INSTITUTO FEDERAL
Fluminense
Campus Quissamã

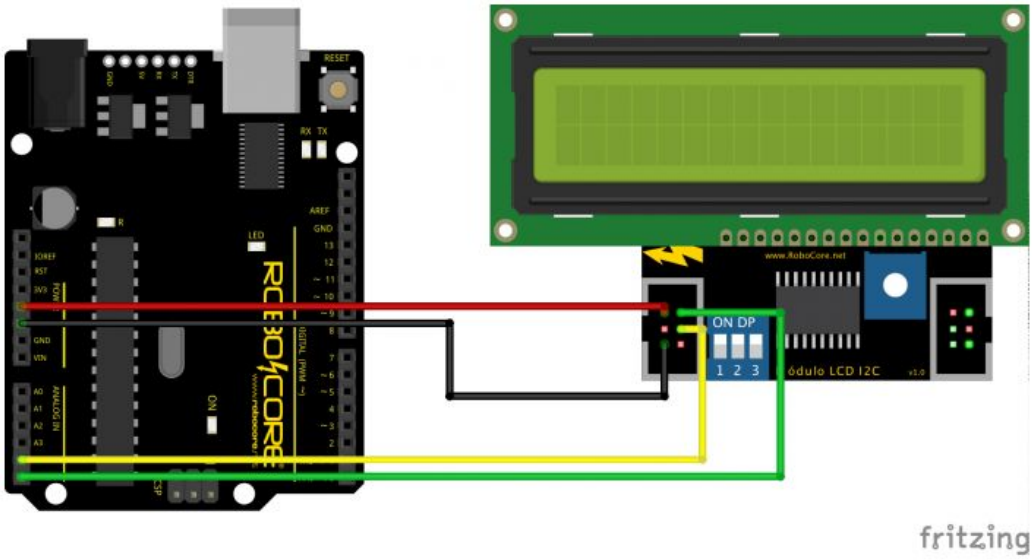
MINISTÉRIO DA EDUCAÇÃO

PÁTRIA AMADA BRASIL
GOVERNO FEDERAL

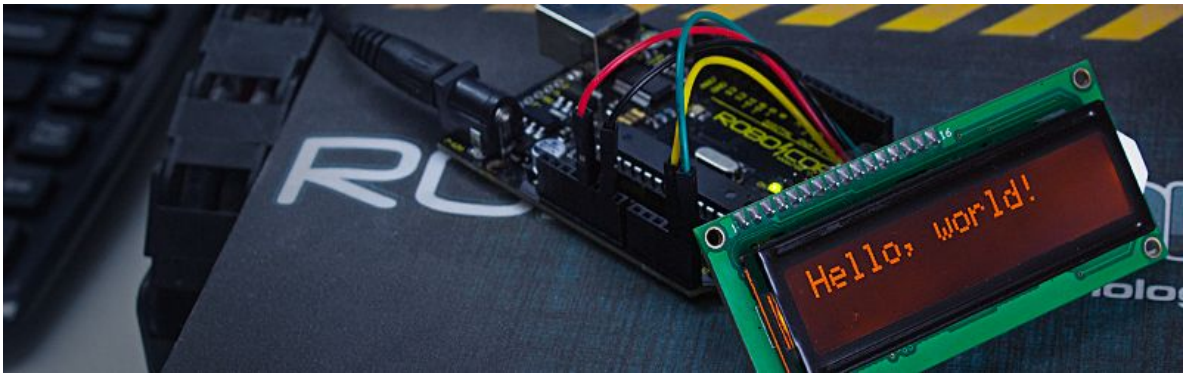
Campus Quissamã
Curso Integrado Informática
Professor: Daniel Vasconcelos
Turma: 2° ano informática
Aluno: Karen Aprigio e Rayane Gomes

Dia 27/08/2019

7ª atividade: Arduino + I2C



https://www.robocore.net/upload/tutoriais/36_img_1_M.png



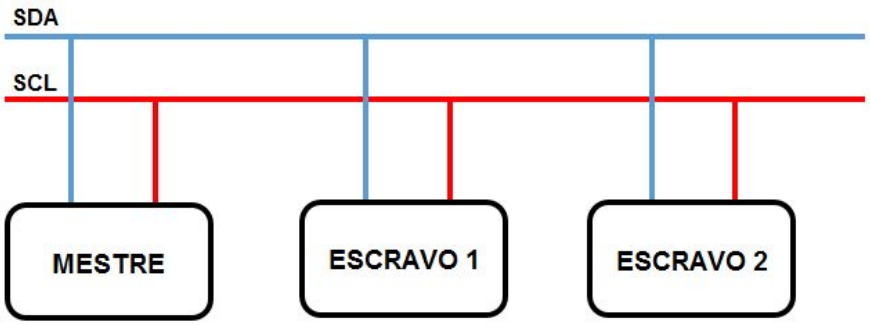
https://www.robocore.net/upload/tutoriais/36_header_H.png

O que é e para que serve:

O protocolo I²C consiste na interação entre dois ou mais dispositivos, como uma relação entre mestre e escravo. A função do “mestre” é gerenciar, requisitar e enviar informações aos “escravos”, os quais têm que responder às requisições. **(Figura 1)**

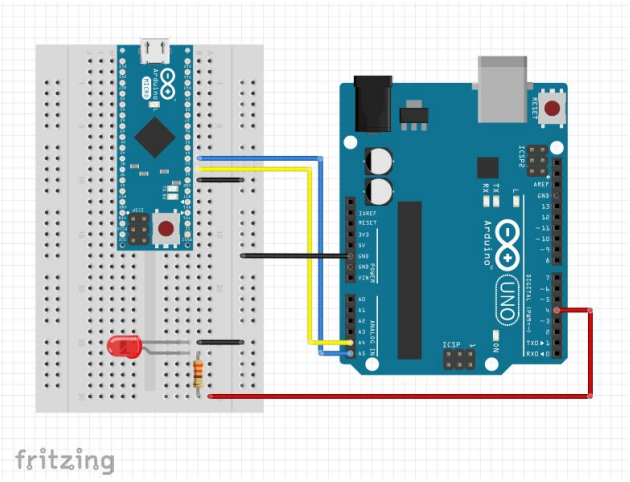
O módulo I2C (I²C) é utilizado em projetos que fazem uso de LCDs, por meio do Arduino ou algum outro microcontrolador que tenham suporte para o módulo, fazendo com que possam se comunicar com apenas duas linhas de código.

Figura 1



Exemplos:

Na montagem deste hardware devemos garantir que os pinos SDA, SCL e GND de ambos os dispositivos estejam respectivamente conectados entre si. No Arduino UNO, os pinos SDA e SCL são os pinos analógicos A4 e A5 respectivamente, ao passo que, no Arduino Micro, estes são os pinos D2 e D3. ***(Figura 2).***



Conforme apresentado no tópico anterior, o projeto a ser desenvolvido neste tutorial utiliza duas placas Arduino, portanto, teremos a presença de dois códigos para serem gravados em ambas as placas. Primeiramente, apresentamos o código a ser gravado no Arduino Micro, que por sua vez, será o mestre do barramento.

Código Mestre:

```
#include <Wire.h>

bool estado_LED;

void setup() {
  Wire.begin();
}

void loop() {
  Wire.beginTransmission(0x08);
  wire.write(estado_LED);
  Wire.endTransmission();

  estado_LED = !estadoLED;

  delay(1000);
}
```

Código Escravo:

```
#include <Wire.h>

void setup() {
  Wire.begin(0x08);
  Wire.onReceive(receiveEvent);
  pinMode(4,OUTPUT);
}

void loop() {
  delay(100);
}


void receiveEvent(int leitura) {

  bool estado = Wire.read();    // receive byte as an integer

  if (estado == 1){
    digitalWrite(4,HIGH);
  }
  else{
    digitalWrite(4,LOW);
  }
}
```

Prática:

Materiais:

- Arduino Nano
- 16 x 2 LCD
- Placa de interface LCD de três pinos
- DS 1307 I2C RTC
- 9 fios

Código:

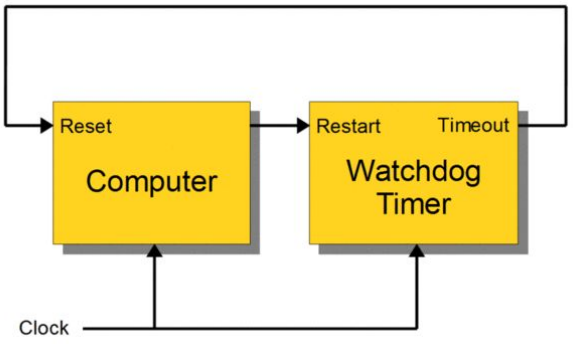
```
#include<Wire.h>

#include "RTClib.h"

RTC_DS1307 RTC;

void setup ()

{

Serial.begin(9600);

Wire.begin();

RTC.begin(); // load the time from your computer.

if (! RTC.isrunning())

{

Serial.println("RTC is NOT running!");// This will reflect the time that your
sketch was compiled

RTC.adjust(DateTime(__DATE__, __TIME__));

}

}

void loop ()

{

DateTime now = RTC.now();

Serial.print(now.month(), DEC);

Serial.print('/');
```

```
Serial.print(now.day(), DEC);

Serial.print('/');

Serial.print(now.year(), DEC);

Serial.print(' ');

Serial.print(now.hour(), DEC);

Serial.print(':');

Serial.print(now.minute(), DEC);

Serial.print(':');

Serial.print(now.second(), DEC);

Serial.println();

delay(1000);

}
```

WATCHDOGS:

É um temporizador cão-de-guarda contra travamento do programa. É uma ferramenta, tanto de software como de hardware.

Figura 1



Essa ferramenta é muito importante, se bem programado, pode deixar o dispositivo bem confiável. Trata-se de um sistema emergencial. Quando ativado, precisamos zerar o Watchdog, caso contrário, ele vai estourar e resetar o sistema. Muito utilizado para prevenir os sistema de possíveis falhas.

Fontes: https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinos/
https://www.robocore.net/tutorials/primeiros-passos-com-modulo-i2c.html
http://www.roboliv.re/conteudo/watchdog-sistema-de-reset-automatico https://pt.wikipedia.org/wiki/Watchdog_timer