

Magnets



how
electric motors work



[EN](#) / [RU](#)

A magnet has a north pole (red) and a south pole (blue). Opposite poles attract, same poles repel each other.

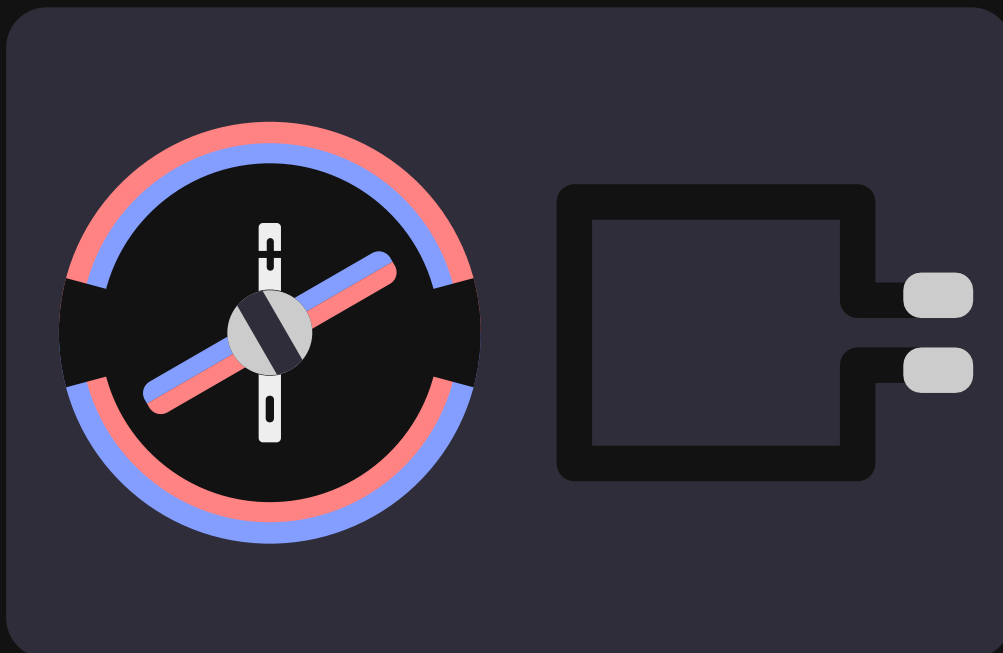
Electromagnets only work as magnets when there's electricity flowing through them. If you turn up the current, the magnet becomes stronger. If you connect it backwards, the poles switch.

Motors

DC Motors

DC motors are the simplest. They have curved magnets on the top and bottom.

In the middle is an electromagnet (dark frame with gray contacts) powered by two (white) brushes.



When you connect power (slide voltage to the right), the top of the electromagnet repels the permanent magnet above and is attracted to the magnet below, so it turns. When it completes half a turn, the brushes disconnect. No power = the electromagnet doesn't work. It keeps spinning (because it is already in motion) until it reconnects with the brushes the opposite way, making its poles switch.

Now the same side repels the bottom magnet and attracts to the top one, so the motor completes the turn.

Powerful DC motors are large and expensive, but small ones are widely used. Look up [“yellow DC motor”](#), they come with a reduction drive (the motor itself spins at 9000+ RPM) and are very common for hobbyist electronics projects.

To control a small one, you need a **DC motor driver** (mx1508, for example), since the motor draws more power than a microcontroller (MCU, a small computer) can supply and requires switching polarity (swapping + and -) to change the direction, which the MCU also can't do. To find how to connect one, look up [“mx1508 pinout”](#) or [“mx1508 arduino”](#).

Control is something like:

```
import board
import pwmio
from adafruit_motor import motor

pwm_a = pwmio.PWMOut(board.D5, frequency=1600)
pwm_b = pwmio.PWMOut(board.D6, frequency=1600)
m1 = motor.DCMotor(pwm_a, pwm_b)

m1.throttle = -0.6 # backward, 60% power
```

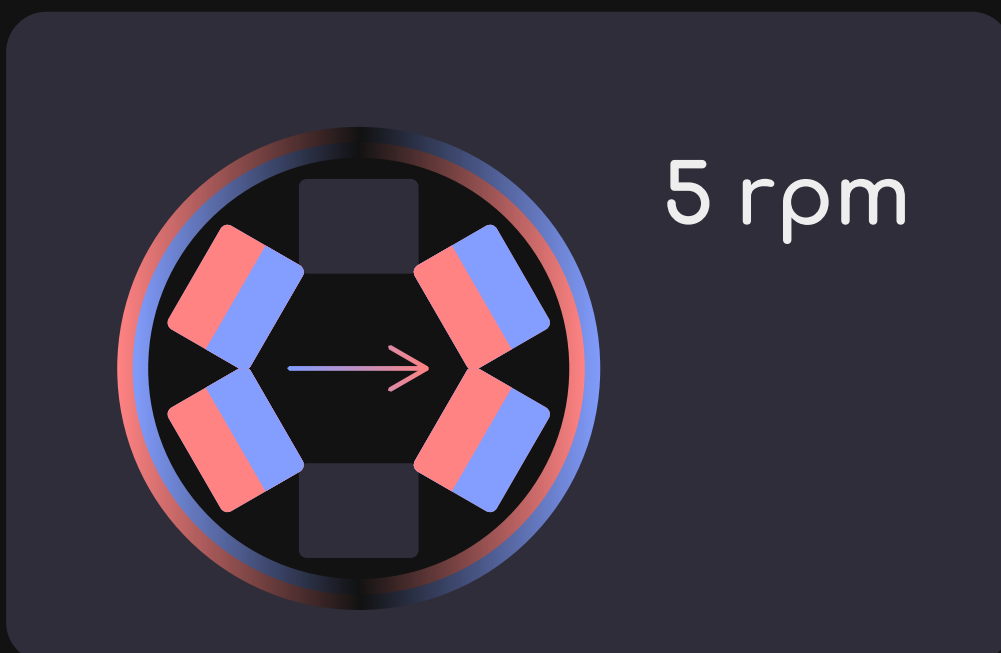
More on that later.

Brushless (synchronous)

Brushless motors are smaller, more powerful, but harder to control.

They have many small magnets along the inside of the rotor (the rotating part) which act as one big round magnet.

In the middle, there are 6 (sometimes 9, usually 12) electromagnets that turn on one after the next in 3 groups, creating a **rotating magnetic field**. The rotor turns in sync with that field (hence the name). Since the electromagnets are not spinning, they don't need brushes, wires connect directly (hence the other name).



The real RPM is usually 3000+.

To connect one, you use an electronic speed controller (ESC). Its job is to power the motor (like a driver) and turn a simple signal that the MCU gives it over a single wire into the complex control signals for the motor (shown above) that go over three.

Control:

```
import board
import pwmio
from adafruit_motor import servo

pwm = pwmio.PWMOut(board.D5, frequency=50)
m1 = servo.ContinuousServo(pwm, min_pulse=1000, max_pulse=2000)

m1.throttle = -0.2 # backward, 20% power
```

Stepper

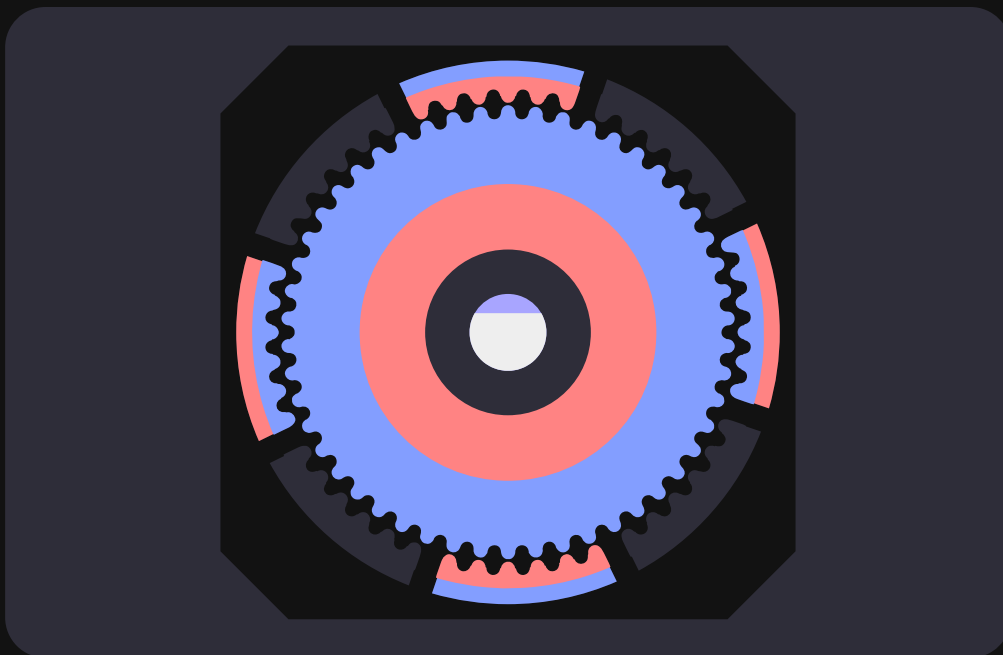
If you need to turn a DC or a brushless motor to a certain angle, you need a sensor to measure its position.

Stepper motors are a simpler alternative. Pulse the STP pin of its driver ([A4988](#), [DRV8835](#)) and the motor turns 1.8 degrees. Pulse it again and it turns another 1.8 degrees. Put 3 volts on the DIR pin of the driver, pulse STP and it moves 1.8 degrees the other way. That's it.

If you need to go slow, pulse it slow. If you need it to go faster, pulse it faster.

The driver also has a current limiting dial that makes sure the motor doesn't overheat, more about it [here](#).

The actual motor has a magnetic rotor, the outer edge of this one is the north pole. The stator has 8 electromagnets with teeth. Notice the bottom teeth are aligned with the rotor and the top are unaligned.



When the stator teeth act as a south pole, the rotor teeth turn to face them. When they act as a north pole, the rotor teeth rotate to be in the gaps, maximizing the distance.

Microsteps is when the driver smoothly controls the power to each set of electromagnets instead of just turning them on/off.

Control:

```
import time
import board
from digitalio import DigitalInOut, Direction

dir = DigitalInOut(board.D5)
stp = DigitalInOut(board.D6)
dir.direction = Direction.OUTPUT
stp.direction = Direction.OUTPUT

def move(steps, direction):
    dir.value = direction
    time.sleep(0.001)

    for _ in range(steps):
        step_pin.value = True
        time.sleep(0.001)
        step_pin.value = False
        time.sleep(0.001)

move(100, False) # half a turn
```