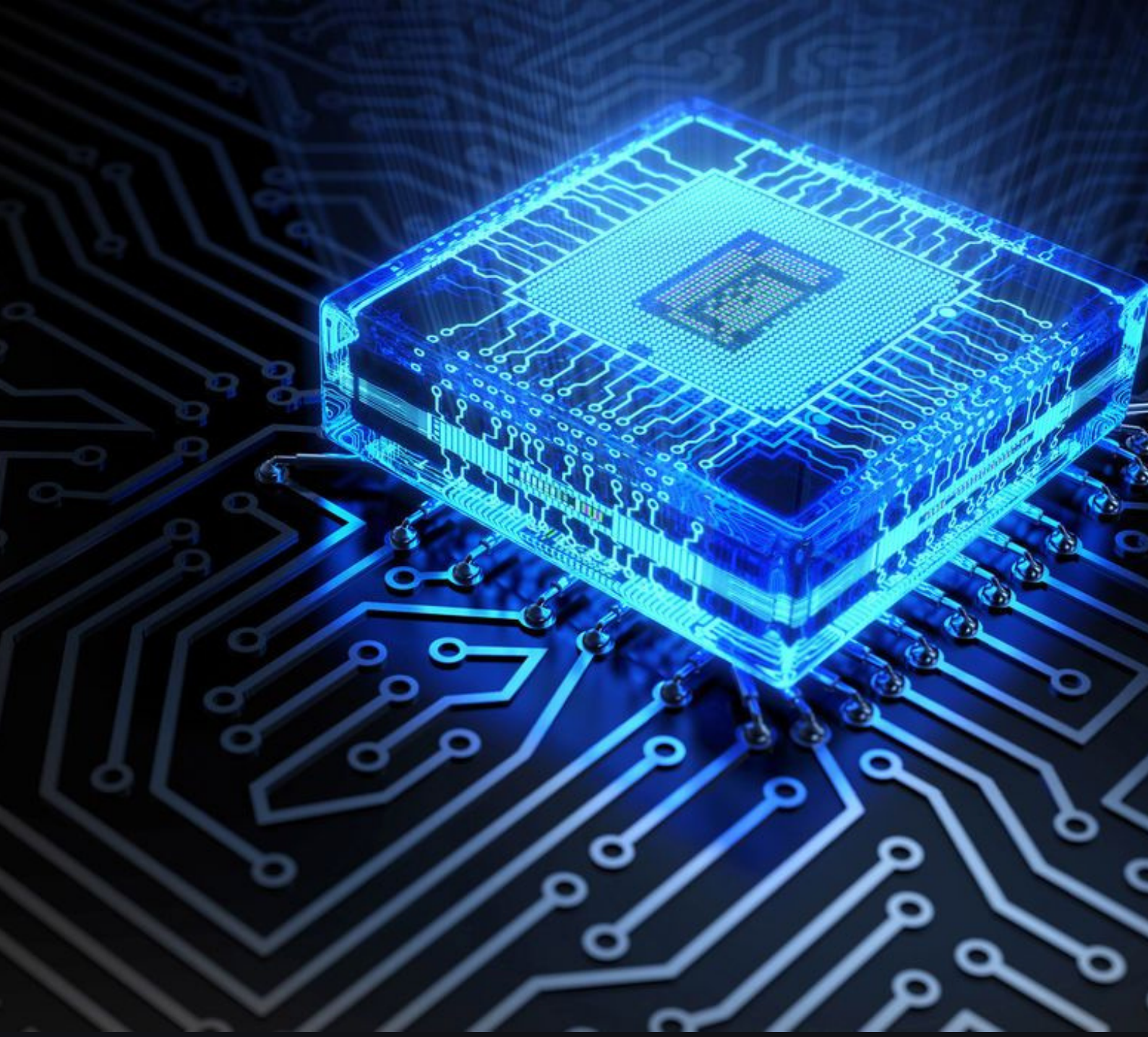


Microprocessor Final Project Report

Team 13

Name	ID
Alaa Mohammed Morsy	4
Alaa Hisham Mostafa	5
Mariam Said Gaber Ibrahim	214
Mariam Sayed Abdelhakem Mohammed	215
Mariam Amr Abdelhamied Bakr	216

Table of Contents

1.Introduction 2
1.1.Micropython 2
1.2.ESP32 3
2.Control a Relay Module with Micropython web server ... 5
2.1.IOT 5
2.2.Project concept 7
3.Software Code 8
3.1.Used Libraries in MIcropython 8
3.2.Code Explanation 9
4.HTML Code 11
4.1.Used Functions 11
4.2.Code Explanation 12
5.Hardware Implementation 13
5.1.Used Components 13
5.2.Connections and Workings 14
6.conclusion 15

Introduction

1.1.Micropython

1.1.1. Introduction about Micropython

Micropython is a smaller version of python programming language that can run on an embedded development board. A simple micropython code can control any hardware instead of using low-level languages (C or C++) that are associated with Arduino. Micropython can do anything Arduino can and better. It can control GPIO pins to blink lights, and flip switches. It can read analog sensors with an analog to digital converter or drive PWM outputs for servos, LEDs.

1.1.2. Micropython Features

Micropython is highly recommended for beginners thanks to its simplicity. It is full-featured and supports most of Additionally, Micropython has its own unique features:

- **Interactive REPL or read-evaluate-print loop.**

This allows a direct connection to any board with quick code execution without compiling or uploading.

- **Extensive software library.**

Micropython has various libraries that allow it to execute many tasks like parsing JSON data from a web service, searching text with a regular expression, or even doing network socket programming. MicroPython even has libraries for controlling other hardware like NeoPixels and LED strips, tiny OLED displays, and more.

- **Extensibility.**

This allows users to mis high-level Micropython code with faster low-level C or C++ code.

1.2.Esp32

1.2.1. Introduction about ESP32

ESP32 is developed by Espressif systems. The chip is most used for mobile devices, wearable tech, and IOT applications.

1.2.2. ESP32 Features

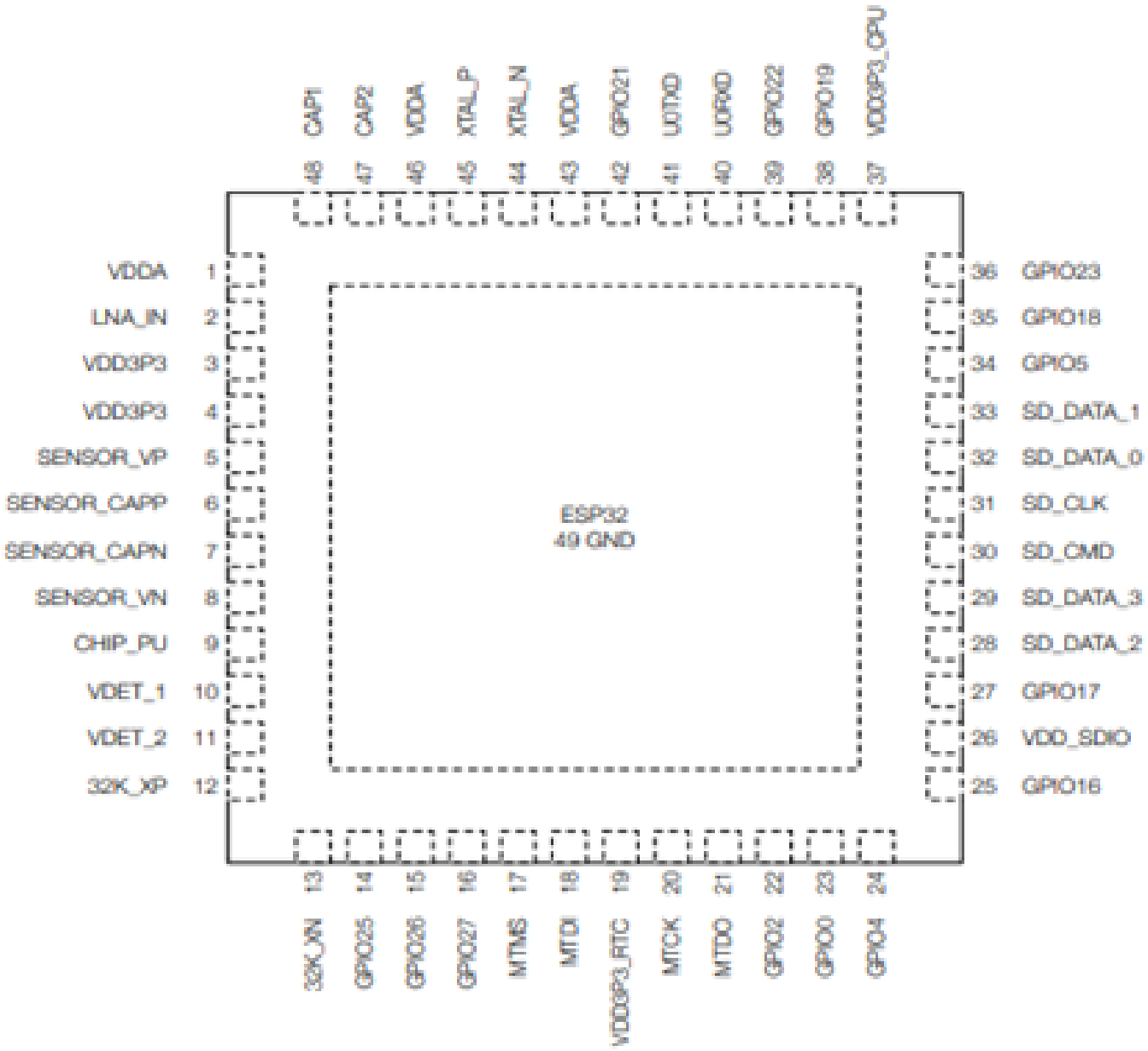
Models are available with combined Wi-Fi and Bluetooth connectivity, or just Wi-Fi connectivity. There are now several different chip models available, including:

- ESP32-D0WDQ6 (and ESP32D0WD)
- ESP32-D2WD
- ESP32-S0WD
- And the system in package (SiP) – ESP32-PICO-D4

Summary of ESP32 Features and Specifications

Core	**2**
Clock	**Tensilica Xtensa LX106 160-240MHz**
RAM	**520KB**
Flash	**Extern QSPI - 16MB**
DAC	**2**
ADC	**18**
GPIO	**22**
Arquitecture	**32 bits**
Bluetooth	**Yes - classic & BLE**
WiFi	**IEEE802.11 b/g/n**
Interfaces	**SPI-I2C-UART-I2S-CAN**

1.2.3. ESP32 Pin layout



2.Control a Relay Module web with Micropython server

2.1.IOT

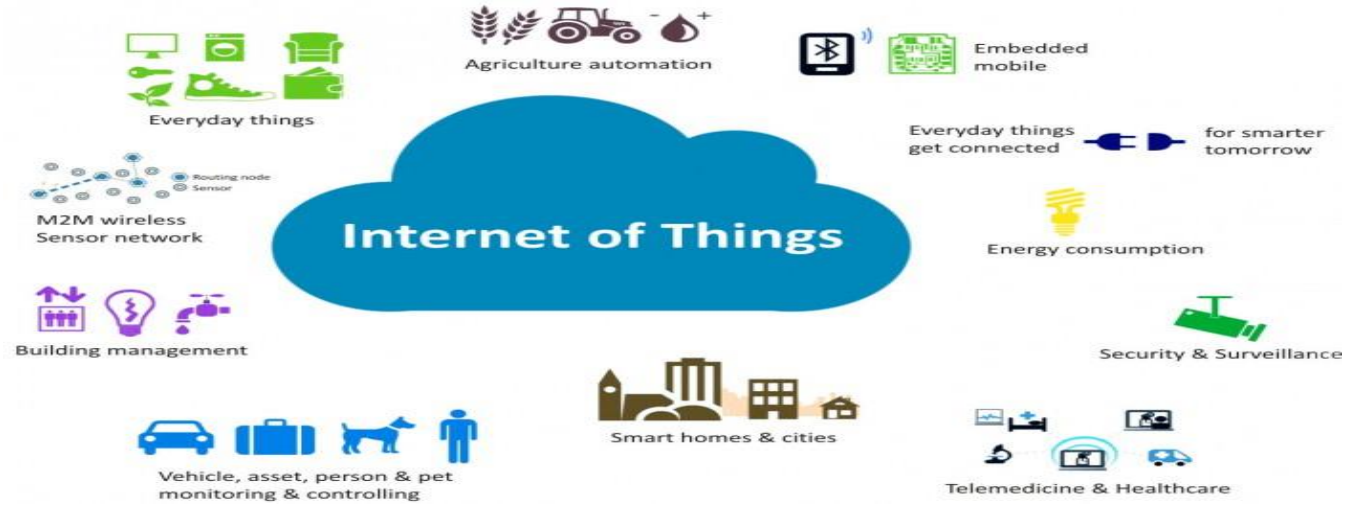
For centuries now Scientists and inventors have worked to develop all sorts of technology to make our daily lives easier. IOT is one of the most important means to achieve the end. The internet of things is defined as the connection between multiple objects and even other humans through the internet. IOT has multiple uses and many companies and startups are adapting IOT based technologies in different sectors and fields.

2.1.1. The history of IOT

The term IOT was first used in 1999 by Kevin Ashton during a presentation for Procter & Gamble. However, the term IOT did not get much attention for the next ten years. IOT reappeared again in 2010, when rumors spread about Google's Street View service had made 360-degree pictures and stored data of people's Wi-Fi networks. In the same year, China announced its intension to add IOT devices as a priority in its five-year plan. After that, the term IOT had a widespread among developers so far that it became a word we here frequently in our everyday lives.

2.1.2. IOT Applications

As mentioned before IOT has unlimited applications in various fields that are constantly increasing.



➢ Wearable Applications

Smart watches, Heart monitors, GPS tracking belts, Virtual reality glasses, and so much more. Wearable IOT devices have a wide range. Big companies like Google, Apple, and Samsung are developing many IOT applications. Soon, IOT base smart clothes will be available. Shirts that will monitor heart rates. Shoes that will measure ground contact time, and stride length.

➢ Health Care Applications

The wearable applications of IOT are used in the medical care to monitor heart rate, blood pressures, and glucose levels. Moreover, Smart beds will soon replace normal hospital beds. The beds will monitor Temperature, Vital signs, do examinations among other things.

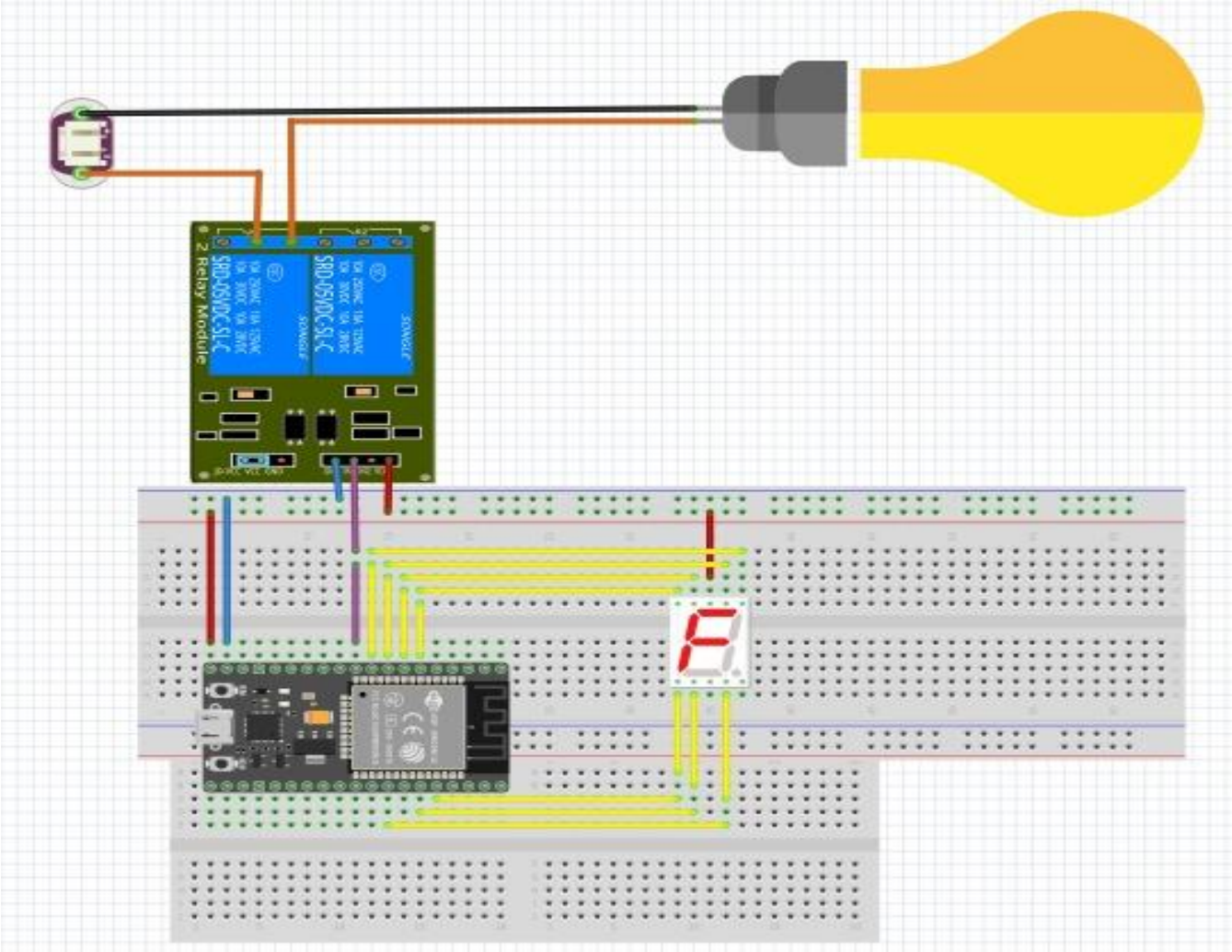
➢ Agriculture Applications

Smart farms use IOT sensors to measure soil moisture, level of acidity, the presence of certain nutrients, temperature, and many other chemical characteristics. IOT devices also help farmers control irrigation, and even discover the presence of diseases in plants and soil.

➢ Smart Home Applications

IOT controlled homes will provide complete comfort. Lights will turn on or off or even dim by the control of a smartphone. Using facial recognition to control the opening and closing of doors. Windows and Curtains can act accordingly to Sunlight. Additionally, all electronic devices like Microwaves, TVs, ovens, washing machines, refrigerators, and air conditioning can be controlled using a smart phone through IOT.
All these applications have great advantages. Automatic control will allow energy optimization which eventually leads to saving money on electricity bills. Another advantage, time management, a person can remotely access and run the washing machine to do the laundry while they are outside or even at work. The refrigerator can even monitor the regular products the residents buy and automatically contact the stores and order a refill.
The smart home application is the potential use of our project.

2.2. Project Concept



Using a relay Module to control AC household loads remotely. The relay itself will be controlled by an ESP32 Chip which will be controlled using a micropython code. The relay will control a lightbulb and a seven-segment display. Using a web server html will enable us to control this entire circuit remotely. Using a smart phone or a laptop with the application on it. We will be able to turn the light on and off. The seven-segment display will be decremented and incremented. This is a simplification of the IOT smart home concept.

3.Software Code

3.1. Used libraries in Micropython.

➢ Usocket

This library Creates a new socket using the given address family, socket type and protocol number.

➢ Machine

The machine library contains specific functions related to the hardware on a particular board. Most functions in this module allow to achieve direct and unrestricted access to and control of hardware blocks on a system (like CPU, timers, buses, etc.). Used incorrectly, this can lead to malfunction, lockups, crashes of your board, and in extreme cases, hardware damage. In this code we are using the machine library to import the pin method. So, we can output on a specific pin.

➢ Network

Using the Network library enables to use network drivers and routing configuration. Here, we are using it because our project requires a WIFI connection.

➢ Esp

The ESP library contains functions related to both the ESP8266 and ESP32 modules. We need in order to use the ESP32.

➢ Garbage collector (gc)

This library provides an interface to the optional garbage collector. It provides the ability to disable the collector, tune the collection frequency, and set debugging options.

3.2.Code Explanation

3.2.1.Explanation of the first boot file

➢First, we imported the library usocket as socket. We used this library to be able to interface with the address. We imported Pin form the machine library. So, we can output on a specific pin. We also imported the Network an ESP libraries. We used the **esp.osdebug(none)** to stop any previously existing debugging. Then we imported the gc library and wrote a command to run it.

➢After that, we created two variables named ssid and password and we entered the Wi-Fi data into them. We also created the variable station to interface with the Wi-Fi and we activated this station.

➢Then, we created a while loop to check the connection. If the connection is false, the loop will run again until the connection is true. The phrase Connection successful will be printed on the screen along with the board's IP address.

➢The we created a variable named relay and made the pin 26 an output.

3.2.2. Explanation of the main file

- In this file, we continue the previous code. We first check the relay state using an if statement. When the relay state is checked we go to the html code.
- After the program returns from the html code, we created a variable named s to have a socket type. We then bind the socket to the address. The listen command is used to enable the server to accept the connection.
- Then, we created a while loop, we first check the free space in the RAM, if we have enough space, we run the garbage collector.
- We use the s. accept command to Accept a connection. The socket must be bound to an address and listening for connections. The return value is a pair (conn, address) where conn is a new socket object usable to send and receive data on the connection, and address is the address bound to the socket on the other end of the connection.
- After that, we set a three second time out. Then the phrase got a connection from address (the address is displayed as a sting) is printed.

- Next, we receive the data and take a request. The code then determines if the request is to turn the relay on or off. If on prints relay on, if off prints relay off.
- Later, all the received data is sent to the html code. This was all done in try so that, if there is an error in any of that the code will go into the except case where the connection will be closed.

- This next part is related to the seven-segment display. Here we created a function that changes the values of the LEDs to display the numbers for zero to nine. The variable num expresses the number displayed on the seven-segment display. We use an if statement in each case to make the LEDs values corresponding the variable num's value. For example, when num=0, we give all the LEDs a value of one for g we give a zero value so that zero is displayed.
- Then, we have three variables, inc for incrementation, dec for decrementation, and reset. When any of them is requested the data is sent to the html code and the operation is executed accordingly.

4.HTML Code

4.1. Used Functions

➢ html and /html

Any code must start with the <html> and closed with </html>.

➢ head

<head> is a container for metadata (data about data). It is closed with </head>.

➢ meta

The meta tag defines metadata about an HTML document. <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

➢ Style

The HTML <style> element contains style information for a document, or part of a document. The <style> element must be included inside the <head> of the document. It is closed with </style>.

➢ Body

The body tag defines the document's body. The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc. There can only be one body element in an HTML document.

➢ Script

The script tag is used to embed executable code or data; this is typically used to embed or refer to JavaScript code.

4.1. Code Explanation

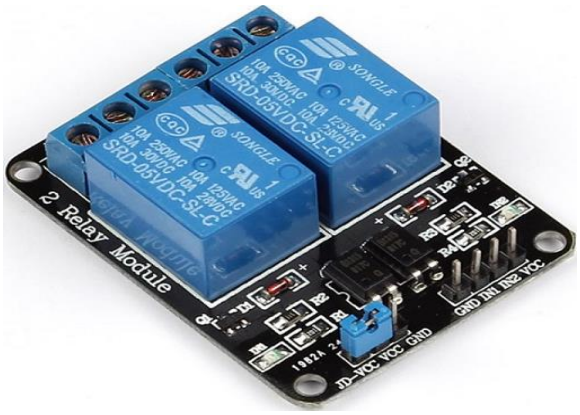
- Starting with html, we opened a head Then we use meta, we defined the name of the project as viewport, its width as the used device's width, the initial scale as one.
- Then, the style started, we choose the font as Arial, the margin, the padding top, and the text alignment. For each header (h1, h2, h3, h4), we chose a font-size.
- Next, we described each element (button, switch, slider). For the button we chose a color for the background, no boarder, and the display will be inline. We chose a font size. Adjusted the margin and the padding. And so, on we are choosing the display settings until the style is closed.
- After opening the script, we check the variable. Then any input taken is assigned to the variable. If checked then relay on, if not then relay off. The script and the head are closed.
- Later, we opened the body, it contains all the headers, definitions, and input types.
- Finally, then we defined a hyper link for each button.

5.Hardware Implementation

5.1. Used Components

➢ Two Channel Relay Module

A relay is simply an electrically controlled switch. It can be turned on and off (let the current pass through or not). It can be controlled with low voltages and can control high voltages. A two-channel relay can control two outputs.



➢ ESP32

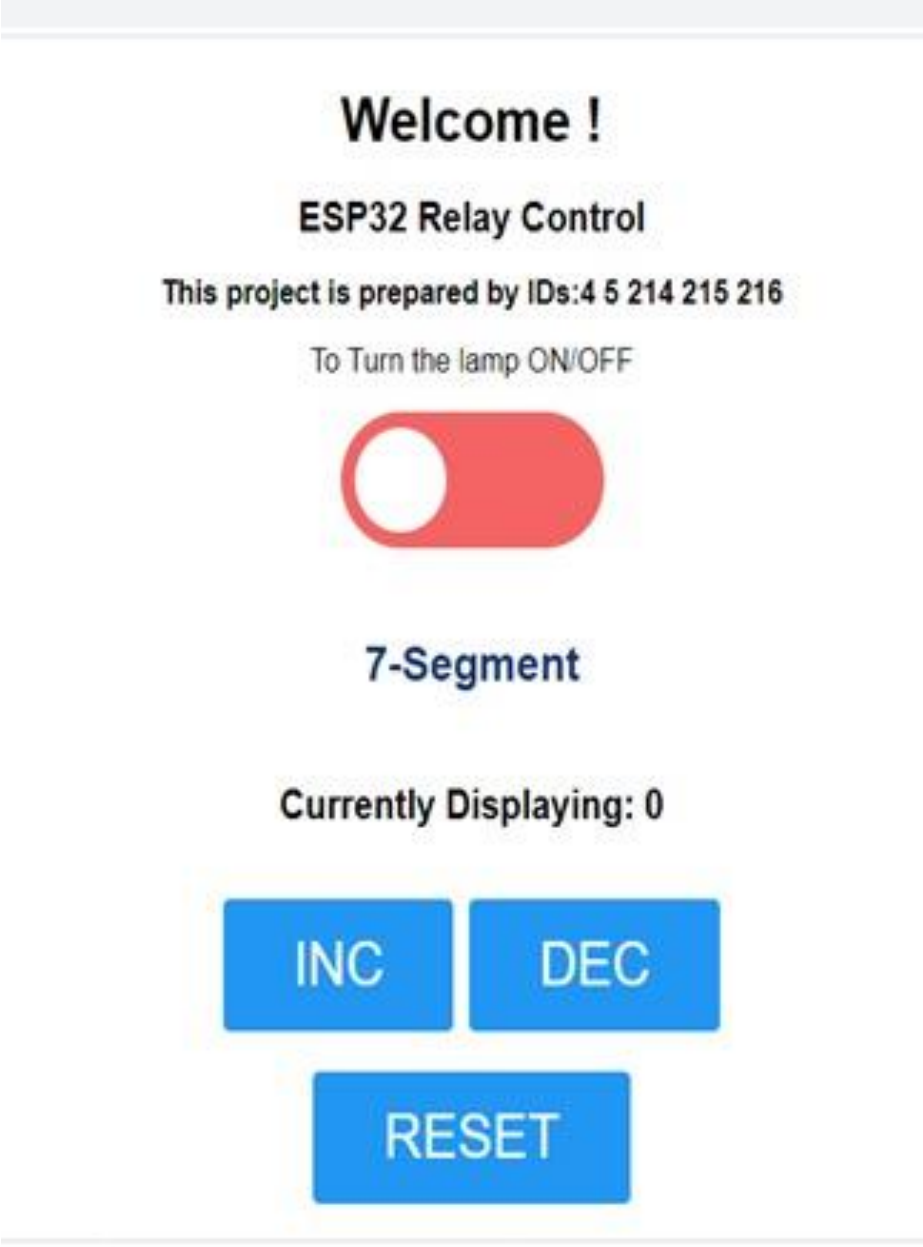
As previously explained. The esp32 controls the relay module.

➢ A Lightbulb, A seven segment display.

These are the outputs connected to each channel of the relay and being controlled.

5.2. Connections and Workings

The ESP32 is controlled through a Micropython code and a HTML code. The chip is programmed to output a 3.3V on the 26th pin. This pin is connected to a relay module that switches between the lamp and the Seven segment display. Using the web application, we get the figure below displayed on a laptop or a mobile screen. The first switch turns the lamp on or off. The seven-segment display starts at zero. The inc button increments it and the dec button decrements. The reset button resets back to display zero.



6.conclusion

As mentioned before this project is a simple implementation of the IOT smart home application. In which a smart phone can control all sort of things like lights, lamps, electric devices, doors, windows, curtains, etc.
This project has a huge potential for development and a widespread of applications. The project can be developed to use a relay module with more channels, this will give us the opportunity to control more loads like a motor. The html and Micropython codes can be developed to offer more options. for example, turning on the lamp for a specific time. IOT applications are increasing every day. The future is smart homes and smart cities that will make our daily lives a lot easier.