

<b>CURSO:</b>	ENGENHARIA ELÉTRICA				
<b>TURMA:</b>	ELT0601N / ELT0602N	VISTO DO COORDENADOR	PROVA	TRAB.	GRAU
<b>DISCIPLINA:</b>	LAB. DE CIRCUITOS DIGITAIS	AVALIAÇÃO REFERENTE:	A1 <input type="checkbox"/>	A2 <input checked="" type="checkbox"/>	A3 <input type="checkbox"/>
<b>PROFESSOR:</b>	RODRIGO BARROS	<b>DATA:</b>	<b>Nº NA ATA:</b>		
<b>UNIDADE:</b>					
<b>NOME DO ALUNO:</b>		<b>MATRÍCULA:</b>			
<b>NOME DO ALUNO:</b>		<b>MATRÍCULA:</b>			

**Atenção!!! Leia, atentamente, as instruções para a realização da prova apresentadas abaixo:**

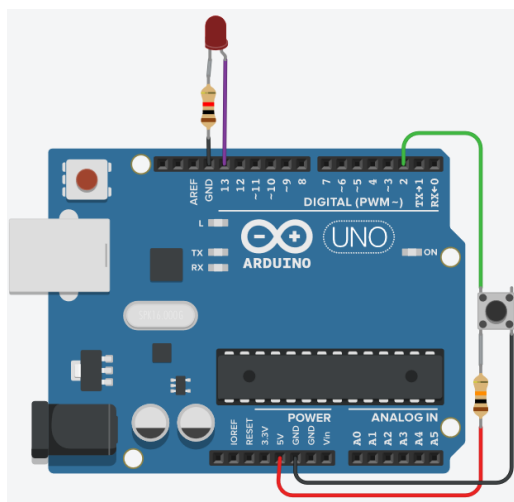
- 1ª) os arquivos contendo o código das questões devem ser entregues em formato “.txt”.
- 2ª) é necessário, também, criar um arquivo “.txt” contendo os links de compartilhamento de cada simulação.
- 3ª) crie um arquivo zip contendo todos os arquivos. Eles deverão ser entregues via Classroom.
- 4ª) os nomes dos arquivos deverão ser compostos da seguinte maneira  
[MATRICULA]\_[PRIMEIRONOME]\_[ULTIMONOME].zip.  
Ex.: 12345678\_Rodrigo\_Barros.zip
- 5ª) o trabalho está valendo de 0,00 a 10,00.
- 6ª) o trabalho poderá ser realizado em dupla.
- 7ª) todos os integrantes deverão postar uma cópia do trabalho no ambiente.
- 8ª) apresentar o nome da dupla nos arquivos do trabalho.

**QUESTÃO 1 (2,00) (NÍVEL 1) (INTERRUPÇÃO EXTERNA)** –Deseja-se fazer um projeto Pisca LED. O projeto apresenta os seguintes requisitos:

- 1 – O LED inicialmente deve piscar com um atraso de 500 ms, ou seja, de 500 ms nível alto e 500 ms nível baixo.
- 2 – Toda vez que o botão (Figura 1) for pressionado, deverá diminuir o tempo atraso de nível alto e o tempo de nível baixo em 100ms.
- 3 – Deseja-se que seja utilizado o recurso de interrupção externa.

Dicas:

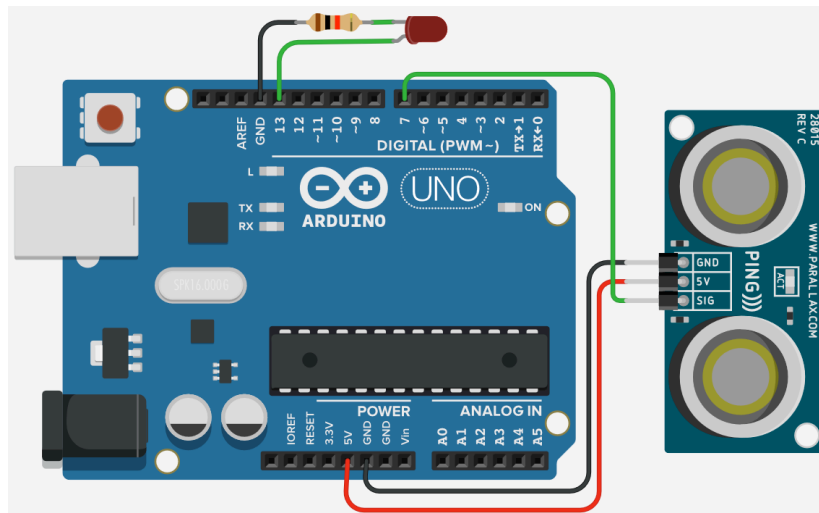
- Utilize a função “*attachInterrupt()*” para realizar a leitura por interrupção.
- Na interrupção utilize o modo FALLING, que verifique a mudança do nível alto para nível baixo do sinal do botão.
- Crie uma variável para armazenar o valor do atraso.





**QUESTÃO 2 (4,00) (NÍVEL II) (COM. SERIAL / SENSOR ULTRASSÔNICO)** – Para o circuito da Figura 2, crie um firmware que:

- Configure o pino 13 como saída digital.
- Configure a serial para uma taxa de conversão de 9600 bits/segundo.
- No loop principal fique verificando a distância do objeto utilizando o ultrassom.
  - Configure o pino 7 como saída
  - Coloque o pino 7 para nível lógico baixo
  - Espere 2 micro segundos
  - Coloque o pino 7 para nível lógico alto por 10 micro segundos;
  - Coloque o pino 7 para nível lógico baixo
  - Configure o pino 7 como entrada
  - Utilizando a função “*pulseIn()*” leia o tempo de retorno do eco em nível alto.
  - Calcule a distância em centímetros utilizando a equação:
    - $\text{distancia} = (\text{tempo de retorno} / (10000.0/343.0)) / 2;$
- O firmware deve apresentar, utilizando a serial, a distância com precisão de três casas decimais.
- O LED deverá piscar com atrasos diferentes para 3 faixas de distâncias:
  - $0 \leq \text{distância} < 50 \text{ cm}$ : atraso de 100 ms
  - $50 \leq \text{distância} < 150 \text{ cm}$ : atraso de 200 ms
  - $\text{distância} \geq 150 \text{ cm}$ : atraso de 500 ms



**FIGURA 2 – Circuito da Questão 2.**

**QUESTÃO 3 (4,00) (NÍVEL I) (RECEPTOR IR / SERVO MOTOR)** – O circuito da Figura 3 representa um receptor de infravermelho e três servo motores. Tanto o receptor IR como os motores são alimentados com 5 VDC. Cada servo motor representa um eixo de um braço robótico de 3 graus de liberdade (Eixo X, Eixo Y e Eixo Z). Deseja-se criar um sistema que controle a posição dos servos utilizando um receptor de infravermelho e um controle remoto. O projeto apresenta os seguintes requisitos:

- 1 – Os botões VOL+ e VOL- devem controlar o eixo Y, que poderá mover de 0 a 180°;
- 2 – Os botões REWARD [ << ] e FORWARD [ >> ] devem controlar o eixo X, que poderá mover-se de 0 a 180°;
- 3 – Os botões CIMA [ ^ ] e BAIXO [ v ] devem controlar o eixo Z que poderá mover-se de 0 a 90°;
- 4 – O botão PLAY [ >|| ] deve posicionar todos os motores na posição 0°;
- 5 – Os comandos enviados pelo controle remoto devem ser apresentados no monitor serial.

Dicas:

- Utilize a biblioteca “IrRemote.h” para realizar a leitura dos comandos do controle remoto.
- Utilize um tipo *long int* para armazenar o valor recebido pelo controle remoto.
- Realize um mapeamento prévios dos códigos dos botões pressionados

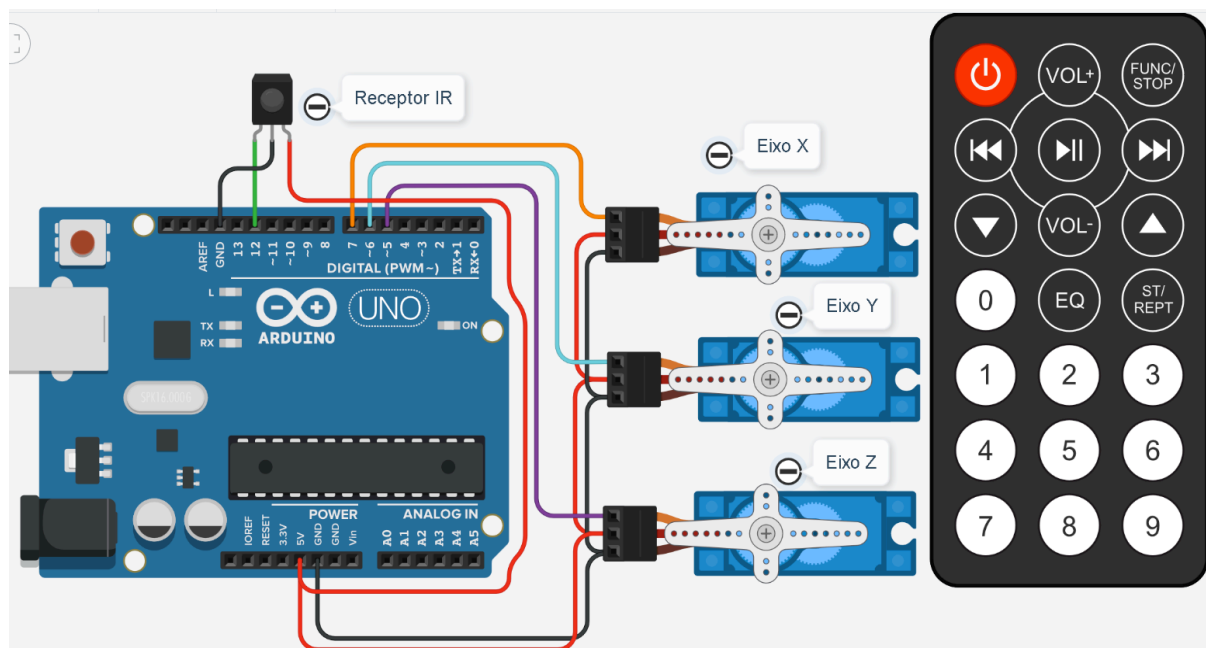


FIGURA 3 – Circuito da Questão 3.