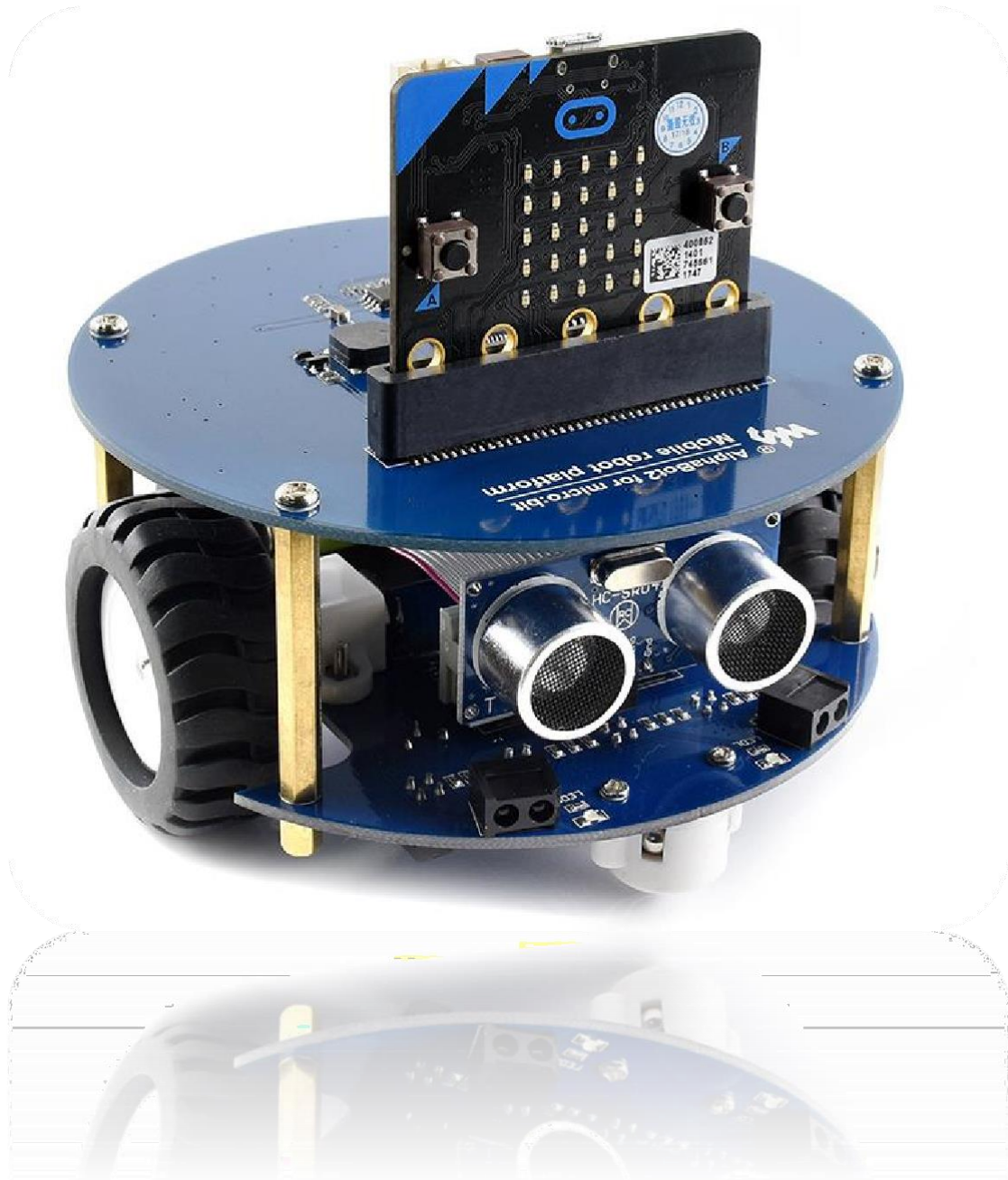




# Alphabot2 for micro:bit

## 사용자 설명서



## 내용

머리말 .....	5
micro:bit .....	5
참고 .....	6
Chapter 1 LED 점등 .....	8
JavaScript Blocks editor.....	8
블록 코딩 .....	9
LED 점등 .....	10
하트 모양 LED .....	13
화면에 글자 출력하기 .....	14
Chapter 2 버튼 .....	15
버튼 누르기 1 .....	15
버튼 누르기 2.....	16
Chapter 3 센서.....	18
온도 센서 .....	18
가속도계 .....	19
자력계 .....	21
광 센서 .....	23
Chapter 4 음악 .....	25
노래 재생 .....	26

피아노 치기 .....	27
노래 재생 2 .....	28
<b>Chapter 5 RGB LED.....</b>	<b>29</b>
패키지 추가.....	30
LED 점등 .....	31
다양한 색상 표시 .....	32
깜박이는 LED .....	33
무지개색 LED 점등 .....	34
<b>Chapter 6 로봇 동작시킴 .....</b>	<b>35</b>
패키지 추가.....	36
모터 동작 .....	37
로봇 동작 1 .....	38
로봇 동작 2 .....	40
로봇 동작 3.....	41
<b>Chapter 7 적외선 장애물 회피 .....</b>	<b>42</b>
적외선 센서 감지 .....	43
장애물 회피 .....	44
<b>Chapter 8 블루투스 .....</b>	<b>45</b>
패키지 추가.....	45
블루투스 연결 1.....	46

블루투스 연결 2.....	51
D-PAD 컨트롤러 1 .....	53
D-PAD 컨트롤러 2.....	55
<b>Chapter 9 2.4GHz 라디오 통신(무선 통신).....</b>	<b>56</b>
패키지 추가.....	56
무선 통신.....	56
마이크로 비트용 조이스틱 1 .....	58
마이크로 비트용 조이스틱 2.....	62
가속도계를 이용해 <b>Micro:bit</b> 의 방향으로 제어 .....	64
<b>Chapter 10 초음파 장애물 회피 .....</b>	<b>66</b>
초음파 센서 .....	67
장애물 회피 .....	68
<b>Chapter 11 라인 트래킹 .....</b>	<b>69</b>
트래커 (추적기) 센서 .....	70
원본 데이터.....	71
보정된 데이터 .....	74
라인 트래킹(추적).....	76

## 머리말

Alphabot2 로봇 키트는 BBC micro:bit를 메인 컨트롤러로 사용하고 여러 기능 모듈과 결합하여 초등학생과 같은 어린 아이들이 라인 추적, 장애물 회피, 초음파 거리 측정, 서보 모터 작동 블루투스 원격 제어 등과 같은 로봇 동작을 쉽게 경험할 수 있습니다.

## MICRO:BIT

BBC micro:bit는 어린이와 초보자를 위한 프로그래밍 학습을 위한 주머니 크기의 컴퓨터입니다.

- **Nordic nRF51822**
  - 16 MHz 32비트 ARM Cortex-M0 마이크로 컨트롤러
  - Bluetooth® 4.0 low energy/2.4GHz RF SoC
  - 16kB RAM
  - 256kB Flash
- **Freescale KL26Z** – 48 MHz ARM Cortex-M0+ MCU
- 자력계 – Freescale MAG3110 3축 자력계, I2C 인터페이스
- 가속도계 – Freescale MMA8652 3축 가속도계, I2C 인터페이스
- 전원 연결 – 3V 전원 공급
- 리셋 버튼 – 시스템 초기화 하기
- 마이크로 **USB** 커넥터 – PC연결, 코드 다운로드, 시리얼(직렬) 통신 등에 사용됩니다.

- **5x5 LED display** – 5x5 LED grid
- **버튼 A/B** – 프로그램 가능한 버튼
- **입력 및 출력 핀** – I/O 링 다섯 개, 에지 커넥터(SPI, UART, I2C, Analog, PWM, etc.) 20핀
- **치수** – 5cm x 4cm

## 참고 사항

1. 패키지마다 구성품이 다를 수 있습니다. 이 설명서에서는 모든 구성 요소를 설명하며, 일부 구성요소는 구매하신 패키지에 포함되지 않을 수 있습니다.
2. 사용 설명서, 회로도, 데모 코드와 데이터시트 등 모든 리소스는 아래 링크에 있는 Wiki에서 다운로드 받을 수 있습니다.  
  
➤ [https://www.waveshare.com/wiki/AlphaBot2\\_for\\_micro:bit](https://www.waveshare.com/wiki/AlphaBot2_for_micro:bit)
3. 본 설명서에서는 제한된 시간으로 인해 오류가 있을 수 있으니, 발견하시면 waveshare팀에 email 문의 주시기 바랍니다.<sup>1</sup>
4. Alphabot2가 손상되는 것을 방지하기 위해 처음 Alphabot2을 시작할 때, 이 설명서를 읽고 따르는 것을 추천 드립니다.
5. 처음 시작할 때, 구입하신 모든 구성품이 포함되어 있는지 확인하시기 바랍니다.  
  
초보자라면, 처음부터 따라 하면서 마이크로 비트 사용법을 먼저 익히는 것을 추천 드립니다.

---

<sup>1</sup> Email: [service@waveshare.com](mailto:service@waveshare.com)

6. 처음, 전원 스위치를 켜고, 배터리에 전원이 정상적으로 공급되는지 확인해야 합니다.  
그렇지 않으면, 전원 LED의 불빛이 약해지며, Alphabot2이 제대로 작동하지 않습니다.
7. 마이크로 비트를 Alphabot2에 꽂거나 빼기 전에 반드시 전원을 꺼 마이크로 비트에 손상이 가지 않도록 하십시오.
8. 마이크로 비트의 LED 매트릭스를 바깥쪽으로 향하도록 유지하세요.
9. 이 설명서에서 언급된 데모 코드는 모두 위키에 제공되어 있습니다. 데모 코드는 Microsoft의 make code micro:bit 프로그래밍 환경을 기반으로 합니다. 데모 코드는 모두 hex 파일이며, 마이크로 비트에 직접 복사하여 실행할 수 있습니다. ( 마이크로 비트를 PC에 연결하면 포터블 드라이브로 인식 되는데, 이 드라이브에 .hex 파일을 복사하여 실행할 수 있게 됩니다. ) 데모 코드를 미스코드 웹사이트로 가져와서 수정 및 플래싱 할 수도 있습니다.
10. 여기에 제공된 모든 사양 및 데모 코드는 언제든지 사전 통지 없이 변경될 수 있습니다.

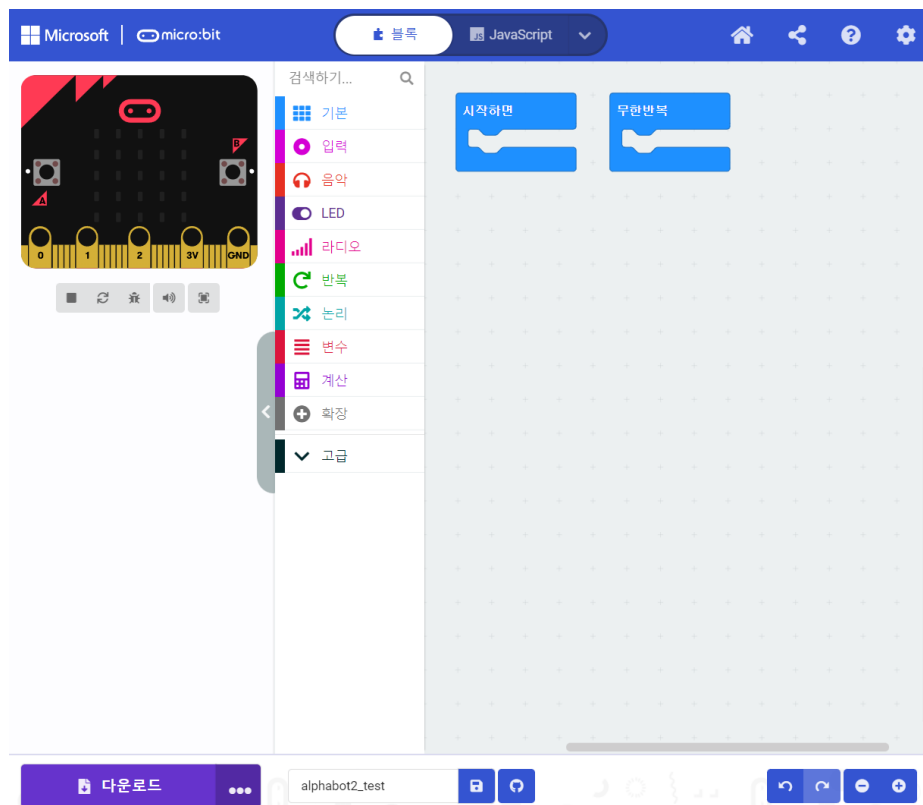
## CHAPTER 1 LED 점등

마이크로 비트에는 5X5로 25개의 개별 프로그래밍이 가능한 LED가 있어 텍스트, 숫자, 이미지를 표시할 수 있습니다. 이러한 LED를 코딩하는 방법을 배워보도록 하겠습니다.

### JavaScript Blocks editor

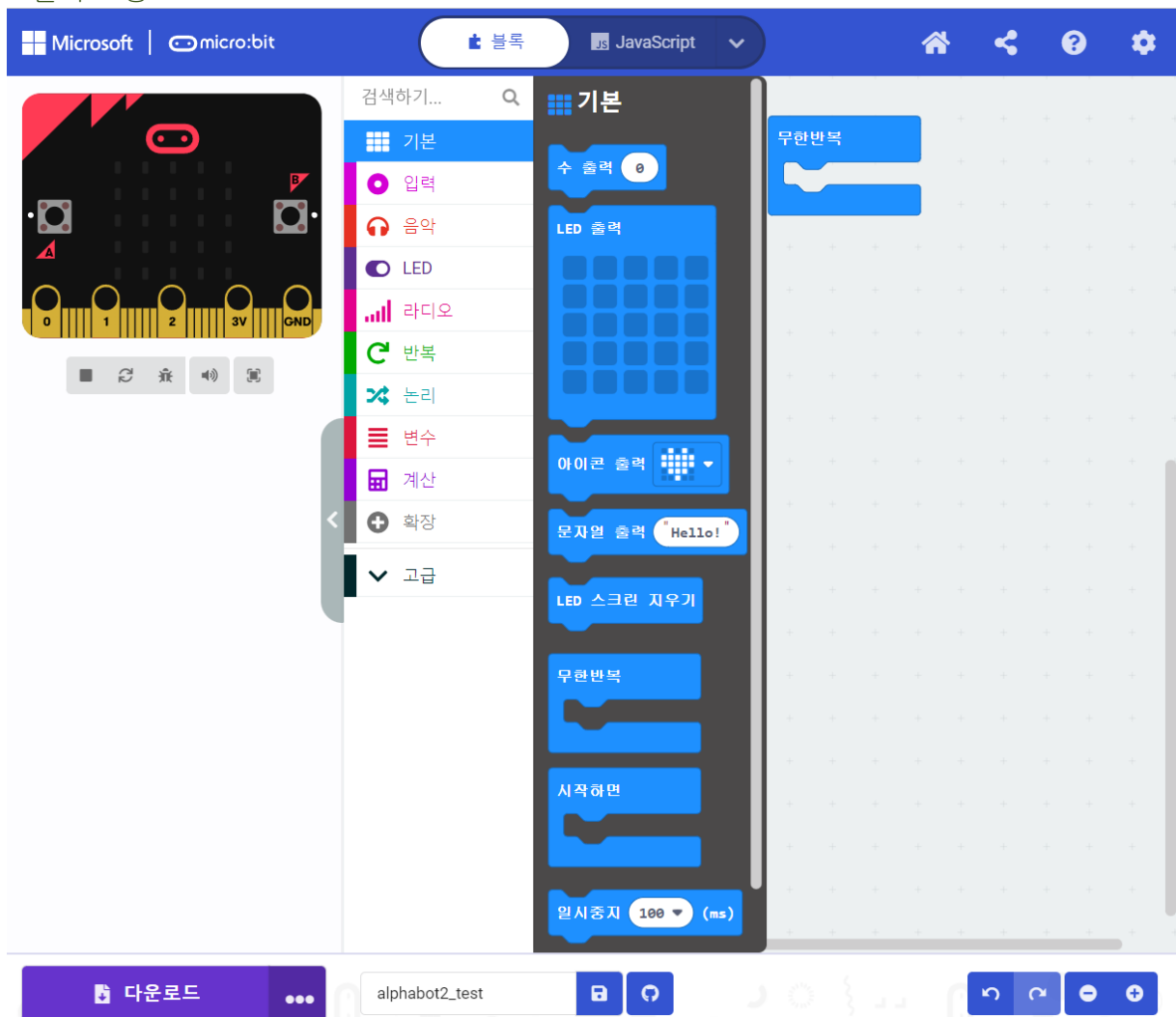
온라인 자바스크립트 블록 에디터 링크: <https://makecode.microbit.org/#>

자바스크립트 블록 에디터를 시작합니다. 위쪽에는 메뉴 모음, 가운데에는 블록, 왼쪽은 시뮬레이션, 오른쪽은 블록 및 자바스크립트 편집 영역으로 구분되어 있습니다.

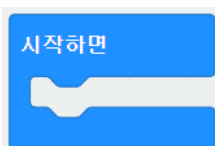




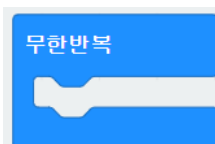
블록 코딩



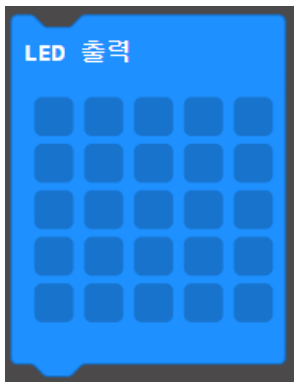
[기본] 패키지의 블록의 주로 LED 제어와 같은 기본 작동에 사용됩니다.



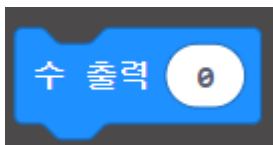
:처음에 코드를 한 번 실행합니다.



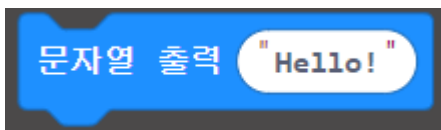
:백그라운드에서 코드를 무한히 반복하여 실행합니다.



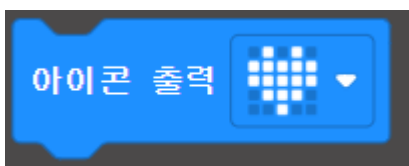
: LED에 이미지를 출력합니다.



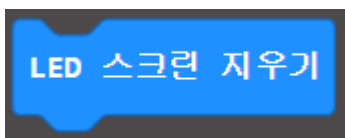
: LED에 숫자를 출력합니다.



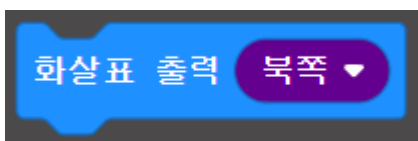
: LED에 텍스트를 한 번에 한 글자씩



: LED에 선택한 아이콘을 그립니다.



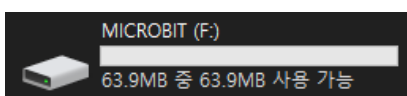
: 모든 LED를 끄게 됩니다.



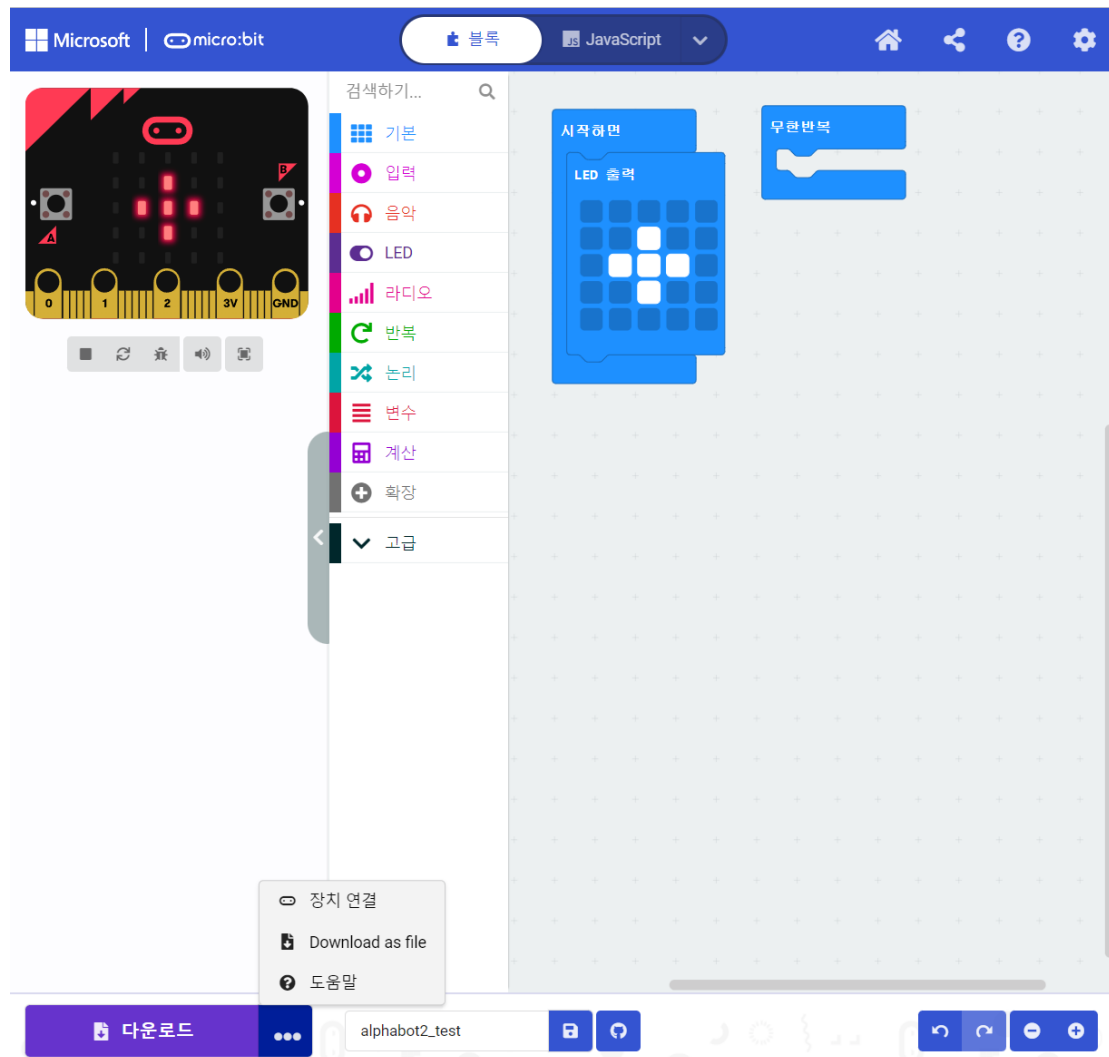
: LED에 화살표를 그립니다.

## LED 점등하기

마이크로 USB 케이블을 이용해 마이크로 비트와 컴퓨터를 연결하면, micro:bit가 컴퓨터에 "MICROBIT"라는 드라이브로 표시됩니다.



이제 위에서 설명한 블록을 이용하여 LED를 켜는 첫 번째 코드를 시작해 보도록 하겠습니다.

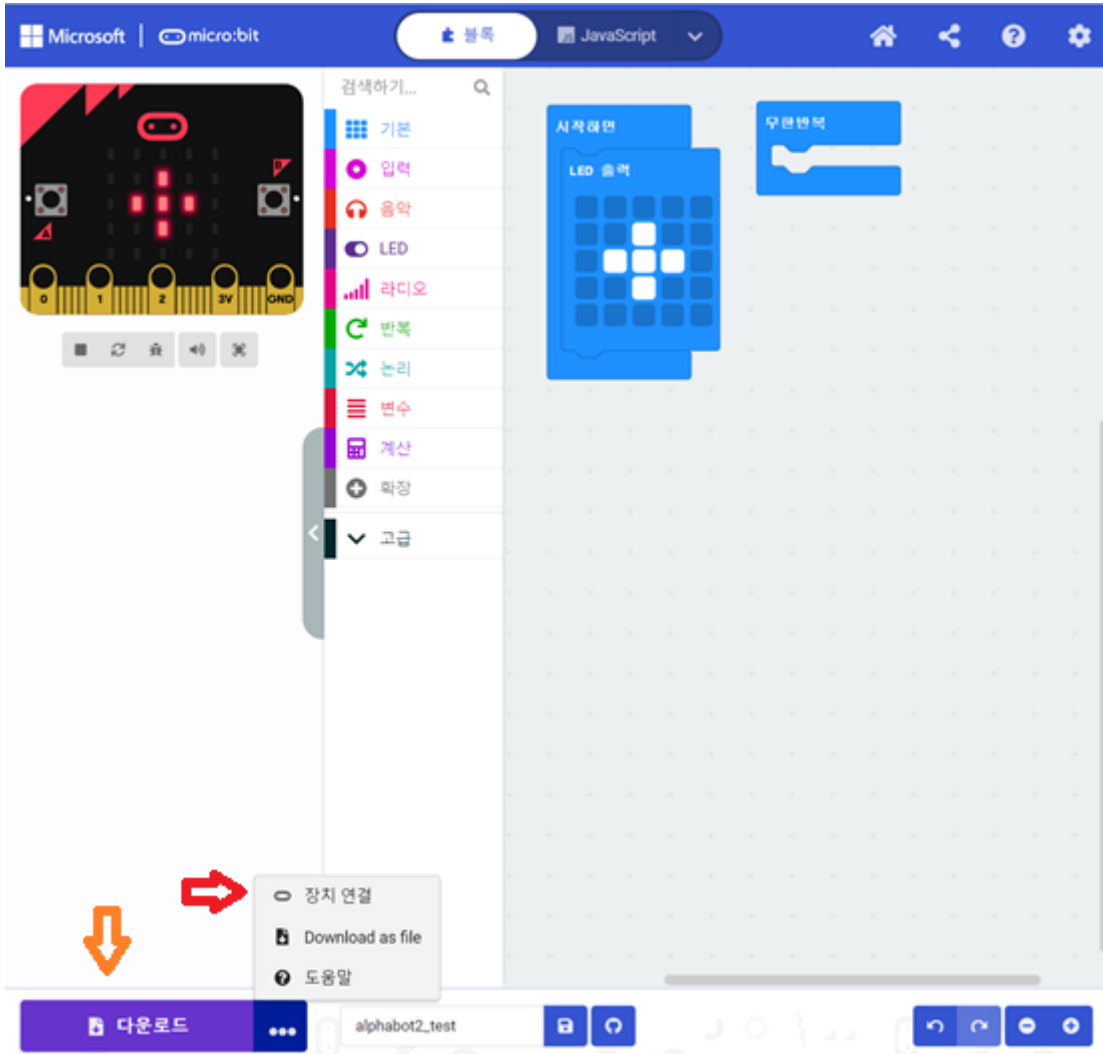


**[LED 출력]** 블록을 클릭하여 원하는 형태로 **led**를 설정하면 왼쪽 화면에 시뮬레이션된

마이크로 비트를 재생시키면 **LED**가 켜지는 것을 확인할 수 있습니다.

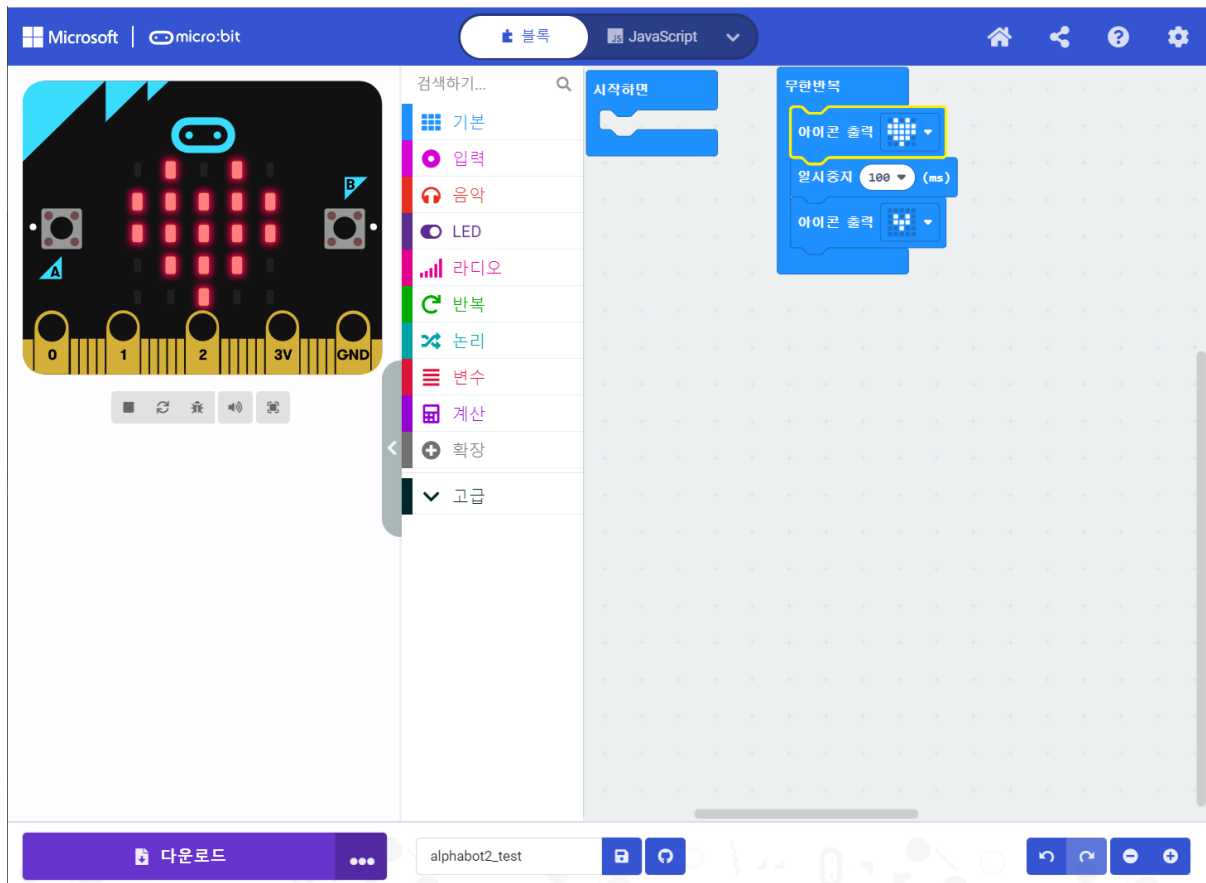
아래에 있는 다운로드를 클릭하여 “**MICROBIT**” 드라이브에 코드를 다운로드 합니다.

블록 코딩 프로그램에서 장치 연결을 한 뒤에 다운로드 버튼을 눌러 바로 다운로드 할 수도 있고, **HEX**파일을 저장한 뒤, 드라이브에 복사할 수도 있습니다.



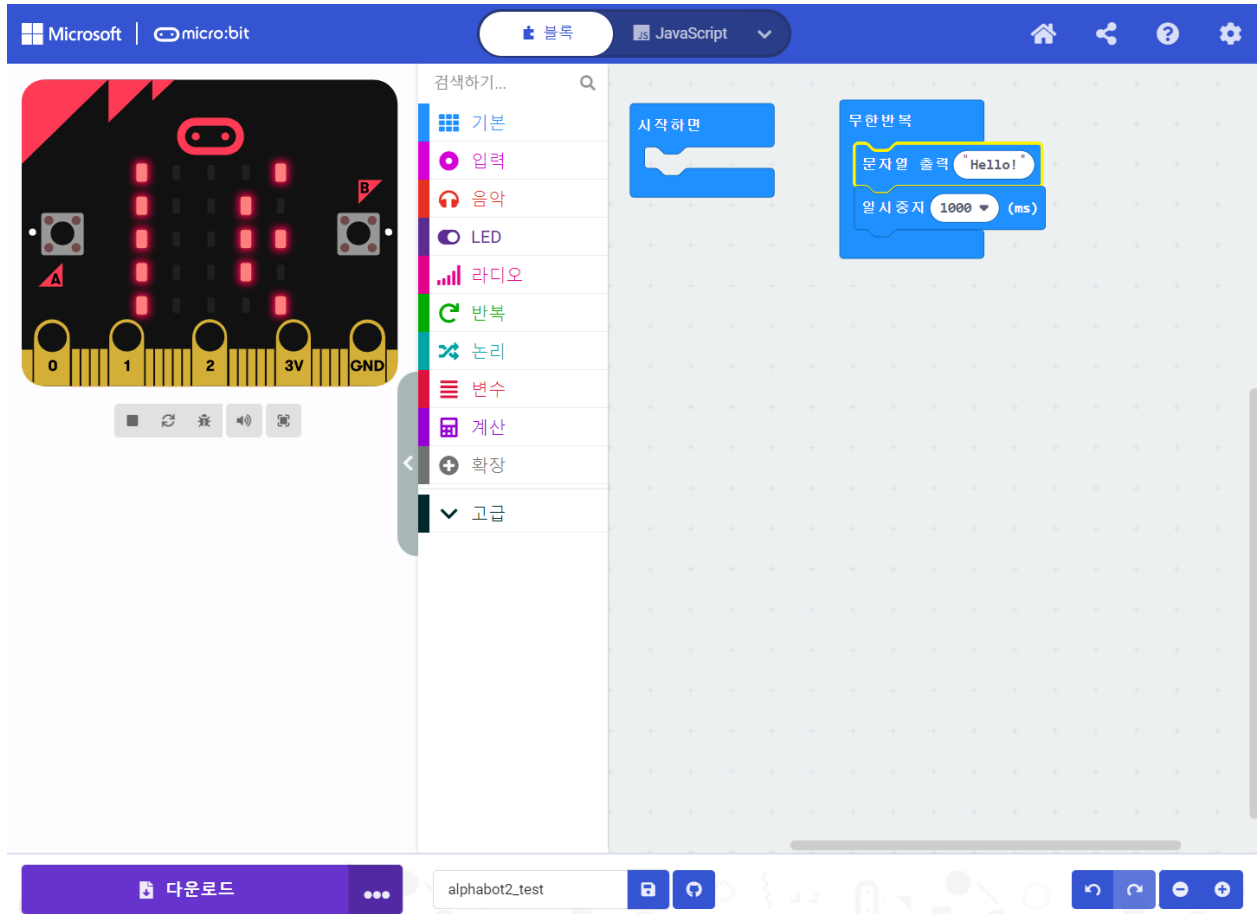
## 하트 모양 LED

이 코드에서는 [아이콘 출력] 블록을 사용하여 큰 하트를 표시한 다음, 작은 하트를 표시합니다. [무한반복] 블록을 통해 심장이 뛰는 것처럼 보이게 할 수 있습니다.



## 화면에 글자 출력하기

이번 코드에서는 [문자열 출력] 블록을 사용하여 LED에 “Hello!” 텍스트를 표시합니다.  
 텍스트는 LED에서 한 글자씩 스크롤 됩니다.

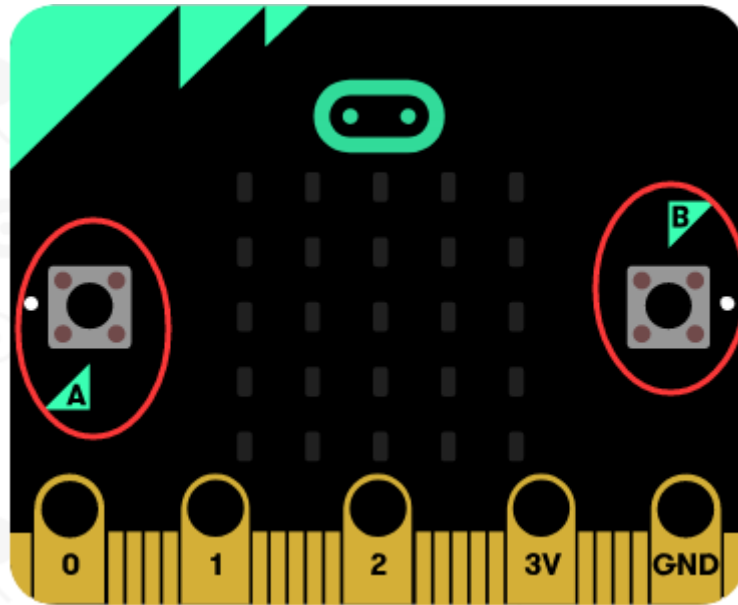


### 【Tips】

블록을 삭제하려면 마우스 오른쪽 버튼을 눌러 삭제하거나 키보드 DELETE 버튼을 누르거나, 왼쪽(블록 꺼내는 부분)으로 드래그하면 됩니다..

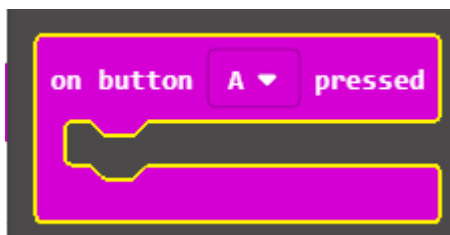
## CHAPTER 2 버튼

마이크로비트 전면에 A와 B라는 프로그래밍 가능한 버튼 두 개가 있는데, 이번 코드에서는 이 버튼으로 코딩 해보겠습니다.



### 버튼 누르기 1

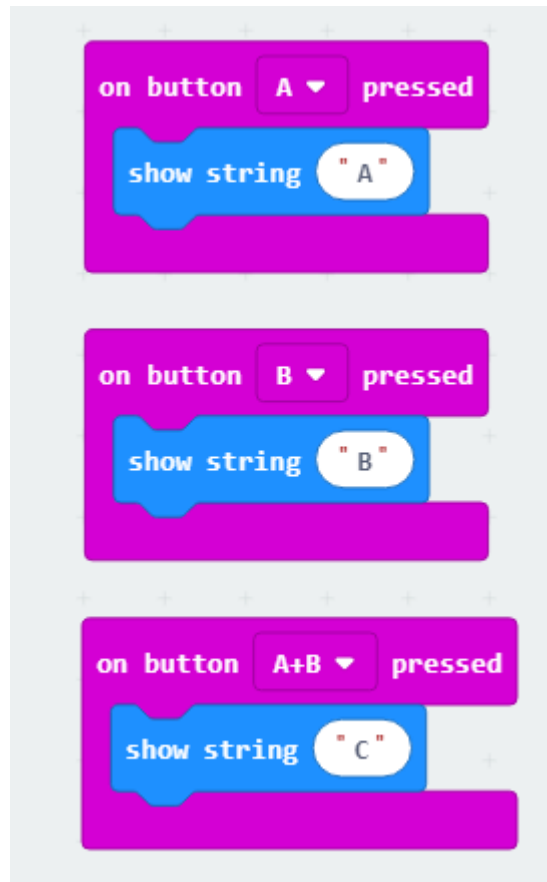
이번 코딩에서 사용하는 핵심 블록은 [입력] 패키지에 포함되어 있습니다. 이 블록을 사용하면 마이크로 비트 전면 버튼인 A와 B 버튼을 누르면 코드가 실행이 됩니다.



버튼을 프로그래밍하여, 버튼 A와 버튼 B를 눌렀을 때, 각각 A와B를 표시하고, 두 버튼을 모두 눌렀을 시에는 C를 표시합니다.

1. 새 프로젝트를 생성하고, 버튼 A, 버튼 B, 혹은 모든 버튼을 눌렀을 때, 코드가 실행되도록 버튼 누름 블록에 배치합니다.

2. **[버튼을 눌렀을 때]** 블록 안에 **[문자열 출력]** 블록을 배치하여, LED 디스플레이에 문자열을 표시합니다. 아래와 같이 스크립트를 복제하여 완성합니다.



3. 마이크로 비트를 PC에 연결하고, [다운로드]를 클릭하여 코드를 보드에 다운로드 합니다. 마이크로 비트의 버튼을 눌러 예상 문자열이 표시되는지 확인합니다.

## 버튼 누르기 2

버튼을 감지하는 또 다른 방법이 있습니다. 마지막 세션에서는 버튼을 눌렀을 때, 코드가 실행되었습니다.

버튼을 눌렀을 때, 코드가 실행되었지만, 여기서는 **[만약 [] 라면]** 블록과 **[버튼이 눌렀을 때]** 블록을 사용하여 동일한 작업을 수행합니다. 코드를 직접 실행하는 대신 버튼을 누르면 Boolean 값 true를, 버튼을 누르지 않으면 false를 변환합니다.

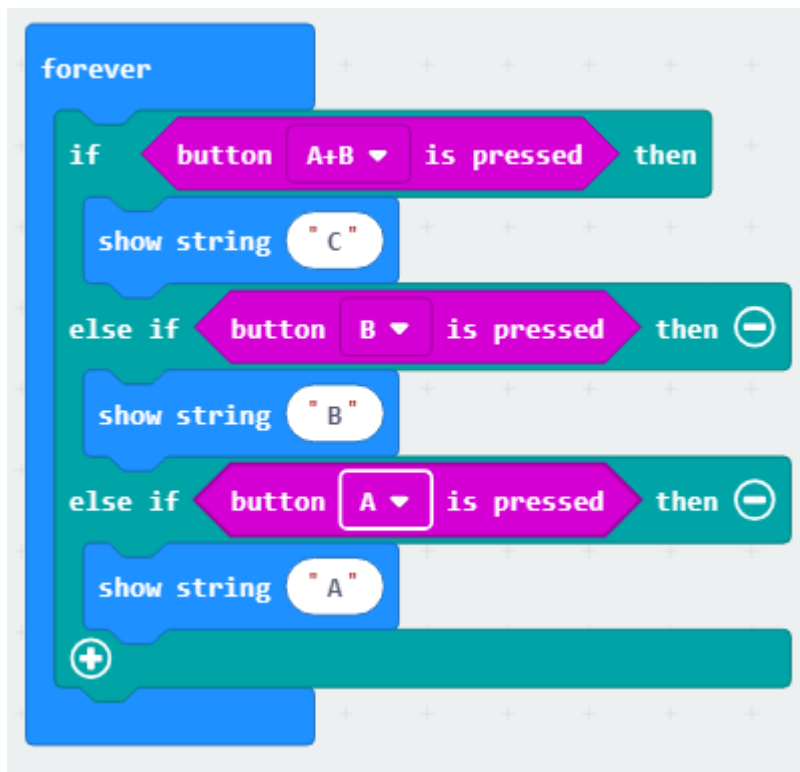
\* Boolean 값 : True 혹은 False만 존재합니다.





[만약 ~이면] 블록은 [논리] 패키지에 있고, [버튼이 눌렸을 때] 블록은 입력 패키지에 있습니다.

1. [만약 ~이면] 블록을 [무한반복] 블록 안에 배치하고, [A 버튼 눌림] 블록을 그 위에 조건 부분으로 넣습니다.
2. [문자열 출력] 블록을, 조건이 충족되었을 시에 동작시킵니다. ⊕ 아이콘을 사용하여 아래와 같이 코드를 작성합니다.



3. 마이크로비트를 PC에 연결하고, [다운로드] 버튼을 눌러 코드를 보드에 다운로드 합니다. 마이크로비트의 버튼을 눌러 예상 문자열이 표시되는지 확인합니다..

**【Tips】**

일부 사용자는 이 코드에서 A 또는 B 대신 A+B가 먼저 감지되는 것을 알 수 있습니다. 감지 순서를 변경하여 어떤 일이 발생하는지 확인할 수 있습니다.

## CHAPTER 3 센서

마이크로 비트는 온도 센서, 가속도계, 나침반 등을 통합하고 있습니다. 이 장에서는 이러한 센서를 사용하는 방법을 배웁니다.

### 온도 센서

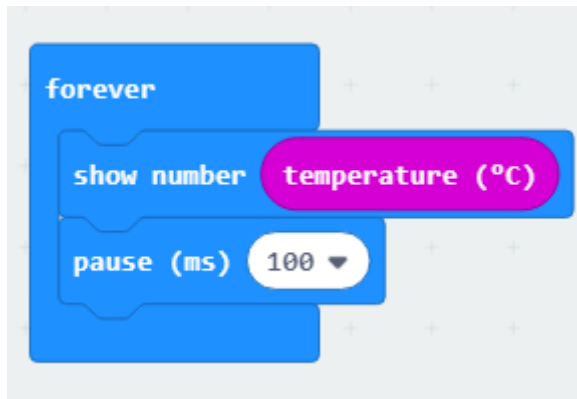
온도 센서는 마이크로 비트가 현재 주변 온도를 섭씨 단위로 감지할 수 있게 해줍니다. 실제로 칩의 온도를 감지하는 데 사용되는 메인 칩 NRF51822에 내장되어 있습니다. 칩의 온도는 주변 온도에 따라 조금씩 다르지만 대략적인 값을 구하는데 사용할 수 있습니다.



**[입력]** 패키지 안에 온도 블록이 있습니다, **[수 출력]** 블록과 결합하면 실내(주변)의 온도를 Led 디스플레이에 표시할 수 있습니다..

1. **[수 출력]** 블록을 **[무한반복]** 블록 안에 배치합니다.
2. 온도 블록을 배치합니다.

3. 딜레이를 위한 일시 정지 블록을 추가합니다..



4. 마이크로 비트에 코드를 다운로드하고, CPU를 잠시 터치하면, 온도가 높아지는 것을 알 수 있습니다.

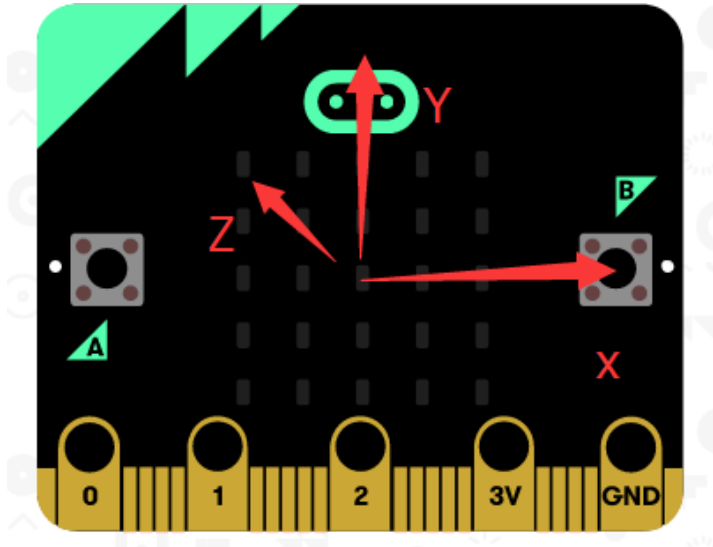
## 가속도계

가속도계 MMA8652는 마이크로 비트의 왼쪽 하단에 통합되어 있습니다.



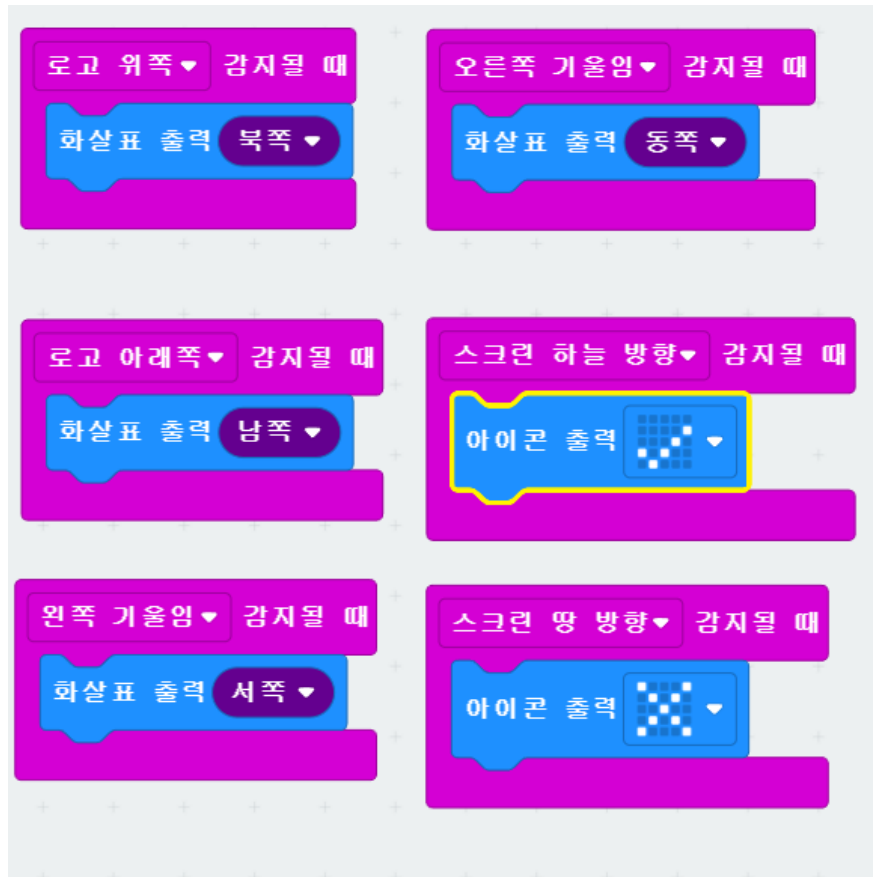
가속도는 움직이는 경향이 있습니다. 물체에 한 방향으로 힘을 가하면 물체는 그 방향으로 움직이려는 경향입니다. 예를 들어, 우리는 항상 중력을 받고 있으며, 발 아래에 지지대가 없으면 자유 낙하 하게 됩니다. 우리는 떨어지는 경향이 있는 데, 이를 중력 가속도라고 합니다.

X,Y,Z 방향의 좌표를 측정합니다. 이 세 가지 값으로 가속도의 방향을 구할 수 있습니다. 중력 가속도만 있는 경우 방향은 아래쪽입니다. 마이크로 비트의 가속도를 구하기 위해 x,y,z, 좌표 값을 계속 측정하고 흔들림, 기울기 등과 같은 동작을 감지합니다.



1. [ ( ) 감지될 때 ] 블록을 배치하고, 작업을 선택합니다.
2. [ 화살표 출력 ] 블록을 배치하고, 화살표를 선택합니다.

3. 아래와 같이 스크립트를 만들어 완성합니다.



4. 마이크로 비트에 코드를 다운로드하고, 마이크로 비트를 흔들어 화살표가 바뀌는지 확인합니다.

## 자력계

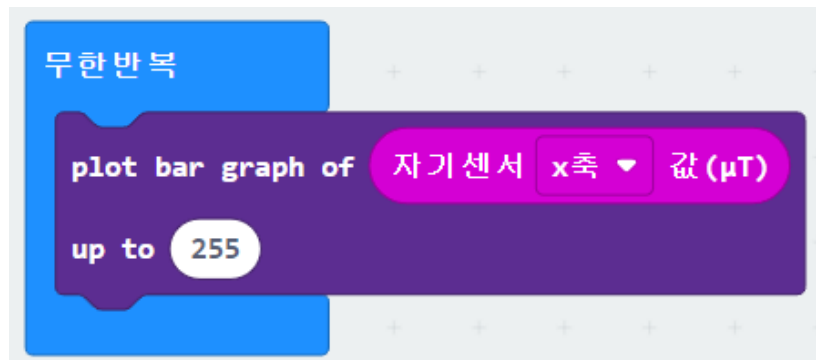
나침반 칩은 빨간색 화살표로 표시된 MAG3110입니다. 나침반은 x,y,z 좌표도 측정하여 지구의 자기장을 감지할 수 있습니다.



## 자력:

자력의 세기를 막대 그래프(LED)로 표시하는 스크립트를 만듭니다.

1. [LED] 패키지에 있는 [plot bar graph of] 블록을 [무한반복] 블록에 배치합니다.
2. [plot bar graph of] 블록에 [자기센서 값 (uT)] 블록을 배치하고, 최대값을 255로 설정합니다.
- \* [자기센서 값 (uT)]은 [입력] -> 더보기에서 찾으실 수 있습니다.



3. 마이크로 비트에 스크립트를 다운로드합니다.
4. 마이크로 비트에서 코드를 처음 실행하게 되면 “화면을 채우려면”이라는 텍스트가 스크롤되어 보정 단계를 먼저 수행해야 함을 알려줍니다. 그런 다음 마이크로 비트를 기울여, LED 화면을 채워야 합니다. 모든 LED가 켜지면 보정이 완료되었음을 알리는 스마일이 표시됩니다.
5. 예상되는 결과로는 자성이 없는 경우 몇 개의 LED가 켜지고, 스피커<sup>3</sup>를 마이크로 비트에 닿으면 더 많은 LED가 켜지는 것입니다. 결과가 예상대로 나오는 지 확인합니다.

## 나침반

가속도계를 사용하면, 중력의 방향, 즉 아래쪽을 알 수 있습니다

자력계는 지구의 자기장, 수평 구성 요소의 영향을 받습니다.

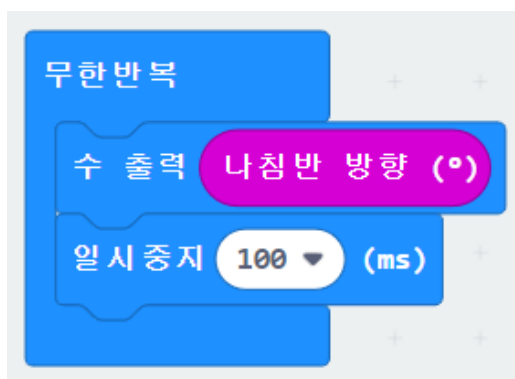
---

<sup>3</sup> 대부분의 스피커에는 자석이 있습니다.

지자기 방향은 항상 지자기 북쪽을 가리킵니다. 가속도계와 자력계를 결합하면 북쪽 방향을 구할 수 있습니다.

**【참고 사항】** 나침반을 사용하는 경우, 마이크로 비트는 다른 자기장의 영향을 받지 않도록 정적 상태를 유지해야 합니다.

1. [수 출력] 블록을 [무한반복] 블록에 배치합니다.
2. [입력] 패키지에서 [나침반 방향 (°)]를 찾습니다.
3. 딜레이를 위해 [일시중지] 블록을 추가합니다.



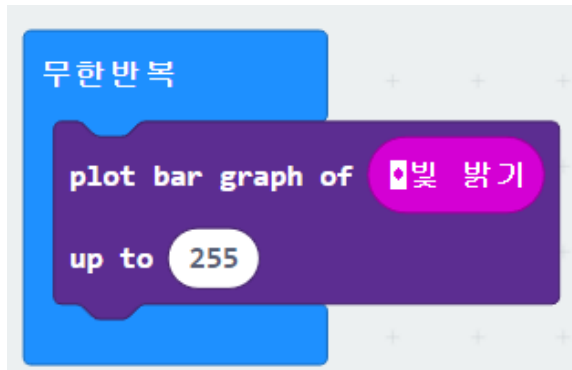
4. 스크립트를 마이크로 비트로 다운로드 하여 사용해 보세요.
5. 다운로드 후 보정도 수행해야 합니다.

## 광 센서

마이크로 비트에는 실제 광 센서가 없지만, 마이크로 비트 전면에 있는 LED 매트릭스를 입력으로 바꾸고 전압의 감쇠 시간을 샘플링 하여 주변 광을 감지할 수 있습니다.

1. [plot bar graph of] 블록을 [무한반복] 블록으로 배치합니다.

2. [ 빛 밝기 ] 블록을 [ plot bar graph of ] 블록으로 드래그하고 최대값을 255로 설정합니다.



3. 마이크로 비트에 스크립트를 다운로드하고 4번을 시도해 보십시오.
4. 휴대폰의 손전등을 열고 LED에 불이 들어오도록 합니다..



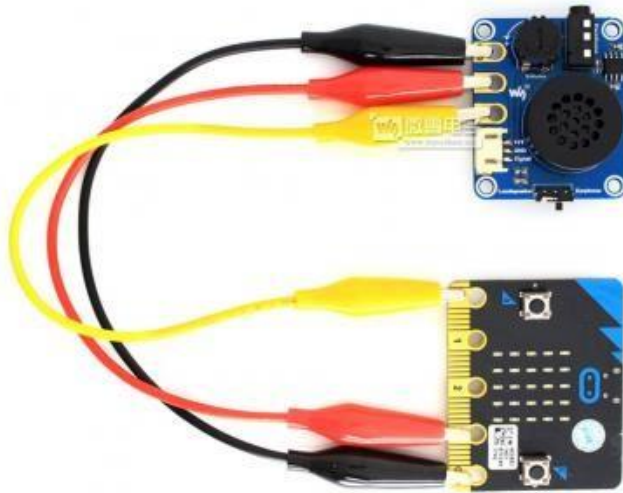
## CHAPTER 4 음악

마이크로 비트에는 부저나 스피커가 없기 때문에 마이크로 비트로 소리나 음악을 재생하려면 외부 스피커를 연결해야 합니다. Alpha2 마이크로 비트용 확장 보드에 버저가 있어 여기에 마이크로 비트를 끼우면 됩니다.

【참고】 이번 챕터를 시작하시기 전에 Alphabot2를 micro:bit와 조립하는 것이 좋습니다.



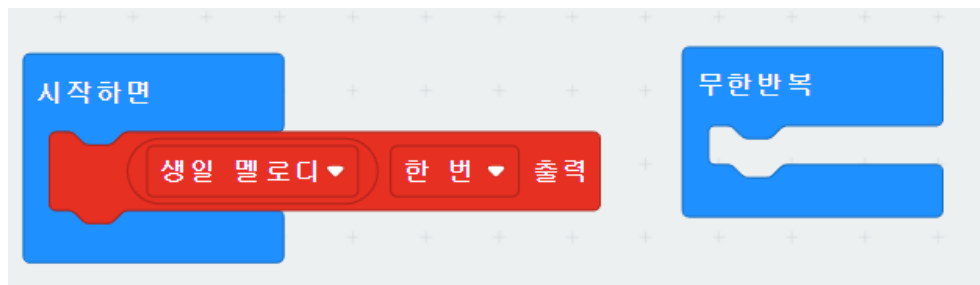
다음 사진을 통해 P0 핀을 통해 버저/스피커를 마이크로 비트에 연결할 수도 있습니다.



## 노래 재생

노래(소리)와 관련된 블록은 **[음악]** 패키지에서 찾으실 수 있습니다.

1. **[(노래)(n번)출력]** 블록을 **[시작하면]** 또는 **[반복하면]** 블록 중에 선택하여줍니다.
2. **[생일]** 과 같은 멜로디를 하나 선택한 뒤, 한 번만 반복되도록 설정합니다..



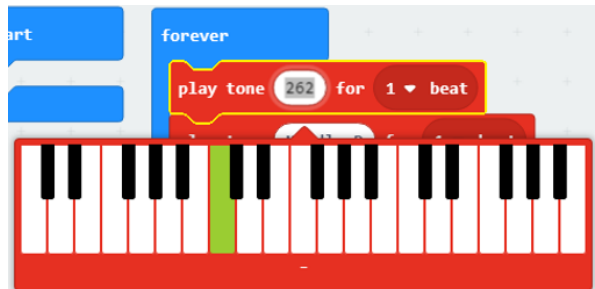
3. 스크립트를 마이크로 비트에 다운로드하고 부저를 통해 멜로디(노래)가 재생되는지 테스트합니다.

### 【Tips】

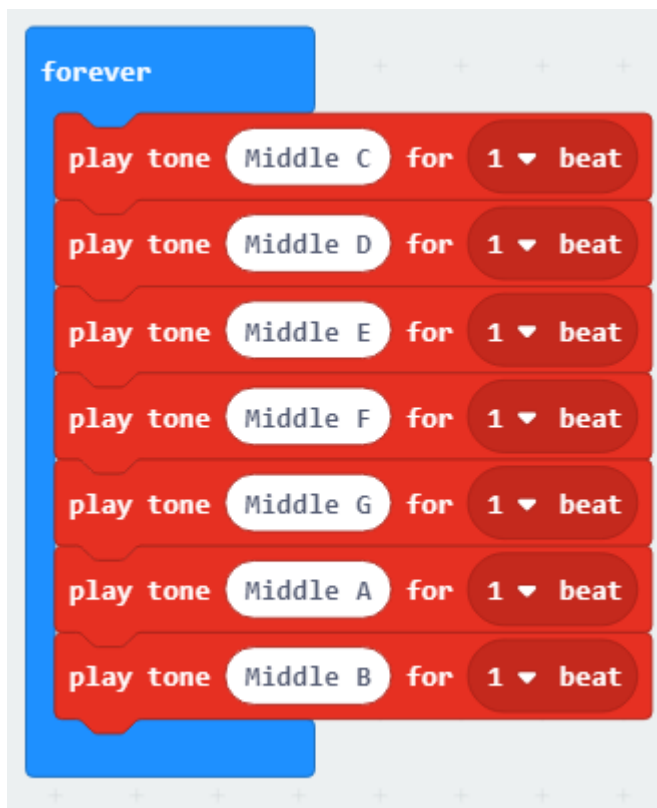
물체를 흔들면 소리가 납니다. 사람은 성대를 다른 주파수로 흔들어서 말을 합니다. 부저는 같은 방식으로 작동하며, 제어 보드에서 다른 주파수로 높거나 낮을 때, 다른 톤을 생성합니다. 다양한 소리를 내기 위해 톤을 변경할 수 있습니다.

## 피아노 치기

블록을 사용하여 피아노를 시뮬레이션 해보겠습니다.



1. **[ (음) (n박자) 연주 ]** 블록을 **[ 무한반복 ]** 블록에 배치합니다. 톤을 중간 C(Middle C)와 1비트로 설정합니다.
2. 아래와 같이 스크립트를 만들어 완성합니다.



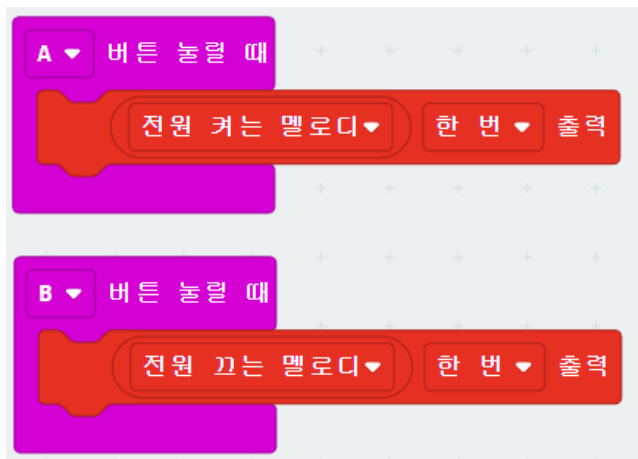
3. 마이크로 비트가 스크립트를 다운로드하면 연결된 스피커가 톤(음)을 반복합니다.

“도” , “레” , “미” , “파” , “솔” , “라” , “시” .

## 노래 재생 2

버튼을 누를 때마다 다른 멜로디가 재생되도록 새 프로젝트를 만듭니다.

1. [ ( ) 버튼 눌릴 때 ] 블록을 놓아두고, 멜로디를 반복하도록 설정합니다.
2. 아래 사진처럼 스크립트를 작성하여 완성합니다.

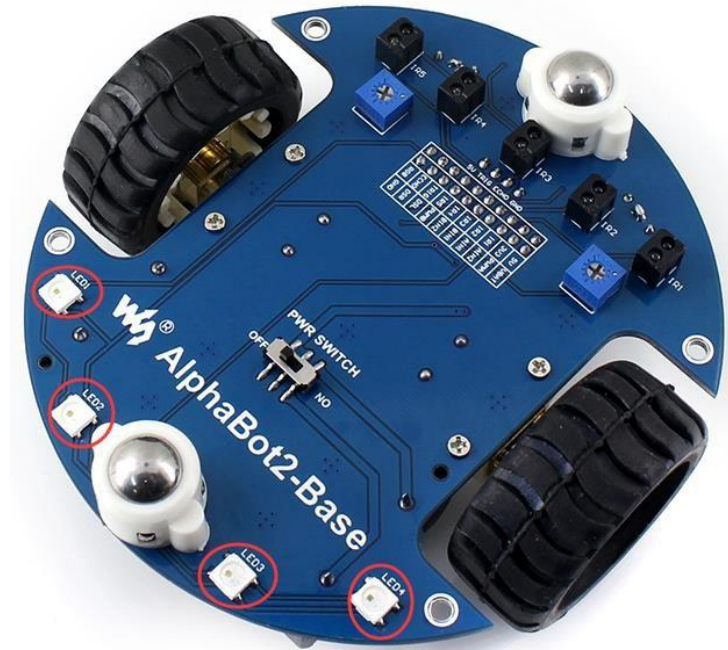


3. 마이크로 비트에 스크립트를 다운로드하고 버튼을 눌러, 맞는 멜로디가 나오는지 확인합니다.

## CHAPTER 5 RGB LED

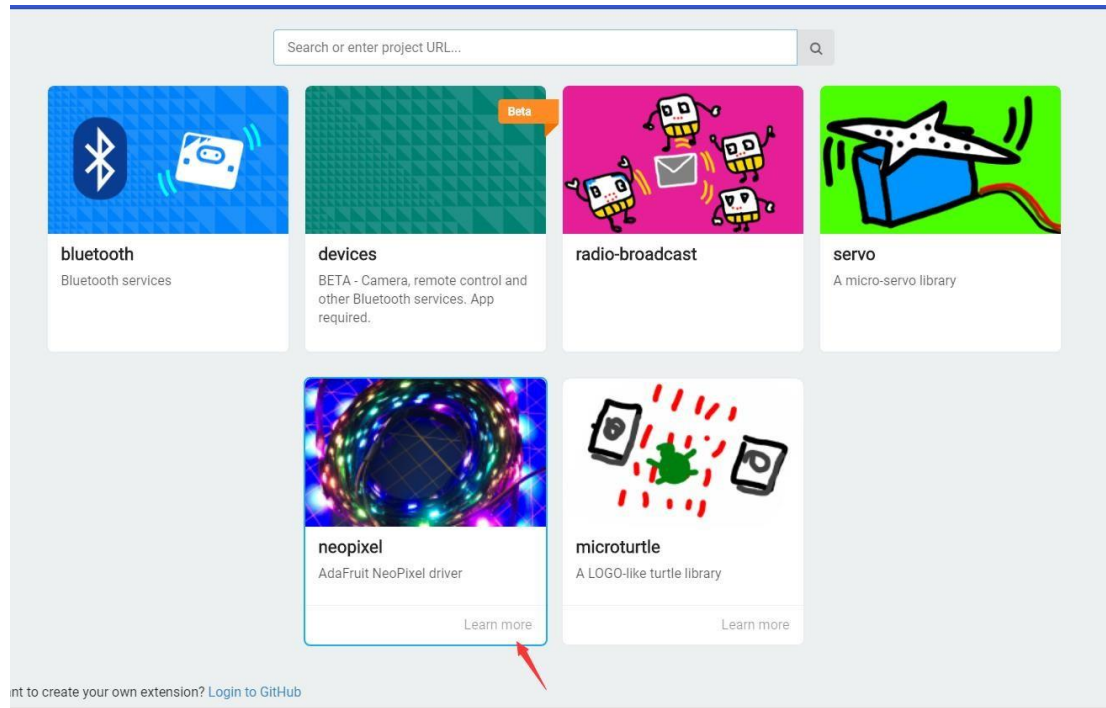
Alphabot2에는 화려한 효과를 내도록 프로그래밍할 수 있는 4개의 RGB LED가 있습니다. 이

장에서는 RGB LED를 이용해 코딩을 해보겠습니다.

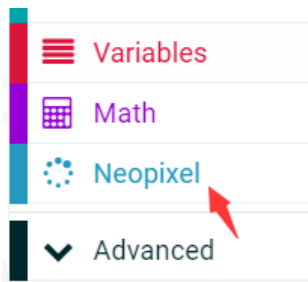


## 패키지 추가

코딩 하기 전에 패키지 추가(확장)를 선택한 다음 네오픽셀을 클릭하여, 네오픽셀 패키지를 추가해줍니다.



네오픽셀 패키지 추가 후, 블록 패키지 네오픽셀이 추가됩니다.



## LED 점등

새 프로젝트를 생성하고, 모든 RGB LED를 점등하는 코딩을 합니다.

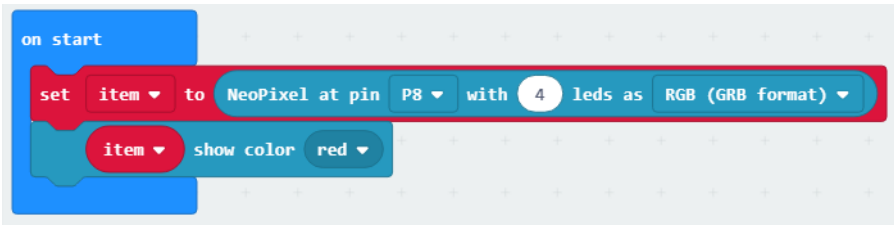
1. 네오픽셀 패키지의 [ item에 NeoPixel at pin P8 with 4 leds as RGB (GRB format) 저장 ]를 [ 시작하면 ] 블록에 배치합니다.

-변수인 item의 이름은 언제든지 바꿀 수 있습니다.

-핀 P8을 선택한 이유는 P8이 Alphabot2의 RGB LED의 기본 핀이기 때문입니다.

2. 네오픽셀 패키지의 [ item show color red ] 블록을 1번에 이어 [ 시작하면 ] 블록에 배치합니다.

-변수 item은 마지막인 [ item show color red ] 블록이 꼭 있어야 하지만, 그의 색은 변경되어도 문제 없습니다. Ex) red -> blue로 바뀌어도 상관 없습니다.

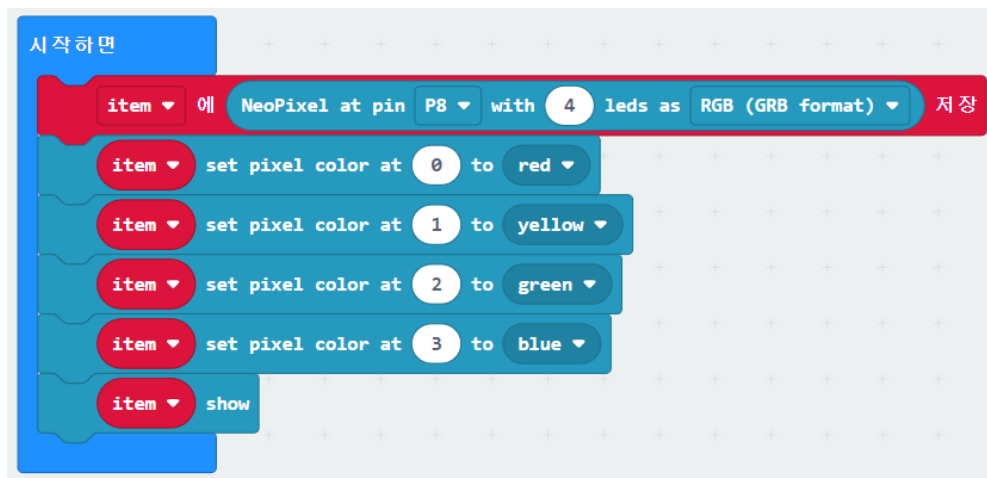


3. 스크립트를 마이크로 비트에 다운로드 하여, RGB LED가 켜지는 것을 확인합니다.

## 다양한 색상 표시

LED 점등의 마지막 블록을 수정하여 빨강, 노랑, 초록, 파랑 등 다양한 색상을 표시하도록 4개의 RGB LED를 만듭니다.

1. 왼쪽으로 드래그, DELETE키, 마우스 오른쪽 클릭 등을 이용해 [ item show color red ] 블록 부분을 수정하기 위해 삭제해줍니다.
2. 네오픽셀 패키지의 [ item set pixel color at 0 to red ] 블록을 배치해줍니다.  
 - 위 블록은 [Neopixel->더보기](#) 에서 찾을 수 있습니다.  
 - 이 블록은 모든 픽셀의 색상을 0부터 설정할 수 있습니다.
3. 세 개의 블록을 추가로 더해 총 4개의 LED에 들어갈 블록을 만들어 줍니다.
4. [ item show ] 블록을 마지막으로 [ 시작하면 ] 블록을 모두 채웁니다.  
 - 이 블록은 매우 중요합니다. 설정 색상을 RGB LED로 표시했지만, 마지막 [ item show ] 블록이 없다면, 이전에 있던 모든 과정은 모두 쓸모가 없어집니다.



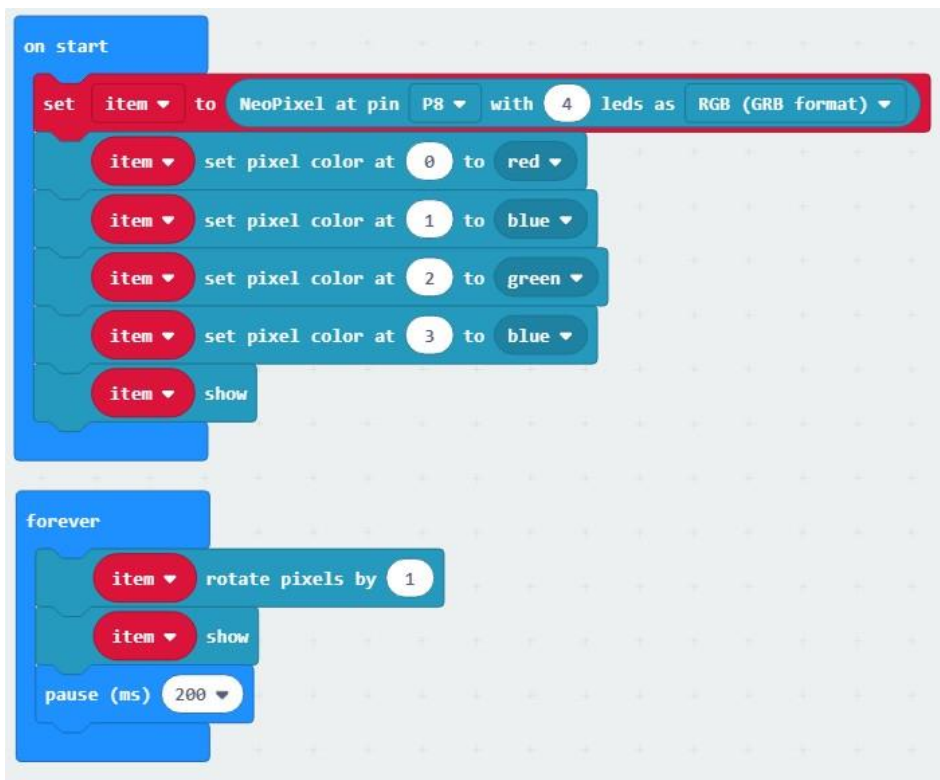
5. 스크립트를 마이크로 비트에 다운로드 하여 RGB LED의 변화를 확인합니다.



## 깜박이는 LED

RGB LED 디스플레이를 순환하도록 마지막 부분에 코딩을 추가해줍니다.

1. [ item rotate pixels by 1 ] 블록을 [ 무한반복 ] 블록에 배치합니다.  
 위 블록은 현재 픽셀의 색상을 다음 픽셀로 전송해줍니다.
2. [ item show ] 블록을 [ item rotate pixels by 1 ] 블록을 이어 [ 무한 반복 ] 블록에 배치합니다.  
 - [ item show ] 블록은 반드시 필요합니다.

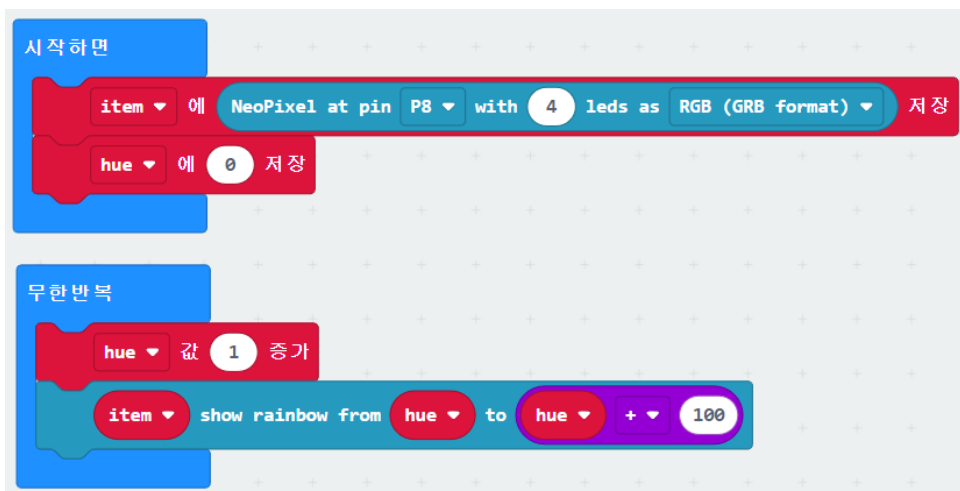


3. 마이크로 비트에 스크립트를 다운로드 하여, RGB LED의 동작을 확인합니다.

## 무지개 색 LED 점등

무지개 효과를 위해 4개의 RGB LED를 코딩 하는 새 프로젝트를 생성합니다.

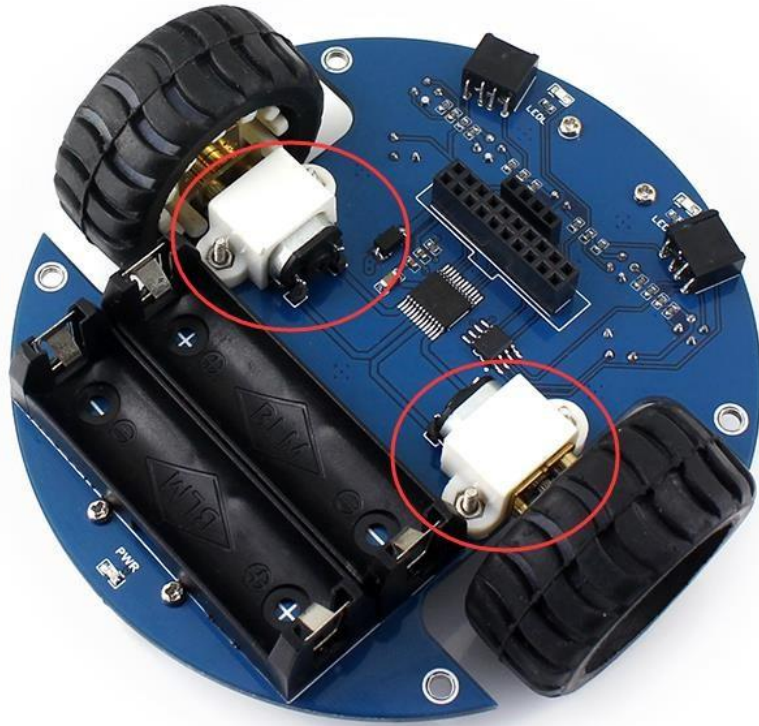
1. 네오픽셀 패키지의 [ item에 NeoPixel at pin P8 with 4 leds as RGB (GRB format) 저장 ]를 [ 시작하면 ] 블록에 배치합니다.
2. hue라는 변수를 만들고, 0으로 값을 초기화합니다.  
- 변수 패키지에서 변수를 생성 및 초기화가 가능합니다.
3. 변수 hue 값을 1씩 증가 시키는 [ hue 값 1 증가 ]를 [ 무한반복 ] 블록에 배치합니다.
4. [ item show rainbow from 1 to 360 ] 블록을 마지막으로 [ 무한반복 ] 블록에 배치합니다.  
- 변수 hue의 범위는 1~360입니다. 다른 hue 값은 다른 색상을 나타내며, 이 블록은 hue의 범위 1~360 중에서 4개의 RGB LED를 표시합니다. RGB LED가 많을수록, 무지개처럼 더 많은 색상이 표시됩니다.
5. [ 0 + 0 ] 블록과 hue 변수를 이용하여 코딩을 마무리 짓습니다.  
- [ 0 + 0 ] 블록은 계산 패키지에서 찾으실 수 있습니다.



6. 마이크로 비트에 스크립트를 다운로드 하고 RGB LED의 동작을 확인합니다.

## CHAPTER 6 로봇 동작시키기

지금까지 LED와 부저를 제어하는 방법을 배웠습니다. 이번 장에서는 Alphabot2를 제어하여 움직이게 해보겠습니다.

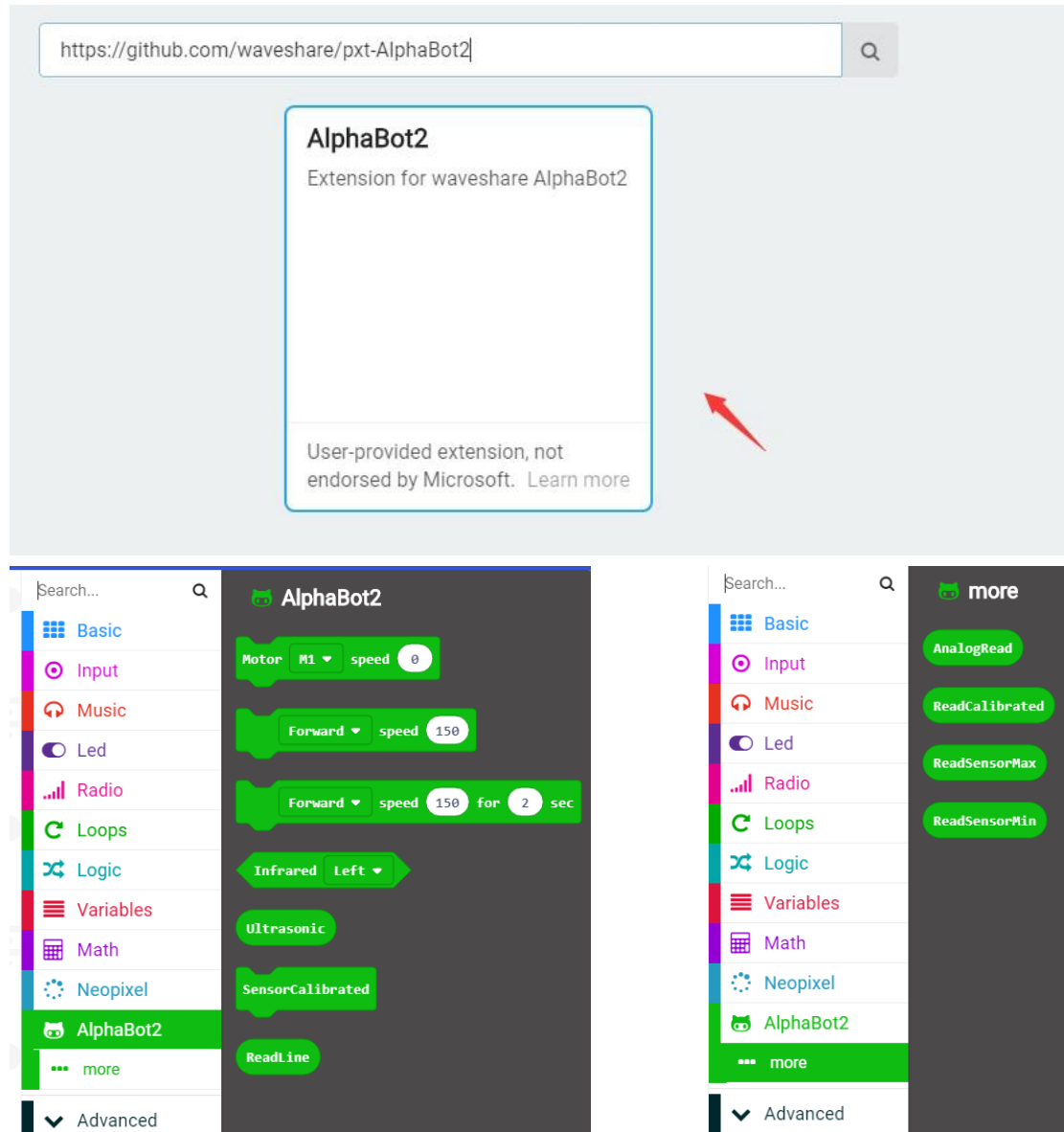


## 패키지 추가

패키지에서 확장에 들어가 검색 란에 아래 주소를 입력해 주어야 합니다.

<https://github.com/waveshare/pxt-AlphaBot2>

아래와 같은 사진이 나왔다면, AlphaBot2을 클릭해 주세요.



## 모터 동작

새 프로젝트를 생성합니다. 새 프로젝트에서는 Alphabot2의 모터를 버튼으로 제어할 것입니다.

1. [만약 ~이면] 블록과 [버튼 눌림] 블록을 [무한반복] 블록 안에 배치합니다.
2. 그 후, Alphabot2 패키지의 [Motor M1 speed 0] 블록을 [만약 ~이면] 블록 안에 배치합니다.
  - 이 블록은 하나의 모터를 제어하는 데 사용되며, 속도 범위는 -255 ~ 255이고, 숫자가 클수록 속도가 빨라집니다. 전진에는 양수, 후진에는 음수가 사용됩니다.
3. 아래와 같이 스크립트를 만들어 완성합니다.



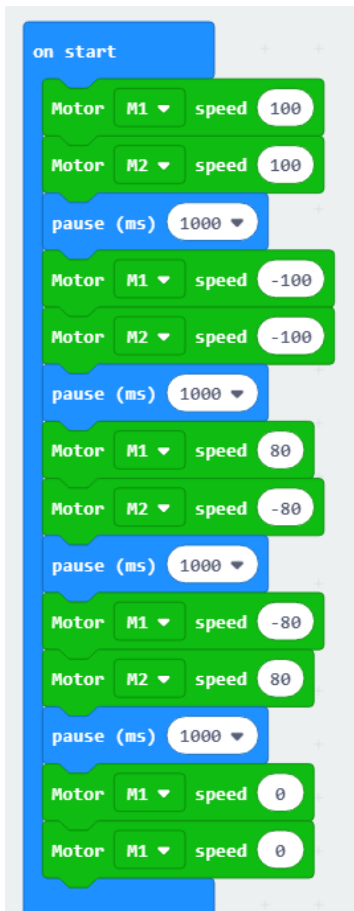
4. 마이크로 비트에 스크립트를 다운로드하고 모터의 동작이 아래 예상 결과처럼 되는지 확인합니다.
5. 예상 결과: A를 누르면 모터 M1이 움직이다가 손을 떼면 멈추고, B를 누르면 모터 M2가 움직이다가 손을 떼면 멈춥니다.

## 로봇 동작 1

새 프로젝트를 생성합니다. Alphabot2가 코딩을 통해 전진, 후진, 좌회전, 우회전을 하도록 합니다.

1. [ Motor M1 speed 0 ] 블록을 [ 시작하면 ] 블록 안에 배치합니다.
  - 모터 모두 앞으로 움직이면 로봇은 앞으로 이동할 것입니다.
  - 모터 모두 뒤로 움직이면, 로봇은 뒤로 이동할 것입니다.
  - 왼쪽 모터가 뒤로 가거나 멈추고, 오른쪽 모터가 앞으로 가면 로봇은 왼쪽으로 회전할 것입니다. (일반적으로 두 모터의 속도가 다르면 로봇이 선회합니다.)
  - 왼쪽 모터가 앞으로 가고, 오른쪽 모터가 뒤로 가거나 멈추면, 로봇은 오른쪽으로 회전할 것입니다. (일반적으로 두 모터의 속도가 다르면 로봇이 선회합니다.)
  - 모터 모두 멈추면, 로봇은 멈출 것입니다.
2. 딜레이를 주기 위해, [ 기본 ] 패키지에 있는 [ 일시 중지 ]를 이용합니다.

3. 아래와 같이 스크립트를 만들어 완성합니다.



4. 마이크로 비트에 스크립트를 다운로드하고, 모터의 동작을 확인합니다.

## 로봇 동작 2

로봇 동작 1에서 사용하지 않은 모터를 제어하는데 사용할 수 있는 다른 블록도 있습니다.

1. [ Forward speed 0 ] 블록을 [ 시작하면 ] 블록 안에 배치합니다.

- 이 블록은 로봇이 모든 방향으로 움직이도록 직접 제어할 수 있습니다.

2. [ 일시 중지 ] 블록 또한, [ 시작하면 ] 블록에 배치했습니다.

3. 아래와 같이 스크립트를 만들어 완성합니다.



4. 마이크로 비트에 스크립트를 다운로드하고, 모터의 동작을 확인합니다.



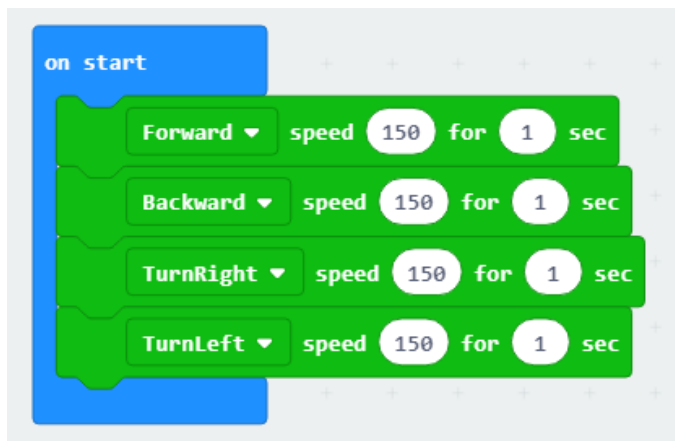
## 로봇 동작 3

로봇이 일정 시간 동안 움직이도록 제어할 수 있는 또 다른 블록이 있습니다.

1. [ Forward speed 150 for 2 sec ] 블록을 [ 시작하면 ] 블록에 배치합니다.

- 이 블록을 사용하면 로봇이 2초 동안 속도 100으로 앞으로 이동한 후 다른 설정이 없으면 정지합니다.

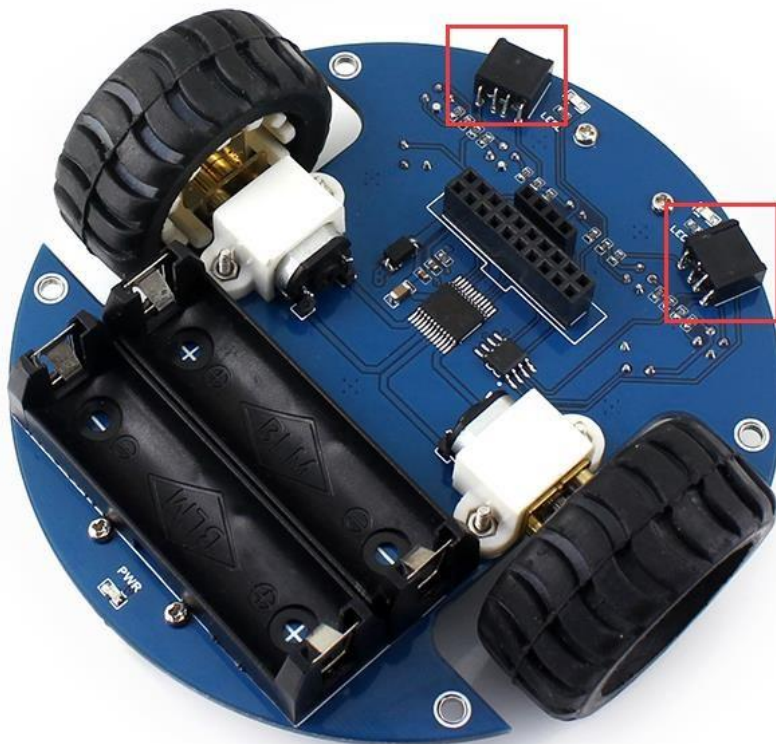
2. 아래와 같이 스크립트를 만들어 완성합니다.



3. 마이크로 비트에 스크립트를 다운로드 하고 아래 예상 결과대로 나오는지 확인합니다.
4. 다운로드 후, 로봇이 1초간 전진, 1초간 후진, 우회전, 좌회전, 마지막으로 정지합니다.

## CHAPTER 7 적외선 장애물 회피

두 개의 적외선 센서가 Alphabot2-Base 보드 전면에 통합되어 있으며, Alphabot2를 조립한 경우, 초음파 센서 아래에 있습니다.



모든 적외선 센서에는 하나의 발신기와 하나의 수신기가 있으며, 발신기에서 적외선을 보내면 전방에 장애물이 있으면 그 중 일부가 반사되어 수신기에서 수신됩니다. 적외선 센서 옆에는 장애물을 나타내는 녹색 LED가 있습니다.



## 적외선 센서 감지

1. [ 만약 ~이면 ] 블록을 [ 무한 반복 ] 블록에 배치합니다.
2. [ 만약 ~이면 ] 블록에서 불(참/거짓)인 [ 그리고 ] 블록을 배치합니다.
3. Alphabet2 패키지에 포함된 [ Infrared ] 블록을 [ 그리고 ] 블록의 양 옆으로 배치합니다.
4. [ 만약 ~이면 ] 블록의 동작 부분에 [ 문자열 출력 ] 블록을 배치합니다.
5. 아래와 같이 스크립트를 만들어 완성합니다.

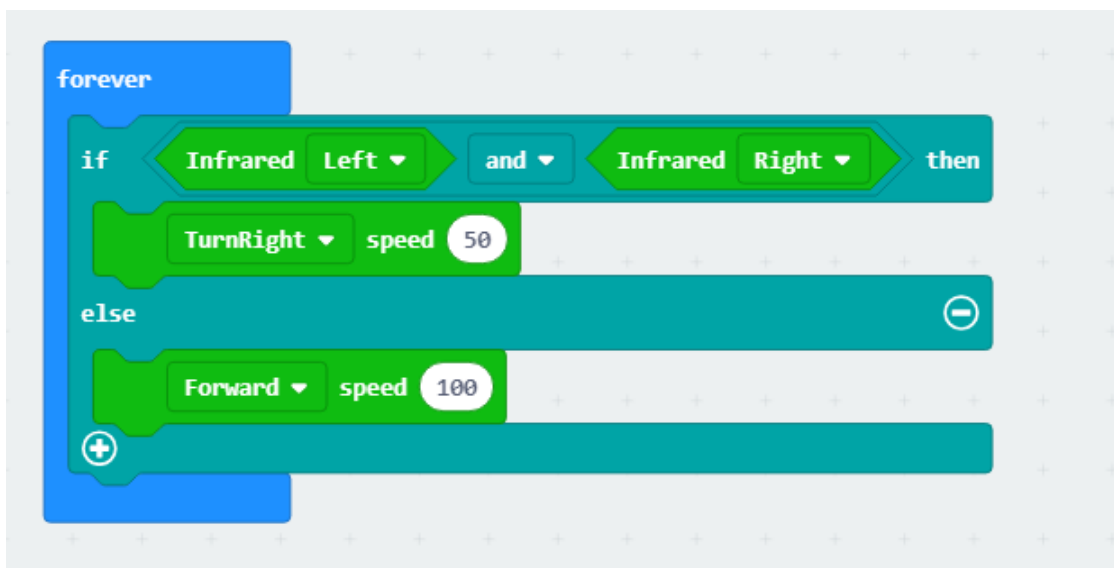


6. 마이크로 비트에 스크립트를 다운로드하고, 아래 예상 결과에 맞게 결과가 나오는지 확인합니다.
7. 다운로드 후, 양쪽 적외선 센서에서 장애물이 감지되면 마이크로 비트의 LED에 “A” 문자열이, 왼쪽 적외선 센서에서 장애물이 감지되면 마이크로 비트의 LED에 “L” 문자열이, 오른쪽 적외선 센서에서 장애물이 감지되면 마이크로 비트의 LED에 “R” 문자열이 표시됩니다.

## 장애물 회피

마지막 스크립트를 수정하여, 장애물을 피하는 로봇을 만듭니다.

1. 장애물이 감지되면 로봇이 우회전하도록 [ 문자열 출력 ] 블록과 [ (방향) speed ] 블록을 함께 [ 만약 ~이면 ] 블록의 동작 부분에 배치합니다.
2. [ 일시 정지 ] 블록 또한, 같이 배치합니다.
3. 로봇이 앞으로 이동하도록 전진 속도 블록을 다른 블록 위에 놓습니다.



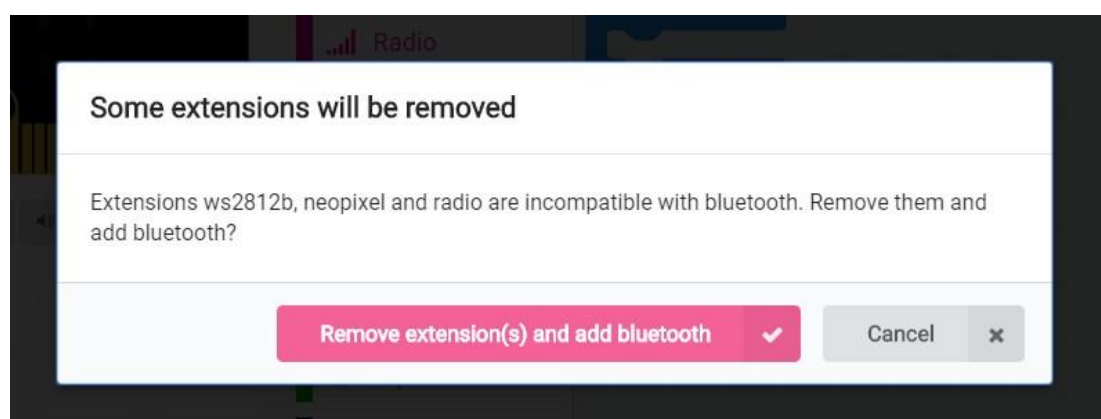
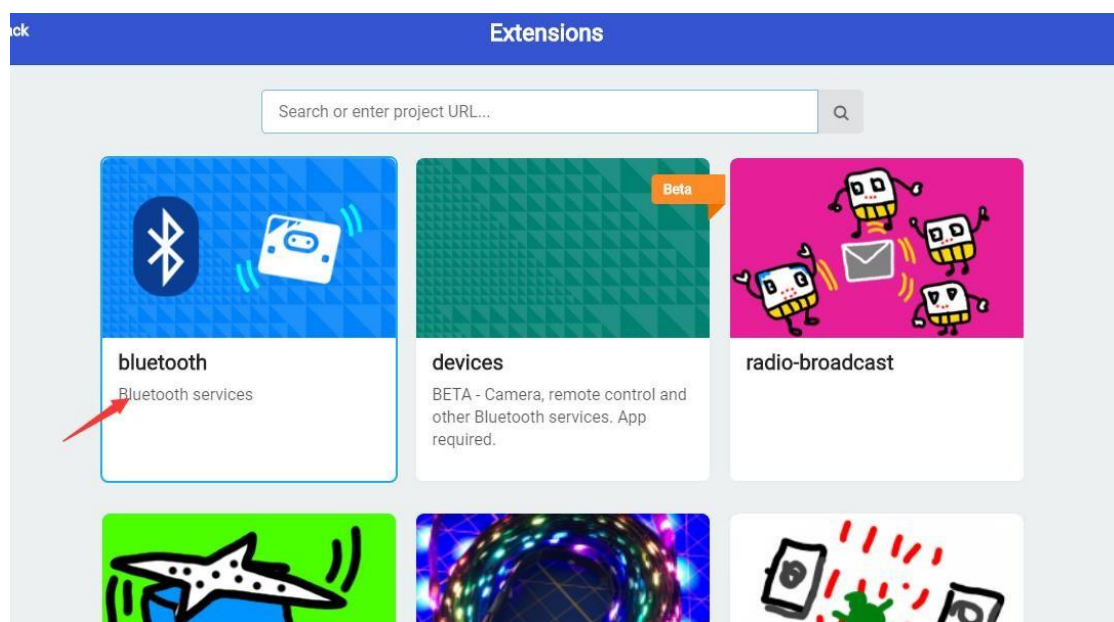
4. 마이크로 비트에 스크립트를 다운로드하고 동작을 확인합니다.

## CHAPTER 8 블루투스

Nrf51822는 마이크로 비트가 블루투스 신호를 송수신 할 수 있도록 하는 BLE(저에너지) 모듈입니다. 이 기능을 사용하면 블루투스를 통해 휴대폰으로 로봇을 제어할 수 있습니다.

### 패키지 추가

블루투스를 사용하려면, 먼저 지원 패키지를 추가해야 합니다. 블루투스 패키지는 다른 패키지인 ws2812b, neopixel, radio와 호환되지 않습니다..

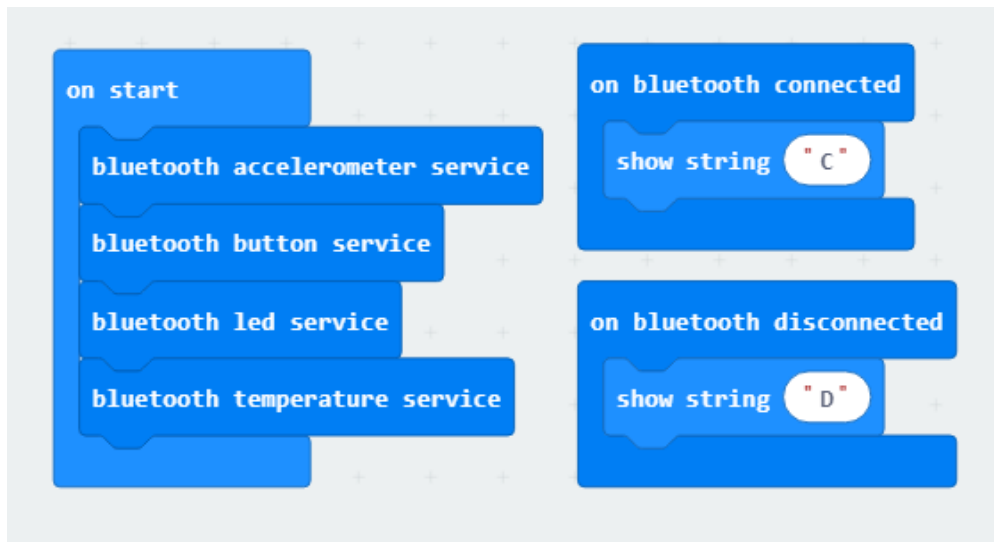


## 블루투스 연결 1

프로그래밍하기 전에 휴대폰에 블루투스 앱을 설치해야 합니다. ( 이 앱은 안드로이드만 지원됩니다. )

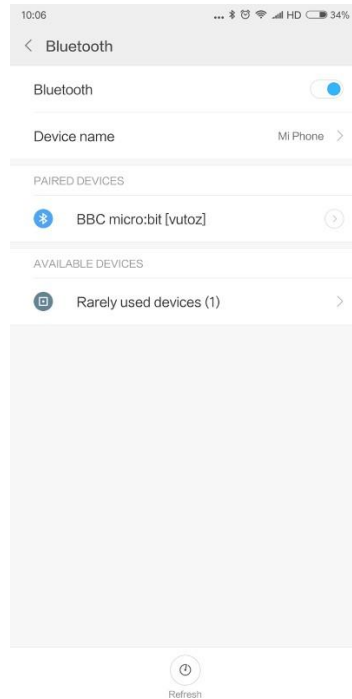
- [Bluetooth App for Android](#)

1. 시작 시 [ 블루투스 ( ) 서비스 시작 ] 블록을 [ 시작하면 ] 블록 안에 배치합니다.
  - 사용할 수 있는 서비스는 다음과 같습니다.
2. [ 블루투스 연결되면 실행 ] 블록안에 [ 문자열 출력 "C" ] 블록을 배치합니다.
  - 블루투스가 연결된 경우, 마이크로 비트 LED가 "C"로 표시되도록 합니다.
3. [ 블루투스 해제되면 실행 ] 블록안에 [ 문자열 출력 "D" ] 블록을 배치합니다.
  - 블루투스 연결이 해제된 경우, 마이크로 비트 LED가 "D"로 표시되도록 합니다.

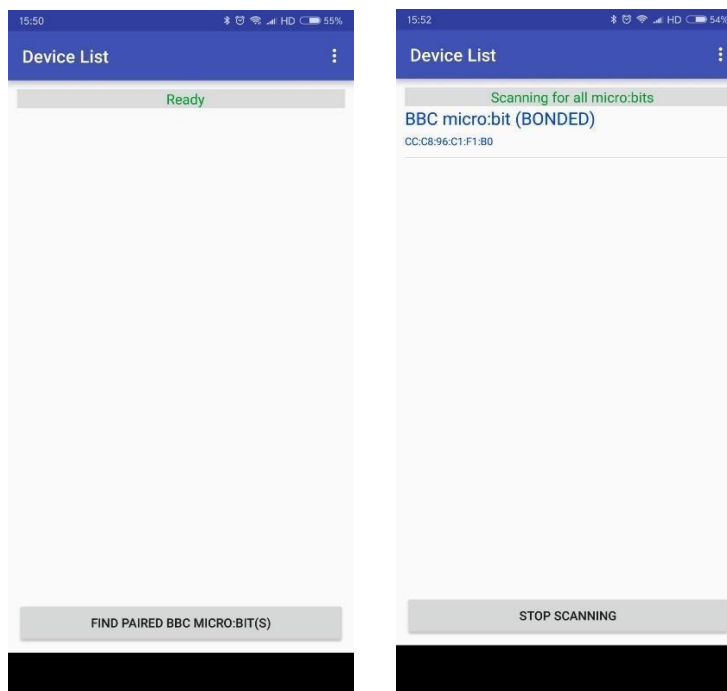


4. 마이크로 비트에 스크립트를 다운로드하고, 동작을 확인합니다.
5. 다운로드 후, 마이크로 비트의 A 버튼과 B 버튼을 동시에 누르고 리셋 버튼을 누르면 LED 매트릭스에 블루투스 아이콘이 표시된 후 버튼을 놓아야 합니다. 이 단계는 마이크로 비트를 페어링 모드로 설정하는 단계입니다.

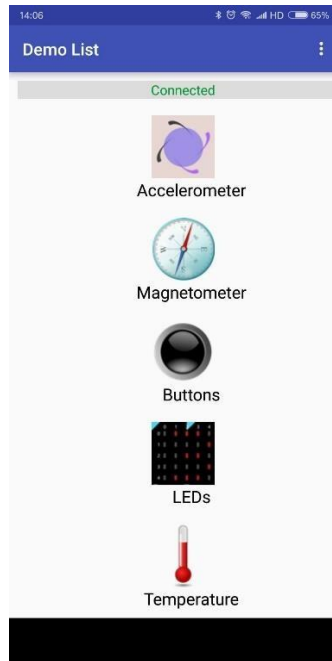
- 휴대폰의 블루투스 서비스를 열고 BBC micro:bit 장치를 검색하여 찾아 페어링하여 연결합니다. 페어링이 완료되면, “√” 아이콘이 표시됩니다. 새로 페어링하기 전에 페어링된 마이크로 비트 장치를 삭제해야 한다는 점에 유의하세요..



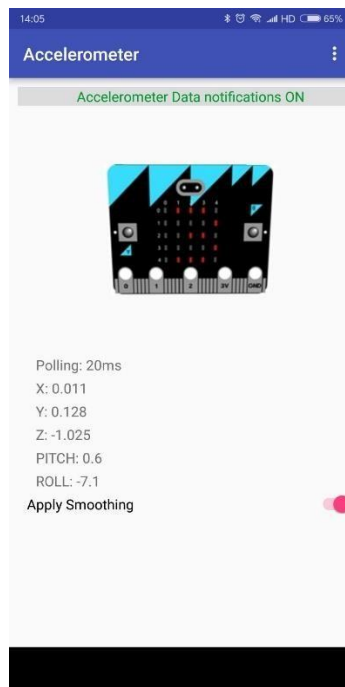
- 블루투스 앱을 열고, “페어링된 BBC micro:bit 찾기”를 클릭합니다.



8. BBC micro:bit (BOUNDED)를 클릭하여 연결합니다. 접미사가 (BOUNDED)인 장치만 연결할 수 있습니다.
9. 연결 후, 제어 옵션은 아래와 같이 나열되며, LED 매트릭스에는 문자열 “C”가 표시됩니다.

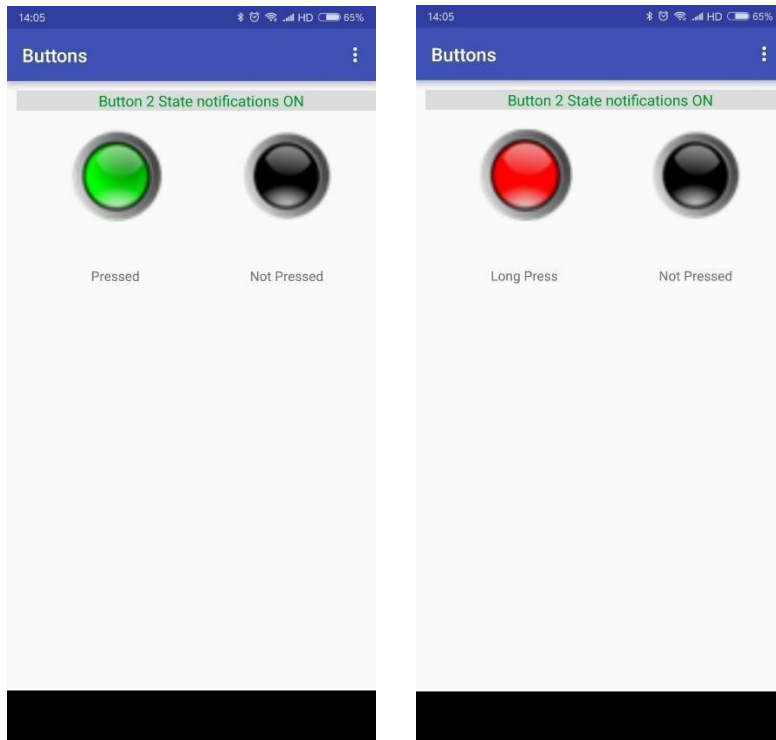


10. 그런 다음 블루투스로 마이크로 비트를 제어할 수 있습니다. 가속도계를 클릭하고, 마이크로 비트를 기울이면 앱의 마이크로 비트 이미지도 기울어집니다.

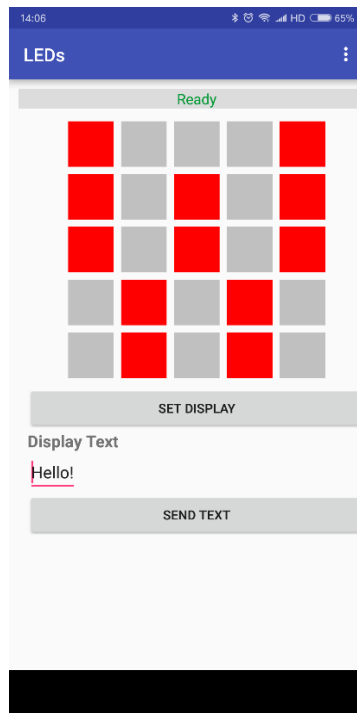




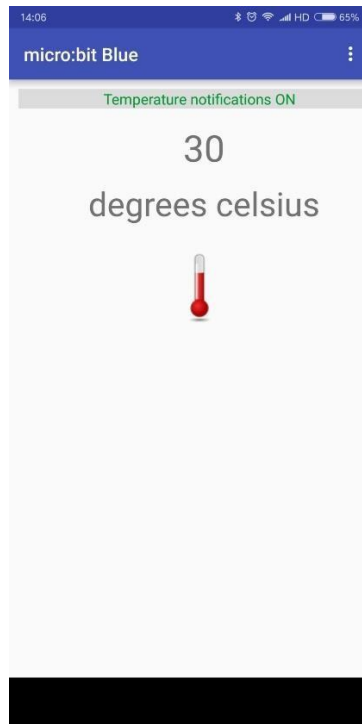
11. 버튼 옵션을 클릭합니다. 마이크로 비트의 버튼을 누르면 앱의 버튼 아이콘이 버튼 이벤트에 반응합니다.



12. LED 옵션을 클릭합니다. LED 매트릭스를 편집한 다음 디스플레이 설정을 변경하거나 문자를 보냅니다.



13. 온도 옵션을 클릭합니다. 현재 온도가 앱에 표시됩니다.



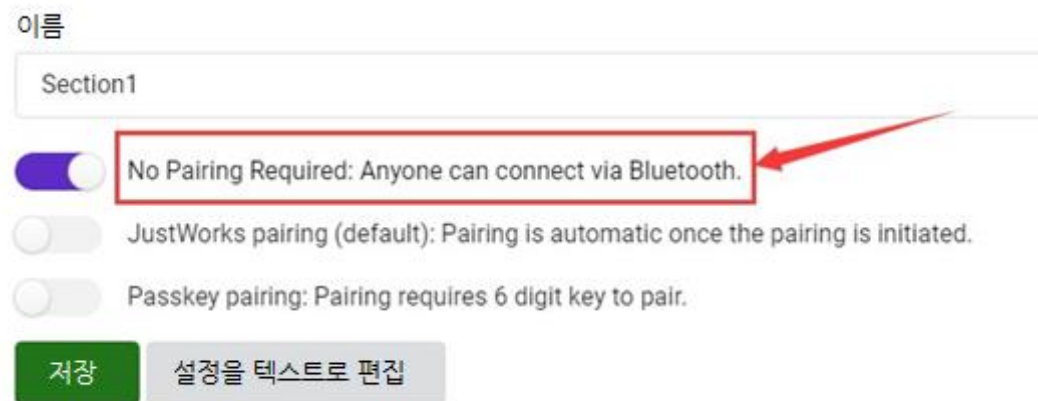
**【참고】** 코드로는 4개의 서비스만 시작하기 때문에, 앱에 표시된 서비스 중 일부는 사용할 수 없습니다.

## 블루투스 연결 2

마이크로 비트와 연결하려면, 먼저 페어링한 다음 연결해야 하는데, 두 단계가 많이 필요합니다. 여기서는 마지막 스크립트를 수정하여 더 간단하게 만들겠습니다.

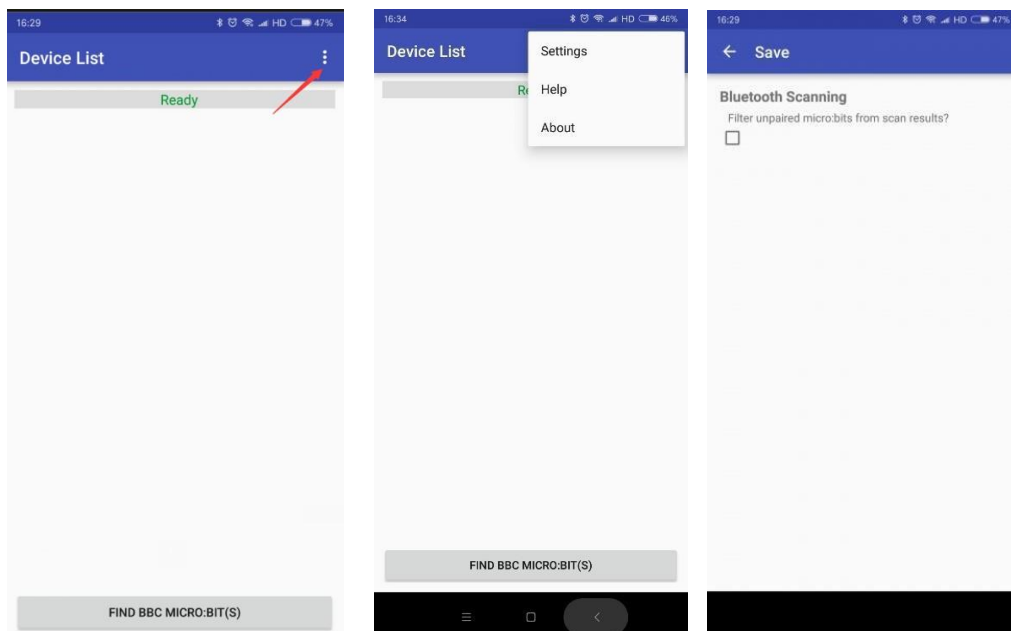
- 오른쪽 상단의 아이콘을 클릭합니다 : 기타... -> 프로젝트 설정에서 첫 번째 옵션을 선택합니다.

“페어링 필요 없음: 누구나 블루투스로 연결 가능”



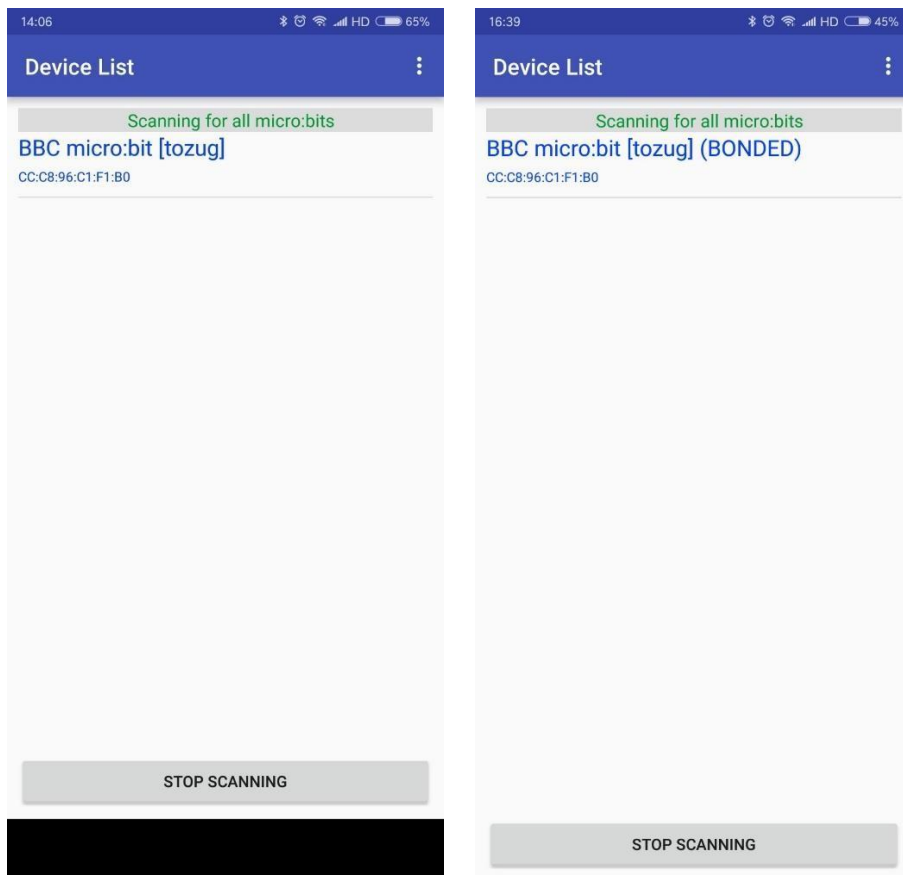
- 마이크로 비트에 스크립트를 다운로드 받습니다.

- 앱을 열고 설정을 변경한 후, 오른쪽 상단의 점 3개 아이콘을 누르고 설정을 선택한 다음 ‘스캔 결과에서 페어링되지 않은 마이크로 비트 필터링’ 옵션을 선택 취소합니다.



4. 설정 후, 장치 스캔을 다시 진행합니다. BBC micro:bit [tozug]<sup>4</sup>가 나열되면 micro:bit와 연결하도록 할 수 있습니다. 페어링된 장

\* 페어링된 장치 (BOUNDED)가 붙은 장치를 제거해야 합니다.

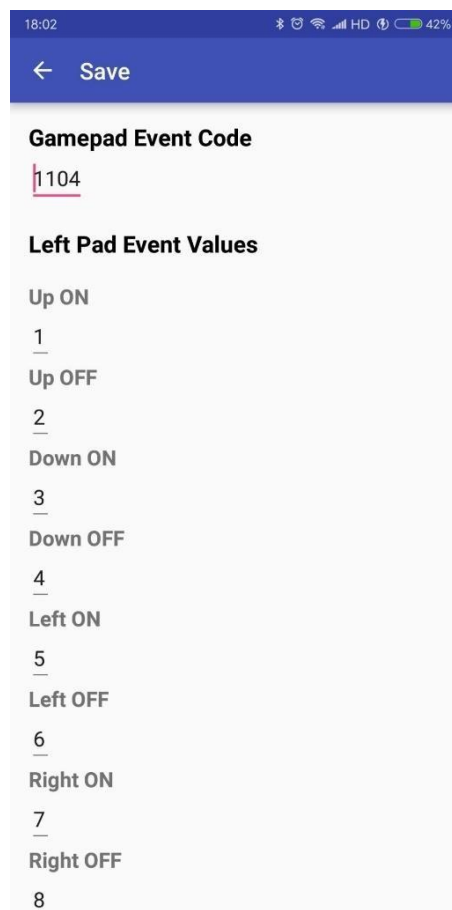
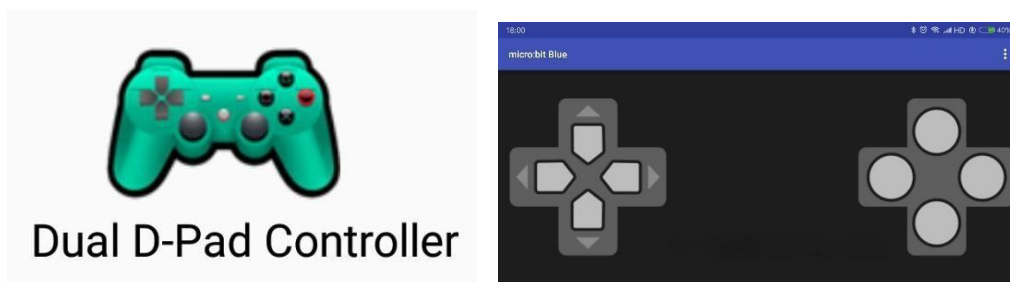


---

<sup>4</sup> [tozug] : 이 정보(문구)는 마이크로 비트마다 다릅니다.

## D-패드 컨트롤러 1

앱에서 Dual D-Pad Controller라는 옵션을 찾을 수 있을 것입니다. 이 옵션을 클릭하면 듀얼 D-Pad(8개의 버튼)가 있는 제어 페이지로 들어갑니다. 우측 상단에 있는 점 3개 아이콘을 클릭하면, 설정을 변경할 수 있습니다. 가상 버튼을 누를 때마다 관련 신호가 블루투스를 통해 마이크로 비트에 전송됩니다. 게임패드는 마이크로 비트에서 이러한 신호에 의해 트리거 됩니다..



이제 스크립트를 작성해 보겠습니다.

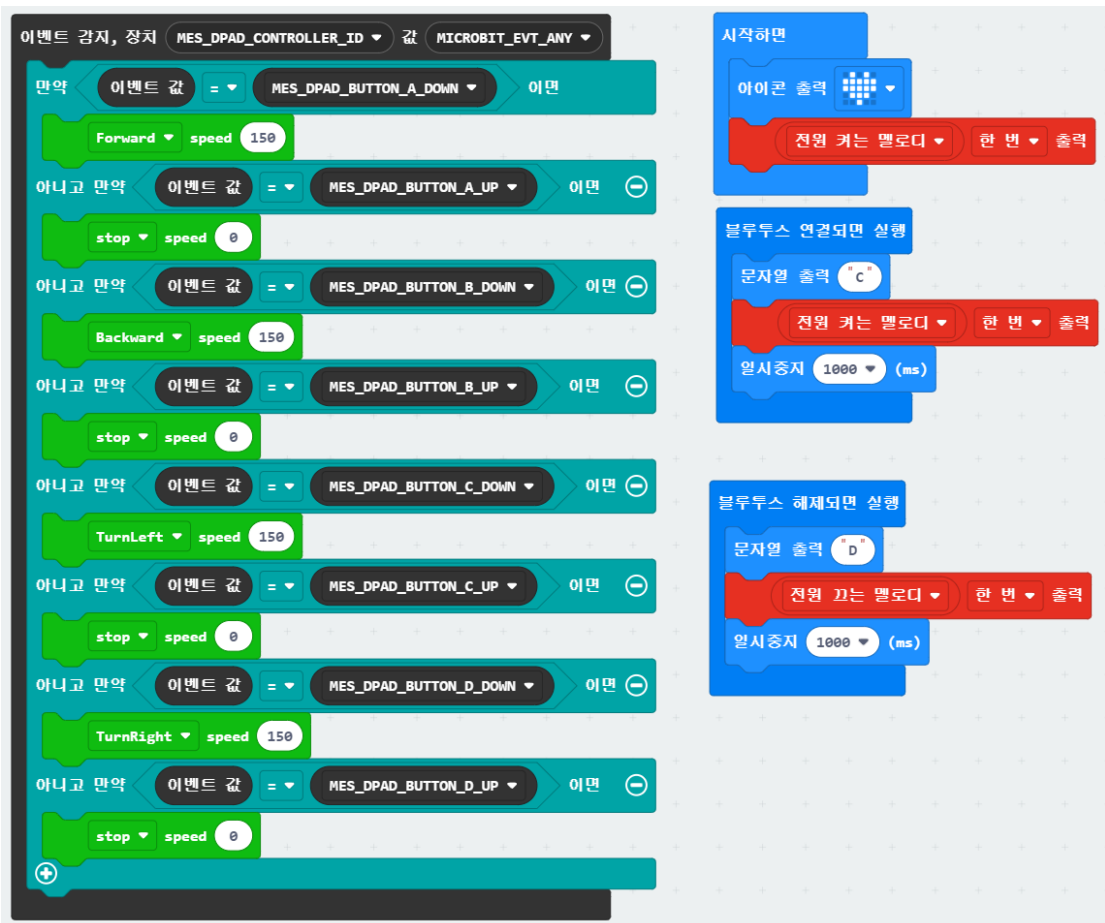
1. 전체 스크립트 The whole script is as blow:



2. [ 이벤트 감지, 장치 ( ) 값 ( ) ] 블록은 고급제어 패키지에서 찾을 수 있습니다.
3. [ MES\_DPAD\_CONTROLLER\_ID ] 블록은 앱에서 받아오는 신호입니다.
4. [ MICROBIT\_EVT\_ANT ] 블록은 이벤트의 값입니다.
5. [ MES\_DPAD\_CONTROLLER\_ID ] 블록과 [ MICROBIT\_EVT\_ANT ] 블록은 고급제어 패키지에서 더보기란에 들어가면 찾을 수 있습니다.
6. 스크립트를 마이크로 비트로 다운로드 하여 결과를 확인합니다.

## D-패드 컨트롤러 2

D-패드 컨트롤러 1에서는 4개의 버튼을 사용했습니다. 이번 D-패드 컨트롤러 2에서는 앱에 있는 Dual D-패드 컨트롤러의 8개의 버튼을 모두 사용하겠습니다. D-패드 컨트롤러 1에서 사용하지 않던 4개의 버튼을 사용할 때, 각기 다른 아이콘을 표시되도록 하겠습니다.

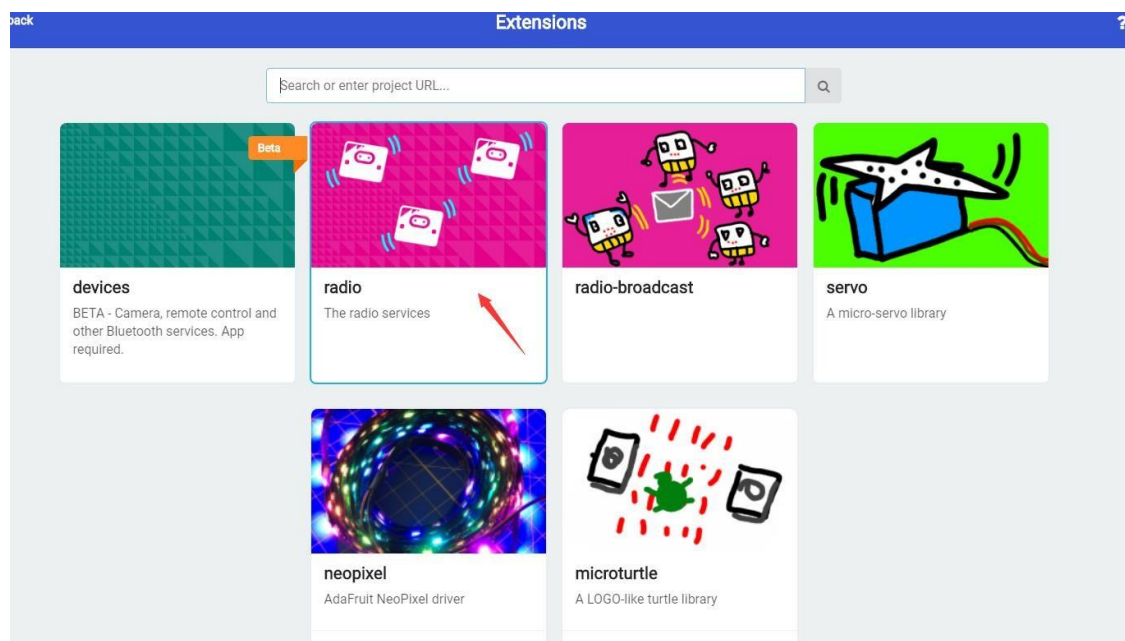


## CHAPTER 9 2.4GHz 라디오(무선 통신)

2.4GHz 라디오는 무선 통신으로, 인터콘과 같은 원격 통신에 널리 사용됩니다. 프로젝트를 수행하려면 최소 2개의 마이크로 비트와 1개의 조이스틱이 필요합니다. [Joystick for micro:bit](#).

### 패키지 추가

제일 먼저 라디오 패키지를 추가해 줍니다.



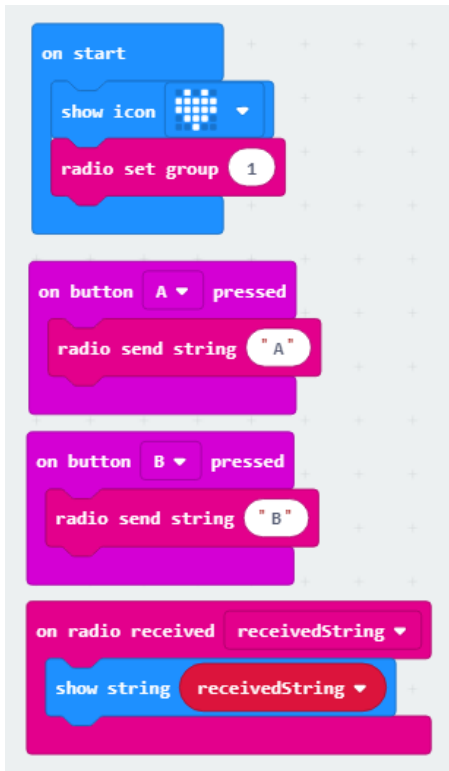
### 무선 통신

유의할 점.

1. 블록 라디오 그룹 1로 설정 해줌으로써 두 개의 마이크로 비트를 하나의 그룹으로 설정해야 합니다.
2. [ 버튼 A 눌릴 때 ] 블록 안에 [ 라디오 전송:문자열 "A" ] 블록을 배치해줌으로, 버튼 A를 누르면 마이크로 비트가 문자열 A를 전송하도록 설정합니다. 버튼 B 또한 동일한 설정입니다.



3. [ 라디오 문자열 수신시 receivedString ] 블록은 문자열을 수신할 사용자 라디오 수신 블록입니다.



4. 마이크로 비트와 테스트 모두에 스크립트를 다운로드 한 뒤, 아래 예상 결과를 따라서 결과를 확인합니다.
5. 예상 결과는 한 마이크로 비트의 버튼을 누르면 다른 마이크로 비트에 버튼에 해당하는 문자열이 표시되는 것입니다. Ex) 1번 마이크로 비트의 버튼 A를 누를 시 2번 마이크로 비트의 LED에 A가 표시된다.

## 마이크로 비트용 조이스틱 1

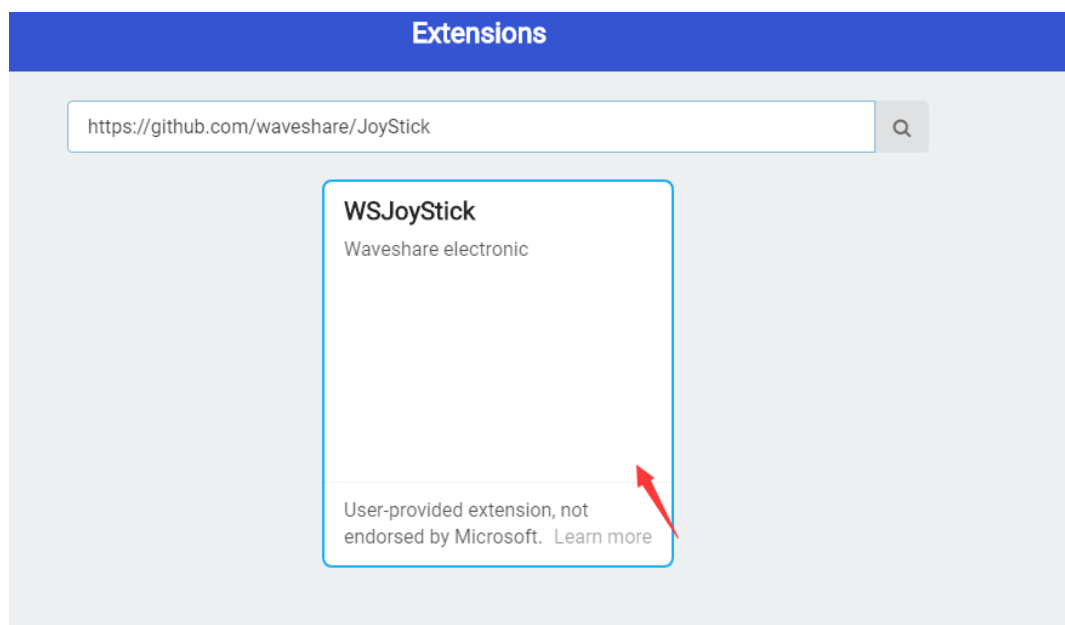
마이크로 비트용 조이스틱은 조이스틱 1개와 버튼 4개가 있는 작은 모듈입니다. 이 마이크로 비트용 조이스틱을 라디오를 통해 로봇을 제어하는 데 사용합니다.



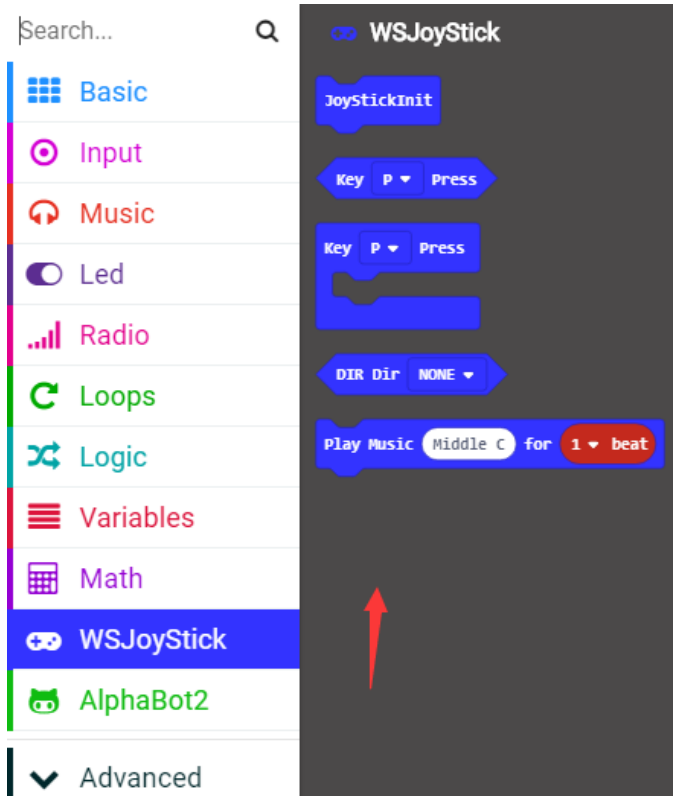
### 1. 패키지 추가

마이크로 비트용 조이스틱 패키지를 제공하며, 다음 주소로 추가할 수 있습니다. :

<https://github.com/waveshare/JoyStick>

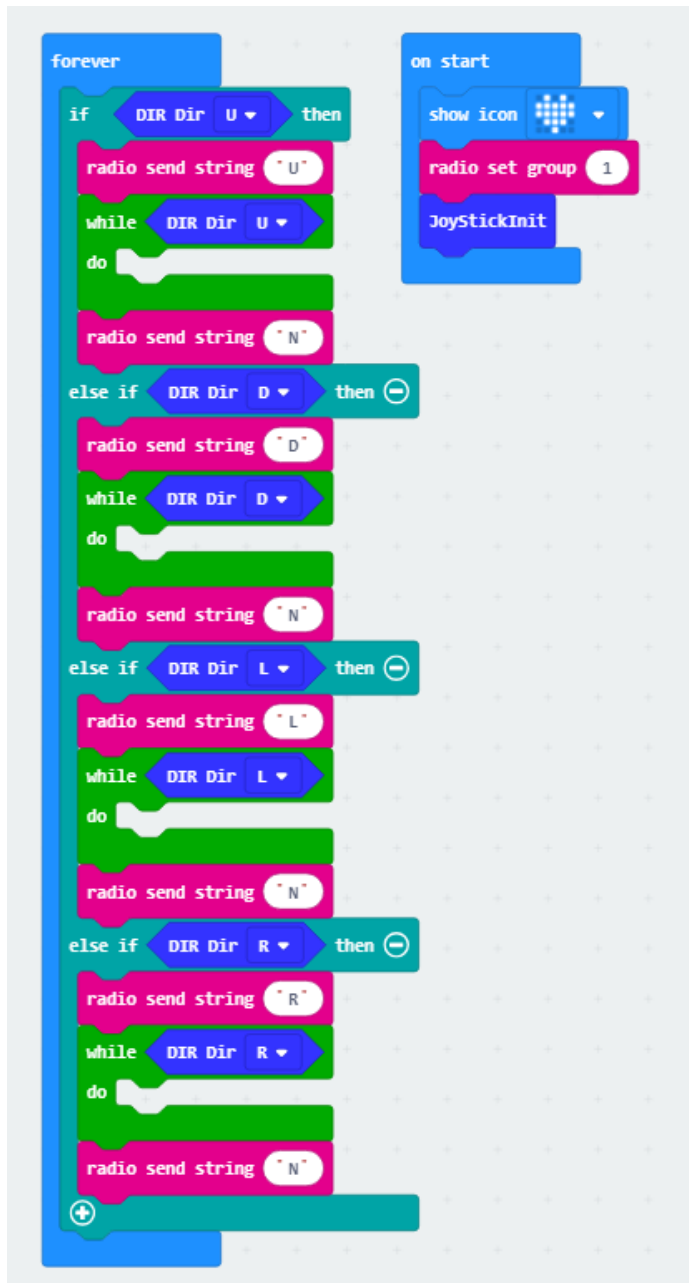


조이스틱 패키지에는 다섯 개의 블록이 있습니다. 조이스틱 모듈을 초기화하려면 시작 시 조이스틱 초기화 블록인 `[ JoyStickInit ]` 블록을 사용해야 합니다.



## 2. 발신자용 스크립트 입니다.

- 이 스크립트를 하나의 마이크로 비트에 발신자로 다운로드 하여, 조이스틱 모듈에 삽입합니다.



## 3. 수신기용 스크립트 입니다.

-수신기와 발신자를 같은 그룹에 설정해야 합니다.

-조건문에 들어간 "U" , " D" , " L" 과 같은 문자열은 고급 -> 문자열에서 찾으실 수 있습니다.

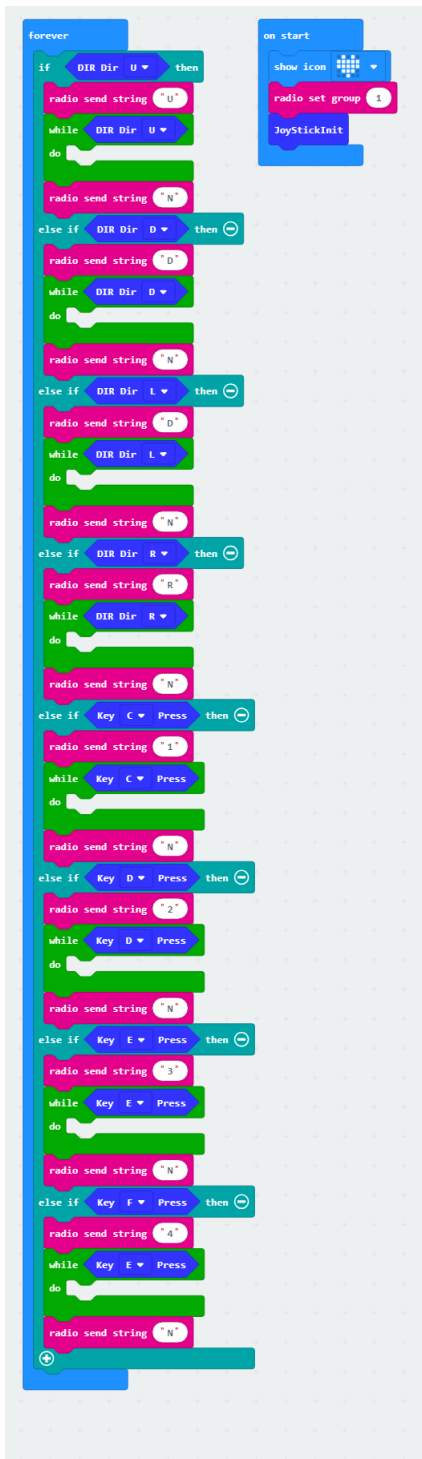
-이 스크립트를 다른 마이크로 비트에 리시버로 다운로드 하여 Alphabot2에 삽입합니다.



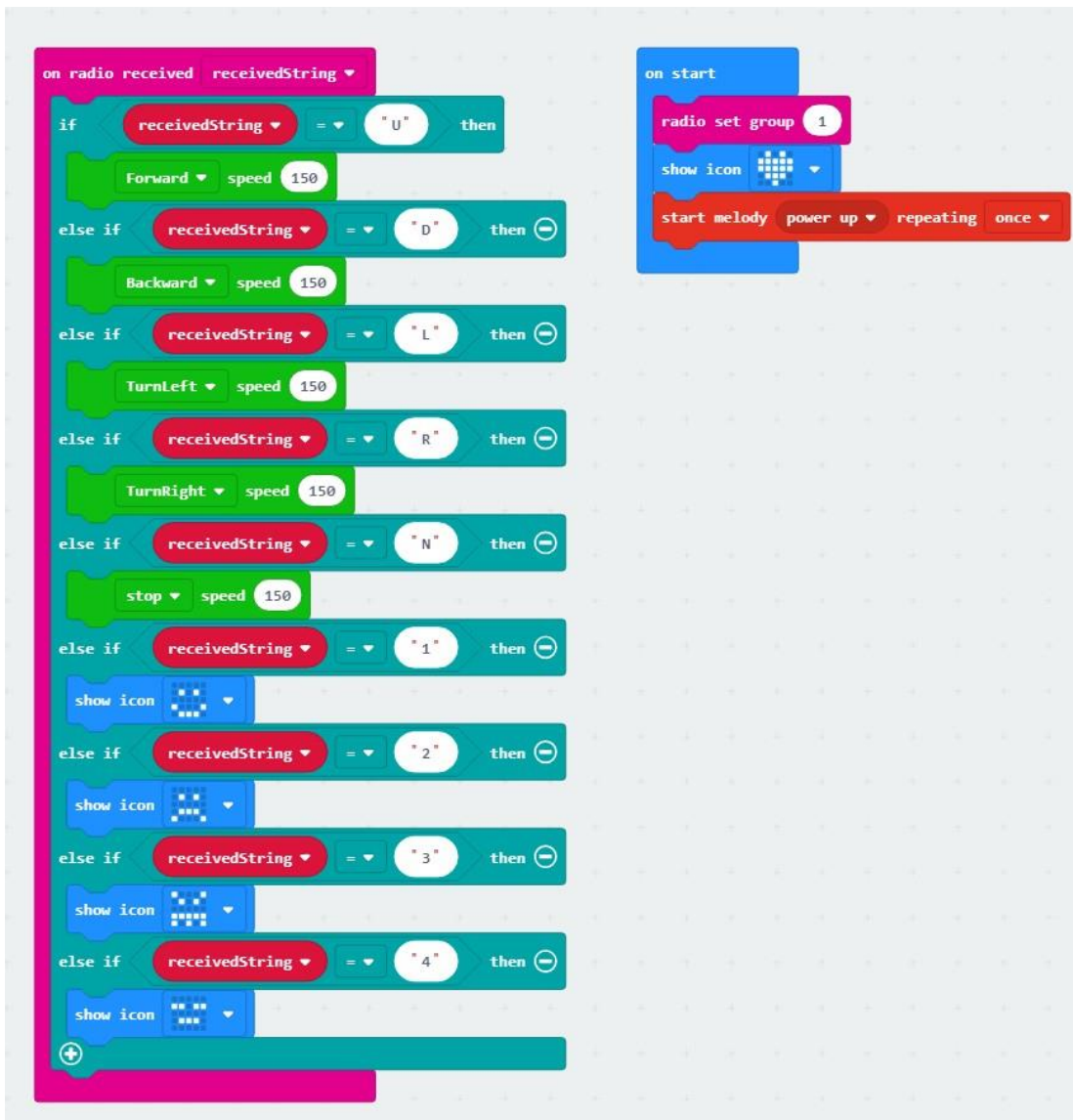
## 마이크로 비트용 조이스틱 2

스크립트를 수정해 조이스틱 모듈의 네 가지 색상 버튼을 누를 때, LED 매트릭스에 이모티콘이 표시되도록 합니다.

### 1. 발신자용 스크립트입니다.



2. 수신자용 스크립트입니다.



```

on radio received receivedString
  if receivedString = "U" then
    Forward speed 150
  else if receivedString = "D" then
    Backward speed 150
  else if receivedString = "L" then
    TurnLeft speed 150
  else if receivedString = "R" then
    TurnRight speed 150
  else if receivedString = "N" then
    stop speed 150
  else if receivedString = "1" then
    show icon [grid icon]
  else if receivedString = "2" then
    show icon [grid icon]
  else if receivedString = "3" then
    show icon [grid icon]
  else if receivedString = "4" then
    show icon [grid icon]
  +

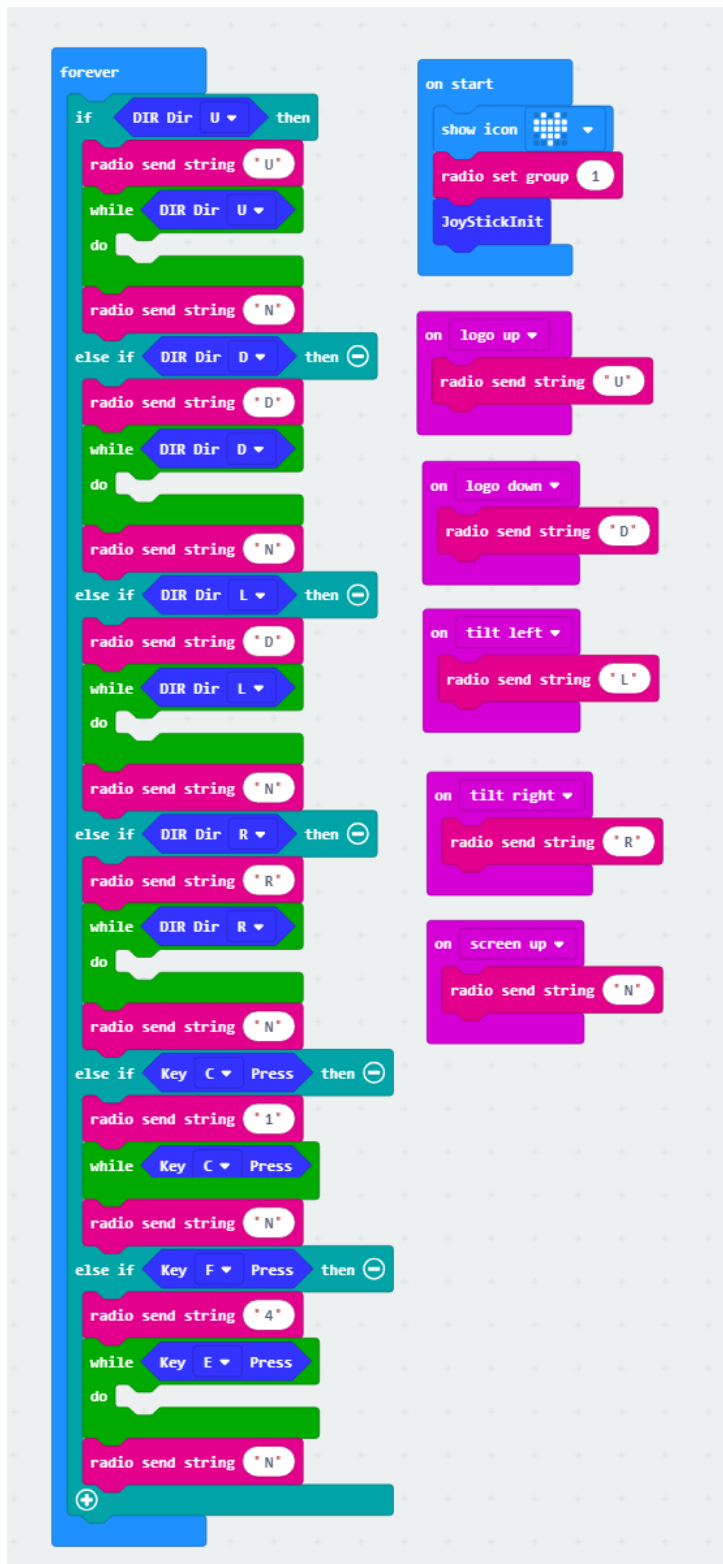
on start
  radio set group 1
  show icon [grid icon]
  start melody power up repeating once
  
```

## 가속도계를 이용해 Micro:bit의 방향으로 제어

3장에서 마이크로 비트의 상태를 감지할 수 있는 센서, 가속도계에 관해 설명했었습니다. 이번 장에서는 동작 제어 부분을 추가합니다.



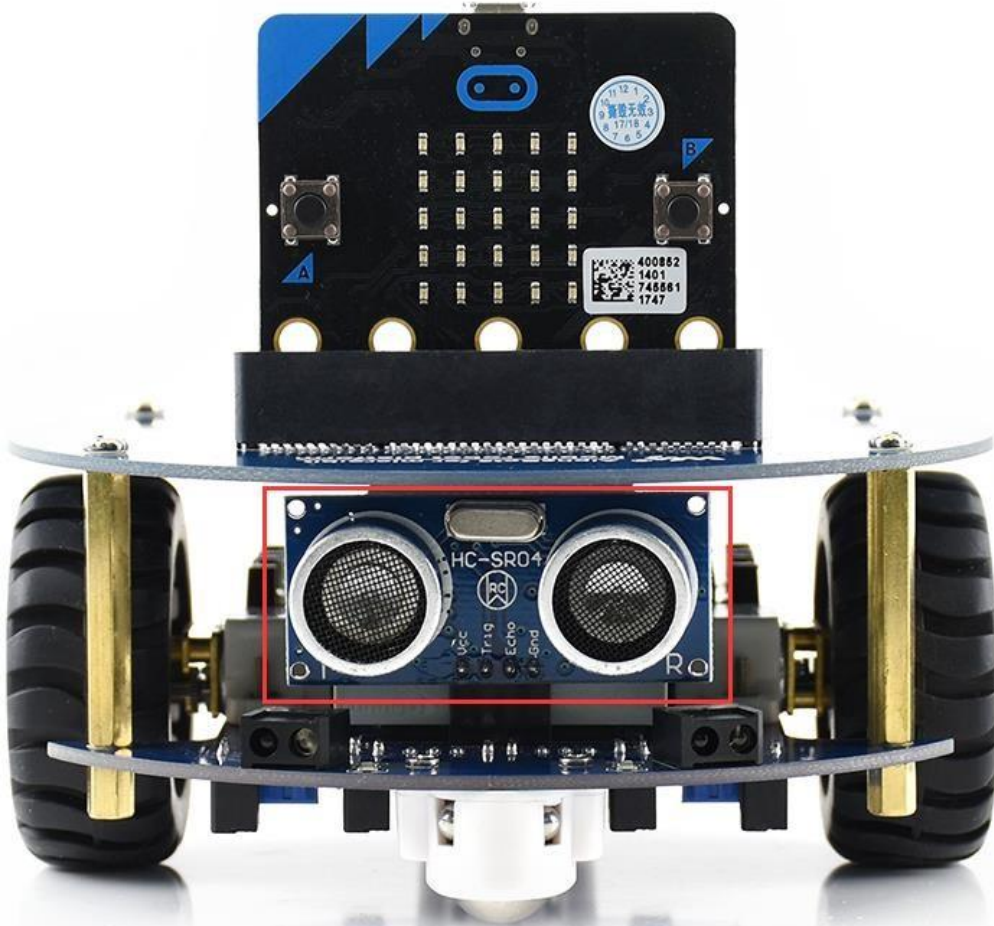
1. 발신자 스크립트



2. 발신자에게 스크립트를 다운로드하고 테스트합니다. 마이크로 비트를 기울이면 로봇이 움직입니다.

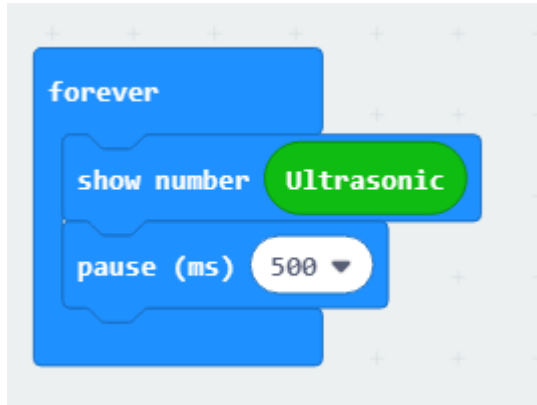
## CHAPTER 10 초음파 장애물 회피

Alphabot2의 눈 역할을 하는 초음파 센서로 장애물 회피에 사용할 수 있습니다.



## 초음파 센서

적외선 센서와 유사하게 초음파 센서에는 송신기와 수신기가 각각 하나씩 있습니다. 초음파 센서는 전방 장애물의 거리를 감지할 수 있습니다. 초음파 센서에서 값을 읽어오는 프로젝트를 새로 만들어 봅시다.



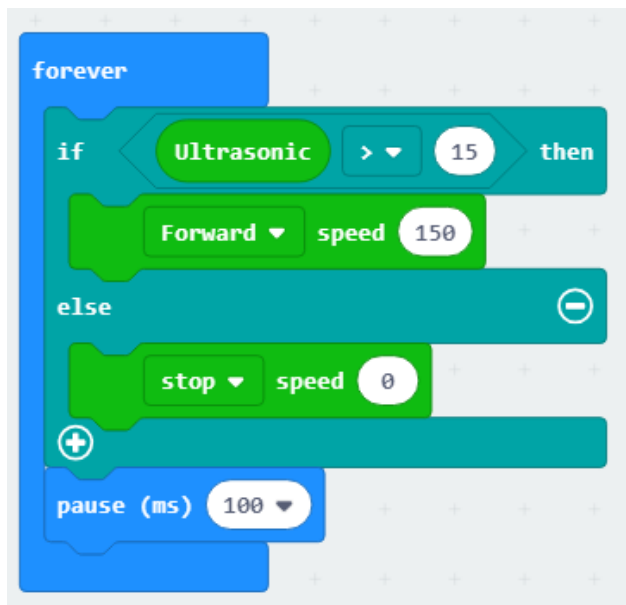
- [ Ultrasonic ] 블록은 Alphabot2 패키지에서 찾을 수 있습니다. 이 블록은 초음파 센서에서 정수인 거리 값을 읽는 데 사용됩니다.

아래 사진 처럼 거리 값을 표시하기 위해 막대 그래프를 그릴 수도 있습니다.



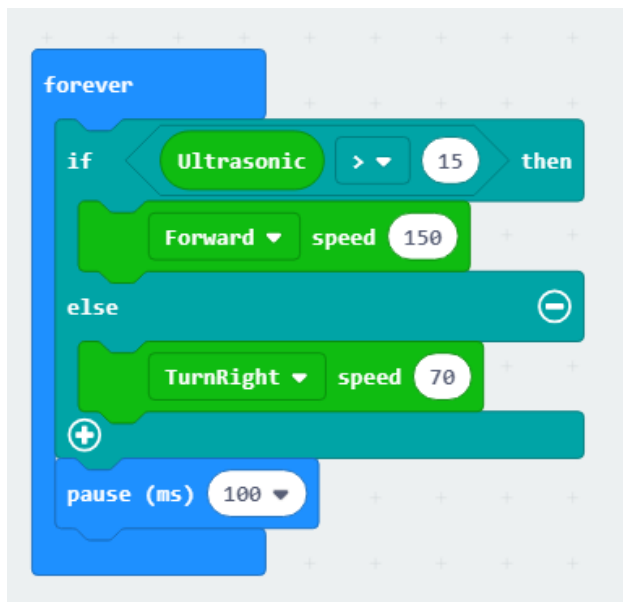
## 장애물 회피

초음파 센서로 전방에 장애물이 있을 경우, 로봇을 멈추게 할 수 있습니다.



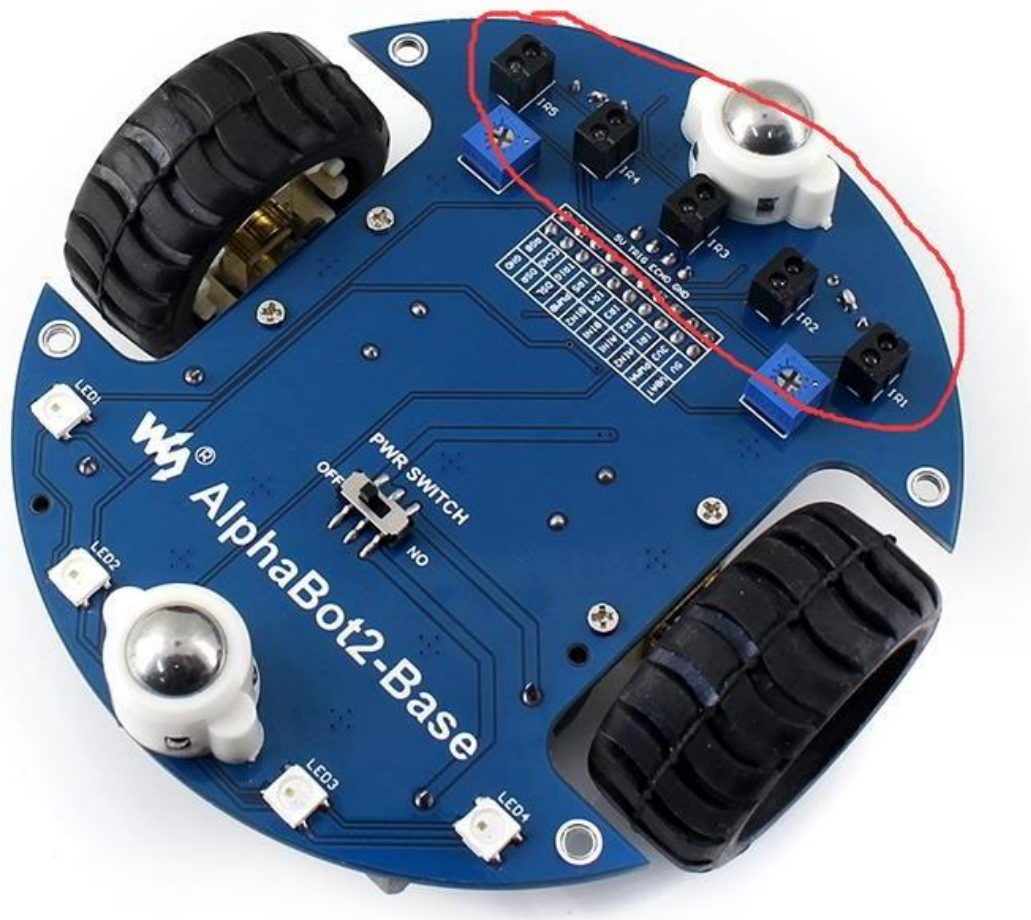
로봇은 전방의 장애물 거리가 15cm보다 길면 계속 전진하고, 15cm보다 짧으면 멈춥니다.

스크립트를 수정하여 장애물이 감지되면 로봇이 우회전하도록 할 수 있습니다.



## CHAPTER 11 라인 트래킹

Alphabot2 base 보드 하단에 추적 기능이 있는 트래커 센서가 내장되어 있습니다.



## 트래커(추적기) 센서

트래커 센서는 로봇이 지면의 검은 선을 따라 움직일 수 있도록 도와줍니다. 센서 하단에 5개의 트래커 센서가 있으며, 적외선 LED 조명을 사용하여 적외선을 지면에 투사하여 검은색 선에 대한 모듈의 편차를 감지합니다.

Alphabot2 패키지 내용:

[ **SensorCalibrated** ] 블록은 정확한 최소 및 최대 값을 위해 센서를 보정하는 데 사용됩니다.

[ **AnalogRead** ] 블록은 적외선 트래커 센서에서 원본 데이터를 읽는 데 사용됩니다.

[ **ReadCalibrated** ] 블록은 보정된 데이터를 읽는 데 사용됩니다.

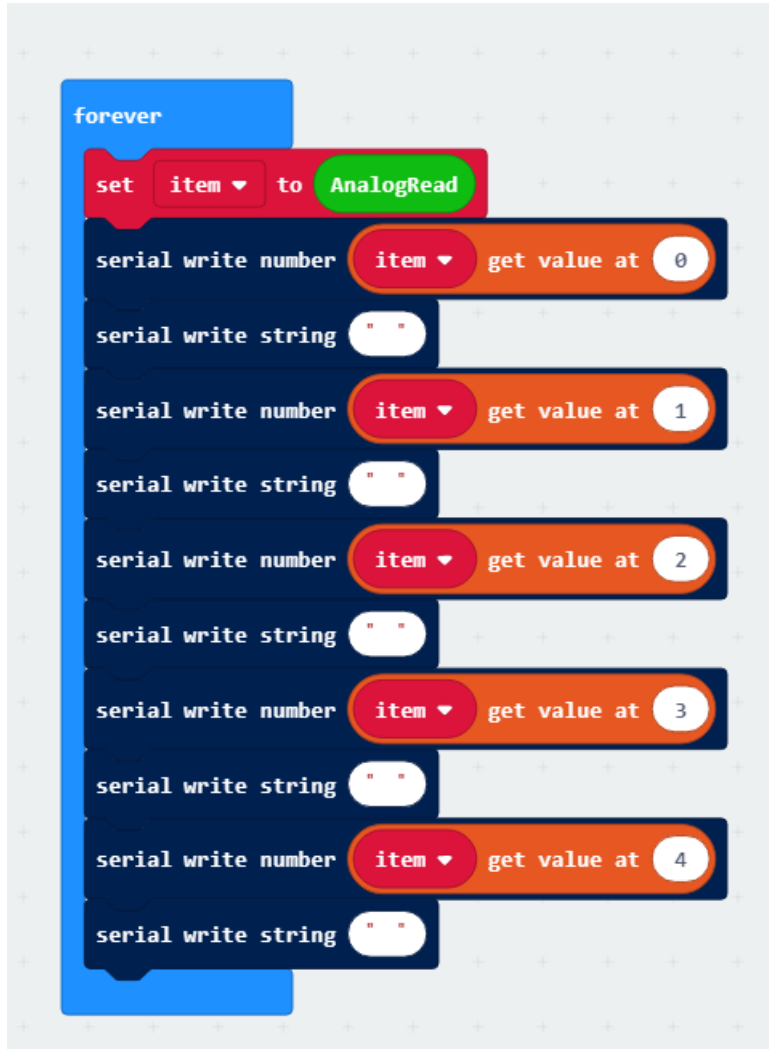
[ **ReadSensorMax** ] 블록은 최대 보장된 값을 읽는 데 사용됩니다.

[ **ReadSensorMin** ] 블록은 보정된 최소값을 읽는 데 사용됩니다.

[ **ReadLine** ] 블록은 0~4000 범위의 검은색 선의 위치를 가져오는 데 사용됩니다. **0**: 로봇이 선 왼쪽에 있습니다. **2000**: 로봇이 선 중앙에 있습니다. **4000**: 로봇이 선 오른쪽에 있습니다.

## 원본 데이터

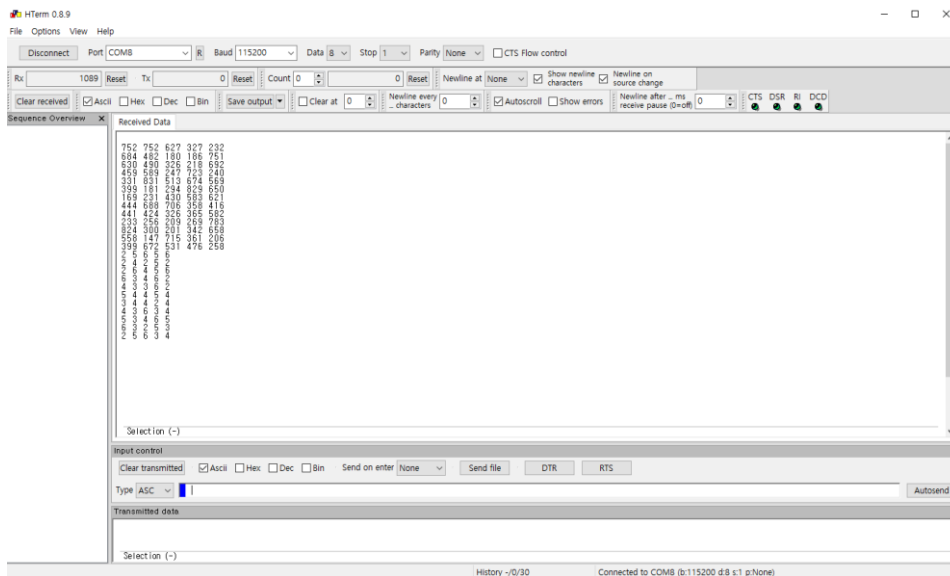
새 프로젝트를 생성하여, 5개의 트래커 센서로 5개의 값을 읽고 다음 주소로 인쇄합니다.



- [ item 에 ( ) 저장 ] 블록은 변수 패키지에서 찾으실 수 있습니다.
- [ 시리얼통신 전송:한 줄 " " ] 블록과 [ 시리얼통신 전송:수 " " ] 블록은 고급 탭에서 시리얼 통신 패키지 에서 찾으실 수 있습니다.
- [ item 의 n 번째 위치의 값 ] 블록은 고급 탭에서 배열 패키지에서 찾으실 수 있습니다.

1. 위와 같이 스크립트를 작성해 스크립트를 완성합니다.

2. 마이크로 비트에 스크립트를 다운로드하고 직렬 소프트웨어를 열고 전송 속도를 115200, 8N1로 설정합니다. 센서의 데이터는 아래와 같이 시리얼로 출력됩니다.



- Alphabot2를 공중에 올려놓으면 출력 값이 0~10 정도로 작은 값이 출력 됩니다.
- Alphabot2를 흰색 바닥에 놓으면 데이터가 500~700까지 올라갑니다.
- Alphabot2를 검은색 선 위에 올려놓으면 약 100의 값으로 출력 됩니다.



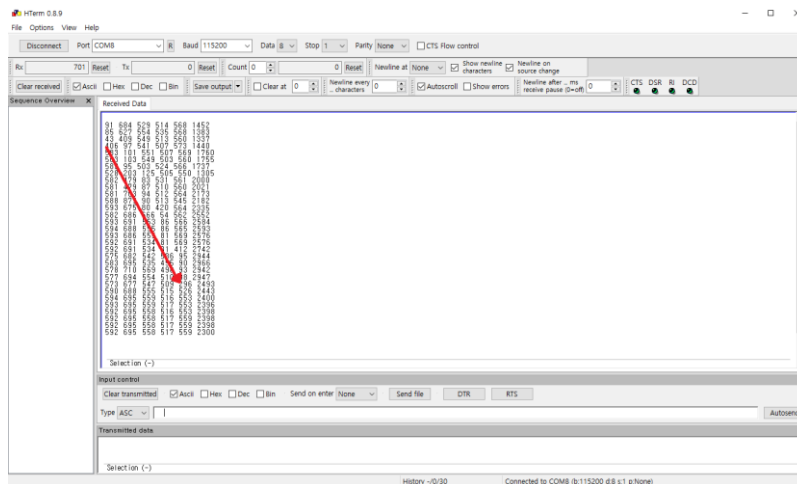
3. 검은 선의 위치를 출력하도록 스크립트를 수정합니다.

```

forever
  set item to AnalogRead
  serial write number item get value at 0
  serial write string ""
  serial write number item get value at 1
  serial write string ""
  serial write number item get value at 2
  serial write string ""
  serial write number item get value at 3
  serial write string ""
  serial write number item get value at 4
  serial write string ""
  serial write number ReadLine
  serial write line ""
    
```

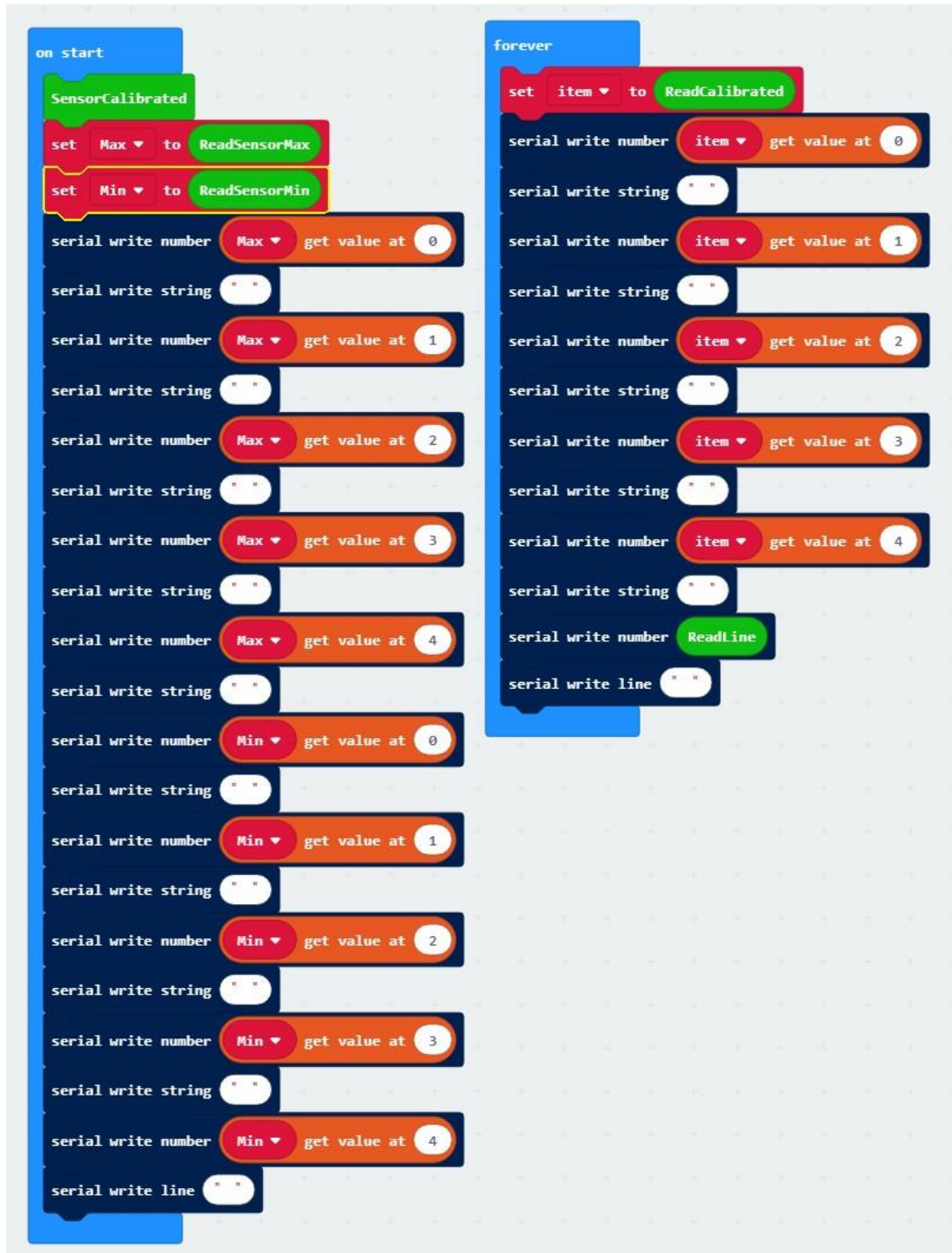
-마이크로 비트에 코드를 다운로드 합니다. 검은 선이 트래커 센서 아래에 있을 때, 데이터는 80~100으로 작아집니다. 가장 오른쪽 데이터가 검은 선의 위치입니다.

이 데이터는 보정되지 않으므로 주의하세요. ( 검은 색 선은 빨간색 화살표에 따라 검출기 1에서 검출기 5로 이동합니다. )



## 보정된 데이터

스크립트를 수정해서 보정된 데이터를 가져옵니다.



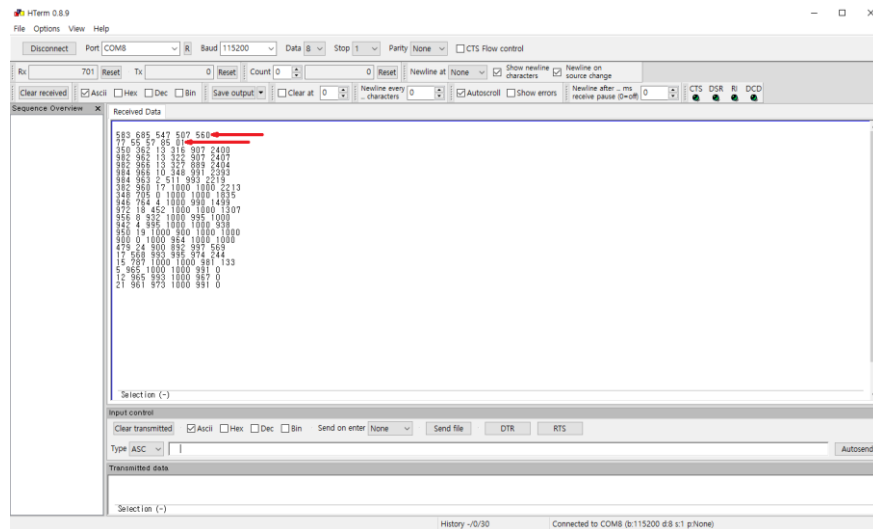
```

on start
  SensorCalibrated
  set Max to ReadSensorMax
  set Min to ReadSensorMin
  serial write number Max get value at 0
  serial write string ""
  serial write number Max get value at 1
  serial write string ""
  serial write number Max get value at 2
  serial write string ""
  serial write number Max get value at 3
  serial write string ""
  serial write number Max get value at 4
  serial write string ""
  serial write number Min get value at 0
  serial write string ""
  serial write number Min get value at 1
  serial write string ""
  serial write number Min get value at 2
  serial write string ""
  serial write number Min get value at 3
  serial write string ""
  serial write number Min get value at 4
  serial write line ""

forever
  set item to ReadCalibrated
  serial write number item get value at 0
  serial write string ""
  serial write number item get value at 1
  serial write string ""
  serial write number item get value at 2
  serial write string ""
  serial write number item get value at 3
  serial write string ""
  serial write number item get value at 4
  serial write string ""
  serial write number ReadLine
  serial write line ""
  
```

-마이크로 비트에 스크립트를 다운로드 합니다.

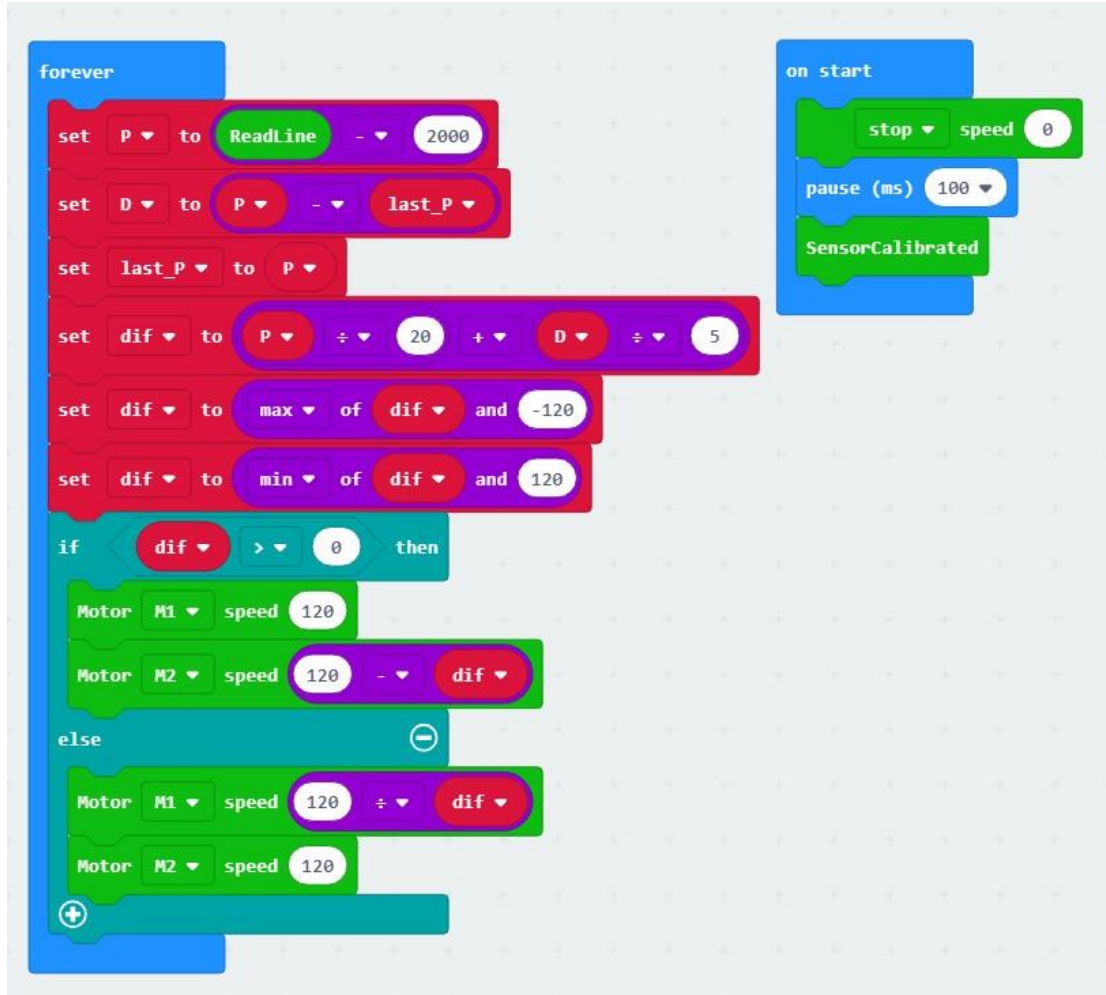
-Alphabot2을 바닥에 놓고 검은색 선이 바닥 중앙에 있는지 확인합니다. 3초 후 (코드 실행)  
 Alphabot2가 캘리브레이션을 시작하고, 먼저 왼쪽으로 회전한 다음 오른쪽으로 회전하고  
 마지막으로 원래 위치로 돌아옵니다. 정확한 트래킹 데이터를 위해 캘리브레이션이 수행됩니다.  
 보정된 데이터는 아래와 같이 출력됩니다.



- 첫 번째 줄은 모든 센서의 최대값을 보여줍니다.
- 두 번째 줄은 모든 센서의 최소값을 보여줍니다.
- 보정된 데이터 흐름. 각 라인마다 트래킹 센서 1부터 트래킹 센서 5까지 보정된 데이터와  
 마지막으로 검은색 선의 위치가 표시됩니다.
- 센서의 보정된 데이터는 0~1000 범위입니다. 값이 작을수록 검은색 선에 가까워집니다.
- 검은색 선의 보정된 위치는 0~4000 범위입니다. 0:극좌측, 2000:중앙, 4000:극우측

## 라인 트래킹(추적)

트래커 센서의 작동 원리를 배웠으니 이제 새 프로젝트를 만들어서 새 프로젝트를 만들어 보겠습니다.



- P는 위치의 허용 오차입니다. 양수: Alphabot2이 오른쪽으로 편차, 음수: Alphabot2이 왼쪽으로 편차.

- D는 현재 P와 마지막 P의 차이로, D가 클수록 Alphabot2의 반응이 빨라집니다.

- dif는 모터가 변경해야 하는 차이이며, 이 값은 P와 D를 기준으로 합니다.

이상으로, Alphabot2에 대한 매뉴얼을 마치겠습니다.