AFAS STORY.

MINI PROJECT LAB (EC681)

Group 1

Aritra Adak 01
Puja Adak 02
Archishman Adhikary 03
Abhranil Aich 04

ABOUT AFAS

The Advanced Farmer Assistance System

- Enhancing Crop Security
- Streamlining Farmer Operations
- Automated Irrigation System
- Remote Monitoring

THE PROBLEM

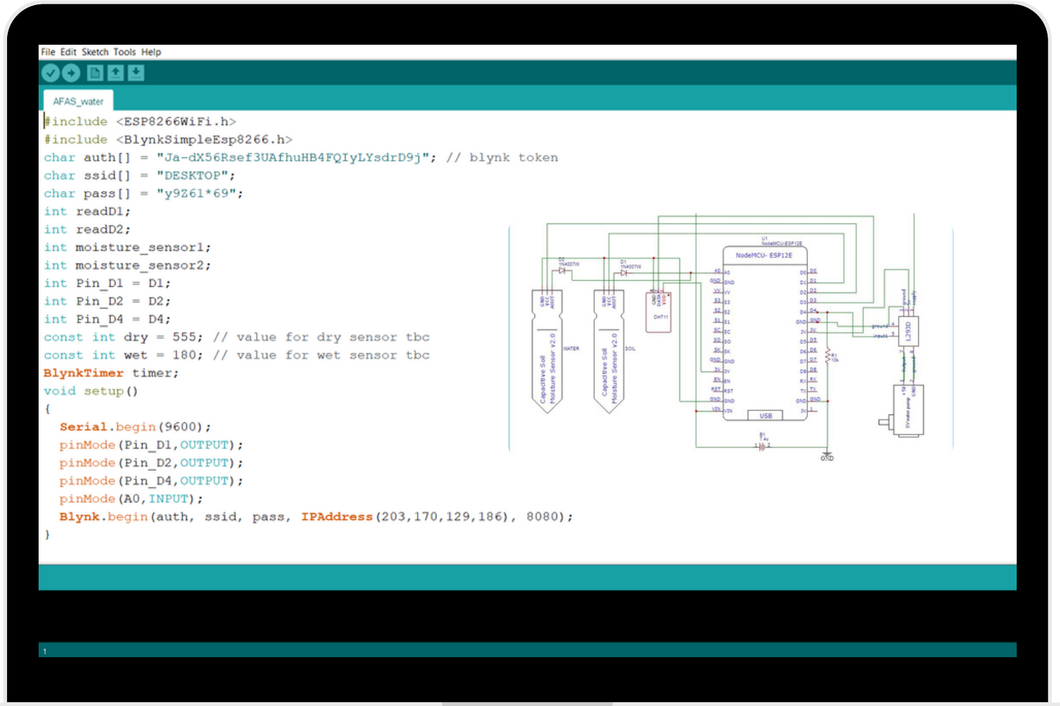
Revolutionizing Agriculture Through Smart Farming Solutions

- Manual labour involved in irrigation and watering of crops.
- Manually monitoring the land all year long
- Dangers of wild animals at night
- Faulty estimations causing over moist or under moist soil both of which are harmfull for crops

SYSTEM OVERVIEW

Components

- ESP8266 NODEMCU
- ESp32 DEVELOPMENT BOARD WITH OV2640 CAMERA MODULE
- FT232RL USB TO TTL
- CAPACITIVE MOISTURE SENSOR
- 1N4007 DIODES
- 5V WATER PUMP WITH 2M HOSE
- L293D MOTOR DRIVER
- PIR SENSOR



System #1

Soil moisture detetction with simple UI/UX in our app, with automatic and manual pump

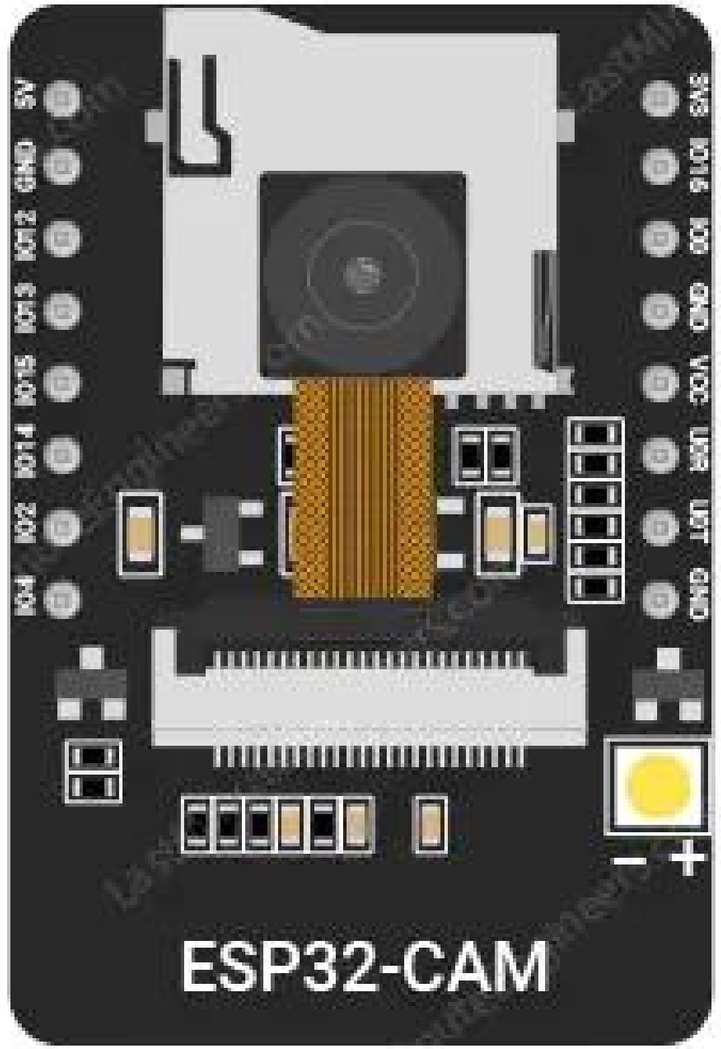
System #2

Remote monitoring through live stream of video feed in the app and also notify if any imotion is detetcted.

REMOTE MONITORING

We are using the ESP32 Camera module which is an advanced IP cam (for prototyping) real life models of these architectures.

- The live video footage of the farm is sent to the farmer's mobile app via the BLYNK Cloud.
- The PIR sensor pauses it's operation while the app and streaming is in use for power management
- While the app is off the PIR sensor continually checks for any abnormal motion and sends a notification



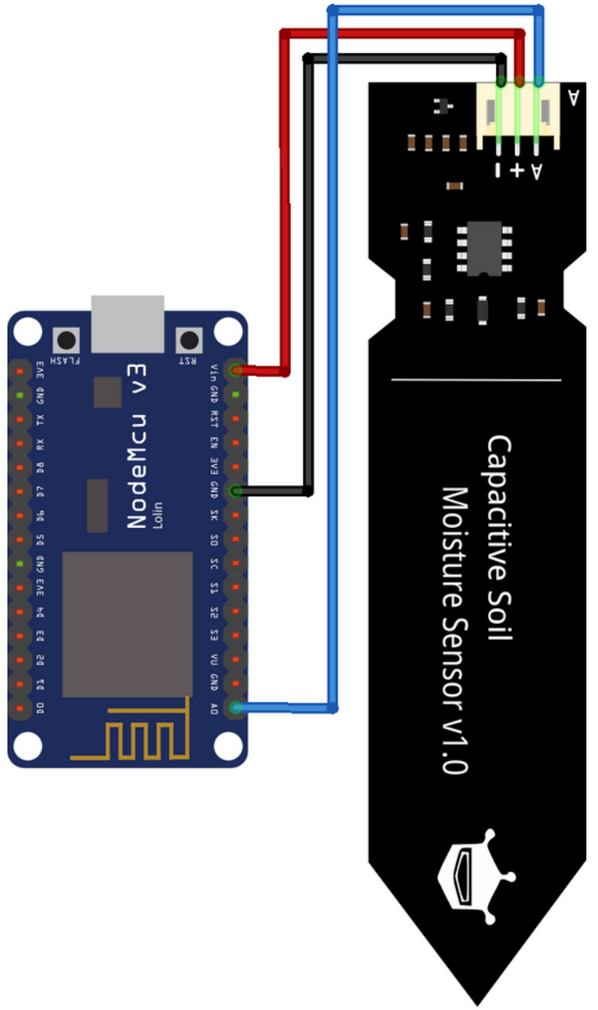
ROLE OF THE PIR SENSOR

- Stands for Passive Infra Red Sensor
- Primarily used to detect any sort of suspicious movement.
- Has an upper hand over Camera Vision during night time.
- Due to it's IR properties is clearly able to mark down and sense motion even at night

- **Ensures Peace of mind of farmers**
- **Round the clock enhanced protection**

AUTOMATIC MOISTURE CONTROL

- The base architecture of this project is based on IOT.
- The micro controller controlling this part of our AFAS is a 32 bit NodeMCU with LX6.
- We chose this over the very commonly used Arduino Uno because of it's inability to connect to the internet by itself.
- Uses 2 Capacitive soil moisture sensors for measuring the soil moisture to the with higher accurancy than the resistive type moisture sensor

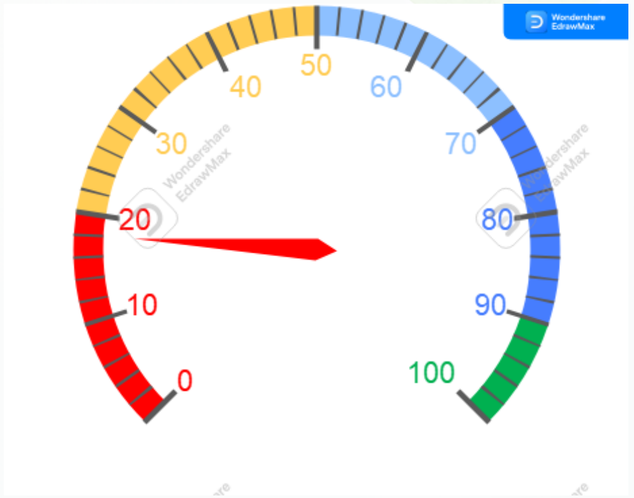


PUMP PROTECTION LOGIC

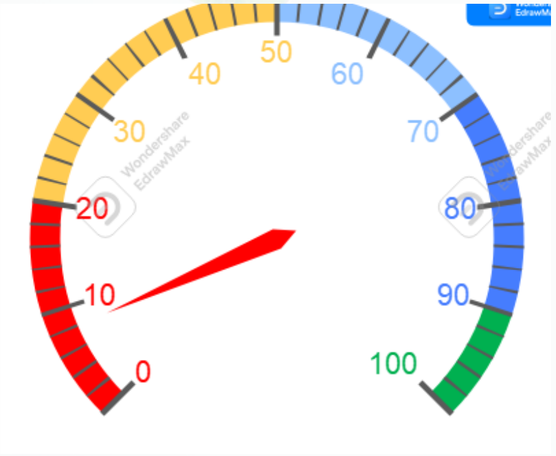
An underrated yet very crucial part of our system

A SPECIAL CASE

- We noticed that if the soil moisture being low triggers the pump, while the water tank is almost empty, technically what we call as a DRY RUN. This will in turn result in the pump motor to burn off.
- To avoid this kind of a situation there are a number of pump protection circuits available.
- But this will increase the cost and load of the system drastically
- So here we are using something that we are calling as a **pump protection logic** where in we will implement the pump protection inside the code itself

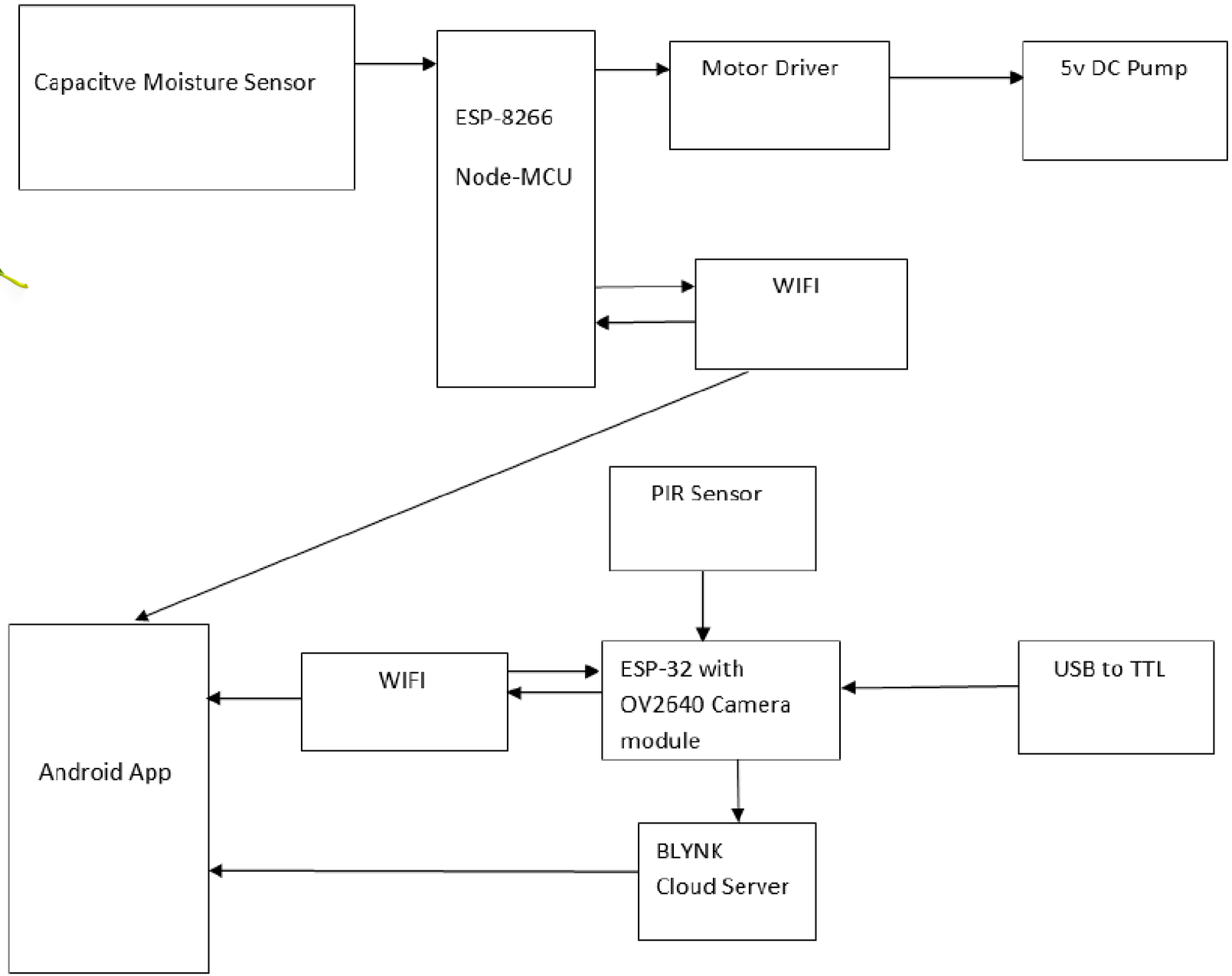


Soil Mositure



Tank Level

BLOCK DIAGRAM



Algorithm 1 Moisture Sensor Algorithm

1: Include libraries: ESP8266WiFi.h, BlynkSimpleEsp8266.h.
2: Declare and initialize variables:
auth = ”Ja-dX56Rsef3UAfhuHB4FQIyLYsdrD9j” ▷ Blynk token
ssid = ”DESKTOP” ▷ Wi-Fi network name
pass = ”y9Z61*69” ▷ Wi-Fi network password
readD1, readD2 ▷ Analog readings
moisture sensor1, moisture sensor2 ▷ Mapped values
Pin D1 = 5, Pin D2 = 4, Pin D4 = 2 ▷ Pin numbers
dry = 555 ▷ Value for dry sensor
wet = 180 ▷ Value for wet sensor
timer ▷ BlynkTimer object
3: procedure setup
4: Initialize serial communication: Serial.begin(9600)
5: Set pinMode for Pin D1, Pin D2, Pin D4 as OUTPUT
6: Set pinMode for A0 as INPUT
7: Begin Blynk connection: Blynk.begin(auth, ssid, pass, IPAd dress(203,170,129,186), 8080)
8: procedure loop
9: Blynk.run
10: First moisture sensor:
11: Turn on Pin D1
12: Delay for 100 milliseconds
13: Read analog value from A0 and store in readD1
14: Turn off Pin D1
15: Delay for 100 millisecond
16: Second moisture sensor:
17: Turn on Pin D2
18: Delay for 100 milliseconds
19: Read analog value from A0 and store in readD2
20: Turn off Pin D2
21: Delay for 100 milliseconds
22: Map readD1 and readD2 to moisture sensor1 and moisture sensor2 re spectively, using wet and dry
23: Print readD1 and readD2 to the serial monitor
24: if moisture sensor2 ≤ 50 then
25: Turn on Pin D4
26: Delay for 10 milliseconds
27: if moisture sensor2 ¿ 50 then
28: Turn off Pin D4
29: Delay for 10 milliseconds
30: Send moisture sensor1 to Blynk virtual pin V5
31: Send moisture sensor2 to Blynk virtual pin V6
32: Run Blynk timer

Algorithm 2 Camera Streaming Algorithm

Include libraries: OV2640.h, WiFi.h, WebServer.h, BlynkSimpleEsp32.h,
WiFiClient.
2: Define constants: CAMERA MODEL AI THINKER, SSID1, PWD1,
PIR PIN.
3.Declare and initialize variables:
auth = ”Ja-dX56Rsef3UAfhuHB4FQIyLYsdrD9j ”
ssid = ”DESKTOP”
pass = ”y9Z61*69”
Declare WebServer object: server(80)
Declare constants: HEADER, BOUNDARY, CTNTTYPE, JHEADER
Declare variables: hdrLen, bdrLen, cntLen, jhdLen
Declare function: handle jpg stream()
Declare function: handle jpg()
Declare function: handleNotFound()
Declare function: checkMotion()
4: procedure SETUP
5.Initialize serial communication: SERIAL.BEGIN(115200)
6: Configure camera parameters in camera config t struct
7.Set PIR PIN as INPUT
8: Begin Blynk connection: address(203,170,129,186), 8080)
9.BLYNK.BEGIN(auth, ssid, pass, IPAd Set interval for checkMotion function to run every second:
TIMER.SETINTERVAL(1000L, checkMotion)
10: Initialize camera with the provided config
11.Configure WiFi mode and connect to the network (SSID1, PWD1)
12: Start the server and define handling functions for specific routes
13.procedure LOOP
14: Handle client requests: SERVER.HANDLECLIENT
Process Blynk commands: BLYNK.RUN
16: Process Blynk timers: TIMER.RUN

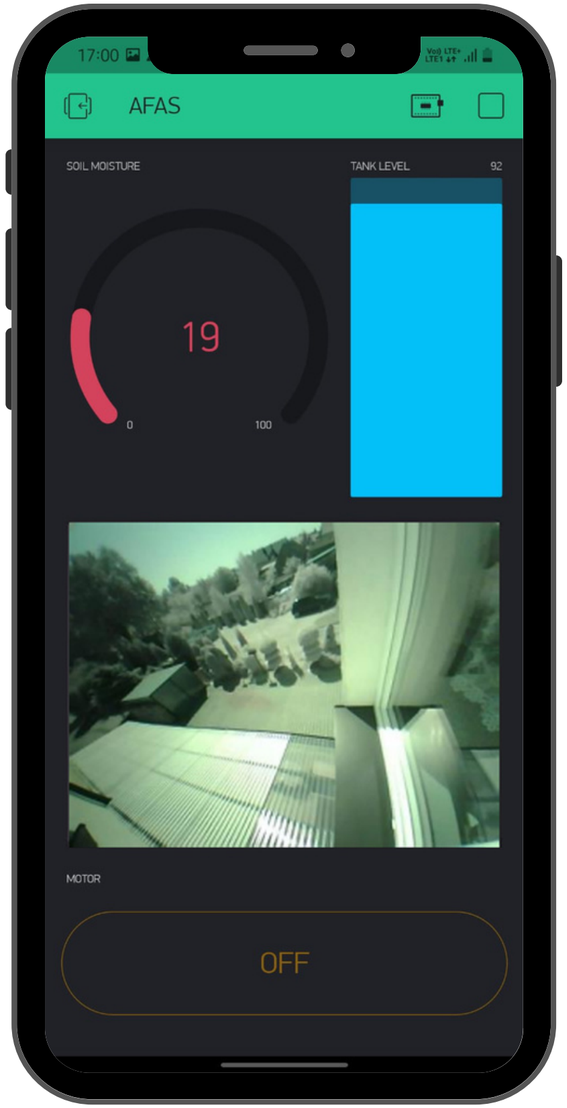
- ▷ Blynk token
- ▷ Wi-Fi network name
- ▷ Wi-Fi network password

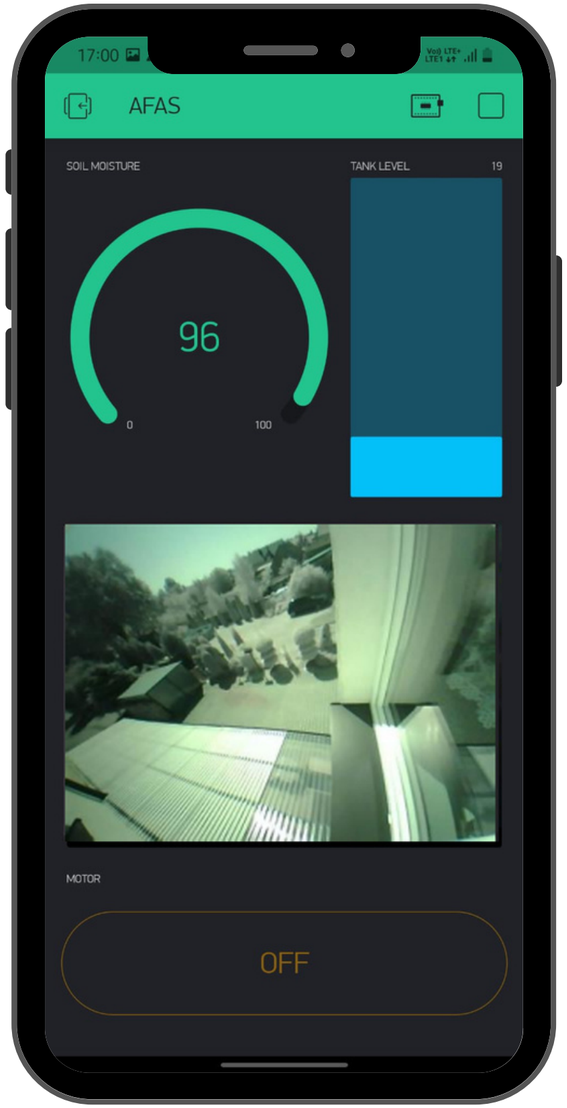
Future Prospects

The Advanced Farmer Assistant System is on the brink of transforming agriculture by blending state-of-the-art technologies with the invaluable knowledge of traditional farming methods. Through the integration of AI, machine learning and data analytics, this system provides an extensive range of tools and functionalities that enhance efficiency, support sustainable practices, and empower farmers to excel in an intricate and fiercely competitive sector. With its capacity to bolster crop yield, minimize environmental harm, and augment decision-making prowess, the Advanced Farmer Assistant System presents an incredibly bright future for the agricultural industry.

CONCLUSION

The Advanced Farmer Assistant System transforms agriculture by leveraging advanced technologies to drive productivity, sustainability, and informed decision-making. With its comprehensive features and capabilities, it paves the way for a more efficient and successful future in farming.





THANK YOU

Any Questions?