



UNIVERSIDAD AUTÓNOMA DE  
**CHIHUAHUA**

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA  
Facultad de Ingeniería



Ingeniería en Ciencias de la Computación

## **SIMULACIÓN DE SISTEMAS**

### **Reporte sobre control de velocidad de motor DC con PWM**

*Trabajo de:*

- ADRIAN (ADORA) GONZÁLEZ DOMÍNGUEZ [359834]
- JOSÉ ANGEL ORTÍZ MERAZ [353195]

*Asesor:* OSCAR RAMSES RUIZ VARELA

*1 de noviembre de 2024*

# Diagramas

Diagrama ESP-32

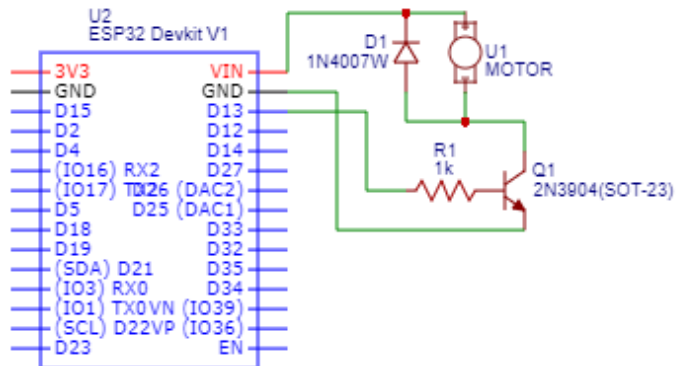
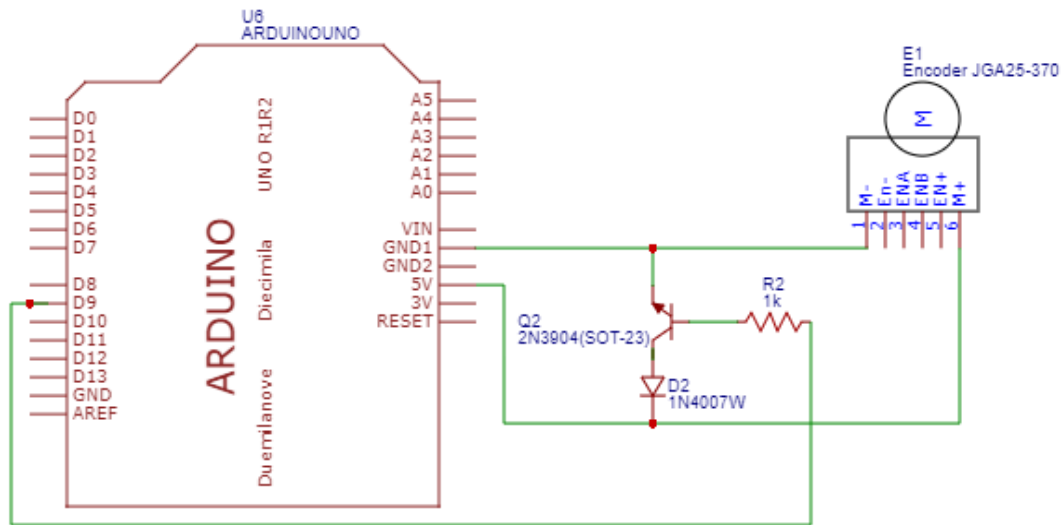
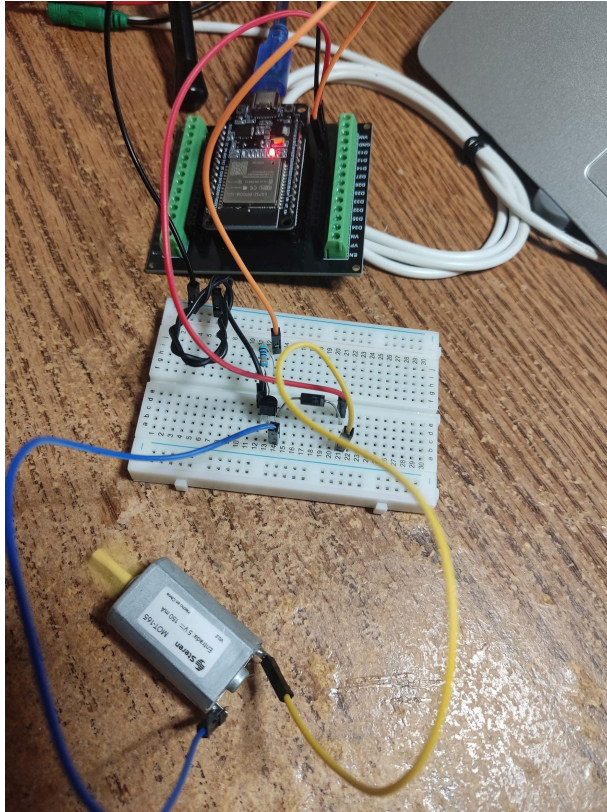


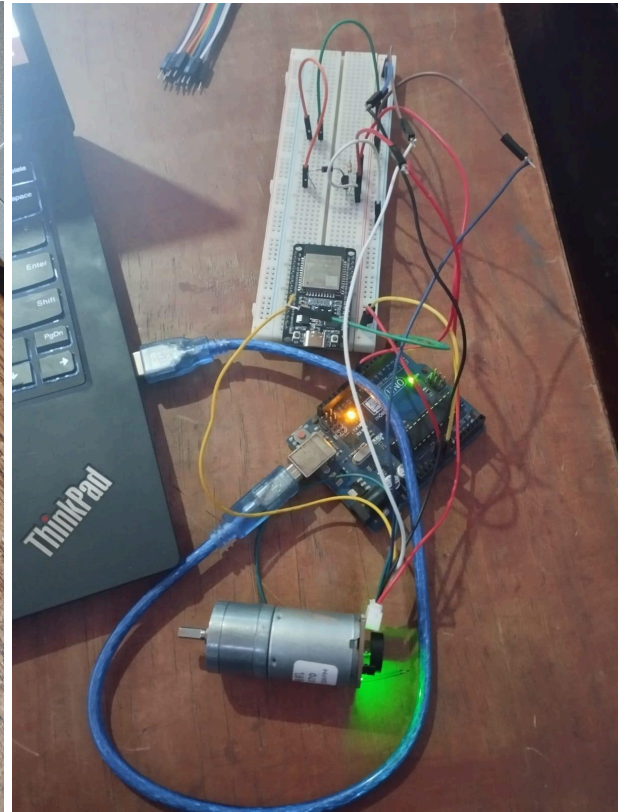
Diagrama Arduino Uno



## Evidencia



Placa ESP-32



Placa Arduino Uno

## Código

### Código ESP-32

```
#include <Arduino.h>
#include <esp32-hal-gpio.h>

#define MIN_DUTY_CYCLE 25
#define MAX_DUTY_CYCLE 200

const int ledChannel = 0;
const int transistorPin = GPIO_NUM_13;
const int frequency = 5000;
const int resolution = 8;
volatile int dutyCycle = MIN_DUTY_CYCLE;
```

```

void (*updateSpeed) ();
void increaseSpeed();
void decreaseSpeed();

void setup()
{
    Serial.begin(9600);
    ledcSetup(ledChannel, frequency, resolution);
    ledcAttachPin(transistorPin, ledChannel);
    updateSpeed=increaseSpeed;
}

void loop()
{
    updateSpeed();
    Serial.printf("Duty Cycle: %d\n", dutyCycle);
    delay(50);
}

void increaseSpeed()
{
    dutyCycle += 2;
    if (dutyCycle >= MAX_DUTY_CYCLE)
    {
        updateSpeed=decreaseSpeed;
        dutyCycle = MAX_DUTY_CYCLE;
    }
    ledcWrite(ledChannel, dutyCycle);
}

void decreaseSpeed()
{
    dutyCycle -= 2;
    if (dutyCycle <= MIN_DUTY_CYCLE)
    {
        updateSpeed=increaseSpeed;
        dutyCycle = MIN_DUTY_CYCLE;
    }
    ledcWrite(ledChannel, dutyCycle);
}

```

Código ESP-32

```
const int motorPin = 9; // Pin PWM
void setup() {
    pinMode(motorPin, OUTPUT); // pin del motor como salida
}
void loop() {
    // Incrementar gradualmente el ciclo de trabajo para aumentar
    la velocidad del motor
    for (int dutyCycle = 0; dutyCycle <= 255; dutyCycle++) {
        analogWrite(motorPin, dutyCycle); // Ajustar la velocidad
        del motor
        delay(50);
    }
    // Decrementar gradualmente el ciclo de trabajo para reducir
    la velocidad del motor
    for (int dutyCycle = 255; dutyCycle >= 0; dutyCycle--) {
        analogWrite(motorPin, dutyCycle); // Ajustar la velocidad
        del motor
    }
    delay(50);
}
```

## **Valores mínimos del PWM**

Los valores mínimos y máximos se calcularon de manera empírica.

El motor para poder arrancar necesita una mayor velocidad de inicio, en el caso de un motor [MOT-165] de 5V y 150mA, el motor se detiene por completo si se coloca un valor por debajo de los 25 ciclos de trabajo, entonces se utilizó este valor como el mínimo.

Por último, a diferencia de un led, si al motor se le indica el ciclo de trabajo mayor a 250, se detiene por completo, sin embargo decidimos utilizar un valor de 200 para no llegar al límite siempre.

El motor con encoder GA25-370 de 12V responde de manera efectiva en todo el rango de PWM, de 0 a 255. Esto significa que con valores de ciclo de trabajo cercanos a 0, el motor se encuentra casi detenido, y a medida que se incrementa el PWM hasta 255, el motor alcanza su máxima velocidad sin problemas de saturación o detención inesperada.