Proyecto r2d2

El objetivo de este proyecto es crear a un r2d2. Emite sonidos reales. Gira la cabeza 180 grados. Se desplaza a cualquier dirección. Evita Obstáculos orientado mediante una brújula. Reacciona al contacto corporal. Seria posible controlar sus movimientos mediante un mando o un software bluetooth. Expresa “emociones” mediante sonidos, luces y movimientos

* Materiales Necesarios:

2 Motores TT.

1 Arduino Nano

Driver puente H, LN298N.

Servo motor MG90S.

Batería 18650.

Módulo Cargador USB de batería litio 18650 modelo 03962A.

Convertidor DC Step-Up Booster.

Interrurtor de palanca.

Rodamiento 608ZZ (22x8x7mm).

Rodamiento de bola de metal (pata central).

Tornillos y tuercas M3.

Tornillos autorroscantes 3×10mm.

Componentes adicionales:

Brújula HCM5883L

Módulo BlueTooth.

5 Leds RGB de cátodo común.

3 transistores 2N2222A.

6 resistencias 1K, 300 ohm.

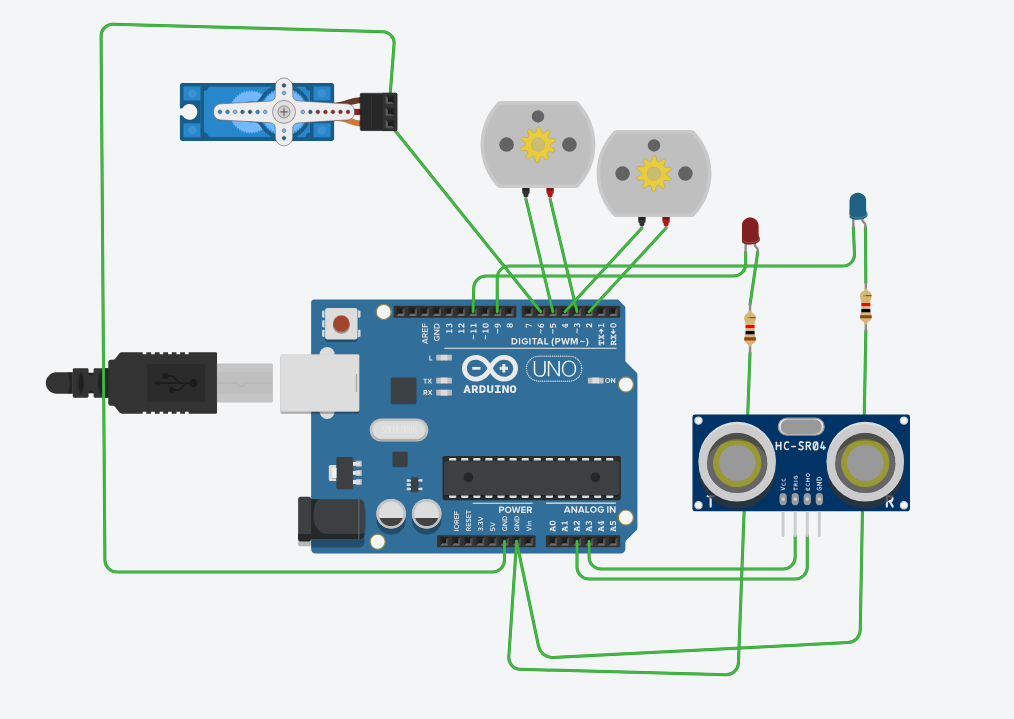
Sensor ultrasonidos HC-SR04.

Sensor capacitivo de contacto TTP223B.

Reproductor de mp3 Open-Smart.

Lente de cristal (de linternas comerciales).

Codigo basico para el proyecto

#include <Servo.h>

#define TRIGGER\_PIN A3

#define ECHO\_PIN A2

#define MAX\_DISTANCE 200

#define STOP\_DIST 35

Servo myservo;

int ledRojo = 9;

int ledVerde = 11;

int motorA1 = 3, motorA2 = 5;

int motorB1 = 2, motorB2 = 4;

int servoPin = 6;

void setup() {

pinMode(ledRojo, OUTPUT);

pinMode(ledVerde, OUTPUT);

pinMode(motorA1, OUTPUT);

pinMode(motorA2, OUTPUT);

pinMode(motorB1, OUTPUT);

pinMode(motorB2, OUTPUT);

pinMode(TRIGGER\_PIN, OUTPUT);

pinMode(ECHO\_PIN, INPUT);

myservo.attach(servoPin);

myservo.write(90); // Posición inicial del servo

Serial.begin(9600);

}

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;

}

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);

}

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}