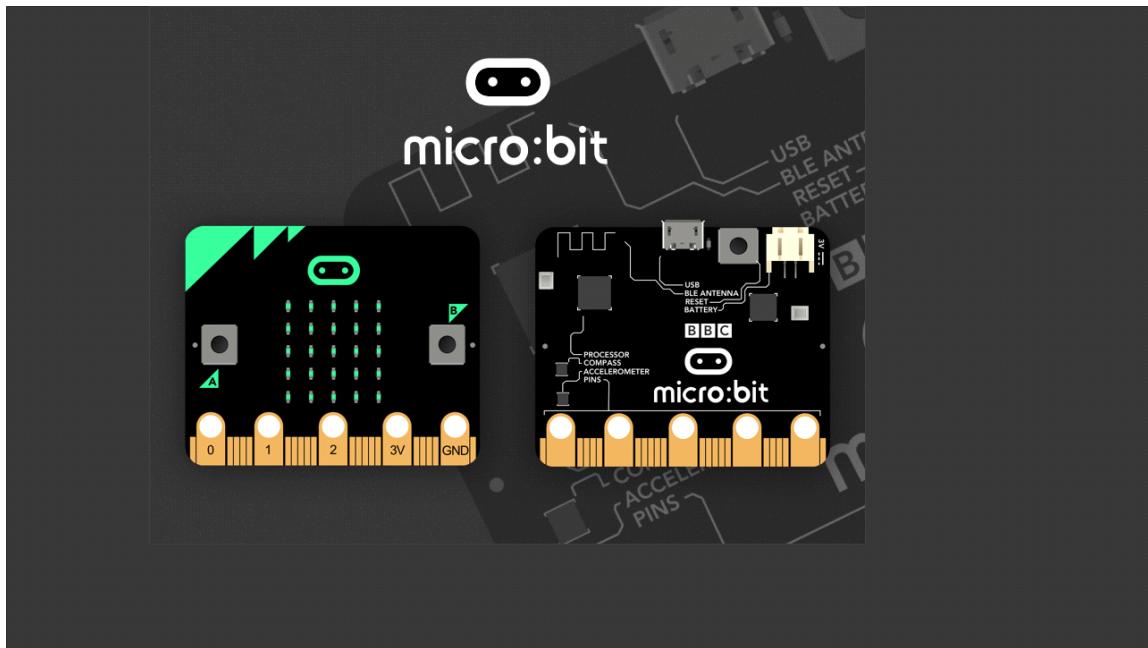


충북교육연구정보원 SW · AI 체험교실

마이크로비트 (초등)

홍덕고등학교 박정진

| | |
|--------------------|----|
| I. 마이크로비트 기초 | 2 |
| 1. 마이크로비트란? | 3 |
| 2. 컴퓨터에 연결하기 | 6 |
| 3. 문제해결 | 11 |
| II. LED 출력 | 13 |
| 1. 문자열 출력 | 14 |
| 2. 아이콘 출력 | 15 |
| 3. LED 출력 | 16 |
| 4. 변수의 활용 | 17 |
| 5. LED 개별제어 | 19 |
| III. 버튼 입력 | 20 |
| 1. 버튼에 반응하기 | 21 |
| 2. 버튼 구분하기 | 23 |
| 3. 게임 블록 | 24 |
| 4. 과녁 맞추기 게임 | 25 |
| IV. 내장 모듈 응용 | 28 |
| 1. 마이크로 소리에 반응하기 | 29 |
| 2. 스피커로 소리 출력하기 | 30 |
| 3. 가속도 센서로 수평계 만들기 | 31 |



피지컬 컴퓨팅은 댄 오설리번(Dan O'Sullivan)과 톰 아이고(Tom Igoe)가 처음 사용한 용어로 ‘물체의 움직임을 감지하는 센서와 이를 제어하는 소프트웨어를 이용하여 현실 세계와 상호 작용이 가능한 물리적 시스템을 만드는 것’을 의미합니다. 예를 들어 어두워지면 켜지는 가로등, 사람이 감지되면 열리는 자동문, 문이 열리면 울리는 사이렌 등이 이러한 피지컬컴퓨팅 사례에 속한다고 볼 수 있습니다.

마이크로비트는 기존에 피지컬컴퓨팅 교육용 교구로 많이 사용되고 있는 아두이노, 라즈베리 파이와 같은 SW 교육용 보드입니다. 외부 입출력을 할 수 있는 많은 단자들이 있으며, 자체 나침반 센서, 온도 센서, 가속도 센서 등을 내장하고 있어 추가적인 부품을 사용하지 않고 마이크로비트만 가지고 있어도 된다는 점이 장점입니다. 또한 통신을 위한 블루투스 4.0을 내장하고 있어 블루투스 모듈 없이 스마트폰 및 스마트 기기와 연동할 수도 있습니다.

1. 마이크로비트란?

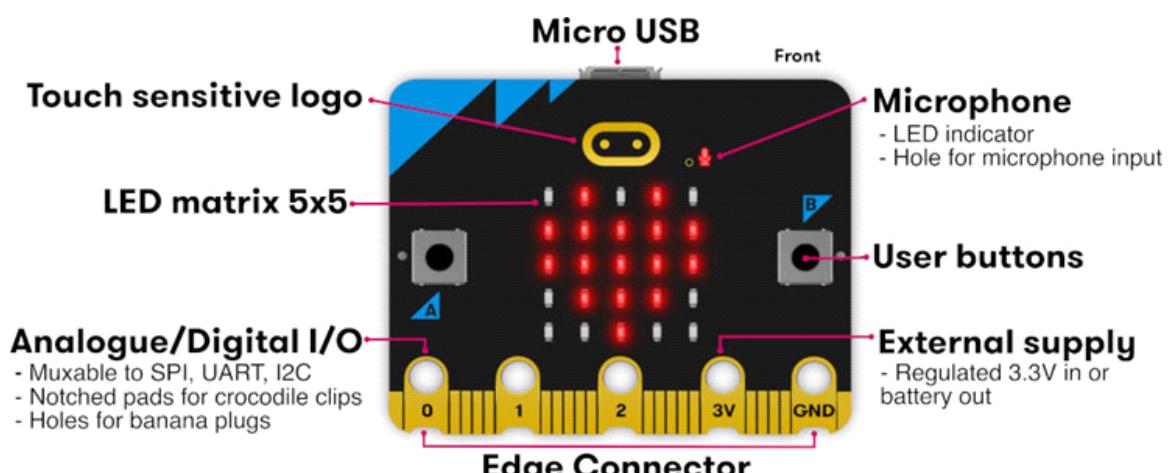
1) 탄생

- 마이크로비트는 영국의 국영방송사인 BBC에서 마이크로소프트, 삼성전자, ARM 등과 같은 회사와 함께 개발한 4 x 5cm 크기의 소형 교육용 보드입니다.
- 외부 입출력을 할 수 있는 많은 단자들이 있으며, 자체 나침반 센서, 온도 센서, 가속도 센서 등을 내장하고 있어 추가적인 부품을 사용하지 않고 마이크로비트만 가지고 있어도 된다는 점이 장점입니다. 또한 통신을 위한 블루투스 4.0을 내장하고 있어 블루투스 모듈 없이 스마트폰 및 스마트 기기와 연동할 수도 있습니다.

2) 재원

- Arm Cortex-M4 32bit processor with FPU @ 64Mhz 마이크로컨트롤러
 - 프로그래밍이 가능한 입/출력 모듈을 가지고 있는 작은 컴퓨터 칩
 - 연산장치, 기억장치, 입출력 장치가 모두 합쳐진 기능 수행
- 512KB ROM, 128KB RAM
- 3.3V에서 작동
- 자체 나침반 센서, 온도 센서, 가속도 센서 등을 내장

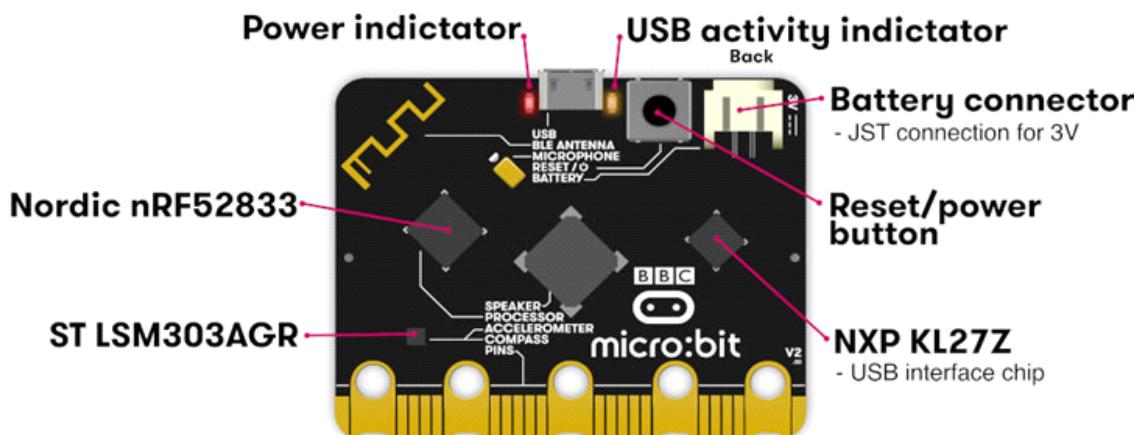
3) 전면부의 구성



[마이크로비트 전면부의 모습]

- 전면부는 다음과 같은 구성을 가지고 있습니다. LED 25개(가로, 세로 5개씩), A 버튼, B 버튼, I/O 핀, 3V, GND.
- LED는 전면부의 가운데 가로와 세로 5개씩 총 25개가 있으며 밝기 조절이 가능하고 디스플레이 용도뿐만 아니라 광센서의 역할도 합니다. A 버튼은 전면부의 가운데 왼쪽에 있는 버튼이며, B 버튼은 전면부의 가운데 오른쪽에 있는 버튼으로 프로그래밍을 통해 사용할 수 있는 푸시버튼으로 게임 및 음악 컨트롤을 위해 사용할 수 있습니다. I/O 핀은 전면부 하단에 있는 5개의 구멍 중 왼쪽부터 3개의 구멍을 말하며 이는 0, 1, 2로 표시되어 있으며 악어 클립 또는 케이블을 사용해서 외부 하드웨어와 연결하여 전력을 공급하거나 디지털 혹은 아날로그 데이터의 입, 출력이 가능한 핀입니다. 3V와 GND는 전원 공급을 필요로 하는 전동 모터 등과 같은 외부 기기에 전원을 공급하는데 사용합니다.

4) 후면부의 구성



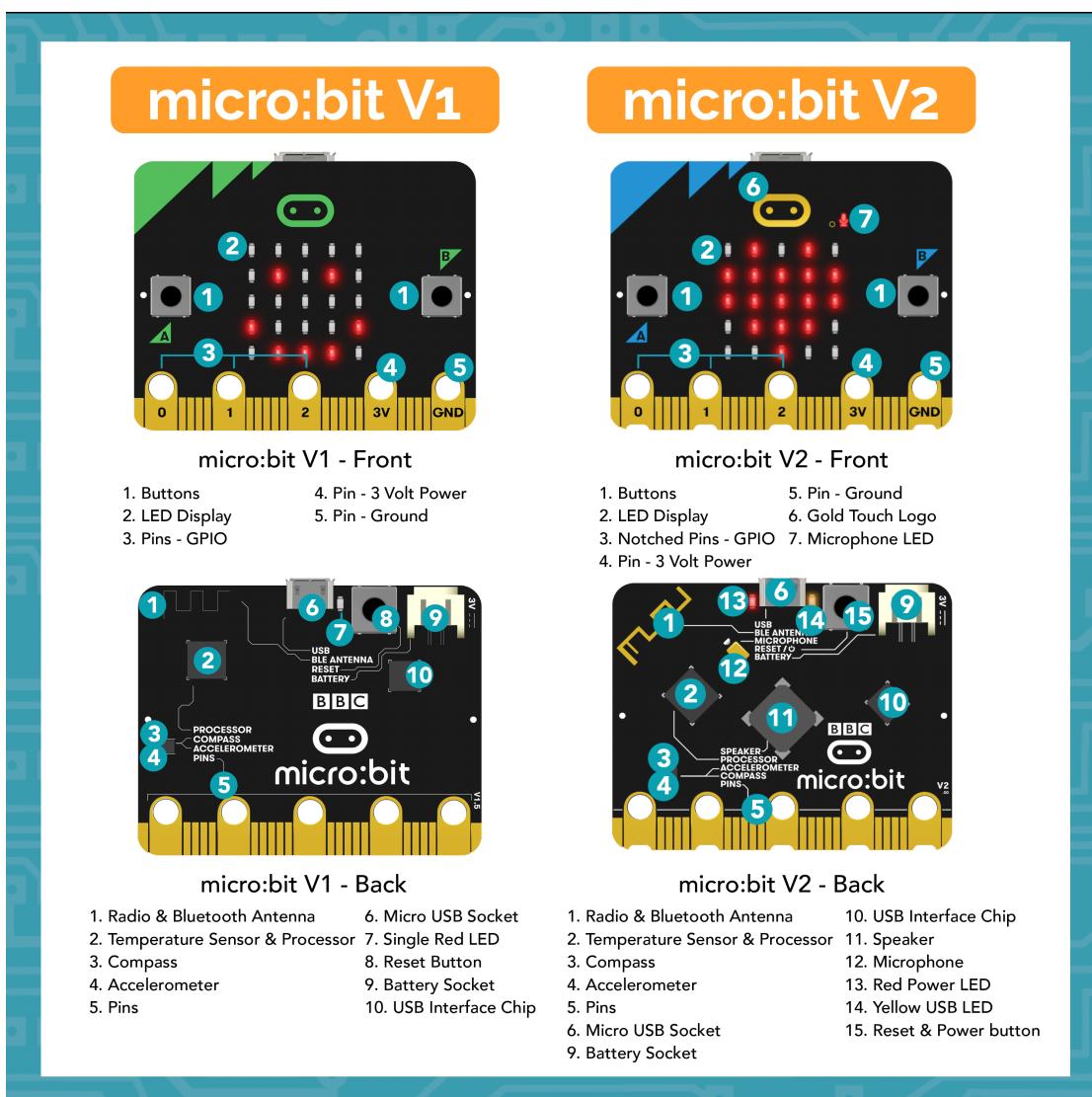
[마이크로비트 후면부의 모습]

- 후면부는 다음과 같은 구성을 가지고 있습니다. 저전력 블루투스 안테나, USB 소켓, 재시작 버튼, 배터리 소켓, 상태 표시 램프, 마이크로컨트롤러, 가속도 및 지자기 센서입니다. 저전력 블루투스 안테나는 후면부 상단 왼쪽에 있는 직선 모양 음파로 되어 나온 부분이며 스마트폰, 태블릿PC 혹은 PC 등 BLE 안테나를 가진 다른 기기들과 정보를 주고받을 수 있어 사물인터넷(IoT)을 위한 프로그램에도 활용할 수 있습니다.
- USB 소켓은 후면부 상단 가운데 있는 처음으로 돌출된 곳이며 제작한 프로그램은 USB 케이블을 통해 마이크로비트로 다운로드할 수 있습니다. 재시작 버튼은 USB 소켓 오른쪽에 있는 돌출된 곳으로 후면부 상단 돌출된 3곳 중 가운데이며 마이크로비트를 재시작할 때 사용합니다. 배터리 소켓은 후면부 상단 돌

출된 3곳 중 가장 오른쪽에 있으며 외장 배터리 팩을 연결하는데 사용됩니다. USB 소켓과 재시작 버튼 사이에는 상태 표시 램프가 있으며 이 램프의 역할은 시스템에서 사용자에게 무언가 알릴 상황이 있으면 노란색 LED 램프가 깜빡이게 됩니다. 후면부 상단 왼쪽에 있는 저전력 블루투스 안테나의 바로 밑에 조금 돌출된 곳이 있는데 이곳은 마이크로컨트롤러로 256KB의 플래시 메모리와 16KB의 RAM을 내장하고 있고, 온도 센싱이 가능합니다.

- 마지막으로 마이크로컨트롤러의 아래쪽에 위치하고 있으며 하단의 왼쪽 첫 번째 홀(GND) 위에 있는 작은 칩은 가속도 및 지자기 센서입니다. 이를 통해 속도 변화, 자기장 감지, 방향 검출 등이 가능합니다.

5) 마이크로비트 버전 비교



2. 컴퓨터와 연결하기

1) USB 케이블로 연결



위와 같이 연결하고 잠시 기다리면 운영체제가 마이크로비트 사용에 필요한 드라이버를 자동으로 설치합니다. 기다리면서 여러분의 마이크로비트가 버전 V1.x 인지 V2.x 인지 확인해 보세요. 다음 과정에서 버전 정보가 필요할 수 있습니다.

2) 사이트 접속

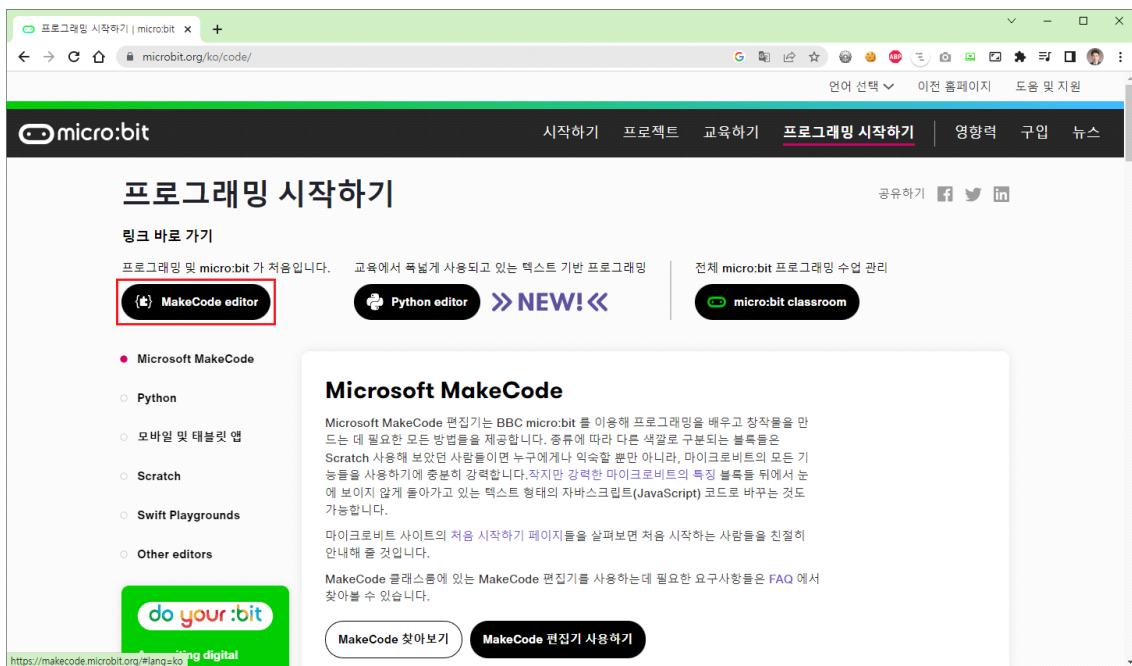
접속 URL: <https://microbit.org/ko/>

Language ^ Archived website Help & support

العربية
Català
Cymraeg
Español
Français
Hrvatski
日本語
한국어
Nederlands
Português (Brazil)
Português (Portugal)
српски
简体中文
繁體中文 (台灣)
Help translate

View archive in your language:
Deutsch
Suomi
עברית
Magyar
Lietuvių
Norwegian
Bokmal
si-LK
தமிழ்
繁體中文 (香港)

Buy News



3) 프로그램 업로드

이제 여러분이 작성한 프로그램을 마이크로비트에 업로드하여 작동시켜보는 테스트를 진행해 보겠습니다. 프로그램을 작동시키려면 마이크로비트에 프로그램을 반드시 업로드해야 하는데 여기에는 두 가지 방법이 있습니다.

첫째, 여러분이 만든 프로그램을 .hex 파일로 다운로드 받고 이 파일을 윈도우 탐색기를 열어서 마이크로비트로 전송

둘째, 여러분이 만든 프로그램을 중간 과정 없이 마이크로비트로 직접전송 (direct flashing)

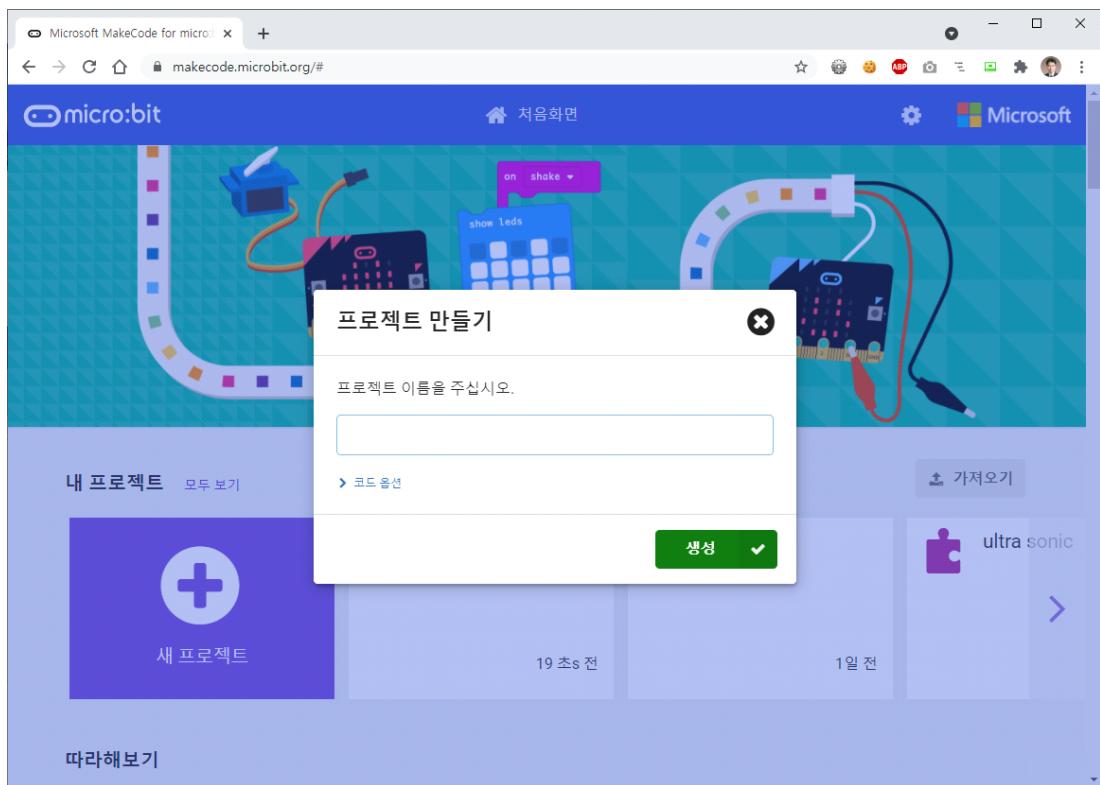
우리는 두 번째 방식을 통해 더 쉽고 편리하게 코딩을 해 볼 예정입니다. 하지만 경우에 따라서는 펌웨어 설치가 필요하거나 윈도우 드라이버를 재설치 해야 할 수도 있습니다. 일단 다음 과정을 따라해보고 문제가 발생한다면 다음 장의 문제해결 과정을 통해 해결해 보도록 하겠습니다.

4) 메이크코드(MakeCode) 열기

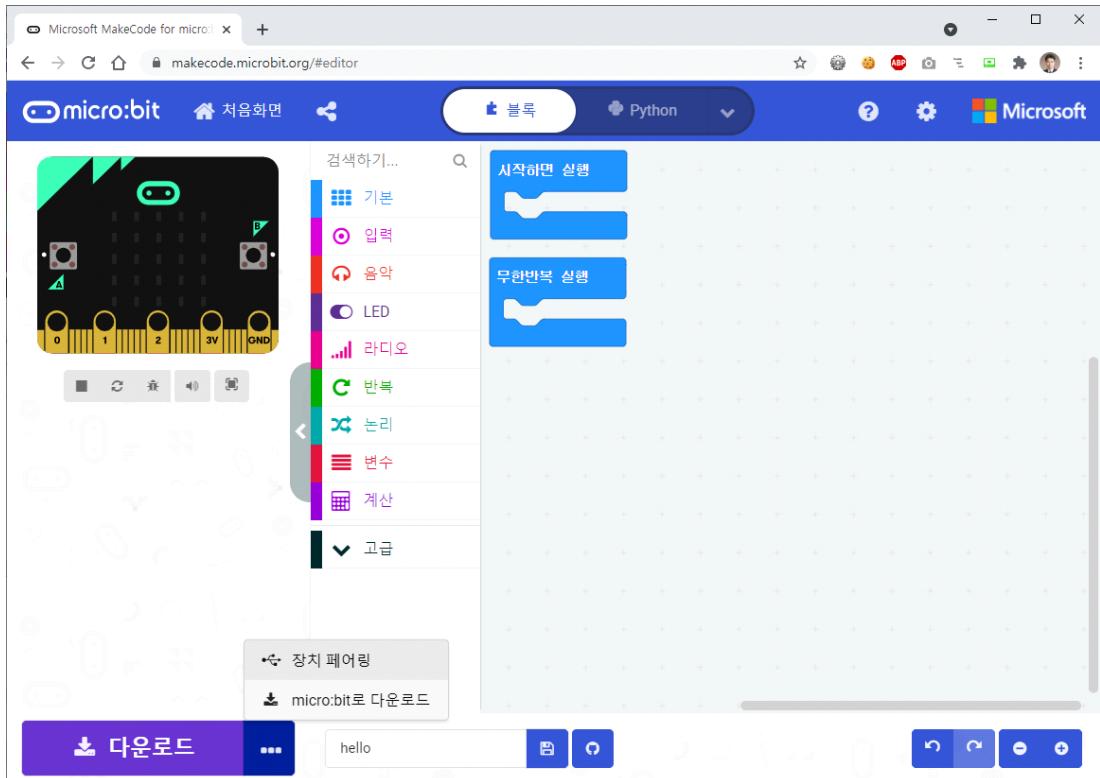
가. 바로 업로드 기능을 사용하기 위해 webUSB를 지원하는 크롬 또는 에지 브라우저를 사용하여 다음 사이트에 접속합니다.

<https://makecode.microbit.org/>

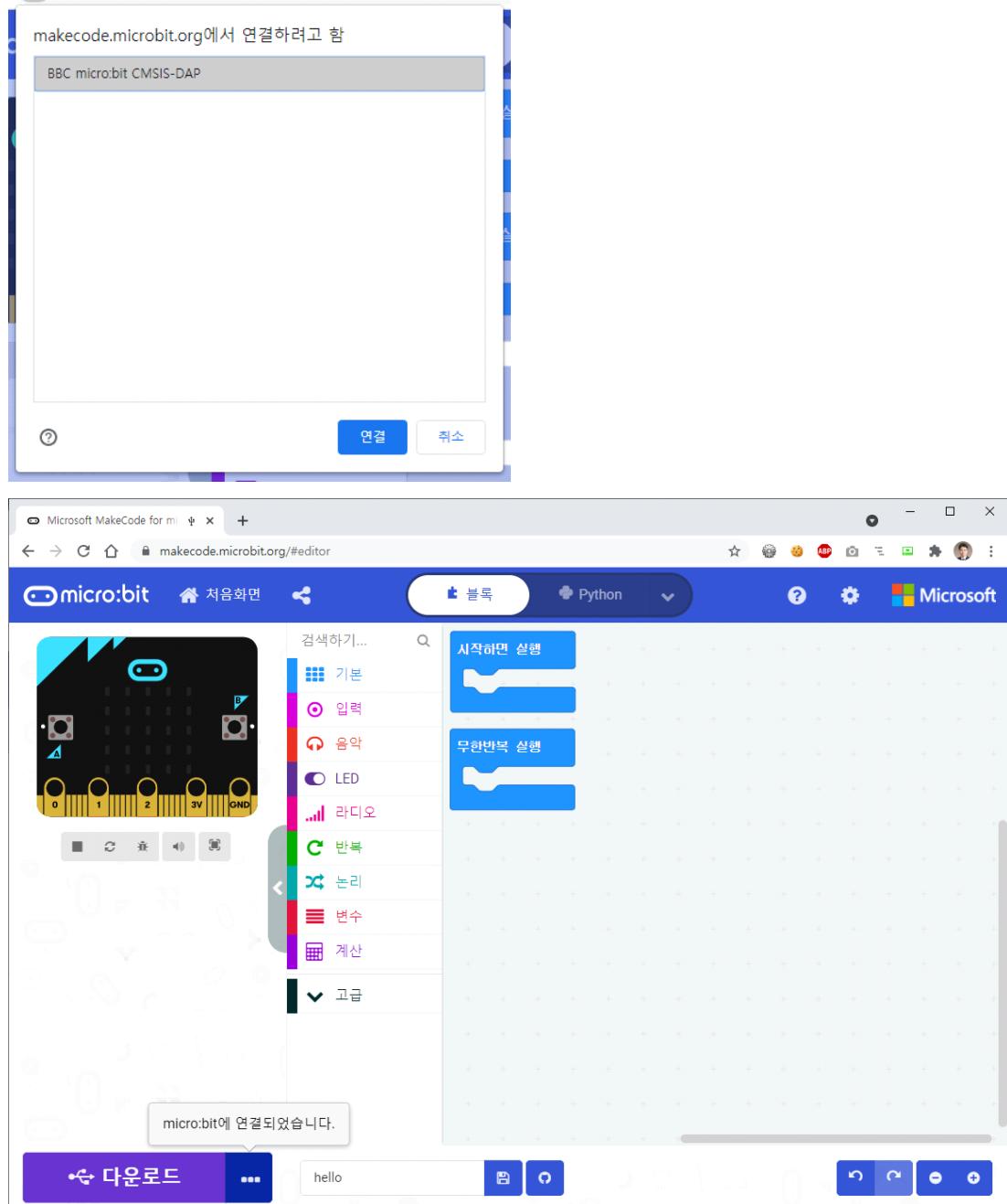
나. 다음 화면에서 ‘새 프로젝트’를 클릭하고 프로젝트 이름을 hello 라고 정합니다.



5) 장치 페어링



왼쪽 하단 다운로드 버튼 오른쪽 …에 마우스 포인터를 올려놓고 ‘장치 페어링’을 선택한 뒤 아래 그림과 같이 장치를 선택하고 ‘연결’ 버튼을 누릅니다.

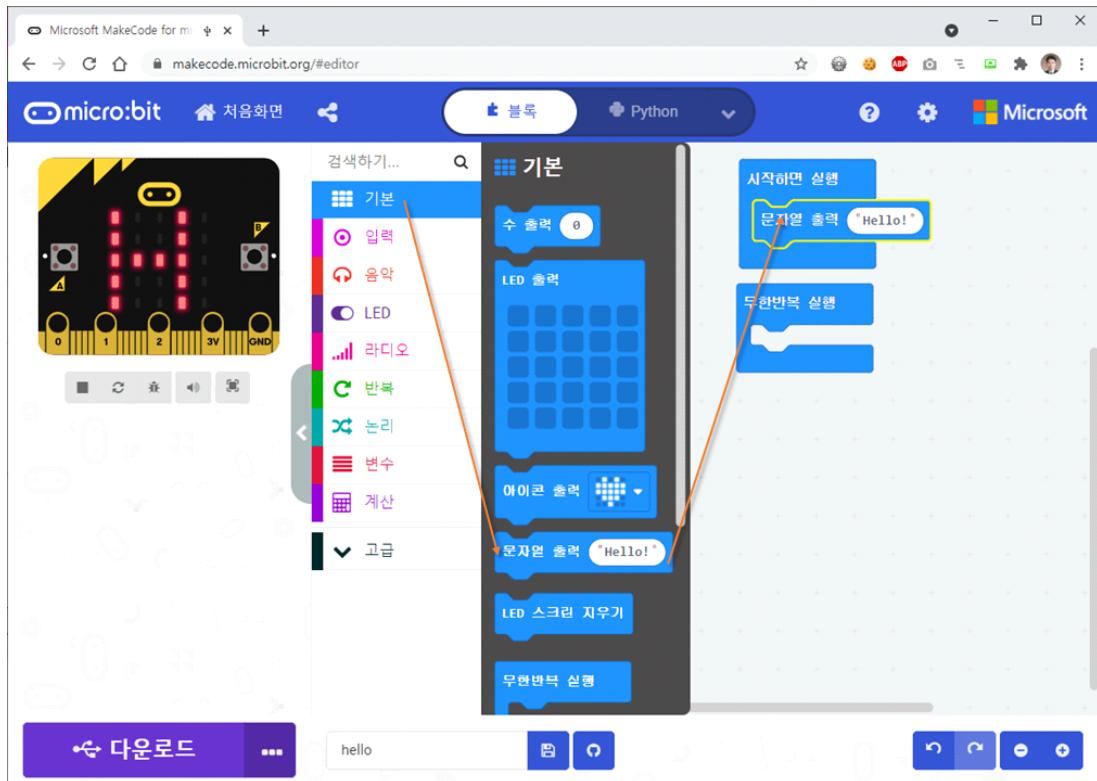


위 그림에서와 같이 ‘micro:bit에 연결되었습니다.’라는 메시지가 보이면 성공입니다.

이제 바로 업로드 기능이 정상작동하는지 테스트 프로그램을 만들어 보겠습니다.

6) 테스트 프로그램 작성

테스트 프로그램은 아래와 같이 만듭니다.



↳ 기본 블록 그룹에서 **문자열 출력 "Hello!"** 블록을 마우스 드래그 앤 드롭으로 끌어와 위 그림과 같이 연결합니다.

マイクロビット가 시작되면 LED에 'Hello' 라는 문자열을 출력하라는 프로그램이지요.

↳ 다운로드 버튼을 누르면 방금 작성한 프로그램이 마이크로비트로 업로드가 개시되고 잠시 후 마이크로비트가 리셋되면서 프로그램이 시작됩니다. 바로 여러분이 방금 만든 프로그램이 실행되는 것입니다.

마이크로비트의 LED 매트릭스에 Hello 글자가 스크롤 되면 성공입니다.

3. 문제해결

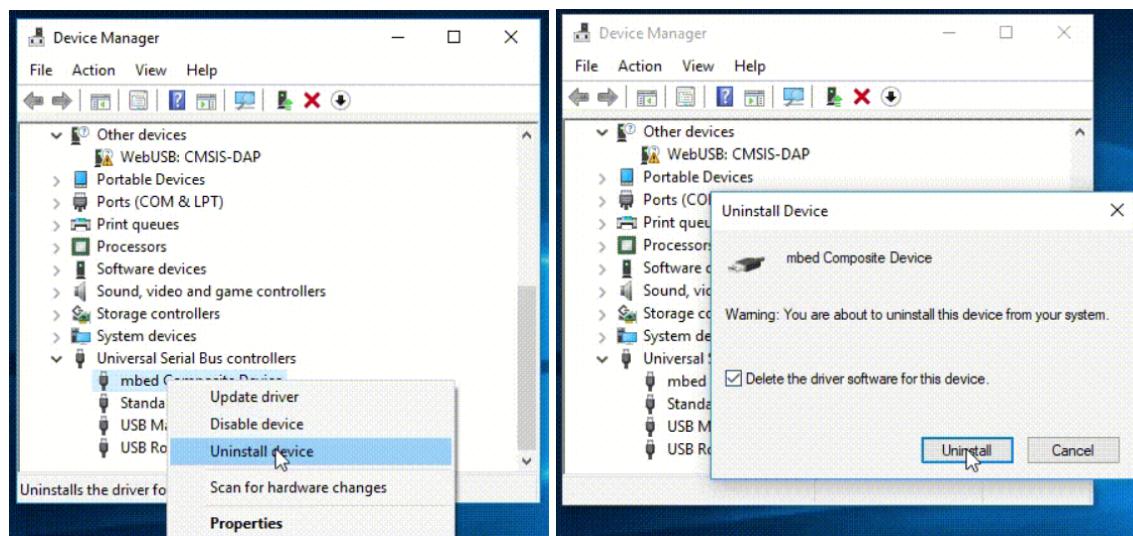
1) WebUSB 요구사항 확인

WebUSB를 사용하려면 운영체제로 Windows 8+, Mac 또는 Linux, 버전 65 이상의 Google Chrome과 같은 크롬 기반 브라우저, 버전 0249 이상의 micro:bit 펌웨어가 필요합니다.

편집기 사이를 이동하는 경우(예: Python > MakeCode) micro:bit에 이전 편집기에서 생성된 프로그램이 이미 포함되어 있으면 webUSB를 통한 플래싱 시간이 더 오래 걸릴 수 있습니다.

2) 윈도우 WebUSB 드라이버 재설치

- 가. Windows 장치 관리자를 열고 micro:bit USB 장치가 표시되는지 확인합니다.
- 나. mbed Composite Device 항목을 찾아 마우스 오른쪽 버튼으로 클릭하고 Uninstall device 를 선택합니다 .
- 다. 새 창이 열리면 이 장치의 드라이버 소프트웨어를 삭제합니다 상자를 선택하고 제거를 클릭합니다 .
- 라. 마이크로비트를 분리했다가 다시 연결하면 USB 장치 아래에 나타납니다.



3) 펌웨어 재설치

가. 펌웨어 다운로드

<https://microbit.org/get-started/user-guide/firmware/>

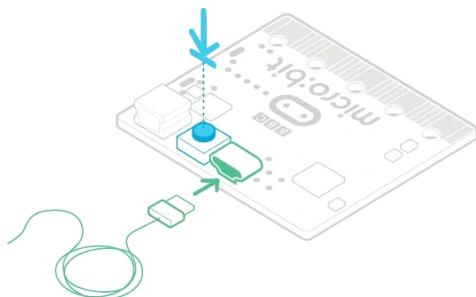
위 사이트에 접속하여 나의 마이크로비트에 해당하는 펌웨어를 다운로드 받습니다. 버전 V1.3, V1.5 등 1.x 이라면 왼쪽 버튼을 누르면 됩니다.

- Download the hex file appropriate for your version of the micro:bit from this page to your computer. Read our article on [identifying the micro:bit version](#), if you are unsure about which version you have.

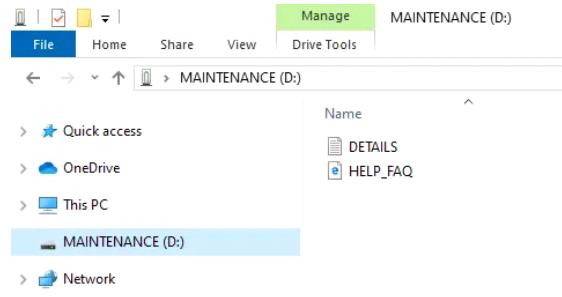
Firmware for micro:bit V1

Firmware for micro:bit V2

나. 마이크로비트 MAINTENANCE 모드 진입



MAINTENANCE 버튼



MAINTENANCE 모드

마이크로비트에서 USB 케이블을 제거하고 왼쪽 그림에서와 같이 MAINTENANCE 버튼을 누른 상태에서 다시 USB 케이블을 연결합니다. 그러면 오른쪽 그림과 같이 유지보수 모드로 진입하게 됩니다. 이제 MAINTENANCE 버튼에서 손을 떼어도 됩니다.

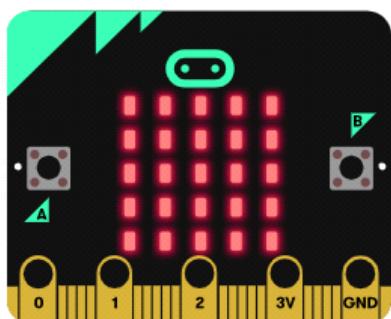
다. 펌웨어 업로드

1번 과정에서 다운로드 받은 펌웨어(.hex) 파일을 마이크로비트 (MAINTENANCE 드라이브)에 드래그 앤 드롭 합니다. 잠시 기다리면 업그레이드가 완료되고 마이크로비트가 자동으로 리셋 됩니다.

이제 모든 과정이 완료되었습니다.

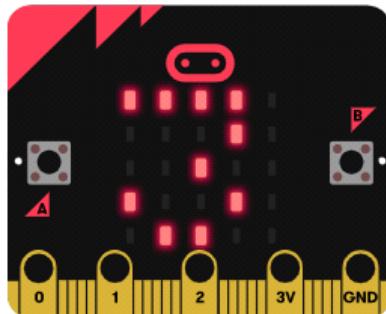
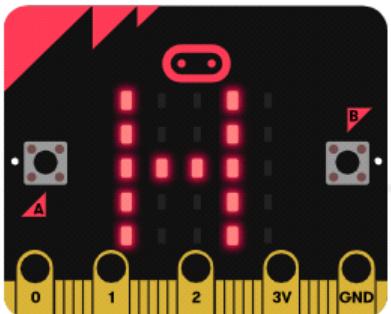
LED 매트릭스

발광 다이오드를 의미하는 LED는 빛을 출력하는 장치입니다. 마이크로비트에는 프로그램 가능한 25개의 LED가 내장되어 있습니다. 프로그램 가능하다는 뜻은 프로그램을 만드는 사람이 원하는 대로 켜거나 끌 수 있다는 뜻입니다.



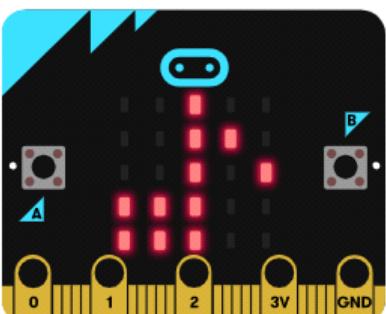
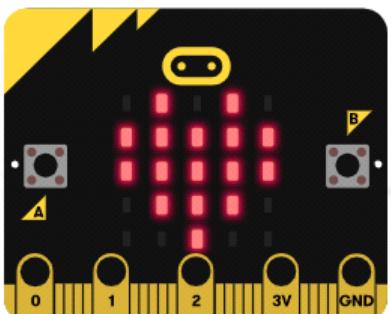
25개의 LED는 왼쪽 그림과 같이 5행 5열로 배치되어 있는데 이러한 형태의 배열을 매트릭스라고 합니다. 여기서는 LED로 매트릭스를 만들었으니 ‘LED 매트릭스’가 되는 것이지요.

LED 매트릭스는 각 점들을 켜거나 끌으로써 여러 가지 의미를 만들어 낼 수 있습니다.



위쪽 그림과 같이 알파벳이나 숫자를 표현할 수도 있고 스크롤 기능을 통해 ‘Hello’와 같은 긴 문자열을 표현할 수 있습니다.

또한 아래 그림과 같이 그림이나 아이콘을 그리는 것도 가능합니다.

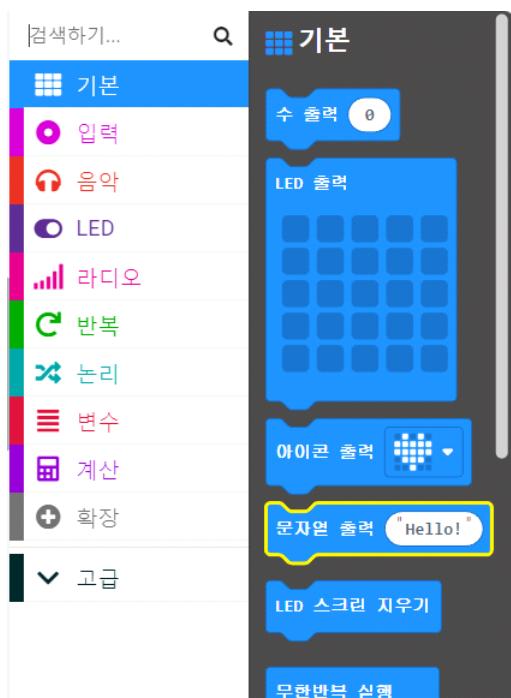


1. 문자열 출력

1) 블록 소개

먼저 코딩의 세계에서 ‘문자열’이 무엇인지 알아봅시다. 여러분이 키보드로 타이핑 할 수 있는 글자 하나하나를 ‘문자’라 부릅니다. 그리고 이러한 문자가 하나이상 모인 집합을 ‘문자열’이라고 합니다. “Hello”, “World”와 같은 문장을 문자열이라 할 수 있지요.

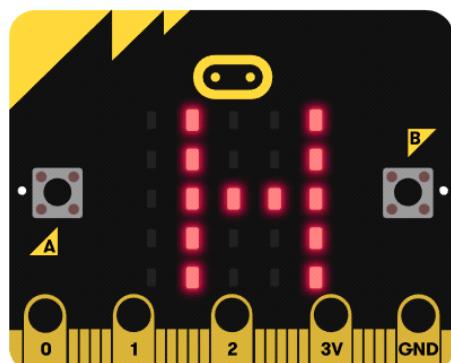
[문자열 출력 블록]



[사용 예시]



2) 실습



Hello라는 문자열이 왼쪽으로 스크롤 되면서 출력됩니다.



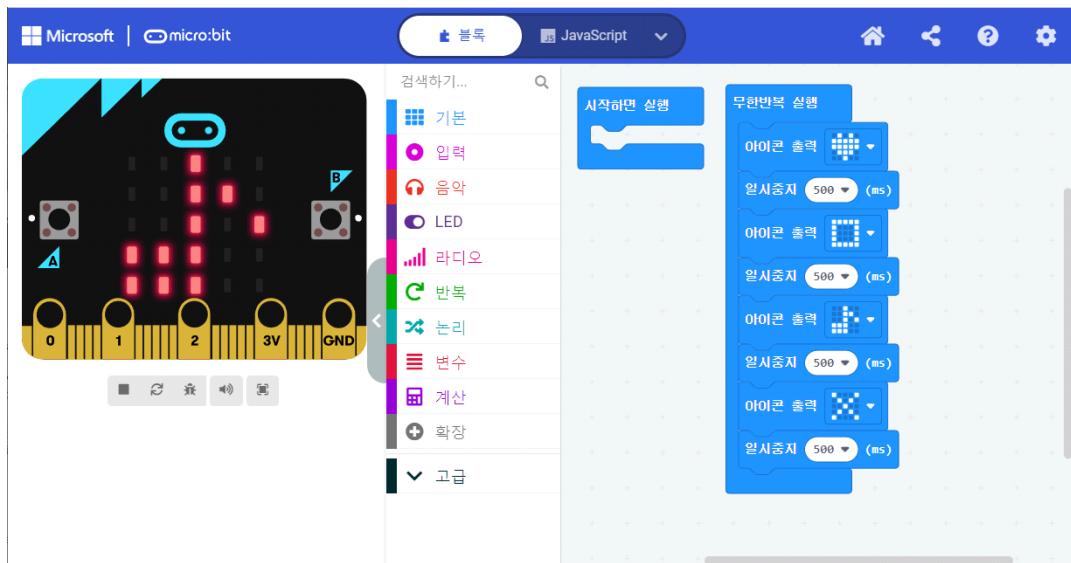
이 블록은 마이크로비트의 동작을 잠시 중지 시킬 때 사용합니다. 단위 (ms)는 밀리초를 의미합니다. 즉 1000분의 1초이기 때문에 1000을 적으면 1초, 500을 적으면 0.5초가 됩니다.

2. 아이콘 출력

1) 블록 소개



2) 실습



♥ □ ♪ ✕ 아이콘이 0.5초 간격으로 순서대로 출력되는 것을 확인할 수 있습니다.

3. LED 출력

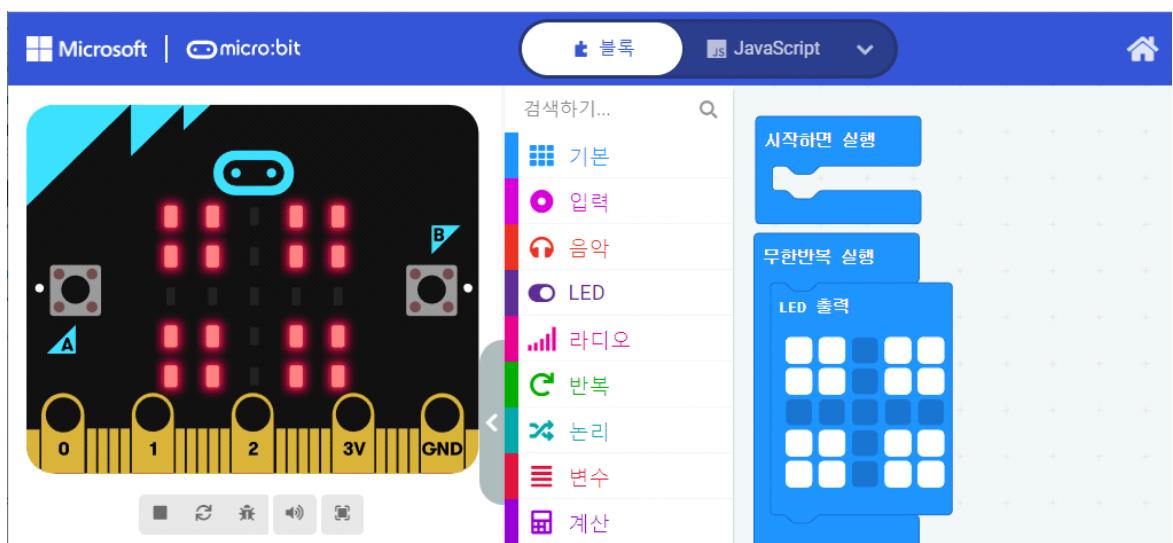
1) 블록 소개



사각형 모양의 네모 칸을 한번 클릭할 때마다 반전됩니다.

이러한 방식으로 원하는 모양을 정의하고 이 모양 그대로 출력을 할 수 있습니다.

2) 실습



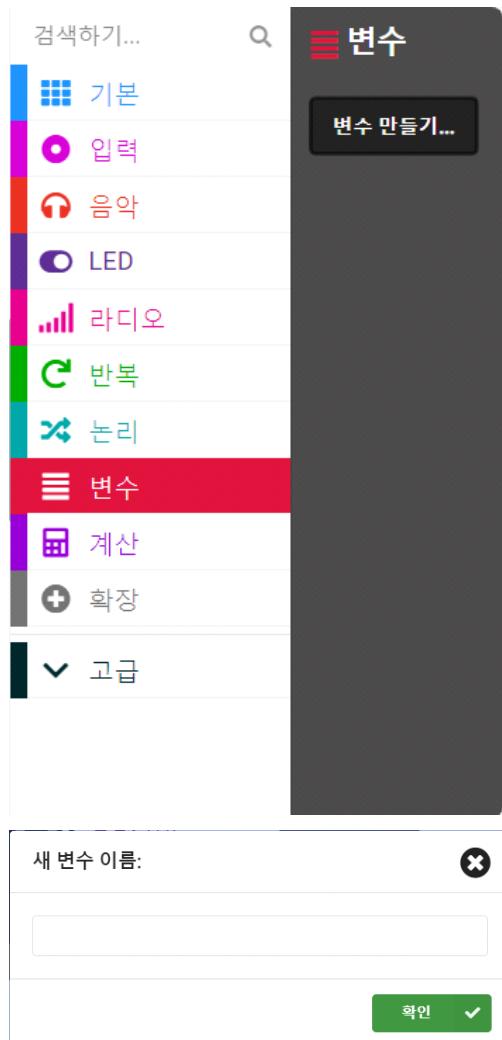
4. 변수의 활용

1) 변수란?

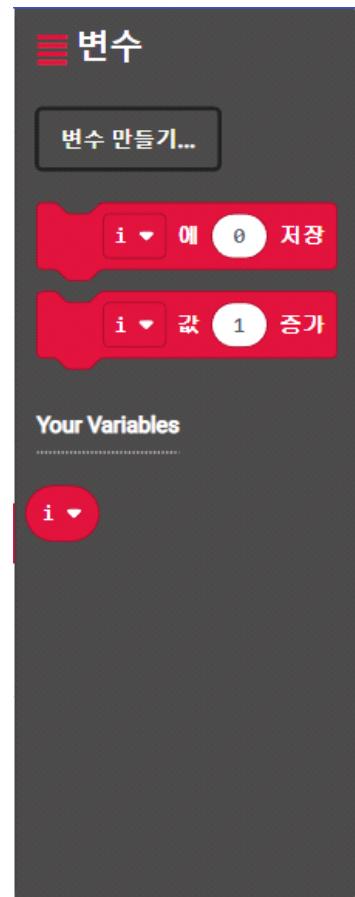
변수는 ‘변하는 수’라는 의미로 코딩의 세계에서 컴퓨터에게 무언가를 기억시켜야 할 때 사용합니다. 예를 들어, 1부터 10까지 차례로 숫자를 출력하여야 한다면 현재 출력 중인 숫자가 무엇인지 기억하고 있어야 다음 숫자로 넘어갈 수 있기 때문에 현재 출력 숫자를 변수에 기억시키는 방식으로 코딩을 합니다. 그리고 변수를 사용하기 위해서는 이름을 붙여줘야 하는데 변수의 이름이 중복되어서는 안됩니다. 이름이 겹치면 서로를 구분할 수 없기 때문이지요.

변수를 만들려면 아래 그림에서와 같이 ‘변수 만들기...’를 클릭하면 나타나는 대화상자에 변수명(변수 이름)을 적어줍니다.

[변수 만들기 블록]



[변수 제어 블록]



저장은 변수에 값을 지정하는 것이고, 증가는 변수의 값을 변화시키는 블록입니다.

2) 실습 예제

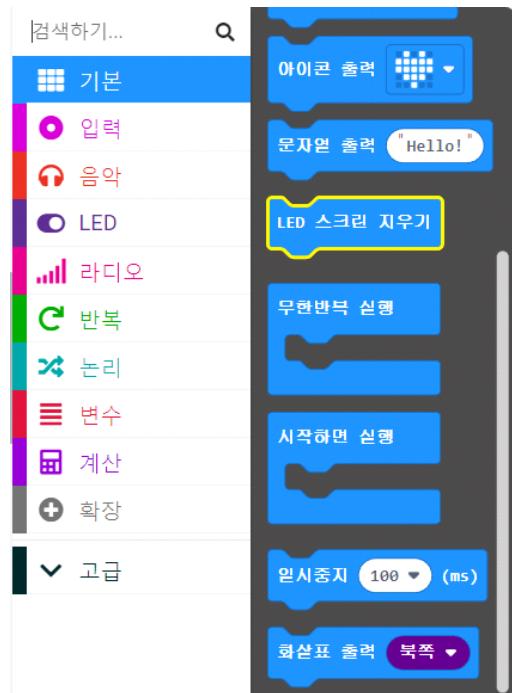
9부터 0까지 카운트 다운을 숫자를 출력한 뒤 폭발하는 별을 출력해 봅시다.

- 카운트 다운 숫자 출력을 위해 변수를 사용합니다.
- 9부터 0까지 숫자를 10번 출력하기 위해 반복 블록을 사용합니다.

[최종 결과]



[LED 스크린 지우기 블록]



9부터 카운트 다운이 시작되므로 변수 *i*에 9를 저장하고 시작합니다.

‘반복(while) 블록’은 조건을 만족하는 동안 내부의 블록들을 반복하여 수행합니다. 여기에서는 숫자 0까지 출력하여야 하므로 조건을 $i \geq 0$ 으로 지정하였습니다. 반복 내용으로는 변수 *i*를 출력하고 다음 숫자로 넘어가기 위해 변수 *i*를 1 감소시켰습니다. 이 반복 블록은 $i \geq 0$ 인 동안 수행되므로 0까지 출력하게 되고 *i*값이 감소하여 -1이 되면, 이 반복 블록을 탈출하게 됩니다.

그리고 폭발하는 별 모양을 만들기 위해 아이콘 3개를 연속하여 출력한 뒤, LED 스크린 지우기 블록을 사용하였습니다. 이 블록은 LED 매트릭스의 모든 LED를 끕니다.

5. LED 개별 제어

1) LED 좌표계

| x y | 0 | 1 | 2 | 3 | 4 |
|--------|---|---|---|---|---|
| 0 | | | | | |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

가로 방향을 x, 세로 방향을 y라하고 좌표의 시작 숫자는 0부터입니다.

따라서 왼쪽 상단의 좌표는 x:0, y:0, 오른쪽 하단의 좌표는 x:4, y:4입니다.

2) 미션

LED 매트릭스에서 오른쪽 아래 대각선 방향으로 LED를 차례로 켜다가 끄는 애니메이션을 만드시오.

[힌트] LED 매트릭스의 (x, y) 위치가, $(0, 0) \rightarrow (1, 1) \rightarrow (2, 2) \rightarrow (3, 3) \rightarrow (4, 4)$ 순서로 반전되면 됩니다.

| x y | 0 | 1 | 2 | 3 | 4 |
|--------|---|---|---|---|---|
| 0 | ● | | | | |
| 1 | | ● | | | |
| 2 | | | ● | | |
| 3 | | | | ● | |
| 4 | | | | | ● |

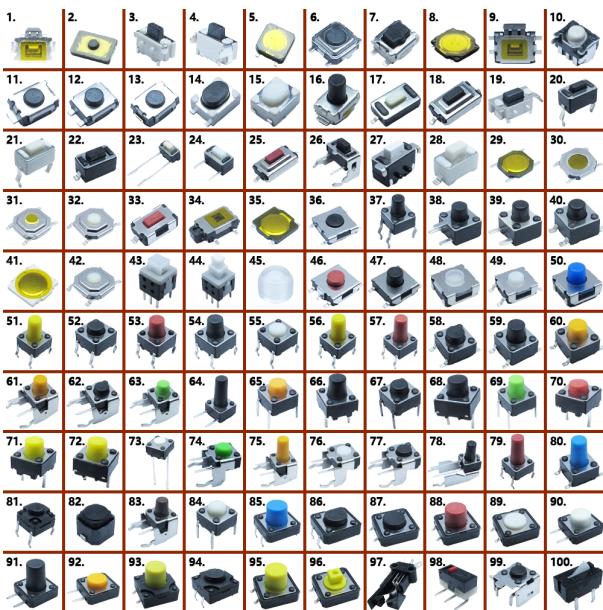
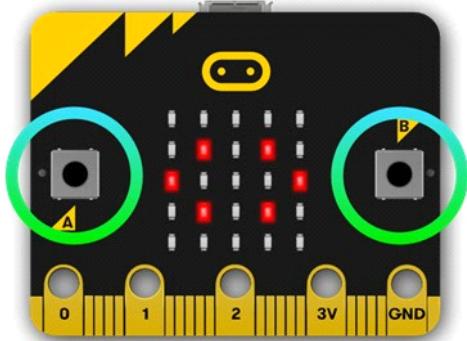


‘반복(for)’ 블록은 특정 변수를 0부터 시작하여 지정한 숫자까지 1씩 증가시켜가며 반복을 수행합니다. 반복 횟수가 고정되어 있는 반복 상황에서 사용하면 유용합니다. 여기에서는 0부터 4까지 변화하여야 하므로 4까지 반복하도록 지정하였습니다.

LED 반전은 on이면 off로 off이면 on으로 바꾸는 것을 의미합니다.

버튼은 기계나 장치를 누르기 위한 스위치를 가리키는 말입니다. 오른쪽 그림과 같이 많은 종류의 버튼이 존재합니다.

マイクロビット에는 총 두 개의 버튼이 내장되어 있습니다.



왼쪽의 버튼을 A 버튼이라 하고 오른쪽의 버튼을 B 버튼이라고 합니다.

버튼의 누름 상태는 아래와 같이 총 세 가지 상태가 존재합니다.

첫째, A 버튼 누름.

둘째, B 버튼 누름.

셋째, A · B버튼 모두 누름.

그리고, 버튼의 이벤트는 아래와 같이 두 가지로 구분할 수 있습니다.

첫째, 버튼이 눌림.

둘째, 눌렸던 버튼이 떨어짐.

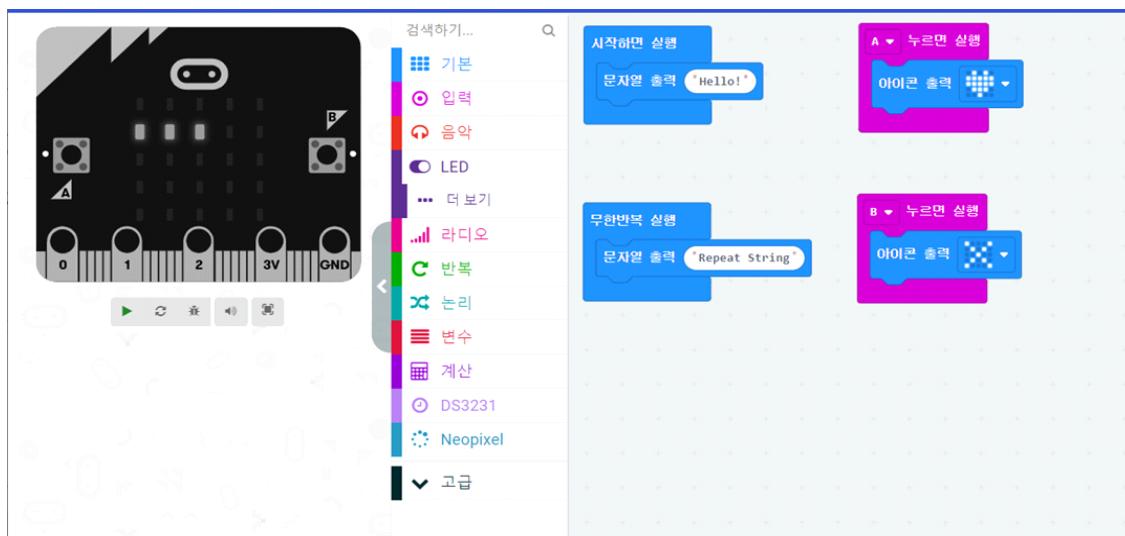
이제부터 이 다양한 버튼의 상태와 이벤트를 프로그램에서 어떻게 다루는지 살펴보겠습니다.

1. 버튼에 반응하기

1) LED 사용중 버튼에 반응하기



아래와 같은 코딩은 문자열이 출력이 모두 완료되어야 버튼이 작동되는 문제 가 있습니다. 이 문제를 어떻게 해결해야 할까요?



아래와 같이 ‘LED 스크린 애니메이션 중단 및 삭제’ 블록을 추가하여야 버튼에 즉시 반응할 수 있습니다.



2) 버튼의 눌림과 떨어짐 감지하기

위에서 사용한 버튼 입력 블록은 버튼을 눌렀다 뗄 때 작동이 일어납니다. 이러한 방식으로는 세밀한 제어가 불가능한 것이지요. 버튼을 누를 때와 뗄 때 실시간으로 감지할 방법은 없을까요? 아래 예제는 버튼이 눌려 있으면, ↓를 출력하고 버튼에서 손을 떼면 ↑를 출력하는 프로그램입니다.



위 그림과 같이 ‘무한반복 실행’ 블록에 ‘만약’ 블록으로 버튼의 눌림 상태를 확인하면 버튼의 눌림과 떨어짐을 실시간으로 감지할 수 있습니다.

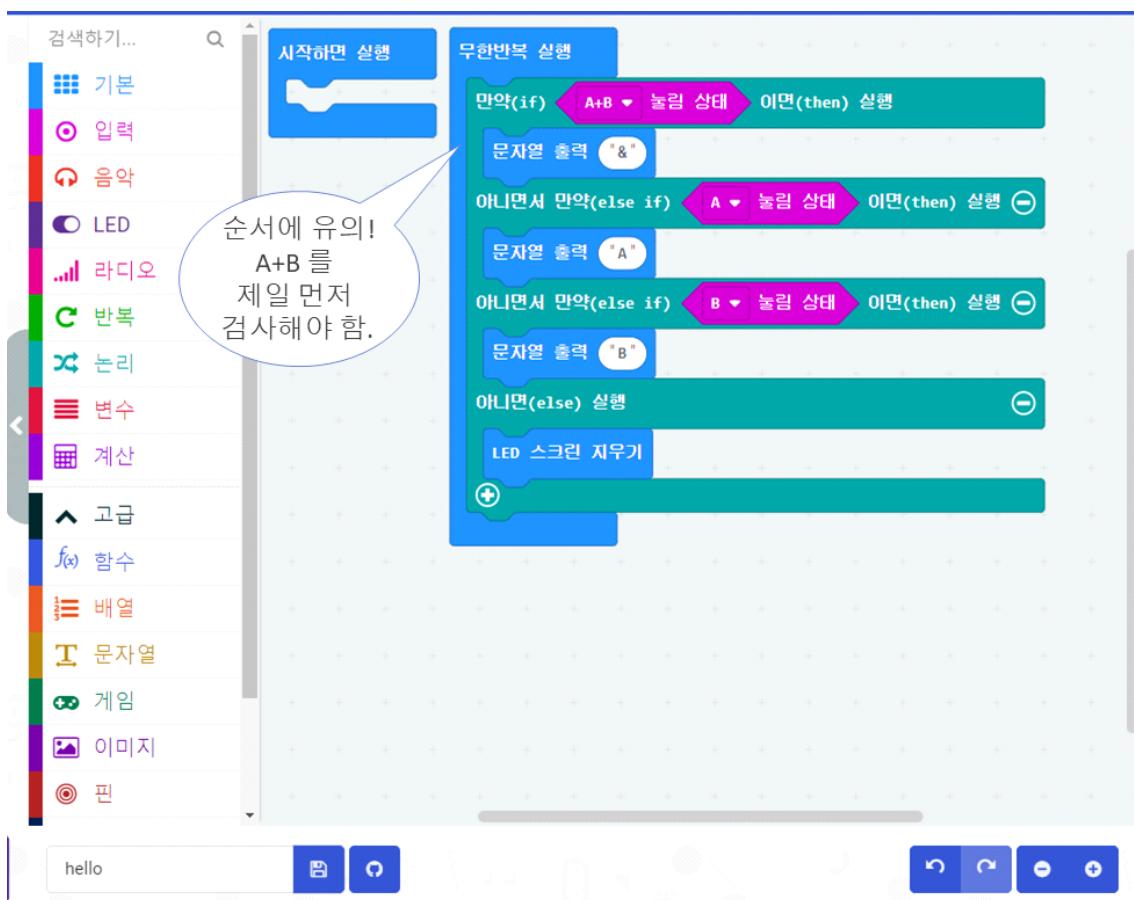
2. 버튼 구분하기

1) 미션

아래와 같은 문제를 해결해 보자.

A버튼이 눌리면 LED 매트릭스에 A를 출력,
B버튼이 눌리면 LED 매트릭스에 B를 출력,
둘 다 눌리면 &를 출력하는 프로그램을 제작하시오.

2) 정답



‘만약’ 블록에서 버튼 눌림 상태를 체크 함에 있어 ‘A+B 눌림 상태’를 가장 먼저 확인해야 합니다. 이렇게 하지 않고 ‘A 눌림 상태’ 또는 ‘B 눌림 상태’를 먼저 확인하는 방식으로 순서를 바꾸게 되면 A와 B버튼을 동시에 누른 상태를 처리할 수 없게 됩니다. 왜냐하면, A와 B버튼을 동시에 누른 상태는 A버튼을 누른 상태에도 속하기 때문에 ‘A’를 출력한 뒤 ‘만약’ 블록을 탈출하기 때문입니다.

3. 게임 블록

1) 게임 블록 소개

메이크코드에서는 여러분이 게임을 쉽게 만들 수 있도록 스프라이트라는 개념을 제공합니다. 스프라이트는 움직일 수 있는 점으로 항상 화면에 표시됩니다.

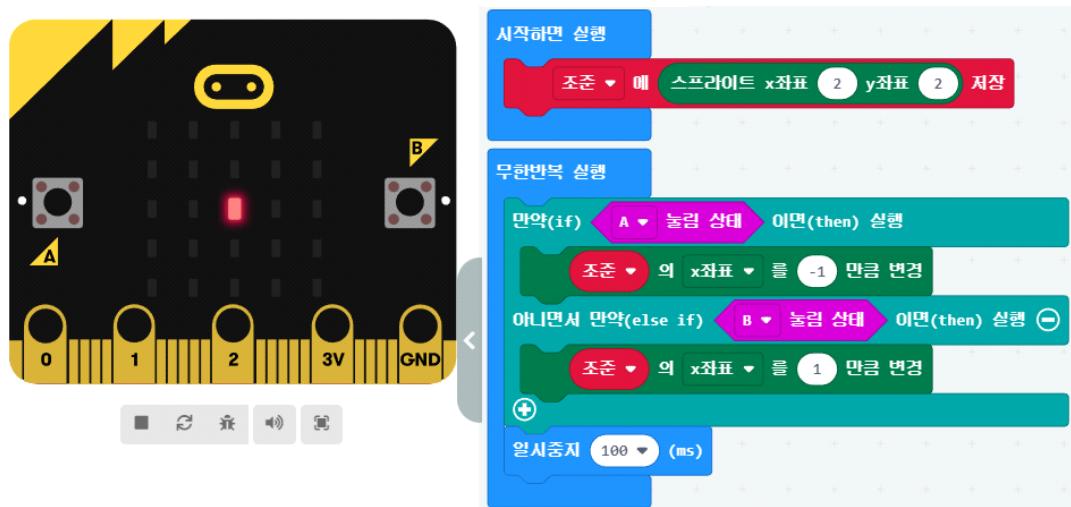


특정 위치에 스프라이트 생성

스프라이트의 기본 이동 방향은 오른쪽이며 그 방향을 변경할 수 있음

2) 실습 예제

변수를 만들어서 스프라이트를 그 변수에 담는 것도 가능합니다. 아래 프로그램이 바로 그 예시입니다. A 또는 B버튼을 눌러서 스프라이트를 움직이도록 만들어 보았습니다.



4. 과녁 맞추기 게임

1) 게임 설계 목표

| x | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| y | | | | | |
| 0 | | | | | |
| 1 | | | | | |
| 2 | ● | ● | ● | ● | |
| 3 | | | | | |
| 4 | | | | | |

- 과녁 스프라이트를 (2, 2)에 위치시킨다.
- 조준 스프라이트는 (0, 2) 위치에서 시작하여 (4, 2) 위치로 1칸씩 서서히 움직인 뒤 다시 (0, 2) 위치로 반복하여 좌우로 이동한다.
- 과녁과 조준의 위치가 일치된 상태에서 A버튼을 누르면 O를 출력, 일치되지 않았을 때 누르면 X를 출력한다.

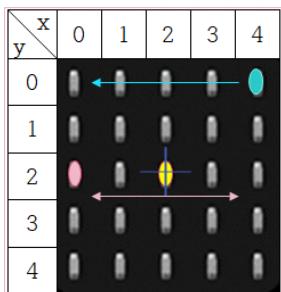
2) 초기구현



조준점의 시작 위치를 매번 다르게 하기 위해 ‘랜덤값’ 블록을 사용할 수도 있습니다.



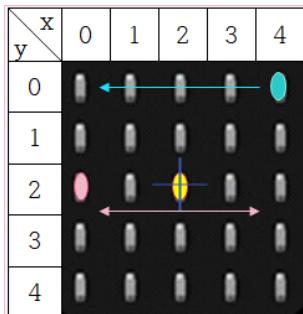
3) 게임 업그레이드 I



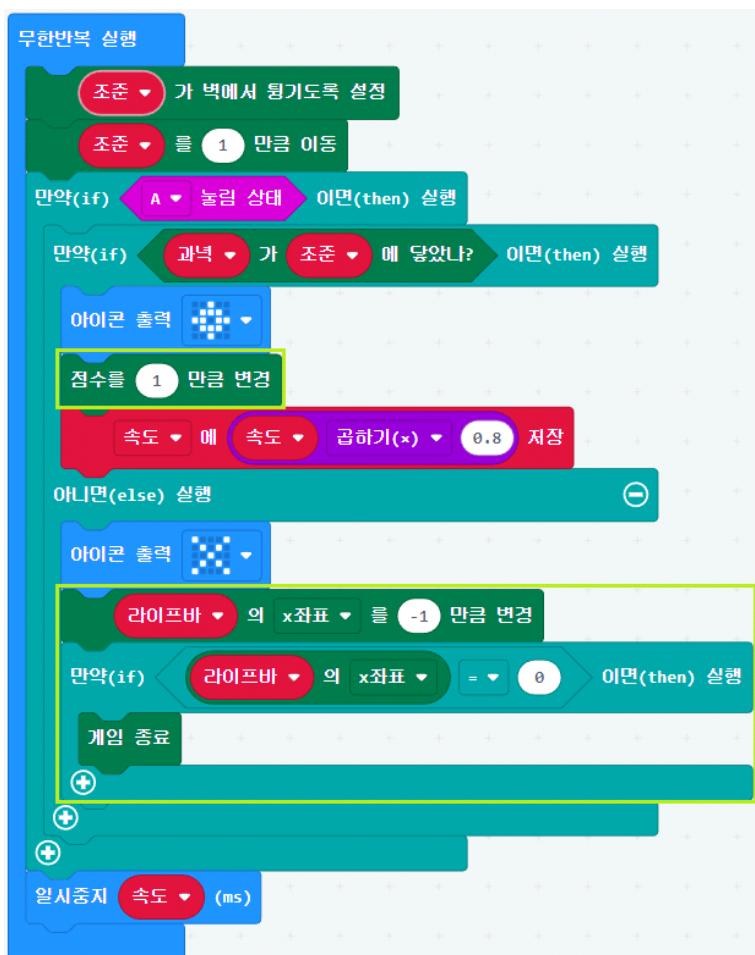
- 과녁의 이동속도가 400ms에서 시작하여 성공할 때마다 20% 씩 점점 빨라진다.
- B 버튼을 누르면 게임이 일시 정지되고 다시 누르면 재개된다.
- A+B 버튼을 누르면 게임이 재시작된다.

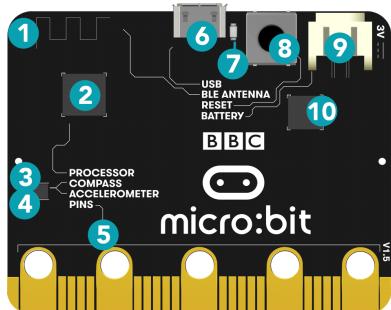


4) 게임 업그레이드 II



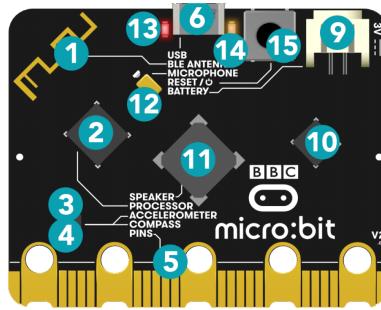
- 과녁에 명중하면 점수를 1점 올리고 그렇지 않으면 라이프를 1 감소(왼쪽 이동)시킨다.
- 상단의 첫 줄을 라이프를 표시줄로 활용한다(오른쪽 끝 점이 라이프 최대이며 여기에서 시작하여 라이프 바가 왼쪽 끝에 닿으면 GAME OVER 처리)





micro:bit V1 - Back

- 1. Radio & Bluetooth Antenna
- 2. Temperature Sensor & Processor
- 3. Compass
- 4. Accelerometer
- 5. Pins
- 6. Micro USB Socket
- 7. Single Red LED
- 8. Reset Button
- 9. Battery Socket
- 10. USB Interface Chip



micro:bit V2 - Back

- 1. Radio & Bluetooth Antenna
- 2. Temperature Sensor & Processor
- 3. Compass
- 4. Accelerometer
- 5. Pins
- 6. Micro USB Socket
- 7. Single Red LED
- 8. Reset Button
- 9. Battery Socket
- 10. USB Interface Chip
- 11. Speaker
- 12. Microphone
- 13. Red Power LED
- 14. Yellow USB LED
- 15. Reset & Power button

<버전 별 마이크로비트 내장 모듈>

마이크로비트에는 여러 가지 모듈이 이미 내장(built in)된 상태에서 출시되었습니다. 내장 모듈을 활용한 마이크로비트의 능력을 알아봅시다.

가. 라디오 & 블루투스 안테나

- 무선으로 정보를 송·수신 할 수 있습니다.

나. 온도센서

- 프로세서에 내장된 온도센서를 통하여 주변 온도를 측정할 수 있습니다.

다. 자기 센서

- 자기력을 감지하는 훌센서가 내장되어 나침반 등을 만들 수 있습니다.

라. 가속도 센서

- 가속도 센서로 기울어짐, 회전, 제스처를 감지할 수 있습니다.

마. 스피커

- 내장 스피커로 소리를 만들어 낼 수 있습니다.

바. 마이크

- 마이크로 주변 소리를 감지할 수 있습니다.

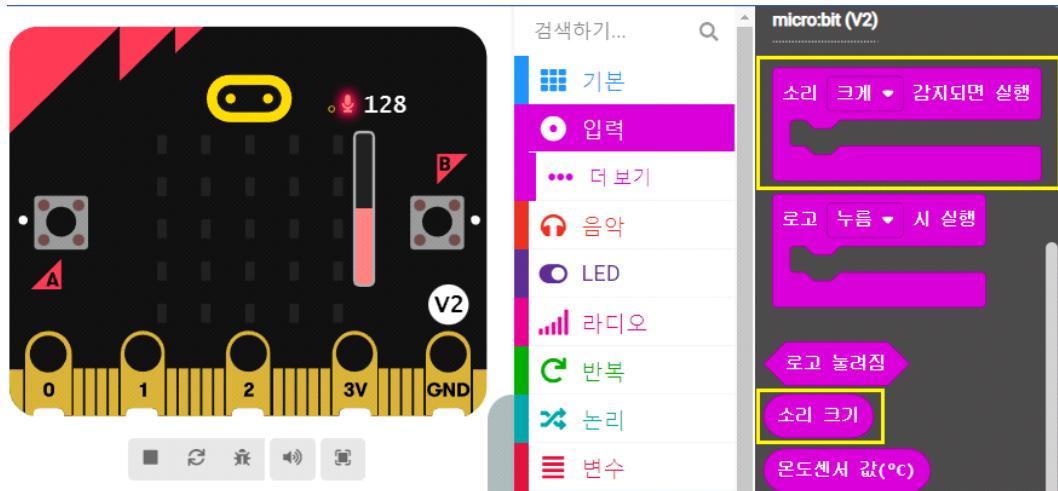
사. 터치센서

- 로고 그림에 터치센서가 내장되어 로고의 터치를 감지할 수 있습니다.

1. 마이크로 소리에 반응하기

1) 블록소개

소리 감지와 관련된 블록은 아래와 같습니다.



2) 실습 예제

박수를 칠 때마다 \uparrow , \downarrow 화살표를 번갈아 출력하는 프로그램을 만들어 봅시다.
먼저 메이크코드에서 새로운 프로젝트를 생성한 뒤 아래와 같이 코딩합니다.



큰 소리가 감지될 때마다 '소리감지' 변수의 값은 1씩 증가하게 됩니다. 이 숫자가 짹수이면 \uparrow 를, 홀수이면 \downarrow 를 출력하기 위한 방법으로 'remainder of (소리감지) \div 2'를 계산식을 사용하였습니다. 2로 나눈 나머지가 0이면 짹수라는 뜻이지요.

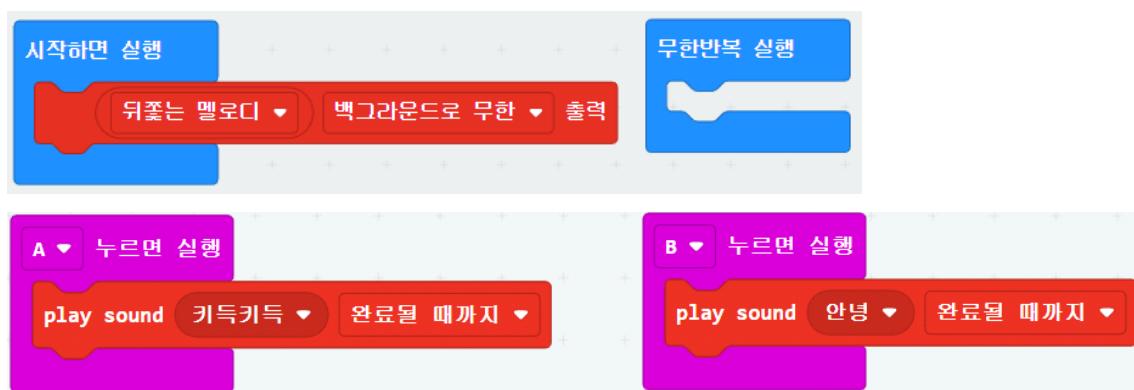
2. 스피커로 소리 출력하기

1) 블록 소개

마이크로비트는 내장된 스피커를 사용하여 여러 가지 소리를 출력할 수 있습니다. 크게 멜로디와 사운드로 분류할 수 있는데 멜로디는 짧은 음악, 사운드는 효과음을 의미합니다. 아래 그림의 두 가지 블록을 확인하세요.



2) 실습 예제



위와 같이 예시 프로그램을 작성해 봅시다. ‘뒤쫓는 멜로디’가 배경 음악으로 출력되는 동안 A 또는 B 버튼을 누르면 효과음이 함께 출력되는 것을 알 수 있습니다. 소리 출력 기능을 어떻게 활용하면 좋을까요? 이전 단원에서 코딩한 과녁 맞추기 게임에 사운드 효과를 추가하여 게임의 완성도를 높여 봅시다.

3. 가속도 센서로 수평계 만들기

1) 가속도 센서란?

가속도 센서란 사물에 얼마만큼의 중력가속도(힘)가 걸리고 있는지 측정하는 센서입니다. 3축(x, y, z)에 걸리는 중력가속도를 측정하여 약간의 계산을 통해 기울기를 알아낼 수 있습니다. 메이크코드에서는 다음 블록으로 계산된 결과를 얻어낼 수 있습니다.



2) 수평계 만들기

수평계는 평면의 기울기를 측정할 수 있기 때문에 건축 현장에서 벽돌을 직선 방향으로 올바르게 쌓았는지 지면에 직각으로 벽을 세웠는지 등을 확인할 때 사용합니다. 기울기 측정 블록을 이용하여 이러한 수평계를 만들어 봅시다.



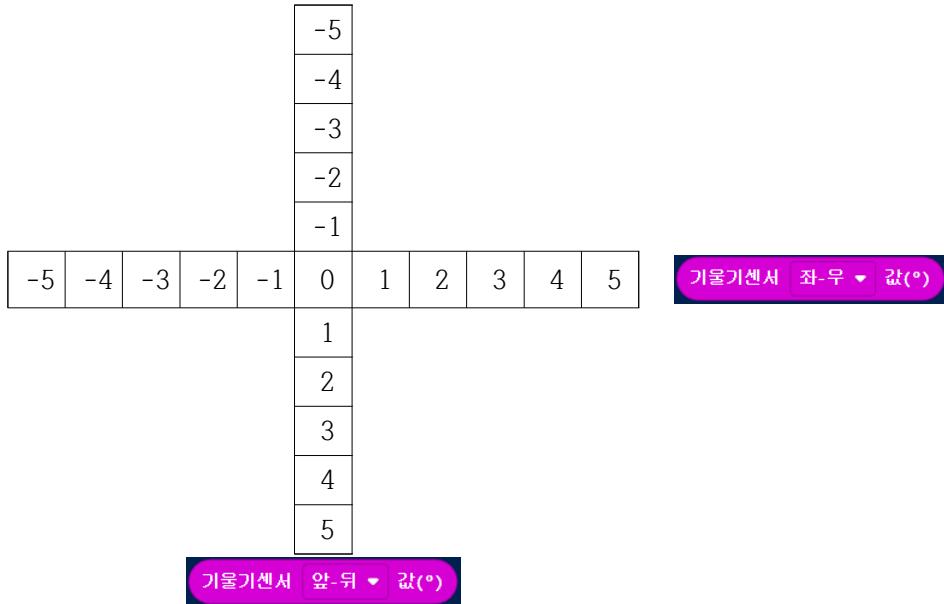
나누기 2 대신 큰 숫자를 집어넣을수록 기울기에 덜 민감하게 반응하게 됩니다.

3) 기울기 센서의 값

기울기센서 좌-우 ▾ 값(°) 블록은 왼쪽으로 많이 기울어질수록 큰 음수값을 오른쪽으로 많이 기울어질수록 큰 양수값을 가지게 됩니다.

기울기센서 앞-뒤 ▾ 값(°) 블록은 앞쪽으로 많이 기울어질수록 큰 음수값을 뒤쪽으로 많이 기울어질수록 큰 양수값을 가지게 됩니다.

그리고, 기울어짐 없이 정확히 수평을 유지할 때 0값을 갖게 되지요.



아래 프로그램을 통해 기울기 센서의 값을 눈으로 확인할 수 있습니다.

