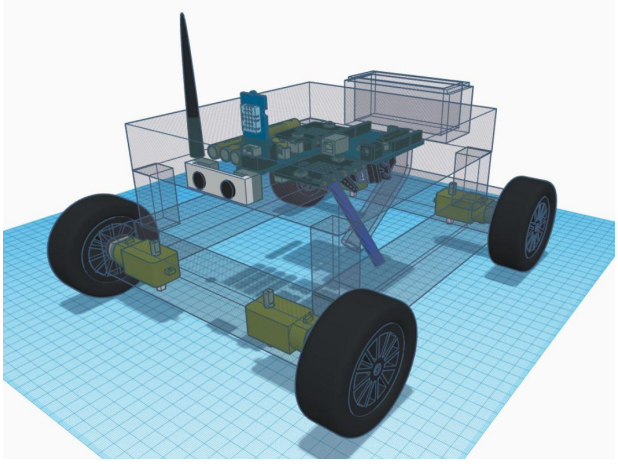
AgriBot
Design Manual



E/16/089 - Dissanayake N.A.D.D
E/16/313 - Randeniya P.M.M.P
E/16/360 - Somarathna N.C.D

Contents

1.0 Introduction

1.1 Current Context of Agriculture

1.1.1 Overview

Agriculture sector has always performed as a major economic force in Sri Lanka, making a significant contribution to the national economy, food security and employment. At the same time agriculture is the livelihood of the majority in the rural sector and plays a key role in alleviating rural poverty. This has been well recognized from the time of independence and there has always been a cabinet portfolio set aside for the agriculture sector.

1.1.2 Problem

Lack of laborers, the difficulty of finding labourers or can't afford daily wages for them are some of the main problems that today's farmers are facing. Not only that less knowledge about environmental conditions and pests also a problem faced by farmers.

1.1.3 Solution

The solution to the problem will be an automated robot to automated the seeding process as well as to identify the environmental conditions.

1.2 Our Solution - AgriBot

1.2.1 Why AgriBot?

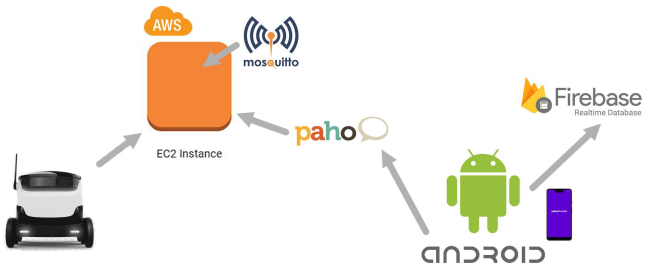
Smart Farming is a widely growing area. In smart farming user can monitor their field via smart device and control the watering, fertilizing autonomously. With this concept, people tried to develop into the next level. They want to use robots into the field to reduce the labour work. There are robots which can do seeding, cropping, identify diseases and literally everything. So now people are not just monitoring the field, they can maintain the whole field and labour cost is minimum.

So you will ask if there are already machines which can seeding, why do we need an AgriBot? The answer is normally there are some machines which consume a lot of power because they have heavy machinery components. They have a lot of disadvantages like, they have huge drilling components which turn upside down the field area, not good for the soil creatures and soil structure, sound and air pollution, not good for small seeds. In that case, AgriBot works really well, it can map the whole field and seeding. Agribot is the only drill point where we need to put seed. Also cost-effective, lightweight and eco friendly. This is the best way to seeding small seeds in big areas. Another thing is very easy to operate through a user-friendly mobile app.

1.2.2 AgriBot for Greenhouse farming

AgriBot is the best solution for modern Greenhouses because those are full covered areas which have sensitive sensors all over the place. Because in greenhouses every condition which plant will depend, is measured and controlled accurately and another thing is there are tap lines all over the place. For areas like this, you can’t use heavy machinery or drones to seed plants and AgriBot is the perfect solution.

1.3 AgriBot Overview



Robot is directly connected to the AWS EC2 instance using *mosquitto* mqtt broker. Robot device uses a cellular network connection and sends sensor data and connection status to the mqtt broker. Broker is able to send those to a mobile app..

Mobile App is connected to both EC2 instances(mqtt broker) as well as the FireBase database also. Mobile app sends the planting parameters and critical control signals to the robot through the MQTT broker and receives all data from the robot. FireBase database is used to store the Usernames and Passwords and other info of the devices and it helps to authorize the device for the mobile app. Which means it helps to create a secure connection between app and device.

We used,

Mobile Application - Java Android Development, Android Studio
Cloud Storage - AWS EC2 instance, Ubuntu virtual machine
Database - Google FireBase realtime database
MQTT - mosquitto broker, Paho mqtt library

For designing the software parts of the AgriBot.

1.4 Overall design

Height :15-20 cm
Length : 20-25 cm
Width : 15-20 cm
Weight : 1 - 1.5 kg
Speed : 0.5 ft/s
Average work time : 60 - 90 min

2.0 Software Deployment

2.1 Mobile Application.

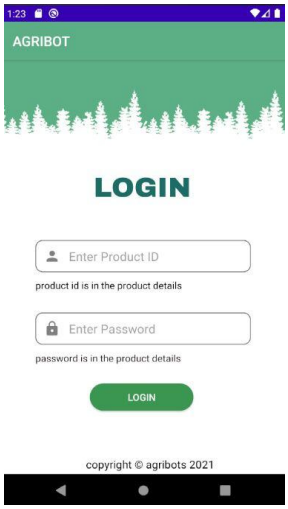
Click [here](#) for the Source Code

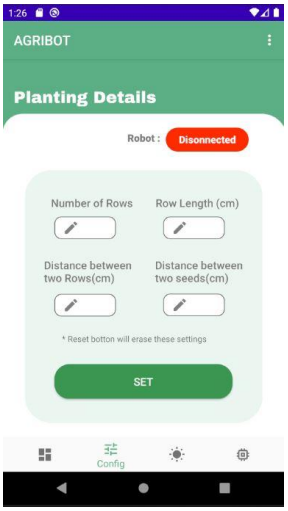
2.1.1 Overview

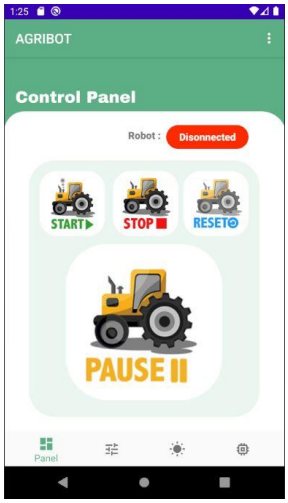
Android Studio is used to develop the mobile application of the AgriBot. Main objectives of the mobile applications are,

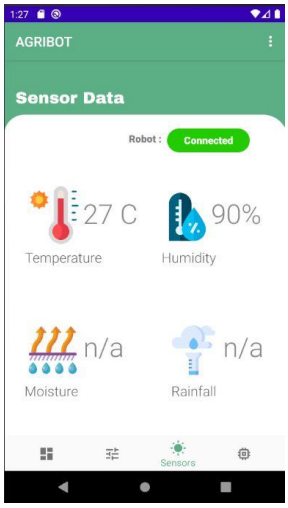
1. Send critical commands to the Robot - Start/Stop/Reset/Pause
2. At the start setup the parameters - row length/no of rows/gap between rows/gap between seeds
3. Get the readings from the robot and display it to the user - Temperature/ Humidity

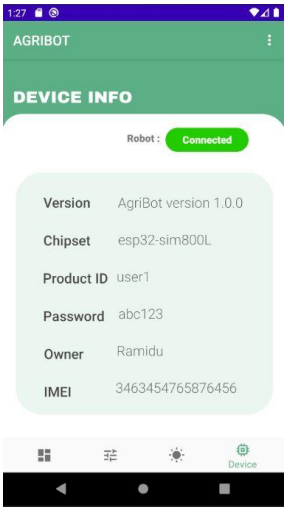
2.1.2 Mobile Application UI

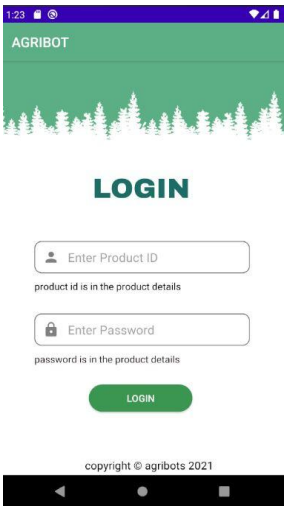






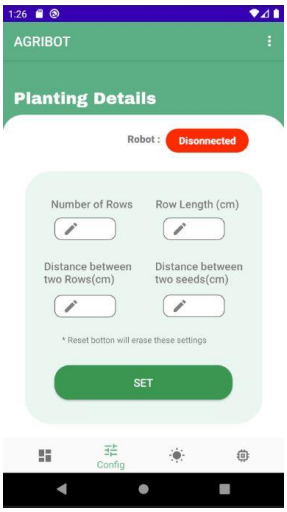






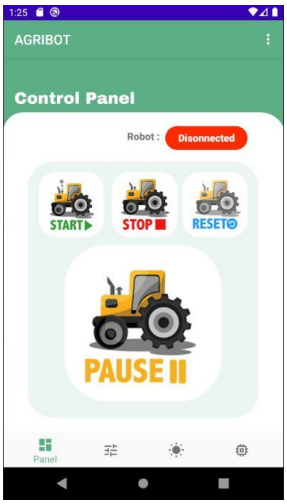
User Login Page

- Users should enter the “Product ID”(which is provided by the company) and the “Password” of the AgriBot account to log into the mobile application .
    - Product ID and Password can be obtained when a user buys the Agribot Product.
    - If a user needs to change the password it can be changed later.
- After you press the login button Details will check with the database and if it is correct, you will directed to the application.



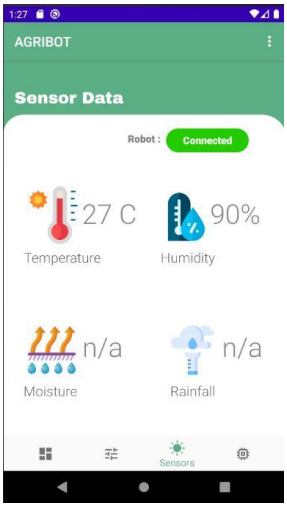
User Input Panel

- Using this panel, users can input the necessary details for the planting seeds which are used to determine the number of seeds that have to be planted.
    - Number of Rows (min:1, max: 1000)
    - Length of a Row (min:10m, max:100)
    - Distance between two Rows (min:10cm, max:1000 cm)
    - Distance between two seeds (min: 5cm, max: 100cm)
- As we mention there are limitations to these parameters, after you set those robot will all ready to start the process



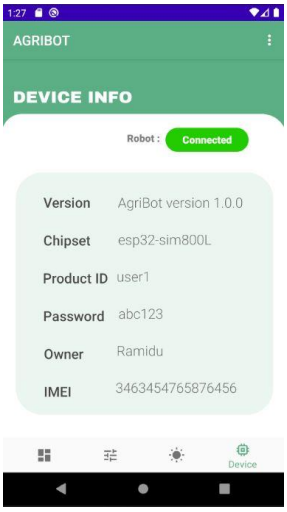
Control Panel

- The Control Panel is used to give critical commands to the Robot.
    - **START** - Start the robot after setting parameters, first user have to setup parameters, if not this won’t work
    - **STOP** - Stop the process of the robot
    - **RESET** - Reset the parameters of the robot, if we press this user have to setup parameters again
    - **PAUSE** - This is the emergency stop for the robot
        - There is an indicator to show the user whether the robot is connected not with the mobile application.



Sensor Details Panel

- This is the window in which Temperature, Humidity & Moisture details are displayed to the user which is obtained by the sensors attached to the robot.



Device Details

- Users can check the device details of the AgriBot using this menu.
  - Device Version
  - Chipset
  - Product ID
  - Password
  - Owner
  - IMEI number

2.1.3 MQTT Architecture in the Mobile App

- In our project we choose mqtt as our data transfer protocol
- Inside the AWS EC2 there is a MQTT broker to handle messages
- So inside the App we have to send and receive data according to the MQTT protocol
- We used Paho MQTT java client library for the app

- MQTT connection
  - Create a new mqtt client with unique client id for each device
  - Port number 1883 and ip is Vm's ip address
  - Set up a connection time out
  - Set up automatic reconnect option true
  - Set up a keep alive interval

- MQTT Subscription
    - Each device have a unique topic, using that app subscribe to that topic
    - After that it receives the sensor data of that device
    - Used wildcards topics to group the data
    - Used clean session option
- MQTT Publish data
    - Published QOS 2 messages
    - Publish last will message when connection ends
    - When publishing data, use the topic which is identical to that device
    - Also used wildcards to group the data when publishing

2.1.4 Firebase Database Connection

- import google firebase dependencies and setup security requirements
- Connect the realtime database to the application
- When user enters the user name it checks with the database and if it exists, check the password also
- If login successful, it fetch the corresponding data relevant to that device
    - Like Mqtt Topic, manufacturing details
- MQTT topic will use to publish data, subscribe to device and all other mqtt activities

2.1.5 Configuration of the Mobile Application

- Download the mobile application from the Play Store. (ISO application also will be available soon) and Install it on the mobile.
- Open the app and log in to the user account using the given Username and the Password.
- Using the “User Input Panel” user should input the necessary details of the seeds which they expect to be planted.

  Eg: Number of Rows which should be planted.
  Length of a Row
  Distance between two rows and two seeds
- After setting the parameters, the user can control the seeding process using the “Control Panel” page.
- To check the sensor readings user have to use the “Sensor Detail Panel” page of the application. In that page all the sensor readings(Temperature, Humidity and Moisture) will be displayed. Those details will update periodically.

2.1.6 Design Techniques Used in the APP

1. **Navigation Bar**
    a. In the mobile app, change between different panels, there is a bottom navigation bar. User can easily operate the app and increase the UX capabilities.
2. **Fragments**
    a. In a one activity there are multiple fragments, this fragments are lighter than activities, so change between activities takes much time. So this will improve the gui and low time consuming method
3. **Green Theme**
    a. Our project is a agricultural one, so we used green color theme all over the app, which makes the app unique and also increase UI/UX

2.1.7 Security of the Mobile Application

- Mobile application is connected with the database and can directly access the respective data for the respective username and password.
- Users only have read access to the database. Only company has the write access to the database
- Users Can’t Start the Robot until they set the seeding parameters.

2.2 AWS Server

2.2.1 EC2 Instance Configuration

- Create an EC2 instance and a Ubuntu virtual machine in the AWS cloud services.
- Instal MQTT broker in the Ubuntu Virtual machine.
- Setup the security rules of the EC2 broker - Give access only to port numbers 1883 and 22 (MQTT, SSH)
- Using the IP of the Virtual Machine application is connected.

EC2 instances act as the “broker” in this setup. Broker has topics for each command(Start, Stop, Pause, Reset & Configurations). Mobile app publish data to each topic when users use each command. Each robot has subscribed to those topics and receives relevant data from the user. After the robot receives data, the robot can work autonomously even without a connection of the mobile application.

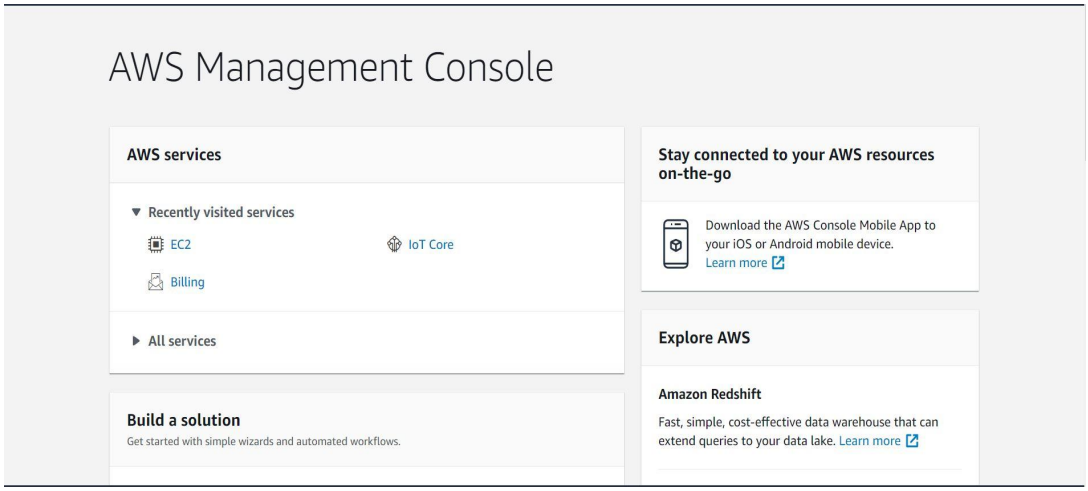
Vice Versa broker has another set of topics for temperature and humidity readings as well. When robot sensor those data and publish them to the broker. Each mobile application subscribes to those topics and receive the necessary sensor readings.

In this process a unique id is used in robots as well as the mobile applications to uniquely identify each robots and applications.

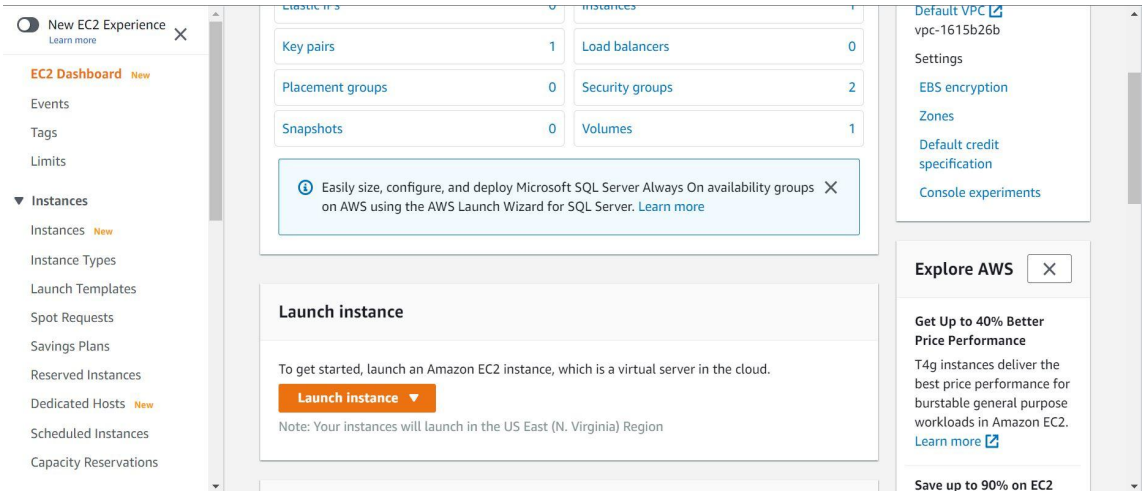
Eg: 21@3467!uj/temperature/#

2.2.2 How to Create EC2 Instance

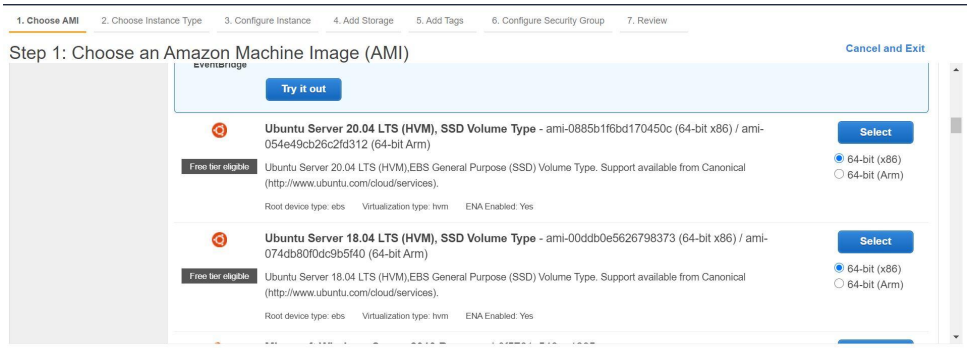
- Go to AWS Management Console & Select EC2.



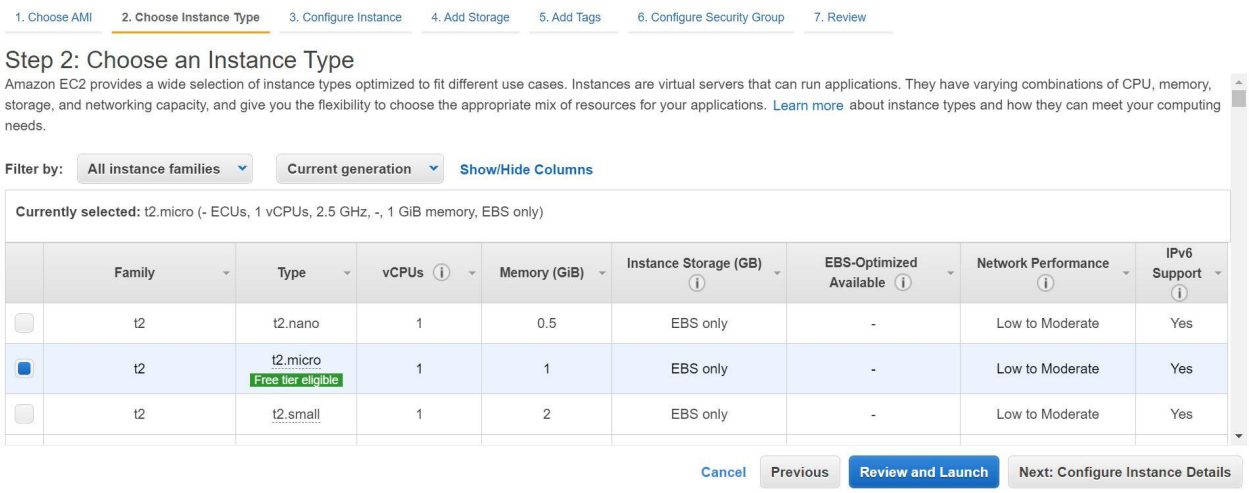
- Select Launch Instances



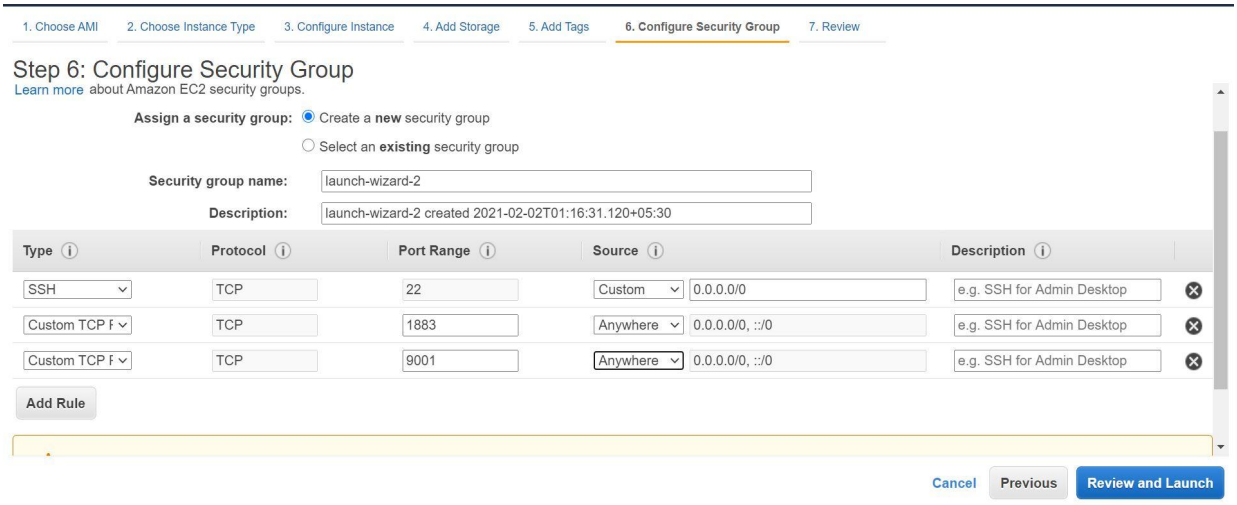
- Select the ubuntu server.



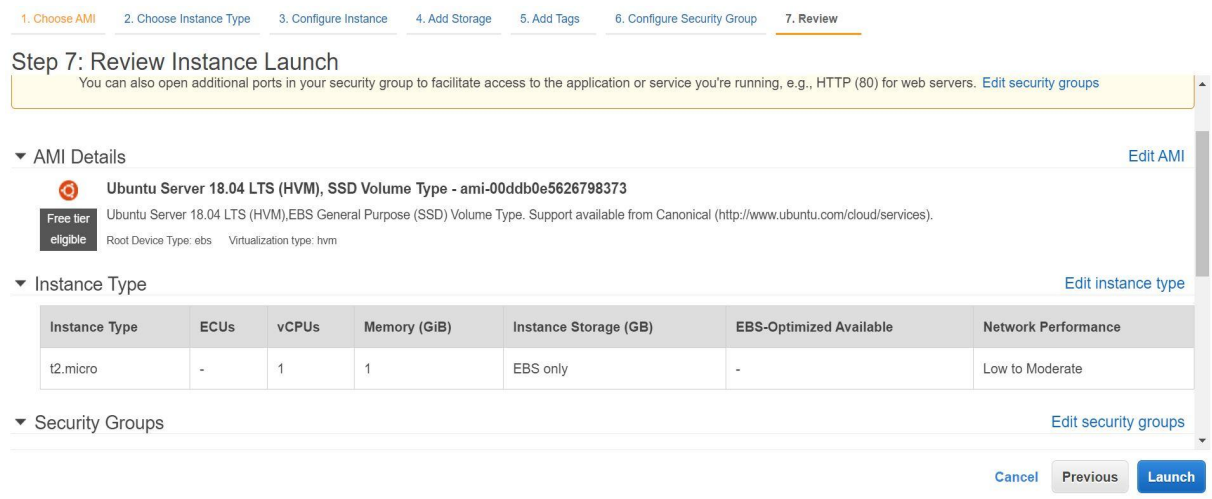
- Select “ Free tier Eligible” Option & Click “Next: Configure Instance Details”



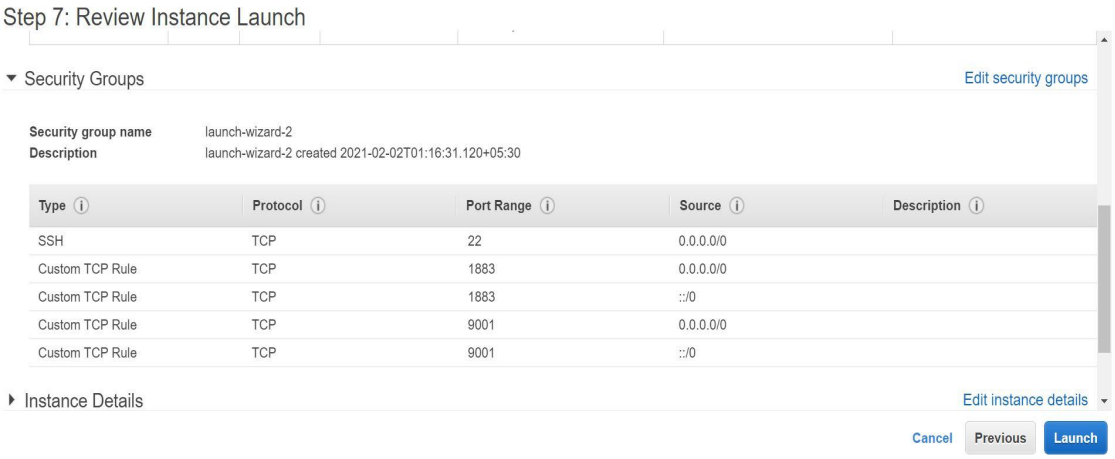
- In “Configure Security Group” Tab add above rules using “Add Rule” option. Then Click “ Review and Launch”
- Include port numbers 1883(mqtt) , 22(ssh) with tcp connection



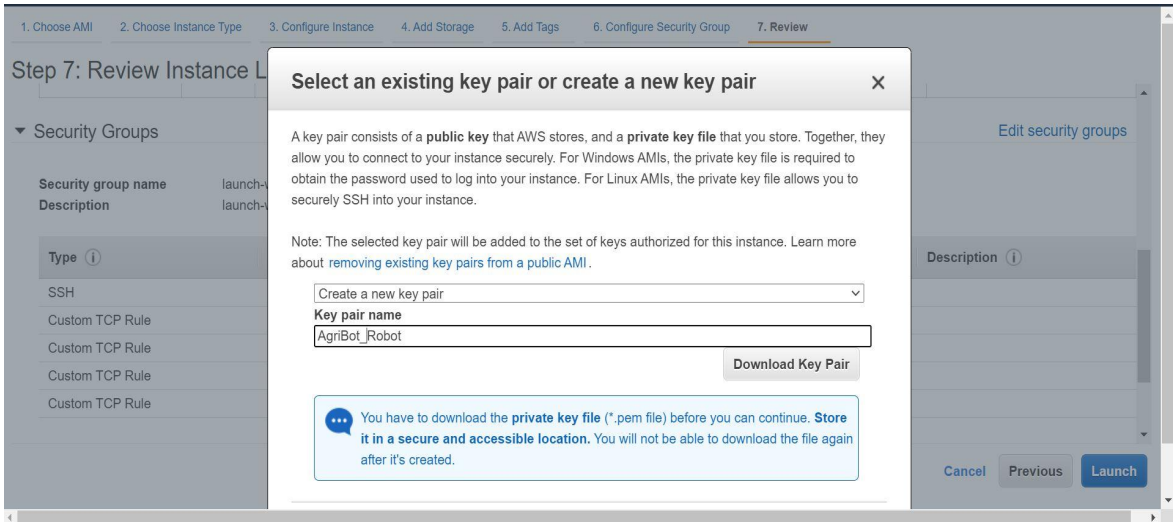
- Click “Launch”



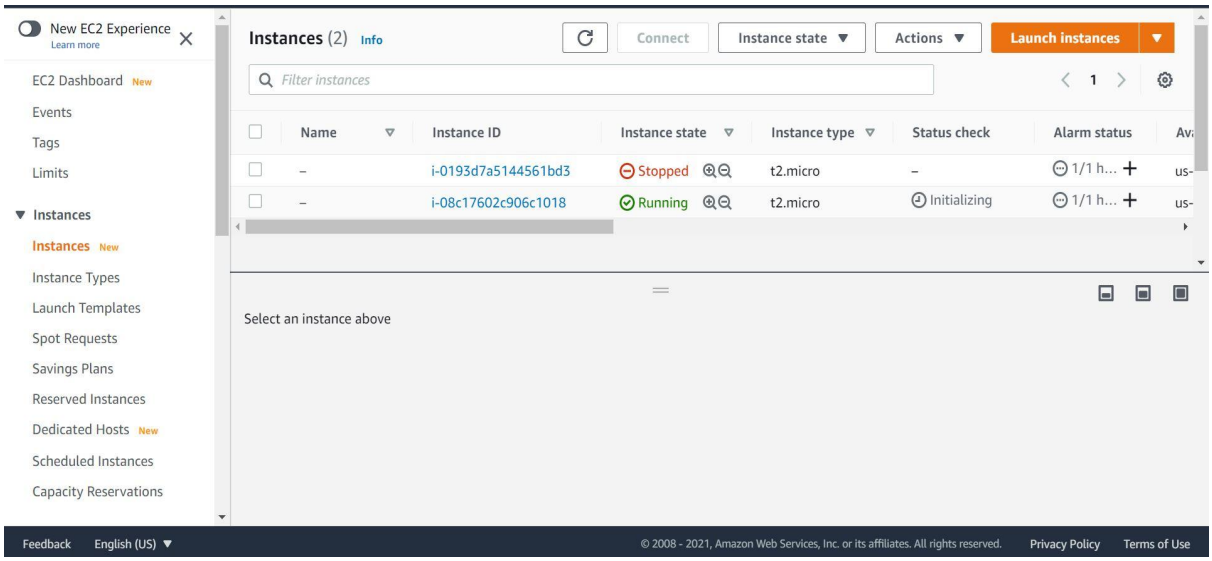
- Review



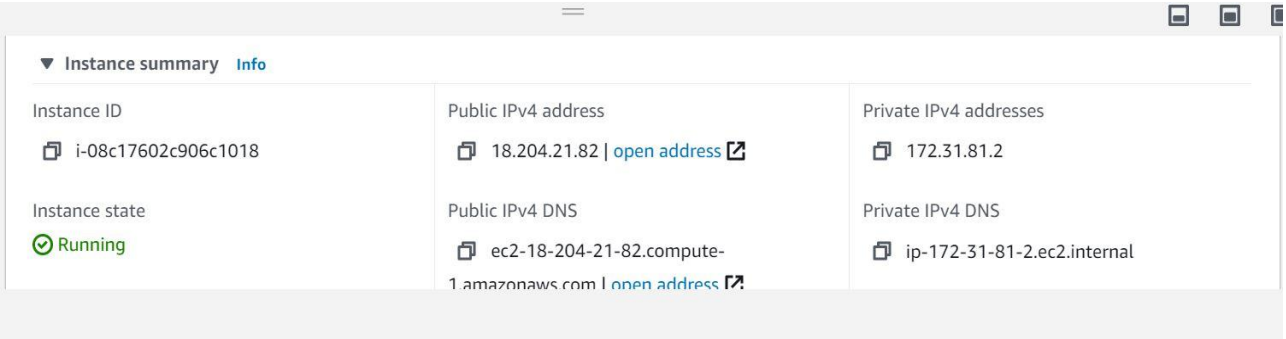
- "Create a new key pair" option & Give a "Key pair name" ,Then Download the key pair and keep the file directory



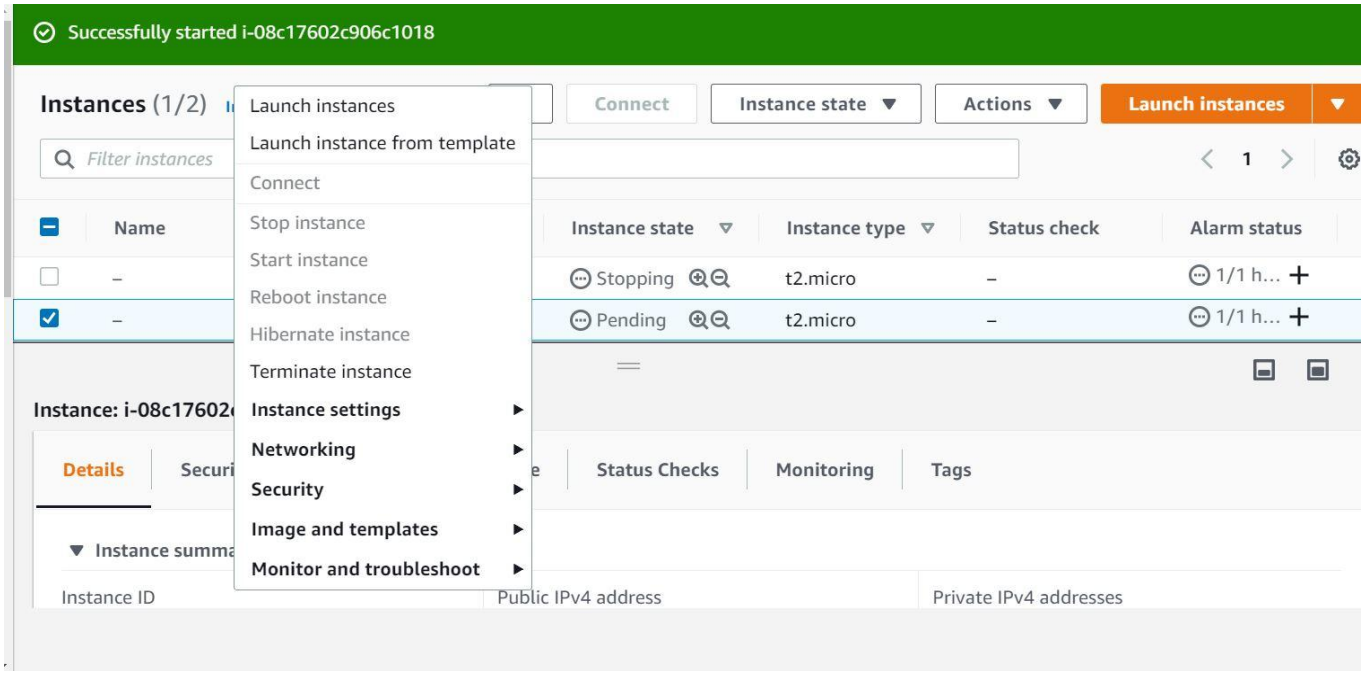
- Newly Created instance is Running there in instances.



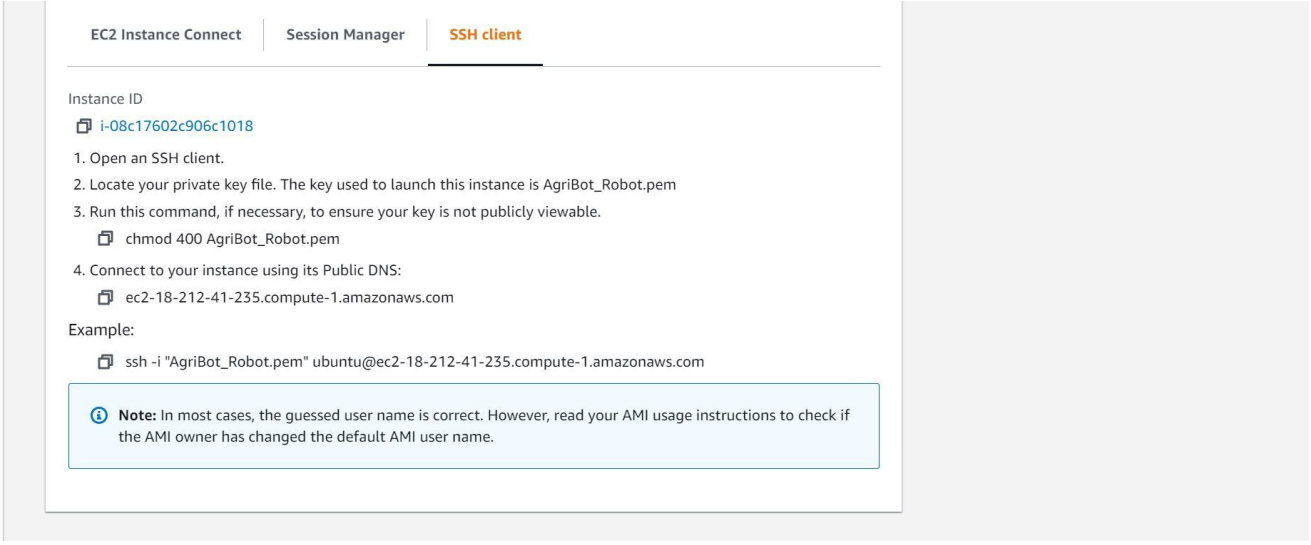
- Details of the instance. “Public IPv4 address” is used to connect with in the instance.



- Start, Stop or Connect the Instance can be done by write clicking on the instance.



- When select “Connect” this dialog box is displayed.

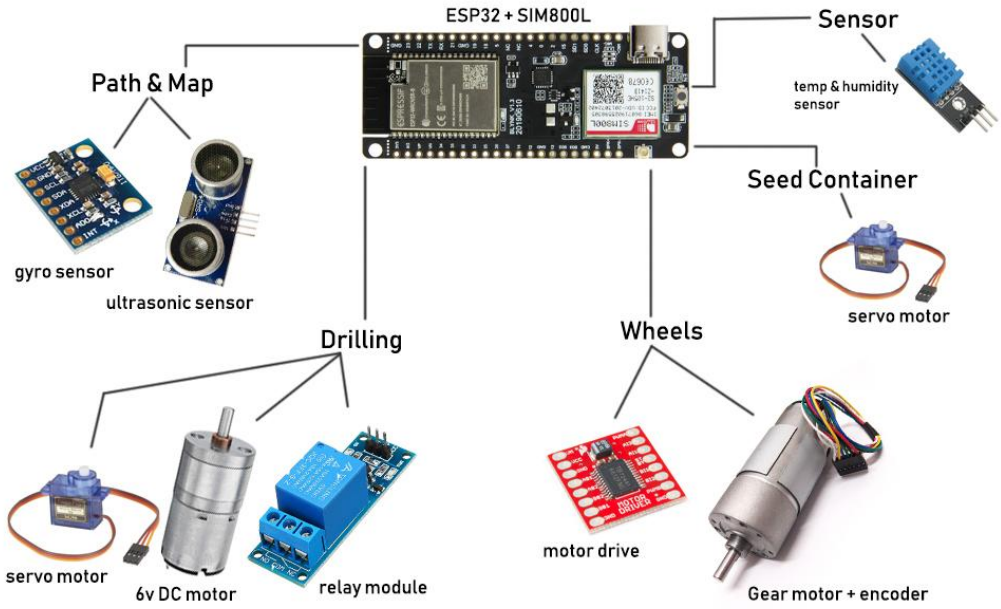


2.2.3 Setup MQTT Broker

1. Locate the ssh client (Terminal or cmd) and also locate your .pem file
2. Run this command, if necessary, to ensure your key is not publicly viewable.
    a. $ chmod 400 agribot.pem
3. Connect to your instance using its Private IP from ssh client
    a. Ex:- $ ssh -i "agribot.pem" ubuntu@172.31.53.237
4. Now you are logged into aws EC2 ubuntu virtual machine, next we have to install broker in VM.
5. Run following commands in Ubuntu VM
    a. $ sudo apt-get update
    b. $ sudo apt-get install mosquitto
6. If we want to add additional security options, locate config file inside the mosquitto directory
7. "mosquitto.conf man page | Eclipse Mosquitto" using the MAN page we can setup additional options
8. In our implementation we protect the broker from username and password
9. Knowing IP address of the VM is not enough to publish messages to broker
    a. Client has to include username , password in mqtt pub/sub commands

3.0 Hardware Development

3.1 Hardware Overview



For the development of the “Robot” of the AgriBot **ESP32 SIM 800L** model is used as the main microcontroller. It is used because of its functionality to use it with a SIM card. It connects the robot with the EC2 broker.

Path & Map

Ultrasonic Sound Sensor - Detect Obstacles in the path.

Gyro Sensor - Map the given area

Drilling -

Servo Motors - Control the Drill Bit

DC Motor - Drilling

Wheels -

Motor Drive & Gear Motors

Seed Container -

Funnel - Store Seeds

Servo Motor - Control the Seeding

Sensors -

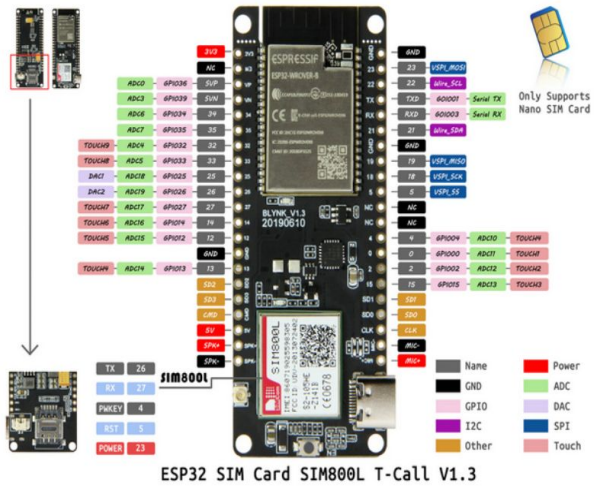
DHT11 - Humidity & Temperature Sensing

3.2 Components

3.2.1 Microcontroller

LILYGO® TTGO T-Call V1.3 ESP32 Wireless Module SIM Antenna SIM Card SIM800L Module

T-Call V1.3 Pin Diagram:



ESP32 SIM Card SIM800L T-Call V1.3

Chipset	ESPRESSIF-ESP32 240MHz Xtensa® single-/dual-core 32-bit LX6 microprocessor
FLASH	QSPI flash 4MB / PSRAM 8MB
SRAM	520 kB SRAM
Button	Reset
USB to TTL	CP2104
Modular interface	UART、SPI、SDIO、I2C、PWM、TV PWM、I2S、IRGPIO
On-board clock	40MHz crystal oscillator
Working voltage	2.7V-3.6V
Working current	About 70mA
Sleep current	About 1.1mA
SIM card	Only supports Nano SIM card
Working temperature range	-40℃ ~ +85℃
Size&Weight	78.83mm*28.92mm*8.06mm(11.77g)
Power Supply Specifications	
Power Supply	USB 5V/1A
Charging current	500mA
Battery	3.7V lithium battery

3.2.1 Sensors

DHT11 Temperature and Humidity Sensor Module is used as the sensors for the AgriBot

- Humidity measurement range: 20% ~95%
- Humidity measurement error: ±5%
- Temperature measurement range: 0℃~50℃
- Temperature measurement error: ±2 ℃
- Operating voltage:3.3 V~5 V, Operating current: 0.3mA
- Outputs both Temperature and Humidity through serial Data(digital)
- DH11 is much cheaper than DH22 and accuracy and ranges are also enough for our requirements

- Using this sensor, take measurements every 10 mins. Measure humidity and temperature of the current environment
- DHT11 sensor module is used , because for module will have a filtering capacitor and pull-up resistor inbuilt. Sono need to configure it.
- The sensor is factory calibrated and hence easy to interface with other microcontrollers.
- Output - 40 bits output

3.2.3 Drilling & Seeding

SRD-3VDC-SL-C Relay Module

- **NO (Normally Open):** the normally open configuration the relay is always open, so the circuit is broken unless you send a signal from the Arduino to close the circuit.(Drill Bit)
- Sealed typed
- Coil nominal voltage - 3 v
- Nominal current - 120 mA
- Power consumption of coil - abt. 0.36W
    - For the drillbit we used bit heavy motor and pin voltage is not enough for the motor
    - Using relay, there is an external power supply for the drilling motor (6v)

TB6612FNG Motor Driver

- Driver Model: TB6612FNG H-Bridge
- Motor supply voltage of 2.5 to 13.5 volts DC.
- Logic supply voltage of 2.7 to 5.5 volts DC.
- Output current of 1.2 amperes continuous, 3.2 amperes peak.
- Built-in thermal shutdown.
- Standby power mode.
- Small module,no heat sink required
- Efficiency 91-95%
- Voltage drop 0.05 - 0.13 V

Servo Motor - SG90

- Operating voltage - 3.0V~7.2V
- Working Frequency - 50Hz
- Motor Type - Brushed DC Motor
- Gear Type - Plastic Gears

- Two servo motors are used to work as a valve of our seed container and other one to place the drill bit in correct position
- Using servo motors, we can rotate a specific angle every time
- To close/open the seed container we can easily use the servo motor
- Drill bit is set to the rack and pinion, so rotating a specific angle we can correctly place the bit.

3.2.4 Wheels

G12-N20 Geared Mini DC Motor

- Model :GA12-N20
- Rated Voltage : 6~12V
- Revolving Speed : 100 RPM @ 6V
- Load Speed: 80 RPM
- Rated Torque: 2 kg.cm
- Stall Torque: 16 kg.cm
- Rated Current: 0.07A
- Stall Current: 1A
- Reduction Ratio: 1:10
- Total Length : 34mm
- Gear Material: Full Metal
- Gearbox Size : 15 x 12 x 10mm (L*W*H)
- Shaft Size : 3 x 10mm(D*L)
- Net Weight : 10g

Tyres

- Four 65mm diameter and 26mm width tyres will be used
- Two of them is connected to 6v gear motors
- Gear motors powered by two 6v Lipo batteries.

3.2.5 Path and Map

MPU 6050 Accelerometer and Gyroscope is to map the path of the Agribot

Accelerometer

- Digital-output triple-axis accelerometer with a programmable full scale range of ±2g, ±4g, ±8g and ±16g
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts

Gyroscope

- The triple-axis MEMS gyroscope in the MPU-60X0
- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full scale range of ±250, ±500, ±1000, and ±2000°/sec
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Factory calibrated sensitivity scale factor

- Using the gyroscope we can measure the change of axial angles
- Using that we can identify if robot has straight path or not
- If there is relative difference, using the motors each, we can straighten the path

RCWL-1601 Ultrasonic Distance Sensor

- Voltage: 3V - 5.5V
- Current: 2.2mA
- Measuring Range: 2cm - 450cm
- Working Temperature: 0ºC to 70ºC
- Compatible with HC-SR04 Ultrasonic Sonar Distance Sensor

- Sensor is used to identify the obstacles along the path
- Measuring the range front of the path, we can easily identify there is a obstacle or not

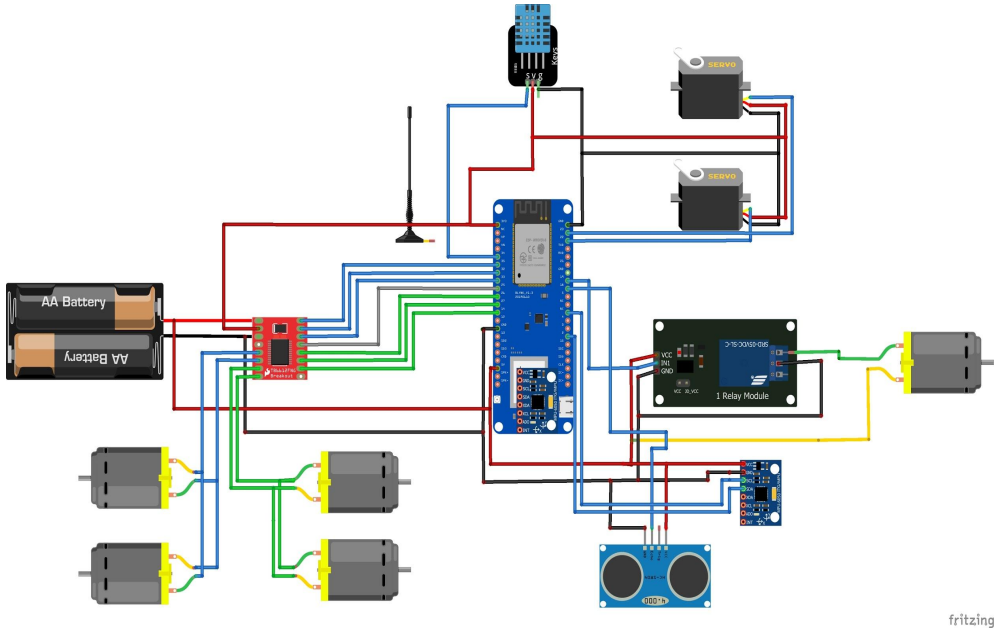
3.2.6 Hardware Programing Libraries

Click here for source codes

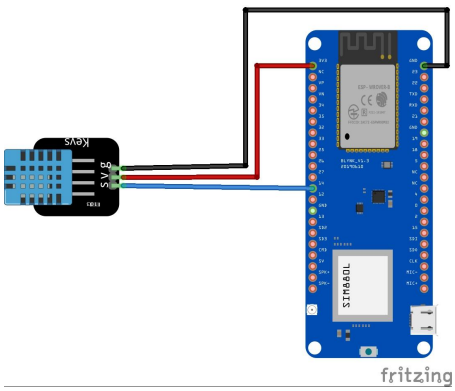
- adafruit mpu6050 - Gyroscope
- Adafruit Unified Sensor - Sensors
- Adafruit Bus IO - Signal IO
- TinyGSM Library - GPRS Module
- PubSubClient - MQTT

3.3 Diagrams

3.3.1 Circuit Diagrams

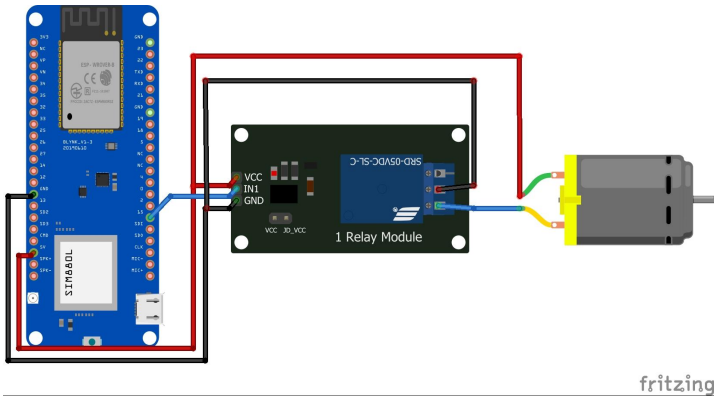


Main Circuit Diagram

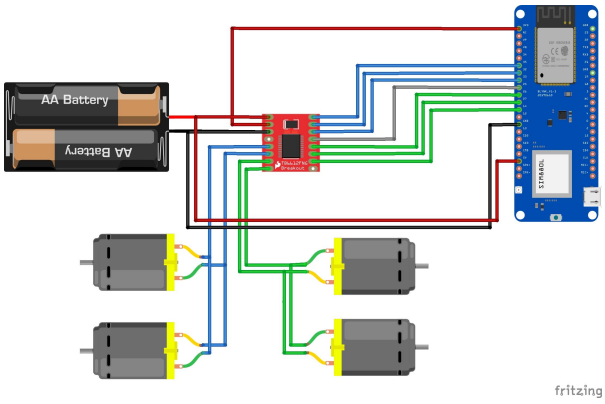


DHT11 Temperature and

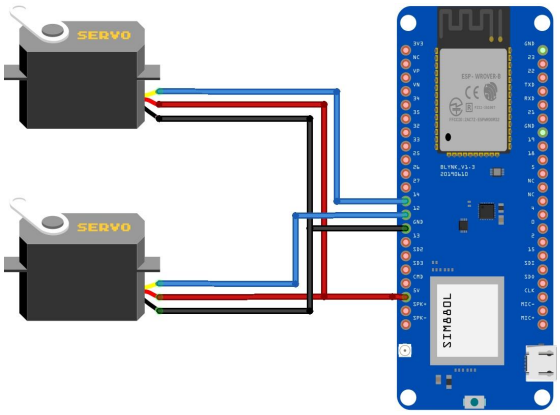
Humidity Sensor Module



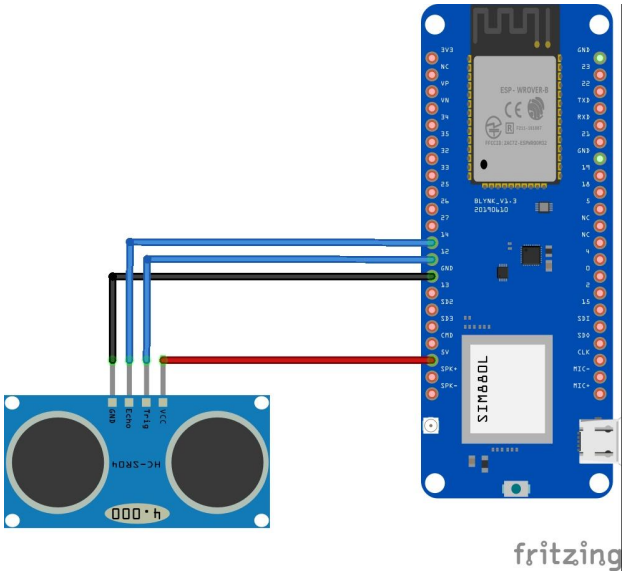
SRD-3VDC-SL-C Relay Module



TB6612FNG Motor Driver

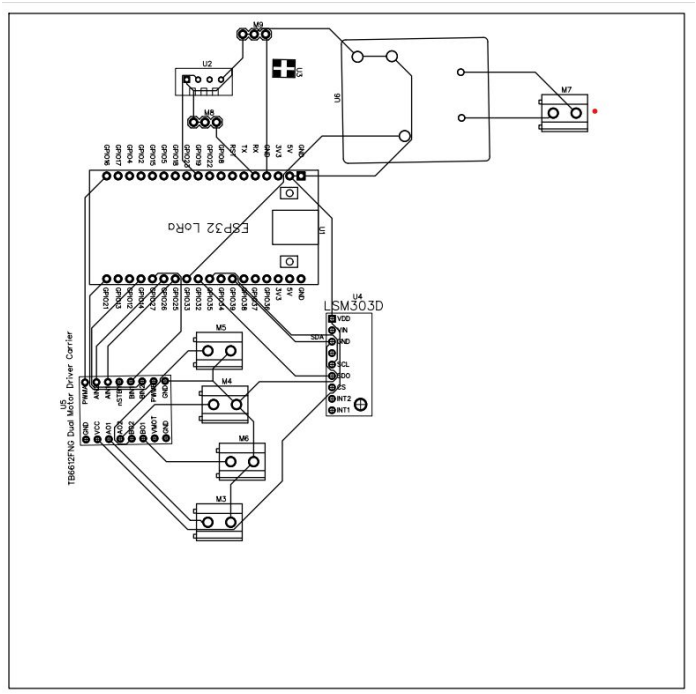


Servo Motor - SG90

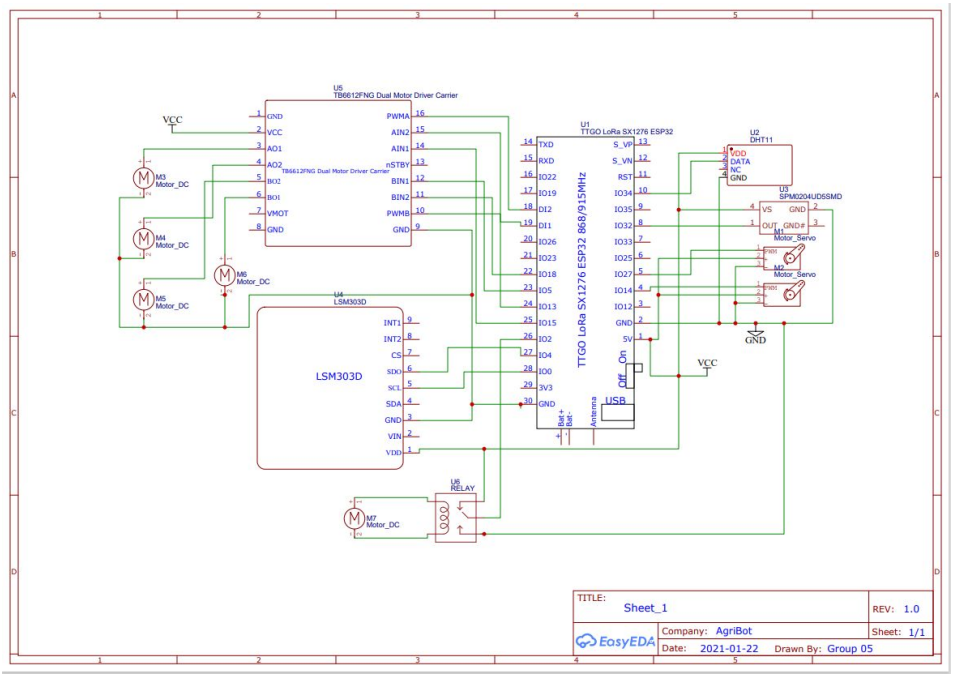


RCWL-1601 Ultrasonic Distance Sensor

3.3.2 PCB & Schematic Designs



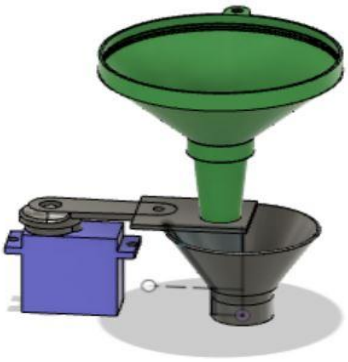
PCB Design



Schematic Design

3.3.3 Other Hardware Parts

Drill BIt - Drill Bit is used to drill the surface to plant seeds.
Motor - Drill
Servo Motor - UP Down the drill
Rack & Pinion - Control the Movements



Seed Container - Storage for the seeds & Direct to relevant places
Funnel - Direct seeds to correct place
Container - Storage for the seeds
Servo Motor - Open & Close the funnel

4.0 Testing

4.1 Software Testing

Testings were done during the development of the app to identify the problems with the mobile app. (Connectivity, User Inputs etc…). Used test cases to check the expected values come as the output. It is easier than manually testing each case over and over again.

4.1.1 Subscribe the Broker

- Checked the subscription - connectivity between the Mobile app and the EC2 Broker.
- This is important because through the Broker only mobile app get the readings from the Robot.
- Though an IP is correct, invalid topic can be subscribed. But this won't happen in the app because topics are assigned when user successfully login to the app (Using correct ID and the Password).
- Subscription is tested using mosquitto broker(Representing Sensors of the Robot) also; using Temperature and Humidity values.

4.1.2. Publish to Broker

- Checked the Publish - connectivity between the mobile app and the EC2 Broker
- It is important to check whether only the correct values are delivered to the Broker from the mobile app.
- Terminal is used as the Robot to check the received values are correct.

4.1.3. User Inputs

- Most critical place the Robot can go wrong is with User Inputs.
- Important to check whether correct values are input by the User to the app.

4.1.4. UI test with espresso

- Additional test to check the functionality of the login page using espresso with test cases.

END