

# 원격환경모니터링 스터디 & 교육

학부모 AI-maker 연구회  
2020 우리동네 과학클럽

1. 본 자료의 내용은 학부모 AI-maker 연구회에서  
2020년도 과학기술진흥기금의 재원으로 한국과학창의재단의 지원을 받아 수행된 성과물입니다.

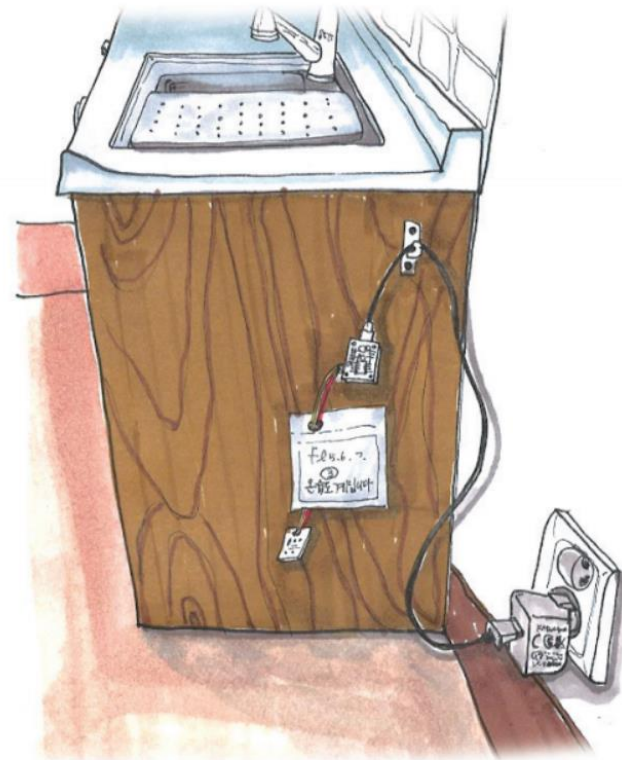
2. 본 자료를 대외적으로 공개하거나 사용할 때는  
반드시 대전마케팅공사(대전과학문화지역거점센터)와 사전에 상의하여야 합니다.

# 원격모니터링 시스템

온습도계 만들기

# 원격모니터링 시스템 스토리

- 펜션을 운영하면서, 손님들의 체감온도와 난방관리의 문제 발생
- 어떻게 하면, 각 방의 온도를 파악하고 합리적으로 대응을 할 것인가?
- 메이커의 생각: 각 방의 온도와 습도를 알 수 있는 원격모니터링 도구를 만들자!
- 원격모니터링 온습도계의 제작 및 설치
  - ESP8266+온습도계 모듈
  - 아두이노 코딩
  - Thinkspeak 에 연결하여 스마트폰앱으로 모니터링





# 제작방법

# 보드 연결하기

- 보드를 연결할 수 있도록 드라이버를 잡아야 한다
- 드라이버 다운로드 주소
- <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>
- 자신의 PC OS에 맞는 것을 다운받는다
- 다운받고 설치한 후에 장치관리자의 포트에서 연결이 잘 되었는지 확인한다
- <https://deneb21.tistory.com/m/591> (참고 링크)

# DHT 라이브러리 설치하기

- 스케치-라이브러리포함하기-라이브러리관리
- Dht를 눌러서 검색하기
- dht 센서라이브러리 설치하기

# 코드 다운받기

- Github 링크를 눌러서 접속한다
- 맨 우측상단의 다운로드 버튼을 눌러서 다운받는다
- 다운받은 txt파일을 열고, 전체 선택 후 복사한다음, 아두이노 스케치의 새로운 창에 붙여넣는다
- 내가 있는 곳의 WiFi아이디와 패스워드를 수정하고, write API key를 수정한다음 컴파일&다운로드
- 코드: [https://gitlab.com/bdm002/esp8266/-/blob/master/50\\_thingspeak\\_f1\\_f2](https://gitlab.com/bdm002/esp8266/-/blob/master/50_thingspeak_f1_f2)



sketch\_apr16a

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

// char *ssid = "iptime";
// char *password = "123456789a";

char *ssid = "KT_GiGA_2G_EB15";
char *password = "8fedlii152";

// char *ssid = "olleh_WiFi_D25A";
// char *password = "0000000433";

// char *ssid = "iptime_ring";
// char *password = "";

// char *ssid = "velopcaffe";
// char *password = "123456789a";

// char *ssid = "ifindhim";
// char *password = "123456789a";
```

```
at java.util.ArrayList.sort(ArrayList.java:1464)
at java.util.Collections.sort(Collections.java:175)
at cc.arduino.contributions.libraries.ui.LibrariesIndexTableModel.updateContributions(LibrariesIndexTableModel.java:208)
```



# Thingspeak.com

- thingspeak.com 에 가입해야 언제 어디서나 모니터링이 가능함
- 가입 후에 new channel 명령으로 채널을 만들고, 세팅을 해야함
- 어떤 피쳐를 모니터링 할 것인지 결정해야 함
- API key 중에 아두이노스케치의 코드에 사용할 것은 write key
- User key는 profile에 있음

# My Profile

## MathWorks Account Settings

MathWorks Account Email	shhan67@gmail.com
User ID	mwa0000018155427
Password	*****

[Edit MathWorks Account Settings](#)[Edit MathWorks Community Information](#)

## ThingSpeak Settings

Time Zone	Seoul	<a href="#">Edit</a>
User API Key	T5F9IKYTI80RSL0M	<a href="#">Refresh</a>
MQTT API Key	<no API key>	<a href="#">Refresh</a>
Alerts API Key	<no API key>	<a href="#">Refresh</a>

## Help

### MathWorks Account Settings:

- MathWorks Account email and password are needed to sign in to ThingSpeak.
- User ID is displayed as the author of your public channels.
- If you edit your MathWorks Account settings, you will need to sign out of ThingSpeak and log back in.

### ThingSpeak Settings:

- Time Zone is used when displaying data in your charts, and when scheduling your ThingSpeak apps.
- User API key is required to create and manage channels using the [REST API](#).
- MQTT API key is required to subscribe to channel topics using the [MQTT API](#).

## API Requests

### Get Channel List

```
GET https://api.thingspeak.com/channels.json?api_key=T5F9IKYTI80RSL0M
```

### Create a Channel

```
POST https://api.thingspeak.com/channels.json
api_key=T5F9IKYTI80RSL0M
name=My New Channel
```

### Clear a Channel Feed

```
DELETE https://api.thingspeak.com/channels/CHANNEL_ID/feeds.json
api_key=T5F9IKYTI80RSL0M
```

### Delete a Channel



# 스마트폰 세팅

- 스마트폰의 구글플레이에서 thingview를 검색해서 설치한다
- Thingview를 실행시키고, 우측 하단의 + 표시를 클릭해서, Add all my channels 를 선택
- thingspeak 사이트의 User API key를 입력하고 search버튼을 클릭하면, thingspeak에서 만든 채널을 볼 수 있게 된다

### 2층 온도와 습도

<https://thingspeak.com/channels/1040001>

Server url  
  
 User API key ?

Add all my channels

Add channel

### 2층 온도와 습도

<https://thingspeak.com/channels/1040001>



친환경공기정화기

# 그린 월(GreenWall)의 필요성

## 1. 공기청정기가 잡을 수 없는 실내 오염물질을 제거

밀폐된 공간에서 공기청정기 지속 가동 시 실내 이산화탄소 농도가 높아지기 때문에 일정 기간 환기가 필요



“반려식물”



### 실내식물의 공기정화 원리

- 앞에서 흡수
- 음이온방출, 증산작용
- 근원부 미생물에 의해 제거 (90% 정도)
- VOC: 55%~85% 제거

참조: 국립원예특작과학원

## 2. 실내공기질을 관리외에 추가가치 제공

실내가든, 실내가습기, 에너지절약, 감성건강, 인테리어 등

# 실내공기정화기의 필요성



환경과학기술연구소(KSD)에서 실시한 실험

# 건강을 위협하는 실내오염 물질 및 발생원인

오염물질	주요 발생원	오염물질	주요 발생원
부유미생물 (곰팡이, 세균)	가습기, 냉방장치, 냉장고, 애완동물(비듬, 털), 인간활동(대화, 재채기 등), 음식물쓰레기, 카펫	휘발성유기화합물, 탄화수소류, 미세먼지, 타르, 니코틴	담배연기
폼알데히드	각종합판, 보드, 가구, 단열재, 담배연기, 화장품, 의류	벤젠	건축재료, 세탁용제, 페인트, 살충제, 석유화학제품, 자동차 배출가스, 연료(석유 등)
아세트 알데히드	합성수지, 접착제, 향료	톨루엔	담배연기, 건축재료, 페인트, 살충제, 난방(석탄, 석유연소)
아세톤	칩보드, 건축재료, 접착제, 락카, 매니큐어 제거제	에틸벤젠	자동차배출가스, 담배연기
연고가스 (CO, NO <sub>2</sub> , SO <sub>2</sub> 등)	난로, 연료연소, 가스레인지	자일렌	접착제, 페인트
먼지, 중금속	외기유입, 생활활동, 의류, 흡연, 연소기구 등	스티렌	접착제, 주방랩, 플라스틱제품, 필름
라돈	토양, 건축자재, 지하수	테트라클로로 에틸렌	카펫용 세제, 얼룩제거제, 드라이크리닝 용제

출처: 환경부·국립환경과학원, 주택 실내공기질 관리를 위한 매뉴얼

# 실내식물 공기정화의 원리

1. 잎에 흡수된 오염물질은 대사산물로 이용되어 제거되고, 일부는 뿌리로 이동되어 토양내 근권부 미생물의 영양원으로 활용되어 제거한다.
2. 음이온, 향, 산소 등의 방출물질에 의해 환경이 정화되며, 증산작용에 의해 공중습도가 올라 가고, 주변 온도를 조절한다.
3. 미생물은 유기물을 분해하여 식물 영양원으로 제공하고, 뿌리 유출물은 (광합성산물의 최대45%) 미생물의 영양원이 되어 상호공생의 역할을 한다. 실내 공기 중의 VOC가 근권부 미생물에 의해 제거된다.
4. 증산작용에 의해 화분 토양내의 부압이 형성되어 오염된 공기가 이동하면 근권부 미생물과 토양 흡착 등에 의해 제거 된다

# 친환경실내공기청정기 개발의 필요성

- 식물바이오피ilter는 실험 및 경험에 의한 제품으로 환경부 기준을 만족하는 공기질을 지속적으로 제공하지 못하므로 풍속, 수분공급 등의 지속적 최적화·지능화가 필요함
- 지속적 최적화·지능화는 AI+IoT를 활용해야 가능
- 현재 국산제품은 없으며, NAAVA(핀란드벤처)제품이 유일하며 AI 부분은 비공개상태임
- 현재 국내 업체가 기술개발 중임

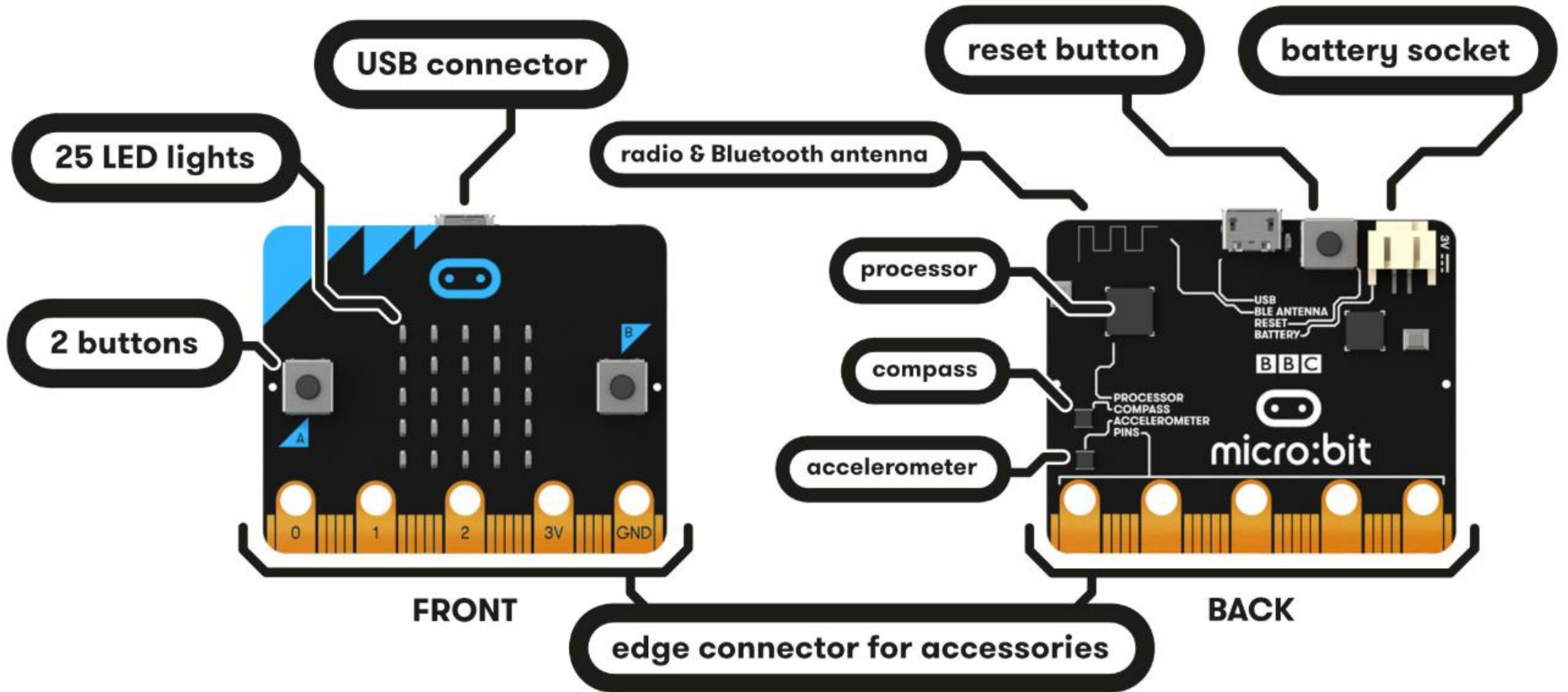


마이크로비트 이해



마이크로비트란?

# 마이크로비트의 구조



# 마이크로비트의 특징

- 따로 따로 제어할 수 있는 25개의 LED 불빛들
- 프로그래밍 가능한 2개의 버튼
- 하드웨어 확장 가능 핀
- 빛센서와 온도 센서들
- 움직임 센서들(가속도 센서와 나침반(자기) 센서)
- 라디오(Radio)와 블루투스(Bluetooth)를 이용한 무선 통신 기능
- USB 인터페이스



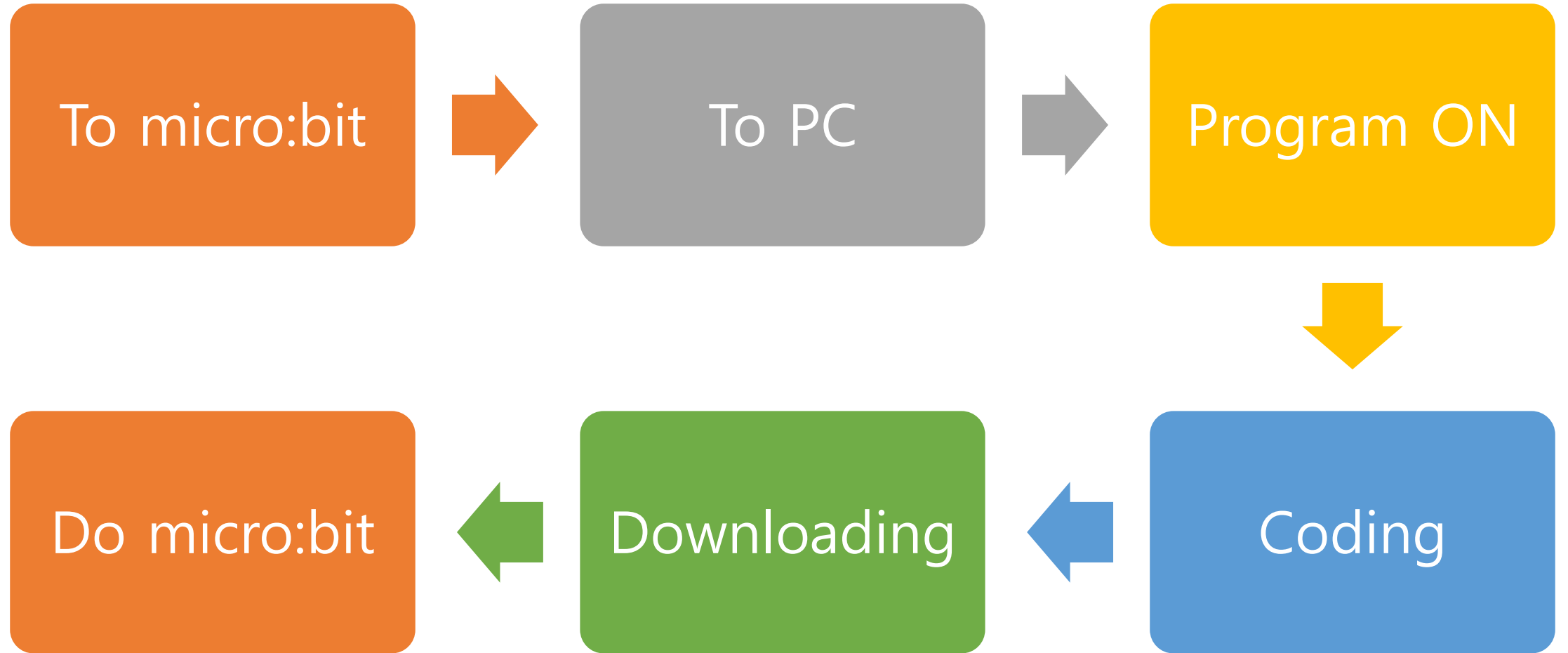
다운로드하기



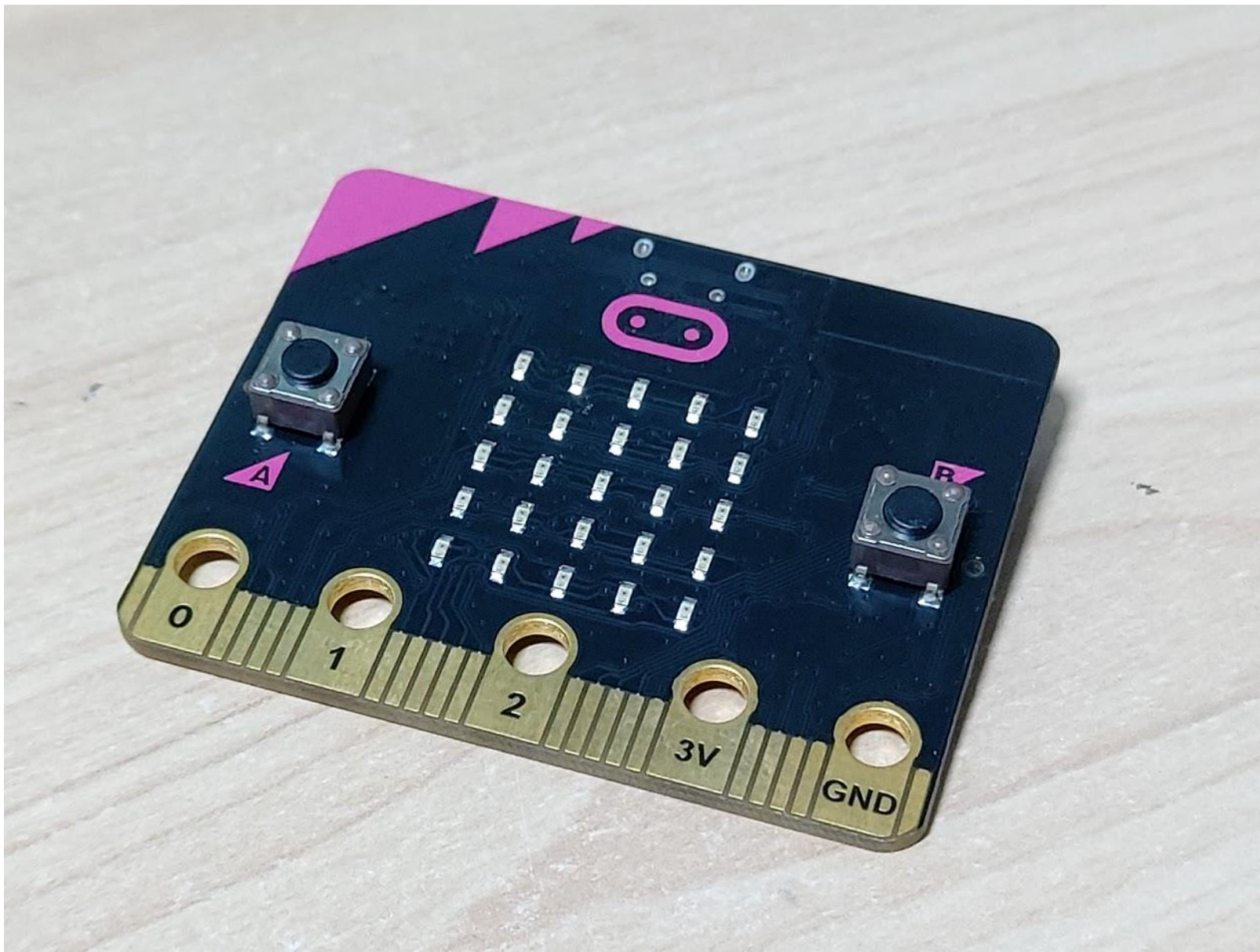
컴퓨터에 연결

# 마이크로비트 사용과정

# 마이크로비트 사용 과정

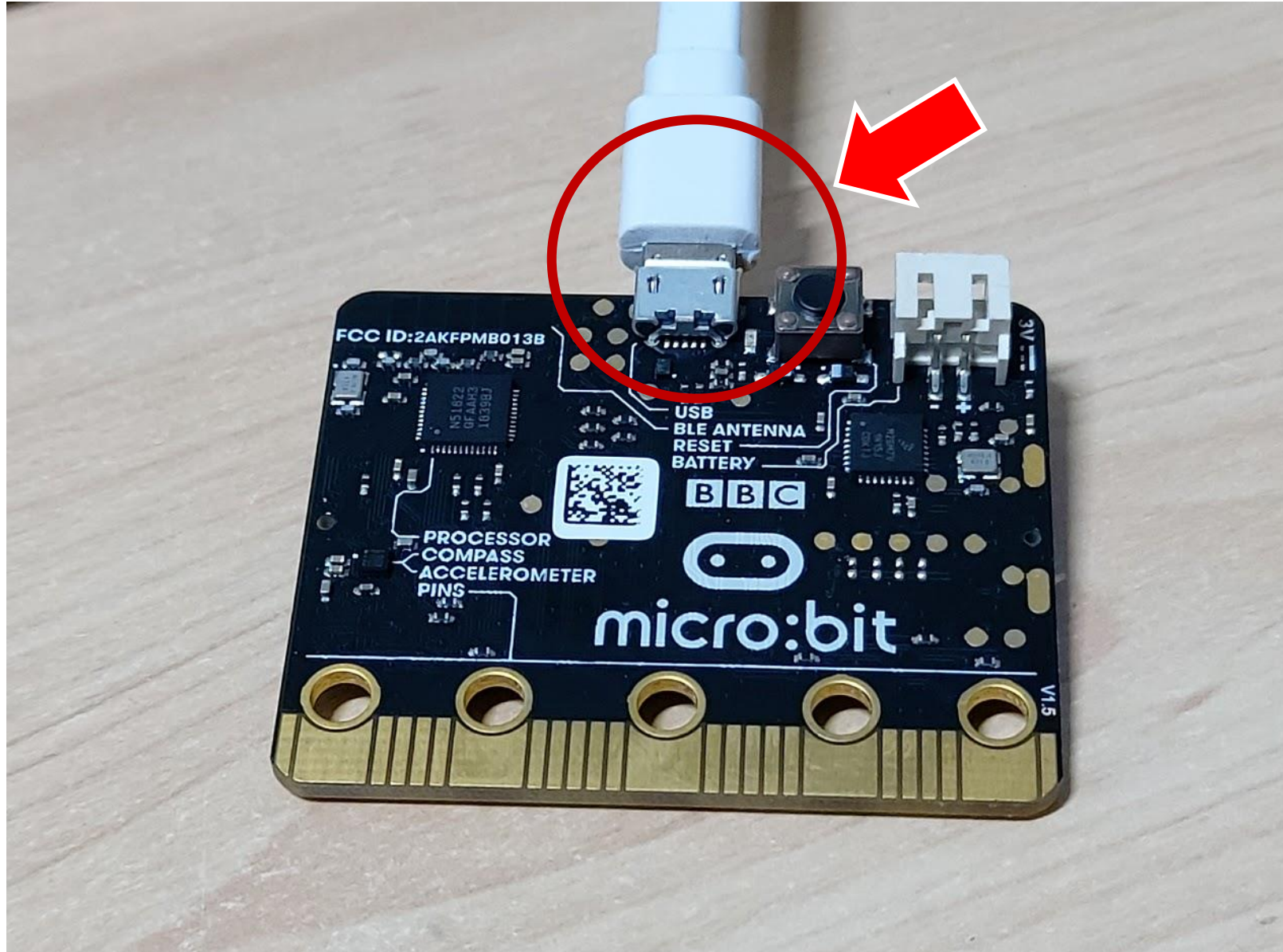


# 마이크로비트를 준비합니다



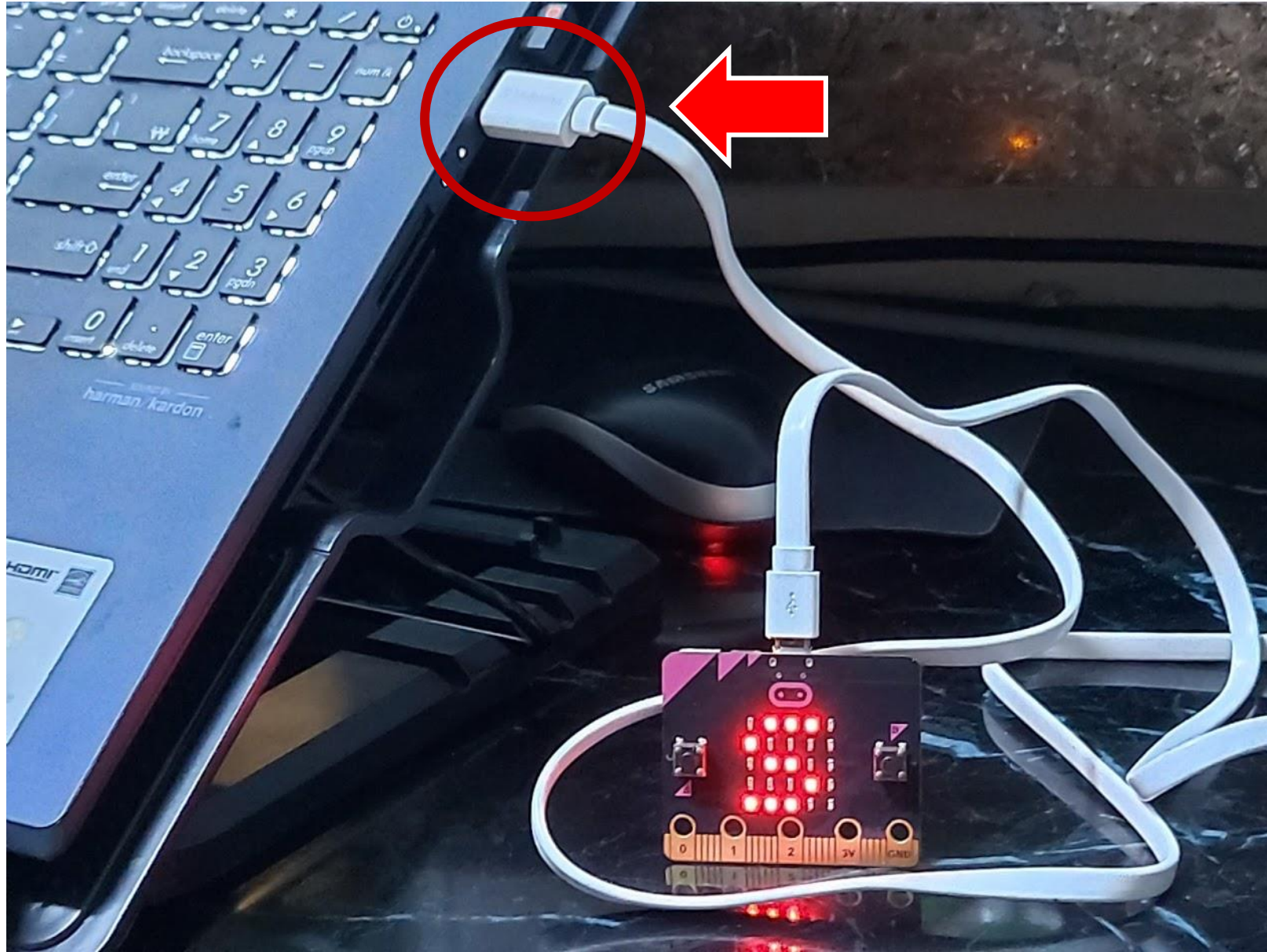


# USB 연결케이블을 꼽습니다





# 컴퓨터의 USB단자와 연결합니다



# Makecode.microbit.org 에 접속합니다

The screenshot shows the Makecode website interface. At the top, there is a navigation bar with the 'micro:bit' logo on the left, a home icon and the text '처음 화면' (Home Screen) in the center, and a settings gear icon and the 'Microsoft' logo on the right. Below the navigation bar is a large banner image featuring two micro:bit boards connected to a breadboard with a blue push-button. One board has a blue 'show leds' block attached to its code editor, and a pink 'on shake' block is also visible. The background of the banner is a green grid pattern.

Below the banner, there is a section titled '내 프로젝트' (My Projects) with a sub-link '모두보기' (View All). To the right of this section is a button labeled '가져 오기' (Get). The main content area displays a grid of project cards. Each card includes a purple puzzle piece icon, the project name, a progress indicator (a green bar with a white icon and a number), and the date '1 일 전' (1 day ago). The projects shown are:

- 새 프로젝트** (New Project): A purple card with a white plus sign icon.
- 나침반** (Compass): Progress 9/9.
- 수평** (Horizontal): Progress 7/7.
- 스톱워치** (Stopwatch): Progress 8/8.
- 주사위** (Dice): Progress 7/7.

At the bottom of the page, there is a section titled '따라해보기' (Follow Me).

# 새 프로젝트를 클릭합니다

The screenshot shows the Microsoft MakeCode website interface. At the top, there is a navigation bar with the 'micro:bit' logo, a home icon labeled '처음 화면', and the Microsoft logo. Below the navigation bar is a large banner image featuring a Micro:bit board connected to a blue keyboard with a 'show leds' block and a pink 'on shake' block. The main content area is titled '내 프로젝트 모두보기' (All my projects) and contains a grid of project cards. The first card is a purple square with a white plus sign and the text '새 프로젝트' (New Project), which is highlighted by a large orange arrow. To its right are four project cards, each with a puzzle piece icon, a title, and a progress indicator: '나침반' (Compass) with 9/9 progress, '수평' (Horizontal) with 7/7 progress, '스톱워치' (Stopwatch) with 8/8 progress, and '주사위' (Dice) with 7/7 progress. Below the project cards is a section titled '따라해보기' (Follow along).

# 프로젝트 이름 입력 후에, 버튼을 클릭합니다

The screenshot shows the Microsoft MakeCode website interface. A modal dialog box titled "프로젝트 만들기 😊" (Create Project 😊) is centered on the screen. The dialog contains the text "프로젝트 이름을 지정하십시오." (Specify a project name.) and a text input field with the placeholder text "시작하기" (Get started). Below the input field is a green button labeled "창조하다" (Create) with a checkmark icon. An orange arrow points to this button. The background shows the website's header with the "micro:bit" logo and "Microsoft" logo, and a main area with a grid of project cards. One card is highlighted with a purple background and a white plus sign, labeled "새 프로젝트" (New Project).

# 코딩을 하고 다운로드를 클릭합니다

마이크로 비트 용 Microsoft MakeCode

makecode.microbit.org/#editor

micro:bit 처음 화면 공유하기 블록 자바 스크립트 Microsoft

검색하기...

- 기본
- 입력
- 음악
- LED
- 라디오
- 반복
- 논리
- 변수
- 계산
- 고급

시작하면 실행

아이콘 출력

무한반복 실행

다운로드

시작하기



# 폴더를 선택하고 코드를 다운로드합니다

The image shows a web browser window with a file explorer overlay and a tutorial overlay. The file explorer is titled "다른 이름으로 저장" (Save with another name) and is located in the "MICROBIT (E:)" drive. The file name is "microbit-시작하기.hex" and the file type is "HEX 파일 (\*.hex)". An orange arrow points to the "MICROBIT (E:)" folder in the left sidebar. The tutorial overlay is titled "무한반복 실행" (Infinite loop execution) and contains two steps:

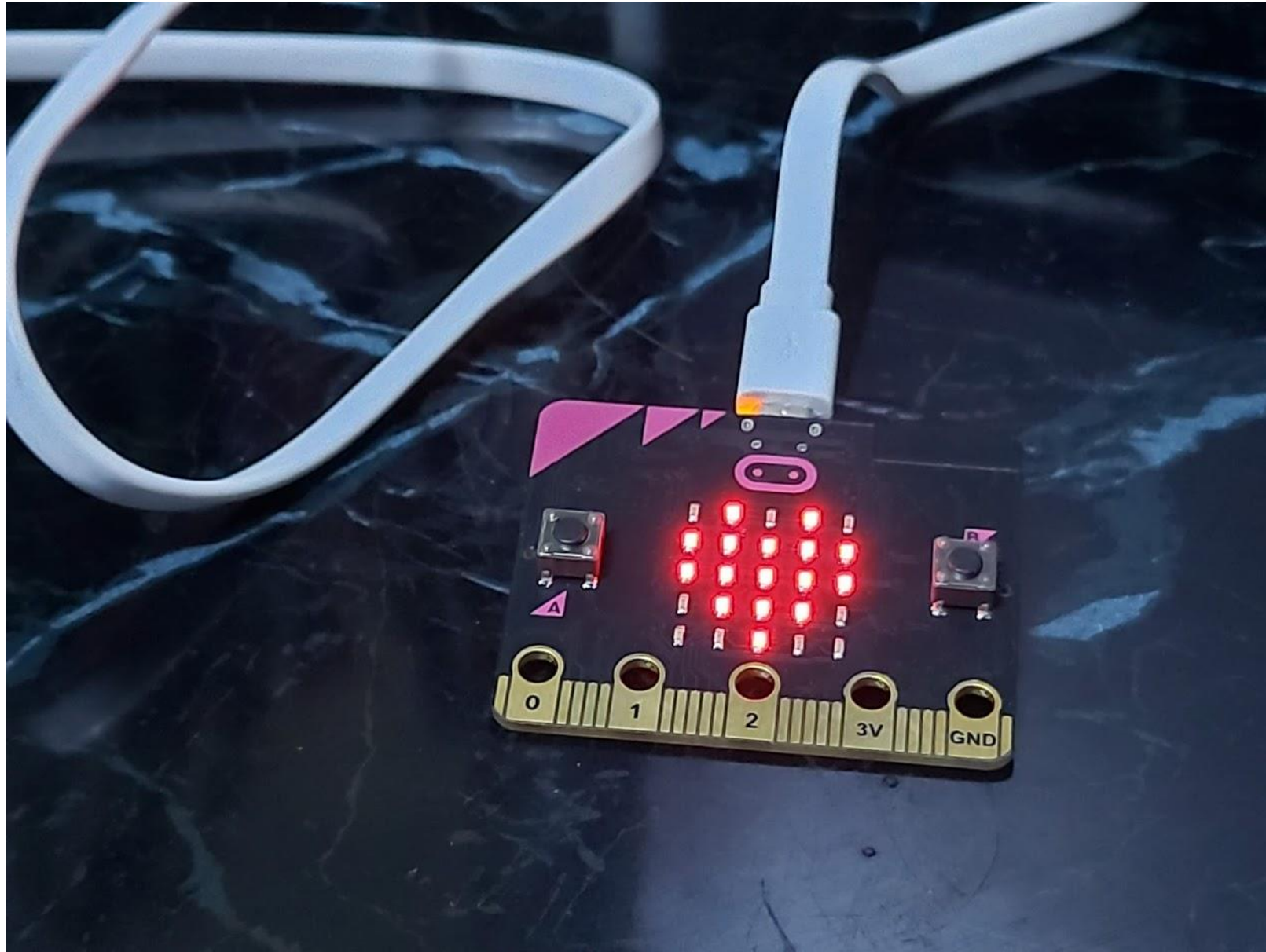
- 1 micro : bit를 USB 케이블로 컴퓨터에 연결합니다.  
micro : bit의 위쪽 부분에있는 microUSB 포트를 사용합니다.
- 2 .hex 파일을 micro : bit로 이동합니다.  
다운로드 한 .hex 파일을 MICROBIT 드라이브로 드래그 & 드롭합니다.

At the bottom of the tutorial overlay, there are buttons for "?", "장치" (Device), and "다시 다운로드" (Download again).

# 다운로드가 완료되었습니다!

The screenshot shows the Microsoft MakeCode Micro:bit editor interface. At the top, the browser address bar displays 'makecode.microbit.org/#editor'. The interface includes a top navigation bar with 'micro:bit', '처음 화면', '공유하기', '블록', and '자바 스크립트' options. On the left, a virtual Micro:bit board is shown with a grid of red LEDs. Below the board are playback controls. A central menu lists various block categories: 기본, 입력, 음악, LED, 라디오, 반복, 논리, 변수, 계산, and 고급. The main workspace on the right contains a script with two blocks: '시작하면 실행' (When green flag clicked) containing an '아이콘 출력' (Show icon) block, and '무한반복 실행' (Repeat forever loop). At the bottom left, a large orange arrow points to the '다운로드' (Download) button. The bottom status bar shows the file name 'microbit-시작하기.hex' and a '모두 표시' (Show all) button.

마이크로비트가 동작합니다







# 스마트폰에 연결하기

젠더를 활용한 유선케이블 연결

Makecode.microbit.org  
에 접속

새프로젝트 버튼 클릭

프로젝트 명 입력 후,  
create 클릭

코딩하기

KT 13:30 76%

makecode.microbit.org

Microsoft MakeCode for micro:bit  
makecode.microbit.org

make

창작소협의회  
makers.ne.kr

MakerCase - Easy Laser Cut Case Design  
makercase.com

...de.microbit.org makes makeup

1 2 3 4 5 6 7 8 9 0  
q w e r t y u i o p  
a s d f g h j k l  
↑ z x c v b n m  
!#1 영 , / \_ . , .com 이동

KT 13:30 76%

makecode.microbit.org



내 프로젝트 View All



따라해보기



라이브 코딩



Navigation icons

KT 13:30 76%

makecode.microbit.org



Create a Project

Give your project a name.

Code options

Create ✓



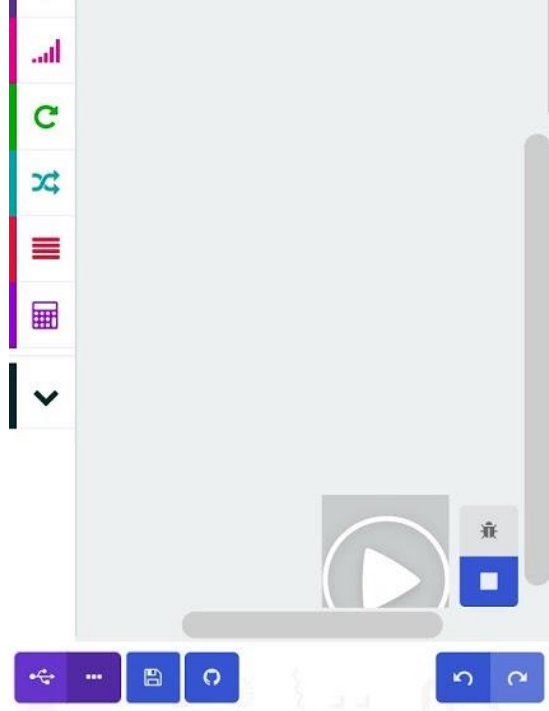
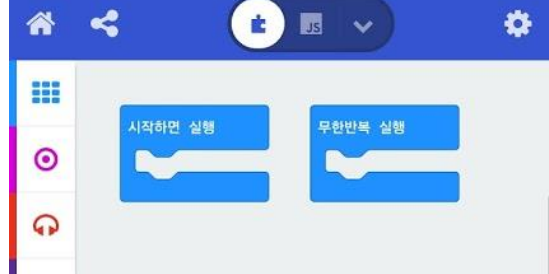
라이브 코딩



Navigation icons

KT 13:30 76%

makecode.microbit.org/#editc



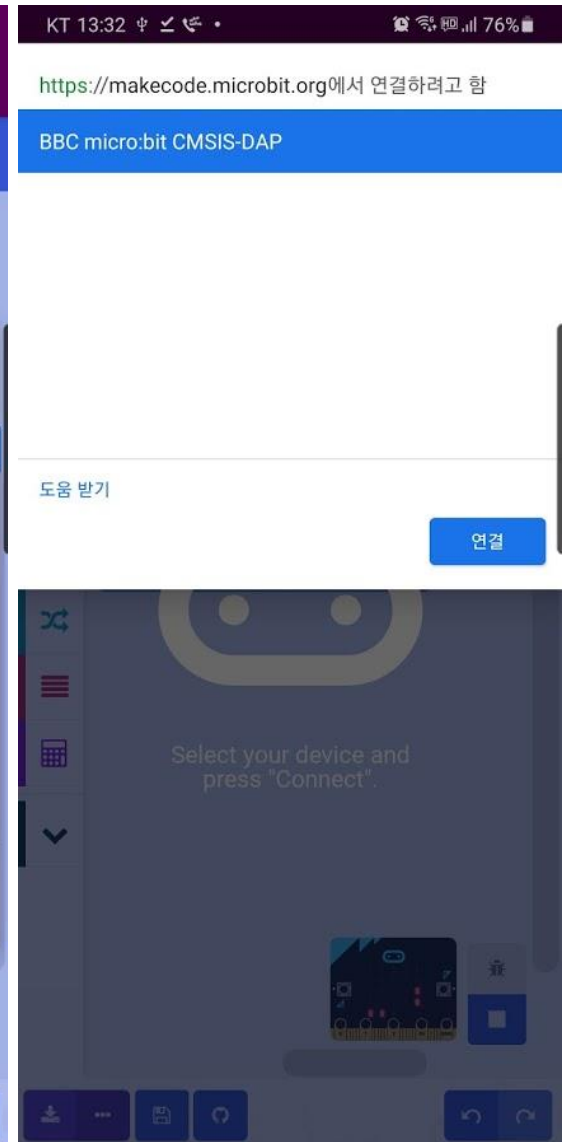
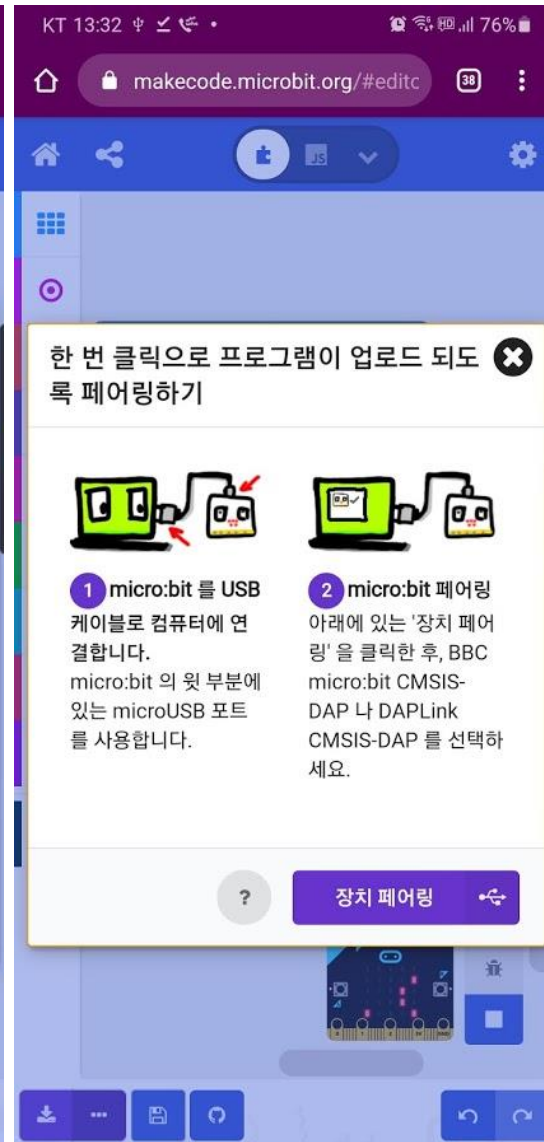
Navigation icons

문자열 추가

다운로드 클릭

장치페어링 클릭

장치선택 후 연결클릭

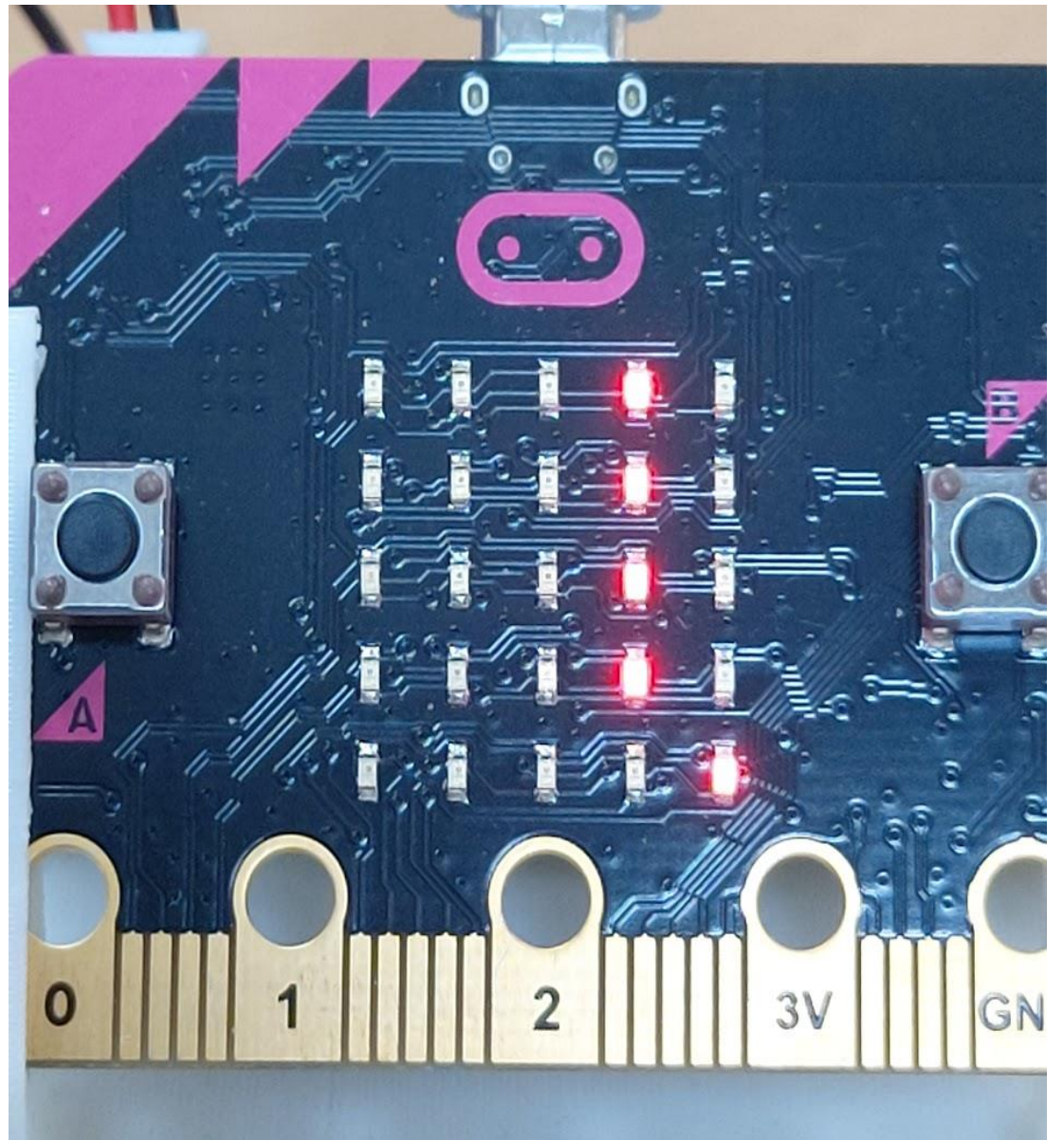
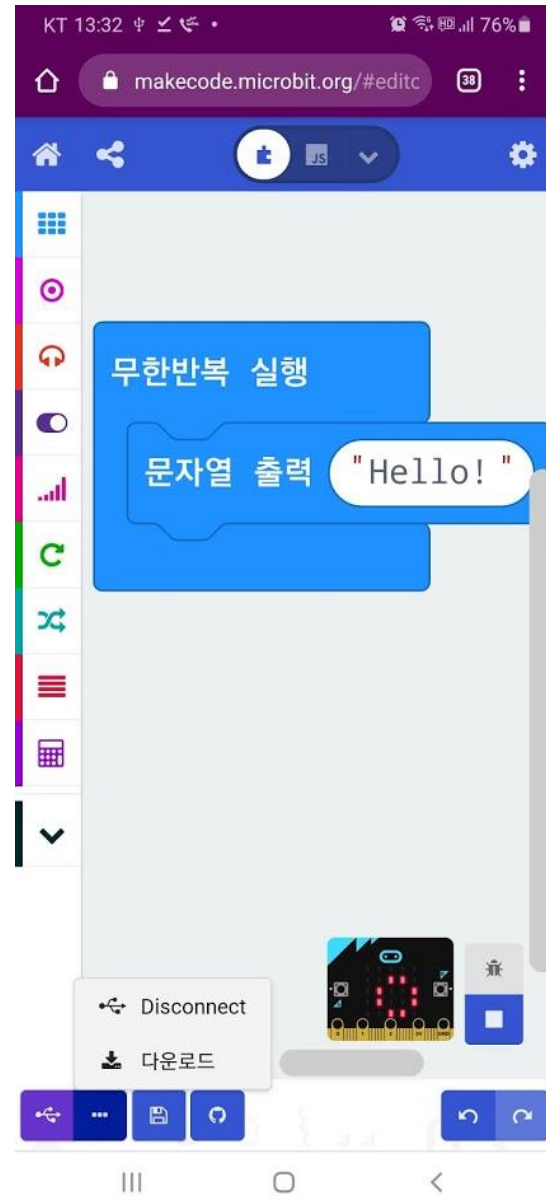




확인 클릭

다운로드 선택

다운로드 완료





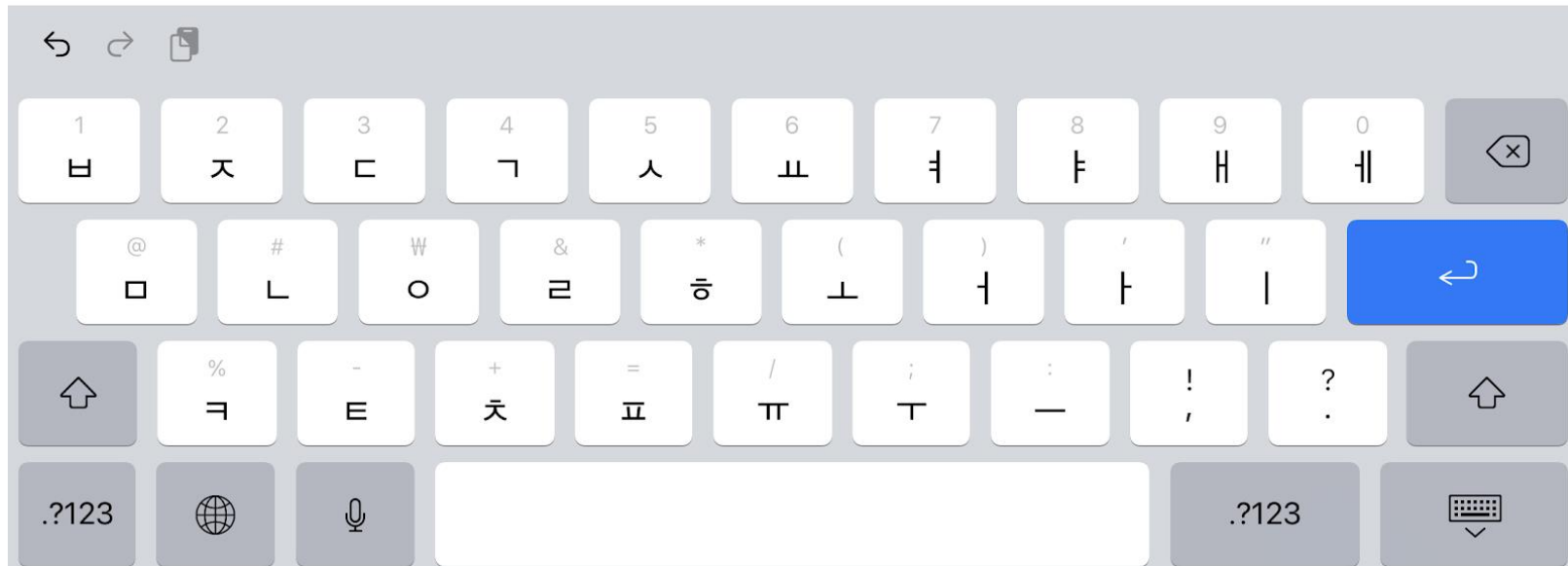
아이패드 에 연결하기



# 앱스토어 클릭



# 마이크로비트 검색



# 앱 설치하기

오후 12:43 12월 22일 화요일

49%

< 검색



**micro:bit**

Micro:bit Educational Foundation

열기

6개의 평가

2.5

★★★★☆

연령

4+

세

카테고리



교육

개발자



Micro:bit Educational Fc

언어

EN

+ 1개 언어

## 새로운 기능

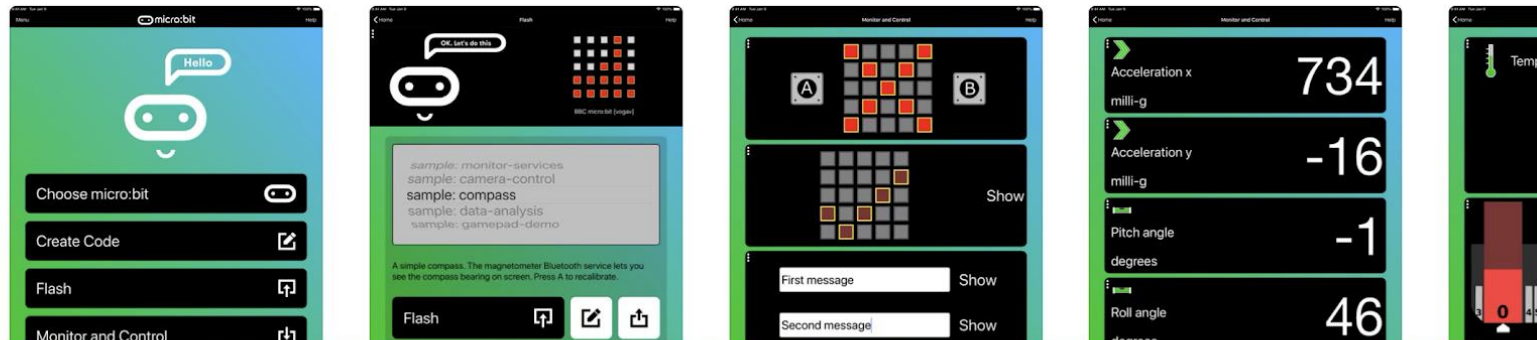
Updated support for micro:bit V2.

버전 기록

2주 전

버전 3.0.6

## 미리보기



투데이

게임

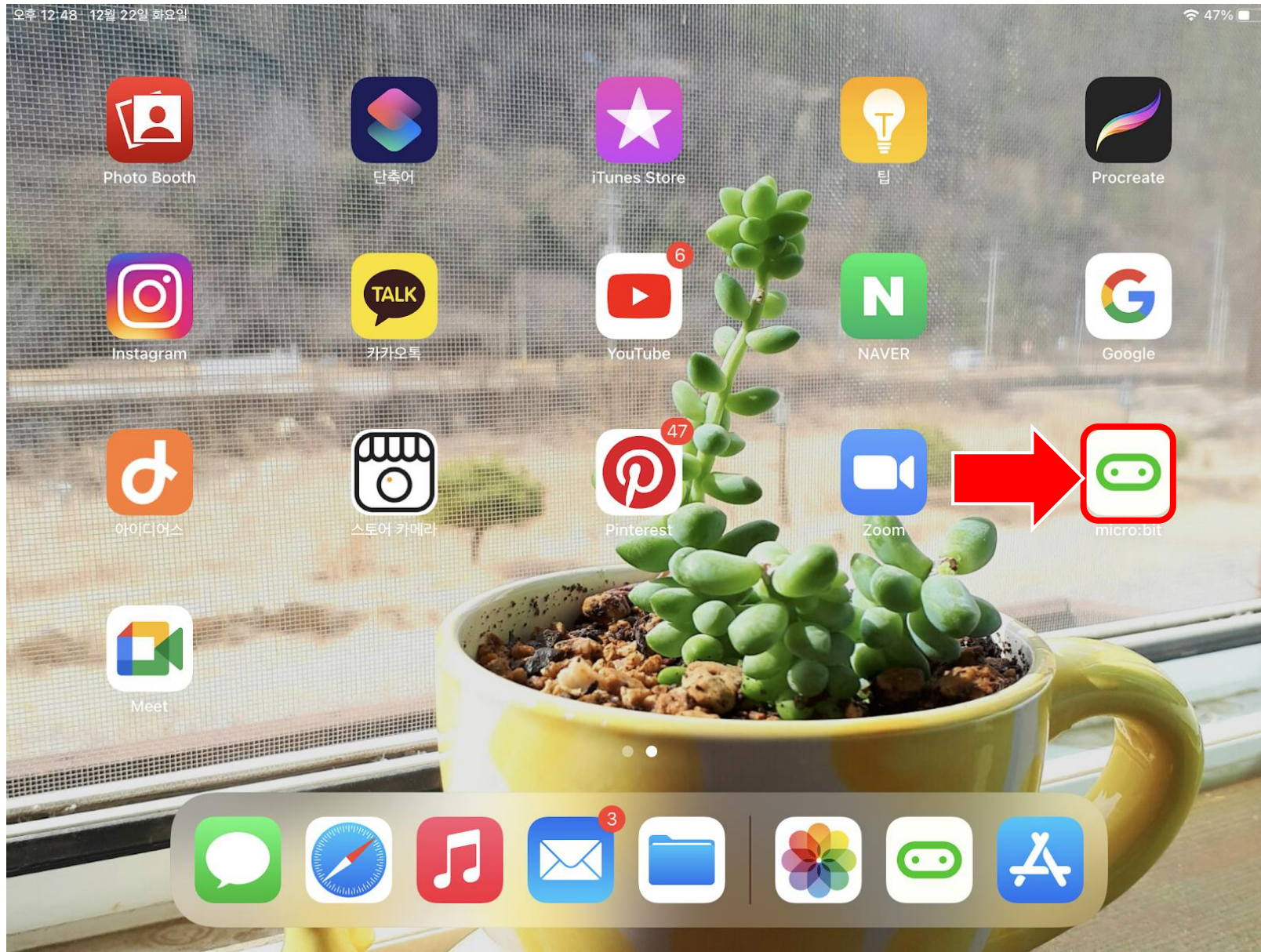
앱

아케이드

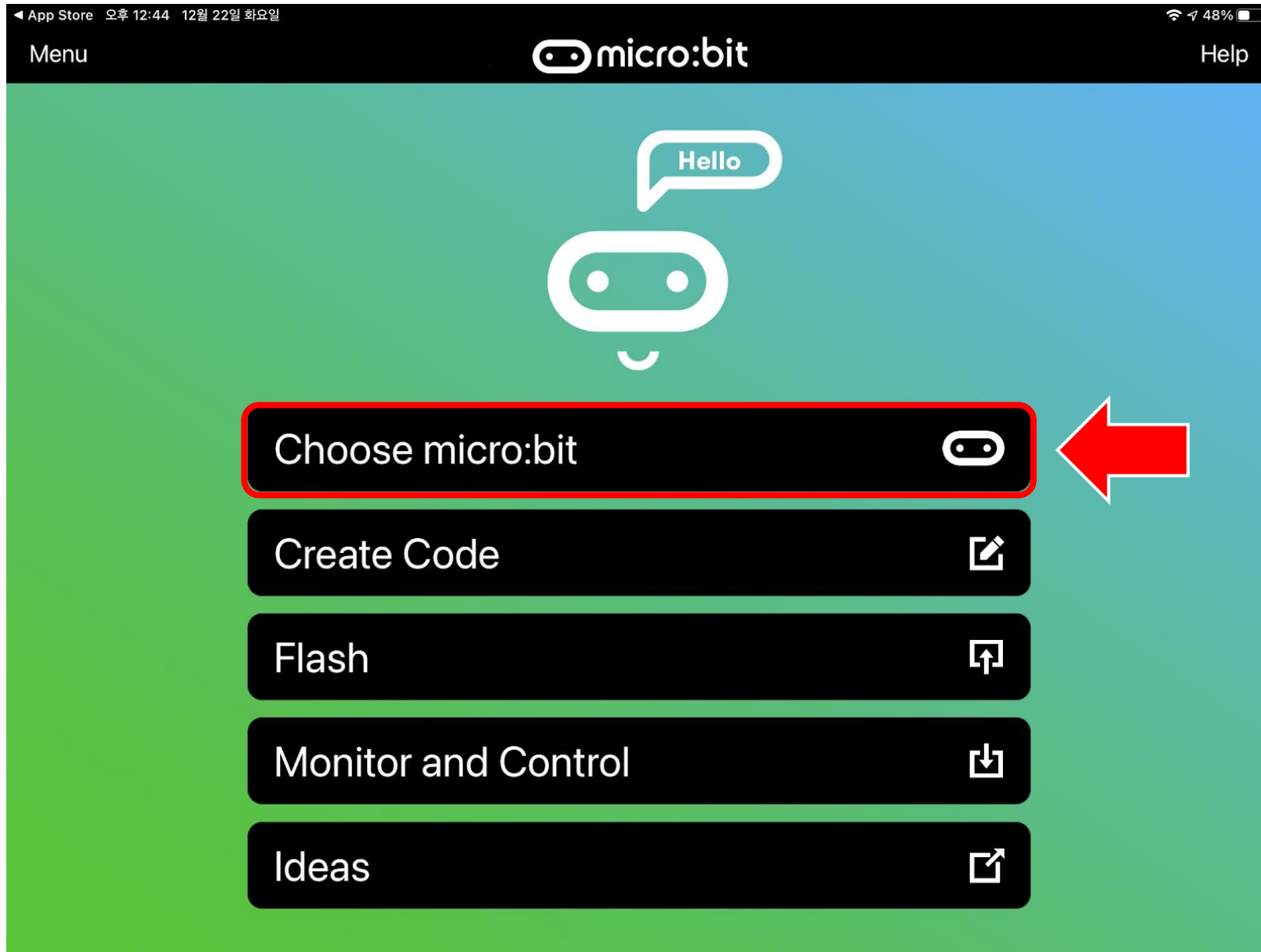
검색



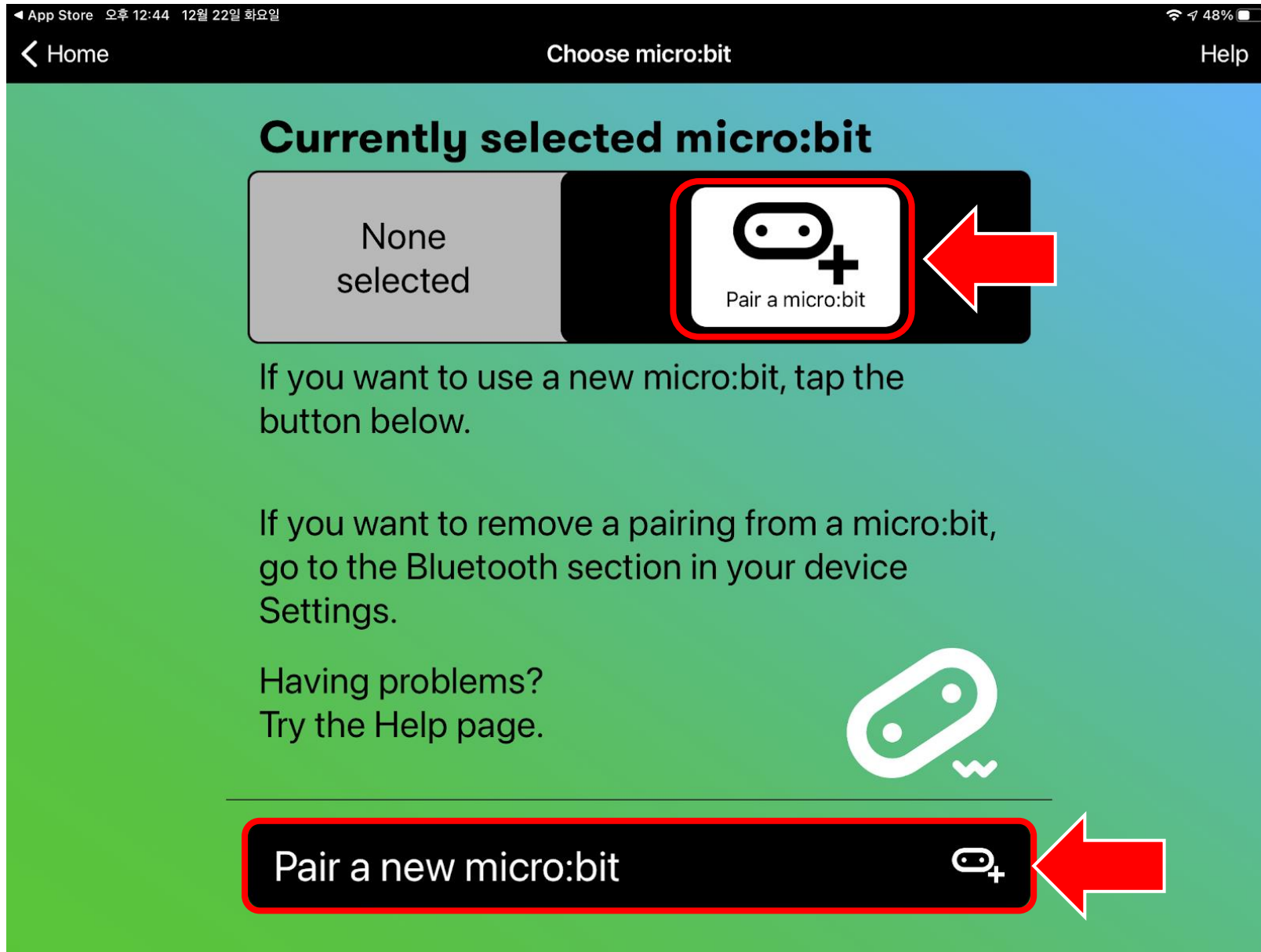
# 앱 실행하기



# 앱 열고, 마이크로비트 선택 클릭



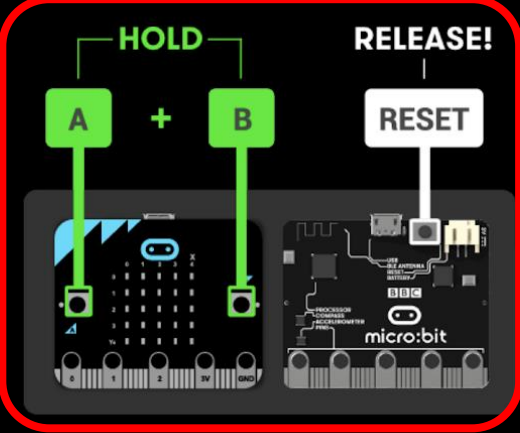
# Pair a micro:bit 클릭



# 마이크로비트의 버튼 클릭 후 Next


App Store 오후 12:44 12월 22일 화요일 48%

## Enter pairing mode

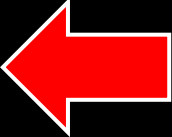


**Step 1**  
HOLD the A and B buttons and  
PRESS and RELEASE RESET

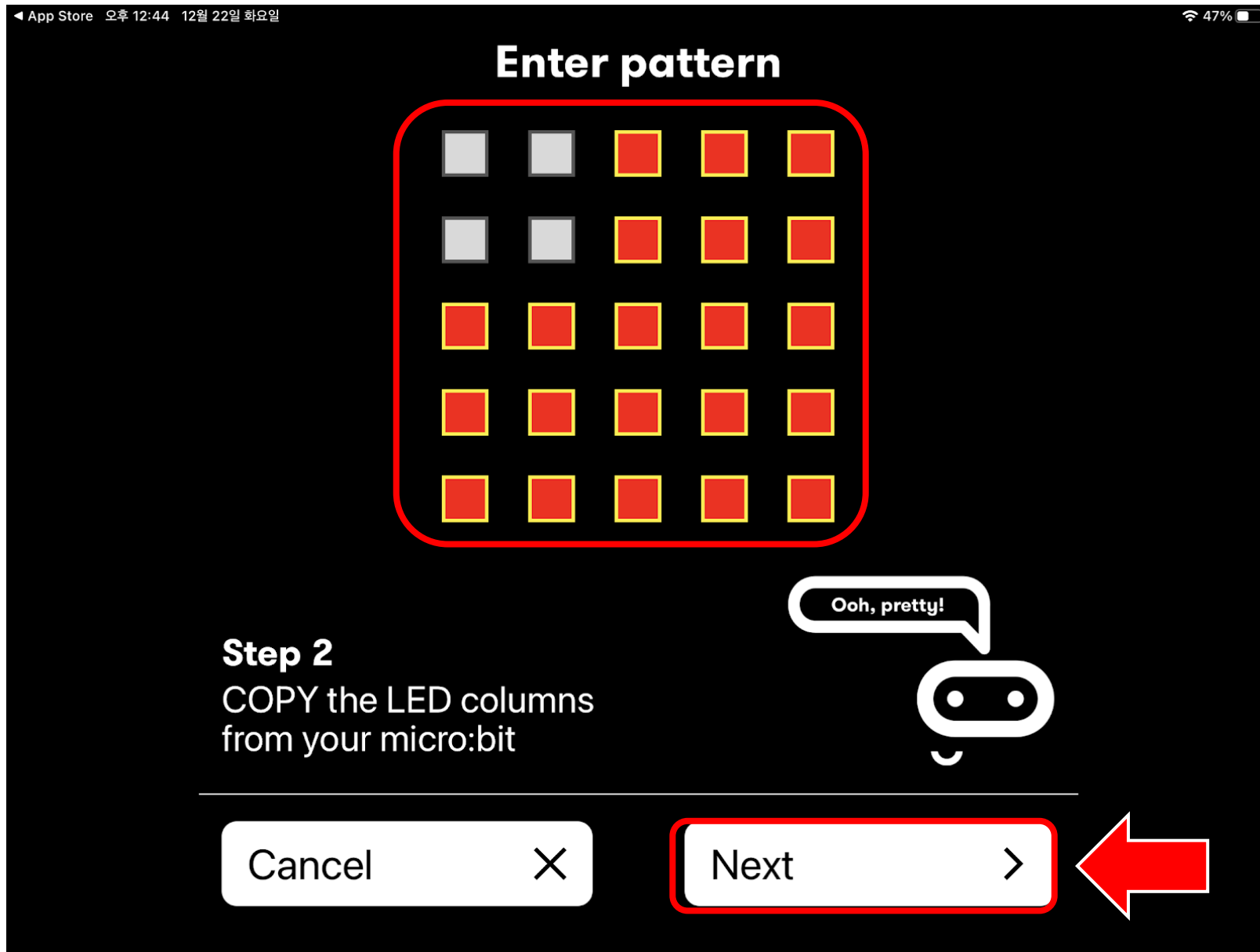
Let's do this



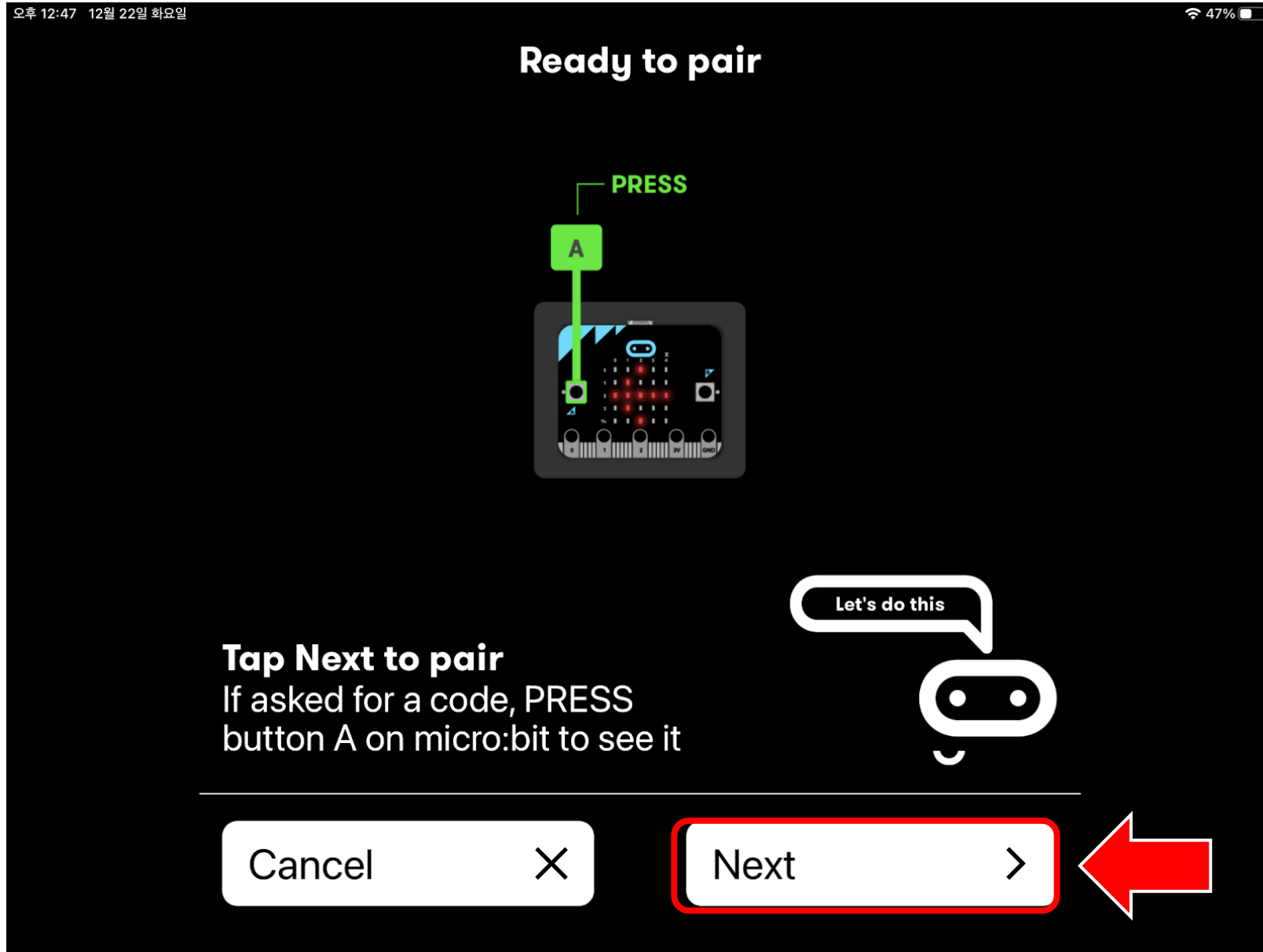
Cancel × Next >



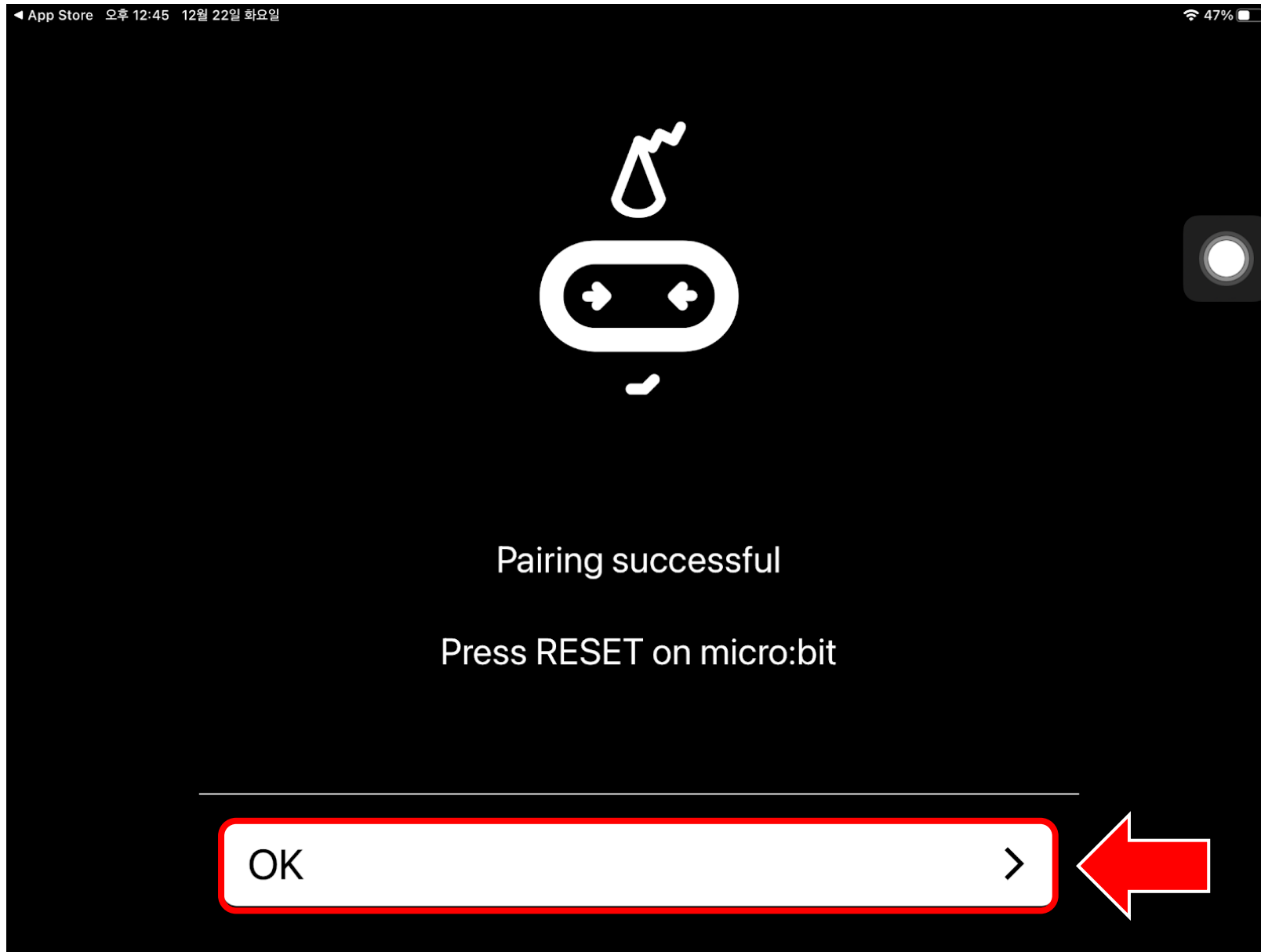
# 마이크로비트와 동일한 패턴 만들고, Next!



# 마이크로비트 A버튼 클릭하고, Next!

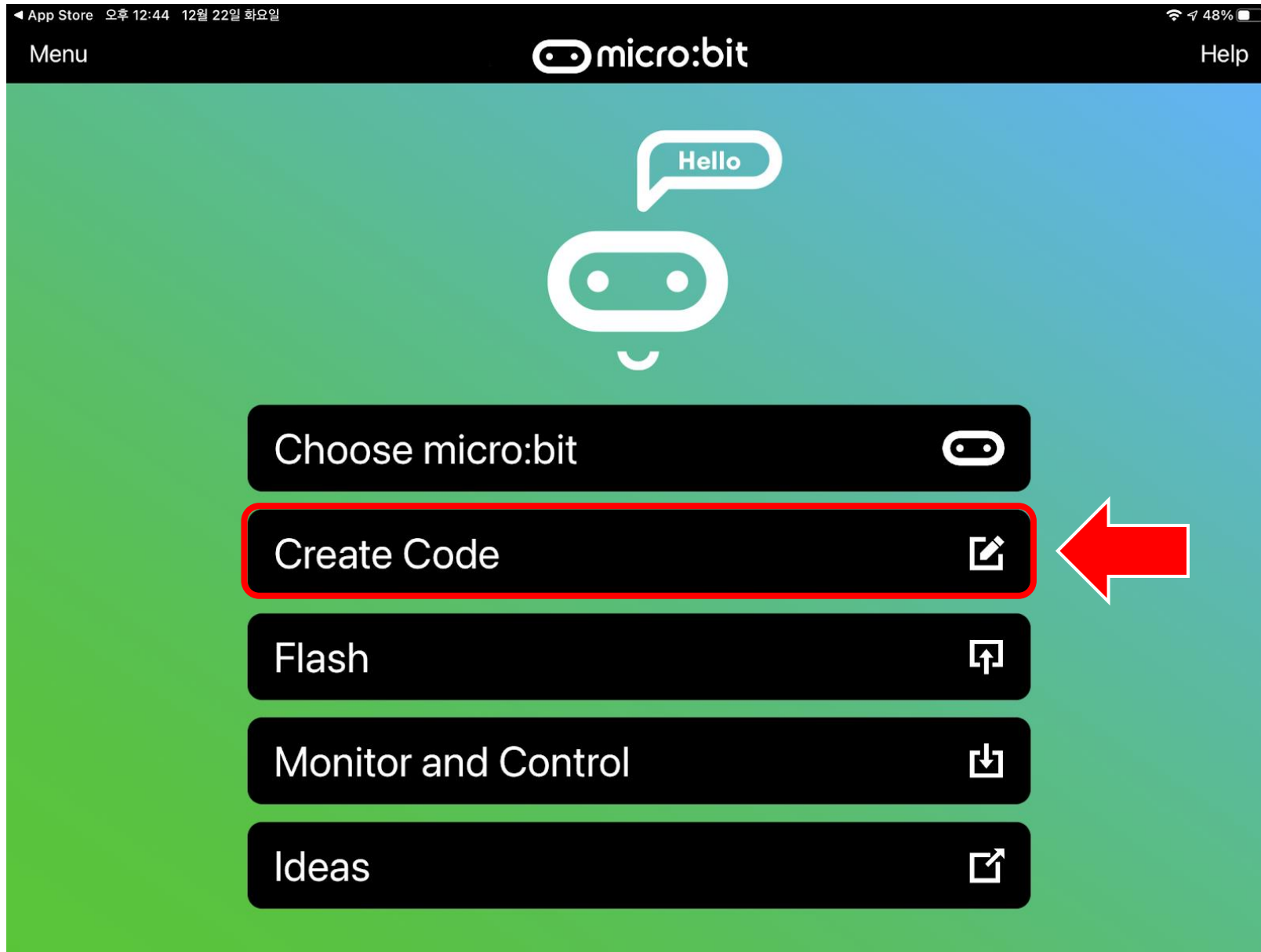


# 연결 성공~! OK 클릭하기





# Create Code 클릭





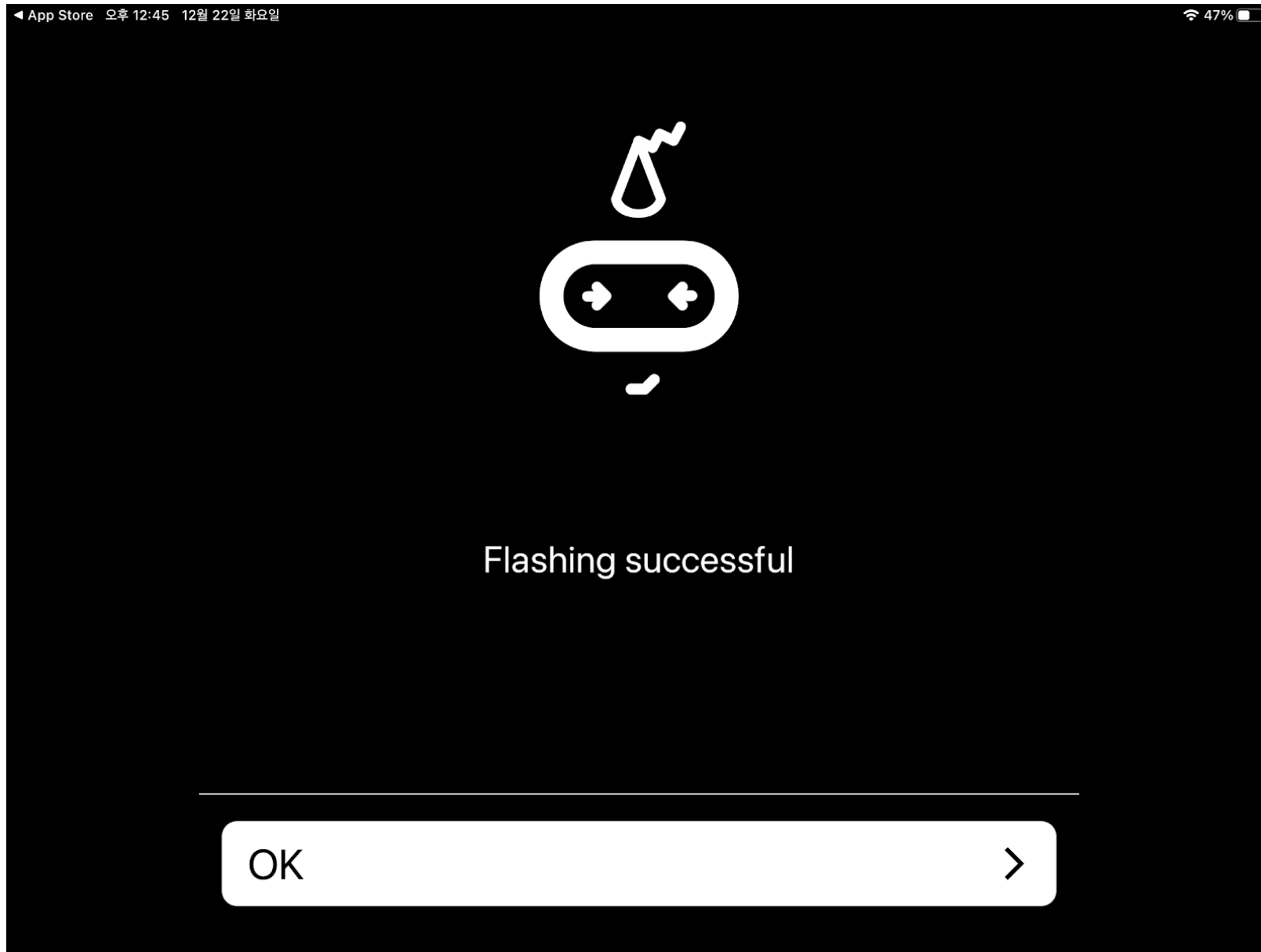
# 코딩하고 다운로드 클릭

The screenshot displays the Microsoft MakeCode editor for a micro:bit. The top navigation bar includes the 'micro:bit' logo, a home icon, a search bar, and the 'JavaScript' language selector. The left sidebar shows a micro:bit image and playback controls. The central block palette lists various categories: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The main workspace contains a '시작하면 실행' (When green flag clicked) block and a '무한반복 실행' (Repeat) block with two '아이콘 출력' (Show icon) blocks. At the bottom, a red arrow points to the '다운로드' (Download) button, which is highlighted with a red box. Other bottom controls include a name field (currently empty), a save icon, a refresh icon, and a play/pause icon.

# 다운로딩



# 다운로딩 성공!!!





# Makecode 사용법

# 실행 블록 소개하기



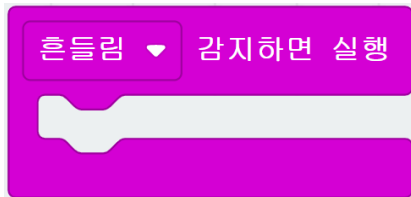
시작할 때만 실행  
명령을 한번만 실행함



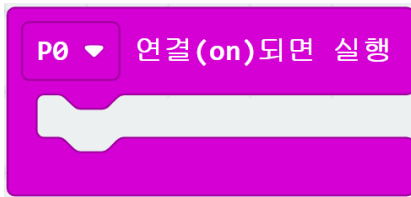
무한반복 실행  
계속 명령을 수행



A, B, A+B 버튼을 클릭하면 명령을 실행



자이로센서와 연관이 있으며, 흔들거나, 뒤집는 등의 액션이 있을 때 명령을 실행함



핀이 연결되었을 경우에 명령을 실행함. 외부센서와 연결할 때 사용 됨

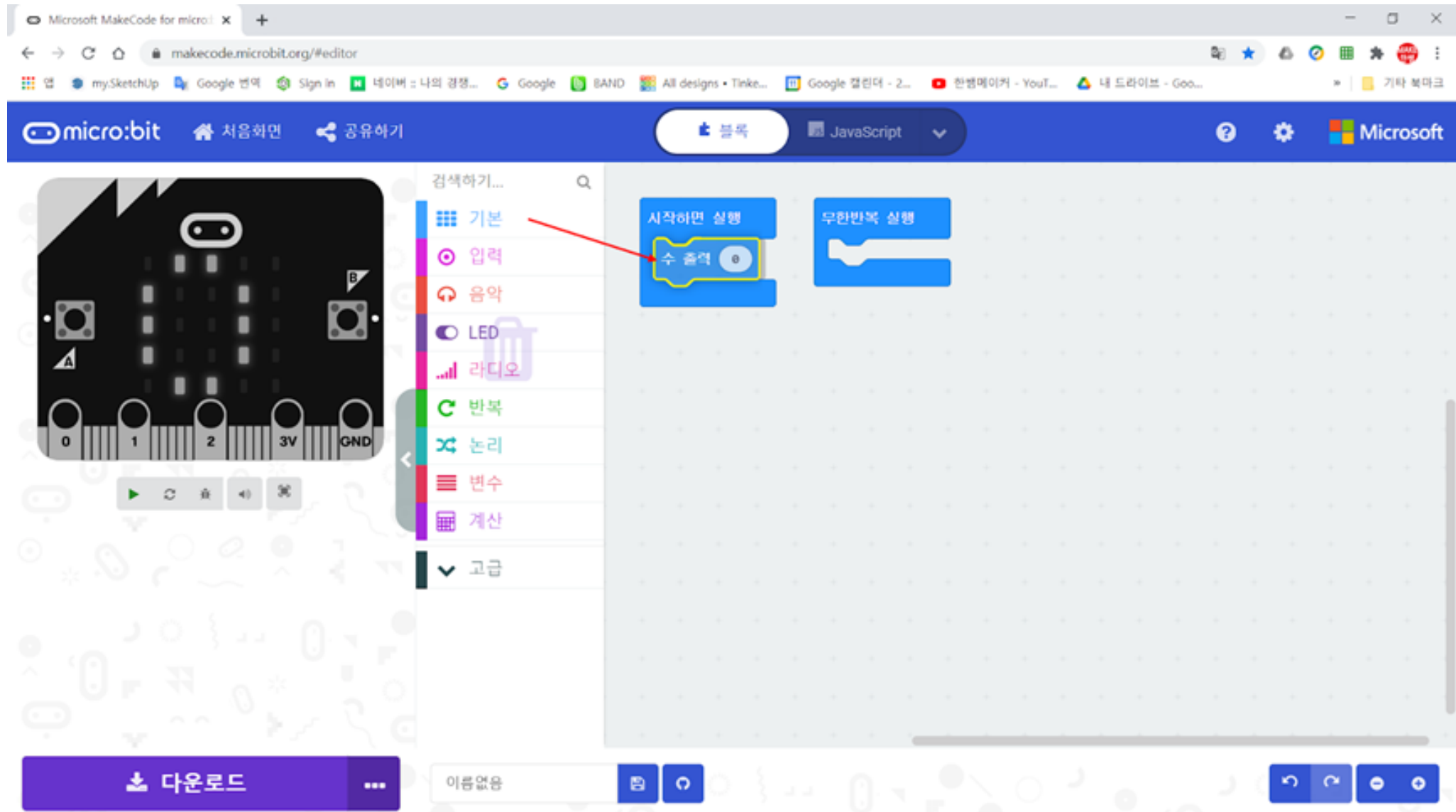


수 출력하기

# 기본 블록을 엽니다

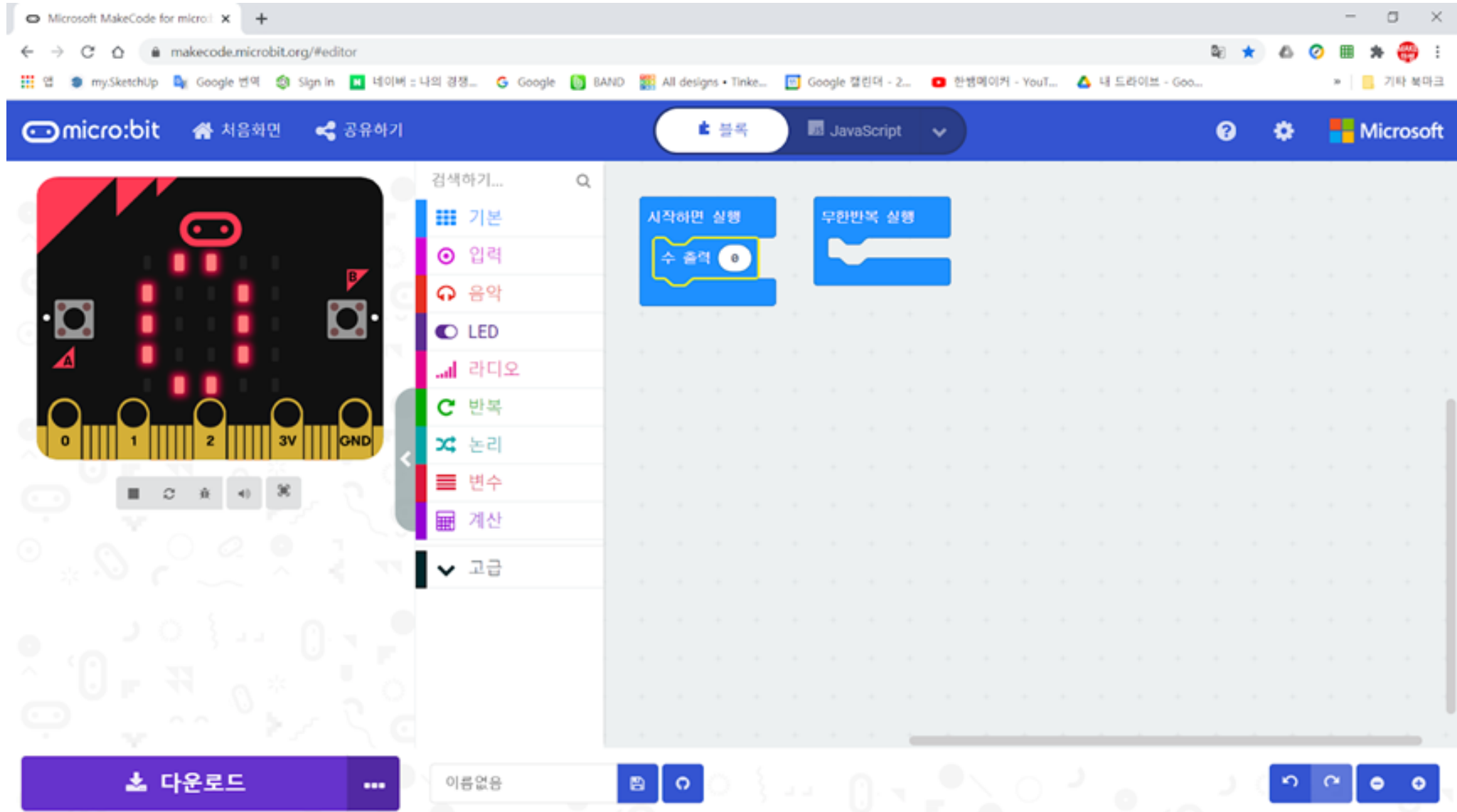
The image shows the Microsoft MakeCode editor for the micro:bit. The interface is in Korean. At the top, there's a browser window with the URL `makecode.microbit.org/#editor`. Below the browser, the MakeCode header features the 'micro:bit' logo, a '처음화면' (Home) button, a '공유하기' (Share) button, and a '블록' (Blocks) dropdown menu currently set to 'JavaScript'. The main workspace is divided into three sections: a micro:bit hardware image on the left, a block palette in the center, and a workspace on the right. The block palette has a search bar and a list of categories: '기본' (Basic), '입력' (Input), '음악' (Music), 'LED', '라디오' (Radio), '반복' (Loops), '논리' (Logic), '변수' (Variables), '계산' (Math), and '고급' (Advanced). The '기본' category is highlighted with a red box. The workspace on the right contains several blue blocks: '수 출력' (Number output) with '0', 'LED 출력' (LED output) with a 4x4 grid, '아이콘 출력' (Icon output) with a grid icon, '문자열 출력' (Text output) with 'Hello!', 'LED 스크린 지우기' (Clear LED screen), and '무한반복 실행' (Infinite loop). At the bottom, there's a purple '다운로드' (Download) button, a file name field containing '이름없음' (No name), and a '실행' (Run) button.

# 수 출력블록을 시작하면 실행에 삽입합니다

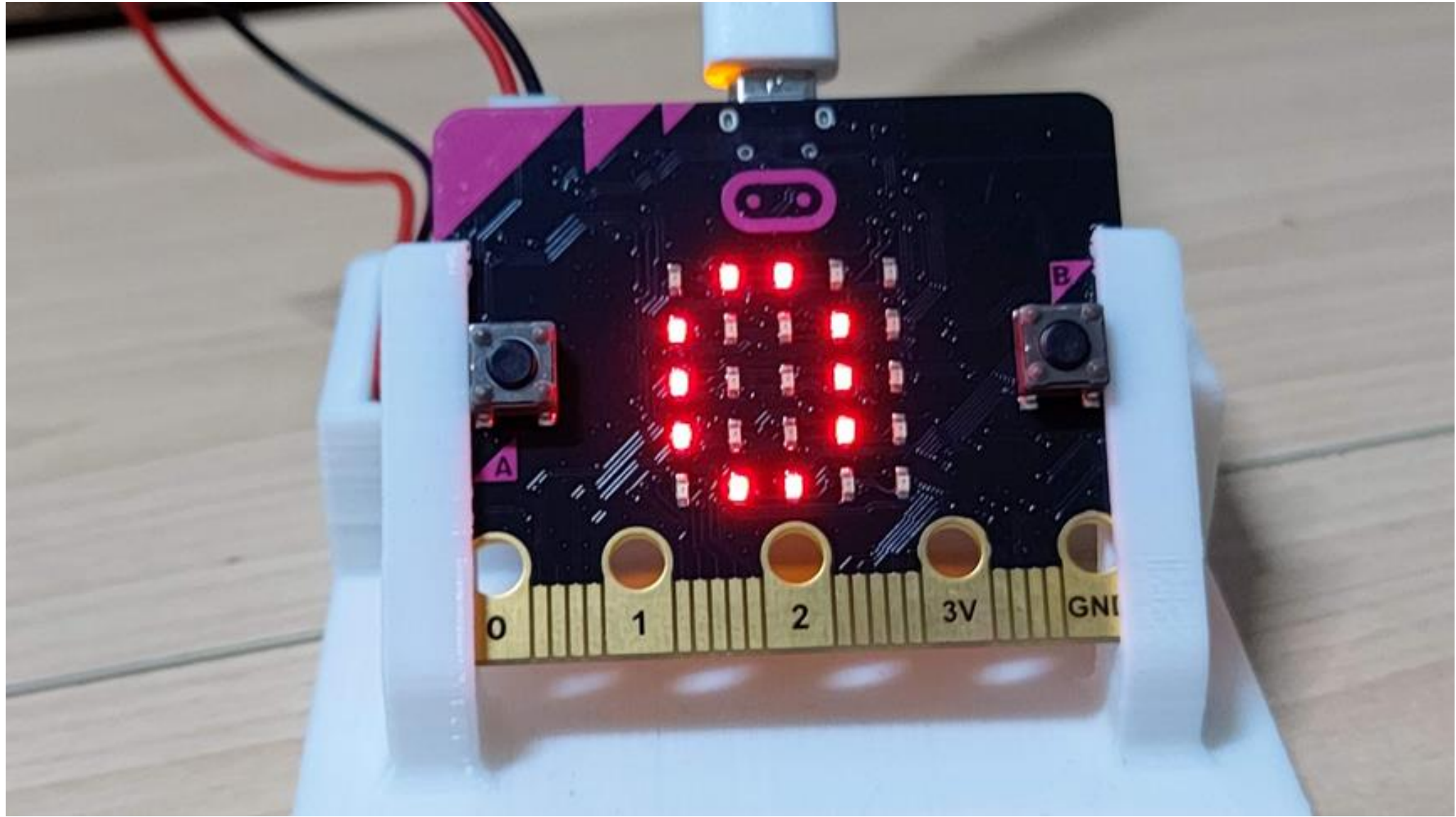




# 왼쪽의 시뮬레이터에 숫자가 출력됩니다



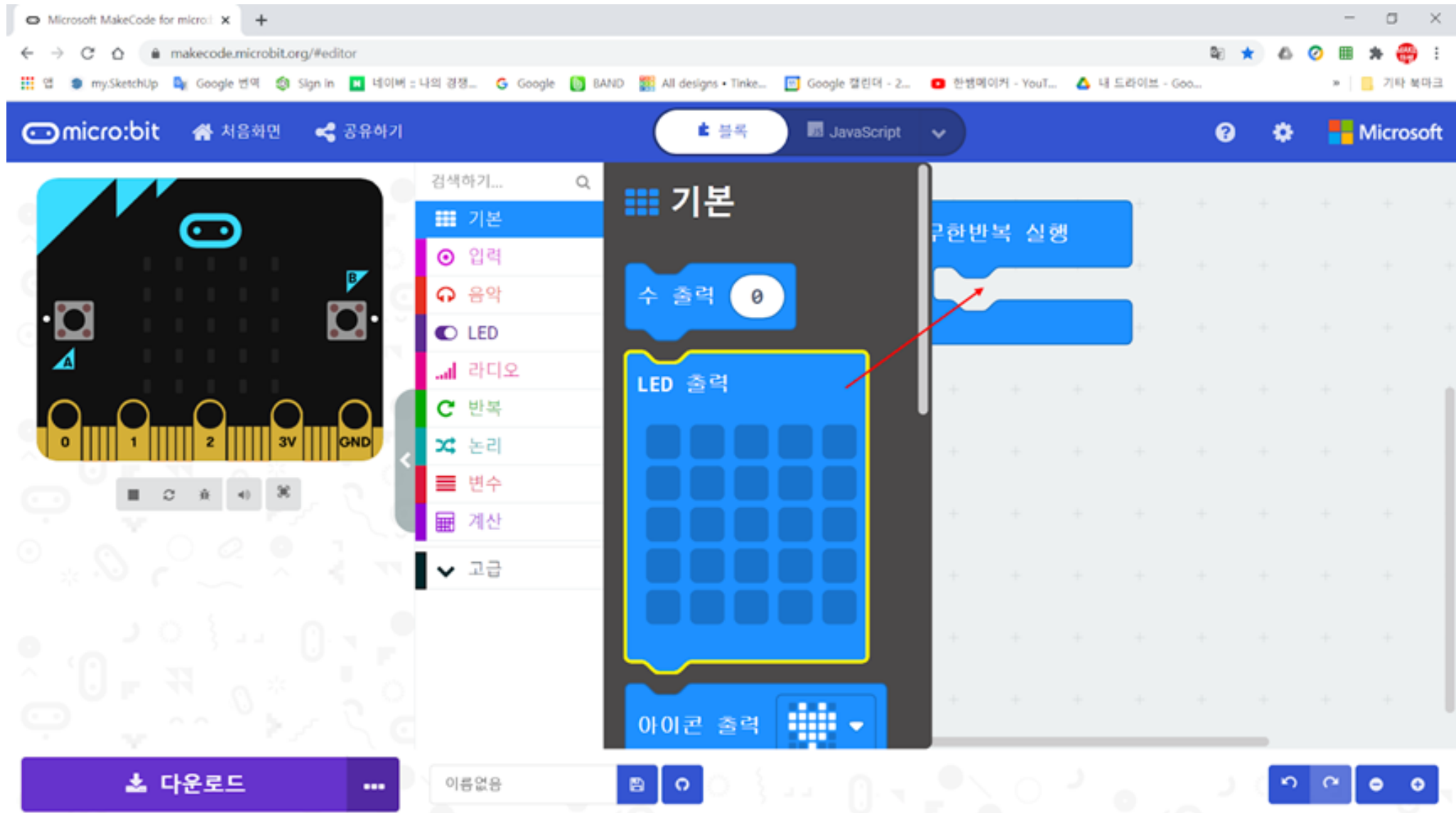
마이크로비트에 다운로드되었습니다





LED 출력하기

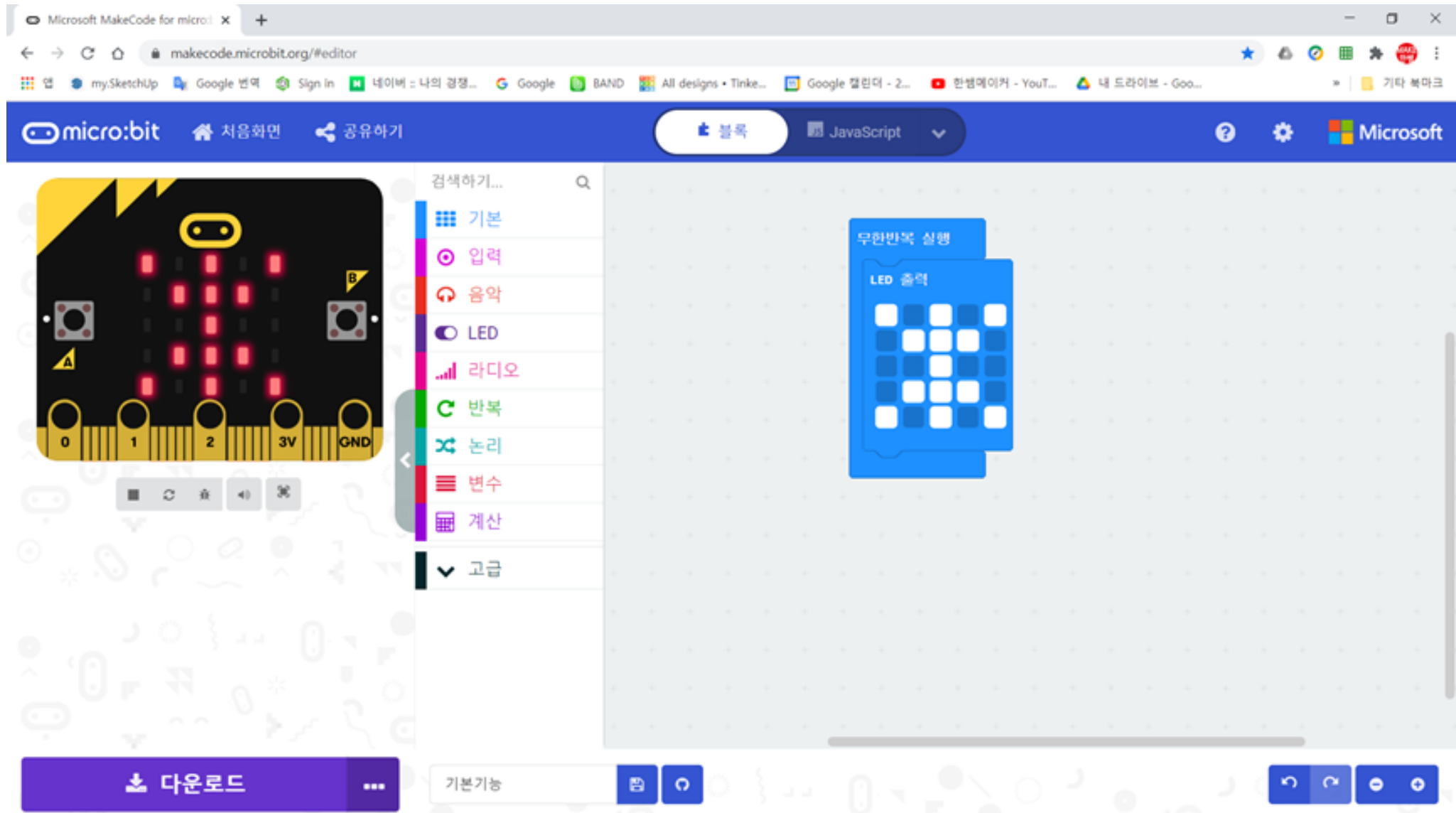
# 기본블록에서 LED출력을 선택합니다



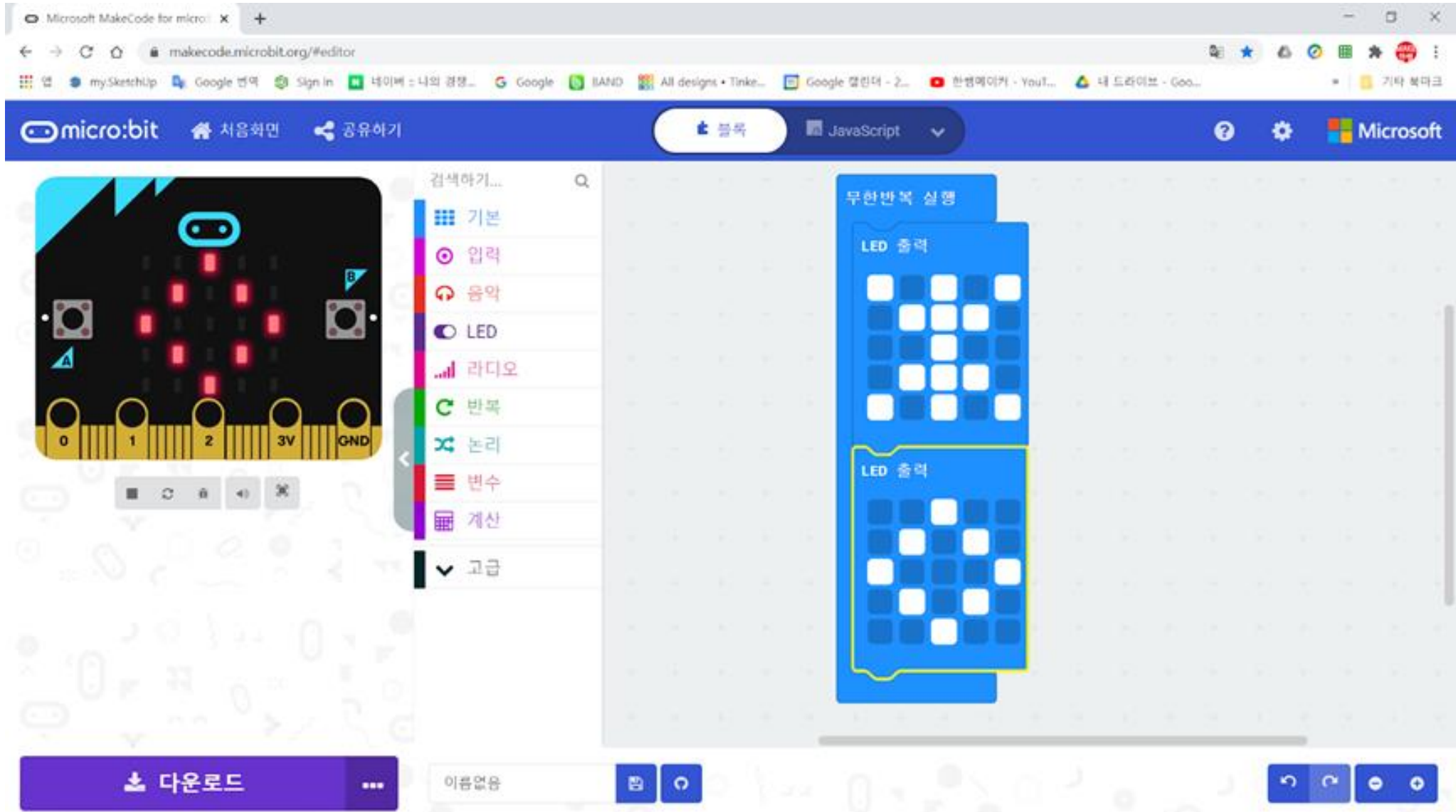
# 무한반복실행에 삽입합니다

The image shows the Microsoft MakeCode for micro:bit editor interface. The main workspace is a grid of plus signs. A blue block labeled '무한반복 실행' (Infinite Loop) is placed in the workspace. Inside this block is another blue block labeled 'LED 출력' (LED Output). The 'LED 출력' block is highlighted with a yellow border. On the left side, there is a virtual micro:bit board with a grid of red LEDs. Below the board are buttons for 'A' and 'B'. In the center, there is a block palette with various categories: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 반복 (Loop), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The top navigation bar includes 'micro:bit', '시작화면', '공유하기', '블록', 'JavaScript', and 'Microsoft'. The bottom bar has a '다운로드' (Download) button and a search bar.

# 무한반복실행에 LED출력 삽입 후, 점 그리기



# LED출력을 아래에 삽입하고 점 그리기

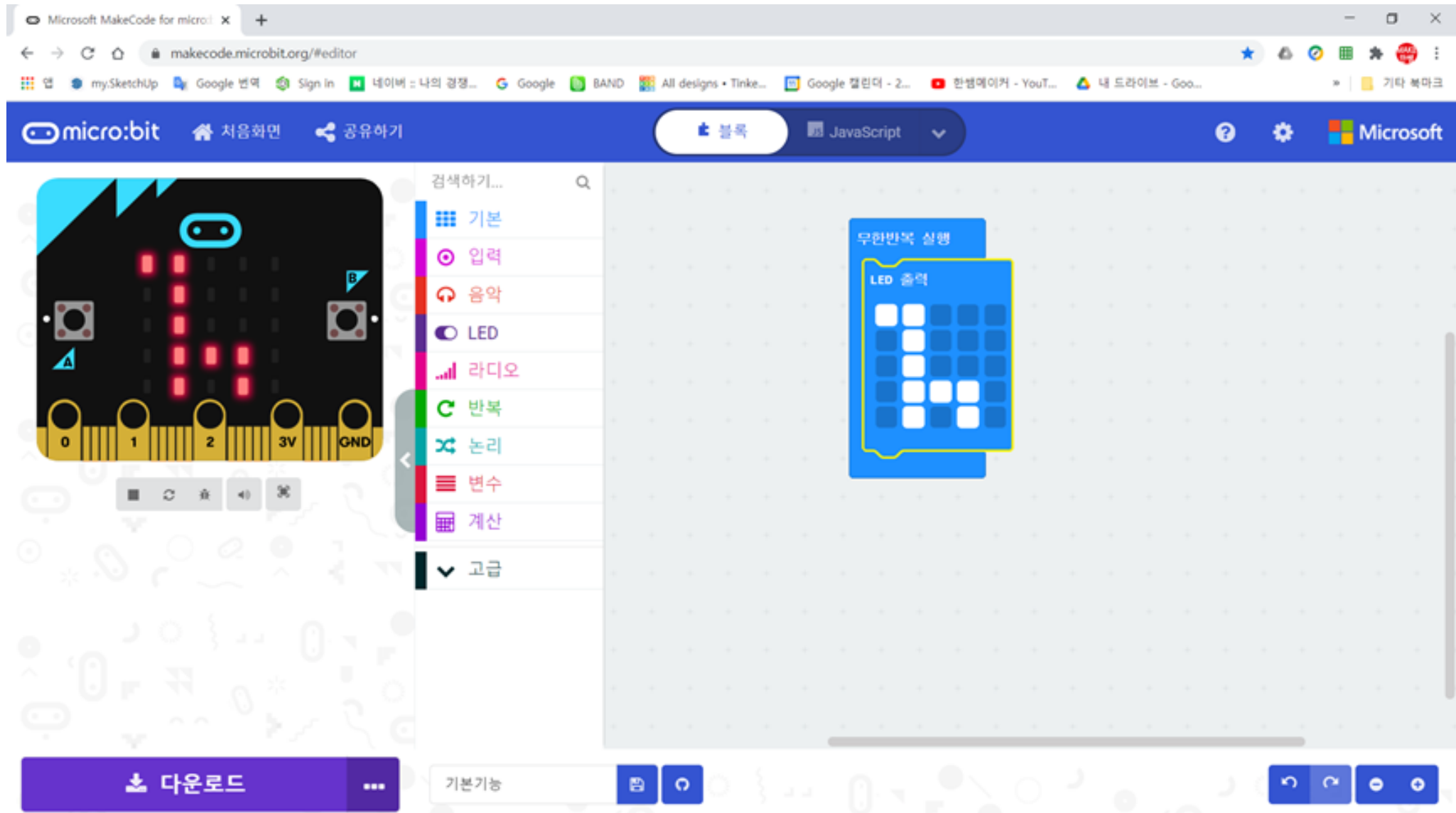




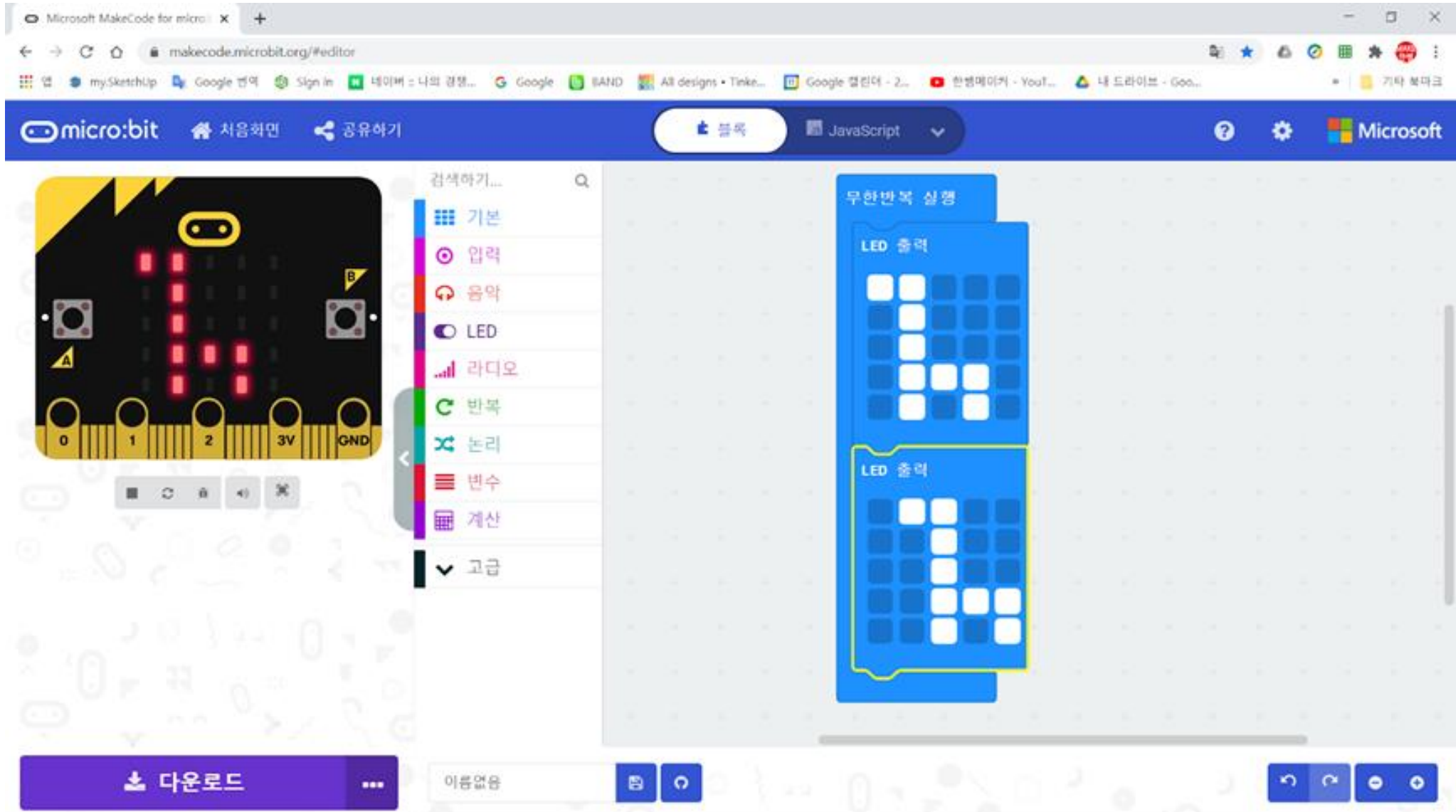
이동하는 효과 만들기



# 무한반복실행에 LED출력 삽입 후, 점 그리기



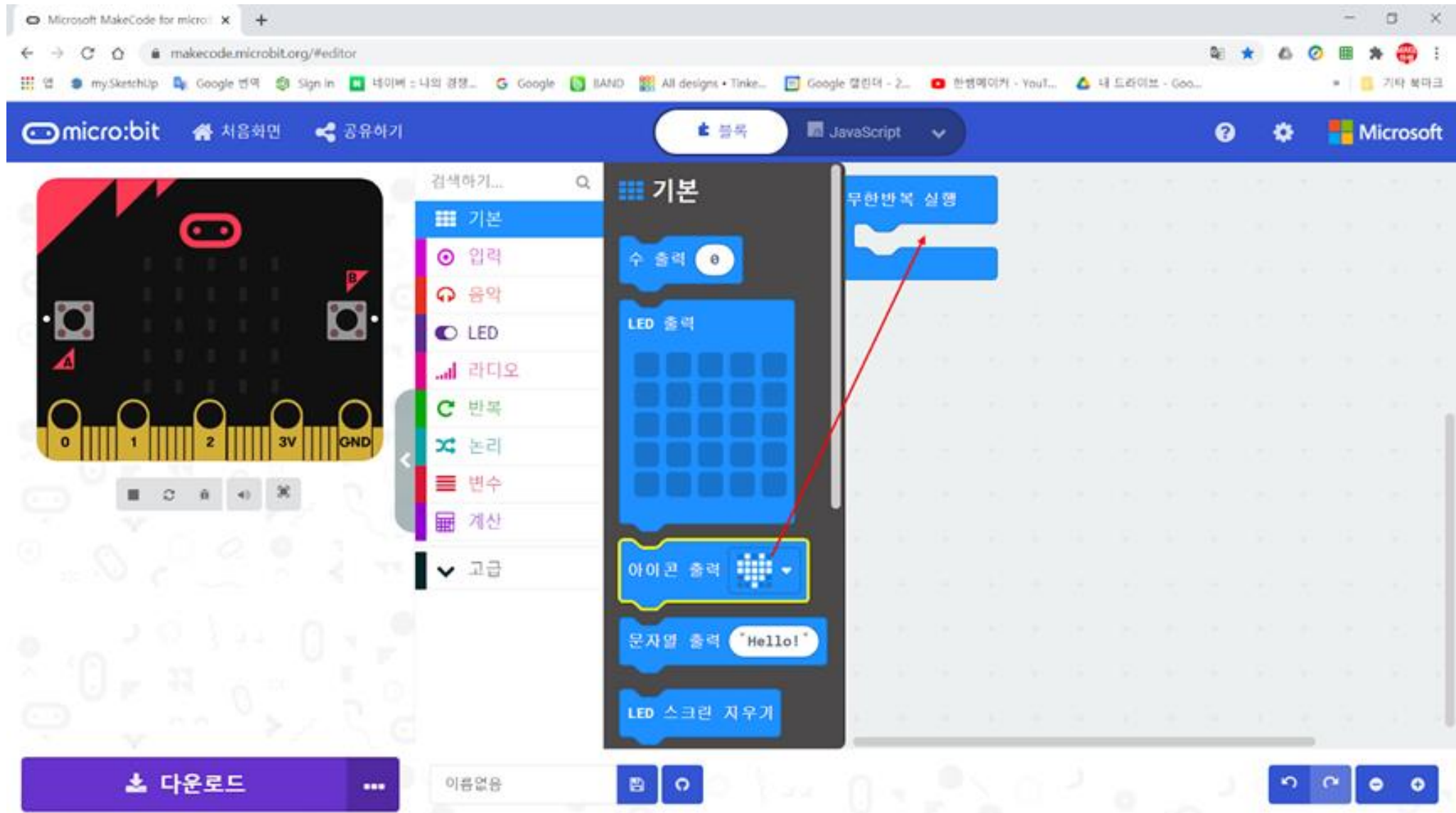
# LED출력을 아래에 삽입하고 점 그리기



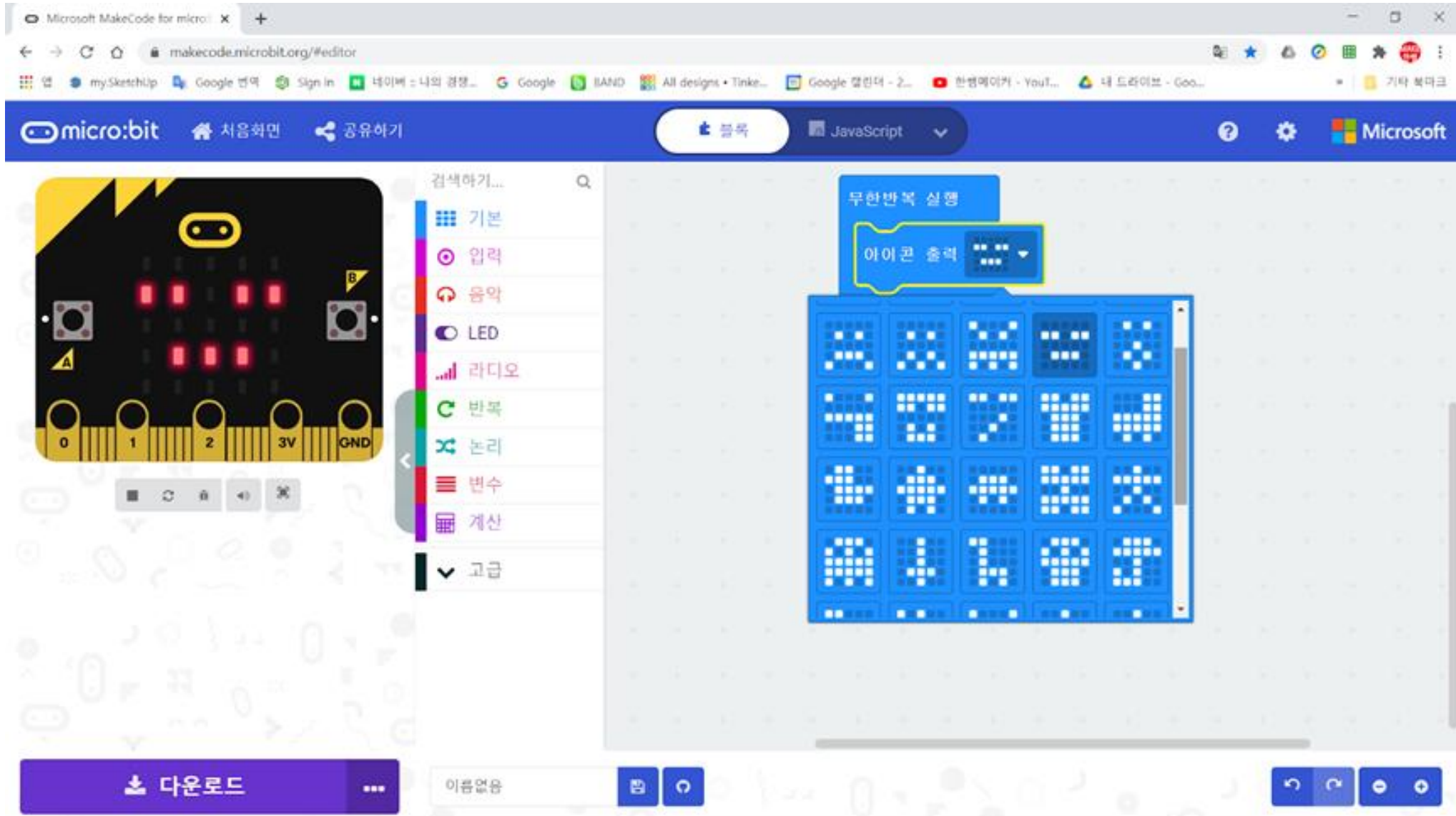


아이콘 출력

# 기본블록에서 아이콘출력을 꺼냅니다



# 다양한 아이콘이 설정되어 있습니다

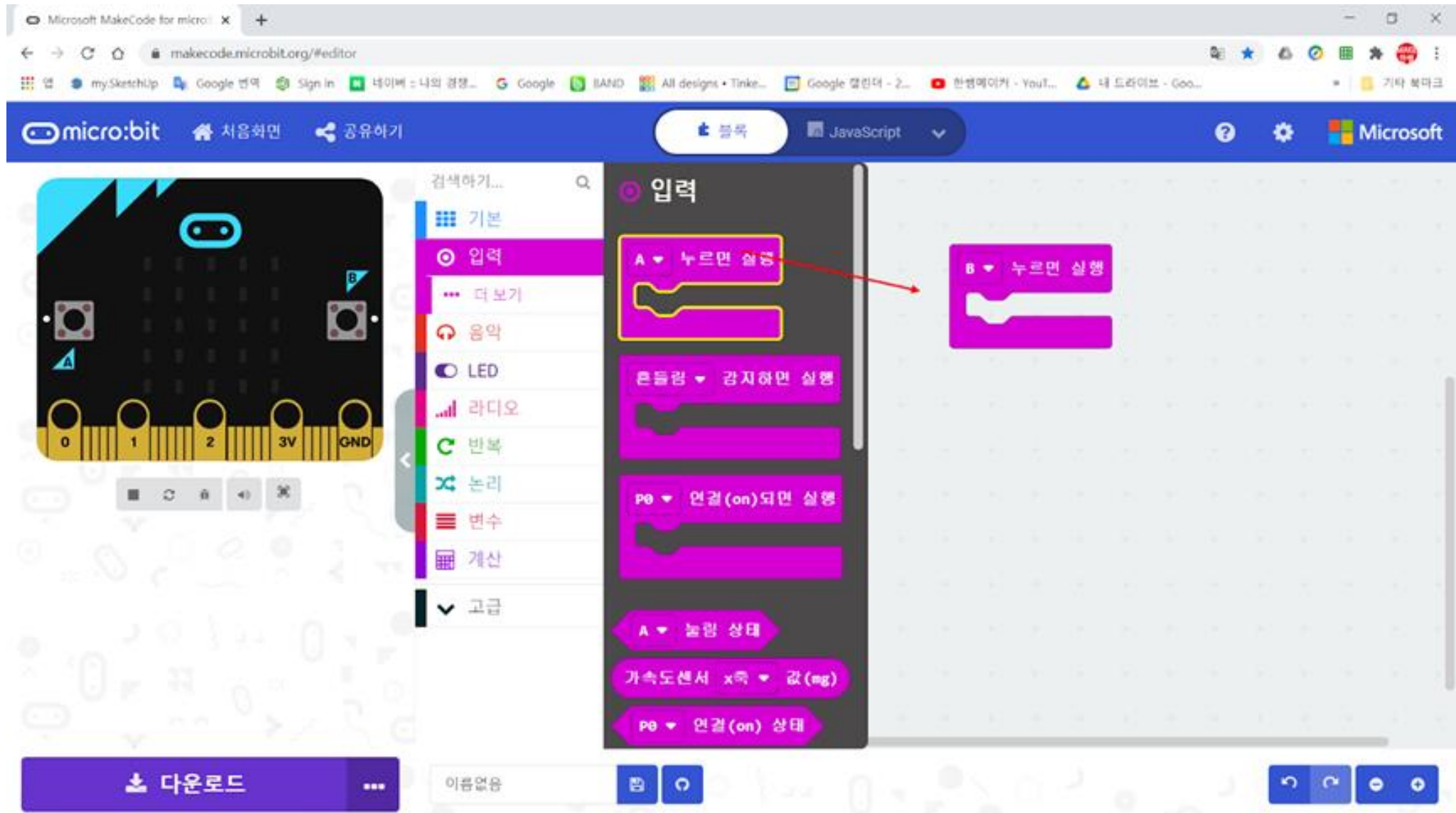




누르면 실행



# 입력블록에서 누르면 실행을 꺼냅니다





# A버튼, B버튼, A+B버튼 클릭 시 명령 수행

The image shows a screenshot of the Microsoft MakeCode for micro:bit editor interface. The browser address bar displays `makecode.microbit.org/#editor`. The top navigation bar includes the `micro:bit` logo, a home icon, a share icon, a search icon, and a dropdown menu currently set to `JavaScript`. The left sidebar features a virtual micro:bit board with buttons A and B highlighted in green. Below the board is a category menu with the following items: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The main workspace contains two event-driven blocks. The first block is triggered by button A and contains a 'Show Icon' block with a 4x4 grid icon. The second block is triggered by button B and also contains a 'Show Icon' block with a 4x4 grid icon. The bottom of the interface includes a '다운로드' (Download) button, a text input field with the text '이름없음' (Nameless), and a '실행' (Run) button.

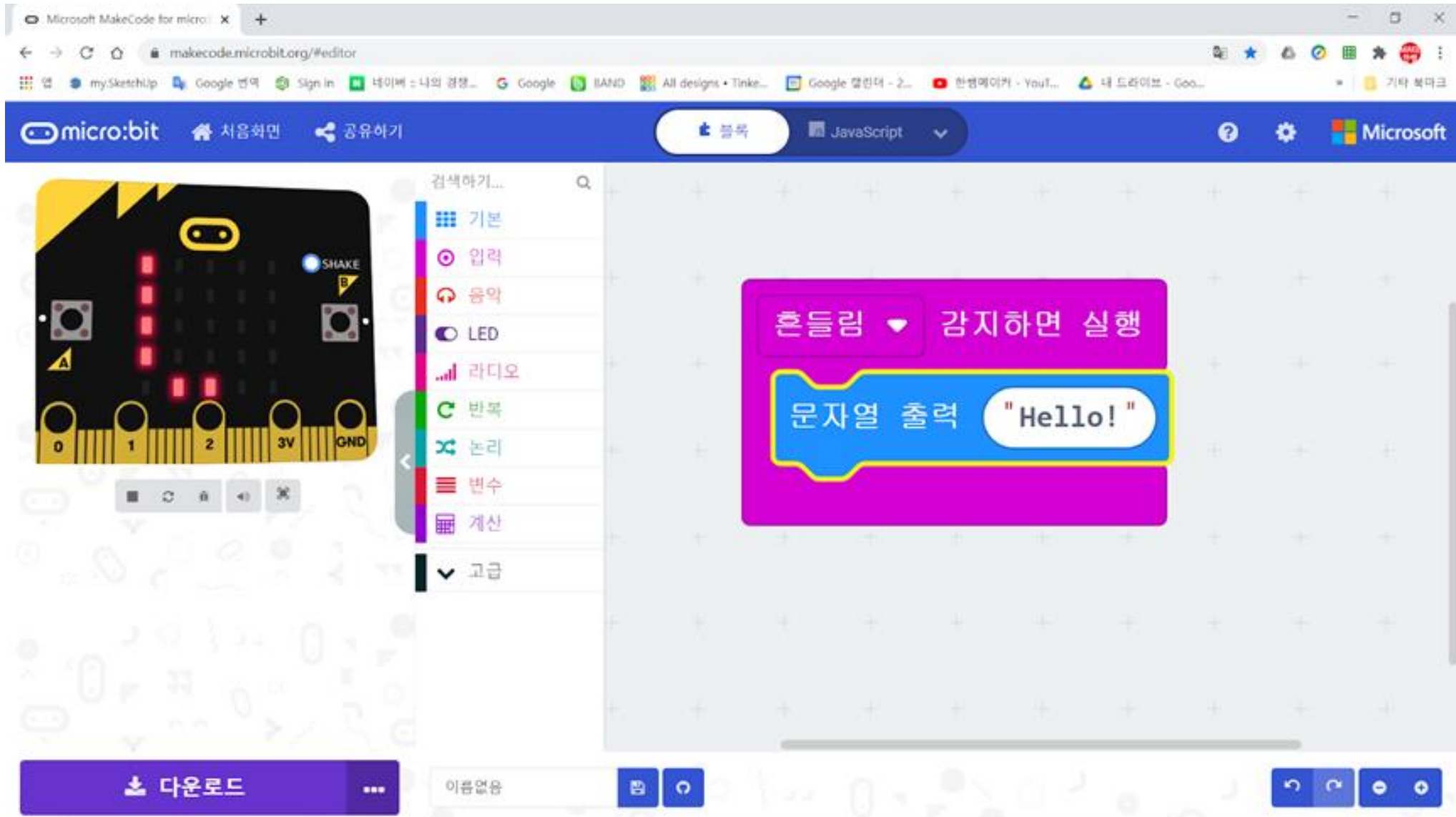


흔들면 감지하기

# 입력블록에서 감지하면 실행을 꺼냅니다

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The browser address bar displays `makecode.microbit.org/#editor`. The top navigation bar includes the `micro:bit` logo, a home icon, a share icon, a search icon, and a dropdown menu for `JavaScript`. The left sidebar shows a list of block categories: `기본` (Basic), `입력` (Inputs), `더 보기` (More), `음악` (Music), `LED`, `라디오` (Radio), `반복` (Loops), `논리` (Logic), `변수` (Variables), `계산` (Math), and `고급` (Advanced). The main workspace is a grid where two input blocks are visible. The top block is labeled `A` and `누르면 실행` (Press to Run). The bottom block is labeled `흔들림` (Shake) and `감지하면 실행` (Detect and Run). The `감지하면 실행` block is highlighted with a yellow border, and a red arrow points from it to the main workspace grid.

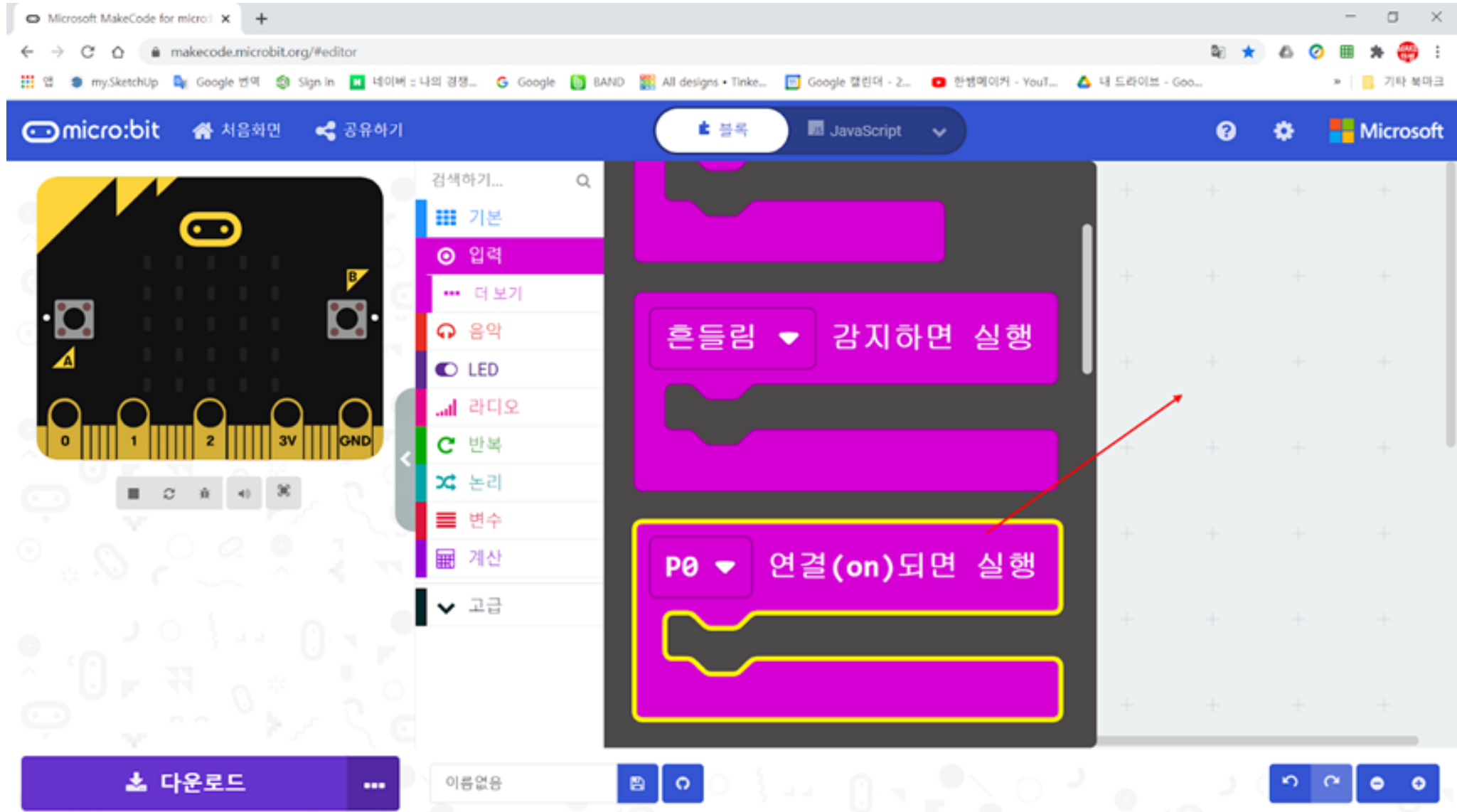
# 흔들면 명령을 수행합니다





연결되면 실행

# 입력블록에서 연결되면 실행을 꺼냅니다



# 핀이 연결되면 명령을 수행합니다

The image shows a screenshot of the Microsoft MakeCode for micro:bit editor. The browser address bar displays 'makecode.microbit.org/#editor'. The interface includes a top navigation bar with 'micro:bit', '시용하면', '공유하기', '블록', 'JavaScript', and 'Microsoft' logos. On the left, there is a visual representation of the micro:bit board with pins labeled 0, 1, 2, 3V, and GND. A central menu lists various block categories: 기본, 입력, 음악, LED, 라디오, 반복, 논리, 변수, 계산, and 고급. The main workspace contains a script with a purple '연결 (on) 되면 실행' block for pin P0. Inside this block is a blue '아이콘 출력' block with a grid icon. The bottom of the screen features a '다운로드' button, a search bar, and control buttons for undo, redo, and refresh.

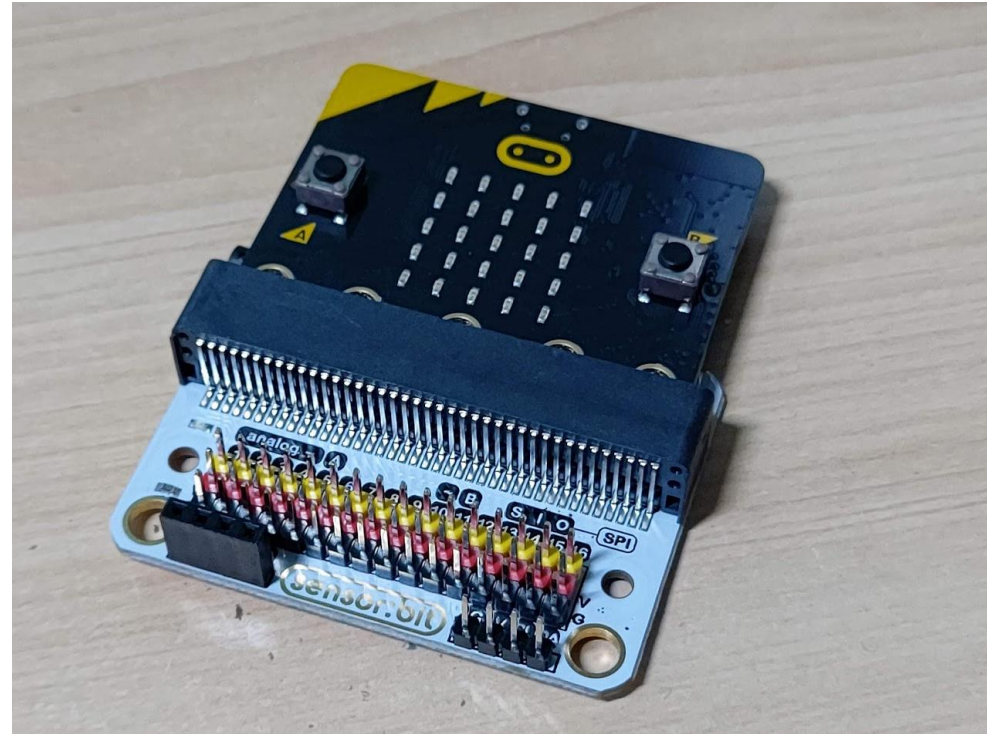




연주하기

# 연주하기 with 확장보드

- 마이크로비트는 소리출력이 되지 않으므로, 출력을 하려면, 핀으로 스피커, 이어폰 등에 연결해야 합니다.
- 이런 불편함을 해결하기 위해서 확장보드를 사용합니다.
- 나만의 곡을 작곡해보세요.^^



# 음악블록에서 도 출력을 꺼냅니다

The screenshot shows the Microsoft MakeCode for micro:bit interface. The top bar includes the 'micro:bit' logo, a home icon, a search icon, and a '블록' (Blocks) button. The main workspace is divided into three sections: a visual representation of the micro:bit board on the left, a block palette in the middle, and a script area on the right. The block palette is currently set to '음악' (Music) and lists various music-related blocks such as '도' (Note), '박자' (Rhythm), '멜로디' (Melody), and '종지' (Stop). The script area contains a sequence of music blocks, including '도 1 박자 출력' (Note C 1 beat output), '도 (Hz) 출력' (Note C (Hz) output), '1 박자 (ms) 유지' (1 beat (ms) hold), '다다동 멜로디 한 번 출력' (Scale C melody play once), '멜로디 재생 감지하면 실행' (Melody playback detected then execute), and '오른 중지' (Right stop). At the bottom, there is a '다운로드' (Download) button and a '연주하기' (Run) button.

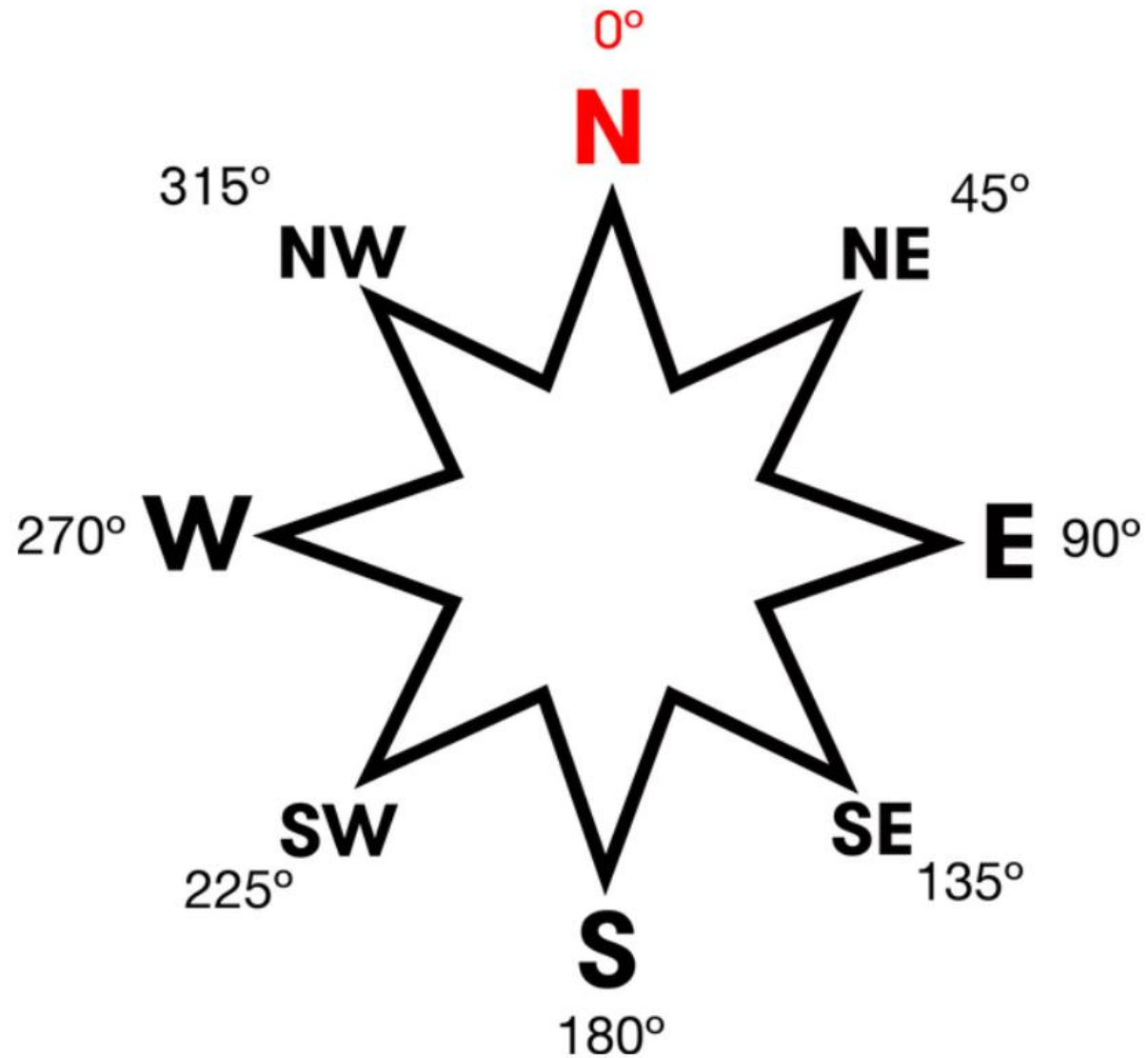
# 누르면 실행에 넣고 계이름을 만듭니다

The screenshot displays the Microsoft MakeCode for micro:bit interface. On the left, a micro:bit board is shown with a USB cable connected. A sidebar menu lists various categories: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The main workspace contains a script starting with '시작하면 실행' (When started) and '무한반복 실행' (Repeat forever). A '누르면 실행' (When button pressed) block is nested within the repeat loop, containing a sequence of six musical notes: 솔 (Sol), 미 (Mi), 미 (Mi), 파 (Fa), 레 (Re), and 레 (Re). Each note block is configured with a duration of 1 beat and a pitch of 400 Hz. At the bottom, there is a '다운로드' (Download) button and a '연주하기' (Run) button.



각도를 보여주는 나침반

# 방위 각도 알아보기

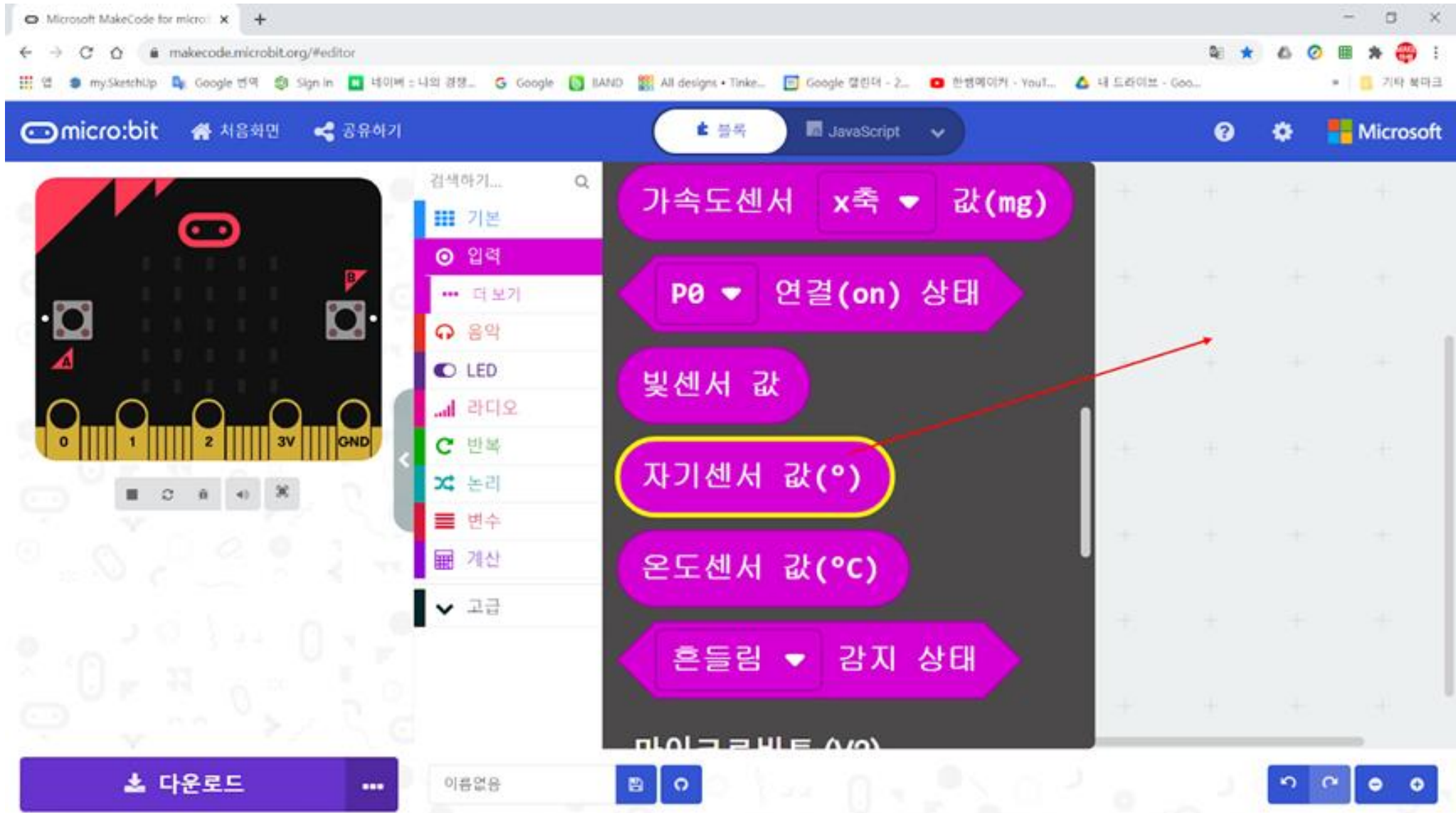


# 캘리브레이션(초기화)

- 자기센서 코딩을 다운받은 후에는 캘리브레이션(초기화)를 반드시 실행해야 한다
- 초기화를 하라는 문구가 지나간 후에, LED 도트가 켜지는데, 마이크로비트를 사방으로 기울여서 25개의 불이 다 켜지도록 하면 된다. 다 켜지면, 웃음 표시가 나면서, 캘리브레이션이 끝난다.



# 입력블록에서 자기센서값을 꺼냅니다.



# A누르면 실행 속, 수 출력에 삽입합니다.

The image shows a screenshot of the Microsoft MakeCode for micro:bit editor. The interface includes a top navigation bar with the 'micro:bit' logo, a home button, a share button, a search bar, and a language dropdown set to 'JavaScript'. On the left, there is a visual representation of the micro:bit board with a yellow pencil icon and the number '46' in the top right corner. Below the board are several control buttons. A central sidebar contains a search bar and a list of block categories: 기본, 입력, 더보기, 음악, LED, 라디오, 반복, 논리, 변수, 계산, and 고급. The main workspace on the right features a grid background with a script consisting of two blocks: a purple 'A 누르면 실행' block and a blue '수 출력 자기센서 값(°)' block. The '수 출력' block is highlighted with a yellow and blue border. At the bottom, there is a '다운로드' button, a text input field containing '이름없음', and several control buttons.

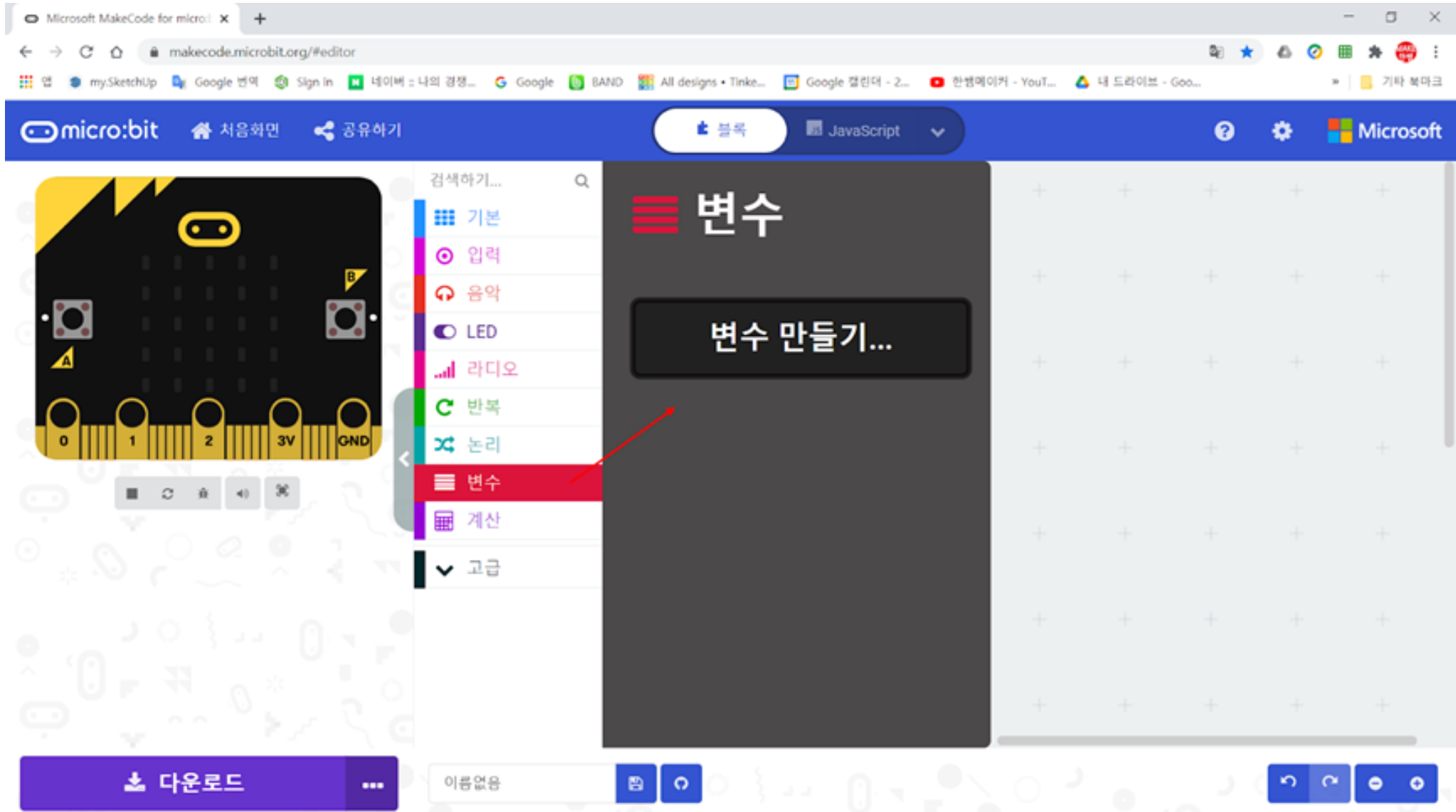


북쪽쪽을 향하면 N자 보여주기

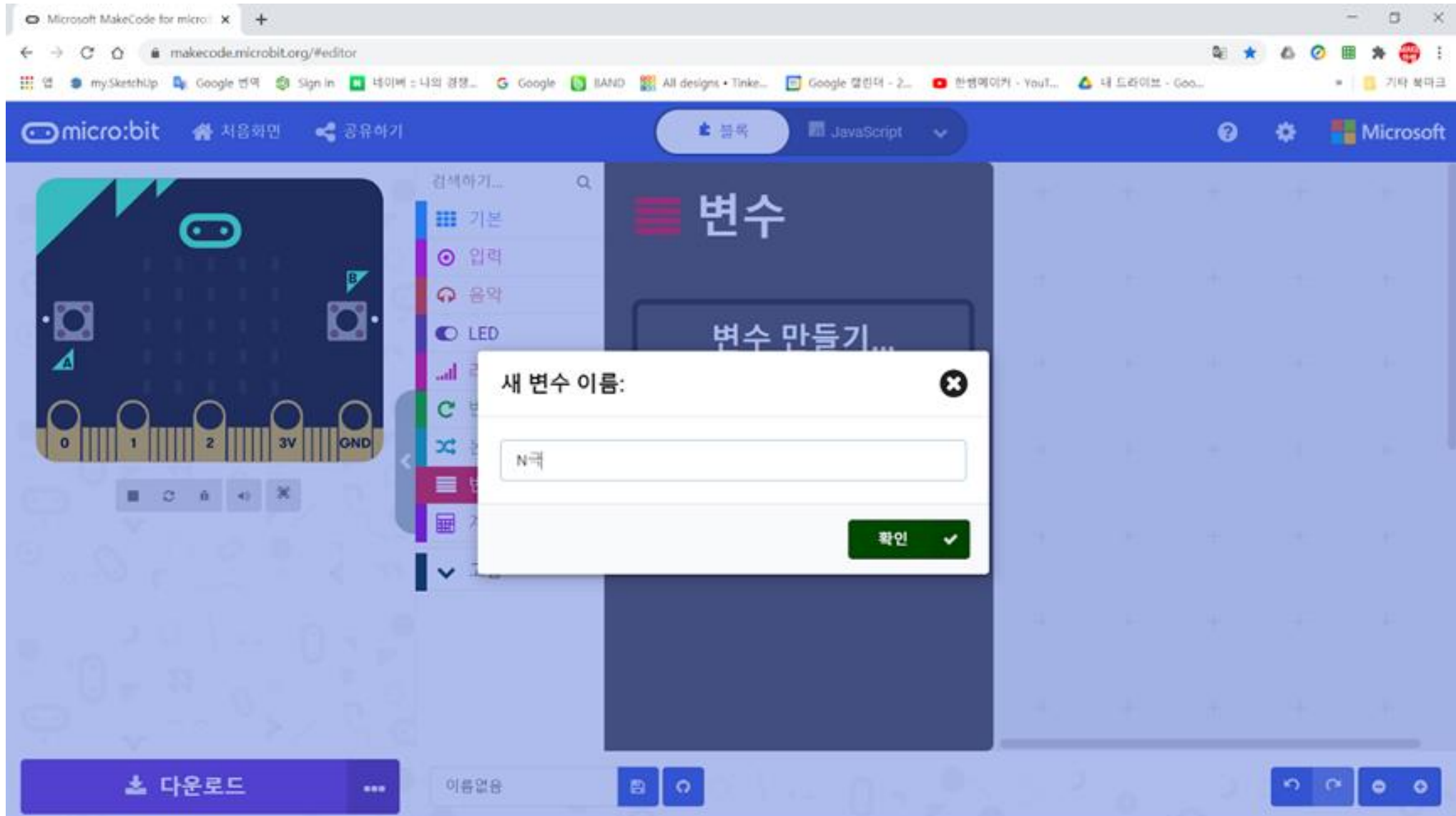
# 변수와 if~else

- 센서로 값을 읽어드릴 경우, 이 값을 저장할 장소가 필요한데, 이 장소를 변수라고 합니다.
- 변수의 이름은 한글, 영어 다 가능하며, 알아보기 쉽게 정합니다
- 우리가 말할 때 "만약 ~라면, ~하고, 아니면 ~합니다" 라고 말하는 것을 조건문이라고 하는데, 조건에 따라 수행하는 명령이 다를 때 if~else 블록을 사용합니다.
- 만약 ~라면 이라는 조건이 1개 일 수도 있지만, 여러 개일 경우에는 +버튼을 클릭해서 조건을 늘일 수 있습니다.

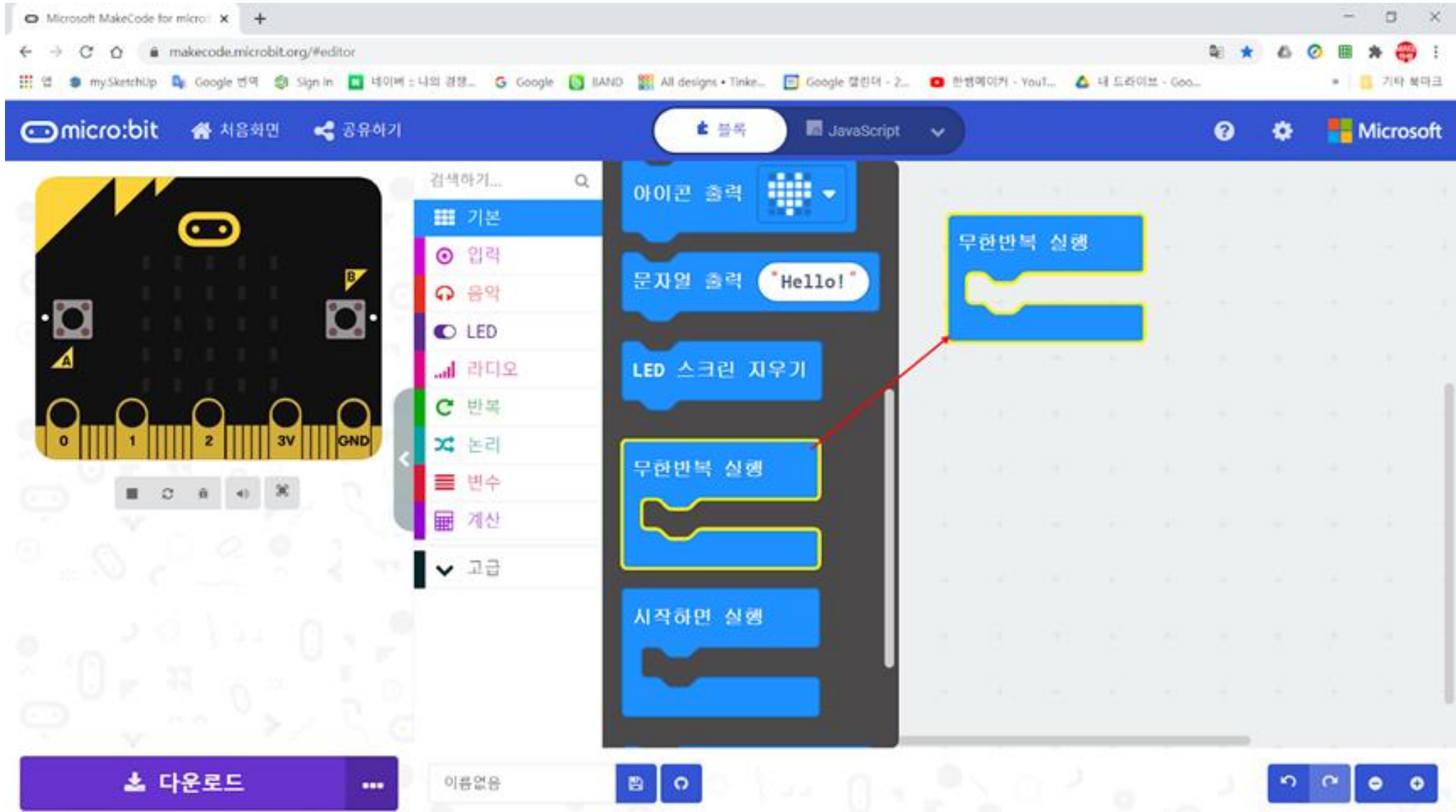
# 변수블록을 클릭하면 변수만들기 클릭



새 변수 이름을 N극이라고 입력하고 확인클릭



# 기본블록에서 무한반복실행 블록 꺼내기





# 변수에서 N극에 저장 꺼내서 삽입

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. On the left is a virtual micro:bit board. A central sidebar contains a menu with categories like '기본', '입력', '음악', 'LED', '라디오', '반복', '논리', '변수', '계산', and '고급'. The '변수' (Variables) category is selected, displaying a '변수' (Variables) panel with a '변수 만들기...' (Create variable...) button and three variable blocks: 'N극' (N-pole), '에 0 저장' (store 0), and '값 1 증가' (increase by 1). A red box highlights the 'N극' dropdown and the '에 0 저장' block. A red arrow points from this box to a '무한반복 실행' (Repeat forever) loop on the workspace, which contains a block with 'N극' (N-pole), '에 0 저장' (store 0), and '값 1 증가' (increase by 1). The workspace also shows a '다ownload' (Download) button at the bottom left and a '이름없음' (Nameless) label at the bottom center.

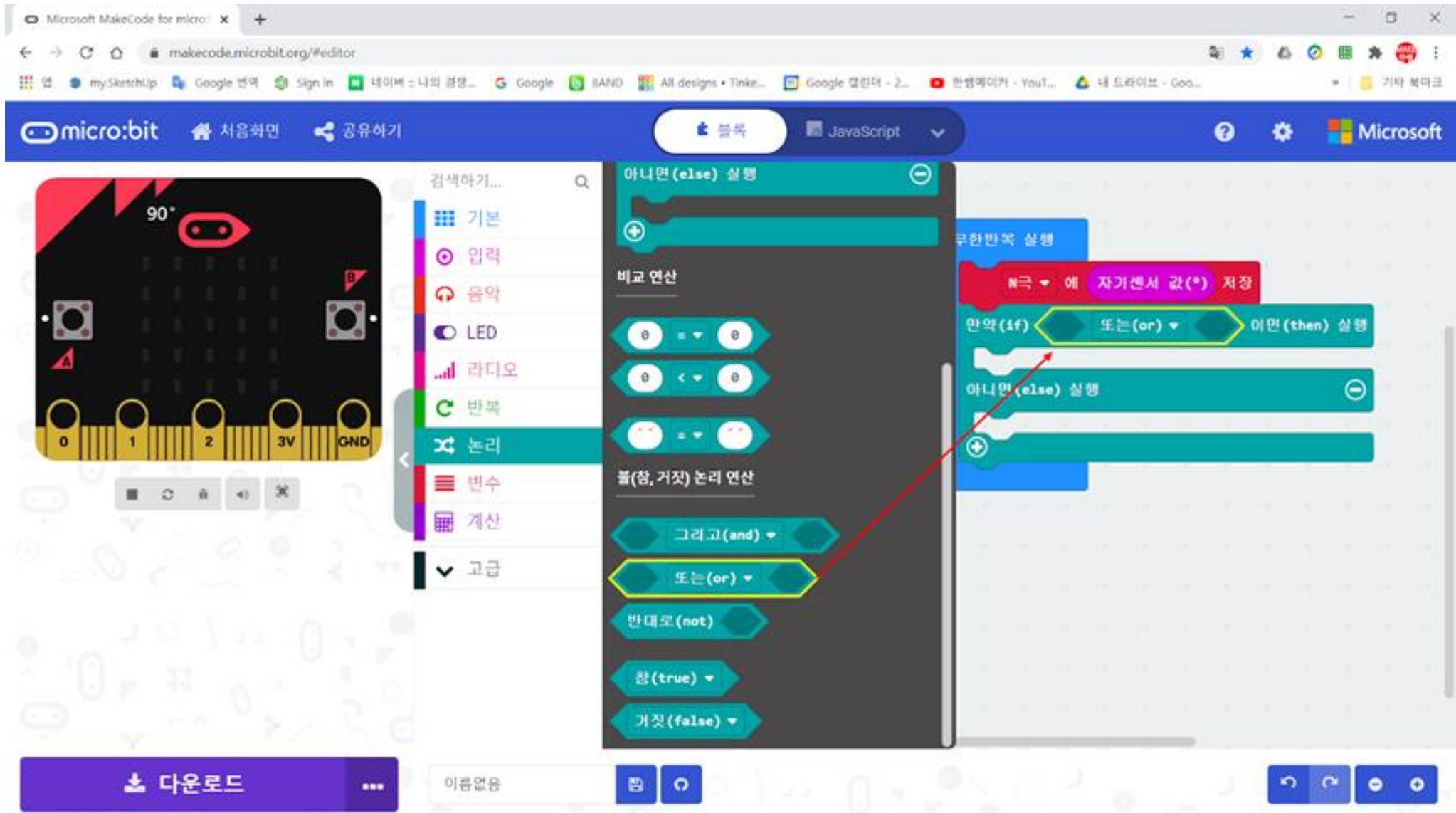
# 입력블록에서 자기센서값을 꺼내서 삽입

The image shows the Microsoft MakeCode for micro:bit editor interface. On the left, there is a visual representation of the micro:bit board with a 90-degree rotation indicator. A central sidebar contains a search bar and a category menu with options like '기본', '입력', '음악', 'LED', '라디오', '반복', '논리', '변수', '계산', and '고급'. The main workspace displays a code block with a 'P0 연결(on)되면 실행' block containing a '자기센서 값(°)' block. A red arrow points from this block to a '무한반복 실행' loop block, where the '자기센서 값(°)' block is being inserted into the 'N극' field. The bottom of the screen features a '다운로드' button and a name input field.

# 논리블록에서 if~else 블록을 꺼내서 삽입

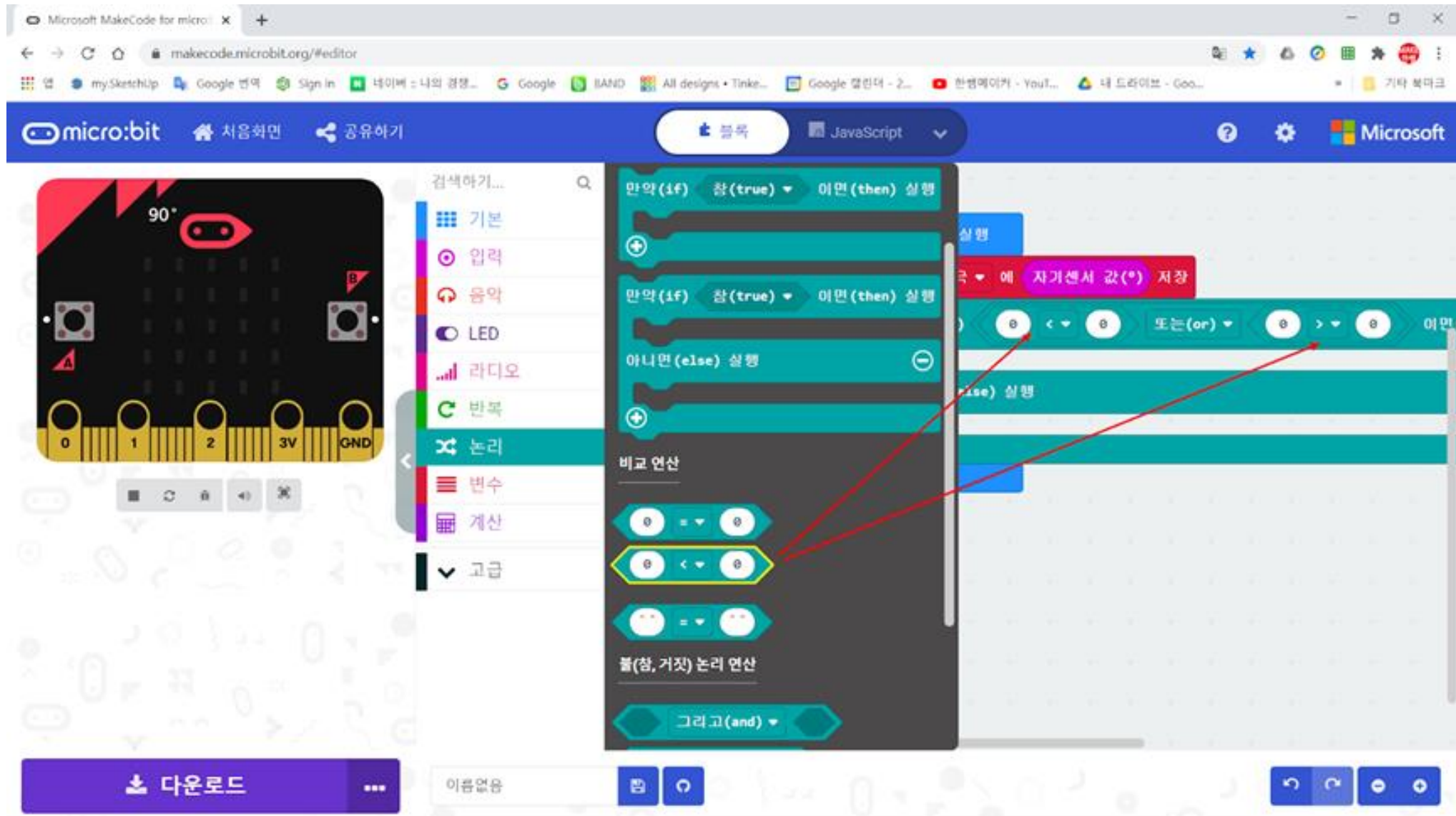
The image shows the Microsoft MakeCode for micro:bit editor interface. On the left, there is a visual representation of the micro:bit board. In the center, a sidebar menu lists various block categories: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The '논리' (Logic) category is selected, displaying a list of logic blocks under '조건/선택 실행' (Conditional/Choice Execution). These include '만약 (if) 참 (true) 이면 (then) 실행' (If true then execute) and '아니면 (else) 실행' (Else execute) blocks. A red arrow points from one of these blocks in the sidebar to a script on the right. The script on the right shows a '무한반복 실행' (Repeat until loop) block containing an 'if-else' structure. The 'if' block is currently set to '참 (true)' and contains a '자기생서 값 (\*) 저장' (Store value) block. The 'else' block is currently empty. The interface also shows a search bar at the top left, a navigation bar with 'micro:bit' and '블록' (Blocks) tabs, and a '다운로드' (Download) button at the bottom left.

# 논리에서 또는 블록을 꺼내어 삽입





# 논리에서 부등호 블록을 꺼내어 삽입



# 변수에서 N극을 꺼내서 2곳에 삽입

The screenshot displays the Microsoft MakeCode for micro:bit web editor. On the left, a virtual micro:bit board is shown with a 90-degree rotation. A central sidebar menu lists various code blocks: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The '변수' (Variables) menu is open, showing options to create a new variable and a dropdown menu with 'N극' (North Pole) selected. A red arrow points from this 'N극' dropdown to two locations in the main workspace: the 'N극' dropdown in a '자기센서 값(\*) 저장' (Save sensor value) block and the 'N극' dropdown in an '만약 (if)' (If) block. The workspace also contains a '무한반복 실행' (Repeat) block and an '아니면 (else) 실행' (Else) block.

# 45와 315를 입력

The image shows a screenshot of the Microsoft MakeCode for micro:bit editor. The interface includes a top navigation bar with the 'micro:bit' logo, a search bar, and a dropdown menu set to 'JavaScript'. On the left, there is a visual representation of the micro:bit board with pins labeled 0, 1, 2, 3V, and GND. A central sidebar contains a search bar and a list of code categories: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The main workspace displays a '무한반복 실행' (Infinite Loop) block. Inside this loop, there is a '자기센서 값(°) 저장' (Store Gyro Sensor Value) block. Below it is an '만약(if)' (If) block with a red 'N극' (North) magnet. The 'if' block contains two conditions: '< 45' and '> 315', both of which are circled in red. The 'if' block is followed by an '아니면(else) 실행' (Else) block, which is currently empty. At the bottom of the editor, there is a '다운로드' (Download) button and a '이름없음' (Nameless) label.



# 문자열 출력 블록을 꺼내어 삽입

The image shows the Microsoft MakeCode for micro:bit editor interface. On the left is a virtual micro:bit board. The central palette is set to '기본' (Basic) and lists various block categories. A '문자열 출력' (Text Output) block containing the text 'Hello!' is highlighted in yellow in the palette. A red arrow points from this block to the '아니면 (else) 실행' (else) branch of an '만약 (if)' (if) statement in the script area. The script area contains a '무한반복 실행' (Forever loop) block with the following logic: 'N극' (North) sensor is greater than or equal to 45, then '문자열 출력' (Text Output) 'Hello!'; otherwise, '문자열 출력' (Text Output) 'Hello!'. The '문자열 출력' block in the palette is being dragged into the '아니면 (else) 실행' branch.

# N과 없음 을 입력

The image shows a screenshot of the Microsoft MakeCode for micro:bit editor. The interface includes a top navigation bar with the 'micro:bit' logo, a search bar, and a dropdown menu currently set to 'JavaScript'. On the left, there is a visual representation of the micro:bit board and a category menu with options like '기본', '입력', '음악', 'LED', '라디오', '반복', '논리', '변수', '계산', and '고급'. The main workspace contains a Scratch-style script:

- A red block: "N극" (North Pole) to "자기센서 값(°)" (Magnetometer value in degrees) 저장 (Save).
- A teal "만약(if)" (If) block with a dropdown set to "N극" (North Pole) and a value of "45".
- Inside the "if" block, a blue "문자열 출력" (Print string) block containing the character "N".
- A teal "아니면(else) 실행" (Else) block.
- Inside the "else" block, a blue "문자열 출력" (Print string) block containing two empty quotes "".

At the bottom, there is a "다운로드" (Download) button and a field for the script name, which is currently empty.



# 라디오 기능 사용하기

# 라디오 기능

- 마이크로비트끼리 무선으로 연결하는 것
- 라디오가 주파수를 맞추면 연결되는 것처럼 주파수를 맞추고 통제하는 것
- 송신을 위한 코딩과 수신을 위한 코딩이 나뉘지며, 송신과 수신 코딩을 함께 다운받기도 한다
- 라디오송신이 가능하려면, 마이크로비트가 동일한 그룹으로 설정되어 있어야 합니다.

# 시작하면 실행블록에 그룹을설정 블록 삽입

The screenshot displays the Microsoft MakeCode for micro:bit editor interface. On the left, a virtual micro:bit board is shown with its pins labeled 0, 1, 2, 3V, and GND. A central menu lists various block categories: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 더 보기 (More), 반복 (Loop), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The '라디오' (Radio) category is selected, and its blocks are visible in the main workspace. A script is being built with the following blocks:

- 라디오 그룹을 1 로 설정 (Set radio group to 1) - highlighted in pink
- 라디오 전송:수 0 (Radio send: count 0) - highlighted in pink
- 라디오 전송:변수값 "name" = 0 (Radio send: variable value "name" = 0) - highlighted in pink
- 라디오 전송:문자열 "" (Radio send: string "") - highlighted in pink
- 라디오 수신하면 실행: receivedString (Radio receive: run: receivedString) - highlighted in pink

On the right side of the workspace, a blue '시작하면 실행' (When green flag clicked) block is present, containing a pink '라디오 그룹을 1 로 설정' (Set radio group to 1) block. This pink block is highlighted with a yellow border, indicating it is the block being inserted into the script.

At the bottom of the editor, there is a '다운로드' (Download) button and a '기본기능' (Basic functions) section with icons for a micro:bit and a radio tower.



# A버튼을 누르면 Hi 라고 출력하도록 코딩

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The top navigation bar includes the 'micro:bit' logo, a home button, a share button, a '블록' (Blocks) button, a language dropdown set to 'JavaScript', and the Microsoft logo. The left sidebar contains a search bar and a category menu with options: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 더보기 (More), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), and 고급 (Advanced). The main workspace displays a code block starting with '시작하면 실행' (When green flag clicked), followed by '라디오 그룹을 1로 설정' (Set radio group to 1). Below this is a '라디오 수신하면 실행' (When radio received) block containing a '문자열 출력 receivedString' (Print string receivedString) block. To the right, an 'A 버튼을 누르면 실행' (When button A pressed) block contains a '라디오 전송: 문자열 "Hi"' (Send radio: string "Hi") block. The bottom of the editor features a '다운로드' (Download) button, a '기본기능' (Basic) dropdown, and several utility icons.

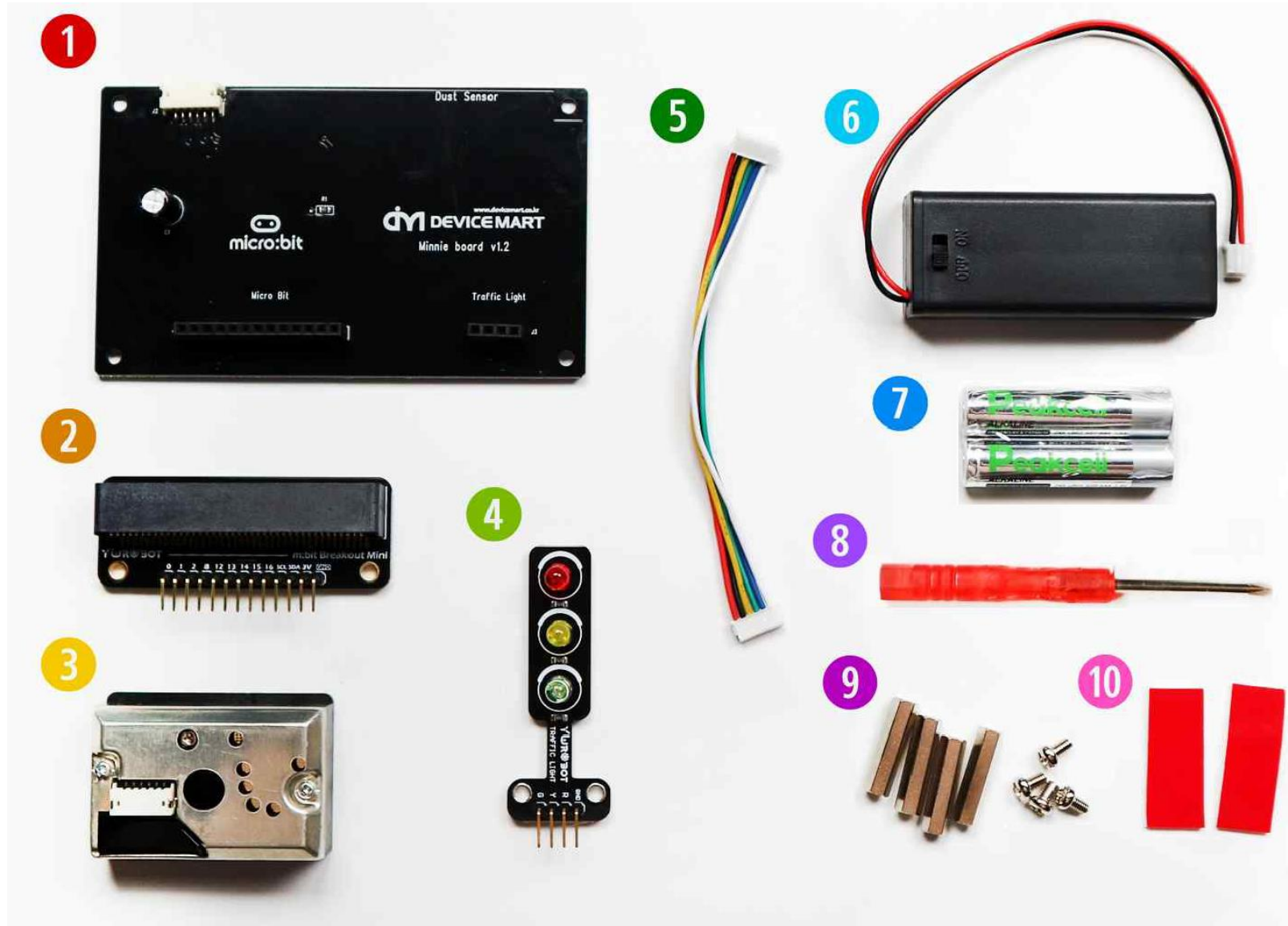
# 미세먼지키트 만들기





# 하드웨어 조립하기

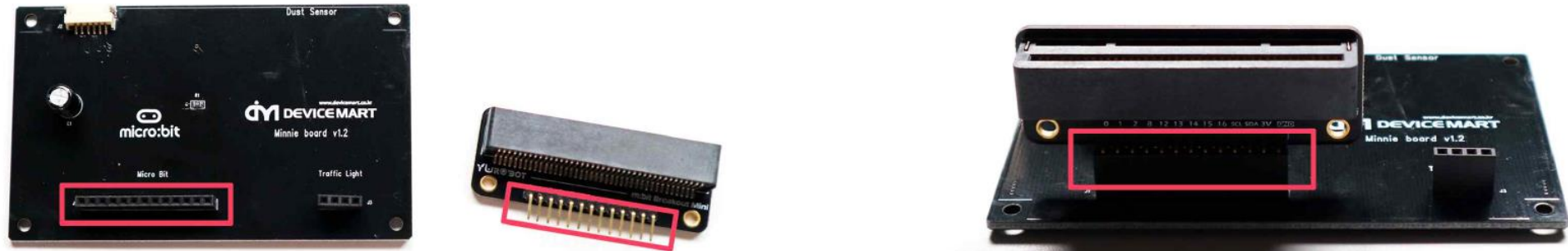
# 미세먼지측정기 부품



# 부품 이름

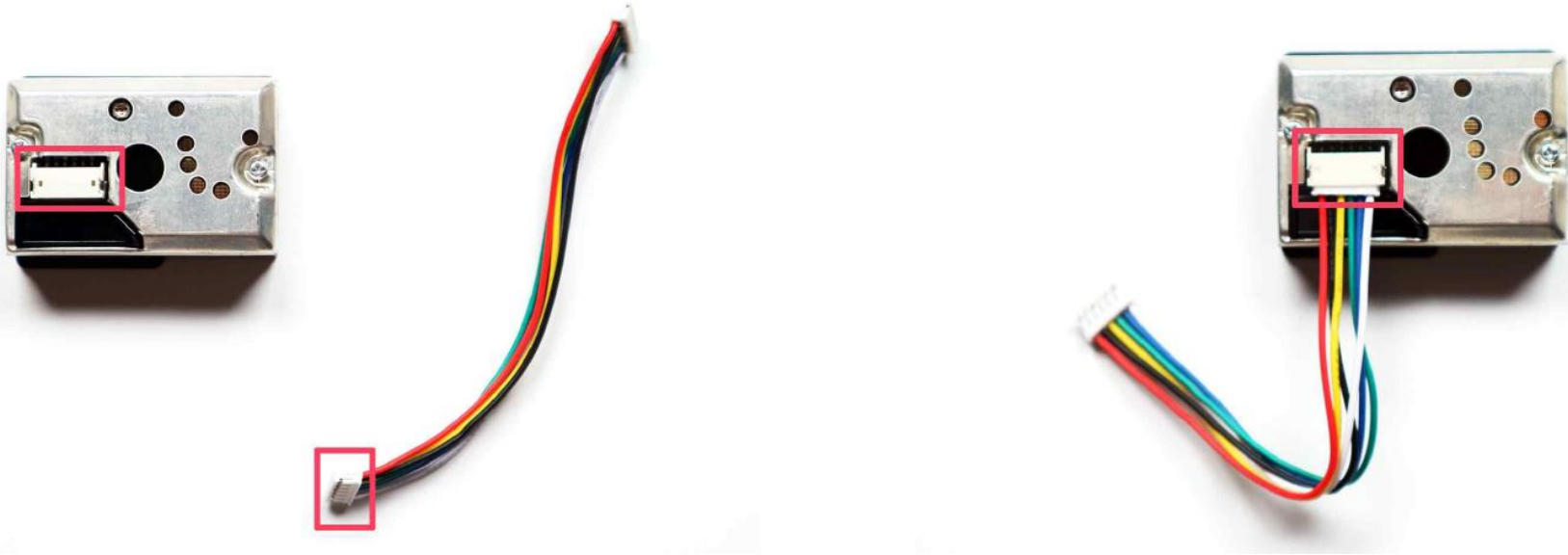
1	미세먼지 PCB 보드	회로를 쉽게 연결하기 위해 제작된 보드
2	마이크로비트 IO확장보드	마이크로비트 연결을 쉽게 하도록 제작된 보드
3	미세먼지 센서	미세먼지를 감지하는 센서
4	신호등 모듈	신호등으로 상태를 표시하는 모듈
5	미세먼지 센서케이블	미세먼지 센서와 회로를 연결하는 케이블
6	AAA 배터리 홀더	AAA 배터리 케이스(단독으로 활용 시 사용)
7	AAA 배터리	
8	미니 십자드라이버	볼트 조립을 위한 드라이버
9	서포트 & 볼트	
10	양면 테이프	

# 미세먼지측정기 조립하기



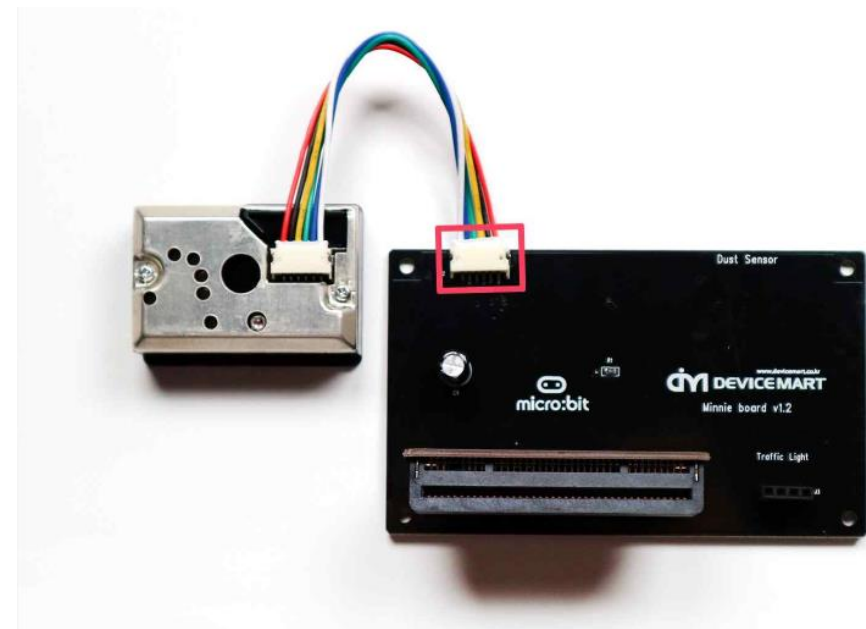
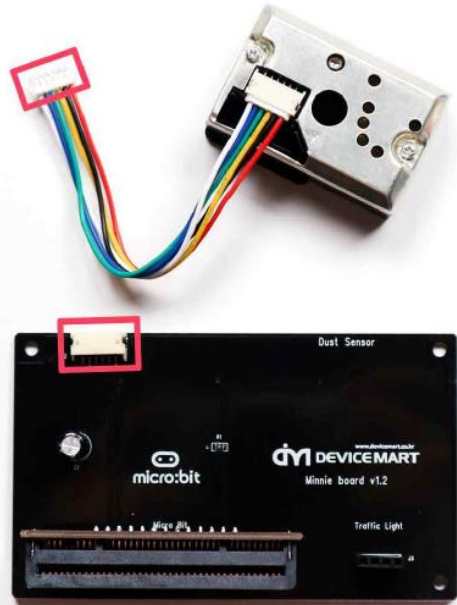
미세먼지 PCB 보드에 마이크로비트 IO 확장보드를 장착합니다

# 미세먼지측정기 조립하기



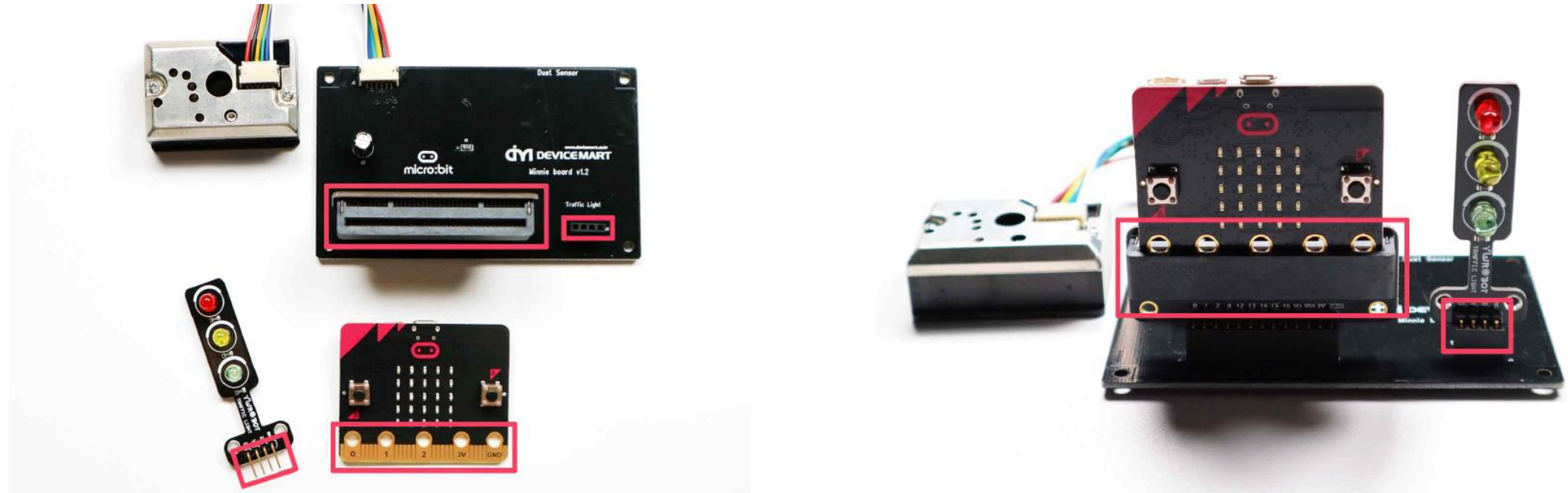
미세먼지 센서에 미세먼지센서 케이블을 연결합니다

# 미세먼지측정기 조립하기



미세먼지 센서에 연결한 케이블의 반대쪽을 미세먼지 PCB 보드에 연결합니다

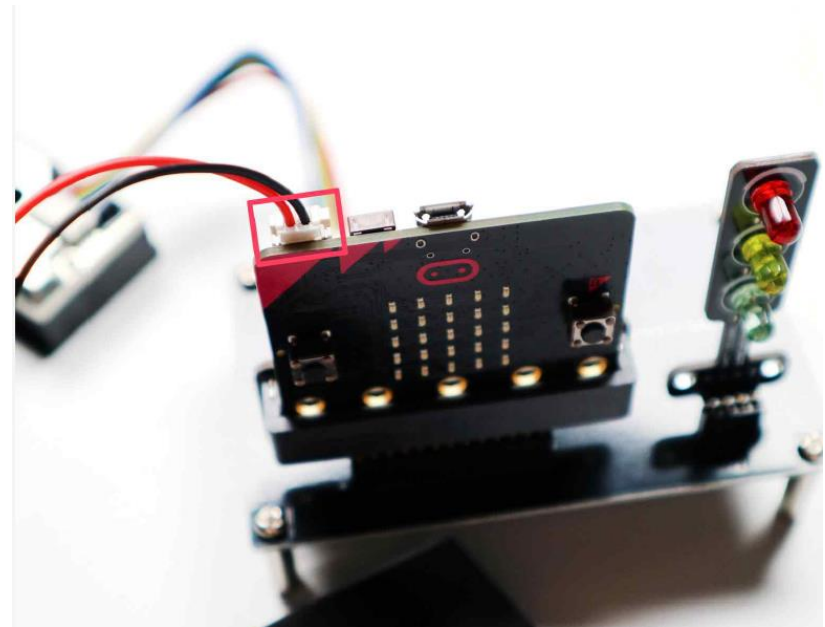
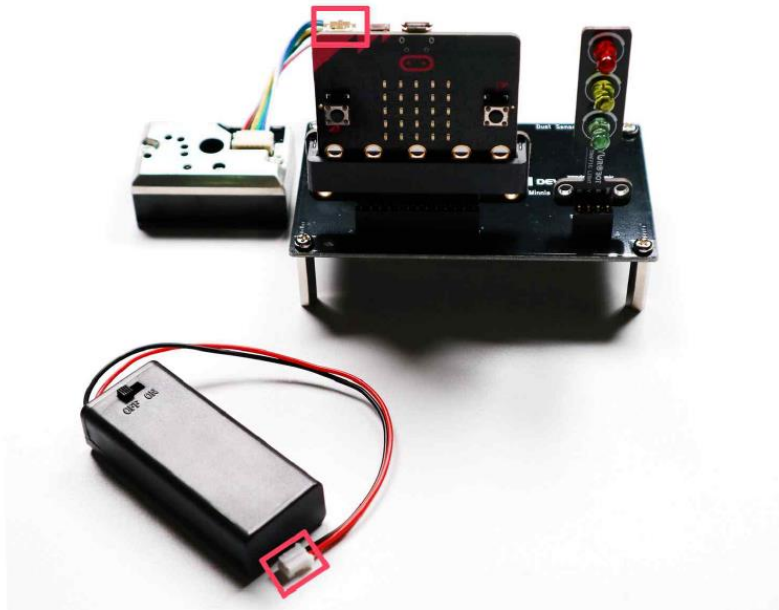
# 미세먼지측정기 조립하기



마이크로비트와 신호등 모듈을 미세먼지 PCB보드에 장착된 확장보드와 4핀 핀헤더 소켓에 각각 결합합니다



# 미세먼지측정기 조립하기

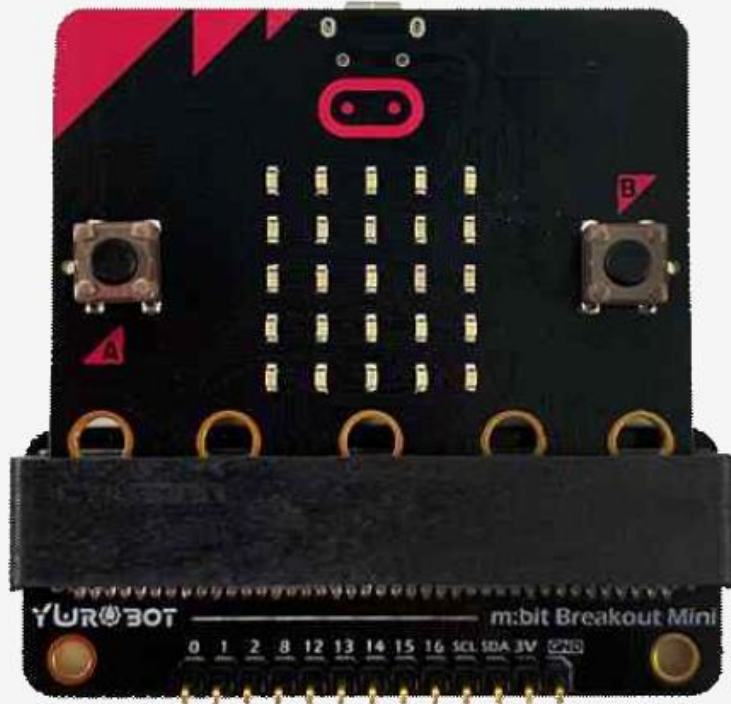


배터리 홀더를 마이크로비트에 연결합니다



코딩하기

# 보드 핀맵 이해하기



P0 → 미세먼지 센서

P1 → 미세먼지 센서

P2 → 신호등 빨간불

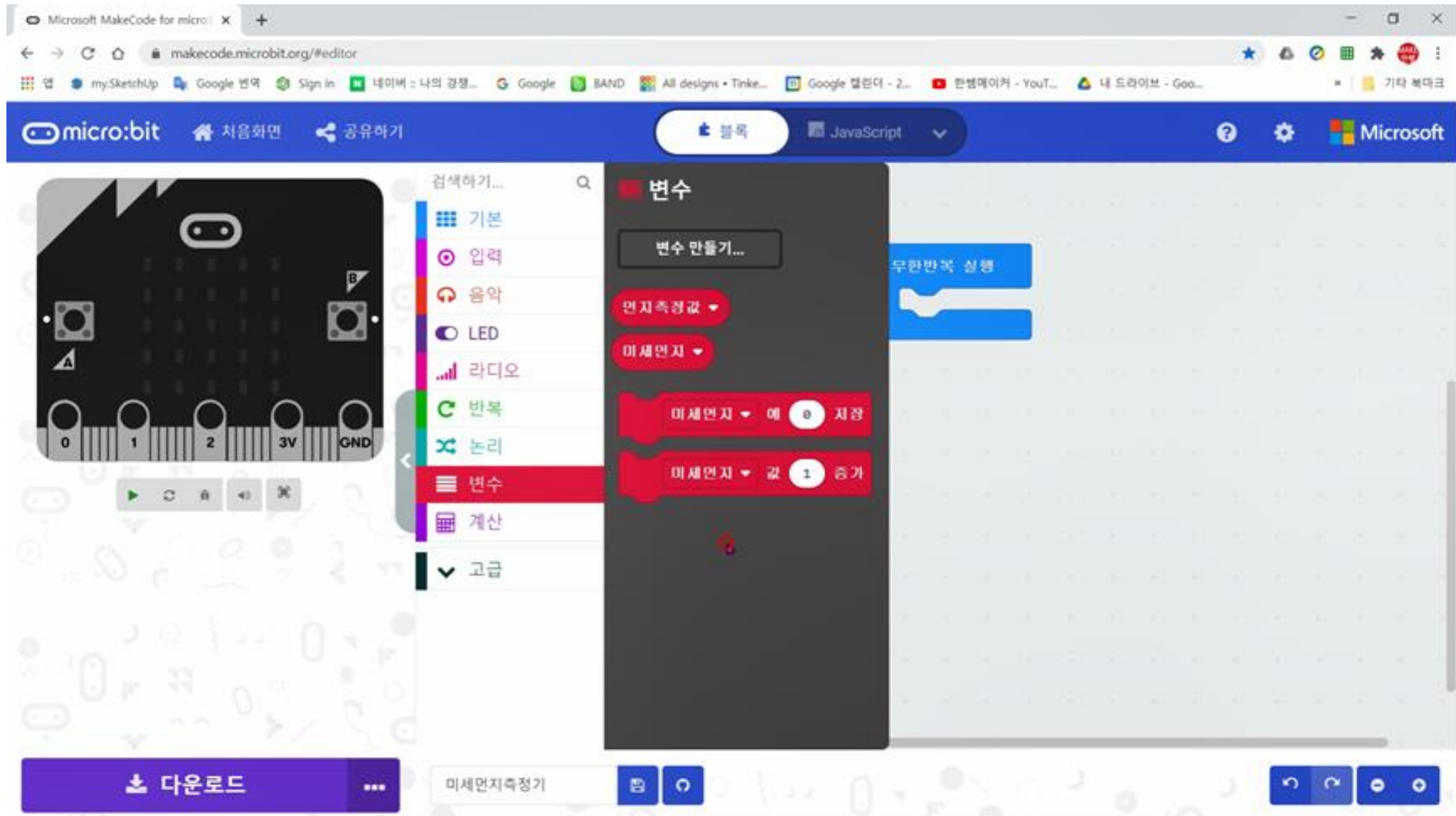
P8 → 신호등 노란불

P12 → 신호등 초록불

# 프로젝트를 시작

The image shows a screenshot of the Microsoft MakeCode for micro:bit website. The browser address bar shows the URL `makecode.microbit.org/#`. The page features a blue header with the 'micro:bit' logo and a 'Microsoft' logo. A central banner displays a micro:bit board with various components and code blocks like 'on shake' and 'show leds'. A white dialog box titled 'Create a Project 😊' is overlaid on the page. It prompts the user to 'Give your project a name.' with a text input field containing the Korean text '미세먼지측정기'. Below the input field is a 'Code options' link. A green 'Create' button with a checkmark is at the bottom right of the dialog. The background shows a '내 프로젝트' (My Projects) section with a '새 프로젝트' (New Project) button and a list of existing projects.

# 먼지측정값, 미세먼지 2개의 변수를 생성

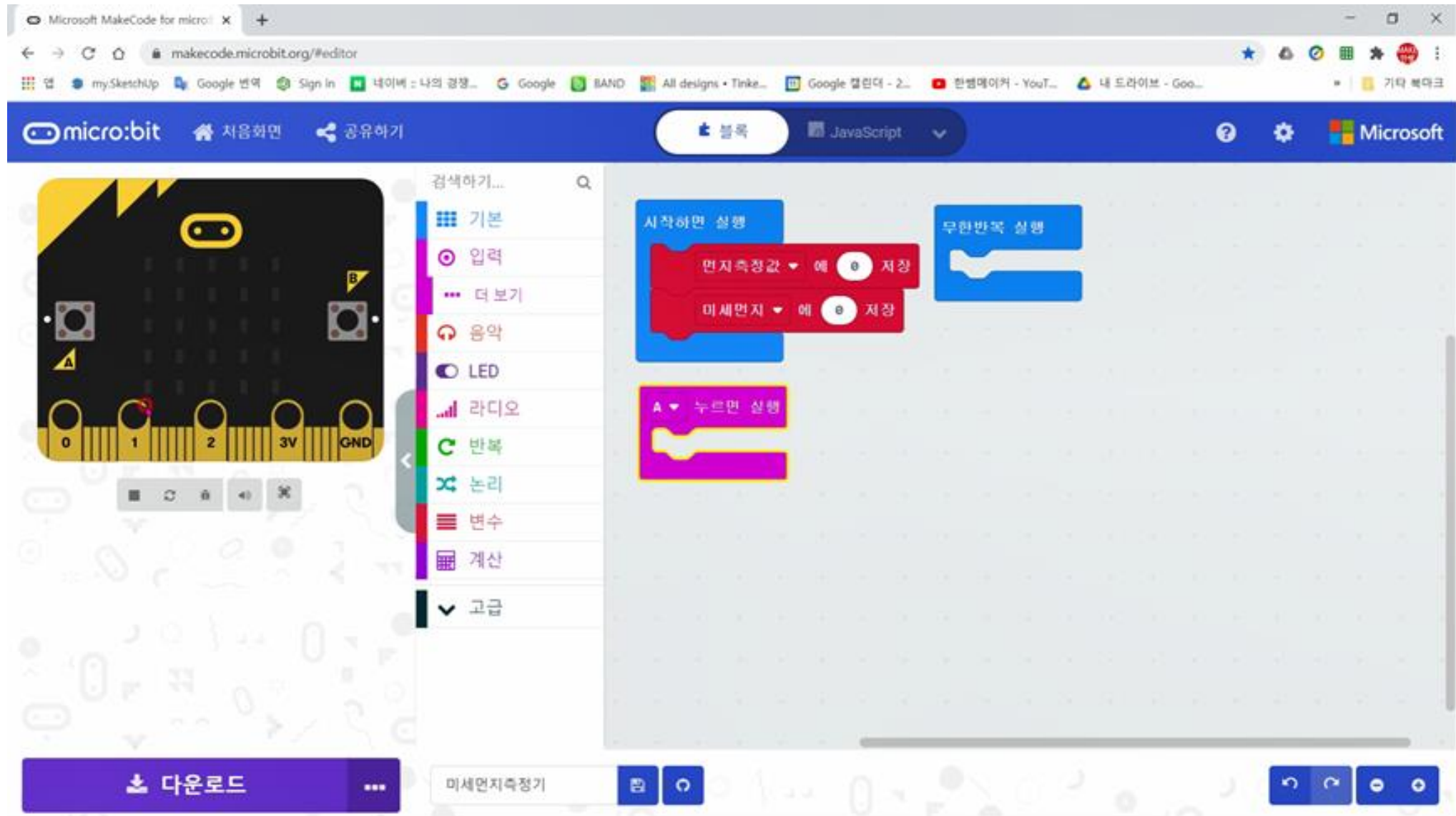


# 먼지측정값,미세먼지에 저장 삽입

The image shows a screenshot of the Microsoft MakeCode for micro:bit editor. The browser address bar displays 'makecode.microbit.org/#editor'. The interface includes a top navigation bar with 'micro:bit', '처음화면', '공유하기', '블록', and 'JavaScript' options. On the left, there is a virtual micro:bit board and a sidebar menu with categories like '기본', '입력', '음악', 'LED', '라디오', '반복', '논리', '변수', '계산', and '고급'. The main workspace contains a script starting with '시작하면 실행' (When green flag clicked), followed by two '저장' (Save) blocks: '먼지측정값' (Dust sensor value) and '미세먼지' (PM2.5), both set to save to '메모리' (Memory). A '무한반복 실행' (Repeat forever) block is also visible. At the bottom, there is a '다운로드' (Download) button and a search bar containing '미세먼지측정기'.



# 누르면 실행 블록 개념





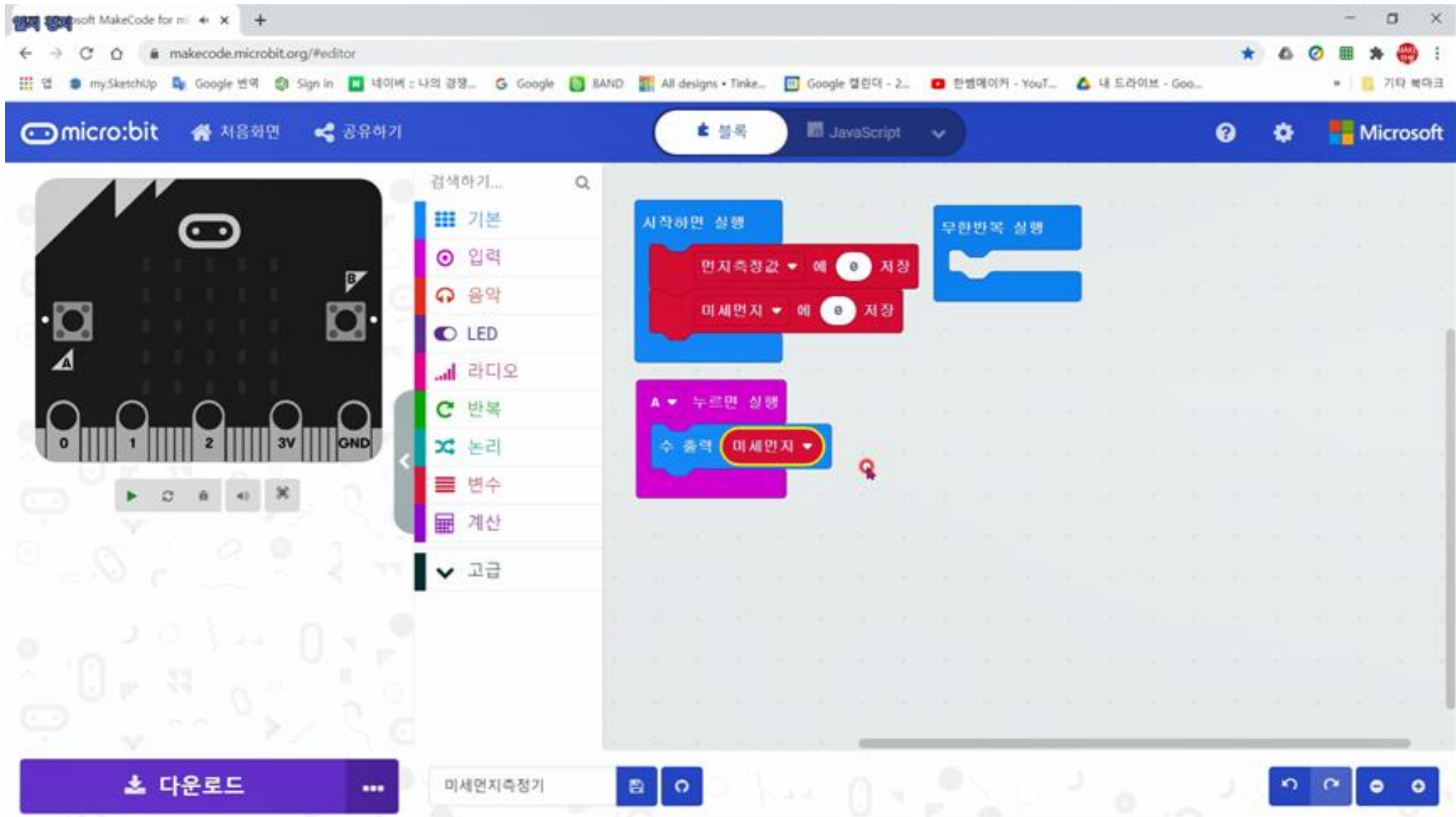
# 수 출력을 누르면 실행에 삽입

The image shows a screenshot of the Microsoft MakeCode for micro:bit editor. The browser address bar displays 'makecode.microbit.org/#editor'. The interface includes a top navigation bar with 'micro:bit', '처음화면', '공유하기', '블록', and 'JavaScript' options. On the left, there is a visual representation of the micro:bit board with pins labeled 0, 1, 2, 3V, and GND. A central menu lists various block categories: 기본, 입력, 음악, LED, 라디오, 반복, 논리, 변수, 계산, and 고급. The main workspace contains a block-based program with the following structure:

- 시작하면 실행** (When green flag clicked):
  - 먼저촉정값에 저장** (Store first touch value)
  - 미세먼지에 저장** (Store PM2.5)
- A 누르면 실행** (When button A pressed):
  - 수 출력** (Show output)
- 무한반복 실행** (Forever loop)

At the bottom, there is a '다운로드' (Download) button and a search bar containing '미세먼지촉정기'.

# 미세먼지를 수 출력에 삽입



# 계산에서 반올림을 꺼내서 미세먼지 삽입

The image shows the Microsoft MakeCode for micro:bit editor interface. The main workspace contains a Scratch-style block-based program. The program starts with a '시작하면 실행' (When started) block, followed by a '먼지측정값' (Dust measurement value) block with a dropdown menu set to '미세먼지' (PM2.5) and a '저장' (Store) block. This is followed by another '미세먼지' (PM2.5) block and a '무한반복 실행' (Repeat forever) block. The '무한반복 실행' block contains a '누르면 실행' (When button A pressed) block, which in turn contains a '수 출력' (Number output) block. The '수 출력' block has a dropdown menu set to '반올림(round)' (Round) and a dropdown menu set to '미세먼지' (PM2.5). The interface includes a sidebar with categories like '기본' (Basic), '입력' (Input), '음악' (Music), 'LED', '라디오' (Radio), '반복' (Loops), '논리' (Logic), '변수' (Variables), '계산' (Math), and '고급' (Advanced). The bottom bar shows a '다운로드' (Download) button and a search bar for '미세먼지측정기' (PM2.5 sensor).

# 핀에서 P1에 디지털값 0 출력 삽입

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The left sidebar contains a list of blocks: 라디오, 반복, 논리, 변수, 계산, 고급, 함수, 배열, 문자열, 게임, 이미지, 핀, 더 보기, 시리얼통신, and 고급제어. The main workspace displays a script with the following blocks:

- 시작하면 실행** (When green flag clicked):
  - 핀지속장치 (P1) 에 0 저장 (Set pin P1 to 0)
  - 미세먼지 (P1) 에 0 저장 (Set pin P1 to 0)
- 무한반복 실행** (Forever loop):
  - 핀 (P1) 에 디지털 값 0 출력 (Output digital value 0 to pin P1)
- A 누르면 실행** (When button A pressed):
  - 수 출력 (Output number) - 반응값 (round) - 미세먼지 (P1) (Output round reaction value to pin P1)

The bottom of the editor shows a '다운로드' (Download) button and a search bar for '미세먼지측정기' (Air quality sensor).



# 고급제어에서 일시중지 꺼냄

The image shows a screenshot of the Microsoft MakeCode for micro:bit editor. The interface is in Korean. On the left, there is a virtual micro:bit board. In the center, a block menu is visible with categories like '라디오' (Radio), '반복' (Loop), '논리' (Logic), '변수' (Variables), '계산' (Math), '고급' (Advanced), '함수' (Functions), '배열' (Arrays), '문자열' (Strings), '게임' (Games), '이미지' (Images), '핀' (Pins), '시리얼통신' (Serial Communication), and '고급제어' (Advanced Control). The '고급제어' block is selected, and its sub-menu is open, showing various control blocks. One block, '일시중지' (Pause), is highlighted with a yellow box. The value '4' is entered in the input field next to the block, and the unit '(µs)' is shown. The background of the code editor is dark grey, and the overall interface has a blue header with the 'micro:bit' logo and navigation options.

# 일시중지 280을 삽입

The image shows a screenshot of the Microsoft MakeCode for micro:bit web editor. The interface includes a top navigation bar with the 'micro:bit' logo, a home button, a share button, a '블록' (Blocks) button, a 'JavaScript' dropdown menu, and a 'Microsoft' logo. Below the navigation bar is a sidebar with various tool categories in Korean: 라디오 (Radio), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), 고급 (Advanced), 함수 (Functions), 배열 (Arrays), 문자열 (Strings), 게임 (Games), 이미지 (Images), 핀 (Pins), 시리얼통신 (Serial Communication), 고급제어 (Advanced Control), and 더 보기 (More). The main workspace contains three code blocks: a '시작하면 실행' (When green flag clicked) block with two '저장' (Set) blocks for '먼지측정값' (Dust sensor value) and '미세먼지' (PM2.5) to 0; a '무한반복 실행' (Forever loop) block with a 'P1에 디지털 값 출력' (Digital value output to P1) block and an '일시중지 280 (µs)' (Delay 280 µs) block; and an 'A 누르면 실행' (When button A pressed) block with a '수 출력' (Output number) block set to '반올림 (round)' (Round) and '미세먼지' (PM2.5). The bottom of the editor features a '다운로드' (Download) button, a search bar for '미세먼지측정기' (PM2.5 sensor), and several utility buttons.

# 먼지측정값에 P0아날로그 입력값 저장 삽입

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The code is written in a block-based language and consists of the following blocks:

- 시작하면 실행 (When started):**
  - 먼지측정값에 0 저장 (Store 0 to dust sensor value)
  - 미세먼지 0에 저장 (Store 0 to PM2.5)
- A 누르면 실행 (When button A pressed):**
  - 수 출력 반올림(round)에 미세먼지 (Output rounded dust sensor value)
- 무한반복 실행 (Forever loop):**
  - P1에 디지털 값 0 출력 (Output digital value 0 to P1)
  - 일시중지 200 (μs) (Wait 200 microseconds)
  - 먼지측정값에 P0의 아날로그 입력 값 저장 (Store P0 analog input value to dust sensor value)
  - 일시중지 40 (μs) (Wait 40 microseconds)

A green callout bubble points to the 40 (μs) delay block with the text "일시중지 40".



# P1에 디지털값 1 출력 삽입, 일시중지 9680

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The code is written in JavaScript and is organized into three main sections:

- 시작하면 실행 (When started):** This section contains two blocks to initialize variables: "변수 지정" (Set variable) for "먼저측정값" (First measurement) to 0, and "변수 지정" for "미세먼지" (PM2.5) to 0.
- A 누르면 실행 (When button A pressed):** This section contains one block: "수 출력" (Output number) with the value "반올림(round)" (Round) and the variable "미세먼지" (PM2.5).
- 무한반복 실행 (Forever loop):** This section contains a loop of four blocks:
  - "P1에 디지털 값 출력" (Output digital value to P1) with the value 0.
  - "일시중지" (Pause) for 280  $\mu$ s.
  - "변수 지정" (Set variable) for "먼저측정값" (First measurement) to "P0의 아날로그 입력 값" (Analog input value of P0).
  - "일시중지" (Pause) for 40  $\mu$ s.
  - "P1에 디지털 값 출력" (Output digital value to P1) with the value 1.
  - "일시중지" (Pause) for 9680  $\mu$ s.

The interface includes a left sidebar with category icons (라디오, 반복, 논리, 변수, 계산, 고급, 함수, 배열, 문자열, 게임, 이미지, 핀, 시리얼통신, 고급제어, 더 보기) and a top navigation bar with "micro:bit", "시용화면", "공유하기", "블록", "JavaScript", and "Microsoft" logos. The bottom bar features a "다운로드" (Download) button, a "미세먼지측정기" (PM2.5 sensor) label, and several control icons.

# 미세먼지에 비례 변환한 먼지측정값 저장

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The browser address bar displays `makecode.microbit.org/#editor`. The left sidebar contains various block categories: 라디오, 반복, 논리, 변수, 계산, 고급, 함수, 배열, 문자열, 게임, 이미지, 핀, 시리얼통신, 고급제어, and 확장. The main workspace contains a JavaScript script with the following blocks:

- 무한반복 실행 (Infinite loop)
- P1에 디지털 값 0 출력 (Output digital value 0 to P1)
- 일시 중지 280 (μs) (Delay 280 μs)
- 먼지측정값에 P0의 아날로그 입력 값 저장 (Store P0's analog input value in dust measurement value)
- 일시 중지 40 (μs) (Delay 40 μs)
- P1에 디지털 값 1 출력 (Output digital value 1 to P1)
- 일시 중지 9680 (μs) (Delay 9680 μs)
- 미세먼지에 비례 변환 (map): 먼지측정값을 0 ~ 1023에서 0 ~ 300 범위로 변환한 값 저장 (Store converted dust measurement value proportional to fine dust: convert dust measurement value from 0 ~ 1023 to 0 ~ 300 range)

The bottom of the editor features a '다운로드' (Download) button, a search bar containing '미세먼지측정기', and navigation controls.

# 논리에서 if~else 블록 꺼내서 삽입

The screenshot shows the Microsoft MakeCode editor for micro:bit. The interface includes a top navigation bar with the 'micro:bit' logo, a home button, a share button, and a dropdown menu currently set to 'JavaScript'. Below this is a sidebar with various block categories: 라디오, 반복, 논리, 변수, 계산, 고급, 함수, 배열, 문자열, 게임, 이미지, 핀, 시리얼통신, 고급제어, and 확장. The main workspace contains a sequence of blocks: a '실행' block, a '반올림(round)' block with '미세먼지' selected, a '일시 중지' block (40 μs), a '출력' block (P1 에 디지털 값 1 출력), another '일시 중지' block (9680 μs), a '미세먼지' block with a '비례 변환(map)' block inside it, and an 'if-else' block. The 'if-else' block has '참(true)' selected for the '만약(if)' condition, and the '이면(then) 실행' block is currently being inserted into the workspace. The '아니면(else) 실행' block is also visible. At the bottom, there is a '다운로드' button and a search bar containing '미세먼지측정기'.

# P2, P8, P12에 0, 0, 1 출력을 삽입

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The main workspace contains a JavaScript script with the following blocks:

- 실행** (Run) block
- 반올림 (round)** block with **미세먼지** (PM2.5) dropdown
- 일시 중지** (Delay) block with **40** ( $\mu$ s)
- P1**에 디지털 값 **1** 출력 (Digital output block for pin P1)
- 일시 중지** (Delay) block with **9680** ( $\mu$ s)
- 미세먼지**에 **비례 변환 (map)**: 먼지측정값 을 **0** ~ **1023** 에서 **0** ~ **300** 범위로 변환한 값 저장 (Map block)
- 만약 (if) 참 (true) 이면 (then) 실행** (If-then block)
  - P2**에 디지털 값 **0** 출력 (Digital output block for pin P2)
  - P8**에 디지털 값 **0** 출력 (Digital output block for pin P8)
  - P12**에 디지털 값 **1** 출력 (Digital output block for pin P12)
- 아니면 (else) 실행** (Else block)

The bottom of the editor shows a **다운로드** (Download) button, a search bar containing **미세먼지측정기**, and navigation controls.



# 논리에서 꺼내서 미세먼지 $\leq 75$ 부등식 작성

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The code is written in JavaScript and includes the following blocks:

- 실행 (Run)** block
- 반올림 (round)** block with **미세먼지 (PM10)** selected
- 일시 중지 (Delay)** block set to 40 ( $\mu$ s)
- 출력 (Output)** block for **P1** with value **1**
- 일시 중지 (Delay)** block set to 9680 ( $\mu$ s)
- 비례 변환 (map)** block: **미세먼지 (PM10)** to **먼지측정값 (Dust)**, range **0 ~ 1023** to **0 ~ 300**
- 만약 (if)** block: **미세먼지 (PM10)  $\leq 75$  이면 (then) 실행**
- 출력 (Output)** blocks for **P2** (0), **P8** (0), and **P12** (1)
- 아니면 (else) 실행** block

The bottom of the editor shows a **다운로드 (Download)** button, a search bar containing **미세먼지측정기**, and navigation controls.

# 미세먼지 ≤ 150 부등식 작성, P8이 1출력

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The script is as follows:

- 만약 (if) 블록:** 조건은 "미세먼지 ≤ 75". 이 블록의 실행 내용은:
  - P2에 디지털 값 0 출력
  - P8에 디지털 값 0 출력
  - P12에 디지털 값 1 출력
- 아니면서 만약 (else if) 블록:** 조건은 "미세먼지 ≤ 150". 이 블록의 실행 내용은:
  - P2에 디지털 값 0 출력
  - P8에 디지털 값 1 출력 (이 블록이 선택되어 있음)
  - P12에 디지털 값 0 출력
- 아니면 (else) 블록:** 실행 내용은 비어 있음.

The interface includes a left sidebar with categories like "라디오", "반복", "논리", "변수", "계산", "고급", "함수", "배열", "문자열", "게임", "이미지", "핀", "시리얼통신", "고급제어", and "확장". The top navigation bar shows "micro:bit", "처음화면", "공유하기", "블록", and "JavaScript". The bottom bar contains a "다운로드" button, a search bar with "미세먼지측정기", and navigation controls.

# P12만 1출력으로 이미지와 같이 작성

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The main workspace contains a JavaScript script with the following logic:

```
만약(if) <미세먼지> ≤ 75 이면(then) 실행
  P2 에 디지털 값 0 출력
  P8 에 디지털 값 0 출력
  P12 에 디지털 값 1 출력

아니면서 만약(else if) <미세먼지> ≤ 150 이면(then) 실행
  P2 에 디지털 값 0 출력
  P8 에 디지털 값 1 출력
  P12 에 디지털 값 0 출력

아니면(else) 실행
  P2 에 디지털 값 1 출력
  P8 에 디지털 값 0 출력
  P12 에 디지털 값 0 출력
```

The P12 output block in the 'else' section is highlighted with a yellow border. The left sidebar shows various block categories like '라디오', '반복', '논리', '변수', '계산', '고급', '함수', '배열', '문자열', '게임', '이미지', '핀', '시리얼통신', '고급제어', and '확장'. The bottom status bar includes a '다운로드' button, a search bar for '미세먼지측정기', and navigation controls.



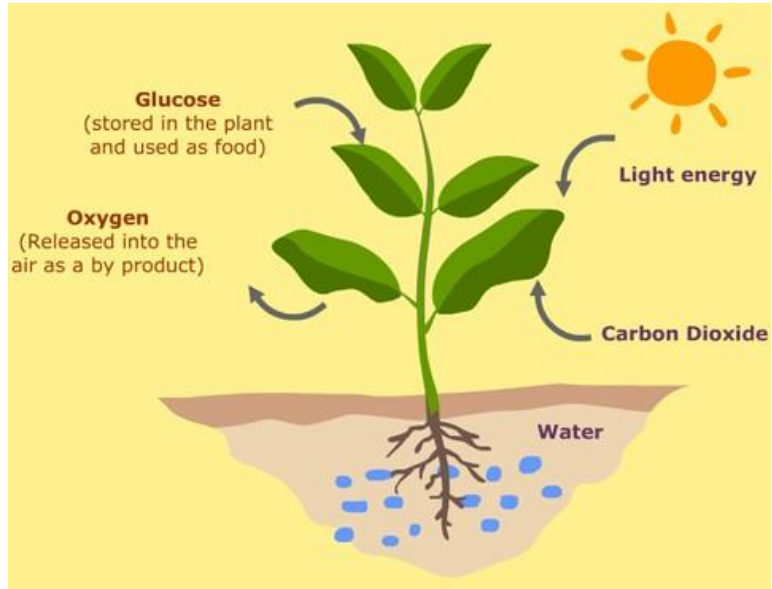
# 아이콘 출력을 3곳에 다른 표정으로 삽입

The screenshot shows the Microsoft MakeCode for micro:bit editor interface. The left sidebar contains a search bar and a list of categories: 기본 (Basic), 입력 (Input), 음악 (Music), LED, 라디오 (Radio), 반복 (Loops), 논리 (Logic), 변수 (Variables), 계산 (Math), 고급 (Advanced), 함수 (Functions), 배열 (Arrays), 문자열 (Strings), and 게임 (Games). The main workspace displays a script with three '아이콘 출력' (Icon Output) blocks. The first block is set to 'P2' with a value of '0'. The second block is set to 'P8' with a value of '1'. The third block is set to 'P12' with a value of '0'. These blocks are connected to an '아니면 (else) 실행' (Else Execute) block, which is preceded by an '아니면서 만약 (else if) <미세먼지> ≤ 150 이면 (then) 실행' (Else if <PM2.5> ≤ 150 then Execute) block. The bottom of the editor features a '다운로드' (Download) button, a search bar for '미세먼지측정기' (PM2.5 sensor), and various utility icons.

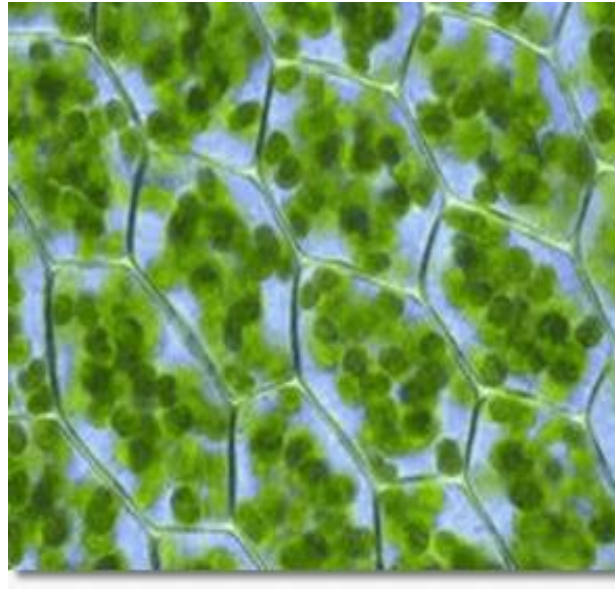
# 스마트 식물재배기 스테디

식물이 자라기 위해 필요한 것

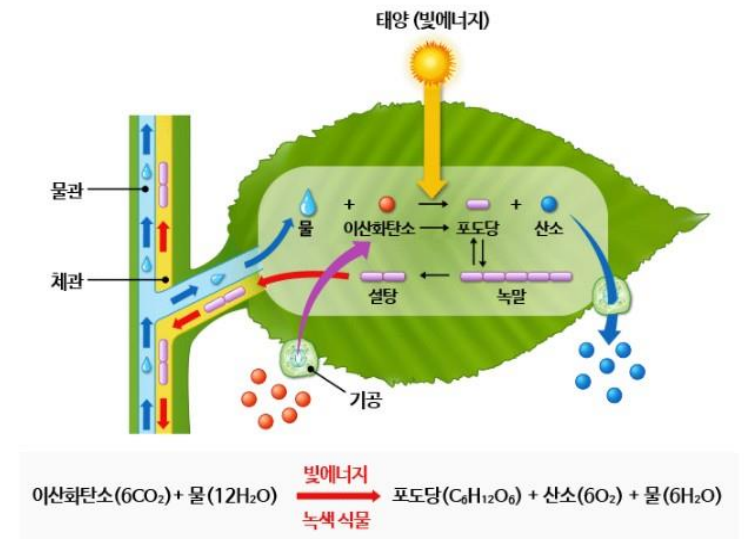
# 식물이 자라기 위해 필요한 것 이해하기



식물과 환경

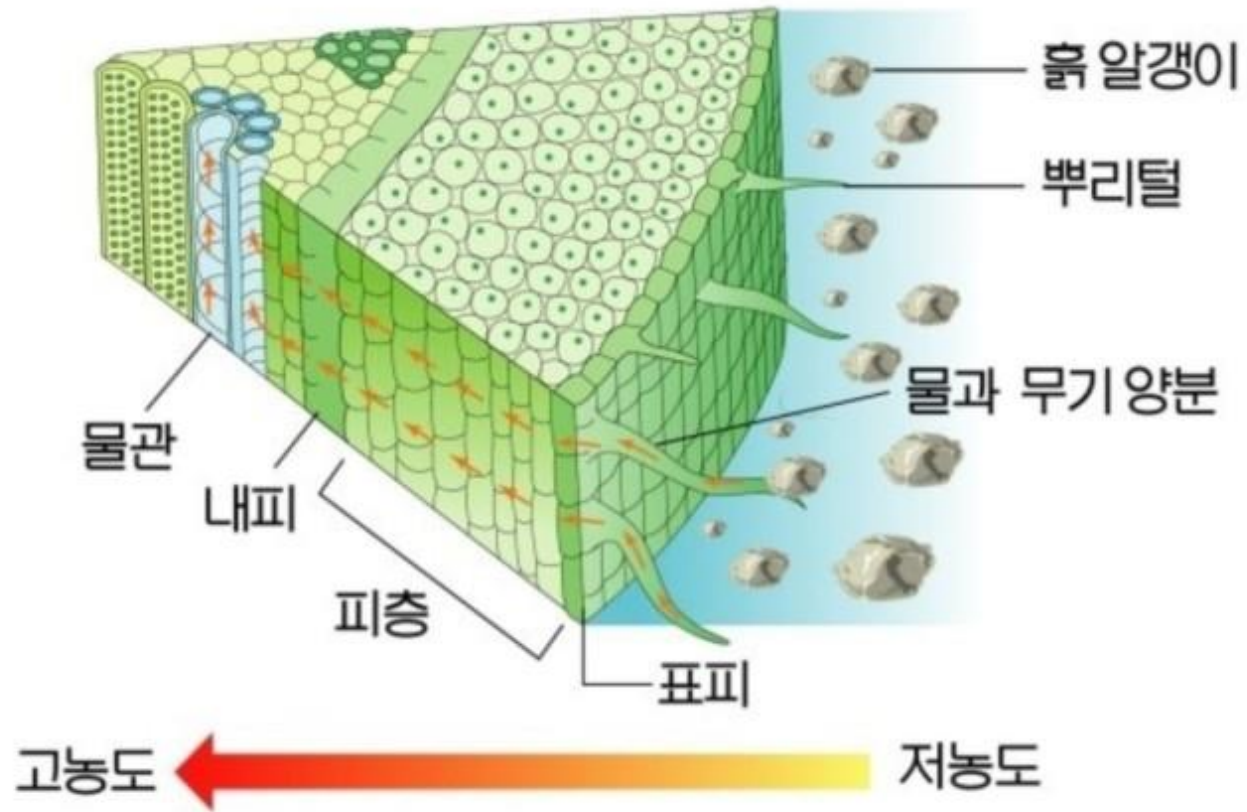


엽록소 확대사진



광합성 과정

# 식물 세포구조



# 스마트 화분의 의미

- 미세먼지 등 심각한 환경오염으로 인한 친환경공기정화기에 대한 수요가 전세계적으로 대두되고 있음
- 핀란드의 나바아(NAAVA)시스템이 대표적이며, 국내에서도 인공지능을 활용한 다양한 친환경 공기정화기 시스템 개발을 위한 연구가 활발히 진행되고 있음
- 메이커용 도구인 아두이노, 마이크로비트와 WiFi연결이 가능한 ESP8266을 활용하여 원격환경측정장치 개발이 가능하며, 실제로 옥천주택에서 활용되고 있음
- 차후 환경을 센싱해서 원격으로 그 데이터를 수신하여 활용하는 연구를 지속하고, 교재 및 교구를 제작하여 공급할 예정임

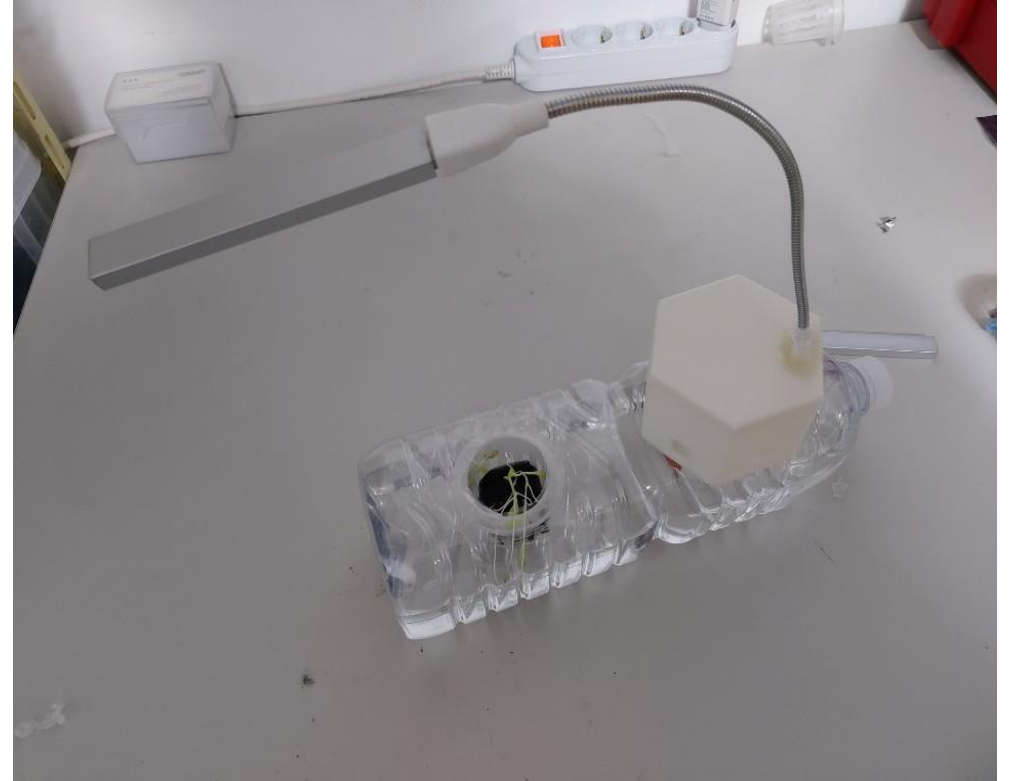
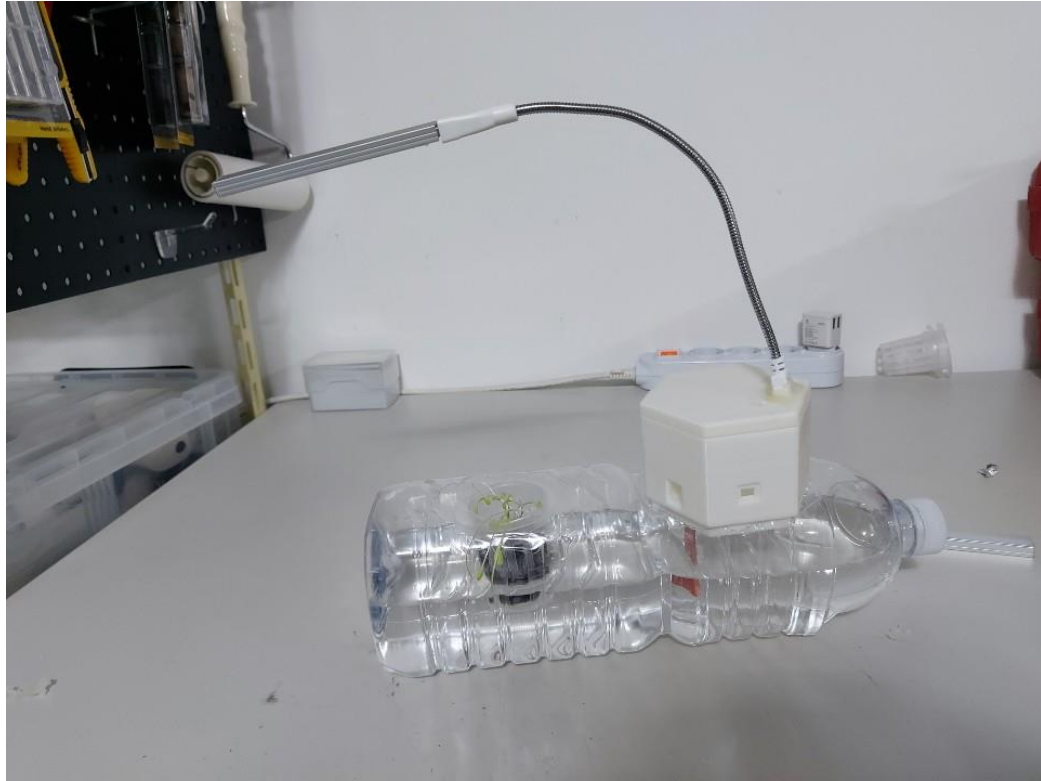


스마트화분: BLOCKFARM

# 블록팜 특징

- 기업체가 아닌 메이커들이 아두이노라는 보드와 각종 부품을 활용해서 만든 제품
- 3D프린팅과 페트병 재활용을 통해 학생들이 스스로 만들 수 있도록 구성 됨
- 블루투스를 활용해서 스마트폰으로 빛과 진동모터의 제어가 가능하도록 함
- 코딩은 오픈소스인 아두이노 스케치와 앱인벤터를 사용
- 부품 및 코딩, 제작방법 등이 모두 공유 됨

# 수경식물재배기(3D프린팅+자원재활용)



# 부품구성

## 준비물

수위센서



x1

식물LED



x1

점퍼 케이블  
(10cm 암수)



x20

## 준비물

미니 브레드보드



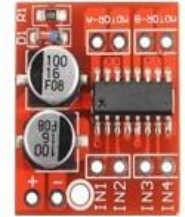
x1

USB Mini  
케이블



x1

2채널 PWM  
모터 드라이버



x1

HC-06  
블루투스 모듈



x1

실리콘 호스



x1

기포기 콩돌



x1

아두이노 나노



x1

3V 모터 펌프



x1

USB 암컷 포트

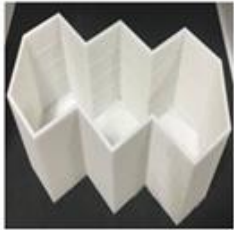


x1

# 부품구성

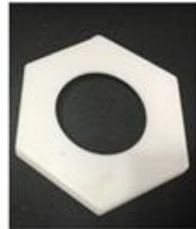
## 준비물

스마트팜 3구  
POT 파트



x1

포트 거치 뚜껑



x2

수경재배 포트



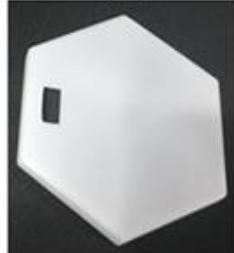
x2

아두이노  
제어부 케이스



x1

제어부 케이스  
뚜껑



x1

수경재배 식물  
스펀지



x2

## 준비물

DHT11 온습도  
센서



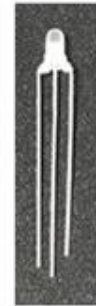
x1

LED 자바라



x1

2색 LED



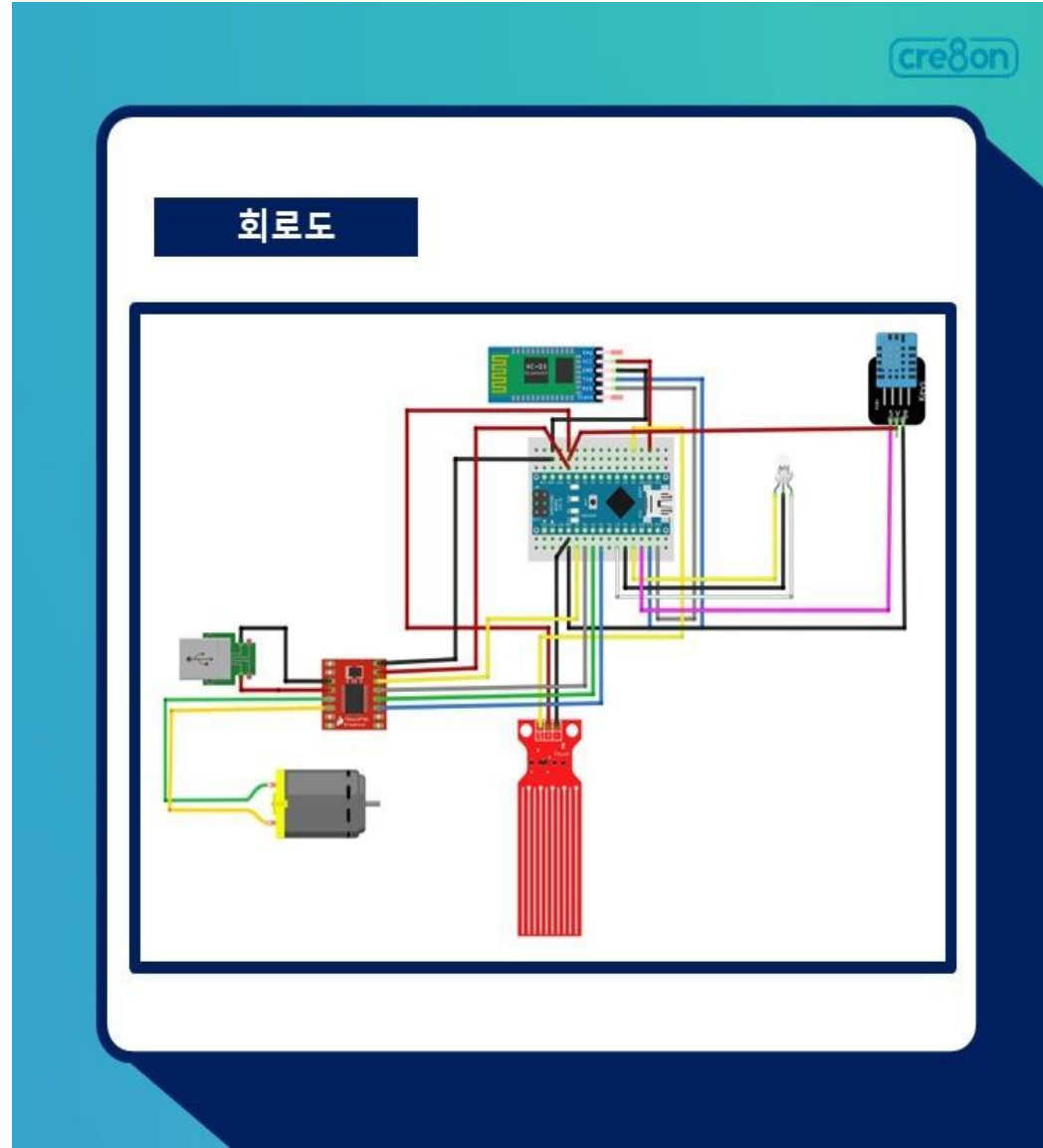
x1

cre8on

cre8on

cre8on

# 회로도





# 부품 연결



# 부품구성

cre8on

## 오픈소스 소프트웨어 코딩(1)

```
#include <SoftwareSerial.h>
#include "DHT.h"

#define UV_LEDON 4
#define UV_LEDOFF 5
#define state_LED 8
#define RED 7
#define GREEN 9
#define Pump_ON 3
#define Pump_OFF 2
#define water A0

#define DHTPIN 10 // Digital pin connected to the DHT
sensor
// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12,
13 or 14 --
// Pin 15 can work but DHT must be disconnected during
program upload.
```

cre8on

## 오픈소스 소프트웨어 코딩(2)

```
// Uncomment whatever type you're using!
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22 (AM2302),
AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

SoftwareSerial bluetooth(11,12);
DHT dht(DHTPIN, DHTTYPE);

void setup() {

    pinMode(UV_LEDON,OUTPUT); //핀모드 설정
    pinMode(UV_LEDOFF,OUTPUT);
    pinMode(state_LED,OUTPUT);
    pinMode(RED,OUTPUT);
    pinMode(GREEN,OUTPUT);
    //pinMode(Pump_ON,OUTPUT);
    pinMode(Pump_OFF,OUTPUT);
```

# 부품구성

cre8on

## 오픈소스 소프트웨어 코딩(4)

```
//pinMode(Pump_ON,OUTPUT);  
pinMode(Pump_OFF,OUTPUT);  
  
digitalWrite(Pump_OFF,LOW); //초기 핀출력 설정  
digitalWrite(UV_LED,LOW);  
digitalWrite(state_LED,LOW);  
  
analogWrite(Pump_ON,255); //초기 핀출력 설정  
digitalWrite(UV_LED,HIGH);  
delay(500);  
digitalWrite(Pump_ON,LOW); //초기 핀출력 설정  
// digitalWrite(UV_LED,LOW);  
  
Serial.begin(9600);  
Serial.println(F("DHTxx test!"));  
bluetooth.begin(9600); //블루투스 통신 속도  
dht.begin();
```

cre8on

## 오픈소스 소프트웨어 코딩(5)

```
}  
int water_state=0;  
int water_value=50; //수위설정  
int receive;  
int Temp_value=27; //온도 설정  
bool LED=0; //UV LED상태  
bool motor=0; //펌프 모터 상태  
unsigned long timer;  
  
//송신 바이트  
byte water_m = 1; //수위 퍼센트  
byte humidity = 0; //습도 퍼센트  
byte temperature = 0; //  
byte state = 0;
```

# 부품구성

cre8on

## 오픈소스 소프트웨어 코딩(6)

```
//알람 동작 함수
void alarm()
{
  digitalWrite(GREEN,LOW);
  timer=millis()%1000;

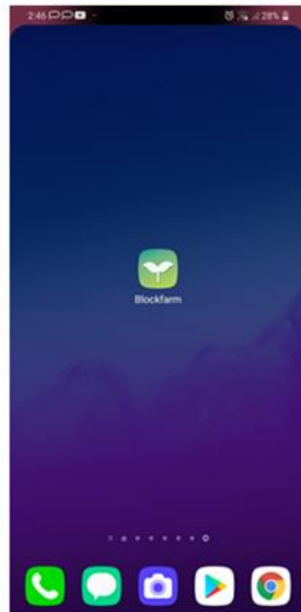
  if(timer<499)
  {
    digitalWrite(RED,HIGH);
  }
  else
  {
    digitalWrite(RED,LOW);
  }
}
```

코딩 소스가 너무 길어서 생략합니다!  
첨부된 코딩 소스 파일을 활용하세요 :)

# 식물재배기 앱

cre8on

## 적정기술 오픈소스 스마트 식물재배기 앱



cre8on

### 프로젝트 요약

스마트 식물재배기 앱 만들기

1. 앱인벤터로 제작한 안드로이드용 어플리케이션
2. 식물용 조명, 기포 발생기, 상태 표시 등을 제어할 수 있습니다
3. 온습도센서, 수위센서를 통해 환경을 인식한 후 캐릭터가 상태에 따라 변화합니다
4. 상태에 맞는 정보가 어플에 나타납니다

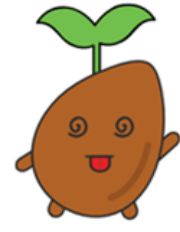
물이 많을 때: 어푸어푸 물이 많아요

물이 적을 때: 목말라요

온도가 높을 때: 더워요

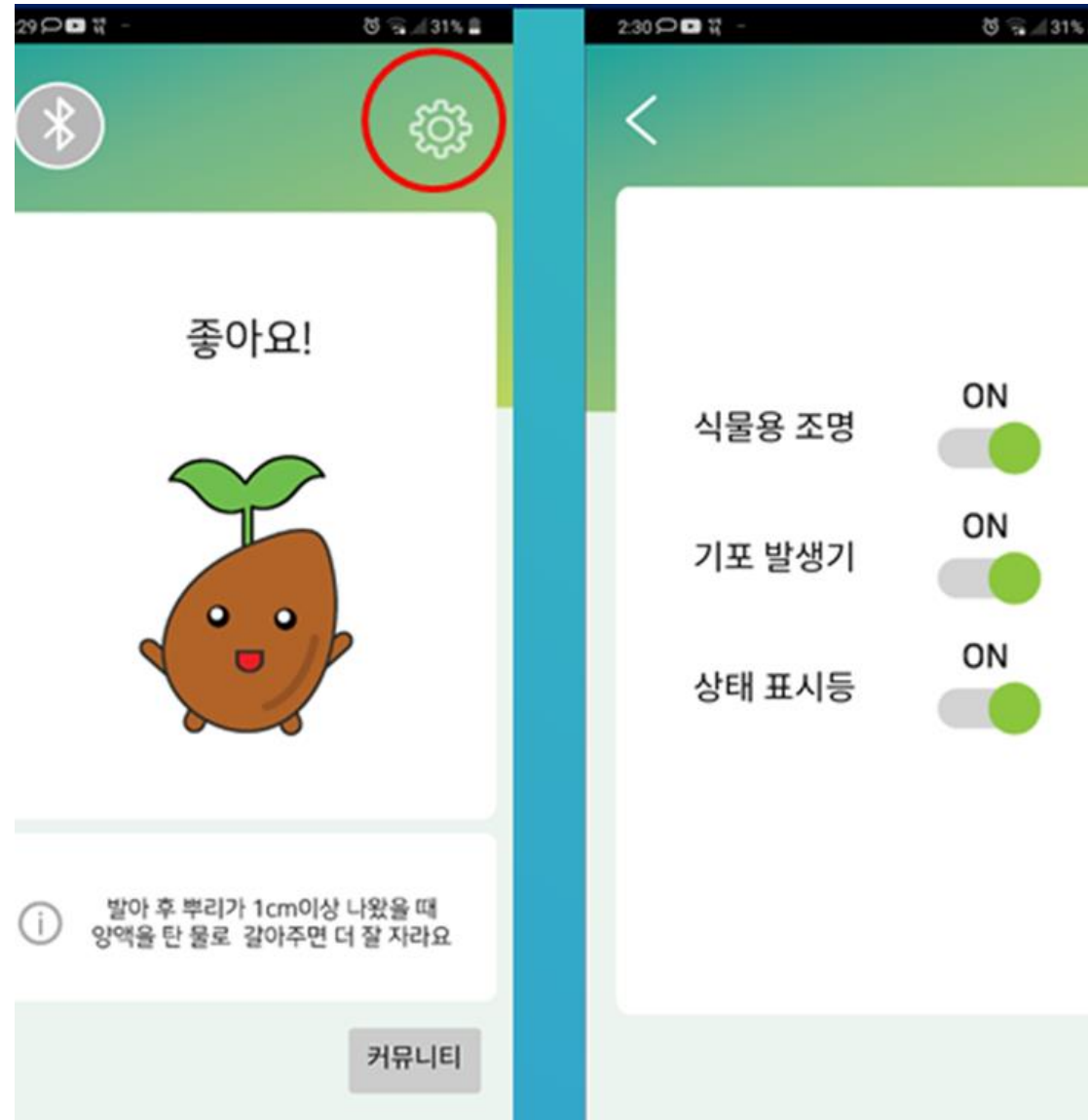
온도가 낮을 때: 추워요

건조할 때 : 건조해요





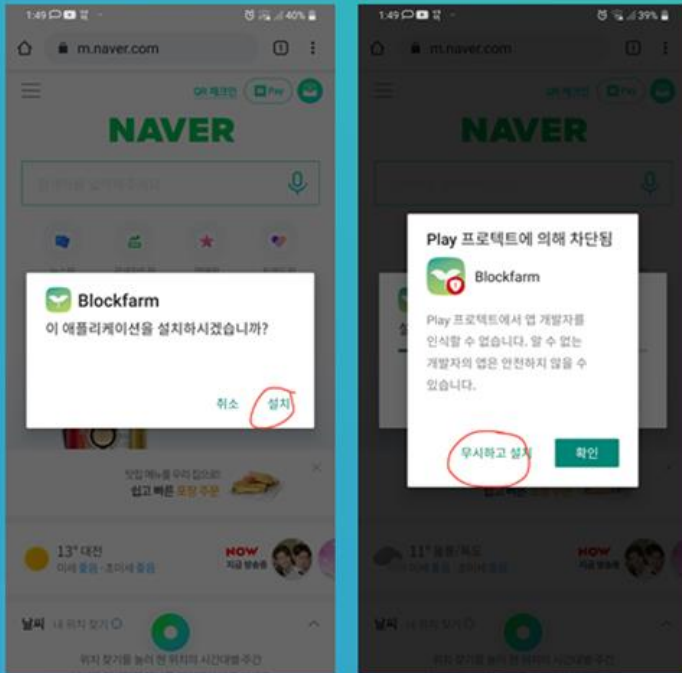
# 식물재배기 앱



# 식물재배기 앱

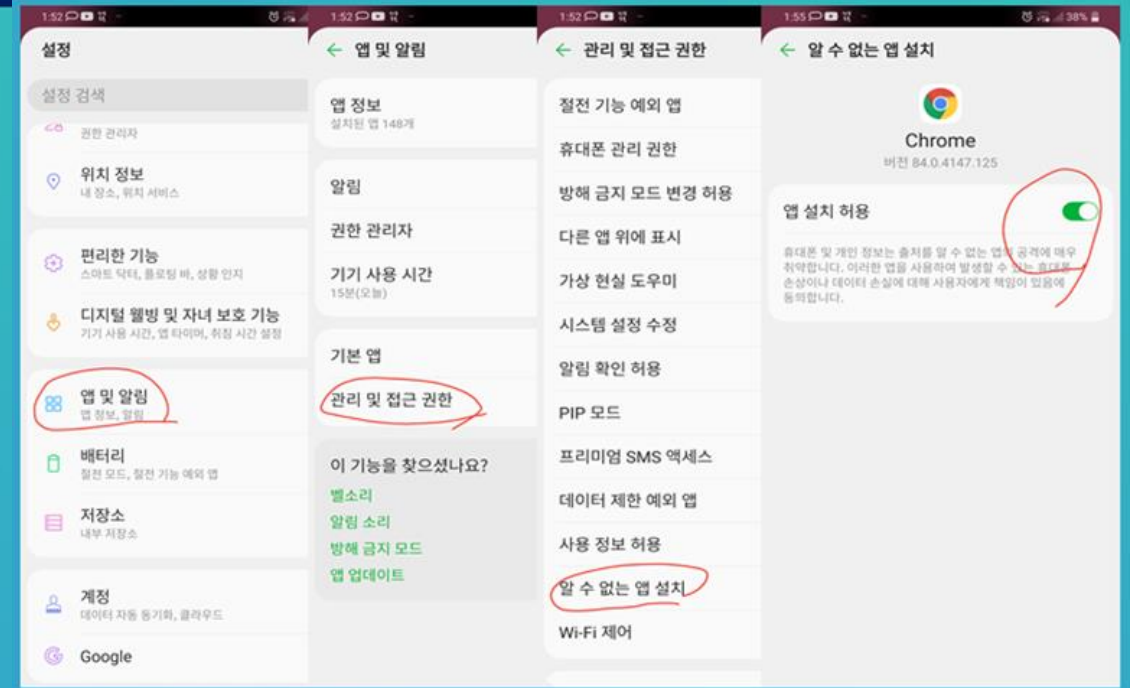
## 01 어플리케이션 설치

- 1) apk 파일을 다운로드 후 실행합니다
- 2) 차단되었다는 메시지가 뜨면 무시하고 설치합니다



## 01 어플리케이션 설치

- 1) **설치가 안 될 경우** 알수없는 앱 설치권한을 허용합니다 (Chrome에서 받아서 바로 설치할 때 Chrome을, 컴퓨터에 옮겨서 설치할 때는 파일관리앱 권한 설정)

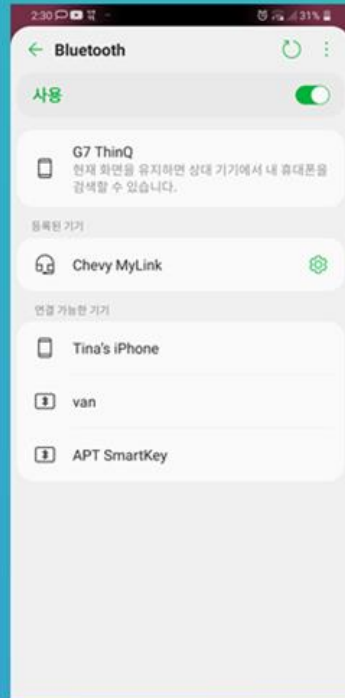




# 식물재배기 앱

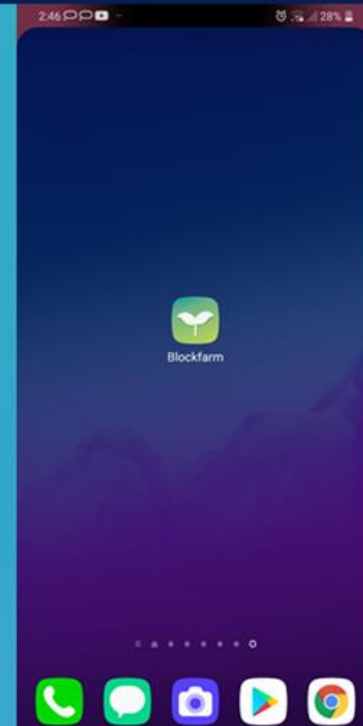
## 02 블루투스 연결

- 1) 처음 설치했을 때 기기 등록을 위해 블루투스 설정 메뉴에 들어갑니다(한 번만 필요)
- 2) '연결 가능한 기기'에서 hc-06 혹은 blockfarm으로 뜨는 기기를 눌러 등록합니다



## 02 블루투스 연결

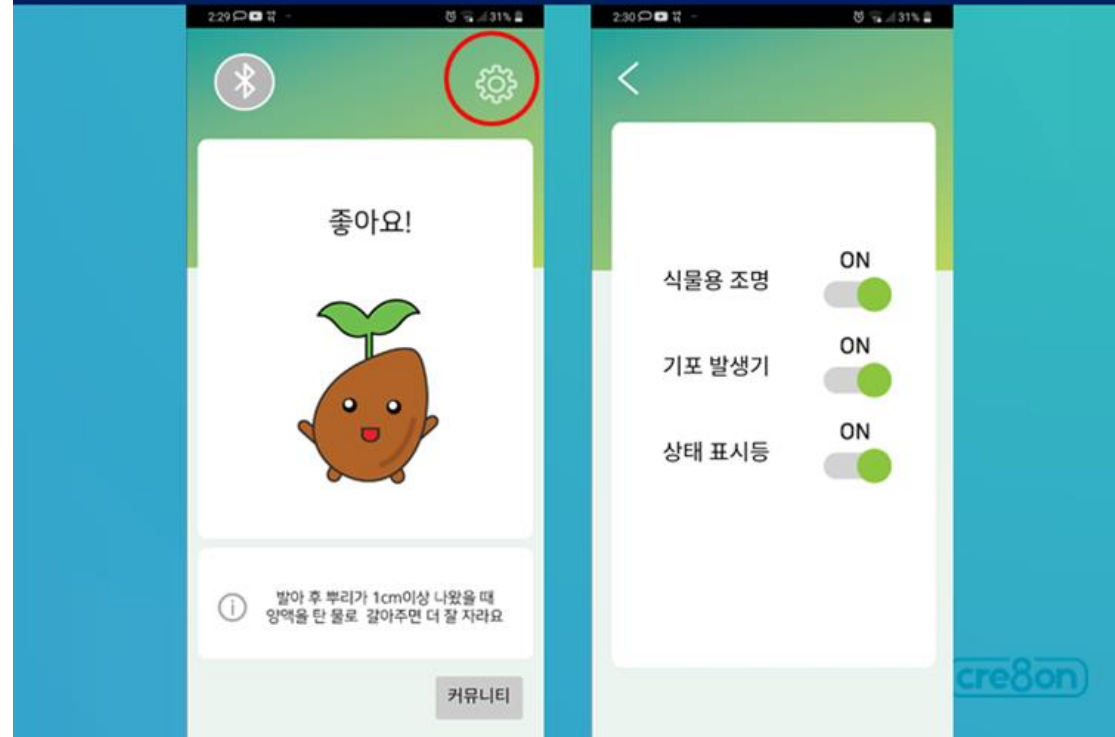
- 1) 설치된 앱을 실행합니다
- 2) 블루투스 버튼을 누른 후, 기기를 선택해 연결합니다



# 식물재배기 앱

## 03 기기 제어하기

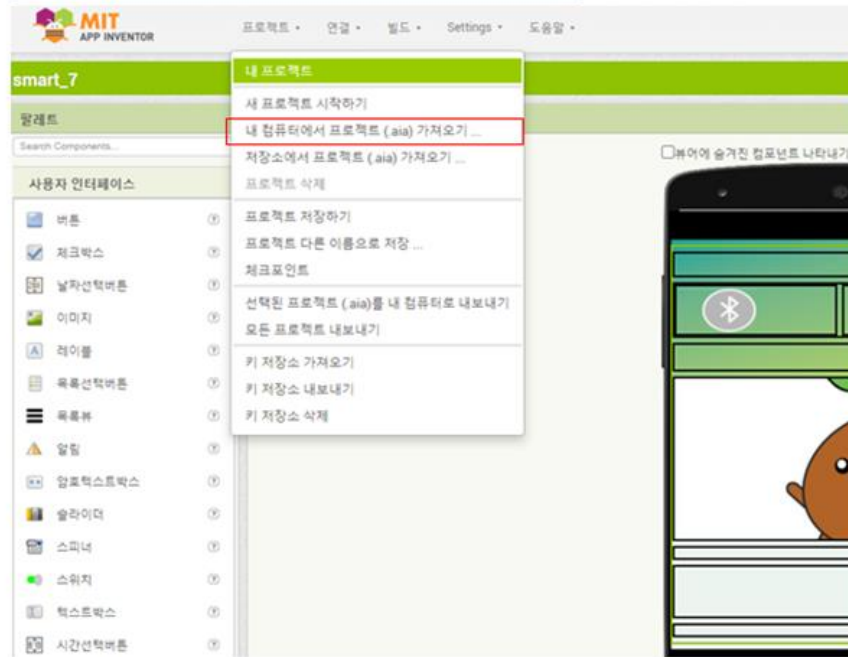
- 1) 환경설정 아이콘을 터치합니다
- 2) 식물용 조명, 기포 발생기, 상태 표시등을 ON/OFF 할 수 있습니다



# 식물재배기 앱

cre8on

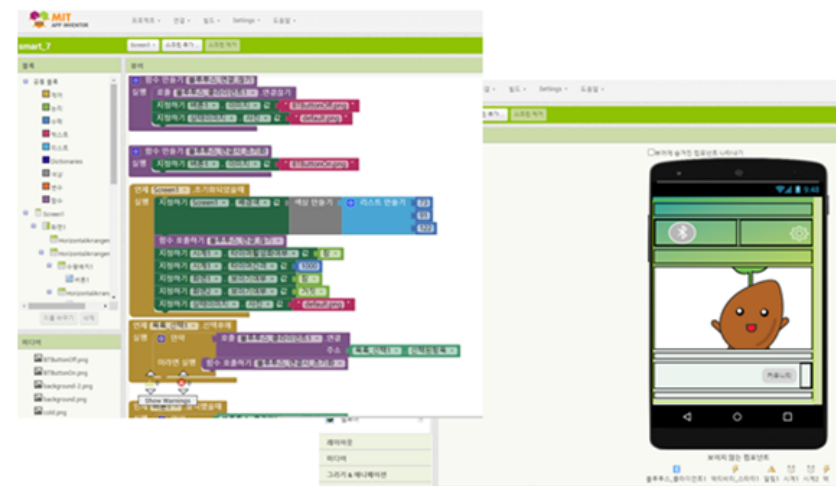
## 오픈소스 소프트웨어 코딩(1)



aia파일을 다운 받은 후 프로젝트-가져오기 합니다

cre8on

## 오픈소스 소프트웨어 코딩(2)



저장된 코딩블록과 이미지를 사용하여 개선버전을 만들 수 있습니다

**주의** 본 캐릭터의 저작권은 CreativeThon에 있으므로, 캐릭터의 상업적 이용을 금지합니다.

스마트화분:Ropot

# 스마트화분의 특징

- 1층에 위치한 카페에서 식물을 키우면서, 학생들의 관찰용 교보재로 활용할 예정임







# 스마트화분:Bloomengine

블루투스 원격제어, 조명등 기능

# 스마트 화분의 특징

- 1층에 위치한 카페에서 식물을 키우면서, 학생들의 관찰용 교보재로 활용할 예정임

# 각 부분 살펴보기







