

Desarrolladores:

Miguel Angel Uribe Zuluaga

Juan Felipe Higuera Perez

David Alejandro Suárez Varón

Realizar la implementación física tanto del sensor LDR, cómo del manejo del puente H utilizando 2 Motores DC

Componentes principales:

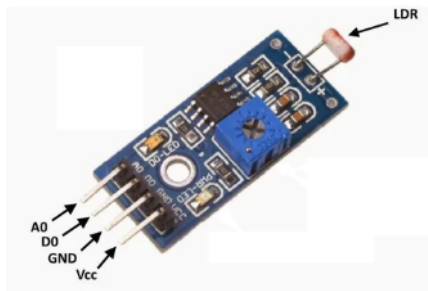
-Arduino UNO



-Protoboard:



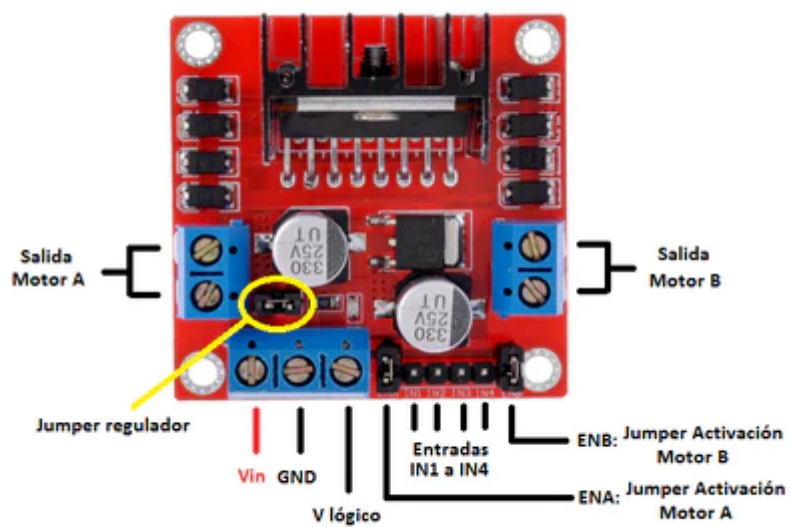
-sensor LDR analogico:



-sensor LDR digital:



-Puentes H L298N:



-2 Motores DC:



- Cables de conexión***
- 4 Pilas doble A***
- Plataforma para el robot:***



- Sensor de Ultrasonido:***



www.julpin.com.co



- Portapilas***
- Pila grande 9V***

Problemas:

Lo que hacía esta práctica diferente a las demás es que por primera vez obligatoriamente nos tocaba llevar el proyecto a una implementación física en vez de ser una simulación en tinkercad o en alguna plataforma, que claramente es algo más difícil ya que mientras que tu en una simulación no tienes que preocuparte por temas del cableado, en el montaje físico al ser algo más completo debes ser muy cuidadoso donde conectas cada uno porque por error podrias quemar el circuito, además de que se deber ser muy cuidadoso con los componentes para no estropearlos al ser algo más delicados, estos fueron algunos problemas que se nos presentaron en este proyecto.

Desarrollo:

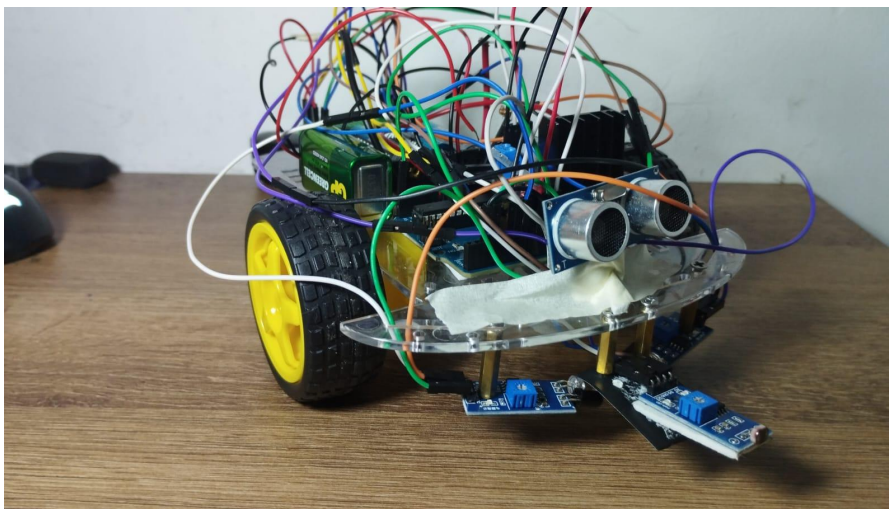
Primero lo que hicimos para iniciar con la creación del robot fue comprar un pequeño kit llamado kit chasis carro 2WD en el que ya venían algunos componentes por defecto como 2 motores DC, un interruptor para encender y apagar el robot cuando lo tengamos completo, entre otras cosas. Compramos aparte el Arduino uno, el sensor LDR, el puente H y los cables de conexión para comenzar lo básico. Siguiendo algunas instrucciones que venían en el kit ensamblamos en la plataforma lo que son los motores, el interruptor, las ruedas del robot y el portapilas con tornillos que venían en el kit para así tener la plataforma completa y poder ensamblar lo demás a esta, así que el siguiente paso fue colocar el arduino UNO a la plataforma con el cual daríamos inicio a colocar otro componentes del mismo estilo como el puente H y el LDR.

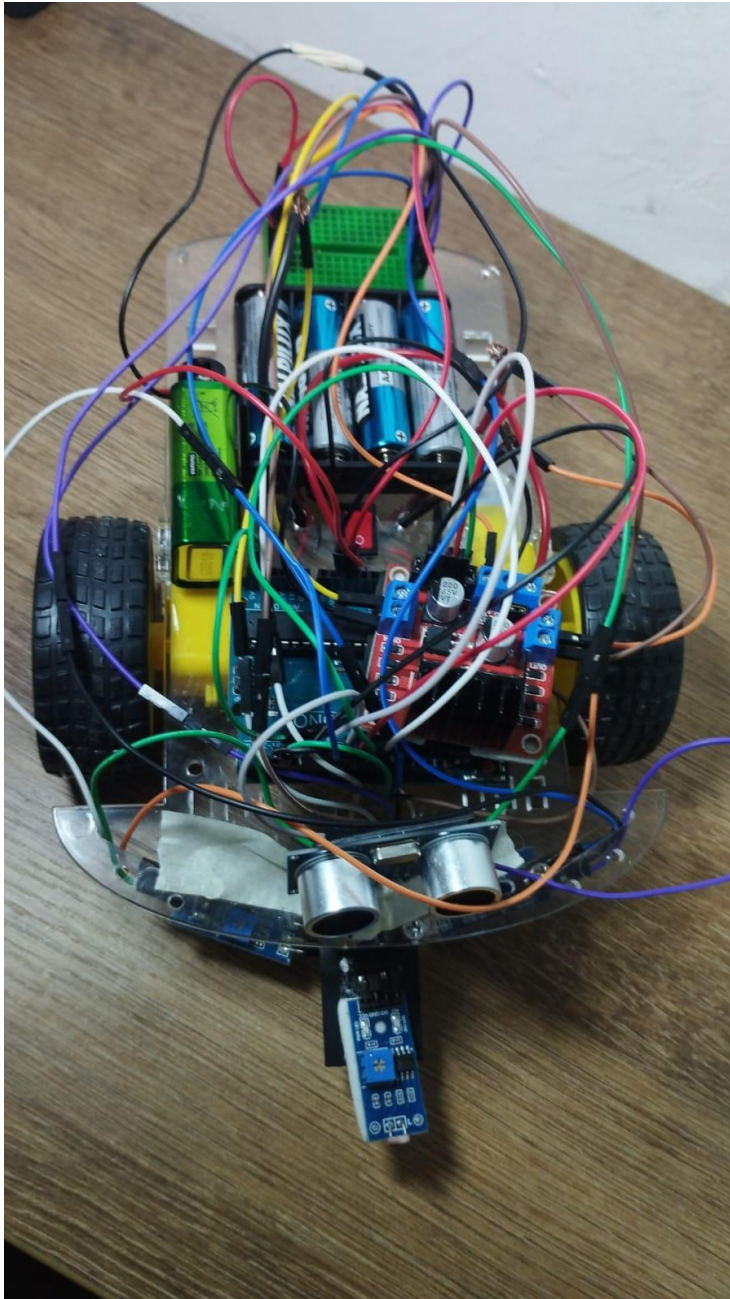
Ahora pasamos a la conexiones en las cuales las dos terminales de cada motor las colocamos en cada salida para motores del puente H con unos cables de conexión positivo y negativo con los que conectaremos con tornillos en cada salida de ese dispositivo, ahora en cada entrada del puente H (IN1, IN2, IN3, IN4) los conectaremos a un pin de entrada digital en el arduino UNO, por ejemplo nosotros conectamos las entradas digitales 5 y 6 que le corresponden a un motor que llamaremos motor 1 mientras que al otro le conectamos las entradas digitales 9 y 10, por último conectamos el voltaje del puente H a 5V y el GND a GND del arduino los enables no lo tocamos en ningún momento del trabajo. Una vez hecho esto pasaremos a conectar el sensor LDR digital al arduino UNO, entonces lo que hacemos es simplemente conectar el voltaje de

este a uno del arduino, el tierra igualmente y ya que este solo posee una entrada digital se conectara a un pin digital aunque si fuera uno de 4 patas que es el analogico y el que luego agregamos solo seria conectar la entrada analogica que en este caso sera A5.

Una vez conectamos lo primordial, solo queda conectar el portapilas para así meter las 4 pilas doble A y conectarlas tanto en el interruptor como en el puente H, para que cada vez que presionemos el interruptor se apague o encienda este componente, pero estas baterías no son suficiente para alimentar todo el circuito por esa razon tambien necesitamos una batería grande de 9V para alimentar el arduino, también compramos una mini protoboard para hacerlo más cómodo y que no se vea tanto tumulto de cables conectado, teniendo la tierra y el voltaje conectados a la protoboard para cualquier otro componente, se conecte por medio de la protoboard y no del arduino. Por último conectamos tenemos el sensor de ultrasonido, este en un principio tiene aparte del GND y el que recibe el voltaje, tiene un pin llamado Trigger que es de salida y Echo que es un pin de entrada, ambos pines lo ponemos en entradas digitales del arduino.

Resultado final:





Realice una prueba de presencia o ausencia de LUZ, para ello utilice el sensor LDR para que el vehículo avance hacia adelante por 5 segundos si el nivel de luz es mayor al 50% del rango del sensor LDR. De lo contrario debe reversar 3 segundos.

Desarrollo:

Ahora en este reto nos dicen que de acuerdo a lo que ensamblamos, hagamos un código en el que si detecta más de 50% de luz se mueva hacia delante 5 segundos y si no que se mueva hacia atras durante 3 segundos, bien,

entonces aquí vamos a necesitar en el código el pin LDR digital que usamos en este reto y el puente H para poder inicializar los pines de los motores:

```
// C++ code
//
// Define los pines para el LDR y el módulo L293D
const int ldrPin = 11; // Pin digital donde está conectado el
const int motor1Pin1 = 5; // Pin 1 del motor 1
const int motor1Pin2 = 3; // Pin 2 del motor 1
const int motor2Pin1 = 6; // Pin 1 del motor 2
const int motor2Pin2 = 10; // Pin 2 del motor 2

void setup() {
  // Configura los pines del motor como salidas
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);
}

void loop() {
  // Lee el valor del LDR (HIGH/LOW)
  int ldrValue = digitalRead(ldrPin);

  // Comprueba si el LDR detecta luz
  int threshold = 512; // Este valor puede ajustarse según la calib:
```

Aquí definimos los pines que están conectados al puente H y el pin digital que está conectado al LDR, luego en void setup inicializamos las variables donde los pines de los motores van a ser salidas, ahora pasamos a void loop donde asignamos a una variable la cantidad de luz que detecte el sensor LDR desde su salida digital y definimos otra variable la cual va a representar el 50% de luz que va a ser 512.

```

// Comprueba si la luz es mayor al 50% del rango
if (ldrValue > threshold) {
  // Si hay suficiente luz, avanza durante 5 segundos
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
  delay(5000);
} else {
  // Si no hay suficiente luz, retrocede durante 3 segundos
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, HIGH);
  delay(3000);
}

// Detiene los motores
digitalWrite(motor1Pin1, LOW);
digitalWrite(motor1Pin2, LOW);
digitalWrite(motor2Pin1, LOW);
digitalWrite(motor2Pin2, LOW);
delay(2000);

```

En esta siguiente parte vamos a comparar la lectura de luz que leyó el LDR y el valor que asignamos para el 50%, si la lectura es mayor al 50% los primeros pines de los motores DC van a entrar en HIGH mientras que los segundos pines van a estar en LOW haciendo que el carro avance durante 5 segundos pero si no es así, los primeros pines estarán en LOW mientras que los dos últimos estarán en LOW haciendo que el carro retroceda por 3 segundos, una vez hecha esa comparación todos los pines entraran en estado LOW durante 2 segundos y se repetirá el proceso hasta que apaguemos el circuito.

Genere un esquema de pruebas en donde su robot gire a la derecha o izquierda de acuerdo a una condición de su elección.

Problemas y consideraciones:

Uno de los problemas que se nos presentó fue el manejo de implementar bien el código al arduino para su correcto funcionamiento, una cosa es hacer un código en una simulación donde tienes diferentes elementos (un ejemplo es tinkercad en el cual no todos componentes están y toca reemplazarlos por los que hay ahí) que carga el código al carro en físico, así que nos tocó cambiar varias veces el código que teníamos para adaptarlo a lo que teníamos.

Desarrollo:

En este reto decidimos usar el sensor de ultrasonido para que dependiendo a la distancia que el sensor detecte un objeto, este se mueve a la derecha o a la izquierda, además aquí cambiamos el LDR digital por uno analogico para que sea mas facil detectar los niveles de luz:

```
// Define los pines para el LDR y el módulo L293D
const int motor1Pin1 = 5; // Pin 1 del motor 1
const int motor1Pin2 = 6; // Pin 2 del motor 1
const int motor2Pin1 = 9; // Pin 1 del motor 2
const int motor2Pin2 = 10; // Pin 2 del motor 2
const int ldrDigitalPin = 11; // Pin analógico donde está cone
const int ldrAnalogPin = A5;
int echoPin = 3;
int trigPin = 2;
int duracion = 0;
int distancia = 0;

void setup() {
  // Configura los pines del motor como salidas
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
```

Primero definimos las variables las cuales algunas son las misma del reto anterior, solo que aquí agregamos un pin analogico para la salida analogica el cual ser A5, los pines del sensor de ultrasonido los cuales van ser 2 para trigger y el pin 3 que va ser para echo, también vamos agregar otras dos variables llamadas duración y distancia que van a ser 0, ahora inicializamos los pines en el void setup como hicimos antes solo que agregando como pin de salida el trigger y como pin de entrada el echo, también vamos agregar serial.begin para que monitorear los datos que pasan por el circuito.

```

void loop() {
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
  // Lee el valor del LDR (HIGH/LOW)
  digitalWrite(trigPin, LOW);
  delay(100);
  digitalWrite(trigPin, HIGH);
  delay(100);
  digitalWrite(trigPin, LOW);
  duracion = pulseIn(echoPin, HIGH);
  distancia = duracion * 0.034 / 2;
  Serial.println(distancia);
  // Lee el valor del LDR (HIGH/LOW)
  int ldrValueAnalog = analogRead(ldrAnalogPin);
  // Comprueba si el LDR detecta luz
  int threshold = 300; // Este valor puede ajustarse según la ca
  Serial.println(ldrValueAnalog);
}

```

Ahora pasando a void loop, configuramos que el carro se mueva siempre al poner el primer pin del motor 1 y el segundo pin del motor 2 en low mientras que los otros en HIGH, esto es debido a que nosotros conectamos de una manera diferente las terminales de los motores DC al puente H entonces por eso debemos colocar el código así para que avance siempre, luego ponemos en bajo a trigger para luego ponerlo en alto y luego ponerlo en bajo para ver cuánto tiempo tarda en rebotar la señal ultrasónica del componente, seguido de eso cambiamos el valor de duración por la cantidad de tiempo que demora el eco en cambiar de estado mientras el eco que se produce por la señal regrese, una vez hecho eso hacemos un pequeño cálculo para sacar la distancia que se encuentra el objeto del robot. Usamos serial.println para que nos imprima cual fue el valor calculado y monitorearlo, ahora hacemos una nueva variable de lo que la salida analógica del LDR leyó del nivel de luz del lugar en el que se encuentra el robot.

```

Serial.println(ldrValueAnalog);

// Comprueba si la luz es mayor al 50% del rango
if (distancia > 30 && distancia < 50) {
    // Si hay suficiente luz, avanza durante 5 segundos
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, LOW);
    delay(2000);
} else if (distancia <= 30) {
    // Si no hay suficiente luz, retrocede durante 3 segundos
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
    delay(2000);
}

```

aquí como en la duración vamos a usar serial.println para monitorear el valor que toma la salida de la analogica, ahora ponemos un condicional donde si la distancia es mayor de 30 pero a la vez menor que 50, se mueva hacia la derecha pero en caso de ser menor que 30 debe moverse a la izquierda, dependiendo del valor se moverá hacia una dirección durante dos segundos.

```

    if (ldrValueAnalog > 150 && ldrValueAnalog < 2000 ) {
        digitalWrite(motor1Pin1, LOW);
        digitalWrite(motor1Pin2, HIGH);
        digitalWrite(motor2Pin1, HIGH);
        digitalWrite(motor2Pin2, LOW);
        delay(2000);
    }
    else if (ldrValueAnalog < 140){
        digitalWrite(motor1Pin1, HIGH);
        digitalWrite(motor1Pin2, LOW);
        digitalWrite(motor2Pin1, LOW);
        digitalWrite(motor2Pin2, HIGH);
        delay(2000);
    }
}

```

Ahora para terminar, ponemos otros condicional donde si el valor que leyó el analogico si resulta mayor que 150 pero menor que 2000, va ir hacia delante pero en caso de que sea menor que 140 va a retroceder, en cualquiera de esas acciones va durar 2 segundos y se repetirá el ciclo hasta que apaguemos el circuito.

Link de video del funcionamiento del carro:

<https://www.youtube.com/shorts/hb1omJ9wQsQ>