/\*\*\*\*\*\*\*\*\*\*\*\*\* Valeur de température mesurée sur 12 points de litière, affichage LCD et enregistrement SD \*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <LiquidCrystal.h> // ajout de la librairie affichage

#include <SD.h> // ajout de la librairie carte SD

#include <RTClib.h> // ajout de la librairie horloge

#include <SPI.h>

#include <Wire.h> // librairie de communication i2C

int analogPin = 2; // voltage mesuré sur la broche 2

int num = 100; // valeur initiale capteur = 100

int n;

int year;

int month;

int day;

int hour;

int minute;

int second;

int yeardeb;

int monthdeb;

int daydeb;

int hourdeb;

int minutedeb;

int seconddeb;

int temps;

Sd2Card card;

const int chipSelect = 10;

char heure;

float temp = 0.0;

int numanc = 0;

float voltage = 0.0;

// Vérifier les broches LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // liaison afficheur à 4 bits de données branché sur les broches 7, 6, 5, 4, 3 et 2

RTC\_DS1307 RTC; // Classe RTC\_DS1307 puce d'horloge

File logfile;

void setup()

{

  Serial.begin(9600); // initialise la communication série à 9600 bauds

  pinMode(analogPin, INPUT); // entrée du capteur

  SD.begin(10); //initialise la carte SD

  analogReference(INTERNAL); // val ref = 1.1V entraine une résolution de 0.1 °C

  lcd.begin(16, 2); // utilisation d'un écran 16 colonnes et 2 lignes

  RTC.begin(); // démarrage de la librairie RTClib.h

  // Si RTC ne fonctionne pas

  if (! RTC.isrunning())

  {

    Serial.println("RTC ne fonctionne pas !");

    // Synchronisation de l’heure de la puce RTC sur celle de l’ordinateur

    RTC.adjust(DateTime(\_\_DATE\_\_, \_\_TIME\_\_));

    //Cela fonctionne également :

    //RTC.adjust(DateTime("Dec  5 2012","12:00:00"));

    //RTC.adjust(DateTime(2014, 1, 21, 3, 0, 0));

  }

  RTC.adjust(DateTime(\_\_DATE\_\_, \_\_TIME\_\_));

  lcd.clear();

  lcd.setCursor(0, 0); // curseur dans le coin supérieur droit

  //Tester si la carte fonctionne

  if (!card.init(SPI\_HALF\_SPEED, chipSelect))

  {

    lcd.print("ERREUR SD");

    delay(20000); // attente 20 secondes

    return;

  }

  else

  {

    lcd.print("SD OK");

  }

  delay(5000); // attente 5 secondes

  // Mise à jour de l’horloge

  DateTime now = RTC.now();

  yeardeb =  now.year();

  monthdeb = now.month();

  daydeb= now.day();

  hourdeb = now.hour();

  minutedeb = now.minute();

  seconddeb =  now.second();

  logfile = SD.open("data2.txt", FILE\_WRITE);

  // Enregistrement du 1er champ sur carte SD

  logfile.print("numero");

  logfile.print(",");

  logfile.print("heure");

  logfile.print(",");

  logfile.print("secondes");

  logfile.print(",");

  logfile.print("temperature");

  logfile.print(",");

  logfile.print("jour:");

  logfile.print(daydeb);

  logfile.print(",");

  logfile.print("mois:");

  logfile.print(monthdeb);

  logfile.println();

  logfile.close();

  // Affichage de l’heure sur LCD

  lcd.clear();

  lcd.setCursor(0, 0); // curseur dans le coin supérieur droit

  lcd.print(hourdeb);

  lcd.print("h");

  lcd.setCursor(5, 0);

  lcd.print(minutedeb);

  delay(10000); // attente 10 secondes

}

void loop()

{

  n = 1;

  while (n < 13)

  {

    numanc = 0.;

    while (abs(num - numanc) > 1 )

    {

      // Récupération de l'heure RTC

       DateTime now= RTC.now();

       year =  now.year();

       month = now.month();

       day = now.day();

       hour = now.hour();

       minute = now.minute();

       second = now.second();

       lcd.clear();

       lcd.setCursor(0, 0); // curseur dans le coin supérieur droit

       lcd.print("At.Temp");

       lcd.setCursor(8, 0);

       lcd.print(n);

       lcd.setCursor(10, 0);

       lcd.print("="); // on écrit le message "At. temp n ="

       delay (5000); // attente 5 secondes

       numanc = num;

       num = analogRead(analogPin); // mesure tension sur la broche 2

       voltage = num \* (1.10 / 1023);

       temp = voltage \* 100; // conversion en température (10mV/°)

       char message[16] = "";

       lcd.setCursor(11, 0);

       lcd.print(temp,1); // afficher la température sur LCD

      lcd.home();

    }

    lcd.clear();

    lcd.setCursor(0, 0); // curseur dans le coin supérieur droit

    lcd.print("Temp OK");

    lcd.setCursor(9,0);

    lcd.print(n); // on écrit le message "Temp OK"

    lcd.setCursor(11, 0);

    lcd.print(temp, 1); // afficher la température finale sur LCD

    temps=abs((hour-hourdeb)\*3600+(minute-minutedeb)\*60+(second-seconddeb));

    // Enregistrement du champ de données sur SD

    logfile = SD.open("data2.txt", FILE\_WRITE);

    logfile.print(n);

    logfile.print(",");

    logfile.print(hour, DEC);

    logfile.print(':');

    logfile.print(minute, DEC);

    logfile.print(':');

    logfile.print(second, DEC);

    logfile.print(",");

    logfile.print(temps);

    logfile.print(",");

    logfile.print(temp);

    logfile.println();

    logfile.close();

    n = n + 1;

    numanc=0;

    num=100;

    delay(5000); // attente 5 secondes

    lcd.clear();

    lcd.setCursor(0, 0); // curseur dans le coin supérieur droit

    lcd.print("point suivant...");

    delay(10000); // attente 10 secondes pour le déplacement de la sonde

  }

}