In this lab, we learn some new modules, such as servo, encoder and ping.

(1) Servo testing and adjustment

In this part, we test the servo and adjust them. At first I can't find where the potential meter is, so I screw it from the button and find there is only PCB board. Fortunately, I found out it is just at the side of case.

Also in this part we learned that the speed and direction is adjustable by using PWM signal.

(2) Optical encoder

In this part we learned how to use encoder, which is a photodiode and it will trigger by dark or bright surface. And we to use ticker to check whether is above the threshold. And here's the result.

```
Before start
encoder = 1
encoder = 2
encoder = 4
encoder = 10
encoder = 13
encoder = 15
encoder = 16
encoder = 19
encoder = 24
encoder = 24
encoder = 24
encoder = 24
```

(3) Ping

In this section, we use two modules, one with ultra sound and the other with laser. Both detect the duration the sound or the light traverse from and back to the module.

Here's the result of ultra sound part.

```
Ping = 19.311138
Ping = 15.434750
Ping = 13.558507
Ping = 13.062895
Ping = 11.930070
Ping = 11.098151
Ping = 10.514037
Ping = 10.478638
Ping = 10.407835
Ping = 8.053682
Ping = 7.912079
Ping = 7.841277
Ping = 7.858978
Ping = 5.805731
```

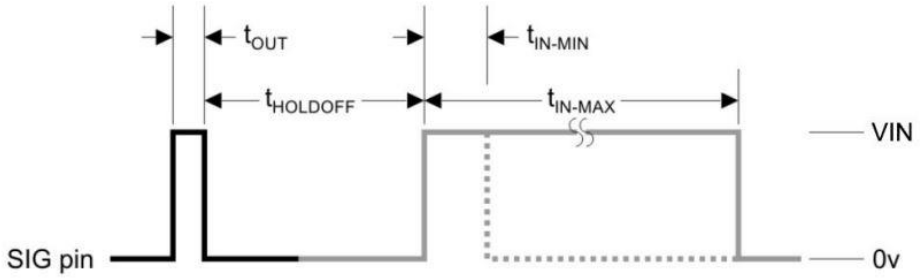
And according to the protocol and formula in datasheet. Here's the result of laser

module.

$$Distance\ (mm) = Pulse\ Width\ (ms) \times 171.5$$

PWM Mode

PWM default mode is designed to be code-compatible with PING))) Ultrasonic Distance Sensor (#28015) code. It can communicate with 3.3 V or 5 V TTL or CMOS microcontrollers. PWM Mode uses a bidirectional TTL pulse interface on a single I/O pin (SIG). The SIG pin will idle low, and both the input pulse and echo pulse will be positive high, at the VIN voltage.



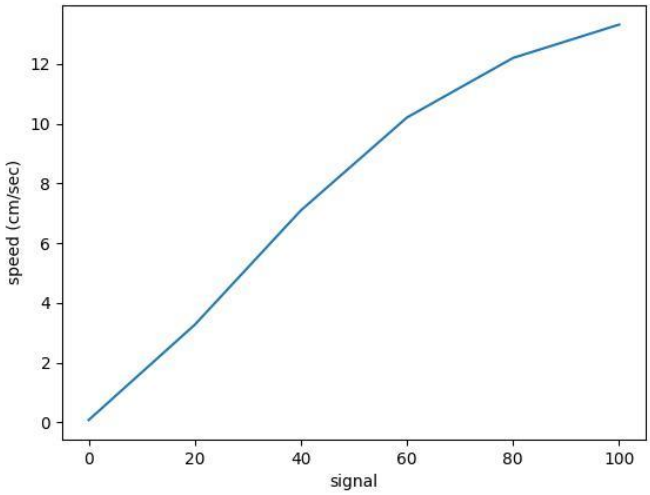

```
Ping = 45.207401 cm
us = 2172.000000
Ping = 37.249798 cm
us = 1879.000000
Ping = 32.224850 cm
us = 1642.000000
Ping = 28.160299 cm
us = 1343.000000
Ping = 23.032450 cm
us = 1099.000000
Ping = 18.847851 cm
us = 743.000000
Ping = 12.742450 cm
us = 365.000000
Ping = 6.259750 cm
us = 204.000000
```

(4) Calibration table and its implementation

In this part we collect some speed data with different steps, and through regression curve we can find rough prediction speed.

So I use normal speed motors and (+15 speed/ step), and here's the result with two wheels.

Wheel 1 :

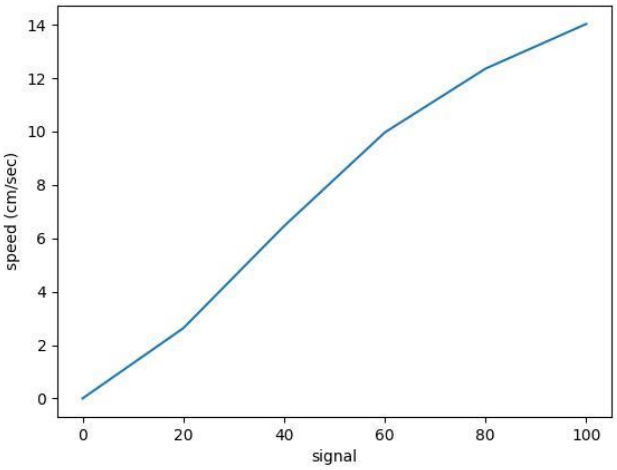



```
b'0.080\r\n'
b'3.269\r\n'
b'7.095\r\n'
b'10.204\r\n'
b'12.197\r\n'
b'13.313\r\n'
```

```
6.776
6.776
6.856
6.776
6.856
6.856
6.776
6.856
```

```
8.132
8.132
8.132
```

Wheel 2:




```
b'0.000\r\n'
b'2.631\r\n'
b'6.457\r\n'
b'9.965\r\n'
b'12.357\r\n'
b'14.031\r\n'
```

```
7.015
6.776
7.015
6.856
6.856
6.856
```

```
8.052
7.892
8.291
7.813
```

(5) Discussion

I spend more than 3 hours to find out what's going on at the last section. And it's strange no one in class ask such question = =. Fortunately, the problem is solved, the regression curve looks more reasonable though there are still some factor that will make servos unstable, the wheel suffers from shuttering will cause encoder not recording data accurately and others little factors may lead to the poor accuracy, but still happy to solve the problem.