



An-Najah National University Faculty of Engineering
Electrical Engineering Department

Wireless Emergency robot and Assistant with arm

By:

Eng. (Wael Jaber)

Eng. (Hala Jaber)

Eng. (Obada Abulail)

Supervisor

Dr. Haneen Al-utt

**Presented in Partial fulfillment of the requirements for Bachelor degree in
'Electrical Engineering '**

The 18 Of may, 2022

Acknowledgment

In 18 years of studying, we had much experiences, we had been affected be many people, friends and teachers. To all of them who helped us to reach this place.

To our parents,,, for pushing us forward every time, for giving hope and prayers when we needed it, and moreover when they hear our name, for teaching us the first letters our had spoken, until our graduation and after that we will always learn from them.

To our doctors and teachers in the bachelor degree study, especially doctor Haneen Al-utt, who had helped us in many ways, to become a better engineers and a good researchers.

To everyone,,, thank you for helping us to reach this place, we are an output for many inputs and factors, which us had made. we will always remember you. And thank you!

Wael jaber

Hala jaber

Obada abulail

Contents

<i>Acknowledgment</i>	<i>I</i>
<i>Table of figures.</i>	<i>IV</i>
ABSTRACT	V
CHAPTER 1. INTRODUCTION	1
1.1 OVERVIEW	1
1.2 EXISTING PROBLEMS	1
1.3 AIMS AND OBJECTIVES	2
1.4 SCOPE OF THE WORK	2
1.4.1 GENERAL OVERVIEW	2
1.4.2 HARDWARE OVERVIEW	3
1.4.3 SOFTWARE OVERVIEW	4
1.5 IMPORTANCE OF THE WORK	4
CHAPTER2. Theoretical Background and Previous Work	5
CHAPTER 3. METHODOLOGY	7
3.1 SOFTWARE IMPLEMENTATION	7
3.1.1 HTML	Error! Bookmark not defined.
3.1.2 CSS	Error! Bookmark not defined.
3.1.3 ARDUINO LANGUAGE	7
3.1.4 C++ language	7
3.2 SOFTWARE DESIGN.....	8
3.2.1 SCHEMATIC.....	8
3.2.2 MOVING CONTROL IN ALL DIRECTIONS	8
3.2.3 MOVING CONTROL OF ROBOT ARM IN ALL DIRICTIONS.....	9
3.3 HARDWARE DESIGN	10
3.3.1 ESP32CAM	10
3.3.2 ESP32	12
3.3.3 BUZZER	13

3.3.4	four SERVO MOTORS	13
3.3.5	DC GEARED MOTOR	14
3.3.6	DRIVERS FOR THE MOTORS.....	14
3.3.7	BATTERY.....	15
3.3.8	ROBOT CHASSIS	15
3.3.9	ROBOT ARM.....	16
3.3.10	DC-DC Buck Convertor.....	16
CHAPTER 4. CONSTRAINTS, STANDARDS AND EARLIER COURSEWORK.....		17
4.1	<i>CONSTRAINS</i>	17
4.2	<i>STANDARDS</i>	17
4.3	<i>EARLIER COURSEWORK</i>	18
	<u>HARDWARE SCHEMATICS (1)</u>	19
	<u>HARDWARE SCHEMATICS (2)</u>	19
	<i>Real Robot(had been bult)</i>	20
CHAPTER 5. THEORITICAL EQUATION.....		21
CHAPTER 6. Result, Conclusions and Recommendation.....		222
6.1	TOTAL CONCLUSION.....	222
6.2	PROJECT RESULT.....	22
6.3	SECURITY	233
6.4	WHERE SUCH A PROJECT CAN BE USED	233
6.5	RECOMMENDATIONS:.....	233
CHAPTER 7. REFERENCES		244
CHAPTER 8. APPWNDICES.....		25

Table of figures

symbol	name	page
Figure 1 4-1-1	Shows High Level Overview of robot vehicle (block diagram system)	3
Figure 3-2-2-1	Shows the flow chart of the moves of project	8
Figure 3-2-3-1	Shows the flow chart of arm moves	9
Figure 3-3-1-1	ESP32CAM module	10
Figure 3-3-1-2	datasheet for ESP32CAM module	11
Figure 3-3-2-1	ESP32 module	12
Figure 3-3-2-2	datasheet of ESP32	12
Figure 3-3-3-1	Buzzer	13
Figure 3-3-4-1	Servo gripper	13
Figure 3-3-4-2	MG996R servo	13
Figure 3-3-5-1	DC gear motor	14
Figure 3-3-6-1	DC motor Driver	14
Figure 3-3-7-1	Battery	15
Figure 3-3-8-1	Robot Chassis	15
Figure 3-3-9-1	Robotic Arm	16
Figure 3-3-10-1	DC-DC Buck Converter	16
Figure 4-1	Hardware schematics(esp32+bridge+two servo)	19
Figure 4-2	Hardware schematics (ESP32+servo)	19

ABSTRACT

Our report is for presenting two technology which when combine together can form and tremendous useful devices for defense system and ultimately society because robot is an unique technology to serve society in different application and in different field so we all knows important of robot now days in society and developing technology the number of robots Used worldwide is constantly increasing.

They are more and more present in different workplaces such as, dangerous areas, processing operations,, medical environments, military, manufacturing inaccessible areas etc... ..again we have one unique communication technology now that is IOT this report will explain mostly IOT technology because IOT is not only just an technology it is an is phenomenal ...Now the Internet is being used to connect various objects such as cars, sensors, controllers, TVs, machinery, transport containers and electrical appliances, creating the Internet of Things (IOT). The networked and user interfaced robots, such as rescue robots, human assisting robots, health care robots and robots for military applications The evolution and growth of the Internet because a technology we can control robot with high distance, high speed and high accuracy. , solution on all communication and controlling over internet which is totally wireless and we can optimistically look forward to an IOT-assisted world that is connected, smarter – and better .

This project report describes the robot had been built using two microcontrollers (esp32cam+esp32) and how it can be controlled and powered in the main tuning condition.

Also, in this project wireless emergency robot and assistant with arm technology had been used to implement our vision on our robot, it was controlled from a mobile application , to move it or to get some objectives transferred from allocation to another with an access on the camera mounted on it, and the user had been able to reach safety movement restrictions, and also using the arm to do some dangerous thing and assistant on behalf of humans.

So, the microcontrollers had been used to control servo motors, DC gear motors, and to view the camera.

Webpage and control page of Robot will be built using HTML, CSS and Arduino language. And will be used to send the command from mobile app to the microcontrollers over the cloud

1 CHAPTER 1. INTRODUCTION

1.1 OVERVIEW

This project is great for companies and civil defense which are trying to make robots do the hard job in good efficiency, high ability and to be controlled at any situation or under any hard impact on it. So, this is an application that solves a newly appeared challenge by using trending technology, to save lives, and using this technology to make people's lives easier.

The robot project consisted of two main parts, the first one is the robot chassis that will carry our all system and the robotic arm that consists from: the chassis which hold all the equipment above or in, so the chassis should be strong and has many joints to do its job in free movement, motors which apply motion and instruction for movement, , esp32camera module, esp32 Wi-Fi and more had been connected all on the chassis. While the second part is the controller to control the robot, the remote controller is mobile application connected on Wi-Fi via IP to control the robot from any place showing all the connected devices of the robot on upper arm grip.

1.2 EXISTING PROBLEMS

Today's technology is moving forward really quickly in fields like programming, micro process building and architecting and communication technologies. So, we have a demanding need to raise our hardware system to a new level to maximizing the benefits coming from the newly technologies.

And in this pandemic time, we are living and we have another demand need to build a system can be used in unusual conditions with a property of low cost, easy use and high efficiency.

1.3 AIMS AND OBJECTIVES

Mainly, this project aims to reducing human losses and providing comfort and assistance to people and to do the overall danger work so it will be with controlled by humans. The objectives differ according to the kind of the job that require a secure sensitive, monitoring people in an area and giving some commands, or like mines detectors etc. This robot meant to be the base idea and the costumer can use this project for anything he wants by requesting other fellows, like sensors or motors or interfacing devices.

Besides that, this project which use internet of robotics things, is one of the firsts in this area, and it have a very shining future.

1.4 SCOPE OF THE WORK

1.4.1 GENERAL OVERVIEW

- A chassis moving with all its contents to get a robot vehicle sub-system (see Figure 1.4.1.1 for a high-level overview).
- The costumer sends the command over the wireless cloud using mobile to the two microcontrollers.
- ESP32 responds by moving 4 servos on order.
- The esp32camera will responds by moving dc motors and get the live stream.
- The power supply calculated to operate them as well as possible without any over current or voltage.

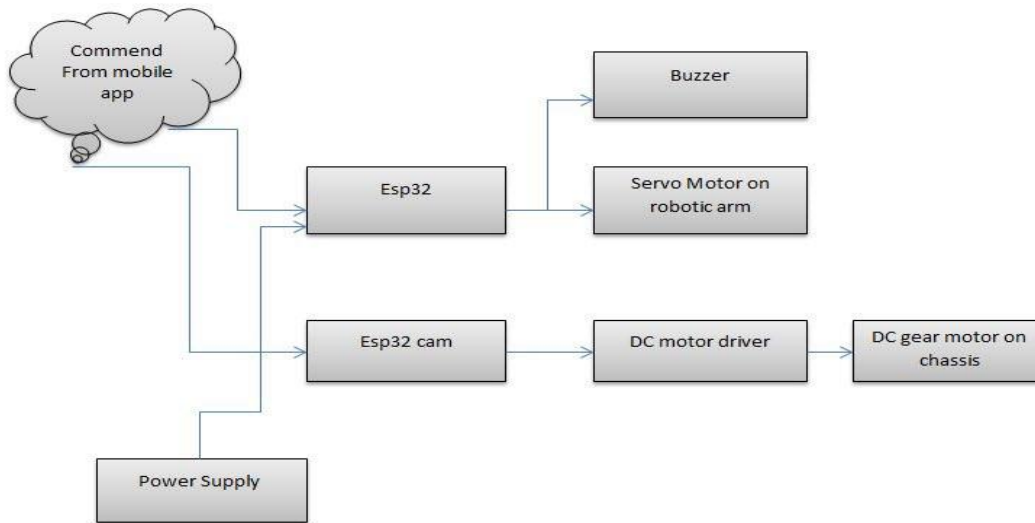


Figure 1-4-1-1 Shows High Level Overview of robot vehicle (block diagram system)

1.4.2 HARDWARE OVERVIEW

- ESP32 Wi-Fi to receive and send commands via Wi-Fi.
- Camera module is the new thing added to a perfect project to let you see where your robot vehicle is going and what catches the arm.
- Buzzer for alarm and warning or advice messages for the nearby to be alert.
- DC motors main parts for the vehicle to get motion on the surface through a motor driver for speed and direction control.
- Servo motors that could help our project to get new features in one project that's the arm for robot to get stick with something to get something hold from the ground and so.
- Lithium battery to feed all circuits and motors.

1.4.3 SOFTWARE OVERVIEW

This project can execute commands via (esp32cam and esp32) at the same time they are issued from the mobile program via Wi-Fi.

These two pieces are a small computer with arduino programming language so the arduino is object- oriented programming language that mean the real-time tasks for the sequential reading and sending of each communication with camera stream and for motors controls.

These two pieces can be opened from laptop and controlled with some software like ARDUINO-1.8.16 that allow access to the two pieces from laptop.

Also, there are software overviews explanations found later in this report to get access to final robot vehicle project.

1.5 IMPORTANCE OF THE WORK

The importance of this project is having a lot of uses that helps humans in doing dangerous or important work through a robot, so that the robot makes the risky tasks easier and cheaper, and can be done in other ways, not only the regular one. Also, this robot can be a side of luxury, as cleaning robot for example, and it can be used in industrial applications, alongside transferring, and organizing products, as we had already said.

The inspiration of this project come from both technology and economy visions. This new technology is setting sail right now in many applications, there is now a use for this technology in medical application with something called internet of bio things, and in smart farming systems for automatic irrigation systems and diseases detectors, also in smart homes application for cheaper and more effective components, so this project can be customized to suit any one of this applications, from cleaning houses robots, to harvesting ones to surgeries robotics and so on, all of this things starts here.

In this project we had to search for previous applications with similar technology, actually we some applications had been found and from recent times because it's a new technology, the most related one was published in **National Institute of Electronics and IT, Aurangabad, India.** [4],

on Dec-2016 by Eng. Pravin F Rane and two others, who build an internet of things robot based, using the arduino uno and GSM and a C++ program languages, a simple robot which can move in the four ways, and with a camera for monitoring only, with very simple robotic arm just used to cut objects for example. Anyway, his robot was built by laser cutting of a piece of light wood, so it's also not designed for any kind of heavy duties, after all of this he said in his conclusion and I quote:

“For bomb disposing robot we can use an basic cutting and wire cutting arm with an forward ,reverse ,left ,right moving car so that it can move in dangerous area according to requirement ,again it have an video camera for shoving up real time image for further controlling..”

This means that now we have a chance to develop this project using modern tools like a esp32cam and esp32 and develop the arm to be 4dof for more free movement to get better results with less efforts.

And before going to another application with this technology we prefer to check if the microcontrollers are capable to handle it, so we will call another paper about **Performance Analysis of Microcontrollers Used In Iot Technology**[5][6][7],

we will take a view on the microcontrollers types we are willing to use, so we are willing to use esp32 to control servos, about it the paper says that this microcontroller has a 240 MHz clock, which is software limited by default to 120 MHz. This software clock limiting serves to reduce power consumption and carve out some time for the built-in TCP/IP stack, which is often relied upon to provide asynchronous I/O via the built-in 802.11b/g/n WiFi transceiver., so we can't use it for anything more than servos in IOT application if we wanted a high performance, the other micro controller we want to use is esp32cam, and the paper says and I quote:

“Arduino and Raspberry Pi do not have built-in support for wireless networks. Developers will have to add a Wi-Fi or cellular module to the board and write code to access the wireless module.

So open source IoT development board called esp32 is used and allows you to code your device using Lua. scripts. One of its most unique features is that it has built-in support for Wi-Fi connectivity.”

Moreover, we know for esp32cam that its clock speed is Up to 240MHz, with 520 KB SRAM, ROM size 448 KB memory, and can be programmed by many languages including arduino which is an easy new language.

Now we will take a look on an application which is IOT Based Camouflage Surveillance Robot [8]

which developed IoT enabled system to defence, Camouflage Robot is efficient and effective in area of strength, calculations, capacity and capabilities. The main intention behind the Camouflage Robot is to minimize the number of humans loses in the military operations, sting operations, terrorist attacks and many such operations which results in human death. Camouflage Robot is operated using seven phases.

And the results were and I quote:

“The Camouflage Robot can be used to replace soldier in war fields for performing high frequency, accuracy, repetitive and risky jobs.”



3.1 SOFTWARE IMPLEMENTATION

It required a Programming Tools of mobile application & arduino Language to build a project.

3.1.1 HTML

Hypertext Mark-up Language, commonly abbreviated as HTML, is the standard mark-up language used to create web pages. Along with CSS, and arduino language, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile. [9]

3.1.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a language. Although most often used to set the visual style of web pages and user interfaces written in HTML. I used HTML and CSS both for the design of website and web page to control the Robot .[10]

3.1.3 ARDUINO LANGUAGE

Arduino programming language can be divided in three main parts: functions, values (variables and constants), and structure.

Functions: For controlling the Arduino board and performing computations.

Variables: Arduino data types and constants.

Structure: The elements of Arduino (C++) code.

3.1.4 C++ language

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

3.2 SOFTWARE DESIGN

3.2.1 SCHEMATIC

The figures 1.4.1.1-2 displays all process that send data from (mobile app) to server to (microcontrollers).

3.2.2 MOVING CONTROL IN ALL DIRECTIONS

The flow chart of this process:

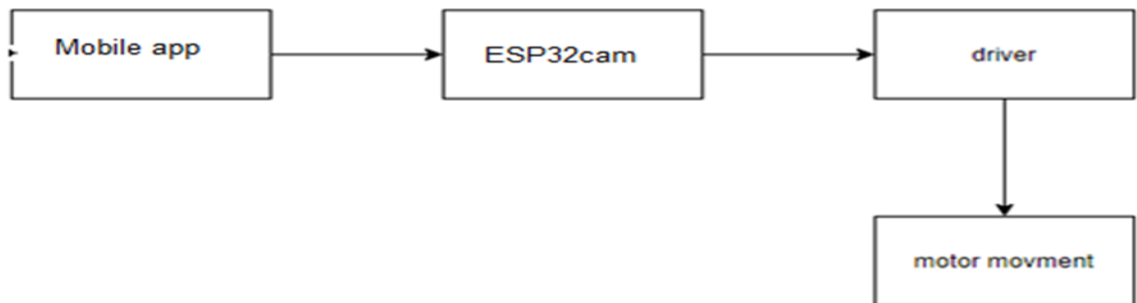
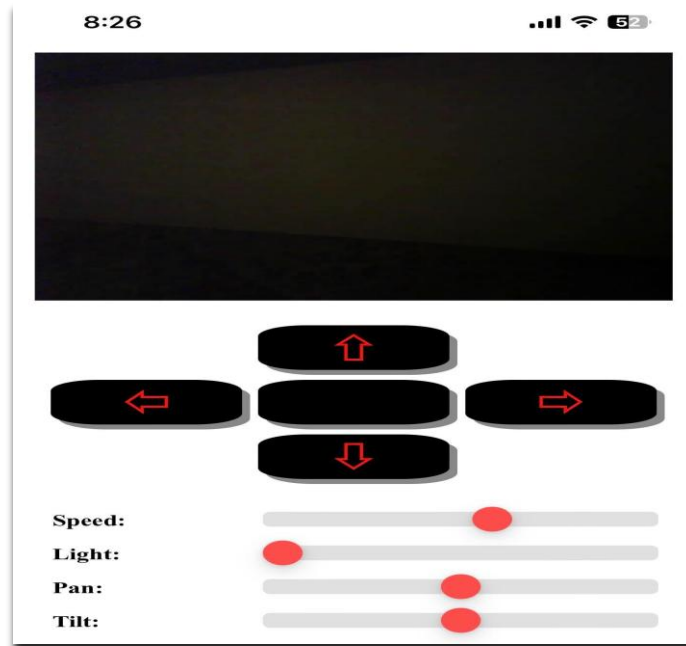


Figure 33-2-2-1 Shows the flow chart of the moves of project

The command would be sent from the client to esp32cam, to driver for move the dc motors as we see . *see code in Appandices.



3.2.3 MOVING CONTROL OF ROBOT ARM IN ALL DIRICTIONS

The flow chart of this process:

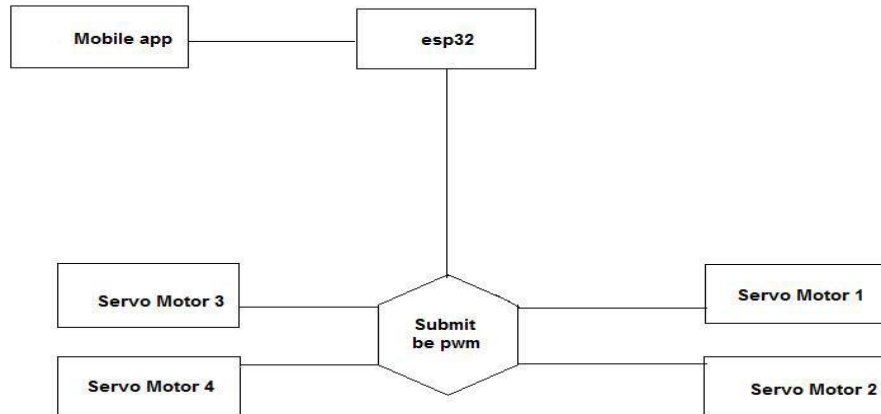
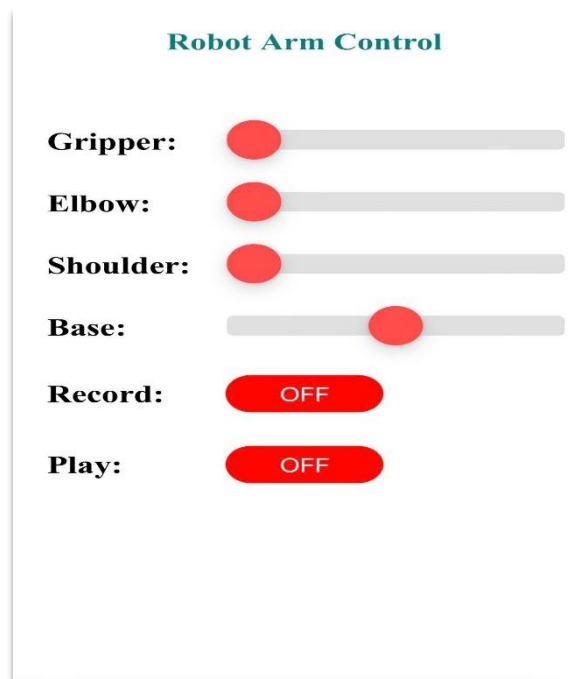


Figure 33-2-3-1 Shows the flow chart of arm moves

The command would be sent from the client, to esp32 and pass the command to servo motors to move in specified directions that client requested as we see.

*see code in Appandices.



3.3 HARDWARE DESIGN

3.3.1 ESP32CAM

- ESP32CAM, to receiving orders from the customer via mobile app with Wi-Fi....and processing and implementation it.
- The board integrates WiFi, traditional Bluetooth and low power BLE , with 2 highperformance 32-bit LX6 CPUs. It adopts 7-stage pipeline architecture, on-chip sensor, Hall sensor, temperature sensor and so on, and its main frequency adjustment range from 80MHz to240MHz.
- Fully compliant with WiFi 802.11b/g/n/e/i and Bluetooth 4.2 standards, it can be used asa master mode to build an independent network controller, or as a slave to other host MCUs to add networking capabilities to existing devices
- ESP32-CAM can be widely used in various IoT applications. It is suitable for homesmart devices, industrial wireless control, wireless monitoring, QR wirelesidentification, wireless positioning system signals and other IoT applications. It is anideal solution for IoT applications.

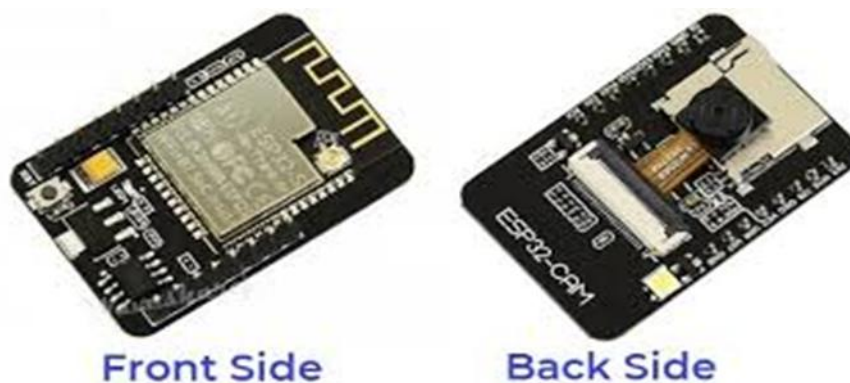


Figure 33-3-1-1 ESP32CAM module

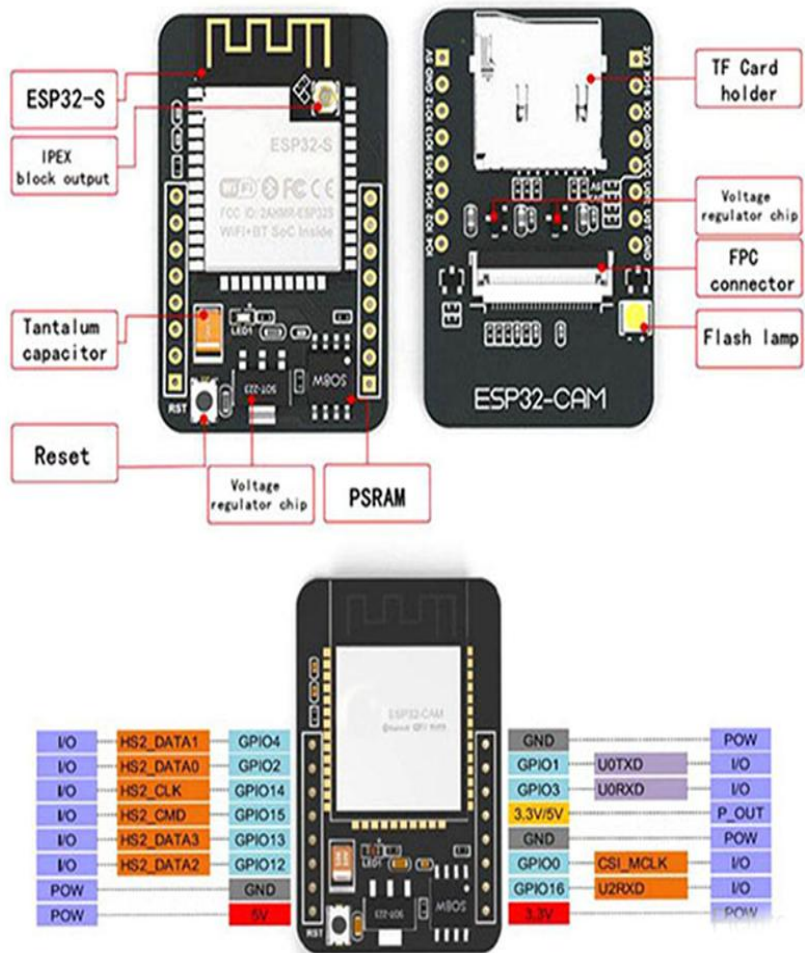


Figure 33-3-1-2 datasheet for ESP32CAM module

3.3.2 ESP32 WIFI

To receiving orders from the customer via mobile app with Wi-Fi...
and implementation it.



Figure 33-3-2-1 ESP32 module

- ESP32 is a low-cost, low-power system on a chip (SoC) series with Wi-Fi & dual-mode Bluetooth capabilities, At its heart, there's a dual-core or single-core Tensilica Xtensa LX6 microprocessor with a clock rate of up to 240 MHz. ESP32 is highly integrated with built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. Engineered for mobile devices, wearable electronics, and IoT applications, ESP32 achieves ultra-low power consumption through power saving features including fine resolution clock gating, multiple power modes, and dynamic power scaling..

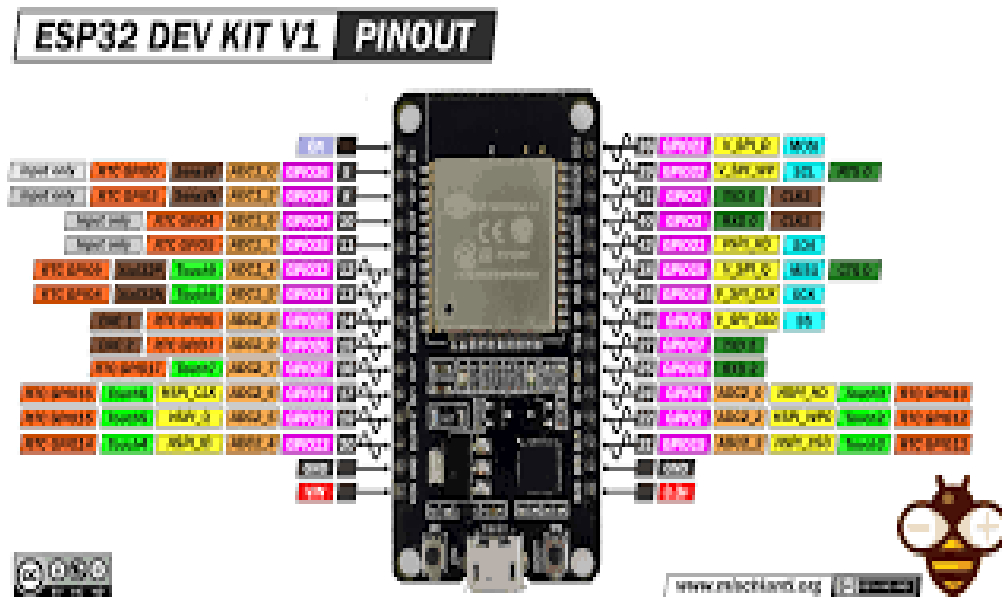


Figure 33-3-2-2 datasheet of NodeMcu

3.3.3 BUZZER

we will use this speaker to interact with the surrounding people or as a horn like cars.

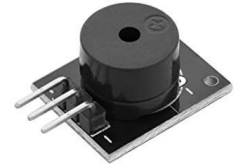


Figure 33-3-3-1 Buzzer

3.3.4 FOUR SERVO MOTORS

- I. The first one is the connected to the gripper one to hold the objects.



Figure 33-3-4-1 Servo gripper

- II. The other ones Towers pro Servos MG996R model to support the wight of objectives.



Figure 33-3-4-2 MG996R servo

3.3.5 DC GEARED MOTOR

To move the whole robot on the desired velocity we need with or without a load.



Figure 3-3-5-1 DC gear motor

3.3.6 DRIVERS FOR THE MOTORS

To control the movement of robot chassis, motor driver with four GPIO ports will be used.



Figure 33-3-6-1 DC motor Driver

3.3.7 BATTERY

i. we use : 18650 rechargeable Li-ion batteries had been used, three in series which supposed to give 12 volts and it gave 12.4, and it was supposed to supply with 9600 mA but its peak value was 2600 mA nearly.



Figure 33-3-7-1 Battery

3.3.8 ROBOT CHASSIS

The chassis should have the ability to handle the weight of all objectives, and do to be steady in many areas.



Figure 33-3-8-1 Robot Chassis

3.3.9 ROBOT ARM

We have chosen the arm to be flexible, strong metal and low in weight to reduce the current drawn by the servos . so, we chose this arm as it has 4 joints that move 180 degrees.

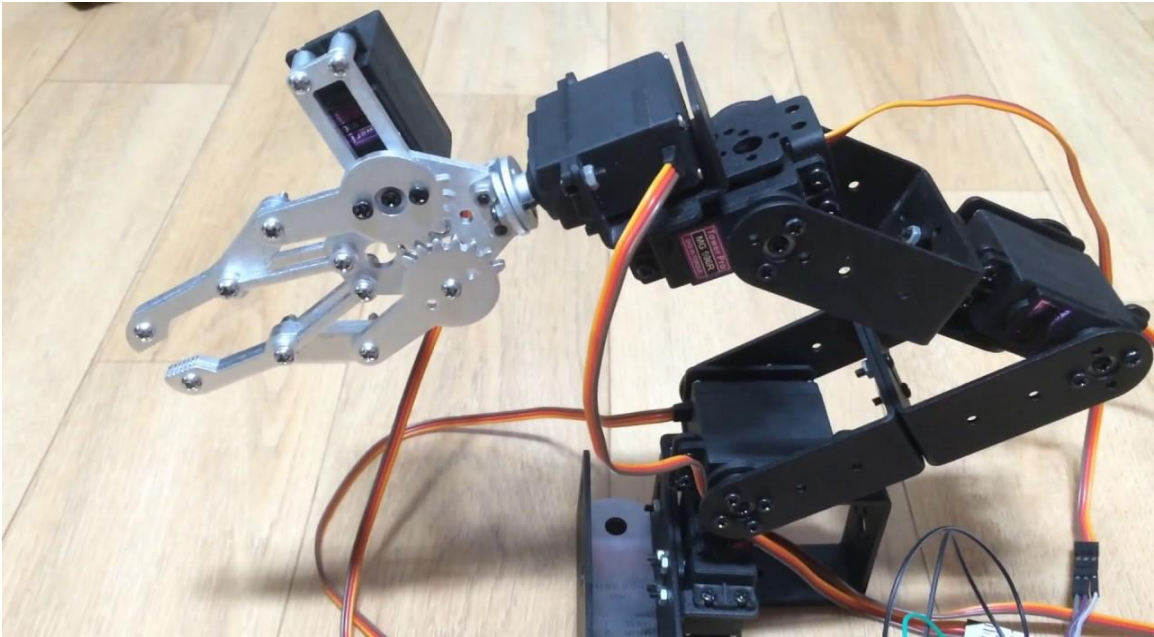


Figure 33-3-9-1 Robotic Arm

3.3.10 DC-DC Buck Converter

To reduce the source voltage to 6v so match the operation of the esp32 and servo motors.



Figure 33-3-10-1 DC-DC Buck Converter

4 CHAPTER 4. CONSTRAINTS, STANDARDS AND EARLIER COURSEWORK

4.1 CONSTRAINTS

- On economy: the budget limitation was the biggest economical constrain, due to the cut of salaries and covid-19 special situation it was pretty hard to obtain the needed amount of money which was pretty big because our market is restricted to two names of suppliers.
- On society: this project counted by many as a way to reach deeply in people's daily life.
- On Manufacturability: the design implemented physically but not as it meant to be due to lack of hardware components.

4.2 STANDARDS

Mainly IEEE standards used in Palestinian projects, due to the lack of information in studying about standards and protocols, and if we wanted to move to industrial form in this design we will use this protocols: At first, **IEEE 802.15.4s-2018** will be used, This standard defines a protocol and procedures for Definitions of MAC related functions to enable spectrum resource management are addressed in this amendment to IEEE Std 802.15.4™. It specifies the following: – Spectrum resource measurements and network performance metrics, such as packet error ratio, delay, etc. – Information elements and data structures to capture these measurements, – Procedures for collecting and exchanging spectrum resource measurement information with higher layers or other devices. [1]

IEEE 802.15.4 is a standard which specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs). It is maintained by the IEEE 802.15 working group. It is the basis for the ZigBee, ISA100.11a, Wireless HART, and MI WI specifications, each of which further extends the standard by developing the upper layers which are not defined in IEEE 802.15.4. Alternatively, it can be used with 6LoWPAN and standard Internet protocols to build a wireless embedded Internet.[2]

IEEE 802.11 uses various frequencies including, but not limited to, 2.4 GHz, 5 GHz, 6 GHz, and 60 GHz frequency bands. Although IEEE 802.11 specifications list channels that might be used, the radio frequency spectrum availability allowed varies significantly by regulatory domain.

[3]

4.3 EARLIER COURSEWORK

In our years in the university we studied different courses, and now as graduation project students we can tell that not all of that courses suit as in earlier jobs, our courses which focus on theories weren't very helpful in our project, not like that the practical and programming ones, after we suggest in this stage to use dual learning which connect both sides in more effective way, and to increase courses which help students to think properly using problem solving methods and the cognitive factors to help them in the self-learning courses and to solve problems like engineers.

Since our study is in English, we got used to make a good report and do a good presentation, and the English courses helped us to improve our capabilities.

Electronics courses helped us to understand power electronics and drive, which we -like- used in our project as a concept on a small scale.

Programming languages we took as out courses and online courses with microcontroller course helped us to think correctly and to build our system as good as we can.

Sensors and Measurements helped us to understand how our system should monitor the surrounding of our project and if our sensor systems are possible to build.

HARDWARE SCHEMATICS(1)

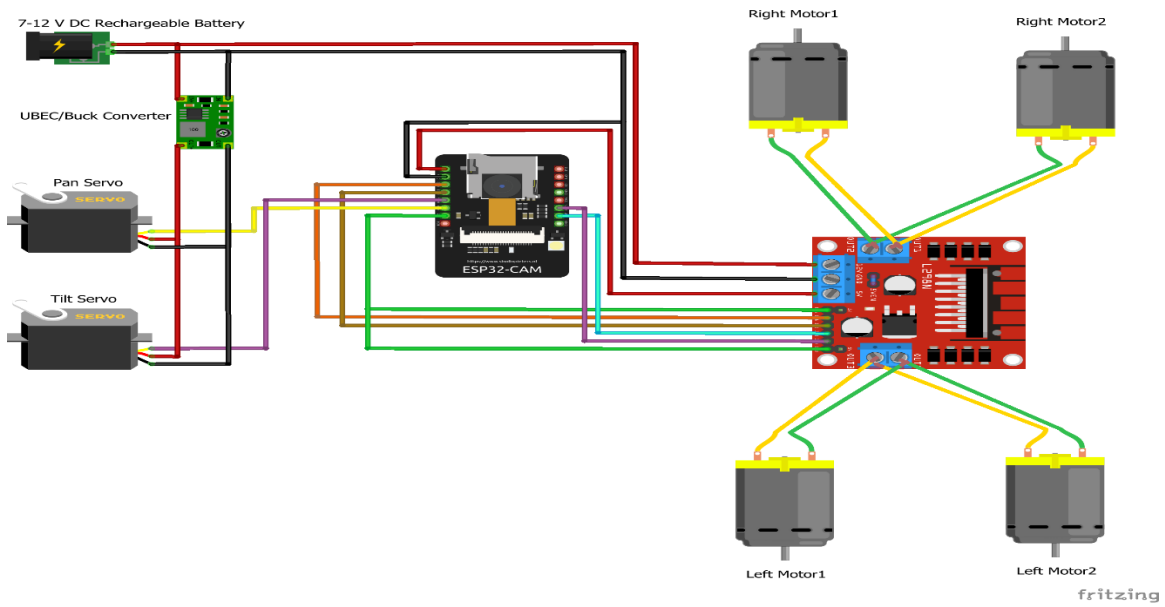


Figure 4-1 Hardware schematics(esp32+bridge+two servo)

HARDWARE SCHEMATICS (2)

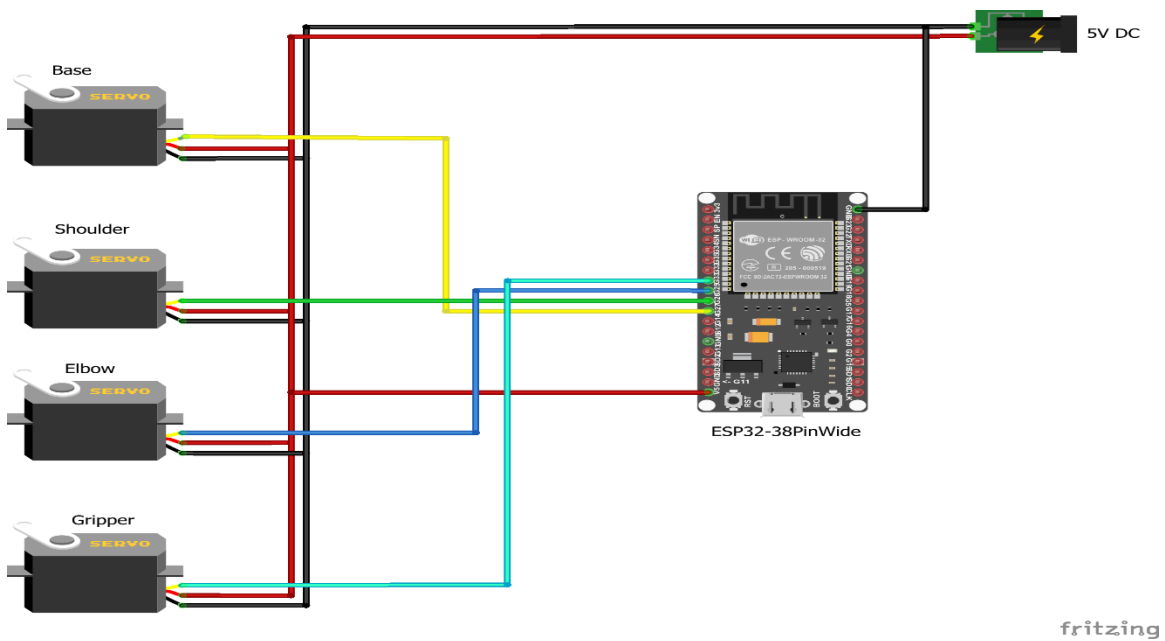


Figure 4-2 Hardware schematics (esp32+servo)

Real robot (had been built)



5 CHAPTER5. THEEORITICAL EQUATION

We need to make all calculation in worse case, so all devices will consume max power:

- First: we have 4 servo motors for arm, each one consume about (0.6A/6v) in max torque at 13kg/m .
so we need in this part about (0.6*4) = 2.4A.
Knowing that it is difficult to make all servos in max load in this project because we control each servo individually so the real power need less than 2.4

- Second: we have two DC motor to drive the vehicle each one consume 1.2A at max load so we need about (1.2*2) = 2.4A.
Knowing that in this project we exclude driving the vehicle and the arm at same time to avoid high current consumption.

- Third : we have two microcontroller { esp32cam and esp32 }:

-Esp32cam : when we turn it + turn on the flash at maximum brightness power consumption =0.3A.

-Esp32: need 0.1A.

So (0.3 + 0.1) = 0.4A.

OVER ALL CURRENT = 2.4 + 0.4 = 2.8 A.

- So finely we have chosen to use three lithium battery each one have (4.1v/3A)
And connected it in series to have about (12.4v/3A).
- We know that the DC motor driver working at 12v so we can connect it directly with the battery, and the servos and microcontrollers need 6v to operating it so we chose to use Dc buck converter to reach to this voltage .

6.1 TOTAL CONCLUSION

In this project self-learning courses and courses which are unfamiliar to electrical engineering students (until now) had been applied, critical thinking methodology and SCAMPER technique had been used to gain the optimum design within market and budget capabilities. Which showed that the team who had done this project is open for new technologies and to adapt these technologies over the courses they had learned on the studying of bachelor degree.

By Using internet of things, our possibilities of controlling and monitoring the systems had been enhanced, and this enhancement can be mirrored on other systems, from power grids, to smart cities, even to brand new elevators, this technology can now be used to improve other technologies, and this was the main goal of selecting this project.

By learning how to build a multi-task robot, a system of sensors and drivers had been applied which act nearly as an integrated system, also by trying this application our minds have opened widely on how to use this application and other ones in a commercial way, which is a very important criteria for engineers that they usually didn't learn in college studies.

This robot is a general form of internet of things programmed robot (IoRT), which can be customized for any application we want.

6.2 PROJECT RESULT

In this project the final system was slightly close to the designed one in the first seminar project, full movement control, live video stream of the system, light control and easy end-user interfacing had been achieved.

We have just two changes from the first seminar project, and they are:

- 1- We lowered the number of dof for robot arm by survey two servos to reach 4dof and that because the dc buck converter cannot pass above 3A and to make 6dof we need at least 3.4 A .
- 2- We have replaced nodemcu module with ESP32 because it is more practical.

After all, the system worked properly well with a low budget in compare with the same project applied without internet of things, yields that we have like a green light to proceed with o project with a high possibility of success.

6.3 SECURITY

This project actually is not fully secured one, since its online on a free server with simple protection, without a cyber security protocol, hacking the system and controlling it would be too easy for any one nearly.

The esp32 and esp32cam have a WI-FI password and dynamic IP so they have to be programmed with a certain library to make the IP fixed for this application locally.

6.4 WHERE SUCH A PROJECT CAN BE USED

- I. Smart homes applications: as cleaning robot, a robot to move stuff, garden coordinator and so on.
- II. Smart cities: to clean the streets, coordinate the public gardens, with Civilian defense forces as fire indicators or fire fighter (extinguisher holder), to give the government an eye on the streets any time with possibility to indicate abnormal voices or abnormal movements, and as a delivery robot.
- III. Disastrous: as in covid-19 pandemic we can use these robots to sterilization the streets and public places or to give orders to people and to make sure that there are no crowds, and in highly polluted areas this robot can save lives money and time.
- IV. Medical usage: some hospitals now puts in respect using robots in surgeries
- V. Industrial applications: like production lines or arranging products workers.

6.5 RECOMMENDATIONS:

- I. Using wifi password and static IP will move this project to a professional level in security and widely applications, and it would make the difference between it and any local system tangible.
- II. In our project the device is always on standby situation. We recommend using another chip as an enable for similar applications to save the power and to give durability to the system. The design depends on the application, in homes for example it would do the job but in systems dependent on limited power sources enable device should be used.
- III. We recommended instead of using microcontrollers using specially designed passive chips to imply the system with the lower possible cost, we suggest using a chip for each independent sub-system to achieve the final integrated system.

7 CHAPTER 7. REFERENCES

<https://standards.ieee.org/ieee/802.15.4s/5970/> [1]

https://en.wikipedia.org/wiki/IEEE_802.15.4 [2]

https://en.wikipedia.org/wiki/IEEE_802.11 [3]

https://www.academia.edu/34469152/IOT_ROBOT_WITH_BOMB_DIFFUSING_APPLICATION [4]

https://www.academia.edu/37082495/Performance_Analysis_of_Microcontrollers_Used_In_Iot_Technolog%20y [5]

https://www.waveshare.com/wiki/E-Paper_ESP32_Driver_Board [6]

https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf [7]

https://www.academia.edu/49653690/IOT_Based_Camouflage_Surveillance_Robot [8]

<https://en.wikipedia.org/wiki/HTML> [9]

<https://en.wikipedia.org/wiki/CSS> [10]

- FOR ESP32CAM :

```
#include "esp_camera.h"
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <iostream>
#include <sstream>
#include <ESP32Servo.h>

#define PAN_PIN 14
#define TILT_PIN 15

Servo panServo;
Servo tiltServo;

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

std::vector<MOTOR_PINS> motorPins =
{
    {2, 12, 13}, //RIGHT_MOTOR Pins (EnA, IN1, IN2)
    {2, 1, 3}, //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,
maximum-scale=1, user-scalable=no">
    <style>
      .arrows {
        font-size:30px;
        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">
```

```
<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","4")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>#8679;</span></td>
        <td></td>
    </tr>
    <tr>
        <td class="button" ontouchstart='sendButtonInput("MoveCar","2")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>#8678;</span></td>
        <td class="button"></td>
        <td class="button" ontouchstart='sendButtonInput("MoveCar","1")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>#8680;</span></td>
    </tr>
    <tr>
        <td></td>
        <td class="button" ontouchstart='sendButtonInput("MoveCar","3")'
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>

```

```

<tr>
  <td style="text-align:left"><b>Pan:</b></td>
  <td colspan=2>
    <div class="slidecontainer">
      <input type="range" min="0" max="180" value="90" class="slider"
id="Pan" oninput='sendButtonInput("Pan",value)'>
    </div>
  </td>
</tr>
<tr>
  <td style="text-align:left"><b>Tilt:</b></td>
  <td colspan=2>
    <div class="slidecontainer">
      <input type="range" min="0" max="180" value="90" class="slider"
id="Tilt" oninput='sendButtonInput("Tilt",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)
    {
        sendButtonInput("Speed", document.getElementById("Speed").value);
        sendButtonInput("Light", document.getElementById("Light").value);
        sendButtonInput("Pan", document.getElementById("Pan").value);
        sendButtonInput("Tilt", document.getElementById("Tilt").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);
    panServo.write(90);
    tiltServo.write(90);
    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);
        }
        else if (key == "Pan")
        {
            panServo.write(valueInt);
        }
        else if (key == "Tilt")
        {
            tiltServo.write(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_4;
config.ledc_timer = LEDC_TIMER_2;
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-startime2 );
}

void setUpPinModes()
{

```

```

panServo.attach(PAN_PIN);
tiltServo.attach(TILT_PIN);

//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());
}

```

- FOR ESP32 :

```

#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>

#include <ESP32Servo.h>
#include <iostream>
#include <sstream>

struct ServoPins
{
  Servo servo;
  int servoPin;
  String servoName;
  int initialPosition;
};
std::vector<ServoPins> servoPins =
{
  { Servo(), 27 , "Base", 90},
  { Servo(), 26 , "Shoulder", 90},

```

```

    { Servo(), 25 , "Elbow", 90},
    { Servo(), 33 , "Gripper", 90},
    { Servo(), 32 , "w1", 90},
    { Servo(), 14 , "w2", 90},
};

struct RecordedStep
{
    int servoIndex;
    int value;
    int delayInStep;
};
std::vector<RecordedStep> recordedSteps;

bool recordSteps = false;
bool playRecordedSteps = false;

unsigned long previousTimeInMilli = millis();

const char* ssid    = "RobotArm";
const char* password = "12345678";

AsyncWebServer server(80);
AsyncWebsocket wsRobotArmInput("/RobotArmInput");

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

    input[type=button]
    {
        background-color:red;color:white;border-
radius:30px;width:100%;height:40px;font-size:20px;text-align:center;
    }

    .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: 20px;
  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: 40px;
  height: 40px;
  border-radius: 50%;
  background: red;
  cursor: pointer;
}

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

```

width: 40px;
height: 40px;
border-radius: 50%;
background: red;
cursor: pointer;
}

```

```
</style>
```

```
</head>
```

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

```
<h1 style="color: teal;text-align:center;">Robot Arm Control</h1>
```

```
<table id="mainTable" style="width:400px;margin:auto;table-layout:fixed"
CELLSPACING=10>
```

```
<tr/><tr/>
```

```
<tr>
```

```
<td style="text-align:left;font-size:25px"><b>w1:</b></td>
```

```
<td colspan=2>
```

```
<div class="slidecontainer">
```

```
<input type="range" min="0" max="180" value="90" class="slider"
id="w1" oninput='sendButtonInput("w1",value)'
```

```
</div>
```

```
</td>
```

```
</tr>
```

```
<tr/>
```

```
<tr>
```

```
<td style="text-align:left;font-size:25px"><b>Gripper:</b></td>
```

```
<td colspan=2>
```

```
<div class="slidecontainer">
```

```
<input type="range" min="0" max="180" value="90" class="slider"
id="Gripper" oninput='sendButtonInput("Gripper",value)'
```

```
</div>
```

```
</td>
```

```
</tr>
```

```
<tr/>
```

```
<tr>
```

```
<td style="text-align:left;font-size:25px"><b>Elbow:</b></td>
```

```

<td colspan=2>
  <div class="slidecontainer">
    <input type="range" min="0" max="180" value="90" class="slider"
id="Elbow" oninput='sendButtonInput("Elbow",value)'>
  </div>
</td>
</tr>
<tr/><tr/>
<tr>
  <td style="text-align:left;font-size:25px"><b>Shoulder:</b></td>
  <td colspan=2>
    <div class="slidecontainer">
      <input type="range" min="0" max="180" value="90" class="slider"
id="Shoulder" oninput='sendButtonInput("Shoulder",value)'>
    </div>
  </td>
</tr>
<tr/><tr/>
<tr>
  <td style="text-align:left;font-size:25px"><b>Base:</b></td>
  <td colspan=2>
    <div class="slidecontainer">
      <input type="range" min="0" max="180" value="90" class="slider"
id="Base" oninput='sendButtonInput("Base",value)'>
    </div>
  </td>
</tr>
<tr>
  <td style="text-align:left;font-size:25px"><b>w2:</b></td>
  <td colspan=2>
    <div class="slidecontainer">
      <input type="range" min="0" max="180" value="90" class="slider"
id="w2" oninput='sendButtonInput("w2",value)'>
    </div>
  </td>
</tr>
<tr/>
<tr>
  <td style="text-align:left;font-size:25px"><b>Record:</b></td>

```

```

        <td><input          type="button"          id="Record"          value="OFF"
ontouchend='onclickButton(this)'></td>
        <td></td>
    </tr>
</tr></tr>
<tr>
    <td style="text-align:left;font-size:25px"><b>Play:</b></td>
    <td><input          type="button"          id="Play"          value="OFF"
ontouchend='onclickButton(this)'></td>
    <td></td>
</tr>
</table>

```

```

<script>
    var websocketRobotArmInputUrl = "ws://\\" + window.location.hostname +
"/RobotArmInput";
    var websocketRobotArmInput;

    function initRobotArmInputWebSocket()
    {
        websocketRobotArmInput = new
WebSocket(websocketRobotArmInputUrl);
        websocketRobotArmInput.onopen = function(event){ };
        websocketRobotArmInput.onclose =
function(event){setTimeout(initRobotArmInputWebSocket, 2000);};
        websocketRobotArmInput.onmessage = function(event)
        {
            var keyValue = event.data.split(",");
            var button = document.getElementById(keyValue[0]);
            button.value = keyValue[1];
            if (button.id == "Record" || button.id == "Play")
            {
                button.style.backgroundColor = (button.value == "ON" ? "green" : "red");
                enableDisableButtonsSliders(button);
            }
        };
    }

    function sendButtonInput(key, value)
    {

```

```

var data = key + "," + value;
websocketRobotArmInput.send(data);
}

function onclickButton(button)
{
    button.value = (button.value == "ON") ? "OFF" : "ON" ;
    button.style.backgroundColor = (button.value == "ON" ? "green" : "red");
    var value = (button.value == "ON") ? 1 : 0 ;
    sendButtonInput(button.id, value);
    enableDisableButtonsSliders(button);
}

function enableDisableButtonsSliders(button)
{
    if(button.id == "Play")
    {
        var disabled = "auto";
        if (button.value == "ON")
        {
            disabled = "none";
        }
        document.getElementById("w1").style.pointerEvents = disabled;
        document.getElementById("Gripper").style.pointerEvents = disabled;
        document.getElementById("Elbow").style.pointerEvents = disabled;
        document.getElementById("Shoulder").style.pointerEvents = disabled;
        document.getElementById("Base").style.pointerEvents = disabled;
        document.getElementById("w2").style.pointerEvents = disabled;
        document.getElementById("Record").style.pointerEvents = disabled;
    }
    if(button.id == "Record")
    {
        var disabled = "auto";
        if (button.value == "ON")
        {
            disabled = "none";
        }
        document.getElementById("Play").style.pointerEvents = disabled;
    }
}

```

```

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

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

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

void onRobotArmInputWebSocketEvent(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());
            sendCurrentRobotArmState();
            break;
        case WS_EVT_DISCONNECT:
            Serial.printf("WebSocket client #%u disconnected\n", client->id());
            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 == "Record")
  {
    recordSteps = valueInt;
    if (recordSteps)
    {
      recordedSteps.clear();
      previousTimeInMilli = millis();
    }
  }
  else if (key == "Play")
  {
    playRecordedSteps = valueInt;
  }
  else if (key == "w1")
  {
    writeServoValues(0, valueInt);
  }
  else if (key == "Base")
  {
    writeServoValues(1, valueInt);
  }
  else if (key == "Shoulder")
  {
    writeServoValues(2, valueInt);
  }
  else if (key == "Elbow")
  {

```

```

        writeServoValues(3, valueInt);
    }
    else if (key == "Gripper")
    {
        writeServoValues(4, valueInt);
    }
    else if (key == "w2")
    {
        writeServoValues(5, valueInt);
    }

}
break;
case WS_EVT_PONG:
case WS_EVT_ERROR:
    break;
default:
    break;
}
}

void sendCurrentRobotArmState()
{
    for (int i = 0; i < servoPins.size(); i++)
    {
        wsRobotArmInput.textAll(servoPins[i].servoName + "," +
servoPins[i].servo.read());
    }
    wsRobotArmInput.textAll(String("Record,") + (recordSteps ? "ON" : "OFF"));
    wsRobotArmInput.textAll(String("Play,") + (playRecordedSteps ? "ON" :
"OFF"));
}

void writeServoValues(int servoIndex, int value)
{
    if (recordSteps)
    {
        RecordedStep recordedStep;
        if (recordedSteps.size() == 0) // We will first record initial position of all servos.
        {

```

```

    for (int i = 0; i < servoPins.size(); i++)
    {
        recordedStep.servoIndex = i;
        recordedStep.value = servoPins[i].servo.read();
        recordedStep.delayInStep = 0;
        recordedSteps.push_back(recordedStep);
    }
}
unsigned long currentTime = millis();
recordedStep.servoIndex = servoIndex;
recordedStep.value = value;
recordedStep.delayInStep = currentTime - previousTimeInMilli;
recordedSteps.push_back(recordedStep);
previousTimeInMilli = currentTime;
}
servoPins[servoIndex].servo.write(value);
}

void playRecordedRobotArmSteps()
{
    if (recordedSteps.size() == 0)
    {
        return;
    }
    //This is to move servo to initial position slowly. First 4 steps are initial position
    for (int i = 0; i < 6 && playRecordedSteps; i++)
    {
        RecordedStep &recordedStep = recordedSteps[i];
        int currentServoPosition = servoPins[recordedStep.servoIndex].servo.read();
        while (currentServoPosition != recordedStep.value && playRecordedSteps)
        {
            currentServoPosition = (currentServoPosition > recordedStep.value ?
currentServoPosition - 1 : currentServoPosition + 1);
            servoPins[recordedStep.servoIndex].servo.write(currentServoPosition);
            wsRobotArmInput.textAll(servoPins[recordedStep.servoIndex].servoName +
", " + currentServoPosition);
            delay(50);
        }
    }
    delay(2000); // Delay before starting the actual steps.
}

```

```

for (int i = 6; i < recordedSteps.size() && playRecordedSteps ; i++)
{
  RecordedStep &recordedStep = recordedSteps[i];
  delay(recordedStep.delayInStep);
  servoPins[recordedStep.servoIndex].servo.write(recordedStep.value);
  wsRobotArmInput.textAll(servoPins[recordedStep.servoIndex].servoName +
", " + recordedStep.value);
}
}

```

```

void setUpPinModes()
{
  for (int i = 0; i < servoPins.size(); i++)
  {
    servoPins[i].servo.attach(servoPins[i].servoPin);
    servoPins[i].servo.write(servoPins[i].initialPosition);
  }
}

```

```

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);

  wsRobotArmInput.onEvent(onRobotArmInputWebSocketEvent);
  server.addHandler(&wsRobotArmInput);

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

```

```
}  
  
void loop()  
{  
  wsRobotArmInput.cleanupClients();  
  if (playRecordedSteps)  
  {  
    playRecordedRobotArmSteps();  
  }  
}
```

THANK YOU