```
//
// 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-E(カソードコモン)専用
//    Arduino ProMini(3.3V/8MHz)対応
//  温湿度センサにSI-7021(高精度センサ)を使用
// ---------------------------------------
//    2017/7/20
//      Ver-1
//
#include <avr/sleep.h>
#include <MsTimer2.h>
#include "SparkFun_Si7021_Breakout_Library.h"
#include <Wire.h>

#define SWPIN 2          // モード選択SW
#define TIMEOUT 10       //  表示消灯タイム

#define NUM_DISP   false     //  数値のみ表示
#define FUNC_DISP  true     //  温度・湿度を合わせて表示

//  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
  B01110110,  // H 19
  B01010000   // r 20
};
```

```
byte static selColumn[] = {10, 11, 12, 13, A0, A1, A2, A3};   //
桁選択　左が上位桁(C1,C2,C3,C4)
byte static segmentPins[] = {3, 4, 5, 6, 7, 8, 9};         // セグメント選択　A..
Gの順(DPは常時点灯)
//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[8]; //  表示用バッファ

static boolean mode = FUNC_DISP;    //  表示モード(true=全表示／false=数値のみ)
static int  dsp_time = 0;           //  表示消灯タイマ

//Create Instance of HTU21D or SI7021 temp and humidity sensor and MPL3115A2
barrometric sensor
Weather sensor;

//
//  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 ) {       //  値かマイナスの場合、
    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 < 8; i++ ) {
    digitalWrite( selColumn[i], HIGH );      //  桁選択=OFF(K)
  }
}

//
//  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：(グローパル変数)
```

```
//          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], LOW );      //  桁選択=ON(K)
  for ( int i = 0; i < 7; i++ ) {
    if (( pattern & 0x01 ) == 0 ) {           //  (K)
      digitalWrite( segmentPins[i], LOW );        //  表示セグメントOFF
    } else digitalWrite( segmentPins[i], HIGH );  //  表示セグメントON
    pattern = pattern >> 1;
  }
  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 >= 8 ) digit = 0;
}

//
//  初期化処理
//
void setup() {
  Serial.begin( 9600 );
  sensor.begin();                   //  温湿度センサーの初期化

  //  桁選択ビットを指定
  for ( int i = 0; i < 8; i++ ) {
    pinMode(selColumn[i], OUTPUT);   //C1～C8
  }
  //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
  }
  MsTimer2::set(1, segDisplay);       // LED表示周期(5ms)
  pinMode(SWPIN, INPUT_PULLUP);       //  表示モード変更SW

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


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

  if ( digitalRead( SWPIN ) == LOW ) {  // SWが押されたら表示モードを変える
```

```
    mode = !mode;
//  ただし、ループは2秒なので、ボタンは長押し(2秒)
  }

  //
  //  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)
  //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;

  //  温湿度センサーから全てを読み込む
  t = sensor.getTemp();       // 温度を読む
  h = sensor.getRH();          // 湿度を読む
  //sprintf( tmp, "%4d", t );
  temp = t * 100;
  humid = h * 100;

  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;                // フロート値を4桁の符号付き整数に変換
  Serial.print("RH: ");
  Serial.print( h );
  Serial.print("%  Temp: ");
  Serial.print( t );
  Serial.println(" *C ");

  Serial.print("-->RH: ");
  Serial.print( humid );
  Serial.print("%  Temp: ");
  Serial.print( temp );
  Serial.println(" *C ");

  if ( mode == NUM_DISP ) {
    getNum( temp, 4, numBuff );
    getNum( humid, 4, numBuff + 4 );
  } else {
    getNum( temp, 4, numBuff );
    numBuff[2] = 17;
    numBuff[3] = 12;
```

```
    getNum( humid, 4, numBuff + 4 );
    numBuff[6] = 20;
    numBuff[7] = 19;
  }
  delay(2000);

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