Initialization of

Smart Home

Developed by…

**Soham Das D202105453**

**Sounava Ghosh D202105456**

**Debjit Mitra D202105432**

**Tanmoy Bag D202105470**

**Shuvankar Dhara D202105450**

Under the guidance & supervision of

**ARIJIT DEY**

Lecturer

Department of Computer Science and Technology The Calcutta Technical School

110 S. N. Banerjee Road, Kolkata – 700013

With immense pleasure, we, Soham Das, Shuvankar Dhara, Sounava Ghosh, Debjit Mitra, and Tanmoy Bag represent the " Initialization of Smart Home" as part of the curriculum of Diploma Engineering. We are grateful to be a part of this project.

We express our thanks to our teacher, Mr. Arijit Dey, for guiding us in completing this project and for giving us the necessary tools and support.

**Soham Das , D202105453**

**Sounava Ghosh , D202105456**

**Debjit Mitra , D202105432**

**Tanmoy Bag , D202105470**

**Shuvankar Dhara , D202105450**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Particulars** | **Page No.** |
| 1 | Introduction | 01 |
| 2 | Motivation | 02 |
| 3 | Overview of the Project | 03 |
| 4 | Detail Description of the Project | 05 – 21 |
| 5 | Conclusion and Scope of Further Study | 22 |
| 6 | References | 23 |

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Particulars** | **Page No.** |
| 1 | Arduino Overview | 04 |
| 2 | Arduino Uno R3 Front | 05 |
| 3 | Arduino Uno R3 Back | 05 |
| 4 | ESP-01 802.11 b/g/n Wi-Fi Module | 10 |
| 5 | ESP-01 Module | 12 |
| 6 | Arduino UNO with esp-01 Wi-Fi module circuit diagram | 17 |
| 7 | Arduino UNO with LED circuit diagram | 17 |
| 8 | Arduino UNO with DC motor circuit diagram | 18 |
| 9 | Connection between Arduino and Bulb via Relay | 19 |
| 10 | Overview of bulbs in working state | 20 |
| 11 | Overview of bulbs and Fans in working state | 20 |

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Particulars** | **Page No.** |
| 1 | Table 1: Pin Descriptions of Arduino Uno | 10 - 11 |
| 2 | Table 2 :Pin Descriptions of Esp-01 | 12 |
| 3 | Table 3: Price chart | 21 |

Home automation refers to the integration of various smart technologies and devices within a residential setting to enhance convenience, security, energy efficiency, and overall comfort for the inhabitants. The primary goal of home automation is to create a more intelligent and interconnected living environment where devices and systems can communicate with each other and be controlled remotely through a central hub or smartphone app.

The concept of home automation dates back several decades, but recent advancements in wireless communication, the Internet of Things (IoT), and artificial intelligence have catapulted it into the mainstream. Today, smart homes are becoming increasingly popular, as homeowners seek to simplify their daily routines, save energy, and improve their quality of life.

Home automation encompasses a wide range of applications, including controlling lighting, thermostats, security cameras, door locks, entertainment systems, and appliances. Users can interact with their smart homes through voice commands, smartphone apps, or automated schedules. This level of automation provides greater flexibility, enabling users to manage their homes remotely, even from thousands of miles away.

As the technology continues to evolve, home automation is expected to become even more sophisticated, with advanced AI systems learning from occupants' behaviors and preferences to anticipate their needs. Additionally, integrating renewable energy sources and optimizing energy consumption will contribute to building sustainable and environmentally-friendly smart homes.

Overall, home automation represents the future of modern living, promising a seamless, intelligent, and connected living experience for homeowners, making their lives more convenient, secure, and efficient.

Home automation offers numerous compelling motivations that drive its growing popularity among homeowners. The primary motivation for adopting home automation is the enhanced convenience and comfort it provides. With smart devices and systems seamlessly integrated, individuals can control and manage various aspects of their homes remotely. Whether it's adjusting the thermostat, turning off lights, or even brewing coffee before getting out of bed, automation streamlines daily tasks, saving time and effort.

Security is another significant motivation. Smart home security systems, including cameras, motion sensors, and smart locks, provide homeowners with real-time monitoring and alerts, bolstering their peace of mind and ensuring the safety of their property and loved ones.

Energy efficiency is a pressing concern in today's world, and home automation addresses this by optimizing energy consumption. Smart thermostats, lighting, and appliances can adapt to occupants' preferences and habits, reducing wastage and lowering utility bills.

Additionally, home automation enables greater accessibility for individuals with mobility challenges or disabilities. Automated systems allow them to control their environment more easily, fostering independence and improving their overall quality of life.

Entertainment is not left behind. With home automation, users can create immersive entertainment experiences, with voice-controlled home theaters and integrated audio systems offering a whole new level of enjoyment.

Finally, the increasing trend towards sustainability and eco-conscious living has driven the motivation for incorporating home automation. Smart homes can integrate renewable energy sources, manage water usage, and reduce overall waste, contributing to a greener and more environmentally responsible lifestyle.

In summary, the motivations for home automation revolve around convenience, security, energy efficiency, accessibility, entertainment, and sustainability. As technology continues to advance, these motivations will drive further innovation and adoption, making home automation an integral part of modern living.

**What is Arduino UNO?**

The Arduino UNO is categorized as a microcontroller that uses the ATmega328 as a controller in it. The Arduino UNO board is used for an electronics project and mostly preferred by the beginners. The Arduino UNO board I type of Arduino board only.

The Arduino board is the most used board of all Arduino boards.

The board contains 14 digital input/ output pins in which 6 are analog input pin, one power jack, USB connector, one reset button, ICSP header, and other components. All these components are attached in the Arduino UNO board to make it functioning and can be used in the project. The board is charged by USB port or can be directly charged by the DC supply to the board.

**What is RemoteXY?**

RemoteXY is a website that allows you to easily build a phone interface that will talk to your hardware over Bluetooth LE, USB, or Ethernet (including Wi-Fi). One thing of interest: even though the interface builder is Web-based, the service claims that the interface structure stays on the controller.

There’s no interaction with the remote servers when operating the user interface so there is no need for an external Internet connection.

**ESP-01 802.11 b/g/n Wi-Fi Module**

The ESP8266 ESP-01 was one of the first low-cost microcontrollers with Wi-Fi connectivity. It has since been replaced by newer versions with more GPIOs, more memory, and more power.

**Operating Led using Arduino UNO-**

In this part of the project, we applied the concept of delay feature in the sketch of Arduino to turn the LED on and off while having a custom delay in between.

**Operating DC motor using Arduino-**

Turning MOTOR on and off through delay control in Arduino Uno- In this part of the project we applied the concept of delay feature in the sketch of Arduino to turn the MOTOR on and off while having a custom delay in between. The main idea behind this is to apply the concept of frequency to the sketch of Arduino to turn the MOTOR on and off. The controlling part of the sketch is done in the setup loop. This loop is executed once when the Arduino board resets and then continuously executes the commands written inside it. In this setup loop we have defined the pin number of the Arduino board and set it as output using the pinMode function. Then we have written the command to make the MOTOR run by setting the pin number as HIGH and written the command to make the MOTOR stop by setting the pin number as LOW. The time delay between these two events is given by the delay function. The time delay is given in milliseconds.

**Controlling Motor & LED Using Arduino with the help of Wi-Fi module-**

The app will be configured to control the LED light and motor by sending commands to the Arduino UNO board. The board will be programmed to receive and process these commands and operate the respective appliances according to the commands received.

This project will focus on controlling the LED light and motor with the help of the RemoteXY app. The Arduino UNO board will be connected to a router through the Ethernet connection. The app will be configured to control the LED and motor by sending commands to the Arduino UNO board. The board will be programmed to receive and process these commands and operate the respective appliances according to the commands received.

**Arduino**

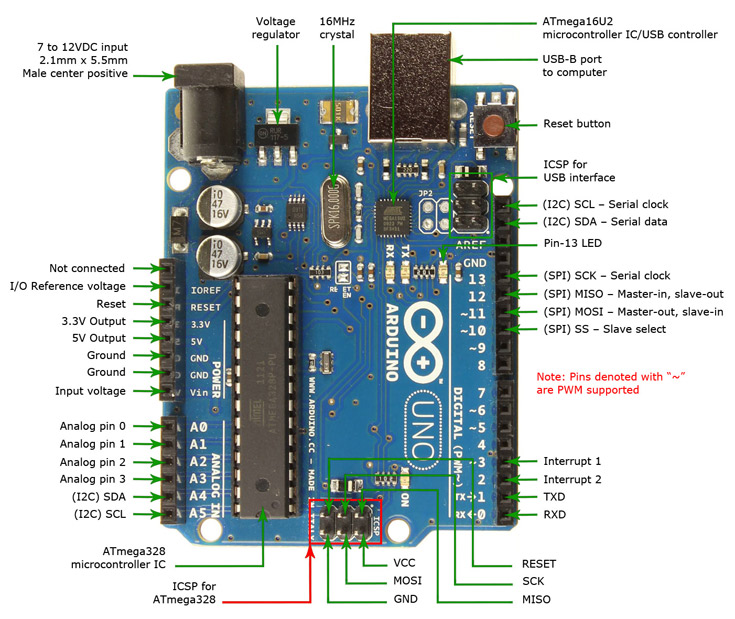
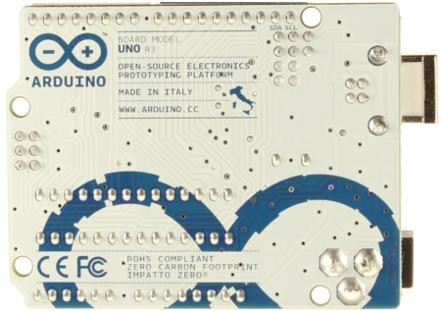


Figure 1: Arduino Overview

**Why Arduino**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics



|  |  |  |
| --- | --- | --- |
| Figure 2 :Arduino Uno R3 Front |  | Figure 3 :Arduino Uno R3 Back |

**Overview**

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode](http://arduino.cc/en/Hacking/DFUProgramming8U2).

Revision 3 of the board has the following new features:

* 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
* Stronger RESET circuit.
* Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](http://arduino.cc/en/Main/Boards).

**Summary**

|  |  |
| --- | --- |
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |

**Details**

* PWM: 3, 5, 6, 9, 10, and 11**.** Provide 8-bit PWM output with the [analogWrite()](http://arduino.cc/en/Reference/AnalogWrite) function.
* SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK**).** These pins support SPI communication using the [SPI library](http://arduino.cc/en/Reference/SPI).
* LED: 13**.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e.,1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the [analogReference](http://arduino.cc/en/Reference/AnalogReference)() function. Additionally, some pins have specialized functionality:

* TWI: A4 or SDA pin and A5 or SCL pin.Support TWI communication using the [Wire library](http://arduino.cc/en/Reference/Wire).

There are a couple of other pins on the board:

* AREF**.** Reference voltage for the analog inputs. Used with [analogReference](http://arduino.cc/en/Reference/AnalogReference)().
* Reset**.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

**Communication**

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed.

However, [on Windows,.inf file is required](http://arduino.cc/en/Guide/Windows#toc4). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [Software Serial library](http://www.arduino.cc/en/Reference/SoftwareSerial) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication.

The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](http://arduino.cc/en/Reference/Wire) for details. For SPI communication, use the [SPI library](http://arduino.cc/en/Reference/SPI).

**Programming**

The Arduino Uno can be programmed with the Arduino software ([download](http://arduino.cc/en/Main/Software)). Select "Arduino Uno from the Tools > Boardmenu (according to the microcontroller on your board). For details, see the [reference](http://arduino.cc/en/Reference/HomePage) and [tutorials](http://arduino.cc/en/Tutorial/HomePage).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](http://arduino.cc/en/Tutorial/Bootloader) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](http://www.atmel.com/dyn/resources/prod_documents/doc2525.pdf), [C header files](http://www.atmel.com/dyn/resources/prod_documents/avr061.zip)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](http://arduino.cc/en/Hacking/Programmer) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

* On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
* On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3886) (Windows) or the [DFU programmer](http://dfu-programmer.sourceforge.net/) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1285962838) for more information.

**Automatic (Software) Reset**

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano-farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half- second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e., anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110-ohm resistor from 5V to the reset line; see this forum thread for details.

**USB Overcurrent Protection**

The Arduino Uno has a resettable polypus that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

**Physical Characteristics**

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100-mil spacing of the other pins.

**Table 1 Pin Descriptions-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** | **Alternative Functions** |
| 1 | RX / D0 | Digital IO Pin 0 / Serial RX Pin | Generally used as RX |
| 2 | TX / D1 | Digital IO Pin 1 / Serial TX Pin | Generally used as TX |
| 3 | D2 | Digital IO Pin 2 |  |
| 4 | D3 | Digital IO Pin 3 | Timer (OC2B) |
| 5 | D4 | Digital IO Pin 4 | Timer (T0/XCK) |
| 6 | D5 | Digital IO Pin 5 | Timer (OC0B/T1) |
| 7 | D6 | Digital IO Pin 6 |  |
| 8 | D7 | Digital IO Pin 7 |  |
| 9 | D8 | Digital IO Pin 8 | Timer (CLK0/ICP1) |
| 10 | D9 | Digital IO Pin 9 | Timer (OC1A) |
| 11 | D10 | Digital IO Pin 10 | Timer (OC1B) |
| 12 | D11 | Digital IO Pin 11 | SPI (MOSI) Timer (OC2A) |
| 13 | D12 | Digital IO Pin 12 | SPI (MISO) |

|  |  |  |  |
| --- | --- | --- | --- |
| 14 | D13 | Digital IO Pin 13 | SPI (SCK) |
| 15 | GND | Ground |  |
| 16 | AREF | Analog Reference |  |
| 17 | SDA / D18 | Digital IO Pin 18 | I2C Data Pin |
| 18 | SCL / D19 | Digital IO Pin 19 | I2C Clock Pin |
| 19 | NC | Not Connected |  |
| 20 | IOREF | Voltage Reference |  |
| 21 | RESET | Reset (Active LOW) |  |
| 22 | 3V3 | Power |  |
| 23 | 5V | +5V Input |  |
| 24 | GND | Ground |  |
| 25 | GND | Ground |  |
| 26 | VIN | Unregulated Supply |  |
| 27 | A0 | Analog Input 0 | Digital IO Pin 14 |
| 28 | A1 | Analog Input 1 | Digital IO Pin 15 |
| 29 | A2 | Analog Input 2 | Digital IO Pin 16 |
| 30 | A3 | Analog Input 3 | Digital IO Pin 17 |
| 31 | A4 | Analog Input 4 | Digital IO Pin 18 I2C (SDA) |
| 32 | A5 | Analog Input 5 | Digital IO Pin 19 I2C (SCL) |

**Arduino IDE**

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. **Arduino Sketch**

The Arduino programming language is based on Processing, which is aimed at visual artists. Hence a development version being a 'sketch'. "Processing is a programming language, development environment, and online community that since 2001 has promoted software literacy within the visual arts.

**ESP-01 802.11 b/g/n Wi-Fi Module**

**Overview**

The ESP8266 ESP-01 is a Wi-Fi module that allows microcontrollers access to a Wi-Fi network. This module is a self-contained SOC (System on a Chip) that doesn’t necessarily need a microcontroller to manipulate inputs and outputs as you would normally do with an Arduino, for example, because the ESP-01 acts as a small computer. Depending on the version of the ESP8266, it is possible to have up to 9 GPIOs (General Purpose Input Output). Thus, we can give a microcontroller internet access like the Wi-Fi shield does to the Arduino, or we can simply program the ESP8266 to not only have access to a Wi-Fi network, but to act as a microcontroller as well. This makes the ESP8266 very versatile, and it can save you some money and space in your projects.

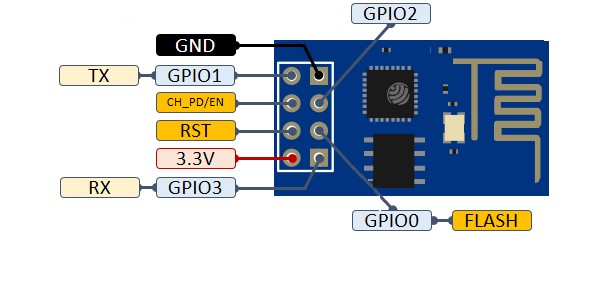
**Pin Descriptions**

Figure 4 : ESP-01 802.11 b/g/n Wi-Fi Module

|  |  |  |
| --- | --- | --- |
| **No.** | **Pin Name** | **Function** |
| 1 | GND | GND |
| 2 | GPIO2 | GPIO, Internal Pull-Up |
| 3 | GPIO0 | GPIO, Internal Pull-Up |
| 4 | GPIO3 | General Purpose IO, Serial Rx |
| 5 | VCC | 3.3V power supply (VDD) |
| 6 | RST | Active Low External Reset signal |
| 7 | CH\_PD/EN | High Active Chip Enable |
| 8 | GPIO1 | General Purpose IO Serial Tx |

**Table 2 Pin Descriptions-**

**Packaging and Dimension**

The external size of the module is 14.3mm\*24.8mm\*3mm, as is illustrated in Figure 3 below. The type of flash integrated in this module is an SPI flash, the capacity of which is 1 MB, and the package size of which is SOP-210miI. The antenna applied on this module is a 3DBi PCB-on-board antenna.

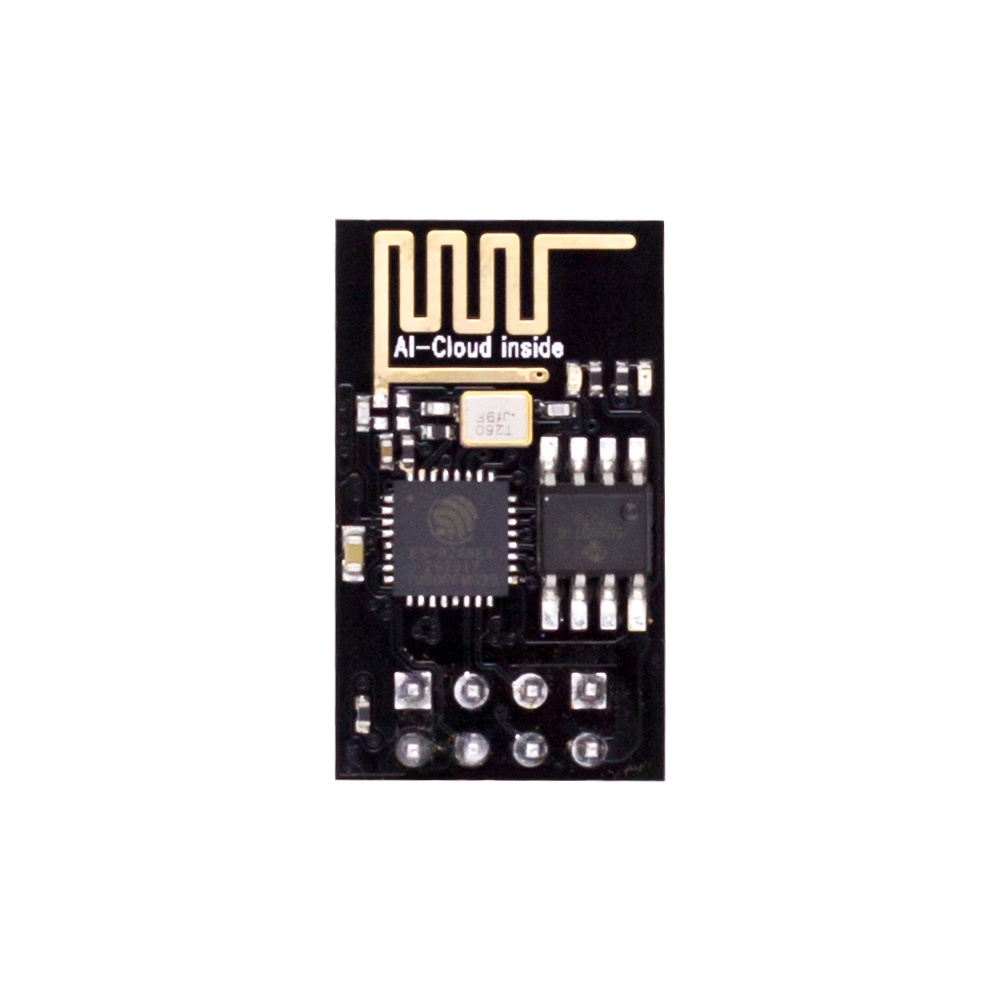


Figure 5 : ESP-01 Module

**Controlling LED and motor using Arduino**

**with the help of WIFI module**

**Introduction: -**

In our final project on the Arduino UNO board, we brought the concept of IOT devices.

We used the help of RemoteXY platform to implement this with the help of ESP-01 in relation with the Arduino UNO to establish connection between the user, IOT platform (RemoteXY) and the hardware we will be controlling with it.

The hardware that will be controlled with it are LED light and a DC motor.

We will be using the smart phone app developed by the RemoteXY platform to initiate control of these appliances using Ethernet connected to a router as the medium.

**Tools & Components: -**

1. DC motor
2. Resistance(1/4 watt , 1k Ω)
3. Wire(500 mA)
4. Arduino Uno R3
5. Diode
6. NPN Transistor
7. 5V DC power Supply
8. LED
9. ESP-01
10. Breadboard
11. Router

**Source code: -**

/\*

-- Home automation final --

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 select connection mode and include library

#define REMOTEXY\_MODE\_\_ESP8266\_HARDSERIAL

#include <RemoteXY.h>

// RemoteXY connection settings

#define REMOTEXY\_SERIAL Serial

#define REMOTEXY\_SERIAL\_SPEED 115200

#define REMOTEXY\_WIFI\_SSID "Debjitroxx"

#define REMOTEXY\_WIFI\_PASSWORD "12345678"

#define REMOTEXY\_SERVER\_PORT 6377

#define REMOTEXY\_ACCESS\_PASSWORD "arduino"

// RemoteXY configurate

#pragma pack(push, 1)

uint8\_t RemoteXY\_CONF[] = // 298 bytes

{ 255,9,0,0,0,35,1,16,24,1,2,0,42,11,13,5,2,26,31,31,

79,78,0,79,70,70,0,4,192,25,77,36,5,120,0,4,192,25,85,36,

5,120,0,4,192,25,93,36,5,120,0,4,192,25,69,36,5,120,0,2,

0,42,48,13,5,2,26,31,31,79,78,0,79,70,70,0,2,0,42,30,

13,5,2,26,31,31,79,78,0,79,70,70,0,2,0,42,39,13,5,2,

26,31,31,79,78,0,79,70,70,0,2,0,42,20,13,5,2,26,31,31,

79,78,0,79,70,70,0,129,0,4,49,14,3,2,66,97,116,104,114,111,

111,109,0,129,0,4,12,13,3,2,66,101,100,114,111,111,109,0,129,0,

4,30,9,3,2,68,105,110,105,110,103,0,129,0,4,40,11,3,2,75,

105,116,99,104,101,110,0,129,0,4,21,16,3,2,66,101,100,114,111,111,

109,32,50,0,129,0,26,3,12,4,2,76,105,103,104,116,115,0,129,0,

27,60,9,4,121,70,97,110,115,0,130,1,244,56,78,1,26,129,0,4,

70,13,3,121,66,101,100,114,111,111,109,0,129,0,4,78,16,3,121,66,

101,100,114,111,111,109,32,50,0,129,0,4,86,9,3,121,68,105,110,105,

110,103,0,129,0,4,94,11,3,121,75,105,116,99,104,101,110,0 };

// 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

int8\_t slider\_2; // =0..100 slider position

int8\_t slider\_3; // =0..100 slider position

int8\_t slider\_4; // =0..100 slider position

int8\_t slider\_1; // =0..100 slider position

uint8\_t switch\_5; // =1 if switch ON and =0 if OFF

uint8\_t switch\_3; // =1 if switch ON and =0 if OFF

uint8\_t switch\_4; // =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)

#define PIN\_SWITCH\_1 2

#define PIN\_SWITCH\_5 12

#define PIN\_SWITCH\_3 7

#define PIN\_SWITCH\_4 8

#define PIN\_SWITCH\_2 4

void setup()

{

RemoteXY\_Init ();

pinMode (PIN\_SWITCH\_1, OUTPUT);

pinMode (PIN\_SWITCH\_5, OUTPUT);

pinMode (PIN\_SWITCH\_3, OUTPUT);

pinMode (PIN\_SWITCH\_4, OUTPUT);

pinMode (PIN\_SWITCH\_2, OUTPUT);

// TODO you setup code

}

void loop()

{

RemoteXY\_Handler ();

digitalWrite(PIN\_SWITCH\_1, (RemoteXY.switch\_1==0)?HIGH:LOW);

digitalWrite(PIN\_SWITCH\_5, (RemoteXY.switch\_5==0)?LOW:HIGH);

digitalWrite(PIN\_SWITCH\_3, (RemoteXY.switch\_3==0)?HIGH:LOW);

digitalWrite(PIN\_SWITCH\_4, (RemoteXY.switch\_4==0)?HIGH:LOW);

digitalWrite(PIN\_SWITCH\_2, (RemoteXY.switch\_2==0)?HIGH:LOW);

int MotorSpeed1 = RemoteXY.slider\_1 \* 2.4 ;

analogWrite(3,MotorSpeed1);

int MotorSpeed2 = RemoteXY.slider\_2 \* 2.4 ;

analogWrite(5,MotorSpeed2);

int MotorSpeed3 = RemoteXY.slider\_3 \* 2.4 ;

analogWrite(6,MotorSpeed3);

int MotorSpeed4 = RemoteXY.slider\_4 \* 2.4 ;

analogWrite(9,MotorSpeed4);

}

**Connection: -**

1. **ESP to Arduino-**
2. RX Pin of Arduino Uno connected to TX port of ESP-01
3. TX in Arduino Uno connected to RX port of ESP-01 through 1k resister
4. 3.3v port in Arduino Uno connected to 3.3v pin in ESP-01 to power it
5. GND port in Arduino Uno connected to GND port in ESP-01
6. RX pin of ESP-01 is connected to GND with a 2k resister

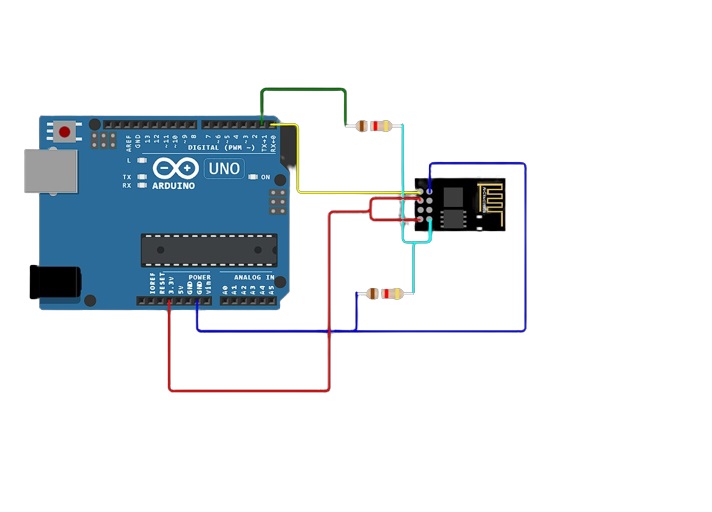


Figure 6:Arduino UNO with esp-01 Wi-Fi module circuit diagram

1. **Bathroom Led to Arduino-**
2. LED Anode is connected to one end of Resistor
3. Another end of resistor is connected to Pin 12
4. LED Cathode is connected to Ground

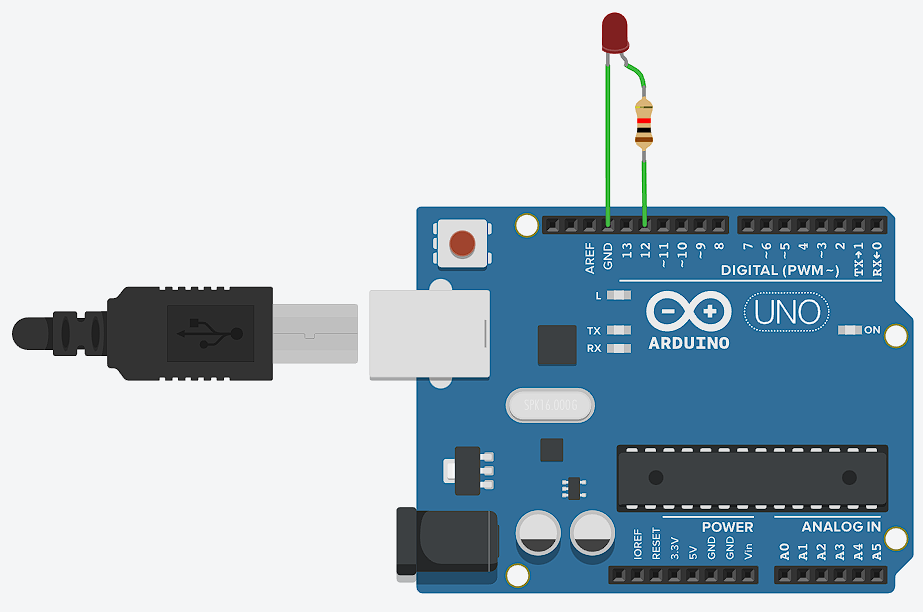
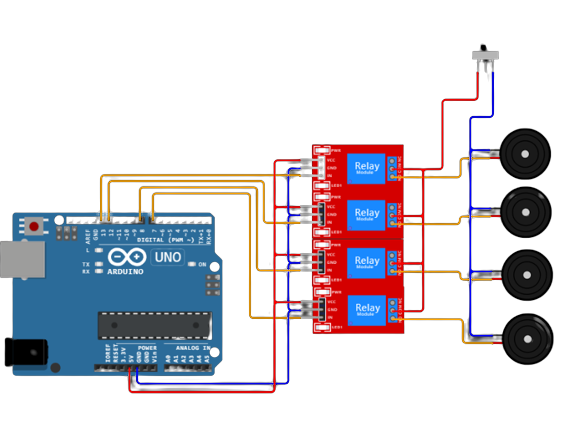


Figure 7:Arduino UNO with LED circuit diagram

****

1. **Motor to Arduino-**
2. Connect one end of the resistor to Pin 9 as an input.
3. Connect the other end of the resistor to the middle Pin (Base) of the transistor
4. Connect Collector of the transistor to the Anode of the diode
5. Connect Cathode of diode to the GND Pin of the Arduino
6. Connect the Cathode of diode to Positive 5V of the supply
7. Connect the positive 5V of supply to the dc motor positive pin
8. Connect the negative 5V of supply to the Emitter of the Transistor
9. Connect the negative dc motor pin to the anode of the diode
10. Repeat the same steps 1-8 using pins 6,5 & 3 as an input .

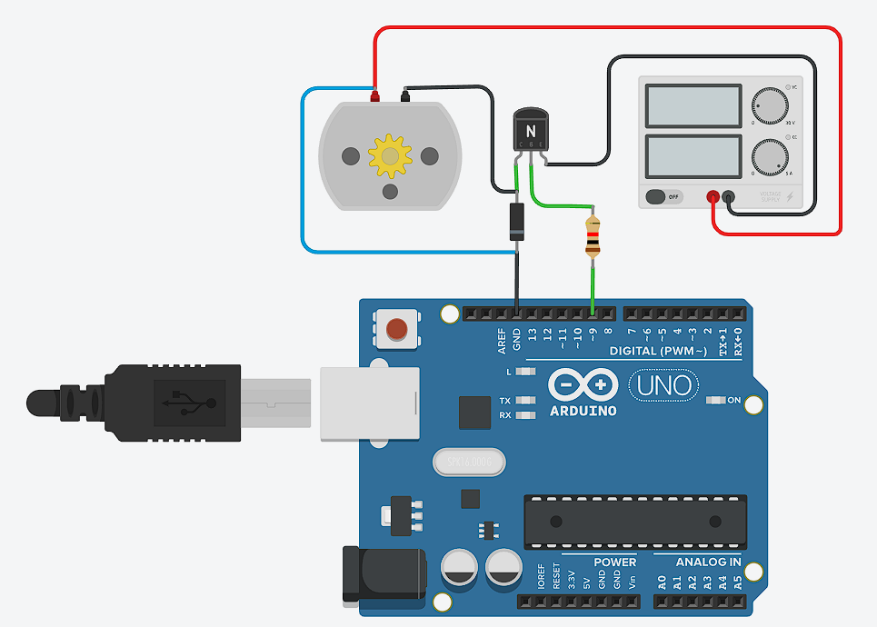
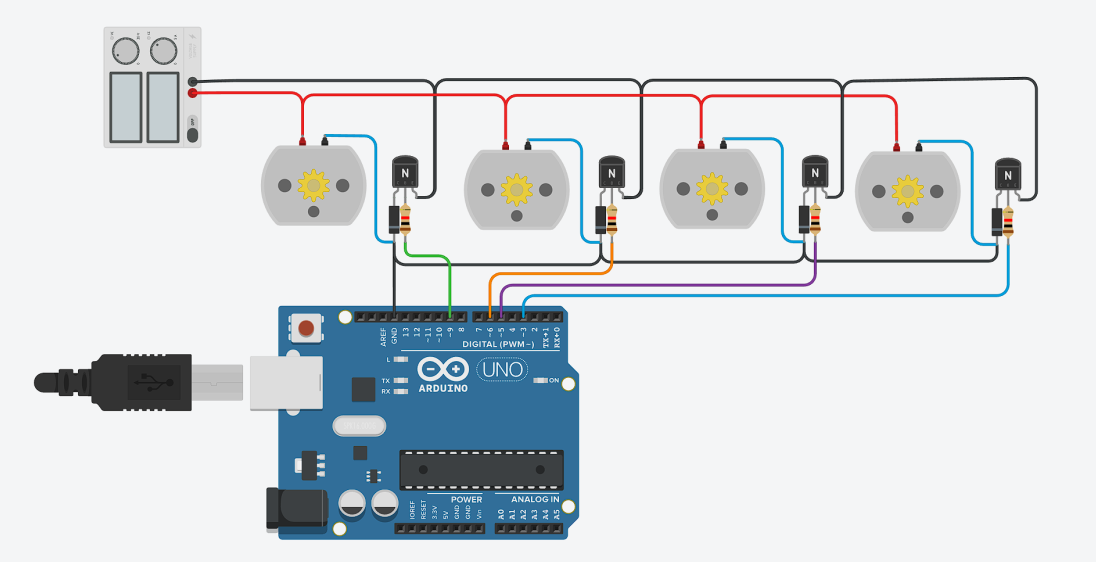


Figure 8:Arduino UNO with DC motor circuit diagram

1. Connect Pin 8 to relay as an input.
2. Connect VCC of relay to the 5V supply of Arduino.
3. Connect GND of relay to GND of Arduino.
4. Connect the output of the relay to a bulb .
5. Repeat the same steps 1-5 using pins 2, 4 & 7 as an input .

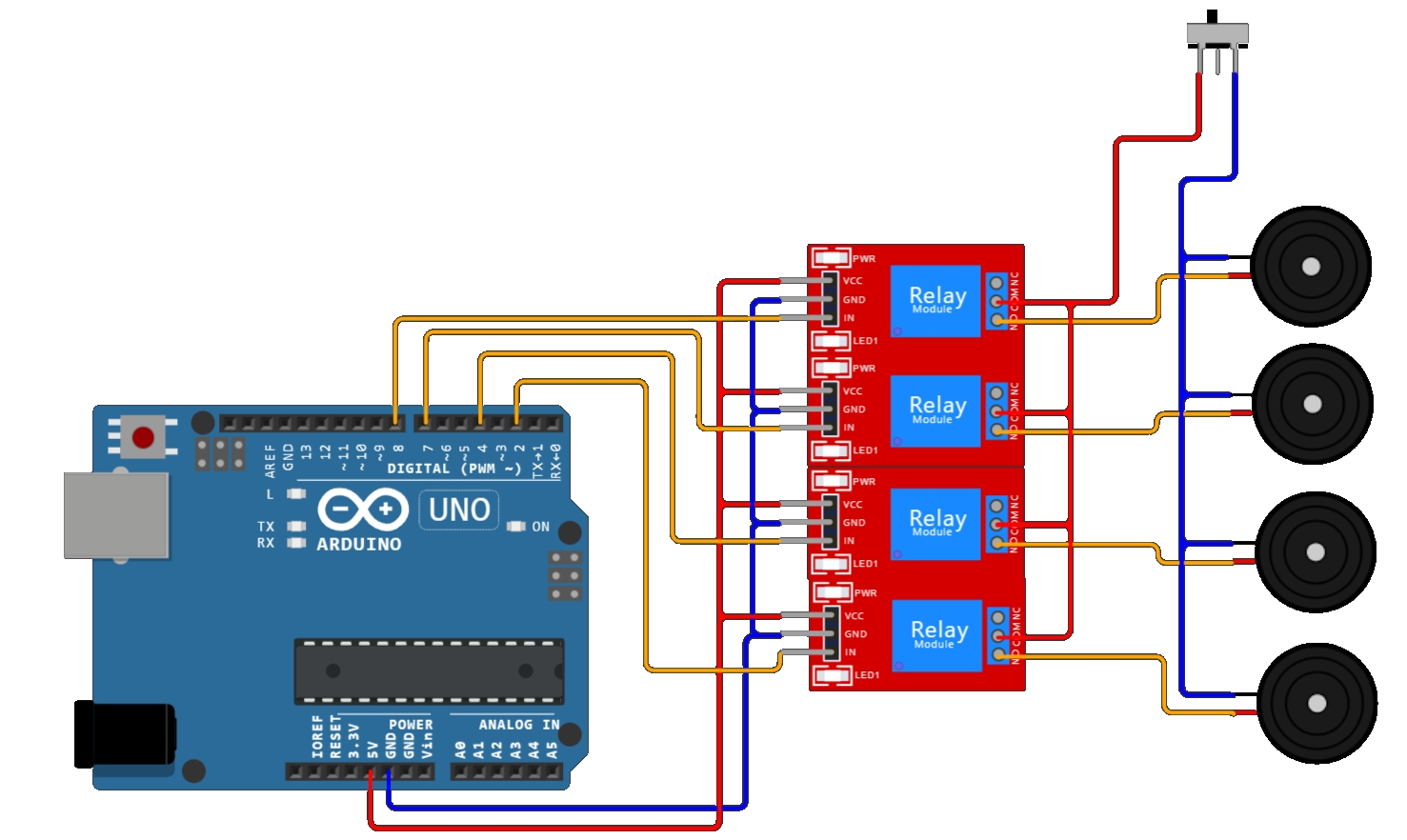


Figure 9: Connection between Arduino and Bulb via Relay



Figure 10 : Overview of bulbs in working state



Figure 11: Overview of bulbs and Fans in working state

|  |  |  |  |
| --- | --- | --- | --- |
| **SL No.** | **Item** | **Piece** | **Price( ₹ )** |
| 1 | 2N2222A | 5 | 15 |
| 2 | 1kΩ (1/4 w) | 10 | 10 |
| 3 | 1N4007 | 10 | 10 |
| 4 | Motor(5v) | 4 | 60 |
| 5 | 4 relay module | 1 | 150 |
| 6 | Vero board(doted) | 1 | 40 |
| 7 | Male connecter | 2 | 12 |
| 8 | Male to female | 60 | 90 |
| 9 | Flux | 1 | 25 |
| 10 | Solder | 1 | 80 |
| 11 | Red wire(2.5 mm, multi core) | 4m | 40 |
| 12 | Blue wire (1mm , multi core) | 5m | 50 |
| 13 | Lamp (0.5 w) | 4 | 160 |
| 14 | Glue stick | 4 | 40 |
| 15 | ESP01 | 1 | 130 |
| 16 | Router | 1 | 1000 |
| 17 | Arduino Uno R3 | 1 | 750 |
| 18 | MFD board | 1 | 320 |
| 19 | Holder | 4 | 80 |
| 20 | Cutter | 1 | 40 |
| 21 | Cutter Blade | 1 | 10 |
| 22 | Single core wire | 2m | 20 |
| 23 | Bread broad | 1 | 100 |
| 24 | Glue Gun | 1 | 240 |
| 25 | Soldering iron | 1 | 220 |
| 26 | Multi meter | 1 | 320 |
| 27 | 5V power supply | 1 | 169 |
| 28 | .5 mm wire | 10m | 100 |
| 29 | Female to Female | 20 | 30 |
| 30 | **Total Amount** | | 4311 |

Table 3: Price chart

**Conclusion: -**

Main purpose of automation system is to provide ease to people to control different components with the help of the android application present in their mobile phones and to save electricity, time and money.

The automation system has been experimentally proven to work satisfactorily by connecting simple components to it and the components were successfully controlled from a wireless mobile device.

**Future Studies on this project: -**

Future studies on home automation will continue to focus on advancing technology to enhance the efficiency, convenience, and sustainability of smart homes. One key area of research will be in the development of highly intuitive and context-aware artificial intelligence (AI) systems that can better understand and anticipate occupants' needs. These AI systems will adapt to individual preferences, learn from behavior patterns, and proactively optimize energy consumption, security, and comfort.

Moreover, researchers will explore the integration of emerging technologies like augmented reality (AR) and virtual reality (VR) to offer more immersive and interactive home control interfaces. This could revolutionize how users interact with their smart homes, making it more user-friendly and accessible.

Another crucial aspect of future studies will be focused on addressing security and privacy concerns related to the growing network of connected devices. Researchers will develop robust encryption protocols, authentication mechanisms, and secure communication channels to safeguard against cyber-attacks and unauthorized access to smart home systems.

Furthermore, home automation research will emphasize interoperability and standardization to ensure seamless communication among diverse smart devices and platforms. This will encourage the growth of a unified smart home ecosystem, allowing users to integrate various products and services more efficiently.

Finally, sustainability will be a paramount concern, leading to studies that explore energy-efficient technologies, renewable energy integration, and waste reduction strategies within smart homes. The aim is to create environmentally friendly homes that contribute to a greener future. Overall, these future studies will pave the way for increasingly intelligent, interconnected, and sustainable smart homes, transforming the way we live and interact with technology.

# 6. References

1. <https://en.wikipedia.org/wiki/ESP8266>
2. Book: Getting Started With Arduino - The Open Source Electronics Prototyping Platform

Publisher ‏-Shroff/Maker Media; Third edition

1. <https://www.amazon.in/Massimo-Banzi/e/B00355CV22/ref=aufs_dp_mata_dsk>
2. https://docs.arduino.cc/hardware/uno-rev3