

Introduction

(<https://www.dfrobot.com/product-1619.html>)

These are three DC motors with their own decelerator and encoder, with a drive voltage of 6V.

The motors come with a Hall-type encoder with a resolution of 11x deceleration ratio and a D-shaped output shaft.

It is a 6V motor with a 34:1 metal gearbox and an integrated quadrature encoder that provides a resolution of 11 counts per revolution of the motor shaft, which corresponds to 341.2 counts per revolution of the gearbox's output shaft.

The motors are ideal choices for a variety of mobile robot projects and medium-sized mobile platforms.



Specification

FIT0520

- Motor Rated Voltage: 6V
- Encoder Rated Voltage: 3.3 / 5V
- Reducer Reduction Ratio: 1: 20
- No load Speed: 300RPM @ 0.1A
- Maximum Efficiency Point: load 0.7kg•cm/245RPM/1.2W/0.4A
- Maximum Power Point: load 1.8kg•cm/160RPM/2W/0.8A
- Stall Torque: 3.6kg•cm
- Stall Current: 2.7A
- Hall Resolution: Hall Resolution 11x Precision Reduction Ratio 20.4 = 224.4 PPR
- Dimension: 50 * Φ 24.4 mm / 1.97 * Φ 0.96inches

- Weight: 96g

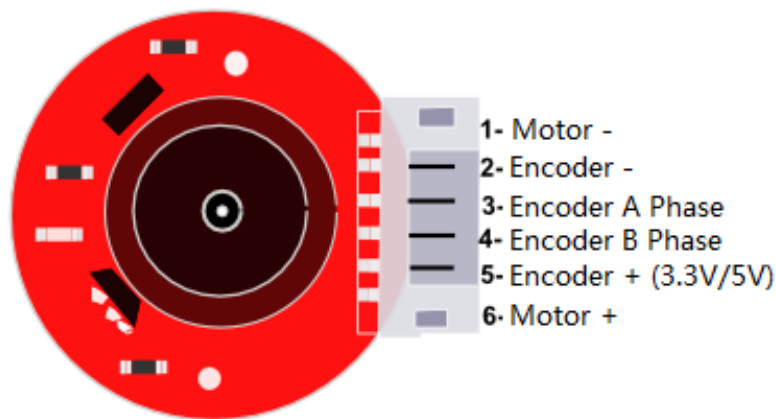
FIT0521

- Motor Rated Voltage: 6V
- Encoder Rated Voltage: 3.3 / 5V
- Reducer Reduction Ratio: 1: 34
- No load Speed: 210RPM @ 0.13A
- Maximum Efficiency Point: load 2.0kg•cm/170RPM/2.0W/0.6A
- Maximum Power Point: load 5.2kg•cm/110RPM/3.1W/1.1A
- Stall Torque: 10kg•cm
- Stall Current: 3.2A
- Hall Resolution: Hall Resolution 11x Precision Reduction Ratio 34.02=341.2 PPR
- Dimension: 52 * Φ 24.4 mm / 2.05 * Φ 0.96inches
- Weight: 96g

FIT0522

- Motor Rated Voltage: 6V
- Encoder Rated Voltage: 3.3 / 5V
- Reducer Reduction Ratio: 1: 75
- No load Speed: 100RPM 0.13A
- Maximum Efficiency Point: load 1.0kg•cm/80RPM/1.7W/0.5A
- Maximum Power Point: load 3.0kg•cm/55RPM/2.8W/1.0A
- Stall Torque: 6.5kg•cm
- Stall Current: 3.0A
- Hall Resolution: Hall Resolution 11x Precision Reduction Ratio 74.83=823.1PPR
- Dimension: 54 * Φ 24.4 mm / 2.13 * Φ 0.96inches
- Weight: 96g

Board Overview



Num	Label	Description
1	GND of motor power supply	GND
2	GND of encoder power supply	GND
3	Encoder A phase	Pulse A
4	Encoder B phase	Pulse B
5	VCC of encoder power supply	3.3V/5.0V
6	VCC of motor power supply	6V

Tutorial

There are two sample codes below, code 1 can be used to validate the motor or to understand how the motor works, and code 2 can be used as a basis for applications such as smart cars.

Requirements

- **Hardware**
 - DFRduino Romeo-All in one Controller (<https://www.dfrobot.com/product-656.html>) (or similar) x 1
 - 6V Power Supply x 1

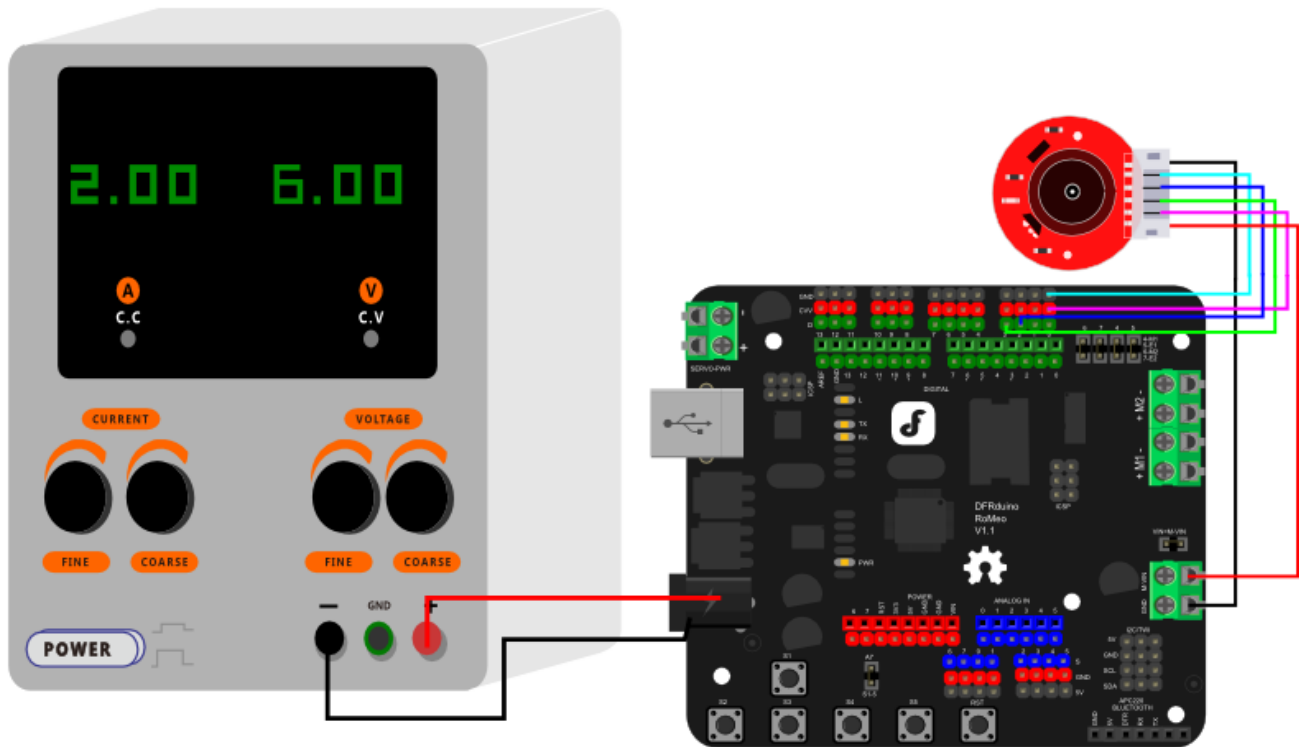
- Jumper wires

- Software

- Arduino IDE (<https://www.arduino.cc/en/Main/Software>)

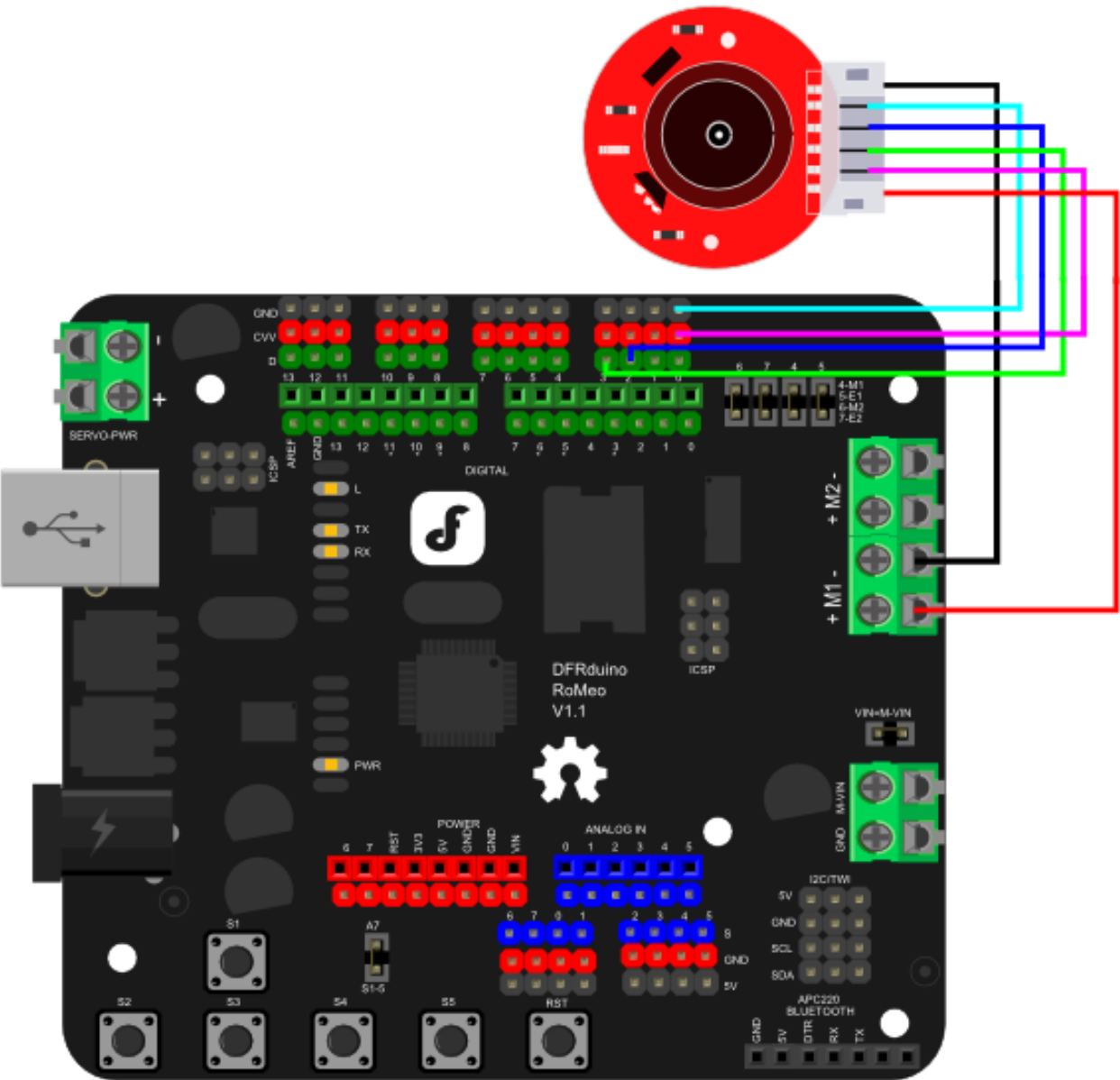
Connection Diagram

Motor 6V power supply wiring(correspond to example1)



Motor PID Wiring (correspond to example2)

Just connect the positive and the negative to M1 and remain other parts as the same, and download example2.



In this tutorial, D2 and D3 are selected, D2 is the interrupt pin and D3 is the signal input pin. In practice, we just need to ensure one interrupt pin, and the other pin can be self-defined (see master of the interrupt pin in the figure below)

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, if you want to use interrupt pin 0 (interrupt 0) on UNO, you can connect the https://wiki.dfrobot.com/Metal_DC_Geared_Motor_wEncoder_SKU:_FIT0520,FIT0521,_FIT0522

For example, if you want to use interrupt pin 0 (interrupt 0), on Uno, you can connect the number 2 digital pin (D2); With Interrupt Pin 1 (Interrupt 1), you can connect number 3 digital pin (D3).

Please check the link for more details via the
link:<http://arduino.cc/en/Reference/AttachInterrupt>
(<http://arduino.cc/en/Reference/AttachInterrupt>)

Sample Code1

```
//The sample code for driving one way motor encoder
const byte encoder0pinA = 2;//A pin -> the interrupt pin 0
const byte encoder0pinB = 3;//B pin -> the digital pin 3
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--;
}

```

Expected Result 1

Data is constantly output from the serial port. When the motor turns clockwise, the output value > 0 ; when the motor turns counterclockwise, the output value < 0 . The faster the motor, the greater the absolute value of the number, otherwise the lower the motor, the smaller the absolute value of the number.

Pulse:-118	Pulse:117
Pulse:-119	Pulse:117
Pulse:-119	Pulse:118
Pulse:-119	Pulse:116
Pulse:-119	Pulse:116
Pulse:-119	Pulse:117
Pulse:-120	Pulse:116
Pulse:-119	Pulse:116
Pulse:-120	Pulse:117
Pulse:-118	Pulse:117
Pulse:-119	Pulse:117
Pulse:-118	Pulse:117
Pulse:-118	Pulse:118
Pulse:-119	Pulse:117
Pulse:-118	Pulse:117
Pulse:-119	Pulse:118
Pulse:-119	Pulse:118
Pulse:-119	Pulse:117
Pulse:-120	Pulse:118
Pulse:-120	Pulse:116
Pulse:-120	Pulse:116
Pulse:-120	Pulse:116
Pulse:-120	Pulse:116
Pulse:-120	Pulse:116
Pulse:-119	Pulse:116

Sample Code2 (PID Control)

Drive the DC motor via the L298P DC motor drive board. The PID algorithm controls the

speed of the motor.

1. Connect the motor power port to the L298 motor drive M1 port.
2. Download and install the Arduino PID Library
(<https://www.dfrobot.com.cn/images/upload/File/FIT0450/20160504134708hpjt1e.zip>)
(About how to install the library
(<https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0>)?)

```

//The sample code for driving one way motor encoder
#include <PID_v1.h>
const byte encoder0pinA = 2;//A pin -> The interrupt pin 0
const byte encoder0pinB = 3;//B pin -> The digital pin 3
int E_left =5; //Connect the enabling port of the L298P DC motor drive shield to digital pin 5
int M_left =4; //Connect the turning port of the L298P DC motor drive shield to digital pin 4
byte encoder0PinALast;
double duration,abs_duration;//The number of the pulses
boolean Direction;//The rotation direction
boolean result;

double val_output;//Provide the PWM output value to motor
double Setpoint;
double Kp=0.6, Ki=5, Kd=0;
PID myPID(&abs_duration, &val_output, &Setpoint, Kp, Ki, Kd, DIRECT);

void setup()
{
    Serial.begin(9600);//Initialize the serial port
    pinMode(M_left, OUTPUT); //Set the control port of the L298P DC motor drive shield to digital pin 4
    pinMode(E_left, OUTPUT);
    Setpoint =80; //Set PID output value
    myPID.SetMode(AUTOMATIC);//Set PID to automatic mode
    myPID.SetSampleTime(100);//Set the PID sampling frequency to 100ms
    EncoderInit();//Initialize the module
}

void loop()
{
    advance();//Motor rotates clockwise
    abs_duration=abs(duration);
    result=myPID.Compute();//The return value is 1 after PID conversion
    if(result)
    {
        Serial.print("Pluse: ");
        Serial.println(duration);
        duration = 0; //Clear count value and waits for the next count
    }

}

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--;

}

void advance()//Motor rotates clockwise
{
    digitalWrite(M_left,LOW);
    analogWrite(E_left,val_output);
}

void back()//Motor rotates counterclockwise
{
    digitalWrite(M_left,HIGH);
    analogWrite(E_left,val_output);
}

void Stop()//Motor stops
{
    digitalWrite(E_left, LOW);
}

```

Expected Result 2

Because the program sets a PID value of 80, the stable speed of the motor is around 80. When external forces (such as the driving voltage change of the motor, the resistance of the motor increased, etc.) force the speed to change, the program adjust the speed to about 80 through the PWM. For example, increase the motor drive voltage, the motor speed will briefly rise to 80. Reduce the motor drive voltage the motor speed will temporarily drop

When rise to 60. Reduce the motor drive voltage, the motor speed will temporarily drop, then rise to 80.

More Documents



Get Metal DC Geared Motor w/Encoder - 6V 300RPM 3.6Kg.cm

(<https://www.dfrobot.com/search-FIT0520.html>) from DFRobot Store or **DFRobot Distributor**.

(<https://www.dfrobot.com/index.php?route=information/distributorslogo>)

Turn to the Top