



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





12V DC Motor 251rpm w/Encoder (SKU: FIT0186)



Contents

- [1 Introduction](#)
- [2 Specification](#)
- [3 Encoder Diagram](#)
- [4 Encoder Sample Code](#)

Introduction

This is a Gear Motor w/Encoder, model No.GB37Y3530-12V-251R. It is a powerful 12V motor with a 43.7:1 metal gearbox and an integrated quadrature encoder that provides a resolution of 64 counts per revolution of the motor shaft, which corresponds to 2797 counts per revolution of the gearbox's output shaft. These units have a 0.61" long, 6 mm-diameter D- shaped output shaft. This motor is intended for use at 12V, though the motor can begin rotating at voltages as low as 1V. The face plate has six mounting holes evenly spaced around the outer edge threaded for M3 screws. These mounting holes form a regular hexagon and the centers of neighboring holes are 15.5 mm apart. This motor is an ideal option for your mobile robot project.

Warning: Do not screw too far into the mounting holes as the screws can hit the gears. Manufacturer recommends screwing no further than 3mm (1/8") into the screw hole.

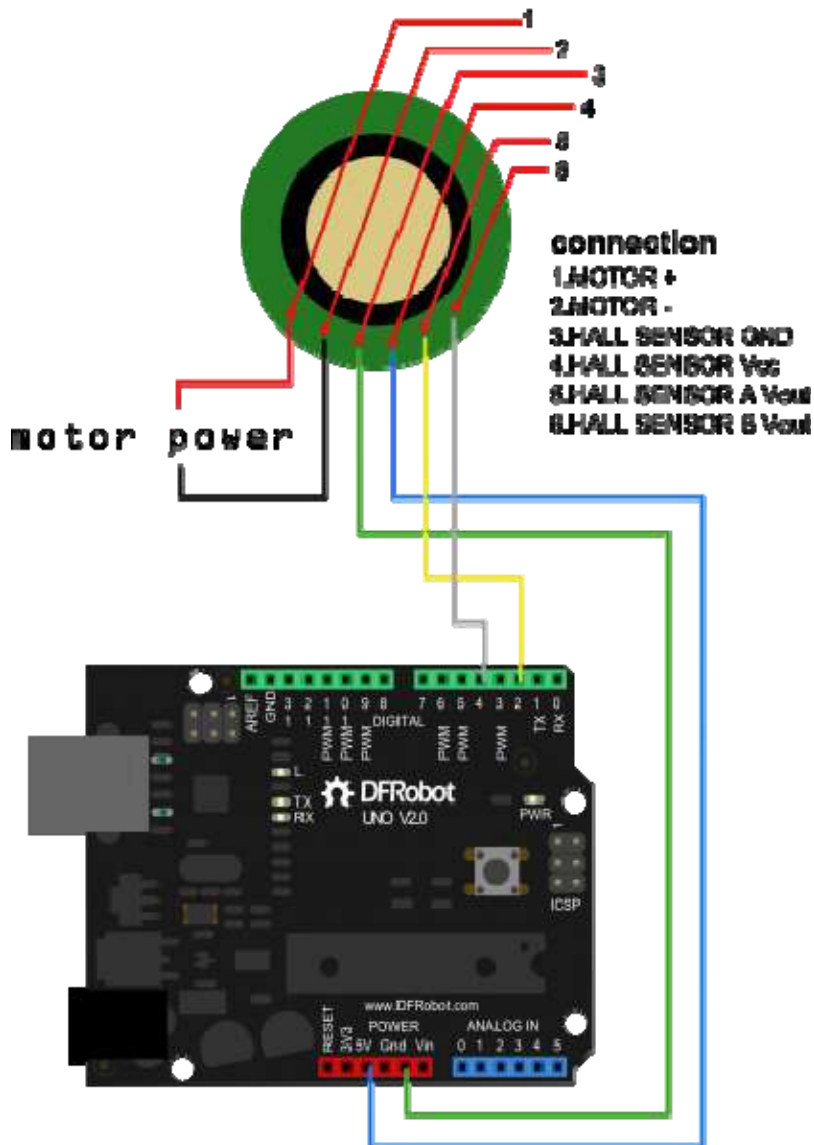
Specification

- Gear ratio: 43.8:1
- No-load speed: 251 + 10% RPM
- No-load current: 350 mA
- Start Voltage: 1.0 V
- Stall Torque: 18 Kg.com
- Stall Current: 7 A
- Insulation resistance: 20 M Ω
- Encoder Operating Voltage: 5 V
- Encoder type: Hall
- Encoder Resolution: 16CPR(motor shaft)/700CPR(gearbox shaft)

- Weight: 205g

Encoder Diagram

====Diagram for UNO====



====Interrupt Port with Different Board====

Notcie:attachInterrupt()

Board	int.0	int.1	int.2	int.3	int.4	int.5
Uno, Ethernet	2	3				
Mega2560	2	3	21	20	19	18
Leonardo	3	2	0	1	7	

For example, with UNO board, you want to use interrupt port 0(int.0). You should connect digital pin 2 with the board. So, the following code is only used in UNO and Mega2560. If you want to use Leonardo or Romeo, you should change digital pin 3 instead of digital pin 2.

See the link for detail <http://arduino.cc/en/Reference/AttachInterrupt>

Encoder Sample Code

```
//The sample code for driving one way motor encoder
const byte encoder0pinA = 2;//A pin -> the interrupt pin 0
const byte encoder0pinB = 4;//B pin -> the digital pin 4
byte encoder0PinALast;
int duration;//the number of the pulses
boolean Direction;//the rotation direction

void setup()
{
  Serial.begin(57600);//Initialize the serial port
  EncoderInit();//Initialize the module
}

void loop()
{
  Serial.print("Pulse:");
  Serial.println(duration);
  duration = 0;
  delay(100);
}

void EncoderInit()
```

```

{
  Direction = true;//default -> Forward
  pinMode(encoder0pinB, INPUT);
  attachInterrupt(0, wheelSpeed, CHANGE);
}

void wheelSpeed()
{
  int Lstate = digitalRead(encoder0pinA);
  if((encoder0PinALast == LOW) && Lstate==HIGH)
  {
    int val = digitalRead(encoder0pinB);
    if(val == LOW && Direction)
    {
      Direction = false; //Reverse
    }
    else if(val == HIGH && !Direction)
    {
      Direction = true; //Forward
    }
  }
  encoder0PinALast = Lstate;

  if(!Direction) duration++;
  else duration--;
}

```