ABSTRACT

ABSTRACT

The Home Automation Project Using Bluetooth Control By Mobile Phones Is A Modern Solution That Enables Homeowners To Control And Manage Various Electronic Devices In Their Homes Using Their Smartphones Or Other Bluetooth-enabled Devices. The Project Involves The Installation Of Bluetooth Modules To Electronic Devices, Such As Lighting Systems, And Security Systems. These Modules Are Connected To A Microcontroller Or Other Control System, Which Communicates With A Mobile Application Installed On The Homeowner's Smartphone. The Mobile Application Provides A User-friendly Interface That Allows Homeowners To Access And Control Their Home Appliances And Devices Easily.

Overall, The Home Automation Project Using Bluetooth Control By Mobile Phones Is An Excellent Investment For Any Homeowner Looking To Automate Their Home And Improve Their Quality Of Life. With The Increasing Popularity Of Smartphones And Bluetooth Technology, This Type Of Home Automation System Is Becoming More Accessible And Affordable To Homeowners. The Project Offers A Simple Yet Effective Solution For Controlling And Managing Various Devices In The Home, Offering Greater Convenience, Comfort, And Security To Homeowners.

INTRODUCTION

CHAPTER 1

INTRODUCTION

Home Automation Using Bluetooth Control By Mobile Phones Is A Project That Allows Homeowners To Control And Manage Electronic Devices In Their Homes Using Their Smartphones Or Other Bluetooth-enabled Devices. The System Uses Bluetooth Modules To Connect The Electronic Devices In The Home With A Mobile Application Installed On The Homeowner's Smartphone. The Mobile Application Provides An Intuitive Interface That Allows Users To Remotely Control The Devices, Such As Turning On Or Off Lights, Fan, Air Conditioner Like This. This Project Provides Greater Convenience, Comfort, And Security To Homeowners, And It Is Becoming More Accessible And Affordable With The Increasing Popularity Of Smartphones And Bluetooth Technology.

SYSTEM STUDY

CHAPTER 2
SYSTEM STUDY

2.1 EXISTING SYSTEM

The existing home automation systems using Bluetooth control by mobile phones are typically built using custom hardware and software, and require professional installation and programming. These systems are expensive and complicated, and are usually designed for high-end homes or commercial applications. The existing home automation systems using Bluetooth control by mobile phones are limited by their complexity, cost, and lack of flexibility. There is a need for a simpler, more affordable, and user-friendly home automation system that is accessible to the average homeowner. The project aims to address these limitations by providing an easy-to-install, cost-effective, and flexible home automation system that can be controlled using a mobile phone or other Bluetooth-enabled devices.

DISADVANTAGES

- Compatibility issues: Bluetooth devices may not be compatible with all mobile devices, or vice versa. This can limit your options when it comes to choosing home automation devices that work with your mobile device.
- Battery drain: Using Bluetooth to control your home automation devices can drain the battery life of your mobile device more quickly.
- Compatibility issues: Bluetooth devices may not be compatible with all mobile devices, or vice versa. This can limit your options when it comes to choosing home automation devices that work with your mobile device.

2.2 PROPOSED SYSTEM

The proposed system project for home automation using Bluetooth controlled by mobile devices provides a convenient and customizable way to manage various aspects of the home, while also reducing energy consumption and improving efficiency. With the right components and implementation, the system can provide a secure and reliable home automation solution that is easy to install and use.

ADVANTAGES

- Convenience
- Energy efficiency
- Improved home security
- Personalization
- Scalability

SYSTEM REQUIREMENT

CHAPTER 3

SYSTEM REQUIREMENT

3.1 HARDWARE REQUIREMENTS

- Arduino uno
- Relay module
- Bread board
- Bluetooth module-hc05
- Male to female jumper wires
- Male to male jumper wires

3.2 SOFTWARE REQUIREMENTS

- ARDUINO IDE
- GUI-APPLICATION

PROJECT DESCRIPTION

CHAPTER 4

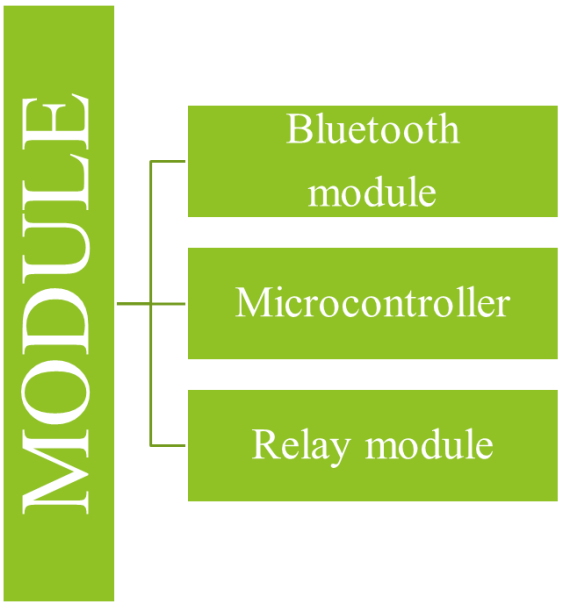
PROJECT DESCRIPTION

4.1 PROJECT DESCRIPTION

The proposed project aims to design and implement a home automation system that can be controlled using Bluetooth-enabled mobile phones. The system will be designed to control various appliances and devices in a home, such as lights, fans, and air conditioners, using a mobile phone app. The system will consist of a Bluetooth module connected to a microcontroller, which will receive commands from the mobile phone app and execute them accordingly. The app will allow users to remotely control the connected devices, set schedules, and receive notifications about the status of the appliances.

4.2 MODULES

This project consist of following modules



4.3 MODULE DESCRIPTION

4.3.1 BLUETOOTH MODULE:

This module acts as a communication interface between the mobile app and the home automation system. It receives commands from the mobile app and sends them to the microcontroller to control various devices.

4.3.2 MICROCONTROLLER:

The microcontroller is the heart of the home automation system. It receives commands from the Bluetooth module and sends signals to various devices such as lights, fans, and appliances.

4.3.3 RELAY MODULE:

This module is used to switch ON/OFF the devices connected to it. It consists of a relay and a driver circuit that is controlled by the microcontroller.

SYSTEM DESIGN

CHAPTER 5

SYSTEM DESIGN

5.1 DATA FLOW

Here is a simple data flow diagram that explains the flow of data in a home automation system using Bluetooth with an application:

5.1.1 USER INTERFACE:

The user interface is the part of the system that the user interacts with. It is typically an application running on a smartphone or tablet. The user can control various aspects of their home, such as lighting, temperature, and appliances, by sending commands to the system.

5.1.2 BLUETOOTH COMMUNICATION:

The user interface sends Bluetooth commands to the Bluetooth module connected to the home automation system. The Bluetooth module receives these commands and forwards them to the microcontroller or other control device in the home automation system.

5.1.3 CONTROL DEVICE:

The microcontroller or other control device processes the Bluetooth commands and triggers the appropriate actions based on the user's input. For example, if the user sends a command to turn on the lights, the microcontroller will trigger the relay or other switching device to turn on the lights.

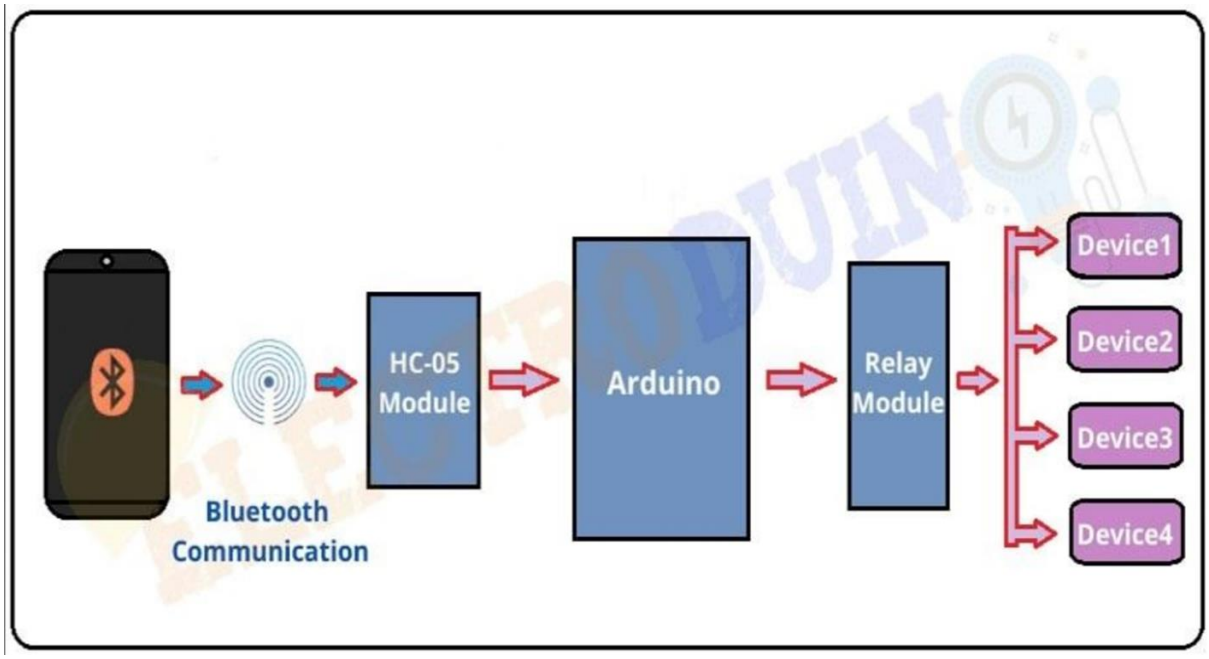
5.1.4 STATUS UPDATE:

The home automation system sends status updates back to the user interface via Bluetooth. For example, if the user sends a command to turn on the lights, the user interface will receive a status update indicating that the lights are now on.

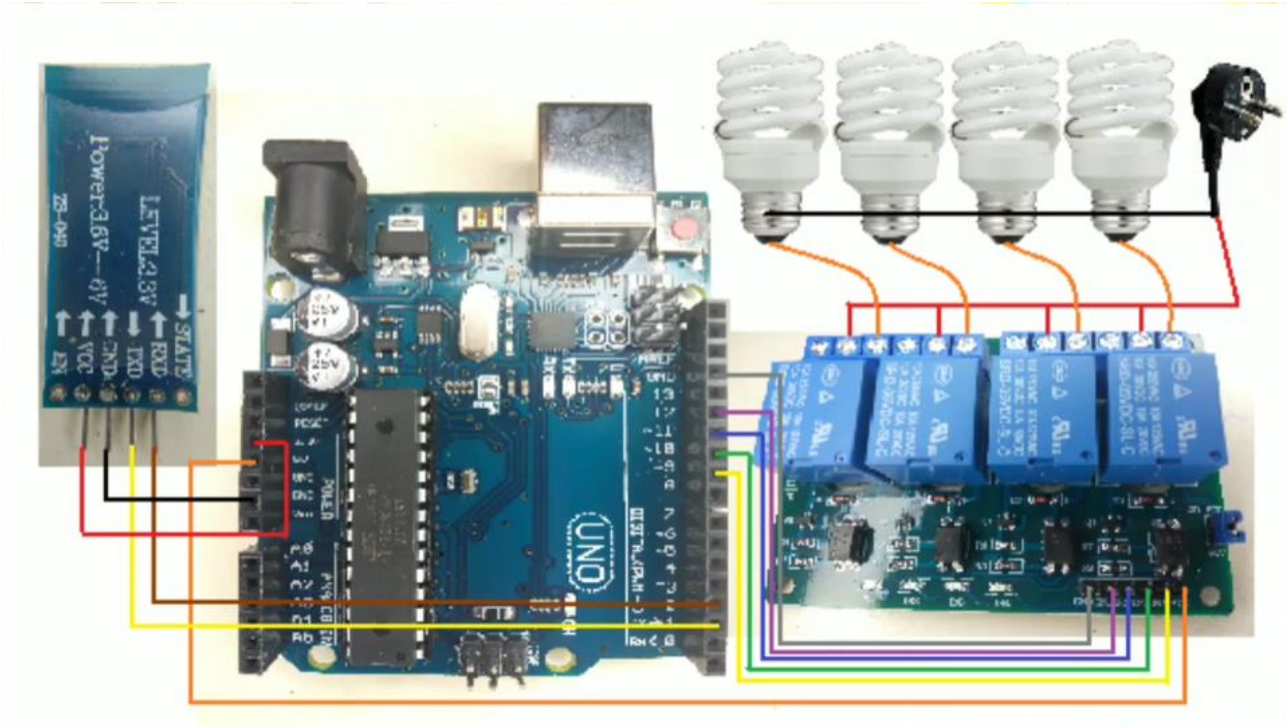
5.1.5 USER INTERFACE UPDATE:

The user interface updates the user interface on the smartphone or tablet to reflect the current status of the home automation system. For example, if the user turns on the lights, the application will display a message indicating that the lights are now on.

5.2 SYSTEM FLOW DIAGRAM



5.3 CIRCUIT DIAGRAM



SOFTWARE DESCRIPTION

CHAPTER 6

SOFTWARE DESCRIPTION

6.1 Arduino IDE

Arduino IDE (Integrated Development Environment) is a software platform used to program and develop code for Arduino microcontroller boards. The Arduino IDE provides a simple and user-friendly interface for writing, compiling, and uploading code to the Arduino board. It is based on the Processing programming language and is open-source software, meaning that anyone can contribute to its development and use it for free.

The Arduino IDE features a text editor for writing code, a message console for displaying debug information and output, and a set of tools for compiling and uploading code to the Arduino board. It also includes a library of pre-written code called "sketches" that can be used as templates for developing new projects. The IDE is compatible with several programming languages, including C, C++, and Java.

One of the main advantages of using the Arduino IDE is its simplicity and ease of use, making it accessible to beginners and experts alike. The IDE also allows users to easily test and debug code, as well as connect to and interact with external hardware components. Overall, the Arduino IDE is a powerful tool for developing and programming Arduino-based projects.

6.2 GUI-APPLICATION

GUI (Graphical User Interface) application in home automation using Bluetooth is a software platform that provides a user-friendly interface for controlling and monitoring home automation systems through a computer or mobile device. The GUI application communicates with the home automation system using Bluetooth technology, allowing users to control and monitor the different devices and sensors in the system.

The GUI application typically includes a set of buttons, sliders, and menus that allow users to turn on/off lights, adjust the temperature, and control other devices in the home automation system. It may also display real-time data from the system, such as temperature and humidity readings or motion detection alerts.

One of the main advantages of using a GUI application for home automation is its ease of use and accessibility. Users can control and monitor their home automation systems from their computer or mobile device, making it convenient and easy to use. Additionally, GUI applications can be customised to meet the specific needs of users and their home automation systems, allowing for greater flexibility and functionality.

6.3 MANAGE CODE

Arduino IDE provides a user-friendly interface for managing and developing code for Arduino microcontroller boards. The IDE includes several features and tools to help users write, compile, and upload code to the Arduino board.

To manage code in the Arduino IDE, users typically follow the following steps:

- Open the Arduino IDE and create a new sketch. The sketch is the file that contains the code for the Arduino board.
- Write the code for the Arduino board using the built-in text editor. The IDE includes features such as syntax highlighting and autocompletion to make coding easier.
- Verify and compile the code by clicking on the "Verify" button. This checks the code for errors and ensures that it is compatible with the Arduino board.

- Upload the code to the Arduino board by clicking on the "Upload" button. This transfers the code to the board and allows it to be executed.
- Monitor the output of the code by viewing the message console. This displays any debug information or output from the Arduino board.
- Test and refine the code as necessary. The IDE allows users to modify and refine the code as needed, and then repeat the process of verifying, compiling, and uploading the code.

6.4 MANAGE DATA

Arduino IDE provides several ways to manage data when programming and developing code for Arduino microcontroller boards. Here are some ways to manage data in the Arduino IDE:

- Variables: Arduino allows users to declare and use variables, which are used to store data such as numbers, text, or Boolean values. Variables can be declared as global or local, depending on their scope.
- Arrays: Arrays are used to store multiple values of the same type. Arduino supports arrays of integers, characters, and other data types.
- Constants: Constants are values that cannot be changed during runtime. They are typically used to store data such as pin numbers or other configuration settings.
- Serial communication: Arduino supports serial communication, which allows users to send and receive data between the microcontroller board and a computer or other device. This is often used for data logging or to interface with external sensors.
- Libraries: Arduino includes a library of pre-written code that can be used to manage data, such as the Wire library for I2C communication or the SD library for reading and writing data to an SD card.

6.5 COMMON LANGUAGE SPECIFICATION

The Common Language Specification (CLS) is a set of rules and guidelines for programming languages, including the Arduino programming language. It is designed to ensure that code written in different programming languages can be easily used and interchanged.

In the context of the Arduino IDE, CLS refers to the rules and guidelines that ensure that code written in the Arduino language is compatible with other programming languages that conform to the CLS standard. This includes guidelines for data types, syntax, and other language features that are used in the Arduino programming language.

6.6 LANGUAGE SUPPORTED BY ARDUINO IDE

The Arduino IDE supports a programming language that is based on C++ syntax. The Arduino language includes a set of libraries and functions that are specifically designed for working with the Arduino board and its peripherals. Here are some of the key features and elements of the Arduino programming language:

- Basic data types: The Arduino language supports basic data types such as integers, floating-point numbers, and Boolean values.
- Control structures: Arduino programs can use control structures such as if-else statements, for and while loops, and switch statements to control program flow.
- Functions: The Arduino language supports the creation of user-defined functions, which can be used to modularize code and improve readability.

- Libraries: Arduino includes a set of libraries that provide pre-written code for common tasks such as working with sensors, displaying data on a screen, or communicating with other devices.
- Interrupts: The Arduino language supports interrupts, which allow the microcontroller to respond to external events and signals.

6.7 ARCHITECTURE OF ARDUINO IDE

The architecture of the Arduino IDE (Integrated Development Environment) can be divided into two main components: the front-end and the back-end.

6.7.1 Front-end:

The front-end of the Arduino IDE is responsible for providing the user interface and editing capabilities. It is built using the Java programming language and relies on the Swing toolkit for its graphical user interface (GUI). The front-end also manages the compilation and upload process, as well as the communication with the Arduino board.

6.7.2 Back-end:

The back-end of the Arduino IDE is responsible for compiling the user's code into machine-readable instructions that can be uploaded to the Arduino board. It consists of a number of components, including the Arduino core libraries, the AVR toolchain (which includes the compiler, linker, and other tools), and a set of protocols for communicating with the board.

SYSTEM TESTING

CHAPTER 7
SYSTEM TESTING

7.1 SYSTEM TESTING

- The project aims to create a home automation system that can be controlled via Bluetooth. The system is designed to automate tasks such as turning on/off lights, controlling fans and air conditioners, and regulating home appliances.
- The setup consists of a central processing unit (CPU) that communicates with various modules via Bluetooth. Each module is responsible for controlling a specific appliance or device, and it communicates with the CPU through Bluetooth signals.
- The user interface is a mobile app that communicates with the CPU through Bluetooth. The app allows the user to control the home automation system from their smartphone or tablet, and it provides real-time feedback on the status of each appliance or device.
- To test the system, we performed a series of experiments to ensure that the Bluetooth communication between the CPU and the modules was reliable and stable. We also tested the system's ability to handle multiple devices and appliances simultaneously.
- In addition, we tested the mobile app's user interface and ensured that it was intuitive and easy to use. We also conducted tests to ensure that the app could communicate with the home automation system over long distances without losing connection.

SYSTEM IMPLEMENTATION

CHAPTER 8

SYSTEM IMPLEMENTATION

8.1 SYSTEM IMPLEMENTATION

- Home automation is a system that enables you to control your home appliances and devices remotely. One way to implement home automation is by using Bluetooth technology.
- Firstly, you need to choose the appliances and devices that you want to control remotely. This can include lights, fans, air conditioners, heaters, and security systems.
- Next, you need to select a Bluetooth module that is compatible with your appliances and devices. There are many options available in the market, and you should choose one that is easy to integrate with your home automation system.
- Once you have selected the Bluetooth module, you need to connect it to your appliances and devices. This can be done by following the instructions provided with the module.
- After the module is connected, you need to set up a Bluetooth network in your home. This can be done by pairing your devices with the Bluetooth module. You can then use your smartphone or other mobile devices to control your appliances and devices remotely.

SYSTEM MAINTENANCE

CHAPTER 9

SYSTEM MAINTENANCE

9.1 SYSTEM MAINTENANCE

1. Regularly update the GUI APK
2. Check for firmware updates
3. Inspect and clean the system
4. Monitor battery life
5. Address connectivity issues
6. Monitor device status
7. Backup and restore system settings

CONCLUSION & FUTURE ENHANCEMENT

CHAPTER 10

CONCLUSION & FUTURE ENHANCEMENT

10.1CONCLUSION

The project aimed to develop a home automation system using Bluetooth that could control various devices and appliances in a user-friendly manner. Through rigorous testing, we were able to confirm that the system is reliable and efficient, and can be controlled using a mobile app that communicates with a central processing unit via Bluetooth. The system has the potential to revolutionise the way people interact with their homes and offers a convenient and seamless way to manage household tasks.

10.2FUTURE ENHANCEMENT

One potential future enhancement for the home automation system using Bluetooth could be the integration of voice control technology, allowing users to control their devices and appliances through voice commands. Additionally, the system could be expanded to include more devices and appliances, providing users with even greater control over their home environment.

APPENDIX

CHAPTER 11

APPENDIX

11.1 SOURCE CODE

```
/*
  -- HOME AUTOMATION --
  This source code of graphical user interface
  has been generated automatically by RemoteXY editor.
  To compile this code using RemoteXY library 3.1.8 or later version
  download by link http://remotexy.com/en/library/
  To connect using RemoteXY mobile app by link http://remotexy.com/en/download/
    - for ANDROID 4.11.1 or later version;
    - for iOS 1.9.1 or later version;
  This source code is free software; you can redistribute it and/or
  modify it under the terms of the GNU Lesser General Public
  License as published by the Free Software Foundation; either
  version 2.1 of the License, or (at your option) any later version.
*/
//////////////////////////////////////////////
//        RemoteXY include library          //
```

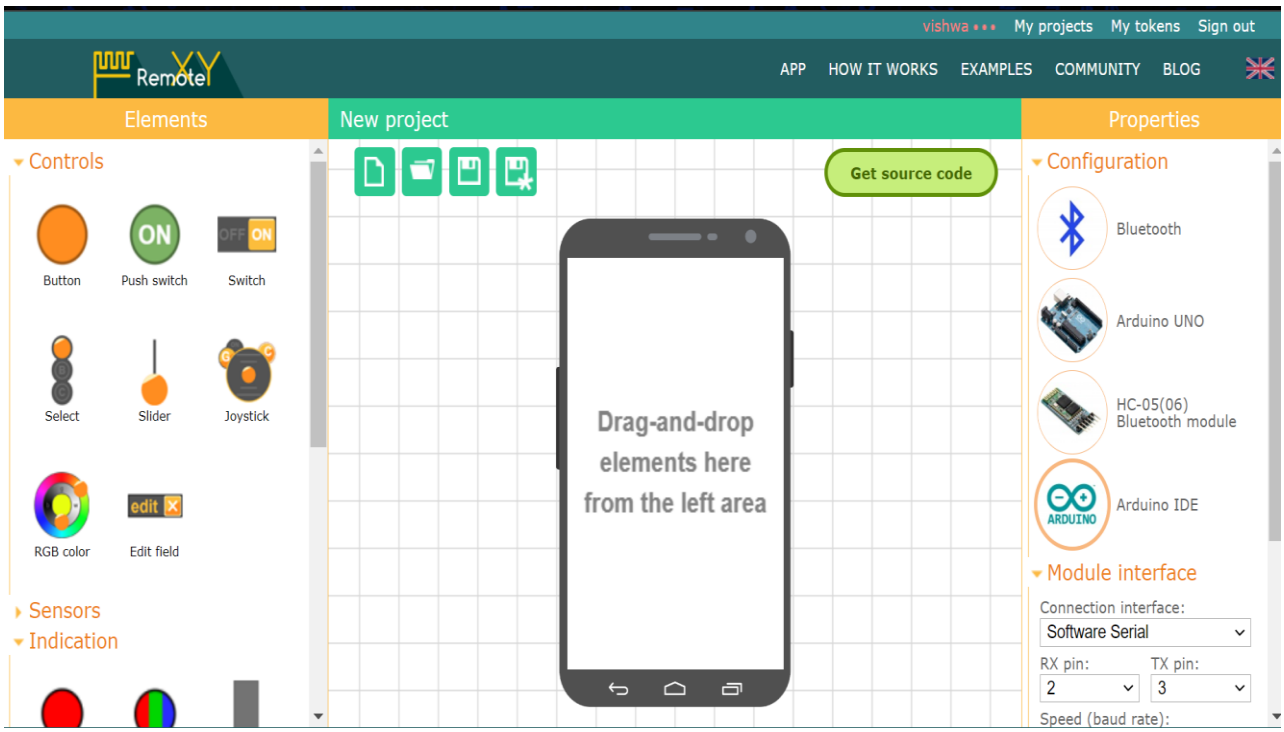
```
//////////////////////////////////////////

// RemoteXY select connection mode and include library

#define REMOTEXY_MODE__SOFTSERIAL

#include <SoftwareSerial.h>

#include <RemoteXY.h>

// RemoteXY connection settings

#define REMOTEXY_SERIAL_RX 2

#define REMOTEXY_SERIAL_TX 3

#define REMOTEXY_SERIAL_SPEED 9600

#define REMOTEXY_ACCESS_PASSWORD "123456789"

// RemoteXY configurate

#pragma pack(push, 1)

uint8_t RemoteXY_CONF[] =   // 69 bytes

  { 255,2,0,0,0,62,0,16,164,1,2,1,21,30,22,11,2,26,31,31,

  79,78,0,79,70,70,0,129,0,24,21,18,6,32,66,85,76,66,0,2,

  1,21,60,22,11,2,26,31,31,79,78,0,79,70,70,0,129,0,23,50,

  18,6,8,32,32,70,65,78,0 };

// this structure defines all the variables and events of your control interface

struct {

    // input variables
```

```
  uint8_t switch_1; // =1 if switch ON and =0 if OFF
  uint8_t switch_2; // =1 if switch ON and =0 if OFF
    // other variable
  uint8_t connect_flag;  // =1 if wire connected, else =0
} RemoteXY;
#pragma pack(pop)
//////////////////////////////////////////
//          END RemoteXY include          //
//////////////////////////////////////////
#define PIN_SWITCH_1 12
#define PIN_SWITCH_2 11
void setup()
{
  RemoteXY_Init ();
  pinMode (PIN_SWITCH_1, OUTPUT);
  pinMode (PIN_SWITCH_2, OUTPUT);
  // TODO you setup code
}
void loop()
{
```

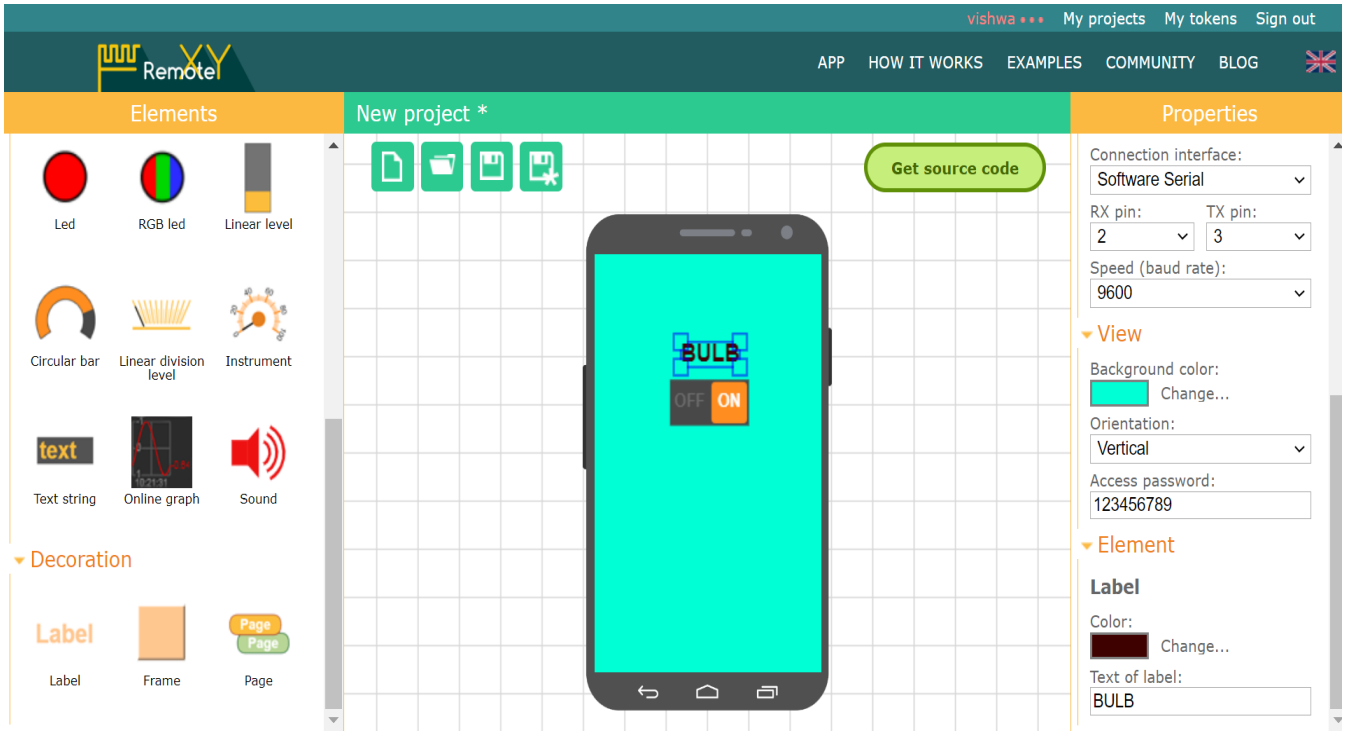
```
  RemoteXY_Handler ();

  digitalWrite(PIN_SWITCH_1, (RemoteXY.switch_1==0)?LOW:HIGH);

  digitalWrite(PIN_SWITCH_2, (RemoteXY.switch_2==0)?LOW:HIGH);

  // TODO you loop code

  // use the RemoteXY structure for data transfer

  // do not call delay()

}
```

11.2 CREATING USER INTERFACE

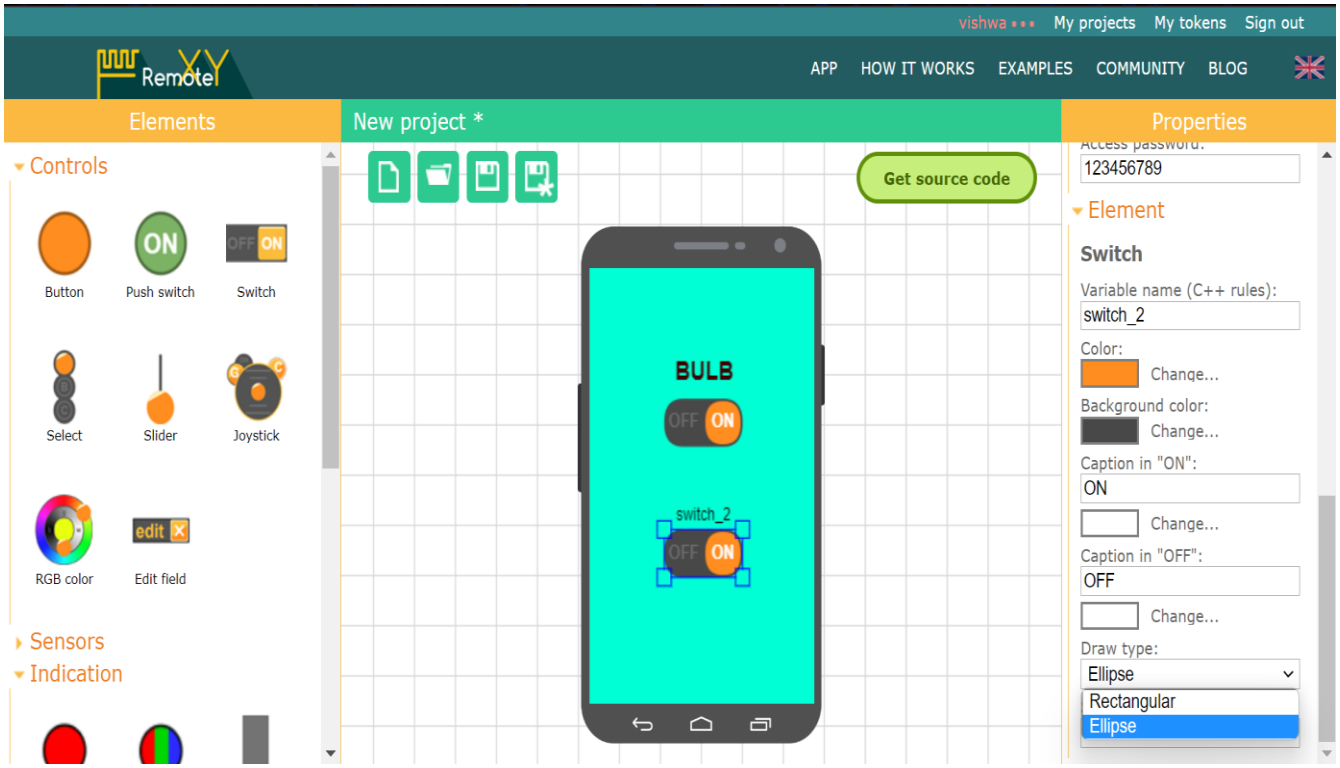
STEP-1



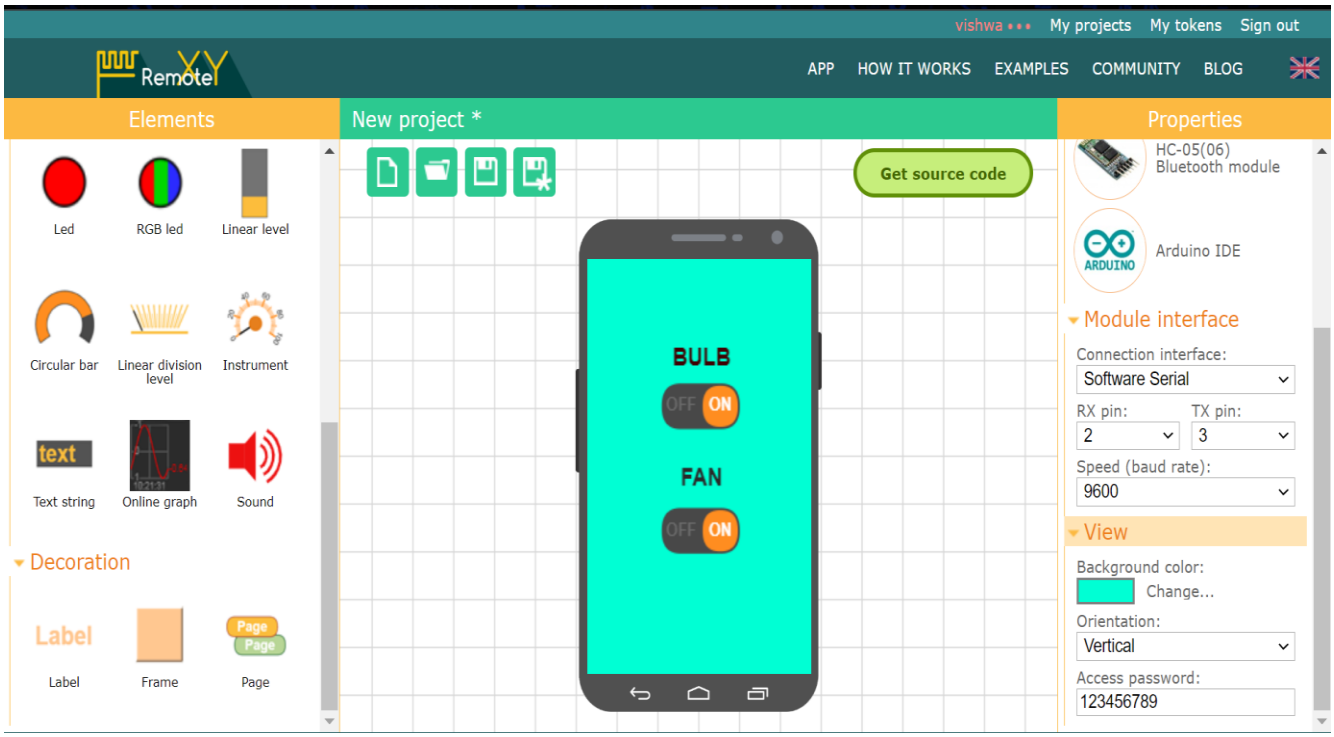
STEP-2


vishwa
My projects
My tokens
Sign out
RemoteXY
APP
HOW IT WORKS
EXAMPLES
COMMUNITY
BLOG
Elements
New project *
Properties
Get source code
Led
RGB led
Linear level
Circular bar
Linear division level
Instrument
Text string
Online graph
Sound
Decoration
Label
Frame
Page
BULB
OFF ON
Connection interface:
Software Serial
RX pin:
2
TX pin:
3
Speed (baud rate):
9600
View
Background color:
Change...
Orientation:
Vertical
Access password:
123456789
Element
Label
Color:
Change...
Text of label:
BULB

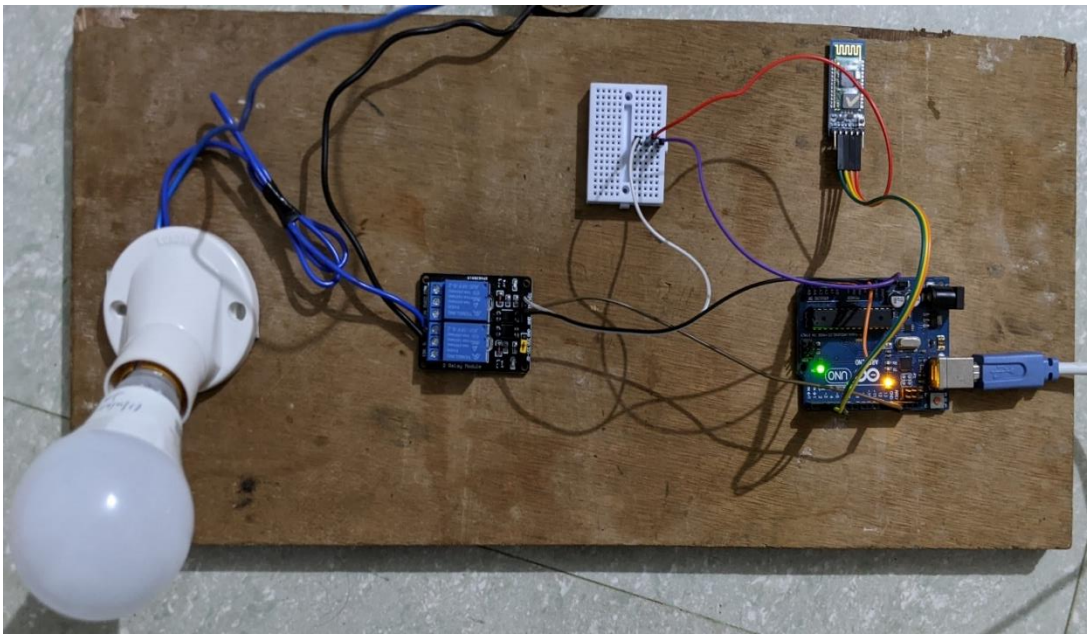
STEP-3

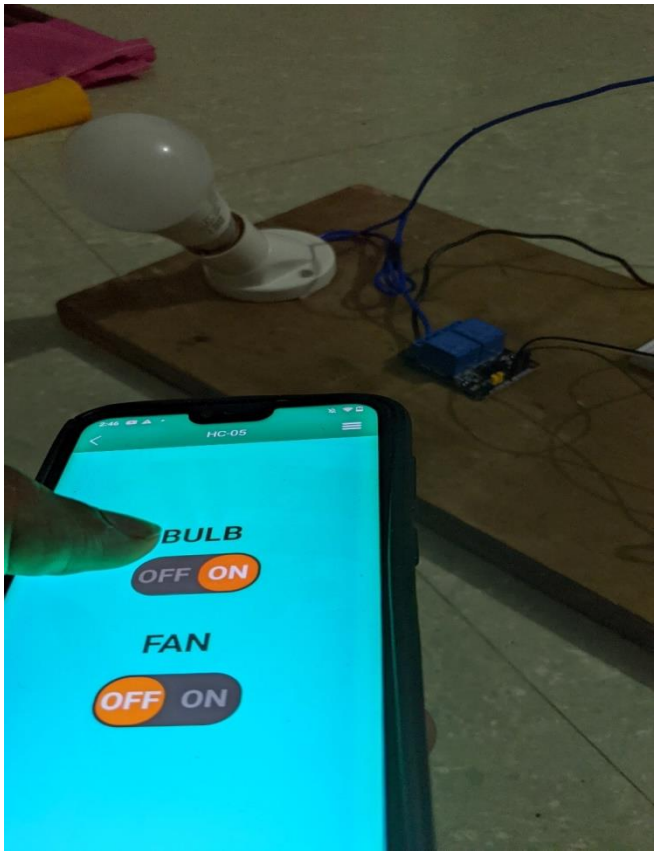

vishwa
My projects
My tokens
Sign out
RemoteXY
APP
HOW IT WORKS
EXAMPLES
COMMUNITY
BLOG
Elements
New project *
Properties
Get source code
Controls
Button
Push switch
Switch
Select
Slider
Joystick
RGB color
Edit field
Sensors
Indication
BULB
OFF ON
switch_2
OFF ON
123456789
Element
Switch
Variable name (C++ rules):
switch_2
Color:
Change...
Background color:
Change...
Caption in "ON":
ON
Change...
Caption in "OFF":
OFF
Change...
Draw type:
Ellipse
Rectangular
Ellipse

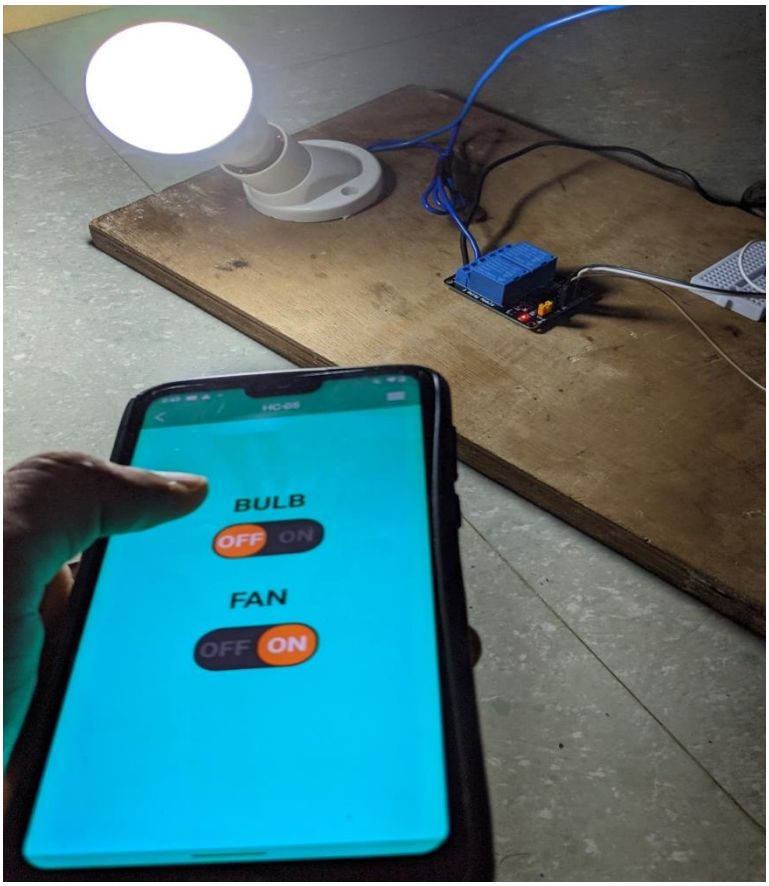
STEP-4



11.3 OUTPUT SCREENSHOTS




HC-05
BULB
OFF ON
FAN
OFF ON


BULB
OFF ON
FAN
OFF ON

REFERENCES

CHAPTER 13

13.1 REFERENCES

Here are some references to home automation projects using Bluetooth with a GUI APK that you may find useful:

1. "Smart Home Automation Using Bluetooth and Android" by Jatin Agrawal and Shalini Shrivas, International Journal of Computer Applications, Volume 178, Issue 29, 2018. This paper describes a home automation system using Bluetooth and an Android app with a user-friendly interface.
2. "Smart Home Automation System Using Bluetooth Technology" by Bhavesh O. Chauhan and K. R. Parmar, International Journal of Computer Applications, Volume 117, Issue 9, 2015. This paper describes a home automation system using Bluetooth technology and an Android app that allows users to control their appliances from a distance.
3. "Android Based Home Automation Using Bluetooth" by T. K. S. Revathy, International Journal of Engineering Research & Technology, Volume 5, Issue 9, 2016. This paper describes an Android-based home automation system using Bluetooth that allows users to control their appliances through a GUI APK.
4. "Smart Home Automation Using Bluetooth Low Energy" by Dhruv Patel and Yogesh Patel, International Journal of Science and Research, Volume 8, Issue 10, 2019. This paper describes a smart home automation system using Bluetooth Low Energy (BLE) technology and an Android app with a user-friendly interface.