



Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia

LEIM

Sensores e Atuadores – SA – 2122SI

## **Relatório 2-Parte Em Grupo**

**Docente Eng<sup>o</sup>:**

**Manfred Niehus**

**António Maçarico**

**Trabalho realizado por:**

**Rafael Cardoso dos Santos N<sup>o</sup>49722**

**Daniela Santos N<sup>o</sup>50032**

**Pedro Silva N<sup>o</sup>48965**

**Turma: 13D**

**Data de Entrega: 08/01/2022**

# Índice

## Conteúdo

Licenciatura em Engenharia Informática e Multimédia.....	1
1. Introdução Parte em Grupo.....	4
2. Exercício 1 (Apenas o Arduino).....	5
2.1. Alínea A.....	5
Implemente e teste código arduino para:.....	5
A)..... Uma variável ‘contador’, que incrementa uma vez por segundo, a mostrar na consola. .....	5
2.2. Alínea B.....	6
Implemente e teste código arduino para:.....	6
B)..... Um contador circular, de 0 a 7, que muda duas vezes por segundo, e mostra o valor na consola. .....	6
2.3. Alínea C.....	7
Implemente e teste código arduino para:.....	7
C)..... Um contador que sobe ao receber o carater ‘u’ (para ‘up’) via Serial, e desce ao receber ‘d’ (para ‘down’). .....	7
Implemente e teste código arduino para:.....	8
D)..... Um contador periódico que alternadamente sobe e desce, entre 0-20, com período 5s. .....	8
2.5. Alínea E.....	9
Implemente e teste código arduino para:.....	9
E)..... Um simulador de onda quadrada/sinusoidal c/ $V_{pp}=10$ , $V_{dc}=5$ , duty=50%, $T=2s$ e 10 pts por ciclo. .....	9
3. Exercício 5 (circuito com sensor distância sonar, circuito atuador com motor servo).....	10
3.1. Alínea A.....	10
3.2. Alínea B.....	11
3.3. Alínea C.....	12
4. Conclusão Parte em Grupo.....	13
Este relatório dá-se, então como concluído com a adição dos relatórios individuais dos membros do grupo, com a seguinte distribuição:.....	13
2º Problema- Rafael Santos Nº49722 3º Problema- Daniela Santos Nº50032 4ºProblema Pedro Silva Nº48965.....	13
Damos então este trabalho e relatório como bem-sucedidos, apesar de nos estar em falta a totalidade do registo de observação de sinal no osciloscópio (em parte por material estragado, confirmado com os docentes na aula de dia 15/12/2021, uma breadboard a fazer má ligação e alguns jumpers, que acabou por atrasar substancialmente o trabalho prático nessa aula) e pontualmente a conclusão, também em falta, das alíneas E) e F) do exercício 3.....	13
Com isto dito, consideramos que como grupo conseguimos alcançar e completar de forma praticamente total todos os objetivos propostos no trabalho pelos docentes.....	13
5. Introdução Parte Individual.....	15
6.1. Preparação e Montagem do exercício 4.....	16
6.2. Alínea A.....	17

Montar o circuito, e implemente e teste código arduino que mede o brilho externo / luminosidade captada pelo LDR, mostra o valor atual na consola, a refrescar cada meio segundo, e [caso piezo buzzer ativo]... regula cadência on/off do som, entre 1-10 Hz, em proporção ao brilho.....	17
[caso piezo buzzer passivo] ...regula frequência sonora entre 200-2000Hz, em proporção ao brilho Implementa a funcionalidade através da função int light2sound(byte pinLdr).....	19
7. Conclusão Parte Individual.....	21
Este relatório dá-se, então como concluído com a adição dos relatórios individuais dos membros do grupo, assim como o relatório de grupo que contém os exercícios 1 e 5. Concluo que neste segmento do trabalho individual foram realizados todos os exercícios propostos como pretendido tendo assim o meu grupo aprendido como trabalhar com o LDR e os piezos ambos passivo e ativo simultaneamente.....	21

## 1. Introdução Parte em Grupo

Para este trabalho prático teve-se como objetivo a montagem de cinco circuitos DC “base” e desenvolver o respetivo código para cada situação de forma a obter o resultado e/ou output desejado.

Para a montagem destes circuitos utilizámos o seguinte material:

- Breadboard;
- Resistências;
- Arduíno;
- Sensor de distância sonar;
- Motor Servo;
- Cabos de ligação (jumpers);
- Multímetro de bancada.

Software de apoio e de programação:

- TinkerCad (preparação das montagens dos circuitos);
- Arduíno IDE (Software de desenvolvimento de código).

Para cada alínea em cada exercício dedicamos uma página inteira, que nela contém: a preparação da sua montagem em TinkerCad (exceto no primeiro exercício, no qual não é necessária montagem uma vez que o arduíno é simplesmente ligado ao computador), uma foto da montagem de acordo com a preparação, uma caixa de texto contendo o código usado e finalmente uma print do output desejado, quando o mesmo é pedido, ou duas fotos do circuito montado, exemplificando as diferenças que podem ser observadas (como por exemplo no Exer.5\_A) a movimentação do motor servo).

Ainda de realçar que neste ficheiro apenas constam os exercícios 1 e 5, da parte em grupo e o exercício 2, da parte individual, sendo os restantes exercícios, 3 e 4, enviados pelos restantes membros do grupo.

## 2. Exercício 1 (Apenas o Arduino)

### 2.1. Alínea A

Implemente e teste código arduino para:

A) Uma variável ‘contador’, que incrementa uma vez por segundo, a mostrar na consola.

```
int contador = 0;
void setup() {
    Serial.begin(9600);
}
void loop() {
    contador++;
    Serial.print(contador);
    Serial.print("\n");
    delay(1000);
}
```

*Figura 1-Código Exer1 A)*

```
14:24:15.103 -> 1
14:24:16.085 -> 2
14:24:17.062 -> 3
14:24:18.091 -> 4
14:24:19.071 -> 5
14:24:20.106 -> 6
14:24:21.087 -> 7
14:24:22.071 -> 8
14:24:23.101 -> 9
14:24:24.082 -> 10
14:24:25.112 -> 11
```

*Figura 2-Output Exer1 A)*

## 2.2. Alínea B

Implemente e teste código arduino para:

B) Um contador circular, de 0 a 7, que muda duas vezes por segundo, e mostra o valor na consola.

```
int contador =0;
void setup() {

  Serial.begin(9600);
}

void loop() {

  for(contador=1;contador<8;contador++){
    Serial.print(contador);
    Serial.print("\n");
    delay(500);
  }
}
```

*Figura 3-Código Exer1 B)*

```
14:23:14.480 -> 1
14:23:14.949 -> 2
14:23:15.463 -> 3
14:23:15.978 -> 4
14:23:16.446 -> 5
14:23:16.960 -> 6
14:23:17.476 -> 7
14:23:17.988 -> 1
14:23:18.458 -> 2
14:23:18.973 -> 3
14:23:19.488 -> 4
14:23:19.954 -> 5
14:23:20.467 -> 6
14:23:20.982 -> 7
```

*Figura 4-Output Exer1 B)*

### 2.3. Alínea C

Implemente e teste código arduino para:

- C) Um contador que sobe ao receber o carater 'u' (para 'up') via Serial, e desce ao receber 'd' (para 'down').

```
int contador=0;
char c;
void setup()
{ Serial.begin(960
0);
}
void loop() {
c = Serial.read();
if(c=='u'){ conta
dor++;
Serial.print(contador);
Serial.print("\n");
}
if(c=='d'){
contador--;
Serial.print(contador);
Serial.print("\n");
}
```

*Figura 5-Código Exer1 C)*

d

*Figura 6-Output Exer1 C)*

```
14:25:43.894 -> 1
14:25:44.366 -> 2
14:25:44.787 -> 3
14:25:45.113 -> 4
14:25:45.441 -> 5
14:25:48.056 -> 4
14:25:48.341 -> 3
14:25:48.575 -> 2
```

## 2.4. Alínea D

Implemente e teste código arduino para:

D) Um contador periódico que alternadamente sobe e desce, entre 0-20, com período 5s.

```
int contador = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  for(contador=1;contador<21;contador++){ Serial.pr
int(contador);
  Serial.print("\n");
  delay(125);
}
for(contador=19;contador>1;contador--
){ Serial.print(contador);
  Serial.print("\n");
  delay(125);
}
}
```

Figura 7-Código Exer1 D)

```
14:31:38.892 -> 1
14:31:39.035 -> 2
14:31:39.175 -> 3
14:31:39.268 -> 4
14:31:39.407 -> 5
14:31:39.548 -> 6
14:31:39.641 -> 7
14:31:39.782 -> 8
14:31:39.921 -> 9
14:31:40.015 -> 10
14:31:40.154 -> 11
14:31:40.293 -> 12
14:31:40.388 -> 13
14:31:40.529 -> 14
14:31:40.672 -> 15
14:31:40.765 -> 16
14:31:40.907 -> 17
14:31:41.051 -> 18
14:31:41.146 -> 19
14:31:41.285 -> 20
14:31:41.425 -> 19
14:31:41.519 -> 18
14:31:41.658 -> 17
14:31:41.797 -> 16
14:31:41.890 -> 15
14:31:42.030 -> 14
14:31:42.169 -> 13
14:31:42.308 -> 12
14:31:42.401 -> 11
14:31:42.541 -> 10
14:31:42.680 -> 9
14:31:42.774 -> 8
14:31:42.915 -> 7
14:31:43.053 -> 6
14:31:43.147 -> 5
14:31:43.288 -> 4
14:31:43.427 -> 3
14:31:43.521 -> 2
14:31:43.662 -> 1
14:31:43.802 -> 2
14:31:43.895 -> 3
14:31:44.037 -> 4
```

Figura 8-Output Exer1 D)



## 2.5. Alínea E

Implemente e teste código arduino para:

- E) Um simulador de onda quadrada/sinusoidal c/  $V_{pp}=10$ ,  $V_{dc}=5$ ,  $duty=50\%$ ,  $T=2s$  e 10 pts por ciclo.

```
int contador = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  for(contador=0;contador<63;contador++){ fl
    oat y= float(contador)/10;
    delay(200);
    Serial.println(sin(y)*100);

  }
}
```

Figura 9-Código Exer1 E)



Figura 10-Output Onda Sinusoidal Exer1 E)

### 3. Exercício 5 (circuito com sensor distância sonar, circuito atuador com motor servo)

#### 3.1. Alínea A

Montar circuito. Implemente e teste código arduino que:

A) Oscila o ponteiro do motor servo como se fosse um para-brisa.

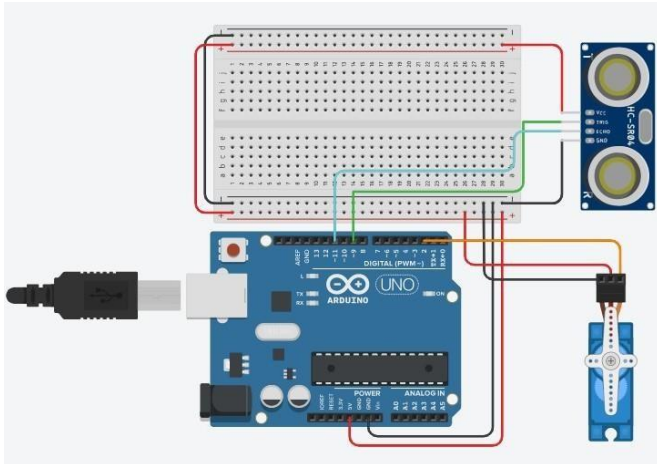


Figura 11-Preparação Exer5 A)

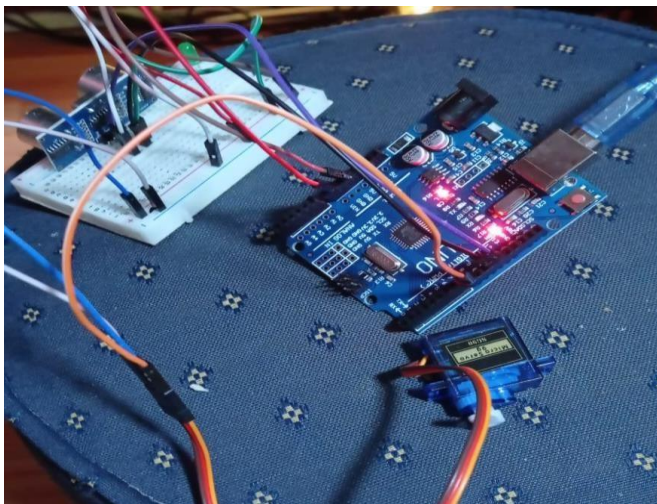


Figura 12-Montagem Exer5 A)



5a\_MOVIMENTO.mp4

Vídeo 1-Movimento do motor servo (para-brisas)

Figura 13-Código Exer5 A)

```
#include <Servo.h>

#define pinServo 8

Servo servo;
int posicao = 0;

void setup()
{
  servo.attach(pinServo);
  Serial.begin(9600);
}

void loop()
{
  for(posicao=0;posicao<=180;posicao++){
    servo.write(posicao);
    delay(10);
  }
  for(posicao=180;posicao>=0;posicao--){
    servo.write(posicao);
    delay(10);
  }
}
```

### 3.2. Alínea B

Montar circuito. Implemente e teste código arduino que:

B) Mede a distância cinco vezes por segundo e apresenta o valor na consola.

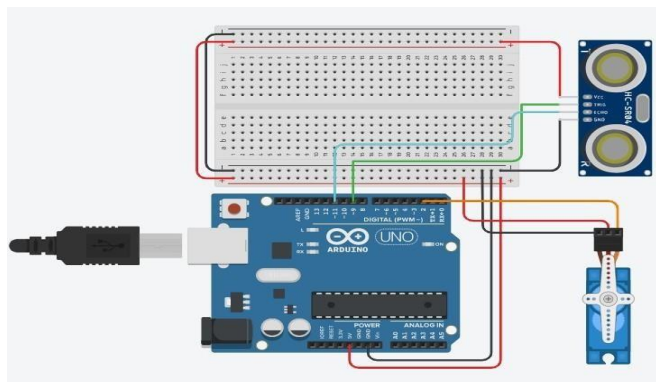


Figura 14-Preparação Exer5 B)

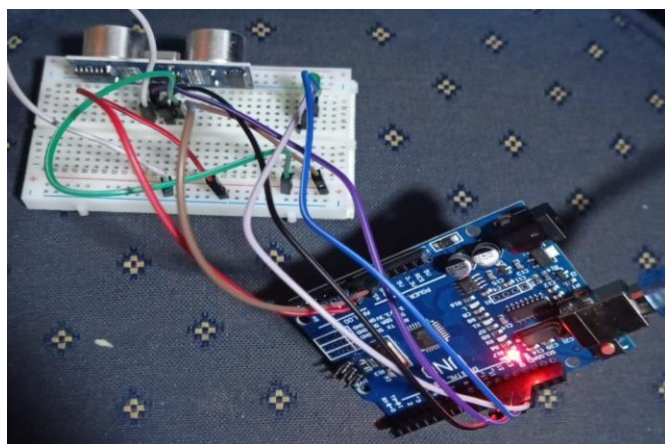


Figura 15-Montagem Exer5 B)

```
16:11:51.022 -> Distância: 5 cm
16:11:51.205 -> Distância: 5 cm
16:11:51.390 -> Distância: 5 cm
16:11:51.620 -> Distância: 7 cm
16:11:51.804 -> Distância: 5 cm
16:11:52.035 -> Distância: 5 cm
16:11:52.219 -> Distância: 12 cm
16:11:52.403 -> Distância: 10 cm
16:11:52.634 -> Distância: 11 cm
16:11:52.818 -> Distância: 14 cm
16:11:53.048 -> Distância: 20 cm
16:11:53.232 -> Distância: 17 cm
16:11:53.462 -> Distância: 17 cm
16:11:53.648 -> Distância: 21 cm
16:11:53.831 -> Distância: 57 cm
16:11:54.062 -> Distância: 57 cm
16:11:54.246 -> Distância: 56 cm
16:11:54.477 -> Distância: 56 cm
16:11:54.661 -> Distância: 56 cm
16:11:54.891 -> Distância: 56 cm
16:11:55.075 -> Distância: 14 cm
16:11:55.306 -> Distância: 13 cm
16:11:55.491 -> Distância: 56 cm
16:11:55.675 -> Distância: 56 cm
16:11:55.906 -> Distância: 57 cm
```

Figura 17-Output Exer5 B)

```
#include <NewPing.h>

#define TRIGGER_PIN 11
#define ECHO_PIN 12

NewPing sonar(TRIGGER_PIN, ECHO_PIN);

void setup()
{ Serial.begin(9600);
}

void loop()
{ delay(200);
  Serial.print("Distância: ");
  Serial.print(sonar.ping_cm());
  Serial.println(" cm");
}
```

Figura 16-Código Exer5 B)

### 3.3. Alínea C

Montar circuito. Implemente e teste código arduino que:

- C) Regula a posição/velocidade do servo, em proporção a posição potenciômetro.

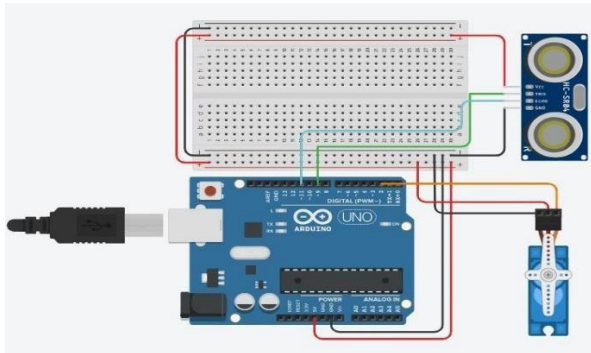


Figura 18-Preparação Exer5 C)

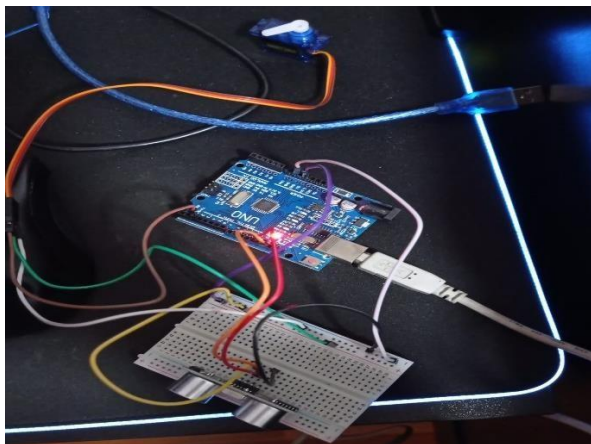


Figura 19-Montagem Exer5 C)



5c\_VIDEO.mp4

Video 2-Exer5 C)

```
#include <NewPing.h>
#include <Servo.h>

#define pinServo 7
#define TRIGGER_PIN 9
#define ECHO_PIN 12

Servo servo;
int posicao = 0;

NewPing sonar(TRIGGER_PIN, ECHO_PIN);
void setup() {
  servo.attach(pinServo);
  Serial.begin(9600);
}
void loop() {
  int dist=sonar.ping_cm();

  Serial.print("Distância: ");
  Serial.print(dist);
  Serial.println(" cm");
  servo.write(dist);

  delay(250);
}
```

Figura 20-Código Exer5 C)

## **4. Conclusão Parte em Grupo**

Este relatório dá-se, então como concluído com a adição dos relatórios individuais dos membros do grupo, com a seguinte distribuição:

2º Problema- Rafael Santos N°49722

3º Problema- Daniela Santos N°50032

4º Problema Pedro Silva N°48965

Damos então este trabalho e relatório como bem-sucedidos, apesar de nos estar em falta a totalidade do registo de observação de sinal no osciloscópio (em parte por material estragado, confirmado com os docentes na aula de dia 15/12/2021, uma breadboard a fazer má ligação e alguns jumpers, que acabou por atrasar substancialmente o trabalho prático nessa aula) e pontualmente a conclusão, também em falta, das alíneas E) e F) do exercício 3.

Com isto dito, consideramos que como grupo conseguimos alcançar e completar de forma praticamente total todos os objetivos propostos no trabalho pelos docentes.

# **Sensores e Atuadores – SA – 2122SI**

## **Relatório 2-Parte Individual**

**Docente Engº:**

**Manfred Niehus**

**António Maçarico**

**Trabalho realizado por:**

**Pedro Silva Nº48965**

**Turma: 13D**

**Data de Entrega: 08/01/2022**

## 5. Introdução Parte Individual

Para este trabalho prático teve-se como objetivo a montagem de um circuito DC “base” e desenvolver o respetivo código para cada situação de forma a obter o resultado e/ou output desejado.

Para a montagem deste circuito utilizámos o seguinte material:

- Breadboard;
- Resistências;
- Arduíno;
- LDR;
- LED's;
- Piezo Ativo;
- Piezo Passivo;
- Cabos de ligação (jumpers);
- Osciloscópio de bancada;
- Multímetro de bancada.

Software de apoio e de programação:

- TinkerCad (preparação das montagens dos circuitos);
- Arduíno IDE (Software de desenvolvimento de código).

Para cada alínea em cada exercício dedicamos uma página inteira, que nela contém: uma caixa de texto contendo o código usado, um print do output desejado, quando o mesmo é pedido, ou duas fotos/vídeo do circuito montado, exemplificando as diferenças que podem ser observadas e finalmente uma foto do sinal observado no osciloscópio de bancada. A preparação e a montagem do exercício 2 foi feita numa página inteira separa das alíneas por um lado por ser universal a todas as alíneas do exercício por outro para obter mais espaço em cada página de alíneas aumentado assim, consequentemente a sua legibilidade.



## 6. Exercício 2 (circuito sensor com botão)

### 6.1. Preparação e Montagem do exercício 4

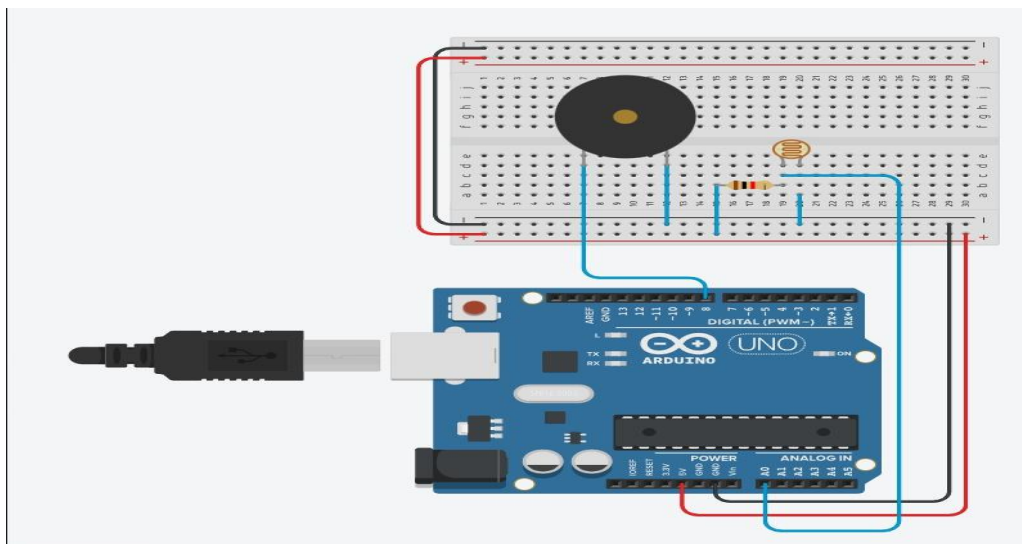


Figura 1-Preparação Exer4

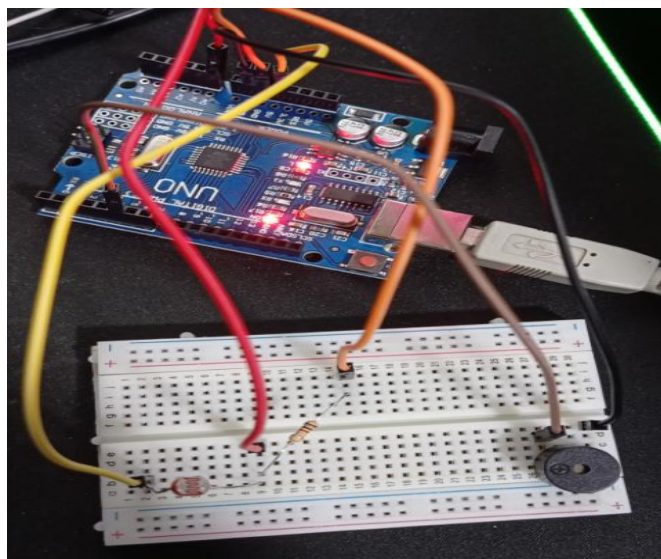


Figura2-Montagem Exer-4A

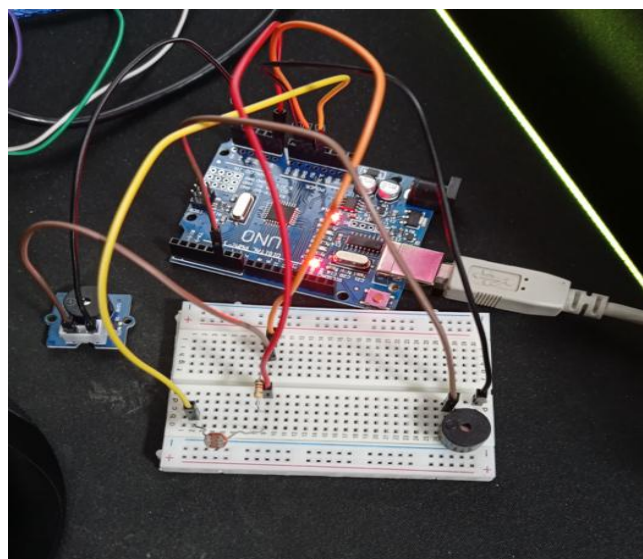


Figura3-Montagem Exer-4B



## 6.2. Alínea A

Montar o circuito, e implemente e teste código arduino que mede o brilho externo / luminosidade captada pelo LDR, mostra o valor atual na consola, a refrescar cada meio segundo, e [caso piezo buzzer ativo]... regula cadência on/off do som, entre 1-10 Hz, em proporção ao brilho

```
#define pinBuzzer 5
#define pinLDR A0
int sensorReading = 0;
void setup()
{
  pinMode(pinLDR, INPUT);
  pinMode(pinBuzzer, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  sensorReading = analogRead(A0);

  int valorHZ = map(sensorReading, 0, 1023, 0, 20);
  Serial.println(valorHZ);
  if(1<valorHZ && valorHZ < 10){
    digitalWrite(pinBuzzer, HIGH);
  } else{
    digitalWrite(pinBuzzer, LOW);
  }
  delay(500);
}
```

```
13:38:13.054 -> 2
13:38:13.561 -> 1
13:38:14.067 -> 3
13:38:14.573 -> 7
13:38:15.082 -> 7
13:38:15.543 -> 8
13:38:16.050 -> 9
13:38:16.557 -> 9
13:38:17.065 -> 3
13:38:17.572 -> 4
13:38:18.081 -> 6
13:38:18.589 -> 6
13:38:19.051 -> 6
13:38:19.558 -> 7
13:38:20.065 -> 5
13:38:20.571 -> 5
13:38:21.079 -> 0
13:38:21.587 -> 0
13:38:22.047 -> 0
```

Figura 5-Output Exer4 A)

Figura 4-Código Exer4 A)

## ***Conclusão da alínea A:***

Para resolver este exercício tivemos que fazer com que o piezo emitisse som em proporção ao brilho para isso utilização “map()” para transformarmos o valor de brilho de entre 0 e 1023 a 0 e 20 pois era nos mais útil um valor menor para termos melhor controlo. Implementámos um “if” com o objetivo de que se o brilho que o LDR recebesse for maior ou igual que 1 e menor ou igual a 10 o piezo emite um som caso contrário isto não acontece. Também utilizámos o “delay()” com um valor de 500 para que a cada meio segundo fosse escrito na consola o valor do brilho que o LDR recebe, como podemos observar no output.

### 6.3. *Alínea B*

[caso piezo buzzer passivo] ...regula frequência sonora entre 200-2000Hz, em proporção ao brilho Implementa a funcionalidade através da função `int light2sound(byte pinLdr)`.

```
#define pinLDR A5
#define pinBuzzer 5
int sensorReading = 0;
void setup() {
  pinMode(pinBuzzer,OUTPUT);
  Serial.begin(9600);
}
int light2sound(byte pinLDR){
  sensorReading = analogRead(A0);
  Serial.println(sensorReading);
  int valorHZ = map(sensorReading, 0, 1023, 200,
2000);
  tone(pinBuzzer,valorHZ,400);
  delay(500);
}
void loop() {
  light2sound(pinLDR);
}
```

```
15:42:20.698 -> 0
15:42:21.113 -> 0
15:42:21.527 -> 12
15:42:21.896 -> 25
15:42:22.312 -> 39
15:42:22.726 -> 112
15:42:23.141 -> 179
15:42:23.511 -> 224
15:42:23.927 -> 227
15:42:24.343 -> 252
15:42:24.712 -> 290
15:42:25.128 -> 341
15:42:25.543 -> 354
15:42:25.910 -> 352
15:42:26.326 -> 359
15:42:26.741 -> 399
15:42:27.110 -> 410
15:42:27.526 -> 393
15:42:27.941 -> 7
15:42:28.310 -> 0
```

Figura 7-Output Exer4 B)

Figura 6-Código Exer4 B)

## ***Conclusão da alínea B:***

Para resolver este exercício tivemos que fazer com que a frequência sonora do piezo fosse entre 200-2000Hz em proporção ao brilho do LDR para isso utilizamos outra vez a função “map” desta vez para transformar o valor de brilho de entre 0 e 1023 para um valor entre 200 e 2000 que vamos usar como a frequência sonora do nosso piezo. Também demos uso à função “tone()” que nos permite que o piezo emita som com a frequência desejada. Utilizámos o “delay()”, outra vez com um valor de 500 para que a cada meio segundo fosse escrito na consola o valor do brilho que o LDR recebe e para que nos desse tempo de ajustar o brilho que este recebe mudando assim a melodia. Podemos observar este processo na Figura 7 em que fomos gradualmente aumentando o brilho que o LDR recebia ao aproximando cada vez mais a luz, isto fez com que o som ficasse cada vez mais agudo. Por fim colocámos estes passos todos dentro de uma função chamada “int light2sound” que vai receber em byte a porta do arduíno em que está ligado o LDR.

## **7. Conclusão Parte Individual**

Este relatório dá-se, então como concluído com a adição dos relatórios individuais dos membros do grupo, assim como o relatório de grupo que contém os exercícios 1 e 5. Concluo que neste segmento do trabalho individual foram realizados todos os exercícios propostos como pretendido tendo assim o meu grupo aprendido como trabalhar com o LDR e os piezos ambos passivo e ativo simultaneamente.