



Universidade Federal do Rio Grande do
Norte
Campus Natal

Material: Programação para Arduino

Orientador: Prof. Dr. Irineu Lopes Palhares Junior

Discente: Steffany Jennyfer Santos de Araújo

Discente: Francisco Douglas Marques

Discente: José Ramos de Oliveira Moreira Chacon

Natal - Rio Grande do Norte
Abril de 2022

Sumário

1 Resumo	2
2 Introdução	2
3 Introdução à Lógica de Programação	2
3.1 Receita de bolo Formigueiro:	2
3.2 Um Algoritmo deve ser:	4
3.3 Algumas ferramentas na programação:	4
4 Conceitos básicos sobre eletrônica	5
5 Conceitos básicos sobre o Arduino	5
5.1 O que é Arduino	5
5.2 Instalação do Arduino	6
5.3 História	7
5.4 Para que serve	7
5.4.1 Tarefa	7
5.5 Tipos de Arduino	7
5.6 Componentes do Arduino	7
5.6.1 Tarefa	9
5.7 IDE	10
5.7.1 Vamos praticar!	10
6 Curso: Programação para Arduino	10
6.1 Relés, Potenciômetro, transistor como chave e regulador de tensão	10
6.2 Display 7 Segmentos, Opto Acoplador, Display LCD 16×2	15
6.3 LED RGB, Neopixel	18
6.3.1 LED RGB	18
6.4 Keypad e Dips switch	19
6.5 Sensor de temperatura, luminosidade, PIR, IR	21
6.6 Sensor de distância, Tilt, gás	26
6.7 Motor DC, Servo motor, Higrômetro	26
6.7.1 Motor DC	26
6.7.2 Servo motor	26
6.7.3 Higrômetro - Sensor de umidade do solo	26
6.7.4 Atividade - estudo de caso	29
6.8 Projeto final	30
7 Curso: Programação para Arduino 2	30
8 Considerações finais	30

1 Resumo

Neste curso de programação para Arduíno iremos introduzir conceitos básicos sobre o Arduíno, eletrônica, protoboard e diodo emissor de luz. Iremos ver também transistor como chave e regulador de tensão, relés e potenciômetro, além de outros conteúdos baseado nos assuntos introduzidos, onde usamos como apoio o simulador autodesk TinkerCad para a montagem dos dispositivos utilizados em aula.

2 Introdução

Arduíno é uma plataforma de prototipagem eletrônica muito versátil e amplamente utilizada por estudantes, hobbistas e profissionais das mais diversas áreas. O objetivo principal é tornar o acesso à prototipagem eletrônica mais fácil, mais barata e flexível. As versões mais simples de placa utilizam um microcontrolador da família Atmel AVR e uma linguagem de programação baseada em C/C++. Com ele é possível criar projetos variados em eletrônica, desde os mais simples até aplicações intermediárias como sistema de automação residencial ou industrial, alarmes e outros. O Arduíno foi desenvolvido com base no conceito open-source, em tradução literal “código aberto”, que significa que o projeto da placa e o firmware podem ser utilizados livremente por outros desenvolvedores e fabricantes.

3 Introdução à Lógica de Programação

Lógica de programação é o modo como se escreve um programa de computador, um algoritmo. Um algoritmo é uma sequência de passos para se executar uma função. Um exemplo de algoritmo, fora da computação, é uma receita culinária, onde devemos seguir todos os passos de forma sequencial para que não haja nenhum problema. É interessante destacar que nós executamos programas diariamente, o que vamos passar para o computador são comandos para que ele execute os programas que nós queremos, de forma lógica. Ao programar iremos escrever uma “receita de torta” (algoritmos), de forma sequencial e lógica para que o computador leia e entenda o que deve ser realizado ao executar o programa. Para isto é necessário uma linguagem de programação. A linguagem de programação é como uma língua normal, um grupo de palavras com significados. Estas linguagens fazem o computador assimilar cada comando e função de um algoritmo, depois executar cada função. A linguagem de programação é somente como se escreve o algoritmo. O grande problema para muitos é o que “dizer” para o computador fazer o que é desejado, por isso é muito importante utilizar a lógica ao realizarmos os programas, pós o computador só irá ler e executar aquilo que for “apresentado”. Vejamos um exemplo prático de como utilizamos algoritmos fora do computador.

3.1 Receita de bolo Formigueiro:

1. Separar os ingredientes:

- 4 colheres de sopa de margarina
- 1 e 1/2 copo de açúcar
- 4 ovos (claras em neve)
- 2 copos de farinha de trigo
- 1 copo de leite
- 100 g de chocolate granulado

- 100 g de coco ralado
 - 1 colher (sopa) de fermento em pó
2. Pegue uma forma;
 3. Bata a margarina com o açúcar até ficar cremoso dentro da forma, junte as gemas e continue batendo;
 4. Acrescente o leite e os demais ingredientes;
 5. Em seguida crescente as claras em neve, mexendo delicadamente;
 6. Leve para assar em forno quente;
 7. Após 45min abra o forno;
 8. Retire o bolo;
 9. Desenforme-o.

Como poderíamos detalhar ainda mais os passos seguidos na receita acima?

Percebe que na receita a cima seguimos uma ordem?

Se modificássemos a ordem, o resultado seria completamente diferente, por exemplo, digamos que, após o terceiro passo executássemos o sexto, sétimo e oitavo e voltássemos para o para o quarto, seguindo a sequência e ignorando apenas os passos já executados, com certeza nosso bolo não teria o mesmo resultado, isso também é valido para nossos programas de computadores, onde é essencial seguirmos uma sequência lógica. Digamos que o objetivo agora seja calcular a média de um aluno e exibir no monitor “aprovado” se $MEDIA \geq 7$ e “reprovado” se $MEDIA < 7$, como isso seria feito?

- Cálculo da média de um aluno:

Algoritmo Calculo_Media

Var Nota1, Nota2, MEDIA: real;

Início

Leia Nota1, Nota2;

MEDIA - (Nota1 + Nota2) / 2;

Se MEDIA ≥ 7 **então**

Escreva “Aprovado”;

Senão

Escreva “Reprovado”;

Fim_se

Fim

Figura 1: Calculo da media

Aplicamos o conceito de algoritmo diariamente sempre que estabelecemos um planejamento mental para realizar uma determinada tarefa, considerando que deveremos executar um conjunto de passos até atingir o objetivo desejado. Mas afinal, **para que serve um algoritmo?**

O algoritmo é uma sequência de passos lógicos e finitos que permite solucionar problemas. O objetivo de aprender a criar algoritmos é que este é a base de conhecimentos para as linguagens de programação. Em geral, existem muitas maneiras de resolver o mesmo problema. Ou seja, podem ser criados vários algoritmos diferentes para resolver o mesmo problema. Assim, ao criarmos um algoritmo, indicamos uma dentre várias possíveis sequências de passos para solucionar o problema. Um programa de computador é um algoritmo escrito em um formato compreensível pelo computador. Na elaboração de um algoritmo devem ser especificadas ações claras e precisas que resultem na solução do problema proposto. A lógica está na **correta sequência de passos que deve ser seguida para alcançar um objetivo específico**, pois o grau de detalhe do algoritmo dependerá da situação em que o programador se encontra.

3.2 Um Algoritmo deve ser:

- Completo, Todas as ações precisam ser descritas e devem ser únicas.
- Sem redundância, um conjunto de instruções só pode ter uma única forma de ser interpretada.
- Determinístico, se as instruções forem executadas, o resultado esperado será sempre atingido.
- Finito, as instruções precisam terminar após um número limitado de passos.

3.3 Algumas ferramentas na programação:

- Condições:

Se <condição> então faça:

Se não faça:

- Condições:

Leia a memória <variável>

Caso <valorX> então faça:

Caso <valorY> então faça:

Caso **contrário** faça:

- Repetições:

Faça:

Enquanto <Condição>

- Repetições:

Enquanto <Condição>:

Fim.

4 Conceitos básicos sobre eletrônica

5 Conceitos básicos sobre o Arduino

5.1 O que é Arduino



Figura 2: Logo Arduino

- O que é Arduíno? O Arduíno é dividido em duas partes

1. Hardware (Componentes elétricos)
2. Software (IDE)

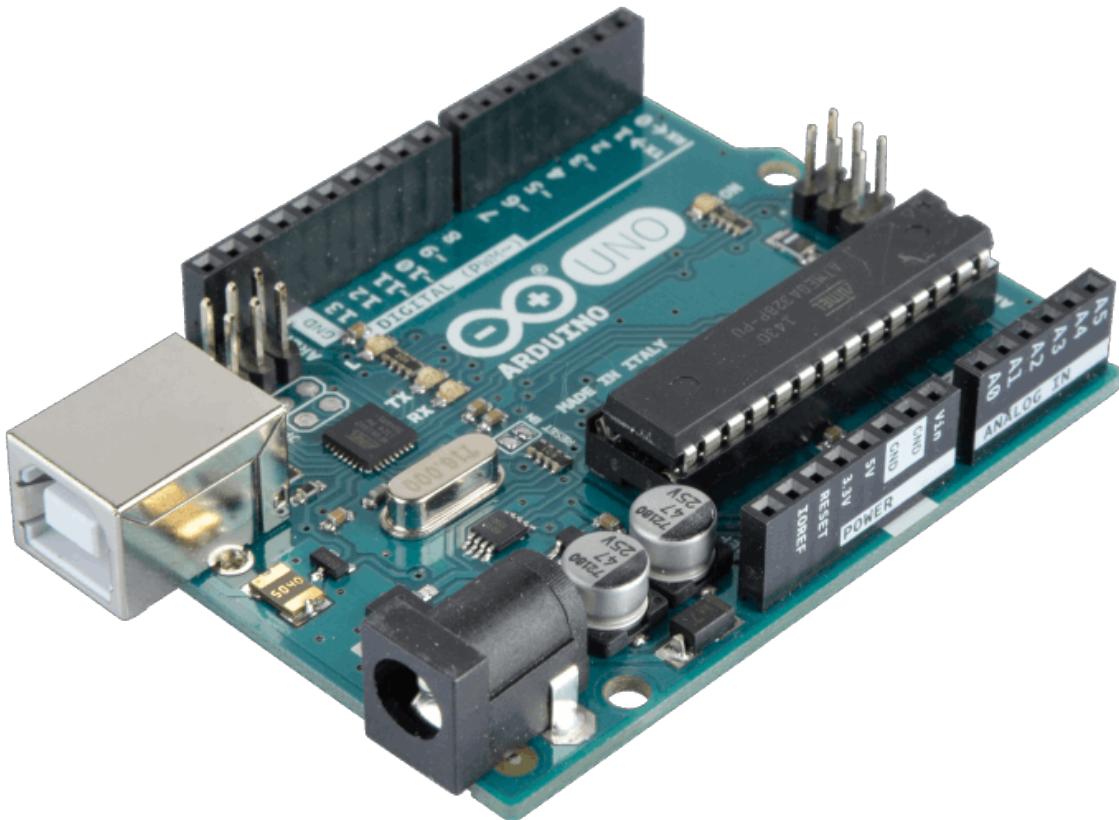


Figura 3: Placa do Arduíno

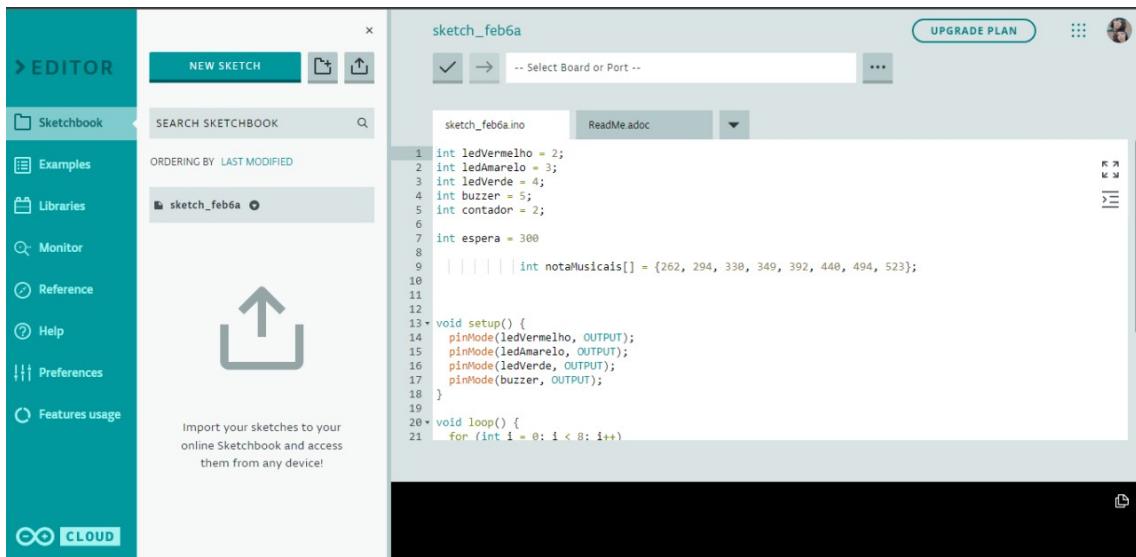


Figura 4: Tela inicial da IDE online do Arduíno

5.2 Instalação do Arduíno

Instalação da IDE no computador

- **Windows:** Entrar no site arduino.cc => Fazer o download do "instalador"=> Executá-lo
- **Linux:** Linha de comando no terminal: sudo apt-get install arduino

5.3 História

- Criando em 2005 por um grupo de italianos
 1. Massimo Banzi
 2. David Cuartielles
 3. Tom Igoe
 4. Gianluca Martino
 5. David Mellis
- Tem o conceito de open source (hardware livre)
- Programado em C/C++

5.4 Para que serve

Há uma possibilidade infinita, podemos automatizar casas, carro, escritórios, criar brinquedos, equipamentos ou até mesmo melhorar um existente.

5.4.1 Tarefa

- **Tarefa 01:** Pesquise sobre projetos com Arduino, escreva e comente o projeto que mais gostou.
- **Tarefa 02:** Com base nas suas pesquisas, elas lhe deram alguma ideia de projeto? Se sim, comente sobre como vai ser.

5.5 Tipos de Arduino

Os Arduinos são divididos em modelos de placas, sendo elas

- Arduino Uno R3, Uno SMD;
- Arduino Mega 2560, Nano, Micro, LilyPad, Mini;
- Arduino DUE, Duemilanove;
- Arduino ADK

Além das incontáveis versões compatíveis desenvolvidas pelos mais diversos fabricantes ao redor do mundo.

O que muda de um Arduino para o outro é o microcontrolador e a quantidade de pinos disponíveis.

5.6 Componentes do Arduino

1. **Microcontrolador Atmel ("Cérebro do Arduino"):** Controle de entrada e saída, armazenamento e execução do código.
2. **Porta USB:** Energização e comunicação.
3. **Porta de alimentação:** Energização via fonte ou bateria (tensão: 7v a 12v).
4. **Regulador de voltagem:** Controla a voltagem de entrada.

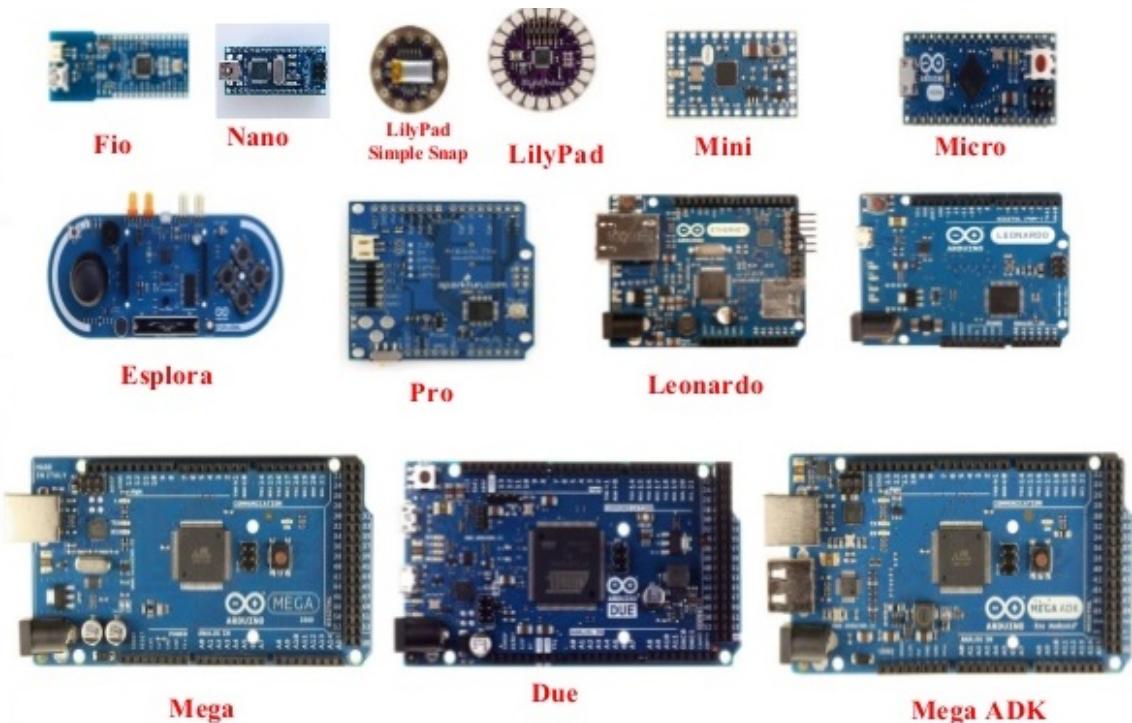


Figura 5: Alguns exemplos de Arduino

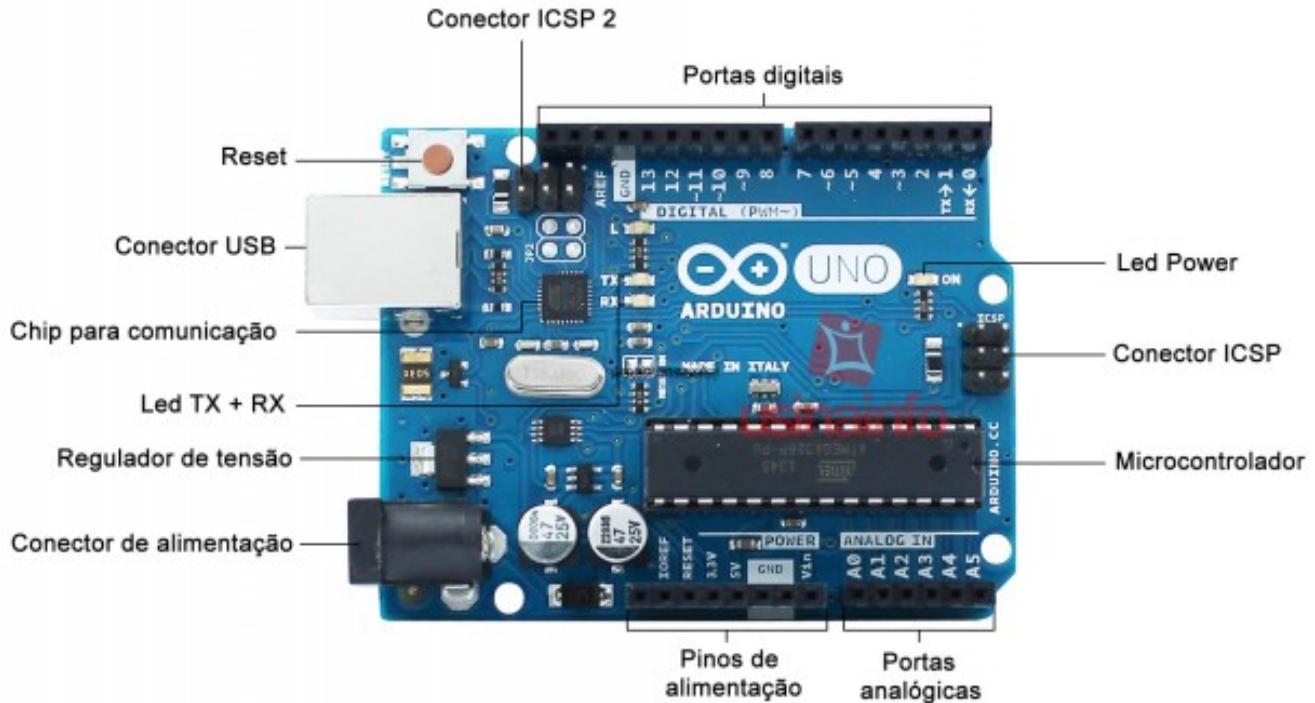


Figura 6: Componentes do Arduino

5. **Reset:** Reiniciar o Arduino.
6. **Shields:** Placa adicional para aumentar a funcionalidade do arduino. Bastante utilizado

em projetos é o shield ethernet, cuja função é permitir que a pessoa tenha acesso à rede através da conexão de um cabo entre o shield e o roteador.

7. Pinos de entradas e saídas digitais/analógicas: Habilitação dos resistores.

(a) Tipos de pinos

- i. **IOREF:** Fornece uma tensão de referência para que shields possam selecionar o tipo de interface apropriada, dessa forma shields que funcionam com a placas Arduino que são alimentadas com 3,3V podem se adaptar para ser utilizados em 5V e vice-versa.
 - ii. **RESET:** Pino conectado a pino de RESET do microcontrolador. Pode ser utilizado para um reset externo da placa Arduino.
 - iii. **3,3V:** Fornece tensão de 3,3V para alimentação de shield e módulos externos. Corrente máxima de 50mA.
 - iv. **5V:** Fornece tensão de 5V para alimentação de shields e circuitos externos.
 - v. **GND:** Pinos de referência, terra.
 - vi. **VIN:** Pino para alimentar a placa através de shield ou bateria externa. Quando a placa é alimentada através do conector Jack a tensão da fonte estará nesse pino.
8. **LEDs TX e RX:** Piscam rapidamente, o que indica que a comunicação e a transmissão do código foram estabelecidas.
 9. **Protoboard:** Também conhecida como matriz de contatos ou placa de prototipagem, a protoboard é uma placa que possui furos e conexões internas para montagem de circuitos, utilizada para testes com componentes eletrônicos. Sua maior vantagem de uso é que ele dispensa a necessidade de solda para conectar tais circuitos.
 10. **Sensores (módulos):** Os módulos são peças que podem aumentar o poder do Arduino, dando ao seu projeto maior robustez. Estas pequenas placas contém sensores, resistores, capacitores ou led's e são utilizadas para determinadas funções quando utilizada junto ao Arduino. Exemplos:

- Módulo microfone;
- Sensor Hall;
- Módulo laser;
- Buzzer;
- Módulo infravermelho;
- Módulo led;
- RGB LED;
- Módulo relé;
- Sensor magnético;
- Sensor de desvio de obstáculos;
- Magic ring;
- Sensor de temperatura

5.6.1 Tarefa

- **Tarefa 01:** Pesquise em que é usado os sensores(módulos) acima.

5.7 IDE

Integrated Development Environment ou conhecidamente IDE é a plataforma onde iremos construir os softwares, onde são escritos os códigos.

1. Principais funções do Arduino IDE

- **setup()** Esta função é responsável por definir os pinos de entrada e saída, bem como inicializar todo o arduino.
- **loop()** Função de repetição, onde os comandos serão construídos.

5.7.1 Vamos praticar!

Vamos construir passo a passo um circuito para acender a luz de um LED.

1. **Passo:** Vamos até o site Tinkercad.com. Caso não tenha um login, será necessário criar um.
2. **Passo:** Ao criar, no painel principal clique em "Circuitos", em seguida "Criar novo Circuito".
3. **Passo:** Vai abrir uma nova tela, onde você pode usar e abusar dela para descobrir novas coisas e aprender. Nesta tela vamos "pesquisar" e pesquisar por **Arduino uno r3** e depois **Placa de ensaio pequena**
4. **Passo:** Vamos configurar a barra de código. Neste mesmo painel tem o nome "Código" clica nela e abrirá outro painel. Clica em "Blocos" e irá aparecer novas opções, então coloca "Texto" e só confirmar.
5. **Passo:** Vamos pesquisar por uma **LED** e colocar na Placa de ensaio Pequena (Pode ser em qualquer lugar).
6. **Passo:** Vamos conectar os Jumpers. Perceba que no led tem uma perna torta (chamada de Anodo) significa perna positiva e uma reta (chamada de Catódica) significa que é negativa. No arduino vamos conectar um jumper do **5V** até o **Anodo** em qualquer conexão dela (**Por padrão Jumper tem que ser vermelho**). Agora no Catódico, vamos conectar o jumper **GND** do arduino até o catódico de qualquer conexão também (**Por padrão é na cor preta**).
7. **Passo:** Pesquisar por um **Resistor** e colocar uma resistência de **150**

6 Curso: Programação para Arduino

6.1 Relés, Potenciômetro, transistor como chave e regulador de tensão

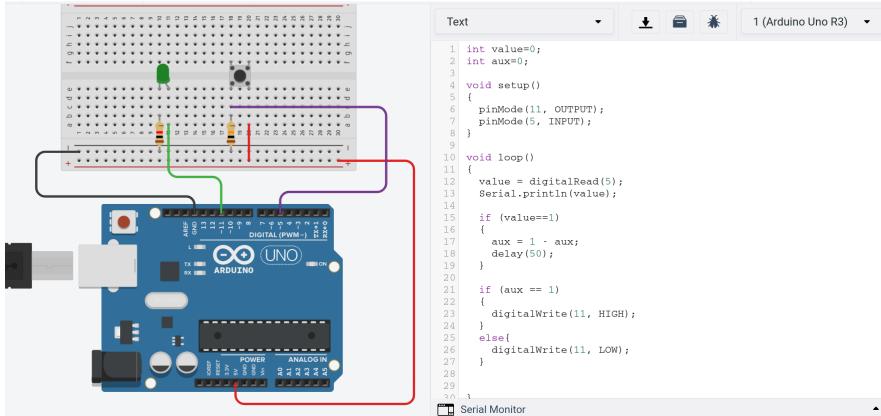
Push Button

Estes permitem enviar um sinal digital, 1, HIGH, ou 0, LOW a um circuito lógico, quer estejamos a falar de circuitos de lógica cabelada ou programada.

Ascender uma LED com o Push Button

Resistor do botão de Pressão: 10 KΩ

Modulo Relé



Basicamente, um relé é um “interruptor” acionado por uma determinada tensão. Ao se aplicar a tensão (no caso, 5V) nos terminais de entrada, uma bobina é acionada, ativando ou desativando o contato interno (ligando ou desligando uma chave).

O que faz o módulo relé tão interessante também é a facilidade de uso. Você tem apenas 3 pinos, sendo 2 de alimentação (Vcc e GND) e um de controle (S).

Conexão no Arduino físico

A conexão do módulo relé no Arduino deve ser feita conforme a imagem abaixo. O pino de controle do módulo relé será a porta digital 7, e a alimentação virá dos pinos 5V e GND do Arduino Uno:

- NC: está conduzindo corrente quando o módulo não está ativado.
- NO: ele não tem passagem de corrente enquanto o relé não for acionado.

Módulo Relé com dois relés

Conexão no Arduino físico

Programação no Arduino físico

Coneção modulo Relé - RU-5-R Modulo relé disponível no tinkercad:

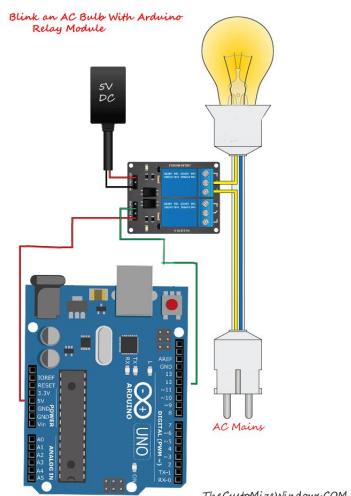
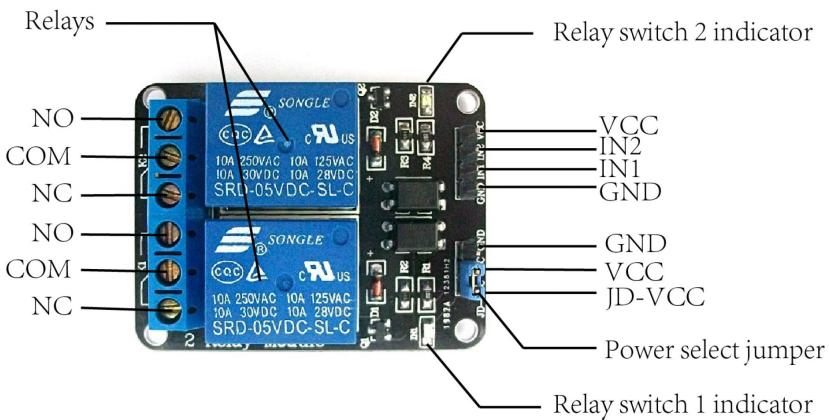
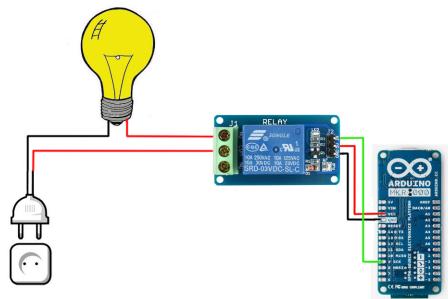
Coneção modulo Relé - RU-5-R

Programação Rele

Tarefa 1: Ascender lâmpada com o botão de pressão Vocês deverão construir um circuito e programar um botão de pressão para ascender uma lâmpada. Utilizem o modulo relé para isso.

Potenciômetro





```

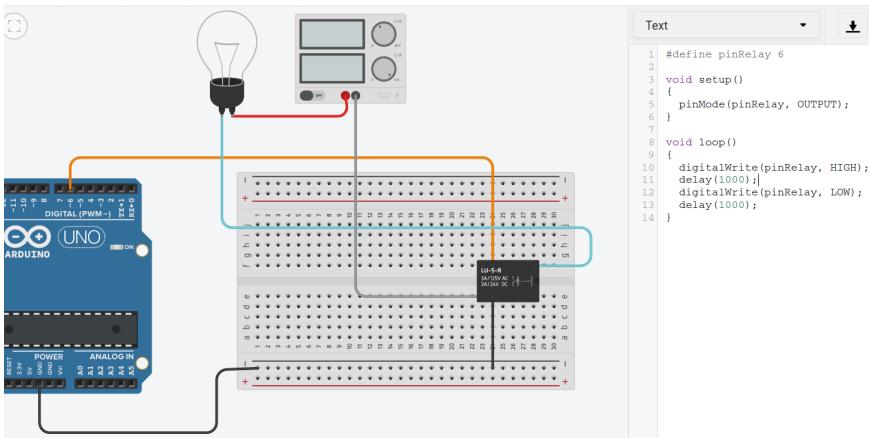
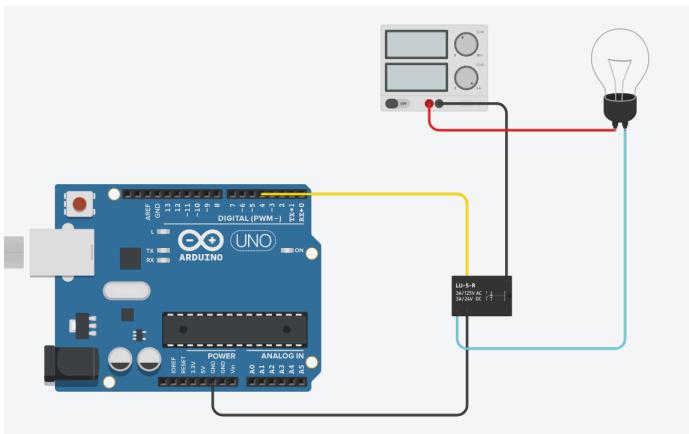
//define a saida a ser utilizada para o acionamento do rele
int sinalparaorele = 7;

void setup()
{
  pinMode(sinalparaorele, OUTPUT); //Define o pino como saida
}

void loop()
{
  digitalWrite(sinalparaorele, HIGH); //Aciona o rele
  delay(5000); //Aguarda 5 segundos
  digitalWrite(sinalparaorele, LOW); //Desliga o rele
  delay(5000); //Aguarda 5 segundos e reinicia o processo
}

```





O potenciômetro consiste em um elemento resistivo, chamado de “pista”, ou “trilha”, e de um cursor móvel, que se movimenta ao longo de um eixo, rotatório ou linear. De acordo com a posição desse cursor ao longo do eixo, a resistência obtida será diferente, dentro de certos limites característicos do componente em questão.

Ligaçāo do potenciômetro no Arduino

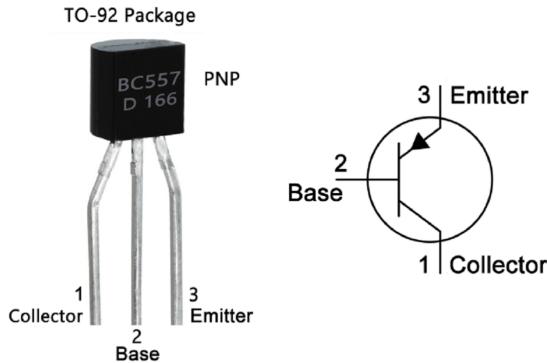
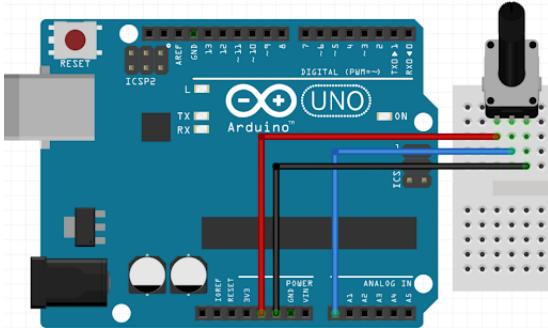
Ler valor do potenciômetro

- valor = analogRead(A0);
- Mudar o intervalo de leitura do potenciômetro: valorLido = map(valor, 0, 1023, 0, 255);
- Controlando brilho da LED (pinos PWM): analogWrite(pino, brilho);

Potenciômetro (Serial monitor)

- Para iniciar o serial monitor precisamos do seguinte comando (dentro da função void setup()): Serial.begin(9600);





- Imprimir mensagem no serial monitor: `Serial.print("mensagem");`
- Imprimir número inteiro no serial monitor: `Serial.println(numero);`
- Verificar se algo foi digitado no serial monitor: `Serial.available()` retorna valor 1
- Lendo um número a partir do serial monitor: `Serial.parseInt();`

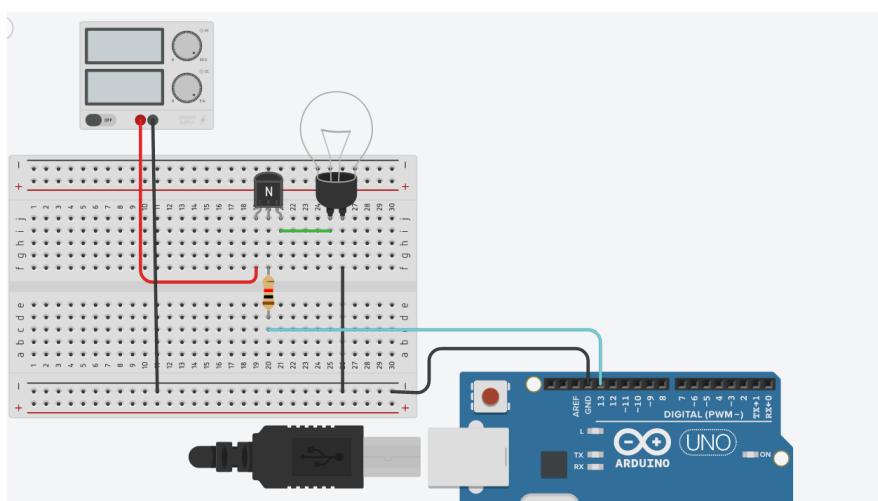
Tarefa 2: Piscar sequência de leds usando o potenciômetro

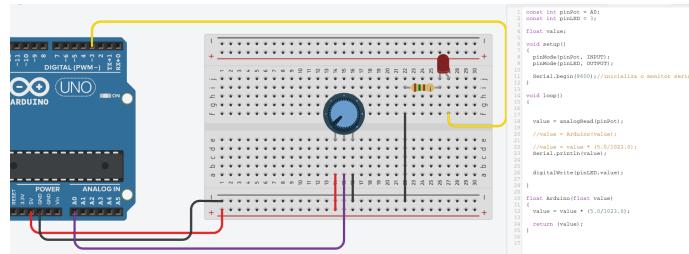
Vocês deverão construir um circuito e programação onde três LEDs piscarão, uma depois da outra, a velocidade com que piscam será alterada por um potenciômetro.

Transistor

Transistor (português europeu) ou transistor (português brasileiro) é um dispositivo semicondutor usado para amplificar ou trocar sinais eletrônicos e potência elétrica.

Transistor como chave





6.2 Display 7 Segmentos, Opto Acoplador, Display LCD 16×2

Regulador de tensão

Montagem no tinkercad do regulador de tensão

```
const int pinPot = A0; const int pinLED = 3;
float value;

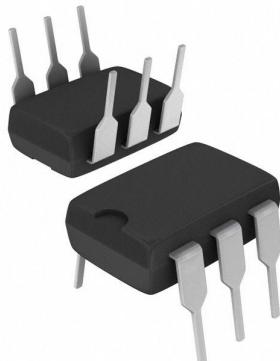
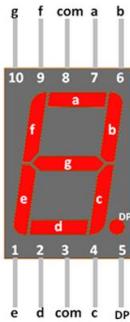
void setup() pinMode(pinPot, INPUT); pinMode(pinLED, OUTPUT);
Serial.begin(9600); // inicializa o monitor serial

void loop()
value = analogRead(pinPot);
//value = Arduino(value);
//value = value * (5.0/1023.0); Serial.println(value);
digitalWrite(pinLED,value);
float Arduino(float value) value = value * (5.0/1023.0);
return (value);
```

Display de 7 segmentos O Display de 7 Segmentos Arduino é um componente que não possui mais do que 8 LEDs e cada LED está separado por segmento que pode ser nomeado como a, b, c, d, e, f, g, DP conforme mostrado na imagem abaixo.

Anodo comum Display de 7 segmentos de ânodo comum (AC) – A exibição comum do ânodo é comumente chamada de exibição AC. Neste tipo, o pino comum no display de 7 segmentos é conectado a todos os oito pinos positivos dos LEDs. Portanto, para fazer esse tipo de display funcionar, devemos conectar o pino no Vcc (+ 5V) e aterrar o pino do segmento necessário para ligá-lo.

Conectando ao arduino Para começar, vamos conectar um dos pinos 3 ou 8 comuns ao pino 5V no Arduino (se você estiver usando um display de ânodo comum de 7 segmentos) ou ao pino GND no Arduino (se estiver usando um display de cátodo comum de 7 segmentos).



Os 4 pinos restantes na posição superior são conectados ao pino digital 2 ao pino digital 5. Os outros 4 pinos na posição inferior com ponto decimal são conectados ao pino digital 6 a 9.

Tarefa 1 Usando a atividade 1, sobre regulador de tensão, construa um circuito com o display de 7 segmentos de modo a mostrar o valor da tensão na tela.

Opto Acopladores 4N35

Os optoacopladores permitem que um circuito controle outro, mais ou menos como um transistor ou um relé também permitem.

Contudo, diferente desses dois, o acoplador permite evitar completamente o contato elétrico entre o circuito controlador e o circuito controlado.

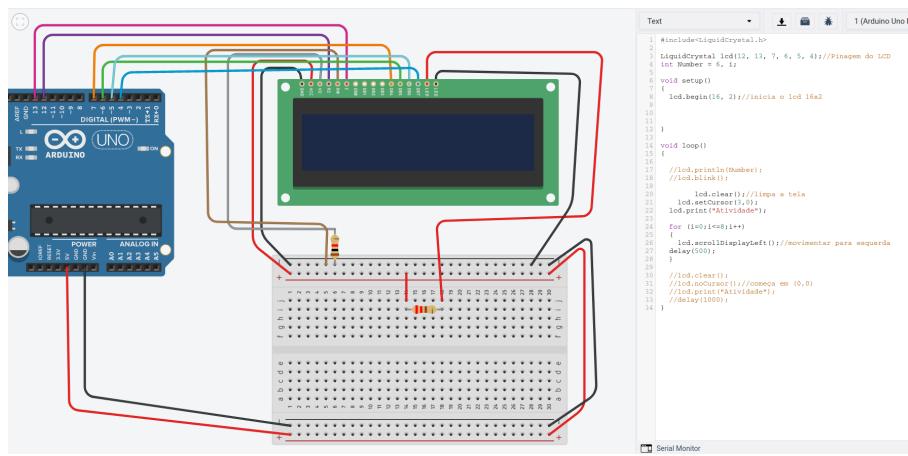
Display LCD 16×2

Módulos de display LCD (Liquid Crystal Display) de caracteres alfanuméricicos são interfaces de comunicação visual muito úteis e atraentes. Eles se encontram em quase todos os aparelhos domésticos, eletroeletrônicos, automóveis, instrumentos de medição etc.

Ligaçāo dos pinos do LCD ao Arduino

- O pino 1 (VSS) é ligado ao GND e o pino 2 (VDD) é ligado ao 5V. Utilize o GND e 5V do Arduino Uno.
- O pino 3 (V0) é responsável pelo ajuste do contraste do display, para isso coloque o resistor de 1K ligado ao GND, que será ajustado na medida certa para não saturar ou apagar as letras e números do display.
- O pino 4 (RS) é responsável pelo registrador de seleção do sinal, é ligado no pino 12 do Arduino Uno.





- O pino 5 (R/W) representa o sinal de escrita e leitura do display, é ligado ao GND.
- O pino 6 (E) habilita ou desabilita o sinal, é ligado no pino 13 do Arduino Uno.

Continuação da ligação dos pinos do LCD

- Os pinos de 7 ao 10 não são utilizados, pois o Data Bus será setado como 4 bits.
- O pino 11 (DB4) é ligado no pino 7 do Arduino Uno.
- O pino 12 (DB5) é ligado ao pino 6 do Arduino Uno.
- O pino 13 (DB6) é ligado ao pino 5 do Arduino Uno.
- O pino 14 (DB7) é ligado ao pino 4 do Arduino Uno.

Os pinos 15 (LED+) e 16 (LED-) são responsáveis por fornecer energia aos LEDs que ficam ao fundo do display para você enxergar o que está escrito nele. O pino 16 é ligado ao GND e o pino 15 ao vcc com um resistor de 220Ω.

Montagem do circuito e programação no tinkercad Algumas funções da biblioteca LiquidCrystal.h

- Adicionar a biblioteca (cabeçalho): [LiquidCrystal.h](#)
- Setar os pinos: `LiquidCrystal lcd(RS, E, DB4, DB5, DB6, DB7);`
- Inicializar o LCD: `lcd.begin(16,2);`
- Limpar a tela: `lcd.clear();`
- Setar onde a escrita irá aparecer: `lcd.setCursor(Coluna, Linha);`
- Deslizar escrita para a esquerda: `lcd.scrollDisplayLeft();`

Tarefa 2 Refazer a tarefa 1 com o display LCD.



#9400D3	RGB 148, 0, 211
#4B0082	RGB 75, 0, 130
#0000FF	RGB 0, 0, 255
#00FF00	RGB 0, 255, 0
#FFFF00	RGB 255, 255, 0
#FF7F00	RGB 255, 127, 0
#FF0000	RGB 255, 0, 0

6.3 LED RGB, Neopixel

6.3.1 LED RGB

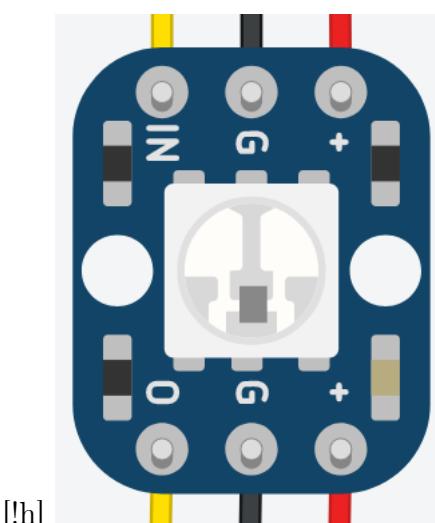
Tabela de cores

Atividade Faça um pequeno programa em que o usuário entrará com a cor a ser exibida pela LED RGB dentre as opções oferecidas (Roxo, Laranja e Amarelo).

Neo Pixel

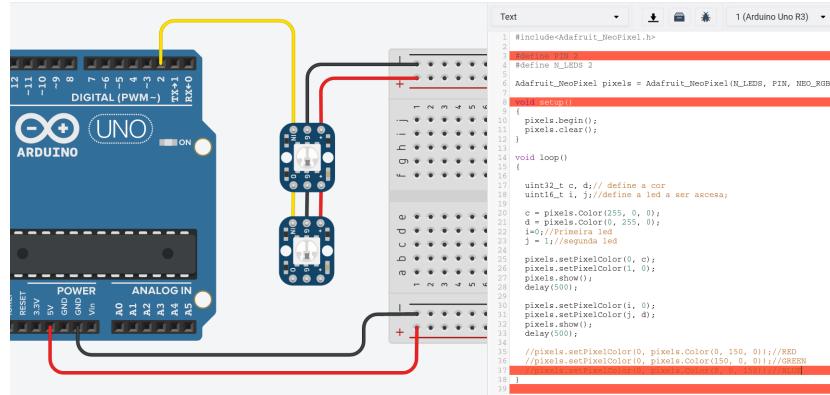
Alguns detalhes sobre a porgramação do NeoPixel

- Incluir a biblioteca: `Adafruit_NeoPixel.h`
- Inicializar a structure do NeoPixel: `Adafruit_NeoPixel(N_LEDs, PIN, NEO_RGB + NEO_KHZ800);`

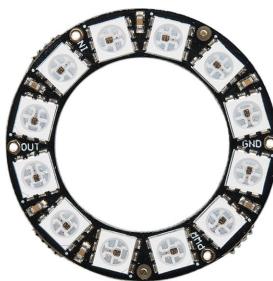


- Habilita o uso da biblioteca da LED: `pixels.begin();`
- Função que define a cor: `pixels.Color(255, 0, 0);`
- Função que define a LED: `setPixelColor(Numero, pixels.Color(255, 0, 0));`
- Ascender a LED: `pixels.show();`

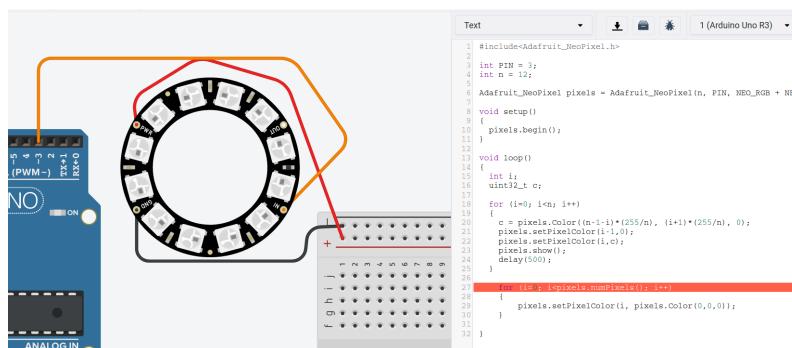
Circuito e programação do NeoPixel



Neo Pixel - Ring



Circuito e programação - Ring



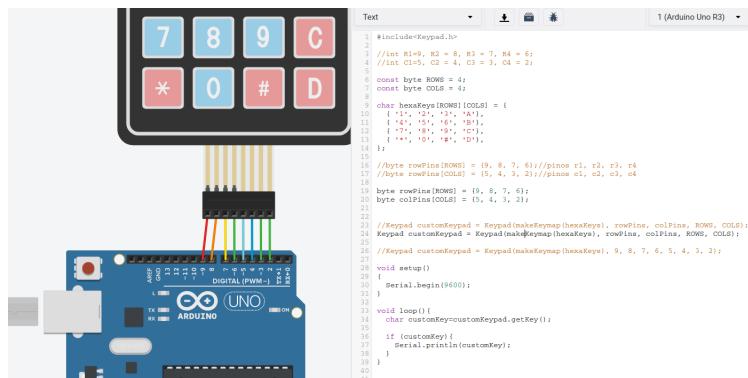
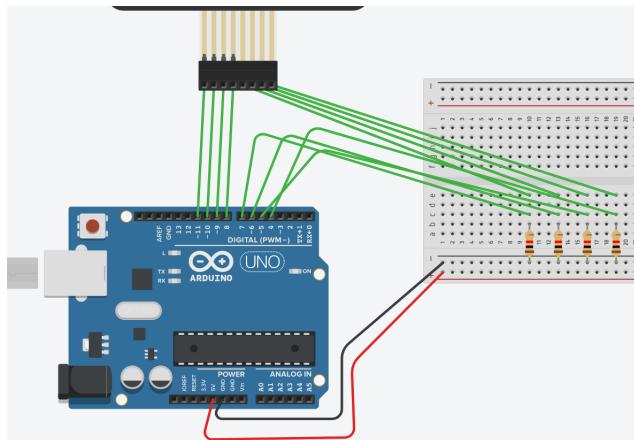
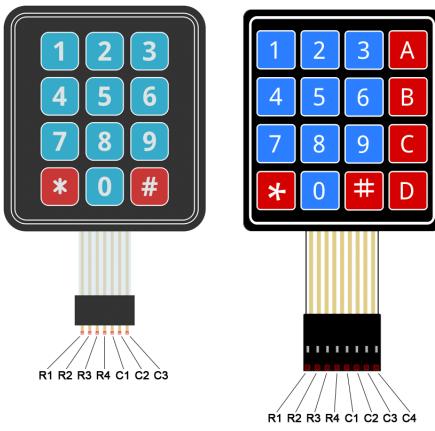
Tarefa Crie três efeitos diferentes de LEDs tais que, estes efeitos mudem quando um botão de pressão for pressionado.

6.4 Keypad e Dips switch

Keypads

Ligaçāo do circuito no Keypad

Atividade SENHA (durante a aula) Crie um programa que necessite digitar uma senha com o Keypad.



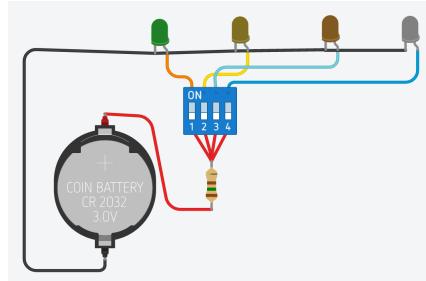
Ligaçāo do circuito no Keypad (Usando biblioteca)

Dips Switch

DIP Switch - Primeira ligação

Atividade - Converter binário pra decimal

Vocēs deverāo criar um código, no qual ao entrar com o númerō binário por meio do DIP Switch esse númerō seja convertido para a base decimal.



Especificações	TMP35	TMP36	TMP37
Temperatura de trabalho	10°C até 125°C	-40°C até 125°C	5°C até 100°C
Linearidade	10 mV/°F	10 mV/K	20 mV/°C
Tensão de saída a 25°C	250 mV	750 mV	500 mV

6.5 Sensor de temperatura, luminosidade, PIR, IR

Sensor de Temperatura - TMP36

Este Sensor de Temperatura TMP36 possui alta precisão e funciona na faixa de 2.7V a 5.5VDC. Além disso, o sensor fornece uma saída de tensão linearmente proporcional a temperatura em graus celsius e o mesmo não necessita de calibração externa para fornecer uma leitura de -40° a 125°C, com precisão de $\pm 2^{\circ}\text{C}$.



Funcionalidades do sensor TMP36 Já as principais funcionalidades do sensor TMP 36, se destacam as seguintes:

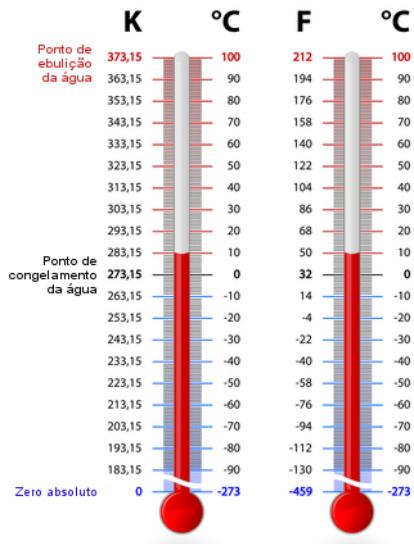
- Possuem uma tensão de baixa operação (2.7V até 5.5V);
- Trabalham na faixa de -40°C até 125°C e operam até no máximo 150°C;
- Não esquentam tão facilmente;
- São estáveis a grandes cargas capacitivas aplicadas;
- Não precisam de calibração para funcionar e são componentes que já fornecem valores na escala Celsius;
- São qualificados para serem utilizados em automóveis.

Especificações

Atividade - Conversão de temperatura Implemente a temperatura nas escalas Kelvin e Fahrenheit.

Sensor de Luminosidade (Fotoresistor)

Light dependent resistors, LDRs, or photoresistors are electronic components that are used to detect light and change the operation of a circuit dependent upon the light levels.



A photoresistor or light dependent resistor is an electronic component that is sensitive to light. When light falls upon it, then the resistance changes. Values of the resistance of the LDR may change over many orders of magnitude the value of the resistance falling as the level of light increases.

Utilidade de um LDR Com o LDR pode-se fazer o controle automático de porta, alarme contra ladrão, controle de iluminação em um recinto, contagem industrial, todos estes foto-controlados para a operação de um relé.



(a) Porta automática

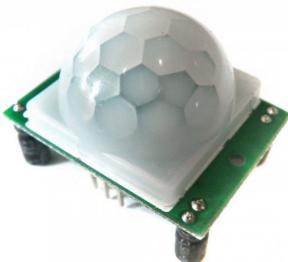
(b) Iluminação automática

Atividade em aula Use o fotoresistor ldr para ascender e apagar uma lampada (dependendo da necessidade). Para isso, utilize o module rele (visto em aulas passadas) para uso de uma fonte externa.



Sensor PIR - detector de movimento O Sensor PIR (Detector) de Movimento é capaz de

detectar movimento de objetos que exalam calor e que estejam dentro do seu raio de detecção que alcança até 7 metros. Com o sensor atuando, qualquer objeto (que exala calor) que se movimentar dentro do seu campo de detecção, fará com que a saída do mesmo seja ativada.



Especificações – Especificações e características:

- Tensão de operação: 4,5 a 20VDC
- Tensão (Nível alto): 3,3V
- Tensão (Nível baixo): 0V
- Raio de detecção: 3 – 7m (pode ser ajustado)
- Ângulo de alcance: 100°
- Potenciômetros para ajustes de sensibilidade e estabilização do piroelétrico
- Temperatura de operação: -20° a 80° celsius

Buzzer

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.



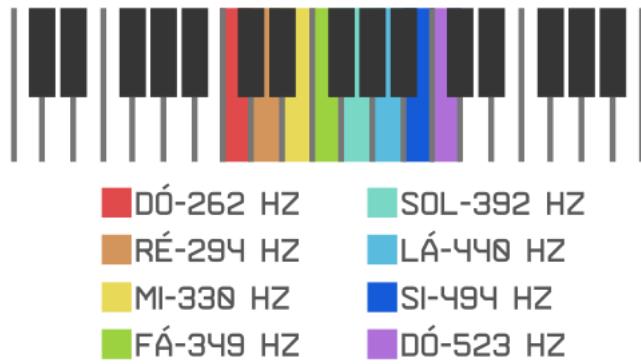
Tabela de frequências

Informações sobre frequências

A frequência é a característica através da qual o ouvido distingue se um som é agudo ou grave. Esta característica está relacionada com a quantidade de ciclos (vibrações) de um onda sonora em um período de um segundo, e é expressa em Hertz (Hz).

O espectro de frequências que o ouvido humano pode entender engloba sons entre 20 Hz e 20.000 Hz ou 20 kHz (1 kHz = 1 Kiloherz ou 1000 Hz).

Aplicando um sinal elétrico em uma determinada frequência, o buzzer produz uma nota musical. As notas variam conforme a frequência utilizada.



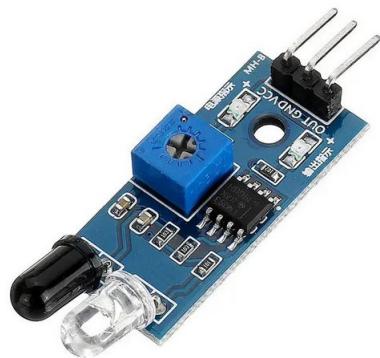
O buzzer é um dispositivo piezoelétrico de sinalização. Você pode usá-lo de diversas maneiras no seu projeto, por exemplo, um clique, um beep ou som curto podem indicar, por exemplo, que um botão foi pressionado, ou que um sensor foi ativado.

Atividade Acionar um alarme quando houver movimento.



Sensor IR

O sensor de obstáculo é um circuito composto por um emissor e um receptor IR. Seu funcionamento é simples: quando algum obstáculo é colocado em frente ao sensor, o sinal infravermelho é refletido para o receptor. Quando isso acontece, o pino de saída OUT é colocado em nível baixo (0), e o led verde do módulo é aceso, indicando que algum obstáculo foi detectado.



Funcionamento do sensor

IR (Infrared) sensor do Arduino

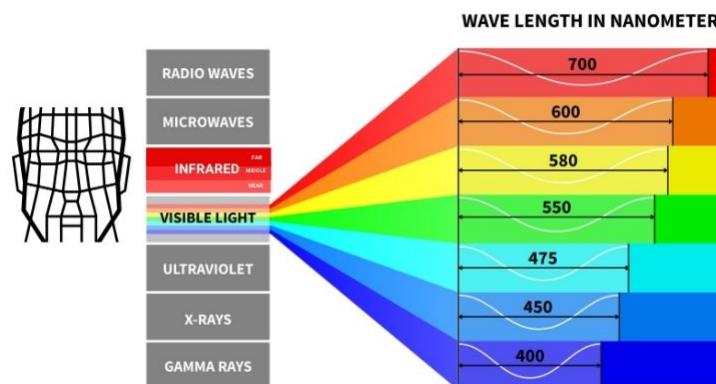
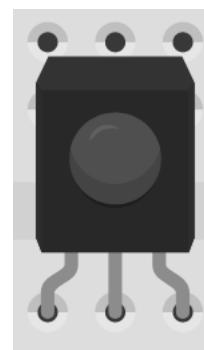
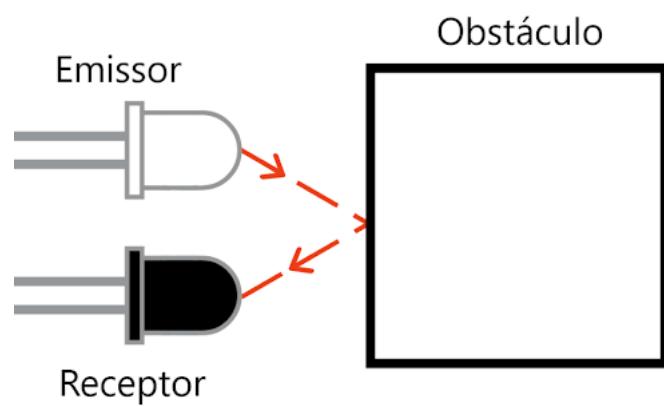
Funcionamento

Círcuito e código IR sensor

Atividade em aula Utilizar o botão de volume para aumentar e diminuir o brilho de uma LED.

Códigos:

- Vol+: FD807F
- Vol-: FD906F



```

1 //include<IRremote.h>
2 int RECV_PIN = 2;
3 #include<IRrecv.h>
4 IRrecv ir(IRrecv_230);
5 decode_results results; //armazenar o sinal
6 void setup()
7 {
8     Serial.begin(9600);
9     ir.enableIRIn();
10 }
11 void loop()
12 {
13     if (IR.decode(results))
14     {
15         Serial.println(results.value, HEX);
16         switch(results.value)
17         {
18             case 0x000000://000
19                 Vol = Vol + 10;
20                 break;
21             case 0x000001://001
22                 Vol = Vol - 10;
23                 break;
24             default:
25                 Serial.print("Just for volume \n");
26                 break;
27         }
28         IR.resume(); //receive the next value
29         delay(1000);
30         analogWrite(pinLED, Vol);
31     }
32     if (IR.decode(results) == 0)
33     {
34         Serial.println(results.value, HEX);
35         IR.resume(); //res para o proximo
36         delay(1000);
37     }
38 }

```

Atividade - Tarefa Utilizar os botões de passar para frente e tras para controlar qual LED, dentre 3 LEDs, deverá ser acesa.

Códigos:

- tras: FD20DF
- frente: FD609F

6.6 Sensor de distância, Tilt, gás

6.7 Motor DC, Servo motor, Higrômetro

6.7.1 Motor DC

A sua comutação (troca de energia entre rotor e estator) pode se dar através de escovas (escovado) ou sem escovas (brushless) e com relação a velocidade, o motor DC pode ser controlado apenas variando a sua tensão, diferentemente de um motor elétrico de corrente alternada (CA) cuja velocidade é variada pela frequência. Na plataforma *Tinkercad*, também podemos utilizá-lo para substituir algum componente que não esteja disponível na plataforma, como é o caso da Mini Bomba de Água, que foi criada especialmente para o desenvolvimento de projetos de prototipagem, incluindo automação residencial (doméstica) e protótipos robóticos baseados em plataformas microcontroladores, entre elas, o Arduino.



Figura 7: Motor DC.

A montagem do motor DC com o microcontrolador Arduino pode ser vista na Figura 8.

6.7.2 Servo motor

Um servo motor é um dispositivo pequeno que possui um eixo de saída. Este eixo pode ser posicionado em posições angulares específicas, enviando ao servo um sinal codificado. Enquanto o sinal codificado existir na linha de entrada, o servo manterá a posição angular do eixo. Se o sinal codificado muda, a posição angular do eixo muda.

A montagem do circuito e a programação do servo motor podem ser vistas na Figura 10.

6.7.3 Higrômetro - Sensor de umidade do solo

O sensor de umidade do solo, apresentado na Figura 11, consiste em duas sondas que são usadas para medir o conteúdo volumétrico da água presente no solo. As duas sondas permitem que a corrente passe pelo solo e, em seguida, obtenha o valor da resistência para medir o valor da umidade. Quando houver mais água, o solo conduzirá mais eletricidade, o que significa

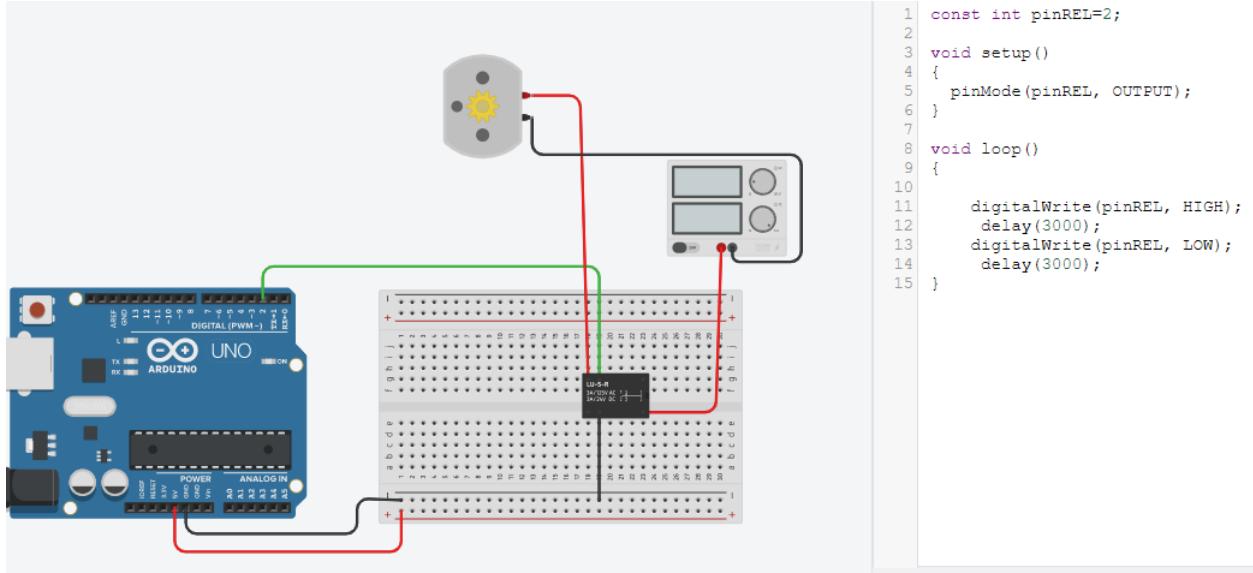


Figura 8: Circuito e programação do motor DC.



Figura 9: Servo motor

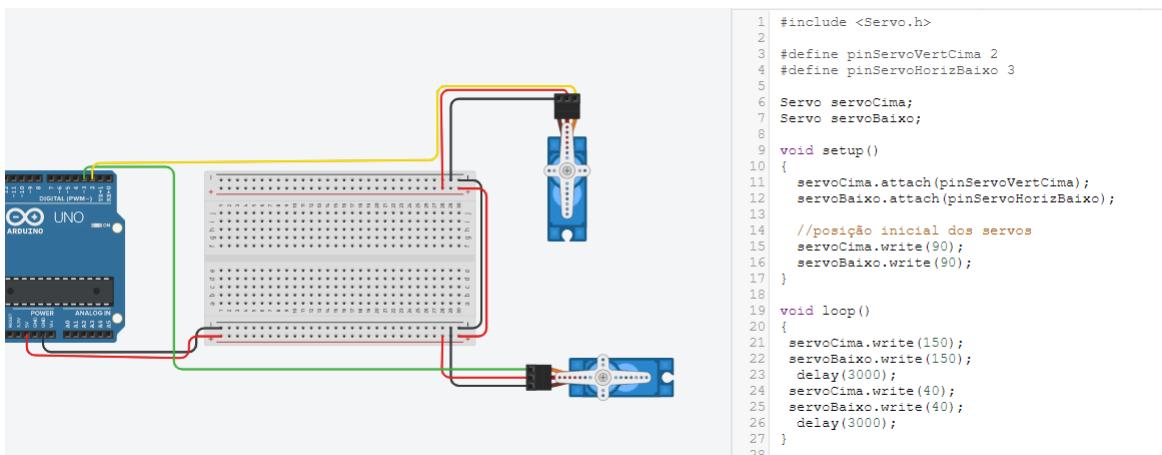


Figura 10: Circuito e programação do servo motor.

que haverá menos resistência. Portanto, o nível de umidade será maior. O solo seco conduz a eletricidade, portanto, quando houver menos água, o solo conduzirá menos eletricidade, o que significa que haverá mais resistência. Portanto, o nível de umidade será menor.

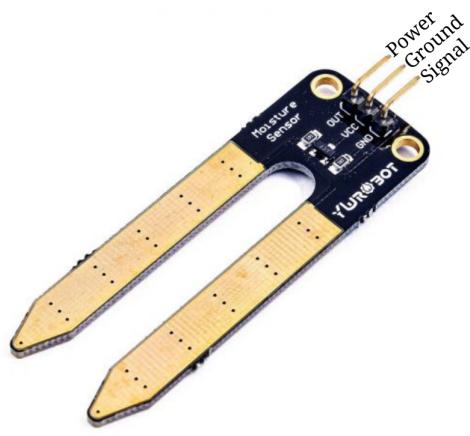


Figura 11: Higrômetro.

A montagem do circuito (disponível no tinkercad) e programação são apresentadas nas Figuras 12 e 13, respectivamente.

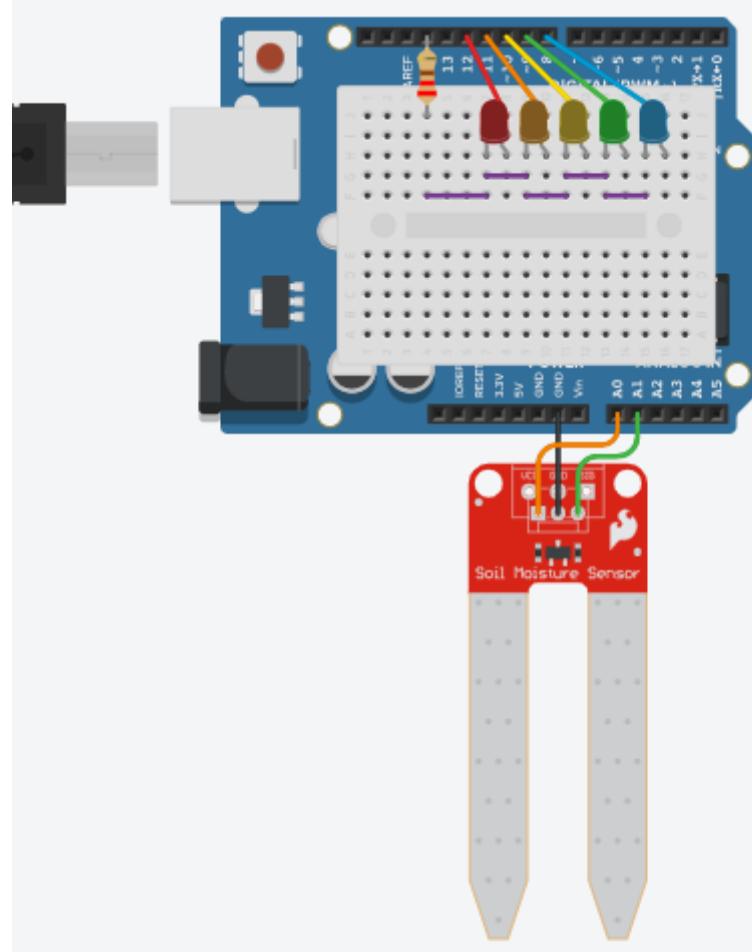


Figura 12: Circuito do higrômetro.

```

1 int pinUmidade = A0;
2
3 void setup()
4 {
5     pinMode(A0, OUTPUT);
6     pinMode(A1, INPUT);
7     Serial.begin(9600);
8     pinMode(8, OUTPUT);
9     pinMode(9, OUTPUT);
10    pinMode(10, OUTPUT);
11    pinMode(11, OUTPUT);
12    pinMode(12, OUTPUT);
13 }
14
15 void loop()
16 {
17     //Aplica energia ao sensor de umidade do solo
18     digitalWrite(A0, HIGH);
19     delay(10); //Aguarde 10 milissegundos
20     pinUmidade = analogRead(A1);
21     // Desligue o sensor para reduzir a corrosão do metal
22     // hora extra
23
24     digitalWrite(A0, LOW);
25     Serial.println(pinUmidade);
26     digitalWrite(8, LOW);
27     digitalWrite(9, LOW);
28     digitalWrite(10, LOW);
29     digitalWrite(11, LOW);
30     digitalWrite(12, LOW);
31     if (pinUmidade < 200) {
32         digitalWrite(12, HIGH);
33     } else {
34         if (pinUmidade < 400) {
35             digitalWrite(11, HIGH);
36         } else {
37             if (pinUmidade < 600) {
38                 digitalWrite(10, HIGH);
39             } else {
40                 if (pinUmidade < 800) {
41                     digitalWrite(9, HIGH);
42                 } else {
43                     digitalWrite(8, HIGH);
44                 }
45             }
46         }
47     }
48     delay(100);
49 }

```

Figura 13: Programação do higrômetro no editor IDE.

6.7.4 Atividade - estudo de caso

Suponha que você foi contratado pela empresa XXZ: Frente à necessidade global de economia de água e à falta de um funcionário dedicado à irrigação da plantação que tem sofrido com tempos de estiagem, o Sr Carlos, que é um produtor rural bastante conhecido em sua região e já atua nesse setor há mais 20 anos, sempre desempenhou um trabalho manual em sua propriedade.

Mesmo sem saber muito sobre tecnologia ele procurou a empresa XXZ solicitando um processo automático que possa diminuir o gasto de água e aumentar a produção de sua plantação em sua propriedade, onde o processo de irrigação só aconteça quando houver necessidade, com sua experiência, ele também não ver necessidade para que o sistema continue atuando durante o período noturno. Após o Sr Carlos ter explicado sua situação para a empresa, o gestor escolheu você para desenvolver esse projeto.

Uma das possíveis soluções do estudo de caso é ...circuito Figura 14 ... Figuras 15, 16, 17...

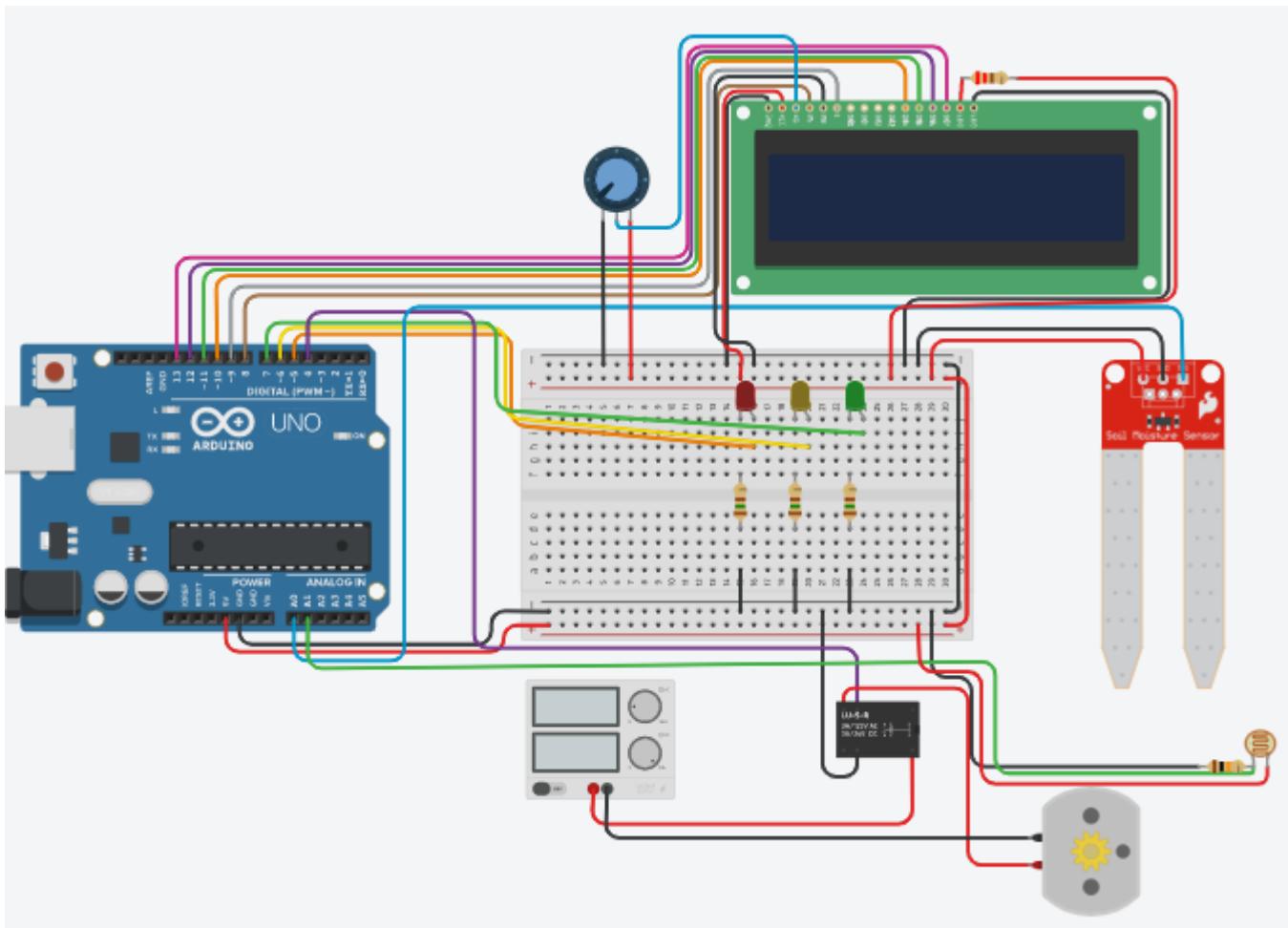


Figura 14: Circuito do estudo de caso.

6.8 Projeto final

7 Curso: Programação para Arduino 2

8 Considerações finais

```

1 #include<LiquidCrystal.h>
2 LiquidCrystal lcd (8,9,10,11,12,13);
3
4 const int pinFotoRes=A1;
5
6 int leituraSensor;
7 int leituraLUZ;
8
9 const int PinSensor=A0;
10 const int PinRele=4;
11 const int pinLvermelho=5;
12 const int PinLamarelo=6;
13 const int PinLverde=7;
14 void setup() {
15
16 lcd.begin(16,2);
17
18 //Sensor
19 pinMode(PinSensor, INPUT);
20 //Atuador
21 pinMode(PinRele, OUTPUT);
22 //Foto Resistor
23 pinMode(pinFotoRes, INPUT);
24 //LEDs
25 pinMode(pinLvermelho, OUTPUT);
26 pinMode(PinLamarelo, OUTPUT);
27 pinMode(PinLverde, OUTPUT);
28
29 Serial.begin(9600);
30 }
31 void loop() {
32
33 int seco=80; //aciona
34 int ideal=750; //Desliga
35 leituraSensor = analogRead(PinSensor);
36
37 int noite=650;
38 leituraLUZ=analogRead(pinFotoRes);

```

Figura 15: Programação parte 1.

```

40 // o sistema de irrigação só funcionara durante o dia.
41 if(leituraLUZ > noite ){
42 digitalWrite(pinLvermelho, LOW);
43 digitalWrite(PinLamarelo, LOW);
44 digitalWrite(PinLverde, LOW);
45 digitalWrite(PinRele, LOW);
46 analogWrite(PinSensor, LOW);
47 }else{
48
49 if (leituraSensor <= seco) {
50 //No estado seco
51 digitalWrite(pinLvermelho, HIGH);
52 digitalWrite(PinLamarelo, LOW);
53 digitalWrite(PinLverde, LOW);
54 digitalWrite(PinRele, HIGH);
55
56 lcd.setCursor(3,0);
57 lcd.print("Solo seco:");
58 delay(2000);
59 lcd.clear(); //apaga
60
61 lcd.setCursor(1,0);
62 lcd.print("Bomba acionada.");
63 delay(2000);
64 lcd.clear();
65
66 } else if (leituraSensor > seco && leituraSensor < ideal) {
67 //No estado de atuação
68 digitalWrite(pinLvermelho, LOW);
69 digitalWrite(PinLamarelo, HIGH);
70 digitalWrite(PinLverde, LOW);
71 digitalWrite(PinRele, HIGH);
72
73 lcd.setCursor(1,0);
74 lcd.print("Solo umidecido:");
75 delay(2000);
76 lcd.clear();
77

```

Figura 16: Programação parte 2.

```

78 lcd.setCursor(4,0);
79 lcd.print("Bomba em");
80 lcd.setCursor(4,1);
81 lcd.print("Atuacao.");
82 delay(2000);
83 lcd.clear();
84
85 } else if (leituraSensor >= ideal) {
86 //No estado ideal, desliga a bomba
87 digitalWrite(pinLvermelho, LOW);
88 digitalWrite(PinLamarelo, LOW);
89 digitalWrite(PinLverde, HIGH);
90 digitalWrite(PinRele, LOW);
91
92 lcd.setCursor(3,0);
93 lcd.print("Solo umido:");
94 delay(2000);
95 lcd.clear();
96
97 lcd.setCursor(0,0);
98 lcd.print("Bomba desligada.");
99 delay(2000);
100 lcd.clear();
101 }
102 }
103
104 Serial.println(analogRead (PinSensor));
105 }
106

```

Figura 17: Programação parte 3.