Internship Report

in German Research Center for Artificial Intelligence (DFKI)

Miaoshun Wu

ESD, Hochschule Bremerhaven

July-October,2019

Catalogue

Arduino Nano board Wireless communication

1. Components

NRF24L01 Transceiver Module[i]

Arduino Nano Board (Save much more space than using Uno board)[ii]

Jump Wires

2. Connection

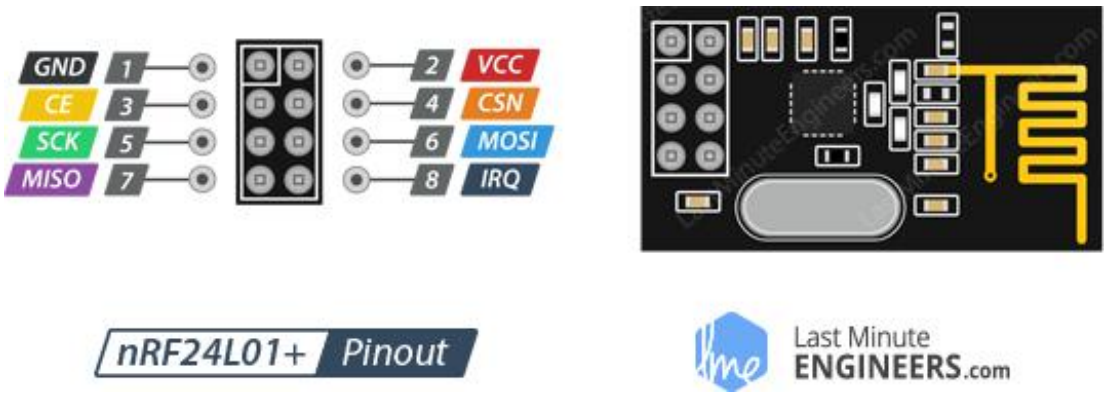


Figure 1. NRF24L01[1]

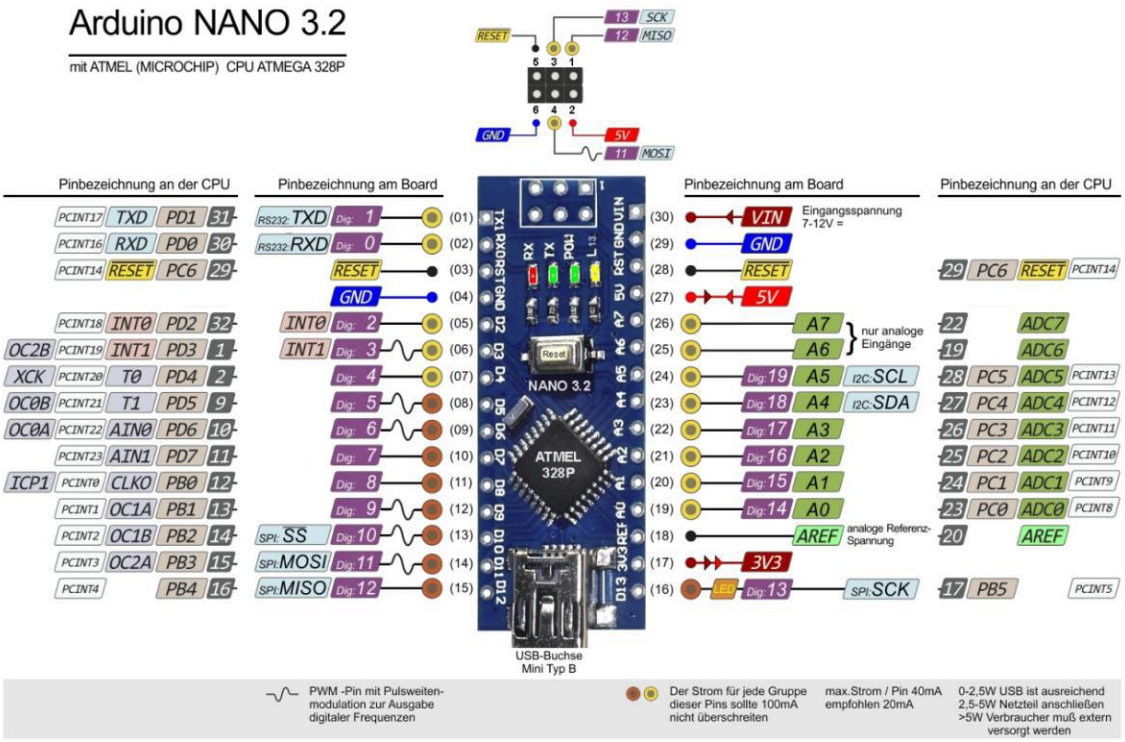


Figure 2. Arduino Nano[2]

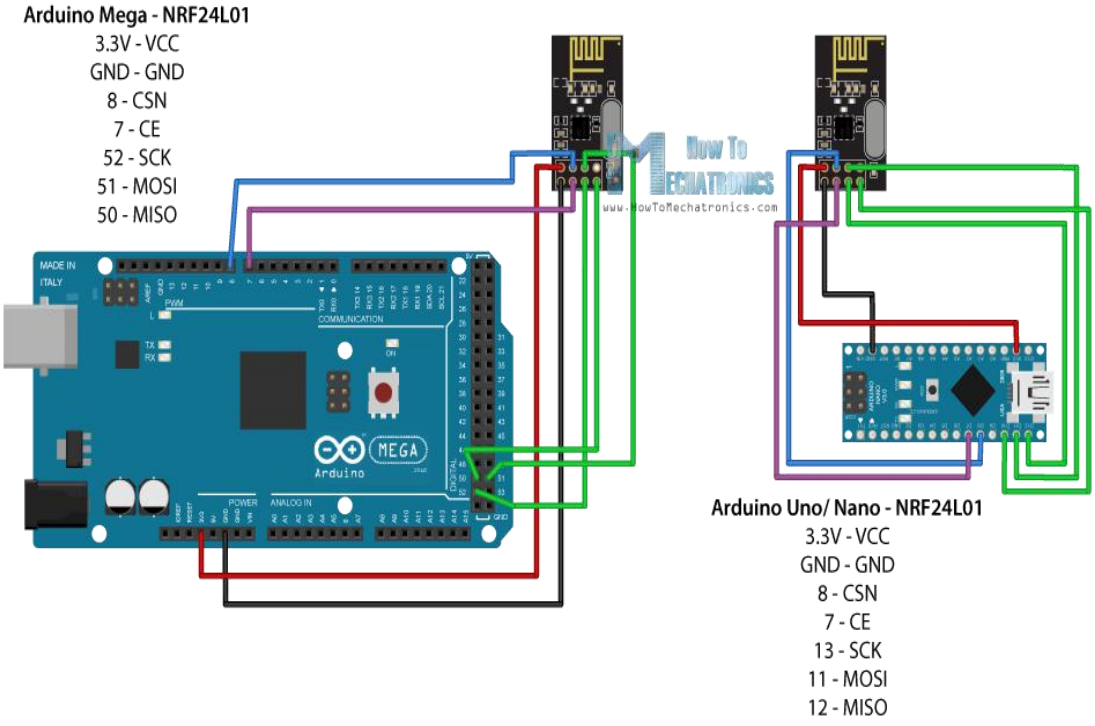


Figure 3. Connection of Arduino Uno/nano-NRF24L01[3]

[i] https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf
[ii] http://avrchip.com/arduino-nano-datasheet-and-tutorial/

[1] https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/
[2] http://goblack.de/arduino/arduino-nachbau.html
[3] https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/

Because the nRF24s uses SPI to communicate with the Arduino they must using Arduino pin 13, 12 and 11 (SCK, MISO and MOSI). It is also necessary to connect the CSN and CE pins and any of the Arduino I/O pins can be used for them. However for demo purposes it seems easiest to use pins 10 and 9 so that all 5 connections are adjacent on pins 13 to 9.

3. One transmitter to one receiver

https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/

From website we can find the basic code for one Transmitter and one Receiver.but we have to notice

1,Between transmitter code and receiver code **the address,the data type** must be the same.

2,Pin connection must be the same as Code as well **(CE,CSN).**

As we using one nano board to control two DC motors,so each time we need to receive two different signals to control these two DC motors.However every receiver can only receiver one signal at each time .Then there are three different ways to transfer one signal to control two DC motors.

Way 1,we use one 8 bit number to divide into two 4 bit numbers like

$$12345678 \Rightarrow 1234,5678$$

but it only works without symbol +,- and the we can not get two 5 bit number neither.So we have to move to Way 2.

Way 2 ,We use strings to get two signed number.like

“ +12345,-67890 ”⇒ +12345, -67890

It works,but it is a little bit complex after we received string and then transferred it into a signed number.

Way 3, We just transfer array of two numbers,like

Data[2]={+12345,-67890} ⇒ Data[0]=+12345,Data[1]=-67890

It’s much easy and simply ,so we choose to use the way 3.

4. One transmitter to more receivers

When one to one wireless communication is done, we have to do one transmitter to more receivers . From the NRF24L01 data-sheet we can know that each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time. the code and the way we can consult the website https://forum.arduino.cc/index.php?topic=421081.

5. One transmitter to more receivers and get the feedback from each motor

We can also consult website above. But notice that we have to write a new address as transmitter receive address.

```
const byte slaveAddress[][6] = {"00011","00022","00033","00044"};
const byte masterAddress[5] = {'T','X','a','a','a'};
```

Figure 4. master and slave address

So we have Master address for receiving all the position feedback from each motor. At this moment,we don't know all the data which received from different motors exactly from which motor. Thus we have to signed data when we give the feedback.

Arduino Nano board Position Control of DC motors

1. Components

- L298N Driver

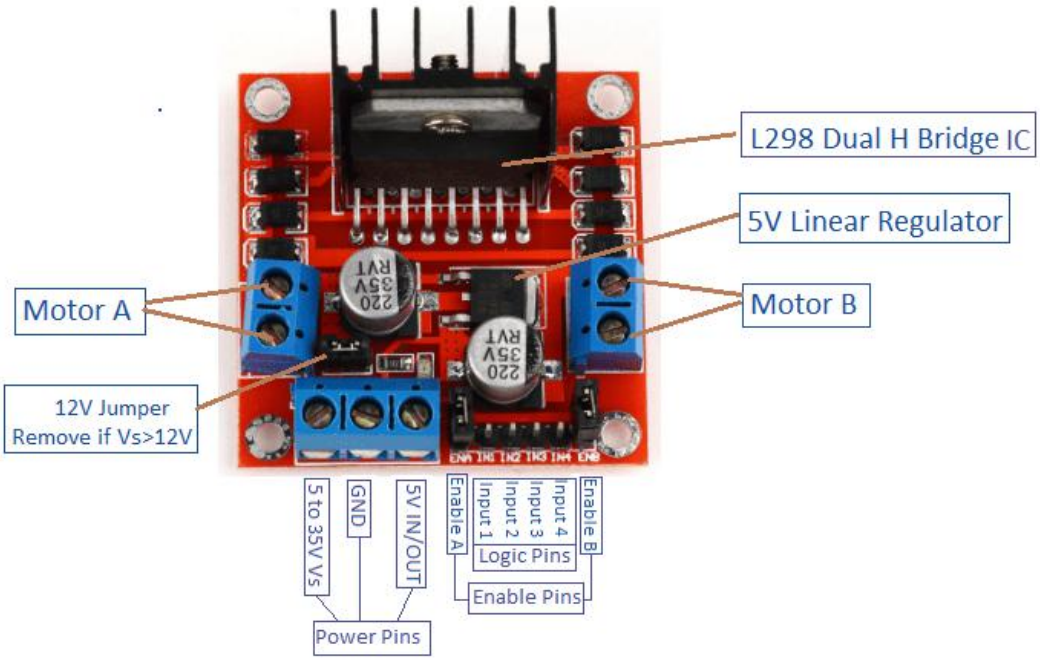
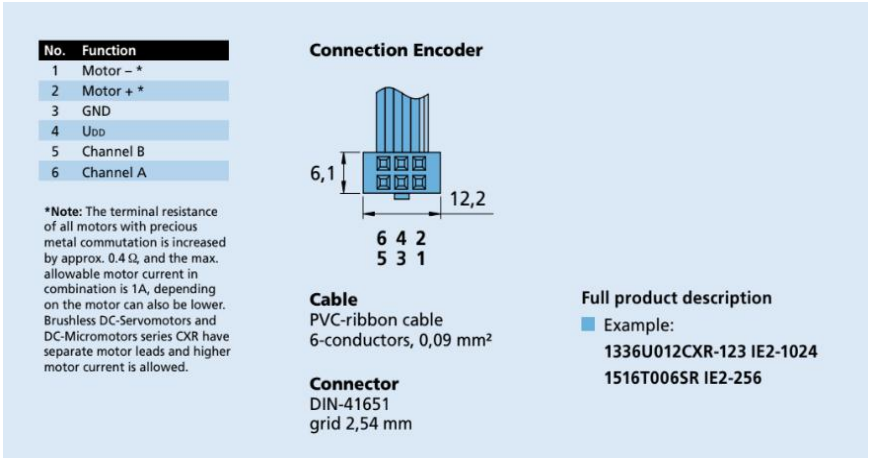


Figure 5. L298N Driver[4]

- DC Motor with encoder

Figure 6. DC Motor with encoder

[4].https://electronicshobbyists.com/controlling-dc-motors-arduino-arduino-l298n-tutorial/



No.	Function
1	Motor – *
2	Motor + *
3	GND
4	U_{DD}
5	Channel B
6	Channel A

*Note: The terminal resistance of all motors with precious metal commutation is increased by approx. 0.4 Ω, and the max. allowable motor current in combination is 1A, depending on the motor can also be lower. Brushless DC-Servomotors and DC-Micromotors series CXR have separate motor leads and higher motor current is allowed.

Figure 7. DC Motor with encoder connection

- Breadboard and Jump Wires
- Arduino Nano board(see Figure 2)

2. Connection

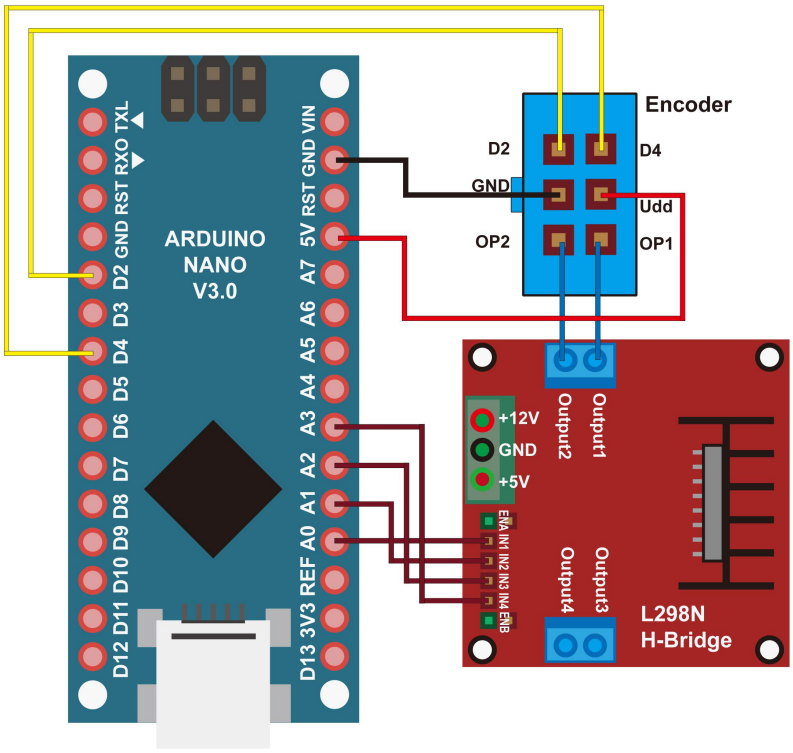


Figure 8. Arduino nano board connect with L298N and DC motor

3.Code

The basic code is from Mr.Yu. And i will put it in the end as Basic code to run the DC motor. You can find all the code from the Motor control folder. But I think this module is not the prefect one because it can only run as the full speed ,we didn't used the PWM EnA Pin to control the Speed of the motor.

You can change the code from Speed control code ,so that the motor will be more smooth when you do the position and speed control.

Arduino Nano board Speed Control of DC motors

We can used the model from above,but in order to control the speed we must connecting the EnA pin ,so we have to use the PWM to control the speed. Then do a closed loop as PID control. The Input of PID control is the Velocity of the motor,from current position subtract previous position to calculate the speed.

Arduino Nano board Position and Speed Control of DC motors

In order to control the Speed ,We can control the input voltage to the motor and the most common method of doing that is by using PWM signal.

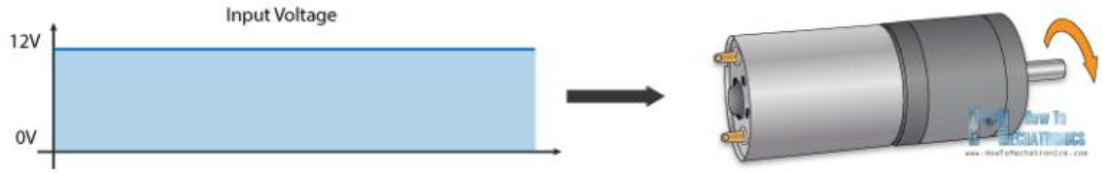


Figure 9. DC-Motor-Speed-Control-Input-Voltage

PWM introduction

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate

voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width.

In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to analogWrite() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time) for example.

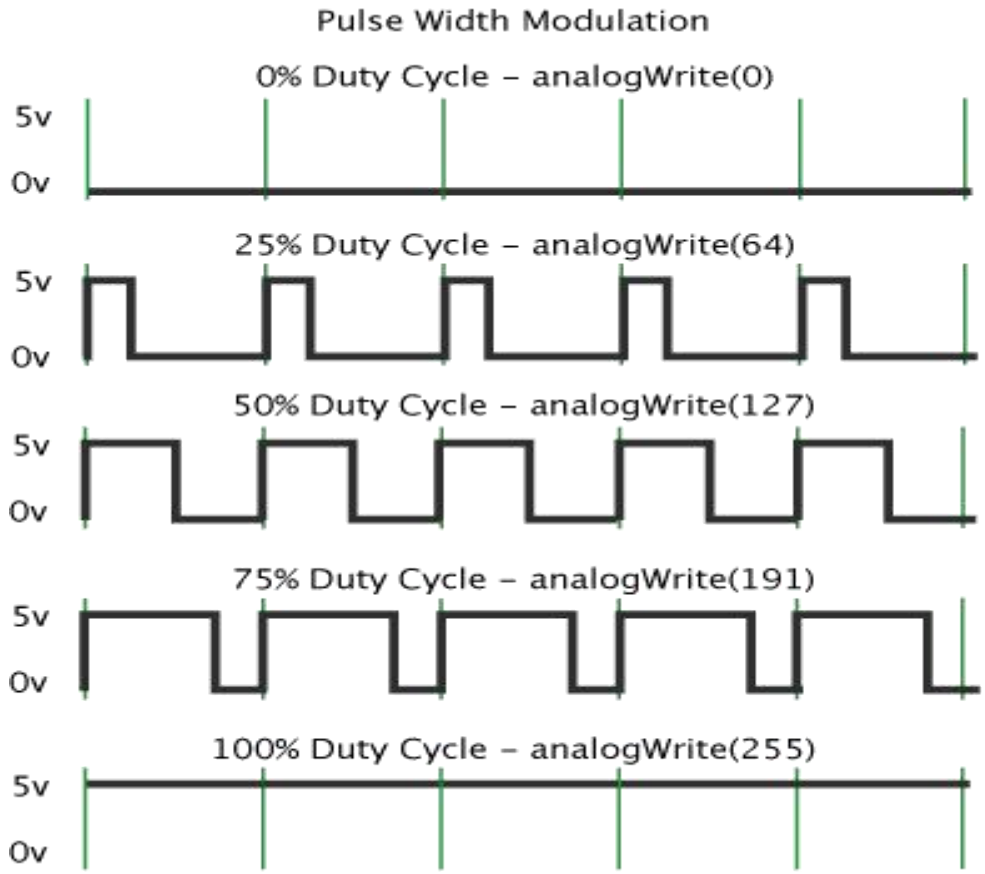


Figure 10.PWM control

We can find out more detail in
https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

So how can we control the Position and speed together ? I choose to use Cascade Control with Speed control inner loop and Position Control

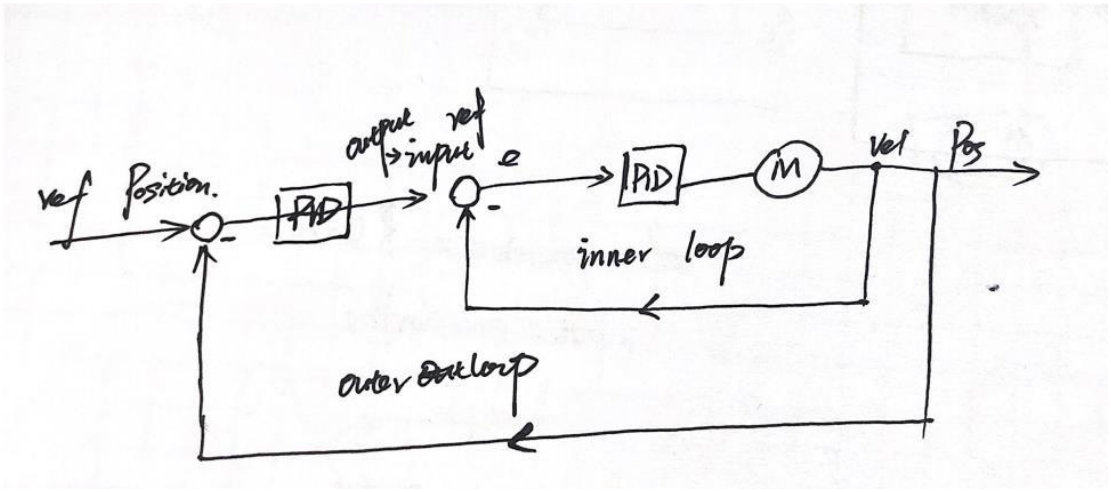


Figure 11.Cascade control of position and speed

Output of outer Loop is input of inner loop . We can easily control the Speed and Position with PID at same time now .

Optimal of Cascade Control

The control can be used now,however,we have to optimal the control to next level as follows:

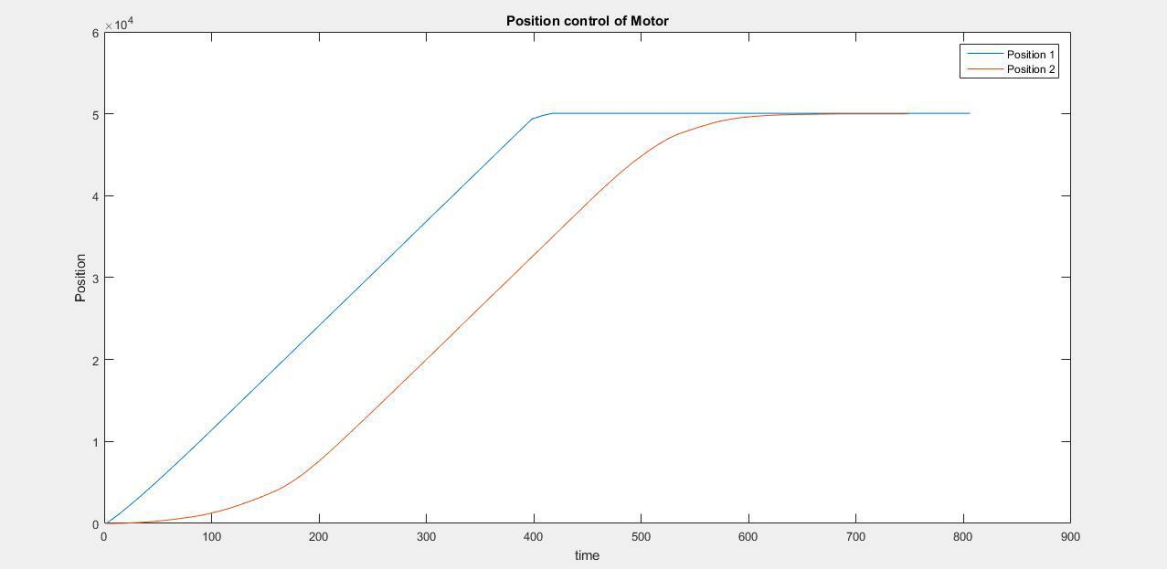


Figure 12.optimal control of position blue-before red-after

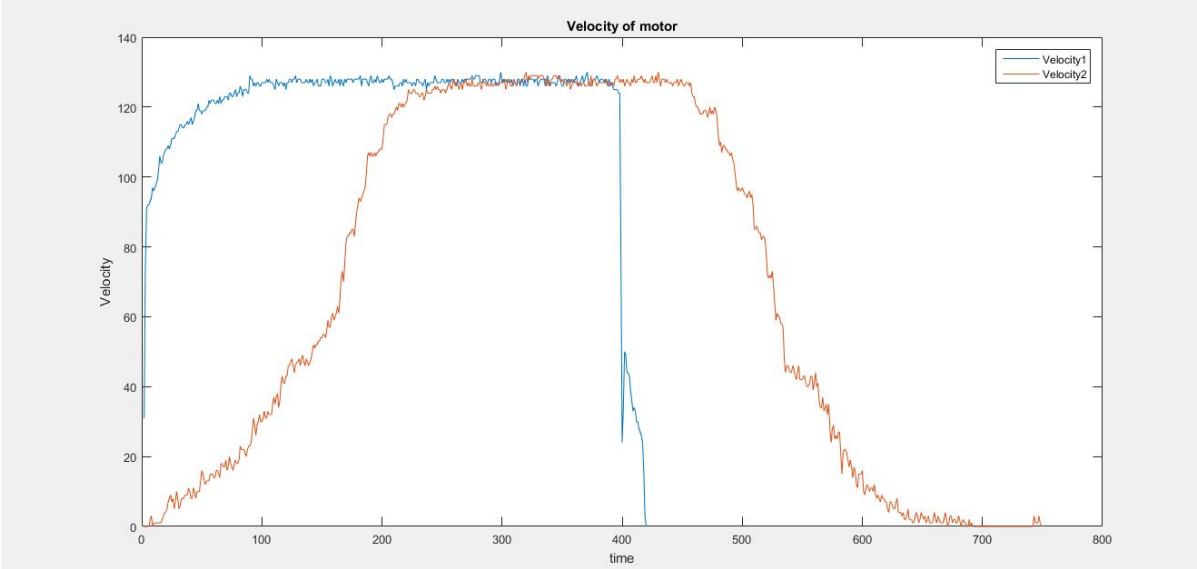


Figure 13.optimal control of Speed blue-before red-after

After optimal control of Speed and Position,the motor works much more smooth and more safety.

Arduino Nano board wireless control DC motor

Thus we have to combine wireless part and Drive motor part together in one Arduino Nano Board. So as we can see one Arduino

Nano board can control two DC motors with one NRF24L01 module.

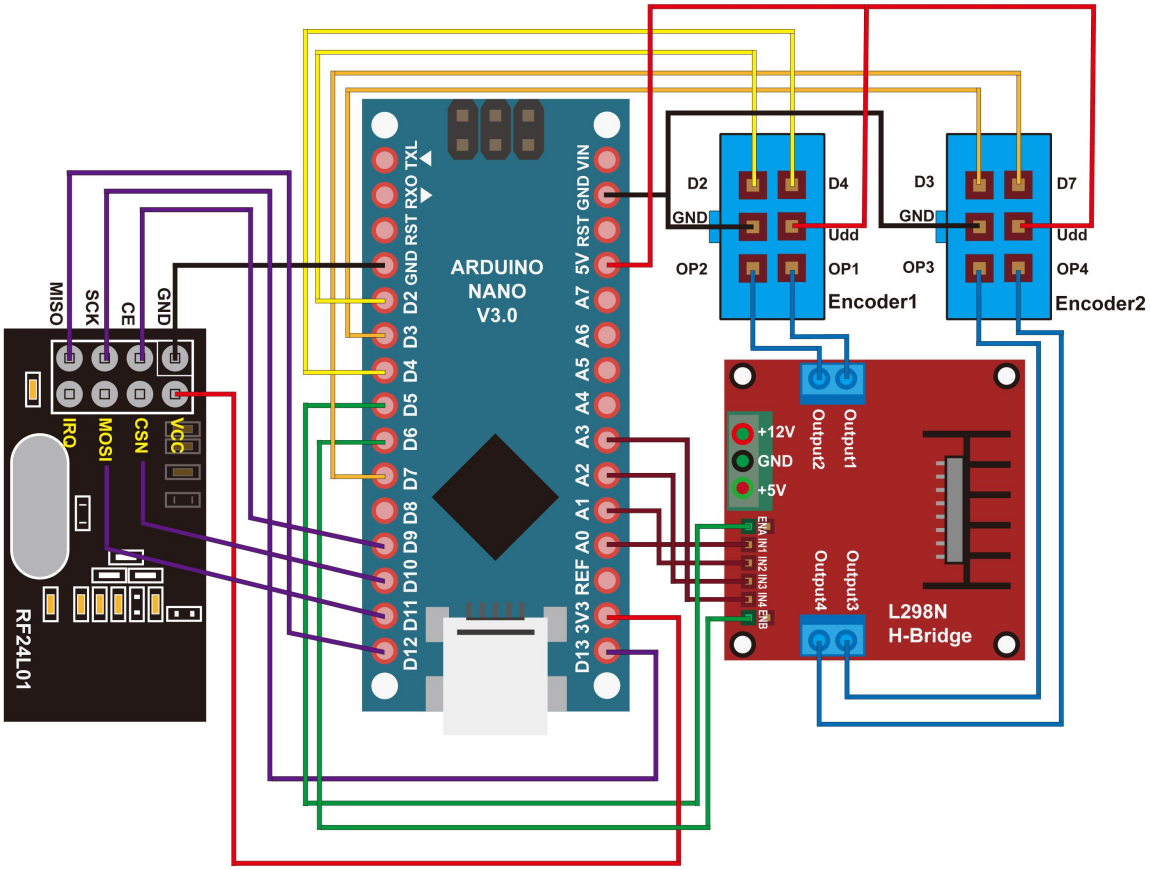


Figure 9. Arduino nano board connect with L298N , 2 DC motors and RF24L01

So a control module is done like follows:

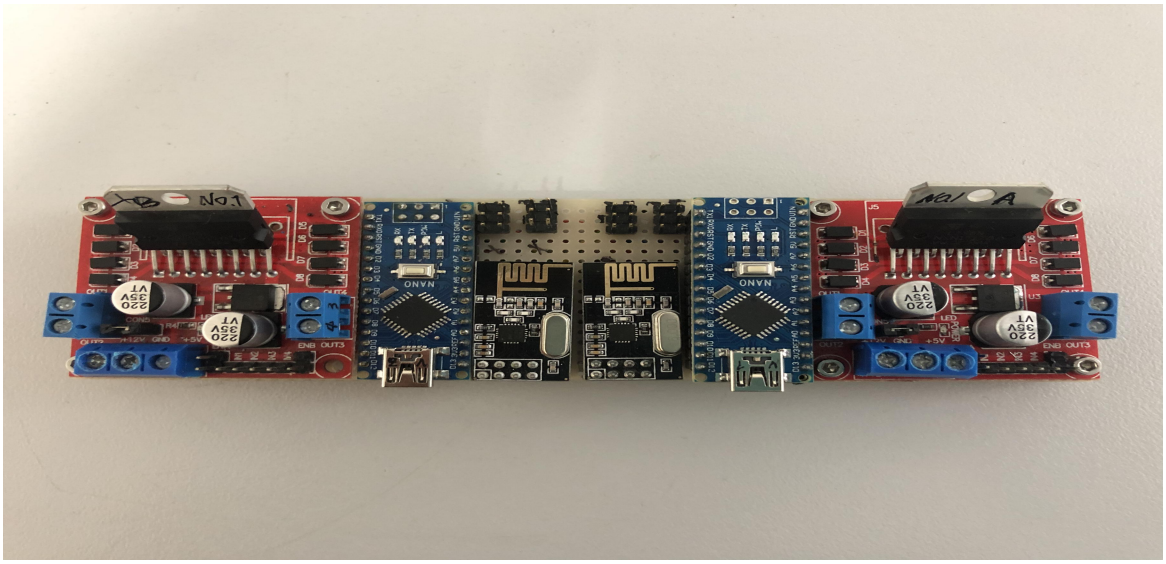


Figure 10. Arduino nano board connect with L298N , 2 DC motors and RF24L01

The Problem we will have :

PIN 2,3 have interrupt function,

so Pin 2 corresponding interrupt 1 so as **attachInterrupt(0, encoder, FALLING);**

so Pin 3 corresponding interrupt 2 so as **attachInterrupt(1, encoder2, FALLING);**

Because we need to wireless control so we are going to change Laboratory power supply into 24V 18650 Battery,so we have to know how to use Battery to drive motor and use the Laboratory power supply to charge the Battery.

Force sensor connects with Arduino Uno and test

1. Components

- force torque sensor DYMH-106

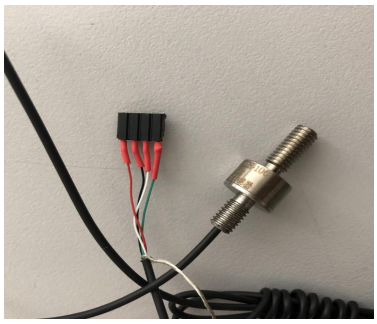


Figure 11. DYMH-106

- HX711

HX711 board reads the information from the Load Cell, amplifies the signals and then sends it to the Arduino for processing.

Common Load cells have four-wire to connect the HX711 to a microcontroller like the Arduino.

To connect the HX711 to the Arduino you only need 2 pins (Clock(SCK) and Data(DT)).

On the opposite side you can see the connection for the Load cell.

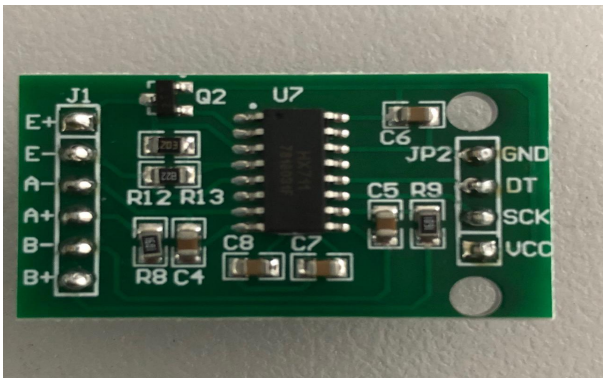


Figure 12. HX711

- Breadboard and Jump Wires
- Arduino Uno board

2. Connection

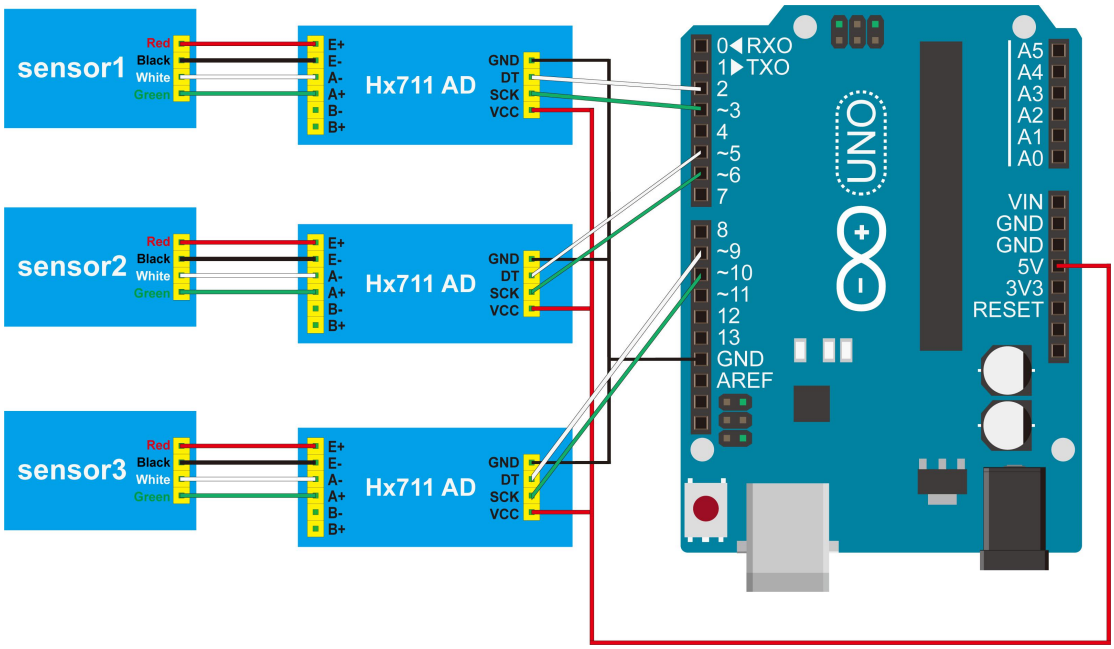


Figure 13. HX711,sensor, arduino connection

3. Code

When we finished connection of the circle we can learn basic code from the web

https://www.instructables.com/id/Arduino-Scale-With-5kg-Load-Cell-and-HX711-Amplifi/ .

And we have to extend to connects 3 HX711 modules to one Arduino board.After we search the data sheet of HX711, we only need to choose two I/O pins from arduino board to connect SCK and DT pin on the HX711.