



\$0.00 cad

Search products

[Forum \(/forum\)](#)[Wishlist \(0\) \(/apps/iwish\)](#)[Register \(/account/register\)](#) Or [Sign in \(/account/login\)](#)
[Home \(/\)](#) › [Tutorials \(/Blogs/Tutorials\)](#) › [How to use a Speed Sensor with Arduino](#)

[Got Questions? Click here and post them in the Brainy-Bits Forum to get help! \(/forum\)](#)

OVERVIEW

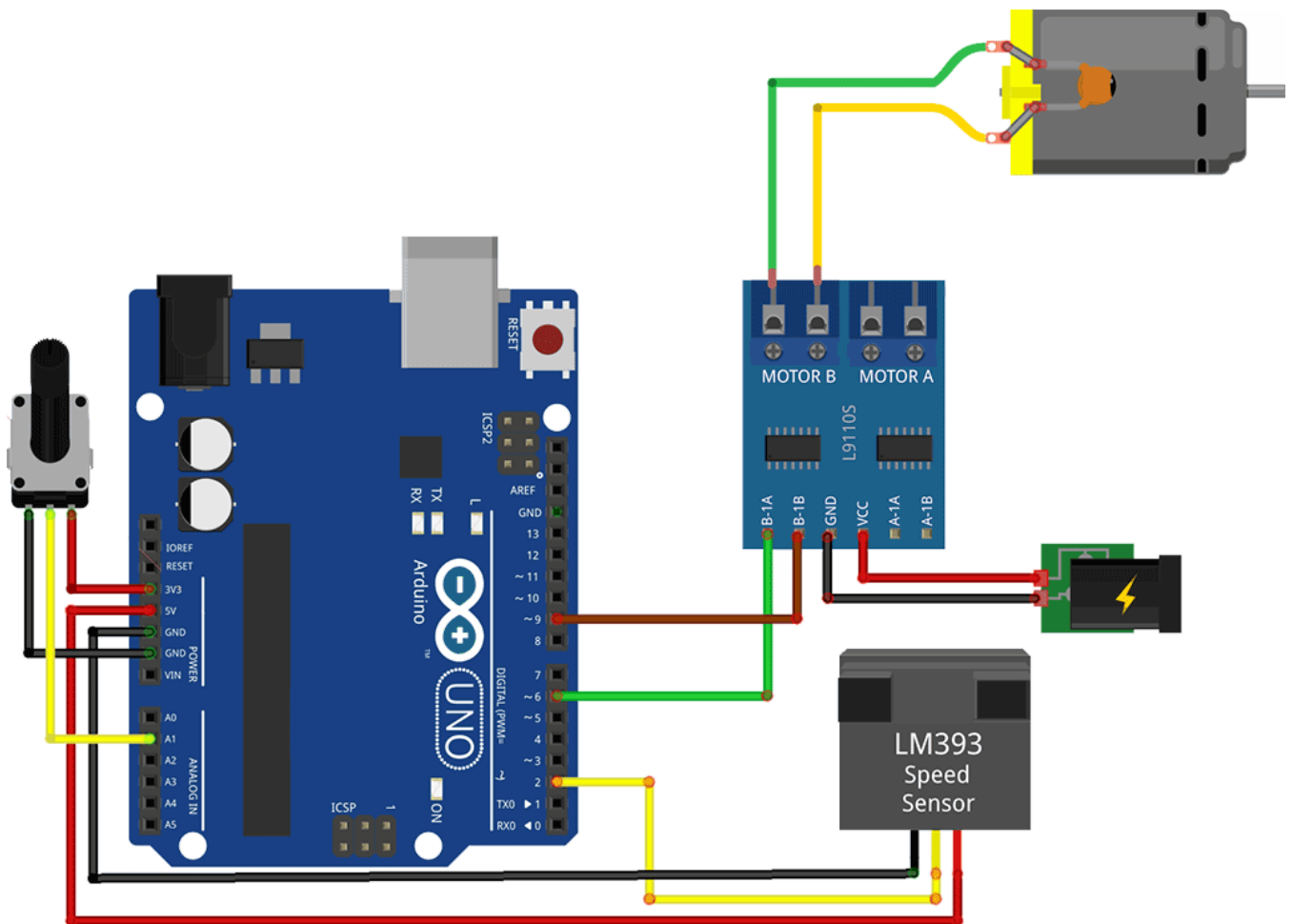
Whether you're building a small robot or using any kind of motors in your projects, it might be useful to know the speed at which you are driving them.

In this tutorial we will see how to connect and use an infrared speed sensor based on the LM393 chip.

The speed sensor uses a disc with holes (encoder disc) to block the infrared beam, thus by counting the number of times the sensors goes from Low to High we can calculate the number of revolution for a given time period.

For our tutorial we will count the number of time the speed sensor goes from Low to High in a second and then divide that number by 20 (number of holes in the encoder disc) to get the number of revolution per second.

SCHEMATIC



fritzing

The speed sensor uses only 1 pin that goes from Low to High to detect holes in the encoder disc.

Since we will be using Interrupt zero to read the speed sensor, we need to connect it to Pin 2 (interrupt 0 pin) on the UNO.

We are also using the L9110 motor driver to control the speed and direction of the geared DC motor.

Pin 2 is connected to the trigger pin of the speed sensor.

5V and GND is connected to the UNO

We connect the potentiometer to 3.3V and GND and use the Analog Pin 1 (A1) to read the value.

Pin 6 and Pin 9 are connected to B-1A and B-1B of the L9110

Power and GND of the L9110 are connected to an external 5V 2A power supply since DC motors can require more current than the UNO can provide.

THE CODE

We will use interrupts and the TimerOne library in our code.

The TimerOne will be used to wait 1 second before displaying the speed of our motor in the serial monitor window.

Interrupts are used so that we can control the speed and direction of our motor at the same time that we read the trigger pin of the speed sensor and to display information in the serial monitor.

So every time the speed sensor pin goes High, interrupt zero will be called since it's connected to Pin 2 of the UNO (which is the interrupt 0 pin), thus increasing the counter variable by 1.

Then when Timer1 reaches 1 second it will call another interrupt to display the results, and reset everything back to zero to start over again.

As you can see in the code, we are using the counter value after 1 second elapsed and dividing it by 20 since our encoder disc has 20 holes. Thus giving us the number of revolution per second.

As always, please check out the tutorial video to have more information.

```
#include
unsigned int counter=0;

int b1a = 6; // L9110 B-1A
int b1b = 9; // L9110 B-1B

void docount() // counts from the speed sensor
{
  counter++; // increase +1 the counter value
}

void timerIsr()
{
  Timer1.detachInterrupt(); //stop the timer
  Serial.print("Motor Speed: ");
  int rotation = (counter / 20); // divide by number of holes in Disc
  Serial.print(rotation,DEC);
  Serial.println(" Rotation per seconds");
  counter=0; // reset counter to zero
  Timer1.attachInterrupt( timerIsr ); //enable the timer
}

void setup()
{
  Serial.begin(9600);

  pinMode(b1a, OUTPUT);
  pinMode(b1b, OUTPUT);

  Timer1.initialize(1000000); // set timer for 1sec
  attachInterrupt(0, docount, RISING); // increase counter when speed sensor pin goes High
  Timer1.attachInterrupt( timerIsr ); // enable the timer
}

void loop()
{
  int potvalue = analogRead(1); // Potentiometer connected to Pin A1
  int motorspeed = map(potvalue, 0, 680, 255, 0);
  analogWrite(b1a, motorspeed); // set speed of motor (0-255)
  digitalWrite(b1b, 1); // set rotation of motor to Clockwise
}
```