

CODIGO PARA EL MOTOR Y SERVOMOTOR:

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

```
// Definición de pines para el motor DC
```

```
const int pinIN1 = 11; // Control de dirección 1
```

```
const int pinIN2 = 12; // Control de dirección 2
```

```
const int pinENA = 10; // Control PWM
```

```
// Crear un objeto Servo
```

```
Servo miServo;
```

```
// Definir el pin donde está conectado el servomotor
```

```
const int pinServo = 9;
```

```
void setup() {
```

```
    // Configurar los pines del motor como salida
```

```
    pinMode(pinIN1, OUTPUT);
```

```
    pinMode(pinIN2, OUTPUT);
```

```
    pinMode(pinENA, OUTPUT);
```

```
    // Asociar el objeto Servo al pin
```

```
    miServo.attach(pinServo);
```

```
}
```

```
// Funciones para controlar el servomotor
```

```
void moverIzquierda() {
```

```
miServo.write(60);  
delay(1000); // Esperar 1 segundo  
}
```

```
void moverCentro() {  
    miServo.write(90);  
    delay(1000); // Esperar 1 segundo  
}
```

```
void moverDerecha() {  
    miServo.write(120);  
    delay(1000); // Esperar 1 segundo  
}
```

```
// Funciones para controlar el motor DC  
void retroceder() {  
    digitalWrite(pinIN1, HIGH);  
    digitalWrite(pinIN2, LOW);  
    analogWrite(pinENA, 255); // Velocidad máxima  
    delay(2000); // Avanzar por 2 segundos  
    detenerMotor(); // Detener el motor  
}
```

```
void avanzar() {  
    digitalWrite(pinIN1, LOW);  
    digitalWrite(pinIN2, HIGH);
```

```
    analogWrite(pinENA, 255); // Velocidad máxima  
    delay(2000); // Retroceder por 2 segundos  
    detenerMotor(); // Detener el motor  
}
```

```
void detenerMotor() {  
    analogWrite(pinENA, 0); // Detener el motor  
}
```

```
void loop() {  
    // Control del motor DC  
    avanzar();  
    delay(1000); // Esperar 1 segundo  
    retroceder();  
    delay(1000); // Esperar 1 segundo
```

```
    // Control del servomotor  
    moverIzquierda();  
    moverCentro();  
    moverDerecha();  
}
```

## CÓDIGO DE UN SOLO SENSOR:

```
#include <NewPing.h>

// Definición de pines para el sensor ultrasónico HC-SR04
#define TRIGGER_PIN 8
#define ECHO_PIN 7

#define MAX_DISTANCE 200 // Distancia máxima en centímetros para medir (en este
caso, 200 cm o 2 metros)

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // Crea un objeto
NewPing

void setup() {
  Serial.begin(9600); // Inicializa la comunicación serial
}

void loop() {
  delay(1000); // Espera medio segundo entre cada medición

  // Realiza una medición de distancia
  unsigned int distancia = sonar.ping_cm();

  // Muestra la distancia medida en el monitor serial
  Serial.print("Distancia: ");
  if (distancia == 0) {
    Serial.println("Fuera de rango");
  } else {
```

```
Serial.print(distancia);  
Serial.println(" cm");  
}  
}
```