

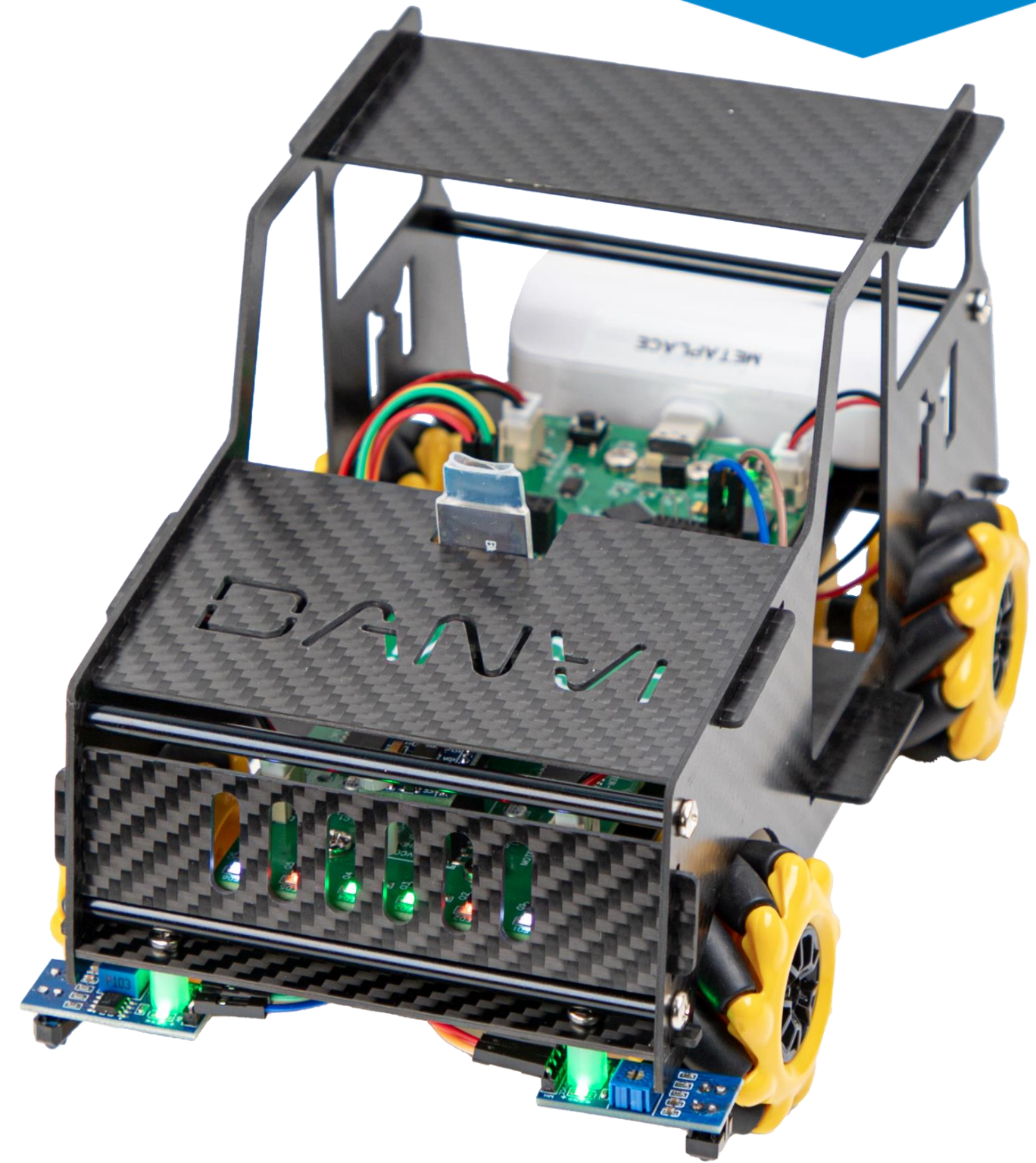
메카넘휠 AI 로봇 "단비"

Mecanum Wheel Robot AI DanVI



메카넘휠 AI 로봇 단비

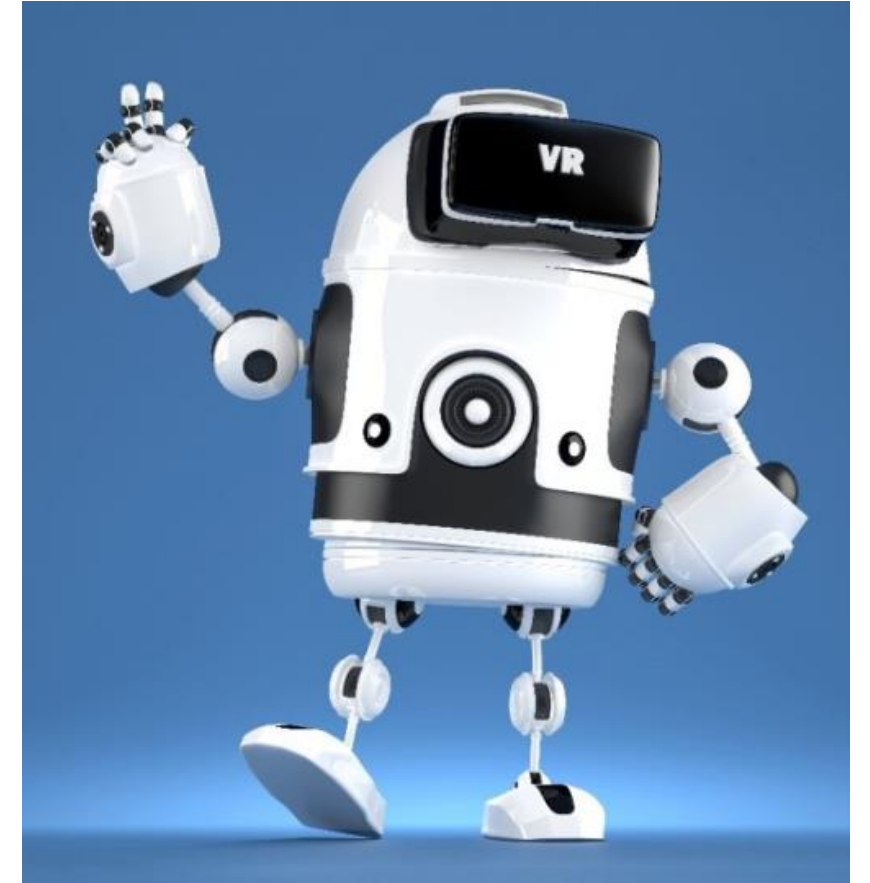
Mecanum Wheel Robot AI DanVI



METAPLACE

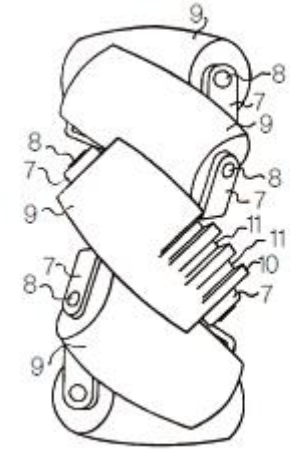
로봇이란?

- 로봇은 프로그래밍 **명령에 따라** 일을 수행하는 기계나 인공적인 장치
- 로봇은 다양한 형태와 크기로 존재하며, **사람의 역할을 대체**하거나 **보조**하기 위해 설계
- 로봇은 산업 현장에서 **제품을 조립**하거나 **공장을 자동화**하는 데 사용되는 산업 로봇
- 의료 분야에서 수술을 보조하는 로봇까지 다양한 용도로 활용
- 로봇은 우주 탐사, 해양 탐사, 구조물 탐사 등의 위험한 환경에서 인간의 안전을 위해 사용
- 최근에는 **인공지능과 센서 기술의 발전**으로 더욱 다양하고 정교한 로봇이 개발



메카넘 휠이란?

- 메카넘 휠(Mecanum Wheel)은 특별한 디자인의 바퀴로, 다방향으로 이동할 수 있는 차량이나 로봇에 사용
- 일반적인 바퀴와 달리 고유한 **롤러가 장착된 원형 바퀴**로, 롤러는 바퀴의 주변에 일정한 간격으로 부착되어 있으며, **서로 45도 각도로 회전되어 있음**
- 메카넘 휠은 바퀴의 롤러가 특정 각도로 회전하면서 차량이나 로봇이 다양한 방향으로 이동하며, **일반적인 바퀴와 달리 전후좌우 및 대각선 방향으로의 이동이 가능하다.**
- 메카넘 휠을 장착한 차량이나 **로봇은 좁은 공간에서의 이동 및 회전이 용이하며**, 복잡한 환경에서의 조작이 가능
- 메카넘 휠은 주로 로봇 경주나 자율주행 차량 등에 사용되며, **다방향 이동이 필요한 다양한 응용 분야에서 활용**되며, 이러한 바퀴는 다른 차량이나 장애물을 피할 필요가 있는 로봇에 특히 유용하다.



인공지능이란?

- 인공지능 로봇은 인공지능 기술을 활용하여 자율적으로 행동하고 의사 결정을 내리는 로봇을 말한다. 이러한 로봇은 센서와 액추에이터를 통해 환경을 감지하고, 수집한 정보를 기반으로 판단하고 행동한다.



- 주요 특징 :

1. 인지능력: 인공지능 로봇은 주변 환경을 인식하고 해석하는 능력
2. 카메라, 라이다(LiDAR), 초음파 센서 등 다양한 센서를 사용하여 주변 환경을 감지하고 분석
3. 의사 결정: 수집한 정보를 기반으로 의사 결정. 이러한 결정은 프로그래밍된 알고리즘 또는 머신러닝 및 딥러닝과 같은 인공지능 기술로 가능
4. 자율성: 인공지능 로봇은 사람의 직접적인 제어 없이 자율적으로 움직임. 환경을 탐색하고 장애물을 피하며, 주어진 작업을 수행
5. 상호작용: 일부 인공지능 로봇은 사람과의 상호작용을 지원
6. 음성 인식, 자연어 처리, 감정 인식 등 다양한 기술을 통해 실현

인공지능 탑재 로봇의 대체 가능성이 낮은 직업 5위

- 1 레크리에이션 치료사
- 2 정비·설치·수리 관련 일선 감독관
- 3 재난관리사
- 4 정신건강·약물남용 관련 사회복지사
- 5 청능사(청력학자)

대체 가능성이 높은 직업 5위

- 1 텔레마케터
- 2 부동산 권리분석사
- 3 재봉사
- 4 데이터 분석·정리 관련 수학자
- 5 개인보험업자

<자료: 옥스퍼드대 칼 프레이 교수 논문>





메카넘 휠이 부착된 지게차 (Airtrax社)와 AGV(Kuka社)



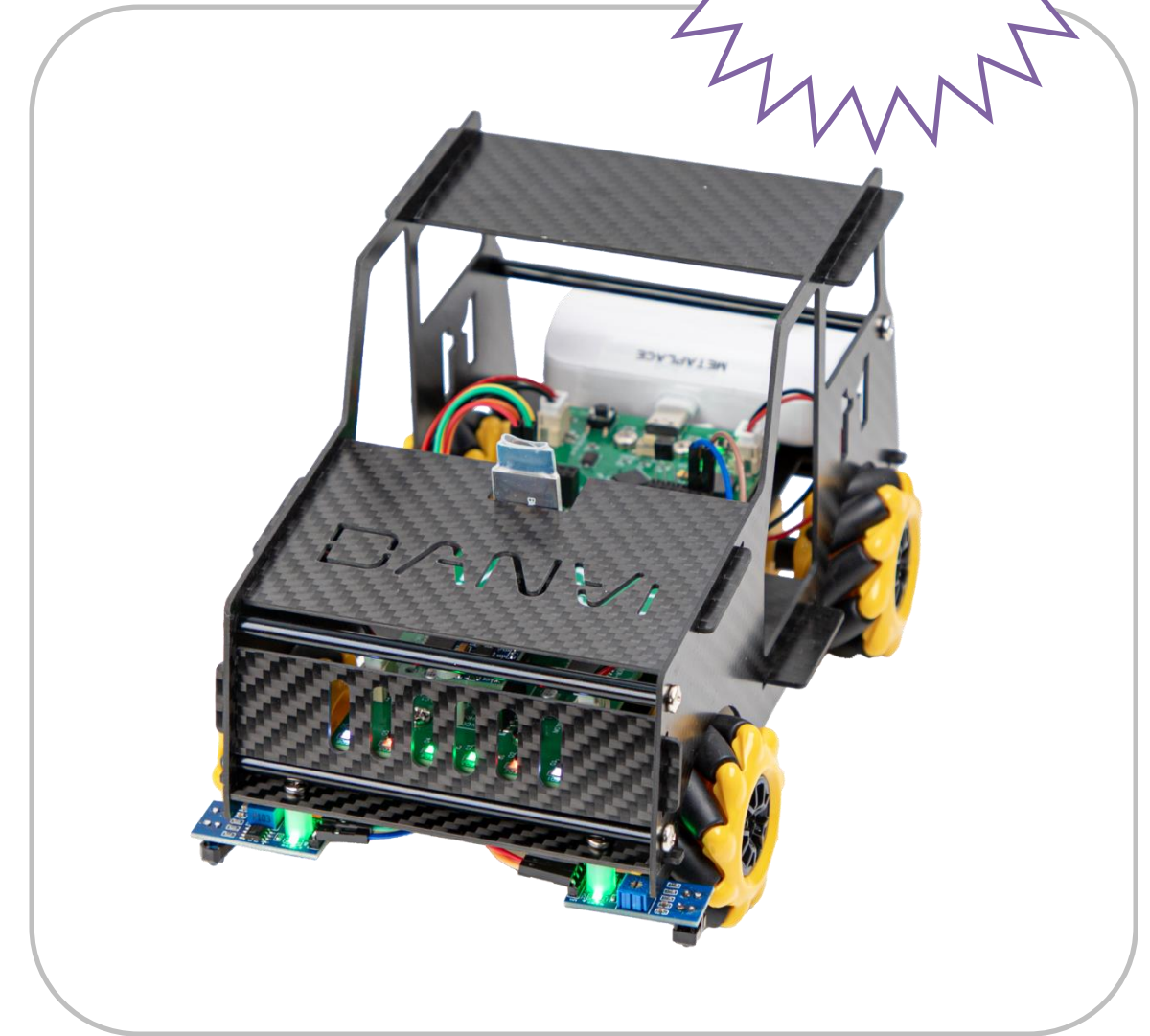
왼쪽은 옴니 휠, 오른쪽은 메카넘 휠

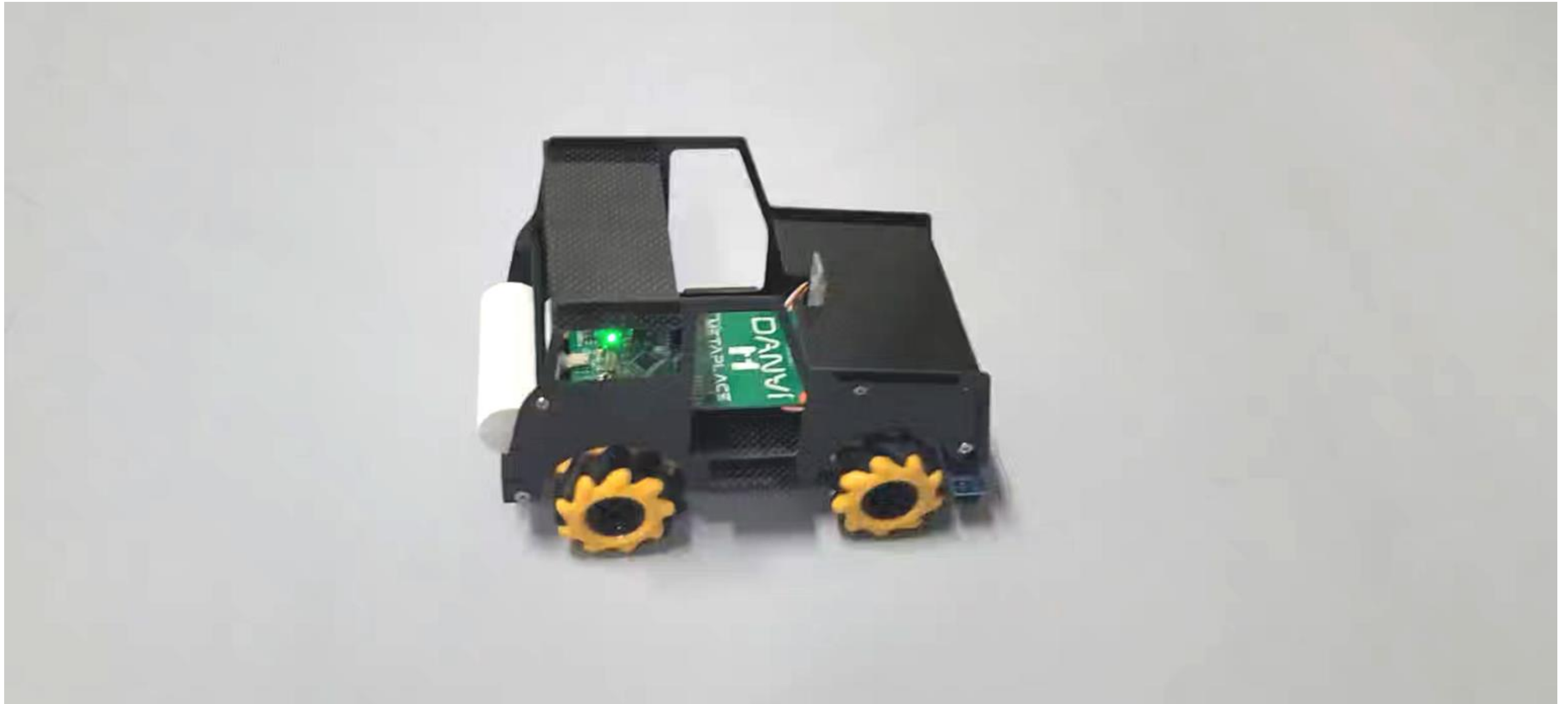


독일 쿠(KUKA)사가 개발한 무인운반차에는 메카넘 휠을 장착했다.[KUKA 홈페이지]



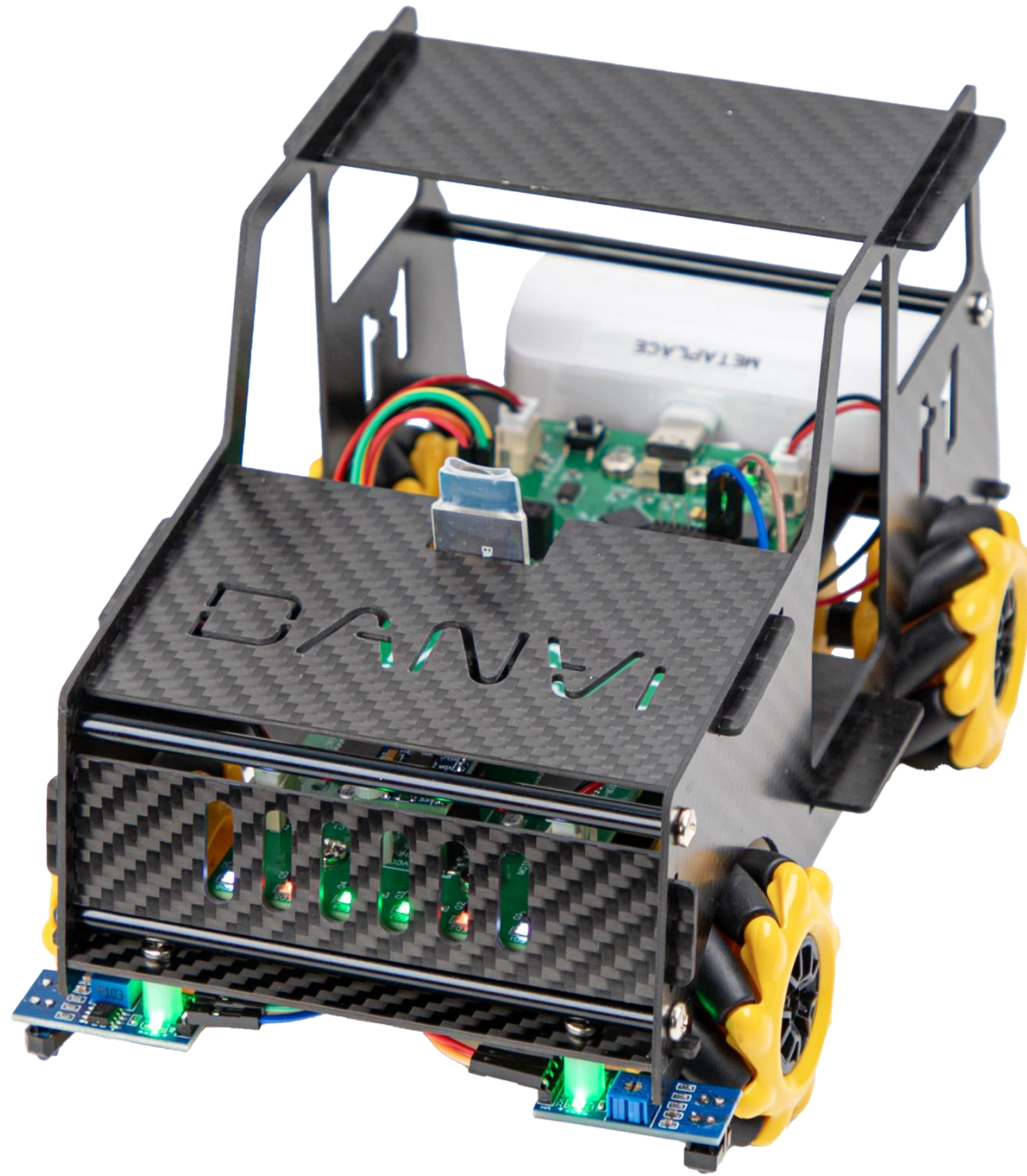
차체 회전없이 전방향 자유로운 이동이 가능한 신개념 휠체어





Arduino IDE(텍스트) 코딩

- 아두이노 설치 : <https://www.arduino.cc>

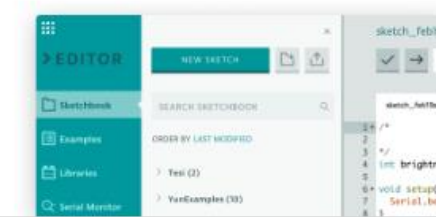


Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

[CODE ONLINE](#)

[GETTING STARTED](#)



Downloads



Arduino IDE 2.3.2

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux AppImage 64 bits (X86-64)

Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.15: "Catalina" or newer, 64 bits

macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

[Help](#)



메카넘휠 코딩로봇 **AI단비**

아두이노 기반의 메플이노보드-D를 사용하여 간단한 조립과 텍스트코딩, 인공지능 교육



아두이노

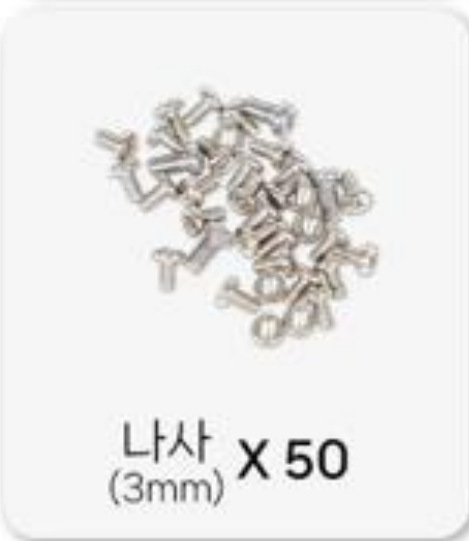
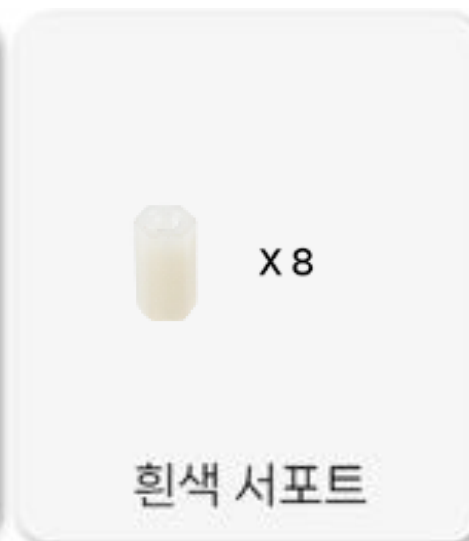


메카넘 휠



아두이노 IDE

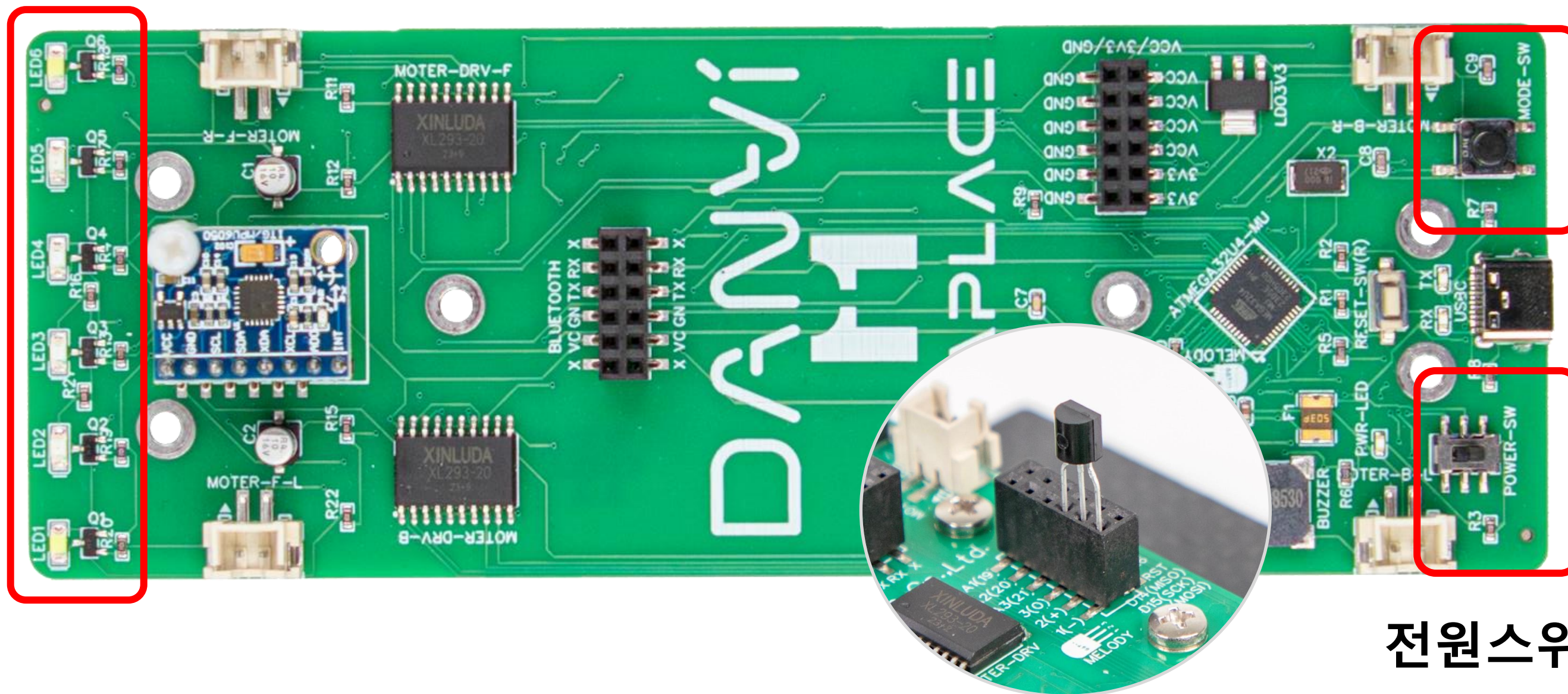
Chapter.02 핵심 부품들의 이름을 알아보기로 해요!



단비의 전원 스위치와 사용할 부품을 알아봐요!

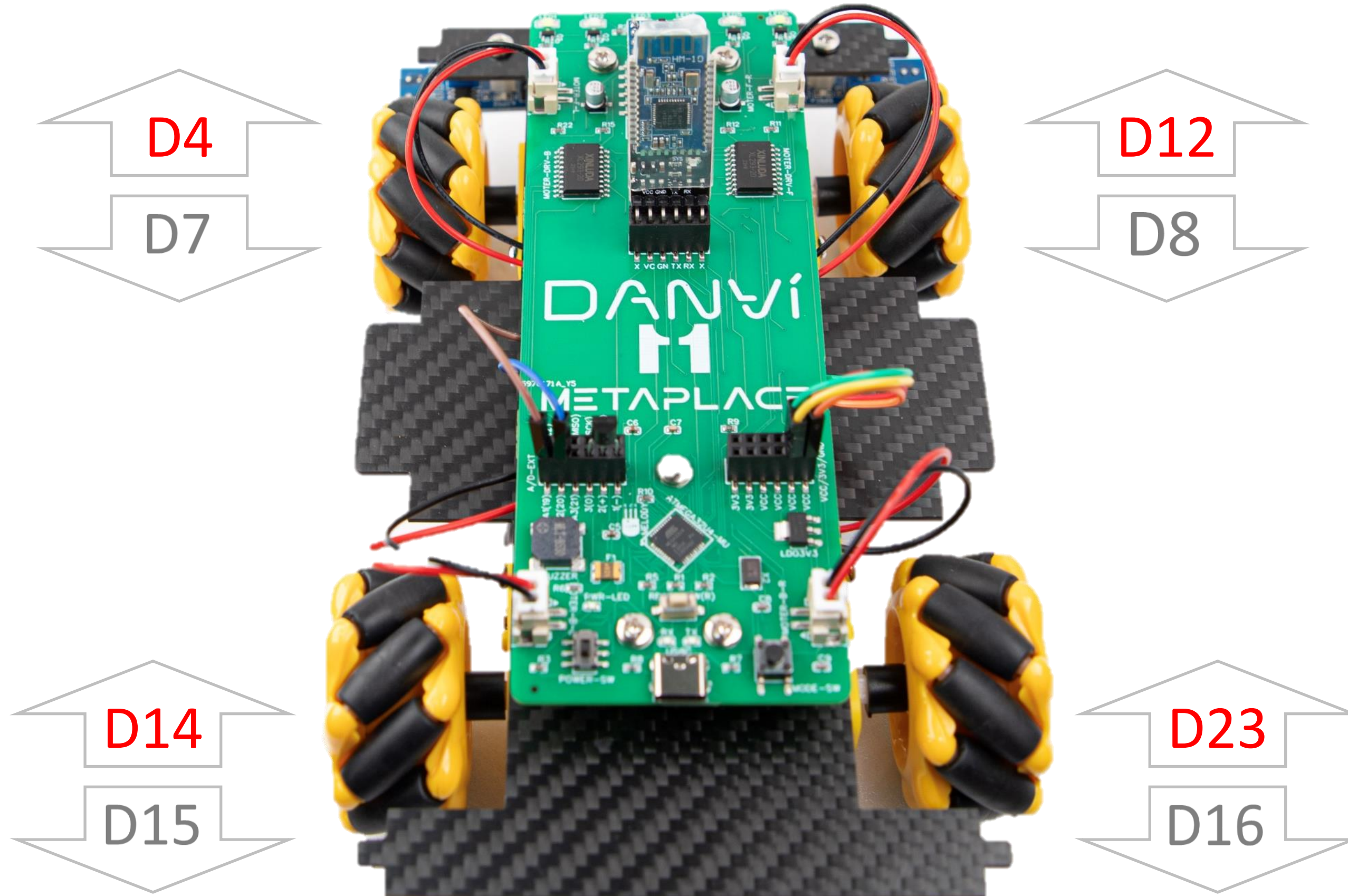
LED 핀번호 : 13

버튼 핀번호 : A0

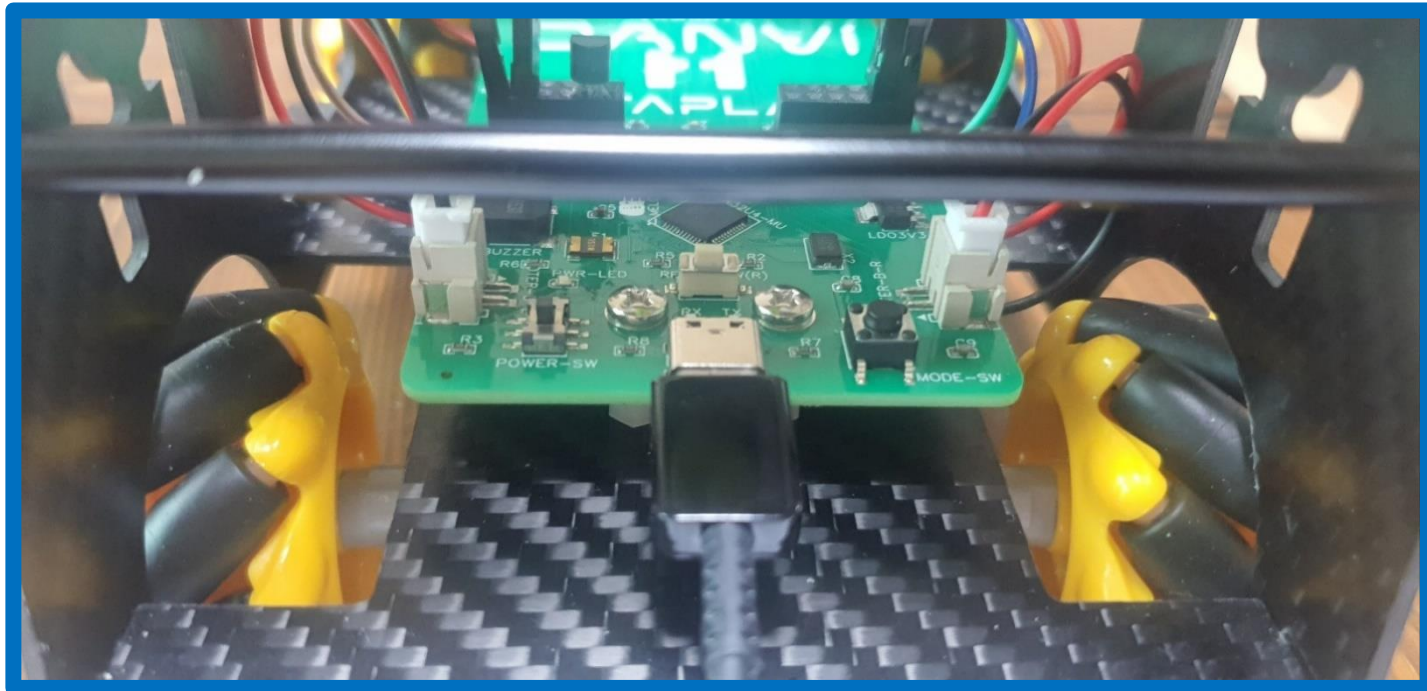


전원스위치

부저 핀번호 : 6



④ 단비는 PC 와 연결해야 코딩을 할 수 있어요!



PC 전면 이나 뒷면

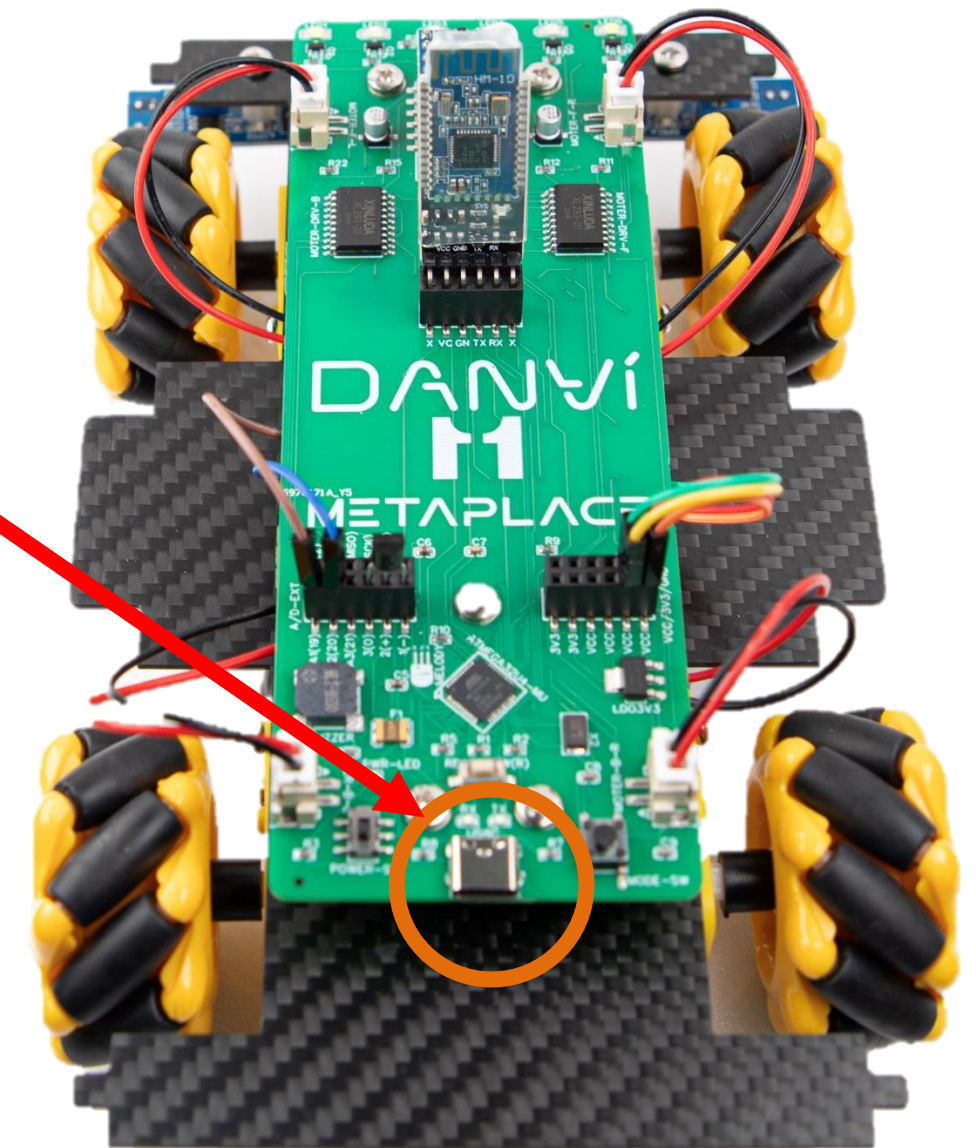
① USB A타입



② USB C타입



USB A to C 케이블

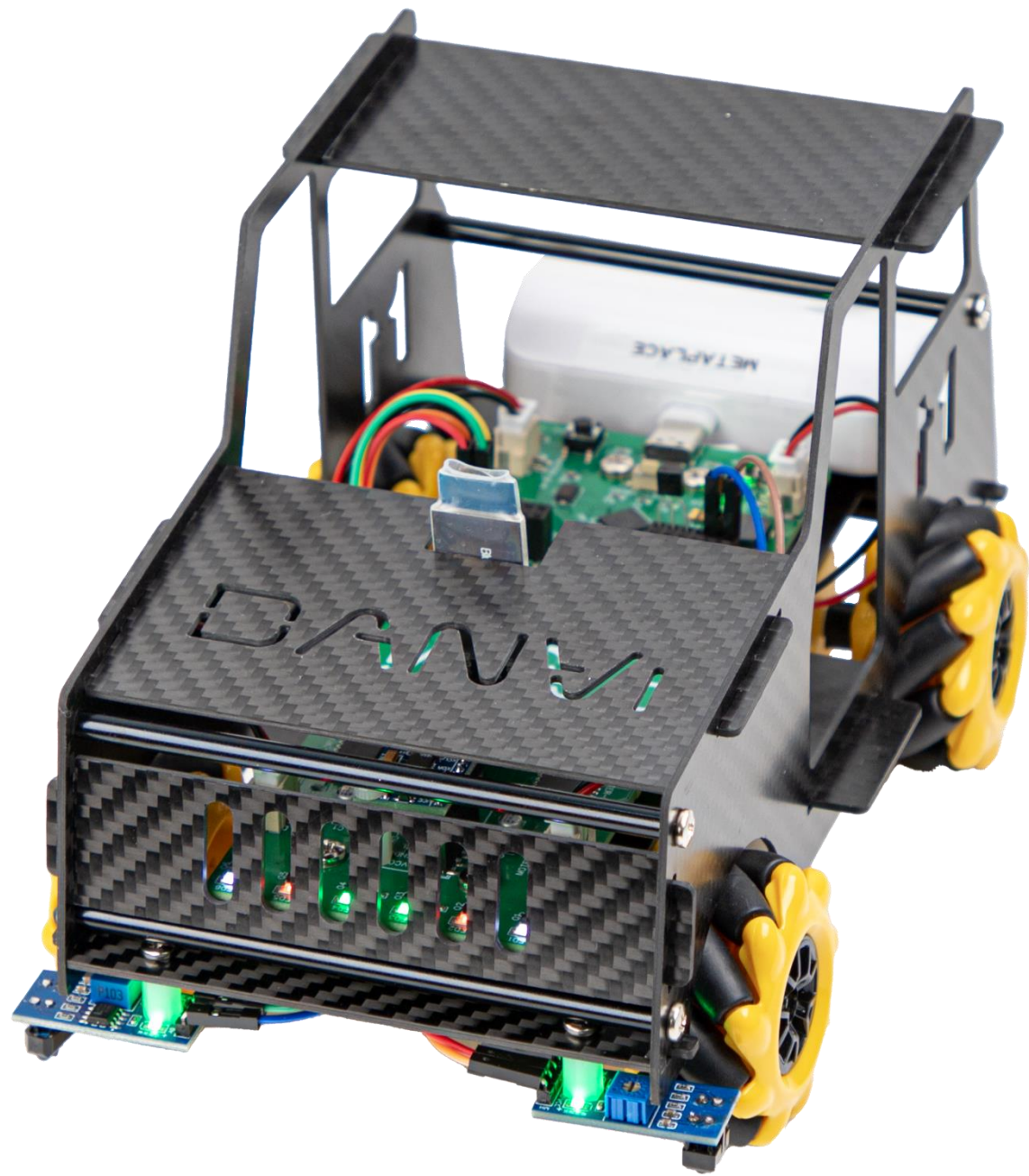


단비 뒷면

- 단비는 전원 스위치로 전원을 켜요! (사용 후 케이블을 빼기 전에 반드시 전원을 꺼주세요!!)



단비와 스케치(IDE-통합개발환경)를
이용해서 코딩해요!



Arduino IDE(텍스트) 코딩

Integrated Development Environment의 약자로, '통합 개발 환경'

- 아두이노 설치 : <https://www.arduino.cc>





1. 메뉴 : 새로운 파일생성 또는 편집 기능 등으로 메뉴에 접근이 가능한 부분
2. 버튼 : 왼쪽부터 순서대로 컴파일 버튼, 업로드버튼, 디버그 버튼으로 구성
3. 기능 : 1번 메뉴에서 쉽게 접근하기 위한 메뉴들을 구성함
4. 연결보드 : 연결된 아두이노 보드의 종류를 나타냄
5. 코드 : C언어 기반의 코드를 작성할 수 있는 부분
6. 모니터링 : 시리얼 플로터 화면과 시리얼모니터 화면을 열수 있는 버튼으로 구성

④ 아두이노의 setup() 과 loop() 함수

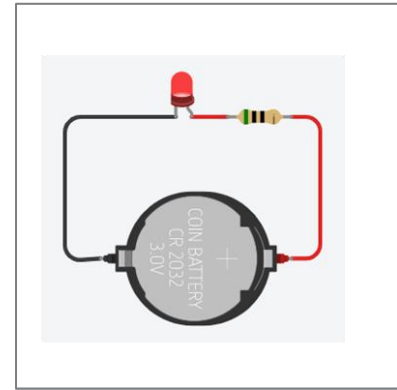
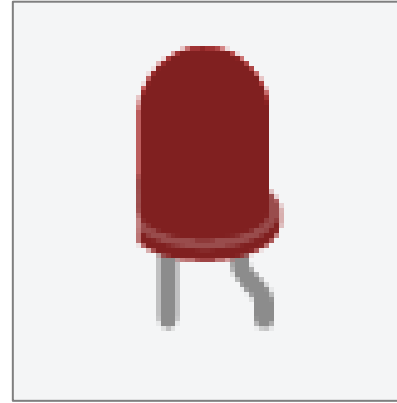
setup() 함수; 처음 설정하기

- 아두이노 보드에 전원을 넣고 프로그램이 처음 시작될 때 한 번만 실행돼요.
- 여기에는 핀 모드 설정, 라이브러리 초기화, 시리얼 통신 시작 등 처음에 필요한 설정을 위한 코드를 넣어요.

loop() 함수; 반복하기

- setup() 함수가 실행된 후, loop() 함수 안에 있는 코드는 계속 반복해서 실행돼요.
- 여기에는 아두이노가 해야 할 일을 넣어요.

예) 센서 읽기, 모터 움직이기, LED 깜빡이기 등 반복해서 해야 하는 동작을 프로그램해요.



1. LED란



ChatGPT

LED(Light Emitting Diode)는 전기가 통과할 때 빛을 방출하는 반도체 소자입니다. 전기가 흐르면 LED는 전자와 양전자 사이의 결합으로 인해 빛을 방출합니다. LED는 주로 조명, 디스플레이, 신호 표시 등 다양한 용도로 사용됩니다. 일반 조명 소스에 비해 저전력, 장수명, 고효율 등의 장점을 가지고 있어 많은 분야에서 널리 사용되고 있습니다. LED는 다양한 색상과 밝기로 제공되며, 이러한 특성들은 다양한 응용 분야에서 사용되고 있습니다.



ChatGPT

아두이노(Arduino)는 오픈 소스 하드웨어 및 소프트웨어 플랫폼으로, 다양한 프로젝트를 위해 사용되는 마이크로컨트롤러 기반의 개발 보드입니다. 아두이노 보드를 사용하여 LED(또는 여러 다른 장치)를 제어할 수 있습니다.

아두이노에서 LED를 제어하는 가장 간단한 방법 중 하나는 디지털 핀을 사용하는 것입니다. 아두이노 보드에는 디지털 핀이 있으며, 이러한 핀을 사용하여 LED를 제어할 수 있습니다.

LED를 제어하려면 다음 단계를 따릅니다:

1. 아두이노 보드에 LED를 연결합니다. 일반적으로 LED는 긴 다리(양극)와 짧은 다리(음극)가 있습니다. 긴 다리는 전원(+)에 연결되고, 짧은 다리는 접지(-)에 연결됩니다.



[아두이노 LED와 단비 LED]

```
cpp Copy code

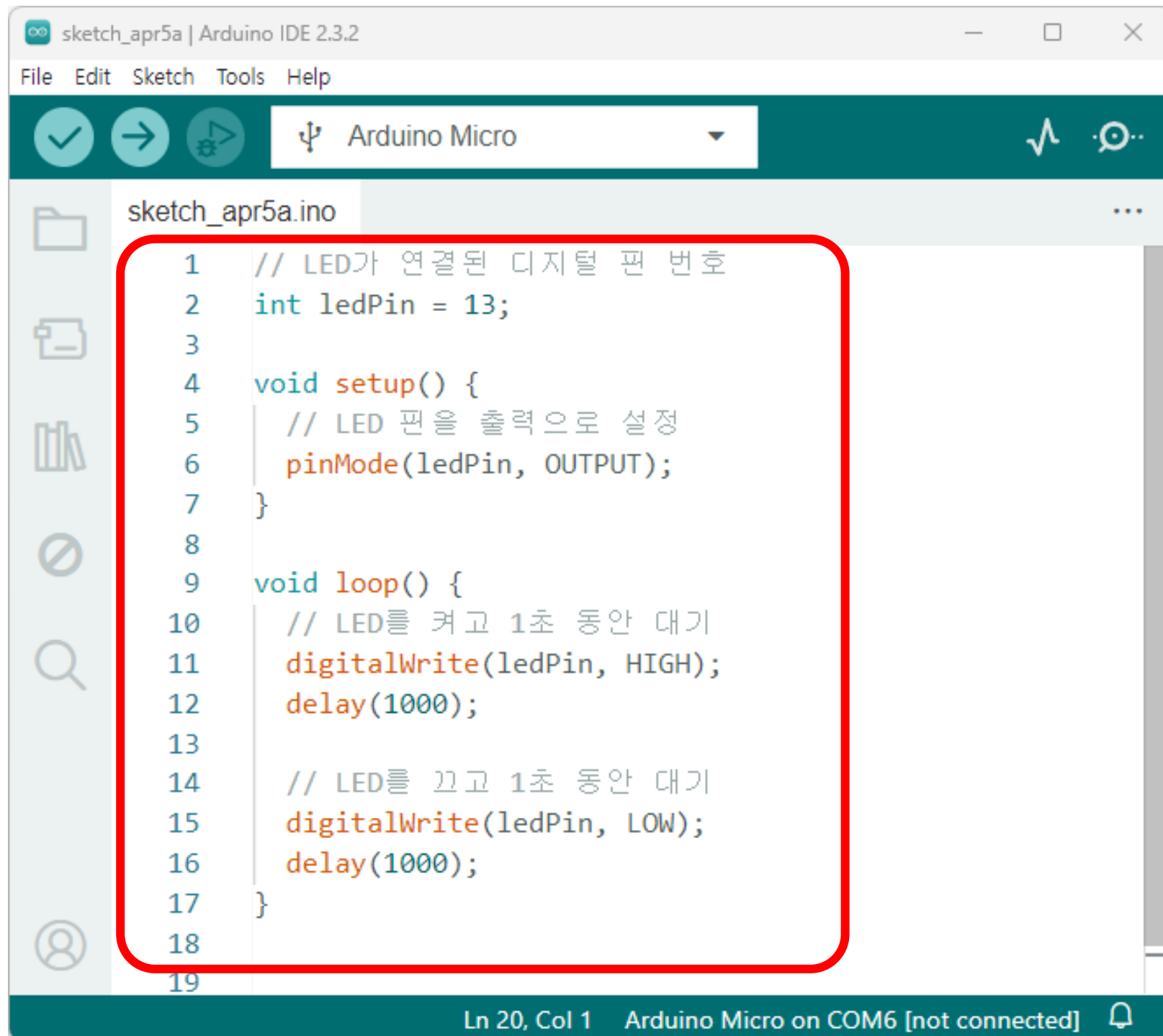
// LED가 연결된 디지털 핀 번호
int ledPin = 13;

void setup() {
  // LED 핀을 출력으로 설정
  pinMode(ledPin, OUTPUT);
}

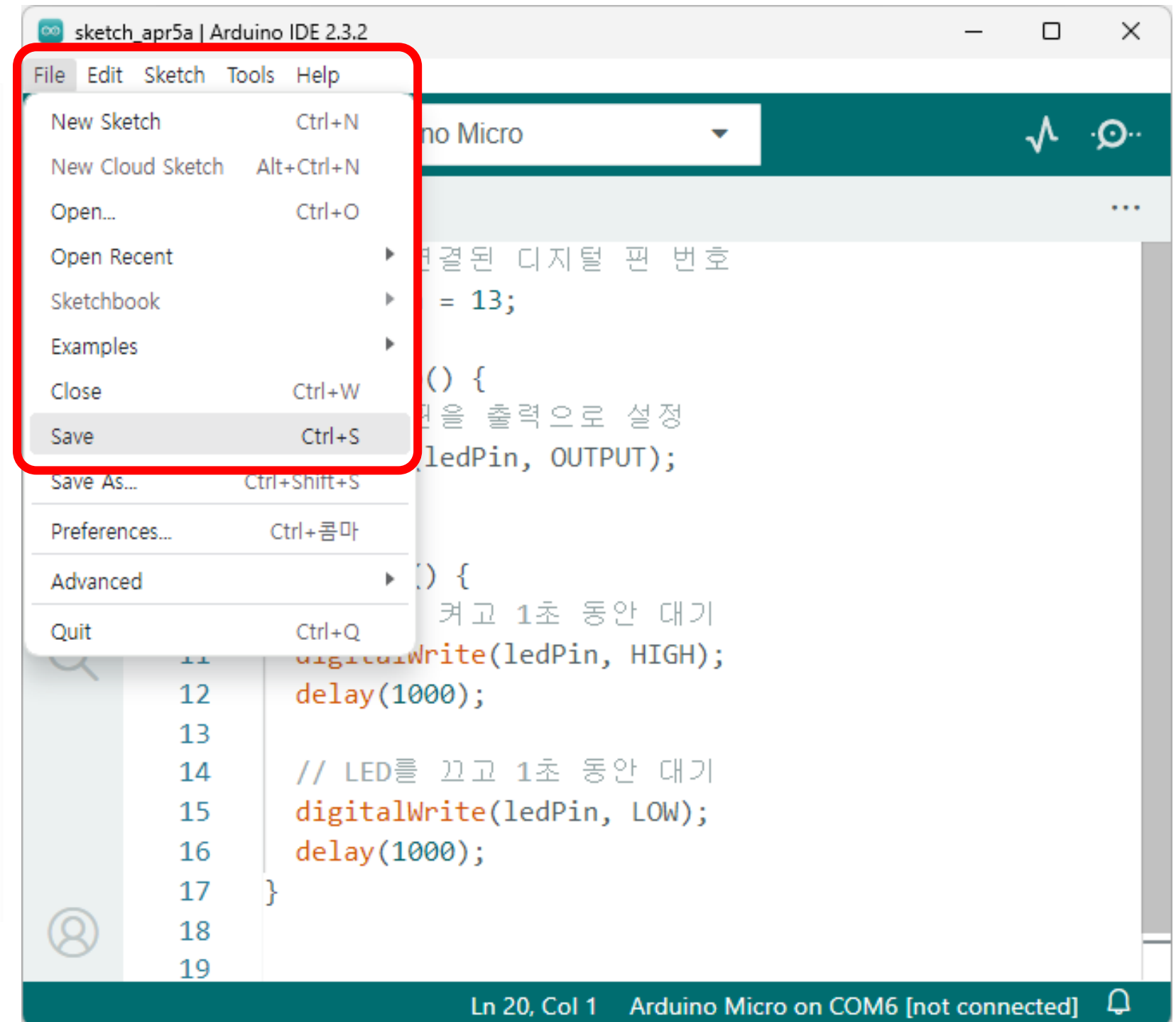
void loop() {
  // LED를 켜고 1초 동안 대기
  digitalWrite(ledPin, HIGH);
  delay(1000);

  // LED를 끄고 1초 동안 대기
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```


- 스케치에서 코드를 작성하고, [File]-[Save]로 저장하는 방법도 알아보자.



```
1 // LED가 연결된 디지털 핀 번호
2 int ledPin = 13;
3
4 void setup() {
5   // LED 핀을 출력으로 설정
6   pinMode(ledPin, OUTPUT);
7 }
8
9 void loop() {
10  // LED를 켜고 1초 동안 대기
11  digitalWrite(ledPin, HIGH);
12  delay(1000);
13
14  // LED를 끄고 1초 동안 대기
15  digitalWrite(ledPin, LOW);
16  delay(1000);
17 }
18
19
```

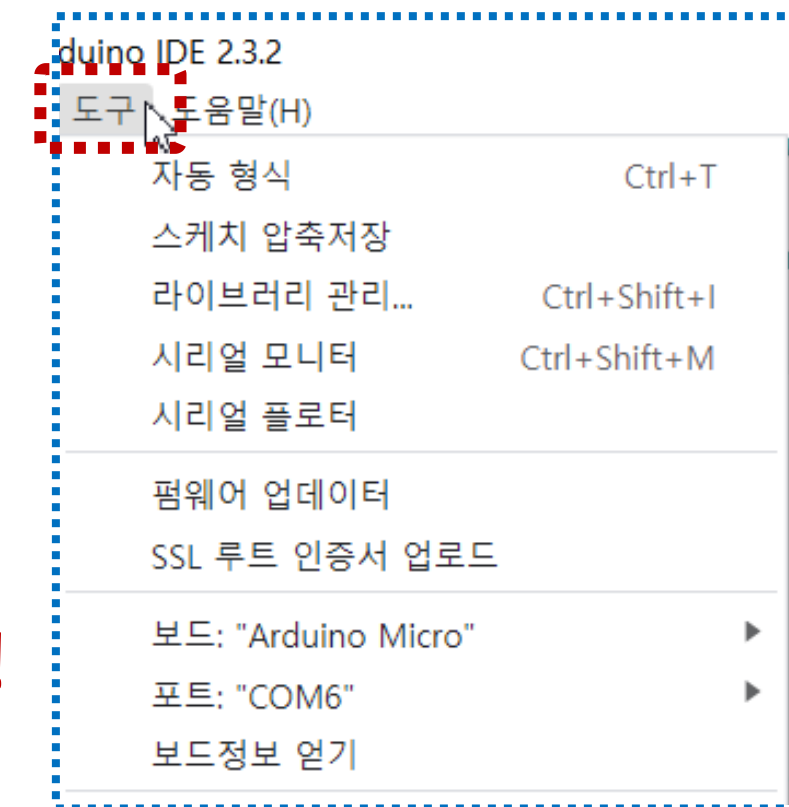
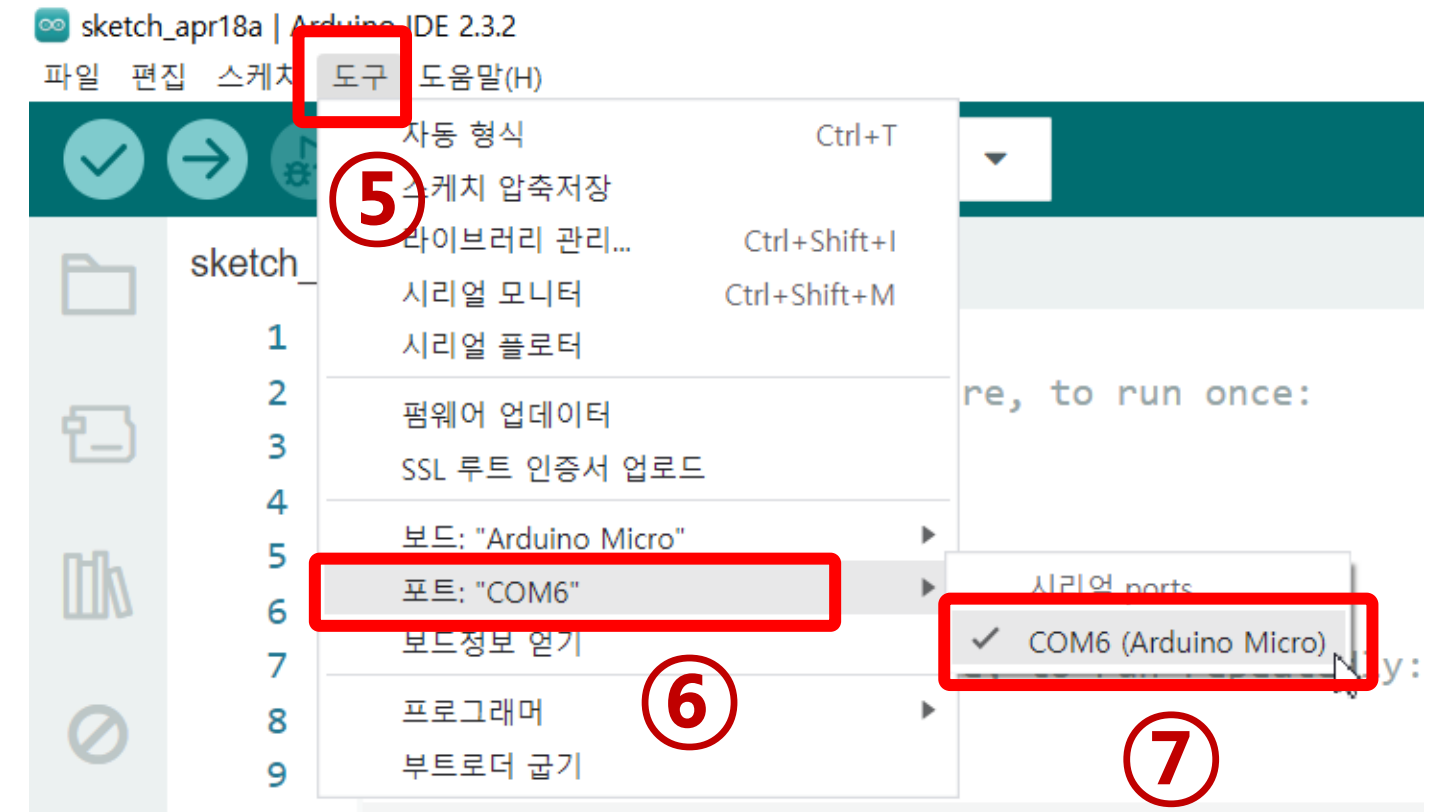
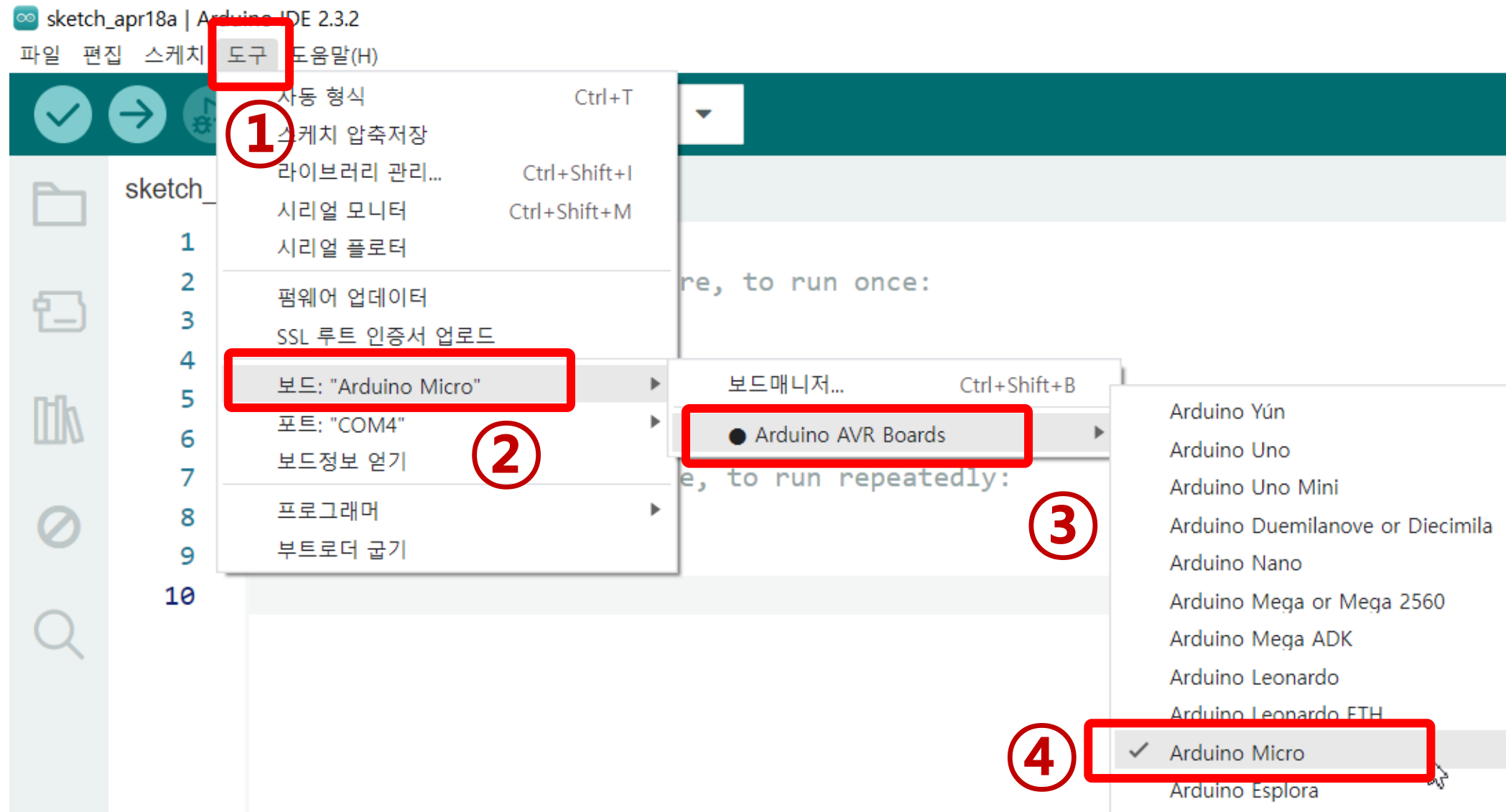


```
File Edit Sketch Tools Help
New Sketch Ctrl+N
New Cloud Sketch Alt+Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Preferences... Ctrl+콤마
Advanced
Quit Ctrl+Q
```

🌀 단비 로봇 LED 켜봐요!



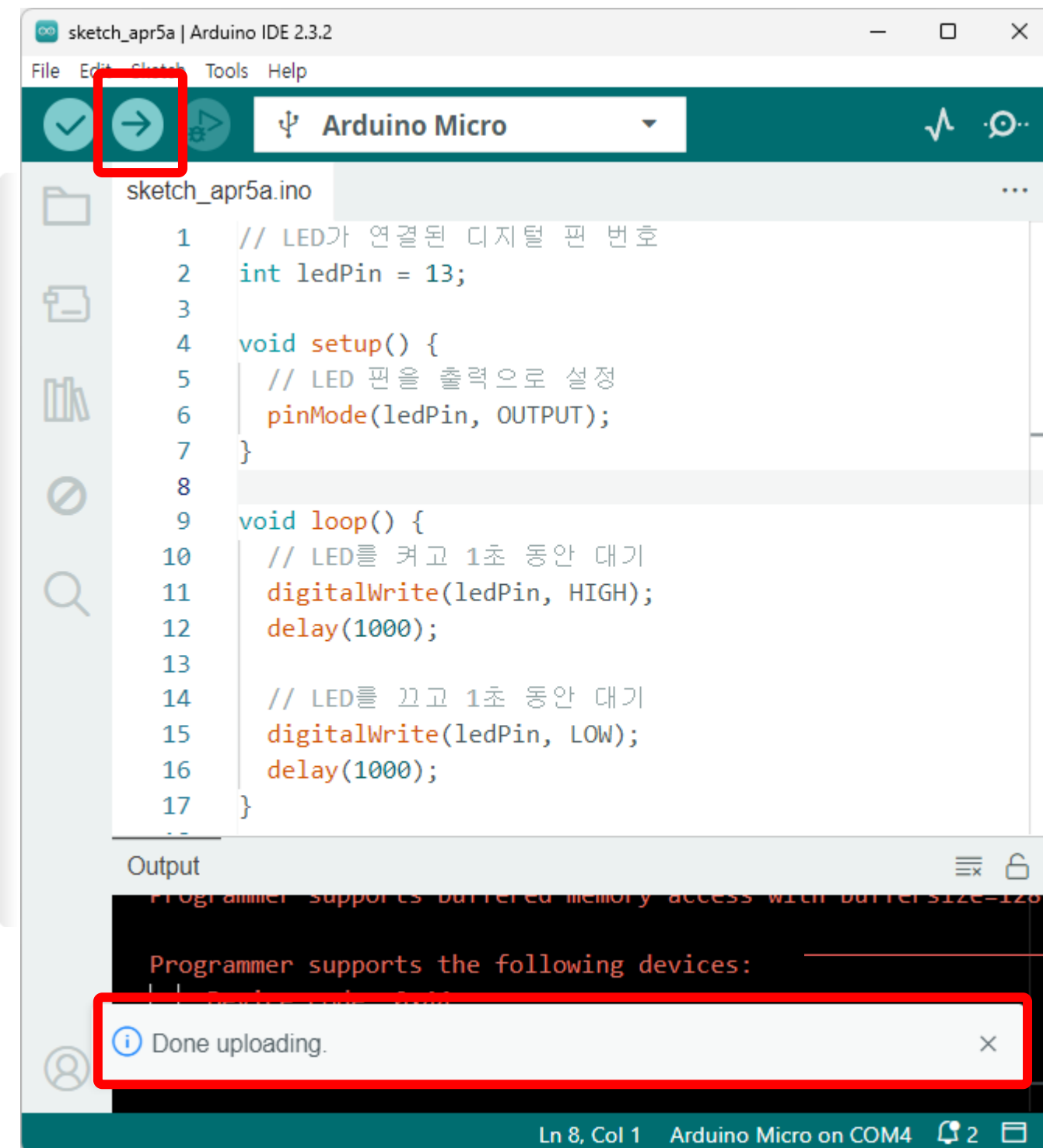
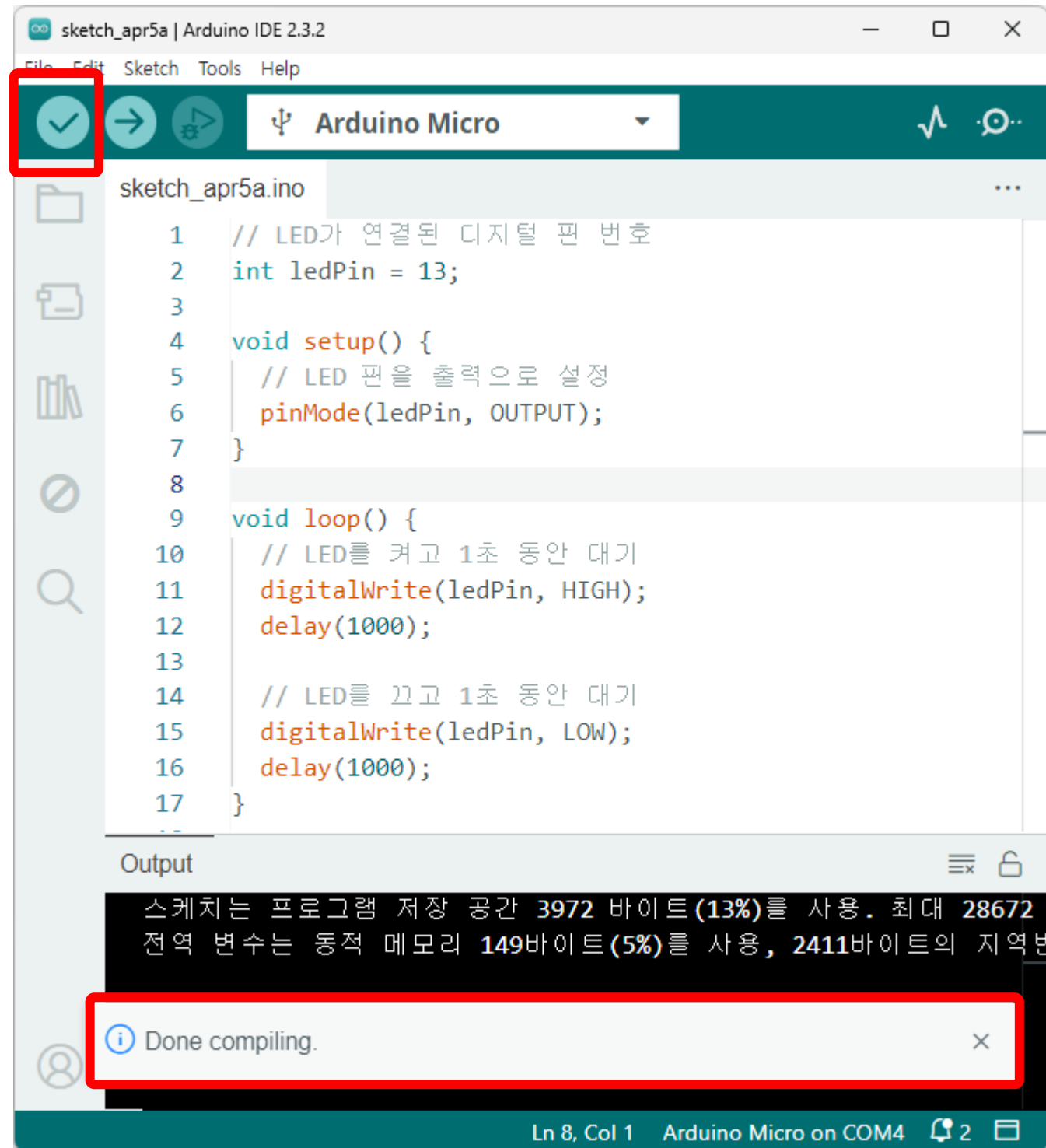
- “단비”와 케이블을 연결한 후, 전원을 받드시 켜야 합니다.



설정 완료 후 확인해 보세요!

Chapter.3 LED 켜는 프로그램 작성후, 컴파일과 업로드 해보기

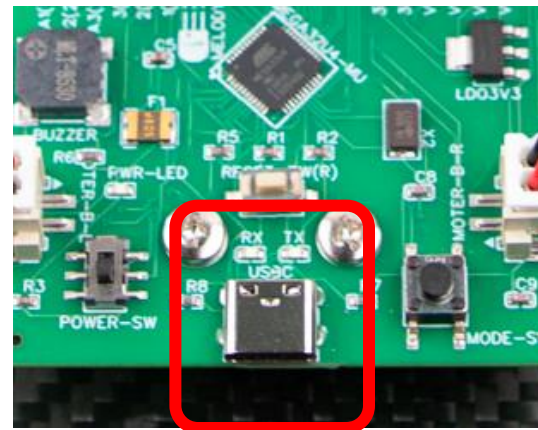
- 프로그램 작성후 **체크 버튼을 누르면 컴파일**이 완료됩니다.
- 오른쪽 화살표는 컴파일과 업로드를 동시에 가능하므로, 오류가 없다면 **오른쪽 화살표만 눌러도 되요!**



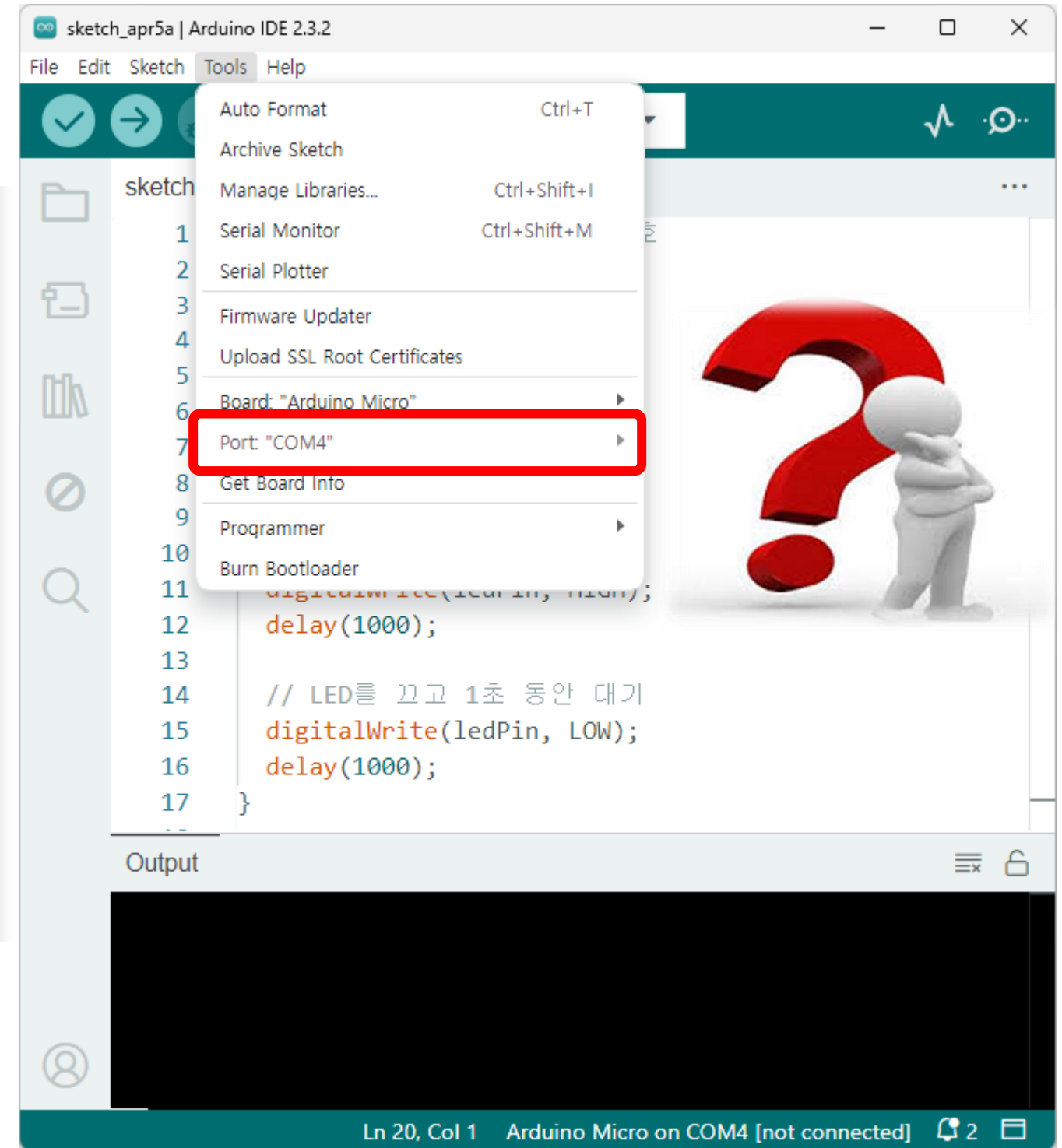
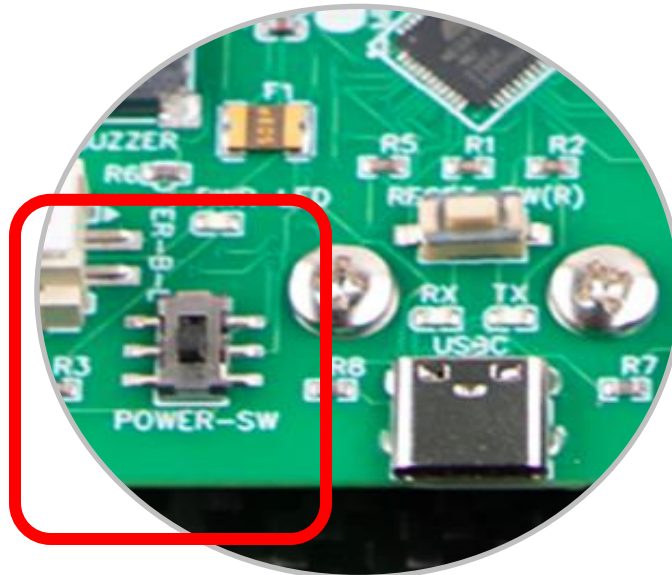
오류아님!
Version 설명

잠깐!! 포트가 활성화 안되나요?

확인 1. C-Type 케이블이 연결되었나요?



확인 2. 전원스위치를 켜나요?

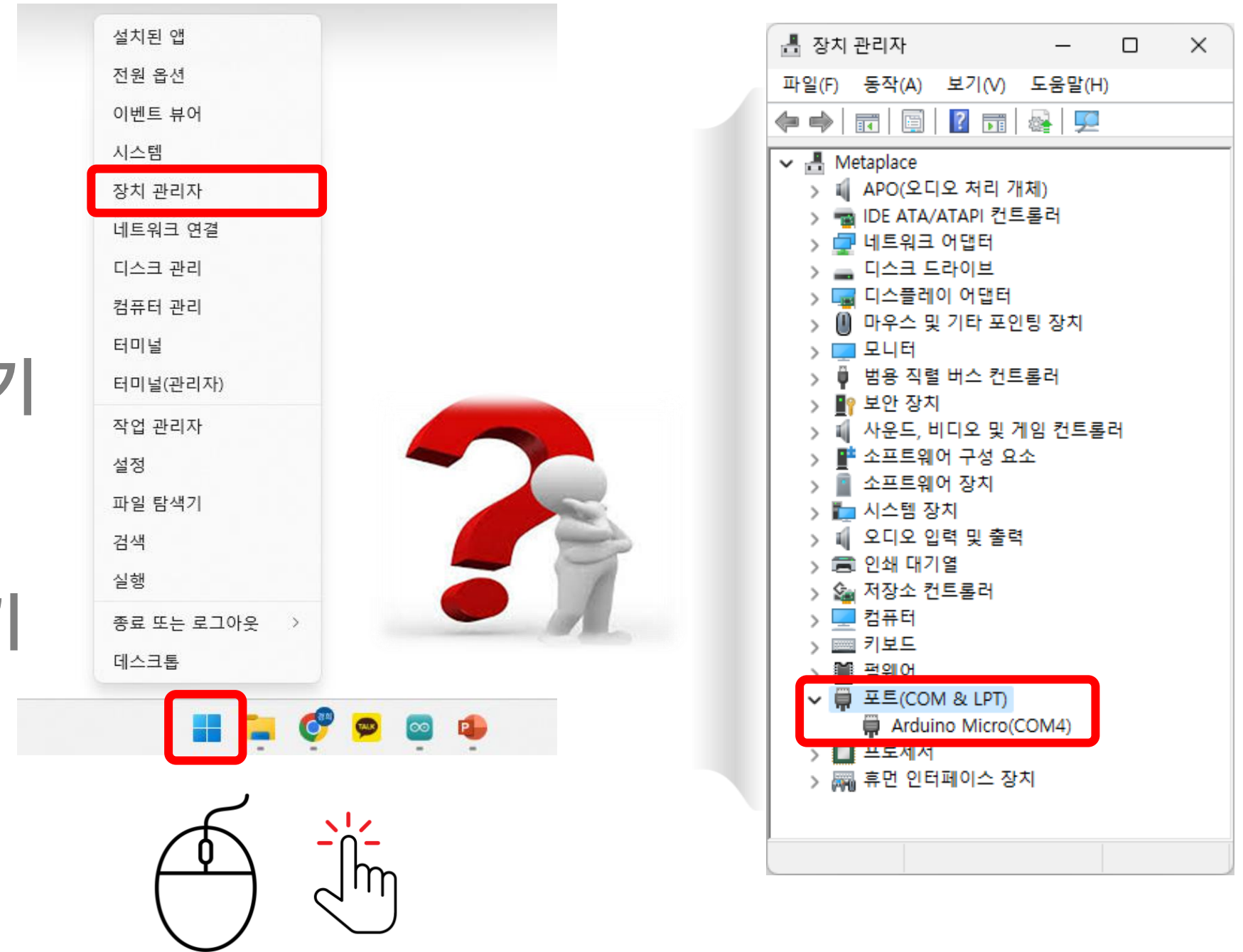


잠깐!! 포트가 활성화 안되나요?

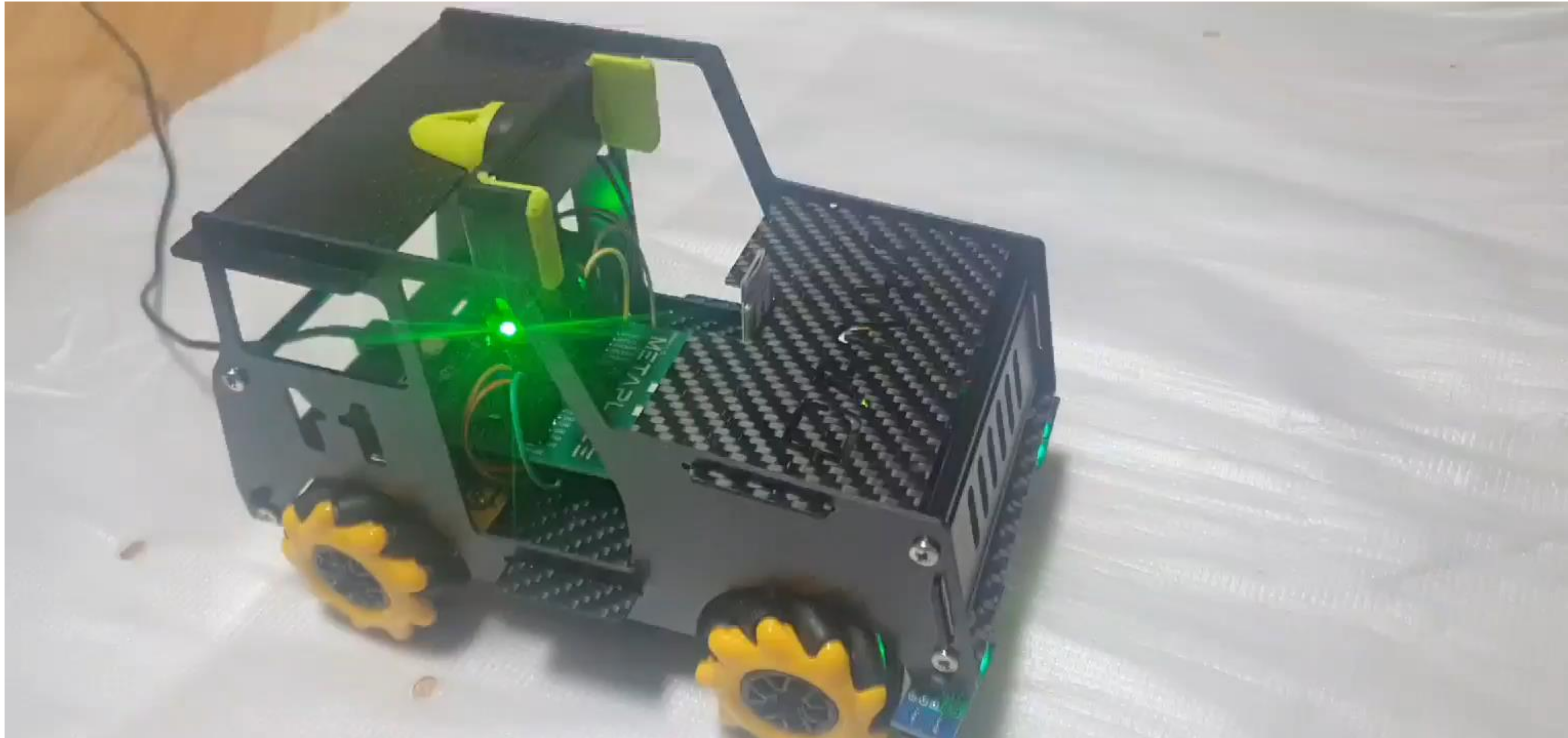
확인 3.

포트가 연결되었는지 장치관리자에서 확인하기

해결방법 : 포트가 활성화 되어 있지 않다면,
Ch340 드라이버를 인터넷에서 설치하기



🌀 단비의 버튼을 클릭하면 LED 라이트 켜봐요!



```

class_00_01_mode_sw_led | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
class_00_01_mode_sw_led.ino
1 // 디지털핀 설계
2 #define led_pin 13 // 13번 디지털핀(led_pin)
3 // 아날로그핀 설계
4 #define button_pin A0 // A0번 아날로그핀(button_pin)
5 // 버튼상태 변수 선언
6 int button_state = 0;
7
8 // 부팅시 만 실행
9 void setup()
10 {
11 // 시리얼통신 초기화(9600)bps(bits per second)
12 Serial.begin(9600);
13 // 디지털핀 초기화(OUTPUT)
14 pinMode(led_pin, OUTPUT); // led_pin 13번 설정
15 }
16

Output
Ln 31, Col 1 Arduino Micro on COM4 2

```



LED 핀번호 : 13



버튼 핀번호 : A0


```
22 // 메인(반복루프)
23 void loop()
24 {
25     // 모드스위치(푸쉬스위치) 눌림 512이상 입력값 생성
26     // 아날로그 입력값[0~1023]
27     if (analogRead(button_pin) > 512)
28     {
29         // 모드스위치(푸쉬스위치) 누르고 기다리는 시간
30         delay(500);
31
32         if (button_state == 0)           // 만약 버튼의 상태가 0일경우
33         {
34             button_state = 1;           // 버튼의 상태를 1로 변경하고
35             digitalWrite(led_pin, HIGH); // LED핀 13번을 켜준다.
36             Serial.println("LED ON");   // 시리얼모니터에서 상태를 확인한다.
37         }
38         else if (button_state == 1)     // 만약 버튼의 상태가 1일경우
39         {
40             button_state = 0;           // 버튼의 상태를 0으로 변경하고
41             digitalWrite(led_pin, LOW); // LED핀 13번을 끈다.
42             Serial.println("LED OFF");   // 시리얼모니터에서 상태를 확인한다.
43         }
44     }
45 }
```



LED 꺼짐



버튼 누름



LED 켜짐



버튼 누름



LED 꺼짐

② 확인

① Micro, port 확인

③ 업로드

④ Done uploading. 확인하기

```

6 pinMode(ledPin, OUTPUT);
7 }
8
9 void loop() {
10 // LED를 켜고 1초 동안 대기
11 digitalWrite(ledPin, HIGH);
12 delay(1000);
13
14 // LED를 끄고 1초 동안 대기
15 digitalWrite(ledPin, LOW);
16 delay(1000);
17 }
    
```

Output

Programmer supports the following devices:

Arduino Micro on COM4

오류 아님!
Version 설명

⑤ 시리얼 모니터 클릭

```

class_00_01_mode_sw_led.ino
3 // 아두이노 핀 번호
4 #define button_pin 2
5 // 버튼상태 변수 선언
6 int button_state = 0;
7
8 // 부팅시 만 실행
9 void setup()
10 {
11 // 시리얼통신 초기화(9600)bps(bits per second)
12 Serial.begin(9600);
13 // 디지털핀 초기화(OUTPUT)
14 pinMode(led_pin, OUTPUT); // led_pin 13번 설정
15 }
16
17
18
19
20
21
    
```

⑥ 결과 확인

Serial Monitor

Message (Enter to send message to 'Arduino Micro') New Line 9600 baud

LED ON
LED OFF

Ln 48, Col 1 Arduino Micro on COM4

🌀 아날로그 값 출력과 시리얼 통신에 대한 명령어를 알아봐요!

```
#define led_pin 13 // 'led_pin'이라는 이름을 13번 핀으로 정해요.
```

```
analogRead(A0); // A0 핀에서 값을 읽어와요.
```

아두이노의 아날로그 입력 값은 0에서 1023 사이에 있어요.

```
Serial.begin(9600); // 시리얼 통신을 9600 속도로 시작해요.
```

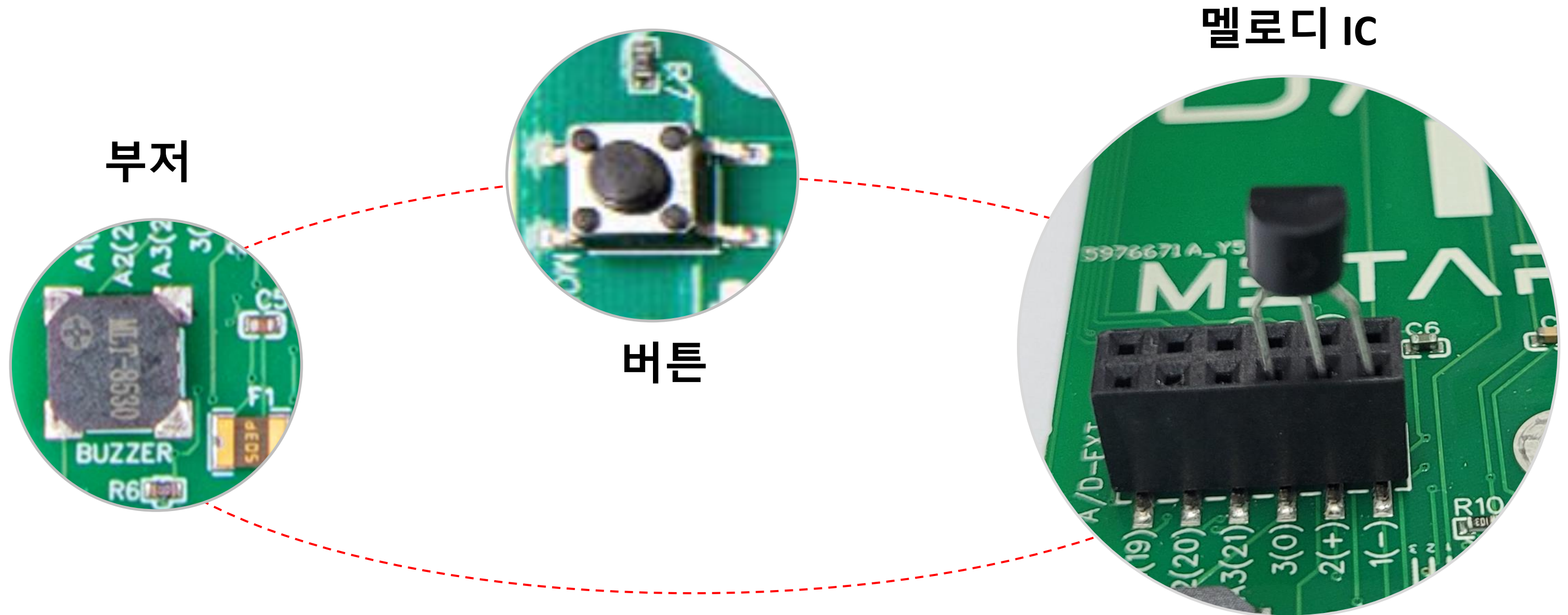
이것은 아두이노와 컴퓨터가 서로 대화할 수 있게 해줘요.

```
Serial.println("LED ON"); // "LED ON"이라는 글자를 시리얼 모니터에 보여줘요.
```

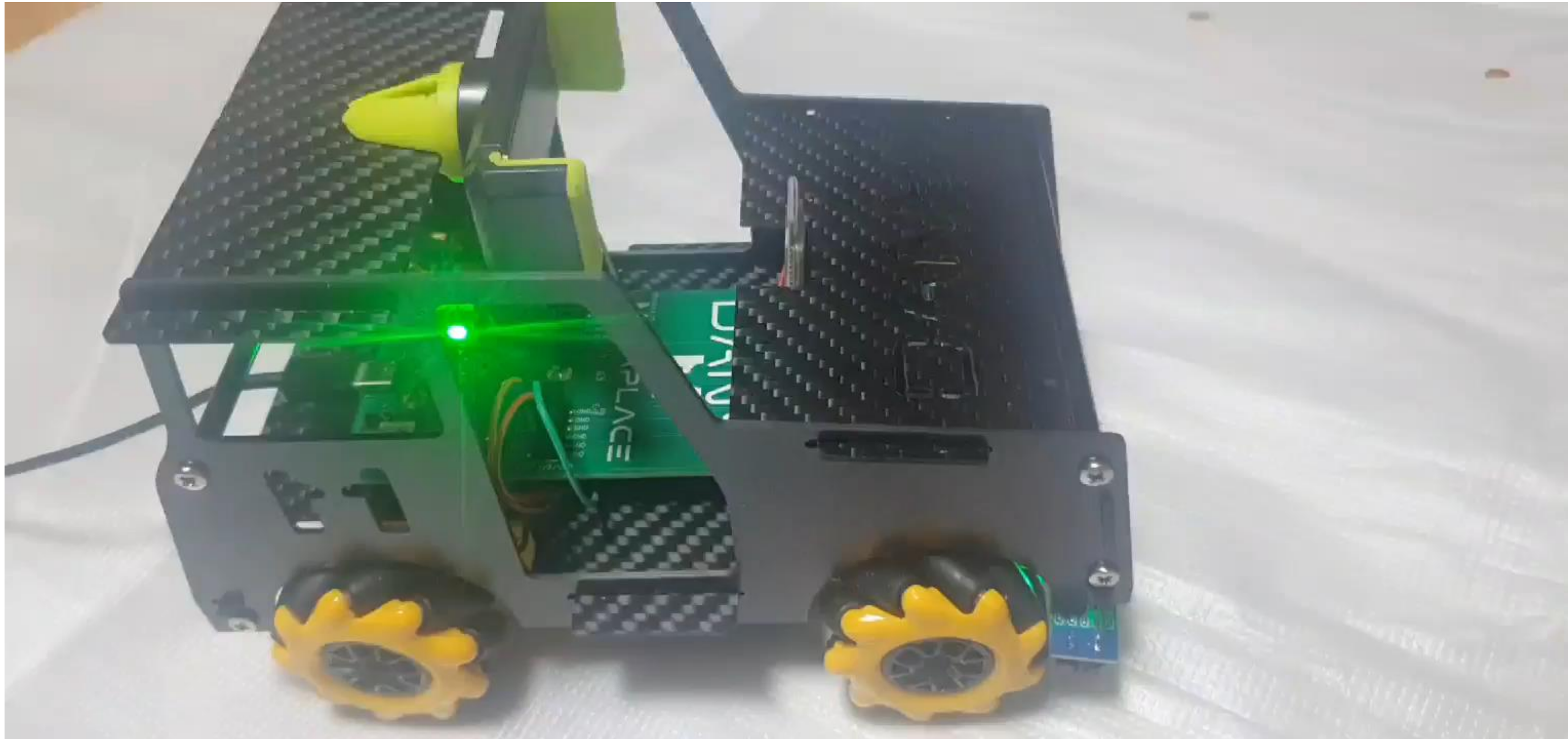
프로그램이 잘 작동하는지 확인할 때 유용해요.

- 보 레이트(baudrate) : 1초 동안 보낼 수 있는 데이터의 양을 말해요.
- 9600 : 1초 동안 9600비트의 데이터를 보낸다는 뜻이에요.

- 소형 멜로디 IC는 짧은 길이의 곡을 1개 저장하고 있다.
- 멜로디 IC는 전원이 연결되면 처음부터 다시 연주를 시작하게 된다.



🌀 단비의 버튼을 클릭하여 연주해봐요!

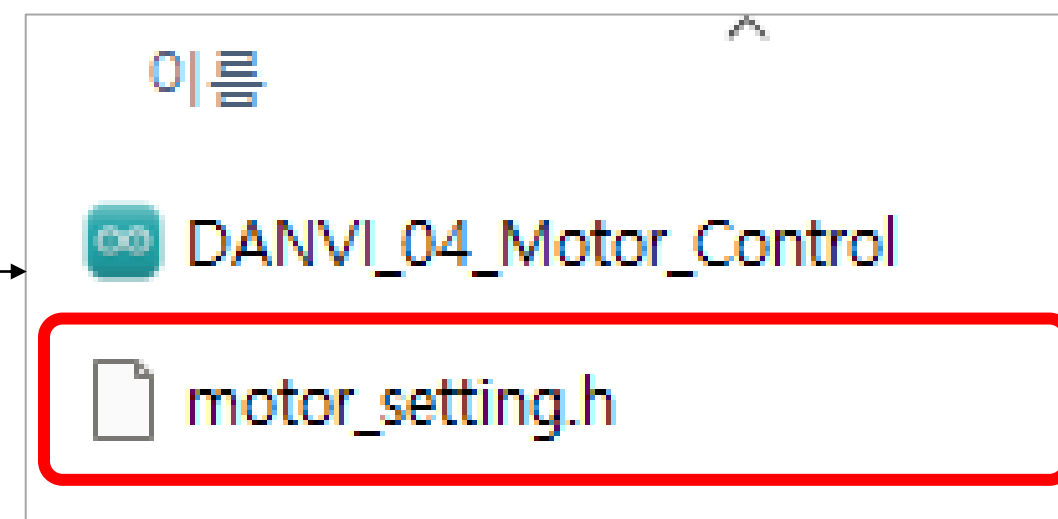
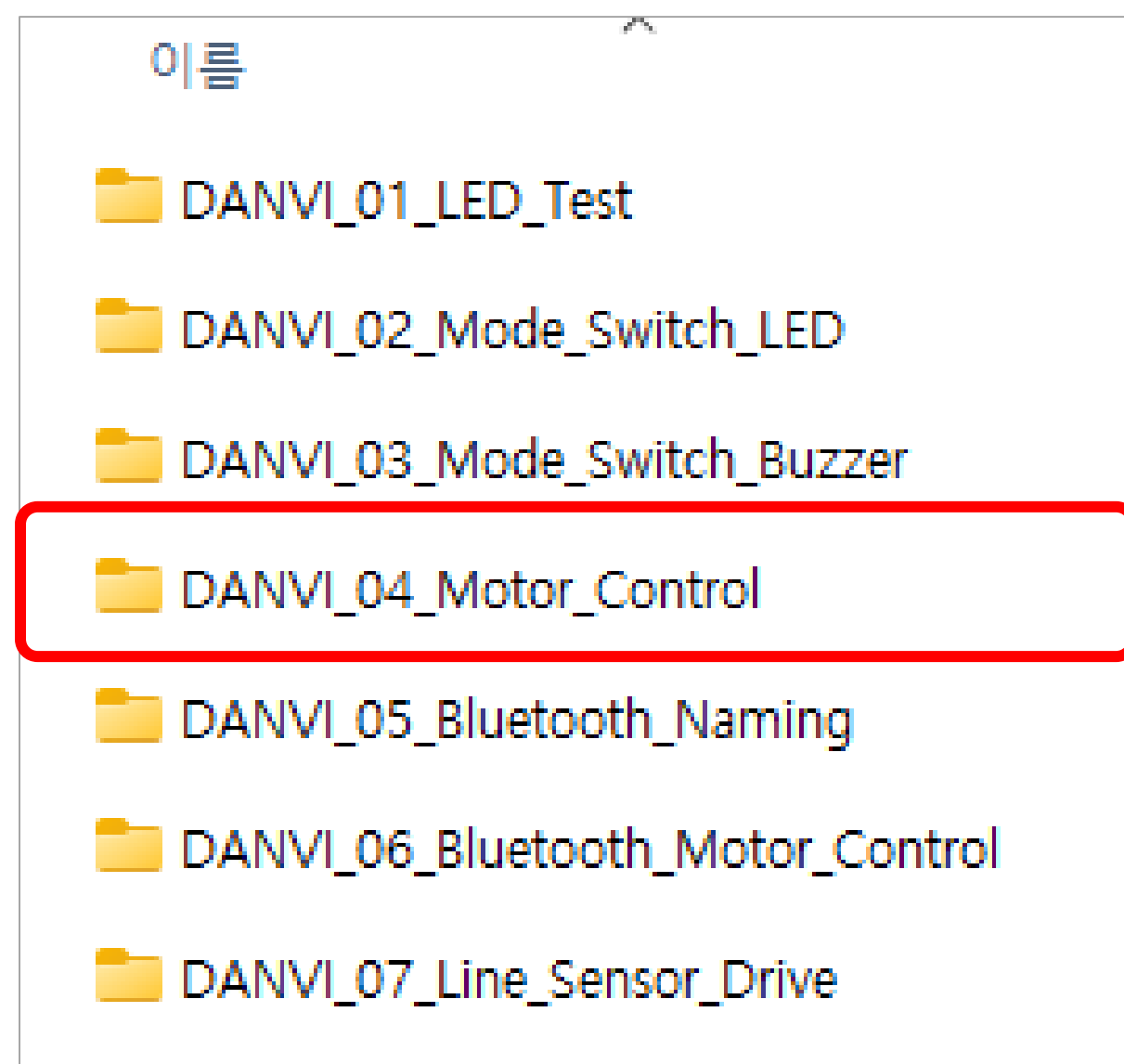


```
DANVI_03_Mode_Switch_Buzzer | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_03_Mode_Switch_Buzzer.ino
1 // 디지털핀 선제
2 #define buzzer_pin 6 // 6번 디지털핀(buzzer_pin)
3 // 아날로그핀 선제
4 #define button_pin A0 // A0번 아날로그핀(button_pin)
5 // 버튼상태 변수 선
6 int button_state = 0; // 버튼상태 변수 선언 초기값은 0
7
8 // 부팅시 만 실행
9 void setup()
10 {
11 // 시리얼통신 초기화(9600)bps(bits per second)
12 Serial.begin(9600);
13 // 디지털핀 초기화(OUTPUT)
14 pinMode(buzzer_pin, OUTPUT); // buzzer_pin 6번 설정
15 }
16
```

```
21 // 메인(반복루프)
22 void loop()
23 {
24 // 모드스위치(푸쉬스위치) 눌림 512이상 입력값 생성
25 // 아날로그 입력값[0~1023]
26 if (analogRead(button_pin) > 512)
27 {
28 // 모드스위치(푸쉬스위치) 누르고 기다리는 시간
29 delay(500);
30
31 if (button_state == 0) // 만약 버튼의 상태가 0일경우
32 {
33 button_state = 1; // 버튼의 상태를 1로 변경하고
34 // BUZZER ON
35 digitalWrite(buzzer_pin, HIGH); // 멜로디가 연주된다.
36 Serial.println("BUZZER ON"); // 시리얼모니터에서 상태를 확인한다.
37 }
38 else if (button_state == 1) // 만약 버튼의 상태가 1일경우
39 {
40 button_state = 0; // 버튼의 상태를 0으로 변경하고
41 // BUZZER OFF
42 digitalWrite(buzzer_pin, LOW); // 멜로디 연주를 끈다.
43 Serial.println("BUZZER OFF"); // 시리얼모니터에서 상태를 확인한다.
44 }
45 }
46 }
47
```

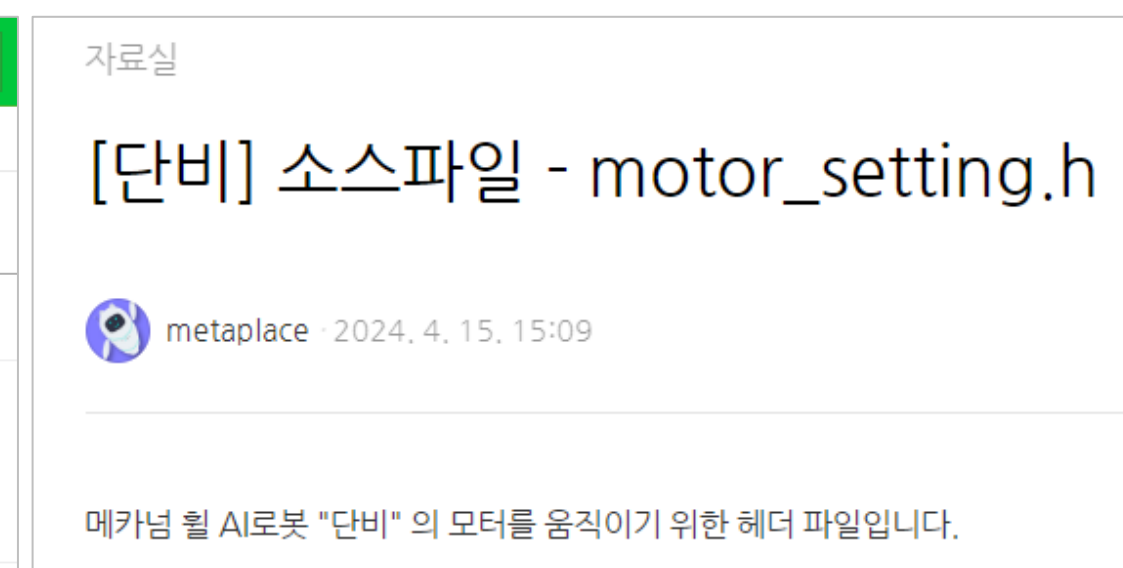
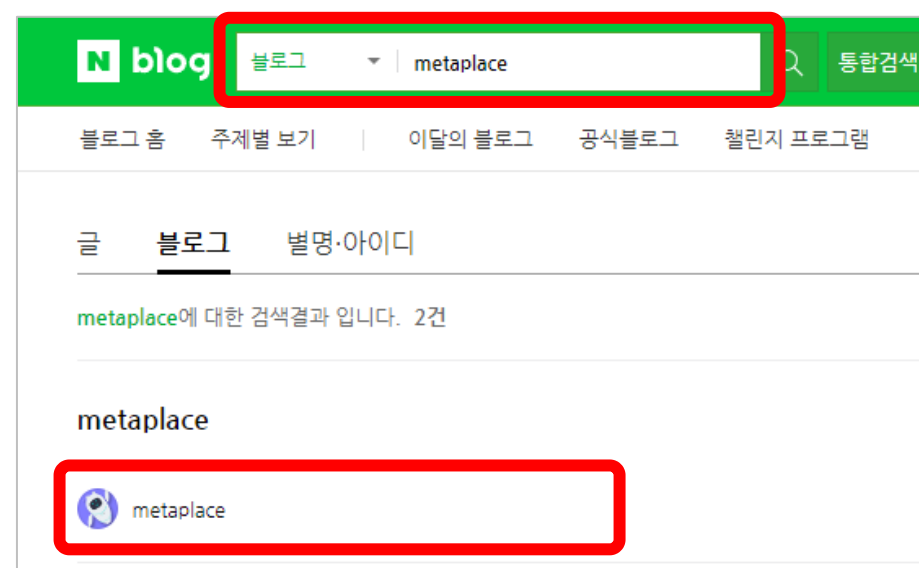
```
Output
[Output area is currently empty]
```

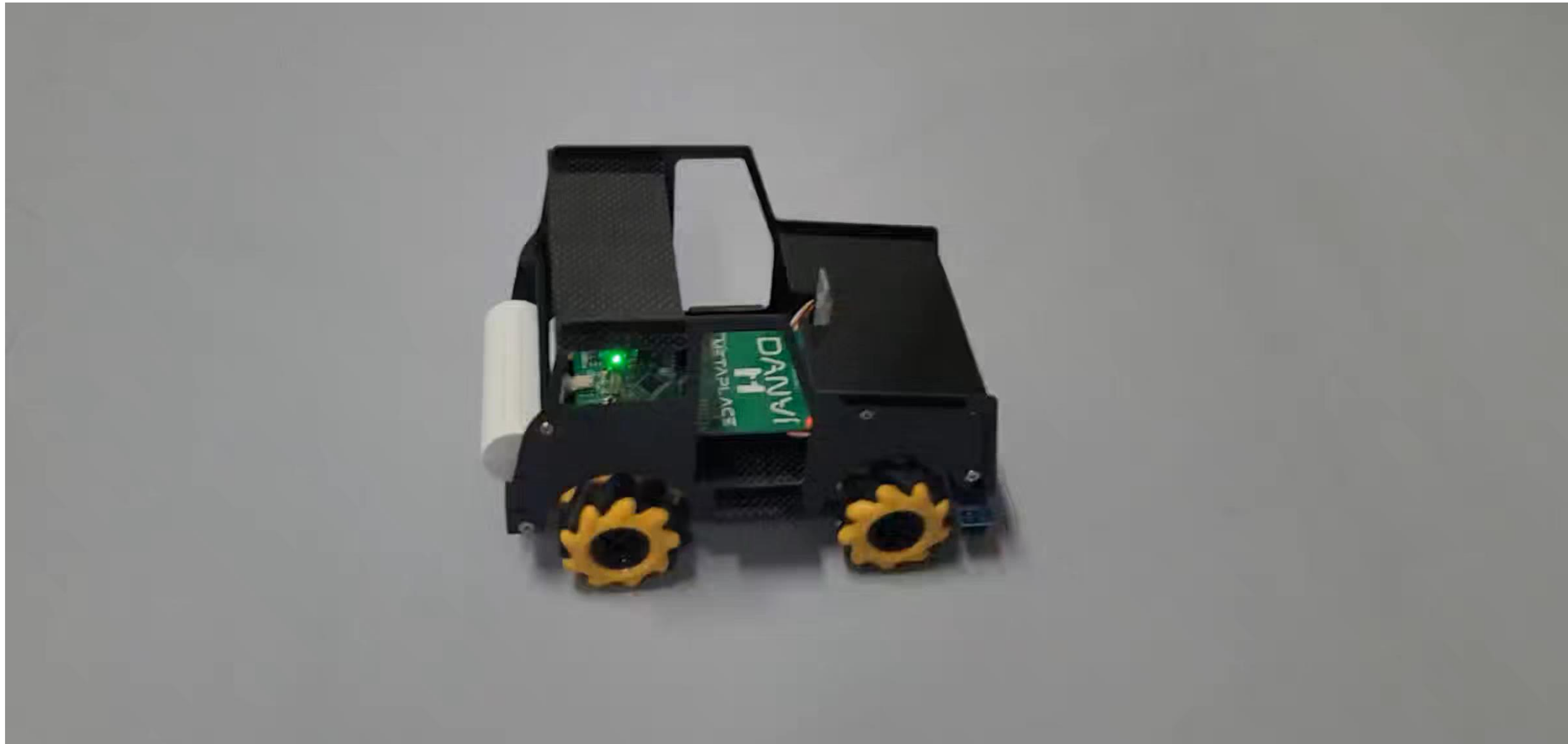

- 모터를 움직이기 위해서는 **motor_setting.h** 파일을 다운로드 한 후,
- 반드시 소스프로그램과 **같은 폴더안에** 넣어 놓아야 합니다.



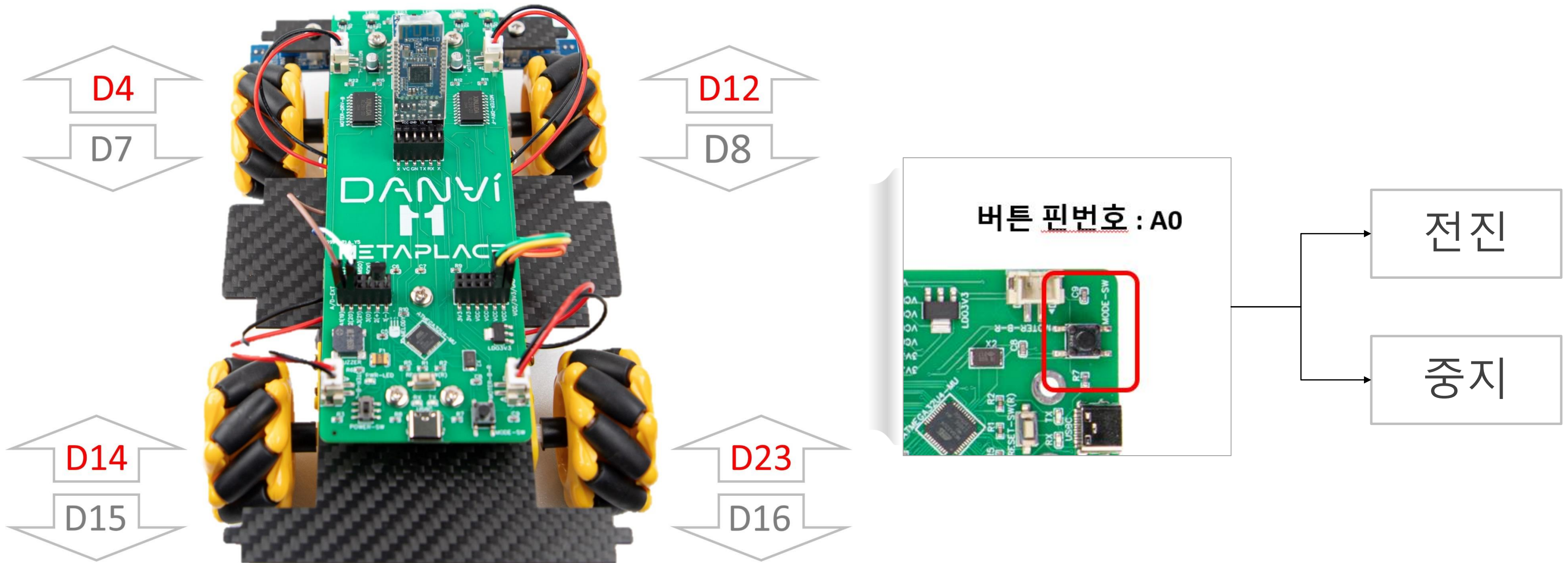
필요한 기능을 미리
함수로 만들어 파일에
저장해 두었어요!
필요할 때 그 함수를
불러와서 사용해요.

- 다운로드경로 : <https://blog.naver.com/metaplace>

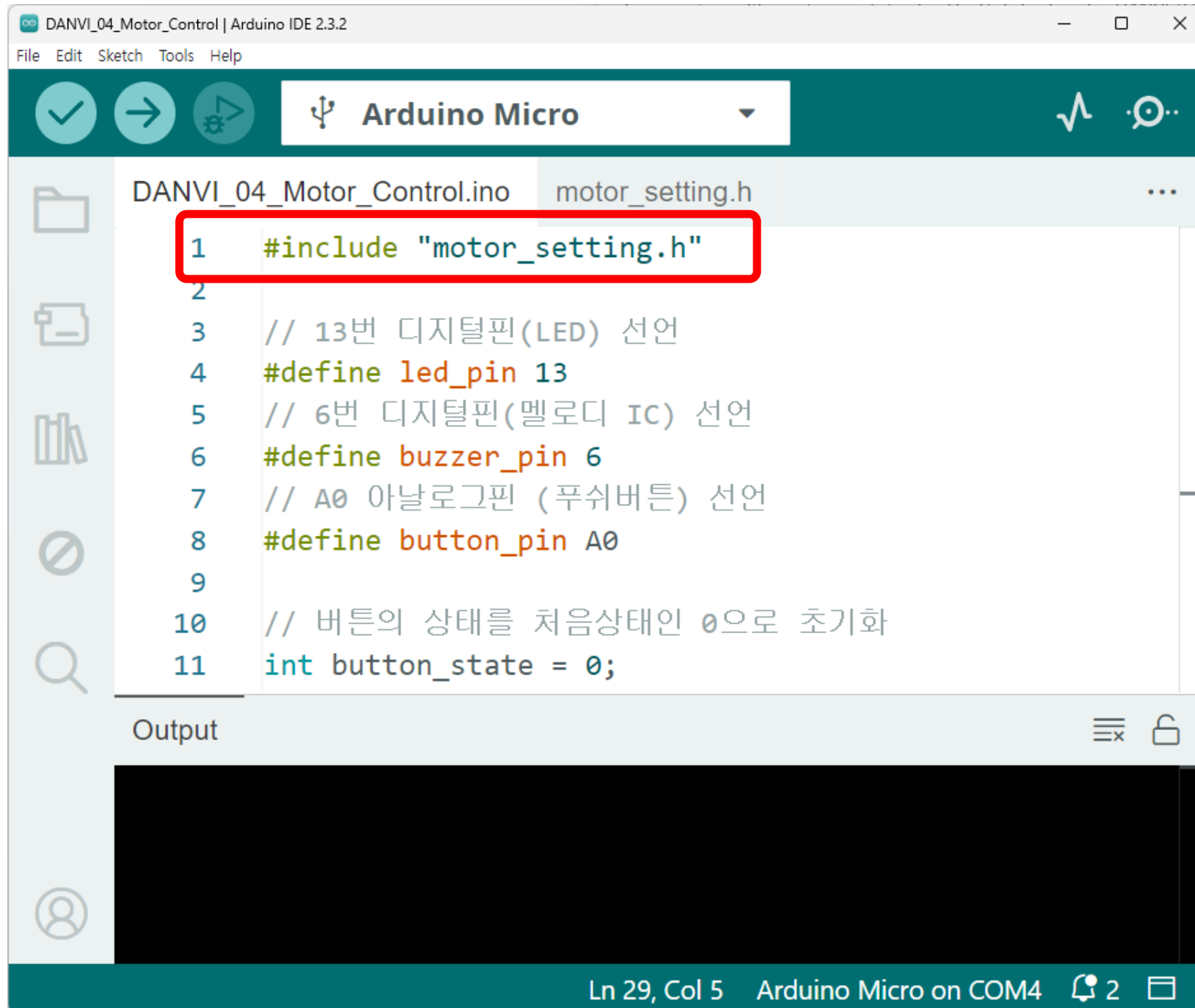




- 모터를 움직이기 위해서 버튼을 누르면 전진합니다.
- 버튼을 한번 더 누르면 주행하던 로봇이 정지합니다.

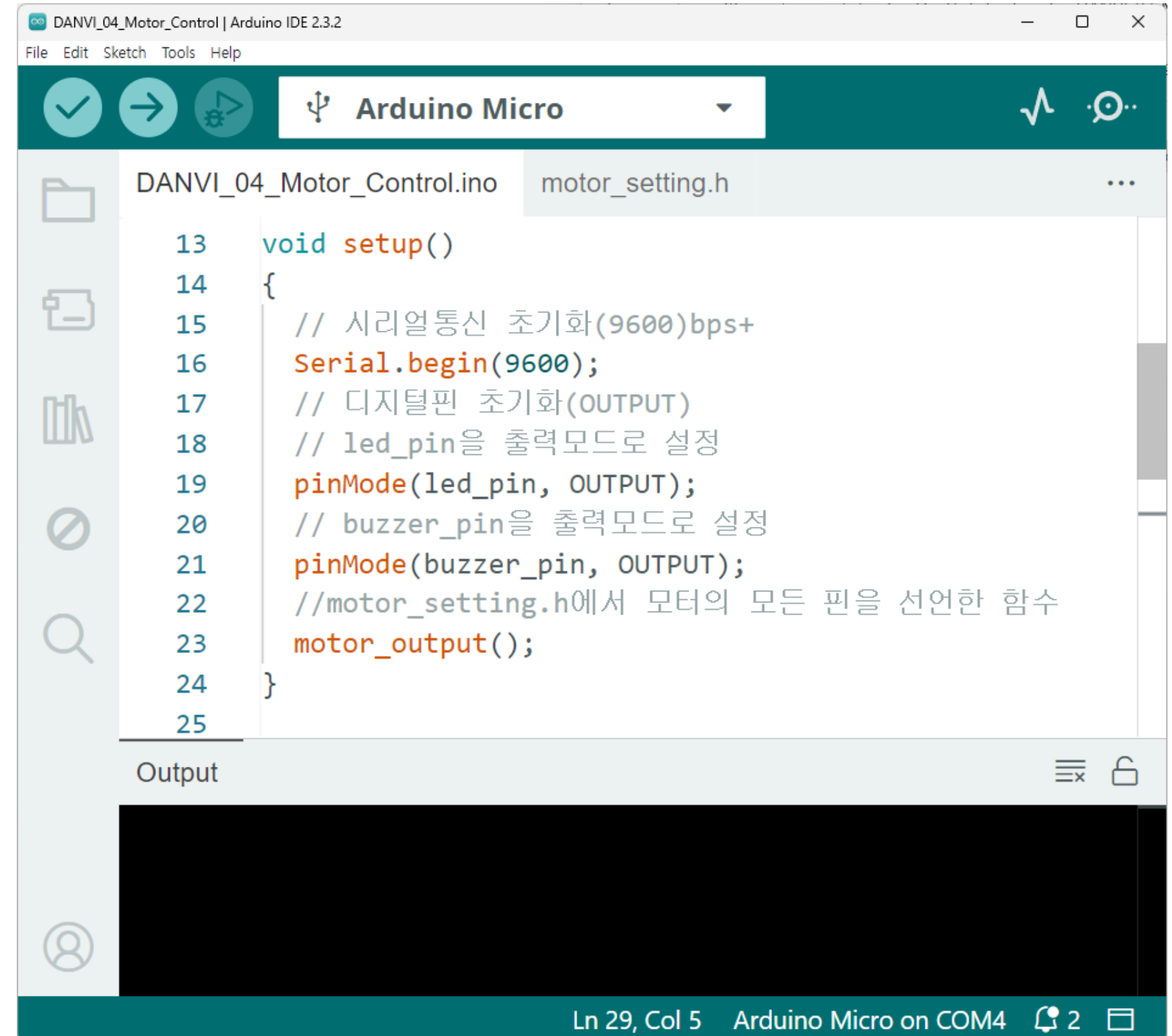


- // 주석문을 이해하고, 소스프로그램을 작성해볼게요.



The screenshot shows the Arduino IDE interface with the file 'motor_setting.h' open. The code includes several preprocessor directives and variable declarations. Line 1, '#include "motor_setting.h"', is highlighted with a red box. The code is as follows:

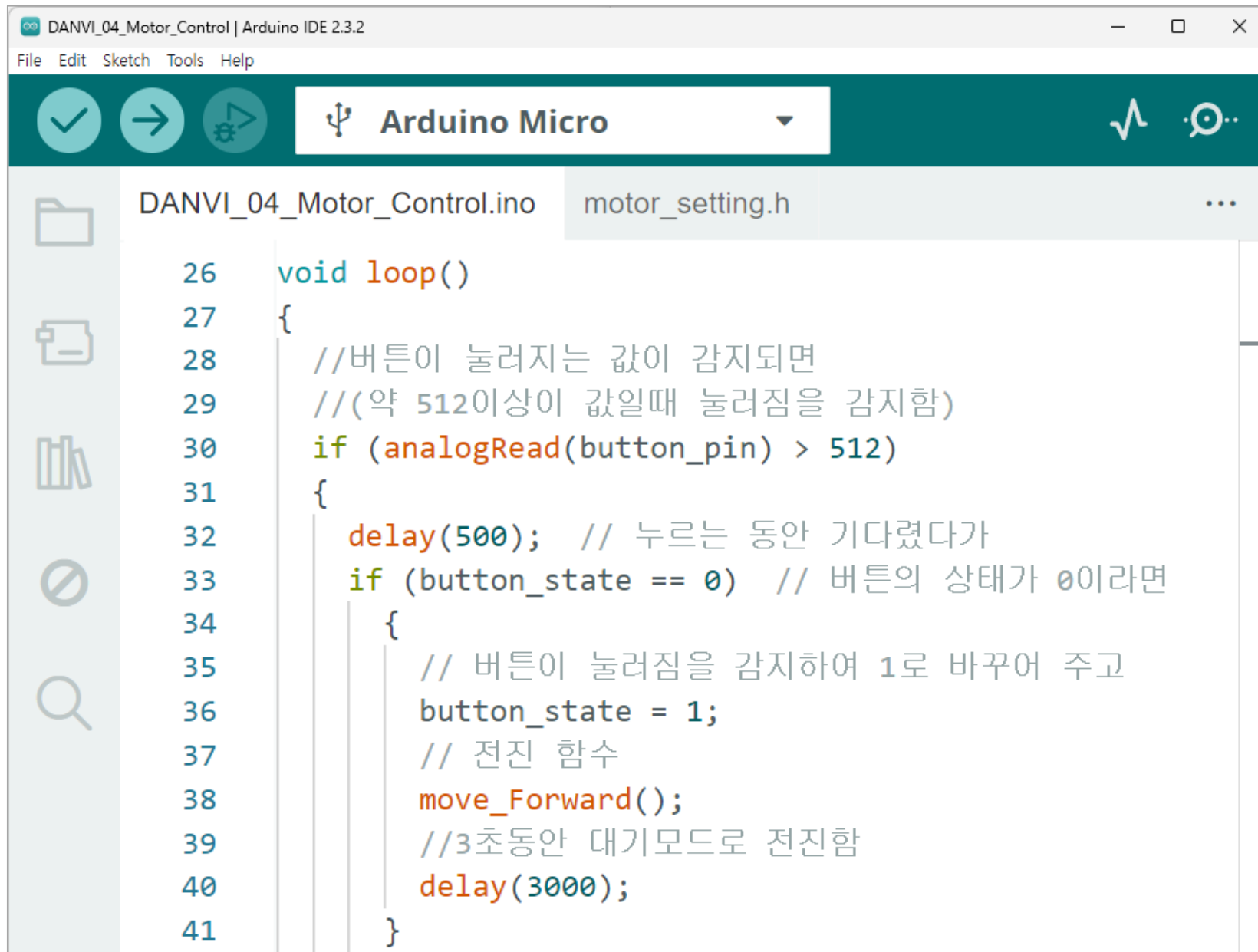
```
1  #include "motor_setting.h"
2
3  // 13번 디지털핀(LED) 선언
4  #define led_pin 13
5  // 6번 디지털핀(멜로디 IC) 선언
6  #define buzzer_pin 6
7  // A0 아날로그핀 (푸쉬버튼) 선언
8  #define button_pin A0
9
10 // 버튼의 상태를 처음상태인 0으로 초기화
11 int button_state = 0;
```



The screenshot shows the Arduino IDE interface with the main sketch file 'DANVI_04_Motor_Control.ino' open. The 'void setup()' function is visible, containing initialization code for the serial port and pins. The code is as follows:

```
13 void setup()
14 {
15     // 시리얼통신 초기화(9600)bps+
16     Serial.begin(9600);
17     // 디지털핀 초기화(OUTPUT)
18     // led_pin을 출력모드로 설정
19     pinMode(led_pin, OUTPUT);
20     // buzzer_pin을 출력모드로 설정
21     pinMode(buzzer_pin, OUTPUT);
22     //motor_setting.h에서 모터의 모든 핀을 선언한 함수
23     motor_output();
24 }
25
```


- 버튼을 누르면 A0핀 값이 512보다 커지면 눌러졌다는 인식을 하게됩니다.
- 초기상태가 0이면 1로 바뀌어지고 전진합니다. 상태가 1인 상태에서 다시한번 눌러 버튼상태가 0이 되면 로봇이 중지합니다~!



```
DANVI_04_Motor_Control | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_04_Motor_Control.ino motor_setting.h
26 void loop()
27 {
28   //버튼이 눌러지는 값이 감지되면
29   //(약 512이상 값일때 눌러짐을 감지함)
30   if (analogRead(button_pin) > 512)
31   {
32     delay(500); // 누르는 동안 기다렸다가
33     if (button_state == 0) // 버튼의 상태가 0이라면
34     {
35       // 버튼이 눌러짐을 감지하여 1로 바꾸어 주고
36       button_state = 1;
37       // 전진 함수
38       move_Forward();
39       //3초동안 대기모드로 전진함
40       delay(3000);
41     }
}
```



```
42   else if (button_state == 1)
43   {
44     button_state = 0;
45     Stop_Moter();
46   }
47 }
48 }
Output
Ln 6, Col 28 Arduino Micro on COM4 2
```

🌀 실행 하여 버튼을 눌러서 앞으로 주행을 확인해봐요!

- 1) [확인] 버튼을 눌러서 오류를 검사해요.
- 2) [업로드] 버튼을 눌러서 실행해요.



3) 컴퓨터와 연결된 케이블을 빼고, 보조 배터리를 연결해요.

- 4) 단비의 버튼을 눌러서 로봇이 앞으로 가는지 확인해요.

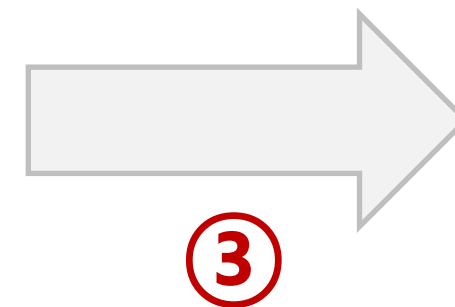


케이블 제거

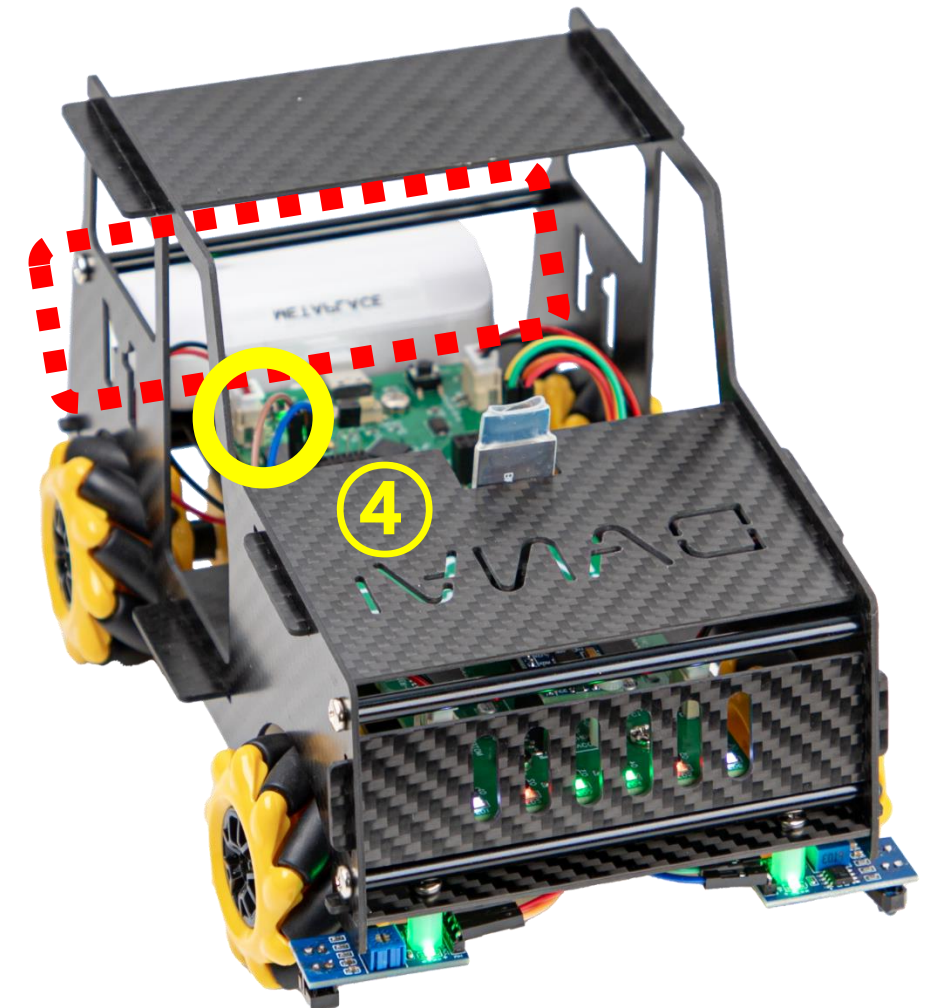


보조배터리

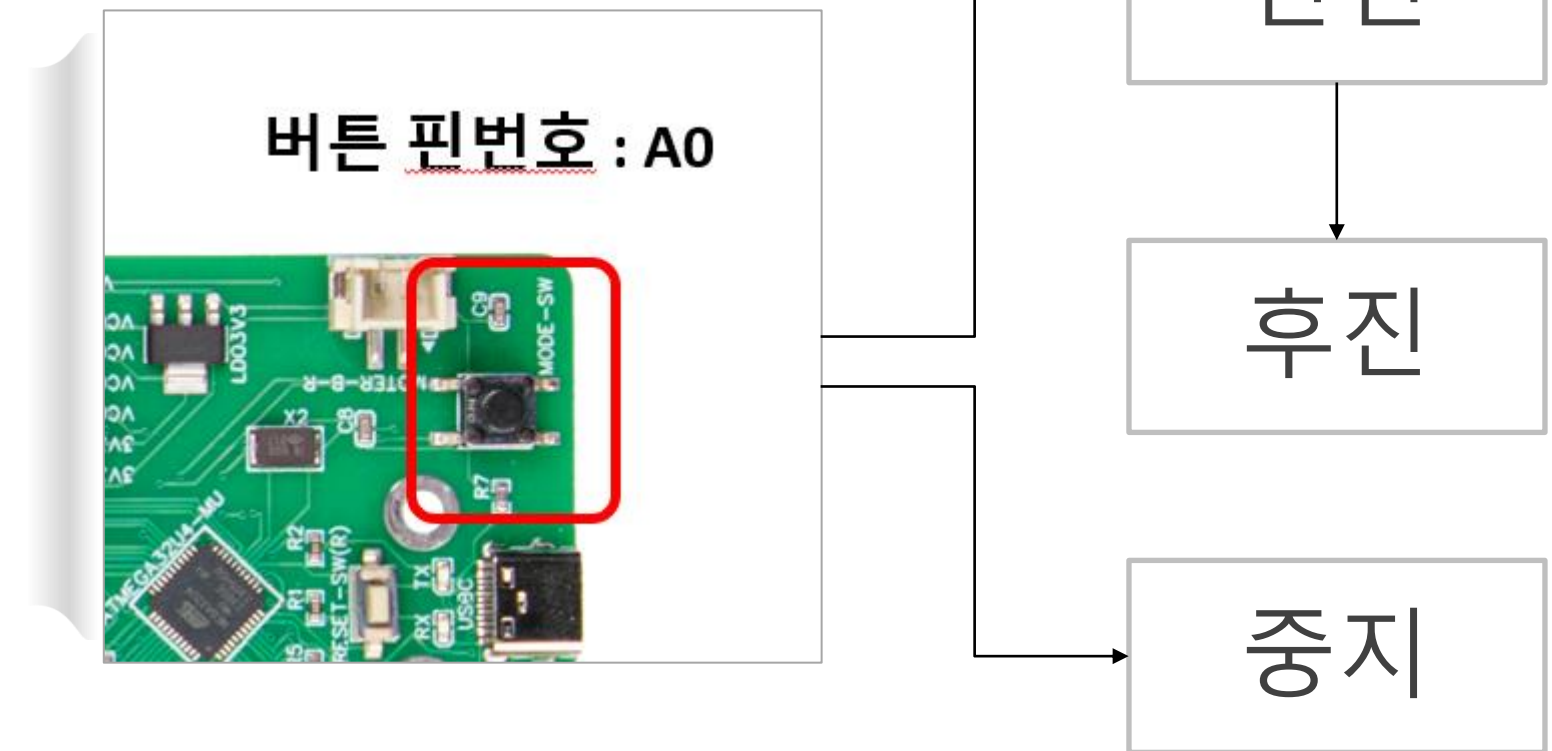
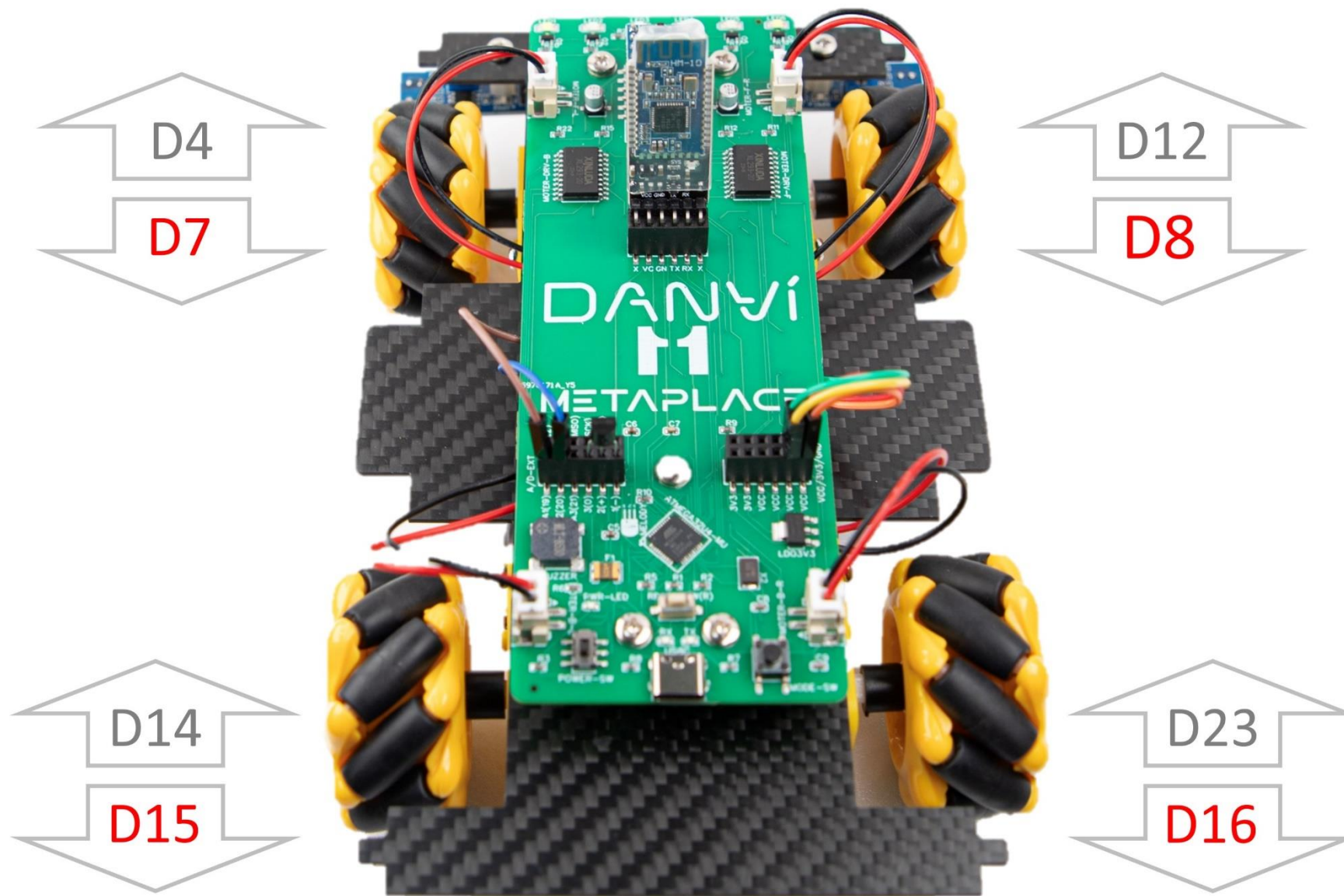
보조배터리 연결



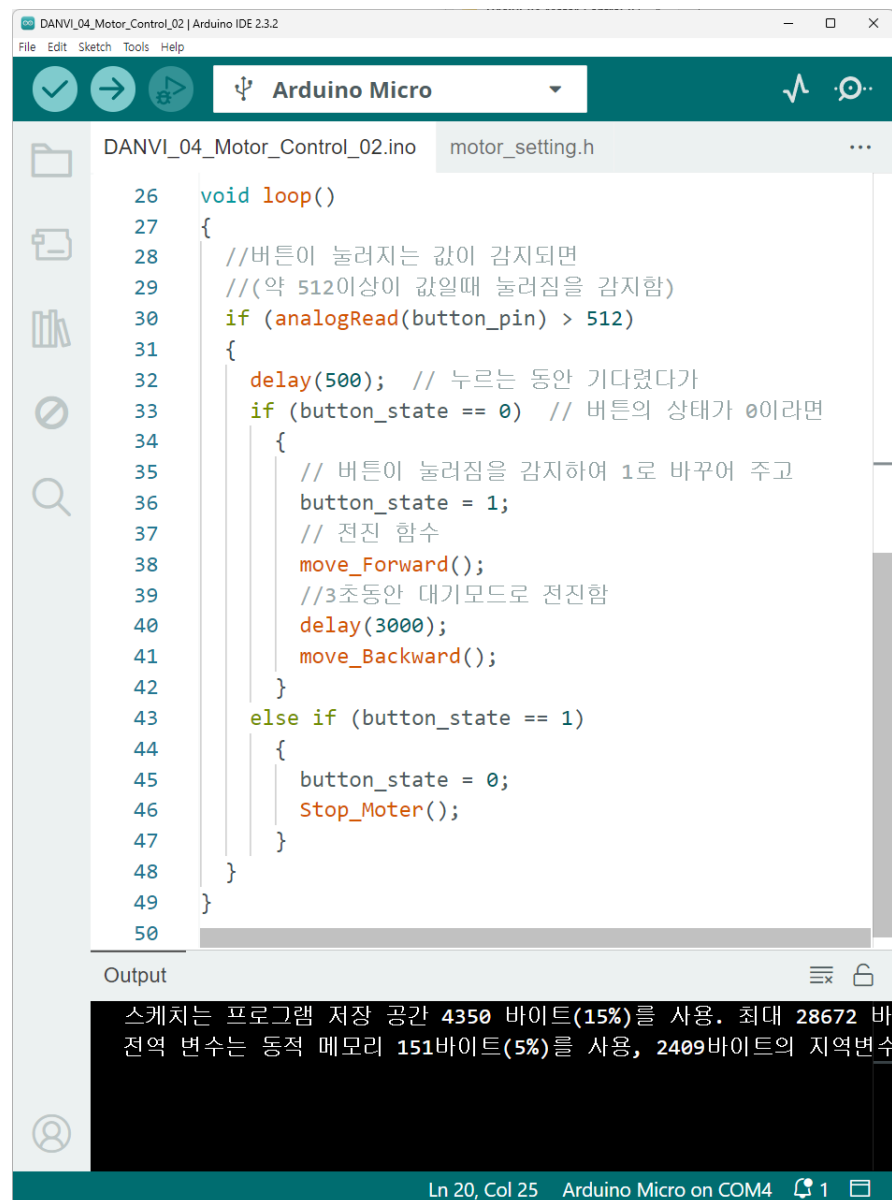
3



- 버튼을 누르면 전진, 다시누르면 중지의 움직임을 잘 해보셨나요?
- 그럼 뒤로 움직이는 코드를 **앞의 코드에서 추가**해볼까요!



- 단비 로봇이 전진하는 코딩에
- 아래의 코드부분을 **1줄만 추가해볼게요**. 로봇이 뒤로 움직일거예요!
- 중지하려면 버튼을 한번 더 누르세요!

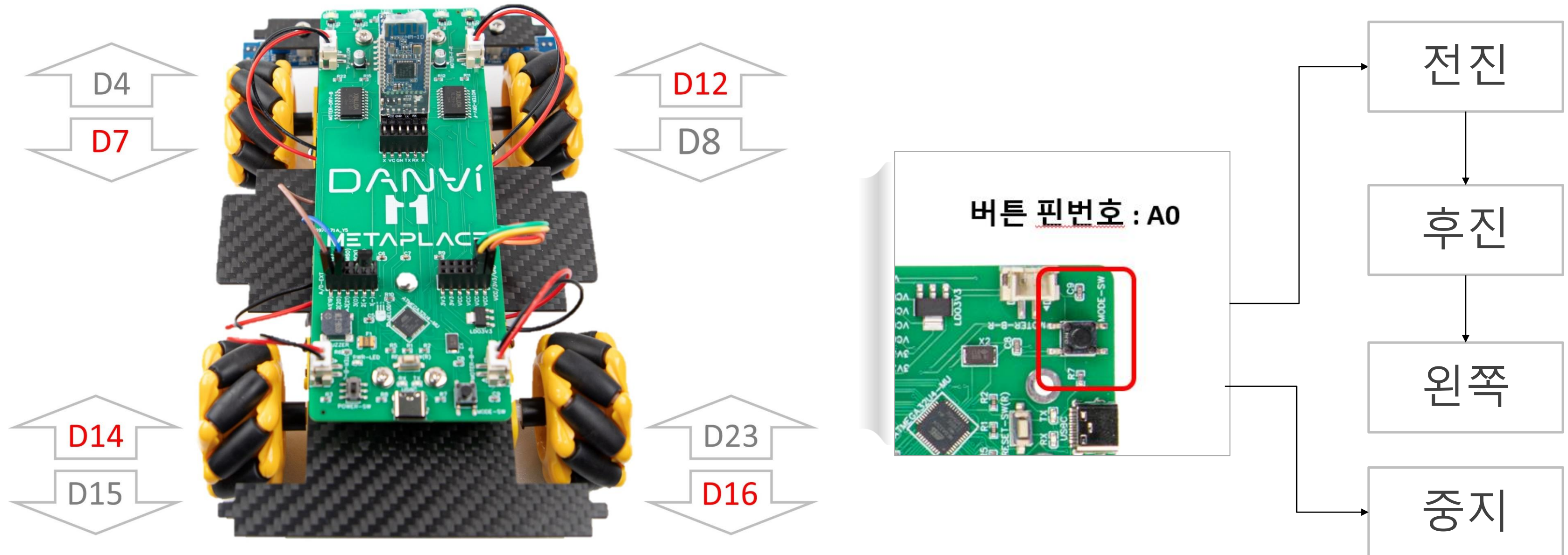


```
26 void loop()
27 {
28   //버튼이 눌러지는 값이 감지되면
29   //(약 512이상일 때 눌러짐을 감지함)
30   if (analogRead(button_pin) > 512)
31   {
32     delay(500); // 누르는 동안 기다렸다가
33     if (button_state == 0) // 버튼의 상태가 0이라면
34     {
35       // 버튼이 눌러짐을 감지하여 1로 바꾸어 주고
36       button_state = 1;
37       // 전진 함수
38       move_Forward();
39       //3초동안 대기모드로 전진함
40       delay(3000);
41       move_Backward();
42     }
43     else if (button_state == 1)
44     {
45       button_state = 0;
46       Stop_Moter();
47     }
48   }
49 }
50
```

```
if (analogRead(button_pin) > 512)
{
  delay(500); // 누르는 동안 기다렸다가
  if (button_state == 0) // 버튼의 상태가 0이라면
  {
    // 버튼이 눌러짐을 감지하여 1로 바꾸어 주고
    button_state = 1;
    // 전진 함수
    move_Forward();
    //3초동안 대기모드로 전진함
    delay(3000);
    move_Backward();
  }
}
```

1줄 추가

- 단비 로봇이 전진, 후진하는 코딩에
- 아래의 코드부분을 **2줄만 추가해볼게요**. 로봇이 전진-후진-왼쪽으로 움직여요!



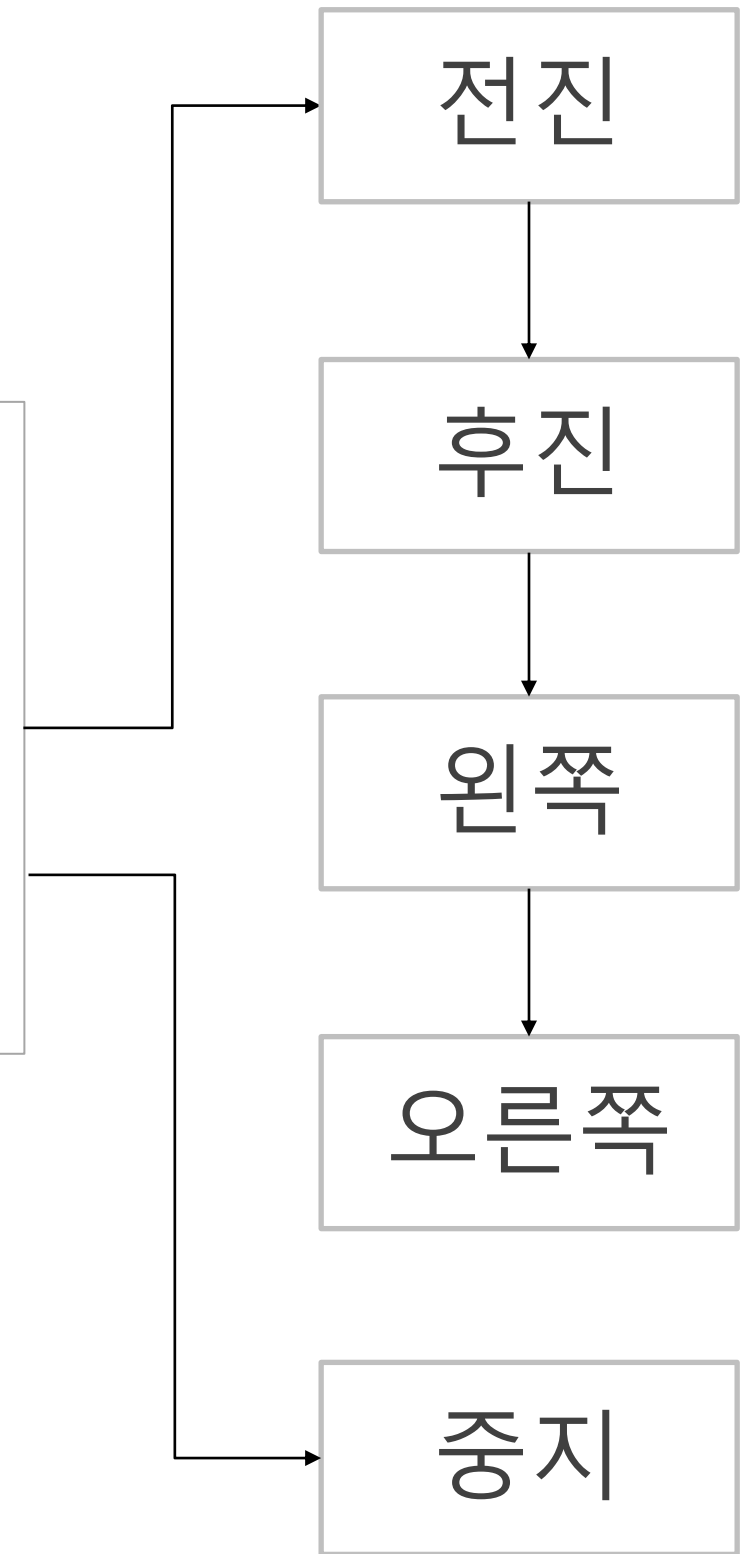
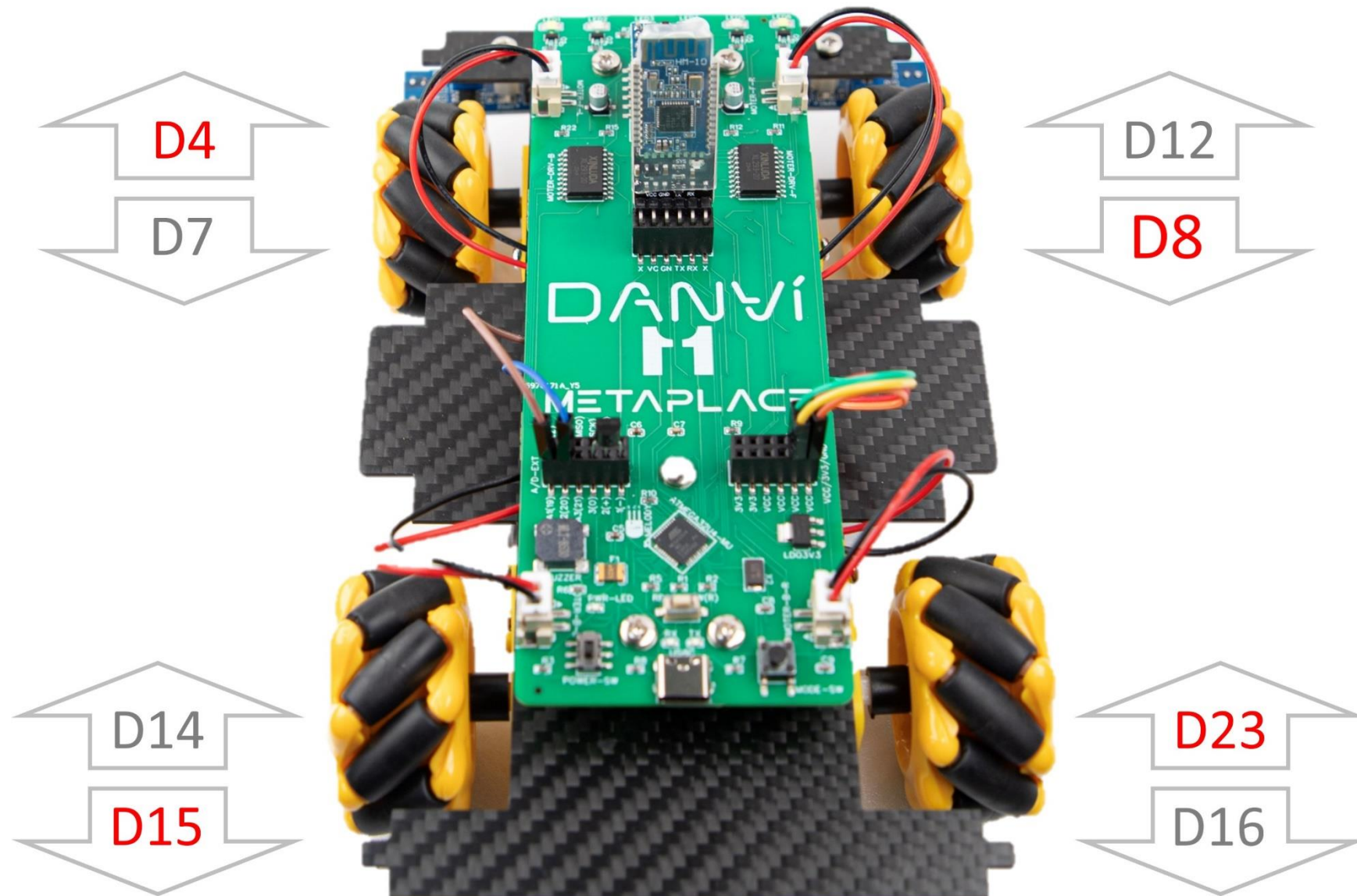
- 단비 로봇이 전진, 후진하는 코딩에
- 아래의 코드부분을 **2줄만 추가해볼게요**. 로봇이 전진-후진-왼쪽으로 움직여요!

```
26 void loop()
27 {
28   //버튼이 눌러지는 값이 감지되면
29   //(약 512이상 값일때 눌러짐을 감지함)
30   if (analogRead(button_pin) > 512)
31   {
32     delay(500); // 누르는 동안 기다렸다가
33     if (button_state == 0) // 버튼의 상태가 0이라면
34     {
35       // 버튼이 눌러짐을 감지하여 1로 바꾸어 주고
36       button_state = 1;
37       // 전진 함수
38       move_Forward();
39       //3초동안 대기모드로 전진함
40       delay(3000);
41       move_Backward();
42       delay(3000);
43       move_Leftward();
44     }
45     else if (button_state == 1)
46     {
47       button_state = 0;
48       Stop_Moter();
49     }
50 }
```

```
if (analogRead(button_pin) > 512)
{
  delay(500); // 누르는 동안 기다렸다가
  if (button_state == 0) // 버튼의 상태가 0이라면
  {
    // 버튼이 눌러짐을 감지하여 1로 바꾸어 주고
    button_state = 1;
    // 전진 함수
    move_Forward();
    //3초동안 대기모드로 전진함
    delay(3000);
    move_Backward();
    delay(3000);
    move_Leftward();
  }
}
```

2줄 추가

- 전진 - 후진 - 왼쪽 - 오른쪽으로 움직여 볼게요.



- 단비 로봇이 전진, 후진, 왼쪽으로 가는 코딩을 해봤어요.
- 아래의 코드부분을 **2줄만 추가해볼게요**. 로봇이 전진-후진-왼쪽-오른쪽으로 움직여요!

```
if (button_state == 0) // 버튼의 상태가 0이라면
{
    // 버튼이 눌러짐을 감지하여 1로 바꾸어 주고
    button_state = 1;
    // 전진 함수
    move_Forward();
    //3초동안 대기모드로 전진함
    delay(3000);
    move_Backward();
    delay(3000);
    move_Leftward();
    delay(3000);
    move_Rightward();
}
else if (button_state == 1)
{
    button_state = 0;
    Stop_Moter();
}
```

2줄 추가

** 미션!

다양한 헤더파일의 함수를 활용하여

-네모

-세모

-원돌리기

움직임 만들어보기



```
//전진 함수
move_Forward();
delay(8000);
```

```
//후진 함수
move_Backward();
delay(8000);
```

```
//왼쪽 이동함수
move_Leftward();
delay(8000);
```

```
//오른쪽 이동함수
move_Rightward();
delay(8000);
```

```
//왼쪽대각선전진 이동함수
move_LeftFordia();
delay(8000);
```

```
//오른쪽대각선전진 이동함수
move_RightFordia();
delay(8000);
```

```
//왼쪽대각선후진 이동함수
move_LeftBackdia();
delay(8000);
```

```
//오른쪽대각선후진 이동함수
move_RightBackdia();
```

```
//왼쪽라운딩 이동함수
move_LeftTurnRound();
delay(8000);
//오른쪽라운딩 이동함수
move_RightTurnRound();
delay(8000);
```

```
//왼쪽코너링전진 이동함수
move_LeftForCornering();
delay(8000);
```

```
//오른쪽코너링전진 이동함수
move_RightForCornering();
delay(8000);
```

```
//왼쪽코너링후진 이동함수
move_LeftBackCornering();
delay(8000);
```

```
//오른쪽코너링후진 이동함수
move_RightBackCornering();
delay(8000);
```

```
//왼쪽라운딩(뒤축기준) 이동함수
move_LeftTurnRoundRearAxis();
delay(8000);
```

```
//오른쪽라운딩(뒤축기준) 이동함수
move_RightTurnRoundRearAxis();
delay(8000);
```

```
//왼쪽라운딩(앞축기준) 이동함수
move_LeftTurnRoundFrontAxis();
delay(8000);
```

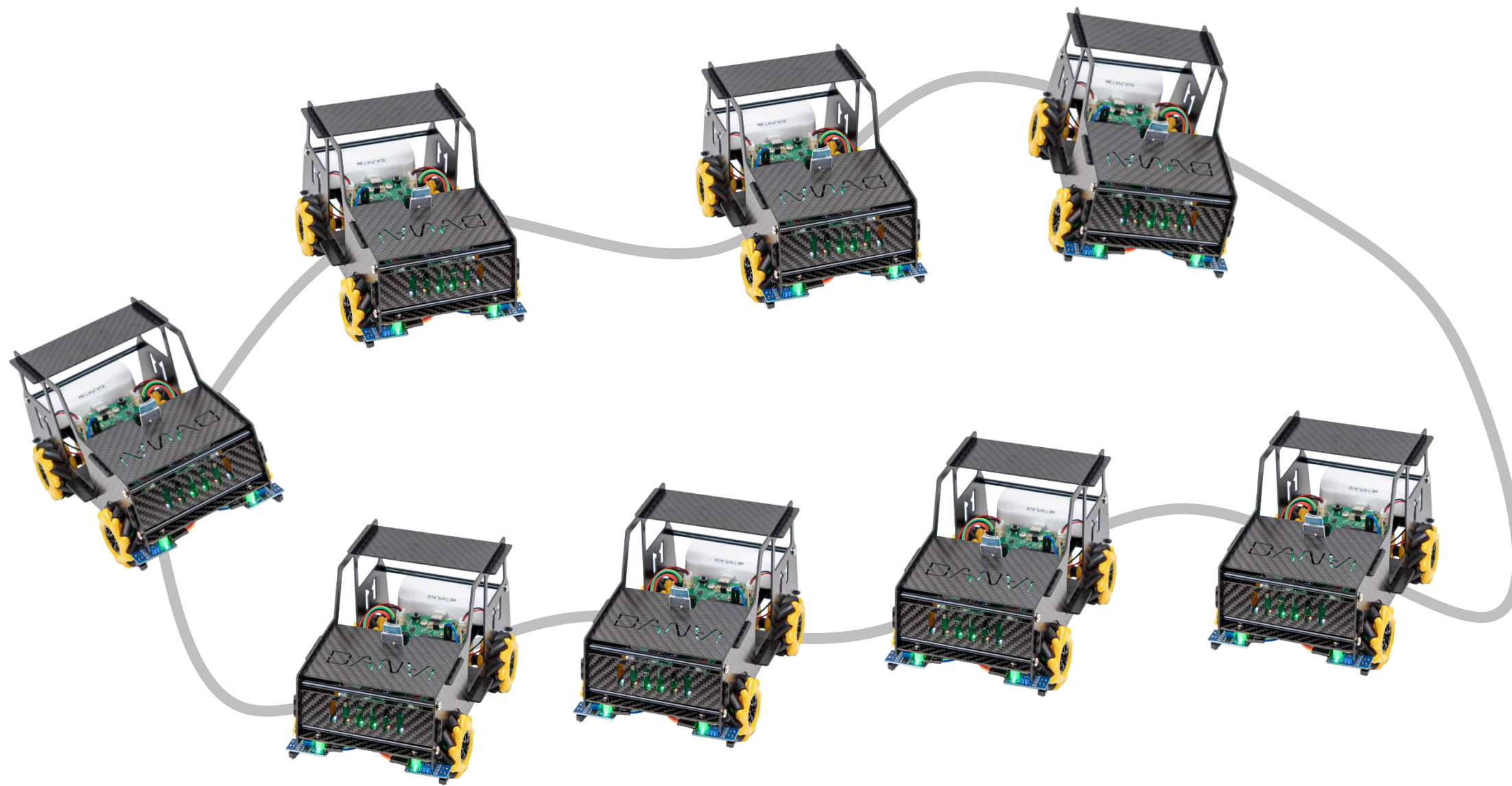
```
//오른쪽라운딩(앞축기준) 이동함수
move_RightTurnRoundFrondAxis();
```

파일명 : DANVI_04_Motor_Control_final.ino
motor_setting.h

미션 : 단비로 자율주행 해봐요!

오른쪽으로 주행하는 코드를 추가해요!

- DANVI_04_Motor_Control_final파일을 불러와서 **실행해보세요!**
- **친구와 함께 동시에 실행해 보세요!** 어떻게 움직이는지 살펴보세요!



미션 : 단비로 자율주행 해봐요! - (친구와 함께 해보기) 자유롭게 코딩해봐요!

- ④ [해보기] 다양한 헤더파일의 함수를 활용하여 다음의 움직임을 만들어봐요!

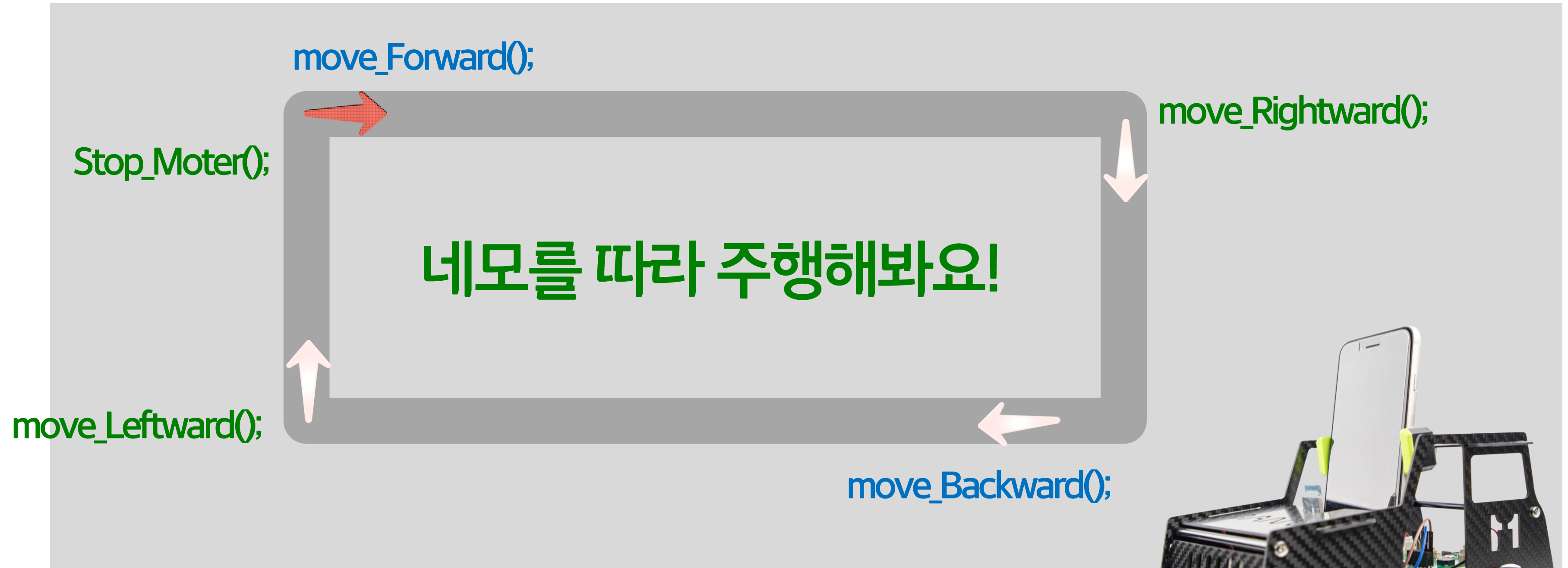
현재 자리에서 왼쪽으로 원을 따라 주행해봐요!



파일명 : DANVI_04_Motor_Control_Mission_Circle.ino
motor_setting.h

미션 : 단비로 자율주행 해봐요! - (친구와 함께 해보기) 자유롭게 코딩해봐요!

🌀 [해보기] 다음의 움직임을 만들어봐요!



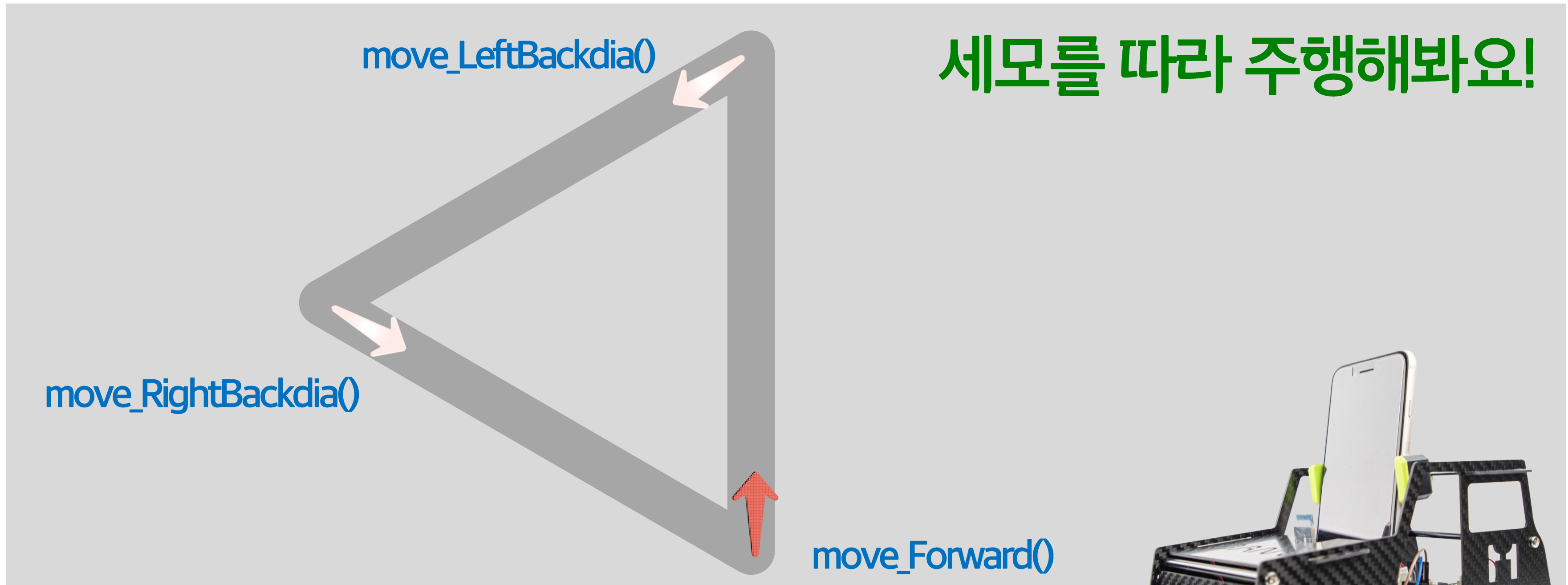
파일명 : DANVI_04_Motor_Control_Mission_Square.ino
motor_setting.h



미션 : 단비로 자율주행 해봐요! - (친구와 함께 해보기) 자유롭게 코딩해봐요!

- ④ [해보기] 다양한 헤더파일의 함수를 활용하여 다음의 움직임을 만들어봐요!

세모를 따라 주행해봐요!



파일명 : DANVI_04_Motor_Control_Mission_Triangle.ino
motor_setting.h



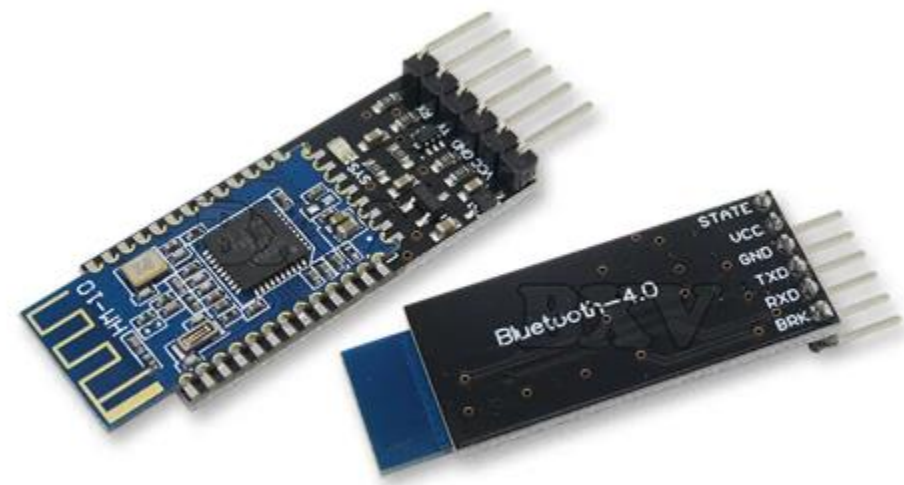
HM-10 블루투스의 이름 바꾸기

: 여러 블루투스가 있으면 나만의 이름으로 알 수 있는 블루투스로 이름을 변경해야 합니다.



블루투스에 이름이 붙어있으면 이 과정은 건너뛰기!





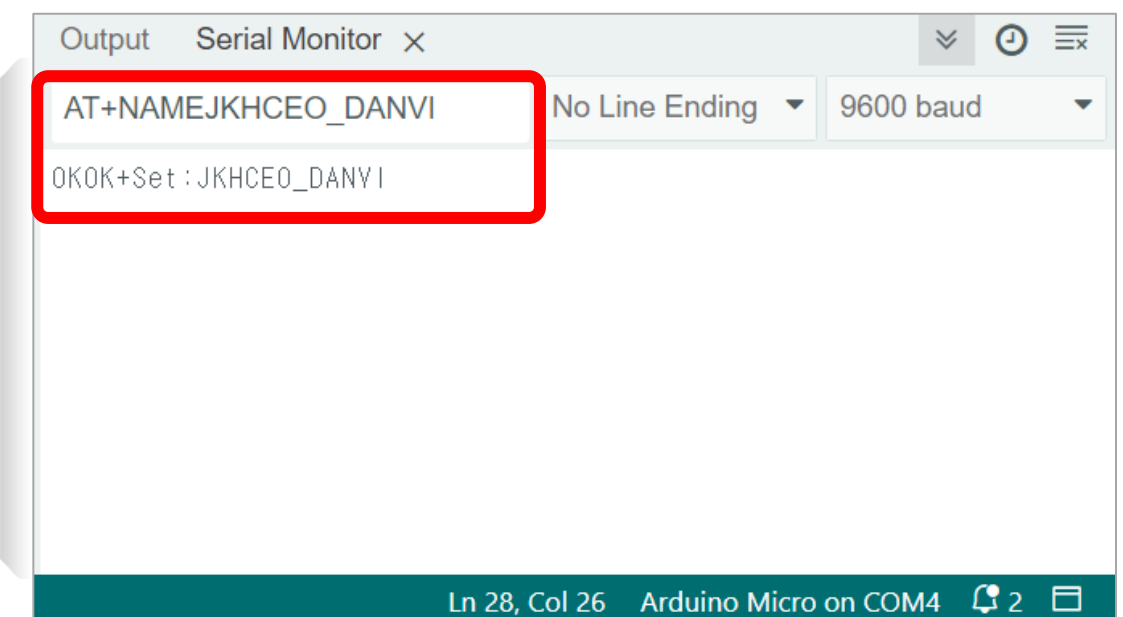
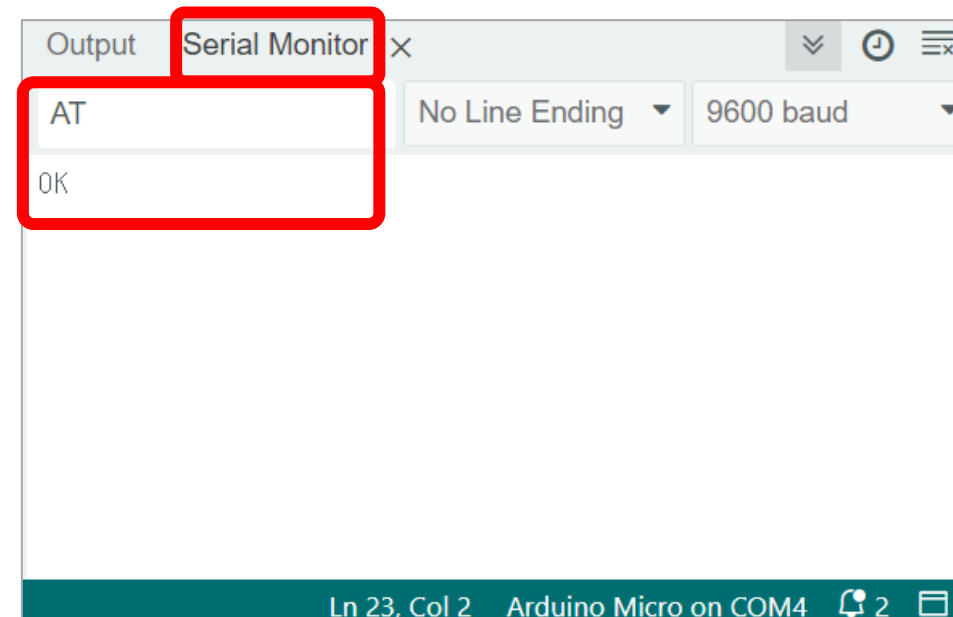
- **HM-10**은 Arduino 및 기타 마이크로컨트롤러와 함께 사용되는 저전력 블루투스 모듈
- Texas Instruments사의 CC2540 블루투스 칩을 기반으로 하며, **시리얼 통신을 통해 간단하게 제어**
- HM-10은 주로 블루투스 4.0 Low Energy (BLE) 기능을 활용하여 블루투스 장치 간의 통신에 사용
- HM-10 모듈은 저전력 특성을 가지고 있어서 배터리 구동 장치에 이상적임.
- **주로 IoT (사물인터넷), 웨어러블 기기, 센서 네트워크 등의 분야에서 사용**
- 예를 들어, 스마트 홈 시스템, 원격 제어 장치, 위치 기반 서비스, 센서 데이터 수집 및 제어 등에 사용함
- **HM-10 블루투스는 [안드로이드와 아이폰] 모두 OK!!!**
- HC-06은 안드로이드에서만 작동되고, 아이폰이 안되요!


```

DANVI_06_Bluetooth_Naming | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_06_Bluetooth_Naming.ino
1 void setup()
2 {
3   //시리얼 모니터 연결
4   Serial.begin(9600);
5   //블루투스시리얼[하드웨어] 초기화(9600)bps
6   Serial1.begin(9600);
7 }
8
9 void loop()
10 {
11 // 블루투스시리얼[하드웨어] 명령 수신
12 if (Serial1.available())
13 // Serial1[하드웨어] 값이 있으면
14 {
15   Serial.write(Serial1.read());
16   //블루투스측 내용을 시리얼모니터에 출력
17 }
18 // 시리얼통신 명령 수신
19 if (Serial.available())
20 {
21   Serial1.write(Serial.read());
22   // 시리얼모니터 내용을 블루투스측에 출력
23 }
24 }
25
26 /* 블루투스 이름 변경방법
27 "No line ending [9600 baud]"
28 AT (OK)
29 AT+NAME이름 (OK+Set:이름)
30 */

```

- 아두이노 Micro 보드에서의 블루투스 명령어 쓰는 방법입니다.
- *Serial1.begin(9600);*
 - 시리얼 통신 속도를 9600으로 설정
- *Serial1.read();*
 - 시리얼 포트에서 데이터 읽음
- *Serial1.write();*
 - 시리얼 포트에 데이터를 씀



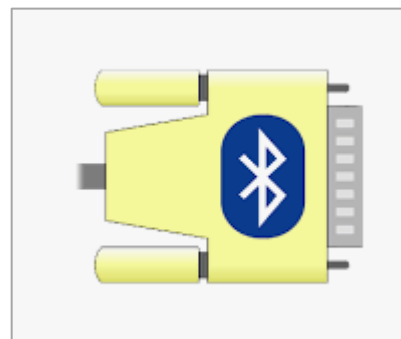
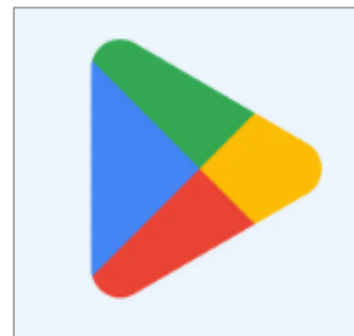
IoT 단비 로봇 조종하기



- 아이폰과 안드로이드폰 어플이 다르므로,
- 본인의 스마트폰에 맞는 어플을 다운로드 받아서 설치하시고 어플 세팅을 따라하세요.



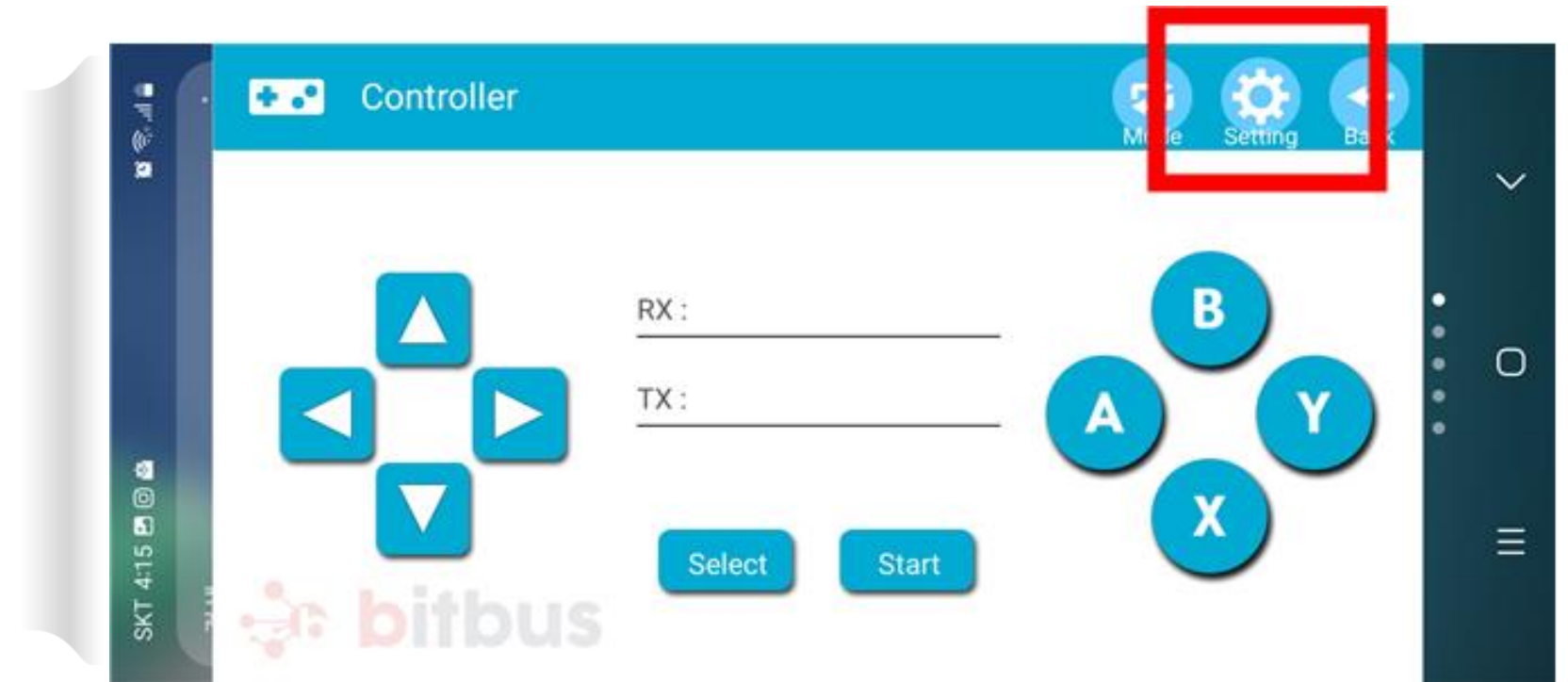
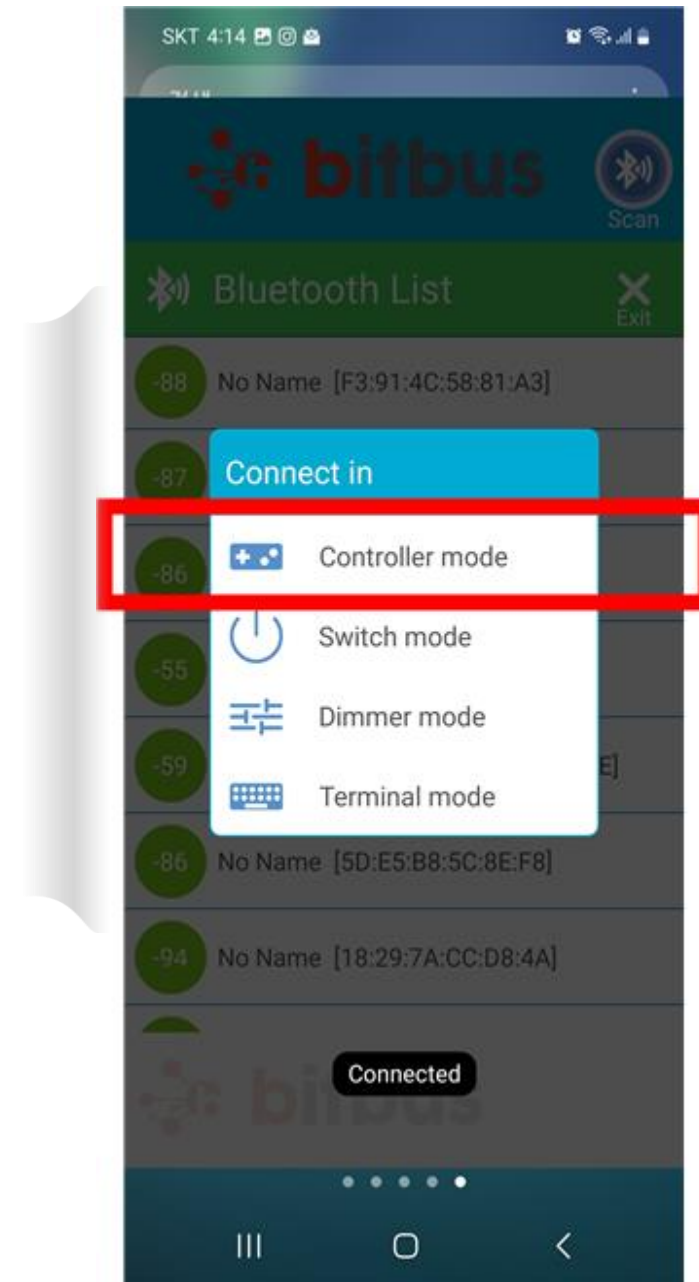
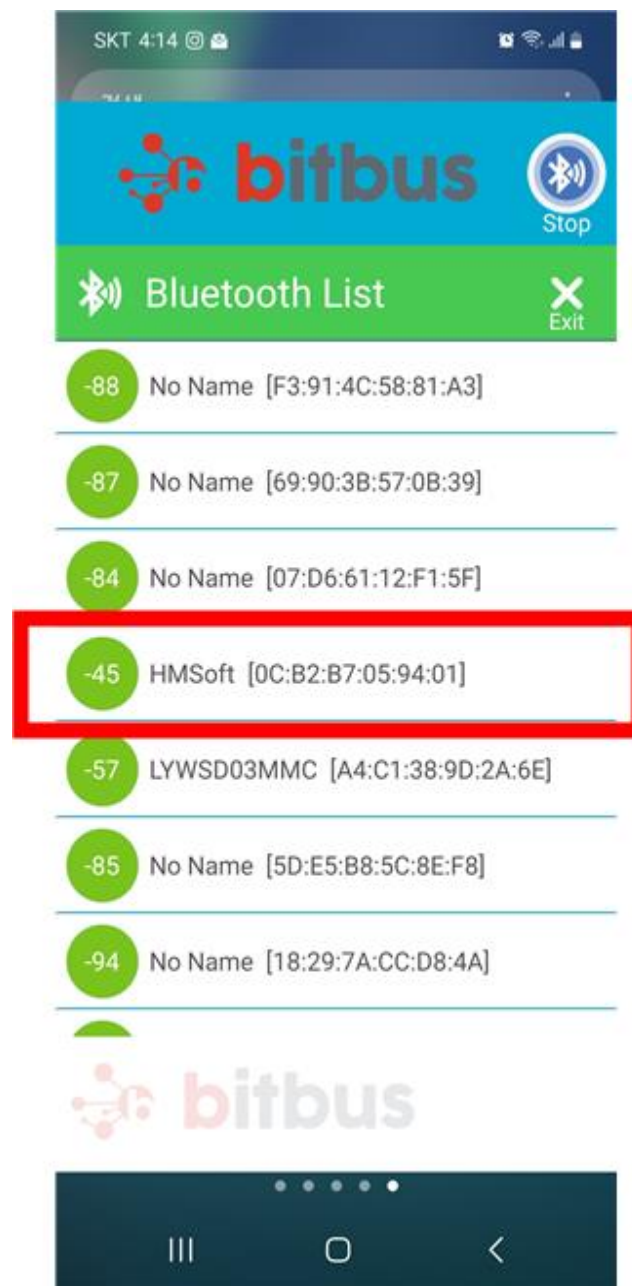
- 아이폰
: bitblue 로 검색



- 안드로이드폰
: serial bluetooth terminal












- 주의할점 : AI 단비의 전원이 켜져있어야 하며, HM10 블루투스기가 연결되어 있어야 해당하는 블루투스 이름이 나타난다. (전원켜기)

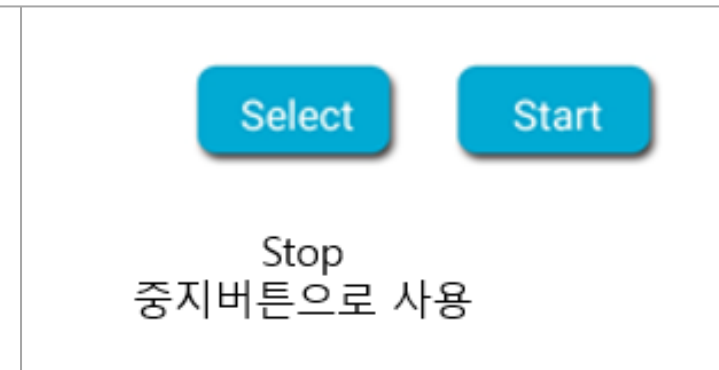
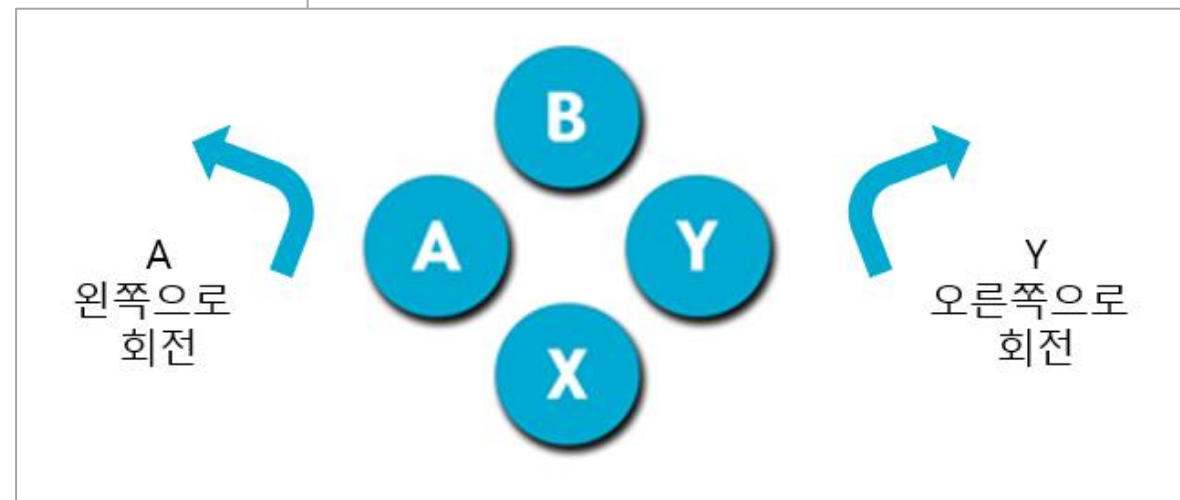
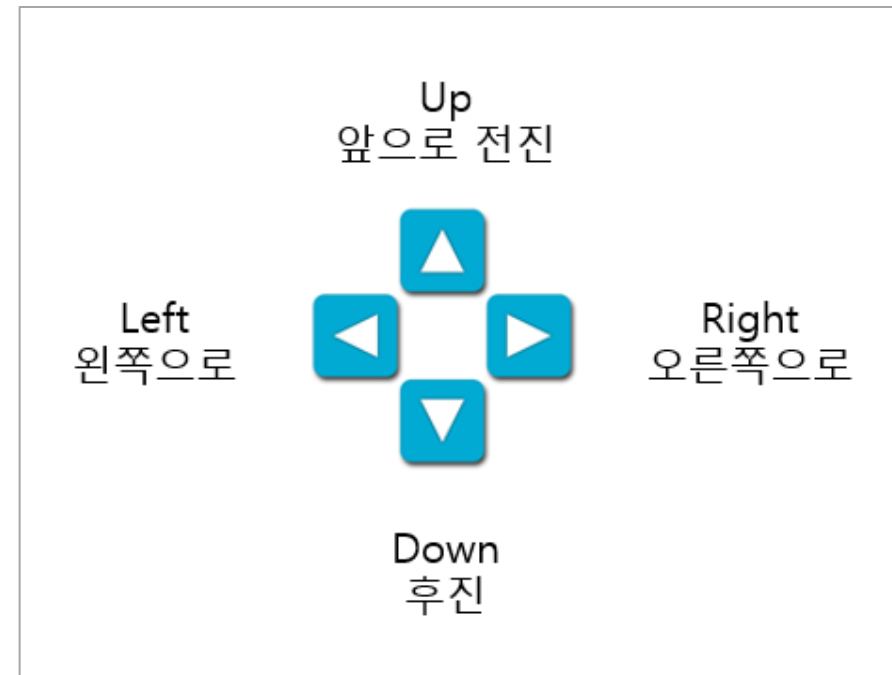




- 주의할점 : AI 단비의 전원이 켜져있어야 하며, HM10 블루투스가 연결되어 있어야 해당하는 블루투스 이름이 나타남 (전원켜기)



 1	 2	 3
 4	 5	 6
 7	 8	 9



Setting

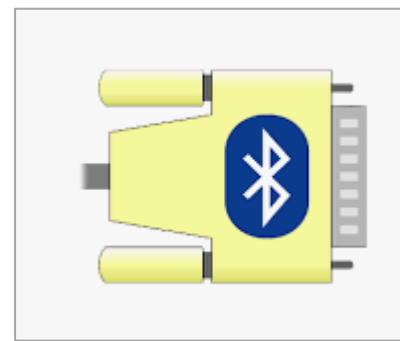
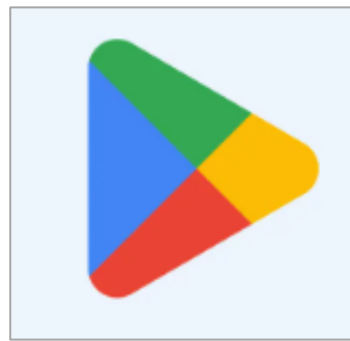
Reset Save Back

Setup the joystick buttons with your own commands to send

Left	4
Up	2
Right	6
Down	8
Select	0
Start	5
A	1
B	3
Y	9
X	7

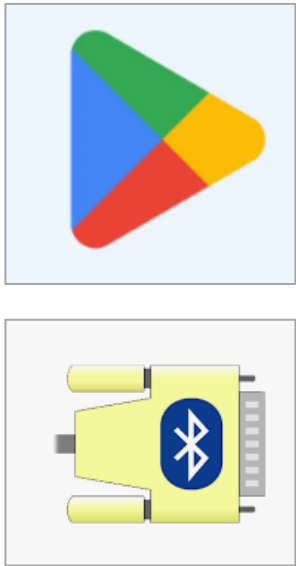
🌀 스마트폰에 어플을 설치 해봐요!

- 아이폰과 안드로이드폰 어플이 다르므로,
본인의 스마트폰에 맞는 어플을 다운로드 받아서 설치하시고 어플세팅을 따라하세요.

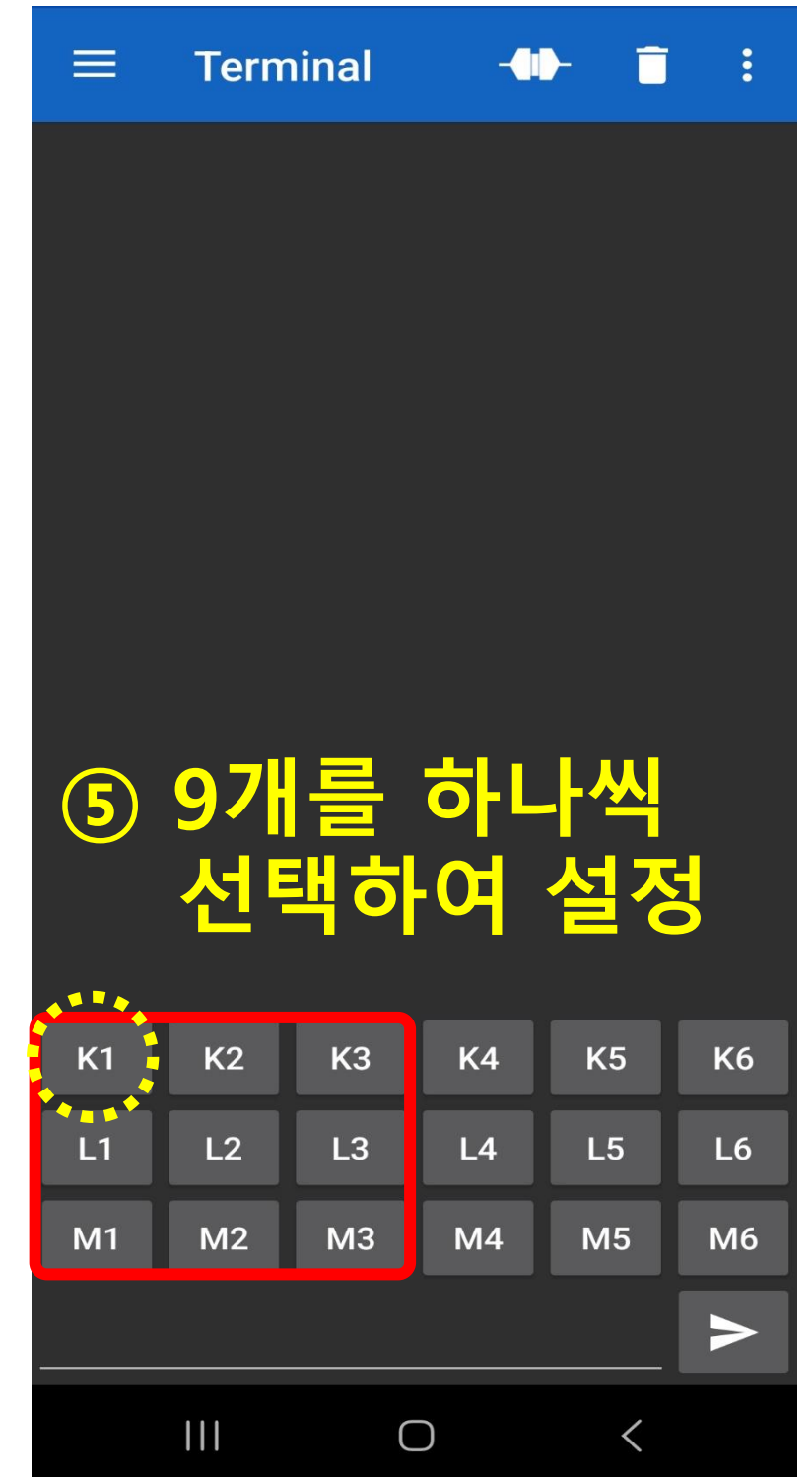
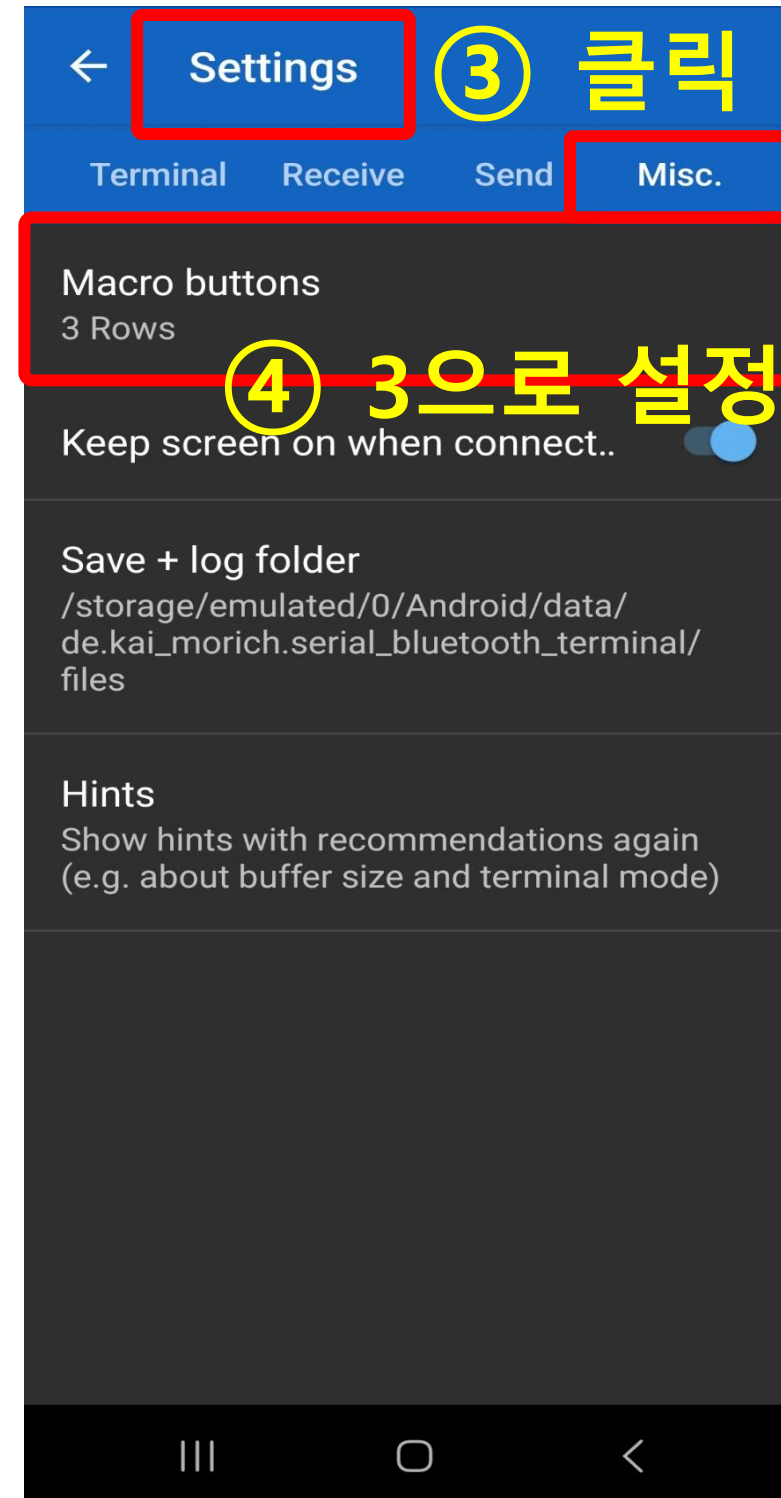
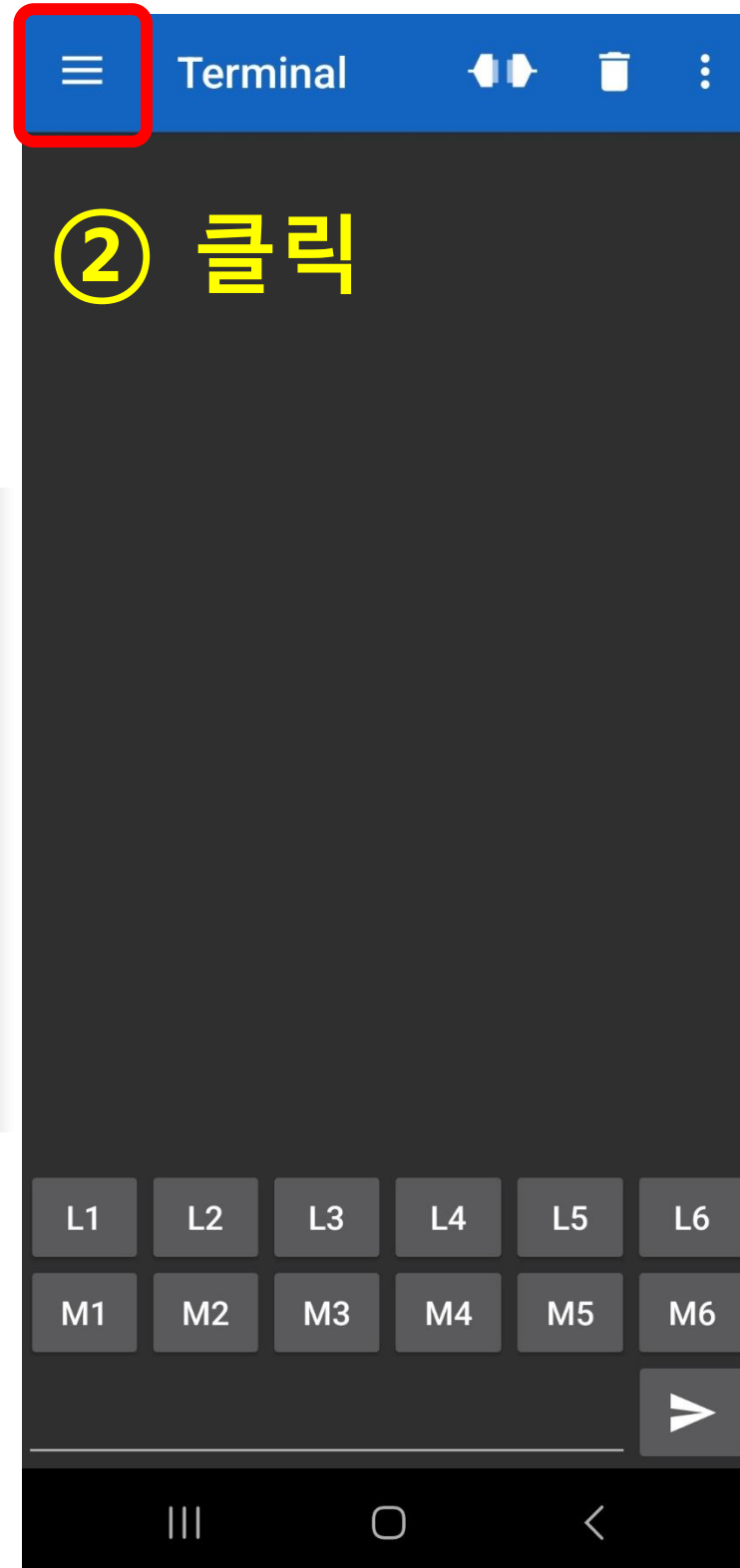
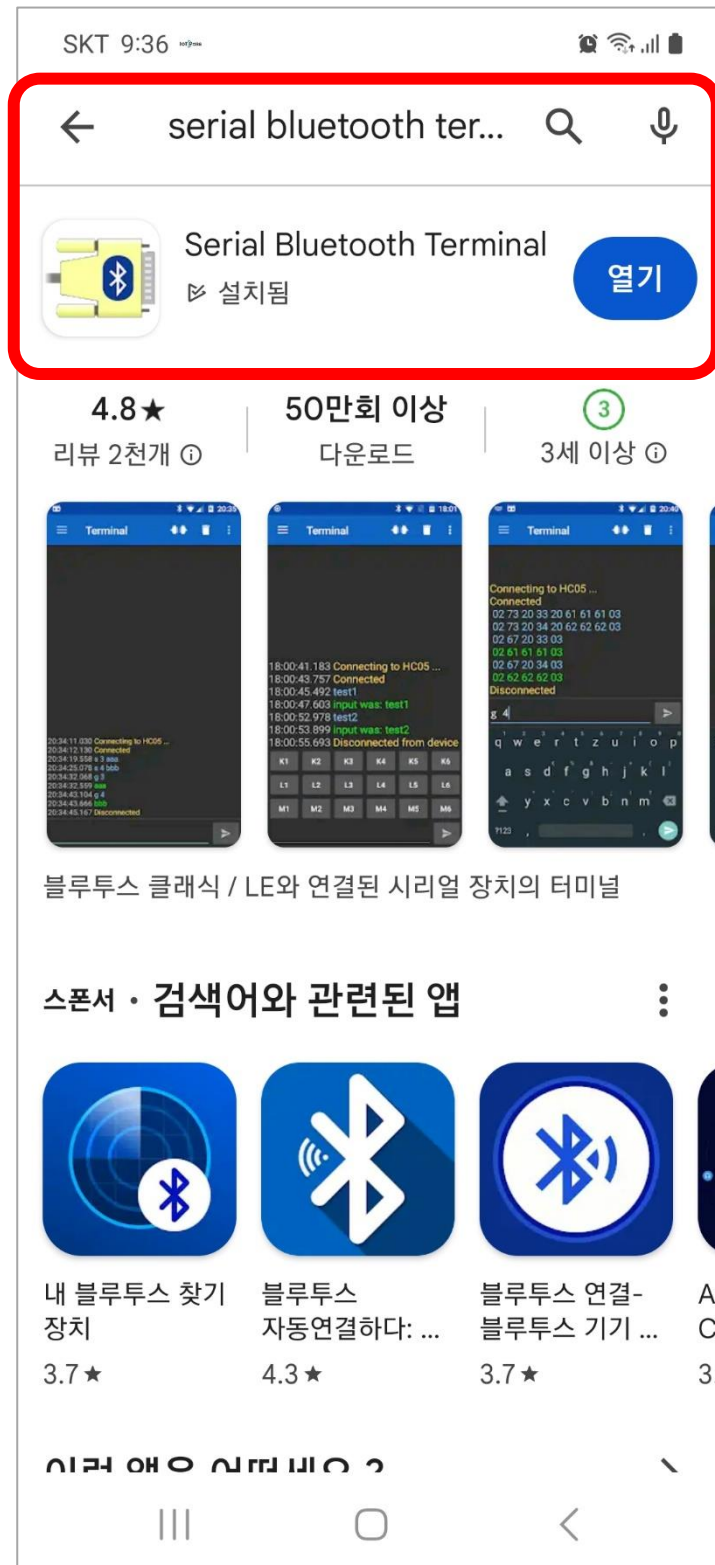


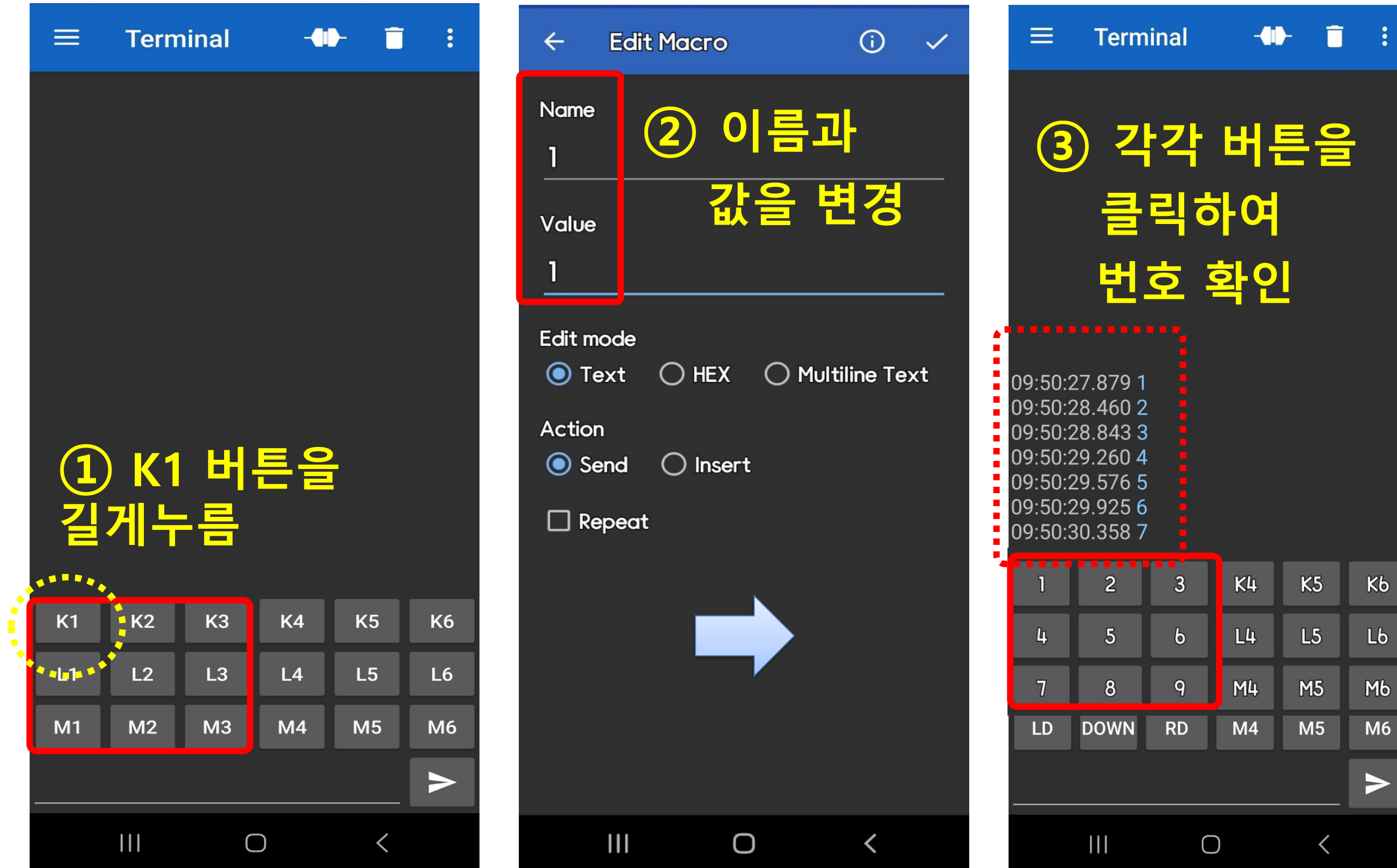
- 안드로이드폰

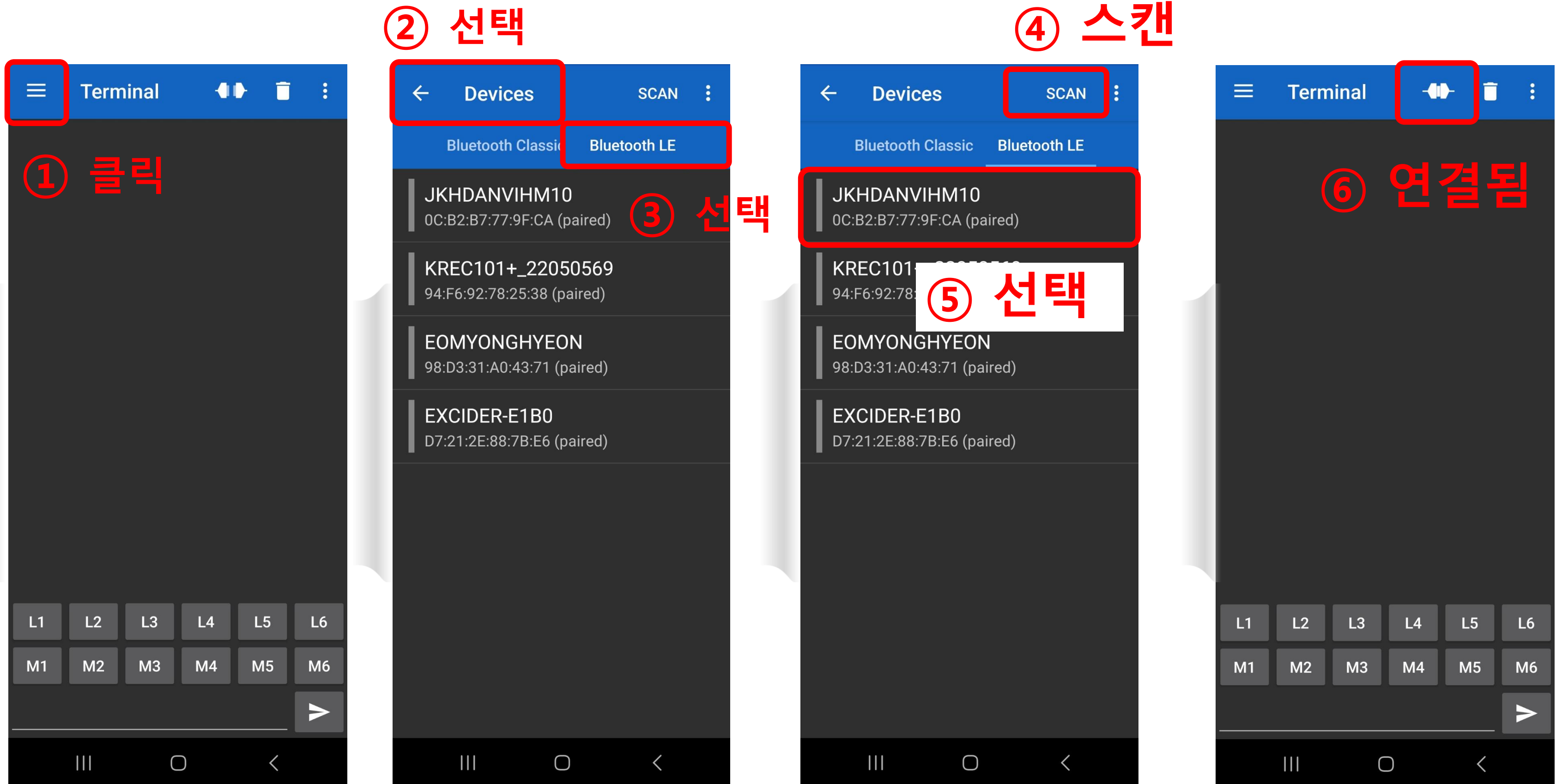
: serial bluetooth terminal

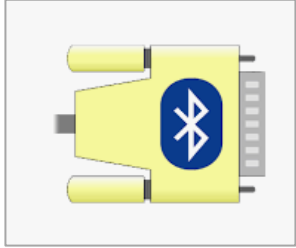



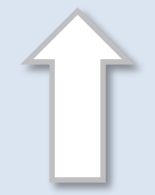

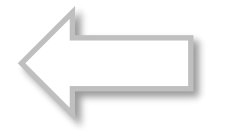

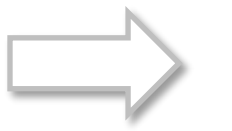

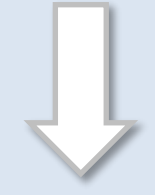

① 검색 후 열기

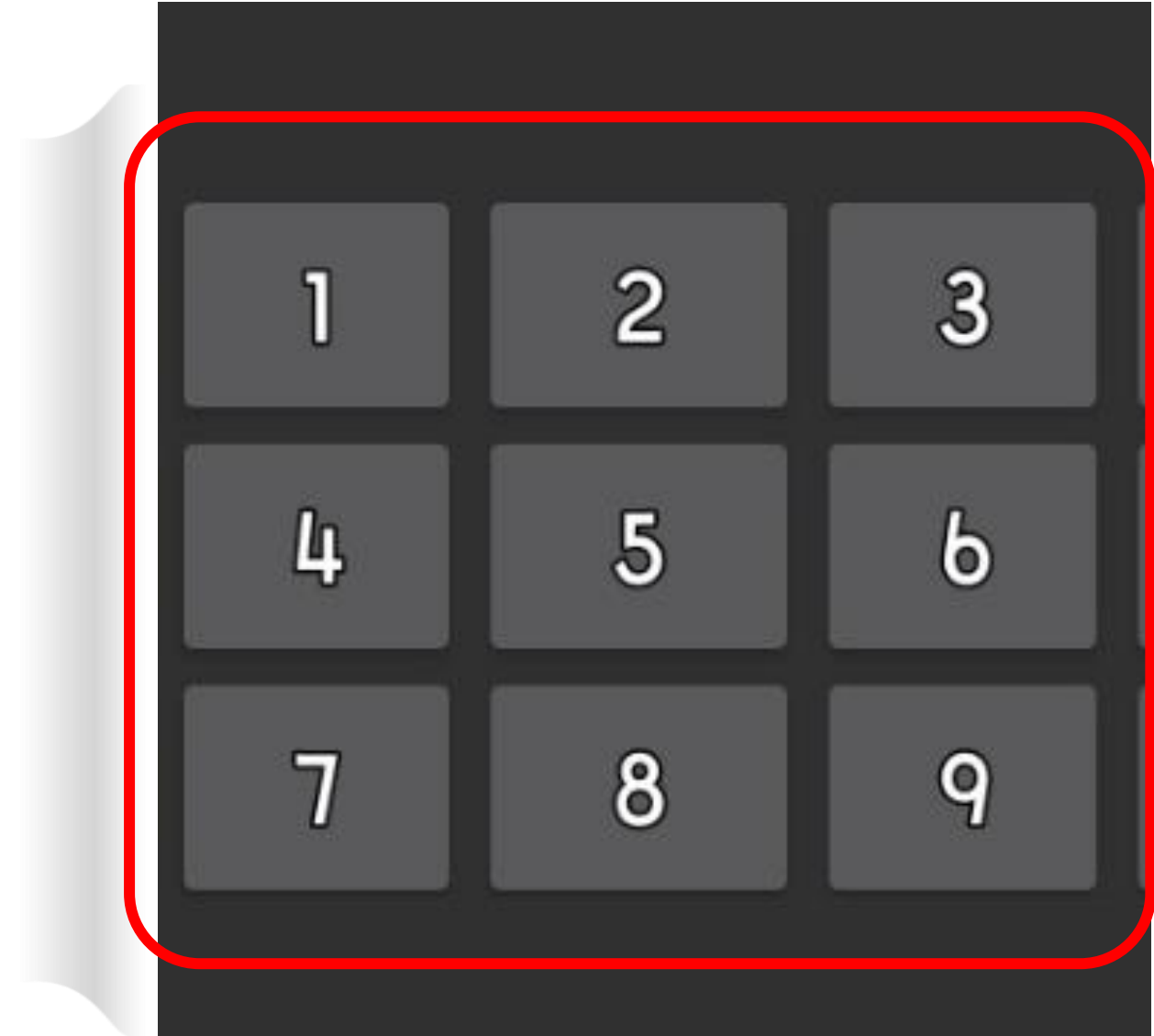






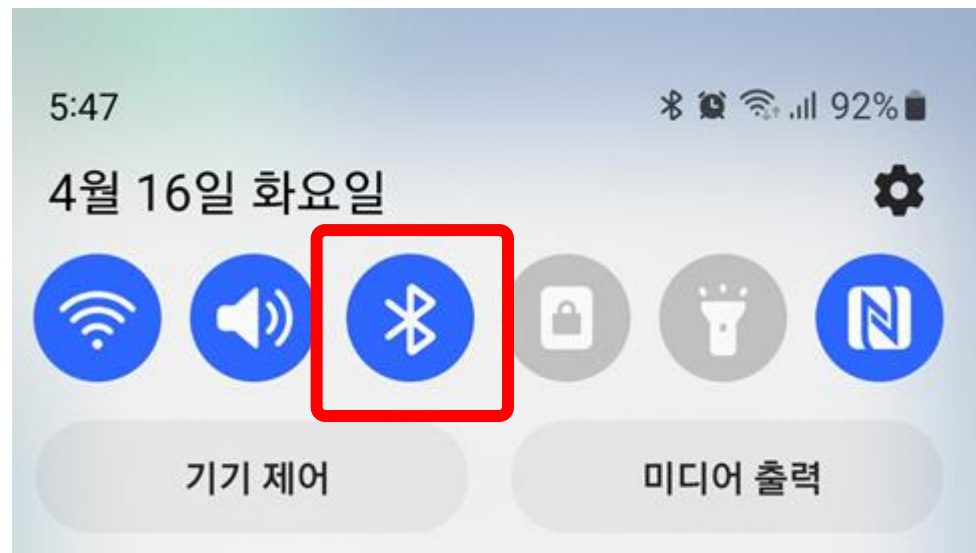


 1	 2	 3
 4	 5	 6
 7	 8	 9

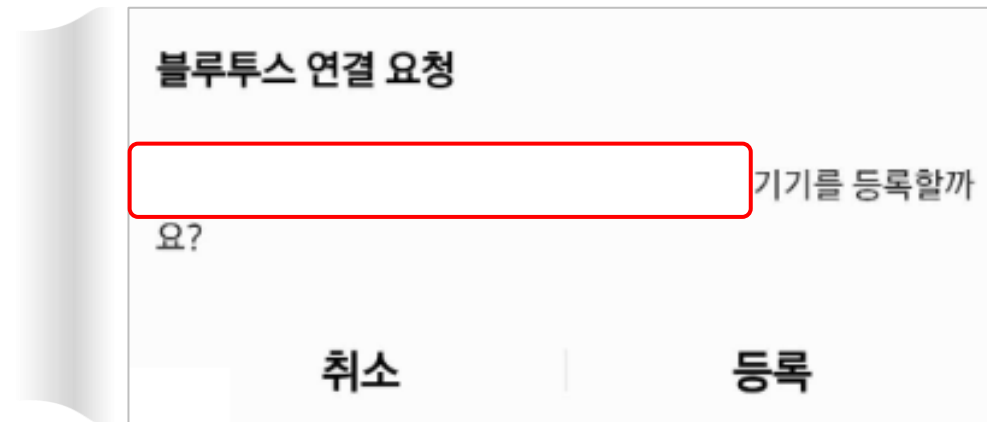
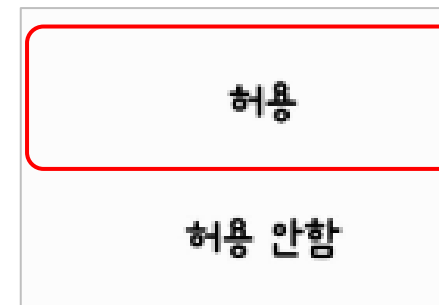


잠깐!! 블루투스가 연결허용했나요?

확인 1. 블루투스가 켜야합니다.

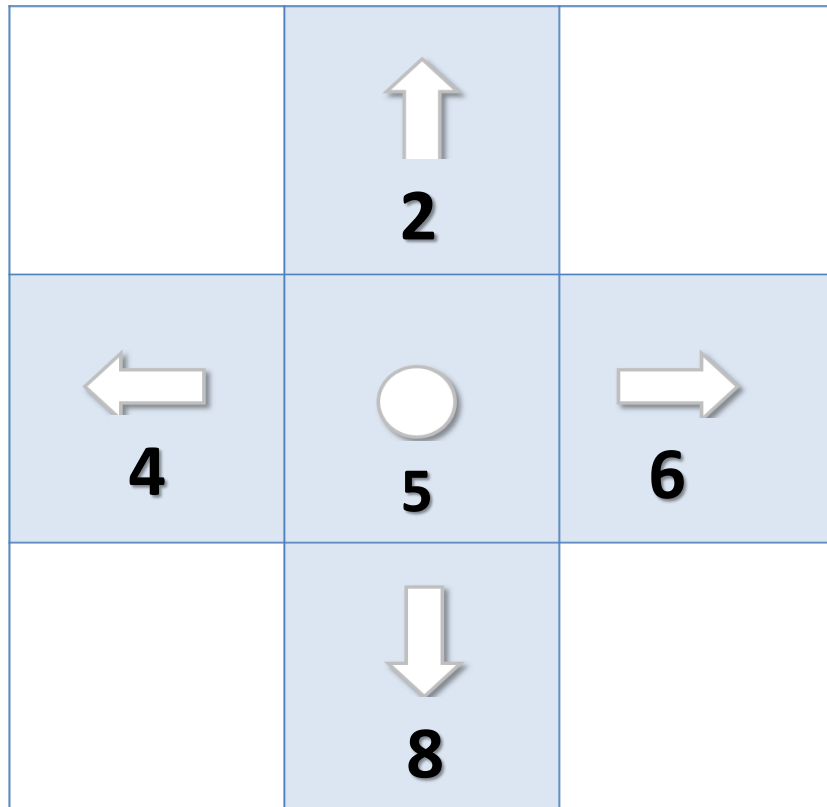


확인 2. 앱실행시 블루투스 허용하고,
연결요청시 등록해야 합니다.



블루투스 이름확인!
비밀번호는 **000000 (6개)**

5개의 버튼만 코딩하기



```

DANVI_06_Bluetooth_Motor_Control.ino  motor_setting.h  ...
1  #include "motor_setting.h"
2
3  // 13번 디지털핀(LED) 선언
4  #define led_pin 13
5
6  // 블루투스 받는 문자열
7  String bluetooth_string = "";
8
9  void setup()
10 {
11  // 시리얼통신 초기화(9600)bps
12  Serial.begin(9600);
13  //블루투스시리얼[하드웨어] 초기화(9600)bps
14  Serial1.begin(9600);
15  // 디지털핀 초기화(OUTPUT)
16  pinMode(led_pin, OUTPUT);
17  motor_output();
18  digitalWrite(led_pin, HIGH);
19  }
    
```

Output

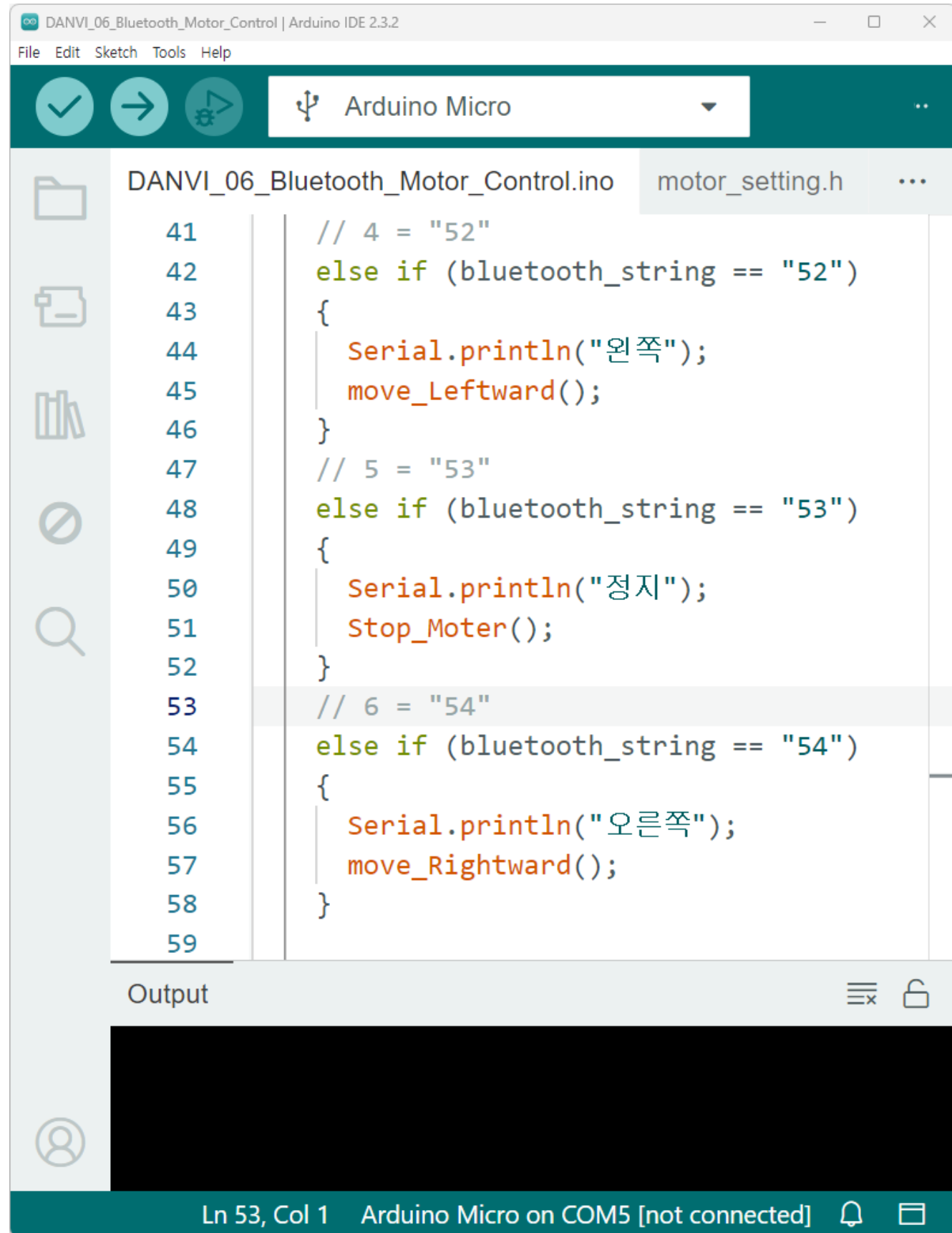
Ln 8, Col 1 Arduino Micro on COM5 [not connected]

```

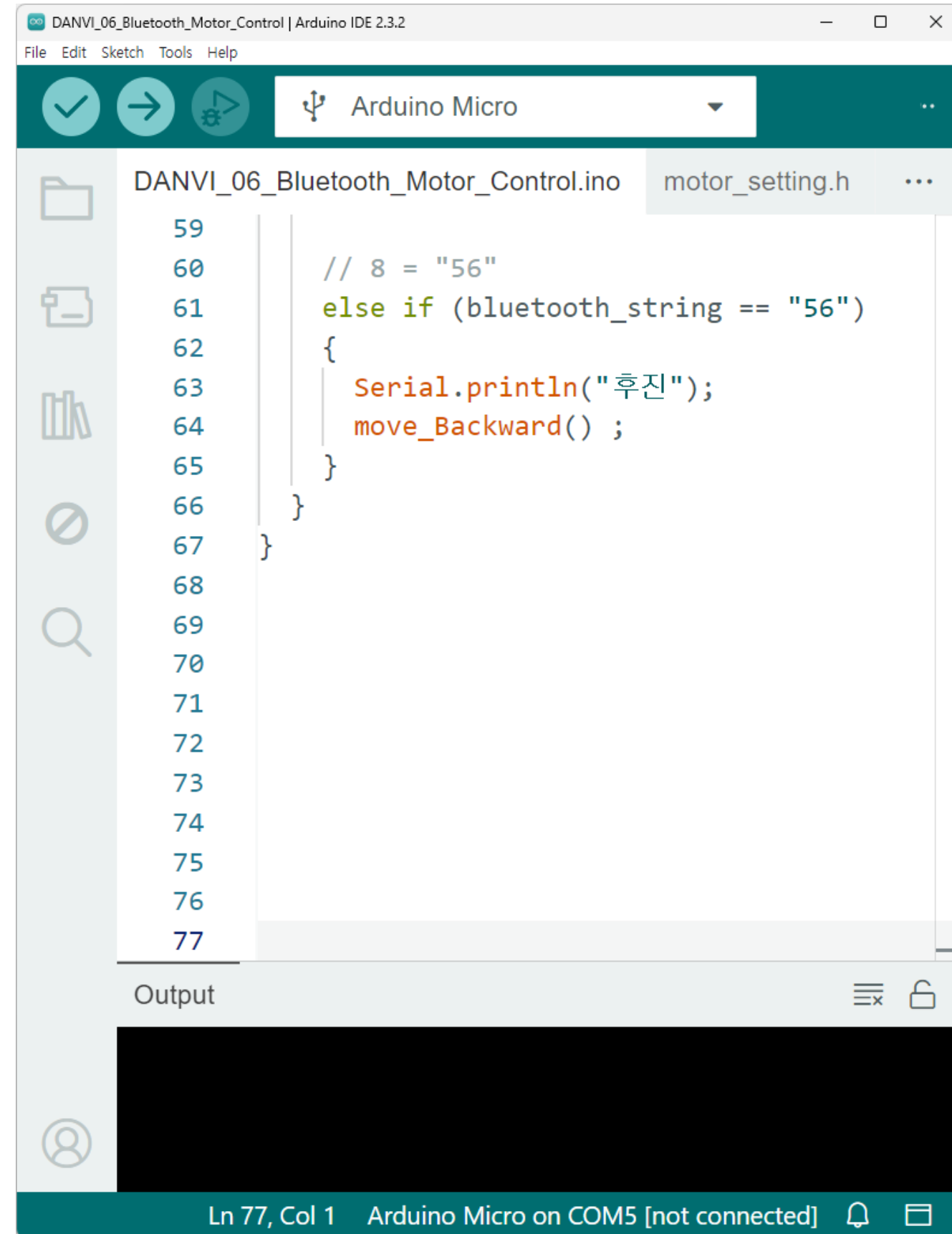
DANVI_06_Bluetooth_Motor_Control.ino  motor_setting.h  ...
21  void loop()
22  {
23  // 블루투스시리얼[하드웨어] 명령 수신
24  if (Serial1.available())
25  { // Serial1[하드웨어] 값이 있으면
26  bluetooth_string = Serial1.read();
27  Serial.println(bluetooth_string);
28  }
29  //시리얼모니터에 bluetooth_string 값 출력
30  //bluetooth_string 값이 있다면
31  if (bluetooth_string != "")
32  {
33  // 실행(1~9)(전화버튼 기준):: ASCII
34  // 2 = "50"
35  if (bluetooth_string == "50")
36  {
37  Serial.println("전진");
38  move_Forward();
39  }
40  }
    
```

Output

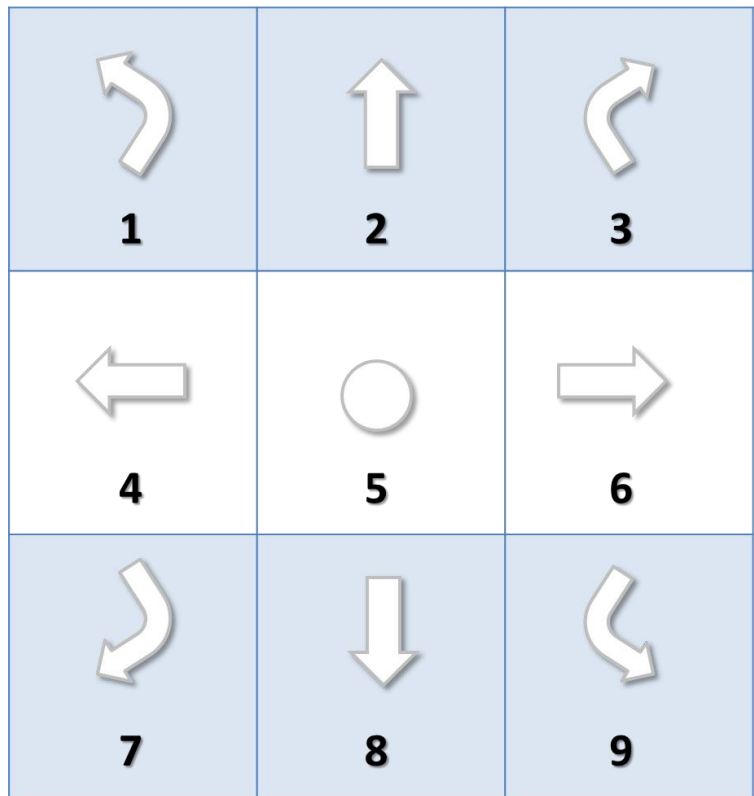
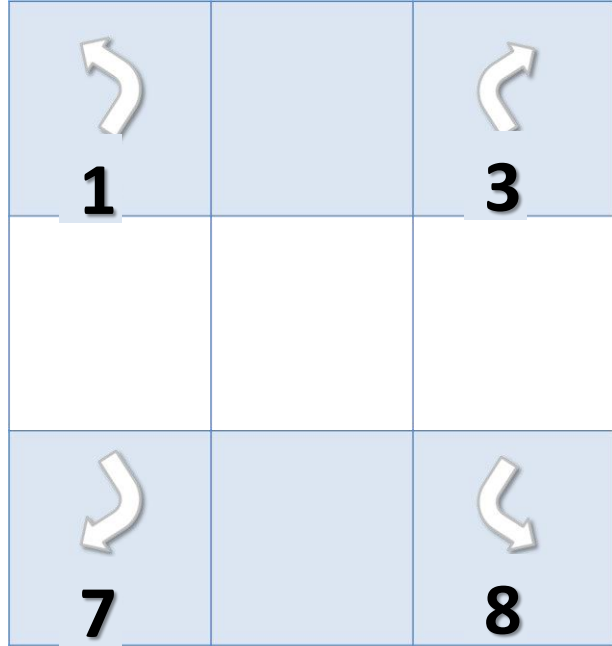
Ln 70, Col 1 Arduino Micro on COM5 [not connected]



```
DANVI_06_Bluetooth_Motor_Control | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_06_Bluetooth_Motor_Control.ino motor_setting.h ...
41 // 4 = "52"
42 else if (bluetooth_string == "52")
43 {
44   Serial.println("왼쪽");
45   move_Leftward();
46 }
47 // 5 = "53"
48 else if (bluetooth_string == "53")
49 {
50   Serial.println("정지");
51   Stop_Moter();
52 }
53 // 6 = "54"
54 else if (bluetooth_string == "54")
55 {
56   Serial.println("오른쪽");
57   move_Rightward();
58 }
59
Output
Ln 53, Col 1 Arduino Micro on COM5 [not connected]
```



```
DANVI_06_Bluetooth_Motor_Control | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_06_Bluetooth_Motor_Control.ino motor_setting.h ...
59
60 // 8 = "56"
61 else if (bluetooth_string == "56")
62 {
63   Serial.println("후진");
64   move_Backward();
65 }
66 }
67 }
68
69
70
71
72
73
74
75
76
77
Output
Ln 77, Col 1 Arduino Micro on COM5 [not connected]
```



```

DANVI_06_Bluetooth_Motor_Control_9button.ino | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_06_Bluetooth_Motor_Control_9button.ino motor_setting ...
1 #include "motor_setting.h"
2
3 // 13번 디지털핀(LED) 선언
4 #define led_pin 13
5
6 // 블루투스 받는 문자열
7 String bluetooth_string = "";
8
9 void setup()
10 {
11 // 시리얼통신 초기화(9600)bps
12 Serial.begin(9600);
13 //블루투스시리얼[하드웨어] 초기화(9600)bps
14 Serial1.begin(9600);
15 // 디지털핀 초기화(OUTPUT)
16 pinMode(led_pin, OUTPUT);
17 motor_output();
18 digitalWrite(led_pin, HIGH);
19 }
20
Output
Ln 9, Col 13 Arduino Micro on COM5 [not connected]
    
```

```

DANVI_06_Bluetooth_Motor_Control_9button.ino | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_06_Bluetooth_Motor_Control_9button.ino motor_setting ...
21 void loop()
22 {
23 // 블루투스시리얼[하드웨어] 명령 수신
24 if (Serial1.available())
25 { // Serial1[하드웨어] 값이 있으면
26 bluetooth_string = Serial1.read();
27 Serial.println(bluetooth_string);
28 }
29 //시리얼모니터에 bluetooth_string 값 출력
30 //bluetooth_string 값이 있다면
31 if (bluetooth_string != "")
32 {
33 // 실행(1~9)(전화버튼 기준):: ASCII
34 // 1 = "49"
35 if (bluetooth_string == "49")
36 {
37 Serial.println("좌전진");
38 move_LeftForCornering();
39 }
40 }
Output
Ln 9, Col 13 Arduino Micro on COM5 [not connected]
    
```

Chapter. 7 스마트폰으로 조종하는 프로그램 -9버튼

```
DANVI_06_Bluetooth_Motor_Control_9button.ino  motor_setting ...  
41 // 2 = "50"  
42 if (bluetooth_string == "50")  
43 {  
44   Serial.println("전진");  
45   move_Forward();  
46 }  
47  
48 // 3 = "51"  
49 if (bluetooth_string == "51")  
50 {  
51   Serial.println("우전진");  
52   move_RightForCornering();  
53 }  
54  
55 // 4 = "52"  
56 else if (bluetooth_string == "52")  
57 {  
58   Serial.println("왼쪽");  
59   move_Leftward();  
60 }
```

Output

Ln 9, Col 13 Arduino Micro on COM5 [not connected]

```
DANVI_06_Bluetooth_Motor_Control_9button.ino  motor_setting ...  
61 // 5 = "53"  
62 else if (bluetooth_string == "53")  
63 {  
64   Serial.println("정지");  
65   Stop_Moter();  
66 }  
67  
68 // 6 = "54"  
69 else if (bluetooth_string == "54")  
70 {  
71   Serial.println("오른쪽");  
72   move_Rightward();  
73 }  
74  
75 // 7 = "55"  
76 else if (bluetooth_string == "55")  
77 {  
78   Serial.println("좌후진");  
79   move_LeftBackCornering();  
80 }
```

Output

Ln 9, Col 13 Arduino Micro on COM5 [not connected]

```
DANVI_06_Bluetooth_Motor_Control_9button.ino  motor_setting ...  
82  
83 // 8 = "56"  
84 else if (bluetooth_string == "56")  
85 {  
86   Serial.println("후진");  
87   move_Backward();  
88 }  
89  
90 // 9 = "57"  
91 else if (bluetooth_string == "57")  
92 {  
93   Serial.println("우후진");  
94   move_RightBackCornering();  
95 }  
96  
97 }  
98  
99  
100  
101
```

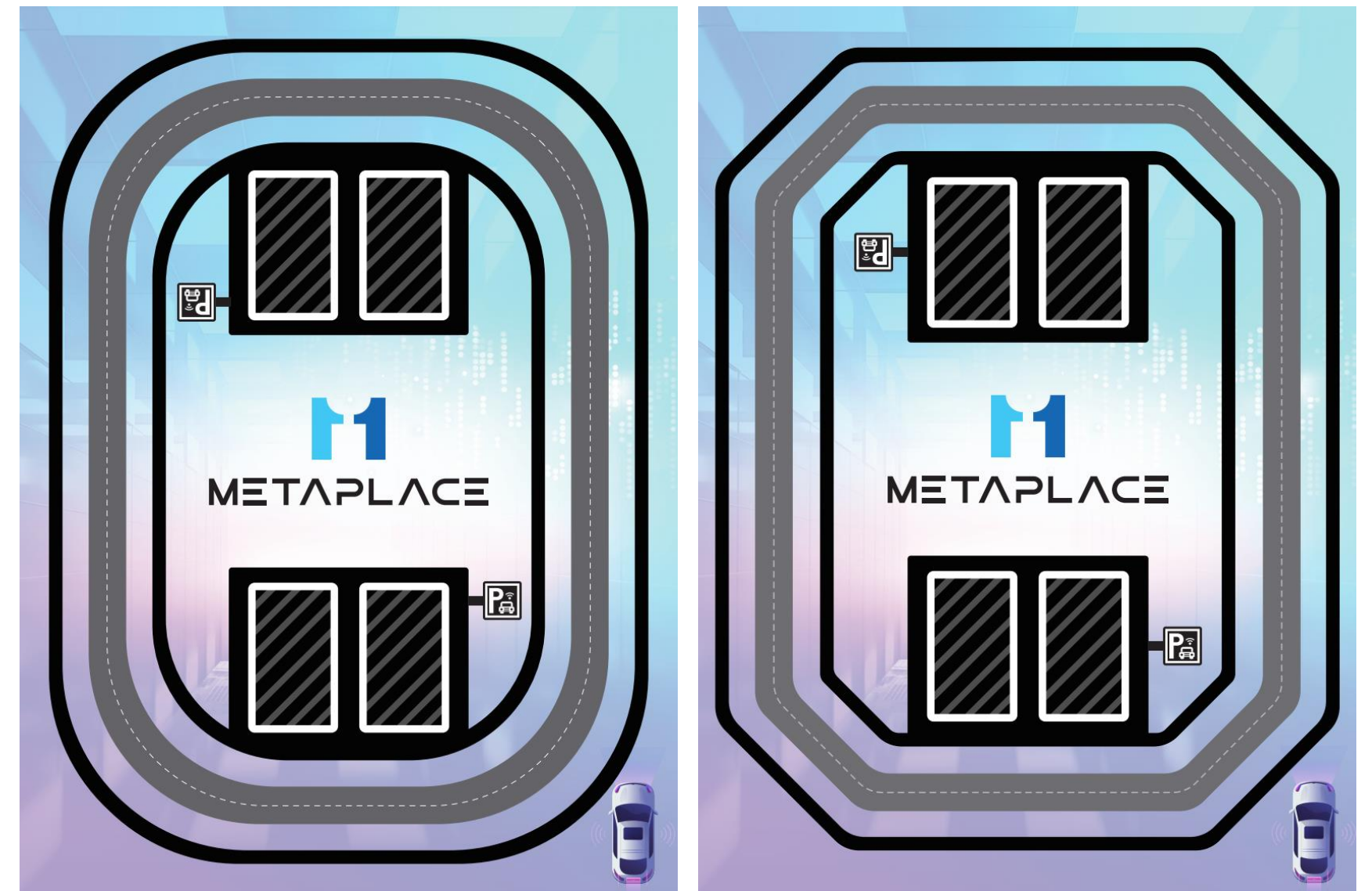
Output

Ln 10, Col 1 Arduino Micro on COM5 [not connected]

라인트레이서로 로봇 움직이기

[수업목표 : 적외선 선세로 라인값 확인하기]

- 적외선 센서의 흰색/ 검은색 변화 범위를 확인한다.
- 매트위에서 라인의 흰색/ 검색은 변화 값을 확인한다.



적외선센서를 이용한 라인트레이서

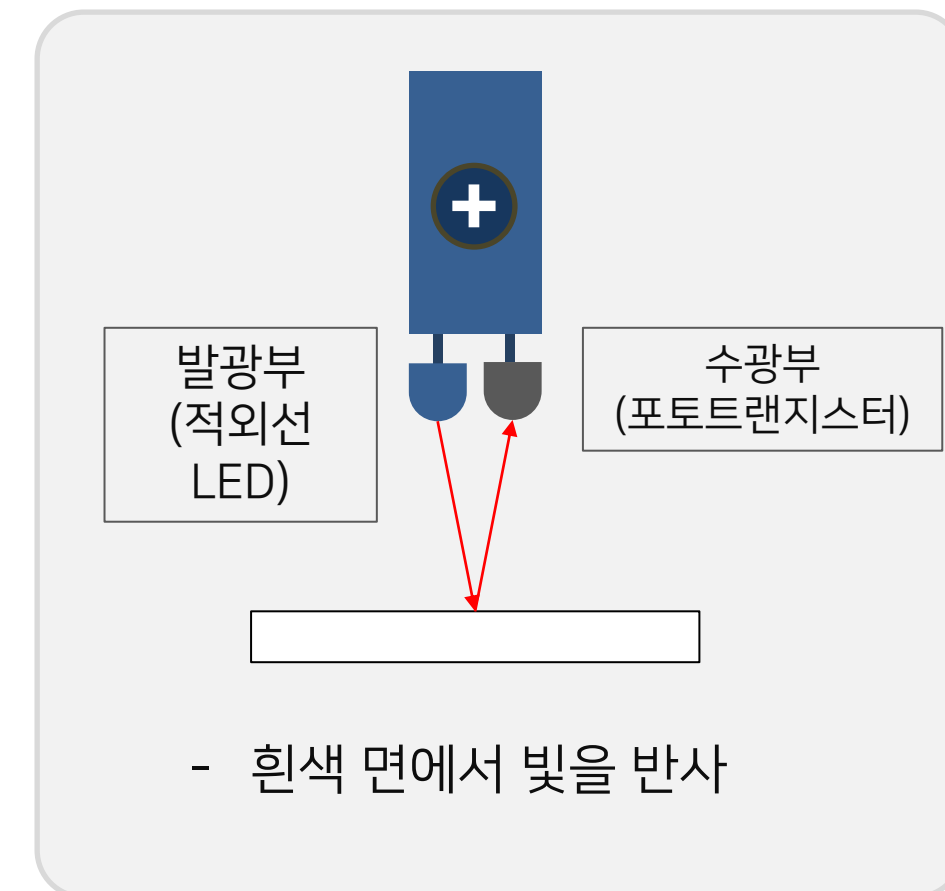
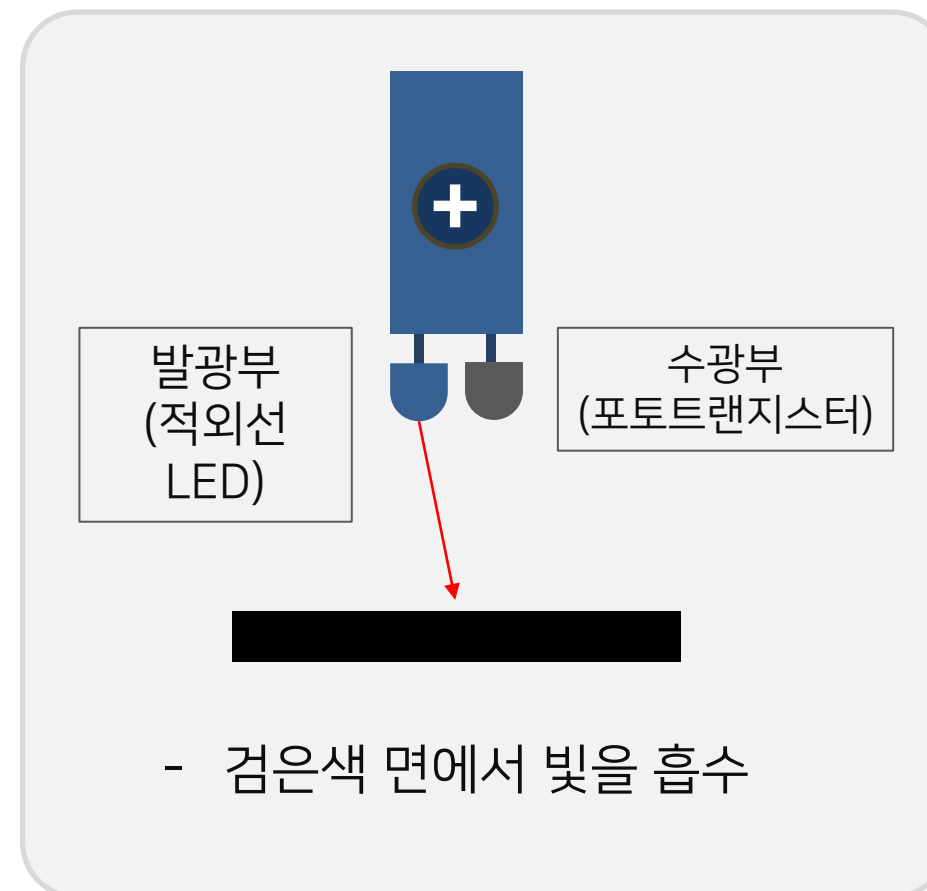
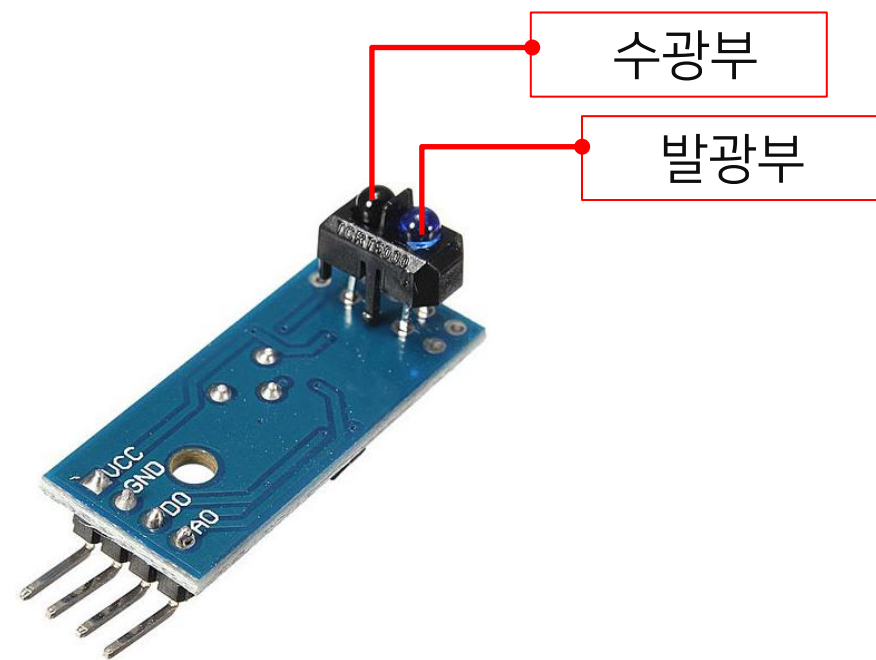
적외선센서를 알아봐요!

적외선 센서 : 두 개의 LED를 사용해서 주변을 감지해요. 이 빛은 사람의 눈에는 보이지 않아요.

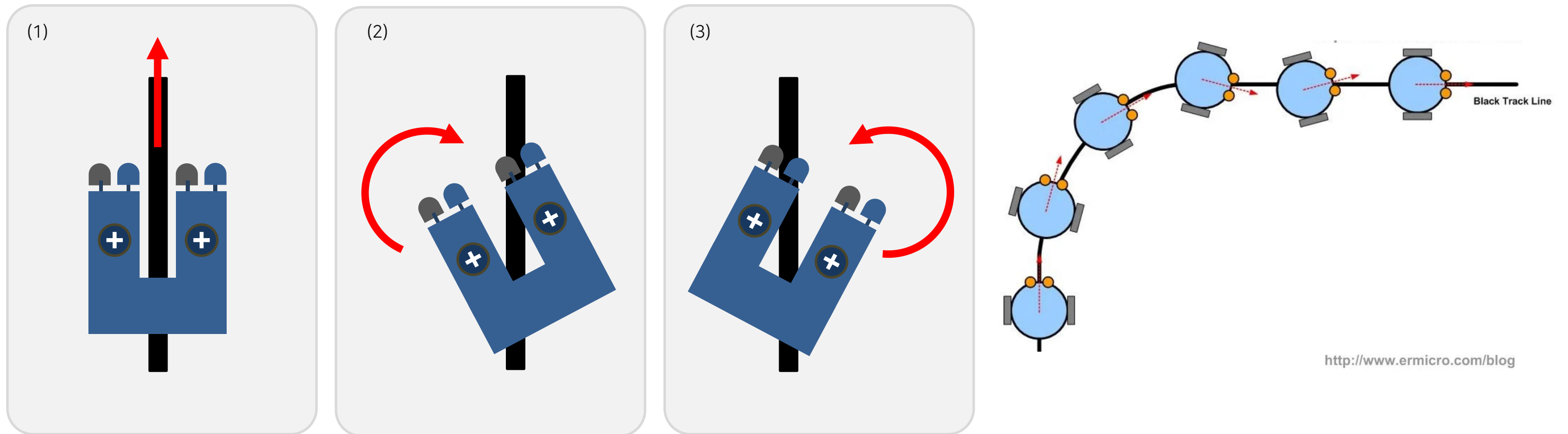
1) 발광부 LED : 적외선 빛을 내보내요.

2) 수광부 LED : 발광부에서 나온 적외선 빛이 반사되어 돌아오는 것을 감지해요.

- 적외선 센서는 2개의 LED가 있다. 하나는 발광부로 적외선 빛을 내보내는 적외선 LED이고, 다른 하나는 반사된 빛을 감지하는 수광부 LED 이다.
- 적외선은 사람의 눈으로 감지할 수 없는 긴 파장의 빛으로, 수광부 LED는 적외선 빛을 감지 가능하다.



- (1)번 그림과 같이 적외선센서가 길 좌우에 위치했을 때, 두 개의 적외선 센서는 모두 흰색 바닥을 검출하게 된다. 따라서 이 상태에서는 직진한다.
- (2)번 그림은 오른쪽 적외선 센서가 검은색에 위치하게 되고 선에 걸렸다는 것을 의미한다. 따라서, 인식된 반대편 센서의 모터를 조금 더 빠르게 회전 하여 센싱된 쪽으로 자동차 몸체를 휘도록 방향 전환을 시킨다.
- (3)번은 반대상황이며, 마지막 그림과 같이 검은색 선을 벗어나지 않도록 적외선 센서의 값을 확인하여 모터값을 변경하며 부드럽게 주행하게 한다.



<http://www.ermicro.com/blog>

DANVI_07_Line_Sensor_Drive | Arduino IDE 2.3.2

File Edit Sketch Tools Help

Arduino Micro

DANVI_07_Line_Sensor_Drive.ino motor_setting.h

```

1  #include "motor_setting.h"
2
3  // 13번 디지털핀(LED) 선언
4  #define led_pin 13
5  // A1번 아날로그핀(왼쪽 적외선센서)
6  #define line_L_pin A1
7  // A2번 아날로그핀(오른쪽 적외선센서)
8  #define line_R_pin A2
9  // 라인센서 변수선언
10 int line_L, line_R;
11 //교실상황에 따라 값을 25 - 50정도로 변경
12 int line_limit = 30;
13

```

Output

Ln 53, Col 26 Arduino Micro on COM5 [not connected]

DANVI_07_Line_Sensor_Drive | Arduino IDE 2.3.2

File Edit Sketch Tools Help

Arduino Micro

DANVI_07_Line_Sensor_Drive.ino motor_setting.h

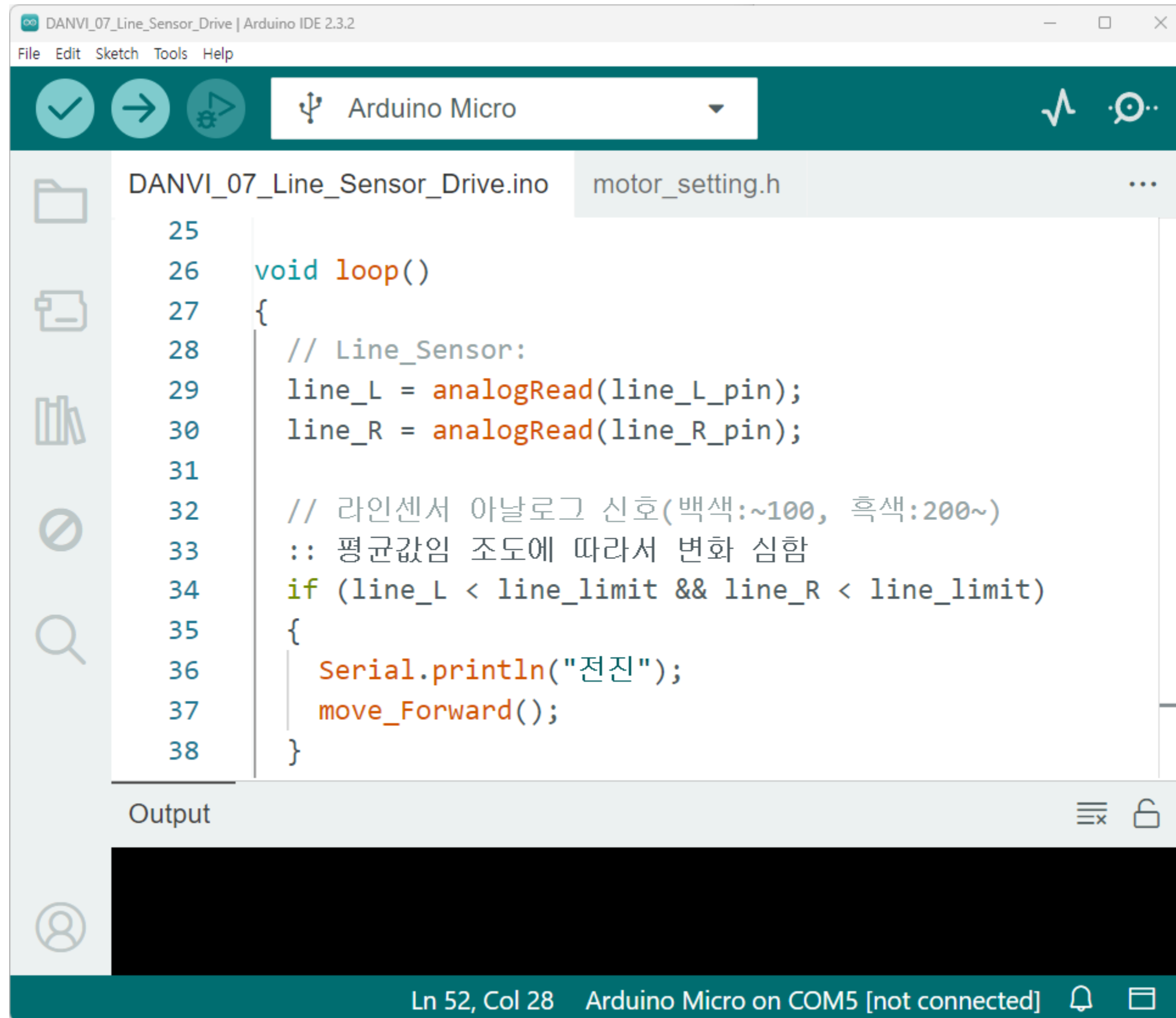
```

14 void setup()
15 {
16     // 시리얼통신 초기화(9600)bps
17     Serial.begin(9600);
18     //블루투스시리얼[하드웨어] 초기화(9600)bps
19     Serial1.begin(9600);
20     // 디지털핀 초기화(OUTPUT)
21     pinMode(led_pin, OUTPUT);
22     motor_output();
23     digital write(led_pin, HIGH);
24 }
25

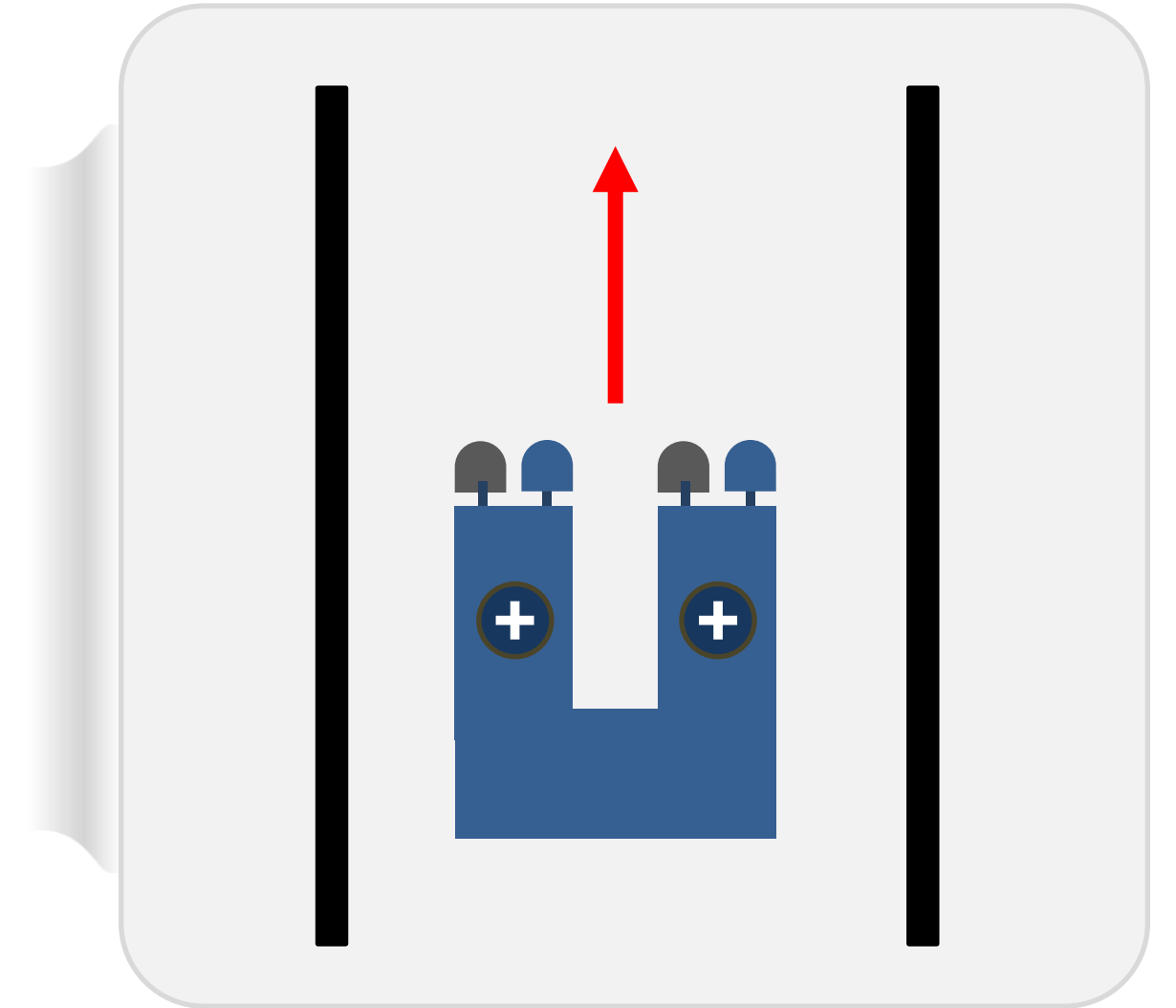
```

Output

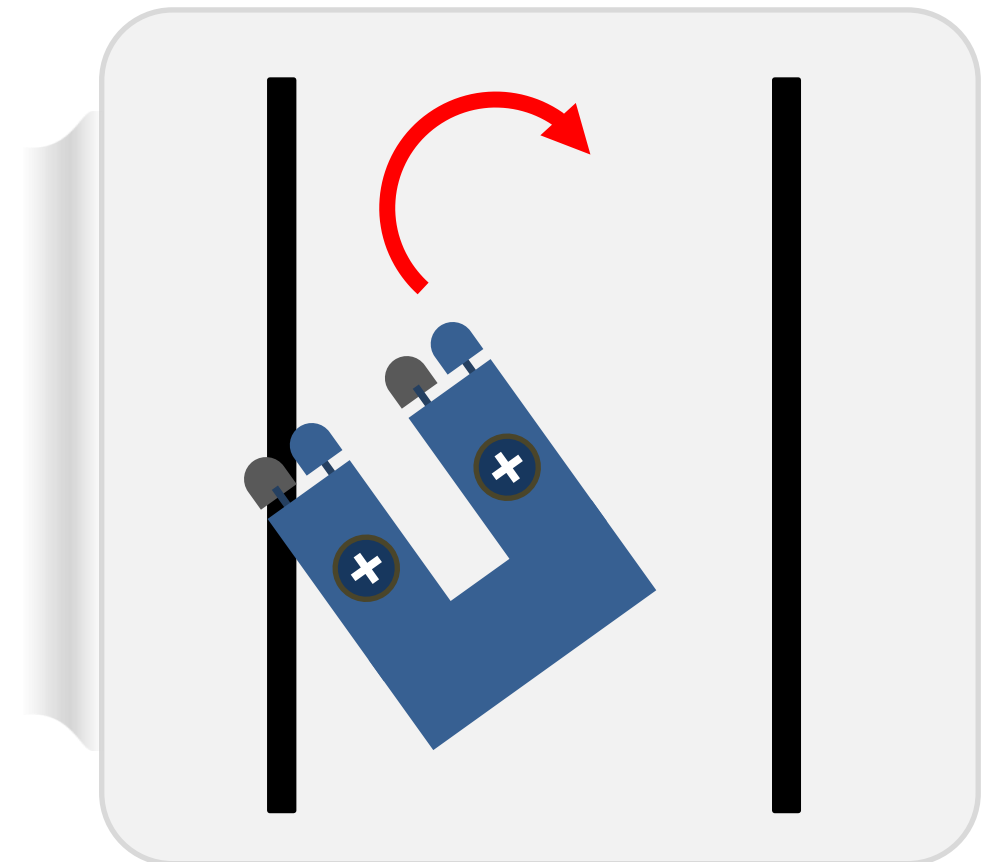
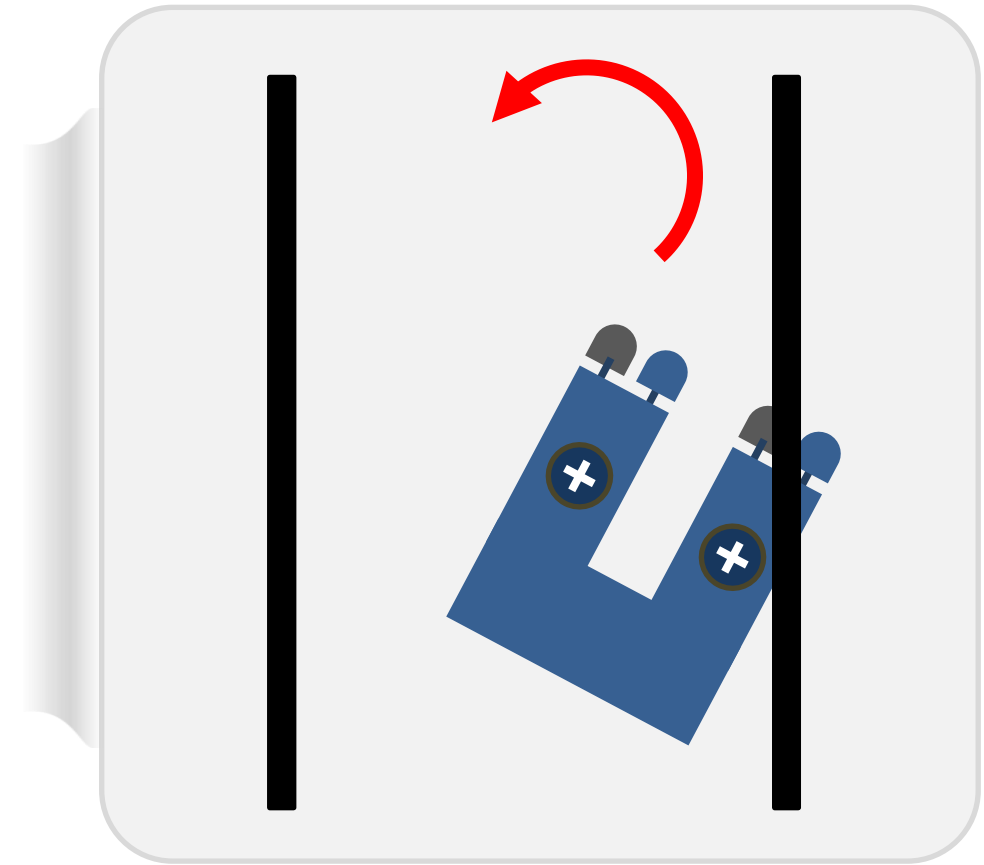
Ln 27, Col 1 Arduino Micro on COM5 [not connected]

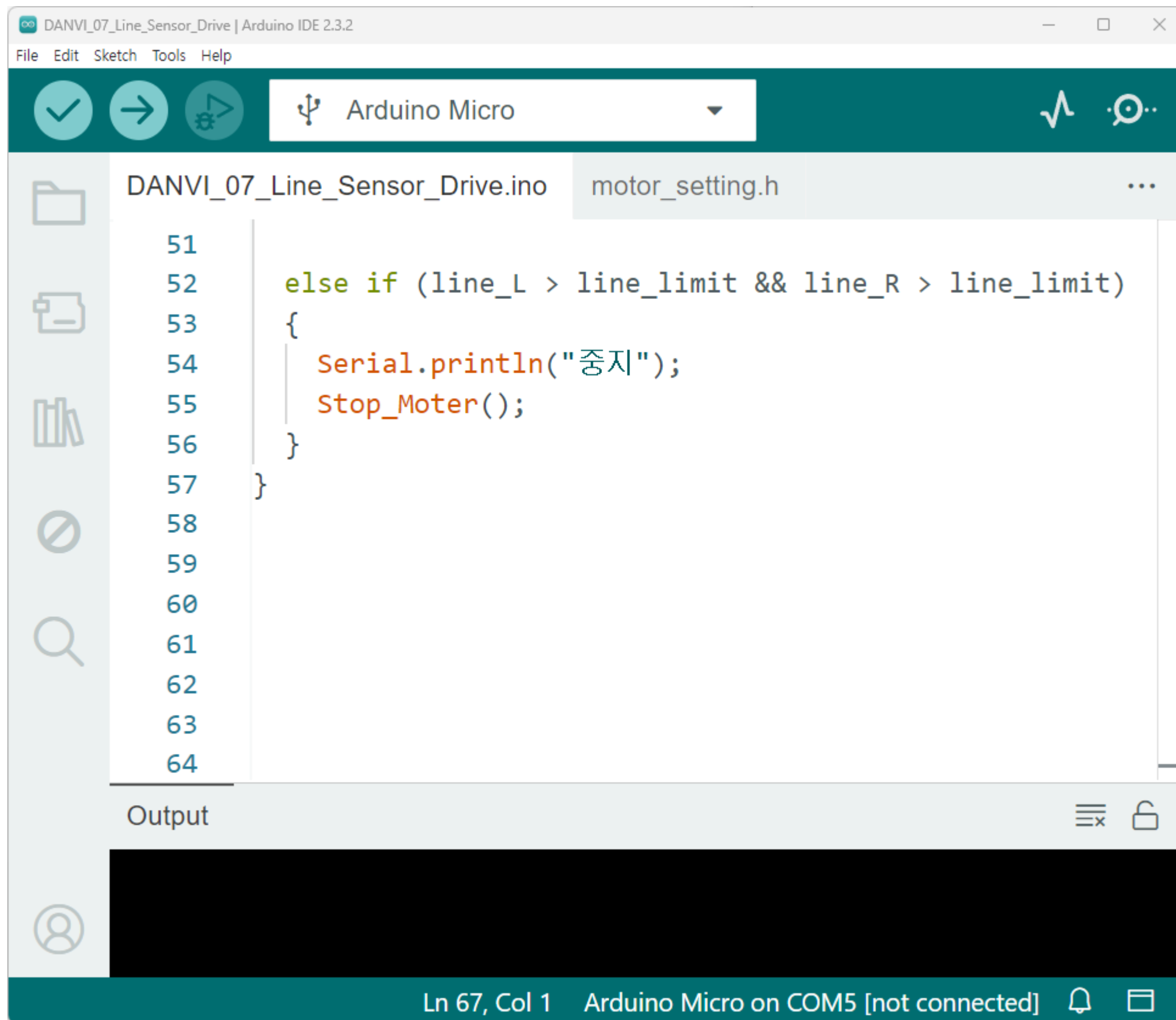


```
25
26 void loop()
27 {
28   // Line_Sensor:
29   line_L = analogRead(line_L_pin);
30   line_R = analogRead(line_R_pin);
31
32   // 라인센서 아날로그 신호(백색:~100, 흑색:200~)
33   :: 평균값임 조도에 따라서 변화 심함
34   if (line_L < line_limit && line_R < line_limit)
35   {
36     Serial.println("전진");
37     move_Forward();
38   }
```

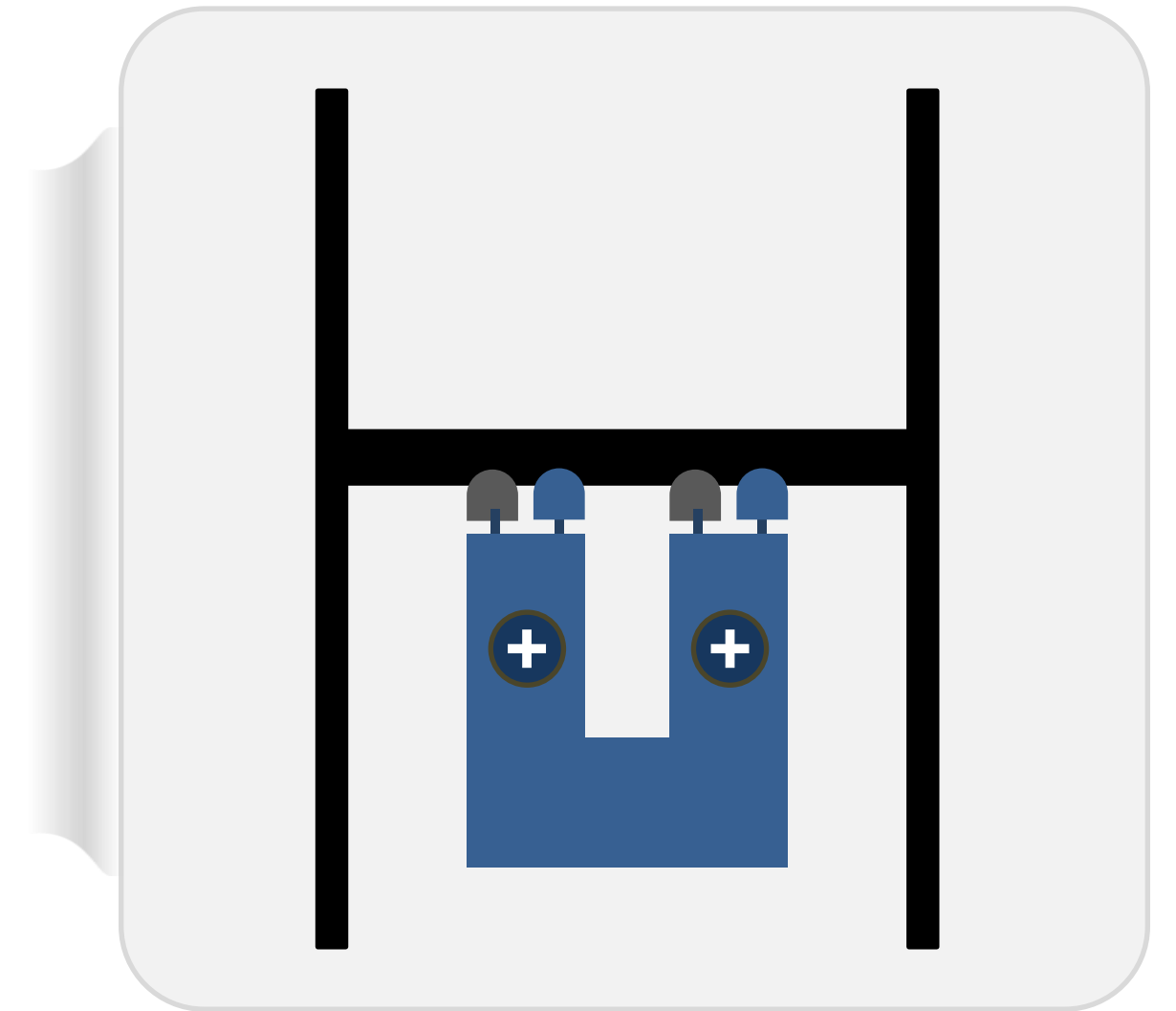


```
DANVI_07_Line_Sensor_Drive | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_07_Line_Sensor_Drive.ino motor_setting.h
39
40 else if (line_L < line_limit && line_R > line_limit)
41 {
42   Serial.println("좌회전");
43   move_LeftTurnRound();
44 }
45
46 else if (line_L > line_limit && line_R < line_limit)
47 {
48   Serial.println("우회전");
49   move_RightTurnRound();
50 }
51
52
Output
Ln 56, Col 1 Arduino Micro on COM5 [not connected]
```





```
DANVI_07_Line_Sensor_Drive | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Micro
DANVI_07_Line_Sensor_Drive.ino motor_setting.h
51
52 else if (line_L > line_limit && line_R > line_limit)
53 {
54     Serial.println("중지");
55     Stop_Moter();
56 }
57 }
58
59
60
61
62
63
64
Output
Ln 67, Col 1 Arduino Micro on COM5 [not connected]
```




티처블 머신으로 인공지능 학습에 대해 알아보기



<https://droidcam.softonic.kr/download>

Windows > 유틸리티 및 도구 > DroidCam > 다운로드




DroidCam PC 다운로드


무료 사용 언어: 한국어/조선말 v. 6.5.2
★ 3.9 (3416 ⭐) 보안 상태

DroidCam 무료 다운로드. Softonic 서버에서 항상 다운로드

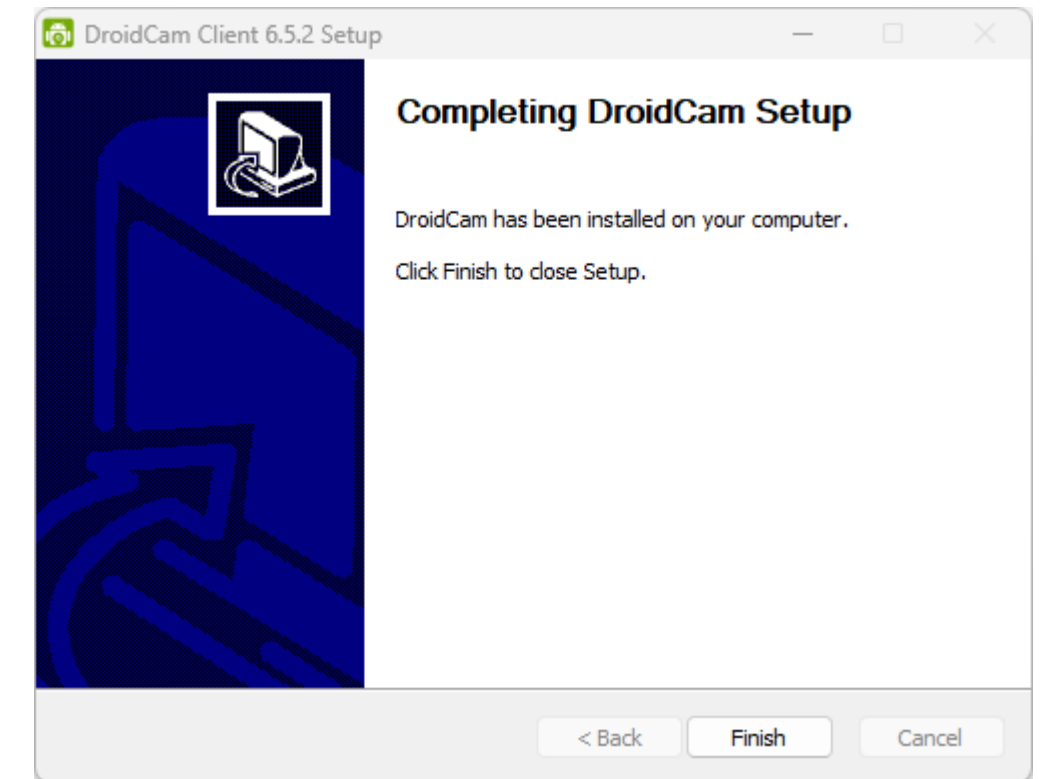
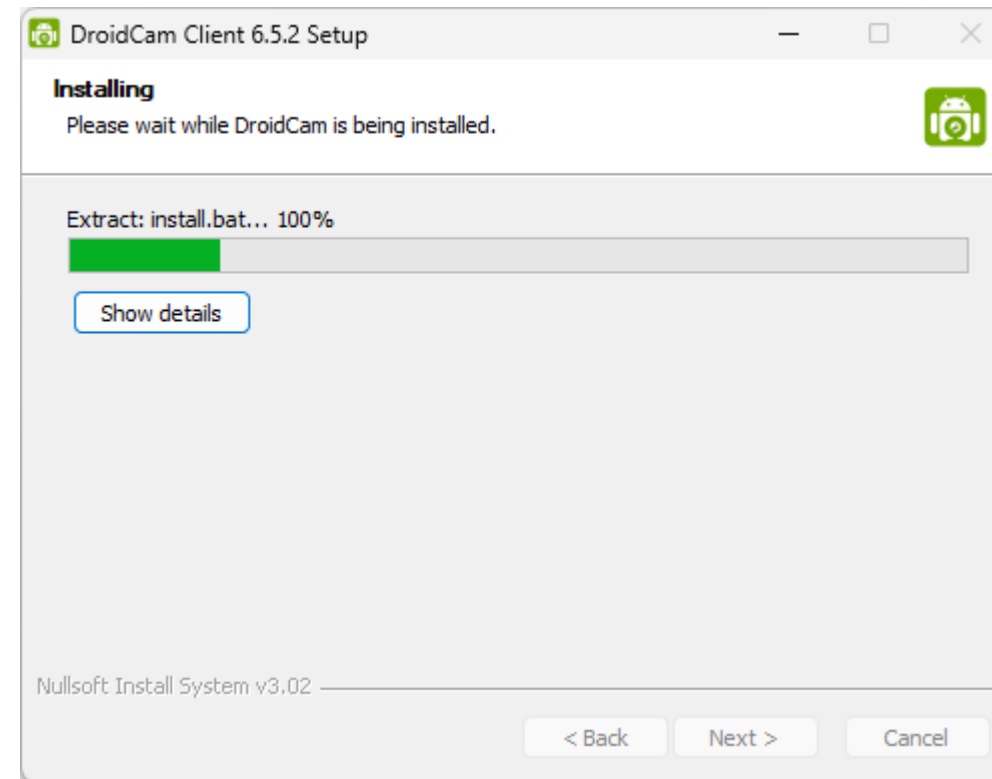
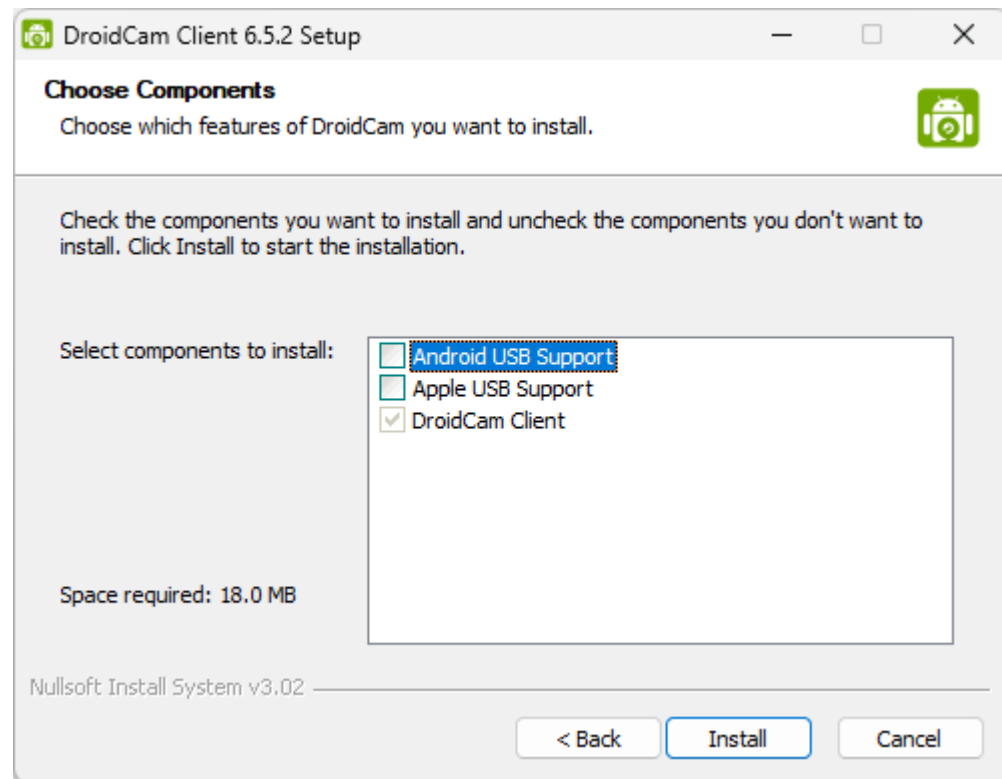
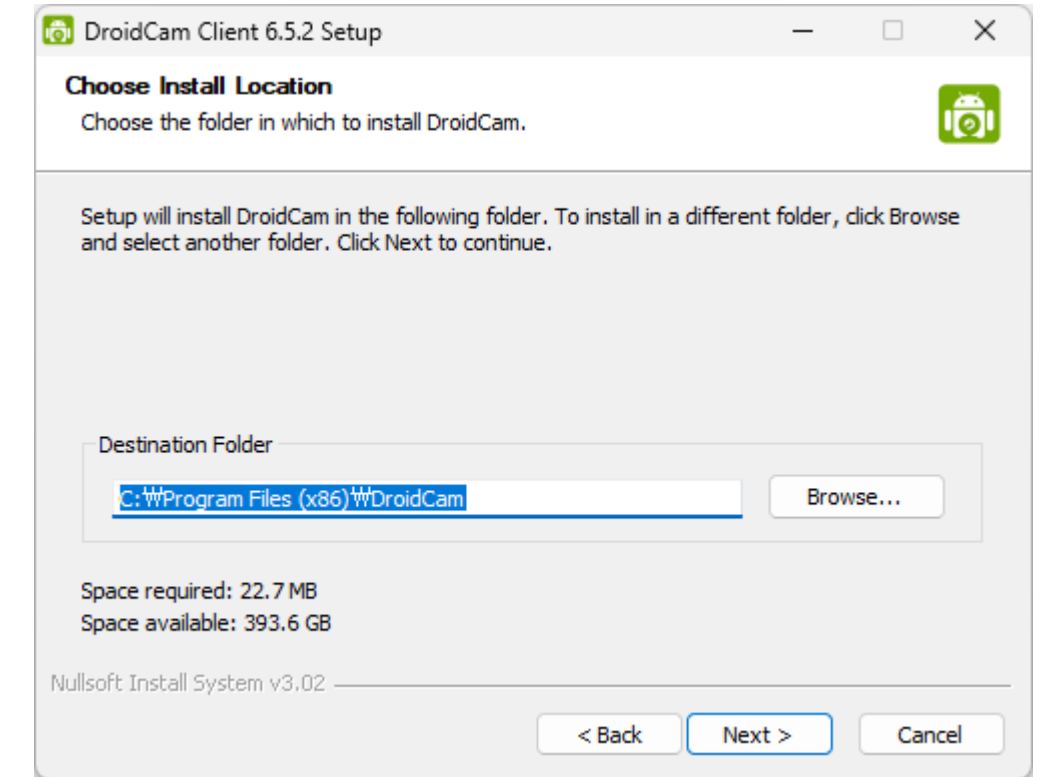
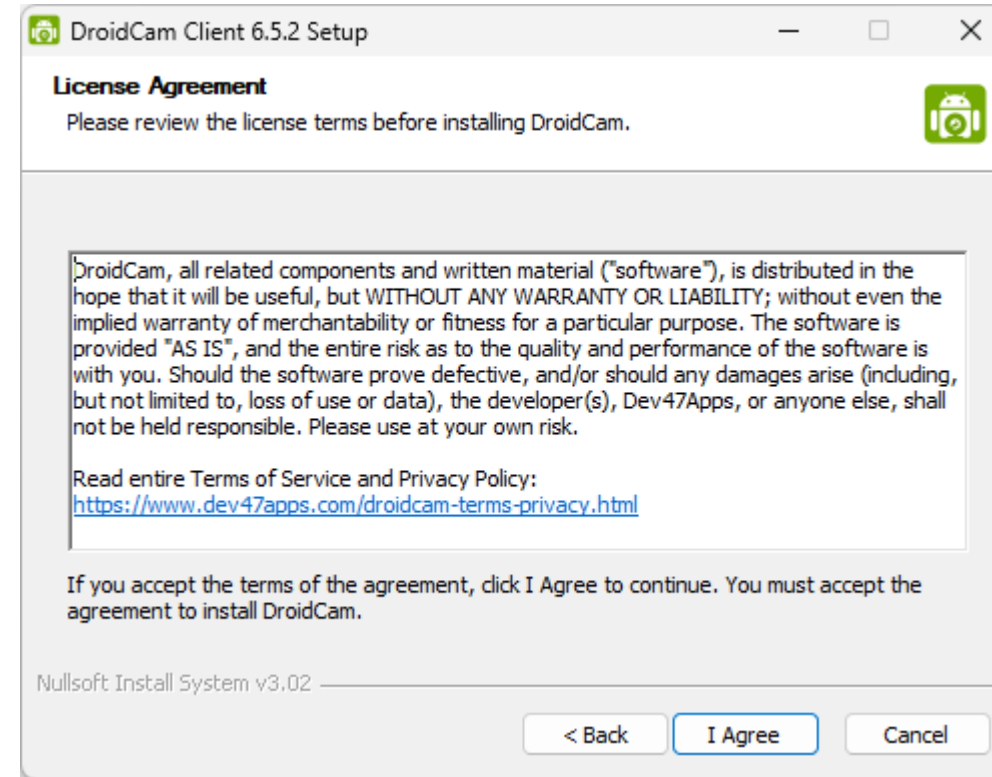
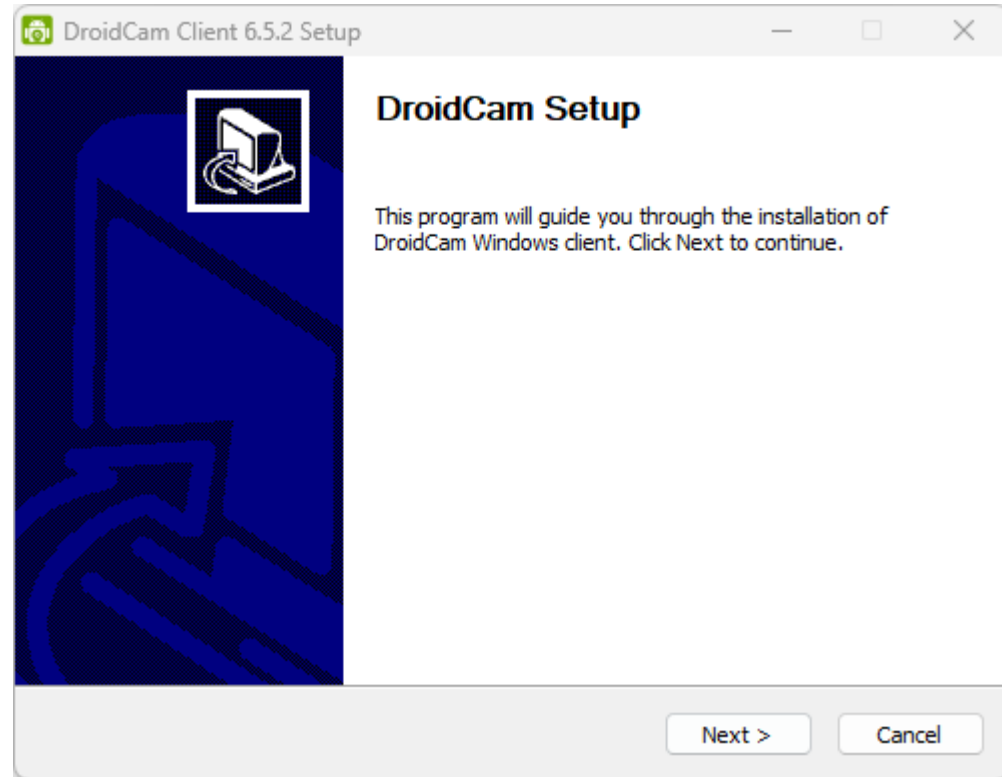
- ✓ 무료 및 빠른 다운로드
- ✓ 항상 사용 가능
- ✓ 바이러스가 없는지 테스트를 수행함

무료 다운로드 
PC 용

최근 다운로드 기록

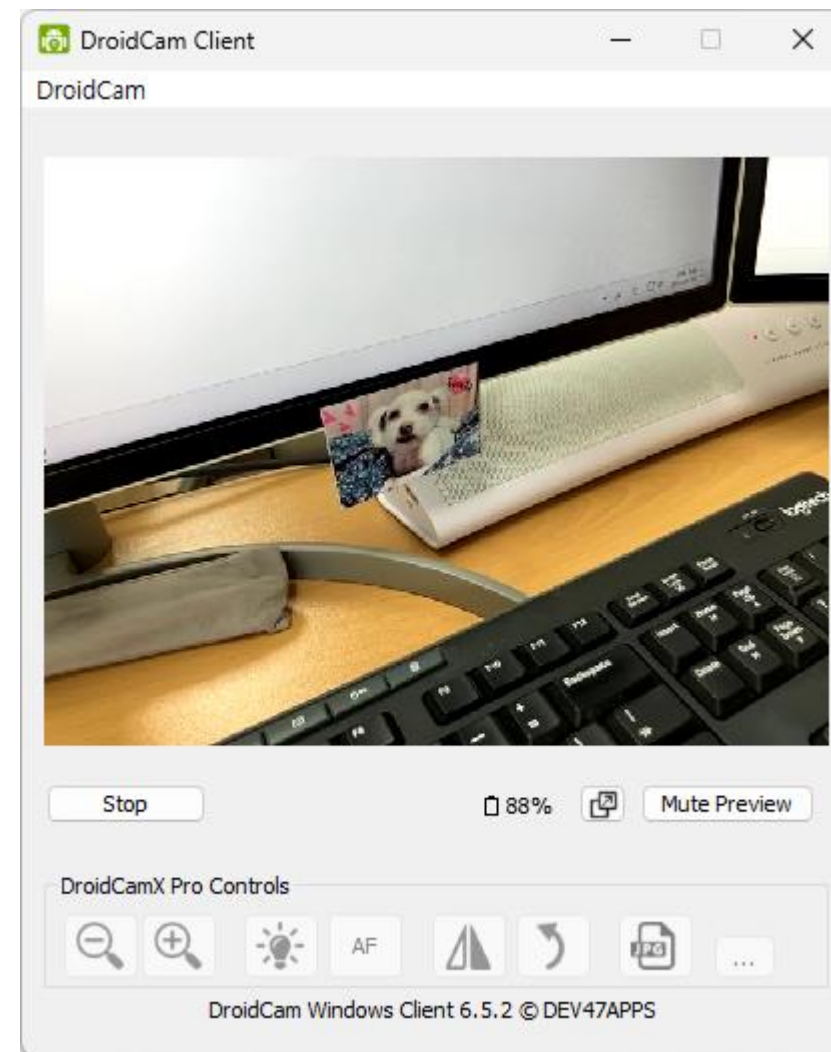
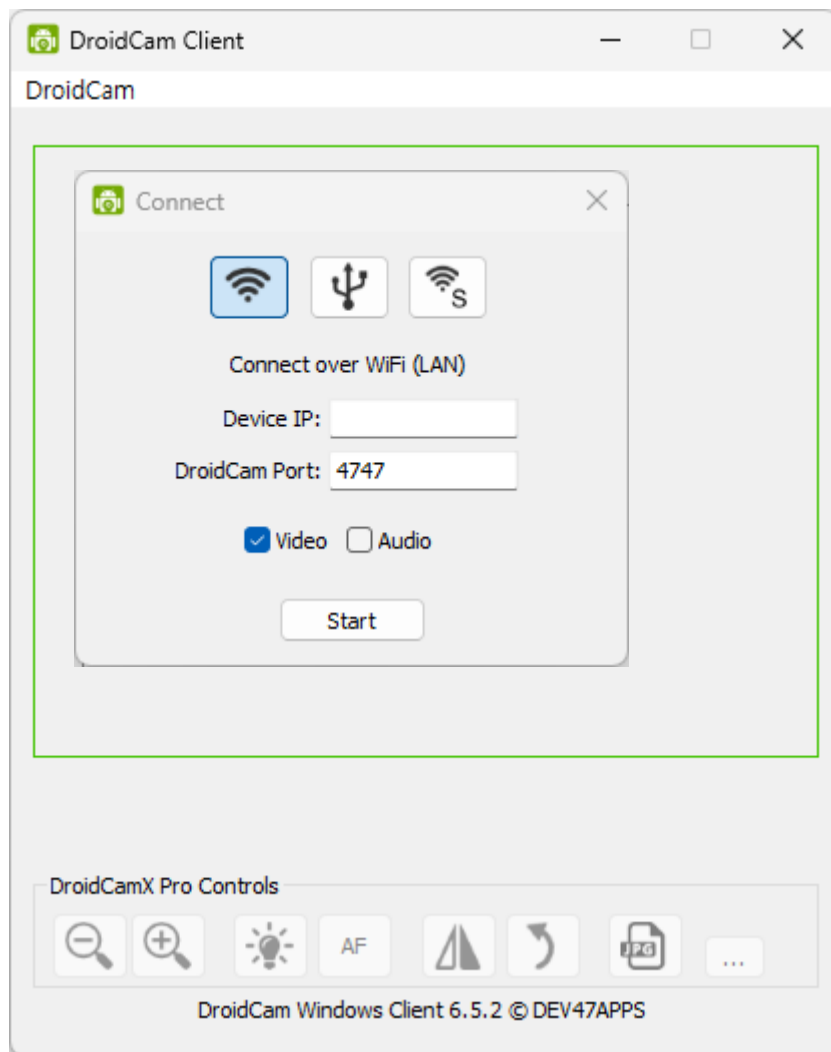


DroidCam.Setup.6.5.2.exe
↓ 2.3/15.6MB • 2분 남음

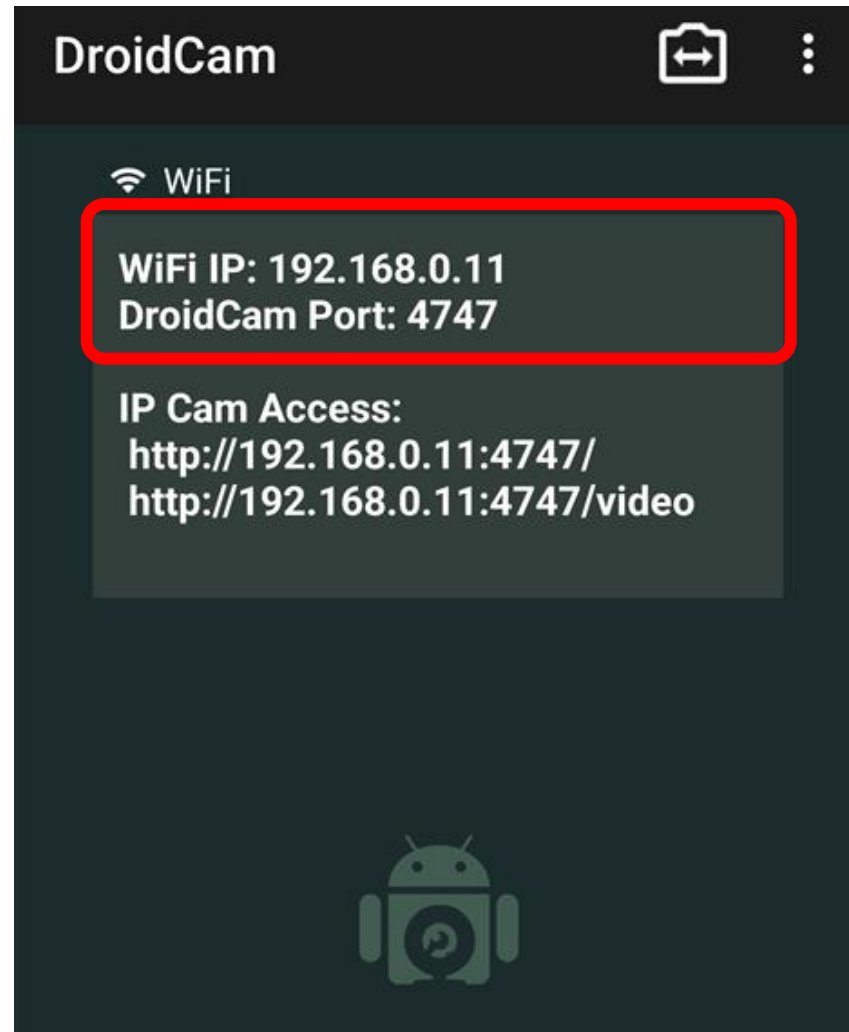


PC화면에서의 Droid Cam client

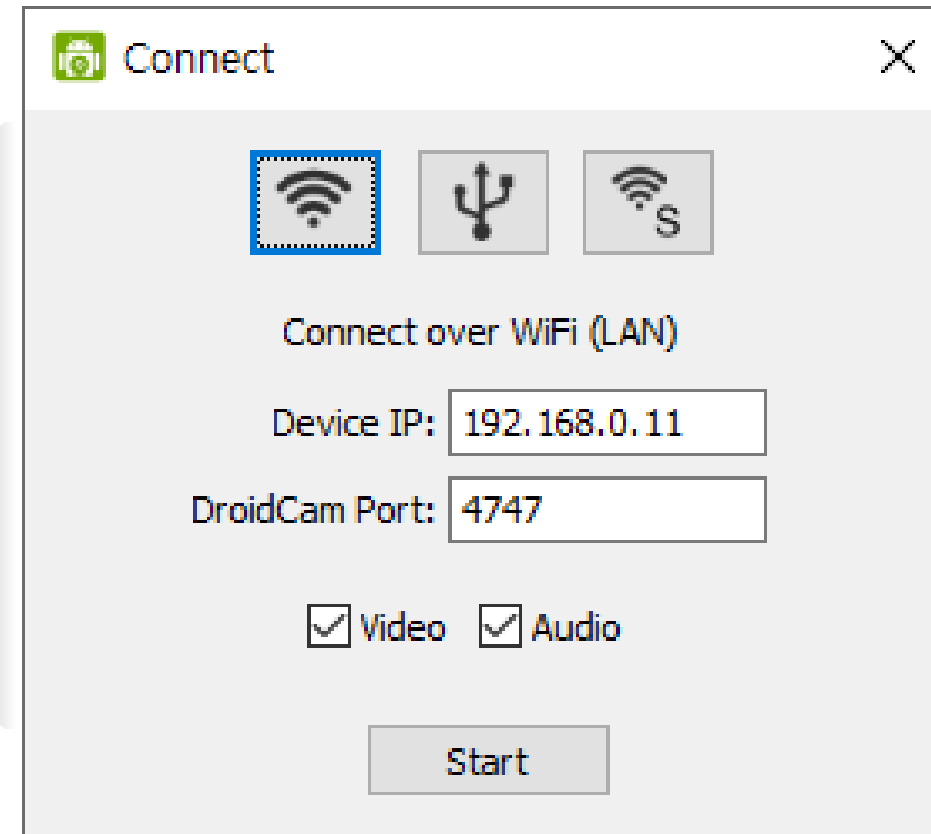
iOS 와 Android폰 아이콘 확인



스마트폰에 드로이드캠을 설치후 실행시키면 다음과 같이 화면이 나타납니다.



[스마트폰 드로이드캠]



[PC 드로이드캠 실행화면]

- 무선와이파이를 이용하기 위해서 스마트폰과 PC가 동일한 무선 와이파이에서 연결되어 있어야 함.
- WiFi 번호와 DroidCam Port 번호를 PC의 드로이드캠에 똑같이 입력합니다.
- 첫 번째 아이콘은 무선으로 웹캠을 연결하겠다는 버튼으로, IP와 Port 번호를 입력후 [Start]버튼을 클릭합니다.

티처블머신은 초보자도 쉽게 머신러닝 모델을 만들 수 있도록 제작된 웹기반 도구입니다.
구글에서 만든 인공지능 엔진으로, CNN기반의 Single object만 학습 가능한 인공지능 플랫폼입니다.

티처블머신이라고 웹브라우저에서 검색하거나, <https://teachablemachine.withgoogle.com/> 접속하시면 됩니다.

정보 FAQ 시작하기

Teachable Machine

이미지, 사운드, 자세를 인식하도록 컴퓨터를 학습시키세요.

사이트, 앱 등에 사용할 수 있는 머신러닝 모델을 쉽고 빠르게 만들어 보세요. 전문지식이나 코딩 능력이 필요하지 않습니다.

시작하기

Marshmallow 60%


Not Marshmallow 40%


↑ ml5 p5.js Coral node ARDUINO


3가지 프로젝트를 사용할 수 있습니다.

- **이미지 프로젝트** : 이미지를 분류하는 모델을 생성할 수 있습니다.
- **오디오 프로젝트** : WAV, MP3 오디오 파일을 분류하는 모델을 생성할 수 있습니다.
- **포즈 프로젝트** : 카메라에 비춰지는 포즈를 분류하는 모델을 생성할 수 있습니다.

새 프로젝트

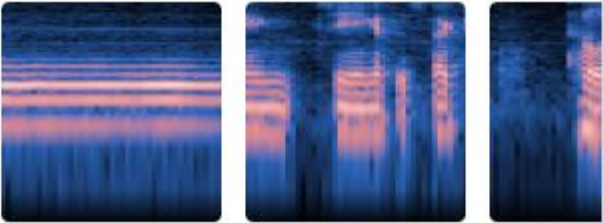
 Drive에 있는 기존 프로젝트를 엽니다.

 파일에서 기존 프로젝트를 엽니다.




이미지 프로젝트

파일 또는 웹캠에서 가져온 이미지를 기반으로 학습시키세요.



오디오 프로젝트

파일 또는 마이크에서 가져온 1초 분량의 사운드를 기반으로 학습시키세요.



포즈 프로젝트

파일 또는 웹캠에서 가져온 이미지를 기반으로 학습시키세요.

우리는 이미지 프로젝트를 활용하여 2가지 이미지를 학습시켜 구분하는 프로젝트를 만들어 보겠습니다.
표준 이미지 모델을 선택하여 프로젝트를 실행합니다.

새 이미지 프로젝트

표준 이미지 모델

대부분의 용도에 적합

224 x 224px 컬러 이미지

TensorFlow, TFLite, TF.js로 내보내기

모델 크기: 약 5mb

삼입된 이미지 모델

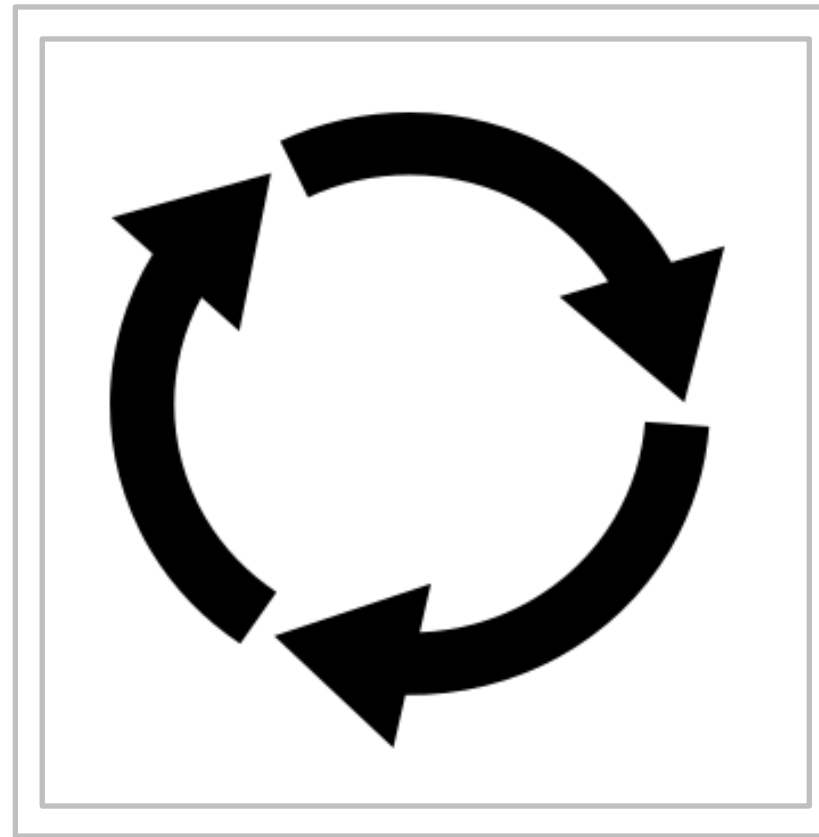
마이크로 컨트롤러에 적합

96 x 96px 그레이스케일 이미지

마이크로컨트롤러용 TFLite, TFLite, TF.js로 내보내기

모델 크기: 약 500kb

[이 모델을 지원하는 하드웨어를 확인하세요.](#)



먼저, 샘플을 수집하고, 모델을 학습시키고 미리보기 과정으로 이루어져 있습니다.

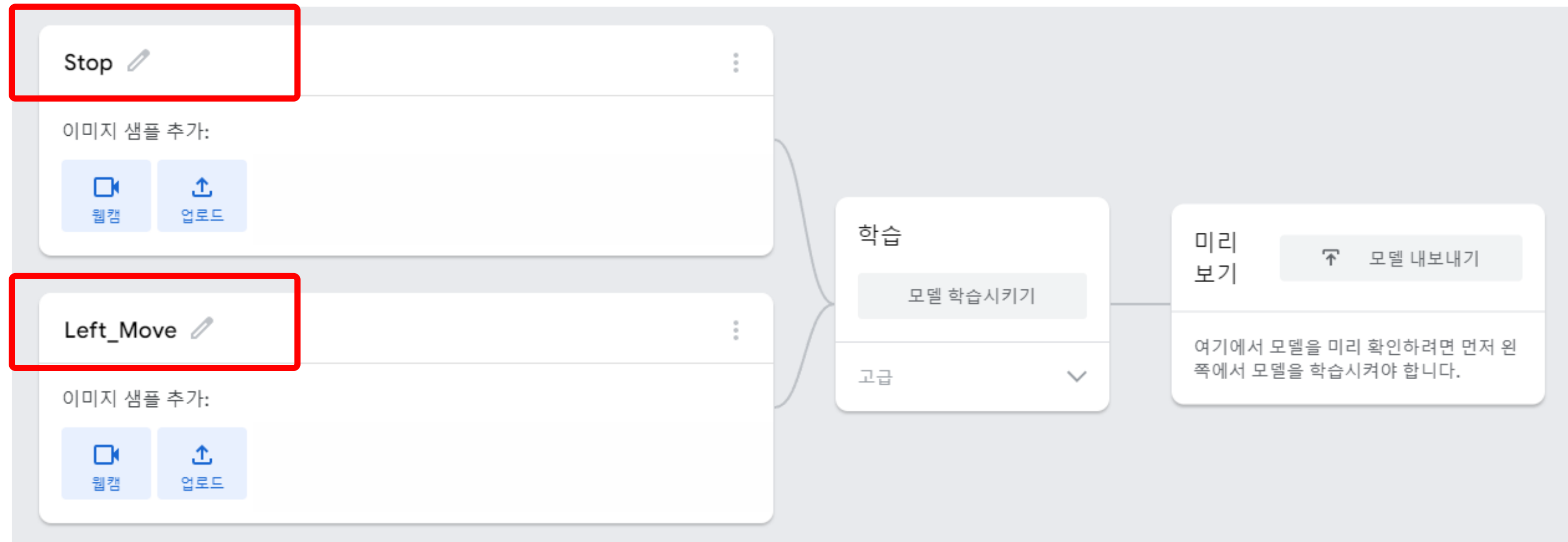
Class1과 Class2의 이름을 그대로 사용해도 되고, 분류하고자 하는 Class명으로 변경할 수도 있습니다.

2개의 예제 이미지를 학습시켜 보겠습니다.



첫 번째 클래스의 이름은 Stop으로 변경하고, 두 번째 클래스의 이름은 Left_Move로 변경하였습니다.

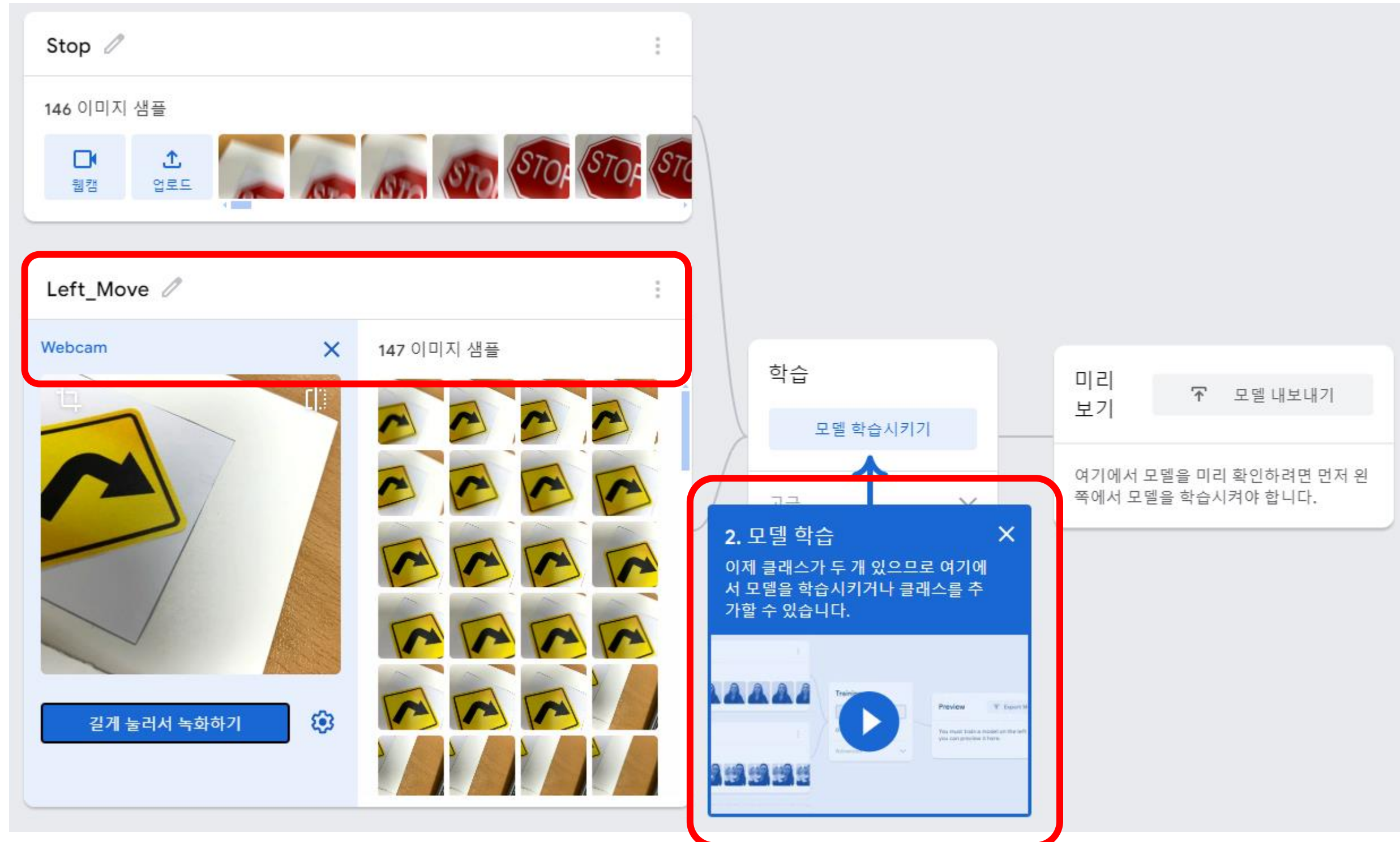
웹캠을 눌러 카메라의 화면을 확인합니다. 만약 카메라가 인식되지 않으면, 웹캠 전환으로 드로이드캠과 연결하도록 합니다.



스마트폰 카메라에 접속을 허용하고, 길게 눌러서 녹화하기 버튼으로 최대한 많은 이미지 샘플을 저장한다.



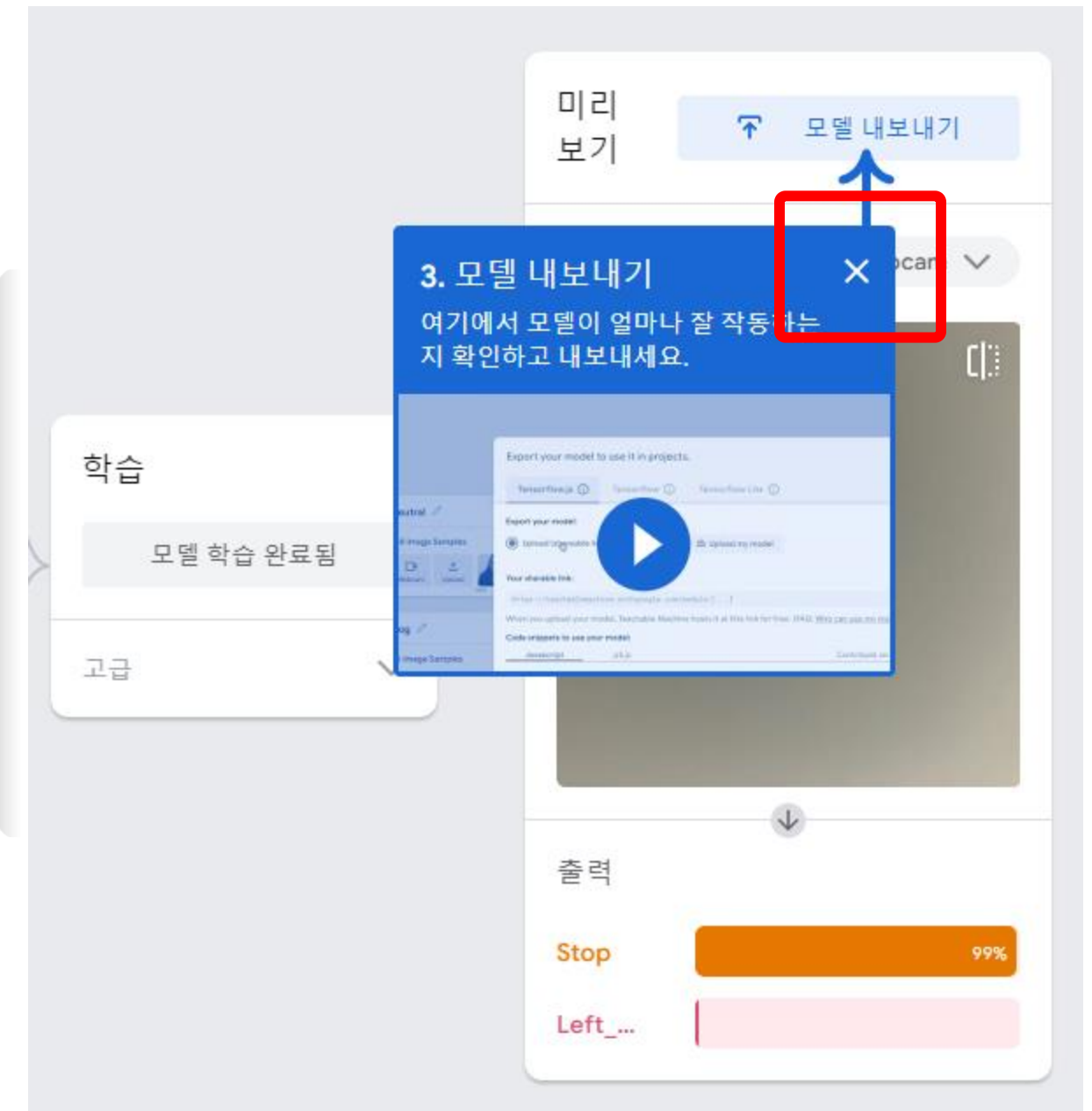
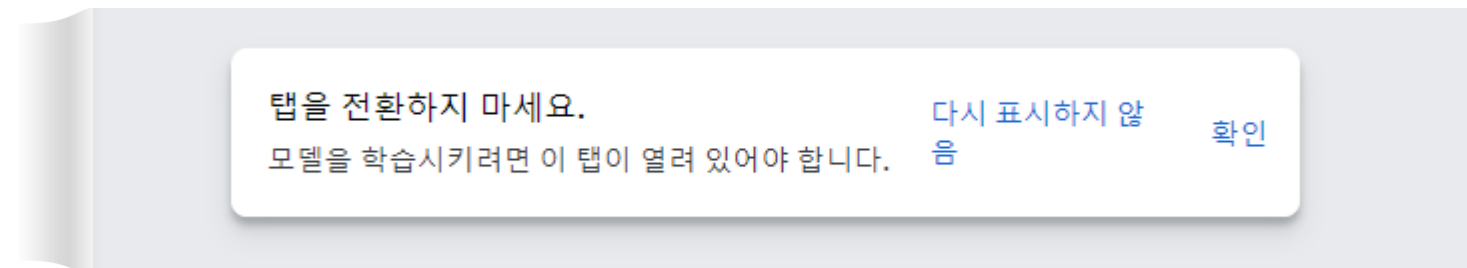
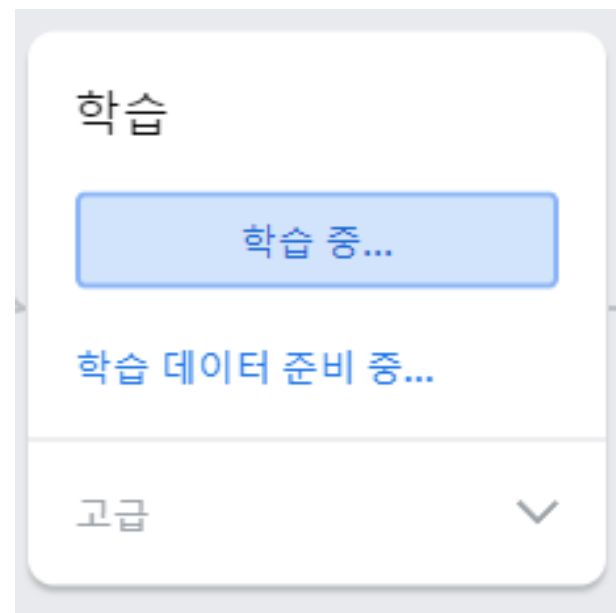
두번째 클래스의 이미지를 첫번째 클래스 이미지 수와 비슷하게 캡처하고,
클래스를 더 추가하거나, 모델학습을 시킬 수 있음.



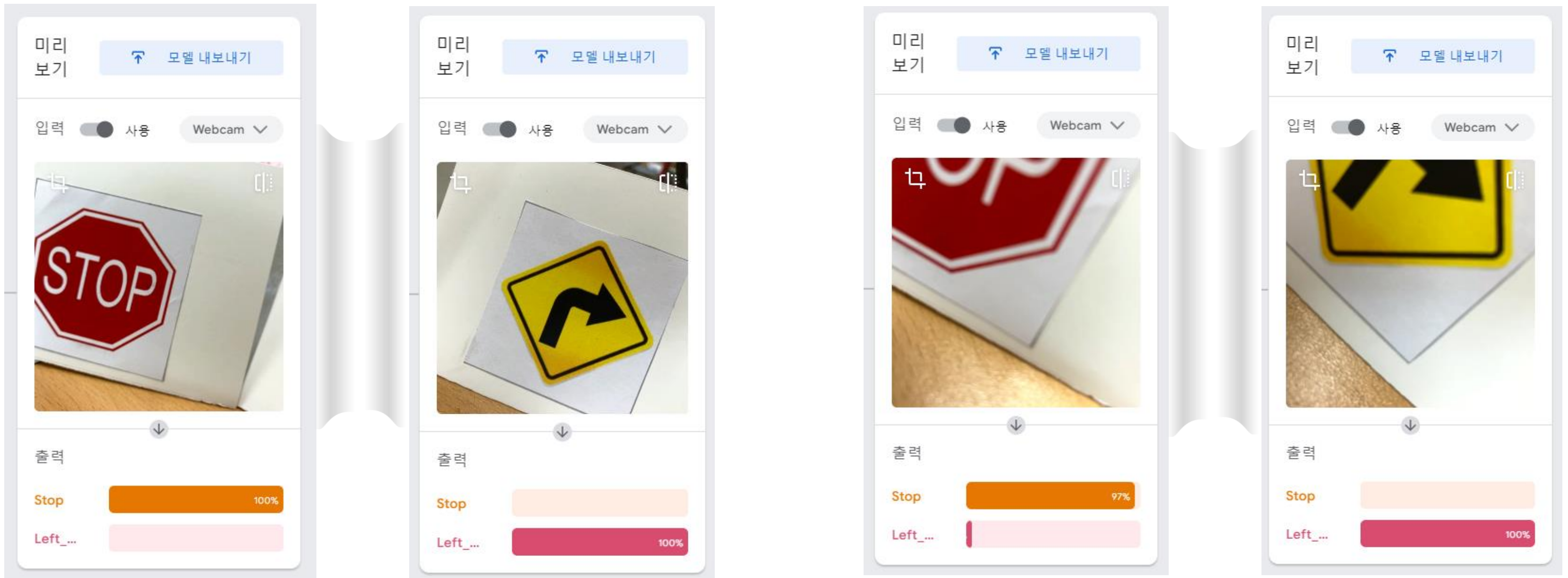
학습중에는 탭을 전환하지 마세요.

모델 학습이 완료되면 다음 단계인 미리보기에서 출력 인식을 확인할 수 있어요.

얼마나 잘 학습되었는지 확인해 볼까요?



첫번째 클래스의 이미지와 두번째 클래스의 이미지를 모두 잘 보이게 했을때,
이미지의 모서리 부분만 보여줘도 잘 인식되는 것을 확인할 수 가 있어요!



Chapter. 8 인공지능 고급 옵션 확인하기 - 이부분은 건너뛰어도 됩니다.

[고급] 스텝을 클릭하면 [에포크], [배치크기],[학습률]을 조정할 있다.

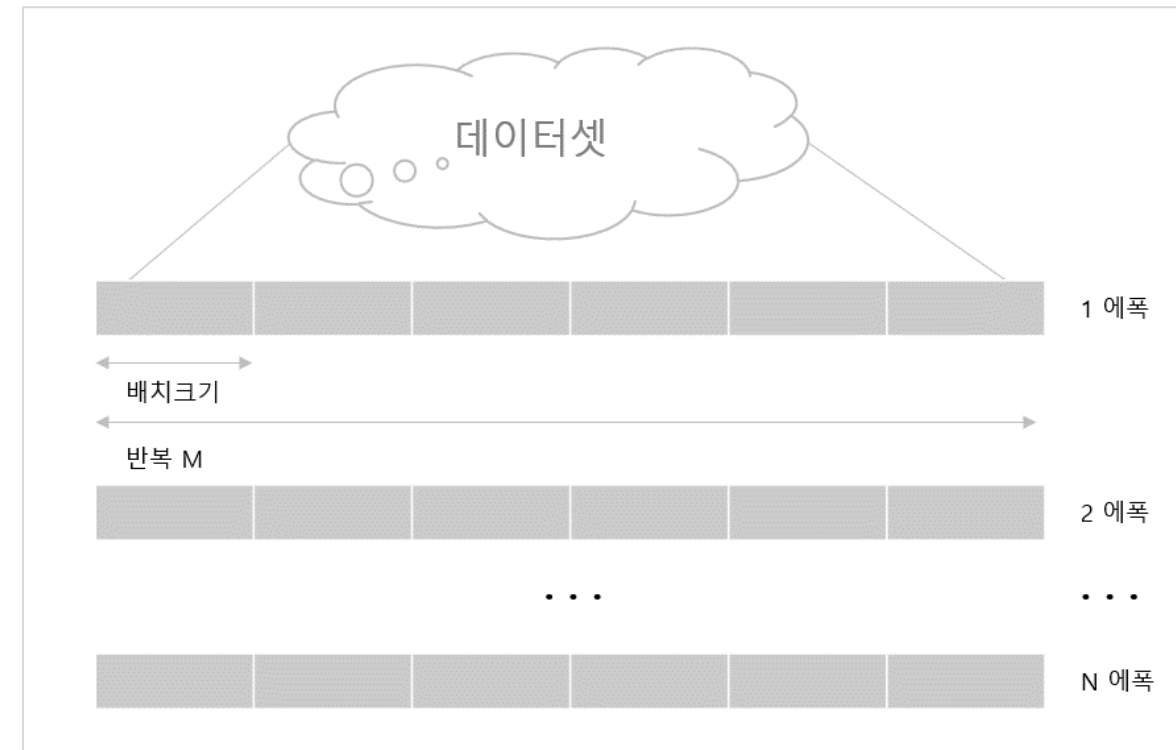
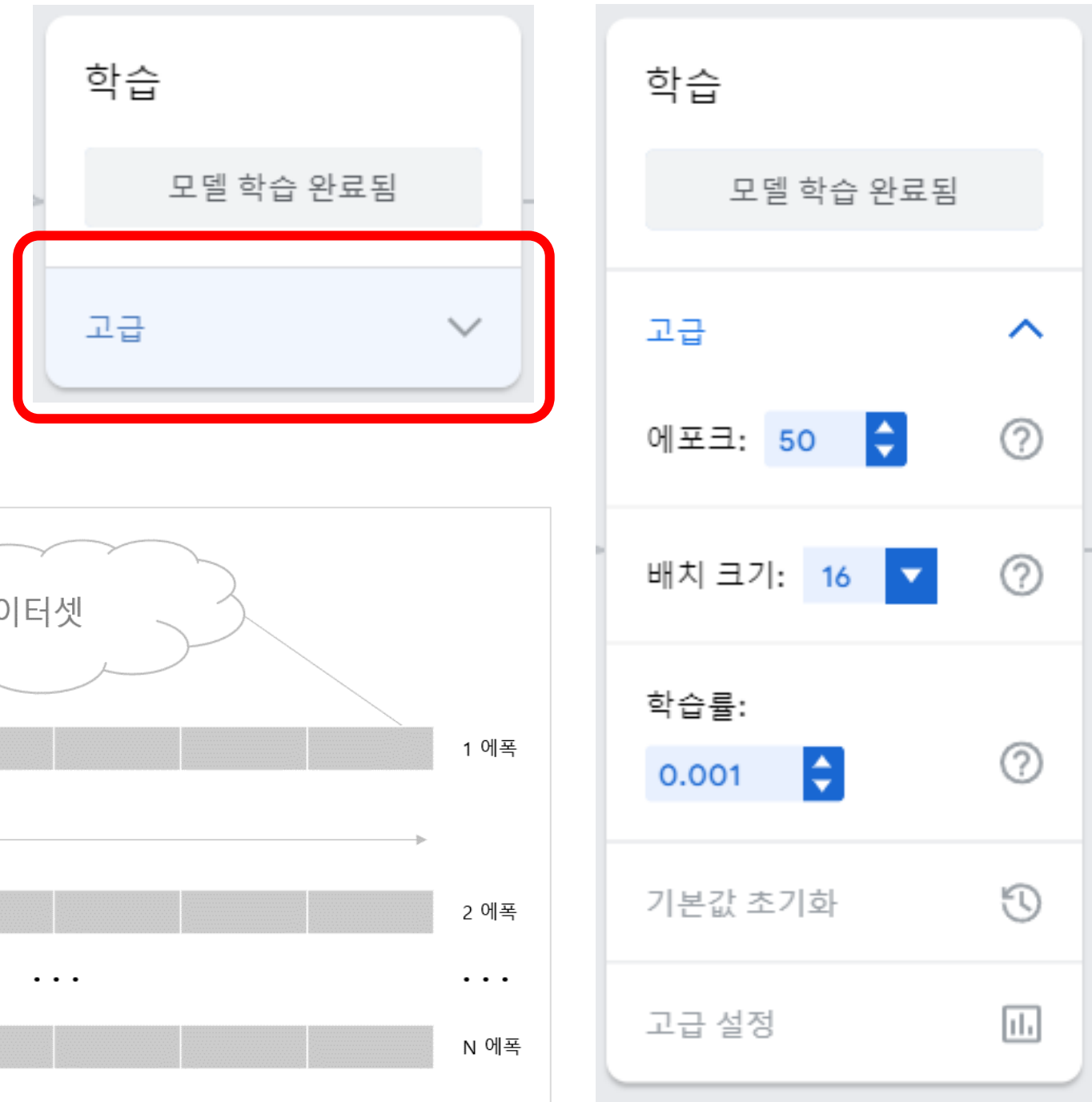
[에포크]란 전체 학습을 트레이닝 시키는 횟수를 의미한다.

훈련 데이터셋에 포함되어 있는 데이터를 한번씩 학습 모델에 통과한 횟수로 지정하는 의미가 되며, 에포크 10이라면, 학습데이터 셋을 10회 학습시켰다는 것이다.

[배치크기]란 연산에 들어가는 데이터 크기, 1회 에포크 안에 m개의 배치가 들어가는 크기를 생각하면 된다.

배치크기가 너무 크면 한번에 처리해야할 데이터 양이 많아 지므로 학습 속도가 느려지고, 메모리 부족문제가 발생할수 있고, 반대로 너무 작으면 데이터를 대상으로 가중치를 업데이트하게 되는데 이 업데이트가 자주 발생하므로, 훈련이 불안정해진다.

[학습률]은 훈련의 속도이다. 일반적인 머신 러닝에서 학습률(learning rate)은 중요한 하이퍼파라미터로, 모델의 가중치를 업데이트할 때 사용되는 스칼라 값입니다. 학습률은 모델이 훈련 데이터에서 얼마나 빠르게 학습해야 하는지를 제어하며, 적절한 학습률을 선택하는 것은 모델의 성능을 크게 영향을 미칩니다.



학습률은 보통 0과 1 사이의 값으로 설정되며, 학습률이 작으면 모델이 더 느리게 수렴하지만 안정적인 결과를 얻을 수 있고, 학습률이 크면 수렴이 빨라질 수 있지만 발산할 위험이 있습니다. 따라서 적절한 학습률을 선택하는 것이 중요합니다.

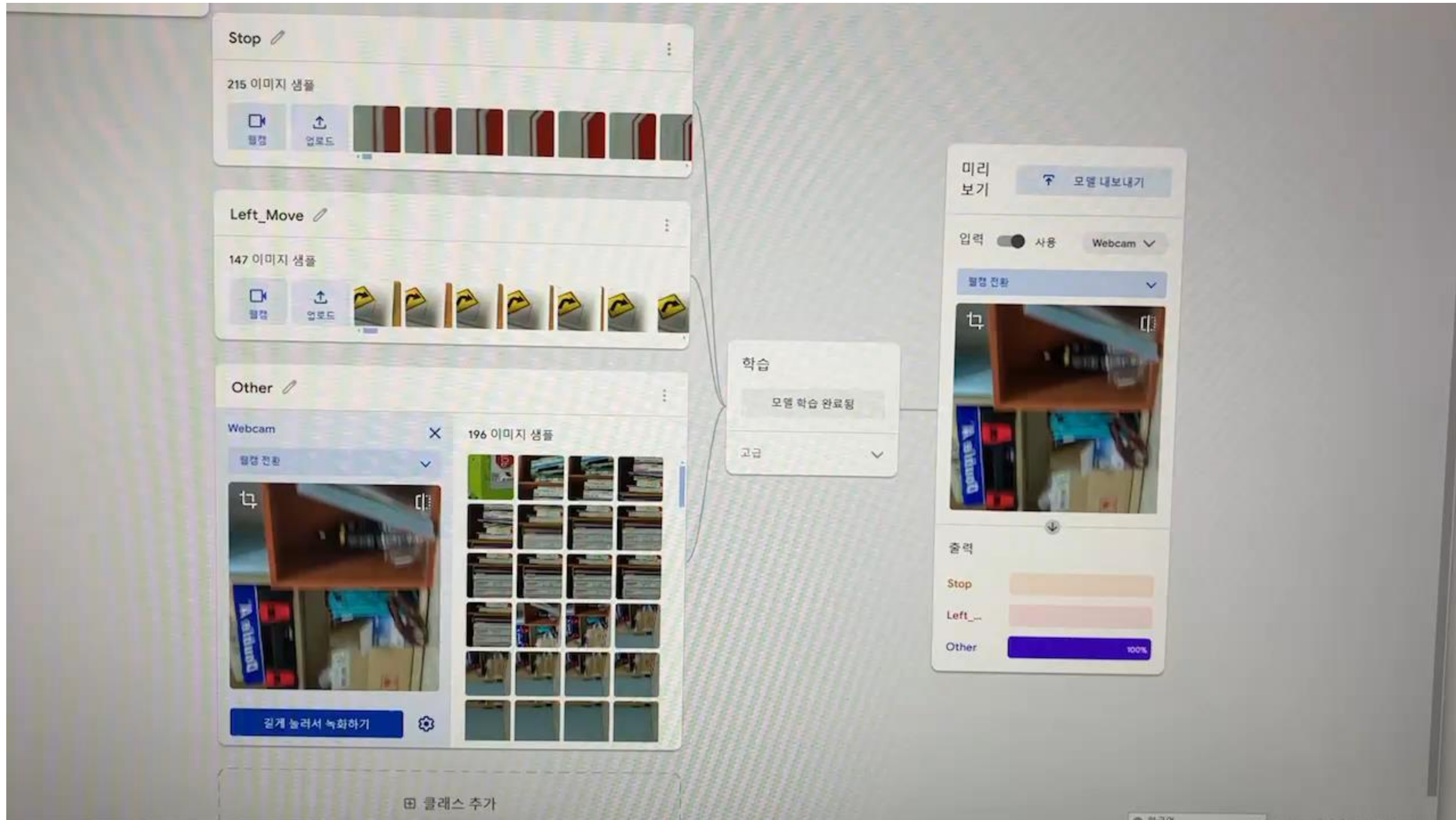
단비를 전용매트에서 라인트레이서 (자율주행)을 하면서 주변 배경을 “Other”그룹으로 생성하고 웹캠으로 찍어 추가 학습을 시켜봅니다.

The image displays two sequential screenshots of a machine learning training interface. The interface is divided into several sections: 'Stop' (215 image samples), 'Left_Move' (147 image samples), and 'Other' (196 image samples). The 'Other' group is highlighted with a red box in both screenshots, indicating the focus of the learning process. In the left screenshot, the 'Other' group is being populated with webcam samples. In the right screenshot, the 'Other' group's training progress is shown as 100% complete. A '학습' (Learning) status box is also visible, indicating '모델 학습 완료됨' (Model training completed).

실습 : 메카넘 휠 로봇 "단비"에 카메라를 장착하고, 조종하거나 라인트레이서 매트에 올려놓고 표지판을 인식하는지 확인해 보세요!







메카넘휠 AI 로봇 "단비"

Mecanum Wheel Robot AI DanVI

THANK YOU

궁금한 부분이 있으시면 아래의 메일로 연락주세요.

E-mail : metaplace@naver.com

Tel : 033-252-4787

