

```

//
// DS1302 Real Time Clock
// カラーLCDシールドに時刻及び温度、湿度を表示する
//
// Ver-3.0 2016/3/5 表示器をカラーLCDに変更
// DHT11の接続をA1(D15)からA5(D19)へ変更
// Ver-2.0 2015/11/3 時刻の設定機能を追加
// Ver-1.0 2015/11/3
//
// I assume you know how to connect the DS1302.
// DS1302: CE pin -> Arduino Digital 13
// I/O pin -> Arduino Digital 12
// SCLK pin -> Arduino Digital 11

#include <DS1302.h>
#include "DHT.h"
#include <MsTimer2.h>

// Humidity sensor(DHT11) definitions
#define DHTPIN 19 // what digital pin we're connected to DHT11
#define DHTTYPE DHT11 // DHT 11

// define some values used by the panel and buttons
#define backLIGHT 10

#define btnRIGHT 0
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
#define btnSELECT 4
#define btnNONE 5

#include <Adafruit_GFX.h> // Core graphics library
#include <Adafruit_TFTLCD.h> // Hardware-specific library
#include <Fonts/FreeSerifBold12pt7b.h>

// The control pins for the LCD can be assigned to any digital or
// analog pins...but we'll use the analog pins as this allows us to
// double up the pins with the touch screen (see the TFT paint example).
#define LCD_CS A0 // Chip Select goes to Analog 3
#define LCD_CD A1 // Command/Data goes to Analog 2
#define LCD_WR A2 // LCD Write goes to Analog 1
#define LCD_RD A3 // LCD Read goes to Analog 0

#define LCD_RESET A4 // Can alternately just connect to Arduino's reset pin

// When using the BREAKOUT BOARD only, use these 8 data lines to the LCD:
// For the Arduino Uno, Duemilanove, Diecimila, etc.:

```

```

// D0 connects to digital pin 8 (Notice these are
// D1 connects to digital pin 9 NOT in order!)
// D2 connects to digital pin 2
// D3 connects to digital pin 3
// D4 connects to digital pin 4
// D5 connects to digital pin 5
// D6 connects to digital pin 6
// D7 connects to digital pin 7
// For the Arduino Mega, use digital pins 22 through 29
// (on the 2-row header at the end of the board).

// Assign human-readable names to some common 16-bit color values:
#define BLACK 0x0000
#define BLUE 0x001F
#define DARKBLUE 0x000F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);
// If using the shield, all control and data lines are fixed, and
// a simpler declaration can optionally be used:
// Adafruit_TFTLCD tft;

char c = 0x00;
float tmp; // 温度
float hum; // 湿度
int array_time[ 3 ]; // 時刻設定用配列 時・分・秒
int mode = 0; // 動作モード 0:通常時計表示/1:設定モード

DS1302 rtc(13, 12, 11);
DHT dht(DHTPIN, DHTTYPE); // 温湿度センサにDHT11を使用

Time rtcTime;

//
// 割り込みタスク
// LCDに日付、時刻、温度及び湿度を表示する
//
void displayLCD() {
  char StrBuf[2];
  //char timestr[8];
  int cls;

```

```

unsigned long start = micros();
char* timestr = rtc.getTimeStr();
switch ( timestr[6] ) {
    case '0':
        cls = 0X0000;    //Black
        break;
    case '1':
        cls = 0X000F;    //DarkBlue
        break;
    case '2':
        cls = 0X01E0;    //DarkGreen
        break;
    case '3':
        cls = 0X7800;    //DarkRed
        break;
    case '4':
        cls = 0X79E0;    //DarkBrown
        break;
    case '5':
        cls = 0X3807;    //DarkPurple
        break;
    default:
        cls = 0x3800;
        break;
}

tft.fillScreen( cls );
    // Time
tft.setCursor(60, 100);
tft.setTextColor(YELLOW); tft.setTextSize(3);
//char* timestr = rtc.getTimeStr();
//Serial.println( timestr );
for ( int i = 0; i < 5; i++ ) {
    tft.print( timestr[i] );
}

// Day-of-Week
tft.setCursor(60, 28);
tft.setTextColor(WHITE); tft.setTextSize(1);
char* dow = rtc.getDOWStr();
tft.print(dow[0]);
tft.print(dow[1]);
tft.print(dow[2]);
tft.print(". ");

// Date
tft.println(rtc.getDateStr(FORMAT_LONG, FORMAT_MIDDLEENDIAN));

```

```

// Time
/*tft.setCursor(60, 100);
tft.setTextColor(YELLOW); tft.setTextSize(3);
//char* timestr = rtc.getTimeStr();
Serial.println( timestr );
for ( int i = 0; i < 5; i++ ) {
    tft.print( timestr[i] );
}
*/
tft.setCursor(40, 150);
tft.setTextColor(CYAN); tft.setTextSize(1);
tft.print( "Celsius          Humidity" );
// 温度
tft.setCursor(30, 210);
tft.setTextColor(GREEN); tft.setTextSize(3);
sprintf(StrBuf, "%02d", int(tmp));
tft.print( StrBuf );
tft.setTextSize(2); tft.print("'C");
tft.print( "   " );

// 湿度
tft.setTextSize(3);
sprintf(StrBuf, "%02d", int(hum));
tft.print( StrBuf );
tft.setTextSize(2); tft.print('%');
}

//
// 設定モードのメッセージ表示
//
void dspMsg() {
    tft.fillScreen(DARKBLUE);
    tft.setCursor(0, 120);
    tft.setTextColor(WHITE); tft.setTextSize(1);
    tft.println("UP:Time +1 Minutes."); // LCDにメッセージを表示
    tft.println("DOWN:Time -1 Minutes.");
    tft.println("Enter:FIX the new time.");
}

//
// RTCの時刻を設定するボタンの処理
//
void adjRTC( int lcd_key ) {
    char StrBuf[2];

    switch ( lcd_key ) {
        case btnUP:

```

```

    dspMsg();
    array_time[ 1 ]++;      // 『分』をプラス
    tft.setCursor(60, 28);
    sprintf(StrBuf, "%02d", array_time[0]);
    tft.print(StrBuf);     // 時を表示
    tft.print( ":" );
    sprintf(StrBuf, "%02d", array_time[1]);
    tft.print(StrBuf);     // 分を表示
    break;

case btnDOWN:
    dspMsg();
    array_time[ 1 ]--;      // 『分』をマイナス
    tft.setCursor(60, 28);
    sprintf(StrBuf, "%02d", array_time[0]);
    tft.print(StrBuf);     // 時を表示
    tft.print( ":" );
    sprintf(StrBuf, "%02d", array_time[1]);
    tft.print(StrBuf);     // 分を表示
    break;

case btnRIGHT:
    rtc.setTime( array_time[0], array_time[1], 0 );      // RTCに時刻をセット
    MsTimer2::start();      // 割り込みタイマ開始
    mode = 0;
    break;
}
}

void getRTCtime( int *array ) {
    rtcTime = rtc.getTime();
    array[0] = rtcTime.hour; // 『時』を得る
    array[1] = rtcTime.min;  // 『分』を得る
    array[2] = rtcTime.sec;  // 『秒』を得る
}

//
// LCDシールドのキーボード(ボタン)を読む
//
// my buttons when read are centered at these valies: 0, 144, 329, 504, 741
// we add approx 50 to those values and check to see if we are close
// We make this the 1st option for speed reasons since it will be the most
// likely result
//
int read_LCD_buttons() {      // read the buttons

    int adc_key_in = analogRead(5);      // read the value from the sensor

```

```

//Serial.println( adc_key_in );
if (adc_key_in > 1000) return btnNONE;
if (adc_key_in < 50)   return btnSELECT;
if (adc_key_in < 220) return btnUP;
if (adc_key_in < 430) return btnDOWN;
if (adc_key_in < 555) return btnLEFT;
if (adc_key_in < 850) return btnRIGHT;

return btnNONE;           // when all others fail, return this.
}

//
// 初期化处理
//
void setup()
{
  tft.reset();
  tft.setFont(&FreeSerifBold12pt7b);
  uint16_t identifier = tft.readID();
  tft.begin(identifier);
  tft.fillScreen(DARKBLUE);
  tft.setRotation(3);

  dht.begin();
  // Set the clock to run-mode, and disable the write protection
  rtc.halt(false);
  rtc.writeProtect(false);

  tmp = dht.readTemperature(); // DHT11から温度を読む
  hum = dht.readHumidity();    // DHT11から湿度を読む
  displayLCD();

  MsTimer2::set(5000, displayLCD); // 表示周期
  MsTimer2::start();              // 割り込みタイマ開始
  // Setup Serial connection
  //Serial.begin(9600);

  // The following lines can be commented out to use the values already stored
  // in the DS1302
  //rtc.setDOW(TUESDAY);           // Set Day-of-Week to FRIDAY
  //rtc.setTime( 15, 38, 00);      // Set the time to 12:00:00 (24hr format)
  //rtc.setDate(3, 11, 2015);     // Set the date to August 6th, 2010
}

void loop()
{
  int lcd_key;           // LCDキーコード
  int t;                 // DHT11読み込み用カウンタ

```

```

if ( ( t % 200 ) == 0 ) {          // 2秒に1回DHT11を読む
    tmp = dht.readTemperature(); // DHT11から温度を読む
    hum = dht.readHumidity();    // DHT11から湿度を読む
}

delay( 10 );                      // 10msに一回キーをチェックする
lcd_key = read_LCD_buttons();     // read the buttons
//
// SELECTボタンが押されたら、設定モードに入る
//
if ( lcd_key == btnSELECT) {
    mode = 1;                      // 設定モードフラグ
    MsTimer2::stop();              // 割り込みタイマ停止
    dspMsg();

    getRTCtime( array_time );      // 現在の時刻を読み込んでおく
}
if ( mode == 1 ) {
    adjRTC( lcd_key );             // 設定キーの処理
    delay( 200 );                 // キーのチャタリング待ちタイマ
}
t++;                               // DHT用タイマを進める
}

```