



CNRLAB

CIM&ROBOTICS LABORATORY

로봇공학입문설계

5주차 모바일 로봇(3)

로봇공학과

Contents

- Report
- Encoder
- Interrupt
- Distance Keeping Robot

모바일 로봇 프로젝트(1)

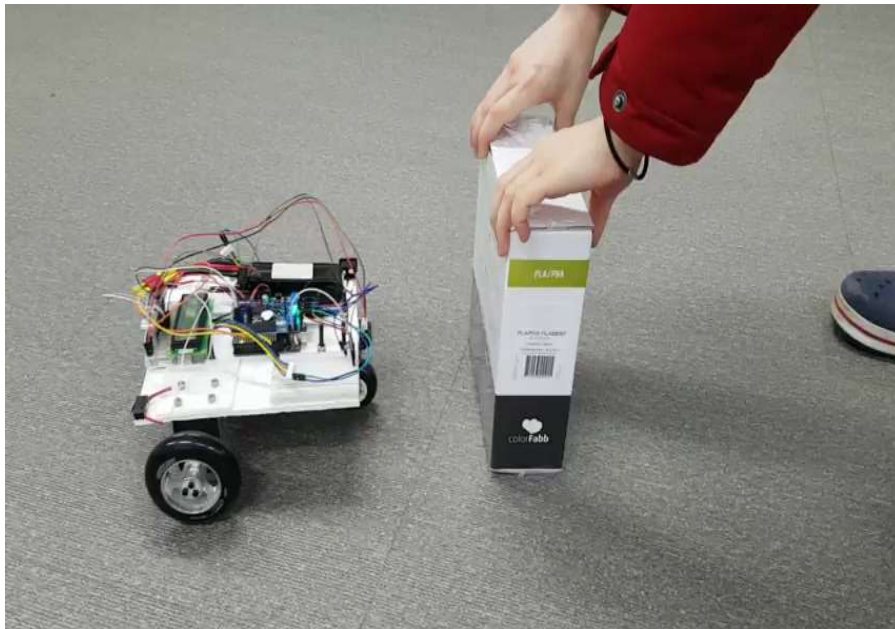
□ 개별 레포트

1. 이론 : 모터, 엔코더, 인터럽트
 - ▶ 원리 및 종류 등에 대하여 핵심사항만 정리
 2. 역할 분담
 - ▶ 팀원 별 역할 수행 내용
 - ▶ 기여도 : 점수 총합 100점 기준으로 자신 이외의 다른 팀원에 대한 점수 부여 (필수 사항 아님)
 3. 실험 내용
 - ▶ 실험 목표
 - ▶ 로봇 구성 : 사용한 부품 및 하드웨어 구성
 - ▶ 알고리즘 : 가급적 순서도 혹은 그림으로 설명
 4. 실험 결과 및 분석
 - ▶ 하드웨어 구성(실험2)
 - ▶ 소스코드 분석
 - ▶ 시행착오, 개선사항, 한계점
 5. 소스코드
- ※ 이론, 실험 결과 및 분석은 팀 내 공유 금지

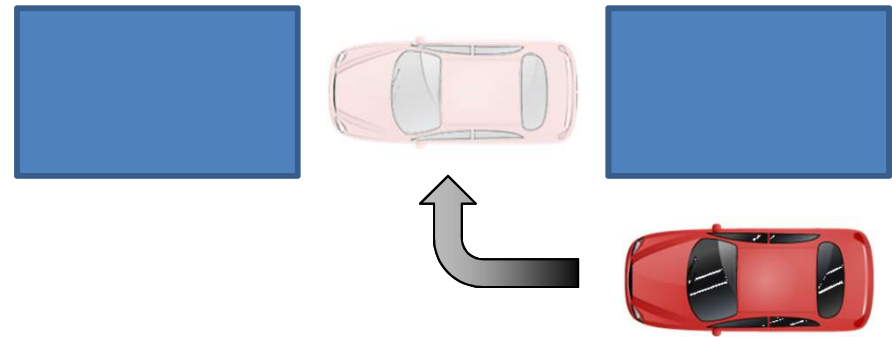
□ 제출기한 : 4월 13일

모바일 로봇 프로젝트(1)

실험 1. Distance Keeping



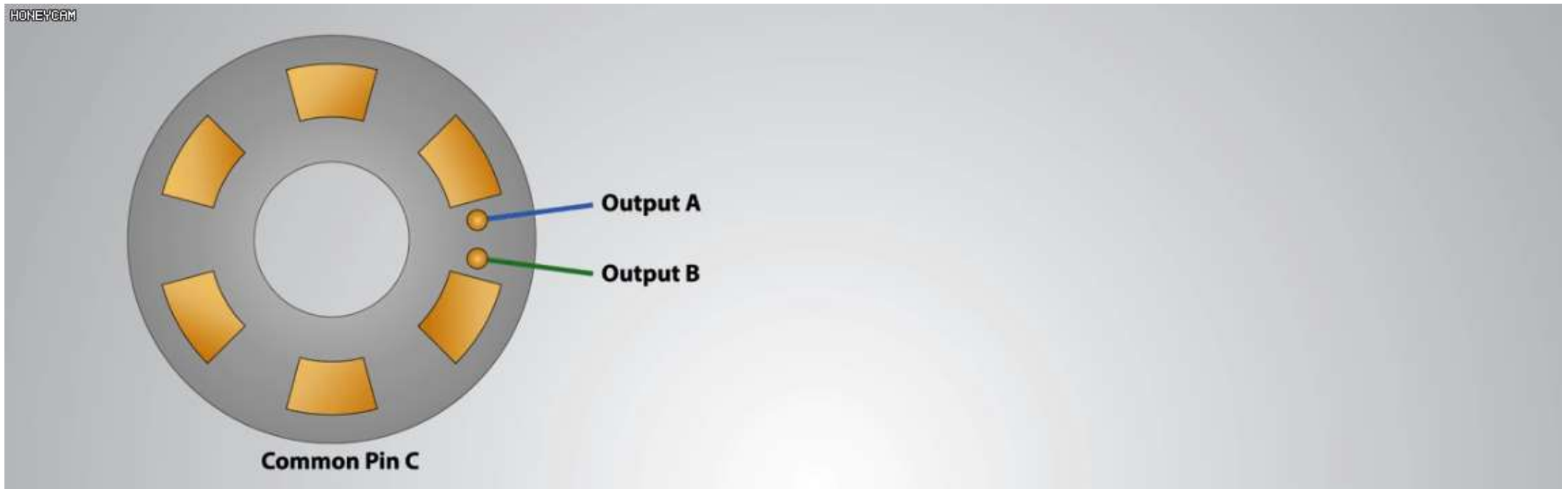
실험 2. Autonomous Parking



실험 영상 : rlagywjd802@naver.com 으로 조별로 전송(개인별 전송 아님)
메일 제목 : [로봇공학과][1조][Distance Keeping]
제출 기한 : 4월 13일

Encoder

□ 구동 원리

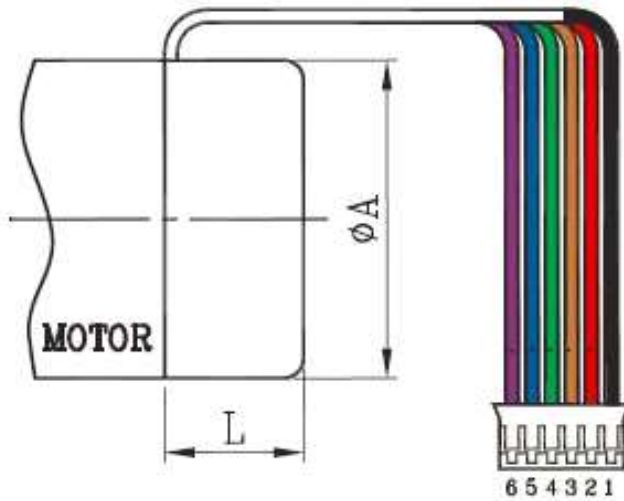
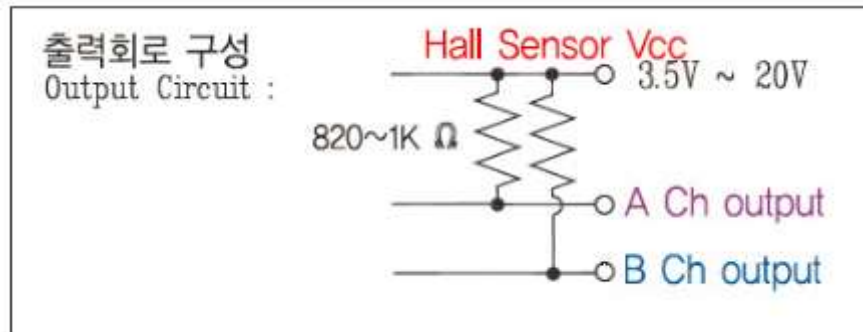


Encoder

❑ Encoder Specification (RA-35GM 11TYPE (12V) with 2channel Encoder)



- Reduction ratio : 1/30
- Encoder : 26P/R



엔코더 컨넥터 핀별 내용 :
Two Channel Encoder Connections :

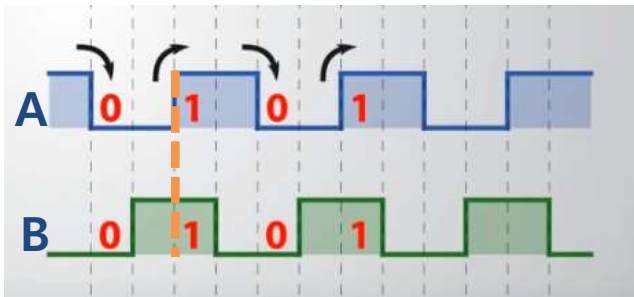
1.	Black	: -MOTOR
2.	Red	: +MOTOR
3.	Brown	: HALL SENSOR Vcc
4.	Green	: HALL SENSOR GND
5.	Blue	: HALL SENSOR B Vout
6.	Purple	: HALL SENSOR A Vout

Encoder

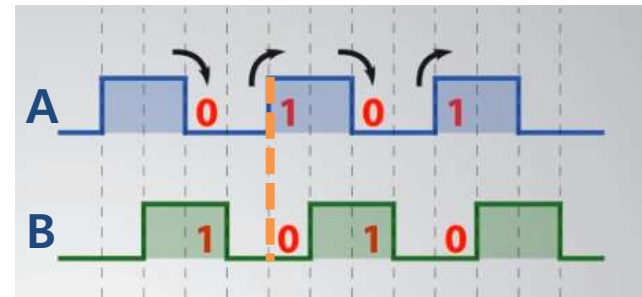
□ Enocoder란?

- 모터의 회전 방향과 회전 속도 측정

□ 회전 방향



정방향



역방향

□ 회전 속도

- 26P/R, 2채널 : 1회전시, ChA, ChB에서 각각 13pulse씩 출력
- 4채배 적용 : 1회전시, 총 $13[\text{pulse}] \times 4 = 52\text{번}$ pulse 체크
- 1회전 하는데 걸리는 시간을 측정하여 회전 속도[RPM]계산

Encoder

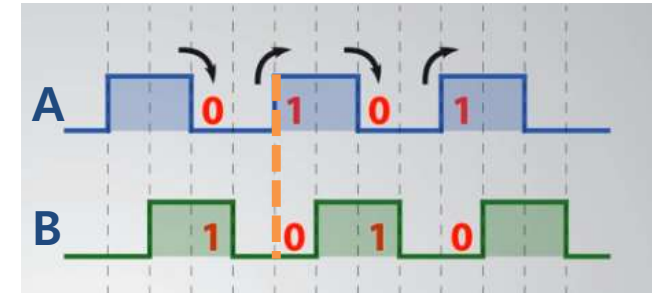
□ 구동 원리

```
// a low-to-high edge on channel A
if(digitalRead(EA_CHA) == HIGH) {
  if (digitalRead(EA_CHB) == LOW)      enAPos = enAPos - 1;
  else                                enAPos = enAPos + 1;
}

// must be a high-to-low edge on channel A
else {
  if (digitalRead(EA_CHB) == HIGH)    enAPos = enAPos - 1;
  else                                enAPos = enAPos + 1;
}
```

```
// a low-to-high edge on channel B
if(digitalRead(EA_CHB) == HIGH) {
  if (digitalRead(EA_CHA) == HIGH)    enAPos = enAPos - 1;
  else                                enAPos = enAPos + 1;
}

// must be a high-to-low edge on channel B
else {
  if (digitalRead(EA_CHA) == LOW)     enAPos = enAPos - 1;
  else                                enAPos = enAPos + 1;
}
```



역방향

Ch A \ Ch B	LOW	HIGH
	역방향	정방향
L→H	역방향	정방향
H→L	정방향	역방향

Ch B \ Ch A	LOW	HIGH
	정방향	역방향
L→H	정방향	역방향
H→L	역방향	정방향

□ 구동 원리

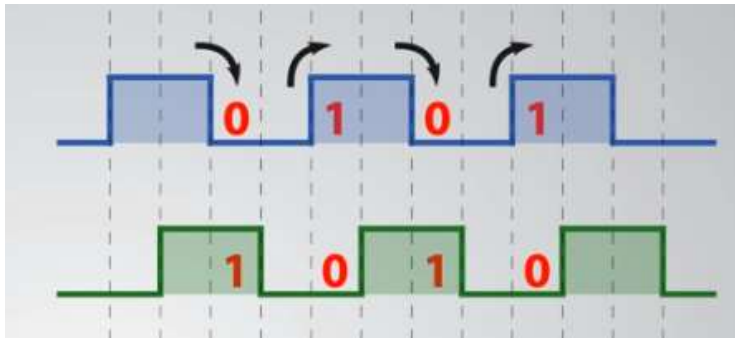
○ RPM계산

- ▶ 52번 pulse 체크(=1회전)하는 동안 걸리는 시간을 측정한다.
- ▶ 기어비 1/30이므로, 모터 축이 회전하는 속도는 1/30배를 해주어야 한다.

$$RPM = \frac{1\text{회전}}{\text{걸린시간}[min]} \times \text{기어비} = \frac{1,000,000}{dt[\mu s]} \times \frac{1}{30} \times 60 = \frac{2,000,000}{dt}$$

Encoder

- Edge를 detect하기 위해서는...?

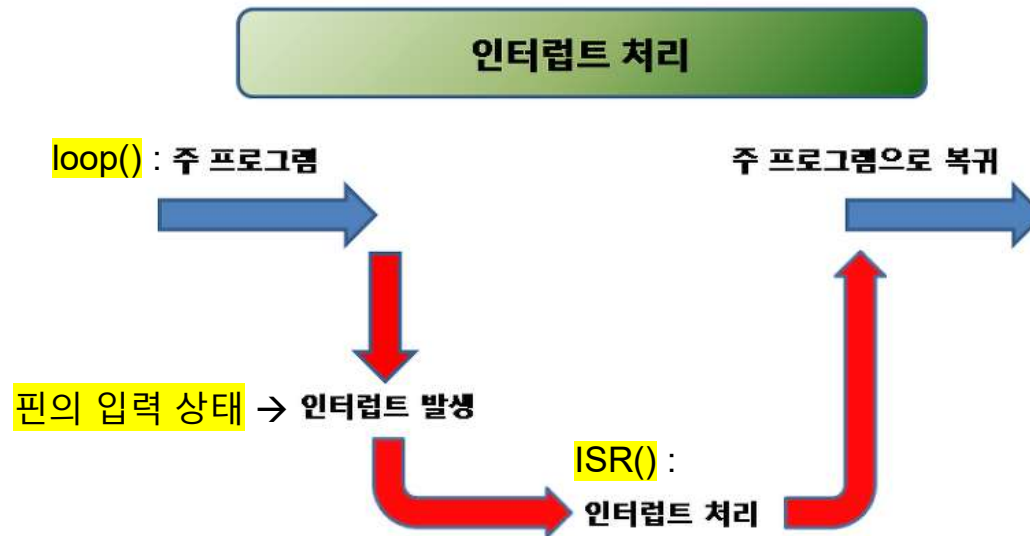


Interrupt 이용!

Interrupt는 **edge detecting**을 할 수 있을 뿐만 아니라 **특정 순간**(rising edge, falling edge 등)에 원하는 일을 수행할 수 있도록 한다.

□ Interrupt 란?

- 인터럽트는 특정 핀의 **입력 상태**가 바뀔 때, 아두이노가 이를 자동으로 감지하여 모든 동작을 잠시 멈춘 다음, ISR이라 부르는 함수를 실행하고 다시 원래 작업으로 복귀 하는 기능을 뜻한다.



□ Interrupt 관련 함수

`attachInterrupt(digitalPinToInterrupt(interruptPin), ISR, mode)`

digitalPinToInterrupt(interruptPin) : interrupt 번호

Board	int.0	int.1	int.2	int.3	int.4	int.5
Uno, Ethernet	2	3				
Mega2560	2	3	21	20	19	18
32u4 based (e.g Leonardo, Micro)	3	2	0	1	7	
Due, Zero, MKR1000, 101	interrupt number = pin number					

ISR : interrupt가 발생할 때 부르는 함수 명(interrupt service routine)

mode : interrupt가 trigger되는 시점을 정의

- CHANGE : LOW→HIGH, HIGH→LOW
- RISING : LOW→HIGH
- FALLING : HIGH→LOW

□ Interrupt 관련 함수

`attachInterrupt(digitalPinToInterrupt(interruptPin), ISR, mode)`

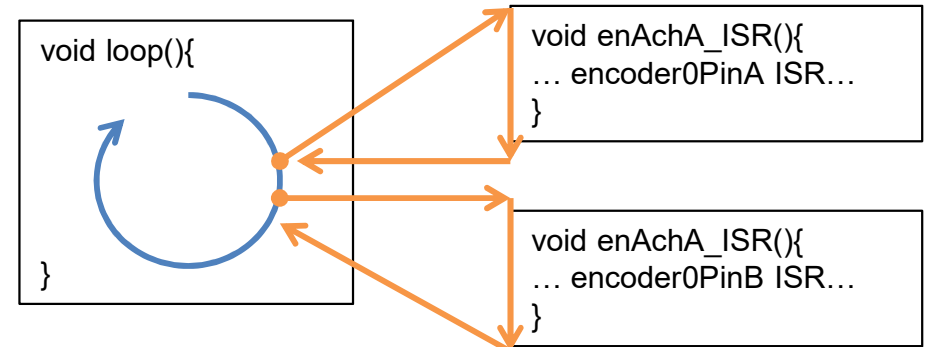
Board	int.0	int.1	int.2	int.3	int.4	int.5
Uno, Ethernet	2	3				
Mega2560	2	3	21	20	19	18
32u4 based (e.g Leonardo, Micro)	3	2	0	1	7	
Due, Zero, MKR1000, 101	interrupt number = pin number					

적용예시

```
#define EA_CHA 2
#define EA_CHB 3

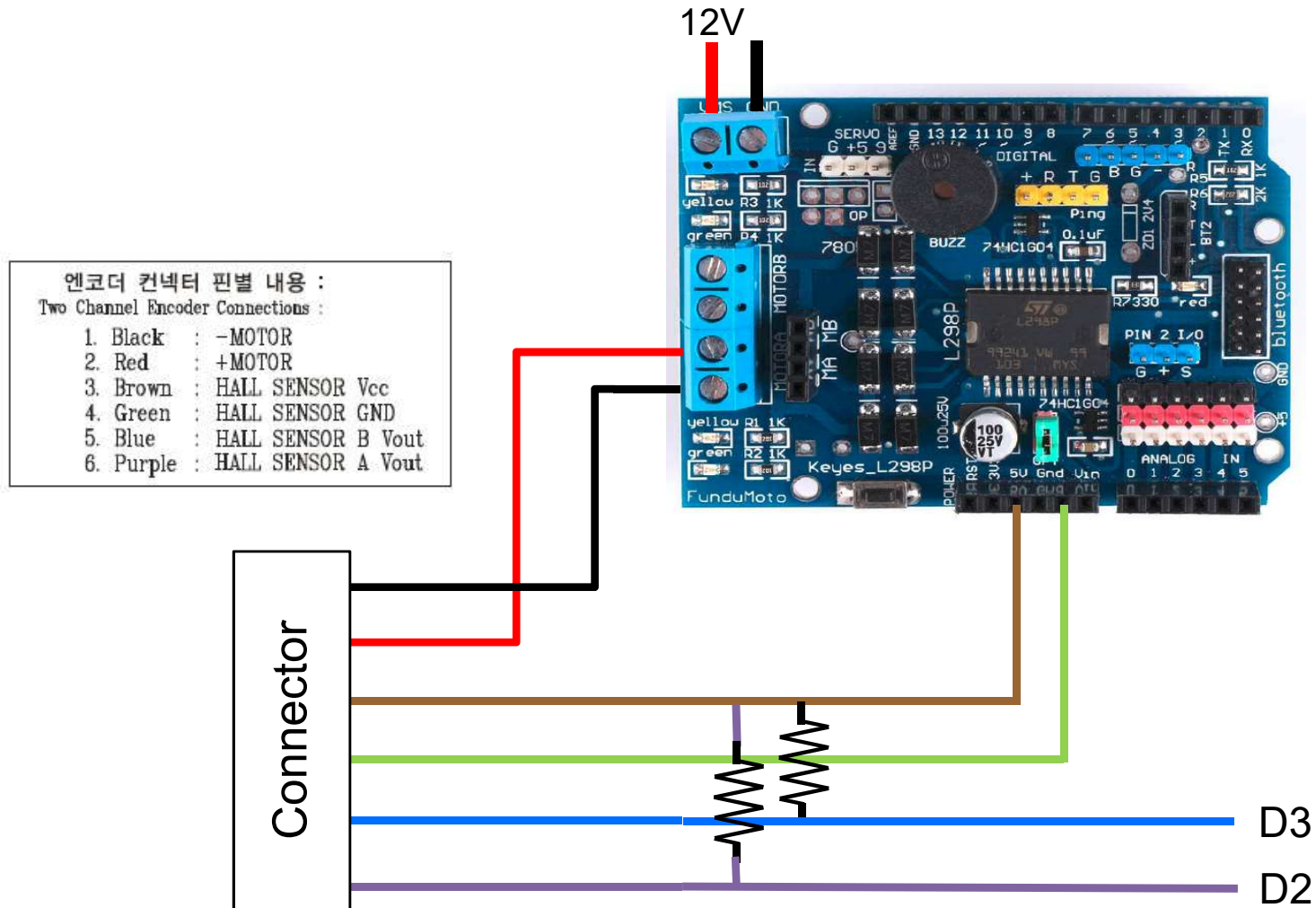
void setup() {
  pinMode(EA_CHA, INPUT);
  pinMode(EA_CHB, INPUT);
  attachInterrupt(digitalPinToInterrupt(EA_CHA), enAchA_ISR, CHANGE);
  attachInterrupt(digitalPinToInterrupt(EA_CHB), enAchB_ISR, CHANGE);
}
```

```
attachInterrupt(0, enAchA_ISR, CHANGE);
attachInterrupt(1, enAchB_ISR, CHANGE);
```



© 2014 Pearson Education, Inc. or its affiliate(s). All rights reserved.

[예제5] 엔코더로 모터의 회전속도 읽기



[예제5] 엔코더로 모터의 회전속도 읽기

```
#define EA_CHA 2
#define EA_CHB 3
#define MA_DIR 12
#define MA_PWM 10
// Encoder
int enAPos = 0;
unsigned long current=0;
unsigned long previous=0;
long dt;
int rpm;

void setup() {
  pinMode(EA_CHA, INPUT);
  pinMode(EA_CHB, INPUT);
  pinMode(MA_DIR, OUTPUT);
  pinMode(MA_PWM, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(EA_CHA), enAchA_ISR,
CHANGE);
  attachInterrupt(digitalPinToInterrupt(EA_CHB), enAchB_ISR,
CHANGE);
  Serial.begin (115200);
}
```

```
void loop(){
  if(enAPos==(13*4)){ //13pulse*4=52pulse
    current=micros();
    dt=current-previous; //us
    rpm = 2000000/dt; //1000000*(1/30)*(60)=2000000
    Serial.print("rpm:");
    Serial.println(rpm);
    enAPos=0;
    previous=current;
  }
  digitalWrite(MA_DIR, HIGH);
  analogWrite(MA_PWM, 100);
}
```

[예제5] 엔코더로 모터의 회전속도 읽기

```
void enAchA_ISR(){
    // a low-to-high edge on channel A
    if(digitalRead(EA_CHA) == HIGH) {
        if(digitalRead(EA_CHB) == LOW) enAPos = enAPos - 1;
        else enAPos = enAPos + 1;
    }

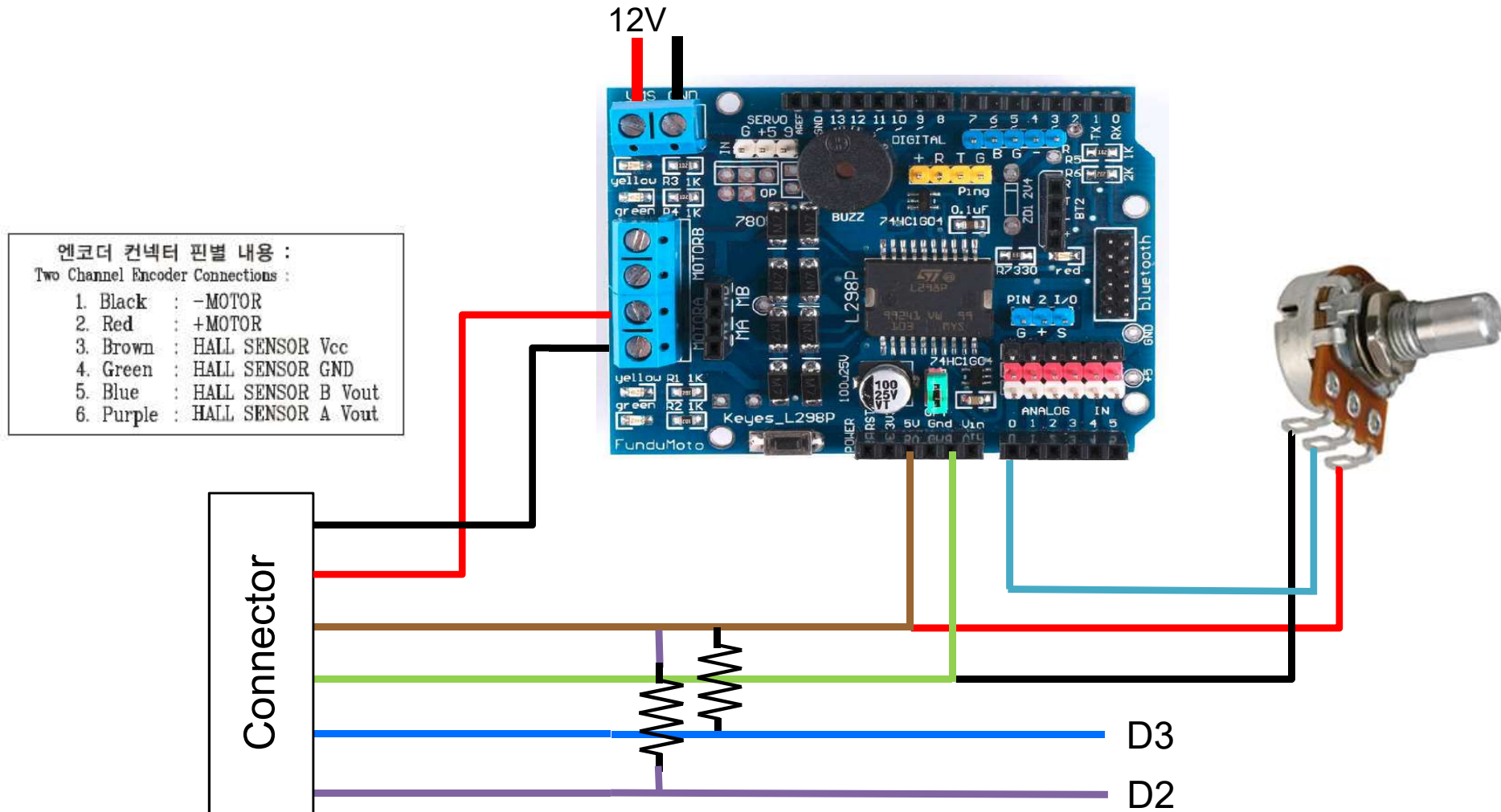
    // must be a high-to-low edge on channel A
    else {
        if(digitalRead(EA_CHB) == HIGH) enAPos = enAPos - 1;
        else enAPos = enAPos + 1;
    }
}
```

```
void enAchB_ISR(){
    // a low-to-high edge on channel B
    if(digitalRead(EA_CHB) == HIGH) {
        if(digitalRead(EA_CHA) == HIGH) enAPos = enAPos - 1;
        else enAPos = enAPos + 1;
    }

    // must be a high-to-low edge on channel B
    else {
        if(digitalRead(EA_CHA) == LOW) enAPos = enAPos - 1;
        else enAPos = enAPos + 1;
    }
}
```


Encoder

[예제6] 가변저항을 이용한 DC Motor 속도조절 값 출력



[예제6] 가변저항을 이용한 DC Motor 속도조절 값 출력

```
#define EA_CHA 2
#define EA_CHB 3
#define MA_DIR 12
#define MA_PWM 10
#define POT_PIN A0
// Encoder
int enAPos = 0;
unsigned long current=0;
unsigned long previous=0;
long dt;
int rpm;
int potValue;
int velocity;

void setup() {
  pinMode(EA_CHA, INPUT);
  pinMode(EA_CHB, INPUT);
  pinMode(POT_PIN, INPUT);
  pinMode(MA_DIR, OUTPUT);
  pinMode(MA_PWM, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(EA_CHA), enAchA_ISR, CHANGE);
  attachInterrupt(digitalPinToInterrupt(EA_CHB), enAchB_ISR, CHANGE);
  Serial.begin(115200);
}
```

```
void loop(){
  potValue = analogRead(POT_PIN);
  velocity = map(potValue, 0, 1023, 0, 255);

  if(enAPos==(13*4)){ //13pulse*4=52pulse
    current=micros();
    dt=current-previous; //us
    rpm = 2000000/dt; //1000000*(1/30)*(60)=2000000
    Serial.print("VEL:");
    Serial.print(velocity);
    Serial.print(",      RPM:");
    Serial.println(rpm);
    enAPos=0;
    previous=current;
  }

  digitalWrite(MA_DIR, HIGH);
  analogWrite(MA_PWM, velocity);
}

void enAchA_ISR(){...}
void enAchB_ISR(){...}
```