
MINISTÉRIO DA EDUCAÇÃO – MEC
SECRETARIA DE EDUCAÇÃO SUPERIOR – SESU
PROGRAMA DE EDUCAÇÃO TUTORIAL – PET

UNIVERSIDADE FEDERAL FLUMINENSE – UFF
ESCOLA DE ENGENHARIA – TCE
GRUPO PET DO CURSO DE ENG. DE TELECOMUNICAÇÕES – PET-TELE

Introdução ao *kit* de desenvolvimento Arduino

Questionários de aulas

(Versão: A2022M12D22)

Manutenção: Pedro Henryque Barbosa da Silva e Kriz
Alexandre Santos de la Vega
Lúcio Folly Sanches Zebendo
Lucas Pontes Siqueira

Autores: Lorraine de Miranda Paiva
Roberto Brauer Di Renna
Thiago Elias Bittencourt Cunha

Tutor: Alexandre Santos de la Vega

Niterói – RJ
Dezembro / 2022

Sumário

1	Arduino e <i>protoboard</i> .	2
2	Programação básica.	3
3	Interação com LED comum.	4
4	Interação com botões e LEDs RGB.	5
5	<i>Serial Monitor</i> e interação com teclado.	6
6	Funções sonoras.	7
7	Funções de temporização e de comunicação com o <i>Serial Monitor</i> .	8
8	Sensor de luminosidade e função <i>analog.Read()</i> .	9
9	Sensor de temperatura e funções matemáticas.	10
10	Sensor de movimento PIR e alarme com <i>buzzer</i> .	11
11	Comunicação IR e controle remoto IR.	12
12	<i>Display</i> de cristal líquido (LCD 16x2).	13
13	Uso de circuito integrado externo.	14
14	Comunicação RF.	15
15	Jogo Genius.	16
16	Interação com <i>webpage</i>	17

1 Arduino e *protoboard*.

Data:
Aluno:
Aluno:

1. Desenhe, na Figura 1, a organização das trilhas da *protoboard*.

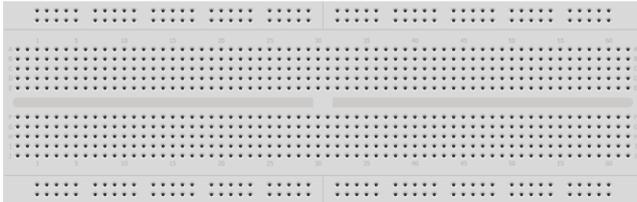


Figura 1: *Protoboard*.

2. Preencha a Tabela 1, identificando cada elemento presente na placa do Arduino UNO, ilustrado na Figura 2.

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	

Tabela 1: Elementos do Arduino.

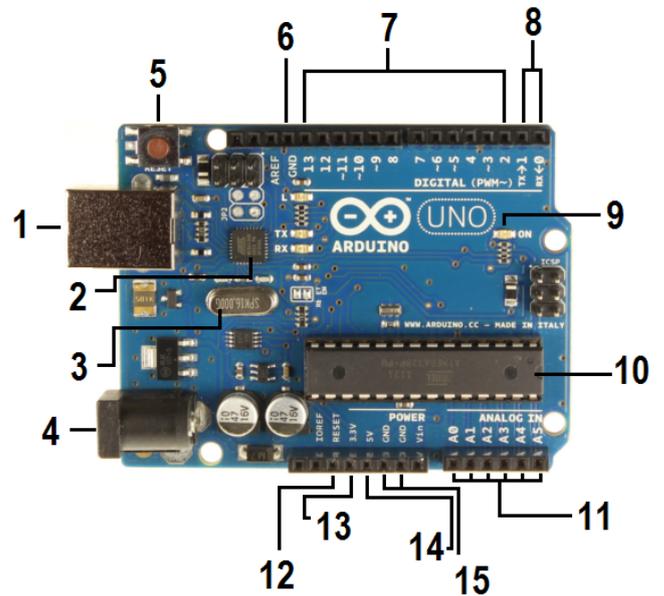


Figura 2: Arduino UNO.

2 Programação básica.

Data:

Aluno:

Aluno:

1. A linguagem de alto nível utilizada para programar o Arduino é similar à seguinte linguagem de programação: _____.
2. Os três elementos que compõem a estrutura geral de um código Arduino são os seguintes: _____, _____ e _____.
3. Descreva, brevemente, a função de cada um dos elementos que compõem a estrutura geral de um código Arduino.
4. No controle do fluxo da execução de um programa, as estruturas mais comumente utilizadas para alterar o fluxo linear são as seguintes: _____ e _____.
5. Apresente a sintaxe das estruturas mais comumente utilizadas para alterar o fluxo linear de execução de um programa.

3 Interação com LED comum.

Data:
Aluno:
Aluno:

1. Se, na função *pinMode()*, um pino for identificado como _____, podemos ler sua tensão como _____ para 5 V ou 3,3 V ou _____ para 0 V ou GND.

Porém, se o pino for identificado como _____, podemos ler os valores do nosso componente.

2. A Figura 3 ilustra um circuito para acionamento de um LED. Analise o circuito e comente o seu funcionamento.

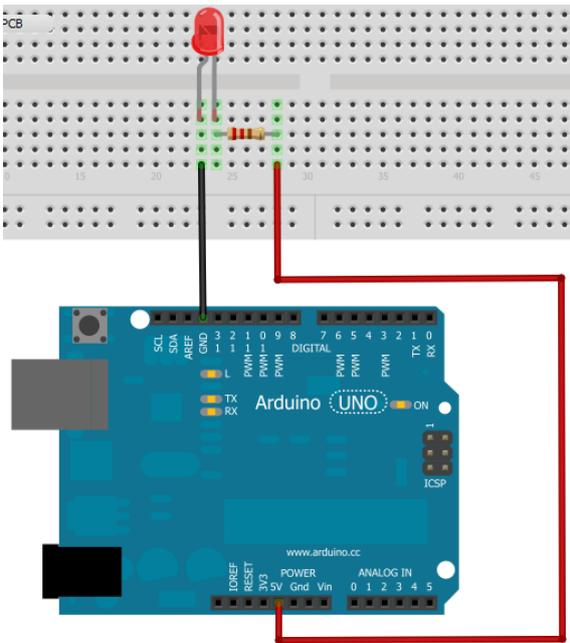


Figura 3: Circuito para acionamento de um LED.

3. A Figura 4 mostra um LED em detalhe. Complete a figura, indicando a sua polarização direta.

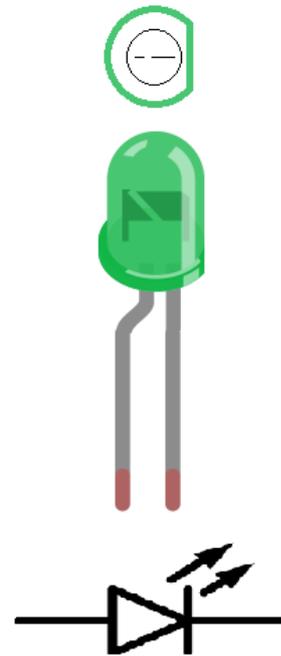


Figura 4: Polarização de um LED.

4 Interação com botões e LEDs RGB.

Data:
Aluno:
Aluno:

1. Nesta aula, além de usarmos a função **digitalWrite()**, também usamos a função _____ para variar a intensidade luminosa dos LEDs _____. Essa função só opera corretamente quando usamos as saídas _____. A função _____ escreve um valor de _____ para o pino digital, que pode variar de _____ a _____. Quanto mais alto o valor escrito, maior a intensidade luminosa do LED.
2. A Figura 5 mostra um LED RGB em detalhe. Identifique a sua pinagem.

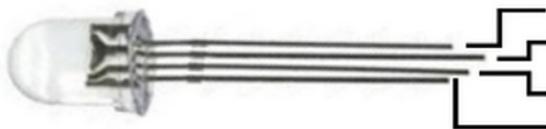


Figura 5: LED RGB.

3. No início do bloco de código, da função `void loop ()`, aparece o seguinte trecho:

```
estadoBotao_1 = digitalRead(botao_1);  
estadoBotao_2 = digitalRead(botao_2);  
estadoBotao_3 = digitalRead(botao_3);
```

Descreva o funcionamento da função **digitalRead()**, usando esse trecho como exemplo.

5 *Serial Monitor* e interação com teclado.

Data:
Aluno:
Aluno:

Na prática de hoje, acendemos um LED a partir do teclado. Proponha uma extensão, agora com uma tecla para apagar e uma outra para acender. Escreva abaixo o código criado.

Em cada caso, escolha uma opção.

1. O *Serial Monitor* é muito usado para:
 - a) Mostrar valores lidos.
 - b) Mostrar o circuito montado.
 - c) Interpretar sinais.
 - d) Mostrar, automaticamente, valores nas escalas do Sistema Internacional.
2. Quando ajustamos a tensão para acender e apagar um LED, escolhemos um pino digital ao invés de um analógico, porque:
 - a) O pino digital mantém a tensão constante em 5 V.
 - b) O pino digital envia sinais de *HIGH* e *LOW*, quando for desejado.
 - c) Há mais pinos digitais do que analógicos.
 - d) Não faz diferença optar por um pino analógico.
3. As funções **Serial.begin()**, **Serial.flush()**, **Serial.read()** e **Serial.print()**, servem, respectivamente, para:
 - a) Iniciar o programa; esvaziar o *buffer*; ler o valor da porta serial; mostrar os dados na tela.
 - b) Determinar a taxa de *bits* por segundo (bps); esvaziar o *buffer*; ler o valor da variável serial; escrever na tela.
 - c) Iniciar o programa; controlar a transmissão de todos os dados do tipo serial e esvaziar o *buffer* da porta de entrada; ler o valor da variável serial; mostrar os dados na tela.
 - d) Determinar a taxa de *bits* por segundo (bps); esvaziar o *buffer* da porta de entrada (nas versões 1.0.x) e esperar o fim da transmissão de todos os dados seriais; ler o valor da porta serial; mostra os dados na tela.

6 Funções sonoras.

Data:
Aluno:
Aluno:

1. A função **tone()** gera, em um pino especificado, uma onda quadrada de frequência especificada, com ciclo de trabalho (*duty cycle*) de 50%. A duração também pode ser especificada. Caso contrário, a onda permanece até que haja uma chamada para a função _____.

Apenas um tom pode ser gerado de cada vez. Se um tom já está tocando em um pino diferente, a chamada para a **tone()** não terá nenhum efeito. Se o tom está tocando no mesmo pino, a chamada irá redefinir a sua frequência.

A ativação da função **tone()** irá interferir na saída _____ nos pinos 3 e 11 (em todas as placas, exceto no Arduino MEGA).

Não é possível gerar tons inferiores a 31 Hz.

Nota:

Se você quiser atuar em diferentes pinos, você precisará chamar **noTone()** em um pino antes de chamar a **tone()** no próximo pino.

Sintaxe:

tone(_____, _____)

ou

tone(_____, _____, _____)

2. Complete a Tabela 2, para cada nota musical.

Nota	Frequência (Hz)	Tecla usada
DÓ		
RÉ		
MI		
FÁ		
SOL		
LÁ		
SI		

Tabela 2: Associação Nota-Frequência-Tecla.

7 Funções de temporização e de comunicação com o *Serial Monitor*.

Data:

Aluno:

Aluno:

1. Apresentamos, na aula de hoje, as seguintes funções de temporização: _____, _____ e _____ .

Vimos que a função **delay()** é responsável por pausar o programa por um tempo determinado em _____. Por sua vez, as funções _____ e **micros()** retornam em milissegundos e _____, respectivamente, o tempo que o programa está em execução.

2. Nesta aula, foi trabalhada a função **Serial.print()**. Escreva um pequeno trecho de código, utilizando-a, para escrever, no *Serial Monitor*, o conteúdo de uma variável que esteja armazenando a quantidade de tempo, em microssegundos, em que o programa está em execução.

8 Sensor de luminosidade e função *analog.Read()*.

Data:
Aluno:
Aluno:

1. Na aula de hoje, compreendemos melhor o uso da função **analog.Read()**. Vimos que esta função é utilizada para _____ de um dos pinos _____ específico.

Estudamos também o conceito de quantização e descobrimos como funciona o conversor **A/D** do Arduino. Dado que este conversor apresenta uma resolução de _____, teremos uma escala com 1024 valores quantizados, variando de _____ a _____. Tendo assim uma resolução de leitura de _____ para cada unidade de medida em uma escala entre de 0 e 5 V.

2. Descreva com suas palavras o funcionamento do processo de quantização e a conversão analógico-digital (A/D) feita pelo Arduino.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

3. Nesta aula, utilizamos um sensor de luminosidade LDR, ilustrado na Figura 6. Este sensor nada mais é do que uma resistência _____, que varia de acordo com a luminosidade incidida. Observamos que, quanto maior a luminosidade incidente, _____ era a tensão lida pelo Arduino, e que, quanto menor a luminosidade, _____ era a tensão lida.



Figura 6: Sensor de luminosidade LDR.

4. Desenhe, abaixo, o diagrama utilizado para a leitura do sensor LDR.

9 Sensor de temperatura e funções matemáticas.

Data:

Aluno:

Aluno:

1. Nesta aula, utilizamos um sensor de temperatura e a função `analog.Read()` para lermos o valor de _____ e identificarmos a temperatura correspondente.

A Figura 7 apresenta uma imagem do sensor. Identifique a sua pinagem.

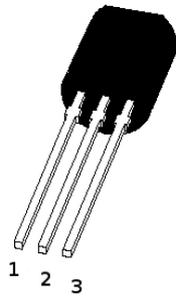


Figura 7: Sensor de temperatura.

Pino / função :

1 -

2 -

3 -

2. Na aula anterior, estudamos a função `analog.Read()` e vimos como funciona o conversor A/D do Arduino. Vimos que o conversor possui uma resolução de leitura de _____, que é calculada dividindo _____ por _____, que é a quantidade de valores quantizados.

Utilizamos esta mesma função para determinar a temperatura lida pelo sensor. Para isso, vimos, em aula, que, para cada _____ lido pelo Arduino, tínhamos uma elevação de 1 K.

Assim, para sabermos a temperatura em graus Celsius basta diminuir a temperatura lida em Kelvin por _____.

10 Sensor de movimento PIR e alarme com *buzzer*.

Data:
Aluno:
Aluno:

1. Nesta aula, foi explicado o funcionamento do sensor de movimento PIR (*Passive InfraRed*). O circuito tem um funcionamento simples. Explique, com suas palavras, o funcionamento do circuito:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

2. O sensor infravermelho pode nos ser útil em diversas aplicações. Entre elas, podemos fazer um alarme sonoro. Um sensor de presença pode ser instalado em um canto da parede, ao lado da porta. Quando uma pessoa atravessar a porta, o sensor captará movimento, acionando um alarme. Desenvolva um *script* para este pequeno projeto, de tal forma que ele acione um *buzzer* quando uma pessoa atravessar a porta.

12 *Display* de cristal líquido (LCD 16x2).

Data:
Aluno:
Aluno:

1. Na aula de hoje, aprendemos a trabalhar com o *display* de cristal líquido (LCD 16x2). Na Figura 8, temos a imagem do LCD usado em aula. Identifique a função de cada pino.

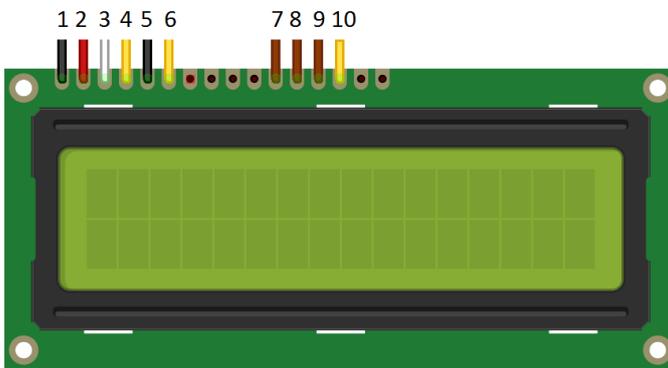


Figura 8: *Display* de cristal líquido LCD 16x2.

Pino / função:

- 1 -
- 2 -
- 3 -
- 4 -
- 5 -
- 6 -
- 7 -
- 8 -
- 9 -
- 10 -

2. Utilizando o LCD 16x2, crie um *script* que exiba o valor de temperatura lido pelo Arduino, a partir de um sensor de temperatura.

13 Uso de circuito integrado externo.

Data:
 Aluno:
 Aluno:

Neste projeto, é utilizado o circuito integrado CD4511, mostrado na Figura 9. Ele é um decodificador, do código BCD-8421 para o código de acionamento de um *display* de 7 segmentos, ilustrado na Figura 10. Isso possibilita a conexão de um Arduino com um *display* de 7 segmentos utilizando um número menor de pinos.

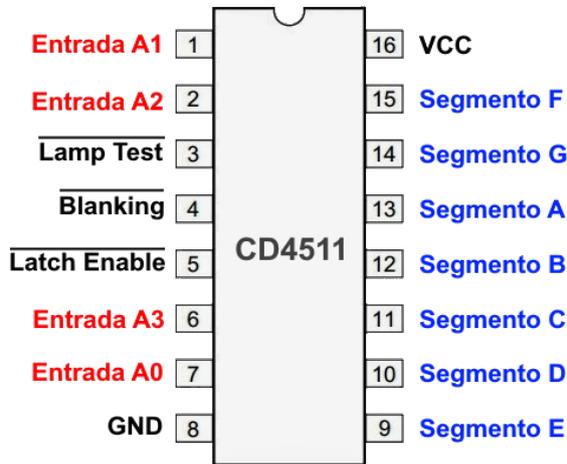


Figura 9: Decodificador CD4511, do código BCD-8421 para um *display* de 7 segmentos.

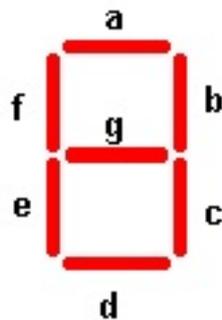


Figura 10: *Display* de 7 segmentos.

1. Apresente os valores binários do código BCD-8421 que geram os algarismos de 0 a 9 no *display* de 7 segmentos:

Valor decimal / valor BCD-8421:

- 0 -
- 1 -
- 2 -
- 3 -
- 4 -
- 5 -
- 6 -
- 7 -
- 8 -
- 9 -

2. Elabore um código Arduino que realize a contagem de 0 a 9 e, posteriormente, de 9 a 0, com o *display* de 7 segmentos.

14 Comunicação RF.

Data:
 Aluno:
 Aluno:

Pino / função:
 1 -
 2 -
 3 -
 4 -

1. Identifique a pinagem do Receptor RF da Figura 11.

3. Elabore um *script* para ler um receptor RF e acender diferentes LEDs, um para cada mensagem distinta recebida.



Figura 11: Receptor RF.

Pino / função:
 1 -
 2 -
 3 -
 4 -
 5 -
 6 -
 7 -
 8 -

2. Identifique a pinagem do Transmissor RF da Figura 12.



Figura 12: Transmissor RF.

15 Jogo Genius.

Data:
Aluno:
Aluno:

Após montar a prática mostrada hoje, adapte-a para que tenha um contador de pontos. A cada rodada, o seu programa deverá mostrar, no *Serial Monitor*, quantos acertos o jogador possui. Quando acontecer um erro, ele deverá zerar a contagem.

Apresente, abaixo, o código proposto.

O Genius é um jogo que foi muito famoso na década de 1980 e que buscava estimular a memória do jogador através de cores e sons.

A prática de hoje trata de uma adaptação do Genius para o Arduino. Precisaremos dos seguintes componentes: 4 botões, 8 resistores de $330\ \Omega$, 4 LEDs de cores diferentes e um *buzzer*.

A montagem do circuito é bem simples, sendo ilustrada na Figura 13. Porém, como temos uma quantidade maior de componentes do que nas práticas anteriores, é necessária uma maior atenção na montagem.

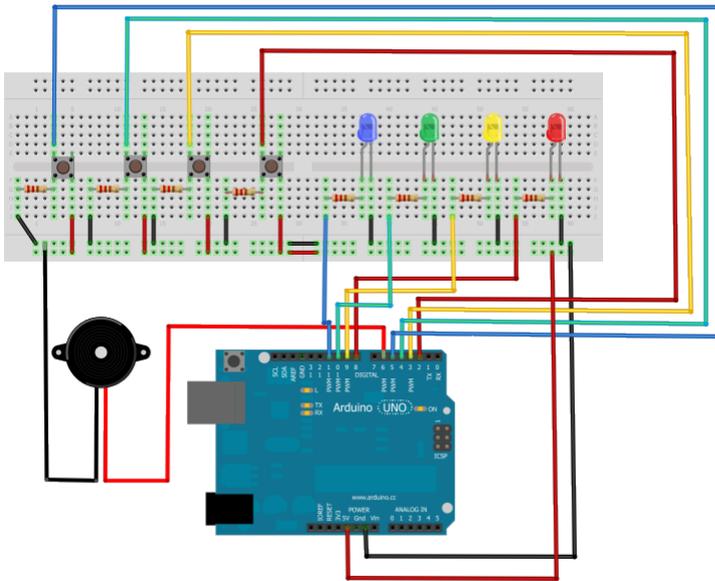


Figura 13: Circuito do jogo Genius.

16 Interação com *webpage*

Data:
Aluno:
Aluno:

1. Na aula que tivemos hoje, aprendemos como funciona um projeto utilizando o Arduino e o *shield ethernet*. Vimos que, para acionarmos diferentes cargas, possuímos diferentes circuitos, um para cada finalidade. Imagine que você possua uma sirene 12 V (DC). Qual circuito você utilizaria para acioná-la com o Arduino? Desenhe o mesmo abaixo:

3. Imagine que você possua uma casa de praia e que, nela, exista um poço de água potável. Recentemente, você esteve lá e deixou a caixa d'água vazia. Agora, você está em sua moradia, mas pretende voltar lá, em breve. Para não faltar água logo que você chegar na casa, você resolve ligar a bomba da sua casa de praia a partir de um projeto com o Arduino. Elabore um *script* que acione a bomba e que, ao identificar a caixa cheia, desligue-a e avise-o que a operação foi executada com sucesso.

2. Imagine, agora, que você possua um ventilador 110 V (AC). Faça o mesmo que foi pedido no item anterior.