

Dupla: Joyce Kelly De Lima e Vitória Conceição
Turma: 2º Ano de Informática
Artigo: I2O
Data: 27/08/2019

O que é?

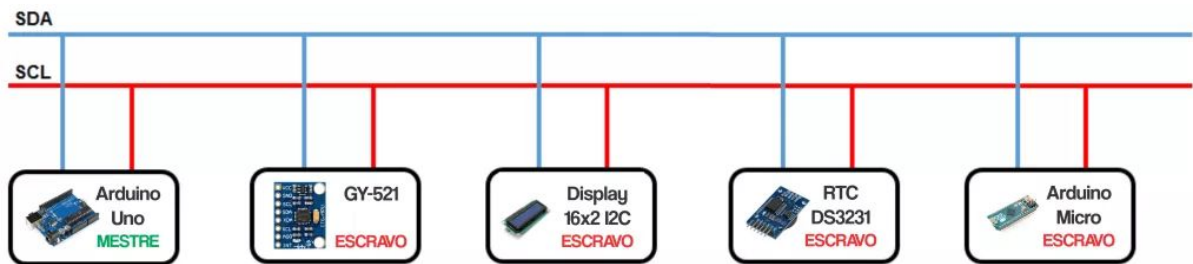
Seu nome significa Circuito Inter-integrado. É um novo barramento, que foi criado em 1992 pela empresa Holandesa (Philips), com objetivo de facilitar a integração de circuitos de caráter final de aplicação como por exemplo sensores e conversores, com um sistema de controle, de modo que eles possam trabalhar com seus sinais de maneira direta. Utilizado conectar periféricos de baixa velocidade a uma placa mãe a um sistema embarcado ou a um telefone celular.

A ideia básica do uso do barramento I2C é permitir a fácil integração em componentes de interface, que possam trabalhar diretamente com seus sinais simplificando assim a comunicação com outros componentes da mesma família

Pra que serve?

O modo como o I2C funciona é baseado na interação entre elementos seguindo a organização mestre/escravo, ou seja, quando se tem vários dispositivos se comunicando segundo esta premissa e pelo menos um desses deve atuar como mestre e os demais serão escravos. O mestre atua na função de realizar a organização de toda a comunicação, pois, ele tem a capacidade de enviar e solicitar informações aos escravos existentes na estrutura de comunicação, os quais, devem responder às solicitações citadas.

O I2C possibilita a utilização de uma grande quantidade de componentes padronizados, os quais podem realizar diversas funções, além de possibilitar a troca eficaz de informações entre eles.



Exemplos

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.

```
#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);
}
```

Em seguida, apresentamos o código a ser gravado no Arduino UNO, que em nosso projeto, se comportará como escravo.

```
#include <Wire.h>

void setup() {
  Wire.begin(0x08);
```

```

Wire.onReceive(receiveEvent);
pinMode(4,OUTPUT);
}

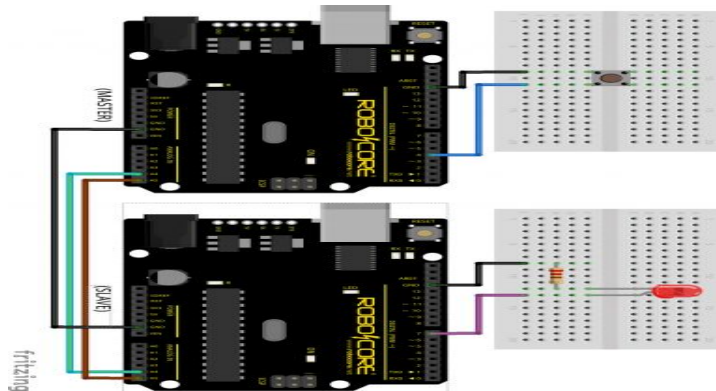
void loop() {
  delay(100);
}

void receiveEvent(int leitura) {
  bool estado = Wire.read();  // receive byte as an integer
  if (estado == 1){
    digitalWrite(4,HIGH);
  }
  else{
    digitalWrite(4,LOW);
  }
}

```

Prática

Confira na imagem como realizar as conexões de cada BlackBoard.



Devemos montar todo o circuito, é necessário programar cada placa com seu código. Devemos verificar se está gravando a BlackBoard correta, por isso e conecte e programe uma placa por vez. Desta forma você não irá se confundir com as portas COM.

Código para a BlackBoard Master (conectada ao botão):

```

#include "Wire.h"

#define buttonPin 4

```

```

    if (reading != lastButtonState)
        lastDebounceTime = millis();
    }

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

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

            if (buttonState == HIGH) {
                ledState = !ledState;

                Wire.beginTransmission(slaveAdress);
                Wire.write(ledState); // envia um byte contendo o estado do LED
                Wire.endTransmission(); // encerra a transmissao
            }
        }
    }

    lastButtonState = reading;
}
#define slaveAdress 0x08

boolean buttonState;
boolean lastButtonState = LOW;
boolean ledState = HIGH;

unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;

void setup() {
    Wire.begin();
    pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
    int reading = digitalRead(buttonPin);

```

Código para a BlackBoard Slave (conectada ao LED):

```

#include "Wire.h"
#define ledPin 7
#define myAdress 0x08

void setup() {

```

```

Wire.begin(myAdress);
Wire.onReceive(receiveEvent);
pinMode(ledPin, OUTPUT);
}

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);
        }
    }
}
}

```

Agora basta alimentar ambas as placas que tudo deve funcionar perfeitamente. Aperte o botão da BlackBoard Master e o LED conectado a placa Slave deve acender. Se você apertar novamente o mesmo deve apagar.

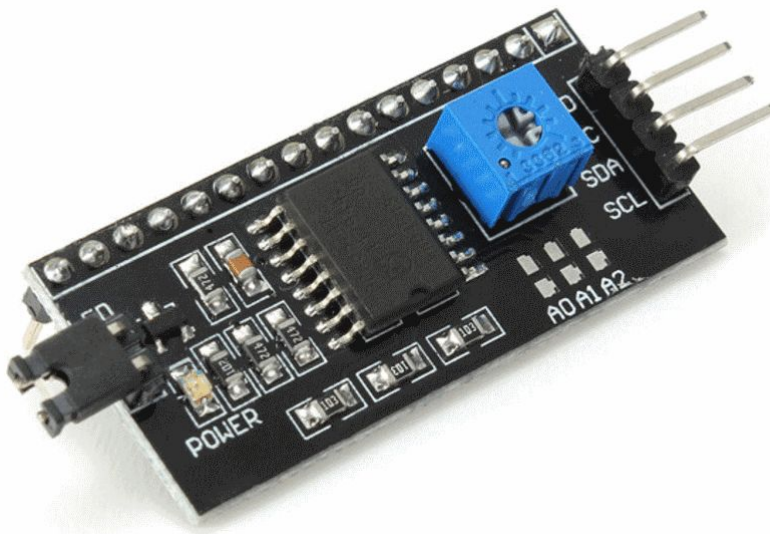
Até este ponto já aprendemos como fazer nossa placa Mestre enviar informações para o Slave

Aplicações

Para aplicar o protocolo I2C em sistemas microcontrolados, foram utilizados os microcontroladores AVR, fabricados pela ATMEL, com isso foi explorado diversos dispositivos I2C facilmente encontrados no mercado. Em cada caso, foi possível saber os princípios básicos de operação de tais dispositivos bem como a construção de pequenos protótipos que permitiram explorar a capacidade de cada um deles. Para o desenvolvimento do firmware foi utilizado a plataforma de desenvolvimento BASCOM.

Como utilizar o módulo I2C com display LCD

Necessitamos pelo menos 6 fios para conexão. Em placas com um número menor de portas, como o Arduino Uno, isso significa sacrificar algumas portas que poderiam ser utilizadas para ligação de outros componentes, como sensores ou motores. Um módulo que pode ser utilizado para contornar esse problema é o módulo I2C para display LCD com CI PCF8574 (datasheet). Com isso você poderá controlar um display LCD, seja ele 16×2 ou 20×4, utilizando apenas dois pinos do Arduino: o pino analógico 4 (SDA) e o pino analógico 5 (SCL), que formam a interface de comunicação I2C



Vídeos no youtube

<https://www.youtube.com/watch?v=jWE0R7fG4k4>

Fonte:

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

<https://www.newtoncbraga.com.br/index.php/microcontrolador/143-tecnologia/12085-conhecendo-o-barramento-i2c-mic098>

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

<http://microcontrolandos.blogspot.com/2012/12/comunicacao-i2c.html>

<http://www.meccomeletronica.com/site/data/uploads/i2c-protocolo-de-comunicacao.pdf>