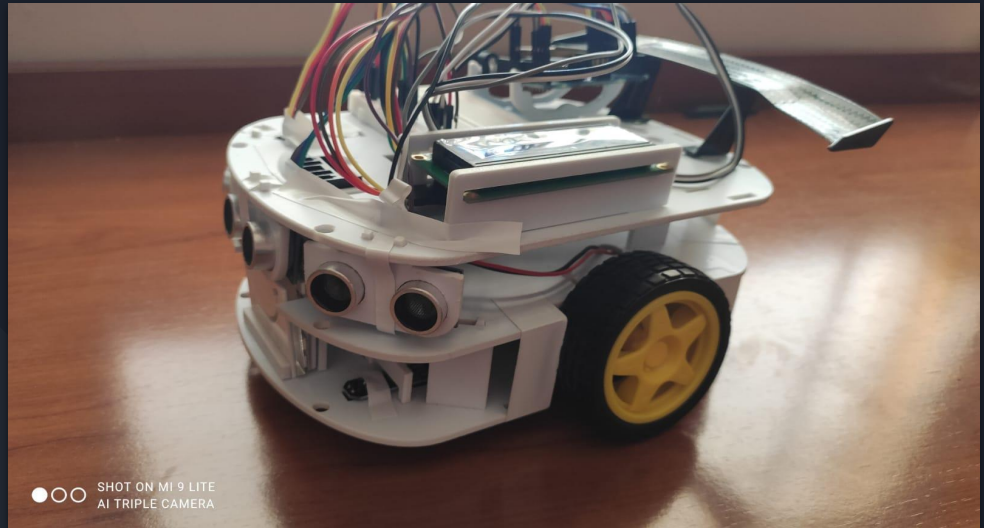


Blade IV

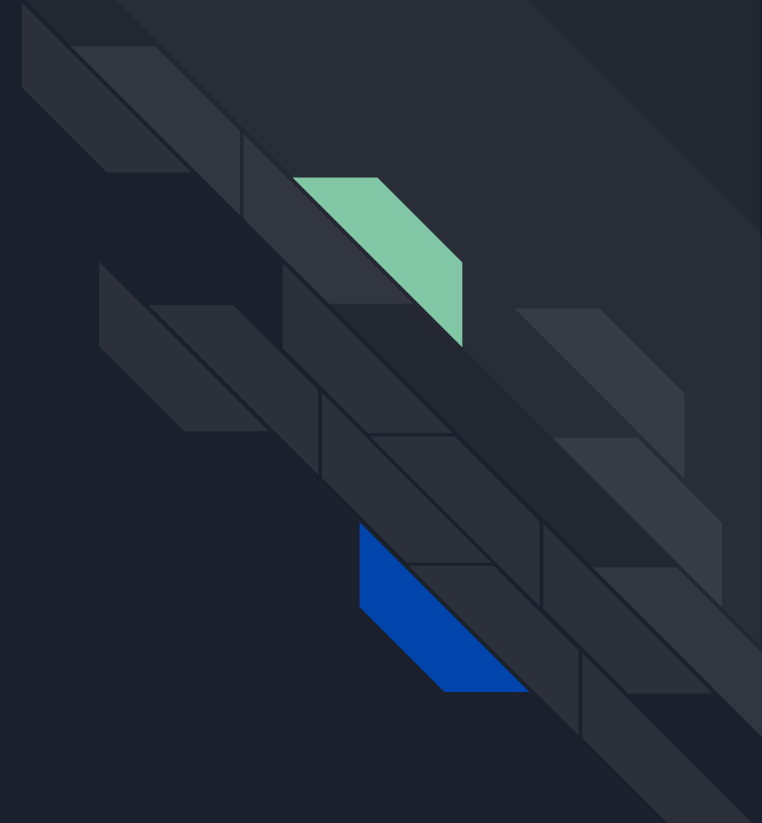
Autores:

Joaquín Sergio García Ibáñez
Juan Navarro Maldonado



Índice

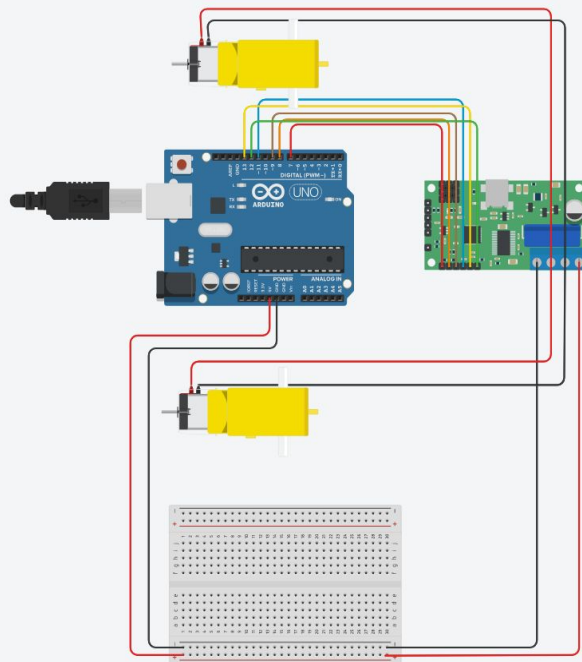
- Esquema
 - PWM
 - Sensores de Ultrasonidos
 - Infrarrojos
 - LCD
- Códigos
 - PWM
 - Sensores de Ultrasonidos
 - Infrarrojos
 - LCD
- Complicaciones



Esquemas TinkerCad



Esquema PWM



Conexión de Pines:

ENA: PIN 9

IN1: PIN 7

IN2: PIN 8

IN3: PIN 13

IN4: PIN 12

ENB: PIN 11

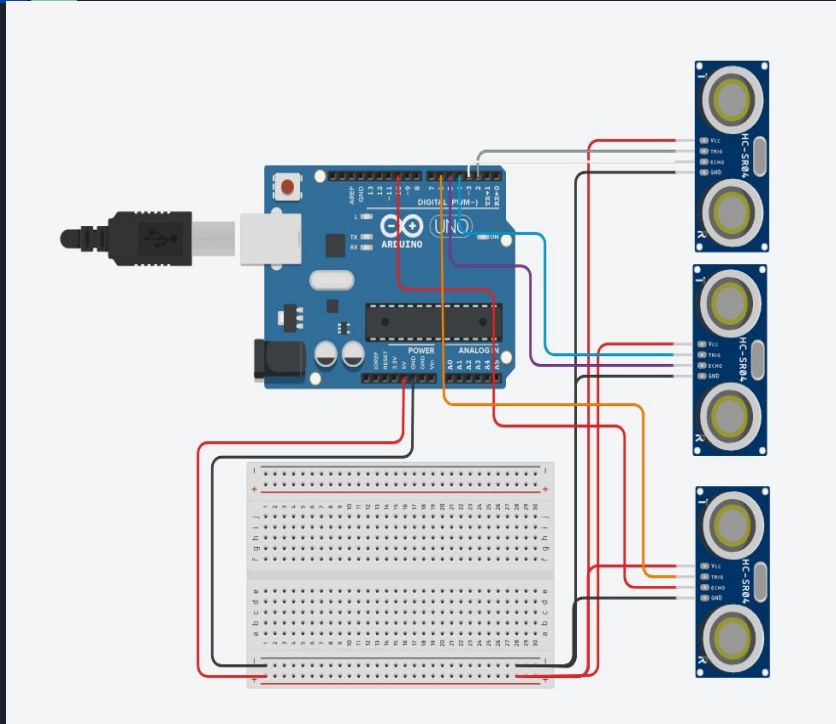
IN1 e IN2 controlan el sentido del motor izquierdo. ENA controla la velocidad del motor izquierdo.

IN3 e IN4 controlan el sentido del motor derecho. ENB controla la velocidad del motor derecho.

Para que un motor vaya hacia delante IN1 debe de estar LOW y IN2 en HIGH, además IN3 debe de estar LOW y IN4 en HIGH.

Esquema Sensores Ultrasonidos

Conexión de Pines



UltraSonido izquierdo:

GND: Tierra

Echo: PIN 3

Trigger: PIN 2

Vcc: Voltaje

UltraSonido Centro::

GND: Tierra

Echo: PIN 5

Trigger: PIN 4

Vcc: Voltaje

UltraSonido Derecho:

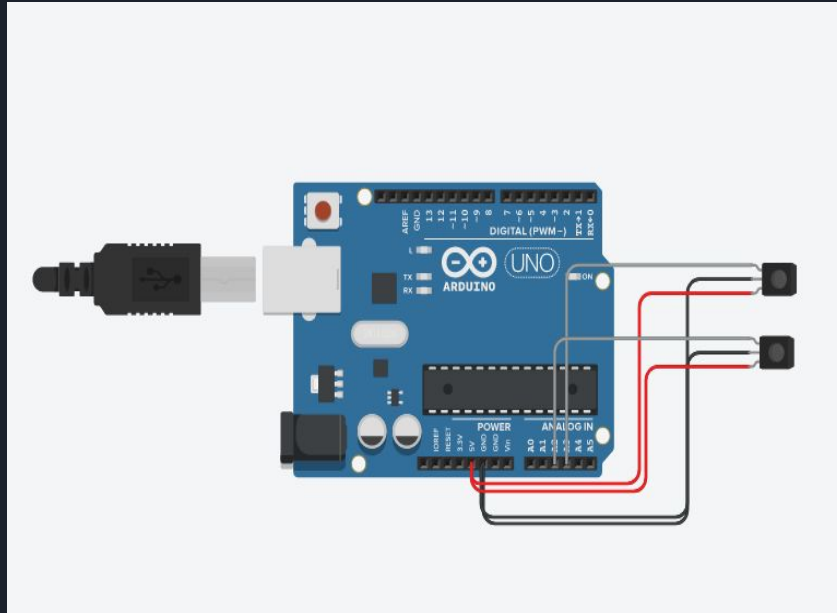
GND: Tierra

Echo: PIN 10

Trigger: PIN 6

Vcc: Voltaje

Esquema Infrarrojos



Conexión de Pines

Infrarrojo Izquierdo

GND: Tierra

Vout: PIN A2

Vcc: Voltaje

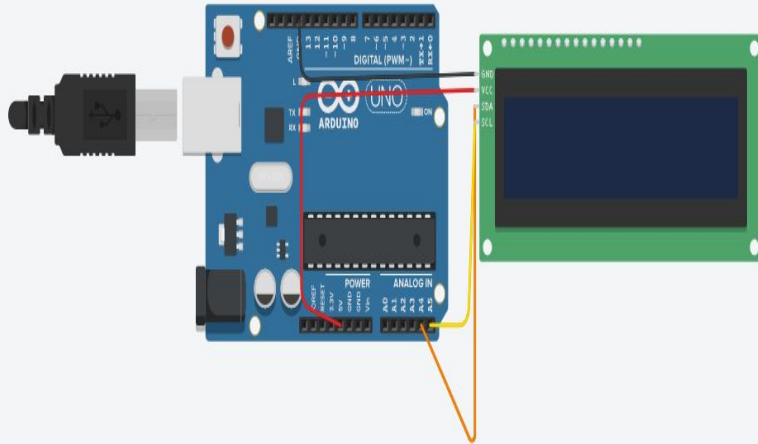
Infrarrojo Derecho

GND: Tierra

Vout: PIN A3

Vcc: Voltaje

Esquema LCD



Conexión de Pines

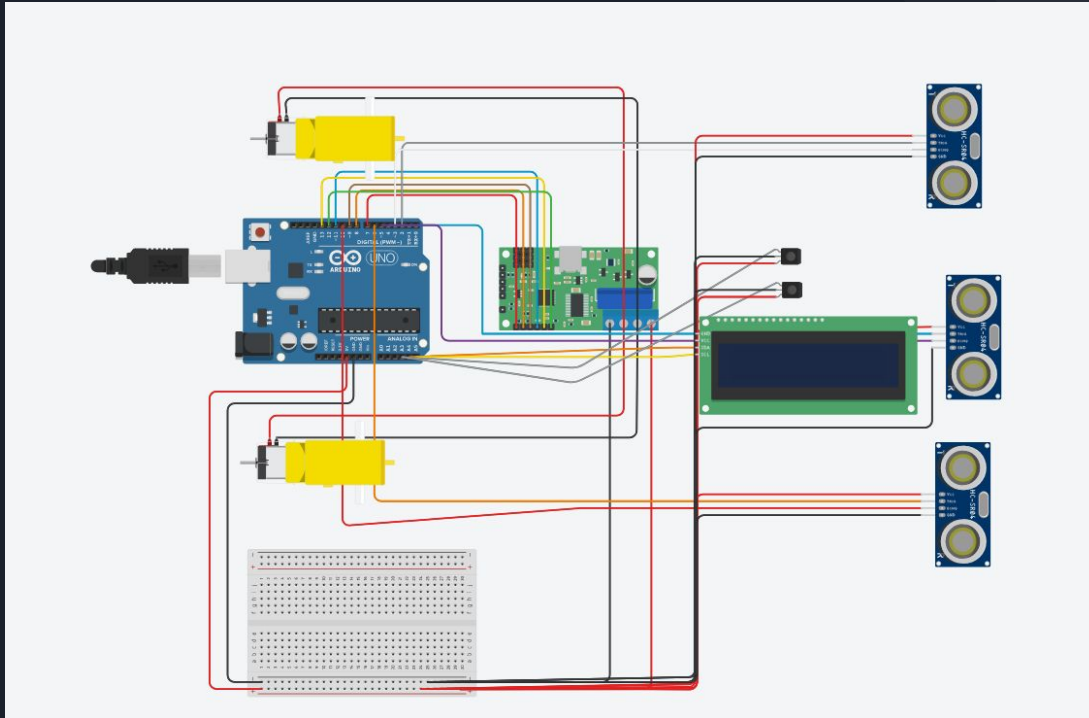
GND: Tierra

SLC: PIN A5

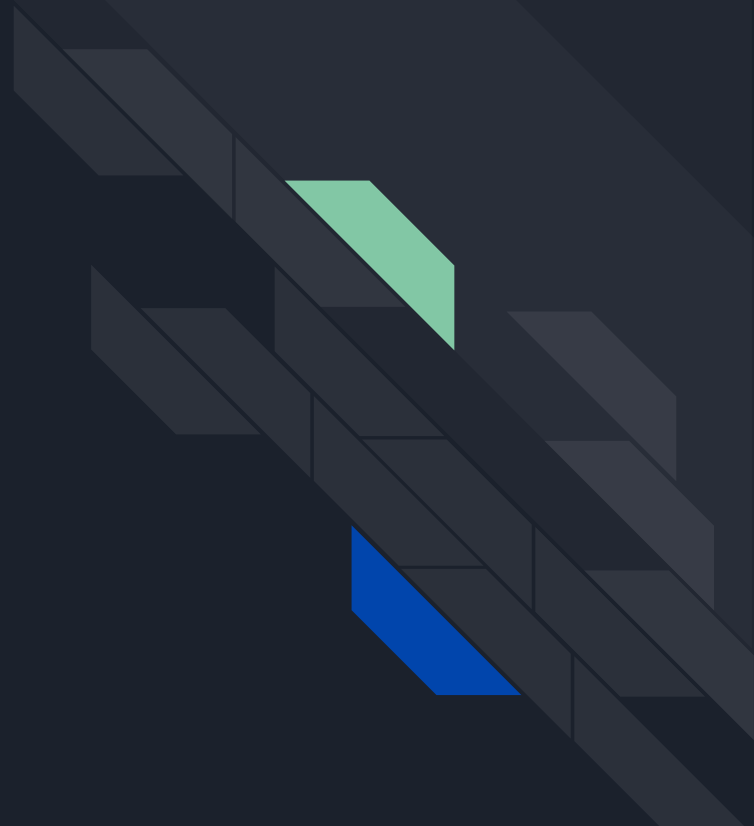
SDA: PIN A4

Vcc: Voltaje

Esquema final



Códigos



Código PWM

```
1 int IN1 = 7;    // IN1 de L298N a pin digital 2
2 int IN2 = 8;    // IN2 de L298N a pin digital 3
3 int ENA = 9;    // ENA de L298N a pin digital 5
4 int IN3 =13;
5 int IN4 =12;
6 int ENB=11;
7
8 void setup(){
9     pinMode(IN1, OUTPUT); // pin 2 como salida
10    pinMode(IN2, OUTPUT); // pin 3 como salida
11    pinMode(ENA,OUTPUT);
12    pinMode(IN3, OUTPUT); // pin 2 como salida
13    pinMode(IN4, OUTPUT); // pin 3 como salida
14    pinMode(ENB,OUTPUT);
15    // pin 5 como salida
16 }
17
18 void loop(){
19
20
21     // el valor de velocidad y aplica a ENA
22     digitalWrite(IN1, LOW); // IN1 en 1
23     digitalWrite(IN2, LOW);
24
25     analogWrite(ENA, 255);
26     // el valor de velocidad y aplica a ENA
27     digitalWrite(IN3, LOW); // IN1 en 1
28     digitalWrite(IN4, LOW); // IN2 en 0
29     analogWrite(ENB, 255);
30 }
```

Código Sensores de Ultrasonidos

```
1 const int Trigger_centro = 4; //Pin digital 2 para el Trigger del sensor
2 const int Echo_centro = 5; //Pin digital 3 para el echo del sensor
3 const int Trigger_izq = 2;
4 const int Echo_izq = 3;
5 const int Trigger_der = 6;
6 const int Echo_der = 10;
7
8 void setup() {
9   Serial.begin(9600); //inicializamos la comunicación
10  pinMode(Trigger_centro, OUTPUT); //pin como salida
11  pinMode(Echo_centro, INPUT); //pin como entrada
12  digitalWrite(Trigger_centro, LOW); //Iniciamos el pin con 0
13
14  pinMode(Trigger_izq, OUTPUT); //pin como salida
15  pinMode(Echo_izq, INPUT); //pin como entrada
16  digitalWrite(Trigger_izq, LOW); //Iniciamos el pin con 0
17
18  pinMode(Trigger_der, OUTPUT); //pin como salida
19  pinMode(Echo_der, INPUT); //pin como entrada
20  digitalWrite(Trigger_der, LOW); //Iniciamos el pin con 0
21 }
22 void loop()
23 {
24   long t_centro, t_izq, t_der, t_min1, t_min2; //tiempo que demora en llegar el eco
25   long d; //distancia en centímetros
26
27   //Distancia centro
28   digitalWrite(Trigger_centro, HIGH);
29   delayMicroseconds(10); //Enviamos un pulso de 10us
30   digitalWrite(Trigger_centro, LOW);
31
32   t_centro = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
33
34   //Distancia izq
35   digitalWrite(Trigger_izq, HIGH);
36   delayMicroseconds(10); //Enviamos un pulso de 10us
37   digitalWrite(Trigger_izq, LOW);
38
39   t_izq = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
40
41   //Distancia der
42   digitalWrite(Trigger_der, HIGH);
43   delayMicroseconds(10); //Enviamos un pulso de 10us
44   digitalWrite(Trigger_der, LOW);
```

```
45
46   //Distancia der
47   digitalWrite(Trigger_der, HIGH);
48   delayMicroseconds(10); //Enviamos un pulso de 10us
49   digitalWrite(Trigger_der, LOW);
50
51   t_der = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
52
53   t_min1 = min(t_centro, t_izq);
54   t_min2 = min(t_min1, t_der);
55
56   if(t_min2 == t_centro){
57     //avanza
58     Serial.println("Centro");
59   }
60   if(t_min2 == t_der){
61     //girader
62     Serial.println("Derecha");
63   }
64   if(t_min2 == t_izq){
65     //Giraizq
66     Serial.println("Izquierda");
67   }
68
69   d = t_min2/59; //escalamos el tiempo a una distancia en cm
70
71   Serial.print("Distancia: ");
72   Serial.print(d); //Enviamos serialmente el valor de la distancia
73   Serial.print("cm");
74   Serial.println();
75   delay(100); //Hacemos una pausa de 100ms
76 }
```

Código Infrarrojos

```
1  const int sensorPin_izq = A2;
2  const int sensorPin_der = A3;
3
4  void setup() {
5      Serial.begin(9600); //iniciar puerto serie
6      pinMode(sensorPin_izq , INPUT); //definir pin como entrada
7      pinMode(sensorPin_der , INPUT); //definir pin como entrada
8  }
9
10 void loop(){
11     int value_izq = 0;
12     int value_der = 0;
13     value_izq = digitalRead(sensorPin_izq ); //lectura digital de pin
14     value_der = digitalRead(sensorPin_der ); // lectura digital del p
15
16     if (value_izq == 0) {
17         Serial.println("Detectado obstaculo izquierda");
18     }
19     if (value_der == 0) {
20         Serial.println("Detectado obstaculo derecha");
21     }
22     delay(1000);
23 }
```

Código LCD

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  //Crear el objeto lcd dirección 0x3F y 16 columnas x 2 filas
5  LiquidCrystal_I2C lcd(0x27,16,2); //
6
7  void setup() {
8      // Inicializar el LCD
9      lcd.init();
10
11     //Encender la luz de fondo.
12     lcd.backlight();
13
14     // Escribimos el Mensaje en el LCD.
15     lcd.print("Hola Mundo");
16 }
17
18 void loop() {
19     // Ubicamos el cursor en la primera posición(columna:0) de la segunda línea(fila:1)
20     lcd.setCursor(0, 1);
21     // Escribimos el número de segundos transcurridos
22     lcd.print(millis()/1000);
23     lcd.print(" Segundos");
24     delay(100);
25 }
```

Complicaciones

