I²C

Inter Integrated Circuit

https://www.prnewswire.com/news-releases/i2c-expande-recursos-para-controle-de-pagamentos-integrados-na-plataforma-300262862.html

© Prof. Engº esp Luiz Antonio Vargas Pinto
www.vargasp.com

Junho de 2022

História

- O barramento Inter Integrated Circuits ou apena I²C foi desenvolvido pela Philips Semiconductors no início de 1982 e vem sofrendo atualizações até 2014 (por enquanto).
- O objetivo é fornecer conexão de baixa velocidade da CPU a componentes periféricos.
- Devido as muitas conexões entre as MCUs e seus dispositivos em placa, frequentemente geram muitas trilhas na placa de circuito impresso - motivo da proposta.
- O I²C usa comunicação serial bidirecional composta de dois fios:
  - Serial Data Line (SDA)
  - Serial Clock Line (SCL)

© Prof. Vargas

Características

- Transferência de dados é bidirecional
  - ☒ Half duplex
- Comunicação síncrona
  - ☒ O clock é controlado quando a linha de Data é alterada
- Com 7 bits pode endereçar até 127 dispositivos
  - ☒ Ressalva na versão mais recente que tem 10 bits
- Baud rate de menos de 100 Kbps até 400 Kbps
- Em high speed (3.4 Mbps) I^2C V2.0
- 2 linhas (wire) de comunicação SDA e SCL

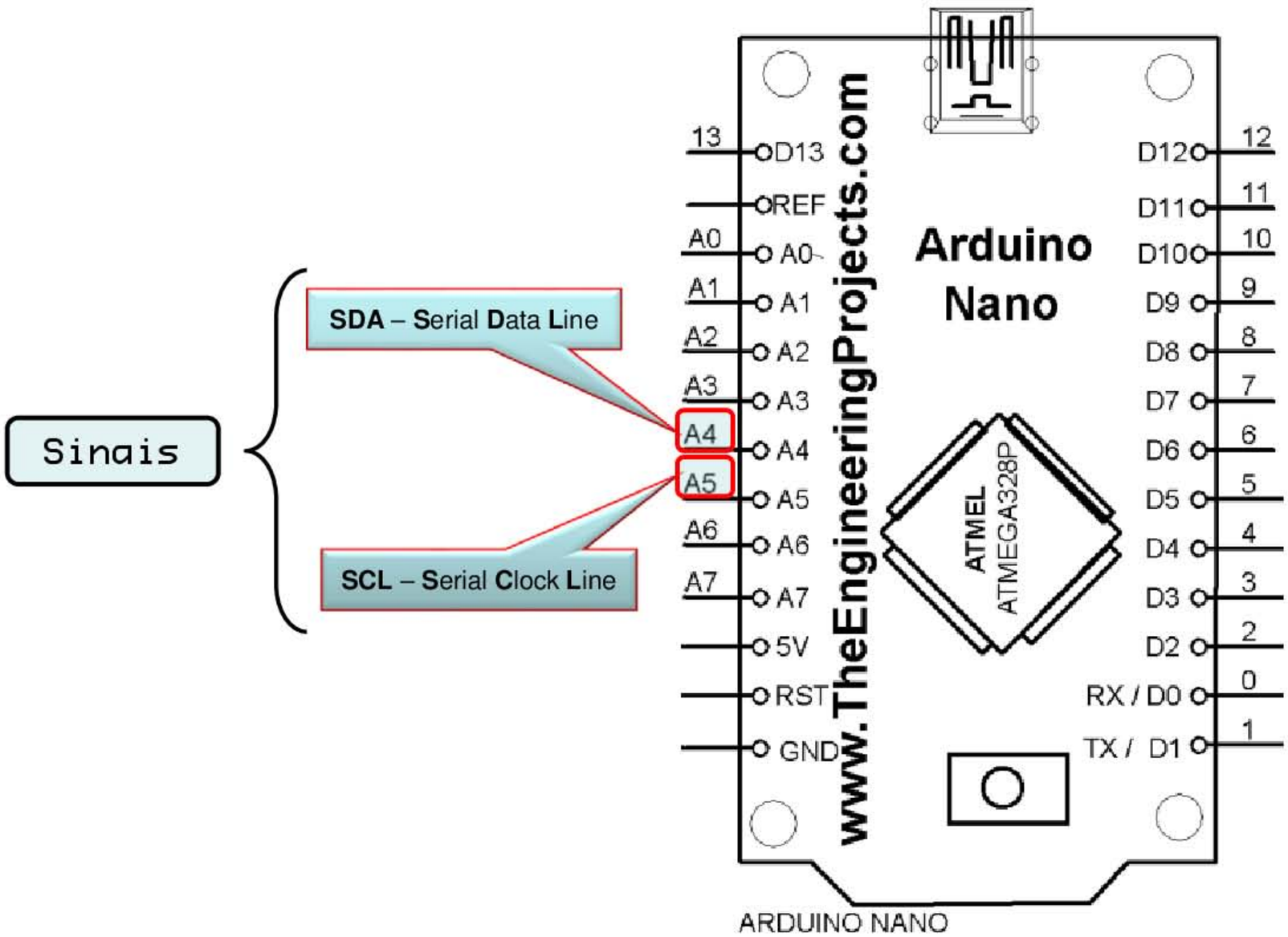
© Prof. Vargas

Funcionamento

- Por se tratar de comunicação síncrona, a sinalização entre SDA - Serial Data Line e SCL - Serial Clock Line é o que permite o trânsito regrado da informação entre Master e Slave
- O repouso da comunicação é HIGH

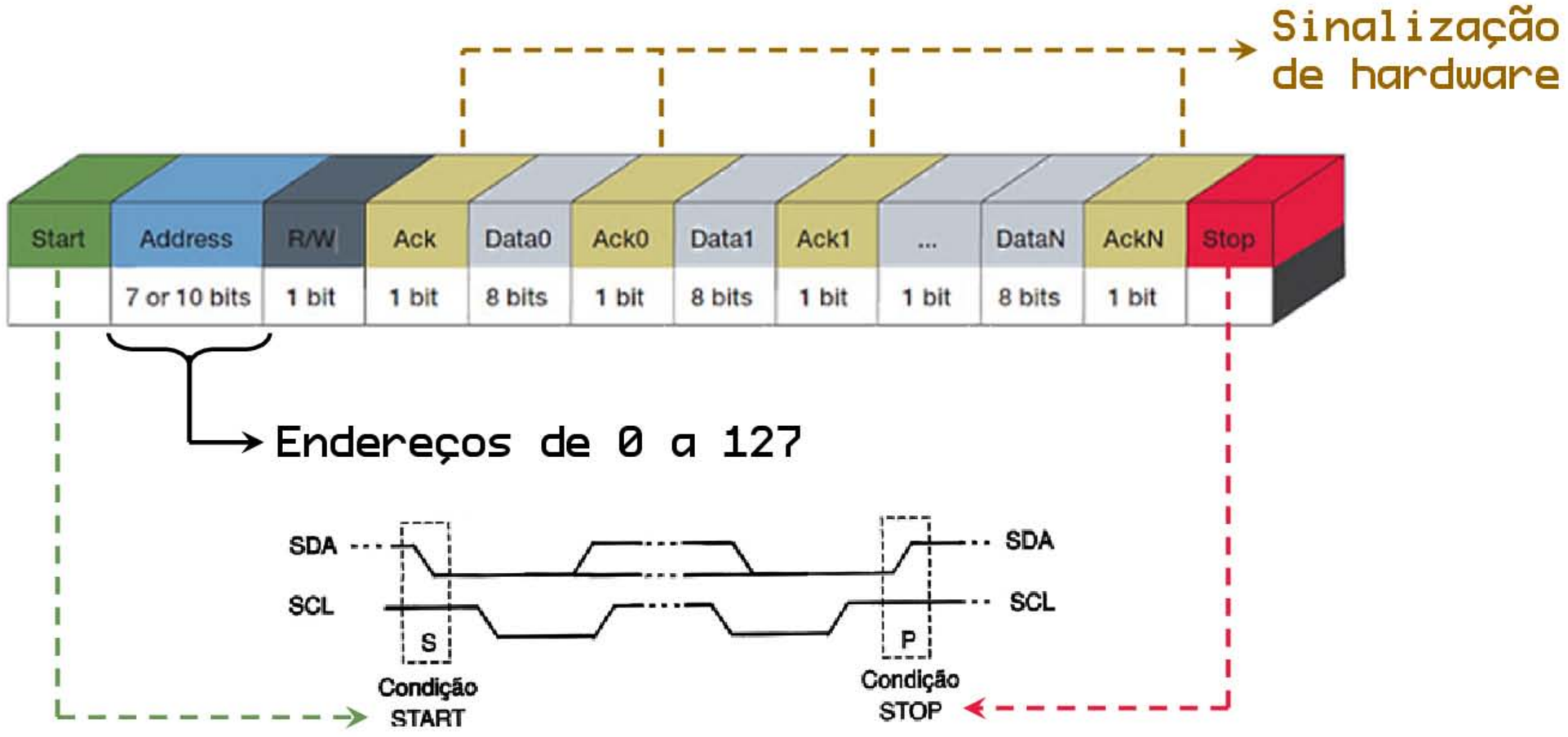
© Prof. Vargas

Características externas



© Prof. Vargas

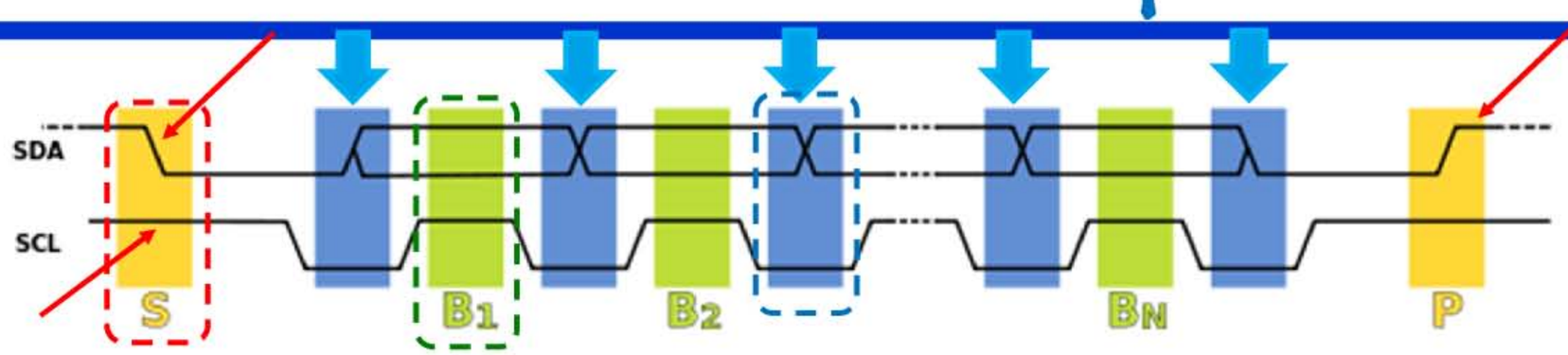
Frame



https://www.newtoncbraga.com.br/index.php/microcontroladores/143-tecnologia/12085-conheca-o-barramento-i2c-mic098.html

© Prof. Vargas

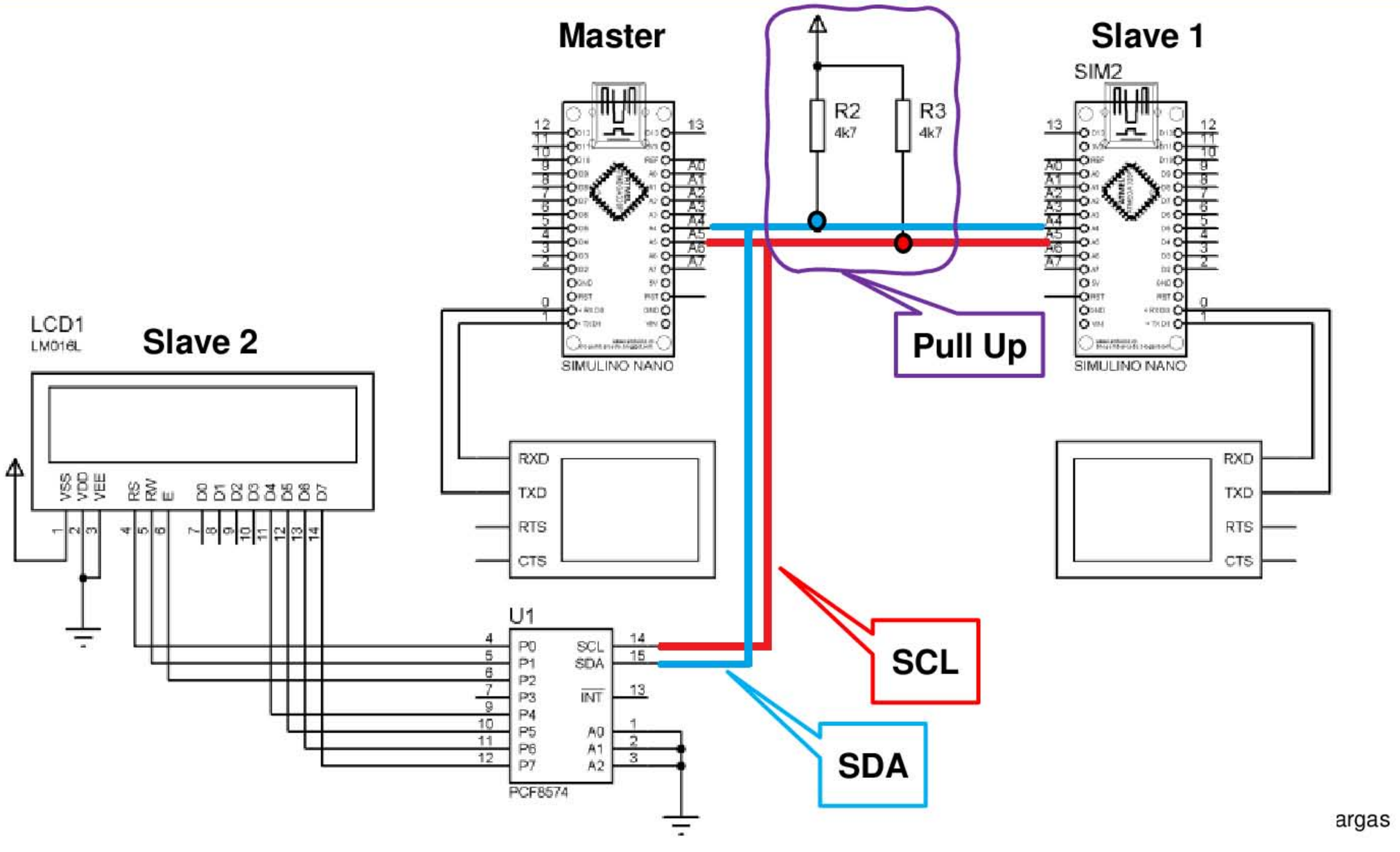
Carta de tempo



1. Bytes são transferidos tal que o MSB (Bit) vai em primeiro lugar
2. O dado transferido inicia com um *START* (S) sinalizado com SDA ↓ e SCL HIGH
3. SDA coloca o primeiro dado em nível de bit mantendo SCL LOW (durante o tempo de barra azul)
4. O dado é recebido quando SCL vai a HIGH (verde) para o 1º bit (B_1)
5. Este processo se repete, SDA ↓↑ enquanto SCL LOW, e os dados sendo lidos quando SCL HIGH (B_2, B_n)
6. Um bit *PARAR* (P) é sinalizado quando SDA ↑ enquanto o SCL HIGH

© Prof. Vargas

Circuito de teste



argas

MASTER

```
/*-----------------------------------------------------------------
  Programa : I2C_master.ino
  Data     : 06/06/2022
  Autor    : https://dronebotworkshop.com
  Editado  : Prof. Vargas
  Descrição: O Master envia um comando ao Slave solicitando o envio
             de uma mensagem pré editada "HELLO"
-----------------------------------------------------------------*/
#include <Wire.h> // Include Arduino Wire library for I2C
#include <LiquidCrystal_I2C.h>

#define BACKLIGHT_PIN 13
#define Slave_1 9    // Define Slave I2C Address
LiquidCrystal_I2C lcd(0x20,16,2);
char b;

// Define Slave answer size
#define Answersize 5

void setup()
{
 // Switch on the backlight
  pinMode (BACKLIGHT_PIN, OUTPUT);
  digitalWrite (BACKLIGHT_PIN, HIGH);

  Wire.begin();           // Inicialize I2C como Mast
  Serial.begin(9600);     // Liga monitor serial
  Serial.println("I2C Master Demonstration");
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("ARDUINO em I2C");
  delay(2000);
}
```

```
void loop()
{
  delay(450);
  Serial.println("Solicita dado do Slave");
  Wire.beginTransmission(Slave_1); // envia mensagem ao Slave_1
  Wire.write(0);
  Wire.endTransmission();
  Serial.println("Aguardando resposta do Slave");

  String response = "";
  Wire.requestFrom(Slave_1,Answersize);   // solicita ao Slave_1 5 caracteres
  while (Wire.available())  // constrói um string
  {
      b = Wire.read();
      response += b;
  }

  Serial.println(response);
  lcd.setCursor (0,1);         // posiciona LCD na linha 1
  lcd.print(response);
  Serial.println("Dado recebido !!");
}
```

© Prof. Vargas

SLAVE

```
/*-----------------------------------------------------------------
  Programa : I2C_slave.ino
  Data     : 20/01/2020
  Autor    : https://dronebotworkshop.com
  Editado  : Prof. Vargas
  Descrição: Aguarda comunicação do Master e, em resposta ao comando
             envia ao Master uma mensagem pré editada "HELLO"
-----------------------------------------------------------------*/
#include <Wire.h>
#define SLAVE_ADDR 9        // Define Slave I2C A
#define AnswerSize 5        // Define Slave answe
String answer = "Hello";    // Define string

void setup()
{
   Wire.begin(SLAVE_ADDR);          // Inicia I2C
   Wire.onRequest(requestEvent); // roda quando
   Wire.onReceive(receiveEvent); // roda quando
    Serial.begin(9600);
    Serial.println("I2C Slave Demonstration");
}
void loop()
{
    delay(250);    // Time delay in loop
}
```

Vetor de caracteres

```
void requestEvent()
{
    byte response[AnswerSize];
    for (byte i=0;i<AnswerSize;i++)
    {
       response[i] = (byte)answer.charAt(i);
    }
    Wire.write(response,sizeof(response));
    Serial.println(sizeof(response));
    for (byte i=0;i<AnswerSize;i++)
    {
      // Serial.print(" ");
      //Serial.print(response[i]);
    }
    Serial.println("Request event");
}
```

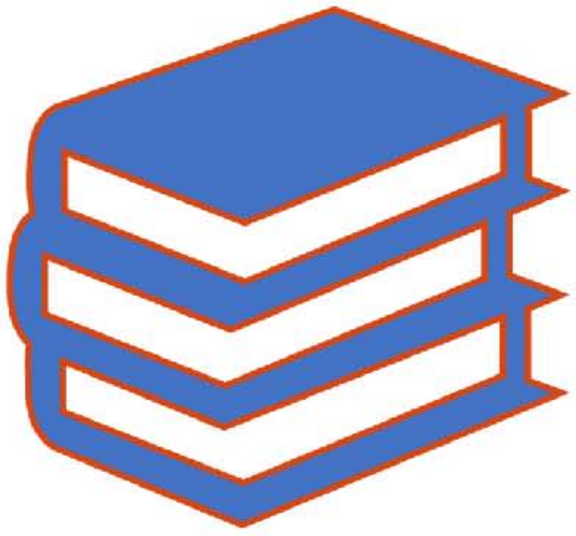
© Prof. Vargas

Ainda slave...

```
27 void receiveEvent()
28 {
29     while (0 < Wire.available())    // Read while data received
30     {
31         byte x = Wire.read();
32     }
33     Serial.println("Receive event");
34 }
```

© Prof. Vargas

Referências



- I2c good to understand, http://blmrgnn.blogspot.com/2018/05/what-is-i2c-inter-integrated-circuit.html. Acesso em junho de 2022
- https://www.newtoncbraga.com.br/index.php/microcontroladores/143-tecnologia/12085-conheca-o-barramento-i2c-mic098.html

Até a próxima !

© Prof. Vargas