



Laboratório de Materiais e Equipamentos Elétricos

Roteiro 11: Servo Motor

1. Equipe

Nome	RA
Gabriel Felipe Ferdinandi de Souza	2669480
Gustavo Ferreira Fonseca	2669510
Gustavo Henrique Gonçalves	2669528

2. Desenvolvimento da prática

Parte A:

Código

```
#include <Servo.h>
//Declarando variaveis:
#define servo_pino 3
#define pot_pin A0
int valor_potenciometro;

Servo servo_motor; //Criando Objeto Servo

void setup() {
    Serial.begin(9600); //Iniciando a comunicação serial
```

```

servo_motor.attach(servo_pino); //Inicializando controle Servo
} //Fim SETUP

void loop() {
    valor_potenciometro = analogRead(pot_pin);

    int posicao = map(valor_potenciometro, 0, 1023, 15, 165); //Mapeia a
    posição apartir da entrada do potenciometro
    servo_motor.write(posicao);

    Serial.println(posicao); //Exibe a posição na serial

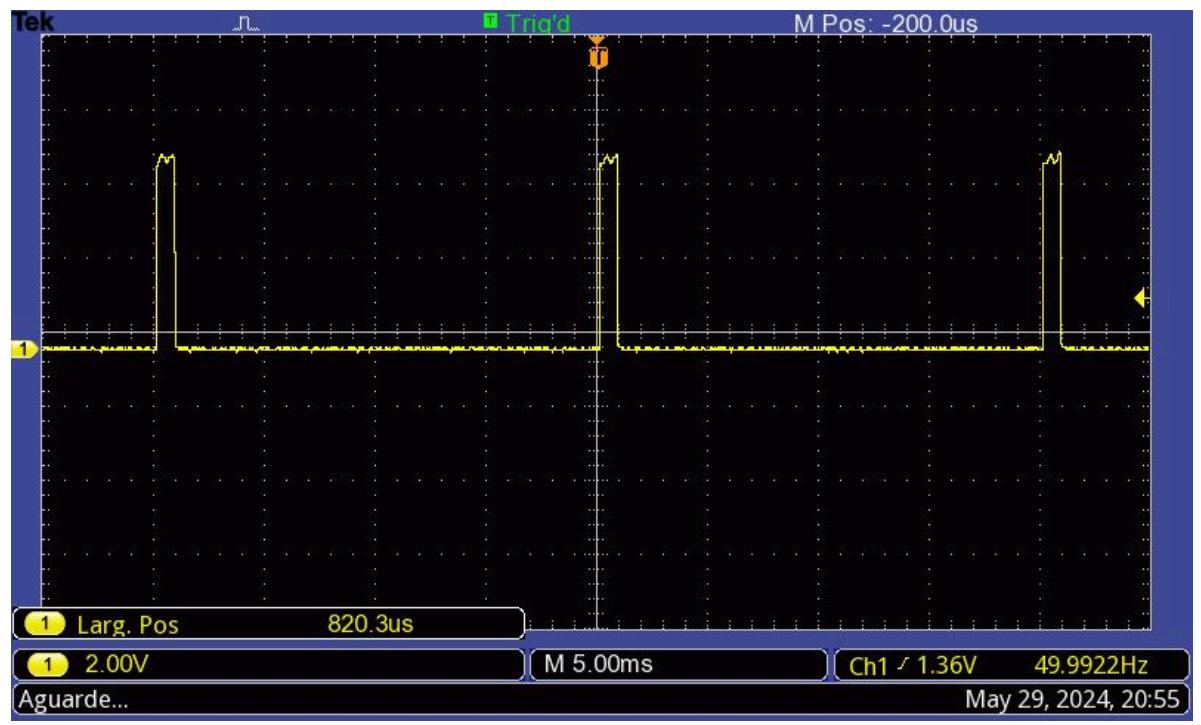
} //Fim loop

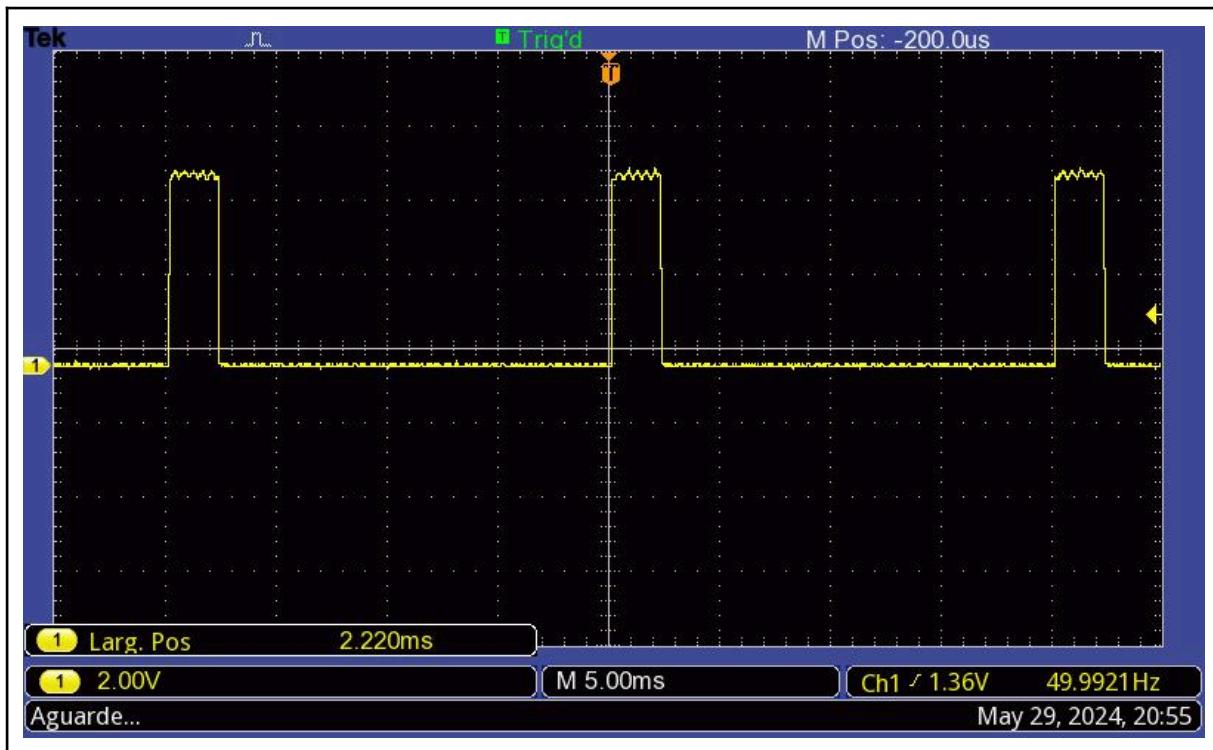
```

PERGUNTA: Qual o período mínimo e máximo da largura de pulso (ver pelo osciloscópio)?

R: O período do pwm de saída, na posição mínima foi de aproximadamente 800 μ s e o na posição máxima foi de aproximadamente 2.2ms.

PWM - Osciloscópio (imagem)





VALOR DO POTENCIÔMETRO - Serial (imagem)

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** alg_a_aino | Arduino IDE 2.3.2
- Menu Bar:** Arquivo, Editar, Rascunho, Ferramentas, Ajuda
- Tool Buttons:** Save, Build, Upload, Select Board (set to Arduino Uno), Select Port.
- Sketch Area:** The code for 'alg_a_aino' is displayed:

```
1 #include <Servo.h>
2 //Declarando variaveis:
3 #define servo_pino 3
4 #define pot_pin A0
5 int valor_potenciometro;
6
7 Servo servo_motor; //Criando Objeto Servo
8
9 void setup(){
10     Serial.begin(9600); //Iniciando a comunicação serial
11 }
```
- Bottom Status Bar:** Mensagem (Arduino Uno + Enter para enviar mensagem para 'COM6' em '2')
- Serial Monitor:** Shows the output: 665 repeated multiple times, followed by 665 at the end.
- Serial Monitor Settings:** Nova linha (Newline) and 9600 baud.
- Bottom Right:** Ln 13, Col 14, Arduino Uno em COM6, a small icon, and a file icon.

Parte B:

Código

```
#include <Servo.h>

//Declarando variaveis:

#define servo_pino 3
#define pot_pin A0
#define botao_1_pin 13
#define botao_2_pin 12

int valor_potenciometro, posicao = 0;
bool valor_botao1 = 0;
bool valor_botao2 = 0;

Servo servo_motor; //Criando Objeto Servo

void setup(){
    Serial.begin(9600); //Iniciando a comunicação serial
    servo_motor.attach(servo_pino); //Inicializando controle Servo
} //Fim SETUP

void loop(){
    valor_potenciometro = analogRead(pot_pin);
    valor_botao1 = digitalRead(botao_1_pin);
    valor_botao2 = digitalRead(botao_2_pin);

    //Checagem e incrementador de posição:
    if(valor_botao1 == 1)
        posicao++;
    if(valor_botao2 == 1)
        posicao--;

    //Limitador de posição - usei de [0, 180], presumindo que o servo
    está calibrado:
    if(posicao < 0)
        posicao = 0;
    if(posicao > 180)
        posicao = 180;
    servo_motor.write(posicao);
    Serial.println(posicao); //Exibe a posição na serial
    delay(map(valor_potenciometro, 0, 1023, 20, 175)); //Usa como delay
    o potenciômetro
} //Fim loop
```

Parte C:

Código

```
//Neste código pode ser incluída funções para encurta-lo, porém com o  
foco é apenas didático, preferi manter mais simples:  
  
#include <Servo.h>  
//Declarando variaveis:  
#define servo_pino 3  
#define pot_pin A0  
#define botao_1_pin 13  
#define botao_2_pin 12  
#define buzer_pin 8  
  
int valor_potenciometro, posicao = 0;  
bool valor_botao1 = 0;  
bool valor_botao2 = 0;  
bool saida = 1;  
unsigned long tempo = 0, tempo_anterior = 0;  
  
Servo servo_motor; //Criando Objeto Servo  
  
void setup(){  
    Serial.begin(9600); //Iniciando a comunicação serial  
    pinMode(buzzer_pin, OUTPUT);  
    servo_motor.attach(servo_pino); //Inicializando controle Servo  
} //Fim SETUP  
  
void buzer(){  
  
    digitalWrite(buzzer_pin, 1);  
    delay(350);  
    digitalWrite(buzzer_pin, 0);  
  
} //Fim função Buzer - emite um beep em um buzzer a cada troca de  
função  
  
void controle_man(){  
  
    while(saida == 0){
```

```

tempo = millis();
valor_botaol = digitalRead(botaol_pin);
valor_botaoo = digitalRead(botaoo_pin);
valor_potenciometro = analogRead(pot_pin);
Serial.println("slave"); //Exibe o modo de operação na serial
Serial.println(posicao); //Exibe a posição na serial

posicao = map(valor_potenciometro, 0, 1023, 15, 165); //Mapeia a
posição apartir da entrada do potenciômetro
servo_motor.write(posicao);

if(valor_botaol == 1 && valor_botaoo == 1){
    if(tempo - tempo_anterior > 3000){

        saida = 1;
        buzer();
        tempo_anterior = tempo;

    } } else{

tempo_anterior = tempo;

}//Fim if-else checagens - entra modo de config por botões

} //Fim while controla função manual (repetição até saida)

} //Fim controle função manual

void loop(){
tempo = millis();

valor_potenciometro = analogRead(pot_pin);
valor_botaol = digitalRead(botaol_pin);
valor_botaoo = digitalRead(botaoo_pin);

//Checagem e incrementador de posição:
if(valor_botaol == 1 && valor_botaoo == 0)
    posicao++;
if(valor_botaoo == 1 && valor_botaol == 0)
    posicao--;
}

```

```
if(valor_botaol == 1 && valor_botaol == 1){
    if(tempo - tempo_anterior > 3000){

        saida = 0;
        tempo_anterior = tempo;
        buzer();
        controle_man();

    } } else{

    tempo_anterior = tempo;

}//Fim if-else checagens - entra modo de config por potenciômetro

//Limitador de posição:
if(posicao < 0)
    posicao = 0;
if(posicao > 180)
    posicao = 180;

servo_motor.write(posicao);

Serial.println("main"); //Exibe o modo de operação na serial
Serial.println(posicao); //Exibe a posição na serial

delay(map(valor_potenciometro, 0, 1023, 20, 175)); //Usa como delay
o potenciômetro

} //Fim VOID
```