A PROJECT REPORT

ON

Development & Implementation of Industrial automation system

*Submitted in partial fulfillment of the requirement*

for

The award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

BY

D. Jyothi [22NN1A0461]

P. Venkata Srikumari [23NN5A0412]

V. Chandana Priya [23NN1A0419]

B. Navya Sri Harshitha [23NN1A0420]

Under The Esteemed Guidance of

**DR.G.Sandhya,Ph.D**

**Professor, Dept. of E.C.E**

****

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VIGNAN’S NIRULA INSTITUTE OF TECHNOLOGY & SCIENCE FOR WOMEN

(An ISO 9001:2008 Certified, Approved by AICTE, Affiliated to JNTU, KAKINADA)

Vignan’s venue, Pedapalakaluru-522009, Guntur (District), A.P (2021-2025)

VIGNAN’S NIRULA INSTITUTE OF TECHNOLOGY & SCIENCE FOR WOMEN

(Affiliated to JNTUK & Approved by AICTE)

Vignan’s Avenue, Pedapalakaluru-522009, Guntur (District), A.P

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

### 

CERTIFICATE



This is to certify that the project report entitled “DEVELOPMENT & IMPLEMENTATION OF INDUSTRIAL AUTOMATION SYSTEM” the bona fide work carried out by D. Jyothi , P.Venkata Saikumari ,V. Chandana Priya , B. Navya Sri Harshitha in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICTION ENGINEERING from J.N.T. UNIVERSITY Kakinada during the academic year 2021-2025.

|  |  |  |
| --- | --- | --- |
| [project Guide]  Dr. G. Sandhya  Professor | External Examiner | [Head of the department]  Dr. G. Sandhya  Professor |

# 

# 

# ACKNOWLEDGEMENT

I wish to avail this opportunity to thank Dr. P. Radhika Principal of Vignan’s Nirula Institute of Technology and Science for Women for her continuous support and giving valuable suggestions during entire period of the project work.

I take this opportunity to express my gratitude to Dr. G. Sandhya, Head of the Department for her great help and encouragement to do this project work.

I would like to express my Unbounded gratefulness to my guide Dr. G. Sandhya

professor in department of Electronics &Communication Engineering for her valuable support and motivation at each and every point in successful completion of the project.

I wish to thank all teaching and non-teaching staff of the Electronics &Communication Engineering Department for giving valuable suggestions in completion of the project.

PROJECT ASSOCIATES

D. Jyothi [22NN1A0461]

P. Venkata Saikumari [23NN5A0412]

V.Chandana Priya [23NN1A0419]

B.Navya Sri Harshitha [23NN1A0420]

**DECLARATION**

We do hereby declare that the project entitled “**DEVELOPMENT & IMPLEMENTATION** OF INDUSTRIAL AUTOMATION SYSTEM” authentic work carried out by us under the esteemed guidance of Dr. G. Sandhya given in the Ethical Code of Conduct of the Institute and that, to the best of our knowledge and belief, it contains no material previously written by another neither person nor material(data, theoretical analysis, figures, and text) for the partially fulfillment of the award of Bachelor of Technology degree and we further admit that this has not been submitted by us anywhere else for the award of any other degree.

PROJECT ASSOCIATES

D. Jyothi [22NN1A0461]

P. Venkata Saikumari [23NN5A0412]

V.Chandana Priya [23NN1A0419]

B.Navya Sri Harshitha [23NN1A0420]

**INDEX**

CONTENTS

ABSTRACT

**CHAPTER 1: INTRODUCTION**

* 1. INTRODUCTION
  2. PROBLEM STATEMENT
  3. PROBLEM SOLUTION
  4. KEY FEATURES OF THE SOLUTION

**CHAPTER 2: BLOCK DIAGRAM**

**CHAPTER 3: COMPONENTS & ITS SPECIFICATIONS**

3.1 ESP8266/NODEMCU MICROCONTROLLER

3.2 ULTRASONIC SENSOR (HC-SR04)

* 1. DHT SENSOR MODULE WITH LED

3.4 MQ -7 GAS SENSOR

3.5 LM7805 POWER SUPPLY MODULE

**CHAPTER 4: SOFTWARE & INSTALLATION PROCEDURE**

4.1 ABOUT ARDUINO IDE AND ITS FEATURES

4.2 WHY ARDUINO IDE?

4.3 WHY ARDUINO IDE FOR ESP32/ESP8266?

4.4 STEP BY STEP INSTALLATION PROCEDURE

**CHAPTER 5: IOT CLOUD & SERVER PLATFORM**

**CHAPTER 6: WORKING PRINCIPAL OF IDEATION**

6.1 TELL ABOUT PROJECT WORKING PROCESSES

**CHAPTER 7: OUTPUT & RESULTS**

7.1 PROJECT PHOTOS

7.2 SCREENSHOTS OF RESULTS

**CHAPTER 8: CONCLUSION**

8.1 PROJECT CONCLUSION

**CHAPTER 9: FUTURE SCOPE**

9.1 FUTURE ADDONS OR UPDATES IN PROJECT

**CHAPTER 10: FINAL CODE & RESOURCES**

10.1 CODE

10.2 RESOURCES

**LIST OF FIGURES**

FIG 2.1. BLOCK DIAGRAM OF HYDROPONIC

FIG 2.2. CIRCUIT DIAGRAM OF HYDROPONIC

FIG 3.1. ESP8266 32BIT-MICROCONTROLLER

FIG 3.2. PH SENSOR (HC-SR04)

FIG 3.3. POTENTIONMETER

FIG 3.4. LM7805 POWER SUPPLY MODULE

FIG 4.1. ARDUINO IDE SCREEN

FIG 4.2. TOOLBAR BUTTON

FIG 4.3. ADD FILE

FIG 4.4. ARDUINO SOFTWARE PAGE

FIG 4.5. RUNNING THE EXECUTABLE ARDUINO IDE 2 INSTALLATION FILE

FIG 4.6. ADDITIONAL BOARDS MANAGER URLS FIELD.

FIG 4.7. UPLOADING THE SKETCH

FIG 4.8. BLYNK LIBRARY INSTALLATION

FIG 5.1. EXAMPLE OF IOT SYSTEM

FIG 5.2. IOT INFRASTRUCTURE

FIG 5.3. PUBLISH-SUBSCRIBE MODEL

FIG 5.4. WORKING OF MQTT

FIG 5.5. ARDUINO IDE READY TO PROGRAM BY USING BLYNK APP

**ABSTRACT**

Hydroponics is an advanced, soil-less farming method that enables plant growth by supplying essential nutrients directly through water. This technique offers higher crop yields, efficient resource utilization, and controlled growing conditions compared to traditional soil-based agriculture. By eliminating soil, hydroponics reduces the risk of soil-borne diseases and allows for cultivation in diverse environments, including urban and indoor settings.

Hydroponic systems come in various types, such as nutrient film technique (NFT), deep water culture (DWC), and aeroponics, each designed to optimize nutrient delivery and plant growth. The method provides precise control over factors like pH, nutrient concentration, and oxygen levels, ensuring optimal plant health. Additionally, hydroponics conserves water by recirculating nutrient solutions, making it an environmentally friendly alternative to conventional farming.

With the integration of automation and IoT, hydroponics can be further enhanced by real-time monitoring and control, improving efficiency and reducing labor costs. As global food demand rises and arable land becomes scarce, hydroponics presents a sustainable and innovative solution for future agriculture.

**CHAPTER 1**

# INTRODUCTION

* 1. **Introduction:** Radar technology plays a vital role in detecting and tracking objects across various sectors, including defense, aviation, automotive, and industrial automation. The development of a radar system using IoT technologies not only makes object detection smarter but also integrates real-time monitoring capabilities. This project aims to design a radar system using ultrasonic sensors mounted on a servo motor to detect object distance and position. The use of Blynk, an IoT platform, will allow users to receive real-time notifications on the detected object's distance and position, providing an efficient and user-friendly system for object tracking and environmental monitoring.

This project focuses on integrating an ultrasonic sensor with a servo motor to rotate the sensor across a specific range to capture a full 360-degree view of the surrounding environment. As the sensor detects objects, it measures their distance and the angle at which they are located, and this information is transmitted to the Blynk IoT platform via an ESP32 / ESP8266 or NodeMCU microcontroller. The data can then be viewed on a mobile device, allowing users to track objects in real-time from a remote location.

* 1. **Problem Statement:** Traditional radar systems are often expensive and complex, making them inaccessible for small-scale applications like home automation, robotics, or personal security systems. Additionally, these systems typically lack integration with IoT platforms, limiting the ability to receive real-time data on mobile devices.

There is a growing need for an affordable, easy-to-implement radar system that can detect and monitor objects in real-time while providing remote accessibility and notifications. This project addresses this gap by creating a cost-effective IoT-based radar system that detects objects’ positions and distances using an ultrasonic sensor and transmits the data to the Blynk platform for easy monitoring

* 1. **Problem Solution** The proposed solution is an IoT-enabled radar system using an ultrasonic sensor and a servo motor for object detection. The servo motor rotates the ultrasonic sensor, which scans the surroundings and measures the distance to any detected object. The system calculates both the distance and the angle of the detected object. The use of an ESP32 / ESP8266 or NodeMCU enables Wi-Fi connectivity to send the collected data to the Blynk IoT platform, where users can view the object’s distance and position in real-time through a mobile or web application**.**

## Key Features of the Solution:

Object Detection and Distance Measurement: The ultrasonic sensor detects the object and measures the distance to it using time-of-flight principles.

Servo Motor for 360-degree Scanning: The servo motor allows the ultrasonic sensor to rotate and cover a wide area for object detection.

IoT Integration with Blynk: The ESP32 / ESP8266 / NodeMCU sends real-time data to the Blynk platform, allowing users to monitor object detection and distance on their mobile devices.

Alerts and Notifications: The system can be configured to send notifications to the user when an object is detected within a certain distance range, enhancing security and surveillance applications.

## Literature Survey

Ultrasonic Radar Systems: Ultrasonic sensors are widely used for distance measurement and object detection due to their accuracy, low cost, and ease of integration. The concept of using ultrasonic waves for distance measurement relies on the transmission and reception of sound waves. When a wave hits an object, it is reflected back to the sensor, which calculates the time taken for the wave to return, determining the object’s distance. Researchers have demonstrated the use of ultrasonic sensors in automated systems, robotics, and industrial applications, but integrating these systems with IoT platforms is a relatively new field of study.

Servo Motor Integration in Radar: The use of servo motors in radar systems is essential for providing angular movement, allowing the sensor to rotate and cover a wider field of view. Servo motors can be precisely controlled using PWM (Pulse Width Modulation), which ensures that the sensor scans the environment at regular intervals. Servo motor-based systems have been employed in various radar applications, particularly in robotic automation and obstacle detection, where full environmental scanning is necessary for navigation.

IoT and Blynk for Real-Time Monitoring: IoT platforms like Blynk have revolutionized how sensor data is monitored and controlled remotely. Blynk is an open-source platform that enables users to build mobile applications to interact with sensors and microcontrollers via the internet. Previous studies on IoT-based radar systems have demonstrated the efficiency of using platforms like Blynk for monitoring real-time sensor data from mobile devices.

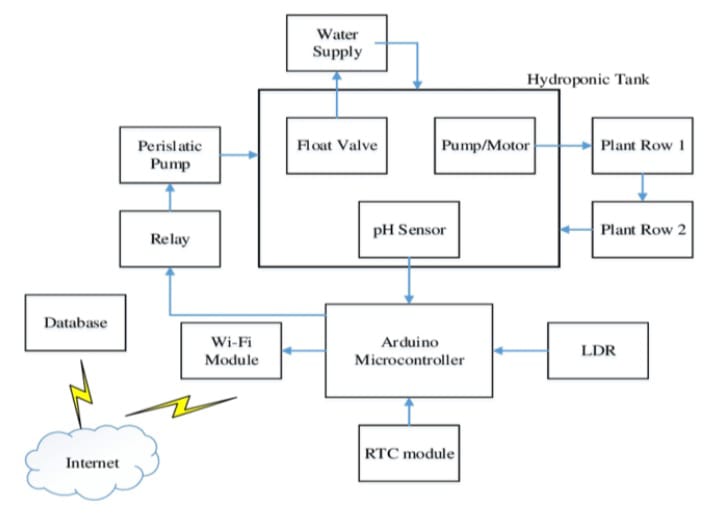
Blynk's ability to provide a user-friendly interface with data visualization tools has made it a popular choice for IoT enthusiasts and developers.

Radar and IoT in Smart Security Systems: The combination of radar technology and IoT has become increasingly significant in developing smart security systems. IoT-enabled radar systems provide remote monitoring capabilities and can send notifications when suspicious objects are detected. In recent years, research has focused on integrating radar with IoT for applications such as perimeter security, unmanned aerial vehicle (UAV) detection, and smart home systems. These studies highlight the potential for using radar systems in IoT networks to enhance real-time monitoring and automation in various industries.

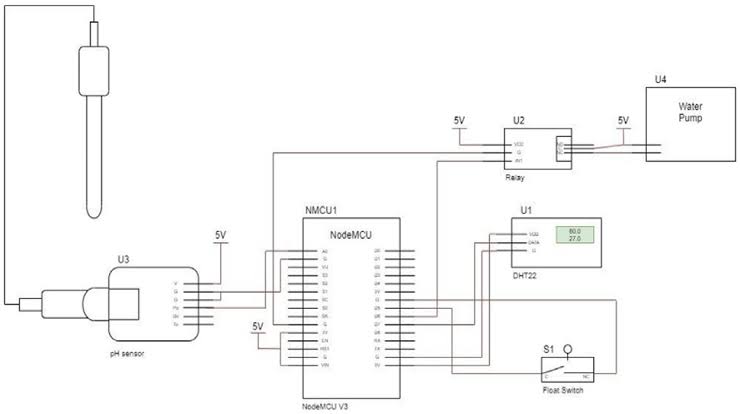
Existing Low-Cost Radar Solutions: Many radar systems used in industries such as defense or aviation are prohibitively expensive. Recent advancements in microcontroller technology, such as the ESP32 / ESP8266 and NodeMCU, along with low-cost sensors like ultrasonic modules, have made it possible to develop affordable radar systems for small-scale applications. Research papers have explored how these components can be combined to create low-cost radar solutions for personal and industrial use. However, there is still a need for more comprehensive systems that integrate object detection, data transmission, and remote monitoring, all of which are addressed in this project.

# BLOCK DIAGRAM & CIRCUIT DIAGRAM

* 1. **Block Diagram**

****

* 1. **Circuit Diagram**

****

# COMPONENTS & ITS SPECIFICATIONS

## ESP32 / ESP8266/NodeMCU Microcontroller

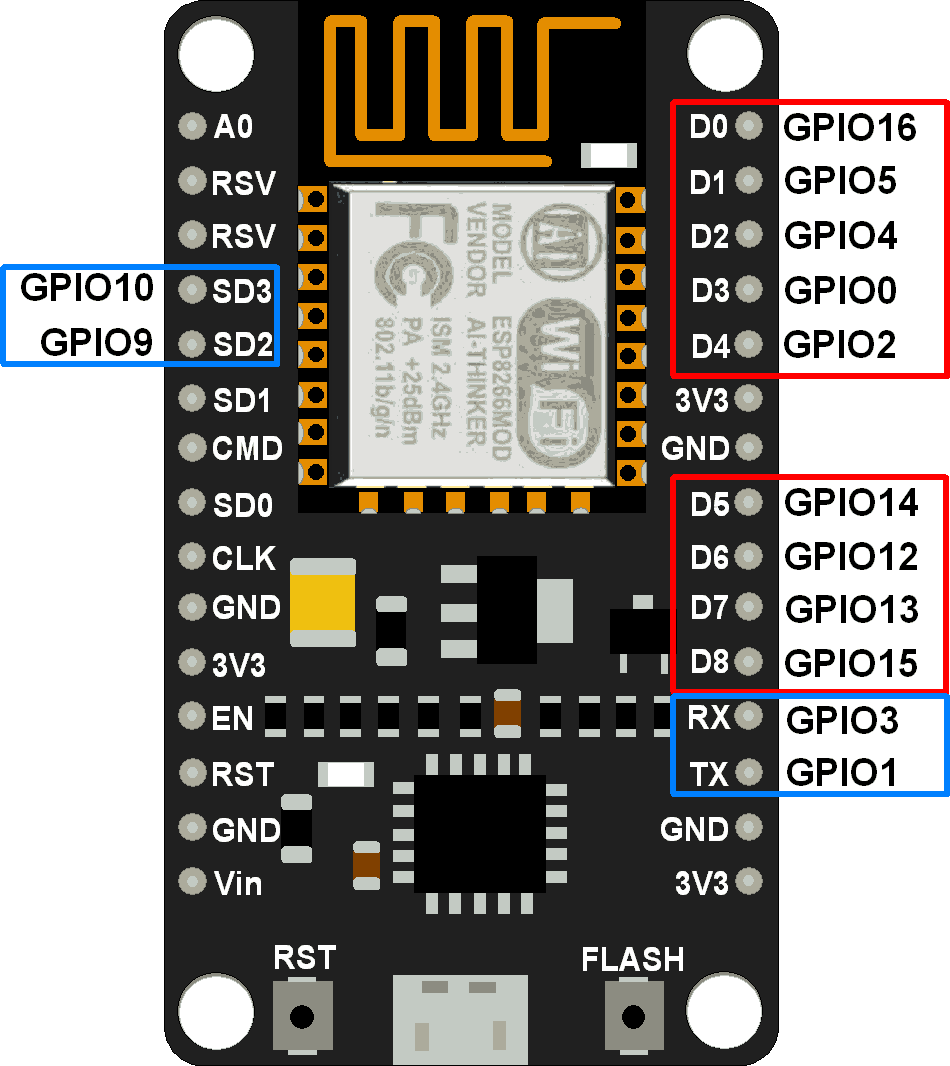
****

Fig 1. ESP8266 32bit-Microcontroller

The ESP32 / ESP8266 or NodeMCU is the heart of the project, handling communication between the ultrasonic sensor and the Blynk IoT platform. It also controls the servo motor and processes the distance and angle data from the ultrasonic sensor.

Microcontroller: ESP32 / ESP8266-WROOM-32 or ESP8266 NodeMCU Operating Voltage: 3.3V

Input Voltage: 7-12V (through Vin pin) Wi-Fi: 802.11 b/g/n

Bluetooth: BLE and Classic (only in ESP32 / ESP8266) GPIO Pins: 36 (ESP32 / ESP8266) or 17 (NodeMCU)

PWM Pins: ESP32 / ESP8266: up to 16 channels, NodeMCU: 8 PWM channels Flash Memory: 4MB

CPU: Dual-core (ESP32 / ESP8266), Single-core (NodeMCU) Clock Speed: 240 MHz (ESP32 / ESP8266), 80 MHz (NodeMCU)

Analog Input Pins: 18 ADC channels (ESP32 / ESP8266), 1 ADC channel (NodeMCU)

## PH Sensor (HC-SR04)

The ultrasonic sensor is used to detect the distance of an object by emitting sound waves and receiving their reflections.



Fig 2. ESP8266 32bit-Microcontroller

Model: REES52

Operating Voltage: 5V DC Quiescent Current: 50Ma

Working Current: 5-10mA

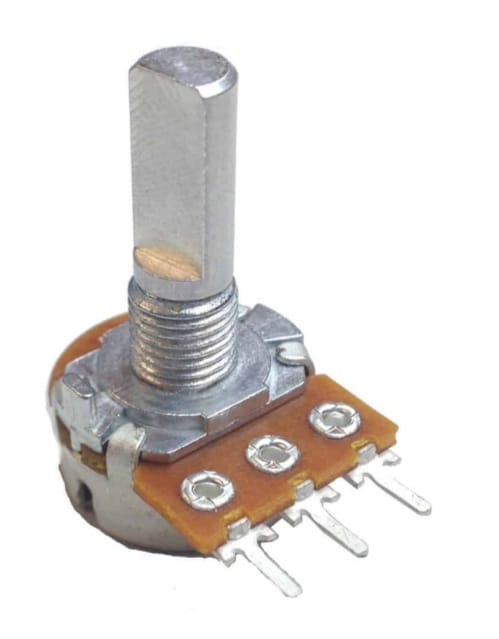
Measurement Range: 0-14

Accuracy: ±0.1mm

Operating Frequency: 40 kHz Trigger Input Signal: 10µs pulse

Echo Output Signal:Digital signal

## POTENTIONMETER:



A potentiometer consists of a resistive track (usually made of carbon, cermet, or a metal oxide) and a wiper that moves along the track to vary the resistance. The wiper is connected to an adjustable knob or slider. It has three terminals:

Two outer terminals – These are connected to a voltage source (typically a fixed voltage).

One middle terminal – This is the "wiper" terminal, which provides an adjustable output voltage based on the position of the wiper.

As the wiper moves, the resistance between the wiper and each outer terminal changes, which alters the output voltage across the wiper. This makes the potentiometer useful for varying voltage levels.

Model:Spice

Operating Voltage: 2V-5V

Operating Speed: 0.1s/60 degrees (4.8V) Stall Torque: 1.8 kgf.cm (4.8V)

Control Signal: Analog Or Digital Converter

Angle of Rotation: 270-360 degrees

Dimensions: 9mm x 1000mm

Weight: 6g

Functionality:

The SG90 servo motor rotates the ultrasonic sensor to scan the surroundings. It moves in

increments to cover a 180-degree field of view, providing distance and angle data for objects detected in the sensor’s.

## LM7805 Power Supply Module

The LM7805 is a widely used linear voltage regulator designed to provide a stable 5V output from a higher input voltage, typically between 7V and 35V. It is ideal for applications where consistent 5V DC power is required, making it a perfect fit for this radar project that involves microcontrollers like the ESP32 / ESP8266, servo motors, and ultrasonic sensors.



Fig 4. LM7805 Power Supply Module

Specifications:

Input Voltage Range: 7V to 35V Output Voltage: 5V ± 2% tolerance Maximum Output Current: 1.5A Quiescent Current (Idle Current): 5mA

Dropout Voltage: 2V (minimum input must be 2V higher than the output, i.e., 7V for a 5V output)

Thermal Protection: Built-in overheat protection Short Circuit Protection: Yes

Operating Temperature Range: 0°C to 125°C Package Type: TO-220

Pinout:

Input (Vin): The voltage input pin, where a higher voltage (7V to 35V) is supplied. Ground (GND): The common ground pin shared by both input and output circuits. Output (Vout): The regulated 5V output pin.

**CHAPTER 4**

# SOFTWARE & INSTALLATION PROCEDURE

## About Arduino IDE and its features

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

The program or code written in the Arduino IDE is often called as sketching. We can connect Arduino board / ESP32 / ESP8266 / NodeMCU with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino '

The Arduino IDE will appear as in Fig 5:

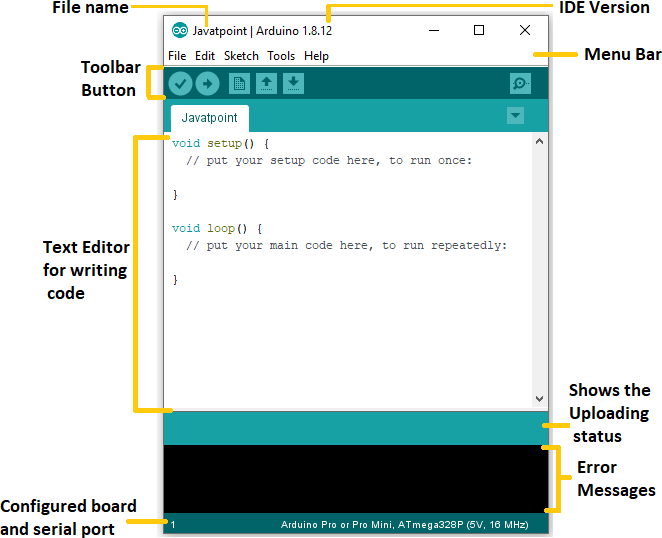
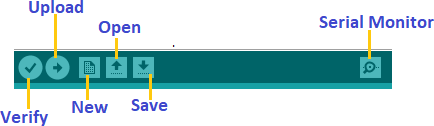


Fig 5. Arduino IDE Screen

Each section of the Arduino IDE display is detailed below.

## Toolbar Button

The icons displayed on the toolbar are **New, Open, Save, Upload,** and **Verify**. It is shown below:



## Upload

The Upload button compiles and runs our code written on the screen. It further uploads the code to the connected board. Before uploading the sketch, we need to make sure that the correct board and ports are selected.

We also need a USB connection to connect the board and the computer. Once all the above measures are done, click on the Upload button present on the toolbar.

The latest Arduino boards can be reset automatically before beginning with Upload. In the older boards / certain ESP32 MCU, we need to press the Reset button present on it. As soon as the uploading is done successfully, we can notice the execution of the program example: The Blink of the LED.

If the uploading is failed, it will display the message in the error window.

We do not require any additional hardware to upload our sketch using the Arduino Bootloader.

A Bootloader is defined as a small program, which is loaded in the microcontroller present on the board.

## Open

The Open button is used to open the already created file. The selected file will be opened in the current window.

## Save

The save button is used to save the current sketch or code.

## New

It is used to create a new sketch or opens a new window.

## Verify

The Verify button is used to check the compilation error of the sketch or the written code.

## Serial Monitor

The serial monitor button is present on the right corner of the toolbar. It opens the serial monitor.

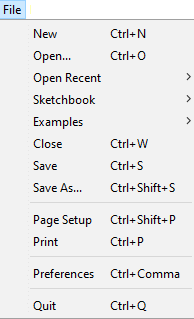
It is shown below:

Arduino IDE

When we connect the serial monitor, the board will reset on the operating system Windows, Linux, and Mac OS X. If we want to process the control characters in our sketch, we need to use an external terminal program. The terminal program should be connected to the COM port, which will be assigned when we connect the board to the computer.

## Menu Bar in IDE File Menu

When we click on the File button on the Menu bar, a drop-down list will appear. It is shown below:



Let's discuss each option in detail.

## New

The New button opens the new window. It does not remove the sketch which is already present.

## Open

It allows opening the sketch, which can be browsed from the folders and computer drivers.

## Open Recent

The Open Recent button contains the list of the recent sketches.

## Sketchbook

It stores the current sketches created in the Arduino IDE software. It opens the selected sketch or code in a new editor at an instance.

## Examples

It shows the different examples of small projects for a better understanding of the IDE and the board. The IDE provides examples of self-practice.

## Close

The Close button closes the window from which the button is clicked.

## Save

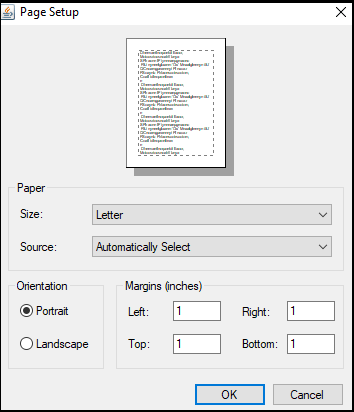
The save button is used to save the current sketch. It also saves the changes made to the current sketch. If we have not specified the name of the file, it will open the '**Save As...'** window.

## Save As

We can save the sketch with a different name using the '**Save As...'** button. We can also change the name accordingly.

## Page Setup

It allows setting the page margins, orientation, and size for printing. The '**Page Setup**' window will appear as:



## Print

According to the settings specified in the 'Page Setup', it prepares the current sketch for printing.

## Preferences

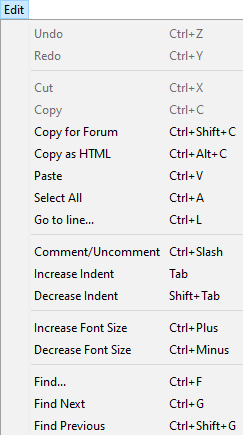
It allows the customization settings of the Arduino IDE.

## Quit

The Quit button is used to close all the IDE windows. The same closed sketch will be reopened when we will open the Arduino IDE.

## Edit Menu

When we click on the Edit button on the Menu bar, a drop-down list appears. It is shown below:



Let's discuss each option in detail.

## Undo

The Undo button is used to reverse the last modification done to the sketch while editing.

## Redo

The Redo button is used to repeat the last modification done to the sketch while editing.

## Cut

It allows us to remove the selected text from the written code. The text is further placed to the clipboard. We can also paste that text anywhere in our sketch.

## Copy

It creates a duplicate copy of the selected text. The text is further placed on the clipboard.

## Copy for Forum

The 'Copy for Forum' button is used to copy the selected text to the clipboard, which is also suitable for posting to the forum.

## Copy as HTML

The 'Copy for Forum' button is used to copy the selected text as HTML to the clipboard. It is desirable for embedding in web pages.

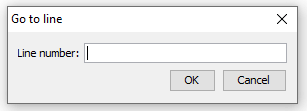
## Paste

The Paste button is used to paste the selected text of the clipboard to the specified position of the cursor.

## Select All

It selects all the text of the sketch.

## Go to line

It moves the cursor to the specified line number. The window will appear as:

## Comment/Decomment

The Comment/ Decomment button is used to put or remove the comment mark (**//**) at the beginning of the specified line.

## Increase Indent

It is used to add the space at the starting of the specified line. The spacing moves the text towards the right.

## Decrease Indent

It is used to subtract or remove the space at the starting of the specified line. The spacing moves the text towards the left.

## Increase Font Size

It increases the font size of the written text.

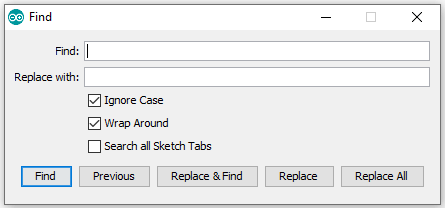
## Decrease Font Size

It decreases the font size of the written text.

## Find

It is used to find the specified text. We can also replace the text. It highlights the text in the sketch.

The window will appear as:



## Find Next

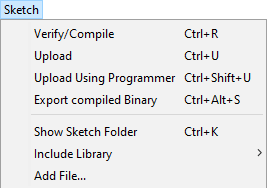
It highlights the next word, which has specified in the '**Find...'** window. If there is no such word, it will not show any highlighted text.

## Find Previous

It highlights the previous word, which has specified in the '**Find'** window. If there is no such word, it will not show any highlighted text.

## Sketch Menu

When we click on the Sketch button on the Menu bar, a drop-down list appears. It is shown below:



Let's discuss each option in detail.

## Verify/Compile

It will check for the errors in the code while compiling. The memory in the console area is also reported by the IDE.

## Upload

The Upload button is used to configure the code to the specified board through the port.

## Upload Using Programmer

It is used to override the Bootloader that is present on the board. We can utilize the full capacity of the Flash memory using the '**Upload Using Programmer**' option. To implement this, we need to restore the Bootloader using the **Tools**-> **Burn Bootloader** option to upload it to the USB serial port.

## Export compiled Binary

It allows saving a .**hex** file and can be kept archived. Using other tools, .hex file can also be sent to the board.

## Show Sketch Folder

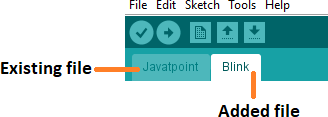
It opens the folder of the current code written or sketch.

## Include Library

Include Library includes various Arduino libraries. The libraries are inserted into our code at the beginning of the code starting with the #. We can also import the libraries from .zip file.

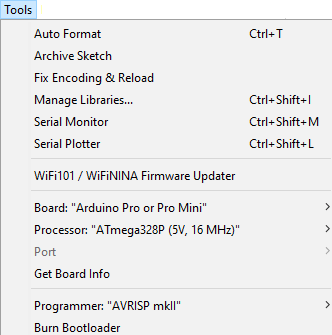
## Add File...

The Add File... button is used to add the created file in a new tab on the existing file. For example, let's add '**Blink**' file to the '**Javatpoint**' file. The tab will now appear as:



## Tools Menu

When we click on the Tools button on the Menu bar, a drop-down list appears. It is shown below:



Let's discuss each option in detail.

## Auto Format

The Auto Format button is used to format the written code. For example, lining the open and closed curly brackets in the code.

## Archive Sketch

The copy of the current sketch or code is archived in the .zip format. The directory of the archived is same as the sketch.

## Fix Encoding and Reload

This button is used to fix the inconsistency between the operating system char maps and editor char map encoding.

## Manage Libraries...

It shows the updated list of all the installed libraries. We can also use this option to install a new library into the Arduino IDE.

## Serial Monitor

It allows the exchange of data with the connected board on the port.

## Serial Plotter

The Serial Plotter button is used to display the serial data in a plot. It comes preinstalled in the Arduino IDE.

## WiFi101/WiFiNINA Firmware Updater

It is used to check and update the Wi-Fi Firmware of the connected board.

## Board

We are required to select the board from the list of boards. The selected board must be similar to the board connected to the computer.

## Processor

It displays the processor according to the selected board. It refreshes every time during the selection of the board.

## Port

It consists of the virtual and real serial devices present on our machine.

## Get Board Info

It gives the information about the selected board. We need to select the appropriate port before getting information about the board.

## Programmer

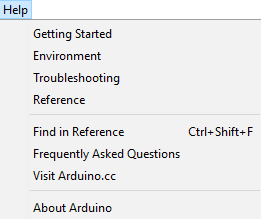
We need to select the hardware programmer while programming the board. It is required when we are not using the onboard USB serial connection. It is also required during the burning of the Bootloader.

## Burn Bootloader

The Bootloader is present on the board onto the microcontroller. The option is useful when we have purchased the microcontroller without the bootloader. Before burning the bootloader, we need to make sure about the correct selected board and port.

## Help Menu

When we click on the Help button on the Menu bar, a drop-down list will appear. It is shown below:



The Help section includes several documents that are easy to access, which comes along with the Arduino IDE. It consists of the number of options such as Getting Started, Environment,

Troubleshooting, Reference, etc. can be accessed without the internet connection as well. It will directly link us to the official website of Arduino.

## Why Arduino IDE?

1. Easy to use: Arduino IDE provides a user-friendly interface for writing, compiling, and uploading code.
2. Cross-platform: Available for Windows, macOS, and Linux.
3. Open source: Free to download, modify, and distribute.
4. Large community: Extensive libraries, tutorials, and forums.
5. Versatile: Supports various microcontrollers, including Arduino, ESP32, and ESP8266.

## Why Arduino IDE for ESP32/ESP8266?

1. Simplified development: Arduino IDE streamlines the development process for ESP32/ESP8266.
2. Library support: Access to a vast library collection, including WiFi, Bluetooth, and IoT-specific libraries.
3. Community resources: Leverage ESP32/ESP8266-specific tutorials, examples, and forums.
4. Easy uploading: Upload code directly from Arduino IDE to ESP32/ESP8266 boards.
5. Cost-effective: No need for expensive development tools or software.

## ESP32-specific benefits:

1. Dual-core processing: Arduino IDE supports ESP32's dual-core architecture.
2. Wi-Fi and Bluetooth: Easily integrate Wi-Fi and Bluetooth capabilities.
3. Low-power consumption: Optimize power usage with Arduino IDE's built-in features.
4. Advanced peripherals: Access ESP32's advanced peripherals, such as SPI, I2C, and UART.

## ESP8266-specific benefits:

1. WiFi integration: Easily integrate WiFi capabilities.
2. Low-cost: ESP8266 is an affordable board.
3. Small footprint: Ideal for IoT projects requiring compact design.
4. Community support: Extensive ESP8266 community and resources.

## Comparison with other IDEs:

1. Platform IO: More complex but offers advanced features and multi-platform support.
2. Visual Studio Code: More versatile but requires additional setup and configuration.
3. ESP-IDF: Official ESP32/ESP8266 IDE, but steeper learning curve.

Arduino IDE provides an ideal balance of ease, versatility, and community support, making it a popular choice for ESP32 and ESP8266 development.

## Step by Step Installation Procedure

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

The Arduino IDE 2 is an open-source project. It is a big step from its sturdy predecessor, Arduino IDE 1.x, and comes with revamped UI, improved board & library manager, debugger, autocomplete feature and much more.

**Step 1** – Download and install the Arduino IDE 2 on your Windows. You can easily download the editor by visit the Arduino Software page: <https://www.arduino.cc/en/software>

## Requirements

* Windows - Win 10 and newer, 64 bits
* Linux - 64 bits
* macOS - Version 10.15: "Catalina" or newer, 64 bits

**For windows installation**: In Arduino IDE 2.3.3 select Win 10 and newer, 64 bits



Fig 6. Arduino Software page

To install the Arduino IDE 2 on a Windows computer, simply run the file downloaded from the software page.

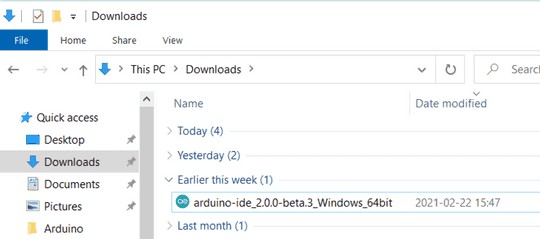
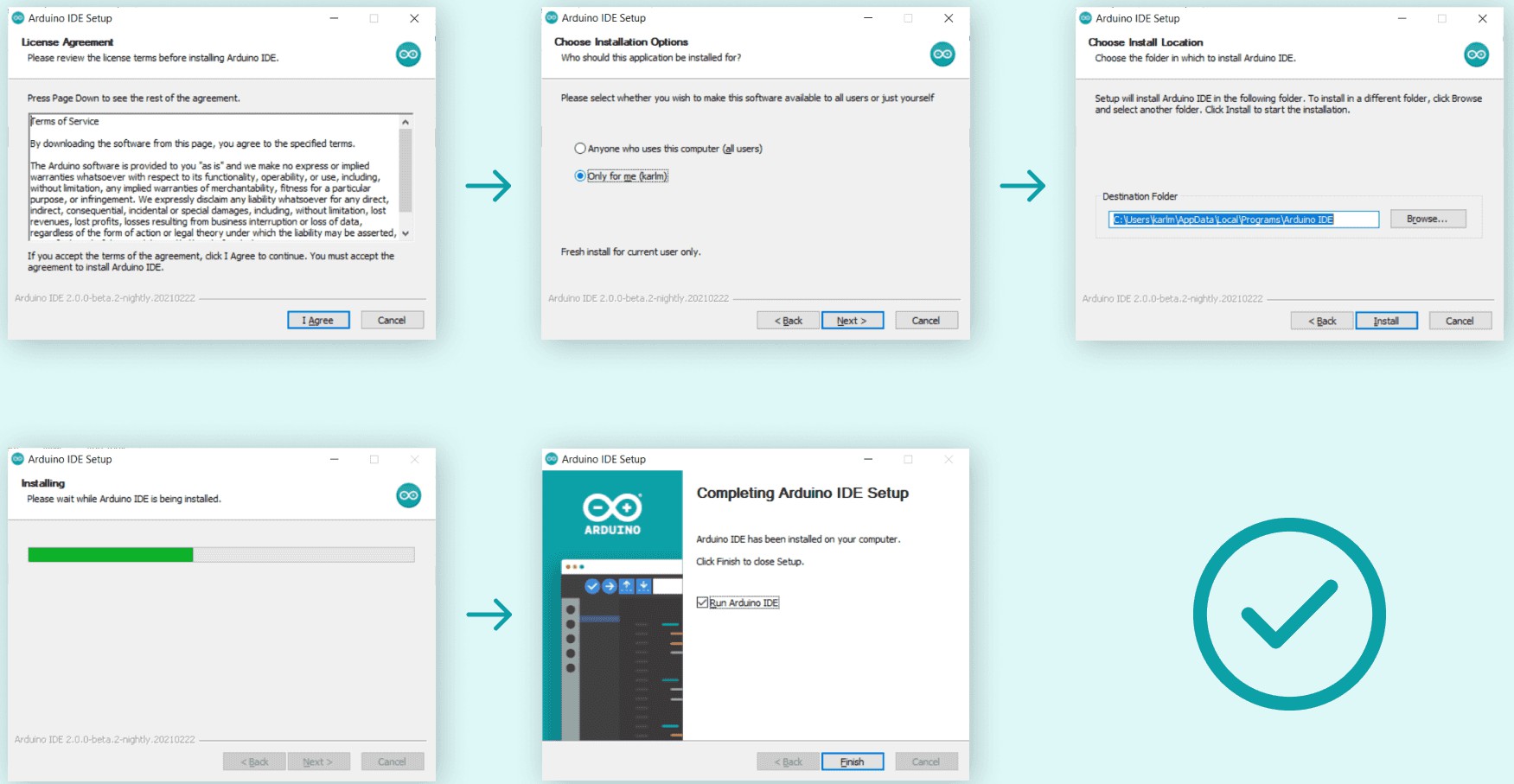


Fig 7. Running the Executable Arduino IDE 2 installation file

Follow the instructions in the installation guide as shown in the image. The installation may take several minutes.



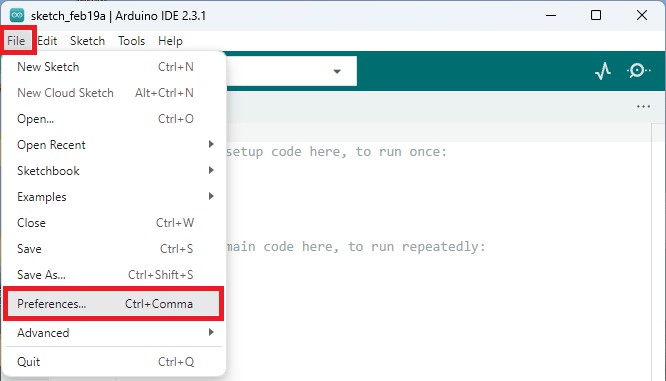
## Installing ESP32 Board in Arduino IDE 2

Before proceeding make sure you have completed above process that is [Arduino IDE](https://www.arduino.cc/en/Tutorial/getting-started-with-ide-v2) [2](https://www.arduino.cc/en/Tutorial/getting-started-with-ide-v2) installed on your computer and follow the instructions given below

## Install ESP32 Add-on in Arduino IDE

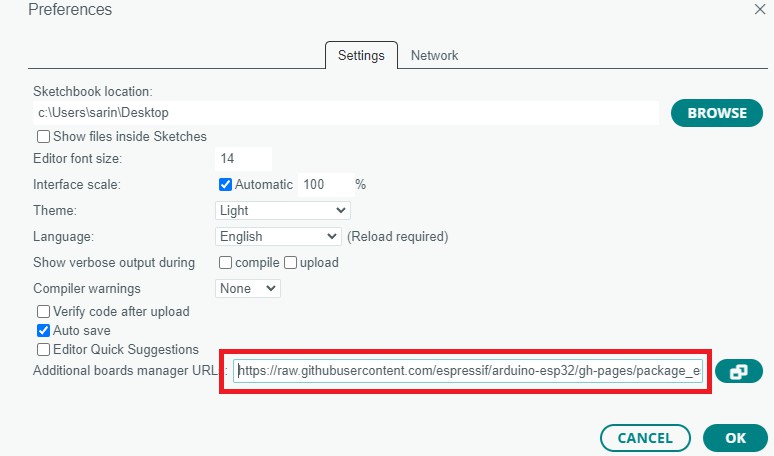
To install the ESP32 board in your Arduino IDE, follow these next instructions:

1. In your Arduino IDE 2, go to **File** > **Preferences**.



1. Copy and paste the following line to the **Additional Boards Manager** URLs field.

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\_esp32\_index.jso](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json) [n](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

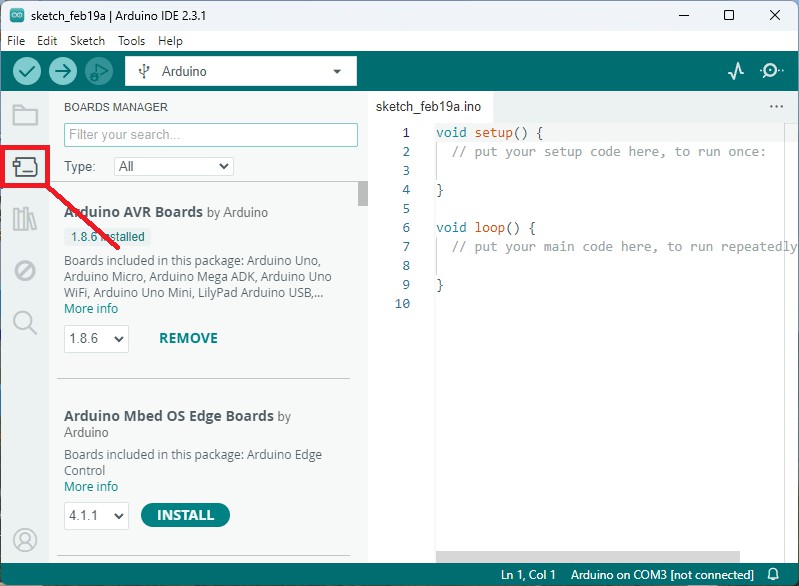


**Note:** if you already have the ESP8266 boards URL, you can separate the URLs with a comma, as follows:

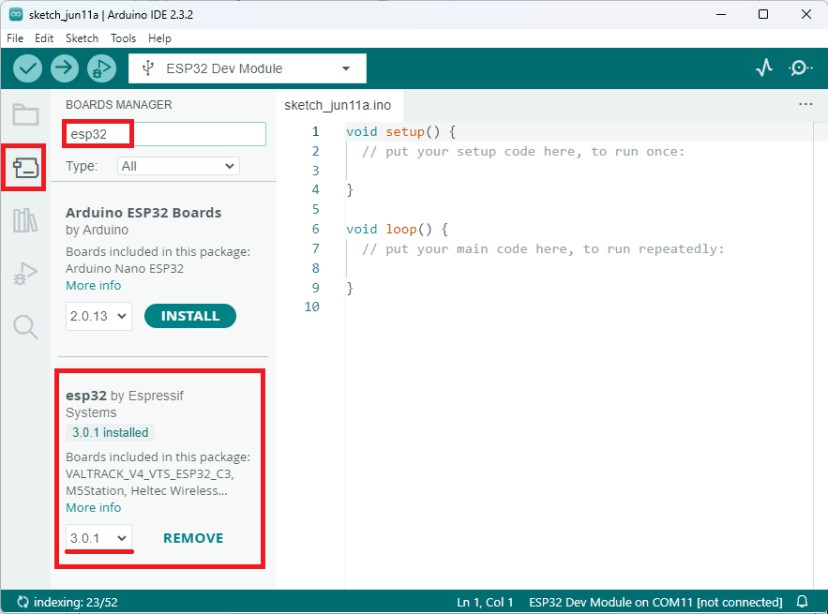
<http://arduino.esp8266.com/stable/package_esp8266com_index.json>,

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\_esp32\_index.jso](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json) [n](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

1. Open the Boards Manager. You can go to **Tools** > **Board** > **Boards Manager…** or you can simply click the **Boards Manager** icon in the left-side corner.



1. Search for **ESP32** and press install button for **esp32 by Espressif Systems version 3.X**.



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

## Testing the Installation

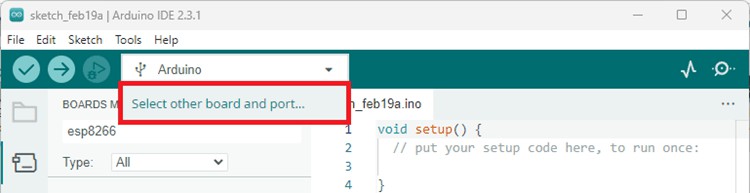
To test the ESP32 add-on installation, upload a simple code that blinks the on-board LED (GPIO 2).

Write the following code to your Arduino IDE:



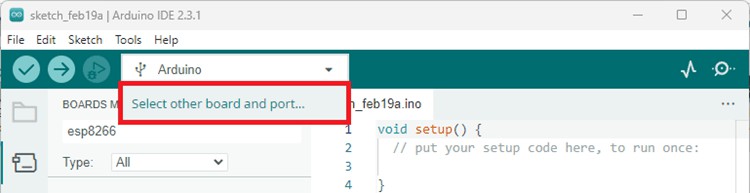
## Uploading the Sketch

Select your board before uploading the code. On the top drop-down menu, click on “**Select other board and port…**”

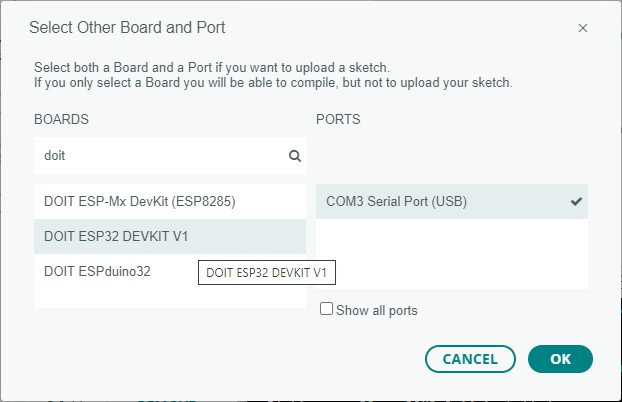


## Uploading the Sketch

Select your board before uploading the code. On the top drop-down menu, click on “**Select other board and port…**”



A new window, as shown below, will open. Search for your ESP32 board model.

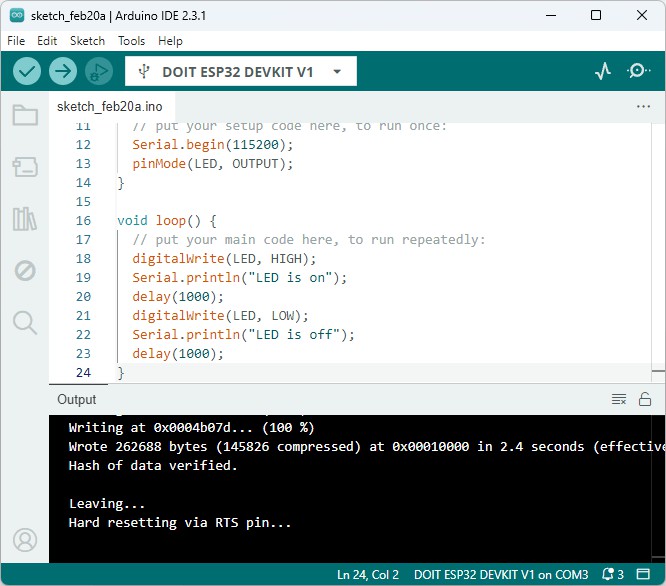


Select the ESP32 board model you’re using, and the COM port. In our example, we’re using the DOIT ESP32 DEVKIT V1. Click **OK** when you’re done.

Now, you just need to click on the **Upload** button shown below.

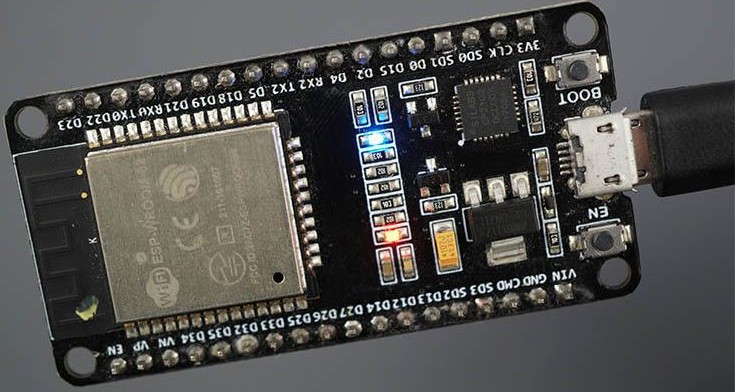
Arduino IDE 2 Upload Button

After a few seconds, the upload should be complete.



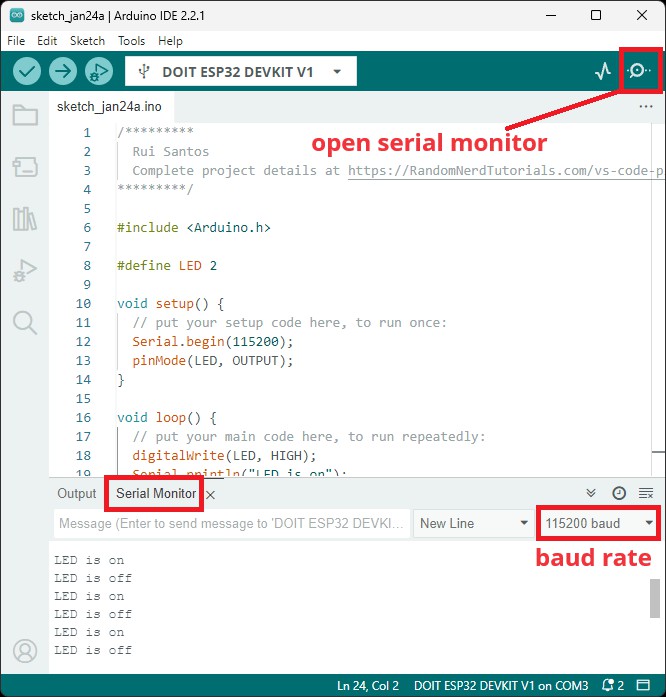
**Note**: some ESP32 development boards don’t go into flashing/uploading mode automatically when uploading a new code and you’ll see a lot of dots on the debugging window followed by an error message. If that’s the case, you need to press the ESP32 BOOT button when you start seeing the dots on the debugging window.

The ESP32 on-board LED should be blinking every second.



## Serial Monitor

You can click on the Serial Monitor icon to open the Serial Monitor tab. Make sure you select the 115200 baud rate.



That’s it! You’ve installed the ESP32 Boards successfully in Arduino IDE 2.

## 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 port 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

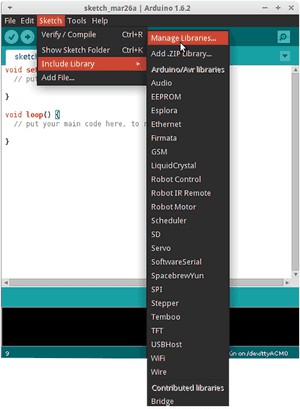
## Installing Libraries

Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the [LiquidCrystal library](https://www.arduino.cc/reference/en/libraries/liquidcrystal/) makes it easy to talk to character LCD displays. There are thousands of libraries available for download directly through the Arduino

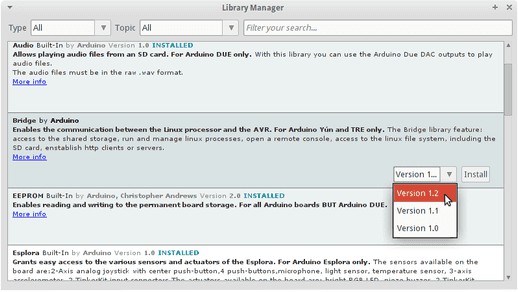
IDE, and you can find all of them listed at the [Arduino Library Reference](https://www.arduino.cc/reference/en/libraries/).

## Using the Library Manager

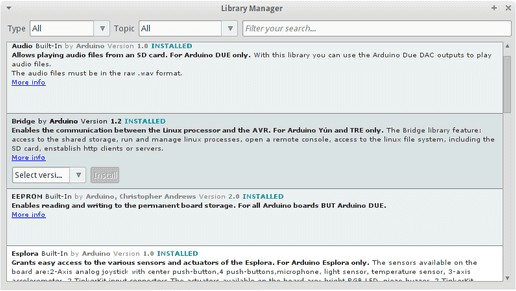
To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.6.2). Open the IDE and click to the "Sketch" menu and then *Include Library > Manage Libraries*.



Then the Library Manager will open, and you will find a list of libraries that are already installed or ready for installation. In this example we will install the Bridge library. Scroll the list to find it, click on it, then select the version of the library you want to install. Sometimes only one version of the library is available. If the version selection menu does not appear, don't worry it is normal.



Finally click on install and wait for the IDE to install the new library. Downloading may take time depending on your connection speed. Once it has finished, an *Installed* tag should appear next to the Bridge library. You can close the library manager.

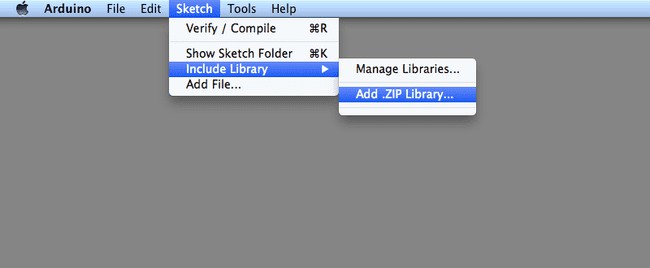


You can now find the new library available in the *Sketch > Include Library* menu. If you want to add your own library to Library Manager, follow [these instructions](https://github.com/arduino/library-registry#adding-a-library-to-library-manager).

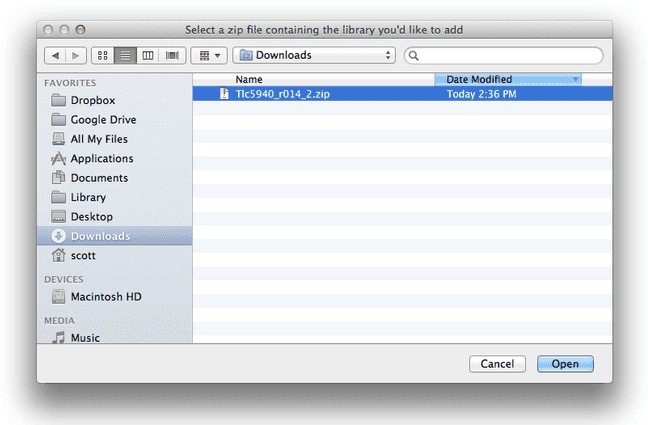
## Importing a .zip Library

Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library. Starting with version 1.0.5, you can install 3rd party libraries in the IDE. Do not unzip the downloaded library, leave it as is.

In the Arduino IDE, navigate to *Sketch > Include Library > Add .ZIP Library*. At the top of the drop-down list, select the option to "Add .ZIP Library''.



You will be prompted to select the library you would like to add. Navigate to the .zip file's location and open it.



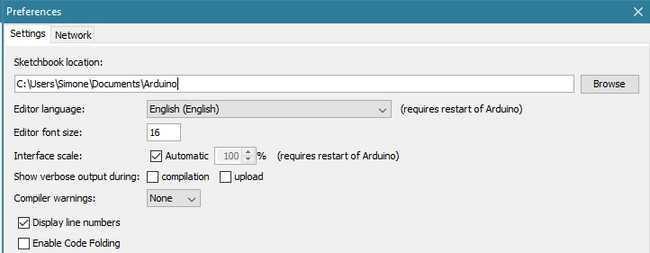
Return to the *Sketch > Include Library menu.* menu. You should now see the library at the bottom of the drop-down menu. It is ready to be used in your sketch. The zip file will have been expanded in the *libraries* folder in your Arduino sketches directory.

Note: the library will be available to use in sketches, but with older IDE versions examples for the library will not be exposed in the *File > Examples* until after the IDE has restarted.

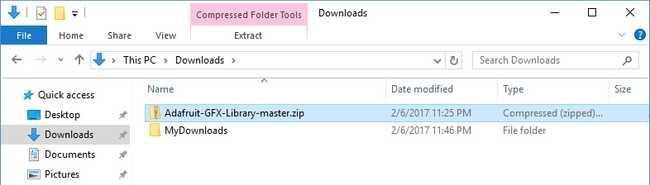
## Manual Installation

When you want to add a library manually, you need to download it as a ZIP file, expand it and put in the proper directory. The ZIP file contains all you need, including usage examples if the author has provided them. The library manager is designed to install this ZIP file automatically as explained in the former chapter, but there are cases where you may want to perform the installation process manually and put the library in the *libraries* folder of your sketchbook by yourself.

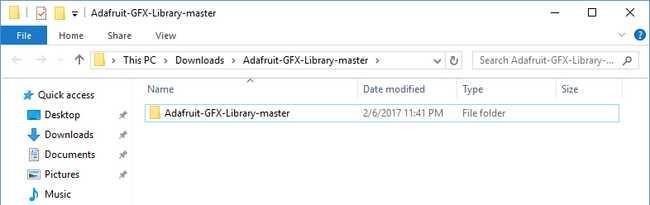
You can find or change the location of your sketchbook folder at *File > Preferences > Sketchbook* location.



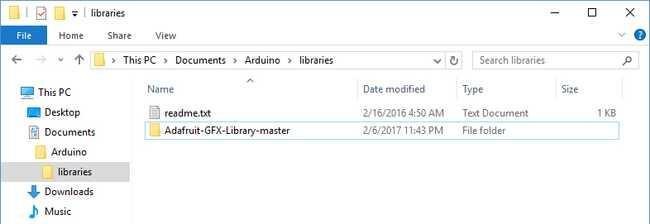
Go to the directory where you have downloaded the ZIP file of the library



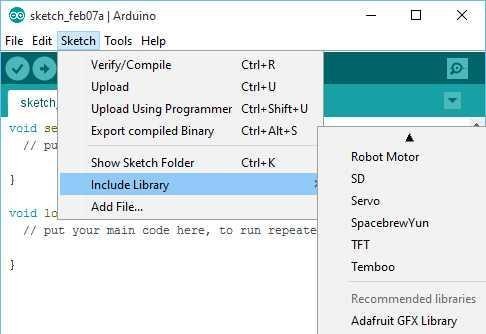
Extract the ZIP file with all its folder structure in a temporary folder, then select the main folder, that should have the library name



Copy it in the "libraries" folder inside your sketchbook.



Start the Arduino Software (IDE), go to *Sketch > Include Library*. Verify that the library you just added is available in the list.

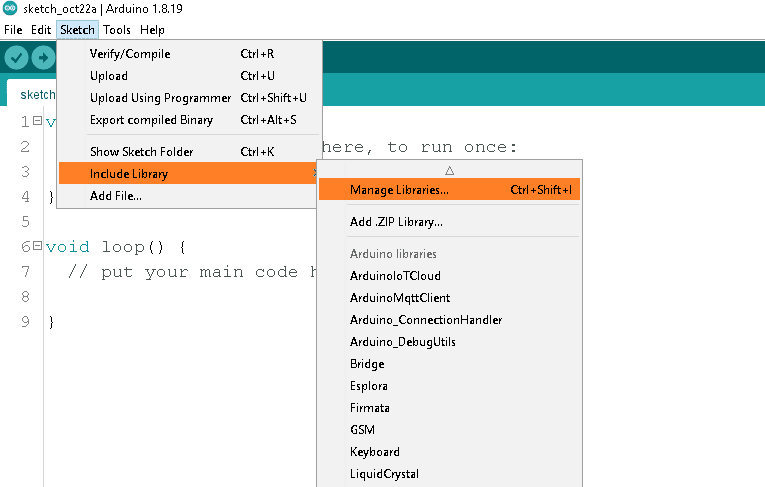


Please note: Arduino libraries are managed in three different places: inside the IDE installation folder, inside the core folder and in the libraries folder inside your sketchbook. The way libraries are chosen during compilation is designed to allow the update of libraries present in the distribution. This means that placing a library in the "libraries" folder in your sketchbook overrides the other libraries versions.

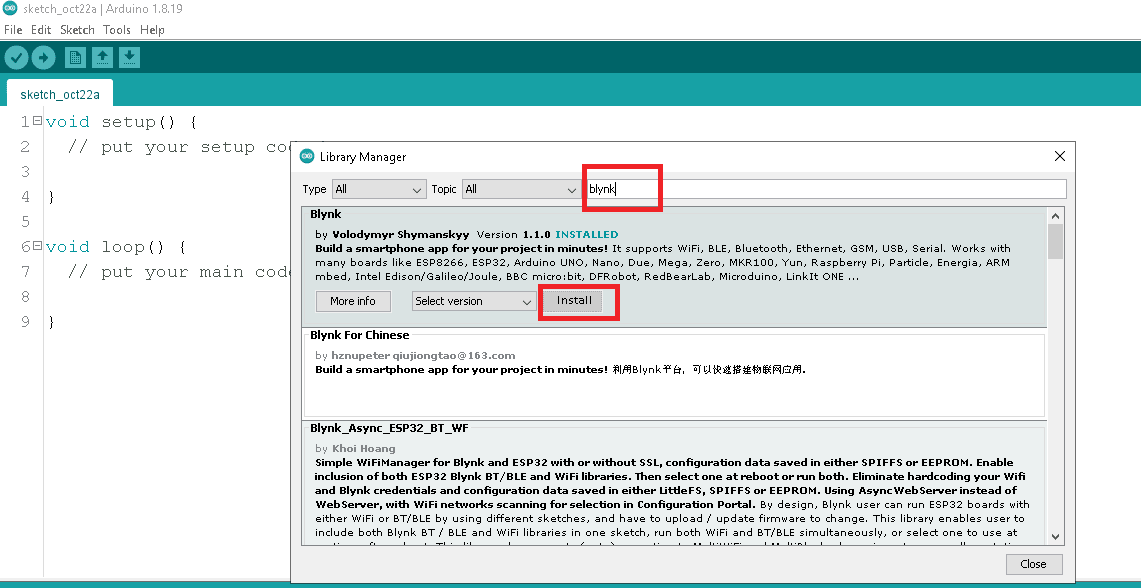
## Blynk Library Installation

To install a new library Blynk into your Arduino IDE, you can use the Library Manager.

First, connect your computer o the Internet. Open the IDE and click to the “Sketch” menu and then Include Library > Manage Libraries.



Then the Library Manager will open and you will find a list of libraries that are already installed or ready for installation. Search for Blynk library and in the version selection choose the latest version to date.



Finally click on Install and wait for the IDE to install the new library. Downloading may take time depending on your connection speed. Once it has finished, an Installed tag should appear next to the Bridge library. You can close the library manager.

**CHAPTER 5:**

# 5. IOT CLOUD &SERVER PLATFORM

What is the internet of things (IoT)?

The internet of things, or IoT, is a network of interrelated devices that connect and exchange data with other IoT devices and the cloud. IoT devices are typically embedded with technology such as sensors and software and can include mechanical and digital machines and consumer objects.

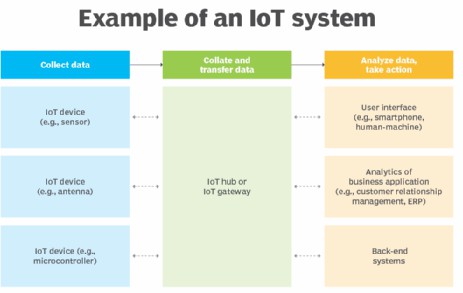
These devices encompass everything from everyday household items to complex industrial tools. Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, deliver enhanced customer service, improve decision-making and increase the value of the business.

With IoT, data is transferable over a network without requiring human-to-human or human-to-computer interactions.

A thing in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low, or any other natural or man-made object that can be assigned an Internet Protocol address and can transfer data over a network.

## How does IoT work?

IoT systems function by gathering data from sensors embedded in IoT devices, which is then transmitted through an IoT gateway for analysis by an application or back-end system



The following four elements are incorporated into an IoT ecosystem for it to function:

## Sensors or devices

An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware to collect, send and act on data acquired from their environments.

## Connectivity

IoT devices can communicate with one another through a network over the internet. These devices share sensor data by connecting to an IoT gateway, which acts as a central hub where IoT devices can send data. Before the data is shared, it can also be sent to an edge device where it is analyzed locally.

## Data analysis

Only the relevant data is used to identify patterns, offer recommendations and identify potential issues before they escalate. Analyzing data locally reduces the volume of data sent to the cloud, which minimizes bandwidth consumption.

Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices. For example, they can set them up, give them instructions or access the data. The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.

IoT can also use artificial intelligence and machine learning to make data collection processes easier and more dynamic.

## Graphical user interface

A graphical user interface ([UI](https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI)) is typically used to manage IoT devices. For example, a website or a mobile app can be used as an UI to manage, control and register smart devices.



Consumer, enterprise and industrial IoT devices include smart TVs and smart sensors outfitted for conference rooms and assembly line machines.

## IoT applications include the following:

* **Agriculture.** IoT can benefit farmers by making their jobs easier. For example, sensors can collect data on rainfall, humidity, temperature and soil content and IoT can help

automate farming techniques. Additionally, IoT devices can be used to oversee the health of livestock, monitor equipment and streamline supply chain management.

* **Construction.** IoT can help monitor operations surrounding infrastructure. Sensors, for example, can monitor events or changes within structural buildings, bridges and other

infrastructure that could potentially compromise safety. This provides benefits such as improved incident management and response, reduced operations costs and improved service quality.

* **Home automation.** A home automation business can use IoT to monitor and manipulate mechanical and electrical systems in a building. Homeowners can also remotely control and automate their home environment by using IoT devices, including smart thermostats, lighting systems, security cameras and voice assistants such as Alexa and Siri for increased comfort and energy efficiency.
* **Smart buildings and cities.** Smart cities can help citizens reduce waste and energy consumption. They can reduce energy costs using sensors that detect how many

occupants are in a room and turning the air conditioner on if sensors detect a conference room is full or lowering the heat if everyone in the office has gone home.

* **Urban consumption systems.** IoT technologies can also be used to monitor and manage urban consumption such as traffic lights, parking meters, waste management systems and public transportation networks.
* **Healthcare monitoring.** IoT devices such as remote patient monitoring systems, smart medical devices and medication trackers let healthcare providers monitor patients' health status, manage chronic conditions and provide timely interventions. IoT gives providers the ability to monitor patients more closely by analyzing the generated data. Hospitals also often use IoT systems to complete tasks such as inventory management for both

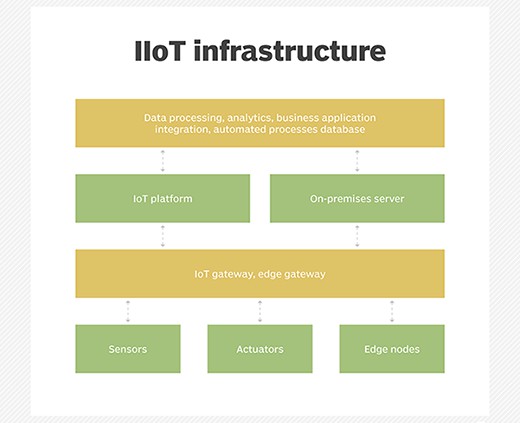
pharmaceuticals and medical instruments.

* **Retail.** IoT sensors and beacons in retail stores can track customer movement, analyze shopping patterns, manage inventory levels and personalize marketing messages. This enhances the shopping experience for customers and optimizes store operations.
* **Transportation.** IoT devices help the transportation industry by monitoring vehicle performance, optimizing routes and tracking shipments. For example, the fuel efficiency of connected cars can be monitored to reduce fuel costs and improve sustainability. IoT devices can also monitor the condition of cargo, so it reaches its destination in optimal condition.
* **Wearable devices.** Wearable devices with sensors and software can collect and analyze user data, sending messages to other technologies about the users to make their lives easier and more comfortable. Wearable devices are also used for public safety -- for example, by improving first responders' response times during emergencies by providing optimized routes to a location or by tracking construction workers' or firefighters' vital

signs at life-threatening sites.

* **Energy management.** IoT-enabled smart grids, smart meters and energy management systems let utility companies and consumers monitor and optimize energy usage,

manage demand-response programs and integrate renewable energy sources more efficiently. For example, the data collected by the IoT devices and sensors helps identify patterns, peak usage times and areas of inefficiency.



## What Is IoT Cloud?

An IoT cloud is a massive network that supports IoT devices and applications. This includes the underlying infrastructure, servers and storage, needed for real-time operations and processing. An IoT cloud also includes the services and standards necessary for connecting, managing, and securing different IoT devices and applications.

## Why IoT Cloud?

IoT clouds offer an efficient, flexible, and scalable model for delivering the infrastructure and services needed to power IoT devices and applications for businesses with limited resources. IoT clouds offer on-demand, cost-efficient hyperscale so organizations can leverage the significant potential of IoT without having to build the underlying infrastructure and services from scratch.

## Servers

Servers in IoT connect physical devices to the internet, allowing them to communicate and

share data. They can be used to monitor and control devices remotely, and to allow devices to interact with each other autonomously.

Here are some types of servers used in IoT:

## IoT Data Server

A data integration controller that can be used in factories, production lines, and cloud

systems. It has standard data management functions for collection, processing, saving, noticing, and publishing.

## IoT web server

A platform that connects devices to the internet, allowing them to communicate and share data. It can be used to monitor and control devices remotely, and to allow devices to interact with each other autonomously.

## Windows Server IoT

A full version of Windows Server that provides enterprise manageability and security to IoT solutions.

## Embedded web server

A small, lightweight web server that can be integrated into other software or hardware systems. It's used to provide a web-based interface for monitoring and controlling devices.

## Network server

A server that plays a crucial role in IoT ecosystems by managing network resources and device communications. It allows IoT networks to scale up to support thousands or even millions of devices.

## Message Queue Telemetry Transport Protocol (MQTT)

Message Queuing Telemetry Transport, or MQTT, is a communications protocol designed for Internet of Things devices with extremely high latency and restricted low bandwidth. Message Queuing Telemetry Transport is a perfect protocol for machine-to-machine (M2M) communication since it is designed specifically for low-bandwidth, high-latency settings.

## What is Message Queue Telemetry Transport Protocol (MQTT)?

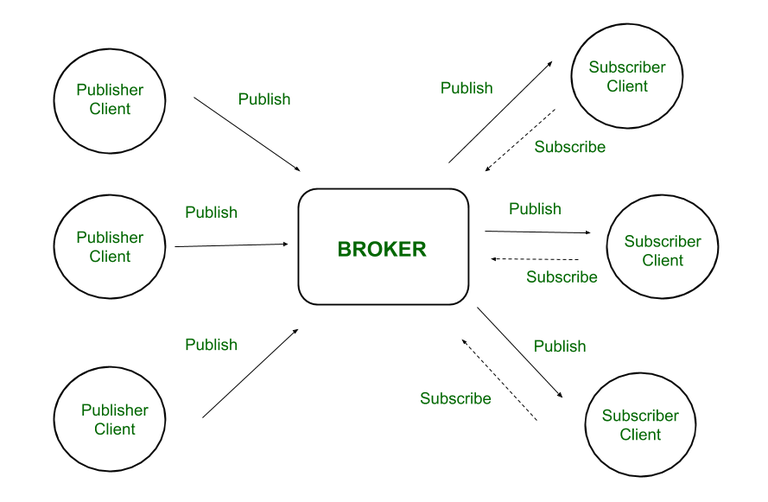
**MQTT** is a simple, lightweight messaging protocol used to establish communication between multiple devices. It is a TCP-based protocol relying on the publish-subscribe model. This communication protocol is suitable for transmitting data between resource-constrained devices having low bandwidth and low power requirements. Hence this messaging protocol is widely used for communication in the IoT Framework.

## Publish-Subscribe Model

This model involves multiple clients interacting with each other, without having any direct connection established between them. All clients communicate with other clients only via a third party known as a Broker.

## MQTT Client and Broker

Clients publish messages on different topics to brokers. The broker is the central server that receives these messages and filters them based on their topics. It then sends these messages to respective clients that have subscribed to those different topics. The heart of any publish/subscribe protocol is the MQTT broker. A broker can handle up to thousands of concurrently connected MQTT customers, depending on how it is implemented. All communications must be received by the broker, who will then sort them, ascertain who subscribed to each one, and deliver the messages to the clients who have subscribed. All persistent customers’ sessions, including missed messages and subscriptions, are likewise kept by the Broker.



*Fig. Publish-Subscribe Model*

Hence client that has subscribed to a specific topic receives all messages published on that topic.

Here the broker is central hub that receives messages, filters them, and distributes them to appropriate clients, such that both message publishers, as well as subscribers, are clients.

## Working of MQTT

MQTT’s publish/subscribe (pub/sub) communication style, which aims to maximise available bandwidth, is an alternative to conventional client-server architecture that communicates directly with an endpoint. In contrast, the client who transmits the message (the publisher) and the client or clients who receive it (the subscribers) are not connected in the pub/sub paradigm. Third parties—the brokers—manage the relationships between the publishers and subscribers because they don’t communicate with one another directly.

Publishers and subscribers, which denote whether a client is publishing messages or has subscribed to receive messages, are examples of MQTT clients. The same MQTT client can be used to accomplish these two features. A publish occurs when a client or device want to submit data to a server or broker.

The term “subscribe” refers to the reversal of the procedure. Several clients can connect to a broker under the pub/sub paradigm and subscribe to subjects that interest them.



When a broker and a subscribing client lose contact, the broker will store messages in a buffer and send them to the subscriber whenever the broker is back up and running. The broker has the right to cut off communication with subscribers and send them a cached message containing publisher instructions if the publishing client abruptly disconnects from the broker.

“Publishers send the messages, subscribers receive the messages they are interested in, and brokers pass the messages from the publishers to the subscribers,” reads an IBM write-up describing the pub/sub paradigm. MQTT clients, such as publishers and subscribers, can only speak with MQTT brokers. Any device or programme that runs a MQTT library can be a MQTT client, ranging from microcontrollers like the Arduino to entire application servers housed in the cloud.

## Characteristics of MQTT

* + **Lightweight:** MQTT is designed to be lightweight, making it suitable for use in aid-restrained environments inclusive of embedded systems and low-strength devices. The protocol minimizes bandwidth and processing overhead, enabling green communication even on restricted networks.
  + **Publish-Subscribe Model:** In the publish-subscribe version, clients (publishers) send messages to subjects, and different clients (subscribers) acquire messages from subjects of interest. This decoupling of producers and purchasers permits for flexible and dynamic conversation styles.
  + **Quality of Service (QoS) Levels:** MQTT supports exclusive stages of message delivery warranty, referred to as Quality of Service (QoS). QoS levels range from 0 to 2, providing various stages of reliability and message transport guarantees, relying at the utility necessities.
  + **Retained Messages:** MQTT lets in agents to store retained messages on topics, making sure that new subscribers acquire the maximum latest message posted on a subject right now after subscribing. This characteristic is beneficial for fame updates and configuration settings.
  + **Last Will and Testament (LWT):** MQTT clients can specify a Last Will and Testament message to be posted by way of the broker in the occasion of an sudden consumer disconnect. This function affords a mechanism for detecting patron failures and dealing with them gracefully.
  + **Security:** MQTT helps various protection mechanisms, consisting of Transport Layer Security (TLS) encryption and authentication mechanisms which include username/password and consumer certificates. These capabilities make certain the confidentiality, integrity, and authenticity of messages exchanged over MQTT connections.

## Advantages of MQTT

This model is not restricted to one-to-one communication between clients. Although the publisher client sends a single message on specific topic, broker sends multiple messages to all different clients subscribed to that topic. Similarly, messages sent by multiple such publisher clients on multiple different topics will be sent to all multiple clients subscribed to those topics. Hence one-to-many, many-to-one, as well as many-to-many communication is possible using this model. Also, clients can publish data and at the same time receive data due to this two-way communication protocol. Hence MQTT is considered to be bi-directional protocol. The default unencrypted MQTT port used for data transmission is 1883. The encrypted port for secure transmission is 8883.

* + Lightweight protocol that is quick to create and allows for efficient data transport
  + Minimal data packet usage, resulting in low network usage
  + Effective data dispersion
  + The effective use of remote sensing and control
  + Prompt and effective message delivery
  + Minimizes power consumption, which is beneficial for the linked devices, and maximizes network capacity.
  + Data transmission is quick, efficient, and lightweight because MQTT messages have small code footprint. These control messages have a fixed header of size 2 bytes and payload message up to size 256 megabytes.

## Introduction about Blynk IoT

Blynk is a software platform that allows users to connect their hardware to the cloud and manage connected devices remotely. Blynk is designed for the Internet of Things (IoT) and can be used for personal projects or commercial products.

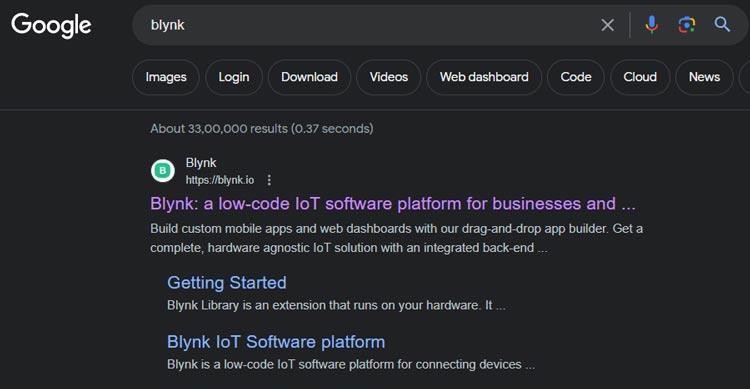
Here are some features of Blynk:

* + No-code approach: Blynk allows users to build and scale IoT without coding. For example, users can control an LED or motor from their phone with just a few clicks.
  + Remote management: Users can remotely control devices from anywhere, receive notifications, and analyze data.
  + Multi-tenancy: Users can configure roles and permissions to control who has access to data.
  + Scalability: Blynk can handle thousands of devices and is used by businesses of all sizes.
  + Open-source: Blynk's server is open-source and can be run locally or on Blynk Cloud.
  + Blynk App: Users can create interfaces for their projects using widgets.
  + Blynk Libraries: Users can communicate with the server and process commands for popular hardware platforms.

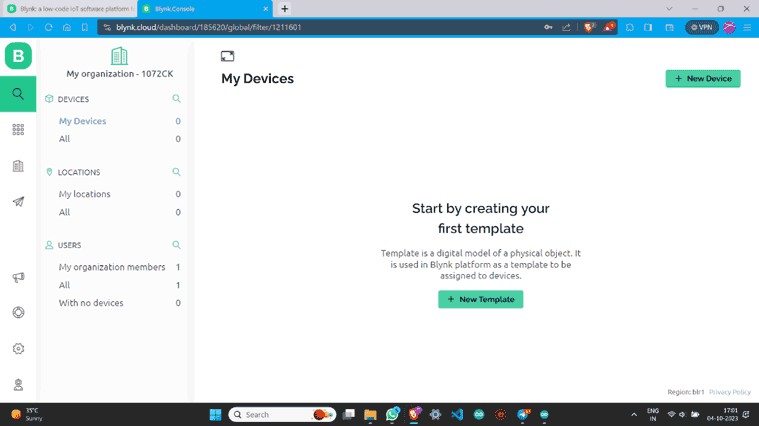
Blynk is free for personal use and prototyping. Businesses can purchase subscriptions to publish Blynk-powered apps for their products.

## Setup Blynk 2.0 App

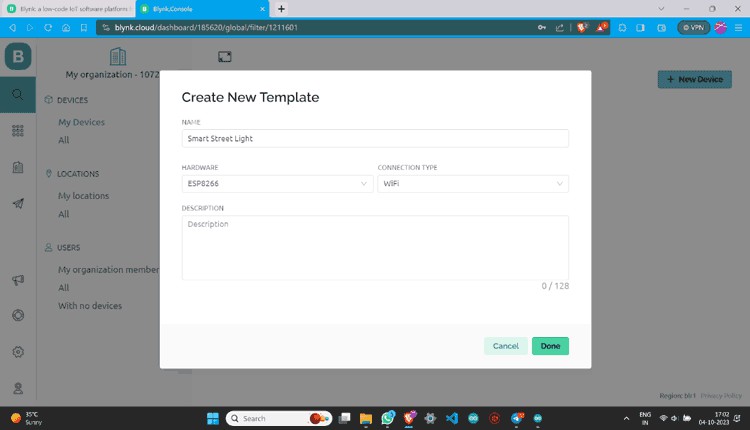
Step 1: Go to the Blynk website and click the "Sign Up" button. Enter your email address and create a password.



Step 2: Once you have signed up on the Blynk and logged in, tap the "+ New Template" button to create a new template.

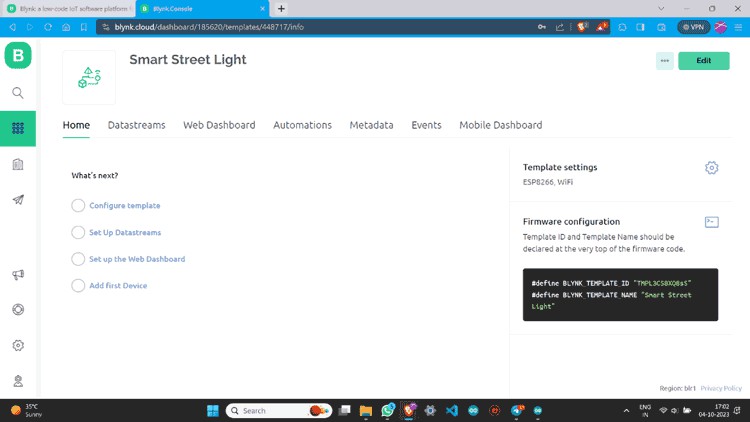


Step 3: Name the template, select the Development Board, set WiFi as the connection type and then save this template.

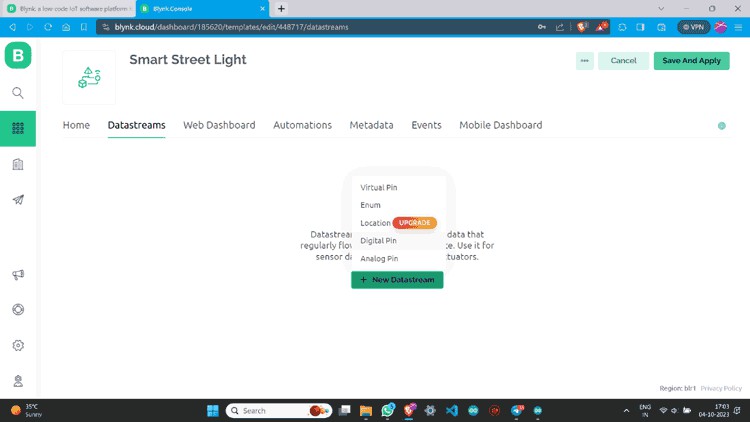


Step 4: Once a new template is created you will see the below screen on your blynk account

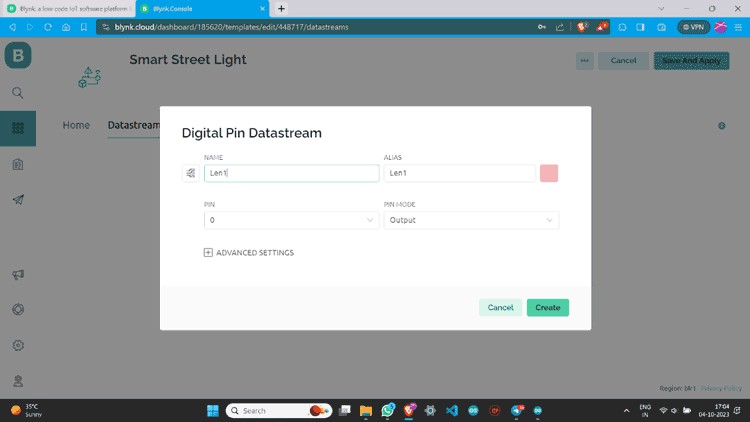
.



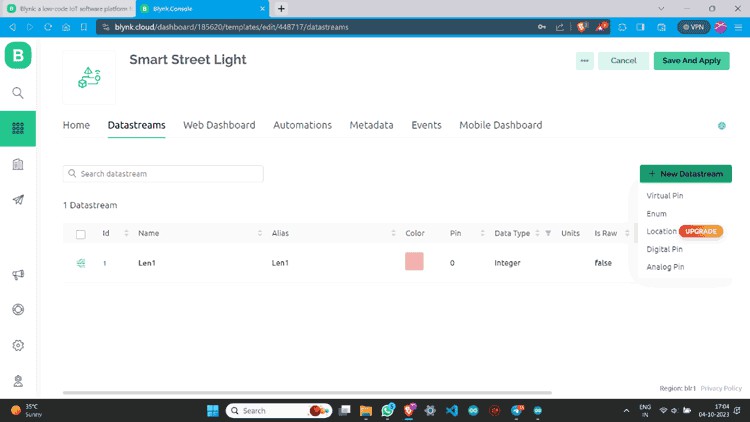
Step 5: Go to Datastreams and click on “+ New Datastream”. Then Choose “Digital Pin”.



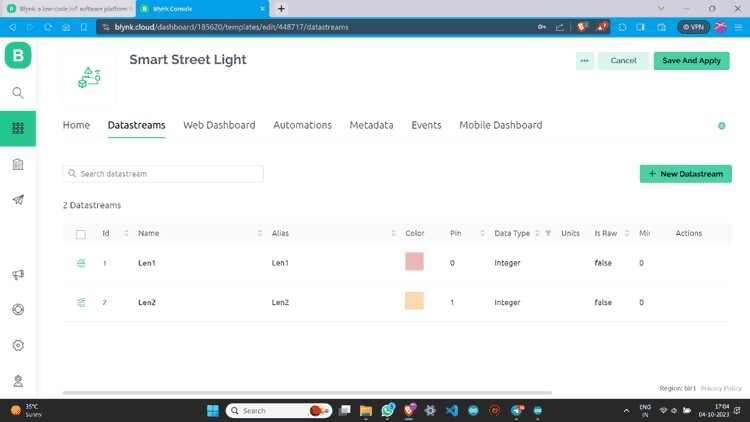
Step 6: Name Datastream and select pin. Assign it’s PIN MODE as Output and Create.



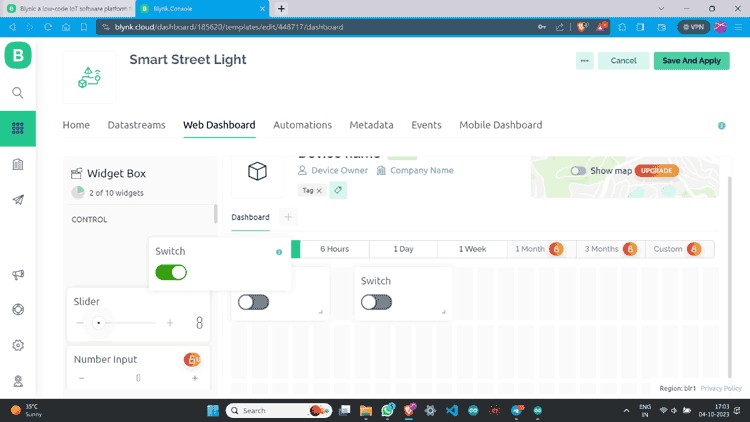
Step 7: Once a datastream is created on the blynk website, it will look something like this below



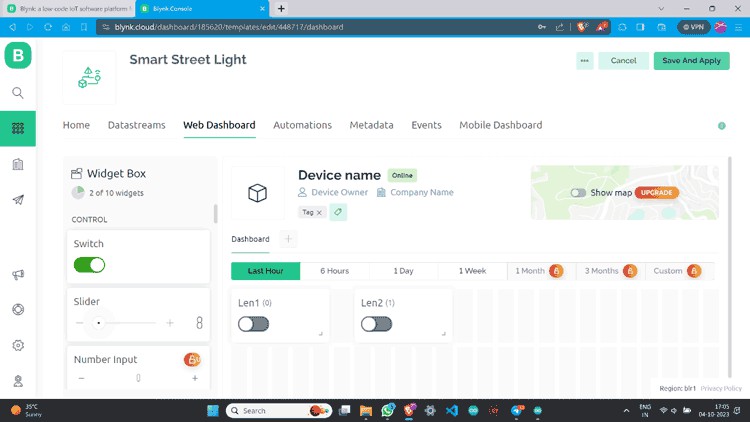
Step 8: I have made one more data stream with the same values.

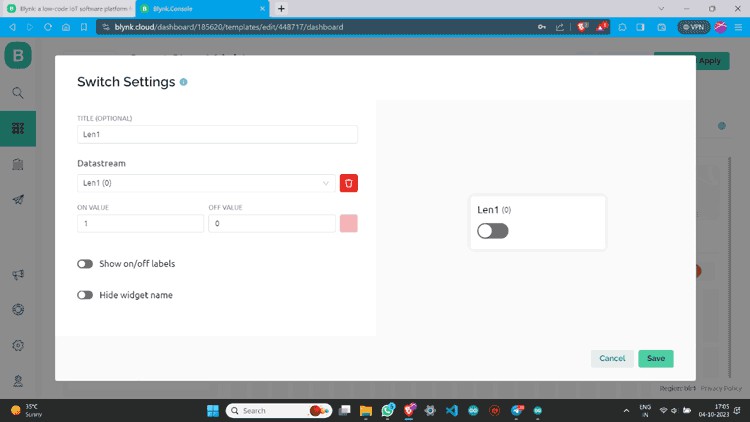


Step 9: Go to “Web Dashboard” and add Switch widgets.

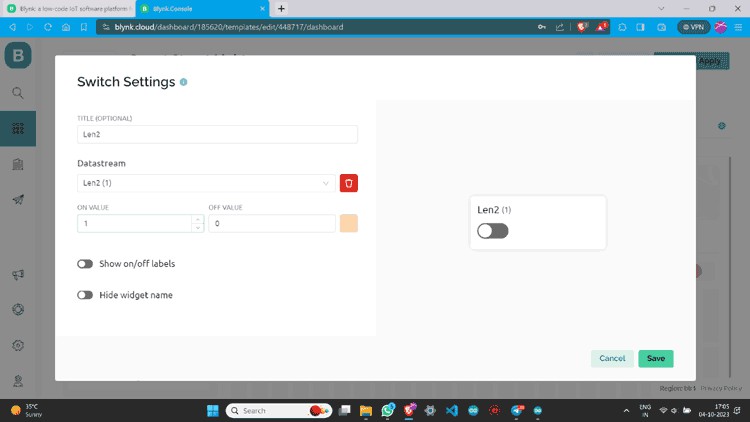


Step 10: Click on the Gear button, give values according to your choice, and save.

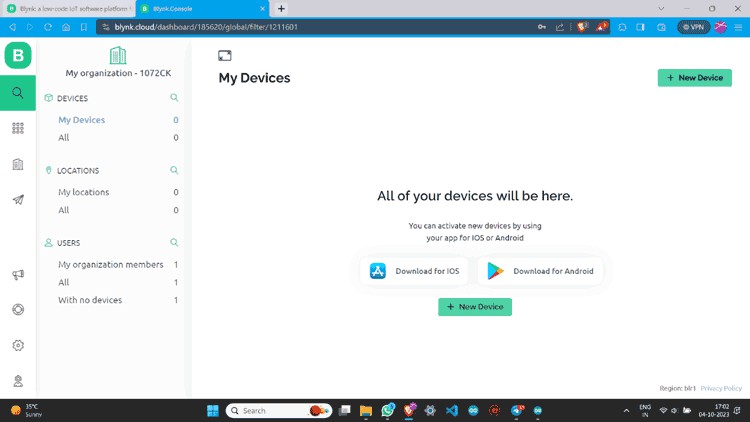




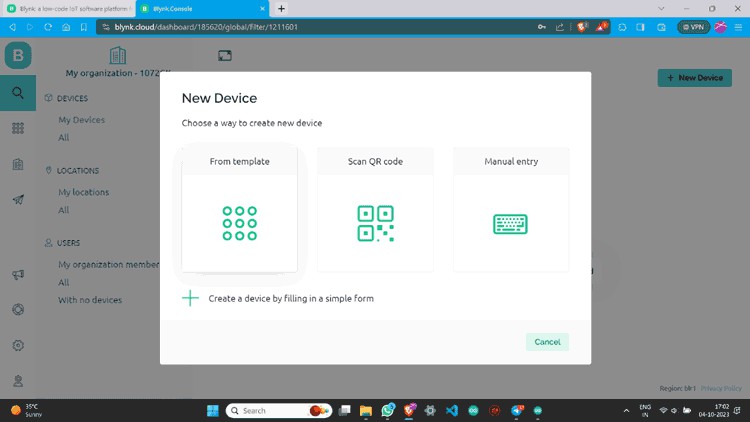
Step 11: I have made one more switch and assigned all values.

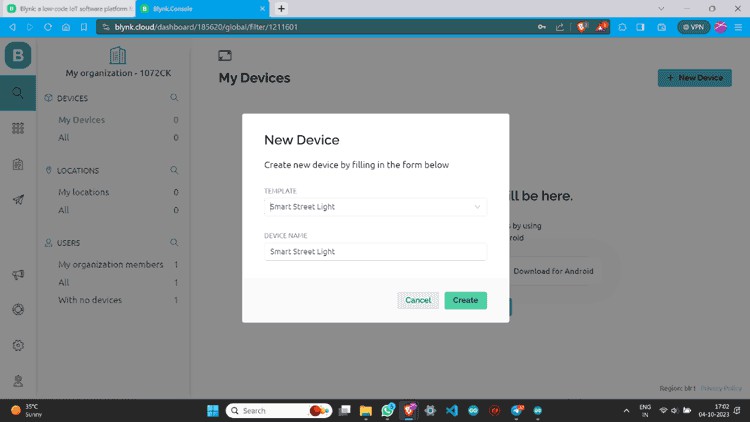


Step 12: Go to the lens icon and click on “+ New Device”.

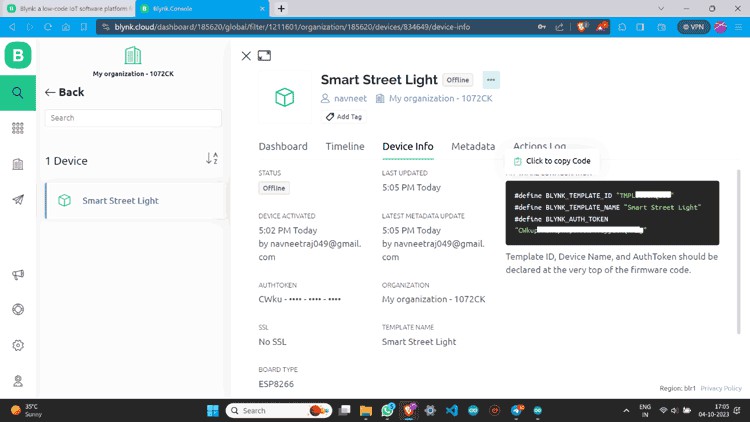


Step 13: Click “From template”, select the template, and create.





Step 14: Now you can see the device. Click on “Device Info” and copy the Auth Token.

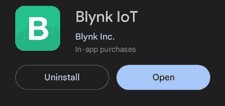


## Blynk 2.0 Application Setup

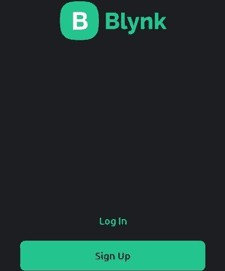
Now after Web setup, we have to make a Mobile App Interface so that we can able to control it from any Android or IOS

Step 1: Download the Blynk app.

The Blynk app is available for both IOS and Android devices. You can download the app from the App Store or Google Play.



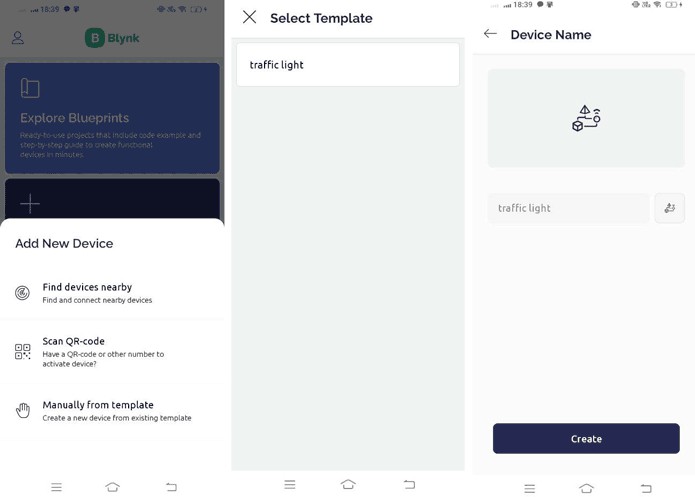
Step 2: Now open the Blynk IOT app and Log in with the credentials.



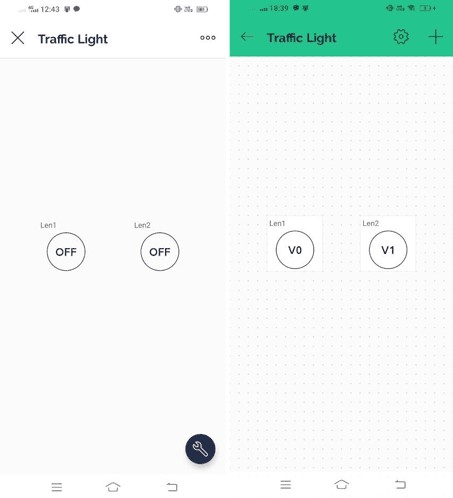
Step 3: After logging in click on “Add Device”.



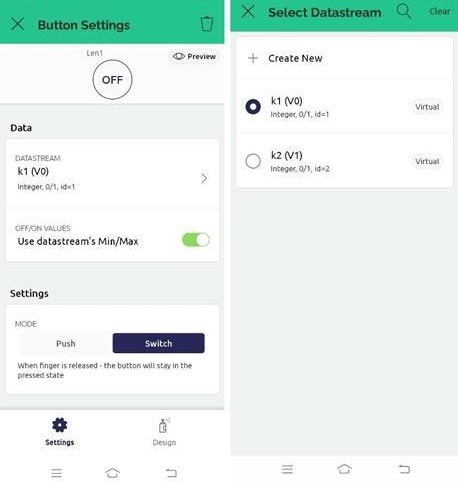
Step 4: Click “Manually from template”, select the device we already created, and then create.



Step 5: Now you are on the dashboard of the device then click on the tool button to enter in Developer Mode.



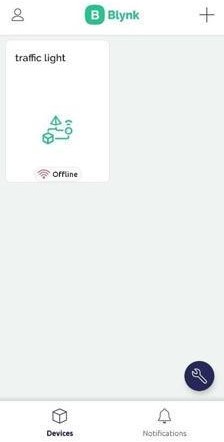
Step 6: Click on the button icons to assign values like Datastream and Mode. You can also change its design and looks.



Step 7: You Have successfully created a device dashboard.



Step 8: You can check the device status also.



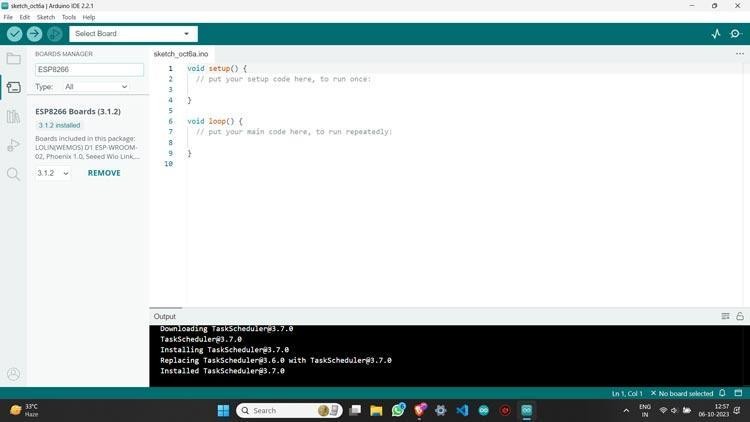
## Setup Arduino IDE to NodeMCU ESP8266 board to use Blynk

Now we have to set our Arduino IDE to program the NodeMCU ESP8266 board. Step 1: In Arduino IDE go to the “BOARDS MANAGER” and install ESP8266.

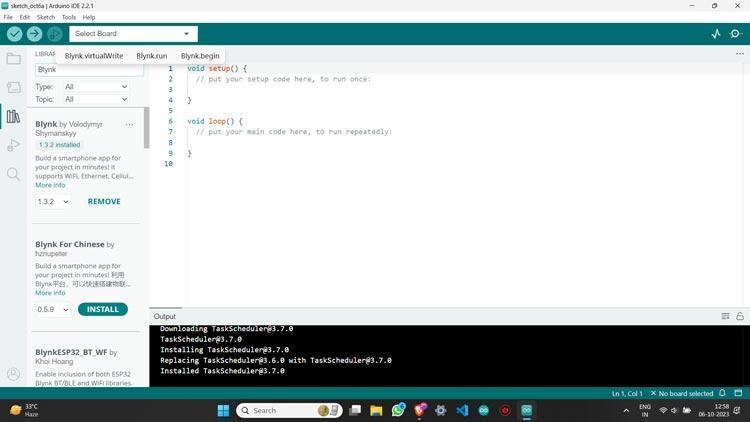
## Install Blynk Library in Arduino IDE

to set our **Arduino IDE** to program the **NodeMCU ESP8266 board**.

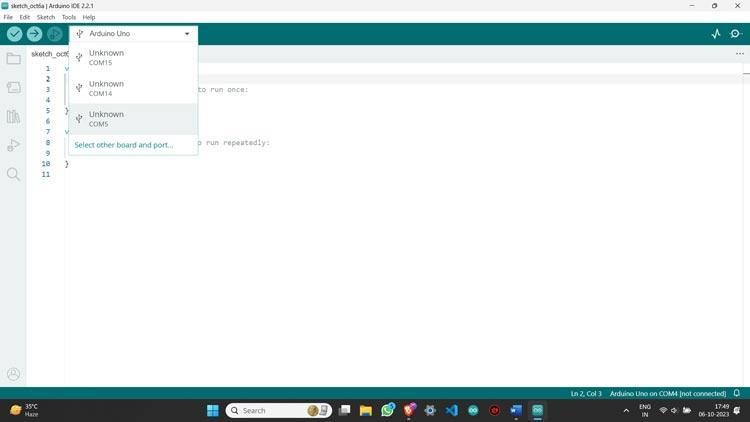
Step 1: In Arduino IDE go to the “**BOARDS MANAGER**” and install **ESP8266**.

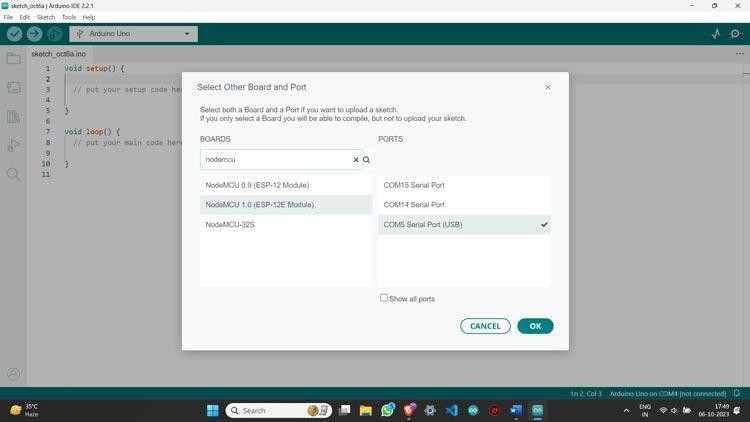


Step 2: Then go to the “**LIBRARY MANAGER**” and install the **Blynk library**.



Step 3: Plug your NodeMCU ESP8266 board into the PC using a **USB cable** and then **select your board and port**.





Step 4: Your Arduino IDE is now **ready** to program your board for using Blynk app

**CHAPTER 6:**

# 6. WORKING PRINCIPAL OF IDE

# Hydrophonics is a method of growing plants without soil, using mineral nutrient solutions in an aqueous solvent. The basic working principle of hydroponics revolves around providing plants with all the essential nutrients they need for growth directly in the water. Here's how it works:

# Nutrient Solution:

# Plants receive water mixed with essential nutrients, such as nitrogen, phosphorus, potassium, calcium, magnesium, and other trace elements. This solution provides all the minerals and nutrients that plants would normally absorb from soil.

# Water and Nutrient Delivery:

# The plant roots are submerged or in close contact with the nutrient solution, allowing them to absorb the water and nutrients directly. This can be done using different hydroponic systems, like deep water culture (DWC), nutrient film technique (NFT), or aeroponics.

# Support Structure:

# Since there is no soil to support the plant, hydroponics systems use inert mediums like perlite, coconut coir, or gravel to anchor the plants and provide some stability for their roots.

# Light:

# Since plants need light for photosynthesis, artificial grow lights (such as LED or fluorescent lights) are often used in indoor hydroponic systems, ensuring plants receive the proper spectrum and intensity of light.

# Environmental Control:

# Hydroponic systems often require controlled environments with optimal temperature, humidity, and CO2 levels to promote healthy plant growth. This is particularly important in indoor or vertical farming systems.

# .Root System:

# The plant roots are suspended in the nutrient solution, directly absorbing the necessary nutrients and water.

# Plants are supported in a variety of media, such as inert materials like rockwool, perlite, or coconut coir, which provide structural support and help to hold the nutrient solution.

# Pumping and Circulation:

# The pump circulates the nutrient solution from the reservoir to the growing chamber.

# Plant Uptake:

# Plants absorb the nutrients and water from the solution through their roots.

# Aeration and Oxygenation:

# The aeration system provides oxygen to the roots of the plants.

# Drainage and Recycling:

# Excess nutrient solution is removed from the growing chamber and recycled back to the reservoir.

# Monitoring and Control:

# The system is monitored and controlled to maintain optimal pH, temperature, and nutrient levels.

# This method allows for efficient water and nutrient usage, reduces the risk of soil-borne diseases, and allows for greater control over plant growth and development.

# In essence, hydroponics optimizes plant growth by eliminating soil-based constraints, providing more precise control over nutrient intake, water usage, and growing conditions.

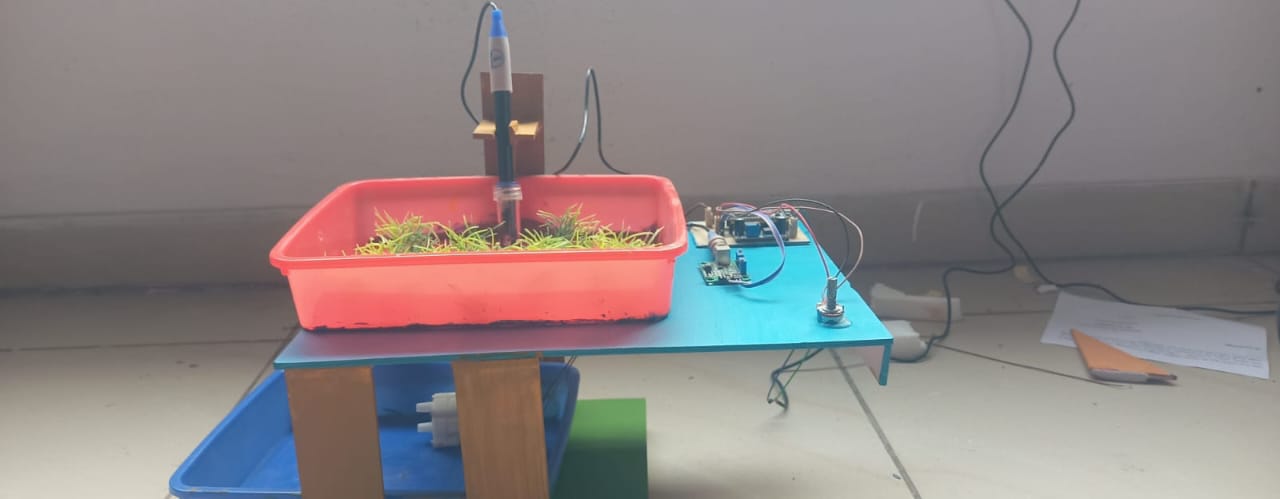
# CHAPTER 7:

# 7. OUTPUT & RESULTS

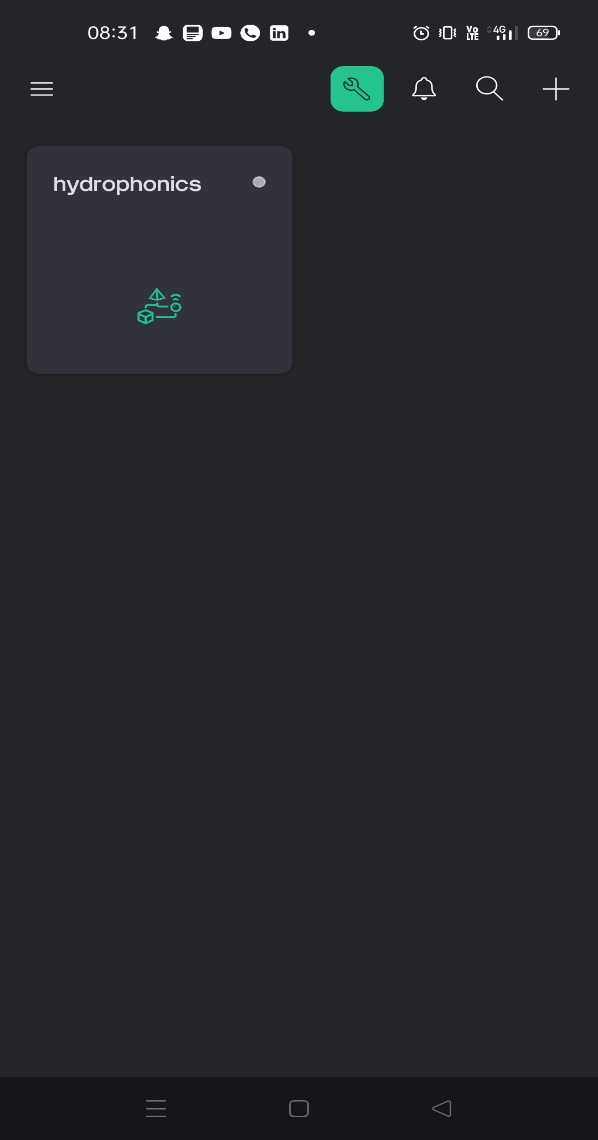
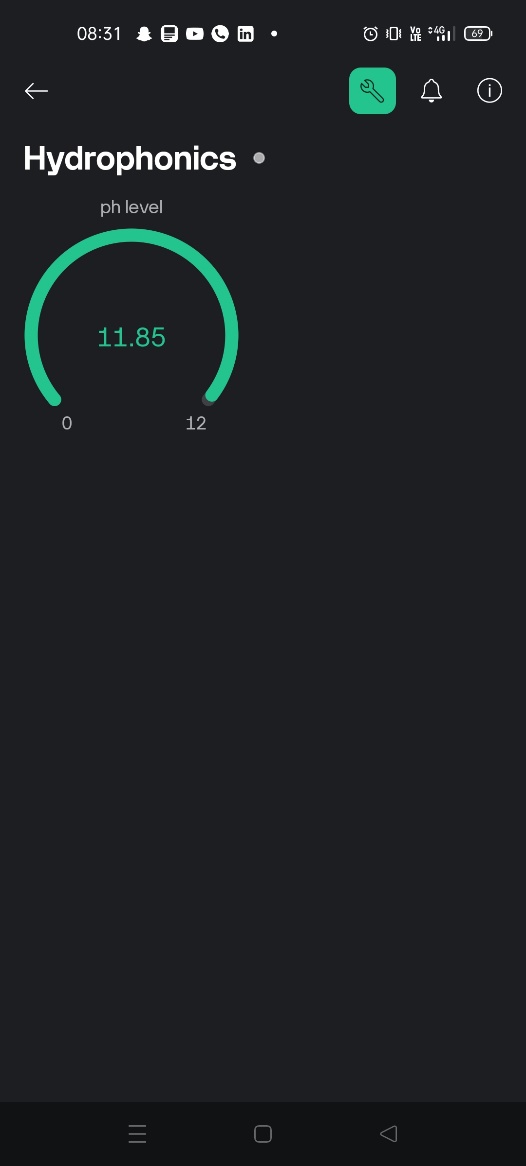
* 1. **Project Photos**

# 

# 

****

**Screenshots of Results**

** **

**CHAPTER 8:**

# 8.CONCLUSION

# The conclusion of a hydroponics system highlights the advantages and challenges of growing plants without soil. Hydroponics offers numerous benefits, such as increased crop yields, efficient use of water, faster plant growth, and the ability to grow in areas with poor soil quality or limited space. It also enables precise control over nutrient levels and environmental conditions, leading to more sustainable farming practices.

# However, hydroponic systems require a significant initial investment in equipment, continuous monitoring, and maintenance. They are also vulnerable to system failures, such as pump malfunctions or nutrient imbalances, which can quickly affect plant health.

# Overall, while hydroponics is an innovative and efficient method for growing plants, it requires careful management and investment to ensure its success and sustainability. As technology advances, hydroponics is becoming a more viable solution for addressing food security and sustainable agriculture in urban and resource-limited areas

# Conclusion

# The hydroponics system is a revolutionary method of growing plants that offers numerous benefits over traditional farming methods. By providing plants with a precise and controlled nutrient delivery system, hydroponics enables faster growth rates, increased crop yields, and improved crop quality. Additionally, hydroponics conserves water, reduces land use, and eliminates soil erosion.

# Key Benefits

# 1. \*Water Efficiency\*: Hydroponics uses significantly less water than traditional farming methods.

# 2. \*Increased Crop Yield\*: Hydroponics can increase crop yields by up to 30% due to precise nutrient delivery.

# 3. \*Improved Crop Quality\*: Hydroponics allows for precise control over nutrient delivery, resulting in higher-quality crops.

# 4. \*Reduced Land Use\*: Hydroponics can be used in small or irregularly shaped spaces, making it ideal for urban areas.

# 5. \*Year-Round Production\*: Hydroponics enables year-round production, regardless of weather conditions or seasonal changes.

# Future of Hydroponics

# As the global population continues to grow, hydroponics is poised to play a vital role inmeeting the increasing demand for food. With advancements in technology and the development of more efficient systems, hydroponics is likely to become even more widespread and accessible.

# Recommendations

# 1. \*Increased Research and Development\*: Continued research and development are needed to improve hydroponic systems and make them more accessible to small-scale farmers.

# 2. \*Government Support\*: Governments should provide support and incentives for farmers to adopt hydroponic systems.

# 3. \*Public Education\*: Public education campaigns should be launched to raise awareness about the benefits of hydroponics and promote its adoption.

# Final Thoughts

# The hydroponics system is a game-changer for modern agriculture, offering a sustainable, efficient, and productive way to grow crops. As the world continues to evolve, hydroponics is likely to play an increasingly important role in feeding the global population.

**CHAPTER-9**

**9. FUTURE SCOPE**

# The future scope of hydroponics systems is promising and holds significant potential due to several factors such as the growing global population, environmental challenges, and the need for sustainable agriculture. Here are some key aspects of the future scope:

# 1. Sustainable Agriculture

# Water Conservation: Hydroponics uses significantly less water than traditional farming methods. As freshwater becomes increasingly scarce, hydroponic systems offer an efficient solution for growing crops with minimal water usage.

# Soil Conservation: With the degradation of agricultural land and loss of arable soil, hydroponics can be a viable alternative to soil-based farming, enabling agriculture to continue without harming the environment.

# 2. Urban Farming and Vertical Farming

# Urbanization: As more people move to cities, space for traditional farming becomes limited. Hydroponic systems are ideal for urban environments where space is scarce. Vertical farming, using hydroponics, can turn buildings, rooftops, and warehouses into food production hubs.

# Localized Food Production: Hydroponics can provide fresh, local produce in urban areas, reducing the need for long-distance transportation, which lowers carbon emissions and transportation costs.

# 3. Global Food Security

# Hydroponics can contribute to addressing food security by allowing food production in regions with poor soil quality, arid conditions, or areas affected by climate change. Countries facing food supply challenges could adopt hydroponics to reduce dependency on imported food and increase local production.

# 4. Integration with Smart Technologies

# Automation: The use of sensors, IoT devices, and artificial intelligence can help optimize hydroponic systems by monitoring variables like nutrient levels, pH, temperature, and humidity in real-time. This integration can make the systems more efficient and reduce the need for human intervention.

# AI and Machine Learning: These technologies can be used to predict optimal growing conditions, detect diseases, and improve crop yields, further enhancing the viability of hydroponics in large-scale farming.

# 5. Higher Crop Yields and Diverse Crop Types

# Hydroponics allows for year-round crop production and better control over growing conditions, leading to higher yields. With continued research, it is expected that more diverse crops, including fruits and vegetables, will be grown in hydroponic systems, even in non-traditional growing areas.

# 6. Integration with Aquaponics

# Aquaponics, which combines hydroponics with aquaculture (raising fish), is gaining traction as a way to create sustainable food systems. This integrated approach can provide both fish and plant crops, contributing to more diverse food production and reducing the environmental impact.

# 7. Economic Opportunities

# Job Creation: The rise of hydroponics systems will create new jobs in the agricultural technology sector, from system design and maintenance to data analysis and farming.

# Entrepreneurship: Smaller-scale, decentralized hydroponic farming systems can allow for new business opportunities in local food production, catering to niche markets, and offering fresh produce to urban populations.

# 8. Research and Innovation

# Ongoing research into improving hydroponic systems will drive innovation. Advances in nutrient solutions, system designs, and energy-efficient technologies will lower costs and increase the accessibility of hydroponic farming.

# 9. Energy-Efficient and Eco-Friendly

# With increasing awareness of sustainability, hydroponics systems are expected to be more energy-efficient. Use of renewable energy sources like solar power could make hydroponics systems even more eco-friendly, making them a key component of green urban infrastructures.

**CHAPTER 10:**

# 10.FINAL CODE & RESOURCE

The final code for the IoT-based radar project that integrates the ultrasonic sensor with the ESP8266, rotating the servo to scan the surroundings, and sending data to the Blynk platform. Additionally, resources such as libraries and references are provided for further development.

## 10.1CODE

## #define PH\_PIN 36 // Define the GPIO pin where the pH sensor is connected

## #define dummyPh\_pin 34

## #define motor 21

## void setup() {

## Serial.begin(115200); // Initialize Serial Monitor at 115200 baud rate

## pinMode(motor, OUTPUT);

## digitalWrite(motor, 0);

## delay(1000); // Wait for the Serial Monitor to initialize

## }

## void loop() {

## int analogValue = analogRead(PH\_PIN); // Read the analog value from the pH sensor

## int analogValue2 = analogRead(dummyPh\_pin);

## // Convert analog value to voltage (assuming ESP32 has 3.3V ADC reference)

## float voltage = analogValue \* (3.3 / 4095.0);

## float dummyPh = analogValue2 \* (11.85 / 4095.0);

## // Convert voltage to pH (this conversion depends on the specific sensor calibration)

## // You may need to calibrate based on your pH sensor's datasheet for accuracy.

## float pHValue = 1.9 \* voltage; // Example conversion factor; adjust as needed.

## Serial.print("Analog Value: ");

## Serial.print(analogValue);

## Serial.print(" | Voltage: ");

## Serial.print(voltage, 2);

## Serial.print(" V | pH Value: ");

## Serial.println(pHValue, 2);

## Serial.print(" V | pH DValue: ");

## Serial.println(dummyPh, 2);

## if (dummyPh < 5.50) {

## digitalWrite(motor, 1);

## } else {

## digitalWrite(motor, 0);

## }

## delay(1000); // Delay 1 second between readings

## }

## 10.2Resources:

* + 1. **Libraries**:
       - **Servo.h**: Library for controlling the servo motor.
       - **ESP8266WiFi.h**: Handles Wi-Fi communication for NodeMCU.
       - **BlynkSimpleEsp8266.h**: Blynk library to interface with the Blynk IoT platform.

## Hardware Components:

* + - * **NodeMCU (ESP8266)**: Microcontroller for Wi-Fi communication.
      * **Ultrasonic Sensor (HC-SR04)**: Used for distance measurement.
      * **Servo Motor (SG90 or similar)**: Controls the sensor's rotation.
      * **Power Supply**: Ensure the NodeMCU has a stable 5V power supply.

## Blynk Platform:

* + - * Virtual pins **V1** and **V2** are used to send the servo's angle and distance data to the Blynk app for real-time monitoring.

## Tools:

* + - * **Arduino IDE**: For writing and uploading code to the NodeMCU.
      * **Blynk App**: To create a user-friendly dashboard for displaying the radar's data.