ADIKAVI NANNAYA UNIVERSITY
UNIVERSITY COLLEGE OF ENGINEERING
RAJAMAHENDRAVARAM

AQUAFARM MONITORING SYSTEM USING IoT

Under the esteemed guidance of:
Dr. D LATHA

Presented by:

CH B NEERAJ SINGH

178297601003

4th B Tech CSE

Index

- Abstract
- Introduction
- Existing system
- Proposed system
- Software requirement specification
- Hardware requirement specification
- System analysis and design.
- Implementation
- Proof of work

Abstract

This system Monitors the quality parameters of the farm and alert the user.

This system collect data from the sensors that measures quality Parameters.

By integrating with the web server we can manage this devices

Introduction

Aqua farming (fishes , Prawns…)

Water quality directly proportional to growth of Aquatic Organisms .

Quality may be PH , Water level , Turbidity etc .Which influence the health of organisms

Due to poor monitoring they may end up with a huge loss for farmers .

Existing system

At present they are checking all the parameters manually

.

Manually by chemist centers.

Manual work.

Lot of risks.

Proposed system

Using IoT in the domain of Aqua farming.

Continues Monitoring of Farm Using sensors

Alerting.

To decrease the human intervention.

Software requirement specification

OS :

Microsoft Windows 8/7/9/10,

Linux, Mac OSX 10.8.5 or higher

Wifi

IDE : Arduino IDE,Notepad.

Programming language :

C/Python,PHP,sql,Js

Hardware requirement specification

Working System RAM:>3GB

ESP 32

Turbidity sensor

DO sensor

PH sensor

Water level Sensor

DTH11.

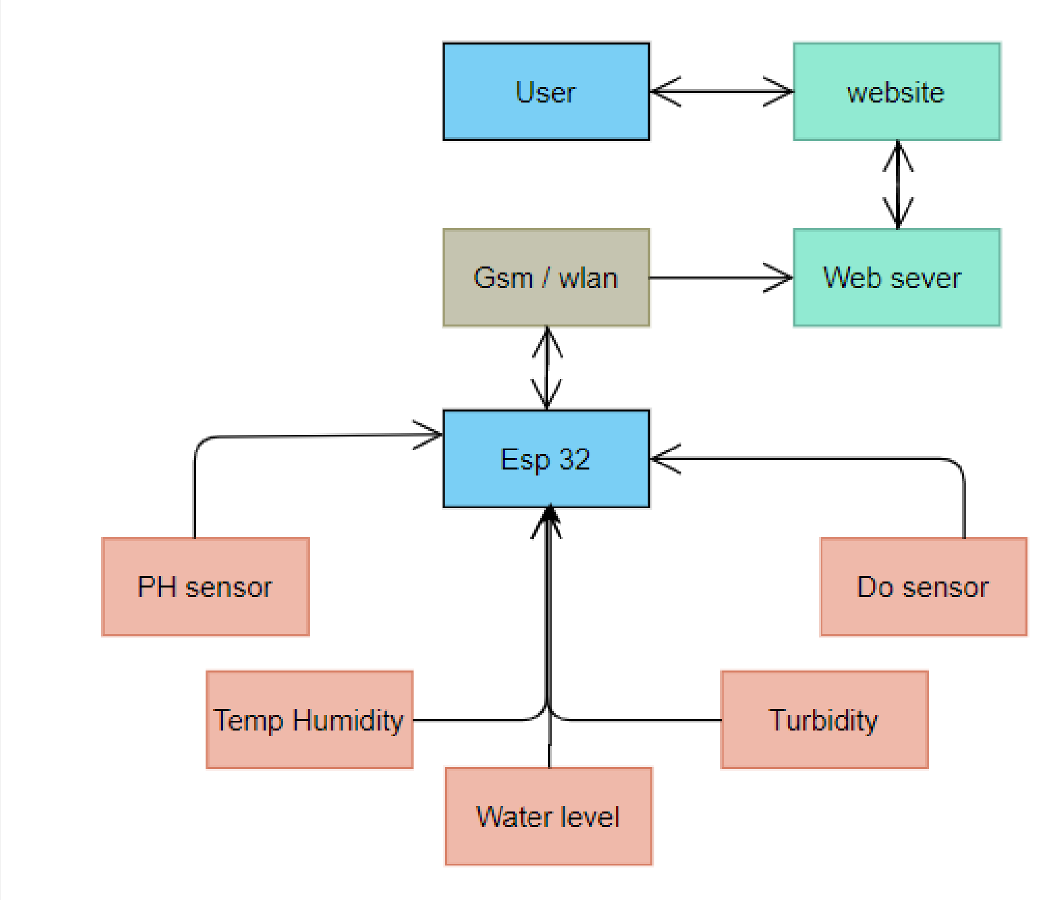
Bread board

Jumper wires

System Design And Analysis

- System Architecture
- UML Diagrams
- Circuit diagram of system
- Components of system

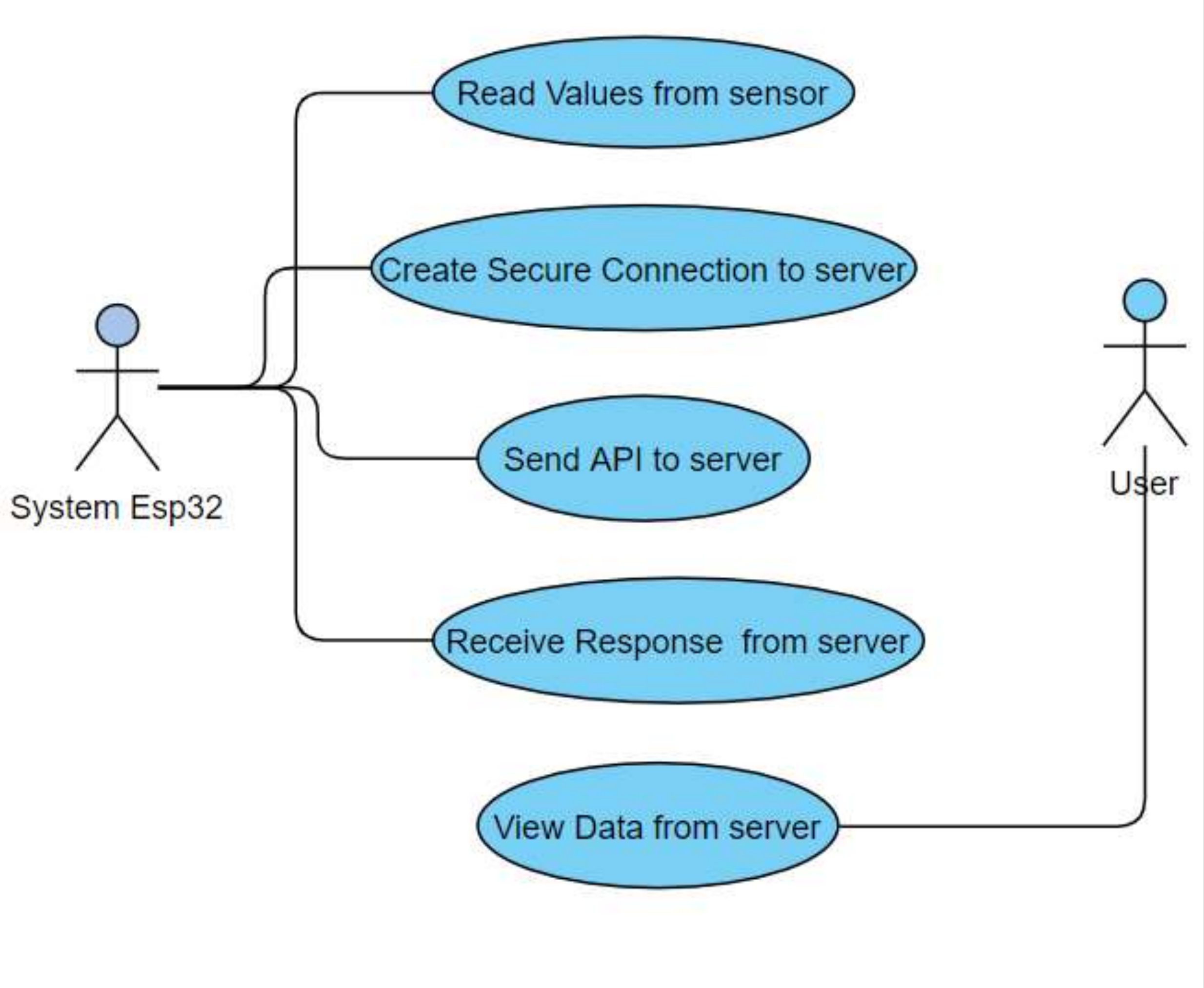
System Architecture



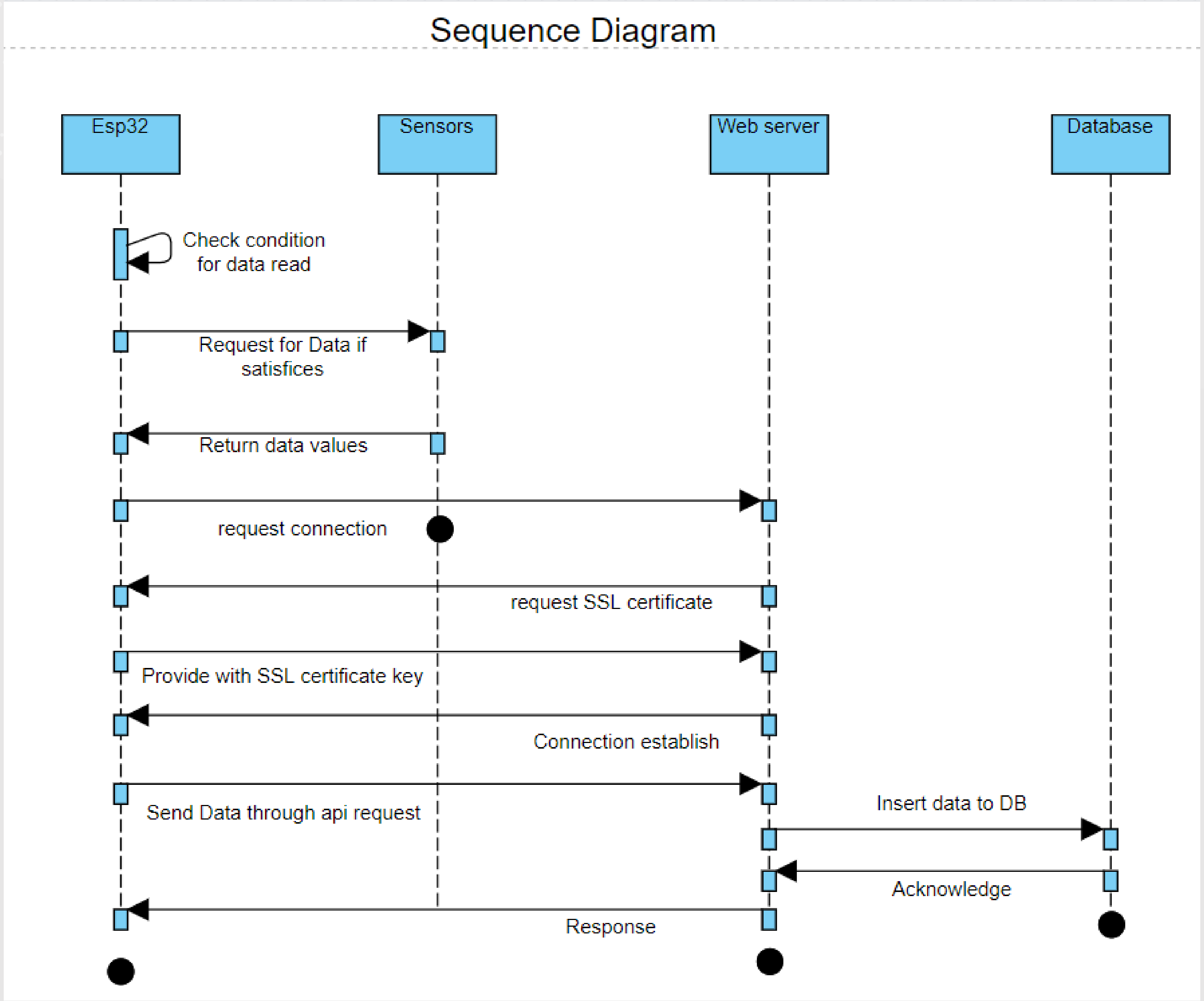
UML Diagram

- Use Case Diagram
- Sequence Diagram
- Activity Diagram

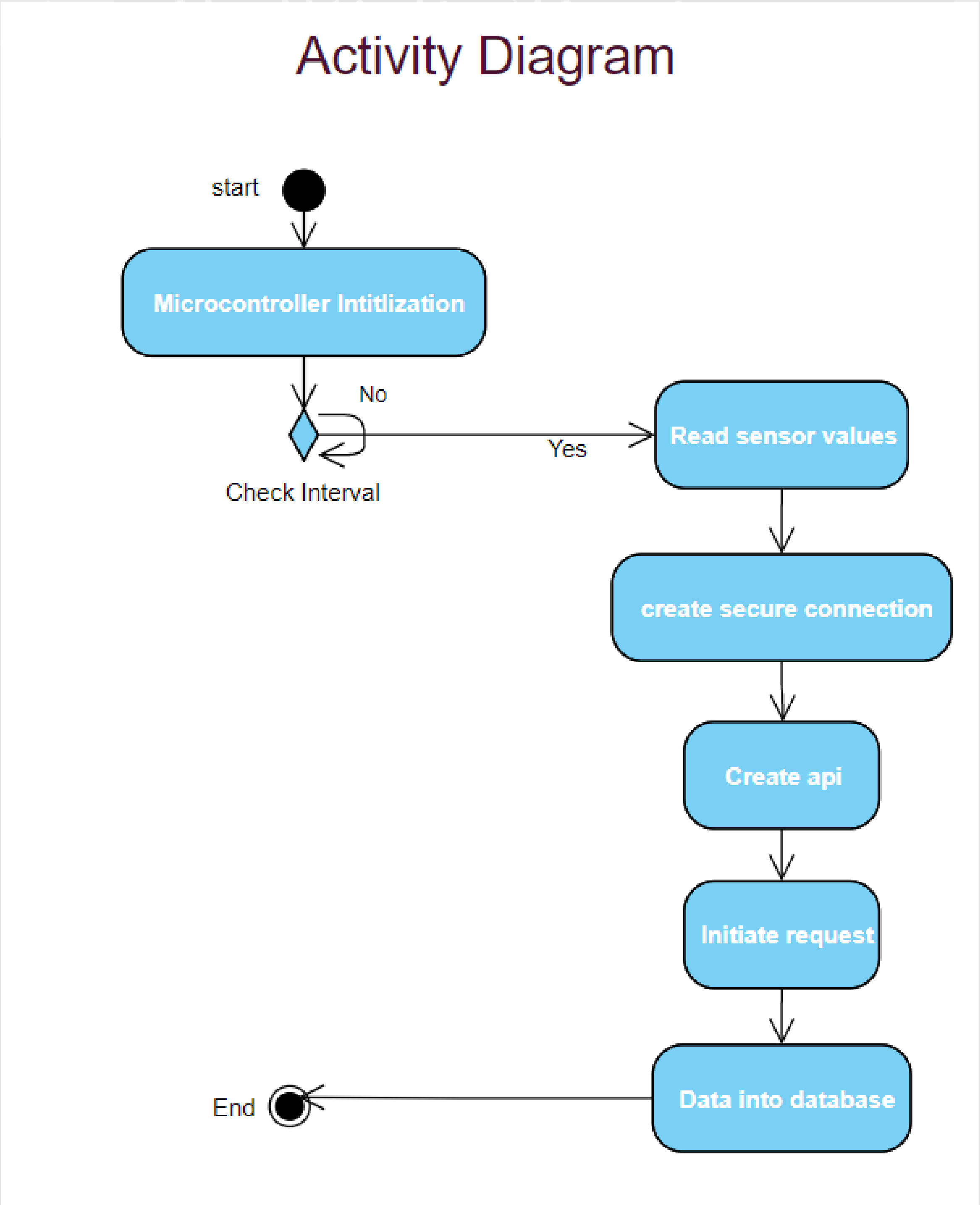
Use Case Diagram



Sequence Diagram



Activity Diagram

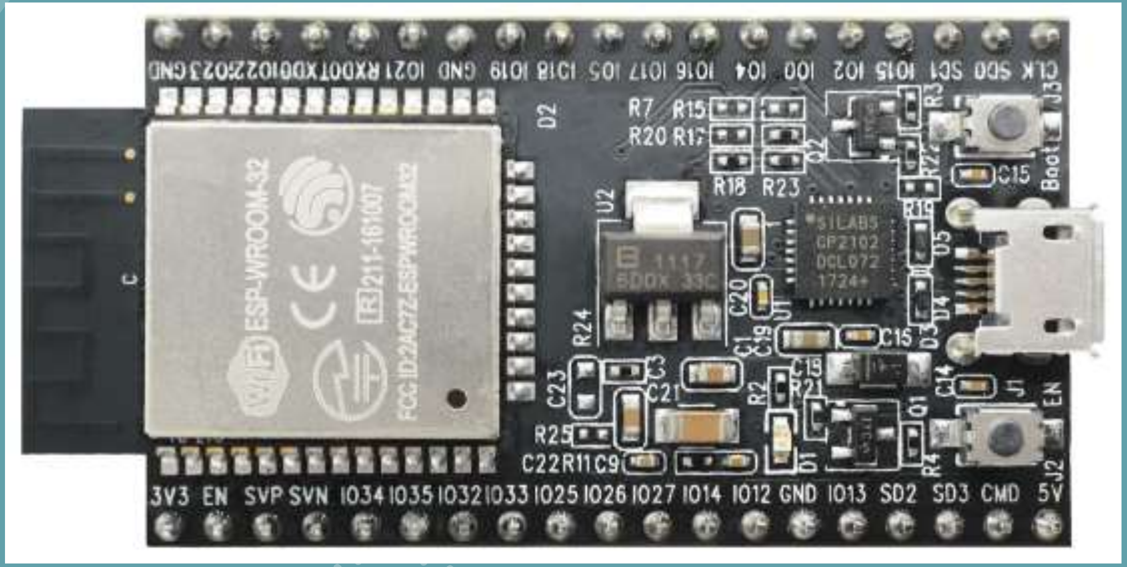


Components of System

- Esp 32
- DTH 11 sensor
- Turbidity sensor
- PH sensor
- DO sensor
- Water level sensor

Components of System

Esp32


GND IO23 IO22 TXD0 RXD0 IO21 GND IO19 IO18 IO5 IO17 IO16 IO4 IO0 IO2 IO15 SD1 SD0 CLK
ESP-WROOM-32
Boot
EN
3V3 EN SVP SVN IO34 IO35 IO32 IO33 IO25 IO26 IO27 IO14 IO12 GND IO13 SD2 SD3 CMD 5V

Tensilica Xtensa 32-bit LX6 microprocessor

Clock Frequency up to 240 MHz

Wi-Fi:

Bluetooth

Ultra Low power consumption 5v and 3V

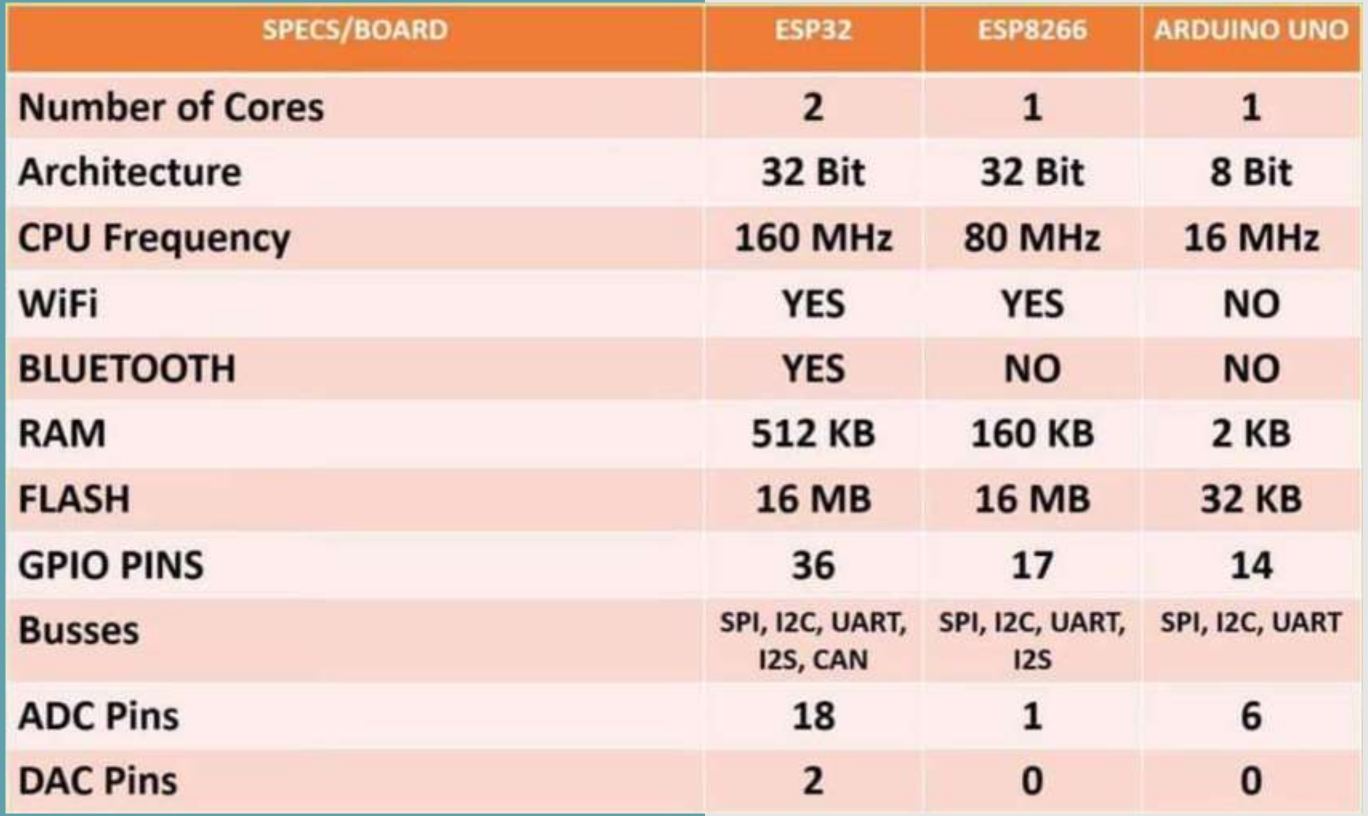
ROM: 448 KIB

SRAM:520KIB

ADCs (analog-to-digital converter), DACs (digital-to-analog converter), I²C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter),

39 pins3UART,External interrupt,34 Gpio

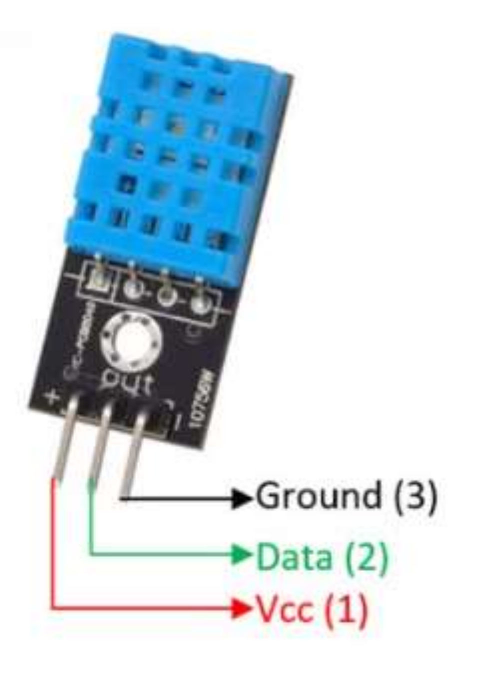
Why Esp32 ?????????



SPECS/BOARD	ESP32	ESP8266	ARDUINO UNO
Number of Cores	2	1	1
Architecture	32 Bit	32 Bit	8 Bit
CPU Frequency	160 MHz	80 MHz	16 MHz
WiFi	YES	YES	NO
BLUETOOTH	YES	NO	NO
RAM	512 KB	160 KB	2 KB
FLASH	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
ADC Pins	18	1	6
DAC Pins	2	0	0

Components of System

Dth11



VCC: Power supply 3.5V to 5.5V

Data: Outputs both Temperature and Humidity

Ground: Connected to the ground of the circuit

Temperature Range: 0°C to 50°C

Humidity Range: 20% to 90%

16bit

NTC Temp sensor(thermister)

Subtract electrodes.

Components of System

Turbidity sensor

Operating Voltage: 5VDC

Operating temp: -30 ° C to 80 ° C

Nephelometric Turbidity Unit

Total suspended solids

Light is passed through a sample of water.

Lens ,Detector

Data

VCC

Ground

Components of System

PH sensor

Operating voltage: 3.3V/5V

Range:0-14PH

Hydrogen-ion activity in water-based solutions

Voltmeter

Electrodes

VCC

GND

NC

S10 or data

Components of System

Do sensor

Operating voltage: 3.3V/5V

Detection Range: 0 -20mg/L

Electrolyte

Membrane

electrodes

VCC

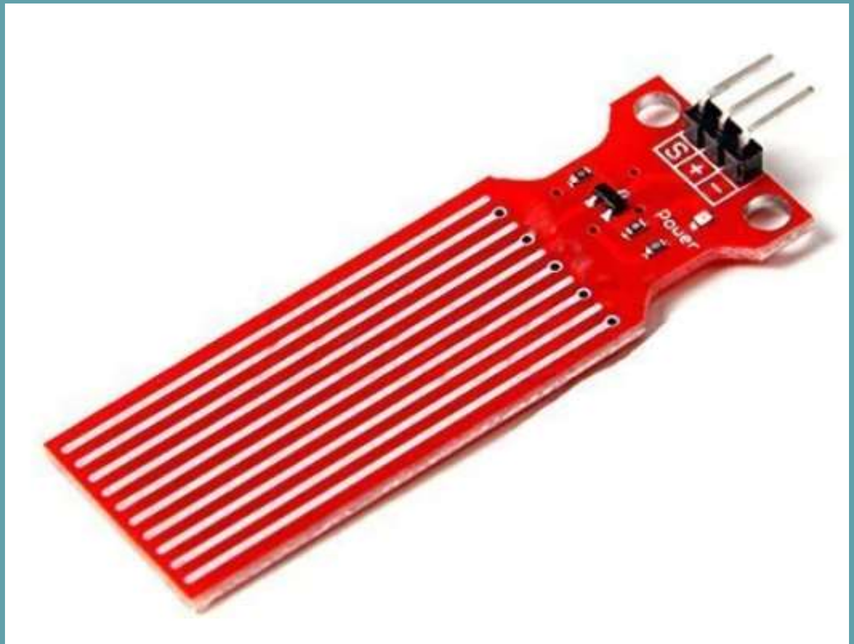
GND

NC

S10 or data

Components of System

Water level sensor



Operating voltage: 3.3V/5V

Humidity: 10% -90% non-condensing

Operating temperature: 10°C-30°C

Parallel conductors as variable resistor

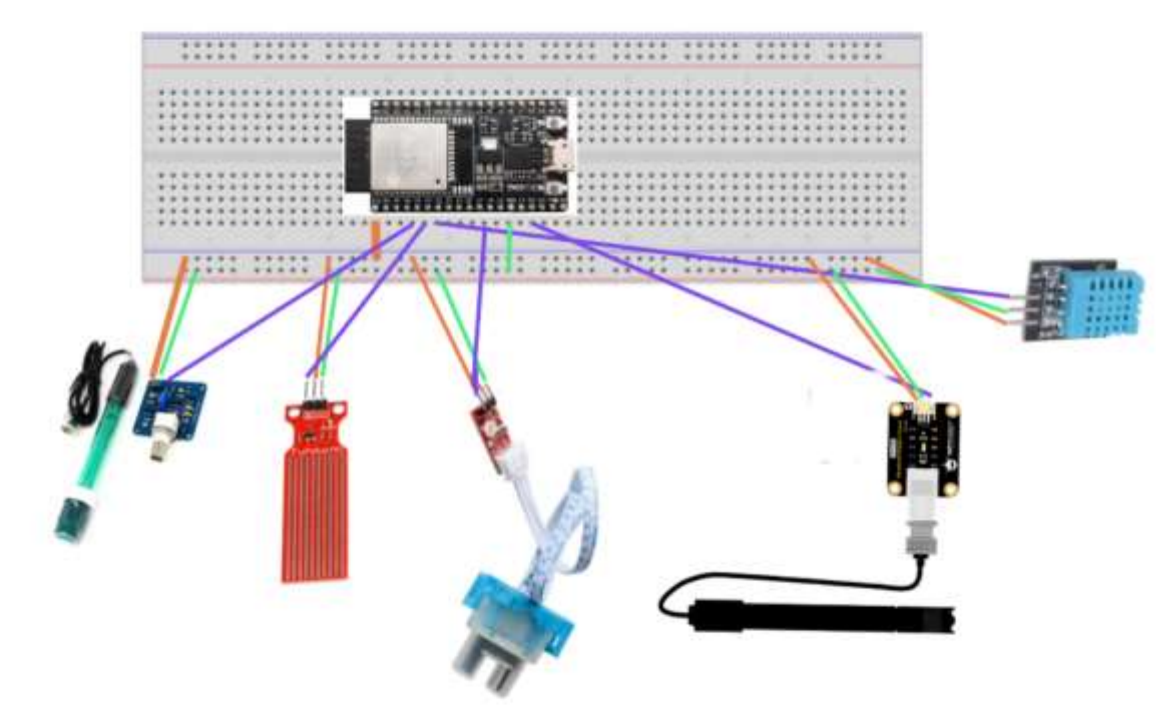
Resistance is inversely proportional to the height of the water

VCC

GND

S10 or data

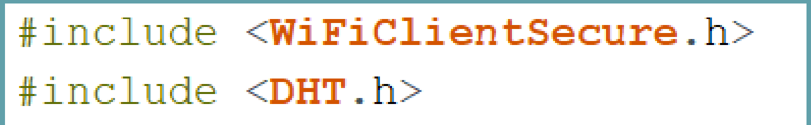
Circuit Diagram



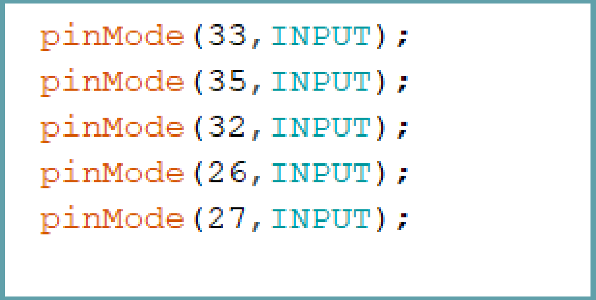
Implementation

- Setting up ESP32
- Connecting to network
- Setting SSL Certificate to client
- Reading Sensor data.
- API Request to server
- Storing Data into Database
- Alert User

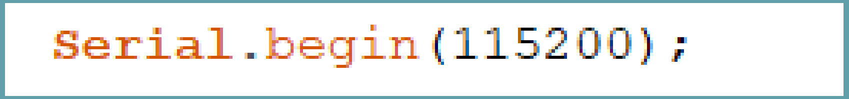
Setting up ESP32



```
#include <WiFiClientSecure.h>
#include <DHT.h>
```



```
pinMode(33,INPUT);
pinMode(35,INPUT);
pinMode(32,INPUT);
pinMode(26,INPUT);
pinMode(27,INPUT);
```



```
Serial.begin(115200);
```

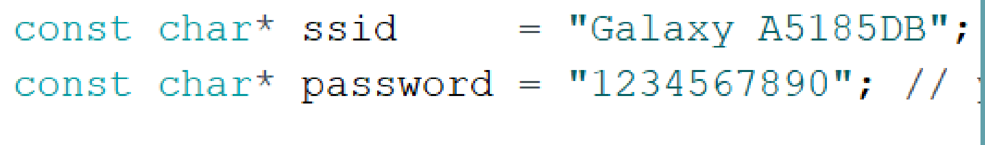
Importing libraries

Setting pin modes

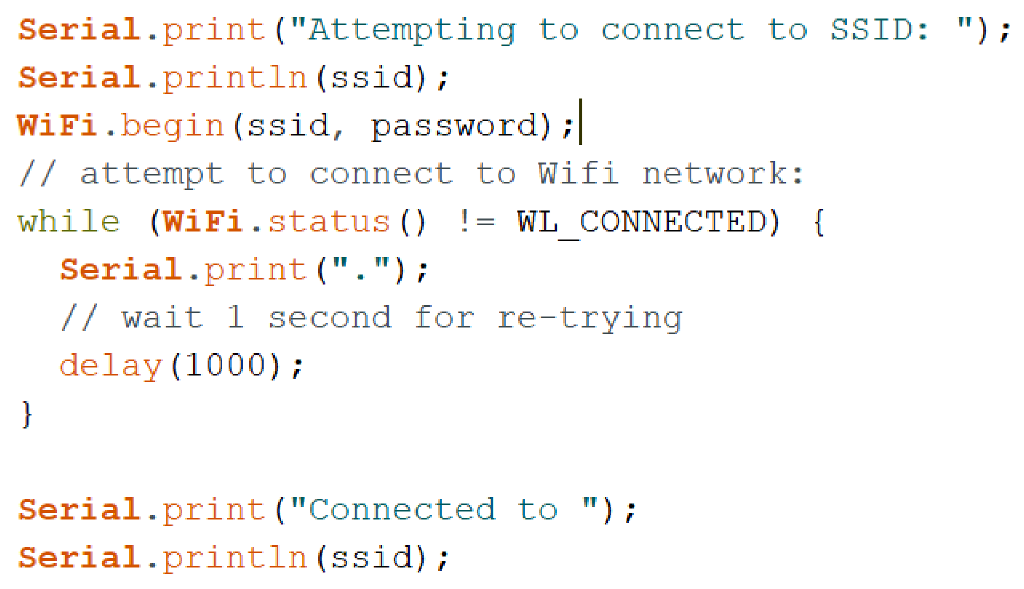
Initializing variables

Setting data transmission speed

Connecting network



```
const char* ssid     = "Galaxy A5185DB";
const char* password = "1234567890"; //
```

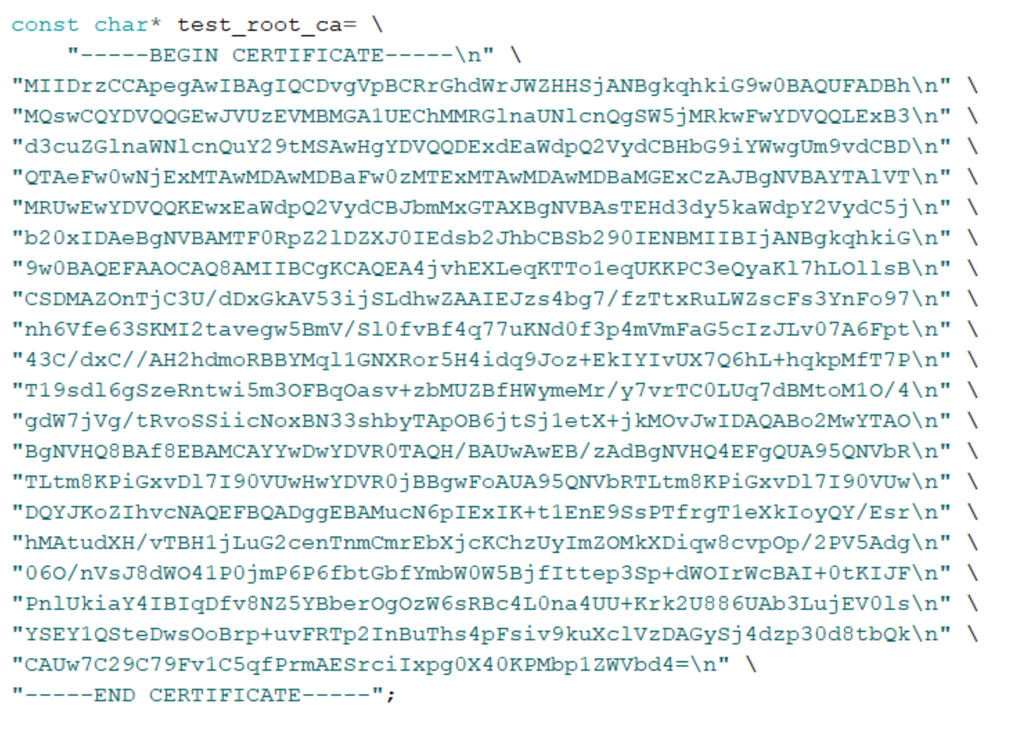


```
Serial.print("Attempting to connect to SSID: ");
Serial.println(ssid);
WiFi.begin(ssid, password);
// attempt to connect to Wifi network:
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  // wait 1 second for re-trying
  delay(1000);
}

Serial.print("Connected to ");
Serial.println(ssid);
```

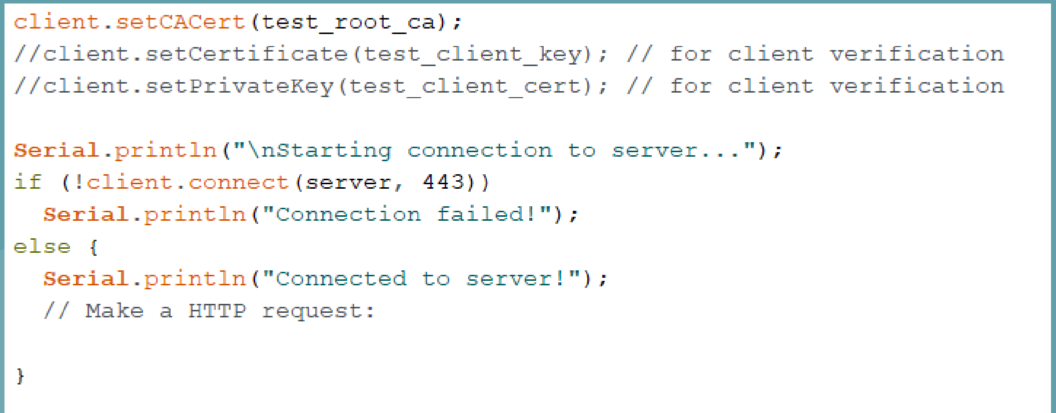
Initializing network parameters

request to connect

Setting SSL to client



```
const char* test_root_ca= \
    "-----BEGIN CERTIFICATE-----\n" \
"MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh\n" \
"MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3\n" \
"d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD\n" \
"QTAeFw0wNjExMTAwMDAwMDBaFw0zMTExMTAwMDAwMDBaMGExCzAJBgNVBAYTAlVT\n" \
"MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j\n" \
"b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IENBMIIBIjANBgkqhkiG\n" \
"9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUKKPC3eQyaKl7hLOllsB\n" \
"CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97\n" \
"nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt\n" \
"43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P\n" \
"T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4\n" \
"gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO\n" \
"BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR\n" \
"TLtm8KPiGxvDl7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDl7I90VUw\n" \
"DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr\n" \
"hMAtudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg\n" \
"06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5BjfIttep3Sp+dWOIrWcBAI+0tKIJF\n" \
"PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls\n" \
"YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4dzp30d8tbQk\n" \
"CAUw7C29C79Fv1C5qfPrmAESrciIxpg0X40KPMbp1ZWVbd4=\n" \
"-----END CERTIFICATE-----";
```



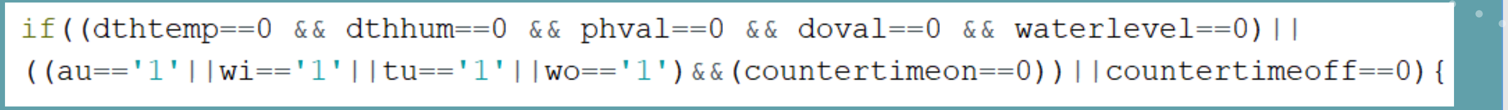
```
client.setCACert(test_root_ca);
//client.setCertificate(test_client_key); // for client verification
//client.setPrivateKey(test_client_cert); // for client verification

Serial.println("\nStarting connection to server...");
if (!client.connect(server, 443))
  Serial.println("Connection failed!");
else {
  Serial.println("Connected to server!");
  // Make a HTTP request:

}
```

SSL Certificate key

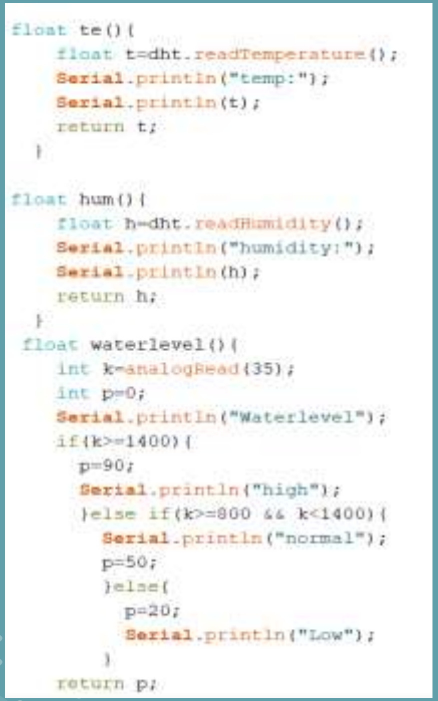
Assign to client

Request to connect

Reading Data



```
if((dthtemp==0 && dthhum==0 && phval==0 && doval==0 && waterlevel==0)||
((au=='1'||wi=='1'||tu=='1'||wo=='1')&&(countertimeon==0))||countertimeoff==0){
```



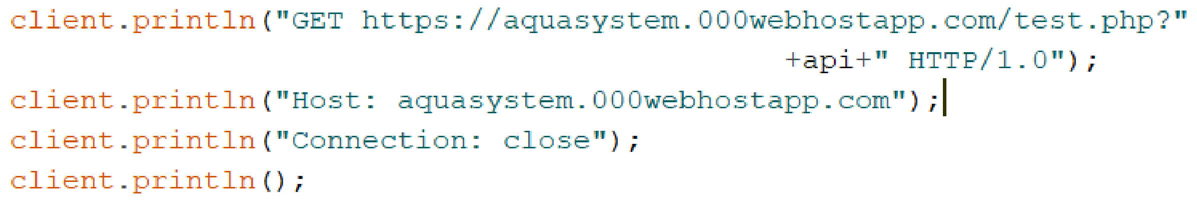
```
float te(){
    float t=dht.readTemperature();
    Serial.println("temp:");
    Serial.println(t);
    return t;
  }

float hum(){
    float h=dht.readHumidity();
    Serial.println("humidity:");
    Serial.println(h);
    return h;
  }
 float waterlevel(){
    int k=analogRead(35);
    int p=0;
    Serial.println("Waterlevel");
    if(k>=1400){
      p=90;
      Serial.println("high");
      }else if(k>=800 && k<1400){
        Serial.println("normal");
        p=50;
        }else{
          p=20;
          Serial.println("Low");
        }
    return p;
```

Check for condition

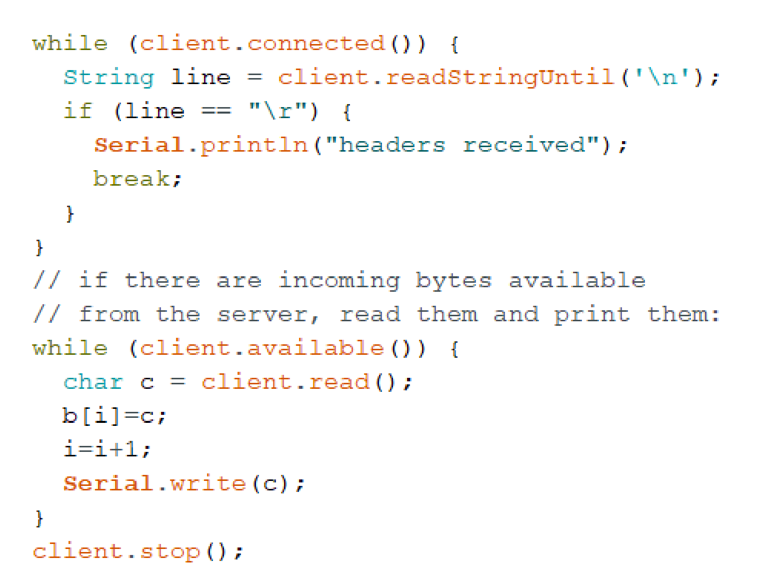
Read data

Map into its values.

API Request to server



```
client.println("GET https://aquasystem.000webhostapp.com/test.php?"
                                          +api+" HTTP/1.0");
client.println("Host: aquasystem.000webhostapp.com");
client.println("Connection: close");
client.println();
```



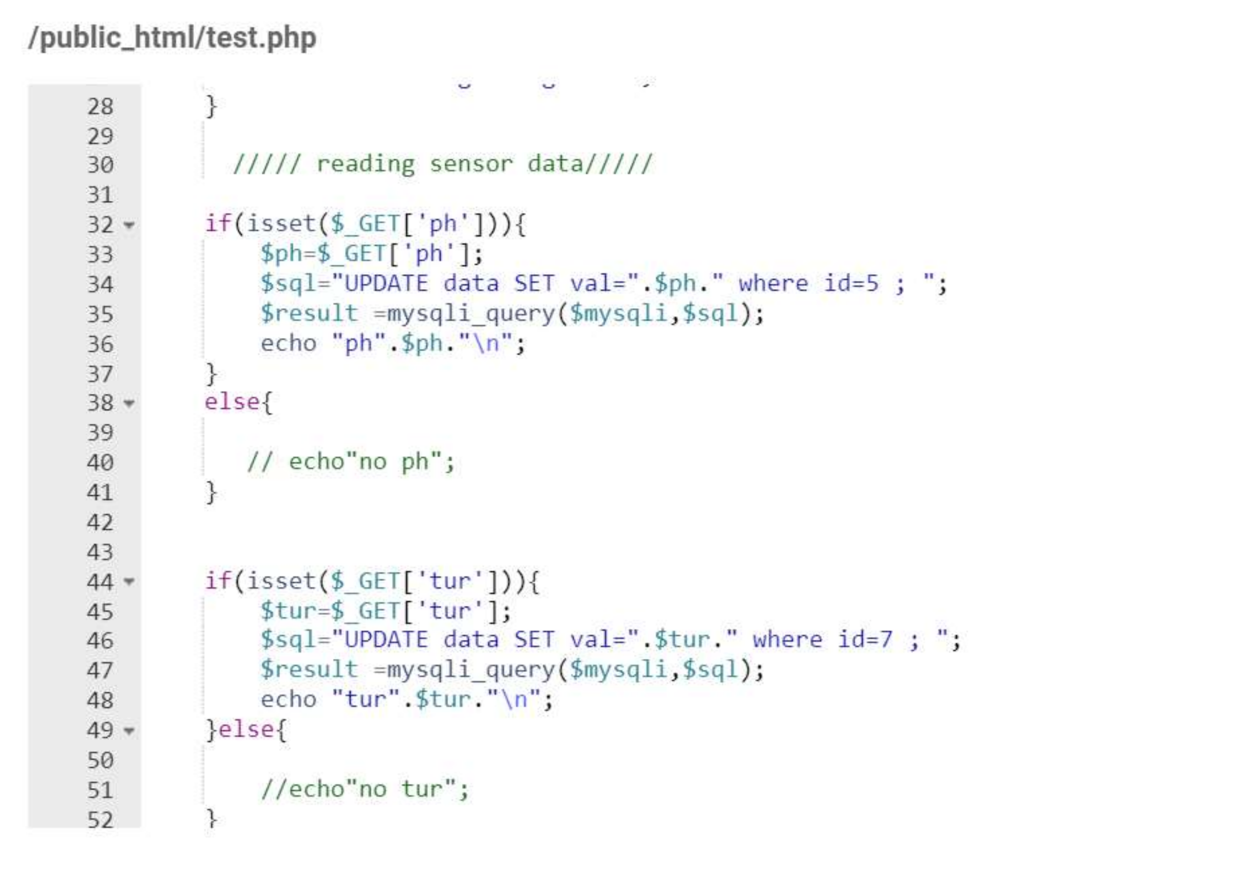
```
while (client.connected()) {
  String line = client.readStringUntil('\n');
  if (line == "\r") {
    Serial.println("headers received");
    break;
  }
}
// if there are incoming bytes available
// from the server, read them and print them:
while (client.available()) {
  char c = client.read();
  b[i]=c;
  i=i+1;
  Serial.write(c);
}
client.stop();
```

Create API URL

Initialize API request

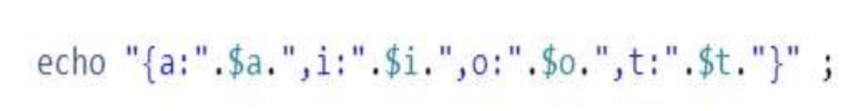
Receive response

Storing Data into Database



/public_html/test.php

```
}

  ///// reading sensor data/////

if(isset($_GET['ph'])){
    $ph=$_GET['ph'];
    $sql="UPDATE data SET val=".$ph." where id=5 ; ";
    $result =mysqli_query($mysqli,$sql);
    echo "ph".$ph."\n";
}
else{

   // echo"no ph";
}


if(isset($_GET['tur'])){
    $tur=$_GET['tur'];
    $sql="UPDATE data SET val=".$tur." where id=7 ; ";
    $result =mysqli_query($mysqli,$sql);
    echo "tur".$tur."\n";
}else{

    //echo"no tur";
}
```



```
echo "{a:".$a.",i:".$i.",o:".$o.",t:".$t."}" ;
```

Receive Data from URL

Initialize database connection

Fire query to save data

Close database connection

Send response to esp32

Alert user

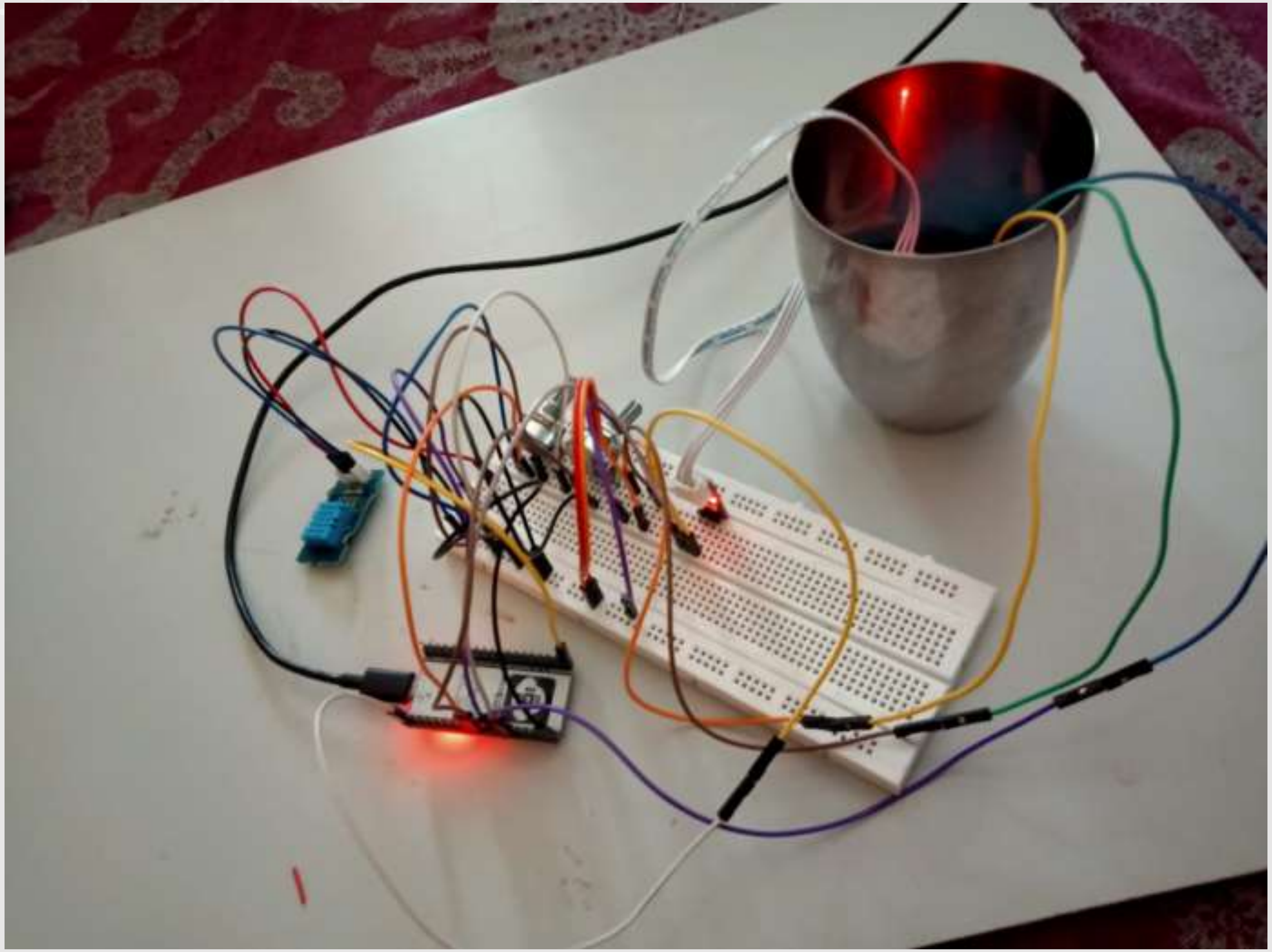
Read the values

Check with Limit value

Alert user when abnormal

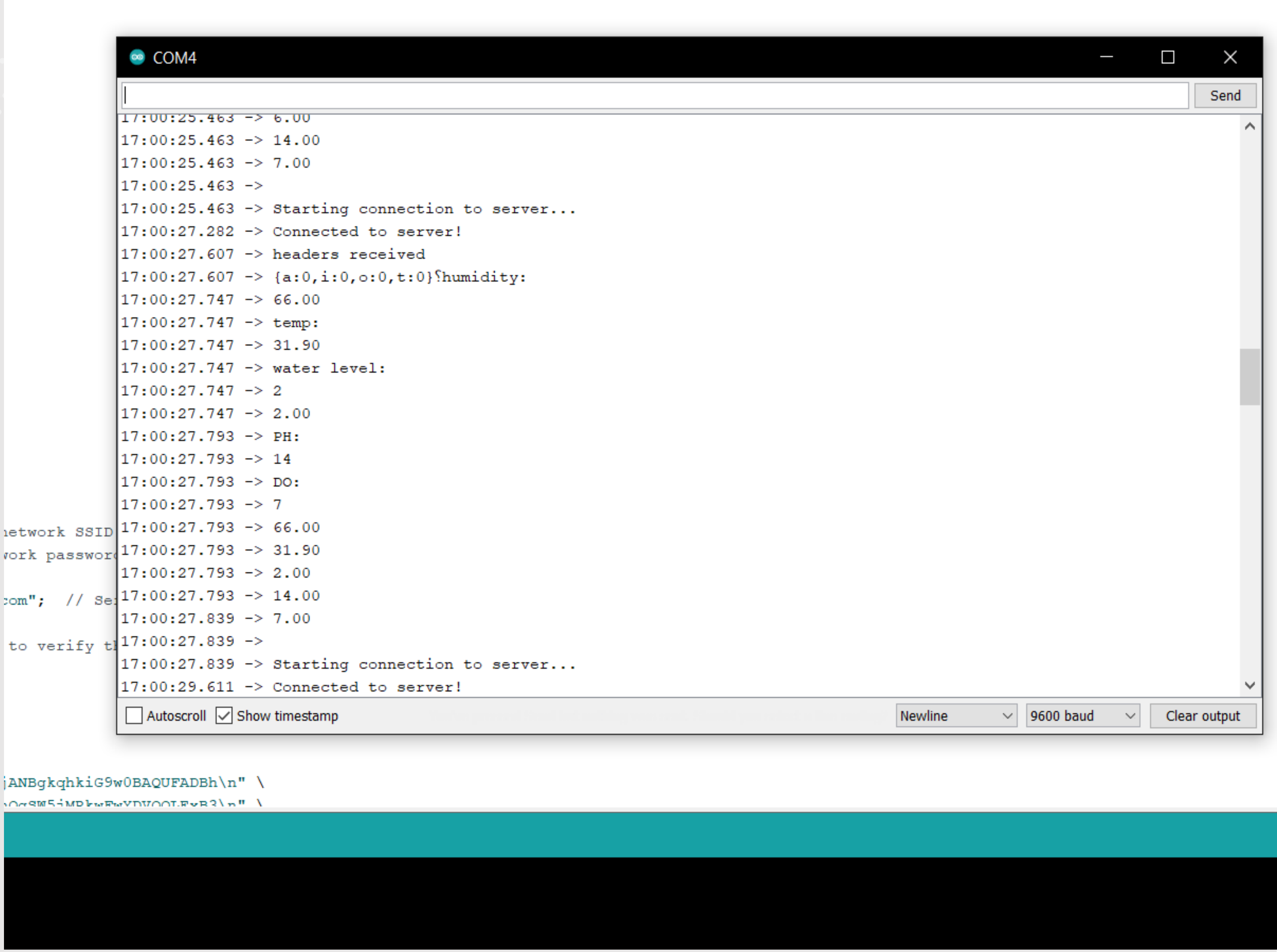
Proof of work

(Connection)



Proof of work

Uploading data from esp32 serial monitor)

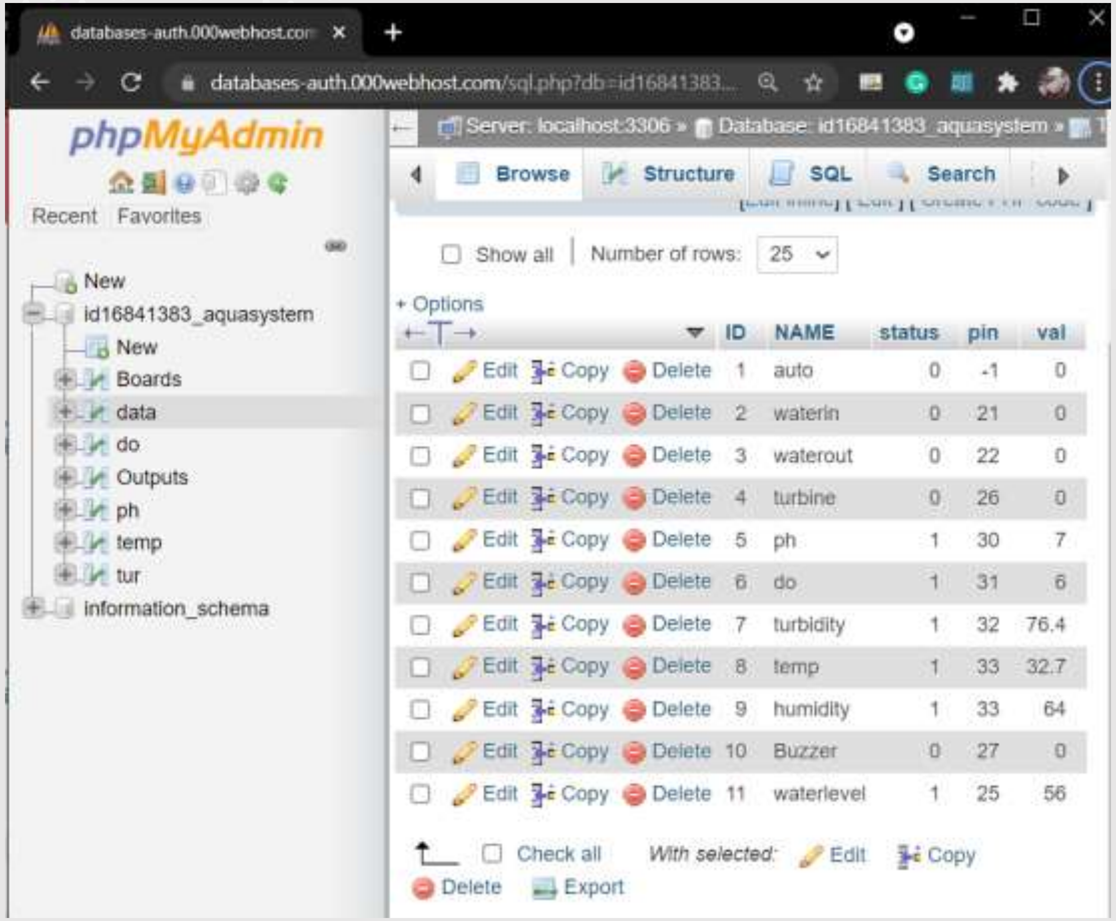


```
17:00:25.463 -> 6.00
17:00:25.463 -> 14.00
17:00:25.463 -> 7.00
17:00:25.463 ->
17:00:25.463 -> Starting connection to server...
17:00:27.282 -> Connected to server!
17:00:27.607 -> headers received
17:00:27.607 -> {a:0,i:0,o:0,t:0}⸮humidity:
17:00:27.747 -> 66.00
17:00:27.747 -> temp:
17:00:27.747 -> 31.90
17:00:27.747 -> water level:
17:00:27.747 -> 2
17:00:27.747 -> 2.00
17:00:27.793 -> PH:
17:00:27.793 -> 14
17:00:27.793 -> DO:
17:00:27.793 -> 7
17:00:27.793 -> 66.00
17:00:27.793 -> 31.90
17:00:27.793 -> 2.00
17:00:27.793 -> 14.00
17:00:27.839 -> 7.00
17:00:27.839 ->
17:00:27.839 -> Starting connection to server...
17:00:29.611 -> Connected to server!
```

```
network SSID
work passwor
com";  // Se
 to verify t

jANBgkqhkiG9w0BAQUFADBh\n" \
```

Proof of work

(Database)

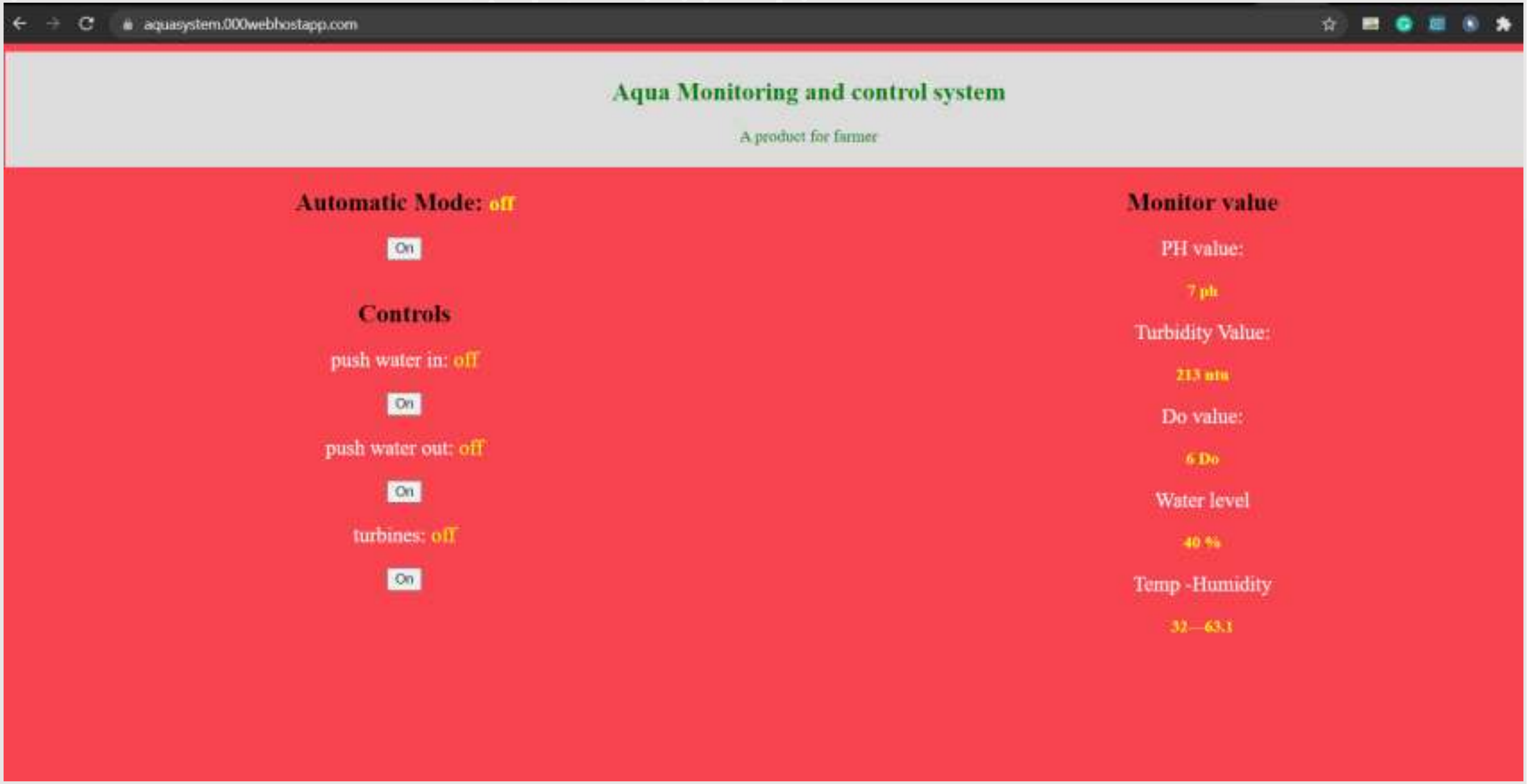


☐ Show all | Number of rows: 25

	ID	NAME	status	pin	val
☐ Edit Copy Delete	1	auto	0	-1	0
☐ Edit Copy Delete	2	waterin	0	21	0
☐ Edit Copy Delete	3	waterout	0	22	0
☐ Edit Copy Delete	4	turbine	0	26	0
☐ Edit Copy Delete	5	ph	1	30	7
☐ Edit Copy Delete	6	do	1	31	6
☐ Edit Copy Delete	7	turbidity	1	32	76.4
☐ Edit Copy Delete	8	temp	1	33	32.7
☐ Edit Copy Delete	9	humidity	1	33	64
☐ Edit Copy Delete	10	Buzzer	0	27	0
☐ Edit Copy Delete	11	waterlevel	1	25	56

Proof of work

(Website)



Aqua Monitoring and control system

A product for farmer

Automatic Mode: off

Controls

push water in: off

push water out: off

turbines: off

Monitor value

PH value:

7 ph

Turbidity Value:

213 ntu

Do value:

6 Do

Water level

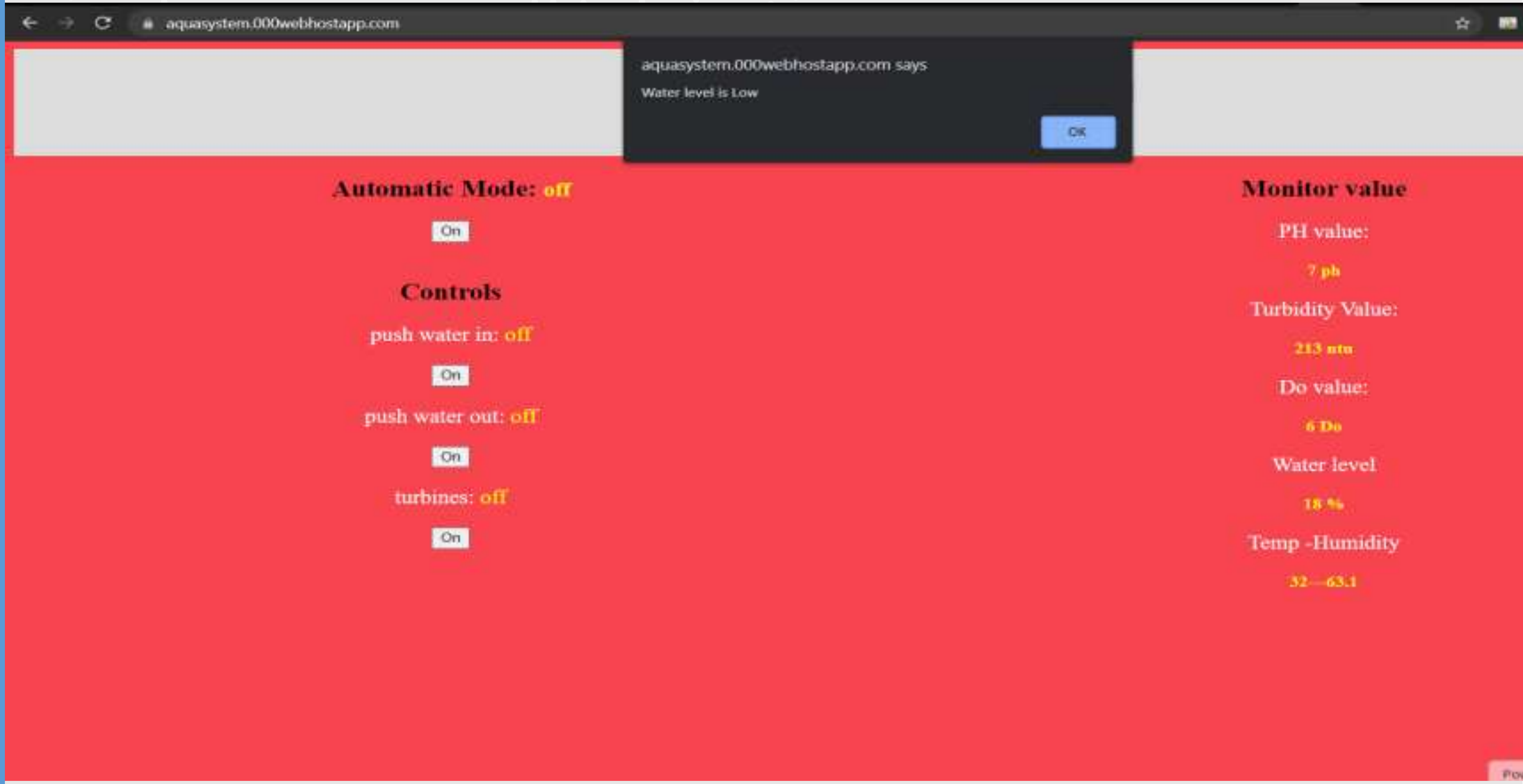
40 %

Temp -Humidity

32—63.1

Proof of work

(alert)



aquasystem.000webhostapp.com

aquasystem.000webhostapp.com says

Water level is Low

Automatic Mode: off

Controls

push water in: off

push water out: off

turbines: off

Monitor value

PH value:

7 ph

Turbidity Value:

213 ntu

Do value:

6 Do

Water level

18 %

Temp -Humidity

32---63.1

THANK YOU