HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY FOR HIGH QUALITY TRAINING

SENIOR PROJECT 2

DESIGN AND IMPLEMENTATION OF A LIGHTING CONTROL SYSTEM

MAJOR OF COMPUTER ENGINEERING TECHNOLOGY

Students: **NGUYỄN GIA HƯNG**

ID: 19119052

NGUYỄN VĂN QUANG HUY

ID: 19119030

HO CHI MINH CITY – DECEMBER/2022

HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY FOR HIGH QUALITY TRAINING

HCMUTE

SENIOR PROJECT 2

DESIGN AND IMPLEMENTATION OF A LIGHTING CONTROL SYSTEM

MAJOR OF COMPUTER ENGINEERING TECHNOLOGY

Student: **NGUYỄN GIA HƯNG**

ID: 19119052

NGUYỄN VĂN QUANG HUY

ID: 19119030

Supervisor**: M.E. LÊ MINH**

HO CHI MINH CITY – DECEMBER/2022

SUPERVISOR APPROVAL

ACKNOWLEDGMENTS

First and foremost, we would to express our gratitude to M.E. Le Minh for his enthusiastic leadership of our Senior Project 2 team. In this semester, we have acquired knowledge that we can employ.

Once more, many thank you to the lecturers in the Computer Engineering – Communication Department for setting up all of the circumstances that allowed us to finish this project on time.

We hope that you and the subject teachers would sympathize with our group since there is not much time to apply our topic and there are still many problems in completion process.

In summary, I would like to sincerely thank everyone who supported during the implementation of the project.

Ho Chi Minh City, December 2022

Student Performance

Nguyễn Văn Quang Huy

Nguyễn Gia Hưng

ABSTRACT

This report summarizes the process of ideation, research, design, construction, results achieved and development direction of the lighting control system. This system can be applied in practice so that users can choose to turn on or off the lights in the house by two modes: manual mode and automatic mode. With a database stored on google firebase, besides the Website, this system will make the work of controlling the lights in the house more convenient and users can monitor the current light level in the house.

This report can be used as a reference for students of Computer Engineering, Telecommunication Electronic Engineering, or those interested in electrical and electronic systems.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

LCD	Liquid Crystal Display
ESP32	Kit Node MCU ESP32
MCU	Microcontroller Unit
DPDT	Double Pole Double Throw
LDR	Light Dependent Resistor
LED	Light Emitting Diode
RTC	Real Time Clock
CPU	Central Processing Unit
SDA	Serial Data
SCL	Serial Clock
TWI	Two-Wire Interface
BLE	Bluetooth Low Energy
UART	Universal Asynchronous Receiver/Transmitter
LSB	Least Significant Bit
MSB	Most Significant Bit
AC	Alternating Current

CHAPTER 1

OVERVIEW

1.1 INTRODUCTION

Nowadays, with the explosive change of the 4.0 technology era with the development of the Internet, smart devices are created to meet human needs, in which the need for connectivity is an important need at the moment. Therefore, Internet of Things technology was created to meet this need. An IoT system is a system of many devices that are able to connect to each other via the Internet or wireless wave environments, helping people control their system remotely, devices become smarter when they are connected to communication to each other.

Along with the development of IoT technology, smart systems were formed such as smart city, smart farm, smart home… To approach the IoT model and learn the basic knowledge of this technology, our group decided to use IoT technology as the foundation for the system with the idea of building a system with the aim of making it easier to control lights in the house. Therefore, our group choose project **“Lighting Control System”**.

1.2 OBJECTIVES

The project “Lighting Control System” was carried out to design a system with the following functions:

- The system allows users to control indoor and outdoor lights in the house by two modes including manual control mode and automatic control mode. The user will select the operating mode of the system with a DPDT self-locking switch.
- In manual control mode, the users can control the lights on or off with two physical buttons or with two buttons on the website. The status of the lights will be sent to google firebase to serve to create buttons on the website.

- In automatic control mode, the indoor light is control by light sensor and the outdoor light is control by timer. The light intensity value is sent from the hardware to google firebase and the website will display it.
- The website makes it easy for users to monitor the status of the lights through the image and color of the button or the light intensity of the light sensor, and the website also displays the information of the system creator.

1.3 SCOPE

In the process of researching and implementing Lighting Control System, our group focuses on the following tasks:

- Build a webserver to control lights and display data from hardware.
- Study the data exchange between the processor board and the server.
- Build a webserver to exchange data.
- The system only works with two processor board, with two 5W LED Bulb.

1.4 RESEARCH OBJECTS

To design and build the system, our group will have to study the following objects:

- Kit ESP32 (NodeMCU ESP32 Wifi BLE).
- Kit Arduino Nano V3.0 ATmega328P.
- Light Dependent Resistor (LDR) Sensor Module.
- LCD 16X2
- I2C LCD Interface Module.
- 2-channel 5V Relay Module.
- Real Time Clock DS3231 Module.
- Electrical equipment: 2 5W LED Bulb, 6 push buttons, 1 DPDT switch.
- Google Firebase for storage.
- Website to create interaction between users and the system.

1.5 OUTLINE

To easily track and capture information, the report is divided into 5 chapters:

Chapter 1. OVERVIEW: General introduction to the topic.

Chapter 2. BACKGROUND: Introduce the technique and major hardware components devices that used in the topic.

Chapter 3. DESIGN AND IMPLEMENTATION: This chapter covers the detail design and implementation of our work.

Chapter 4. RESULT: This chapter describes the detail results of our work, including hardware implementation, software design and operating of our system.

Chapter 5. CONCLUSIONS AND FUTURE WORK: To summarize what our study has accomplished and propose some utilizations or development for the topic in the future work.

CHAPTER 2

BACKGROUND

2.1 KIT ESP32 (NODEMCU ESP32 WIFI BLE)

2.1.1 Introduction to KIT ESP32

NodeMCU-32 is a core development board that is designed by Emerson based on the ESP-32S module. The development board continues the classic design of the NodeMCU 1.0, bringing out most of the I/O pins on both sides, allowing developers to connect peripherals according to their needs. When using breadboard for development and debugging, the standard pin headers on both sides make the operation easier and more convenient. ESP32 is an integrated antenna and RF balun, power amplifier, low-noise amplifier, filters and power management module. The module is optimized to take as little space as possible.

2.1.2 KIT ESP32 Specifications:

- Main IC: ESP32 (ESP32-WROOM-32)
- Firmware version: NodeMCU Lua
- Chip feeder and UART interface: CP2102.
- GPIO fully compatible with Node MCU firmware
- GIPO communication level: 3.3VDC
- Power used: 5VDC (via MicroUSB or Vin)
- Wifi standard: 802.11b/g/n/e/I
- Bluetooth: BR/EDR+BLE
- RAM: 520Kbytes
- ROM: 448Kbytes
- Built-in LED status indicator, Reset button, Flash.
- Fully compatible with the compiler: Arduino IDE

- Dimensions: 55mm x 28mm

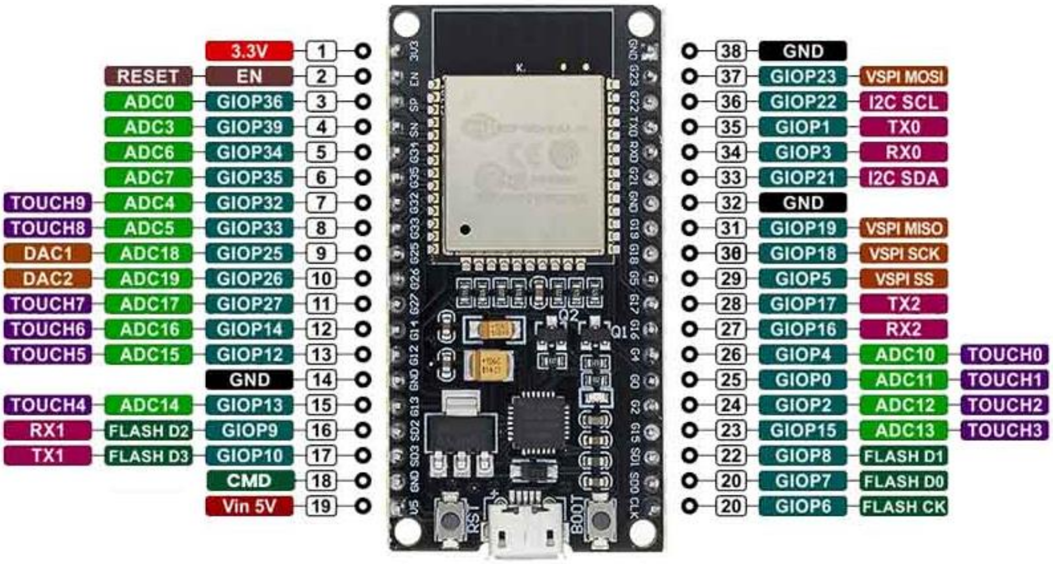


Figure 2.1: Kit ESP32 [1]

2.2 KIT ARDUINO NANO

2.2.1 Introduction to KIT Arduino Nano

Arduino Nano is one type of microcontroller board, and it is designed by Arduino.cc. It can be built with a microcontroller like Atmega328. This microcontroller is also used in Arduino UNO. It is a small size board and also flexible with a wide variety of applications. This board has many functions and features like an Arduino Duemilanove board. However, this Nano board is different in package. It does not have any DC jack so that the power supply can be given using a small USB port otherwise straightly connected to the pins like VCC & GND. This board can be supplied with 6 to 20 volts using a mini USB port on the board.

2.2.2 KIT Arduino Nano Specifications:

- Microcontroller: ATmega328
- Architecture: AVR
- Operating Voltage: 5V
- Flash Memory: 32KB of which 2 KB used by bootloader
- SRAM: 2KB
- Clock Speed: 16 MHz

- Analog IN Pins: 8
- EEPROM: 1KB
- DC Current per I/O Pins: 40 mA(I/O Pins)
- Input Voltage: 7-12V
- Digital I/O Pins: 22 (6 of which are PWM)
- PWM Output: 6
- Power Consumption: 19 mA
- PCB Size: 18 x 45mm
- Weight: 7g

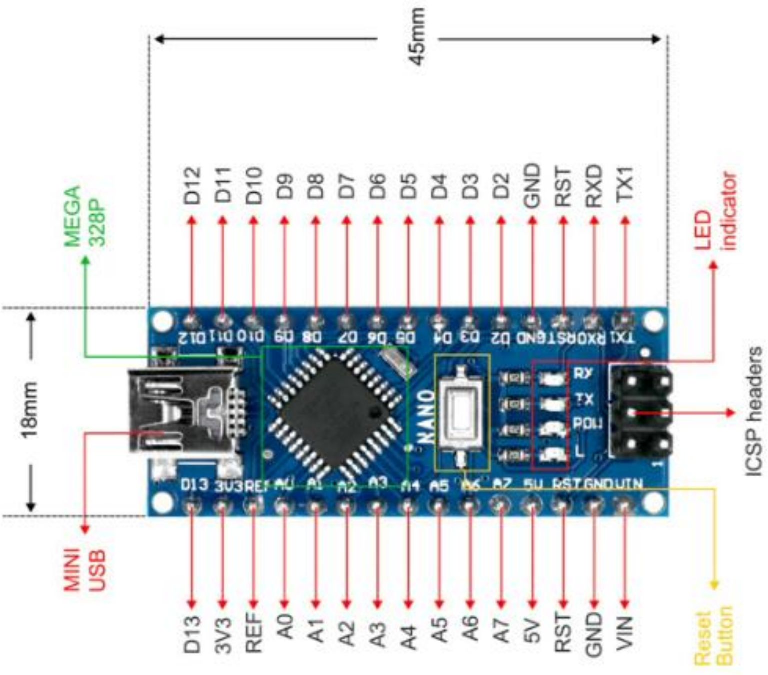


Figure 2.2: Kit Arduino Nano [2]

2.3 LDR SENSOR MODULE

2.3.1 Introduction to LDR Sensor Module

The LDR Sensor Module is used to detect the presence of light/ measuring the intensity of light. It is associated with both analog pin and digital pin labelled as A0 and D0 respectively on the board. When there is light, the resistance of LDR will become low according to the intensity of light.

2.3.2 LDR Sensor Module Specification:

- Using photosensitive resistance sensor sensitive type
- Working voltage: 3.5V-5V
- Output form: digital switch show (0 and 1) or analog value
- Fixed bolt hole, easy installation
- Small PCB size: 3.2cm x 1.4cm
- The comparator output, signal clean, good waveform, driving ability is strong for more than 15 mA
- With adjustable potentiometer to adjust light brightness

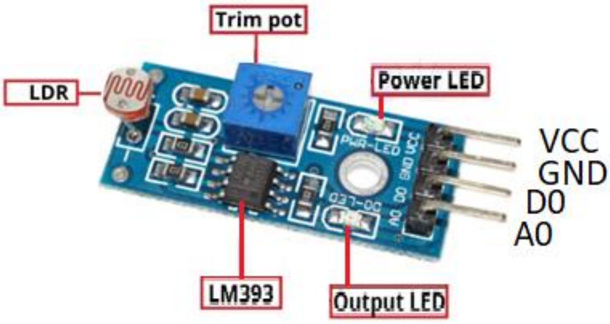


Figure 2.3: LDR Sensor Module [3]

2.4 LCD 16X2

2.4.1 Introduction to LCD 16X2

The term LCD stands for liquid crystal display. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x8 pixel matrix. The 16x2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.

2.4.2 LCD 16X2 Specification:

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16 characters
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5x8 pixel box
- The alphanumeric LCDs alphabets & numbers

- Is display can work on two modes like 4-bit and 8-bit
- These are obtainable in Blue & Green Backlight
- It displays a few custom generated characters

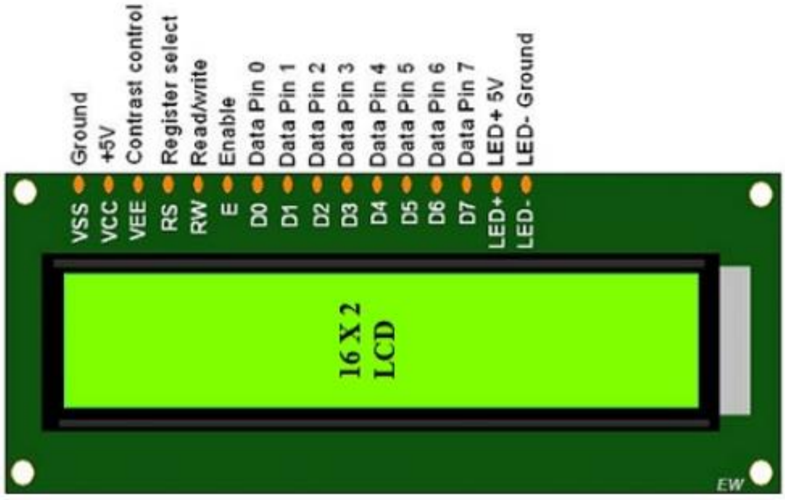


Figure 2.4: 16x2 LCD [4]

2.5 I2C LCD INTERFACE MODULE

2.5.1 Introduction to I2C LCD Interface Module

I2C LCD is an easy-to-use display module, it can make display easier. Using it can reduce the difficulty of make, so that makers can focus on the core of the work. I2C Module is a parallel to serial converter compatible with LCD2004 and LCD 1602. By using this module, LCD can be interfaced with using only 2 wires.

2.5.2 I2C LCD Interface Module Specification:

- Operating Voltage: 5V DC
- I2C control using PCF8574
- Can have 8 modules on a single I2C bus
- I2C address: 0x20-0x27 (the original address is 0x20)

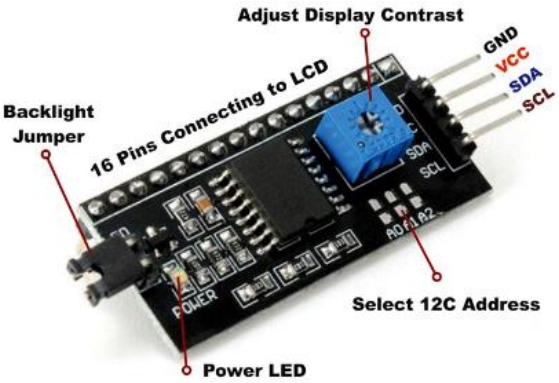


Figure 2.5: I2C LCD Interface Module [5]

2.6 2-CHANNEL 5V RELAY MODULE

2.6.1 Introduction to 2-channel 5V Relay Module

This 2 Channel 5V Relay Module is a set of single-pole double-throw relays rated for maximum of 10A/250VAC or 10A/30VDC, and is ideal for controlling large AC or DC loads with Development Platforms like Arduino or Raspberry Pi. The relays accept a both 5V and 3V control inputs from Arduino Shields and boards, and help to protect users and other components from electrical leakage or faults. This of course is important when operating high current parts like motors, solenoids or electromagnets, as electrical leakage can not only potentially harm a user, but can also destroy other components that are not designed to handle high current or voltage.

2.6.2 2-channel 5V Relay Module Specification:

- Operating Voltage: 3V-5V DC
- Max Voltage (AC): 250V
- Max Voltage (DC): 30V
- Max Current: 10A
- Min Current: 15-20mA (Per Relay)
- Coil Voltage: 5V
- Isolation Methods: Optical Isolation + Galvanic Isolation
- Status Indication: LED
- Dimensions: 46x32mm (W x L)

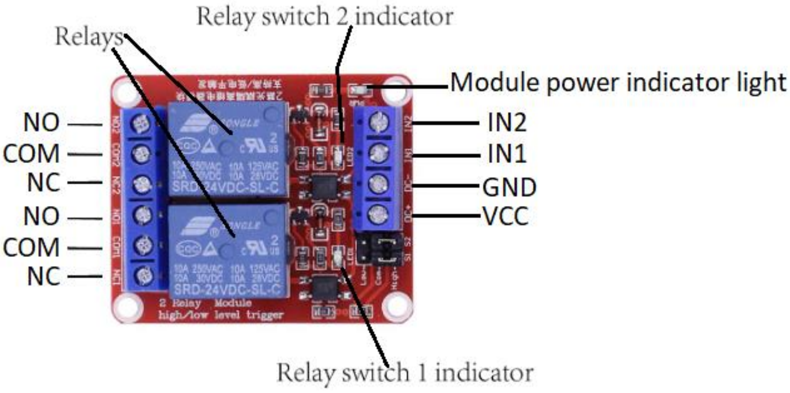


Figure 2.6: 2-channel 5V Relay Module [6]

2.7 RTC DS3231 MODULE

2.7.1 Introduction to RTC DS3231 Module

DS3231 RTC module is a famous real-time clock in the market for its high accurate results. RTC stands for real-time clock. It is an affordable system that manages Time and Date with high precision. This small-size module has 6 pins, an inter-integrated (I2C) interface for data transmission, and a battery cell for power-backup. DS3231 Real-Time Clock Module comes with a TWI interface, two alarm clocks, and an inbuilt temperature-compensated crystal Oscillator.

2.7.2 RTC DS3231 Module Specification:

- Operating Voltage: 2.3V-5.5V DC
- Operating Temperature: -45 ºC – 80 ºC
- Real Time Clock Chip: DS3231 I2C
- EEPROM Chip: AT24C32 32K I2C EEPROM memory
- I2C interface: 400Khz
- Battery type: LIR2032 rechargeable lithium battery
- Battery Backup Current: 500 mA
- Power consumption: 500 nA
- Digital temperature sensor accuracy: ±3ºC
- Clock Provide: Hour, Minutes, Seconds, AM/PM.
- Calendar provides: Day, Date, Month, Year.
- PCB Size: 38mm (length) * 22mm (W) * 14mm (height)

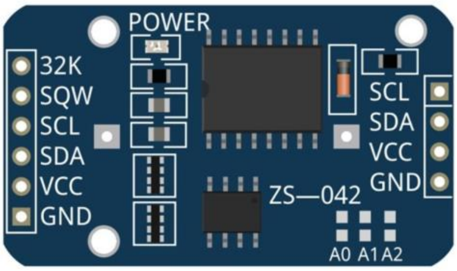


Figure 2.7: RTC DS3231 Module [7]

2.8 PUSH BUTTON

2.8.1 Introduction to Push Button

A push-button or simply button is a simple switch mechanism to control some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Push buttons or switches connect two points in a circuit when you press them.

2.8.2 Push Button Specification:

- Mode of Operation: Tactile feedback
- Power Rating: MAX 50mA 24V DC
- Insulation Resistance: 100Mohm at 100V
- Operating Force: 2.55±0.69N
- Contact Resistance: MAX 100mOhm
- Operating Temperature Range: -20 to + 70 ºC
- Storage Temperature Range: -20 to +70 ºC

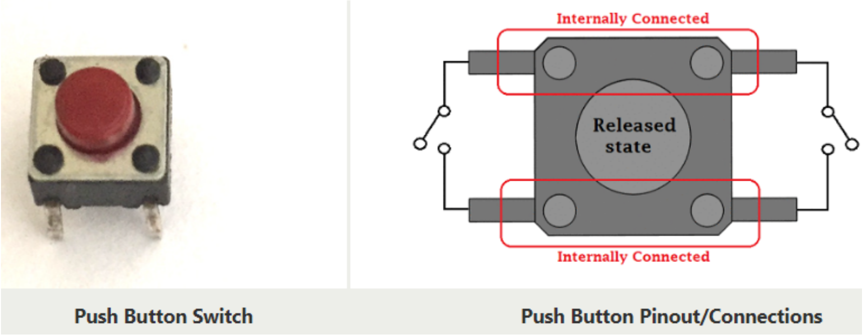


Figure 2.8: Push Button [8]

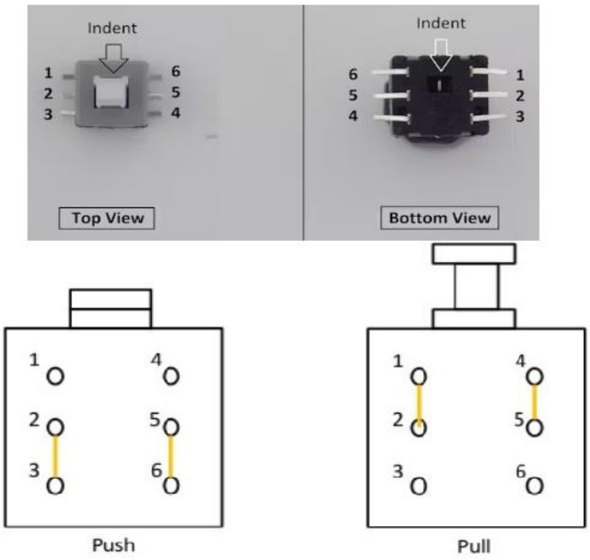
2.9 DPDT SELF-LOCKING SWITCH

2.9.1 Introduction to DPDT Self-locking Switch

A 6 Pin Push Switch also known as Mini DPDT Push Switch, is nothing but a combination of two push switches placed together inside one package. Unlike momentary switches which connect the wires of the switch only for a second, this switch retains its ON-OFF state till pushed later on. For example, If I push it once so that it is turned on, it will remain in ON state till it is pushed again That is why this switch is useful in controlling power connections most of the time.

2.9.2 DPDT Self-Locking Switch Specification:

- Switch type: DPDT
- Size: 9 x 9 mm
- Maximum voltage DC 30V
- Operating Current: 0.1 A
- PCB through-hole mounting



Left: DPDT switch in the 'push' position: 2 -> 3; 5 -> 6
Right: DPDT switch in the 'pull' position: 2 -> 1; 5 -> 4

Figure 2.9: DPDT Self-Locking Switch [9]

2.10 PROTOCOLS

2.10.1 UART

The UART is a type of asynchronous serial communication. UART stands for Universal Asynchronous Receiver/Transmitter. UART is commonly used in industrial computers, communications, microcontrollers, or a number of other communication devices. The purpose of UART is to transmit signals between each other or from microcontroller to microcontroller, from laptop to microcontroller.

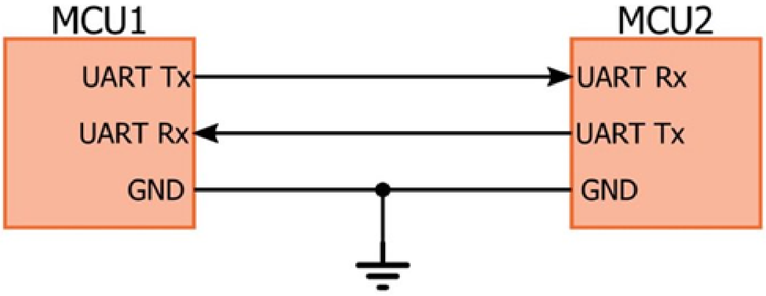


Figure 2.10: UART Communication Standard

UART Communication Parameters:

- Baud rate: Number of bits transmitted per second
- Transmission frame: Because it is very easy to lose data, in addition to the speed, the transmission frame is also installed from the beginning to avoid this data loss. The transmission frame specifies the number of bits in each transmission, alarm bits such as start and stop, check bits such as parity, and the number of bits in a data.
- Start bit: The start bit in the transmission frame. This bit is intended to tell the device that transmission has begun.
- Data: Data to be transmitted. In UART the LSB bit is transmitted first. The MSB bit is transmitted later.
- Parity Bit: Is the data check bit. There are two types of parity: even and odd. Even parity is the parity bit added to the number 1 in data + parity = even number; odd parity is the parity bit added to the number 1 in data + parity = odd number. The parity bit is optional and can be used or not.

- Stop bit: is the bit that reports the end of the transmission frame. Usually 5v level and maybe 1 or 2 stops.

2.10.2 I2C

I2C is the abbreviation for inter-integrated circuits. It is a synchronous, serial communication protocol which intended to allow multiple slave digital ICs to communicate with one or more master chips. This type of communication is best suitable for short distance communications within a single device.

I2C combines the best features of SPI and UARTs. With I2C you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, ot multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.

Like UART communication, I2C only uses two wires to transmit data between devices. In UART we have SDA (Serial Data) – The line for the master and slave to send and receive data. SCL (Serial Clock) – The line that carries the clock signal.

Like SPI, I2c Is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master.

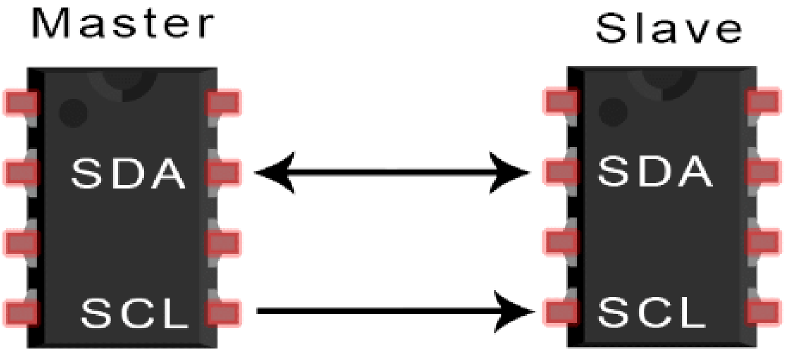
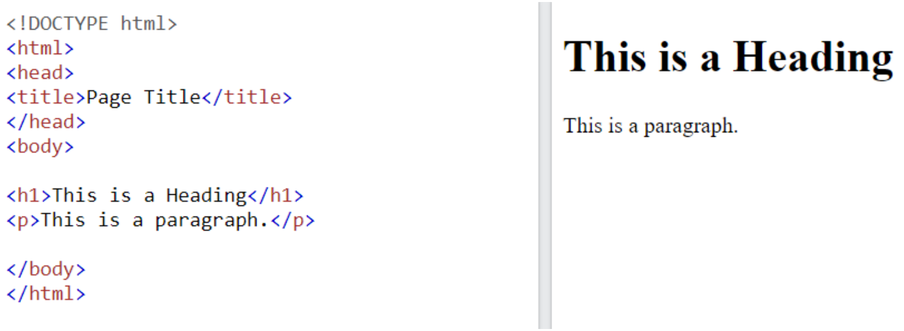


Figure 2.11: I2C Communication Standard

2.11 HTML, CSS, JS, GOOGLE FIREBASE

2.11.1 HTML

HTML stands for Hyper Text Markup Language, HTML is the standard markup language for creating website, HTML describes the structure of a Web page, HTML consists of a series of elements, HTML elements tell the browser how to display the content, HTML elements label pieces of content such as “This is a heading”, “this is a paragraph”. Etc.



```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

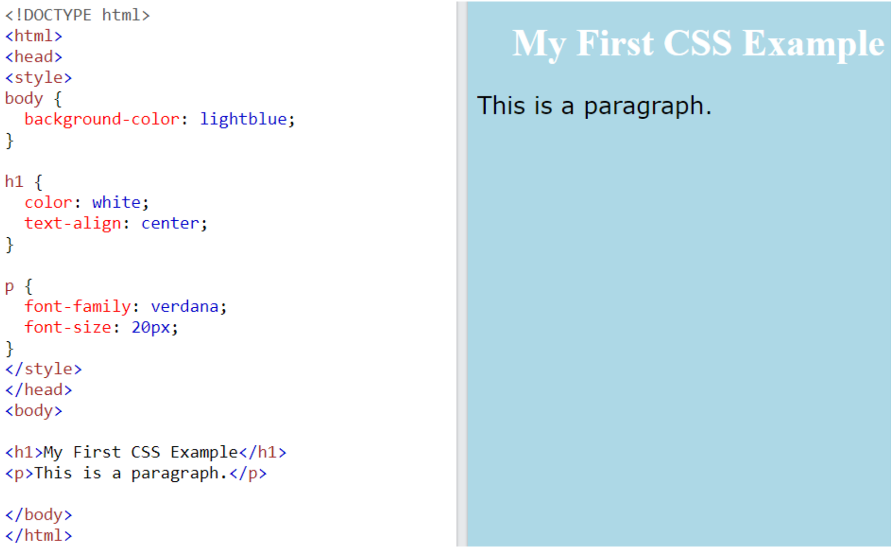
This is a Heading

This is a paragraph.

Figure 2.12: A Simple HTML Document

2.11.2 CSS

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.



```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

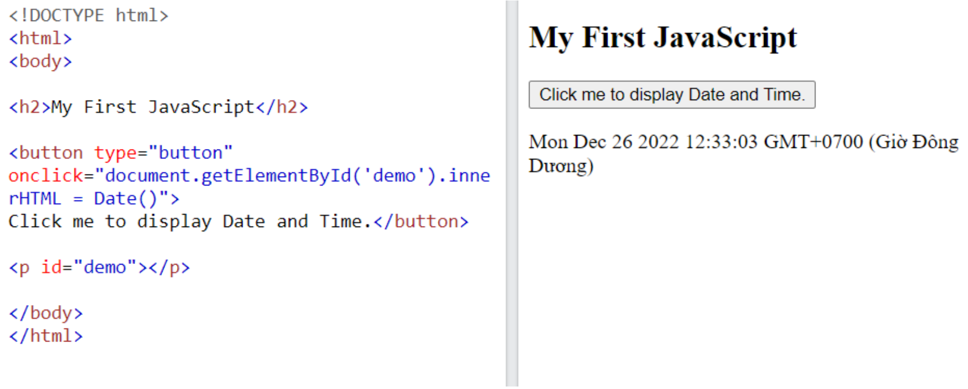
My First CSS Example

This is a paragraph.

Figure 2.13: An example of CSS

2.11.3 JavaScript

JavaScript is the world’s most popular programming language. JavaScript is the programming language of the Web. It is easy to learn.



```
<!DOCTYPE html>
<html>
<body>

<h2>My First JavaScript</h2>

<button type="button"
onclick="document.getElementById('demo').inne
rHTML = Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```

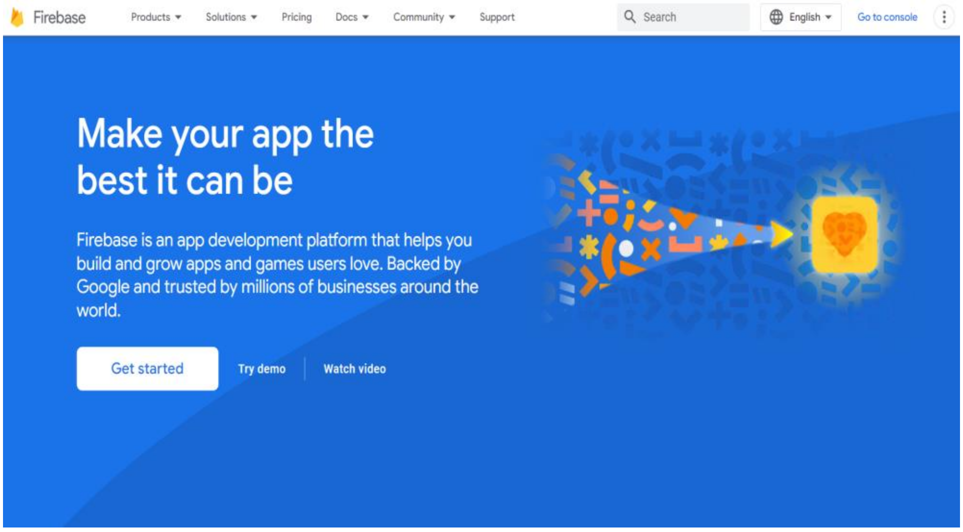
My First JavaScript

Mon Dec 26 2022 12:33:03 GMT+0700 (Giờ Đông Dương)

Figure 2.14: An example of JavaScript

2.11.4 Google Firebase

Google Firebase is a cloud-based database service. Accompanied by an extremely powerful server system of Google. Its main function is to help users program applications by simplifying database operations. Specifically, simple API application programming interfaces. The aim is to increase the number of users and get more profits. In particular, it is also a versatile service and extremely good security. Firebase supports both Android and IOS platforms. It is surprising that many developers choose Firebase as the first platform to build apps for millions of users worldwide.



Make your app the best it can be

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

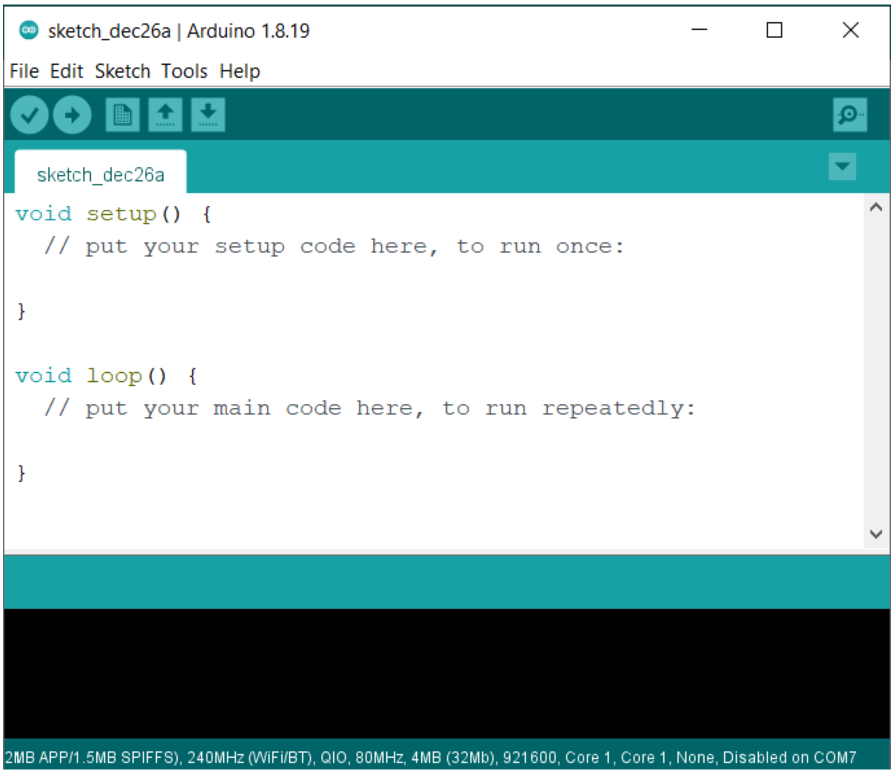
Figure 2.15: Google Firebase Service

2.12 SOFTWARE

2.12.1 Arduino IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment.

To program for Kit ESP32 we used the Arduino IDE.



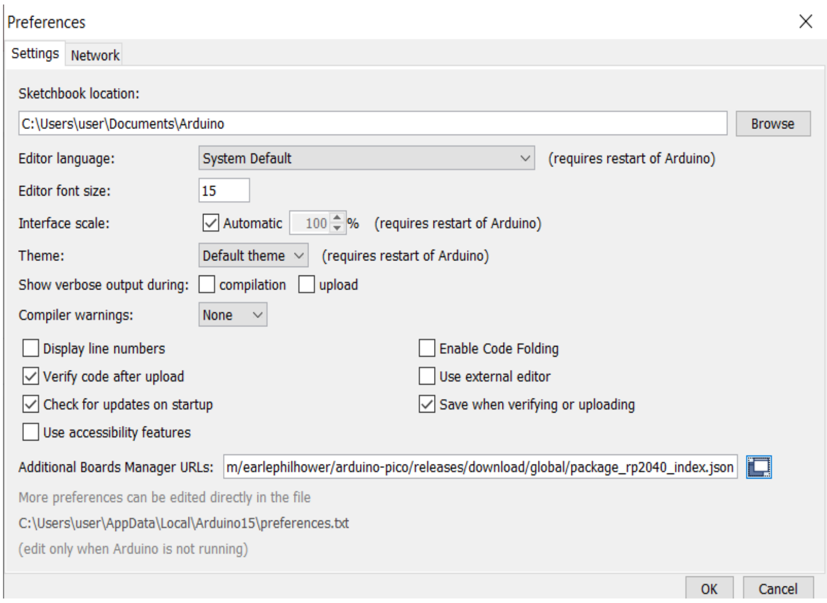
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled on COM7
```

Figure 2.16: Arduino IDE Software

After install the Arduino IDE we also need to go to File->Preferences, copy and paste this URLs line to Additional Boards Manager URLs text box:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



Preferences

Settings | Network

Sketchbook location:

C:\Users\user\Documents\Arduino [Browse]

Editor language: System Default (requires restart of Arduino)

Editor font size: 15

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

☐ Display line numbers

☑ Verify code after upload

☑ Check for updates on startup

☐ Use accessibility features

☐ Enable Code Folding

☐ Use external editor

☑ Save when verifying or uploading

Additional Boards Manager URLs: m/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

More preferences can be edited directly in the file

C:\Users\user\AppData\Local\Arduino15\preferences.txt

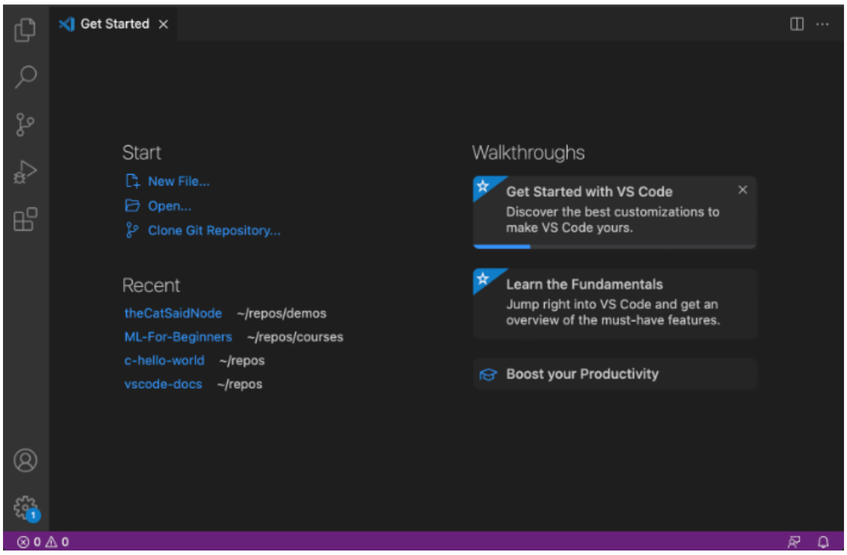
(edit only when Arduino is not running)

[OK] [Cancel]

Figure 2.17: Preferences Table

2.12.2 Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Window, Linux and MacOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. We use VS Code to work with HTML, CSS, JavaScript.



Get Started ×

Start

New File...

Open...

Clone Git Repository...

Recent

theCatSaidNode ~/repos/demos

ML-For-Beginners ~/repos/courses

c-hello-world ~/repos

vscode-docs ~/repos

Walkthroughs

Get Started with VS Code

Discover the best customizations to make VS Code yours.

Learn the Fundamentals

Jump right into VS Code and get an overview of the must-have features.

Boost your Productivity

⊗ 0 ⚠ 0

Figure 2.18: Visual Studio Code

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1 SYSTEM REQUIREMENTS

In order to achieve the set target functions, our team decided to design a lighting control system includes two microcontrollers, two power supply, one LDR sensor module, one RTC DS3231 module, six push button, one DPDT self-locking switch, one LCD 16x2, one I2C LCD interface module, one 2-channel 5V relay module and two 5W LED Bulb.

Two microcontrollers are Kit ESP32 and Kit Arduino Nano. The Kit ESP32 is the central processing microcontroller which is directly control the relay module to turn on or turn off the LED Bulb. And the Kit Arduino Nano performs the timer function.

In this project, we have two modes. There are manual control mode and automatic control mode:

- In manual control mode, we connected two push buttons to the Kit ESP32 for users can turn the relay module on or off so that the light will be turn on or off with physical buttons. The state of the light will also be sent to google firebase for creation of toggle button on the website. The LDR sensor module is also directly connect to the Kit ESP32 so that the Kit Esp32 can easily sent the value of light sensor to google firebase to serve for the automatic control mode, and we also connected one DPDT self-locking switch to Kit ESP32 for changing mode of the system.
- In automatic control mode, we connected second microcontroller to four push buttons, these push buttons allow users set date, time, timer and move up or move down in the LCD menu. One LCD 16x2 is connected to Kit Arduino Nano to display the timer setting, date, time and temperature. One RTC DS3231 module is also connected to Kit Arduino Nano for functions related to real time. In

addition to the function of being able to turn an outdoor light on and off by timer. This mode also allows users to control an indoor light with a LDR sensor module.

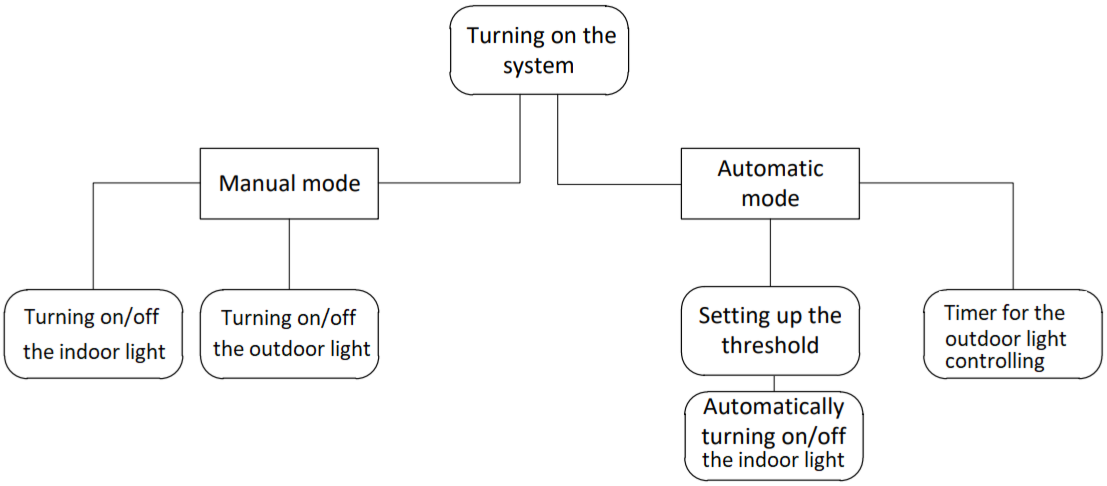

Turning on the system
Manual mode
Automatic mode
Turning on/off the indoor light
Turning on/off the outdoor light
Setting up the threshold
Timer for the outdoor light controlling
Automatically turning on/off the indoor light

Figure 3.1: The operating diagram of the lighting control system

Our group built a google firebase database where stores the data sent from the hardware and from that database the data will be sent to the website.

Our group also built a website for display the value of LDR sensor module, a website which can allow users can control the light bulb remotely. In addition, that website can allow user easily to monitor the status of the indoor light or the outdoor light through the image of the light bulb on the website and the color of the toggle button on the website.

3.2 BLOCK DIAGRAM

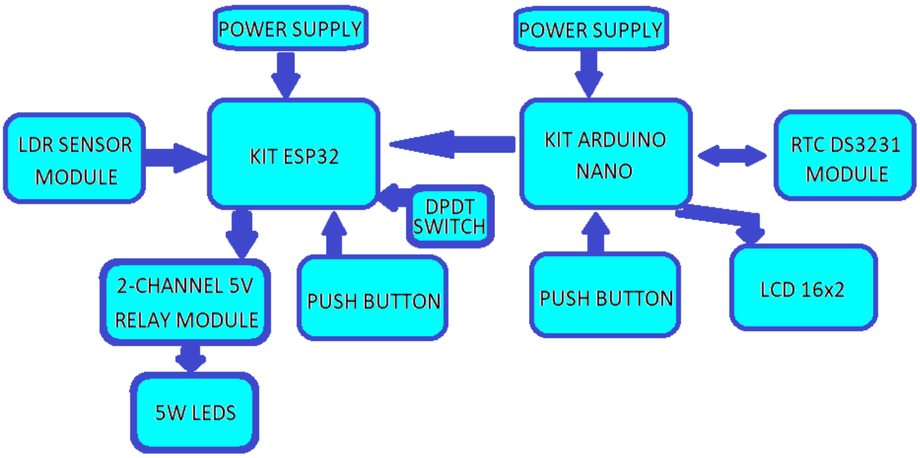


Figure 3.2: Block Diagram of The System

The blocks have specific functions as follows:

- The Kit ESP32 block has the function of collecting and processing data from the LDR sensor block, from the Kit Arduino Nano block and physical push button block for controlling 2-channel 5V relay module block.
- The Kit Arduino Nano block has the function of collecting and processing data from RTC DS3231 module block, from the physical push button block for setting timer, the data read from RTC DS3231 block will be sent to the LCD 16x2 block.
- Power Supply block provide the power for Kit ESP32 block and Kit Arduino Nano block.
- LDR sensor module block which main function is to read data of the light sensor measured in the environment and transmit the data to Kit ESP32 block.
- RTC DS3231 module block which main function is to receive and transmit real-time data with Kit Arduino Nano block.
- DPDT Switch block sent signal to Kit ESP32 block for changing the mode of the system, push button block sent signal for Kit ESP32 can turn the relay module on or of, it is also sent signal for Kit Arduino Nano can set the Date, Time or Timer.
- Display block consisting of LCD 16x2 and I2C LCD interface module.
- 2-channel 5V relay block receives signal from Kit ESP32 block and rely on that signal to control the indoor light and the outdoor light.

3.3 DETAIL HARDWARE DESIGNS

In order for the system to operate stably and in accordance with the target function, our team selected modules with appropriate functions. These modules are interconnected as shown in Figure 3.2:

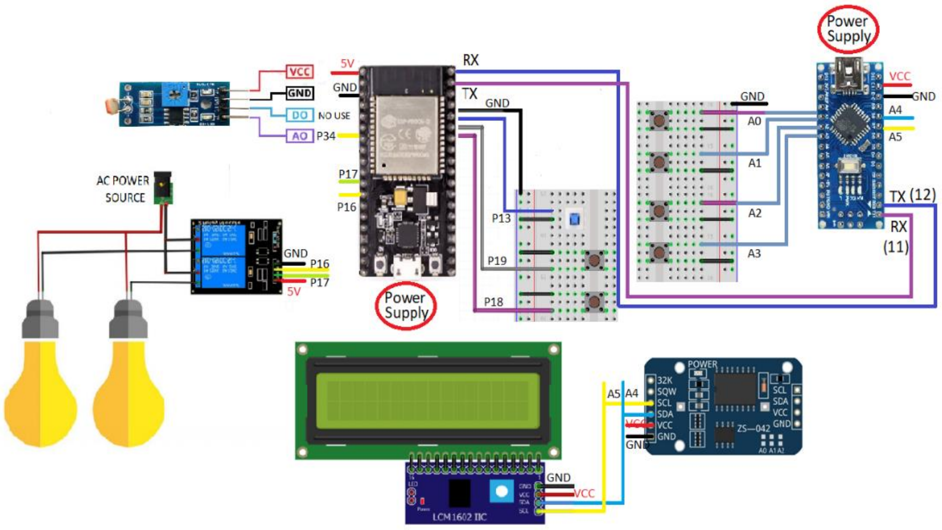


Figure 3.3: Lighting control system hardware diagram

Our group divided the lighting control system into 7 blocks.

3.3.1 Central processing block

Our group selected the Kit ESP32 as the central processing block with the following characteristics:

- Using ESP32-WROOM-32 chip is a chip commonly used in IoT projects with powerful processing chip.
- The Kit ESP32 has 38 GPIO pins and interfaces such as UART/USART, I2C, SPI, CAN, USB to increase flexible connection with other modules.
- The Kit ESP32 work stably and there are many free support libraries.

3.3.2 Second processing block

Our group also selected the Kit Arduino Nano as the second processing block for the timer function of the project, the Kit Arduino Nano block has the following characteristics:

- Using ATmega328 chip is a chip commonly used in IoT projects or Arduino project with powerful processing chip.
- The Kit Arduino Nano has 22 input/output pins in total. 14 of these pins are digital pins. It also has 8 analogue pins and 6 PWM pins among digital pins. So that users can easily use this kit for each of their projects.
- This kit has an easy-to-connect design and has a diverse support library.
- Users can upload code fast for testing new system functions.

3.3.3 LDR sensor module block

LDR sensor module is LDR sensor module block:

- This module detects the presence of measuring the intensity of light in environment and the Kit ESP32 will read this data and send this data to website through google firebase.
- This module has a compact design help users can easily use in their projects.
- This module communicates with the Kit ESP32 by one digital pin(D0) or one analogue pin (A0).

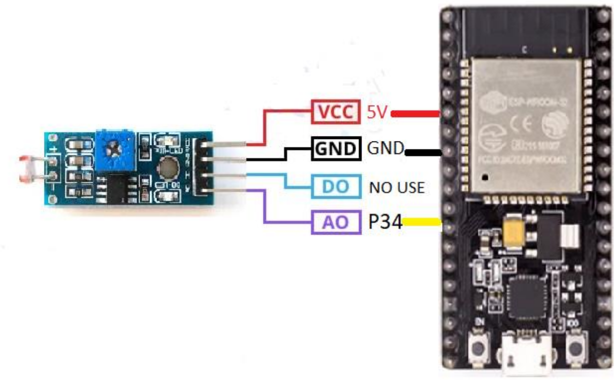


Figure 3.4: Connection diagram of LDR sensor module block

3.3.4 RTC DS3231 module block

RTC DS3231 module is RTC DS3231 block:

- This module communicates with the Kit Arduino Nano block using the I2C protocol.
- This module is an affordable system that manage Time and Date with high precision.

- This small size module has 6 pins, a battery cell and an inbuilt temperature - compensated crystal Oscillator.

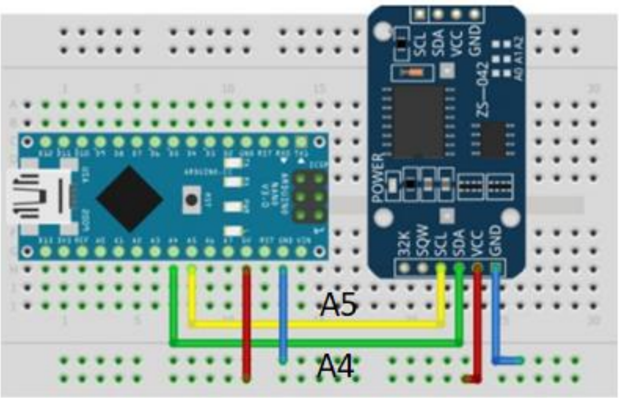


Figure 3.5: Connection diagram of RTC DS3231 module block

3.3.5 Display block

LCD 16x2 with I2C LCD interface module is the display block:

- LCD 16x2 and I2C LCD interface module has a compact design help users can easily use in their projects.
- I2C LCD interface module help users easily to use LCD 16x2 or LCD 20x4. Instead of using SPI protocol to connect lcd to the microcontroller, user will need more than two wire to connect, I2C LCD interface module only needs two wire to connect the LCD 16x2 to the microcontroller.

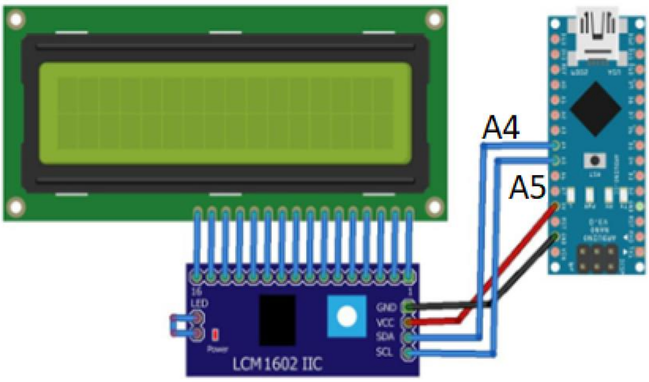


Figure 3.6: Connection diagram of display block

3.3.6 Button block

Push button and DPDT self-locking switch are a button block:

- Users can easily turn the relay on or of by push button, and also can easily change the mode of the system by DPDT self-locking switch.
- Users can easily write code for functions when users use push button and DPDT self-locking switch.

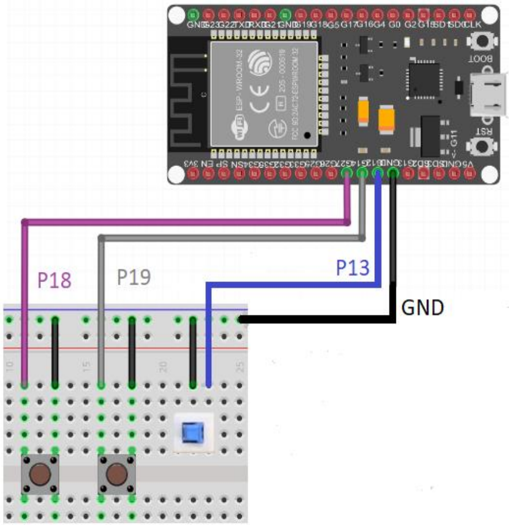


Figure 3.7: Push button, DPDT switch and Kit ESP32 connection diagram

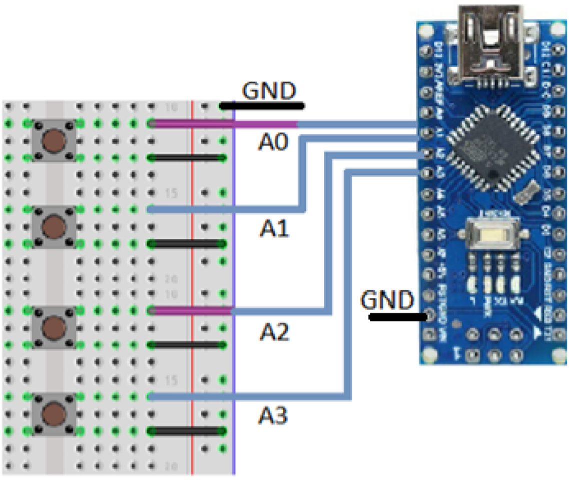


Figure 3.8: Push button and Kit Arduino Nano connection diagram

3.3.7 2-channel 5V relay module block

2-channel 5V relay module is 2-channel 5V relay module block:

- This module is ideal for controlling large AC or DC loads.
- This module helps to protect users and other components from the electrical leakage or faults, which are not designed to handle high current or voltage.

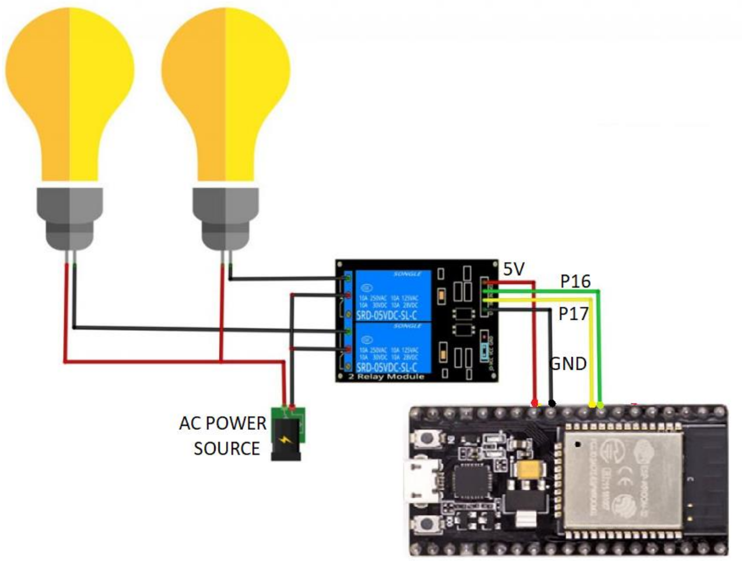


Figure 3.9: Relay module connection diagram

3.3.8 Power supply block

Our group divided the power supply block into two power supply block for two processing block and one ac power source for light bulbs.

- Our group use Micro USB-TTL CH340G Cable for Kit ESP32 and Mini USB Cable for Kit Arduino Nano. Both two USB Cable are all provide 5V for the microcontrollers. And the sensor modules is provided by the power pin of the microcontrollers.
- Our group also use AC power source to provide power for the light bulbs.

3.4 SOFTWARE DESIGNS

3.4.1 Operational functions of the software

With initial objectives, the software will have 4 parts:

- Software for Kit ESP32.
- Software for Kit Arduino Nano.
- Server (Google Firebase)
- Website

The software for the processing blocks is code written in Arduino IDE software, which uses the C language to write processing instructions, controlling the entire hardware of the system.

The Software allows the Kit ESP32 block to control the light bulbs and send data from the sensor modules to server (Google Firebase).

Web page is written in HTML, CSS, JavaScript are written in VS Code Software. There is a function to help users control the lights and monitors the value of the LDR sensor module.

3.4.2 Algorithmic flowchart

In order to help the system operates stably and stay on target our group has restructured the requirements that the system needs to achieve, the problems that need to be process into common algorithmic flowchart for the entire processing system. It is conceptualized as shown in Figure 3.9.

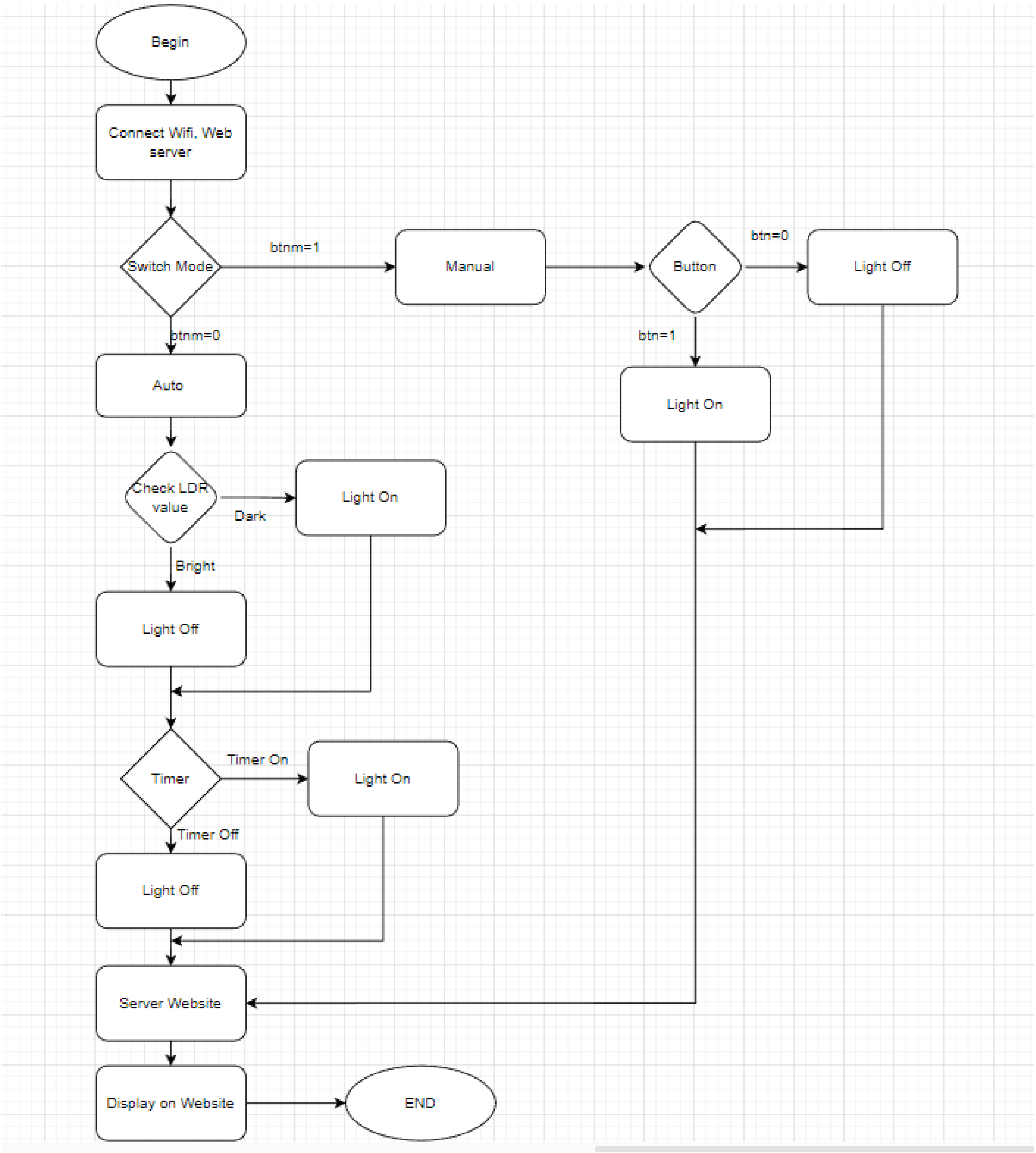


Figure 3.10: Algorithm flowchart

First, when the system starts the Kit ESP32 will connect to the Wifi network, the LCD displays the date, time and temperature.

Next, the lighting control system will let the users choose the operating mode of the system with a switch mode button, if the value of switch button is 1, then allow the system to operate in manual control mode, if the value of the switch button is 0 the system works in automatic control mode.

In manual control mode, users can control the lights with push button, if the value of the button is 0 the light will be turn off, and if the value of the button is 1 the light will be turn on, we also can control the light on the website.

In automatic control mode, the light will be controlled automatically by the value of LDR sensor module or the light will be controlled by timer, users will set up date, time, timer on, timer off for turning on or turning off the light.

With LDR module, if the surrounding environment is dark mean the value of LDR sensor module greater than 1500 the indoor light will be turned on. And if the value is less than 1500 means the surrounding environment is bright so the light will be turned off. The value of LDR sensor module is sent to the website via google firebase.

With timer the Kit Arduino Nano is connected to the Kit ESP32 by using UART protocol. The Kit Arduino Nano has the task of timing to turn on or off the outdoor light. When the timer finished counting, it will send the command sequence to the Kit ESP32 to notify that the timing has done. When the Kit ESP32 receives the signal from the Kit Arduino Nano, it will output the light on or off depending on the signal it received.

3.4.3 Server (Google Firebase)

Kit ESP32 block is not only a central processing block but it also a network communication block that can be connected to the server via the internet.

The processing flowchart of the network communication block is depicted in the figure 3.10.

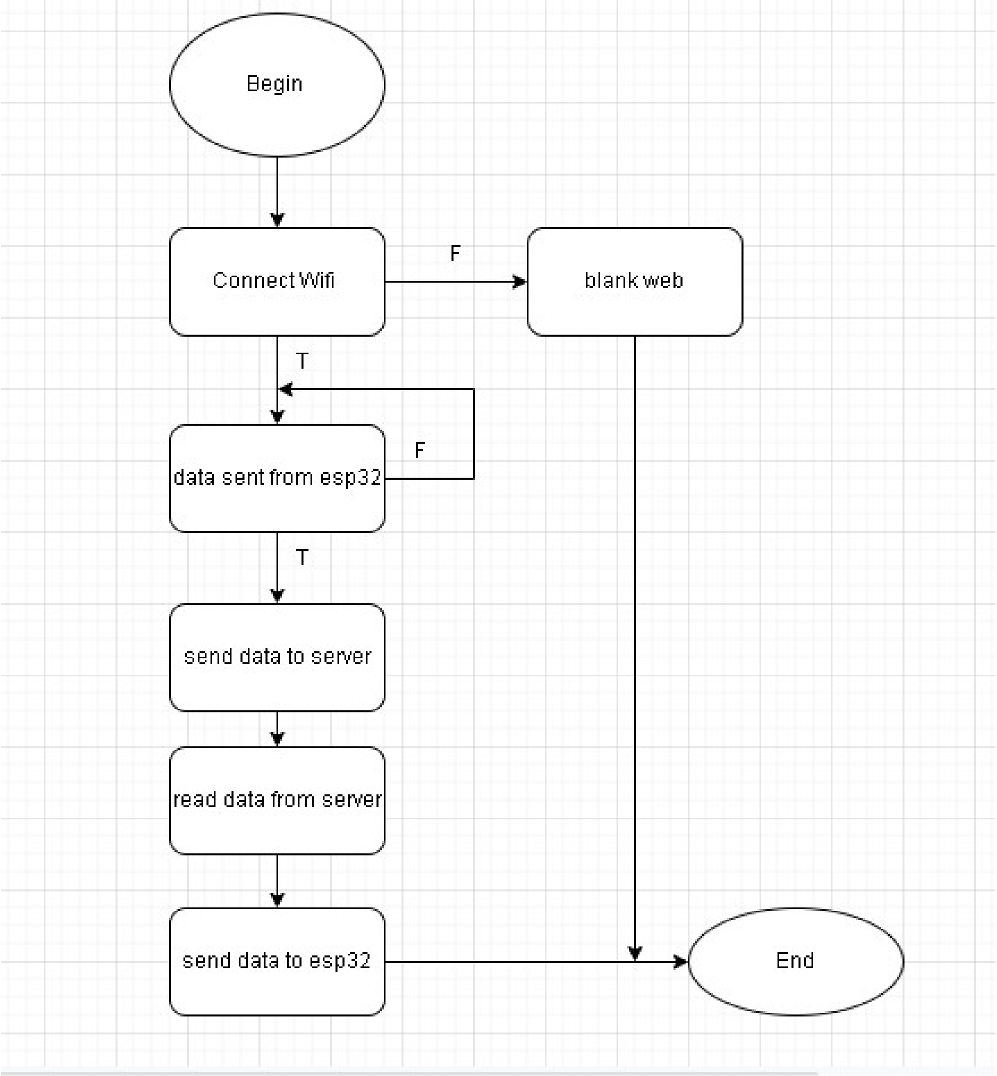


Figure 3.11: The communication network block flowchart

The laptop and the Kit ESP32 are connected to the same Wifi network to synchronize the data.

Kit ESP32 will receives and processes data from LDR sensor module and send it to website via google firebase. The state of the lights will also be sent to google firebase to create toggle buttons on the website.

To create a google firebase database, we need to follow these steps:

- As a first, visit https://firebase.google.com/ to sign up for an account and then create new project.

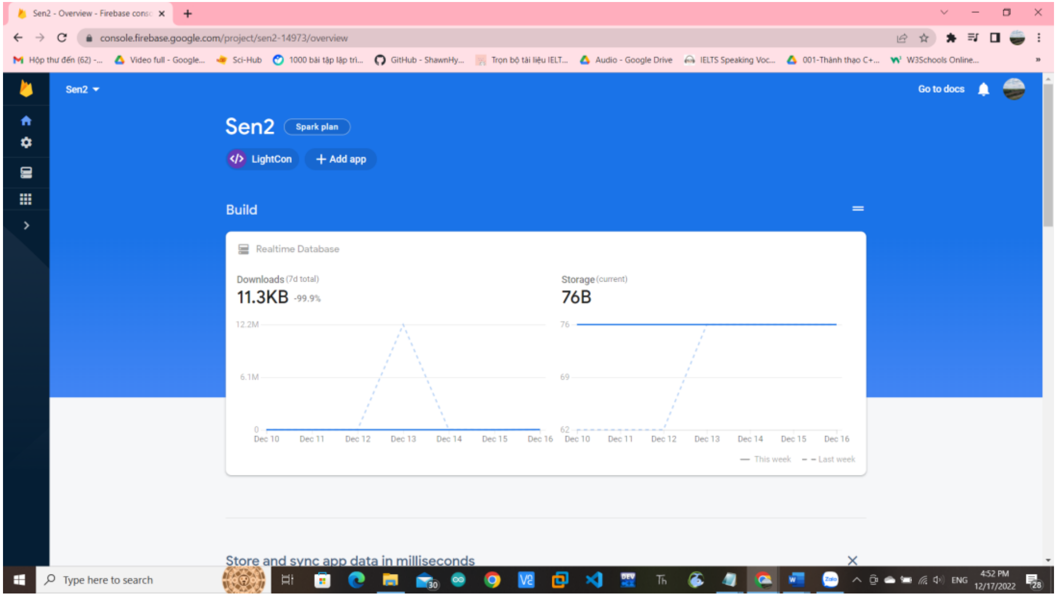
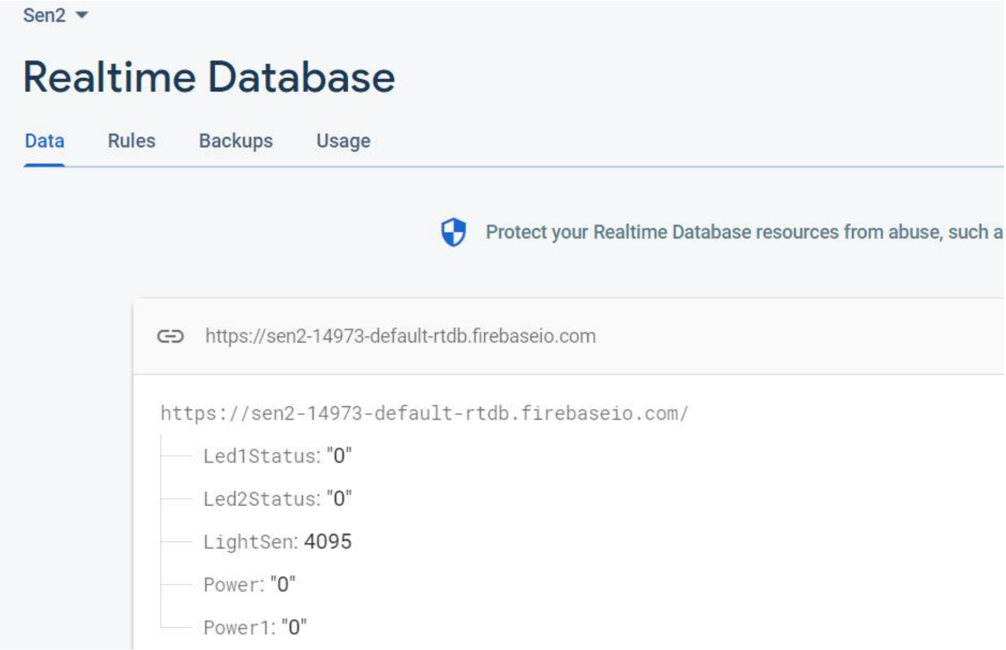


Figure 3.12: Create new project

- Secondly, initialize parameters on the application. And copy https line code.



Sen2

Realtime Database

Data Rules Backups Usage

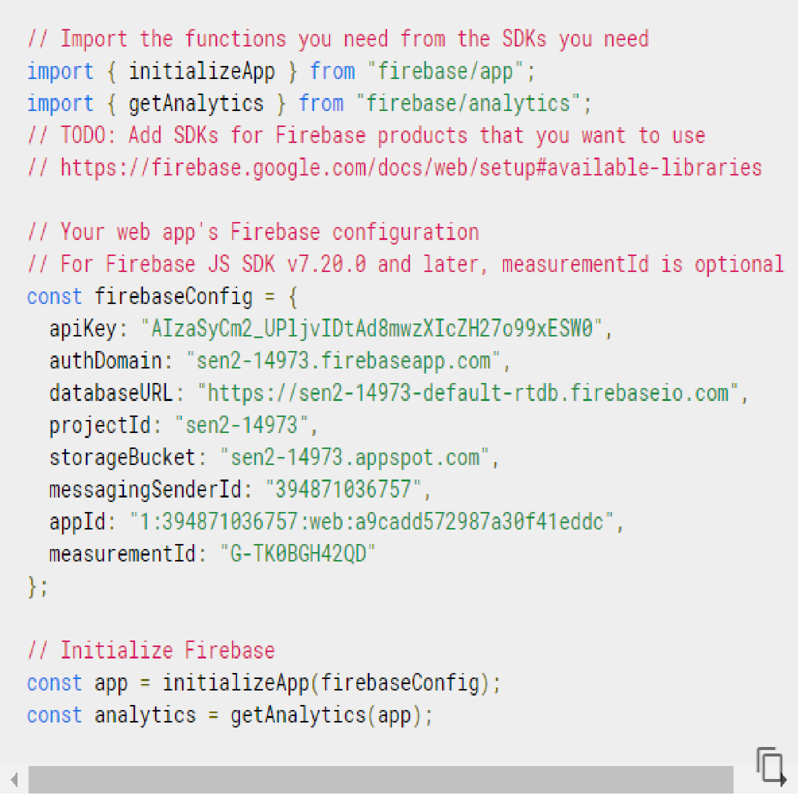
Protect your Realtime Database resources from abuse, such a

https://sen2-14973-default-rtdb.firebaseio.com

```
https://sen2-14973-default-rtdb.firebaseio.com/
  Led1Status: "0"
  Led2Status: "0"
  LightSen: 4095
  Power: "0"
  Power1: "0"
```

Figure 3.13: Initialize parameters on the application

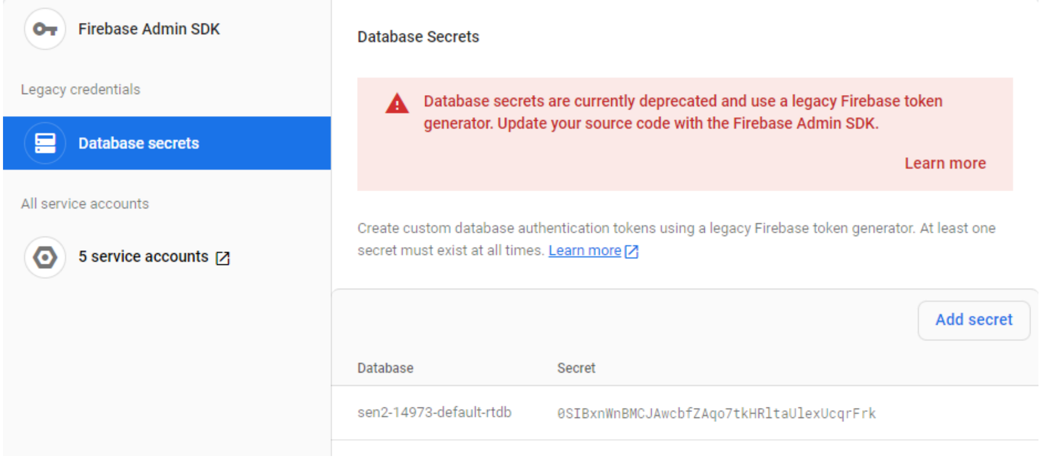
- After completing the second step, we move to Project Overview and choose add an apps, then choose add web. After register name for the application, we will see a script of code, we need to copy this script of code and use it to link to the website.



```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyCm2_UPljvIDtAd8mwzXIcZH27o99xESW0",
  authDomain: "sen2-14973.firebaseapp.com",
  databaseURL: "https://sen2-14973-default-rtdb.firebaseio.com",
  projectId: "sen2-14973",
  storageBucket: "sen2-14973.appspot.com",
  messagingSenderId: "394871036757",
  appId: "1:394871036757:web:a9cadd572987a30f41eddc",
  measurementId: "G-TK0BGH42QD"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

Figure 3.14: Script of code

- In project setting move to service account copy the database secret code which use to link the Kit ESP32 to google firebase.



Database Secrets

Database secrets are currently deprecated and use a legacy Firebase token generator. Update your source code with the Firebase Admin SDK.

Learn more

Create custom database authentication tokens using a legacy Firebase token generator. At least one secret must exist at all times. Learn more

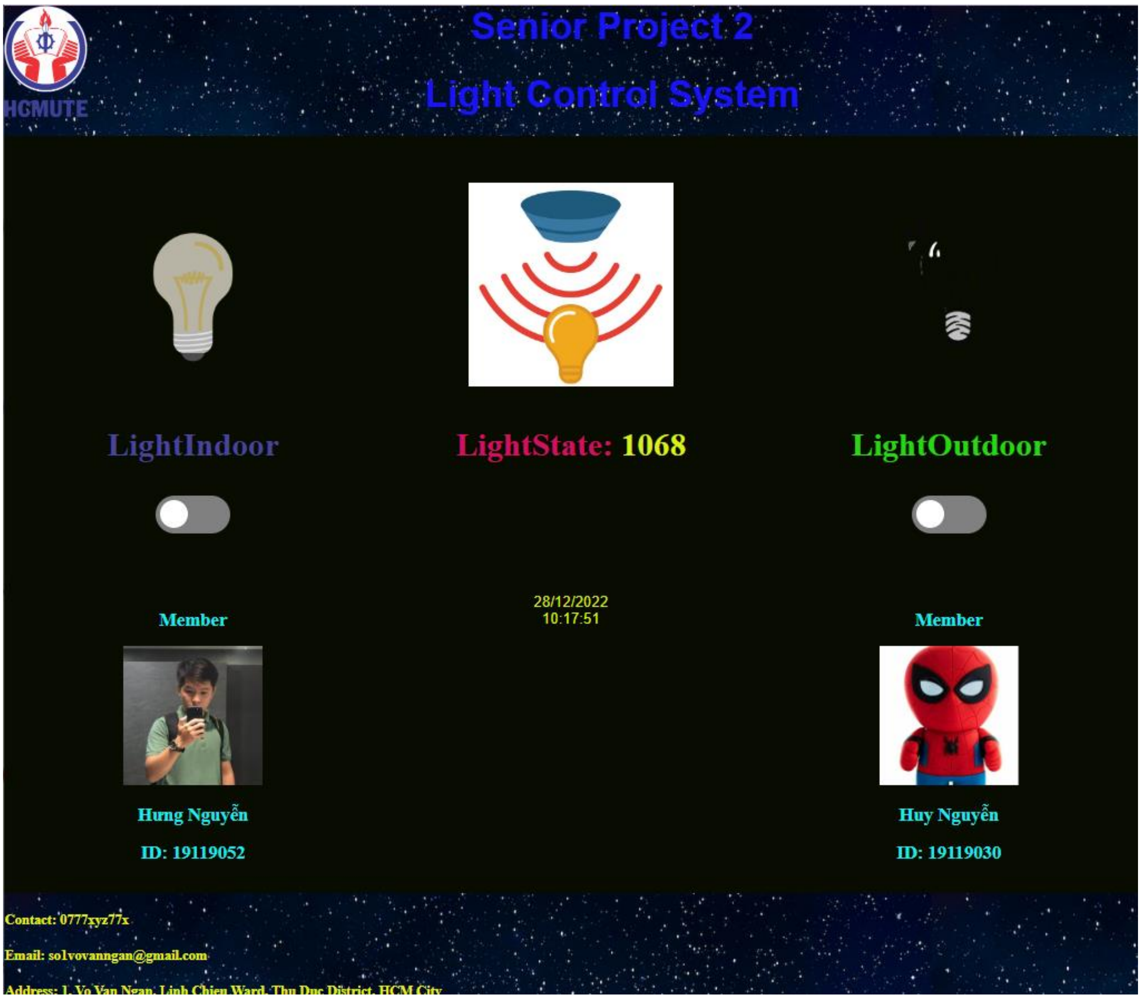
Add secret

Database	Secret
sen2-14973-default-rtdb	0SIBxnWnBMCJAwcbfZAqo7tkHRltaUlexUcqrFrk

Figure 3.15: Database secret code

3.4.4 Website

Aiming to create interaction between the users and the lighting control system through Internet, our group built a website that functions such as turning lights on or off, displaying the value of LDR sensor module.



HCMUTE

Senior Project 2

Light Control System

LightIndoor

LightState: 1068

LightOutdoor

28/12/2022
10:17:51

Member

Hưng Nguyễn

ID: 19119052

Member

Huy Nguyễn

ID: 19119030

Contact: 0777xyz77x

Email: so1vovanngan@gmail.com

Address: 1, Vo Van Ngan, Linh Chieu Ward, Thu Duc District, HCM City

Figure 3.16: Website of lighting control system

CHAPTER 4

RESULTS

4.1 LIGHTING CONTROL SYSTEM IMPLEMENTATION

After the implementation process, our group completed the model of the system as shown in figure 4.1.

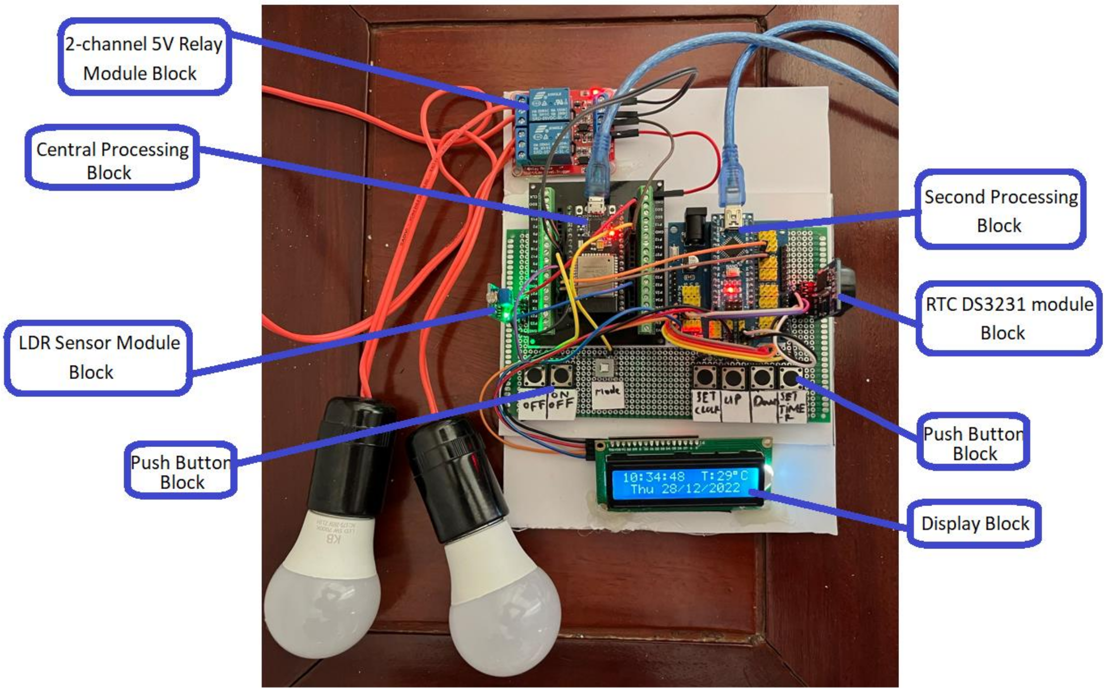
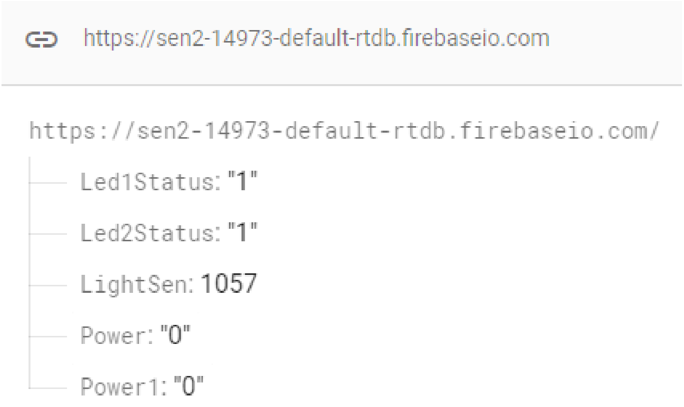
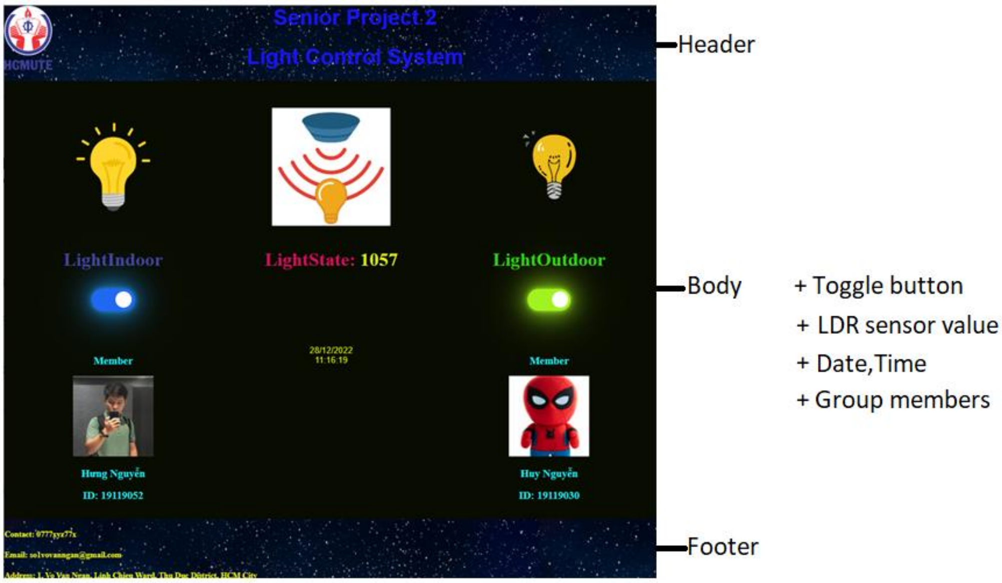


Figure 4.1: Model of lighting control system



```
https://sen2-14973-default-rtdb.firebaseio.com

https://sen2-14973-default-rtdb.firebaseio.com/
├── Led1Status: "1"
├── Led2Status: "1"
├── LightSen: 1057
├── Power: "0"
└── Power1: "0"
```

Figure 4.2: Database



Header

Body

+ Toggle button
+ LDR sensor value
+ Date,Time
+ Group members

Footer

Figure 4.3: Website Interface

4.2 SYSTEM OPERATION

The process of operation of the lighting control system is described as follows:

After starting the system will allow users to choose one of the two operating modes of the system by the DPDT self-locking switch.

Figure 4.4: Mode button

If the mode button is pressed means the lighting control system is in manual control mode, on the contrary, the lighting control system is in automatic control mode.

In manual mode, users can directly control the lights by physical push button or toggle buttons on the website.

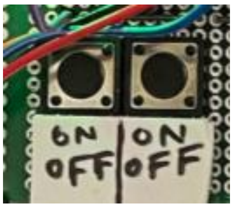


Figure 4.5: Physical push buttons

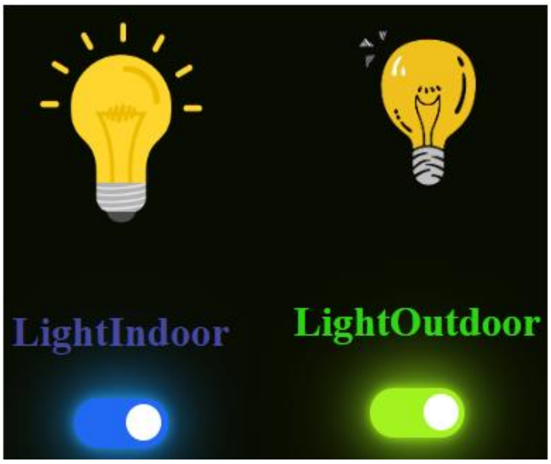


Figure 4.6: Toggle buttons

These buttons are to turn on or of the lights manually in manual control mode.

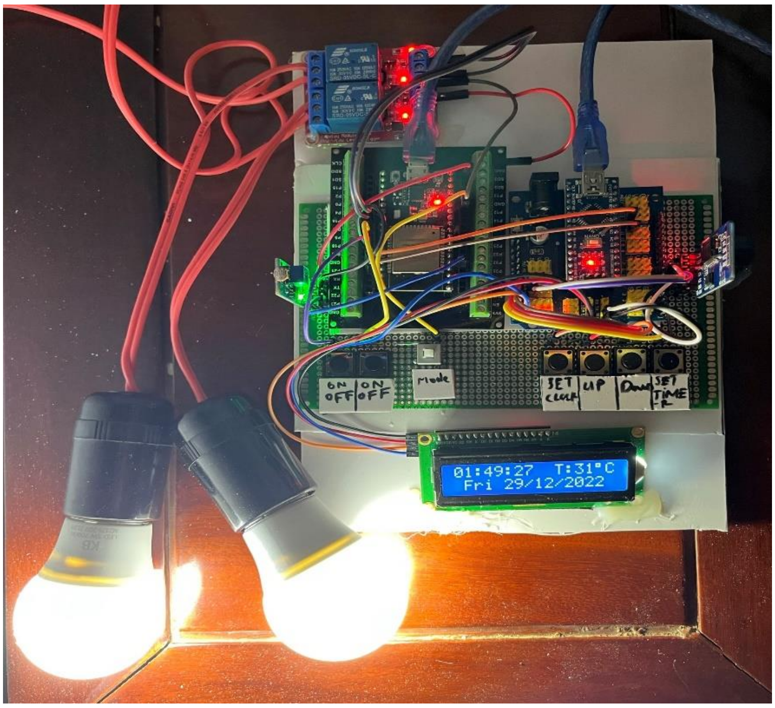
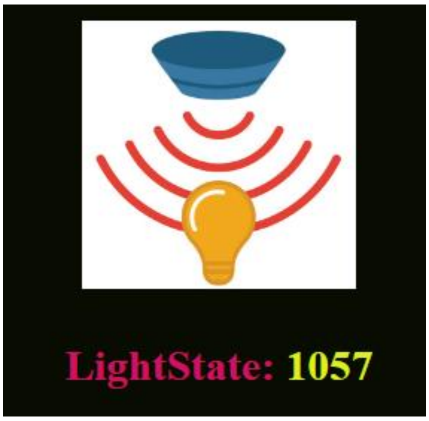


Figure 4.7: The lights are turned on in manual mode

The value of LDR sensor module will be always update real-time to google firebase. This value is to control the indoor light in automatic mode.



LightState: 1057

Figure 4.8: LDR sensor value

In automatic control mode, the system will automatically control the indoor light by LDR sensor value, and the outdoor light by timer.

The indoor light will be automatically turned on if the surrounding environment is dark or it will be automatically turned off if the surrounding environment is bright.

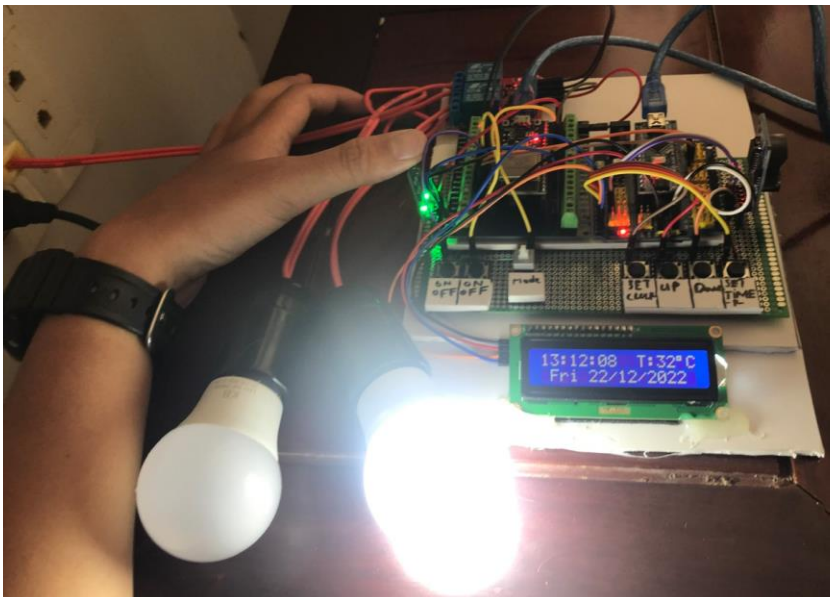


Figure 4.9: The indoor light is automatically turned on.

To control the outdoor light by timer the users need to set date, time, timer on, timer off for the Kit Arduino Nano.

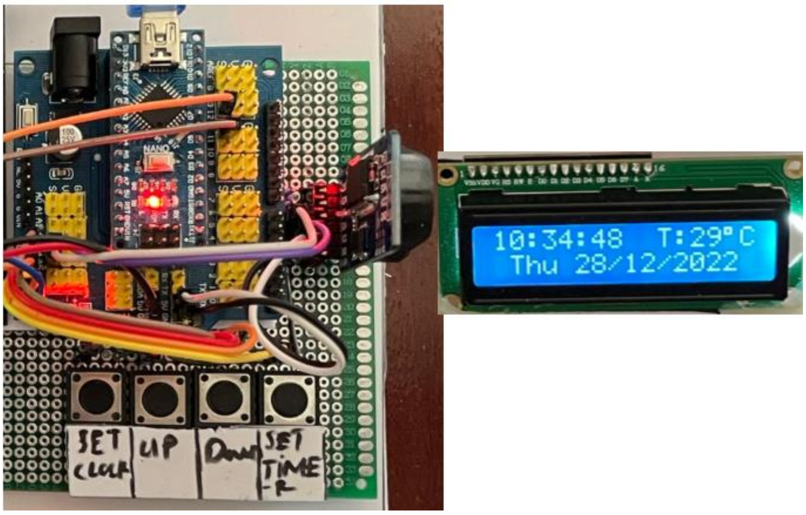
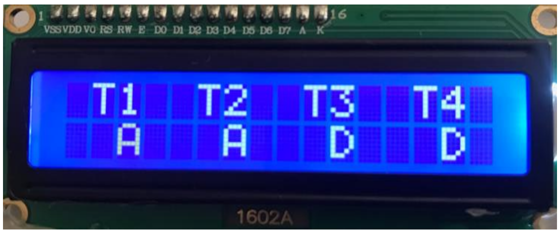


Figure 4.10: Timer block

To set up date, time, timer on, timer off, users use these four push buttons. The date, time, timer on, timer will be show in the lcd display block.

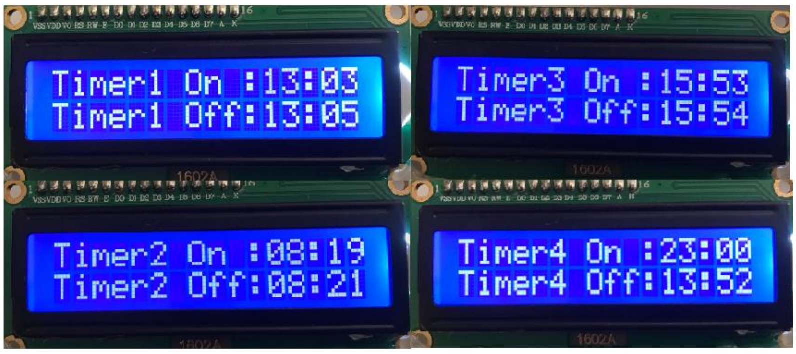
When user press the set timer button. The lcd will display the timer menu option, we have four timer setting. If in the LCD the row below the timer option is A, it means timer function is active, by contrast to D it means Timer function is not active.



```
T1  T2  T3  T4
 A   A   D   D
```

Figure 4.11: Timer options menu

After users decided to chose which timer is active the lcd will display the hour and minute for turning on or off the outdoor light.



```
Timer1 On :13:03
Timer1 Off:13:05
```

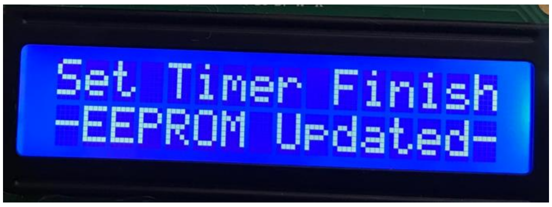
```
Timer3 On :15:53
Timer3 Off:15:54
```

```
Timer2 On :08:19
Timer2 Off:08:21
```

```
Timer4 On :23:00
Timer4 Off:13:52
```

Figure 4.12: Timer setting details

After successful setting date, time, timer on and timer off, the lcd will display the text informs the users.



```
Set Timer Finish
-EEPROM Updated-
```

Figure 4.13: Setting successfully

When the timer finished, the Kit Arduino Nano will send a signal to the Kit ESP32, then the Kit ESP can also turn on/off the outdoor light.

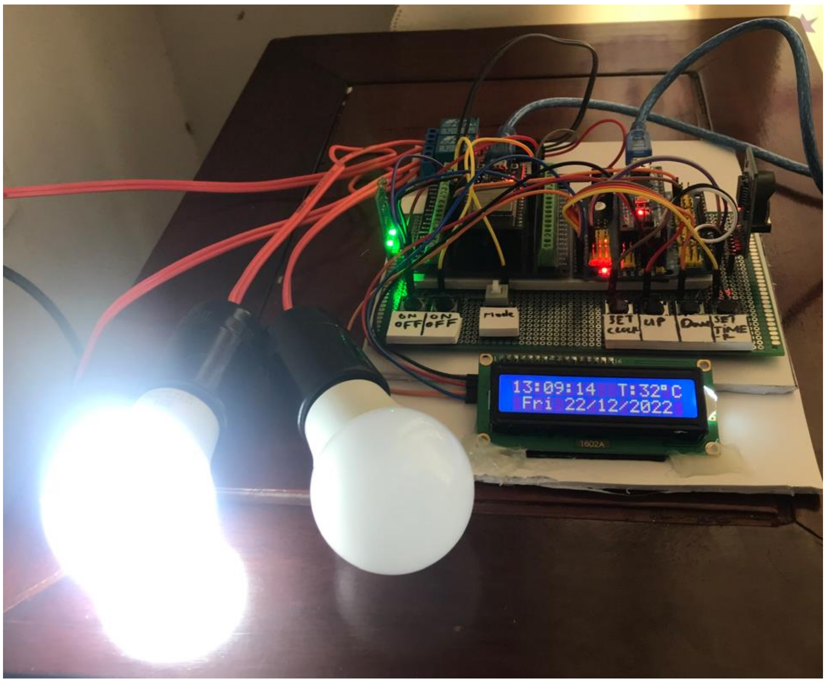


Figure 4.14: The outdoor light is turned on by timer

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

After a period of research, theoretical basis research and implementation, our group has successfully built a lighting control system that meets the initial objectives:

- Successfully build a hardware board processing model for lighting control system operated with manual control mode and automatic control mode.
- Successfully design databases for the lighting control system.
- Successfully design a website for user can control the lights manually and monitor the value of light sensor.

After completion, the system has the following advantages:

- Instead of only being able to turn the lights on or off with physical push button now users can turn the lights on or off remotely via the Internet.
- In addition, users can choose the automatic mode so that the light is automatically controlled by the LDR sensor module or automatically turn on or off the lights by timer.

However, the lighting control system has the following problems:

- The system stability depends on the internet connection.
- The wires used to connect module in the system are not good, so there is still latency.

5.2 FUTURE WORK

If further development, the lighting control system testing can be optimized to make the more stable while building some more functions:

- Replace higher-tech modules to improve system accuracy and improve data exchange speed with databases server.

- In addition to website control, it is possible to design a smartphone application for users have more choices to use the system.
- Can be applied to other equipment in the house such as fans, air conditioners, air purifiers.

REFERENCES

[1] Studiopieter, "ESP32-PINOUT", Sep 2022 [Online], Available: https://www.studiopieters.nl/esp32-pinout/

[2] David watson, "Introduction to Arduino Nano", Nov 2022 [Online], Available: https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html

[3] ELECTRODUINO, "LDR sensor module | How LDR Sensor Works", Oct 2022 [Online], Available: https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html

[4] Components101, "16x2 LCD pinout datasheet", Sep 2022 [Online], Available: https://components101.com/displays/16x2-lcd-pinout-datasheet

[5] Akshay Joseph, "Interface I2C 16x2 LCD with Arduino Uno", Sep 2022 [Online], Available: https://create.arduino.cc/projecthub/akshayjoseph666/interface-i2c-16x2-lcd-with-arduino-uno-just-4-wires-273b24

[6] IoTMAKER, "Module Relay 2-Channel 5V", Sep 2022 [Online], Available: https://iotmaker.vn/relay-2-channel-5v.html

[7] Last Minute ENGINEERS, "DS3231 RTC Arduino Tutorial", Oct 2022 [Online], Available: https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial/

[8] Components101, "Push button", Sep 2022 [Online], Available: https://components101.com/switches/push-button

[9] Utkarsh Verma, "6 Pin Push Switch Working", Sep 2022 [Online], Available: https://www.hackster.io/UtkarshVerma/6-pin-push-switch-working-7bb5ef