IOT Based Fire Fighting Car Robot

Submitted By

Md. Asmaul Haque

ID: 1834902083

AND

Omar Faruk

ID: 1834902168

A project submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering from City University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CITY UNIVERSITY, DHAKA, BANGLADESH

FEBRUARY 2023

DECLARATION

We hereby do solemnly declare that the work presented in this project has been carried out by us and has not been previously submitted to any other university for an academic degree. This project being submitted as a particular fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering commenced from Spring 2023 of City University is the result of our own project work. We hereby declare that no part of this project consists of materials copied or plagiarized from other published or unpublished source have either been put under quotation or duly acknowledged with full reference in appropriate places.

Signature of Author's Signature of Supervisor

…….. ……

Md.Asmaul Haque
ID#1834902083
B.Sc. in CSE
City University, Dhaka, Bangladesh

Md. Rakib Uddin
Coordinator & Lecturer
Department of CSE
City University, Dhaka, Bangladesh

……
Omar Faruk
ID#1834902168
B.Sc. in CSE
City University, Dhaka, Bangladesh

ACKNOWLEDGEMENT

First of all, we would like to pay our sincerest gratitude to the almighty “ALLAH" for keeping us in sound mind and health during the work and giving us the ability to work hard successfully. We are grateful to Prof. Dr. Engr. Md. Humaun Kabir, Dean, Faculty of Science & Engineering for allowing us to do the project. Specially, we would like to thanks to our supervisor, Md. Rakib Uddin, Coordinator & Lecturer, Department of Computer Science and Engineering, City University. His constant supervision, constructive criticism, valuable advice, Scholarly guidance, and encouragement at all stages of our work have made it possible to complete this project.

We would like to give thanks to all the teachers of the Department Computer Science and Engineering for their constructive criticism and valuable advice and cordial co-operation during the study at the University. Specially thank my family members who gave us huge support and inspiration during our whole B.Sc. Course in CSE.

In this connection, we also thank the officers and staff of the university for their co-operation and to all others not stated here for their help and co-operation. Computer Science and Engineering, City University (Bangladesh), who has supported us in any case with his kind and all the time during our project.

Finally, we would like to give thank our course mates, friends, and well-wishers who gave us huge support inspiration and their friendly and lovely company during our project.

ABSTRACT

Fire incident is a disaster that can potentially cause the loss of life, property damage and permanent disability to the affected victim. They can also suffer from prolonged psychological and trauma. Fire fighters are primarily tasked to handle fire incidents, but they are often exposed to higher risks when extinguishing fire, especially in hazardous environments such as in nuclear power plant, petroleum refineries and gas tanks. They are also faced with other difficulties, particularly if fire occurs in narrow and restricted places, as it is necessary to explore the ruins of buildings and obstacles to extinguish the fire and save the victim. With high barriers and risks in fire extinguishment operations, technological innovations can be utilized to assist firefighting. Therefore, this paper presents the development of a firefighting robot dubbed FFR that can extinguish fire without the need for fire fighters to be exposed to unnecessary danger. FFR is designed to be compact in size than other conventional firefighting robot in order to ease small location entry for deeper reach of extinguishing fire in narrow space. FFR is also equipped with an ultrasonic sensor to avoid it from hitting any obstacle and surrounding objects, while a flame sensor is attached for fire detection. This resulted in FFR demonstrating capabilities of identifying fire locations automatically and ability to extinguish fire remotely at particular distance. FFR is programmed to find the fire location and stop at maximum distance of 40 cm from the fire. A human operator can monitor the robot by using camera which connects to a smartphone or remote devices.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

Abbreviations

SL No	Short Name	Abbreviation
1	IoT	Internet of Thing
2	MCU	Microcontroller Unit
3	I2C	Inter Integrated Circuit
4	UART	Universal Asynchronies Receiver Transmitter

CHAPTER 1
INTRODUCTION

1.1 Introduction

A robot is an automated device which performs functions usually attributed to humans or machines tasked with repetitive or flexible set of actions. Numerous studies have shown that robot can be beneficial in medicine [1], rehabilitation [2-6], rescue operation [7, 8] and industry [9]. Over the years, robotics has been introduced in various industries. The industrial robots are multi-function manipulators designed for more specialized materials, divisions, gadgets or devices through various programmatic movements to perform various tasks [10]. In line with the Fourth Industrial Revolution (4IR), there is demand for a one system that can control, communicate and integrate different robots regardless of their types and specifications. Machine learning has also heated up interest in robotics, although only a portion of recent development in robotics can be associated with machine learning. Recent robotic development project has embedded machine learning algorithms [11-15] to increase the intelligence in robots. This will increase the productivity in industry while reducing the cost and electronic waste in a long run. Studies on the use of humanoid robots are actively carried out to minimize firefighters' injuries and deaths as well as increasing productivity, safety, efficiency and quality of the task given [16]. Robot can be divided into several groups such as Tele-robots, Telepresence robots, Mobile robots, Autonomous robots and Androids robots. Telepresence robot are similar to a tele-robot with the main difference of providing feedback from video, sound and other data. Hence, tele-presence robots are widely used in many fields requiring monitoring capability, such as in child nursery and education, and on improving older adult's social and daily activities [17, 18]. Mobile robot is designed to navigate and carry out tasks with the intervention of human beings [19, 20]. Meanwhile, autonomous robots can perform the task independently and receive the power from the environment, as opposed to android robots which are built to mimic humans [21]. In this paper, a firefighting robot is proposed. The main function of this robot is to become an unmanned support vehicle, developed to search and extinguish fire. There are several existing types of vehicles for firefighting at home and extinguish forest fires [22]. Our proposed robot is designed to be able to work on its

own or be controlled remotely. By using such robots, fire identification and rescue activities can be done with higher security without placing fire fighters at high risk and dangerous conditions. In other words, robots can reduce the need for fire fighters to get into dangerous situations. Additionally, having a compact size and automatic control also allows the robot to be used when fire occurs in small and narrow spaces with hazardous environments such as tunnels or nuclear power plants [23, 24]. Thermite and FFR are two current available fire fighter robots that have been used widely in industry. Thermite (produced by Howe and Howe Technologies Inc) is a firefighting robot that uses a remote control and can operate as far as 400 m. It can deliver up to 1200 gpm of water or 150 psi of foam. The size of this robot is 187.96 cm x 88.9 cm x 139.7 cm. This robot powers up to 25 bhp (18.64 kW) using a diesel engine. The main component in the design of this robot are multi-directional nozzle that is backed by a pump that can deliver 600 gpm (2271.25 l/min). This robot is designed for use in extreme danger areas, such as planes fires, processing factories, chemical plants or nuclear reactors. FFR is a fire-fighting vehicle controlled by a single operator via remote control. It extinguishes fire without intervention of fire fighters with a high pressure on a hydraulic arm that pumps water up to 55 m away. It also can carry 1800 litre of water and 600 litre of foam in its two on board tanks. The coating on FFR allows it to withstand critical temperature of 250ºC and thermal radiation of 23 kW/m for a period of 30 minutes. In this study, a compact and small firefighter robot has been developed. This robot is named FFR, which is short form of Rescue Robot. This robot can evade obstacles, search and extinguish fire. Furthermore, this robot can increase the productivity, safety, efficiency and quality of the task given. FFR is more compact and more flexible compared to Thermite and FFR. Another advantage of FFR is in its ability to enter location with small entrance or narrow space.

1.2 Objective of this Project

- To detect the fire automatically.
- To spray water in fire until the fire off.
- To live stream the fire condition in mobile or pc.
- To spray water and detect fire from multiple side.
- To carry water with robot.

1.3 Motivation

- Fire fighters face risky situations when extinguishing fires and rescuing victims, it is an inevitable part of being a fire fighter.
- In contrast, a robot can function by itself or be controlled from a distance, which means that fire fighting and rescue activities could be executed without putting fire fighters at risk by using robot technology instead.
- In other words, robots decrease the need for fire fighters to get into dangerous situations.

1.4 Proposed System

The methodology is divided into three parts. The first part is on the mechanicals schematics, followed by hardware description and the finally on the programming design. All parts were assembled together and experiments were then performed to determine the optimal distance of FFR to extinguish the fire were carried out.

1.5 Organization of the Project Book

This book consists of six chapters.

Chapter 1: Introduction

Gives a brief discussion of the project introduction, project objectives, project motivation and history.

Chapter 2: Literature Review

In this chapter the literature review and history are describe of our project also as a summary.

Chapter 3: Methodology

Here used some hardware equipment's like, a rechargeable 12v battery pack, A micro usb data cable, fire sensors, relay, water pump A microcontroller esp32 cam and a motor display and wood board as the chassis of the system.

Chapter 4: Results and Discussion

Here we describe and show our result and output. And the cost analysis of the project of our system is here in this chapter.

Chapter 5: Conclusion and Recommendation

Concludes overall the project. And advantages of this project. How this project will give it’s best in our daily work.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review

Author - Mrs. Bhavna K. Pancholi, Miss. Kena Patel The robot presented here is an embedded device in real time. C language is the software used to implement this type of robots. During automatic mode the robot tracks the environment to detect fire accidents. This robot uses IR sensors and output of this electrical sensors is fed to amplifier transistor. This signal is later fed into the micrcontroller's INpin. When a fire is observed, the microcontroller drives the motors and triggers the actuators. A water reservoir is mounted on the frame which has 10rpm DC pump motor. The water reservoir is attached to the hose and the end of which is placed on the robot's head. The water is sprayed to the flames in order to eliminate the fire. This paper also shows us how a robot is voice controlled. The robot is operated through the speech system. The commands are given to the robot in order to make them operated. The advantage of this type of robot is , hand free operation and fast data input. Disadvantage is the robot is affected by the environmental or external noise. B. Paper [2] –"Intelligent fire fighting tank robot." Author - E. Merry Sartika , Kristi Kosasih, Dan Muliady, M. Jimmy Hasugian. The tank robot is made of a mixture of acrylic, plastic, aluminum and iron. There are two servo motors for each wheel, two DC motors for two flame extinguishing fans, there are many other sensors like ultrasonic, compass, flame thermal array and many other. The robot is switched on by the sound and a sound activator circuit is mounted on it. The audio activation circuit consists of a Dual Tone Multi Frequency receiver and transmitter. Microcontroller AVR ATMega16 receives data from a sound activation circuit, an infrared and photodiode circuit as a white detector, a micro switch sensor as a furniture detector, UVTRON and TPA81 as flame detectors and thermal detectors, CMPS03 as navigation detectors, SRF04 as ultrasonic sensors. Microcontroller processes signal inputs and delivers signal outputs to the servo motor (GWS S03 4.8V) on the front-left wheel and the front-right wheel, and the DC motor to

spin the fan to extinguish the flame. C. Paper [3] –"Design and Implementation of Remote Controllers for Rescue Robots Used at Fire Sites." Author - Ho Kang, YoungDuk Kim, Duk-Han Sun, YoungSun Ryuh , Jeon-Il , Jinung An . There are 2 types of remote controllers in this project namely ultramobile personal computer and the other is Joystick personal platform. Though these are 2 different platforms their architectures are identical. The entire remote control system and the rescue robot is installed at the fire station. The communication between the controller and the robot starts with the use of fire station. Since fast data transmission is required , separate wireless channels are used. These separate wireless channels also offers low signal interference. This robot is mainly mounted with a camera for visual display , a LED to guide the evacuation path and different other sensors to measure the intensity of gases which are poisonous in the atmosphere. The informations are gathered and will be sent to the user using channels which are wireless. This robot plays a very important role in performing rescue operations. For easy transportation and portable size UMPC type is used and joystick type is used for stable manipulations. D. Paper [4] –" Advance Virtual RISC based Fire Fighting Robot" Author - Miss. Shraddha K. Dubal, Miss. Supriya S. Kadam, Miss Pratima S. Mane, Miss. Dipali A. Mali. This paper deals with the AVRbased firefighting robot. Infrared waves are not apparent to human eyes. So an infrared sensor is used. This is a tiny circuit which is used to send or receive the radio signals on range of the carrier frequencies. GSM is a data communication device for sending and receiving Radio Frequency signals wirelessly, which requires a wireless carrier sim card for its operation. The GSM requires a supply of DC voltage which is of 5V. The radio frequency transmitter takes the help of antennae and transforms electrical signal to electromagnetic signal. The antenna which are used here are wired loop antenna. A decoder is used to separate the address and also to convert it serial to parallel. The main of this project is ,it will detect the location's address where the incident got occurred. Since a buzzer is mounted on it , it alerts the people who are surrounded in that particular place. The robot basically displays the location and it sends the message to the fire brigade. E. Paper [5]- "Fire Locator, Detector and Extinguisher Robot with SMS Capability" Author - Arabiran, M.P., J.R. Mazo, J Frades, Undug, J. In this prototype we are used here is fire alarm system and smoke detector with sms capability. When we get an trigger from the smoke detector the robot will activate and it will undergo to the find mode. In the central unit is also embedded with an GSM module, so when it detect any thing it automatically contact to the owner and fire station about the

incident. when the robot is activated the infrared proximity sensor used to maintain the distance between the wall and robot and also used as navigator which robot itself guide and moves parallel to the walls. The photo elective sensor is used to move the robot forward direction, when the robot finds the wall it turns left automatically and robot finds the fire it rotates 360 degree clockwise and anticlock wise depending upon the room to move towards the fire. F. Paper [6] –"Fire fighting robot" Author - Mr. Adhav Gitanjali Subhash , Mr. Borse Karan Dipak, Miss. Shelke Amruta Ashok, Mr.Bansode Vishal Laxman, Mr. Gadekar Atish Mahadeo. In this project we are using vedio streaming robot, which will continuously capture the images with the help of camera of an android phone and sent to the web server. This robot will monitor the temperature where that will present with the help of temperature sensor, where these will be sent through android phone via Bluetooth module. The fire detection will be done with the help of smoke sensor also present in the system, after detecting the fire, the robot will automatically turn on the pump to extinguish the fire. For the obstracle detection IR sensor is used in the path of the robot, hence data from the robot is sent to the android phone through bluetooth module to the controller and sent to the web server. G. Paper

[7]:" CeaseFire: The Fire Fighting Robot" Author - Shiva Mittal, Meenakshi Mataray , Mayank Bhardwaj Manish Kumar Rana. The robot is extinguishing the fire using water and carbon dioxide(CO_2) sprays, where water tank is filled with a high pressure pump, the direction of the water pipes were controlled via the metal geared servos. It is switchable using solenoid valves. The CO_2 emission from the cylinder that can be controlled through another gas valve i.e. switchable wirelessly. In this camera is used over the top of the survey real fire situation, it also contain night vision camera to serve in dark to the hazard sight. It also contain a motors which is capable of providing much torque to carry the weight of the robot and also extinguish the fire equipment while moving its maximum speed. H. Paper [8]–"Development and Implementation of

Arduino Microcontroller Based Dual Mode Fire Extinguishing Robot" Author - Dinanath S. Nair, Sheik Mohammed S, Joyal Raju, Johaan Varkey Paul , Georgy Abel John . In fire fighting mode has both automated and manual mode. In automated mode, the robot is completely automated to search for the presence of possible flames and obstacle's. This robot moves around and search for the signs of flames. In case the presence of an obstracle is in the path of robot, it will deviate until it clear the obstracle. When the fire is detected in the long range the flame sensor which is placed either side of the robot, it will detected and move forward until the short range flame sensor is

lower than the cut off value. In manual mode, a Bluetooth module is used. The system is connected to smart phone by using a Bluetooth mode. The device can be configured to receive corresponding serial data controlling the robot which is transmitted from the smart phone.

2.2 History

The first commercial firefighting robot in the U.S. was developed by Maine-based Howe & Howe Technologies. Named the RS1-T2 Thermite, the initial firefighting robot was unveiled in 2012 and was based on technology the company created for the U.S. Army. The Thermite was designed to fight fires in environments where it is too dangerous for humans to go, such as airplane fires, nuclear reactors and other highhazard situations. In addition to going into dangerous scenarios, the Thermite is also more mobile and agile, which provides fire departments with an added tool in their arsenal. Since the initial Thermite was developed by Howe & Howe, they expanded the technology considerably. Look no further than the Thermite RS3, which boats a flow rate of 2,500 gallons per minute and has the strength to push vehicles from its path and pull up to 8,000 pounds. Although the Thermite was the original, it's certainly not the only robot that has been called into battle blazes. Firefighting robots became more wellknown in 2019 when crews used them to help battle the flames at the Notre Dame Cathedral in Paris. The Paris Fire Brigade used the Colossus robot. The tank-like technology navigated the centuries-old structure and provided vital information to firefighters on the outside. In an interview with the Institute of Electrical and Electronics Engineers, Cyril Kabbara, cofounder of Shark Robotics, explained how important a role the Colossus played. "Colossus acts as a kind of technical support station to the firefighting team by supplying information from its sensors to both the remote pilot and the other firefighters in real-time," Kabbara said. "This is an essential function in the dangerous circumstances the workers face when they enter an emergency scene, and it's very important that all the information is in the same place, as opposed to different sensors that the team has to carry with them into the building." While not every fire department will encounter such daunting challenges, firefighting robots provide many benefits. In 2020, the Los Angeles Fire Department became the first department in the United States to debut this game-changing technology. The LAFD debuted its Thermite RS3 in October 2020 and put it to work right away. Before the department even announced its

acquisition, it was helping out at an early-morning structure fire. Take a look at the video for details. In an interview with the Los Angeles

Times, LAFD Chief Ralph Terrazas said the biggest advantage offered by firefighting robots is that it provides the department a look inside dangerous environments when they would not risk sending humans inside. “I can afford to lose one of these wonderful machines. I cannot afford to lose a firefighter,” Terrazas told the newspaper. Whether it’s the RS3, Colossus robot or any future technology, it’s clear that firefighting robots have changed the industry and are helping keep firefighters and citizens safe.

Chapter 3

METHODOLOGY

3.1 Introduction

In this chapter we are going to describe all the hardware that we used for our project also describing all the circuit diagram block diagram software etc.

3.1.1 Required Components

1. NodeMCU
2. Battery
3. DC Motor
4. Motor Driver
5. Relay
6. Servo Motor
7. Water Pump
8. Spraying Nozzle
9. ESP 32 Cam

3.2 Node MCU

NodeMCU is an open source Lua based firmware for the ESP8266 WiFi SOC from Espressif and uses an on-module flash-based SPIFFS file system. NodeMCU is implemented in C and is layered on the Espressif NON-OS SDK. The firmware was initially developed as is a companion project to the popular ESP8266-based NodeMCU development modules, but the project is now community-supported .General-purpose input/output (GPIO) is a pin on an IC (Integrated Circuit). It can be either input pin or output pin, whose behavior can be controlled at the run time. NodeMCU Development kit provides access to these GPIOs of ESP8266. The only thing to take care is that NodeMCUDev kit pins are numbered differently than internal GPIO notations of ESP8266 as shown in below figure and table. For example, the D0 pin on the

NodeMCUDev kit is mapped to the internal GPIO pin 16 of ESP8266. [4]

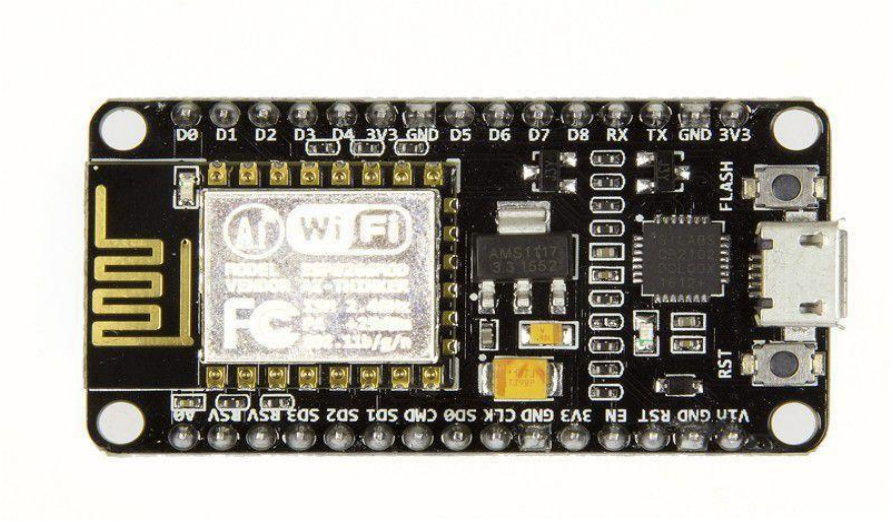


Figure 3.1: Node MCU

3.2.1 Pins of Node MCU

Table 3.1: Pins of Node MCU

SL No	Pin Names on NodeMCU	ESP8266 GPIO Pin number
1	D0	GPIO16
2	D1	GPIO5
3	D2	GPIO4
4	D3	GPIO0
5	D4	GPIO2
6	D5	GPIO14
7	D6	GPIO12
8	D7	GPIO13
9	D8	GPIO15
10	D9/RX	GPIO3
11	D10/TX	GPIO1
12	D11/SD2	GPIO9
13	D12/SD3	GPIO10

3.2.2 Communication of NodeMCU

There are three types of communications that Node MCU use commonly and they are

UART

UART stands for **Universal Asynchronous Reception and Transmission** and is a simple communication protocol that allows the NodeMCU to communicate with serial devices. The UART system communicates with digital pin 0 (RX), digital pin 1 (TX), and with another computer via the USB port.

This peripheral, found on all NodeMCU boards, allows the NodeMCU to directly communicate with a computer thanks to the fact that the Arduino has an onboard USBto-Serial converter. Therefore, programs written on a Windows, Mac, or Linux OS can be used with anNodeMCU connected to a USB port as if it was a serial port (serial port communication is trivial compared to USB communication).

I2C

I2C, which stands for **inter-integrated-circuit**, is a serial communications protocol specially designed for microcontrollers.

While this peripheral is almost never used for PC-device communication, it is incredibly popular with modules and sensors, making it useful for projects that require many parts working together. In fact, I2C allows you to potentially connect up to 128 devices to your main board!

When connecting two circuits to one another, think of the main device as the “master” and the connected devices—such as sensors, pin expansions, and drivers—as “slaves”. I2C makes it possible to connect multiple masters and slaves to your board while maintaining a clear communication pathway.

Maintaining a clear communication pathway is possible because I2C uses an address system and a shared bus, meaning many devices can be connected to the exact same wires. However, the NodeMCU must first select a specific device by transmitting a unique address before sending data. This provides each slave device with what it needs while also supporting multiple masters. I2C uses fewer wires and all data is transmitted

on a single wire, keeping your pin count low. The tradeoff for this simplified wiring is slower speeds than SPI.

SPI

SPI stands for **Serial Peripheral Interface**. Like I2C, SPI is a different form of serialcommunications protocol specially designed for microcontrollers to talk to each other.

Hwever, it has some key differences from its I2C counterpart.

3.3 L298 Motor Driver

L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

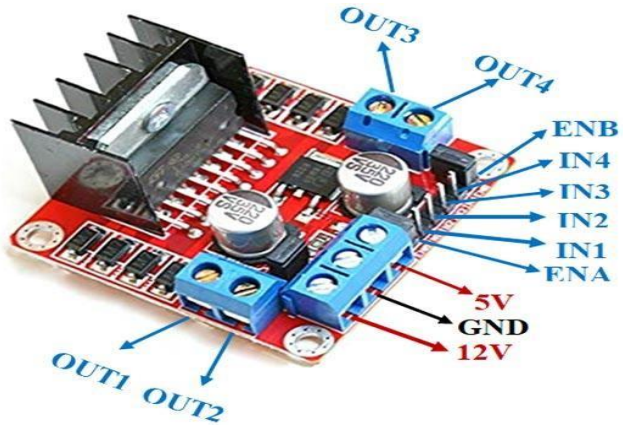


Figure 3.2: Motor Driver

3.3.1 Pins of L298 Motor Driver IC

Table 3.2: Pins of L298N Motor Driver

SL No	Pin Name	Description
1	IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
2	IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
3	ENA	Enables PWM signal for Motor A
4	ENB	Enables PWM signal for Motor B
5	OUT1 & OUT2	Output pins of Motor A
6	OUT3 & OUT4	Output pins of Motor B
7	12V	12V input from DC power Source
8	5V	Supplies power for the switching logic circuitry inside L298N IC
9	GND	Ground pin

3.4 5V Relay Module

Relay is an electromagnetic switch which his used to defer two circuits electrically and connect magnetically. When Nodemcu transmit the signal then relay driver receive signal and start its work. They are frequently used to interface an electronic circuit (working at low voltage) to an electrical circuit which works at extremely high voltage. For instance, a hand-off can make a 5V DC battery circuit to switch 230V AC mains circuit. In this way a little sensor circuit can drive, say, a fan or an electric knob. A transfer switch can be separated into two sections: information and yield. The info area has a loop which creates attractive field when a little voltage from an electronic circuit is connected to it. This voltage is known as the working voltage. Generally utilized transfers are accessible in various arrangement of working voltages like 6V, 9V, 12v, 24V and so on. In a basic hand-off there are three contactors: ordinarily shut (NC), regularly open (NO) and normal (COM).At no info express, the COM is associated with NC. At the point when the working voltage is connected the transfer curl gets charged and the COM changes contact to NO. Diverse transfer setups are accessible like SPDT and DPDT which have distinctive number of changeover contacts. By utilizing legitimate blend of contactors, the electrical circuit can be turned on and off. So as to drive the hand-off, we use transistor and just less power can be utilized to get the transfer driven. Since, transistor is an intensifier so the base lead gets adequate current to make increasingly current stream from Emitter of Transistor to Collector. In the event that the

Base once gets control that is adequate, at that point the transistor lead from Emitter to Collector and power the transfer. When the power is transmit to the relay works as switch due to electromagnetic effect so that we can switch ON or OFF our home appliances. [3]

Figure 3.3: 5V Relay Module

3.4.1 Pin Description of 5V Relay Module

Table 3.3: Pin Description of 5V Relay Module

Pin Name	Descriptions
DC +	Positive supply voltage
DC -	Ground
IN	Relay control port

3.5 Water Pump

This is a low cost mini submersible type water pump that works on 12V DC. It is extremely simple and easy to use. Just immerse the pump in water, connect a suitable pipe to the outlet and power the motor with 12V to start pumping water. Great for building science projects, fire-extinguishers, firefighting robots, fountains, waterfalls, plant watering systems etc .

Figure 3.4: Water Pump

3.6 DC Gear Motor

A Direct Current (DC) motor is a rotating electrical device that converts direct current, of electrical energy, into mechanical energy. An Inductor (coil) inside the DC motor produces a magnetic field that creates rotary motion as DC voltage is applied to its terminal.

Figure 3.5: DC Gear Motor

3.7 Jumper Wire

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

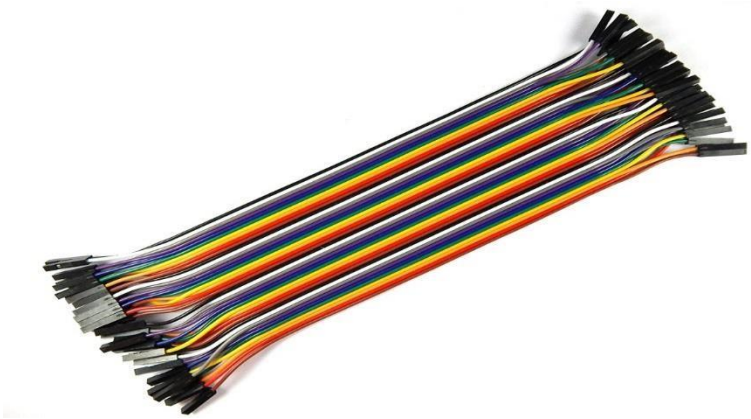


Figure 3.6: Jumper wires

3.8 12V DC Power Supply

We need two power supply for our system. One for full device and one 12 volts DC for our pump. As 12 volts power supply we use three 4 volt's Li-Po battery which is connected in series.

Figure 3.7: Li-Po Battery

3.9 Smart Phone

To see the real time condition of cornfield and also control the pump and other electrical device we need an android or ios device with internet connection.

Figure 3.8: Mobile Phone

3.10 ESP 32 CAM

The ESP32-CAM is a development board with an ESP32-S chip, an OV2640 camera, microSD card slot and several GPIOs to connect peripherals. In this guide, we'll take a look at the ESP32-CAM GPIOs and how to use them.

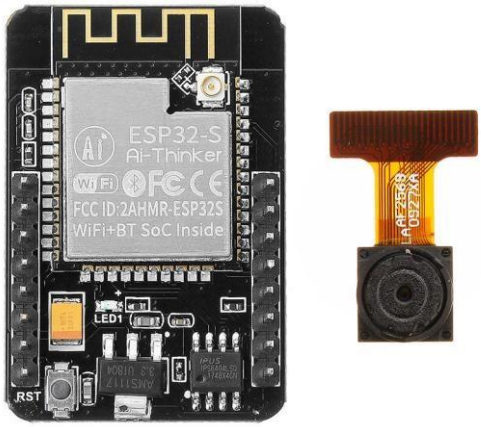


Figure 3.9: ESP 32 CAM

3.10.1 Pins of ESP 32 CAM

Table 3.4: Pins of ESP 32 CAM

SL No	OV2640 CAMERA	ESP32	Variable name in code
1	D0	GPIO 5	Y2_GPIO_NUM
2	D1	GPIO 18	Y3_GPIO_NUM
3	D2	GPIO 19	Y4_GPIO_NUM
4	D3	GPIO 21	Y5_GPIO_NUM
5	D4	GPIO 36	Y6_GPIO_NUM
6	D5	GPIO 39	Y7_GPIO_NUM
7	D6	GPIO 34	Y8_GPIO_NUM
8	D7	GPIO 35	Y9_GPIO_NUM
9	XCLK	GPIO 0	XCLK_GPIO_NUM
10	PCLK	GPIO 22	PCLK_GPIO_NUM
11	VSYNC	GPIO 25	VSYNC_GPIO_NUM
12	HREF	GPIO 23	HREF_GPIO_NUM
13	SDA	GPIO 26	SIOD_GPIO_NUM
14	SCL	GPIO 27	SIOC_GPIO_NUM
15	POWER PIN	GPIO 32	PWDN_GPIO_NUM

3.11 Arduino IDE Software

To make this project we used 2 software. One is programming code editor software which name is Arduino Ide. The one is an Android app which is for controlling our appliances For coding for Node MCU we need an IDE which is compatible to our NodeMCU. The Arduino IDE is one of the most easiest and compatible IDE for Node MCU. So we choice Arduino IDE for our coding.

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++.It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards. Arduino Pro IDE

Repository github.com/arduino/Arduino Edit this at Wikidata Written in C, C++ Operating system Windows, macOS, Linux Platform IA-32, x86-64, ARM TypeIntegrated development environment License LGPL or GPL license Website blog.arduino.cc/2019/10/18/arduino-pro-ide-alpha-preview-with-advanced-features/
With the rising popularity of Arduino as a software platform, other vendors started to implement custom open source compilers and tools (cores) that can build and upload sketches to other microcontrollers that are not supported by Arduino's official line of

microcontrollers. [4]

Figure 3.10: Arduino IDE

3.12 Controlling Blynk App

We create this app through an app named Blynk. Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, and it can store data, visualize it and do many other cool things. There are three major components in the platform. Blynk Server - responsible for all the communications between the smartphone and hardware. [4]

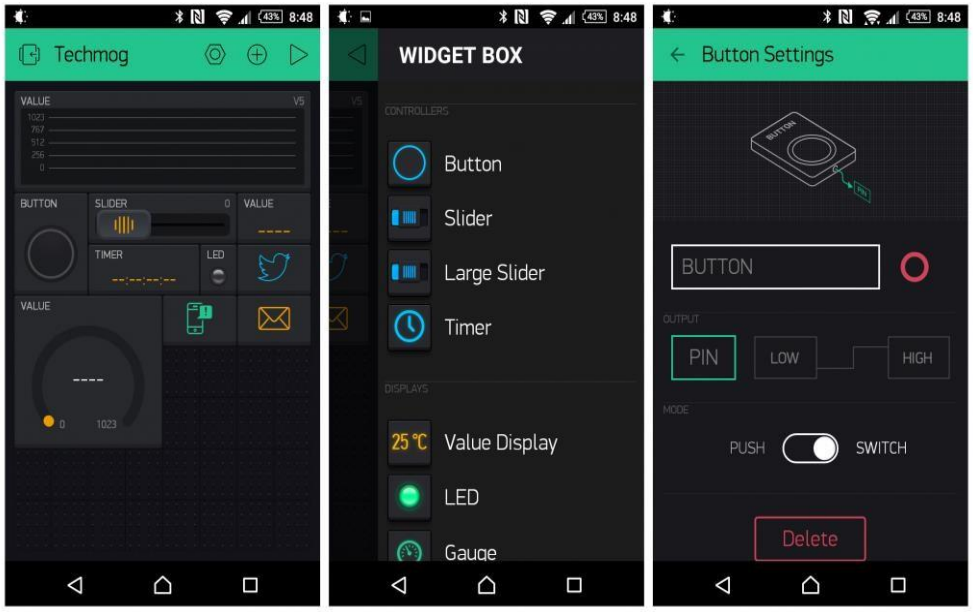


Figure 3.11: Android app.

3.13 Flame Sensor

A sensor which is most sensitive to a normal light is known as a flame sensor. That's why this sensor module is used in flame alarms. This sensor detects flame otherwise wavelength within the range of 760 nm – 1100 nm from the light source. This sensor can be easily damaged to high temperature. So this sensor can be placed at a certain distance from the flame. The flame detection can be done from a 100cm distance and the detection angle will be 600. The output of this sensor is an analog signal or digital signal. These sensors are used in firefighting robots like as a flame alarm. A flamesensor is one kind of detector which is mainly designed for detecting as well as responding to the occurrence of a fire or flame. The flame detection response can depend on its fitting. It includes an alarm system, a natural gas line, propane & a fire suppression system. This sensor is used in industrial boilers. The main function of this is to give authentication whether the boiler is properly working or not. The response of these sensors is faster as well as more accurate compare with a heat/smoke detector because of its mechanism while detecting the flame.

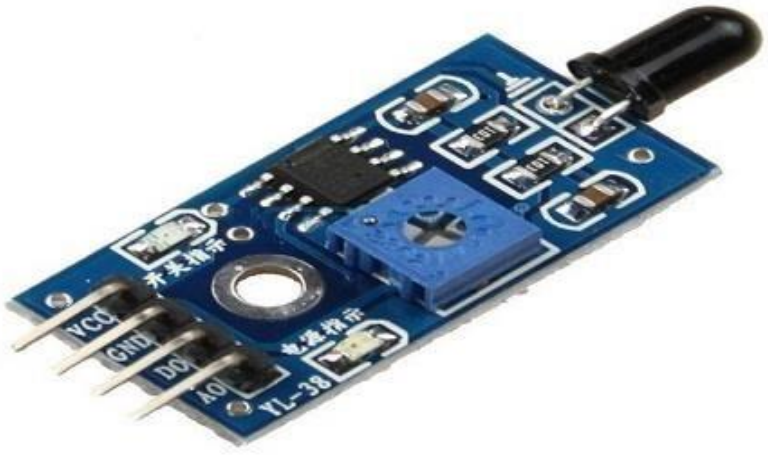


Figure 3.12: Flame Sensor

3.14 Servo Motor

Micro Servo Motor SG90 is a tiny and lightweight server motor with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like

the standard kinds but smaller. You can use any servo code, hardware or library to control these servos.

Figure 3.13: Servo Motor

3.15 Circuit Diagram

Hardware connection is same as like as the circuit diagram.1st we program the Nodemcu and ESP 32 Cam then we connect all the equipment as per our circuit diagram.

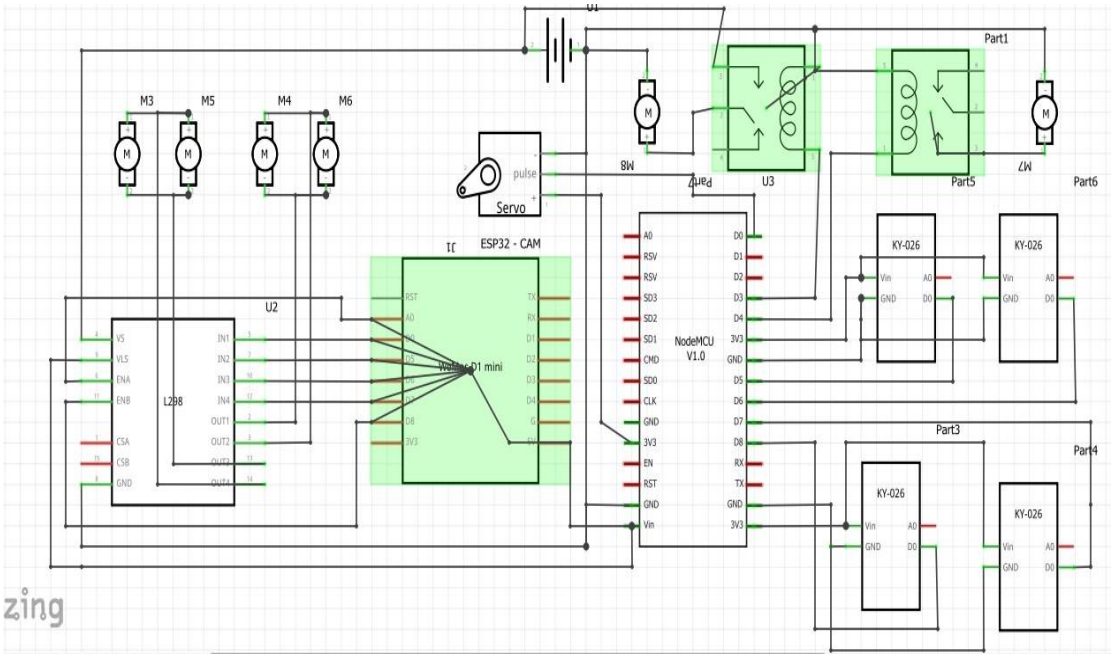


Figure 3.14: Circuit Diagram

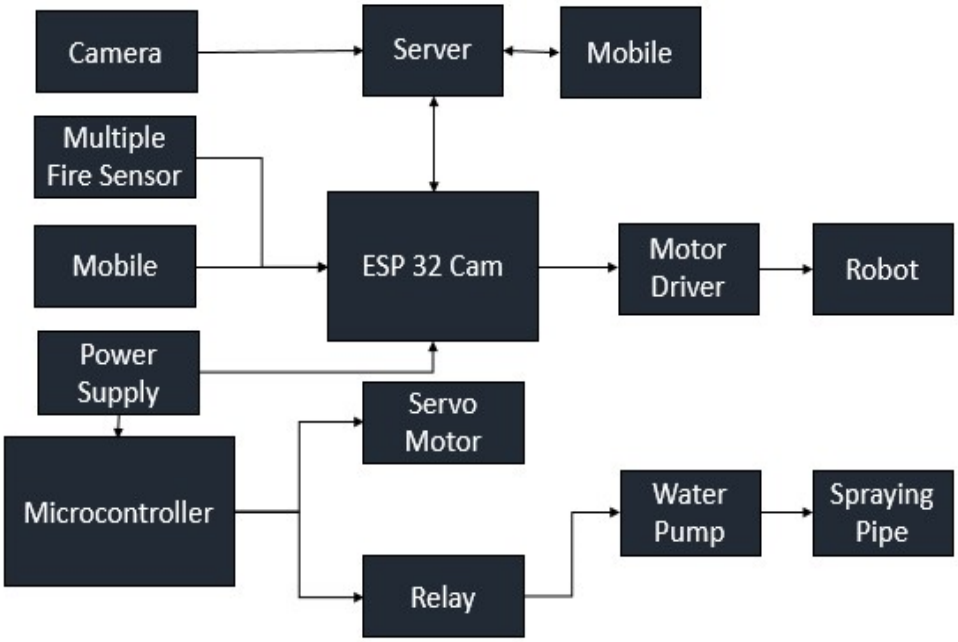
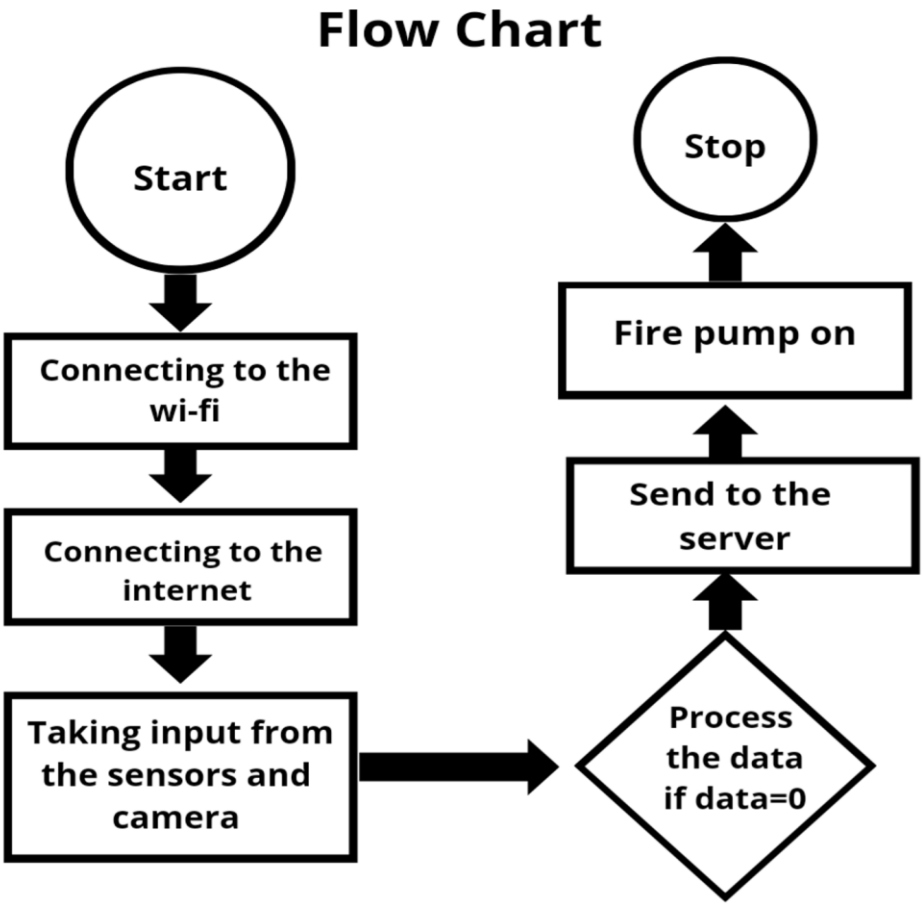


Figure 3.14: Block Diagram



3.16 Required Software and Their Setup Process

To complete our project, we need many types of software, most important of them are following

1. Arduino IDE-1.8.9

3.17 Software

The software that is used to program the microcontroller is open-source-software and can be downloaded for free on www.arduino.cc. With this “Arduino software” we can write little programs with the microcontroller. These programs are called “Sketch”.

In the end the sketches are transferred to the microcontroller by USB cable. More on that later on the subject “programming”.

3.17.1 Arduino IDE Installation

Now one after another the Arduino software and the USB driver for the board have to be installed.

3.17.2 Installation and setup of the Arduino software

1. We have downloaded the Arduino software from www.arduino.cc and installed it on the computer (This was NOT connected to the PC). After that we opened the software file and installed the program named arduino.exe.

Two set ups on the program are important and should be considered.

a) The board that we want to connect has to be selected on the arduino software.

The “Arduino Uno” is here known as “Arduino / Genuino Uno ,Nano, Lilipo or any name can be”.

Fig. 3.15: Program installation process -1

b) We have to choose the right “Serial-Port”, to let the Computer know to which port the board has been connected. That is only possible if the USB driver has been installed correctly. It can be checked this way:

At the moment the Arduino wasn’t connected to the PC. If we now choose “Port”, under the field “Tool”, we will already see one or more ports here (COM1/ COM2/ COM3…).

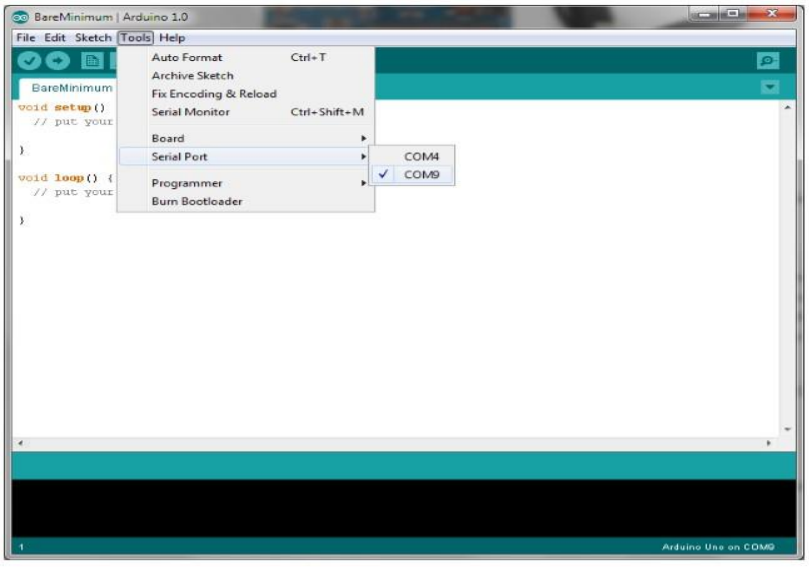
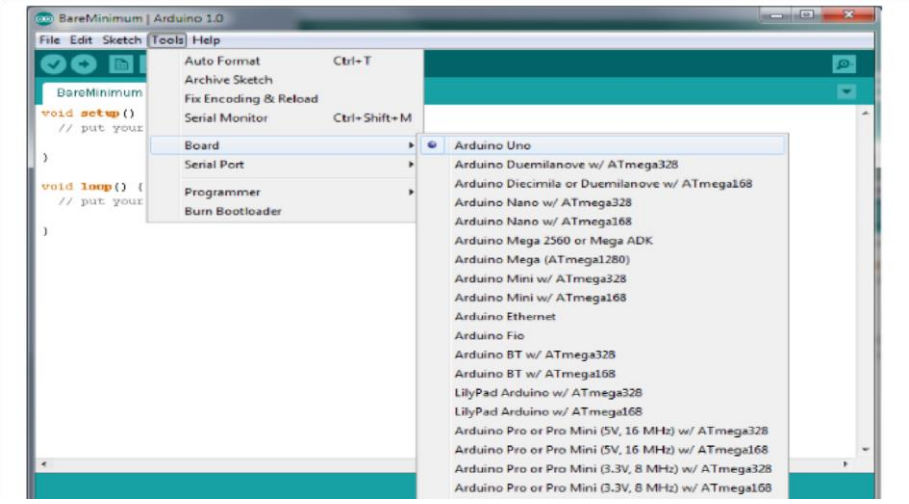
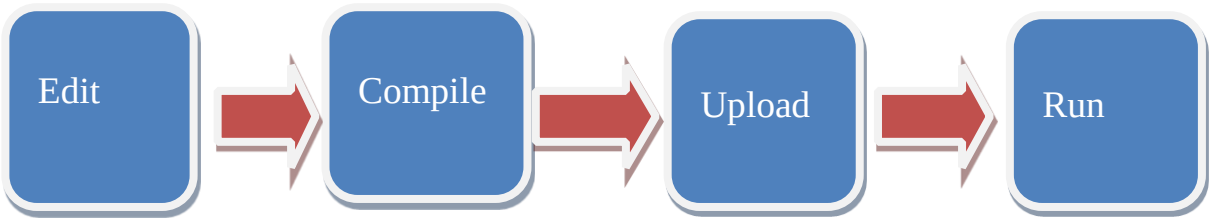


Figure 3.16: Program installation process -2





3.18 Programming

The development cycle is divided into 4 phases:

Figure 3.16: Flowchart of the compiling process

Compile: Compile means to translate the sketch into machine language, also known as object.

Code Run: Arduino sketch is executed as soon as terminates the step of uploading on the board.

3.18.1 Arduino Program Development

• Based on C++ without 80% of the instructions.
• A handful of new commands.
• Programs are called 'sketches'.
• Sketches need two functions:
• void setup ( )
• Void loop ( )
• Setup ( ) runs first and once.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter we are going to elaborate our project's final output and result. Also we discussed about how our project works. How it extinguished fire how spray the water and when. How the system up lift and down lift the spray pipes. How the camera stream the video.

4.2 Result

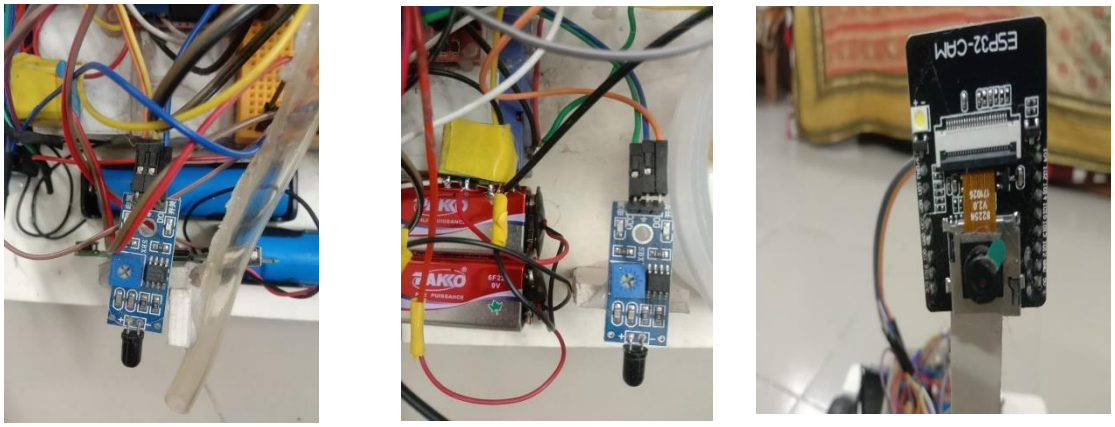


Figure 4.1: Sensors and Camera Placement

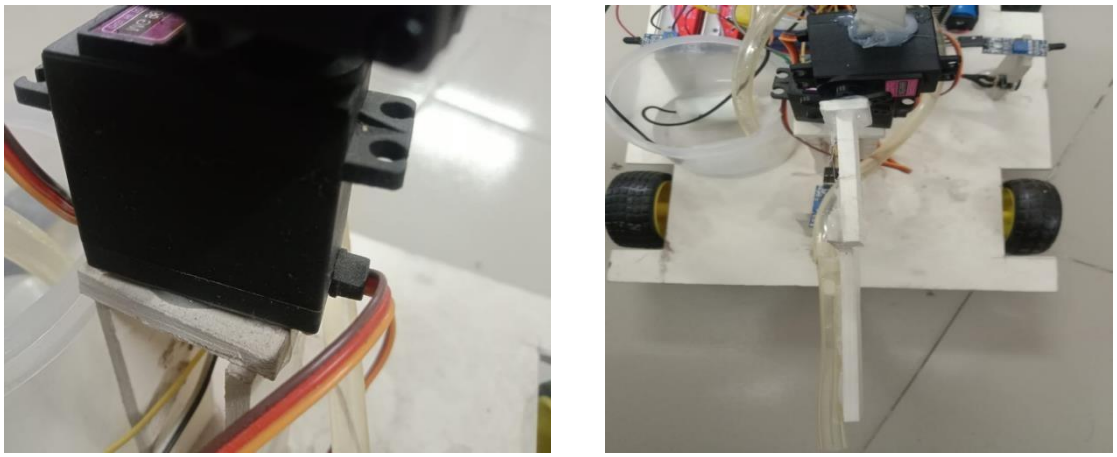


Figure 4.2: Servo Motor Placement for Rotating Pipe

4.3 Project Outlook

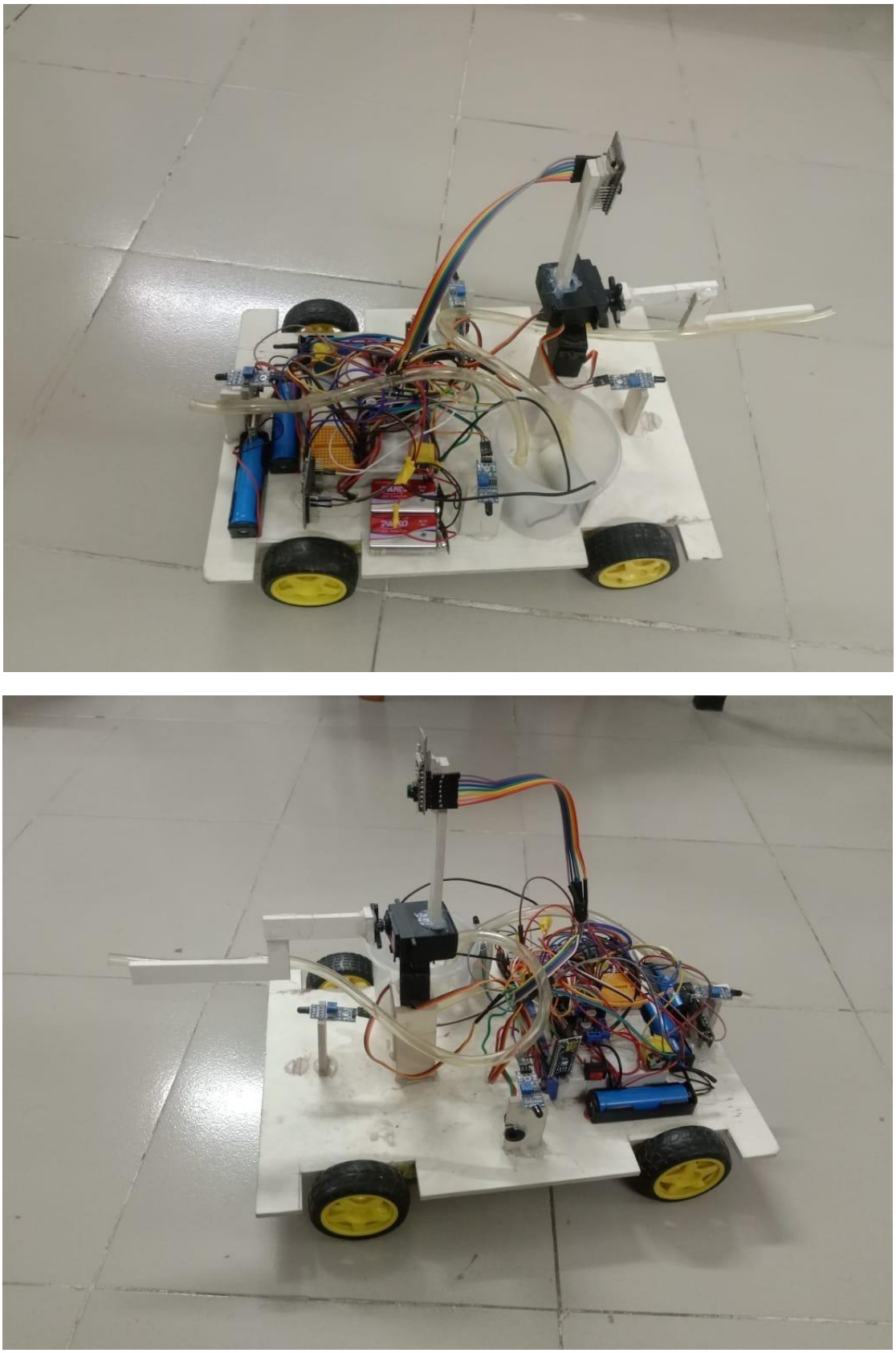


Figure 4.3: Project Outlook

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter we are going to describe our project's advantages, limitations, Future Improvement, Cost analysis and conclusions**.**

5.2 Advantages

- To detect the exact direction of the fire source.
- Capability of sensing accurately with increased flexibility.
- Reduce human effort.
- Reliable and economical.
- Not sensitive to weather conditions.

5.3 Limitations

- As a prototype the sensing area is short.
- Due to Wi-Fi connectivity sometimes lost the connection.
- Low water storage capacity.
- Camera resolution.

5.4 Future Scope

- Sensors can be more calibrated against a range of temperature and humidity.
- Self – Calibration techniques can be implemented in future.
- Controlling Port
- In future we will add own server

5.5 Cost Analysis

Component	Quantity	Cost per unit	Total Cost
ESP32	1	1200tk	1200tk
Arduino Nano	1	1000tk	1000tk
Node MCU	1	700tk	700tk
Servo Motor	2	1100tk	2200tk
Fire Sensor	4	600tk	2400tk
Relay	2	150tk	300tk
Battery	8	200tk	1600tk
Motor Driver	1	300tk	300tk
Gear Motor	4	600tk	2400tk
Wheel	4	100tk	400tk
Pump	2	200tk	400tk
Maintenance Cost			1000tk
Total cost			12600tk

Conclusion

Nowadays, the concept of fire-fighter robot is very common. But this IoT fire-fighter robot solves some issues in the existing fire-fighter robots. They are both autonomous and controlled. It has a monitoring system to know the fire and people who stuck in the environment. Also the monitoring system can be implemented as an Android application. The robot helps other fire-fighter men by giving them a safer path to fire.

Because lots of fire-fighter men have lost their life while saving other's life. Using the robotic arm in the robot, it can pick up the people who stuck in there for help. This IoT fire-fighter robot can be implemented as in the form of drone. Using the application of artificial intelligence, the robot can be trained to find fire type and to choose appropriate extinguishing methods. For some large fires like forest fire, the extended version of this robot helps to extinguish the fire in a large amount. So, nowadays the relevance of this robot is increasing.

REFERENCES

[1] Jeelani, S., et al., Robotics and medicine: A scientific rainbow in hospital. Journal of Pharmacy & Bioallied Sciences, 2015. 7(Suppl 2): p. S381-S383.

[2] Aliff, M., S. Dohta, and T. Akagi, Simple Trajectory Control Method of Robot Arm Using Flexible Pneumatic Cylinders. Journal of Robotics and Mechatronics, 2015. 27(6): p. 698-705.

[3] Aliff M, D.S., and Akagi T, Control and analysis of simple-structured robot arm using flexible pneumatic cylinders. International Journal of Advanced and Applied Sciences, 2017. 4(12): p. 151-157.

[4] M. Aliff, S. Dohta and T. Akagi, Trajectory controls and its analysis for robot arm using flexible pneumatic cylinders," IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), 2015, pp. 48-54.

[5] M. Aliff, S. Dohta and T. Akagi, Trajectory control of robot arm using flexible pneumatic cylinders and embedded controller, IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2015, pp. 1120-1125.

[6] C. Xin, D. Qiao, S. Hongjie, L. Chunhe and Z. Haikuan, Design and Implementation of Debris Search and Rescue Robot System Based on Internet of Things, International Conference on Smart Grid and Electrical Automation (ICSGEA), 2018, pp. 303-307.

[7] Yusof, M., and Dodd, T., Pangolin: A Variable Geometry Tracked Vehicle With Independent Track Control, Field Robotics, pp. 917-924. [9] Day, C.-P., Robotics in Industry—Their Role in Intelligent Manufacturing. Engineering, 2018. 4(4): p. 440-445.

[8] J. Lee, G. Park, J. Shin and J. Woo, Industrial robot calibration method using denavit — Hatenberg parameters, 17th International Conference on Control, Automation and Systems (ICCAS), 2017, pp. 1834-1837

[9] Sani, N. S., Shamsuddin, I. I. S., Sahran, S., Rahman, A. H. A and Muzaffar, E. N, Redefining selection of features and classification algorithms for room occupancy detection, International Journal on Advanced Science, Engineering and Information Technology, 2018, 8(4- 2), pp. 1486-1493.

[10] Holliday, J. D., Sani, N., and Willett, P., Calculation of substructural analysis weights using a genetic algorithm, Journal of Chemical Information .

Appendix

A. Programming Code for the Project

```
#include "esp_camera.h"
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <iostream>
#include <sstream>

struct MOTOR_PINS
{   int pinEn;
int pinIN1;   int
pinIN2;
};

std::vector<MOTOR_PINS> motorPins =
{
  {12, 13, 15},  //RIGHT_MOTOR Pins (EnA, IN1, IN2)
  {12, 14, 2},  //LEFT_MOTOR  Pins (EnB, IN3, IN4)
};
#define LIGHT_PIN 4

#define UP 1
#define DOWN 2
#define LEFT 3
```

```
#define RIGHT 4
#define STOP 0

#define RIGHT_MOTOR 0
#define LEFT_MOTOR 1

#define FORWARD 1
#define BACKWARD -1

const int PWMFreq = 1000; /* 1 KHz */
const int PWMResolution = 8; const int
PWMSpeedChannel = 2; const int
PWMLightChannel = 3;

//Camera related constants
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26 #define
SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
```

```
const char* ssid     = "MyWiFiCar"; const
char* password = "12345678";
AsyncWebServer server(80);

AsyncWebSocket wsCamera("/Camera");
AsyncWebSocket wsCarInput("/CarInput"); uint32_t
cameraClientId = 0;


const char* htmlHomePage PROGMEM = R"HTMLHOMEPAGE(
<!DOCTYPE html>
<html>
 <head>
 <meta name="viewport" content="width=device-width, initial-scale=1,
maximumscale=1, user-scalable=no">

   <style>
.arrows {      font-
size:40px;
color:red;

   }    td.button {      background-
color:black;      border-radius:25%;
box-shadow: 5px 5px #888888;

   }
   td.button:active {
transform: translate(5px,5px);
box-shadow: none;

   }


   .noselect {
     -webkit-touch-callout: none; /* iOS Safari */
       -webkit-user-select: none; /* Safari */
        -khtml-user-select: none; /* Konqueror HTML */
          -moz-user-select: none; /* Firefox */
```

```
        -ms-user-select: none; /* Internet Explorer/Edge */
user-select: none; /* Non-prefixed version, currently
supported by Chrome and Opera */
   }

   .slidecontainer           {
width: 100%;
   }

   .slider {
     -webkit-appearance: none;
width: 100%;       height:
15px;       border-radius: 5px;
background: #d3d3d3;
outline: none;       opacity: 0.7;
-webkit-transition: .2s;
transition: opacity .2s;
   }

   .slider:hover              {
opacity: 1;
   }

   .slider::-webkit-slider-thumb {
-webkit-appearance: none;
appearance: none;       width:
25px;       height: 25px;
border-radius: 50%;
background: red;       cursor:
pointer;
   }
```

```
.slider::-moz-range-thumb {
```

```
    width: 25px;
height: 25px;
border-radius: 50%;
background: red;
cursor: pointer;
  }

  </style>

 </head>
 <body class="noselect" align="center" style="background-color:white">

  <!--h2 style="color: teal;text-align:center;">Wi-Fi Camera &#128663;
Control</h2-->

  <table id="mainTable" style="width:400px;margin:auto;table-layout:fixed"
CELLSPACING=10>
    <tr>
     <img id="cameraImage" src="" style="width:400px;height:250px"></td>
    </tr>
    <tr>
     <td></td>
     <td class="button" ontouchstart='sendButtonInput("MoveCar","1")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>&#8679;</span></td>
     <td></td>
    </tr>
    <tr>
     <td class="button" ontouchstart='sendButtonInput("MoveCar","3")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>&#8678;</span></td>
     <td class="button"></td>
```

```
      <td class="button" ontouchstart='sendButtonInput("MoveCar","4")' ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8680;</span></td>
     </tr>
     <tr>
      <td></td>
      <td class="button" ontouchstart='sendButtonInput("MoveCar","2")' ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8681;</span></td>
      <td></td>
     </tr>
     <tr/><tr/>
     <tr>
      <td style="text-align:left"><b>Speed:</b></td>
      <td colspan=2>
       <div class="slidecontainer">
         <input type="range" min="0" max="255" value="150" class="slider" id="Speed" oninput='sendButtonInput("Speed",value)'>
       </div>
      </td>
     </tr>
     <tr>
      <td style="text-align:left"><b>Light:</b></td>
      <td colspan=2>
       <div class="slidecontainer">
         <input type="range" min="0" max="255" value="0" class="slider" id="Light" oninput='sendButtonInput("Light",value)'>
       </div>
      </td>
     </tr>
    </table>
```

```
<script>       var webSocketCameraUrl = "ws:\/\/" + window.location.hostname +
"/Camera";       var webSocketCarInputUrl = "ws:\/\/" + window.location.hostname
+ "/CarInput";            var websocketCamera;       var websocketCarInput;


    function initCameraWebSocket()
    {
      websocketCamera = new WebSocket(webSocketCameraUrl);
websocketCamera.binaryType = 'blob';        websocketCamera.onopen
= function(event){};        websocketCamera.onclose   =
function(event){setTimeout(initCameraWebSocket, 2000);};
websocketCamera.onmessage = function(event)

      {
       var       imageId       =       document.getElementById("cameraImage");
imageId.src = URL.createObjectURL(event.data);

      };
    }


    function initCarInputWebSocket()
    {
      websocketCarInput         =         new         WebSocket(webSocketCarInputUrl);
websocketCarInput.onopen    = function(event)

      {
       var speedButton = document.getElementById("Speed");
sendButtonInput("Speed", speedButton.value);          var
lightButton = document.getElementById("Light");
sendButtonInput("Light", lightButton.value);

      };
      websocketCarInput.onclose   =
function(event){setTimeout(initCarInputWebSocket, 2000);};
websocketCarInput.onmessage = function(event){};
```

```
}

function initWebSocket()
```

```
    {
      initCameraWebSocket                                    ();
initCarInputWebSocket();
    }

    function sendButtonInput(key, value)
    {
      var       data       =       key       +       ","       +       value;
websocketCarInput.send(data);
    }

    window.onload = initWebSocket;
document.getElementById("mainTable").addEventListener("touchend",
function(event){         event.preventDefault()
    });
   </script>
  </body>
</html>
)HTMLHOMEPAGE";



void rotateMotor(int motorNumber, int motorDirection)
{
  if (motorDirection == FORWARD)
  {
    digitalWrite(motorPins[motorNumber].pinIN1,                                    HIGH);
digitalWrite(motorPins[motorNumber].pinIN2, LOW);
  }
  else if (motorDirection == BACKWARD)
  {
```

```
   digitalWrite(motorPins[motorNumber].pinIN1,                    LOW);
digitalWrite(motorPins[motorNumber].pinIN2, HIGH);
  }
  else
  {
   digitalWrite(motorPins[motorNumber].pinIN1,                    LOW);
digitalWrite(motorPins[motorNumber].pinIN2, LOW);
  }
}

void moveCar(int inputValue)
{
  Serial.printf("Got value as %d\n", inputValue);    switch(inputValue)
  {

    case UP:
      rotateMotor(RIGHT_MOTOR, FORWARD);
rotateMotor(LEFT_MOTOR, FORWARD);                  break;

    case DOWN:
      rotateMotor(RIGHT_MOTOR,                          BACKWARD);
rotateMotor(LEFT_MOTOR, BACKWARD);
      break;

    case LEFT:
      rotateMotor(RIGHT_MOTOR, FORWARD);
rotateMotor(LEFT_MOTOR, BACKWARD);       break;

    case RIGHT:
      rotateMotor(RIGHT_MOTOR,                          BACKWARD);
rotateMotor(LEFT_MOTOR, FORWARD);
```

```
     break;

    case STOP:
     rotateMotor(RIGHT_MOTOR, STOP);
rotateMotor(LEFT_MOTOR, STOP);          break;

   default:
     rotateMotor(RIGHT_MOTOR, STOP);
rotateMotor(LEFT_MOTOR, STOP);          break;
  }
}

void handleRoot(AsyncWebServerRequest *request)
{                    request->send_P(200,       "text/html",
htmlHomePage);
}

void handleNotFound(AsyncWebServerRequest *request)
{    request->send(404, "text/plain", "File Not Found");
}

void onCarInputWebSocketEvent(AsyncWebSocket *server,
                  AsyncWebSocketClient *client,
AwsEventType type,                  void *arg,
uint8_t *data,                  size_t len)
{                    switch
(type)
  {
   case WS_EVT_CONNECT:
     Serial.printf("WebSocket client #%u connected from %s\n", client->id(), client-
```

```
>remoteIP().toString().c_str());
     break;
   case WS_EVT_DISCONNECT:
     Serial.printf("WebSocket client #%u disconnected\n", client->id());
moveCar(0);       ledcWrite(PWMLightChannel, 0);         break;     case
WS_EVT_DATA:
     AwsFrameInfo *info;      info
= (AwsFrameInfo*)arg;
     if (info->final && info->index == 0 && info->len == len && info->opcode ==
WS_TEXT)
     {
       std::string myData = "";
myData.assign((char *)data, len);
std::istringstream ss(myData);
std::string key, value;         std::getline(ss,
key, ',');         std::getline(ss, value, ',');
       Serial.printf("Key [%s] Value[%s]\n", key.c_str(), value.c_str());
int valueInt = atoi(value.c_str());              if (key == "MoveCar")
       {         moveCar(valueInt);
       }
       else if (key == "Speed")
       {
         ledcWrite(PWMSpeedChannel, valueInt);
       }
       else if (key == "Light")
       {
         ledcWrite(PWMLightChannel, valueInt);
       }           }       break;
case WS_EVT_PONG:
case WS_EVT_ERROR:
```

```
    break;
default:
break;
  }
}

void onCameraWebSocketEvent(AsyncWebSocket *server,
                  AsyncWebSocketClient *client,
AwsEventType type,                  void *arg,
uint8_t *data,                  size_t len)
{                  switch
(type)
  {
   case WS_EVT_CONNECT:
    Serial.printf("WebSocket client #%u connected from %s\n", client->id(), client-
>remoteIP().toString().c_str());
cameraClientId = client->id();
break;    case
WS_EVT_DISCONNECT:
    Serial.printf("WebSocket client #%u disconnected\n", client->id());
cameraClientId = 0;      break;    case WS_EVT_DATA:      break;
case WS_EVT_PONG:    case WS_EVT_ERROR:
    break;
default:
break;
  }
}
```

```
void setupCamera()
{   camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;


  config.frame_size = FRAMESIZE_VGA;
config.jpeg_quality = 10;   config.fb_count
= 1;


  // camera init   esp_err_t err =
esp_camera_init(&config);   if (err !=
ESP_OK)
  {
```

```
    Serial.printf("Camera init failed with error 0x%x", err);
                                                        return;
  }

  if (psramFound())
  {
   heap_caps_malloc_extmem_enable(20000);
   Serial.printf("PSRAM initialized. malloc to take memory from psram above this
size");
  }
}

void sendCameraPicture()
{    if (cameraClientId ==
0)
  {
return;
  }      unsigned long    startTime1 =
millis();
  //capture a frame   camera_fb_t * fb =
esp_camera_fb_get();   if (!fb)   {
    Serial.println("Frame buffer could not be acquired");
return;
  }

  unsigned long  startTime2 = millis();
wsCamera.binary(cameraClientId, fb->buf, fb->len);
esp_camera_fb_return(fb);

  //Wait for message to be delivered  while
(true)
```

```
  {
    AsyncWebSocketClient * clientPointer = wsCamera.client(cameraClientId);
    if (!clientPointer || !(clientPointer->queueIsFull()))
    {
break;     }
delay(1);
  }

  unsigned long  startTime3 = millis();
  Serial.printf("Time taken Total: %d|%d|%d\n",startTime3 - startTime1, startTime2 -
startTime1, startTime3-startTime2 );
}

void setUpPinModes()
{
  //Set up PWM   ledcSetup(PWMSpeedChannel, PWMFreq,
PWMResolution);   ledcSetup(PWMLightChannel, PWMFreq,
PWMResolution);

  for (int i = 0; i < motorPins.size(); i++)
  {
    pinMode(motorPins[i].pinEn, OUTPUT);
pinMode(motorPins[i].pinIN1, OUTPUT);     pinMode(motorPins[i].pinIN2,
OUTPUT);

    /*  Attach  the  PWM  Channel  to  the  motor  enb  Pin  */
ledcAttachPin(motorPins[i].pinEn, PWMSpeedChannel);
  }
  moveCar(STOP);

  pinMode(LIGHT_PIN, OUTPUT);     ledcAttachPin(LIGHT_PIN,
PWMLightChannel);
```

```
}
```

```
void setup(void)
{   setUpPinModes();
Serial.begin(115200);

  WiFi.softAP(ssid, password);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);

  server.on("/", HTTP_GET, handleRoot);
server.onNotFound(handleNotFound);

  wsCamera.onEvent(onCameraWebSocketEvent);
server.addHandler(&wsCamera);

  wsCarInput.onEvent(onCarInputWebSocketEvent);
server.addHandler(&wsCarInput);

  server.begin();
  Serial.println("HTTP server started");

  setupCamera();
}

void loop()  {
wsCamera.cleanupClients();
wsCarInput.cleanupClients();
sendCameraPicture();
  Serial.printf("SPIRam Total heap %d, SPIRam Free Heap %d\n",
ESP.getPsramSize(), ESP.getFreePsram());
```

```
}
```