10.Appendix

IEEE Code of Ethics

We, the members of the IEEE, in recognition of the importance of our technologies in affecting the quality of life throughout the world, and in accepting a personal obligation to our profession, its members and the communities we serve, do hereby commit ourselves to the highest ethical and professional conduct and agree:

I. To uphold the highest standards of integrity, responsible behaviour, and ethical conduct in professional activities.

1. to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment;

2. to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

3. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;

4. to avoid unlawful conduct in professional activities, and to reject bribery in all its forms;

5. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others;

6. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

II. To treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others.

7. to treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;

8. to not engage in harassment of any kind, including sexual harassment or bullying behaviour;

9. to avoid injuring others, their property, reputation, or employment by false or malicious actions, rumors or any other verbal or physical abuses;

III. To strive to ensure this code is upheld by colleagues and co-workers.

10. to support colleagues and co-workers in following this code of ethics, to strive to ensure the code is upheld, and to not retaliate against individuals reporting a violation.

Code for Control Unit:

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
WiFiClient wifiClient;

#define ON_Board_LED 2
const int Valve_pin = 13;

const char* ssid = "G";            // --> wifi name
const char* password = "12345678";  //--> wifi password

const char *host = "https://domestic-irrigation.000webhostapp.com:80/";

String inString = "";
void setup()
{
  Serial.begin(115200);
  delay(500);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");
  pinMode(ON_Board_LED, OUTPUT);
  digitalWrite(ON_Board_LED, HIGH);
  pinMode(Valve_pin, OUTPUT);
  pinMode(D0, OUTPUT);
  digitalWrite(D0, LOW);
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    digitalWrite(ON_Board_LED, LOW);
    delay(250);
    digitalWrite(ON_Board_LED, HIGH);
    delay(250);
  }
  digitalWrite(ON_Board_LED, HIGH);
  Serial.println("");
  Serial.print("Successfully connected to : ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
}

void loop() {
  HTTPClient http;
  String GetAddress, LinkGet, getData, LEDStatResultSend;
  int wid = 0;
  GetAddress = "GetData.php";
  LinkGet = host + GetAddress;
  getData = "ID=" + String(wid);
  Serial.println("----------------Connect to Server-----------------");
  Serial.println("Get LED Status from Server or Database");
  Serial.print("Request Link : ");
  Serial.println(LinkGet);
  http.begin(wifiClient, LinkGet); //--> Specify request destination
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
  int httpCodeGet = http.POST(getData);
  String payloadGet = http.getString();
    Serial.print("Response Code : ");
  Serial.println(httpCodeGet);
  Serial.print("Returned data from Server : ");
  Serial.println(payloadGet);
  String Status = payloadGet.substring(0, payloadGet.indexOf(","));
  Serial.println(Status);
  if(Status.indexOf('0')!=-1)
  {
    digitalWrite(Valve_pin, HIGH);
    Serial.println("Valve OFF");
  }
  else if(Status.indexOf('1')!=-1)
  {
    digitalWrite(Valve_pin, LOW);
    Serial.println("Valve ON");
  }
  Serial.println("----------------Closing Connection----------------");
  http.end();
  Serial.println();
  Serial.println("Please wait 5 seconds for the next connection.");
  Serial.println();
  delay(5000);
}
```

Code for Sensor Unit:

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
```

```
#include <WiFiClient.h>


WiFiClient wifiClient;


#define ON_Board_LED 2


const char* ssid = "G";              // --> wifi name
const char* password = "12345678";  //--> wifi password
int status;
const char *host = "https://domestic-irrigation.000webhostapp.com:80/";
String inString = "";
void setup() {
  Serial.begin(115200);
  delay(500);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");


  pinMode(ON_Board_LED, OUTPUT);
  digitalWrite(ON_Board_LED, HIGH);


  pinMode(D0, OUTPUT);
  digitalWrite(D0, LOW);
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    digitalWrite(ON_Board_LED, LOW);
    delay(250);
    digitalWrite(ON_Board_LED, HIGH);
    delay(250);
  }
  digitalWrite(ON_Board_LED, HIGH);
  Serial.println("");
  Serial.print("Successfully connected to : ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  pinMode(A0, INPUT);
}

void loop() {
  HTTPClient http;
  String GetAddress, LinkGet, getData, LEDStatResultSend;
  int moisture = int(100.00-((analogRead(A0)/1023.00) * 100.00)) ;
  if(moisture<40)
  {
    status = 1;
  }
  else
  {
    status = 0;
  }
  GetAddress = "SendData.php";
  LinkGet = host + GetAddress;
  getData = "status="+String(status)+"&&moisture="+String(moisture);
  http.begin(wifiClient, LinkGet); //--> Specify request destination
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
  int httpCodeGet = http.POST(getData);
  String payloadGet = http.getString();
  Serial.print(status);
  Serial.print(",");
  Serial.println(moisture);
  Serial.println(payloadGet);

  Serial.println("----------------Closing Connection----------------");
  http.end();
  Serial.println();
  Serial.println("Please wait 5 seconds for the next connection.");
  Serial.println();
  delay(5000);
}
```