AGHMCS – Smart Irrigation System

Smart Irrigation System in AGHMCS :

The smart irrigation system can be done by measuring the moisture content in the soil. Normally, in solids, electrical conduction occurs due to the electrons freely moving through the conductors. Electrical conduction is the ability of the material to conduct electricity. But in liquids, current flows due to the movement of charged particles (i.e.) ions. Ionic compounds dissolved in water as the result of which they increase the conductivity of water. This principle is used to measure the moisture content of water in the soil.

The conductivity of the solution depends on,

- Concentration of ions in soil.
- Temperature of ions.
- Nature of ions.

For the moisture content to be high in soil, there must be high ionic compounds dissolved in water. So that it leads to high electrical conductivity of soil.

To measure the soil moisture YL-38 interface and YL-69 probes are used. The YL-38 interface is the resistant sensor. The probes measure the resistance of current in the soil. The moisture content is inversely proportional to the resistant to the flow of current. If the moisture content is low, the sensor module output is high level of resistance.

After measuring the water content in the soil, if the moisture content is low, the water should be pumped automatically. For this process, a water pump is connected along with the motor driver. Centrifugal pumps are the most preferred pumping devices in the hydraulic world. The electric motor is used to convert the electrical energy into mechanical energy.

The microcontroller Arduino UNO board operated on the voltage of 5V. But the motor needs 9V to run. For this purpose we are using the controlling circuit called the motor driver. The motor driver is an interface between the microcontroller and the motor. The motor driver used in this system is Motor driver L293D.

Schematic of Smart irrigation system in AGHMCS :

The Arduino UNO is the central driving system of this smart irrigation system module. The microcontroller is connected with three different modules to work precisely.

AGHMCS – Smart Irrigation System

i. Soil moisture sensor and water pump.
ii. ESP8266 Wi-Fi module.
iii. ThingSpeak API.

Soil moisture sensor and water pump:

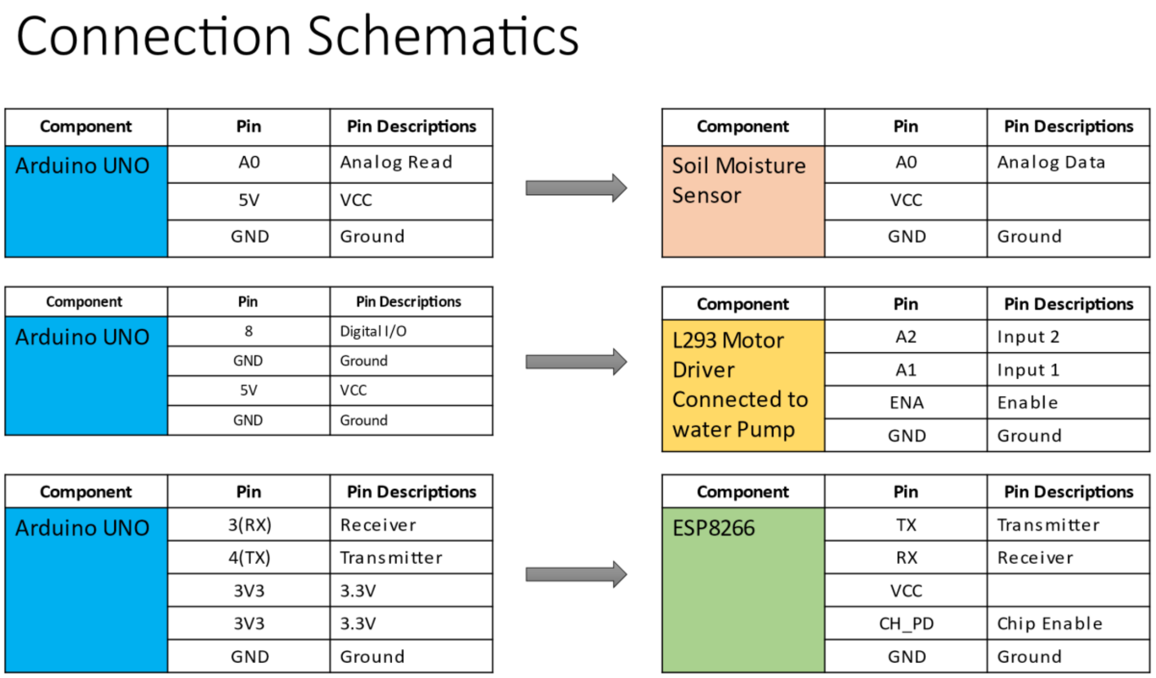
- Read data from soil moisture sensor.
- Activate pump based on sensor data.

ESP8266 Wi-Fi module:

- Connect ESP8266 to Arduino.
- Send AT commands from Arduino to ESP8266.

ThingSpeak API:

- Connect Arduino to Wi-Fi.
- Send data to ThingSpeak.



Connection Schematics

Component	Pin	Pin Descriptions
Arduino UNO	A0	Analog Read
	5V	VCC
	GND	Ground

Component	Pin	Pin Descriptions
Soil Moisture Sensor	A0	Analog Data
	VCC	
	GND	Ground

Component	Pin	Pin Descriptions
Arduino UNO	8	Digital I/O
	GND	Ground
	5V	VCC
	GND	Ground

Component	Pin	Pin Descriptions
L293 Motor Driver Connected to water Pump	A2	Input 2
	A1	Input 1
	ENA	Enable
	GND	Ground

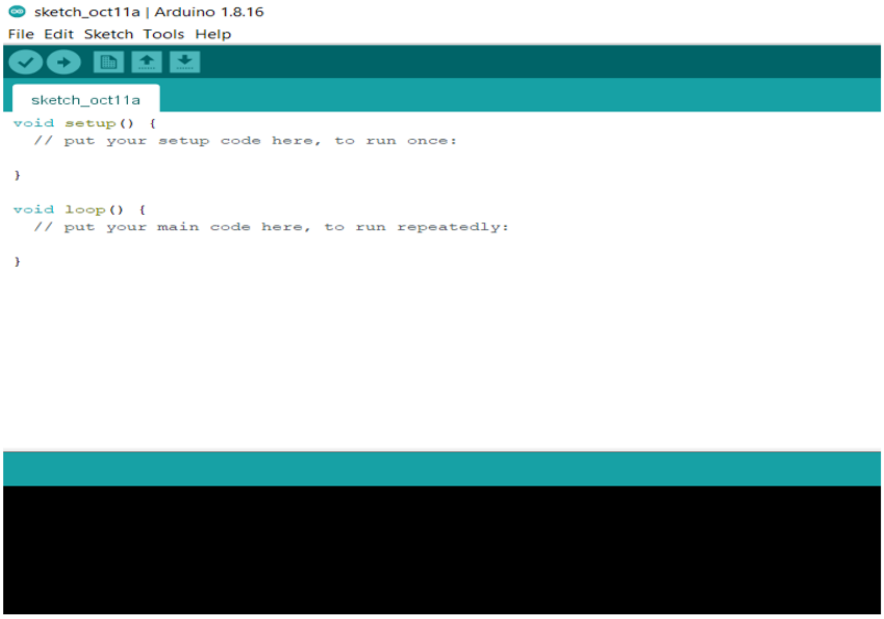
Component	Pin	Pin Descriptions
Arduino UNO	3(RX)	Receiver
	4(TX)	Transmitter
	3V3	3.3V
	3V3	3.3V
	GND	Ground

Component	Pin	Pin Descriptions
ESP8266	TX	Transmitter
	RX	Receiver
	VCC	
	CH_PD	Chip Enable
	GND	Ground

Programming Logic of Smart irrigation System in AGHMCS :

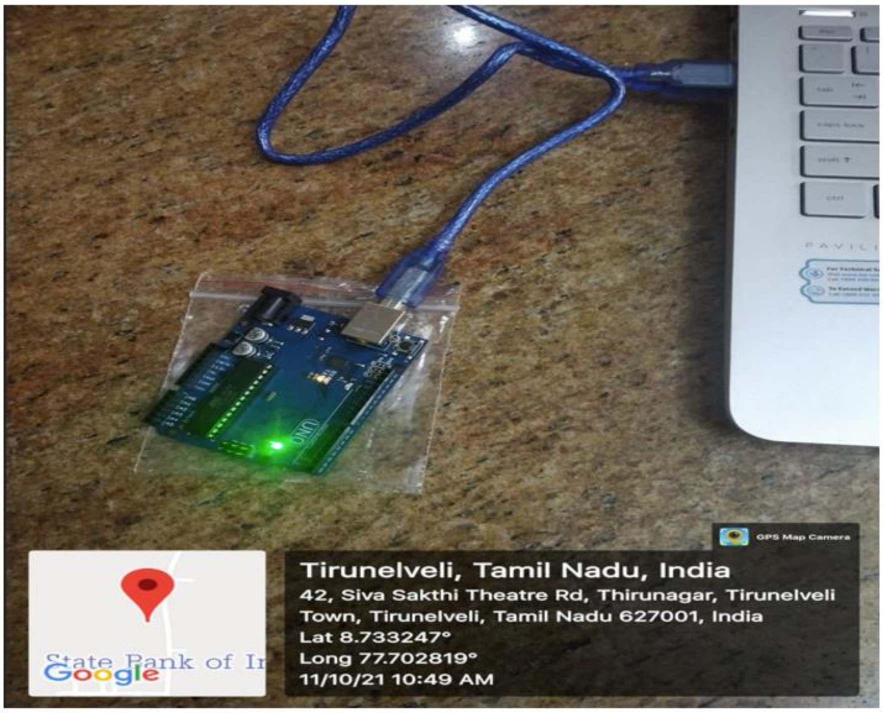
i. Reading data from soil moisture sensor.
ii. Running pump based on the sensor data.
iii. Connect and send commands to ESP8266.
iv. Connect system to internet and send data.

Arduino 1.8.16 Installation

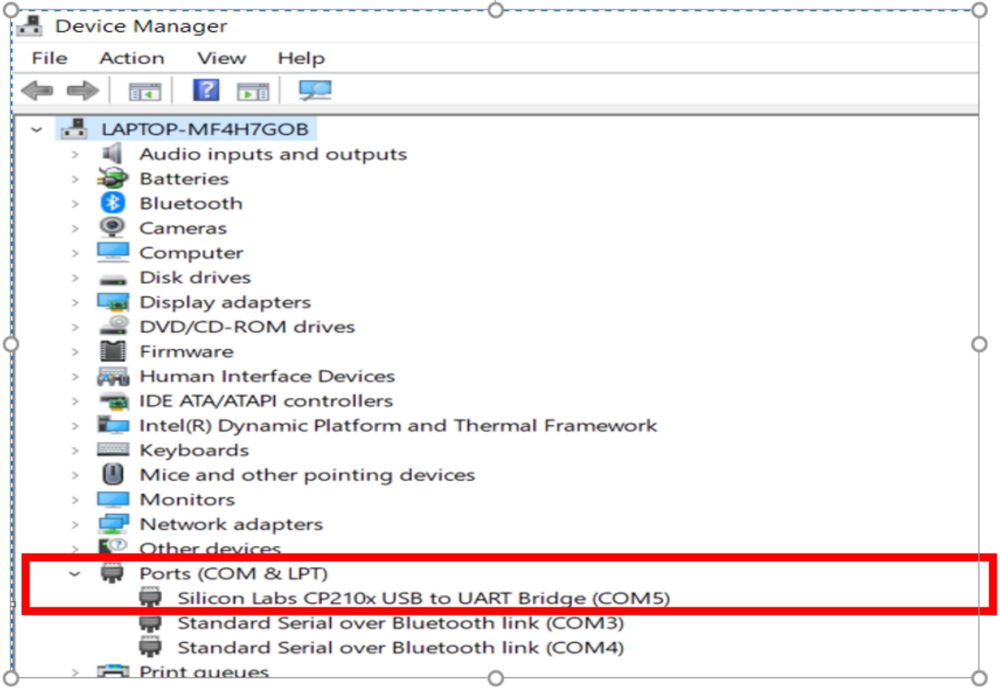


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Interfacing an Arduino

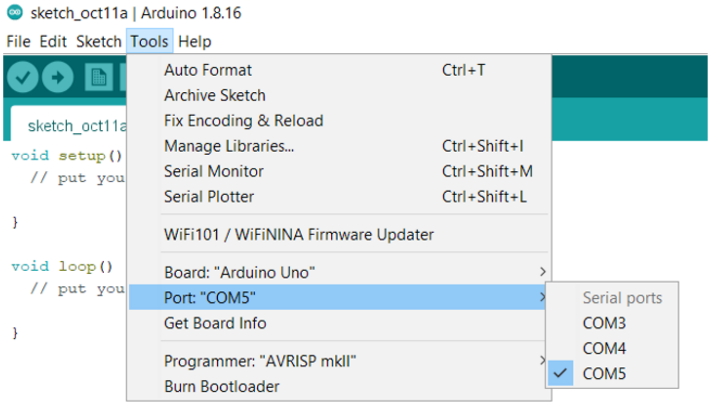

GPS Map Camera
Tirunelveli, Tamil Nadu, India
42, Siva Sakthi Theatre Rd, Thirunagar, Tirunelveli
Town, Tirunelveli, Tamil Nadu 627001, India
Lat 8.733247°
Long 77.702819°
11/10/21 10:49 AM
State Bank of In
Google

Arduino board recognized by computer



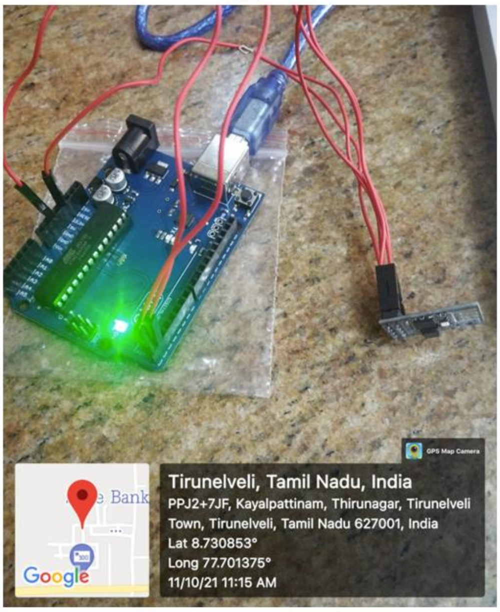
- LAPTOP-MF4H7GOB
  - Audio inputs and outputs
  - Batteries
  - Bluetooth
  - Cameras
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Firmware
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Intel(R) Dynamic Platform and Thermal Framework
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Other devices
  - Ports (COM & LPT)
    - Silicon Labs CP210x USB to UART Bridge (COM5)
    - Standard Serial over Bluetooth link (COM3)
    - Standard Serial over Bluetooth link (COM4)
  - Print queues

Verifying the identification of Arduino Board


sketch_oct11a | Arduino 1.8.16
File Edit Sketch Tools Help
sketch_oct11a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM5"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM3
COM4
✓ COM5

```
void setup()
  // put you

}

void loop()
  // put you

}
```

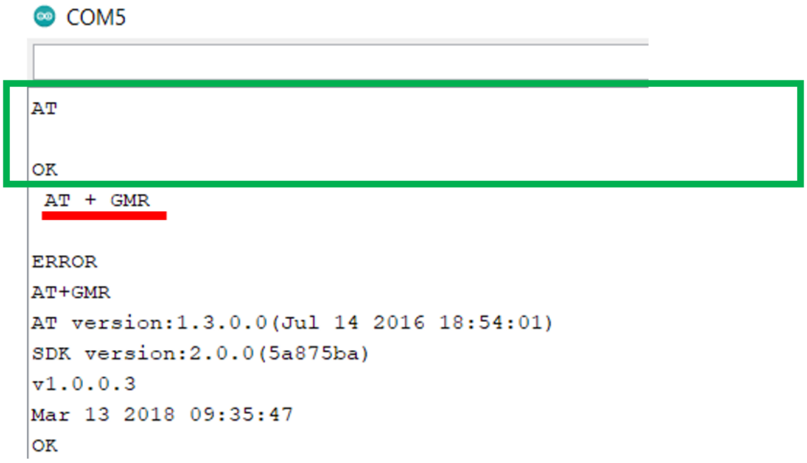
Interfacing ESP8266 with Arduino


GPS Map Camera
Bank
Google
Tirunelveli, Tamil Nadu, India
PPJ2+7JF, Kayalpattinam, Thirunagar, Tirunelveli Town, Tirunelveli, Tamil Nadu 627001, India
Lat 8.730853°
Long 77.701375°
11/10/21 11:15 AM

Connection Schematic

ESP8266	Arduino UNO
VCC	3.3V
CH_PD	
GND	GND
RX	Pin 4
TX	Pin 3

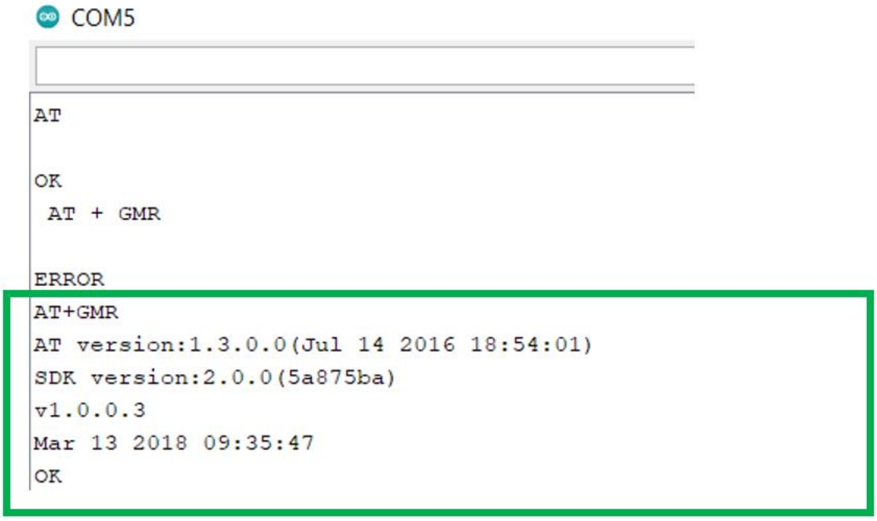
Testing ESP8266 Wi-Fi Module



```
AT

OK
 AT + GMR

ERROR
AT+GMR
AT version:1.3.0.0(Jul 14 2016 18:54:01)
SDK version:2.0.0(5a875ba)
v1.0.0.3
Mar 13 2018 09:35:47
OK
```

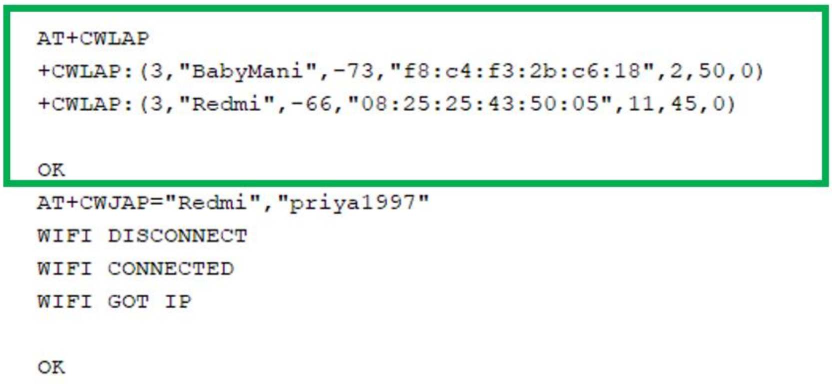
AT – Checks whether wi-Fi module is connected properly with Arduino using attention command.

AGHMCS – Smart Irrigation System



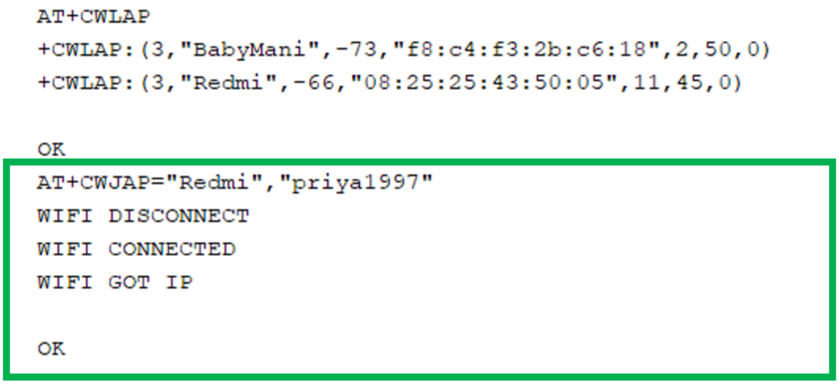
```
AT

OK
 AT + GMR

ERROR
AT+GMR
AT version:1.3.0.0(Jul 14 2016 18:54:01)
SDK version:2.0.0(5a875ba)
v1.0.0.3
Mar 13 2018 09:35:47
OK
```

AT+GMR – gives the version and manufacturer name.



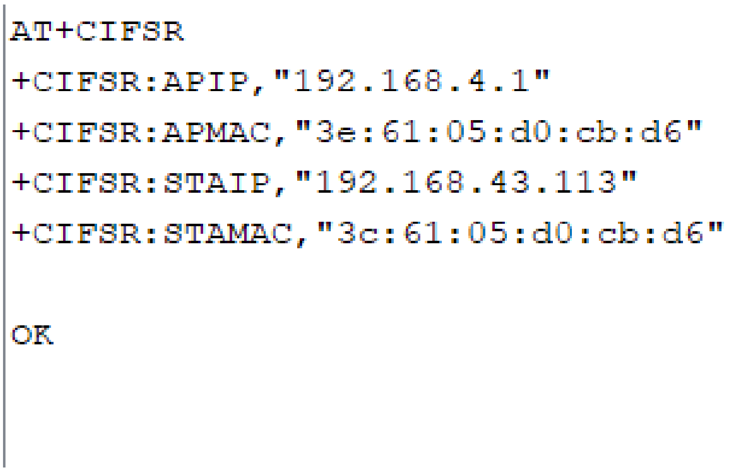
```
AT+CWLAP
+CWLAP:(3,"BabyMani",-73,"f8:c4:f3:2b:c6:18",2,50,0)
+CWLAP:(3,"Redmi",-66,"08:25:25:43:50:05",11,45,0)

OK
AT+CWJAP="Redmi","priya1997"
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP

OK
```

AT+CWLAP – returns the available Wi-Fi around the device.



```
AT+CWLAP
+CWLAP:(3,"BabyMani",-73,"f8:c4:f3:2b:c6:18",2,50,0)
+CWLAP:(3,"Redmi",-66,"08:25:25:43:50:05",11,45,0)

OK
AT+CWJAP="Redmi","priya1997"
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP

OK
```

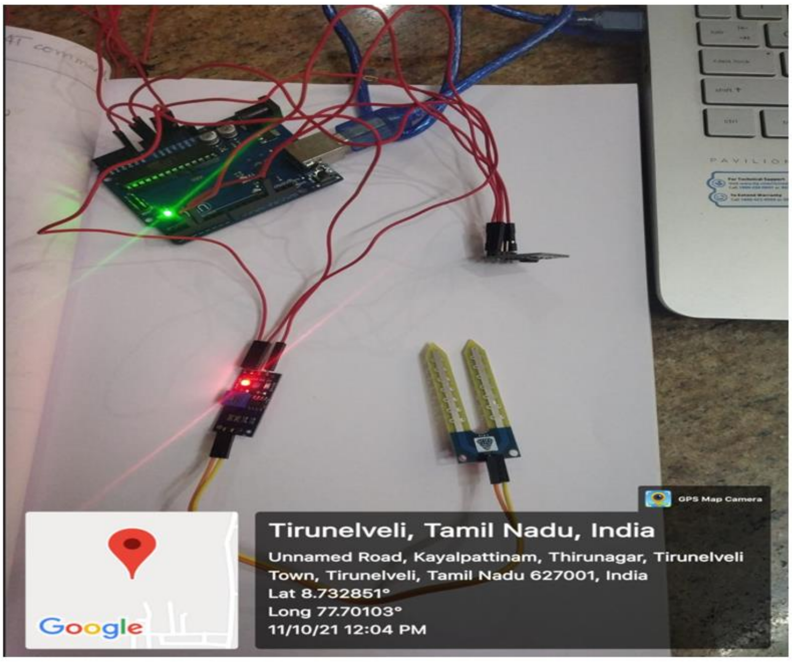
AT+CWJAP=”DEVICE_NAME”, “PASSWORD” – for establishing connection with specified device.



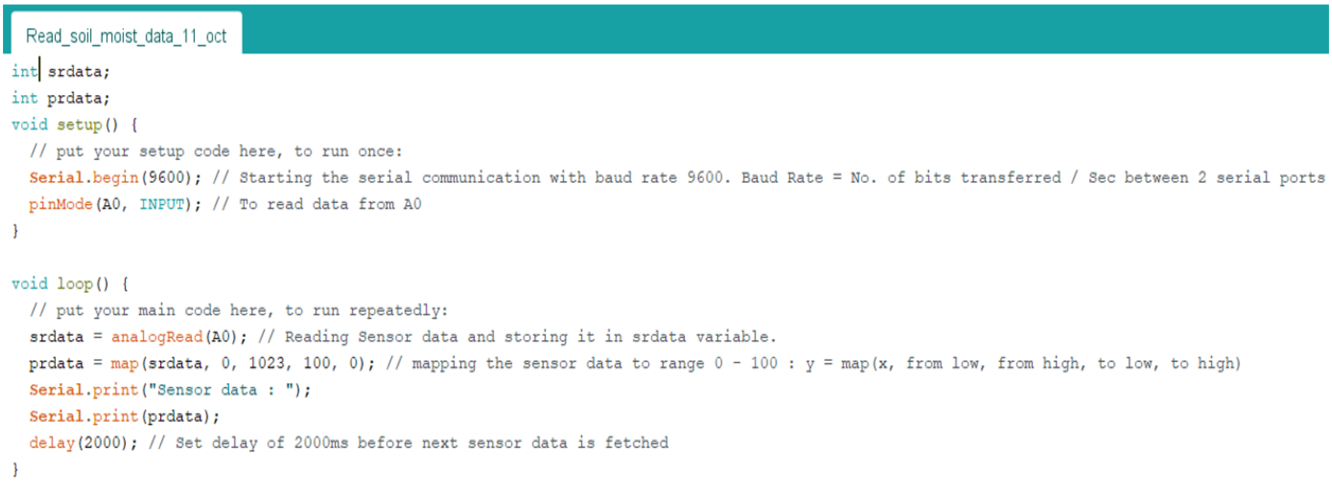
```
AT+CIFSR
+CIFSR:APIP,"192.168.4.1"
+CIFSR:APMAC,"3e:61:05:d0:cb:d6"
+CIFSR:STAIP,"192.168.43.113"
+CIFSR:STAMAC,"3c:61:05:d0:cb:d6"

OK
```

AT+CIFSR – gives the IP address and MAC address.

Interfacing Soil Moisture Sensor with Arduino

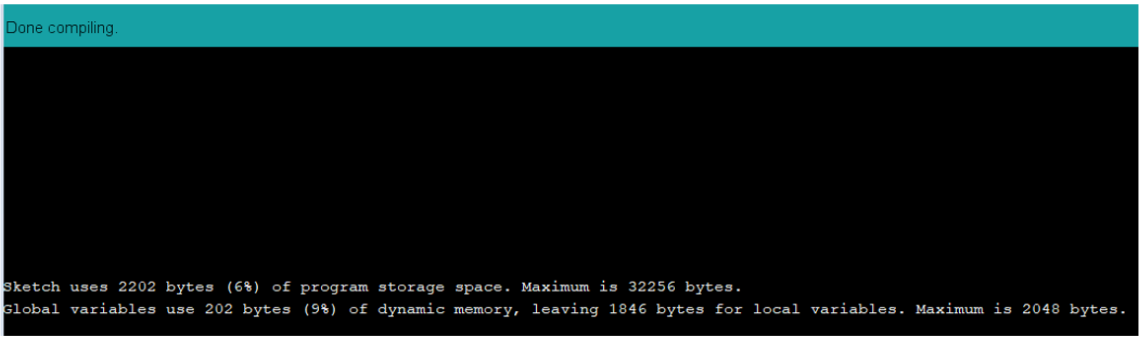

GPS Map Camera
Tirunelveli, Tamil Nadu, India
Unnamed Road, Kayalpattinam, Thirunagar, Tirunelveli Town, Tirunelveli, Tamil Nadu 627001, India
Lat 8.732851°
Long 77.70103°
11/10/21 12:04 PM
Google

Reading Data from Soil Moisture Sensor



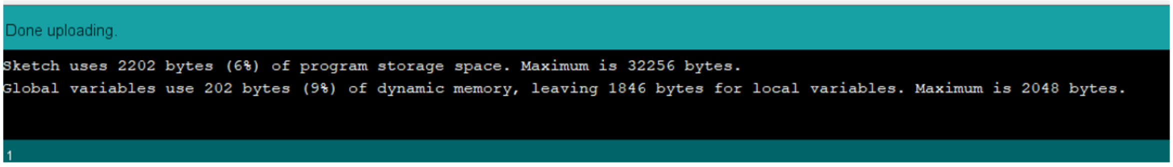
```
int srdata;
int prdata;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Starting the serial communication with baud rate 9600. Baud Rate = No. of bits transferred / Sec between 2 serial ports
  pinMode(A0, INPUT); // To read data from A0
}

void loop() {
  // put your main code here, to run repeatedly:
  srdata = analogRead(A0); // Reading Sensor data and storing it in srdata variable.
  prdata = map(srdata, 0, 1023, 100, 0); // mapping the sensor data to range 0 - 100 : y = map(x, from low, from high, to low, to high)
  Serial.print("Sensor data : ");
  Serial.print(prdata);
  delay(2000); // Set delay of 2000ms before next sensor data is fetched
}
```

Reading Data from Soil Moisture Sensor



Done compiling.

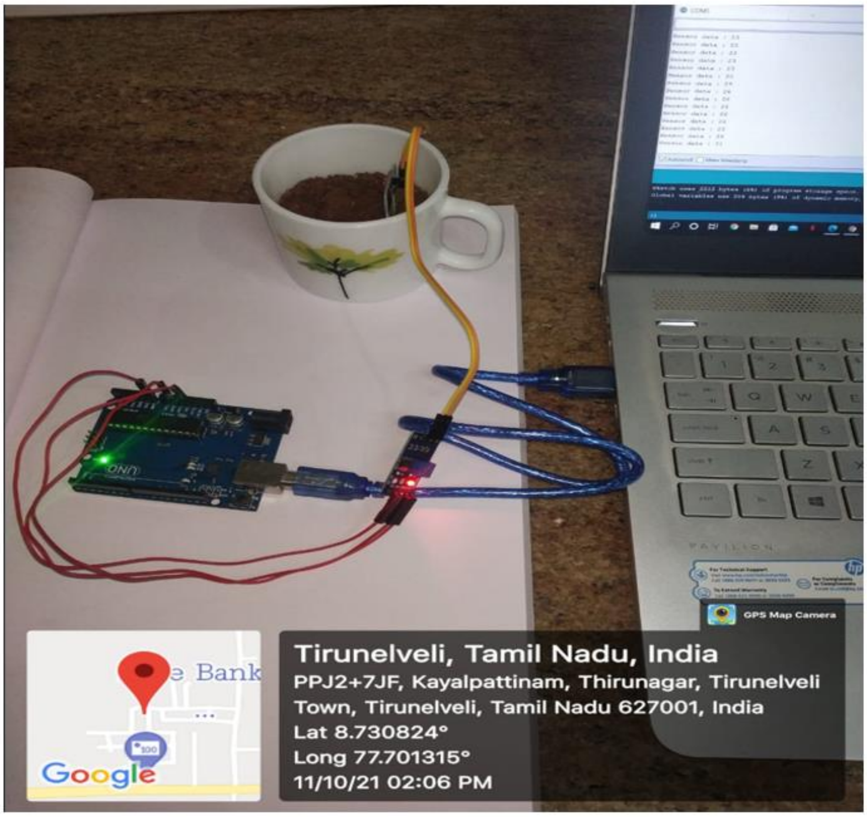
```
Sketch uses 2202 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 202 bytes (9%) of dynamic memory, leaving 1846 bytes for local variables. Maximum is 2048 bytes.
```



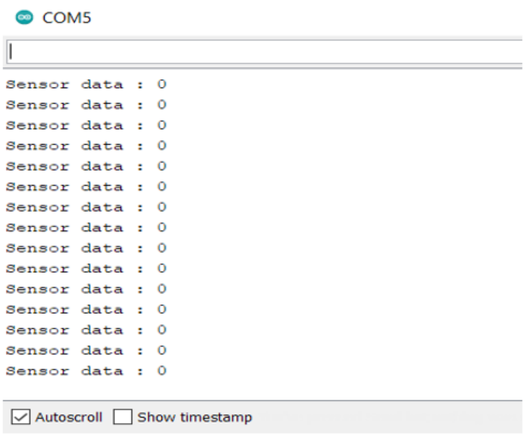
Done uploading.

```
Sketch uses 2202 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 202 bytes (9%) of dynamic memory, leaving 1846 bytes for local variables. Maximum is 2048 bytes.
```

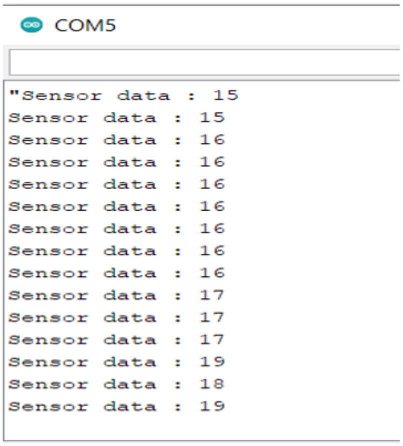
Reading Data from Soil Moisture Sensor


GPS Map Camera
Bank
Google
Tirunelveli, Tamil Nadu, India
PPJ2+7JF, Kayalpattinam, Thirunagar, Tirunelveli Town, Tirunelveli, Tamil Nadu 627001, India
Lat 8.730824°
Long 77.701315°
11/10/21 02:06 PM
PAVILION

Reading Data from Soil Moisture Sensor



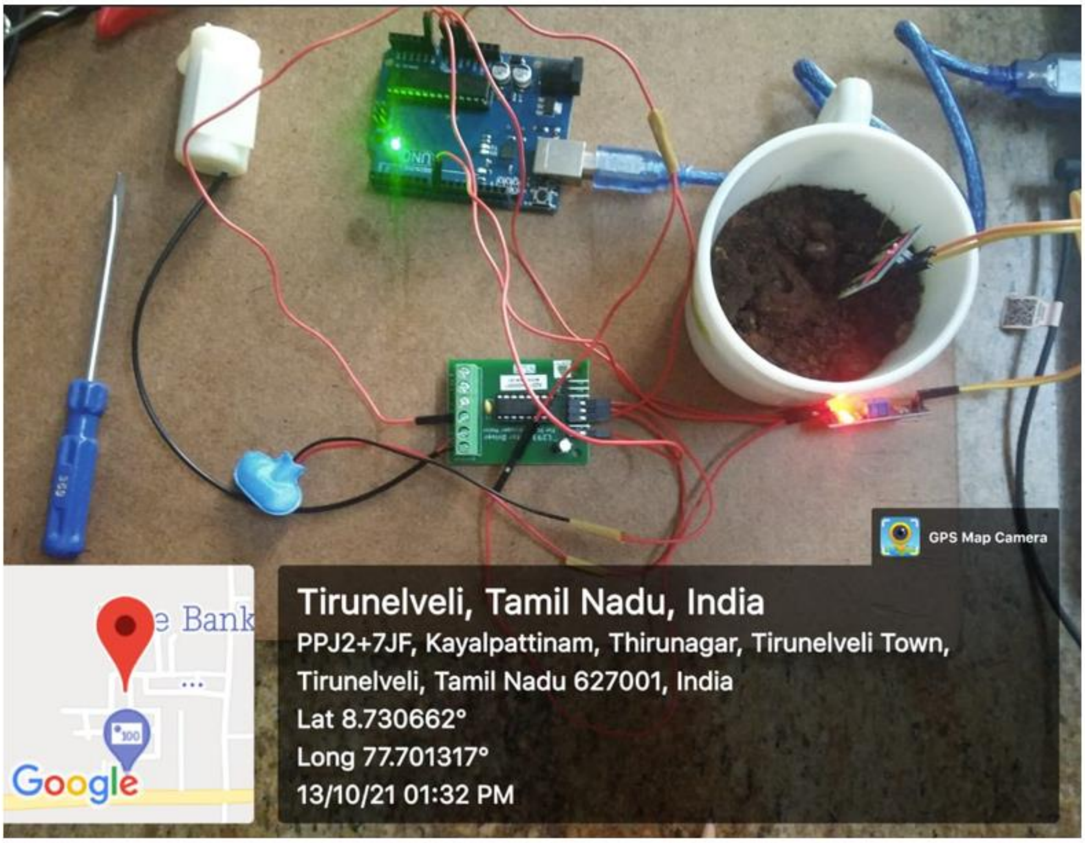
```
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
Sensor data : 0
```



```
"Sensor data : 15
Sensor data : 15
Sensor data : 16
Sensor data : 16
Sensor data : 16
Sensor data : 16
Sensor data : 16
Sensor data : 16
Sensor data : 16
Sensor data : 17
Sensor data : 17
Sensor data : 17
Sensor data : 19
Sensor data : 18
Sensor data : 19
```

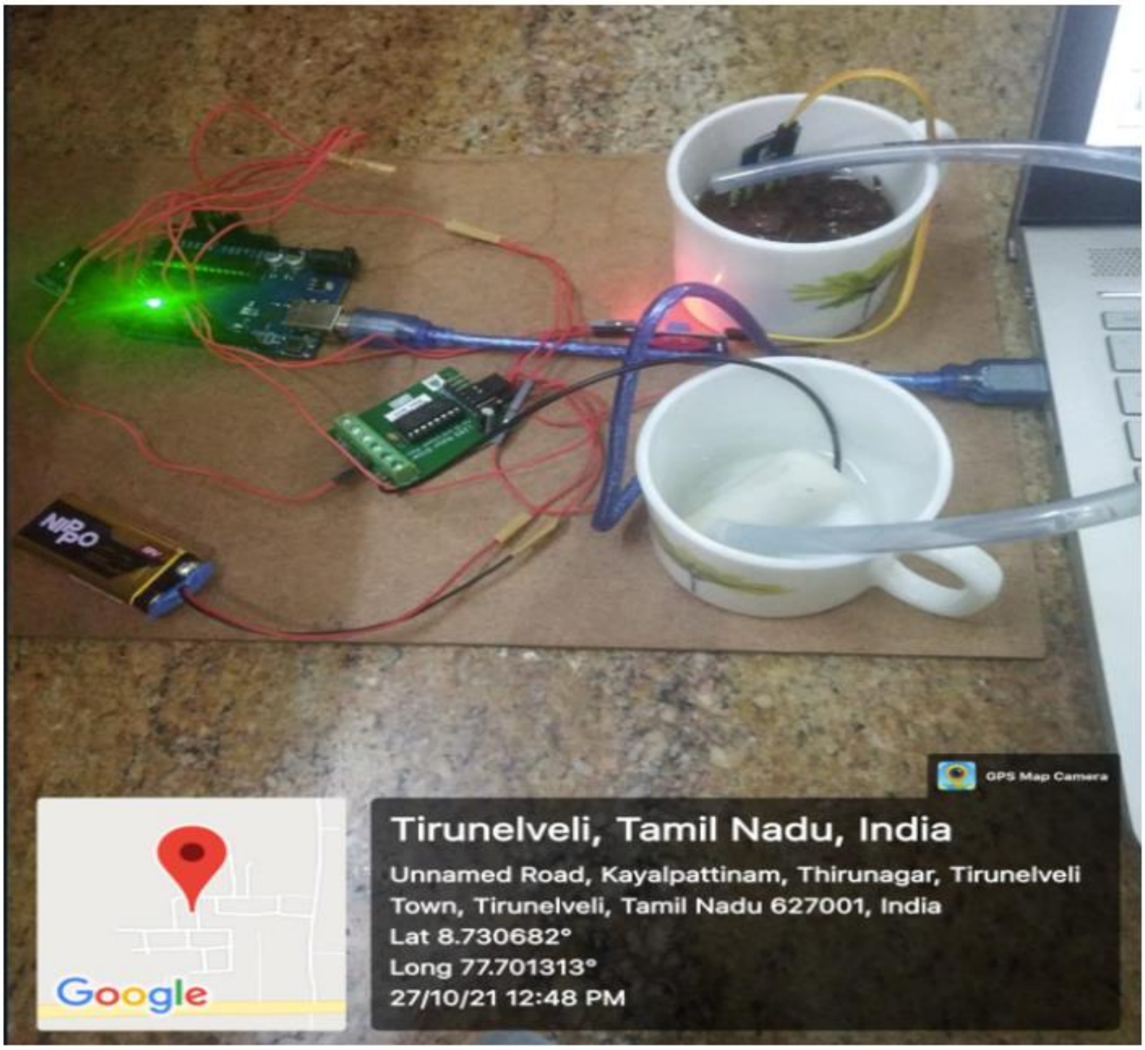
In the above result, while the probe is not inserted to the soil, it shows the sensor data value as 0, when inserted, it shows the reading.

Interfacing Water Pump



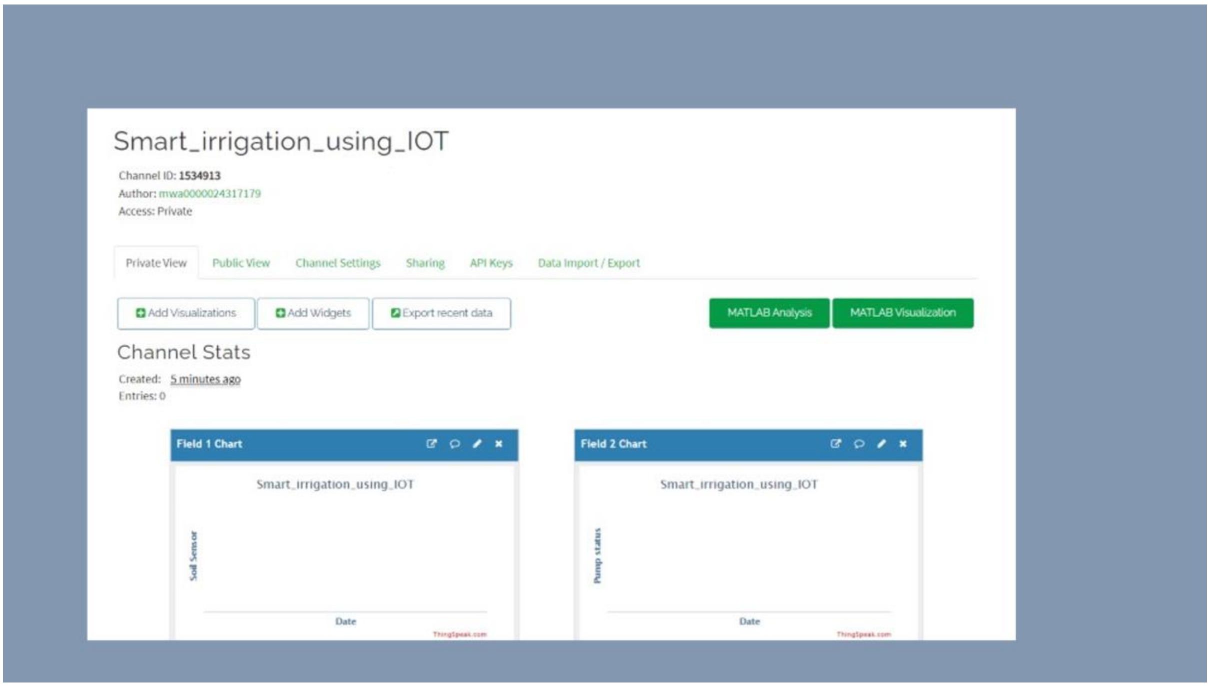
Tirunelveli, Tamil Nadu, India
PPJ2+7JF, Kayalpattinam, Thirunagar, Tirunelveli Town,
Tirunelveli, Tamil Nadu 627001, India
Lat 8.730662°
Long 77.701317°
13/10/21 01:32 PM

Interfacing Water Pump



Tirunelveli, Tamil Nadu, India
Unnamed Road, Kayalpattinam, Thirunagar, Tirunelveli Town, Tirunelveli, Tamil Nadu 627001, India
Lat 8.730682°
Long 77.701313°
27/10/21 12:48 PM

AGHMCS – Smart Irrigation System

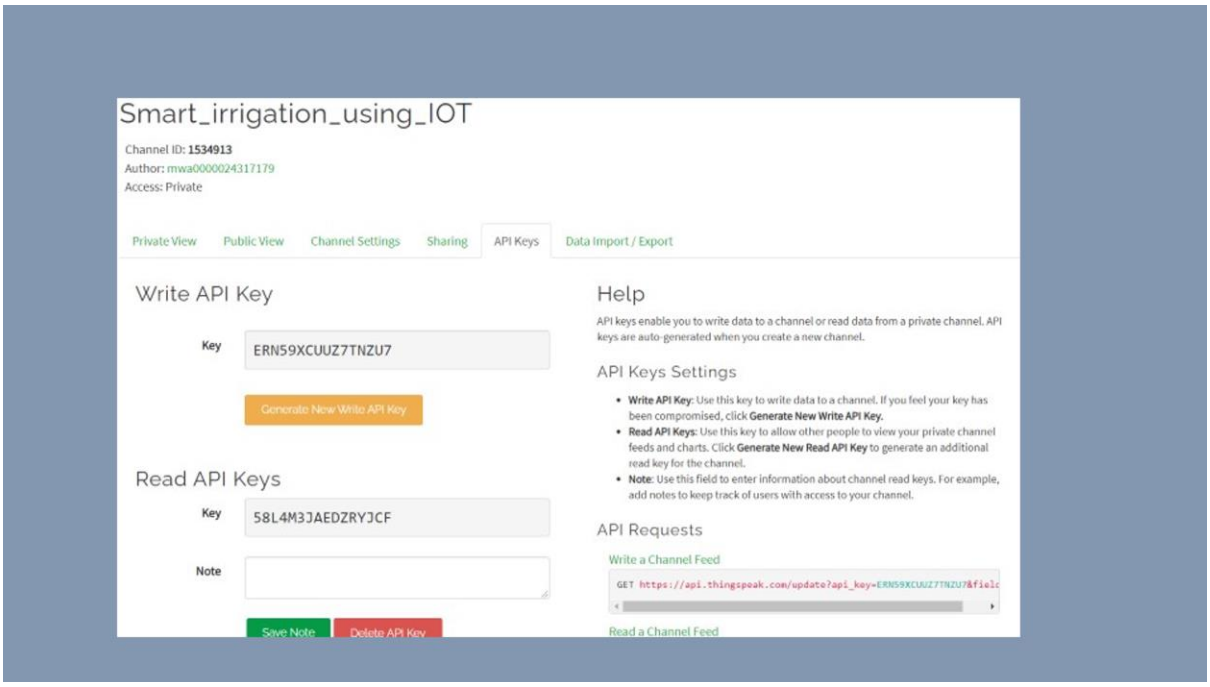


Smart_irrigation_using_IOT

Channel ID: **1534913**
Author: mwa0000024317179
Access: Private

Channel Stats

Created: 5 minutes ago
Entries: 0



Smart_irrigation_using_IOT

Channel ID: **1534913**
Author: mwa0000024317179
Access: Private

Write API Key

Key: ERN59XCUUZ7TNZU7

Read API Keys

Key: 58L4M3JAEDZRYJCF

Note:

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key**: Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys**: Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note**: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

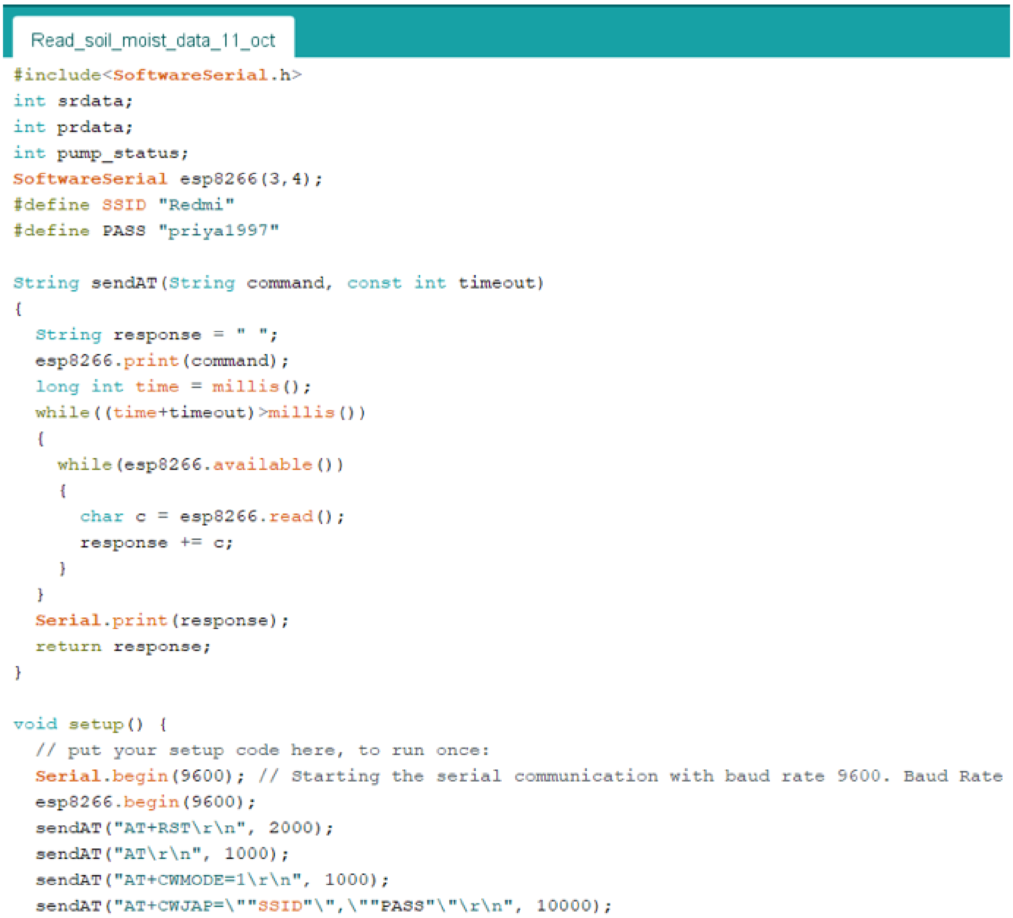
Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=ERN59XCUUZ7TNZU7&fielc
```

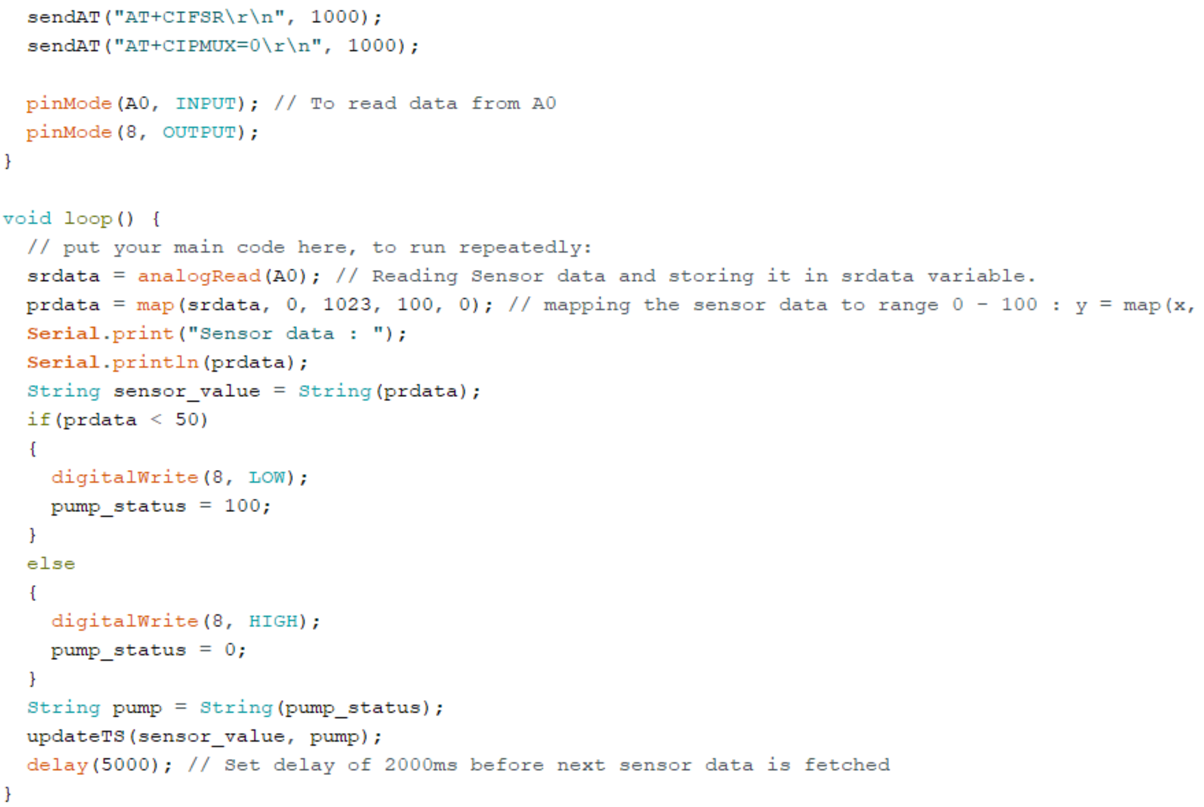
Read a Channel Feed

Channel is established in ThingSpeak API.

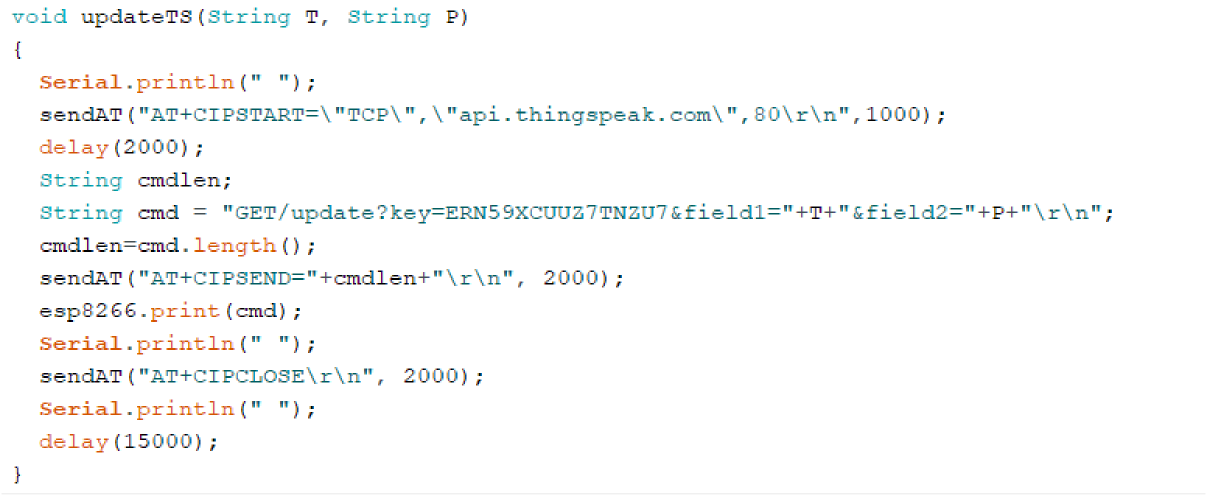
Working of Water Pump



```
#include<SoftwareSerial.h>
int srdata;
int prdata;
int pump_status;
SoftwareSerial esp8266(3,4);
#define SSID "Redmi"
#define PASS "priya1997"

String sendAT(String command, const int timeout)
{
  String response = " ";
  esp8266.print(command);
  long int time = millis();
  while((time+timeout)>millis())
  {
    while(esp8266.available())
    {
      char c = esp8266.read();
      response += c;
    }
  }
  Serial.print(response);
  return response;
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Starting the serial communication with baud rate 9600. Baud Rate
  esp8266.begin(9600);
  sendAT("AT+RST\r\n", 2000);
  sendAT("AT\r\n", 1000);
  sendAT("AT+CWMODE=1\r\n", 1000);
  sendAT("AT+CWJAP=\""SSID"\",\""PASS"\"\r\n", 10000);
```

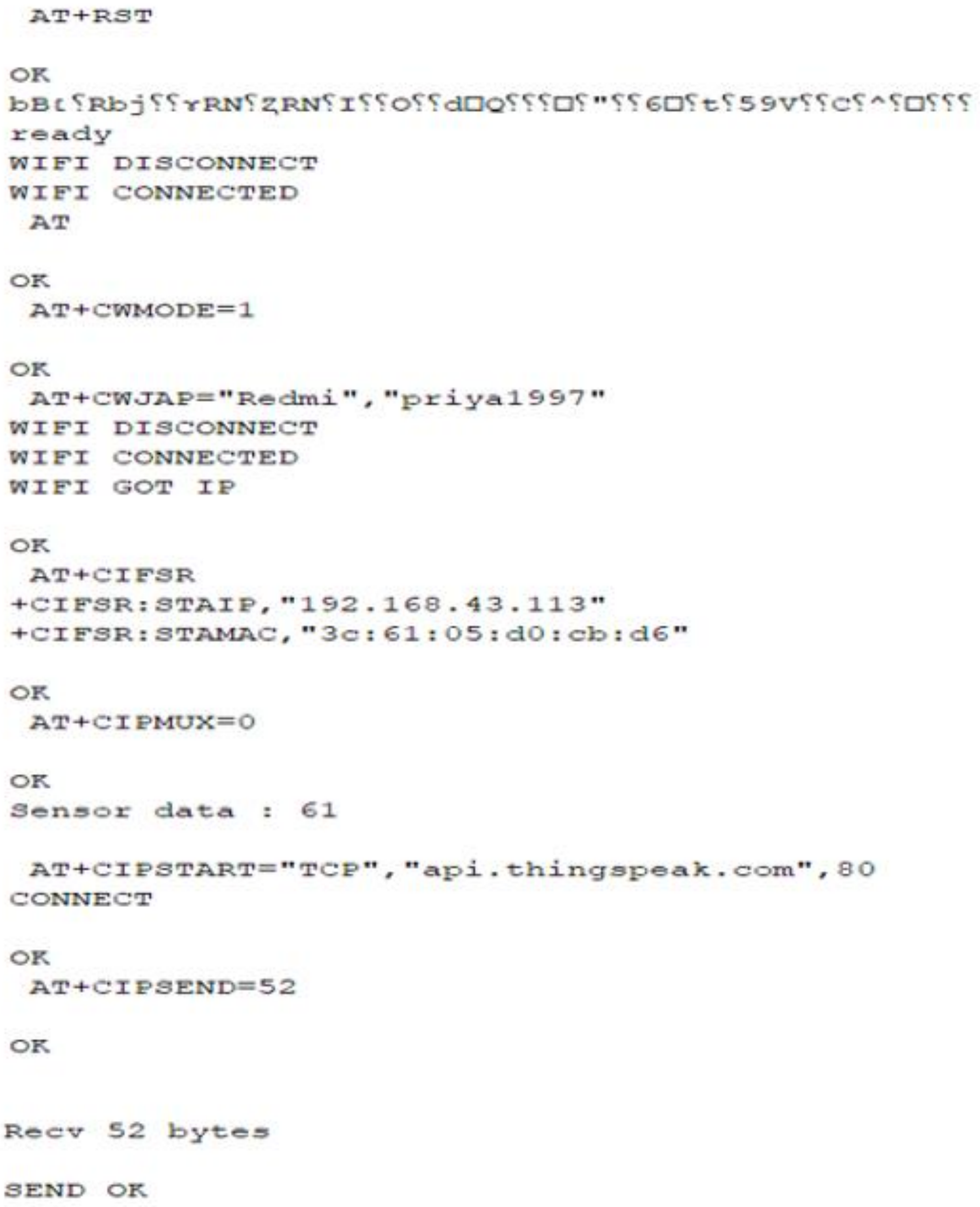


```
  sendAT("AT+CIFSR\r\n", 1000);
  sendAT("AT+CIPMUX=0\r\n", 1000);

  pinMode(A0, INPUT); // To read data from A0
  pinMode(8, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  srdata = analogRead(A0); // Reading Sensor data and storing it in srdata variable.
  prdata = map(srdata, 0, 1023, 100, 0); // mapping the sensor data to range 0 - 100 : y = map(x,
  Serial.print("Sensor data : ");
  Serial.println(prdata);
  String sensor_value = String(prdata);
  if(prdata < 50)
  {
    digitalWrite(8, LOW);
    pump_status = 100;
  }
  else
  {
    digitalWrite(8, HIGH);
    pump_status = 0;
  }
  String pump = String(pump_status);
  updateTS(sensor_value, pump);
  delay(5000); // Set delay of 2000ms before next sensor data is fetched
}
```



```
void updateTS(String T, String P)
{
  Serial.println(" ");
  sendAT("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n",1000);
  delay(2000);
  String cmdlen;
  String cmd = "GET/update?key=ERN59XCUUZ7TNZU7&field1="+T+"&field2="+P+"\r\n";
  cmdlen=cmd.length();
  sendAT("AT+CIPSEND="+cmdlen+"\r\n", 2000);
  esp8266.print(cmd);
  Serial.println(" ");
  sendAT("AT+CIPCLOSE\r\n", 2000);
  Serial.println(" ");
  delay(15000);
}
```

Testing



```
 AT+RST

OK
bBɩʕRbjʕʕɣRNʕʐRNʕIʕʕoʕʕd□ǫʕʕʕ□ʕ"ʕʕ6□ʕtʕ59vʕʕcʕ^ʕ□ʕʕʕ
ready
WIFI DISCONNECT
WIFI CONNECTED
 AT

OK
 AT+CWMODE=1

OK
 AT+CWJAP="Redmi","priya1997"
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP

OK
 AT+CIFSR
+CIFSR:STAIP,"192.168.43.113"
+CIFSR:STAMAC,"3c:61:05:d0:cb:d6"

OK
 AT+CIPMUX=0

OK
Sensor data : 61

 AT+CIPSTART="TCP","api.thingspeak.com",80
CONNECT

OK
 AT+CIPSEND=52

OK


Recv 52 bytes

SEND OK
```