

Aluno: Kallever B.F de Matos e Alice Ribeiro

Turma: 2º Ano Informática

Professor(a): Renato

Data: 27/08/2019

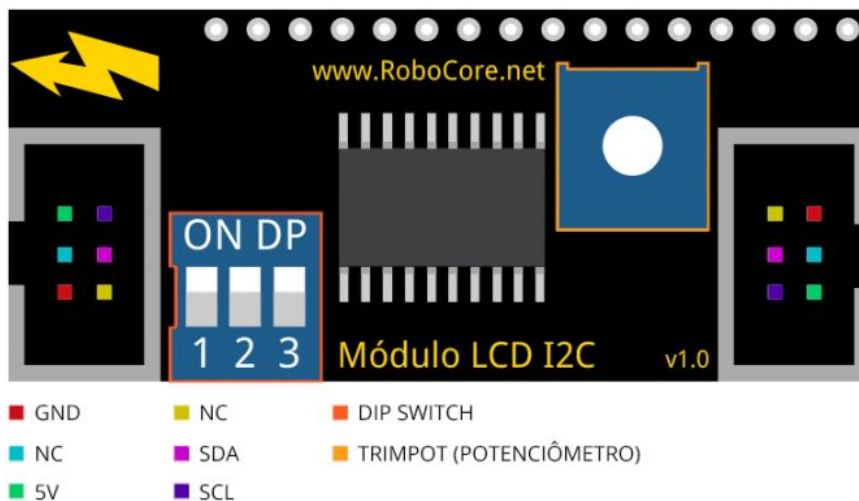
Disciplina: Microcontroladores

Arduino módulo i2C

O módulo i2C é utilizado principalmente com projetos envolvendo LCDs, podendo estar envolvido com Arduino ou outros Microcontroladores que tenha suporte ao protocolo I2C e eles irão se comunicar apenas por 2 linhas de dados (SDA e SCL).

Ele serve para manter a comunicação entre 2 placas ou LCDs, para saber a temperatura, fazer projetos mais complexos, entre outras funções.

Exemplo de modo Módulo LCD I²C:

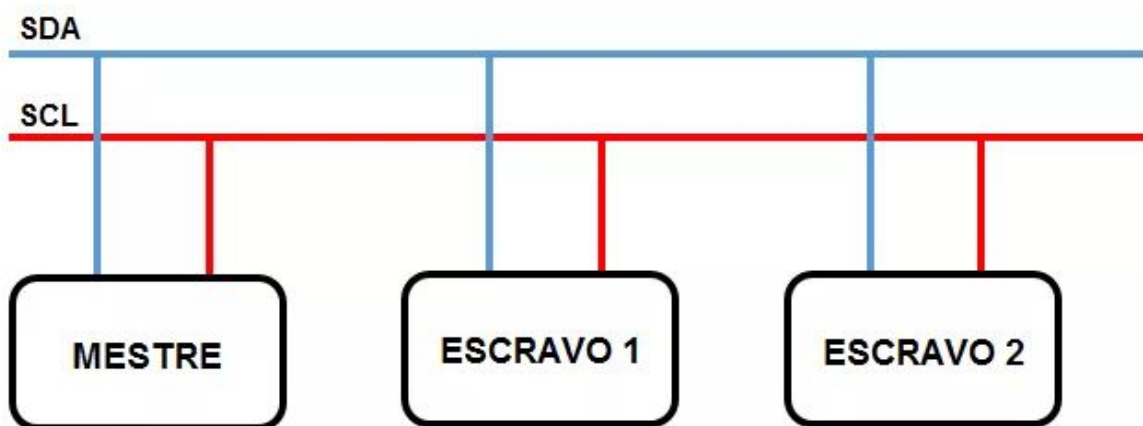


- DIP SWITCH: Define o endereço do seu dispositivo (podendo ir de 0x20 a 0x27).
- Trimpot (potenciômetro): Para quantificar a luminosidade do Backlight do seu Display LCD
- Conectores Latch (ou IDC): Utilizando Jumpers M/F ou F/F você conectará o módulo tanto com seu Mestre (no caso desse tutorial uma BlackBoard) quanto a seus Slaves.

O protocolo de comunicação I2C:

“O modo de funcionamento do protocolo I2C é baseado na interação entre elementos seguindo a hierarquia mestre/escravo, ou seja, quando temos vários dispositivos se comunicando segundo esta premissa, pelo menos um destes deve atuar como mestre e os demais serão escravos. A função do mestre consiste em realizar a coordenação de toda a comunicação, pois, ele tem a capacidade de enviar e requisitar informações aos escravos existentes na estrutura de comunicação, os quais, devem responder às requisições citadas.

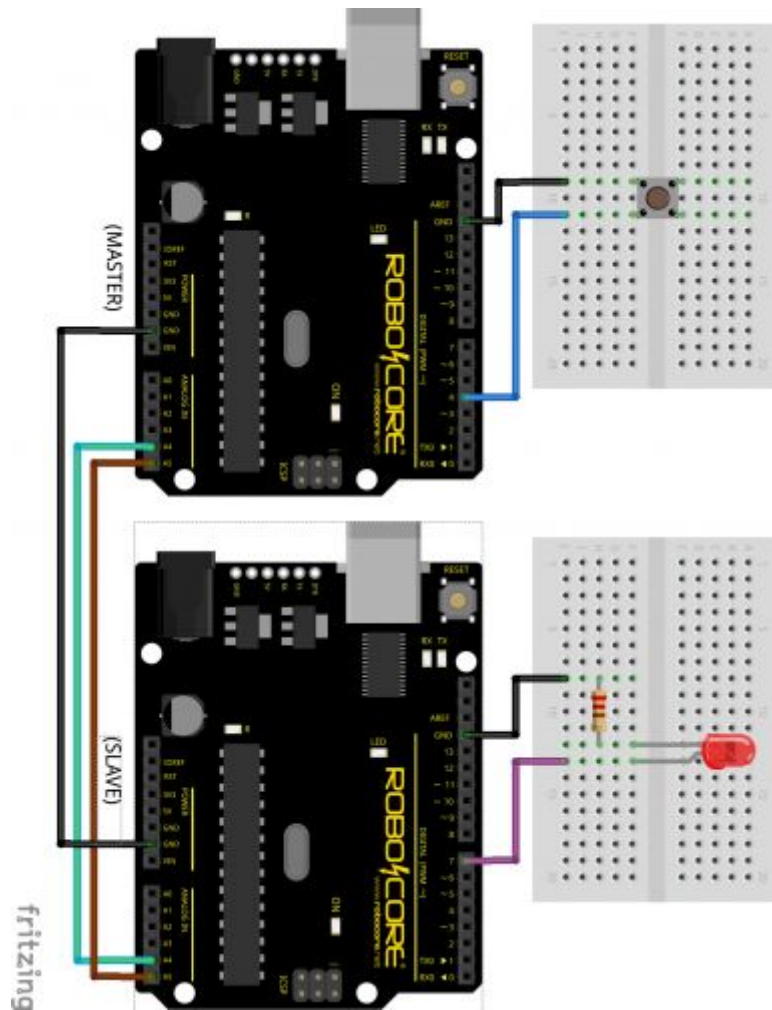
A estrutura na qual o protocolo I2C atua é uma estrutura de barramento, que por sua vez, consiste em um arranjo em que todos os elementos encontram-se conectados a um ramal principal.”



“Na figura acima pode-se ver um modelo simplificado da estrutura de barramento citada anteriormente. Note que, neste arranjo, o barramento de comunicação I2C pode ser dividido em dois barramentos. Um deles é denominado SDA (Serial Data), o qual, é responsável pela troca de dados entre os dispositivos do arranjo e o outro barramento, denominado SCL (Serial Clock), possui a função de sincronizar os dispositivos e garantir a confiabilidade do sistema.”

Comunicação entre 2 arduinos:

Você pode conectar ao barramento diversos Arduinos. Tal como os módulos que suportam I2C, o Arduino também precisa ter um endereço cujo o qual você definirá via software.



Utilizaremos a biblioteca Wire para realizar esta tarefa.

O protocolo serial síncrono I2C, também conhecido como TWI, utiliza de dois fios (SDA: Dados; SCL: Clock) para realizar uma comunicação half duplex, ou seja, é possível transmitir e receber informações, mas não ao mesmo tempo, apenas um sentido por vez. Forma-se um barramento endereçável onde cada componente na rede possui um endereço para ser identificado para que a informação possa ser designada para o destinatário correto.

Nessa primeira parte do tutorial a placa Master estará conectada ao botão e, ao pressionar este botão, o estado do led conectado à placa Slave será alterado. A placa Master trabalhará como o emissor e a placa Slave como o receptor.

Exemplo de código:

/*

Exemplo de como comunicar Arduinos utilizando o protocolo I2C.

Altera o estado do LED conectado a placa Slave quando o botão ligado a placa Master for pressionado.

Na placa Master foi utilizado um botão conectado ao pino 4, com o resistor de pullup interno da placa acionado e realizou-se um debounce em software. Na placa Slave foi utilizado um LED conectado ao pino 7.

Referências:

Exemplo Debounce - <https://www.arduino.cc/en/Tutorial/Debounce>

Tutorial Master Writer - <https://www.arduino.cc/en/Tutorial/MasterWriter>

*/

#include "Wire.h"

#define buttonPin 4 // numero do pino onde o botão está conectado

```
// endereco do modulo slave que pode ser um valor de 0 a 255

#define slaveAdress 0x08


boolean buttonState;          // estado atual do botao

boolean lastButtonState = LOW; // valor da ultima leitura do botao

boolean ledState = HIGH;      // estado atual do LED


// as variaveis a seguir sao do tipo long por conta que o tempo, medido
// em milissegundos alcancara rapidamente um numero grande demais para
// armazenar em uma variavel do tipo int

unsigned long lastDebounceTime = 0; // tempo da ultima modificacao do estado do LED


// tempo de debounce; aumentar se o LED oscilar; espera-se que o LED acenda
// apenas se o botao for pressionado por mais de 50ms

unsigned long debounceDelay = 50;


void setup() {

    Wire.begin(); // ingressa ao barramento I2C


    // configura o pino do botao como entrada com resistor de pullup interno

    pinMode(buttonPin, INPUT_PULLUP);
```

```
}
```

```
void loop() {
```

```
    // le o estado do botao e salva em uma variavel local
```

```
    int reading = digitalRead(buttonPin);
```

```
    // verifica se voce apenas apertou o botao (i.e. se a entrada foi de LOW
```

```
    // to HIGH), e se ja esperou tempo suficiente para ignorar qualquer ruido
```

```
    // se a entrada foi alterada devido ao ruido ou botao ter sido pressionado:
```

```
    if (reading != lastButtonState) {
```

```
        // reseta o tempo do debounce
```

```
        lastDebounceTime = millis();
```

```
    }
```

```
    if ((millis() - lastDebounceTime) > debounceDelay) {
```

```
        // qualquer que seja a leitura atual, ela se manteve por um tempo maior
```

```
        // que o nosso debounce delay, então atualizemos o estado atual:
```

```
        // se o estado do botao foi alterado:
```

```
        if (reading != buttonState) {
```

```

buttonState = reading;

// apenas altera o estado do LED se o novo estado do botao e HIGH
if (buttonState == HIGH) {

    ledState = !ledState;

    // incia a transmissao para o endereco 0x08 (slaveAdress)

    Wire.beginTransmission(slaveAdress);

    Wire.write(ledState); // envia um byte contendo o estado do LED

    Wire.endTransmission(); // encerra a transmissao

}

}

}

// salva a leitura. No proximo laco este sera o ultimo

// estado do botao (lastButtonState)

lastButtonState = reading;

}

```

Fontes:

<https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinos/>,

<https://www.robocore.net/tutoriais/comunicacao-entre-arduinos-i2c-parte1>

<https://www.robocore.net/tutoriais/primeiros-passos-com-modulo-i2c.html>

