

search

- DFRobot Products (DFR) Index >
- Sensor (SEN) Index >
- Telecom Module (TEL) Index >
- Motor Driver (DRI) Index >
- Fitting (FIT) Index >
- Robot (ROB) Index >
- micro:bit (MBT) Index >
- Kit (KIT) Index >
- Toy (TOY) Index >
- Servos (SER) Index >
- What is XX series >
- How to select series >
- How it works series >
- PinPong Library >

## Product Introduction

- Features
- Application
- Specification
- Dimension (Unit: mm)
- Tutorial

## Product Introduction

Motor control has never been so simple! The N20 miniature metal motor series is an indispensable tool for your project. Compared to traditional motors, we adopt the PWM servo control method, fundamentally simplifying wiring and programming difficulties.

Conventional motors typically require the additional purchase of a drive module, increasing project costs. Moreover, they face complex and cumbersome wiring and programming tasks because conventional motors usually do not come with a built-in drive chip, requiring connection to at least 4 ports for normal control. Our product cleverly integrates the drive chip at the tail of the motor and adopts the control method of a servo, requiring only one PWM port to easily control the motor's speed and direction. This not only eliminates wiring troubles but also makes programming a breeze.

In addition, we have deeply optimized performance, especially by increasing the stop point pulse width, successfully reducing standby power consumption to less than 1mA, with minimal current consumption when no control signal is present.

To cater to different application scenarios, we offer a variety of gear ratios and output speeds for motor selection. The table below provides detailed performance parameters for each motor model, making your selection more convenient.

### Integrated Drive N20 Series Motor Parameter Table

SKU	DFR1114	DFR0399	DFR0429	DFR0429
Motor and Gearbox Parameters				
Motor Speed (Unreduced)	20000	220000	220000	220000
Gearbox Reduction Ratio	1:150	1:75	1:50	1:30
No-load Output Speed	133	290	440	730
Stall Torque	1.0kg*cm	0.8kg*cm	0.5kg*cm	0.3kg*cm
Power Supply Parameters				
Operating Voltage	3-6V	3-6V	3-6V	3-6V
Rated Voltage	6V			
No-load Current	50mA	70mA	70mA	70mA
Stall Current	640mA	1000mA	1000mA	1000mA
Static Current	<1mA			
Control Parameters				
PWM Frequency	500Hz	500Hz	500Hz	500Hz
Signal Resolution	1us	1us	1us	1us
Forward Pulse Range	500-1400us	500-1400us	500-1400us	500-1400us
Stop Pulse Width Range	1400-1600us	1400-1600us	1400-1600us	1400-1600us
Reverse Pulse Range	1600us-2500us	1600us-2500us	1600us-2500us	1600us-2500us
Other Parameters				
Weight	10g	10g	10g	10g
Dimensions(mm)	L40mm x W12mm x H20			

## Features

- All-metal N20 motor, compact in size, and reliable in quality.
- Integrated 360-degree servo motor drive mode for ease of use.
- PH2.0-3P interface, Gravity line sequence standard, convenient for connection and insertion.

## Application

- Robot
- Mechanical motion control
- Valve control
- Door lock control

## Specification

### Power Parameters

- Operating Voltage Range: 3V-6V
- Rated Voltage: 6V
- No-load Current: 50mA (under 6V)
- Static Current: <1mA (no PWM control signal input)
- Stall Current: 640mA

### Motor Parameters

- Motor Speed (Unreduced): 20,000 RPM
- Gearbox Reduction Ratio: 150:1
- No-load Output Speed: 133 RPM
- Stall Torque: 1.0 kg\*cm

### Control Parameters

- PPM Signal Resolution: 1μs

- PPM Signal Positive Pulse Width Range: 500μs-2500μs

- Forward Pulse Width Range: 500μs-1400μs (maximum speed at 500μs)

- Stop Pulse Width Range: 1400μs-1600μs

- Reverse Pulse Width Range: 1600μs-2500μs (maximum speed at 2500μs)

- PWM Drive Frequency: 500 Hz

### Other Parameters

- Weight: 10g

- Dimensions: L40mm x W12mm x H20mm

## Dimension (Unit: mm)



## Tutorial

### Requirements

- Hardware
  - DFRduino UNO x1
  - DFR1114 Integrated Drive N20 Motor x1
- Software
  - Arduino IDE: [Click to Download Arduino IDE from Arduino](#)
  - Arduino Servo library: [Arduino Servo library](#) (If Arduino IDE is already built-in, there is no need for additional download.)

## Connection Diagram



### Sample Code

Code Explanation: The following code utilizes a 180-degree servo motor drive method to control the N20 motor. The program functions as follows:

- Pause for 1 second when the program is running.
- Rotate the motor clockwise at the maximum speed for 2 seconds.
- Rotate the motor counterclockwise at the maximum speed for 2 seconds.
- Gradually reduce the clockwise speed to a stop, then vary the counterclockwise speed from slow to fast.

The above logic is in an infinite loop.

```
#include <Servo.h>
#include <DFR1114.h>
```

```
#define speed_maxP 0 //Clockwise rotation (Max speed)
```

```
#define speed_maxN 180 //Anticlockwise rotation (Max speed)
```

```
#define speed_stop 90 //Stop
```

```
Servo mymotor; // create servo object to control a servo (my
```

```
motor can be controlled on mos
```

```
int pos=0;
```

```
void setup()
```

```
{ mymotor.attach(9); //attaches the motor on pin 9 to the ser
```

```
}
void loop()
{
    mymotor.write(speed_stop); //Stop
    delay(1000); //Delay 1s
    mymotor.write(speed_maxP); //Clockwise rotation
    delay(1000); //Delay 1s
    mymotor.write(speed_maxN); //Anticlockwise rotation
    delay(1000); //Delay 1s
    for(pos=0;pos<speed_maxP;pos+=2) //Slow down, change
    {
        mymotor.write(pos);
        delay(50);
    }
}
```

Since every motor has a little difference, we enlarge the stop range to ensure that the motor can be stopped correctly.

### Note

- "0-180" degree corresponds to the pulse width signal "500μs-2500μs".

- When we give a value between 81-99° (1400-1600μs), the motor will stop;

- When we give a value less than 81° (1400μs), the motor rotates clockwise, the smaller the value, the faster the speed, 0° (500μs) is the maximum speed;

- When we give a value greater than 99° (1600μs), the motor rotates anticlockwise, the larger the value, the faster the speed, 180° (2500μs) is the maximum speed;

- Since every motor has a little difference, we enlarge the stop range to ensure that the motor can be stopped correctly.