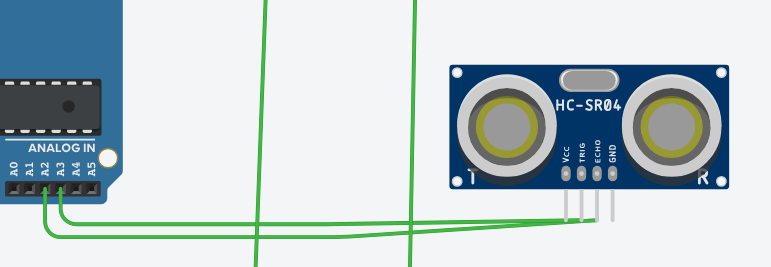
Conexiones por separado



Sensor HC-SR04

Codigo:

void loop() {

int distancia = medirDistancia(); // Llama a la función para medir distancia

Serial.print("Distancia: ");

Serial.println(distancia);

if (distancia == 0 || distancia >= STOP\_DIST) {

moverAdelante();

digitalWrite(ledVerde, HIGH);

digitalWrite(ledRojo, LOW);

} else {

detener();

digitalWrite(ledRojo, HIGH);

digitalWrite(ledVerde, LOW);

girarServo();

}

delay(100);

}

int medirDistancia() {

// Envía un pulso de 10 microsegundos para iniciar la medición

digitalWrite(TRIGGER\_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIGGER\_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIGGER\_PIN, LOW);

// Lee el tiempo que tarda en recibir el eco

long duracion = pulseIn(ECHO\_PIN, HIGH);

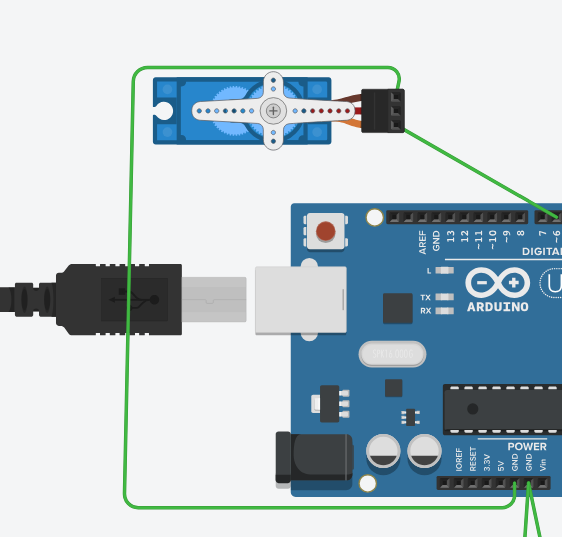
// Calcula la distancia en cm (velocidad del sonido: 343 m/s)

int distancia = duracion \* 0.034 / 2;

if (distancia > MAX\_DISTANCE) return 0; // Si está fuera del rango

return distancia;

}



Servo motor

Codigo:

void girarServo() {

myservo.write(0); // Gira a la izquierda

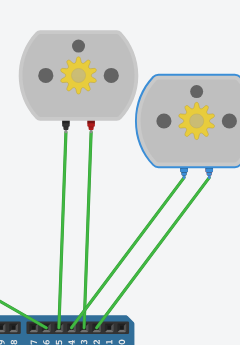
delay(500);

myservo.write(180); // Gira a la derecha

delay(500);

myservo.write(90); // Regresa al centro

}



Motores CC

Codigo:

void moverAdelante() {

digitalWrite(motorA1, HIGH);

digitalWrite(motorA2, LOW);

digitalWrite(motorB1, HIGH);

digitalWrite(motorB2, LOW);

}

void detener() {

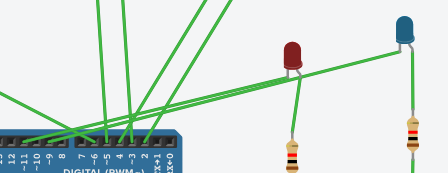
digitalWrite(motorA1, LOW);

digitalWrite(motorA2, LOW);

digitalWrite(motorB1, LOW);

digitalWrite(motorB2, LOW);

}



Leds

Codigo:

if (distancia == 0 || distancia >= STOP\_DIST) {

moverAdelante();

digitalWrite(ledVerde, HIGH);

digitalWrite(ledRojo, LOW);

} else {

detener();

digitalWrite(ledRojo, HIGH);

digitalWrite(ledVerde, LOW);

girarServo();

}