UNIVERSITY OF RWANDA: AFRICAN CENTER OF EXCELLENCE IN INTERNET OF THINGS

I. Team names:

No.	NAMES	Registration Number
1	MUGABO Asaph	220015226
2	MUKAMANA Jacqueline	220004047
3	IRAMBONA Oscar	220014217
4	NTIVUGURUZWA Jean de la Croix	220014132
5	INGABIRE Ariane	220014279

II. **Project Supervisor:** Professor Kayalvizhi Jayavel

Title: Smart car parking system

Objectives:

- Remote and real time identification of available parking slots

III. Requirement

1. Hardware requirement:

- Infra-Red sensors
- LED (Light Emitting Diode)
- Microcontrollers (NodeMCU ESP8266)

2. Software requirement:

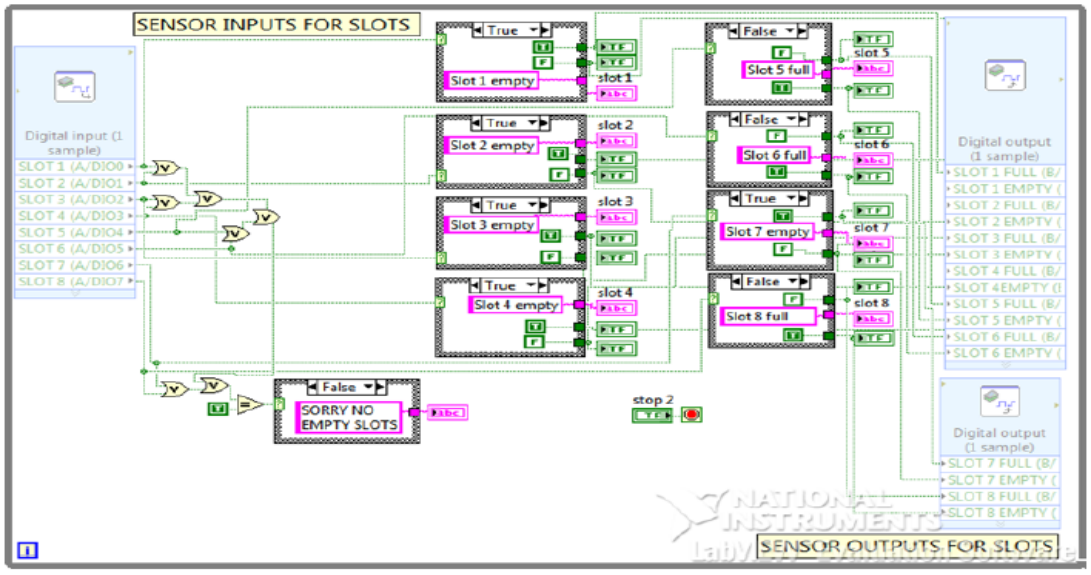
- Arduino IDE
- Firebase

IV. Methodology

Due to the proliferation in the number of vehicles on the road, traffic problems are bound to exist. This is due to the fact that the current transportation infrastructure and car park facility developed are unable to cope with the influx of vehicles on the road. To alleviate the aforementioned problems, the smart parking system has been developed. With the implementation of the smart parking system, patrons can easily locate and secure a vacant parking space at any car park deemed convenient to them. Vehicle ingress and egress are also made more convenient with the implementation of hassle free payment mechanism. With vehicle detection sensors aplenty on the market, the choices made may defer due to the different requirements in addition to the its pros and cons. Subsequently, the various sensor systems used in developing the systems in addition to the recent research and commercial system on the market are examined as vehicle detection plays a crucial role in the smart parking system.

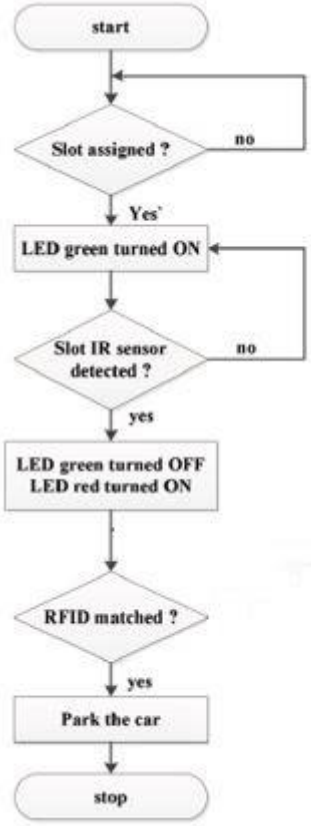
Proposed model and its experimental setup

Implementation of Smart Car Parking System



Front panel simulated result

FLOWCHART



How it works

The smart Car parking system is a system that is able to calculate and detect the presence of cars in the parking area. The data obtained in the parking area will be sent to the cloud database. The data will be used by Android application to inform the users where parking space they should go to.
The system consists of Node MCU, ESP8266, IR Sensor, and LED.

NodeMCU

Node MCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit) .The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits. The IR Sensor, are connected to the Node MCU microcontroller. The node MCU will do the processing.

ESP8266 provides WiFi connection to the Cloud. The **ESP8266** is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability

The LCD monitor is placed at the entrance/gate to notify the drivers the available and occupied parking area. Two LEDs are used only to provide information visually about the presence of the car. If there is no car in parking slot, then one LED will light up, and if the parking slot occupied, another LED will light up.

ESP8266 WiFi Module serves as a data sender medium from the microcontroller to the local server. ESP-8266 is used as a communication medium between local and Arduino servers.

Software side of the system

As per the above specified software requirements, we will need to use three software for the functionality of our system notably Arduino, Firebase and Blynk App

1. Setting up Arduino IDE
2. Setting up Firebase
3. Setting up Blynk app

Description for Each

1. **Setting up Arduino IDE**

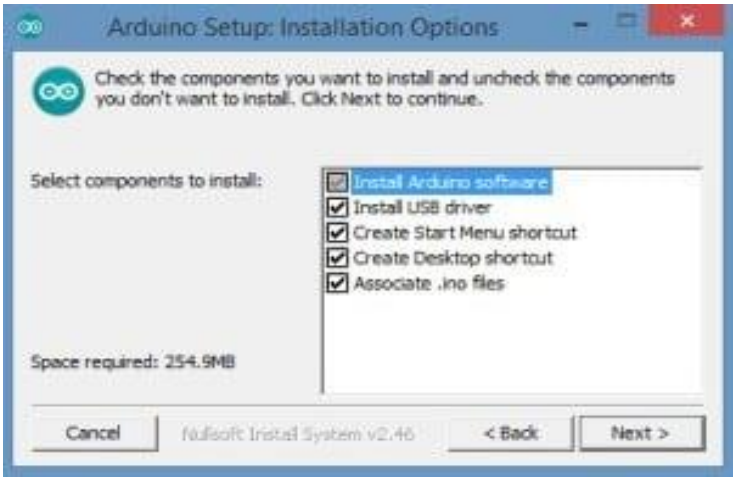
The Arduino is a powerful prototyping tool for many reasons, including its lack of a dedicated programmer, its wide range of available libraries, and the simplicity of its IDE.

1.1 Downloading and installing Arduino IDE software

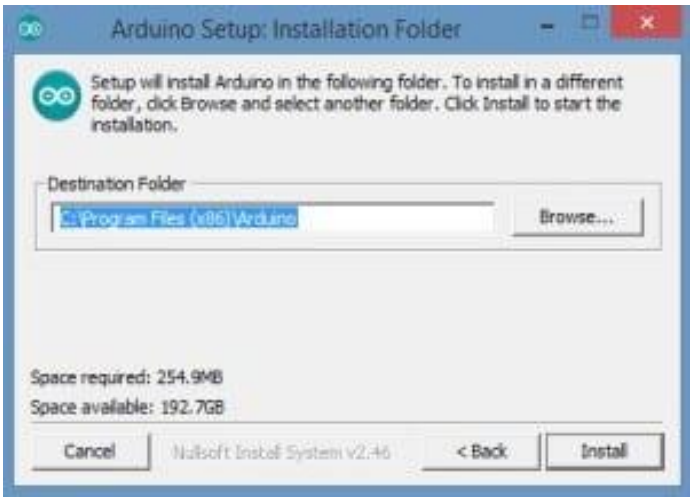
Below are the steps for installing the Arduino IDE software:

1. Visit http://www.arduino.cc/en/main/software to download the latest Arduino IDE version for your computer's operating system.
2. Save the .exe file to your hard drive.
3. Open the .exe file.
4. Click the button to agree to the licensing agreement:

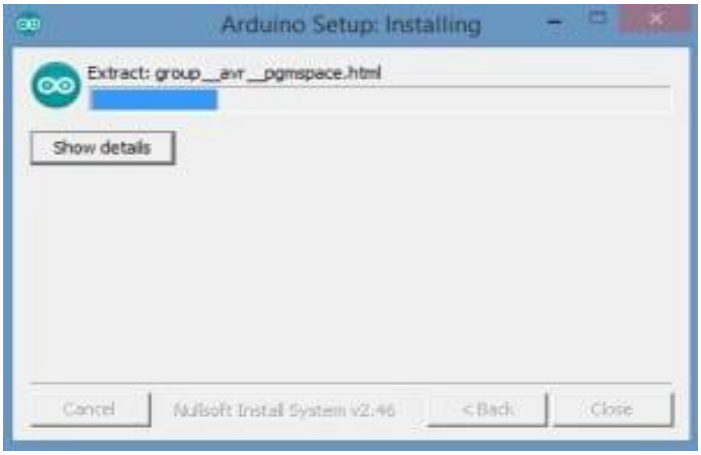
5. Decide which components to install, then click "Next":



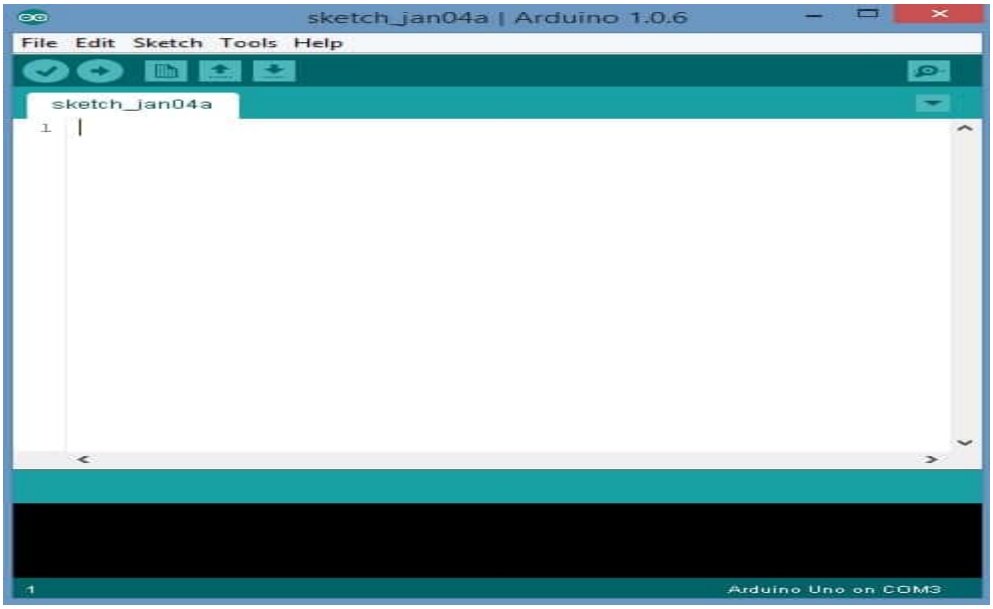
6. Select which folder to install the program to, then click "Install":



7. Wait for the program to finish installing, then click “Close”:



8. Now find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you’ll see the code editor:



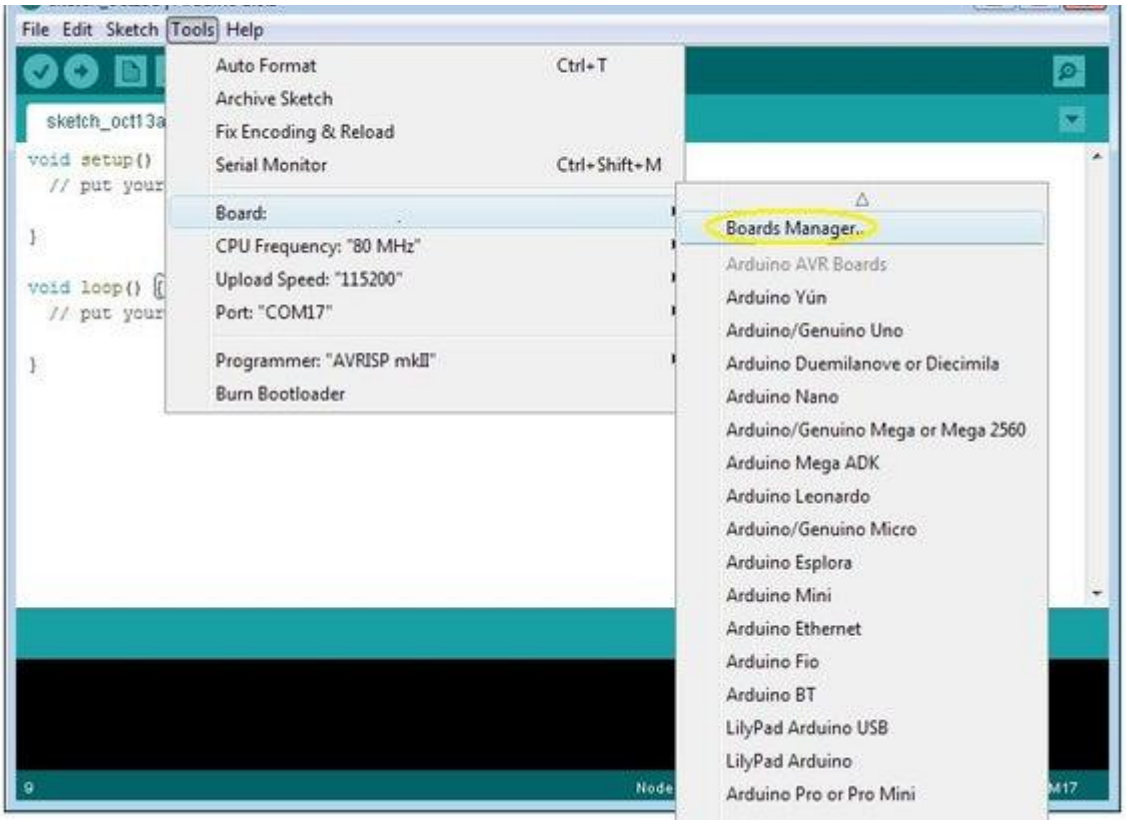
Arduino Software

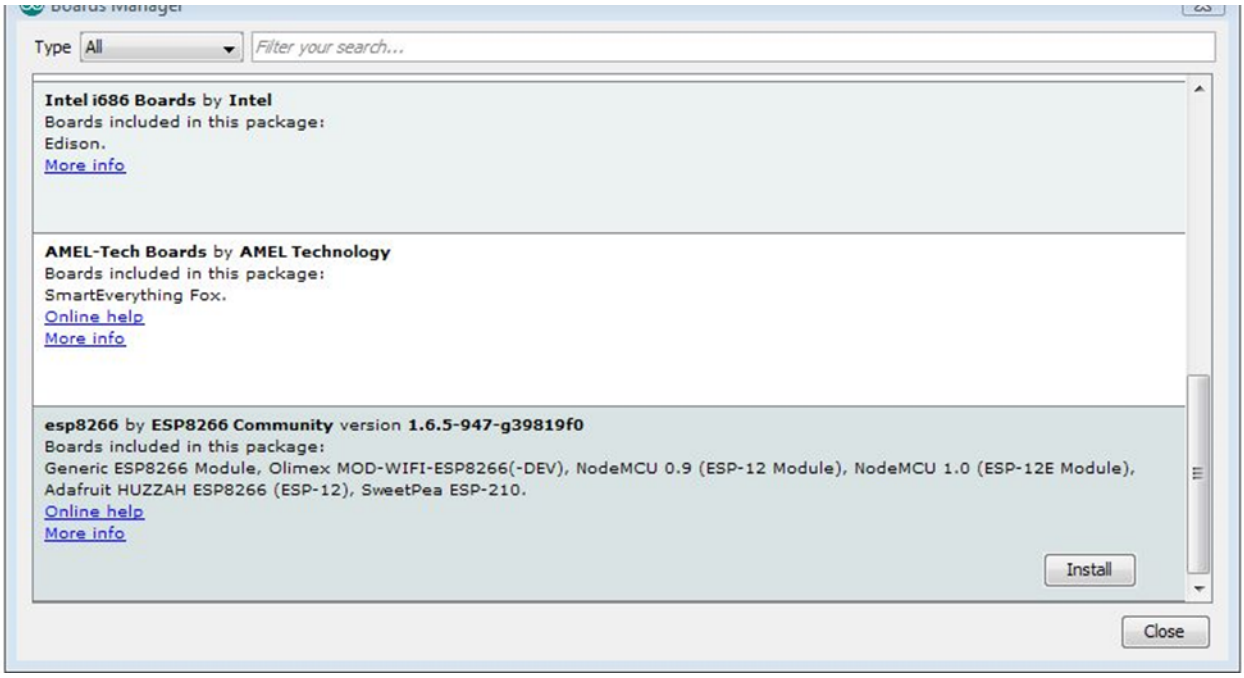
1.2 Configuring the Arduino IDE

You'll need to connect the board to the computer. This is done via a USB connection, we do not need to provide power to the Arduino, as the USB provides 5V up to 2A.The first step in setting up the ESP8266 Arduino IDE is to plug in the NODEMCU and check the Windows device Manager for an entry in the Ports section.

If you don't see the Node MCU showing up as shown below then you will need to install drivers.

Step 1: Installing the ESP8266 Board via the Board Manager

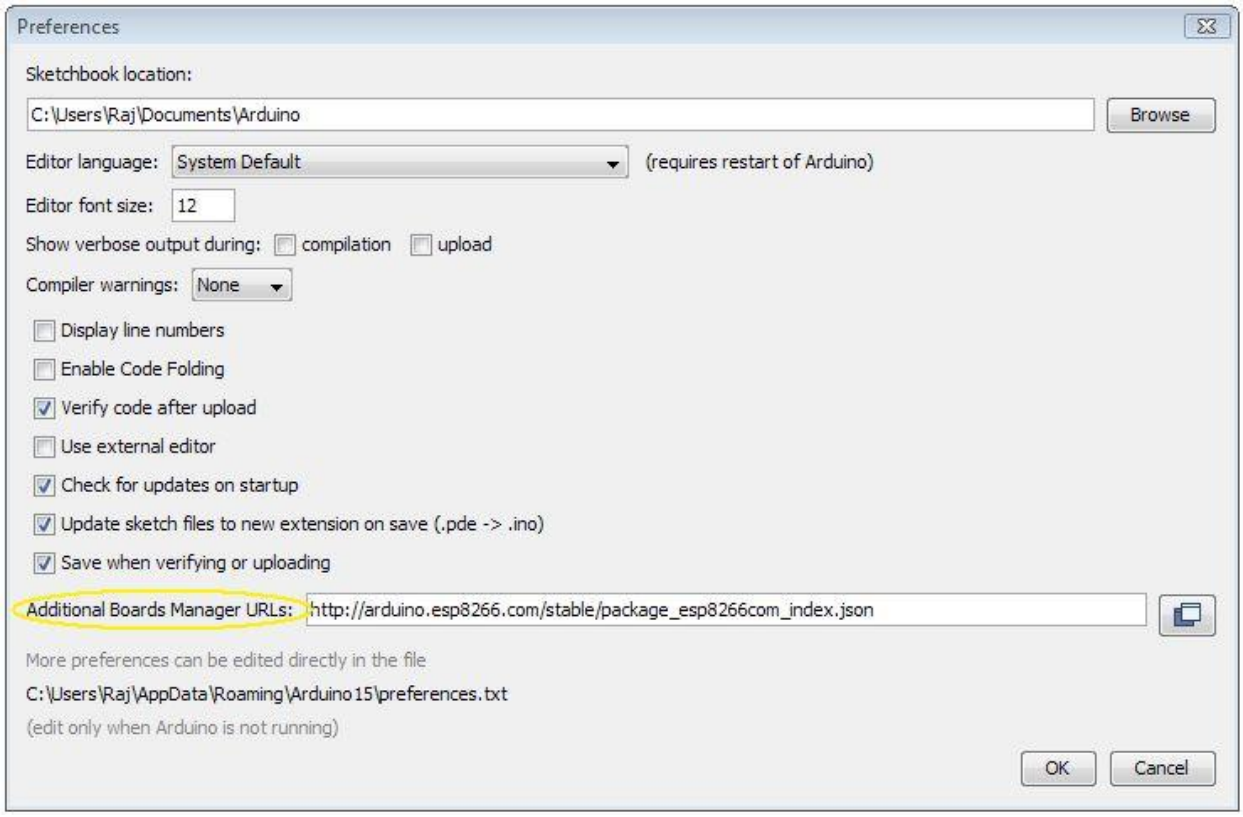




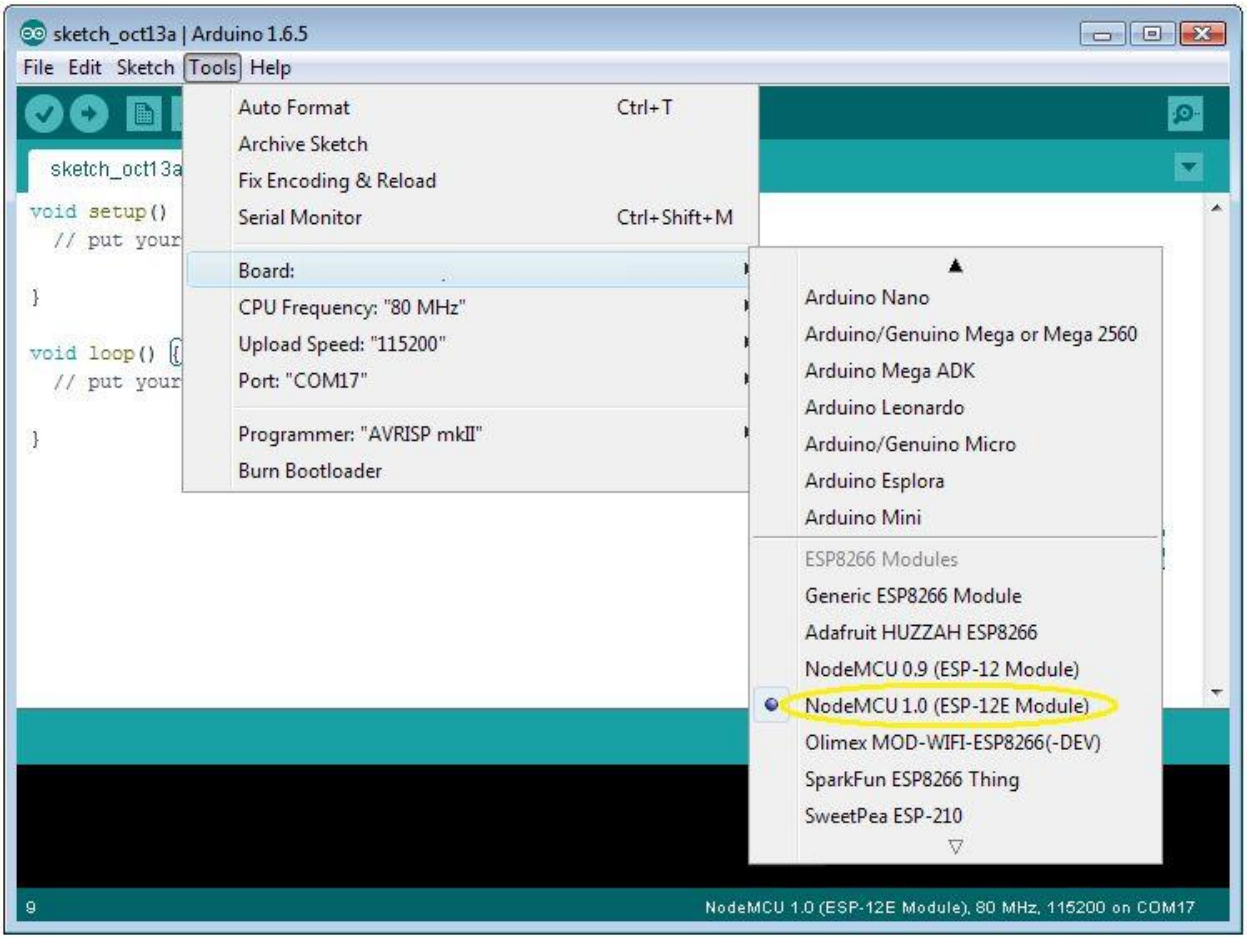
e install I shut the Arduino an

Step 2: Telling Arduino Where to Find the ESP8266-E12 Library

The ESP8266-E12 board can be added to the Arduino software/IDE by installing the necessary libraries.

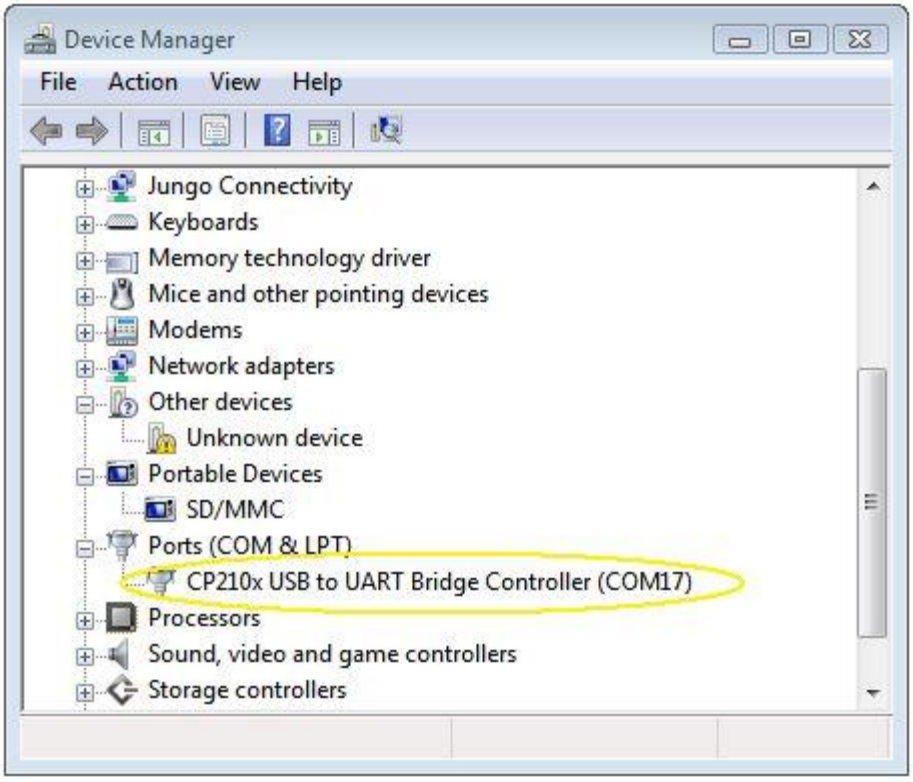


Step 3: Selecting the ESP8266-E12 Board



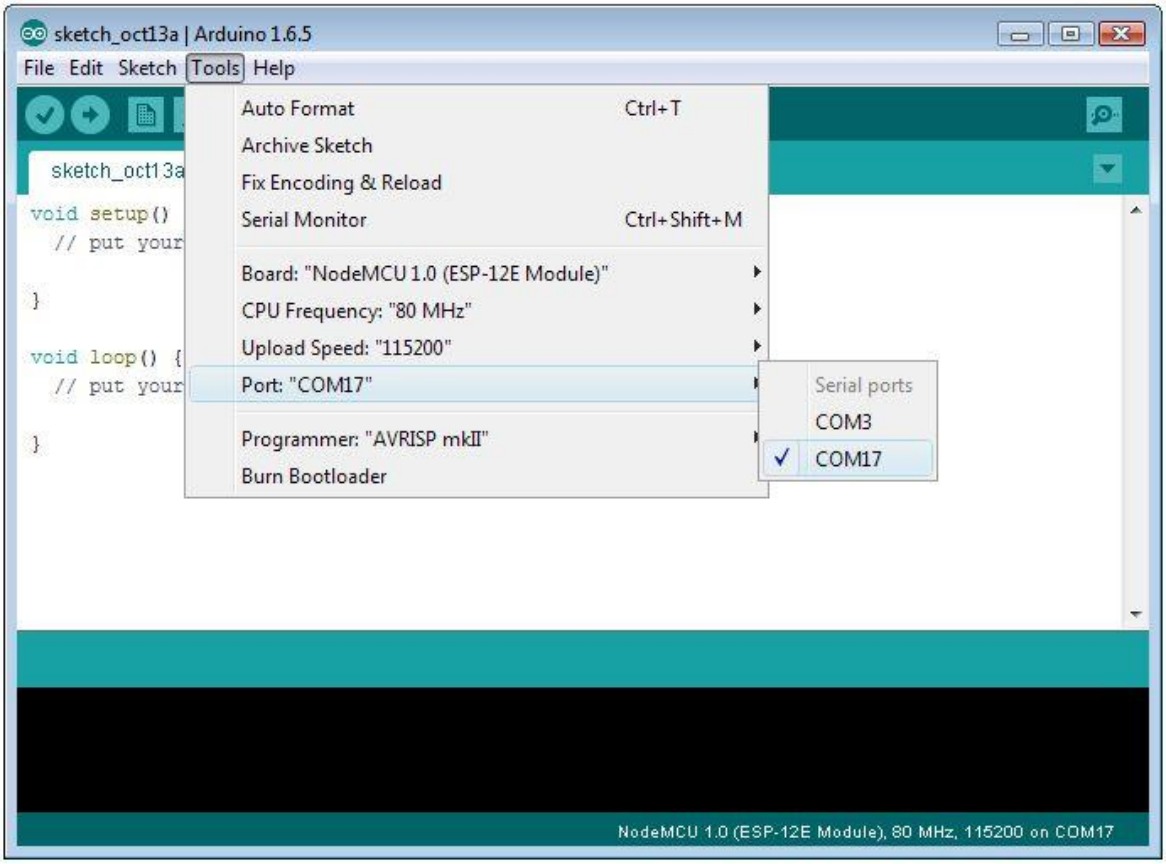
Step 4: Determining the COM Port ESP8266-12E Shows Up On

When the Arduino is connected, the operating system should recognize the board as a generic COM port. Once it's recognized, we will need to find out what port number it has been assigned. The easiest way to do this is to type "device manager" into Windows Search and select Device Manager when it shows.



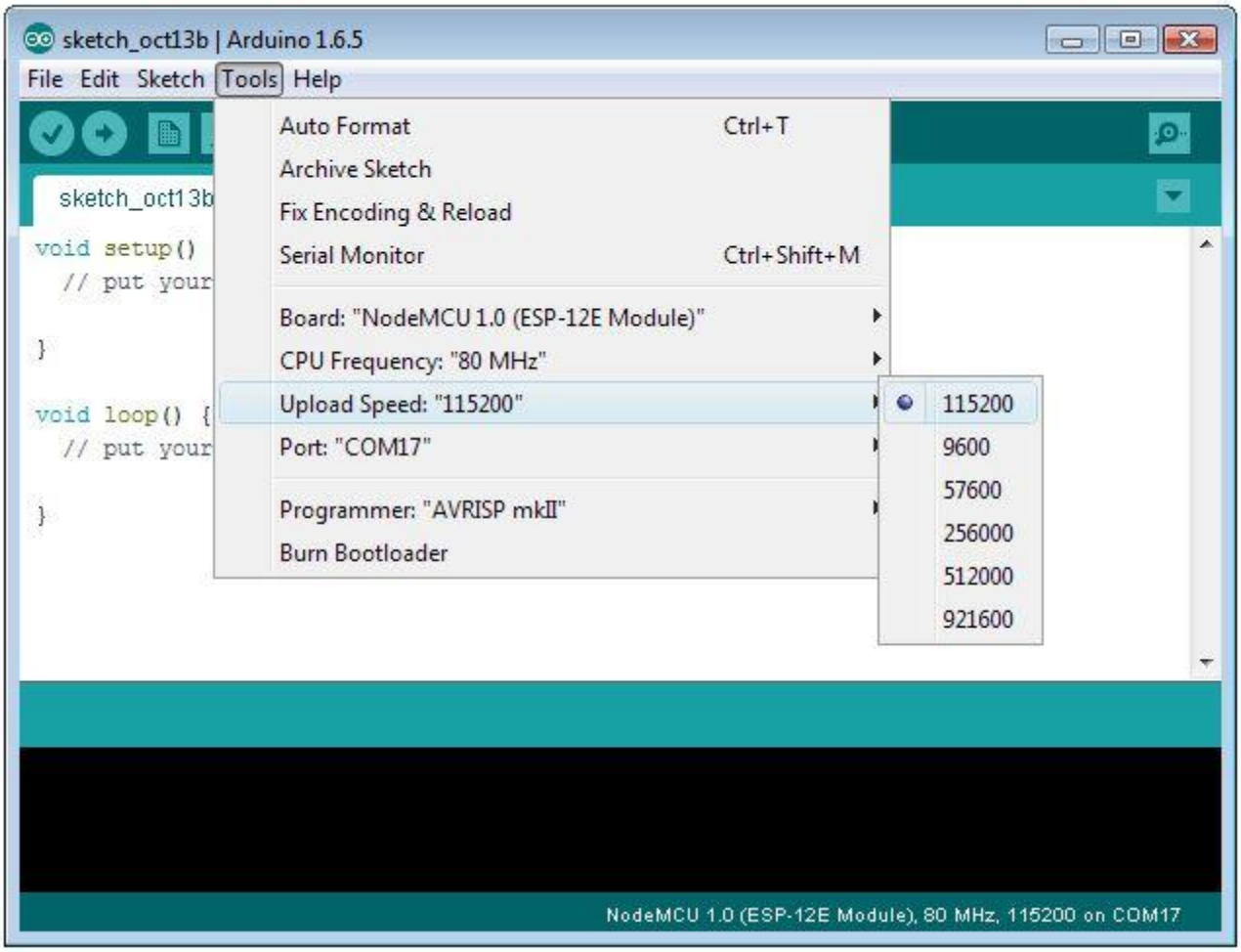
If not connected connect the ESP8266-E12 board to your computer. Give it a couple of minutes to settle down. Go to Device Manager and click on *"Ports (COM & LPT)"*. There you will find the port number that your ESP8266-12E is connected on. In our case it is COM17.

Step 5: Configuring COM Port



Step 6: Configuring COM Port Speed

To select the speed of communication between the ESP8266-12E and the computer.
Go to "*Tools | Upload" Speed:* and select 115200. We are ready to upload



Step7: Uploading and compilation of code

It's time to verify and upload the code. The verify stage checks the code for errors, then compiles the ready-for-uploading code to the Arduino. The upload stage actually takes the binary data, which was created from the code, and uploads it to the Arduino via the serial port. To verify and compile the code, press the check mark button in the upper left window.

The "Verify" button will compile the Arduino code.

2. Setting up Firebase

Firebase is a mobile and web application development platform based on Google.

A. Creation of an account

If it is the First time you are going to create an account, Search Firebase Console into search engine called google.com ,then go to the Firebase website and sign up for an account.

B. Login into your Account.

Once, you have already an account in Firebase, You have to login using your Gmail account in order to access easily.

C. Creation of a new project

When you log in, you should go to the Firebase console where you should manage all your projects that you have there.

Steps followed to create a new project are as follows:

- Click on **CREATE NEW PROJECT** button.

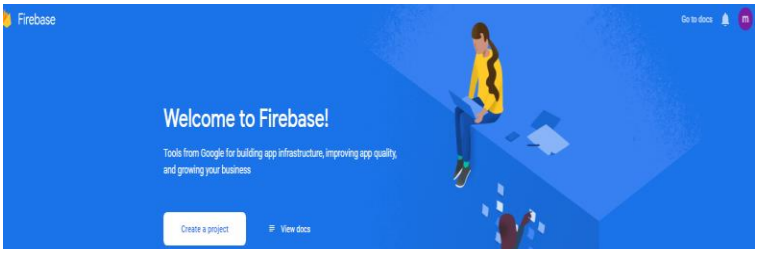


Figure 1: This Screenshot shows where to click when you are going to create a new project.

- **Enter name of the Project, then Click on Continue**

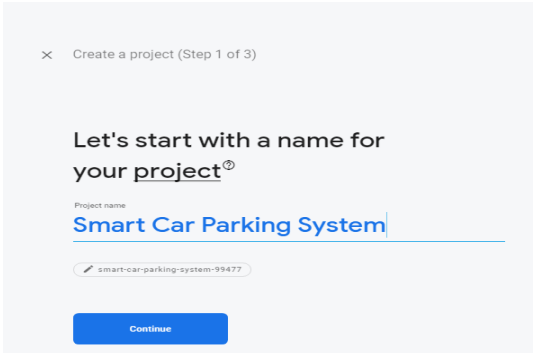
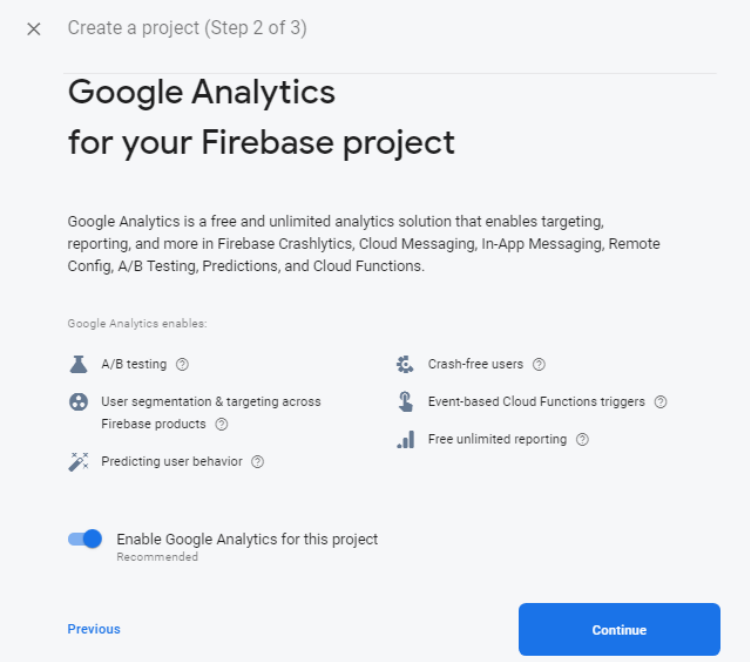


Figure 2: This Screenshot allow us to enter the name of the project

- **Click on Continue**



- **Click on Create Project.**

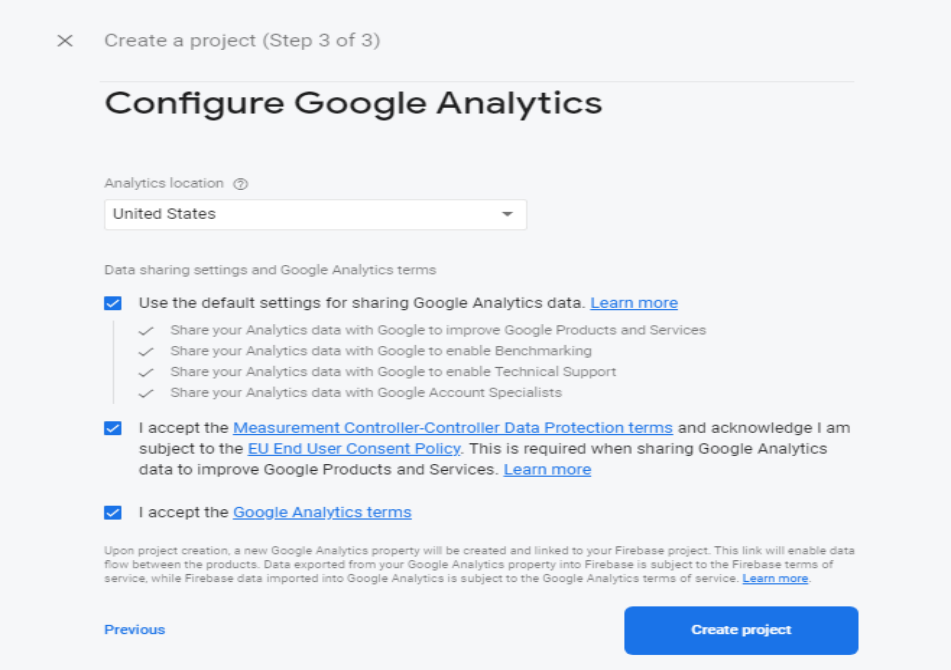


Figure 3: This screenshot indicate where the create project button

- **The created Project is loaded to finish.**

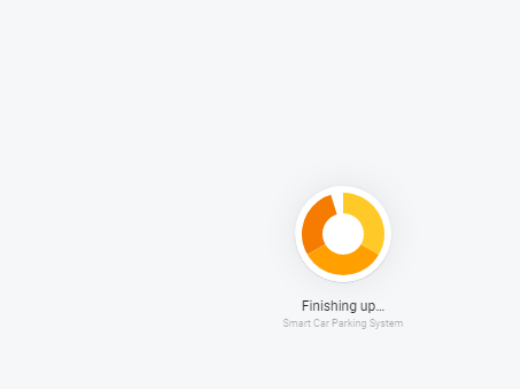


Figure 4: This Screenshot indicate that the project is still loading to be finished

- **The Project is finished and Ready for use**

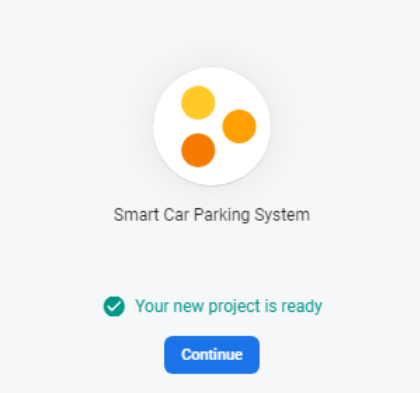


Figure 5:This screenshot indicates that The creation of project is done and Ready to use

- **Then click on Continue, Therefore the project is created**

After we create our project, it redirected to our project's console overview. If we check the URL at the top of our browser, we see something like https://console.firebase.google.com/project/smart-car-parking-system-99477/overview where the part of the URL after **/project/** matches our project's name.

To run the project you will need to get the **database link** and the **secret key**

D. Database Secret key

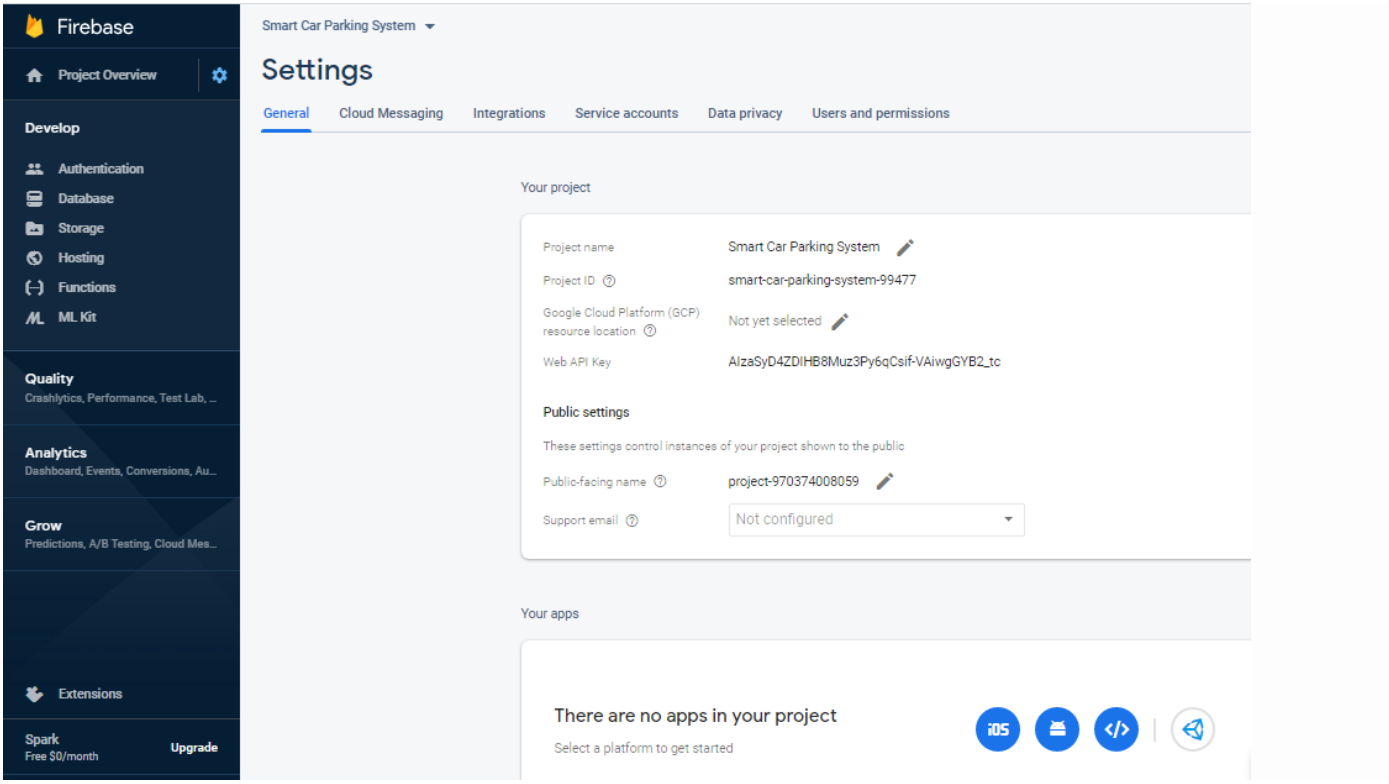


Figure 6: This Screenshot indicate our project Settings like Project name, Project Id, Web API key.

Steps followed to find out the database Secret are as follow:

- Click on Project overview
- Click on Users and Permissions
- Click on Service Accounts
- Click on Database Secret
- Click on show the secret key
- Make copy of that Secret key and Paste it in the Following line of code appear in the arduino.

define FIREBASE_AUTH’’Z3a7ckM0ScjCVN82oYdLMqcNOWxuIKhVukBmTAHb’’.

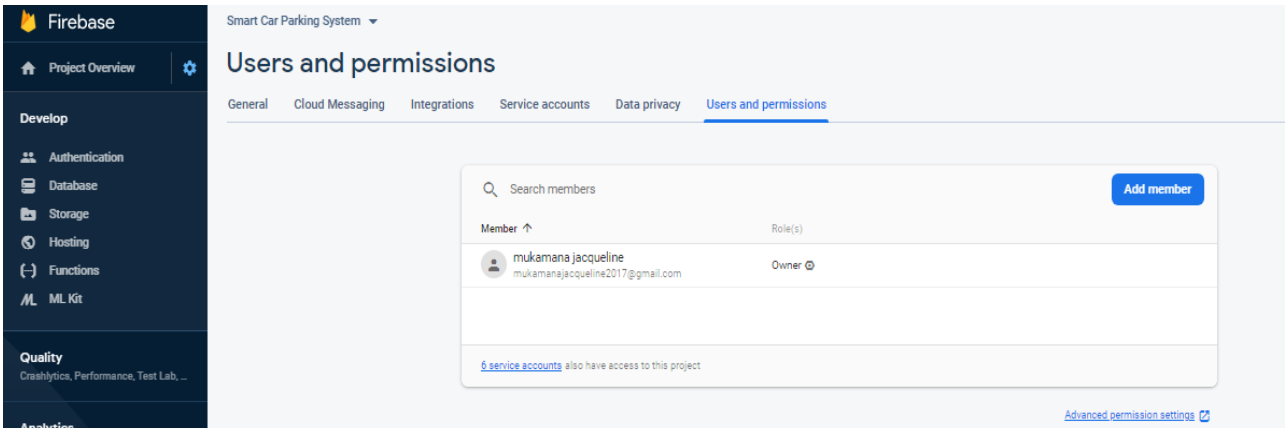


Figure 7 : This Screenshot indicate the users and Permissions we already have.

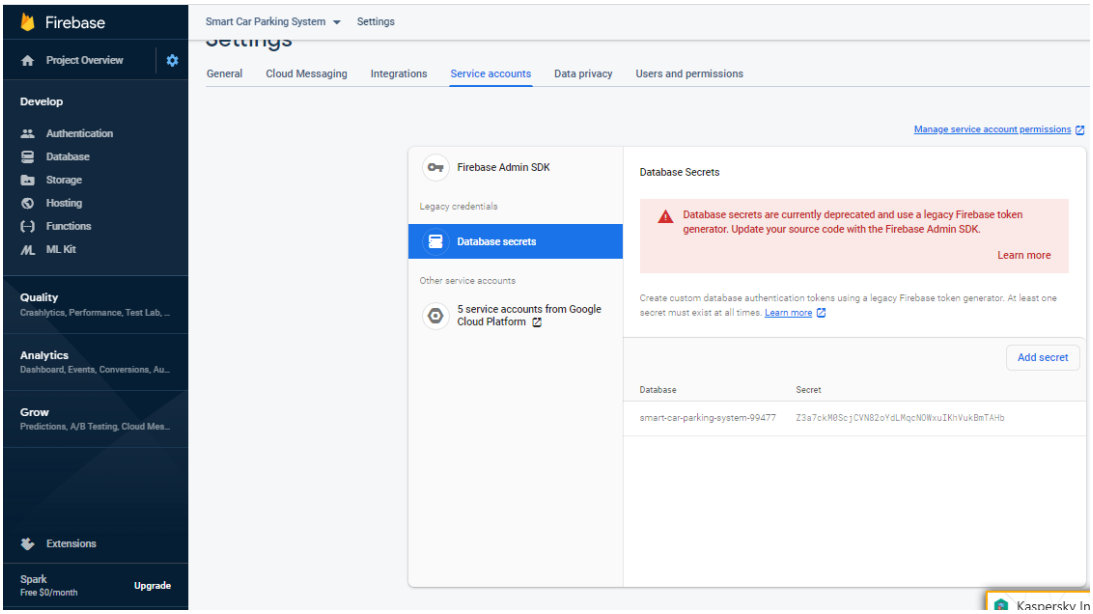


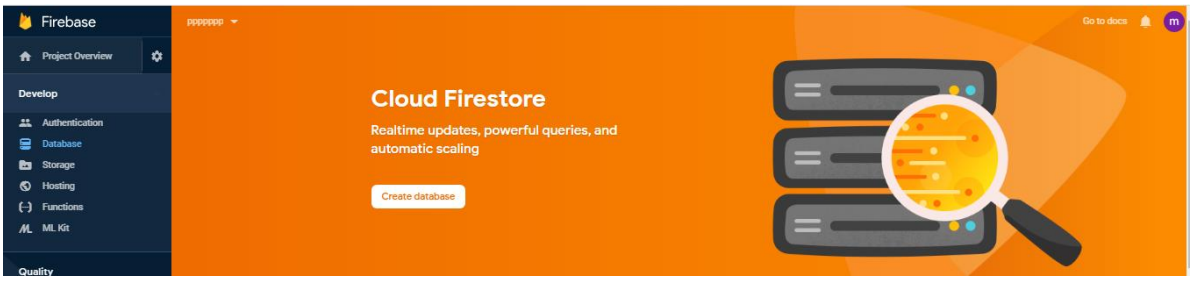
Figure 8: This Screenshot indicates the database secret used in our project

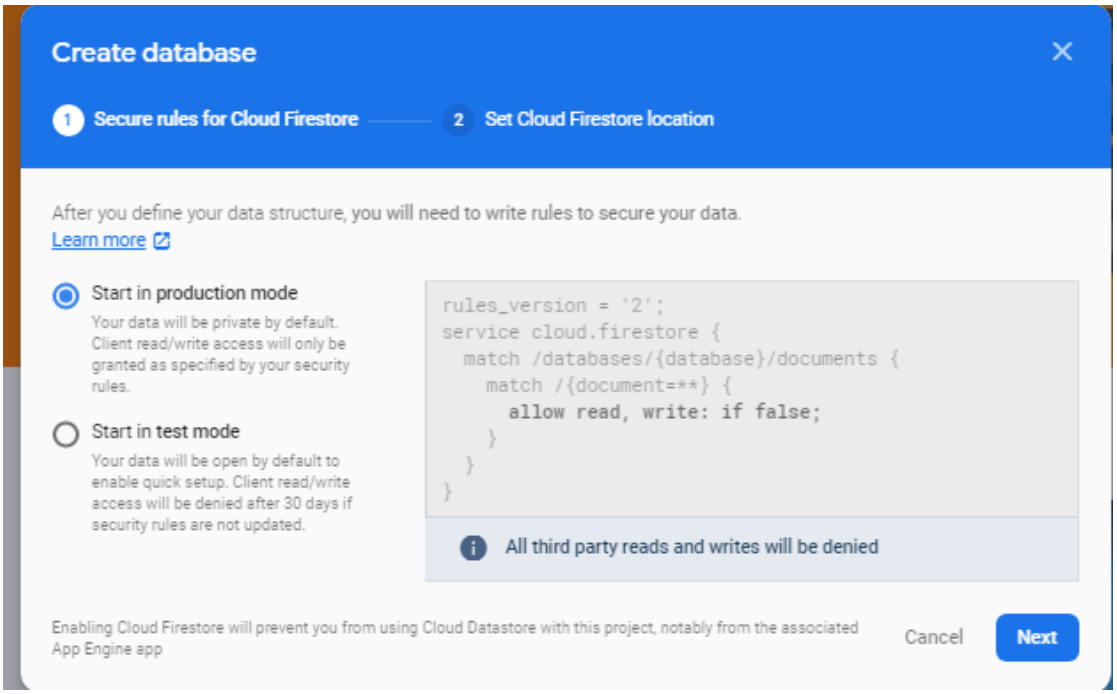
E. Database Link

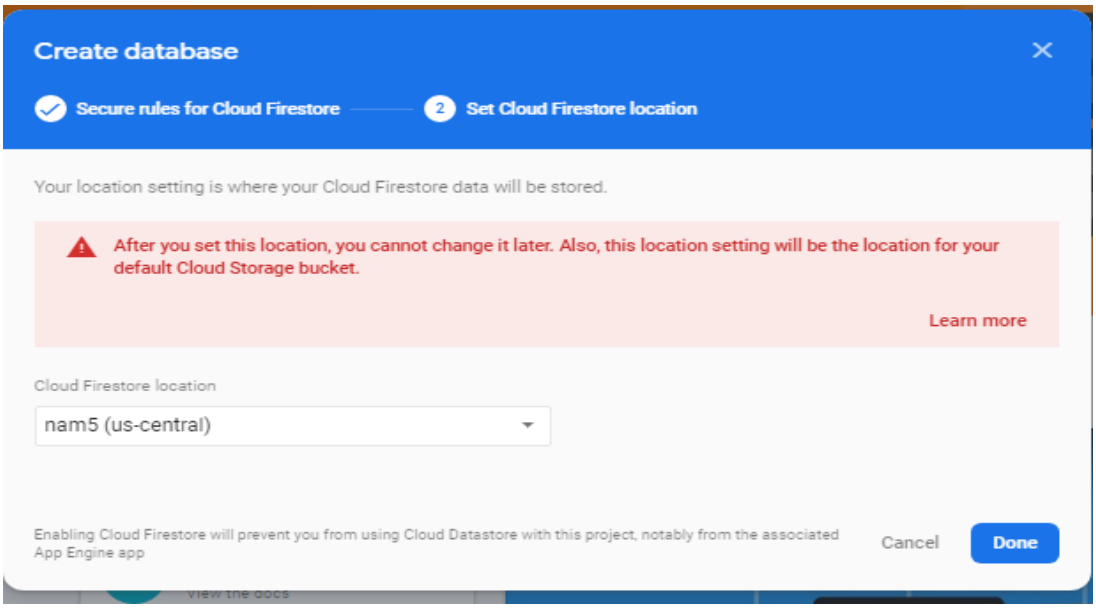
Steps followed to find out the database Link are as follow:

- Click on Project overview
- Develop
- Click on Database
- Create a Database
- Click Next
- Done

Choose **Real-time database** instead of using **Cloud Database** because our project display the slot which are empty and occupied by Car at the real time.




Create database
1 Secure rules for Cloud Firestore
2 Set Cloud Firestore location
After you define your data structure, you will need to write rules to secure your data.
Learn more
Start in production mode
Your data will be private by default. Client read/write access will only be granted as specified by your security rules.
Start in test mode
Your data will be open by default to enable quick setup. Client read/write access will be denied after 30 days if security rules are not updated.
rules_version = '2';
service cloud.firestore {
match /databases/{database}/documents {
match /{document=**} {
allow read, write: if false;
}
}
}
All third party reads and writes will be denied
Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app
Cancel
Next


Create database
Secure rules for Cloud Firestore
2 Set Cloud Firestore location
Your location setting is where your Cloud Firestore data will be stored.
After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.
Learn more
Cloud Firestore location
nam5 (us-central)
Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app
Cancel
Done

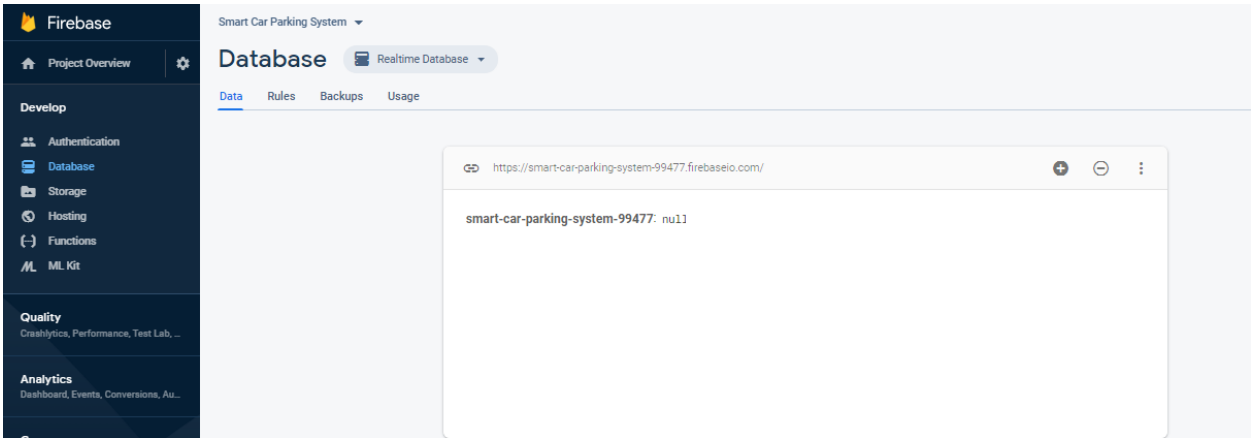


Figure 9: This screenshot shows the real time data of how smart car System works.

Our Project ID appears in a few places here. There's a URL that looks something like **your-project-id.firebase.io** (smart-car-parking-system-99477.firebase.io).This is our **Firebase URL** and it's pretty important. It describes where to find all the data for this app.

For database link (Uniform Resource Locator) on database and copy the URL Exclude the starting https:// and the Ending / and then put in the following line of code in Arduino.

define FIREBASE_HOST "smart-car-parking-system-99477.firebase.io.com"

F. Database View

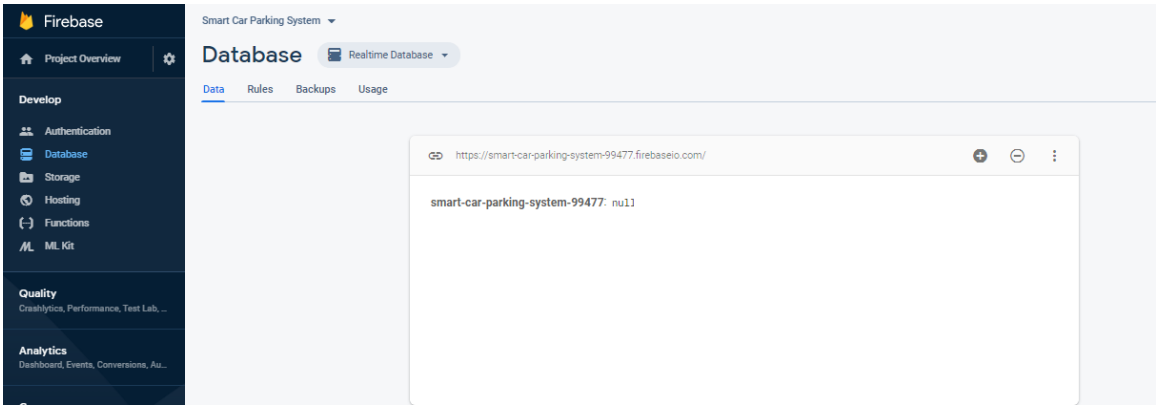
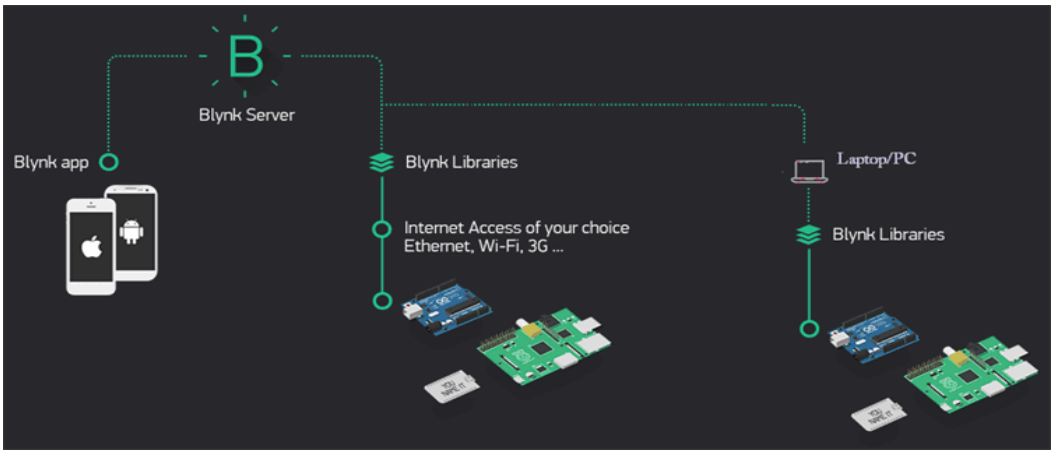


Figure 10: This screenshot allows seeing the database data at the real time according to our project (Smart car Parking System).

3. Setting up Blynk App

Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.

Blynk is an IoT (Internet of Things) platform using which you can easily and remotely control hardware. Additionally, you can also view sensor data, store the data, visualize the data etc. all over the internet.



There are three major components in the Blynk Platform. They are:

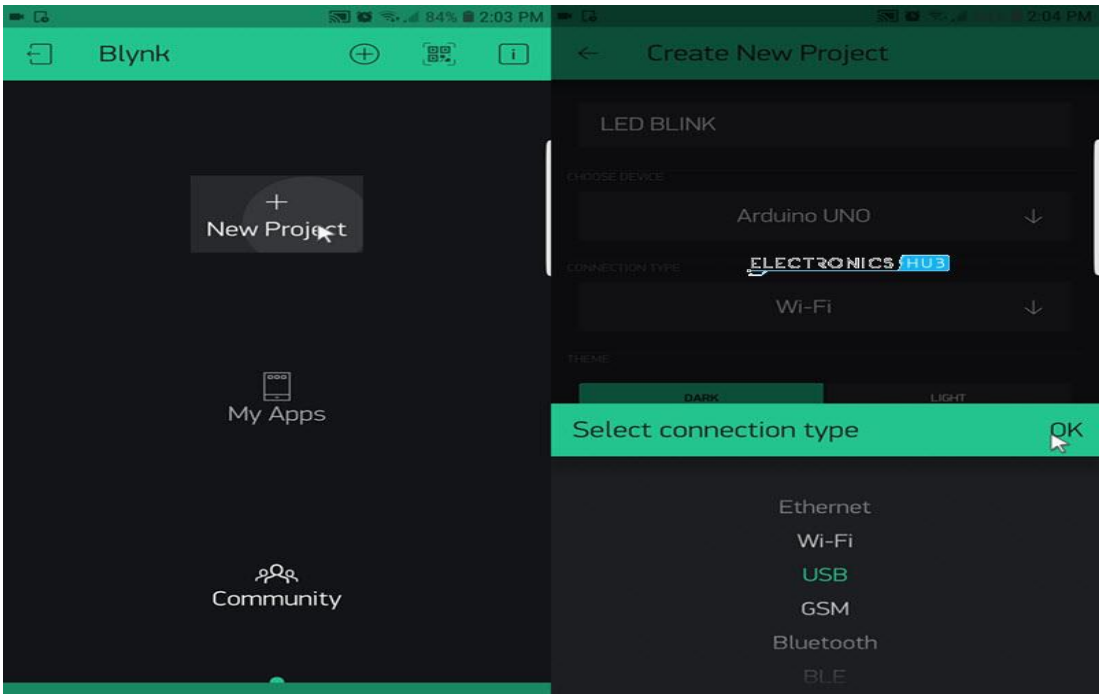
1. Blynk App
2. Blynk Server
3. Blynk Libraries

Getting started with Blynk app

- Sign in to **Blynk app** and create a new project
- Give a name to your project. Choose device as **Arduino** UNO. ...
- Check your E-mail for Author Token.
- Previous Next. Select "ADD DEVICE" and add Button. ...
- In Code copy and paste your Author Token.
- Open Command Prompt. ...
- Now write **blynk**-ser.

The first step is install the Blynk App on your smartphone. You might have to register with a valid e-mail ID to completely utilize the Blynk services.

Install the Blynk app from App Store or Google Play Store depending on your device. Open the app and enter your credentials, if required. Then select “New Project”. Provide an appropriate name to your project and also select the hardware i.e. Arduino UNO.

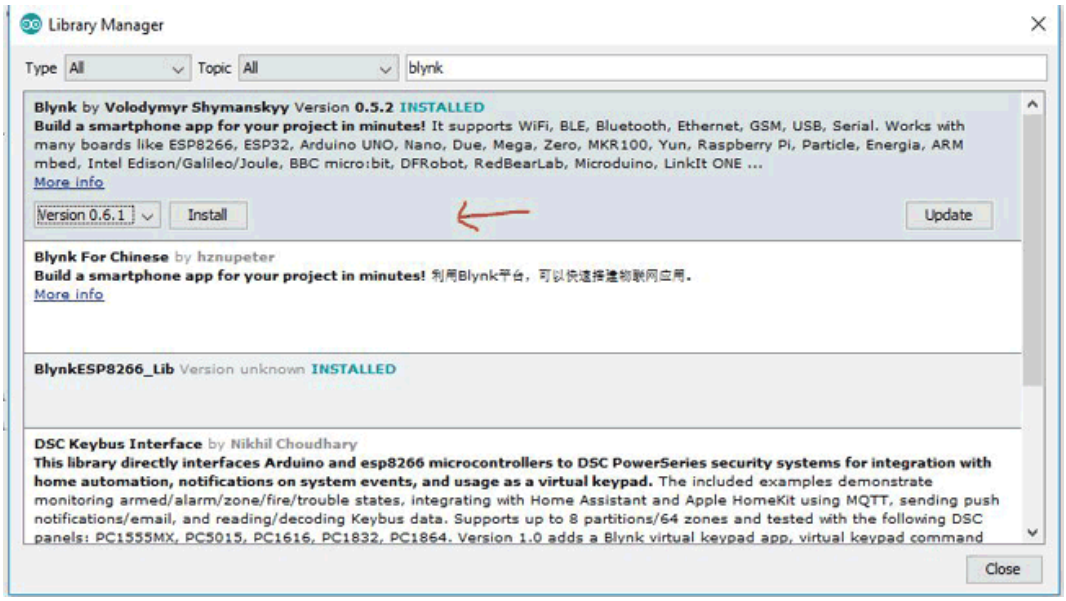


Additionally, you have to select the type of connection you are going to use. So, select “USB”. Then click on “Create” option. Your Blynk project will be created and an Authentication Token will be sent to the registered e-mail address. Make a note of this Auth Token.

Installing Blynk Libraries

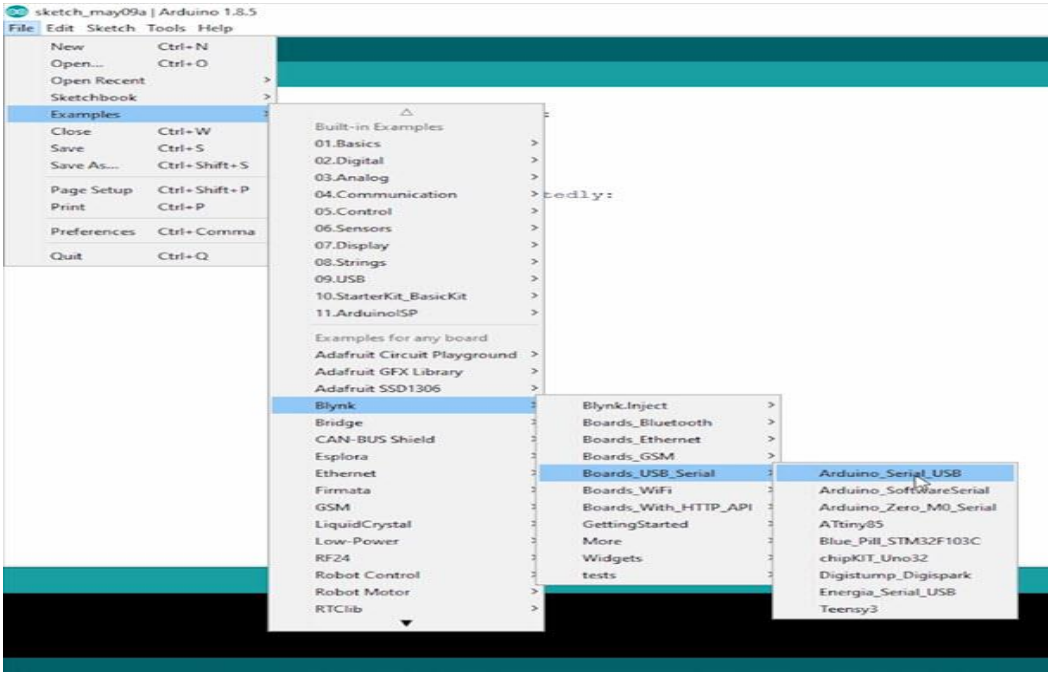
Go to the official GitHub page of Blynk and download the latest release. Download the zip file. Extract the contents of the zip file and copy the contents of the “libraries” folder and paste them into the libraries folder of Arduino.

To install the library, Go to *Sketch -> Include Libraries -> Manage Libraries*. Then search for Blynk and install the latest version as shown below.



Uploading the Code to Arduino

Open Arduino IDE and select File –> Examples –> Blynk –> Boards_USB_Serial –> Arduino_Serial_USB sketch.

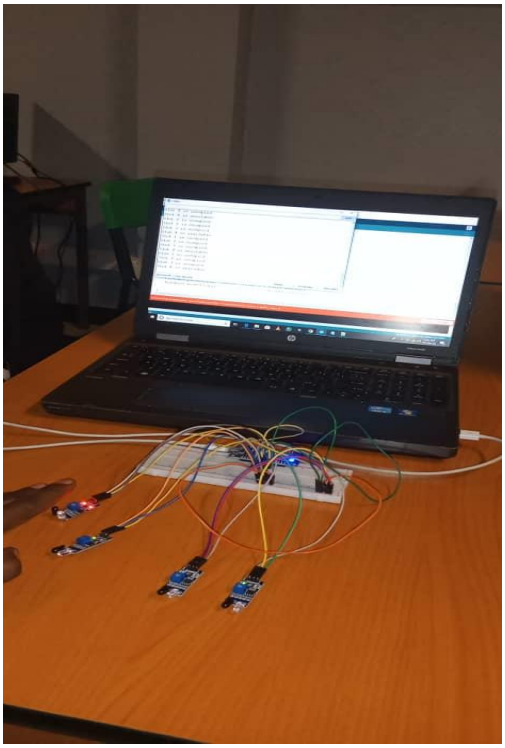


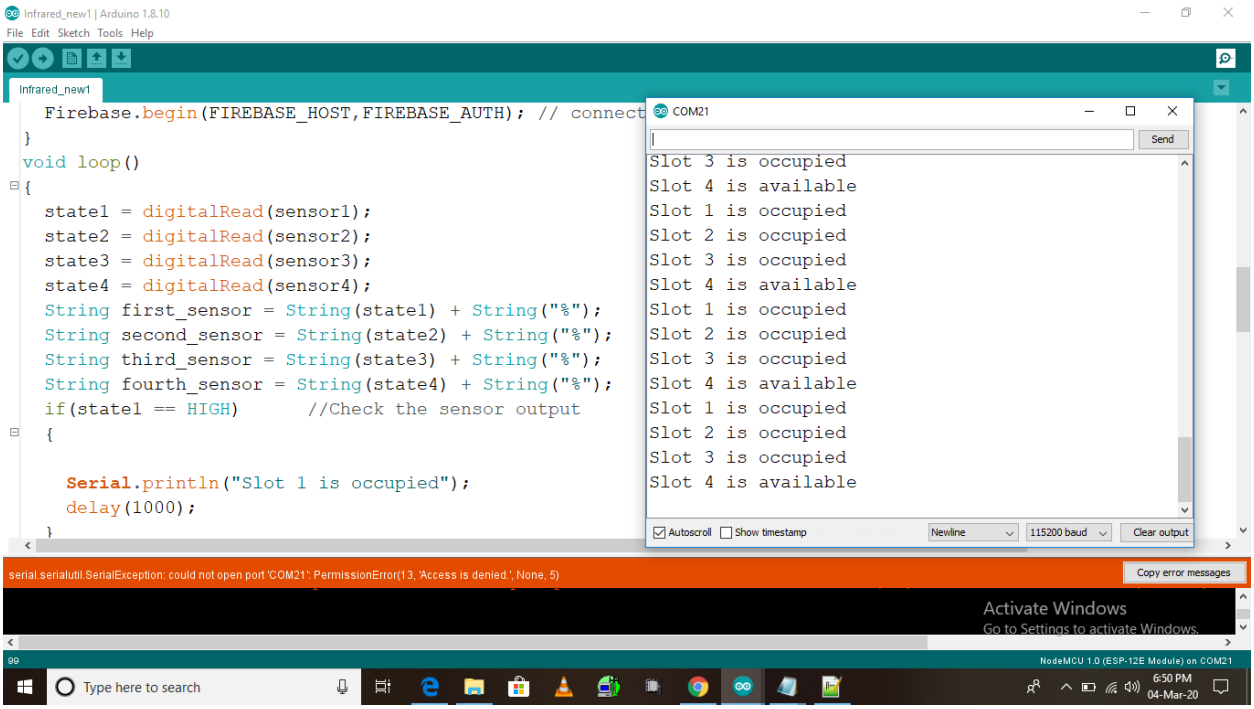
Scroll down to the line that says ‘char auth[]=”YourAuthToken”;’, and add the Auth Token you received in mail in place of YourAuthToken. Select the correct COM Port (COM5 in my case) and upload the sketch to Arduino.

References

1. https://www.electronicshub.org/control-arduino-using-blynk/
2. https://iotdesignpro.com/projects/control-arduino-remotely-using-blynk-app
3. https://www.circuitbasics.com/arduino-basics-installing-software/
4. https://www.digikey.com/en/maker/blogs/2018/how-to-get-started-with-arduino
5. http://mariechatfield.com/tutorials/firebase/step1.html

Photos




Infrared_new1 | Arduino 1.8.10
File Edit Sketch Tools Help
Infrared_new1
Firebase.begin(FIREBASE_HOST,FIREBASE_AUTH); // connect
}
void loop()
{
state1 = digitalRead(sensor1);
state2 = digitalRead(sensor2);
state3 = digitalRead(sensor3);
state4 = digitalRead(sensor4);
String first_sensor = String(state1) + String("%");
String second_sensor = String(state2) + String("%");
String third_sensor = String(state3) + String("%");
String fourth_sensor = String(state4) + String("%");
if(state1 == HIGH) //Check the sensor output
{
Serial.println("Slot 1 is occupied");
delay(1000);
}
COM21
Send
Slot 3 is occupied
Slot 4 is available
Slot 1 is occupied
Slot 2 is occupied
Slot 3 is occupied
Slot 4 is available
Slot 1 is occupied
Slot 2 is occupied
Slot 3 is occupied
Slot 4 is available
Slot 1 is occupied
Slot 2 is occupied
Slot 3 is occupied
Slot 4 is available
Autoscroll
Show timestamp
Newline
115200 baud
Clear output
serial.serialutil.SerialException: could not open port 'COM21': PermissionError(13, 'Access is denied.', None, 5)
Copy error messages
Activate Windows
Go to Settings to activate Windows.
99
NodeMCU 1.0 (ESP-12E Module) on COM21
Type here to search
6:50 PM
04-Mar-20