

```

#include<SoftwareSerial.h>
#include<LowPower.h>
#include<DallasTemperature.h>
#include<OneWire.h>

float voltagem;//voltagem a variavel, em float (numero real com 8
casas decimais), precisao
float temp = 0;//temperatura do sensor
int aviso = 0;//aviso a variavel, em int (numero real com duas casas
ou quatro casas decimais), começa em 0

void setup() { //codigo que so corre uma vez, no inicio do programa
  Serial.begin(9600);//inicio de comunicacao com o computador a 9600
bits, se necessario
}

int warning() { //função de aviso (por causa de ruídos de voltagem,
por vezes o circuito pode ir abaixo de 1volts)
  //se a voltagem se manter abaixo de 1 volt por mais de 4 medições
entao o sensores ativam se

  if (voltagem < 1) { //se a voltagem for menor que 1 volt
    aviso++;//então adiciona um aviso
    Serial.println(aviso);

    if(aviso > 4){ // e se o aviso for maior q 4, entao corre os
sensores
      sensor_temp();//função dos sensor temperatura
      Serial.println("4");
    }

  } else { //se a voltagem for maior q 1 entao os avisos voltam a zero
    Serial.println("vivo");
    Serial.println(aviso);
    aviso = 0;
    return aviso;
  }
}

int sensor_temp() { //função do sensor temperatura. é chamado pela
função warning

```

```

#define ONE_WIRE_BUS 2 // Data wire is conntec to the Arduino
digital pin 2, as chamadas sao feitas aqui para poupar energia
OneWire oneWire(ONE_WIRE_BUS); // Setup a oneWire instance to
communicate with any OneWire devices
DallasTemperature sensors(&oneWire); // Pass our oneWire reference
to Dallas Temperature sensor
sensors.begin();

// Call sensors.requestTemperatures() to issue a global temperature
and Requests to all devices on the bus
// Why "byIndex"? You can have more than one IC on the same bus. 0
refers to the first IC on the wire
sensors.requestTemperatures(); //pedida da leitura dos sensores
para a biblioteca dallas
temp = sensors.getTempCByIndex(0); // é escolhido o primeiro sensor
na linha one wire
Serial.print("Celsius temperature: ");
Serial.print(temp);
if(temp > 90){ //se a temperatura for maior q 90°C, chama a função
do GSM
    gsm_sms();
}

}

void gsm_sms() {
    #define SIM800_TX 8
    #define SIM800_RX 9
    SoftwareSerial sim800(SIM800_TX, SIM800_RX);
    sim800.begin(9600);
    delay(1000);
    sim800.write("AT+CMGF=1\r");
    delay(1000);
    sim800.write("AT+CMGS=\"918661154\"\r");
    delay(1000);
    sim800.write("Data: Local: Temp:");
    delay(1000);
    sim800.write(0x1A);
    Serial.println("sms grrehhhehehethheheehr");
}

/*int low_power(int sleep_time) {

```

```

    for(int low_temp = 0; low_temp < sleep_time; low_temp++){//dorme
durante o tempo escolhido
        LowPower.powerDown(SLEEP_2S, ADC_OFF, BOD_OFF);
        Serial.println(low_temp);
    }

} */

int volt_med() {
    int sensorValue = analogRead(A0); //le a porta A0
    voltagem = sensorValue * (5.0 / 1023.0); //a porta le entre 0-1023
valores. Esta é a conversao para volts
    Serial.println(voltagem); //mostra o valor que le, ctrl + shift + m
}

void loop() { //codigo que corre repetidamente

    volt_med(); //função de medição de voltagem
    warning(); //chama a função warning
    //low_power(3); //dorme durante um tempo
    delay(500); //espera 0.5s para fazer a proxima medição

}

```