

```

//
// 7セグLED表示の温度・湿度計
// LEDダイナミック表示はライブラリを使用しない。
//   BW428G-B15(アノードコモン)専用
//   2015/11/10
//   Ver-3
//   Arduino ProMini(5V/16MHz)対応   ミニシールド用
//   温度のマイナス表示に対応、「温度」「湿度」の順に表示
//
#include <avr/sleep.h>
#include <MsTimer2.h>
#include "DHT.h"

#define DHTPIN 14    // what digital pin we're connected to DHT11
#define DHTTYPE DHT11 // DHT 11

#define SWPIN 15    // Start/Stop SW.
#define TIMEOUT 10 // 表示消灯タイム

// LED点灯パターン
// 配列は「0,G,F,E,D,C,B,A」の順
#define MINUS 18
static const byte digitCodeMap[] = {
  B00111111, // 0 0
  B00000110, // 1 1
  B01011011, // 2 2
  B01001111, // 3 3
  B01100110, // 4 4
  B01101101, // 5 5
  B01111101, // 6 6
  B00000111, // 7 7
  B01111111, // 8 8
  B01101111, // 9 9
  B01110111, // A 10
  B01111100, // b 11
  B00111001, // C 12
  B01011110, // d 13
  B01111001, // E 14
  B01110001, // F 15
  B01111000, // t 16
  B00000000, // BLANK 17
  B01000000 // Minus 18
};

byte static selColumn[] = {10, 11, 12, 13}; // 左が上位桁(C1,C2,C3,C4)
byte static segmentPins[] = {2, 6, 8, 4, 5, 9, 3, 7}; // A..G, DPの順

```

```

byte static digit = 0; // 次に表示する桁位置
byte static numBuff[4]; // 表示用バッファ

static boolean mode = true; // 表示モード(true=表示点灯/false=消灯)
static int dsp_time = 0; // 表示消灯タイマ

DHT dht(DHTPIN, DHTTYPE);

//
// 各桁の数値を取り出し配列にセットする
// 入力：数値、桁数
// 出力：バッファ
//
void getNum( signed int n, int i, byte *buff ) {
    if ( n < 0 ) { // 値がマイナスの場合、
        buff[0] = MINUS; // 最上位はマイナス記号
        n = n * -1; // 表示値は絶対値を取る
    }
    while ( n >= 1 ) {
        buff[i - 1] = n % 10; // 一の位の余り
        n = n / 10; // 取得済みの余りを"n"から除去
        i--;
    }
}

//
// LED消灯
// 桁選択ビットをOFFにしてLEDを消灯させる
//
void dspOFF() {
    for ( int i = 0; i < 4; i++ ) {
        digitalWrite( selColumn[i], LOW ); // 桁選択=OFF
    }
}

//
// 割り込みタスク
// タイマー割り込みで、LEDの表示をリフレッシュする
//
// 入力条件：(グローバル変数)
//     digit=選択する桁(0~3)
//     numBuff=表示数字(4桁)
//
void segDisplay() {

```

```

int n = numBuff[ digit ];    // この割り込みで表示する数字を取り出す
byte pattern = digitCodeMap[n];

dspOFF();
digitalWrite( selColumn[ digit ], HIGH );    // 桁選択=ON
for ( int i = 0; i <= 7; i++ ) {
    if (( pattern & 0x01 ) != 0 ) {
        digitalWrite( segmentPins[i], LOW );    // 表示セグメントON
    } else digitalWrite( segmentPins[i], HIGH ); // 表示セグメントOFF
    pattern = pattern >> 1;
}
digitalWrite( segmentPins[7], LOW );    // コロン表示 ON
digit++;    // 次回の割り込みで処理する桁位置を指定しておく
if ( digit > 3 ) digit = 0;
}

//
// 初期化処理
//
void setup() {
    // 桁選択ビットを指定
    for ( int i = 0; i < 4; i++ ) {
        pinMode(selColumn[i], OUTPUT); //C1~C4
    }
    pinMode( 6, OUTPUT );
    // セグメント表示を指定
    for ( int i = 0; i < 8; i++ ) {
        pinMode(segmentPins[i], OUTPUT); //A..G,DP
    }
    MsTimer2::set(5, segDisplay);    // LED表示周期
    pinMode(SWPIN, INPUT_PULLUP);    // 表示スタート・ストップSW

    MsTimer2::start();    // 割り込みタイマ開始
    dsp_time = 0;    // 表示時間カウンタ開始
}

void loop() {
    signed int temp;
    int mod, i;

    if ( digitalRead( SWPIN ) == LOW ) { // SWが押されたらモードを変える(表示を
ON/OFF)
        mode = !mode;    // ただし、ループは2秒なので、ボタンは長押し(2
秒)
        if ( mode != false ) {

```

```

    MsTimer2::start();          // 割り込みタイマ開始
    dsp_time = 0;              // 表示時間カウント開始
}
}
//
// スリープモード
//
if ( mode == false ) { // SW長押しでスリープ(自動スリープしない)
//if (( mode == false ) || ( dsp_time > TIMEOUT )) { // SW長押し、タイムアウトでス
スリープ
    MsTimer2::stop();          // 割り込みタイマ停止
    dspOFF();                  // LED消灯
    mode = false;
    // 表示を停止してスリープ
    //set_sleep_mode(SLEEP_MODE_PWR_DOWN);
    //sleep_mode();
}

//float t = -9;
//float h = 23;
float t = dht.readTemperature(); // DHT11から温度を読む
float h = dht.readHumidity();    // DHT11から湿度を読む
if ( t < 0 ) h = h * -1;         // 温度がマイナスの時、湿度もマイナスにして表示を
合わせる
temp = t * 100 + h;             // LEDの上位2桁に温度、下位2桁に湿度を表示

getNum( temp, 4, numBuff );
delay(2000);                    // DHT11は2秒おきにサンプルする

dsp_time++;                     // 消灯用タイマーをダウンカウント
}

```