VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590 018

Project Report on

IoT BASED UNMANNED GROUND VEHICLE

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

BACHELOR OF ENGINEERING

in

ELECTRICAL AND ELECTRONICS ENGINEERING

Submitted by

AKARAPU NIKITHA	**1MV15EE004**
KOOSURU VENKATESH	**1MV15EE045**
RAGALA CHETHAN	**1MV15EE073**
RANJITH V	**1MV15EE076**

Under the guidance of

Mrs Rekha Radhakrishnan

Assistant Professor

Dept. of Electrical and Electronics Engg,

Sir MVIT

Department of Electrical and Electronics Engineering

SIR M VISVESVARAYA INSTITUTE OF TECHNOLOGY

(Approved by AICTE New Delhi, Affiliated to VTU, Belagavi, ISO 9001:2008 Certified)

Krishnadevarayanagar, Airport road, Bengaluru-562 157

2018-2019

SIR M VISVESVARAYA INSTITUTE OF TECHNOLOGY

(Approved by AICTE New Delhi, Affiliated to VTU, Belagavi, ISO 9001:2008 Certified)

Krishnadevarayanagar, Airport road, Bengaluru-562 157

Department of Electrical and Electronics Engineering

CERTIFICATE

Certified that the project entitled **"IoT BASED UNMANNED GROUND VEHICLE"** has been carried out by **AKARAPU NIKITHA (1MV15EE004), KOOSURU VENKATESH (1MV15EE045), RAGALA CHETHAN (1MV15EE073), RANJITH V (1MV15EE076),** bonafide students of **Sir M Visvesvaraya Institute of Technology, Bengaluru,** in partial fulfillment for the requirements for the award of the degree of **Bachelor of Engineering** in **Electrical & Electronics Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year **2018-2019.** It is certified that all corrections/suggestions indicated have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the above-mentioned degree.

Signature of Guide	Signature of HoD	Signature of Principal
Mrs Rekha Radhakrishnan	Dr H L Suresh	Dr V R Manjunath
Assistant Professor	Professor & HoD	Principal
Dept. of EEE, Sir MVIT	Dept. of EEE, Sir MVIT	Sir MVIT

EXTERNAL VIVA

Name of the Examiner	Signature with date
1.	
2.	

SIR M VISVESVARAYA INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi, Affiliated to VTU, Belagavi, ISO 9001:2008 Certified)
Krishnadevarayanagar, Airport road, Bengaluru-562 157

Department of Electrical and Electronics Engineering

DECLARATION

I hereby declare that the project entitled **"IoT BASED UNMANNED GROUND VEHICLE"** has been carried out by me and submitted in partial fulfilment for the requirements for the award of the degree of **Bachelor of Engineering** in **Electrical & Electronics Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year **2018-2019**. The matter embodied in this project report has not been submitted to any other university or institute for the award of any other degree or diploma.

Place: Bengaluru
Date: 14/06/2019

Ragala Chethan
USN: 1MV15EE073

ACKNOWLEDGEMENT

It gives me immense pleasure to express my sincere gratitude to the management of **Sir M Visvesvaraya Institute of Technology**, Bengaluru for providing the opportunity and resources to accomplish my project in their premises.

On the path of learning, the presence of an experienced guide is indispensable and I would like to thank my guide, **Mrs Rekha Radhakrishnan**, Assistant Professor, Dept. of Electrical and Electronics Engineering, for her invaluable help, constant support and guidance.

I would like to convey my regards and sincere thanks to **Dr H L Suresh**, Head of Department, Dept. of Electrical and Electronics Engineering, for his encouragement and suggestions.

Heartfelt and sincere thanks to **Dr V R Manjunath**, Principal, Sir M Visvesvaraya Institute of Technology, for providing the infrastructure and facilities required to develop my project.

I would also like to thank the staff of Department of Electrical and Electronics Engineering and lab-in-charges for their cooperation and suggestions. Last but not the least, I would like to thank my family who helped me to reach my goals without the slightest discomfort and all my friends who helped me for the completion of project successfully.

ABSTRACT

Unmanned Ground Vehicles (UGV) are robotic platforms that are used as an extension of human capability. This type of robots are generally capable of operating outdoors and over a wide variety of terrain, functioning in place of human. The aim of our project is to design and implement a prototype of an unmanned ground vehicle which can detect the obstacles in the path of challenging environments. The vehicle comprises of ultrasonic sensors to detect the obstacle blocking the path and GPS to know the exact location of a particular obstacle. The UGV collects information of the surroundings using gas and temperature sensors which are mounted on the vehicle. This information collected can be sent to the authorities concerned through IoT, for further action. This project thus focuses on developing a semiautonomous vehicle which can be used in war field and during natural calamities to assess the acts as a first hand informer regarding the status of situation without endangering the life of rescue personnel.

Keywords:

Unmanned Ground Vehicle, Obstacle Detection, GPS, Internet of Things

CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

An Unmanned Ground Vehicle (UGV) is a vehicle which operates on ground without an onboard human presence. UGV can be used for many applications where it can be inconvenient, dangerous, or impossible for human presence. Usually, the vehicle has a set of sensors to get a first-hand perception about the surroundings. UGV can be autonomous in decision making or remotely operated from a different location by a human operator.

Based on application, UGV will have a platform, sensors, control system, communication link, and system integration features. The platform can be an all-terrain vehicle with independent power source. The purpose of UGV sensors is navigation and detection of surroundings. UGVs are usually remote operated and autonomous, although human control is necessary in certain situations. Communication between UGV and remote control can be done using radio-frequency, microwave, or Wi-Fi. System integration includes hardware and software interplay.

They are many applications of UGV in the present world. Mostly, UGVs are used to replace humans in hazardous situations. Military applications include surveillance, reconnaissance, and target identification. UGVs can face the first fire from insurgents, thereby reducing the casualties. Further, UGV can be used in rescue and recovery missions.

A reliable wireless connection between the operator and UGV is critical in many Urban Search and Rescue (USAR) missions. During a mission, the operator needs to keep track of not only the physical parts of the mission, such as navigating through an area or searching for victims, but also the variations in network connectivity across the environment.

The Internet of Things (IoT) is the extension of internet connectivity into physical devices and everyday objects. Embedded with electronics, internet connectivity, and other forms of hardware such as sensors, these devices can communicate and interact with others over the internet, and they can be remotely monitored and controlled. The definition of the internet of things has evolved due to the convergence of multiple

technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation, and others all contribute to enabling the Internet of things.
The IoT concept has faced prominent criticism, especially in regards to privacy and security concerns related to these devices and their intention of pervasive presence.

1.1 Objectives

- Design and implement a prototype of an unmanned ground vehicle which can move in desired direction in all terrains.
- The UGV should detect obstacle in its path, surrounding temperature, and presence of poisonous gases. Also, GPS should feed the location of obstacle.
- The information is to be sent to end-user through IoT for further action.

1.2 Motivation

Human safety has been a topic of concern since past few years. We have so many challenging environments around, where we cannot send the human directly without prior knowledge of the surroundings. This will lead to loss of human life in many cases. In order to avoid such incidents, we need to get information about the surroundings before sending a human.

There are many environments which are dangerous for human presence. These jobs are ideal to be replaced by the Unmanned Ground Vehicles. The areas may be closed or open environments. There are many enclosed areas like mining areas, large industries. There is a need to get the prior information about the particular enclosed area before the human intervention.

We have seen many incidents in mining areas where emission of poisonous gases cause lung diseases, which may even lead to death of the person. The other case where there are furnaces in the large industries where we cannot send a person directly without knowing the information about the temperature inside the furnace.

Even in some places we need to detect the obstacle ahead the path and send the information about the location where exactly the obstacle is detected using the GPS. Our aim is to design a prototype of unmanned ground vehicle which is a multipurpose vehicle has extended application to ensure the safety of a person.

1.3 Literature Survey

Johann Borenstein and Yoram Koren [1] developed a mobile robot system, capable of performing various tasks for the physically disabled. The obstacle avoidance strategy in the mobile robot uses ultrasonic range finders for detection and mapping. This paper explains the methodology of ultrasonic detection.

Peng Zhang, Bo Huang and Takaaki Baba [2] designed and implemented an ultrasonic obstacle detecting system, which will be fixed on the vehicle. An algorithm is also proposed to calculate the collision time. This paper explains the working of ultrasonic sensors for obstacle detection.

Vivek Agarwal, N Venkata Murali and C Chandramouli [3] described an accurate and fast driver assistant system that detects the obstacles and warns the driver. Ultrasonic sensors are used to detect obstacles because they several advantages over other sensors. The proposed system gives an insight about the interface of sensors with microcontroller.

M. Z. H. Noor, S. A. S. M. Zain and L. Mazalan [4] designed and developed a remote operated multi-directional Unmanned Ground Vehicle (UGV). The movement of multi-directional UGV is achieved by using Mecanum wheels with differential drive configuration. This paper gives idea about the movement of the UGV in various directions such as forward, backward, left and right using differential drive.

Mohammad Ariful Islam, Wahidul Ahad, Mohammad Faisal, and Hasan U Zaman [5] proposed the design of a robot to detect obstacles in the path ahead. The proposed design is cost-effective in terms of performance and battery back-up.

R. Girisrinivaas and V. Parthipan [6] proposed a system which monitors water level and gas level in sewage system and the measured values will be stored in cloud storage. The data is sent to nearby corporation office using GSM module. This paper gives an insight to Internet of Things (IoT).

Sourabh Sarkar, Srijita Gayen and Saurabh Bilgaiyan [7] designed and implemented Android Based Home Security Systems using Internet of Things (IoT) and Firebase. The design and implementation of wireless home automation control using Wi-Fi technology handles the devices with integral security. The proposed system gives an insight about the technology of wireless transmission of data from NodeMCU to Firebase and then, to the Android application of authenticated user.

Chapter 2

Project Description

2.1 Block Diagram

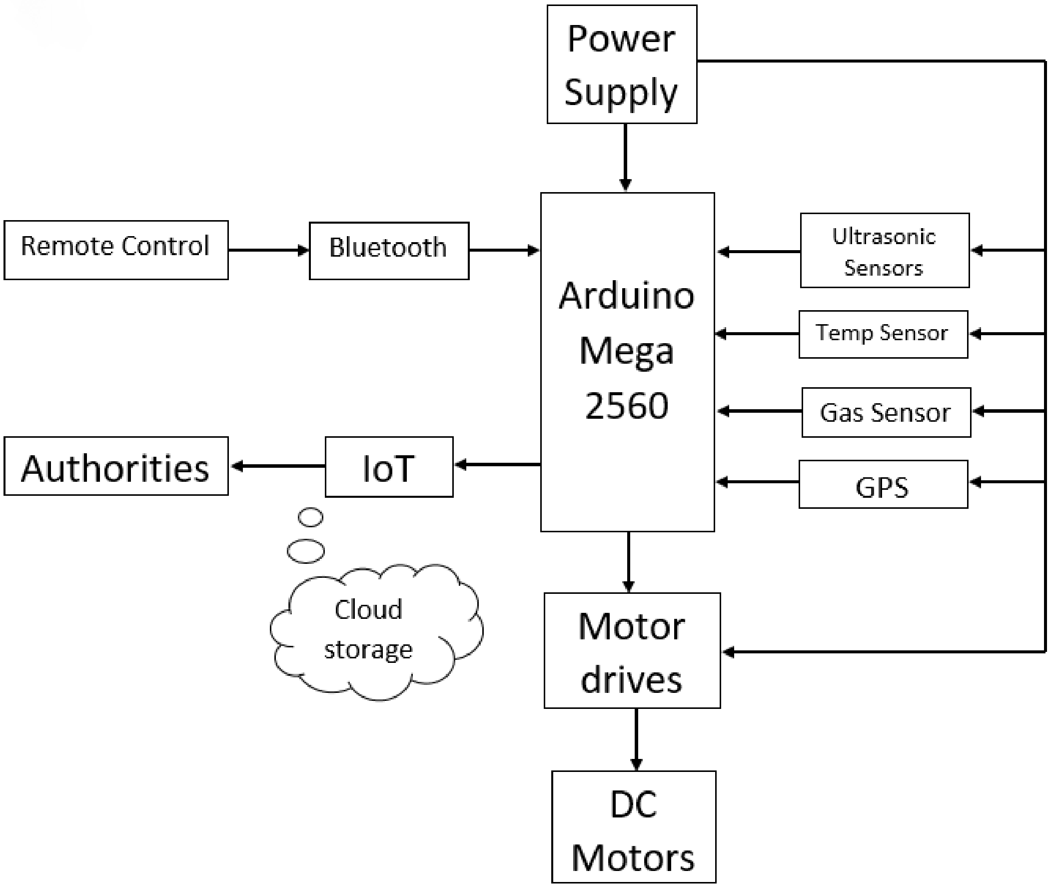


Fig 2.1: Block Diagram

The block diagram of IoT based unmanned ground vehicle is shown in Fig 2.1. The vehicle consists of a microcontroller, various sensors, motors and motor drivers, and power supply. The vehicle is controlled by a remote controller over Bluetooth channel. The information is sent to authorities using IoT.

2.2 Explanation

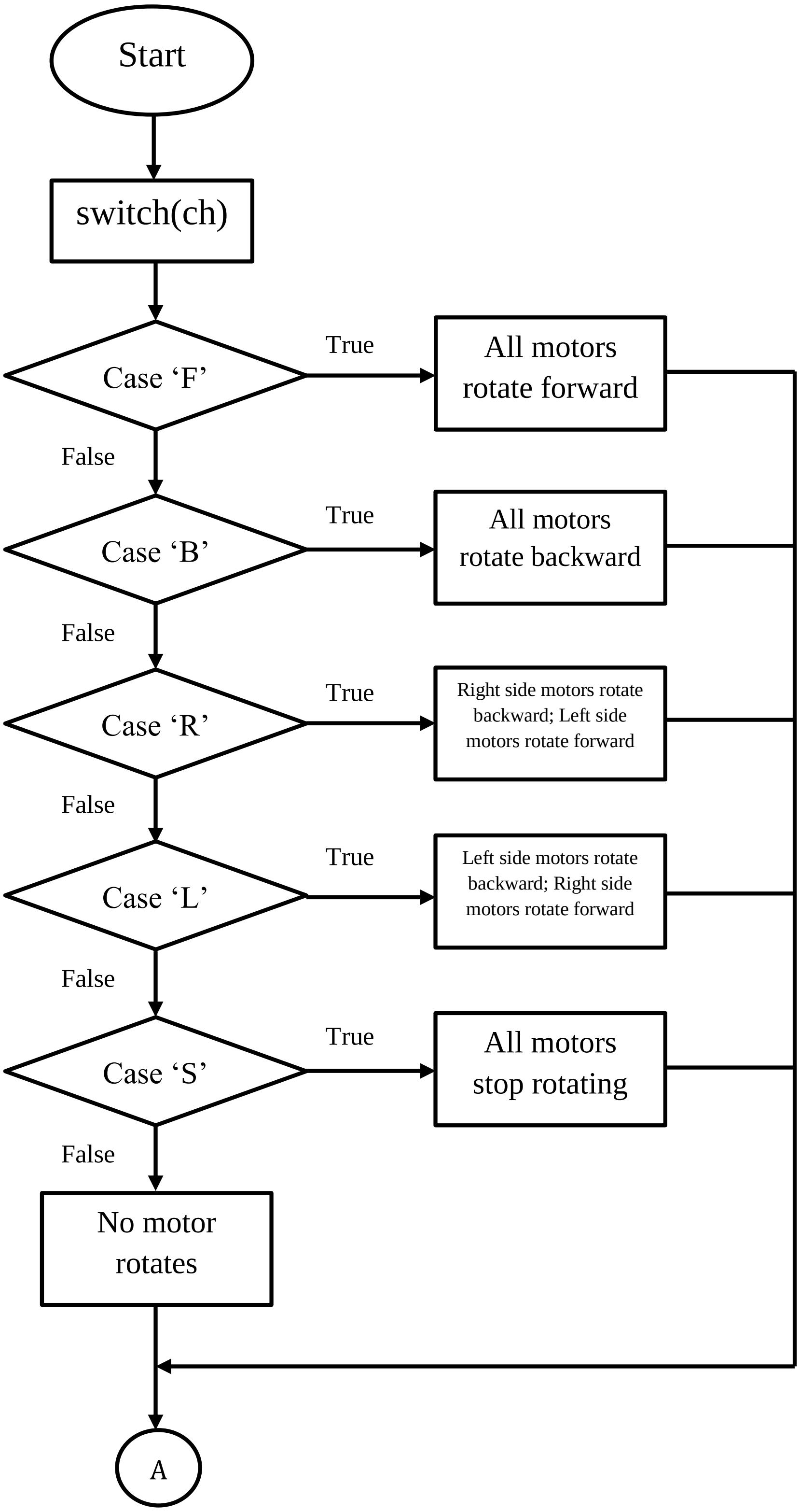
The vehicle is controlled over Bluetooth channel by a remote controller, who is in the vicinity of the vehicle. The vehicle has four dc motors driven by two drivers. The four wheeled drive helps to achieve the desired direction easily and better drive in rough terrains. The integration of 4 motors provides more overall torque. The four wheeled drive has more advantages when compared to the two wheeled drive. The remote control can be achieved by an android application. The android application is developed using the MIT App Inventor 2.

The presence of obstacle is detected by an ultrasonic sensor. Ultrasonic sensor continuously sense the path ahead. When an obstacle is detected, the vehicle moves back for few seconds and stops. Then, all the sensors give the output. The temperature sensor gives the temperature of the surroundings. The gas sensor which is interfaced, will detect any poisonous gas that harms the human. The poisonous gases may be Carbon-monoxide (CO), Liquid Petroleum Gas (LPG), Methane (CH_4). The GPS gives the exact location where the obstacle is located. These are the sensors that are mounted on the unmanned ground vehicle to ensure the human safety.

Information from the sensors is stored in the microcontroller. From the microcontroller, information is stored in the cloud using a Wi-Fi. The cloud storage is provided by Firebase. From Firebase, the information is sent to the authorities concerned for further action. The concerned authorities may want to remove the obstacle, the information sent will help them to determine whether the conditions are safe for human involvement.

A rechargeable battery with voltage regulator acts as power source for the vehicle. The power source supplies power to Arduino, motor drives, and all the sensors.

2.3 Flowchart



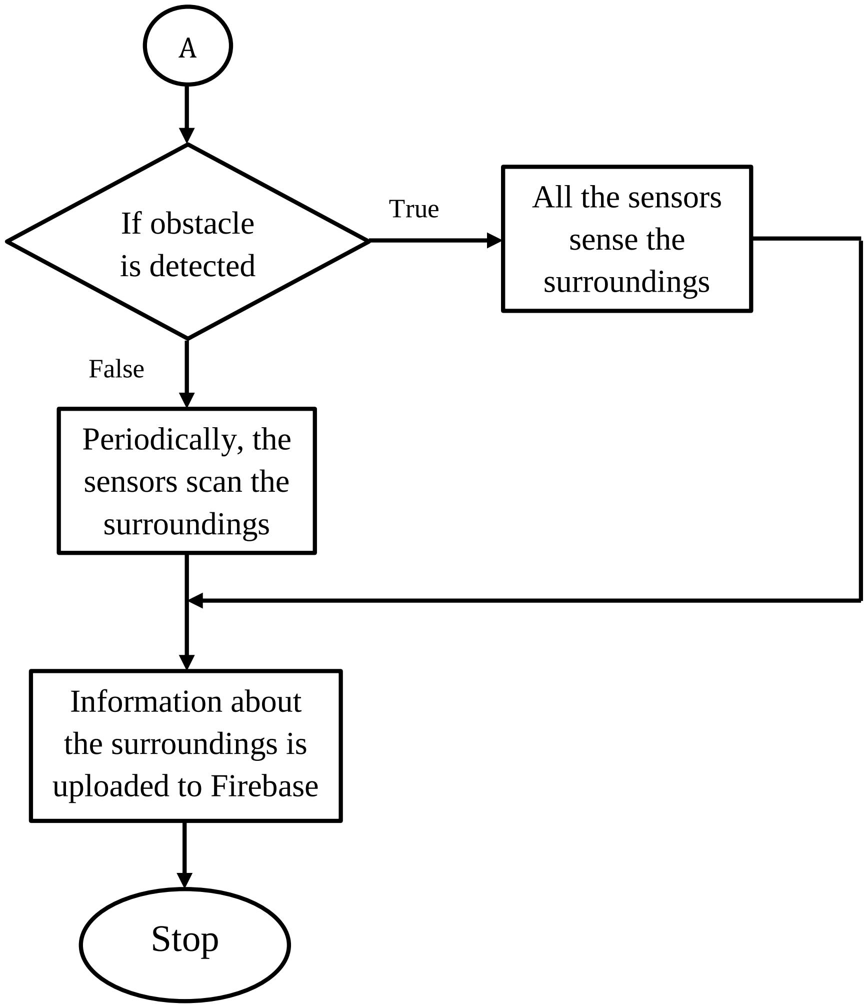


Fig 2.2: Flowchart

2.4 Working

An Android app is connected to the vehicle through Bluetooth. The vehicle acts on the input given from the app. The input is a character representing the desired movement of the vehicle. Based on the character, the vehicle moves in the desired direction through differential rotation of motors as shown in Fig 2.3.

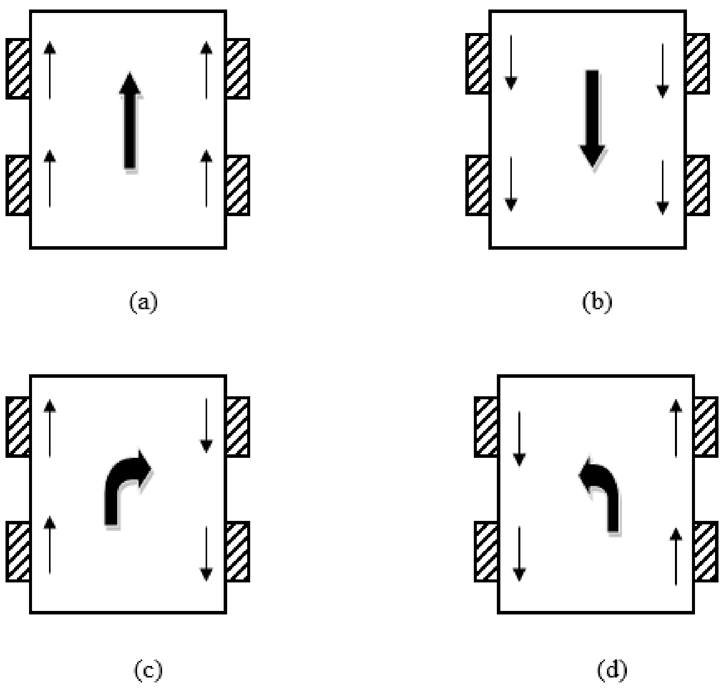


Fig 2.3: Directional Movement of UGV

(a) Forward - All four wheels rotate in forward direction

(b) Backward - All four wheels rotate in backward direction

(c) Turning Right - Right side wheels rotate in backward direction and Left side wheels rotate in forward direction

(d) Turning Left - Left side wheels rotate in backward direction and Right side wheels rotate in forward direction

The vehicle continuously checks for obstacles in its path. If any obstacle is detected, information about the surroundings is recorded. Periodically, all sensors scan the surroundings. The data is stored in Firebase for further action.

Chapter 3

Hardware and Software Details

3.1 Hardware Details

Microcontroller: Arduino Mega 2560

Motors: 45rpm, 3kg-cm torque DC motors

Motor drives: L298N

Bluetooth module: HC-05

Wi-Fi module: NodeMCU 1.0

Rechargeable battery: 12V, 1.3Ah Lead battery

Sensors:

- Ultrasonic sensor: HC-SR04
- Temperature sensor: LM35
- Gas sensor: MQ9
- GPS module: Questar G702-001UB

3.1.1 Arduino Mega

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 processor. It has 54 digital pins, of which 14 can be used as PWM outputs, 16 analog inputs, 4 UARTs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button as shown in Fig 3.1. It contains everything required 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.

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). 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 attachInterrupt() function for details.

- PWM: 0 to 13. Provide 8-bit PWM output with the analogWrite() function.

- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

- TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, 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 analogReference() function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller.

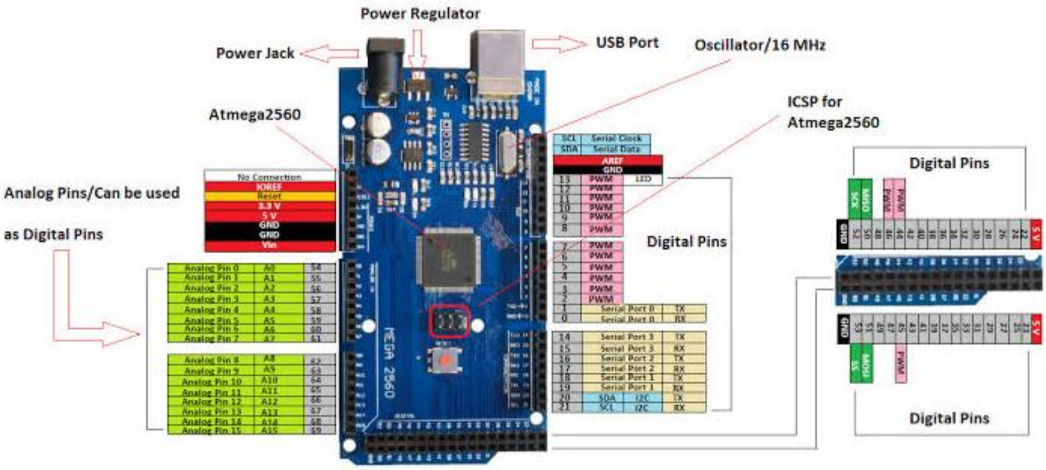


Fig 3.1: Arduino Mega 2560

3.1.2 DC Motors

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the 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.

A simple DC motor has a stationary set of magnets in the stator and an armature with one or more windings of insulated wire wrapped around a soft iron core that concentrates the magnetic field. The windings usually have multiple turns around the core, and in large motors there can be several parallel current paths. The ends of the wire winding are connected to a commutator. The commutator allows each armature coil to be energized in turn and connects the rotating coils with the external power supply through brushes.

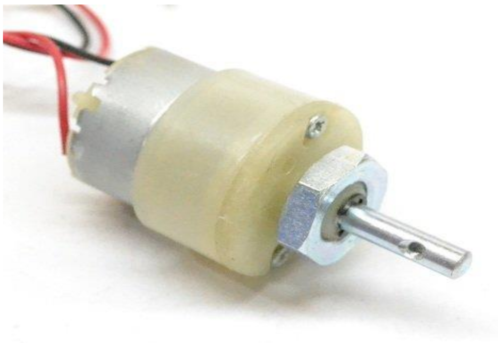


Fig 3.2: DC Motor

3.1.3 L298N Motor Driver

The L298N is an integrated monolithic circuit in a 15- lead Multi watt and PowerSO20 packages. It is a high voltage, high current dual full bridge driver designed to accept standard TTL logic levels and drive inductive loads such relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together the corresponding external terminal can be used for the connection of an external sensing resistor.

Motor drivers are made from discrete components which are integrated inside an IC. The input to the motor driver IC or motor driver circuit is a low current signal. The function of the circuit is to convert the low current signal to a high current signal. This high current signal is then given to the motor.

Features:

- Operating supply voltage up to 4-6 V
- Total DC current up to 4 A
- Low saturation voltage
- Logical "0" input voltage up to 1.5 V (HIGH NOISE IMMUNITY)
- Two motor direction indicator LEDs
- An onboard user-accessible 5V low-dropout regulator

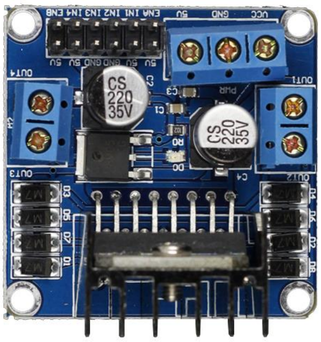


Fig 3.3: L298N Driver

3.1.4 HC-05 Bluetooth Module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. It can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). The Bluetooth module HC-05 is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc.

Hardware Features:

- Typical +80dBm sensitivity.
- Up to +4dBm RF transmit power.
- 3.3 to 5 V I/O.
- PIO (Programmable Input/Output) control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

The HC-05 Bluetooth Module has 6 pins. They are as follows:

ENABLE: When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate. When enable is left open or connected to 3.3V, the module is enabled i.e. the module remains on and communication also takes place.

Vcc: Supply Voltage 3.3V to 5V

GND: Ground pin

TXD & RXD: These two pins acts as an UART interface for communication

STATE: It acts as a status indicator. When the module is not connected to / paired with any other Bluetooth device, signal goes low. At this low state, the led flashes continuously which denotes that the module is not paired with other device. When this module is connected to/paired with any other Bluetooth device, the signal goes high. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

BUTTON SWITCH: This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other Bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

Fig 3.4: HC-05 Bluetooth Module

3.1.5 NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits.

The ESP8285 is an ESP8266 with 1 MiB of built-in flash, allowing for single-chip devices capable of connecting to Wi-Fi.

The pin out is as follows for the common ESP-01 module:

1. VCC, Voltage (+3.3 V; can handle up to 3.6 V)
2. GND, Ground (0 V)
3. RX, Receive data bit X
4. TX, Transmit data bit X
5. CH_PD, Chip power-down
6. RST, Reset
7. GPIO 0, General-purpose input/output No. 0
8. GPIO 2, General-purpose input/output No. 2

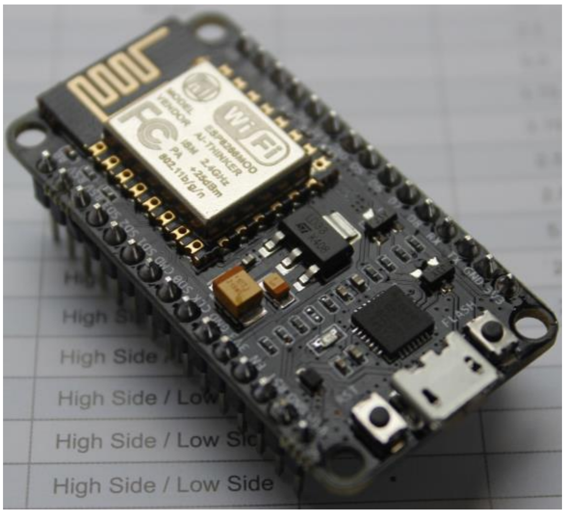


Fig 3.5: NodeMCU

3.1.6 Power Supply

Rechargeable Battery

A rechargeable battery is a type of electrical battery which can be charged, discharged into a load, and recharged many times, as opposed to a disposable or primary battery which is supplied fully charged and discarded after use. It is composed of one or more electrochemical cells. It accumulates and stores energy through a reversible electrochemical reaction. These are produced in many different shapes and sizes, ranging from button cells to megawatt systems connected to stabilize an electrical distribution network. Several different combinations of electrode materials and electrolytes are used, including lead-acid, nickel-cadmium (NiCd), nickel–metal hydride (NiMH), lithium-ion (Li-ion), and lithium-ion polymer (Li-ion polymer).

Initially, these are more costly than disposable batteries, but have a much lower total cost of ownership and environmental impact, as they can be recharged inexpensively many times before they need replacing. Some rechargeable battery types are available in the same sizes and voltages as disposable types, and can be used interchangeably with them.

Fig 3.6 (a): Rechargeable Battery

Fig 3.6 (b): Voltage Regulator

Voltage Regulator

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. A voltage regulator may be a simple "feed-forward" design or may include negative feedback control loops. Depending on the design, it may be used to regulate one or more AC or DC voltages.

Features:

- Input Voltage 4 – 35V DC
- Output Voltage 1.25 – 33V DC
- Output Current(max) 2A (3A if heat-sink is used)
- Conversion Efficiency(max) 92%
- Switching Frequency 150KHz
- Load Regulation ± 0.5%
- Voltage Regulation ± 2.5%

3.1.7 Ultrasonic Sensor

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. Ensured stable signal within 5m, gradually faded signal outside 5m till disappearing at 7m position. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

(1) Using IO trigger for at least 10us high level signal.

(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.

(3) If the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time * velocity of sound (340 m/s)) / 2.

Pins Description:

TRIG: Trigger Pulse Input

ECHO: Echo Pulse Output

GND: Ground

VCC: 5V Supply

Features of Ultrasonic Sensor:

Working Voltage	5V DC
Working current	16 mA
Working Frequency	40 Hz
Min - Max Range	2 – 400 cm

Fig 3.7: HC-SR04 Ultrasonic Sensor

3.1.8 Temperature Sensor

LM35 is a precision IC temperature sensor with its output proportional to the temperature (in ^oC). The sensor circuitry is sealed and therefore it is not subjected to oxidation and other processes. With LM35, temperature can be measured more accurately than with a thermistor. It also possess low self-heating and does not cause more than 0.1 ^oC temperature rise in still air.

The operating temperature range is from -55°C to 150°C. The output voltage varies by 10mV in response to every °C rise/fall in ambient temperature, i.e. its scale factor is 0.01V/°C.

Pin Description:

Supply Voltage	Vcc (5 V)
Output voltage	Output (6 V to -1 V)
Ground	0 V

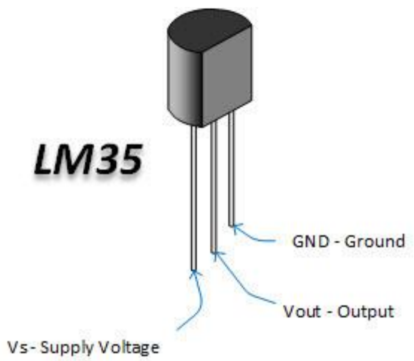


Fig 3.8: LM35 Temperature Sensor

3.1.9 Gas Sensor

The Gas Sensor (MQ9) module is useful for gas leakage detection (in home and industry). It is suitable for detecting LPG, CO, and CH4. Due to its high sensitivity and fast response time, measurements can be taken as soon as possible. The sensitivity of the sensor can be adjusted by using the potentiometer.

MQ 9 is an analog output sensor and needs to be connected to any one analog socket.

Features:

1. Good sensitivity to CO/Combustible Gas
2. High sensitivity to Methane, Propane, and CO
3. Long life and low cost
4. Simple drive circuit
5. Input voltage: DC 5±0.2V
6. Current Consumption: 150mA

Pin Description:

Supply	Vcc (5 V)
Ground	GND (0 V)
Analog Output	Analog signal Output
Digital Output	Digital signal Output

Fig 3.9: MQ-9 Gas Sensor

3.1.10 GPS

The GPS QUESTER TTL is a compact all-in-one GPS module solution intended for a broad range of Original Equipment Manufacturer (OEM) products, where fast and easy system integration and minimal development risk is required. The receiver continuously tracks all satellites in view and provides the accurate satellite positioning data. The GPS QUESTER TTL is optimised for applications requiring good performance, low cost, and maximum flexibility , suitable for a wide range of OEM configurations, including Handhelds ,sensors, asset tracking. PDA - centric personal navigation system, and vehicle navigation products. Its 56 parallel channels and 4100 search bins provide fast satellite of -140dBm and tracking sensitivity of -162 dBm offers good navigation performance even in urban canyons having sky view. Satellite - based augmentation systems, such as WAAS and EGNOS, are supported to yield improved accuracy. USB level serial interface is provided on the interface connector. Supply voltage of 3.8V-5.0V is supported. Their wireless solutions connect machines, vehicles and people to locate their exact positions and communicate via short range (WI-FI, Bluetooth) or cellular networks.

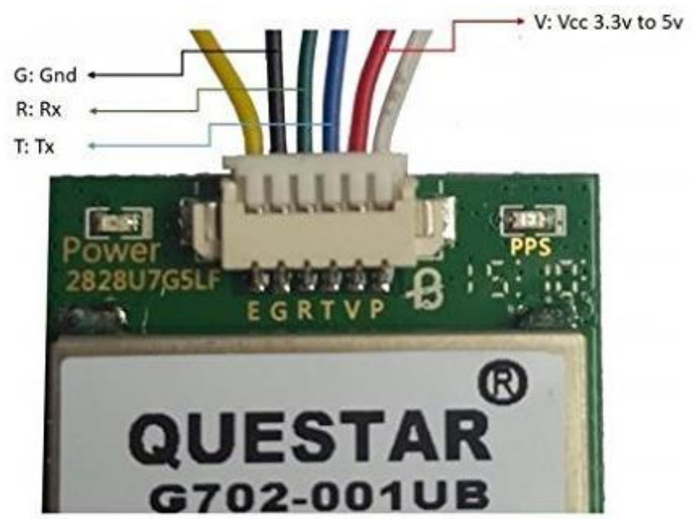


Fig 3.10: Questar G702-001UB GPS Module

3.2 Software Details

3.2.1 Arduino IDE

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer (e.g. Flash, Processing, and MaxMSP). The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

Why Arduino?

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

1. Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

2. Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

3. Simple, clear programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino

4. Open source and extensible software- The Arduino software and is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

5. Open source and extensible hardware - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168microcontrollers. The plans for the modules are published under a Creative Commons license.

3.2.2 Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of October 2018, the Firebase platform has 18 products, which are used by 1.5 million apps.

Firebase Real-time Database

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. The company provides client libraries that enables integration with Android, iOS, JavaScript, Java, ObjectiveC, Swift and Node.js applications.

The database is also accessible through a REST API and bindings for several JavaScript frameworks such as Angular JS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real-time database can secure their data by using the company's server-side-enforced security rules.

3.2.3 MIT App Inventer

App Inventor for Android is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT), which allows newcomers to computer programming to create software applications for the Android operating system (OS).

It uses a graphical interface very similar to Scratch and the StarLogoTNG user interface, which allows users to drag-and-drop visual objects to create an application that can run on Android devices. In creating App Inventor, Google drew upon significant prior research in educational computing, as well as work done within Google on online development environments.

App Inventor and the projects on which it is based are informed by constructionist learning theories, which emphasizes that programming can be a vehicle for engaging powerful ideas through active learning. As such, it is part of an ongoing movement in computers and education that began with the work of Seymour Papert and the MIT

Logo Group in the 1960s and has also manifested itself with Mitchel Resnick's work on Lego Mindstorms and StarLogo.

App Inventor also supports the use of cloud data via an experimental Firebase DB Component.

The App Inventor team was led by Hal Abelson and Mark Friedman. In the second half of 2011, Google released the source code, terminated its server, and provided funding for the creation of The MIT Center for Mobile Learning, led by App Inventor creator Hal Abelson and fellow MIT professors Eric Klopfer and Mitchel Resnick. The MIT version was launched in March 2012.

Chapter 4

Implementation

4.1 Configuring Arduino Mega

Installing the Arduino IDE

- Download Arduino IDE from https://www.arduino.cc/en/Main/Software
- Open the executable file Arduino-1.8.9-windows.exe
- Follow the steps and install Arduino IDE

Launch

- Launch the software, Arduino IDE.
- Open Blink sketch from examples: File > Examples > 1. Basics > Blink
- Select the type of Arduino board: Tools > Board > Arduino Mega 2560
- Select the serial port to which Arduino is connected: Tools > Port > COMx
- With the Arduino connected and serial port open, upload the sketch by clicking 'Upload' button.
- Arduino IDE will compile the sketch and upload it to the controller. Status of upload can be seen through blinking LEDs on the controller.
- After successful upload of Blink sketch, the on-board LED will blink.

4.2 Interfacing Motors with Arduino

For smooth movement of the vehicle, four DC motors have been used, one for each wheel. Motor drivers L298N are used to drive the motors. Two motors can be connected to a single driver. A driver has three pins for each motor: one enable and two control pins. The movement of vehicle depends on the two control pins while enable pin has to be high. A high to control pin 1 and low to control pin 2 will rotate the motor in clockwise direction. A low to control pin 1 and high to control pin 2 will rotate the motor in anti-clockwise direction. Supply to all the motors is given from power source through the drivers.

In order to identify the motors easily during connections, each motor has been named as Front Left (FL), Front Right (FR), Rear Left (RL), and Rear Right (RR). In the same way, the motor drivers are named as D1 and D2, where D1 is the front motor drive and D2 is the rear motor drive.

4.3 Controlling the UGV

The movement of the vehicle is controlled from an Android app. The app is developed using MIT App Inventor 2, which is an open-source platform for developers. The app layout is shown in Fig 5.3.1. On the other hand, a Bluetooth module HC-05 is connected to Arduino on the vehicle.

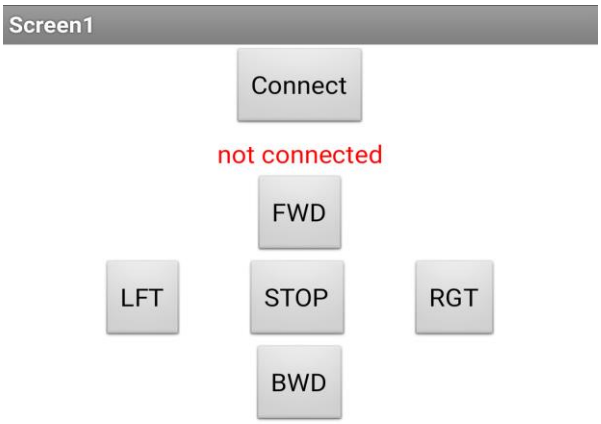


Fig 4.1: Controller App Layout

The android app connects to the vehicle over Bluetooth channel. On clicking ‘Connect’ button, the mobile is paired with Bluetooth module placed on the vehicle and “Connected” message is displayed in the app. On clicking ‘FWD’, the app sends a character ‘F’ to the Arduino which drives the vehicle forward. On clicking ‘BWD’, the app sends a character ‘B’ to the Arduino which drives the vehicle backward. On clicking ‘LFT’, the app sends a character ‘L’ to the Arduino which turns the vehicle left. On clicking ‘RGT’, the app sends a character ‘R’ to the Arduino which turns the vehicle right. On clicking ‘STOP’, the app sends a character ‘S’ to the Arduino which stops the vehicle.

4.4 Interfacing Sensors with Arduino

4.4.1 Ultrasonic Sensor

An ultrasonic sensors has 4 pins: Vcc, Gnd, Trig, and Echo. Vcc and Gnd pins are connected to the power supply. The trigger pin is connected to pin 2 and the echo pin is connected to pin 3 of the Arduino. A signal is continuously sent from Arduino to ultrasonic sensor which emits ultrasonic waves. If there is an obstacle, the wave reflects back. The echo senses the reflected wave and gives an analog output of time taken for the wave to return. From the time taken by the wave to travel to and fro, distance between the vehicle and obstacle can be calculated.

4.4.2 Temperature Sensor

LM35 is a three terminal linear temperature sensor with the pins Vcc, Gnd, and analog output. Vcc and Gnd pins are connected to the power supply. The analog output is connected to analog pin A0 of the Arduino. From the analog value read through pin A0, temperature can calculated in degree Celsius.

4.4.3 Gas Sensor

MQ-9 gas sensor has sensitivity to carbon monoxide, methane, and LPG. MQ-9 consists of 4 pins: Vcc, Gnd, analog output and digital output. Vcc and Gnd are connected to the power supply. The analog output is connected to analog pin A1 of the Arduino. Initially, the reference value of surrounding has to be determined. Using the analog output and reference value, concentration of gas can be determined.

4.4.4 GPS

Questar G702-001UB is a GPS module based on U-Blox. Before connecting GPS to the Arduino, GPS should be useable in our custom environment. GPS connection wires can be extended with longer wires. GPS module consists of Vcc, Gnd, RX, TX, PPS, and En. Vcc and Gnd are connected to the power supply. TX is connected to the pin and RX is connected to the pin. Using TinyGPS++ library, the data received from GPS module through serial communication can be read as coordinates.

4.5 Implementation of IoT

Internet of Things (IoT) is implemented using Firebase through NodeMCU. NodeMCU transfers information from Arduino to Firebase. The information is stored in Firebase Real-time Database.

4.5.1 NodeMCU

The NodeMCU is powered from power supply using Vcc and Gnd pins. NodeMCU is connected to the Arduino using RX and TX pins of both the boards. Data is transferred from Arduino to NodeMCU using serial communication. NodeMCU is connected to Firebase using a library FirebaseESP8266.

4.5.2 Firebase

Firebase is an open source database by Google LLC. A user with Google ID can access Firebase. A new project is to be created with Real-time Database. In the database, attributes can be created to store the data. Information about the obstacle and surroundings is sent from NodeMCU to Firebase. The data is stored in various attributes created with an identification of its path. The attributes are GPS Longitude, GPS Latitude, Gas concentration, and Temperature as shown in Fig 5.5.1 .

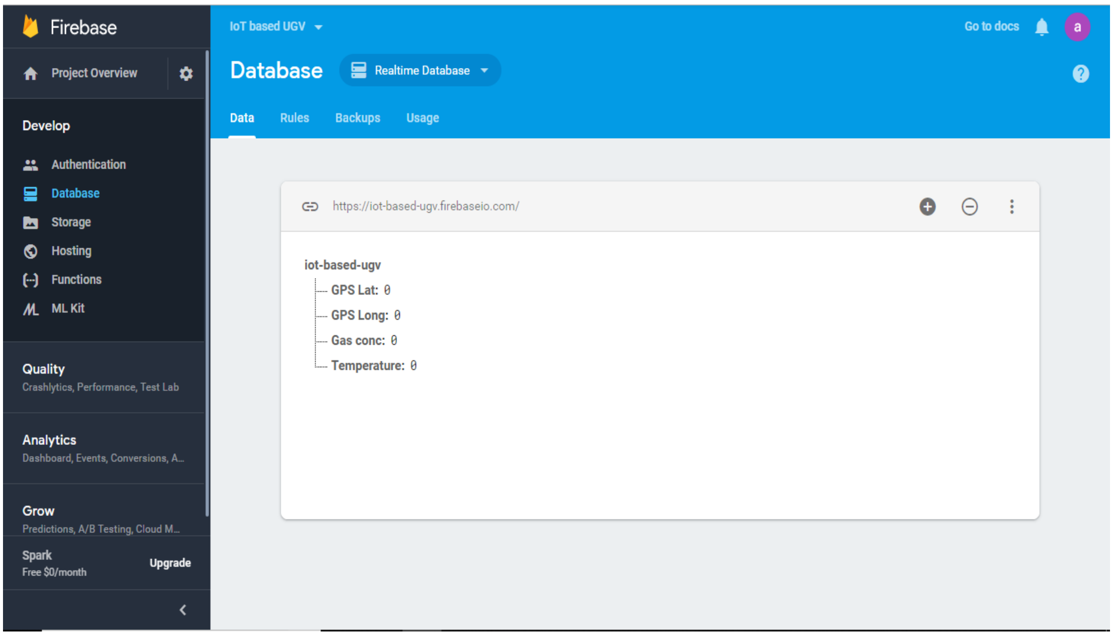


Fig 4.2: Firebase Layout of Real-time Database

Chapter 5

Results

Unmanned Ground Vehicle is a four-wheeled robot mounted with HC-SR04 ultrasonic sensor, MQ-9 gas sensor, LM-35 temperature sensor, and GPS. All these sensors work in tandem to get a perspective about the surroundings in a closed environment where human presence is difficult or dangerous.

The completed prototype of UGV is shown in Fig 5.1.

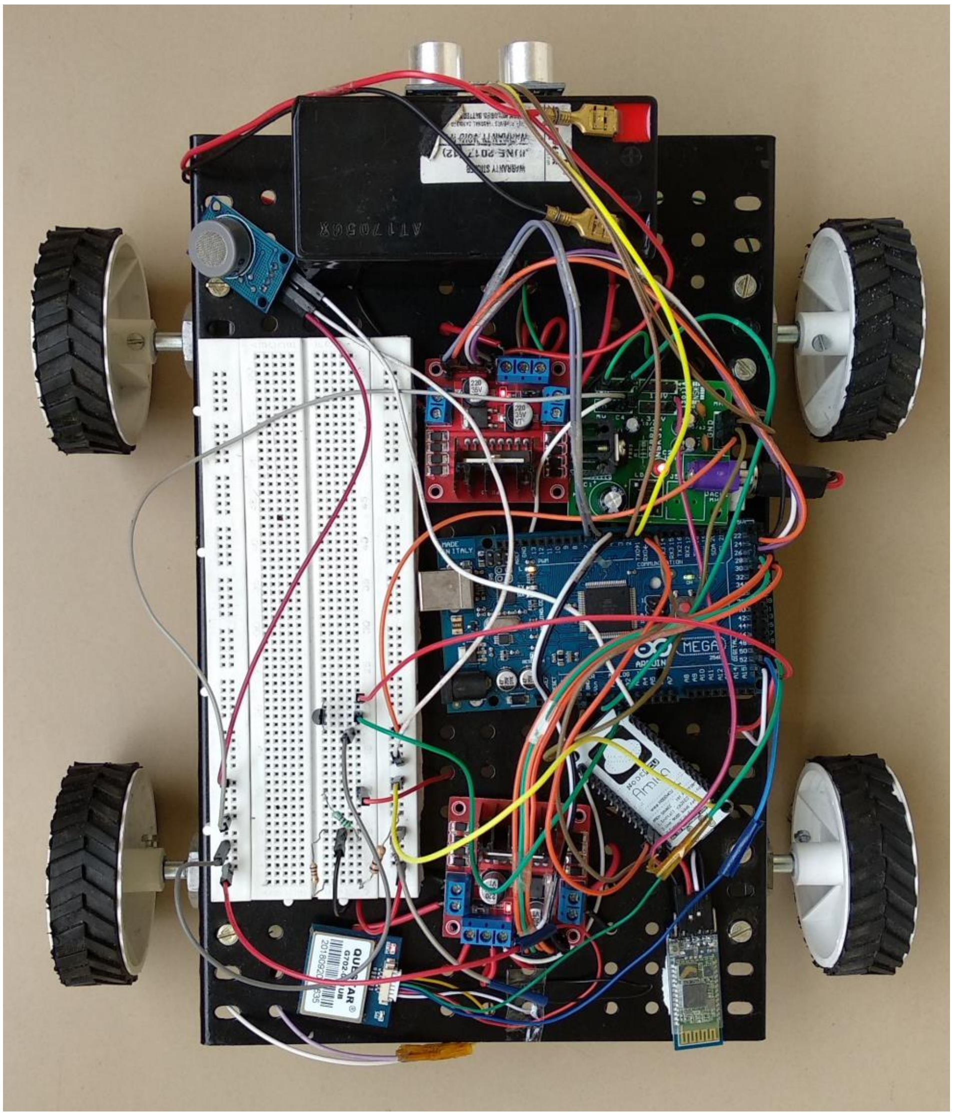


Fig 5.1: Unmanned Ground Vehicle

The ultrasonic sensor detects obstacles in its path through continuous emission and recognition of ultrasonic waves. If an obstacle is detected, all the other sensors feed the output to the microcontroller. The information is stored in the cloud using Firebase. Information stored in Firebase is as shown in Fig 5.2.

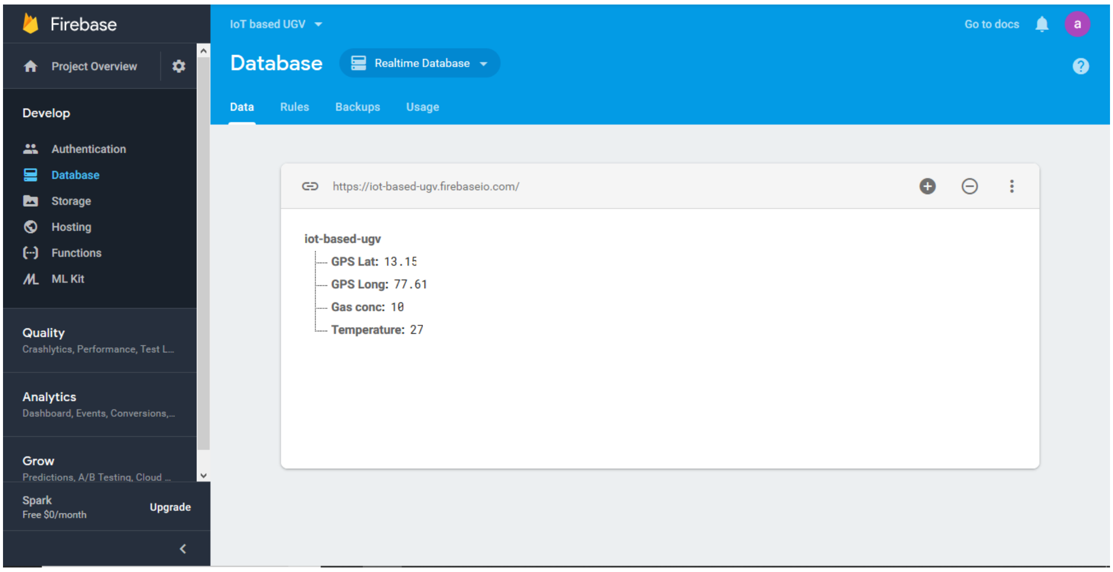


Fig 5.2: Firebase Layout with Results

Conclusion

Unmanned Ground Vehicle (UGV) has many applications and the demand is ever increasing. In the early 1960s, such autonomous vehicles were developed primarily for military applications. Today, autonomous vehicles are used extensively to provide valuable information during emergency situations like natural calamities.

Nowadays, ensuring the safety of people in challenging environments is the most important and difficult task. There are so many situations where humans cannot reach without prior knowledge of the surroundings. The surroundings may or may not be harmful for humans, but a first impression would be helpful.

In this project, information about the surroundings like temperature, presence of gas etc. is gathered using on-board sensors. Sensors interfaced on the vehicle are temperature and gas sensors which help to detect dangerous conditions. The temperature sensor senses the surrounding temperature and the gas sensor senses the presence of poisonous gases. GPS helps to know the exact location of the obstacle. This vehicle can be used by rescue personnel during the natural calamities like earthquake, soldiers in the battlefield and other surveillance personnel to plan and execute rescue operations effectively.

Future Scope

Semiautonomous Robotics for Future Combat Systems

The focus is to develop semiautonomous mobility technology critical to achieving the transformation envisioned for the future combat systems. A number of on-road and cross-country mobility experiments have successfully demonstrated initial capabilities in controlled environments. Continuing technical efforts are focused on perception and sensor technologies, intelligent vehicle control, tactical environment behaviors, and supervision of unmanned ground systems. Key technologies include obstacle avoidance, terrain characterization and classification, and fusion of data from multiple classifiers.

Obstacle Marking and Vehicle Guidance

The focus of this effort is to dispense smart markers to transmit and receive critical navigation information through and around obstacles or minefields. Successful development of this technology could be applicable not only to manned vehicles and dismounted forces but also to the path planning and path following of unmanned vehicles. It is planned to have a complete smart marker system that will be timely for evaluation with the robotic follower and semiautonomous robotic vehicle.

Mobility Support for Objective Force Maneuver

The results of this research will provide the capabilities and algorithms to quantify mobility and physical agility parameters that are essential for characterizing unmanned systems. The work addresses current deficiencies in modelling the breaching and crossing of complex obstacles with lighter vehicles than are now in the inventory. The current program plan projects that the technologies to quantify mobility in urban environments, asses and negotiate obstacles, and model reliable driving behaviors will be available. The products will include vehicle performance assessment tools and measures of performance assessment tools and measures of performance of physical agility.

References

[1] Johann Borenstein and Yoram Koren, "Obstacle Avoidance with Ultrasonic Sensors", IEEE Journal of Robotics and Automation, vol. 4, issue 2, pp. 213-218, April 1988

[2] Peng Zhang, Bo Huang and Takaaki Baba, "Obstacle Detecting System for Vehicle Collision Safeguard", 2007 International Conference on Communications, Circuits and Systems, Kokura, Japan

[3] Vivek Agarwal, N Venkata Murali and C Chandramouli, "A Cost-Effective Ultrasonic Sensor-based Driver-Assistance System for Congested Traffic Conditions", IEEE Transactions on Intelligent Transportation Systems, vol. 10, issue 3, pp. 486-498, September 2009

[4] M. Z. H. Noor, S. A. S. M. Zain and L. Mazalan, "Design and Development of Remote-Operated Multi-Direction Unmanned Ground Vehicle", 2013 IEEE 3rd International Conference on System Engineering and Technology, Shah Alam, Malaysia

[5] Mohammad Ariful Islam, Wahidul Ahad, Mohammad Faisal and Hasan U Zaman, "A Cost-Effective Design and Development of a Surveillance Robot", 2015 International Conference on Advances in Electrical Engineering, Dhaka, Bangladesh

[6] R. Girisrinivaas and V. Parthipan, "Drainage Overflow Monitoring System using IoT", 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, Chennai, India

[7] Sourabh Sarkar, Srijita Gayen and Saurabh Bilgaiyan, "Android based Home Security Systems using Internet of Things and Firebase", 2018 International Conference on Inventive Research in Computing Applications, Coimbatore, India

Appendix

Arduino Code

```
#include<TinyGPS++.h>

#include<SoftwareSerial.h>

SoftwareSerial Ps(50,51);

SoftwareSerial Bl(52,53);

TinyGPSPlus gps;

#define enFL 7

#define enFR 6

#define enRL 5

#define enRR 4

#define FL1 22

#define FL2 23

#define FR1 24

#define FR2 25

#define RL1 26

#define RL2 27

#define RR1 28

#define RR2 29

#define LED 13

#define echo 2

#define trig 3

#define ts A0

#define gs A1

char dc;

int i,durt,dist,tem,gasr,Lat,Long;

float temv,gasc,gasv,RS;
```

```
void setup()
{
pinMode(enFL,OUTPUT);
pinMode(enFR,OUTPUT);
pinMode(enRL,OUTPUT);
pinMode(enRR,OUTPUT);
pinMode(FL1,OUTPUT);
pinMode(FL2,OUTPUT);
pinMode(FR1,OUTPUT);
pinMode(FR2,OUTPUT);
pinMode(RL1,OUTPUT);
pinMode(RL2,OUTPUT);
pinMode(RR1,OUTPUT);
pinMode(RR2,OUTPUT);
pinMode(LED,OUTPUT);
pinMode(trig,OUTPUT);
pinMode(echo,INPUT);
pinMode(ts,INPUT);
pinMode(gs,INPUT);
Serial.begin(9600);
Ps.begin(9600);
Bl.begin(9600);
}
void MOV()
{
switch(dc)
{
case 'B':
```

```
  digitalWrite(FL1,LOW);
  digitalWrite(FL2,HIGH);
  digitalWrite(FR1,LOW);
  digitalWrite(FR2,HIGH);
  digitalWrite(RL1,LOW);
  digitalWrite(RL2,HIGH);
  digitalWrite(RR1,LOW);
  digitalWrite(RR2,HIGH);
  if(i!=256)
  acc();
  break;
case 'F':
  digitalWrite(FL2,LOW);
  digitalWrite(FL1,HIGH);
  digitalWrite(FR2,LOW);
  digitalWrite(FR1,HIGH);
  digitalWrite(RL2,LOW);
  digitalWrite(RL1,HIGH);
  digitalWrite(RR2,LOW);
  digitalWrite(RR1,HIGH);
  if(i!=256)
  acc();
  break;
case 'R':
  digitalWrite(FL2,LOW);
  digitalWrite(FL1,HIGH);
  digitalWrite(FR1,LOW);
  digitalWrite(FR2,HIGH);
```

```
 digitalWrite(RL2,LOW);
 digitalWrite(RL1,HIGH);
 digitalWrite(RR1,LOW);
 digitalWrite(RR2,HIGH);
 if(i!=256)
 acc();
 break;
case 'L':
 digitalWrite(FL1,LOW);
 digitalWrite(FL2,HIGH);
 digitalWrite(FR2,LOW);
 digitalWrite(FR1,HIGH);
 digitalWrite(RL1,LOW);
 digitalWrite(RL2,HIGH);
 digitalWrite(RR2,LOW);
 digitalWrite(RR1,HIGH);
 if(i!=256)
 acc();
 break;
case 'S': OFF();
default: OFF();
}
}
void acc()
{
for(i=0;i<256;i++)
{
analogWrite(enFL,i);
```

```
analogWrite(enFR,i);
analogWrite(enRL,i);
analogWrite(enRR,i);
delay(20);
}
}
void OFF()
{
digitalWrite(FL1,LOW);
digitalWrite(FL2,LOW);
digitalWrite(FR1,LOW);
digitalWrite(FR2,LOW);
digitalWrite(RL1,LOW);
digitalWrite(RL2,LOW);
digitalWrite(RR1,LOW);
digitalWrite(RR2,LOW);
}
void Ultra()
{
digitalWrite(LED,LOW);
digitalWrite(trig,HIGH);
delay(50);
digitalWrite(trig,LOW);
durt=pulseIn(echo,HIGH);
dist=(durt/2)/29.1;
if(dist<10)
{
digitalWrite(LED,HIGH);
```

```
dc='B';
MOV();
temp();
gas();
pos();
dc='S';
}
delay(50);
digitalWrite(LED,LOW);
}
void temp()
{
temv=analogRead(ts);
tem=temv/1024*500;
Serial.write(tem);
delay(100);
}
void gas()
{
gasv=analogRead(gs);
gasc=gasv*5.0/1024;
RS=(5-gasc)/gasc;
gasr=RS/2.03;
Serial.write(gasr);
delay(100);
}
void pos()
{
```

```
if(Ps.available() && gps.encode(Ps.read()))
{
if(gps.location.isValid())
{
Lat=gps.location.lat();
Long=gps.location.lng();
Serial.write(Lat);
delay(100);
Serial.write(Long);
}
}
}
void loop()
{
if(Bl.available())
dc=Bl.read();
temp();
gas();
pos();
MOV();
Ultra();
}
```