rexlab | Desde 1997

Introdução a Arduino

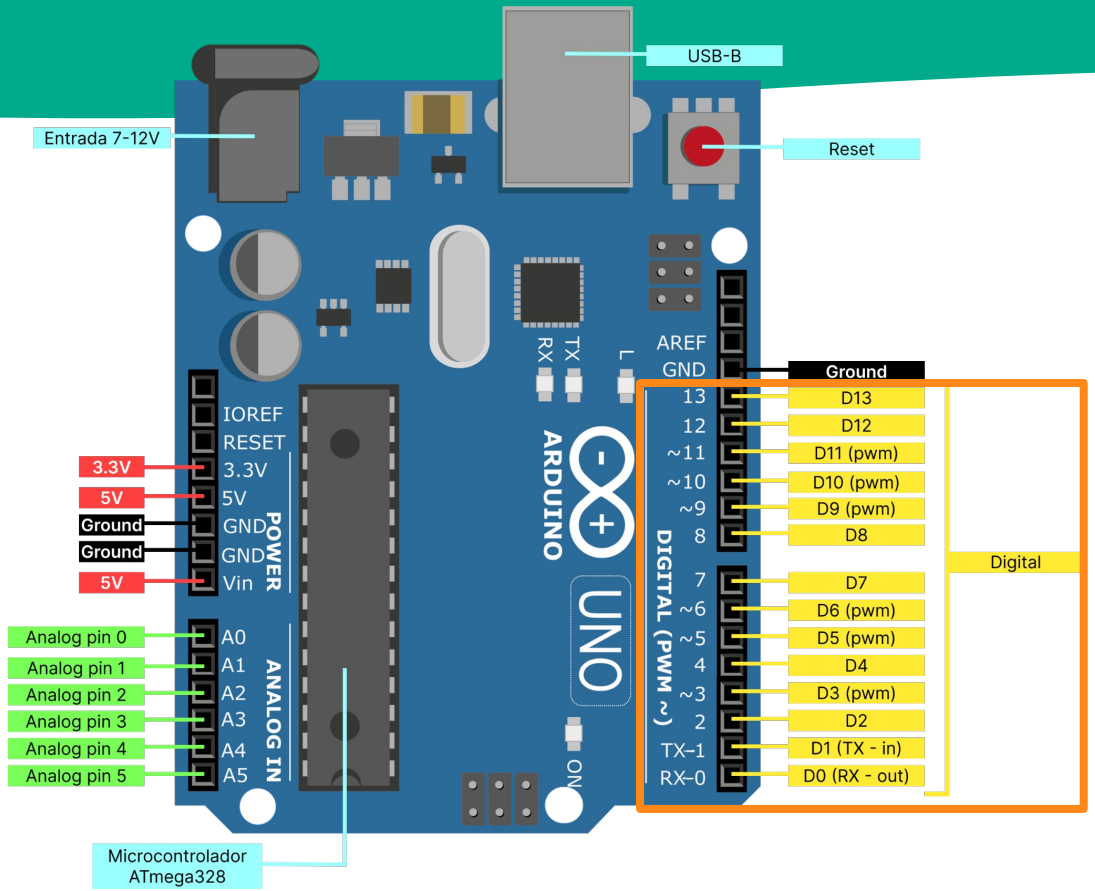
Sensores e Atuadores

Agenda

- Revisão
- Sensores e Atuadores
- Bibliotecas
- Exemplos
- Exercícios

Revisão

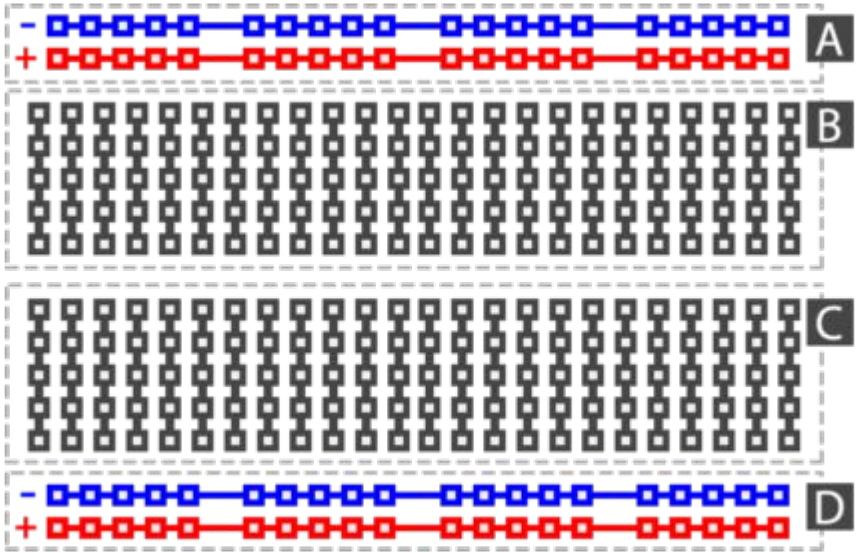
Revisão



rexlab | Desde 1997

Revisão

Uma protoboard serve para prototipagem de circuitos eletrônicos. É de fácil utilização e funciona da seguinte forma:



Revisão

sketch_apr23a.ino

```
1   void setup() {
2     // put your setup code here, to run once:
3
4   }
5
6   void loop() {
7     // put your main code here, to run repeatedly:
8
9   }
10
```

Revisão

A manipulação de estados no Arduino é o processo de **alterar o estado de um pino digital** de entrada ou saída
O estado de um pino digital pode ser **alto (HIGH ou 1)** ou **baixo (LOW ou 0)**

HIGH representa um nível lógico **alto**, geralmente **5V**
LOW representa um nível lógico **baixo**, geralmente **0V**

A forma mais comum de manipular o estado do pino é pela função **digitalWrite()**

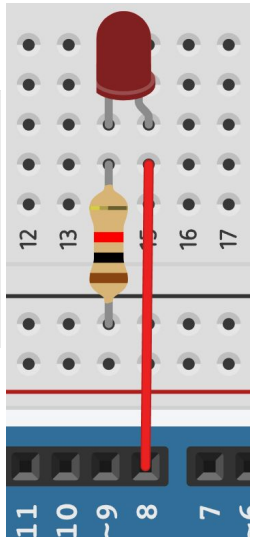
```
1  // Define o pino 8 como alto
2  digitalWrite(8, HIGH);
3
4  // Define o pino 8 como baixo
5  digitalWrite(8, LOW);
6
```

Revisão

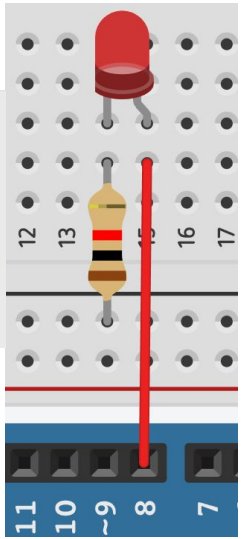
Os sinais high e low também são usados para controlar dispositivos externos.

Por exemplo, um LED pode ser ligado ou desligado enviando um sinal high ou low para o pino do LED.

```
1  void setup()
2  {
3    pinMode(8, OUTPUT);
4  }
5
6  void loop()
7  {
8    digitalWrite(8, LOW);
9
10 }
11
```



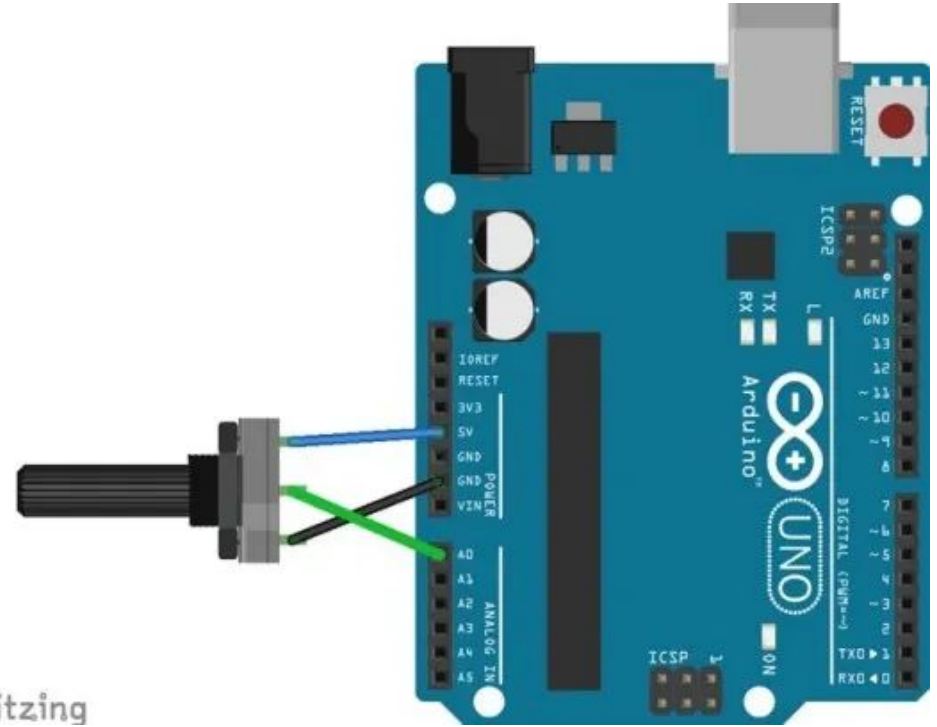

```
1  void setup()
2  {
3    pinMode(8, OUTPUT);
4  }
5
6  void loop()
7  {
8    digitalWrite(8, HIGH);
9
10 }
11
```



Revisão

```
void setup() {
  Serial.begin(115200);
}

void loop() {
  int potValue = analogRead(A0);
  Serial.println(potValue);
}
```



Sensores e Atuadores

Sensores e Atuadores

Os sensores são dispositivos que captam informações do ambiente e as convertem em sinais elétricos que podem ser interpretados por um sistema de controle.

Os atuadores são dispositivos que convertem sinais elétricos em uma ação física, como mover uma parte de uma máquina ou mudar o estado de um sistema.

rexlab | Desde 1997

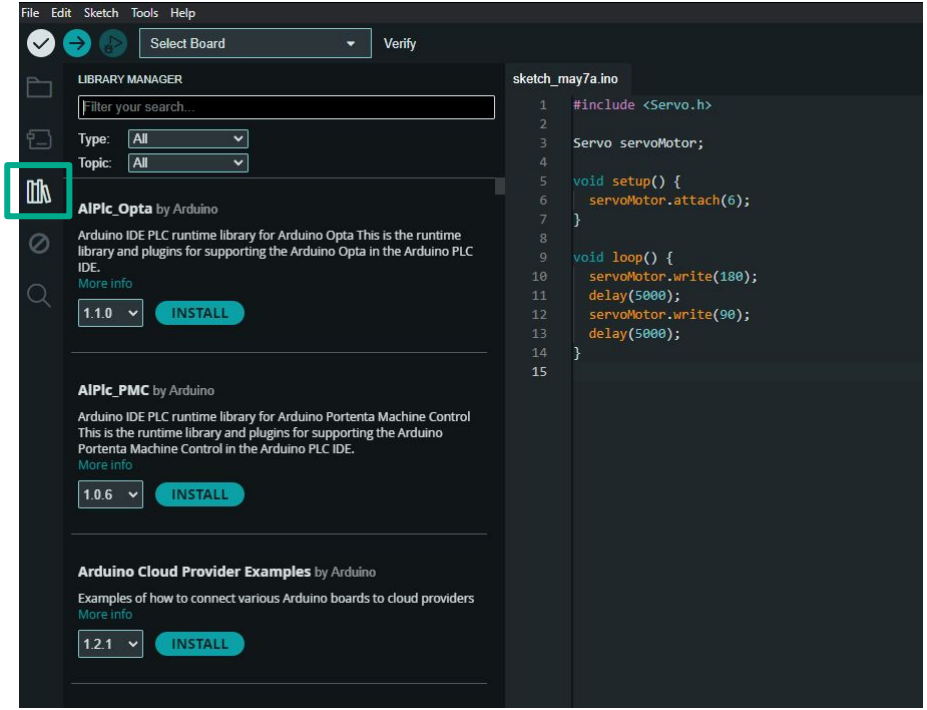
Bibliotecas

Bibliotecas

Muitos sensores ou atuadores utilizam bibliotecas específicas que economizam trabalho dos desenvolvedores.

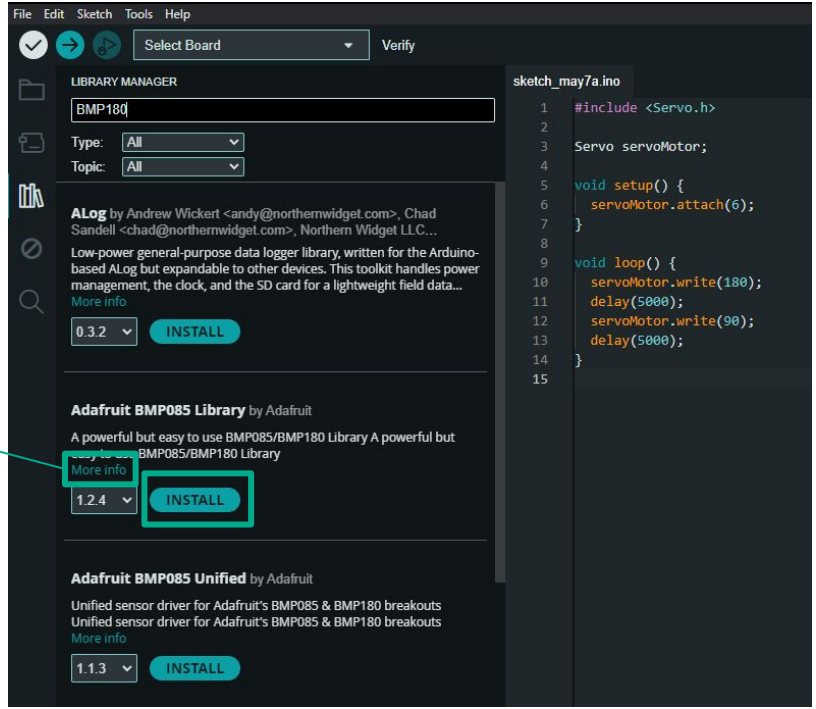
- **Servo Motores**
- **Sensores de Temperatura**
- **Sensores Infravermelho**
- **Sensores de Corrente e Tensão**
- **Teclados Matriciais**

Instalando uma Biblioteca



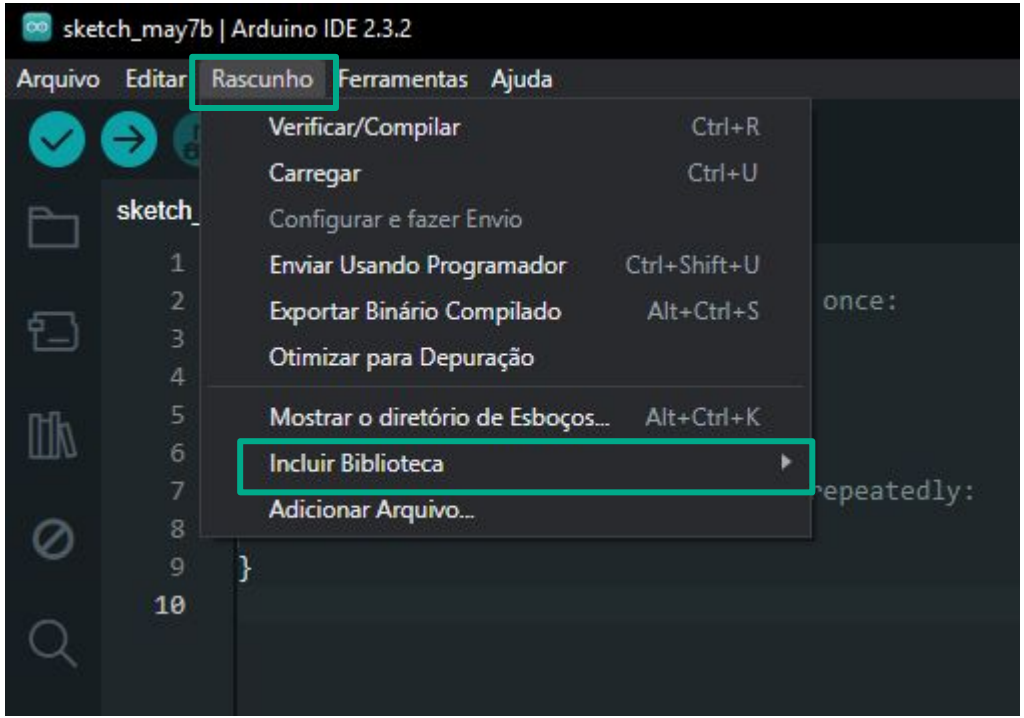
Instalando uma Biblioteca

Github / Documentação

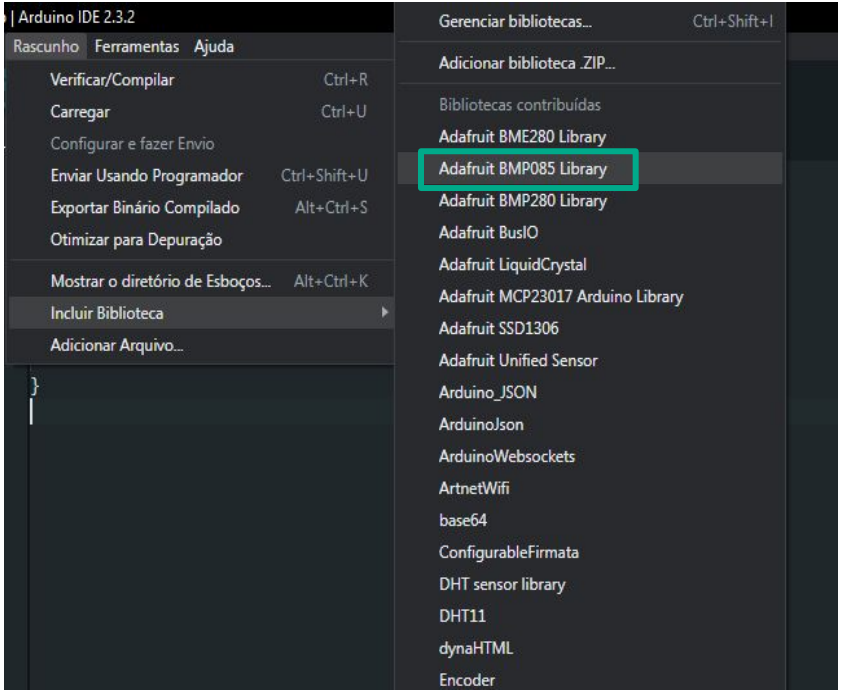


rexlab | Desde 1997

Instalando uma Biblioteca



Instalando uma Biblioteca



rexlab | Desde 1997

Instalando uma Biblioteca

may7b.ino

```
#include <Adafruit_BMP085.h>

void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Servo Motor

Como funciona o Servo Motor

Um servo motor é um dispositivo que converte um sinal elétrico em movimento rotacional preciso, usando um sistema com potenciômetro interno para controlar sua posição ou velocidade.

rexlab | Desde 1997

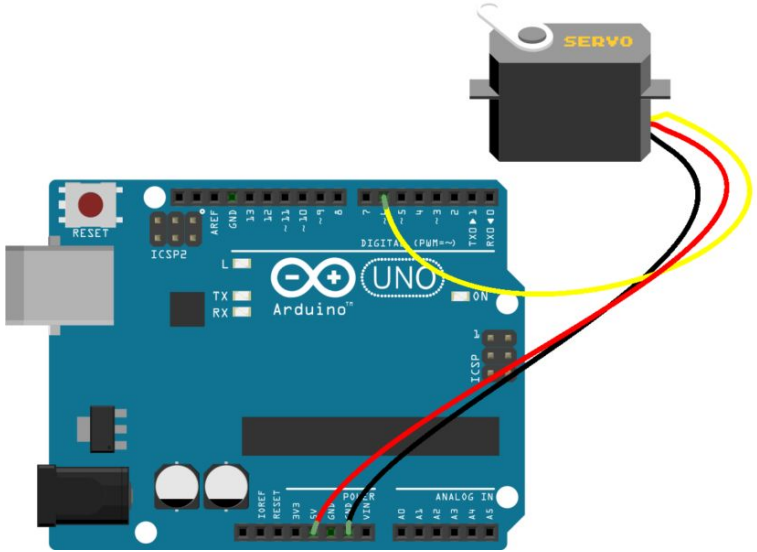
Na Prática

```
#include <Servo.h>

Servo servoMotor;

void setup() {
  servoMotor.attach(6);
}

void loop() {
  servoMotor.write(180);
  delay(5000);
  servoMotor.write(90);
  delay(5000);
}
```



Exercício 1 - Controle de Servo com Potenciômetro

Objetivo:

Conecte um potenciômetro ao Arduino e um servo motor a um pino adequado, utilizando a leitura do potenciômetro para controlar a posição angular do servo, de 0 a 180 graus.

Dica: Use a função map(valor, deMin, deMax, paraMin, paraMax) para converter a faixa de leitura do potenciômetro (por exemplo, 0-1024) para a faixa de ângulos do servo (0-180 graus) .

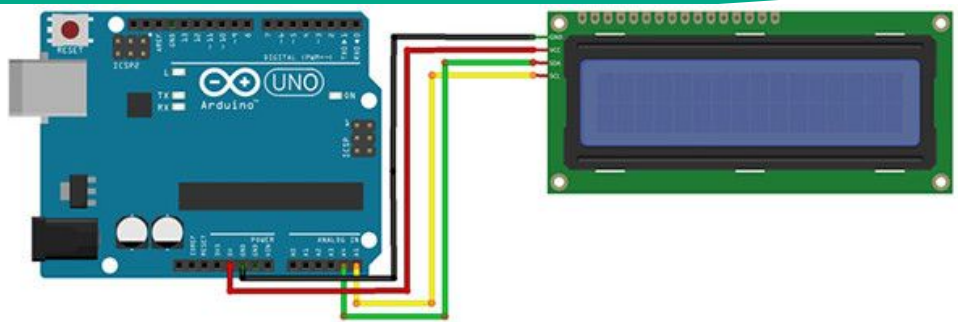
Display LCD

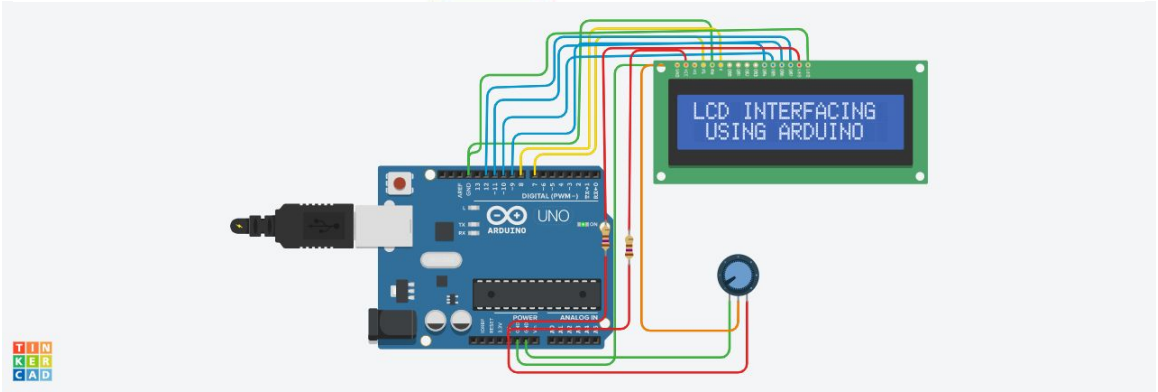
Display LCD

Permite a exibição de textos e números em uma tela de cristal líquido. O módulo I2C facilita a comunicação do display com microcontroladores e outros dispositivos, utilizando apenas dois fios para transmissão de dados. Isso simplifica significativamente o esquema de conexões e otimiza o uso de pinos em projetos de eletrônica.

rexlab | Desde 1997

I2C na Prática





LCD na Prática

sketch_may7b.ino

```ino
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Configura o display LCD no endereço I2C 0x27 e especifica as dimensões (16 colunas x 2 linhas)
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  // Inicializa o display
  lcd.init();
  // Liga a luz de fundo
  lcd.backlight();
  // Exibe uma mensagem de texto
  lcd.setCursor(0, 0); // Posiciona o cursor na primeira coluna da primeira linha
  lcd.print("Hello, world!");
  lcd.setCursor(0, 1); // Posiciona o cursor na primeira coluna da segunda linha
  lcd.print("LCD I2C Test");
}
```

Exercício 2 - Posição do Servo no LCD

Objetivo:

Desenvolva um algoritmo que gire o servo de 0 a 180 graus, depois de 180 a 0 graus. Sempre mostre o valor da posição atual do servo no display LCD.

Dica:

Utilize dois laços de repetição dentro do loop para fazer o servo girar. Você pode utilizar o for para isso.

Exemplo: `for(int pos = 0; pos <= 180; pos++)`

HC-SR04

Sensor de Distância Ultrassônico

rexlab

Desde 1997

HC-SR04: Sensor de Distância Ultrassônico

Dispositivo de medição de distâncias ultrassônico, amplamente utilizado em projetos de robótica e automação. Funciona emitindo ondas sonoras e medindo o tempo que elas levam para retornar após bater em um objeto, calculando assim a distância até o mesmo. É conhecido por sua precisão e baixo custo.

rexlab | Desde 1997

Na Prática

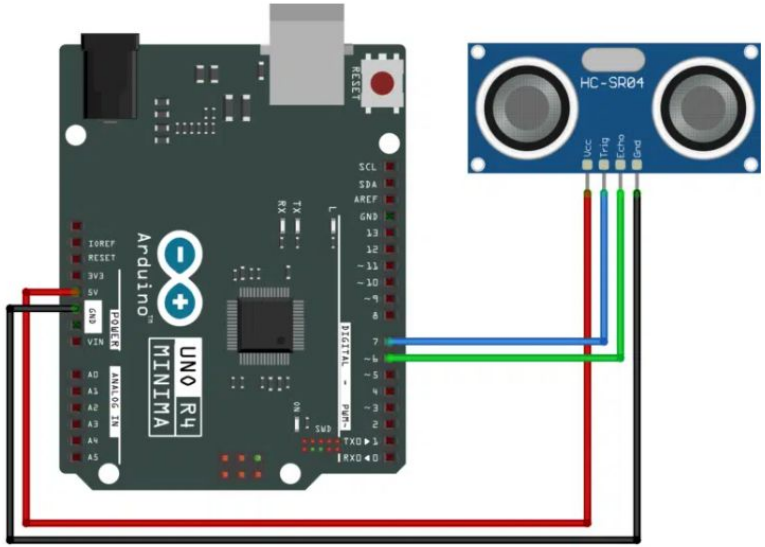
```ino
#include <Ultrasonic.h>

#define TRIGGER_PIN 7  // Pino usado para enviar o pulso ultrassônico
#define ECHO_PIN    6  // Pino usado para receber o eco

Ultrasonic ultrasonic(TRIGGER_PIN, ECHO_PIN); // Instancia o objeto Ultrasonic

void setup() {
  Serial.begin(9600); // Inicia a comunicação serial
}

void loop() {
  long distance = ultrasonic.read(); // Lê a distância em centímetros
  Serial.print("Distância: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(1000); // Espera um segundo antes da próxima medição
}
```



Exercício 3 - Display LCD com Sensor de Distância

Objetivo:

Mostre em um display LCD com módulo I2C, os valores lidos por um sensor de distância ultrassônico.

Termistor

Termistor

Um tipo de resistor cuja resistência varia significativamente com a temperatura. Eles são amplamente utilizados em aplicações de medição e controle de temperatura, sendo classificados em dois tipos principais: NTC (coeficiente de temperatura negativo) e PTC (coeficiente de temperatura positivo).

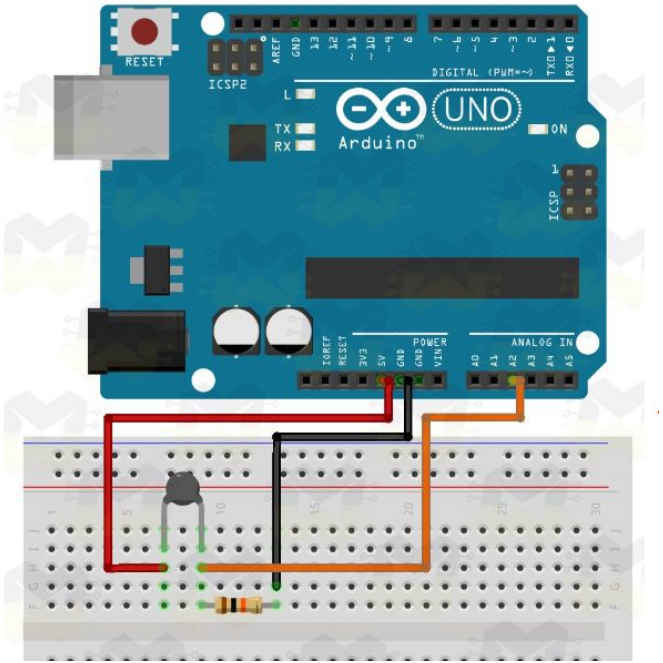
Steinhart-Hart

A equação de Steinhart-Hart é um modelo matemático usado para descrever a relação não linear entre a resistência e a temperatura em termistores. No Arduino, essa equação é usada para calcular a temperatura precisamente ao converter o valor da resistência do termistor obtido através do circuito. Ela envolve três coeficientes característicos do termistor (A, B, C) que são usados na fórmula para determinar a temperatura em Kelvin.

Na Prática

```
#include <Thermistor.h> //INCLUSÃO DA BIBLIOTECA

Thermistor temp(2); //VARIÁVEL DO TIPO THERMISTOR,
//INDICANDO O PINO ANALÓGICO (A2) EM QUE O TERMISTOR ESTÁ CONECTADO
void setup() {
  Serial.begin(9600); //INICIALIZA A SERIAL
  delay(1000); //INTERVALO DE 1 SEGUNDO
}
void loop() {
  int temperature = temp.getTemp(); //VARIÁVEL DO TIPO INTEIRO QUE
  //RECEBE O VALOR DE TEMPERATURA CALCULADO PELA BIBLIOTECA
  Serial.print("Temperatura: "); //IMPRIME O TEXTO NO MONITOR SERIAL
  Serial.print(temperature); //IMPRIME NO MONITOR SERIAL A
  //TEMPERATURA MEDIDA
  Serial.println("*C"); //IMPRIME O TEXTO NO
  //MONITOR SERIAL
  delay(1000); //INTERVALO DE 1 SEGUNDO
}
```

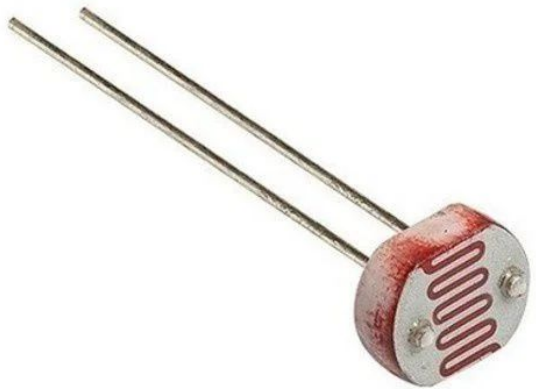


Desde 1997

Fotoresistor

LDR - Resistor Dependente de Luz

Um LDR (Resistor Dependente de Luz) é um componente eletrônico cuja resistência varia de acordo com a intensidade da luz que incide sobre ele. Geralmente, sua resistência diminui quando exposto a mais luz e aumenta na escuridão. Este dispositivo é comumente utilizado em circuitos que precisam detectar e reagir à luz, como em sistemas automáticos de iluminação.



rexlab Desde 1997

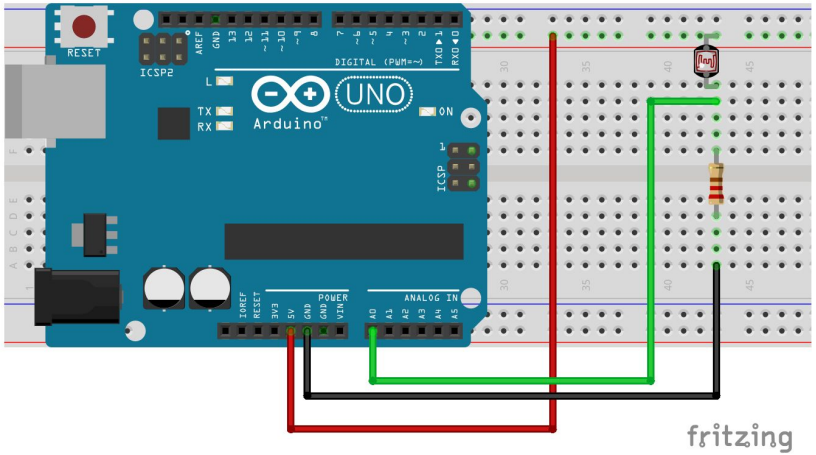
Na Prática

```
const int pinoLDR = A5;

void setup(){
  pinMode(pinoLDR, INPUT);
  Serial.begin(9600);
}
void loop(){

  int valorLDR = analogRead(pinoLDR);
  Serial.println(valorLDR);

}
```



Exercício 4 - Display LCD com leitura de sensores

Objetivo:

Mostre em um display LCD os valores lidos de um termistor NTC e de um fotoresistor.

A temperatura deve ser vista na parte superior do LCD de duas linhas e o valor de luminosidade na parte inferior.

O valor lido de luminosidade deve ser convertido para mostrar o valor na faixa de 0 a 100%.

Exemplo: “Temp: 25C”
“Luz: 50%”

Trabalho Final

Trabalho Final

Objetivo:

Com todos os componentes que temos acesso no laboratório, crie uma versão do Blockino, que contenha alguns sensores e atuadores conectados nas portas do Arduino.

O objetivo da sua versão do Blockino deve ser de permitir que estudantes programem arduino, sem precisar ensinar eletrônica e montagem dos componentes.

Dessa forma eles têm acesso a uma plataforma com todos os componentes já conectados no Arduino para realizarem práticas de programação.

Trabalho Final

Entrega:

Postagem no InTecEdu de um documento em formato PDF explicando o motivo pelo qual escolheu os componentes para montar a sua versão do Blockino.

Utilizando um simulador (Tinkercad, Wokiwi ou Fritzing) desenhe as conexões que cada componente deve ter com o arduino. Anexe a imagem ao PDF.

Entrega até dia 15/05 - Quarta-feira que vem.