Ch 9 Stepper Motors

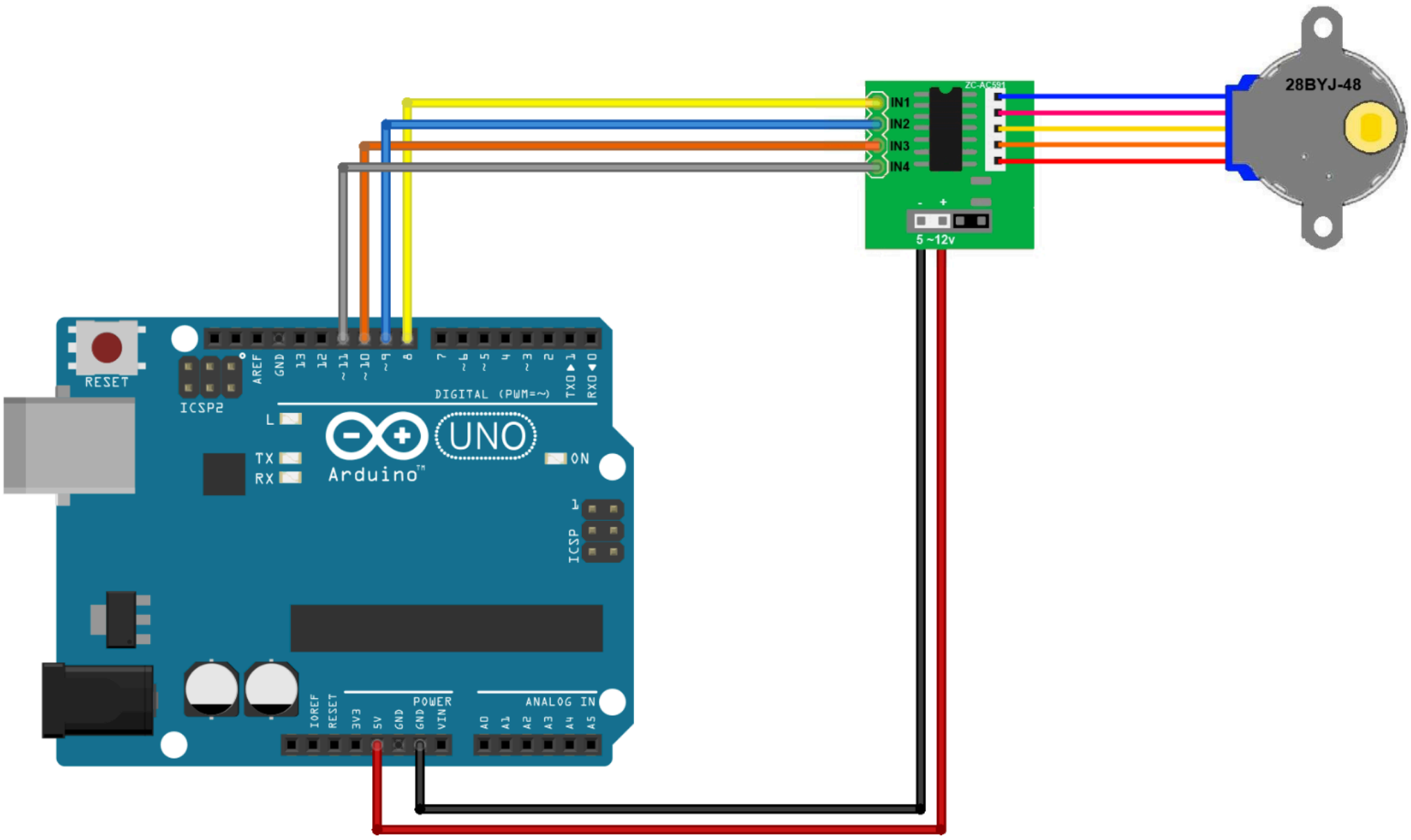
Servo motors and stepper motors are both controllable by an Arduino. Both servo motors and stepper motors can be controlled to rotate to the specific angle and move at a specific speed. This next statement is general in nature and have some exceptions:

Servo motors can rotate from 0 to 180 degrees, the “know” where they are thanks to a sensor (potentiometer) that sends a feedback signal to confirm that angle. This means that a servo motor can increase the amount of power it uses to hold a certain angle. This means that the harder you push a servo motor the more it will push back until it breaks a gear and totally fails. This is very different from a stepper motor with unlimited 360 degree rotation that has no feedback circuit. This means that a servo motor can be overcome with too much input torque (load on the motor), but when it fails to hold position it will simply give in and rotate in the direction of the greater force without permanent damage.

A stepper motor would made a great clock if you add hands to it, but a servo would work great to waive a flag back and fourth.

Do you find it interesting that the further we get into this book with more complex engineering tools that the names of this tools get worse and worse? How about calling this chapter 28BYJ-48 + ULN2003? Yeah, I know, terrible names! I don’t know why is things get more more technical the names get worse and worse. I fully expect you the reader to catch up with me, but at the moment of walk down this path further than you - trust me the names keep getting worse.

The ULN2003 is, you guessed it, another $1 component. This is the control board (green below) the translates the signals sent from the Arduino and amplifies them to the point that they're strong enough to energize electromagnets inside the 28BYJ-48 stepper motor (which also costs a whopping $1 on Amazon). We will be sending signals from Arduino to the control board and the board will send those to the stepper motor.

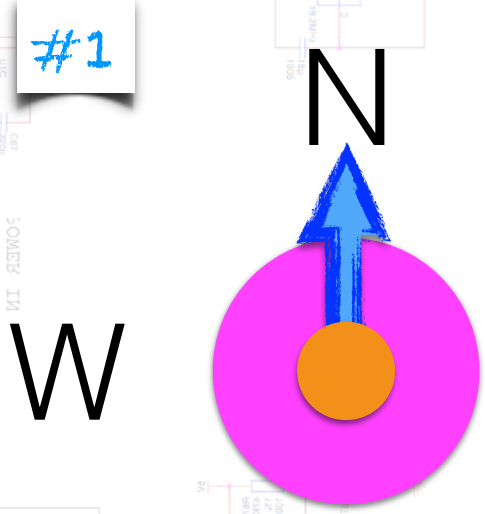


How Stepper Motors Work

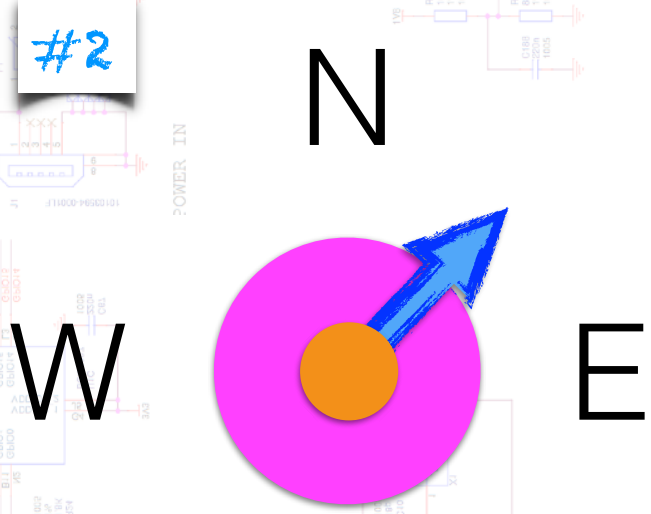
Inside the stepper motor are coils that form different electromagnets. Look at the diagram below and picture someone holding a magnet at each of the north south east and west positions. Imagine that the blue needle it Is free to pivot around the orange circle in the middle it is attracted to the strongest magnet. In case #1 we have a situation where the person is standing north with the strong electromagnet and the three people standing west, south and east have their magnets turned off. In that case we have a needle pointing due north. In case #2 the person standing in the north has NOT turned off the magnet, but the person standing to the east has turned on their magnets at the same strength as a person in the north. This results in a tug-of-war between the north and the east and and the needle ends up exactly between the two at 45° off of north. In case #3 the person in the north is still holding their magnet at full strength, but the person in the east has turned magnet down to half power. In this case the needle lands at 22.5° (which is half of 45°). Can you look at case #4 and decide the configuration of the magnets in the north, east, west and south?

As long as you understand that we're going to have for signals coming in to the motor control board off of the Arduino and then those signals will be amplified and sent into the stepper motor we are ready to rock'n roll. As usual their are caveats to what I'm saying with the intent of getting you up and running quickly

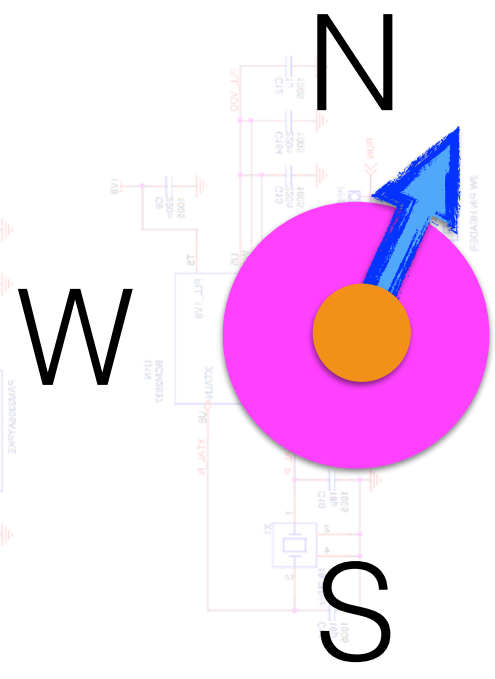
#1



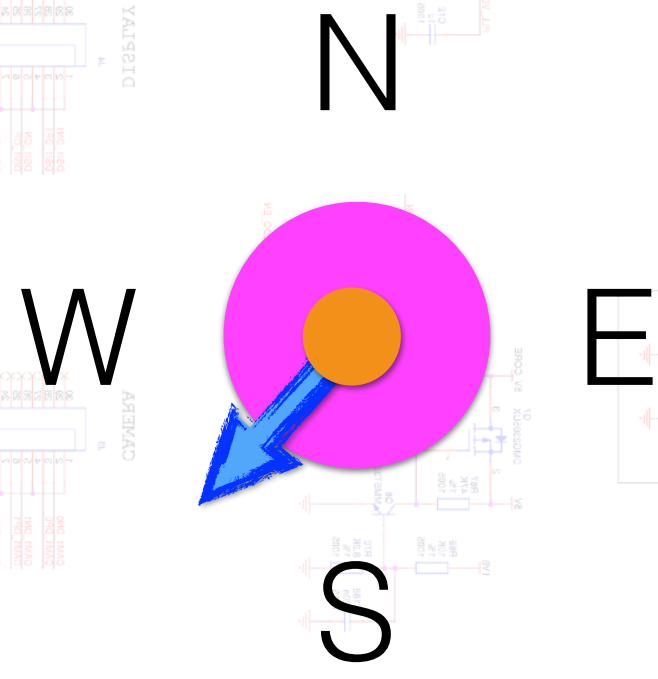
#2



#3



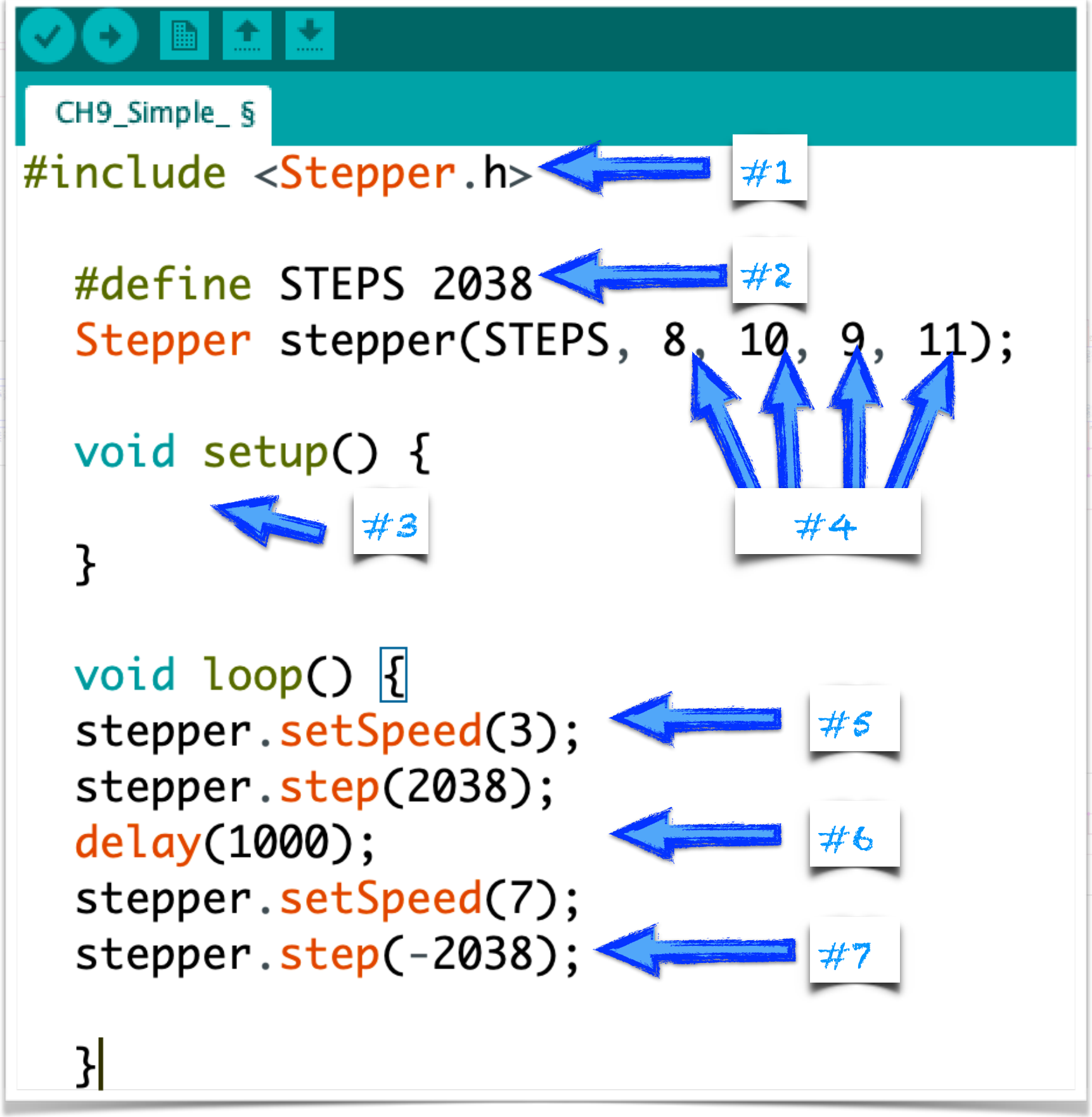
#4



Rotating

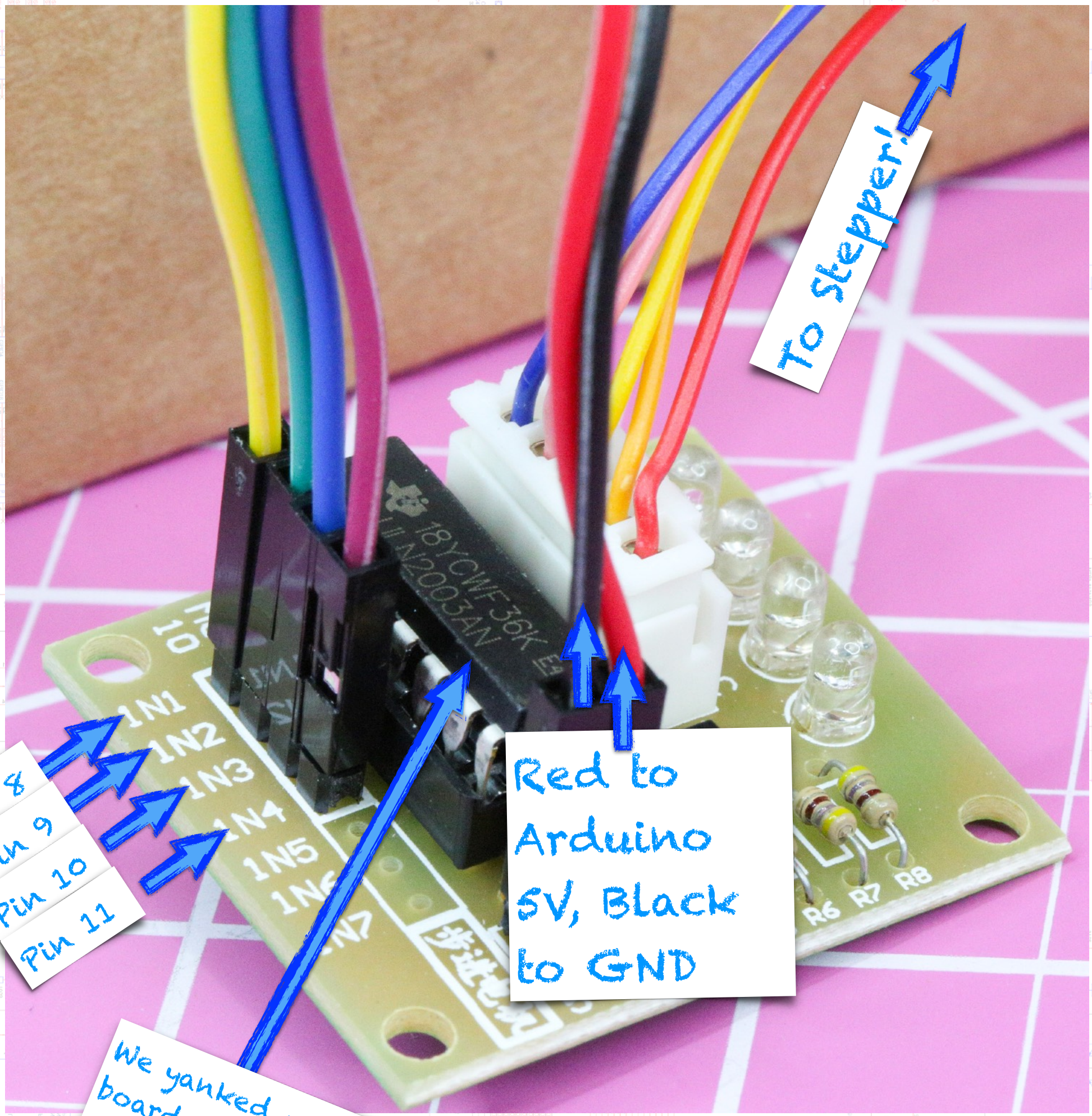
Below is a simple program that will allow you to get started with the stepper motor as quickly as possible. You can see that there is a library included in this sketch which means that we'll need to make sure the library it's installed and accessible to the Arduino IDE. My suggestion this time is it you simply build and run the program and then try to verify it without taking the time to try to install a library. the reason for this is the newer versions of the Arduino IDE are including more and more libraries by default. <Stepper.h> is an increasingly common library and my guess is he a reasonable chance of finding it already installed on your current version of Arduino IDE. If that's not the case simply follow exactly the same steps that you used before with the servomotor to install the <Stepper.h> library.

Looking at the code below you can see that #1) the library that we're calling on. #2) This is an interesting one: as long as those people standing on the north south east and west turn on and off there magnets in the correct order the needle will rotate around the circle. But how do we know how many pulses sent to the stepper motor equals one revolution of the motor? #2) is a place in the code where we tell the Arduino that it takes 2038 pulses to equal one rotation of the motor. This number changes quite a bit with different stepper motor designs. This is a key number to have in order to make your code get the expected results with your project. #3) we don't need to put anything in the setup area in this case. #4) these are the pin numbers that we are going to be using to send signals from the Arduino to the motor control board. Order matters here. #5) Here we set the speed in revolutions per minute. #6) Wait a second before reversing direction. #7) Go back in the opposite direction one rotation.



Making a Crazy Clock

Below is the motor control board with wires on the left coming from the Arduino and wires on the right going to the Stepper motor. Wire IN1 to pin 8 on Arduino, IN2 to pin 9, In3 to pin 10 and IN4 to pin 11.



Ch 9 Stepper Motors

A close up of the reverse view coming from the Arduino to the motor control board. The same case with IN1 to pin 8 on Arduino, IN2 to pin 9, In3 to pin 10 and IN4 to pin 11.