

Introduction to Robot Making Class ***- Week 2 -***

Department of Electrical and Computer Engineering
Seoul National University

Seong Hyeon Park and Jae Young Chung

목차

1. 아두이노 프로그래밍

1.1 변수와 자료형

1.2 조건문

1.3 클래스

2. 모터제어

2.1 PWM이란?

2.2 모터 2개 전진 제어

2.3 모터 2개 후진 제어

3. 조도센서 제어

2.1 조도센서 동시에 제어

4. 로봇제작 실습 2주차

4.1 자율주행 자동차를 만들어보자

아두이노 프로그래밍

■ 변수

- 데이터를 읽고 쓰는 공간: 메모리
- 예시: 센서값, 스위치 ON/OFF상태 등등
→ 메모리에 저장
- 데이터를 저장하는 곳: “변수”
- 파이썬과 다르게, “자료형” 존재!

```
int value;  
value = 8;  
int answer;  
Answer = value;
```

변수 이름	변수 값	자료형
value	8	int

■ 자료형 (아두이노 우노 한정)

자료형	int	char	float	bool
데이터 종류	정수	정수	실수	진리값
데이터 범위	-32768~32767	-128~127	-3.4E38~3.4E38	true/false
변수 용량	2byte	1byte	4byte	1byte

(char의 경우 글자의 ASCII 코드값을 저장하는데 흔히 사용됩니다)

- 자료형의 중요성: 자료형에 알맞게 데이터가 변환되기 때문
- 오른쪽 예시의 결과는? 5입니다.

```
int val = 3;  
int ans = val + (val*7-3)/7;
```

■ 연산자

□ 산술 연산자는 숫자의 사칙연산 가능케 함 *연산자의 =은 같다가 아니라 대입의 의미

연산자	+	-	*	/	%	-
의미	덧셈	뺄셈	곱셈	나눗셈	나머지	음수
a=7 , b=4	a+b = 11	a-b = 3	a*b = 28	a/b = 1	a%b = 3	-a = -7

□ 계산과 동시에 대입을 하는 연산자들 (쓰지말았으면 하는 문법!)

연산	a += b	a -= b	a *= b	a /= b	a++	a--
의미	a = a+b	a = a-b	a = a*b	a = a/b	a += 1	a -= 1

□ 비교연산자는 값의 같음 혹은 다름의 우위를 판별함

연산자	==	!=	<	<=
의미	같다	다르다	작다	작거나 같다
a = 3, b = 4	(a==b) : false	(a!=b) : true	(a<b) : true	(a<=b) true

아두이노 프로그래밍

■ 조건문

- if 문 : 조건이 참 (TRUE)인 경우, {}안의 코드가 실행됨
- switch 문 : case: 안에 꼭 break;를 붙여주는 습관을 들이도록 합시다.

```
if (진리값 1)
{
    진리값 1이 참일 경우 실행할 코드;
}
else if (진리값 2)
{
    진리값 1은 거짓이지만 진리값 2가 참일 경우 실행할 코드;
}
else
{
    진리값 1, 진리값 2 모두 거짓인 경우 실행할 코드;
}
```

```
switch (index)
{
    case 1:
        num = 2;
        break;
    case 2:
        num = 3;
        break;
    case 3:
        num = 5;
        break;
    default:
        num = 0;
        break;
}
```

case 마다 break; 를 안 붙이면, 다음 case를 실행하는 특성 때문에 break; 를 꼭 붙이도록 합시다

Index가 case에 없는 경우엔 default: 내부의 내용이 실행됨

아두이노 프로그래밍

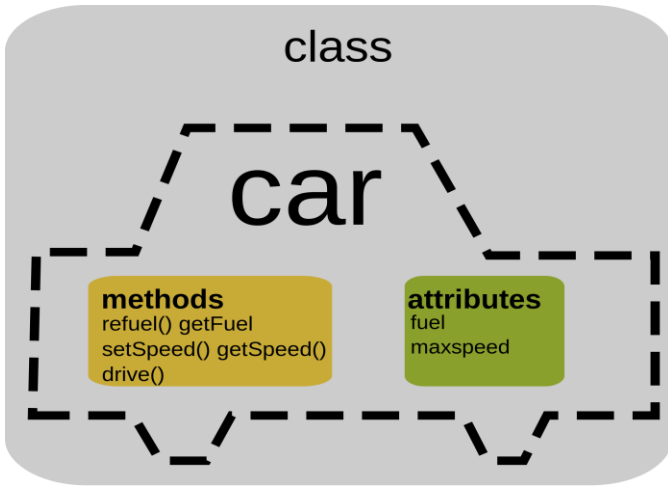
■ 클래스

□ 클래스를 쓰는 이유?

관련된 함수와 변수를 여러 번 정의하지 않으면서, 한 데에 묶어서 관리할 수 있음

□ 예시: 아두이노의 모터 각도, 속도, 앞/뒤 함수를 Motor 클래스로 묶어서 사용

□ 지금 당장은 사용하지 않을 거지만, 나중에 사용하면 편한 경우가 많음



프로그래밍에서의 클래스 개념

LED 제어 클래스 예시:

```
13 class Blinker {
14     private:
15         byte pinLED;
16
17         boolean ledState = LOW;
18
19         unsigned long timeLedOn;
20         unsigned long timeLedOff;
21
22         unsigned long nextChangeTime = 0;
23
24     public:
25         Blinker(byte pinLED, unsigned long timeLedOn, unsigned
26             this->pinLED = pinLED;
27             this->timeLedOn = timeLedOn;
28             this->timeLedOff = timeLedOff;
29
30         pinMode(pinLED, OUTPUT);
```

모터 제어

■ PWM란?

□ Pulse Width Modulation

- 아두이노는 0~5 V를 0~255로 아날로그 신호를 출력할 수 있음
- 주로 언제 아날로그 신호를 필요로 할까?

ex]

- 모터 돌리기
- LED 제어
- 다양한 소리 내기

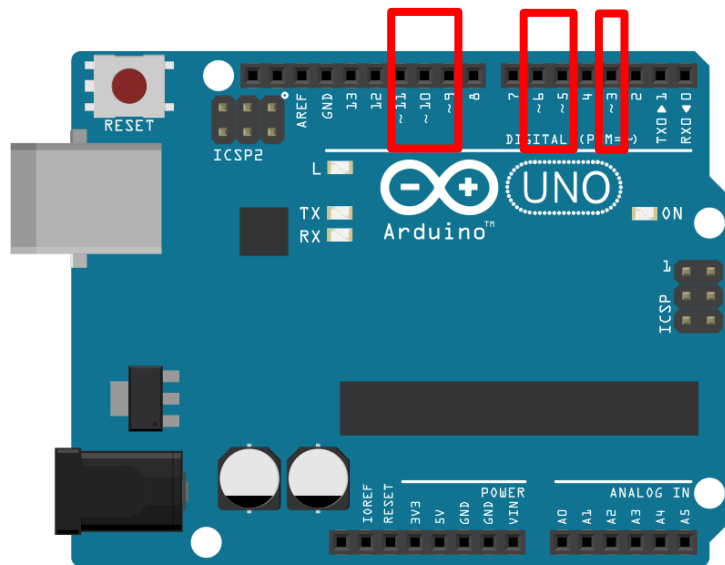
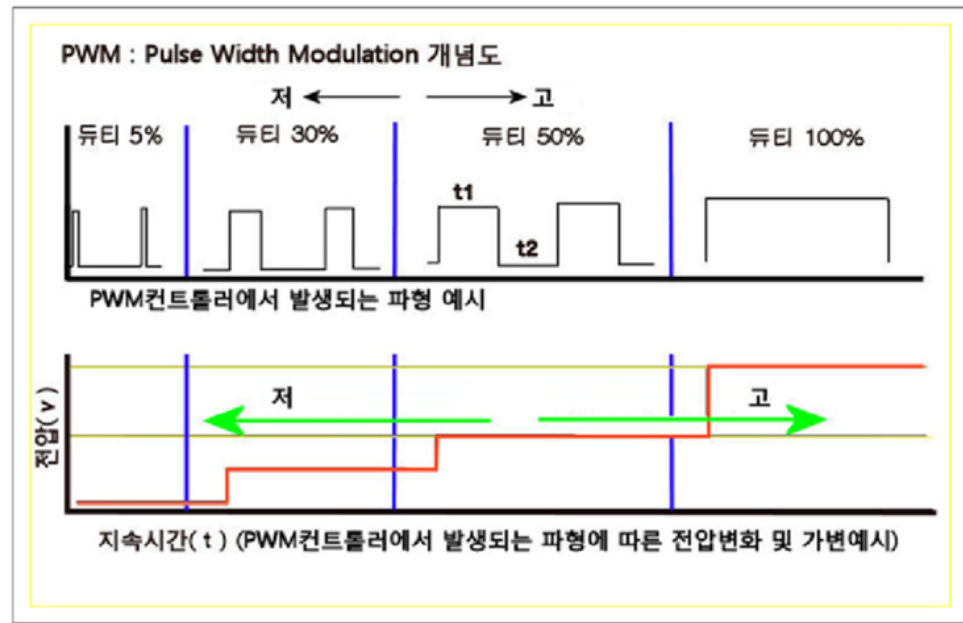
□ PWM을 사용할 수 있는 핀 (물결표 있음)!

□ 디지털핀쪽의 (3,5,6,9,10,11)

□ PWM을 사용하는 방법

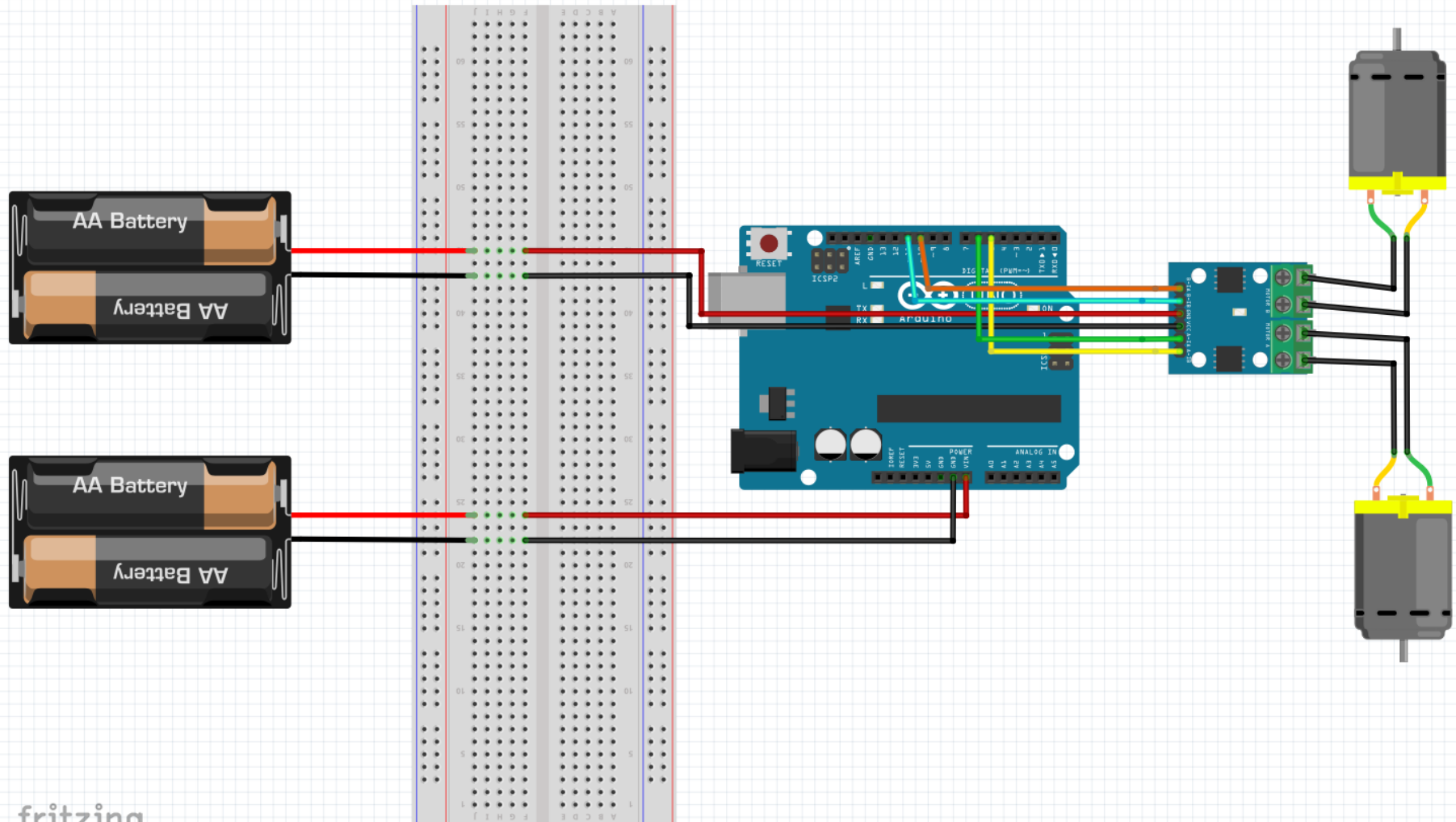
□ ex] analogWrite(5, 128)

* 5번 핀을 통해 $128/255 (=1/2) * 5 \text{ V} = 2.5 \text{ V}$ 를 출력 가능



모터 제어

- 저번시간에 만들었던 회로 (모터 부분만)
- 모터 A: 디지털 핀(5,6), 모터 B: 디지털 핀(10, 11)



모터 제어

■ PWM으로 모터 제어해보기 (앞)

□ 코드 설명

□ 디지털 핀 5,6,9,10을 출력핀으로써 사용

□ 5번핀의 출력값을 0부터 255까지 크게 해가면서, 모터 속도를 증가

□ 5번핀의 출력값을 255에서 0까지 작게 해가면서, 모터 속도를 감소

(5번 핀이 차의 왼/오른쪽 모터의 전진인지 후진인지 판별 가능!)

□ 9번핀의 출력값을 0부터 255까지 크게 해가면서, 모터 속도를 증가

□ 9번핀의 출력값을 255에서 0까지 작게 해가면서, 모터 속도를 감소

(9번 핀이 차의 왼/오른쪽 모터의 전진인지 후진인지 판별 가능!)

□ 코드 위치:

<https://github.com/WhenTheyCry96/RobotTutorial/tree/master/arduino> 에서 폴더별로 하나씩 있음

```
motorspeed1 | 아두이노 1.8.10
파일 편집 스케치 툴 도움말

motorspeed1

void setup() {

  pinMode(5, OUTPUT); // 5번핀을 출력모드로 설정합니다.
  pinMode(6, OUTPUT); // 6번핀을 출력모드로 설정합니다.
  pinMode(9, OUTPUT); // 10번핀을 출력모드로 설정합니다.
  pinMode(10, OUTPUT); // 11번핀을 출력모드로 설정합니다.

}

void loop() {

  for(int i=0;i<255;i++){
    // for 문 반복문
    analogWrite(5,i); // 5번 핀에 (i/255*5 [V])만큼 아날로그 신호 출력
    delay(10);        // 10 ms 만큼 쉬었다가
                      // 속도가 점점? -> 빨라진다
  }
  for(int i=255;i>0;i--){
    analogWrite(5,i); // 5번 핀에 (i/255*5 [V])만큼 아날로그 신호 출력
    delay(10);        // 10 ms 만큼 쉬었다가
                      // 속도가 점점? -> 느려진다
  }
  for(int i=0;i<255;i++){
    analogWrite(9,i); // 9번 핀에 (i/255*5 [V])만큼 아날로그 신호 출력
    delay(10);        // 10 ms 만큼 쉬었다가
                      // 속도가 점점? -> 빨라진다
  }
  for(int i=255;i>0;i--){
    analogWrite(9,i); // 9번 핀에 (i/255*5 [V])만큼 아날로그 신호 출력
    delay(10);        // 10 ms 만큼 쉬었다가
                      // 속도가 점점? -> 느려진다
  }
}

저장 완료.
스케치는 프로그램 저장 공간 1204 바이트(3%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 9바이트(0%)를 사용, 2039바이트의 지역변수가 남음. 최대는 2048 바이트.

33 Arduino/Genuino Uno on COM3
```

대소문자 조심!! pinMode

대소문자 조심!! analogWrite

모터 제어

■ PWM으로 모터 제어해보기 (뒤)

□ 코드 설명

□ 디지털 핀 5,6,9,10을 출력핀으로써 사용

□ 6번핀의 출력값을 0부터 255까지 크게 해가면서, 모터 속도를 증가

□ 6번핀의 출력값을 255에서 0까지 작게 해가면서, 모터 속도를 감소

(6번 핀이 차의 왼/오른쪽 모터의 전진인지 후진인지 판별 가능!)

□ 10번핀의 출력값을 0부터 255까지 크게 해가면서, 모터 속도를 증가

□ 10번핀의 출력값을 255에서 0까지 작게 해가면서, 모터 속도를 감소

(10번 핀이 차의 왼/오른쪽 모터의 전진인지 후진인지 판별 가능!)

□ 코드 위치:

<https://github.com/WhenTheyCry96/RobotTutorial/tree/master/arduino> 에서 폴더별로 하나씩 있음

```
motorspeed2 | 아두이노 1.8.10
파일 편집 스케치 툴 도움말

motorspeed2
void setup() {

    pinMode(5, OUTPUT); // 5번핀을 출력모드로 설정합니다.

    pinMode(6, OUTPUT); // 6번핀을 출력모드로 설정합니다.

    pinMode(9, OUTPUT); // 10번핀을 출력모드로 설정합니다.

    pinMode(10, OUTPUT); // 11번핀을 출력모드로 설정합니다.

}

void loop() {

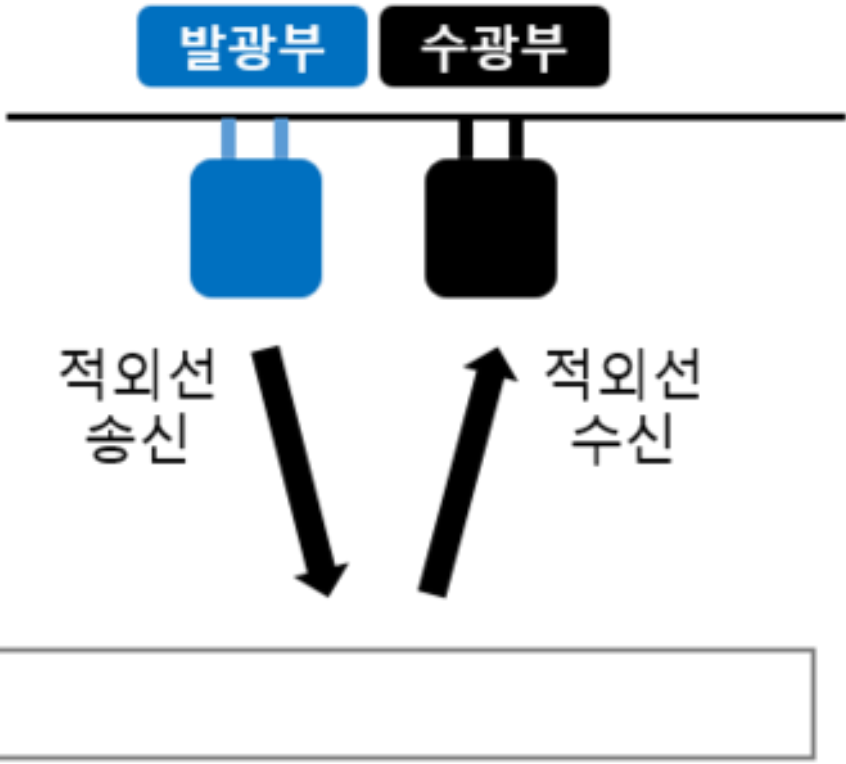
    for(int i=0;i<255;i++){
        // for 문 반복문
        digitalWrite(6,i); // 6번 핀에 (i/255*5 [V])만큼 아날로그 신호 출력
        delay(10);         // 10 ms 만큼 쉬었다가
                           // 속도가 점점? -> 빨라진다
    }
    for(int i=255;i>0;i--){
        digitalWrite(6,i); // 6번 핀에 (i/255*5 [V])만큼 아날로그 신호 출력
        delay(10);         // 10 ms 만큼 쉬었다가
                           // 속도가 점점? -> 느려진다
    }
    for(int i=0;i<255;i++){
        digitalWrite(10,i); // 10번 핀에 (i/255*5 [V])만큼 아날로그 신호 출력
        delay(10);          // 10 ms 만큼 쉬었다가
                           // 속도가 점점? -> 빨라진다
    }
    for(int i=255;i>0;i--){
        digitalWrite(10,i); // 10번 핀에 (i/255*5 [V])만큼 아날로그 신호 출력
        delay(10);          // 10 ms 만큼 쉬었다가
                           // 속도가 점점? -> 느려진다
    }
}

대소문자 조심!! pinMode
대소문자 조심!! digitalWrite
```

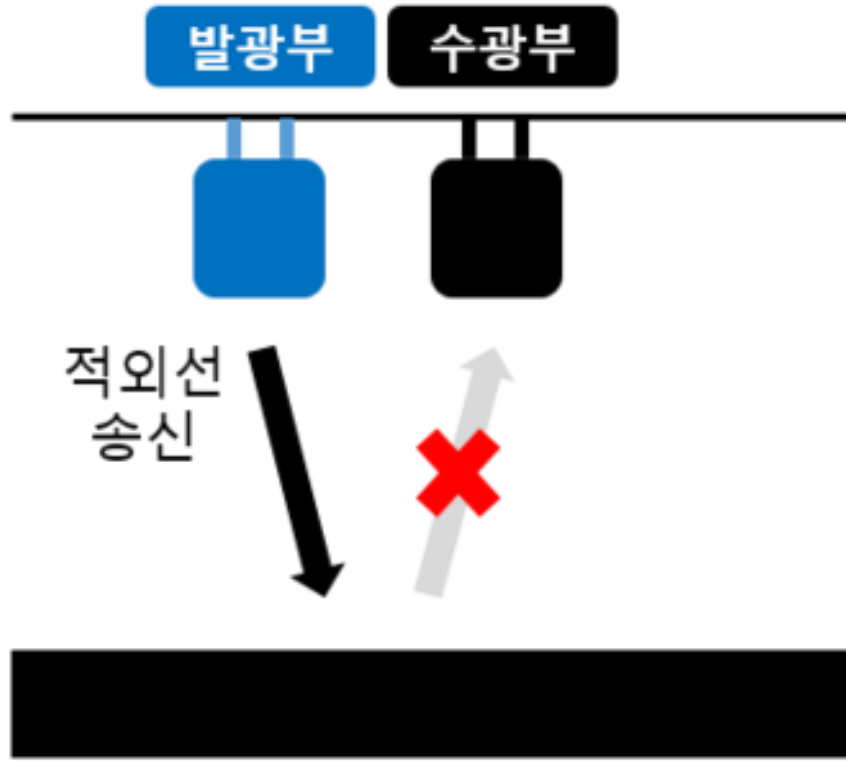
조도 센서 제어

■ 조도센서 (디지털) 입력 받기

- 발광부에서 빛을 쏘서 수광부에서 빛을 읽어서, 지면이 어두운지 밝은지를 판별하는 원리
- 밝은지 어두운지에 대한 조절은 가변저항을 드라이버로 돌리면서 구현 가능



흰색 면에서 **반사**



검정색 면에서 **흡수**

조도 센서 제어

■ 조도센서 (디지털) 입력 받기

□ 코드 설명:

□ 조도센서가 연결된 디지털 핀 2번을 다음과 같이 `#define SENSOR1 2`로 아예 설정 가능

- `#define` 변수 값 // 이런 식으로 사용
- 여기서 ; 이나 = 을 붙이면 컴파일 에러 발생

□ `pinMode(SENSOR1, INPUT);` // 입력 핀으로써 사용

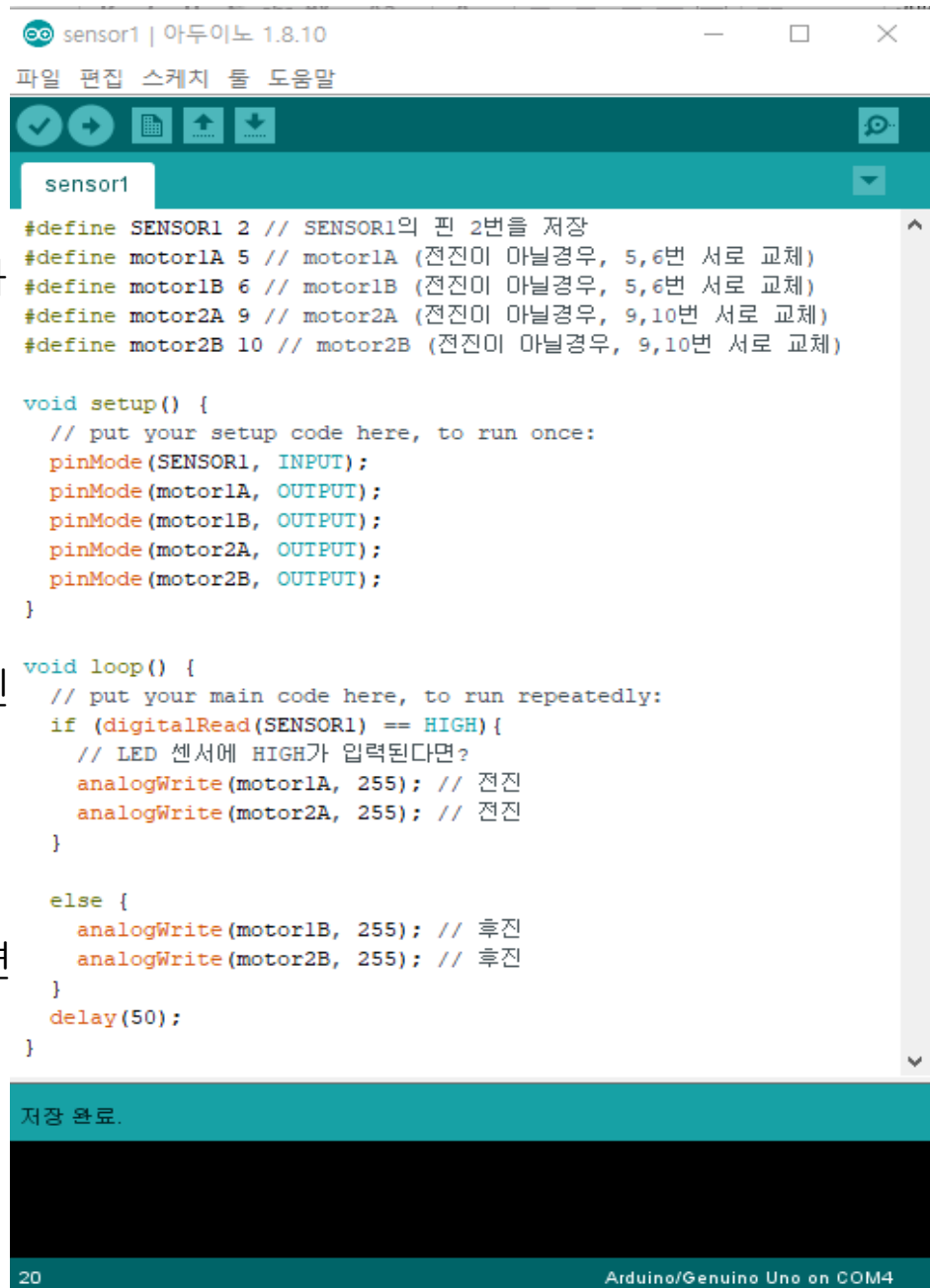
□ 조도센서 (왼쪽/가운데/오른쪽)의 디지털 핀 연결은 가능한 2,3,4번으로 해주길 바람

OR

□ 조도센서가 연결된 회로를 바꾸는 게 어려울 경우, 코드에서의 `#define` 부분을 바꿔주면 됨

□ 코드 위치:

<https://github.com/WhenTheyCry96/RobotTutorial/tree/master/arduino> 에서 폴더별로 하나씩 있음



```
sensor1 | 아두이노 1.8.10
파일 편집 스케치 툴 도움말

sensor1

#define SENSOR1 2 // SENSOR1의 핀 2번을 저장
#define motor1A 5 // motor1A (전진이 아닐경우, 5,6번 서로 교체)
#define motor1B 6 // motor1B (전진이 아닐경우, 5,6번 서로 교체)
#define motor2A 9 // motor2A (전진이 아닐경우, 9,10번 서로 교체)
#define motor2B 10 // motor2B (전진이 아닐경우, 9,10번 서로 교체)

void setup() {
  // put your setup code here, to run once:
  pinMode(SENSOR1, INPUT);
  pinMode(motor1A, OUTPUT);
  pinMode(motor1B, OUTPUT);
  pinMode(motor2A, OUTPUT);
  pinMode(motor2B, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (digitalRead(SENSOR1) == HIGH) {
    // LED 센서에 HIGH가 입력된다면?
    analogWrite(motor1A, 255); // 전진
    analogWrite(motor2A, 255); // 전진
  }

  else {
    analogWrite(motor1B, 255); // 후진
    analogWrite(motor2B, 255); // 후진
  }
  delay(50);
}

저장 완료.

20 Arduino/Genuino Uno on COM4
```

조도 센서 제어 (3개)

```
sensor2

#define SENSOR1 2 // SENSOR1의 핀 2번을 저장(제일 왼쪽)
#define SENSOR2 3 // SENSOR2의 핀 3번을 저장 (가운데 센서)
#define SENSOR3 4 // SENSOR3의 핀 4번을 저장 (제일 오른쪽)
#define motor1A 5 // motor1A (전진이 아닐경우, 5,6번 서로 교체)
#define motor1B 6 // motor1B (전진이 아닐경우, 5,6번 서로 교체)
#define motor2A 9 // motor2A (전진이 아닐경우, 9,10번 서로 교체)
#define motor2B 10 // motor2B (전진이 아닐경우, 9,10번 서로 교체)

void setup() {
  // put your setup code here, to run once:
  pinMode(SENSOR1, INPUT);
  pinMode(motor1A, OUTPUT); // motor1 : 왼쪽 모터
  pinMode(motor1B, OUTPUT); // motor2 : 오른쪽 모터
  pinMode(motor2A, OUTPUT);
  pinMode(motor2B, OUTPUT);
}
```

□ 코드 사용 시 주의 점:

1. SENSOR1,2,3의 위치를 각각 파악할것!
2. motor1,2가 각각 왼쪽 오른쪽 모터인지 판단!
3. motor의 전진/후진이 각각 (5,6) (9,10)인지 확인!

본인의 회로에 맞추어서 코드에서의 #define 부분을 변경하면 됩니다.

```
void loop() {
  // put your main code here, to run repeatedly:
  int S1 = digitalRead(SENSOR1); // SENSOR1의 상태값 저장
  int S2 = digitalRead(SENSOR2); // SENSOR2의 상태값 저장
  int S3 = digitalRead(SENSOR3); // SENSOR3의 상태값 저장

  if (s1 == HIGH && s3 == LOW) {
    // 조도센서 왼쪽은 HIGH 그리고 오른쪽은 LOW일 때 (왼쪽에 선)
    analogWrite(motor1A, 150); // 전진
    analogWrite(motor2B, 150); // 후진
    // 우회전
    delay(100);
  }

  else if ((s1 == LOW && s3 == HIGH)) {
    // 조도센서 왼쪽은 LOW 그리고 오른쪽은 HIGH일 때 (오른쪽에 선)
    analogWrite(motor1B, 150); // 후진
    analogWrite(motor2A, 150); // 전진
    // 좌회전
    delay(100);
  }

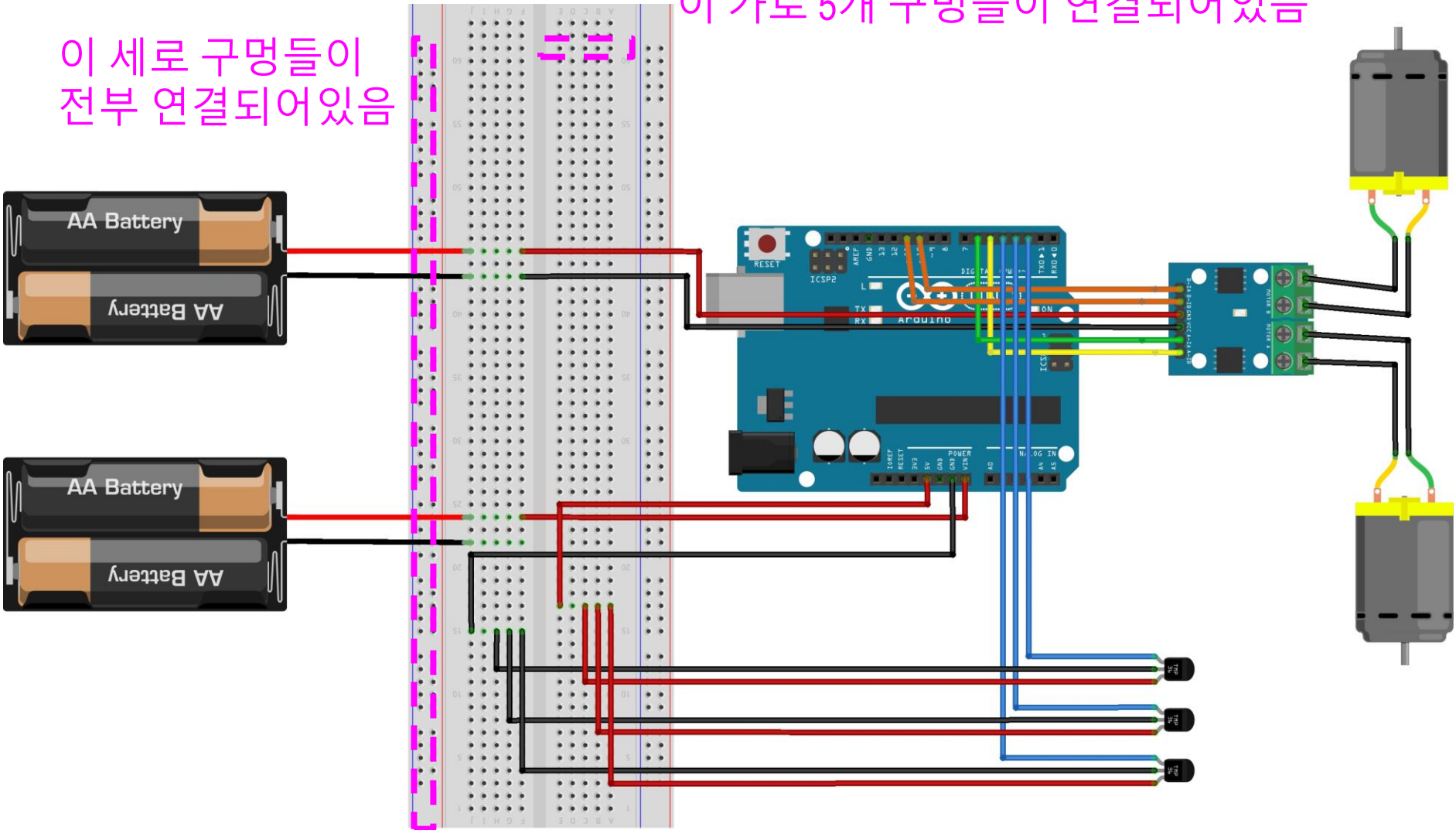
  else if ((s1 == HIGH && s2 == HIGH && s3 == HIGH)) {
    // 바로 앞에 선이 있을 때
    analogWrite(motor1B, 150); // 후진
    analogWrite(motor2B, 150); // 후진
    // 뒤로 후진
    delay(100);
  }

  else {
    // 위의 두 경우가 아닐 때
    analogWrite(motor1A, 150); // 전진
    analogWrite(motor2A, 150); // 전진
    // 앞으로 전진
    delay(100);
  }
}
```

자율주행 자동차 회로도

이 세로 구멍들이
전부 연결되어있음

이 가로 5개 구멍들이 연결되어있음



자율주행 자동차 소개

- 4차 산업 혁명 중 혁신적인 생활 변화를 이끌어 낼 기술

Automated/autonomous car

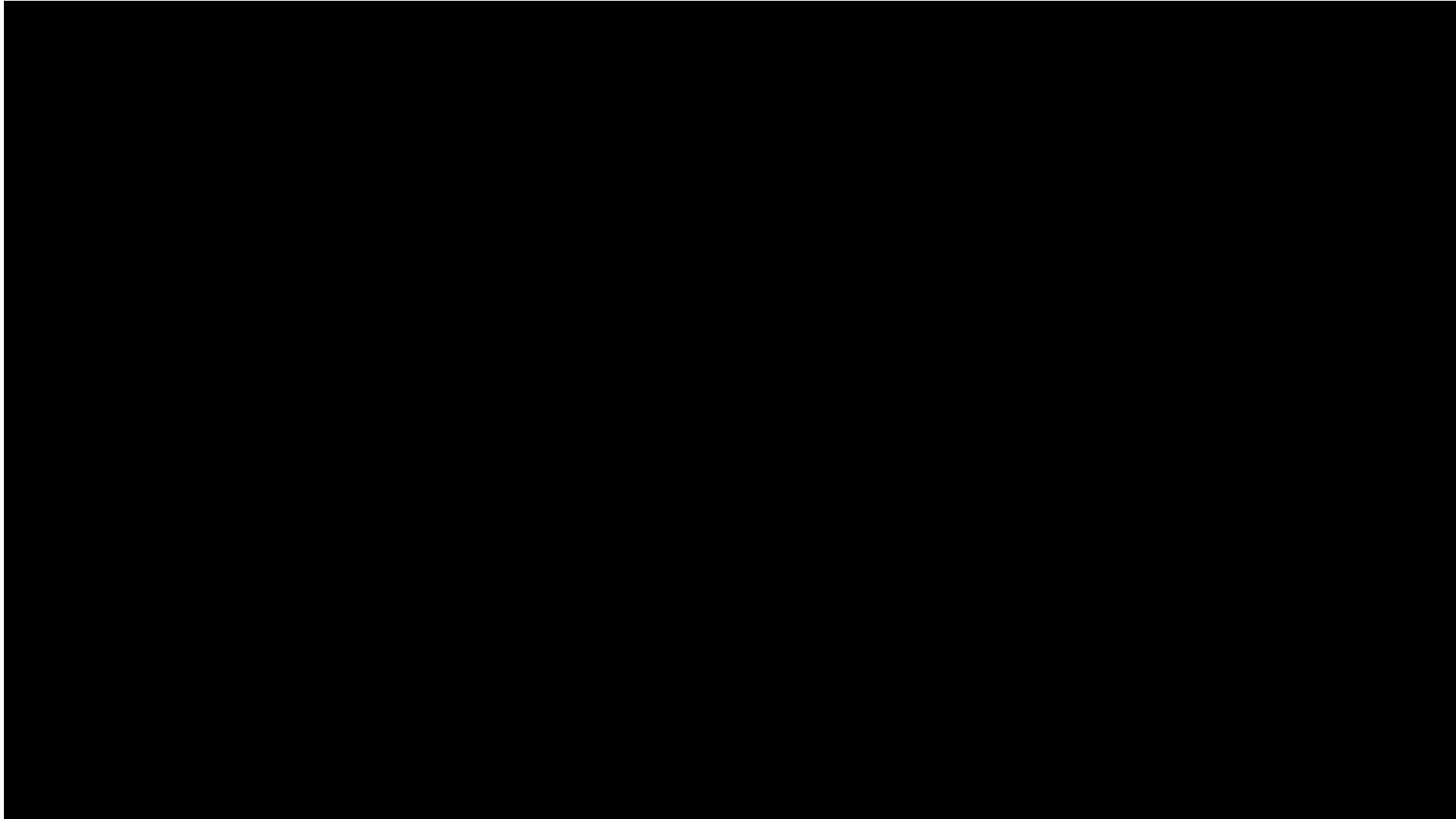


자율주행 자동차 소개

■ 일반적인 자율주행 자동차

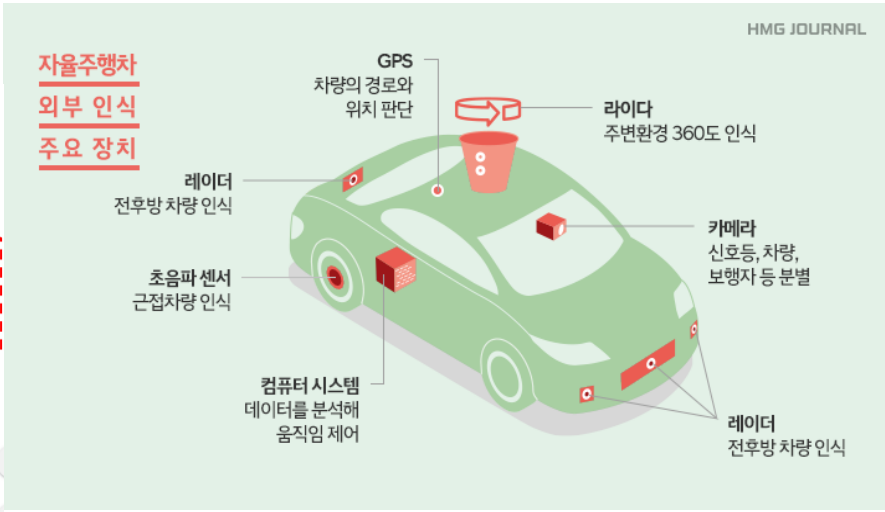
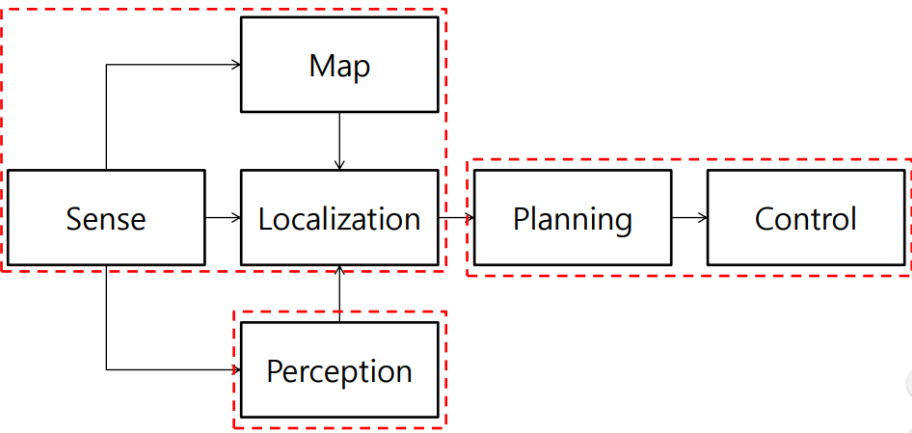
□ 테슬라 자율 주행 데모 (2019년)

<https://www.youtube.com/watch?v=tIThdr3O5Qo>

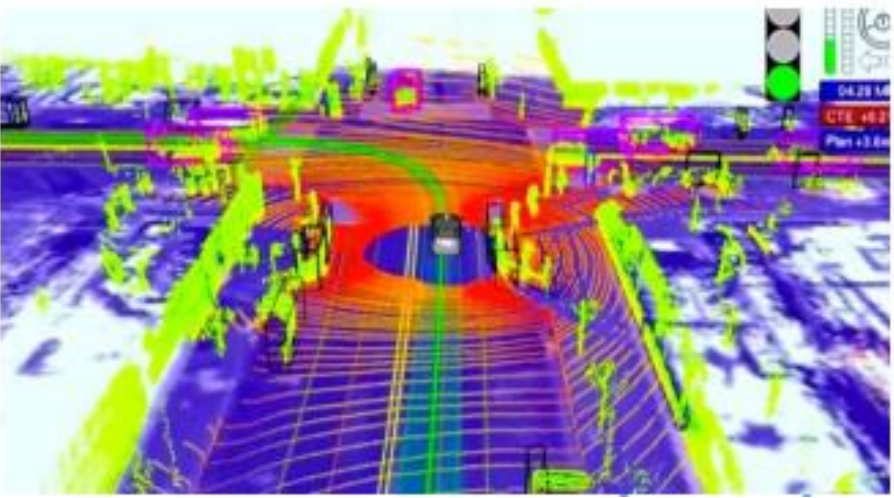


자율주행 자동차 소개

■ 일반적인 자율주행 자동차의 원리



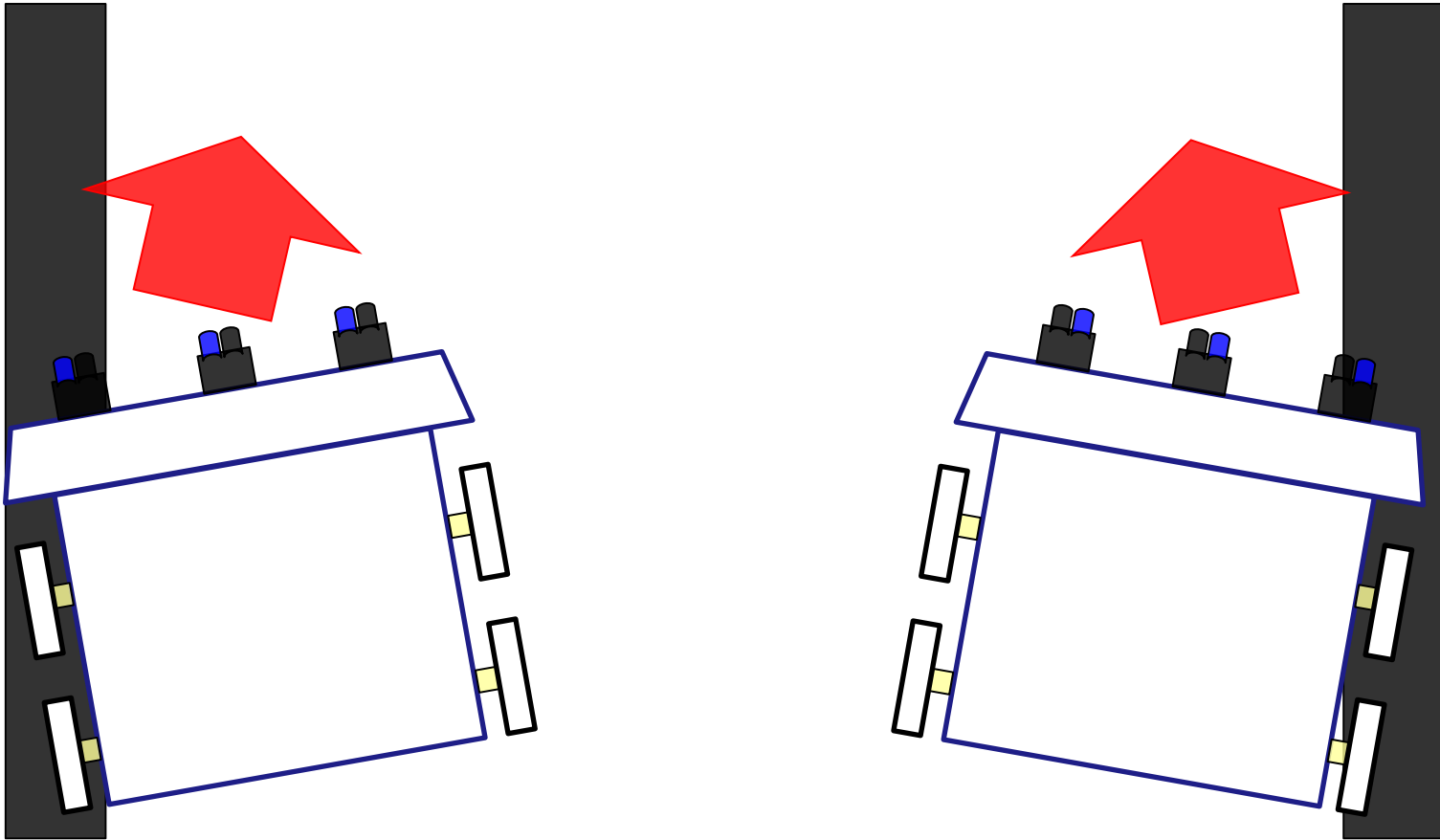
■ 일반적인 자율주행 자동차의 동작



<https://www.youtube.com/watch?v=rSMgAxC50A0>

로봇 실습 (2): 자율주행 자동차 기본 원리

■ 조도센서 + 모터 제어를 통한 트랙킹 로봇



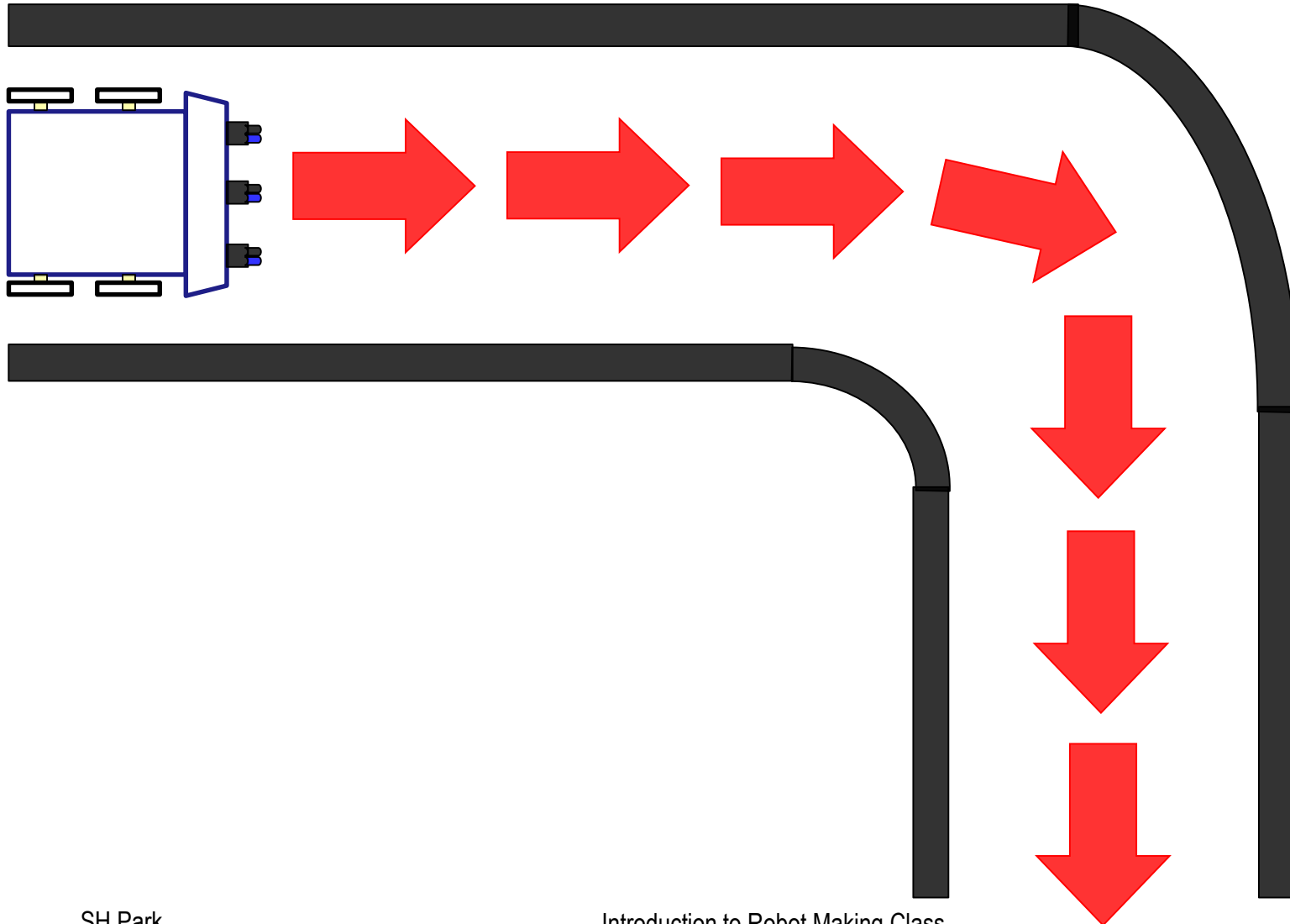
□ 코드 위치:

<https://github.com/WhenTheyCry96/RobotTutorial/tree/master/arduino/autodriving/>

에서 아두이노 파일 (.ino)에 있음 (상세한 상수값, 시간 조절등은 본인의 차량에 맞게!)

로봇 실습 (2): 자율주행 자동차

■ 조도센서 + 모터 제어를 통한 자율주행 로봇



과제: 자율주행 자동차 보강하기

- 조도센서 + 모터 제어를 통한 자율주행 로봇
(완주하도록 노력해 봅시다!)

