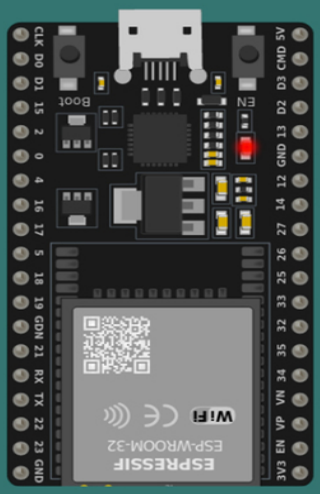
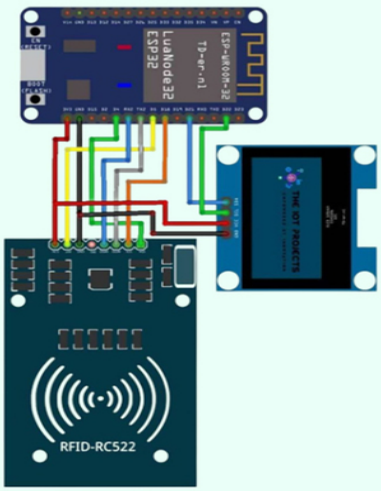
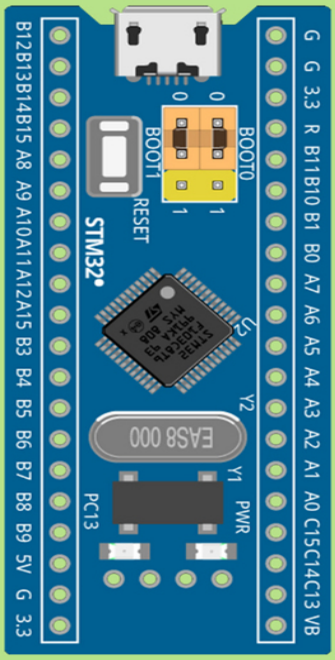






THE COMPLETE GUIDE TO ESP32 AND ARDUINO FOR IOT

Unleash the Power of the Internet of Things Build Connected Devices and Automate Your World

THE COMPLETE GUIDE TO ESP32 AND ARDUINO FOR IOT

Unleash the Power of the Internet of Things Build Connected Devices and Automate Your World

By

Roronoa Hatake

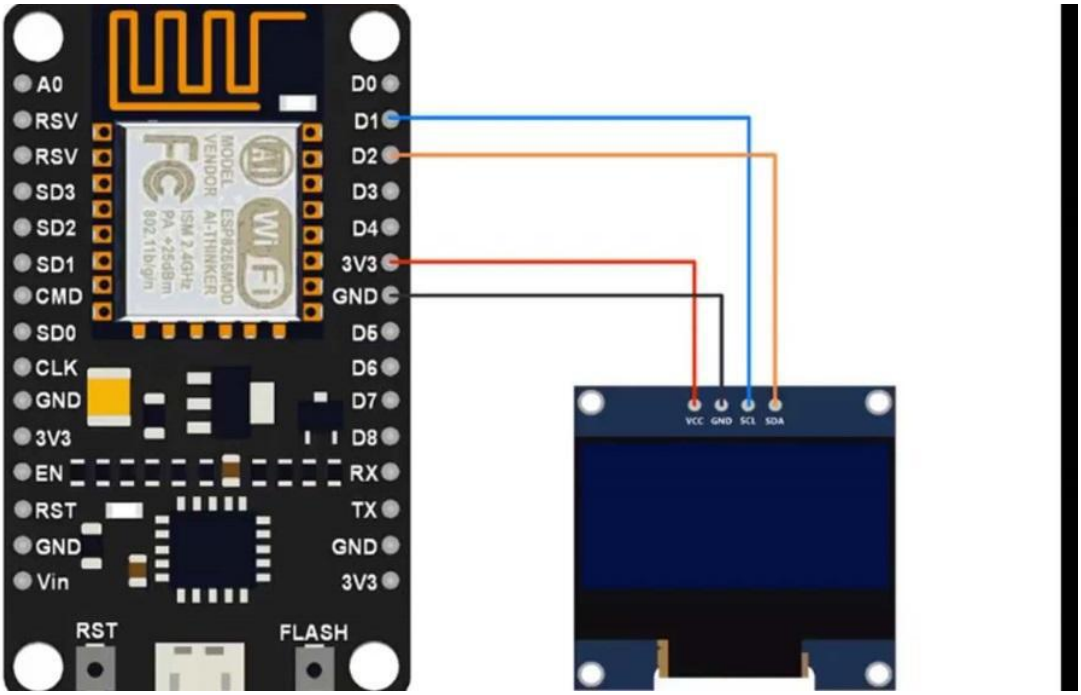
TABLE OF CONTENTS

LORA BASED REMOTE APPLIANCES CONTROL PROJECT
LORA BASED WIRELESS WEATHER STATION MONITORING SYSTEM
LORA SX1278_76 _ ESP8266 TRANSMITTER RECEIVER
MQTT WIND WEATHER STATION PROJECT USING GSM
POWERFUL ALTERNATIVE TO ESP32 CAM
SENDING SENSOR DATA WIRELESSLY WITH LORA ARDUINO
SHELLY IOT RGB LED STRIP CONTROLLER VIA ANDROID
SONOFF NSPANEL SMART SCENE WALL SWITCH DEMO
TESTING SX1276 868-915MHZ WITH STM32 MICROCONTROLLER
USING RASPBERRY PI PICO W WITH MICROPYTHON CODE
TOUCH SCREEN CAMERA PROJECT GAME SLIDE SHOWER
USING ARDUINO IOT CLOUD WITH ESP8266
VOICE BASED HOME AUTOMATION WITH NODEMCU AND ALEXA
WS2812B NEOPIXEL RGB LED STRIP CONTROL VIA BLYNK

IOT BASED ANALOG DIGITAL OLED INTERNET CLOCK

Today, we are back to another project, and it's about analog clocks or digital clocks using no MCU. an OLED display. The time is actually downloaded from the internet, and there is no need for the RTC model. So you can see This is an OLED border loop, ground VCC SD and SCL pins, and we have connected SD and SLC into digital pins D1 and D2 are not MCU.

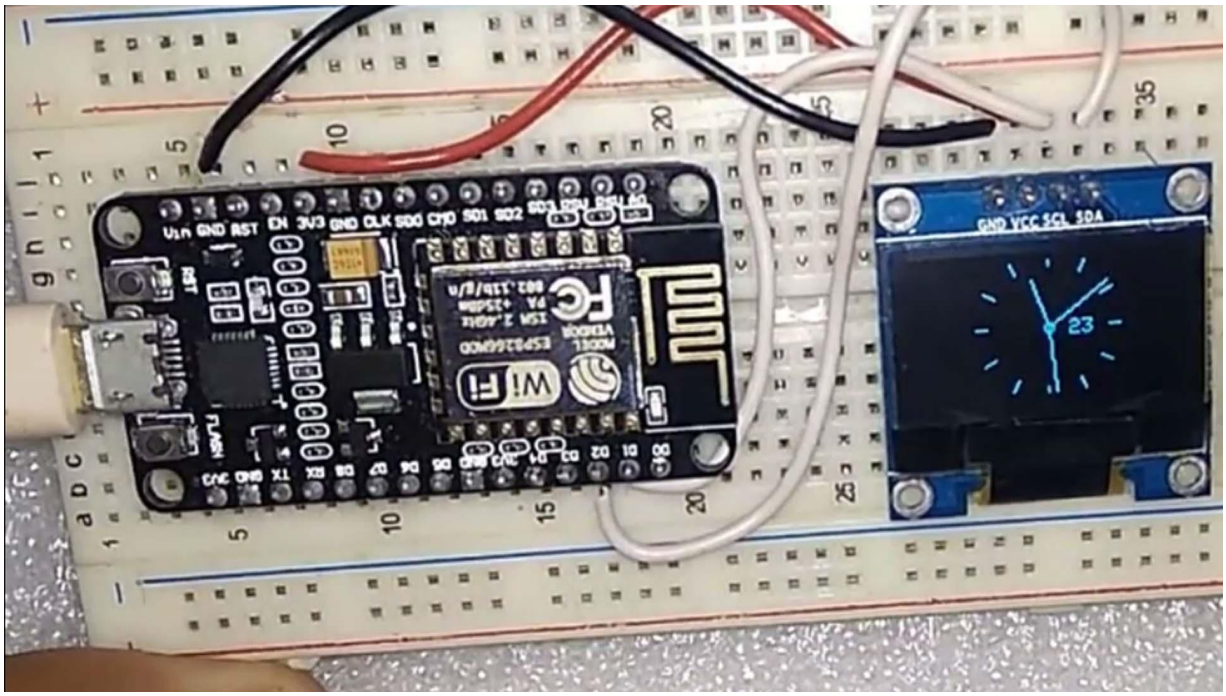
So this is the connection. D1 and D2 are connected to SELL at STA. And similarly, we are supplying 3.3 volt power supply and ground. Now, once the power is supplied, so also, what happens? So it is connecting to WiFi, and it will download the IP and it will display at 8 o'clock.



if you go for a digital plug. And after some time, it will automatically detect the time zone, and it will detect the time. So today, it's February 23. So, it's displaying February 23. When you reset it, it will go back to 8 PM.

or sorry, 8 AM, 1970, and after some time, it will automatically detect the time, and it will change. Same thing will be happening with the other clock as well. In case of another clock, it will also be displaying it. So the time is automatically uploaded. So you can see it's currently around 11:30 and you can see there are 23 written.

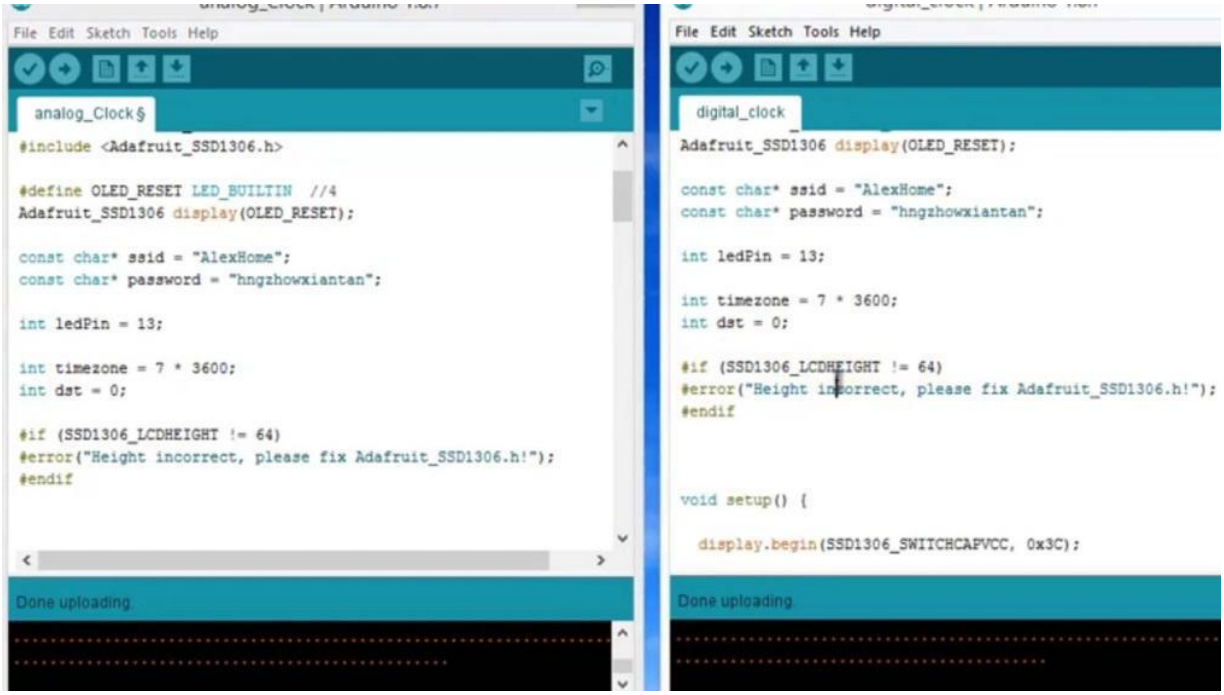
Now, this 23 is a date. That is , the 23rd of February. So this is how you can make an analog and digital plug. So there is no need for the RTC model. This is a sketch of both analog and digital.



Both are similar to each other. So we have included some of the header files as you can see here, so just go through it. So, you can edit the SSID password. Wharton analog clock and in case of digital clock as well. So, here, you need to edit.

and now you can see a time zone. This is 7. The time zone is of some other countries, not of India. For India, the time is 5.5. That is high power for 30 minutes.

It is multiplied by 3600. So, same thing. We can sub digital as well. So, this is the I2c address of OLED decimal 3 into D. So, this is how that occurs.



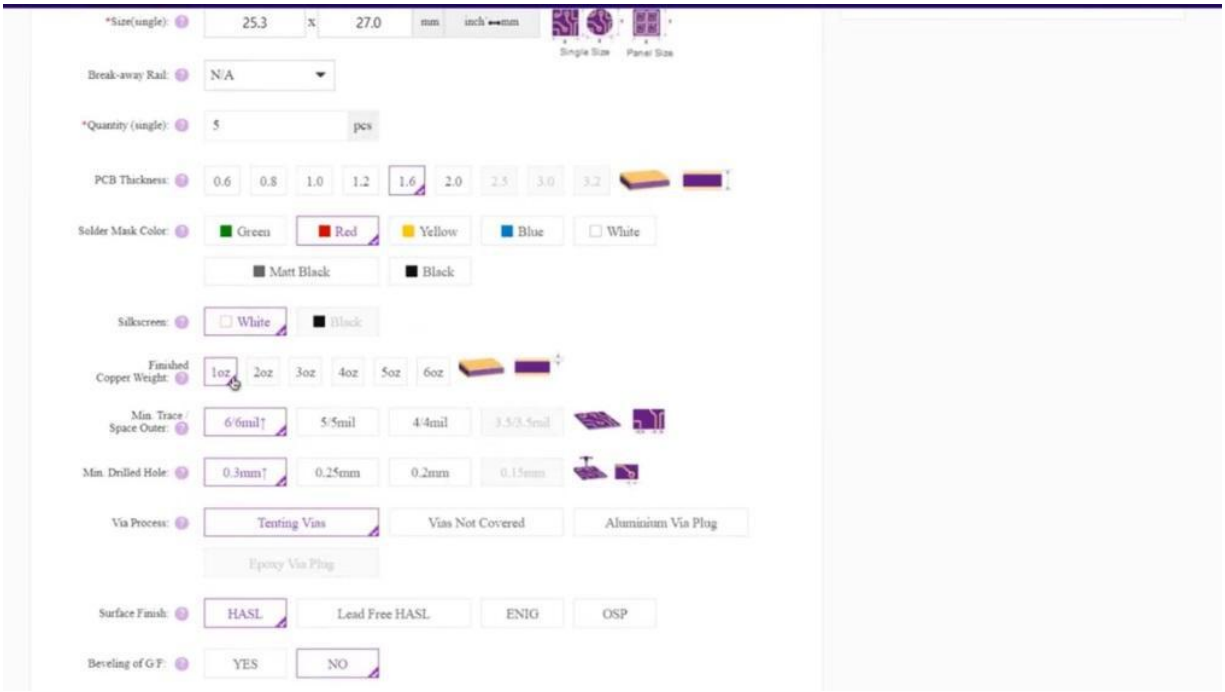
This is how the time is configured in the time zone to DST. DST daylight saving. Wudder and DP. LNG is a server.

From here, the time is downloaded, and this is used for creating o'clock. So, this is a minute holiday display. This is how our end is displayed. This is second hand. And, similarly, in the case of digital, you just need to display some particular numbers, so numbers are displayed.

DISCHARGING STATUS VOLTAGE MONITORING SYSTEM WITH ESP8266

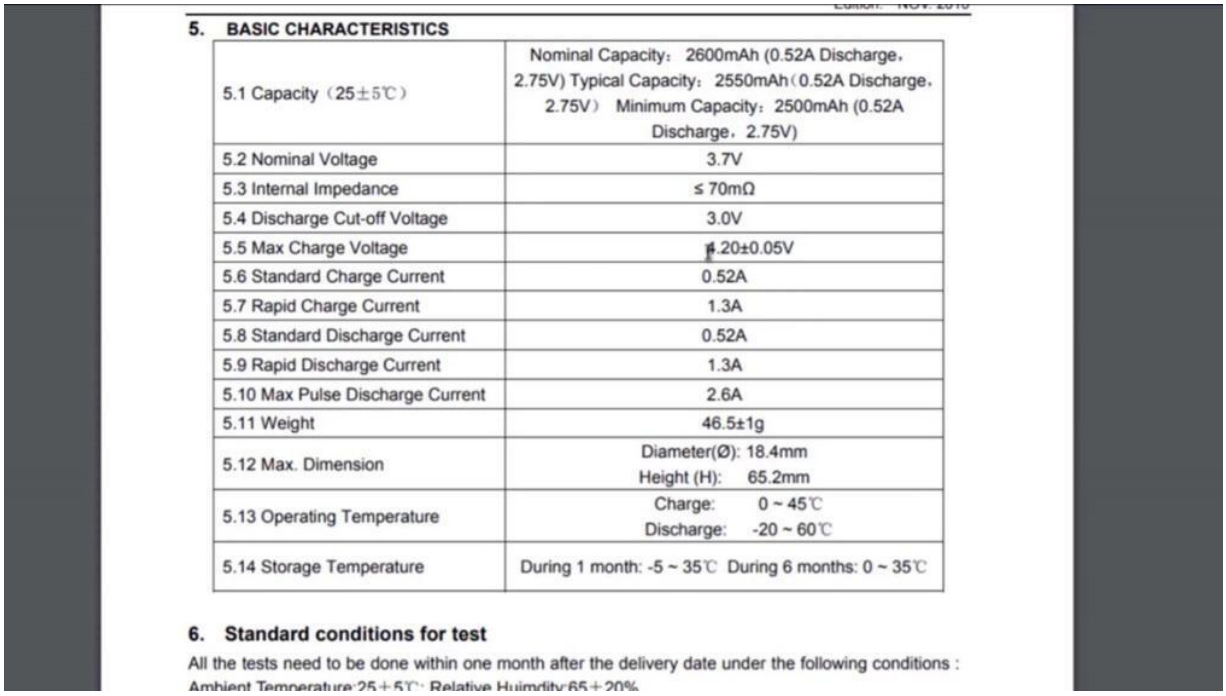
The battery is the most important component for any device as it powers the whole system, and it's important to monitor the voltage level of the battery as improper charging and discharging of a lithium battery may lead to a big safety issue. Due to the advancement in technology, Now the internet of things can be used to notify the manufacturer and users remotely regarding the battery status. They can check the battery status of the car's battery on their smartphones from anywhere in the world, and this is considered as one of the maintenance support provided by the manufacturer. So in this project, we will also build an IOT based battery monitoring system where you cannot only monitor the charging and discharging status of the battery. Here, we will use Wemos d 1 mini with ESP 8 to double 6 chip to send the battery status data to think speed cloud.

The thick speed will display the battery voltage along with the battery percentage in both the charging and discharging cases. So without any delay, let's get started with this interesting tutorial. This project is sponsored by my favorite PCB manufacturer company called Next PCB. They offer PCB board and PCB assembly services at the lowest affordable price. You can get trial PCB, 2 layer PCB, and full layer PCB with 3 PCB Assipping services up to a faster time of 24 hours.



All you need to do is visit nexvisibility.com and upload the Gerber file. Select vis a vis quantity, color, material type, and all the details. then place an order. You will get the high quality PCV within 48 hours. These are the few lithium ion batteries that I have been using for very long for many of my projects.

lithium ion batteries have a nominal voltage of 3.7 volt, but might have different capacities. This battery has a capacity of 1000 MH, whereas this bigger battery has a capacity of 1950 MH. There is a simple BMS circuit along with the autocot of the system. Some batteries have a nominal voltage of 3.7 volt. with a maximum voltage of 4.2 volt and a cutoff voltage of 3.4 volt or 3 volt.



5. **BASIC CHARACTERISTICS**

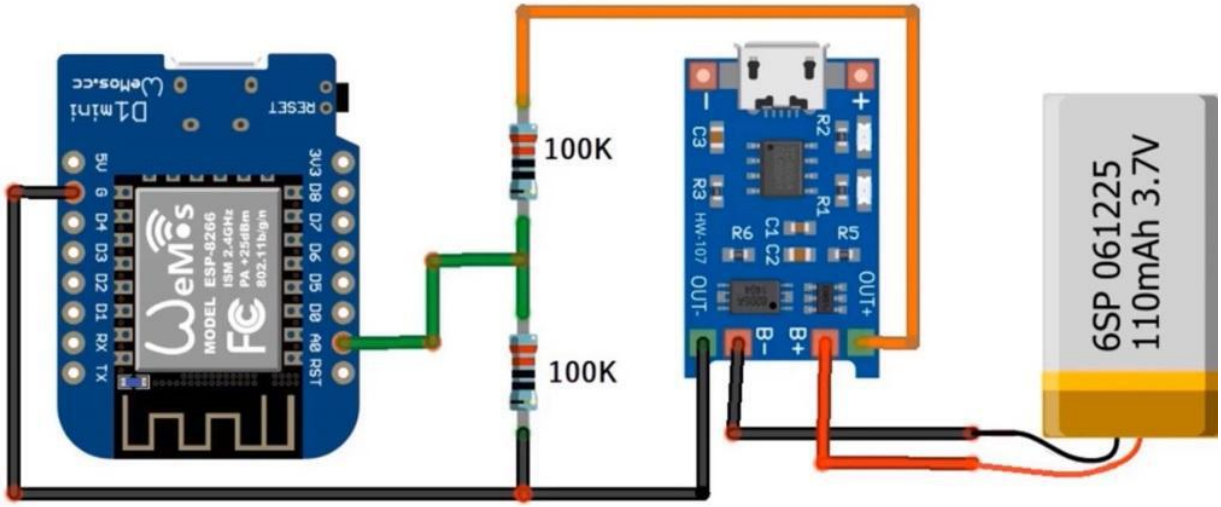
5.1 Capacity（25±5℃）	Nominal Capacity：2600mAh (0.52A Discharge, 2.75V) Typical Capacity：2550mAh（0.52A Discharge, 2.75V） Minimum Capacity：2500mAh (0.52A Discharge，2.75V)
5.2 Nominal Voltage	3.7V
5.3 Internal Impedance	≤ 70mΩ
5.4 Discharge Cut-off Voltage	3.0V
5.5 Max Charge Voltage	4.20±0.05V
5.6 Standard Charge Current	0.52A
5.7 Rapid Charge Current	1.3A
5.8 Standard Discharge Current	0.52A
5.9 Rapid Discharge Current	1.3A
5.10 Max Pulse Discharge Current	2.6A
5.11 Weight	46.5±1g
5.12 Max. Dimension	Diameter(Ø): 18.4mm Height (H): 65.2mm
5.13 Operating Temperature	Charge: 0 ~ 45℃ Discharge: -20 ~ 60℃
5.14 Storage Temperature	During 1 month: -5 ~ 35℃ During 6 months: 0 ~ 35℃

6. **Standard conditions for test**

All the tests need to be done within one month after the delivery date under the following conditions :

1 of the battery data sheets give different information as 3.7 volt nominal voltage and maximum charge voltage of 4.2 volt or slightly up. It also has a discharge cutoff voltage of 3 volt, but the battery I am using has a discharge cutoff voltage of 2.75 point on your around 2.8 volt. I will design a system to monitor this battery voltage along with charging and discharging status. So let's see the circuit now. I'm using Wemos d 1 mini, which has an ESPN 8 to double 6 Wi Fi enabled chip. This chip connects to the Wi Fi network and uploads that regularly to the server.

I use the TP4056 module to charge the battery as it's best suited for battery management applications. The SPA to double 6 chip can only support the input analog voltage of 3.3 volt, but battery voltage goes up to 4.2 volt. Hence, we have to form a voltage divider network to lower down the input voltage. So we need to do the calculation first. The battery maximum voltage is 4.2 volt, and the cutoff voltage is 2.8 volt. Let's first step down the upper voltage level.

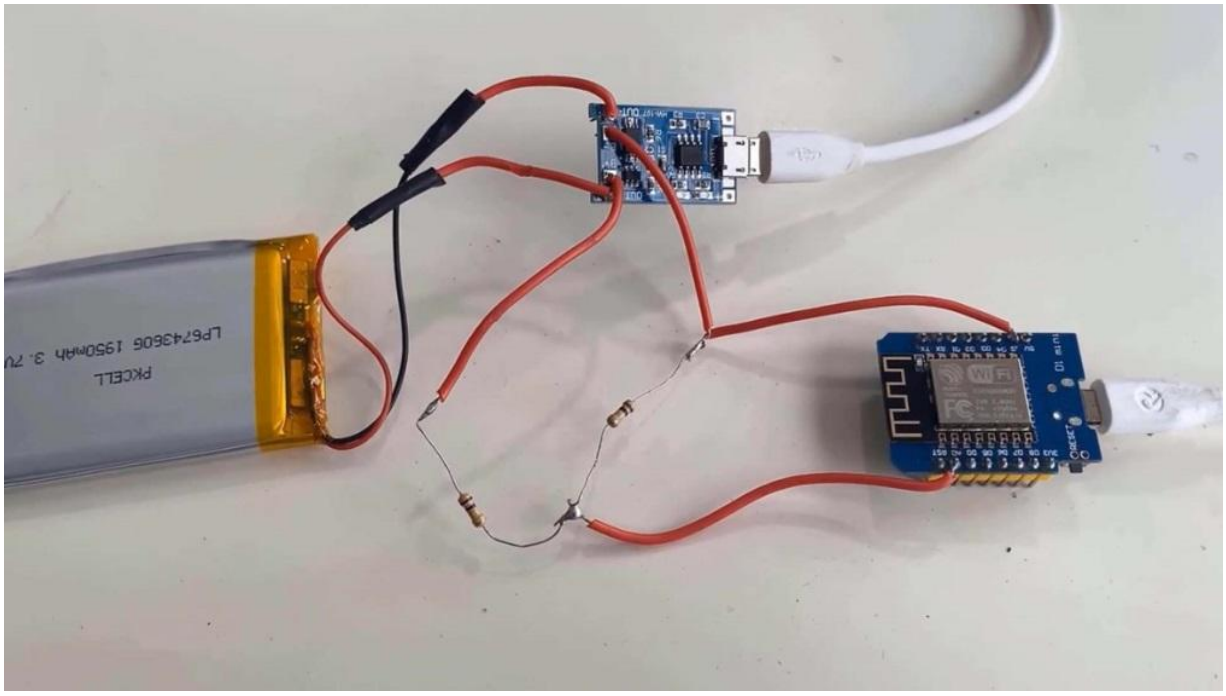


Circuit: IoT Based Battey Voltage Monitoring System

The source voltage is 4.2 volt, and there is a pair of 100 k resistors. This will give an output of 2.1 volt. Similarly, the minimum voltage is 2.8 volt as a car of voltage, which steps down to 1 0.4 volt using the same voltage divider network. Now both the upper and lower voltage is supported by ESPA to double 6 analog pins. I now used the general assembly to assemble the circuit.

Then I connected the micro USB data cable to both the WEMOS D1 mini board and TP 4056 module. I also connected the USB cable to my computer so that I can program the board. Here is the complete assembly of the project. This is the same connection that I saw in the circuit diagram. For now, I'm using a lithium and battery of 1950 MAH capacity.

Instead of using this poor connection, I recommend you use the PCB board to assemble the circuit. It's time to set up the teams now. For that, just visit the website thinspeak.com. It's the best and free platform for your IOT device. You can collect the data and visualize it in graphical format.



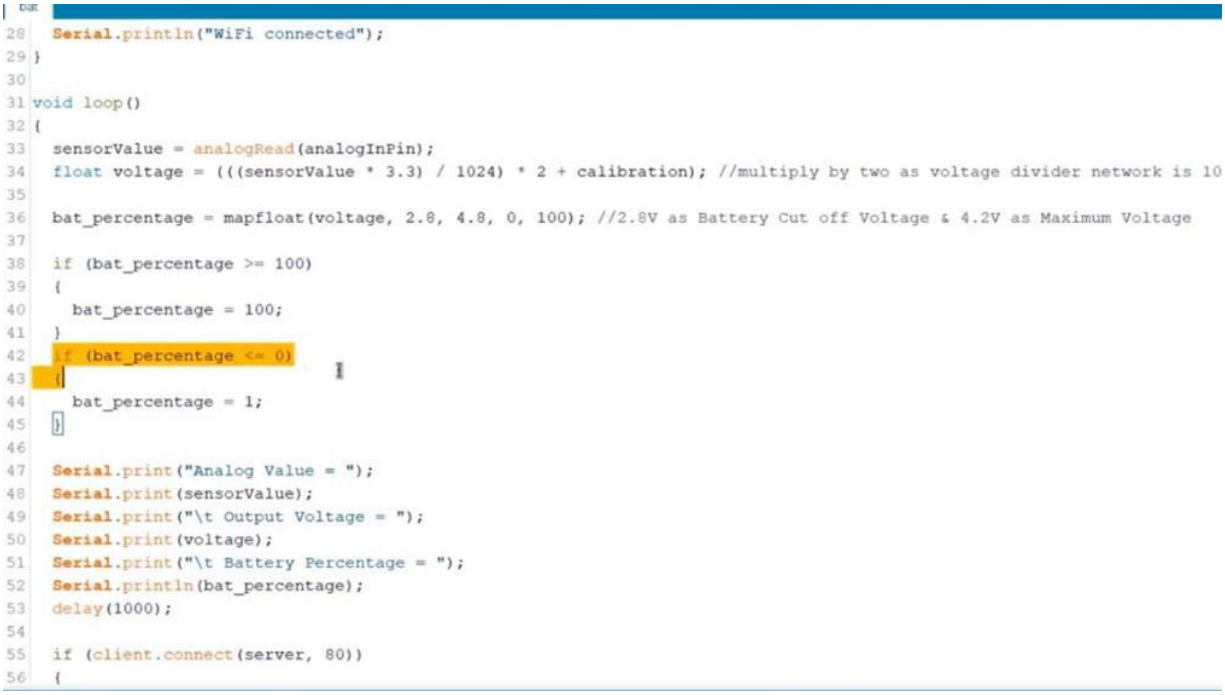
You need to sign up using your email ID and password. After that, you can create a new channel here. Fill the channel details like channel name and the fields that you want like I filled to fields, and then click on save the channel. So here you can see it filled with a graphical chart. From here, you need to copy your API key.

In the code part, We will use the inbuilt library for ESP 8 to double 6. From here, replace the API key that you copied earlier. They ingest the WiFi SSID and password. From here, since the calibration factor, you can get this value by taking the voltage displayed on the serial monitor as well as the voltage you get on the multimeter. Compare the values and get the calibration factor.

In the loop part of the code, We are reading the 10 bit analog value from the analog team of ESP 8 to double 6. We convert the sensor value to voltage using this equation. We are multiplying by 2 to get the actual voltage due to the voltage divider network. Then we add the calibration factor to get the correct voltage. Then we are mapping the battery voltage to convert the value into a person test.

We are using this condition so that the battery person does not go beyond 100% or negative. Then we print all these values on the serial monitor using the HTTP post method. We will send the battery voltage and battery percentage to the server. That's all from the code part. From the tools menu, select the Wemos d 1 mini board or node MCU 1.0 board if you are using the ESP 8 to double 6 based chips, then upload the code. Open the serial terminal.

The device will try connecting to the Wi Fi network. Currently, the battery voltage is around 3.61, which is the 40% charge. It also shows the analog value of around 5 or 5. Then the device will try uploading these values to the server after an interval of 15 seconds. You can set it to any time as per your requirement.



```
  Serial.println("WiFi connected");
}

void loop()
{
  sensorValue = analogRead(analogInPin);
  float voltage = (((sensorValue * 3.3) / 1024) * 2 + calibration); //multiply by two as voltage divider network is 10

  bat_percentage = mapfloat(voltage, 2.8, 4.8, 0, 100); //2.8V as Battery Cut off Voltage & 4.2V as Maximum Voltage

  if (bat_percentage >= 100)
  {
    bat_percentage = 100;
  }
  if (bat_percentage <= 0)
  {
    bat_percentage = 1;
  }

  Serial.print("Analog Value = ");
  Serial.print(sensorValue);
  Serial.print("\t Output Voltage = ");
  Serial.print(voltage);
  Serial.print("\t Battery Percentage = ");
  Serial.println(bat_percentage);
  delay(1000);

  if (client.connect(server, 80))
  {
```

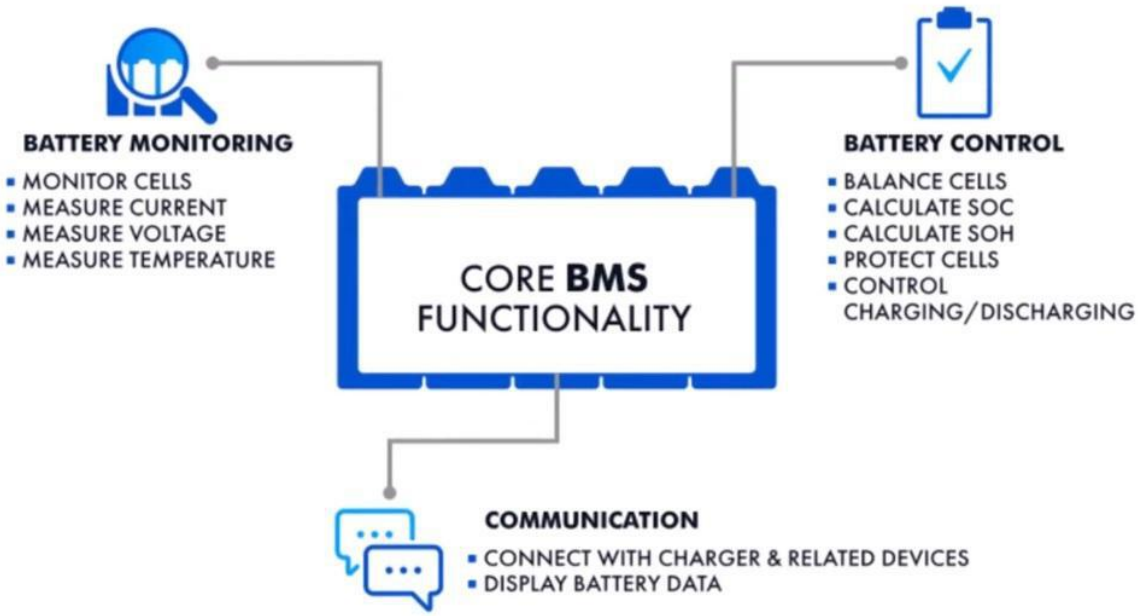
Go to a private view of things. So here's the battery voltage and battery percentage entry. The data gets uploaded frequently as long as the device is powered on and connected to the network. The battery charging or discharging curve along with the battery percentage is displayed on the dashboard. I will fast forward this project to 5 x to show you the quick demo of the project.

Here, you can see how the graph is rising as the battery is charging. So this is how a battery voltage status can be monitored from anywhere in the world using thingspeak cloud and ESP 8 to double 6 Wi Fi module.

IOT BASED BATTERY MONITORING SYSTEM DIY LIPO

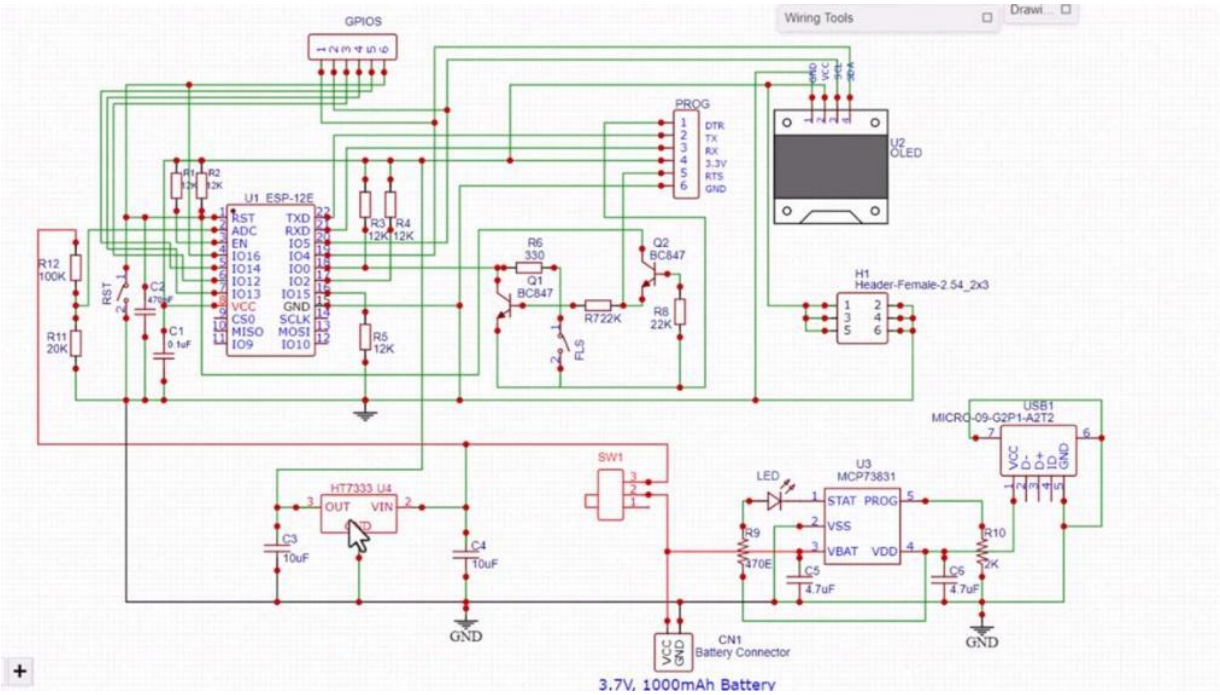
This project is a small shirt by PCV. This is a DIY lipo battery charger device with a battery management system. This device can charge any single cell lithium ion or lithium polymer battery. Apart from battery charging, the device has an ES be 82 double 6 raw chip. This tip is capable of connecting to the wifi network and sending the battery voltage and battery version test data to any IoT app.

In case you don't want to use the wifi feature, you can monitor the battery status on this tiny OLED display. The battery being a device primary powers requires constant voltage monitoring to prevent damage or system failure from improper charging discharging. This task is handled by the battery management system. The BMS keeps track of the battery voltage, current, temperature, and has an auto caught up system, ensuring safe uses of lithium ion or lithium polymer batteries. Previously, PMS manually monitored battery conditions and altered users through battery indicators.



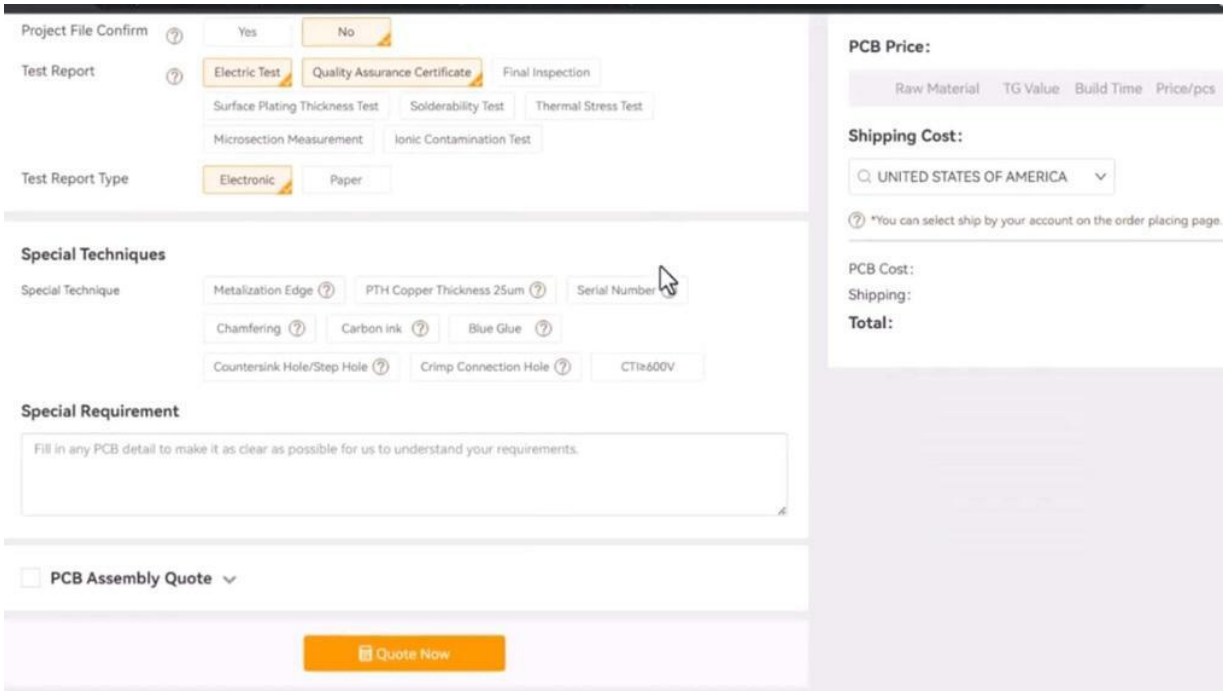
But with the advent of the internet of things, users can now receive remote notifications and check their battery status from anywhere using smartphones or computer dashboards. In this project, we designed our own PC and assembled ESP 82 double 6 raw chip, a low power LTO, and a battery charger chip. This battery charger chip has all BMS features. Given the device's low power consumption, it uses the battery power to transmit data to the things shoreboard. The things picked asphalt visually represents the battery charging and discharging status.

So let's build a LIPO battery charger with an IoT voltage monitoring system. Let us take a look at this design part of this project. Starting from here, it has a micro USB port for battery charging. The battery charging is managed by this IC called MCP 72831, which is a very efficient battery charging chip from microchip. This LED indicates battery is charging.



The 2.7 volt lithium ion battery can be connected here. This is a slight switch to turn on and off the entire circuit. Then we have a low power allele called ST70 3 from Holtec. This supplies 3.3 volt power to the entire system. The main important thing is this fold is the divider network registers value, which is 8 to the ESP 82 double 6 ADC pin, which calculates the battery voltage.

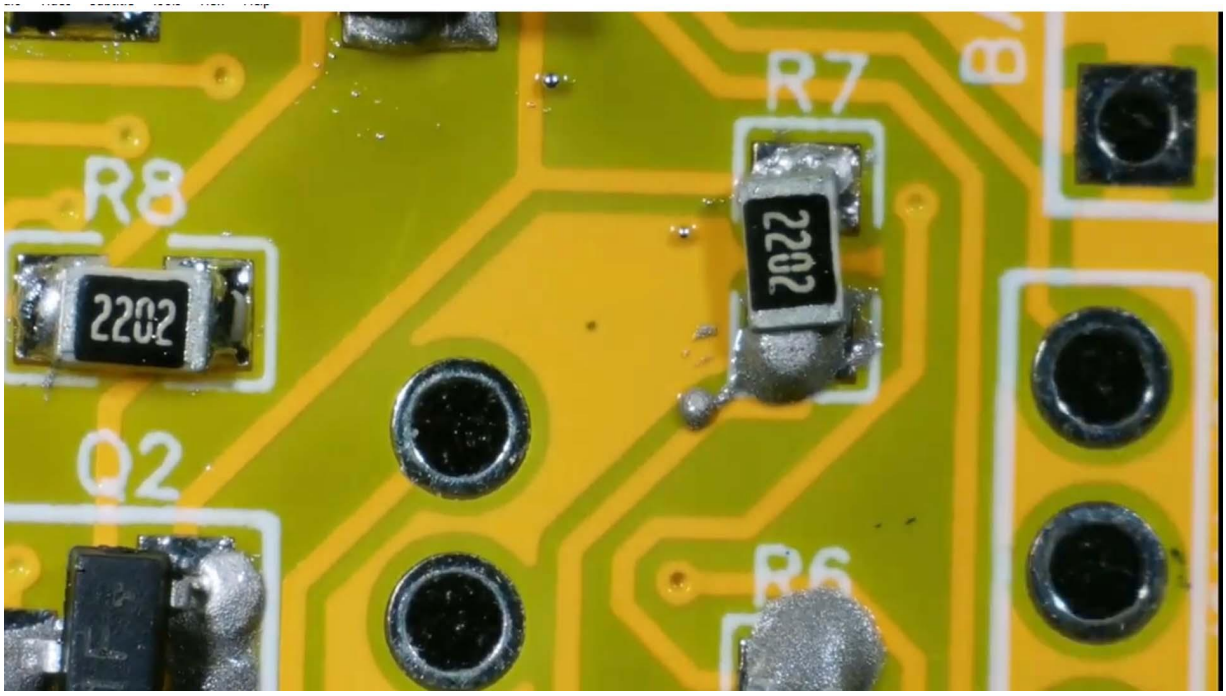
Then we have ESP 82 double 6 logic which controls the entire operations for battery management. Using these header pins, we can program the raw ESP 82 double 6 tip. The programming operation is handled by these 2 transistors which are automatic. There is no need to press any rushate or flash button. everything is automatically managed. These GPS ports are optional, you may use them to connect any digital sensors or peripheral This OLED display is also optional, in case you don't want to use the wifi feature.



After designing the schematic, I converted the commits to PCV. The PCV is very tiny and the components needed to be soldered on both sides. This is the theory of the PCV. The PCV looks awesome. Now, I generated the Java file.

Now, it's time to order the pcp. So I visited all pcp, which is the official sponsor of the project as well. You can get your trial PCV at only $1 here. It is very cheap compared to all other PCV manufacturers. I uploaded the Java file and filled in the details like material type, dimensions, quantity, thickness, solder, mask color, and other required parameters.

And then I clicked on a quote now. Here, you see the price is only $1. Now I selected my country of shipment and placed the order. Now, after 5 days, I received this recipe. Look at this recipe quality.



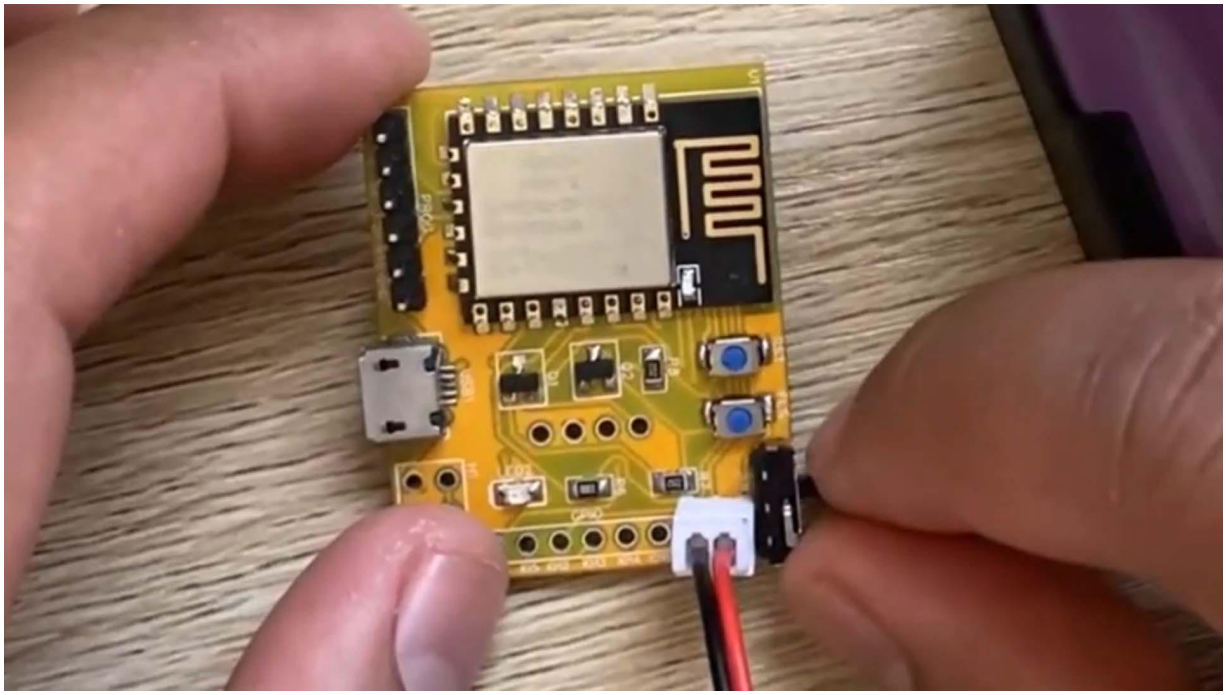
It is very premium and has a perfect design for my project. If you want to order the PCV $81, check the first link in the description. Now, it is time to solder all the components on this PCP board. So I collected resistors and capacitors from my resistors' capacitors book. Then I manage transistors, I see, battery holder, and all other required components.

I have posted the bill of materials on my web with a footprint. Then, I started soldering all the components. On the front side, frost soldered all the assembly components like registers, capacitors, changes search, LED, post buttons, and micro USB port. Be careful about the SMD LED polarity. bested in the proper direction.

After shortening all these, you can solder the ESP 82 double 6 proxy. On the backside, you can solder all the registers, capacity charts, MCP 73821 IC ST 7, triple 3, I see. After soldering all the assembly components, you can solder the through hole components like slight swiss, battery connector pins, and male female header pin. So, everything is soldered now. The board looks awesome and perfect for this project as I soldered everything perfectly.

So it's time to test this hardware now. To do that, frost connects the 3.7 volt, lithium, ion battery to the battery port. Slide the switch down on the system. When the switch is red, the LED on ESP HD 2 double 6 tip blinks. Voice indicates the hardware is fine.

To test the battery charging, ensure the micro USB charger to micro USB port. The red LED on the turns on immediately indicating the battery charging system is working fine. To test whether you are able to upload the code to the ESP 82 double 6th trip or not, connect the FTTA module to the board using the paints mentioned in the PCB. Make sure to connect the RTS ping from the side of the FTDA module as well. Connect the FTDA port to your computer and open the audio blinks case.



From the board list, select generate ESP 82 double 6 port and also select the COM port. Now hit the upward button to upload the code. After successful uploading, the audio ID will show the following messages. This means the code is successfully uploaded and you are ready to go. On the other hand, the onboard LED on the ESP 82 double 6 will turn on.

One second. So, the testing part is done. Now, let's move to the main project that is DIY Lipo battery charger with IoT voltage associate monitoring system. All we need is the code for a battery charger voltage testing and sending it to the things quicker. In order to monitor the battery data on Think Speak Sharver, you first need to set up the Think Speak.

To set up the Think Speak Sharver, visit Think speak.com. Create an account or simply sign in if you created the account earlier. Then create a new channel with the following details. Then go to the API section of the dashboard and copy the API key. This APA code will be used in the code to receive the data from ESP 82 double 6 hardware.

Now, let's move to the coding part. The script is for an ESP 82 double 6 microcontroller to read battery voltage, calculate the percentage, and then send this information to a Think Speak Channel using a wifi connection. From the following lines, change the wifi SSID password and things speed API key. We are reading the battery voltage using the ADC pin of ESP 82 double 60. The battery maximum voltage is 4.2 volt, and the quarter voltage is 2.8 volt.

To convert the ADC value into voltage, we multiplied ADC value by a factor of this number. This factor is an experimental value which we got from the voltage divider network by manual testing. The rest is explained in our website article. Now, let's upload this chord. After uploading the code, it is time to test the Lipo battery charger with an online voltage charts monitoring system using ESP 82 double 6 on Think Speak Server.

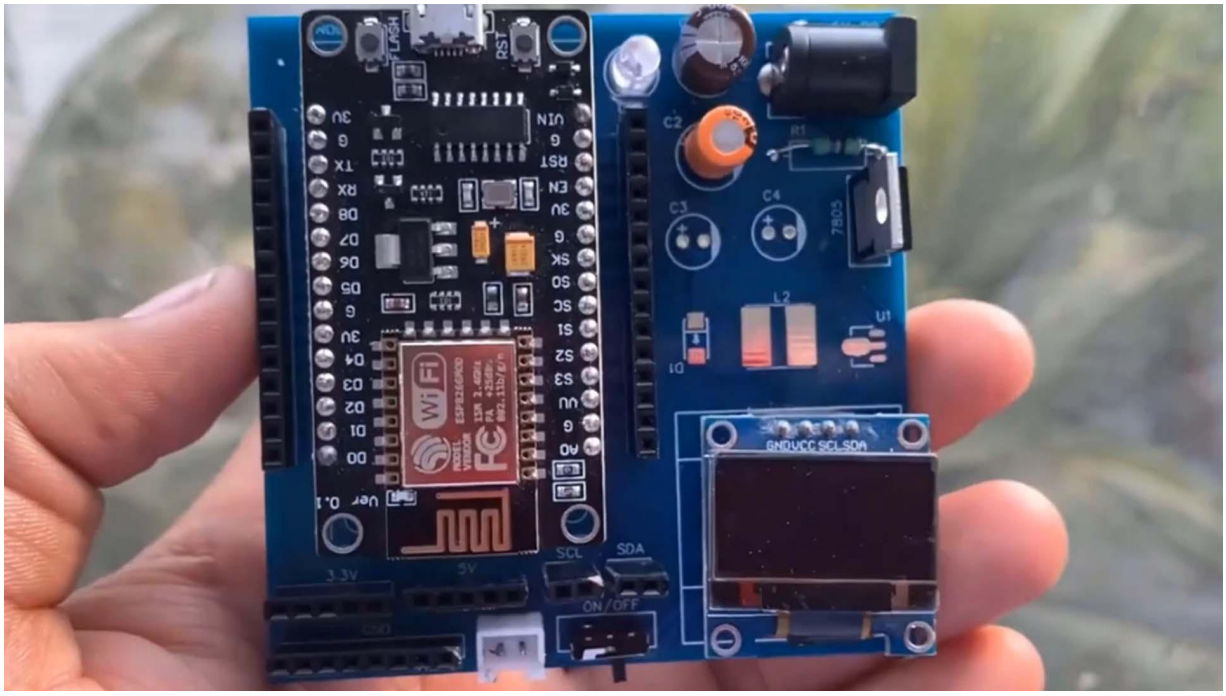
Open the serial monitor. The serial monitor will display the ADC value battery voltage and battery SOC charts. The ESP 82 double 6 will simultaneously connect to the wifi network and will upload the data to the things speak server. To view the data on thingspeak server, go to the private view section on the things speak dashboard. Here is the charging graph that shows the rising voltage for time as the battery was put in a charging state.

It just took 30 minutes to attend full charging, as I was charging the device using a 500 MRs LIPO battery. In order to see the discharge curve, I had to leave the device turned on for the whole night. As the device is low, the discharging process is slow. Here is the discharge graph on the big server. You may set up the widgets and graphical display on the big server according to your wits. Apart from monitoring the voltage alignment, what if you do not have a Wi Fi connection?

In this case, you may solve it at 0.96 inches. I square over the LED display on the back side and right go to display battery voltage and it is on OLED. In conclusion, we have successfully created our DIY Lipo battery charger, That integrates an IoT voltage monitoring system using an ESP HD to double export. Our custom PCV design is awesome.

IOT BASED BIDIRECTIONAL VISITOR COUNTER USING ESP8266 MQTT

I explained how you make a bidirectional visitor counter using an arduino, ir sensor, and OLED display. The device displaced the number of incoming, outgoing, and total number of current visitors. It also had an automatic light control system which controls the light on the basis of visitors presence, but the drawback of the system was there was no internet connectivity with the system because of this, we are unable to send the data to any server. But in this proposed system, we replaced audio and anode with the Wi Fi module, which has Wi Fi connectivity. Using the Wi Fi connection, the system uploads the data to the server regularly.

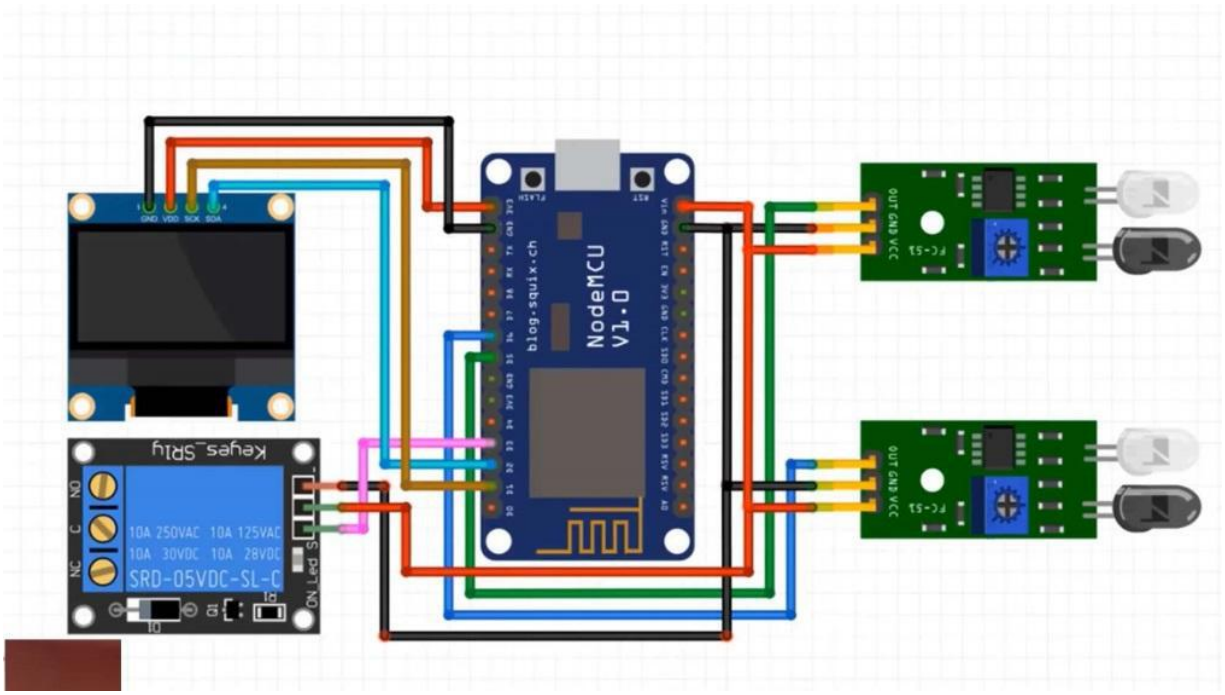


To monitor the data online, I use Ubidot's platform. And using the MQTT connection, I send the visitor's data to the UBROS cloud. Using EUROS MQTT connection, you can view the incoming outgoing and current visitors online from any part of the world. The dashboard also shows the status of the light whether it's turned on or not. So without getting any delay, let's see how you can build this entire system.

The PCB board used in this project is sponsored by PCV. Next, PCB is one of the largest PCB manufacturer companies in China. You can order the PCB services as well as PCB SMB services at an affordable price. They also have the next CFM PCD analysis tool. Using the next CFM, you can directly analyze the manufacturability of the PCB.

download the tool from the description link. Alright. For making an IoT visitor counter project, we need the following components. We need a pair of infrared sensors. These are used as obstacle detectors.

A simple 0.96 inch SSD1306 OLED display module with I Square C pins. a five volt single channel relay module for a light control system. And the most important is the ESP 8 to double 6, Wi Fi module. You can either use the low end node MCU board or you can go with the Amica node MCU board. Let's see the circuit now.



I used fetching software to design the schematic. All the passive components like IR sensors, relay module, and OLED display ARCON get to note MCU GPI openings. You can use this schematic to assemble the circuit on the breadboard, but I prefer a custom PCB. For that, I used easy EDA to design the schematic. This schematic contains the power supply unit along with the relay control unit in a single board.

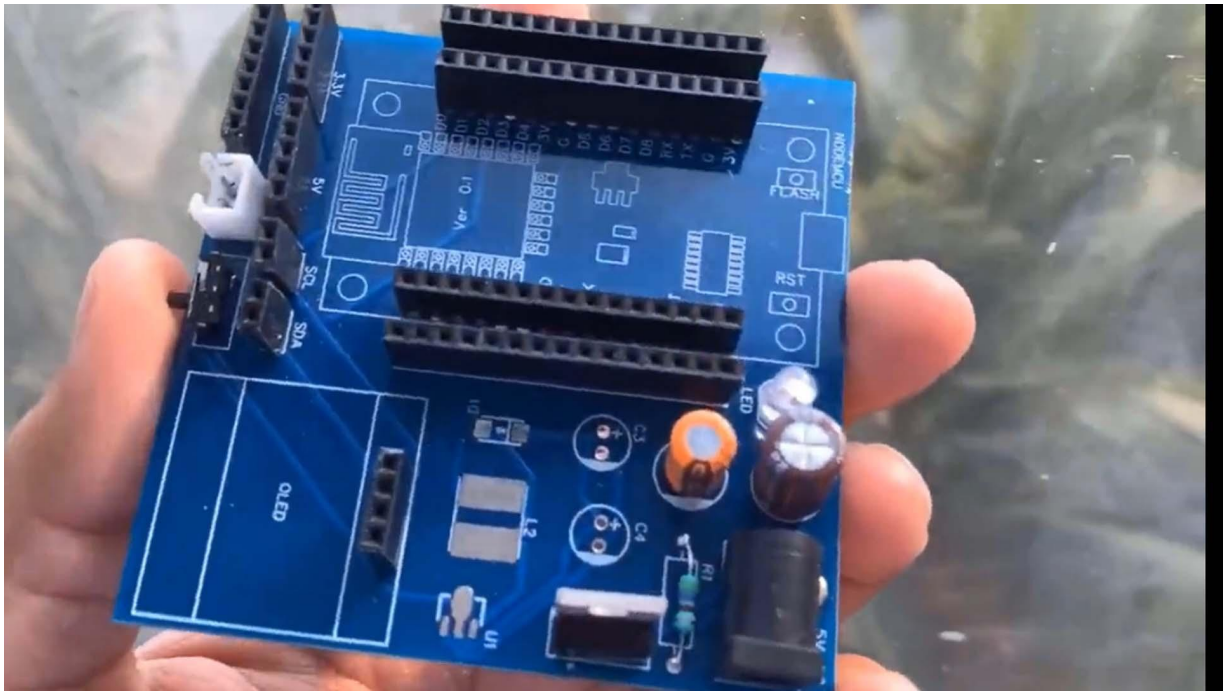
I then converted the schematic to a custom PCV. The PCV size is very small and all the components are placed perfectly on the board. Here is a three d view of the PCV board, which gives an overview how the PCV will look like. Now it's time to order the PCV. So I visited pcb.com and went for an instant quote.

Then I uploaded the cover file. Then I selected the pcb material dimensions quantity, thickness, mask, and rest of the order details. Then I selected the country of shipment and finally pressed the order. So after a week, I received 5 samples of PCBs. The PCB looks something like this.

The PCB quality is as excellent as I expected. The top and bottom layer solder masks are perfect. Because of the great quality, it's easy to solder the electronics components on the PCB. Alright. Now, I have to solder all the components on the board.

So, I soldered all the male female header spins along with capacitors, registers, LED, and switches. After soldiering, I placed or inserted the note MCU board and OLED display on the PCV. The device can be powered via a 3.7 volt battery or using a 9 volt DC adapter. I also connected a pair of IR sensor modules and a relay module. Hence, our final hardware is ready.

You can place the 2 sensors at separate positions for testing. 1 at the indoor and other at the outer would be the perfect position to count the visitors. Alright. It's time to set up the software part. For this, visit Ubidots and create an account or simply sign in.



I would directly sign in here. So on the demo dashboard, nothing is displayed. And also on the device part, No device is created now. Click on the right side of the profile and click on API credentials. So here you can see the default token.

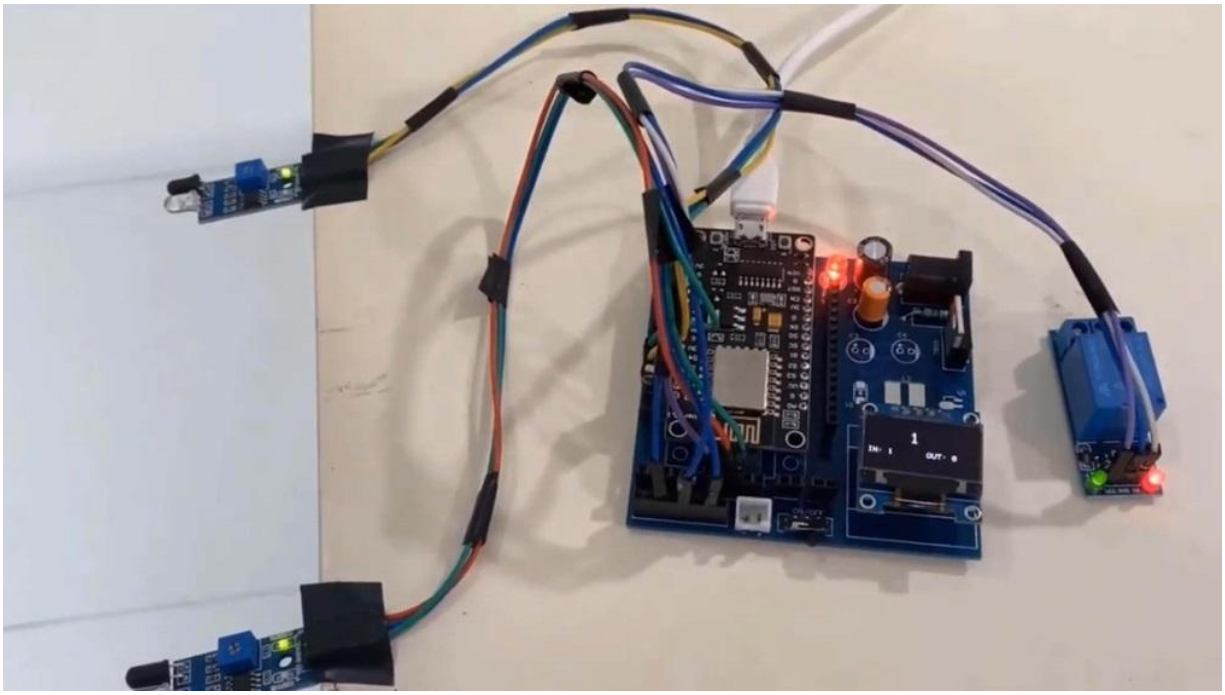
Click the zone, the token, and then copy it. This token will be used in the code to connect your node MCU board to the wifi network. In this code part, replace the token with the token you just copied, change the WiFi society and password. In this code, You will also need multiple libraries like dots and QTT library and OLED display library. The rest of the code is similar to previous codes.

In the bottom part, we are sending it out now and relaying status data using the client dot add function. From the tools, select the note MCO 1.0 board and also the COM port, then upload the code. Once the code is uploaded, open the serial monitor. The device will connect to the Wi Fi network and will print its IP address. Then it will establish the amputee connection and start sending the data to the Ubidot server.

The serial monitor will also display the visitor's data like incoming visitors or outgoing visitors and a total number of visitors as well as the life status. So let's test the hardware now. As soon as you power on the device, the OLED display will have no visitors as no event has occurred till now. To check the device, you have to bring some obstacles in front of the IR sensors. So, let's do the testing.

One of the sensors counts the visitors as to how many people have entered through the gate. Whenever an interrupt is deducted, the counter will add one value to the previous value. The relay only turns on whenever the current visitor count is more than 1. Similarly, the other sensor shows the total number of outgoing visitors through the different gates. It also adds one value to the previous value whenever an interrupt is detected.

The total number of current visitors is calculated by subtracting the outgoing visitors from the incoming visitor. Alright. Let's check the Ubidot's dashboard now. On the device part, when I hit the refresh button, you can see a new device is added now. Click on the device.



So on this dashboard, you can see 4 different variables appearing and the last data entered were just a second ago. So go back to the demo dashboard. From here, you can add widgets. To do that, click on the plus sign and select the widgets. then select the device and select variables.

Do the necessary settings like color or size. So, our first widget is ready for the 1st variable. Similarly, create the widgets for all other

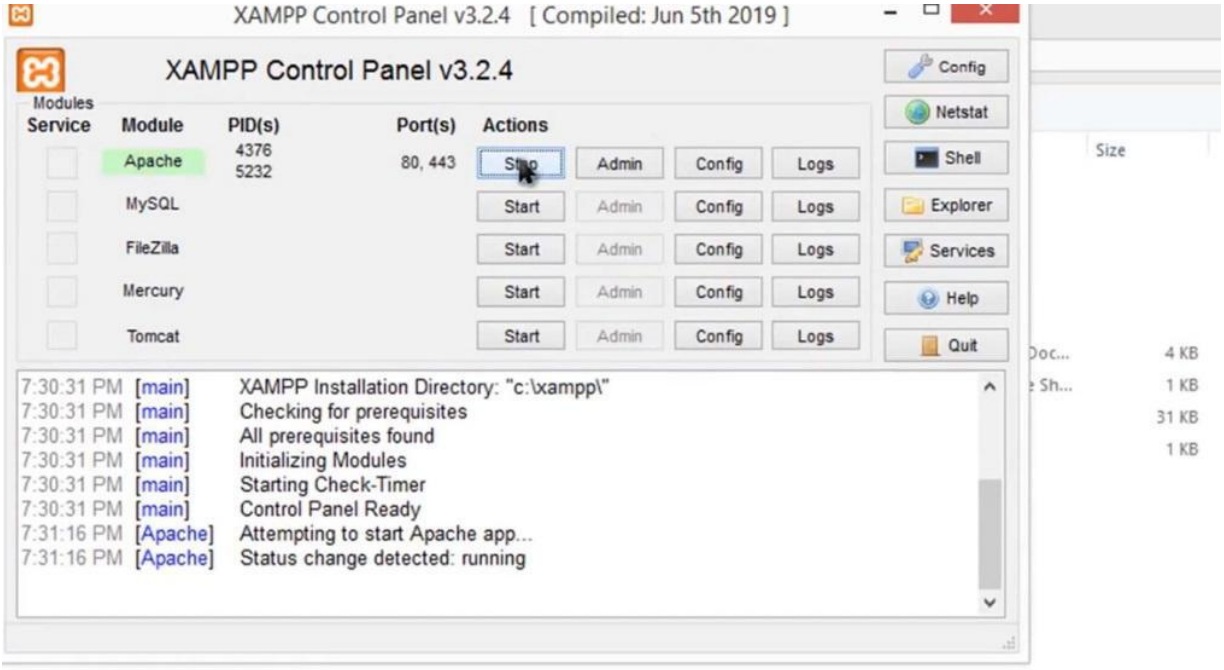
remaining variables. Finally, we are done now. So whenever a visitor change takes place, the dashboard gets updated, And here you can see the visitor's status is getting changed.

The light turning on and off status is also represented by green and red color. And the visitor frequency can be seen in graphical format. Alright. This is such a great IOT based by directional visitor counter project.

IOT BASED BIOMETRIC FINGERPRINT ATTENDANCE SYSTEM WITH NODEMCU ESP8266

And in today's project, we'll learn about fingerprint sensors and biometric extension systems using node mcu, OLED display and R 305 fingerprint sensor, and you will send this data to the IOT website. That is the data from the node MC will be directly uploaded to the Internet from where you can assess the data and see the fingerprint attendants of record service today. So let's get started. So first, you have to create a website.

So if you don't have a website, you can do this by an app called jam. So simply search it on Google, and, you can download this from the link, select your operating system, Windows 7, 8, or whatever you are using. Download recording to your requirement. For me, I am downloading 64 bit. Now, once the download is completed, just install it.

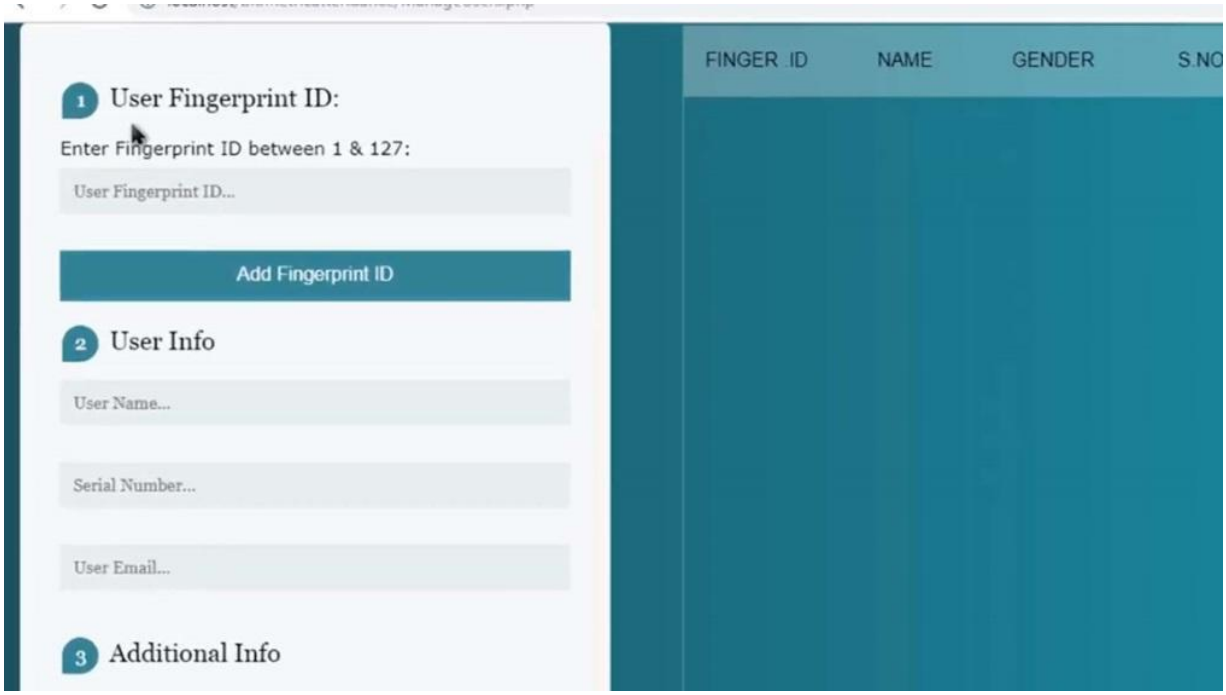


So once the installation is completed, select the language, and a jam control panel will appear. For now, just minimize it, and this is the folder which contains the information of all the databases and websites. So what you do is copy this folder. I have given the link in the description. Go to the c drive, go to the jam folder, and HD docs, and just simply paste it over here.

So now you can go to the jam control panel and start a path and MySQL. So both of these have been initialized and just started. Now, you can simply minimize it or close it. Now, go to the connect database file, and edit it using Notepad plus plus. Now, edit the server name, username, password, and database if you are using our website, but if you are not using the website, you are going with a jam folder, no need to edit.

Similarly, also edit this file. So once the editing is completed, just save. Now go to the web browser and type local host biometric attendance task install. So the database and tables are successfully created here. Now go to the local host by Metricate and then /index.pse.

So a website will open where you will see this type of interface. You just log, manage users with ID name, serial number, gender, finger, ID, date, and timing. So on the user log, you can see the log of the students or employees. So here you can select the date for which you want to know the record. And, this option is to select the date.

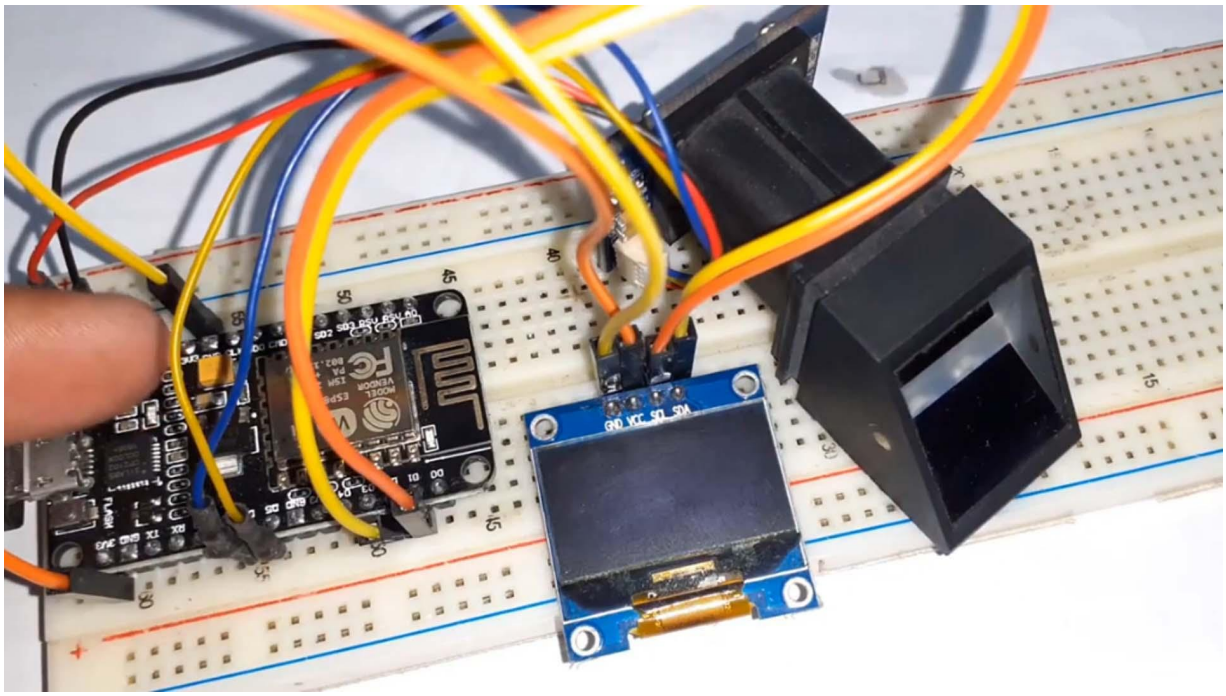


Now, from here, you can even import the data to Excel. So this managerial area is used to eat the user or delete the users. So you can add a total of 127 users here. So user info is just filled here and registered via fingerprint. So you can see it, user update, user, and demo feature.

All the functions are here. So this is the circuit diagram. So we need an OLED display, 305 fingerprint sensor, and no MCU. OLED displays are connected to the I2C pins , STH is connected to D1, and SCL is connected to D2. It is supplied with 3.3 volts.

Similarly, the fingerprint sensor is connected to D5 and D6 pins via UAT. and it is separated five volts from the hinge pins. So you can see this is how we have assembled the circuit on a breadboard. This is a 0.96 inch OLED display. This is the R 305 fingerprint sensor.

You can use any sensor like R307 or anything compatible with it. And then, interfacing is the same, exactly shown in the circuit diagram. Now, connect the USB port to the That is my Cresp port to the node MCU, and then we need to upload the code to it. So now let's see the programming section and modification that you need. So this is the program.

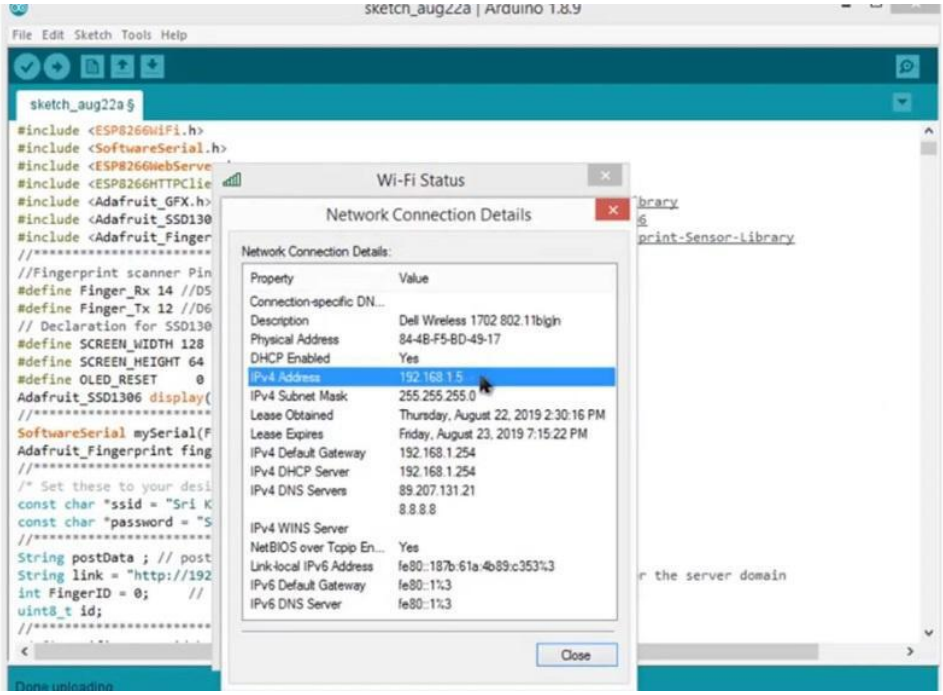


So it has so many header files, and no library required for other header files except these 3 header files. That is GFX, SSD 1306, and a diaphragm fingerprint. So we need a library for these trees. So the library link is given in the description, or you can simply go to the live library manager and install the library from here itself. So the fingerprint library is already installed here.

You can see. And then we have the GFX library. So even the GFX library is installed here. And then you have ssd 1306. Digitalsoinstrant.

So Also change the SSID and password. So, you can edit here in the setting. simply enter your WiFi name and then password. So here, you need to make a modification. You need to add your computer IP.

or your server domain in case if you are using a website right now, I am using a computer, so I'll use my computer IP here. So for this, go to the network setting and then click on the Wi Fi, to which you are connected. I'm on Windows 8.1. So this is the interface. carriage the IP address.



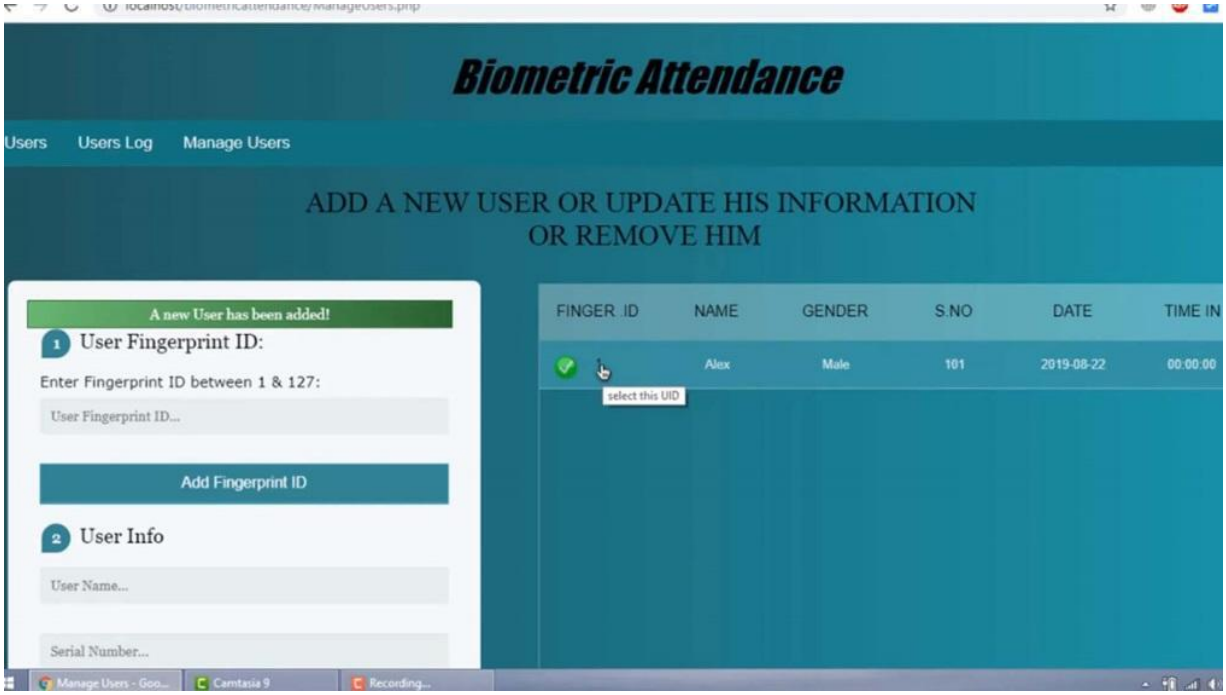
So I just need to copy this IP address and enter it here. It's something like 192 0.168.1.5, then select the board. That is when using a node MCU board. So we have selected node MC 1.0 to be a keyboard. and then go to the tools and select the port.

And then simply upload the code to no DMCU code. Once the code is uploaded, it will start connecting to the wifi, and it will also show the wifi name. So once it got connected, you'll also connect it. Now, you need to air some fingerprints. Yeah.

For that, before that, you can see whatever the IP is connected to and Wi Fi. you can also see on the serial monitor. Even if your finger print is generated, the serial monitor will show here. So Now, we'll add some fingerprints to the fingerprint sensor. So just type the fingerprint ID here, and then click on add Now, select it, and then give the user info, like, username, serial number, and then, user email address.

And then simply add the timing. But now I'm using 9 AM. and then select the male or female, and then click on add user. So once you click on add user, you can see number 1 is printed over here. with name, gender, serial number, data time, everything automatically filled.

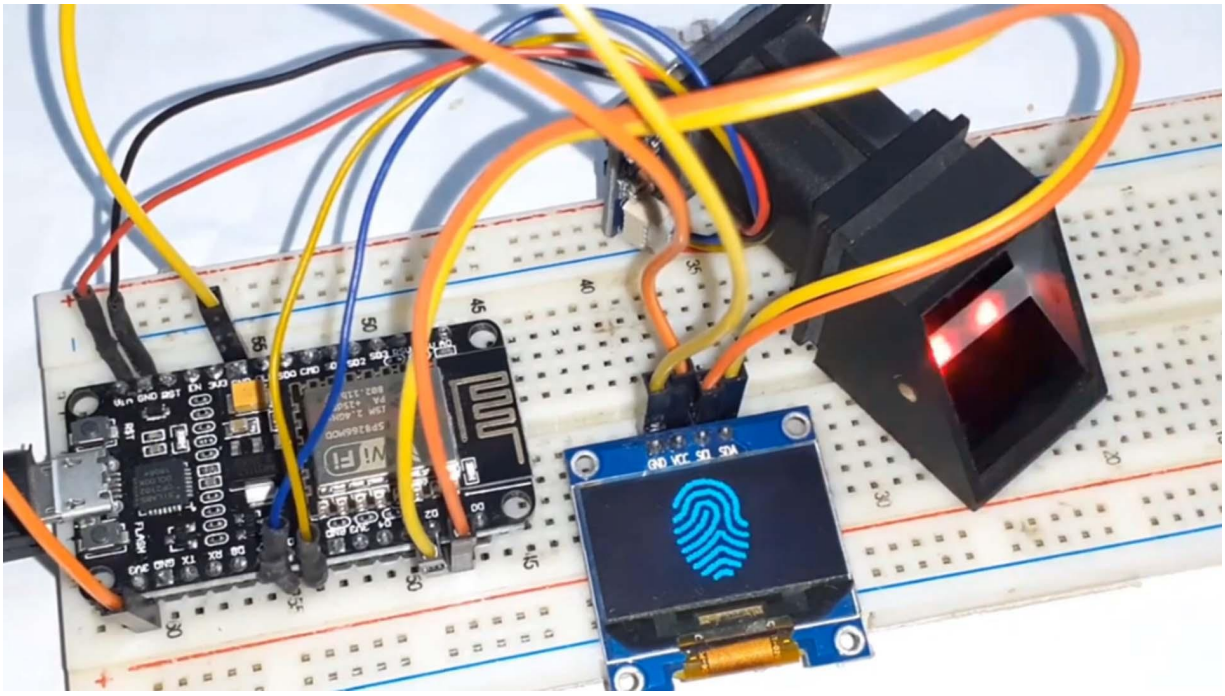
Now, once you feel here, your OLED display will start scanning. So when it displays the scanning, put your finger, and then remove, and then again, So the fingerprint has been added to the database. Now, we'll do this by adding another finger. So for that, Click on number. So I'm giving number 2, and then selecting the number 2, and also entering user info again.



For example, I am using the other finger by the name Lucinda, and other in force here. So it sees the female, Now, again, scan the finger of the second user, user 2, user 3, user 4, however, so the fingerprint is going to be stored here. So the fingerprint has been hidden. Now, do the same thing for multiple ugas, like I have done for 5 ugas here. You can do whatever you want.

Now, once the data is stored, you can simply click, and it will show welcome. And then, on the second time, if you give the finger, it will say goodbye. On the first time, it will display you the welcome

message. On the second time, it will display the goodbye message assigned from morning to evening working hours. So this is how the fingerprint sensor works. So if the finger doesn't match, it will display a trust sign.



So you can see that trust sign is displayed over here. So now when you go to the users, you can see 5 pages are here. And, all the users time in and out will be displayed when you go to the users log. So here, 5 pages. So you can see the date will be loaded.

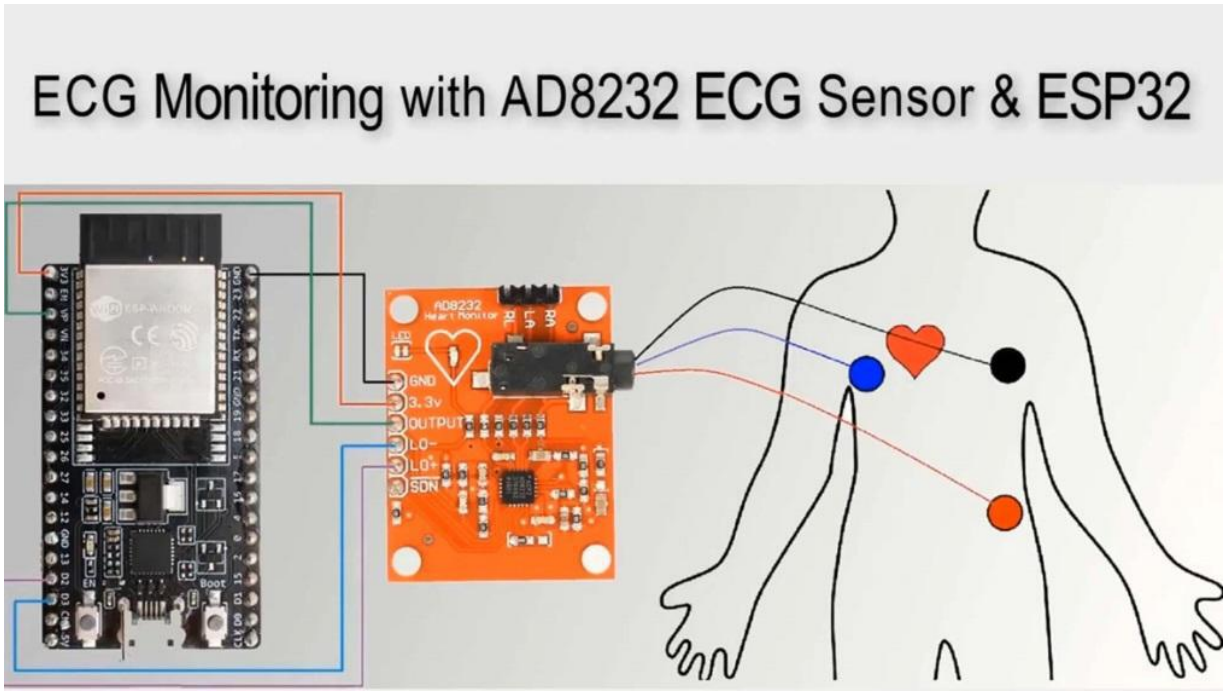
So it will show that today. To see, you can see the fingerprint ID date time in and time out. Both the sync data are already showing here. So you can export these 2 Excel files where the Excel file is downloaded here.

IOT BASED ECG MONITORING WITH AD8232 ECG SENSOR ESP32

So in one of my previous projects, we used an ADA to 32 ECT sensor with Ardeno to measure the electrical activity in our body, and we generated a graph that is the repetition of voltage. So this is an ADA232 ECG sensor that we interface with Ardeno in one of my previous projects. So I will describe all the pins and representations. And then we just placed electrodes on the body of one of my friends, and then we connected it to the serial promoter.

And Once the serial plotter was connected, we observed the ECG signal away from on the screen itself. So this is a low cost ECG module. So now in today's project, we'll use ESP32 and generate an signal using 88232 sensor, and then we'll connect the electrode 2 and or simply fund the body like in the previous project, and when the web form will be generated, we'll use an online platform called UB Duts where we'll upload the sensor data, like the data from the ECG, as you can see here, the ECG signal is directly uploaded to internet. So you can monitor the ECG data from any part of the world using IoT devices. So let's begin.

So, this is a low cost AD8232 ECG sensor that is connected to ESP 32. Ground is connected to ground 3b3 of ESP 30 b is connected to 3b3 of 88232. DP is the output pin LO plus is D2 and LO Minus is D3. Now, this is how the electrode placement is done. We have 3 different leads.

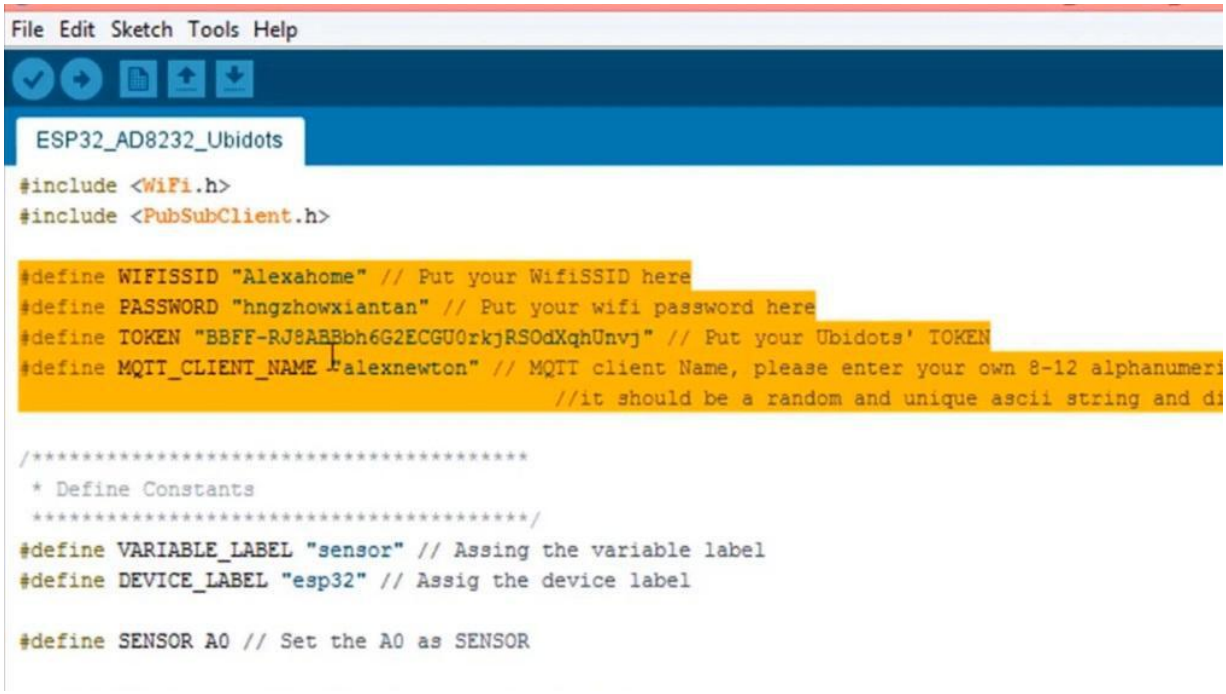


The placement electrode was already explained in the previous project. You can see if you want. So from here, you can see, I have done exactly the same connection from ESP32. All the high pins That is ground 3b3 output, l o minus, and l o plus. I'll have this connected to my hand.

So we have a dead electrode, low electrode, and Willow electrode. Also be placed in a triangular shape, like we placed in the heart. This won't give the appropriate signal, but I'm making this project just for a demonstration purpose. If you want to measure the good signal, then you can place it in your body. So let's go to the coding now.

So we have 2 header files that are wi fi dot h and pop up slides. And then to upload it to the internet, we need 4 parameters. That is Wi Fi SSID, password, token from UB Duts and then MQTT client. Okay. So WiFi SSID and password is your phone local ID or the Wi Fi that you're using on your phone.

It's better if you use your phone, Wi-Fi. So now let's go to the UV dots. So just go to this website. So, this is the first link. And then, just create an account if you haven't created an account or simply sign in if you have an account.



So I'll make an account very quickly. You can make whatever data it asks. So give the app name. And then insert a password and just simply click to create. So this is how you can create the account.

So you can go to the account. Okay? So you can see how the data are represented here in numerical value, in graphical value in the point of button form or everything. Simply, you will just see such a great platform for everything. Okay.

Now, I will go to my dashboard as my account is already also. You can see 20 days of trial is left for me. Okay? And you can get a 30 days trial if you are a new user. Now click on device And then when you click on device, you need to add one of your devices.

So, currently, you can see no devices are there. So you need to create a device. So from this list, click on black device and name this as ESP32. Don't name any other thing. If you name, you have to make changes in code.

So I'll explain it later. So our dev device is created and there is no last activity as nothing is sent or received. Now just when you click here, you'll get an option to 8 variables. So you will get raw and synthetic. Just click on raw and then rename this new variable.

rename it as sensor. If we rename it to any other thing, then you need to make changes in your code. So now you can see, there is a description tab where you can add the description. So the device is created and its parameter is also created. Now, we need to go to the dashboard and create a graphical representation.

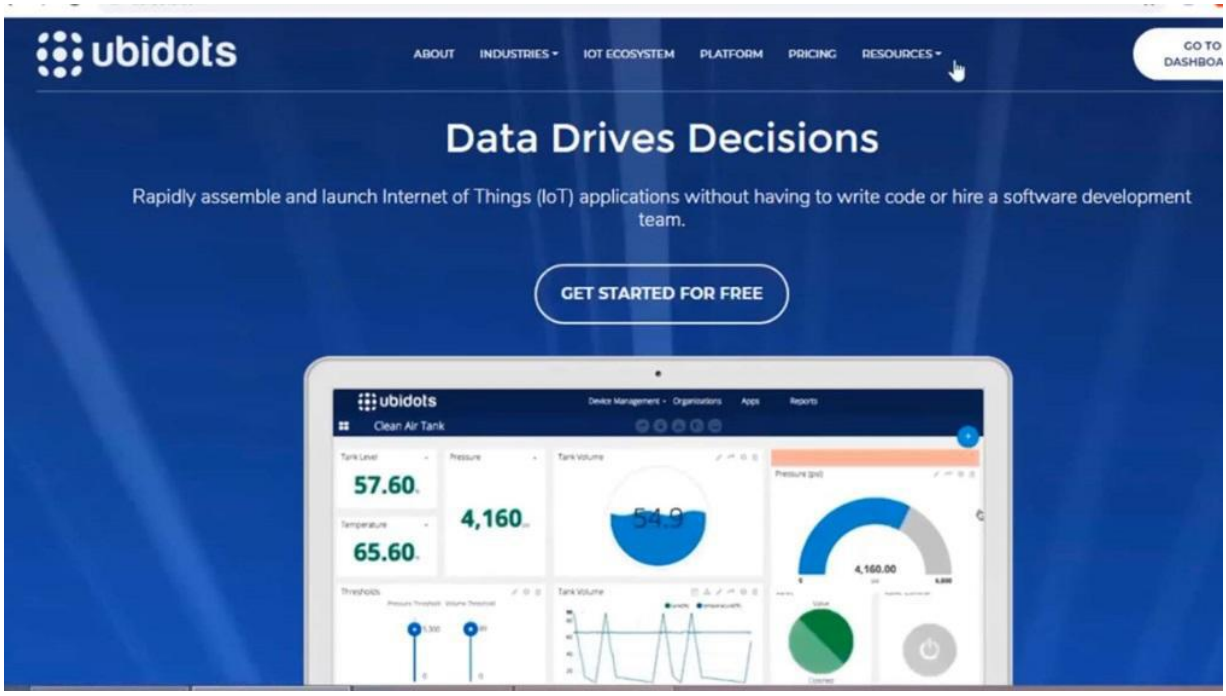
So, click 1, add a new widget. So, you can click plus sign to create So here you can see there are so many widgets. So from here, I will select the line chat only. So click 1 plus air variable. So 8 ESP32 that was created earlier.

And then click on sensors. So, done. So leave everything blank except the appearance name. rename it as ECG monitoring or ECG sensor or whatever you want to do. So I'll rename it as ECG monitoring.

then click on the green tick. So, you are done with the data. Currently, no data is found as we haven't updated anything like that. Okay, sir. From here, you have so many options to edit, download, and share the audible modifications you want to make.

Now let's go to the program again. So here you can see ESP 32, the device name, which was created earlier, KETI, I ask you to rename the same thing. If and here you can see it's written as a sensor. So you can see on the left side, it's a sensor. So if you want to rename any other thing, you need to make changes in the code too.

or change it in the ub. That's the account API label. Now, go to the dashboard itself. So it's still a, you know, activity. So now make changes in Wi Fi SSID and password.



We need a token. For that, go to your ID and click on API credentials. So here, click here to show. So this is my token. Now copy it.

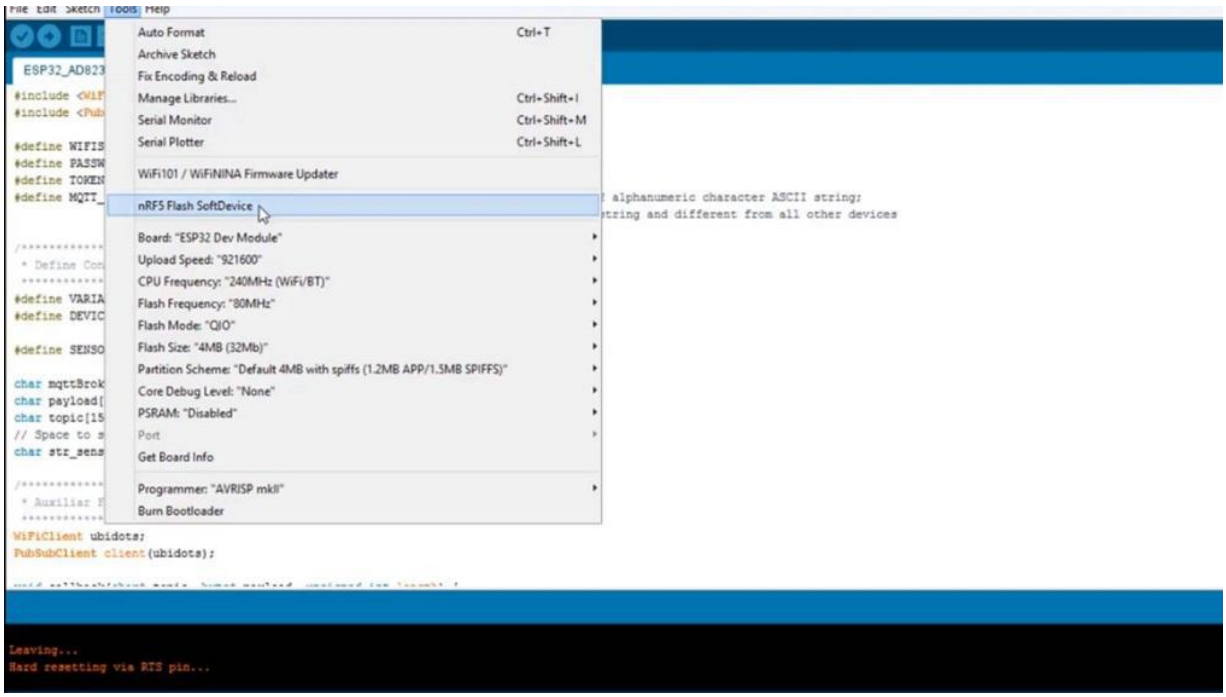
Now go back to the program. Make a copy paste here. So my token is the same. And, also change your SSID and password. mqtt cloud name can be anything you can name anything whatever you like.

It should be between 8 to 12 characters. Now, connect the ESP32 with 88232 and connect it on your end or in EHS. Now select that device. Okay. So the ESP 32 development module Uploaded it should be lesser than 11520, and leave everything as default.

Okay. So once you connect the device to the computer, it will detect the port. So my port is comb type. So simply upload the data to the ESP32. Press on the boot button while uploading.

If you want, press sometimes the data might not get uploaded. So once the serial monitor is clicked, it's waiting for the Wi-Fi. So once the WiFi is connected, it can get an IP address and it starts

publishing the data to UBS Cloud. So now go to the UBS account. Refresh it.



So now you can see there is a blue map. This means the device is online. And the last activity was a few seconds ago. Click on the same device. And then here you can see There is changes in data.

Since I haven't placed the electrode on my chest, I am not getting accurate data. Also, remove your charger from your laptop so that there is no disturbance in the noise. Use only DC value from the USB port of your computer. So go to the dashboard again. And then on the dashboard, you can see a representation of an online graph.

So I'll maximize it. So even after the interval of every 1 or 2 seconds, data will be up to a date. So you can change the frequency timing.

IOT BASED ECG MONITORING WITH AD8232 ECG SENSOR ESP8266 ON UBIDOTS

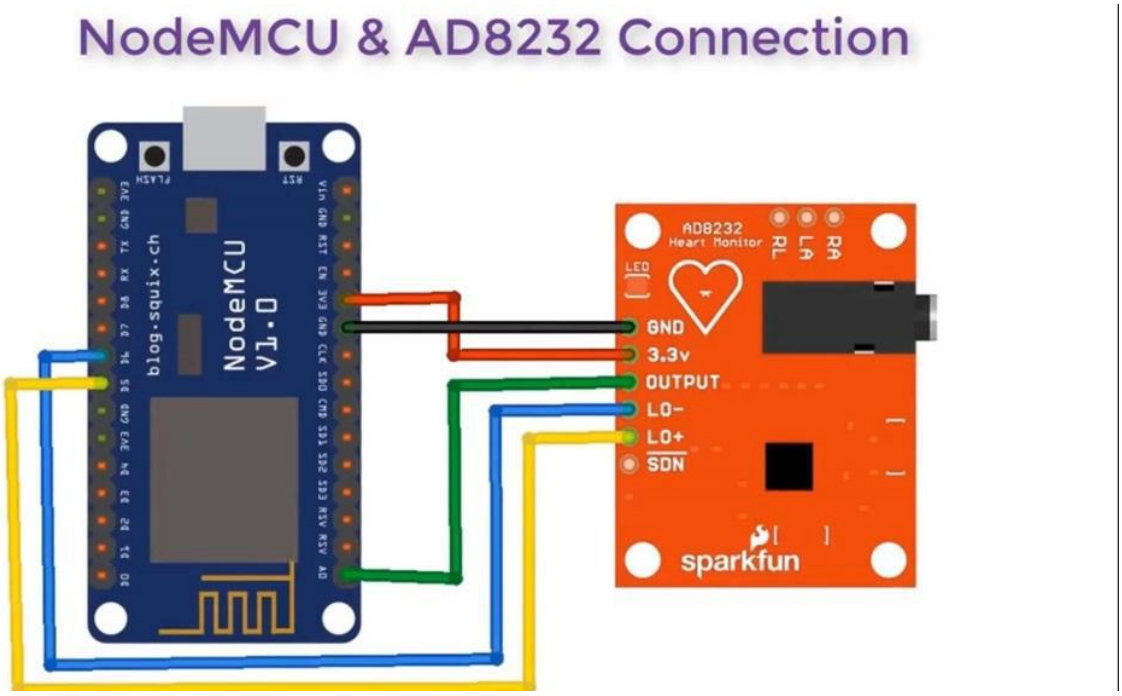
We'll learn how to make IOT based ECG monitoring time using 88232 ECG sensor and nodem's USB 8266 2 LE board. We'll simply interface the 88232 ECG sensor with the nodemcu board and connect it to the internet. We'll connect this lead to the patient body and we'll monitor the ECG. Similarly, we'll use an IoT platform called UB Duts where we'll use the amputee protocol to send the data over the cloud, and then we'll simply monitor the ECG of a person online using the 88232 ECG sensor.

So in the UB DUS dashboard, we can simply monitor the ECG of a person online from any part of the world. So let's get it started, but before that, I'd like to tell you the official sponsor of this project is the PCB. Next PCB is one of the biggest PCB manufacturer companies in China. They offer high quality PCB air, at a very general price. You can get 2 layer, 4 layer, and multi layer PCB.

Can currently due to the COVID 19, they're giving the biggest giveaway offer. So you can get an infrared thermometer worth around $50 if you order a PCB for $500. So at any order, you'll get a 10% instant discount. So If you are ordering a PCB assembly unit, you'll get a discount of $30 immediately. Now let's begin with the circuit diagram first.

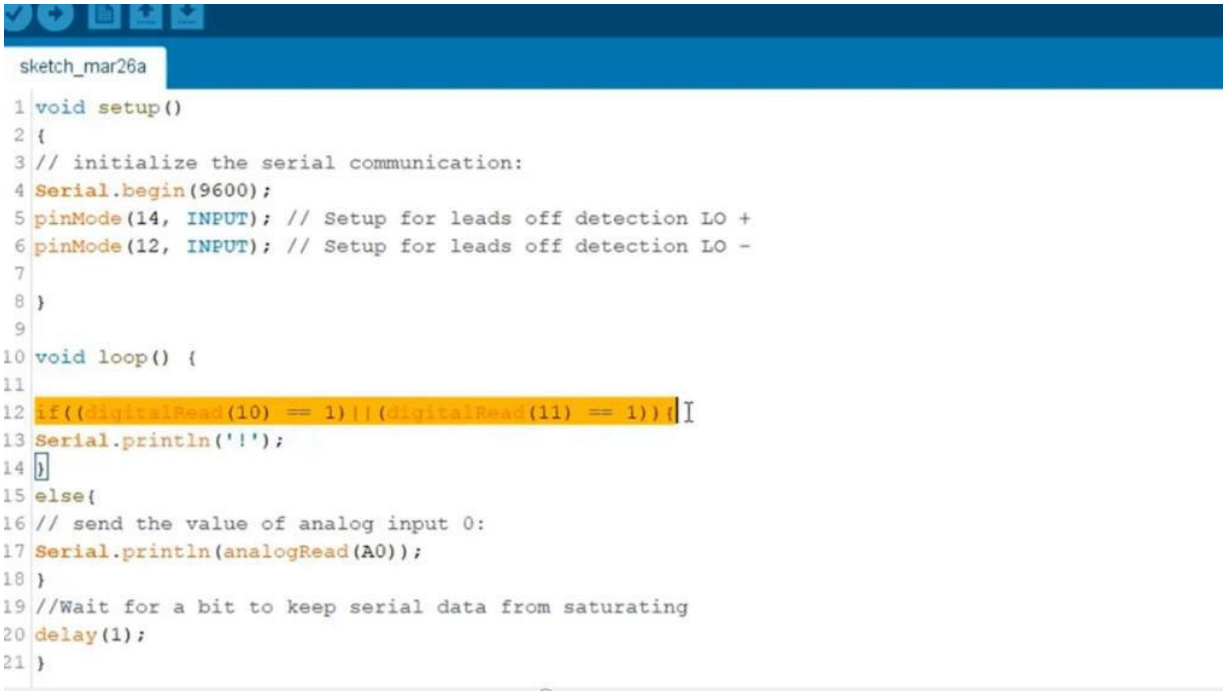
So 88230 has 6 SPIN. Out of them, STN is not used to connect the output pin to the analog pin in our top node MCU, connect the yellow plus and yellow minus to D5 and D6. Similarly, connect 3 B 3 to 3 B3 and ground to ground. So Here is a connection, how the ECG leads are placed on a patient's body. So we have r a, l, a, and r l.

That is the right arm, left arm, and right leg. So you can place it on your chest as well. So here is how you have assembled the ADA232 and nodemcu on a breadboard. The connection is exactly the same as shown in the figure earlier in the circuit diagram. We have 3 electrodes, that is ECG leads.



All these EasyG leads need to be placed in a triangular shape, either in the chest or in the arm or in the leg. So here is one of my friends where I'm just demonstrating how we can put the ECG, you know, towards. So, the red one is when the lip chest wallows in the right, and the green one is at the tone button near the kidney section. So, it's a complete triangular save. Now, let's begin with the simple program first.

So under the bird set of functions, we have initialized the serial beginning at 96100. So we are detecting l o plus and l o minus as gpio14 and gpio 2L. So we are just comparing the digital output from the 2 electrostatic l o plus and l o minus. Under the else condition, we are just reading the analog value that is generated from. Upload the code.



```
sketch_mar26a
1  void setup()
2  {
3  // initialize the serial communication:
4  Serial.begin(9600);
5  pinMode(14, INPUT); // Setup for leads off detection LO +
6  pinMode(12, INPUT); // Setup for leads off detection LO -
7
8  }
9
10 void loop() {
11
12 if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
13 Serial.println('!');
14 }
15 else{
16 // send the value of analog input 0:
17 Serial.println(analogRead(A0));
18 }
19 //Wait for a bit to keep serial data from saturating
20 delay(1);
21 }
```

Go to the serial plotter. So, you can see a beautiful ECC signal is generated. That is the PQRX diagram. To know more about the diagram, you can visit my previous project. I have given the link of my previous project in the description where you can understand what about this diagram.

So this is how we can measure the simple ECG. Now what I want is to upload this ECG signal to any IoT cloud platform where I can monitor the ECG signal online from any part of the world. For that, I choose UB Duts. UBS is one of the biggest and great platforms for IoT projects. So, just visit UBS dot com, simply create an account or simply log in if you have an account earlier.

I created the account today, so I am getting a trial of 30 days. So you can purchase a license from them as well. Now let's see the code. So we have e s p 8266 wifi dot s Wi Fi communication, and we have pubsubclient.h. You can get this library from the link in the description.

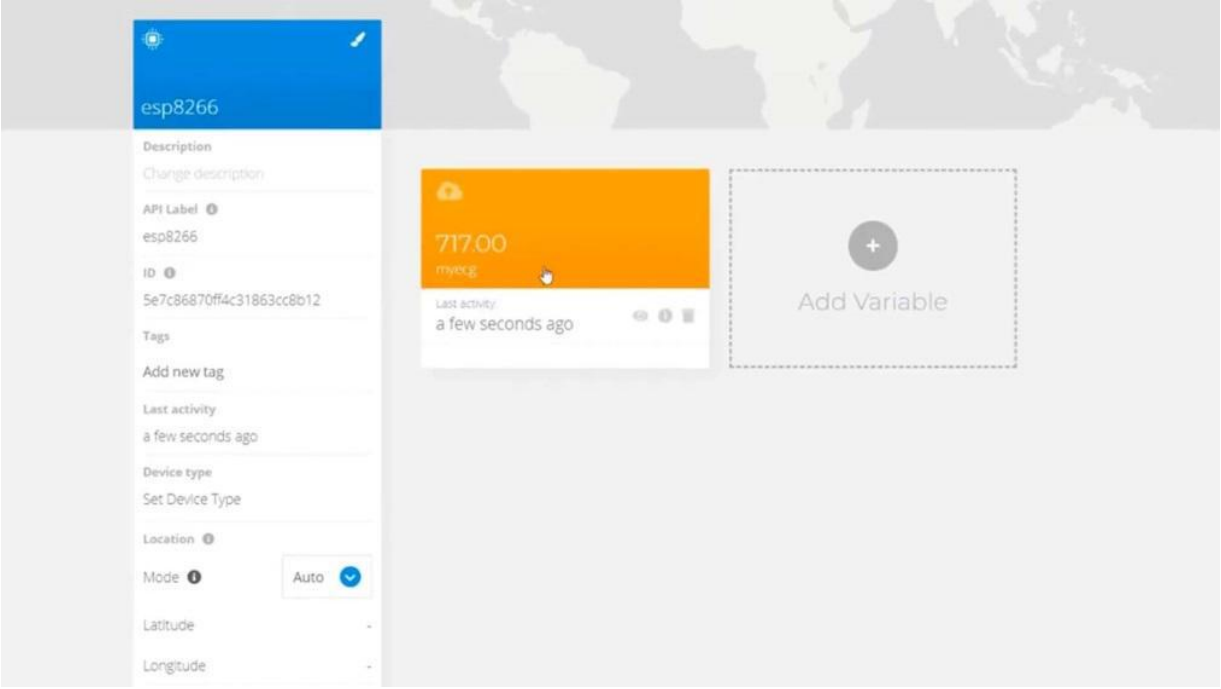
I have added the link and you need to change the WiFi SSID. And, you also need to change the password as well. Under the token, you

need to get it from the UB dots, so how do we get this? Just visit the API cadence here. Just click here you can see the tokens, copy the token, get back to the code, and paste it over here.

Under the empty, the client name, you can write any name, whatever you like. There is a variable label and device label. These 2 are the only important parameters that will be displayed using the MQT connection. Rest of the code is similar as earlier, you can just go through the code and understand that the entity protocol is used for sending the data over the Internet. We have also used a pop- up client for that.

Okay. So they are solved from the code section. If you are facing a problem in understanding code, you can comment in the project section below. So once the code is uploaded, you can open the serial monitor and check the boundary to 115200. So it will start connecting to WiFi, and it will print the IP address, and it will immediately establish a MQTT connection and publish the data to the UBS Cloud.

So go to the UBS Click on devices. Now after clicking here, nothing is displayed. So what you do is just insert your USB cable to the computer port. So once you insert it, just refresh this. So you'll get the device name at ESP 8266.



Now, remember, this device has come automatically from the code that you upload and this variable called my ACG is also generated from the code. So just click here so you can see the entry of the last data and the current data with the wave from above below. You can just change the color of the graph, whatever you want. Now go to the data, click on the dashboard. I'll just want to delete this chart because I don't need this one.

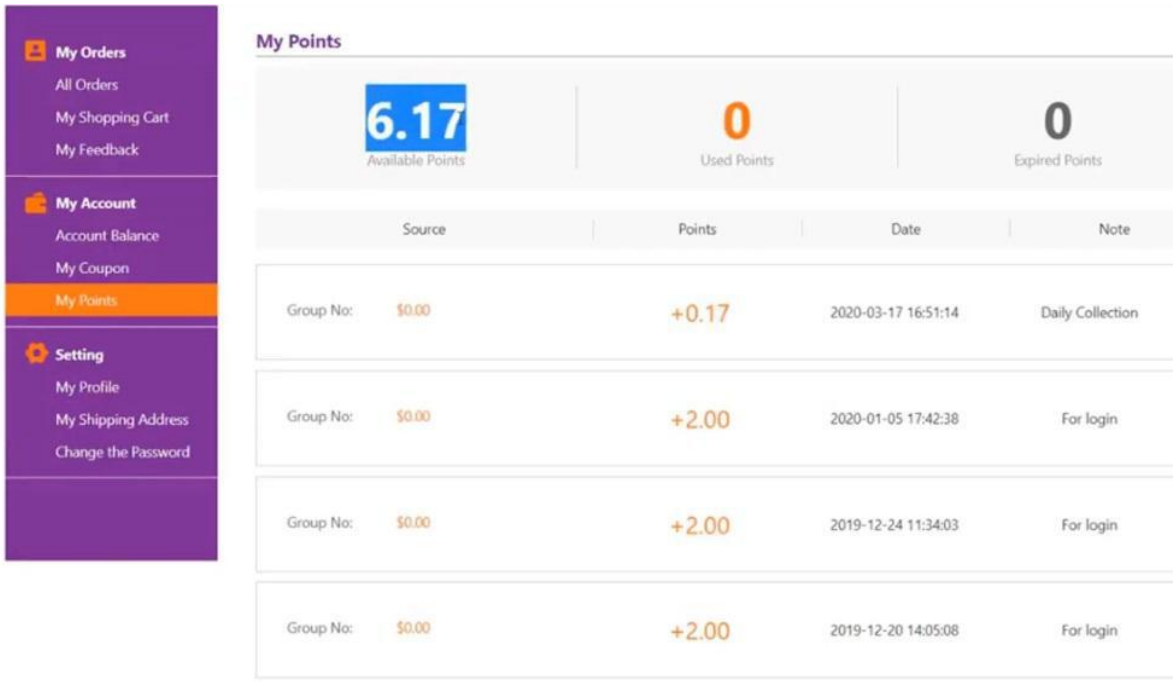
So click the plus sign here. click on the line chart and just name it something like that. MyECG sensor. like my ECG graph and the variable. So under the 8th variable, you'll get the device name as ESP266 and my ECG.

So this has come automatically just by clicking on the green tick. So once you click on here, you'll get the ECG graph that was seen on the dashboard. So, with time, the ECC signal from your body is uploaded here. So uploading will be done immediately. The graph won't be smooth as you saw earlier in the previous code.

IOT BASED PATIENT HEALTH MONITORING SYSTEM USING ESP32 WEB SERVER

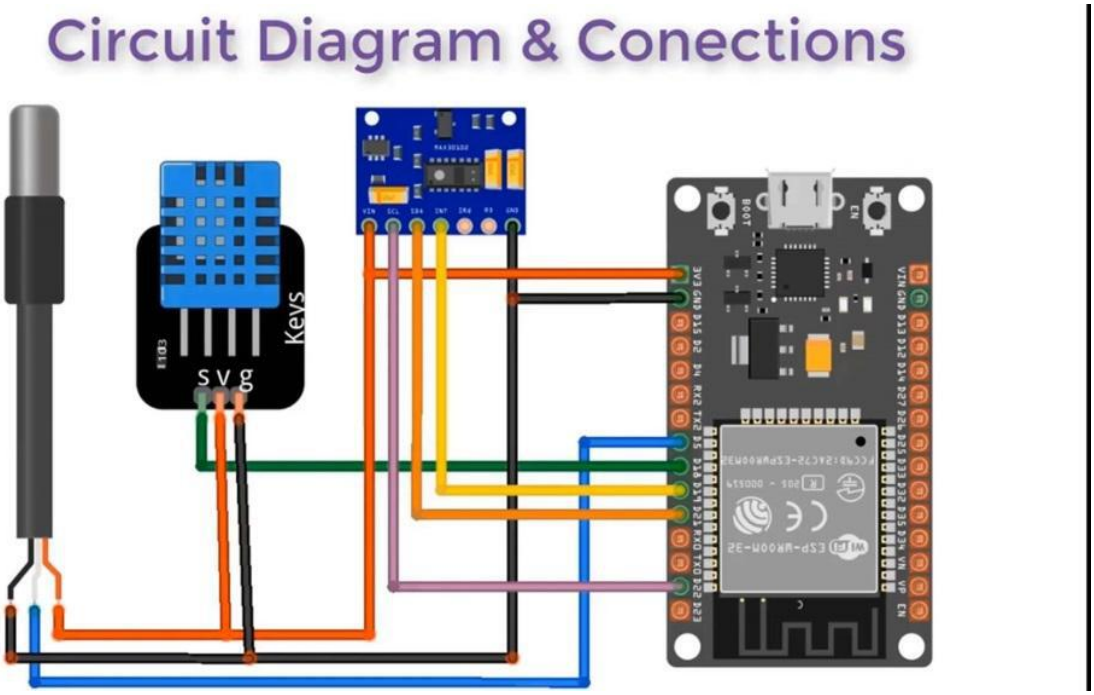
In this episode, you will learn how to make and design an IoT based patient health monitoring system using ESP32 web server. So simply what we'll do is we'll interface some of the sensors like pulse meter humidity sensor and temperature sensor with ESP32. We'll use our web browser like a mobile phone or the web browser of the computer where we'll display the parameters like blood, oxygen, heart rate, body temperature, room temperature, and room humidity. We'll learn the interfacing process.

We'll learn about the coding, and we'll learn how the parameters can be monitored remotely online through the web server. So let's just start with the project. The official sponsor of this project is the PCB. Next PCB is one of the highly experienced manufacturer companies in China. They offer high quality PCB at a very low price and they manufacture all different types of PCB, including collared PCB and 2 layers. Currently, they are having an offer.



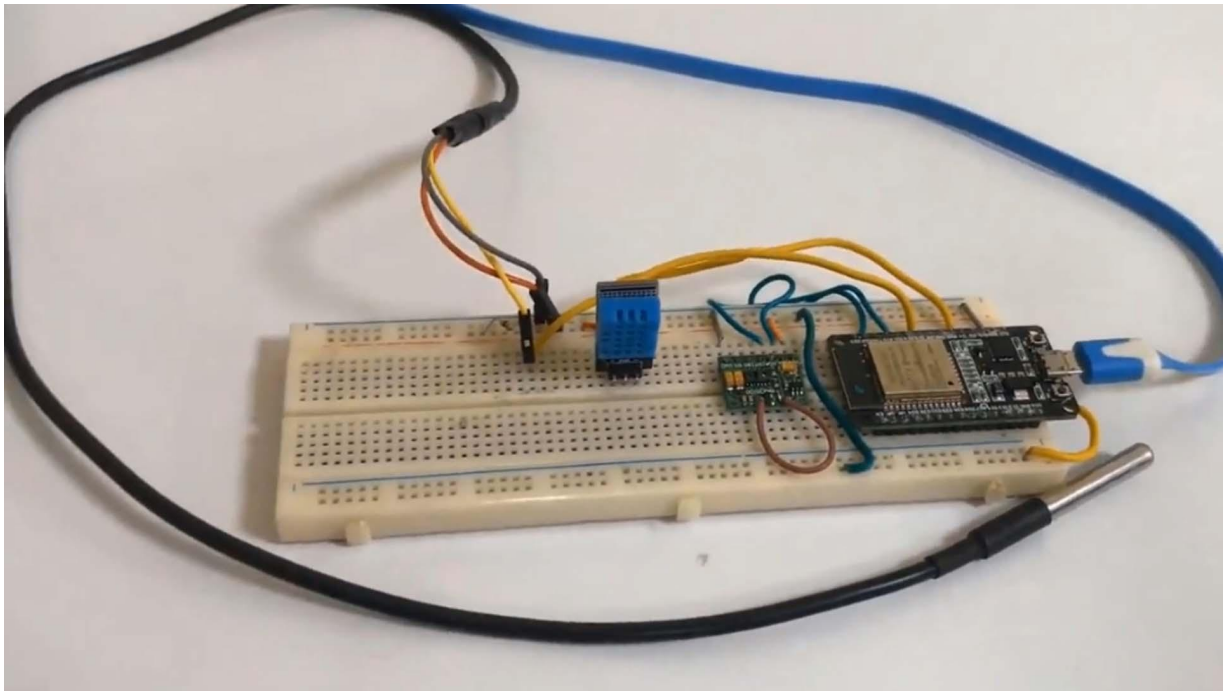
So, in Delhi, you can visit their website and collect the points. So by collecting the points, you can get a huge discount on orders. Like, I have collected 6.17 points. So 2.6 equivalent to $1. So the more the points, the more the TC will visit their website.

Fill all these details, select the country, and then upload the firewall file and place the order. So once the order is done, you'll get a PCB within 24 hours, and the PCB quality is very excellent as you can see. Now let us begin with the circuit diagram first. All these three sensors need to be interfaced with ESP32. So connect output enough DS-seventy 22.



GPI of 5 ESP 32, DST 11 output pin 2, gpio 18, and I2c pin up pulse oximeter 2, I2c pin up ESP32. Connected it into GPI019. All this model works at 3.3 volt power supply. So supply 3.3 from ES p32, and all the modules ground need to be connected to the ground. Connect a 4.7 kilohertz resistor to the DSA TV 20 output pin via 3.3 volt BCC.

Now, if you want to learn about the individual sensor, I have already made a dedicated project for all of these So max 30100 projects are here. 2 projects are there. And similarly, if you want to learn more about the DST lab and humidity temperature sensor, you can see this project.



Now this is the interfacing on the breadboard as I saw you in the circuit diagram. So this is a DS18b20 waterproof temperature sensor to measure the body temperature. This ESP32 module and we have max 30100 for measuring blood oxygen and heart rate. DHT 11 for measuring room temperature and humidity. All this can be interfaced or bed board.

but for interfacing in Broadboard, it's a complicated task. So I have made a dedicated PCB for this using an online PCB editing tool that is easy EDA. So, here is the PCB. It's very small inside. You can simply see the front view and back view here.

A very small size, compact size, portable, and very easy to use. So download the driver file from the link in the description below. So let's see the program now. Also, we have Wi Fi dash h and web server dot h. So these are for generating the web server.

So we have wires for ITC communication. This is the four pulse oximeter, 1 wire and the last for DS18b20 and dht.h1 Dht11, humidity, and temperature sensor. The DHT11 output pin is connected to GPI 18. DS18 will be 22 GPI 5. So we have defined a parameter a millisecond. That is 1000 for a pulse oximeter.

We have defined 5 variables, temperature immunity, BPM, SPO2, and body temperature. Change the SSID and password from here. So from here, we have initialized all the sensors. So we have assigned a port of a different web server. From here, it will start connecting to the network.

And when connected, it will display the IP. So this will check if the pulse oximeter is connected or not. So, 7, 6,000,000 per current is for better penetration of pulse oximeters. under the loop section. We are just reading the temperature.

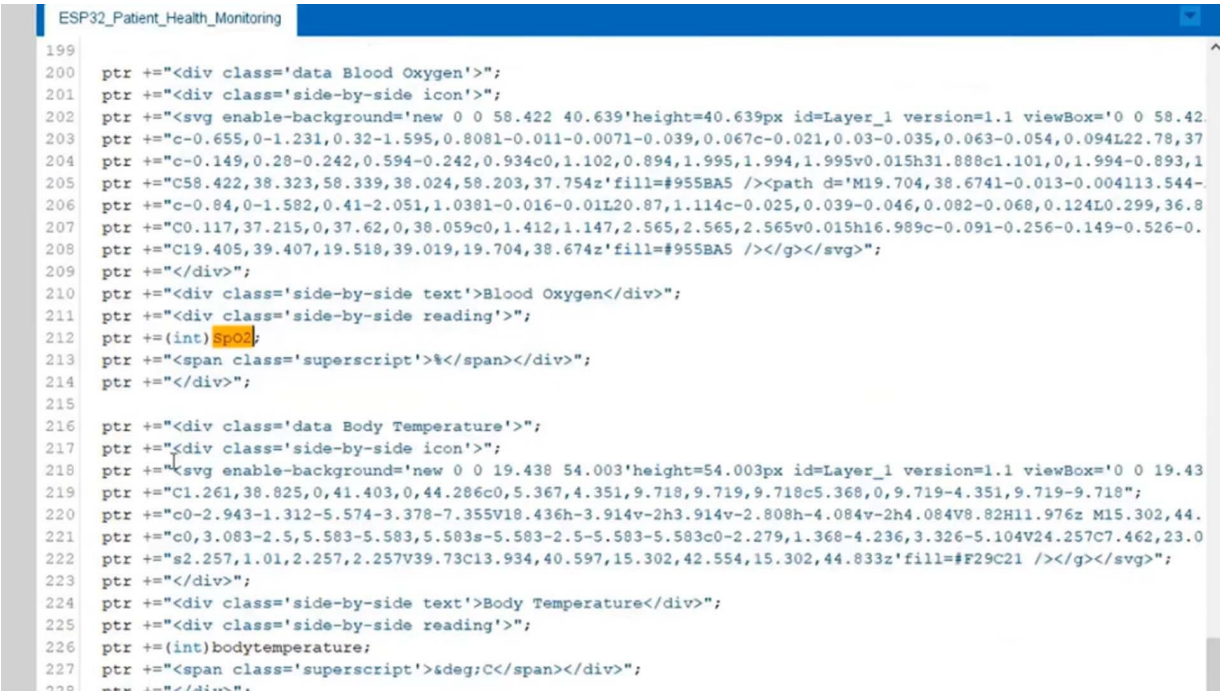
Okay. Like temperature, humidity, BPM S and body temperature. This all is displayed on the serial monitor. So once the display on the serial monitor is done, the WiS server will connect to the server. The SP 32 is connected to the local web server using the IP and it will

send the HTML data like temperature, humidity, DPM SPO2, and body temperature as shown here.

Okay. So, this is the HTML text. So, this line is for temperature. As you can see, the temperature parameter is defined. This one is for humidity. We have also defined humidity and percentage, similarly.

This one is for BPM. That is the heart rate. And this line is for the blood oxygen. That is spo2. And similarly, These lines are for body temperature that is from DS18b20.

Now what you do is just Go to the tools. Select the right board. Select the right board as well. k. Once the board selection is done, You can simply upload to the controller.

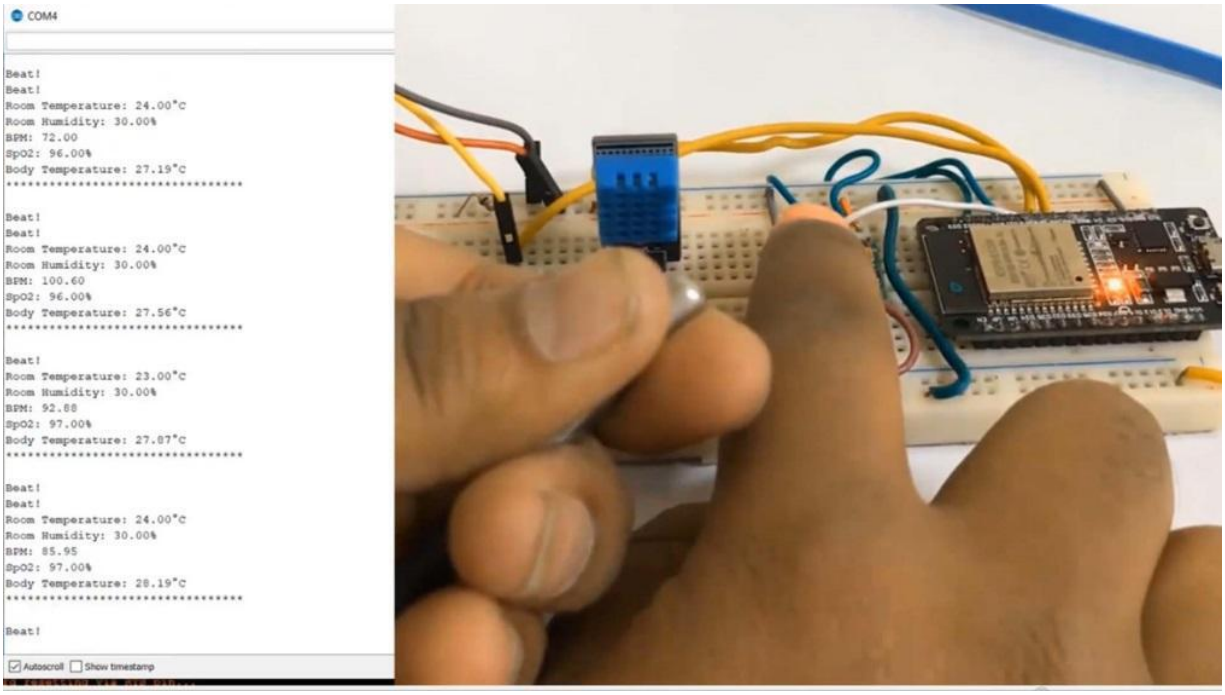


So after uploading, just press the reset button of ESP32. So you can see it is displaying all the parameters like room temperature, immunity, BPMSPO2, and body temperature. So place your hand or finger tip on the top of a pulse oximeter so you can see the BPM and s p 202 value is changing. So all the parameters can be observed on

a serial monitor. Now what I want is to check this value on the web server.

So how would I do that? So for doing that, you need the IP address of the node MCU for the local web server. Okay. So how will it get that? So simply, again, press the reset button up, ESP32.

After pressing the reset button, you will get an IP address. Okay. 192.168.43.45 or something like that. Okay. Copy this IP address.



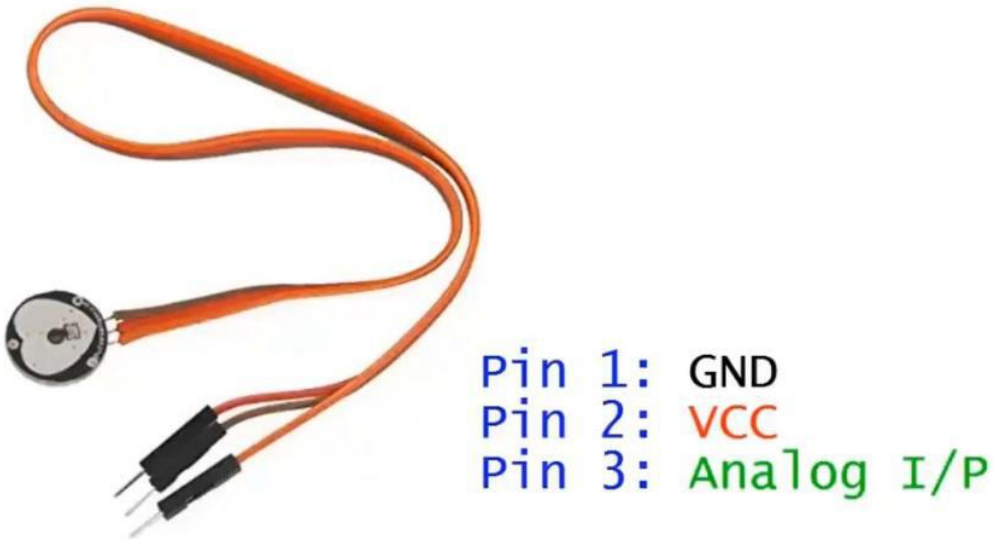
Open a web browser and simply paste it over here. So once testing is done, it will request the data from ESP32 and look at the beautiful widget displayed over here. So keep refreshing. Once the refreshing is done, the parameters will change. So you can monitor the room temperature, room humidity, heart rate, blood oxygen, and body temperature.

Now I want to monitor it on my smartphone. So simply, again, enter the same IP address. So you can see the same parameters are again displayed on mobile phones as well. So this was all about the best project related to IoT based patient monitoring systems.

IOT BASED PATIENT HEALTH MONITORING SYSTEM USING ESP8266

We'll be presenting you health rate monitoring over the internet using IoT devices. Rdino and things speak. Actually, we are using a call sensor and temperature sensor that will measure the VPN value and temperature and we'll be sending it over the Internet using ESPN266, and you can check if the data is staying in any part of the world. So let's get it started. So let's learn about the pulse sensor.

This is a pulse sensor from pulse sensor.com that has a diet at the top of the point that it takes the blood capillary contraction and relaxation, which measures the pulse rate. This is the front side. Who did it that took the skin? And this is the backside which glows. You can see there are 3 different pins. The first pin is brown.



That is black in color. then a PC pin and analog pin that is given to the analog input of Ardeno. You know? This is how the gripping is done on the pulse sensor. The pulse sensor that has hardship is the place where the finger is placed.

This is an lm35 temperature sensor. which will measure the temperature. The first bin is vis a vis 4 to 20 volt input. The second one is output and the third one is ground. This is a very accurate temperature sensor and does not require any calibration.

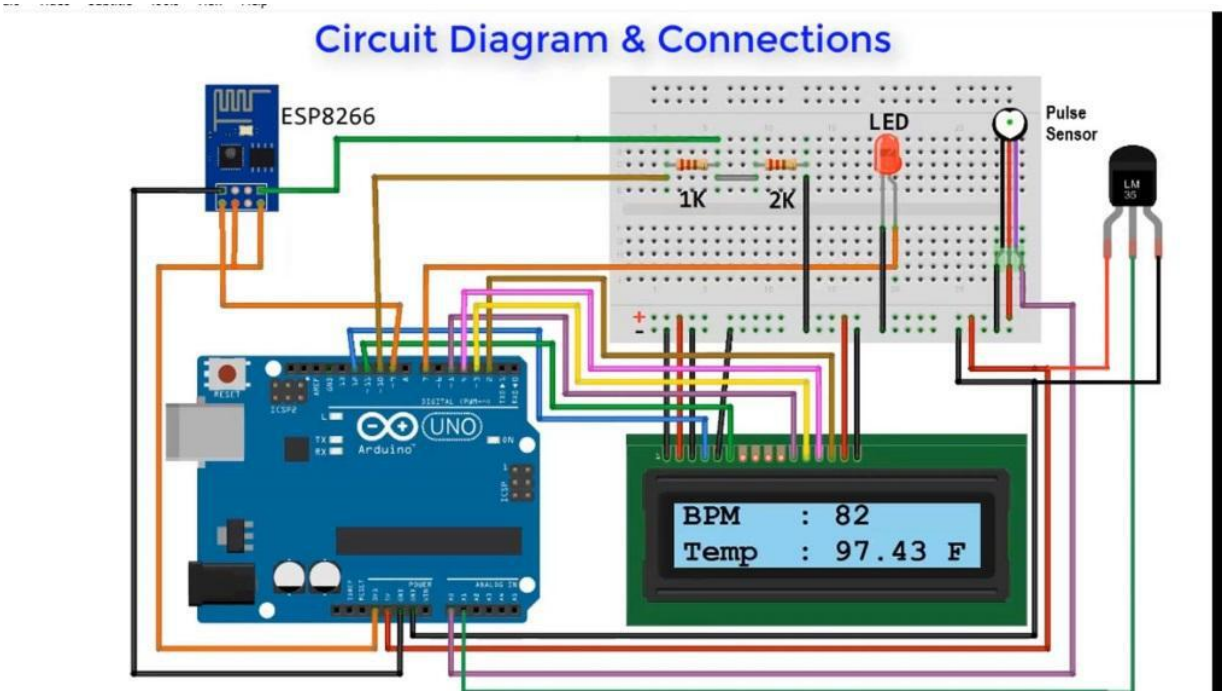
We'll be interfacing this pulse sensor with ESPA266 along with a temperature sensor. ESP at 266 has 8 pins. The pin 1 is ground and pin at each PC pin which requires 3.3 volts. We have TX and Rx, and there are 3 gpi open. That is gpi00, gpi01, and gpi02.

The components required for this project are. ESP a 266 Wi Fi module, pulse sensor, LCD display, a register, LED, red, blue, white, an L M Thirty 5 temperature sensor used to register 1 of 2 kilo of another 1 kilo. This is a circuit diagram. We have connected the number 12115432ofrdin02lcdp. and analog pin.

It is where anyone is connected to pulse sensor and LM35 temperature sensor. t x and r x of ESB at 26 are connected to the number 9 and 10 of are, you know, you know, but we have used 2 k and 1 k registers for dividing the voltage. For example, ESPN to 661. 3.3 volt. So two can 1 k divided voltage.

This is how we have assembled the circuit on a breadboard. We have our, you know, you know, ESP266. pulse sensor. And LED for indication. system to 2 LCD and a temperature sensor.

Now we will connect This circuit to the computer. This is the code. The coding is 12.1152 LCD pins, and we are measuring pulse and temperature. So this is a string. We need to input this string.



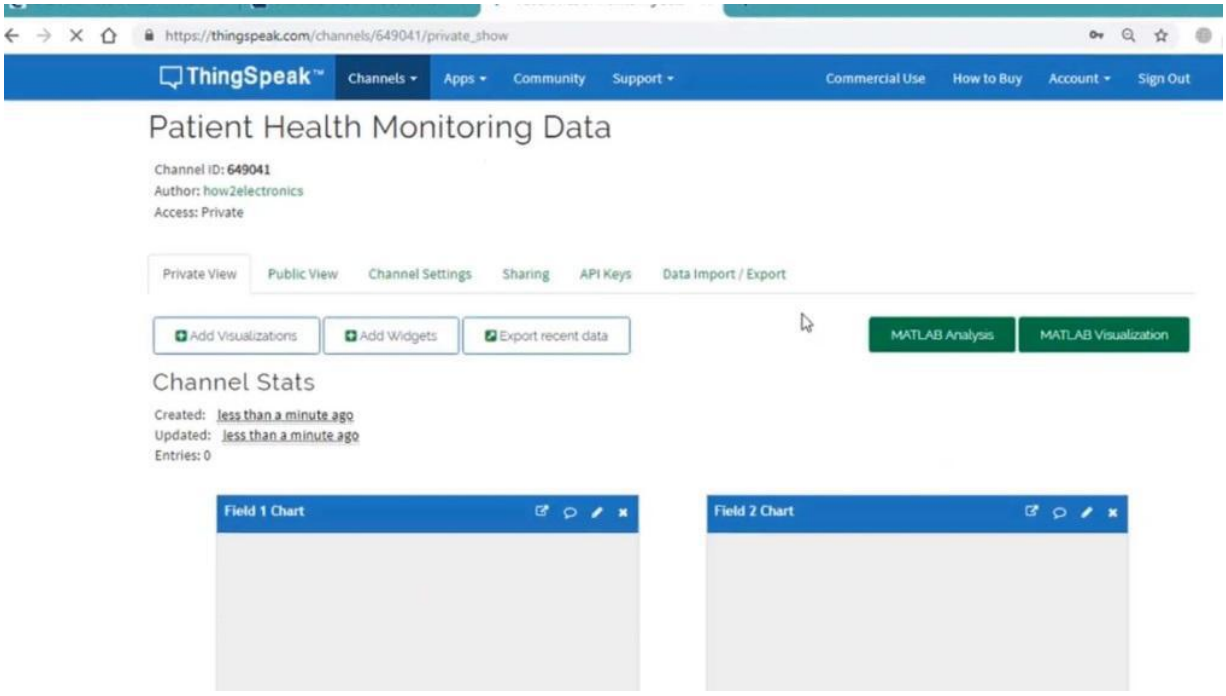
I'll explain later how to enter this string. A word to get this string. Paul's PIN is inert. A little blinking is 713 is the default PIN that will blink. Since the programming is very long, I'll not be explaining each and every point.

And this final thing is about measuring temperature. That is how the temperature is measured by a one p. Go to things speak.com. If you

haven't created the ID, click on sign up, else, if you have connected, click on sign in. Enter your email address and password.

And finally sign in. Create a new channel. Write the name of the channel. For example, I will be giving patient monitoring data. We have measured 2 fields: pulse rate and temperature.

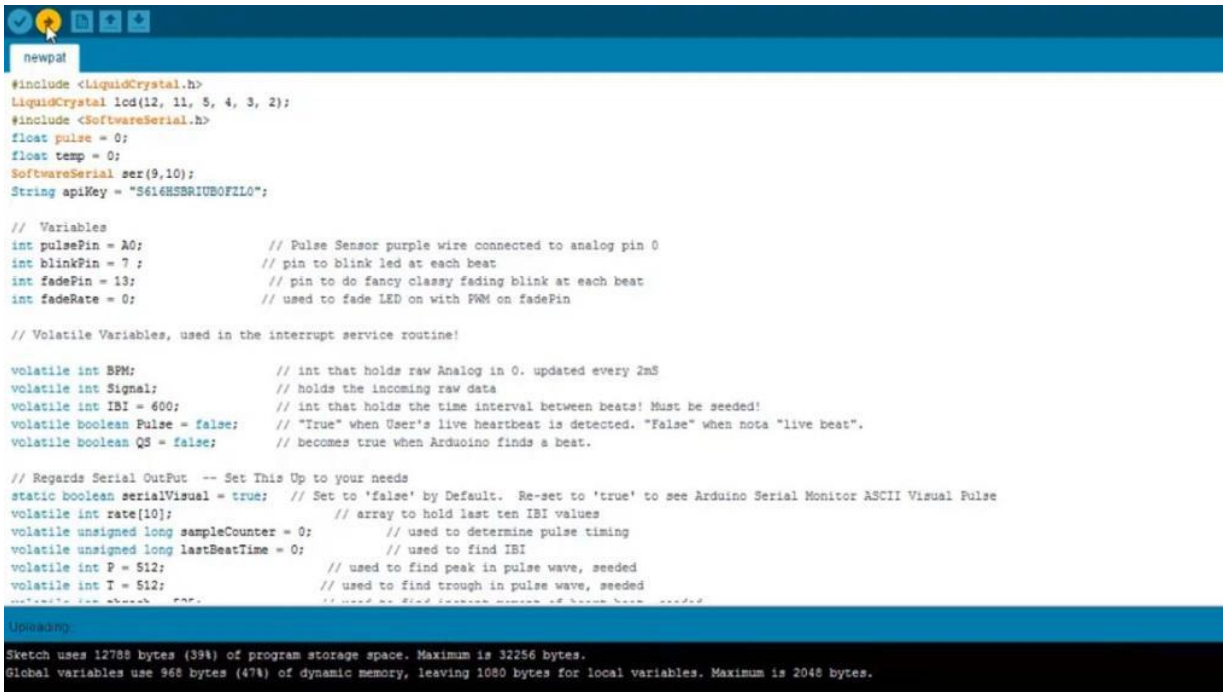
So give pulse state in BPM and temperature in Fahrenheit. After that, we need to add widgets as well. So click on edit with just. Click on goals. Now click 2 different or, sorry, make 2 different get this.



1 for pulse and another for temperature. What is the minimum pulse? It's 360, and I am giving 700 peak intervals should be 22, 30. Anything, whatever you like. And since IOT data is valid, updated after 15 seconds, so select it to add 15th.

Now you can see the pulse rate gauge is designed. Now we need to edit the gauge itself. Temperature as well. So I'm giving the name body temperature. It is for field 2.

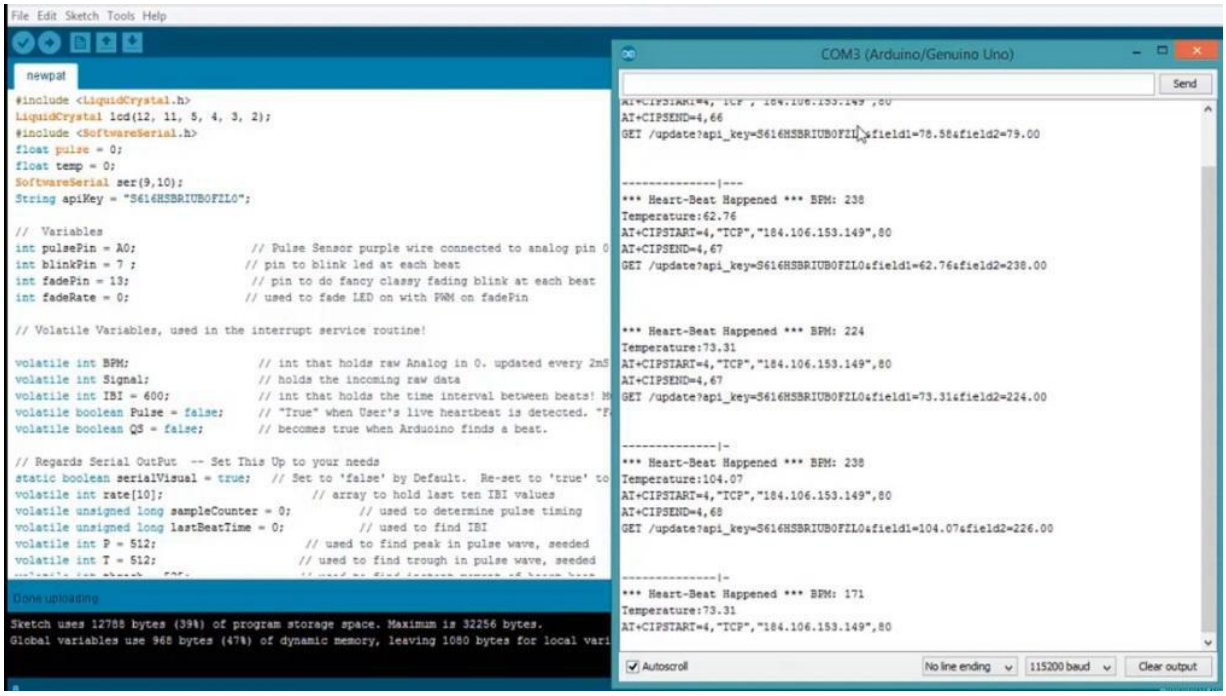
So the maximum body temperature is 60 to 120. I'm assigning the value. You can assign anything according to the accuracy. The unit is for a height. So give it a night.



Take intervals. Give anything. So We have 2 pages created. Now go to API keys. Copy this key.

That is the right API key. Paste it over your admin program. Now, compile this code and upload it to your Ardeno, you know, board. So I'm directly uploading it. So the uploading is done.

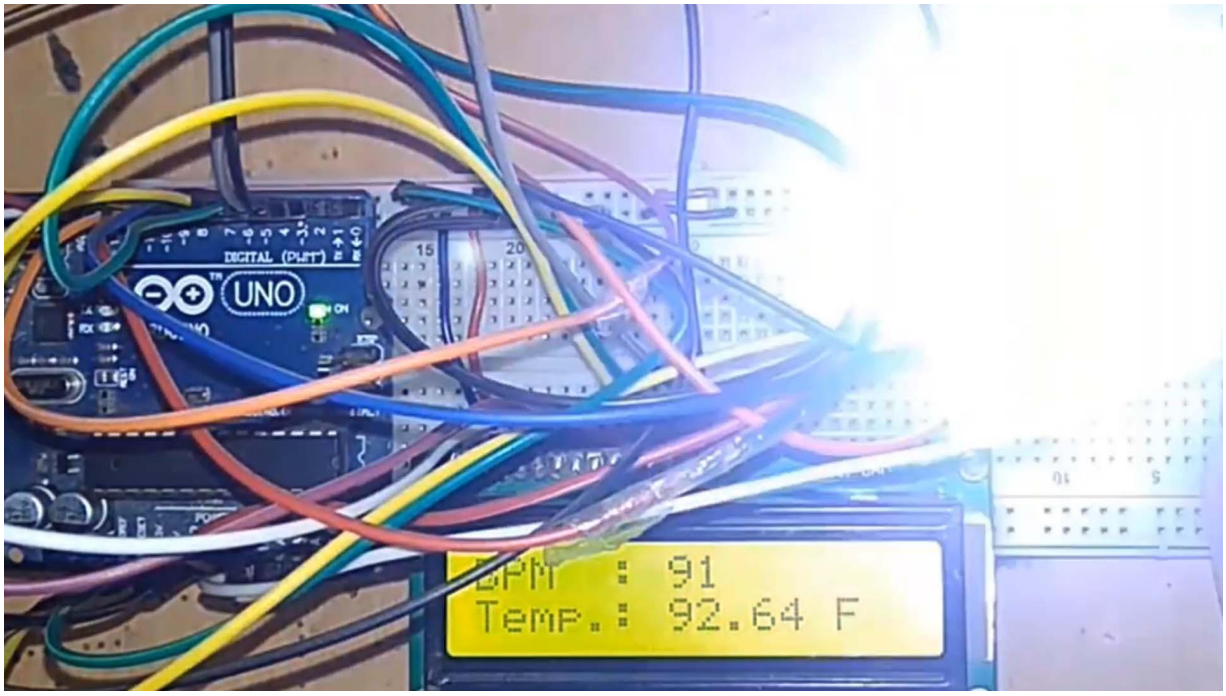
Now, after you upload, you will see This one on your LCD screen. It will initialize and it will get data. getting data might take a few seconds. Now, put your hand over both the sensors. That is the pulse sensor and temperature sensor.



So you can see the BPM and temperature put on a display respectively. No. Open the serial monitor and select about 115200. So you can see it is displaying both the BPM and temperature. And also, you are seeing the gate update API key.

Now this is very important. If it is not displayed on a serial monitor, no data will be sent to the internet. So make your gate API key updated on this serial monitor. So it is actually working finally. Now the same data will be displayed on things to speak.

So go to the private key. we'll be seeing the data. What is it recorded? You're even seeing the body temperature. PIN is moving, and the BM pin is moving as well.

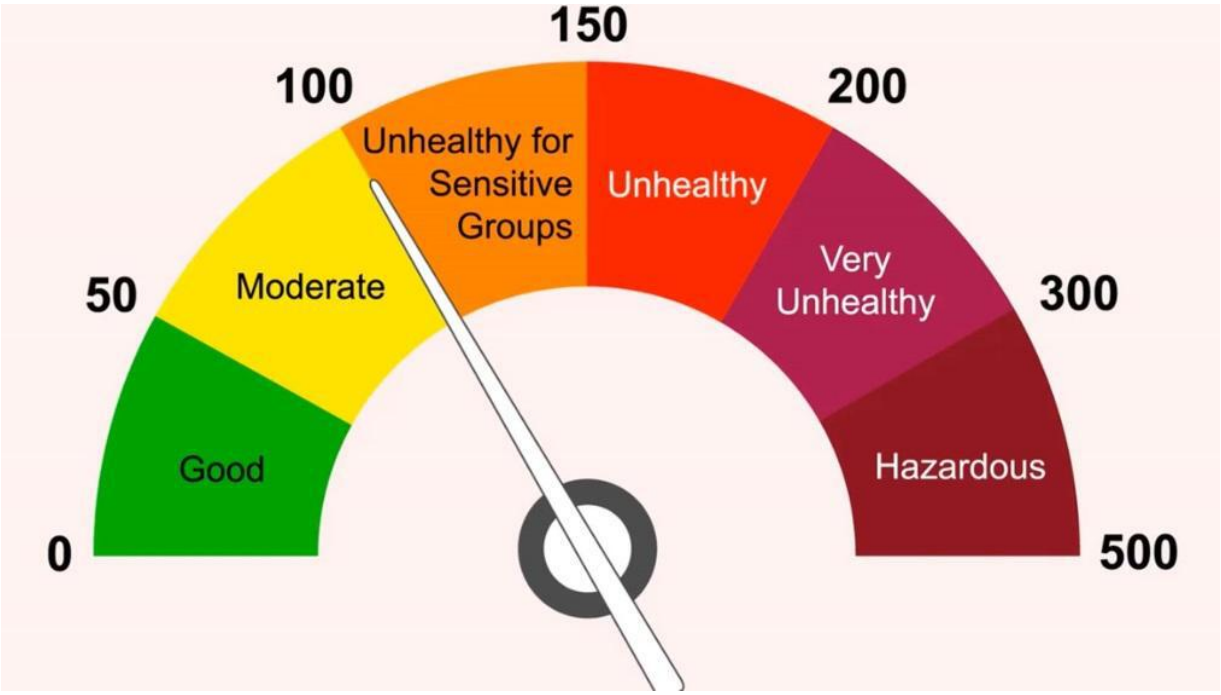


depending upon the chart. I have fastened the project because after 15 seconds, the data is uploaded. So you can see the same BPM and temperature is even displayed on the LCD screen. Because of the calibration issue, the VPN and temperature values are rising or behaving so differently. This is the data that you are seeing. So you can monitor this data staying in any part of the world. If you have a field ID or channel ID of your IOT thing.

IOT BASED PM2.5 MONITORING WITH AUTOMATIC AIR FRESHENER SYSTEM

In this project, we will design an IT based particulate matter monitoring system along with an automatic air pressure system. The device can monitor and measure PM 2.5 and PM 10 concentration in the year. It can also activate the air freshener when the PM level exceeds the danger value.

The device can be used in the house, industry, schools, office, and around crowded places. Partic matter refers to microscopic particles of solid or liquid matter suspended in the air. PM tank is particulate matter with 10 micrometer diameter or less, and PM 2.5 means particulate matter with 2.5 micrometer or less. We proposed an IIT based system for measuring and controlling air pollution due PM concentration. The device can be embedded on a small PCV.

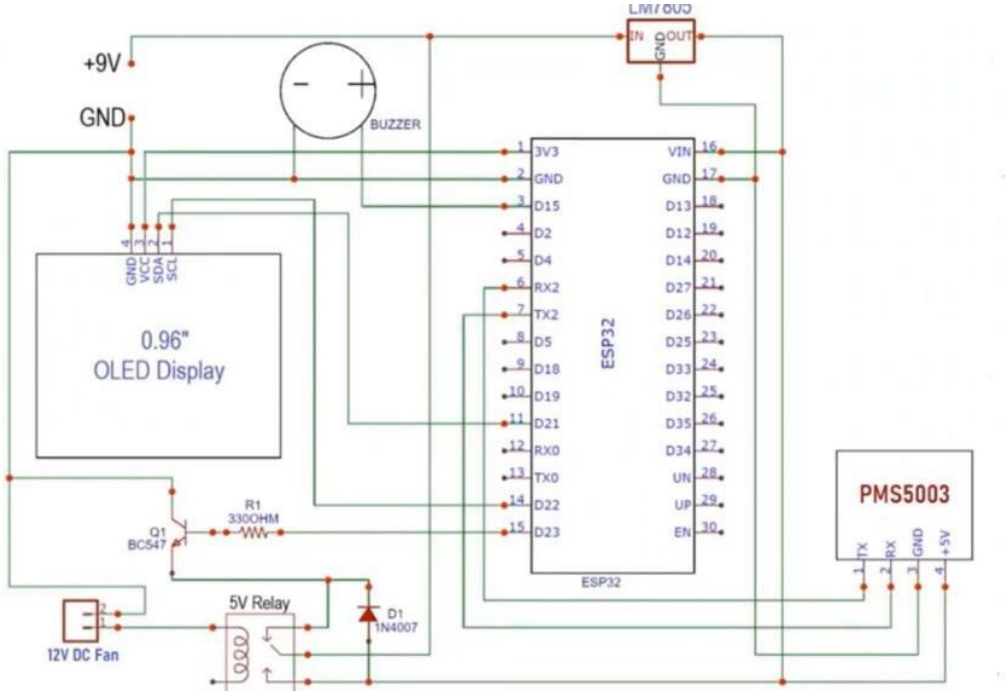


The device went powered on, started measuring the particulate matter concentration in air and displayed the real time value on a 0.96 inch or later display. The ASP 32 also connects to the wifi network and uploads the data to the web server. Using the local IP address of ESP32, you can get the pm 1.0 pm 2.5 and pm 10 value on any web browser. The air quality index value, anything above 50 is unhealthy for breathing. So, when the pollution level exceeds a year, the fan turns on automatically to blow fresh air.

Instead of a fan, you can use any other ear pressing device like a humidifier that can be connected to the relay. So let's begin with making this great project. The sponsor of this project is busy the next day. Currently, they are having a great offer for new users. You can get $100 for free from the next PCV.

This $100 can be used to order the PCV as well as the PCV assembly service. You can also go for the PCV assembly as they have the latest modern tools that can assemble any type of PCV including flexing PCV or four layers with any type of component package. Currently the next piece of it is doing a sponsorship activity, they will sponsor the creative post with 3 pieces of it and

components. They will also offer winners free shipping coupons and other prizes. I have put the intern link below the project kit involved in it and one more think they're also doing a giveaway to celebrate the Chinese New Year. The prices are 37 in 1 Art Junior kit, 45 in 1 sensor module board kit, and some coupons.



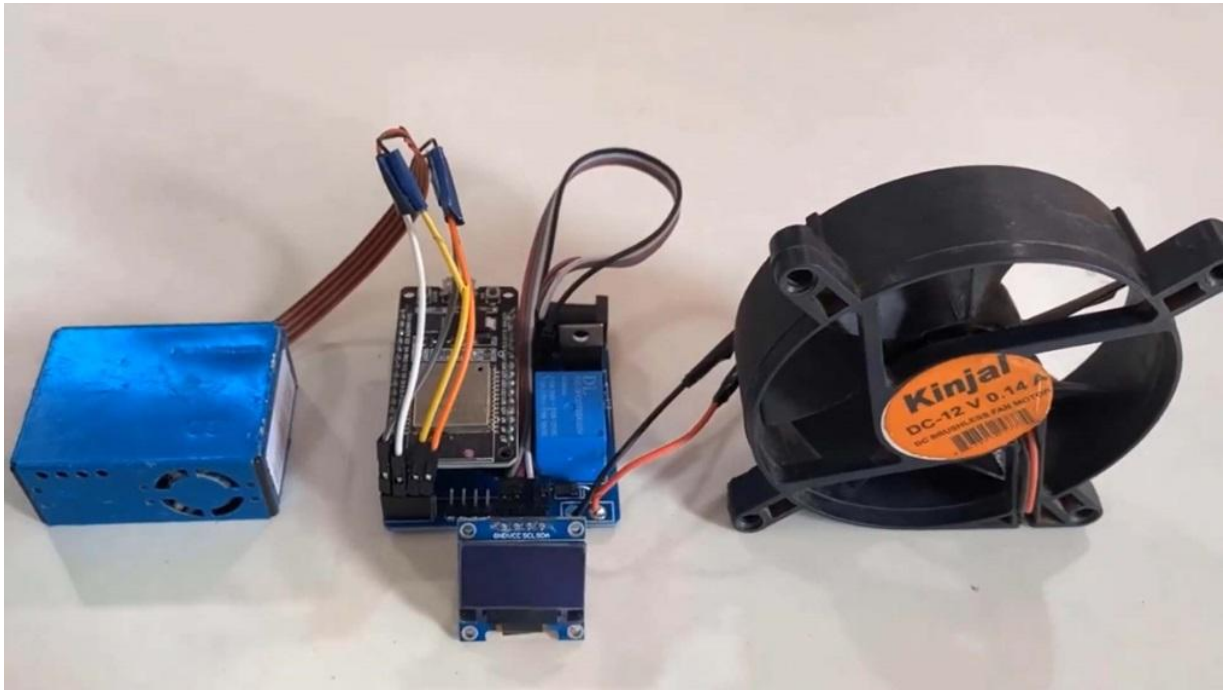
follow their Facebook post from the description link. Welcome back friends. Let's just say the components required for this project. The first important component is the PMS 5003 sensor, which measures particulate matter concentration in air and has 4 pins UR interface. We will then need a 0.96 inch I Square C OLED display.

Similarly, we will need a 12 volt DC fan as a demo. You can use a humidifier as well, and then we need a power supply unit. For this, I used a 9 volt or 12 volt adapter. Apart from these components, a long list of other materials is required like ESP, a word relay, some voltage regulators, diode, resistor, buzzer transistor, and some jumper wires. All these components can be easily purchased from Amazon.

Alright. Now let's just say the hardware design and assembly. This is the hardware unit schematic which has been designed in an easy ADA schematic and PCB designing tool. Don't be so panicked because of so many connections. I have already designed a custom PC board for the project.

You can use the PCV and assemble all the components on it. So this is how the board looks after all the components are assembled. The device size is small and compact. The board has 2 female headers for OLED display and a PMS 5003 sensor. This keypad is not required.

You can ignore it. You can solder the DC fan or any humidifier unit here. The device can be fitted in any 3 d casing from its spec site. Okay. I have connected the OLED display and TC fan here.



So, the entire device looks something like this. Please pick your full while connecting the PMS 5003 and OLED display as the pins might be different. The device now needs to be powered by a 12 volt adapter, so I connected the DC jack. As soon as the device is parted, the ESP32 will connect to the wifi network, and the OLED

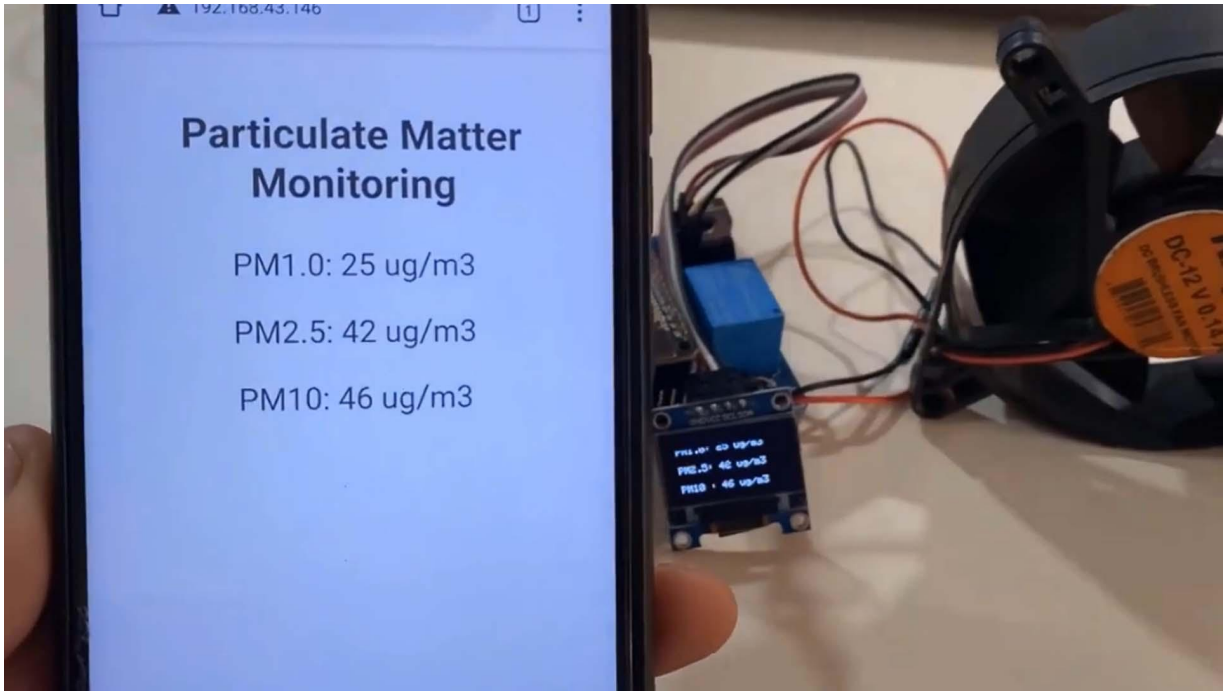
display will display the local IP address of the ESP32 board. Then you can see the pm 1.0, pm 2.5 and pm 10 values in the display as 0s.

This is because the PMS 5003 requires some time for getting heated. As soon as its gate is heated, the fan inside it rotates and starts collecting the external air sample data. At this moment, the value of PM 1.0, pm2.5 and pm10 is less than 50 micrograms per cubic meter. The center gets that the air inside my room is very fresh and pure. At this time, the DC fan that is connected through the relay is not active.

This volume gets active if the PM concentration exceeds threshold value. I have set the value as 150. Now to make changes to the PM values, you need to introduce those particles, smoke, or any gasses near the sensor. I chose incense sticks to produce smoke, so let me put it near the sensor. Now the value has exceeded very high as seen in the OLED screen. The fan is also active now.

This is because the threshold value is crossed and the fan has to turn on to purify the air. This fan will turn off again when the sensor beam values reach less than 150. So now it has gone below 150, and so the fan has stopped completely. This is how our entire system works. You can now use the IP address and paste it into your web browser or computer or mobile phone.

After hitting enter, A page will be displayed where you can see pm1.0, pm2.5, and pm10 data. You don't need to refresh the page again and again to see the data as data is updated automatically after a few seconds. This is because the code already has the HX script so that we can request data from the server asynchronously in the background without refreshing the page. Finally, let's say the code part now. So the code requires inbuilt wifi and a web server library, you also need PMS and OLED library for this code.



The 3 parameters are defined for all 3 pm values. The relay and positive line is defined as ZPA O 23 and ZPA O 15. Using the wifi SSID and password from here, then using the library function, we are calculating the quantity of particulate matter concentration in air. You can change the threshold value from here. Currently it is safe to 150, you can use anything that you desire for.

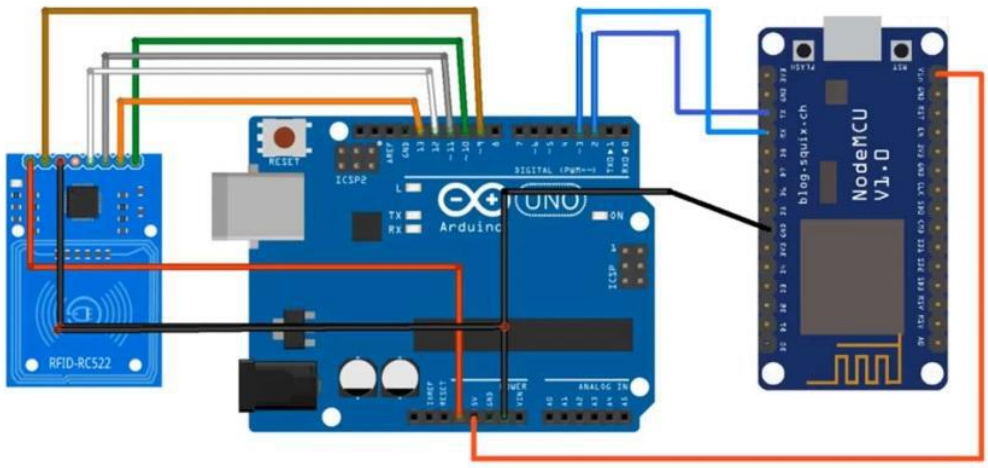
The rest of the things are related to HTML requests and web based data. This part contains a JX code for auto retrieving data and these are the lines for displaying beyond values on a web page. The region tutorial for this project is given in the website article, and it also contains the information of component purchase link, the circuit diagram connection, as well as the Pacific Harbor file link. The source code and library can also be found here Well, that's all from today's project.

IOT BASED RFID ATTENDANCE SYSTEM USING ARDUINO ESP8266

You will learn how to make an IOT based RFID attendance system using node MCU, arbino, and adafruit.io platform simply by using MQTT broker. So we will use RFID MFRC522, arbino nano, and node MCU 12b board. Arvino and RFID scanner scans the RFID cards and then logs the data to aDAFRuit IO cloud platform with the help of a wifi module. This information can be displayed in the Adafruit IO dashboard and can be accessed by the required authorities to view and analyze the attendance over the internet from anywhere at any time. So let's get started.

Circuit Diagram & Connection

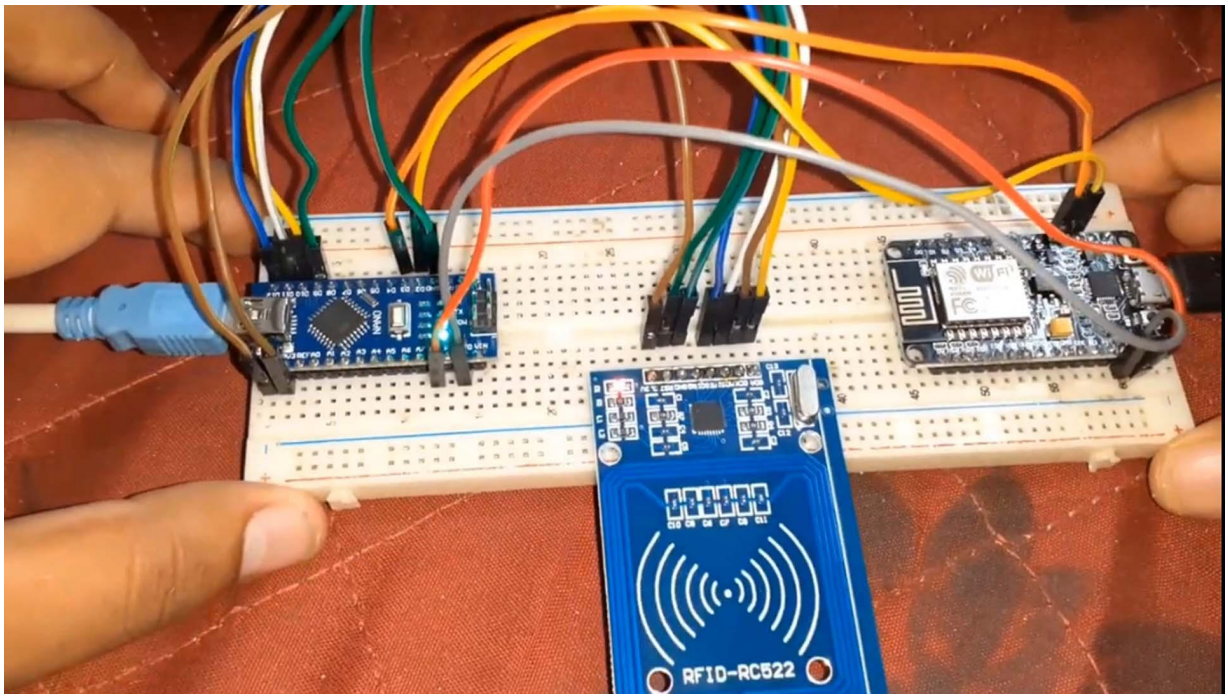
Pin	Wiring to Arduino Uno
SDA	Digital 10
SCK	Digital 13
MOS	Digital 11
MIS	Digital 12
IRQ	Unconnected
GNE	GND
RST	Digital 9
3.3v	3.3V



Let's first start with the circuit diagram. We just need our Reno Nano node MCU board in MFRC522 RFID. So the connection diagram is exactly the same as shown in the figure. connect as a pin of RFID to Arduino Digital Pin 10. Similarly as CK to D13, MOSI to D11, meso to D12 GND to GND RST pin to deny and supply 3.3 volts power using 3.3 volts pins.

IRQ is not connected. Similarly, connect nodeMCU TX Rx to Arvino D2 D3 Peans, respectively. also supply power to node MCU by connecting its VCC and GND. So here is a hardware complete setup. You can see our Vino nano, node MCU board, and RFID scanner here.

All are connected as per circuit diagram simply on a breadboard. MFRC522 is an ASPI module connected to Arvino. No MCU is connected via UR teams. Here are a few RFID cards actually totaling 6 cards for six people. All of the cards have a frequency of 13.56 kilohertz.

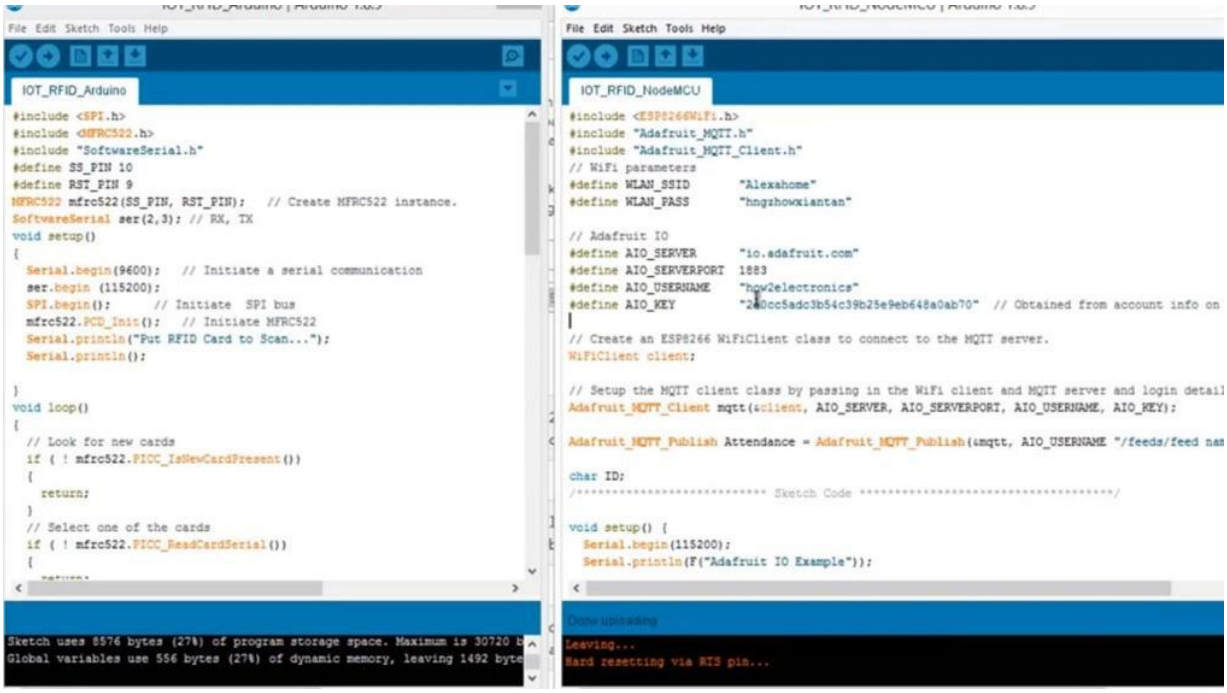


We need to set up a DAF route account first. but ADAFRuit IO is an open data platform that allows you to aggregate, visualize, and

analyze live data on the cloud. using Adaptroot IO, you can upload, display and monitor your data over the Internet and make your project IoT enabled. For aDafruit IO setup, the first thing you will need to do is to sign up to aDafruit IO. To sign up, go to aDafruit IO site IO.

fruit.com and click on get started for free on the top right of the screen. After this, a window will pop up where you need to fill in your details. I have already created the account. I will simply log in here. So once the account is created, simply go back to Iodotadafruit.com.

On the left side, click on view a i o key. From here, you need to copy the username and also the AIO key. So copy these keys simply on notepad as we will need this late to use in the program as I am doing here. Just copy and paste. Now on the left side, click on the dashboard and create a new dashboard using any name.



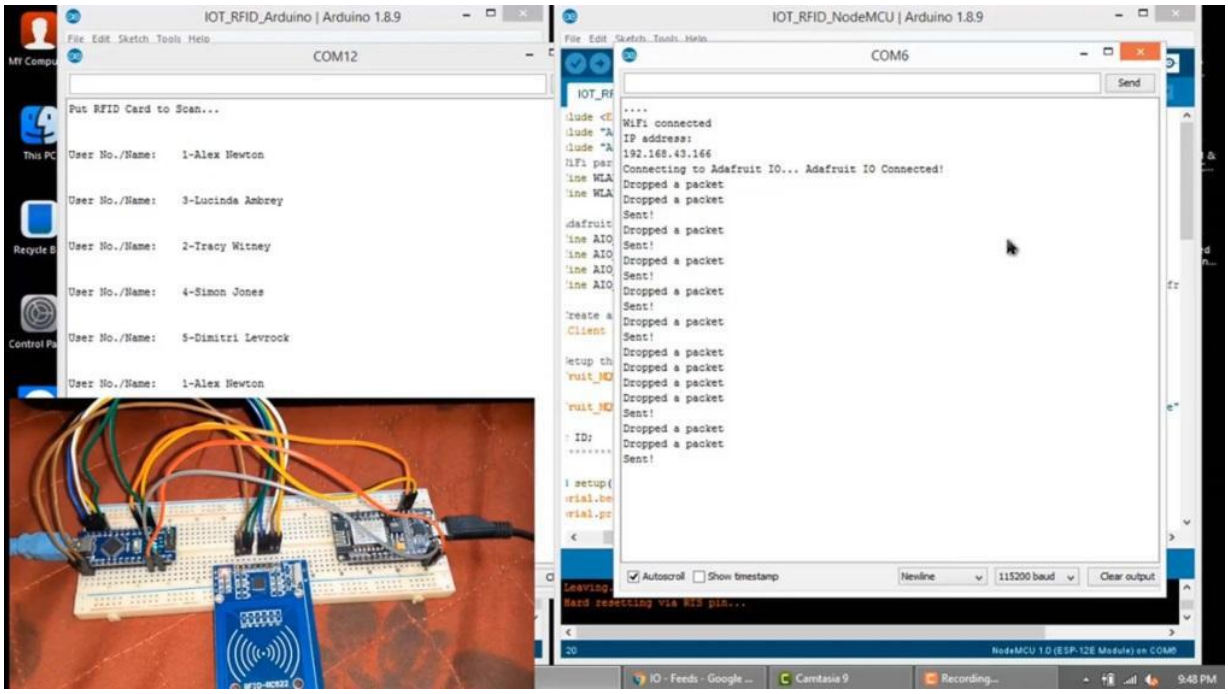
Example, RFID _ attendance. You may or may not give the description. So once the dashboard is created, there is no data currently as we haven't connected our device. Now similarly go to feeds and create a new feed. Give the name to it.

Now copy this name and do a slight modification in the program. Just paste it to this part of node MCU code. So we have 2 quotes. 1 for Arvino and others for the MCU. I will explain the program later on.

For now, upload the code to both the boards and then open the serial monitor. So now you can easily see what is happening. The modem crew is connected to WiFi and ready to transfer data to a DAF dashboard. So let's scan some of our cards and see the data transfer. You can put any number of card details on the code.

But for now, let's scan all the cards here whose details are there in the code. Now let's go to the feeds again on aDAFRuit IO dashboard. Here click on the feeds that you created earlier. Now this looks great. All the attendance with time and card number are stored here.

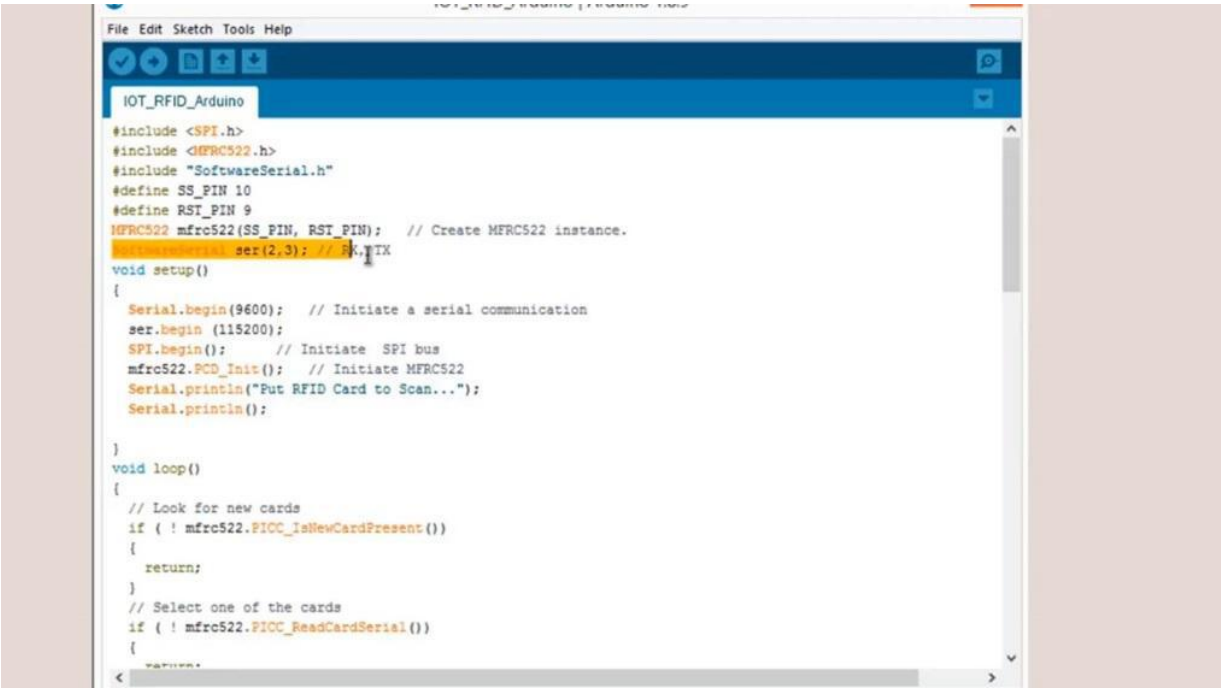
So you can see the date and time as well as the card number on the right side that is assigned to a particular person. You can also see the graph here though this graph is not necessary for us. Similarly, you can also download the data on an Excel sheet for recording attendance. So we have 2 quotes, one for our other for node MCU. Let's see the arbino code first.



Mfrc522 library is used for the RFID scanner. SDA and RST Pines are connected with 10 and 9 pins of the arduino as shown in the circuit diagram. You can download the library as zip from below Lincoln added to your. Inside the set up function, we initialize the serial communication at 9600 and software serial at 115,200 board rate. We have also opened communication with MFRC522.

Inside the void loop function, we will check if a new card is present. And if a new card is present, then it will check the UID of the card. For a valid card, it will print the UID of the card on the serial monitor. Otherwise, it will print unauthorized cards. These are the RFID numbers which your RFID card has.

There is a simple program in examples to identify these RFID numbers. You can find and paste it here. Similarly, sir, dot right function has number 1, 2, 3 set these numbers. serial number which will be uploaded in a DAFRude dashboard. You can assign these numbers with a few names as you can see here.



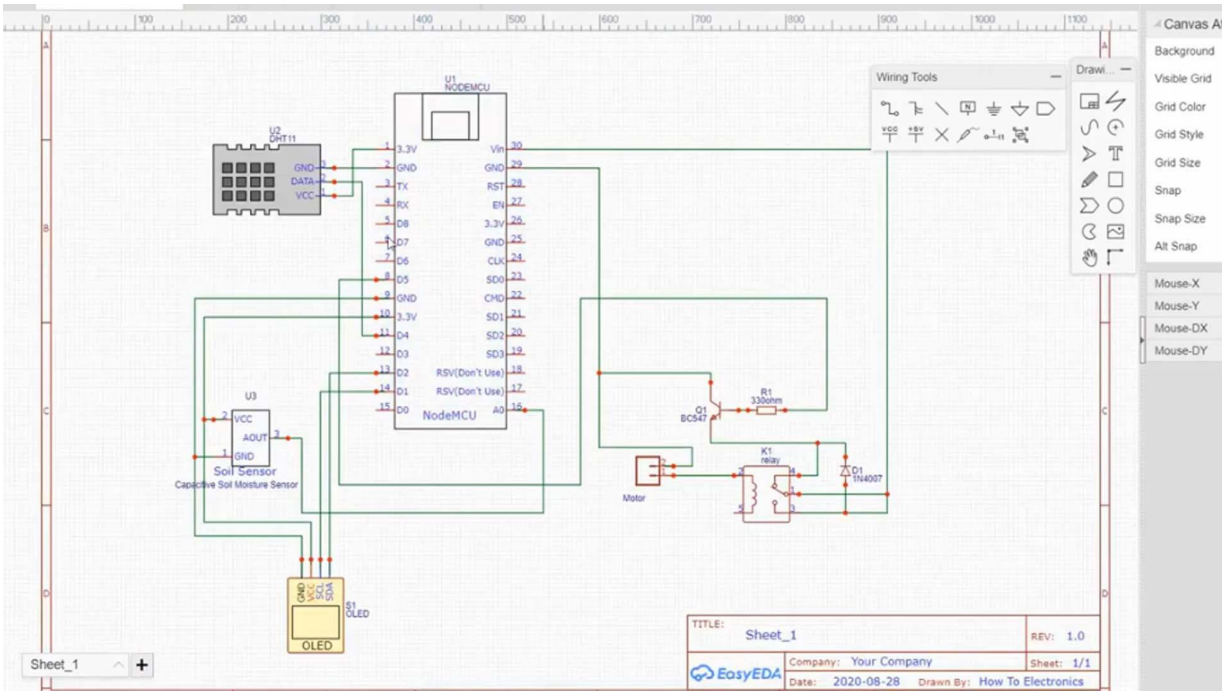
These names will be displayed on the serial monitor while scanning. So select the arbino board that you are using and upload it. Let's see the MCU code now. Begin the program by adding the required libraries. You can get the MQTT library from the library manager or simply download the zip file from our link below.

After the center, the Wi Fi credential and password with the Adafruit IO username and a i o key that we obtained earlier. Using the setup function, we will begin the serial communication at 115,200 board rate. Then we will connect our WiFi client to the Adaptive IO server. Inside the void loop function, ESP will read the data from the serial and then publish it to the route IO server. Inside the void connect function, the conditional assignment is given. Now select the correct node MCU board and upload the code.

IOT BASED SMART AGRICULTURE AUTOMATIC IRRIGATION SYSTEM WITH ESP8266

In this project, we will make IoT the best smart agriculture with an automatic irrigation system. Here, we will use a capacitive soil moisture sensor to measure moisture content present in the soil. We will also use the Dht11 humidity temperature sensor to measure the air temperature and humidity.

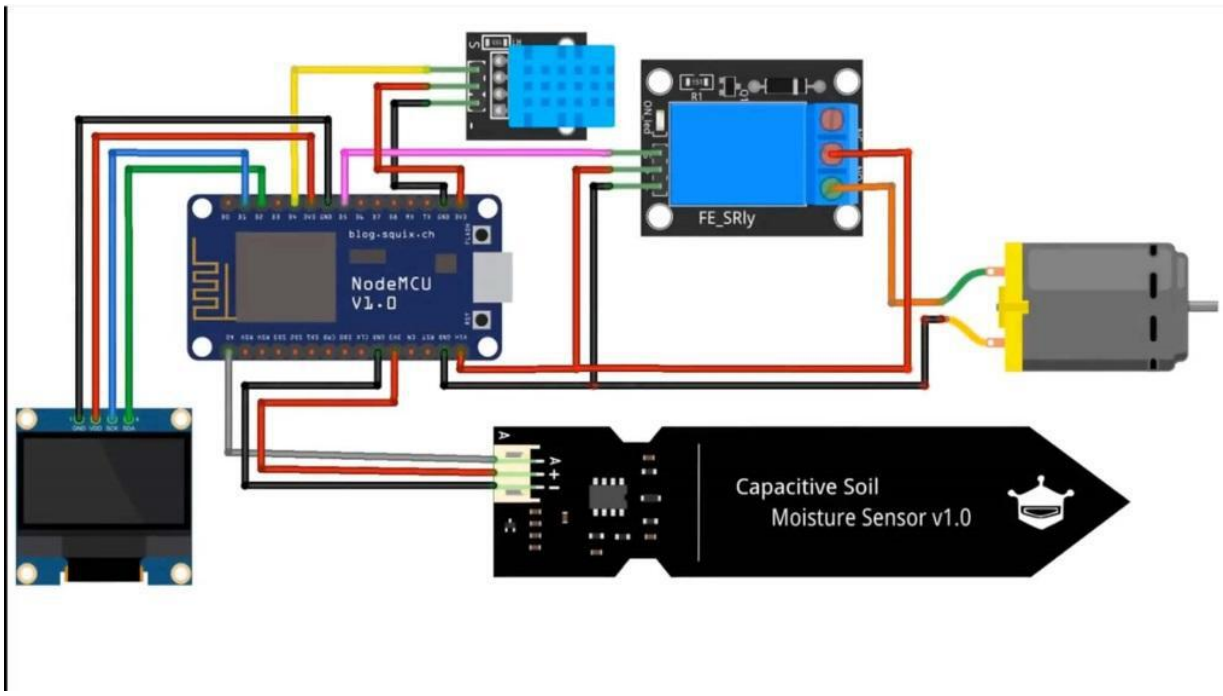
Using a 5 volt power relay, we will control the water pump. Whenever the low quantity of moisture is detected in the soil, the motor automatically turns on and, hence, irrigation is done automatically. Once this well becomes heavy, the motor turns off. All this happening can be monitored and remotely through fixed big servers online from any part of the world. This project also requires a custom PCV.



So I used an easy idea to design the custom PCV. First, I designed the schematic, then I converted the schematic into a PCV. So you can see the three d view of the PCV front side as well as the back side. Then I generated the Gerber file. Now, you can visit the Pacific Office website and order the PCV at a very cheap price.

All you need is to upload the Gerber file. Then view your PCV anchor reviewer. Then select the details like Pacific quantity, Pacific color, and thickness. Then select the country of shipment and then place an order. So that's all.

Currently, the next piece of it is giving a biggest discount with literature and free order. Link is given in the description. There is 20 percent of our PCV orders and 100% lottery draw winning rate. You can get a free PCV order, a free SMT order, $30 to $100, and a $10 to $5 coupon without limit. Click here for that.



Go through these easy rules for getting participated in this lottery draw. Apart from this 20% of Pacific orders and 15% of SMT orders, you can go for PCV prototype mass production, multi year PCV board, For this project, we need to note MCU board. A capacitive soil moisture sensor A 5 fold relay module. 0.96 inch or late display. DHT 11 sensor, break port, and a five volt DC pump motor.

The motor is used for drawing water from inlet and drawing water through the outlet. A pipe can be connected to it. So now here is the schematic of the project I used freed zinc to make and the schematic, which is beautifully designed. Just please unconnect the component. It's super easy.

So the final schematic is ready. So a moisture sensor is connected to a not of node MCU and DHT11 to the 4 pin. The motor is connected to a relay. The relay is controlled using a T5 pin of node MCU. The OLED display is connected to a 12 c pin of node MCU.

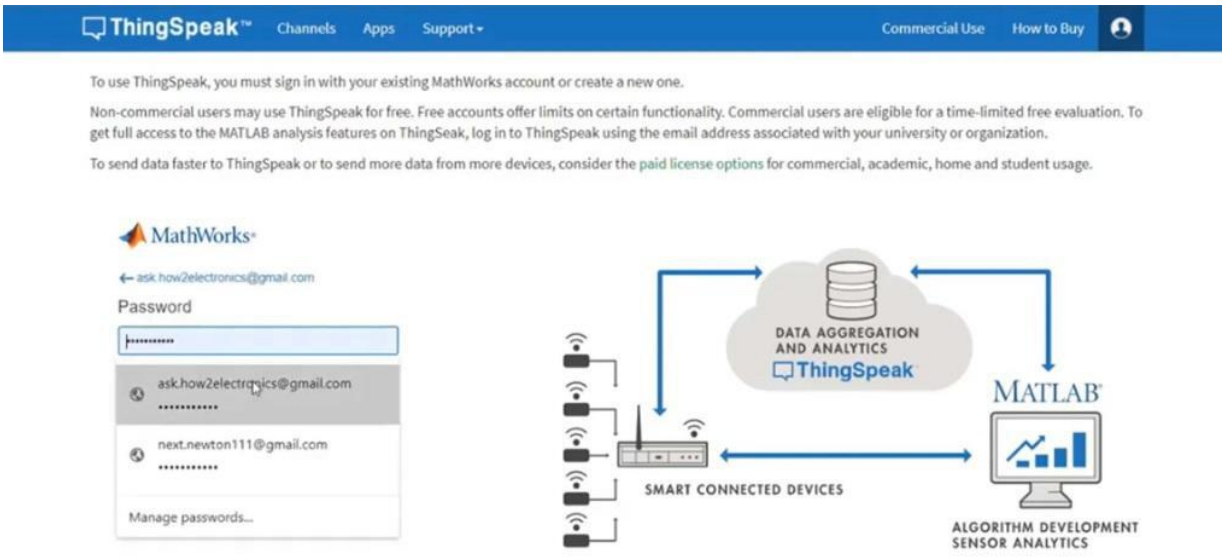
So here is the final assembly on the breadboard. I highly recommend you to do this project on a PCB board. The breadboard connection might give you some issues. For demonstration purposes, I use the

breadboard for assembling the socket. So I have a small tube with soil and grass growing it.

and the tube with water. The tube is a water source from where the motor can draw water whenever it's necessary. This is a soil moisture sensor, DHT11 sensor, and a three way module. OLED display. Note MCU board and a water pump.

This water pump needs to be fully submerged in water. The outlet pipe is kept in a field for irrigation. Similarly, soil moisture sensors are deep in soil. As soon as the device is powered, the OLED will start displaying the soil humidity, air humidity and also air temperature. It's showing the real time data.

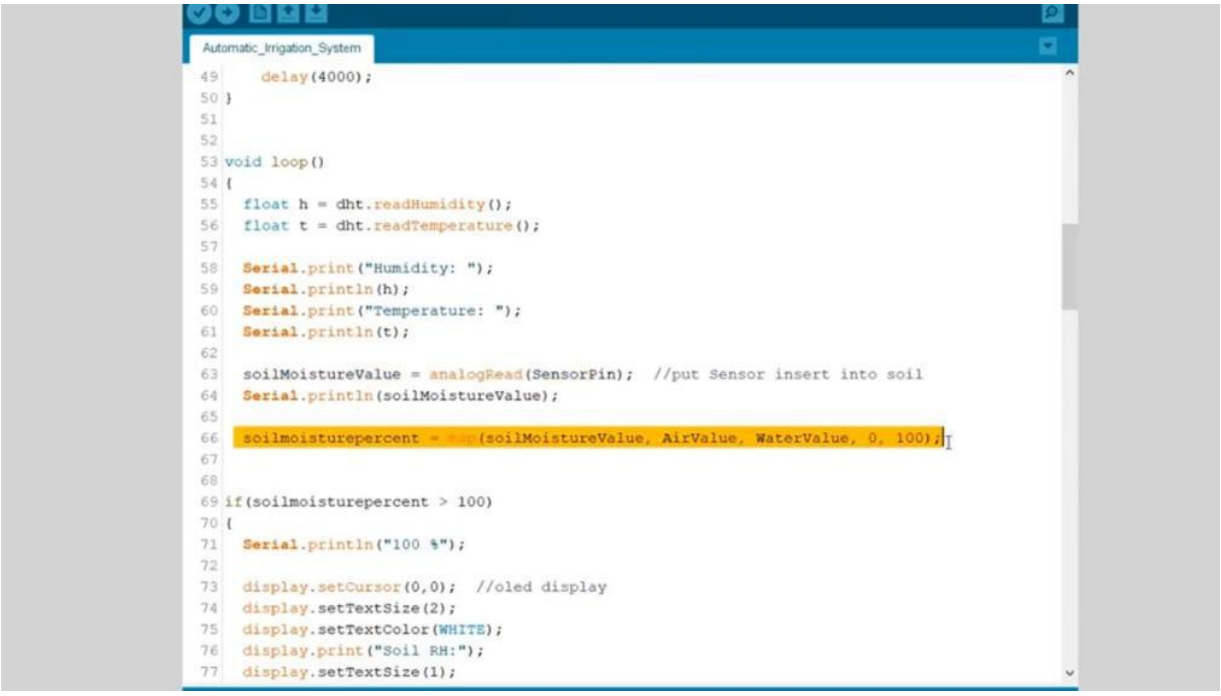
As you can see here, when the soil moisture content is reduced, the water pump turns on and irrigates the field until the required moisture is achieved. You can check its full working here in this project. Now what I want is to monitor the data online on the Internet from any part of the world. So create an account on ThinkSpace server. As I have already created an account, as I simply sign in.



After that, create a new channel. Write the name of the channel and also feel the 4 different fields as soil moisture, humidity, temperature, and motor status, and save the general. Then go to the API keys and copy the right API key. This API key is required in code, and you need to pass it here. Now let's set the code part. We need a few libraries for that.

All the libraries link is given below. The code is pretty simple and easy to understand, but you need to make some changes in a few parameters. For example, change the API key, change the Wi Fi access ID and password. Sure. We are uploading the data to things.

PIX Arbor. This line is used for retrieving the value of air temperature and humidity. Similarly, we are reading the soil moisture data from soil moisture sensor and converting it into percentages using a map function. Then we are displayed the values on the OLED screen using some built-in library function. These lines are for turning on and off the motor when a particular threshold is achieved.



I have defined 30% as a threshold value You can use yours depending upon the application. Now using the post string function,

we are uploading the 4 data to thingspeak server. You can see from field 1 to field 4 as soil moisture, humidity, temperature, and release status. You need to upload this code now to the node MCU board. So select the right board and then a right port and hit the upload button.

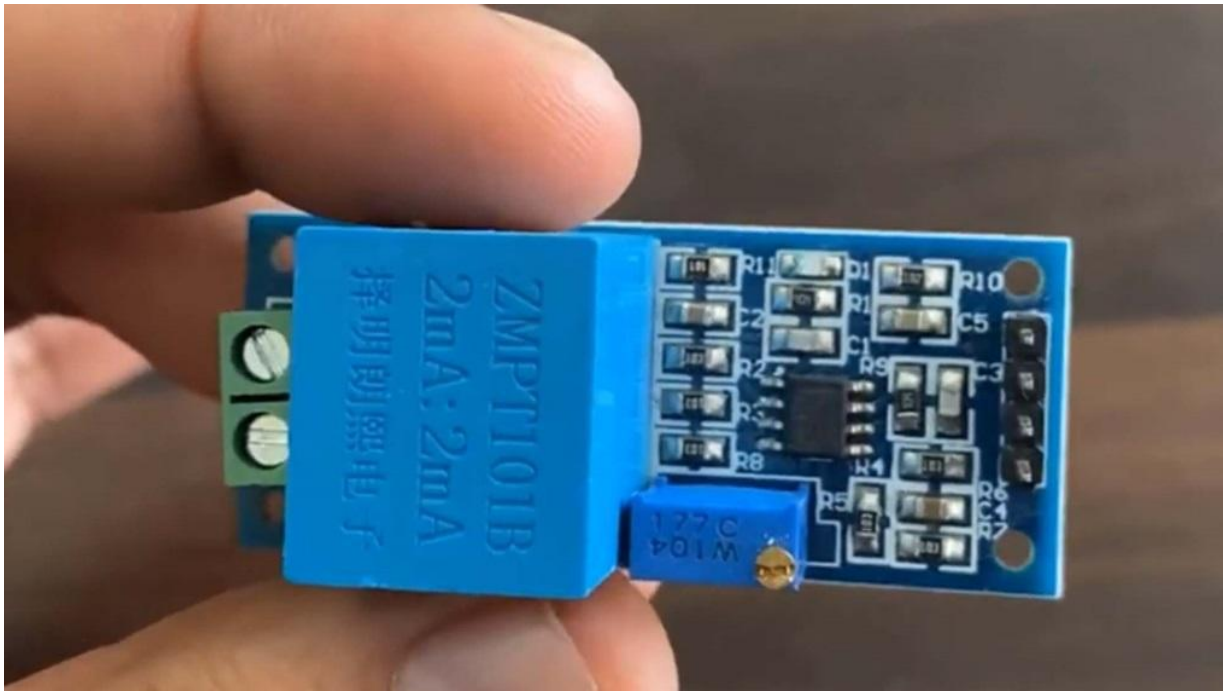
So once the code is uploaded, you are done. You can go to the private view of the ThinkSpace server. Here you can see the status of soil moisture, humidity, and temperature as well as relay status. So that's all from today.

IOT BASED SMART ELECTRICITY ENERGY METER USING ESP32

In this new project, we will learn how to make our own IIT based electricity energy meter using ESP 32 and monitor data on the blink application. With the current technology, you need to go to the meter reading room and take down readings. Thus, monitoring and keeping track records of your electricity consumption is a tedious task.

To automate this, we can use the internet of things. The internet of things saves our time and money by automating remote data collection. Smart Energy has received quite a lot of acclaim across the glove in recent years. So we need to build our own ID based electricity energy meter. For this, we need to select the current sensor as well as the voltage sensor.

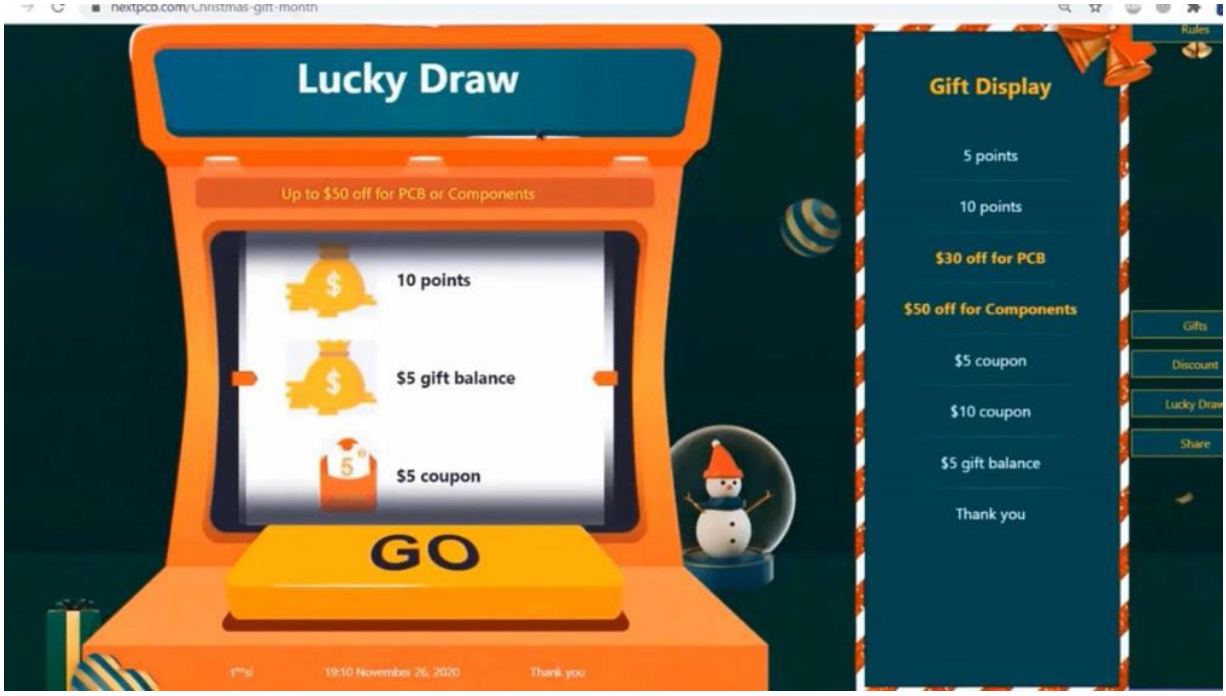
so that the current and voltage can be measured. And thus, we can know about the power consumption and total power consumed. The best current sensor available in the market is SDD013. This non-invasive SDD013 is the current sensor, plate core type, Clank meter sensor, which can be used to measure current up to 100 MRs. Similarly, the best voltage sensor is the AC voltage sensor module ZMP 01p.



The ATTEMPT 101p AC voltage sensor is the best where we need to measure the accurate AC voltage with a voltage transformer. Using the SCT 1 trick current sensor and z10pt101re voltage sensor, We can measure all the required parameters needed for the electricity energy meter. We will interface the current sensor and voltage sensor with ESP 32 Wi Fi model and send the data to blink application. The blink application dashboard will display the voltage, current, power, and total unit consumed in kilowatt hour. Isn't it a great way to monitor the bar consumption at your home?

So now without getting any delete, let's get started with this great project. This project is sponsored by next bispecific. Let's cheer for Thanksgiving Day and the coming Christmas day. As the holiday is all around us, the next Pacific prepares a massive gift for you. Cheers for the Christmas part.

If you place the order on the next recipe website, you can get the following 3 gift gifts: a dollar 10 coupon. gift b is dollar 20 cash plus ten points. Gipsey is 60 dollars plus 10 points. For a PCV order, they give 15% off for a PCV assembly order. 10% off.



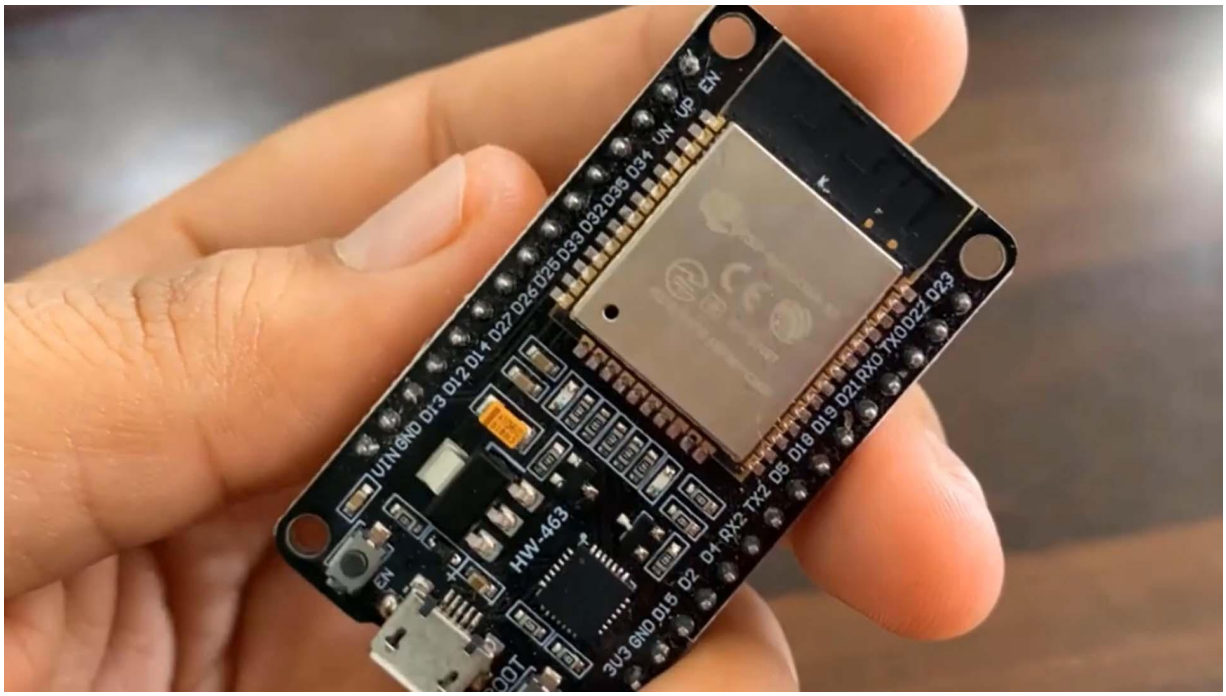
Everyone will get 3 chances to participate in the lucky draw. You can get up to $50 off for components and other gifts. You can check here how to get the 3 chances to participate in the La Quinta. Currently, there is a PCV design contest on Instagram. From next You can follow the Instagram page and follow the rules to participate and win a laptop gift. Let's just say that SCT 013 current sensor.

So, this is the sensor that I recently purchased. It is a non-invasive current sensor, split cord type, calameter sensor that can be used to measure up to 100 NPR. Current crossover sensors are for measuring the alternating current. I have the pair of sensors that you can see here. One can measure up to 30 amperes and the other can measure up to 100 amperes.

The sensor is an analog type and can be interfaced with the analog pin of any controller. The city 1 recurrence sensor can be clipped straight either to the life or neutral war without having to do any high voltage electrical war. Like any other transformer, This sensor has a primary winding, a magnetic core, and a secondary winding. You just need to place a coin carrying wire into this hole. The inductive sensor has a pin out of 3.5 millimeter audio jack.

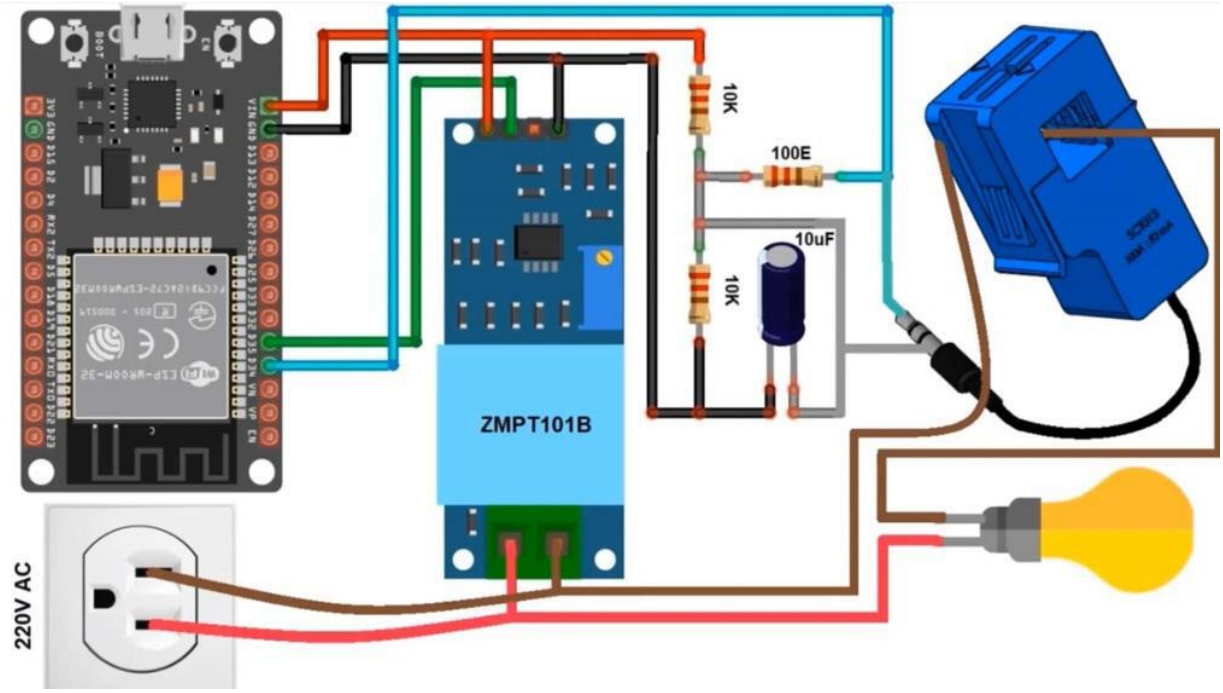
I think this audio jack may not be useful to me. So, cut the audio jack using the cutter. So, here you can see, it has 3 wires of different colors. One of the wires here is less so I will remove this wire. Alright.

We have 3 wires left which are fully needed. Now, let's just say the voltage sensor So this is a ZMP T101 B single phase AC voltage sensor. This module is based on a high precision voltage transformer used to measure the accurate ACV with a voltage transformer. The modules can measure the voltage within 250 volts, and the corresponding analog output can be adjusted. It has 2 headers for connecting the AC voltage lines and has 4 pins as BCC, out, ZND, and AND, where the output voltage is analog.



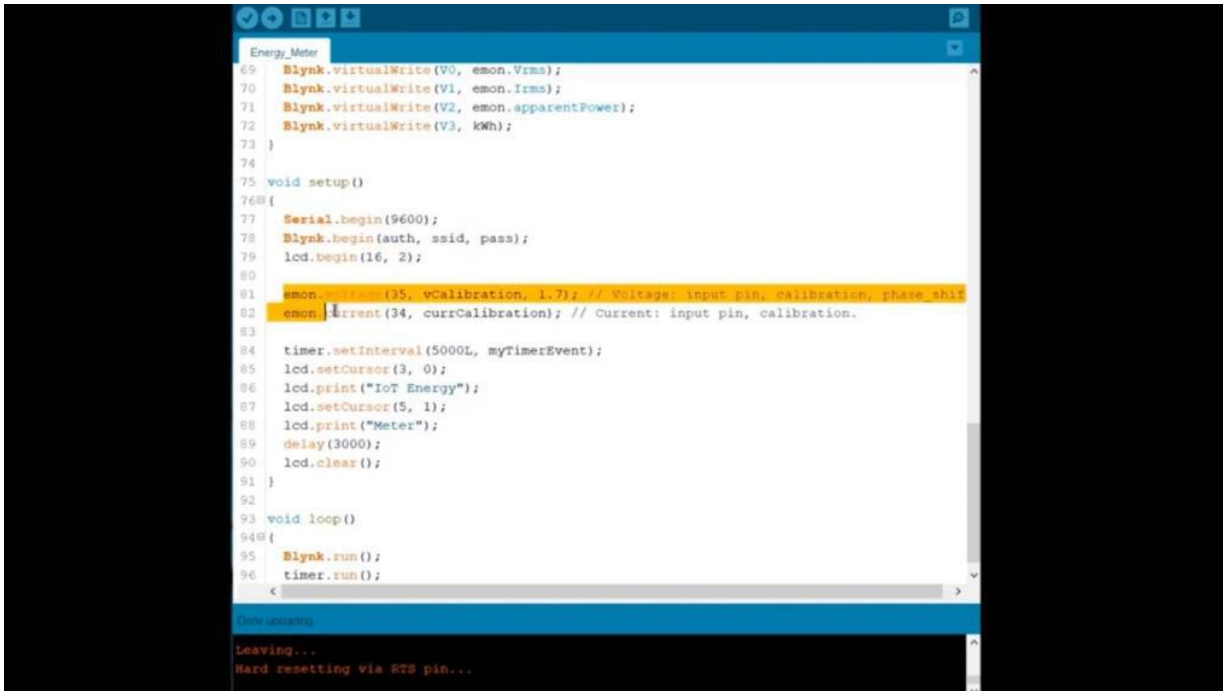
Apart from these 2 sensors, we need ESP 32 wifi module. The reason why I am not using ESP 82664, this project is because of the analog bin requirement. The SP 32 module has many analog bins, whereas ESP 8266 has a single analog bin. Both the chips have wifi features and can connect to a wifi network for IIT applications. Now, let's just see the circuit diagram and set up the hardware.

The circuit diagram has been designed using bridging software. The output bin of the voltage sensor is connected to GPIO 35 of ESP32. and the output pin of the current sensor is connected to ZPI O34 of ESP32. Apart from this, you can assemble the entire circuit as for the circuit diagram. So here is the circuit assembly.



I assembled the circuit on a breadboard. You can see here is the SP 32 WAFE module. This is a 16 by 2 LCD display which is optional. There are 3 resistors and a capacitor, and then we have a voltage sensor and then a current sensor. And as a load, I used a ball. I will measure the power consumption and all energy meter parameters using this bulb as a reference.

There is no need for an LCD. You can remove it from the project. You now need to install a blank application on your Android or iPhone. Download it from the App Store and set it for a new project. In this coding part, we use the LCD library and assign the LCD bin.



We also included the necessary libraries for the ESP 32 board. Yvonne Leaf handles the retrieval of data from the both sensors. blink SIMPolisP32 integrates the program to the blink mobile app. The energy monitor object is created and calibration factors are defined. The Blink timer object is then created to handle the sending of data to the Blink mobile app.

Then we defined the SSID and password on our local wifi network and inserted authentication code from Blink. The Millis and kilowatt values have to be initialized. The kilowatt hour starts at 0 and will slowly go up as time goes on. The values from the sensors are being retrieved and calculated. Using this function, the real power, appearing power, power factor, VRMS and IMS are being calculated.

We then use Blink virtual, right, to send the data to blink based on the virtual pin set. Under the set of functions, we initialized the serial portrait and set the current and voltage sensor analog pins as ZPI 034 and ZPI 035. Then we set the timer to 5000 l for an update time of 5 second In Site loop function, we are running the timer and blink. Now, it's time to upload the code. So connect the SP3 to your

computer And then from the tools menu, select the ASP 32 board and also select the COM port.

Then hit the upload button. Once the code is uploaded, you are ready to test the device. So open your serial monitor. You can see the SP 32 board is connected to WiFi and is sending the data to Blink Cloud. Initially, the sensor will show the wrong value.

Then it will start sewing the correct value. So here you can see the light is turned on. In the LCD display, you can see the power consumption and also the total kilowatt hour value. It is also displaying VRMS and IRMS values. The power and kilowatt hour value increases as the time goes on, and this kilowatt hour value depends upon the power.

Similarly on the serial monitor, the value of all four parameters can be absorbed. You can verify the current voltage value using the standard multimeter. Apart from all this, the data can be monitored in blink application. You can see here The voltage current and the power value displayed in golf or in America format. The energy

meter data is uploaded to the blink application after the interval of every 5 seconds.

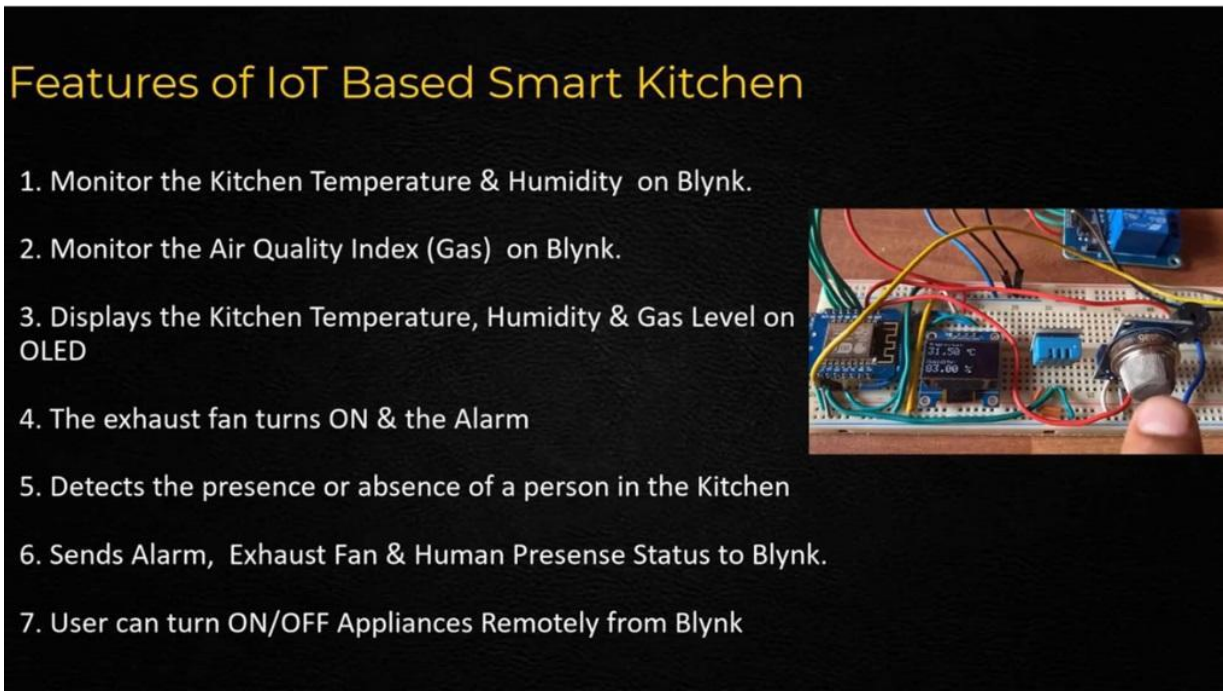
Thus, this is how you can make your own smart inter semmeter at your home to monitor your home electricity bill. That's all from today's project.

IOT BASED SMART KITCHEN AUTOMATION MONITORING

In this project, we will build an IoT based smart kitchen with automation and monitoring system using node MCU, ESP8266. The kitchen is one of the important places in a house. The safety factor is the main expectation that must be taken into account during the activity in the kitchen. The existence of gas leakage is uncontrolled fire, excessive temperatures and a moist environment must be quickly identified and addressed.

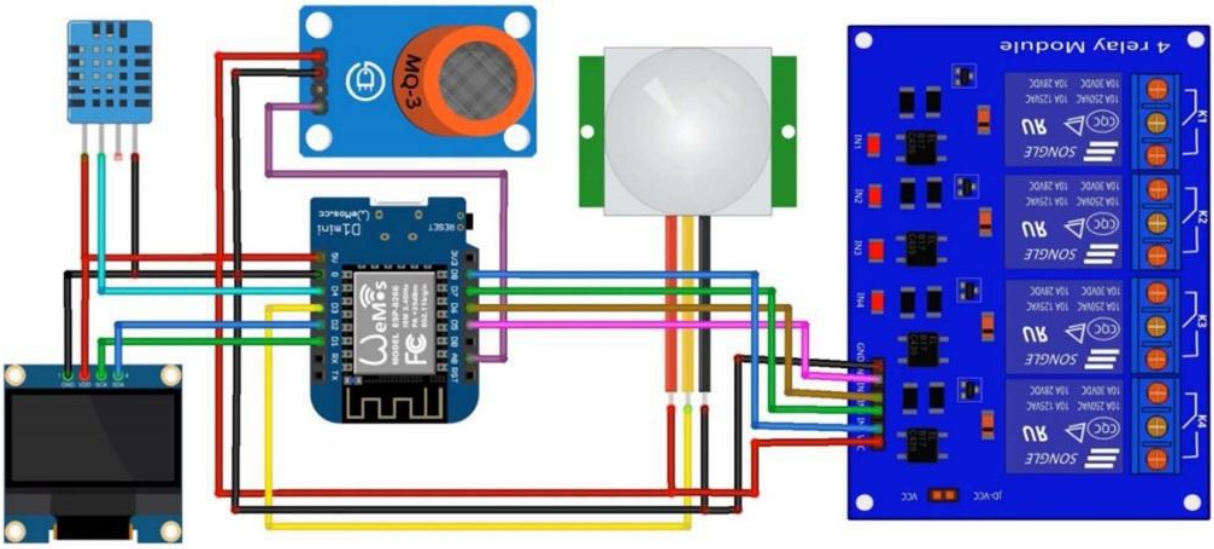
Apart from this, it's necessary to monitor and control kitchen appliances like lights, freeze, oven, etcetera remotely. The main motto of this project is to make a prototype of an IOT based smart kitchen using the internet of things. The system uses multiple sensors, relays, and node MCU ESP8266 boards. We can monitor all the sensor data on blink applications. We can also send commands from the blink app to control kitchen appliances.

Basically, the IoT smart kitchen does the following task. Monitor the kitchen temperature and humidity using the DST11 sensor on the blink app. monitor the air quality index gas using MQ135 gas sensor on blink app. Disperse the kitchen temperature, humidity, and gas level on a 0.96 inch OLED display. The exhaust fan turns on and the alarm activates once gas level exists.



Detects the presence or absence of a person in the kitchen using a PIR sensor. Since alarm status, exhaust fan status and person in room status to blink app. Users can turn on or fridge oven room lights remotely from the Blink app. So let's see how we can build this entire system. And without getting any delayed, let's get started, Now, let's see the required components for this project.

We will need the DSC 11 sensor to measure room humidity and temperature. This is the MQ 135 gas sensor which measures the gas level leakage in the kitchen. Then we have a passive infrared sensor that takes the presence or absence of humans in the kitchen. This is a simple five volt buzzer for alarm when the gas level exceeds. Then we have a 0.96 inch OLED display for displaying room parameters.



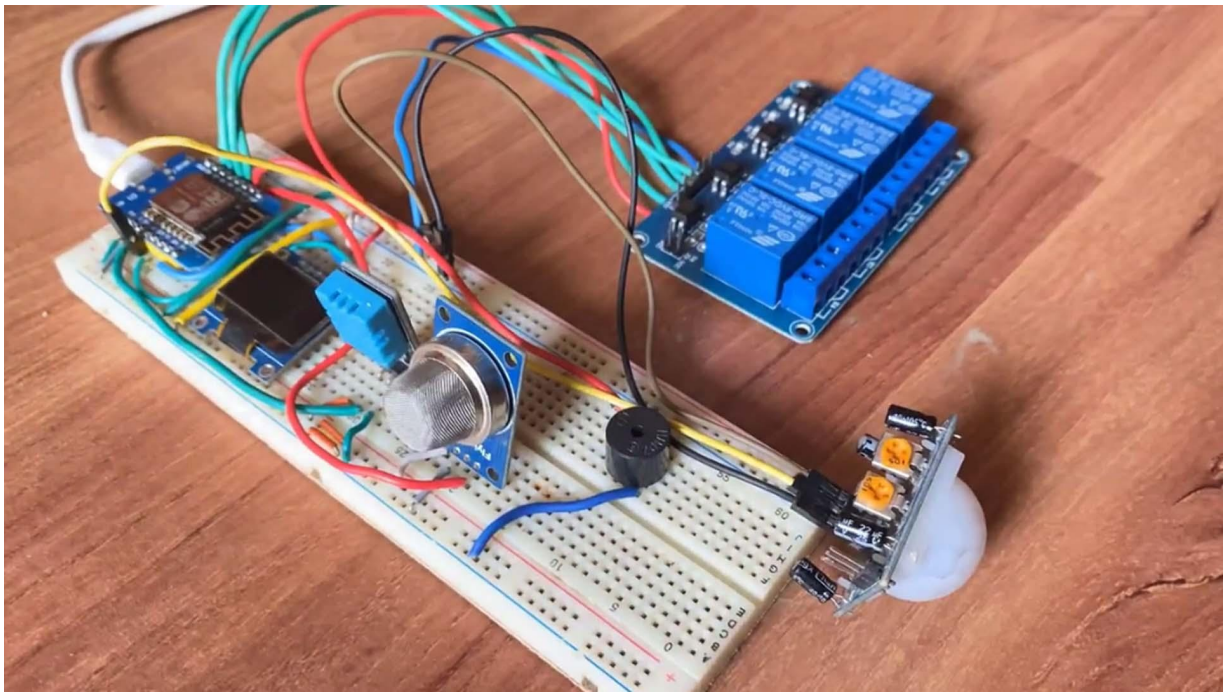
Circuit Diagram & Connections

A four channel relay module out of which one is connected to an exhaust fan and the other 3 to room appliances. And the most important thing is the ESP 8 to double 6 based chip called Wemos D1 Mini. You can also use the note MCU board for this project. This is the schematic of the project designed using phishing software. We are operating its sensor and module using a 5 volt supply from Wemos 5 volt pin.

All the modules are connected to a digital pin of Wemos d 1, except the MQ 135 gas sensor which is connected to the analog bin of WIMOS. You can assemble the circuit on the break port. In case, if you need a custom PCB, then you can download my PCB files. Here is the schematic that I designed using the easy EDA tool. Then I converted the schematic to a PCD.

The PCV is very simple and you can design it in any shape with compact dimensions. Here is a three d view of the PCB. The PCB looks very beautiful and compact. All the components fit perfectly on the board. You can download the Gerber file for this PCD from the description link.

So it's time to order the PCD. You can visit the next PCB for PCB services. Next PCB is also my favorite PCB manufacturer as they provide trial PCB 2 layer PCB and full layer PCB with free PCB SML Shipping services up to a fast lead time of 24 hours. To place an instant cord, click here. Then, upload the Gerber file, then select the PCV material, layers, board type, and also the PCV quantity.



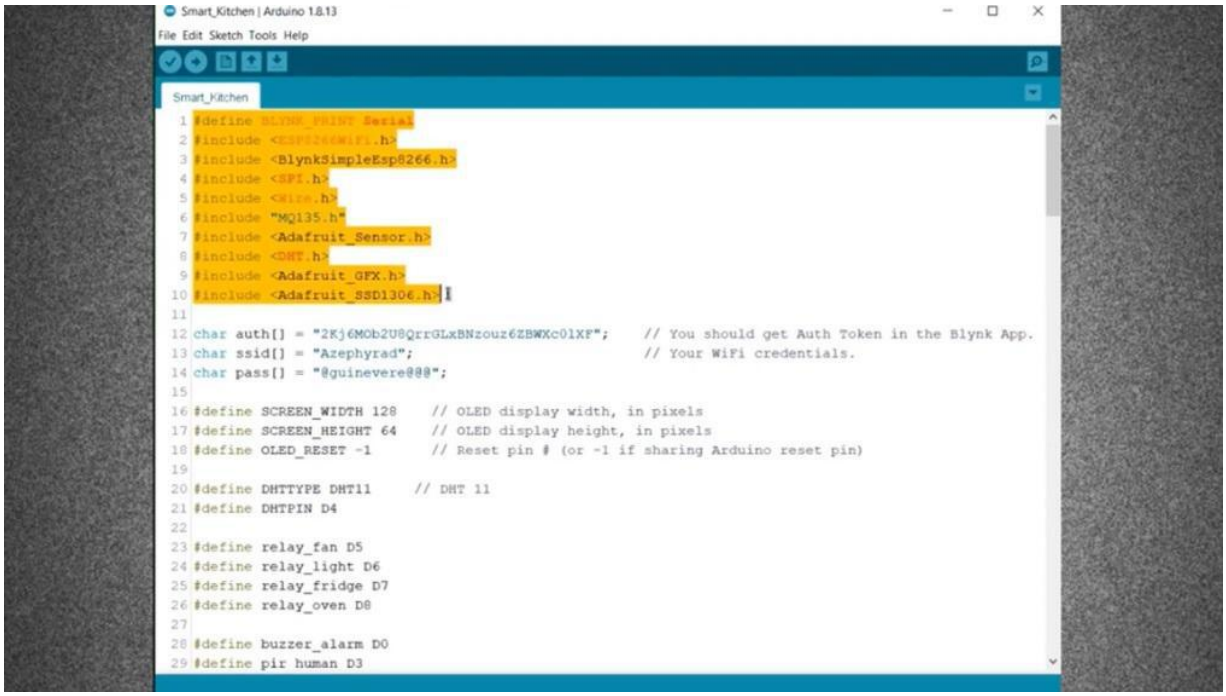
Select the PCV thickness, color, surface finish, and all the parameters. Then you can select your country for shipment and then place an order. After verification, payment, and manufacturing, you can get your PCB in the next couple of days. I preferred assembling the circuit on a breadboard for testing purposes. So this is my circuit assembly. I used jumper wires for a connection.

This is a PIR sensor, then a buzzer. MQ 135 gas sensor, DST11 sensor, OLED display, and a 4 channel relay. The connection is exactly the same as per the circuit diagram. Now, we need to set up the blink application so that we can receive the data from ESB 8 to double 6. So download and install the Blink application from Google Play Store.

Once the installation is completed, open the app and sign up using your email ID and password. From the dashboard, create a new project and select note MCU board and Wi Fi connection, then drag and drop or add 3 buttons to style buttons, and 3 calls from the widget list. Assign the 3 variables and name them like temperature, humidity, and air quality index value as per code. Similarly, assign the 3 variables from the button and 2 variables for the style button. You will get the authentication code in the mail.

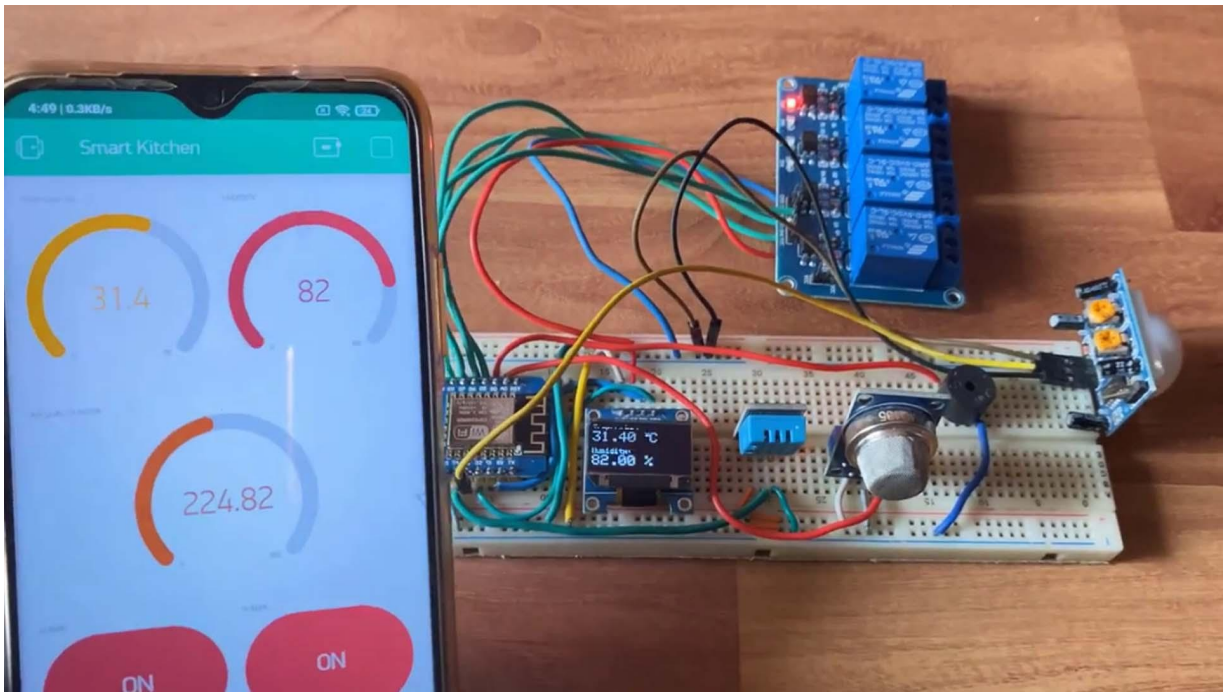
Copy this authentication code. This will be used in your code. So here is our code. In this code part, we need so many libraries like DSC11 and MQ135 sensor library. We will need a blank library as well.

And a couple of OLED libraries. Download all these libraries from the website articles. From here, change the blink authentication token that you copied earlier, then change the wifi access ID and password. Then from the tools menu, select the WEMOS T1 board from the board manager. Also, select the COM port.



Finally, upload the code. After uploading the code, you can check the serial monitor. The serial monitor will display temperature, humidity, and air quality index value. It will also show the alarm status as well as friend status and also human presence. On the other hand, The OLED display will display the temperature, humidity, and gas level of the kitchen.

Once it connects to the Wi Fi network, It will start sending the data to the Blink application. The Blink application will receive the temperature, humidity, and air quality index data, and display them on the gauze. It will also show whether the alarm is on or off, as well as the presence of humans inside the room or not. The relay one that connects to the existing fan automatically activates when the gas level reaches the threshold value. You can check this by introducing gas, smoke or any perfume near the sensor.



Meanwhile, I'm using body spray for testing. So once the vascular exceeds the alarm turns on and the relay activates, This relay is connected to an exhaust fan for cleaning the air. You can send the command from the blink application to turn on the kitchen appliances like freeze, oven, and lights. Connecting any of the home appliances here using the following schematic is very simple and useful. So this

is how you can design your own IOT based smart kitchen using node MCU ESP 8 to double 6 with automation and monitoring system on blink application.

IOT BASED SOIL NUTRIENT MONITORING ANALYSIS SYSTEM

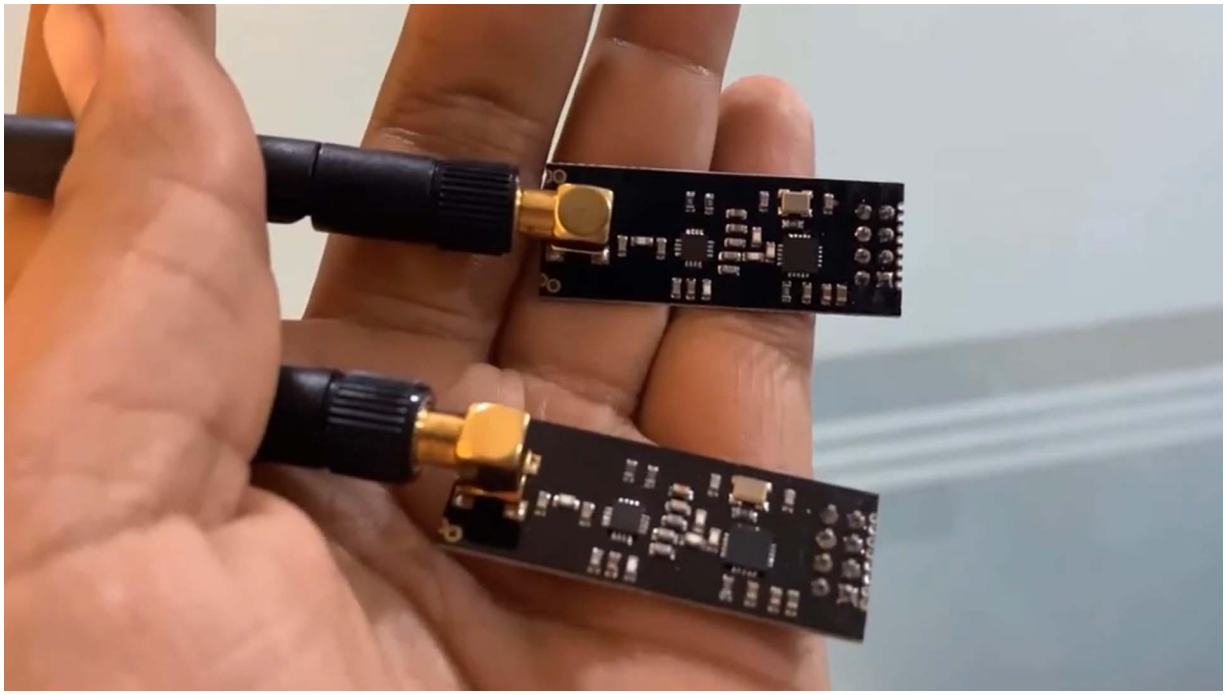
In this project, we'll be learning about IT based soil nutrient monitoring and analyzing the system using Arduino and ESB 32. Soil is the base of agriculture. So it provides nutrients that increase the growth of a crop.

Some chemical and physical properties of soil such as its moisture, temperature, soil, nitrogen, phosphorus, and potassium content heavily affect the yield of a crop. These properties can be sensed by the open source hardware and they can be used in the field. In this project, A soil nutrient monitoring and analysis system is proposed in which the farmer will be able to monitor soil moisture, solar temperature, and soil nutrient content like nitrogen, phosphorus, and potassium. The farmer can monitor all these parameters wirelessly on a mobile phone or the crystal system. To measure the soil moisture, we will use a capacitive soil moisture sensor.

The temperature of the soil can be measured by using a ds18b20 waterproof temperature sensor. Similarly, in order to measure the soil and PK values, we will use soil and pick a sensor. All these sensors can be easily interfaced with the Ardeno. We will use the ThinkSpace server to monitor the data in graphical and numerical format. We will use an RF2401 wireless transceiver module to send the data from sensor node to gateway.

The data from the transmitter can be transmitted wirelessly from a kilometer distance to the receiver. The receiver is built using the

ESP32 Wi Fi module, which has access to the Wi Fi network. using this wifi network, the data can be uploaded to Think's Big Server. So let's build an IT based soil nutrient content analysis monitoring and testing system simply using wireless sensors network, Ardeno and ASV32. This project is sponsored by Next Pacific.



Currently, Next Pacific has developed a PCV design analysis software called NextDFM. which is a design problem detector or an engineering solution provider. To use this software, you need to sign in using the next PC account. you need to import a Java file with just one click. The project graphics are easy to read, so you can make sure that the file contains all the necessary data.

It helps you quickly familiarize DFM Design specification and production needs to determine whether there are any manufacturing constraints. It also effectively checks whether there are hidden dangers in the design file, selects available parts, and evaluates the cost in real time. So try to start using the stool for a recipe analysis. The link is given in the description. Welcome back.

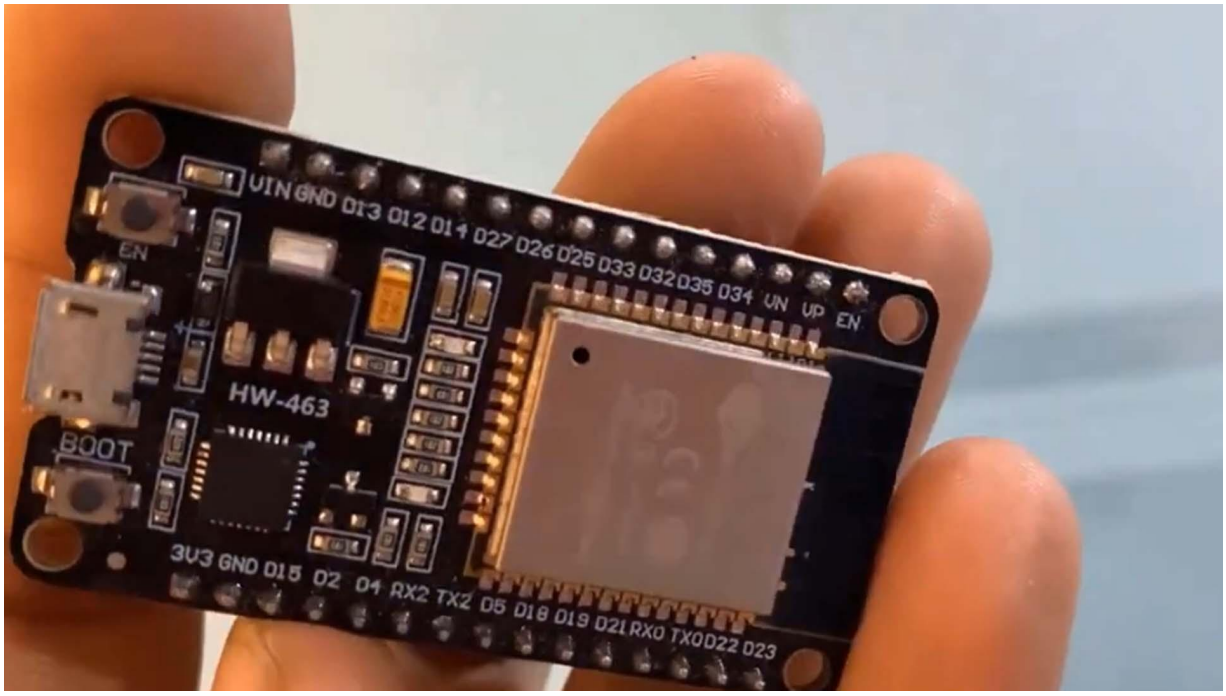
Now let's just see the components required for this project. The first sensor that we need is the soil and piggy sensor. The sensor measures the content of nitrogen plus ferrous and potassium in the soil and helps in determining the fertility of the soil. For this, it uses a modbus protocol for communication. Then we needed a temperature sensor, so I chose a DS18b20 waterproof temperature sensor.

This sensor can be inserted into the soil easily, and then the soil temperature can be measured. The sensor is based on a one wire protocol. We will also need a soil moisture sensor to measure the soil moisture content. For that, I selected the capacitive soil moisture sensor. The soil moisture sensor can be easily inserted into the soil, and soil moisture content can easily be determined.

The sensor is an analog type and gives analog value output. Apart from all these sensors, we need a wireless communication module So, I chose a pair of NRF24l01 wireless receiver modules. Each module can both sync and receive it a wireless It works within the frequency of 2.4 gigahertz. The modules, when operated efficiently, can cover a distance of eleven hundred meters, and I choose ESP 3rd to do wifi models for making the gateway or the receiver circuit. Then I selected the RD unit enabled for making the note or transmitter circuit.

This is the RS 45 mode module used with URDU and Sewer and picking a sensor to read mode data. And finally, is the 4.7 k resistor used to beat the DS 1 8b20 as a pool upper resistor. Now let's see the schematic or the circuit. So this is the circuit for the transmitter or sensor node. As we can see, all the sensors are connected to the Arduino Nano Board.

The Aneta 24l01 transceiver model is connected through SPI pins here. All the components are powered through Juno 360 ground pins except the soil and pick a sensor, which requires it to avoid power supply. So, here is the complete assembly of the sensor node on a break board. This is the cell and picks a sensor antenna temperature sensor and a soil moisture sensor. This is the Anurag 24l01 module with upgrade 2.4 uharzantina.



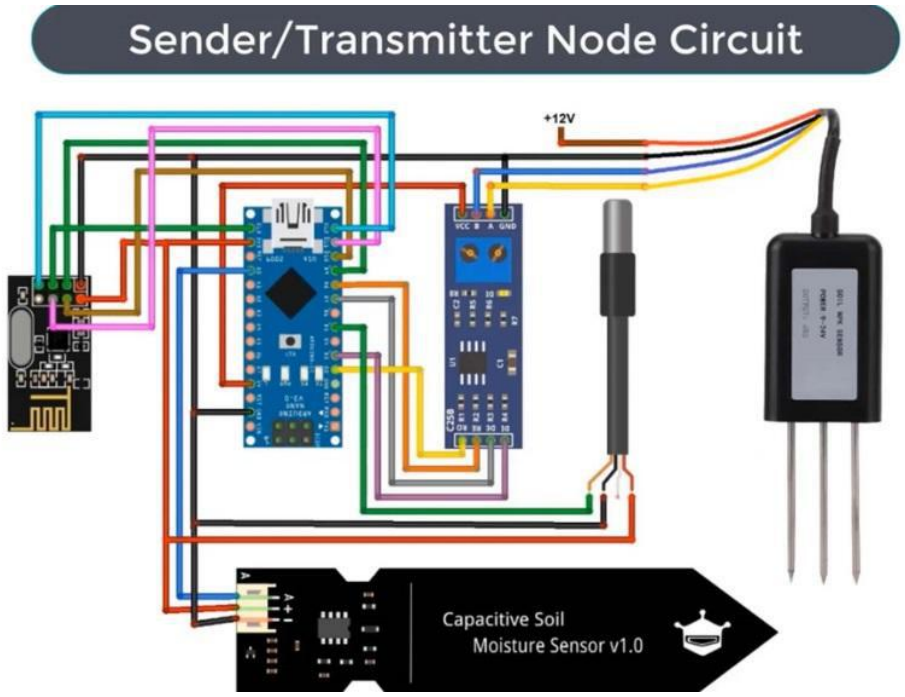
This one is the MAX 485 Modbus module, and then our unit nano board and a 4.7 k are connected to DS 18b20 and Mississippi. And this is the receiver or gateway circuit The receiver is made using ESP 32 Wi Fi module and NRF24l01 module with the SPIP interface. So you can see the symbol circuit here on the break port. The same 2.4 gigahertz wireless receiver module is used for receiving data. We will now state that the ThinkSpace server will receive the data on the cloud from our hardware.

Think Big Server is an IIT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. All you need is to visit big dotcoms and create an account. After creating an account, click on the channel and create a new channel with 5 different variables like nitrogen, phosphorus, potassium, soil temperature, and soil moisture. After that, save the channel. Now quickly go to the API keys and copy the API key, which will be used in the code.

Now let's just move to the programming part. In the transmitter program, there are many libraries used which you will get from the description link. In the receiver code, replace this API key with your

API key. Also change the WiFi SSID and password. From this line, you need to put some factor for calibrating the soil moisture sensor.

Using the function, we will send so many bytes of data to the receiver using the 24l01 module. The rest of the code is the same. You just need to read the sensor data from sensors and send it to the receiver circuit. The receiver will upload the data to the Think Speak Cloud Suburbs. Upload the code of the transmitter to the Arduro network and also upload the code of the receiver to the ESP 32 board.



After uploading the code, your device is ready for testing, evaluation, and observations. Open the serial monitor for both devices. The sender will send the data to the receiver, and this receiver will initially connect to the wifi network. All the happenings can be observed in the serial monitor. The gateway collects the data and uploads it to the sync specs server.

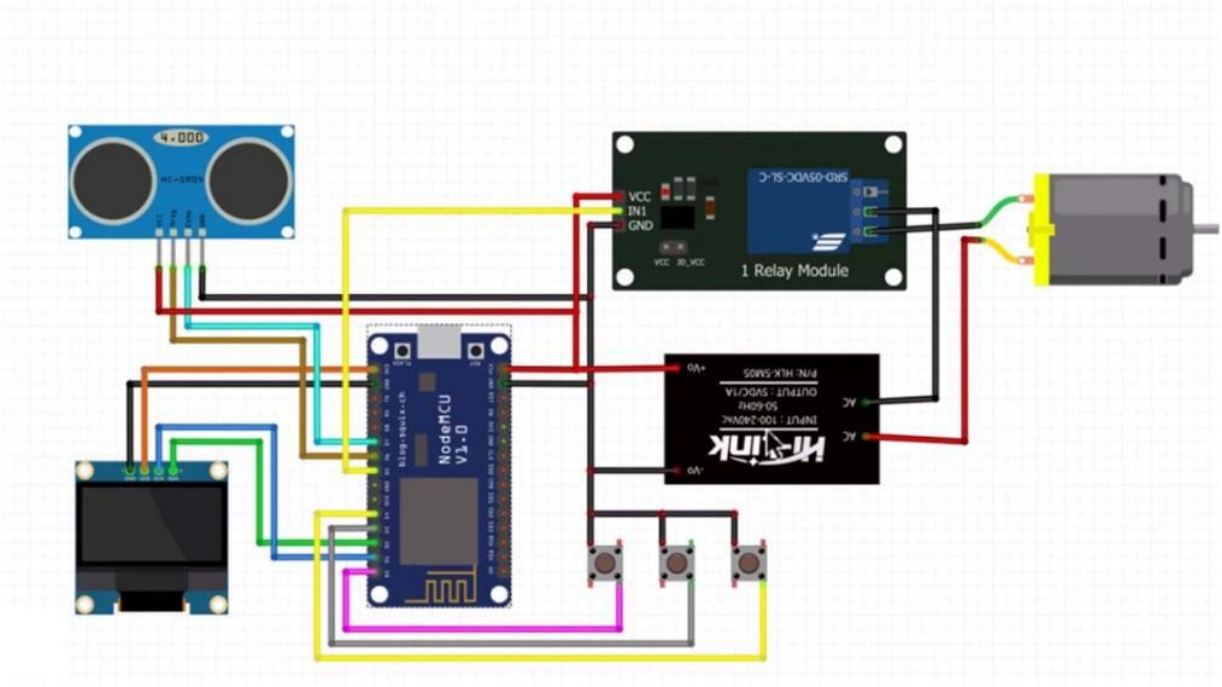
You can just visit the private view of the sync's big dashboard and you will see the data being locked into the sync specs server. The data is uploaded after an interval of 15 seconds regularly. This is the

graph of nitrogen phosphorus, potassium, soil moisture, and soil temperature.

IOT BASED WATER LEVEL CONTROL MONITORING SYSTEM WITH ESP8266 BLYNK

In this project, we will build an IoT best water level control and monitoring system using the ESP 82 double 6 Wi Fi module. We will use a waterproof ultrasonic sensor to measure the water level. The water label can be absorbed on an OLED display and the blink platform.

This water level controller project can be operated in both automatic and manual modes. In manual mode, you can switch the water pump on and off it anytime. In automatic mode, you can check the logic for when the pumps are turned on or off. You can also manually turn the relay on or off. For the demonstration, I am using one bucket as the water source and another bucket is the tank.



I have placed the waterproof ultrasonic sensor at the top of the pocket. So this project works as expected. Now let's get started and see how we can build this system. The components required for this project are ESP80266 Wi Fi module. 0.96 inches I Square C OLED display.

High link AC to 5 volt DC converter module. single channel relay module. A pair of push button switches. JSN SR040 Waterproof ultrasonic sensor. SE water pump, you may use any other pump as well.

Water pipe for water transportation from reservoir to 10. 0 PCB board. Now let's have a look at this schematic. I used Greetings Update to draw this schematic. In this schematic, we are powering the entire device using the high link SE to 5 volt DC converter module.

The relay is used here to control the turning on and off of the motor. The motor is directly powered via 220 Volt AC. The OLED display is connected to the I square c pins of the ESP. The ultrasonic sensors eco and trip pins are connected with the digital pins of the ESP 82

double 6. There are 3 post buttons for modulation, buzzer, and relay control.

In our project, we are only using 2 buttons and ignoring the project functionality. I used a Jira PCV to assemble the circuit. All components fit perfectly on the board. After assembling all the components, I connected the AC lines to the high link info terminal and to the relay. Finally, I plug the ESP80 into a double 6 board into the holder.

Then I connected the ultrasonic sensor to the Fourfin terminal. The assembly looks awesome. You may create your own PC to give it a professional look. The hardware setup is done. Now, let's set up the blink app.

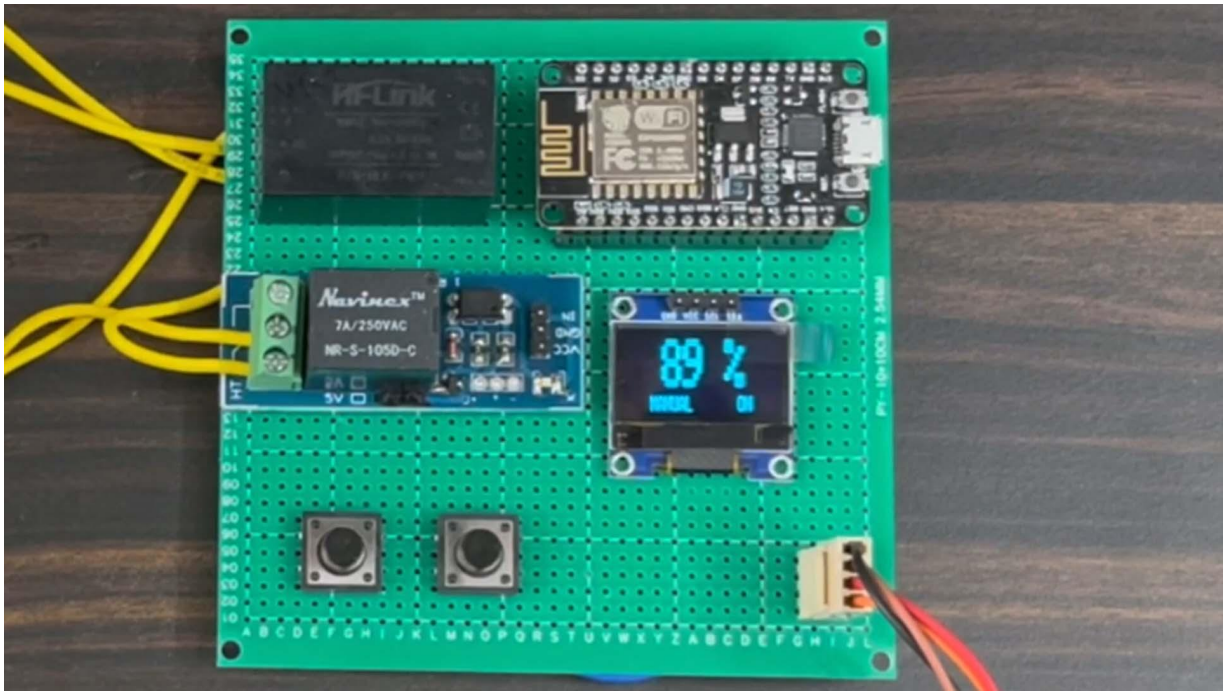
Log in to your blink wave dashboard. Here, we need a gauge for indicating the water level and a pair of switches for relay control and mode selection. 1st, set the goals by shining the virtual pin v 1 and auto settings as per the project. Similarly, set up the virtual digital pins to configure the button functions. After setting the web dashboard, you can set up the mobile dashboard in the same way.

Moving on to the coding part, the code contains multiple libraries. First, you need to add libraries to the library folder. Here are all the pin assignments as per the circuit diagram. From these lines, change the blink authentication token and other details. Similarly, you need to make changes to the wifi, SSID, and password.

That is all the modification needed. Now you can select the note MCU board from the port list. Select the COM port and upload the code. After upgrading the code, you may start testing the project. The ESP 82 double 6 will connect to the wifi network and start sending data to Plink.

For the demo, I have used two buckets. One bucket is the source of water and the other is the tank. I placed the ultrasonic sensor at the

top of the tank. Here, you can see the OLED displaying the water level in percentage and showing the status of manual mode. In this case, the motor will start and pull the water to fill the jack.



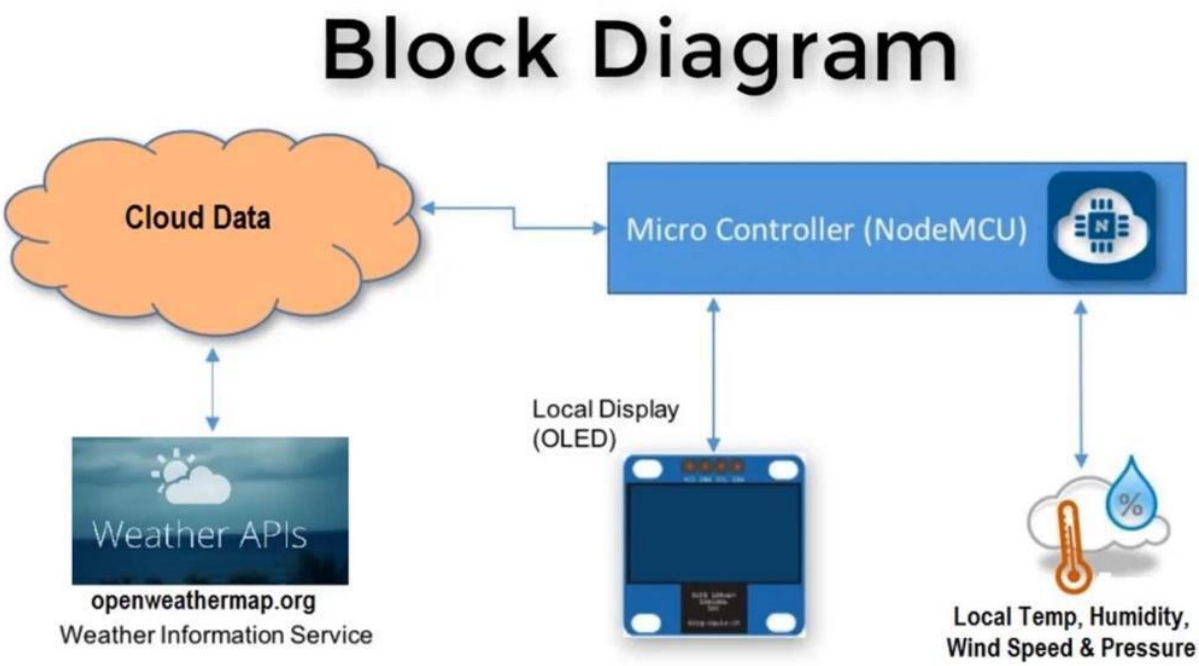
At the same time, the ultrasonic sensor is measuring the water label. As a result, there is an increase in the water level here. The water level keeps rising and reaches 100%. The water will stop Now let's see the working in auto mode. In this mode, you can assign winter start and winter stuff.

There is no need to manually start. The same data is entered in the blinkup. The blank dashboard will show the bottom status along with the water level. In auto mode, the water is rising continually. Now when the water label reaches 96 percent, the motor turns off automatically.

IOT BASED WEATHER STATION NODEMCU WITH OLED OPENWEATHERMAP

We'll learn how to design an online IoT based, whether it's testing using low MCU and OLED display. In the previous project, we learned how to upload the data to the internet. That is to speak by uploading the data from the MCU. But in this project, we'll learn how to download the data from the Internet and display it on an OLED display. So the process is just the reverse.

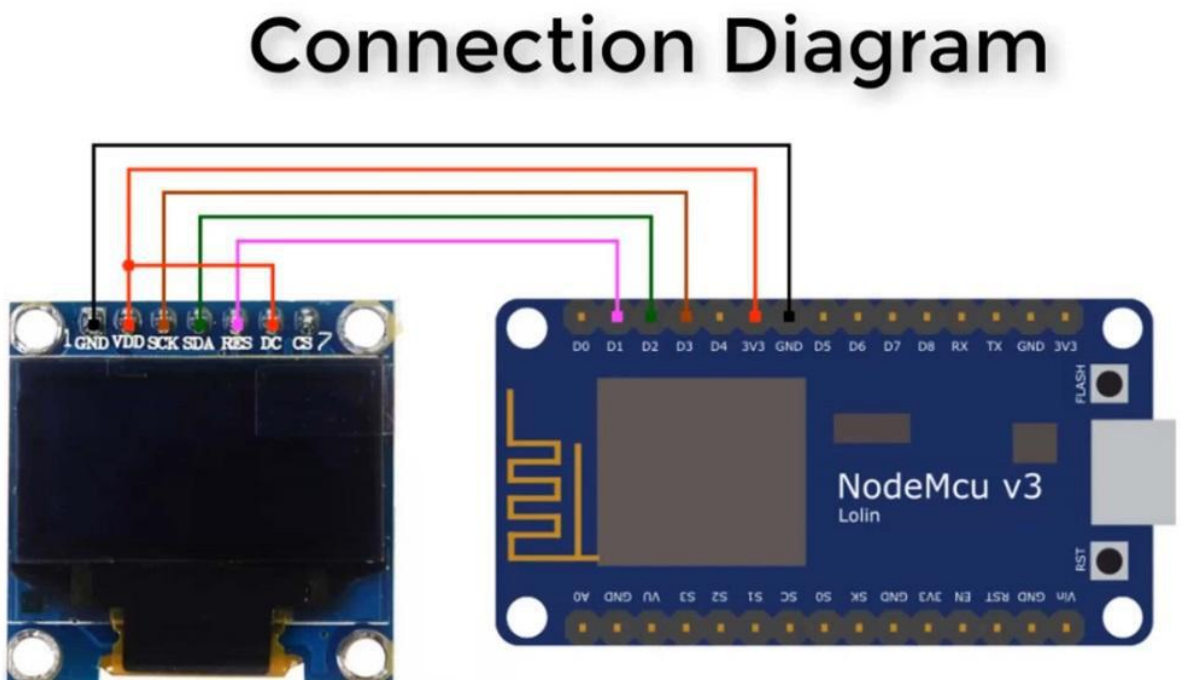
So here, we'll be displaying temperatures, humidity, pressure, wind, speed, and wind degree. And there is no need for any sensors. The downloader will be automatically So this is the block diagram. So open weather dotorg the online service, which uploads the data on the internet of weather of the entire world. So this data is on the cloud, and the node MCU receives the data from the cloud and the local temperature, humidity and pressure.



These are displayed on 128 into 64 OLED displays. So the data from the cloud is described by node MCU and displayed on an OLED display. So the combination diagram for it is, simply, we have collected the pin number D2 and D32. That's the end, SDK teams, and BCC is 3.3 power supply. ground is directly connected to ground. You can use ITC LED.

Now how to do this. So let's learn about programming. So we need 3 different libraries. That is ardenojessan.hdFX library and OLED library. So just go to the ECS and open, manage the library.

Now here, just search the 3 different libraries. that I explained. Now, first, let's search the library for Ardeno Jason. So, here you can see you can get the library from here. So simply select the button I have already downloaded from the library.



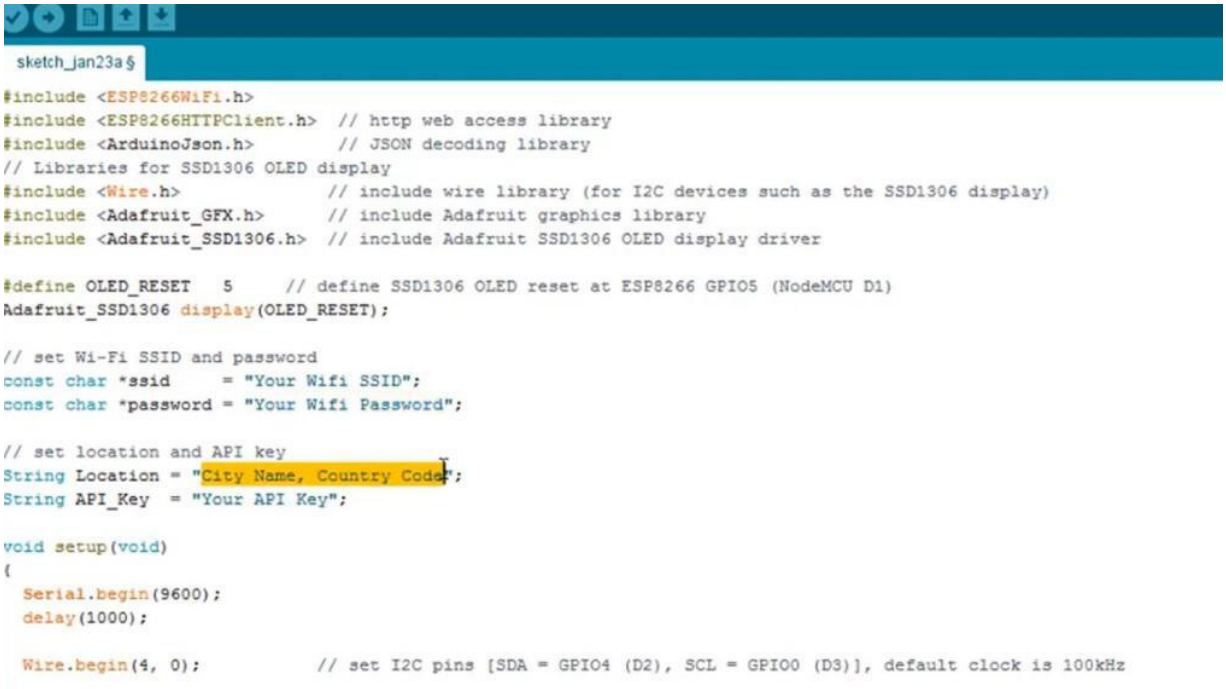
Now let's get to the library. This is the SSD 1306. So are the libraries installed? Similarly, install the GFX library. So the DFS Library installed as well.

Now here, you need to modify the program by entering the to Wi Fi SSID. So simply enter the Wi Fi SSID from here and also the password from here. Now here you need to enter the city name and your API. So for uploading or downloading the data, you need an online service So this is provided by openweathermap.org. So visit this link and just create an account here.

I have already created So I'll just sign in. So after you sign in, you can see at the bottom, there is a central API key. So click here and you'll get a default key by something like this. And if you want to create a new key by yourself, enter the ear and just generate. and open your arduino code and just paste it over here.

Now, we need to set the city name and the country code. So from where do we get the country code? to open here and search for your city. For example, I was staying in India. If you are staying in any part

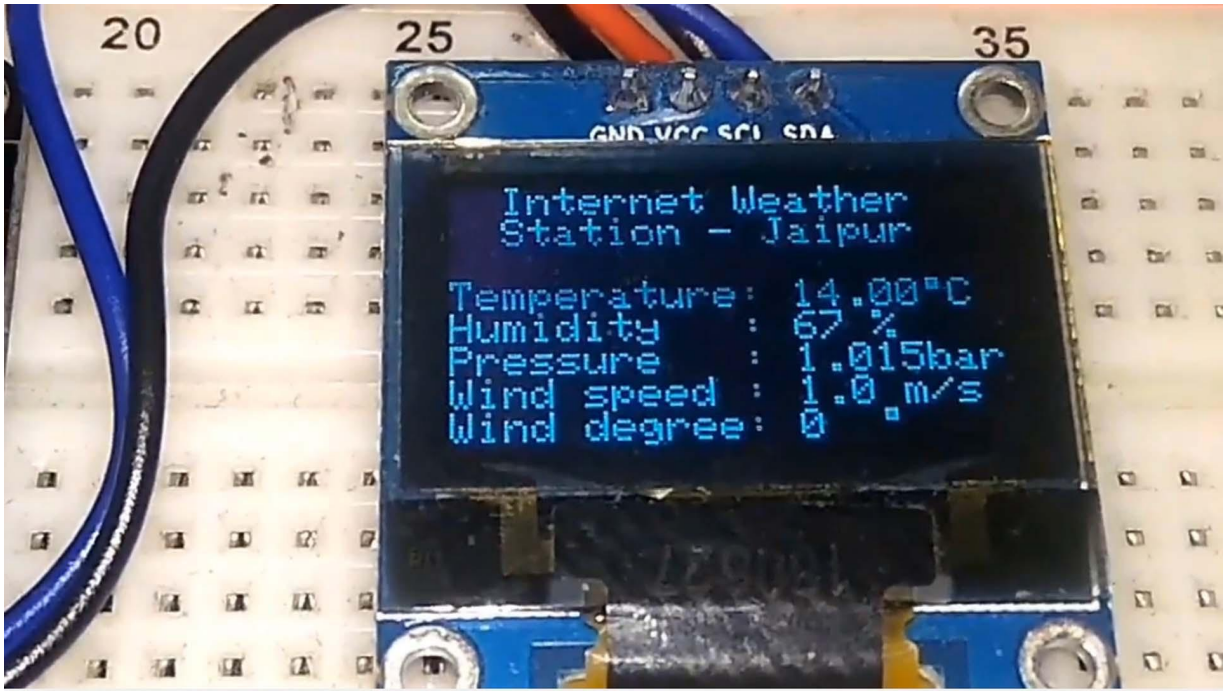
of the world like London, Washington, whatever, enter your city name.



So here, I'm seeing the city name Jaipur is given and suddenly the country code. That is India. I am here. So copy this and simply paste it over here. Now, go to the tools and select your node MCU board.

I'm selecting no MCU ESP tool report. Select the port and compile the code after you compile, upload it. So once you upload it, I'm currently uploading. So you can see the MCU and Well, it is connected over here. So you can see the network error is special.

Location is enabled. Now it is connecting. Now it's connected. after the server disconnected. So you can see the current temperature, the humidity pressure, windy speed, and wind degree.



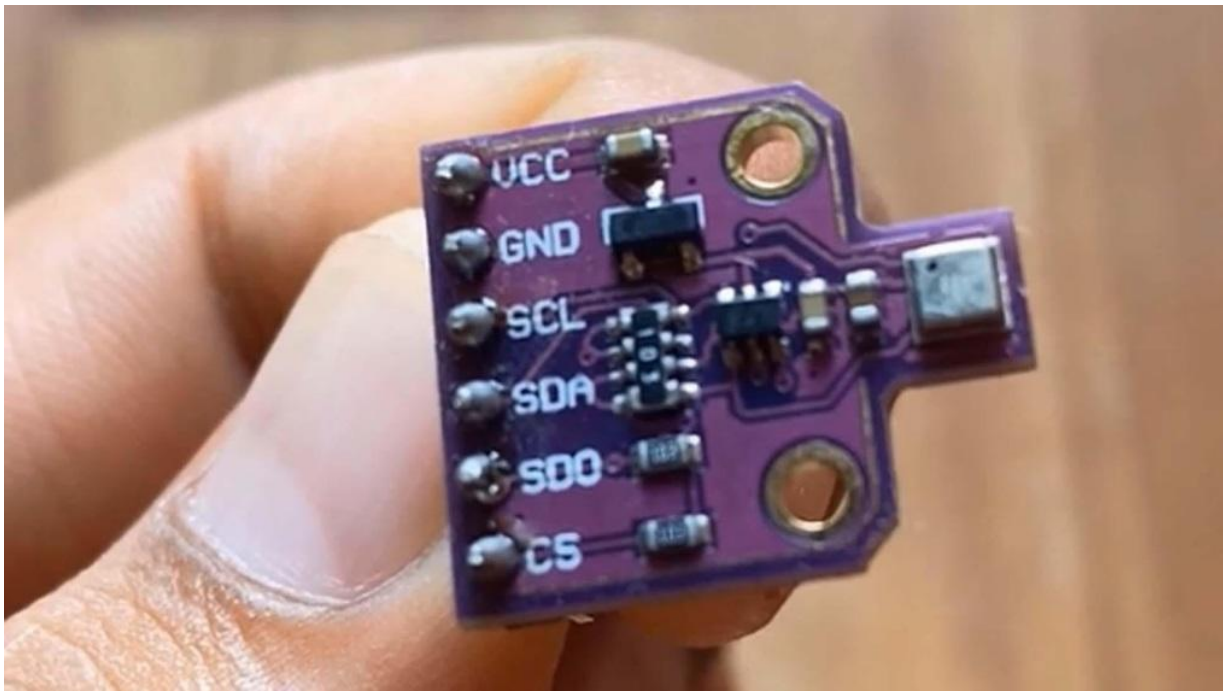
Since wind speed is nothing. That is one meter per second, wind degrees also nothing. So there is no zero degree at all. So this is how you can design a simple weather station. using Node MCU and OLED by retrieving the data from the server.

IOT INDOOR AIR QUALITY MONITORING (IAQ_ CO2_ VOC)

I explained about the BMEC 80 integrated environmental sensor. We went to 1st BM E 680 with Audio, as well as ESP 8 to double 6 or ESP 32 boards. We designed our simple weather station project using Arduino and further made an IOT based weather station using ESP 8 to double 6.

We monitor the weather data on the MQTT platform called UBS. But the drawback of the project was we could only measure mental parameters like temperature, humidity, pressure, altitude, dew point, and gas resistance. We were unable to calculate the air quality index value. We could not even get the equivalent carbon dioxide reading and also the percentages of volatile organic compounds. So in this project, We will use a highly advanced BME 6 area library called the BSEC library, which stands for BOSC's analytic environment cluster.

Using this library, we can get the value of IAQ, that is the index of air quality, and also the equivalent bon dioxide or total volatile organic compound. We will use the Blink Cloud platform to monitor the environmental data remotely. We will use the Wi Fi connection to upload the data regularly to the Blink server. In this way, we can monitor indoor air quality or outdoor air quality. So without getting any delay, let's see how we can build this entire system.



This project is sponsored by my favorite PCV manufacturer company called Next PCV. Next PCV is one of the top manufacturing companies in China. They offer PCB board and PCB assembly services at the lowest affordable price. You can get trial PCB, 2 layer PCB, and 4 layer PCB with free PCB assembling services up to a fast lead time of 24 hours. There is a special offer for you from the next pcp.

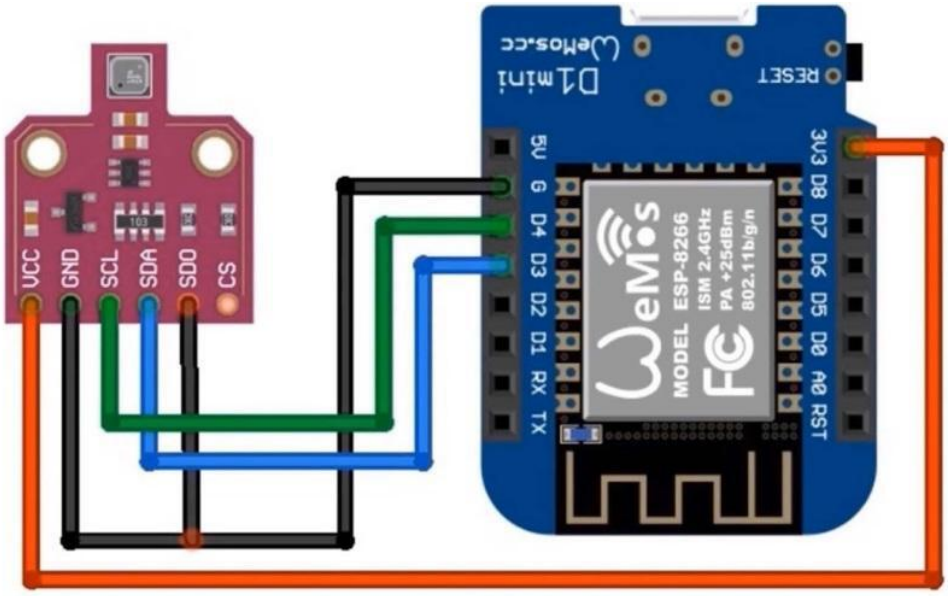
Share your adviser comments on next pcp products on your social media to participate in the La Quinta. There is a 100% winning rate to receive gifts, $300 cash balance, $100 Amazon gift cards, and coupons are awaiting for you to draw. So register now and place an order. Welcome back again. The BME 680 is a digital 4 in one sensor with gas, humidity, pressure, and temperature measurement based on proven sensing principles.

The gas sensor on the BME 680 can detect a wide variety of volatile organic compounds to monitor indoor air quality. The temperature measurement range of the BME 680 sensor is -40 to plus 85 degree centigrade. And the humidity measurement range is 0 to 100%. It

can measure the air quality index from 0 to 500 PPM. It can also measure equivalent carbon dioxide or total volatile organic compound and breath VOC equivalent The sensor operates between 1.7volt to 3.6 volt and communicates on I Square C or SPI protocol.

The main component used in this project is Wemos d 1 mini board. You can also use the node MCU board. All these boards have ESP826612e chip, which is a 32 bit, highly advanced and fast controller. The chip has a built-in wifi chip that has the ability to upload the data to the internet or server using a wifi network. Here is a connection diagram between Wemos G1 and BME 680 sensor.

Connect the BME 680 SCANS DAP to D4 and D3 of the WEMOS port and connect the STO to GND. Supply the sensor is 3.3 Volt VCC through the WEMOS port. You can try this connection on a breadboard or simply use your own custom design PCB board. I prefer a breadboard connection for testing the circuit. Now since all the connections are done, you are ready to do the testing and programming part.



Circuit & Connection

BME 680 users are complicated in an advanced library called Bsec Library. Here is a GitHub repository for this library. BSCC means BOSK Sensortech Environment cluster. The library has been conceptualized to provide a higher level signal processing and Fusion for the BME 680. The library receives compensated sensor values from the sensor API.

It processes the BME 680 signals to provide the requested sensor outputs. You can simply check this link to learn more about this library. The library is supported by a 3216 and 8 bit controller. It does not support most of the audio boards, but supports ARM controllers ESP 8 to double 6, ESP32, MSP 430, and Raspberry Pi. Before using this library, you need to modify some system files as part of the trucks here.

Open your Arduino IDE, go to Sketch and then manage libraries. Search for BSEC. So you can see the library here as well. So you can install the latest version of the library by clicking here. Now go to the files and then select examples.

From this example list, select the BSCC basic example code So here is the complete code for BME 680. A small modification is needed in the code. So look for the wire to begin to function and put 0 commas inside the braces. That's it. Come down.

So here you can see how the sensor data is printed on a serial monitor using the inbuilt library function. This library won't compile unless you make modifications to some file system. So go to app data, then click on local, original 15, packages, ESP 8 to double 6 hardware. Again, ESP 8 to double 6, Verizon. And then find a file called platform dot TXT.

Open this file with notepad plus plus. Go to line number 96 and add some line here as I'm doing here. You can find this line in the website article. So copy it from there. Similarly, go to line number 122.

You need to modify some lines here as well. Slide the cursor and add some text between the quotes before WL. That's it. You can save the file now. On the hardware side, connect the micro USB data cable to the Wemos D1 mini board.

and also connect the other end of the cable to the computer USB port. Hence, now the device is ready for programming. So go to the tools and then select the Wemos D 1 board from the list. You can also select node MCU boards from the list. Then select the COM port.

Hit the upload button and upload the code. Open the serial monitor now. So here you can see time stamps, temperature, humidity, pressure, IAQ, CO2 equivalent, VOCEQ valid data displayed serially and separated by commas. Initially, it will show the IAQ values as 25 only. The sensor takes almost 15 to 20 minutes to give a stable reading.

So you can leave the sensor for a while to get it stable. So here I am back after 20 minutes. So you can see the stable reading. The index of air quality has become almost more than 25 along with CO2 values of more than 500. Now you can do small modifications to this code.

Uncommend all these lines, this will make it easy to get the readings identified in the serial monitor. You can again upload the code. After uploading, open the serial monitor. So now you can see the value of pressure, temperature, humidity, IAQ, CO2, and VOC. The gas values are in PPM.

This is how you can display and identify the pressure, temperature, humidity, IAQ, CO2, and VOC readings. It's time to set up the Blink app now. So go to the Play Store and download the Blink app. Click on a new project and name it anything. Select the MCU board from the list.

Click on create. Now click on the setting and then hit the email button. This will send an authentication token to your mail. This token is required in the code. Now, open your email that you received from Blink.

Now copy this authentication token from here. This is a complete code for the project now. You need to add some blink library as well as ESB 8 to double 6 library. From here, replace the authentication token. Also, change the WiFi SSID and password.

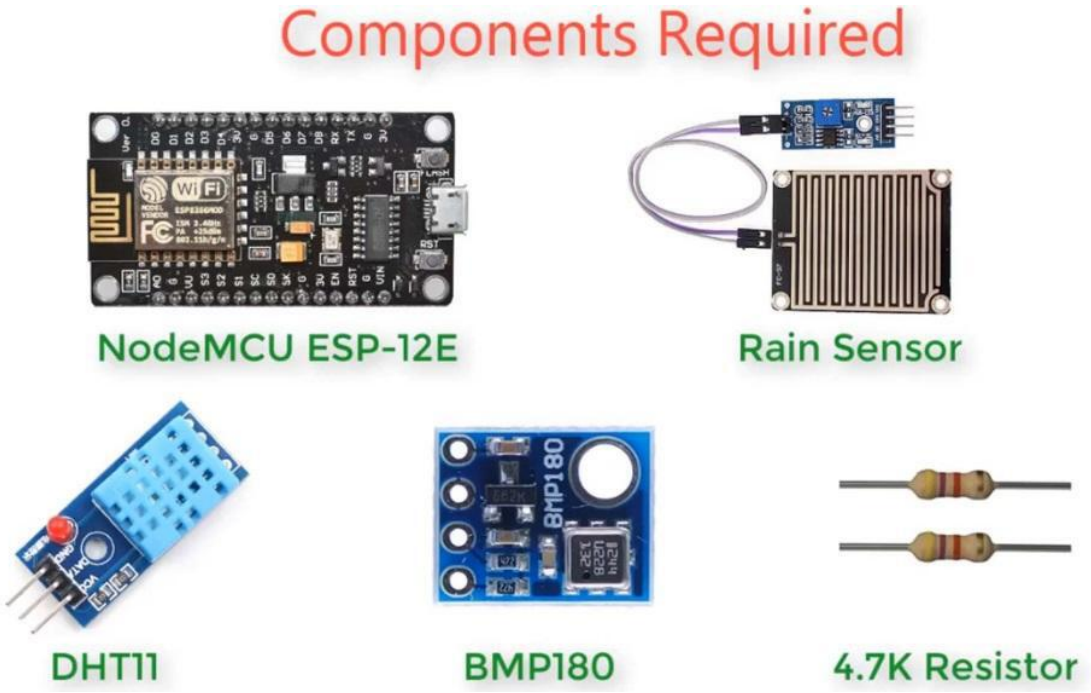
Here are some lines that are aided to upload the BME 680 data from ESP 8 to double 6 to blink cloud. Now go to the Blink app and add 6 different types of widget. assign a virtual pin and name the variable to every widget. After all the setting is done, Click on the play button here. So here is the data from the sensor on the blink app.

The data changes whenever the sensor pushes some values. The beautiful cause for pressure, temperature, humidity, IAQ, CO2, and VOC will appear here.

IOT LIVE WEATHER STATION MONITORING USING ESP8266

An integrated project will learn how to make a live weather station using modem CU, BMP 180, DST11, and red sensor. We'll be interfacing all this sensor with node MCU and displaying the data on the Internet using a web server. So, this is the beautiful engine that is displaying daytime temperature pressure, then, as well as humidity. So, how to do this, we'll be learning here.

So let's get it started. So for the components required for the sun, modems USB, 8266 twili board. a ring sensor. We need a DHT 11 humidity or temperature sensor. You can use DHT 22 as well.

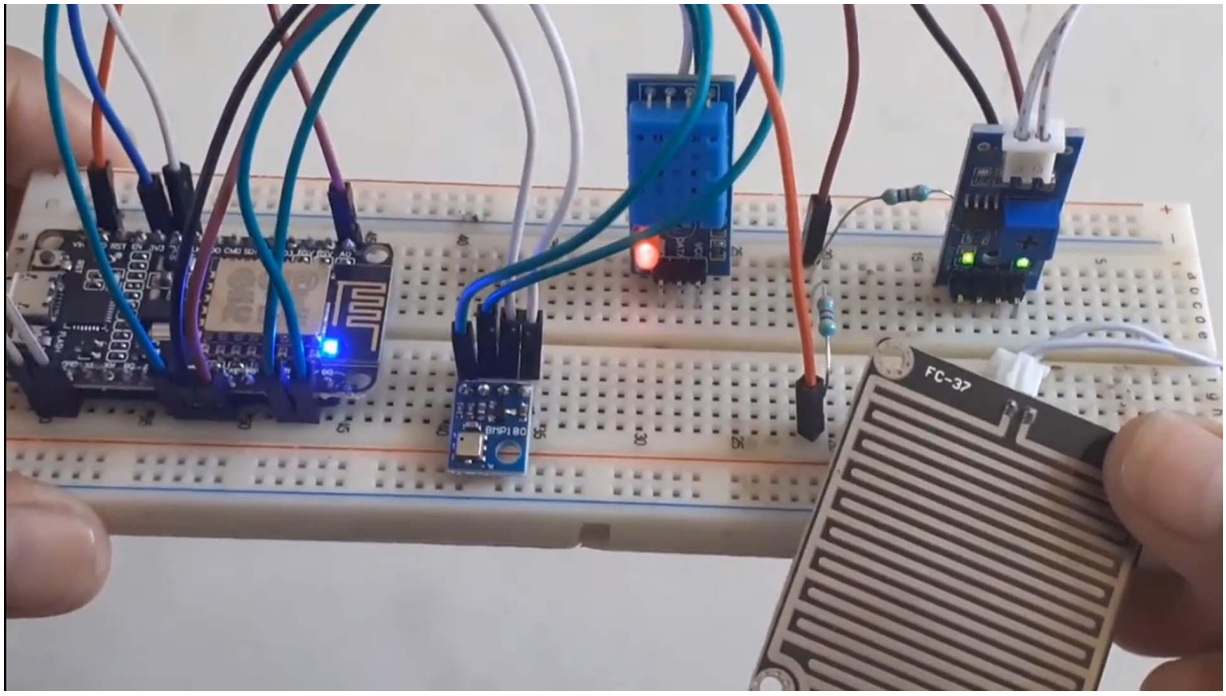


BMP 1 or BMP 280 barometric pressure sensor, and we need 4.7 kilo am registered. That is of two numbers. So Let's go to the circuit diagram. The BMP 180 is ITC protocol based, so you'll be using an SDN SCL printer. It's the 1 and 2 two of node MCU similarly, we'll be connecting and developing a BST 112.

Must be fine. Similarly, the sensor is connected via point 7 kilometer register using voltage divided network, and it's connected to a knot. So this is how we have assembled the same circuit on bareboat. So this is the BMP180, a barometric pressure sensor. And this is DHT11.

humidity and temperature sensor. So this is FC 1 sorry, FC 37 range sensor. It also interfaces with node MCU, and we have used a 4.7 k resistor for the voltage divided network. So let's take a closer look at it. So this is a node MCU, and we have the MP180 barometric pressure sensor.

and DST11, which data feed is connected to Defy. And, Dan's sensor for detecting the quantity of rain. So I have supplied power to it. Now, let's go through the coding section of how the code is and how the code works. So this is the code.



So we are using this all these libraries. So these smaller libraries are for communicating between web server. So We have the PMP 180 library, and we have the dstsp.library. We have a wide library for ITC communication. And BTSP for DST11.

So we'll be adding this to the library. So given the link in the description download both of this library and aid. So I have added both of these libraries here. You can see Now, here you can see a one header file. That is written as an index dot at So how to create this file.

This file is for displaying HTML on one web page using JavaScript. So this is the file that I was talking about. That includes index dot x. So this is an HTML file written with JavaScript. So here you can see.

Temperature. Similarly, and humidity. So all parameters will be Apart from this, we'll also be displaying time and clock. So you can see the time and clock command. So save this file as index.h, and save it to the folder where your adeno footage looks.

It's the code that won't compile. So I'm renaming it as index dot apps, and I am saving it. So the file is saved. Now, open your, you know, code folder. And here you'll find a file called index dot h.

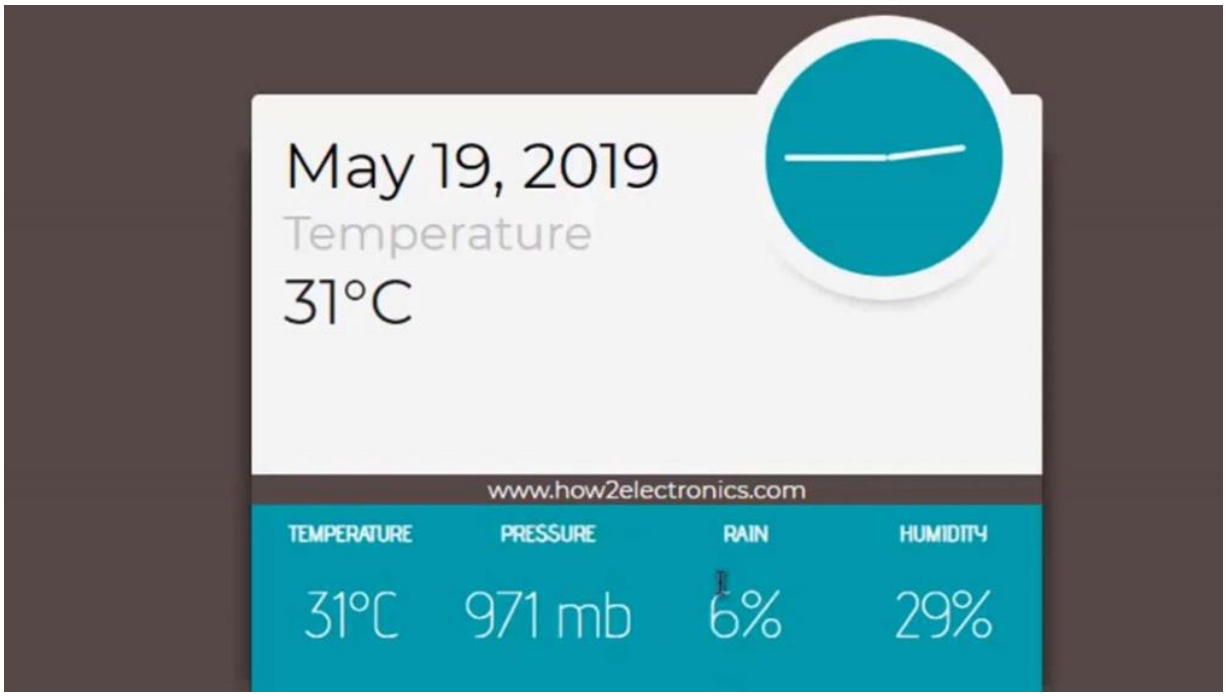
Now, this is a very, really important step. Without this, your code won't compile. So this is the code, the same code that I was showing you earlier. So now, we have found LADF for the pin tool and DST pin up in 14. So, similarly, this is a BMP 1 beginning, and LTV is defined in meters as 1655.

```
</div>

<script>
setInterval(drawClock, 2000);

function drawClock(){
    var now = new Date();
    var hour = now.getHours();
    var minute = now.getMinutes();
    var second = now.getSeconds();

    //Date
    var options = {year: 'numeric', month: 'long', day: 'numeric' };
        var today  = new Date();
    document.getElementById("date").innerHTML = today.toLocaleDateString("en-US", options);

    //hour
    var hourAngle = (360*(hour/12))+((360/12)*(minute/60));
    var minAngle = 360*(minute/60);
    document.getElementById("hour").style.transform = "rotate("+(hourAngle)+"deg)";
    //minute
    document.getElementById("min").style.transform = "rotate("+(minAngle)+"deg)";

    //Get Humidity Temperature and Rain Data
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
      var txt = this.responseText;
      var obj = JSON.parse(txt); //Ref: https://www.w3schools.com/js/js_json_parse.asp
        document.getElementById("rain").innerHTML = obj.Rain + "%";
        document.getElementById("temperature").innerHTML = Math.round(obj.Temperature) + "&deg;C";
        document.getElementById("temp").innerHTML = Math.round(obj.Temperature) + "&deg;C";
        document.getElementById("humidity").innerHTML = Math.round(obj.Humidity) + "%";
        document.getElementById("pressure").innerHTML = Math.round(obj.Pressuremb) + " mb";
```

So change the name of SSID and password. Here I have used mine. You need to check. And we have used the server as a t port as a t. So select the correct comport and compile the code.

And after compiling, just upload the code. So here you can see the code is being uploaded. So, the code is uploaded successfully. As you can see here, once the code is uploaded, click on the serial monitor. So here, when the wifi gets connected, you'll get the IP address of your modem view.

Just copy the IP address. Now, open your web browser. I am using Google Chrome. So paste the same IP address and hit intern. So you can see once the IP address is SS.



So you can see it is displaying a beautiful widgets suite date today's May 19, 2019, and you can see my room temperature is 31 degrees c. Below, you can see temperature data pressure in MB rain in percentage So this is the correct ring, like, and you are also seeing the humidity in percentage. Similarly, you can see the same on your serial monitor. So this is the data that is displayed on the serial monitor, and it contains more detailed information. So you can see the temperature display in both degrees c and f.

You are seeing absolute pressure, a little pressure, and computed altitude. You are seeing humidity temperature run through the hull as well.

IOT LORA BASED SMART AGRICULTURE WITH REMOTE MONITORING SYSTEM

In this project, we'll be learning about IT based soil nutrient monitoring and analyzing the system using Arduino and ESB 32. Soil is the base of agriculture. So it provides nutrients that increase the growth of a crop.

Some chemical and physical properties of soil such as its moisture, temperature, soil, nitrogen, phosphorus, and potassium content heavily affect the yield of a crop. These properties can be sensed by the open source hardware and they can be used in the field. In this project, A soil nutrient monitoring and analysis system is proposed in which the farmer will be able to monitor soil moisture, solar temperature, and soil nutrient content like nitrogen, phosphorus, and potassium. The farmer can monitor all these parameters wirelessly on a mobile phone or the crystal system. To measure the soil moisture, we will use a capacitive soil moisture sensor.

The temperature of the soil can be measured by using a ds18b20 waterproof temperature sensor. Similarly, in order to measure the soil and PK values, we will use soil and pick a sensor. All these sensors can be easily interfaced with the Ardeno. We will use the ThinkSpace server to monitor the data in graphical and numerical format. We will use an RF2401 wireless transceiver module to send the data from sensor node to gateway.

The data from the transmitter can be transmitted wirelessly from a kilometer distance to the receiver. The receiver is built using the ESP32 Wi Fi module, which has access to the Wi Fi network. using the swap and network, the data can be uploaded to Think's Big Server. So let's build an IT based soil nutrient content analysis monitoring and testing system simply using wireless sensors network, Ardeno and ASV32. This project is sponsored by Next Pacific.

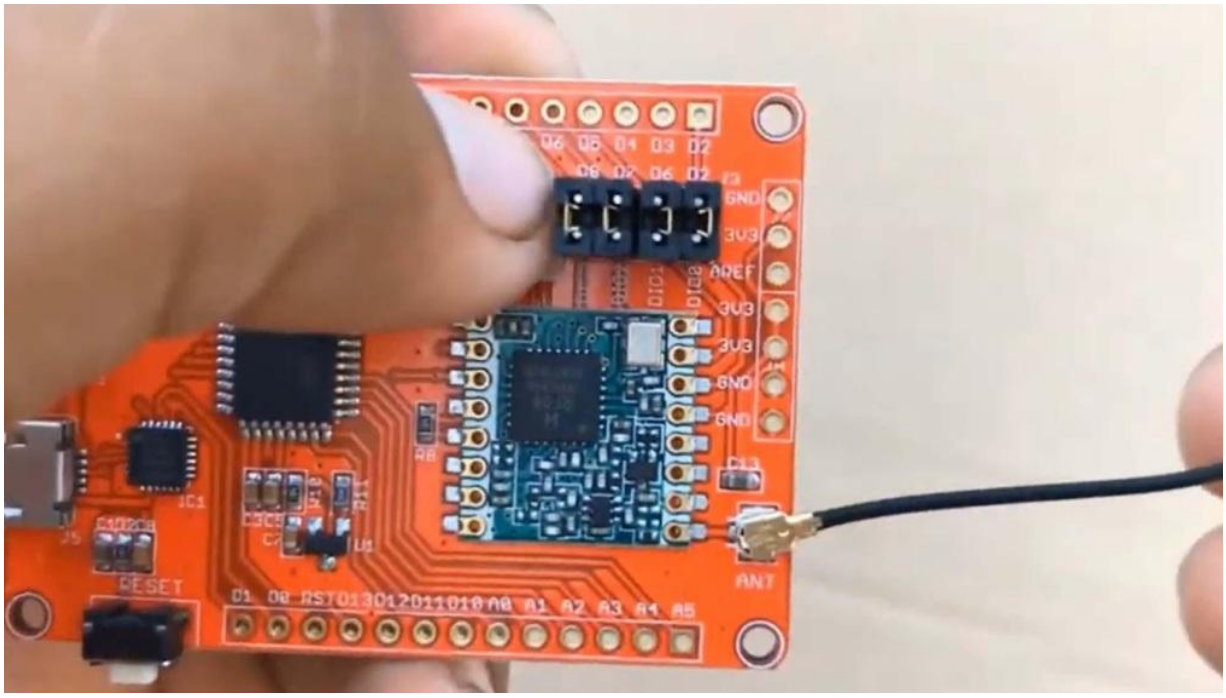
Currently, Next Pacific has developed a PCV design analysis software called NextDFM. which is a design problem detector or an engineering solution provider. To use this software, you need to sign in using the next PC account. You need to import a Java file with just one click. The project graphics are easy to read, so you can make sure that the file contains all the necessary data.

It helps you quickly familiarize DFM Design specification and production needs to determine whether there are any manufacturing constraints. It also effectively checks whether there are hidden dangers in the design file, selects available parts, and evaluates the

cost in real time. So try to start using the stool for a recipe analysis. The link is given in the description. Welcome back.

Now let's just see the components required for this project. The first sensor that we need is the soil and pick a sensor. The sensor measures the content of nitrogen plus ferrous and potassium in the soil and helps in determining the fertility of the soil. For this, it uses a modbus protocol for communication. Then we needed a temperature sensor, so I chose a DS18b20 waterproof temperature sensor.

This sensor can be inserted into the soil easily, and then the soil temperature can be measured. The sensor is based on a one wire protocol. We will also need a soil moisture sensor to measure the soil moisture content. For that, I selected the capacitive soil moisture sensor. The soil moisture sensor can be easily inserted into the soil, and soil moisture content can easily be determined.



The sensor is an analog type and gives analog value output. Apart from all these sensors, we need a wireless communication module So, I chose a pair of NRF24l01 wireless receiver modules. Each module can both sync and receive it a wireless It works within the

frequency of 2.4 gigahertz. The modules, when operated efficiently, can cover a distance of eleven hundred meters, and I choose ESP 3rd to do wifi models for making the gateway or the receiver circuit. Then I selected the RD unit enabled for making the note or transmitter circuit, This is the RS 45 mode module used with URDU and Sewer and picked a sensor to read mode data.

And finally, is the 4.7 k resistor used to beat the DS 1 8b20 as a pool upper resistor. Now let's see the schematic or the circuit. So this is the circuit for the transmitter or sensor node. As we can see, all the sensors are connected to the Arduino Nano Board. The Aneta 24l01 transceiver model is connected through SPI pins here.

All the components are powered through Juno 360 ground pins except the soil and pick a sensor, which requires it to avoid power supply. So, here is the complete assembly of the sensor node on a break board. This is the cell and picks a sensor antenna temperature sensor and a soil moisture sensor. This is the Anurag 24l01 module with upgrade 2.4 uharzantina. This one is the MAX 485 Modbus module, and then RD nano board and a 4.7 k versus are connected to DS 18b20 and Mississippi.

And this is the receiver or gateway circuit The receiver is made using ESP 32 Wi Fi module and NRF24l01 module with the SPIP interface. So you can see the symbol circuit here on the break port. The same 2.4 gigahertz wireless receiver module is used for receiving data. We will now state that the thing is a big server to receive the data on the cloud from our hardware. Things big server is an IIT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud.

All you need is to visit big dotcoms and create an account. After creating an account, click on the channel and create a new channel with 5 different variables like nitrogen, phosphorus, potassium, soil temperature, and soil moisture. After that, save the channel. Now

quickly go to the API keys and copy the API key, which will be used in the code. Now let's just move to the programming part.

In the transmitter program, there are many libraries used which you will get from the description link. In the receiver code, replace this API key with your API key. Also change the WiFi SSID and password. From this line, you need to put some factor for calibrating the soil moisture sensor. Using the function, we will send so many bytes of data to the receiver using the 24l01 module.

The rest of the code is the same. You just need to read the sensor data from sensors and send it to the receiver circuit. The receiver will upload the data to the Think Speak Cloud Suburbs. Upload the code of the transmitter to the Arduro network and also upload the code of the receiver to the ESP 32 board. After uploading the code, your device is ready for testing, evaluation, and observations.

Open the serial monitor for both devices. The sender will send the data to the receiver, and this receiver will initially connect to the wifi network. All the happenings can be observed in the serial monitor. The gateway collects the data and uploads it to the sync specs

server. You can just visit the private view of the sync speed dashboard and you will see the data being locked into the sync specs server.

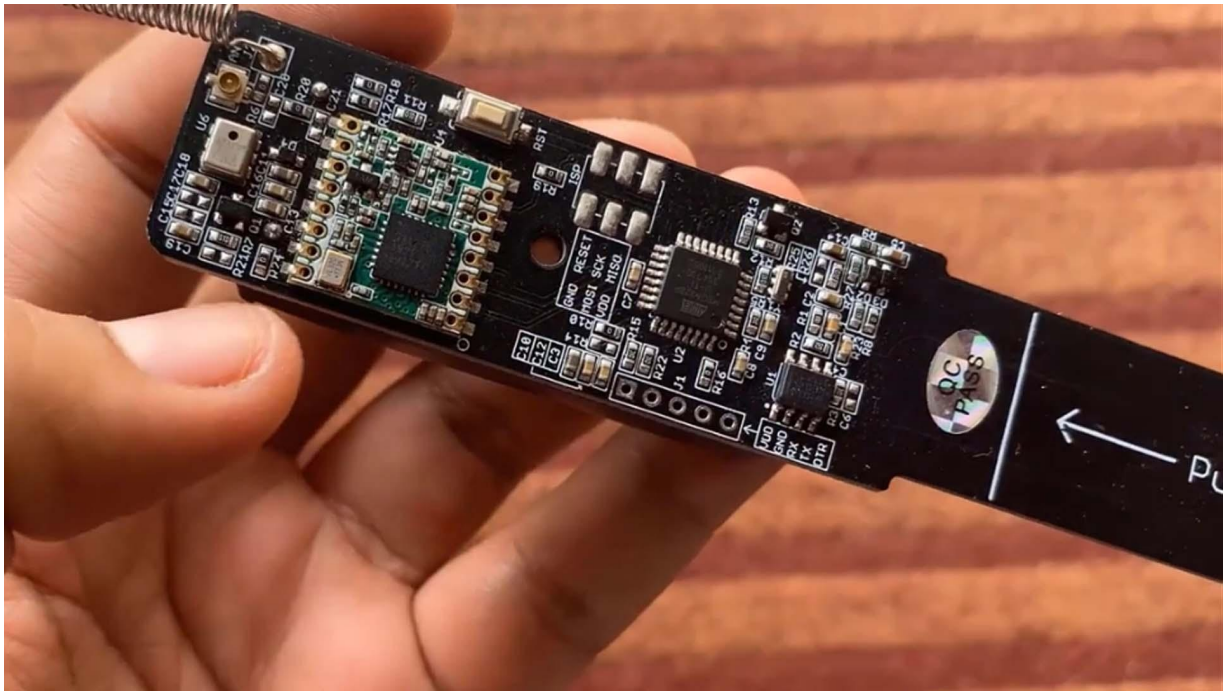
The data is uploaded after an interval of 15 seconds regularly. This is the graph of nitrogen phosphorus, potassium, soil moisture, and soil temperature.

IOT LORA BASED SMART SOIL SENSOR NETWORK DATA MONITORING SYSTEM

I explained smart agriculture using LoRa based soil moisture sensor. The sensor is capable of measuring the soil moisture value. It can then send the sensor data, wireless the overall LoRa radio or a wireless sensor network. But the limitation with the previous project was The sensor data can only be monitored on a serial monitor. But this time, I came with the project operation.

Instead of using a serial monitor, You can observe the sensor data on a web server network. For this, I incorporated ESP32 and LoRa based gateway network, which has Wi Fi accessibility. The LoRa soil moisture sensor acts as a sensor node. It reads the soil moisture value in analog form after an interval of every 5 seconds or more. You can set the timing in the code part.

The sensor node sends the data to the gateway using the LoRa radio. The receiver is also made using ESP32 and the LoRa module. The receiver collects the data from 1 or more sensor nodes. Then, the receiver creates a local wave server using the ESP32 inbuilt library function. You can access the web page using the IP address that appears on the OLED display on whether the soil moisture value is good, normal, or axis.

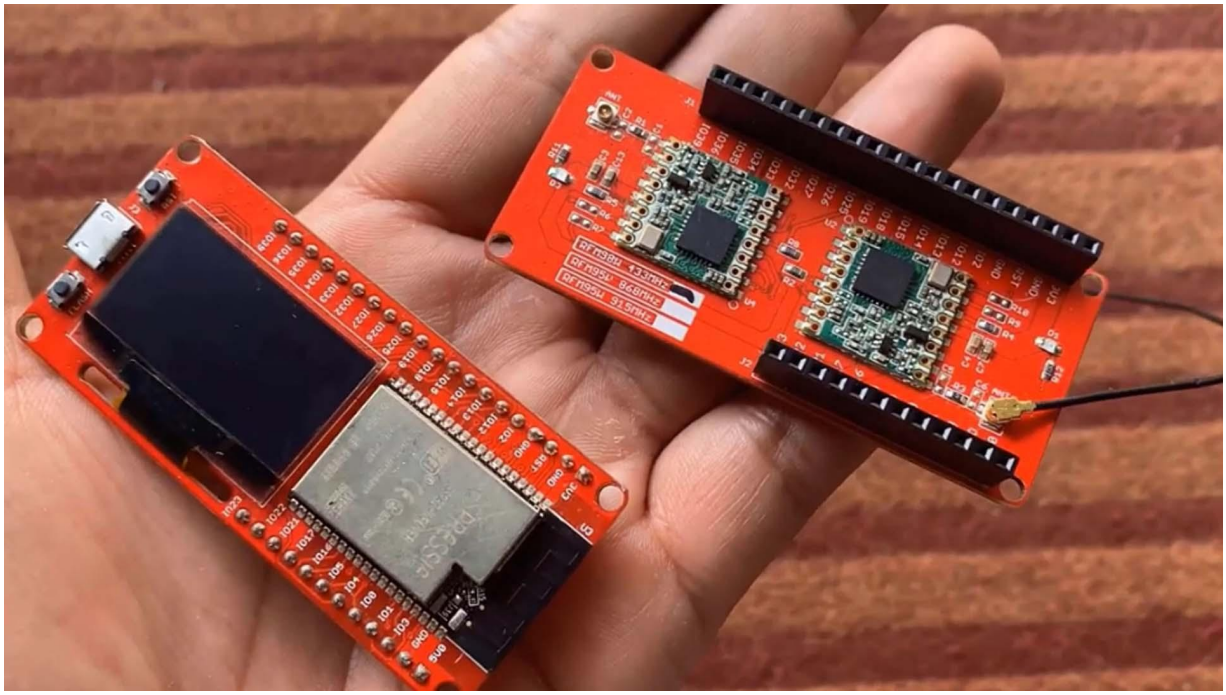


While the project is a little long, I will explain everything in detail, So, without getting delayed, let's say this interesting tutorial. Most of the PCB boards using most of my projects are sponsored by the next PCB. The next PCB is one of the biggest bakery manufacturer companies in China. They offer the PCV board and PCB assembly services at the lowest affordable price. You can get a trial PCV to their PCV, and full air PCD with pre PCB assembly shipping services up to a fast lead time of 24 hours.

There is a special offer for you from PCB . Share your advice comments on the next piece of your products on your social media to participate in the Lacidra, which is a 100% willing way to receive gifts $300 cash balance, $100 amazing gift cards, and coupons are waiting for you to draw. Welcome back again. Let's first see this LoRa soil moisture sensor node. So this is the transmitted part that consists of the LoRa soil moisture sensor and the LAST 10 humidity temperature sensor. This is a capacitive soil moisture sensor that is painted with waterproof paint for antiquorism.

This is an atmega328 microcontroller which is preloaded with Audio Pro Mini bootloader. This is a LoRa module called RFM98 with an

operating frequency of 433 Megahertz. The LoRa module is interfaced with the atmega328 controller using the SPI pins. There is already a helical antenna for the LoRa radio. Apart from that, You can use an external antenna of 1.5 dB or a higher value for maximum distance.



There is a triple 5 timer IC Forecap zitive soil moisture sensor, which generates from analog voltage with variable frequency. At the back of this module, there is a slot for a AAA battery. You can use a pair of AAA batteries to power this device. The device is already low powered, so it is designed to operate at low voltage with a removable battery. I purchased a pair of batteries for this module.

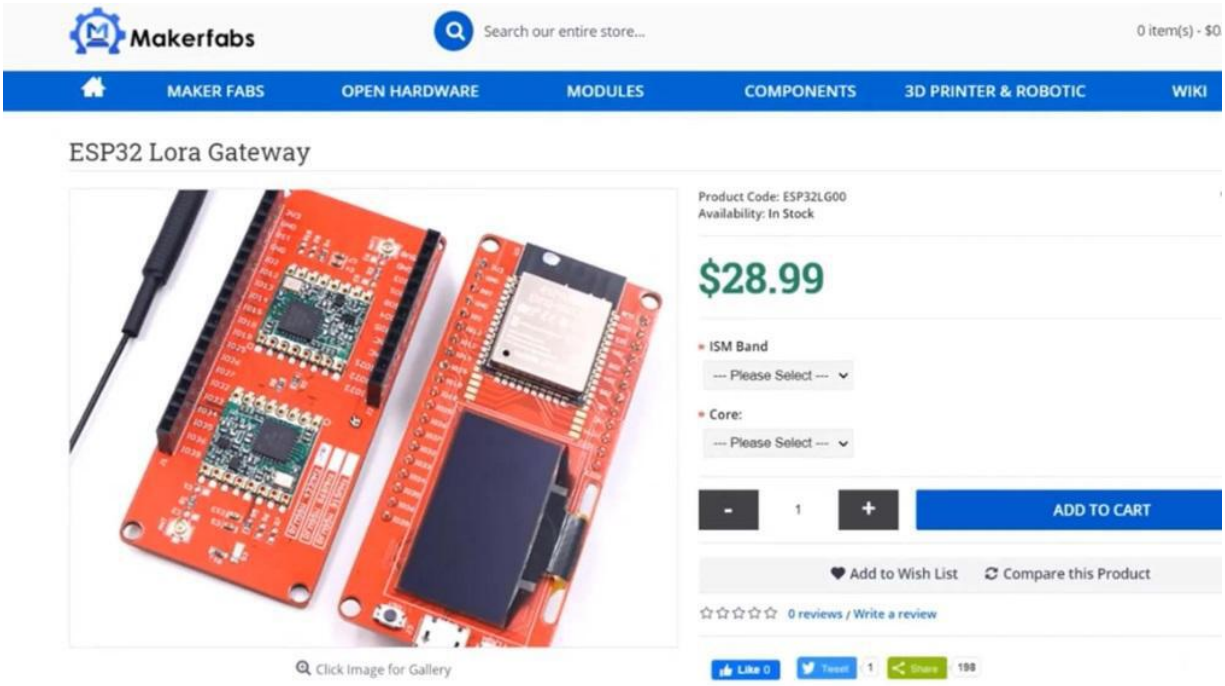
Simply install it on the battery connector. Be careful about the battery polarity. as there is no reverse polarity protection on this module. After inserting the battery, the device is powered on and ready for operation. The most important part is programming this port.

So here is the UR connector port. The pins are named VCC GND, Rx, DX, and DTR. You need any USB to TTL converter module to program this board. I prefer the CP 2104 driver module for this

application. This module is manufactured by maker fabs and has a simple interface.

There is a voltage selection option as either 3.3 volt or as a 5 volt. This FDDI module directly fits on the LoRa soil moisture sensor node. Now, let's talk about the gateway. The gateway is made using ESP32 and the LoRa module. This LoRa gateway is based on the make Python ESP32 and 2 channel LoRa expansion board.

This is the first board made using the ESP32 wifi module. It has the same number of GPU pins as in case of other ESP 32 boards. You can see the pin mapping and details of input output ports from SPI to I Square C or even UR pins. The best part about this board is it has a 1.3 inch OLED display. This can be useful while displaying something.



This is the Make Python LoRa Expansion shield. The combined board contains the LoRa module, RFM95W, which handles the 433 Megahertz band. The LoRa gateway offers bandwidth options ranging from 7.8 kilohertz to 500 kilohertz, with spreading factors ranging from 6 to 12. The 2 LoRa gateways are used for long

distance communication in the complex environment of the city. The actual test distance reaches 2.8 kilometers.

You can either use 1 LoRa module in a gateway or both at the same time. The GPU pins are similar to the ESP32 boards, so that the shield can be fitted perfectly on the board. You just need to combine these boards together. Hence, your long distance LoRa gateway is ready now. You can purchase the LoRa soil moisture sensor at $14.90 from this link.

In case if you want the 3 d casing, then you can add an extra $2 for the casing. Select the frequency band and then press the order. Similarly, here is the LoRa gateway as well. The gateway will cost you around $28.99. You don't have to buy 2 boards separately.

Rather, you can buy them together, select your frequency, and place an order. Now, let's set up the LoRa sensor node. This is the code part. You will need a LoRa radio library. The LoRa SPI pins along with all the power pins are defined in the code.

We are reading the analog value of the LoRa soil moisture sensor from the code. Then, simply we are sending it to the receiver using the LoRa code. To upload this code on the board, connect SP to TTL module with a micro USB data cable, and then connect it to the sensor node. On the other side, connect the cable to the computer. From the tools menu, select the Ardenopromini board with 3.3 volt bootloader, and then you can upload the code.

After uploading the code, open the serial monitor. If everything goes fine, then LoRa radio will initialize as okay. and it shows the preamble length along with the analog sensor value. You can map the analog sensor value to a percentage as well if you need that. My current analog sensor value is around 875, which is when the sensor is dry and exposed to air.

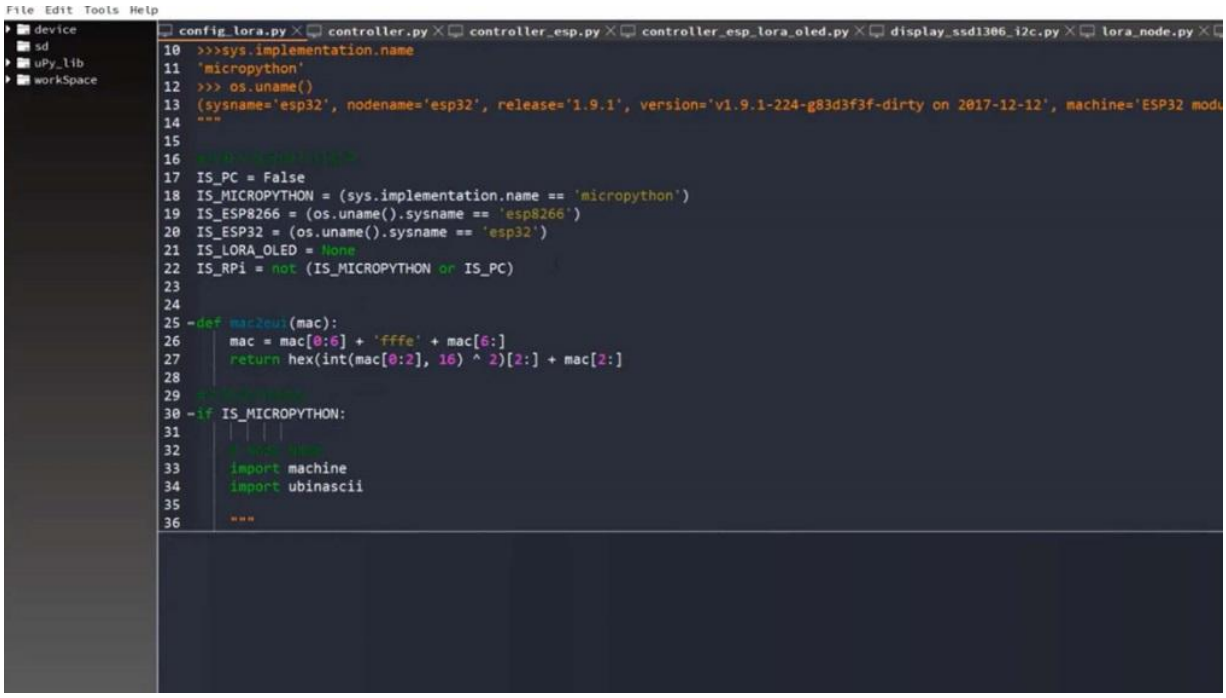
The value goes around 520 when put in water. To test the sensor, select a suitable site For example, you can select the garden. The garden may have a variety of soil and moisture values depending upon the season or rainy conditions. Just insert the probe of the sensor into the soil like this. Do not dig the soil.

All you need is only placing the sensor. Give the electronics part away from the wet sand as they are not waterproof. As soon as the sensor is inserted, the analog value is read by the controller. The analog sensor value rises up and may go down depending upon the availability of the soil moisture. I would like to test this sensor for a couple of weeks to really know about the performance of the sensor. I initially tested it for 3 days, and so far, the result has been pretty impressive.

Alright. The sensor data is required to be sent to the gateway. So let's set up the gateway now. connect the micro USB cable to the combined gateway. The gateway requires micro Python code.

To use the micro Python code, I prefer the new Minecraft ID. From the tools menu, select the ESP 32 board and also the comport. Initially, it will ask to load the micro Python firmware to the board. So select the board and then address and it is the flash and select user. So there is a bin file that can be selected and used as a framework from the ESP 32 micro Python.

Now the firmware will be uploaded in a minute. So once the firmware is uploaded, You then need to load many pie scripts to the ESP 32 board. So load all the pie scripts to ASP 32 1 by 1. The PyScript is for the controller, wifi, LoRa module, web server, OLED display, and other functionalities. Once all these scripts are loaded 1 by 1, You can press the reset button on the ESP32 gateway.



The console window will display the IP address along with the Soul Moisture received data. On the hardware part, you can again press the reset button. So on the OLED display, you can see the IP address of the entire board. Using this IP address, you can get the 12 booster data on the web server. Now, it's time to test the device in different conditions.

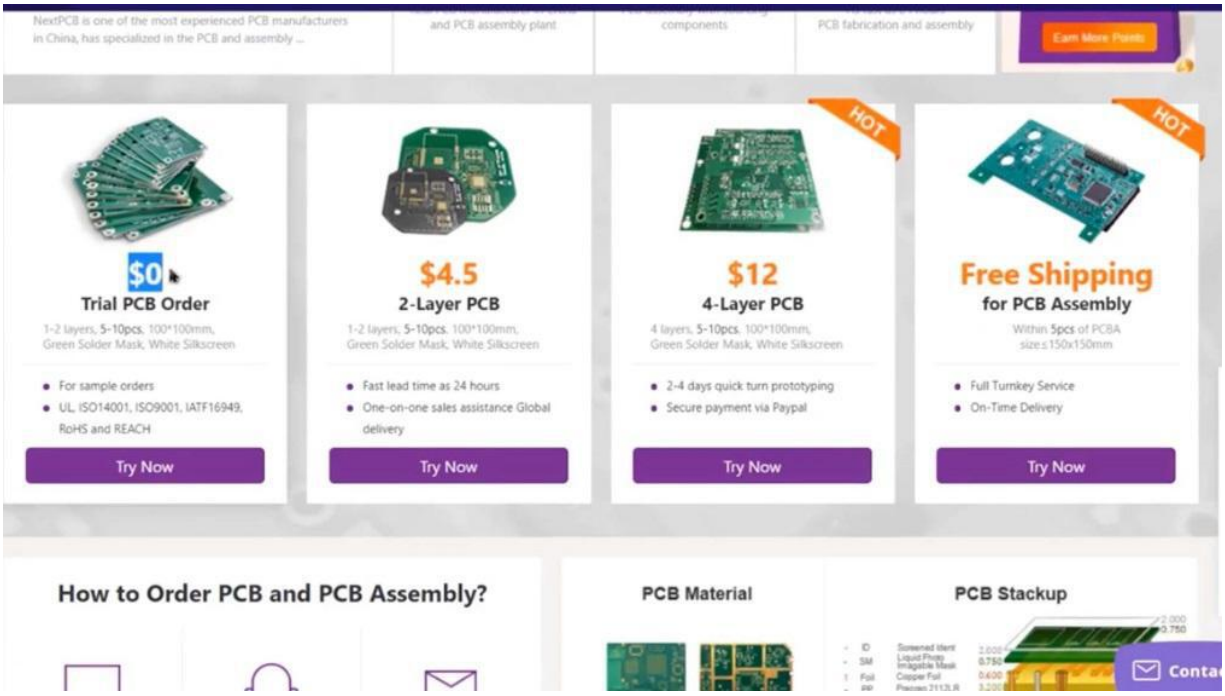
First, visit the IP address on your favorite web browser. So here we go, a beautiful web page from maker fabs. The 12 sensor value is 832 means it is dry. Now put the sensor in the soil. Now, you can see the soil moisture value is very good.

Now, remove the soil moisture sensor from the soil and put it in water. And then refresh the web page So here we go, the data shows the swell is too wet. This is how you can monitor the swell moisture data on the web server.

IOT MQTT BASED HEART RATE MONITOR USING ESP8266 ARDUINO

I explained how you can use the easy pulse sensor with audio to measure the pulse rate or BPM value. I also displayed the BPM value on the OLED display. The pulse sensor has a clip that can be easily fitted on the finger. But this time, I got another sensor similar to the previous one. The only difference was instead of putting the finger on the probe, This time, we will attach it to the earlobe.

So this is an earlobe pole sensor that is highly accurate and can be used for learning purposes. This time, I would like to use this sensor with node MCU ESP 8 to double 6 boards. And using some cloud server, I want to monitor it online. So, I preferred the MQTT protocol for sending the data to the server. The best MQTT platform I have been using for a long time is the USDOT.



using UV dots, you can visualize and analyze the sensor data from any part of the world. So let's learn in detail how we can use this pulse sensor with ESP 8 to double 6 and display the VPN value both on an OLED display as well as in the cloud. This project is sponsored by Nexpicity. Next, PCB is one of the biggest PCB manufacturers in China. They offer PCB board and PCB assembly services at the lowest affordable price.

You can get trial PCB 2 layer PCV and 4 layer PCB with free PCV assembly sleeping services up to a fast head time of 24 hours. There is an exciting giveaway offer for you from the next PCB. You can win exciting gifts like Xiaomi's screwdriver, mini wear oscilloscope, Xiaomi sports watch, portable soldering iron kit and so many. 3 winners will be chosen from the participants. To participate in this giveaway, you can check the next PCB official Instagram account.

The link and rules for participation can be found in the description. Welcome back again. This is the easy process that was sent to me by an Indian company called Grey Logix. This is one of the best poll sensors available on the market and works better than the world

famous poll sensor. This pulse sensor has a clip that can be attached to the earlobe.

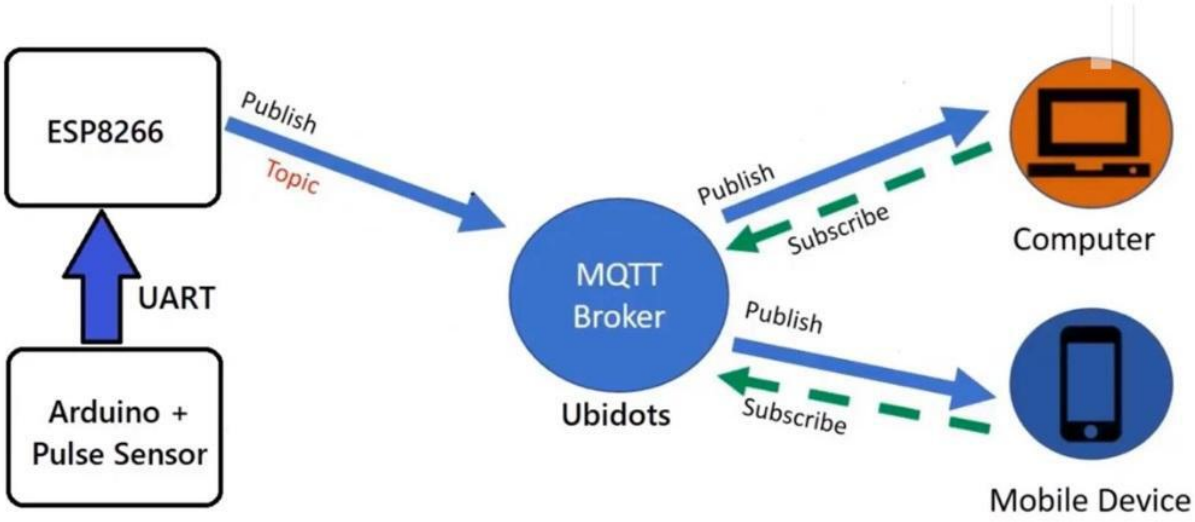
The sensor uses an infrared source to illuminate the finger on one side. On the other side, a photo detector measures small variations in the transmitted light intensity due to a change in blood volume inside the tissue. I got the 2 versions of this sensor. You can use any of the sensors for this project. 1 of the sensor clipped to the earlobe and other to the finger. The result will be almost the same for any of the sensors.

I prefer an earlobe type sensor for my application. For easy and clear understanding, let's go through the block diagram. We first connect the pulse sensor to Ardeno using UR communication We send the data from RD node to node MCU ESP 8 to double 6. We could have directly connected the pulse sensor to node MCU ESP 8 to double 6 boards. but the pulse sensor does not seem to be working and giving any output on the serial monitor, so I preferred this method for this project.

The node MCU ESP8 to double 6 connects to the Wi Fi network. It then uploads or publishes the BPM topic to the MQTT cloud called Ubidots. The published data can be viewed on computer or mobile phones using the Ubidot's dashboard as they are the subscriber. We can convert this block diagram to a circuit diagram. The node MCU and arduino nano are connected to each other via a UART software serial pin. The pulse sensor is connected to an arduino alloy pin and an OLD to node MCU, I square C pin.

I use a breadboard for assembling my circuit. This is a node MCU board that is connected to our Then we have an OLED display and pulse sensor connected to the circuit. Alright. Let's now configure Ubidot's account. Sign in to Ubidot's account or create an account if you are new to it.

After signing in, you can quickly go to the dashboard. Currently, no devices are available here. In the code part, we will need the pulse sensor library for Arduino Nano. In this part, we are reading the analog value from the pulse sensor output and sending it to the node MCU board via UR communication. In the node MCU code, we are receiving the same data via serial communication.

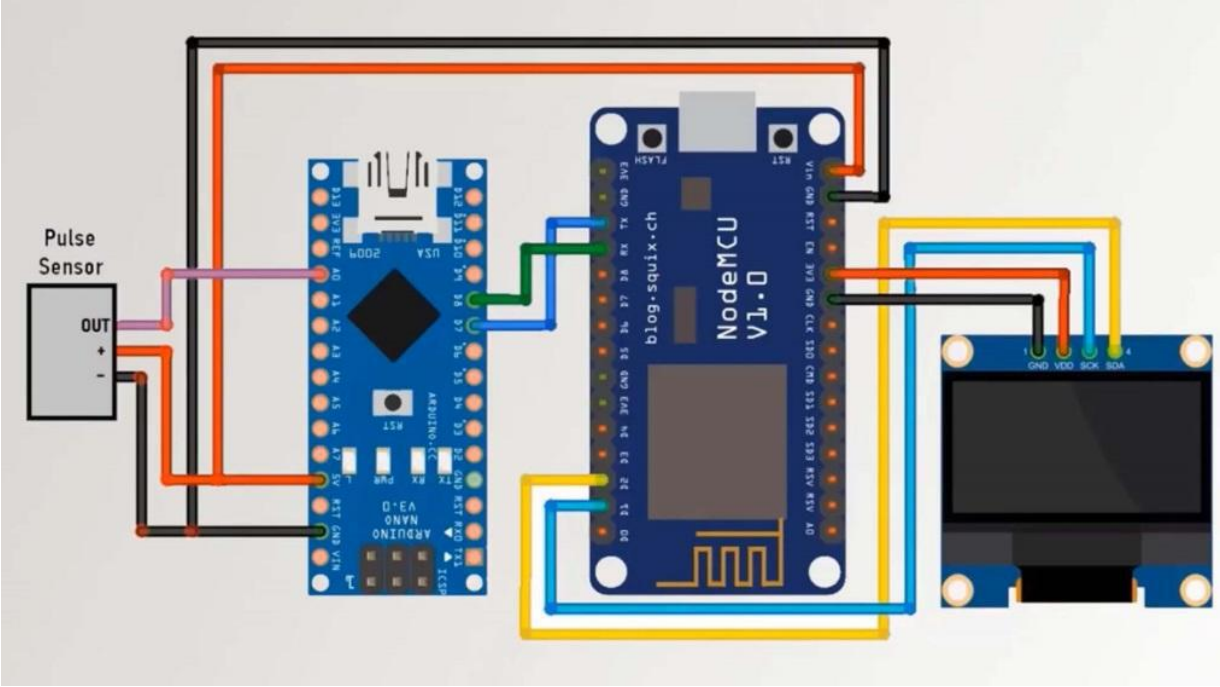


For this node MCU code, we need a couple of libraries for OLED display. We also need ESP 8 to double 6 and MQTT library for UBDOTs. Remember to install a pop- up client library as well. From this part, you need to change the WiFi SSID and password And in this line, you need to change the USB token. To get this token, go to the API credentials section of UBS dashboard.

Click on this default token and then copy it. Go back to the code and paste it here. Now, upload the first code of the Arduino Nano board and second to the Anode MCU board. After uploading the code, it's time to measure the heart rate. So connect the earlobe probe of the sensor to the earlobe of the patient. It's just like a clip and it's very easy to attach.

then open the serial monitor for Arduino Nano. So, here we go. The pulse sensor has started working, and the pulse sensor data is displayed on the serial monitor. initially, it will display the false value. Then the value becomes correct later.

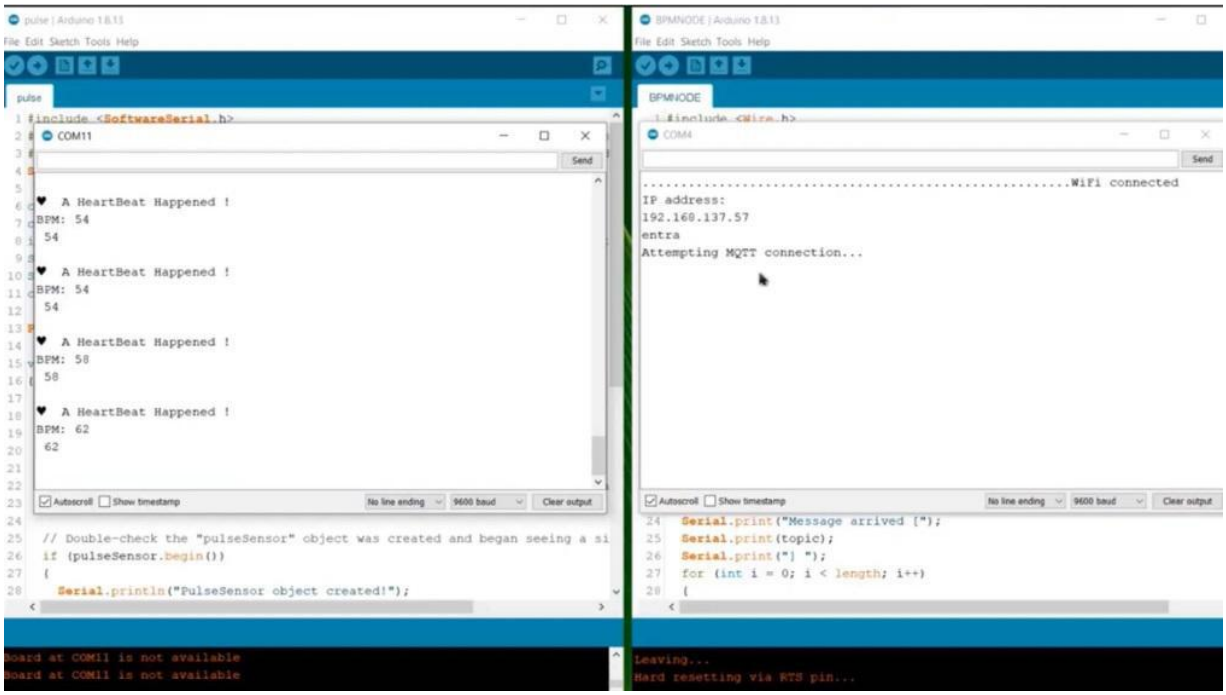
You can also view the BPM value on OLED. Currently, the heart rate shown here is around 76 PPM, which is the average value of a person. The LED will also blink as the heartbeat is detected. Now open the serial monitor for node MCU as well. The node MCU will connect to the Wi Fi network.



Then it will start pushing the heart rate value to the ubidots server via MUTT connection. Now, go to the ubidots dashboard and refresh it. So a new device is added to the dashboard. Click on the device. You will see the latest logged value of BPM on ubot Click on this value, then you can see the graphical representation of logged data along with the data in a table.

Go to the demo dashboard now and add widgets by selecting the new widget from the list. In my case, I use the GaOS indicator for the widget. Then select the ESP a to double 6 board from the variable

list and further select the BPM. Give a name to your website. Also, assign the minimum and maximum value as soon as everything is set perfectly.



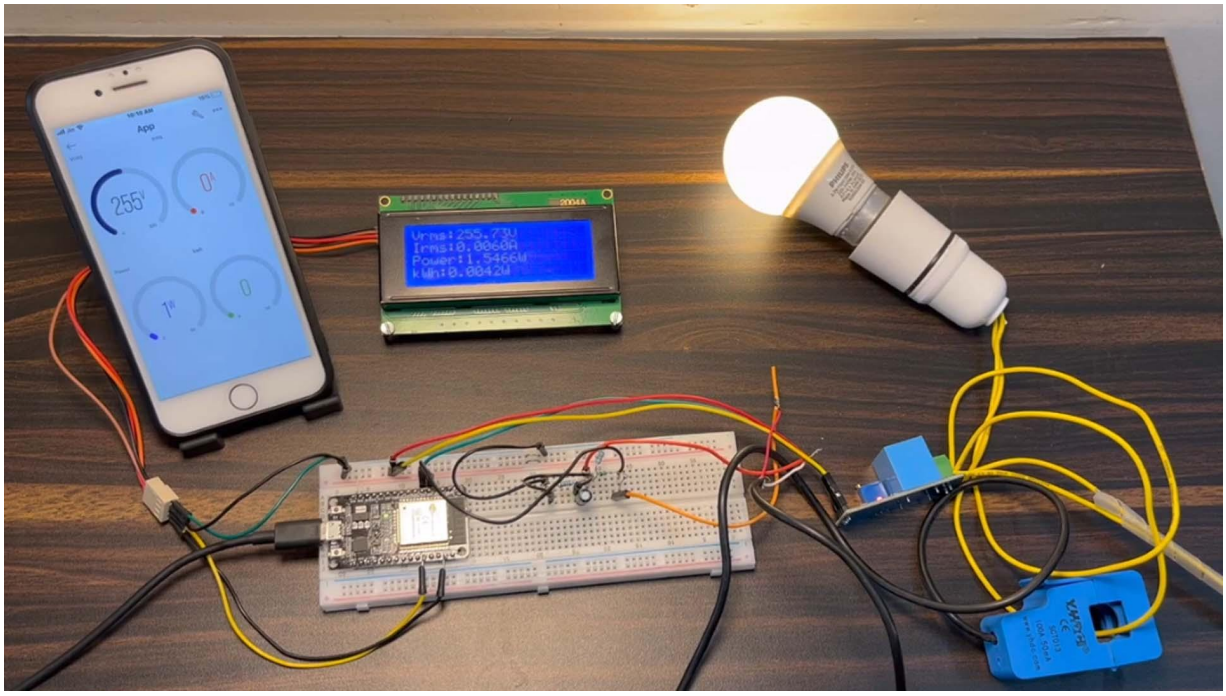
the BPM value starts sowing in a gauze. You can minimize or maximize the gauze to observe and monitor the heart rate value. The best part about this is you can monitor the data from any part of the world. I didn't like the purple color, so I changed it to green now. Not only using the computer, but you can also monitor the hard data online on the mobile dashboard as well.

Just log in to the ub docs dashboard using your favorite browser, and then you can monitor the hard data here. This is such an interesting and exciting project to watch. A few years ago, nobody thought that this could monitor the patient's health status online from any part of the world, but now it's possible through IoT.

IOT SMART ELECTRICITY ENERGY METER USING ESP32 BLYNK 2.0

I developed an energy meter project using the blink application. For that, I used a CT013 Coordinate Center and the JetMPT 101 V Voltage Center. This attention was pretty accurate in reading the current value, voltage value, as well as power, and total energy However, the blink application recently underwent an update, rendering it on this project. As a result, I decided to rebuild the project using the newly updated Blink 2.0 application. In this project, we will build an IoT based smart electricity energy meter using the ESP32 and the newly updated blink 2.0 application.

By using the best current center, SCT 013 and voltage center, second PT101b, We can measure voltage, current, power, and total energy consumed in kilowatt hour. The readings will be sent to a blink 2.0 application and displayed on a dashboard accessible from any location. In case of power outages, the energy meter data will be stored in the ASP, AEP, RVM memory. Thus, ensuring that the readings are not lost and are continuous. So let's start our project to automate electricity consumption monitoring.



This project is sponsored by PCV and SKU online. The SKU online is a 1 stop electronic component sourcing platform. SKU online provides a complete set of electronic component business services. All you need to do is search for the electronic component with a part number. It has the fastest charger engine. It will show you the list of all available parts.

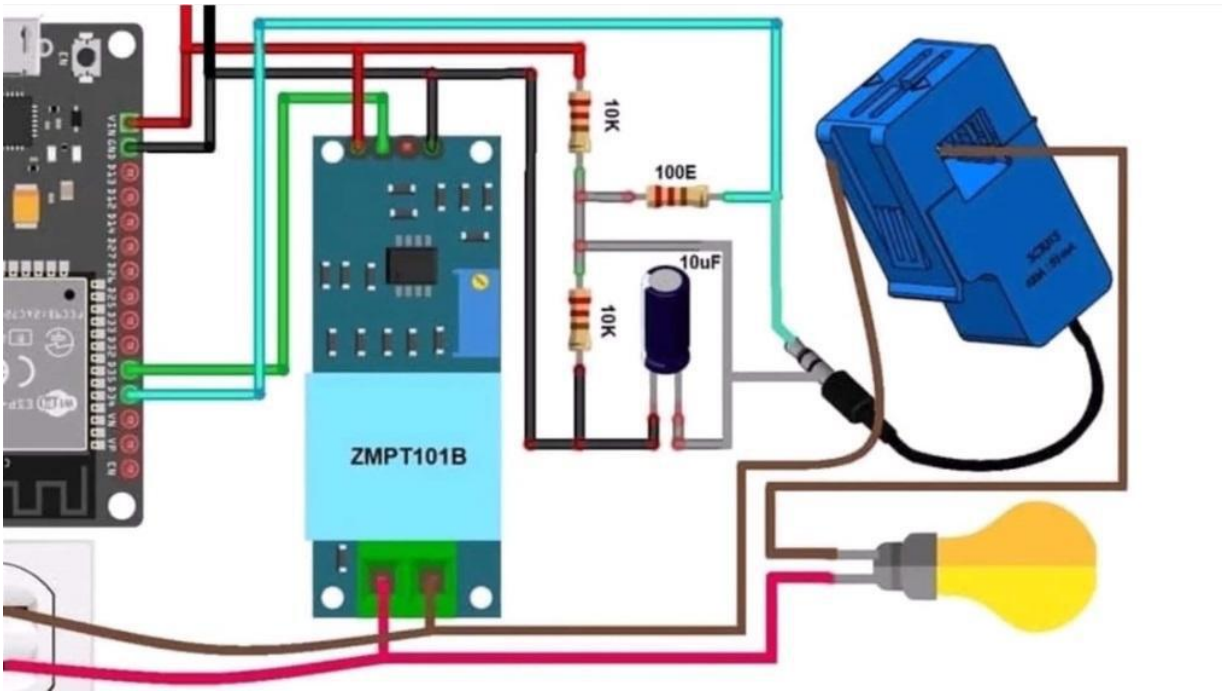
Add the part to your cart, then place the order to get started, check the link in the description. The components required for this project are a USB 32 Wi Fi module or development board. Then our 20 cross 4, I square C LCD display. The LCD is optional for this project. Then we need a SAT013 non-invasive AC current sensor.

These are available with 10 ampere, 30 ampere, and 100 ampere according to your requirements. For the voltage part, we need a certain pt101v single phase voltage transformer module. Some registers with 10k100 home values or 10 microfarad capacitors. Ball for any load, which can consume power. For the valve, I will be using a holder.

For assembly, I will use this bare board. Let's have a look at the circuit for this project. We are using analog pins 3435 to measure the current and voltage parameters. The 20 cars for LCD are connected to the i square c pins of ESP32. Registers are used here for a voltage divider network for the current center.

The wires carrying SA voltage can be connected to the voltage sensor, load, and power using this image. Current center is a clamp type of sensor where you need to plane live wires. I have also designed this schematic using easy ADA software. Then the schematic is converted into PCV. You can download the Java files and order the PCV for easy access and portability of the device.

Alright. For testing, I used BRAIT Virtual assembly. Using the jumper wires, I established the connection between registers, capacitors, LCD, and sensors. Finally, I connected the load as well. A load and the SA power supply can be directly connected and is screwed to the voltage transformer this way.



And for the current sensor part, Just ensure the live wire between the clamp meter. Alright. The hardware part is done now. Now let us quickly set up the link app. For that, go to the bank website and sign in using your email ID and password.

From the web dashboard, create full resets cards. The 4 widgets are here to display the value of VRMS, IRMS, power, and kilowatt hour. Do the settings as per the instruction here. Finally, the webcast would look something like this and it's ready to receive the smart energy meter data from ESP 32. Apart from the web dashboard, you can also set up your mobile app dashboard. You can download and install the Blink application from Google Play Store.

iOS users can download it from the App Store. Let us look at the code part now. 1st, we include a library for LCT display and EP REM. The emon lip is an energy meter that can measure current voltage and power. This is a blank library for a s v 32.

These are the calibration factors to read the current voltage data correctly. From this line, change the WiFi SSID, password, and blink authentication code. Under the my timer event function, We calculate

the value of current voltage power and total power consumed. All the values are displayed in a 20 cross 4 LCD display. Similarly, using the AB REM update function, we are updating 7 the current and voltage parameters to AB REM memory.

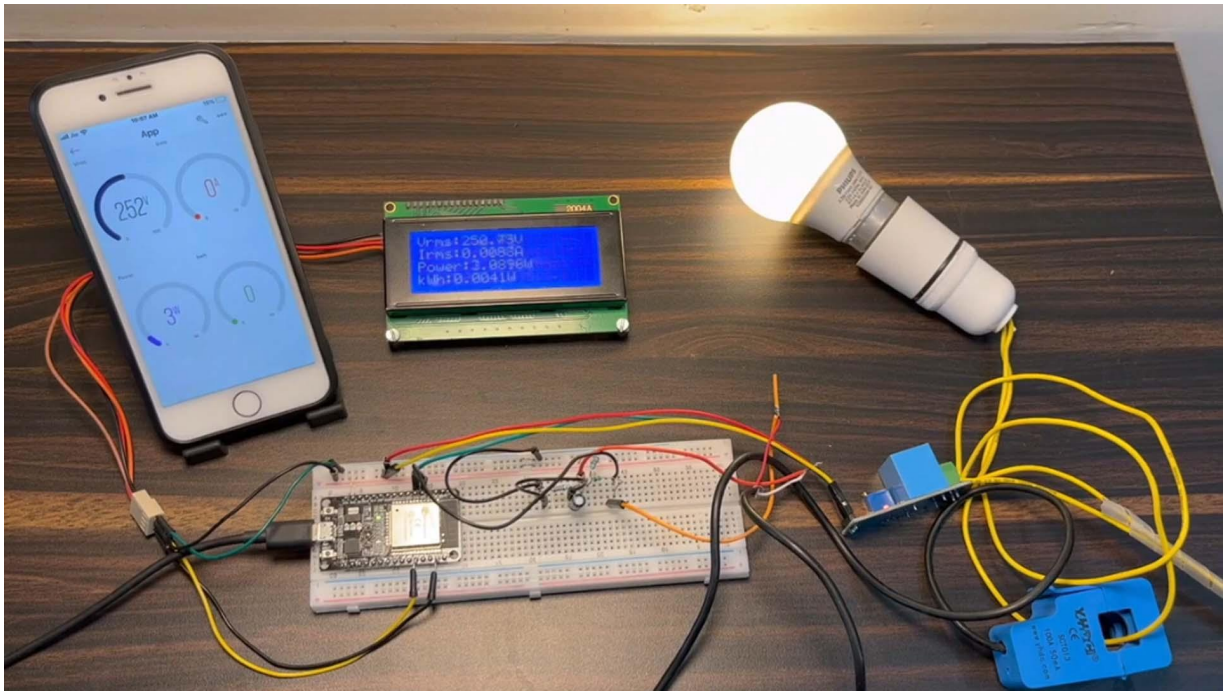
```
Final_code §
#include <EEPROM.h>
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

EnergyMonitor emon;
#define vCalibration 83.3
#define currCalibration 0.50

BlynkTimer timer;
char auth[] = "nrexz-lNuB963age87Dlm7GvtmGure2y";
char ssid[] = "prateeksingh";
char pass[] = "kumar@12345";

float kWh = 0;
unsigned long lastmillis = millis();

void myTimerEvent()
{
  emon.calcVI(20, 2000);
  kWh = kWh + emon.apparentPower * (millis() - lastmillis) / 3600000000.0;
  yield();
  Serial.print("Vrms: ");
  Serial.print(emon.Vrms, 2);
  Serial.print("V");
```

This is useful in case there is a power outage or The device is sitting down. It sets the previous value. Using the Blink virtual white function, we are updating the value to the Blink server. In the setup part, we initialized serial begin, LCD, EV, REM, and blink, and also shine the pins for the current and voltage sensor. In the loop part, blink run and timer run loop continually.

Upload this code to the ESP32 board. Now, let's see the demo. The load is connected to the circuit. Therefore, there is some power consumption. The LCD displays the current voltage power and total power consumption in kilowatt hours.

Similarly, these values are also upgraded in the Blink dashboard. You can access the Blink dashboard from any part of the world. To check the EPRM functionality, we can disconnect the power. Even after the power is disconnected and then connected back, the LCD

will show the value from the previous value, which means there is no loss of data. In conclusion, the deployment of an IOT based smart electricity energy meter using ASV32 and Blink 2.0 will bring about a revolution in the monitoring and measurement of electricity consumption. The IT based solution eliminates manual metal readings, saving time and money.



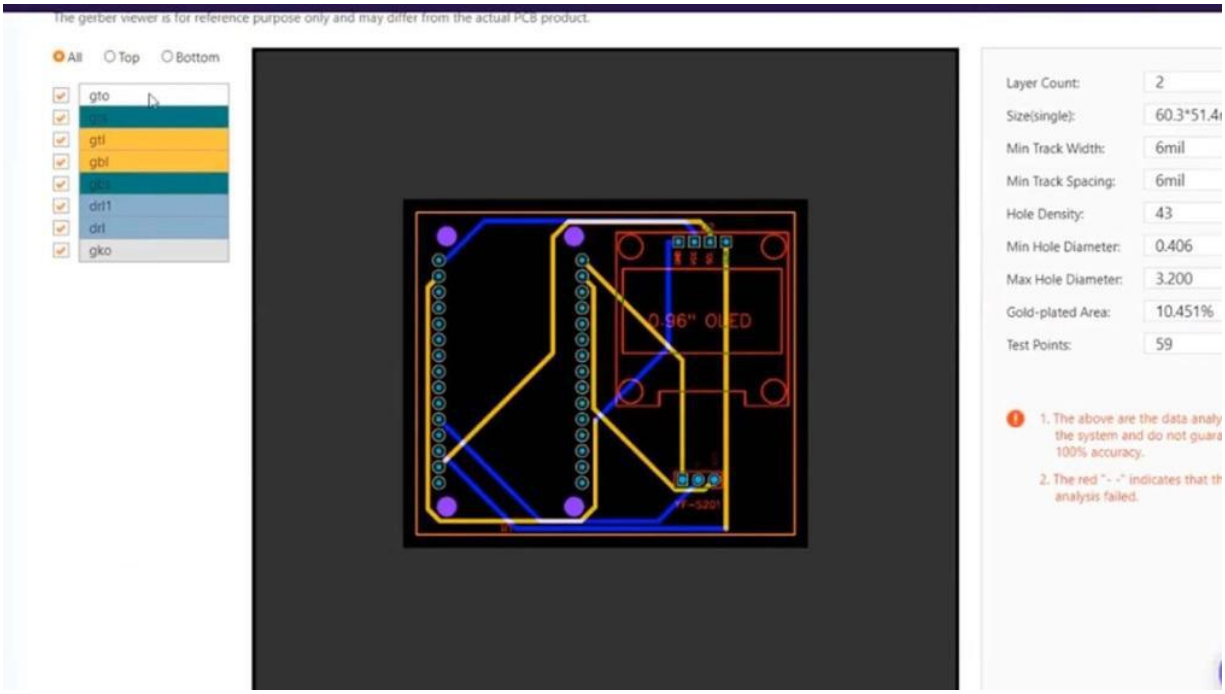
This project presents an opportunity to automate electricity consumption monitoring and make it a more streamlined experience. Alright. That's all from the project part today. The complete project details, including device information, purchase link source code and written guide can be followed in the how to electronics website article.

IOT WATER FLOW METER USING NODEMCU ESP8266

We'll design a water flow meter using the sensor. We'll also measure the water flow rate and volume using the same sensor. This can be done by interfacing Waterproof sensor with nodemcu esp8266 2 early board.

We'll use the I to see an OLED display that will display the instantaneous floor rate and total burn of water passed to the water pipe. This tie up system can be used in water management systems to monitor the water consumption as well as leakage. The flow rate and volume data can be sent to any online server like THINK server using the peak server. We can monitor the water flow rate and volume remotely from any part of the world. This project requires a custom PCB. So I have designed a custom PCB using E GEDA online PCB reading tool.

So you can see this is the 3 d view of the PCB. This is the back view, and this one is the front view. So I have uploaded the driver file in the web chat article. You can download it from there. So after downloading the driver file, you can visit next pcb.com.



Currently, they are giving 10% off on any PCB and PCB assembly services. You can get full air, PCBR, $12. All you need is installed. So just click here. Upload your Gerber file.

So I uploaded the Gerber file for the water flow meter. After uploading the Gerber file, you can view the Gerber file from the Gerber viewer. So this is the top view, bottom view. you can select the quantity, then the board color, select the country of the shipment, and then it to cart. After that, your order will be submitted. Now let's begin with the project.

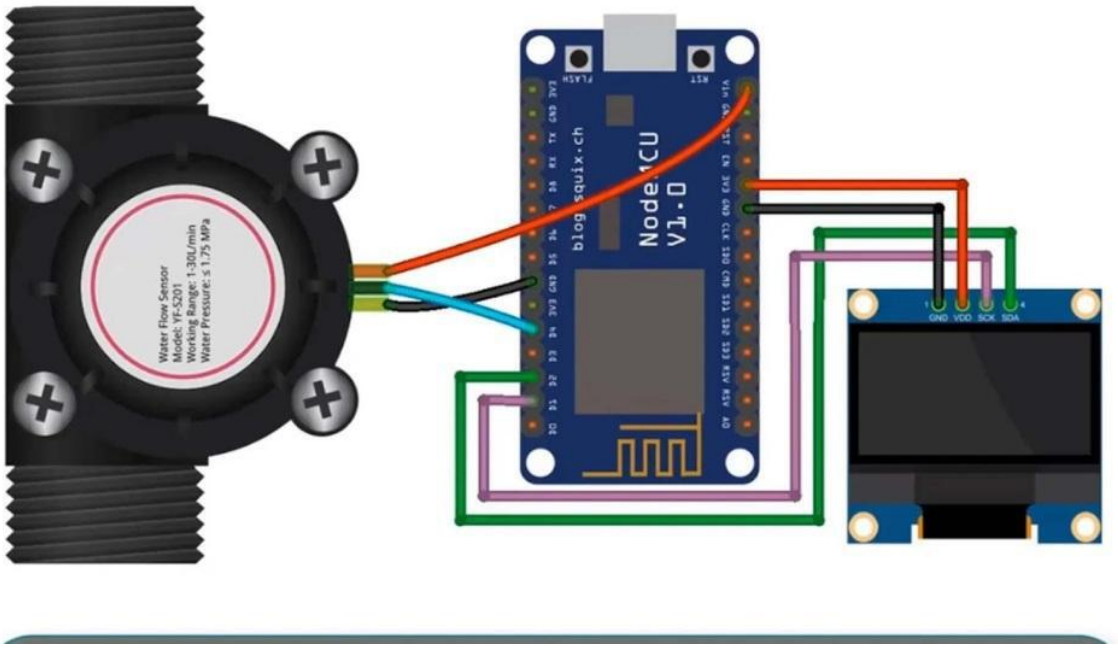
So this is the YFS 201 water flow sensor that I recently purchased from Amazon. You can measure the water flow date up to 30 liters per minute using the sensor. It can get the pressure reading up to 1.75 megapascal. So This is the front side and the backside of the sensor. 1 is the inlet for the water pipe, and another is the outlet for the water pipe.

YF-S201 Water Flow Sensor

So you can see there is a direction indicated that the water should flow from this part to this part that is from inlet bulb to outlet bulb. The sensor has 3 pins that are BCC that work at high volt, ground, and a digital output pin that generally measures the pulse. Now when you disassemble this sensor, you'll get a wheel-like structure, and there is also a hall effect sensor. Now this hall effect sensor measures the output pulse and this output pulse varies according to the flow of water. So just insert pipe on both the end and the inlet end and also on the outlet end like this.

Now let us see the circuit diagram. So just connect an output into G PRO 2. That is the default of node MCU. We are using an ITC OLED display. So connect it or I to seep into d2ndone.

That is SDN SEL now. Let us see how we have assembled the circuit on a breadboard. So this is how I assemble the circuit on bed, but you can just use the ECB that I explained earlier. Initially, when the motor is not turned on, the will be displayed at 0, and the quantity will also be displayed at 0. As you can see, the r is the date and the b is the volume.



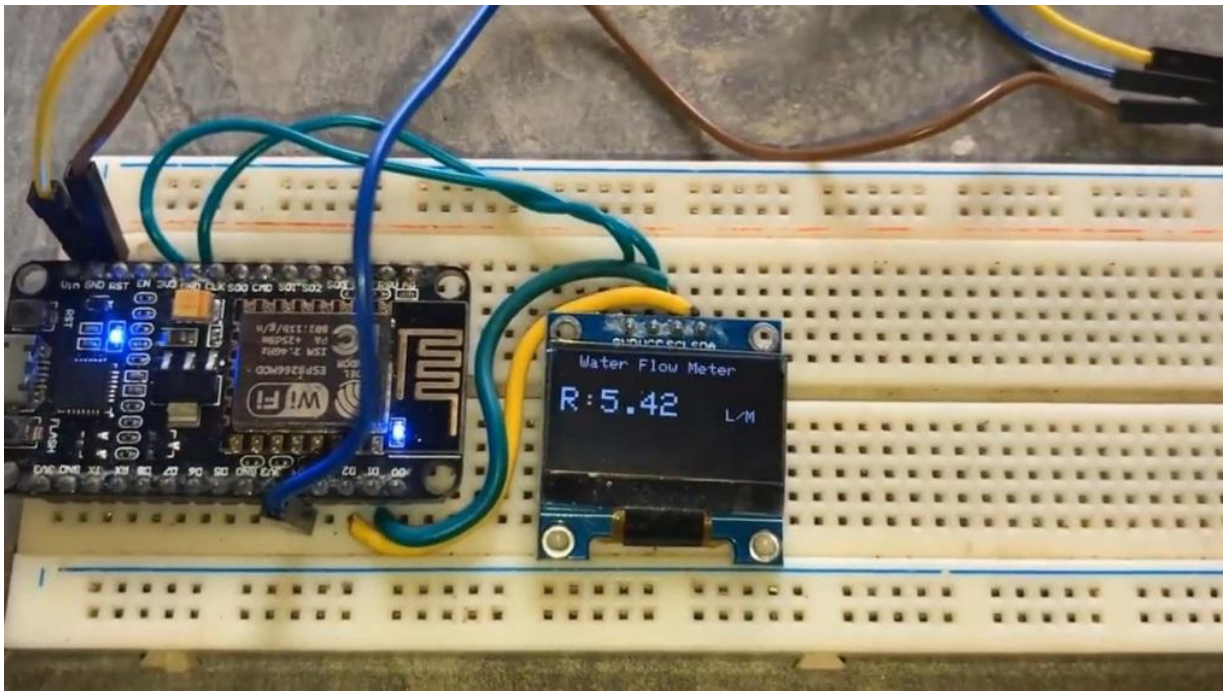
When you turn on the motor, the water will start flowing from the inlet valve to the outlet valve. Now here you can see the rate is almost 5 to 7 liters per minute and volume is increasing and increasing. So you can see here on the serial monitor, the product is given and also the liquid quantity is given. It is increasing. Now it has become almost 2.8 to 3 liters.

So this is how you can measure the instantaneous water flow rate and the total water flowed from the pipe. The chloride depends upon the cross section area of the pipe and also the velocity. So simply you'd know by knowing the velocity, you can determine the flow rate as well as the volume since the cross section area is constant. Now you can go to the THINK server and here you can see the water product data and the volume in numeric order is given. Now this data is logged after the interval of 15 seconds.

You can use the MQTT protocol or just use a blank app for better regions. The process of signing and creating an account in THINK server is given in the website article. So you can see the OLED display displaying the water fluid and volume instantaneously. So

now let's see the programming part. We need the tool library for OLED display.

This parameter is related to the OLED display library. You can get in from the link in the description Change the API key. For that, you need to go to your THINK account and click on the API key, copy the right API key and then paste it on the ID note code. Okay. Similarly, you need to change the Wi Fi access ID and password.



We have defined the LED PIN as 14 for Nordance, you, and sensor pin as 2, we have defined so many parameters including the milliseconds. We're initializing the OLED display and assigning every variable as a 0. We are using ISL function. That is intra service routing for measuring the delay. And using the milliseconds function, we are comparing the flow rate.

This formula will just give you the flow rate in milliliters and liters, and all these parameters are now displayed in an OLED display and serial monitor using this display command. And then this line will simply send the water flow date and volume to the fixed server. All

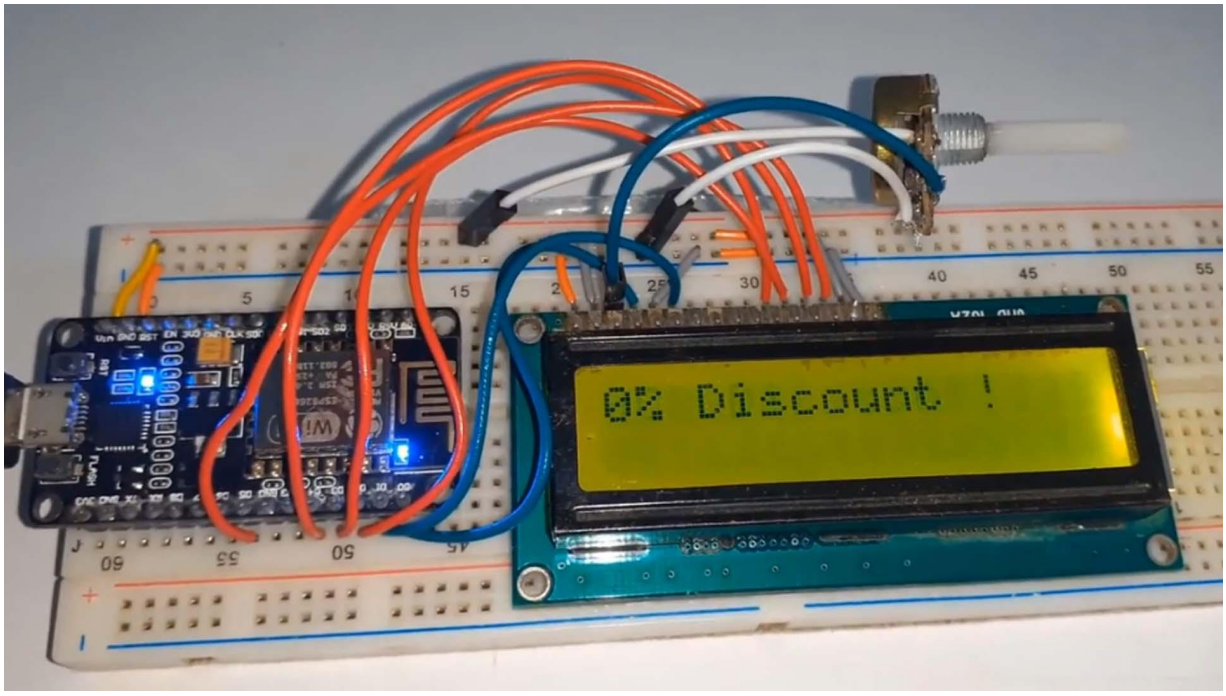
you need is select the right board and then select the right port and upload code to the ESP 8266 board.

IOT WEB CONTROLLED SMART NOTICE BOARD USING NODEMCU ESP8266

In this DIY IoT project, we will make an IoT based smart web controlled notice for using nodemcuesp8266 and LCD display. We will create a local web server for demonstration, And then using any web browser, we will enter the message, whatever we want to display. On sending the message, the LCD screen will display the notice. Probably the best alternative for a traditional notice board system.

The official sponsor of this project is the PCB. Next, PCV is 1 of the largest PCV prototype enterprises in China. They offer high quality multiple application PCVs at a very reasonable price. They're a high-tech manufacturer specializing in quick PCB prototype and small batch PCB production. They have large scale assembly units with advanced equipment , strict management and superior quality.

If it's your first order, you will get a discount. The discount is given in the description. But now let's get started with the project. This project deals with a wireless notice board. The main objective of the project is to develop a wireless notice board that displays messages sent from the web server.

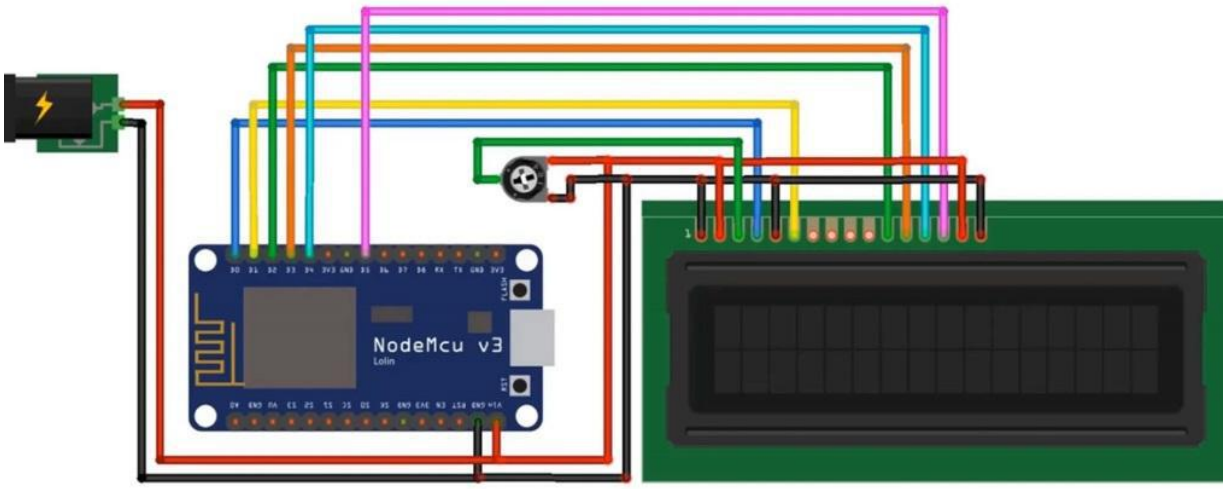


When a user sends a message, it is received by a wifi module through a local web server. The display connected to a service should continuously listen for the incoming messages from the user process it and display it on the LCD screen. it should be updated every time the user sends new information. So here is the circuit diagram for the project. We just need node MCU board 16 cross to LCD display and a 10 k potentiometer.

The connections are exactly the same as shown here. pin 123 of LCD is grounded and pinned to and 16 is supplied with 5 volts VCC. 10 k potentIometer is attached to pin 3 of LCD to adjust the contrast. Rest of the LCD pins are connected to node MCU GPI pins. Now, let us see the programming.

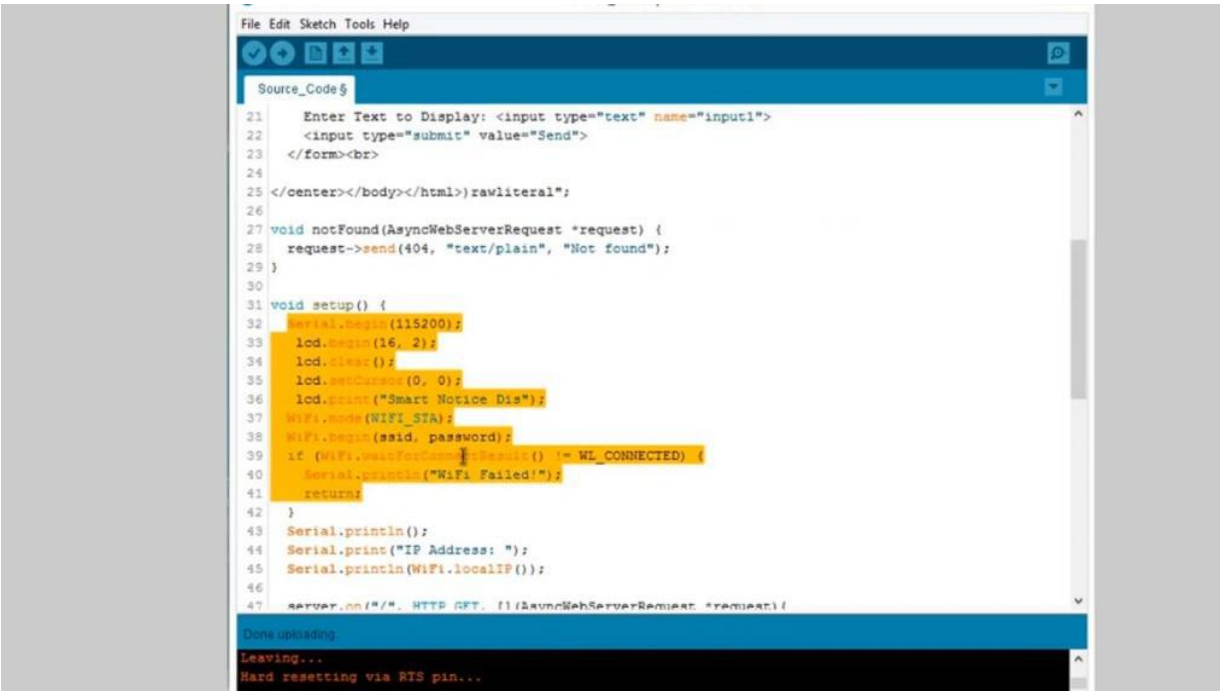
So we need a few libraries here. ESPN 2, double 6, WiFi library to connect to WiFi. LCD Library to interface LCD display with node MCU. We need the most important library that is the SPA Sync TCP Library and ESPAsync web server. Download these libraries from the link in the description.

Circuit Diagram & Connections



These two libraries are used for creating local web server, then we declare all the required pins for LCD and make an instance to use in the program. SSID and password is used for connecting to WiFi network. These lines are used to make a simple HTML page to enter. Here, you can enter the message and hit the send button to send the message on lCD. The send message is stored in memory of node MCU.

This command is used to send the web page with input fields to the client computer. This command will send a get request and then print the received message on the LCD screen. Under the loop function, we print the scrolling value of the text without a break. Now, select the right board and right port. And then upload the code to the node MCU board.



Now, you can open the serial monitor. The serial monitor will display the local IP address after getting connected to wifi. Copy the IP address and paste it in your web browser and hit enter. It will display the web page. Now, you can enter the message here and send it by clicking on the send button.

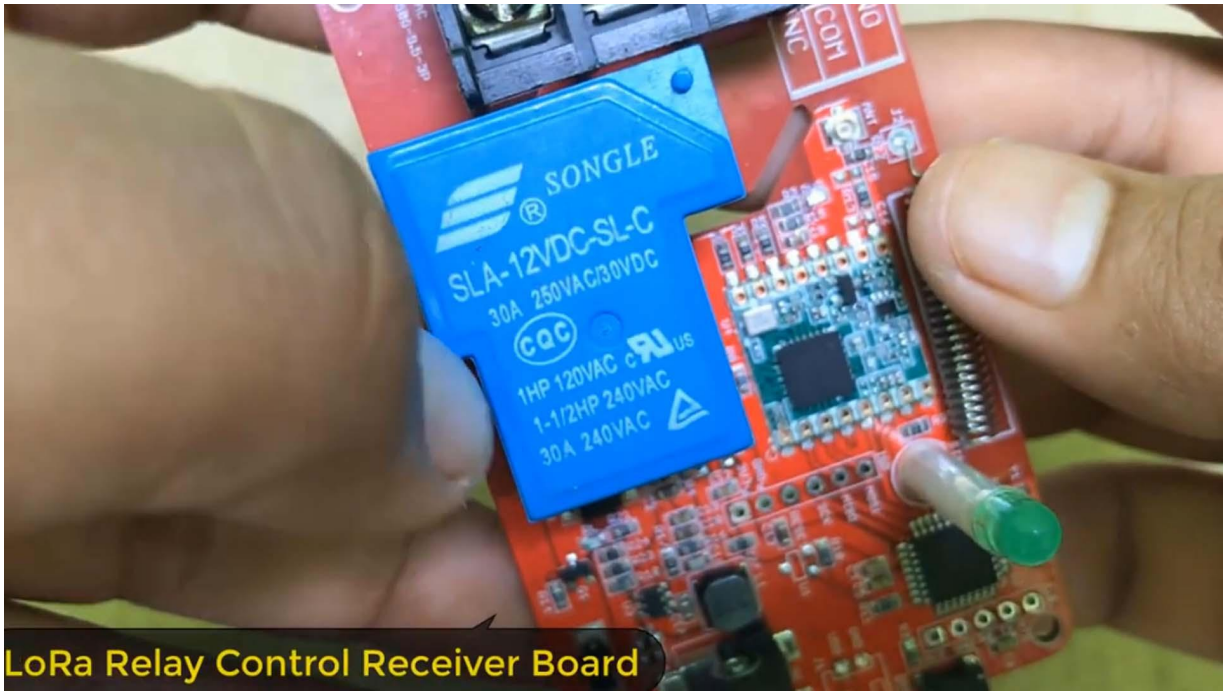
Once the message is sent, the serial monitor will display whatever is sent. You can now see the same message on the LCD screen. So this project is one of the best alternatives for traditional message boards.

LORA BASED REMOTE APPLIANCES CONTROL PROJECT

Using this Control System, you can turn on and off the home appliances using the LoRa remotely. This system can also be used to turn on irrigation projects like a water pump or motors. You can also use this system to turn on the server or turn on the any appliances remotely from the distance of 4 to 5 kilometer.

The board used in this project is purchased from makers.com. So this is the Lora delay. 30 MPL. That I purchased for it, $15.90. So you can visit this link and do more about this board.

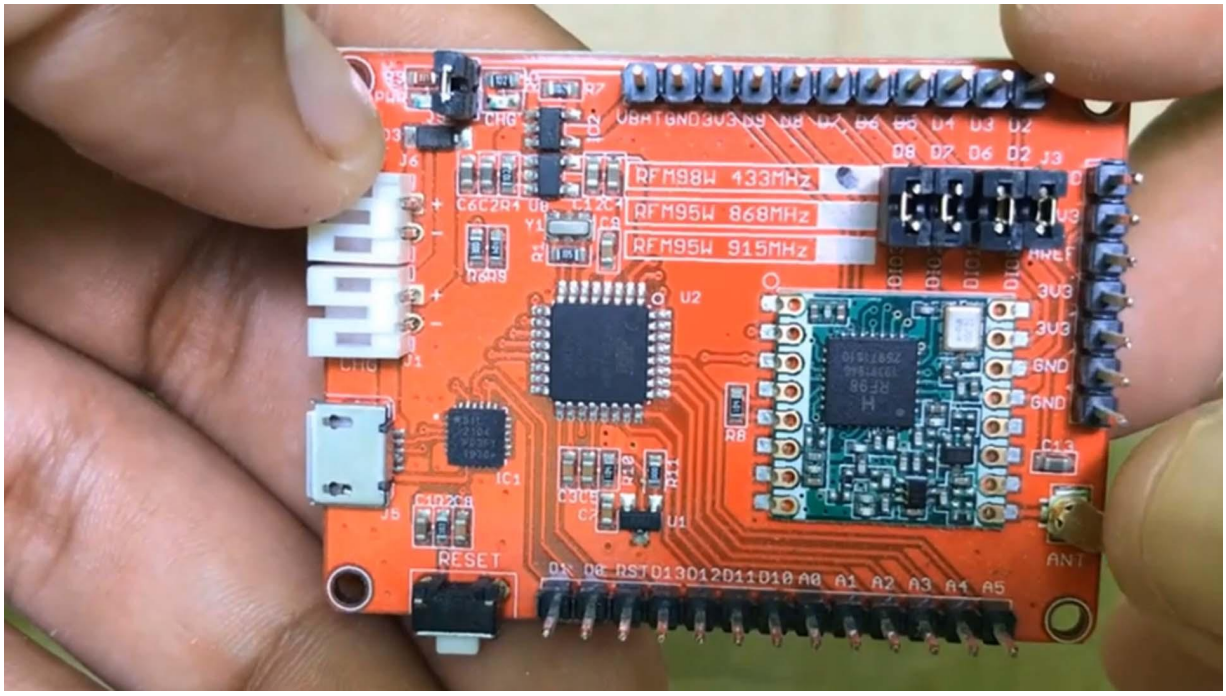
Now, you can make a purchase. Now, along with this board, I have a transmitter board as well. So the transmitter board is called Marino LoRa radio. The board cost around $18.90. So you can purchase the board from the same link and to learn more about this board, you can visit this description part.



So in this project, we'll be discussing this as well as the transmitter part. So first, let's learn about the So the relay module is operated at 2 50 board with a maximum current rating of 30 MPL. And the LoRa used here is RFM 98. That is basically for a region. For Europe and America, you will be using RFM 95.

So it has the output terminal like n o, com, and n c. From here, you can use the 12 volt to impair power supply to power the model. It has a 8 megacity microcontroller with ADNOCromine bootloader uploaded on it. and then here is the external ISP for uploading the code. You can press the reset button while uploading the code or resetting the whole device.

And then on the back side, you can see the frequency selection along with the model that is RFM 98 and 95, best open 43868 and 15 megahertz frequency. Now this module requires a transmitter. So here is the transmitter board. The transmitter board has 8 meg capacity to 8 chromini board and RFM98. There are input and output pins from d02d13andag02a5, the same code can be uploaded using USB to TTL converter or by micro USB.



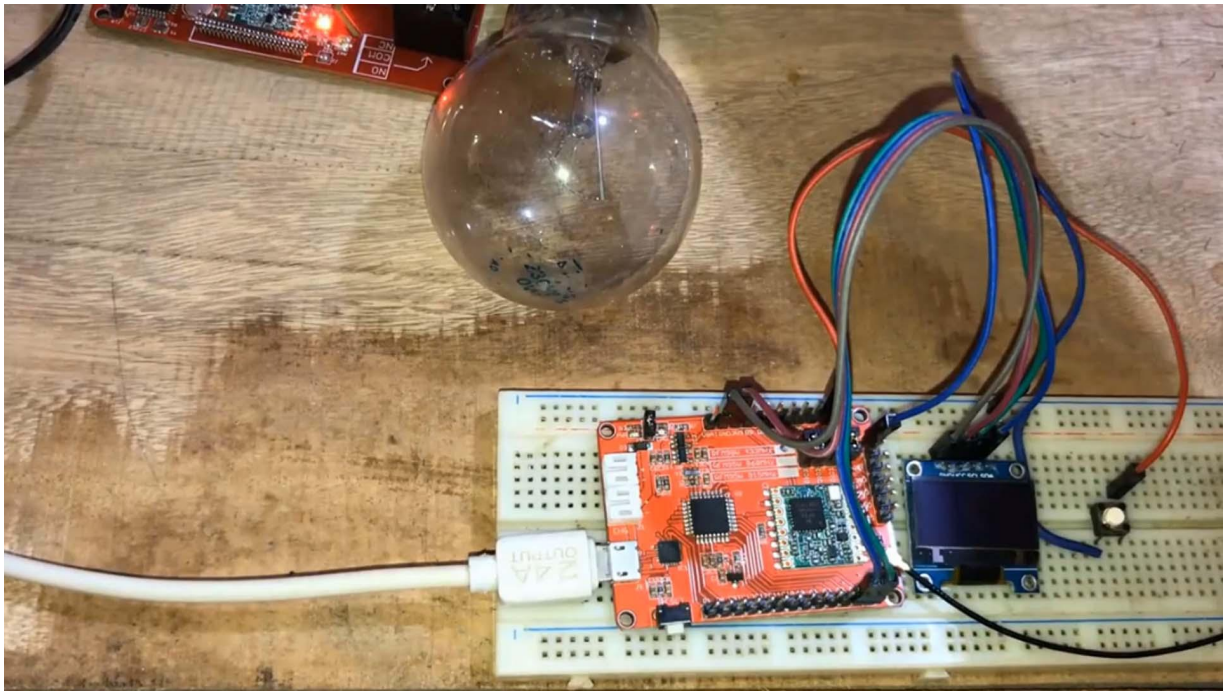
It has a 2.1 for battery and 1 for charger. On the back side, you can see the name of the company that manufactures this port. This is the antenna that receives the signal remotely. Now using these 2 modules, we'll make a communication system to control a bulb. So here is a circuit diagram.

The relay model has a bulb, a test, and on the transmitter side, we have an OLED display and a push button pin. So here is a transmitter part This is the LoRa board with LoRaFM Ninety eight OLED display connected to the ITC pin and button connected to the d three pin. Okay. From here, we'll press a reset button and then control the device remotely. That you will be turning on this bulb on and off remotely.

And the status will be displayed on the OLED screen. So make a connection like this. Okay. Now you can connect this to a 230 volt power supply. but first connect the 12 volt adapter to power on the device.

and then, 230 volt supply to power on the relay. On the transmitter side, we'll be powering the device using a micro SD data cable along

with the adapter. So the device is powered on. You can see the initial initialization and okay. Let us control this device.



When you press a push button, the OLED will display the logo and the bulb on the relay side turns on. Now, when you press the button again, it will display off and the bulb will turn off. Again, place to return on. And after pressing again, it will turn up. So in this way, you can control the appliances remotely just by placing a single push button.

You can also connect multiple relays at the receiver so that if you want to go make a home appliances project, I'll be making such projects in other projects. Okay. So now let's move to the programming part. Now let us see this code part. So we have 2 codes.

1 is for the transmitter and another is for the delay receiver. On the transmitter side, we are using a lower library from Sandeep Ministry and we are also using the library for an OLED display. We are defining some of the parameters for an OLED display. And here, we

are defining the logo height and the width. Now this logo is used to display on and off logos on an OLED screen.

Okay. So we are defining the button PIN s 3 and also we are defining the status of the button pin, whether the pin is pressed 1 or whether the pin is pressed twice. Okay. And using the load a message and, using the string function, we are sending 2 variables. That is the ID.

That is the data number. And, also, we are sending the on and off status. That is we are sending 1, but we are sending 0. Okay. On the receiver side, we are just extracting the same parameter.

That is the string, do we receive the loaded data and the data will be extracted using this function? and the string is then converted into an integer function using 2 int. Okay. And then using this 1 and 0 command, we are controlling the on and off status of the relay as well as the LED. This is the unveiled LED, and this is the relay.

Both had control using the same command. To upload this code, you need to select the Pro Mini board. And then on the Promini part, you need to select 8 megahertz to a 3.3.8 megahertz. on the receiver

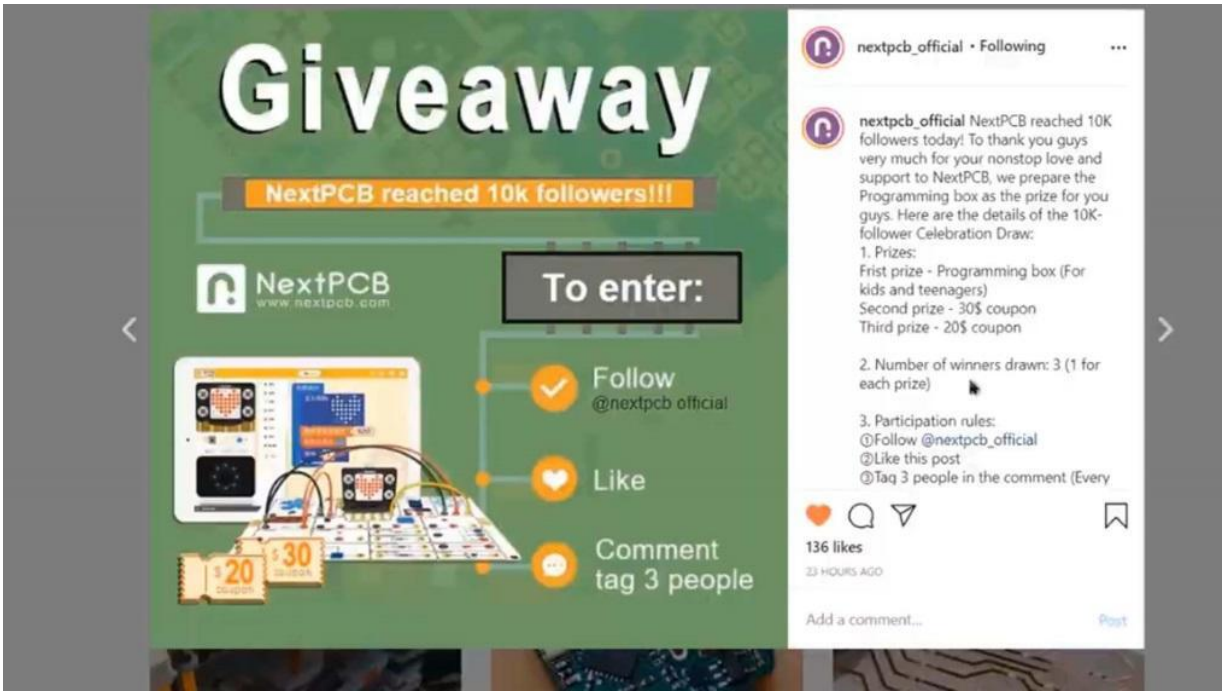
side, the same body is used. 8 meg capacity to add 3.3 volt and 8 megahertz.

LORA BASED WIRELESS WEATHER STATION MONITORING SYSTEM

We will learn how to make a LoRa based wireless weather station, which can be powered by a battery. This is the weather station that I keep at the top of my roof, and I'm using a BME 2h0 sensor along with a B 1750 light sensor and also a rain sensor. Basically, this weather station is monitoring the environment parameters like temperature, humidity, pressure, altitude, dew point, rainfall, and light intensity.

Using the LoRa, I'm able to monitor the data from a few kilometers away. This is the gateway located indoors inside my farmhouse. The gateway is made using Lara, an ESP32 wifi module. The receiver collects the data from the sender and uploads it to the server. I had designed the 3 system altogether, where you can monitor the data.

The first system is the web server. Using the local IP address, you can monitor the weather station data on your web browser, either in PC or in mobile phone. You just need to refresh the base to retrieve the data. The second way to monitor the weather station data is the thanks pick server. The server is free.

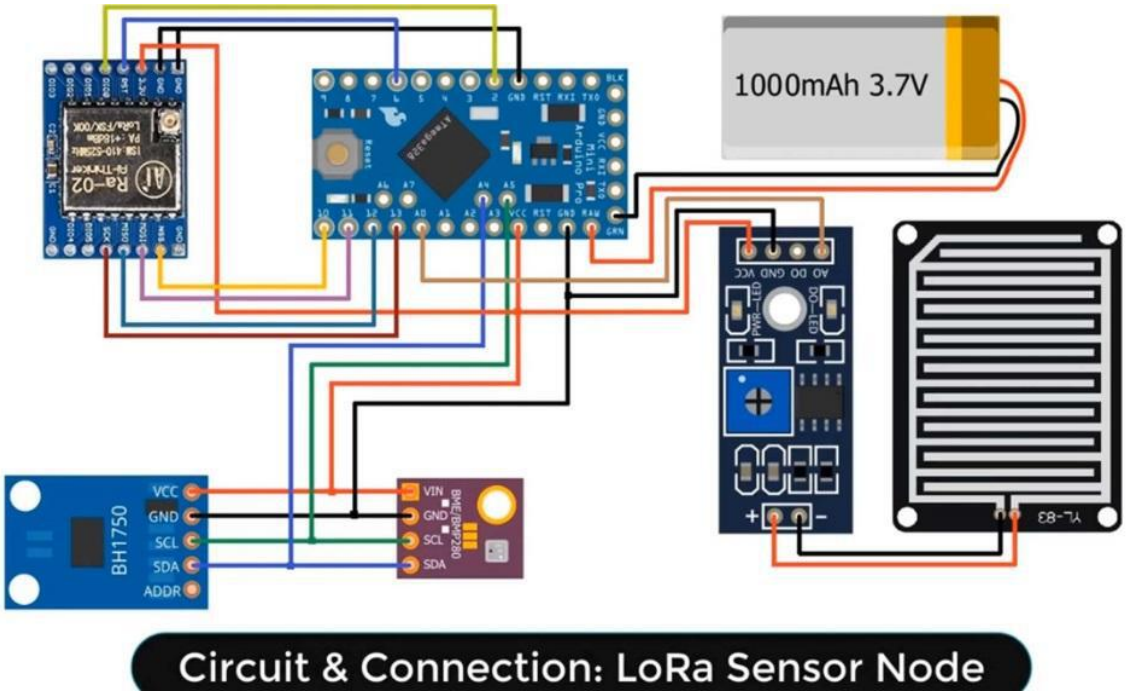


You can simply sign in and configure the dashboard to monitor the data in graphical format as Sonia. This is the best way to collect and store the weather station data online. Well, the 3rd method is by using the Blink application. In this method, the SB32 LoRa gateway connects to the blink server and pushes the data immediately when it receives an interrupt. The parameters like temperature, humidity, rainfall, dew point, like index can be monitored on smartphones.

You can choose any of the meters to monitor the weather data, such as a great wireless weather station, you can do it by yourself. So without any delay, let's get started. This project is sponsored by Next Visit. Next Visit is one of the leading Pacific manufacturer companies in China. Currently, specific is giving free shipping service to all the customers.

All you need to do is clear the order. Fill these details like specific quantity, number of layers, specific size and all other details. On the PCV assembly part, select quantity, number of PCV, quantity of through holes, Number of pets, x-ray testing, and go to place the order. Now, you can see the shipping chart is 0. Isn't it great?

Now, you can place the order by uploading the Gerber files billet materials. And there is one thing you guys need to know. Next is doing a giveaway on his Instagram official account. The prize is a programming box and 20 to 30 coupons for PCV and PCV orders. So visit the link in the project section below.



Follow the next visavis official account and participate in the giveaway. All the instructions are easy and hardly take 1 or 2 minutes. Let's begin with the note circuit first. I selected a low bar Arjuno board. For that, I used Arjuno Pro Mini that is powered by 3.3 volts and runs on 8 megahertz clock frequency.

The BME to 80 can measure temperature, humidity, atmospheric pressure, altitude, and also the dew point. The pH 1750 metric sensor measures light intensity in locks. The sensor measures the value of rainfall in person. The lower module I used is FX 1278 from ai Thinker. The BH1750 and BME 2 80 sensor works on I square zip protocol, and Laura Sx1278 works on SBI protocol.

The device is powered by a 3.7 volt lithium ion battery connected to the Robin of Pro Meaning. On the gateway side, we only have the

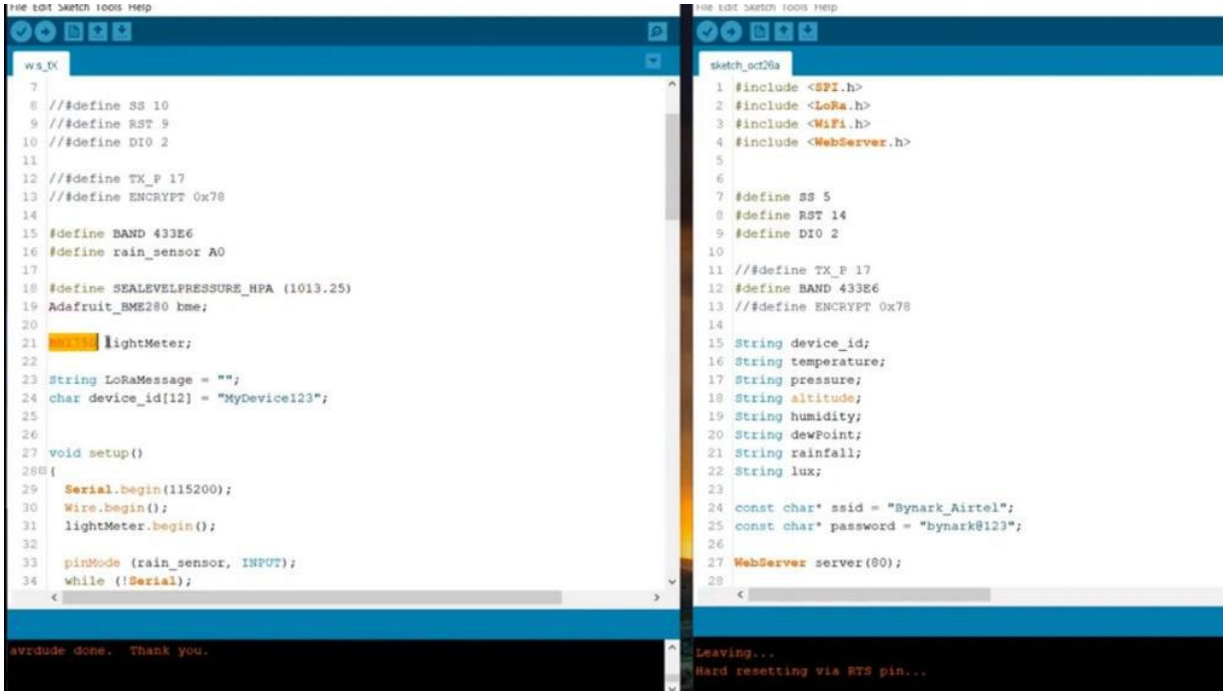
LoRa module S X 1278 and a SP 32 WAFE module. LoRa is going to throw a spare pin, and this unit is power through you and speakable. So you can see here I'm at my farmhouse, which is almost six hundred meters from my house. I've kept the sander note at the roof of my house, The circuit is assembled on Redboard.

You don't need to be panicked. I have designed a custom PCV for this project. You can get the PCV Gorber file and order it online from next week. All sensors can be placed in a small waterproof box, except the rain sensor that needs to be placed outside to monitor the rainfall. The sensor node operates with very small power and putting the device to sleep mode will increase the battery life.

Also removing the necessary voltage regulator and using low power LD or buck converter IC, gateway down the power further. The LoRa module operates on frequency 433 Megahertz but you can start it by 868 or 915 megahertz frequency according to your reason. You can use any other LoRa module with different antennas for the availability in your region. Remember, this is not a waterproof device, so place it inside the waterproof casing. This is the gateway unit based inside my room 600 meters away from the node.

The USB 32 Wi Fi module is connected to the local WAVA network and the modem receives the data from the sensor node. Using the ESP 32 WAFE module, the data is uploaded to the server. Let's see the coding part now. The codings for Laura Center and the web server receiver. Let's just see the sender note go first. We are using a few libraries like LoRa Library and BM2 AR0 and PH 175 0 library.

Then we are defining the LoRa frequency at 433 Megahertz. instance, sir, analog ping is defined here, then we create instances for BME to AR 0 and BH1750 sensor. We are assigning a string and also the device ID for identification. Under the set of functions, we are initializing LoRa, and all other sensors. On the voided function, we are calculating the temperature sir, altitude, and humidity from the library.



```
//#define SS 10
//#define RST 9
//#define DIO 2

//#define TX_P 17
//#define ENCRYPT 0x78

#define BAND 433E6
#define rain_sensor A0

#define SEALEVELPRESSURE_HPA (1013.25)
Adafruit_BME280 bme;

BH1750 lightMeter;

String LoRaMessage = "";
char device_id[12] = "MyDevice123";


void setup()
{
  Serial.begin(115200);
  Wire.begin();
  lightMeter.begin();

  pinMode (rain_sensor, INPUT);
  while (!Serial);
```

```
avrdude done.  Thank you.
```

```
#include <SPI.h>
#include <LoRa.h>
#include <WiFi.h>
#include <WebServer.h>


#define SS 5
#define RST 14
#define DIO 2

//#define TX_P 17
#define BAND 433E6
//#define ENCRYPT 0x78

String device_id;
String temperature;
String pressure;
String altitude;
String humidity;
String dewPoint;
String rainfall;
String lux;

const char* ssid = "Bynark_Airtel";
const char* password = "bynark@123";

WebServer server(80);
```

```
Leaving...
Hard resetting via RTS pin...
```

Similarly, using some mathematical calculations, we are deriving the value of DuPont. For calculating DuPoint, The parameter is defined below the code. Then we are converting the analog value to percentages in order to create the rainfall value using the map function. Check your minimum and maximum analog value for the Ring sensor. It may vary in your case.

Using this formula, we are measuring the light intensity from the BH1750 Library. Then We are printing the data on the serial monitor. Using the lower message string, we are combining all the variables into a single variable separated by some characters And finally using this function, we are transmitting the data packet wirelessly. Now let's just see the gateway code. We are again using the lower library. We are using the web server library to create a web server.

Similarly, we are defining the SS, RST, and DTP. Then We are defining the stream for each individual parameter. Using this wifi access ID and password, we will connect to the wifi network. For handling a web server request, We used a function called server dot on and server dot on not found. Then we started the web server.

If a packet is received, we are parsing the data depending upon the index position of the soft string. Here, the individual parameter is extracted and printed on a serial monitor. Then this today is handled by the server handler client. All the remaining function and HTML codes are given below. The STML code will simply create a web page and display the WAVE station detail page.

Okay. Now you can upload the code to the original Promini and ASP 32 board. Once the code is uploaded, open the 0 for both the transmitter and then a receiver part. If every connection is correct, the center will initialize and start sending the data. On the receiver's side, the receiver will connect to the network and will print the IP address.

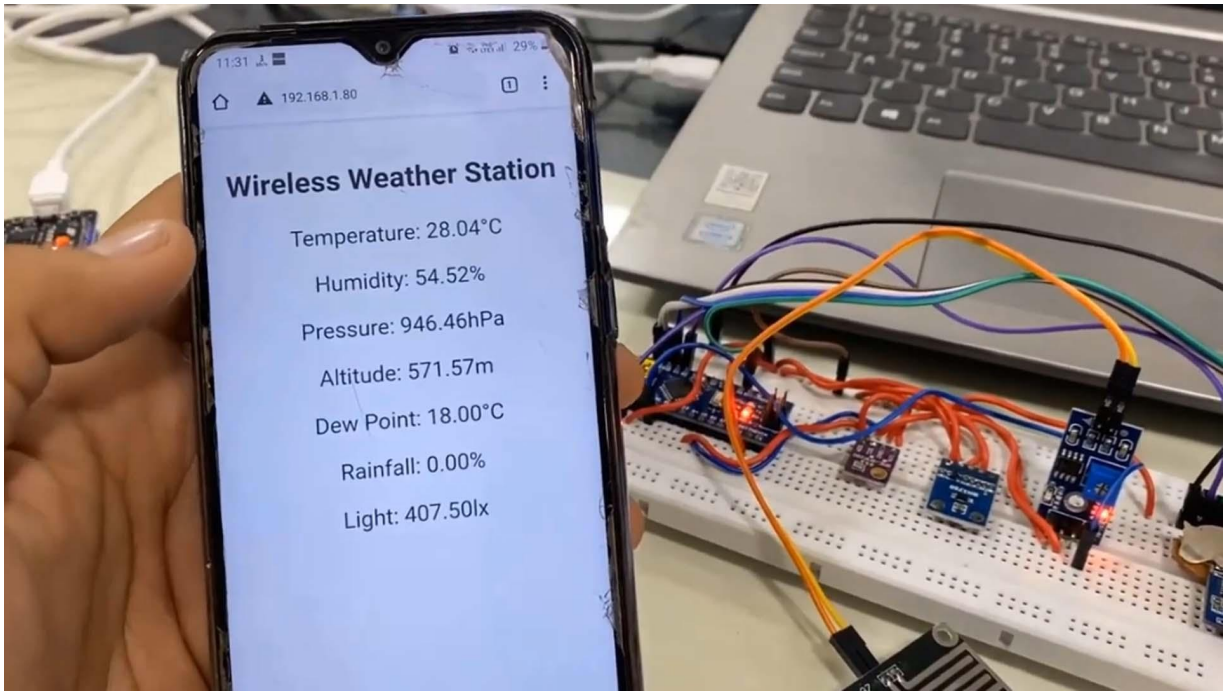
Copy the IP address and paste it on the web browser. The Wither station data will be simply displayed in the web browser. You can reload the page to automatically refresh 8 or simply use the ejects function in code. You can refresh data without reloading the base. So that's all about the web server part.

Now let's monitor the weather station data on the server. For that, create an account on ThinkSpace server or simply log in if you have created the account earlier. Then create a new channel with following details like temperature, humidity, pressure, altitude, dew point, rainfall, and light intensity. Then go to the API key and copy the right API key. Now let's see the receiver code now.

In this code, change the API key from here and replace it with your big channel API key. change the Wi Fi access ID and password, rest of the codes are the same. So just to move to this part and from here, we are sending the data to a big server, by defining the fill and string values. That's all from the scoring part. So upload the code to the ESP 32 module.

Now, again, open the serial monitor. If you see the data is sent and received, means your note and gateway both are working fine. Now,

go to the Think Speak PrivateView. Here you will see the data logged in in the graphical format. The data is received after the interval of 15 seconds, so you can increase the delay.



Let's see the Blink part code. In this code, we are using a few Blink libraries. list of the quarter almost the same. We are just assigning the blinker function, and here we are signing the virtual pin to send the data to the blank server. Make sure to change the blink authentication code from your mobile application.

So when the code is uploaded, the USB 32 will connect to the blink server. Now you can check-in your mobile phone. The mobile phone will receive the weather station data. You can again change the delay in the code to receive the data frequently as you want. Well, this is how you can make a wireless project station.

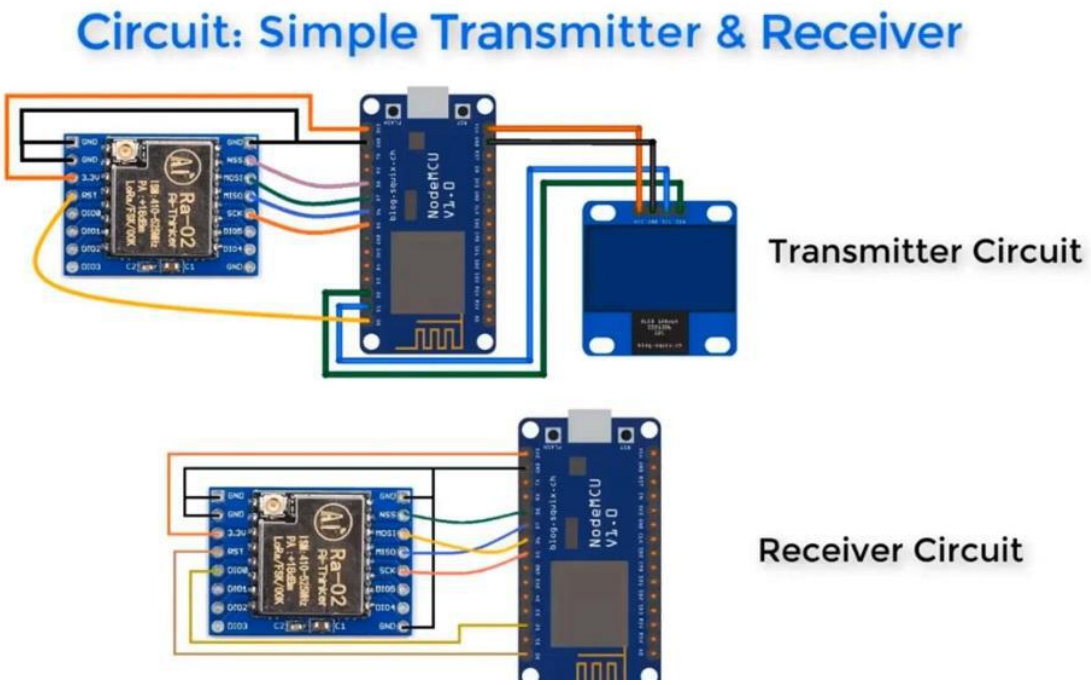
LORA SX1278_76 _ ESP8266 TRANSMITTER RECEIVER

And in today's episode, we'll learn how to make a transmitter and receiver circuit using LoRa module S X 1 to 78th with modem's U ESP 8266 to LE board. will first make a simple circuit in which we'll simply transmit a simple hello wall type packet on the second circuit. We'll interface the DST11 humidity and temperature sensor with node MCU, and we'll transmit the temperature and humidity data wirelessly using LoRa modules. So let's begin with the project.

The sponsor of this project is the PCB. Next, PCB is one of the biggest PCB manufacturer companies in China. They manufactured high quality and reliable PCBs at very low prices. If this is your first order, you will get a huge discount. The discount code is added in the link in the description below.

with modern machinery and better equipment, they also offer assembling and SMT services. So you can just visit their website and order all the services that you want for your complete electronics projects. But now, let's begin with our project. Let us see the first example. So here is the transmitter and receiver circuit.

On the transmitter circuit, we have used SPI communication with nodemcu to interface the LoRa model, and we used ITC communication to interface the OLED display. On the receiver circuit, only there is a connection between node MCU and ESP 8266. 2 l e board. So here is the practical circuit. We have an interface exactly the same as you saw in the circuit diagram on the breadboard.



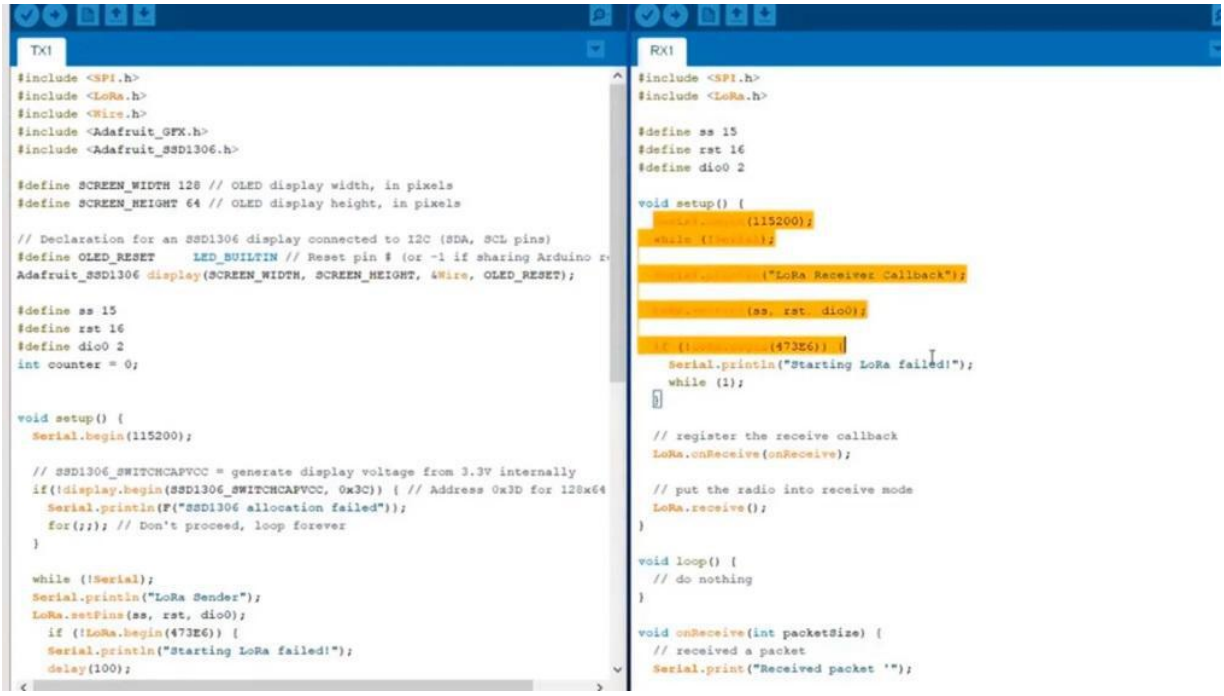
So the module having SSD 1306 OLED displays the transmitter module and the module having no display is the receiver module. So now let us see the simple code, how we have done the communication. So here is the transmitter code, and the other code is the receiver code. So we have used a couple of libraries, like splIFF and wide library. You can get the GFX and SST 1306 library from the link in the description below.

And we have also used the LoRa Dutch library for LoRa. Okay? and we have defined the width and height of the OLED display. This line will reset the OLED display and will clear the OLED. So this is the initialization of the OLED display.

Similarly, we have defined chip select, reset, and input output pins. And we have initialized the counter as a 0. under the void set of function, we have initialized the baud rate 115200. So this line will eat the data from the valid display. That is, it has the I to see address up 0 cross 3 c.

When the power is low, it will initialize by taking the pin as a reference. We have set the frequency of 433 megahertz as this

LoRa model works at 433 megahertz. Okay. If you are using another LoRa module, you can see frequencies like 915 or 868. So this line will clear the OLED display. So under the loop function, we are displaying a simple LoRa transmission as the number of packets that is transmitted. Okay, using LoRa begin packet function and load up print function will send the load up packet to the receiver end.



So counter plus plus will increase the counter after the interval of one second. So a delay of one second is provided. You can change the delay as you like. So now let's see the code on the receiver side. On the receiver side, the library for SPI and load is still the same.

The same chip select, reset, and DI 0 pin is defined, and this all are the same function. as I explained earlier. So this line will register the receipt callback packet. Okay? And Laura, that receipt will make this LoRa module on a receiver mode.

And that void loop function, we have to do nothing, but on void on a receive function, the data received and the packet is read. Once the packet is read, it will display or it will print the packet number on the serial monitor. So upload both of this code to the transmitter and

receiver. And now you can see one of the LoRa modules. There is a transmitter modulated OLED display that has started sending the data packet.

So you can see the packet number is transferred. So whether the packet is being received or not, we need to see this. So for seeing this, you can just open the serial monitor on both the terminals. So on the serial monitor, you can see that data is transferred and the data is received. Okay.

So this is how we can do simple communication. Now let's do a complex communication by sending the sensor data of DST 1. That is temperature. and a humidity sensor. So here is the transmitter and the receiver circuit.

Now while talking about the code, the code is still the same. Only change I have made is that I have defined the DST team. Okay. And I have said how to read the DST data. and the DST temperature and humidity data is displayed while it displays using these commands, and we are sending the load up packet in the form of temperature and humidity.

The receiver code is still the same. There is no need to make any changes on the receiver code. So here is the perfect example. So we can see the transmitter section is displaying the temperature and humidity data. And while rubbing the hand, you can see there are changes in temperature and humidity value.

Now this value needs to be transferred to the receiver end. So we will see this data on the serial monitor. So you can see this monitor is receiving the packet with humidity and temperature. So this is all about today's project.

MQTT WIND WEATHER STATION PROJECT USING GSM

A couple of weeks ago, I got a product from a company called Maker Fabs. This is basically a weather station box for multipurpose applications. Along with this box, I also got an anemometer sensor that can measure the wind speed. Inside the box, there are sensors like THT11 sensor, Bmp280 sensor, buzzer, and some input output ports for connecting external sensors. The external sensors include a PM 2.5 sensor and anemometer.

Also, there are several GPIO ports for connecting whatever you like. There is a dedicated port for charging the device using a solar panel. Since the product is made using a GSM module, along with a 32 bit atmel microcontroller, you can program the device using Audio ID. So in this project, we will see the functionalities of the box and check how the board is designed. We will upload the basic code to read the weather station data on an OLED display.

A button can be pressed here to enable or disable the OLED display. And finally, we will make an MQTT based wind weather station and monitor the data on the MQTT dashboard. The project is a little detailed, so watch it till the end to know about this exciting product. So without wasting any more time, let's get started. This project is sponsored by my favorite PCB manufacturer company called NextPCB.

They offer PCB board and PCB assembly services at the lowest for double price. You can get trial PCB, 2 layer PCB, and 4 layer PCB with free PCB assembling services up to a costly time of 24 hours. There is good news. That is next PCV has acquired Kiki PCV. You can use a Kiki PCV account to log into the next PCV and directly place an order There is another great news.

You can get up to 30% off for the PCB offer and up to 20% off For the PCBA offer, you can check the activity rules to learn more about this. Let's go through the unboxing of the wind weather station box. It measures the outdoor humidity temperature and atmospheric pressure when displayed and transmit the results remotely to the server for real time checking. These are the external ports for connecting animator and PM 2.5 sensor. And here, you can connect the solar panel.

On the other side, there are pins for connecting I Square C or analog sensors. Here is a slide switch to turn the device on. This is a cup type anemometer sensor basically with NPN output. This detects even a slight movement of air and major stowing velocity. The cost of

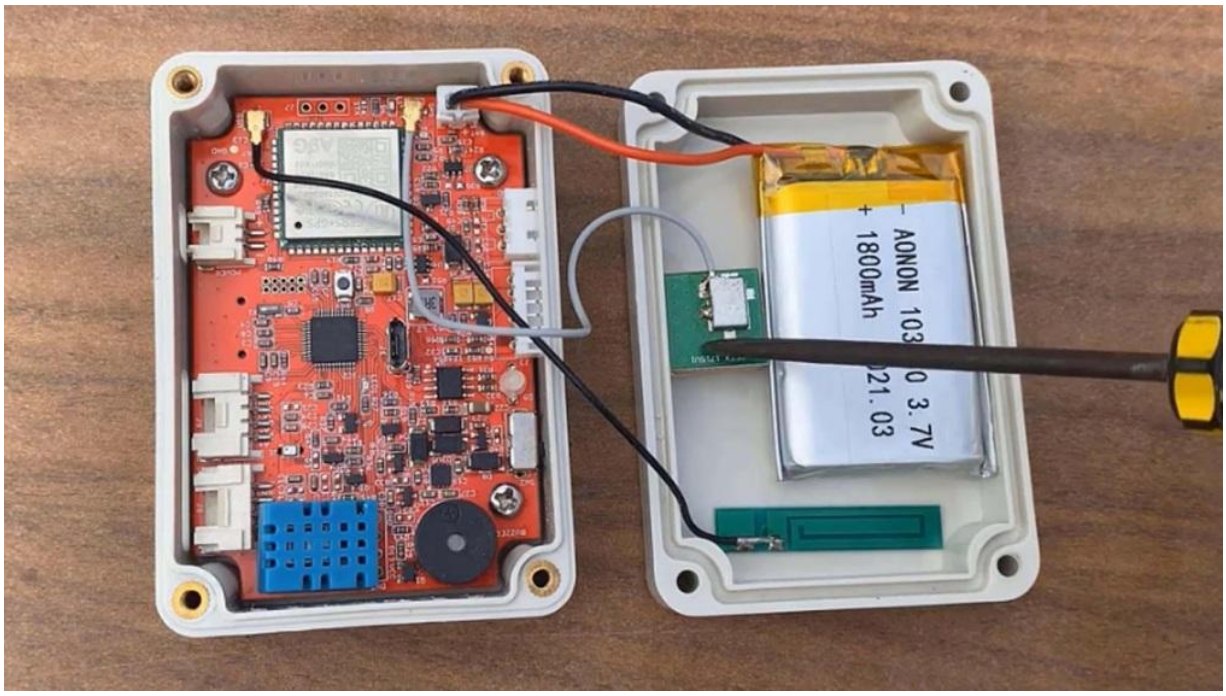
the sensor is a bit higher, but it comes combined with a weather station box.

Unlike the analog output version, this animator output pulse reflects the wind speed more accurately. Besides, This animator is totally waterproof with a stable installation design suitable for output applications. It also has a 4 pin milliliter type probe like another animator. And with this, you will get a connector or expansion cable. You can insert and rotate it to fix this.

On the next end of the cable, there is a three pin connector that fits perfectly on the weather station box. Simply check the three pin port and insert the animator. You can place the box along with the animator at the top of the house or outdoors where you want to measure the weather parameters. Since I'm at home during my vacation, I will place it on the roof of my house and measure the weather parameters of my surroundings. Alright.

Let's open the box and see what's inside it. That is, let's check the PCB board and construction of this system. To open this, I use my screwdriver and remove the 4 screws from 4 corners. So after removing this screw, I opened the box. So here is the complete package inside with a PCV board, few antennas, and a battery.

This is a lithium ion battery with 3.7 volts and 1800 MAH capacity. This is a GPS antenna and this 1, a GSM antenna. The GPS and GSM antennas are connected to the board with a UFL connector. There is a 2 pin removable battery connector. I will remove all these connectors now.



To view the PCB board, we need to remove these 3 screws as well. Let me remove it. So finally, we have taken out the PCB board. Now, let's see this board. This is an A9 G, GSM plus GPS plus GPS module.

The board is low powered and manufactured by an AI thinker. This is a microcontroller called ATSAMD 21z18, which is a 32 bit Armcortex M0 plus microcontroller from microchip. This is a micro USB port for programming and serial application. Then we have a few connectors explained earlier. This is the DST11 humidity and temperature sensor.

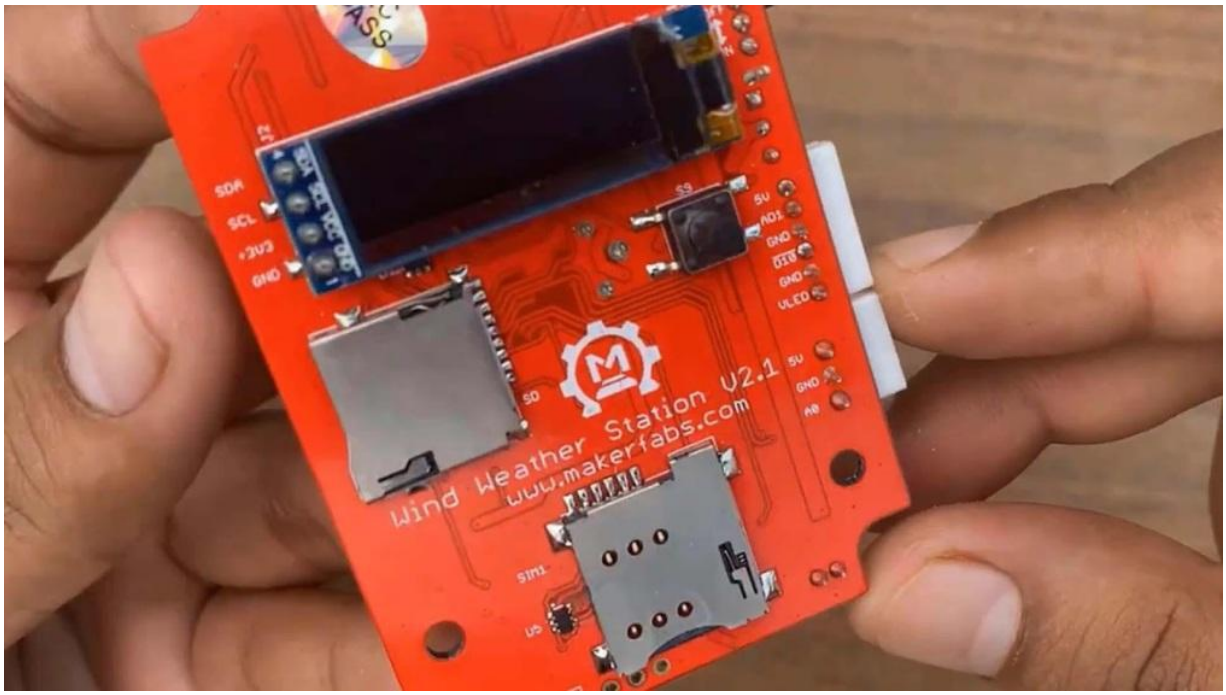
Here is a 5 volt buzzer for indication or some sound application. This is a BMP280 barometric pressure sensor that measures pressure, temperature, and altitude. This is a wrist button for debugging and testing. This LED is for indication the same as a buzzer. You can disable it for lower power consumption.

Again, there are few output ports for connecting sensors. This port is for the solar panel. You can connect a six volt solar panel to charge

the lithium ion battery. On the backside of this board, We have 4 important components. This is a half inch OLED display.

It also has an SD card slot so that you can insert the SD card and use it as a data logger. This is a SIM card tree where you can insert any twosy SIM for GPS applications. Then this is the reset button which has a long header. This button is used for enabling or disabling the OLED display. This is the turn on off switch for powering the device.

You can insert a mini SIM inside the SIM tray. Be careful while inserting the SIM and check the direction of the SIM. This is the PCB designed using Eagle Software. This schematic along with the PCB file can be found in the GitHub link. This is the 3 d view of the PCB file.



The PCB is compact and designed by a very experienced engineering team. The front side and back side look perfect. To purchase this weather station kit, you can check maker fab's official website. This link is given in the description. The combined board will cost you around $45.80.

In case you need an additional PM 2.5 sensor, then you can choose it from here. The price will be $50.70. You can also see the product description and watch some default projects. All the specifications features along with the part list are given here. Let's move to the test code or the default example code first.

Open your original ID and paste this code. This is the test code to check whether the device works or not. The 32 bit ordinal board is not installed here. So first, let's install the ATS AMD board. From the tools, select the board and click on the board manager.

Search for Ardeno 0 board. So here you can see a 32 bit ARM cortex board. You just need to install this board. I've already installed the latest version of this board. Now, from the tools, select the board.

You need to select the ordinal 0 board with a native USB port. And from the port list, select the comport with native USB port. The code requires some of the libraries like the BNP280 Library, Adafruit SST 1306, Adafruit GFX, and also the DST11 library. You can go through the rest of the codes in the website article. As the code is long here, I want to explain everything here.

Alright. Now connect the micro USB cable to the USB port of the PCB and on the other end, connect the cable to the computer. Now click on the upload button, and then the code will be uploaded within a few seconds. Once uploading is done, open the serial monitor. So here you can see the wind display value.

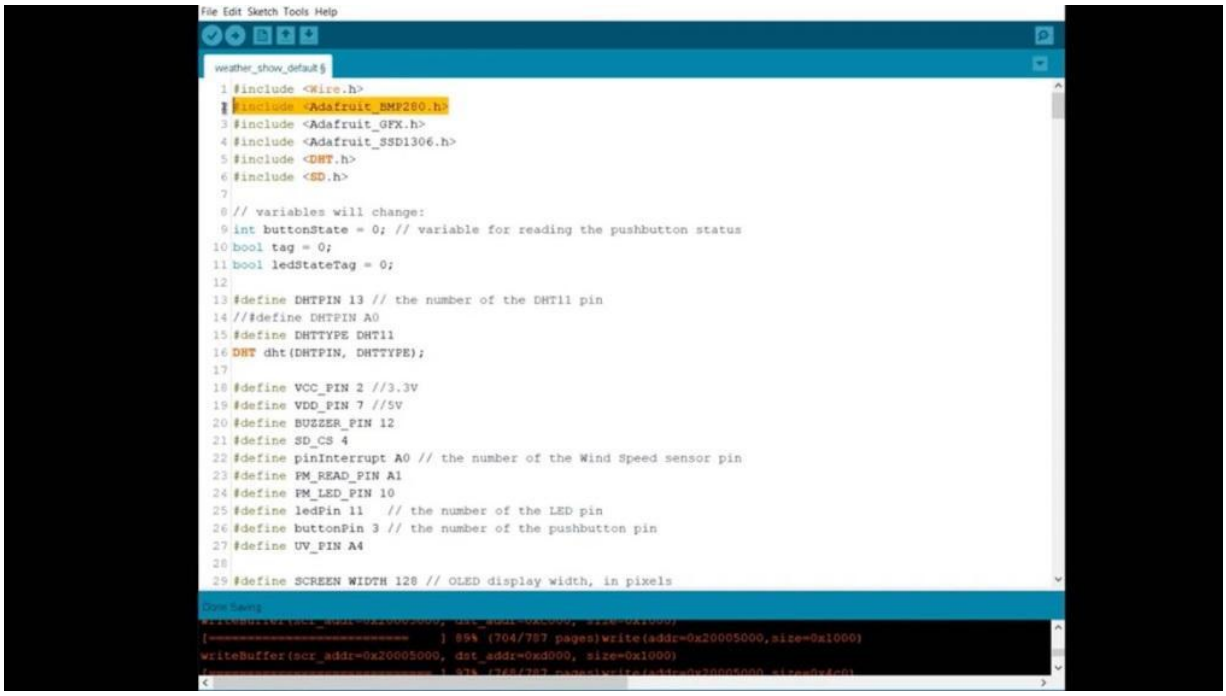
Also, the temperature, pressure, altitude, value from the BNP280 sensor. And from the DSD sensor, we get the displayed value for humidity, temperature, and heat index value. So this means the sensors are working pretty well. Alright. You can place the sensor outdoors or at the roof of your house and see all the parameters getting displayed on an OLED display.

But the display is not clear here, so let me take this device inside and show you the demo. So, alright, the display is clear here. So you can see the humidity value and will display the value here. The more the animator cap rotates, the higher is the wind speed. The OLED also displays all the parameters temperature, UV ray index, atmospheric pressure, and PM 2.5 value.

The PM 2.5 value is 0 as no sensors are connected. One thing is that you can use this external button. This button when pressed will disable the OLED display. This is particularly used for reducing the battery charge as the OLED consumes a little more power. Once the button is clicked, the OLED will be displayed and on the next click, it is enabled.

Let's convert this product into an IoT based product using the GSM GPRS applications. Here is the complete code for that. Only the GSM and GPRS part is added to the project. The low power mode and power or reset key functionalities are added here. The GPS connection is established using the AT command.

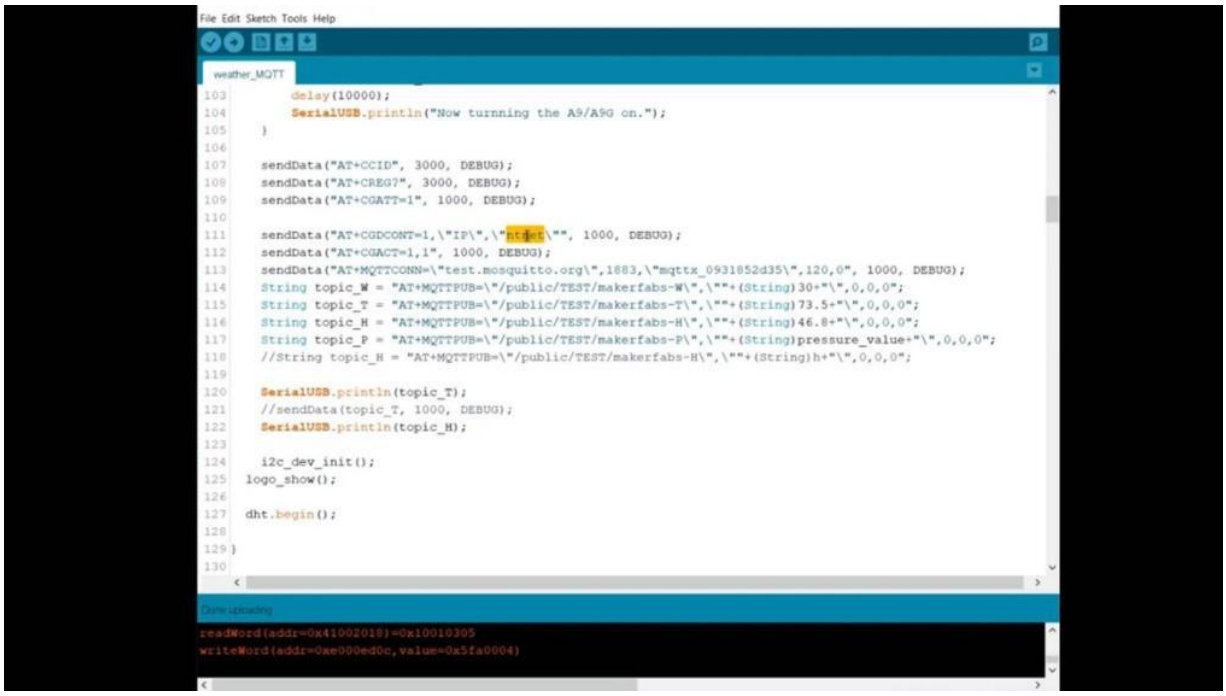
From this line, you need to change the SIM APN. I'm staying in Nepal, so I'm using Nepal Telecom SIM. So my APN is ending. You can check your APN from the service provider. These lines are for establishing an MQTT connection with Mosquito.



We will use Mosquito To monitor the MQTT data, you can go through the rest of the codes. It's very long and requires too much time for understanding. Again, go to the tools and select the right word and poet and then upload the code. After uploading the code, open the serial monitor. So here, you can see that A 9 G board is turned on and SIM is ready for establishing our connection via AT commands.

Alright. The connection has been established. The device is reading all the sensor sensor data and sending it to the MQT dashboard. This time, you can place the device in a remote location or far away. This is an IoT device.

Now, communication is based on the GPS. For the demo, I paste it on my neighbor's house. Now, let's configure the dashboard on the mobile phone now. Install the MQTT client Android app on your Android phone and then open the app. Go to the settings.



Enter the URL as test.moscuro.0iz. In the port, enter 1883 and enter the username as maker fabs dot g. Use the MQTT client and subscribe to the topics which are entered here as shown on the screen. It will receive the temperature, humidity , wind speed and pressure data. Just wait for some time and see how the data is regularly uploaded.

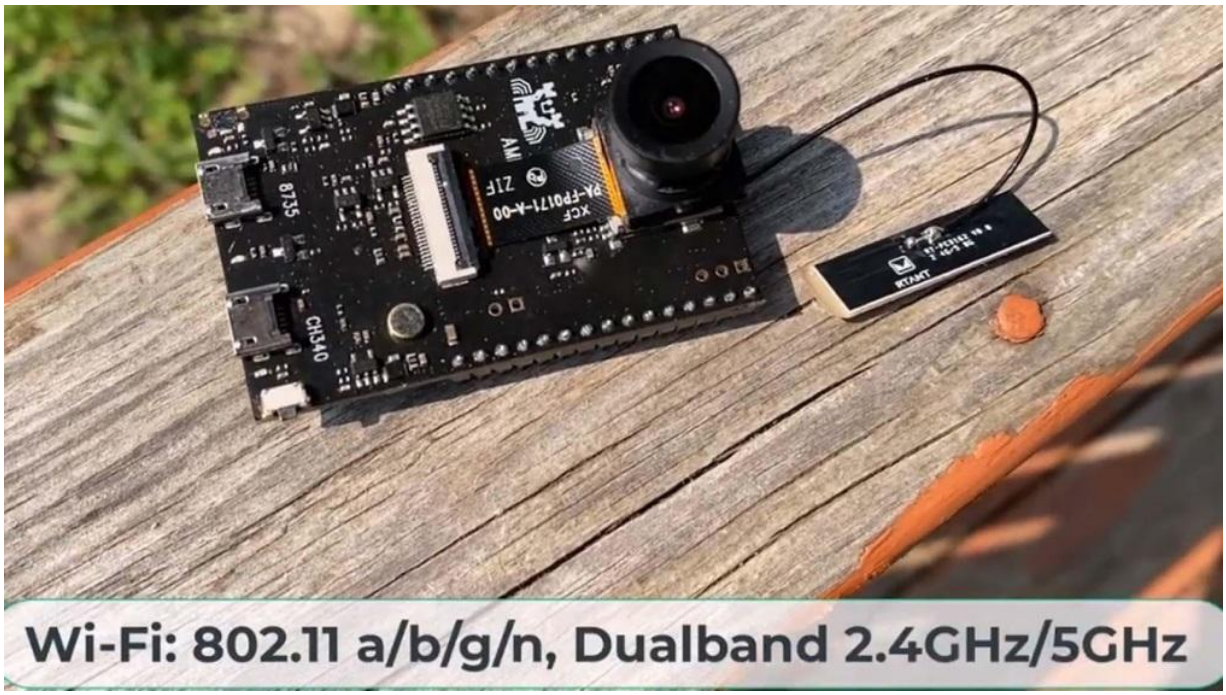
You can check each topic individually. The message comes with the value of a sensor. So, this is how you can build your own MQTT based weather station using GSM in Audio. and monitor the data on the MQTT dashboard.

POWERFUL ALTERNATIVE TO ESP32 CAM

I got a very powerful product from Realtek. This board is a powerful replacement for the ESP 32 camera module. The name of this board is AMB a 2 mini, which is basically an IV AI camera or, you know, developing board. This port is designed for artificial intelligence, machine learning, and neural network applications.

Also, the board has ultra low power consumption, which basically means there is no need to worry about power consumption. This board has an astray camera with a resolution of 1080p. When This board is compared to ESP32 cam. It is wide ahead. ESP32 has a CPU that operates it to 40 megahertz.

whereas this AMB a 2 mini board has a CPU speed of 500 megahertz. ESV data only supports 2.4 gigahertz wifi band, but this port can support 5 gigahertz WiFi. Apart from WiFi, it has a BLE chip that supports Bluetooth low energy 5.1. It also supports MCM embedded TDR2, TDR2L memory up to 128 MB. such a powerful chip.



Right? In this tutorial, we will learn about the board design, print description, features, and capabilities of this board. Then we will set up the Arduino IDE and learn how to program it using the Arduino IDE. We will also undertake some practical exercises with this part, like making an Additive link. project streaming over a web server, object detection, and identification.

and 1080 p project recording. So let's get started and learn more about this port in detail. Welcome back again. First, let's take a look at this port. The Realtek AMB with too many IITA camera boards is a developing tool.

which is designed to streamline the creation of AI network camera applications. This port has a highly integrated Realtek RTL 8735 BDM SOC, which features WLAN and BLE solutions. The microcontroller has armed we attempt 32 bit architecture running at a super fast speed of 500 megahertz. The board is designed for AI ML and neutral network applications, which is done by NPU intelligent engine 8 0.4 T OPS. It has 768 kilobytes dong, 5 to 12 kilobytes RAM and 16 MB Flash memory.

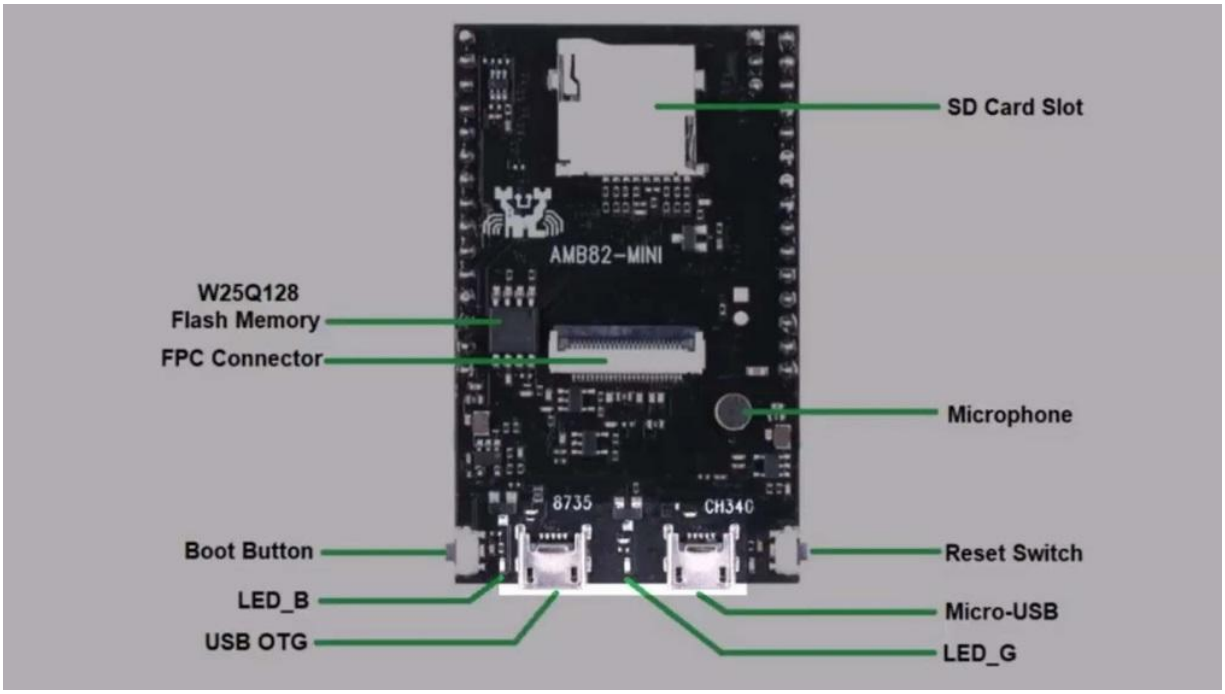
It also supports MCM emitted ddr2ddr3l memory up to 128 MB. It supports dual band wifi networks of 2.4 gigahertz and 5 gigahertz. The BLH chip embedded in it is bluetooth low energy 5.1. The board has intelligent security support. such as hardware cryptographic engine, secure boot trust 1, etcetera.

As you can see, a microphone is here with supports and audio codec like ADC, DSC, and ITS. It can render STR3DNR, WDR Project8, 1080p and 720p resolution with 30 frames per seconds. It has a powerful camera module called ZACF37, which is 1920 cross 1080 full SD CMOS image sensor with white view angle, FUV 130 degree optical lens with a 1920 cross 1080 pixel resolution The camera provides crystal clear image with fine details, making it ideal for a variety of applications, including security, photography, and projectgraphy. To use this camera first, you need to connect the camera to the accuracy terminal of this board. Just pull this connector, then slightly insert the camera module.

Then post the connector again so that camera will perfectly fit. The board needs an antenna for the Wi -Fi network. Here is the UFL connector for the antenna. Connect to the antenna here. On the front side, you can see the SD card slot here.

You can insert an SD card here which can store images and projects. There are 2 push buttons in this port. 1 is the boot button and the other is reset. There are 3 LEDs here. The red LED is a power LED.

2 others are LED B and LED g. There are 2 USB ports as well. They are micro USB and USB OTG. Using the micro USB, we can upload the code to the board or establish a studio communication with the computer. On both sides of the board, there are GPU pins, which also supports I Square C UART SPI PWM ADC.



Through these interfaces, AMPA 2, meaning can connect with external electronic components and sensors. One more thing. I have made a different stable about the AMB A2 mini port with ESP 32 cam. you can follow our website article to learn in detail. There is documentation about this module from the official manufacturer.

All the important links are shared in the website article. In case you want to purchase this port, you can visit sit studio and place an order. The board will cost you around $25 only. Okay. That is enough about the board.

Now let's move to the programming part. The best part of Realtek AMB, too , is that it supports Urdino ID. You can check the Urdino ID SDK for this board from the github repository. Open your audio ID. Go to file, then preference, and paste the following link in the additional boards manager URLs failed.

Go to both manager icons and type IMEBA. So here, you can see the board. Just install it. It might take some minutes for installation. Finally, we are done now.

Now let's apply the blink's case and check the working of this port. Open the Blink's case from the examples menu. Connect the AMB a 2 mini board to your computer using the micro USB cable. From this mode selection option, select the AMPA to mini boot and the COMBORT. The port doesn't have an automatic programmer.

So in order to program the board following the following sequence, press and hold the boot button, then press and release the reset button. Finally, release the boot button. The board has now entered programming mode. Now you just need to click on the upload button to upload the code. After uploading the code, the ordinal ID will show falling messages.

Now press the reset button on AMPA to mini bot to run it in normal mode. The LED will blink for every once again. This completes the testing part. Now we can start using the port for artificial intelligence, machine learning, and neural network applications. Now let us test the camera and stream the project using a web server.

This example uses the camera to capture a JPG image repeatedly. And sends the images to a browser continuously using HTTP creating the effect of a project. Go to files, examples, AMAPA, multimedia, Captured Zap, HTTP, displays APEC continues. Open this case, From this line, change the WiFi as a ID and password and replace with that of yours. Put the device in programming mode again using the boot and reset button.

Now upload the code. After the code is uploaded successfully, open your serial monitor and press the reset button. and look for the messages in the serial monitor. The Realtek AMB to mini IoT board will connect to the wifi network and free in the IP address of the board. Copy the IP address and paste it on your web browser address bar and hit enter.

The project stream window will appear here now. Now you can move the camera in any direction to check the project streaming in the web

server. you will observe fast moving project streaming without a lag on your web browser. Also, look at the project quality. very clear and impressive.

Right? If you are facing lag in the project or if you have slow internet connection, increase the delay in this falling piece of code in the loop section. You may increase the delay to 5, 10, 15, 20 milliseconds, to absorb the sensors. Now let's use this camera module for object detection and identification. The Realtek AMBA 2 mini IoT AI camera board uses algorithms like Yolo, V3, v4v7.

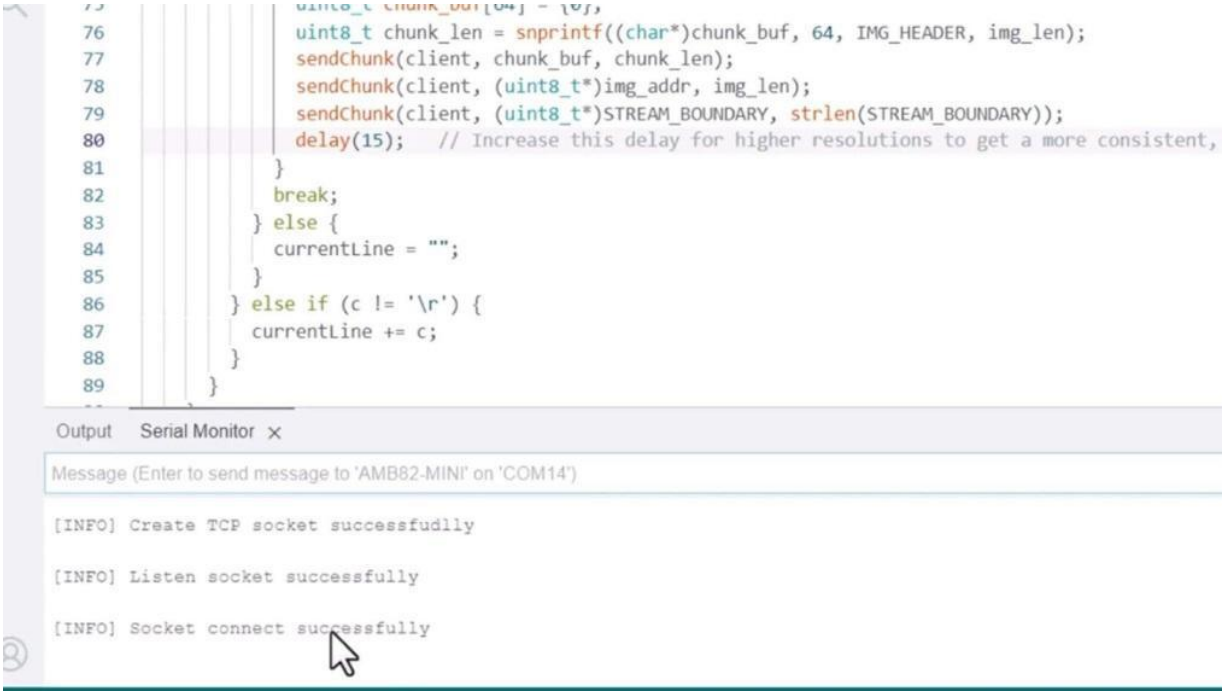
To test this, open the object detection example code from Amiibo and an object detection loop code. This code has 2 parts from this line, change the WiFi, a society, and password. In this header file, there are a total of 80 objects which are trained for this model. Now, compile the code. After your compilation, you can upload the code to the AMB 18 mini port.

Once code gets uploaded, the ordinal IDE will show the following messages. Press the reset button and wait for the AMB A2 mini board to connect to the Wi Fi network. The serial monitor will show the board's IP address and network port number for RTSB. The result of the detected object can be validated using VLC. You may download the VLC media player and install it on your computer.

Now go to media and open network streams. Copy the IP address with this syntax. The default RTSP port number is 554. Now click on play. Open successful object detection, our rectangular frame known as a bounding box will appear around the identified objects with a confidence score.

reflecting the system's coordination of its identification. Here are some samples of images and objects recognized inside the room. Here are some samples of images and objects recognized outdoors. The device is more able to identify the objects that are trained. There

are 80 objects to deactivate the detection of shorting objects and set the filter value to Jiro.



For example, set the filter value to Jiro to exclude the detection of cards. Then upload the code again. Now you can see no car is being detected. For project recording examples, ensure the SD card in the SD card slot. From the example menu, open 1 of the record MP4 examples in file, examples, AMIVA, multimedia, record MP4.

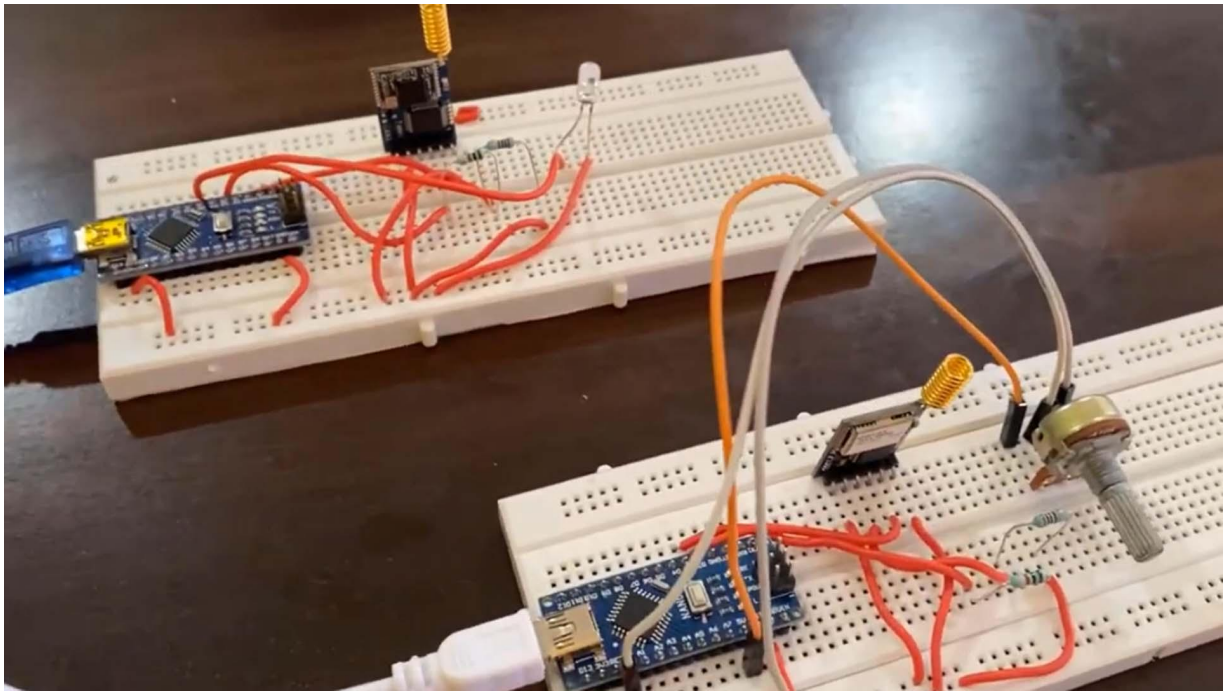
From this line, change the time as for how long you wanna record the project. Now compile the code. After compilation, you can upload the code to the AMBA 2 mini bot. After pressing the reset button, the Amiibo pro to board will start recording MP4 to test record. After the recording duration has passed, the mp4 file will stop recording.

You can now remove the SD card and check the recorded project on your computer. Look at the project quality. Very, very impressive. Right. Alright.

SENDING SENSOR DATA WIRELESSLY WITH LORA ARDUINO

We'll learn how to use RayX, rylr890, Laura Model with R. D. No. The red x LoRa transitional module features the LoRa wrong range modem that provides ultra long range spectrum communication and high interference immunity with minimum power consumption.

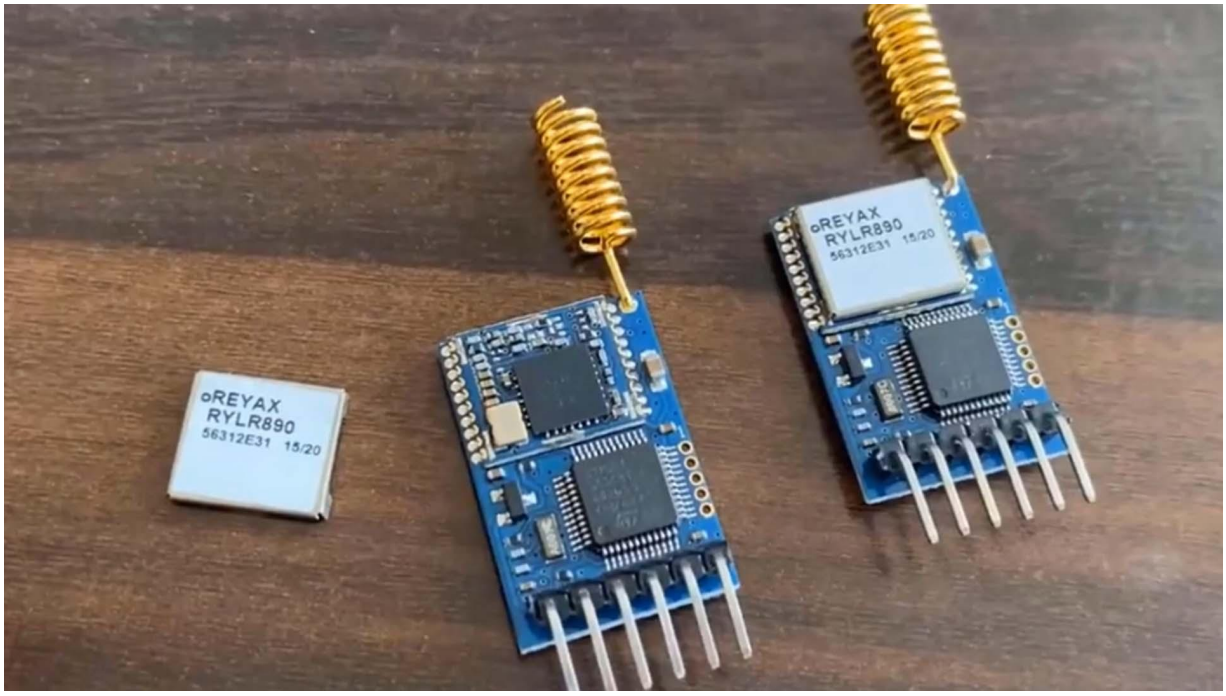
Earlier, we learned about the LoRa module SX 1278 and its interfacing with ADNO through SPIPs. But today, we'll learn about the UAT LoRa module. The u rt loader model can be used with any microcontroller using only 4 pins. The LoRa model, rylr890 communicates up to a range of up to 15 kilometers, and is designed using the best noise detection technology. The lower amortization can be easily interfaced with adenobode ESP 8266 ESP32 or any STM 32 microcontroller. The power condition of this module is very, very minimal So in this tutorial, we'll first make a simple adenolora transmitter, receiver circuit, and do a point to point communication like controlling a LED brightness wirelessly using potentiometer.



In the second example, it will send the sensor data wirelessly from transmitter to receiver. The sensor we are going to use is the BME280 barometric pressure sensor. This sensor can measure environmental temperature, humidity, pressure, and altitude So without getting any delays, let's get started with the LoRa tutorial. Now let us first learn about this LoRaAM model from React Technologies. The LoRaAModILR890 or rylr896 from React Technologies.

It's based on the same tech LoRa chip, Sx1278, and STM32 L151 C86 microcontroller from ST electronics. The S X 1 to 76 chips operate on frequency of S68 or 915 Megahertz. If you remove the outer covering on this board, you will see that the s x 1 to 78 chip interfaces to STM32 l 1 microcontroller via SPI pins. The axrylr8900 LoRa model can be easily interfaced with, you know, using the u r pins The model is breadboard friendly and operates between 2.8 volt to 3.6 volt. It has a very small copper antenna with greater noise immunity for long distance wireless communication, you will at least need a pair of lunar module or many models for establishing wireless communication, there is a list of AT command that can be used to

perform any task like sending the data or receiving the data or putting the device to sleep mode.



You can learn more about the AT commands from the technical AT command documentation, I have attached the documentation part in the website article. So in this project, I have assembled a LoRa model with Arino Nano on a breadboard. You can use a custom design PCV file. This application now Let's go through the product specification. So it is based on the Semtech Sx1276 chip.

and it has an excellent blocking immunity. The receiver current is very low, and it is having a very high sensitivity of -148 dB It is controlled via AT commands. It has a RSSI of 127 DB, and it has a very small antenna that is indicated in the PCB itself. The data encryption is very good and is of AES 128. As explained, the lunar motor is having a UAT interface and operates on the frequency of 868 or 915 megahertz.

So its application includes the IoT applications, and it can also be used in mobile equipment or home security systems in industrial applications, you can use this for monitoring sensor data and for

controlling the equipment. It can also be used for car alarms and so many other applications. Here is a front side and back side of the lower amadeus So on the front side, you can just see 2 chips. There is a React chip and the STM 32 chip. On the back side, it has a reserve position for LED and register.

and also for the SMT 1 connector. And when you see the pins, it is having 6 pins. So in the block diagram, you can see the STM32 processor is connected to a LoRa model via SPIP. And then the one processor, the M32 processor, can be connected to another processor using T XRx and NR ST pins. So in the specification, you can see the power supply is between 2.8to3.6 with the typical voltage of 3.3.

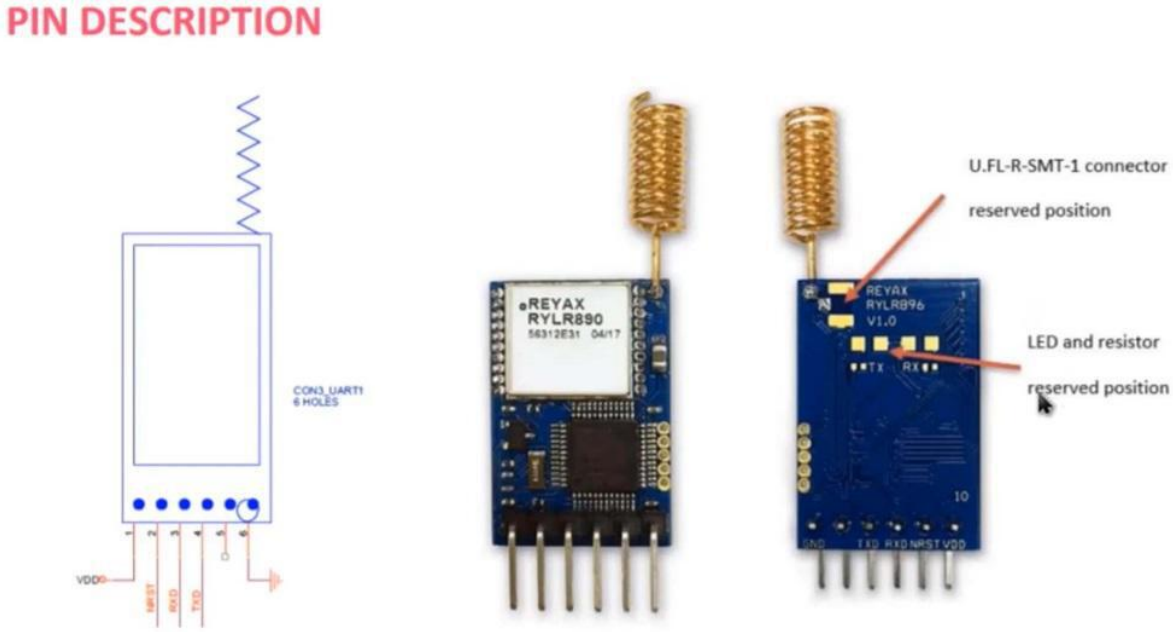
The RF output power is -42 15 DBM, the RF sensitivity, RF input level, the operating frequency, and the frequency accuracy, everything is given here. The communication is a minimum of 4.5 kilometers to 15 kilometers as a maximum. That depends upon the RF parameter. The transmitter current is 43 milli ampere, receiver current is 16.5 milli ampere, and the slip current is around 0.5 micro mperes. All other details like digital input level, digital output level, cycling, and weight, operating temperature.

Everything is given. You can go through the data sheets to learn more about this LoRa module. Now let us go through the first example. In the first example, we'll control the LED brightness wirelessly using a potentiometer. So here is the transmitter circuit.

We have a potentiometer attached to the inert pin up the adeno and the LoRa TX and R X pin is connected to pin number 23 of adenovaya software serial. We have used a 4.7 k and 10 k resistor to form a voltage divided network as its Ardeno has 5 volt output pins, but LoRa supports a 3.3 volt. So in the receiver circuit, the QR communication is still the same, but we're talking about the LED connection. The positive leg of the LED is connected to pin 5 of Ardeno. So here is the assembly on the breadboard. You can see

the Arduino Nano along with the module and then, a couple of registers, LED on the receiver side and potentiometer on the transmitter side.

So when you rotate the potentiometer, the brightness will be twelve, but, before that, we need to upload the code to the board. So here is the code. So we have the first code. We have assigned pin number 2 and 3 as a separate serial pin. Inert is defined as a pin for the potentiometer, and we have initialized the pin mode for the potential meter as an input bin in the loop function.

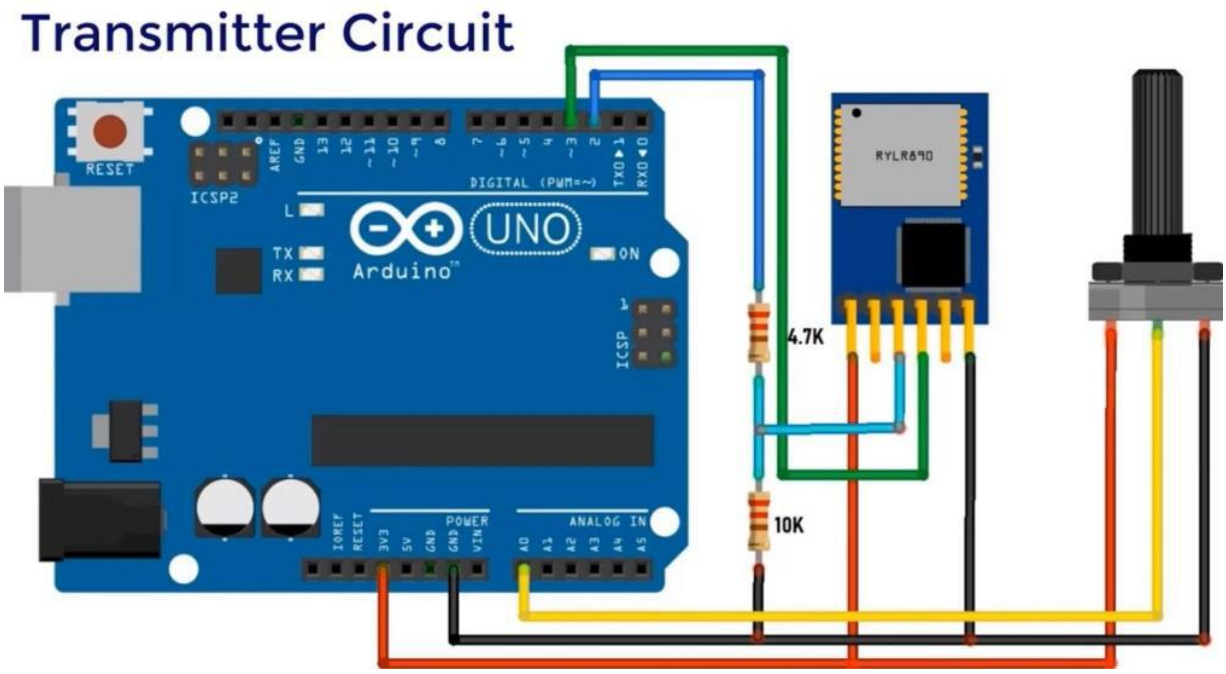


We are mapping the value from the potentiometer to the range of 0 to 255. And then, we are just assigning a string and using the AT plus CND equal to 0 function we are sending the value of, potentiometer from the LoRa transmitter to LoRa receiver and the receiver side, we are again using the software serial, and we are assigning the LED pin as 5, an initial value of LED pin is 0. And we are initializing the LED along with the LoRa. And then if the LoRa string is received, then the LED brightness will increase and decrease. On the basis of the data, data g received from the transmitter module.

So the cell from the code section now, the code is uploaded now you can see when you turn on the potentiometer, the LED brightness will increase or decrease on the basis of the rotation. So when you rotate to a small value, the brightness will be less. When you rotate more, the brightness will increase. And when you reverse rotating, the brightness will completely go away. So you can see this is how we can control LED wirelessly using the LoRa transceiver module.

Now in the second example, we'll send a sensor data So this is a BME280 barometric pressure sensor, which has a ITC interface. So here is a connection. We have connected the SCLP into a 4 of Ardeno and SDA to a 5 of Ardeno. And the connection for the LoRa is still the same as on the sensors. Only there is a connection between Ardeno and the LoRa material that is a Duarte Communication.

So here is the assembly 1 breadboard. So this is a transmitter circuit with BME280 sensor, and, this is a receiver circuit with only LoRa module and Arduino Nano. So we'll just write a code on the transmitter side to read the data from the BME 280 sensor. So this is good for sending the sensor data. So we have included a couple of libraries, including software serial and BME to 80 libraries, and then we have initialized the software serial.



And in the Boyd function, we have just read the data from the library. And we are converting the data to string value. And we are sending it using the AT plus send equal to 0 command to the receiver. So all the values from the transmitter are sent to the receiver after the interval of 5 seconds. On the receiver side, we are doing nothing, but waiting for the packet to receive from the transmitter, and then the packet is printed on the serial monitor.

Now let us see the demo for this project. So after uploading the code, just open the serial monitor on both the side that is on the transmitter and on the receiver side, So here, you can see the temperature is around 28. Presser around 944 altitude around 5.58 humidity around 23 This is how you can send data wirelessly from transmitter to receiver. You can also see there is a minus return That is the RSSI.

SHELLY IOT RGB LED STRIP CONTROLLER VIA ANDROID

In today's project, we will be controlling the RCB LED stripe using stem design Android application. The Diwali time is coming, so I decided to make this project for all of my subscribers.

This is the LED light controller module from cellicloud. The module is wifi based and can connect to your home network. All you need is to connect the LED stripe or LED bulb to this module. In fact, the module, there is an ESP 8266 wifi enabled controller. The ESP 8266 communication is through a sailing car server.

This is the RCV LED stripe of 3 different colors. By mixing the different colors, we can generate multiple colors. It's simple and easy to use. You can control and program the light color and light the pattern you want. You can choose in color and by using the Android application, you can wirelessly control the light.

You can turn it on and off whenever you want. You can even change the intensity. Alright. The sponsor for this project is next busy. Who are of a giant PCV manufacturer in current industry, you can order the PCV by uploading the gold bar file.

Then select the PCV size, quantity, and color. Set the country for shipment and submit an order, and there is good news to tell you. Now, you can get commissioned by inviting your friends to register and place an order next . All you need is to use this link. invite friends to sign up on nextvisavis.com. You can do this by sharing the link on social media or email.

Once your friend registered, you will get a $5 coupon. And once your friend placed an order, you will get a 10% commission, and the commission will be recharged into your balance and you can use that to dissolve the order amount. So this package has been sent all the way from Bulgaria, Europe. This is an IT product from Cine Cloud called Staley RZB W. Cine Cloud is a part of Ultra Electronics.

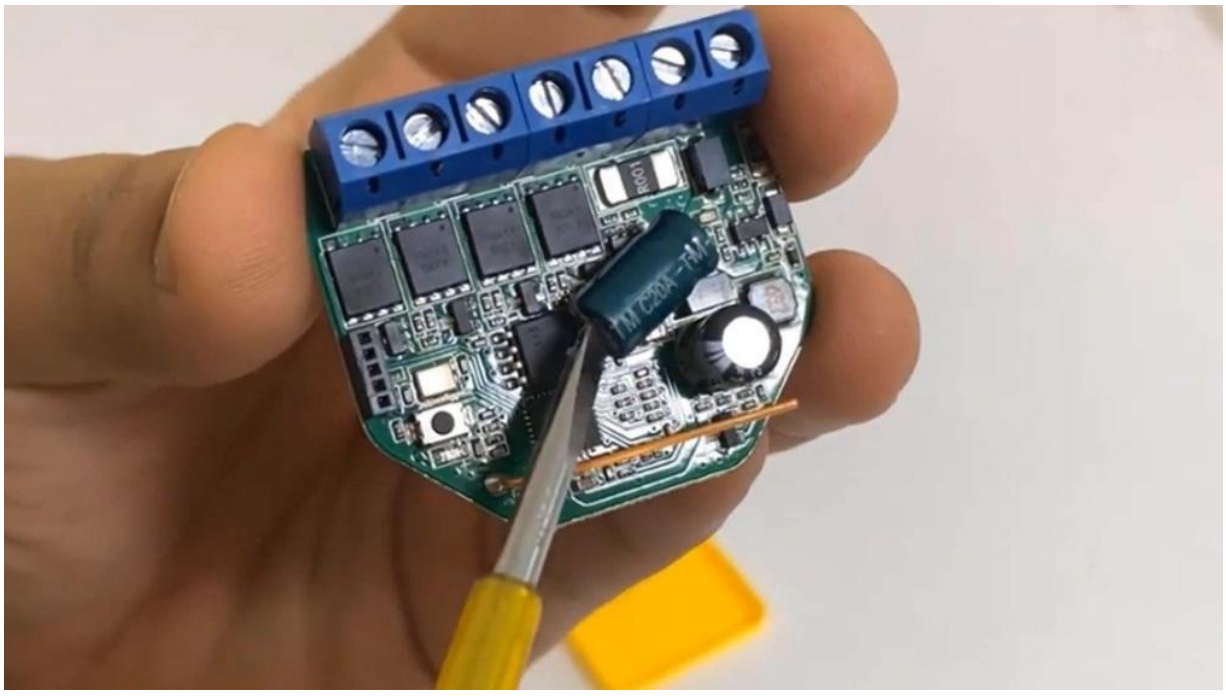
Now, the model uses this 2.4 gigahertz wifi network for communication. It operates on a 12 to 24 volt DC power supply. The power consumed is up to 288 watt. The system is implemented with an embedded web server, and it has a 4 way dimmable night with SSL secure connectivity. You can see here a great product from a Belgian company.

Here is a QR code you can scan it using your Android or iPhone and download the application. I'll explain that later on. But before that, read the official documentation of the product in the description link. It states that the device is wireless and that seems true. It has a remote control feature, and this device is highly compatible with Android iOS, Amazon, Alexa, or Google Home.

It has a white voltage support with serial timer and multiple effects. You will just love this product when you use it. But now let's see what is inside the box. So here it is a bird in yellow, blue, and color, very small color. The black connector also comes with this device which can be connected with 220 volt AC directly along with the limit switch.

On the backside, the current voltage specification is maintained. On the top, the connection pins are mentioned as W, B, ZR, ZND, DC, and I. You can simply connect the wires from RSV and power supply here. Now when I disassembled this box and tried finding what was inside the box, I was surprised to see. The board was having a huge number of components assembled in a small Pacific board.

Singer, it's an ESP 8266 WiFi chip. The same chip is available in modem support. Along with ESP 8266 chips, there is a push button for reset and few power controllers I see for stepping down the power. It has the copper antenna for 2.4 gigahertz frequency. There is also a port where you can connect the UART module and try programming.

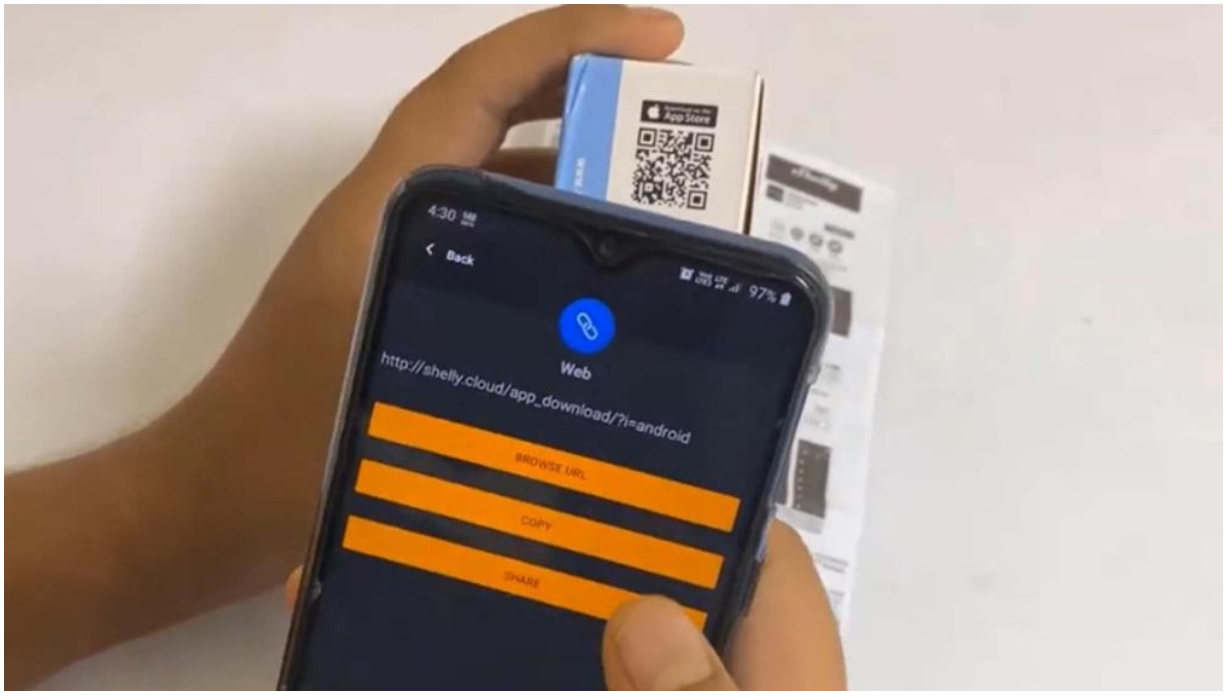


Now here is a dual world RZV LED stripe that I recently purchased from Amazon. The LED stripe has 4 wires. One for the VCC and order for rate, green, and blue LED stripe connection. Inside the box, there is an instruction menu that explains How you can connect the RSV stripe of different models or a colorful LED light. can be directly connected through 220 Volt AC or 12 L224 Volt selector.

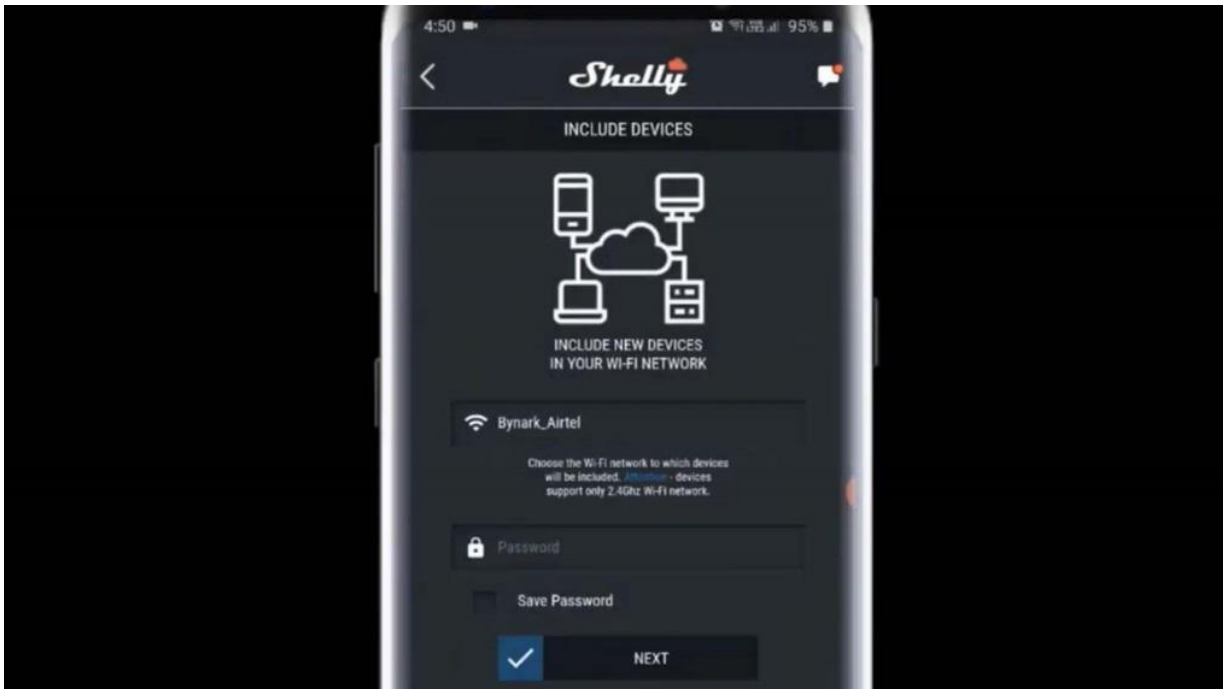
Well, inside the box, there is another instruction manual that explains how you can download the app and set the device. And I will be explaining the instructions in this project. I will be powering this device with a dual voltage adapter. So first, I connect the VCC wire to discipline and ZND wire to ZND. From the LED stripe, connect the VCC wire to the DC of the cellular ZV controller.

After that, I connected the RGB's Stripe to the respective Coreport. On the instruction manual or on the box, there is a QR code. You can start scanning the code to download the Siri app. I'm using an Android phone, so I scan the QR code. So it will take you to the link and the link will forward you to the Play Store.

So install the app and open it. So here is the city app. On the first installation, you need to create an account here, so create an account using a Gmail ID. Once the account is created, you can log in. Now it's time to add the device to the software.



So power on the device with 12 volt DC, then on your mobile phone, open your Wi Fi. The Sally RSV model creates a hotspot, so connect to that access point with a name something like Sally. Now go to the seedy RCP app. So here you can see a device has appeared. So click on a device and then select the device and enter the Wi Fi SSID and password.



So after hitting enter, the device stores the WiFi credentials. You can explore more features of this app by yourself. So now it's time to control the night. On the Android app, select the color that you want. Add just the brightness, check to fix it, etcetera.

So many features and so many functionality in just a simple touch, such a great and a delight controller isn't it. Since this is the time of Diwali, I have decorated the office with LED lights.

SONOFF NSPANEL SMART SCENE WALL SWITCH DEMO

It was sent to me from Shenzhen, China. And surprisingly, the parcel was too big. Out of curiosity, I immediately opened the parcel and found there were a lot of products from Sanjan Sonoff Technologies.

1 of the products was Wifi Best SmartLEDRZV light script. And this was the smart screen wall switch called Anna's panel, which looks awesome. I also got a wifi smart LED filament bulb with creamy white light features. Then this is a smart plug with wifi features, which is also comparable with Google Home and Alexa is my favorite blog. Inside the box, there was an RSV LED strip that contained 5050 RSV with multiple colors.

The same box also contains multiple adapters as well as controllers. I will explain all the products in some other projects, but now I will explain about this awesome product called NS panel from Sonoff Technologies. The panel is a machine while the switch integrates 3 interactive matters, including touch screen control, voice control, and app control, which means you can easily put home control in a place to manage. This touch screen panel has a TFT LCD display on indications, LED, a temperature sensor, and some ports for connecting devices. You can download the app using the QR code for your phone and configure the device using the Bluetooth connection.

The Annie panel will show you the beauty widgets. Play with room temperature, current time, and date. You can press 2 buttons to control 2 different electrical lines. You can manage modules. You can also control your thermostat settings as it has an internal thermostat.

I tested the LED light and fan which were controlled using 2 buttons of the NS panel. Similarly, the Android or iOS app can be used to control the 2 outputs as well. The app will display temperature and 2 buttons to turn on and off the appliance. In this way, you can manage any appliance at your home just by using the mobile app from anywhere in the world. The Anna's panel also has voice interactive features which can be implemented with Google home or Alexa to control the home appliance. Let me show you the demo.

Turn on auto electronics outlet 1. Okay. Alexa, turn off electronics outlet 1. Okay. Hey, Alexa.

Turn on the electronics outlet too. So this project is gonna be exciting. So without wasting any further time, let's get started. Welcome back again. Let me do the unboxing and so you are inside the box.

So this box comes with very nice packing. and Sonoff has named this product as a smart screen while switching. This product is very

unique as nobody else has made this sort of product so far. Thanks to Sonoff for providing me with this unique product. The product has an application called EWE Link app, which is available for both Android users as well as iPhone users.

Behind the box, you can go through the features specifications and abilities of this Anna's panel. The Anna's panel has been designed for both US standards and European standards. Similarly, it also compiles with the guidelines with Google, Apple, and also Amazon, Alexa. Let me open this box and show you what is inside it. The box comes with a user manual book for easy setup And this is an analyst panel with all in one control center.

It is featured by powerful sensors and processors. At the backside of this board, there are 5 pin outs for input output connections out of the 2 r input and 2 r output and one is the unused port. There are two buttons on the panel which can be manually pressed. The left and right buttons can be used for configuring the device as well. The button portion, you can see a small LED indication and also our temperature sensor.

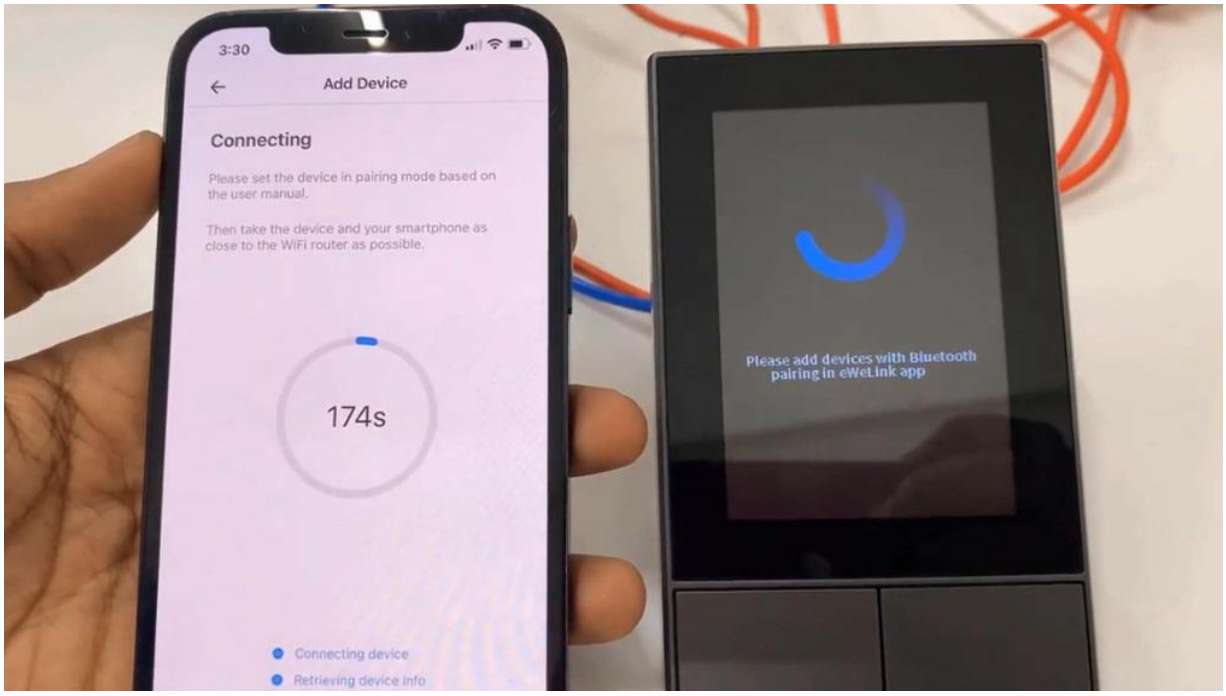
On the backside, you can remove this plastic covering so that you can have access to the screws. The back end front portion of this panel is disabled. Simply by pulling the device, you can remove the font from the back, but the device becomes non usable separation. There are 2 long screws inside the packet as well. The screws can be useful in fixing the device on the wall.

A manual is also provided in the box which gives all the information regarding the detailed method of configuring and setting up the analyst panel. There are four ports at the backside of the panel to connect the electrical appliance. First, use a screwdriver to loosen all these screws. On the panel, there is a mark as n l in and l out. The n is the neutral whereas l is the lib wire. Connect the 220 VSE input to this 2 socket.

That is n and l in. So now the device is ready to turn on as we have done the SC input wiring. Connect the output terminal of the panel to the 2 out devices as per the circuit diagram here. The connection is similar to the relay. In my case, I will be controlling the SE bulb often.

After the connections are done, cover the casing with the slider. To turn the device, supply the device with 220 VSE supply. So now you are ready to go. When Anna's panelists are in power mode, It enters into a configurations mode and asks you to add a device with Bluetooth pairing. Download the EWE link and through play store or Apple store simply by scanning the QR code on the instruction manual.

After downloading, open the app, and grant all the permissions, then create an account using your email or phone number and then sign in. Now there will be an option to add a device. So click on add. Choose Bluetooth pairing to put the device in a pairing mode, click on the button of Anna's panel for at least 5 seconds. Once the Bluetooth device appears on your phone tap, it connects with an s panel.



Once connected, it will ask you to select the wifi network and enter the wifi credentials. So enter the password and save it. Now the device will try connecting to the Wi Fi network. The device has been successfully added now. from the list, choose any option.

So our dashboard appears on Anna's panel with the beautiful widgets. And there will be 2 devices appearing on a mobile app as channel 1 and channel 2. Let's go for the testing part now. As you can see here, the Anna's panel screen displays the current time and date along with indoor and outdoor temperature. Pacing the button will turn on the bulb and fan.

So this is like a manual switch. You can turn it on and off repeatedly. I use the CFO bulb and fan as an example. you can use anything else for your application. Now, when you slide the analyst panel window, It will show an option to add widgets, which can be done through the app.

Similarly, when you slide the screen from top to down, it will show you the option to change the brightness and also to change the device orientation. Currently, it's on the portrait view, and I would like

to change it to a landscape mode. So this is how the system works. On the mobile applications, there are 2 buttons at channel 1 and channel 2. You can press them to activate or deactivate the electrical appliance again.

This is very interesting. You can use your smartphone from anywhere in the world to control your home appliance. You can also expand the app with a key type view. You can also tap these two buttons here again to control the light or fan. The app also has an option for scheduling.

You can select and activate the schedule to turn the end of the appliance automatically. Select and save the timing from the app manually. The app also has an for setting the timer, you can add the timer in hours and minutes from the list. Then allow channel 1 or channel 2 from the list. You can also activate the loop timer option from this option, you can enable, disable, or repeat the activity again and again automatically.

The best part about the device is it has a thermostat setting. From the button, select the thermostat, and here you can see 2 options. A heater or a cooler. Select any of them from the list. Once selected, a screen will appear both on app and on Anna's panel screen. Click on the power on button to activate the thermostat, you can slide this circle to set the upper limit or the lower limit temperature.

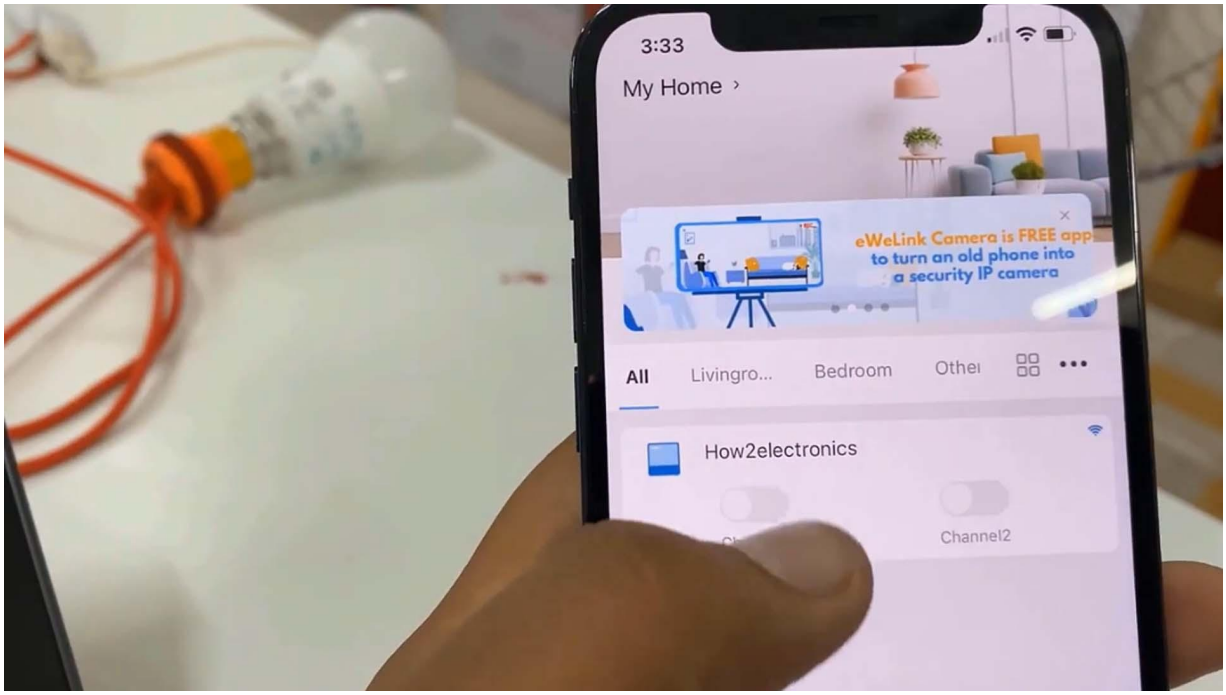
Whatever changes are made on the app, the same will be absorbed on the analyst panel screen. There is also an option to activate the manual mode or an auto mode. The same thing can also be done from the analyst panel screen. On the app, there is also an option to add screens. This is like adding the if and if conditions.

The conditions may be any timing of the day or maybe sunrise sunset, or a smart device. Select anything from the list. The device also can be controlled via Google Home and Alexa. I got both

Google Home and Alexa with me. As an example, I will activate one of these for the demo.

I selected Amazon Alexa, which is my favorite. First power on Alexa. Then from the app store download Amazon Alexa, I have already downloaded it So I will open it simply from the Edge device option. You can add the Annie panel. Go search for the link from the source option and edit.

The Alexa will start looking for the device to connect, it may take up to 45 seconds to connect. So here you can see that Alexa has found 3 devices. Click on the next and add all the devices 1 by 1. So now you are ready to go. Let us see the demo now.



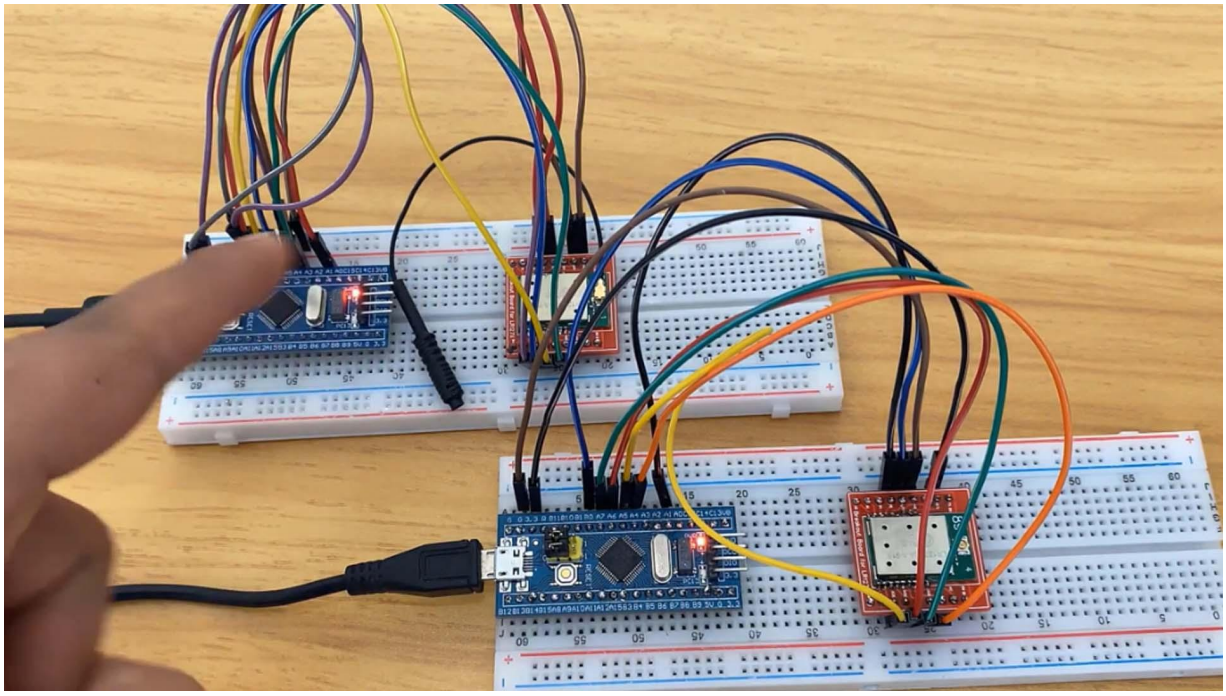
Hey, Alexa. Turn on electronics. Okay. Hey, Alexa. Turn off auto electronics.

Okay. Hey, Alexa. Turn on our electronics outlet 1. Okay. Alexa, turn off electronics outlet 1.

TESTING SX1276 868-915MHZ WITH STM32 MICROCONTROLLER

The parcel contained a few LoRa modules called LR 1 to 76. This model uses an SX 1 to 76 LoRa chip and works on 950 Megahertz frequency. The 61276 incorporates the LoRa sprayed Spectrum modem, which is capable of achieving a significantly longer range than existing systems, based on FSK modulation.

As this board operates between 1.8 volt to 3.7 volt, I decided not to use this board with Arduo. The Arduino boards have 5 volt SK pins, but Sx1276 doesn't tolerate Five volt. So the best alternative to Arduino is STM 32 board. The SDM32 is a lower power controller with 3.3 voltmeter pins. In this project, we will interface sets 1 to 7 or LR1276 model with SDM 32 microcontroller.



This tutorial consists of 2 examples. In the first example, we will send a simple hello world message from Laura Center or transmitter to receiver. But in the second example, We will send the sensor data wirelessly. The DHT 11 humidity temperature sensor is best suited for testing application. The SDM 32 LoRa Center will send the sensor data to SDM 32 LoRa receiver.

Now, let's just learn about G Plus IIT, Laura More LR 1 to 76. These are the pair of LoRa modules for g plus IOT. We will be using this for point to point communication. Dellar 1276 model is designed based on SX1276. The SX1276 incorporates the LoRa's Spirit Spectrum modem which is capable of achieving a significantly longer range than existing systems.

At maximum data rates of LoRa, the sensitivity is 80 decibel, which is better than FSK. But using a 20 PPM crystal, Laura can improve sensitivity by more than 20 decibels. And this model has the best communication reliability. For maximum flexibility, you can focus on sprayed spectrum modulation bandwidth, sprayed factor, and error collection rate. Let's check its data set now.

The maximum link budget is 168 decibel with an RF output of +20 DBM at 100 Megawatt. Using a power amplifier, the efficiency is +14 DBM. The program you will be treated to is 300 Kbps, and sensitivity is up to minus 148 DVM. The receiver current is only around 9.9 milliampere, and this supports all the modulation techniques like FSK, GFSK, a SK Lora TM, etcetera. The dynamic RSSI is 127 decibels with a package engine up to 256 pipes with CRC.

It also has a building temperature sensor along with low battery indication. The application areas of this LoRa module is automated meter reading, building, home automation, alarms, and security systems and also in industry and agriculture. You can go through this data state to learn more about this module. This is the block diagram and this is the specification table. Here you can see the operating voltage between 1.8volt to 3.7volt.

The current consumption in idle, standby, receive, and transmit mode is very, very low. You can use a helical antenna by directly soldering on the module. You can also use an external UFL antenna with better power by directly connecting to this terminal. These are the details of the pin as it has a total of 18 pins. The pins are SPI pins, QR pins, and some GPI pins.

As clear, you mentioned here about the PCV details and dimensions. I decided to design the PCV using the same details. I used Easyeda to design the PCV. Here is a PCV file. In the 2 d and 3 d view, the PCV looks something like this.

You can download the Gober files from the link in the description and order your own PCV. Here is the PCV, your total number of 5 for the product. The PC looks exactly the same as soon in the three d view of the software. The build quality and finish are just superb. Sure.

You will need to solder mail headers on both sides to use this specifically with the break board. After soldering the LoRa model on the PCV, It looks something like this. I used a 915 Megahertz antenna here and connected it to the UFL connector. You can use a brake board and interface this model with any microcontroller with 3.3 volt logic. So, let's just start the interfacing process.

I used an SDM 32 F103 C service microcontroller from SD microelectronics. This board is known as blue pill and has support for using arduino programming, and the GPI pins are having a 3.3 volt logic. Therefore, Cloromodu is friendly to this controller. Here is a connection mapping between Sx1276 and the STM32F103 C development board. connect the SPIP of the LoRa module as instructed here.

I used a break board for the assembly. I assembled the pair of sockets here. 1 of the circuits will act as a transmitter and other will act as a receiver. Use the antenna of a given frequency on both modules, Otherwise, you won't be able to achieve maximum distance and signal power will be very low. Jerry is the code for both sides.

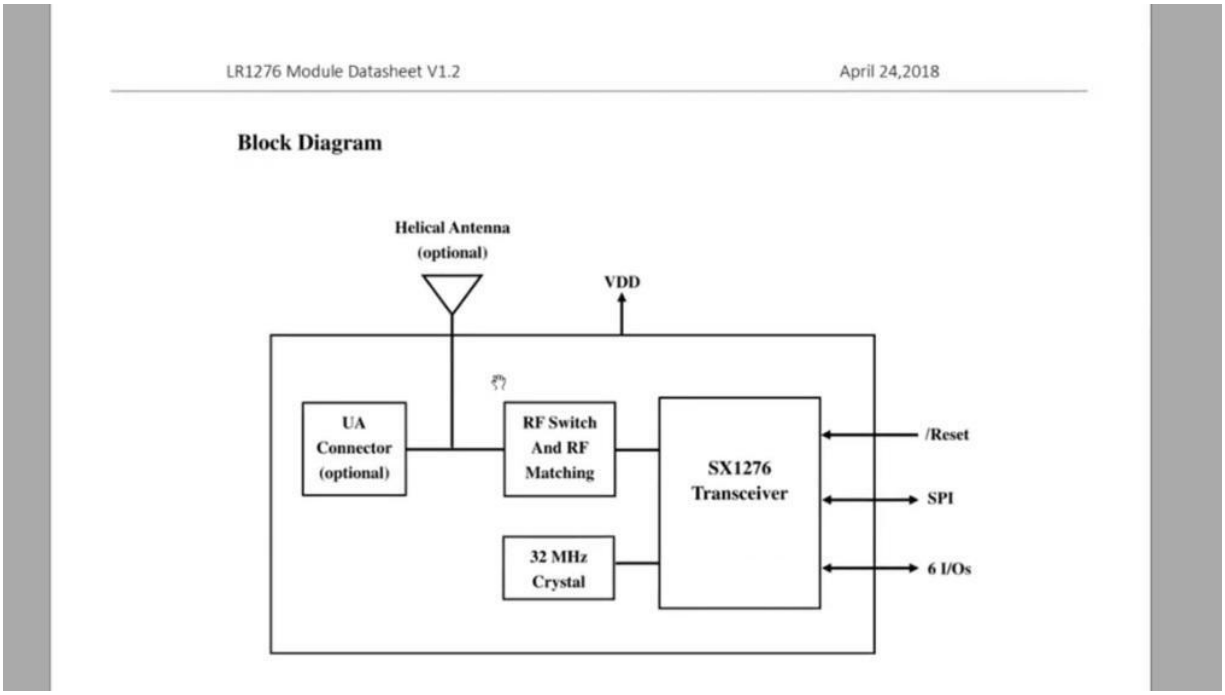
One is the transmitter code and other is the receiver code, and this code uses the STM32 LoRa library. Then we defined SS, RST, NTI 0p. We also defined transmitting power, LoRa frequency band, and

then encryption code. The encryption code on both codes should be the same. Else, the receiver won't receive the message.

In the setup section, we initialize the serial function and save the other error parameters. In the loop function, we send the hello message with the counter number. Laura will transmit the message after an interval of every 5 seconds. Similarly, in the receiver part, everything is the same except the loop part. In this part, when the message is received, the LoRa package will be parsed and will display the parsed message with RSSI.

Now go to tools and select STMF103 series board. Select the variant as mine is 64 k flash. select the upload method that you are using. Here, I'm using the upload method as STM 32. Do we know bootloader as I have already installed the bootloader.

and leave everything else the same. Select the com port and hit the upload button to upload the code. And using the same method, upload the code on the receiver side as well. After uploading the code, open the serial monitor on both transmitter and receiver sketch. In the transmitter sketch, you will observe the message has been sent.

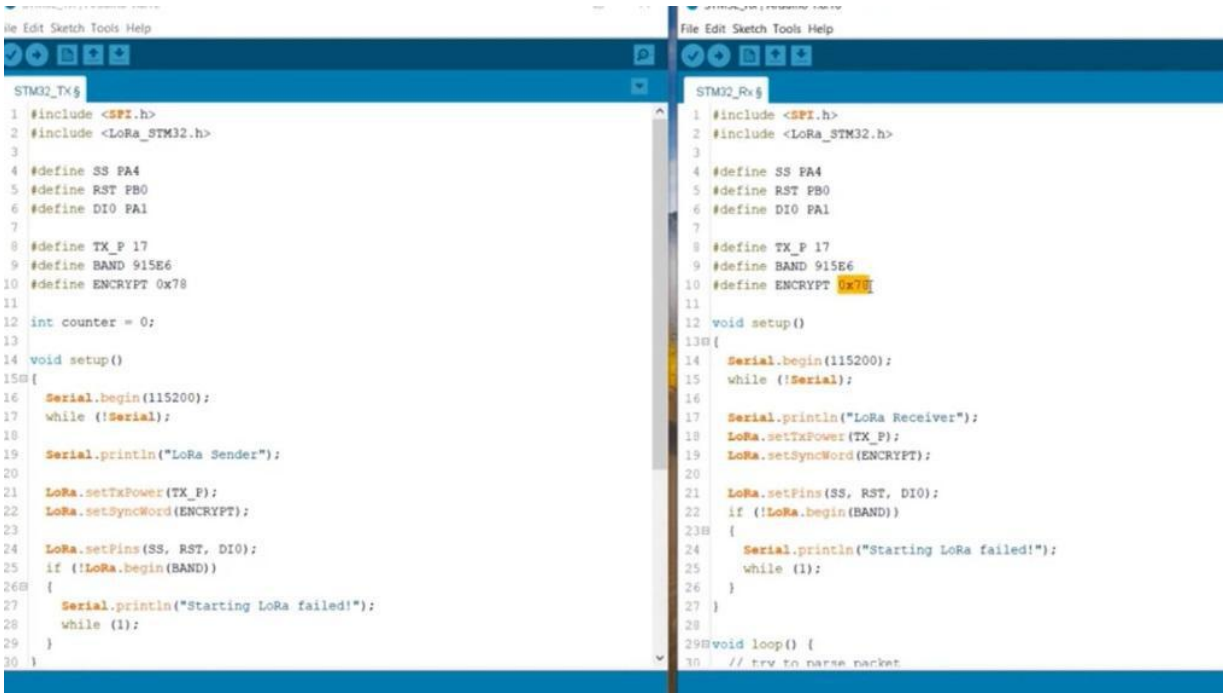


and the receiver has received the same message along with the signal strength. Now, let's see the second example. In this example, we will send sensor data wirelessly. The sensor we have used here is DHT11 humidity and then press the sensor, and we will read the sensor data in the transmitter end and send it to the receiver end. To display the sensor reading at the receiver, we have used a 0.96 inch I Square C OLED display.

Though late will display the received message from the transmitter, Both the connections are easy and require a few wires. In the transmitter, we connected the DHT 1 sensor to the same previous circuit. And in the receiver part, we connected the only display to the I Square C pins. The connection for both circuit boards is pretty simple. You can refer to the website article for the connection.

While moving to the cooling part, we have added a DHT sensor library on the transmitter and then use the library to read the humidity and temperature value. Then we combine all the strings together and send the message in the array. While on the receiver part, we added the library for our latest play like SSD 1306 and GFX library. In the loop part, we define the position of the received

message to separate the received variables. The received variables are then displayed on an OLED display.



```
#include <SPI.h>
#include <LoRa_STM32.h>

#define SS PA4
#define RST PB0
#define DI0 PA1

#define TX_P 17
#define BAND 915E6
#define ENCRYPT 0x78

int counter = 0;

void setup()
{
  Serial.begin(115200);
  while (!Serial);

  Serial.println("LoRa Sender");

  LoRa.setTxPower(TX_P);
  LoRa.setSyncWord(ENCRYPT);

  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND))
  {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}
```

```
#include <SPI.h>
#include <LoRa_STM32.h>

#define SS PA4
#define RST PB0
#define DI0 PA1

#define TX_P 17
#define BAND 915E6
#define ENCRYPT 0x78

void setup()
{
  Serial.begin(115200);
  while (!Serial);

  Serial.println("LoRa Receiver");
  LoRa.setTxPower(TX_P);
  LoRa.setSyncWord(ENCRYPT);

  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND))
  {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}

void loop() {
  // try to parse packet
```

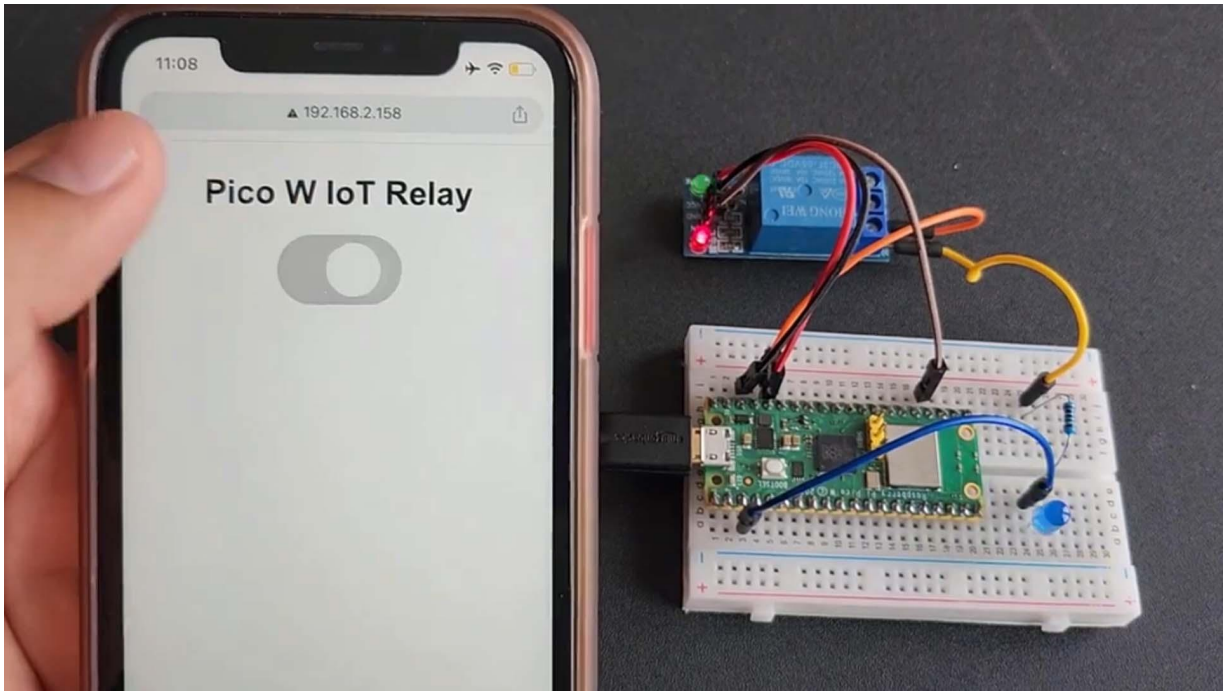
So upload both the code again on the transmitter part as well as on the receiver part. So here is the test. The transmitter side is reading the humidity and temperature sensor data and sync the data wirelessly to the receiver. On the receiver side, the DHT 1 month temperature and humidity data is displayed wirelessly on an OLED screen. You can keep both circuits apart and check the distance as well.

USING RASPBERRY PI PICO W WITH MICROPYTHON CODE

This kit is one of the starter kits for Raspberry Pi Ico W. The Raspberry Pi Ico W was released almost 6 months ago and has wifi capability like ESP Chips. The ultimate kit has 450 plus components.

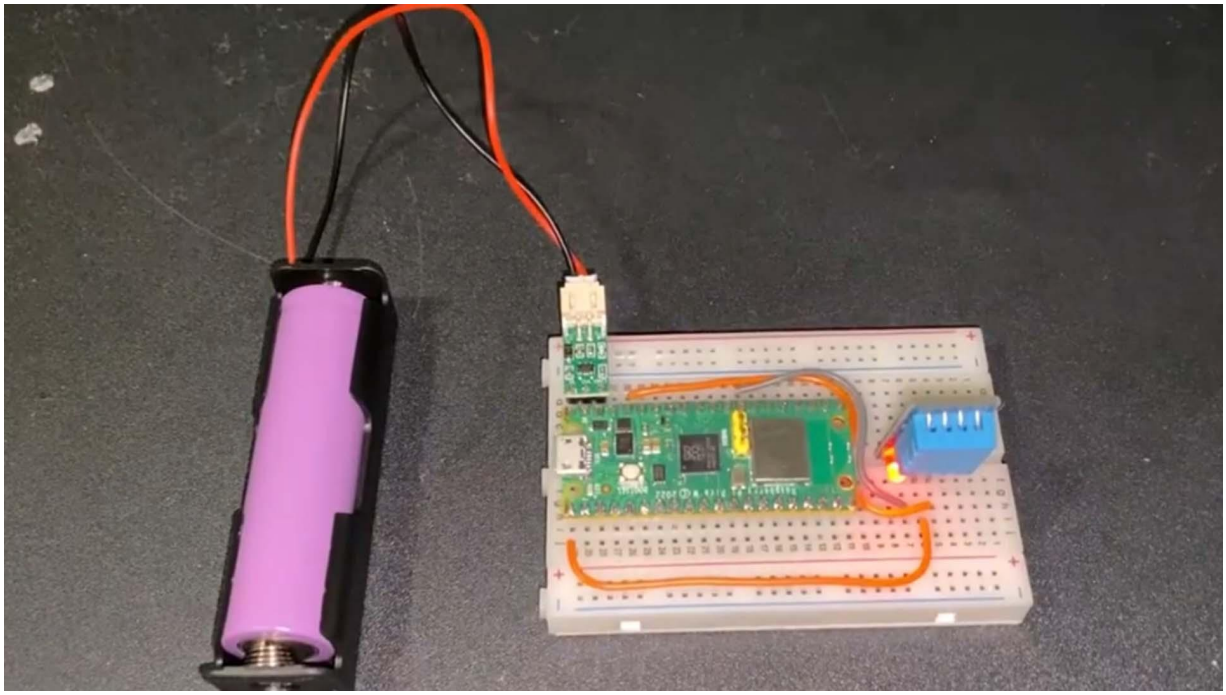
including registers, capacitors, transistors, sensors, modules, breadboard, and jumper wires. Using this, you can build 117 projects. Can imagine 117 projects. It is really awesome. In this guide, we will cover 10 amazing IoT projects using the Raspberry Pipico.

We will use IoT platforms like Thin Speak, order fruit, IO, link and open weather map among others. We will also use a web server for a few updates projects. So let's type Deep and see what the top end projects are. This project is sponsored by PCV and SKU online. The SKU online is a 1 stop electronic component sourcing platform.



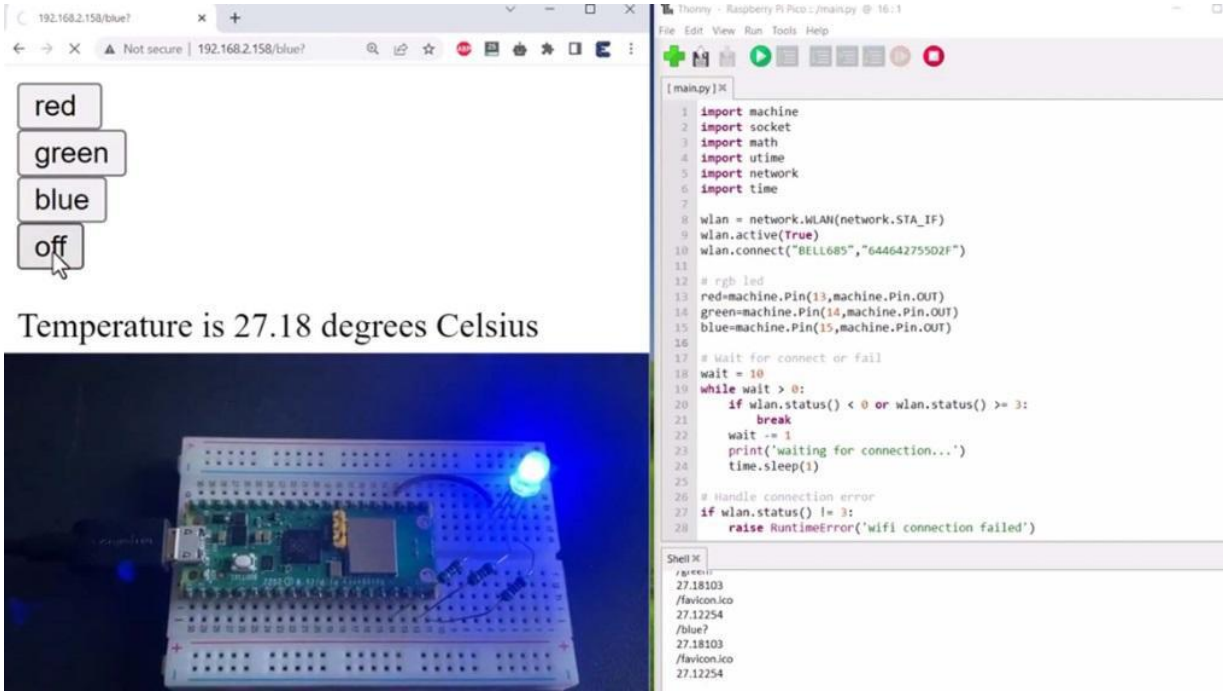
SKU online provides a complete set of electronic component business services. All you need to do is search the electronic component with a part number. It has the fastest charger engine. It will show the list of all available parts. Add the part to your cart, then place the order To get started, check the link in the description.

In the first project, we will use the ThinkSPEAK platform to monitor temperature and humidity data. The DST lab in humidity and temperature sensor is connected to the Raspberry Piico W. Using the Micro Python code, we did the sensor data and sent it to the thin speech server. Here, you can monitor the data from any part of the bot. The second project on our list is related to the Pipico w wireless web server.



Using this web page, we can control the RGB LED or 3 different LEDs. The web page will also display the internal temperature sensor readings. The 3rd project is ready to avoid the station using the open weather map API. The Raspberry Pi Code WE can be configured with micro Python code that has API information, which retries the weather data from an open weather map. The way the data can be displayed on a section cross to ICORC LCD display.

Using this, NTP times are we can display the time and date parameters. In the 4th project, We will use the great IoT platform called Adafruit IO. Using the Adafruit IO dashboard, we can control the new feature ArchIP reality strip. You can generate any color you want. And by mixing Archippy color quantities, you can glue the LED strip with the color you want.

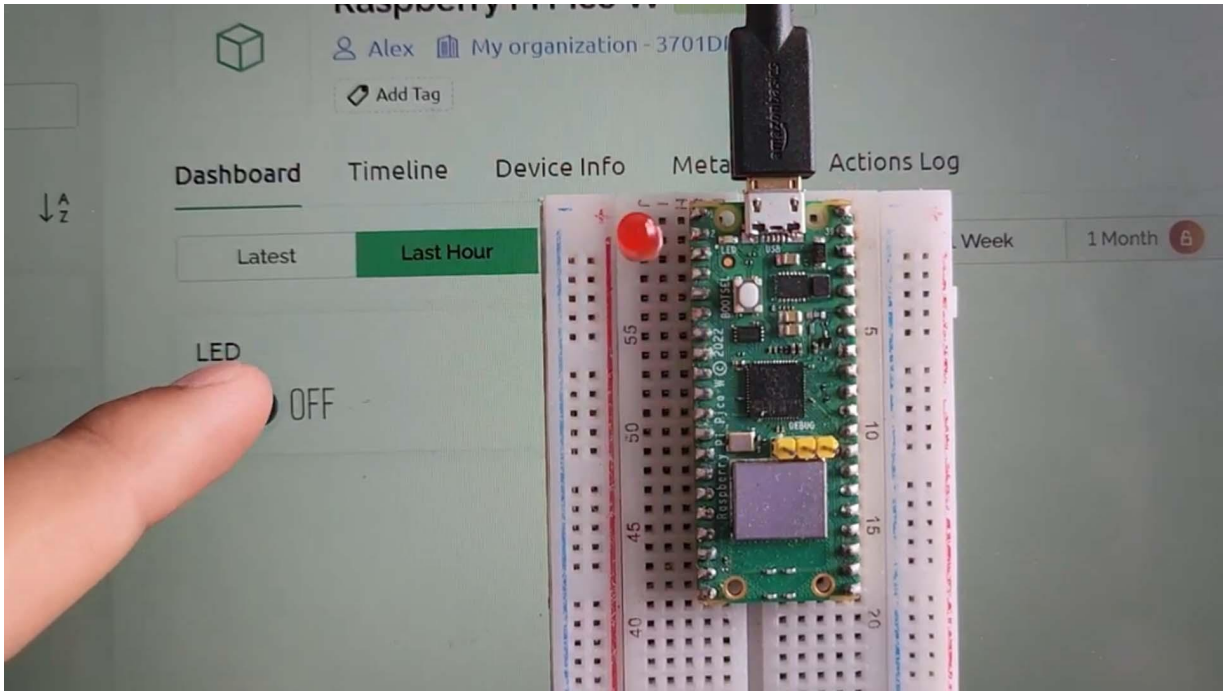


The 5th project is related to the web server. In this project, we will use the BMI 20 sensor to read the temperature pressure and humidity values using microvital code. The readings are displayed in a beautiful workplace, and the reading is operated automatically after a few seconds. You may replace Miami Tuiety with any other center. In the 6th project, we will control our relay with IOD.

Using the Pipico W web server, we can send on and off commands to control the relay. You can connect any home appliance across the Relay and control it using the web page within the same network. In the 7th project, we will use the most popular IoT platform called Blink. using the Blink application, we can control an LED. When an on command needs to be sent through the Blink dashboard, the LED turns on.

When an off command is sent through the blink dashboard, the LED turns off. The 8th project is again related to the Blink application. In this project, we will monitor the soil moisture parameter using the capacity deep, soil moisture sensor. We will rate the sensor analog data and convert it into our fashions. Then, you will push this reading to the Blink dashboard.

You can monitor this full moisture value from any part of the world and use this project in smart farming. The 9th project is a home automation project using the blink app and raspberry Pipico w. The Pipico w can be programmed to control 4 resi home appliances remotely using the print app. For the demo, we used 4 different color LEDs for this project. you may use a real and control multiple home appliances.



The 10th project is again awaiting station projects using the BAMI 280 sensor. Instead of a web server we will use the Blink application to push the BME 2 80 sensor pretty. The web server is local, and Hence can be monitored from any other networks. But, using Blink, you can monitor this online from any part of the world. So, that is all for the 10 projects.

TOUCH SCREEN
CAMERA PROJECT
GAME SLIDE SHOWER

We will learn about this great product from MegaFabs which is an embedded USB 32 camera model with a 3.5 inch TFT touch screen display. The TFT LCD driver is ILI 9488 It uses SPI for communication with ASV 32, and it also has a 2 megapixel OV 2640 Camera for photography or face recognition applications.

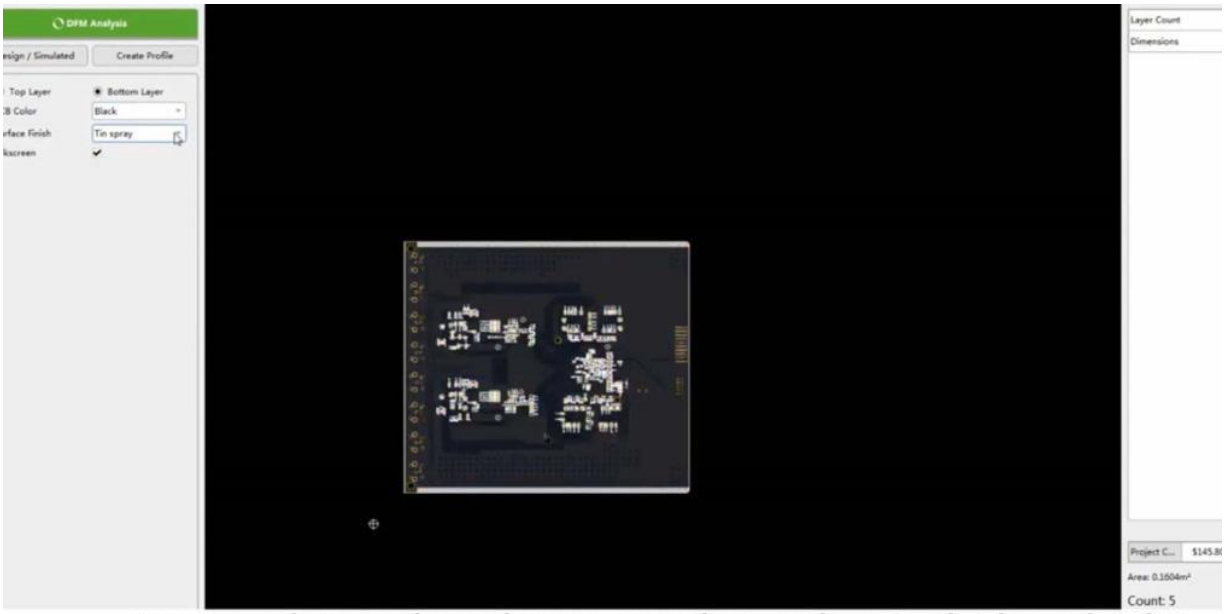
The board support Arduo or micro Python programming. This version is a touch cabinet. You can also get the resistive type product. Both are precise and stable. Using this product, we will make 5 different projects as an example here. First, we will make a game called 2048.

This game can be played easily like a normal project game or a mobile game. We will also make a flappy bird game and get a score as long as you are able to fly the bird. In the third example, we will use this device for photography. We will capture images and display them. Likewise, in the first example, we will draw a picture that is based on touch.

You can draw or write anything. What do you want? And finally, on the 5th example, we will make a slideshow and will display many different images as a slideshow. Isn't this device so exciting and fun to use with a lot of applications? We will learn more about this product now.

So without any delay, let's get started. This project is sponsored by Nextvisavis. Currently, Nextvisavis has developed a PCV design analysis software called Next DFM, which is a design problem detector or an engineering solution provider. You need to import a cover file with just one click. The visual graphics are easy to read, so you can make sure that the file contains all the necessary data.

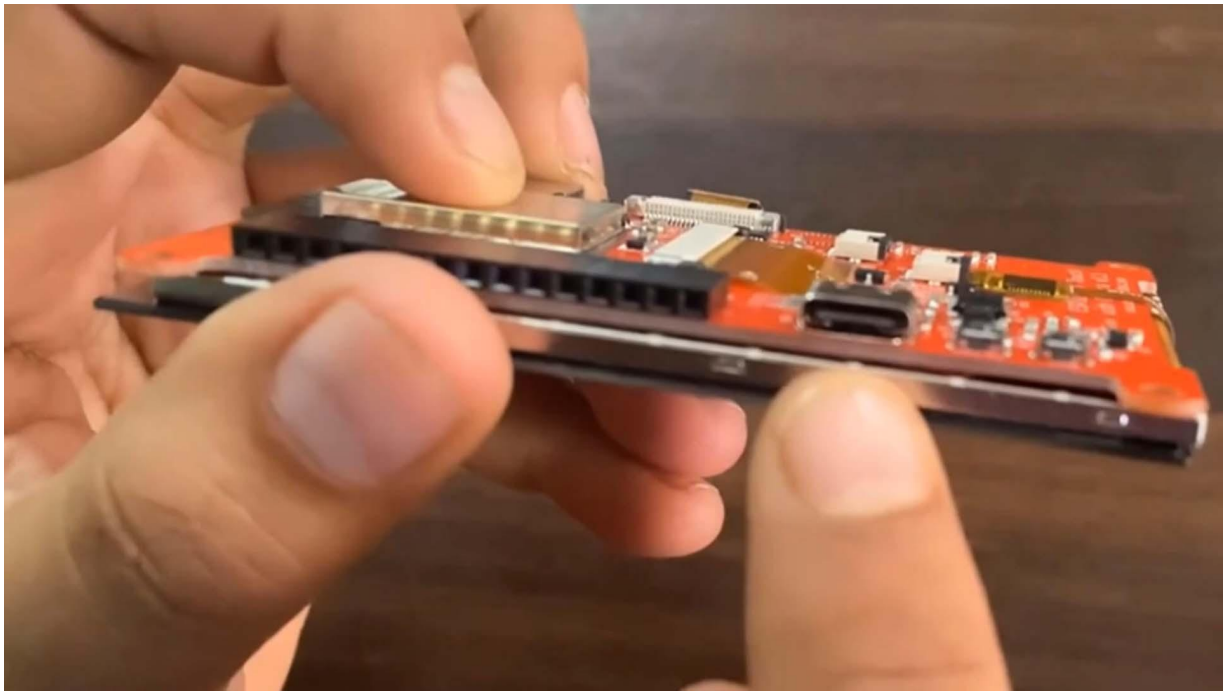
It helps you quickly familiarize CFM design specification and production needs to determine whether there are any manufacturing constraints. It also effectively checks whether there are hidden dangers in the design file. select the available part and evaluate the cost in real time. So try to start using the stool for Pacific analysis. The link is given in the description.



You can choose the color you need to replace in the board color, and the surface finish you need

Now let's learn about the ASP 30 to 3.5 inch touch screen camera. So this is the product that I recently got from Make a beautiful 3.5 inch touch screen display based on ESP Thirty 2 rover. It has a built in two megabit over 2640 camera, which makes it a perfect platform for your E SP 32 project. At the back of this product, you can see an E SP 32 chip from Spercip system. And there are a lot of connections for LCD interfacing and also for camera module interfacing.

The DFT LCD driver is ILI 9488. It uses SPI for communication with ESP 32. This is the micro SD card slot for editing an external SD card. This card can be used for storing files and images. This is the type of USB port, basically, or US speed to your art converter for ESP 32 programming.



You need to type c data cable. And simply by connecting to the board and to the computer, the ESP 32 chip can be programmed. This is a female header part. The ports are basically the input output port from ESP Thirty to the board where you can connect multiple sensors and other modules easily. The SP 32 TF Detouch supports Arduo or micro Python programming.

This version is capacity type, but you can also use the resistive both are precise and stable. You can select the one you prefer. And with this camera, you can make applications such as remote photography, face recognition, and so many IoT applications. To use this board for camera and storage applications, you need an SDGard. I use a 16GB SDGard.

This card can be inserted into the SD card pocket simply by pushing. Apart from this, there are 2 push buttons for put and reset as present in ESP 32 board. Now let's see some of the specifications of this board that you need to know. The board has a 3.5 inch DFT display with 320 by 480 screen resolution. The touch type is capacitive or resistive.

You can buy a d. The a s p 32 controller used to erase a s p 32 rover controller. It has a 2 megapixel of a 2640 camera. The board supports both VLE and Wi Fi type connection. The board has the onboard USB to u art converter used for programming applications.

As explained earlier, it has an onboard SD card socket and the USB connector type is the type c cable. If you want to purchase the ready made board, then you can visit this official site. The capacity of the type board will cost you around $34.8. The production description and documentation is nicely explained here. Similarly, this is the resistive type board.

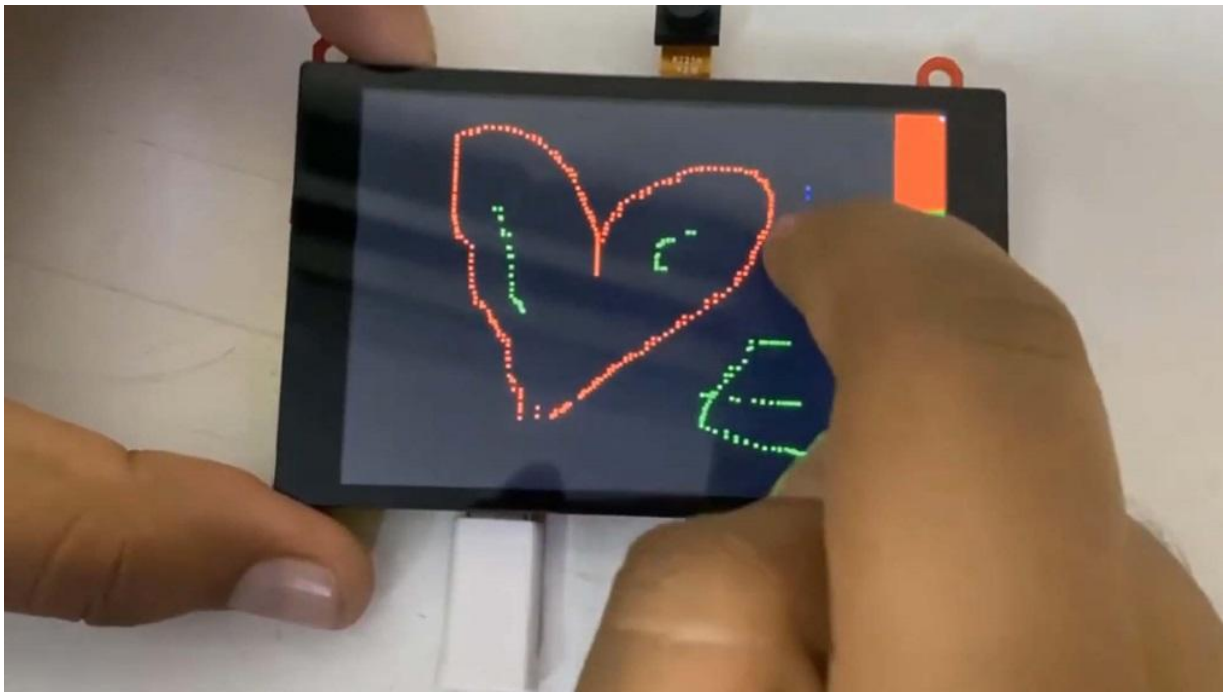
The board cost around $32.8, slightly cheaper than the capacitive type. and The documentation and specification of this board is explained here as well. Now let's go to the few applications of the product, we will first make a game called 2048. The game will start working as soon as the code is uploaded to this port. So here we go.

Let's play it again now. You need to move your finger and slide it left, right, up, and down to play it. Give it a try. Now let's see another game here. It is a flappy bird game.

You need to keep touching or tapping the screen in order to prevent the board from colliding and dying. If you are a pro gamer, then you should try the simple game as well. Apart from Kim, Now let's just use the application of the camera. So after uploading the camera code, the camera UI will appear on the screen. There are 3 options: take a photo, last photo, and start the stream.

In order to take a photo, click on the tick photo option. So in this part, the clicked photo name appears. You can then click on the last photo to view the clicked photo. In order to start it again, click on the start streaming. Apart from this, you can also use this camera for facial recognition applications.

You can also use the touch based screen to draw a picture and write some characters. Just do you need to choose the color and start drawing anything randomly? Not while you join, you can write anything or draw any alphabet. Such are core products for artists and those who are interested in learning painting. The screen can also be used for making a picture presentation as a slight soap.



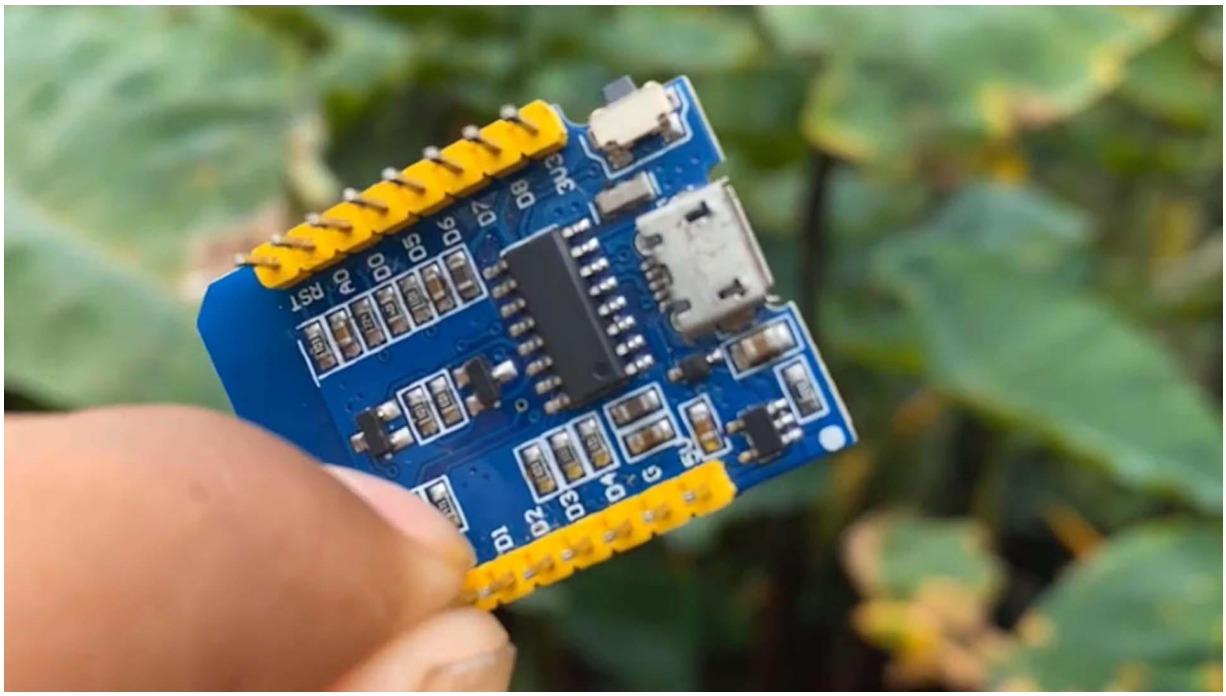
Just to save a few pictures with a resolution of 480 by 320. The image formats load to BNP format So once the code is uploaded and the device is run, the slide begins and you can see pictures scrolling. Such a great product, isn't it? Well, these are the basic applications of all the examples used for this product. In some other projects, I will come with more interesting applications.

I hope you liked this project. And if you are new to this channel, please hit the subscribe button.

USING ARDUINO IOT CLOUD WITH ESP8266

Recently, the Oceano community launched their IoT platform called Odeon IoT Cloud. The ordinal IoT cloud provides an end to end solution that makes building connected projects easy for makers, IoT enthusiasts, and professionals from start to finish. The platform enables different methods of interactions, including HTTP REST API, mqtt command line tools, javascript, and wave sockets.

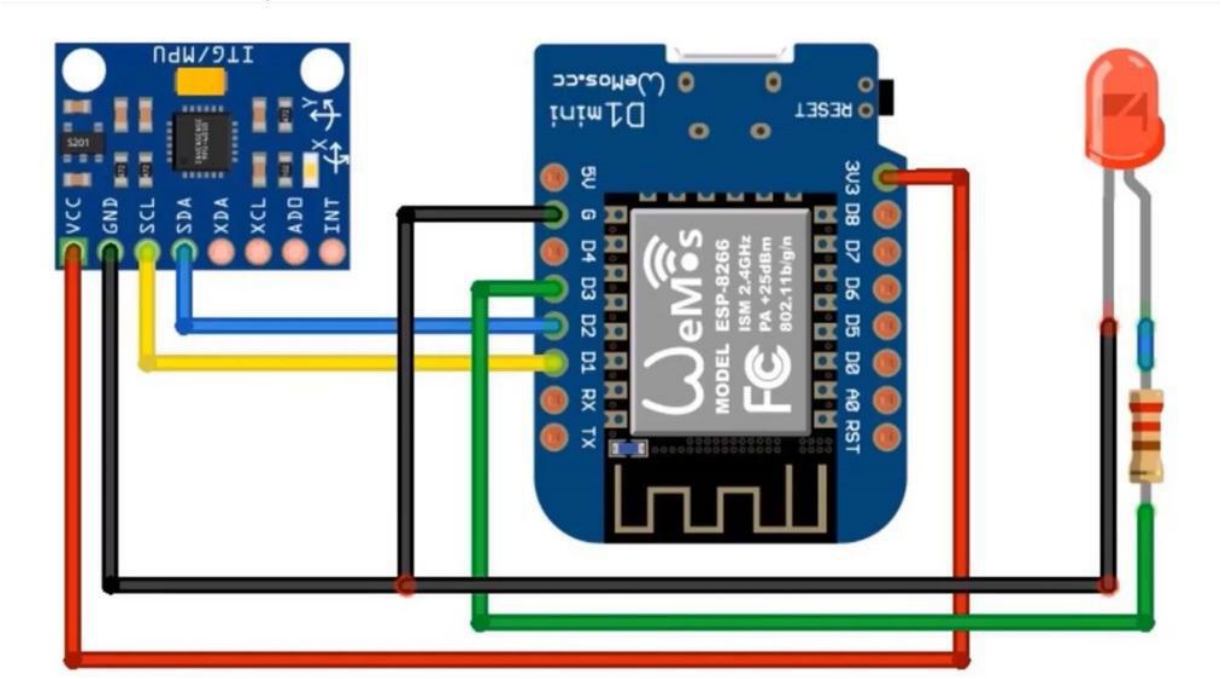
So, basically, this is the getting project, and we will learn about using the Arduino IoT cloud with ESP 8 to double 6. We will control an LED from the IoT cloud. And we will also send the sensor data to the dashboard and visualize it in various beautiful widgets. So let's begin with this awesome tutorial. This project is a small project by my favorite PCB manufacturer company called NextPCB.



They offer PCB board and PCB assembly services at the lowest affordable price. You can get trial PCB, 2 layer PCB, and 4 layer PCB with free PCB Shipping services up to a costly time of 24 hours. There is good news. That is next PCV has acquired Kiki PCV. You can use a Kiki PCV account to log into the next PCV and directly place an order. For the order placed in Kiki, PCB will arrange a dedicated salesperson for you to follow-up and communicate with you.

For the current orders you place with Kiki PCV, next PCV will take all over them. There is another great news. You can get up to 30% off for the PCB offer and up to 20% off for the PCBA offer. You can check the activity rules to learn more about this. Before getting started with the Arduino IoT cloud, we need some hardware.

The first component that we need is ESPN 8 to double 6 based boards. I selected the Wemos d1 mini board for this purpose. You can use the node MCU board as well. Then we will need a sensor For the demo, I'm using an MPU6050 gyroscope as a repeater sensor that can measure tilt angle. A 5 LED of any color can be used to control the LED from the cloud dashboard.

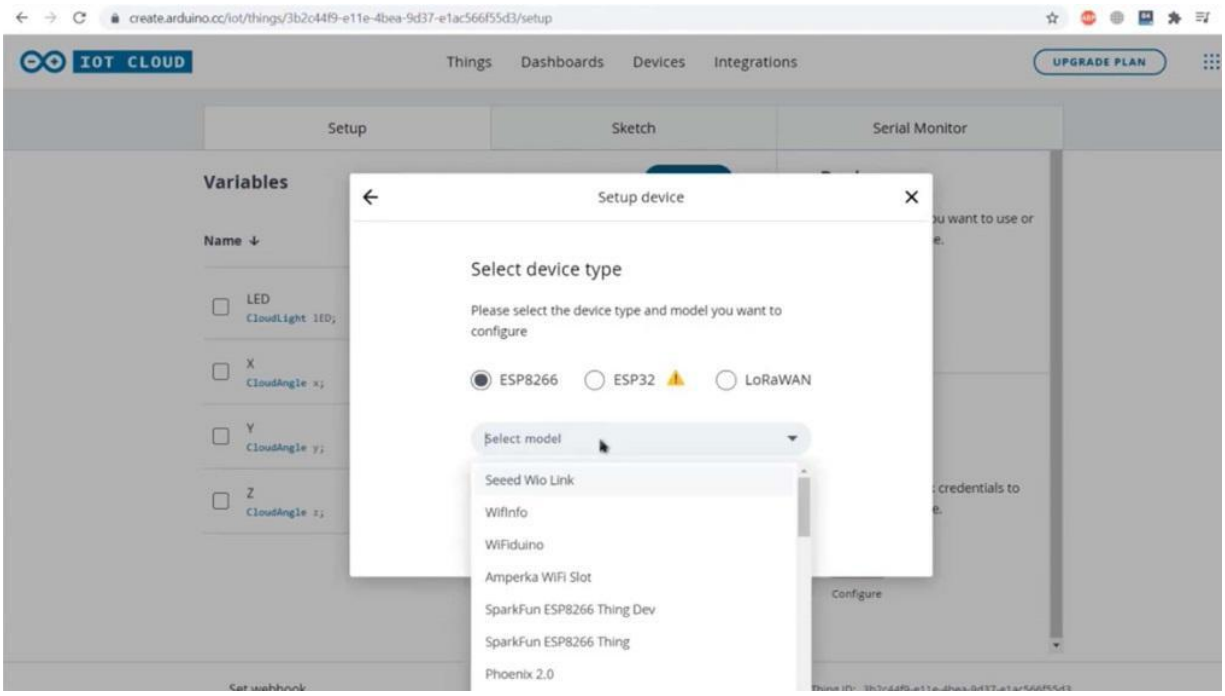


Similarly, a 220 ohm 330 ohm for using an LED. Make a simple connection as shown in emails here. Connect the MPU 6050 to the I Square C pin and LED to any digital pin. Now, let's set up the Arduino IoT cloud dashboard. To do that, visit the store audio link first.

You need to create an audio account and simply sign in. I have already done that. From here, click on IoT cloud. So here is the dashboard where we need to set up everything. Click on create thing now.

You can see here the setup section as well as the sketch and serial monitor section. From here, click on add variable. Give it any name. For example, for controlling an LED, I will give it the name LED. For the variable type, select anything like light as we need to send the 1 and 0 commands.

As you can see, a variable is already declared as a cloud light LED. You can assign the variable permission as either read write or read only. So here for LEDs, I assign a read and write function. Finally, click on the add variable. So a variable has been created successfully here.



Now, let's add the variable for MPU 6050 as we will be measuring the tilt angle. So, click on add variable and give it any name like x. And in the variable list, select angle. So when automatic declaration is assigned here, since we are reading the sensor value only, a sign the permission as read only. And finally, click on add variable.

So, the second variable for angle has been added. Let's assign another 2 variables for the y engine angles as well. Now, we need to set up the device as well as the network. To do that, select the device. Since ESP 8 to double 6 is a third party device, we will select a third party device.

From this list, select ESP 8 to double 6 boards. And from the model, select your ESP 8 to double 6 type. For example, you can either select a node MCU board or Wemos D1 mini board. Click on continue, and then name the device anything you would like. So here, a device ID and a secret key has been generated.

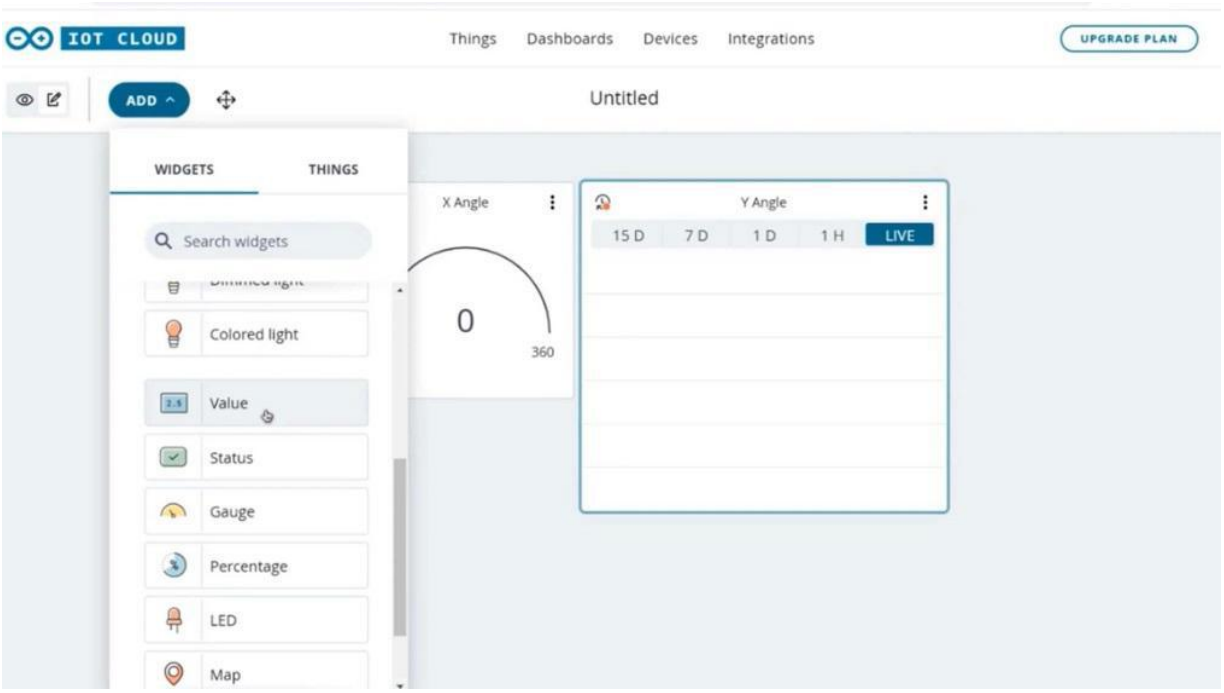
These are very important wire parameters encoding. You need to copy it or simply click on download the PDF. So here, all the information with the device ID and secret key has been downloaded.

Go back to the dashboard and click on continue so you are finally done here. Now, we also need to configure the network.

So, click here. enter the wifi access ID and then the wifi password. Similarly, put your secret key here that you downloaded earlier. Then, click on save. So, everything is set now.

Now, let's go to the dashboard to set up the widgets. Click on the build dashboard here. Click on this edit sign here. Click on add here. From this widget list, Select switch as we want to turn on and off the LED.

Give any name to the switch. The link is available here. So, LEDs appear here. So select it and then click on the link variable. Similarly, add another widget for X angle.



This time, I will select gauze from the widget list. Give Wixit any name. Then select a linked variable. This time, select acts. From the value, range select the minimum and maximum angle.

The angle range is from 0 to 360. so I will assign the same and click on done. We also have to create a widget for y and z angles. For my angle, I would select a chart. Again, name the chart anything you like.

Assign a linked variable as y and click undone. Do the same thing for the jet angle. From the widget, select anything like a numeric display. Again, assign a variable and click on done. So, our website display dashboard is ready now.

Go to the things again. All the variables are here. Now, click on the sketch tab. So a default escapes based on different variables has been created. There are some header files defined here and parameters are set for the LED.

So let's modify the code now. I will click on the open full editor as it's easy to edit the code here. So on the LED chains function, we need to do something to turn on and off the LED. So I would assign a condition here Like if LED is 1, then the LED should turn on. The LED is connected to GPIO pin 0.

In the else condition, the LED should turn off. We also need to define the LED pin as the output pin in the setup section. Now, we need to add the code for MPU 6050 as well. Here is a sample code that is used to measure a tilt angle. What I would do is I would copy all these codes and add in the necessary places of the code.

But before that, let's see the library as we need a wire library. For that, click on the library. In the list, There are thousands of available libraries which have already been added to the list. As you can see, it's section has 10, 15 libraries and there are more than 82 pages. You just don't need to worry about the library as they are already available here.

Back to the coding, I will create a function called mpu6050 read and add it to the loop function in the code. Then, I will copy the sample

code for the MPU 6050 tails angle and add it to the code editor. So here you can see our final code is ready. There are variable declarations and so many pin assignments. If you check the other tab of this code, you can see some instructions and bill of materials on the read me tab.

In the Think Properties lab, there are other predefined variables. And on the secret tab, the Wi Fi SSID, password, and secret key are defined. Alright. Let's compile the code now. To do that, select the board from this list first.

No port appears right now as a driver from the browser is not installed here. So, here, you will get an instruction to install the agent. So, click here. It will take you to the page to download the EXE file. So, I have a 64 bit window, and I will download it now.

So, click on next, and then again, next to install the driver. Assign all the permission so that the browser can detect the port. Once done, go back to the editor. So, our hardware setup is here on the breadboard. It is connected as per the circuit diagram shown earlier.

From this list, select the board. I am using a Wemos D1 mini board. Similarly, select the port as well. For me, COMP 6 appears here. Then click okay.

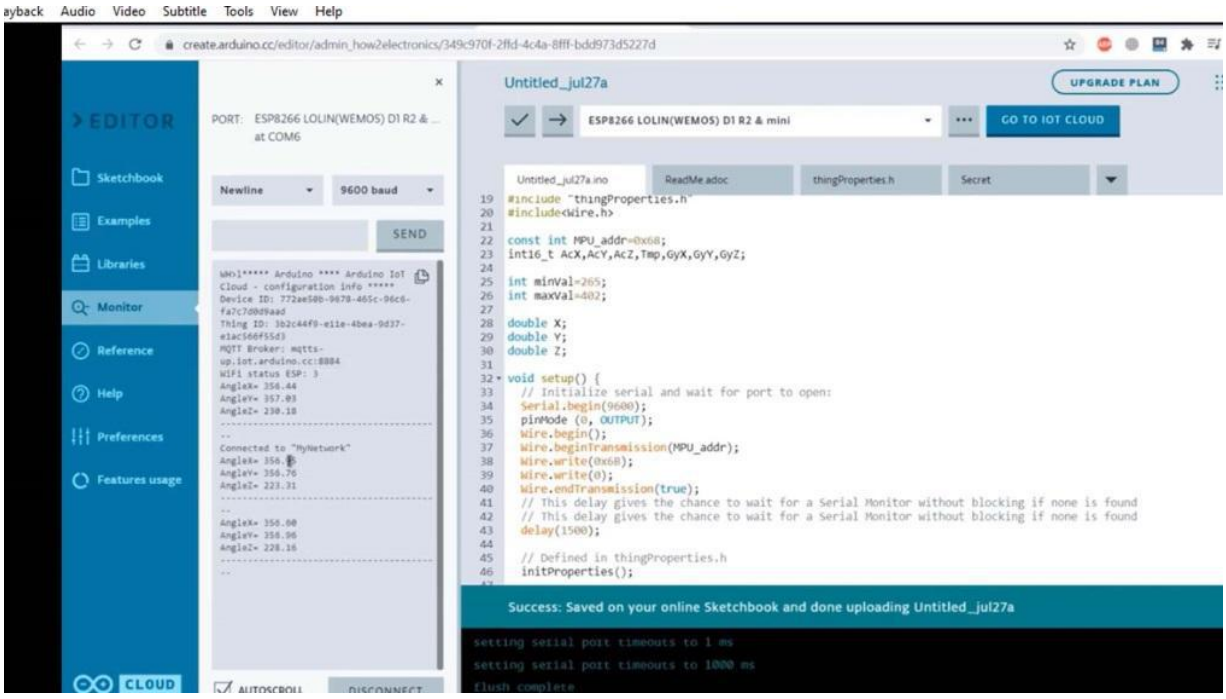
Now you can compile and upload the code. Now click on the upload button here. It will take some time to upload the code. Once uploading is done, you can see the status on the console window here. Now you can open the serial monitor here.

So here on the serial monitor, you can see the MQTT connection is established and the device is connected to the wifi network. It is also displaying the tilt angle for the x, y, and z axis. Now, go back to the dashboard to check the exit. As you can see, all the data is uploaded here after a certain interval. To check the working, tilt the sensor and check the value on the serial widget screen.

Similarly to turn on and off the LED, send the command from here. So as you can see, everything is working fine here. So this is how you can use the Arduino IoT cloud for your project. Go back to the sketch part, Here you can see the option to export this sketch as well. So, download this sketch, a zip file will be downloaded.

Expect the G file, open the Audino main file using Arduino IDE. So, here you can see the audio sketch with 3 different tabs. You need to install the audio IOT cloud library to the audio ID. to use the code from here. In this tab, insert the wifi access ID password and also the secret key.

You can simply compile the code and then upload it from here. Finally, let's check the mobile dashboard for the project. To do that, visit the Play Store or go to the iOS App Store. Here you can see the app. Install the app from here.



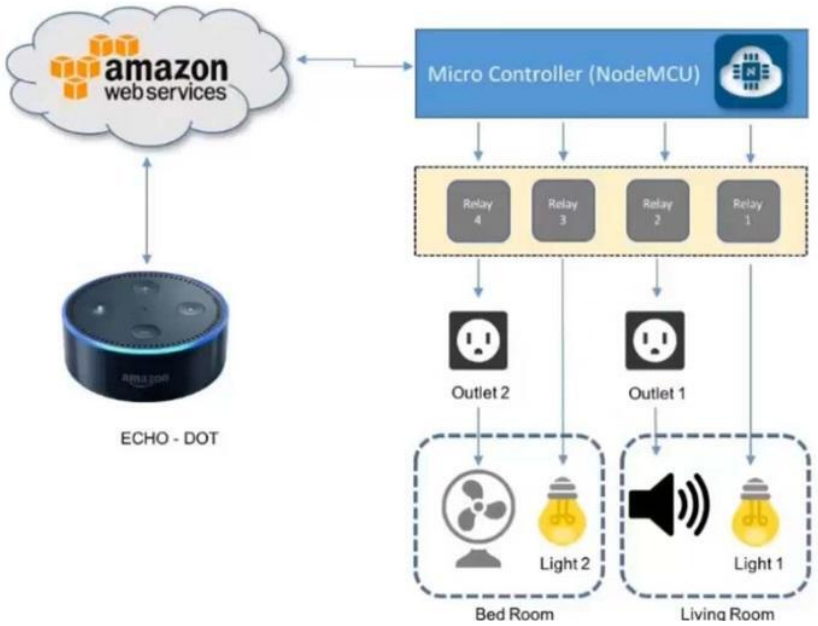
Open the app now and sign in using the same username and password. Go to the dashboard now. So, here you can see how well all the data is uploaded to this smartphone dashboard. The best part is you don't need to set up a separate dashboard for mobile phones

as in the blink app. I hope this tutorial has cleared your doubt about the audio IOT cloud.

VOICE BASED HOME AUTOMATION WITH NODEMCU AND ALEXA

In this project, we'll learn about home automation with voice with no MCU and Alexa Eko.

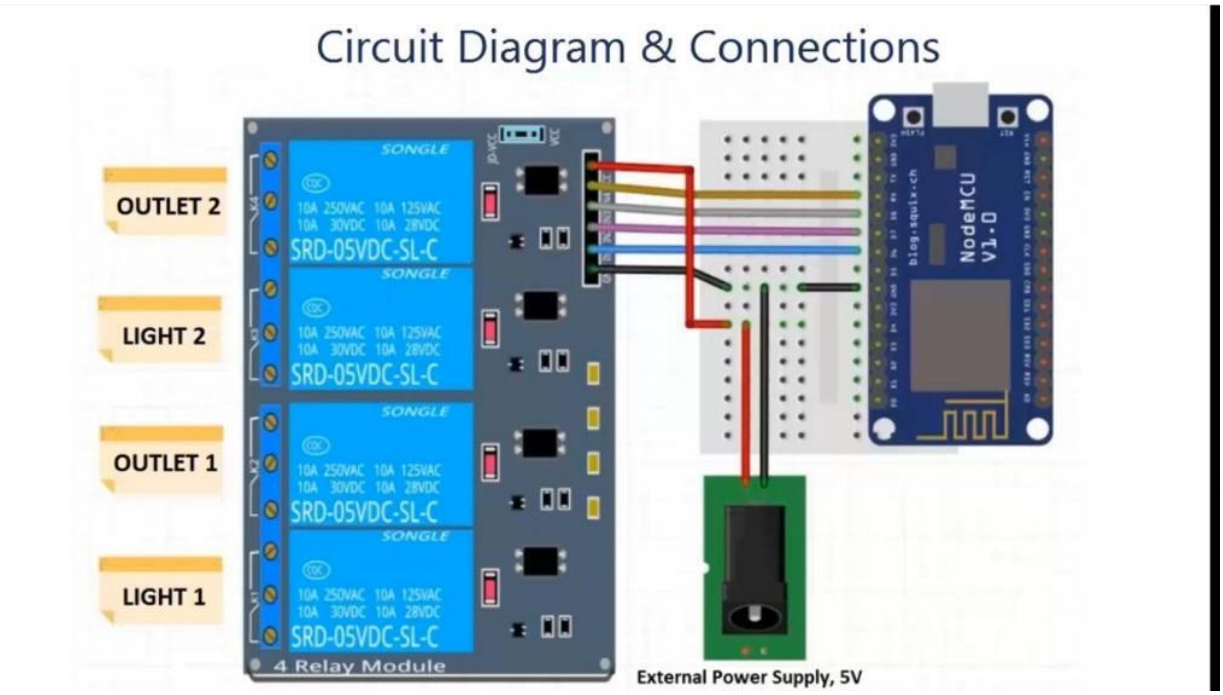
He uses Pamax ESP. Now, Phamax Exp is a library for E SP 8266 best devices that emulates pilots, who lights, and allows you to control them using this protocol. In particular from Alexa power devices like that. So let's learn the details about it and the components required. So we do not need MCU E SP at 26620 model.



We need a 4 channel relay module. 2nd generation from. We need 2 X Mini breadboards. some of the female male female DuPont cables. And, finally, a 5 volt power supply that is an adapter or any sort of battery.

So this is an Alexa module that is capable of voice integration, music playback, making to do listening, setting alarms, stinging podcasts, playing audio books, and providing weather traffic and real time information. Alexa can also be used to control several smart devices using itself as a home automation hub. It will use this product equal to that allows users to activate the devices using a white word such as Alexa or computer So this is the circuit diagram. We are just using a 4 channel delay that is connected to node MCU with 5 volt power supply. that will control the 4 relay home automation using Voice command. So this is the program.

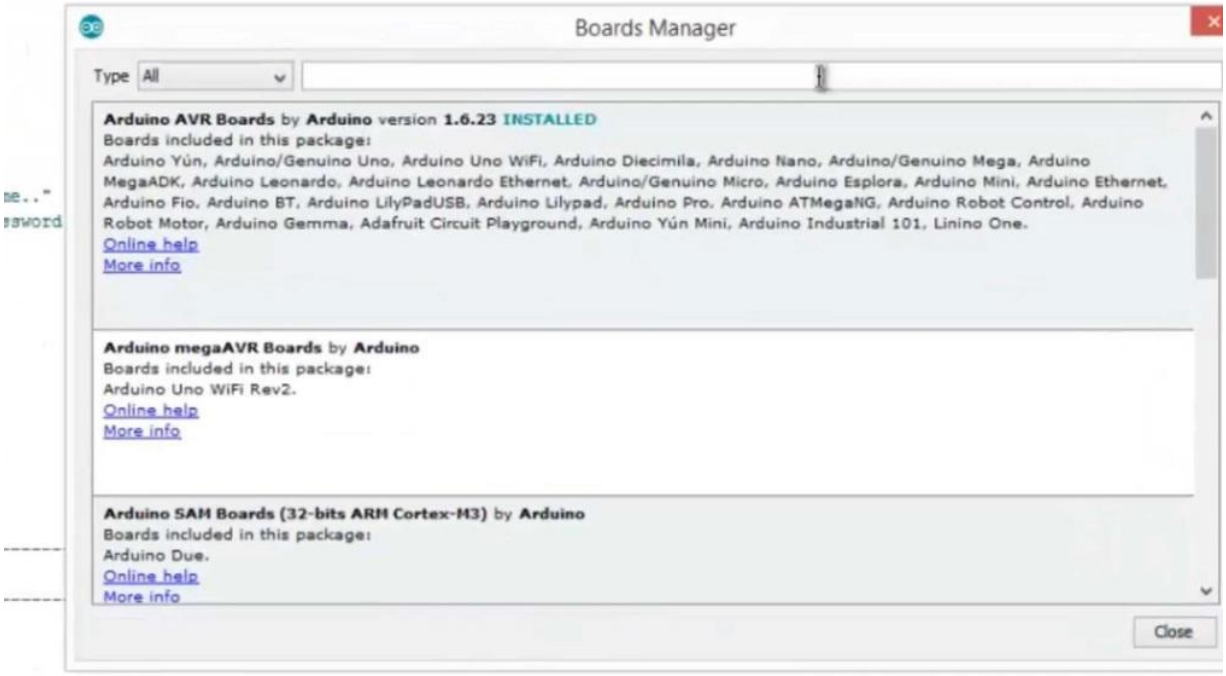
Now the program section contains the header file for ESP 32 and wi fi dot h as well as e s p 8266 wifi. And it also includes powermax esp.s. But before going to upload it directly, we need to add some of the files. Now what are those files? So first, go to the preference and we need to add a board because this ID no ID doesn't have a board currently.



So at this additional board manager URL, you need to add a link that the link is given in the description below. Copy this. The link from the below and directly hit enter. So this is the link. Now why is this link required?

Now we'll learn later. First, copy this link and paste it over here. Now, click on okay. Now go to the tools. Now you need to select the node cubert.

but there is no node MCU board or any board that is related to IOT. So First, we need to add the board. That is why the link was added there so that we can download the board. So go to the board manager. Just enter ESP 8266.



So this is the e s p a 2 success community, and I have already installed the 2.3.0 version over here. If you haven't installed any of this person, you can just edit from here. So Just edit or install it. It is about 150 MB. So it will take so much time to download.

Now after that, You can see the word section. So at this section, you can see there are so many words that are already added. There is a related IOT board like node MCU. So select no MCU 1.0. CPU frequency, flash file, and upload speed limit as it is.

Now you need to add 3 different libraries. I have already given the link in the description below. So download the zip file. Now, extract the zip file. So after extracting, you'll get 3 different library files.

So you need to add. So go to sketch. It has a deep library file. View the link where the file is Downloaded. So one by one, you have to add all these 3 libraries.

So I have already added a library. So it is saying it already exists. So I have added all the three libraries here. Now it is almost done, which

means that it is completed. Now you know, to edit the Wi Fi F society.

It's given any name, whatever you like. For example, I'll give the name of my channel. That is how electronics work. Set your WiFi password. For example, I am setting it as 1, 3, 4, 5, 7, 8, 9, 8, 3, 0, as well.

Do whatever you like. Now upload the same file in your load MCU after connecting it. save the file. Save the file with a new name. Now you can see the compiling is going on.

So finally The code is compiled. So compiling is done. So this is what you need to do. Now let's see the demonstration of the project. Once you upload, you will see the demonstration.

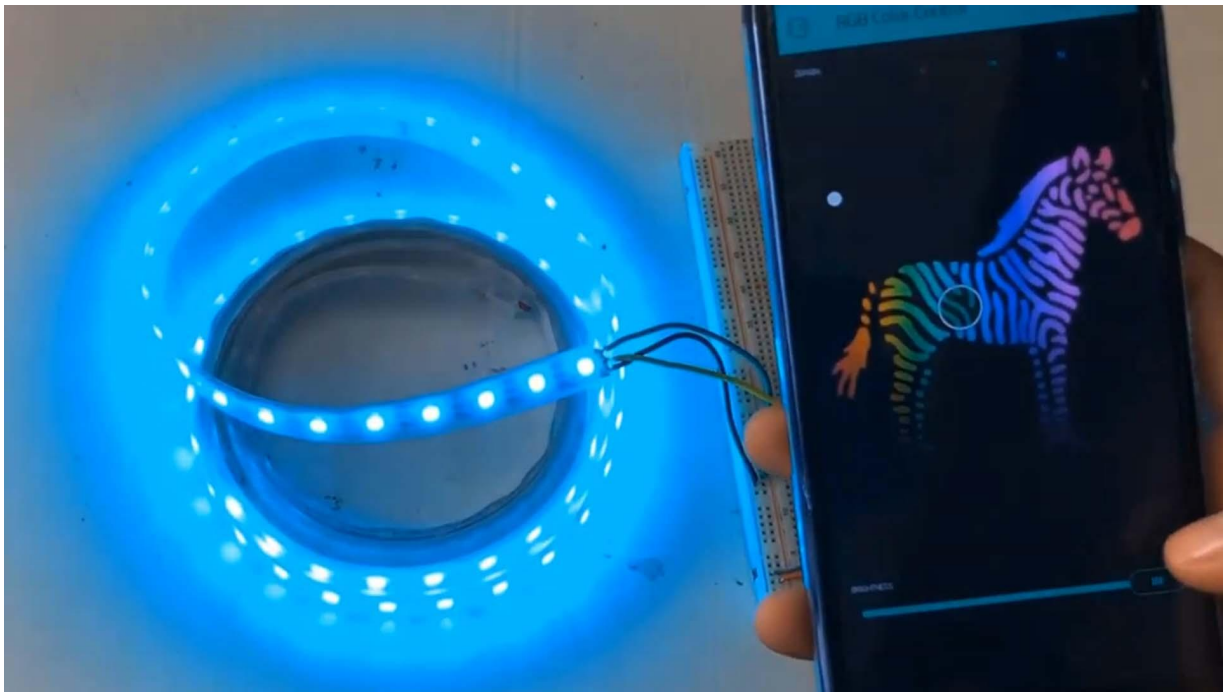
Computer. Turn on light 1. Computer. Turn on light 2. Computer.

Turn on outlet 1. Computer. Turn off light 1. Computer. Turn off all devices.

Computer. Turn on the bedroom. Computer. Turn off the bedroom.

WS2812B NEOPIXEL
RGB LED STRIP
CONTROL VIA BLYNK

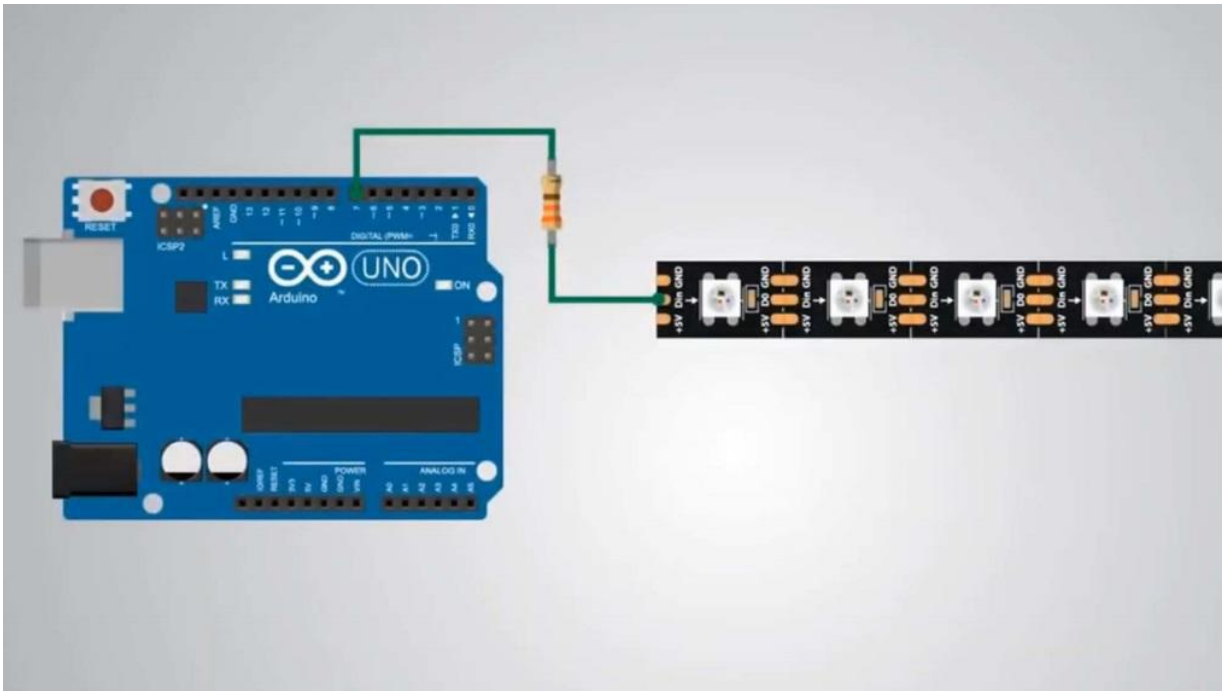
We will learn how to control individually w s 28 2 will be LED straight Using node MCU WiFi module and blink application, the WOS 2812b is an addressable RSV latest trip That is extremely flexible and easy to use. The best part about this LED strip is each layer of that strip can be controlled separately by using a microcontroller. Using a wifi module like ESPN 8266, the late street can be controlled.



There is no need for any external driver IC. The system can be connected to a Wi -Fi network. Then you can configure the blank application on your smartphone and control the brightness and color from any parts of the world. As you can see here, I am able to create

various colors by mixing red, green, and blue colors. Even the brightness can be controlled by a sliding slider.

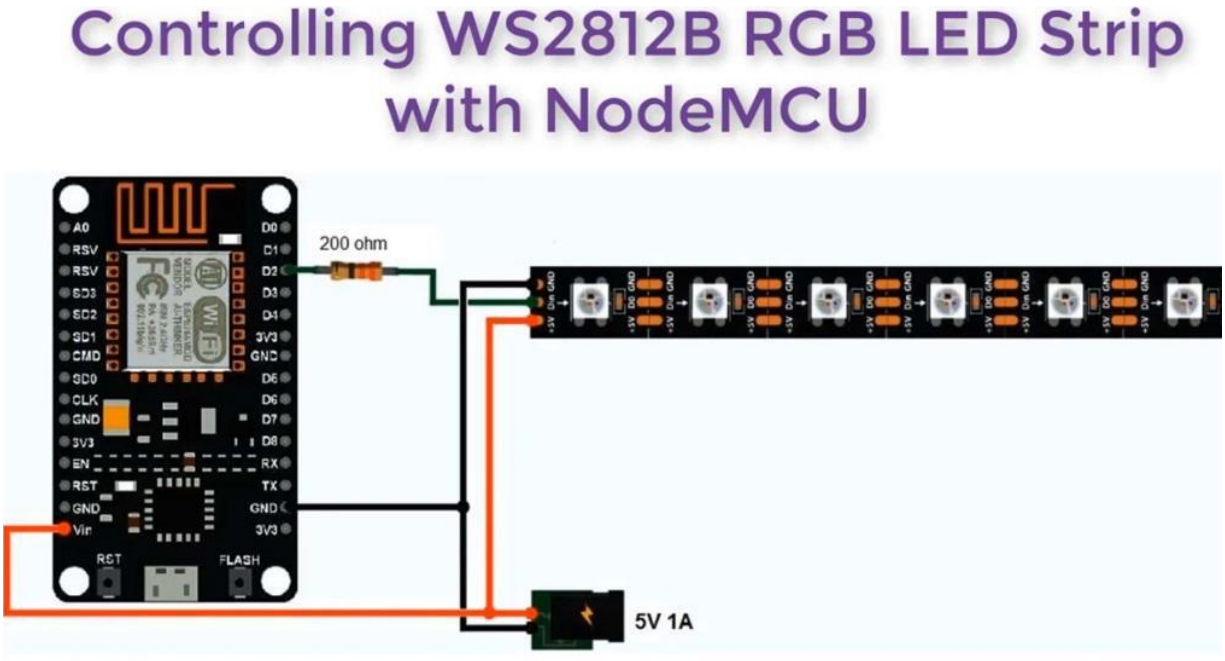
So without any delay, let's get started. The project is sponsored by Nick visibility. Currently, the next recipe is offering only $7 for assembly prototype orders. The prototype includes 1 to 5 pieces with no company limit with free stencil and SMT plus DIP service. Next PC has the latest and best technology for SMT services.



Components are mounted by placing them directly onto the PCV service. They have capabilities to assemble the as empty prototype PCBs in small production runs with manual and automated assembly production processes, including single or double sided component insertion with any package assembly. Let's start by taking a closer look at the late strip. It consists of 3550, 50, 50 are cabinets in which the very compact of less 28 will be late driver IC is integrated. Depending on the intensity of the 3 individual red, green, and blue lights, we can stimulate any color we want.

What's great about this lates is that we can control even the entire latest strip with just a single pin from our Arjuno board. Each light

has 3 connectors at each end, 2 for the power and 1 for the data. The arrow indicates the data flow direction. The data output path of the previous lead is connected to the data input path of the next lead. to any size we want, as well as distance the lates using some wires.

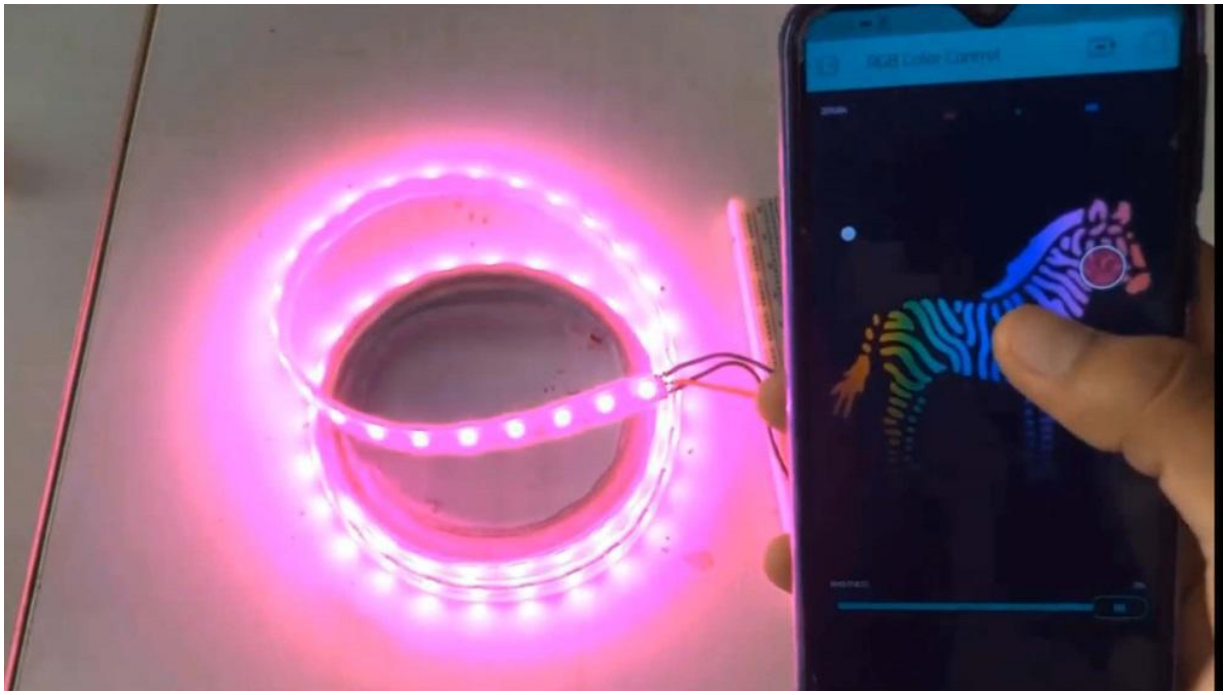


As for the powering, they work on fivefold and each red, green, and blue draws around 20,000,000 pairs. So, the total of 60,000,000 pairs for each leg at full brightness. Note that where the arduino or nodemcu is powered through USB, the 5 volts pin can handle only around 400 milliampere. and bring power using the barrel power connector. The 5 volts pin can handle around 900 milliampere.

The connection is the same for no damage to your board as well. The digital input pin can be connected to any desktop pin of known MCU through a 200 ohm resistor. The LED strip is powered by 5 fold to what ampere external power supply. Now, let's just configure the blink application. So, download the blink application from Play Store Then, log in your email ID and password.

Create a new project and rename it something like RCV color control. Now, let me quickly complete the setup process. You can follow my project to complete the setup guide. Once the setup is completed, you can request for the authentication token. So, the authentication key can be sent to your mail.

Copy the token and replace the arrow in this code part. You also need to change your wifi access ID and password. You need a blank library and also the Fast LED library. Get the library from the library manager, or you can directly get it from the link in the description. So, that's all from the code set apart.



Now, you can select the new MCU board from the board manager and simply click and upload the code. Once the code is uploaded, and the modem suite gets connected to wifi. You are ready to control the new pixel and it is tried using Blink.