```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

const char *ssid = "**********";             //user setting
const char *password = "**********";    //user setting

const IPAddress ip(192, 168, 11, 3);            //IP アドレスの指定
const IPAddress gateway(192, 168, 11, 1);
const IPAddress subnet(255, 255, 255, 0);

WebServer server ( 80 );

int sensorPin = 32;       // select the input pin for the potentiometer
int sensorValue = 0;      // variable to store the value coming from the sensor

const int VALVE_OPEN = 19;
const int VALVE_CLOSE = 18;
const int RESET = 14;
const int LED = 17;
int waterLV = 0;
int DIPSW = 0;
int VL = 0;

unsigned long base_time = 0;
unsigned long previous_time = 0;
unsigned long second = 0;
unsigned long minute = 0;
unsigned long hour = 0;
unsigned long pulseCounter = 0;

int n = 0;
int pulse[70];

void onRising() {
   ++pulseCounter;
```

```
}

void setup ( void ) {
  //変数初期化
  pinMode(sensorPin, INPUT);
  pinMode(RESET, OUTPUT);
  pinMode(VALVE_OPEN, OUTPUT);
  pinMode(VALVE_CLOSE, OUTPUT);
  pinMode(LED, OUTPUT);
  digitalWrite(RESET, 0);
  digitalWrite(VALVE_OPEN, 0);
  digitalWrite(VALVE_CLOSE, 0);
  digitalWrite(LED, 0);
  WiFi.mode(WIFI_STA);

  attachInterrupt(digitalPinToInterrupt(sensorPin), onRising, RISING);
  Serial.begin(9600);

 if (!WiFi.config(ip,gateway,subnet)){
        Serial.println("Failed to configure!");
  }
 WiFi.begin(ssid, password);
 while (WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
 }
 Serial.println("");
 Serial.print("Connected to ");
 Serial.println(ssid);

 Serial.print("IP address: ");
 Serial.println(WiFi.localIP());
   if (MDNS.begin("esp32")) {
     Serial.println("MDNS responder started");
   }

  server.on ( "/", handleRoot );
  server.on ( "/test.svg", drawGraph );
  server.on ( "/inline", []() {
  server.send ( 200, "text/plain", "this works as well" );
```

```
    } );

  server.onNotFound ( handleNotFound );
  server.begin();
  Serial.println ( "HTTP server started" );
}

void checkDIP() {
  pinMode(25, INPUT);
  pinMode(26, INPUT);
  pinMode(27, INPUT);
  pinMode(33, INPUT);
     if(digitalRead(25) == 1 && digitalRead(26) == 0 && digitalRead(27) == 0 && digitalRead(33) ==
0){
        DIPSW = 100;
      }else if(digitalRead(25) == 1 && digitalRead(26) == 1 && digitalRead(27) == 0 &&
digitalRead(33) == 0){
        DIPSW = 150;
      }else if(digitalRead(25) == 0 && digitalRead(26) == 1 && digitalRead(27) == 0 &&
digitalRead(33) == 0){
        DIPSW = 200;
      }else if(digitalRead(25) == 0 && digitalRead(26) == 1 && digitalRead(27) == 1 &&
digitalRead(33) == 0){
        DIPSW = 250;
      }else if(digitalRead(25) == 0 && digitalRead(26) == 0 && digitalRead(27) == 1 &&
digitalRead(33) == 0){
        DIPSW = 300;
      }else if(digitalRead(25) == 0 && digitalRead(26) == 0 && digitalRead(27) == 1 &&
digitalRead(33) == 1){
        DIPSW = 350;
      }else if(digitalRead(25) == 0 && digitalRead(26) == 0 && digitalRead(27) == 0 &&
digitalRead(33) == 1){
        DIPSW = 400;
      }else if(digitalRead(25) == 1 && digitalRead(26) == 1 && digitalRead(27) == 1 &&
digitalRead(33) == 1){
        DIPSW = 1000;
      }else{
        DIPSW = 500;
      }
}
```

```
void valveactO(){
   digitalWrite(LED, HIGH);
   digitalWrite(VALVE_OPEN, HIGH);
   delay(500);
   digitalWrite(VALVE_OPEN, LOW);
   VL = 1;
}

void valveactC(){
   digitalWrite(LED, LOW);
   digitalWrite(VALVE_CLOSE, HIGH);
   delay(500);
   digitalWrite(VALVE_CLOSE, LOW);
   VL = 0;
}

void loop ( void ) {
   checkDIP();
   server.handleClient();
   delay(500);

   Serial.println("pulseCounter: " + String(pulseCounter));
   Serial.println("at " + String(millis()));
   previous_time = millis() - base_time;

 if(previous_time >= 86400000) {
        base_time = millis();
        pulseCounter =0;
        n = 0;

        digitalWrite(RESET, HIGH);
        delay(200);
        digitalWrite(RESET, LOW);

        waterLV = pulseCounter;
          }else if((pulseCounter > DIPSW) && (VL == 1)) {
            valveactC();
            waterLV = DIPSW;
              if(minute == 24*n && n <= 60){          //記録時間
```

```
                pulse[n] = waterLV / 2;
                  n++;
               }
          }else if((pulseCounter > DIPSW) && (VL == 0)) {
             waterLV = DIPSW;
               if(minute == 24*n && n <= 60){        //記録時間
                  pulse[n] = waterLV / 2;
                    n++;
               }
          }else if((pulseCounter <= DIPSW) && (VL == 0)){
             valveactO();
             waterLV = pulseCounter;
          if(minute ==   24*n && n <= 60){             //記録時間
             pulse[n] = waterLV / 2;
             n++;
           }
          }else{
              waterLV = pulseCounter;
          if(minute ==   24*n && n <= 60){             //記録時間
             pulse[n] = waterLV / 2;
             n++;
           }
      }
      second = previous_time / 1000;
      minute = second / 60;
      hour = minute / 60;
}

void drawGraph() {
   String out = "";
   char temp[200];
   int ost_x = 100;
   int ost_y = 20;

   out += "<svg xmlns=¥"http://www.w3.org/2000/svg¥" version=¥"1.1¥" width=¥"780¥"
height=¥"320¥">¥n";
   sprintf(temp, "<rect x=¥"%d¥" y=¥"%d¥" width=¥"600¥" height=¥"250¥" fill=¥"rgb(204, 204,
255)¥" stroke-width=¥"1¥" stroke=¥"rgb(0, 0, 0)¥" />¥n",ost_x,ost_y);
   out += temp;
      for(int j=0;j<4;j++){
```

```
        sprintf(temp, "<line x1=¥"%d¥" y1=¥"%d¥" x2=¥"%d¥" y2=¥"%d¥" fill=¥"none¥"
stroke=¥"gray¥" stroke-width=¥"1¥" stroke-dasharray=¥"5 5¥" />¥n", ost_x, ost_y+50*(j+1),
ost_x+600, ost_y+50*(j+1));
        out += temp;
      }
   out += "<g stroke=¥"black¥">¥n";

  //y ラベル
   sprintf(temp, "<text writing-mode=¥"tb¥" x=¥"45¥" y=¥"%d¥">%s</text>¥n",115,"飲水量[ml]");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"65¥" y=¥"%d¥">%s</text>¥n",30,"500");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"65¥" y=¥"%d¥">%s</text>¥n",80,"400");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"65¥"
y=¥"%d¥">%s</text>¥n",130,"300");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"65¥"
y=¥"%d¥">%s</text>¥n",180,"200");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"65¥"
y=¥"%d¥">%s</text>¥n",230,"100");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"85¥" y=¥"%d¥">%s</text>¥n",290,"0");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"135¥" y=¥"%d¥">%s</text>¥n",290,"2");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"185¥" y=¥"%d¥">%s</text>¥n",290,"4");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"235¥" y=¥"%d¥">%s</text>¥n",290,"6");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"285¥" y=¥"%d¥">%s</text>¥n",290,"8");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"335¥"
y=¥"%d¥">%s</text>¥n",290,"10");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"385¥"
y=¥"%d¥">%s</text>¥n",290,"12");
      out += temp;
```

```
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"435¥"
y=¥"%d¥">%s</text>¥n",290,"14");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"485¥"
y=¥"%d¥">%s</text>¥n",290,"16");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"535¥"
y=¥"%d¥">%s</text>¥n",290,"18");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"585¥"
y=¥"%d¥">%s</text>¥n",290,"20");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"635¥"
y=¥"%d¥">%s</text>¥n",290,"22");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"685¥"
y=¥"%d¥">%s</text>¥n",290,"24");
      out += temp;
   sprintf(temp, "<text writing-mode=¥"horizontal-tb¥" x=¥"640¥" y=¥"%d¥">%s</text>¥n",310,"経過
時間");
      out += temp;

         int i=0;
         int y = pulse[i];     //  グラフの y の初期値
            for (int x = 100; x < 90+10*n; x += 10) {
            int y2 = pulse[i+1]; //  次の y の値算出;
            sprintf(temp, "<line x1=¥"%d¥" y1=¥"%d¥" x2=¥"%d¥" y2=¥"%d¥" stroke-width=¥"1¥"
/>¥n", x, 270 - y, x + 10, 270 - y2);
            // %d は 10 進数表示の書式
            out += temp;
            y = y2;
            i++;
           }
      out += "</g>¥n</svg>¥n";
      server.send ( 200, "image/svg+xml", out);
}

void handleRoot(){
   String mes = "¥
   <html lang=¥"ja¥">¥n¥
```

```
  <meta charset=¥"utf-8¥">¥n¥
  <meta name=¥"viewport¥" content=¥"width=device-width, initial-scale=1.0¥">¥n¥
  <head>¥n¥
  <meta http-equiv=¥"Refresh¥" content=¥"10¥">¥n¥
  <title>Water Level</title>¥n¥
  </head>¥n¥
  <body style=¥"font-family: sans-serif; background-color: #e0ffff ;¥" >¥n¥
  <h2>                618 飼育室 E1 サル飲水量</h2>¥n¥
  <h2>                ・経過時間： "+ String(hour)+ " 時間 "+ String(minute-hour*60)+ " 分
</h2>¥n¥</body>¥n¥
  <h2>                ・設定水量 ＝ "+ String(DIPSW)+ " [ml]</h2>¥n¥</body>¥n¥
  <h2>                ・飲 水 量 ＝ "+ String(waterLV)+ " [ml]</h2>¥n¥
  <img src=¥"/test.svg¥" />¥
  </body>¥n¥
  </html>¥n";
  server.send(200, "text/html", mes);
}

void handleNotFound() {
  String message = "File Not Found¥n¥n";
  message += "URI: ";
  message += server.uri();
  message += "¥nMethod: ";
  message += ( server.method() == HTTP_GET ) ? "GET" : "POST";
  message += "¥nArguments: ";
  message += server.args();
  message += "¥n";

  for ( uint8_t i = 0; i < server.args(); i++ ) {
    message += " " + server.argName ( i ) + ": " + server.arg ( i ) + "¥n";
  }
  server.send ( 404, "text/plain", message );
}
```