



**Curso de Eletrônica com Arduino -
Aula 06**

Funções

- 1º Declara-se a função que se quer criar.
- 2º Ao final do sketch, define-se o que a função faz.



ServoMotor

1 - O Servo Motor tem três terminais:

- GND
- PWM
- 5V



2 - Inclui-se no sketch a biblioteca Servo.h:

```
SERVO_MOTOR $  
#include <Servo.h>
```

3 - Declare objeto da biblioteca Servo.h:

```
Servo servol;
```

4 - Comando semelhante ao pinMode(), onde se relaciona o objeto da biblioteca ao pino:

```
void setup() {  
    servol.attach(pinServo);  
}
```

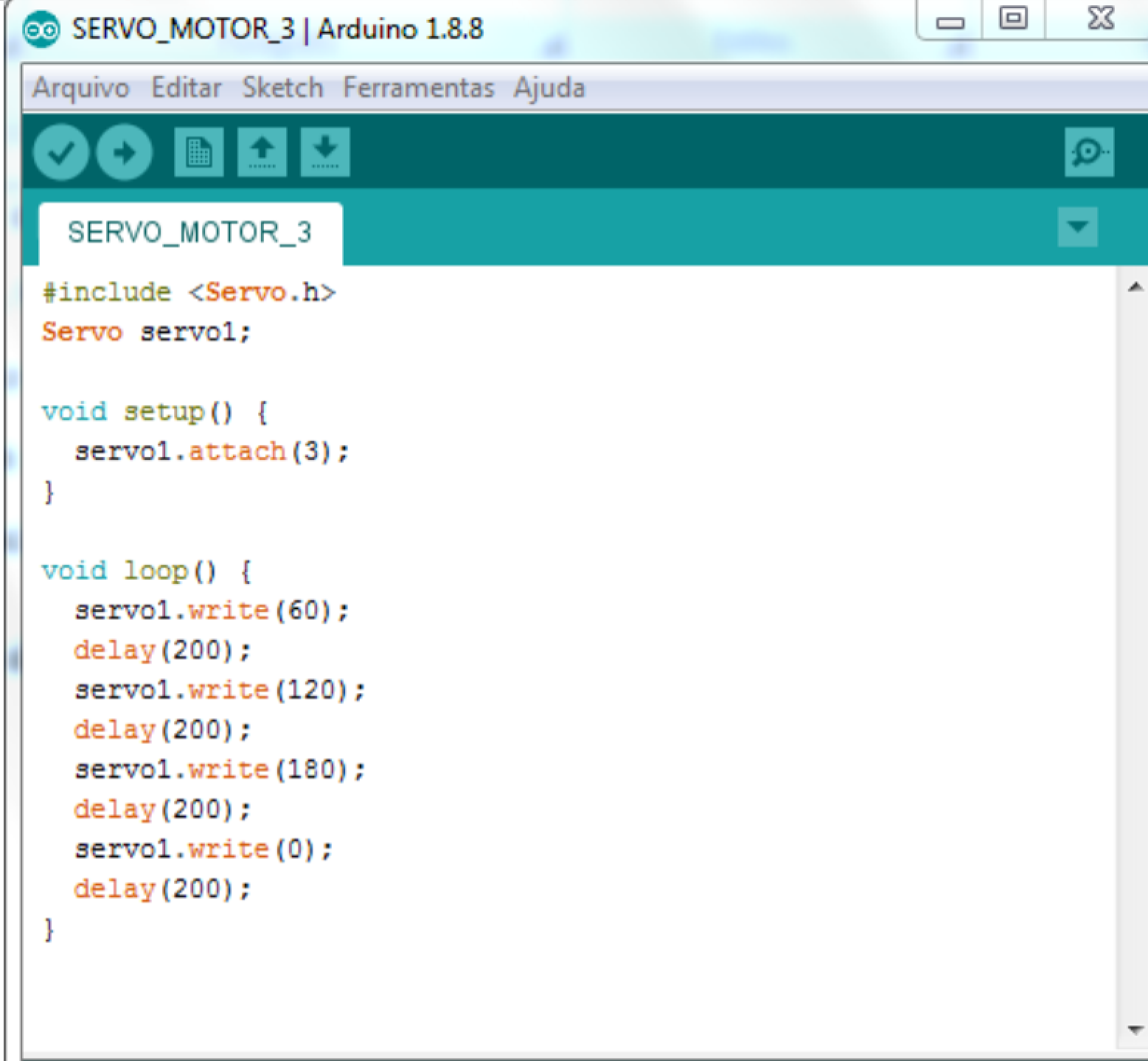


The image shows the Arduino IDE interface with a sketch titled "SERVO_MOTOR". The sketch is written in C++ and uses the Servo library to control a servo motor based on the position of a potentiometer. The code includes the Servo.h library, defines pins for the servo and potentiometer, and implements setup and loop functions. The loop function reads the potentiometer value, maps it to a servo angle, and writes the angle to the servo while also printing the raw and mapped values to the serial monitor.

```
SERVO_MOTOR $  
  
#include <Servo.h>  
  
#define pinServo 3  
#define pinPotenciometro A0  
  
Servo servol;  
  
void setup() {  
  servol.attach(pinServo);  
  Serial.begin(9600);  
}  
  
void loop() {  
  int lido = analogRead(pinPotenciometro);  
  byte angulo = map(lido, 0, 1023, 0, 179);  
  
  servol.write(angulo);  
  
  Serial.print("A0:");  
  Serial.print(lido);  
  Serial.print(" angulo:");  
  Serial.println(angulo);  
}
```

Controlando motor com potenciometro

Movimentando motore
4 angulos diferentes

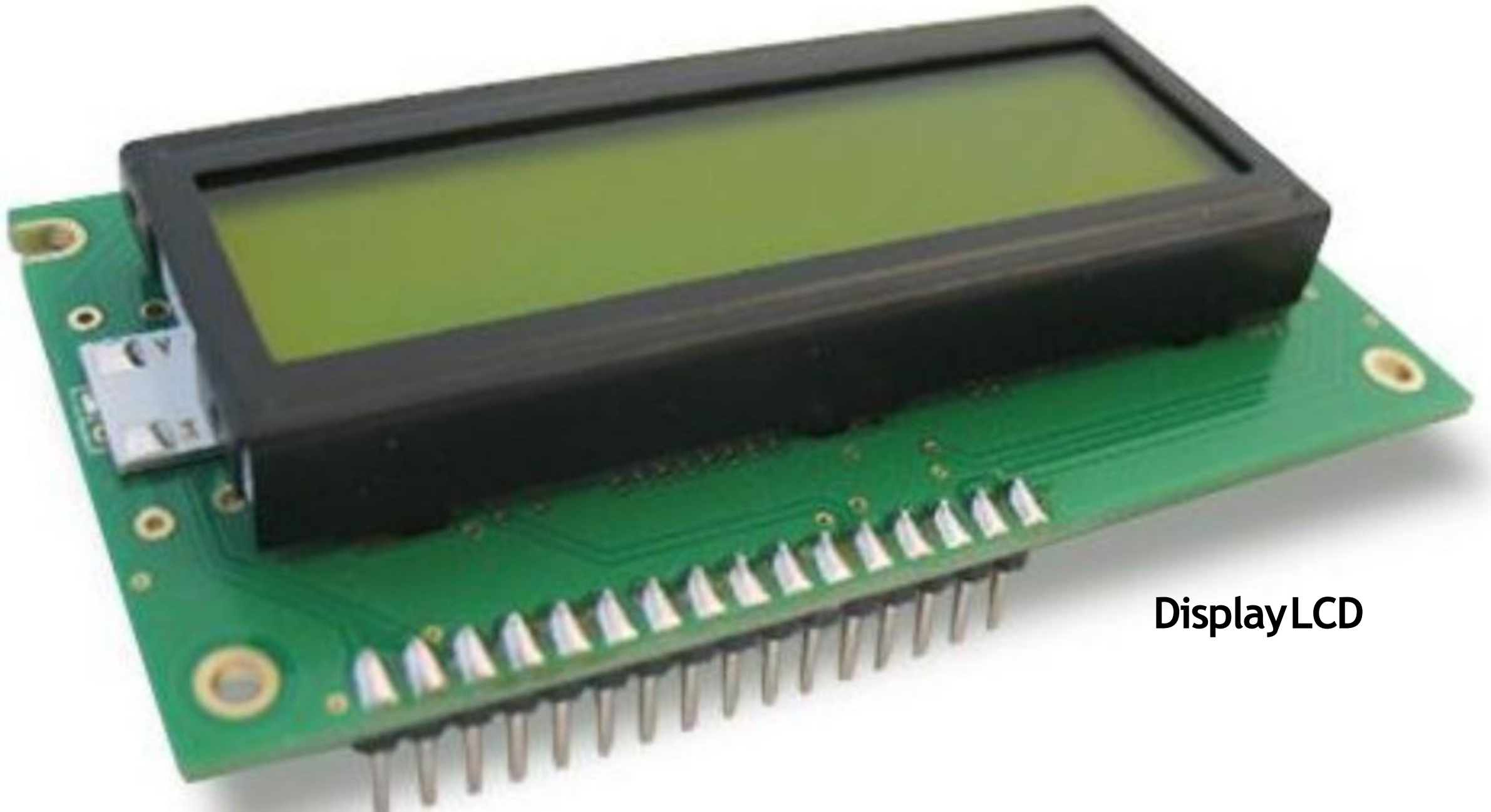


The screenshot shows the Arduino IDE interface with the title bar 'SERVO_MOTOR_3 | Arduino 1.8.8'. The menu bar includes 'Arquivo', 'Editar', 'Sketch', 'Ferramentas', and 'Ajuda'. The toolbar contains icons for checking, running, uploading, and downloading. The sketch name 'SERVO_MOTOR_3' is displayed in the tab. The code in the editor is as follows:

```
#include <Servo.h>
Servo servo1;

void setup() {
  servo1.attach(3);
}

void loop() {
  servo1.write(60);
  delay(200);
  servo1.write(120);
  delay(200);
  servo1.write(180);
  delay(200);
  servo1.write(0);
  delay(200);
}
```

Display LCD

1 - O Display pode ter 16 terminais:

- VSS -> GND
- VDD -> 5V
- VO -> potenciometro ou GND
- RS -> porta digital
- RW -> porta digital
- E -> porta digital
- D0 até D7 -> portas digitais
- A -> porta digital, com 220ohms
- K -> GND

2 - Inclui-se no sketch a biblioteca LiquidCrystal.h:

```
#include <LiquidCrystal.h>
```

3 - Declare objeto da biblioteca LiquidCrystal.h informando todos, os pinos conectados:

```
LiquidCrystal LCD(2,3,4,6,7,8,9,10,11,12,13);
```

4 - Comando semelhante ao Serial.begin(), que inicializa o display

```
void setup() {  
    LCD.begin(16,2);  
}
```

Entre parênteses devemos colocar o número de pinos do Display, seguido do número de linhas do Display.

5- Comandos usados:

```
LCD.print("Ola Mundo!!!");
```

```
LCD.clear();
```