HOME AUTOMATION USING ARDUINO UNO

A

PROJECT REPORT

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATIONS ENGINEERING

SUBMITTED BY:

KASHISH GOYAL, KRISHNA BHAT, MAINAK SEN

ROLL NO. UE215047, 50 53

SIXTH SEMESTER

UNDER THE SUPERVISION OF:

Dr. NARESH KUMAR

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY,

PANJAB UNIVERSITY, CHANDIGARH, INDIA

May 2024

UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY,

PANJAB UNIVERSITY, CHANDIGARH

CANDIDATE'S DECLARATION

I hereby certify that the work presented in the Project entitled "**Home Automation using Arduino Uno**" in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in Electronics and Communications Engineering from University Institute of Engineering and Technology, Panjab University, Chandigarh, is an authentic record of my own work carried out under the supervision and guidance of Dr Naresh Kumar, Professor.

Date: 06/06/2024

Kashish Goyal, Krishna Bhat, Mainak Sen
UE215047, 50, 53

Place: Chandigarh

(Sixth Semester)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 28/05/2024
Place: Chandigarh

Dr. Naresh Kumar
Professor
ECE, UIET, Panjab University, Chandigarh

ABSTRACT

This project presents the development and implementation of a home automation system using Arduino Uno microcontroller boards, offering an affordable and accessible solution for automating household tasks and enhancing living comfort. The system can integrate various sensors, actuators, and communication modules to enable remote monitoring and control of home appliances, lighting, and environmental conditions.

The core of the system is built around Arduino Uno boards, which serve as the central processing units orchestrating the interactions between different components. Sensors such as motion detectors, temperature and humidity sensors, and light sensors can provide real-time data on environmental parameters if used, while actuators such as relays and servo motors enable control over appliances and devices.

The project employs a modular approach, allowing users to customize and expand the system according to their specific requirements and preferences. A user-friendly interface, either through a dedicated mobile application or a web-based dashboard, enables remote access and control of the home automation system from anywhere with internet connectivity.

Through this project, individuals with basic programming and electronics knowledge can gain hands-on experience in designing and implementing a practical home automation system using Arduino Uno microcontrollers. The open-source nature of the Arduino platform fosters collaboration and innovation, making it an ideal platform for DIY enthusiasts, hobbyists, and students interested in exploring the possibilities of home automation technology.

In this Arduino project, we made an Arduino IoT project using ESP8266 wi-fi module, Arduino UNO, Relay device, sound sensors, manual switches, and mobile control. We have used all the free tools for this IoT-based home automation system.

Table of Contents

---

1. Introduction

1.1 Background

Home automation, often referred to as domotics or smart home technology, has its roots in the concept of automating household tasks and functions to simplify and improve daily life. The idea of controlling home appliances and systems remotely or automatically has been explored for decades, but recent advancements in technology have accelerated its adoption and made it more accessible to consumers.

The history of home automation can be traced back to the early 20th century, with the invention of labour-saving devices such as washing machines, refrigerators, and vacuum cleaners. These innovations laid the groundwork for the automation of household chores and set the stage for further developments in the field.

In the 1970s and 1980s, the introduction of microprocessors and digital technology paved the way for more sophisticated home automation systems. Early adopters began experimenting with simple automation tasks using programmable controllers and X10 protocol, which allowed devices to communicate over existing electrical wiring.

The 1990s saw the emergence of proprietary home automation solutions, offering integrated systems for lighting control, climate control, and security. However, these systems were often expensive, complex to install, and lacked interoperability between different devices and manufacturers.

The early 2000s marked a significant shift in the home automation landscape with the proliferation of wireless communication protocols such as Zigbee, Z-Wave, and Wi-Fi. These standards enabled seamless connectivity between devices and laid the foundation for the Internet of Things (IoT) revolution.

1.2Motivation

Today, home automation has become more mainstream, driven by the widespread adoption of smartphones, voice assistants, and smart devices. Consumers can now easily control and monitor their homes remotely using mobile apps or voice commands, while advancements in AI and machine learning enable intelligent automation and predictive analytics.

- Convenience and Comfort: Home automation projects offer the opportunity to simplify daily tasks and routines, providing greater convenience and comfort for homeowners. Automated lighting, temperature control, and appliance management systems can streamline household chores and improve overall living standards.

- Energy Efficiency: Building a home automation system allows individuals to optimize energy usage and reduce utility costs. Smart thermostats, occupancy-based lighting controls, and energy monitoring systems enable users to identify and eliminate wasteful practices, contributing to a more sustainable and eco-friendlier lifestyle.

• Personalization and Customization: Home automation projects can be tailored to suit the specific needs and preferences of homeowners. Whether it's creating personalized lighting scenes, setting custom schedules for appliances, or implementing unique automation routines, DIY enthusiasts have the freedom to design a system that aligns with their lifestyle and preferences.

• Learning and Skill Development: Building a home automation project provides a hands-on learning experience in electronics, programming, and system integration. It offers an opportunity to develop practical skills in hardware design, software development, and troubleshooting, empowering individuals to tackle more complex projects in the future.

• Security and Peace of Mind: Home automation systems can enhance security measures and provide peace of mind for homeowners. Integrated security cameras, motion sensors, and smart door locks offer real-time monitoring and alerts, deterring intruders and ensuring the safety of occupants and property.

• Innovation and Creativity: Home automation projects encourage creativity and innovation, allowing individuals to explore new ideas and experiment with emerging technologies. Whether it's incorporating voice control, implementing AI algorithms, or integrating IoT devices, there's ample opportunity to push the boundaries of what's possible in smart home technology.

• Cost Savings: While commercial smart home products can be expensive, DIY home automation projects offer a cost-effective alternative. By leveraging affordable components such as Arduino microcontrollers, sensors, and actuators, individuals can build custom solutions that meet their needs without breaking the bank.

1.3 Objectives

The primary objective of this project is to develop a home automation using Arduino Uno that has the benefits of manifold, ranging from increased energy efficiency and cost savings to enhanced security and convenience. Smart thermostats if used can optimize heating and cooling schedules based on occupancy patterns, while smart lighting systems can adjust brightness and color temperature to suit different activities and moods.

The project employs a modular approach, allowing users to customize and expand the system according to their specific requirements and preferences. A user-friendly interface, either through a dedicated mobile application or a web-based dashboard, enables remote access and control of the home automation system from anywhere with internet.

In this Arduino project, we made an Arduino IoT project using ESP8266 Arduino UNO WIFI control Relay with sound sensor, manual switches, and mobile control. I have used all the free tools for this IoT-based home automation system.

1.4 Scope

The scope of a home automation system project utilizing Arduino Uno encompasses various aspects of designing, implementing, and deploying a functional system to automate tasks within a household. Here's a breakdown of the scope:

- Accessibility and Affordability: The use of Arduino Uno microcontrollers makes home automation projects more accessible and affordable to DIY enthusiasts, students, and hobbyists. Arduino boards are widely available and cost-effective, allowing individuals with limited resources to build custom automation solutions for their homes without significant financial investment.

- Educational Value: Home automation projects using Arduino Uno provide a practical and hands-on learning experience in electronics, programming, and system integration. They serve as valuable educational tools for students and beginners, helping them develop essential skills in hardware design, software development, and problem-solving.

- Empowerment and Independence: By building their own home automation systems, individuals gain a sense of empowerment and independence in controlling their living environment. They can customize and tailor the system to meet their specific needs and preferences, without relying on proprietary or commercial solutions.

- Innovation and Creativity: Arduino-based home automation projects encourage innovation and creativity, allowing individuals to experiment with new ideas and explore emerging technologies. Whether it's integrating voice control, implementing machine learning algorithms, or incorporating IoT devices, there's ample opportunity for creativity and experimentation.

- Customization and Flexibility: Arduino Uno-based home automation projects offer flexibility and customization options that may not be available in off-the-shelf solutions. Users can design and implement unique automation routines, personalize user interfaces, and integrate a wide range of sensors and actuators to create a system that perfectly suits their requirements.

- Community and Collaboration: The Arduino community is vibrant and supportive, providing a wealth of resources, tutorials, and forums where users can share knowledge, collaborate on projects, and seek assistance when needed. Home automation projects using Arduino Uno benefit from this collaborative ecosystem, fostering learning, innovation, and community engagement.

- Scalability and Expandability: Arduino Uno-based home automation projects are highly scalable and expandable, allowing users to add new features, sensors, and devices as needed. The modular nature of Arduino-based systems facilitates easy integration of additional components, enabling users to adapt and extend their automation systems over time.

- Empirical Understanding: Building a home automation project using Arduino Uno provides individuals with a deeper understanding of the underlying technologies and principles involved. Through hands-on experimentation and troubleshooting, users gain firsthand

experience in electronics, programming, and system architecture, enhancing their technical knowledge and skills.

By addressing these aspects within the scope of the project, developers can create a robust and versatile home automation system using Arduino Uno that caters to the specific needs and preferences of homeowners while offering opportunities for customization and expansion.

1.5 Significance

The significance of a home automation system using Arduino Uno lies in its ability to democratize and personalize smart home technology. Here's why it's significant:

• Accessibility: Arduino Uno boards are affordable and widely available, making them accessible to hobbyists, students, and DIY enthusiasts. This accessibility allows a broader range of individuals to participate in the creation of smart home solutions without significant financial investment.

• Customization: Arduino Uno platforms offer a high degree of customization, allowing users to tailor their home automation systems to meet their specific needs and preferences. From selecting sensors and actuators to programming control logic, users have the flexibility to design a system that fits their unique requirements.

• Learning Opportunity: Building a home automation system with Arduino Uno provides an excellent learning opportunity in electronics, programming, and system integration. Users gain hands-on experience in hardware design, coding, and troubleshooting, which can be valuable for students and beginners looking to develop practical skills.

• Scalability: Arduino-based home automation systems are highly scalable, allowing users to start with a basic setup and expand it over time as needed. Additional sensors, actuators, and features can be easily integrated into the system, providing flexibility for future upgrades and enhancements.

• Community Support: The Arduino community is vast and supportive, offering a wealth of resources, tutorials, and forums where users can share knowledge and seek assistance. Community support fosters collaboration, innovation, and problem-solving, empowering users to overcome challenges and explore new ideas.

• Empowerment: By building their own home automation system, users gain a sense of empowerment and control over their living environment. They become active participants in shaping their smart home experience, rather than relying on pre-packaged solutions that may not fully meet their needs.

• Affordability: Compared to commercial smart home products, DIY solutions based on Arduino Uno are often more affordable, particularly for individuals on a tight budget. This affordability allows users to experiment with home automation technology without breaking the bank, making it more accessible to a wider audience.

● Innovation: Arduino Uno platforms encourage innovation and experimentation, enabling users to explore creative ideas and implement novel solutions to everyday problems. Whether it's developing new automation routines, integrating sensors, or experimenting with IoT connectivity, there's ample opportunity for innovation in Arduino-based home automation projects.

In summary, the significance of a home automation system using Arduino Uno lies in its ability to democratize smart home technology, empower users to customize their systems, provide valuable learning opportunities, foster community collaboration, and encourage innovation—all at an affordable price point.

1.6 Structure of the Report

This report is organized into several chapters, each addressing specific aspects of the project:

Chapter 2 provides an overview of the architecture of Home Automation system using Arduino Uno. It includes working of general home automation system overview, components used and software used.

Chapter 3 discusses the working of this project. It includes basic working, circuit and explanations, setting up Blynk, uploading code.

Chapter 4 offers insights and recommendations based on the observations along with potential avenues for future research.

Finally, Chapter 5 concludes the report with a summary of key findings.

2. Architecture of Home Automation System using Arduino Uno

2.1 Home Automation System Overview:

A home automation system allows users to remotely control and monitor various household appliances and systems, such as lighting, heating, ventilation, air conditioning (HVAC), security cameras, and more. The system can be controlled via a user interface, typically a smartphone app, web interface, or voice commands through voice assistants like Google Assistant or Amazon Alexa.

Components of the Home Automation System:

- Microcontroller: The central processing unit of the system, responsible for executing commands, interfacing with sensors and actuators, and communicating with external devices. Common microcontrollers used in home automation projects include Arduino, Raspberry Pi, ESP8266, or ESP32.

- Sensors: Devices that detect changes in the environment or system parameters. Example include temperature sensors, motion sensors, light sensors, humidity sensors, gas sensors, etc.

- Actuators: Devices that perform physical actions based on commands received from the microcontroller. Examples include relays, motors, servo motors, solenoid valves, etc.

- Communication Modules: Components that enable communication between the microcontroller and external devices or networks. Common communication protocols include Wi-Fi, Bluetooth, Zigbee, Z-Wave, etc.

- User Interface: Interface through which users interact with the home automation system to issue commands and receive feedback. Can be a smartphone app, web interface, physical switches, or voice commands via voice assistants.

- Power Supply: Provides electrical power to the microcontroller, sensors, actuators, and other components of the system. Can be mains power or battery-powered, depending on the application and requirements.

Working Principle:

- Sensors detect changes in the environment or specific parameters (e.g., motion, temperature, light intensity).

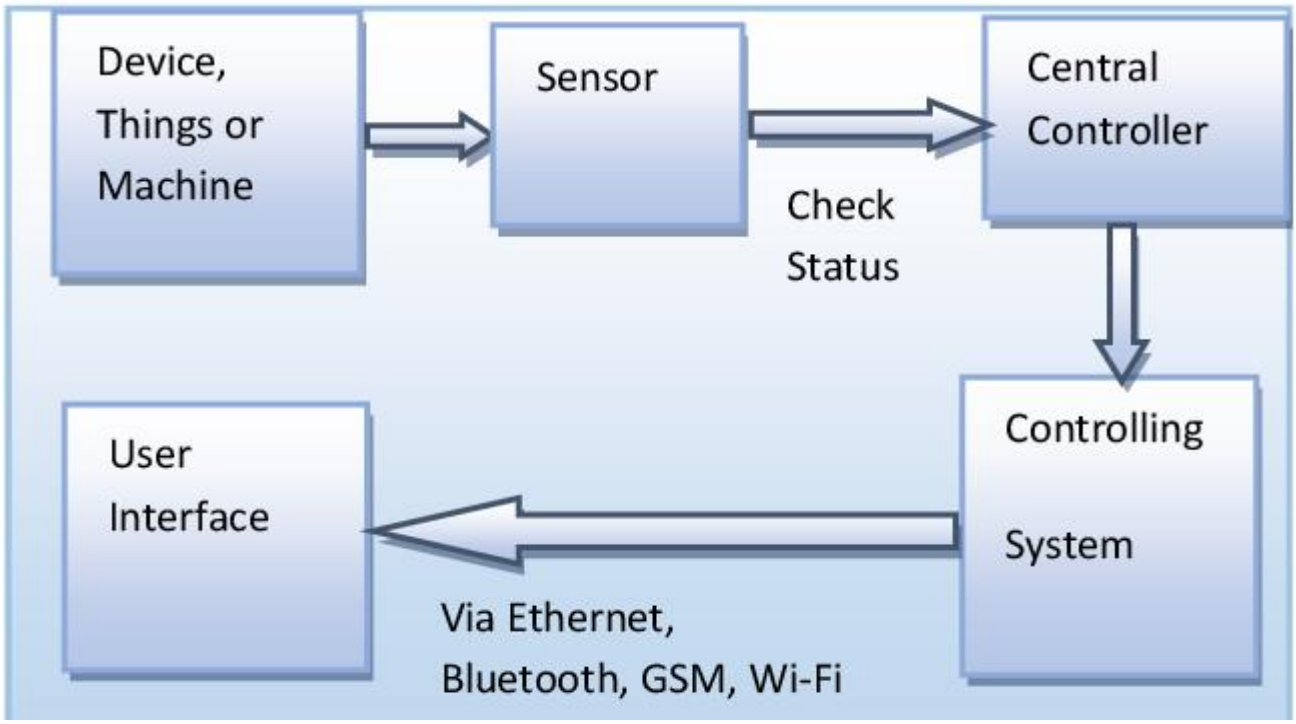
- The microcontroller processes data from the sensors and executes predefined logic based on programmed instructions or user commands.

- Based on the inputs received, the microcontroller sends commands to actuators to perform desired actions (e.g., turning on/off lights, adjusting thermostat settings).

- Users interact with the system through a user interface (smartphone app, web interface, or voice commands) to control and monitor the connected devices remotely.

- The microcontroller communicates with the user interface and other devices using communication modules such as Wi-Fi or Bluetooth.

- This system architecture enables users to create a customized and flexible home automation solution tailored to their specific needs and preferences.



2.2 Components used:

Required Components for this IoT Project (without PCB)

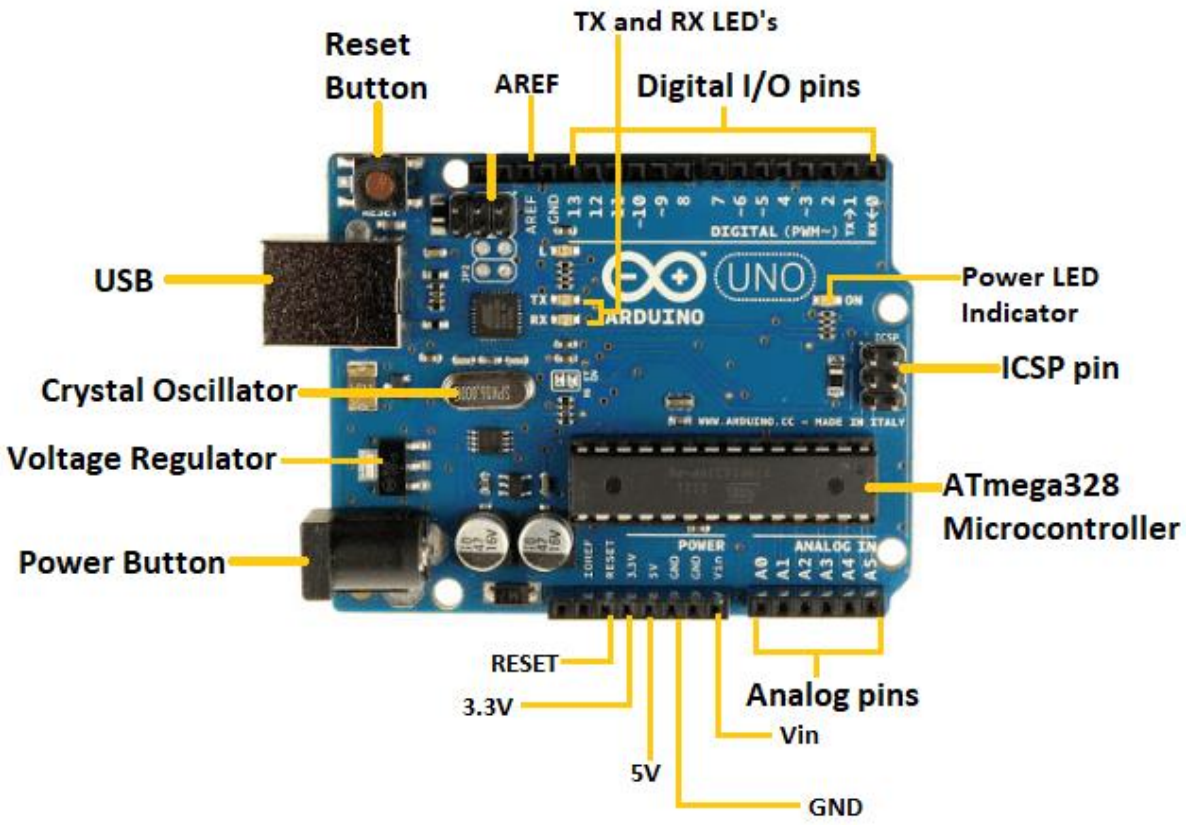
1. Arduino UNO

The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.

The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms.

The components of Arduino UNO board are shown below:



- ATmega328 Microcontroller- It is a single chip Microcontroller of the ATmel family. The processor code inside it is of 8-bit. It combines Memory (SRAM, EEPROM, and

Flash), Analog to Digital Converter, SPI serial ports, I/O lines, registers, timer, external and internal interrupts, and oscillator.

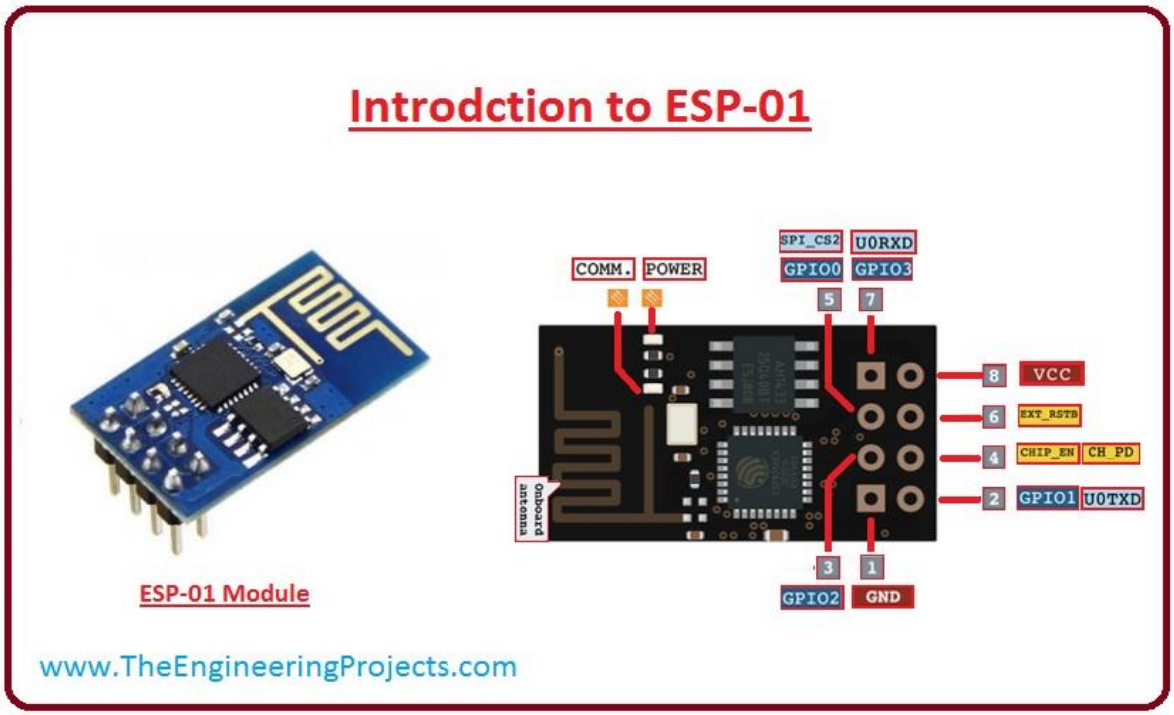
- ICSP pin - The In-Circuit Serial Programming pin allows the user to program using the firmware of the Arduino board.
- Power LED Indicator- The ON status of LED shows the power is activated. When the power is OFF, the LED will not light up.
- Digital I/O pins- The digital pins have the value HIGH or LOW. The pins numbered from D0 to D13 are digital pins.
- TX and RX LED's- The successful flow of data is represented by the lighting of these LED's.
- AREF- The Analog Reference (AREF) pin is used to feed a reference voltage to the Arduino UNO board from the external power supply.
- Reset button- It is used to add a Reset button to the connection.
- USB- It allows the board to connect to the computer. It is essential for the programming of the Arduino UNO board.
- Crystal Oscillator- The Crystal oscillator has a frequency of 16MHz, which makes the Arduino UNO a powerful board.
- Voltage Regulator- The voltage regulator converts the input voltage to 5V.
- GND- Ground pins. The ground pin acts as a pin with zero voltage.
- Vin- It is the input voltage.
- Analog Pins- The pins numbered from A0 to A5 are analog pins. The function of Analog pins is to read the analog sensor used in the connection. It can also act as GPIO (General Purpose Input Output) pins.

The technical specifications of the Arduino UNO are listed below:

- There are 20 Input/Output pins present on the Arduino UNO board. These 20 pins include 6 PWM pins, 6 analog pins, and 8 digital I/O pins.
- The PWM pins are Pulse Width Modulation capable pins.
- The crystal oscillator present in Arduino UNO comes with a frequency of 16MHz.
- It also has a Arduino integrated WIFI module. Such Arduino UNO board is based on the Integrated WIFI ESP8266 Module and ATmega328P microcontroller.
- The input voltage of the UNO board varies from 7V to 20V.
- Arduino UNO automatically draws power from the external power supply. It can also draw power from the USB.

2.ESP8266

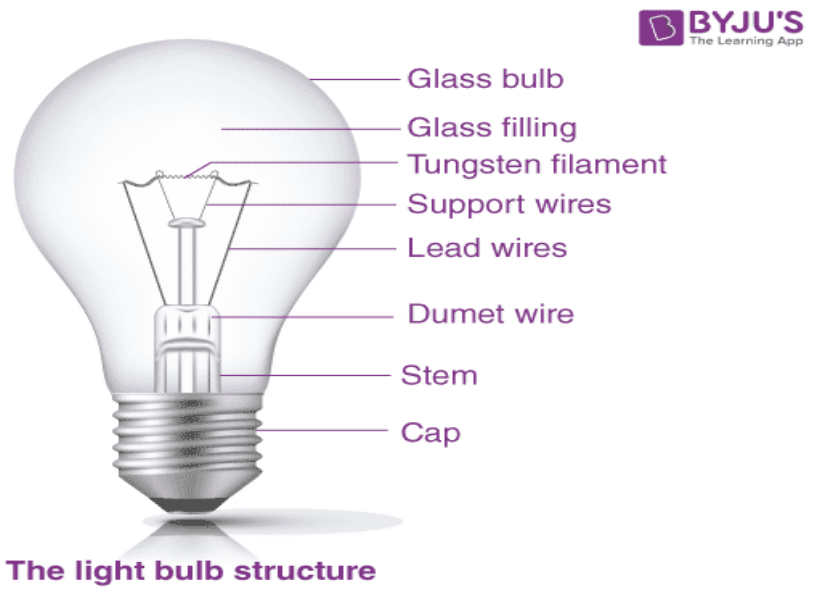
The ESP8266 ESP-01 is a Wi-Fi module that allows microcontrollers access to a Wi-Fi network. This module is a self-contained SOC (System On a Chip) that doesn't necessarily need a microcontroller to manipulate inputs and outputs as you would normally do with an Arduino, for example, because the ESP-01 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.



3. Electronic Bulbs

An electronic bulb is a small and simple light source that uses a wire filament to glow on the application of electricity. The structure of incandescent light bulbs is shown in the figure below. The light bulb consists of three key parts

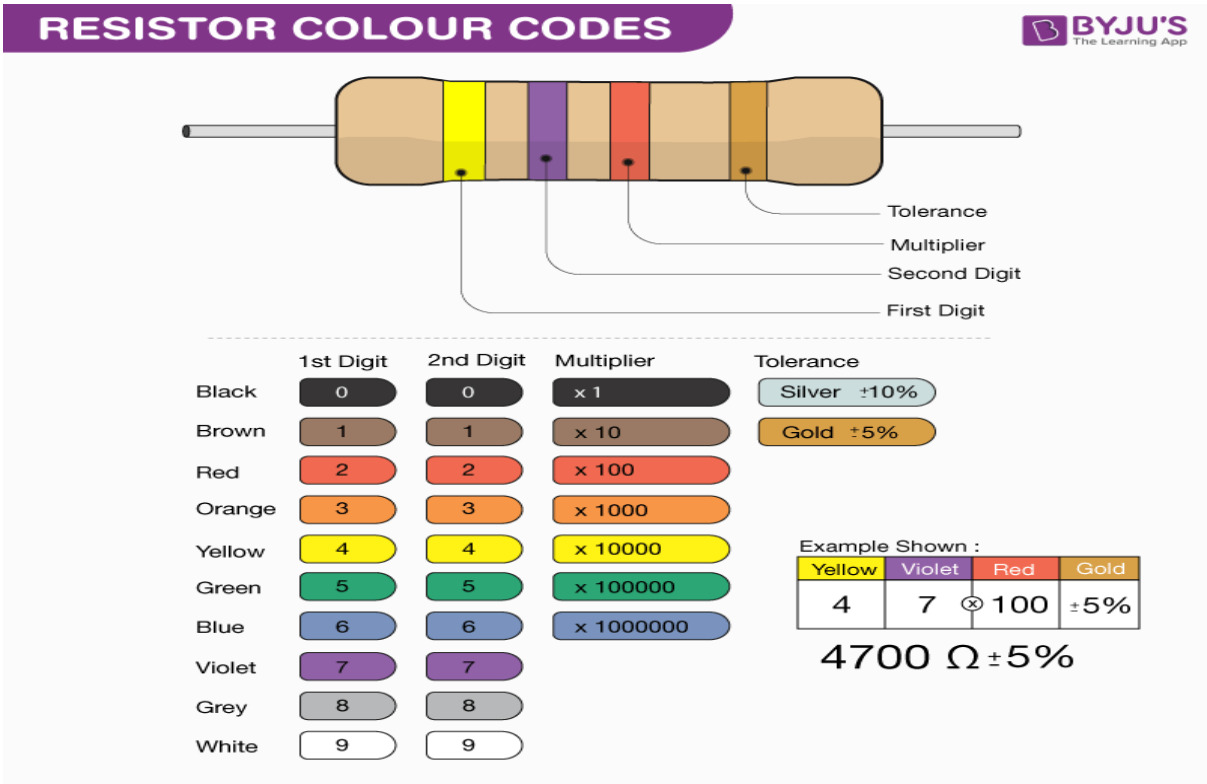
- The filament
- The glass bulb
- The base of the light bulb



4. 1k, 2k, 4.7k resistors (1/4 watt)

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.

Resistor color code:



5. 5-mm LED

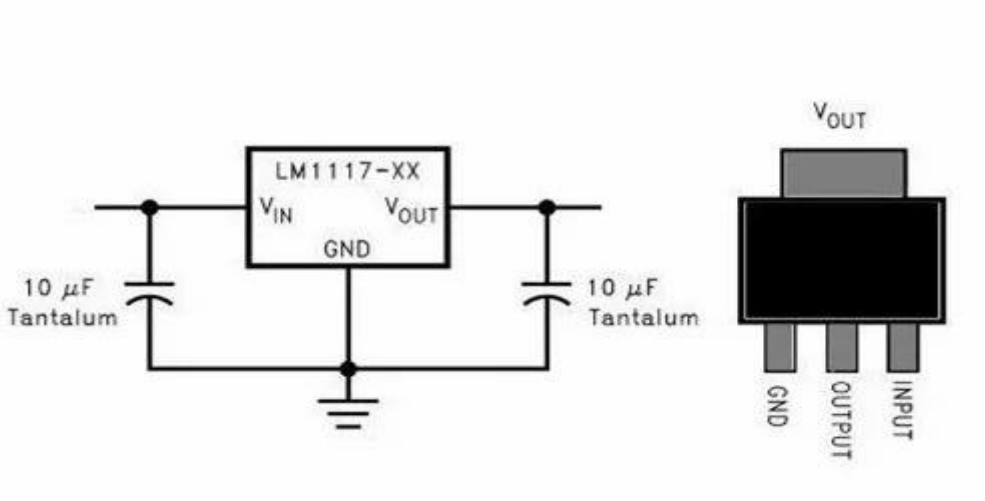
A light-emitting diode is a semiconductor device that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light is determined by the energy required for electrons to cross the band gap of the semiconductor.

The 5mm LED can be used anywhere where you need low power, high-intensity reliable light, or indication. They go quickly into a breadboard and will add that extra zing to your project. The 5mm T1 3/4 LED is the most common size of LED available.

6. 1117 3.3V voltage regulator

A voltage regulator is a component of the power supply unit that ensures a steady constant voltage supply through all operational conditions. It regulates voltage during power fluctuations and variations in loads. It can regulate AC as well as DC voltages. It is used in case of Arduino nano but since we are using Arduino uno we can avoid it.

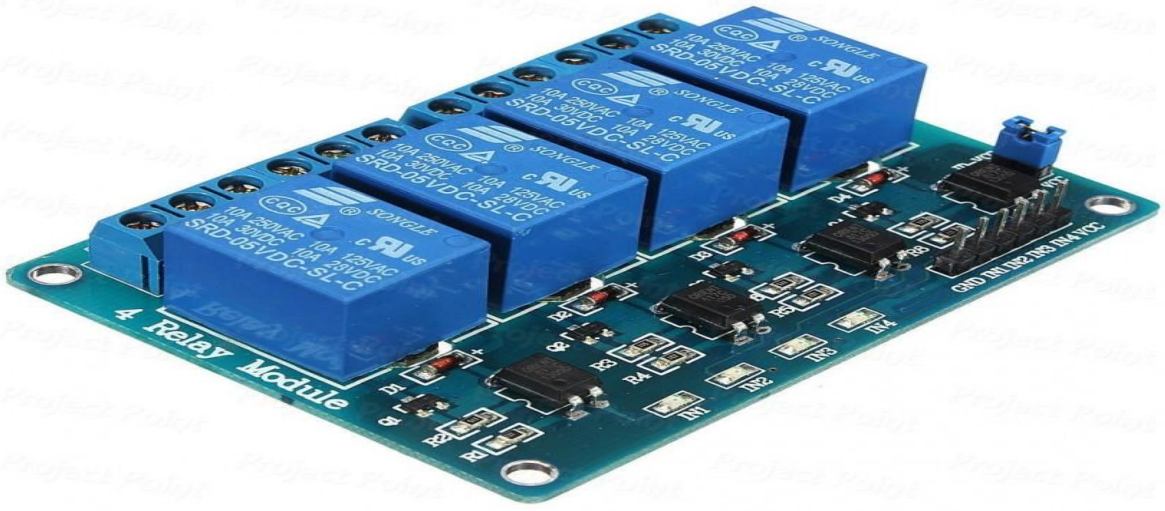
AMS1117-3.3 is a cost effective, low dropout (LDO) Voltage Regulator designed to provide current up to 1 amp. The maximum input voltage it can support is 15V.



7. 4-channel 5V SPDT Relay Module

The 4 Channel Relay Module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontroller such as Arduino, PIC and etc. The relays terminal (COM, NO and NC) is being brought out with screw terminal.

Use this relay to control high current motors, 240V mains appliances, etc. To ensure maximum safety the relay is completely closed and provides complete isolation between the driving circuit and the load. These are high-quality Single Pole - Double Throw (SPDT) sealed 5V Sugar Cube relays.



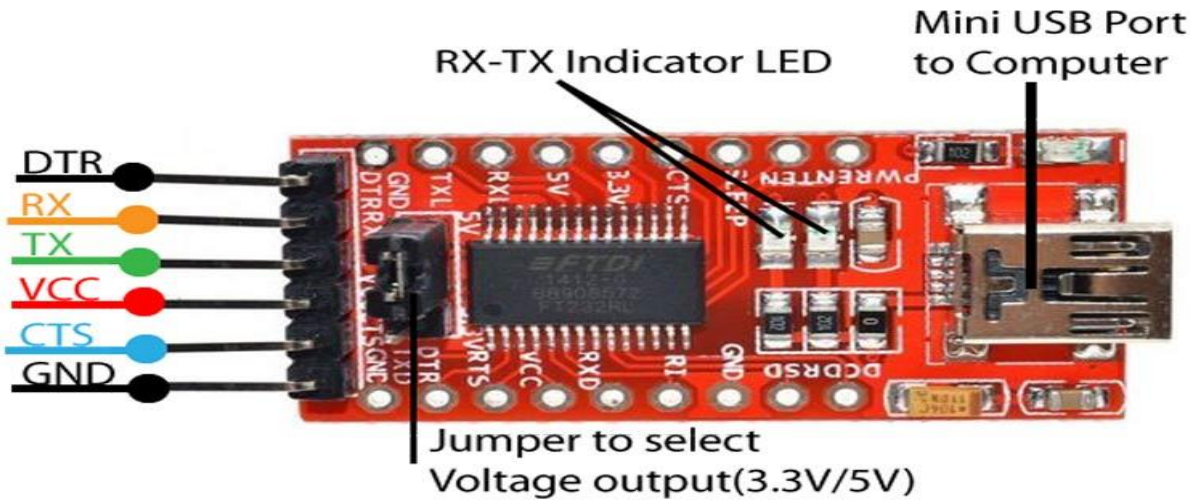
8. Switches or Push Buttons

Switch: a device for making, breaking, or changing the connections in an electrical circuit.

Push buttons: A push button switch is a mechanical device used to control an electrical circuit in which the operator manually presses a button to actuate an internal switching mechanism. They come in a variety of shapes, sizes, and configurations, depending on the design requirements.

9. FTDI232 USB to TTL

The TTL-232R-3V3 is a USB to Serial (TTL level) converter cable which allows for a simple way to connect TTL interface units to USB. This version of FTDI's USB to TTL serial adapter cables has it's I/O pins configured to operate at 3.3V levels.

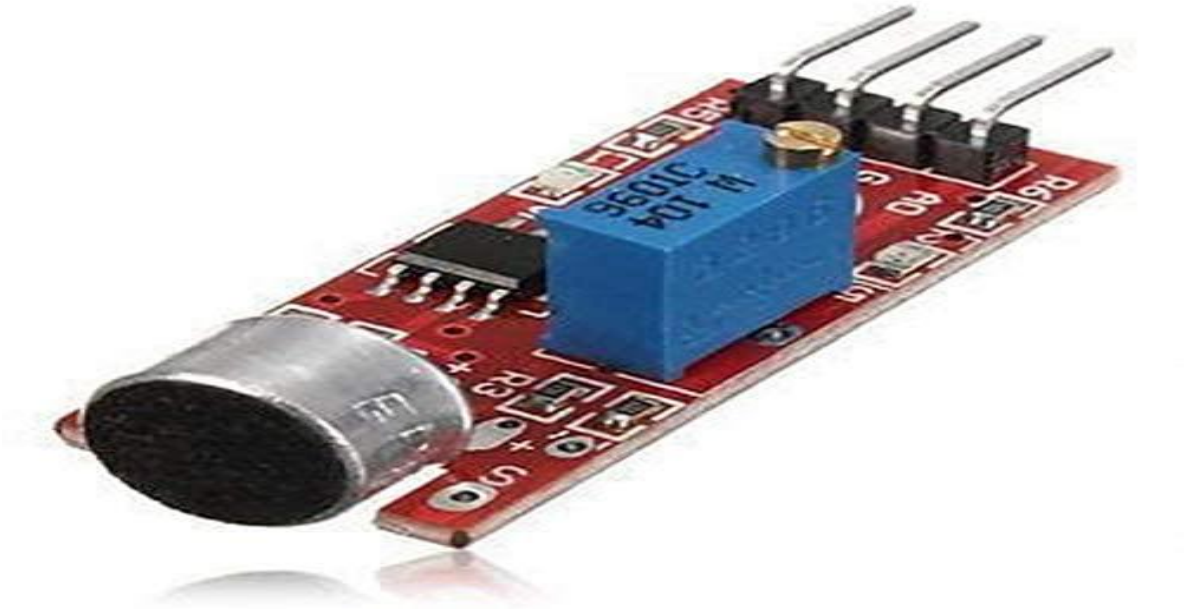


10. 5V DC supply.

A DC power supply provides direct current (DC) voltage to power and test a device under test such as a circuit board or electronic product.

11. Sound Sensor

A sound sensor is a device that converts sound waves into an electrical signal that can be processed by an electronic circuit. They can be used to detect and measure the amplitude, frequency, and duration of sound waves

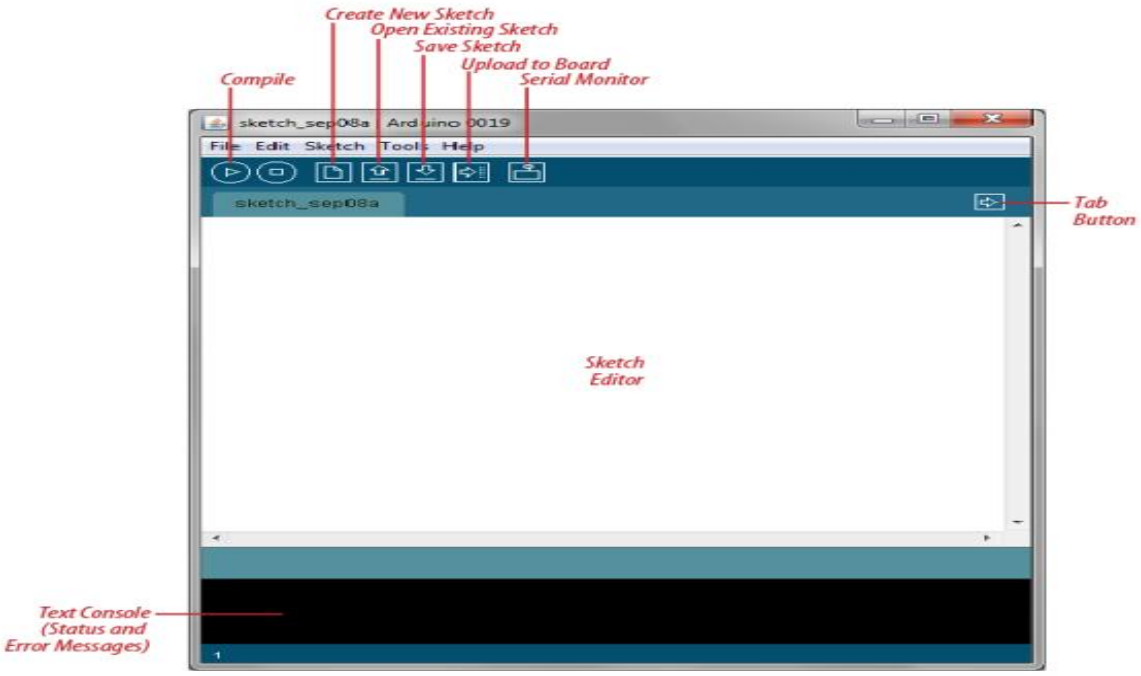


2.3 Software used

a) Arduino IDE

- It is an open-source software.
- Easy to write code and upload it to the physical board.
- Easy to learn programming language with its inbuilt functions.
- Runs on windows, Mac OS and Linux.
- This software can be used with any Arduino board.
- A program written with the IDE for Arduino is called a sketch.
- The IDE enables to write and edit code and convert this code into instructions that Arduino hardware understands. The IDE also transfers those instructions to the Arduino board (a process called as uploading).
- The Arduino IDE supports the languages C and C++ using special rules of code structuring.

- It consists of only two functions, setup and loop.
- The setup function is used to initialize variables, input and output pin modes and other libraries needed in the sketch.
- After setup has been called, function loop is executed repeatedly in the main program. It controls the board until the board is powered off or reset.



We can program the Arduino UNO using the Arduino IDE. The Arduino IDE is the Integral Development program, which is common to all the boards.

We can also use Arduino Web Editor, which allows us to upload sketches and write the code from our web browser (Google Chrome recommended) to any Arduino Board. It is an online platform.

The USB connection is essential to connect the computer with the board. After the connection, the PWR pins will light in green. It is a green power LED.

The steps to get started with Arduino UNO are listed below:

- Install the drivers of the board.

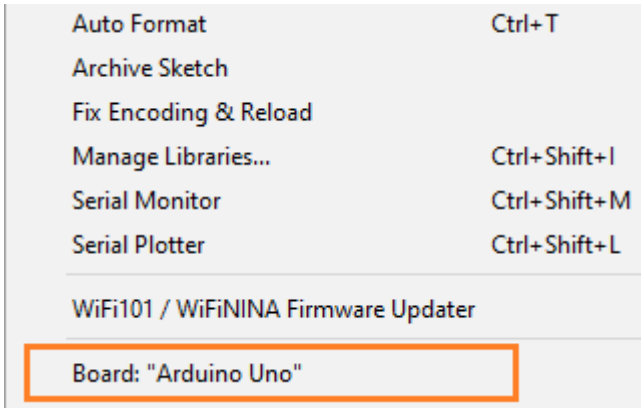
As soon we connect the board to the computer, Windows from XP to 10 will automatically install the board drivers.

But, if you have expanded or downloaded the zip package, follow the below steps:

- Click on Start -> Control Panel -> System and Security.
- Click on System -> Device Manager -> Ports (COM &LPT) -> Arduino UNO (COMxx). If the COM &LPT is absent, look Other Devices -> Unknown Device.
- Right-click to Arduino UNO (COmxx) -> Update Driver Software -> Browse my computer for driver software.
- Select the file "inf" to navigate else, select "ArduinoUNO.inf" .
- Installation Finished.

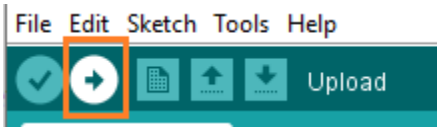
- Open the code or sketch written in the Arduino software.
- Select the type of board.

Click on 'Tools' and select Board, as shown below:



- Select the port. Click on the Tools -> Port (select the port). The port likely will be COM3 or higher. For example, COM6, etc. The COM1 and COM2 ports will not appear, because these two ports are reserved for the hardware serial ports.
- Now, upload and run the written code or sketch.

To upload and run, click on the button present on the top panel of the Arduino display, as shown below:



Within the few seconds after the compile and run of code or sketch, the RX and TX light present on the Arduino board will flash.

The 'Done Uploading' message will appear after the code is successfully uploaded. The message will be visible in the status bar.

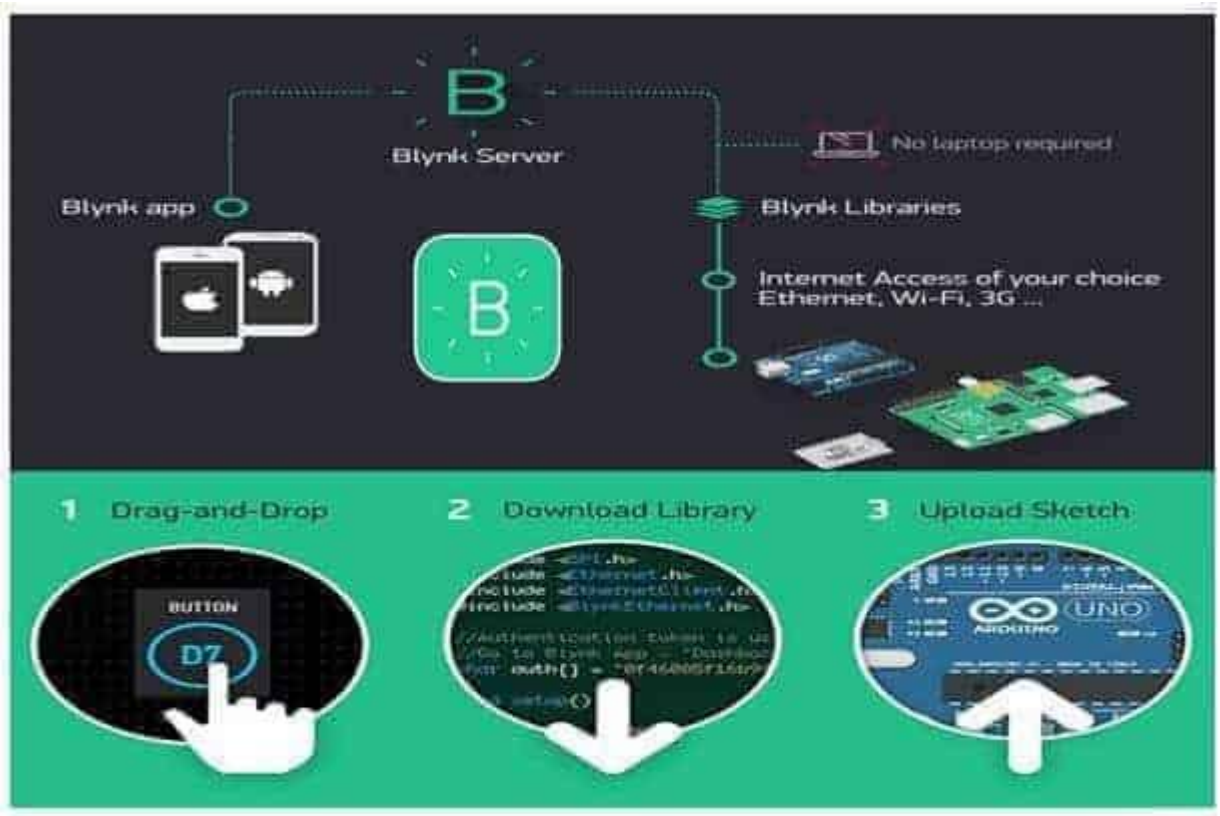
b) Blynk IOT

Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet. This application is used to create a graphical

interface or human machine interface (HMI) by compiling and providing the appropriate address on the available widgets.

Everything you need to build and manage connected hardware: device provisioning, sensor data visualization, remote control with mobile and web applications, over-the-air firmware updates, private cloud, data analytics, user and access management, alerts, automations and much more…

Entrepreneurs, engineers, and large enterprises use Blynk's low-code platform to build user-friendly IoT apps and manage devices, data, and customers in a secure cloud environment.

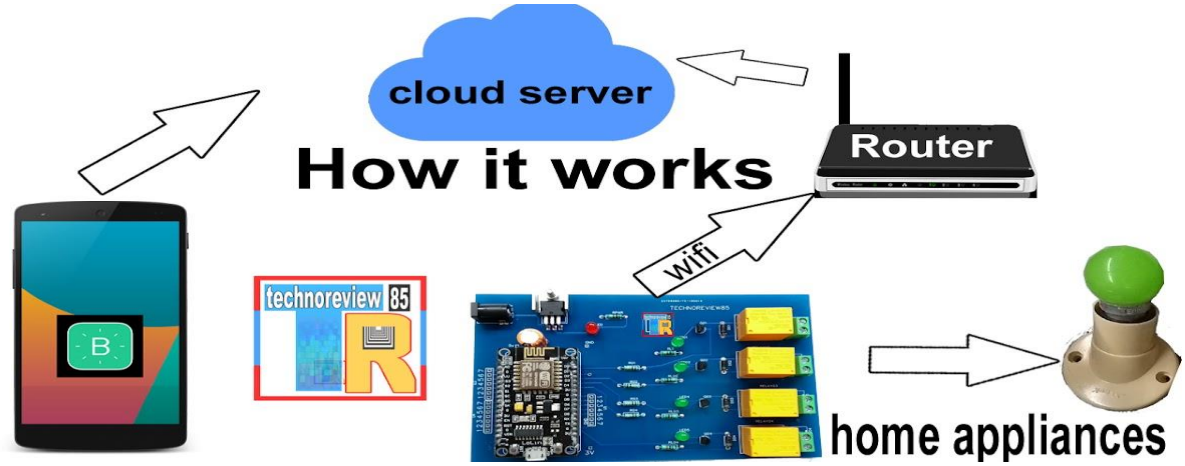


3. Working of Home Automation using Arduino Uno

3.1 Basic working

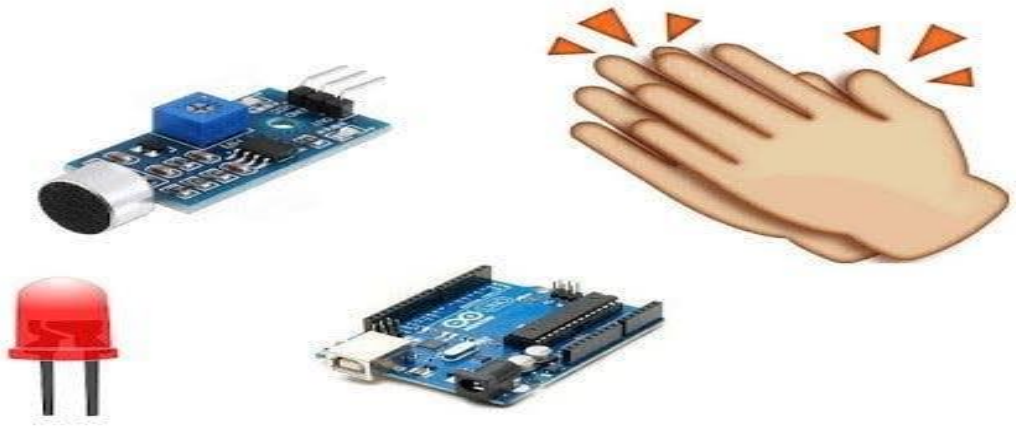
It is an Arduino IoT project using ESP8266 Arduino UNO WIFI control Relay with sound sensor, manual switches, and mobile control. After the power cut Arduino can remember the previous state so when the power comes back the appliances can automatically turn on according to the previous state.

When you press on your device in the Blynk app through your smartphone, it sends the signal to Blynk server, then ESP8266 will receive the signal from Blynk and forward the same signal to Arduino through the serial communication.



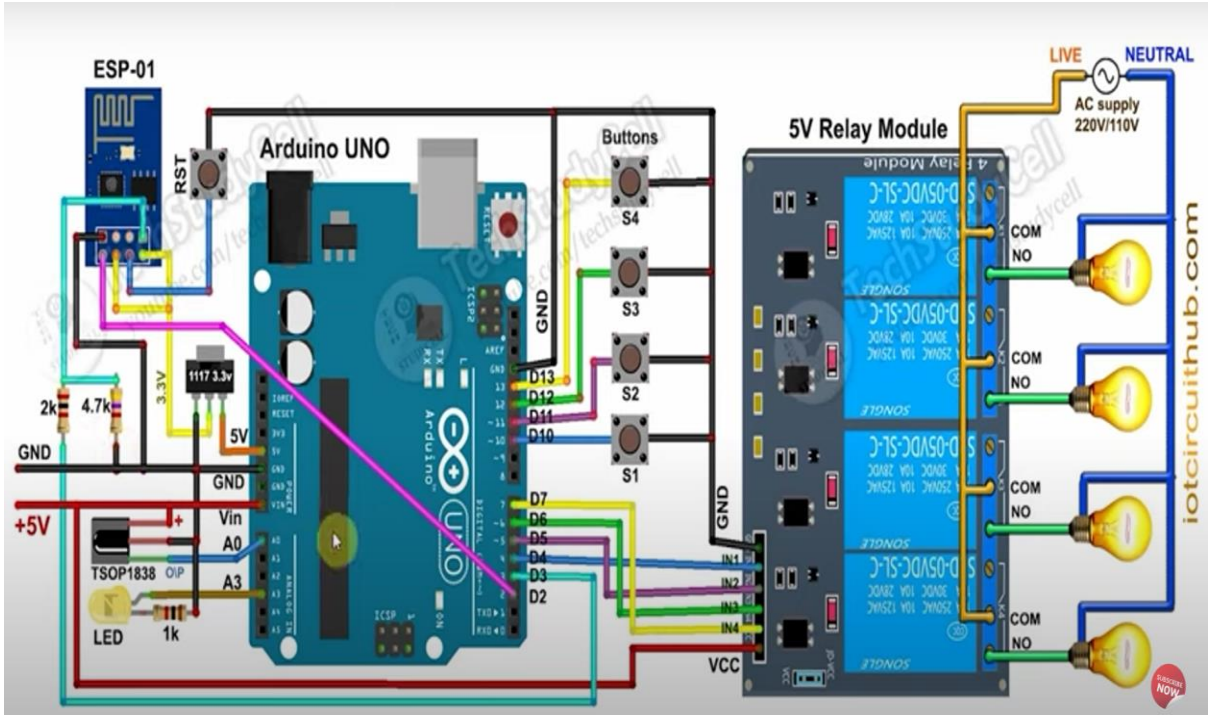
Then Arduino process that signal and accordingly turn on or off the relays and send feedback to ESP8266 again through serial communication. Then ESP8266 again sends feedback to Blynk server.

For sound sensor automation, the sound sensor senses the physical sound coverts it to suitable input and give it to Arduino which then gives digital output to relay module to control the appliances/ devices i.e. turn them on and off.



3.2 Circuit and explanations

For WIFI control through Blynk IOT:



D4, D5, D6 & D7 GPIO are used to control the 4-channel relay module.

And the GPIO D10, D11, D12 & D13 are connected with switches to control the relay module manually.

For the serial communication with the ESP8266, we have used D2 as RX and D3 as TX with the SoftwareSerial library.

Here we have made a voltage divider using 2k and 4.7k resistors to drop down the 5V logic level to the 3.3V logic level for the serial communication with the ESP8266 WIFI module.

We have used the INPUT_PULLUP function in Arduino IDE instead of using the pull-up resistors with each switch.

As per the source code, when the control pins of the relay module receive the LOW signal the relay will turn on and the relay will turn off for the HIGH signal in the control pin.

We have used an 1117 3.3V voltage regulator to supply the ESP8266. If you use Arduino UNO then you can use the 3.3V pin instead of the 1117 3.3V regulator but for Arduino Nano, we have to use the 1117 3.3V voltage regulator.

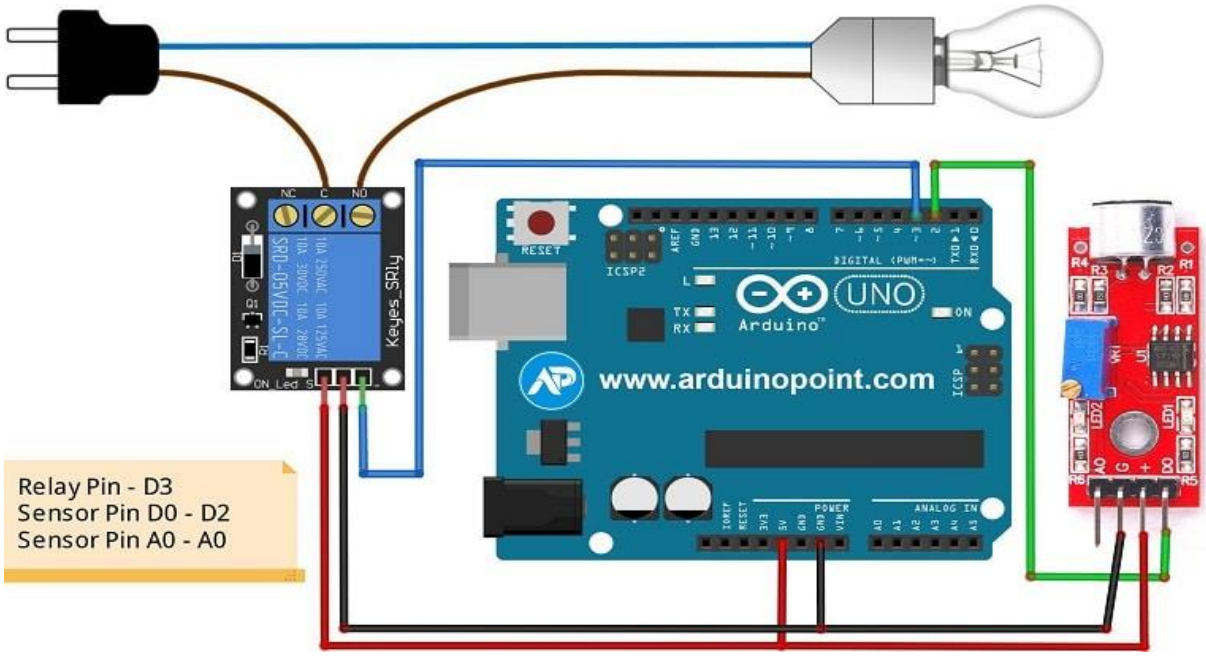
Switches are connected across the digital pins of Arduino and ground. The two resistors act as voltage divider to drop down the 5-volt logic level from Arduino UNO to 3.3-volt for the ESP8266. LED is connected to analog pins of Arduino UNO through a 1-ohm resistor.

ESP8266 is connected to Arduino for serial communication in both directions. Arduino is connected to relay board through digital pins to give the output to the relay board according to which the relay board turns on or off the appliances. Switches are also connected in between them to manually operate the devices when WIFI is off or when we want manual operation.

In case the WIFI is off the LED will be on and ESP-01 is not connected to WIFI which implies that the real time feedback is obtained through manual switches. When the WIFI is on the LED is off so ESP8266 is connected to WIFI so the appliances can be controlled through Blynk-Smartphone interface.

The input to microcontroller (when WIFI is on) is through Blynk-smartphone interface which comes into its analog pins, from which the Arduino UNO generates a digital output corresponding to the input and burnt code which is then given to relay board for appliance control. When WIFI is off relay board gets input from manual switches for appliance control.

For clap/ sound control:



Here the relay pin is connected to digital pin D3 of Arduino Uno and sensor pin to digital pin D2. Here the sound sensor senses the clap and passes input to Arduino which then gives digital output to relay to turn on the devices.

3.3 Uploading Code

We have two codes one for Arduino and ESP8266 and one for the sound sensor. Different Arduino codes are there for switch and push buttons.

In the ESP8266 code after installing all the libraries enter WIFI name and password, template id, template name and auth token from Blynk, and device IDs from devices section in Blynk. After writing the rest of the code it is uploaded from the tools section by selecting the appropriate board and port.

Code for sound sensor:
https://docs.google.com/document/d/1Yh_O6wBigRyDwGRGq9V3wb1fO4q_TCq-kpuM1dcX5UE/edit

Code for Blynk:

```
#define BLYNK_TEMPLATE_ID "TMPL3z3LVCvxm"
#define BLYNK_TEMPLATE_NAME "iot tempelate"
#define BLYNK_AUTH_TOKEN "iYNcmoDJtPq9twdp1KfwLCi7qbfgLH8n"

#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "#redkme";
char pass[] = "12345678";

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
  // Set incoming value from pin V0 to a variable
  int value = param.asInt();

  // Update state
  Blynk.virtualWrite(V1, value);
}


// This function sends Arduino's uptime every second to Virtual Pin 2.
void myTimerEvent()
{
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V2, millis() / 100);
}

void setup()
{
  // Debug console
  Serial.begin(115200);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  timer.setInterval(100, myTimerEvent);
}

void loop()
{
  Blynk.run();
  timer.run();

}
```

3.4 Setting up Blynk IOT

Steps to getting started with Blynk:

Step 1. DOWNLOAD BLYNK APP FOR ANDROID OR iOS

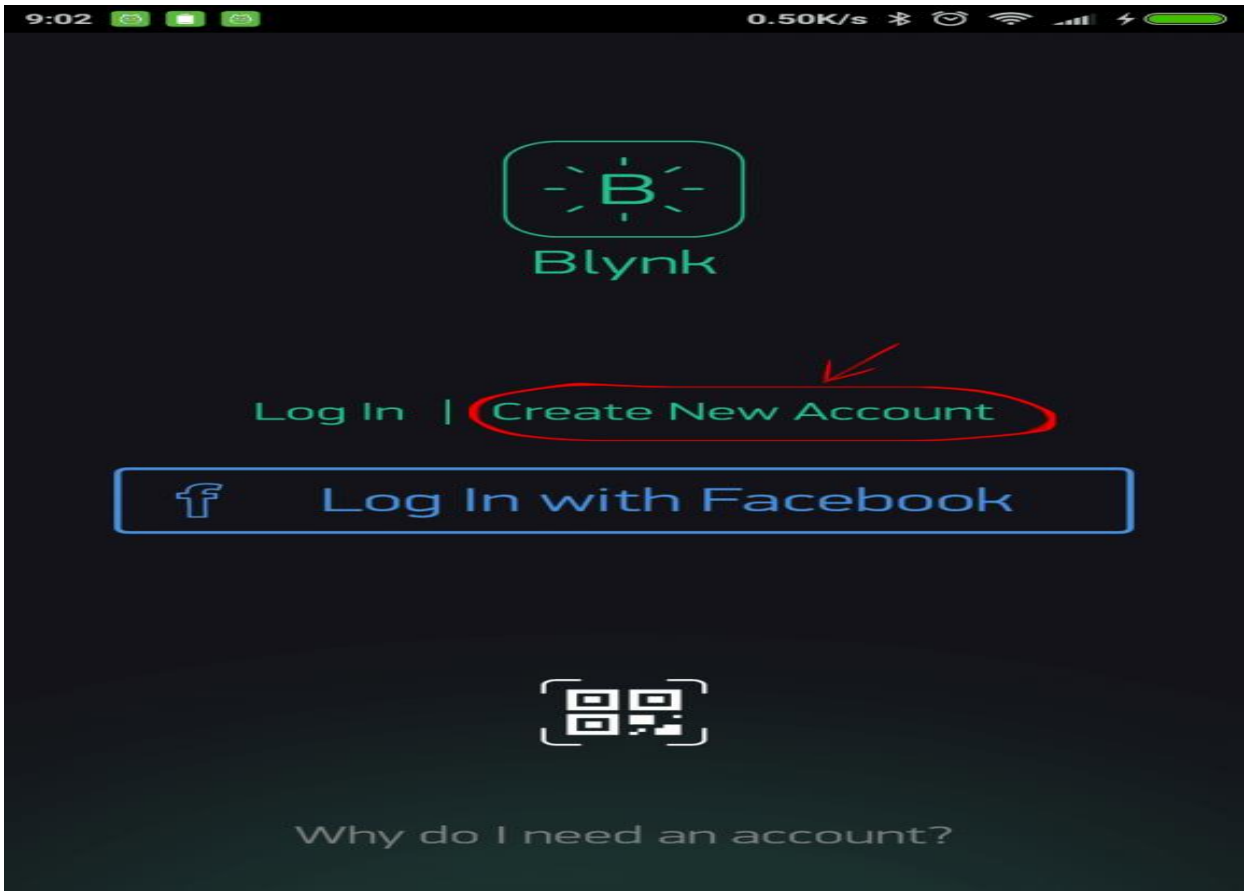
- For Android Blynk
- For iPhone Blynk

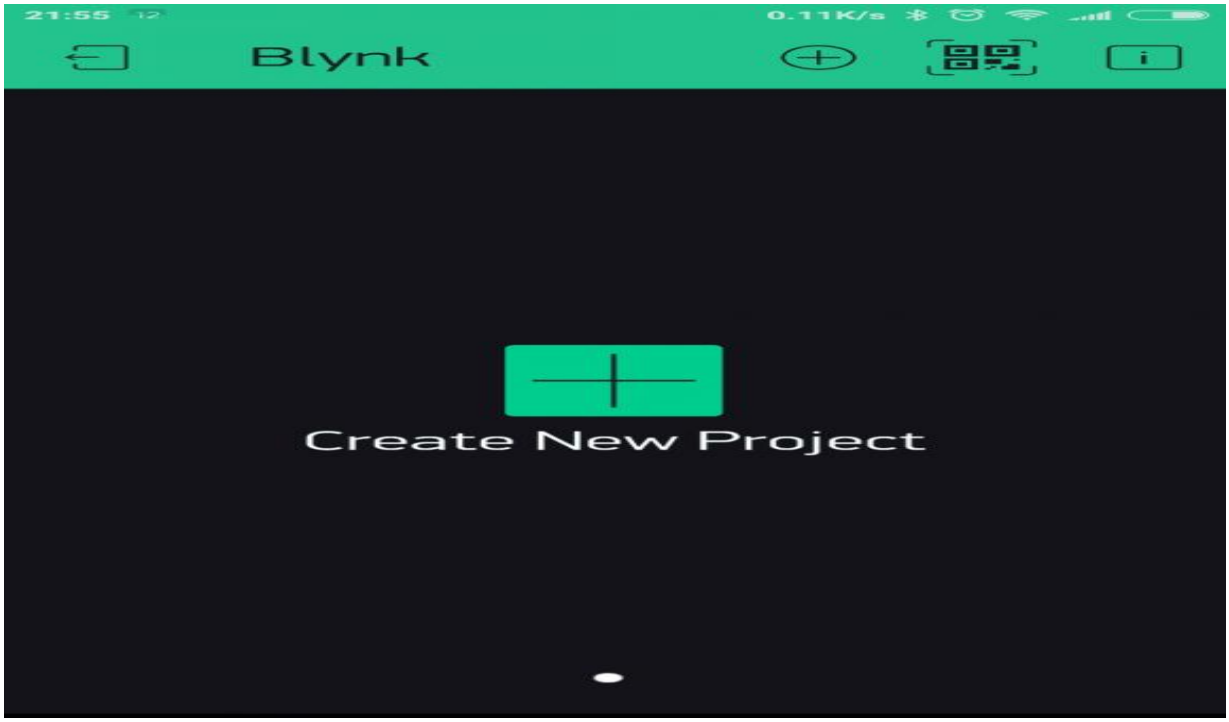
Step 2. INSTALL BLYNK LIBRARY

You can download Blynk library for Arduino from above. Download the zip file and unzip it to the library folder of Arduino IDE (C:\Program Files\Arduino\libraries) if you are unable to find this folder follow this guide.

Step 3. BUILD YOUR FIRST APP

- Open Blynk IoT App on your smart phone and create new account and then new project after successfully login.





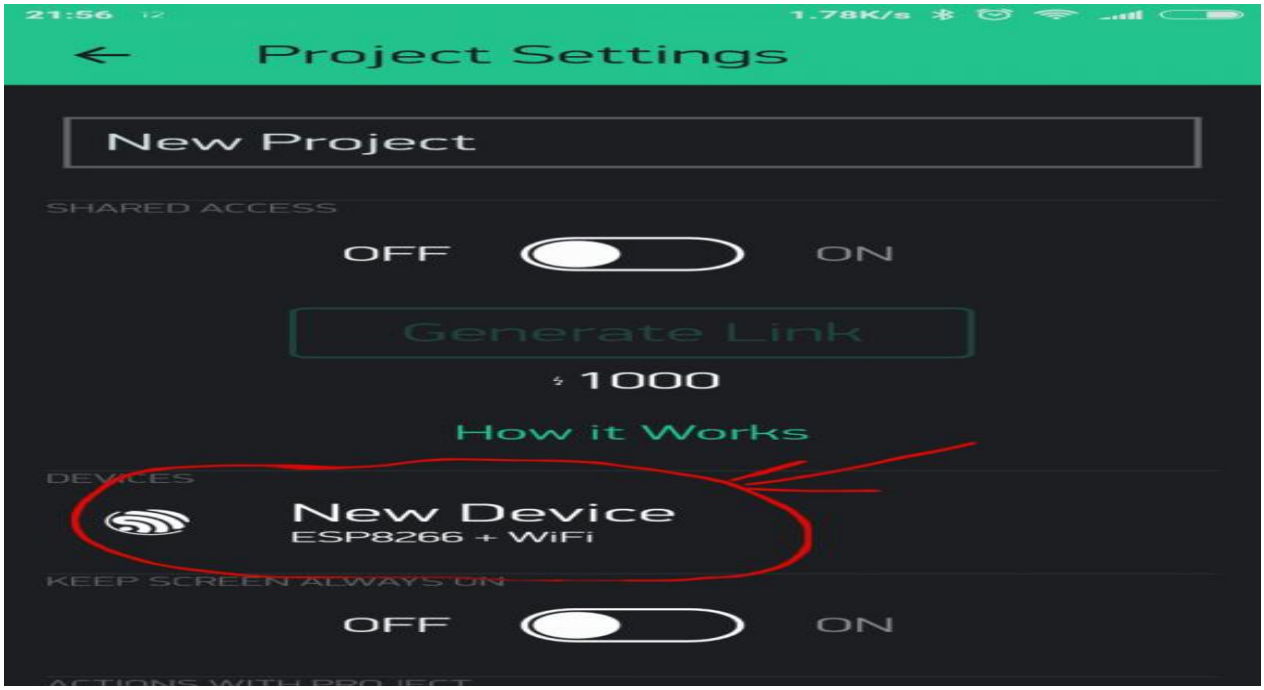
Give name to it

- Choose the hardware and communication type you are going to use click here to check list of supported devices

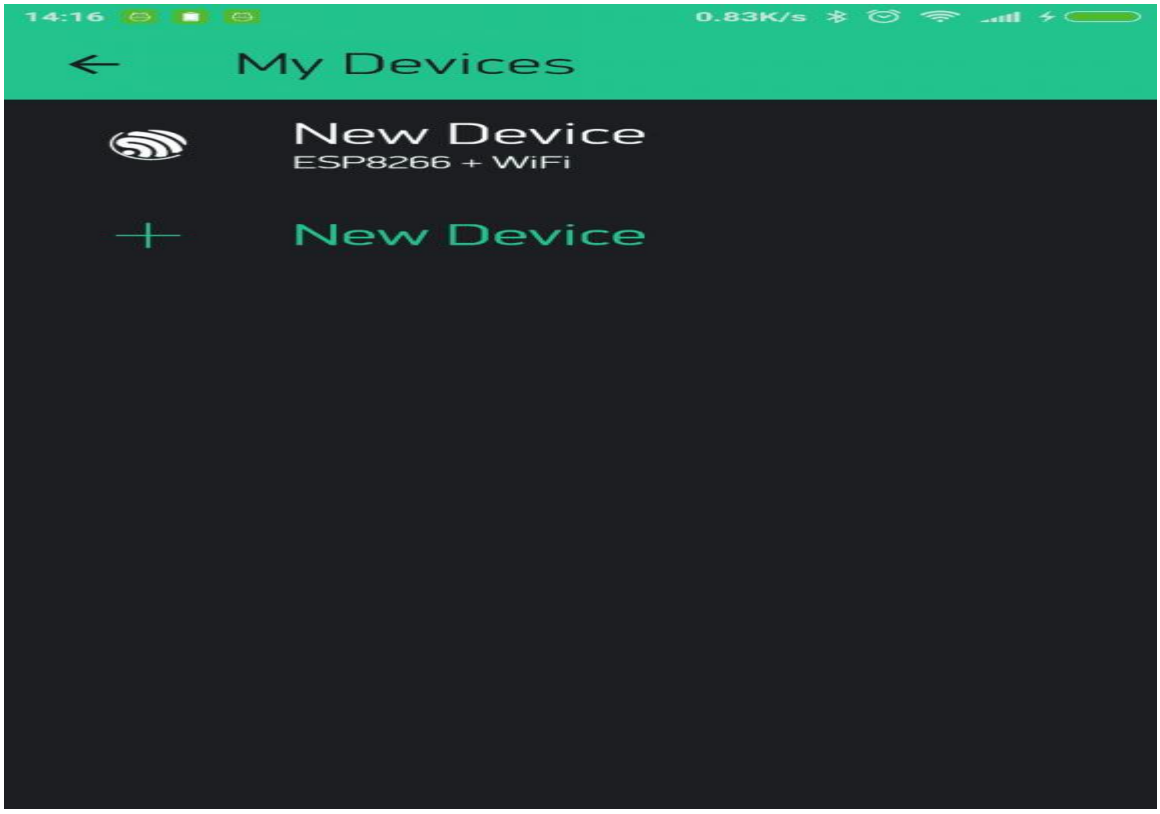
- Obtain Auth Token:

For obtaining token follow following steps:

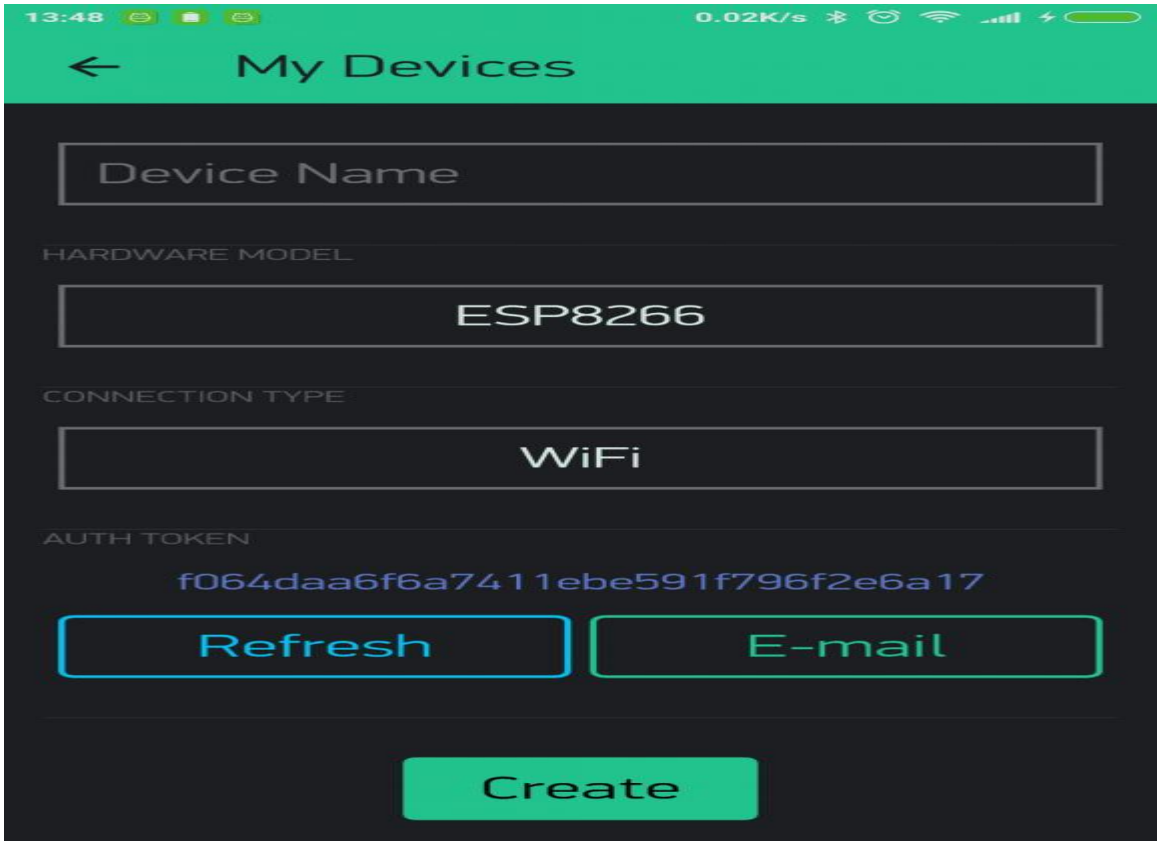
a) Select the device for communication:



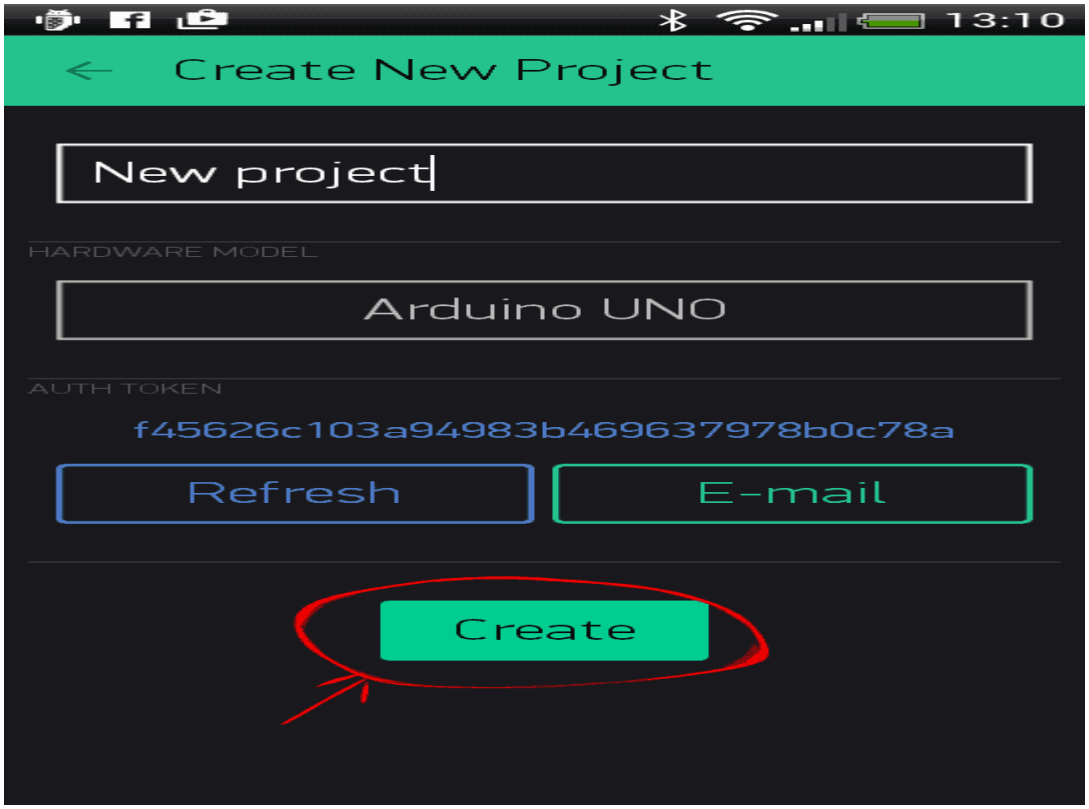
b) Click on device



c) See the Auth Token

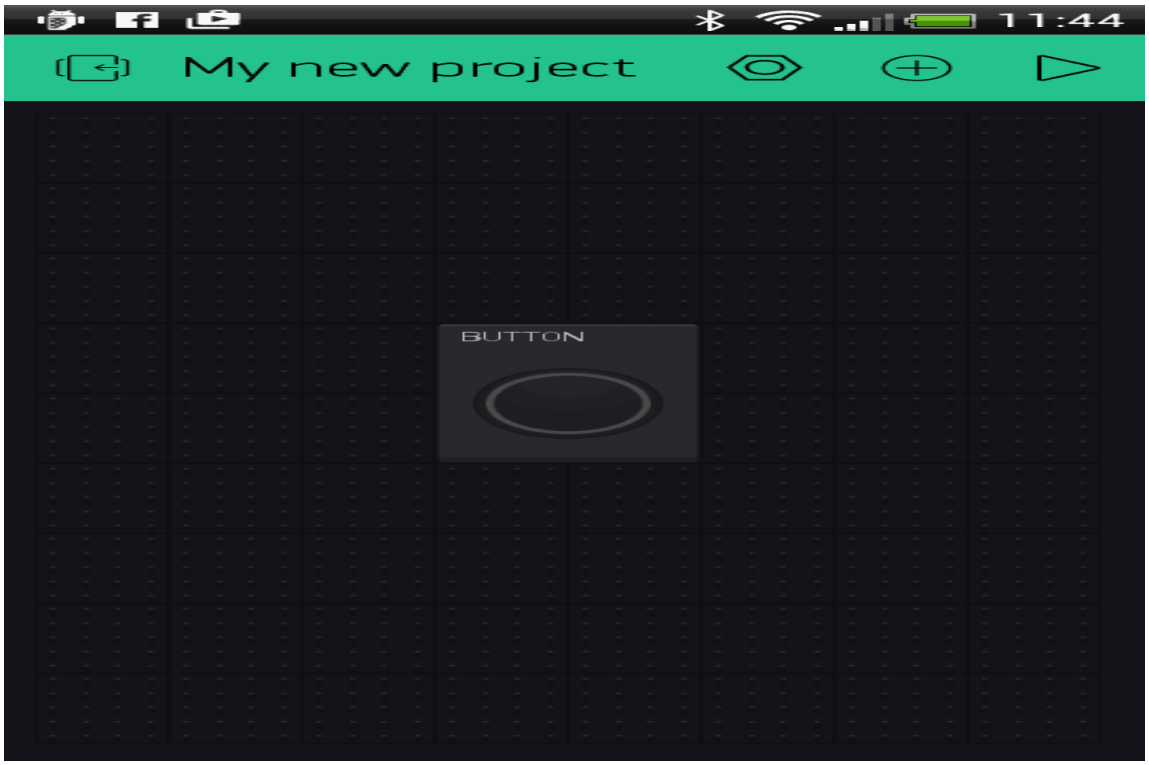


d) Press create button

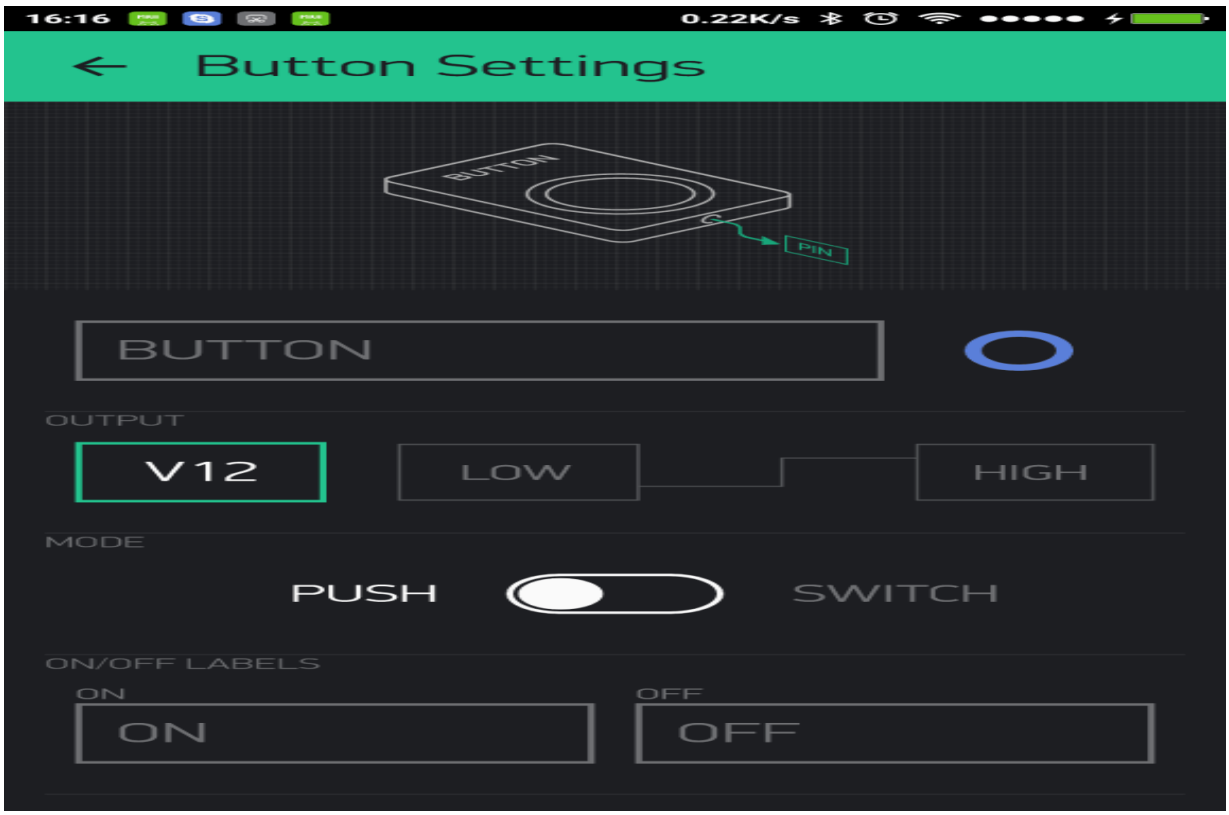


- Add the Button widget. Tap once to get to Widget Settings

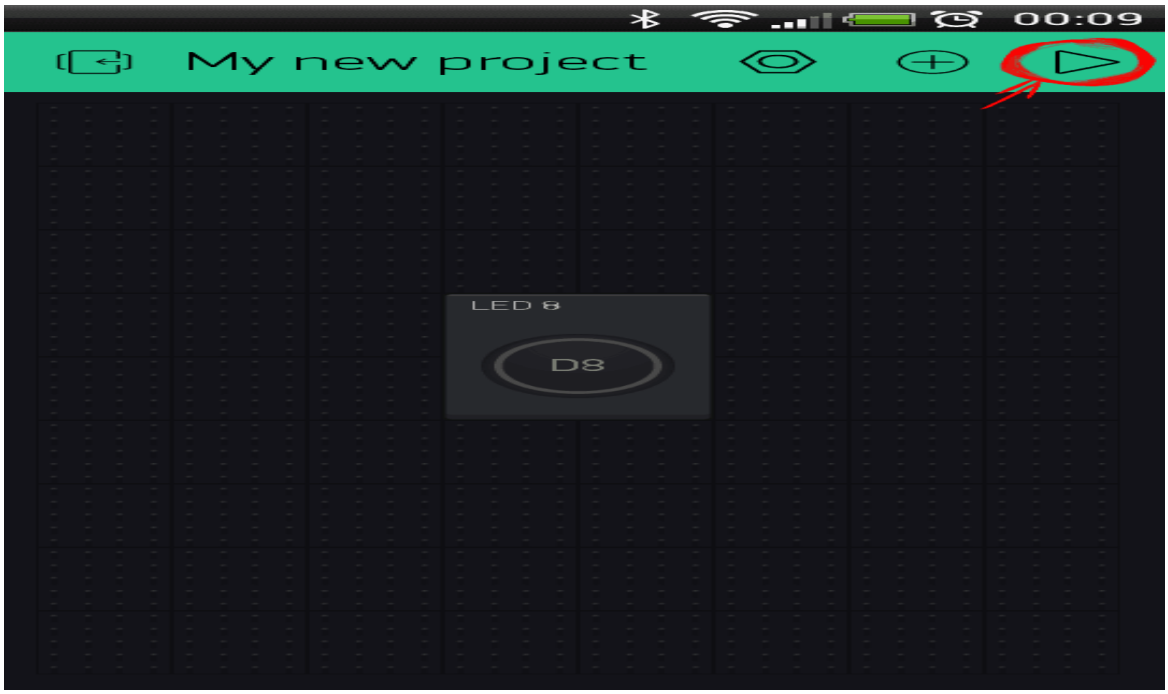
You can hold and drag it to reposition



- In Widget Settings set the PIN you want to control (e.g. if your LED is connected to pin 7 – choose pin D7): Each Widget has its own settings. Tap on the widget to get to them.



- Run the project



4. Experimentation and Future Work

Experimentation:

1. Performance Testing:

Latency Analysis: Measure the response time of the system when commands are issued through Blynk and claps. Evaluate any delays and identify potential bottlenecks.

Reliability Testing: Run the system continuously to assess its stability over time. Check for any instances of dropped connections or unexpected behavior.

2. User Experience Evaluation:

User Feedback: Collect feedback from users who interact with the system. Identify pain points, areas of confusion, or suggestions for improvement.

Usability Testing: Conduct usability tests to ensure that the integration with Blynk is intuitive and user-friendly.

3. Scalability Assessment:

Multiple Device Support: Test the system with multiple ESP8266 devices controlled by a single Blynk account. Evaluate how well the system scales with increased device load.

Network Congestion Testing: Simulate network congestion scenarios to determine how the system handles high traffic volumes.

Future Work:

1. Enhanced Functionality:

Integration with Other IoT Devices: Explore the possibility of integrating the system with other IoT devices or platforms, such as smart lights, thermostats, or security cameras.

Custom Voice Commands: Allow users to define custom voice commands for controlling specific actions or sequences within the system.

2. Security Improvements:

Data Encryption: Implement encryption protocols to secure communication between the Arduino Uno, ESP8266, and cloud-based services.

Access Control: Implement user authentication mechanisms to restrict access to the system and prevent unauthorized usage.

3. Optimization:

Code Refactoring: Refactor the Arduino code to improve efficiency and reduce resource consumption, particularly on the ESP8266.

Power Consumption Optimization: Investigate methods for reducing power consumption to prolong battery life or minimize energy usage in the system.

4. Integration with AI/ML:

Predictive Analysis: Explore the use of AI or machine learning algorithms to analyze usage patterns and predict user commands, improving the system's responsiveness and user experience.

Behavioral Adaptation: Develop algorithms that allow the system to adapt its behavior based on user preferences and historical interaction data.

5. Cross-Platform Compatibility:

Support for Additional Voice Assistants: Investigate the feasibility of integrating the system with other voice assistant platforms such as Google Assistant and Alexa, Siri or Cortana.

6. Documentation and Community Building:

Comprehensive Documentation: Create detailed documentation covering installation instructions, usage guidelines, troubleshooting tips, and examples for extending the system's functionality.

Community Forums: Establish an online community where users can share their experiences, ask questions, and collaborate on further development and enhancements.

5. Conclusion

The Arduino IoT project integrating Blynk with ESP8266, Arduino Uno, WIFI control, sound sensor and relay has showcased the immense potential of smart home automation. Through this project, we have successfully demonstrated how everyday devices can be seamlessly controlled and monitored using sound commands.

The experimentation phase provided valuable insights into the performance, user experience, and scalability of the system. By conducting performance testing, we identified areas for optimization to enhance response times and reliability. User feedback and usability testing enabled us to refine the interface, ensuring a smooth and intuitive interaction process for users.

Looking ahead, there are numerous opportunities for future work and enhancements to further elevate the capabilities of the system. By exploring additional integrations with IoT devices, enhancing security measures, optimizing power consumption, and leveraging AI/ML algorithms, we can continue to push the boundaries of smart home automation.

Furthermore, fostering documentation and community building efforts will empower users to fully leverage and customize the system to suit their specific needs and preferences. By creating a supportive ecosystem of users and developers, we can collectively drive innovation and unlock new possibilities in the realm of IoT and home automation.

In conclusion, this Arduino IoT project represents just the beginning of a journey towards creating smarter, more connected homes. By leveraging emerging technologies and embracing a spirit of continuous improvement, we can create truly transformative experiences that enhance convenience, comfort, and efficiency in our daily lives.

References

- https://www.youtube.com/watch?v=hZAX2d8qrbs

- https://www.youtube.com/watch?v=aYgBtyw2Bd0

- https://www.wikipedia.org/

Appendix