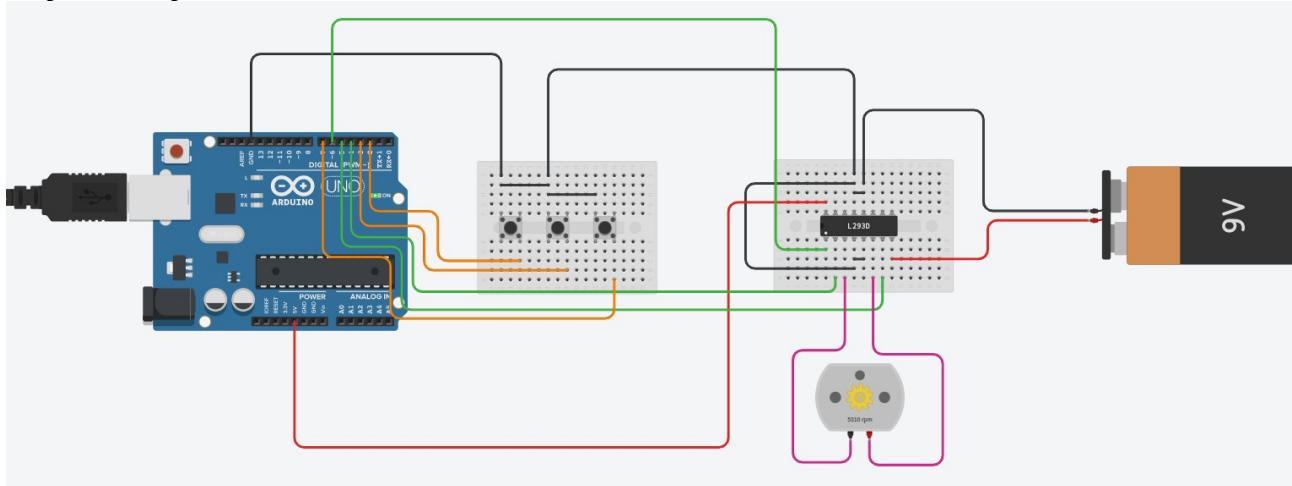


## SISTEMAS CON MICROPROCESADORES: PRÁCTICA 3

**Dirección web de los proyectos Tinkercad:**

- Ejercicio 1: <https://www.tinkercad.com/things/6jNeKPrnIZ>
- Ejercicio 2: <https://www.tinkercad.com/things/9gZ4YKle0XO>

**Captura de pantalla del funcionamiento del motor al 33% de velocidad:**



**Código fuente:**

```
// C++ code
//

bool estado_botpower;
bool estado_botvel;
bool estado_botsentido;
bool pin_4;
bool pin_5;
int vel_actual;

void setup()
{
  Serial.begin(9600);
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, INPUT_PULLUP);
```

```
estado_botpower = digitalRead(2);
estado_botvel = digitalRead(3);
estado_botsentido = digitalRead(7);
```

```
vel_actual=0;
pin_4 = LOW;
pin_5 = HIGH;
```

```
analogWrite(6, vel_actual);
digitalWrite(4, pin_4);
digitalWrite(5, pin_5);
```

```
}
```

```
void power_off()
```

```
{
```

```
    if(estado_botpower!=digitalRead(2)){
        estado_botpower=digitalRead(2);
```

```
        if(estado_botpower == LOW){
            if(vel_actual == 0){
                vel_actual+=85;
                analogWrite(6, vel_actual);
            }
```

```
            else{
                vel_actual=0;
                analogWrite(6, vel_actual);
            }
```

```
        }
```

```
    }
```

```
}
```

```
void cambio_sentido()
```

```
{
```

```
    if(estado_botsentido != digitalRead(3)){
        estado_botsentido = digitalRead(3);
```

```

    if(estado_botsentido == LOW){
        bool aux = pin_4;
        pin_4 = pin_5;
        pin_5 = aux;

        digitalWrite(4, pin_4);
        digitalWrite(5, pin_5);
    }
}

void cambio_velocidad()
{
    if(estado_botvel != digitalRead(7)){
        estado_botvel = digitalRead(7);

        if(estado_botvel == LOW){
            if(vel_actual != 0){
                if(vel_actual == 255){
                    vel_actual = 85;
                }

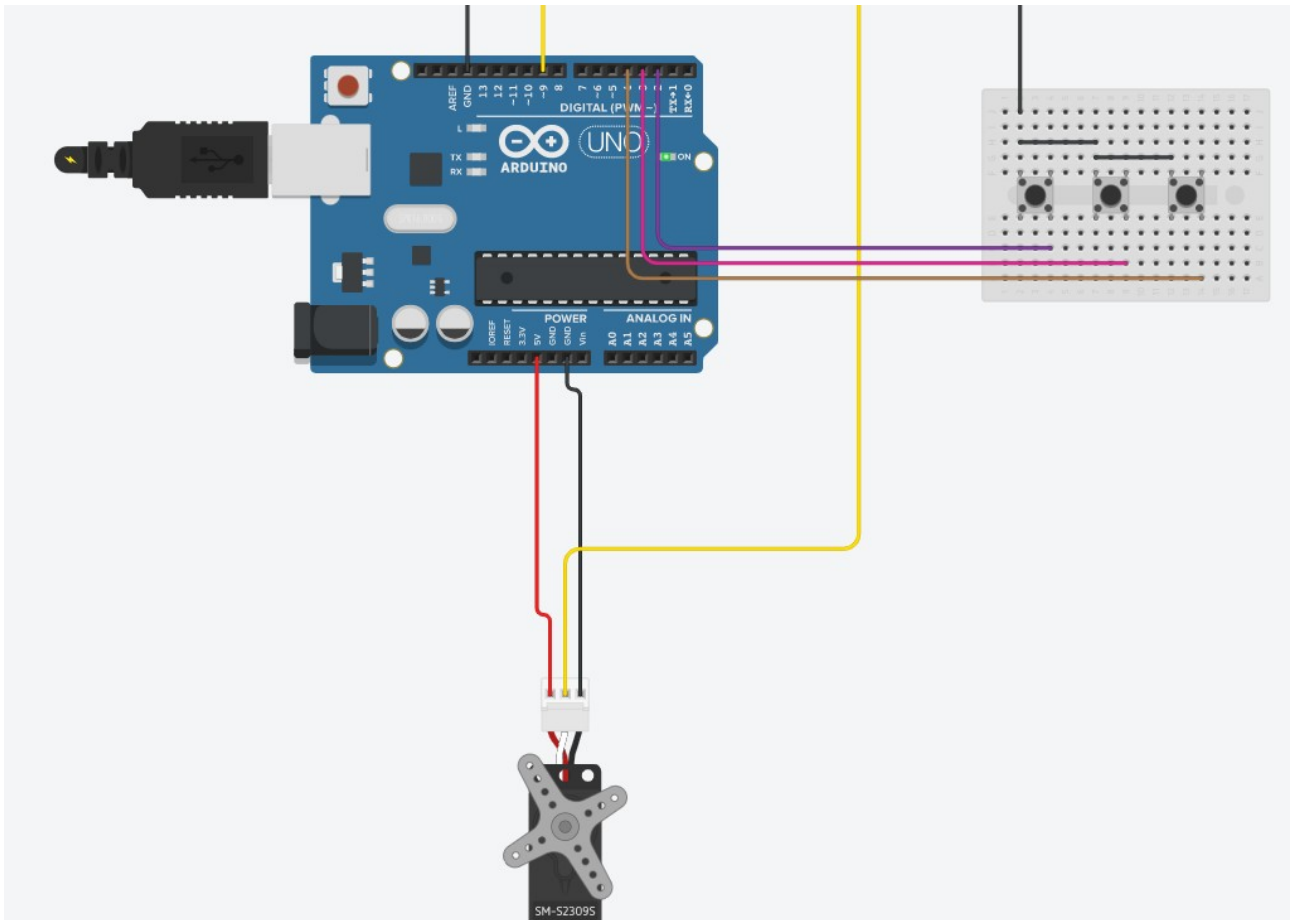
                else{
                    vel_actual += 85;
                }

                analogWrite(6, vel_actual);
            }
        }
    }
}

void loop()
{
    power_off();
    cambio_sentido();
    cambio_velocidad();
}

```

## Captura de pantalla del funcionamiento del servo en la posición de 60°



### Código fuente

```
// C++ code
//
#include <Servo.h>
Servo Miservo;
bool boton_giropositivo;
bool boton_gironegativo;
bool boton_reinicio;

int angulo_act;

void setup()
{
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(4, INPUT_PULLUP);
```

```
Miservo.attach(9);
Miservo.write(0);

angulo_act=0;

boton_giropositivo = digitalRead(2);
boton_gironegativo = digitalRead(3);
boton_reinicio = digitalRead(4);

}

void giro30to180(){
  if(boton_giropositivo != digitalRead(2)){
    boton_giropositivo = digitalRead(2);

    if(boton_giropositivo == LOW){
      if(angulo_act < 180)
        angulo_act += 30;

      Miservo.write(angulo_act);
    }
  }
}

void giro30to0(){
  if(boton_gironegativo != digitalRead(3)){
    boton_gironegativo = digitalRead(3);

    if(boton_gironegativo == LOW){
      if(angulo_act > 0)
        angulo_act -= 30;

      Miservo.write(angulo_act);
    }
  }
}
```

```
void reinicio_giro(){

    if(boton_reinicio != digitalRead(4)){
        boton_reinicio = digitalRead(4);

        if(boton_reinicio == LOW){
            for(int i = angulo_act-1; i >= 0; i--){
                Miservo.write(i);
                delay(20);
            }

            delay(1000);

            Miservo.write(angulo_act);
        }
    }
}

void loop()
{
    giro30to180();
    giro30to0();
    reinicio_giro();
}
```