

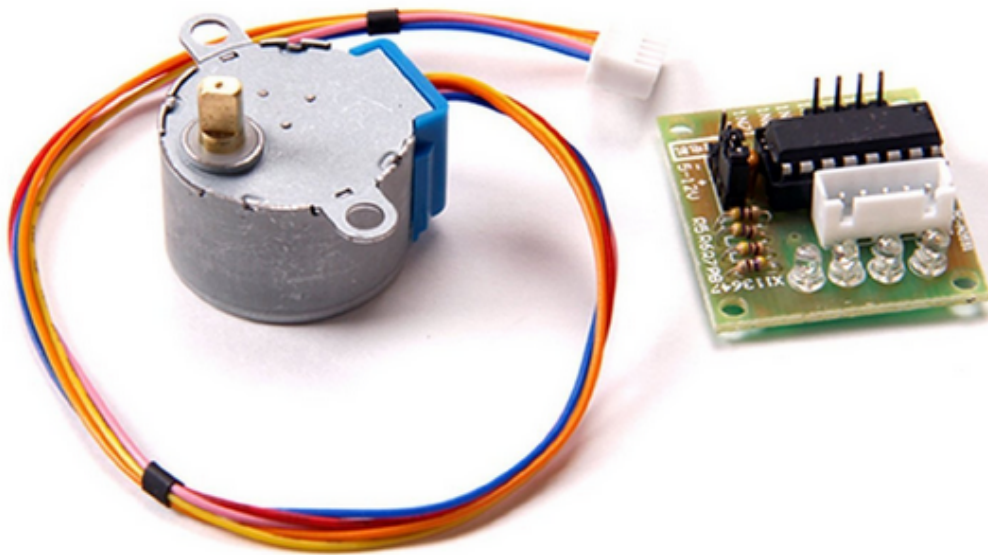
Exercises 7: Stepping Motors

Goal:

A stepping motor is a device which is often used when a precise amount of movement is required: an exact angle or an exact position on a linear scale. While DC motors move while they are powered a stepping motor requires a sequence of impulses to move step by step as its name suggests.

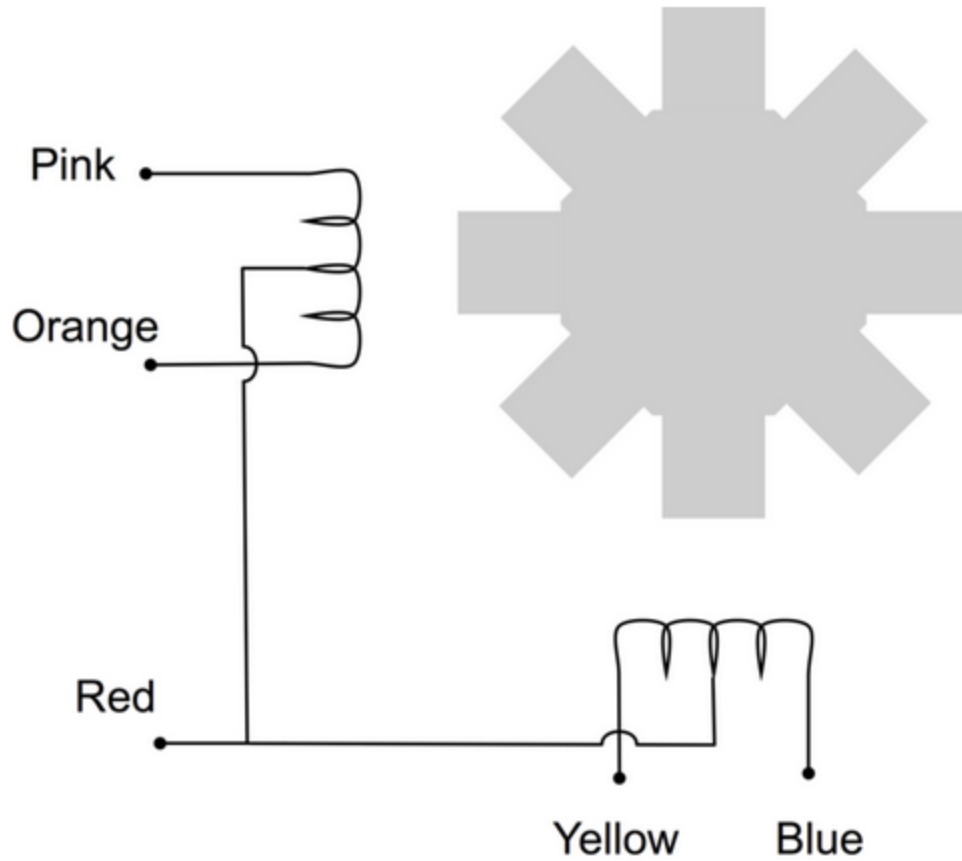
In this exercise we will learn how to program different step sequences for full step or half step movement and how to move the motor forward and backward.

This is a photo of the motor itself and its driver module featuring an ULN2803 driver chip.



Exercise 1: Make sure the connections are correct

The stepping motor is has 5 connections which are provided by the cable from the driver module to the motor itself. The 2 coils must be energized in the right order.



To do this we have 6 connections from the driver board to the Raspberry Pi:

- 5V (check for “5V +” and “-” on the board)
- Gnd
- 4 phases denominated In0..In3

The state of each of the In lines is shown on its corresponding LED. As we will have to provide a sequence of

In 0	In 1	In 2	In 3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Connect the stepping motor to the driver board and the driver board to the Raspberry Pi. The In0 .. In3 lines must be connected to GPIO pins (e.g. 17,18,27,22). Now set only the GPIO line, to which In0 is connected, to high, all the others to low and make sure that the corresponding LED lights up. Do the same thing for all the connected GPIO pins.

Exercise 2: Write a program to move the stepping motor by 512 steps.

The above sequence must be sent to the stepping motor to make it move 1 full step. Make sure to wait at least 25 ms in between each setting:

- set 1 0 0 0, wait 25ms
- set 0 1 0 0, wait 25 ms ...

Write a subroutine which moves the motor by 1 step. Then call this subroutine 512 times. Use C arrays to define which In line corresponds to which GPIO line and a 2 dimensional array to mimic the above table.

How far does the motor move with 512 steps? How can you lower the speed of the movement?

Exercise 3: Improve your program to give user control over its parameters

Modify your program in such a way that the user can define its behavior through command line parameters:

- -f or -forward: move forward (this is the default)
- -b or -backward: move backward
- -n or -numberOfSteps: number of steps (default 512)
- -v or -velocity: velocity of movement
- -s or -fullStep: full step (this is the default)
- -h or -halfStep: half step