"PET FEEDER"

A IOT Report submitted in partial fulfilment of the requirement for the award of degree of

MASTER OF COMPUTER APPLICATIONS

Of

Visvesvaraya Technological University

BY

N DILIP KUMAR

(*1BO23MC027*)

Under the Guidance of

R Y Naidu

HOD-MCA

Department of Master Of Computer Applications

Brindavan College of Engineering

Dwarakanagar, Bagalur Main Road, Yelahanka,Bengaluru-560063

2024-2025

Brindavan College of Engineering

Dwarakanagar, Bagalur Main Road, Yelahanka,Bengaluru-560063

Department of Master Of Computer Applications

CERTIFICATE

This is to certify that *N DILIP KUMAR (1BO23MC027)* has completed his III Semester IOT Mini Project [22MCA32] titled "*Pet Feeder"* as a partial fulfilment for the award of Master of Computer Applications degree, during the academic year 2024-25 our supervision.

Guide **HOD-MCA** **Principal**

Examiners:

1.

2.

DECLARATION

I, N Dilip Kumar student of 3rd Semester MCA (VTU), Brindavan College of Engineering, Bangalore, bearing (USN No 1B023MC027) hereby declare that the project titled "Pet Feeder" has been carried out by me under the supervision of Internal project Guide Prof. R. Y Naidu, Professor, Dept. of MCA at Brindavan College of Engineering. and submitted in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications by Visvesvaraya Technological University during the academic year 2024- 2025. This report has not been submitted to any other Organization/University for any award of degree or certificate.

ACKNOWLEDGEMENT

I am thankful to the Principal of Brindavan College of Engineering- Dr. Bhagappa Sir, for extending his support and encouraging us to take up this challenging IOT Mini Project.

I extend my gratitude to the guide Prof. R Y Naidu, HOD of MCA Department, for his constant support and encouragement. Sir has been helping me out in every aspect of my project. I am grateful to you for all your help. I immensely thank my family for their continuous support. My parents and Sisters have been the backbone to me and without their suggestions, I would not have completed the project in the stipulated time. I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of MCA which helped us in successfully completing our project work.

N.Dilip Kumar

(1BO23MC027)

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

Automated pet feeder is one of the new technologies for feed pet. It will help pet owner to take care of their pet while they are not at home. Even the owners are not at home, they still can feed their pet. Automated pet feeder is built to help pet owner taking care of their pet. Automated pet feeder is one of the pet feeders that will be controlled by a wireless infra-red remote control. The automated pet feeder will be automatically dispenses predetermined amounts of food at the exact times user choose with controlled by a wireless infra red remote control. As pet lovers, user should understand those pets also need a proper diet management. Sometimes, the responsibilities of life inhibit pet owners from properly caring for their pets. Whether user away from home unexpectedly or simply would like one less chore to worry about, user can feel secure that the beloved pet will be cared for and fed on time, every time. Pet care should be fun, not burdensome and so the goal of this project is to assist owner with pet care by providing an automatic pet feeder. The purpose of the project helps the owner of the pet feeding their pet on time even when they are not at home. Other than that, it also can help the owner know the diet of their pet. Knowing the diet of the pet is very important for the owner to make sure that the pet is in good health. This system assist pet owner to feed the pet. The system act in two ways, one is feeding the pet and sends the feeding information to owner. After it feed the pet, the system will stop responding for certain time in order to make sure that
the pet do not eat too much.

1. RELATED WORK

Automated pet feeder is one of the new technologies for feed pet. It will help pet owner to take care of their pet while they are not at home. Even the owners are not at home, they still can feed their pet. Automated pet feeder is built to help pet owner taking care of their pet. Automated pet feeder is one of the pet feeders that will be controlled by a wireless infra-red remote control. The machine driven pet feeder will be automatically dispenses predetermined lots of food at the accurate times user choose with controlled by a wireless infra red remote control. As pet lovers, user should understand those pets also need a proper diet management. Sometimes, the responsibilities of life inhibit pet owners from properly caring for their pets. Whether user away from home unexpectedly or simply would like one less chore to worry about, user can feel secure that the beloved pet will be cared for and fed on time, every time.

1.2 OBJECTIVES

1. More personalized experience of keeping pets.
2. No longer need to worry about their pets during business trips or vacations.
3. No longer need to purchase multiple feeder for multiple pets.
4. No longer need to worry about the cross--eating between different pets.

1.3 PROBLEM STATEMENT

- Pets require special diet and lot of time to take care. There is no such type of application to feed pets on time. So, we came with an project like mobile application pet feeder which is cost friendly for the users

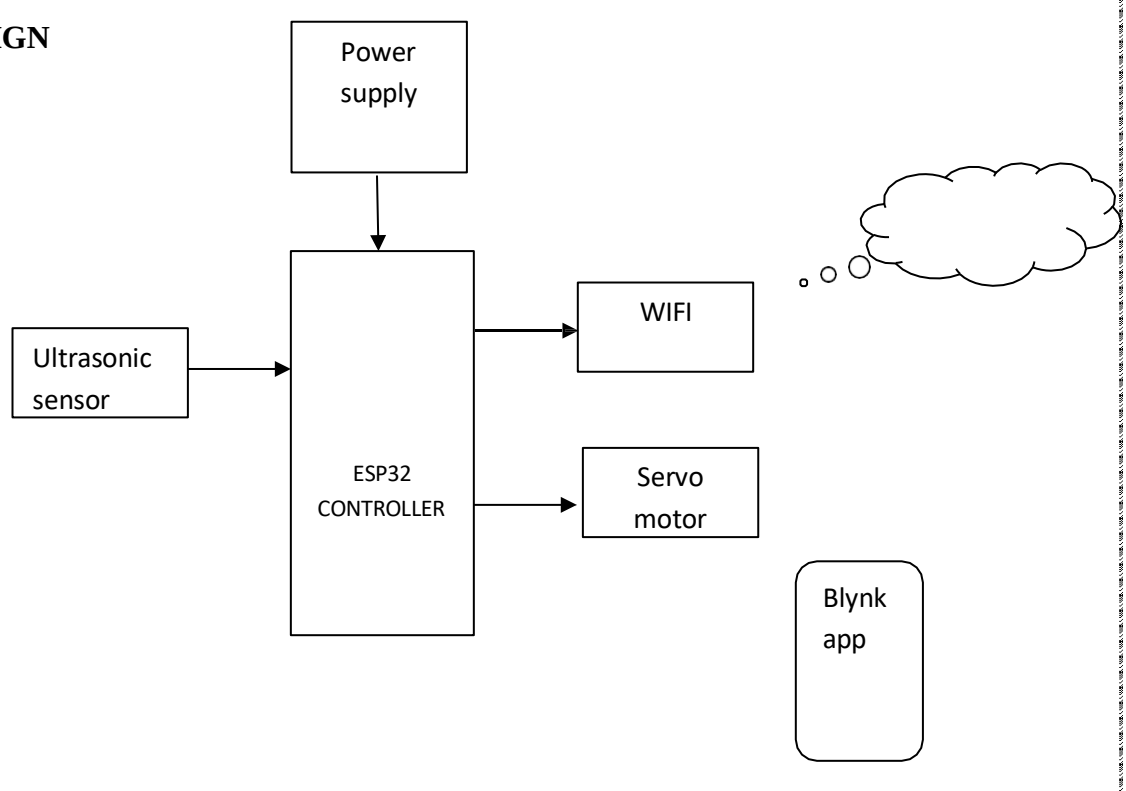
CHAPTER 2

SYSTEM DESIGN & PROPOSED SYSTEM

2.1 PROPOSED SYSTEM

- There are two methods : 1. smart mode
  2. manual mode
- Smart mode : owner can feed the pet by using application without wasting his/her time
- Manual mode : owner can feed the feed pet directly when he/she is free

2.2 SYSTEM DESIGN



CHAPTER-3

HARDWARE AND SOFTWARE REQUIREMENTS

3.1 Hardware components

1. ESP32 module
2. Servo motor
3. power supply
4. Ultrasonic sensor
5. Jumper wires

3.2 Software requirements

1. Arduino IDE
2. Embedded C
3. Blynk App

CHAPTER 4

COMPONENTS DESCRIPTION

- **ESP32 CONTROLLER MODULE**

ESP32 is a system on a chip that integrates the following features:

- Wi-Fi (2.4 GHz band)
- Bluetooth
- Dual high performance Xtensa® 32-bit LX6 CPU cores
- Ultra Low Power co-processor
- Multiple peripherals

Powered by 40 nm technology, ESP32 provides a robust, highly integrated platform, which helps meet the continuous demands for efficient power usage, compact design, security, high performance, and reliability.

Espressif provides basic hardware and software resources to help application developers realize their ideas using the ESP32 series hardware. The software development framework by Espressif is intended for development of Internet-of-Things (IoT) applications with Wi-Fi, Bluetooth, power management and several other system features.

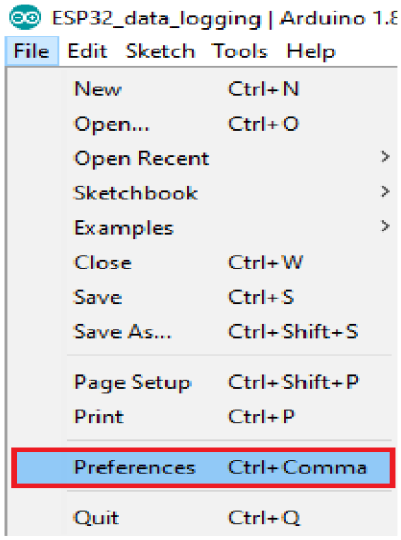
Hardware:

- An **ESP32** board
- **USB cable** - USB A / micro USB B
- **Computer** running Windows, Linux, or macOS

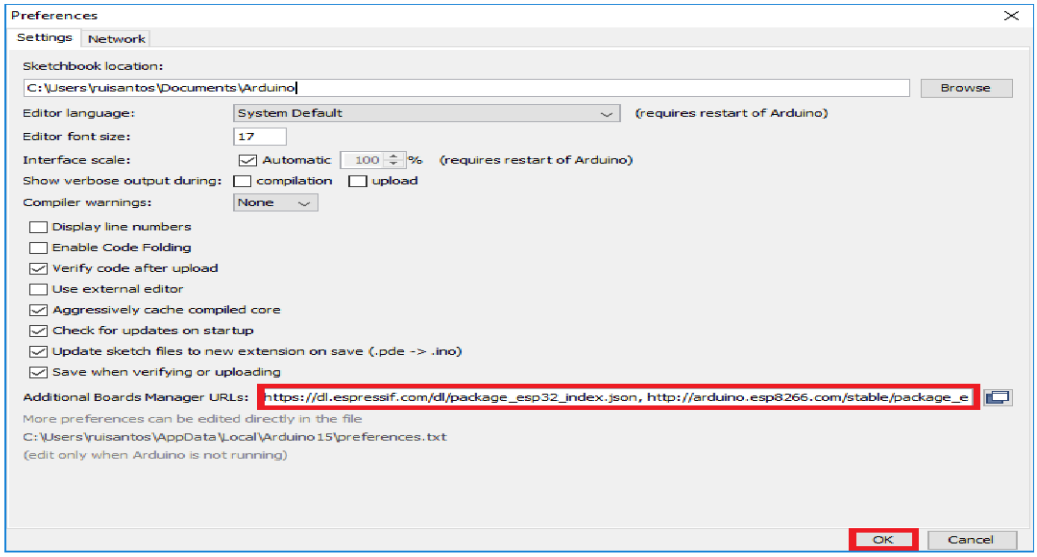
Installing ESP32 Add-on in Arduino IDE

To install the ESP32 board in your Arduino IDE, follow these next instructions:

1. In your Arduino IDE, go to **File**> **Preferences**



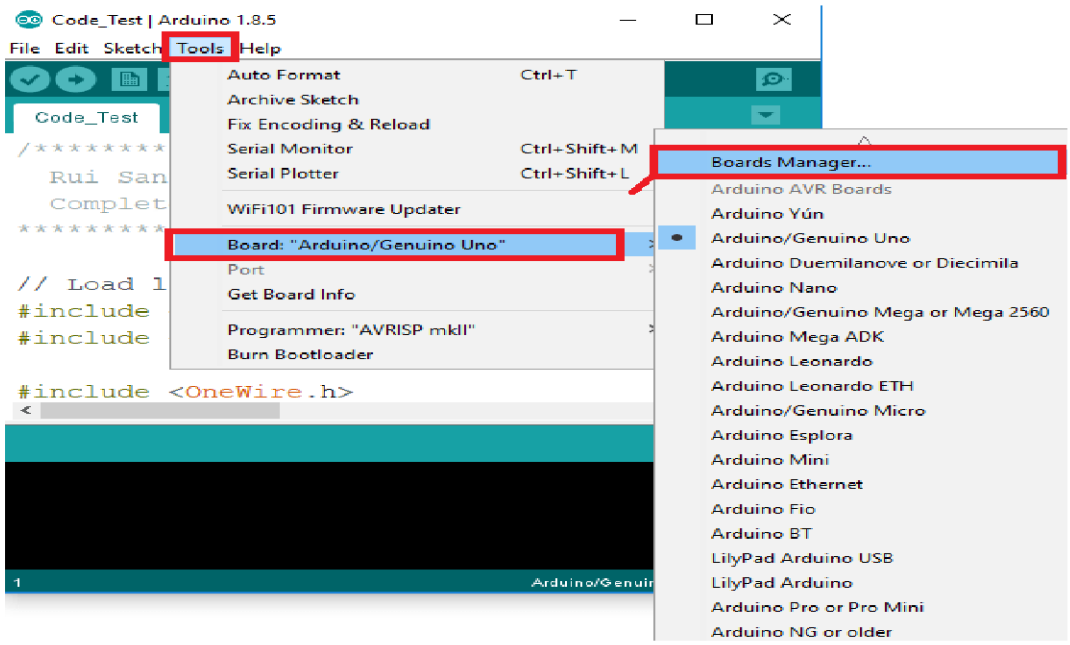
2. Enter **https://dl.espressif.com/dl/package_esp32_index.json** into the "Additional Board Manager URLs" field as shown in the figure below. Then, click the "OK" button:



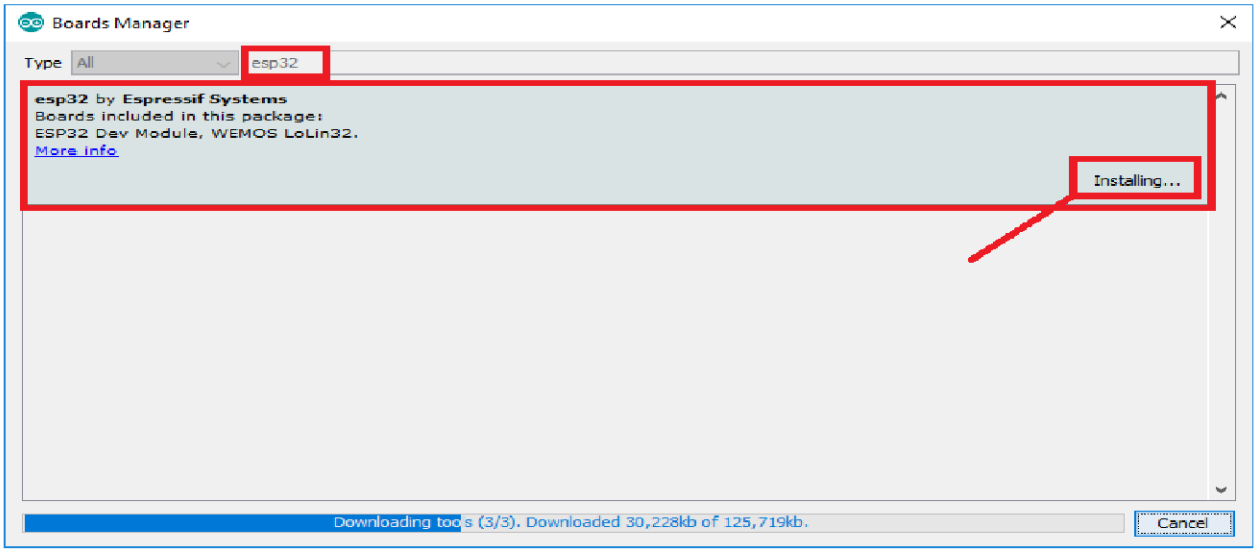
Note: if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:

https://dl.espressif.com/dl/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

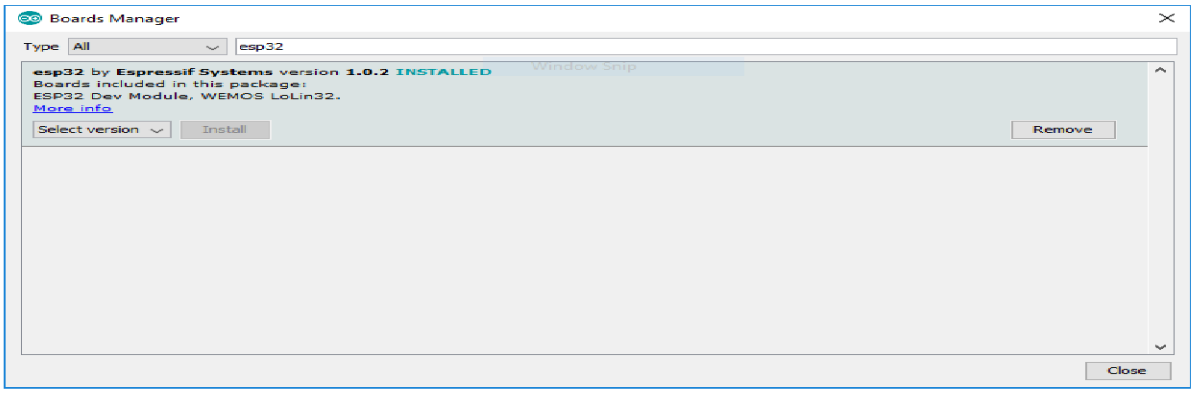
3. Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager**



4. Search for **ESP32** and press install button for the "**ESP32 by Espressif Systems**



5. That’s it. It should be installed after a few seconds

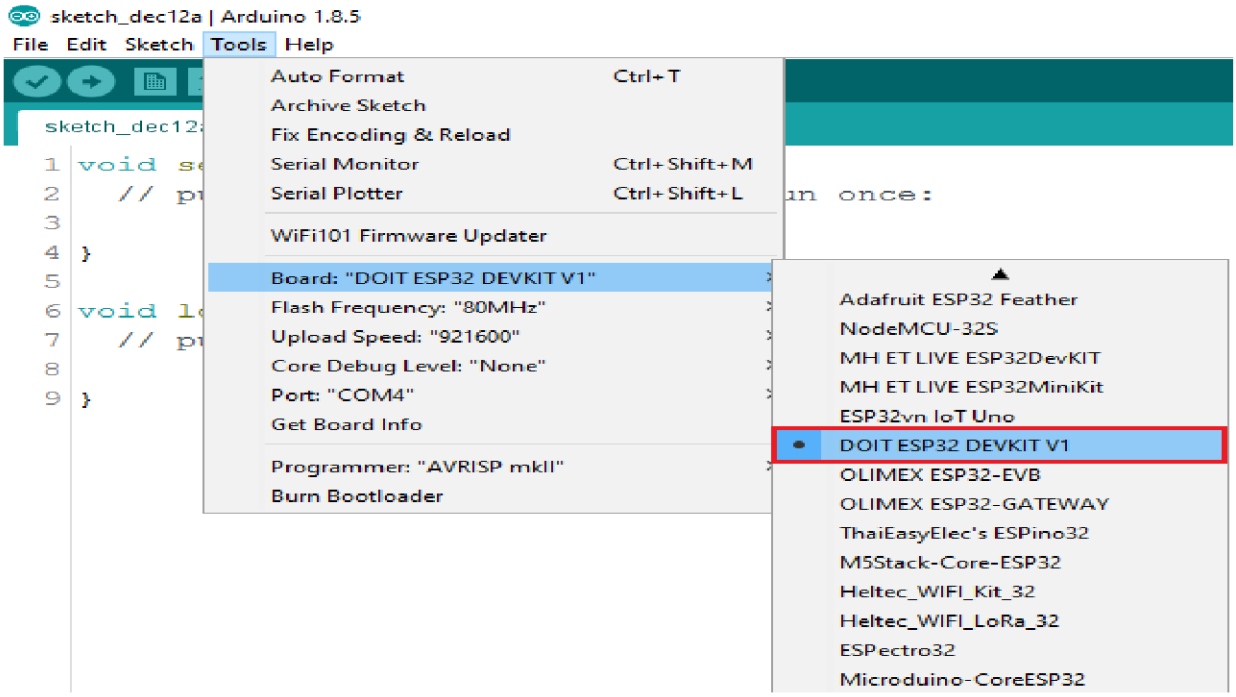


CHAPTER 5

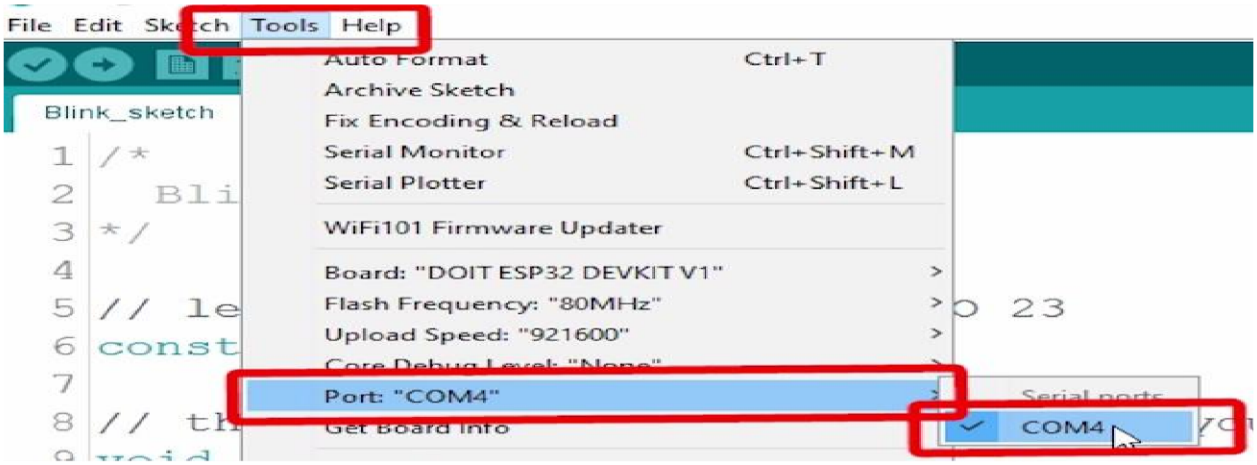
IMPLEMENTATION & TESTING THE INSTALLATION

Plug the ESP32 board to your computer. With your Arduino IDE open, follow these steps:

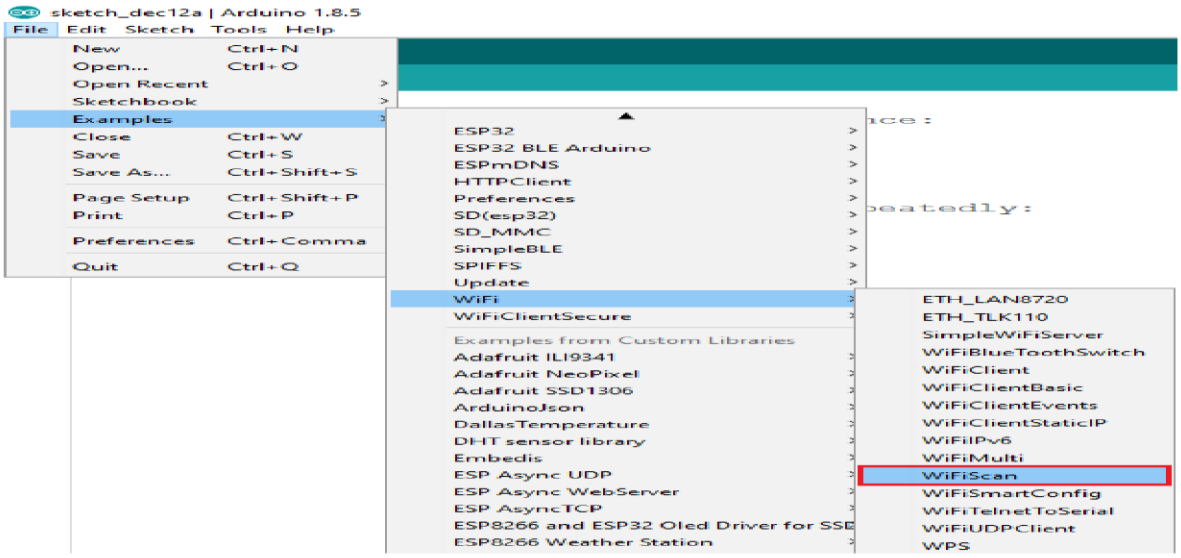
1. Select your Board in **Tools** > **Board** menu (in my case it's the **DOIT ESP32 DEVKIT V1**)



2. Select the Port (if you don't see the COM Port in your Arduino IDE, you need to install the CP210x USB to UART Bridge VCP Drivers):



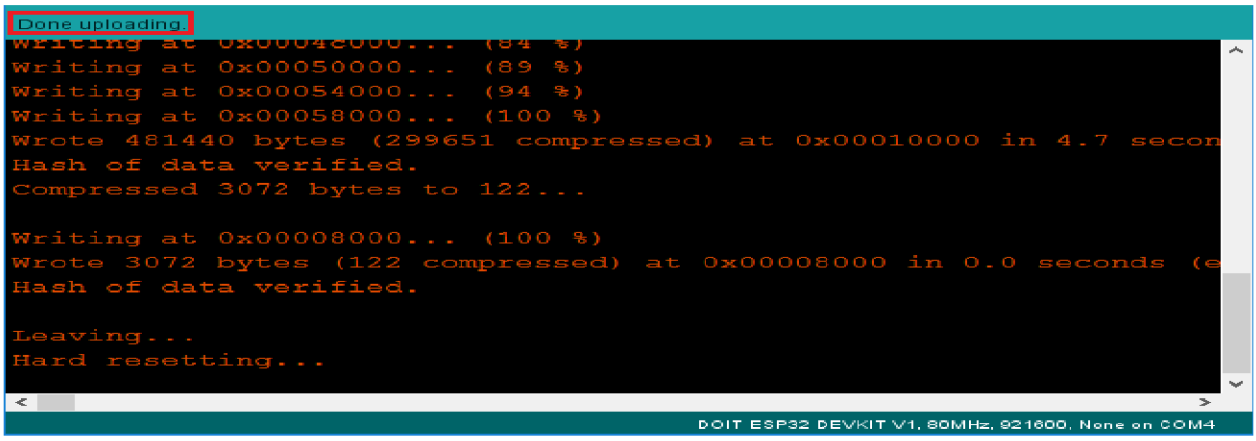
3. Open the following example under **File** > **Examples** > **WiFi (ESP32)** > **WiFiScan**



4. A new sketch opens in your Arduino IDE:

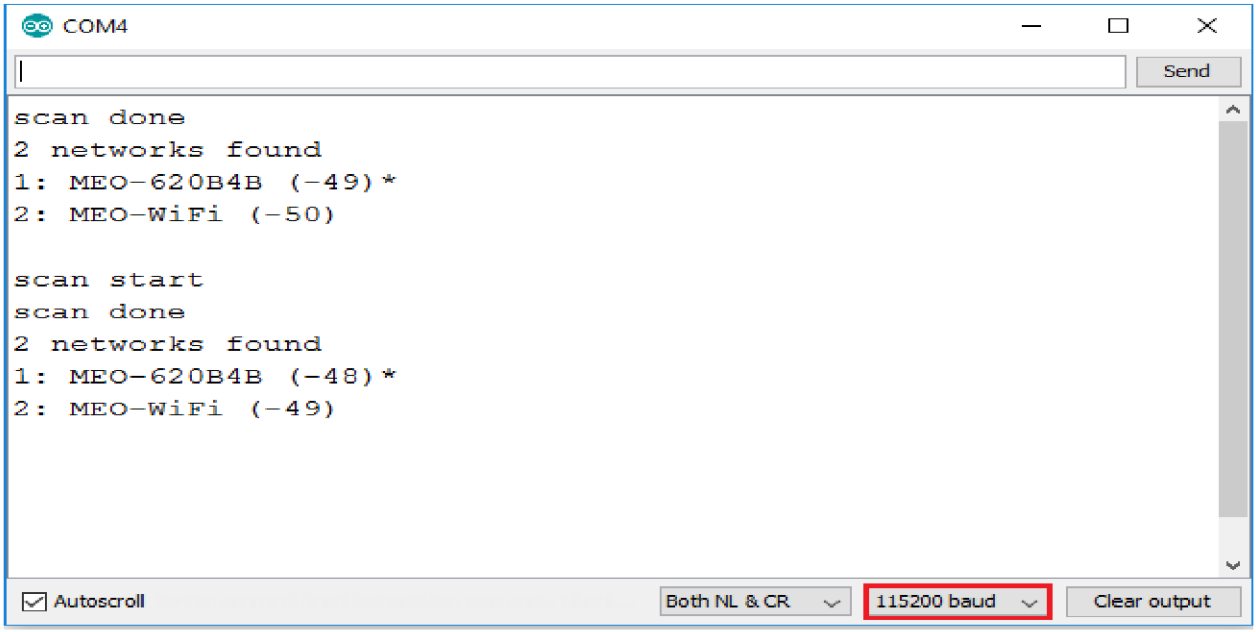
5. Press the **Upload** button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.

6. If everything went as expected, you should see a "**Done uploading.**" message.



7. Open the Arduino IDE Serial Monitor at a baud rate of 115200:

8. Press the ESP32 on-board **Enable** button and you should see the networks available near your ESP32:

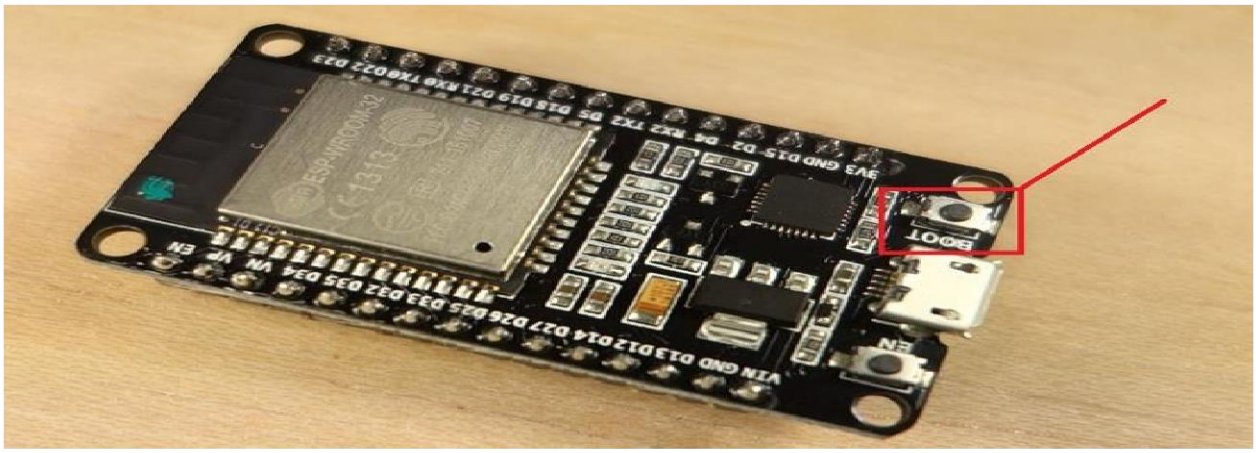


Troubleshooting

If you try to upload a new sketch to your ESP32 and you get this error message "*A fatal error occurred: Failed to connect to ESP32: Timed out… Connecting…*". It means that your ESP32 is not in flashing/uploading mode.

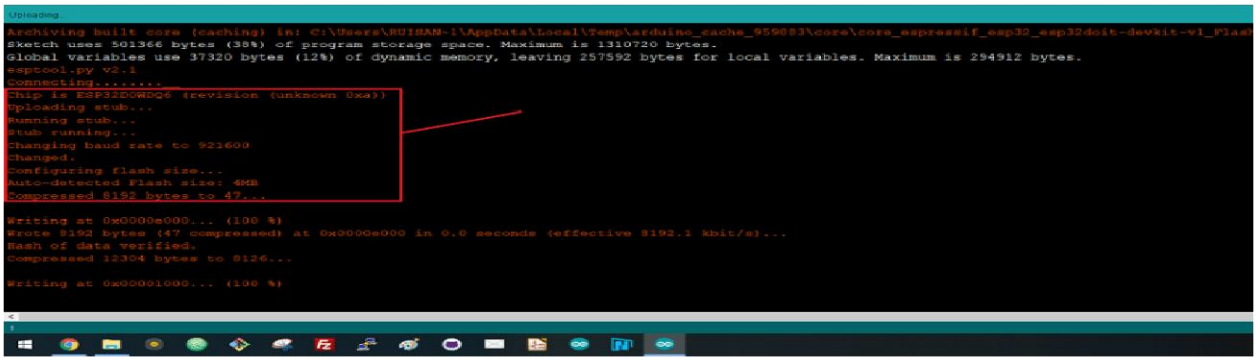
Having the right board name and COM por selected, follow these steps:

- Hold-down the "**BOOT**" button in your ESP32 board



- Press the "**Upload**" button in the Arduino IDE to upload your sketch:

- After you see the "**Connecting….**" message in your Arduino IDE, release the finger from the "**BOOT**" button:



- After that, you should see the "**Done uploading**" message

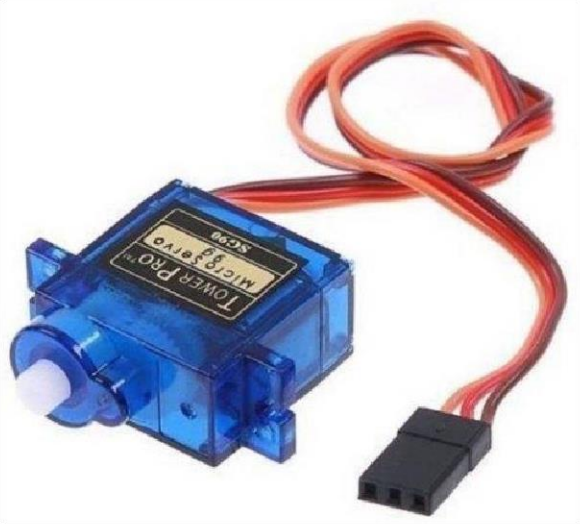
That's it. Your ESP32 should have the new sketch running. Press the "**ENABLE**" button to restart the ESP32 and run the new uploaded sketch

Servo motor

This TowerPro SG90 Continuous Rotation 360 Degree Servo Motor is special among all the available servo motors because its operation is very different from that of a standard servo. As instead of going to a specified angle, this servo will be static at a 1.5ms pulse, a longer pulse gives forward rotation and a shorter pulse give backward rotation. It equips Carbon Fiber Gears which makes the servo motor much lighter than same metal gear motor**.**

For small load applications using the metal gear servo motor adds on unnecessary weight, so we suggest using this lightweight plastic gear servo motors. It is a Digital Servo Motor which receives and processes PWM signal faster and better. It equips sophisticated internal circuitry that provides good torque, holding power, and faster updates in response to external forces. The good optimized performance and reliability of our servos have made them the favorite choice of many RC hobbyists.

- Structural Material: copper metal teeth, coreless motor, double ball bearing
- The attachment contains: helm, the line is long 30CM, fixing screws, damping pouches and aluminium sets and other Accessories (see really making plans)



Interfacing Servo with Arduino

Step - 1

If you already know the installation of Arduino IDE or have the Arduino IDE, please skip this step.

Download the leatest Arduino IDE Software.

After the installation of Arduino IDE open it.

Step -2

In this project we need to use a library named "Servo.h". This is a inbuilt library with Arduino IDE.

There is two way to include a library

Go to Sketch>Include library>Servo

OR

Just type #include<Servo.h>

```
#include<Servo.h>
```

Step -3

Next we need to declare a integer type variable. Now I am using the variable named as "pos". This variable can only hold integer values.

Step - 4

In the declaration part, we need to declare another variable. Here wed using the variable name as "Myservo". You can also use any other names. This variable is used to connect the corresponding servo to the entire code. Here we use the keyword "Servo". The syntax is "Servo variable_name". The code is look like,

```
Servo Myservo;
```

Step - 5

Every Arduino program is consist of a "void setup()" function and "void loop()" function. When we turn on the Arduino the functions/statements in "void setup()" will work first and it will work once. After that the functions/statements in void loop() will work like a loop.

We need to set the signal pin of servo motor to Arduino pin In the "void setup()". In the clear voice, Here we are declaring which pin of Arduino to control the servo motor. Here we are using Digital Pin 3 of Arduino to control the servo motor.

Here we using the keyword attach(). The syntax is "variable_name.attach(pin_number)". The code is look like,

```
Myservo.attach(3);
```

Step - 6

Let discuss about the servo control statement. The keyword is "write()". and the syntax is "variable_name.write(pos)". We can either use a variable or number which is in between 0 and 360 in write function.

```
Myservo.write(pos);
```

Step - 7

Now we need set a loop function. Here we are using "for()" loop to control the servo motor. In for() loop, set the minimum value is 0. ie. (.pos=0) and maximum value is 180. ie. (pos<=180). Then incriment the pos ie.(pos++). Then implement the write function(function in Step-6) in the for loop. Then add a delay function. This delay function help to waits 15ms for the servo to reach the position.

```
for(pos=0;pos<=180;pos++){
Myservo.write(pos);
delay(15);
}
```

Step - 8

Then add another delay function after the for() loop. This delay function for wait a corresponding time(which is included in delay function) after above for loop.

```
delay(1000);
```

Step - 9

Now we need set another loop function. Here we are using " for()" loop to control the servo motor. In for() loop, set the minimum value is 180. ie. (.pos=180) and maximum value is 0. ie. (pos>=0). Then decriment the pos ie.(pos--). Then implement the write function(function in Step-6) in the for loop. Then add a delay function. This delay function help to waits 15ms for the servo to reach the position.

```
for(pos=180;pos>=0;pos--){
Myservo.write(pos);
delay(15);
}
```

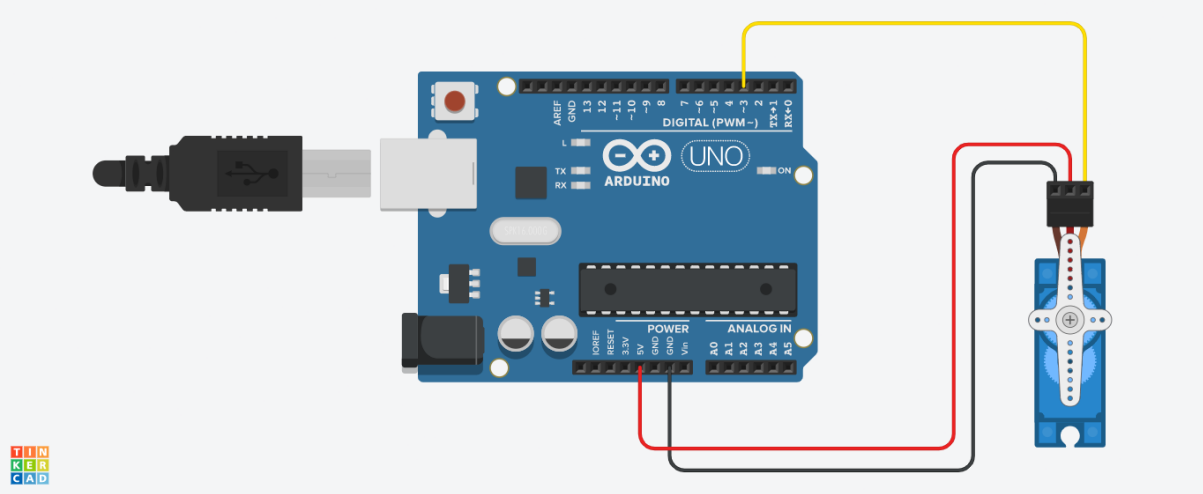
Step - 10

Then add another delay function after the for() loop. This delay function for wait a correspond ing time(which is included in delay function) after above for loop.

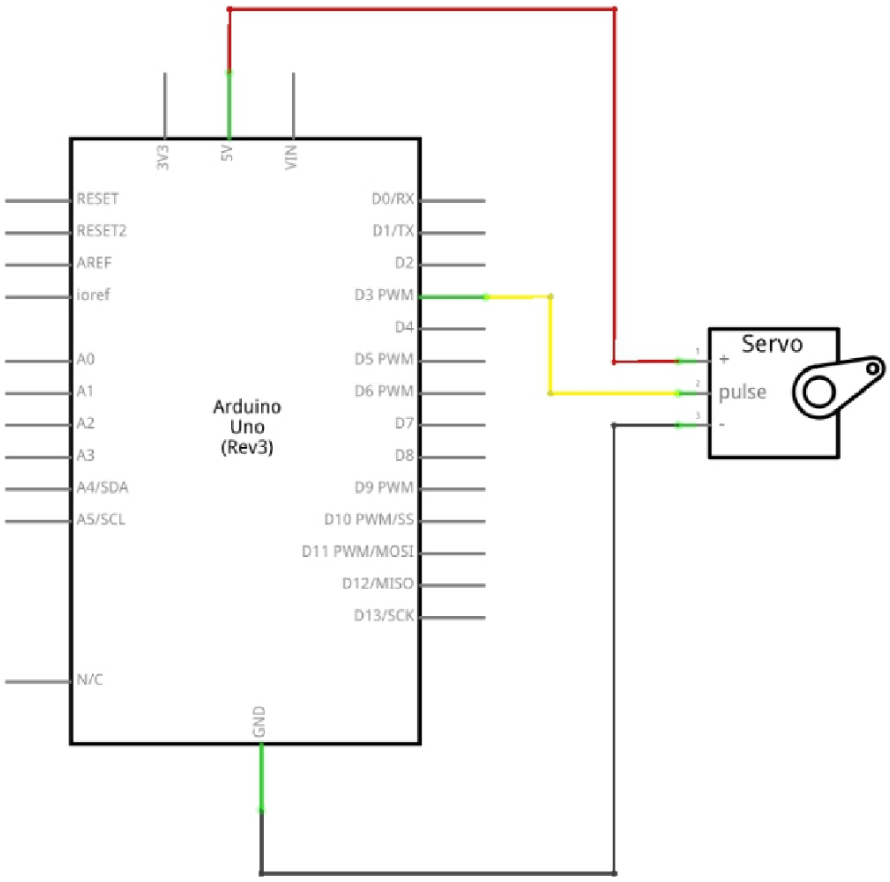
```
delay(1000);
```

The code is completed. The complete code is given in the software part.

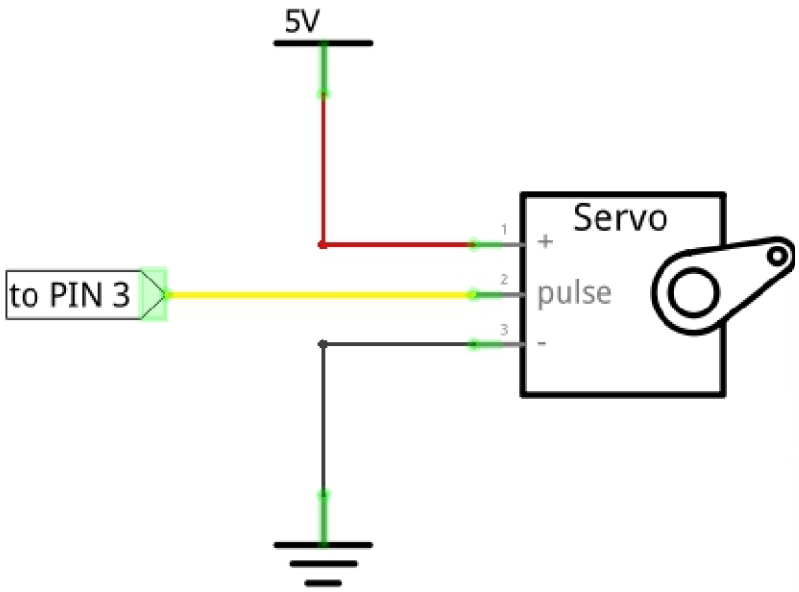
Type the whole code and double check for errors. Then Upload (The arrow button on top) the code to Arduino Board.


DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

Schematic


3V3
5V
VIN
RESET
RESET2
AREF
Ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
Arduino
Uno
(Rev3)
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
Servo
+
pulse
-

Circuit Diagram



➢ Implementation:

Sample Code for Servo to Arduino

```
#include<Servo.h>
Servo  Myservo;
int pos;
void setup(){
Myservo.attach(3);
}
void loop(){
for(pos=0;pos<=180;pos++){
Myservo.write(pos);
delay(15);
}
  delay(1000);
for(pos=180;pos>=0;pos--){
Myservo.write(pos);
delay(15);
}
  delay(1000);
}
```

Ultrasonic sensor

In this project we are using Ultrasonic ranging module HC-SR04 so that blind person can feel safe while walking on the road. If any objects arrived with in a range of 2cm to 400cm then then stick gives him a response by vibrating and also with a buzzer sound so he can aware of that object. We added pair of ultrasonic ranging modules so the stick can detect the lower objects small stones and higher objects too. The ranging module is shown in the below Figure 3.5

Working principle:

As shown in the above Figure 3.3 is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple formula that is,

$$\textit{Distance} = \textit{Speed} \times \textit{Time}$$

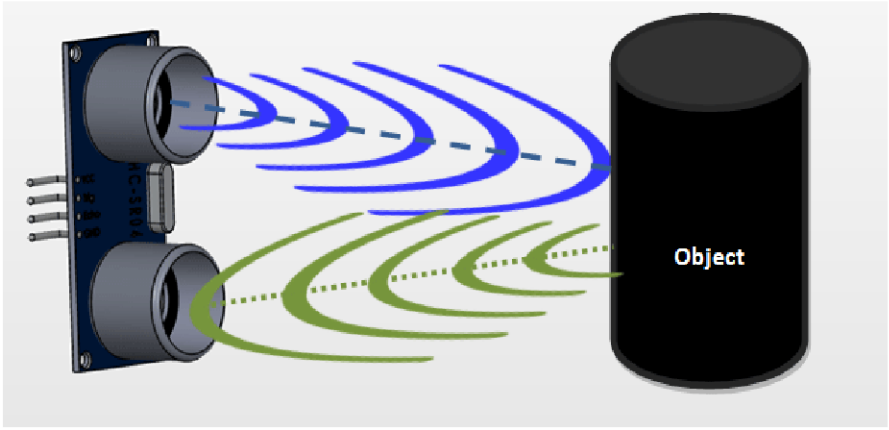
The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is

observed by the Ultrasonic receiver module as shown in the below figure 3.6. The working principle of a ultrasonic sensor is as below:

- Send at least 10us of HIGH signal to trigger pin.
- Detect the pulse width of echo signal sent back.

Working Principle

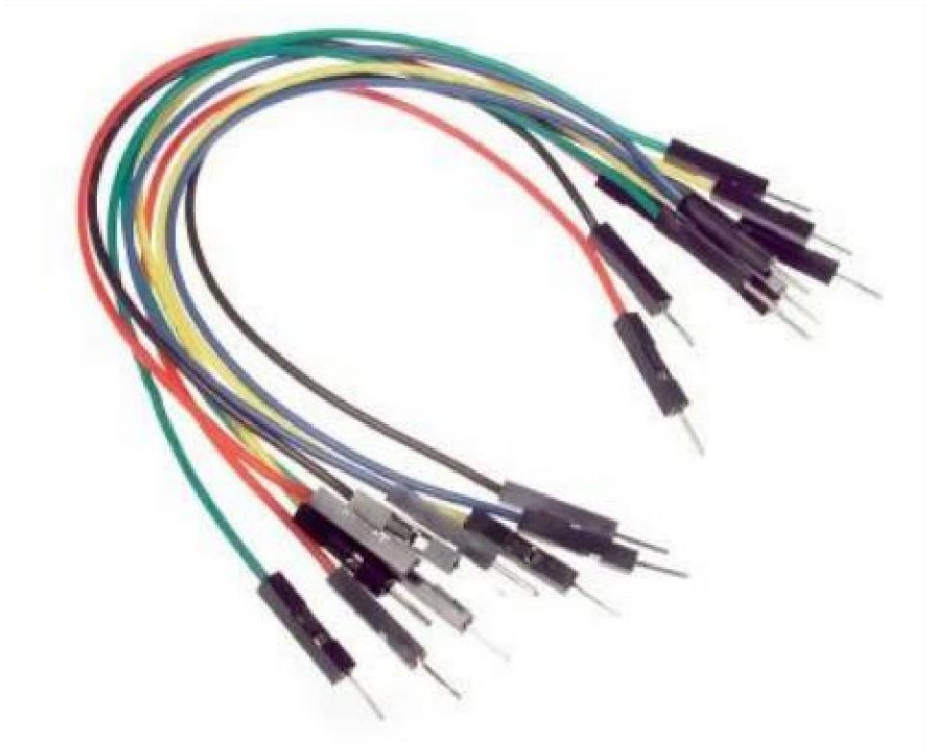
To calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of ultrasonic wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the ultrasonic wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance



using a microcontroller or microprocessor

Jumper Wires

used for making connections between items on your breadboard and your Arduino's header pins. Use them to wire up all your circuits. Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires. Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often.



SOFTWARE REQUIREMENTS

Embedded C

Embedded C is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems. Embedded C programming typically requires nonstandard extensions to the C language in order to support enhanced microprocessor features such as fixed-point arithmetic, multiple distinct memory, banks, and basic I/O operations. Embedded C uses most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc. The embedded c programming language is used in the microcontrollers. The embedded c language is a general-purpose programming language that provides code efficiency, elements of structured programming and a rich set of operators. Embedded c is not a big language and is not designed for any one particular area of application. It's generally combined with its absence of restriction, makes embedded c a convenient and effective programming solution for a wide variety of software tasks. Many applications can be solved more easily and efficiently with embedded c than with other more specialized languages. The embedded c language on its own is not capable of performing operations (such as input and output) that would normally require intervention from the operating system. Instead, these capabilities are provided as a part of standard library. Because these functions are separated from the language itself, embedded c is especially suited for producing code that is portable across wide platforms.

Arduino IDE(Compiler):

The Arduino Compiler 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.ItconnectstotheArduinoandGenuinehardwaretouploadprogramsandcommunicate with them. Arduino is an open-source electronics platform based on easy-to-use hardw

BLYNK APP

Blynk is a toolset for all makers, badass inventors, designers, teachers, nerds and geeks who would love to use their smartphones to control electronics like Arduino, RaspberryPi and similar ones. We've done all the hard work of establishing internet connection, building an app and writing hardware code.

With Blynk, you simply snap together an amazing interface from various widgets we provide, upload the example code to your hardware and enjoy seeing first results in under 5 minutes! It works perfectly for newbie makers and saves tons of time for evil geniuses.

Blynk will work with all popular boards and shields. We wanted to give you full freedom when deciding how to plug Blynk into your existing or new project.You will also enjoy the convenience of Blynk Cloud. Which is, by the way is free and open-source.

Imagine a prototyping board on your smartphone where you drag and drop buttons, sliders, displays, graphs and other functional widgets. And in a matter of minutes these widgets can control Arduino and get data from it.

Blynk is not an app that works only with a particular shield. Instead, it's been designed to support the boards and shields you are already using. And it works on iOs and Android.

UPD: Blynk also works over USB. This means you can tinker with the app by connecting it to your laptop or desktop while waiting for some internet shield to arrive.

Blynk works over the Internet. So the one and only requirement is that your hardware can talk to the Internet.

No matter what type of connection you choose - Ethernet, Wi-Fi or maybe this new ESP8266 everyone is talking about – Blynk libraries and example sketches will get you online, connect to Blynk Server and pair up with your smartphone.

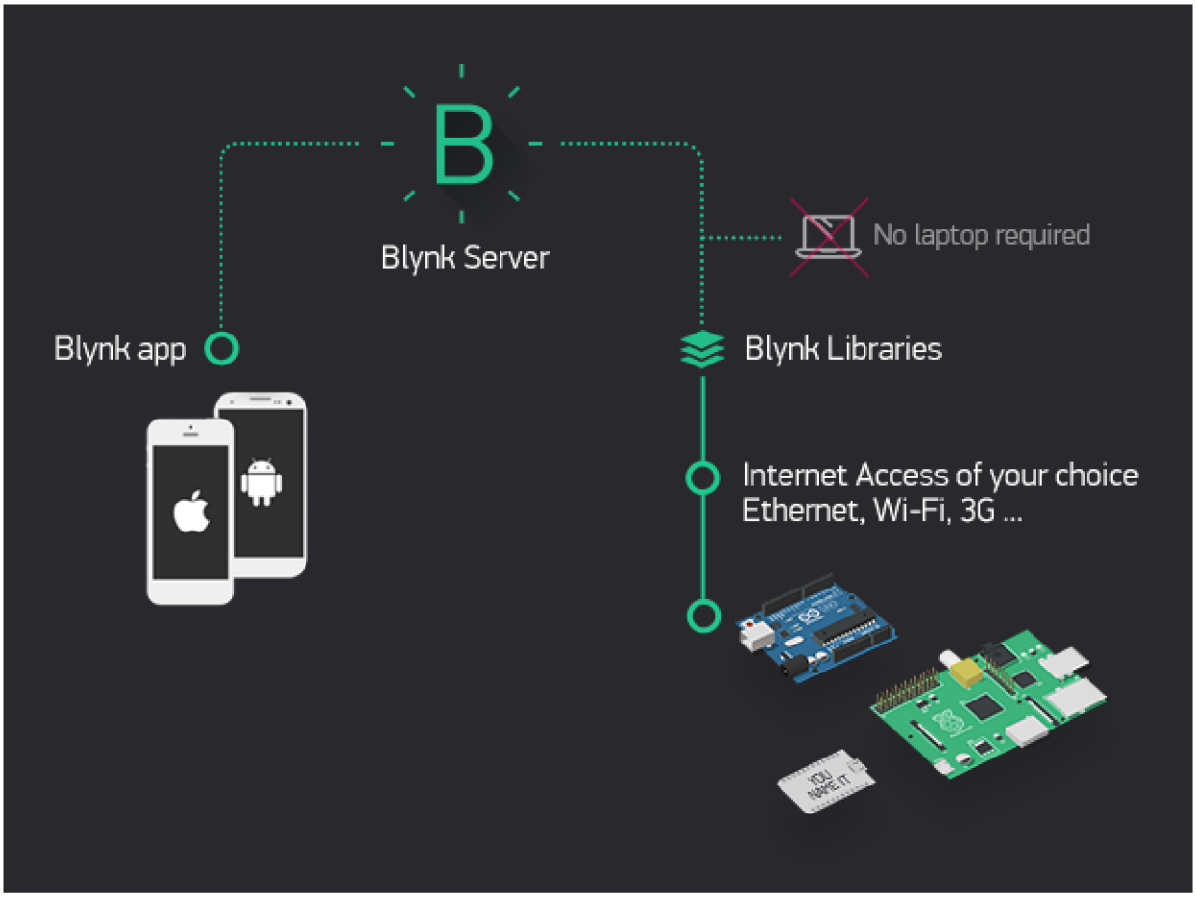


FIG: Blynk architecture

Currently, Blynk libraries work with:

- USB
- Ethernet shield
- WiFi shield
- Arduino with Ethernet
- Arduino YÚN (testing in progress)
- ESP8266
- Raspberry Pi (Blynk will communicate with Pi's GPIOs)
- more Arduino compatible shields and boards (this list will be updated as we test the compatibility)

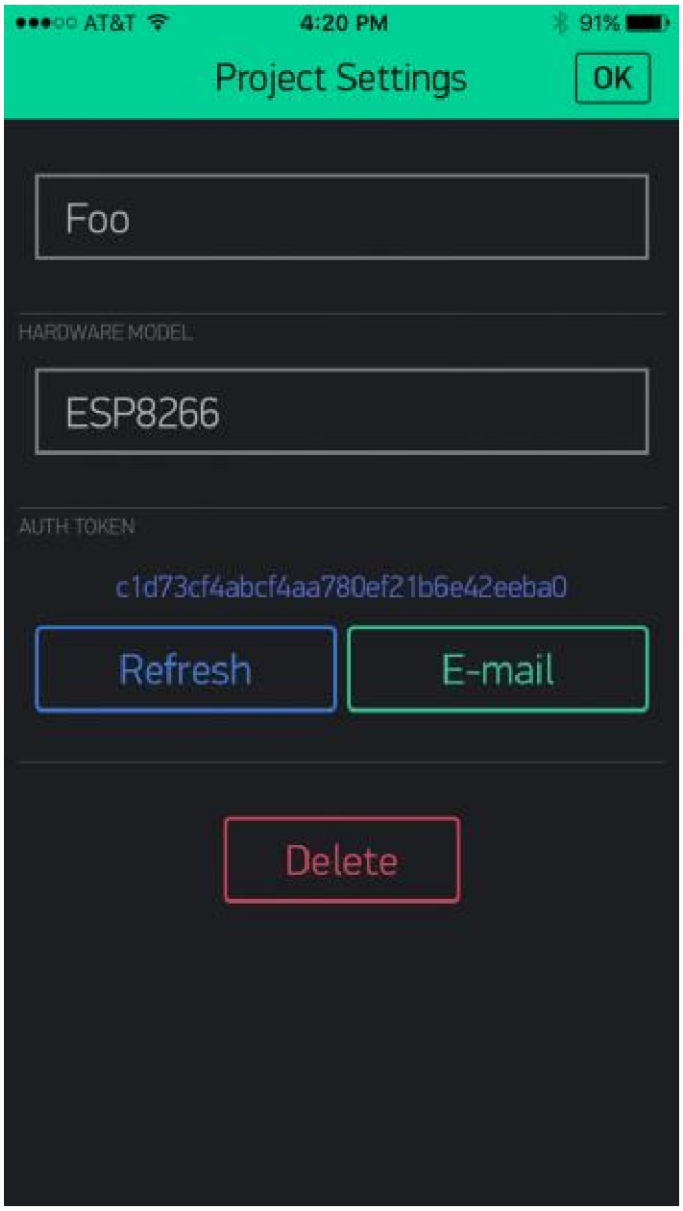
It's not that easy to take Arduino out of your home network, so we've built a Blynk server. It handles all the authentication and communication, and also keeps an eye on your board while the smartphone is offline. Blynk server runs on Java and is open-source. You will be able to run it locally if you really need to. Messaging between mobile apps , Blynk Server and Arduino is based on a simple, lightweight and fast binary protocol over TCP/IP sockets. **CREAING A PROJECT IN BLYNK AP**

After downloading the app, create an account and log in.

You'll also need to install the **Blynk Arduino Library**, which helps generate the firmware running on your ESP8266. Download the latest release from Blynk's GitHub repo, and follow along with the directions there to install the required libraries.

Create a Blynk Project

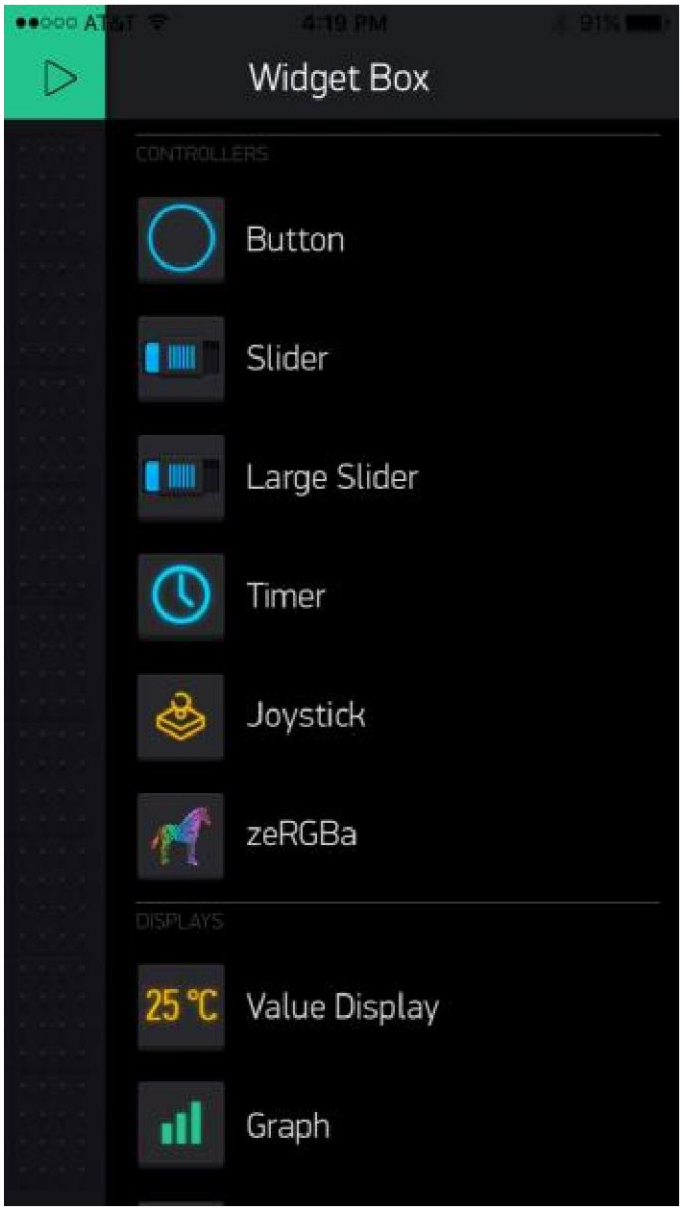
Next, click the "Create New Project" in the app to create a new Blynk app. Give it any name you please, just make sure the "Hardware Model" is set to **ESP8266**.



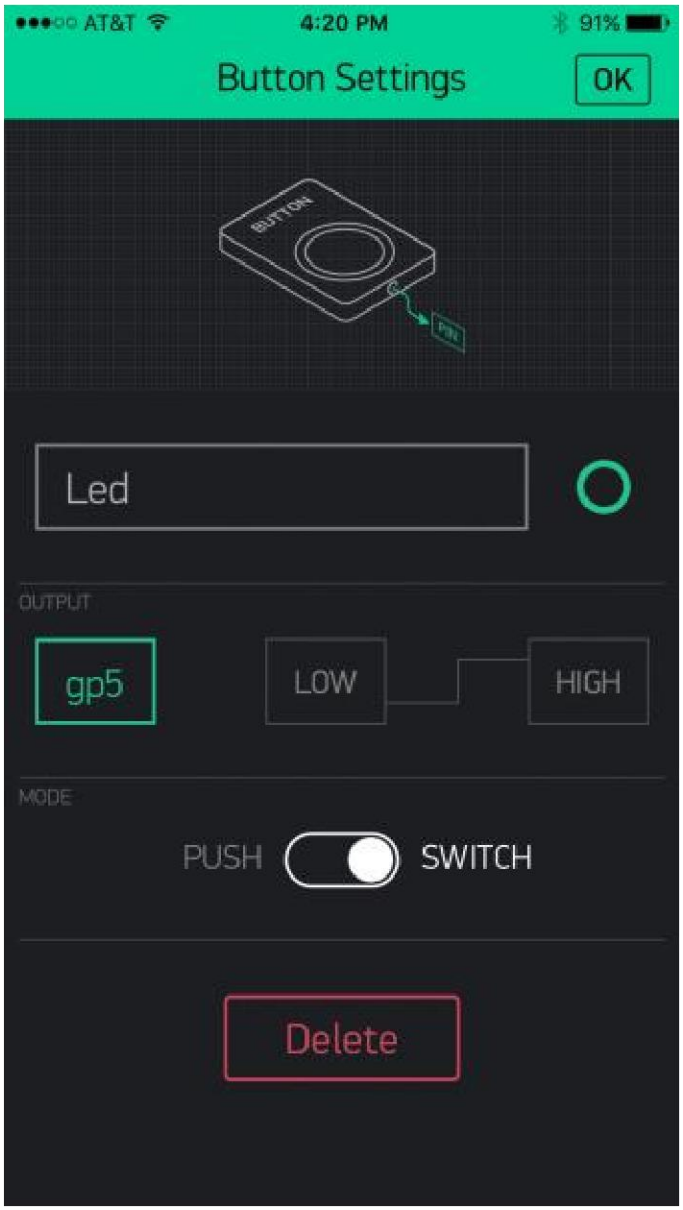
The **Auth Token** is very important – you'll need to stick it into your ESP8266's firmware. For now, copy it down or use the "E-mail" button to send it to yourself.

Add Widgets to the Project

Then you'll be presented with a blank new project. To open the widget box, click in the project window to open.

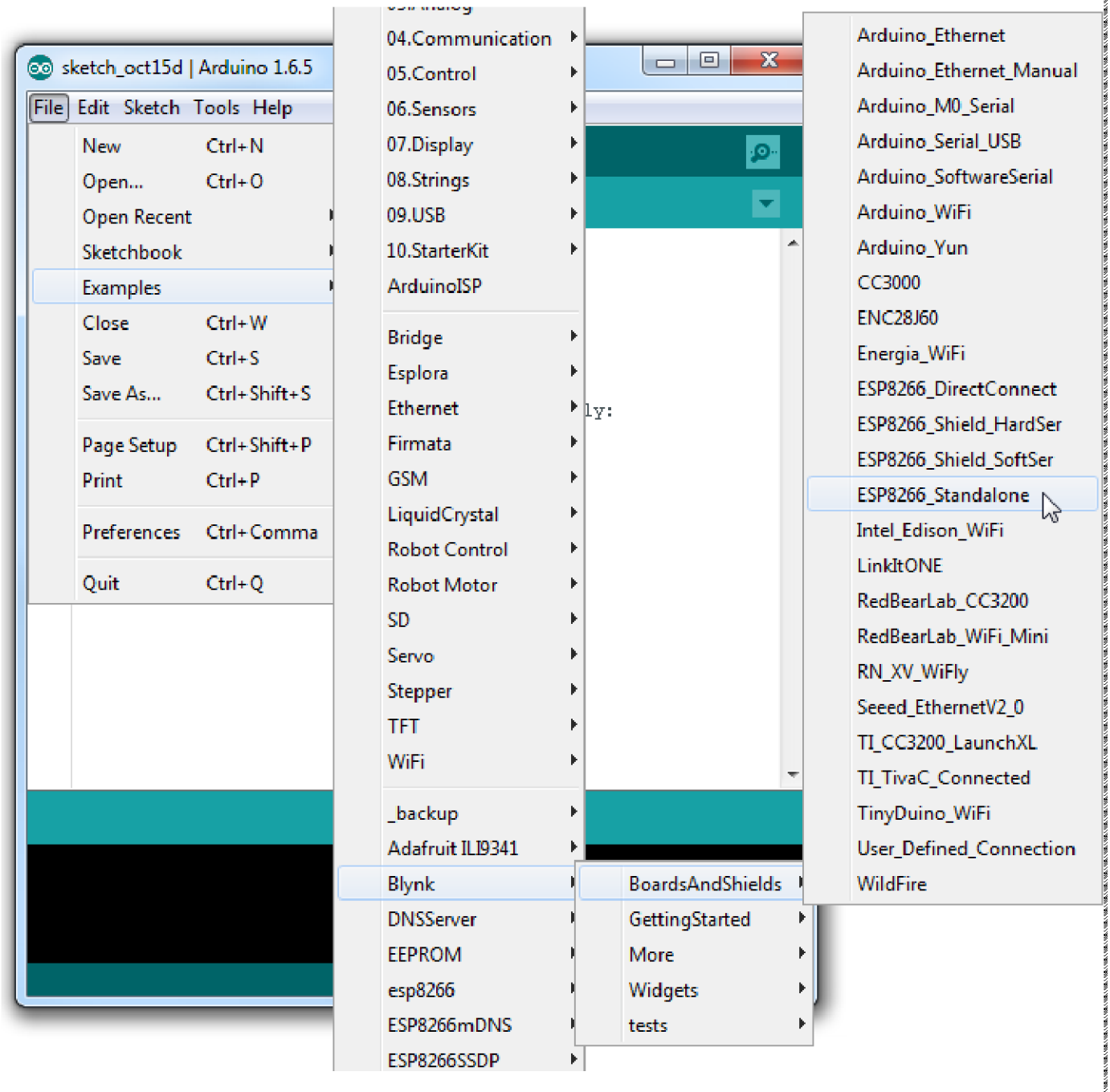


Add a **Button**, then click on it to change its settings. Buttons can toggle outputs on the ESP8266. Set the button's output to **gp5**, which is tied to an LED on the Thing Dev Board. You may also want to change the action to "Switch."



Upload the Blynk Firmware

Now that your Blynk project is set up, open Arduino and navigate to the **ESP8266_Standalone** example in the **File**>**Examples**>**Blynk**>**BoardsAndShields** menu.

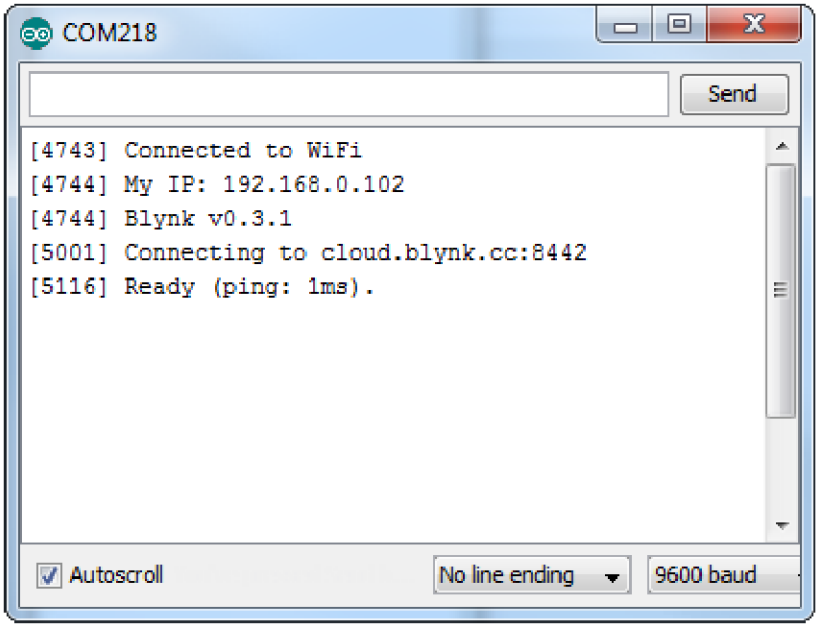


Before uploading, make sure to paste your **authoriazation token** into the auth[] variable. Also make sure to **load your WiFi network settings into the Blynk.begin(auth, "ssid", "pass")** function.

Then upload!

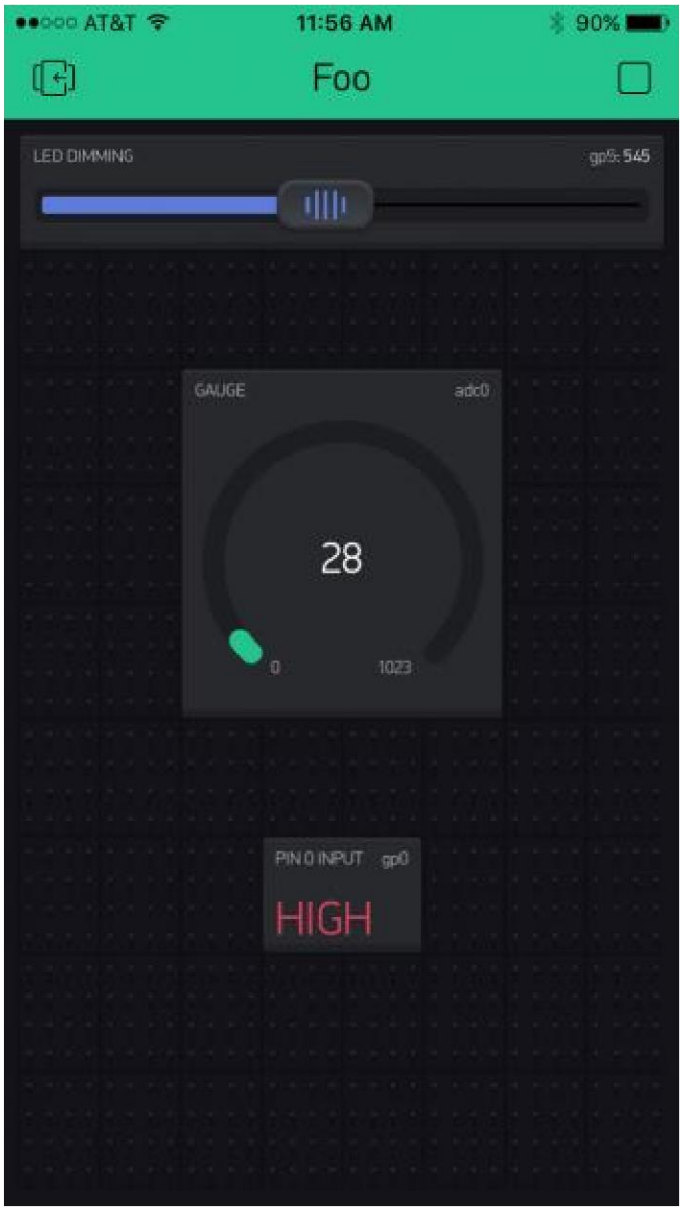
Run the Project

After the app has uploaded, open the serial monitor, setting the baud rate to 9600. Wait for the "Ready (ping: xms)." message.



Then click the "Run" button in the top right corner of the Blynk app. Press the button and watch the LED!

Then add more widgets to the project. They should immediately work on the ESP8266 without uploading any new firmware.



You can add analog output sliders, digital input monitors, and analog input gauges.

CLOUD

Cloud Computing provides us means of accessing the applications as utilities over the Internet. It allows us to create, configure, and customize the applications online.

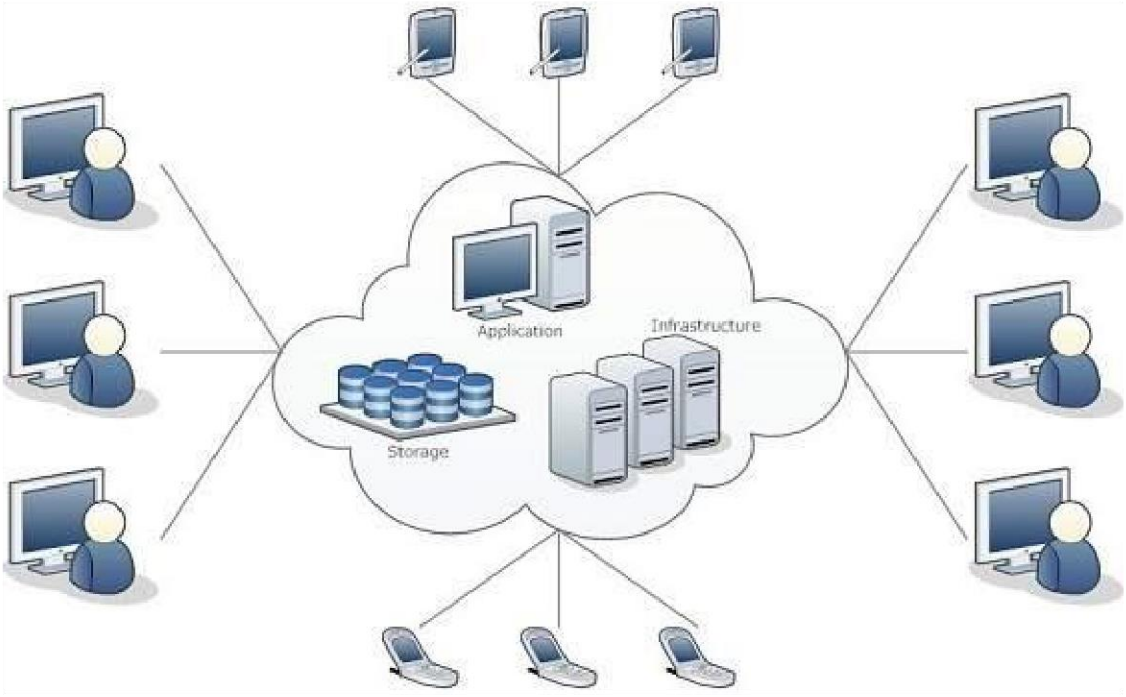
What is Cloud?

The term **Cloud** refers to a **Network** or **Internet.** In other words, we can say that Cloud is something, which is present at remote location. Cloud can provide services over public and private networks, i.e., WAN, LAN or VPN.

Applications such as e-mail, web conferencing, customer relationship management (CRM) execute on cloud.

What is Cloud Computing?

Cloud Computing refers to **manipulating, configuring,** and **accessing** the hardware and software resources remotely. It offers online data storage, infrastructure, and application.



Cloud computing offers **platform independency,** as the software is not required to be installed locally on the PC. Hence, the Cloud Computing is making our business applications **mobile** and **collaborative.**

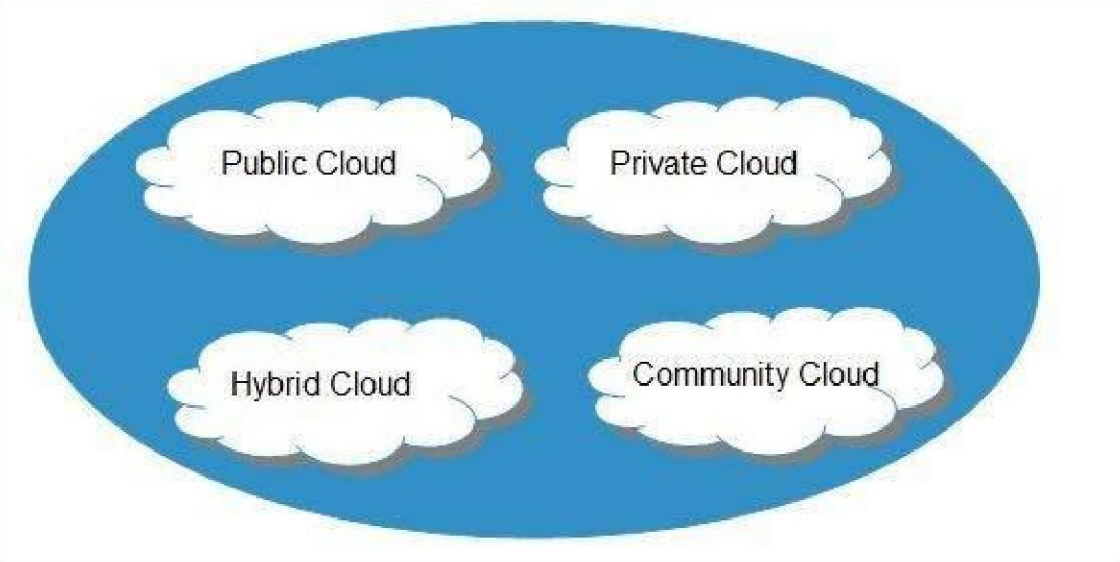
Basic Concepts

There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users. Following are the working models for cloud computing:

- Deployment Models
- Service Models

Deployment Models

Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access: Public, Private, Hybrid, and Community.



Public Cloud

The **public cloud** allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness.

Private Cloud

The **private cloud** allows systems and services to be accessible within an organization. It is more secured because of its private nature.

Community Cloud

The **community cloud** allows systems and services to be accessible by a group of organizations.

Hybrid Cloud

The **hybrid cloud** is a mixture of public and private cloud, in which the critical activities are performed using private cloud while the non-critical activities are performed using public cloud.

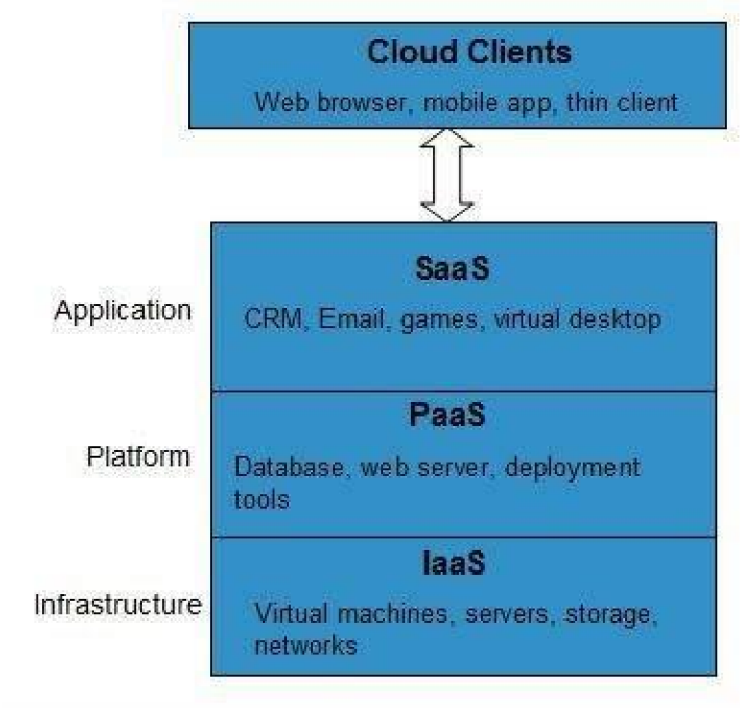
Service Models

Cloud computing is based on service models. These are categorized into three basic service models which are -

- Infrastructure-as–a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Anything-as-a-Service (XaaS) is yet another service model, which includes Network-as-a-Service, Business-as-a-Service, Identity-as-a-Service, Database-as-a-Service or Strategy-as-a-Service.

The **Infrastructure-as-a-Service (IaaS)** is the most basic level of service. Each of the service models inherit the security and management mechanism from the underlying model, as shown in the following diagram:



Infrastructure-as-a-Service (IaaS)

IaaS provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc.

Platform-as-a-Service (PaaS)

PaaS provides the runtime environment for applications, development and deployment tools, etc.

Software-as-a-Service (SaaS)

SaaS model allows to use software applications as a service to end-users.

CHAPTER 7
CONCLUSION

This design of pet feeder provides the features which will make pet care more convenient for both owner and the pet this system also provides all the information about the pet's feeding like is it taking feed or not, This design also helps in stopping wastage of feed by providing the left feed first. And this design also gives rid of the conventional manual setting of the pet feeders with its interactive remote control. And as go for the advancement some of the features can be modified as using cam at place of sensor for priority feed of pet.

CHAPTER 8

BIBLIOGRAPHY

[1]"Perfect Petfeeder Lux Model." Pillar Pet Products, Inc.

[2] "Pet Product Review: Perfect Petfeeder." Itchmo: News For Dogs & Cats.

[3] "ERGO 8 Day Feeder." Pet Street Mall.

[4] "Industry Statistics & Trends". American Pet ProductsManufacturers Association,Inc.

[5] "Do You Like Pets Better Than People?". CBS News.

[6] "The Pet Economy". Business Week

[7] "It's a Pet Economy". Sacramento Business Journal.

[8] "Dog Owner's Guide: Obesity". CanisMajor.com.