

Below is a **detailed beginner guide to build your first obstacle-avoiding robot** using the parts you listed. I'll explain the **assembly, wiring diagram, and programming logic** step-by-step so you can actually build it.

1. Components Needed

Use these main parts:

- Microcontroller: Arduino Uno
- Motor controller: L298N Motor Driver Module
- Distance sensor: HC-SR04 Ultrasonic Sensor
- Motors: N20 DC Gear Motor (2 pieces)
- Robot base: 2WD Smart Robot Car Chassis Kit
- Power supply: 18650 Lithium-Ion Battery
- Jumper wires
- Breadboard

Optional but useful:

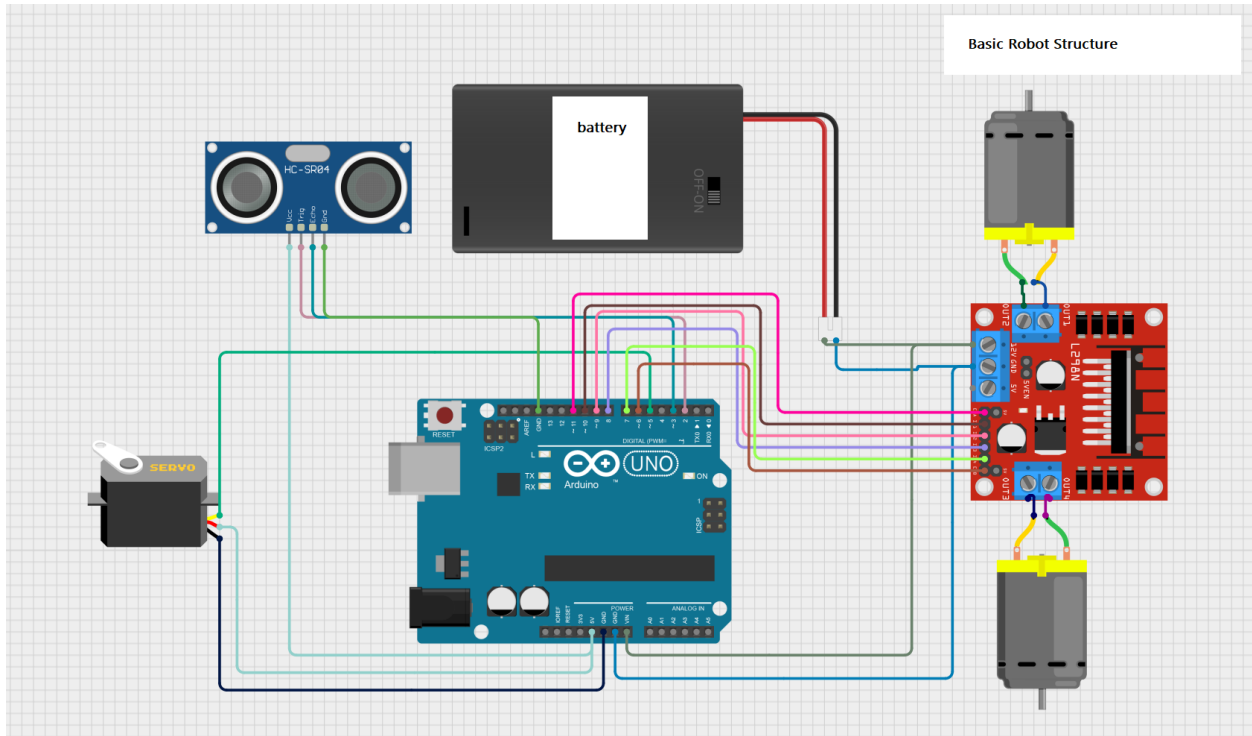
- On/off switch
 - Battery holder
-

2. Basic Robot Structure

Your robot will look like this conceptually:

```
[ Ultrasonic Sensor ]
  ||
-----
|   Arduino   |
-----
  ||||
-----
|   Motor Driver   |
-----
  |               |
Left Motor      Right Motor
  |               |
Wheel           Wheel

[ Battery Underneath ]
```

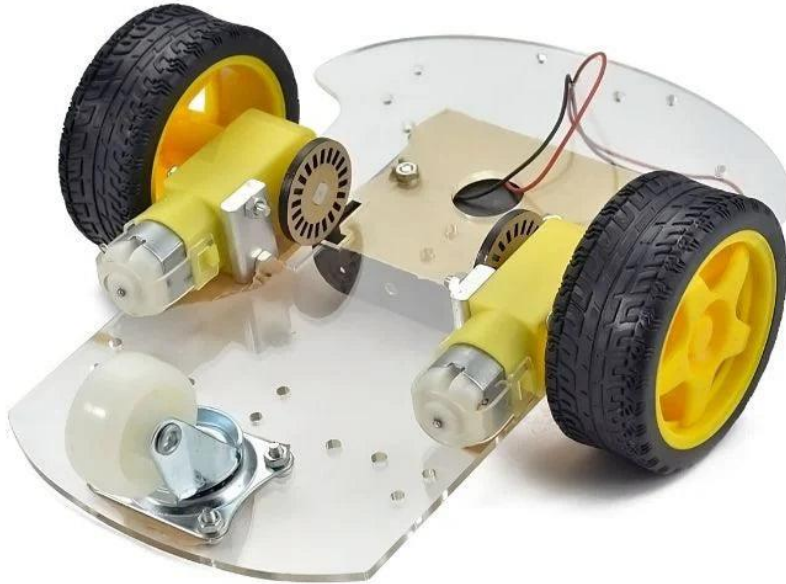


The sensor detects obstacles, the Arduino processes data, and the motor driver moves the wheels.

1. Assemble the Chassis

Steps:

1. Take the **2WD Smart Robot Car Chassis Kit** base plate.



2. Attach the **2 gear motors** to the motor holders.
3. Fix the **wheels** onto the motor shafts.
4. Attach the **battery holder** under or on the chassis.
5. Mount the **Arduino Uno** using screws.
6. Fix the **ultrasonic sensor** at the front of the robot.

Robot structure idea:

Front

↑
[Ultrasonic Sensor]

[Arduino Board]

[Motor Driver]

Left Wheel Right Wheel

Battery

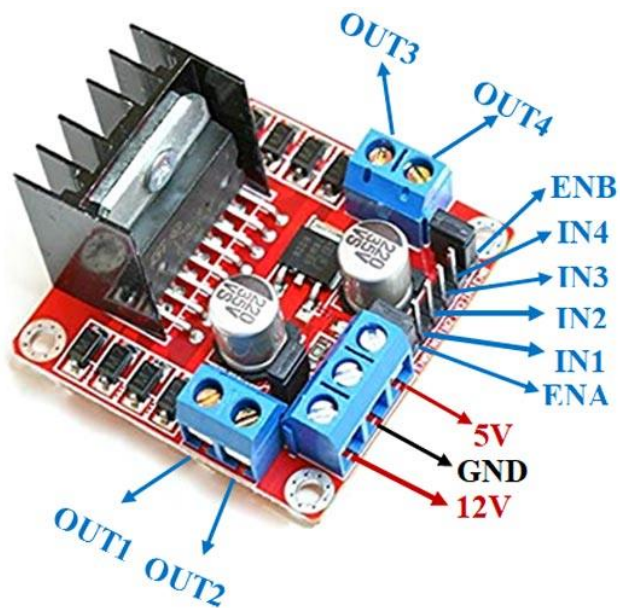
4. Wiring Diagram

Motor Driver to Motors

L298N Motor Driver

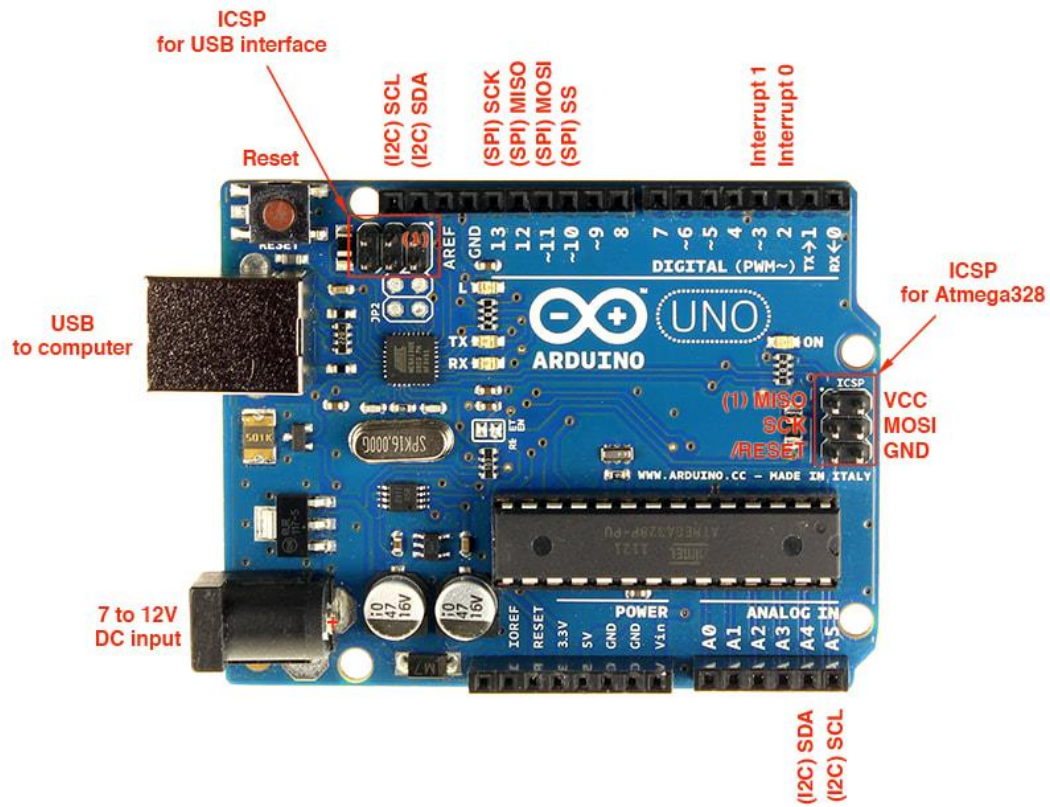
OUT1 ---- Left Motor +
OUT2 ---- Left Motor -

OUT3 ---- Right Motor +
OUT4 ---- Right Motor -



Motor Driver to Arduino

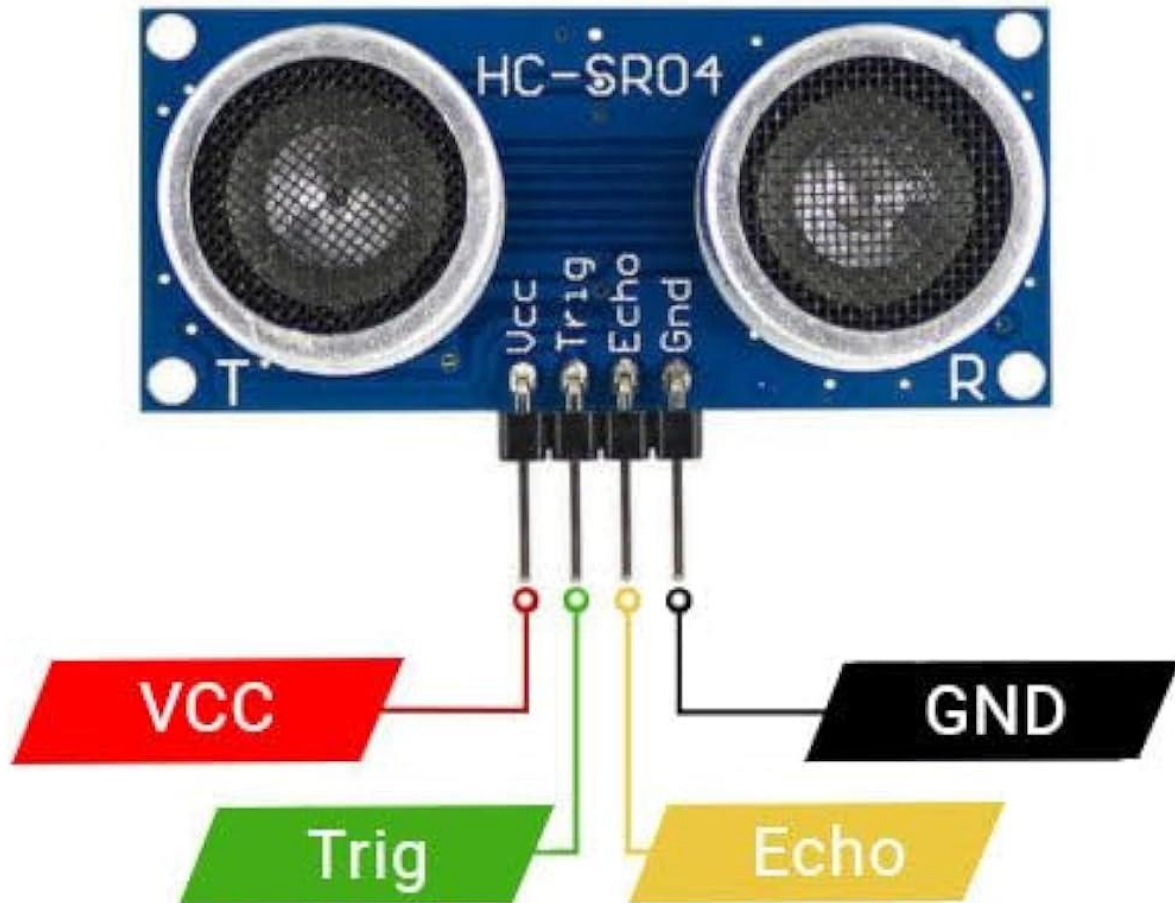
Arduino	L298N
-----	-----
Pin 8	-----> IN1
Pin 9	-----> IN2
Pin 10	-----> IN3
Pin 11	-----> IN4
5V	-----> 5V
GND	-----> GND



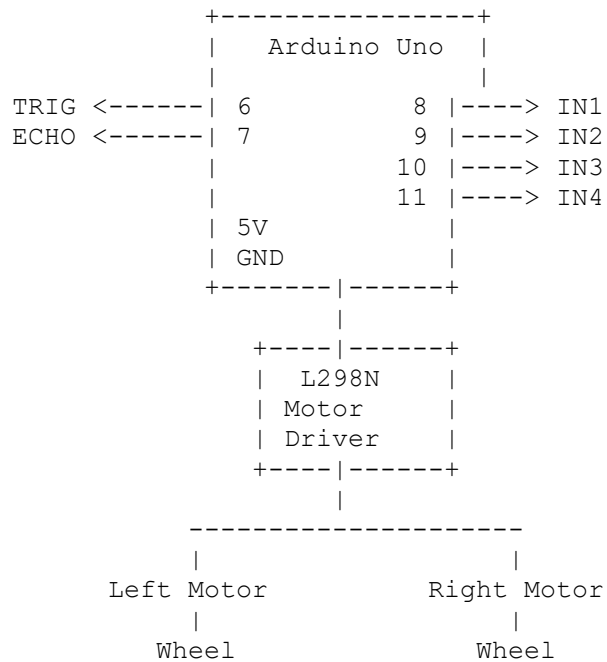
Ultrasonic Sensor Wiring

HC-SR04 Sensor Arduino

VCC -----> 5V
GND -----> GND
TRIG -----> Pin 6
ECHO -----> Pin 7



5. Complete Connection Diagram



6. Install Programming Software

Download:

- Arduino IDE

Steps:

1. Install the software.
 2. Connect Arduino using USB.
 3. Select **Tools** → **Board** → **Arduino Uno**.
 4. Select the correct **COM Port**.
-

7. Simple Robot Logic

Your robot will follow this logic:

Start

Measure distance

IF distance > 20 cm
 Move forward

IF distance < 20 cm
 Stop
 Turn right

Repeat

8. Example Arduino Code (Simple)

```
#define trigPin 6
#define echoPin 7

#define IN1 8
#define IN2 9
#define IN3 10
#define IN4 11

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  Serial.begin(9600);
}

void loop()
{
  long duration;
  int distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
```

```
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;

if(distance > 20)
{
forward();
}
else
{
stopRobot();
delay(500);
turnRight();
delay(500);
}
}

void forward()
{
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,LOW);
}

void stopRobot()
{
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,LOW);
}

void turnRight()
{
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);
}
}
```

Upload the code to the Arduino.

9. Power the Robot

1. Insert **18650 Lithium-Ion Battery** into the battery holder.
2. Turn on the power switch.
3. Place the robot on the floor.

The robot will:

- Move forward
 - Detect objects
 - Turn when obstacle appears
-

10. Ideas to Improve the Robot

Add features like:

- Bluetooth control using HC-05 Bluetooth Module
 - Camera vision using Raspberry Pi Camera Module
 - Voice control with Python
 - Line-following sensors
-

□ After this project you will understand:

- Sensors
- Motor control
- Arduino programming
- Robot design basics

Let's break your code **line by line** and explain how it works with the **HC-SR04 sensor** and the **robot motor control** pins. I'll connect it directly to what we discussed about **TRIG** and **ECHO**.

1. Pin Definitions

```
#define trigPin 6
#define echoPin 7
#define IN1 8
#define IN2 9
#define IN3 10
#define IN4 11
```

- `trigPin` → Arduino pin 6 → HC-SR04 **TRIG** pin (sends ultrasonic pulse)
 - `echoPin` → Arduino pin 7 → HC-SR04 **ECHO** pin (receives reflected pulse)
 - `IN1, IN2, IN3, IN4` → Pins 8–11 → Control **two DC motors** of the robot via an H-bridge or motor driver:
 - `IN1/IN2` → Motor 1
 - `IN3/IN4` → Motor 2
-

2. Setup Function

```
void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  Serial.begin(9600);
}
```

- **TRIG** pin set as **OUTPUT** → Arduino will send trigger pulses.
 - **ECHO** pin set as **INPUT** → Arduino reads the pulse duration.
 - Motor pins set as **OUTPUT** → Control direction and movement of motors.
 - Serial communication initialized at 9600 bps → Useful to print distance for debugging.
-

3. Loop Function – Distance Measurement

```
long duration;
int distance;

// Send pulse to trigger ultrasonic sensor
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Read echo pulse duration
duration = pulseIn(echoPin, HIGH);

// Convert duration to distance in cm
distance = duration * 0.034 / 2;
```

Step by step:

1. `digitalWrite(trigPin, LOW); delayMicroseconds(2);`
 - o Ensures a **clean LOW pulse** before triggering.
2. `digitalWrite(trigPin, HIGH); delayMicroseconds(10);`
 - o Sends **10 µs HIGH pulse** → HC-SR04 emits **ultrasonic sound**.
3. `digitalWrite(trigPin, LOW);`
 - o Ends the trigger pulse.
4. `duration = pulseIn(echoPin, HIGH);`
 - o Waits for ECHO pin to go **HIGH** and measures the **time it stays HIGH** → time of flight.
5. `distance = duration * 0.034 / 2;`
 - o Converts **time to distance** in cm.
 - o 0.034 cm/µs is approximate speed of sound in air.
 - o Divide by 2 for **round-trip**.

4. Robot Behavior Based on Distance

```
if(distance > 20)
{
  forward();
}
else
{
  stopRobot();
  delay(500);
  turnRight();
  delay(500);
}
```

- **If distance > 20 cm:** No obstacle → Robot moves forward.
 - **If distance ≤ 20 cm:** Obstacle detected → Robot stops, then turns right for 500 ms.
-

5. Motor Control Functions

Move Forward

```
void forward()  
{  
    digitalWrite(IN1,HIGH);  
    digitalWrite(IN2,LOW);  
    digitalWrite(IN3,HIGH);  
    digitalWrite(IN4,LOW);  
}
```

- Both motors spin forward:
 - Motor 1 → IN1 HIGH, IN2 LOW
 - Motor 2 → IN3 HIGH, IN4 LOW
 - Robot moves straight ahead.
-

Stop Robot

```
void stopRobot()  
{  
    digitalWrite(IN1,LOW);  
    digitalWrite(IN2,LOW);  
    digitalWrite(IN3,LOW);  
    digitalWrite(IN4,LOW);  
}
```

- All motor pins LOW → both motors stop → robot halts.
-

Turn Right

```
void turnRight()  
{  
    digitalWrite(IN1,HIGH);  
    digitalWrite(IN2,LOW);  
    digitalWrite(IN3,LOW);  
    digitalWrite(IN4,HIGH);  
}
```

- Motor 1 spins forward, Motor 2 spins backward → robot rotates to the right.
-

6. How TRIG & ECHO Connect to Robot Motion

- **TRIG (Pin 6)** → Tells sensor to send a pulse.
- **ECHO (Pin 7)** → Measures how far the obstacle is.
- **Logic:**

Distance

Action

> 20 cm Move forward (`forward()` function)

≤ 20 cm Stop → turn right (`stopRobot() + turnRight()`)

So your **ultrasonic sensor acts as the “eyes”** and the Arduino uses the data to **control the motors**.

□ Analogy:

- TRIG = “Look ahead”
 - ECHO = “How far is the obstacle?”
 - IN1–IN4 = “Legs/engines move robot accordingly”
-

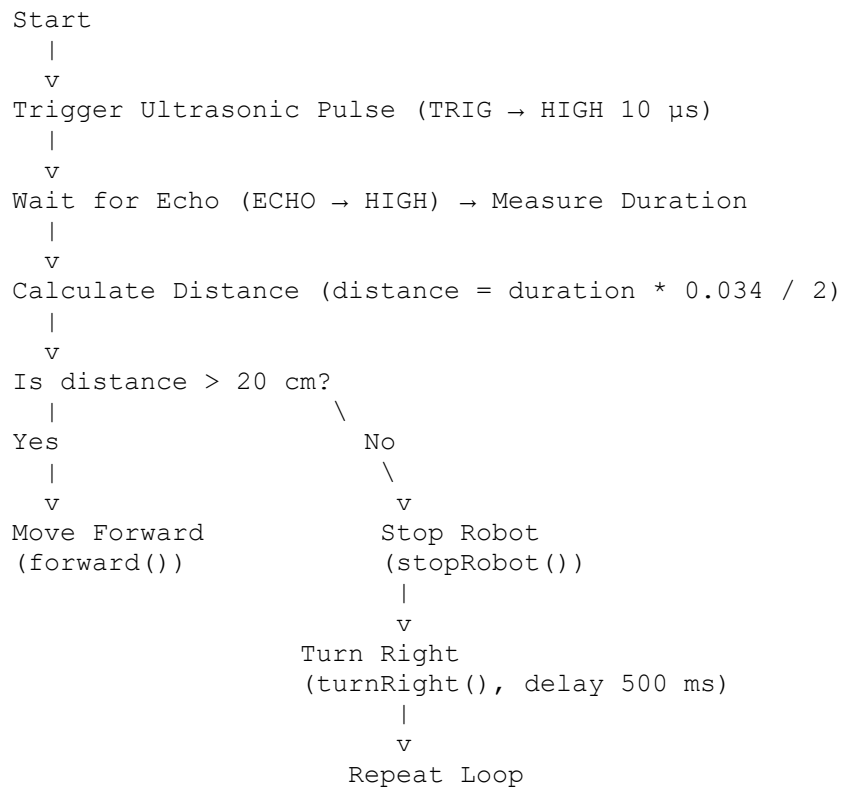
If you want, I can **draw a full flow diagram** showing how **distance** → **decision** → **motors** works, which makes this much easier to visualize.

Do you want me to do that?

yes

Perfect! Let’s create a **flow diagram** of your robot’s logic based on the HC-SR04 sensor and motor control. I’ll explain it step by step.

Flow Diagram of HC-SR04 Robot Logic



Explanation of Each Step

- 1. Trigger Pulse (TRIG HIGH 10 μs)**
 - Arduino “tells” the HC-SR04 to send an ultrasonic sound wave.
- 2. Measure Echo (ECHO HIGH duration)**
 - Arduino listens for the pulse bouncing back. The **longer it stays HIGH**, the farther the obstacle.
- 3. Distance Calculation**
 - $\text{distance} = \text{duration} * 0.034 / 2$ → converts pulse time to centimeters.
- 4. Decision Making**
 - **If distance > 20 cm** → path is clear → robot moves forward.

- **If distance ≤ 20 cm** → obstacle detected → robot stops and turns right.
5. **Motor Control Functions**
- `forward()` → both motors move forward.
 - `stopRobot()` → all motors stop.
 - `turnRight()` → rotate robot clockwise by spinning motors in opposite directions.
6. **Loop Repeats**
- Continuously checks distance and decides movement.
-

□ **Tip:** You can adjust the 20 cm threshold to make the robot react sooner or later depending on the environment.