

```

/**
 * Write code for InfluxDBClient library for Arduino
 * Data can be immediately seen in a InfluxDB UI: muscle_sensor measurement
 * Enter WiFi and InfluxDB parameters below
 *
 * Measures signal from Myoware sensors and sends data to Influxdb
 * This example supports only InfluxDB running from unsecure (http://...)
 * For secure (https://...) or Influx Cloud 2 use SecureWrite example
 */

```

```

// Initialize the Client
#ifdef ESP32
#include <WiFiMulti.h>
WiFiMulti wifiMulti;
#define DEVICE "ESP32"
#elif defined(ESP8266)
#include <ESP8266WiFiMulti.h>
ESP8266WiFiMulti wifiMulti;
#define DEVICE "ESP8266"
#endif

```

```

#include <InfluxDbClient.h> //You might need to download the library for this
#include <InfluxDbCloud.h>

```

```

// WiFi AP SSID
#define WIFI_SSID "BTHub6-39SG"
// WiFi password
#define WIFI_PASSWORD "FcFhrnJdWmw6"
// InfluxDB v2 server url, e.g. https://eu-central-1-1.aws.cloud2.influxdata.com
(Use: InfluxDB UI -> Load Data -> Client Libraries)
#define INFLUXDB_URL "http://192.168.1.218:8086" //192.168.1.218 is my
computer ip address, 8086 is the port that the influxdb is running from
// InfluxDB v2 server or cloud API authentication token (Use: InfluxDB UI ->
Data -> Tokens -> <select token>)
#define INFLUXDB_TOKEN
"5O8UT2pNQy4KPByHgdRkAJF_pEJSVyut29puYUNpA-
dv5ns6wm9I10M5z6ul6f7ZDbliYBzJgUxlpfPeJWiUxA=="
// InfluxDB v2 organization id (Use: InfluxDB UI -> User -> About -> Common
Ids )
#define INFLUXDB_ORG "londonparkourproject"
// InfluxDB v2 bucket name (Use: InfluxDB UI -> Data -> Buckets)
#define INFLUXDB_BUCKET "mymacbookpro"

```

```

// Set timezone string according to https://www.gnu.org/software/libc/manual/

```

```

html_node/TZ-Variable.html
// Examples:
// Pacific Time: "PST8PDT"
// Eastern: "EST5EDT"
// Japanesse: "JST-9"
// Central Europe: "CET-1CEST,M3.5.0,M10.5.0/3"
#define TZ_INFO "CET-1CEST,M3.5.0,M10.5.0/3"

// InfluxDB client instance with preconfigured InfluxCloud certificate
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET,
INFLUXDB_TOKEN, InfluxDbCloud2CACert);

// Data point
Point sensor("muscles_sensor");

void setup() {
  Serial.begin(115200);

  //Setup pins - Ovando added
  pinMode(39, INPUT);

  // Setup wifi
  WiFi.mode(WIFI_STA);
  wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);

  Serial.print("Connecting to wifi");
  while (wifiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    delay(1);
  }
  Serial.println();

  // Add tags
  sensor.addTag("device", DEVICE);
  sensor.addTag("SSID", WiFi.SSID());

  // Accurate time is necessary for certificate validation and writing in batches
  // For the fastest time sync find NTP servers in your area: https://
  www.pool.ntp.org/zone/
  // Syncing progress and the time will be printed to Serial.
  timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov");

  // Check server connection
  if (client.validateConnection()) {
    Serial.print("Connected to InfluxDB: ");
    Serial.println(client.getServerUrl());
  }
}

```

```

    } else {
        Serial.print("InfluxDB connection failed: ");
        Serial.println(client.getLastErrorMessage());
    }
}

```

```

// Write data - sends the data to influxdb on host1
void loop() {
    // Clear fields for reusing the point. Tags will remain untouched
    sensor.clearFields();

    float sensorValue = analogRead(39);
    float millivolt = (sensorValue/1023)*5;

    Serial.print("Sensor Value: ");
    Serial.println(sensorValue);

    Serial.print("Voltage: ");
    Serial.print(millivolt*1000);
    Serial.println(" mV");
    Serial.println("");

    // Store measured value into point
    // Report RSSI of currently connected network
    // sensor.addField("rssi", WiFi.RSSI());

    // Report voltage
    sensor.addField("analog", sensorValue);
    sensor.addField("voltage", millivolt*1000);

    // Print what are we exactly writing
    Serial.print("Writing: ");
    Serial.println(sensor.toLineProtocol());

    // If no Wifi signal, try to reconnect it
    if ((WiFi.RSSI() == 0) && (wifiMulti.run() != WL_CONNECTED)) {
        Serial.println("Wifi connection lost");
    }

    // Write point
    if (!client.writePoint(sensor)) {
        Serial.print("InfluxDB write failed: ");
        Serial.println(client.getLastErrorMessage());
    }
}

```

```
}  
  
//Wait 0.1ms  
Serial.println("Wait 10 ms");  
delay(1);  
}
```