



Gurukul Technicals Institute

Computer education · Robotics · Science projects · Delhi

FREE EBOOK

ROBOTICS AND ELECTRONICS

Motors, Gears and Motor Drivers

Make things move: DC motors, servos, steppers and how to drive them

INSIDE THIS BOOK

- How a DC motor works
- Speed, torque and current
- BO motors and gear ratios
- Changing direction with an H-bridge
- The L293D driver chip
- ...and 6 more chapters, practice questions and a glossary

Beginner to intermediate
Robotics students and project makers

engineergurukul.com
+91 97116 99511

Motors, Gears and Motor Drivers

Learn how DC motors, servo motors and stepper motors work, how gears trade speed for strength, and how to control motors from an Arduino with L293D and L298N drivers. By the end you can wire, code and power a small robot base and fix its common problems safely.

WHO IT IS FOR	Robotics students and project makers
LEVEL	Beginner to intermediate
SUBJECT AREA	Robotics and electronics
WHAT IS INSIDE	11 chapters, practice questions with answers, and a glossary

HOW TO USE IT

Read one chapter at a time and try the examples as you go: on a computer, at your workbench or in your notebook. The boxes give tips, safety points and short activities. When you finish, test yourself with the practice questions and check your answers at the end. Words you may not know are explained in the glossary.



Gurukul Technicals Institute

A-10, Chief Commissioner Colony, opposite Rana Pratap Bagh, Delhi 110007
Phone 011 2721 1512 · Mobile and WhatsApp +91 97116 99511
engineergurukul@yahoo.co.in · engineergurukul.com

Free ebook. You may read, print and share this ebook for learning. It may not be sold or changed.

© 2026 Gurukul Technicals Institute. First edition, October 2026. The information was checked when it was written; software, prices and rules change, so for anything official always check the latest source.

What you will find inside

1	How a DC motor works	4
2	Speed, torque and current	5
3	BO motors and gear ratios	7
4	Changing direction with an H-bridge	9
5	The L293D driver chip	10
6	The L298N driver module	12
7	Speed control with PWM	14
8	Servo motors	17
9	Stepper motors	19
10	Wheels, chassis and batteries	21
11	Common problems and safety	23
	Practice questions	24
	Answers	25
	Glossary	26

How a DC motor works

Every robot that rolls, lifts or turns needs a motor, which changes electrical energy into movement. This book covers three kinds: DC motors, servo motors and stepper motors. You will learn how each works and how to control it from an Arduino.

Direct current

DC stands for **direct current**, which flows one way only, as from a battery. The supply in the walls of your home is AC (alternating current), which changes direction many times a second. Mains electricity can kill, so every project in this book runs on batteries of 12 volts or less.

The parts inside

In science class you may have learnt that a wire carrying a current inside a magnetic field feels a push. This is the motor effect, and Fleming's left-hand rule tells you which way the push acts. A DC motor uses that push again and again.

Part	What it is	What it does
Stator	The fixed outer case, holding two permanent magnets in small motors	Gives a steady magnetic field
Rotor (armature)	The spinning part: coils of thin copper wire on an iron core	Gets pushed round by the field
Shaft	A metal rod through the rotor	Carries the turning out of the motor
Commutator	A small copper ring cut into segments, fixed to the shaft	Swaps the current in the coils as the rotor turns
Brushes	Two springy strips of metal or carbon	Carry current to the spinning commutator

Why it keeps turning

1. Current flows in through one brush, through the commutator into a coil, and out through the other brush.

2. The coil becomes an electromagnet, and its poles are pulled towards the opposite poles of the fixed magnets.
3. The rotor turns. Just as the coil is about to settle, the commutator segments slide along under the brushes.
4. The current now flows through the coils in a new direction, so the pull starts again the same way round.
5. This repeats many times a second, so the shaft spins smoothly.

REMEMBER

Swap the two battery wires and a DC motor spins the other way. This one fact is the secret behind every motor driver in this book.

The motor is also a generator

When the rotor spins, its coils move through the magnetic field and make a small voltage of their own. This voltage pushes against the battery, so it is called **back EMF** (EMF means electromotive force, an old name for voltage). The faster the motor spins, the bigger the back EMF and the smaller the current. When the shaft is held still there is no back EMF, so the current is at its highest. That is why a stuck motor heats up quickly.

TRY THIS

Connect a toy motor to two AA cells (3 V), with a flag of tape on the shaft. Swap the wires and watch it reverse. Now remove the battery, set a multimeter to DC volts and spin the shaft fast with your fingers. The small voltage shows your motor working as a generator.

DID YOU KNOW

Brushless motors have no brushes or commutator. Electronics switch their coils instead. They are used in drones and in the energy-saving BLDC ceiling fans found in many Indian homes.

CHAPTER 2

Speed, torque and current

Two numbers tell you what a motor can do: how fast and how strongly it turns. A third, current, tells you what it asks from your battery.

Speed and torque

Speed is measured in **RPM**: revolutions (full turns) per minute. A motor at 3,000 RPM makes $3,000 \div 60 = 50$ turns every second.

Torque is turning force: how hard the shaft can twist. Think of a spanner. Push near the nut and it is hard to turn; push at the far end of the handle and the same push turns it easily.

Torque = force $\times$ distance from the centre

Scientists measure torque in newton-metres (N·m). Hobby sellers often use **kg·cm**. A motor with 1 kg·cm of torque can just hold a 1 kg weight hanging from an arm 1 cm long, or 0.5 kg on an arm 2 cm long. 1 kg·cm is about 0.098 N·m.

EXAMPLE

Ananya's motor gives 1.2 kg·cm, and her wheel has a radius of 3 cm. The push at the tyre is $1.2 \div 3 = 0.4$ kg-force, about the push of a 400 g weight. With a bigger 4 cm wheel the push drops to $1.2 \div 4 = 0.3$ kg-force. The robot goes further in each turn, but pushes less hard.

The trade-off

A motor cannot give its highest speed and its highest torque at the same time. As the load grows, it slows down and draws more current.

Situation	Speed	Torque	Current
No load (spinning freely)	Highest	Almost zero	Lowest
Normal working load	Medium	Medium	Medium
Stall (shaft held still)	Zero	Highest	Highest

Current and voltage

Current is measured in amperes (A) or milliamperes (mA); 1 A = 1,000 mA. The **stall current** flows when a motor is stuck or just starting, and it can be several times the running current. Your driver and battery must handle the stall current, not just the running current.

More voltage makes a motor faster and stronger, but too much makes it overheat, so stay close to its rated voltage. Here is how a seller might describe a small geared motor. These figures are made up for learning; read the data for your own motor.

Rating	Example value
Rated voltage	6 V
No-load speed	150 RPM
No-load current	80 mA
Stall torque	1.5 kg-cm
Stall current	600 mA

TIP

To measure running current, connect a multimeter in series with the motor using the meter's 10 A socket and range. The small mA socket has a low-current fuse that a motor can blow. Ask an adult to check your connections first.

CHAPTER 3

BO motors and gear ratios

A bare DC motor spins very fast, often thousands of RPM, but its torque is tiny, and a wheel fixed straight onto it hardly moves a robot. A **gearbox** slows the turning down and multiplies the torque.

BO motors

The most common motor in Indian school robotics is the **BO motor**, short for battery-operated motor: a small DC motor joined to a plastic gearbox, usually in a yellow case. Its shaft has two flat sides, so a wheel fits without slipping. There are straight and L-shaped (90 degree) types, with one or two shafts. Sellers list them by output speed, for example 60, 100, 150 or 300 RPM. Many are rated for about 3 to 12 V and run well near 6 V; check the rating for yours. The listed RPM is usually measured with nothing attached, so on a loaded robot the speed is lower.

What a gear ratio is

When two gears mesh, the gear doing the turning is the **driver** and the gear being turned is the **driven** gear.

Gear ratio = teeth on the driven gear $\div$ teeth on the driver gear

- Output speed = input speed $\div$ gear ratio
- Output torque $\approx$ input torque $\times$ gear ratio (a little less in real life, because of friction)

EXAMPLE

A 10-tooth gear on the motor drives a 30-tooth gear. The ratio is $30 \div 10 = 3$, written 3:1. If the motor spins at 6,000 RPM, the output turns at $6,000 \div 3 = 2,000$ RPM, with about three times the torque.

Gear trains

Real gearboxes use several pairs of gears in a row. Multiply the ratios of the pairs to get the total.

Stage	Driver teeth	Driven teeth	Ratio
First pair	10	50	5:1
Second pair	12	48	4:1
Total			$5 \times 4 = 20:1$

With this 20:1 gearbox, a motor at 9,000 RPM gives $9,000 \div 20 = 450$ RPM.

A **worm gear** is a screw-shaped gear turning a toothed wheel. Each turn of a single-thread worm moves the wheel by one tooth, so a worm and a 40-tooth wheel give 40:1 in one small step. A worm gear also holds its position when the motor stops, which is handy in robot arms.

From RPM to robot speed

In one turn a wheel rolls a distance equal to its circumference: $\pi \times \text{diameter}$, with π about 3.14.

EXAMPLE

Aarav fits 65 mm wheels to 100 RPM motors. One turn moves the robot $3.14 \times 65 \approx 204$ mm. In a minute that is $100 \times 204 = 20,400$ mm, about 20.4 metres, or 0.34 metres per second. His 3-metre classroom takes about $3 \div 0.34 \approx 9$ seconds to cross, or a little more under load.

Motor speed	Top speed with a 65 mm wheel (no load)
60 RPM	about 0.20 m/s
100 RPM	about 0.34 m/s
150 RPM	about 0.51 m/s
300 RPM	about 1.02 m/s

TIP

Slower motors make robots easier to control. Many beginners choose 100 to 150 RPM motors for line followers and obstacle avoiders, and faster ones for a remote-controlled car.

CHAPTER 4

Changing direction with an H-bridge

Swapping the battery wires reverses a DC motor, but a robot cannot ask you to swap wires every time it backs up. The electronic way to do the swap is a circuit called an **H-bridge**.

Four switches in an H

Picture four switches arranged like the capital letter H, with the motor in the middle bar. S1 (top left) and S3 (top right) connect the motor's left and right terminals to battery positive. S2 (bottom left) and S4 (bottom right) connect them to negative.

Switches closed	Current through the motor	Result
S1 and S4	Left to right	Turns one way (call it forward)
S3 and S2	Right to left	Turns the other way (reverse)
None	No current	Coasts to a stop
S1 and S3, or S2 and S4	Both terminals at the same voltage	Brakes quickly
S1 and S2, or S3 and S4	Straight from positive to negative	Short circuit. Never do this.

The last row is dangerous: it connects the battery straight across itself, so a huge current flows, wires heat up and parts are destroyed. This mistake is called **shoot-through**. Driver chips are designed to prevent it when used correctly.

Transistors as switches

Real H-bridges use **transistors**: electronic switches that a small signal from the Arduino turns on and off. A driver chip packs the transistors and the logic that controls them safely into one part.

Protecting against spikes

A motor's coils store energy in their magnetic field. When the current is switched off suddenly, the coil releases it as a short high-voltage spike, which can damage transistors and confuse your Arduino. **Flyback diodes** give the spike a safe path back to the supply. Some driver chips have them built in.

TRY THIS

Build a hand-operated reverser with a DPDT (double pole, double throw) toggle switch, which has six pins in two rows of three. Connect the motor to the two middle pins. Join the top-left pin to the bottom-right pin and connect battery positive there. Join the top-right pin to the bottom-left pin and connect battery negative there. The two joining wires cross like an X. Flip the switch to reverse the motor. Use crocodile clips, or ask an adult to help with soldering.

REMEMBER

An H-bridge lets a small signal choose the direction of a big current. The Arduino gives orders; the driver does the heavy work.

CHAPTER 5

The L293D driver chip

The **L293D** is a 16-pin chip containing two H-bridges, so one chip drives two DC motors forward and backward. It fits straight into a breadboard and has flyback diodes built in; the plain L293, without the D, does not.

What it can handle

- Motor supply: about 4.5 V up to 36 V (our projects use 6 to 9 V)
- Logic supply: 5 V
- Current: about 600 mA per motor continuously, with short peaks up to 1.2 A

That is enough for BO motors and toy motors, but not for big 12 V geared motors.

Finding pin 1

Hold the chip with its legs down and the notch (a small half-moon cut) at the top. Pin 1 is to the left of the notch, often marked with a dot. Pins count down the left side (1 to 8), then up the right side (9 to 16).

Pin	Name	Connect to
1	Enable 1,2	Arduino pin 5 (speed of motor A)
2	Input 1	Arduino pin 7
3	Output 1	Motor A, first wire
4, 5	Ground	Ground
6	Output 2	Motor A, second wire
7	Input 2	Arduino pin 8
8	Motor supply	Battery positive (6 to 9 V)
9	Enable 3,4	Arduino pin 6 (speed of motor B)
10	Input 3	Arduino pin 4
11	Output 3	Motor B, first wire
12, 13	Ground	Ground
14	Output 4	Motor B, second wire
15	Input 4	Arduino pin 3
16	Logic supply	Arduino 5V pin

Wiring it on a breadboard

1. Disconnect all power. Press the chip across the middle gap of the breadboard.
2. Connect pins 4, 5, 12 and 13 to the ground rail. These pins also carry heat away from the chip.
3. Connect pin 16 to the Arduino's 5V pin, and pin 8 to the motor battery's positive wire.
4. Connect the battery's negative wire and the Arduino's GND pin to the same ground rail.
This shared connection is called a **common ground**.

5. Wire the inputs, enables and motors as the table shows, then check every wire before switching on.

Input 1	Input 2	Enable 1,2	Motor A
HIGH	LOW	HIGH	Forward
LOW	HIGH	HIGH	Reverse
LOW	LOW	HIGH	Brakes quickly
HIGH	HIGH	HIGH	Brakes quickly
Any	Any	LOW	Off, free to coast

SAFETY

The L293D gets warm when motors work hard. If it is too hot to touch, switch off at once. A stalled motor or a short circuit is the usual cause.

DID YOU KNOW

Driver chips are not perfect switches. The L293D wastes about 2 V inside itself, more at high currents, so a 6 V battery may give your motors only about 4 V.

CHAPTER 6

The L298N driver module

The **L298N module** is a small board with a big black heat sink, and the most popular motor driver for school robots in India. Like the L293D it holds two H-bridges, but it handles up to about 2 A per motor with its heat sink. Screw terminals mean no soldering.

A tour of the board

Part	What it does
OUT1, OUT2	Motor A
OUT3, OUT4	Motor B
Power terminal marked 12V (VCC or VS on some boards)	Motor battery positive
Power terminal GND	Battery negative and Arduino GND

Part	What it does
Power terminal 5V	5 V from the board's own regulator
IN1, IN2 and IN3, IN4	Direction of motor A and motor B
ENA, ENB (each with a jumper cap)	Speed of motor A and motor B
Jumper near the power terminals	Switches the 5 V regulator on

The jumpers

The L298N chip needs 5 V for its own logic. With the jumper next to the power terminals in place, a small regulator on the board makes this 5 V from your motor battery. For batteries of 12 V or less (all our projects), leave this jumper on. The 5V terminal then becomes an output; in this book we leave it unconnected.

Each enable pin has a jumper cap that joins it to 5 V, so that motor always runs at full speed. To control speed, pull the cap off and connect the enable pin to an Arduino PWM pin. One of the two pins under the cap is 5 V and the other is the real enable input. Read the label on your board, or power the board and measure each pin to GND with a multimeter: the pin showing 5 V is the supply pin, so use the other one.

Wiring table

L298N	Connect to
ENA	Arduino pin 5
IN1	Arduino pin 7
IN2	Arduino pin 8
IN3	Arduino pin 4
IN4	Arduino pin 3
ENB	Arduino pin 6
12V terminal	Battery positive, through an on/off switch
GND terminal	Battery negative and Arduino GND
5V terminal	Nothing
OUT1, OUT2	Left motor
OUT3, OUT4	Right motor

Also connect battery positive (after the switch) to the Arduino's VIN pin, so the Uno's own regulator makes its 5 V. The Uno works best with 7 to 12 V on VIN, so a pack of two Li-ion cells (7.4 V) or six AA cells suits it.

REMEMBER

The GND wire from the Arduino to the driver is not optional. Without a common ground the driver cannot understand the Arduino's signals, and the motors twitch or do nothing.

EXAMPLE

Kabir runs his robot on a 7.4 V pack. The L298N loses about 2 V inside itself, so his 6 V BO motors receive roughly 5 to 5.5 V, which suits them. On a 4.8 V pack they would get under 3 V and crawl.

CHAPTER 7

Speed control with PWM

An Arduino pin is either fully on (5 V) or fully off (0 V). To run a motor at half speed it uses a trick called **PWM**, short for pulse width modulation.

Switching very fast

PWM switches the enable pin on and off hundreds of times a second. If the pin is on for half of each cycle, the motor gets power half the time and runs at roughly half speed. The fraction of time the pin is on is the **duty cycle**.

On an Arduino Uno only pins 3, 5, 6, 9, 10 and 11 can make PWM; they are marked with a wavy line (~). You set the duty cycle with `analogWrite(pin, value)`, where the value goes from 0 to 255.

Value	Duty cycle	Average voltage from a 7 V supply
0	0%	0 V (stopped)
64	25%	about 1.75 V
128	50%	about 3.5 V
191	75%	about 5.25 V
255	100%	7 V (full speed)

On an Uno, PWM switches about 490 times a second on most pins and about 980 times on pins 5 and 6. At low speeds motors may hum; that is normal.

REMEMBER

The IN pins choose the direction; the EN pins choose the speed.

Find your motor's starting value

Below a certain value a motor cannot overcome friction and just hums. Load this sketch with the wheels lifted, open the Serial Monitor at 9600 baud, and note the number shown when the wheel starts to turn.

CPP

```
// Find the lowest PWM value that makes the left motor turn
const int ENA = 5;
const int IN1 = 7;
const int IN2 = 8;

void setup() {
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  digitalWrite(IN1, HIGH); // choose one direction
  digitalWrite(IN2, LOW);
  Serial.begin(9600);
}

void loop() {
  for (int speed = 0; speed <= 255; speed += 5) {
    analogWrite(ENA, speed);
    Serial.println(speed);
    delay(300);
  }
  analogWrite(ENA, 0); // stop, rest, then repeat
  delay(3000);
}
```

A complete two-motor test

This sketch drives a robot forward, backward and round on the spot. `drive()` takes a speed for each wheel from -255 (full backward) to 255 (full forward).

```
// Two-motor test for an L298N or L293D and an Arduino Uno
const int ENA = 5; // left motor speed (PWM)
const int IN1 = 7; // left motor direction
const int IN2 = 8;
const int ENB = 6; // right motor speed (PWM)
const int IN3 = 4; // right motor direction
const int IN4 = 3;

void setup() {
    pinMode(ENA, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(ENB, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    delay(2000); // time to put the robot down
}

void loop() {
    drive(150, 150);    delay(2000); // forward
    drive(0, 0);        delay(1000); // stop
    drive(-150, -150);  delay(2000); // backward
    drive(0, 0);        delay(1000);
    drive(-130, 130);   delay(800);  // spin left
    drive(130, -130);   delay(800);  // spin right
    drive(0, 0);        delay(3000);
}

// Set both motors. Positive = forward, negative = backward.
void drive(int leftSpeed, int rightSpeed) {
    setMotor(ENA, IN1, IN2, leftSpeed);
    setMotor(ENB, IN3, IN4, rightSpeed);
}

void setMotor(int enPin, int inA, int inB, int speed) {
    if (speed > 0) {
        digitalWrite(inA, HIGH);
        digitalWrite(inB, LOW);
    } else if (speed < 0) {
        digitalWrite(inA, LOW);
        digitalWrite(inB, HIGH);
        speed = -speed; // make it positive for analogWrite
    } else {
        digitalWrite(inA, LOW);
        digitalWrite(inB, LOW);
    }
}
```

```
    analogWrite(enPin, constrain(speed, 0, 255));  
}
```

If a motor turns the wrong way, swap its two wires at the driver's OUT terminals instead of changing the code.

TRY THIS

Change `drive(150, 150)` to `drive(150, 100)`. The left wheel now turns faster than the right, so the robot curves to the right. Steering by speed difference is how almost every two-wheeled robot turns.

CHAPTER 8

Servo motors

A DC motor just spins. Sometimes you want something else: turn to exactly 45 degrees and stay there. That is the job of a **servo motor**. Servos point robot heads, move robot arms and steer model cars.

What is inside a servo

1. A small DC motor.
2. A gearbox that slows it down and adds torque.
3. A **potentiometer** (a turning sensor) on the output shaft, which measures the shaft's angle.
4. A control circuit that compares the angle you asked for with the angle measured, and drives the motor until they match.

Checking its own position like this is called **closed-loop control**, or feedback. Push the arm away and the servo pushes back to its commanded angle.

The SG90 micro servo

The **SG90** is the small blue servo in most robotics kits, often called a 9 g servo because it weighs about 9 grams. It turns through about 180 degrees and runs on about 5 V (4.8 to 6 V).

Wire colour	Connect to
Brown	GND

Wire colour	Connect to
Red	5 V
Orange	Arduino signal pin (pin 9 here)

How the signal works

The Arduino sends a pulse about every 20 milliseconds (50 times a second), and the length of each pulse sets the angle. Roughly, 1 ms means one end, 1.5 ms the middle and 2 ms the other end. Real servos vary, and many need a slightly wider range to reach the full 180 degrees. The built-in Servo library times the pulses for you.

CPP

```
// Sweep an SG90 servo from 0 to 180 degrees and back
#include <Servo.h>

Servo myServo;           // make a servo object

void setup() {
  myServo.attach(9);      // signal wire on pin 9
}

void loop() {
  for (int angle = 0; angle <= 180; angle++) {
    myServo.write(angle);
    delay(15);            // give the servo time to move
  }
  for (int angle = 180; angle >= 0; angle--) {
    myServo.write(angle);
    delay(15);
  }
}
```

TRY THIS

Connect a 10 kΩ potentiometer with its outer legs to 5V and GND and its middle leg to A0. In the sketch above, replace everything inside the curly brackets of `loop()` with three lines: `int angle = map(analogRead(A0), 0, 1023, 0, 180);` then `myServo.write(angle);` then `delay(20);`. Now the servo follows the knob.

TIP

If the servo buzzes at 0 or 180 degrees, it is pressing against its internal stop, so use a smaller range such as 10 to 170. Never twist the arm by hand: the plastic gears strip easily.

REMEMBER

On an Arduino Uno, the Servo library stops PWM working on pins 9 and 10, even if your servo is on another pin. Keep motor-speed (EN) wires on pins 3, 5, 6 or 11 when a servo shares the project.

CHAPTER 9

Stepper motors

A **stepper motor** turns in small, equal steps. Send it 100 steps and it moves exactly 100 steps, then holds still. Steppers move 3D printer heads, where exact positions matter.

How a stepper moves

A stepper has several coils around a magnetic rotor. The driver switches the coils on one after another, and each switch pulls the rotor on by one step. Reverse the order and it turns the other way; switch faster and it turns faster.

A stepper does not check its own position. This is called **open-loop control**. If the load is too heavy or the switching too fast, it can miss steps, and the Arduino will not know.

The 28BYJ-48 and the ULN2003 board

The **28BYJ-48** is a small, cheap 5 V stepper with a built-in gearbox. It usually comes with a **ULN2003** driver board. The ULN2003 chip holds seven transistor switches, of which the board uses four, plus four LEDs that blink as each coil switches on, which helps with fault-finding.

Because of its gearbox, the 28BYJ-48 needs about 2,048 steps (in full-step mode) to turn its output shaft once. It is slow but precise: its top speed is about 10 to 15 RPM, and if you ask for more it buzzes and misses steps.

ULN2003 board	Connect to
IN1, IN2, IN3, IN4	Arduino pins 8, 9, 10, 11
Power + (marked 5-12V on many boards)	5 V supply
Power –	GND, joined to Arduino GND
White socket	Motor plug (it fits only one way)

The motor draws a couple of hundred milliamps. For a short test the Uno's 5V pin on USB power works, but a separate 5 V supply with a common ground is better for long runs.

Code for one turn each way

The Stepper library is built in. Notice the pin order: for this motor the coils must be listed as IN1, IN3, IN2, IN4. With the plain order 8, 9, 10, 11 the motor only vibrates.

CPP

```
// 28BYJ-48 stepper with a ULN2003 board: one turn each way
#include <Stepper.h>

const int STEPS_PER_TURN = 2048;
Stepper stepper(STEPS_PER_TURN, 8, 10, 9, 11); // IN1, IN3, IN2, IN4

void setup() {
  stepper.setSpeed(10); // 10 RPM
}

void loop() {
  stepper.step(STEPS_PER_TURN); // one turn forward
  delay(1000);
  stepper.step(-STEPS_PER_TURN); // one turn back
  delay(1000);
}
```

EXAMPLE

Sana wants the shaft to turn 90 degrees. A full turn is 360 degrees and 2,048 steps, so she needs $2,048 \times 90 \div 360 = 512$ steps: `stepper.step(512)`; . For 30 degrees she needs $2,048 \times 30 \div 360 \approx 171$ steps.

Need	Best choice	Why
Wheels that spin continuously	Geared DC (BO) motor	Simple, fast, strong
Point to an angle and hold it	Servo	Built-in feedback
Move an exact amount, slowly	Stepper	Counts steps precisely

SAFETY

A stepper's coils stay powered even when it is still, so the motor and driver get warm. Unplug it when you finish testing.

Wheels, chassis and batteries

The wheels, body and battery decide whether a robot glides or limps.

Two-wheel drive

Most school robots use **differential drive**: two motor-driven wheels, one on each side, plus a free-turning **caster** (often a metal ball in a holder) to stop the robot tipping. There is no steering wheel. The robot turns by running its wheels at different speeds.

Left wheel	Right wheel	Robot does
Forward	Forward	Goes straight
Forward fast	Forward slow	Curves right
Stopped	Forward	Pivots left around the left wheel
Backward	Forward	Spins left on the spot

The chassis

The **chassis** is the robot's body frame. Ready-made 2WD acrylic kits include motor clamps, wheels, a caster, a battery holder and screws, or you can cut your own from MDF or foam board (sunboard). Either way:

- Mount both motors exactly in line, so the robot drives straight.
- Push the wheels fully onto the shafts so they do not wobble.
- Keep the battery low and near the middle, so the robot does not tip.

SAFETY

Cutting acrylic or MDF and using hot glue need an adult. Wear safety glasses when drilling or cutting.

Choosing a battery

First add up the current your robot needs. These are rough examples; check your own parts.

Part	Rough current
Two BO motors, running	200 to 400 mA together, much more when starting

Part	Rough current
Arduino Uno	about 50 mA
Sensors and LEDs	20 to 50 mA
Total while driving	about 300 to 500 mA

Battery capacity is printed in **mAh** (milliamp-hours). In theory, running time in hours = capacity in mAh ÷ current in mA.

EXAMPLE

Priya's robot draws about 400 mA from a 2,000 mAh pack. In theory it runs $2,000 \div 400 = 5$ hours. Starting surges and falling voltage give less, so she plans for well under 5 hours.

Battery	Voltage	Good for robots?
9 V PP3 (rectangular)	9 V	No: it cannot supply motor currents, so robots crawl and Arduinos reset
4 × AA alkaline	about 6 V	Fine for small robots, but weakens fast under load
6 × AA rechargeable (NiMH)	about 7.2 V	Good, and handles high current well
2 × 18650 Li-ion cells	7.4 V (8.4 V fully charged)	Very good, with care

EXAMPLE

Suppose six AA alkaline cells cost ! 150 and last four practice sessions. A club meeting 40 times a year needs ten sets: ! 1,500. If six rechargeable cells and a charger cost ! 1,200 and last the year, the club saves ! 300 and throws away fewer cells.

SAFETY

Li-ion cells store a lot of energy. Use protected cells or a pack with a protection board, charge only with a proper Li-ion charger, never leave them charging unattended, and never short, crush or pierce them. Students should handle them with an adult's guidance.

Common problems and safety

Most motor troubles come from weak power, electrical noise or loose wiring.

Brown-outs

Starting motors draw a big surge of current. A weak battery's voltage dips for a moment, and if it dips too low the Arduino resets and starts its program again. This is a **brown-out**. Signs: a robot that stops and restarts every time it moves, or a servo that jumps back to its starting position.

1. Use fresh or fully charged batteries, never a 9 V PP3 for motors.
2. Keep power wires short and thick enough.
3. Add a large capacitor, for example 470 μF rated at 16 V or more, across the motor supply near the driver.
4. Power the Arduino through VIN, never straight from the motor battery to its 5V pin.

SAFETY

Electrolytic capacitors have a positive and a negative leg, and the stripe on the side marks negative. Fitted backwards, a capacitor can swell or burst.

Electrical noise

Brushes spark slightly as they slide over the commutator, and the sparks make electrical noise that can upset sensors and the Arduino. A small 0.1 μF ceramic capacitor (marked 104) soldered across each motor's terminals removes much of it, and twisting each motor's two wires together helps too. Soldering needs an adult's supervision.

Fault-finding table

Problem	Likely cause	What to do
Motors hum but do not turn	PWM value too low, or weak battery	Raise the speed value; charge the battery
One motor runs backward	Its wires are swapped	Swap them at the OUT terminals
Speed control has no effect	ENA or ENB jumper cap still on	Remove the cap and wire the enable pin

Problem	Likely cause	What to do
Nothing moves, Arduino works	No common ground, or battery switched off	Join the grounds; check the switch
Robot curves on a straight run	Motors are not identical	Lower the faster side's speed value
Driver very hot	Motor stalled, or too much current	Switch off; free the wheels; check for shorts
Stepper vibrates but does not turn	Wrong pin order	Use the order 8, 10, 9, 11 in the code
Servo jitters	Weak 5 V supply	Use a separate 5 V supply with a common ground

Safety rules for moving machines

- Use batteries of 12 V or less. Never connect any project to mains power.
- Use red wire for positive and black for negative, and switch off before changing any wire.
- Keep fingers, hair, dupattas and loose sleeves away from gears and wheels.
- Lift the wheels off the table when testing new code, so the robot cannot run off the edge.
- Switch off at once if anything smells hot, smokes or crackles.
- Soldering, hot glue, cutting and Li-ion charging need an adult nearby.

REMEMBER

Check power, ground and wiring before every run. A robot that is safe to test is a robot you can improve.

CHECK YOURSELF

Practice questions

1. What does DC stand for, and how is it different from AC?
2. Name the part of a DC motor that swaps the current in the coils as the rotor turns.
3. What is back EMF, and why does a stalled motor get hot?
4. A motor turns at 1,800 RPM. How many turns does it make each second?

5. A motor gives 2 kg·cm of torque and drives a wheel of radius 4 cm. What push does the tyre give?
6. A 12-tooth gear drives a 48-tooth gear. What is the gear ratio, and what is the output speed if the input is 6,000 RPM?
7. A gearbox has two stages, 5:1 and 6:1. What is the total ratio?
8. A robot has 65 mm wheels on 150 RPM motors. About how far does it travel in one minute with no load?
9. In an H-bridge, which pairs of switches must never be closed together, and why?
10. What job do flyback diodes do?
11. On the L298N module, what happens to a motor's speed if the ENA jumper cap is left on?
12. Why must the Arduino GND be connected to the motor driver's GND?
13. Name the six PWM pins on an Arduino Uno.
14. What does `analogWrite(5, 128)` do to a motor whose ENA is on pin 5?
15. What are the three wire colours on an SG90 servo, and where does each connect?
16. Why should motor enable wires avoid pins 9 and 10 when a servo is used on an Uno?
17. About how many full steps turn the 28BYJ-48 output shaft once? How many give 45 degrees?
18. Why is a 9 V PP3 battery a poor choice for a robot with motors?
19. A robot draws 500 mA from a 2,500 mAh pack. What is the running time in theory?
20. What is a brown-out? Name two ways to prevent one.

ANSWER KEY

Answers

1. Direct current. DC flows in one direction only, as from a battery; AC changes direction many times a second.
2. The commutator.
3. Back EMF is the voltage a spinning motor makes, which pushes against the supply. A stalled motor makes none, so a large current flows and heats the coils.
4. $1,800 \div 60 = 30$ turns per second.

5. $2 \div 4 = 0.5$ kg-force.
6. $48 \div 12 = 4$, so 4:1. Output speed = $6,000 \div 4 = 1,500$ RPM.
7. $5 \times 6 = 30$, so 30:1.
8. One turn is about $3.14 \times 65 \approx 204$ mm, and $150 \times 204 = 30,600$ mm, so about 30.6 metres.
9. The two switches on the same side (S1 with S2, or S3 with S4). They connect the battery straight across itself: a short circuit called shoot-through.
10. They give the voltage spike from a motor coil a safe path when the current switches off, protecting the transistors and the Arduino.
11. The enable pin stays joined to 5 V, so the motor always runs at full speed and PWM control does nothing.
12. Signals are measured against ground. Without a shared ground the driver cannot read the Arduino's HIGH and LOW signals.
13. Pins 3, 5, 6, 9, 10 and 11.
14. It sets a 50% duty cycle, so the motor runs at roughly half speed.
15. Brown to GND, red to 5 V, orange to the Arduino signal pin.
16. On an Uno, the Servo library stops PWM working on pins 9 and 10, so speed control there would fail.
17. About 2,048 steps. For 45 degrees: $2,048 \times 45 \div 360 = 256$ steps.
18. It cannot supply the large current motors need, so its voltage drops, the robot crawls and the Arduino may reset.
19. $2,500 \div 500 = 5$ hours in theory, and less in real use.
20. A voltage dip, caused by a motor surge, that resets the Arduino. Prevent it with charged batteries, short thick wires and a large capacitor across the supply.

WORD LIST

Glossary

Back EMF

The voltage a spinning motor makes, which pushes against the supply and lowers the current.

BO motor

A small battery-operated DC motor with a plastic gearbox, common in school robots.

Brown-out

A short dip in supply voltage that can reset an Arduino.

Common ground

A shared GND connection between the Arduino, the driver and the battery.

Commutator

A segmented ring on a motor shaft that swaps the current in the coils as it turns.

Differential drive

Steering by running the left and right wheels at different speeds.

Duty cycle

The percentage of each PWM cycle for which the signal is on.

Flyback diode

A diode that gives the voltage spike from a motor coil a safe path.

Gear ratio

Teeth on the driven gear divided by teeth on the driver gear.

H-bridge

A circuit of four switches that can send current through a motor either way.

L293D

A 16-pin chip with two H-bridges and built-in diodes, for small motors.

L298N

A dual H-bridge driver chip, usually sold on a module with a heat sink, for motors up to about 2 A.

mAh

Milliamp-hours, a measure of how much charge a battery can deliver.

PWM

Pulse width modulation: switching a pin on and off fast to control average power.

Servo motor

A motor with feedback that turns to a commanded angle and holds it.

Shoot-through

A short circuit caused by closing both switches on one side of an H-bridge.

Stepper motor

A motor that moves in fixed, equal steps.

Torque

Turning force, equal to force multiplied by distance from the centre.



Gurukul Technicals Institute

Computer education institute in Delhi, with authorised member centres, robotics classes and science projects, working models and DIY kits.

COURSES

Computer & Office Skills · Accounting & Business Software · Programming & Web Development · Design & Digital Skills · Hardware & Networking · Robotics & Coding · Other Programmes

PROJECTS AND KITS

Ready-made school projects · DIY and STEM kits · Robotics and coding · Components and spares · Engineering projects

CONTACT

Head office: A-10, Chief Commissioner Colony, opposite Rana Pratap Bagh, Delhi 110007

Phone: 011 2721 1512 · **Mobile and WhatsApp:** +91 97116 99511

Email: engineergurukul@yahoo.co.in

Website: engineergurukul.com · **Instagram:** [@science_projects_club](https://www.instagram.com/science_projects_club)

More free ebooks for computer courses, robotics and science projects: engineergurukul.com/resources/ebooks/

Free to read, print and share. Not for sale.