

사물인터넷

10주차 - 아날로그 IO 인터페이스 센서

01. 조도 센서 모듈

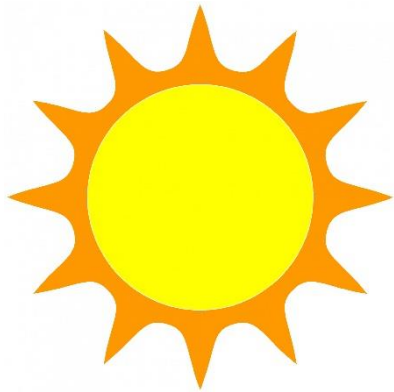
조도센서란?

- 빛의 세기에 감지하는 센서 이다.
- Cds 센서라고도 한다

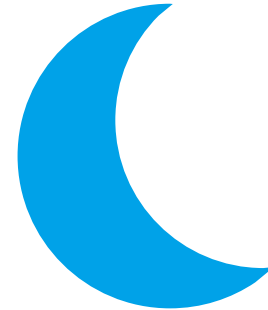
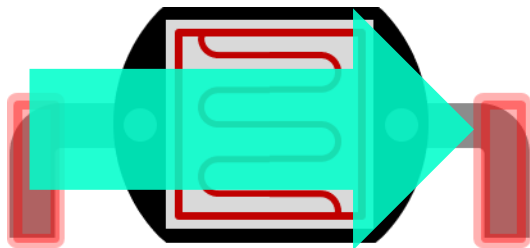


그림3. 황화 카드뮴 셀 CDS

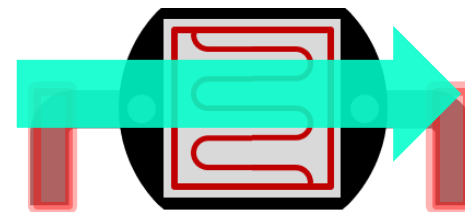
조도센서 특성



빛의 세기가 강할 경우 cds 저항이 낮아져 많은 전류가 흐르게 된다.



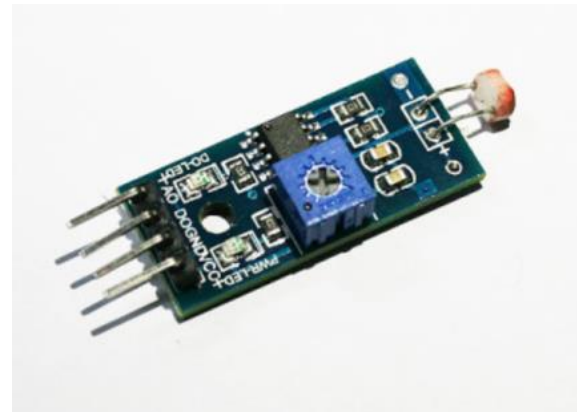
빛의 세기가 약할 경우 cds 저항이 높아져 적은 전류가 흐르게 된다.



조도 센서 모듈 소개

■ 조도 센서 모듈

- ❖ 빛 센서를 이용해서 밝기를 측정.
- ❖ 빛의 세기를 아날로그 전압으로 출력한다.
- ❖ 가변 저항을 이용해서 감도를 조절할 수 있다.
- ❖ 모듈은 센서에 감지된 전압을 0~5V로 출력을 하기위해 내부에 증폭회로(앰프)가 들어 있다.



아날로그 입력 함수 소개.

■ analogRead(pin)

- ❖ 아두이노 내부에 ADC를 이용하여 아날로그 값을 읽어 온다.
- ❖ 0~5V의 입력 전압을 0~1023 사이의 값으로 표현한다.

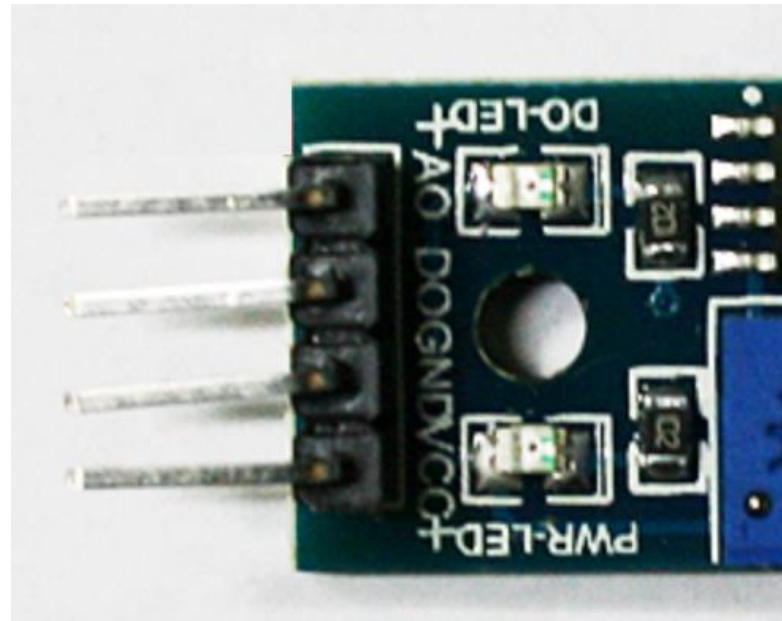
■ map(value, fromLow, fromHigh, toLow, toHigh)

- ❖ 최대/최소 값을 비율로 하여 맞추어 준다.

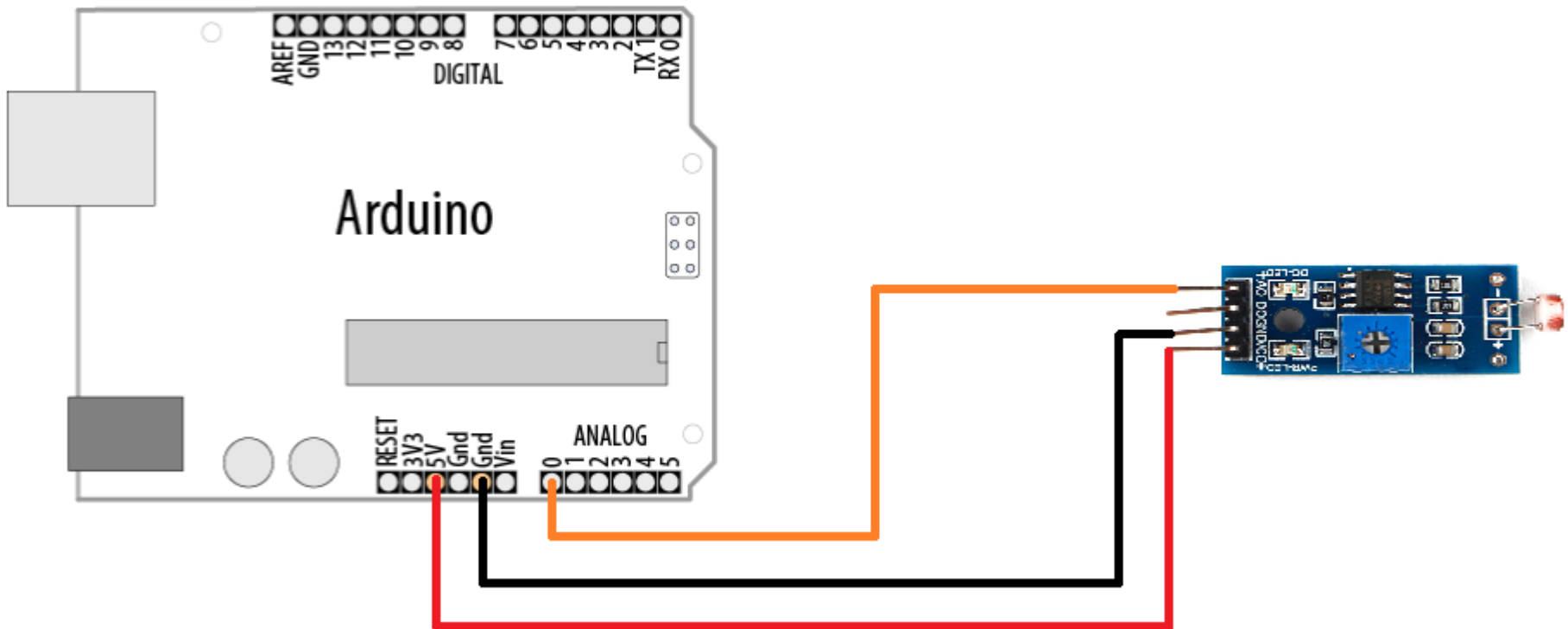
조도 센서 모듈 연결하기

■ 다음과 같이 연결

- ❖ VCC – 5V
- ❖ GND – GND
- ❖ AO – A0



동작 예시 1 – 조도 센서 연결



동작 예시 1

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  delay(1000);
}
```

동작 예시 1 – 동작 결과 확인

- 업로드 후 시리얼 모니터 확인
- 센서를 손으로 가렸다 떼었다 하면서 값의 변화 확인

Serial Monitor Window (/dev/ttyUSB0)

```

12:43:32.218 -> 136
12:43:33.244 -> 116
12:43:34.237 -> 121
12:43:35.264 -> 585
12:43:36.257 -> 533
12:43:37.251 -> 541
12:43:38.278 -> 136
12:43:39.276 -> 123
12:43:40.269 -> 558
12:43:41.295 -> 560
12:43:42.289 -> 530
12:43:43.282 -> 532
12:43:44.309 -> 131
12:43:45.302 -> 127
12:43:46.295 -> 114
12:43:47.322 -> 131
  
```

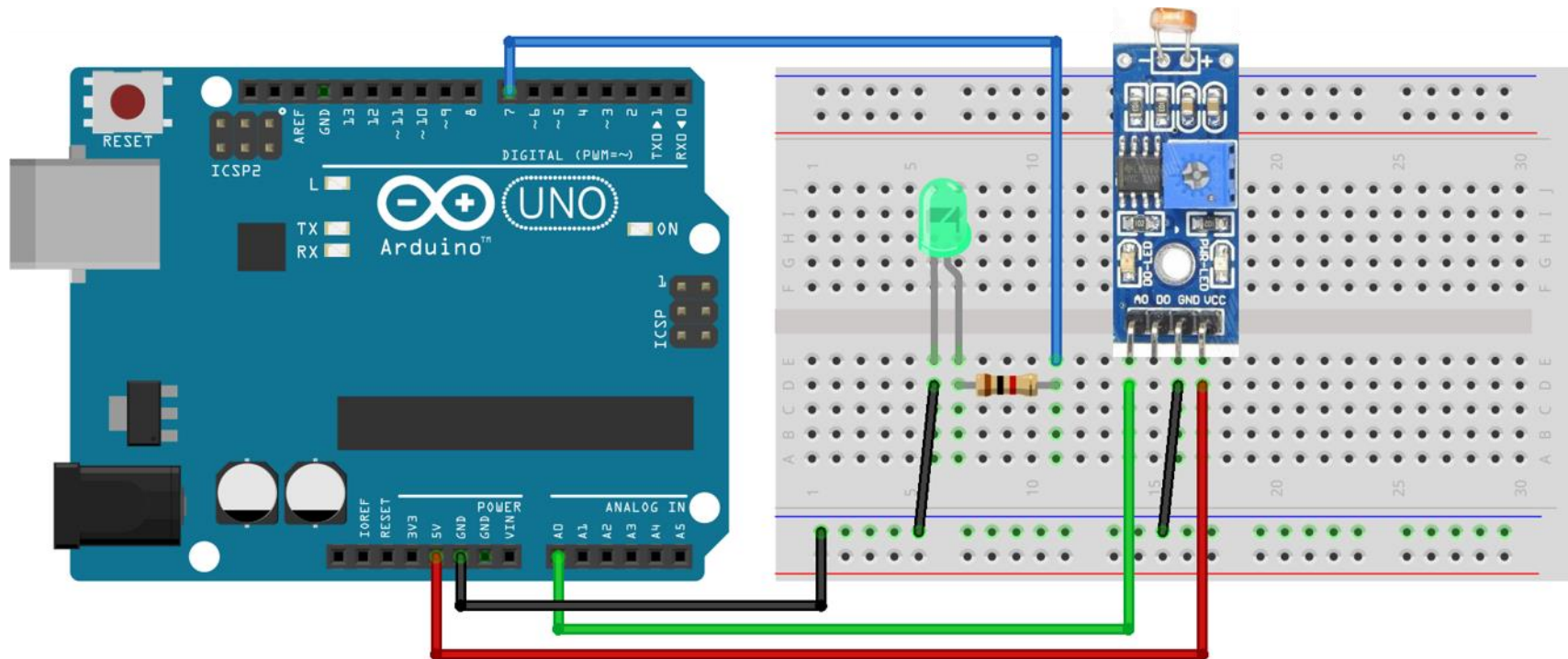
센서를 가렸을 때 결과 값

전송

자동 스크롤, 타임스탬프 표시, line ending 없음, 9600 보드레이트, 출력 지우기

동작 예시 2 – 조도센서로 LED 제어하기

■ 회로도



동작 예시 2 – 예제 코드

```
#define LED_PIN 7

void setup()
{
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
}

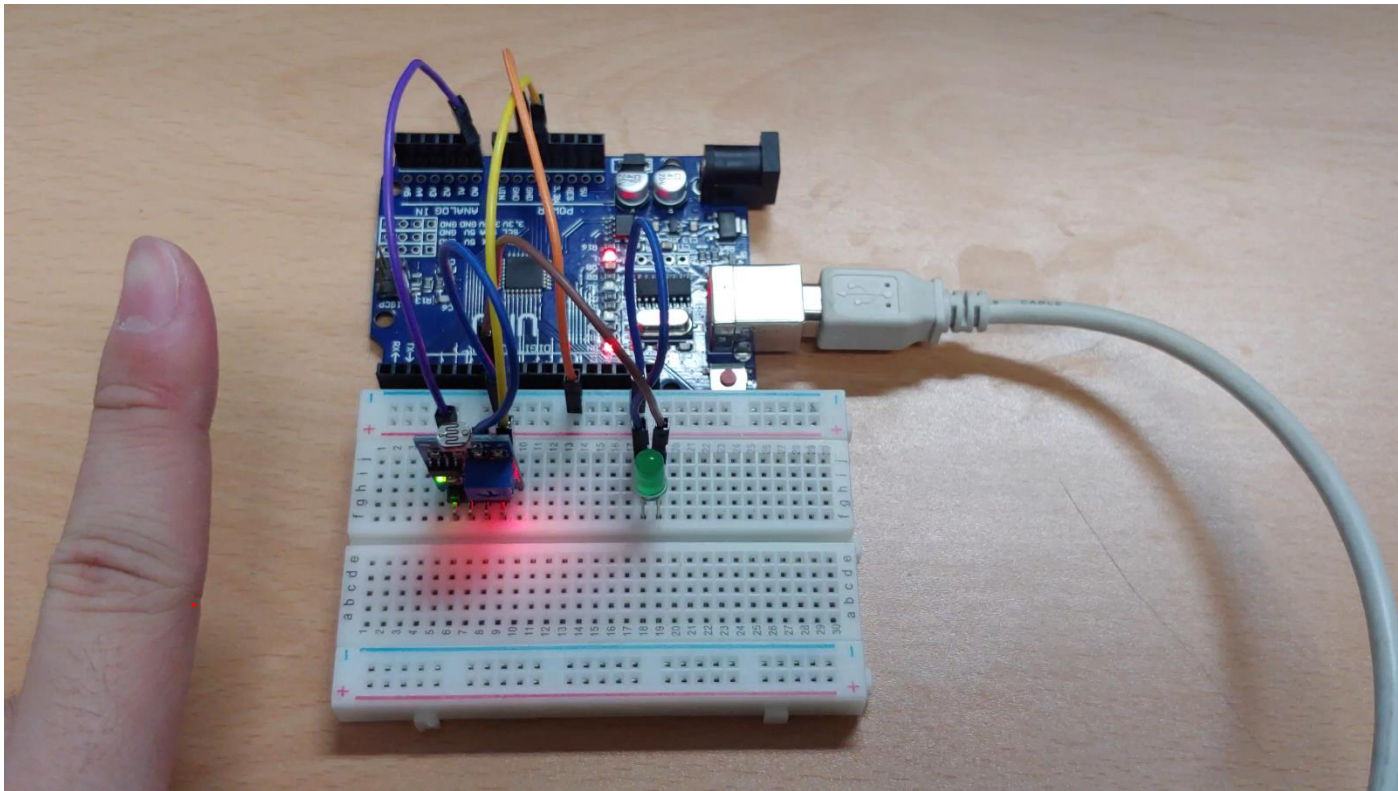
void loop()
{
  int luminance = analogRead(A0);
  Serial.println(luminance);

  if(luminance >= 400)
  {
    digitalWrite(LED_PIN, HIGH);
  }
  else
  {
    digitalWrite(LED_PIN, LOW);
  }

  delay(1000);
}
```

동작 예시 2 - 조도 센서로 LED 동작 확인

- 업로드 후 시리얼 모니터 확인
- 센서를 가렸다 떼었다 하면서 LED의 변화 확인



조도 단위

■ 조도의 단위는 LUX 이다.

한국 표준 조도

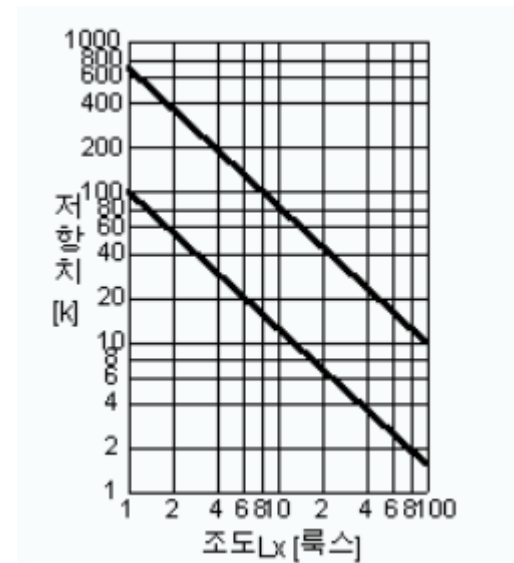
럭스(LUX)	빛 종류	공간/ 작업
1	촛불 1개	
5	보름달이 비추는 빛	
70	10W 전구식 형광등 1개	침실, 현관, 복도
100	15W 전구식 형광등 1개	욕실
150	20W 전구식 형광등 1개	TV보기, 대화, 놀이
200		세탁, 조리, 화장, 식사
300		일반 사무, 노부모 방
400~500		공부, 독서
1000		재봉, 수예

단위: 럭스=촛불 1개 불빛이
1m 떨어진 1m²의 면에 골고루
비추는 빛의 양

[자료: 표준협회]

조도 센서로 조도 표현하기.

- Cds 는 여러 종류가 존재 한다.
- 종류에 따라 전압 특성이 다르다.
- 정확한 자료는 데이터 시트에 기재 되어 있다.



02. PWM을 이용한 멜로디 재생하기

아두이노 간단한 소리 내기

- 아두이노에 부저 (버저, buzzer)를 사용하여 간단한 소리를 낼 수 있다
- 부저의 원리는 단순하더라도 2만Hz이상의 초음파를 출력할 수 있기 때문에 다양하게 사용이 가능하다
- 옥타브 음계에 따라 부저에게 신호를 준다면 그에 맞는 옥타브 음계가 출력된다

능동 부저와 수동 부저의 차이

■ 능동 부저

- ❖ 회로가 내장되어 있어 전류가 흐르면 알아서 소리를 출력한다.
- ❖ LED 켤 때와 코드가 같음.
- ❖ 경보기, 알람 등 단일음에 적합.

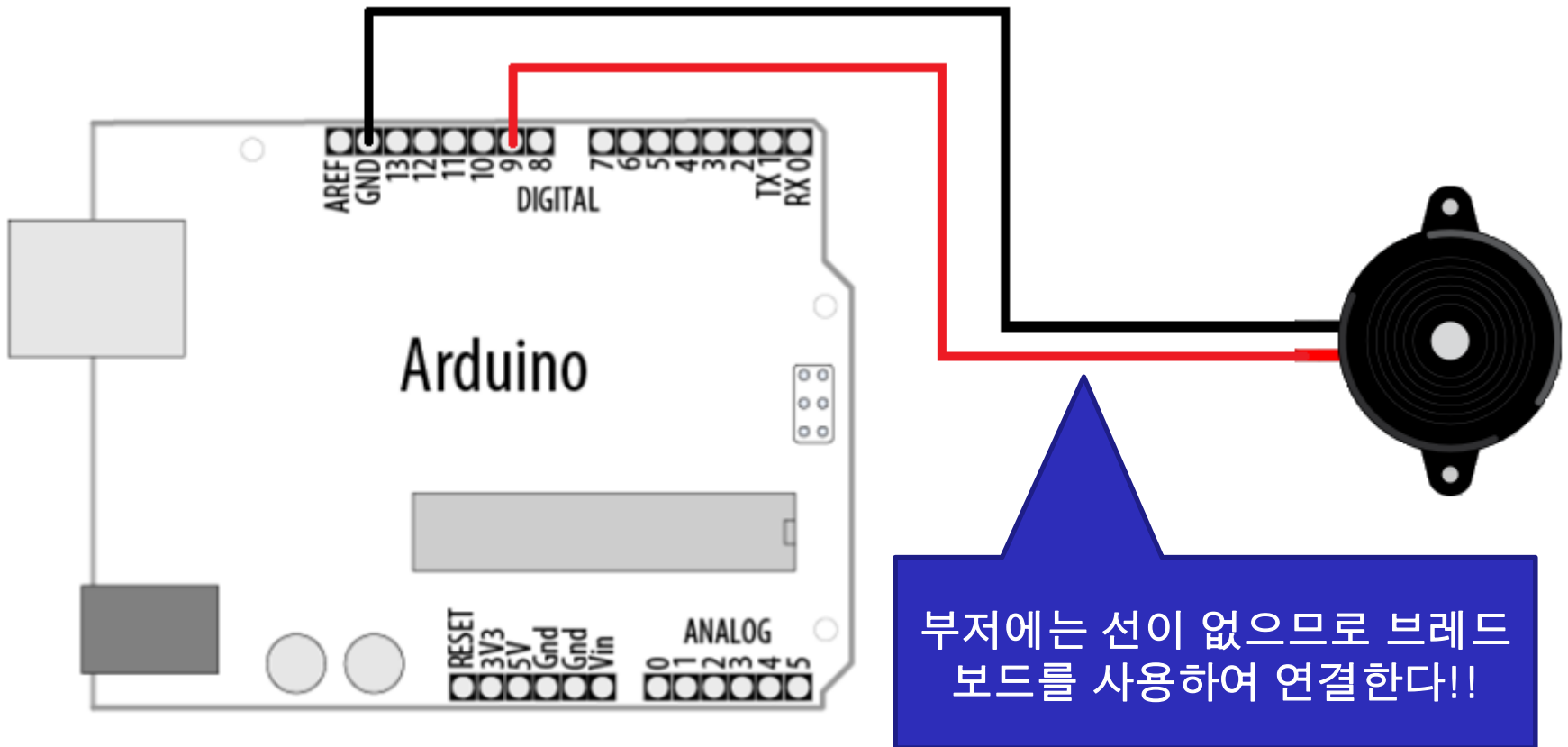
■ 수동 부저

- ❖ 주파수를 활용하여 음계를 출력한다.
- ❖ `tone()` 함수를 이용하여 출력하고자 하는 소리를 출력한다.
- ❖ 다양한 음을 내는 멜로디에 적합.

옥타브 음계 (주파수)

옥타브	1	2	3	4	5
C	33	65	131	262	523
C#	35	69	139	277	554
D	37	73	147	294	587
D#	39	78	156	311	622
E	41	82	165	330	659
F	44	87	175	349	698
F#	46	93	185	370	740
G	49	98	196	392	784
G#	52	104	208	415	831
A	55	110	220	440	880
A#	58	117	233	466	932
B	62	123	247	494	968

연결도



부저 예제

```
const int speakerPin = 9;
char noteNames[] = {'C','D','E','F','G','a','b'};
unsigned int frequencies[] =
{262,294,330,349,392,440,494};
const byte noteCount = sizeof(noteNames);

// 공백은 쉼표를 의미한다.
char score[] =
"CCGGaaGFFEEDDC GGFFEEDGGFFEED CCGGaaGFFEEDDC
";
const byte scoreLen = sizeof(score);

void setup()
{
}
```

부저 예제

```
void loop()
{
  for (int i = 0; i < scoreLen; i++)
  {
    int duration = 333;
    playNote(score[i], duration);
  }

  delay(4000);
}
void playNote(char note, int duration)
{
  for (int i = 0; i < noteCount; i++)
  {
```

부저 예제

```
if (noteNames[i] == note)
    tone(speakerPin, frequencies[i], duration);
}
delay(duration);
}
```

score[] 에 정의 되어진 데로 아
두이노는 소리를 출력한다!

코드 분석

- noteNames는 악보에 있는 음표를 식별하는 문자 배열이다
- 이 배열의 각 항목은 notes배열에 정의되어 있는 주파수와 연결되어 있다. 예를 들어 음표 C의 주파수는 262Hz이다
- score는 재생할 음표가 들어 있는 배열이다
- score 배열의 각 문자가 noteNames 배열에 있는 문자 중 하나와 일치하면 해당 음표가 재생된다

과제 – “나비야” 재생 코드를 만들어 보자

- “부저 예제” 를 수정하여 아래의 나비야 게으름을 넣어 재생되는 코드를 제출하시오

나비야 나비야 이리날라 오너라
솔미미 파레레 도레미파 솔솔솔

호랑나비 흰나비 춤을추며 오너라
솔미미미 파레레 도미솔솔 미미미

봄바람에 꽃잎이 방긋방긋 웃으며
레레레레 레미파 미미미미 미파솔

참새도 짹짹 노래하며 춤춘다
솔미미 파레레 도미솔솔 미미미

101111000101001010011000101001111010101010111110011111010100001111100001000010101011001111101010000111110000100001010101111100111110101000011111000010000101011001111101010000111110000100001010110011111010100001111100001000010101010000101

MOTOR

■ 전기로 움직이는 모터

- ❖ 전기로 움직이는 모터는 전기 에너지를 받아 운동 에너지로 바꾸는 기계이다.
- ❖ 이것을 ‘전동기’ 라고도 부른다.

■ 모터의 종류

- ❖ 모터의 종류는 전원의 방식에 따라 두 가지로 나뉜다.
 - DC를 받아 운동하는 직류 모터
 - AC를 받아 운동하는 교류 모터

MOTOR – 다양한 모터

■ 기어드 모터 (Geared Motor)

- ❖ 기어드 모터는 회전이 빠른 모터에 기어 비율을 높여 속도를 늦추는 대신에 힘을 강하게 한다.
- ❖ 모터의 속도는 힘의 반비례하기 때문이다.
- ❖ AC, DC 모두에도 사용되지만, 완구용 등의 상품에는 DC모터가 사용된다.



기어드모터



미니카에 쓰인 DC모터

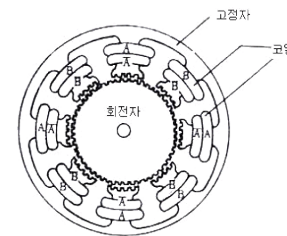
MOTOR – 다양한 모터

■ 스텝 모터

- ❖ 여러 개의 전자석이 금속을 중심으로 배치되어 있다.
- ❖ 별도의 드라이버가 필요하며, 정밀한 동작이 가능하다.
- ❖ 한번에 회전할 수 있는 각도가 정해져 있으며, 이 것을 Step 이라고 한다.
- ❖ DC 모터에 비해 가격이 높다.



스텝 모터



복합형 스텝 모터 구조

MOTOR – 다양한 모터

■ 서보 모터

- ❖ 모터의 축을 특정 각도로 회전시키는 경우에 사용하며, 피드백을 받아 정확하게 이동 되었는지 확인 할 수 있는 구조를 가지고 있다.
- ❖ 단순한 스텝모터는 각도를 조절하여 회전시킬 수 있으나 어느 정도 까지 회전 되었는지 확인 할 수 없고, 스텝 모터는 스텝 단위로 회전을 시켜 원하는 각도에 위치 시키지만 서보 모터는 신호에 따라 원하는 각도로 회전된다.



서보모터



서보 모터가 적용된 로봇 팔

MOTOR – 예제

■ 위의 모터 외에도 복합적으로 적용된 모터까지 추가하면 종류는 다양하다

■ DC 모터의 회전

- ❖ DC 모터는 단순히 보면 두 개의 단자의 연결을 통하여 역회전과 정회전을 하게 된다.
- ❖ DC 모터에 걸리는 극성에 따라 방향이 결정된다.

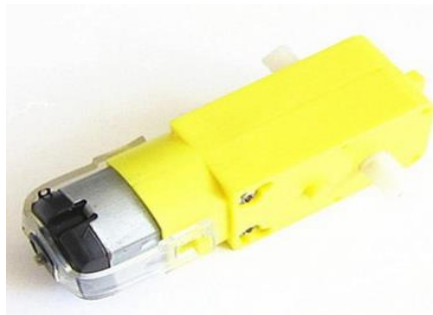
■ 모터 드라이버

- ❖ 모터는 전류를 많이 소모하는 소자여서 GPIO로 공급하기에는 너무 크다.
- ❖ 충분한 전류를 공급하기 위한 방법으로 모터 드라이버라는 IC를 사용한다.

DC MOTOR 소개

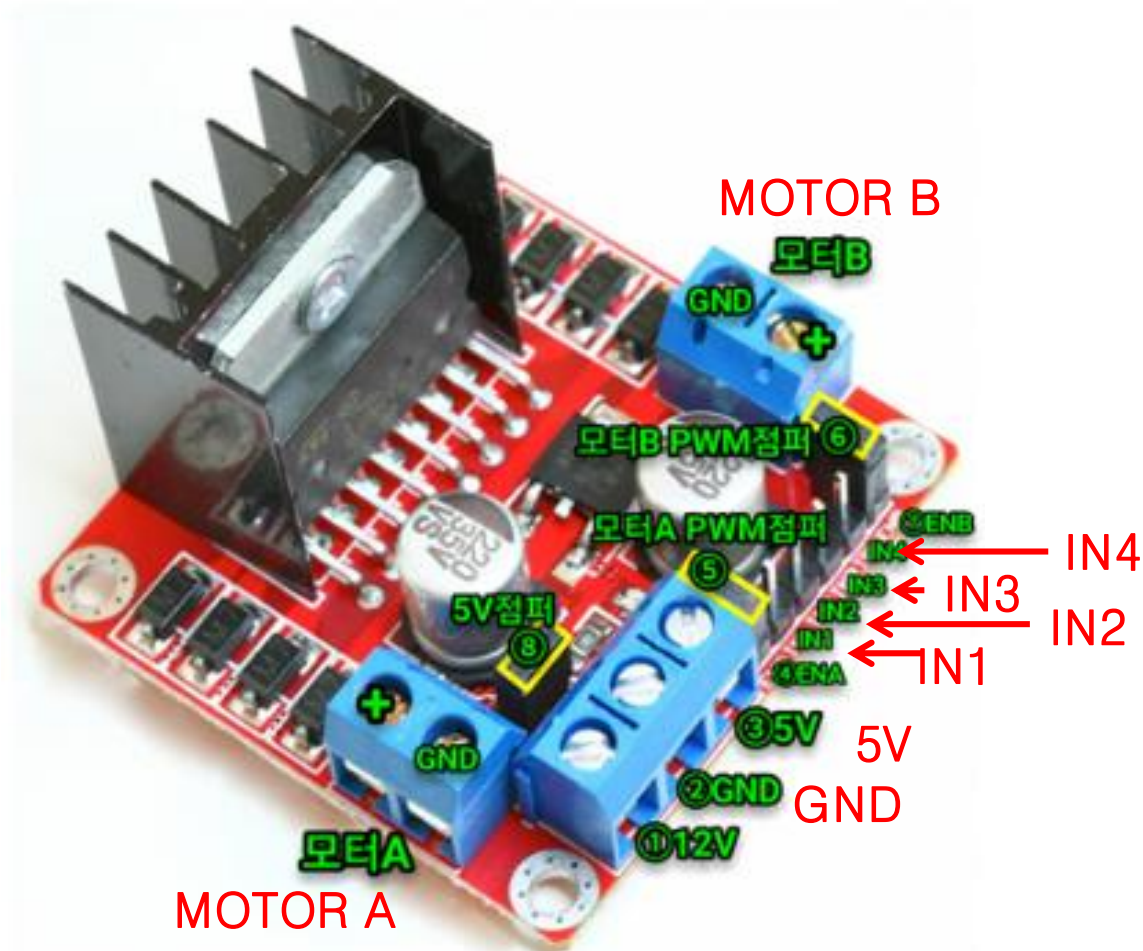
■ DC MOTOR

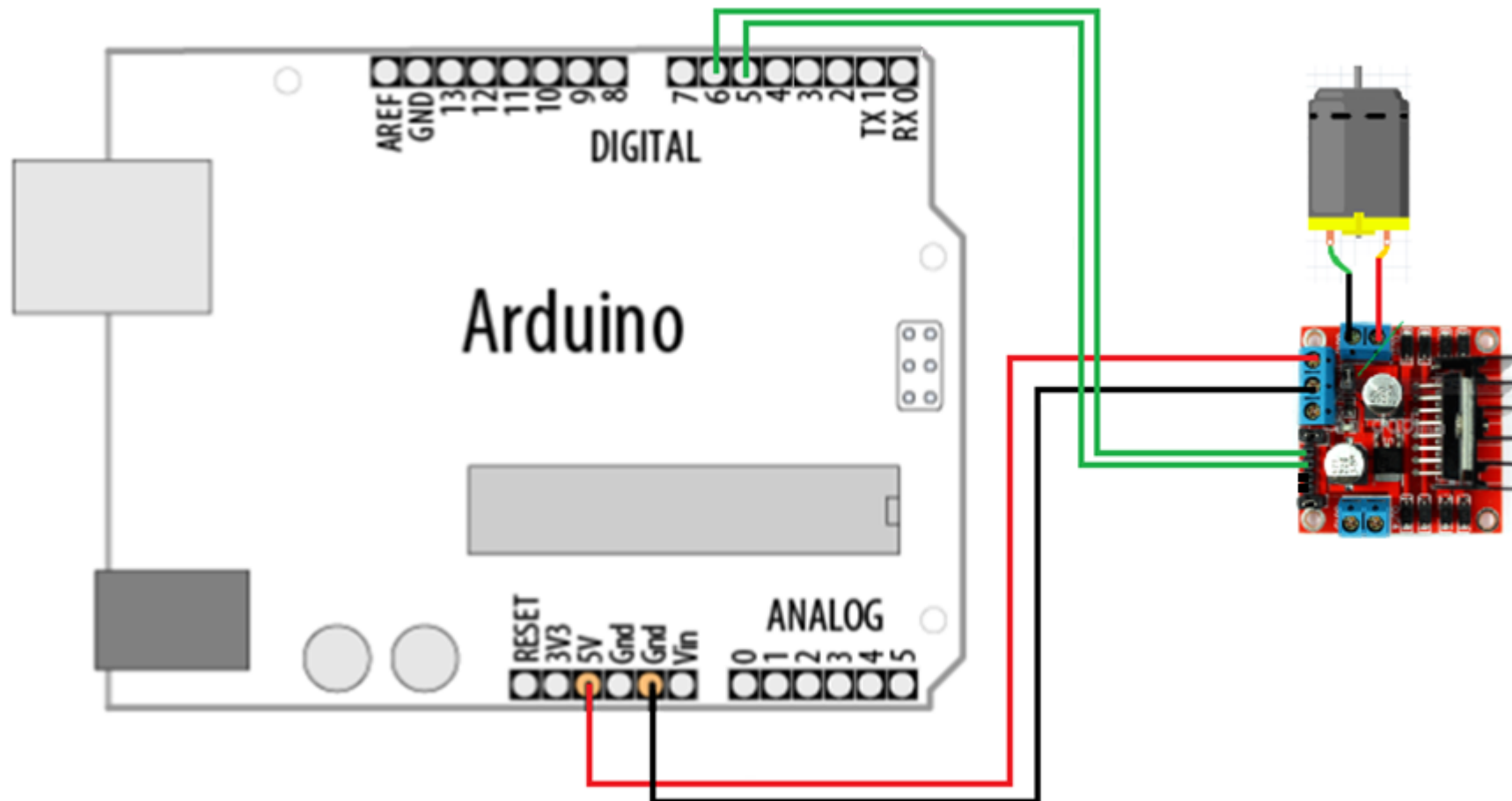
- ❖극성을 반드시 맞춰야 할 필요는 없으나, 정회전과 역회전을 확인하기 위해서는 한쪽을 VCC, 다른 쪽을 GND 기준으로 잡아 정,역회전 구분을 하는 것이 좋다.
- ❖전선으로 납땜 되어있어 보드에 사용이 편하게 되어있다.



DC 모터의 외형

모터 드라이버 모듈





모터 1개 제어 하기 예제

```
const int IN[2] = {6,5};

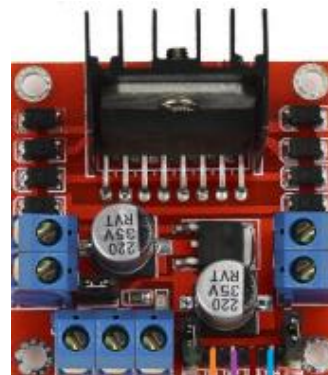
void setup() {
  for(int i=0;i<2;i++){
    pinMode(IN[i], OUTPUT);
  }
}

void loop() {
  pinMode(IN[0],HIGH);
  pinMode(IN[1],LOW);
  delay(1000);
  pinMode(IN[1],LOW);
  pinMode(IN[0],HIGH);
  delay(1000);
}
```

모터 드라이버

■ 모터 드라이버란?

- ❖ 모터 드라이버는 말 그대로 모터를 구동시키기 위한 IC 혹은 모듈을 칭한다.
- ❖ 모터 드라이버를 사용하는 이유는 아두이노의 핀(GPIO)에서 제공하는 공급전류가 부족하기 때문이다.
- ❖ 직접 연결하여 작동시켜보면 전류 부족으로 모터가 돌지 않거나 아두이노가 꺼질 수 있다.
- ❖ 심한 경우는 핀이 손상되거나 하는 등의 문제가 생겨, 해당 핀은 사용할 수 없는 상태가 된다. 보드가 고장 나기도 한다.



모터 드라이버

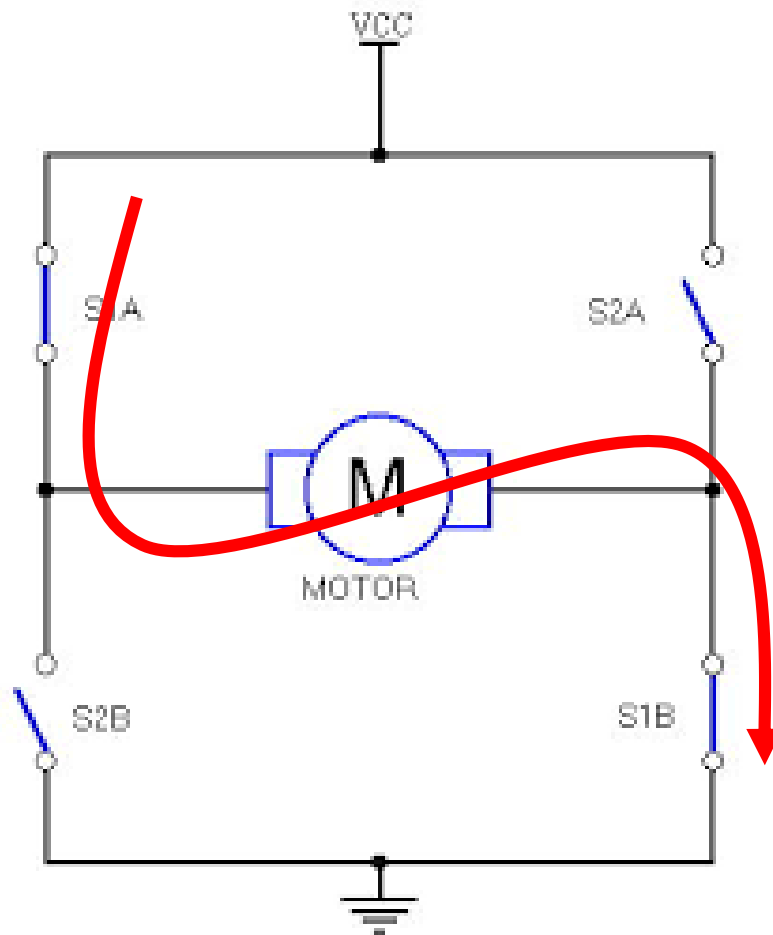
H-브릿지를 사용하여 모터 제어

■ H-브릿지란?

- ❖ 모터 드라이버 중 한 종류로서 큰 전류를 공급해주는 역할도 하지만 방향을 바꿀 수 있게 해주는 회로 장치를 모터 드라이버이다.
- ❖ DC 모터는 2개의 선이 있습니다. +에서 - 또는 -에서 +로 전류가 흐르는 가에 따라 모터의 회전 방향이 바뀐다. GPIO를 사용하여 극성을 바꿀 수 있게 해준다.

■ 결과적으로 H-브릿지를 사용하지 않으면 모터의 방향을 바꿀 수 없다!

방향 제어 방법

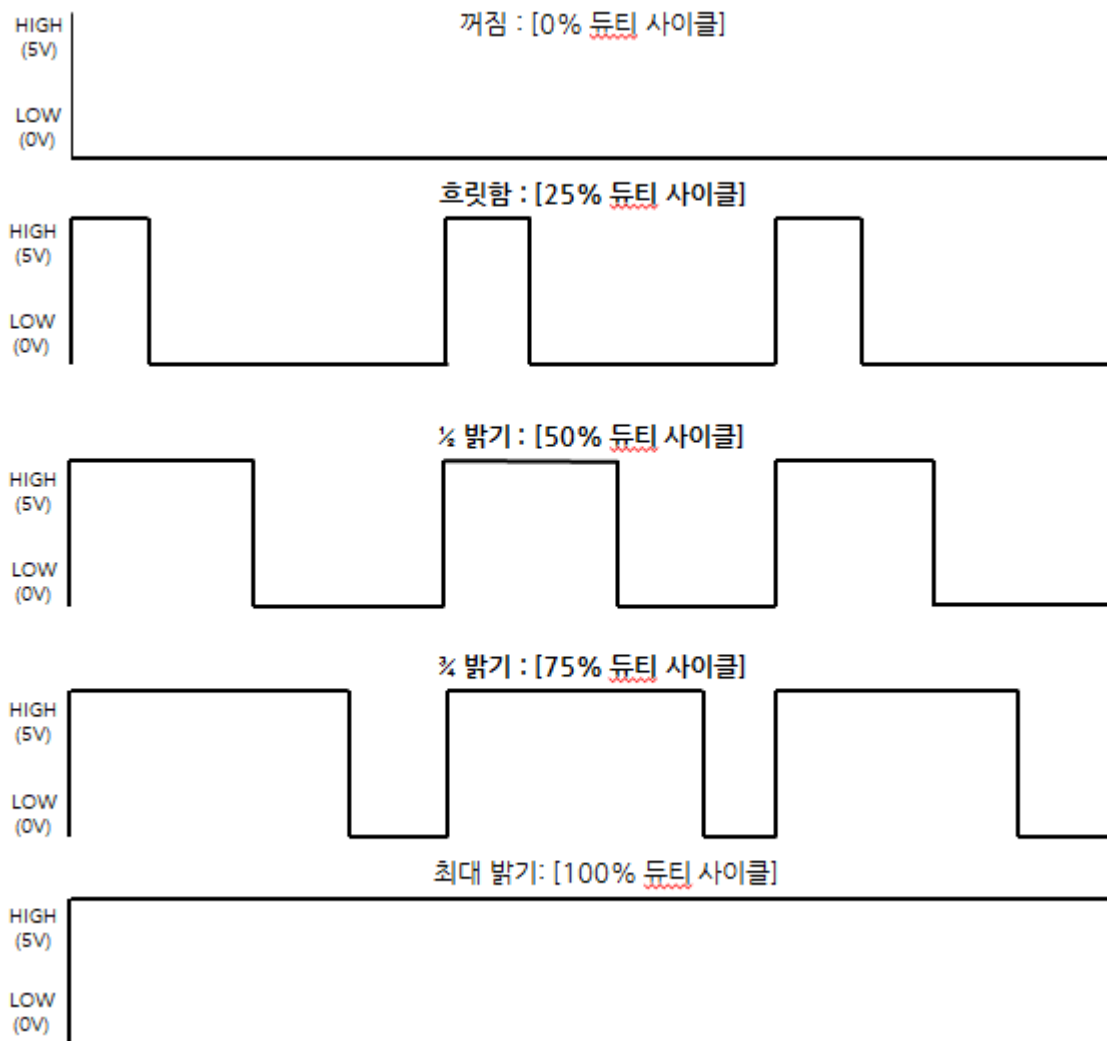


H브릿지를 사용하여 모터 제어

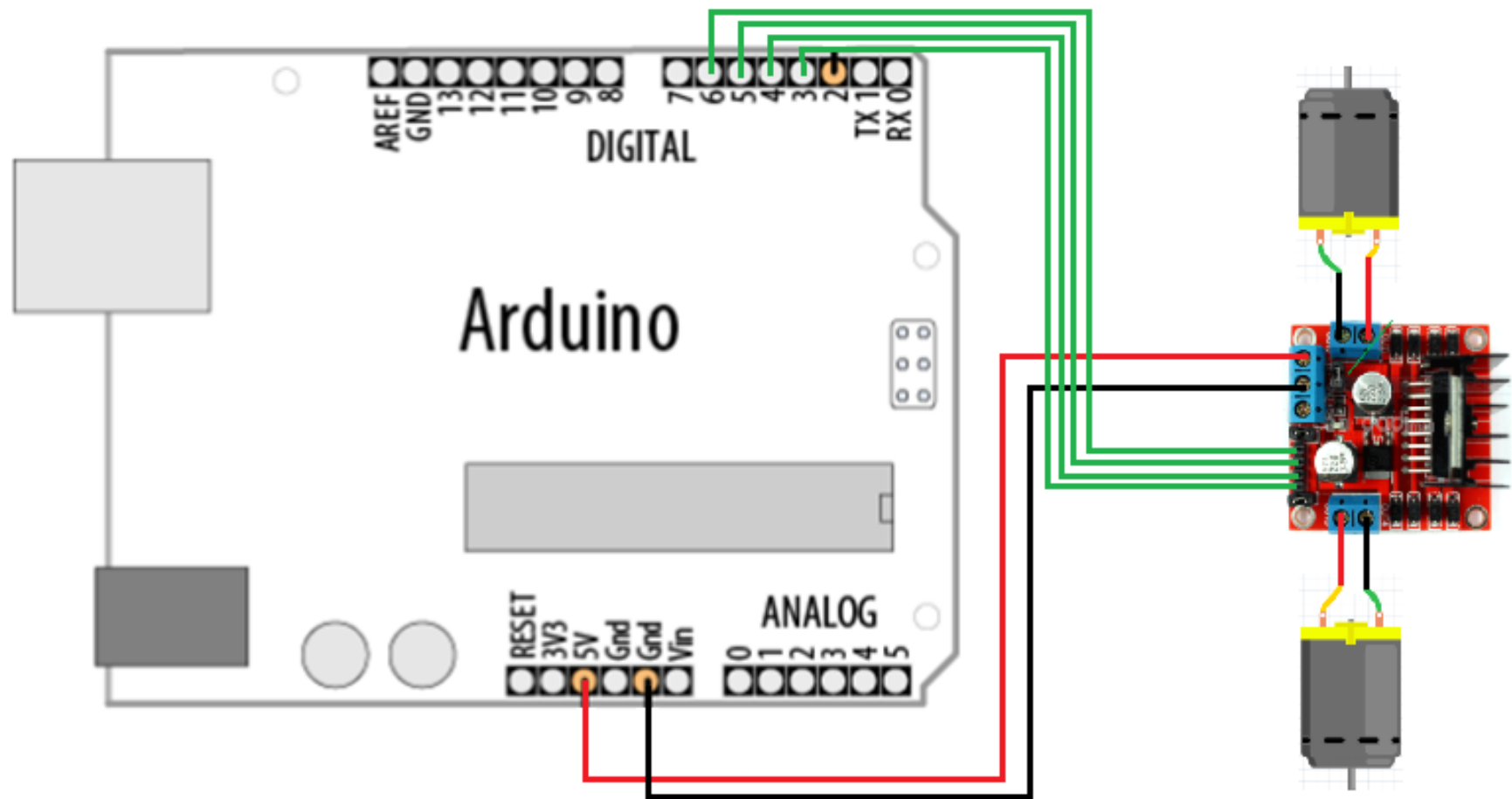
■ 표 8-1 H브릿지 논리표

EN	IN1	IN2	기능
HIGH	LOW	HIGH	시계 방향으로 회전
HIGH	HIGH	LOW	반 시계 방향으로 회전
HIGH	LOW	LOW	모터 정지
HIGH	HIGH	HIGH	모터 정지
LOW	무시	무시	모터 정지

모터 속도 제어



회로도



예제 소스

```
const int IN[4] = {6,5,4,3};

void setup() {
  for(int i=0;i<4;i++){
    pinMode(IN[i], OUTPUT);
  }
}

void loop() {
  STOP(1500);
  Forward(1500);
  Left(1500);
  Right(1500);
  Backward(1500);
}
```

모터의 속도 제어

■ PWM을 이용하여 속도를 제어한다

- ❖ 드라이버는 프로세서에서 작은 전류를 제어하여 모터의 속도를 제어할 수 있게 해주고, 전압의 변화는 전류를 변화 시키기 때문에 아두이노에서는 아날로그 출력 (analogWrite)를 이용하여 속도를 제어할 수 있다.
- ❖ 전압이 커지면 속도가 증가하고, 전압이 작으면 속도가 감소합니다. 전압이 증가하는 것은 전류의 증가를 의미하기 때문입니다. 하지만 DC 전압을 손쉽게 제어하기 힘들기 때문에 PWM (Pulse Width Modulation) 제어한다.

H-브릿지 드라이버의 속도 제어

■ EN 핀을 사용하여 속도를 제어할 수 있다.

- ❖ EN 핀에 PWM 입력을 하여 속도를 제어할 수 있다.
- ❖ 0~255까지 출력되지만 일정 크기까지 모터는 움직이지 않는다. 이는 전압이 너무 낮기 때문이다.

실습 - 스위치를 연결하여 모터를 제어해보자

- 스위치 두 개를 추가한다.
- 각 스위치는 각 모터를 On/Off 한다. 누르면 On, 안누르면 Off 되게 한다.

마무리

- DC 모터란, 고정자로 영구자석을 사용하고, 회전자(전기자)로 코일을 사용하여 구성한 것으로, 전기자에 흐르는 전류의 방향을 전환함으로써 자력의 반발, 흡인력으로 회전력을 생성시키는 모터이다.
- 모터가 구동 되기에는 5V는 작다. 사각 배터리 같은 것을 이용하여 9V를 입력하면 모터는 더 빠르게 구동 될 수 있다.
- 아두이노에 9V 건전지로 전원을 인가하면 5V핀으로 9V가 출력된다.
 - ❖ 단, USB 케이블을 제거하고 연결을 하여야 한다.
- 아두이노는 한번 업로드된 프로그램은 롬에 기억되어 있다.