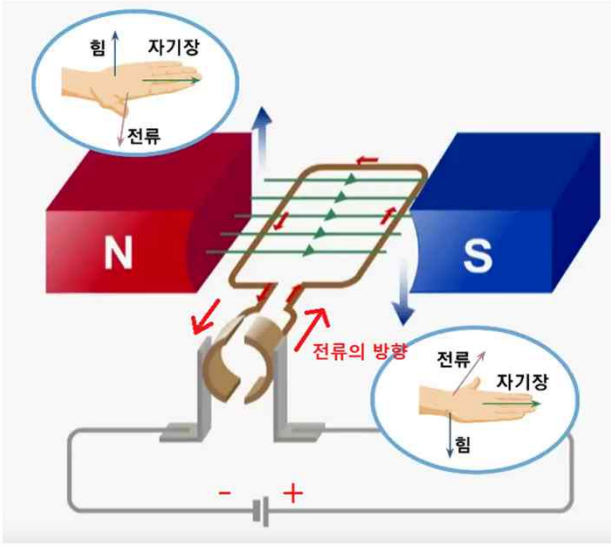
엔코더 DC모터 제어

백지훈

모터

<정의>

전기에너지를 기계에너지로 바꾸는 물건



https://www.youtube.com/watch?v=LAtPHANEfQo

모터 제어 방법 - 속도

- 모터의 + - 단자의 전압을 조절한다.

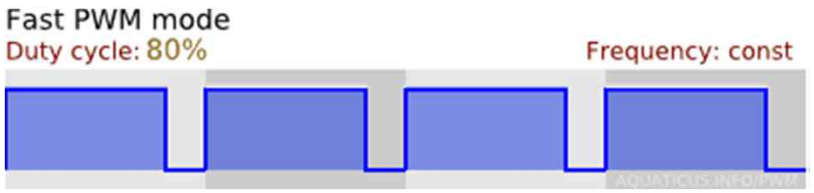
$$V = IR \ , \qquad I = \frac{V}{R}$$

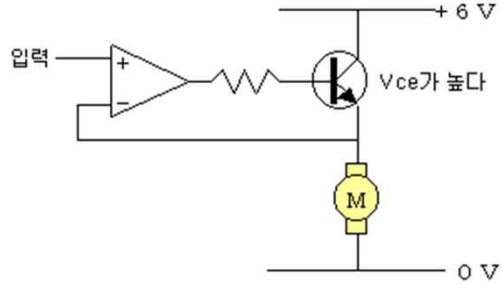
- 옴에법칙에 의하면 전압을 제어함에 따라 전류는 정비례한다.
- 전류가 제어됨에 따라 모터의 회전속도 또한 제어된다.

- PWM을 이용한 제어 (Pulse Width Modulation : 펄스 폭 변조)

: 구형파의 듀티사이클을 변화시키는 변조 방식을 이용하여 트랜지스터를 스위칭하여 제어 – 전압의 평균값이 변화되는 원리를 이용

주의사항 – 고속의 PWM을 이용하지 않으면 모터가 연속하여 회전하지 않고 그 때문에 고장의 원인이 되기도 한다.



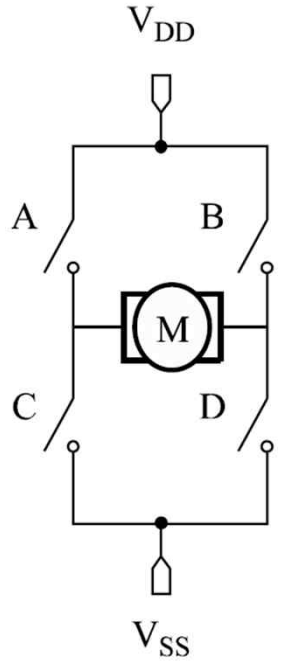


모터 제어 방법 - 방향

□ DC 모터의 정회전과 역회전을 위한 H-브리지 회로

➢ **H-브리지 회로**

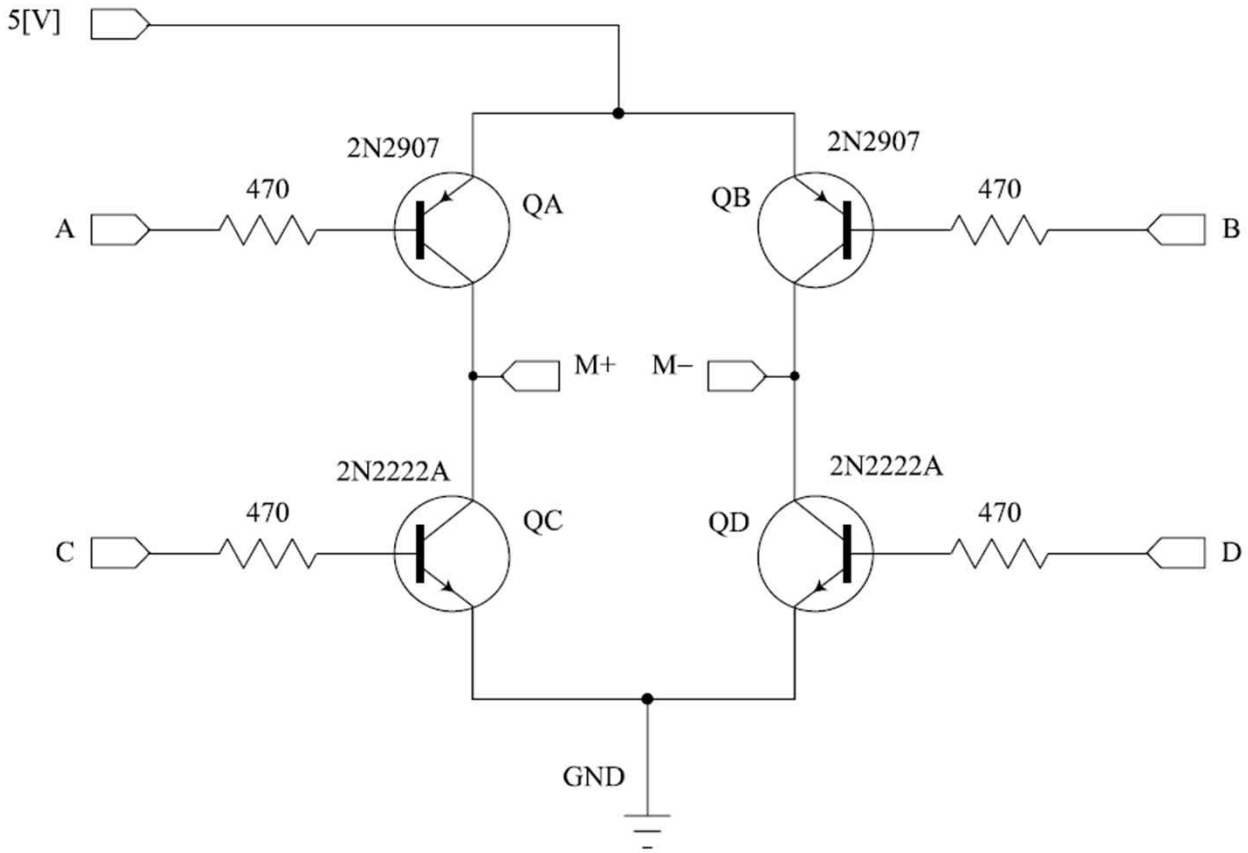
- 모터의 전류 흐름 : 좌 → 우
  - » A, D 단락
  - » B, C 개방
- 모터의 전류 흐름 : 우 → 좌
  - » A, D 개방
  - » B, C 단락
- 모터 전류 차단
  - » A, B, C, D 모두 개방
  - » (혹은) A, B 개방 혹은 C, D 개방



[H-브리지 회로와 스위치]

모터 제어 방법 - 방향

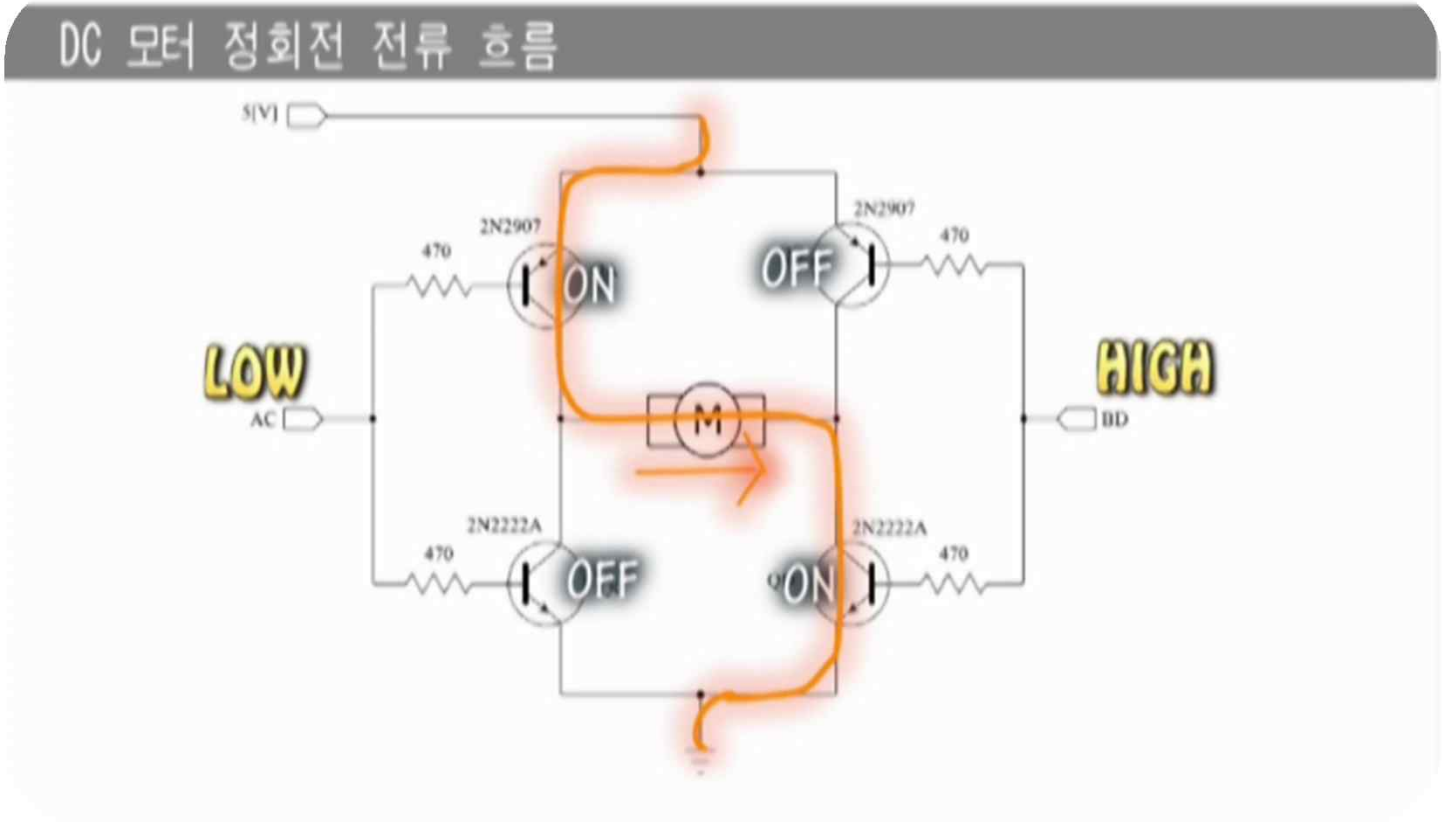
□ 소형 모터 구동을 위한 H-브리지 회로



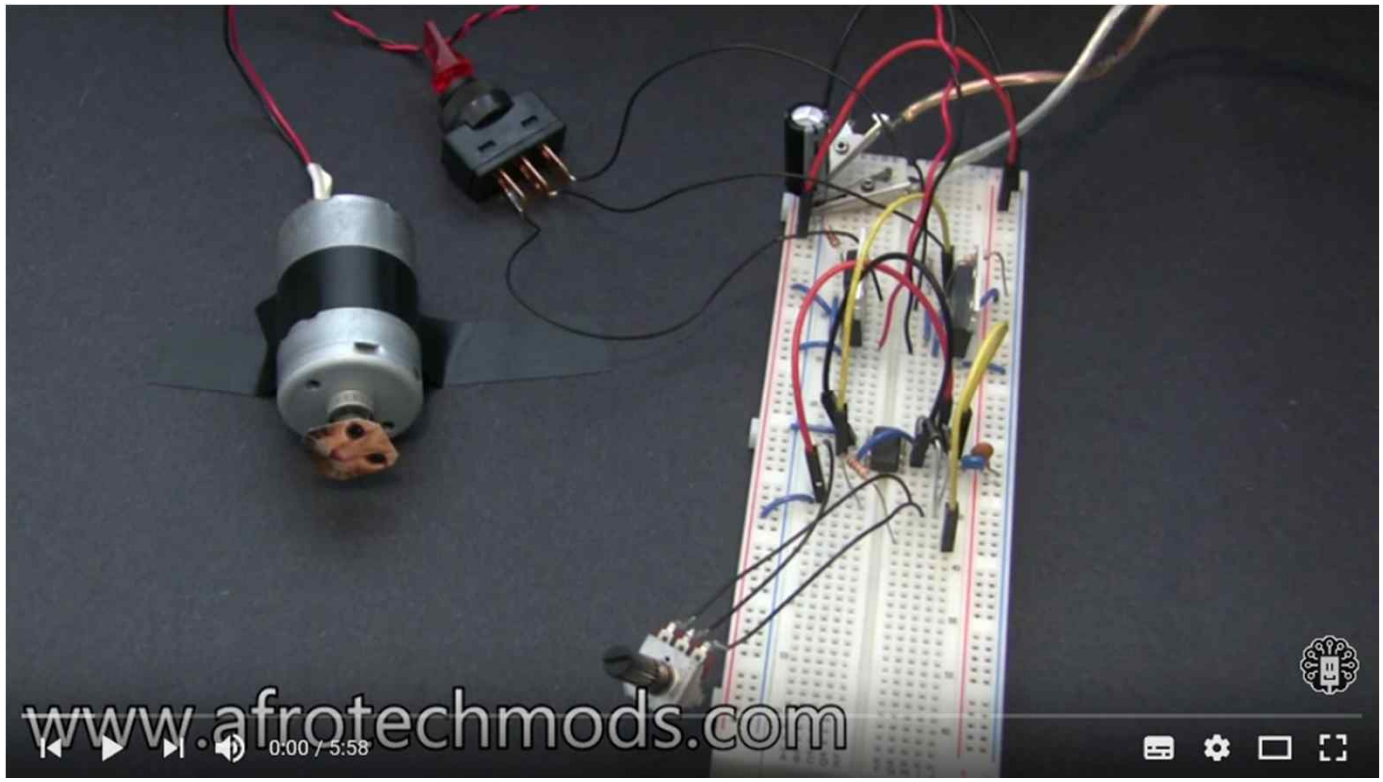
[H-브리지 회로를 이용한 DC 모터 정·역방향 제어 회로]

모터 제어 방법 - 방향

□ H-브리지 회로를 이용한 소형 DC 모터 정•역회전



모터 제어 방법 : 참고 링크



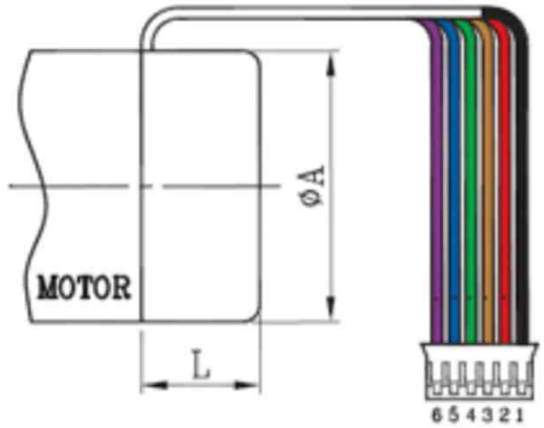
https://www.youtube.com/watch?v=iYafyPZ15g8&list=PL39ETiCoYrTP513dH-OdUKLlAfjiSQRJ_

실험에 사용한 모터

RB-35GM+ENCODER 11TYPE (12V)

엔코더결합형 감속기어모터

지능형,청소로봇에 사용되며, 정밀제어가 가능한 엔코더 부착형 모터
감 속 비 : 헬리컬타입 1/10 ~1/3000 (감속비율 총 20종)
정 격 토 크 : 0.5 kg-cm ~ 6.0 kg-cm
정격회전수 : 490 rpm ~ 2.0 rpm
장착된모터 : DC 12v / 6,200 rpm / 3.14 W Motor
엔코더사양 : 26Pulses (13Pulses x 2CH)

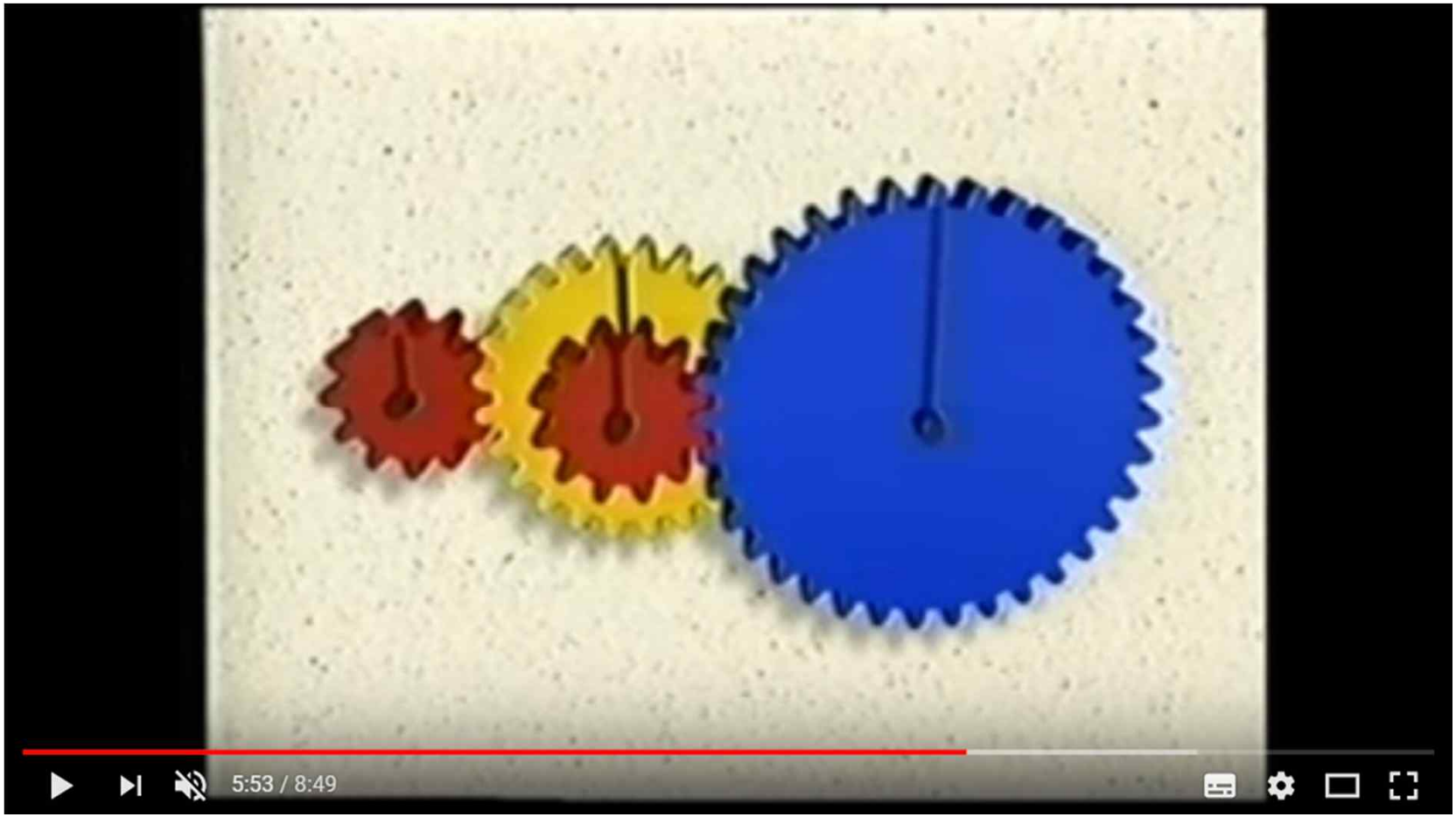


엔코더 컨넥터 핀별 내용 :
Two Channel Encoder Connections :

1. Black	:	-MOTOR
2. Red	:	+MOTOR
3. Brown	:	HALL SENSOR Vcc
4. Green	:	HALL SENSOR GND
5. Blue	:	HALL SENSOR B Vout
6. Purple	:	HALL SENSOR A Vout

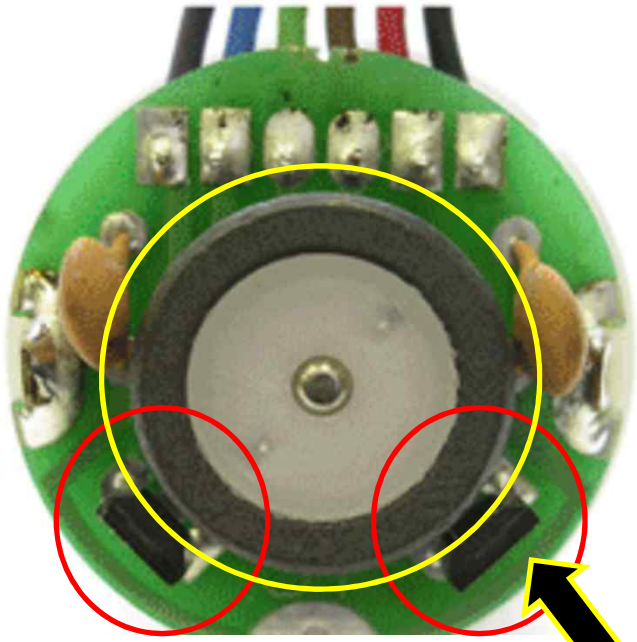
홀 센서를 이용하여 회전 수를 알 수 있고, 이동거리와 속도를 구할 수 있다.
두 개의 홀 센서를 이용하여 모터의 회전 방향을 알 수 있다.

감속 비(기어 비)



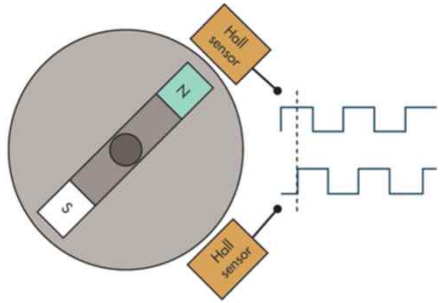
https://www.youtube.com/watch?v=D_i3PJIYtuY

엔코더



홀 센서

자석



홀 센서를 이용하여 회전 수를 알 수 있고, 이동거리와 속도를 구할 수 있다.
두 개의 홀 센서를 이용하여 모터의 회전 방향을 알 수 있다.

엔코더

모터 기어비 : $\frac{1}{50}$

엔코더 기어비 : $\frac{1}{13}$

홀센서 : 엔코더 자석이 한 바퀴 회전할 때 마다 Pulse 출력

엔코더 자석이 1바퀴 회전할 때 모터 회전 : $\frac{1}{13} \times \frac{1}{50} = \frac{1}{650}$

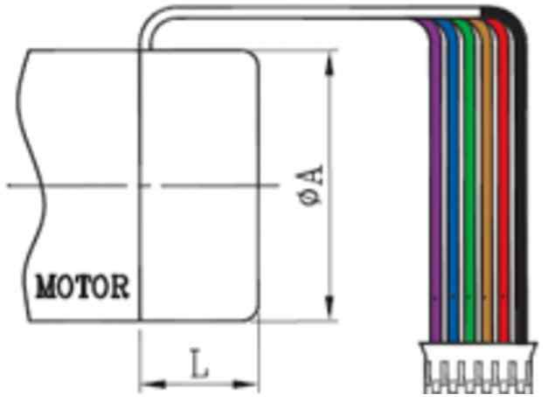
즉 엔코더에서 펄스가 650번 출력이 되면 모터는 1바퀴 회전한다!

- **펄스당 회전각 : 2π/650**
- **펄스당 바퀴의 주행거리 : 바퀴의 반지름 * 2π/650**

엔코더 실험1. 전압에 따른 펄스 출력

	모터 입력전원 : 3.9V	모터 입력전원 : 10.2V
주기	2.270~2.416ms	830~849us
주파수	413~440Hz	1.148~1.211khz
듀티사이클	약 43~46%	
오버 슈트	6.122%	
진폭	4.9V	

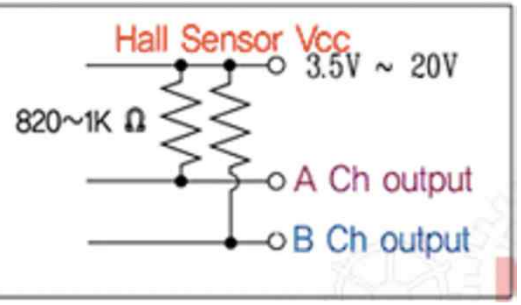
회로 구성



엔코더 컨넥터 핀별 내용 :
Two Channel Encoder Connections :

1. Black : -MOTOR
2. Red : +MOTOR
3. Brown : HALL SENSOR Vcc
4. Green : HALL SENSOR GND
5. Blue : HALL SENSOR B Vout
6. Purple : HALL SENSOR A Vout

출력회로 구성
Output Circuit :

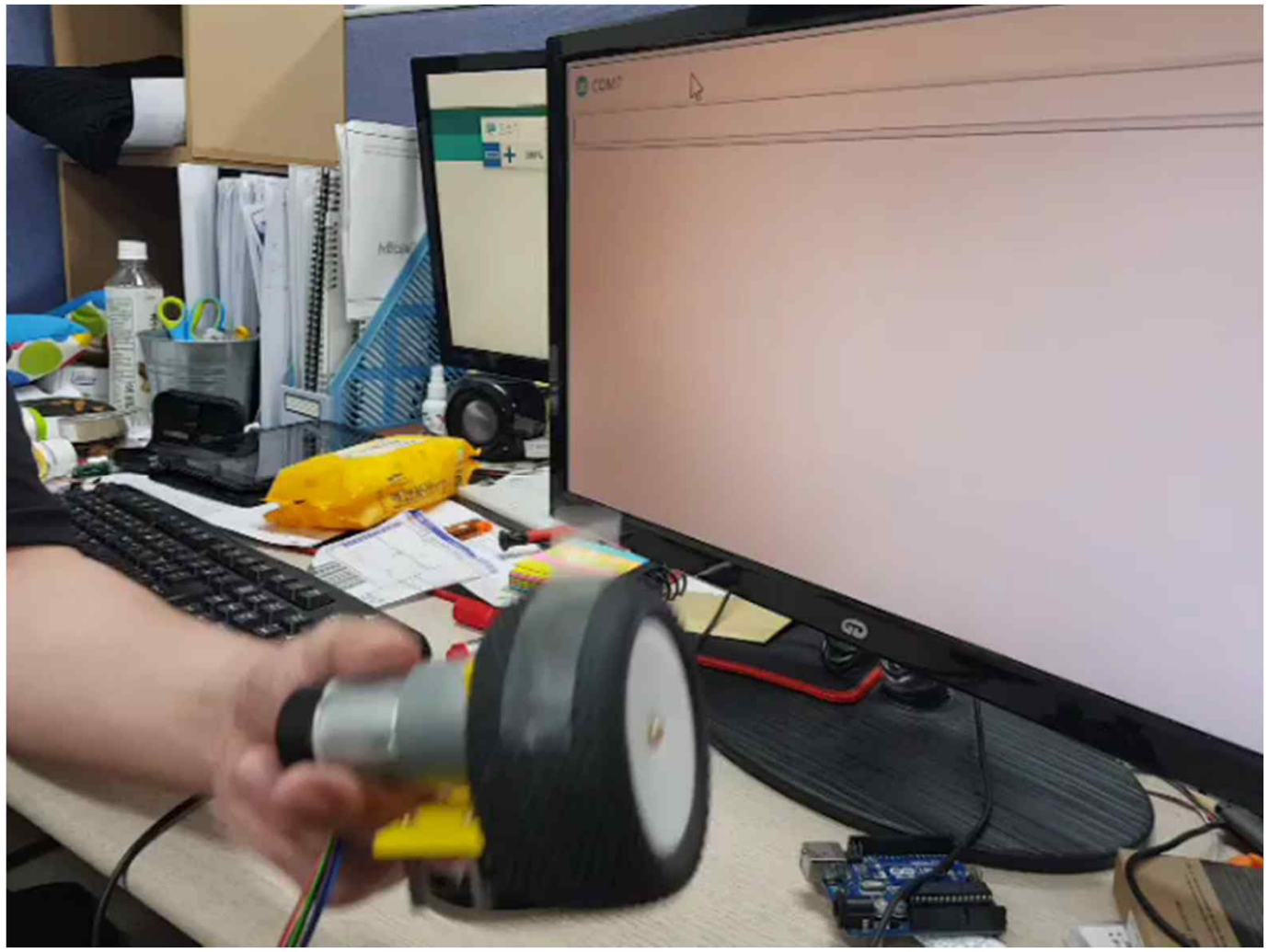


엔코더 실험3. 아두이노 실험

```
const int encoderPinA = 2;
const int encoderPinB = 7;

const int encoderPinC = 3;
const int encoderPinD = 8;

int encoderPos1 = 0;
int encoderPos2 = 0;
int wheel1 = 0;
int wheel2 = 0;

String Ser1 = "R+";
String Ser2 = "R-";
String Ser3 = "L+";
String Ser4 = "L-";

unsigned char num = 0;
void doEncoderA(){ // 빨녹일 때
  if(digitalRead(encoderPinB)==HIGH)
    encoderPos1++; // 정회전
  else
    encoderPos1--; // 역회전

    wheel1 = encoderPos1/65;
    if(wheel1 > 0)
    {
      encoderPos1 = 0;
      Serial.println(Ser1);
    }
    if(wheel1 < 0)
    {
      encoderPos1 = 0;
      Serial.println(Ser2);
    }
}
```

```
void doEncoderC(){ // 빨녹일 때
  if(digitalRead(encoderPinD)==HIGH)
    encoderPos2++; // 정회전
  else
    encoderPos2--; // 역회전

    wheel2 = encoderPos2/65;
    if(wheel2 > 0)
    {
      encoderPos2 = 0;
      Serial.println(Ser3);
    }
    if(wheel2 < 0)
    {
      encoderPos2 = 0;
      Serial.println(Ser4);
    }
}

void setup() {
  pinMode(encoderPinA, INPUT_PULLUP);
  attachInterrupt(0, doEncoderA, RISING);
  pinMode(encoderPinB, INPUT_PULLUP);

  pinMode(encoderPinC, INPUT_PULLUP);
  attachInterrupt(1, doEncoderC, RISING);
  pinMode(encoderPinD, INPUT_PULLUP);

  Serial.begin(115200);
}

void loop() {
}
```

엔코더 실험3. 아두이노 실험

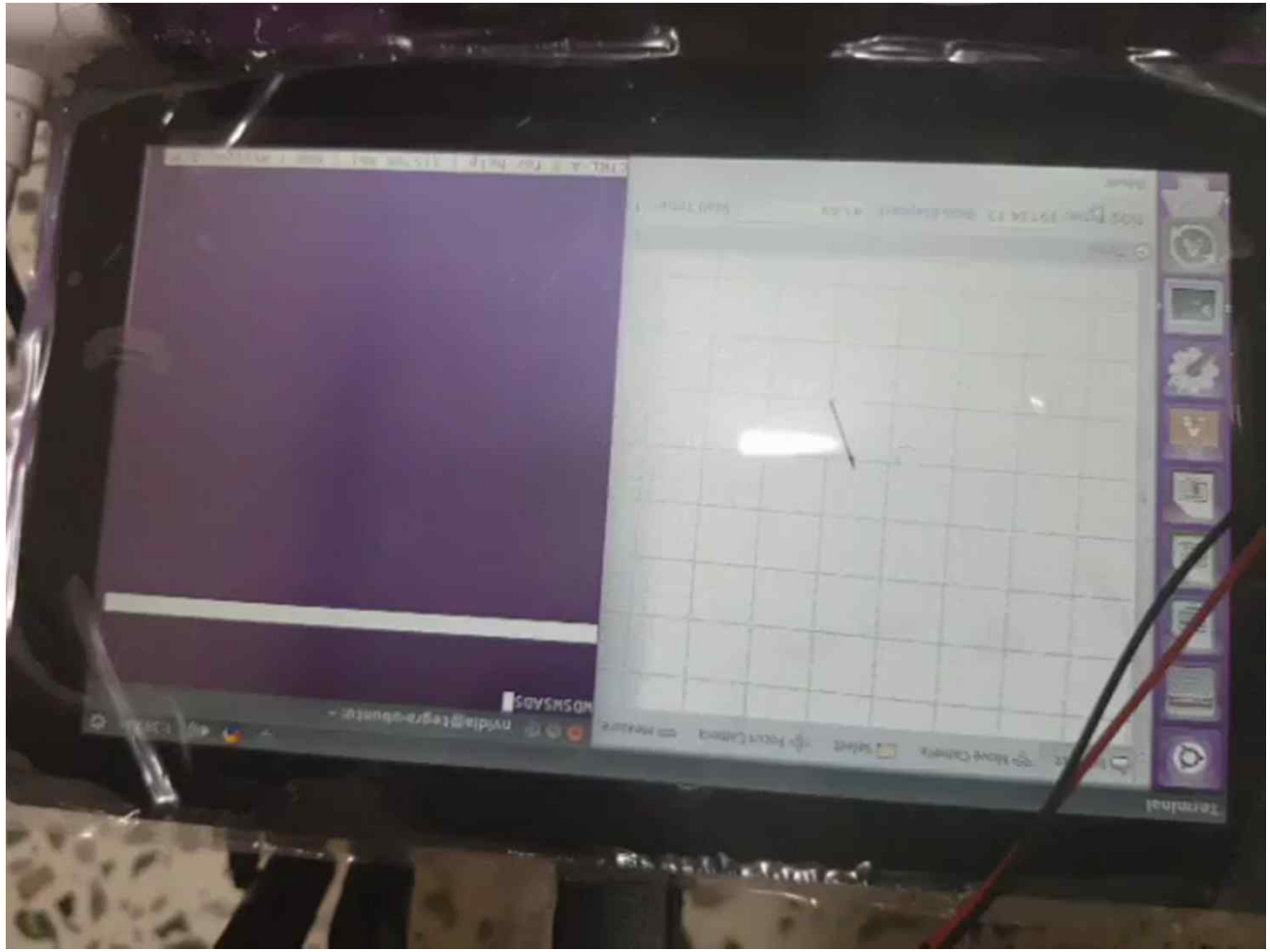


엔코더 실험3. 아두이노 실험

ROS 주행거리 테스트

아두이노의 연산 속도를 고려하여 펄스가 65번 발생할 때마다 시리얼 데이터 송신
실제 주행 거리 : 45[cm]
계산된 거리 : 43.83[cm]

엔코더 실험4. 로봇 주행거리 측정



엔코더 모터와 Dead Reckoning을 이용하여 주행거리 측정

결론

1. 모터를 회전시키는 힘은 자기력이다.

2. 모터의 각속도는 전류가 클수록 커진다.
(정격 이상의 전압 인가시 과부하)

3. 모터 제어 방법
브릿지 회로 – 방향 제어
고속 PWM - 속도 제어

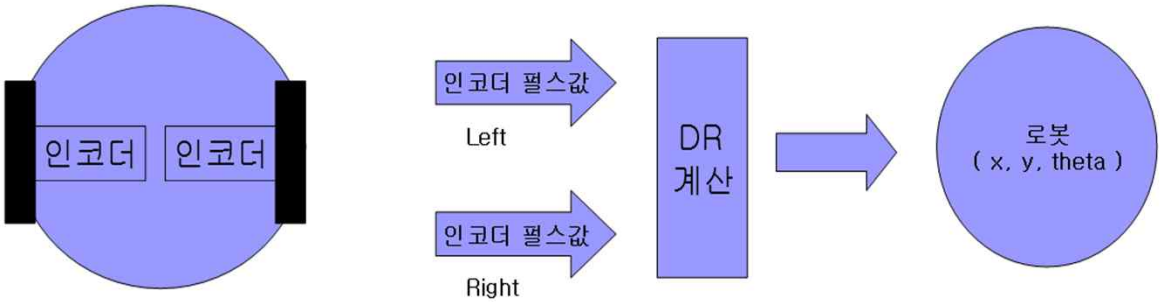
4. 엔코더의 펄스 발생수를 통하여 이동거리 계측 및 속도 계산이 가능하다. (엔코더가 결합된 모터를 사용할 경우에 한함.)

5. 엔코더의 채널 1, 2번 펄스를 통하여 모터의 회전방향을 알 수 있다.

ROS 주행거리 테스트 – Dead Reckoning

Dead Reckoning

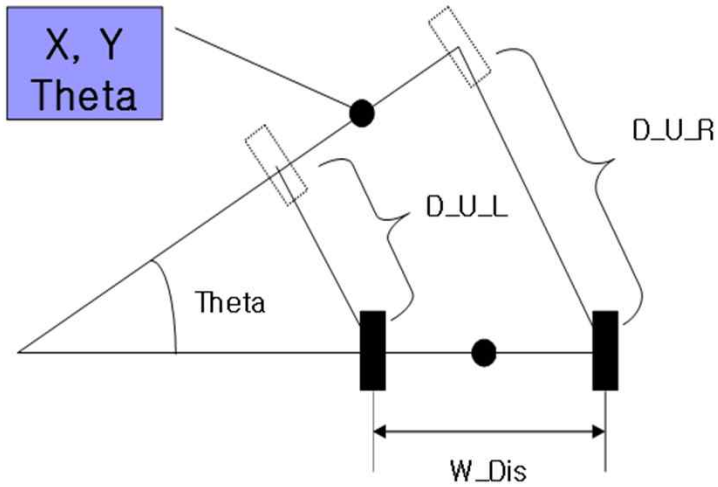
- 정 의
  - Navigation using only sensing internal to the robot
  - 외부의 입력이 아닌 로봇 자체에서 측정되는 sensing 값으로 로봇의 위치계산
- 로봇에서 DR



장점 : 내부의 sensing 값만으로 로봇의 위치를 간단한 수식으로 계산 가능
단점 : 센서값에 의존도가 높아 바퀴의 슬립 등 외부환경의 요인이 로봇에 영향을 미칠 경우
계산 값에 오차는 커지고 그런 외부영향에 대처하기가 힘듬

ROS 주행거리 테스트 – Dead Reckoning

DR 원리



Theta = (D_U_R − D_U_L) / W_Dis

D_U = (D_U_L + D_U_R) / 2
X = D_U * cos(Theta)
Y = D_U * sin (Theta)