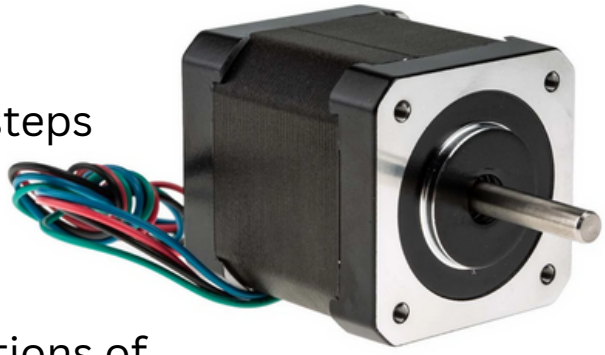
Gears and motion motorized

Continuation from "gears and motion" worksheet

What is a stepper motor?

- Unlike convensional dc motors that spin continuously, stepper motors rotate in steps where each step is a combination of electromagnets being activiated.

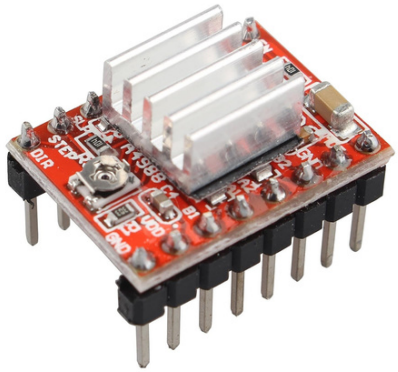


Nema 17 Stepper motor
200 steps per revolution

- to get it to rotate, you activate combinations of the electromagnets in a specific order.

How to control it with a pico?

- Directly connecting any motor to a pi pico (apart from small servos) is a bad idea because of how much power they require in comparison to other electrical components.



A4988 Driver board

- To work around this limitation, “driver boards” are used as an intermediary between motors and microcontrollers, combining the power supplied from a seperate source and the data signals provided by a micro controller (Pi Pico)

- The driver board simplifies the logic we have to implement in software to 3 main controls: motor direction, when to step, resolution of each step.

Remember to add a capacitor between the 12V power and ground pins to prevent the motor from jittering

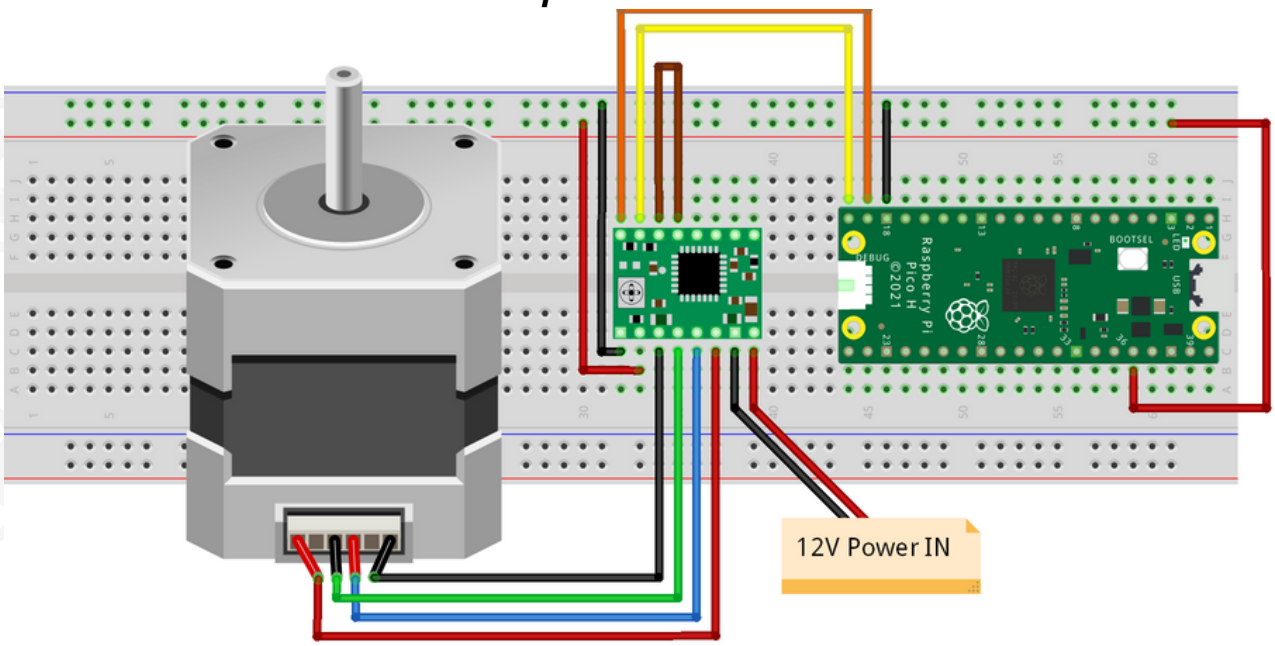
- Motor direction is self explainatory, to step you simply provide a high value to a pin on the driver board, and specifying the resolution allows you to do 1/2, 1/4, 1/8 and 1/16 steps for finer movement control.

Part 1: Making a stepper motor spin

Before we start adding stepper motors to last weeks mechanical turret, lets get a single motor spinning.

- the diagram bellow shows you what needs to be wired up where with the Stepper motor on the left, driver board in the middle and Pi Pico on the right.
- Because motors require much more power that what a usb port or Pi Pico can provide, we need to give the driver board an extenal supply.

To prevent accidental short circuits, do not connect the 12V supply until you are ready to test your code and confident with the wiring, if you are unsure please ask

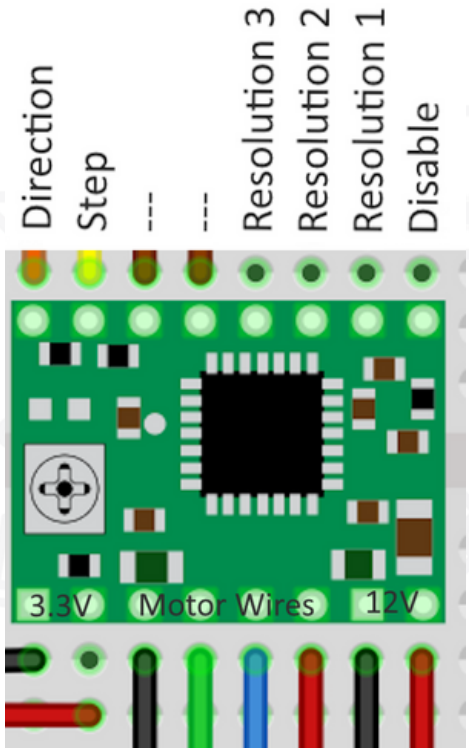


You can add some masking tape to the motor to better visualize rotation

- Once everything is wired up (except 12V power) you can connect your Pi Pico to a computer and implement the following code:

```
from machine import Pin    # Importing the basic libraries
import time


direction = Pin(14, Pin.OUT)  # Specifying which wire controls the direction
step = Pin(15, Pin.OUT)       # Specifying which wire controls the step

direction.on() # sets the direction, replacing .on() with .off() will flip the direction


# A full stepper motor rotation consists of 200 steps

for i in range(200):     # This loop performs 200 steps by providing 200 pulses to the step pin
    step.on()
    time.sleep(0.01)
    step.off()
    time.sleep(0.01)
```

- As stated earlier it is possible to specify the resolution of a step by connecting up some extra wires.
- For a 1/2 step connect **Resolution 1** to **V++**
- For a 1/4 step connect **Resolution 2** to **V++**
- For a 1/8 step connect **Res 1 and 2** to **V++**
- And for 1/16 step connect **Res 1, 2 and 3** to **V++**



- Try out different resoltions and see the effect it has on the motor.
- You may need to ground unused **Resolution** pins if your motor is behaving oddly

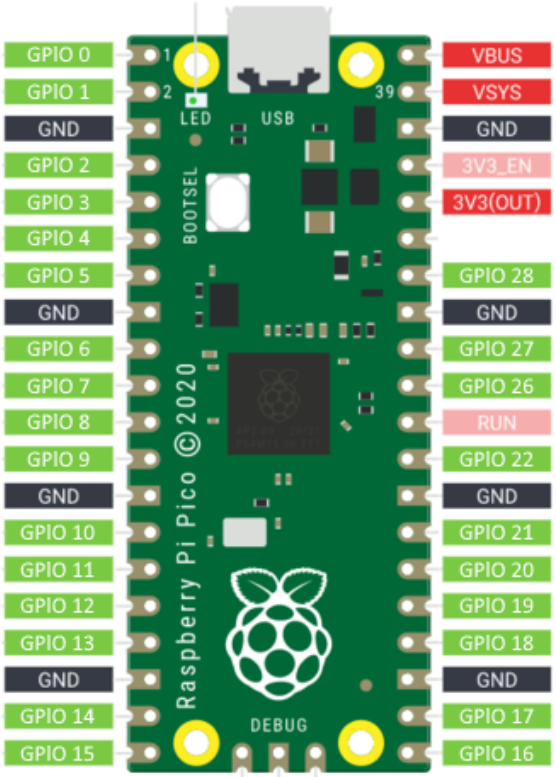
Part 2: Making 2 motors spin

By now you should have a single motor spinning, now duplicate the circuit and code to allow 2 motors to spin.

To help, your code should look like this:

```
from machine import Pin
import time

direction1 = Pin(14, Pin.OUT)
step1 = Pin(15, Pin.OUT)

direction2 = Pin(16, Pin.OUT)
step2 = Pin(17, Pin.OUT)


for i in range(800):
    step1.on()
    step2.on()
    time.sleep(0.002)
    step1.off()
    step2.off()
    time.sleep(0.002)
```

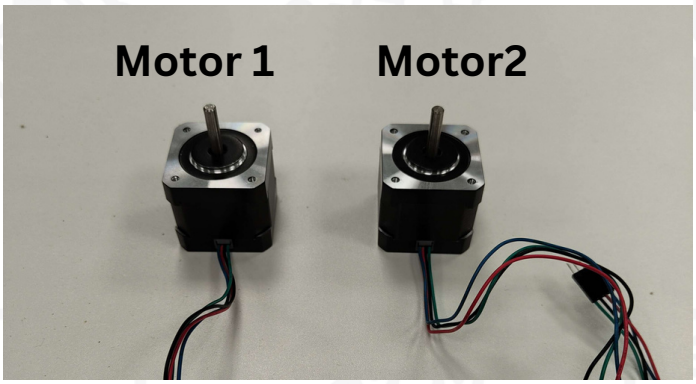
Remember to add a capacitor between the 12V power and ground pins to prevent the motor from jittering

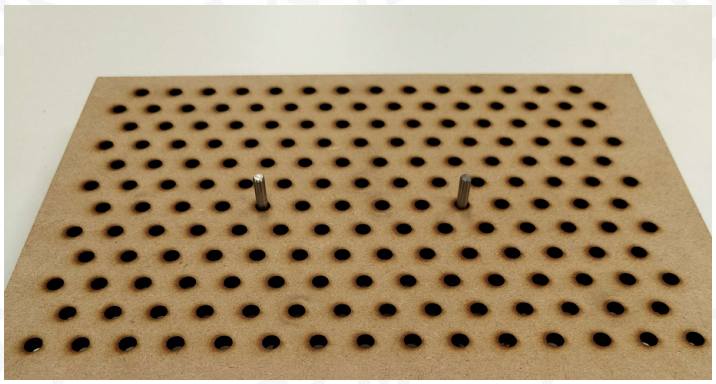


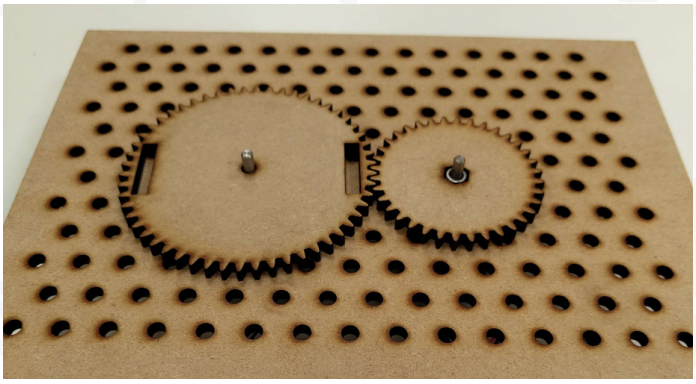
Task 3: Making a motorized turret

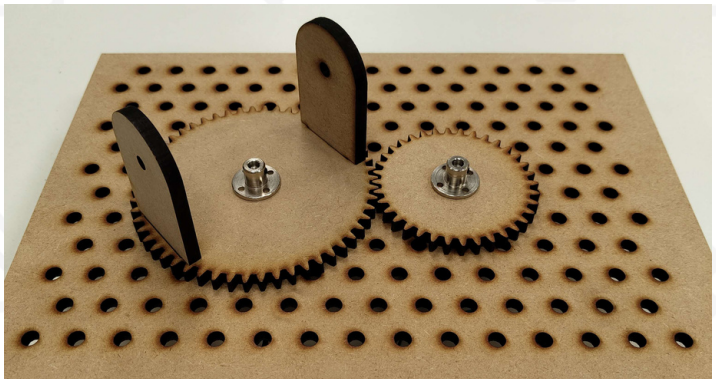
In the “Gears and Motion” worksheet you would have covered how to make a mechanical turret, now we will motorize it.

To get started rebuild the turret as you did previously, visual instructions are bellow if you forgot

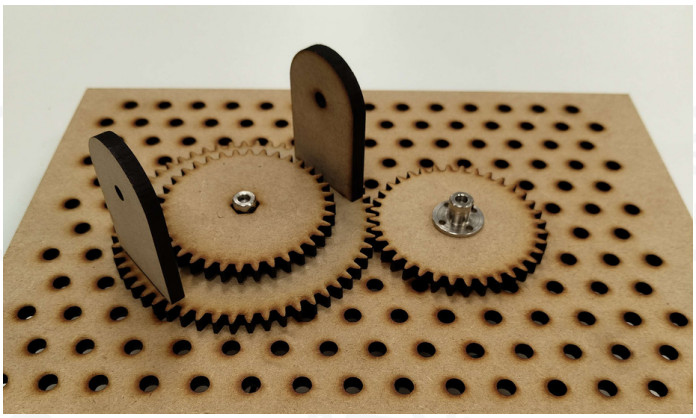
1. 

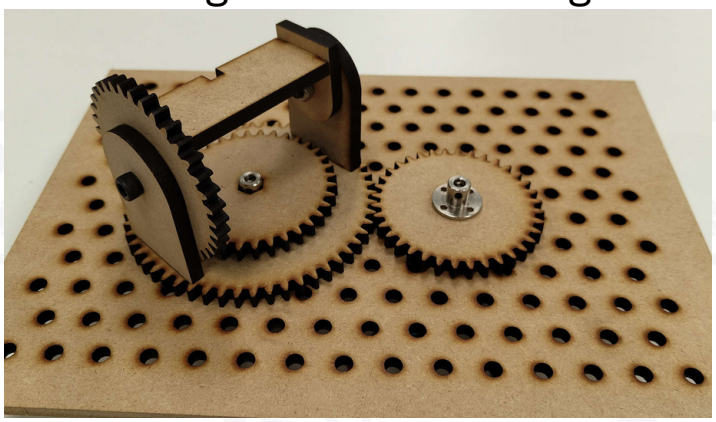
2. 

3. 

4. 

Make sure the grub screws are tight ^

5. 

6. 

Once you have assembled the turret we can modify our code to control both motors with set angles:

```
from machine import Pin
import time

stepDivision = 0.25
stepsPerDegree = (200 / stepDivision) / 360

move1 = 30 # degrees for motor 1 to rotate
move2 = 80 # degrees for motor 2 to rotate

direction1 = Pin(14, Pin.OUT)
step1 = Pin(15, Pin.OUT)

direction2 = Pin(16, Pin.OUT)
step2 = Pin(17, Pin.OUT)

#direction1.on()  # uncomment this line to switch motor 1's spin direction
#direction2.on()  # uncomment this line to switch motor 2's spin direction

noSteps1 = move1 * stepsPerDegree
noSteps2 = move2 * stepsPerDegree

for i in range(max(noSteps1, noSteps2) * stepsPerDegree): # iterates over the biggest number of step

    if i < noSteps1: step1.on() # if motor 1 still has steps remaining
    if i < noSteps2: step2.on() # if motor 2 still has steps remaining
    time.sleep(0.002)
    if i < noSteps1: step1.off()
    if i < noSteps2: step2.off()
    time.sleep(0.002)
```

Review and extra challenges:

Congrats! Today you have combined everything you have learnt so far in electronics mechanics and code to create your first proper robot.

Challenge 1: Make both motors start and stop at the same time even if they have a different number of degrees to move, so one motor would move at a slower rate than the other.

Challenge 2: Incorporate a potentiometer to control one of the motors.

Thinking point: If you were having trouble getting your motor to spin consistently, what do you think could've cause this?

Pi Pico Code Docs

Worksheet produced by Lukas Hastings for use by HackSussex
https://github.com/supersand21/Robotics-Workshop