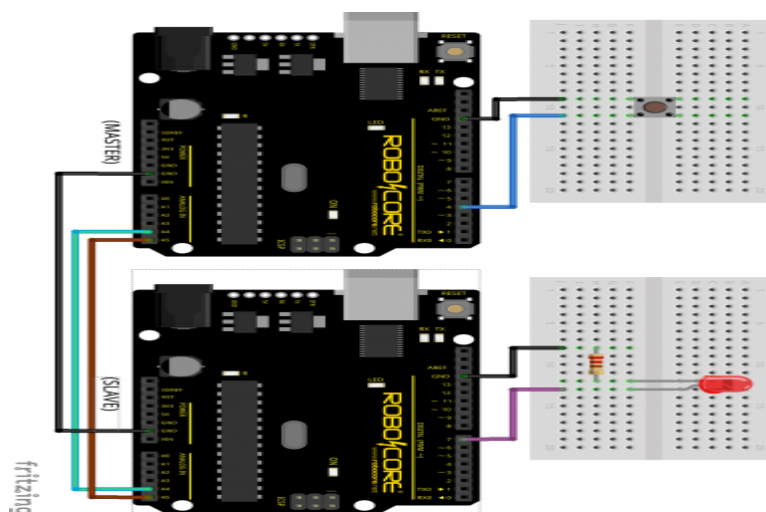


**Campus Quissamã.**  
**Professor: Renato.**  
**Turma: 2º Ano Integrado em Informática .**  
**Aluno:Álvaro Pessanha e Igor Costa.**  
**Assunto: Arduino I2C.**  
**Quissamã, 2019.**

O protocolo serial síncrono I2C, também conhecido como TWI, utiliza de dois fios 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 encaminhado, em que cada componente na rede possui um endereço para ser identificado para que a informação possa ser dirigida para o destinatário correto.

Neste exemplo, a placa Master trabalhará como o emissor e a placa Slave como o receptor. Sendo necessário os componentes:

- BlackBoard V1.0
- LED
- Resistor
- PushButton
- Jumpers M/M



[https://www.robocore.net/upload/tutoriais/59\\_fritz\\_H.png?354](https://www.robocore.net/upload/tutoriais/59_fritz_H.png?354)

Após ter montado todo o circuito, é necessário programar cada placa com seu código. Neste momento, é importante verificar se está gravando a BlackBoard correta, por isso é melhor conectar e programar uma placa por vez.

**Código para a BlackBoard Master conectada ao botão:**

```
#include "Wire.h"
```

```
#define buttonPin 4 // numero do pino onde o botao esta conectado
```

```

#define slaveAddress 0x08
boolean buttonState;           // estado atual do botao
boolean lastButtonState = LOW; // valor da ultima leitura do botao
boolean ledState = HIGH;       // estado atual do LED
unsigned long lastDebounceTime = 0; // tempo da ultima modificacao do estado do LED
unsigned long debounceDelay = 50;

void setup() {
  Wire.begin(); // ingressa ao barramento I2C
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  int reading = digitalRead(buttonPin);
  if (reading != lastButtonState) {
    // reseta o tempo do debounce
    lastDebounceTime = millis();
  }
  if ((millis() - lastDebounceTime) > debounceDelay) {
    if (reading != buttonState) {
      buttonState = reading;
      if (buttonState == HIGH) {
        ledState = !ledState;
        Wire.beginTransaction(slaveAddress);
        Wire.write(ledState); // envia um byte contendo o estado do LED
        Wire.endTransmission(); // encerra a transmissao
      }
    }
  }
  lastButtonState = reading;
}

```

### **Código para a BlackBoard Slave conectada ao LED:**

```

#include "Wire.h"
#define ledPin 7 // numero do pino onde o LED esta conectado
#define myAdress 0x08
void setup() {
  Wire.begin(myAdress);
  Wire.onReceive(receiveEvent);
  pinMode(ledPin, OUTPUT); // configura o pino do LED como saida
}

void loop() {
}

void receiveEvent(int howMany) {
  if (Wire.available()) {
    char received = Wire.read();
    if (received == 0) {
      digitalWrite(ledPin, LOW);
    }
  }
}

```

```

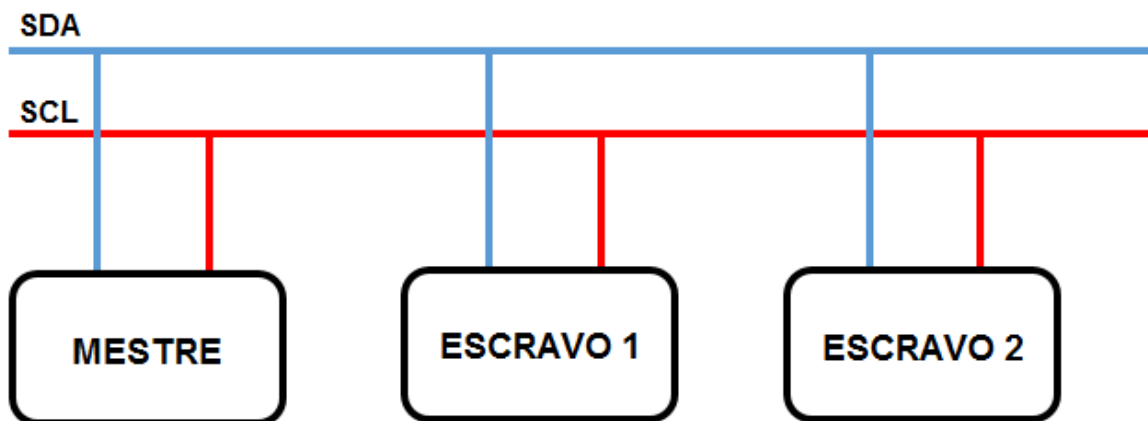
}
    if (received == 1) {
        digitalWrite(ledPin, HIGH);
    }
}
}
}

```

O Módulo Adaptador I2C foi desenvolvido com o objetivo de simplificar a conexão de display LCD ao microcontrolador. Para uma conexão de 4 bits entre o display LCD e o microcontrolador é necessário ao menos 6 cabos, ou seja, se o microcontrolador tiver poucas portas digitais, isso poderá ser um problema. Com o Módulo Adaptador I2C para Display LCD é necessário apenas 2 cabos de comunicação entre o display LCD e o microcontrolador.

O modo de funcionamento do protocolo I2C é baseado na interação entre elementos seguindo a hierarquia mestre/escravo, ou seja, quando se tem vários dispositivos se comunicando segundo esta sequência, 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 ponto principal.



<https://i1.wp.com/portal.vidadesilicio.com.br/wp-content/uploads/2017/11/barramentoi2c.png?w=655&ssl=1>

#### **Vantagens:**

- Baixo consumo.
- Uso de vários componentes.

#### **Desvantagens:**

- Chance de Travamento.
- Endereço limitado a 7 bits, ou seja possibilitando o máximo de 127 dispositivos conectados ao barramento.

**Fontes:**

- <https://www.robocore.net/tutoriais/comunicacao-entre-arduinos-i2c-parte1>
- <http://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-adaptador-i2c-para-display-lcd-16x2-20x4/>
- <https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinos/>