# Table of Contents

## Scenario:

Host a website that establishes connection between an (Arduino Mega + ESP01(8266)).

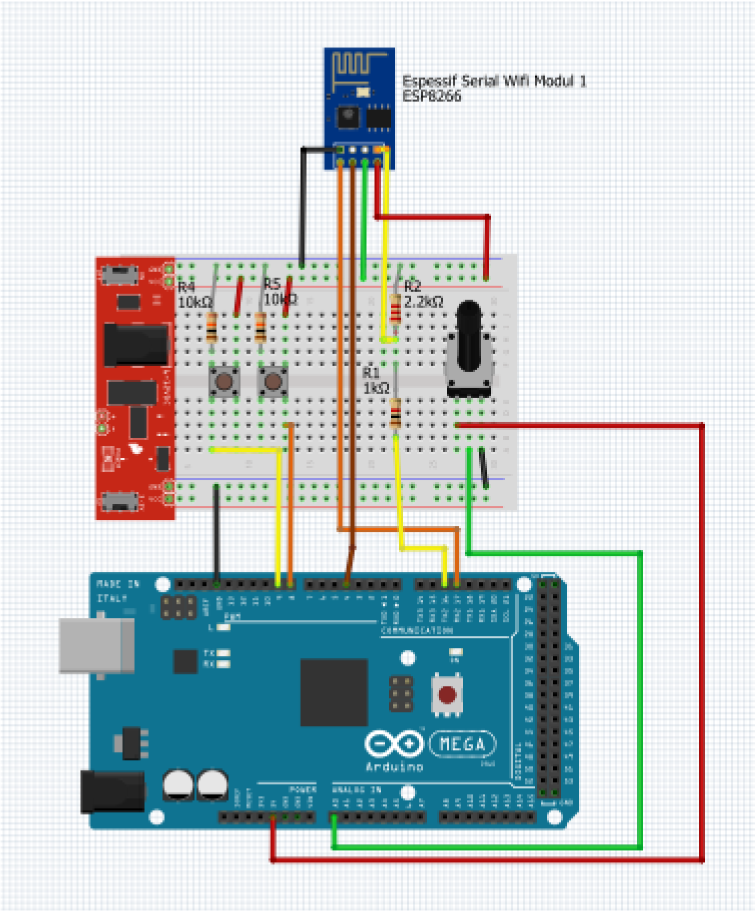
## Goal:

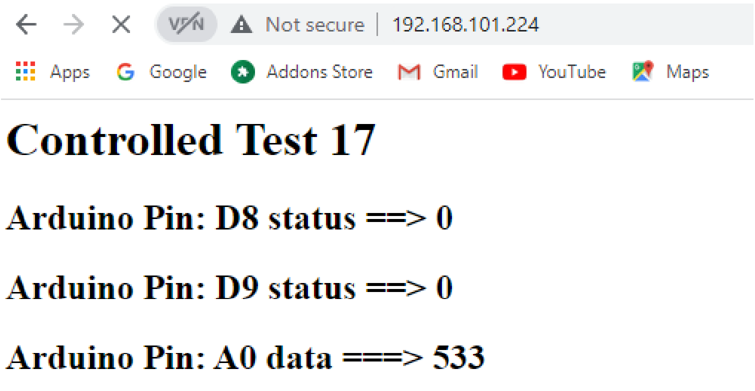
Fully automate a temperature sensitive environment with 1 * (6kW heating element), 3 * (216W Liquid cooling Radiators), 3 * (MAX6675 Thermocouples), 1 * (DS3231 Timer).

## Progress:

The following is done Using the Arduino IDE (C++) along with Hosting HTML using the ESP01(8266).

Using the Static IP of the ESP connected to a network it hosts a Website, Sending data to the Website.





# Controlled Test 17

### Arduino Pin: D8 status ==> 0

### Arduino Pin: D9 status ==> 0

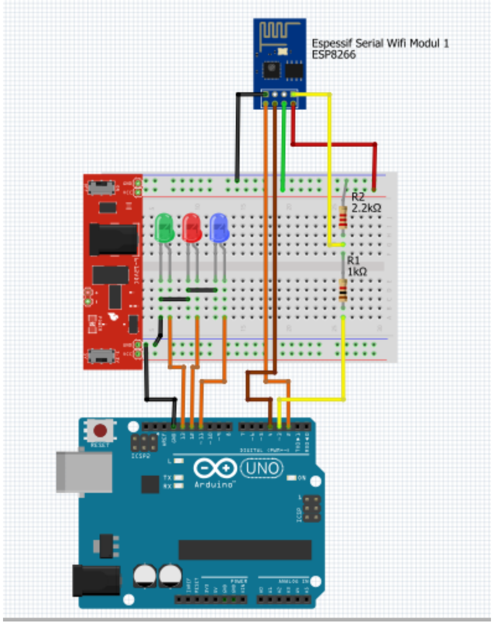
### Arduino Pin: A0 data ===> 533

```
#define
esp8266
Serial2
           #define CH_PD 4
           #define speed8266 115200 // This is the speed that works with ESP8266

           #define DEBUG true

           void setup()
           {
             esp8266.begin (speed8266);
             Serial.begin(9600);
             reset8266(); // Pin EN needs a reset before start communication
             InitWifiModule() // Inciate module as WebServer
           }

           void loop()
           {
             if (esp8266.available()) // enquire if 8266 is sending data
             {
               if (esp8266.find("+IPD,"))
               {
                 delay(300);
                 int connectionId = esp8266.read() - 48;

                 String webpage = "<head><meta http-equiv=""refresh"" content=""3"">";
                 webpage += "</head><h1>==Controlled Test 17==</h1><h2>Arduino Pin: ";
                 webpage += "D8 status ==> ";
                 int a = digitalRead(8);
                 webpage += a;
                 webpage += "<h2>Arduino Pin: D9 status ==> ";
                 int b = digitalRead(9);
                 webpage += b;
                 webpage += "<h2>Arduino Pin: A0 data ===> ";
                 int c = analogRead(0);
                 webpage += c;
                 webpage += "</h2>";

                 String cipSend = "AT+CIPSEND=";
                 cipSend += connectionId;
                 cipSend += ",";
                 cipSend += webpage.length();
                 cipSend += "\r\n";
```

```
      sendData(cipSend, 1000, DEBUG);
      sendData(webpage, 1000, DEBUG);

      String closeCommand = "AT+CIPCLOSE=";
      closeCommand += connectionId; // append connection id
      closeCommand += "\r\n";

      sendData(closeCommand, 3000, DEBUG);
    }
  }
}

/*************************************************/
void InitWifiModule()
{
  sendData("AT+RST\r\n", 2000, DEBUG); // reset
  sendData("AT+CWJAP=\"LANtern  [2Ghz]\",\"A52E9A7197\"\r\n", 2000, DEBUG); //Connect network
  delay(3000);
  sendData("AT+CWMODE=1\r\n", 1000, DEBUG);
  sendData("AT+CIFSR\r\n", 1000, DEBUG); // Show IP Address
  sendData("AT+CIPMUX=1\r\n", 1000, DEBUG); // Multiple connections
  sendData("AT+CIPSERVER=1,80\r\n", 1000, DEBUG); // start comm port 80
}

/*************************************************/
 // Send AT commands to ESP module
String sendData(String command, const int timeout, boolean debug)
{
  String response = "";
  esp8266.print(command);
  long int time = millis();
  while ( (time + timeout) > millis())
  {
    while (esp8266.available())
    {
      // The ESP has data so display its output to the serial window
      char c = esp8266.read(); // read the next character.
      response += c;
    }
  }
  if (debug)
  {
    Serial.print(response);
  }
  return response;
}

/*************************************************/
// Reset function to accept communication
void reset8266 ()
{
  pinMode(CH_PD, OUTPUT);
  digitalWrite(CH_PD, LOW);
  delay(300);
  digitalWrite(CH_PD, HIGH);
}
```

## Current Challenge:

From the example followed from, Mjrovia https://github.com/Mjrovai/ESP8266 Part 3, he replaces all data sending components with 3 data receiving components, LEDs. He removes the whole of the HTML code, Places it into a Text editor (Visual Studio Code) and uses the Desktop to Emulate the Website through a local connection. The challenge is that I see is no connection between the Arduino even when placing the ESP Static IP into the JQuery get command and letting the Desktop Emulate the Website



## Arduino IDE Code:

```
//#define esp8266 Serial2 <===== In case of using MEGA, take out the comment
#include <SoftwareSerial.h>   // In case of using MEGA, mark the line as comment
SoftwareSerial esp8266(2,3);  // In case of using MEGA, mark the line as comment
//Rx ==> Pin 2; TX ==> Pin3

#define speed8266 9600  // <========= Change the Speed for the one used at
Module
#define CH_PD 4
#define DEBUG true

int red =12;
int green =13;
int blue =11;


void setup()

{
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);

  digitalWrite(red,LOW);
  digitalWrite(green,LOW);
  digitalWrite(blue,HIGH);

  Serial.begin(9600);
  esp8266.begin(speed8266);

  reset8266();
  InitWifiModule();

  digitalWrite(blue,LOW);
  digitalWrite(green,HIGH);
  delay(2000);
  digitalWrite(green,LOW);
}

void loop()
{
  if(esp8266.available())
  {
    if(esp8266.find("+IPD,"))
    {
      delay(1000);
      int connectionId = esp8266.read()-48;
      esp8266.find("pin=");
      int pinNumber = (esp8266.read()-48)*10;
      pinNumber += (esp8266.read()-48);
      int statusLed =(esp8266.read()-48);

      digitalWrite(pinNumber, statusLed);

      String closeCommand = "AT+CIPCLOSE=";
      closeCommand+=connectionId;
      closeCommand+="\r\n";
      sendData(closeCommand,1000,DEBUG);

    }
  }
}

/*******************************************************************************
**********
* Name: sendData
* Description: Function used to send data to ESP8266.
* Params: command - the data/command to send; timeout - the time to wait for a
response; debug - print to Serial window?(true = yes, false = no)
* Returns: The response from the esp8266 (if there is a reponse)
*/
String sendData(String command, const int timeout, boolean debug)
{
    String response = "";

    esp8266.print(command); // send the read character to the esp8266
    long int time = millis();
    while( (time+timeout) > millis())
    {
      while(esp8266.available())
      {
        char c = esp8266.read(); // read the next character.
        response+=c;
      }
    }
    if(debug)
    {
      Serial.print(response);
    }
    return response;
}




/***********************************************************************/
// This initializes the Wifi Module as a server
void InitWifiModule()
{
  sendData("AT+RST\r\n", 2000, DEBUG); // reset
  sendData("AT+CWMODE=1\r\n", 1000, DEBUG);
  //sendData("AT+CWJAP=\"Your_WiFi_Network_Name\",\"password\"\r\n", 2000,
DEBUG); //Connect network
  delay (1000);
  sendData("AT+CIFSR\r\n", 1000, DEBUG); // Show IP Adress
  sendData("AT+CIPMUX=1\r\n", 1000, DEBUG); // Multiple conexions
  sendData("AT+CIPSERVER=1,80\r\n", 1000, DEBUG); // start comm port 80
}

/*************************************************/
// Reset funtion to accept communication
void reset8266 ()
{
  pinMode(CH_PD, OUTPUT);
  digitalWrite(CH_PD, LOW);
  delay(300);
  digitalWrite(CH_PD, HIGH);
}
```

## HTML Code:

```
<html
lang="en">
              <head>
                <meta charset="utf-8">
                <title>MJRoBot WebServer ESP8266</title>
                <style>
              <!-- edita o paragrafo -->
                p {
                  color: red;
                  margin: 5px;
                  cursor: pointer;
```

```
  }
  p:hover {
    background: yellow;
  }
  </style>
  <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
  <META HTTP-EQUIV="refresh" CONTENT="15">
</head>
<body>

         <center>
                   <!-- Headlines -->
                   <h1>MJRoBot WebServer ESP8266</h1>
                   <h2>Arduino GPIO Control</h2>

        <!-- buttons for GPIO 11 -->
         <h4>GPIO 11
                   <button style="color:blue"; id="111" class="led">ON</button>
                   <button style="color:red"; id="110" class="led">OFF</button>

                   <!-- buttons for GPIO 12 -->
                   <h4>GPIO 12
                   <button style="color:blue"; id="121" class="led">ON</button>
                   <button style="color:red"; id="120" class="led">OFF</button>

                   <!-- buttons for GPIO 13 -->
                   <h4>GPIO 13
                   <button style="color:blue"; id="131" class="led">ON</button>
                   <button style="color:red"; id="130" class="led">OFF</button>

         </center>

<script>
// when a button is clicked
$( ".led" ).click(function() {
        // read value of GPIO11, GPIO12, GPIO13 id's
         var p = $(this).attr('id');

         // send via GET to IP address with the parameter "pin" and value "p"
         $.get("http:/192.168.101.224:80/", {pin:p});
});
</script>

</body>
</html>
```

### *Objective:*

The website may be hosted through either the Arduino or the Desktop but take into consideration: Arduino local host gives only the user access to the tech or in like terms “Self-Service”, whereas if it could be hosted from another gaming desktop or Server Desktop, qualified personnel could assist.

All that is needed is to Send Data from the website to the Arduino, from the Arduino to the Website is not perfected but the concept is present and working.

Thank you for your time.