

INSTITUTO FEDERAL

Santa Catarina

Câmpus Tubarão

Display LCD, Comunicação I2C e Sensores

Curso Superior de Tecnologia em Sistemas Embarcados

Professor: Fernando Silvano Gonçalves

fernando.goncalves@ifsc.edu.br

Maio de 2023

Cronograma

| Encontro | Data | Nº Aulas | Conteúdo |
|----------|---------|----------|---|
| 1 | 29-fev. | 04 | Recepção e Apresentação do Unidade / Apresentação do Plano de Ensino / Avaliação Diagnóstica / Introdução a sistemas embarcados |
| 2 | 02-mar. | 04 | Conceitos e Características e Aplicações de Sistemas Embarcados / Histórico de Sistemas Embarcados / Práticas com Arduino |
| 3 | 07-mar. | 04 | Microcontroladores, Microprocessadores / Periféricos / Introdução ao Arduino / Introdução ao C |
| 4 | 14-mar. | 04 | Introdução à Linguagens de Programação / Comunicação Serial |
| 5 | 21-mar. | 04 | Entrada de Dados via Serial |
| 6 | 28-mar. | 04 | Linguagem C para Arduino |
| 7 | 04-abr. | 04 | Variáveis e Operadores |
| 8 | 11-abr. | 04 | Estruturas Condicionais |
| 9 | 18-abr. | 04 | Práticas com Arduino e entradas e saídas analógicas e digitais |
| 10 | 25-abr. | 04 | Práticas com Arduino e entradas e saídas analógicas e digitais |

Cronograma

| Encontro | Data | Nº Aulas | Conteúdo |
|----------|---------|----------|---|
| 11 | 02-mai. | 04 | Avaliação 01 |
| 12 | 09-mai. | 04 | Avaliação 01 |
| 13 | 16-mai. | 04 | Correção da Avaliação / Introdução a estruturas de repetição |
| 14 | 18-mai. | 04 | Estruturas de Repetição / Comunicação I2C / Display 16x2 |
| 15 | | 04 | Timers, Interrupções / Sensores (LDR) |
| 16 | | 04 | Comunicação I2C / Display LCD / Sensores (Ultrassônico, DHT-22) |
| 17 | | 04 | PWM, Ponte H, Atuadores (Servomotor, Motor DC, Relé) |
| 18 | | 04 | Comunicação Bluetooth / Práticas com Sensores e atuadores |
| 19 | | 04 | Práticas com Sensores e atuadores |
| 20 | | 04 | Conselho de Classe / Atividades de Recuperação e Encerramento da UC |
| | | 80 | |

Pauta

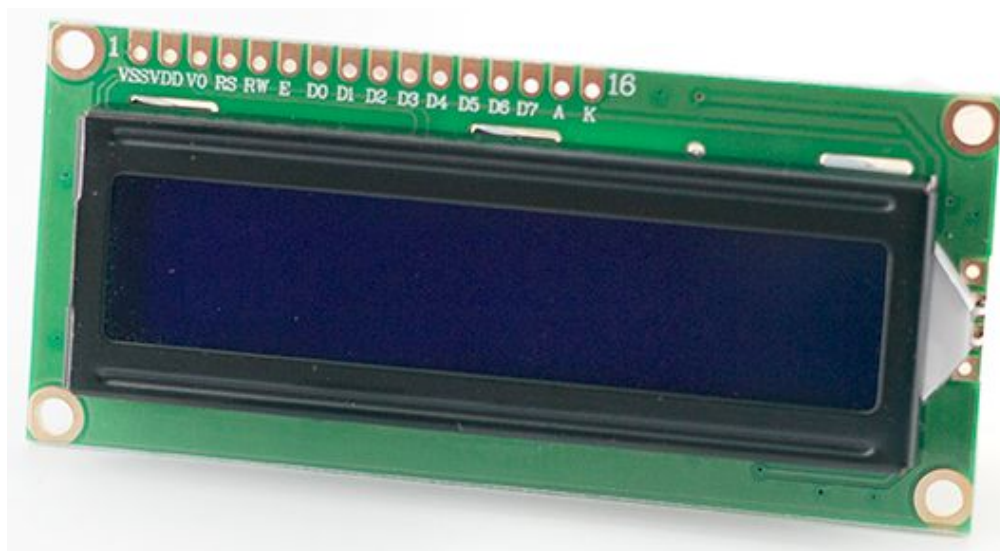
- Display 16x2;
- Comunicação I2C;
- Sensor ultrassônico;
- Sensor de Umidade e Temperatura (DHT-22);

Display

Display

Display LCD

- ❑ Display possui 16 colunas e 2 linhas para exibição das mensagens;
- ❑ Para conexão são 16 pinos, dos quais 12 são utilizados para uma conexão básica.

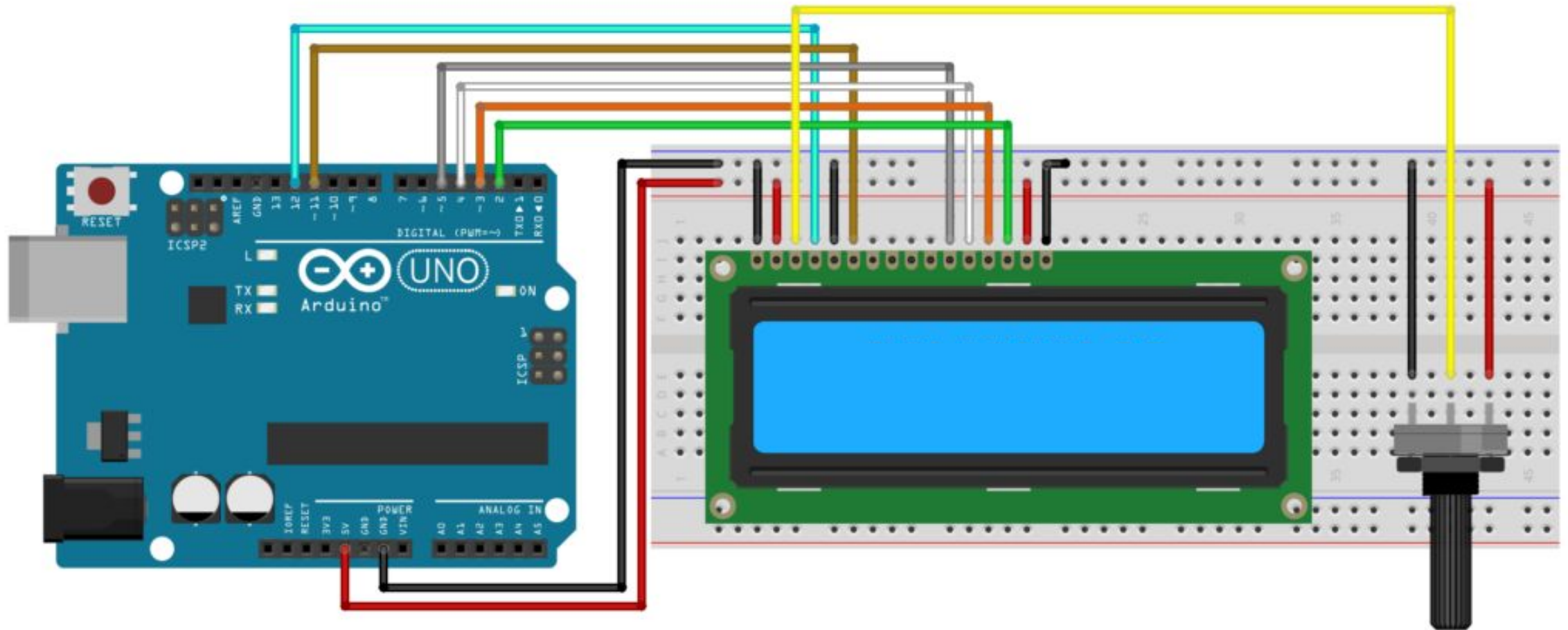


Display LCD

Conexão LCD 16x2 – HD44780

| Pino LCD | Função | Ligação |
|----------|--------|----------------------------|
| 1 | Vss | GND |
| 2 | Vdd | Vcc 5v |
| 3 | V0 | Pino Central Potenciômetro |
| 4 | RS | Pino 12 Arduino |
| 5 | RW | GND |
| 6 | E | Pino 11 Arduino |
| 7 | D0 | N/C |
| 8 | D1 | N/C |
| 9 | D2 | N/C |
| 10 | D3 | N/C |
| 11 | D4 | Pino 5 Arduino |
| 12 | D5 | Pino 4 Arduino |
| 13 | D6 | Pino 3 Arduino |
| 14 | D7 | Pino 2 Arduino |
| 15 | A | Vcc 5v |
| 16 | K | GND |

Display LCD



Controlando Display LCD

- ❑ O controle desse display será realizado utilizando a biblioteca **LiquidCrystal**, já embutida na IDE do Arduino.
- ❑ No início do programa definimos os pinos que serão utilizados pelo displays, nesse formato:

LiquidCrystal lcd(<pino RS>, <pino enable>, <pino D4>, <pino D5>, <pino D6>, <pino D7>)

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(8,9,4,5,6,7);
```

```
void setup(){  
  lcd.begin(16,2);  
  lcd.clear();  
  lcd.setCursor(0,0);  
  lcd.print(" FIC ARDUINO");  
  lcd.setCursor(0,1);  
  lcd.print(" IFSC – TUB");  
}
```

```
void loop(){  
}
```

Prática com Display LCD

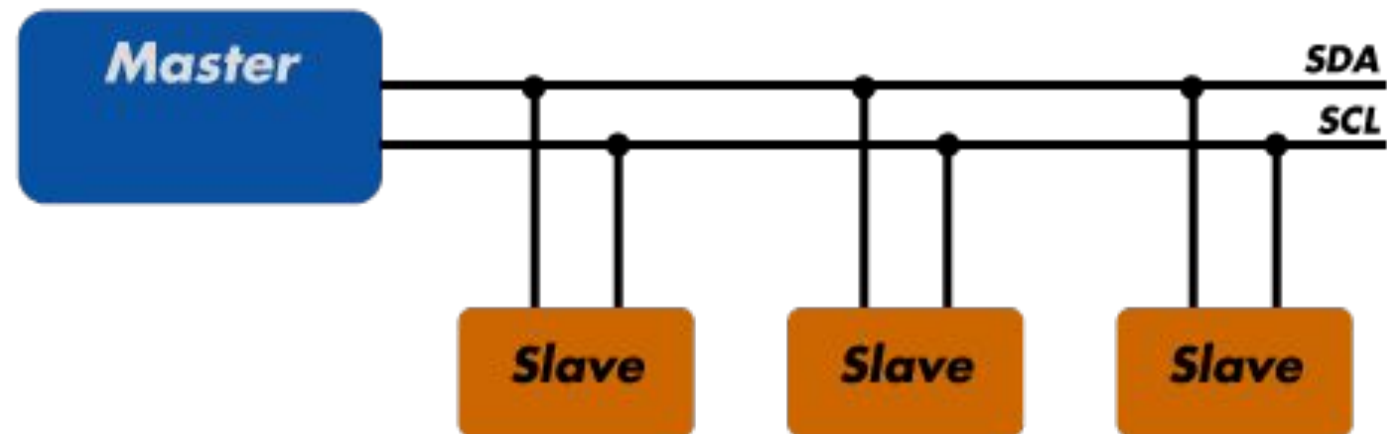
1. Com base no exemplo anterior, crie um software com as seguintes características:
 - a) Escreva na linha superior o seu nome;
 - b) Na linha inferior a cada 2 segundos alterne entre as frases “IFSC-TUBARAO” e “PROG SIST EMB”;

Protocolo I2C Arduino

LCD I2C Arduino

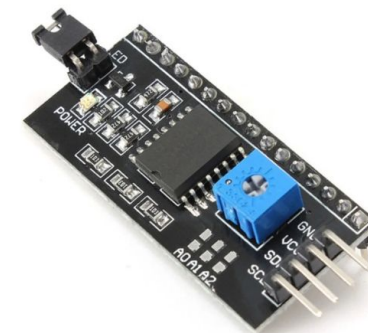
- ❑ Criado pela empresa Phillips por volta dos anos 80;
- ❑ Este protocolo foi criado com o propósito de padronizar a comunicação entre diferentes dispositivos.
- ❑ Ele necessita apenas de dois pinos (SDA – para dados, SCL – nosso clock) para poder conectar até 112 dispositivos para barramentos de 7 bits.
- ❑ Foi criado para distâncias curtas, e com uma velocidade média de operação (por volta de 400KHz).
- ❑ Em virtude dessa sua facilidade de utilização, diversos componentes e módulos foram desenvolvidos voltados para este tipo de interface.
- ❑ Você pode encontrar Display LCD, RTCs entre outros.

LCD I2C Arduino



LCD I2C Arduino

LCD I2C Arduino



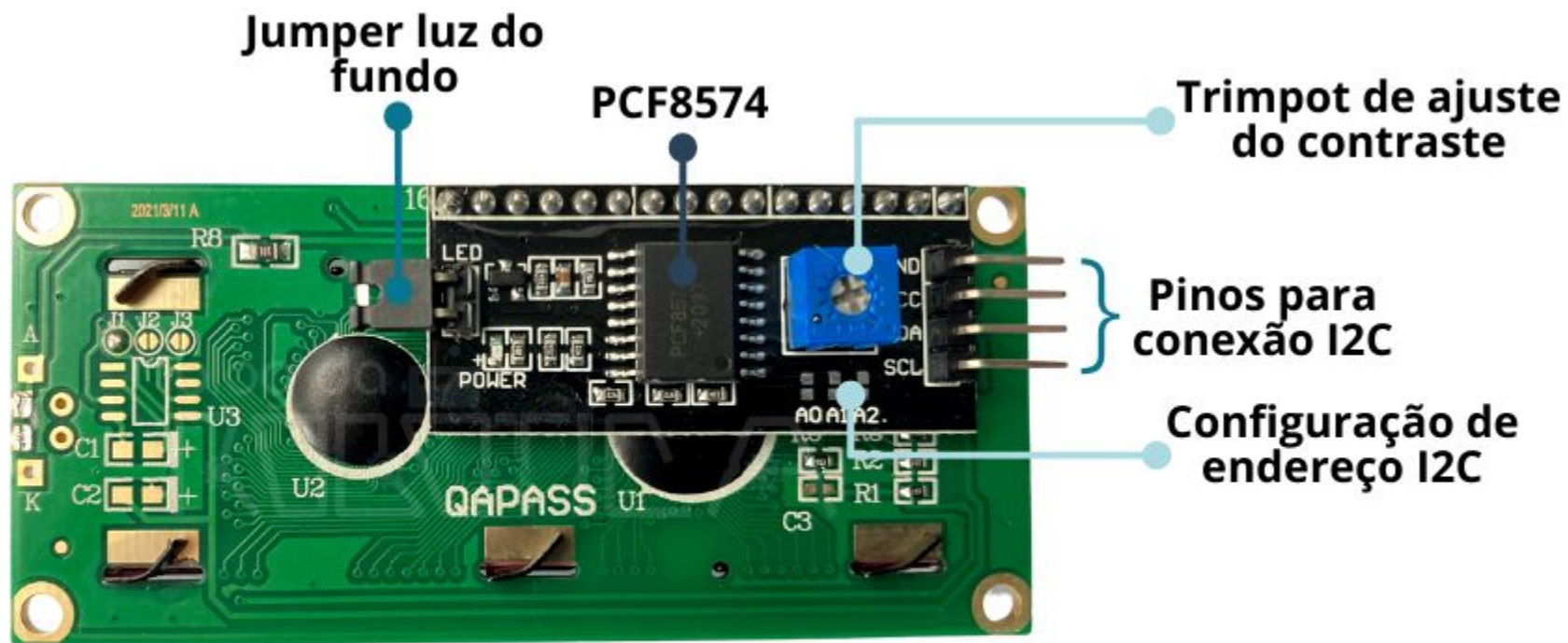
- ❑ O display LCD requer muitas conexões para seu funcionamento;
- ❑ No entanto, isso pode ser contornado por meio do uso do módulo I2C;
- ❑ O módulo I2C para display LCD 16×2 ou 20×4 tem como principal função simplificar as conexões e facilitar a utilização de displays LCD ao microcontrolador.
- ❑ O módulo I2C deve ser acoplado ao display LCD por meio da conexão dos seus 16 pinos.



LCD I2C Arduino

- ❑ No centro do módulo I2C, está o chip PCF8574, um expensor de 8 bits que converte os dados I2C nos dados paralelos exigidos pelo display LCD.
- ❑ Na parte lateral, o módulo I2C conta com quatro pinos para conexão com o Arduino.
- ❑ Estes quatro pinos são: GND, VCC, SDA e SCL.
 - O pino SDA (do inglês, Serial Data) é o responsável pela transferência e recebimento de dados.
 - Por sua vez, o pino SCL (do inglês, Serial Clock) é utilizado para temporização.
- ❑ Ele também conta com dois pinos para controle da luz de fundo do visor do display, que se encontram conectados por um jumper. Assim, a luz de fundo estará sempre ligada. Você pode remover este jumper para desligar a luz de fundo do display.
- ❑ Além disto, a placa conta com um pequeno trimpot para fazer ajustes finos no contraste do display e um barramento para configuração do endereço I2C.

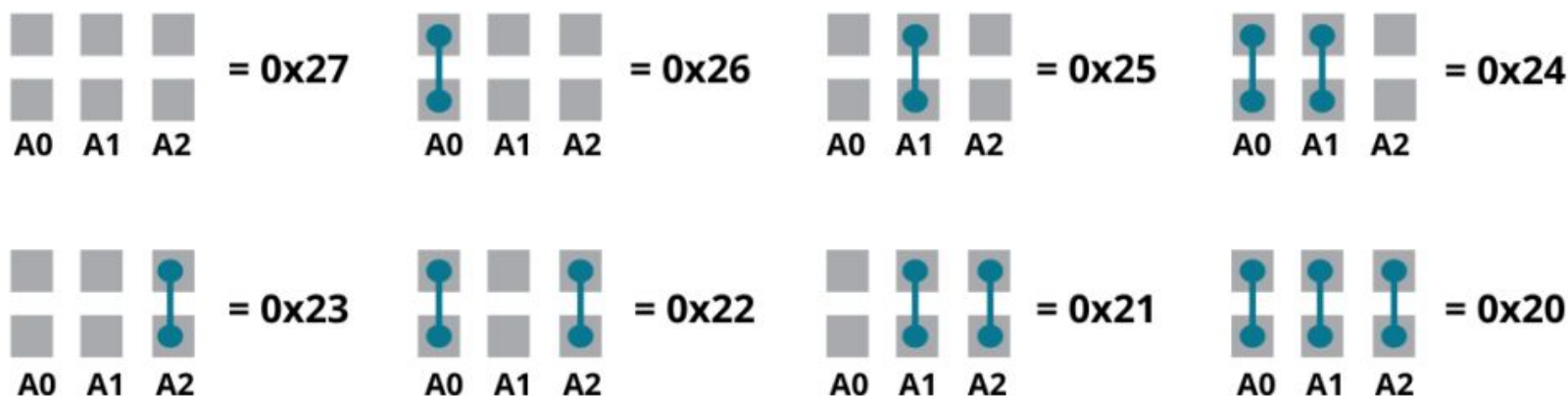
LCD I2C Arduino



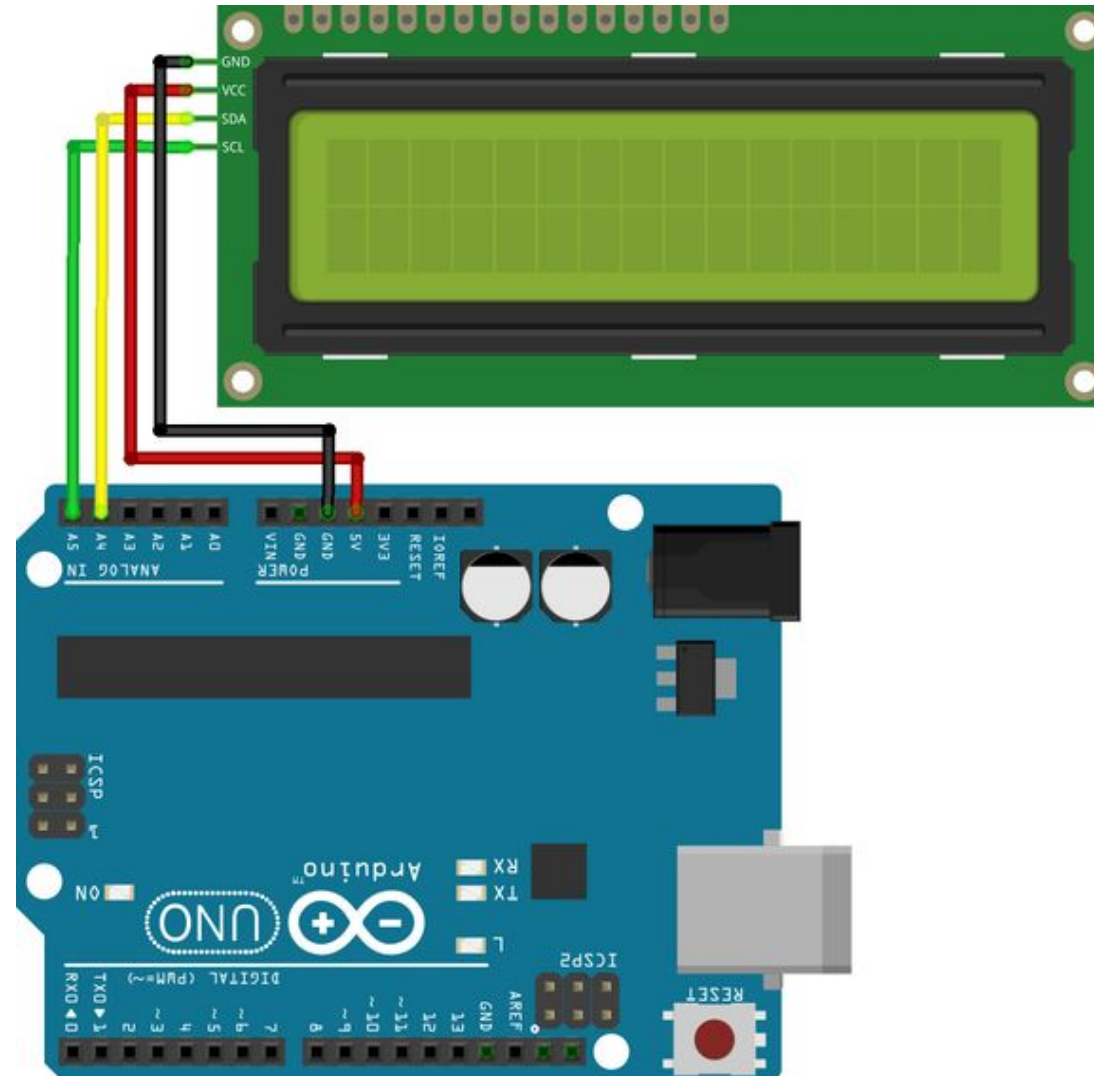
blog da
ROBÓTICA®

LCD I2C Arduino

- ❑ O barramento de configuração do endereço possui três jumpers de solda (A0, A1 e A2) ou pads de solda, usados para codificar o endereço.
- ❑ Acima de cada pad há uma conexão de aterramento (GND).
- ❑ Para alterar o endereço, basta conectar qualquer pad ao pino GND localizado acima deles, soldando-os.



LCD I2C Arduino



Conhecendo o endereço do módulo I2C

```
#include <Wire.h>
void setup()
{
    Wire.begin();

    Serial.begin(9600);
    while (!Serial);
    Serial.println("\nI2C Scanner");
}

void loop()
{
    byte error, address;
    int nDevices;
    nDevices = 0;
    for(address = 1; address < 127; address++ )
    {
        Wire.beginTransmission(address);
        error = Wire.endTransmission();
        if (error == 0)
        {
            Serial.print("Endereço I2C encontrado: 0x");
            if (address<16)
                Serial.print("0 ");
            Serial.println(address,HEX);

            nDevices++;
        }
    }
}
```

```
else if (error==4)
{
    Serial.print("ERRO ");
    if (address<16)
        Serial.print("0");
    Serial.println(address,HEX);
}
}
if (nDevices == 0)
    Serial.println("Nenhum endereço i2C encontrado ");
else

    Serial.println(" Feito !");

delay(5000);
}
```



LCD I2C Arduino

- ❑ Para utilizar o display é necessário instalar a biblioteca LiquidCrystal_I2C.h

```
#include <Wire.h> // Biblioteca utilizada para fazer a comunicação com o I2C
#include <LiquidCrystal_I2C.h> // Biblioteca utilizada para fazer a comunicação com o display 20x4

#define col 16 // Serve para definir o numero de colunas do display utilizado
#define lin  2 // Serve para definir o numero de linhas do display utilizado
#define ende  0x27 // Serve para definir o endereço do display.

LiquidCrystal_I2C lcd(ende,col,lin); // Chamada da função LiquidCrystal para ser usada com o I2C

void setup() //Inicia o display
{
  lcd.init(); // Serve para iniciar a comunicação com o display já conectado
  lcd.backlight(); // Serve para ligar a luz do display
  lcd.clear(); // Serve para limpar a tela do display
}

void loop()
{
  lcd.setCursor(0,0); // Coloca o cursor do display na coluna 1 e linha 1
  lcd.print("  TESTE I2C  "); // Comando de saída com a mensagem que deve aparecer na coluna 2 e linha 1.
  lcd.setCursor(0, 1); //Coloca o cursor do display na coluna 1 e linha 2
  lcd.print(" Automacao IFSC"); // Comando de saída com a mensagem que deve aparecer na coluna 2 e linha 2
  delay(3000); // delay de 5 segundos com todas as mensagens na tela
  lcd.clear(); // Limpa o display até o loop ser reiniciado
}
```

Prática com Display LCD

1. Altere o exemplo anterior para utilizar um LCD I2C:
 - a) Escreva na linha superior o seu nome;
 - b) Na linha inferior conforme o click do botão alterne entre as frases “IFSC-TUBARAO” e “PROG SIST EMB”;

Praticando com LCD I2C

- ☐ Crie um circuito com dois botões e o display LCD;
- ☐ Configure o display para apresentar uma mensagem inicial;
- ☐ Ao clicar em cada um dos botões você deve alternar as mensagens exibidas na tela.

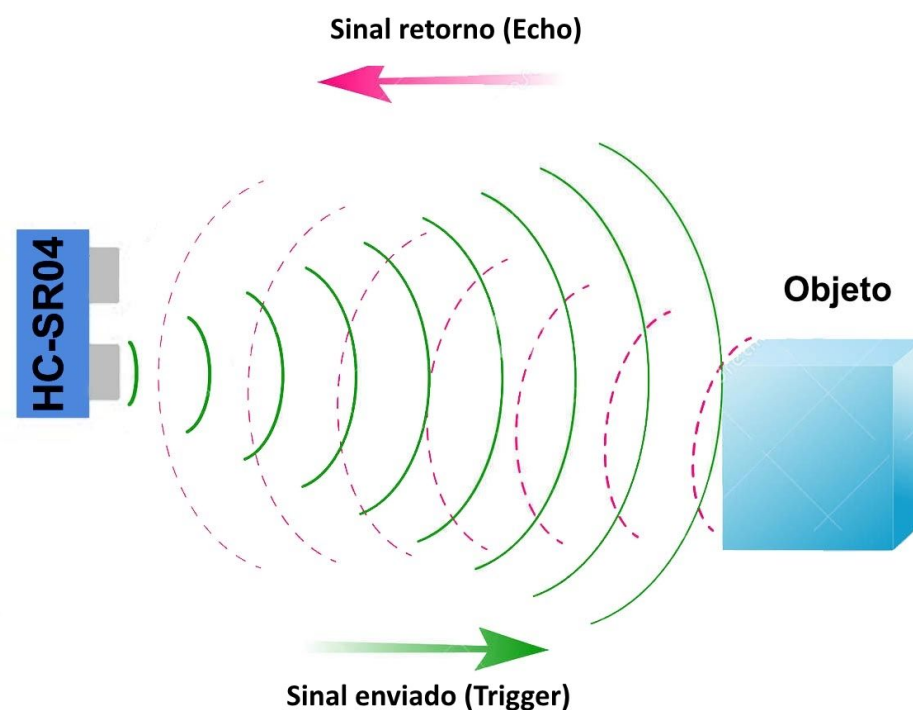
Sensor Ultrassônico

Sensor Ultrassônico

- ❑ O Sensor Ultrassônico HC-SR04 é um componente muito comum em projetos com Arduino;
- ❑ Ele permite que você faça leituras de distâncias entre 2 cm e 4 metros, com precisão de 3 mm;
- ❑ Pode ser utilizado simplesmente para medir a distância entre o sensor e um objeto, como para acionar portas do microcontrolador, desviar um robô de obstáculos, acionar alarmes, etc.

Sensor Ultrassônico

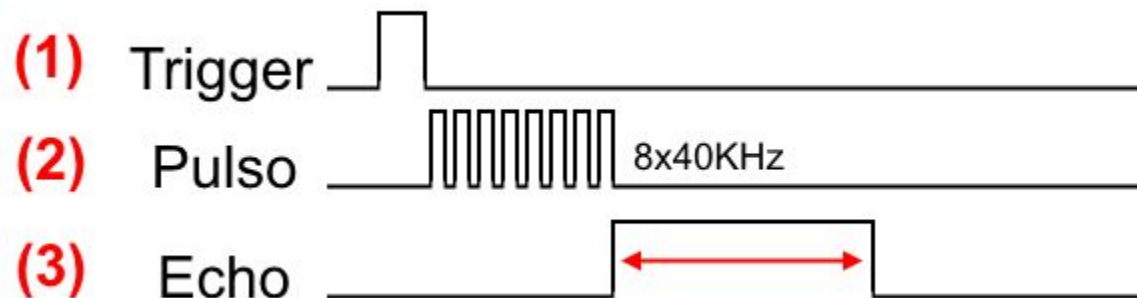
- ❑ O funcionamento do HC-SR04 se baseia no envio de sinais ultrassônicos pelo sensor, que aguarda o retorno (echo) do sinal, e com base no tempo entre envio e retorno, calcula a distância entre o sensor e o objeto detectado.



Sensor Ultrassônico

- ❑ Primeiramente é enviado um pulso de 10μs, indicando o início da transmissão de dados;
- ❑ Depois disso, são enviados 8 pulsos de 40 KHz e o sensor então aguarda o retorno (em nível alto/high), para determinar a distância entre o sensor e o objeto, utilizando a equação:

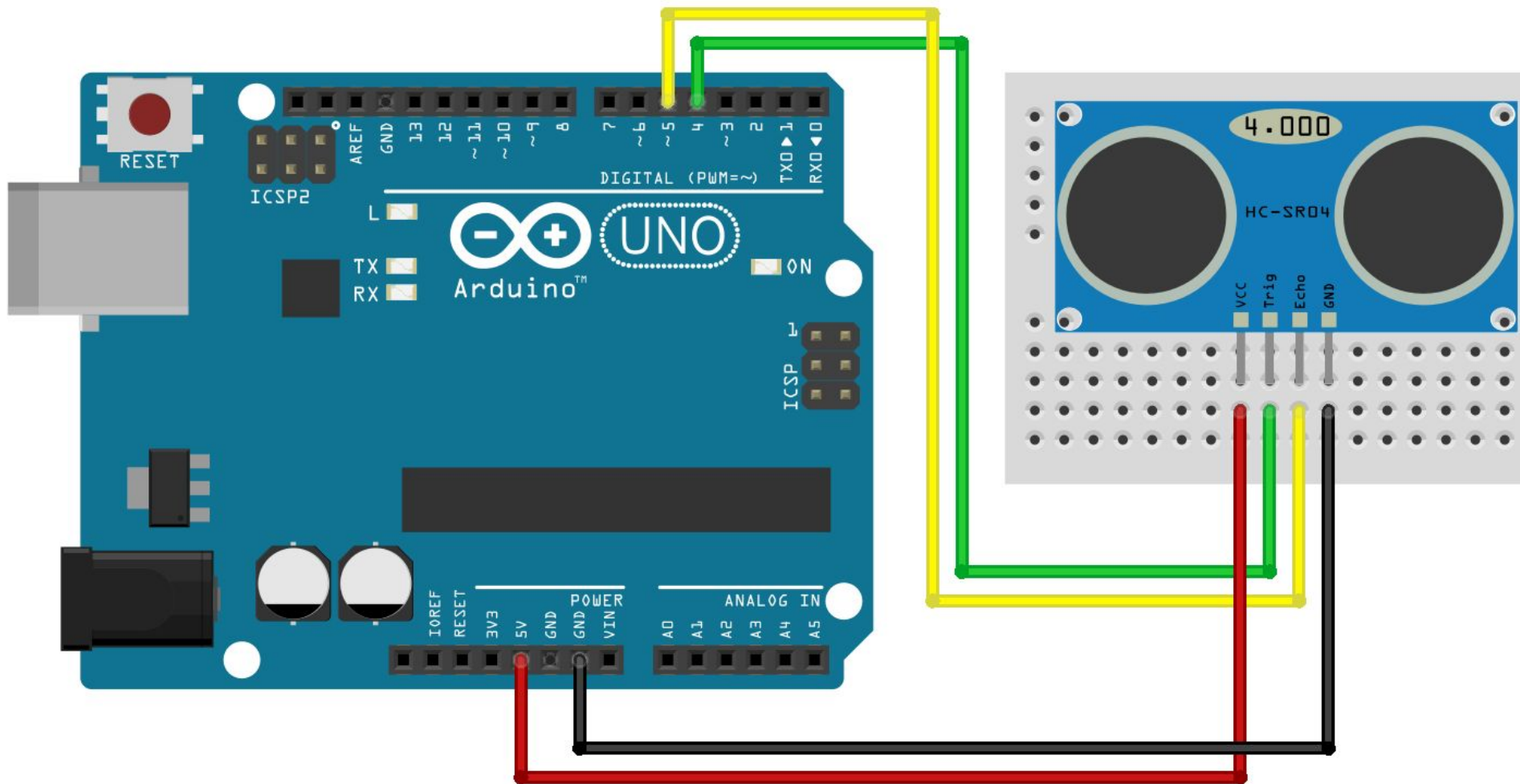
$$\text{Distância} = (\text{Tempo echo em nível alto} * \text{velocidade do som}) / 2$$



Conectando o Sensor Ultrassônico

- ❑ Para ligação do sensor ao microcontrolador, são utilizados 4 pinos: Vcc, Trigger, ECHO e GND.
- ❑ A ligação com o Arduino deve utilizar, além dos pinos de alimentação, os **pinos digitais**, neste caso 4 para o Trigger, e 5 para o Echo.
- ❑ A alimentação será feita pelo pino 5V do Arduino.
- ❑ Necessário o uso de uma biblioteca: [link](#)

Sensor Ultrassônico



Obtendo a Distância com o Sensor Ultrassônico

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "Ultrasonic.h"

#define col 16 // Serve para definir o numero de colunas do display utilizado
#define lin  2 // Serve para definir o numero de linhas do display utilizado
#define ende  0x27 // Serve para definir o endereço do display.

Ultrasonic ultrasonic(4,5);
LiquidCrystal_I2C lcd(ende,col,lin);

void setup() //Inicia o display{
  lcd.init(); // Serve para iniciar a comunicação com o display já conectado
  lcd.backlight(); // Serve para ligar a luz do display
  lcd.clear(); // Serve para limpar a tela do display
}
void loop() {
  lcd.setCursor(0,0); // Coloca o cursor do display na coluna 1 e linha 1
  lcd.print("TESTE SONAR"); // Comando de saída com a mensagem que deve aparecer na coluna 2 e linha 1.
  lcd.setCursor(0, 1); //Coloca o cursor do display na coluna 1 e linha 2
  lcd.print(ultrasonic.Ranging(CM));
  delay(300); // delay de 5 segundos com todas as mensagens na tela
  lcd.clear(); // Limpa o display até o loop ser reiniciado
}
```

Praticando com Sensor Ultrasônico

- ❑ Crie um circuito com um display LCD 16x2 e um sensor ultrasônico;
- ❑ Você deve apresentar no display a distância obtida pelo sensor e conforme este valor apresentar uma mensagem adicional:
 - ❑ Caso o objeto esteja a mais de 100 cm, você deve apresentar a mensagem “Sem obstaculo”;
 - ❑ Caso o objeto esteja entre 50 cm e 100 cm, você deve apresentar a mensagem “Obstaculo prox.”;
 - ❑ Caso o objeto esteja a menos de 50 cm, você deve apresentar a mensagem “Cuidado pare!”



Sensor de Umidade e Temperatura DHT-22

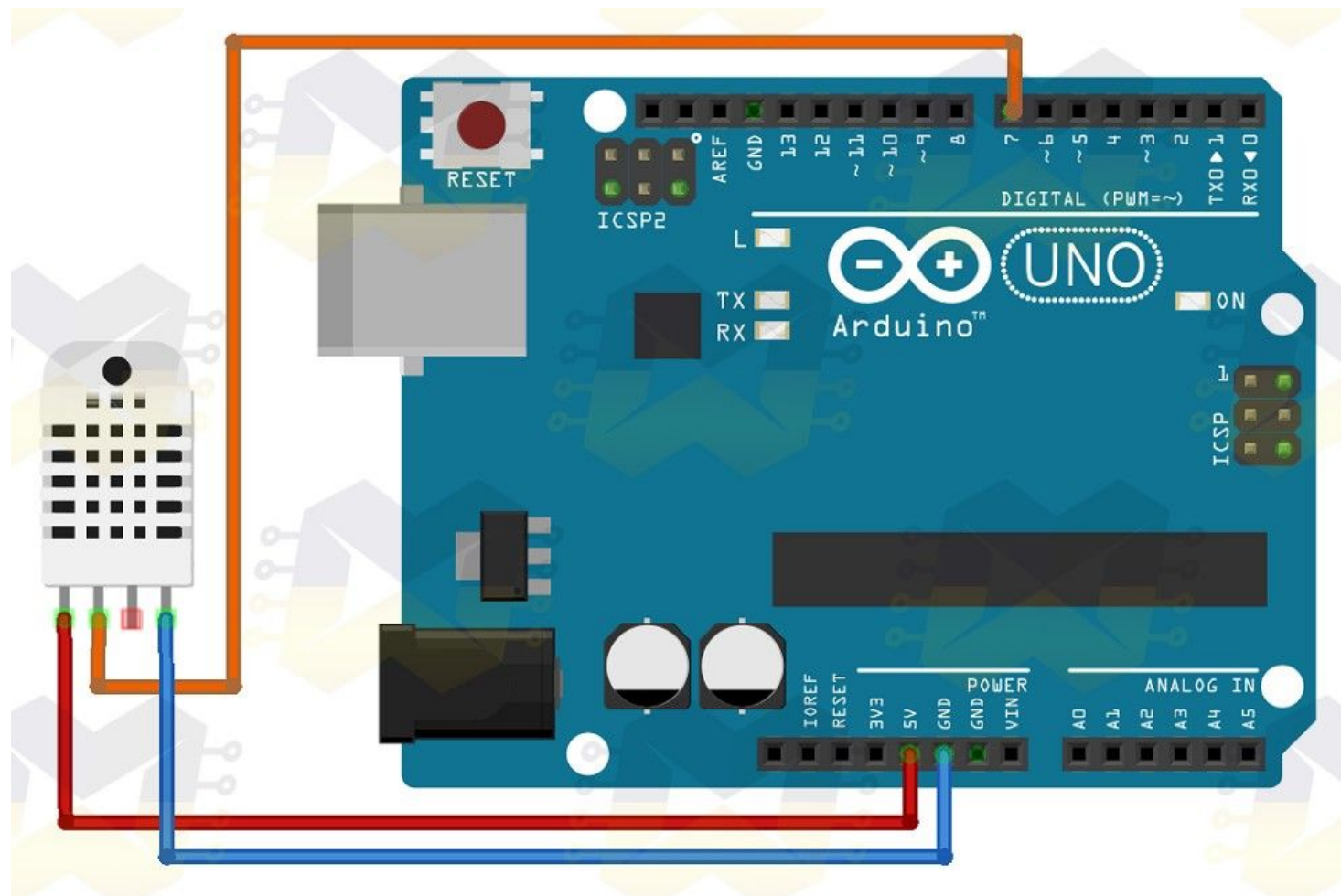
DHT-22

- ❑ O Sensor de Umidade e Temperatura DHT22 / AM2302 é um dos componentes mais utilizados em projetos que envolva medição de temperatura e umidade ambiente.
- ❑ Este sensor faz medições de temperatura de -40° até 80° celsius e mede a umidade do ar nas faixas de 0 a 100%.
- ❑ A precisão (margem de erro) do sensor para medição de temperatura é de aproximadamente $0,5^{\circ}$ celsius e para umidade é de 2%

DHT-22 - Especificações e Características

- ❑ Modelo: AM2302
- ❑ Faixa de medição (umidade): 0 a 100%
- ❑ Faixa de medição (temperatura): -40° a 80°C
- ❑ Tensão de operação: 3 – 5VDC
- ❑ Corrente de operação: 2,5mA
- ❑ Corrente em stand by: 100uA a 150 uA
- ❑ Precisão (umidade): $\pm 2\%$
- ❑ Precisão (temperatura): $\pm 0,5^\circ$ celsius
- ❑ Tempo de resposta: 2s

DHT-22



DHT-22

```
#include <DHT.h>; //INCLUSÃO DE BIBLIOTECA

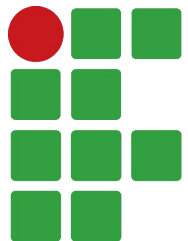
#define DHTPIN 7 //PINO DIGITAL UTILIZADO PELO DHT22
#define DHTTYPE DHT22 //DEFINE O MODELO DO SENSOR (DHT22 / AM2302)

DHT dht(DHTPIN, DHTTYPE); //PASSA OS PARÂMETROS PARA A FUNÇÃO

void setup(){
  Serial.begin(9600); //INICIALIZA A SERIAL
  dht.begin(); //INICIALIZA A FUNÇÃO
  delay(2000); //INTERVALO DE 2 SEGUNDO ANTES DE INICIAR
}
void loop(){
  Serial.print("Umidade: "); //IMPRIME O TEXTO NA SERIAL
  Serial.print(dht.readHumidity()); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO
  Serial.print("%"); //IMPRIME O TEXTO NA SERIAL
  Serial.print(" / Temperatura: "); //IMPRIME O TEXTO NA SERIAL
  Serial.print(dht.readTemperature(), 0); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO E REMOVE A PARTE DECIMAL
  Serial.println("*C"); //IMPRIME O TEXTO NA SERIAL
  delay(2000); //INTERVALO DE 2 SEGUNDOS * NÃO DIMINUIR ESSE VALOR
}
```

Praticando com Sensores e LCD

- ❑ Crie um circuito com:
 - ❑ Um display LCD 16x2
 - ❑ Um sensor ultrassônico;
 - ❑ Um botão;
 - ❑ Um sensor DHT-22;
 - ❑ Um Potenciômetro;
- ❑ A cada clique do botão você deve alternar entre os sensores e apresentar no display qual o sensores está sendo monitorado e qual o valor medido.



INSTITUTO FEDERAL

Santa Catarina
Câmpus Tubarão

Obrigado!

Fernando Silvano Gonçalves

fernando.goncalves@ifsc.edu.br

se.cst.tub@ifsc.edu.br