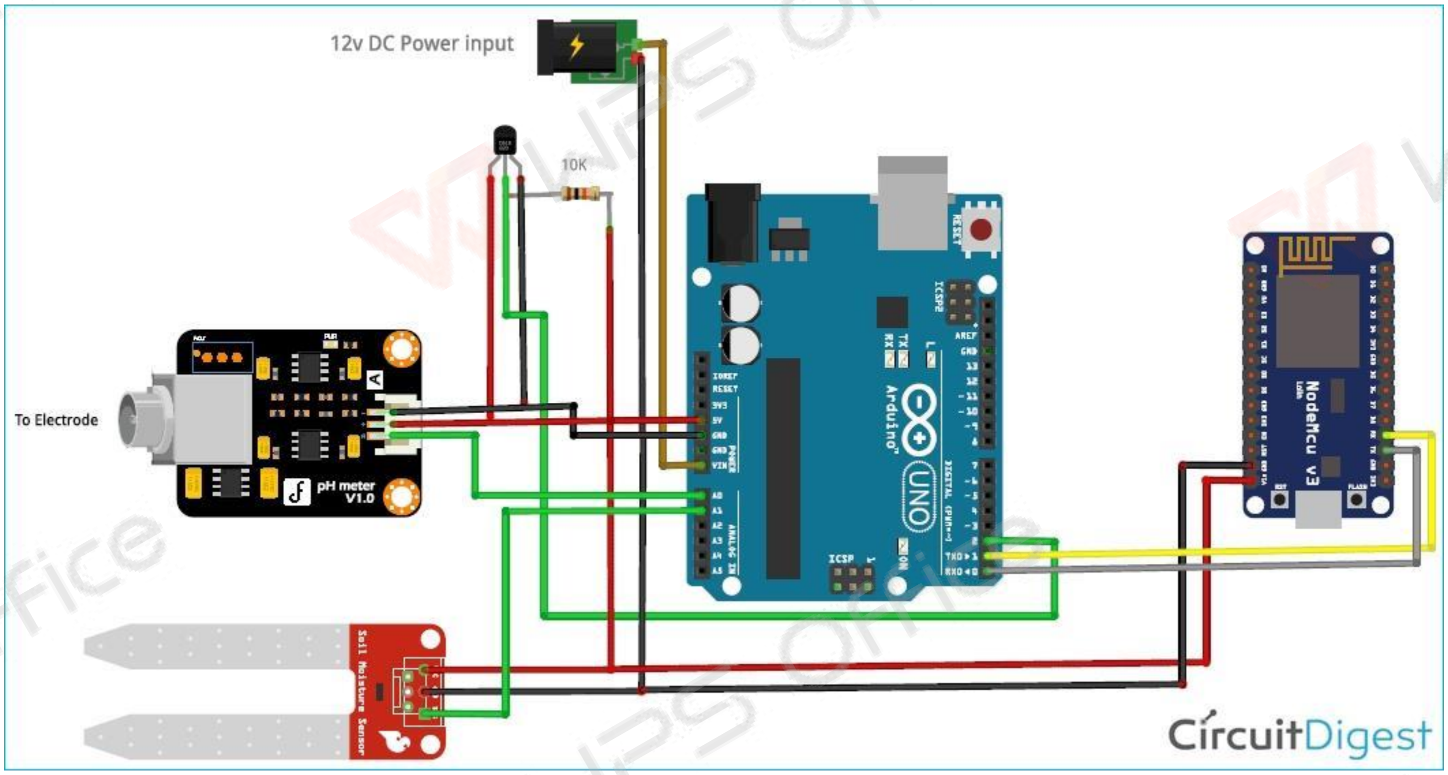
Developing of smart water Fountains

IoT based smart water using Adriano:

Agriculture is the backbone of our country and it is very important to know the parameter of soil and water for efficient harvesting. The various parameters that can be monitored are Soil moisture, pH of water, Temperature, etc. We previously measured these parameters in different tutorials but today we will not only combining them but also display them on a webpage so that they can be monitored from anywhere in the world.



Temperature sensors like LM35 or DHT11 can also be used to measure the temperature but DS18B20 is available in a waterproof casing so it is a perfect choice to monitor the temperature of the water

Circuit diagram explanation:

The devices are used in the Circuit Arduino UNO,NodeMCU,Gravity Analog pH sensor,DS18B20 temperature sensor,Soil moisture sensor,Power supply,Jumpers,Breadboard the technical sufficient used in this circuit isTemperature range: -55 to 125°C,Bit selectable resolution: 9-12 bit,1-Wire interface,Unique 64-bit address enables multiplexing,Accuracy: ±0.5°C,Operating Voltage: 3-5 VDC,Conversion time: 750ms at 12-bit.

Pinout of DS18B20:

VCC: Power input: (3.3 – 5) V DC

Ground: Ground pin of the circuit

Data: 1 Wire temperature value data output pin

Signal module are Supply Voltage: 3.3~5.5V,BNC Probe Connector,High Accuracy: ±0.1@25°C,Detection Range: 0~14,Operating Temperature Range: 5~60°C,Zero(Neutral) Point: 7±0.5,Easy calibration,Internal Resistance: <250MΩ.

Code :

# Arduino Code

#include <OneWire.h>

#include <DallasTemperature.h>

#include <ArduinoJson.h>

OneWire oneWire(2);

DallasTemperature temp_sensor(&oneWire);

Float calibration_value = 21.34;

```
Int phval = 0;

Unsigned long int avgval;

Int buffer_arr[10], temp;

LVoid setup()

{

  Serial.begin(9600);

  Temp_sensor.begin();

}

StaticJsonBuffer<1000> jsonBuffer;

JsonObject& root = jsonBuffer.createObject();

Void loop() {

  For (int I = 0; I < 10; i++)

  {

   Buffer_arr[i] = analogRead(A0);

Delay(30);

  }

  For (int I = 0; I < 9; i++)

{

    For (int j = I + 1; j < 10; j++)

 {
    If (buffer_arr[i] > buffer_arr[j])
     {
        Temp = buffer_arr[i];
      Buffer_arr[i] = buffer_arr[j];
     Buffer_arr[j] = temp;
      }
    }
}
  Avgval = 0;
  For (int I = 2; I < 8; i++)
    Avgval += buffer_arr[i];
  Float volt = (float)avgval * 5.0 / 1024 / 6;
  Float ph_act = -5.70 * volt + calibration_value;
  Temp_sensor.requestTemperatures();
  Int moisture_analog=analogRead(A1);
  Int moist_act=map(moisture_analog,0,1023,100,0);
  Root["a1"] = ph_act;
  Root["a2"] = temp_sensor.getTempCByIndex(0);
  Root["a3"] = moist_act;
  Root.printTo(Serial);
  Serial.println(“”);

}

NodeMCU Code:

#include<ESP8266WiFi.h>

#include<WiFiClient.h>

#include<ESP8266WebServer.h>

Const char* ssid = “admin”;//Replace with your network SSID

Const char* password = “12345678”;//Replace with your network password

ESP8266WebServer server(80);

String page = “”;

Int data1, data2, data3;

Void setup()

{

  Serial.begin(9600);

  WiFi.begin(ssid, password);

While (WiFi.status() != WL_CONNECTED)

  {

  Delay(500);

  Serial.print(“.”);

  }
  Serial.println(WiFi.localIP());

  Server.on(“/”, []()

  {

    Page = “<html><head><title>IoT Design</title></head><style type=\”text/css\”>”;

    Page += “table{border-collapse: collapse;}th {background-color:  green ;color: white;}table,td
{border: 4px solid black;font-size: x-large;”;

    Page += “text-align:center;border-style: groove;border-color:
rgb(255,0,0);}</style><body><center>”;

    Page += “<h1>Smart Aquaculture Monitoring using IoT</h1><br><br><table style=\”width:
1200px;height: 450px;\”><tr>”;

    Page += “<th>Parameters</th><th>Value</th><th>Units</th></tr><tr><td>PH
Value</td><td>”+String(data1)+”</td><td>N/A</td></tr>”;

    Page +=
“<tr><td>Temperature</td><td>”+String(data2)+”</td><td>Centigrade</td></tr><tr><td>Moisture</
td><td>”+String(data3)+”</td><td>%</td>”;

    Page += “<meta http-equiv=\”refresh\” content=\”3\”>”;

    Server.send(200, “text/html”, page);

  });

  Server.begin();

}

Void loop()

{

  StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& root = jsonBuffer.parseObject(Serial);

  If (root == JsonObject::invalid())

  {

    Return;

    Serial.println(“invalid”);

  }

  Data1 = root[“a1”];

  Data2 = root[“a2”];

  Data3 = root[“a3”];

  Serial.println(data1);

  Serial.println(data2);

  Serial.println(data3);

  Server.handleClient();

}
```

Code defenition:

```
#include <OneWire.h>

#include <DallasTemperature.h>

#include <ArduinoJson.h>
```

Next, define the connection pin of Arduino, where the output pin of the DS18B20 sensor will be connected, which is digital pin 2 in my case. Then, objects for onewire class and

DallasTemperature class are defined which will be required in the coding for temperature measurement.

```
OneWire oneWire(2);

DallasTemperature temp_sensor(&oneWire);
```

Next, the calibration value is defined, which can be modified as required to get an accurate pH value of solutions.

```
Float calibration_value = 21.34;
```

Then a JSON Object is defined which will be required for sending parameters from the Transmitter part to the Receiver part.

```
StaticJsonBuffer<1000> jsonBuffer;

JsonObject& root = jsonBuffer.createObject();
```

Inside loop(), read 10 sample Analog values and store them in an array. This is required to smooth the output value.

```
For(int i=0;i<10;i++)

 {

 Buffer_arr[i]=analogRead(A0);

 Delay(30);

 }
```

Then, we have to sort the Analog values received in ascending order. This is required because we need to calculate the running average of samples in the later stage.

```
For(int i=0;i<9;i++)

 {

 For(int j=i+1;j<10;j++)
 {
 If(buffer_arr[i]>buffer_arr[j])
 {
 Temp=buffer_arr[i];
 Buffer_arr[i]=buffer_arr[j];
 Buffer_arr[j]=temp;
 }
 }
 }
```

Finally, calculate the average of a 6 centre sample Analog values. Then this average value is converted into actual pH value and stored in a variable.

```
For(int i=2;i<8;i++)
 Avgval+=buffer_arr[i];
 Float volt=(float)avgval*5.0/1024/6;
 Float ph_act = -5.70 * volt + calibration_value;
```

To send a command to get the temperature values from the sensor, requestTemperatures() function is used.

```
Temp_sensor.requestTemperatures();
```

Now, analog values from the soil moisture sensor are read and this is mapped to percentage using map() function as shown below:

```
Int moisture_analog=analogRead(A1);
```

Int moist_act=map(moisture_analog,0,1023,100,0);

Finally, the parameters which are to be sent to NodeMCU, are inserted into JSON objects and they are sent via serial communication using root.printTo(Serial) command.

```
Root["a1"] = ph_act;

Root["a2"] = temp_sensor.getTempCByIndex(0);

Root["a3"] = moist_act;

Root.printTo(Serial);

Serial.println("");
```

Conclusion:

The system proposed in this paper is an efficient, inexpensive IoT solution for real-time water quality monitoring. The developed system having Arduino Mega and NodeMCU target boards are interfaced with several sensors successfully. An efficient algorithm is developed in real-time, to track water quality.

Thank you!