O Voltage:3.6V(max)
- I/O source current:12mA(max)

- Built-in low power 32-bit MCU@80MHz
- 512kB Flash Memory
- Can be used as Station or Access Point or both combined
- Supports Deep sleep(<10uA)
- Supports serial communication hence compatible with many development platform like Arduino.

5.4.3 Pin Configuration

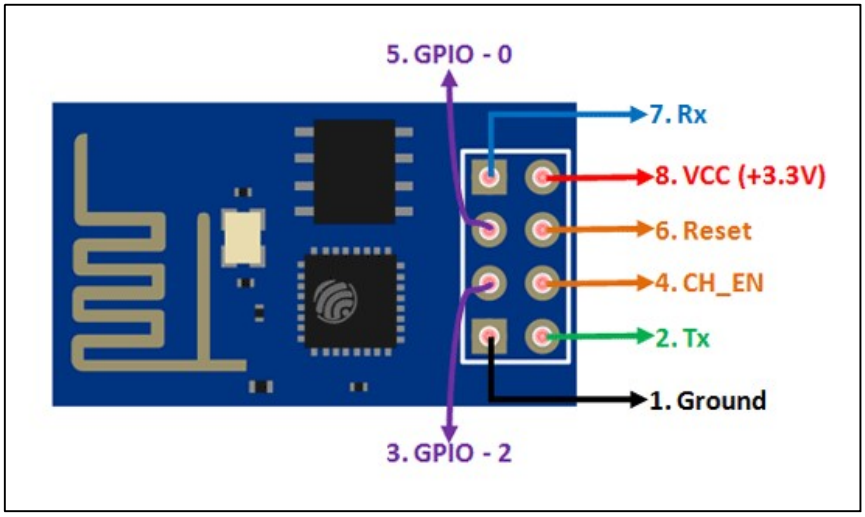


Fig.5.7 ESP8266 Pin Configuration.

Table.1.2 Pin Configuration Of ESP8266

Pin Number	Pin Name	Alternate Name	Normally used for	Alternate purpose
1	**Ground**	-	Connected to the ground of the circuit	-
2	**TX**	GPIO–1	Connected to Rx pin of programmer/uC to upload program	Can act as a General purpose Input/output pin when not used as TX

3	**GPIO-2**	-	General purpose Input/output pin	-
4	**CH_EN**	-	Chip Enable–Active high	-
5	**GPIO-0**	Flash	General purpose Input/output pin	Takes module into serial programming when held low during start up
6	**Reset**	-	Resets the module	-
7	**RX**	GPIO-3	General purpose Input/output pin	Can act as a General purpose Input/output pin when not used as RX
8	$\mathbf{V_{cc}}$	-	Connect to+3.3V only	

The **ESP8266 module** works with 3.3V only,anything more than 3.7V would kill the module hence be cautions with your circuits.The best way to program an **ESP-01** is by using the FTDI board that supports 3.3V programming.If you don't have one it is recommended to buy one or for time being you can also use an Arduino board.One commonly problem that every one faces with ESP-01 is the powering up problem.The module is a bit power hungry while programming and hence you can power it with a 3.3V pin on Arduino or just use a potential divider.So it is important to make a small voltage regulator for 3.31v that could supply a minimum of 500mA.One recommended regulator is theLM317which could handle the job easily.

CHAPTER 6

SOFTWARE IMPLEMENTATION

6.1 Arduino IDE

6.1.1 Introduction

Where IDE stands for Integrated Development Environment–An official software introduced by Arduino.cc,which is mainly used for writing,compiling and uploading the code in the Arduino Device.Almost all Arduino modules are compatible with this software that is open-source and is readily available to install and start compiling the code on the go.In this post,I'll take you through the brief Introduction of the Software,how you can install it,and make it ready for your required Arduino module.Let's dive in and get down to the nitty-gritty of this Software.

Arduino IDE is open-source software that is mainly used for writing and compiling the code into the ArduinoModule.

- It is official Arduino software,making code compilation too easy that even a common person with noprior technical knowledge can get their feet wet with the learning process.
- It is easily available for operating systems like MAC,Windows,Linux and runs on the Java Platformthat comes with inbuilt functions and commands that play a vital role in debugging,editing and compiling the code in the environment.

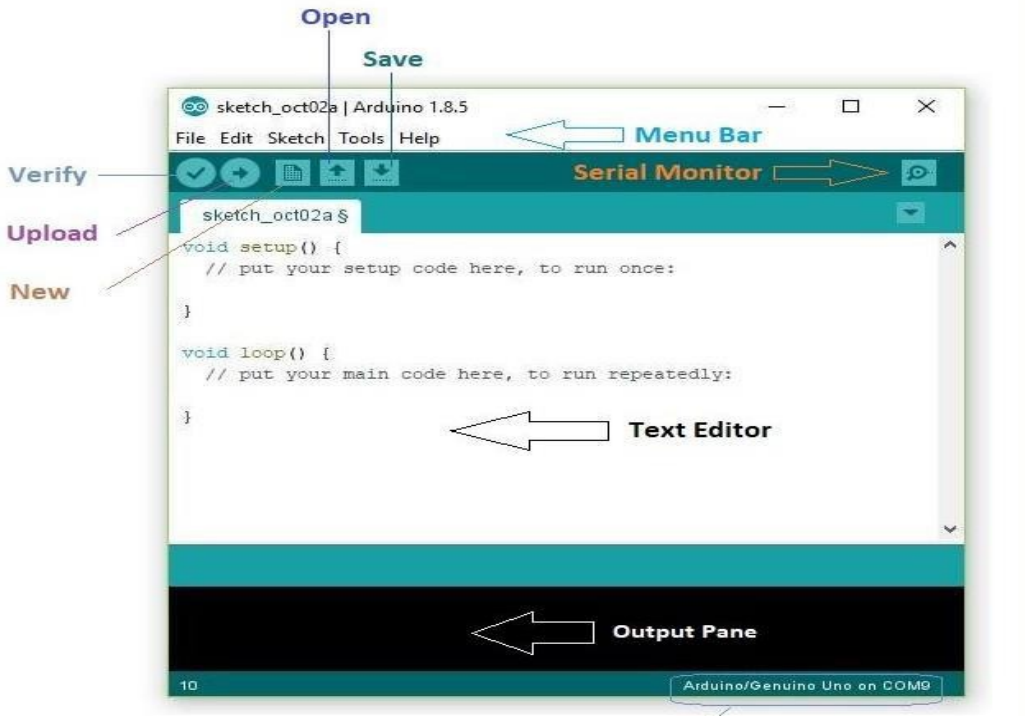


Fig.6.1 Introduction to Arduino IDE

A range of Arduino modules available including Arduino Uno,Arduino Mega,Arduino Leonardo,Arduino Micro and many more.

- Each of them contains a MicroController on the board that is actually programmed and accepts the information in the form of code.
- The main code,also known as a sketch,created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
- The IDE environment mainly contains two basic parts:Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given ArduinoModule.
- This Environment Supports Both C and C++.

6.1.2 How to Download Arduino IDE:

You can download the Software from Arduino's main website.As I said earlier,the software is available for common operating systems like Linux,Windows,and MAX,so make sure you are downloading the correct software version that is easily compatible with your operating system.If you aim to download Windows app version,make sure you have Windows 8.1 or Windows 10,as the app version is not compatiblewith Windows 7 or older version of this operating system.

The IDE environment is mainly distributed into three sections

- **Menu Bar**
- **Text Editor**
- **Output Pane**

As you download and open the IDE software,it will appear like an image below.

The bar appearing on the top is called **Menu Bar** that comes with five different options as follow

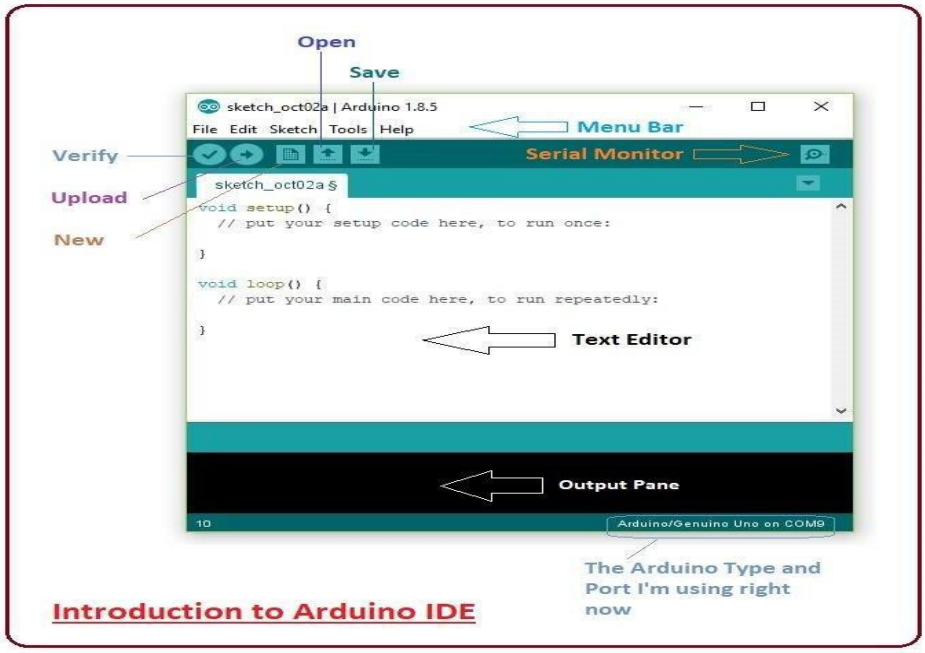


Fig.6.2 IDE Environment

File–You can open a new window for writing the code or open an existing one.The

following table shows the number of further subdivisions the file option is categorized into.

Table.1.3 File Options

Option	Description
New	This is used to open new text editor window to write your code
Open	Used for opening the existing written code
Open Recent	The option reserved for opening the recently closed program
Sketchbook	It stores the list of codes you have written for your project
Examples	Default examples already stored in the IDE software
Close	Used for closing the main screen window of the recent tab.If two tabs areopen,it will ask you again as you aim to close the second tab
Save	It is used for saving the recent program
Save as	It will allow you to save the recent program in your device folder
Page setup	Page setup is used for modifying the page with portrait and landscape options.Some default page options are already given from which you canselect the page you intend to work on
Print	It is used for printing purpose and will send the command to the printer
Preferences	It is page with number of preferences you aim to setup for your text editorpage
Quit	It will quit the whole software all at once

At the end of compilation,it will show you the hex file it has generated for the recent sketch that willsend to the Arduino Board for the specific task you aim to achieve.

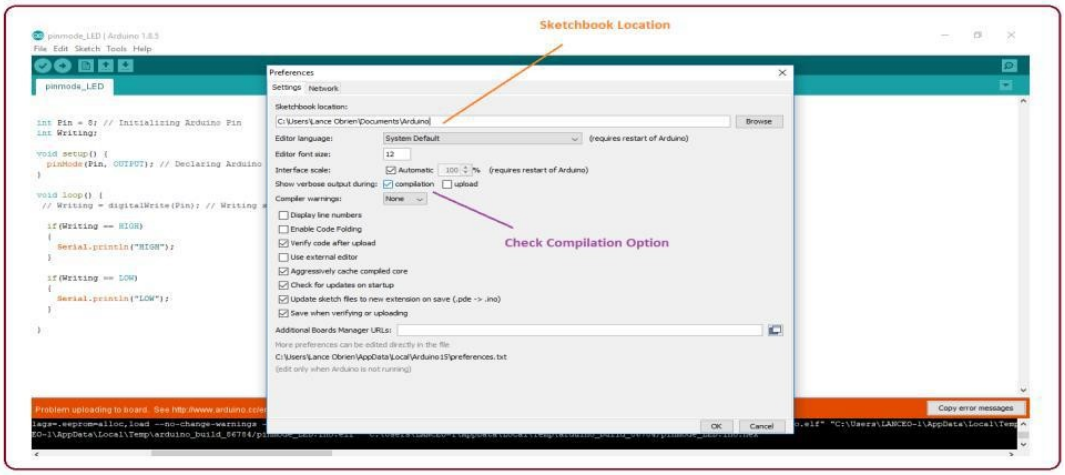


Fig.6.3 Code Compilation

At the end of compilation,it will show you the hex file it has generated for the recent sketch that willsend to the Arduino Board for the specific task you aim to achieve.

- **Edit**–Used for copying and pasting the code with further modification for font.
- **Sketch**–For compiling and programming
- **Tools**–Mainly used for testing projects.The Programmer section in this panel is used for
- burning a bootloader to the new MicroController.
- **Help**–In case you are feeling sceptical about software,complete help is available from
- getting started to troubleshooting.

The **Six Buttons** appearing under the Menu tab are connected with the running as Followsprogram as follows:

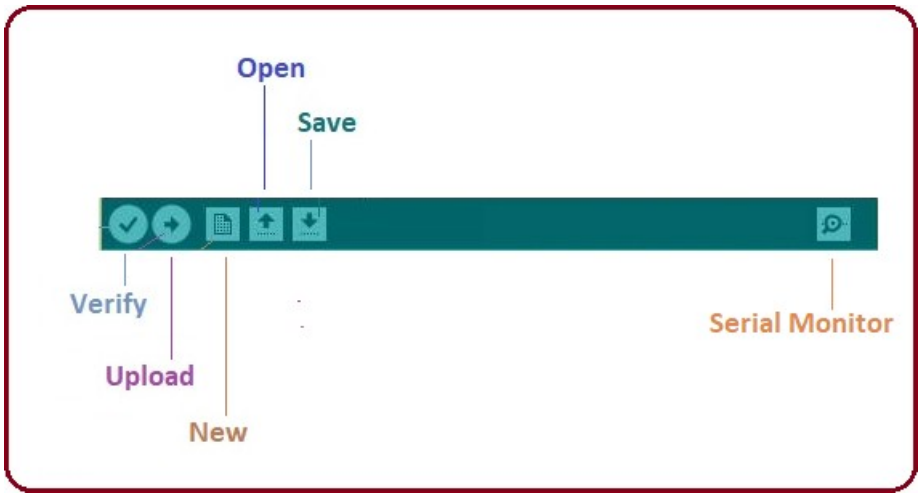


Fig.6.4 Menu Tab

- The dotted paper is used for creating a new file.
- The upward arrow is reserved for opening an existing Arduino project.
- The downward arrow is used to save the current running code.
- The button appearing on the top right corner is a **Serial Monitor**–A separate pop-up window that actsas an independent terminal and plays a vital role in sending and receiving the Serial Data.You can also go to the Tools panel and select Serial Monitor,or pressing Ctrl+Shift+M all at once will open it instantly.
- The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating.Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.
-

 You need to select the baud rate of the Arduino Board you are using right now.For my Arduino Uno,Baud Rate is 9600,as you write the following code and click the Serial Monitor,the output will showas the image below.

The main screen below the Menu bard is known as a simple text editor used for writing the required code.

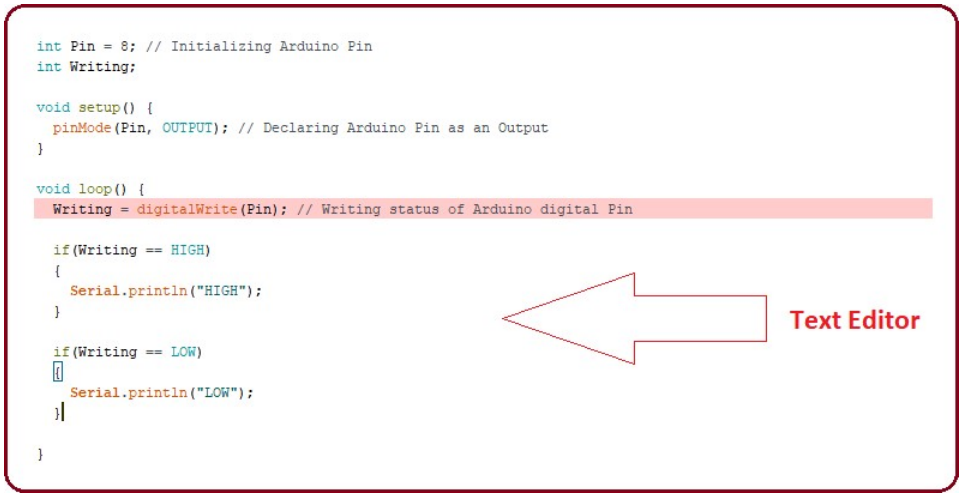


Fig.6.5 Text Editor

The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code:the memory used by the code,and errors occurred in the program.You need to fix those errors before you intend to upload the hex file into your Arduino Module.

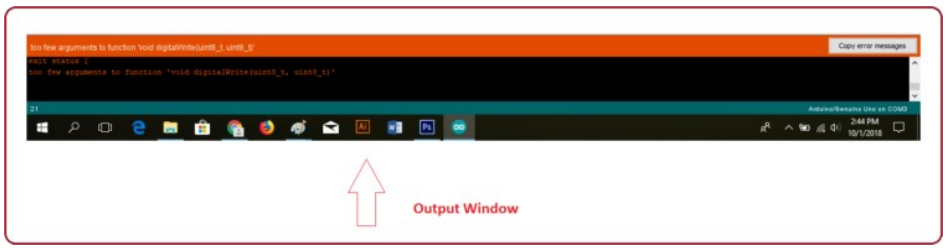


Fig.6.6 Output Window

More or less,Arduino C language works similarly to the regular C language used for any embedded system MicroController,however,there are some dedicated libraries used for calling and executing specific functions on the board.

6.1.3 Libraries

Libraries are very useful for adding extra functionality to the Arduino Module.There is a list oflibraries you can add by clicking the Sketch button in the menu bar and going to Include Library.As you click the Include Library and Add the respective library it will on the top of the sketch with a#include sign.Suppose,I Include the EEPROM library,it will appear on the text editor as#include<EEPROM.h>.

Most of the libraries are preinstalled and come with the Arduino software.However,you can also download them from external sources.

Making Pins Input or Outputs

The digital Read and digital Write commands are used for addressing and making the Arduino pins as input and output respectively.These commands are text sensitive you need to write them down the exact waythey are given like digital Write starting with small“d”and write with capital“W”.Writing it down with Digital write or digital write won’t be calling or addressing any function.

6.1.4 How to Select the Board

In order to upload the sketch,you need to select the relevant board you are using and the ports for that operating system.As you click the Tools on the menu,it will open like the figure below.

Just go to the“Board”section and select the board you aim to work on.Similarly,COM1,COM2,COM4,COM5,COM7 or higher are reserved for the serial and USB board.You can look for the USB serial device in the ports section of the Windows Device Manager.

After correct selection of both Board and Serial Port,click the verify and then upload button appearingin the upper left corner of the six-button section or you can go to the Sketch section and press verify/compileand then upload.

- The sketch is written in the text editor and is then saved with the file extension.Ion.It is important tonote that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software,however,the older version may require the physical reset on the board.

- Once you upload the code,TX and RX LEDs will blink on the board,indicating the desired program is running successfully.
- The amazing thing about this software is that no prior arrangement or bulk of the mess is required to install this software,you will be writing your first program within 2 minutes after the installation of the IDE environment.

6.1.5 Bootloader

As you go to the Tools section,you will find a bootloader at the end.It is very helpful to burn the code directly into the controller,setting you free from buying the external burner to burn the required code.

When you buy the new Arduino Module,the bootloader is already installed inside the controller.However,if you intend to buy a controller and put in the Arduino module,you need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader.

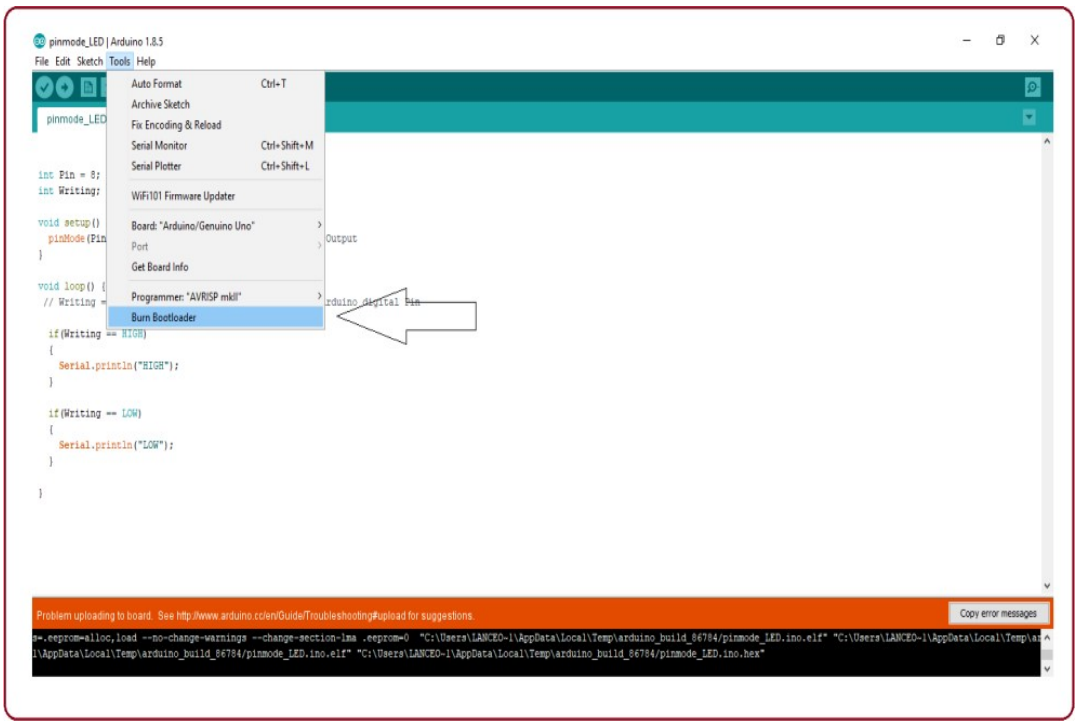


Fig.6.7 Arduino Bootloader

6.2 Thingspeak Configuration

we need to create the thingspeak channel and get the key

Step 1

Go to[https://thingspeak.com/,](https://thingspeak.com/)register an account and login to the platform

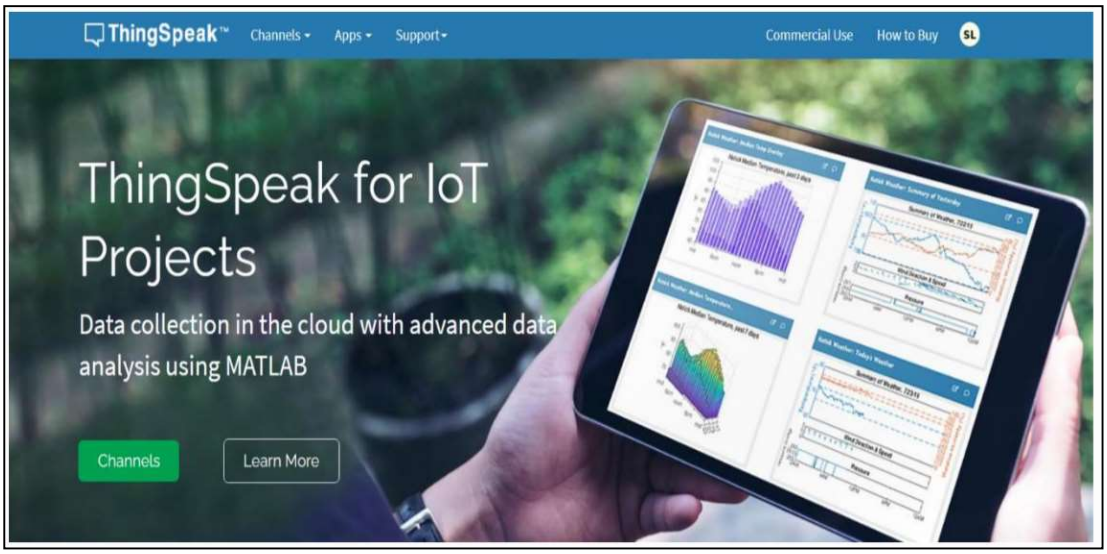


Fig.no 6.8 Home page

Step 2

Choose Channels->My Channels->New Channel

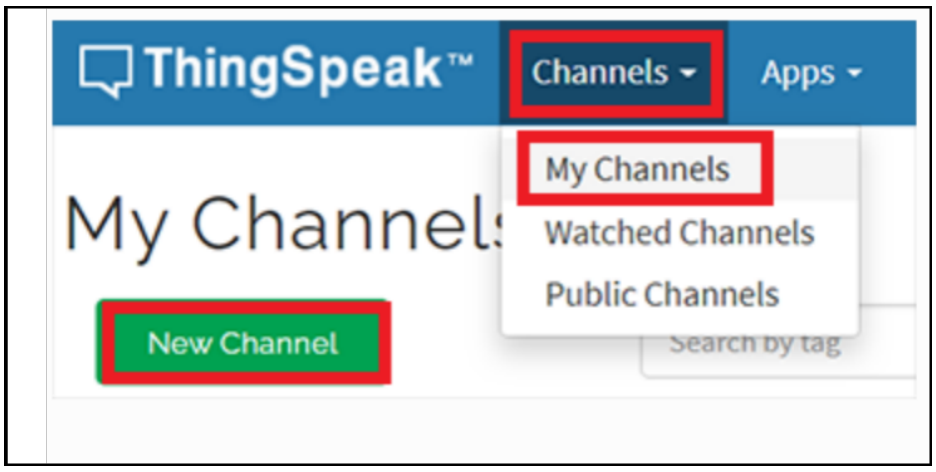


Fig.no 6.9 Choose channel

Step 3
Input Channel name,Field1,then click“Save Channel

- Channel name:smart-house
  Field 1:Temperature

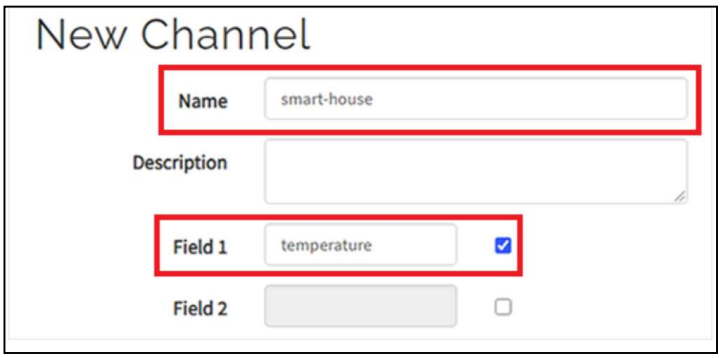


Fig.no 6.10 Create new channel

Step 4

You will see a chat for data field1

Fig.no 6.11 Data Field Char

Step 5

Open your web browser,go to https://thingspeak.com,select your channel>"API Keys", copy the API key as follows:

Fig.no 6.12 API keys page

6.3 MIT APP INVENTOR

The smartphone is an information nexus in today's digital age,with access to a nearly infinite supply of content on the web,coupled with rich sensors and personal data.However,people have difficulty harnessing the full power of these ubiquitous devices for themselves and their communities.Most smartphone users consume technology without being able to produce it,even though local problems can often be solved with mobile devices.How then might they learn to leverage smartphone capabilities to solve real-world,everyday problems?MIT App Inventor is designed to democratize this technology and is used as a tool for learning computational thinking in a variety of educational contexts,teaching people to build apps to solve problems in their communities.MIT App Inventor is an online development platform that anyone can leverage to solve real-world problems.

It provides a web-based"What you see is what you get"(WYSIWYG)editor for building mobile phone applications targeting the Android and iOS operating systems.It uses a block-based programming language built on Google Blockly(Fraser,2013)and inspired by languages such as StarLogo TNG(Begel&Klopfer,2007)andScratch(Resnicketal.,2009;Maloney,Resnick,Rusk, Silverman,&Eastmond,2010),empowering anyone to build a mobile phone app to meet a need.

To date,6.8 million people in over 190 countries have used App Inventor to build over 24 million apps.We offer the interface in more than a dozen languages.The platform has also been adapted to serve requirements of more specific populations,such as building apps for emergency/first responders(Jain et al.,2015)and robotics(Papadakis&Orfanakis,2016).In this chapter,we describe the goals of MIT App Inventor and how they have influenced our design and development—from the program's inception at Google in 2008,through the migration to MIT,to the present day.We discuss the pedagogical value of MIT App Inventor and its use as a tool to teach and encourage people of all ages to think and act computationally.

We also describe three applications developed by students in different parts of the world to solve real issues in their communities.We conclude by discussing the limitations and benefits of tools such as App Inventor and proposing new directions for research.

6.3.1 MIT App Inventor Overview

The MIT App Inventor user interface includes two main editors:the design editor and the blocks editor.The design editor,or designer,is a drag and drop interface to lay outthe elements of the application's user interface(UI).The blocks editor(see Fig.3.2)is an environment in which app inventors can visually lay out the logic of their apps using color-coded blocks that snap together like puzzle pieces to describe the program.

To aid in development and testing,App Inventor provides a mobile app called the App Inventor Companion(or just"the Companion")that developers can use to test and adjust the behavior of their apps in real time.In this way,anyone can quickly build a mobile app and immediately begin to iterate and test MIT App Inventor.In the design of MIT App Inventor,introducing mobile app development in educational contexts was a central goal.

Prior to its release,most development environments for mobile applications were clunky,only accessible with expertise in systems level or embedded programming,or both.Even with Google's Android operating system and the Java programming language,designing the user interface was a complex task.Further,use of the platform required familiarity with Java syntax and semantics,and the ability to debug Java compilation errors(e.g.,misspelled variables or misplaced semicolons)for success.

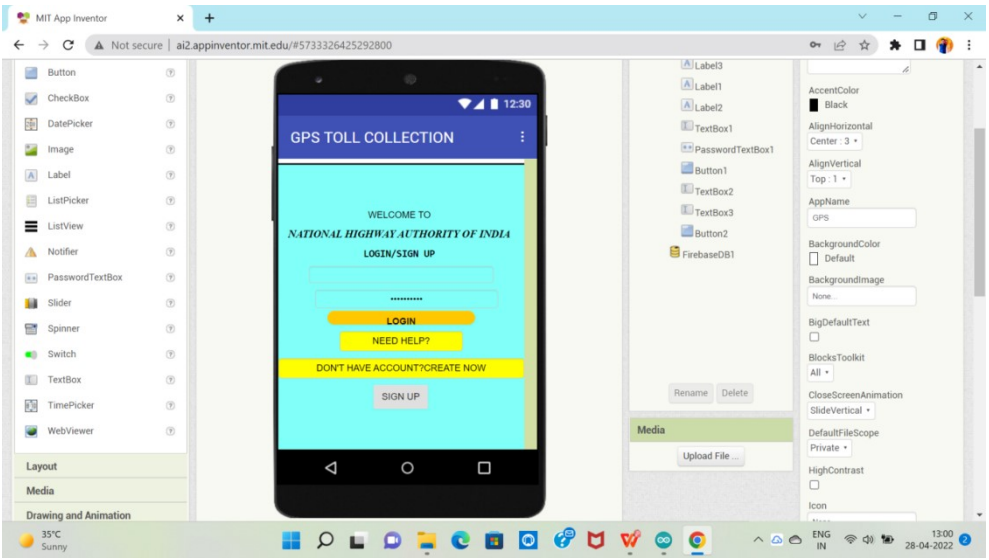


Fig.no 6.13 Home page of MIT APP Inventor

These challenges presented barriers to entry for individuals not versed in computer science,App Inventor's target demographic.We briefly highlight and discuss design goals for the App Inventor project,specifically,the use of components to abstract some of the complexity of platform behavior,and the use of blocks to eliminate complexity of the underlying programming language.These goals can be further explained as aligning the visual language to the mental models of young developers and enabling exploration through fast,iterative design.

6.4 Source Code

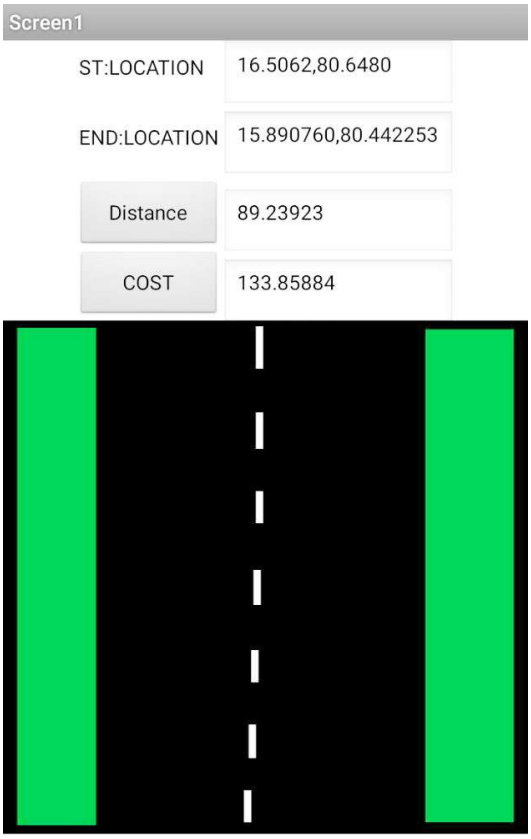
```
#include<SoftwareSerial.h>
#include<LiquidCrystal.h>
#include<TinyGPS.h>
#include<Wire.h>

TinyGPS gps;
float flat=0,flon=0;
const int rs=2,en=3,d4=4,d5=5,d6=6,d7=7;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);
SoftwareSerial mySerial(9,8);

void read_gps()
{
bool newData=false;
unsigned long chars;
unsigned short sentences,failed;
for(unsigned long start=millis();millis()-start<1000;)
{
while(Serial.available())
{
char c=Serial.read();
if(gps.encode(c))
newData=true;
}
}

if(newData)
{

unsigned long age;
gps.f_get_position(&flat,&flon,&age);
```

```
}
}
void setup(){
Serial.begin(9600);
mySerial.begin(115200);
lcd.begin(16,2);
lcd.print("WELCOME");
wifi_init();
}

void loop(){
read_gps();
lcd.clear();
lcd.print("LT:"+String(flat,6));
lcd.setCursor(0,1);
lcd.print("LG:"+String(flon,6));

upload_iot();
delay(5000);
}

void wifi_init()
{
mySerial.println("AT+RST");
delay(2000);
mySerial.println("AT+CWMODE=1");
delay(1000);
mySerial.print("AT+CWJAP=");
mySerial.write("");
mySerial.print("SRC 24G");//ssid/user name
mySerial.write("");
mySerial.write(',');
mySerial.write("");
```

```
mySerial.print("src@internet");//password
mySerial.write("");
mySerial.println();
delay(1000);
}

void upload_iot()//ldr copied int to-x and gas copied into-y
{

String cmd="AT+CIPSTART=\"TCP\",\"";
cmd+="184.106.153.149";//api.thingspeak.com
cmd+="\",80";
mySerial.println(cmd);
delay(1500);

String getStr="GET/update?api_key=E976ZYBL2SGHTHBD&field1=";
getStr+=String(flat,6);
getStr+="&field2=";
getStr+=String(flon,6);
getStr+="\r\n\r\n";
cmd="AT+CIPSEND=";
cmd+=String(getStr.length());
mySerial.println(cmd);
delay(1500);
mySerial.println(getStr);
delay(1500);
```

CHAPTER 7
RESULTS

Device Result



Application Result

We divided our project into two parts. In the first part with the help of Arduino Atmega 328p and by using GPS module we have obtained the results of longitude and latitude values of the location. In the second part with the help of an application we acquired the results from the hardware from the Thinkspeak Cloud Platform and later based on the distance covered by the vehical we calculated the cost and displayed it on the application.

CONCLUSION

The project for future electronic toll collect systems and have different attributes pros and cons.For many years DSRC systems have been preferred,due to their simplicity of operation,need road side equipment typically mounted on a gantry,with electronic tags in the vehicles which may be read only,read write or smart card based.The proposed system based on a combination of IoT and a satellite based global positioning system(GPS).An innovative log on unit OBU,which automatically calculate the amount of charge due and take in to account,depends on the type of the vehicle.It will also act like a platform for vehicle identification and prove effective in tracking stolen vehicles.With regard to future expansion and development,the satellite based toll collection system will be a better solution,especially with regard to flexibility when it comes to extending toll collection to every road category and in terms of cost efficiency in an implementation operation.

FUTURE SCOPE

The Technology developed day by day with intelligent System and Internet of Things.This project will help to easy to collect the toll from the vehicles with out using manpower.In future all are digital world every thing will be connected to the internet and operated wireless using smart phone applications.we can reduce the time to collect toll and reduce the traffic during festival seasons and also safe&secure payment also included future development projects.

REFERENCES

[1] Gabriel Nowacki,Izabella Mitraszewska,Tomasz KamiÑski,The National Automatic Toll Collection System For The Republic Of Poland,Transport And Telecommunication,2008,Volume 9,No 2,24–38.

[2] Ms.Rajkuwar patil,Ms.Kajal Thakur,Ms.Shruthi Kharat,Mr.V.V.Shetkar published a paper“GPS Based Toll Collection System”,International journal for scientific research&development.

[3]Dr.Khali Persad,Dr.C.Michael Walton,Shahrivar Hussain,Toll Collection Technology and Best Practices,Project 0Ğ5217:Vehicle/License Plate Identification For Toll Collection Applications Aug 2006;Revised Jan 2007.

[4]Automated Toll Collection Using Satellite Navigation Ms.Kirti A.Lonkar , Ms. Pratibha P. Kulkarni , Ms Monalisha Dash , Mr.Abhishek Dhawan , Mr.Hemant R.Kumbhar , Mr.Monika P.Gagtap.International Journal of Computational Engineering Research.

[5]Development of GPS Based Highway Toll Collection System.Jin Yeong Tan,pin Jern Ker,Dineis Mani,Puvanesan Arumugam.IEEE Conference on Control System,Computing and Engineering.

[6]Supriya Jagtap ,Shweta Ghatage ,Supriya Koli ,Neelam Yadav.RFID Based Automated Toll Collection System.International Journal Of Advanced Research in Science and Engineering.

[7]Suryanti Awang and Nik Mohamad Aizuddin Nik Azmi.Automated Toll Collection System based on Vehicle Type Classification using Sparse-Filtered Convolutional Neural Networks with Layer-Skipping Strategy

[8]B. Senapati, P. M. Khilar, Naba Krushna Sabat.An Automated Toll Gate SystemUsing VANET.IEEE First International Conference on Energy,System and Information Processing.

[9] Ganesh K. Andurkar, Vidya R. Ramteke, Smart Highway Electronic Toll Collection System.International Journal of Innovation Research in Computer and Communication Research.

[10] V. Aruna Kumari1, B. V. Brindashree , Chaitanya Tejas , D. Mithun Kumar , N. P. Hemanth Kumar , J. C. Arpitha.An Intelligent,Automated Toll Payment System.International Journal Of Research in Engineering and Management.

[11] Sudheer Kumar Nagothu, Om prakash kumar, G Anitha, "Autonomous monitoring and attendance system using inertial navigation system and GPRS in predefined location",2014 3rd International Conference on Eco-friendly Computing and Communication Systems (ICECCS), Year: 2014,Pages: 261 - 265, DOI: 10.1109/Eco-friendly.2014.60.

[12] Himanshu Bhatt, Bill Glover “RFID Essentials” publisher: O’Reilly. January 2006.

[13] Frank Thornton, Brad Haines & Anand M. Das “RFID Security” Syngress publishing, Canada, April,2006.

[14] Sanchit Agarwal, Shachi Gupta, Nidheesh Sharma” Electronic Toll Collection System Using Barcode Laser Technology”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 2, March – April 2014 pages 180-182.