



라즈베리파이 모터제어

로봇SW 교육원

최상훈(shchoi82@gmail.com)

학습 목표

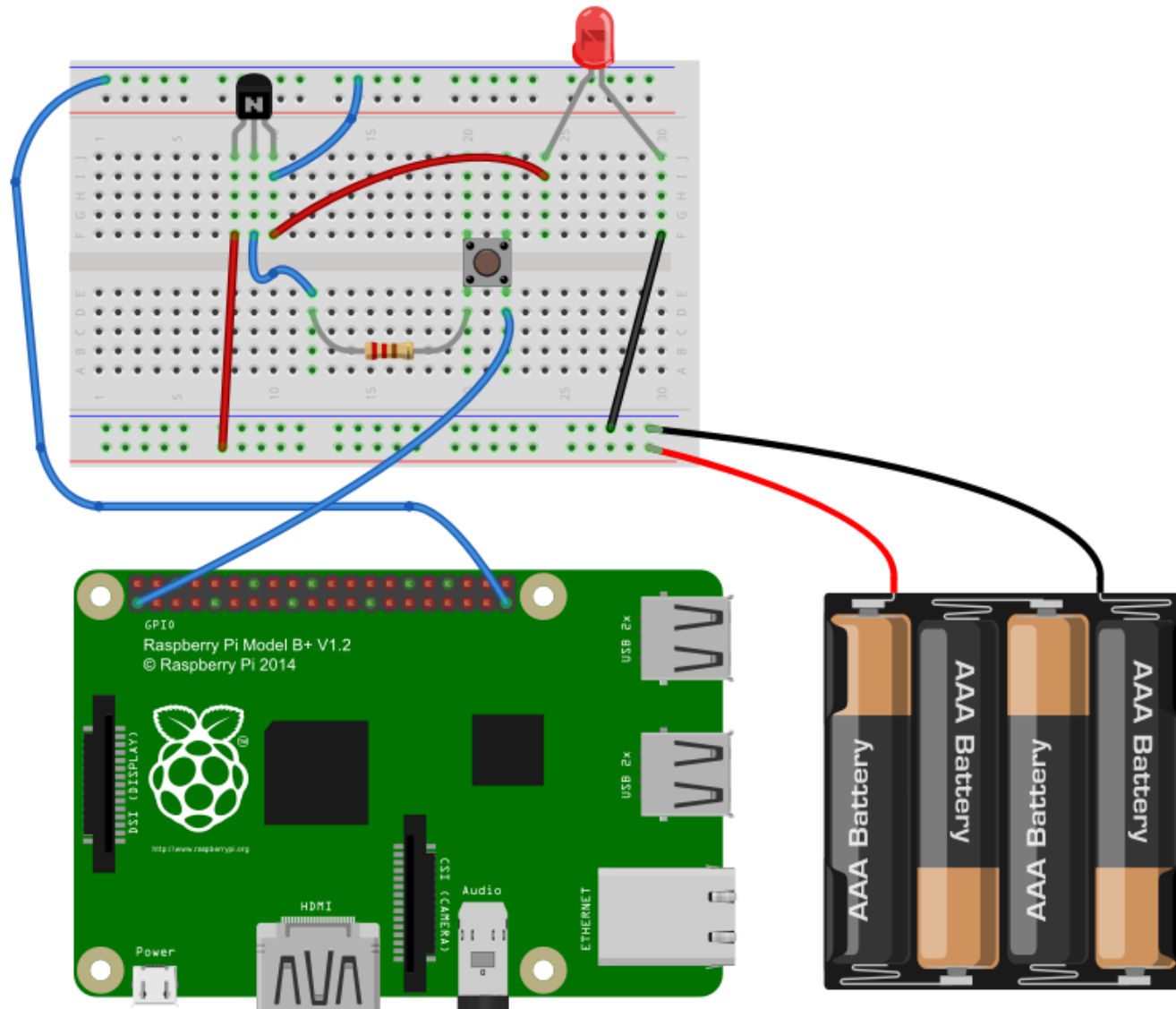
2

- 트랜지스터를 이용한 모터 제어
- 모터드라이버를 이용한 모터 제어

실습1 : 트랜지스터

3

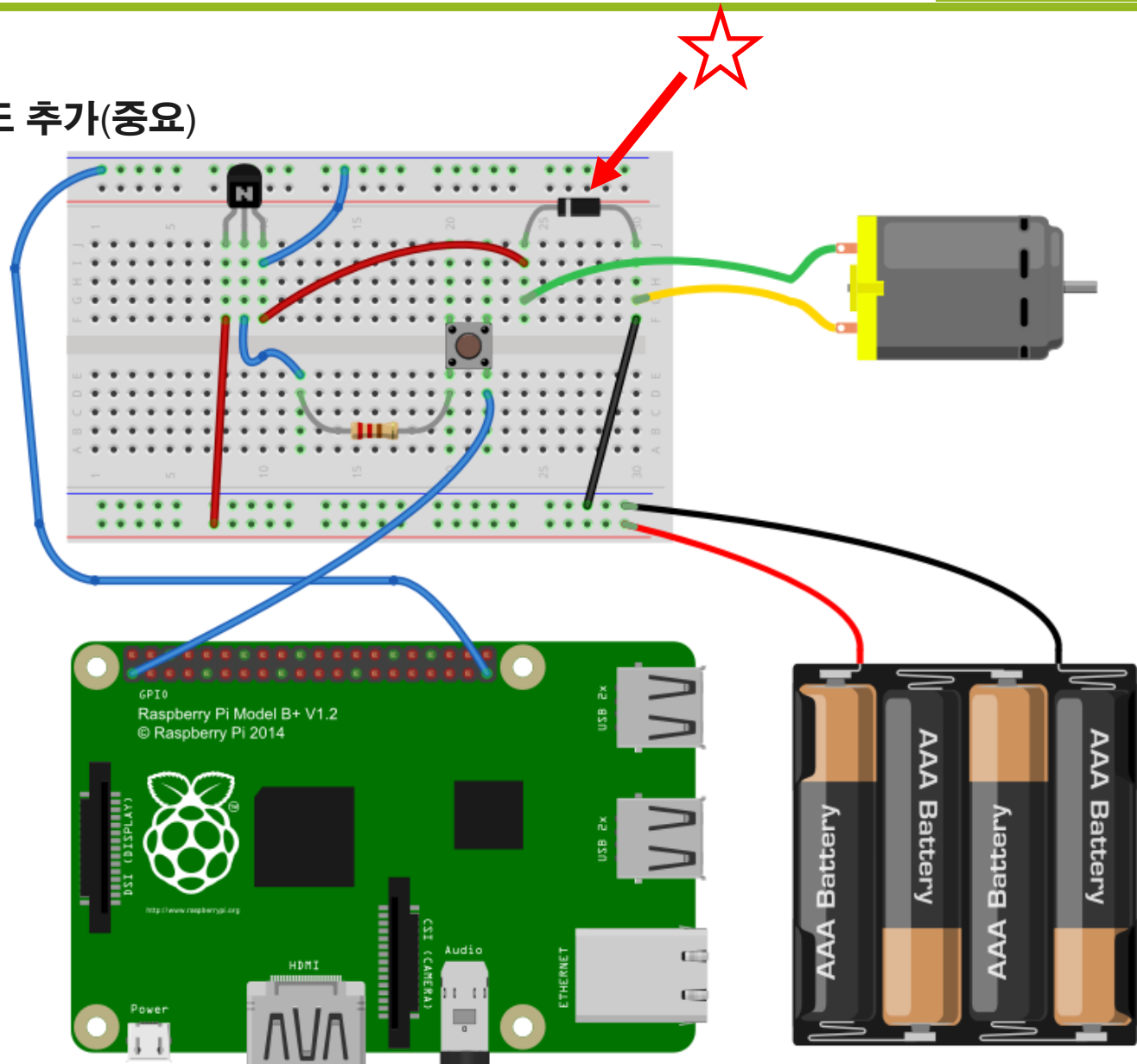
- 구성
 - LED 1개
 - 스위치 1개
 - 트랜지스터 1개
 - 배터리 AA 4개
- 라즈베리파이
 - 3.3v
 - GND
- 스위치 버튼으로
LED On/Off



실습2 : 모터 제어

4

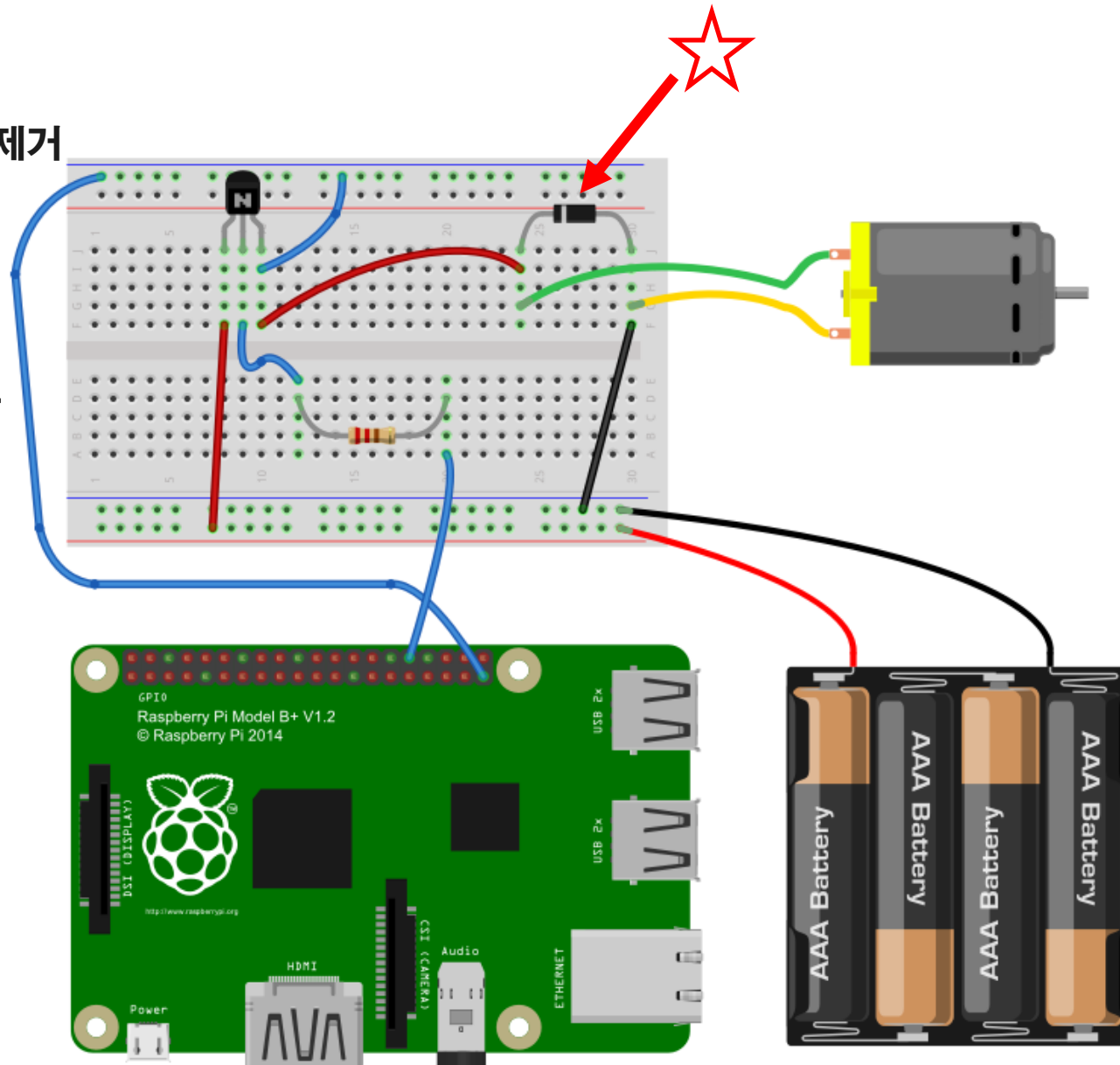
- 구성
 - 모터와 다이오드 추가(중요)
- 라즈베리파이
 - 3.3v
 - GND
- 스위치 버튼으로
모터 On/Off



실습3 : 모터 GPIO 제어

5

- 구성
 - 스위치 제거
 - 3.3v 와이어 연결 제거
- 라즈베리파이
 - GPIO 12 연결
- GPIO 디지털 출력으로
모터 On/Off
- GPIO PWM 출력으로
모터 속도제어



실습3-1 : 모터 On/Off

6

- 파일명 : motor.c

```
#include <stdio.h>
#include <wiringPi.h>

#define MOTOR 12

int main(void)
{
    int i = 0;
    wiringPiSetupGpio();

    pinMode(MOTOR, OUTPUT);

    for(i = 0 ; i < 5 ; i++){
        digitalWrite(MOTOR, HIGH);
        delay(100);
        digitalWrite(MOTOR, LOW);
        delay(100);
    }

    return 0;
}
```

실습3-1 : 모터 On/Off

7

- 컴파일

```
$ gcc -lwiringPi motor.c -o motor
```

- 실행

```
$ sudo ./motor
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

실습3-1 : 모터 On/Off

8

- GPIO 명령 제어

```
$ gpio -g mode 12 out  
$ gpio -g write 12 1  
$ gpio -g write 12 0  
$ gpio -g write 12 1  
$ gpio -g write 12 0
```


실습3-2 : 모터 속도 제어

9

- 파일명 : motor_pwm1.c

```
#include <stdio.h>
#include <wiringPi.h>
#define MOTOR_PWM 12

int main(void)
{
    wiringPiSetupGpio();
    pinMode(MOTOR_PWM, PWM_OUTPUT);

    // 범위 0 ~ 1024
    pwmWrite(MOTOR_PWM, 0);
    delay(1000);
    pwmWrite(MOTOR_PWM, 400);
    delay(1000);
    pwmWrite(MOTOR_PWM, 700);
    delay(1000);
    pwmWrite(MOTOR_PWM, 900);
    delay(1000);
    pwmWrite(MOTOR_PWM, 1024);
    delay(1000);
    pwmWrite(MOTOR_PWM, 0);
    delay(1000);

    return 0;
}
```

실습3-2 : 모터 속도 제어

10

- 컴파일

```
$ gcc -lwiringPi motor_pwm1.c -o motor_pwm1
```

- 실행

```
$ sudo ./motor_pwm1
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

실습3-3 : 모터 속도 제어

11

- 파일명 : motor_pwm2.c

```
#include <stdio.h>
#include <wiringPi.h>

#define LED 12

int main(void)
{
    int i, j;
    wiringPiSetupGpio();
    pinMode(LED, PWM_OUTPUT);

    for(i = 0 ; i <= 1024 ; i++)
    {
        pwmWrite(LED, i);
        delay(10);
        if(i % 100 == 0)
            printf("현재 PWM:%d\n", i);
    }

    pwmWrite(LED, 0);

    return 0;
}
```

실습3-3 : 모터 속도 제어

12

- 컴파일

```
$ gcc -lwiringPi motor_pwm2.c -o motor_pwm2
```

- 실행

```
$ sudo ./motor_pwm2
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

실습3-2 : 모터 속도 제어

13

- GPIO 명령 제어

```
$ gpio -g mode 12 pwm
$ gpio -g pwm 12 1023
$ gpio -g pwm 12 900
$ gpio -g pwm 12 800
$ gpio -g pwm 12 700
$ gpio -g pwm 12 600
$ gpio -g pwm 12 500
$ gpio -g pwm 12 400
$ gpio -g pwm 12 300
$ gpio -g pwm 12 200
$ gpio -g pwm 12 100
$ gpio -g pwm 12 0
```

미션

14

- `motor_pwm2.c`은 모터의 속도를 점점 증가시키다가 정지하는 프로그램이다. 점점 감소하도록 프로그램을 수정하시오.

라즈베리 파이 모터제어

모터 드라이버

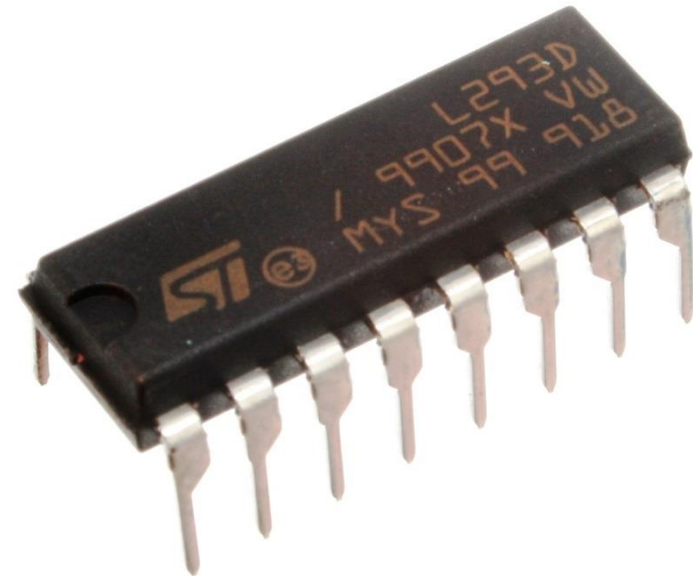
로봇SW 교육원

최상훈(shchoi82@gmail.com)

L293D

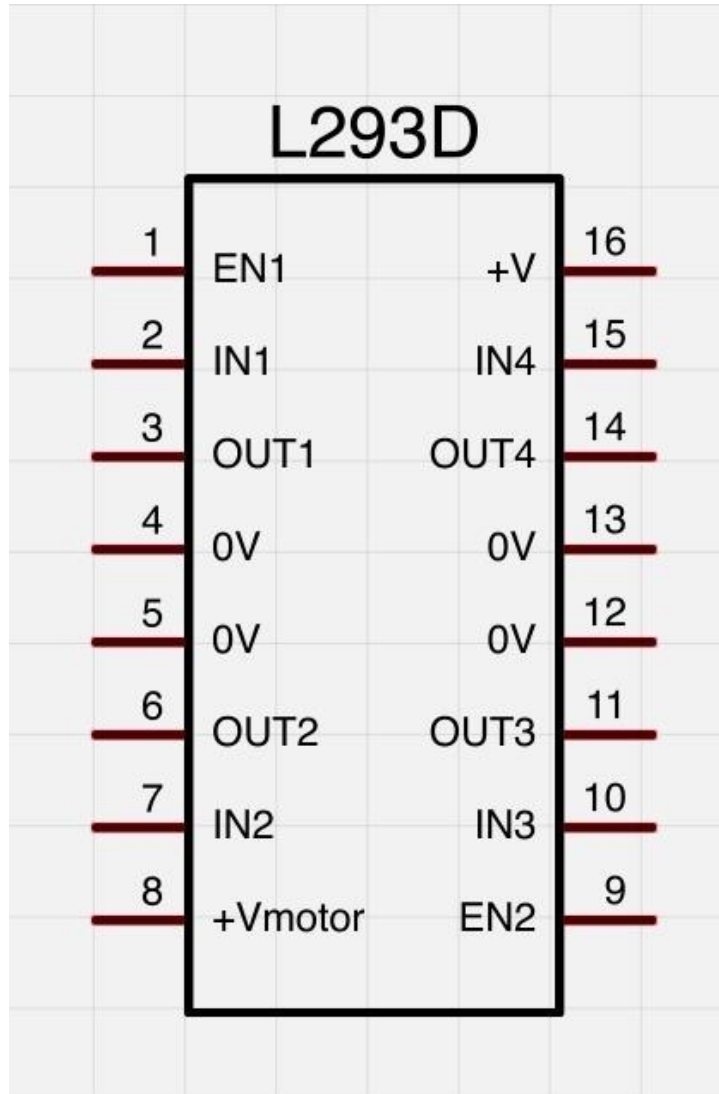
16

- L293D
 - 4.5v ~ 36v 전압 제어 가능
 - 채널당 600mA의 전류 제어 가능
 - OUTPUT 4개
 - INPUT 4개
 - Enable 핀을 통한 PWM제어 가능
 - 역전류 방지를 위한 다이오드 내장
- 모터 드라이버
 - 모터의 속도 제어
 - 모터의 방향 제어



L293D

17

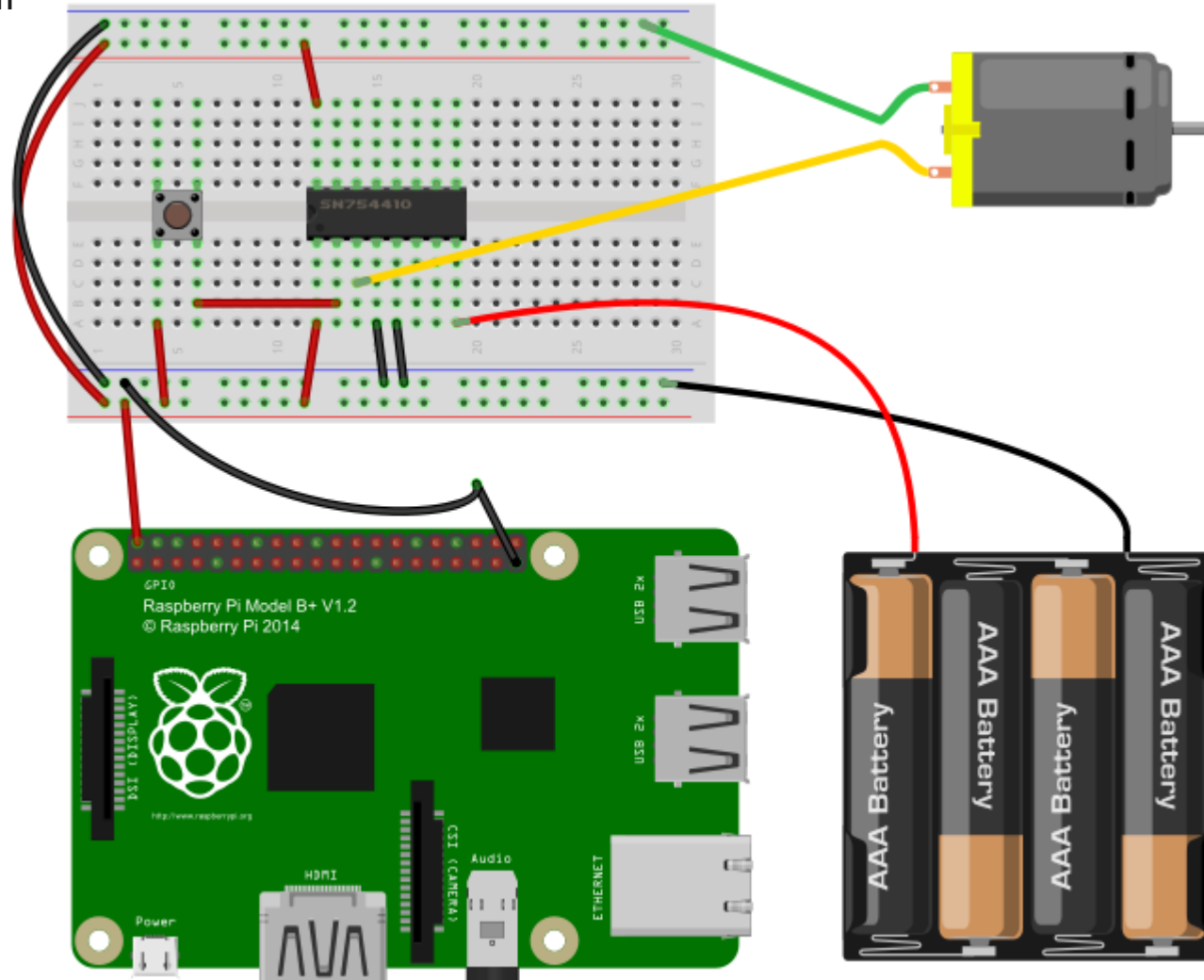


+V	L293D 전원
+Vmotor	OUT _n 전원
EN1	OUT ₁ , OUT ₂ Enable/Disable
EN2	OUT ₃ , OUT ₄ Enable/Disable
IN _n	OUT _n 제어

실습1 : L293D (Input 1 제어 , 단방향)

18

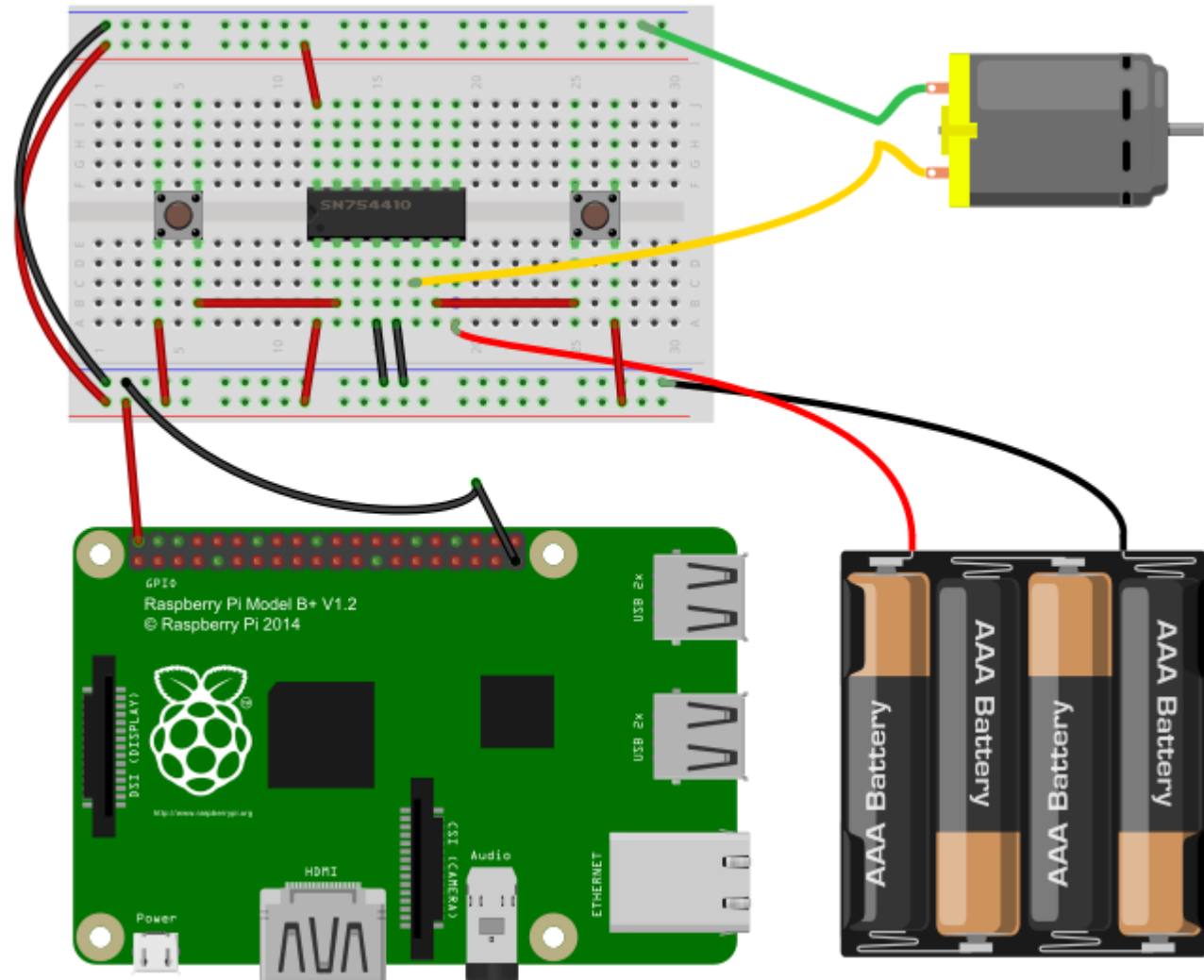
- 스위치 버튼으로 IN₁ 입력
- OUT₁에 연결된 모터 On/Off
- 구성
 - 스위치
 - L293D
 - DC모터
- 라즈베리파이
 - 3.3v
 - GND



실습2 : L293D (Input 2 제어, 단방향)

19

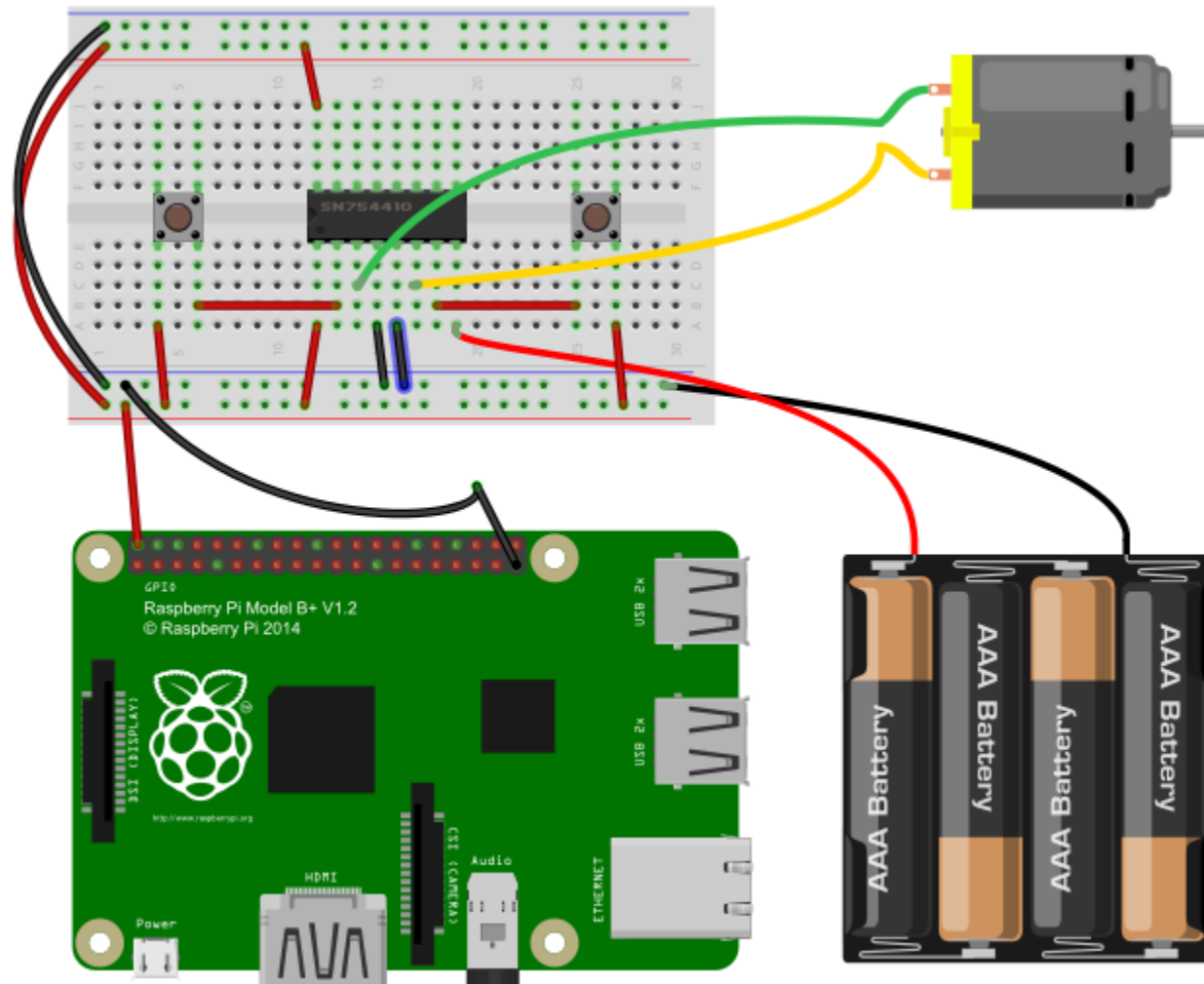
- 스위치 버튼으로 IN₂ 입력
- OUT₂에 연결된 모터 On/Off
- 구성
 - 스위치 2개
 - L293D
 - DC 모터
- 라즈베리파이
 - 3.3v
 - GND



실습3 : L293D (Input1/2, 양방향)

20

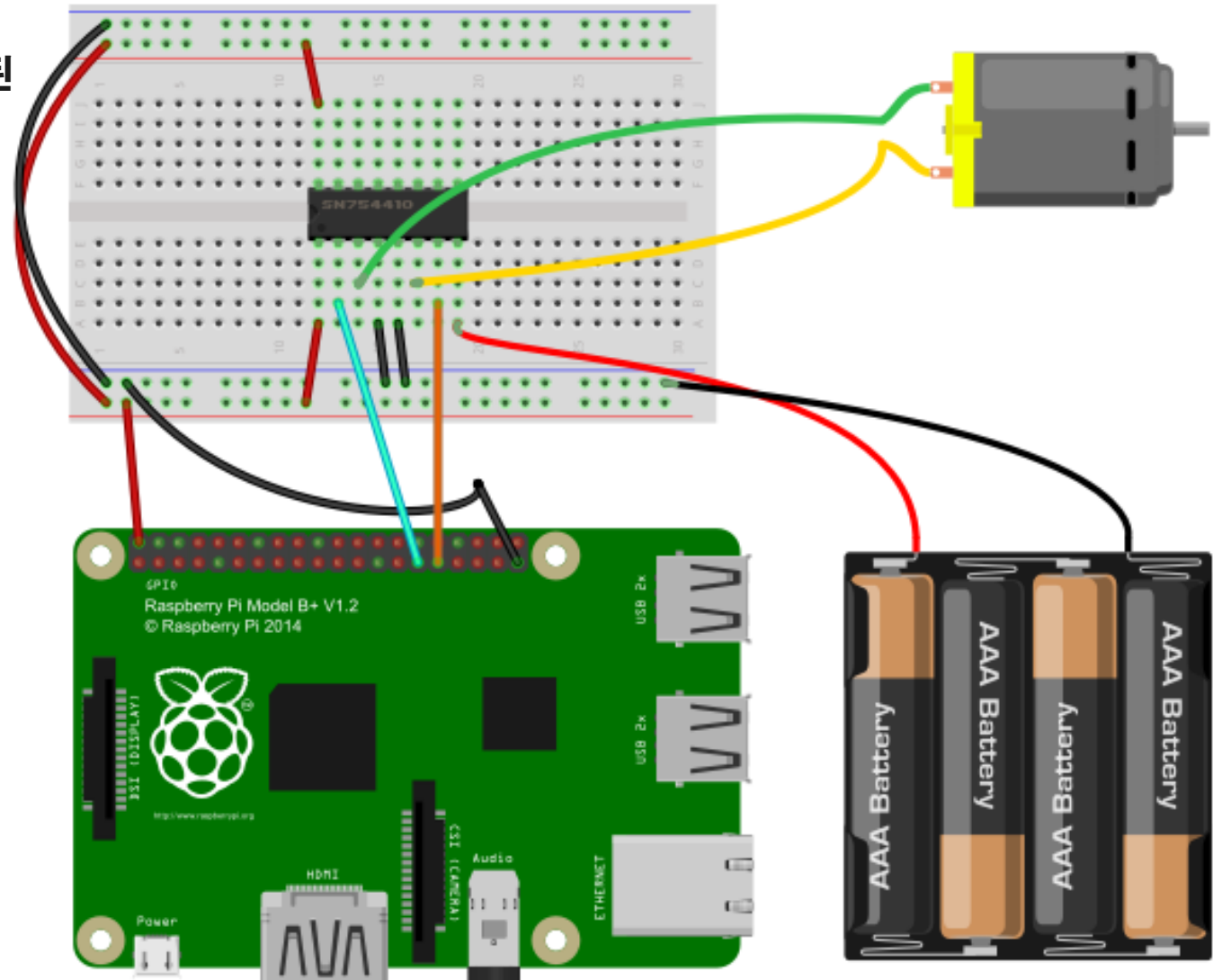
- 스위치 버튼으로 IN₁, IN₂ 입력
- OUT₁, OUT₂에 연결된 모터 회전 방향 확인
- 구성
 - 스위치 2개
 - L293D
 - DC 모터
- 라즈베리파이
 - 3.3v
 - GND



실습4 : L293D (GPIO제어)

21

- IN₁, IN₂에
GPIO로 디지털 출력
- OUT₁, OUT₂에 연결된
모터 제어
- 구성
 - L293D
 - DC 모터
- 라즈베리파이
 - 3.3v
 - GND
 - GPIO 5, 6



실습4-1 : wiringPi 명령어 사용

22

```
$ gpio -g mode 5 out  
$ gpio -g mode 6 out
```

GPIO 5와 6을 출력모드로 변경

```
$ gpio -g write 5 1  
$ gpio -g write 6 0
```

모터 A방향으로 회전

```
$ gpio -g write 5 0  
$ gpio -g write 6 0
```

모터 정지

```
$ gpio -g write 5 0  
$ gpio -g write 6 1
```

모터 B방향으로 회전

```
$ gpio -g write 5 0  
$ gpio -g write 6 0
```

모터 정지

실습4-2 : wiringPi 라이브러리 사용

23

```
#include <stdio.h>
#include <wiringPi.h>
```

```
#define INPUT1 5
#define INPUT2 6
```

```
int main(void)
```

```
{
```

```
    int i;
```

```
    wiringPiSetupGpio();
```

```
    pinMode(INPUT1, OUTPUT);
```

```
    pinMode(INPUT2, OUTPUT);
```

```
    for(i = 0 ; i < 5 ; i++){
```

```
        digitalWrite(INPUT1, HIGH);    // A 방향
```

```
        digitalWrite(INPUT2, LOW);
```

```
        delay(500);
```

```
        digitalWrite(INPUT1, LOW);    // B 방향
```

```
        digitalWrite(INPUT2, HIGH);
```

```
        delay(500);
```

```
    }
```

```
    digitalWrite(INPUT1, LOW);
```

```
    digitalWrite(INPUT2, LOW);
```

```
    return 0;
```

```
}
```

- 파일명 : motor_ex1.c

실습4-2 : wiringPi 라이브러리 사용

24

- 컴파일

```
$ gcc -lwiringPi motor_ex1.c -o motor_ex1
```

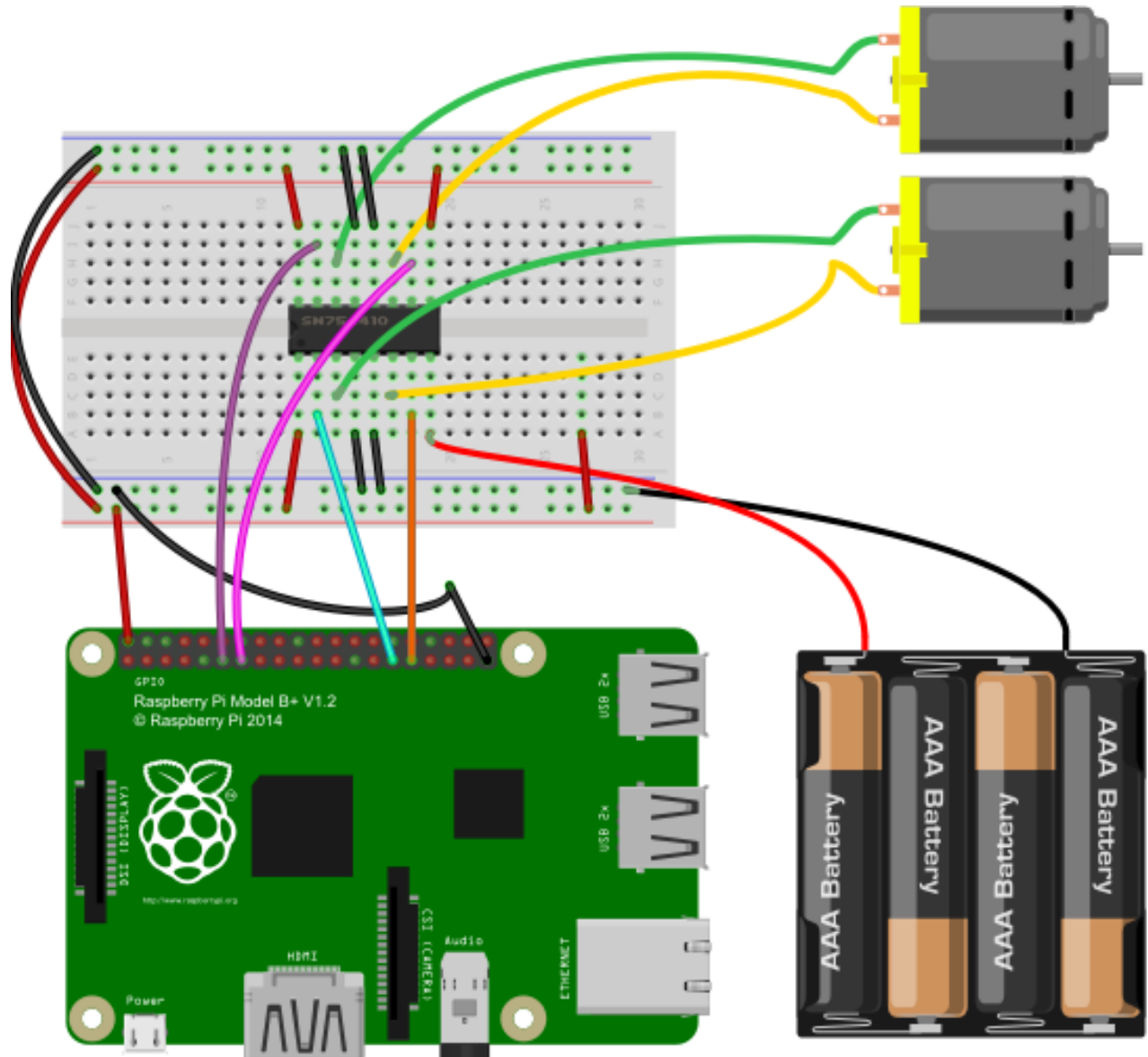
- 실행

```
$ sudo ./motor_ex1
```


실습5 : L293D (Input3/4, 양방향)

25

- 구성
 - L293D
 - DC 모터 2개
- 라즈베리파이
 - 3.3v
 - GND
 - GPIO 5, 6
 - GPIO 17, 27



실습5-1 : wiringPi 명령어 사용

26

- GPIO 명령 제어

```
$ gpio -g mode 17 out  
$ gpio -g mode 27 out  
$ gpio -g write 17 1  
$ gpio -g write 27 0
```

모터 A방향으로 회전

```
$ gpio -g write 17 0  
$ gpio -g write 27 0
```

모터 정지

```
$ gpio -g write 17 0  
$ gpio -g write 27 1
```

모터 B방향으로 회전

```
$ gpio -g write 17 0  
$ gpio -g write 27 0
```

모터 정지

실습5-2 : wiringPi 라이브러리 사용

27

```
#include <stdio.h>
#include <wiringPi.h>
```

```
#define INPUT3 27
#define INPUT4 17
```

```
int main(void)
```

```
{
```

```
    int i;
    wiringPiSetupGpio();
```

```
    pinMode(INPUT3, OUTPUT);
    pinMode(INPUT4, OUTPUT);
```

```
    for(i = 0 ; i < 5 ; i++){
        digitalWrite(INPUT3, HIGH);    // A 방향
        digitalWrite(INPUT4, LOW);
        delay(500);
        digitalWrite(INPUT3, LOW);     // B 방향
        digitalWrite(INPUT4, HIGH);
        delay(500);
    }
```

```
    digitalWrite(INPUT3, LOW);
    digitalWrite(INPUT4, LOW);
```

```
    return 0;
```

```
}
```

- 파일명 : motor_ex2.c

실습5-2 : wiringPi 라이브러리 사용

28

- 컴파일

```
$ gcc -lwiringPi motor_ex2.c -o motor_ex2
```

- 실행

```
$ sudo ./motor_ex2
```

실습5-3 : wiringPi 라이브러리 사용

29

```
#include <stdio.h>
#include <wiringPi.h>

#define INPUT1 5
#define INPUT2 6
#define INPUT3 27
#define INPUT4 17

int main(void)
{
    int i;
    wiringPiSetupGpio();

    pinMode(INPUT1, OUTPUT);
    pinMode(INPUT2, OUTPUT);
    pinMode(INPUT3, OUTPUT);
    pinMode(INPUT4, OUTPUT);

    for(i = 0 ; i < 3 ; i++){
        digitalWrite(INPUT1, HIGH);
        digitalWrite(INPUT2, LOW);
        digitalWrite(INPUT3, HIGH);
        digitalWrite(INPUT4, LOW);
        delay(500);
        digitalWrite(INPUT1, LOW);
        digitalWrite(INPUT2, HIGH);
        digitalWrite(INPUT3, LOW);
        digitalWrite(INPUT4, HIGH);
        delay(500);
    }
```

- 파일명 : motor_ex3.c

```
digitalWrite(INPUT1, LOW);
digitalWrite(INPUT2, LOW);
digitalWrite(INPUT3, LOW);
digitalWrite(INPUT4, LOW);

return 0;
}
```

실습5-3 : wiringPi 라이브러리 사용

30

- 컴파일

```
$ gcc -lwiringPi motor_ex3.c -o motor_ex3
```

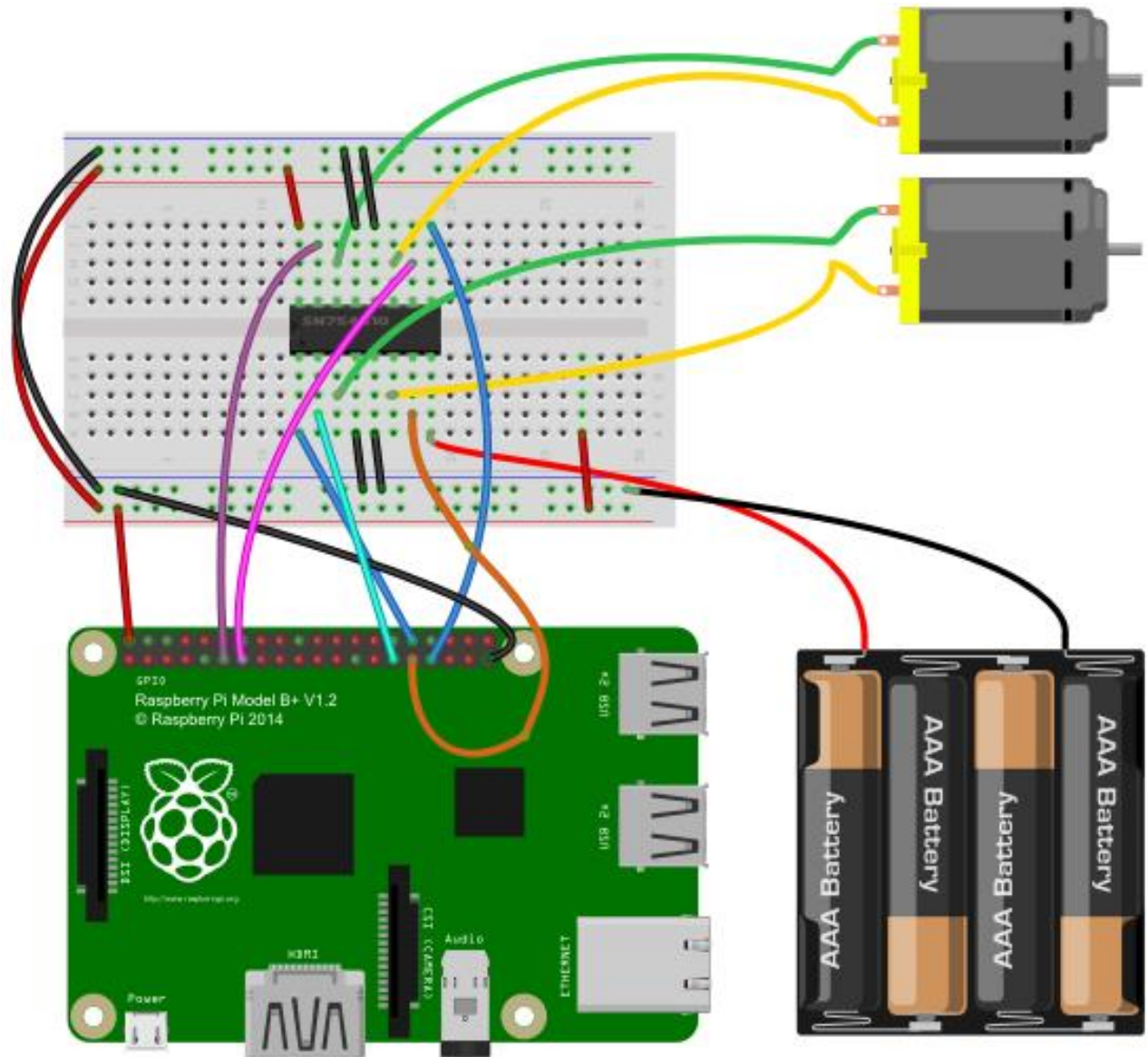
- 실행

```
$ sudo ./motor_ex3
```

실습6 : 모터 속도 제어

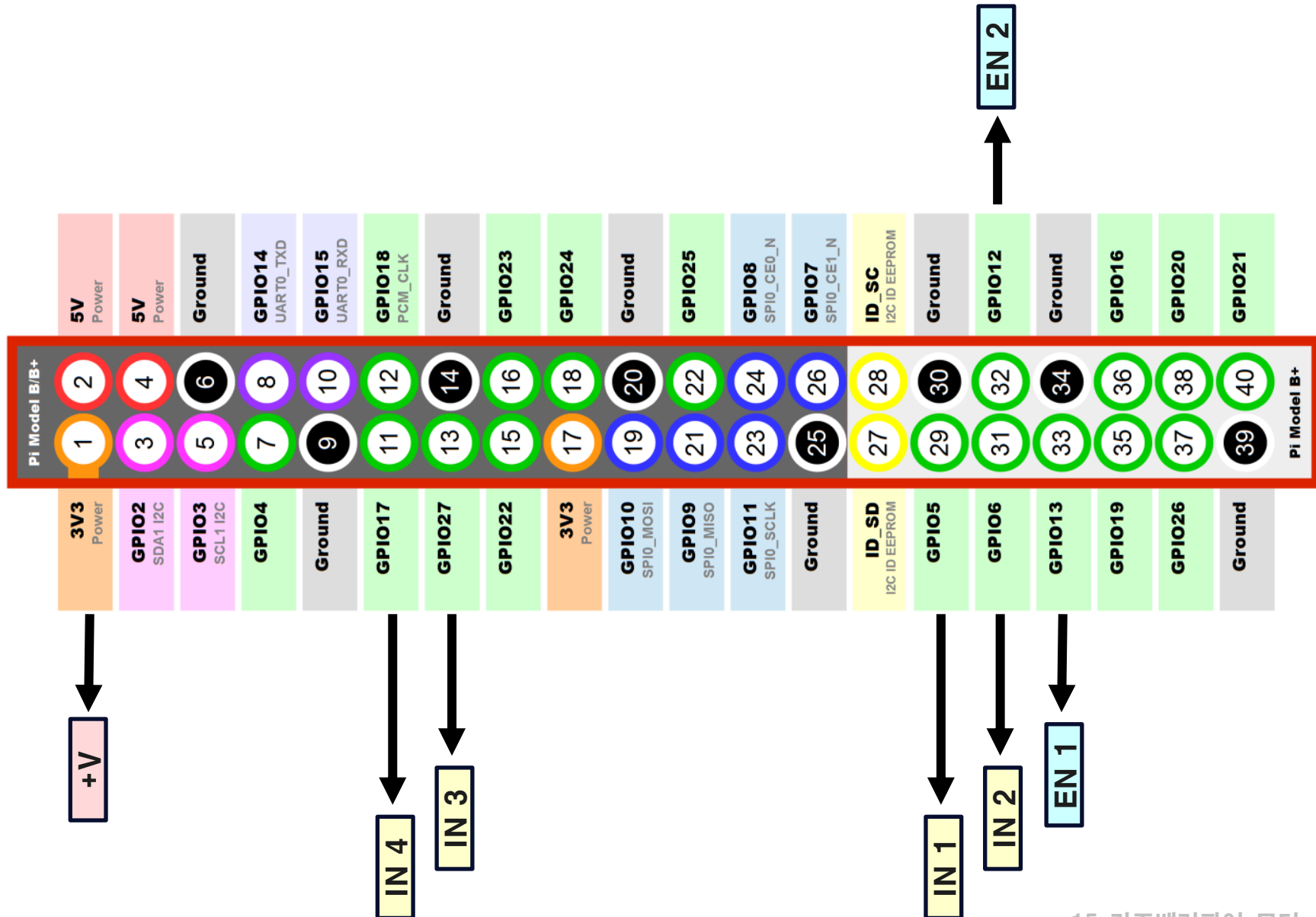
31

- 구성
 - L293D
 - 모터 2개
- 라즈베리파이
 - 3.3v
 - GND
 - GPIO 5, 6
 - GPIO 17, 27
 - GPIO 12, 13 (PWM)



RaspberryPi B+ GPIO

32



실습6-1 : wiringPi 명령어 사용

33

```
$ gpio -g mode 5 out
$ gpio -g mode 6 out
$ gpio -g mode 17 out
$ gpio -g mode 27 out
```

GPIO 5, 6, 17, 27을
출력모드로 변경

```
$ gpio -g mode 13 pwm
$ gpio -g mode 12 pwm
```

GPIO 12, 13 PWM 모드로 변경

```
$ gpio -g pwm 13 1023
$ gpio -g pwm 12 1023
```

모터 속도 변경

```
$ gpio -g write 5 1
$ gpio -g write 6 0
$ gpio -g write 17 1
$ gpio -g write 27 0
```

모터 ON

```
$ gpio -g pwm 12 700
$ gpio -g pwm 12 500
$ gpio -g pwm 12 300
```

모터 속도 변경

```
$ gpio -g write 5 0
$ gpio -g write 17 0
```

또는

```
$ gpio -g pwm 13 0
$ gpio -g pwm 12 0
```

모터 OFF

실습6-2 : wiringPi 라이브러리 사용

34

```
#include <stdio.h>
#include <wiringPi.h>

#define INPUT1 5
#define INPUT2 6
#define PWM1 13

#define INPUT3 27
#define INPUT4 17
#define PWM2 12

int main(void)
{
    int i;
    wiringPiSetupGpio();

    pinMode(INPUT1, OUTPUT);
    pinMode(INPUT2, OUTPUT);
    pinMode(PWM1, PWM_OUTPUT);
    pinMode(INPUT3, OUTPUT);
    pinMode(INPUT4, OUTPUT);
    pinMode(PWM2, PWM_OUTPUT);

    digitalWrite(INPUT1, HIGH);
    digitalWrite(INPUT2, LOW);
    digitalWrite(INPUT3, HIGH);
    digitalWrite(INPUT4, LOW);
```

• 파일명 : motor_ex4.c

```
// 범위 0 ~ 1024
pwmWrite(PWM1, 0);
pwmWrite(PWM2, 0);
delay(1000);
pwmWrite(PWM1, 400);
pwmWrite(PWM2, 400);
delay(1000);
pwmWrite(PWM1, 700);
pwmWrite(PWM2, 700);
delay(1000);
pwmWrite(PWM1, 900);
pwmWrite(PWM2, 900);
delay(1000);
pwmWrite(PWM1, 1024);
pwmWrite(PWM2, 1024);
delay(1000);

digitalWrite(INPUT1, LOW);
digitalWrite(INPUT3, LOW);

return 0;
}
```

실습6-2 : wiringPi 라이브러리 사용

35

- 컴파일

```
$ gcc -lwiringPi motor_ex4.c -o motor_ex4
```

- 실행

```
$ sudo ./motor_ex4
```

실습6-3 : wiringPi 라이브러리 사용

36

```
#include <stdio.h>
#include <wiringPi.h>

#define INPUT1 5
#define INPUT2 6
#define PWM1 13

#define INPUT3 27
#define INPUT4 17
#define PWM2 12

int main(void)
{
    int i;
    wiringPiSetupGpio();

    pinMode(INPUT1, OUTPUT);
    pinMode(INPUT2, OUTPUT);
    pinMode(PWM1, PWM_OUTPUT);
    pinMode(INPUT3, OUTPUT);
    pinMode(INPUT4, OUTPUT);
    pinMode(PWM2, PWM_OUTPUT);

    digitalWrite(INPUT1, HIGH);
    digitalWrite(INPUT2, LOW);
    digitalWrite(INPUT3, HIGH);
    digitalWrite(INPUT4, LOW);
```

• 파일명 : motor_ex5.c

```
for(i = 0 ; i < 1024 ; i++)
{
    pwmWrite(PWM1, i);
    pwmWrite(PWM2, i);
    if(i % 100 == 0)
        printf("PWM : %d\n", i);
    delay(10);
}

delay(3000);

digitalWrite(INPUT1, LOW);
digitalWrite(INPUT3, LOW);

return 0;
}
```

실습6-3 : wiringPi 라이브러리 사용

37

- 컴파일

```
$ gcc -lwiringPi motor_ex5.c -o motor_ex5
```

- 실행

```
$ sudo ./motor_ex5
```