/\* Programa para aquisição de pressão e temperatura

Autor: Victor Oliveira Ferreira

Data: 10/02/2020\*/

//Definição do número de dados adiquiridos pela média móvel

#define N 50

//Definição do fundo de escala em bits

float SensorOffset = 21;

//Inclusão do termopar

#include <Thermistor.h> //Pacote

Thermistor temp(2); //Definição da entrada do termopar

//Definição de valores usados para medição do tempo

int temporal = 0; //Tempo na análise

int temporalaux1 = 0; //Tempo total

int temporalaux2 = 0; //Tempo total no início da análise

//Definição da condição para que o void loop seja reiniciado

int condition = 1;

void setup() {

Serial.begin(9600);

//Pino do LED

pinMode(8,OUTPUT); //Define qual pino

digitalWrite(8,LOW);//Define qual o estado

}

//Declaração de variáveis

//=============== PRESSÃO ===============

int j = 1, k = 0, P\_antiga = 0; //Contadores de tempo, número de valores para cálculo da média

float medida; //medida obtida no sensor de pressão em bits descontando o fundo de escala

float filtro; //valor filtrado via média móvel para a medida de pressão

float vals[N]; //array das medidas com N valores para a pressão

float H0 = 0, P0 = 0; //Valores iniciais para a pressão

int dt = 200; //intervalo de tempo entre as medidas da média temporal

int fim = 4 /\*minutos\*/ \*60;//Definição do tempo para finalizar a média temporal

//============= TEMPERATURA =============

int filtered; //valor filtrado via média móvel para a medida de temperatura

int valsT[N]; //array das medidas com N valores para a temperatura

void loop() {

condition = 1; //Retorna o valor para condição um, permitindo que o programa seja executado por completo

//=============== PRESSÃO ===============

//É necessário escolher com qual tipo de medida vai utilizar

medida = analogRead(A0) - SensorOffset ; //valores para a pressão em bits descontando o fundo de escala

//Filtro média móvel para a pressão

for(int i = N - 1; i > 0; i--){

vals[i] = vals[i-1];

}

vals[0] = medida; //com o SensorOffset

int sum = 0;

for(int i = 0; i < N; i++){

sum = sum + vals[i];

}

filtro = sum / N;

//Medida da pressão a partir dos bits lidos pelo sensor

float P\_medida = 1000\*((filtro/(1023)) - 0.04)/0.09; //equação presente no datasheet do sensor

//============= TEMPERATURA =============

int value = temp.getTemp(); //valor obtido pelo thermistor

//Filtro média móvel para a temperatura

for(int l = N - 1; l > 0; l--){

valsT[l] = valsT[l-1];

}

valsT[0] = value;

int sumT = 0;

for(int l = 0; l < N; l++){

sumT = sumT + valsT[l];

}

filtered = (sumT / N);

//================ AMBOS ================

j = j+1; //Contador para a condição de início da média temporal

temporal = millis()/1000-temporalaux2; //Tempo de análise

//Início da aquisição: dados não levados em consideração para a média temporal

if (j < 4\*N){//(dif > 1){

Serial.print (P\_medida);

Serial.print ("Pa");

Serial.print ('\t');

Serial.print (filtered);

Serial.print ("°C");

Serial.print ('\t');

Serial.print (temporal);

Serial.println ("seg");

k=0; //Retorna o valor 0 para o número de dados de pressão obtidos

delay(dt/2); //Espera entre aquisição de dados não levados em consideração na média temporal

//Início da aquisição com média temporal

}else{

k = k+1; //conta a quantidade de dados adiquiridos

P0 = P0 + P\_medida; //Soma o dado antigo com o novo

float P\_medtemp = P0/k; //Calcula a média de todos os dados obtidos

/\* H0 = H0 + h1;

float H = H0/k;\*/

Serial.print (P\_medtemp);

Serial.print ("Pa");

//Serial.print(H);

Serial.print ('\t');

Serial.print (filtered);

Serial.print ("°C");

Serial.print ('\t');

Serial.print (temporal);

Serial.println ("seg");

delay(dt); //Tempo de espera entre aquisição de dados

//Algorítimo de final de aquisição de dados

if (temporal >= fim){ //critério de parada

Serial.println ("FIM!");

Serial.println ("TROQUE A TOMADA DE PRESSAO!");

for (condition = 1; condition < 2; condition = 1\*condition){ //Enquanto o valor condicional for um, só piscar o led e esperar comando

digitalWrite (8, HIGH);

delay (500);

digitalWrite (8, LOW);

delay (500);

//Abertura da porta serial

if (Serial.available() > 0){

if (Serial.read()!=0){ //Condição para reinício do void loop

condition = 2;

j = 4\*N; //Retira a necessidade da fase inicial da aquisição de dados, pulando direto para a média temporal

k = 0; //Zera contador de valores da média temporal

P0 = 0; //Zera valor "zero" da pressão

temporalaux1 = millis()/1000; //Calcula o tempo total até o comando de reinício

temporalaux2 = temporalaux1; //armazena o tempo total até o comando de reinício

fim = 3 /\*minutos\*/ \* 60; //Definição do novo tempo para finalizar a média temporal

}

}

}

}

}

}