

```

//
// 温湿度計プログラム TempMeter Ver. 1.2
// Seeed Studio XIAO RA4M1版
//
#include <Wire.h>

const int SHT31_ADDR = 0x45;          // Sensor AE-SHT31 のアドレス
const int AQM0802_ADDR = 0x3E;       // LCD AE-AQM0802 のアドレス
const int AQM0802_RESET = 7;         // LCD AE-AQM0802 のハードリセットはデジタルピン7

void setup() {
    Wire.begin();                     // I2Cを初期化

    // Sensor AE-SHT31の初期化
    writeSensorCommand(0x30, 0xA2);   // ソフトリセット
    writeSensorCommand(0x30, 0x41);   // ステータスレジスタのデータ消去

    // LCD AE-AQM0802の初期化
    pinMode(AQM0802_RESET, OUTPUT);   // ハードリセットピンを出力に設定
    digitalWrite(AQM0802_RESET, LOW); // ハードリセット開始
    delay(50);
    digitalWrite(AQM0802_RESET, HIGH); // ハードリセット終了
    delay(50);
    writeLCDCommand(0x38);            // 拡張コマンド終了 IS = 0
    writeLCDCommand(0x39);            // 拡張コマンド開始 IS = 1
    writeLCDCommand(0x14);            // 拡張コマンド Internal OSC frequency
    writeLCDCommand(0x70);            // 拡張コマンド Contrast set
    writeLCDCommand(0x56);            // 拡張コマンド Power/ICON control/Contrast set
    writeLCDCommand(0x6C);            // 拡張コマンド Follower control
    writeLCDCommand(0x38);            // 拡張コマンド終了 IS = 0
    delay(200);
    writeLCDCommand(0x0C);            // コマンド Display ON
}

void loop() {
    int i, data[6];
    float temp, humi;
    char buffer[9];

    writeSensorCommand(0x24, 0x00);   // 単発測定
    delay(30);
    Wire.requestFrom(SHT31_ADDR, 6);  // I2C通信開始 6byteのデータの要求

```

```

for (i = 0; i < 6; i++) {
    data[i] = Wire.read();           // データ読み込み
}
Wire.endTransmission();           // I2C通信終了

temp = -45.0 + 175.0 * (data[0] * 256.0 + data[1]) / 65535.0;
humi = 100.0 * (data[3] * 256.0 + data[4]) / 65535.0;

String temp_str(temp, 1);         // tempを小数点以下2位を四捨五入して文字列に変換
while(temp_str.length() < 5) {   // 5文字未満なら空白を前に追加
    temp_str = ' ' + temp_str;
}
temp_str = "T " + temp_str + 'C'; // TとCを追加
temp_str.toCharArray(buffer, 9);
writeLCD_1(buffer);              // LCDの1行目に表示

String humi_str(humi, 1);        // humiを小数点以下2位を四捨五入して文字列に変換
while(humi_str.length() < 5) {  // 5文字未満なら空白を前に追加
    humi_str = ' ' + humi_str;
}
humi_str = "H " + humi_str + '%'; // Hと%を追加
humi_str.toCharArray(buffer, 9);
writeLCD_2(buffer);              // LCDの2行目に表示

delay(1000);
}

// Sensorコマンド書き込み
void writeSensorCommand(byte msb, byte lsb) {
    Wire.beginTransaction(SHT31_ADDR); // I2C通信開始
    Wire.write(msb);                  // MSB書き込み
    Wire.write(lsb);                  // LSB書き込み
    Wire.endTransmission();           // I2C通信終了
    delay(1);
}

// LCDコマンド書き込み
void writeLCDCommand(byte command) {
    Wire.beginTransaction(AQM0802_ADDR); // I2C通信開始
    Wire.write(0x00);                   // control 最終データとコマンドを指定
    Wire.write(command);                // data コマンド書き込み
    Wire.endTransmission();             // I2C通信終了
}

```

```

    delay(1);
}

// LCDの1行目にデータ書き込み
void writeLCD_1(char str[8]) {
    int i;
    Wire.beginTransaction(AQM0802_ADDR); // I2C通信開始
    Wire.write(0x80); // control 連続データとコマンドを指定
    Wire.write(0x80); // data コマンド Set DDRAM address, 1行目の1文字目のアドレスは0x00
    for (i = 0; i < 8; i++) {
        if (i == 7) {
            Wire.write(0x40); // control 最終データとデータを指定
        } else {
            Wire.write(0xC0); // control 連続データとデータを指定
        }
        Wire.write(str[i]); // data データ 文字書き込み
    }
    Wire.endTransmission(); // I2C通信終了
}

// LCDの2行目にデータ書き込み
void writeLCD_2(char str[8]) {
    int i;
    Wire.beginTransaction(AQM0802_ADDR); // I2C通信開始
    Wire.write(0x80); // control 連続データとコマンドを指定
    Wire.write(0x80 + 0x40); // data コマンド Set DDRAM address, 2行目の1文字目のアドレスは0x40
    for (i = 0; i < 8; i++) {
        if (i == 7) {
            Wire.write(0x40); // control 最終データとデータを指定
        } else {
            Wire.write(0xC0); // control 連続データとデータを指定
        }
        Wire.write(str[i]); // data データ 文字書き込み
    }
    Wire.endTransmission(); // I2C通信終了
}

```