Code:

#include <SD.h>

#include <LiquidCrystal.h>

#include <RTClib.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

RTC\_DS3231 rtc;

DateTime now;

#define button1 A1

#define button2 A2

#define DS18B20\_PIN A3

File dataLog;

boolean sd\_ok = 0;

void setup()

{

pinMode(button1, INPUT\_PULLUP);

pinMode(button2, INPUT\_PULLUP);

rtc.begin();

lcd.begin(20, 4);

lcd.setCursor(0, 3);

lcd.print("Temp:");

Serial.begin(9600);

Serial.print("Initializing SD card...");

if ( !SD.begin() )

Serial.println("initialization failed!");

else {

sd\_ok = 1;

Serial.println("initialization done.");

if( SD.exists("Log.txt") == 0 )

{

Serial.print("\r\nCreate 'Log.txt' file ... ");

dataLog = SD.open("Log.txt", FILE\_WRITE);

if(dataLog) {

Serial.println("OK");

dataLog.println(" DATE | TIME | TEMPERATURE");

dataLog.println("(dd-mm-yyyy)|(hh:mm:ss)|");

dataLog.close();

}

else

Serial.println("error creating file.");

}

}

Serial.println("\r\n DATE | TIME | TEMPERATURE");

Serial.println("(dd-mm-yyyy)|(hh:mm:ss)|");

}

void loop()

{

now = rtc.now();

RTC\_display();

if( !digitalRead(button1) )

if( debounce() )

{

while( debounce() );

byte hour = edit( now.hour() );

byte minute = edit( now.minute() );

byte day = edit( now.day() );

byte month = edit( now.month() );

byte year = edit( now.year() - 2000 );

rtc.adjust(DateTime(2000 + year, month, day, hour, minute, 0));

while(debounce());

}

static byte p\_second;

if( (now.second() % 2 == 0) && (p\_second != now.second()) )

{

unsigned int ds18b20\_temp;

char buffer1[12], buffer2[26];

bool sensor\_ok = 0;

p\_second = now.second();

if( ds18b20\_read(&ds18b20\_temp) )

{

sensor\_ok = 1;

if (ds18b20\_temp & 0x8000)

{

ds18b20\_temp = ~ds18b20\_temp + 1;

sprintf(buffer1, "-%02u.%04u%cC", (ds18b20\_temp/16) % 100, (ds18b20\_temp & 0x0F) \* 625, 223);

}

else

{

if (ds18b20\_temp/16 >= 100)

sprintf(buffer1, "%03u.%04u%cC", ds18b20\_temp/16, (ds18b20\_temp & 0x0F) \* 625, 223);

else

sprintf(buffer1, " %02u.%04u%cC", ds18b20\_temp/16, (ds18b20\_temp & 0x0F) \* 625, 223);

}

}

else

sprintf(buffer1, " ERROR ");

lcd.setCursor(5, 3);

lcd.print(buffer1);

sprintf( buffer2, " %02u-%02u-%04u | %02u:%02u:%02u | ", now.day(), now.month(), now.year(),

now.hour(), now.minute(), now.second() );

if(sensor\_ok) {

buffer1[8] = 194;

buffer1[9] = 176;

buffer1[10] = 'C';

buffer1[11] = '\0';

}

Serial.print(buffer2);

Serial.println(buffer1);

if(sd\_ok)

{

dataLog = SD.open("Log.txt", FILE\_WRITE);

dataLog.print( buffer2 );

dataLog.println( buffer1 );

dataLog.close();

}

}

delay(100);

}

void RTC\_display()

{

char \_buffer[17];

char dow\_matrix[7][10] = {" SUNDAY ", " MONDAY ", " TUESDAY ", "WEDNESDAY",

"THURSDAY ", " FRIDAY ", "SATURDAY "};

lcd.setCursor(4, 0);

lcd.print( dow\_matrix[now.dayOfTheWeek()] );

sprintf( \_buffer, "TIME: %02u:%02u:%02u", now.hour(), now.minute(), now.second() );

lcd.setCursor(0, 1);

lcd.print(\_buffer);

sprintf( \_buffer, "DATE: %02u-%02u-%04u", now.day(), now.month(), now.year() );

lcd.setCursor(0, 2);

lcd.print(\_buffer);

}

byte edit(byte parameter)

{

static byte i = 0, y\_pos,

x\_pos[5] = {6, 9, 6, 9, 14};

char text[3];

sprintf(text,"%02u", parameter);

if(i < 2)

y\_pos = 1;

else

y\_pos = 2;

while( debounce() );

while(true) {

while( !digitalRead(button2) ) {

parameter++;

if(i == 0 && parameter > 23)

parameter = 0;

if(i == 1 && parameter > 59)

parameter = 0;

if(i == 2 && parameter > 31)

parameter = 1;

if(i == 3 && parameter > 12)

parameter = 1;

if(i == 4 && parameter > 99)

parameter = 0;

sprintf(text,"%02u", parameter);

lcd.setCursor(x\_pos[i], y\_pos);

lcd.print(text);

delay(200);

}

lcd.setCursor(x\_pos[i], y\_pos);

lcd.print(" ");

unsigned long previous\_m = millis();

while( (millis() - previous\_m < 250) && digitalRead(button1) && digitalRead(button2) ) ;

lcd.setCursor(x\_pos[i], y\_pos);

lcd.print(text);

previous\_m = millis();

while( (millis() - previous\_m < 250) && digitalRead(button1) && digitalRead(button2) ) ;

if(!digitalRead(button1))

{

i = (i + 1) % 5;

return parameter;

}

}

}

bool debounce ()

{

byte count = 0;

for(byte i = 0; i < 5; i++)

{

if ( !digitalRead(button1) )

count++;

delay(10);

}

if(count > 2) return 1;

else return 0;

}

bool ds18b20\_start()

{

bool ret = 0;

digitalWrite(DS18B20\_PIN, LOW);

pinMode(DS18B20\_PIN, OUTPUT);

delayMicroseconds(500);

pinMode(DS18B20\_PIN, INPUT);

delayMicroseconds(100);

if (!digitalRead(DS18B20\_PIN))

{

ret = 1;

delayMicroseconds(400);

}

return(ret);

}

void ds18b20\_write\_bit(bool value)

{

digitalWrite(DS18B20\_PIN, LOW);

pinMode(DS18B20\_PIN, OUTPUT);

delayMicroseconds(2);

digitalWrite(DS18B20\_PIN, value);

delayMicroseconds(80);

pinMode(DS18B20\_PIN, INPUT);

delayMicroseconds(2);

}

void ds18b20\_write\_byte(byte value)

{

byte i;

for(i = 0; i < 8; i++)

ds18b20\_write\_bit(bitRead(value, i));

}

bool ds18b20\_read\_bit(void)

{

bool value;

digitalWrite(DS18B20\_PIN, LOW);

pinMode(DS18B20\_PIN, OUTPUT);

delayMicroseconds(2);

pinMode(DS18B20\_PIN, INPUT);

delayMicroseconds(5);

value = digitalRead(DS18B20\_PIN);

delayMicroseconds(100);

return value;

}

byte ds18b20\_read\_byte(void)

{

byte i, value;

for(i = 0; i < 8; i++)

bitWrite(value, i, ds18b20\_read\_bit());

return value;

}

bool ds18b20\_read(int \*raw\_temp\_value)

{

if (!ds18b20\_start())

return(0);

ds18b20\_write\_byte(0xCC);

ds18b20\_write\_byte(0x44);

while(ds18b20\_read\_byte() == 0);

if (!ds18b20\_start())

return(0);

ds18b20\_write\_byte(0xCC);

ds18b20\_write\_byte(0xBE);

\*raw\_temp\_value = ds18b20\_read\_byte();

\*raw\_temp\_value |= (unsigned int)(ds18b20\_read\_byte() << 8);

return(1);

}