

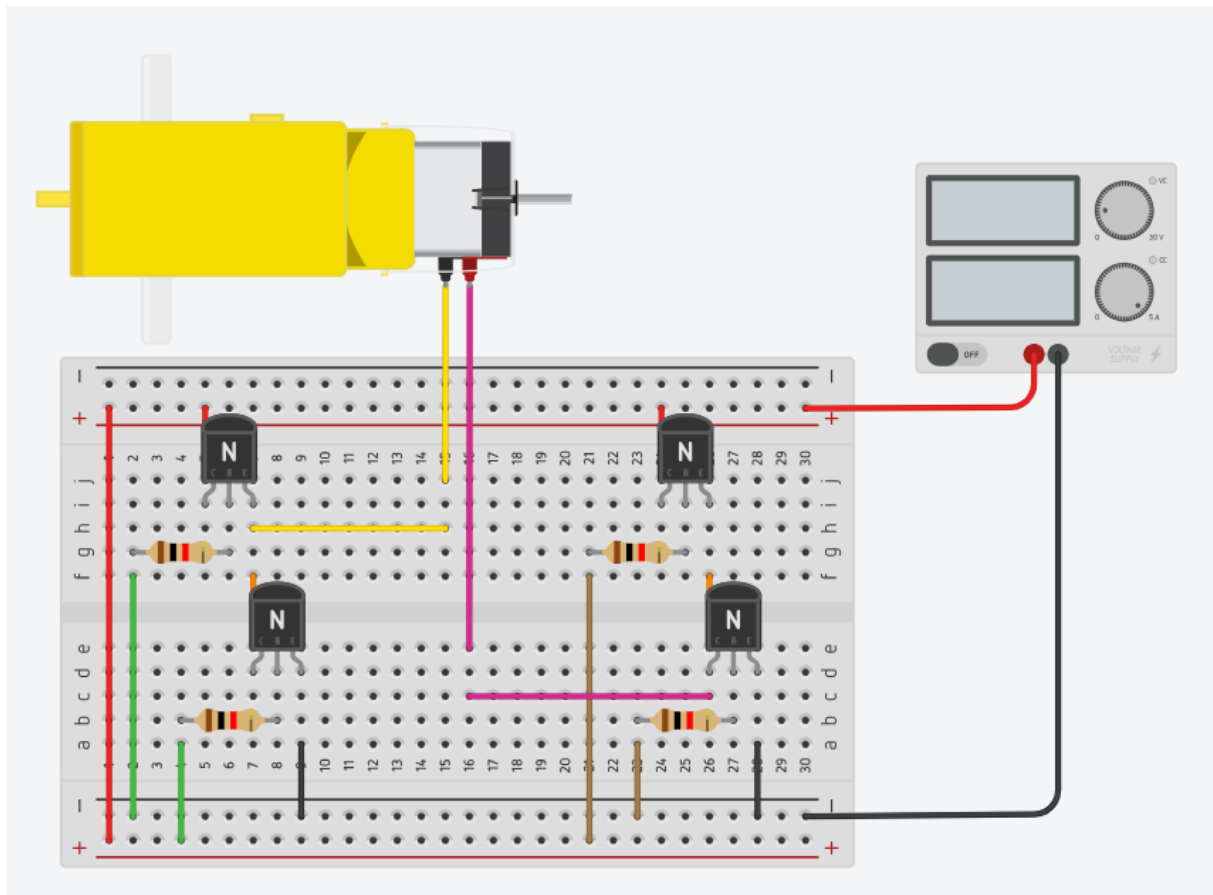
## 4. Analog Outputs (Part 2)

4. Analog Outputs (Part 2) .....	1
4.3. H-Bridge Construction with Transistors and Mosfets.....	2
4.3.1.a.H-Bridge Construction using NPN transistors .....	2
4.3.1.b.H-Bridge Construction using NPN and PNP transistors .....	3
4.3.2.a.H-Bridge Construction Using NMosfets .....	4
4.3.2.b.H-Bridge Construction Using NMOS and PMOS .....	5
4.3.3. H-Bridge Construction Using PNP and NMOS .....	6
4.4. L293 Driver .....	7
4.4.1.a.Observing L293 Driver Inputs and Outputs .....	7
4.4.1.b.DC Motor Control using L293 Driver.....	9
4.4.1.c.Servo Motor Control using L293 Driver .....	11
4.4.1.d.Step Motor Control using L293 Driver .....	12
4.4.2.a.DC Motor Control using Buttons and L293 Driver.....	13
4.4.2.b.DC Motor Control using Buttons and L293 Driver_opt .....	15
4.4.2.c.DC Motor Control with Safety Buttons and L293 Driver .....	17
4.4.3.a.Hobby Motor Control with Pot using L293 Driver .....	19
4.4.3.b.DC Motor Control using Pot and L293 Driver .....	20
4.4.4.a.1.Speed Control of Two Hobby DC Motors with L293 Driver .....	22
4.4.4.a.2.Speed Control of Two Hobby DC Motors with L293_opt.....	24
4.4.4.b.1.Control Two Hobby DC motors with L293 driver (pullup).....	26
4.4.4.b.2.Control Two Hobby DC motors with L293 (pullup_opt) .....	29
4.4.4.c.1.Control Two DC Motors with Two Buttons and L293 Driver .....	31
4.4.4.c.2.Control Two DC Motors with Two Buttons and L293 opt .....	33

## 4.3. H-Bridge Construction with Transistors and Mosfets

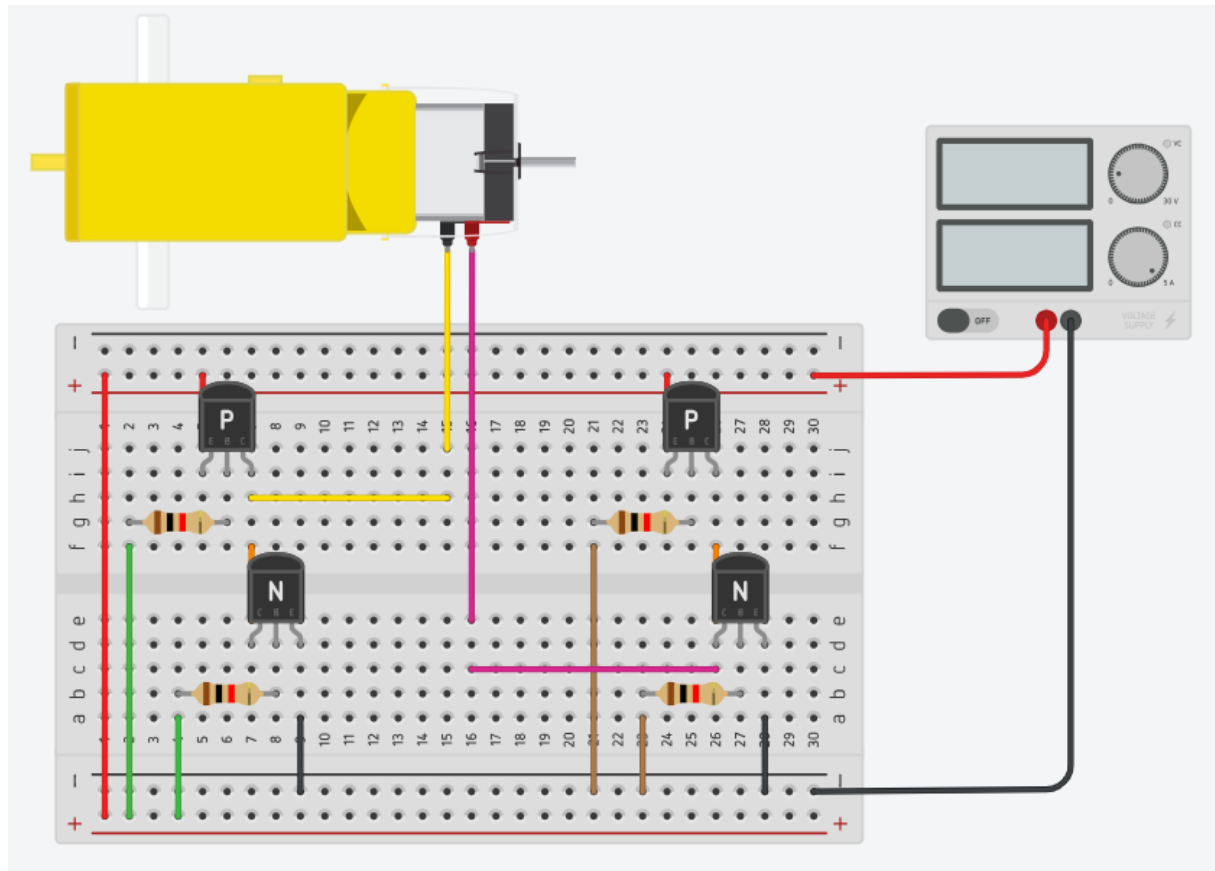
### 4.3.1.a.H-Bridge Construction using NPN transistors

<https://www.tinkercad.com/things/9QAU7bThOfk-431ah-bridge-construction-using-npn-transistors->



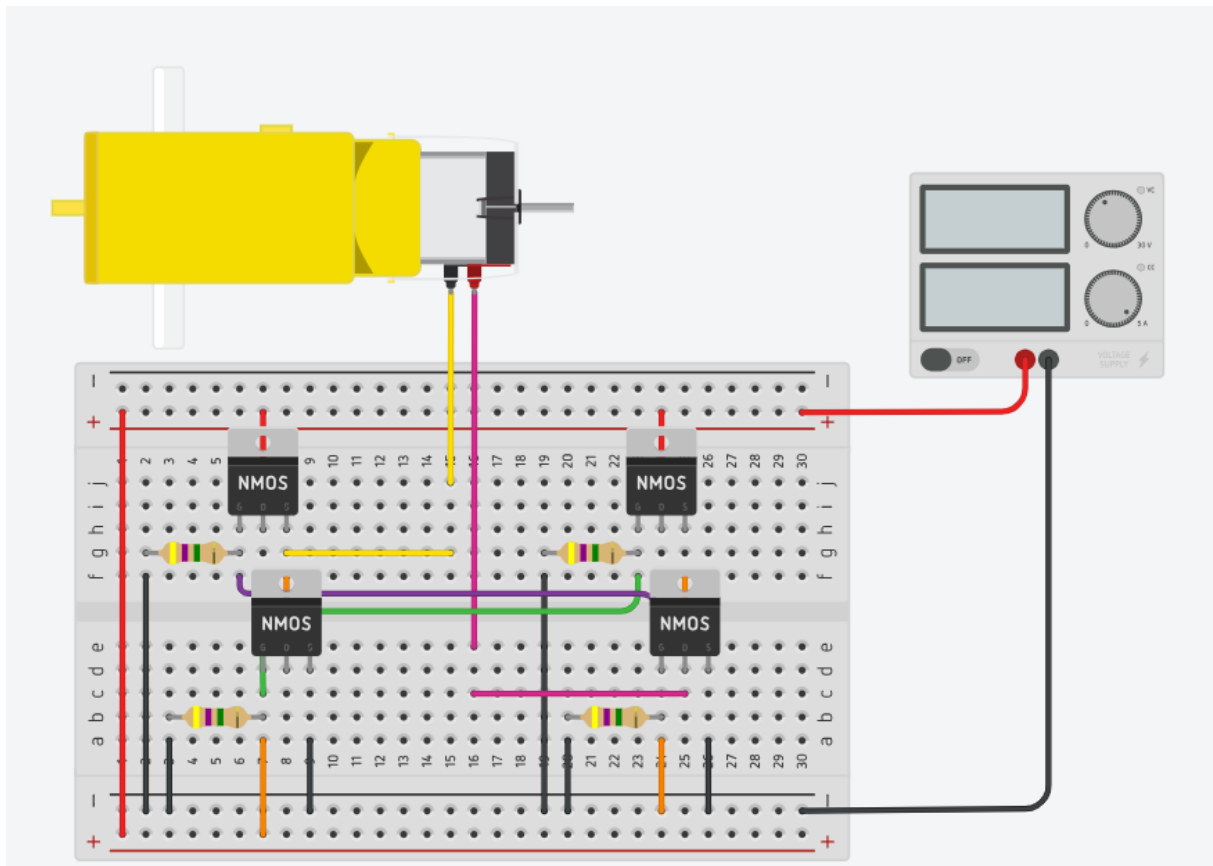
### 4.3.1.b.H-Bridge Construction using NPN and PNP transistors

<https://www.tinkercad.com/things/l7tyB7QHZZRH-431bh-bridge-construction-using-npn-and-pnp-transistors->



### 4.3.2.a.H-Bridge Construction Using NMosfets

<https://www.tinkercad.com/things/6gw5mh8z74B-432ah-bridge-construction-using-nmosfets>

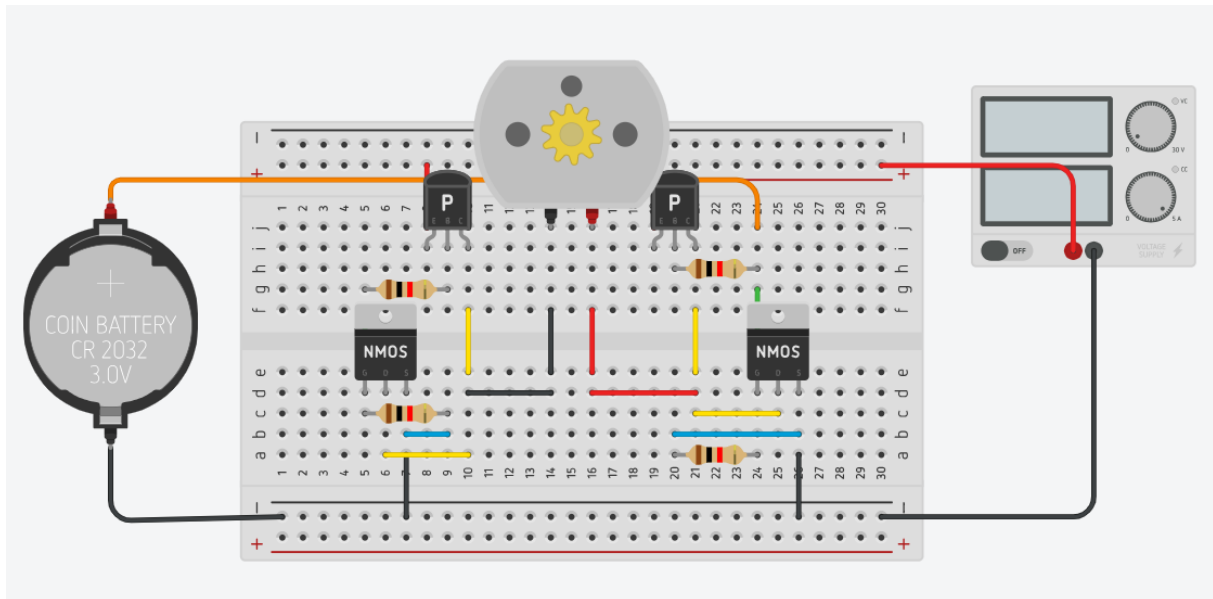


<https://www.tinkercad.com/things/6L8wnFabNKf-432bh-bridge-construction-using-nmos-and-pmos->



### 4.3.3. H-Bridge Construction Using PNP and NMOS

<https://www.tinkercad.com/things/12OnaFh5m4m-433-h-bridge-construction-using-pnp-and-nmos->



## 4.4. L293 Driver

### 4.4.1.a.Observing L293 Driver Inputs and Outputs

<https://www.tinkercad.com/things/3wL0JiXx0m0-441aobserving-l293-driver-inputs-and-outputs>

```

/*****
**

```

Program Name: Observing L293 Driver Inputs and Outputs  
on an Oscilloscope

Programın Objective:

This program is designed to create a PWM signal on pin 3 that can be used to control a motor through an L293 motor driver. The function is used to generate a PWM signal with a duty cycle corresponding to the analogWrite150 value (possible from 0 to 255). This setup is particularly useful for observing how the L293 driver operates in response to different PWM signals when analyzed with an oscilloscope. The short delay in the loop ensures a stable and consistent signal output.

Written by: Kamil Bala  
kamilbala42@gmail.com  
tw: @tek\_elo  
Yalova Deneyap Atelyesi / 2022

```

*****/

```

```

void setup()
{
  pinMode(3, OUTPUT); // Set pin 3 as an OUTPUT. This pin will be used to send signals to
the L293 driver.
}

```

```

void loop()
{
  analogWrite(3, 150); // Write a PWM signal with a value of 150 to pin 3.
                        // This value is approximately 59% of the maximum duty cycle (255),
                        // and it will be used to control the speed or direction of a motor via the L293
driver.

```

```

  delay(10); // Delay for 10 milliseconds to stabilize the loop execution.
              // This small delay can help in producing a more consistent signal,
              // especially when observing the waveform on an oscilloscope.
}

```





### 4.4.1.b.DC Motor Control using L293 Driver

<https://www.tinkercad.com/things/7C7FBG6O3IL-441bdc-motor-control-using-l293-driver>

```

/*****
**

```

Program Name: DC Motor Control using L293 Driver

Program's Purpose:

In this code, a DC motor is controlled using the L293 motor driver. enablePin is used to turn the motor on and off, in1A and in2A to determine the direction of the motor. The engine is first started in one direction for 5 seconds, then stopped for 1 second, started in the opposite direction for another 5 seconds, and finally stopped again for 1 second. This loop is repeated continuously loop(). Using the digitalWrite control pins on HIGH and on LOW changes the direction of the motor.

Yazan: Kamil Bala  
 kamilbala42@gmail.com  
 tw: @tek\_elo  
 Yalova / 2023

```

*****/
//

```

```

int enablePin = 11;
int in1A = 2;
int in2A = 3;

```

```

void setup()
{
  pinMode(enablePin, OUTPUT); // Set the enable pin of the L293 driver as OUTPUT
  pinMode(in1A, OUTPUT); // Set the control pin in1A as OUTPUT
  pinMode(in2A, OUTPUT); // Set the control pin in2A as OUTPUT
  digitalWrite(enablePin, LOW); // Initially disable the motor
}

```

```

void loop()
{
  digitalWrite(enablePin, HIGH); // Enable the motor by setting the enable pin HIGH
  digitalWrite(in1A, LOW); // Set in1A to LOW
  digitalWrite(in2A, HIGH); // Set in2A to HIGH - this combination makes the motor turn in
one direction
  delay(5000); // Continue turning in this direction for 5 seconds

  digitalWrite(enablePin, LOW); // Disable the motor

```

```
delay(1000); // Wait for 1 second

digitalWrite(enablePin, HIGH); // Enable the motor again
digitalWrite(in1A, HIGH); // Set in1A to HIGH
digitalWrite(in2A, LOW); // Set in2A to LOW - this combination makes the motor turn in
the opposite direction
delay(5000); // Continue turning in this direction for 5 seconds

digitalWrite(enablePin, LOW); // Disable the motor again
delay(1000); // Wait for 1 second before the next loop iteration
}
```

### 4.4.1.c.Servo Motor Control using L293 Driver

<https://www.tinkercad.com/things/gnKJdmtfaWH-441cservo-motor-control-using-l293-driver>

```

/*****
**

```

Program Name: Servo Motor Control using L293 Driver

Program's Purpose:

Observing the operation of the servo motor with the L293D driver

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

```

****/
//

```

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

```
Servo myservo; // Create a servo object to control a servo
                // Up to twelve servo objects can be created on most boards
```

```
int pos = 0; // Variable to store the servo position
```

```
void setup() {
  digitalWrite(10, HIGH); // This line seems to be setting pin 10 to HIGH, but without
  knowing the specific use case,
                        // it's unclear how this relates to the servo. It might be for enabling a driver or
  a different component.
  myservo.attach(9); // Attaches the servo on pin 9 to the servo object
}
```

```
void loop() {
  for (pos = 0; pos <= 180; pos += 3) { // Goes from 0 degrees to 180 degrees in steps of 3
  degrees
    myservo.write(pos); // Tell servo to go to the position specified in variable 'pos'
    delay(5);           // Waits 5ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 3) { // Goes from 180 degrees back to 0 degrees in steps of
  3 degrees
    myservo.write(pos); // Tell servo to go to the position specified in variable 'pos'
    delay(5);           // Waits 5ms for the servo to reach the position
  }
}
```

#### 4.4.1.d.Step Motor Control using L293 Driver

<https://www.tinkercad.com/things/2fTLMHbh7hr-441ddc-motor-control-using-l293-driver>

```

/*****
**

```

Program Name: DC Motor Control using L293 Driver

Program's Purpose:

In this program, a stepper motor is controlled using the L293 motor driver and library.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

```

*****/

```

```
#include <Stepper.h>
```

```
#define STEPS 200 // Define the number of steps per revolution for the motor
```

```
// Create a Stepper object named 'stepper', specifying the number of steps and control pins
Stepper stepper(STEPS, 9, 10, 11, 12);
```

```
void setup()
```

```
{
  stepper.setSpeed(100); // Set the speed of the stepper motor to 100 RPM (rotations per
minute)
}
```

```
void loop()
```

```
{
  stepper.step(-200); // Move the stepper motor backwards 200 steps
                      // The negative value indicates the reverse direction
  delay(500); // Wait for 500 milliseconds before the next iteration of the loop
}
```

### 4.4.2.a.DC Motor Control using Buttons and L293 Driver

<https://www.tinkercad.com/things/d3HmfDm5t0f-442adc-motor-control-using-buttons-and-l293-driver>

```
/******
```

```
**
```

Program Name:DC Motor Control using Buttons and L293 Driver

Program's Purpose:

This program controls a DC motor using two buttons and an L293 motor driver. When button 1 is pressed, the motor rotates in one direction, and a white LED turns on. When button 2 is pressed, the motor rotates in the opposite direction, and a yellow LED turns on. When neither button is pressed, the motor stops, and both LEDs are off. The program uses the digitalRead function to check the state of the buttons and digitalWrite to control the motor and LEDs. The Enable pin is set to HIGH in the setup() function to activate the motor driver.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

```
*****
```

```
***/
```

```
int buttonPin1 = 3; // Button for activation
int buttonPin2 = 4; // Another button for activation
int Motor_Pin1 = 5; // Pin 2 on the L293D
int Motor_Pin2 = 6; // Pin 7 on the L293D
int Enable = 7; // Pin 1 on the L293D
int led1 = 8; // White LED
int led2 = 9; // Yellow LED
```

```
void setup() {
  pinMode(buttonPin1, INPUT); // Set buttonPin1 as input
  pinMode(buttonPin2, INPUT); // Set buttonPin2 as input
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);

  // Set motor pins and ENABLE pin as outputs
  pinMode(Motor_Pin1, OUTPUT);
  pinMode(Motor_Pin2, OUTPUT);
  pinMode(Enable, OUTPUT);
  digitalWrite(Enable, HIGH); // Set ENABLE to HIGH to enable the motor
}
```

```
}

void loop() {
  if (digitalRead(buttonPin1) == HIGH) { // Check the state of button 1

    // Set the motor to rotate in one direction
    digitalWrite(Motor_Pin1, LOW);
    digitalWrite(Motor_Pin2, HIGH);
    digitalWrite(led1, HIGH); // Turn on white LED
  } else {
    // If the button is not pressed, stop the motor
    digitalWrite(Motor_Pin1, LOW);
    digitalWrite(Motor_Pin2, LOW);
    digitalWrite(led1, LOW); // Turn off white LED
  }

  // OTHER PART

  if (digitalRead(buttonPin2) == HIGH) { // Check the state of button 2

    // Set the motor to rotate in the opposite direction
    digitalWrite(Motor_Pin1, HIGH);
    digitalWrite(Motor_Pin2, LOW);
    digitalWrite(led2, HIGH); // Turn on yellow LED
  } else {
    // If the button is not pressed, stop the motor
    digitalWrite(Motor_Pin1, LOW);
    digitalWrite(Motor_Pin2, LOW);
    digitalWrite(led2, LOW); // Turn off yellow LED
  }
}
```

### 4.4.2.b.DC Motor Control using Buttons and L293 Driver\_opt

<https://www.tinkercad.com/things/3z5pobomYkN-442bdc-motor-control-using-buttons-and-l293-driveropt>

```
/******  
**
```

Program Name:DC Motor Control using Buttons and L293 Driver

Program's Purpose:

This program controls a DC motor using two buttons and an L293 motor driver. When button 1 is pressed, the motor rotates in one direction, and a white LED turns on. When button 2 is pressed, the motor rotates in the opposite direction, and a yellow LED turns on. When neither button is pressed, the motor stops, and both LEDs are off. The program uses the digitalRead function to check the state of the buttons and digitalWrite to control the motor and LEDs. The Enable pin is set to HIGH in the setup() function to activate the motor driver.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

```
*****  
***/
```

```
// DC Motor Control using Buttons and L293 Driver
```

```
int buttonPin1 = 3; // Button for activation  
int buttonPin2 = 4; // Another button for activation  
int Motor_Pin1 = 5; // Pin 2 on the L293D  
int Motor_Pin2 = 6; // Pin 7 on the L293D  
int Enable = 7;    // Pin 1 on the L293D  
int led1 = 8;      // White LED  
int led2 = 9;      // Yellow LED
```

```
void setup() {  
  pinMode(buttonPin1, INPUT); // Set buttonPin1 as input  
  pinMode(buttonPin2, INPUT); // Set buttonPin2 as input  
  pinMode(led1, OUTPUT);  
  pinMode(led2, OUTPUT);
```

```
// Set motor pins and ENABLE pin as outputs  
  pinMode(Motor_Pin1, OUTPUT);  
  pinMode(Motor_Pin2, OUTPUT);  
  pinMode(Enable, OUTPUT);
```

```
digitalWrite(Enable, HIGH); // Enable the motor
}

void loop() {
  controlMotor(buttonPin1, Motor_Pin1, Motor_Pin2, led1);
  controlMotor(buttonPin2, Motor_Pin2, Motor_Pin1, led2);
}

void controlMotor(int buttonPin, int motorPin1, int motorPin2, int ledPin) {
  if (digitalRead(buttonPin) == HIGH) {
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(ledPin, HIGH); // Turn on the LED
  } else {
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(ledPin, LOW); // Turn off the LED
  }
}
```



### 4.4.2.c.DC Motor Control with Safety Buttons and L293 Driver

<https://www.tinkercad.com/things/8USf5sudzKu-442cdc-motor-control-with-safety-buttons-and-l293-driver>

/\*\*\*\*\*\*

\*\*

Program Name:DC Motor Control using Buttons and L293 Driver

Program's Purpose:

Two safety buttons (for forward and reverse directions) are used in this program. Based on the forward direction button, the motor rotates in the forward direction and the white LED turns on. When the reverse direction button is used, the motor rotates in the opposite direction and the yellow LED lights up. If no button is pressed, the motor stops and the two LEDs go out. This structure ensures that the motor only works in a certain direction and moves in the corresponding direction based on the buttons.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

\*\*\*\*\*

\*\*\*/\*

// DC Motor Control with Safety Buttons and L293 Driver

int forwardButtonPin = 3; // Forward activation button

int backwardButtonPin = 4; // Backward activation button

int Motor\_Pin1 = 5; // Pin 2 on the L293D

int Motor\_Pin2 = 6; // Pin 7 on the L293D

int Enable = 7; // Pin 1 on the L293D

int led1 = 8; // White LED for forward direction

int led2 = 9; // Yellow LED for backward direction

void setup() {

pinMode(forwardButtonPin, INPUT); // Set forward button as input

pinMode(backwardButtonPin, INPUT); // Set backward button as input

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

// Set motor pins and ENABLE pin as outputs

pinMode(Motor\_Pin1, OUTPUT);

pinMode(Motor\_Pin2, OUTPUT);

pinMode(Enable, OUTPUT);

digitalWrite(Enable, HIGH); // Enable the motor

```
}  
  
void loop() {  
  if (digitalRead(forwardButtonPin) == HIGH) {  
    digitalWrite(Motor_Pin1, HIGH);  
    digitalWrite(Motor_Pin2, LOW);  
    digitalWrite(led1, HIGH); // Turn on white LED for forward  
    digitalWrite(led2, LOW); // Turn off yellow LED  
  } else if (digitalRead(backwardButtonPin) == HIGH) {  
    digitalWrite(Motor_Pin1, LOW);  
    digitalWrite(Motor_Pin2, HIGH);  
    digitalWrite(led1, LOW); // Turn off white LED  
    digitalWrite(led2, HIGH); // Turn on yellow LED for backward  
  } else {  
    // Stop the motor if neither button is pressed  
    digitalWrite(Motor_Pin1, LOW);  
    digitalWrite(Motor_Pin2, LOW);  
    digitalWrite(led1, LOW); // Turn off white LED  
    digitalWrite(led2, LOW); // Turn off yellow LED  
  }  
}
```

### 4.4.3.a.Hobby Motor Control with Pot using L293 Driver

<https://www.tinkercad.com/things/90sy8bul7k9-443ahobby-motor-control-with-pot-using-l293-driver>

/\*\*\*\*\*\*

\*\*

Program Name: Hobby Motor Control with Potentiometer using L293 Driver

Program's Purpose:

This program controls the speed of the hobby motor using a potentiometer and L293 motor driver. The value read from the potentiometer is used to adjust the speed of the motor. The speed value is sent as a PWM signal via pin 11, while pin 10 is used to adjust the direction of the motor. The function `analogRead` reads the analog value of the potentiometer, which is then mapped to a suitable range for PWM control. This `Serial.println` expression helps monitor the value sent to the engine for speed control.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

\*\*\*\*\*

\*\*\*/\*

`int a; // Variable to store the analog value read from the potentiometer`

```
void setup() {
  Serial.begin(9600); // Initialize serial communication at 9600 bits per second
  pinMode(11, OUTPUT); // Set pin 11 as OUTPUT, used for motor speed control
  pinMode(10, OUTPUT); // Set pin 10 as OUTPUT, used for motor direction control
  pinMode(A0, INPUT); // Set pin A0 as INPUT, for reading the potentiometer value
}

void loop() {
  a = analogRead(A0); // Read the value from the potentiometer
  a = map(a, 0, 1023, 0, 255); // Map the potentiometer value (0 to 1023) to a value usable for PWM (0 to 255)
  analogWrite(11, a); // Write the mapped value to pin 11 to control the motor speed
  digitalWrite(10, LOW); // Set pin 10 to LOW to set the direction of the motor
  delay(100); // Wait for 100 milliseconds
  Serial.println(a); // Print the value of 'a' to the Serial Monitor for debugging
}
```

### 4.4.3.b.DC Motor Control using Pot and L293 Driver

<https://www.tinkercad.com/things/8J4FO0r0MwB-443bdc-motor-control-using-pot-and-l293-driver>

/\*\*\*\*\*\*

\*\*

Program Name: DC Motor Control using Potentiometer and L293 Driver

Program's Purpose:

This program controls the speed of a DC motor using a potentiometer and an L293 motor driver. The potentiometer's analog value is read from analogPin and then converted to a range suitable for PWM control (0-255). This value is used to control the speed of the motor connected to motorPin. The motor's rotation direction is set using the L293\_1A and L293\_2A pins, ensuring that the motor only rotates in one direction. The analogWrite function is used to send a PWM signal to the motor for speed control.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

\*\*\*\*\*

\*\*\*/\*

// DC Motor Control using Potentiometer and L293 Driver

int motorPin = 11; // Pin for motor speed control

int analogPin = A0; // Input pin for the potentiometer

int val = 0; // Variable to store the value read from the potentiometer

int L293\_1A = 13; // Control pin of the L293 motor driver for one direction

int L293\_2A = 12; // Control pin of the L293 motor driver for the other direction

void setup() {

pinMode(motorPin, OUTPUT); // Set motorPin as OUTPUT for motor speed control

pinMode(analogPin, INPUT); // Set analogPin as INPUT to read potentiometer value

pinMode(L293\_1A, OUTPUT); // Set L293\_1A as OUTPUT for one direction of motor control

pinMode(L293\_2A, OUTPUT); // Set L293\_2A as OUTPUT for the other direction of motor control

// Set the motor to rotate only in one direction

digitalWrite(L293\_1A, HIGH); // Enable one direction

digitalWrite(L293\_2A, LOW); // Disable the other direction

}

```
void loop() {  
    // Read the value from the potentiometer and set the motor speed  
    val = analogRead(analogPin); // Read the value between 0 - 1023 from the potentiometer  
    analogWrite(motorPin, val / 4); // Convert the value to 0 - 255 range and set the motor  
    speed  
    delay(100);  
}
```

#### 4.4.4.a.1.Speed Control of Two Hobby DC Motors with L293 Driver

<https://www.tinkercad.com/things/7gVAYt9FrIW-444a1speed-control-of-two-hobby-dc-motors-with-l293-driver>

/\*\*\*\*\*\*

\*\*

Program Name: L293 sürücü ile İki Hobi  
DC motor hız kontrolü

Program's Purpose:

This program is designed to control the speed of two hobby DC motors using an L293 driver. It includes a pause function that stops the motors for 3 seconds. The loop function sets different combinations of HIGH and LOW signals to the motor control pins, effectively changing the direction and operation of each motor in a cyclic manner. The delay(3000) in the loop ensures that each motor configuration is maintained for 3 seconds before switching to the next configuration or pausing.

Written by: Kamil Bala  
kamilbala42@gmail.com  
tw: @tek\_elo  
Yalova / 2023

\*\*\*\*\*

\*\*\*/\*

// Speed Control of Two Hobby DC Motors with L293 Driver

```
void pause() {
  // Set all motor control pins to LOW to stop the motors
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  delay(3000); // Wait for 3000 milliseconds (3 seconds)
}
```

```
void setup() {
  // Initialize motor control pins as OUTPUT
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}
```

```
void loop() {
  // First motor configuration
```

```
digitalWrite(2, HIGH); // Enable one direction for motor 1
digitalWrite(3, LOW); // Disable the other direction for motor 1
digitalWrite(4, LOW); // Disable one direction for motor 2
digitalWrite(5, HIGH); // Enable the other direction for motor 2
delay(3000); // Wait for 3000 milliseconds (3 seconds)
```

```
pause(); // Pause both motors
```

```
// Second motor configuration
```

```
digitalWrite(2, LOW); // Disable one direction for motor 1
digitalWrite(3, HIGH); // Enable the other direction for motor 1
digitalWrite(4, HIGH); // Enable one direction for motor 2
digitalWrite(5, LOW); // Disable the other direction for motor 2
delay(3000); // Wait for 3000 milliseconds (3 seconds)
```

```
pause(); // Pause both motors
```

```
// Third motor configuration
```

```
digitalWrite(2, LOW); // Disable one direction for motor 1
digitalWrite(3, HIGH); // Enable the other direction for motor 1
digitalWrite(4, LOW); // Disable one direction for motor 2
digitalWrite(5, HIGH); // Enable the other direction for motor 2
delay(3000); // Wait for 3000 milliseconds (3 seconds)
```

```
pause(); // Pause both motors
```

```
// Fourth motor configuration
```

```
digitalWrite(2, HIGH); // Enable one direction for motor 1
digitalWrite(3, LOW); // Disable the other direction for motor 1
digitalWrite(4, HIGH); // Enable one direction for motor 2
digitalWrite(5, LOW); // Disable the other direction for motor 2
delay(3000); // Wait for 3000 milliseconds (3 seconds)
```

```
pause(); // Pause both motors
```

```
}
```

#### 4.4.4.a.2.Speed Control of Two Hobby DC Motors with L293\_opt

<https://www.tinkercad.com/things/5aYR2MYNbfD-444a2speed-control-of-two-hobby-dc-motors-with-l293opt>

/\*\*\*\*\*\*

\*\*

Program Name: L293 sürücü ile İki Hobi  
DC motor hız kontrolü

Program's Purpose:

This program is designed to control the speed of two hobby DC motors using an L293 driver. It includes a pause function that stops the motors for 3 seconds. The loop function sets different combinations of HIGH and LOW signals to the motor control pins, effectively changing the direction and operation of each motor in a cyclic manner. The delay(3000) in the loop ensures that each motor configuration is maintained for 3 seconds before switching to the next configuration or pausing.

Written by: Kamil Bala  
kamilbala42@gmail.com  
tw: @tek\_elo  
Yalova / 2023

\*\*\*\*\*

\*\*\*/\*

// Speed Control of Two Hobby DC Motors with L293 Driver

```
void motorControl(int pin1, int pin2, int pin3, int pin4) {
  digitalWrite(pin1, HIGH);
  digitalWrite(pin2, LOW);
  digitalWrite(pin3, HIGH);
  digitalWrite(pin4, LOW);
}
```

```
void pause() {
  // Set all motor control pins to LOW to stop the motors
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  delay(3000); // Wait for 3000 milliseconds (3 seconds)
}
```

```
void setup() {
  // Initialize motor control pins as OUTPUT
  pinMode(2, OUTPUT);
```



```
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
}

void loop() {
  // Cycle through different motor configurations
  motorControl(2, 3, 4, 5);
  delay(3000); // Wait for 3000 milliseconds (3 seconds)
  pause();

  motorControl(3, 2, 5, 4);
  delay(3000); // Wait for 3000 milliseconds (3 seconds)
  pause();

  motorControl(3, 2, 4, 5);
  delay(3000); // Wait for 3000 milliseconds (3 seconds)
  pause();

  motorControl(2, 3, 5, 4);
  delay(3000); // Wait for 3000 milliseconds (3 seconds)
  pause();
}
```

#### 4.4.4.b.1. Control Two Hobby DC motors with L293 driver (pullup)

<https://www.tinkercad.com/things/84wboRuwg3F-444b1control-two-hobby-dc-motors-with-l293-driver-pullup>

/\*\*\*\*\*\*

\*\*

Program Name: Control Two Hobby DC motors with L293 driver

Program's Purpose:

This code controls a four-wheeled robot (or similar device) using four direction buttons: forward, backward, left, and right. The pins in1 to in4 control the motor, and each direction has its own function (forward, back, left, right). The buttons are connected to pins f, b, l, r and are set up with internal pull-up resistors. When a button is pressed, the corresponding function is called to move the robot in that direction. If no button is pressed, all motors are stopped. The delay(500) statements are used to keep the robot moving in a direction for a short time before checking the button states again.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

\*\*\*\*\*

\*\*\* /

```
int in1 = 4;
int in2 = 5;
int in3 = 6;
int in4 = 7;
int f = 11; // Forward button pin
int b = 12; // Backward button pin
int l = 10; // Left button pin
int r = 9; // Right button pin
```

```
void setup() {
  // Set motor control pins as OUTPUT
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Set button pins as INPUT with internal pull-up resistor
  pinMode(f, INPUT_PULLUP);
  pinMode(b, INPUT_PULLUP);
  pinMode(l, INPUT_PULLUP);
  pinMode(r, INPUT_PULLUP);
}
```

```
// Initialize motor control pins to LOW (motor off)
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}

// Function to move forward
void forward() {
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

// Function to move backward
void back() {
  digitalWrite(in2, HIGH);
  digitalWrite(in1, LOW);
  digitalWrite(in4, HIGH);
  digitalWrite(in3, LOW);
}

// Function to move left
void left() {
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in4, HIGH);
  digitalWrite(in3, LOW);
}

// Function to move right
void right() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void loop() {
  // Check button states and move accordingly
  if (digitalRead(f) == LOW) {
    forward();
    delay(500);
  } else if (digitalRead(b) == LOW) {
    back();
    delay(500);
  } else if (digitalRead(l) == LOW) {
    left();
    delay(500);
  }
}
```

```
} else if (digitalRead(r) == LOW) {  
  right();  
  delay(500);  
} else {  
  // Stop all motor activity if no button is pressed  
  digitalWrite(in1, LOW);  
  digitalWrite(in2, LOW);  
  digitalWrite(in3, LOW);  
  digitalWrite(in4, LOW);  
  delay(500);  
}  
}
```

#### 4.4.4.b.2. Control Two Hobby DC motors with L293 (pullup\_opt)

<https://www.tinkercad.com/things/3XMCQsV3Qb0-444b2control-two-hobby-dc-motors-with-l293-pullupopt>

```

/*****
**

```

Program Name: Control Two Hobby DC motors with L293 driver  
(pullup-opt)

Program's Purpose:

This code controls a motor using four buttons connected to an Arduino, each corresponding to a different direction (forward, backward, left, right). The move function sets the state of the motor control pins to either HIGH or LOW, depending on the button pressed. This function simplifies the motor control process and increases the readability of the code. The use of INPUT\_PULLUP for the button pins enables the internal pull-up resistors, eliminating the need for external resistors.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

```

*****/

```

```

int in1 = 4; // Motor control pin
int in2 = 5; // Motor control pin
int in3 = 6; // Motor control pin
int in4 = 7; // Motor control pin
int forwardButton = 11; // Button for moving forward
int backButton = 12; // Button for moving backward
int leftButton = 10; // Button for moving left
int rightButton = 9; // Button for moving right

```

```

void setup() {
  // Initialize motor control pins as OUTPUT
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Initialize button pins as INPUT with internal pull-up resistor
  pinMode(forwardButton, INPUT_PULLUP);
  pinMode(backButton, INPUT_PULLUP);
  pinMode(leftButton, INPUT_PULLUP);
  pinMode(rightButton, INPUT_PULLUP);
}

```

```
// Function to control the motor movement
void move(int pin1State, int pin2State, int pin3State, int pin4State) {
  // Set each motor control pin to the specified state
  digitalWrite(in1, pin1State);
  digitalWrite(in2, pin2State);
  digitalWrite(in3, pin3State);
  digitalWrite(in4, pin4State);
}

void loop() {
  // Check the state of each button and move accordingly
  if (digitalRead(forwardButton) == LOW) {
    move(HIGH, LOW, HIGH, LOW); // Move forward
  } else if (digitalRead(backButton) == LOW) {
    move(LOW, HIGH, LOW, HIGH); // Move backward
  } else if (digitalRead(leftButton) == LOW) {
    move(HIGH, LOW, LOW, HIGH); // Move left
  } else if (digitalRead(rightButton) == LOW) {
    move(LOW, HIGH, HIGH, LOW); // Move right
  } else {
    move(LOW, LOW, LOW, LOW); // Stop if no button is pressed
  }
  delay(500); // Delay for a short period
}
```

### 4.4.4.c.1. Control Two DC Motors with Two Buttons and L293 Driver

<https://www.tinkercad.com/things/eNAweSLmbQo-444c1control-two-dc-motors-with-two-buttons-and-l293-driver>

/\*\*\*\*\*\*

\*\*

program name: Control Two DC Motors with Two Buttons and L293 Driver

#### Program's Purpose

This program controls two DC motors using an L293 motor driver and five buttons for different directions (forward, backward, right, left, and stop). The rightM1, leftM1, rightM2, and leftM2 pins are used to control the motors. When a button is pressed, the corresponding motor control pins are set to HIGH or LOW to drive the motors in the desired direction. The buttonStop is used to stop both motors by setting all control pins to HIGH. This setup allows for basic directional control of a vehicle or robot.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

\*\*\*\*\*

\*\*\*/\*

//

// L293D input pins, 12 and 13 for Motor1; 6 and 7 for Motor2.

int rightM1 = 12;

int leftM1 = 13;

int rightM2 = 6;

int leftM2 = 7;

// Pins for the buttons.

int buttonRight = 9;

int buttonLeft = 10;

int buttonUp = 8;

int buttonDown = 4;

int buttonStop = 3;

void setup() {

pinMode(rightM1, OUTPUT);

pinMode(rightM2, OUTPUT);

pinMode(leftM1, OUTPUT);

pinMode(leftM2, OUTPUT);

pinMode(buttonRight, INPUT);

pinMode(buttonLeft, INPUT);

pinMode(buttonUp, INPUT);

```
pinMode(buttonDown, INPUT);
pinMode(buttonStop, INPUT);

// Initialize motor control pins to LOW (motors off)
digitalWrite(leftM1, LOW);
digitalWrite(rightM1, LOW);
digitalWrite(rightM2, LOW);
digitalWrite(leftM2, LOW);
}

void loop() {
  if (digitalRead(buttonUp) == LOW) {
    // Move both motors forward
    digitalWrite(leftM1, HIGH);
    digitalWrite(rightM1, LOW);
    digitalWrite(rightM2, HIGH);
    digitalWrite(leftM2, LOW);
  }
  if (digitalRead(buttonDown) == LOW) {
    // Move both motors backward
    digitalWrite(leftM1, LOW);
    digitalWrite(rightM1, HIGH);
    digitalWrite(rightM2, LOW);
    digitalWrite(leftM2, HIGH);
  }
  if (digitalRead(buttonRight) == LOW) {
    // Turn right
    digitalWrite(leftM1, LOW);
    digitalWrite(rightM1, HIGH);
    digitalWrite(rightM2, HIGH);
    digitalWrite(leftM2, LOW);
  }
  if (digitalRead(buttonLeft) == LOW) {
    // Turn left
    digitalWrite(leftM1, HIGH);
    digitalWrite(rightM1, LOW);
    digitalWrite(rightM2, LOW);
    digitalWrite(leftM2, HIGH);
  }
  if (digitalRead(buttonStop) == LOW) {
    // Stop both motors
    digitalWrite(leftM1, HIGH);
    digitalWrite(rightM1, HIGH);
    digitalWrite(rightM2, HIGH);
    digitalWrite(leftM2, HIGH);
  }
}
```



#### 4.4.4.c.2. Control Two DC Motors with Two Buttons and L293 opt

<https://www.tinkercad.com/things/3TjR9yLiQ06-444c2control-two-dc-motors-with-two-buttons-and-l293-opt>

```

/*****
**

```

program name: Control Two DC Motors with Two Buttons and L293 Driver

##### Program's Purpose

This program controls two DC motors using an L293 motor driver and five buttons for different directions (forward, backward, right, left, and stop). The rightM1, leftM1, rightM2, and leftM2 pins are used to control the motors. When a button is pressed, the corresponding motor control pins are set to HIGH or LOW to drive the motors in the desired direction. The buttonStop is used to stop both motors by setting all control pins to HIGH. This setup allows for basic directional control of a vehicle or robot.

Written by: Kamil Bala

kamilbala42@gmail.com

tw: @tek\_elo

Yalova / 2023

```

*****/
//

```

// Control Two DC Motors with Two Pulldown Buttons and L293 Driver

```

int rightM1 = 12;
int leftM1 = 13;
int rightM2 = 6;
int leftM2 = 7;

```

```

int buttonRight = 9;
int buttonLeft = 10;
int buttonUp = 8;
int buttonDown = 4;
int buttonStop = 3;

```

```

void setup() {
  pinMode(rightM1, OUTPUT);
  pinMode(rightM2, OUTPUT);
  pinMode(leftM1, OUTPUT);
  pinMode(leftM2, OUTPUT);

  pinMode(buttonRight, INPUT);
  pinMode(buttonLeft, INPUT);

```

```
pinMode(buttonUp, INPUT);
pinMode(buttonDown, INPUT);
pinMode(buttonStop, INPUT);

stopMotors(); // Initially stop the motors
}

void loop() {
  if (digitalRead(buttonUp) == LOW) {
    moveMotors(HIGH, LOW, HIGH, LOW); // Move forward
  } else if (digitalRead(buttonDown) == LOW) {
    moveMotors(LOW, HIGH, LOW, HIGH); // Move backward
  } else if (digitalRead(buttonRight) == LOW) {
    moveMotors(LOW, HIGH, HIGH, LOW); // Turn right
  } else if (digitalRead(buttonLeft) == LOW) {
    moveMotors(HIGH, LOW, LOW, HIGH); // Turn left
  } else if (digitalRead(buttonStop) == LOW) {
    stopMotors(); // Stop
  }
}

void moveMotors(int stateM1Left, int stateM1Right, int stateM2Right, int stateM2Left) {
  digitalWrite(leftM1, stateM1Left);
  digitalWrite(rightM1, stateM1Right);
  digitalWrite(rightM2, stateM2Right);
  digitalWrite(leftM2, stateM2Left);
}

void stopMotors() {
  digitalWrite(leftM1, HIGH);
  digitalWrite(rightM1, HIGH);
  digitalWrite(rightM2, HIGH);
  digitalWrite(leftM2, HIGH);
}
```