

Ministério da Educação

**Universidade Tecnológica Federal do Paraná**

Câmpus

Santa Helena

Circuitos Digitais

DS

Ciência da Computação



**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**CIRCUITOS DIGITAIS**

**CIÊNCIA DA COMPUTAÇÃO**

**MATHEUS FERREIRA RAMOS**

**RANDER CARNEIRO**

**RUAN LISBOA**

**BLUEFROZER**

**SANTA HELENA**

**2018**

**SUMÁRIO**



[1 DESCRIÇÃO 3](#_Toc13985)

[2 LISTA DE MATERIAIS 4](#_Toc13986)

[3 IMAGENS DO PROJETO 5](#_Toc13987)

[4 CÓDIGO FONTE 8](#_Toc13988)

[5 CONCLUSÃO 13](#_Toc13989)

# DESCRIÇÃO

O projeto foi feito para conseguir medir a temperatura interna do Isopor através do Sensor de Temperatura DS18B20, o objetivo inicial era construir um Cooler Refrigerador com Placas de Peltier, porém o mesmo teria um custo acima do esperado, então optamos por focar na parte de desenvolvimento do código do Arduino, que não alteraria independente das Placas de Peltier, para melhorar ainda mais o projeto optamos por exibir a temperatura interna do Isopor online, ou seja, tudo faz parte de um projeto de Internet das coisas.

Ao longo do projeto usamos algumas bibliotecas, cada uma com uma determinada função, a SoftReset, na qual sua função é de resetar o código quando o sistema trava, agora tanto OneWire e o DallasTemperature são bibliotecas obrigatórias para a utilização do Sensor, já o LiquidCrystal é o responsável para o funcionamento do display que mostra as informações.

**Sensor de Temperatura DS18B20 Prova D’água**

Falando mais sobre este componente, o O sensor de temperatura DS18B20 pode efetuar leituras com precisão de até ±0,5 ºC, e enviar as informações para o microcontrolador utilizando apenas 1 fio. Outras características do DS18B20:

-CI : DS18B20 (Datasheet)

-Tensão de operação: 3 a 5,5V

-Faixa de medição de temperatura: -55 ºC a +125 ºC

-Precisão: +/- 0,5 ºC entre -10 ºC e +85 ºC

-Ponta de aço inoxidável (6 x 50mm), com cabo de 108 cm de comprimento

Cada sensor possui um endereço físico, assim você pode colocar vários sensores como esse em um mesmo barramento, utilizando uma única porta do microcontrolador, e efetuar a leitura individualmente.

**Objetivo**

O objetivo inicial do projeto era que o mesmo, através do wifi shield, pegasse os dados imprimidos da temperatura e mandasse, inicialmente, através do NODE-RED, os valores para a internet, porém, com muita dificuldade decidimos trocar para outro programa, Thingspeak, que também encontramos muita dificuldade, e por falta de tempo, decidimos não manda os dados para a rede.

# 2 LISTA DE MATERIAIS

Para que você possa reproduzir o projeto você irá desses seguintes itens:

* 1x Arduino Mega 2560
* 1x Raspberry Pi 3
* 1x Caixa de Isopor
* 1x Sensor de Temperatura DS18B20
* 1x Display LCD 20x4
* 1x Resistor pull-up de 4,7K
* 1x Protoboard 840 furos (Item opcional)

# IMAGENS DO PROJETO

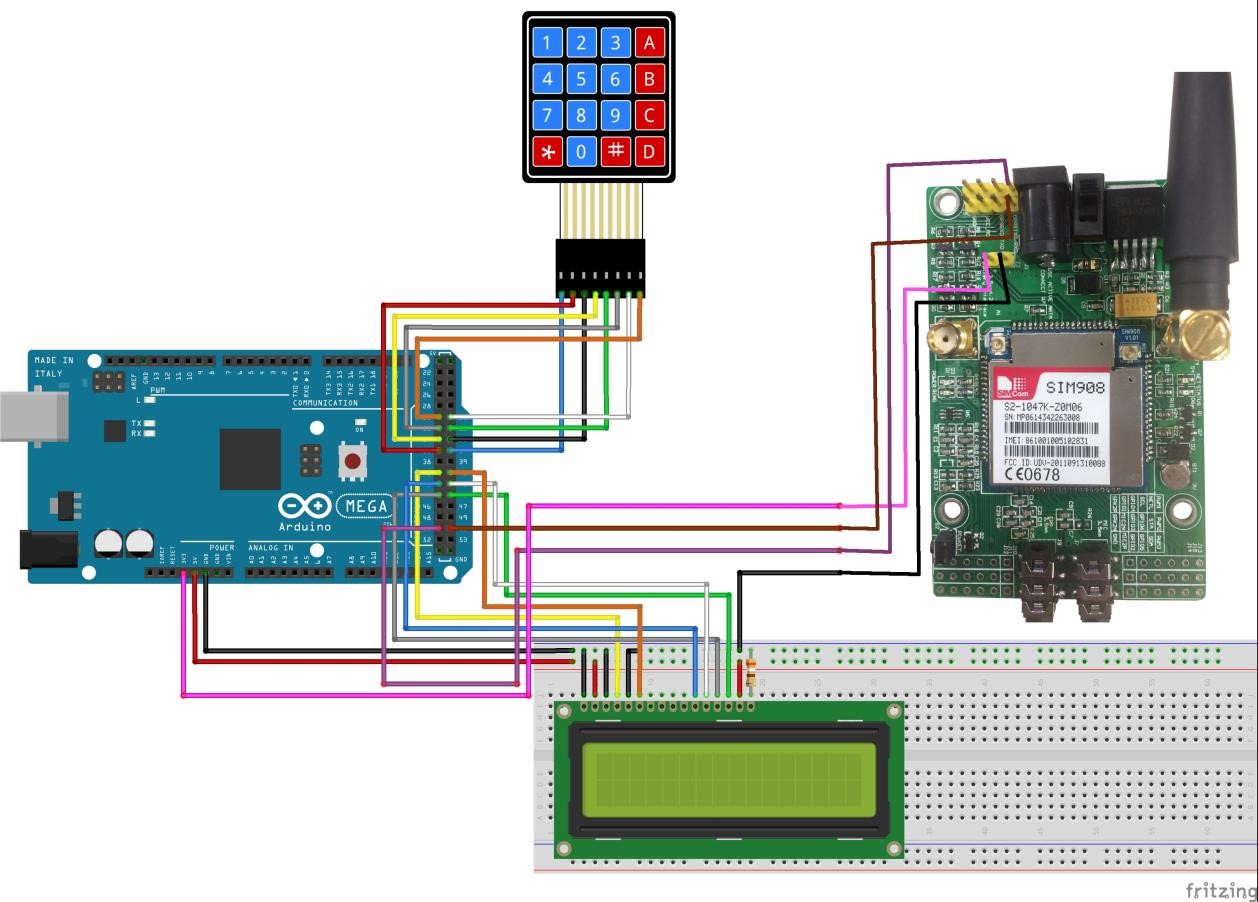


Imagem para auxiliar na reprodução do projeto

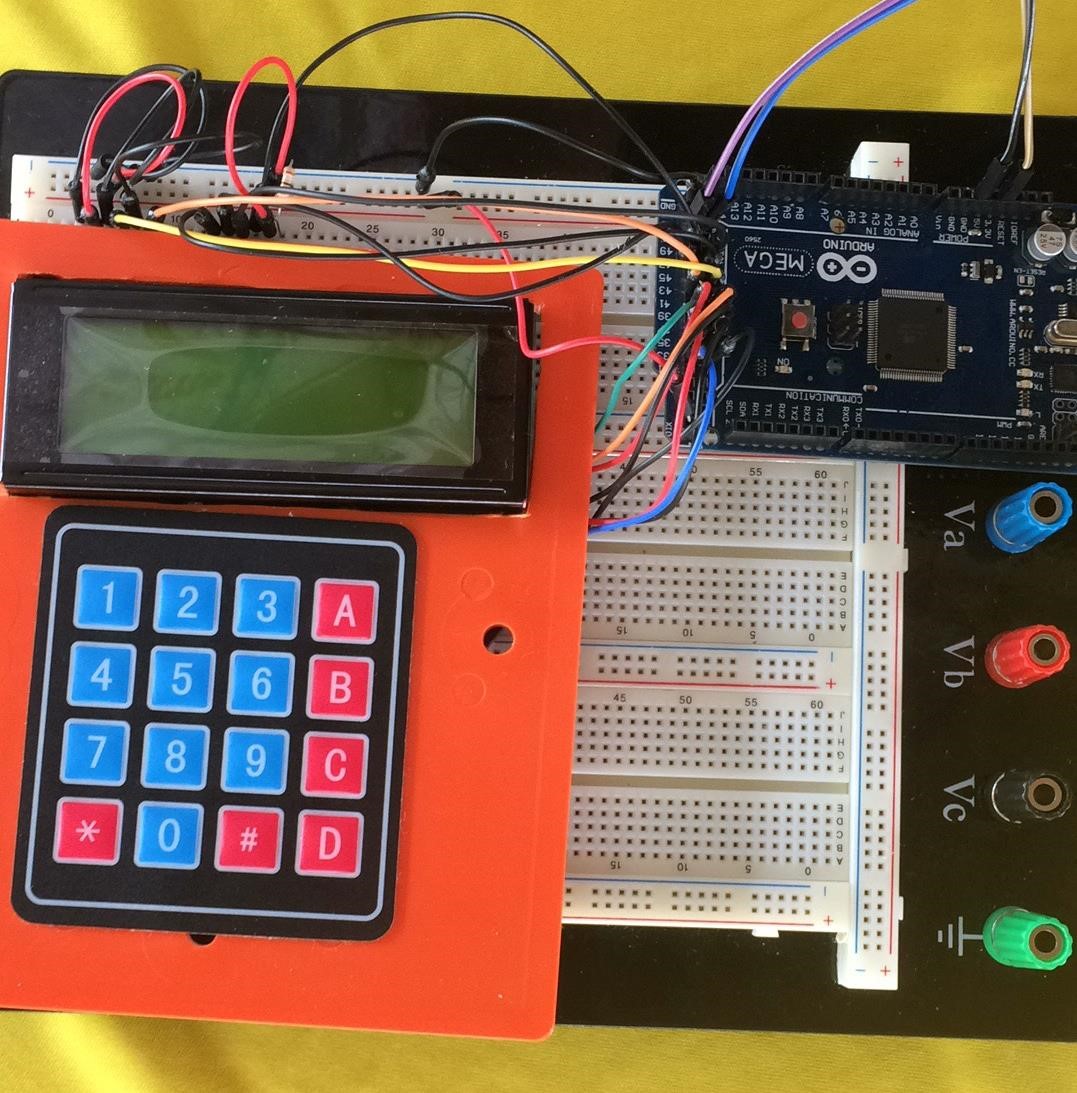


Imagem do Arduino com o Teclado Matricial

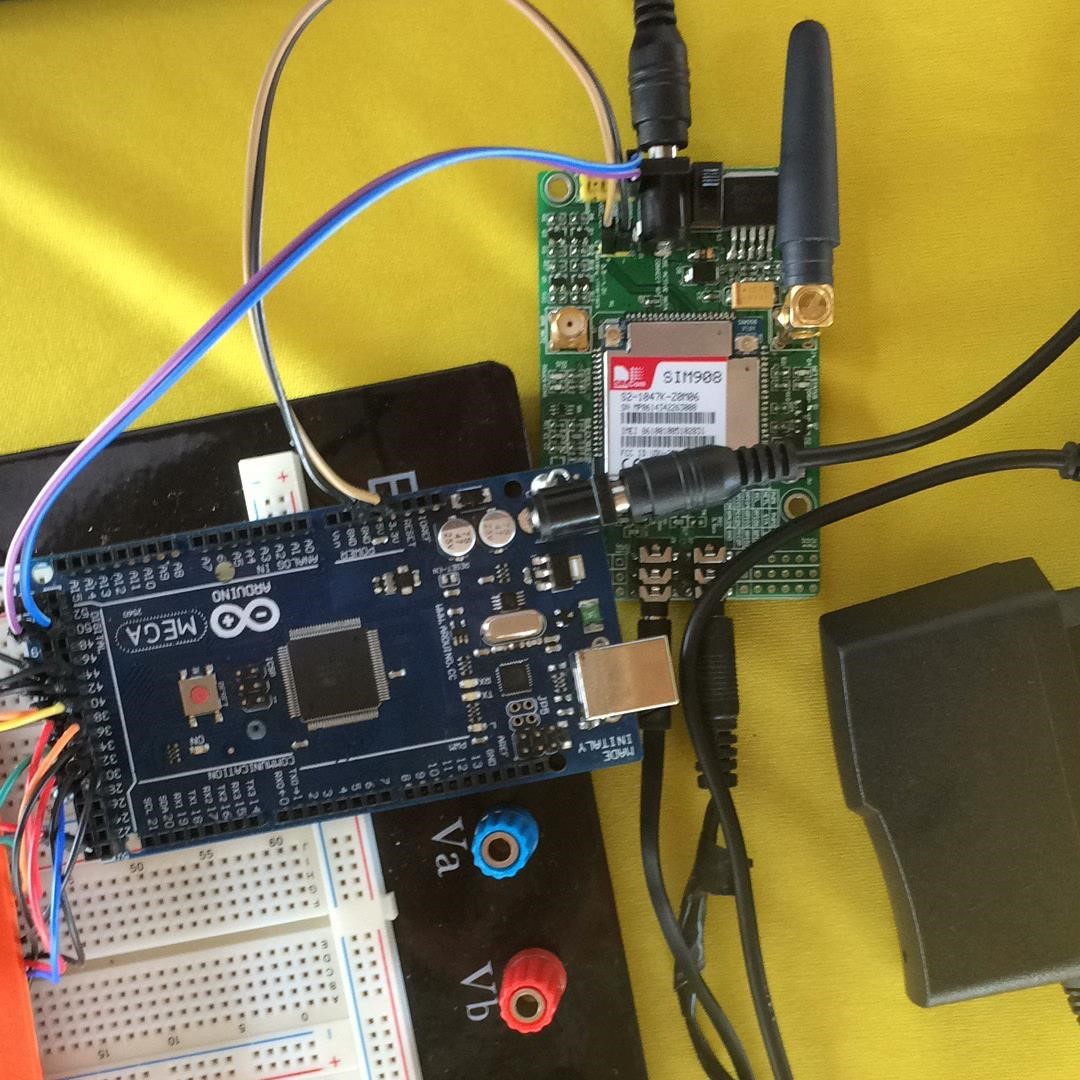


Imagem do Arduino Mega com o Módulo SIM908

# CÓDIGO FONTE

//BIBLIOTECAS USADAS

#include <SoftReset.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#include <LiquidCrystal.h>

#include <Adafruit\_CC3000.h>

#include <SPI.h>

#include "utility/debug.h"

#include "utility/socket.h"

//---------------------------------------//

//DEFINIÇOES

// Porta do pino de sinal do DS18B20

#define ONE\_WIRE\_BUS 28

//Pinos WIFI

#define ADAFRUIT\_CC3000\_IRQ   3  //Pino de interrupcao (D2 ou D3)

#define ADAFRUIT\_CC3000\_VBAT  5  //Pode ser qualquer pino digital

#define ADAFRUIT\_CC3000\_CS    10 //Preferencialmente pino 10 do Arduino Uno

Adafruit\_CC3000 cc3000 = Adafruit\_CC3000(ADAFRUIT\_CC3000\_CS, ADAFRUIT\_CC3000\_IRQ,

                                         ADAFRUIT\_CC3000\_VBAT,SPI\_CLOCK\_DIVIDER);

#define WLAN\_SSID       "TP-LINK\_88359E"   //Nome da rede - Ate 32 caractereres

#define WLAN\_PASS       ""     //Senha da rede

#define WLAN\_SECURITY   WLAN\_SEC\_UNSEC  //Tipo de seguranca

/\*

#define WLAN\_SSID       "Quantum Go 2"   //Nome da rede - Ate 32 caractereres

#define WLAN\_PASS       "12345678"     //Senha da rede

#define WLAN\_SECURITY   WLAN\_SEC\_WPA2  //Tipo de seguranca

\*/

//--------------------------------------------------------------//

//VARIAVEIS GLOBAIS

// Define uma instancia do oneWire para comunicacao com o sensor

OneWire oneWire(ONE\_WIRE\_BUS);

// Armazena temperaturas minima e maxima

float tempMin = 999;

float tempMax = 0;

float tempC;

DallasTemperature sensors(&oneWire);

DeviceAddress sensor1;

// Inicializa o LCD

LiquidCrystal lcd(22, 23, 24, 25, 26, 27);

int screenWidth = 20;

int screenHeight = 4;

//SCROLL LCD

String line1 = "SEJA BEM VINDO";

String line2 = " BLUEFROZEN";

int stringStart, stringStop = 0;

int scrollCursor = screenWidth;

int tamanho =0;

 //---------------------------------------------------//

//prototypes

void printarSerial();

void bemVindo();

void scroll\_sup();

void iniciar();

//void enviarParaThing();

void lerSensor();

void mostrarLcd();

void openSystem();

//void resolveWebsite();

//-------------------------------------------------------//

//FUNÇOES

//SCROLL

void scroll\_sup(){

  lcd.clear();

  if(stringStart == 0 && scrollCursor > 0)

  {

    scrollCursor--;

    stringStop++;

  } else if (stringStart == stringStop){

    stringStart = stringStop = 0;

    scrollCursor = screenWidth;

  } else if (stringStop == line1.length() && scrollCursor == 0) {

    stringStart++;

  } else {

    stringStart++;

    stringStop++;

  }

}

//---------

//ABRIR SISTEMA

void openSystem(){

  // Localiza e mostra enderecos dos sensores

  Serial.println("Localizando sensores DS18B20...");

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Localizando sensores DS18B20...");

  delay(3000);

  Serial.print("Foram encontrados: ");

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Foram encontrados: ");

  Serial.print(sensors.getDeviceCount(), DEC);

  Serial.println(" sensores.");

  lcd.setCursor(19,0);

  lcd.print(sensors.getDeviceCount(), DEC);

  delay(2000);

  if (!sensors.getAddress(sensor1, 0)) {

     Serial.println("Sensores nao encontrados !");

     lcd.clear();

     lcd.setCursor(0,0);

     lcd.print("Sensores nao encontrados!");

     delay(2000);

  }

  // Mostra o endereco do sensor encontrado no barramento

  Serial.print("Endereco sensor: ");

  mostra\_endereco\_sensor(sensor1);

  Serial.println();

  Serial.println();

  Serial.println(F("Inicializando..."));

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Inicializando...");

  if (!cc3000.begin())

  {

    Serial.println(F("Shield nao encontrado. Verifique as conexoes !"));

    while(1){

        soft\_restart();

    }

  }

  Serial.println(F("Tentando conectar-se a rede: ")); Serial.println(WLAN\_SSID);

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Tentando conectar-se");

  lcd.setCursor(0,1);

  lcd.print("a rede: ");

  lcd.setCursor(8,1);

  lcd.print(WLAN\_SSID);

  if (!cc3000.connectToAP(WLAN\_SSID, WLAN\_PASS, WLAN\_SECURITY))

  {

    Serial.println(F("Falha !"));

    while(1);

  }

  Serial.println(F("Conectado!"));

  Serial.println(F("Requisitando endereco DHCP"));

  while (!cc3000.checkDHCP())

  {

    delay(100);

  }

  //Exibe as informacoes da conexao

  while (! displayConnectionDetails())

  {

    delay(1000);

  }

  Serial.println(F("Aguardando conexao..."));

}

//ABRIR SISTEMA TERMINA AQUI

//INICIAR

void iniciar(){

    Serial.begin(115200);

    sensors.begin();

    lcd.begin(20, 4);

    pinMode(A2, INPUT);

}

//FIM DO INICIAR

//INICIO DO DA FUNÇAO PARA LER O SENSOR

void lerSensor(){

  // Le a informacao do sensor

  sensors.requestTemperatures();

  tempC = sensors.getTempC(sensor1);

  // Atualiza temperaturas minima e maxima

  if (tempC < tempMin)

  {

    tempMin = tempC;

  }

  if (tempC > tempMax)

  {

    tempMax = tempC;

  }

  //resolveWebsite();

}

//FIM DO LER SENSOR

//MOSTRAR NA SERIAL

void printarSerial(float temp){

   // Mostra dados no serial monitor

  Serial.print("Temp C: ");

  Serial.print(tempC);

  Serial.print(" Min : ");

  Serial.print(tempMin);

  Serial.print(" Max : ");

  Serial.println(tempMax);

}

//FIM

//MOSTRAR NA LCD

void mostrarLcd(){

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Temp.:       ");

  lcd.write(223);

  lcd.print("C");

  lcd.setCursor(7,0);

  lcd.print(tempC);

  lcd.setCursor(0,1);

  lcd.print("L: ");

  lcd.setCursor(3,1);

  lcd.print(tempMin,1);

  lcd.setCursor(8,1);

  lcd.print("H: ");

  lcd.setCursor(11,1);

  lcd.print(tempMax,1);

  lcd.setCursor(5,3);

  lcd.print("BLUEFORZEN");

  delay(3000);

}

//FIM

//MOSTRAR BEM VINDO NA TELA

void bemVindo(){

   for(int i=0; i<53; i++){

      lcd.setCursor(scrollCursor, 0);

      lcd.print(line1.substring(stringStart,stringStop));

      lcd.setCursor(4, 2);

      lcd.print(line2);

      delay(250);

      scroll\_sup(); //Chama a rotina que executa o scroll

      //Verifica o tamanho da string

      tamanho = line1.length();

      if (stringStart == tamanho)

      {

        stringStart = 0;

        stringStop = 0;

      }

   }

}

//FIM ------------

//FUNÇAO BOOLEAN PARA DETALHES DAS CONEXAO

bool displayConnectionDetails(void){

  uint32\_t ipAddress, netmask, gateway, dhcpserv, dnsserv;

  if(!cc3000.getIPAddress(&ipAddress, &netmask, &gateway, &dhcpserv, &dnsserv))

  {

    Serial.println(F("Nao foi possivel ler o endereco IP!\n"));

    return false;

  }

  else

  {

    Serial.print(F("\nIP Addr: ")); cc3000.printIPdotsRev(ipAddress);

    Serial.print(F("\nNetmask: ")); cc3000.printIPdotsRev(netmask);

    Serial.print(F("\nGateway: ")); cc3000.printIPdotsRev(gateway);

    Serial.print(F("\nDHCPsrv: ")); cc3000.printIPdotsRev(dhcpserv);

    Serial.print(F("\nDNSserv: ")); cc3000.printIPdotsRev(dnsserv);

    Serial.println();

    return true;

  }

}

//FIM DA BOOLEAN E DAS FUNÇOES

//----------------------------------------------

//FUNÇOES PRINCIPAIS

//SETUP PARA CHAMAR AS FUNÇOES

void setup(void)

{

  // Iniciar sistema

  iniciar();

  // Mensagem do BlueFrozen

  bemVindo();

  delay(300);

  // Localiza e mostra enderecos dos sensores

  openSystem();

}

void mostra\_endereco\_sensor(DeviceAddress deviceAddress){

  for (uint8\_t i = 0; i < 8; i++)

  {

    if (deviceAddress[i] < 16) Serial.print("0");

    Serial.print(deviceAddress[i], HEX);

  }

}

//-------------------------------------------------------------

//LOOP

void loop(){

  // Le a informacao do sensor

  lerSensor();

  // Mostra dados no serial monitor

  printarSerial(tempC);

  // Mostra dados no LCD

  mostrarLcd();

  delay(60000);

}

# CONCLUSÃO

Não conseguimos chegar ao objetivo final do nosso projeto, que era manda todos os valores de temperatura para a internet at. Porém a parte de imprimir os dados no próprio computador deu certo, realizamos diversos testes e não conseguimos chegar ao resultado esperado, logo, deixamos aqui a sugestão para nos próximos semestres, utilizar raspberry pi 3 para passar os dados via bluetooth.