Learning Unit	
Subject	Programming
Title	Programming in C++ Arduíno
Author	Eduardo Leite
School	FORAVE – Associação para a Educação Tecnológica do Vale do Ave
Description of the unit	The aim of this unit is to learn the basics of programming in C++ using Arduino.
Contents	Programming in C++ - Algorithms - Using the Arduino IDE; - Leds and Resistors; - Servomotor; - Buzzer; - LCD.
Learning Outcomes / Skills	Learning to create, plan and solve problems by connecting tangible artefacts such as the Arduino, building something with a purpose, while also providing links with contents from different areas of knowledge. Learn techniques for programming the Arduino controller. Develop specific programming for using sensors, actuators and motors associated with Arduino. Learning how to connect communication components to Arduino.
Target students/class	Secondary school (15 – 17 years old)
Prerequisites	Students should be able to: - Create electronic schematics - Create flowcharts - Assemble electronic equipment - Identify the various expressions, commands and operators in the C++ language - Use the Tinkercad simulation environments
Time expected	5 hours
Interdisciplinary links	ICT
Methodology	- Explanation of contents; - Solving exercises and problems; - Solving worksheets and working in groups.
Human Resources (internal and/or external)	Technical Studies Teacher

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Learning Unit	
Resources	- Worksheets; - Arduino - LEDs, resistors, LCDs, servomotor. - Computers with Arduino IDE
Lesson Plan	**<u>1st Lesson</u>** **Summary**: - Basic concepts in electronics Introduction of the theoretical concepts related to the topic of the lesson. After the introduction of the concepts and analysis of the solved example, students are asked to assemble an LED that is activated by a button. This assembly will be carried out in a simulation scenario and in a real laboratory scenario. Clarification of doubts. **<u>2nd Lesson</u>:** **Summary**: - Algorithms. - Flowcharts. - Pseudocode. Introduction of theoretical concepts related to the topic of the lesson. After the introduction of the concepts and analysis of the solved example, students are asked to create the 'Hello World' program in C++. In this lesson, the students will also be asked to assemble the LED on the Arduino board and to program it in C++. Clarification of doubts. **<u>3rd Lesson</u>:** **Summary**: - Programming in C++ (expressions, commands and operators) Introduction of the theoretical concepts related to the topic of the lesson. After the introduction of the concepts and the analysis of the solved example, students are asked to carry out the exercises on worksheet no. 1 in pairs. Clarification of doubts. **<u>4th Lesson</u>:** **Summary**: - Understanding an engine Step-by-step Introduction to the theoretical concepts related to the topic of the lesson. After the introduction of the concepts and the analysis of the solved example, students are asked to carry out the exercises on worksheet no. 2 in pairs. Clarification of doubts. **<u>5the Lesson</u>:** **Summary**:

Co-funded by the Erasmus+ Programme of the European Union

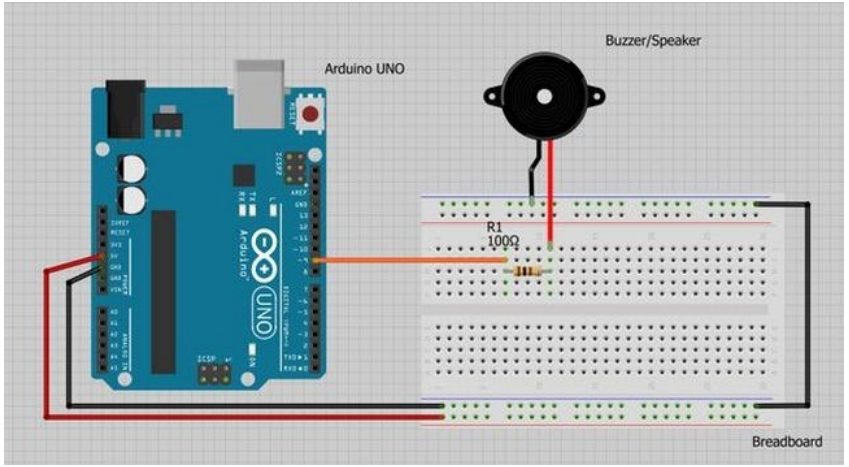
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Learning Unit	
	- Buzzer - LCD The teacher t introduces the theoretical concepts related to expressions, commands and operators. After introducing the concepts and analysing the solved example, the teacher asks the students to solve worksheet no. 3 in pairs. Clarification of doubts.
21st Century Skills	Critical thinking: students will be able to analyse data during practical experiments and communicate their conclusions. Collaboration: students will be able to collaborate within their groups and with other groups, helping each other to understand the content and the experimental activities. Communication: students should be able to share conclusions and doubts with their classmates and teachers. Technological literacy: students will be able to use different technological tools to carry out tasks.
Assessment	**Formative assessment:** - Attendance; - Punctuality; - Behaviour: - Attention and participation in class; - Completion of worksheets (direct observation grids).
Remarks	--

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

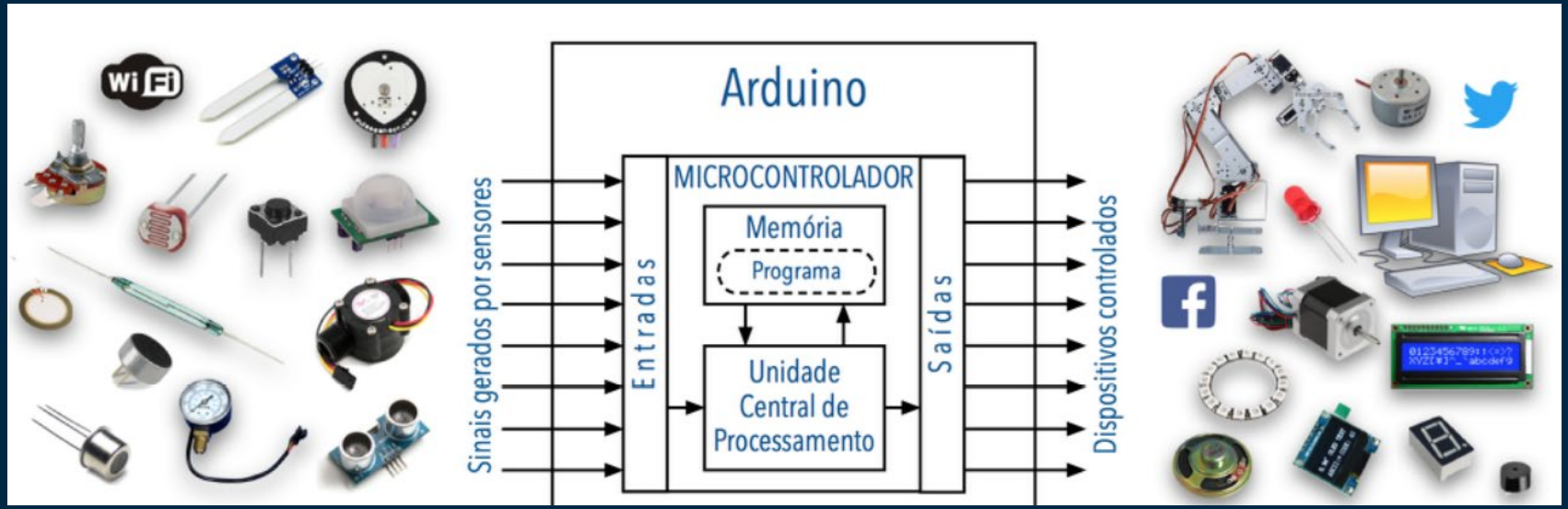
Programação



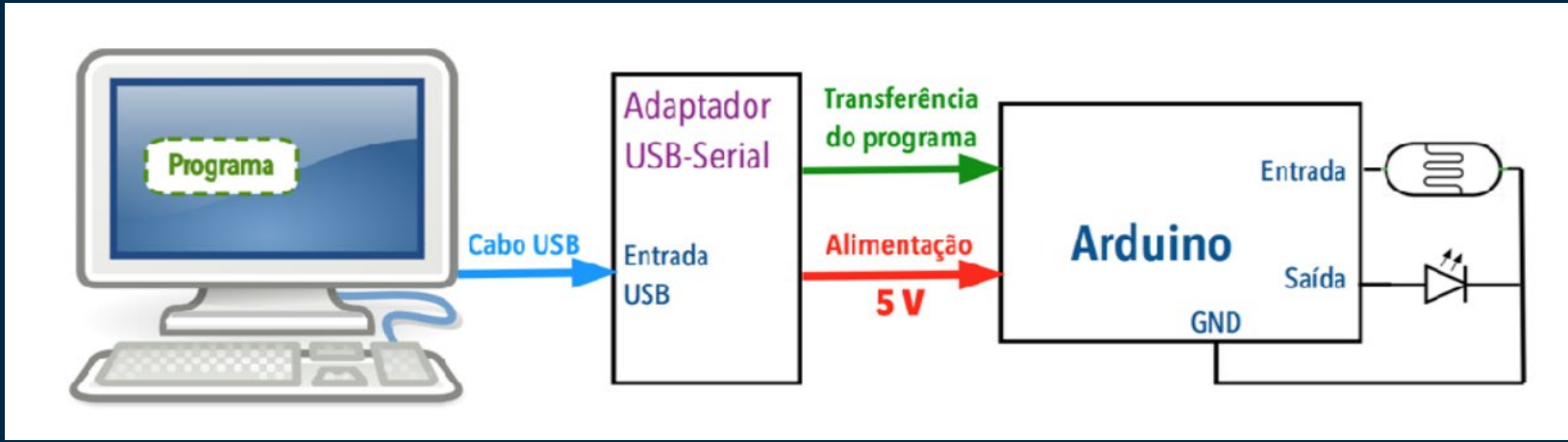
Co-funded by the Erasmus+ Programme of the European Union

PROGRAMAÇÃO ARDUÍNO DO BUZZER

ARDUÍNO


Wi Fi
Arduino
MICROCONTROLADOR
Memória
Programa
Unidade
Central de
Processamento
Entradas
Saídas
Sinais gerados por sensores
Dispositivos controlados

CONCEITO



- Linguagem de alto nível (baseada na Linguagem C)
- Programa é compilado (traduzido para a linguagem máquina do Arduino)
- A transferência é feita através da saída USB do computador que também fornece energia para a placa.

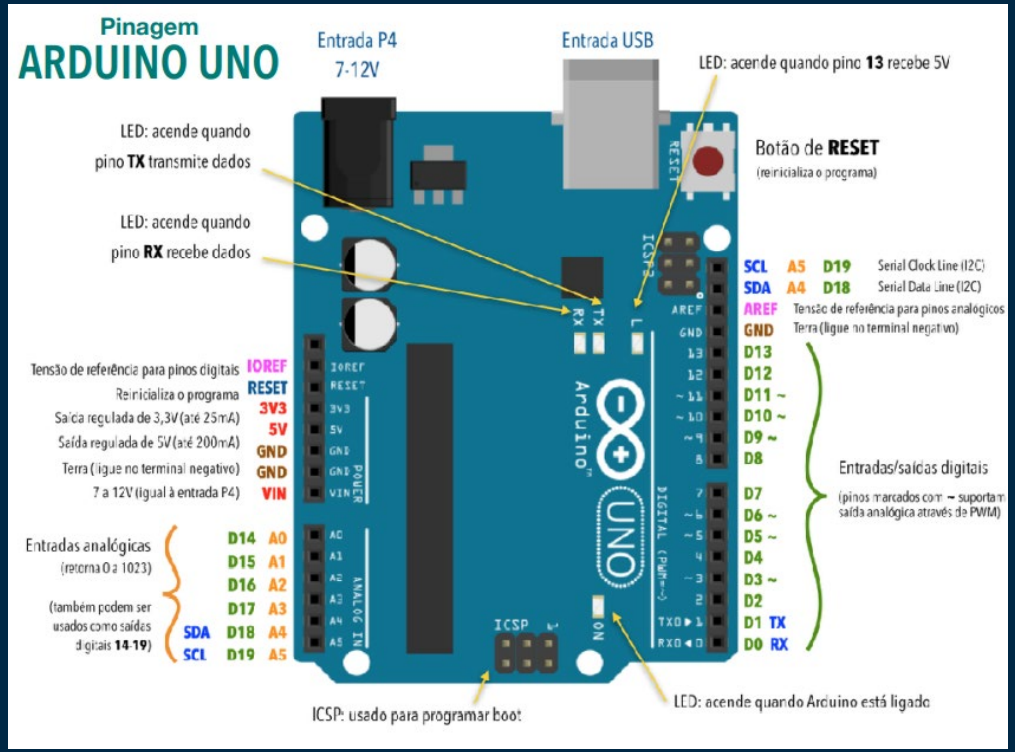
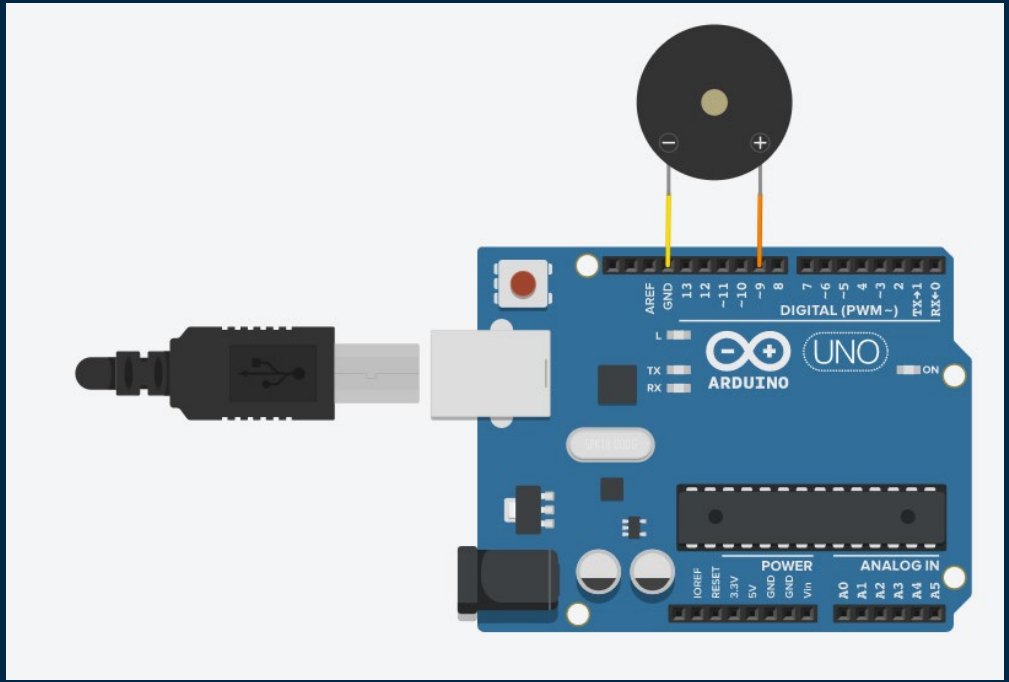

Pinagem
ARDUINO UNO
Entrada P4
7-12V
Entrada USB
LED: acende quando pino 13 recebe 5V
LED: acende quando pino TX transmite dados
LED: acende quando pino RX recebe dados
Botão de RESET
(reinicializa o programa)
SCL A5 D19 Serial Clock Line (I2C)
SDA A4 D18 Serial Data Line (I2C)
AREF Tensão de referência para pinos analógicos
GND Terra (ligue no terminal negativo)
Tensão de referência para pinos digitais IOREF
Reinicializa o programa RESET
Saída regulada de 3,3V (até 25mA) 3V3
Saída regulada de 5V (até 200mA) 5V
Terra (ligue no terminal negativo) GND
7 a 12V (igual à entrada P4) VIN
Entradas analógicas
(retorna 0 a 1023)
(também podem ser usados como saídas digitais 14-19)
D14 A0
D15 A1
D16 A2
D17 A3
SDA D18 A4
SCL D19 A5
D13
D12
D11 ~
D10 ~
D9 ~
D8
D7
D6 ~
D5 ~
D4
D3 ~
D2
D1 TX
D0 RX
Entradas/saídas digitais
(pinos marcados com ~ suportam saída analógica através de PWM)
ICSP: usado para programar boot
LED: acende quando Arduino está ligado

Diagrama elétrico



Vamos programar!!!

www.tinkercad.com

Programação

```
int pinBuzzer = 8; // o pino onde o buzzer está conectado
```

Programação

```
void setup() {
  pinMode(pinBuzzer, OUTPUT); // configura o pino do buzzer
como saída
}
```

Programação

```
void loop() {
  tone(pinBuzzer, 1000); // faz o buzzer tocar em uma
frequência de 1000 Hz
  delay(1000); // aguarda 1 segundo
```

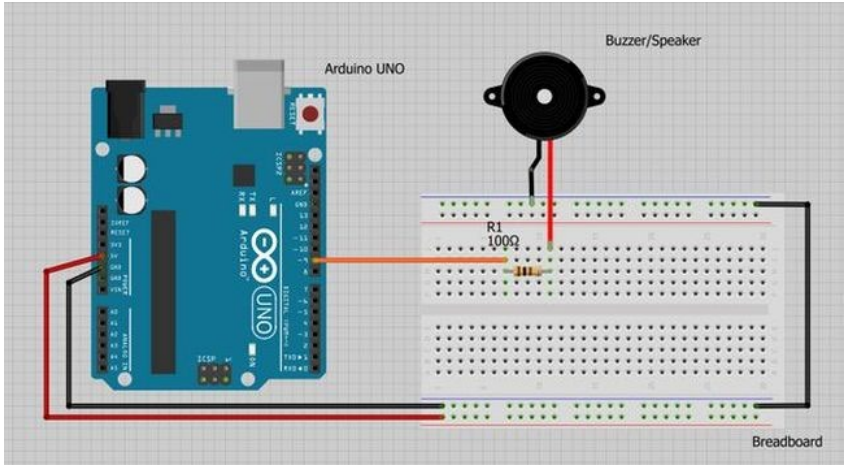
Programação

```
  noTone(pinBuzzer); // para o som do buzzer
  delay(1000); // aguarda mais 1 segundo
}
```

Programação

```
void loop()
{
    // Aciona o buzzer na frequencia relativa ao Dó em Hz
    tone(buzzer,261);
    // Espera um tempo para Desativar
    delay(200);
    //Desativa o buzzer
    noTone(buzzer);
    // Aciona o buzzer na frequencia relativa ao Ré em Hz
    tone(buzzer,293);
    delay(200);
    noTone(buzzer);
    // Aciona o buzzer na frequencia relativa ao Mi em Hz
    tone(buzzer,329);
    delay(200);
    noTone(buzzer);
    // Aciona o buzzer na frequencia relativa ao Fá em Hz
    tone(buzzer,349);
    delay(200);
    noTone(buzzer);
    // Aciona o buzzer na frequencia relativa ao Sol em Hz
    tone(buzzer,392);
    delay(200);
    noTone(buzzer);
}
```

Programação



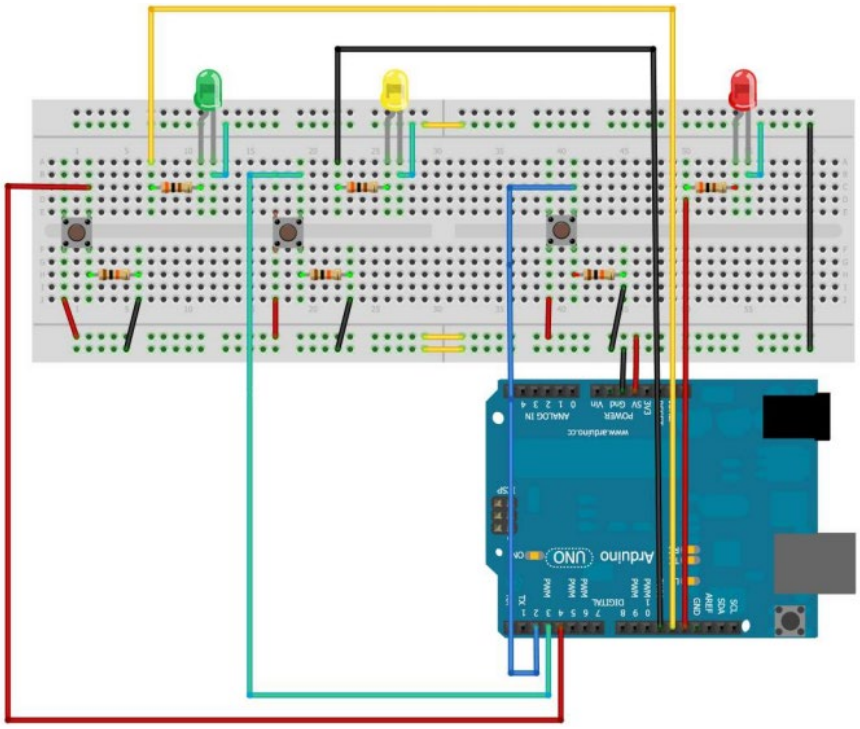
Co-funded by the Erasmus+ Programme of the European Union

Worksheet No. 1

Learning Unit	**Programação em C++ de Arduíno**
Author	Eduardo Leite
School	FORAVE –Associação para a Educação Tecnológica do Vale do Ave
Date	

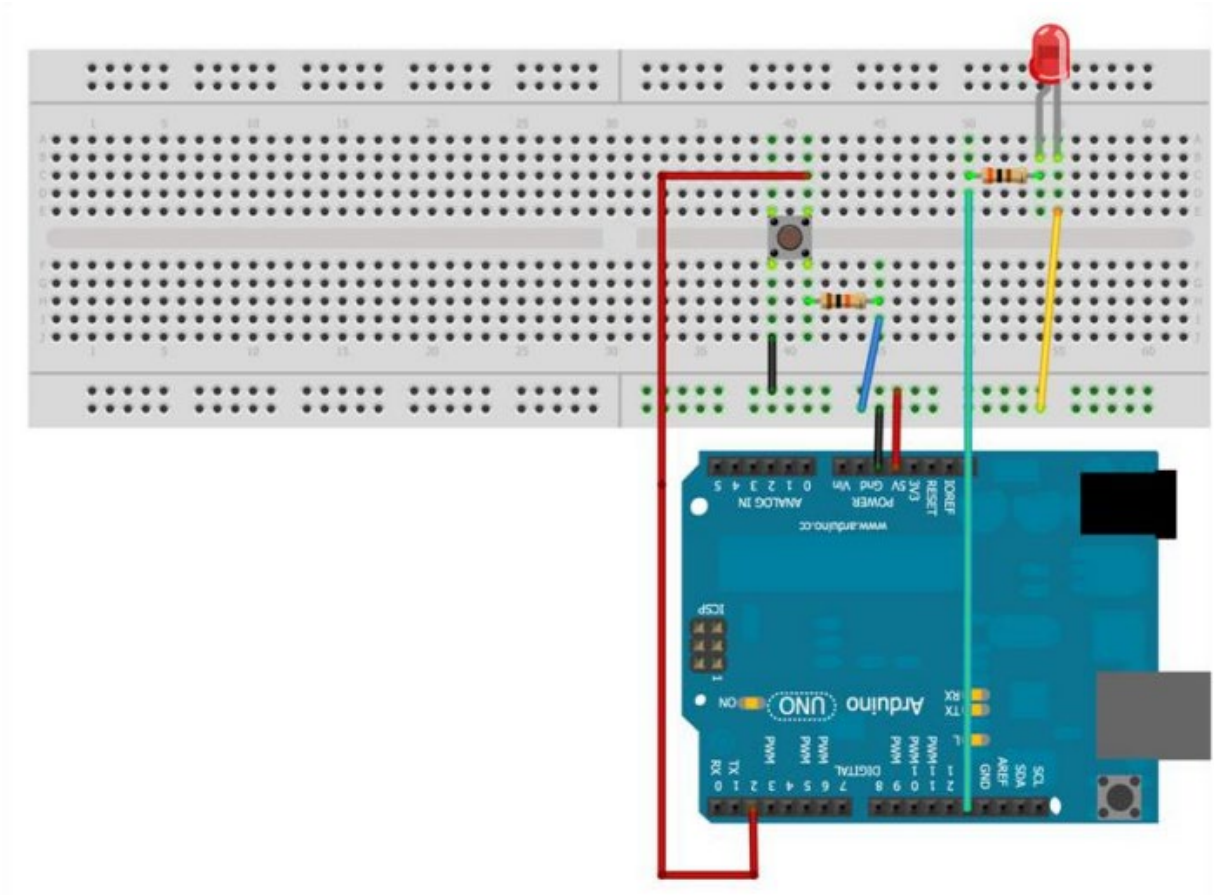
Note: Access the Tinkercad platform https://www.tinkercad.com and create a new project for each exercise.

1. Run the circuit on the above-mentioned platform as shown below. Use the buttons in the exercise to run a programme so that each button turns an LED on and off..



Co-funded by the Erasmus+ Programme of the European Union

2. Run the circuit on the aforementioned platform in the manner shown below. Note that the circuit will be very similar to the first one, however, the ‘while’ structure will be used instead of the ‘if’.



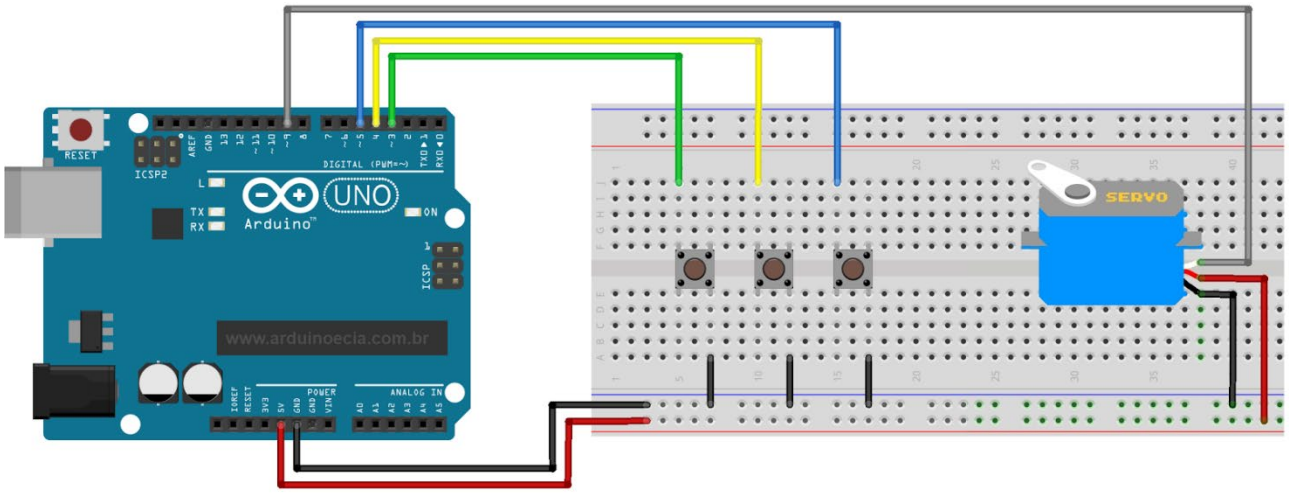
Co-funded by the Erasmus+ Programme of the European Union

Worksheet No. 2

Learning Unit	**Programação em C++ de Arduíno**
Author	Eduardo Leite
School	FORAVE –Associação para a Educação Tecnológica do Vale do Ave
Date	

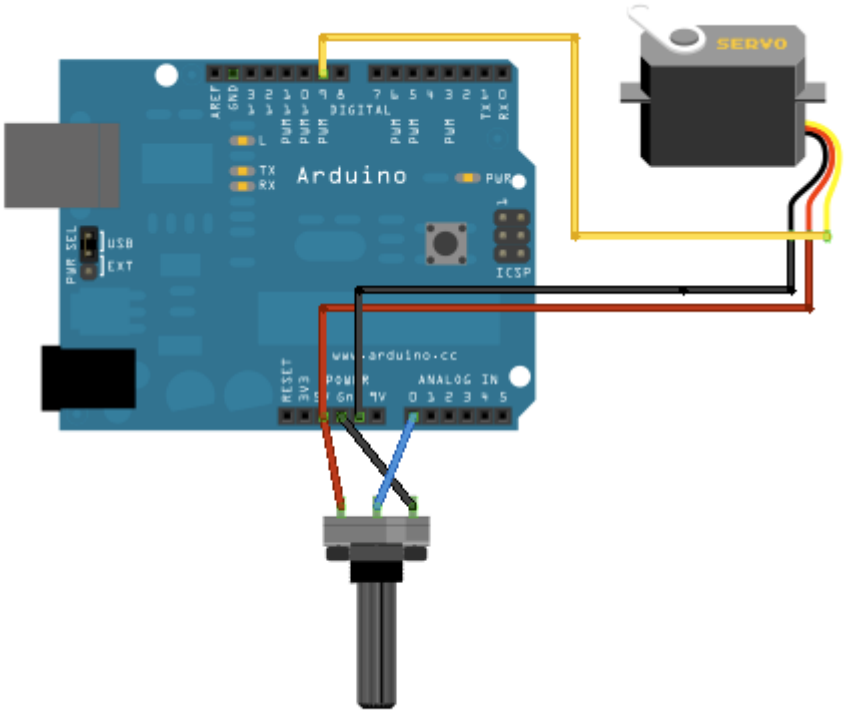
Note: Access the Tinkercad platform https://www.tinkercad.com and create a new project for each exercise.

1. Run the circuit on the above-mentioned platform as shown below. Use the buttons in the exercise to run a program so that each button positions the servo in three different positions.



Co-funded by the Erasmus+ Programme of the European Union

2. Run the circuit on the aforementioned platform as shown below. Use the potentiometer in the exercise to run a program so that varying the potentiometer changes the position of the servo by 180º..



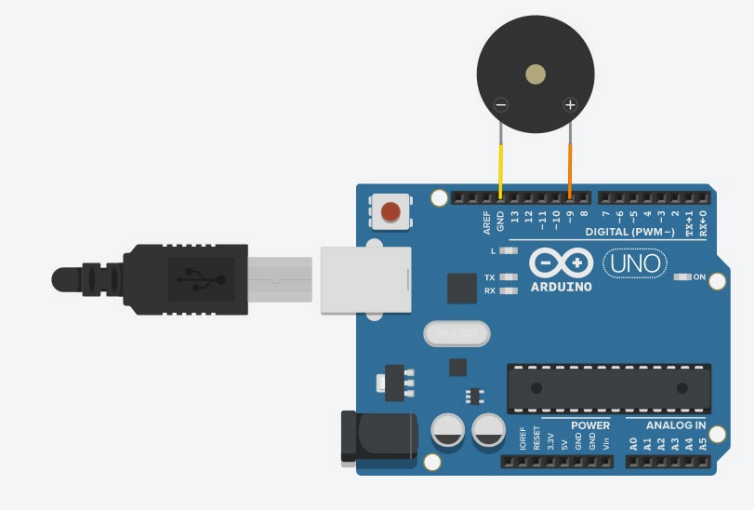
Co-funded by the Erasmus+ Programme of the European Union

Worksheet No. 3

Learning Unit	**Programação em C++ de Arduíno**
Author	Eduardo Leite
School	FORAVE –Associação para a Educação Tecnológica do Vale do Ave
Date	

Note: Access the Tinkercad platform https://www.tinkercad.com and create a new project for each exercise.

1. Run the circuit on the above-mentioned platform as follows. Use the buzzer in the exercise to run a program to reproduce a sound.



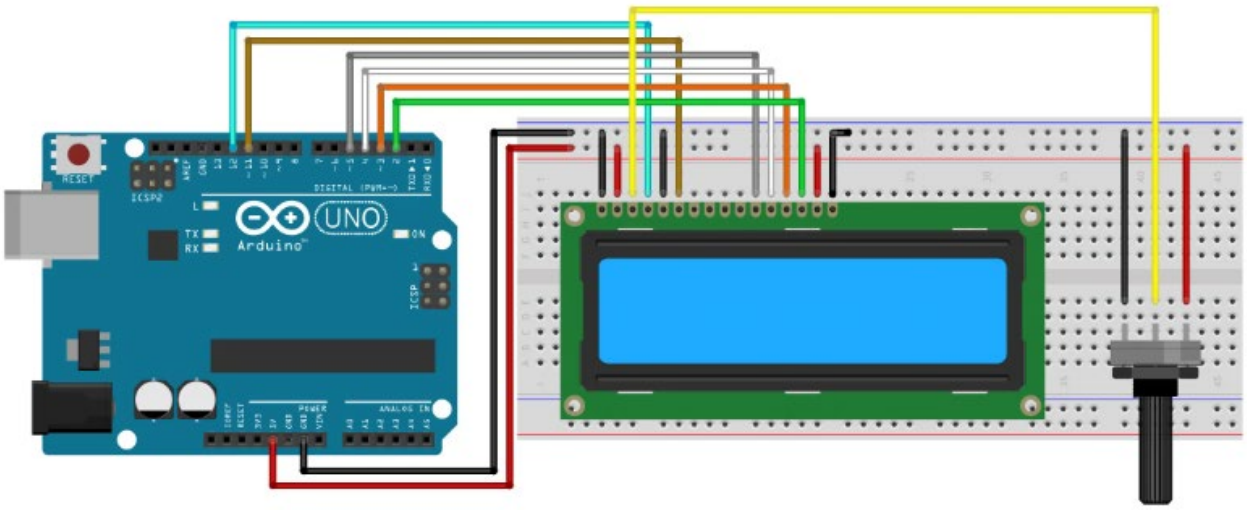
2. Using the circuit created in point 1, create a program that plays the notes Dó, Ré, Mi, Fá e Sol through the buzzer.

Nota	Frequency [Hz]
Dó	261
Ré	293
Mi	329
Fá	349
Sol	392

Co-funded by the Erasmus+ Programme of the European Union

3. Run the circuit on the above-mentioned platform as follows. Using the LCD in the exercise, run a program to play a message on the LCD.

Co-funded by the Erasmus+ Programme of the European Union



```
#include <LiquidCrystal.h>
//Define os pinos que serão utilizados para ligação ao display
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
  //Define o número de colunas e linhas do LCD
  lcd.begin(16, 2);
}
void loop()
{
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("Hello World");
  lcd.setCursor(3, 1);
  lcd.print(" LCD 16x2");
  delay(5000);
    for (int posicao = 0; posicao < 3; posicao++)
  {
    lcd.scrollDisplayLeft();
    delay(300);
  }
  for (int posicao = 0; posicao < 6; posicao++)
  {
    lcd.scrollDisplayRight();
    delay(300);
  }
}
```

Co-funded by the Erasmus+ Programme of the European Union