
마이크로 비트를 이용한

코딩 교육 시작하기

 micro:bit



 LK EMBEDDED

이경남 저

목차

제 1 장 마이크로 비트 (MICRO:BIT)	3
1.1 마이크로 비트 탄생	3
1.2 마이크로 비트 회로도(CIRCUIT)	3
1.3 마이크로 비트 하드웨어(HARDWARE)	10
1.4 마이크로 비트 소프트웨어(SOFTWARE)	13
제 2 장 마이크로 비트 실습	24
2.1 LED 실습	24
2.2 부저 실습	28
2.3 버튼(스위치) 실습	32
2.4 가속도 센서 실습	34
2.5 나침반 실습	38
2.6 외부 LED 실습	41

제1장 마이크로 비트 (Micro:bit)

1.1 마이크로 비트 탄생

마이크로 비트는 영국 BBC 에서 2015 년도, 'IT 교육'을 활성화 시키자는 목적에서 개발 되었습니다. 어린 학생을 대상으로 개발이 진행되었기 때문에, 디바이스의 성능 및 '안정성'뿐 아니라 디바이스를 제어하는 접근성 및 결과를 확인할 수 있는 '직관성'까지 염두를 두고 개발이 이루어졌습니다.

이 사업에는 마이크로소프트(Microsoft)와 삼성(Samsung)등의 굵직한 기업들이 참가하였으며, 그 결과 지금의 '마이크로 비트(Micro:bit)'가 탄생하였습니다.

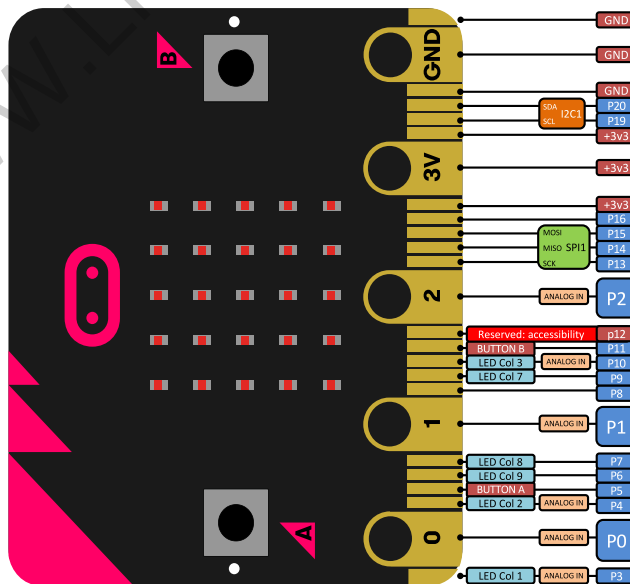


<그림 1-1-1> 마이크로 비트(Micro:bit)

마이크로 비트는 뛰어난 '접근성'과 '직관성'을 바탕으로 전세계 코딩교육의 기자재로 널리 사용되고 있습니다.

1.2 마이크로 비트 회로도(Circuit)

1.2.1 개요



<그림 1-2-1> 엣지 커넥터 핀(Edge Connector Pins)

본 장에서는 마이크로 비트의 전기적 연결구조와 BOM(Bill Of Materials) 즉, 구성요소에 대해 설명합니다. 해당 자료들은 BBC Microbit GitHub 사이트인 [<https://github.com/bbcmicrobit/hardware>]를 통하여 다운로드 받으실 수 있으며, 엘케이임베디드 자료실을 방문하셔도 확인하실 수 있습니다.

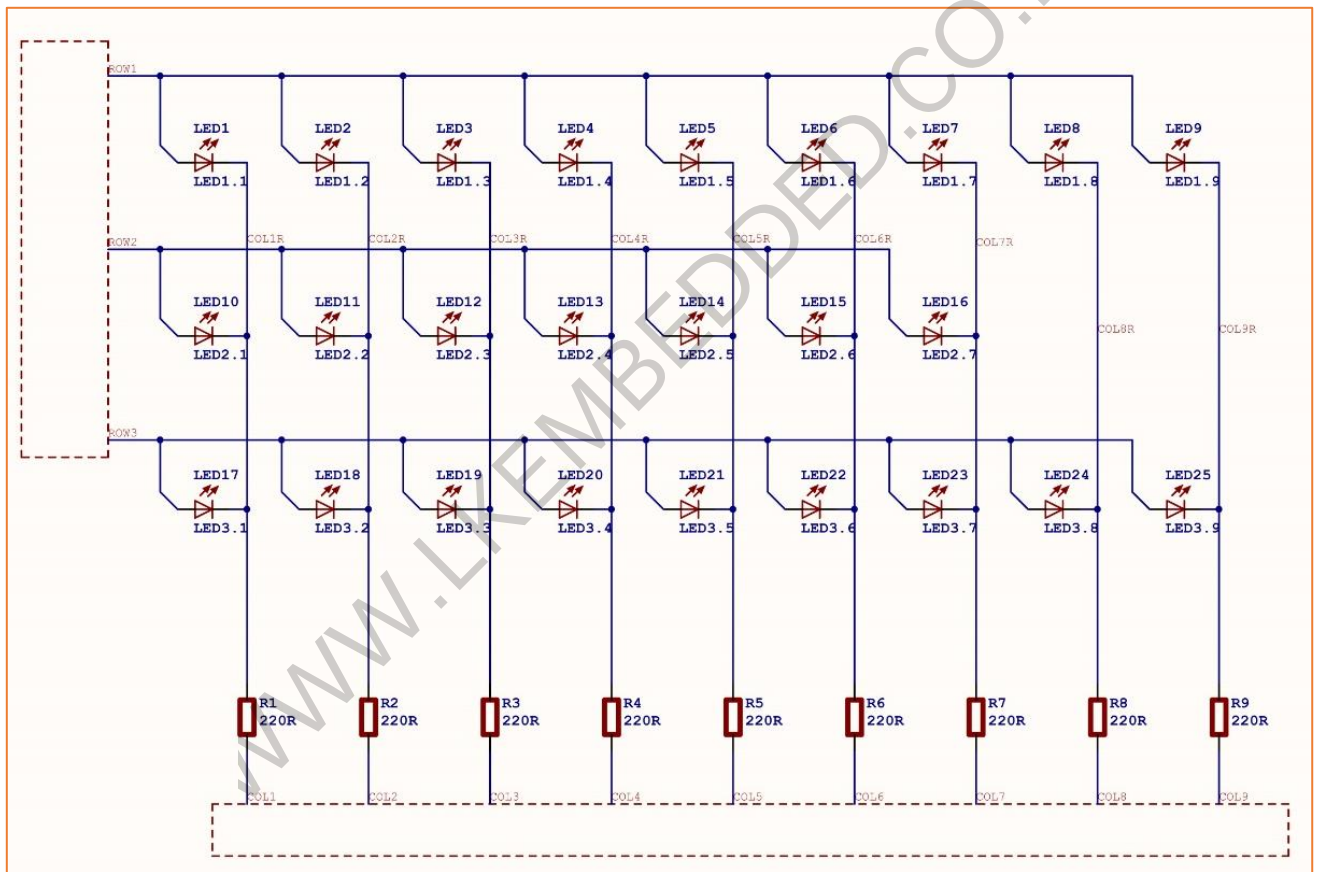
해당 회로도를 이해함으로, 마이크로 비트에 내장된 센서 외의 다른 기능을 갖고 있는 모듈을 마이크로 비트와 연동할 수 있습니다. 또는 마이크로 비트의 강력한 무선 기능(nRF51822)을 타 프로세스(Processor)에서 마이크로 비트와 연결을 해줌으로 사용이 가능합니다. (해당 내용은 본 장에서 다루지 않습니다.)

1.2.2 주요특징

마이크로 비트를 사용하는데 있어, 핵심적인 요소들은 아래와 같습니다.

- ✧ LED 9X3
- ✧ 인터페이스
- ✧ 센서
- ✧ 전원공급장치
- ✧ 응용프로그램 프로세서
- ✧ 에지 커넥터

✓ LED 9 x 3

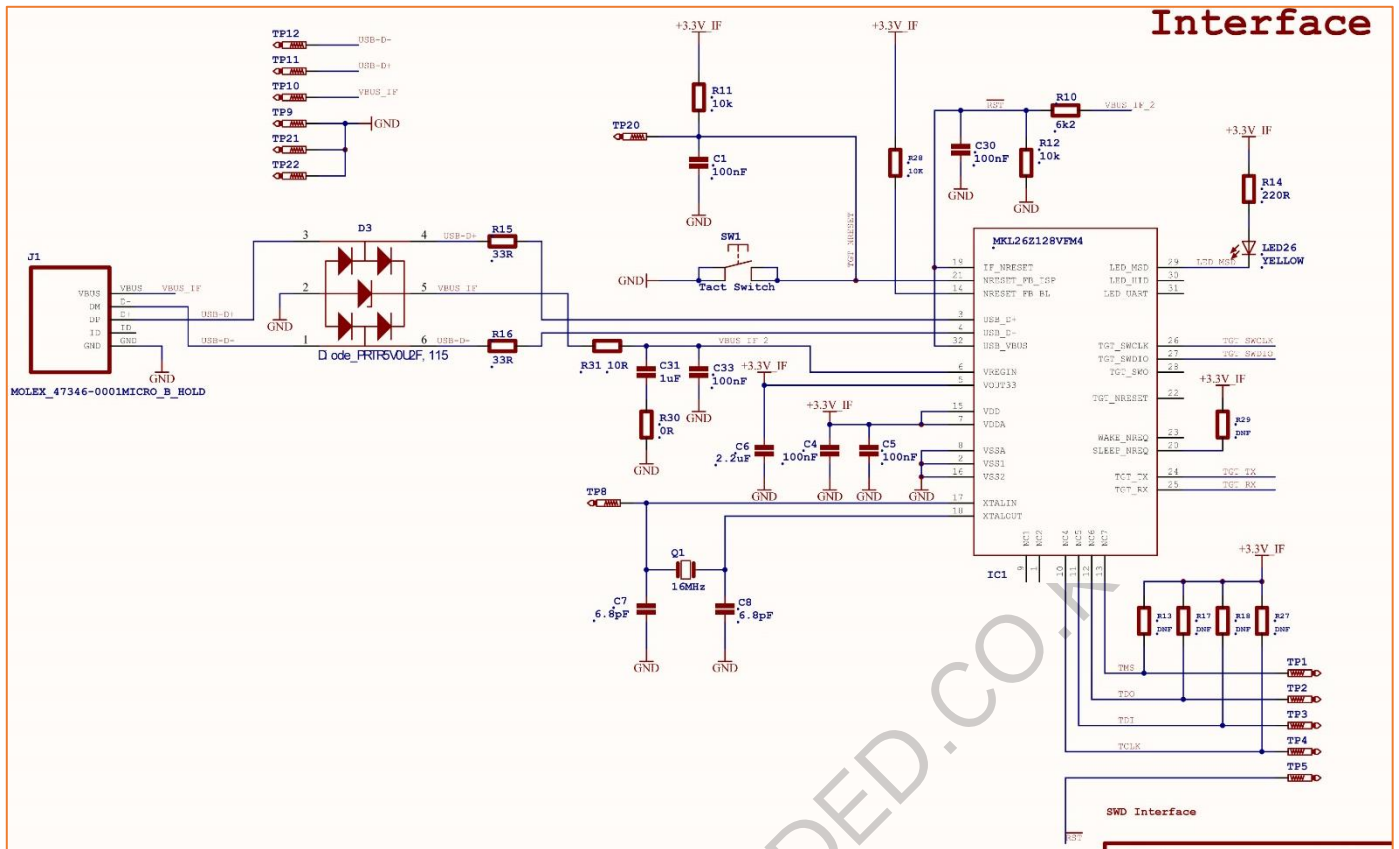


<그림 1-2-2> LED 매트릭스 회로도

마이크로 비트의 전면부를 보면, LED 5x5 로 배치되어 있습니다. '왜 LED 9x3 이지?' 의문을 갖는 분들이 있겠지만, 위의 회로도를 확인해보면 전기적인 연결은 9x3 의 매트릭스(Maxrix - 즉, 9 행 3 열)로 설계되었습니다. 그 중, '2 행 8 열'과 '2 행 9 열'은 사용되지 않아 총 25 개의 LED 만 동작이 됩니다.

이 LED 매트릭스는 소프트웨어를 통해 구현된, '고속멀티플렉스(High-Speed multiplex)' 방식을 사용하여 동작됩니다. 그렇기 때문에, 해당 LED 의 행과 열에 관련된 GPIO 핀들을 다른 용도로 사용하는 동안은 해당 기능 ('고속멀티플렉스(High-Speed multiplex)')을 사용할 수 없습니다.

✓ 인터페이스

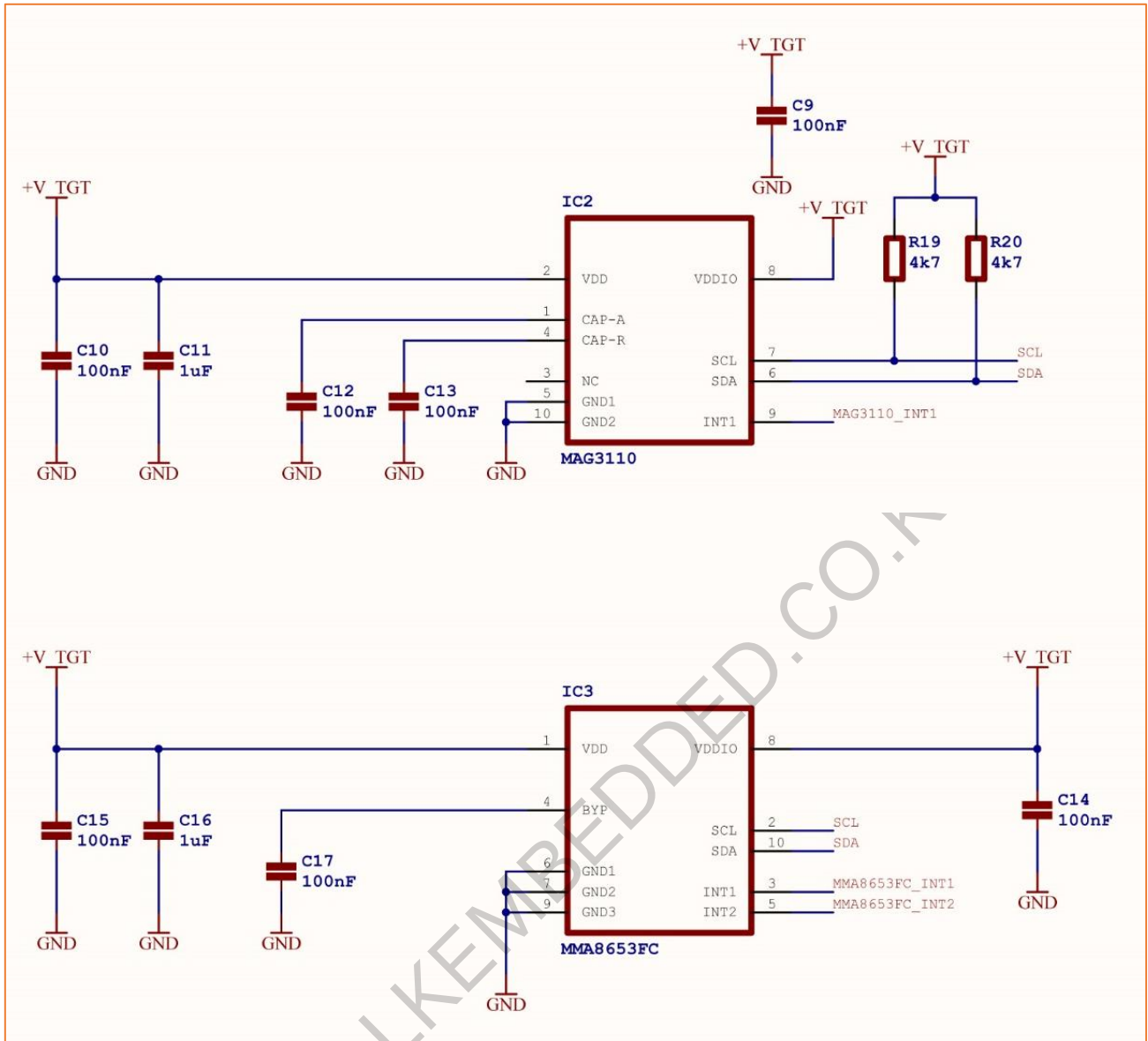


<그림 1-2-3> 인터페이스 회로도

인터페이스 회로도에서는 USB 프로토콜을 이용한 ARM 프로세서인 'KL26' 를 사용한 회로를 보여줍니다. 이 회로도는 사용자가 '.HEX 파일'을 마이크로 비트에 저장해주면, 해당 파일을 로드(Load)하여 프로그래밍을 해주는 역할을 합니다. 이는 별도의 프로그래머 장치가 없어도 마이크로 비트를 쉽게 사용할 수 있는 이유 중 하나 입니다.

또한 온-보드 레귤레이터(On-Board Regulator)가 내장되어 있어, 외부 USB 커넥터로부터 입력받은 5V 전압을 3.3V로 낮춰주어, 120mA의 전류를 공급할 수 있습니다. 전원공급간 USB 커넥터에서 발생할 수 있는 ESD(Electro Static Dscharge, 정전기 방출) 스파크 및 과전압을 TVS 다이오드를 통하여 방지합니다.

✓ 센서



<그림 1-2-4> 센서 회로도

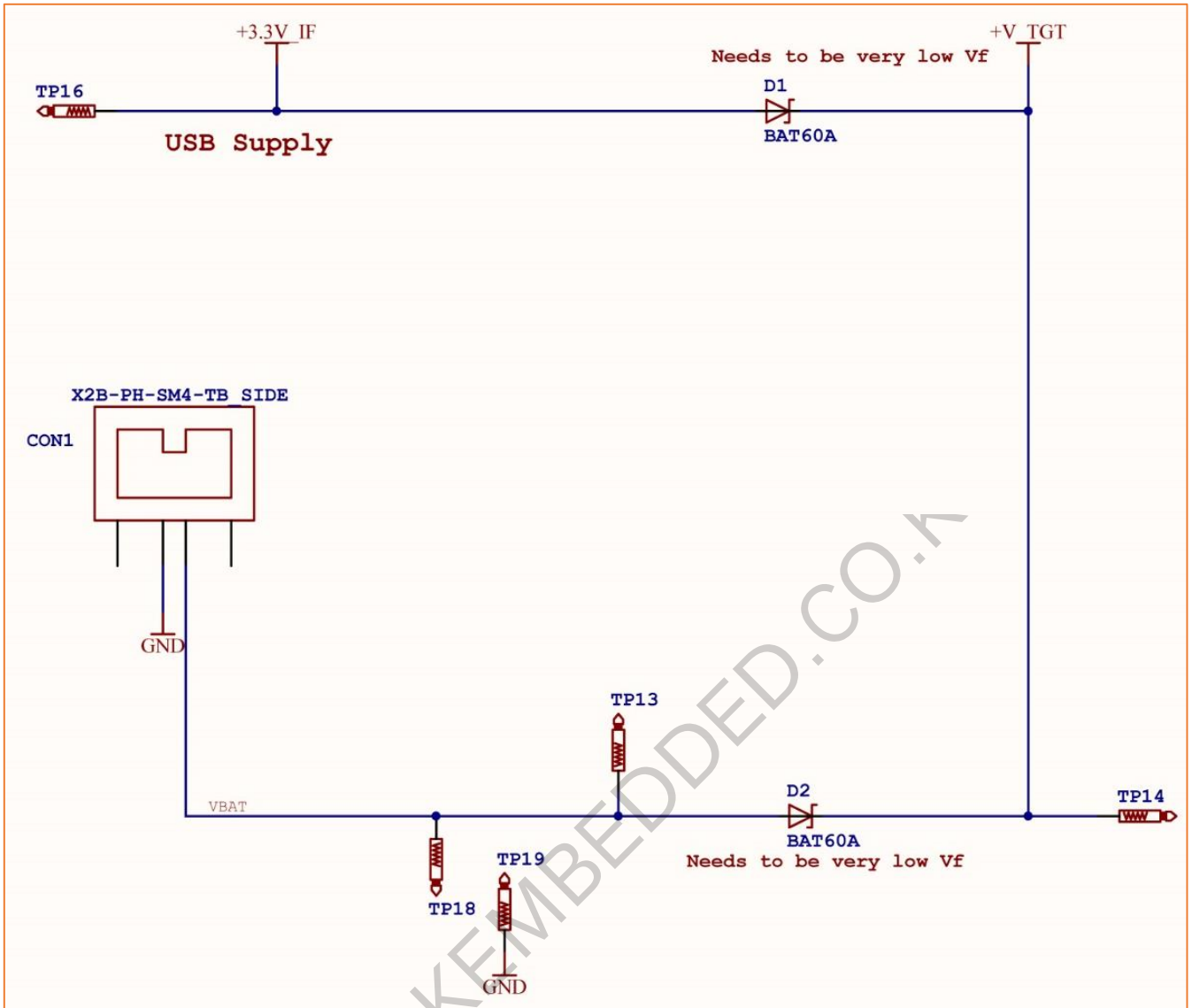
마이크로 비트에는 가속도계 및 자력계를 측정할 수 있는 IC가 2개 장착되어 있습니다. 가속도계는 3축(X, Y, Z)의 가속도를 측정할 수 있으며, 자력계는 자기장 감지뿐만 아니라 나침반 역할로도 사용할 수 있습니다.

가속도계, 자력계 두 장치는 풀 업 저항을 통하여 프로세서의 I2C 버스(BUS)에 연결되어 있으며, 동시에 마이크로 비트의 에지 커넥터(Edge Connector)의 두 핀에도 연결이 되어 있습니다.

자력계 IC인 MAG3110 (회로도 상단)은 하나의 인터럽트를 생성할 수 있으며, 가속도계 IC인 MMA8653FC는 2개의 인터럽트를 생성할 수 있습니다. 더 자세한 사항은 각 IC의 데이터시트(Datasheet)를 통하여 확인하실 수 있습니다.

참고로 두 IC가 장착이 되는 방향에 따라서 프로세서에서 연산되어지는 값이 달라질 수 있기 때문에, 실제로 장착되는 방향이 중요합니다.

✓ 전원 공급 장치



<그림 1-2-5> 전원 공급 장치 회로도

마이크로 비트는

- 1) USB 커넥터
- 2) 배터리 커넥터
- 3) 에지 커넥터의 3V 패드

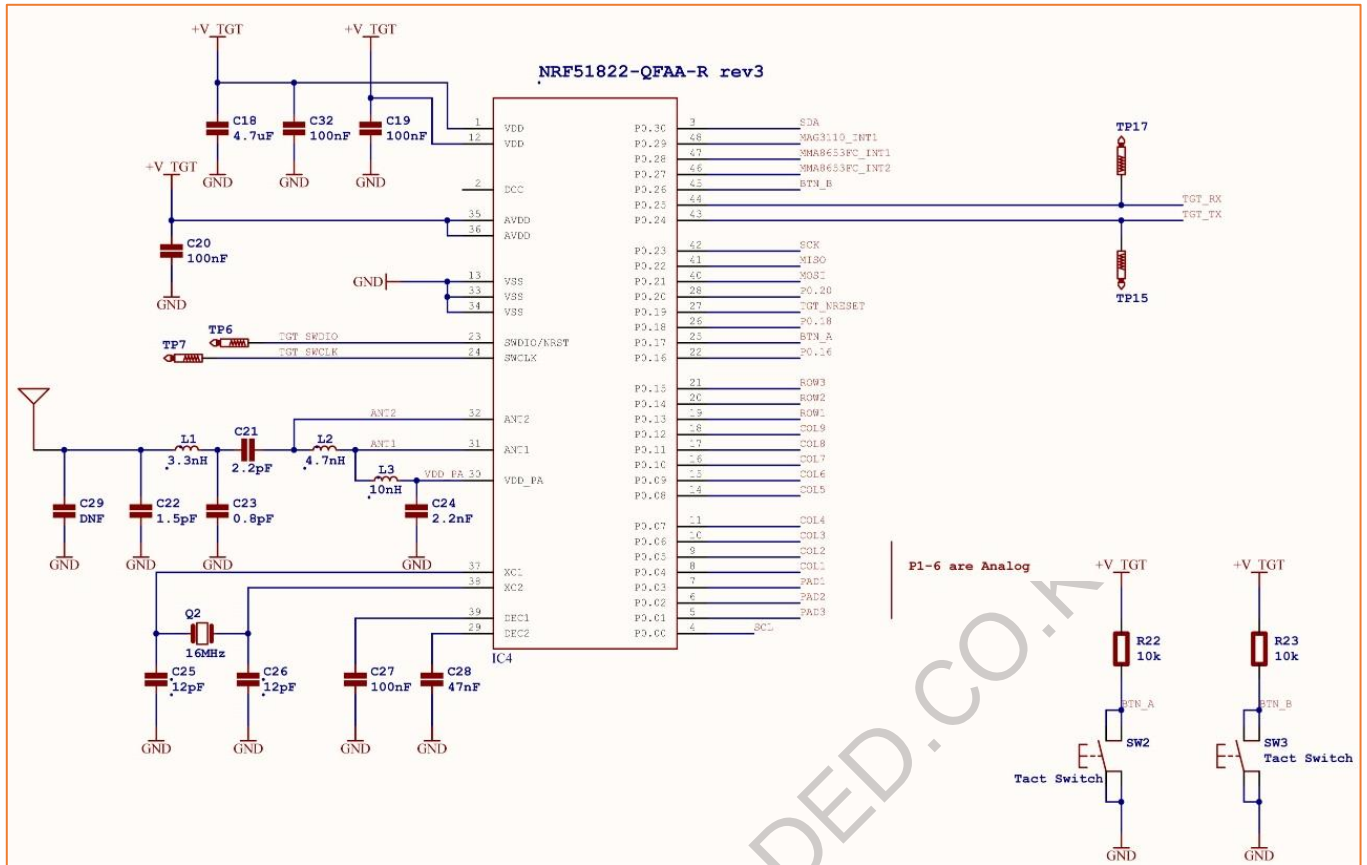
총 3 가지 커넥터를 통하여 전원을 입력할 수 있습니다.

USB 커넥터를 통한 전원 공급의 경우 앞서 설명한 바와 같이, 'KL26' 인터페이스 프로세서에 온-보드 레귤레이터(On-Board Regulator)가 내장되어 있어 외부 USB 전압 5V를 마이크로 비트에 알맞는 3.3V 전압으로 변환하여 공급할 수 있습니다.

낮은 V_f (Forward Voltage, 순방향 전압)을 갖고 있는 다이오드는 다른 전원 공급원으로부터 역전력(back-powering)을 방지함으로, 전원 공급을 원활하게 하기 위하여 사용됩니다. (이 회로에서 사용된 다이오드의 순방향 전압 V_f 는 최대 약 0.23V)

특히, 에지 커넥터의 3V 패드가 보드의 IC에 직접 연결되어 있으므로 패드를 통하여 마이크로 비트 전원을 공급하는 경우, 주의를 기울여야 합니다.

✓ 응용 프로그램 프로세서

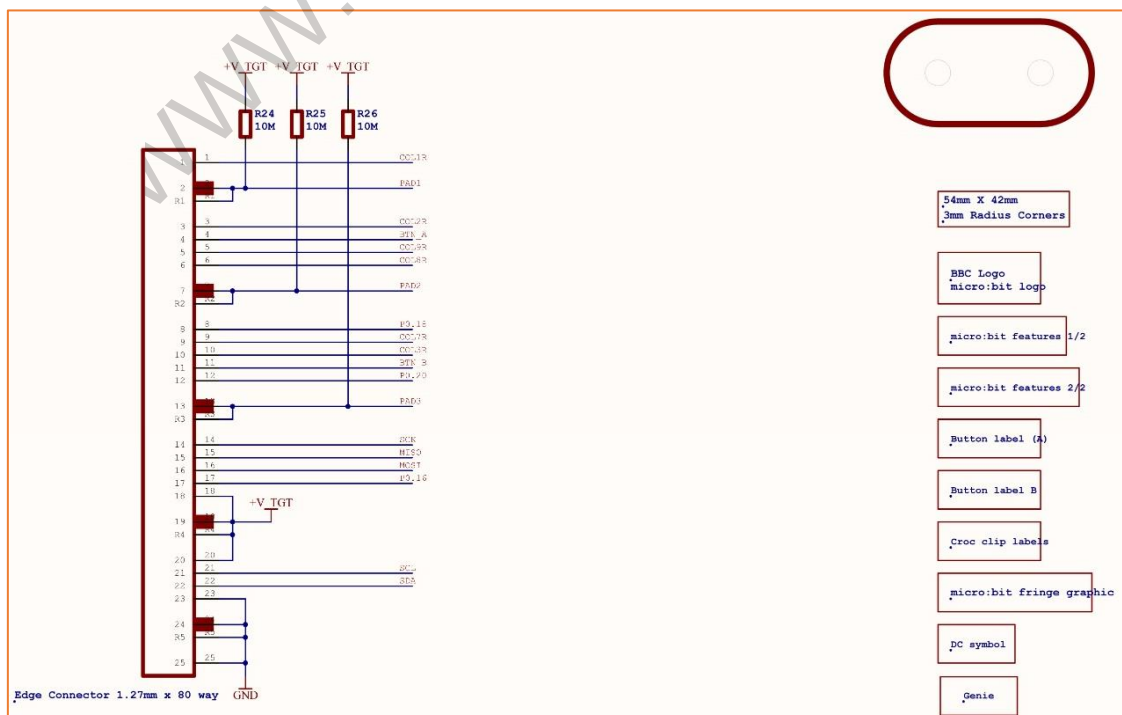


<그림 1-2-6> 응용 프로그램 프로세서 회로도

사용자가 작성한 프로그램은 상단에서 소개한 USB 인터페이스를 통하여 해당 메인 프로세서에 Runtime 코드와 함께 프로그래밍이 됩니다.

에지 커넥터의 모든 GPIO 핀은 해당 메인 프로세서에 의해 처리되며, 버튼 입력도 메인 프로세서로 처리됩니다. 또한 저전력 블루투스 기능 내장되어 있어, 해당 메인 프로세서를 통하여 동작됩니다.

✓ 에지 커넥터



<그림 1-2-7> 에지 커넥터 회로도

에지 커넥터는 마이크로 비트와 외부 구성 요소간의 연결을 위한 기본 인터페이스이며, 외부 회로를 구성하는데 사용됩니다.

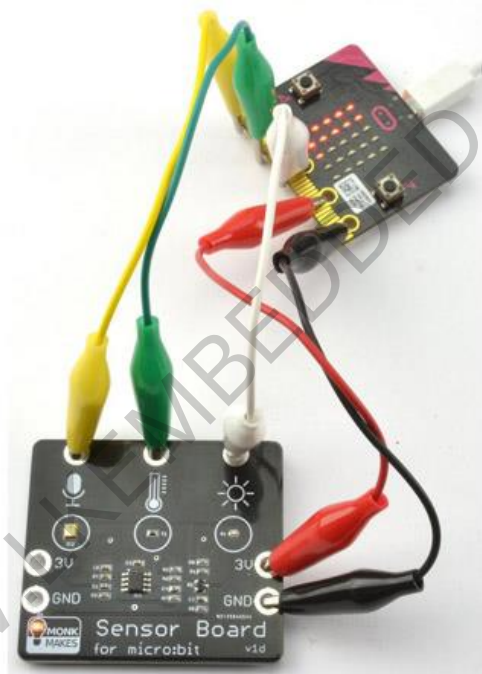
‘4mm 바나나 플러그’ 또는 ‘악어클립’과 함께 사용할 수 있도록 5 개의 고리(Rings) 모양의 커넥터가 있으며, 이 중 3 개는 일반 입,출력 GPIO 핀이며 2 개는 마이크로 비트에 전원 공급을 할 수 있는 3V, GND 용도의 패드입니다.

이 외에도 디지털, 아날로그, 터치, PWM 및 시리얼 통신 등의 다양한 기능을 할 수 있습니다.

P0, P1 및 P2 에 10M ohm 의 약한 풀업저항이 장착되어 있습니다. 이를 이용하여, 사용자가 한손에 GND 패드를 만지고, 해당 패드(P0, P1, P2) 를 만진다면 해당 핀은 0V로 내려가게 됩니다. 이는 터치센서 역할을 할 수 있습니다.

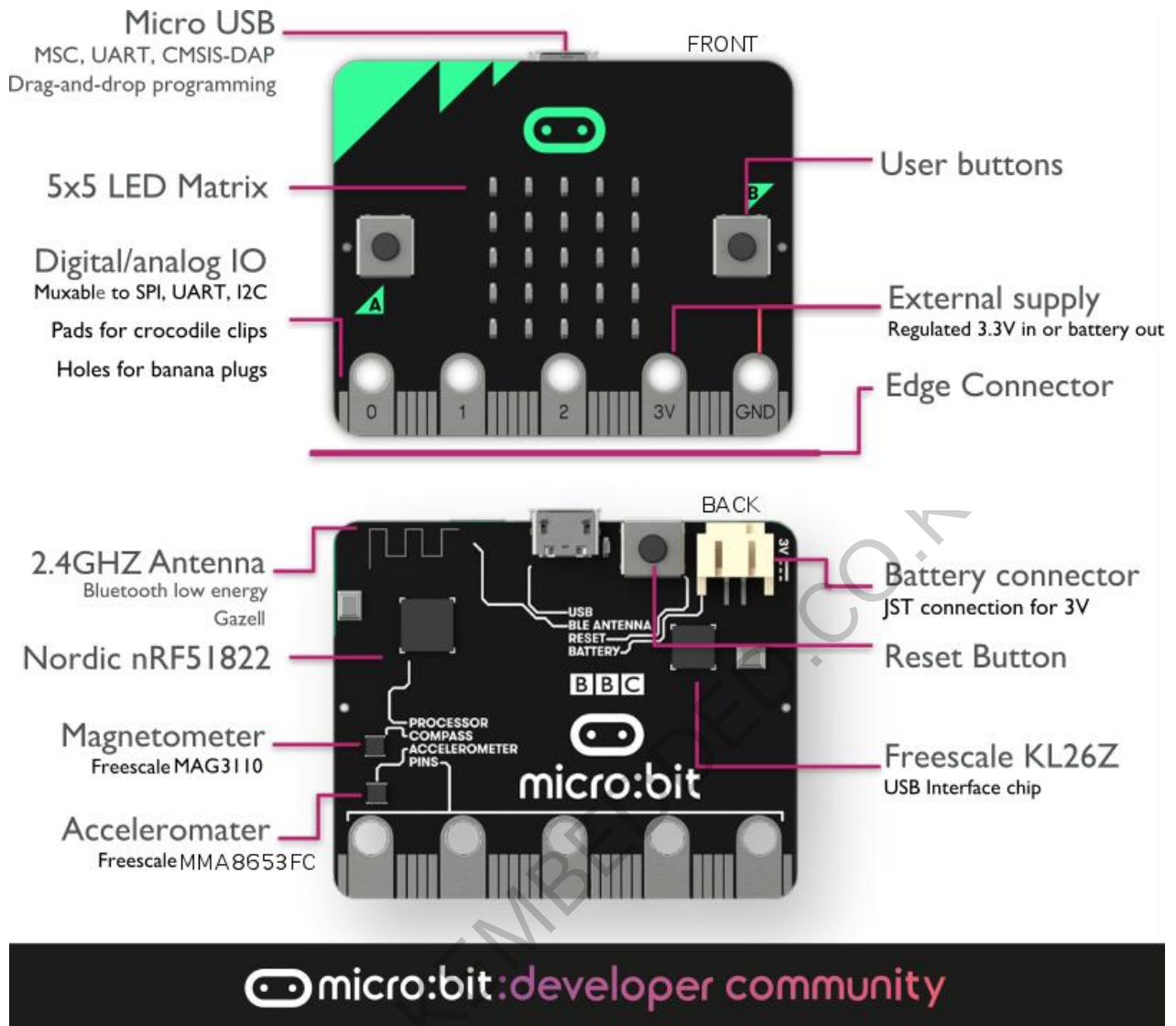
각 핀은 사용되지 않을 때, 소프트웨어를 통한 풀다운 설정으로 인하여, 로우(LOW, 0V)를 유지합니다. 이는 디바이스의 오작동을 막아줄 수 있습니다.

마이크로 비트의 많은 GPIO 핀들이 각각의 기능을 가지고 있지만, 일반 GPIO 목적으로 해당핀을 쓰기를 원한다면 해당 기능을 해제(Disable)하여 사용할 수 있습니다.



<그림 1-2-8> 악어클립을 이용하여 마이크로 비트(microbit)와 센서보드를 연결 한 모습

1.3 마이크로 비트 하드웨어(Hardware)



<그림 1-3-1> 마이크로 비트 각 부품별 위치

항목	상세설명
Nordic nRF518 Application Processor / 프로세서	
모델(Model)	Nordic nRF51822-QFAA-R rev 3
코어 계열(Core variant)	ARM Cortex-M0 32 bit 프로세서
플래쉬 롬(Flash ROM)	256KB
램(RAM)	16KB
클럭속도(Clock speed)	16MHz
Bluetooth Wireless Communication / 블루투스 통신	
스택(Stack)	Bluetooth 4.1 with Bluetooth low energy
대역폭(Band Width)	2.4GHz ISM(Industrial, Scientific and Medical) 2.4GHz ~ 2.41GHz
Low Level Radio Communication / RF 통신	
프로토콜(Protocol)	Nordic Gazell
주파수 대역(Freq band)	2.4GHz
통신속도(Channel rate)	1Mbps or 2Mbps

Buttons / 버튼	
사용처(Type)	입력버튼(A & B) 2 개와 리셋(Reset)용 버튼 1 개
입력주기(debounced)	입력버튼(A & B)은 소프트웨어적으로 54ms 주기를 갖음
풀업저항(PullUp)	외부버튼 4K7 옴, 시스템 10K 옴
Display / LED 디스플레이	
모양(Type)	소형 빨간색 LED(PCB 표면실장형)
물리적(Physical) 구조	5x5 매트릭스
전기적(Electrical) 구조	3x9
Accelerometer / 가속도(자이로) 센서	
모델(Model)	Freescale MMA8653FC
특성(Features)	3 축, 2/4/8g(중력가속도) 범위
분해능(Resolution)	10 bits (0 ~ 1023)
최대 데이터 출력 속도	800Hz
Magnetometer / 자기(나침반) 센서	
모델(Model)	Freescale MAG3110
최대 업데이트 속도	80Hz
Full Scale range	1000uT
Sensitivity	0.10uT
Temperature Sensing / 온도 센서	
Type	on-core nRF51
Sensing range	-25°C ~ 75°C
Resolution	0.25C steps
General Purpose Input/Output Pins / GPIO 핀	
Rings(링)	마이크로 비트는 3 개의 IO 링과 2 개의 전원 링이 있습니다. 이는 4mm 의 플러그 또는 악어클립을 사용하기 적합합니다.
GPIO features(특성)	사용이 가능한 19 개의 GPIO 핀이 있습니다.
	I2C 통신이 가능한 2 개의 핀이 있습니다.
	6 개의 핀은 전면 LED 를 제어하는 데 사용됩니다.
	2 개의 핀은 버튼의 입력을 감지합니다.
	리셋 버튼을 감지하여, 동작을 초기화 합니다.
	19 개의 핀은 모두 디지털 입력/출력으로 사용 가능합니다.
	19 개의 핀 중, 최대 3 개까지 PWM 출력으로 사용될 수 있습니다.
	19 개의 핀 중, 시리얼 송수신 핀으로 1 개씩 사용 가능합니다.
	6 개의 핀은 아날로그 입력 핀으로 사용 가능합니다.
	3 개의 핀은 옵션 선택으로 SPI 통신용으로 사용 가능합니다.
	3 개의 핀은 터치 감지 입력으로 사용될 수 있습니다.
ADC resolution(분해능)	10 bit (0 ~ 1024)
Pitch(핀 간격)	1.27mm 80 way double sided
Pads	5 pads, with 4mm holes
Power Supply / 전원공급	

동작 전원	1.8V ~ 3.6V
USB 전류	120mA max
평균 전류 소모량	30mA
배터리 커넥터	JST 사의 모델명 X2B-PH-SM4-TB
Interface (USB connection) / USB 연결 인터페이스	
모델	Freescale MKL26Z128VFM4
코어 계열(Core variant)	ARM Cortex-M0+
플래쉬 롬(Flash ROM)	128KB
램(RAM)	16KB
클럭속도(Clock speed)	16MHz
디버그(Debug) 기능	SWD(Serial Wire Debug)
USB Communication / USB 통신	
커넥터	USB micro, MOLEX_47346-0001
USB 버전(version)	1.1 Full Speed device
통신 속도(Speed)	12Mbit/sec
USB 지원 클래스(Class)	대용량 저장 클래스 - Mass Storage Class(MSC)
	통신 디바이스 클래스 - Communications Device Class(CDC)
Debugging / 디버깅	
통신규격(프로토콜)	CMSIS-DAP
디버그 옵션(Options)	J Link/OB (via different firmware)
Mechanical / 하드웨어 정보	
크기(Dimensions)	5 cm(w) x 4 cm(h)
무게(Weight)	5g

1.4 마이크로 비트 소프트웨어(Software)

1.4.1 프로그래밍 개발 환경과 IDE

✓ 자바스크립트 블록 - Javascript Blocks(powered by Makecode) : [<https://makecode.microbit.org/>]

마이크로 비트를 사용할 때, 일반적으로 사용하게 되는 자바스크립트 블록입니다. 이 소프트웨어는 자바스크립트를 이용하기 때문에 별도의 펌웨어가 필요하지 않으며, explorer 나 chrome 등의 웹 프로그램을 통하여 프로그래밍이 가능합니다.

코딩뿐만 아니라, 내가 만들어 놓은 블록이 어떻게 구현되는지 시뮬레이션 결과도 확인할 수 있어, 마이크로 비트를 이용하시는 초보자 분들도 쉽게 이해하며 사용할 수 있습니다. 또한 구성된 블록을 자바스크립트 언어로 변환할 수 있어 프로그래밍 언어를 이해하고자 하시는 분에게도 유용합니다.

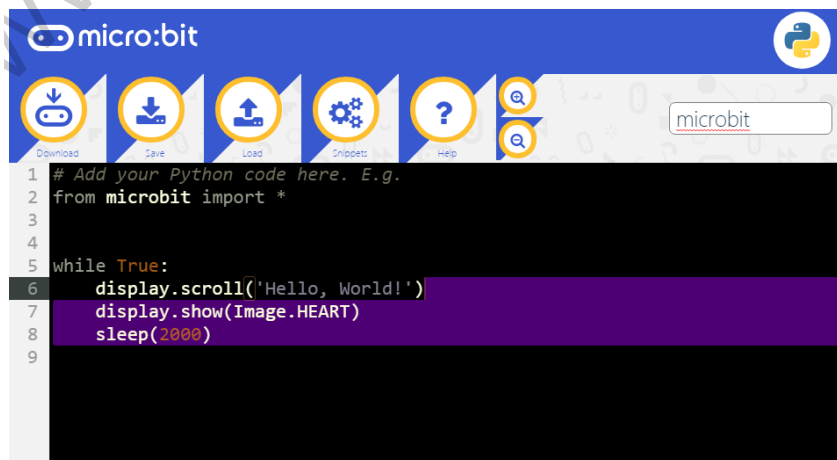


<그림 1-4-1> 마이크로 비트 자바스크립트 블록 웹페이지

✓ 파이썬 - Python : [<https://python.microbit.org/v/1>]

마이크로 비트는 파이썬과도 협약하여 적용이 가능하도록 하였으며, 웹 프로그램을 통하여 프로그래밍이 가능하도록 하였습니다. 즉, 별도의 프로그램 설치 없이 웹을 통하여 코딩 및 프로그래밍이 가능합니다.

파이썬은 간결하고 생산성이 높은 프로그래밍 언어로, 입문자들도 쉽게 이해할 수 있으며, 다양한 분야에 활용할 수 있습니다.



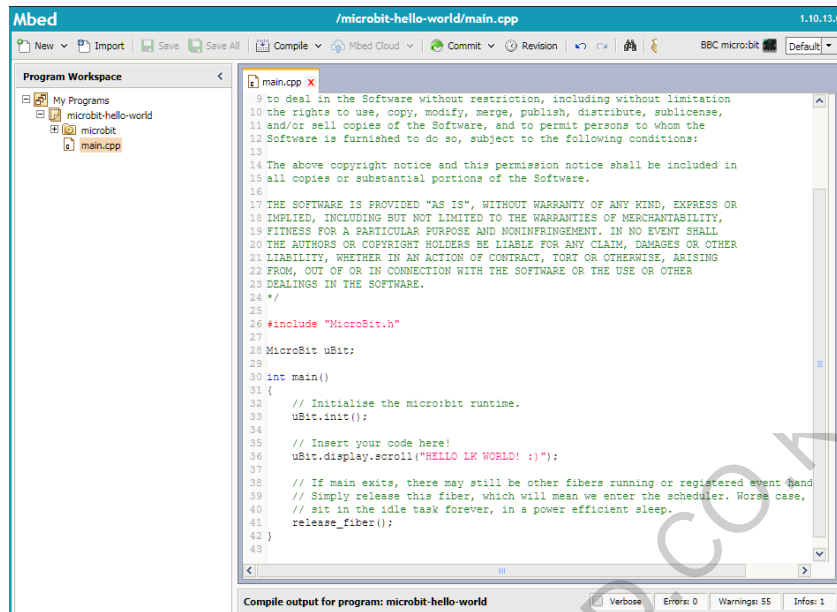
<그림 1-4-2> 마이크로 비트 파이썬 웹페이지

✓ Mbed 온라인 컴파일러 - Mbed Online Compiler : [<https://os.mbed.com/platforms/Microbit>]

ARM사에서 제공하는 Mbed 컴파일러를 사용하면 C/C++ 언어로 코딩이 가능합니다.

위에서 소개했던 개발도구와 마찬가지로 웹 프로그래밍(on-line)이 가능하며 설치 프로그램(off-line)을 통하여도 실행이 가능하며, 별도의 디버거 및 프로그래머가 없이도 .hex 파일이 다운로드 가능합니다.

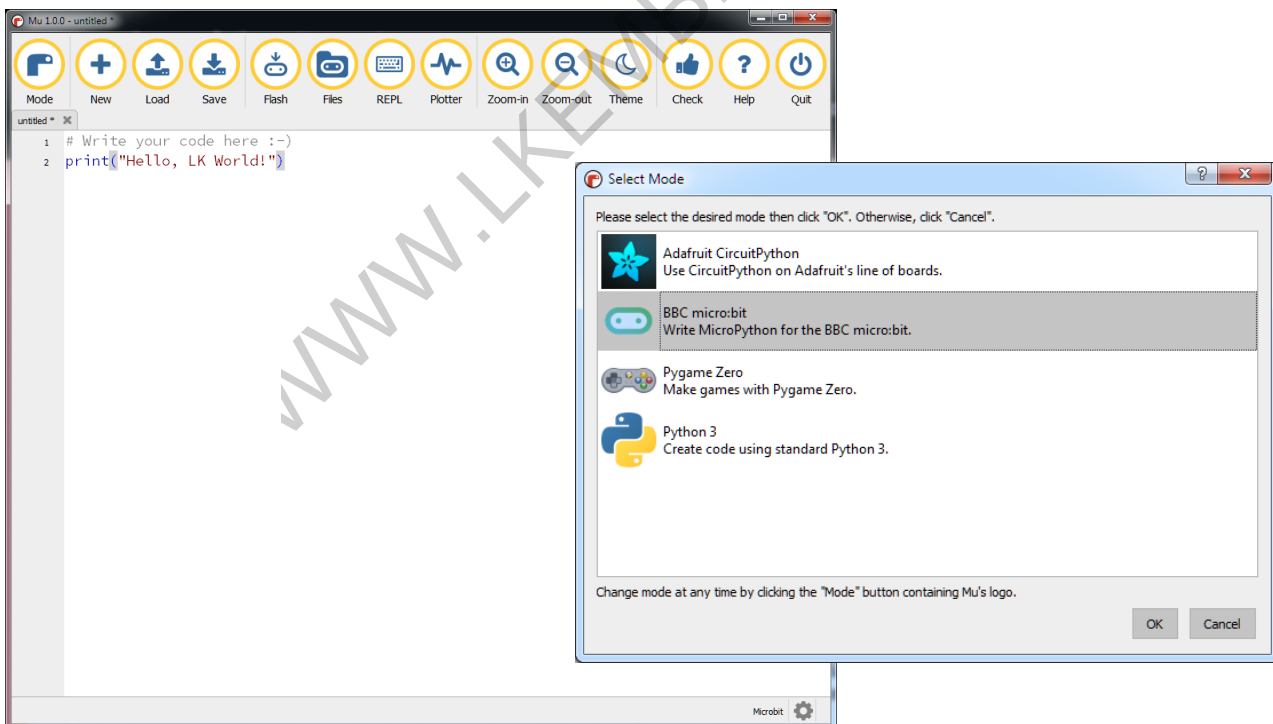
[<https://os.mbed.com/platforms/Microbit>] 주소에서 회원가입을 통하여 OS 사용이 가능하오니 참고하시길 바랍니다.



<그림 1-4-3> 마이크로 비트 mbed 웹페이지

✓ Mu 오프라인 컴파일러 - Mu offline Python editor : [<https://codewith.mu/>]

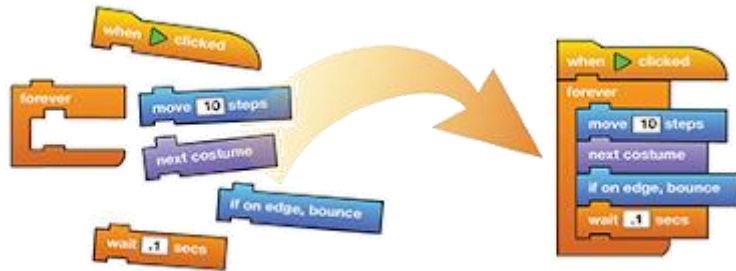
Mu 는 간단하게 설치하여 파이썬 프로그래밍을 사용할 수 있는 컴파일러 입니다. Off-line 환경에서 코딩을 진행할때 적합합니다.



<그림 1-4-4> 마이크로 비트 Mu 컴파일러 실행화면

1.4.2 블록 코딩이란?

블록을 원하는 순서대로 끼워 맞춰 코딩을 한다 하여, 블록 코딩이라는 명칭을 얻게 되었습니다. 프로그래밍 언어가 생소할 수 있는 어린이를 대상으로 교육이 진행되었으며, 호기심과 재미를 갖게 하는데 성공했습니다.



<그림 1-4-5> 블록 코딩

블록코딩의 장점으로서는 어떤 것이 있을까요?

첫번째로, 낮은 진입장벽 입니다. 블록코딩은 어렵게 키보드로 문자를 입력할 필요가 없습니다. 필요한 블록을 드래그 하여 원하는 위치에 붙여놓고 실행만 하면 됩니다. 이는 학생뿐만 아니라 교사의 입장에서든 쉽게 이해할 수 있어, 빠르게 배우고 학생들에게 교육을 해줄 수 있습니다.

두번째로, 프로그래밍의 개념을 쉽게 배울 수 있습니다. 명령어, 이벤트, 순차 처리 등 프로그래밍을 하는데 있어서 필요한 개념을 쉽게 정립할 수 있습니다. 이렇게 키운 논리와 개념으로 후에, 프로그램 언어를 습득하는데 큰 도움이 될 뿐만 아니라 쉽게 적응하여 활용할 수 있습니다.

세번째로, 실험정신 및 호기심을 촉진시킵니다. 블록 코딩은 쉽게 적용할 수 있어서, 간단하게 구성을 하고 결과를 바로 눈으로 확인할 수 있습니다. 이는 여러가지 실험을 할 수 있도록 하여, 호기심을 촉진시켜 점차 사용자가 원하는, 사용자가 상상해온 것을 이룰 수 있도록 합니다.

블록코딩으로 모든 것을 구현하는 데는 한계가 있습니다. 하지만 코딩의 개념 및 흥미를 입문자들도 쉽게 가지게 함으로써, 텍스트 기반의 코딩(C/C++, java 등)에 쉽게 다가갈 수 있는 역할을 합니다.

1.4.3 마이크로 비트 블록 코딩 인터페이스



<그림 1-4-6> 블록 코딩 인터페이스

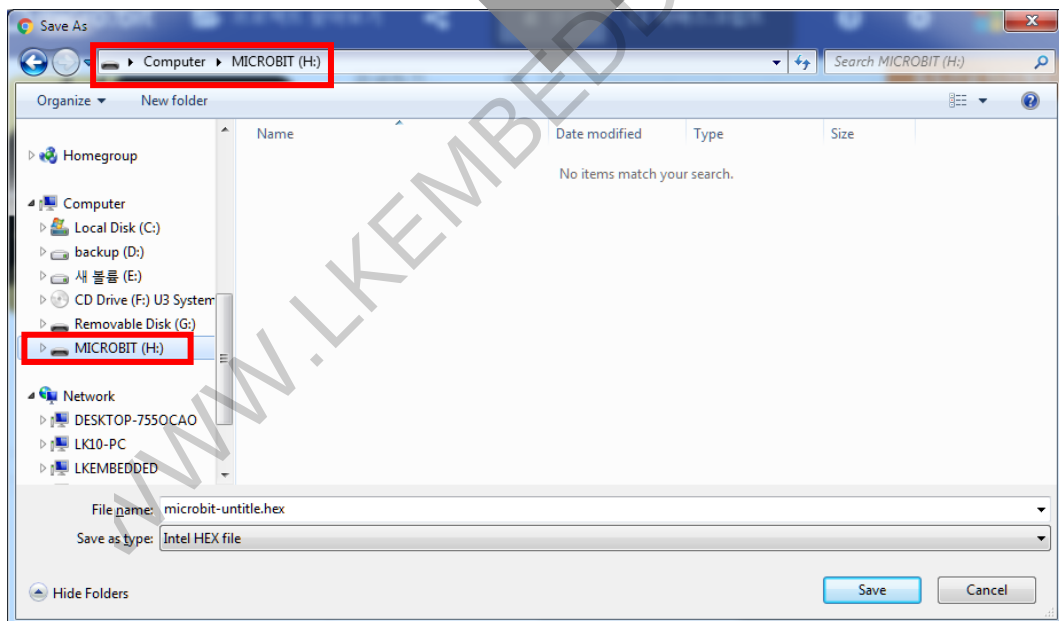
✓ 블록코딩 프로그래밍 방법(PC)

1) 생각한 동작을 블록 코딩으로 작성합니다.



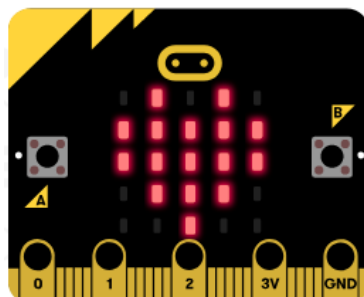
<그림 1-4-7> 블록코딩을 진행

2) 마이크로 비트를 USB 케이블을 이용하여 PC와 연결 후, 하단의 다운로드를 눌러 줍니다.
이때, 저장 위치는 마이크로 비트 저장소로 선택해줍니다.



<그림 1-4-8> 다운로드 진행

3) 다운로드가 완료된 후, 마이크로 비트의 동작을 확인합니다.



<그림 1-4-9> 마이크로 비트 동작 확인

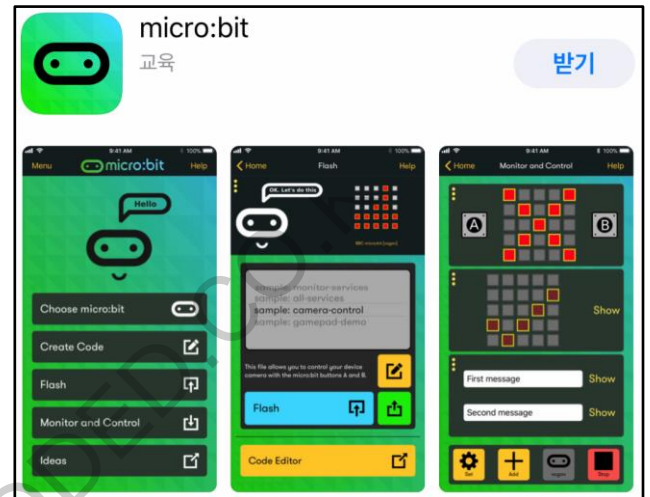
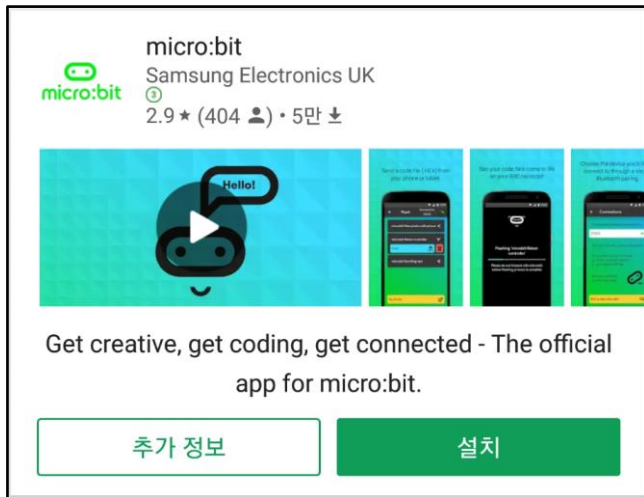
1.4.4 스마트폰을 이용한 마이크로 비트 코딩 방법

마이크로 비트의 큰 장점은 스마트폰이나 태블릿 PC 등을 이용하여 선 연결 없이도 블루투스를 통하여 코딩 및 프로그래밍이 가능하다는 장점이 있습니다.

1) 'Google Play' 또는 'App Store' 를 통하여 마이크로 비트 공식 어플을 다운로드&설치를 진행합니다.

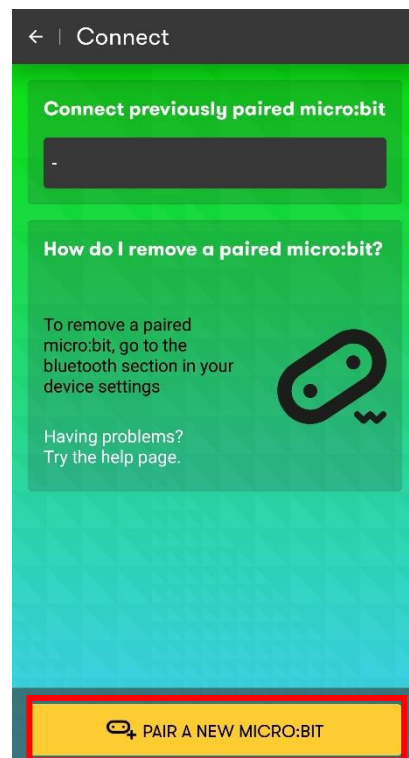
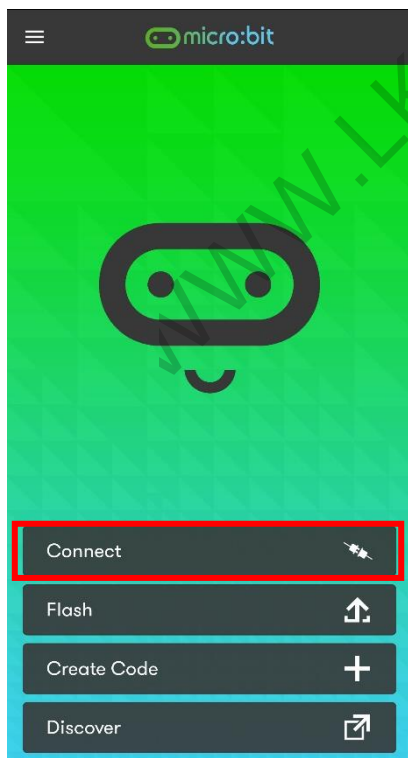


(* Android 버전 4.4 이상 , iOS 버전 8.2 이상)



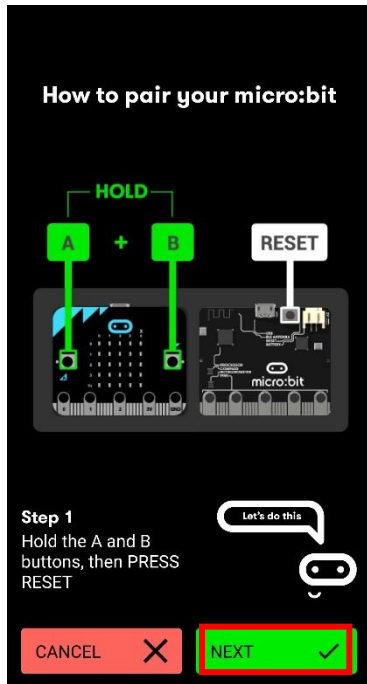
<그림 1-4-10> 'Google Play' 및 'App Store' 에 등록되어 있는 micro:bit 어플리케이션

2) 어플리케이션 실행 및 마이크로 비트 연결(Connect)



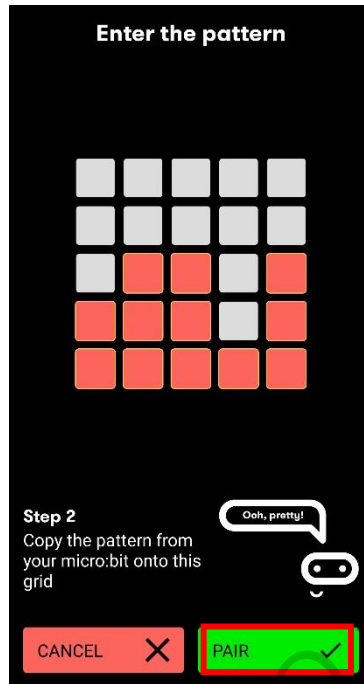
어플리케이션을 실행하면 위와 같은 창을 확인할 수 있습니다. 최초 실행 시, 'Connect'를 눌러 디바이스를 등록하도록 합니다.

'PAIR A NEW MICRO:BIT'를 선택해 줍니다. 스마트폰의 블루투스 기능이 비활성화되어 있다면 활성화 하도록 합니다.



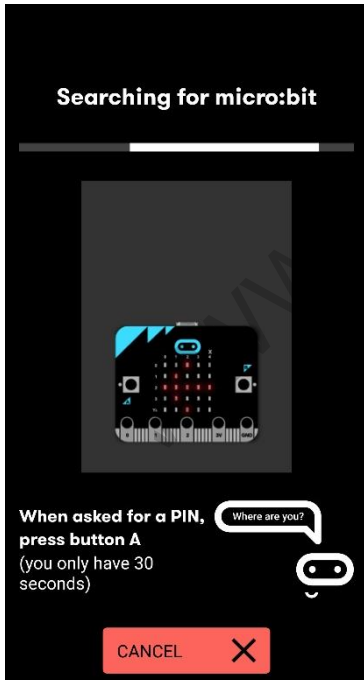
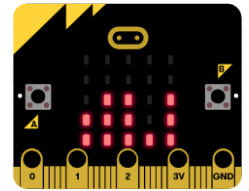
'페어링모드'에 진입하면 위와 같은 창이 뜹니다. 그림의 설명처럼 마이크로비트의 'A버튼과 B버튼을 누른 상태에서 리셋버튼을 눌렀다가 떼줍니다.(A, B 버튼은 유지). 몇 초가 지나면, LED에 'PAIRING MODE'라는 글자가 출력됩니다.

'NEXT'를 눌러 다음 페이지로 이동합니다.



패턴을 입력하는 창이 나오는데, 패턴은 마이크로 비트에서 확인할 수 있습니다. 페어링을 하려고 하는 마이크로 비트의 LED에 패턴의 모양이 표시가 되며, 해당 패턴을 확인 후, 동일하게 스마트폰에도 입력을 해주시면 됩니다.

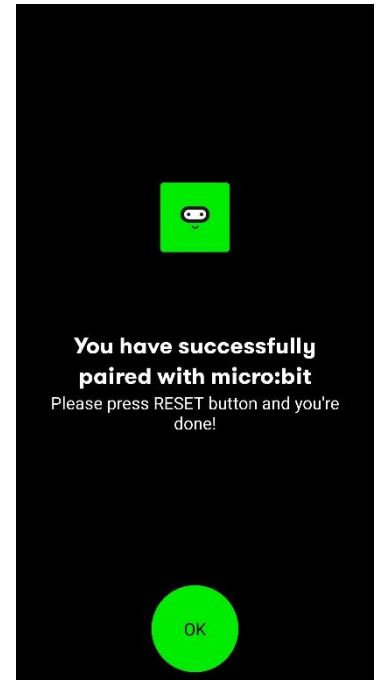
입력 후, 'PAIR'을 눌러 다음 단계로 이동합니다.



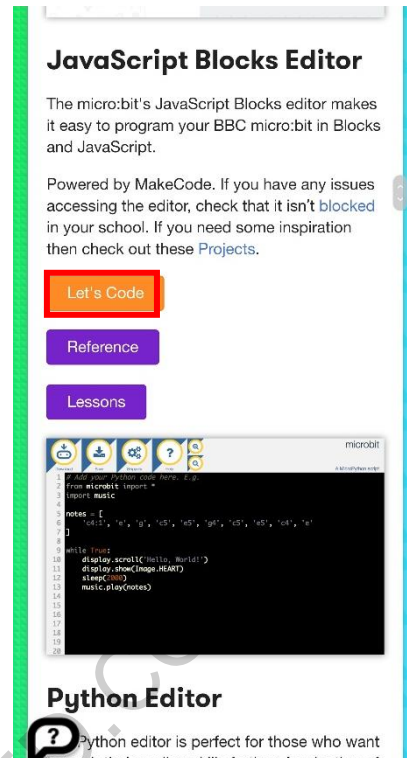
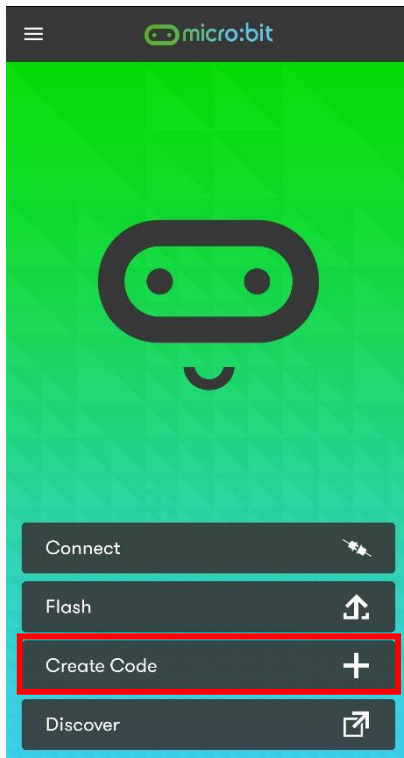
패턴까지 입력을 했다면, **마지막으로 핀(PIN)번호를 입력해야 합니다.** 위 그림과 같이 마이크로 비트에 'A' 버튼을 누르라는 화살표가 나옵니다. 'A' 버튼을 누르면 LED 화면에 숫자가 하나씩 표시가 됩니다. (총 6자리이며, 연속되는 숫자의 경우에 조금 더 긴 시간동안 표시되니 주의하세요!)



앞서서 확인했던 '핀(PIN)번호 6 자리'를 입력해준 뒤, 확인을 눌러줍니다. 정확하게 입력을 했다면 오른쪽 사진과 같이, 완료가 된 사진을 볼 수 있습니다.

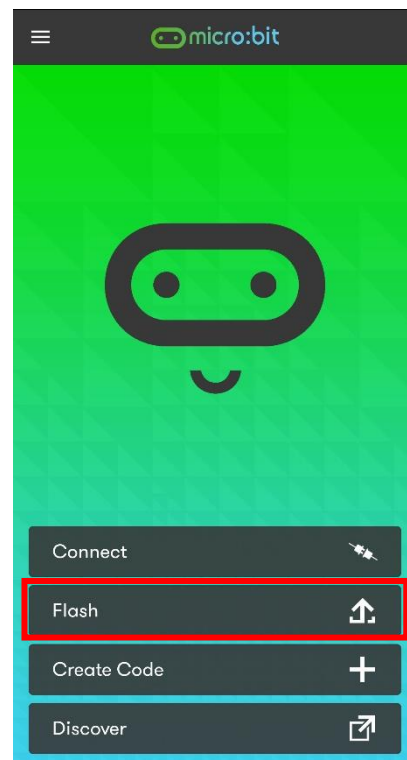
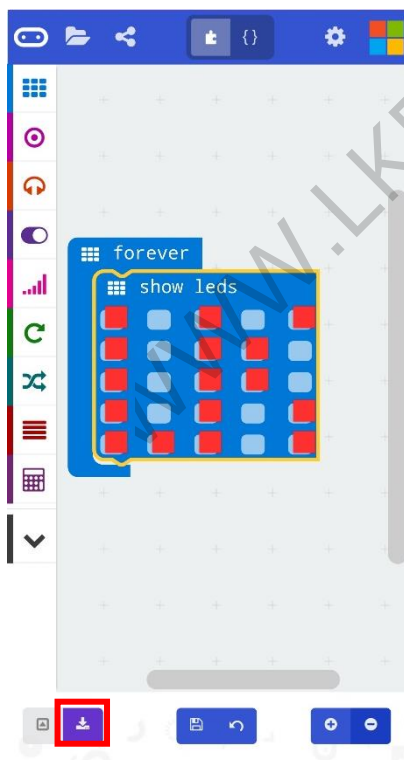


3) 스마트폰으로 코딩 후 프로그래밍



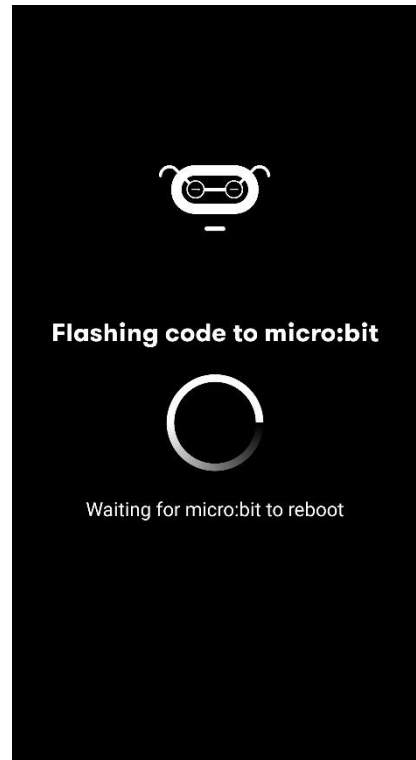
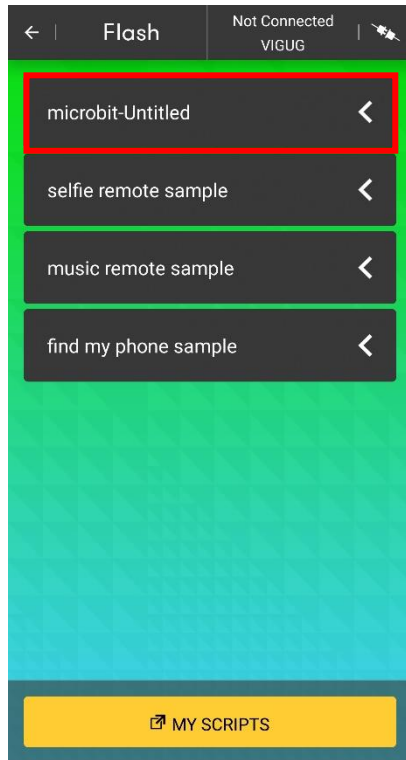
'Create Code'를 누르면, PC에서와 같이 코딩을 진행할 수 있는, 웹창이 팝업됩니다.

'Let's Code'를 눌러서, 코딩을 시작합니다.



코딩을 완료했다면, 하단의 다운로드 버튼을 눌러서, 파일을 다운로드 합니다. (간혹, 스마트폰에서 확장자를 자동으로 txt로 변경하는 경우가 있습니다. 꼭 확장자를 'hex'로 지정해 다운로드를 진행해 주세요. 예) text.hex)

다시 어플리케이션으로 돌아와 프로그래밍을 위해 'Flash'를 눌러 프로그래밍창으로 이동합니다.



앞서서 저장한 hex 파일의 이름이 상단에 표시가 되어
있습니다. (저장한 파일명을 확인해주세요.) 해당 파일을 누르면
Flash 를 여부를 물어본 뒤, 프로그래밍을 진행합니다.

<프로그래밍이 진행되는 모습>

📖 어플리케이션을 이용한 프로그래밍 참고동영상

✓ 안드로이드 (GooglePlay)

- <https://youtu.be/9Bmp91uT674>

✓ iOS (AppSotore)

- https://youtu.be/2uFuW8kA_lw

1.4.5 마이크로 비트 기본 블록 소개

마이크로 비트에서 사용되는 기본 블록에 대해서 알아보까요?

시작하면 실행

시작하면 실행 / [기본]

마이크로 비트에 전원이 투입되고, 최초 1 번만 실행되는 부분입니다.

예를 들어, 부저음을 울리는 블록을 넣고 프로그램을 하면 최초 한번만 부저음이 출력됩니다.

아두이노 스케치의 'setup()' 함수 역할을 하는 블록입니다.

무한 반복 실행

무한 반복 실행 / [기본]

반복해서 실행을 하는 블록입니다.

마이크로 비트는 이 블록안에 있는 코드에 따라서 반복하여 실행을 하게 됩니다.

아두이노 스케치의 'loop()' 함수 역할을 하는 블록입니다.

시작하면 실행

일시 중지(ms)

일시 중지(ms) / [기본]

직전까지의 코드를 수행하고, 입력한 시간만큼 일시 중지하는 블록입니다. 단위는 ms(10^{-3} seconds) 입니다. 즉, 100 을 입력하면 0.1 초만큼 정지했다가 다음 코드를 수행합니다.

아두이노 스케치의 'delay()' 함수 역할을 하는 블록입니다.

시작하면 실행

만약(if)

이면(then) 실행

아니면(else) 실행

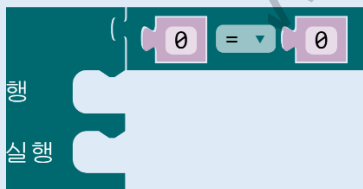
참(true)

만약(if) / [논리]

조건에 따라서 실행시킬 수 있는 조건문 블록입니다. 조건이 참이면, 하위 코드를 실행하고, 거짓이면 실행하지 않습니다.

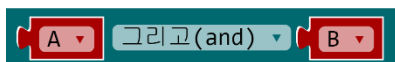
왼쪽 상단의 톱니바퀴()를 눌러서 'if', 'else if', 'else' 항목을 추가할 수 있습니다.

아두이노 스케치의 'if() {}', 'else if() {}', 'else {}' 조건문과 같은 역할을 합니다.



비교연산자 (=, ≠, <, ≤, >, ≥) / [논리]

변수 혹은 숫자를 비교하여, 참 또는 거짓으로 반환하는 블록입니다. 주로 조건문의 조건 부분에 부착하여 코딩을 합니다.



논리연산자 (그리고(and), 또는(or)) / [논리]

2 개의 조건을 확인해야할 때 사용하는 블록입니다.

그리고(and): A와 B 가 모두 참 이여야 '참'을 반환합니다.

또는(or): A와 B, 둘 중에 하나만 참이여도 '참'을 반환합니다

시작하면 실행

A 값 10 저장

변수 값 저장 / [변수]

변수에 값을 저장하는 블록입니다. 기준이 되는 값을 여러 블록에 적용시킬 때 유용하게 사용할 수 있습니다.

시작하면 실행

A 값 1 증가

변수 값 증가 / [변수]

변수에 값을 증가시키는 블록입니다. 입력한 값만큼 증가를 시키며, 음의 정수값도 입력 가능합니다.

시작하면 실행

item 값 랜덤 선택(정수): 0 ~ 4 저장

랜덤 선택(정수) / [계산]

랜덤한 값을 반환하는 블록입니다. 0 부터 입력한 숫자 중, 랜덤으로 하나의 정수를 선택하여 반환해 줍니다.

시작하면 실행

디지털 출력 P0 값 A

디지털 출력 / [고급] → [핀]

사용자가 원하는 핀에서 하이(HIGH, 1, 3.3V) 또는 로우(LOW, 0, 0V)를 출력합니다. 외부에 있는 센서나 부품들에 신호를 출력할 때 사용하는 블록입니다.

아두이노 스케치의 'digitalWrite(핀, 신호);' 함수와 같은 역할을 합니다.

시작하면 실행

A 값 디지털 입력 P0 저장

디지털 입력 / [고급] → [핀]

원하는 핀의 상태 값을 읽어올 수 있습니다. 하이(HIGH, 3.3V)이면 '1', 로우(LOW, 0V)이면 '0'을 반환합니다. 센서나 스위치같은 외부 입력 부품들로부터 디지털 신호를 입력받을 때 사용합니다. 아두이노 스케치의 'digitalRead(핀);' 함수와 같은 역할을 합니다.

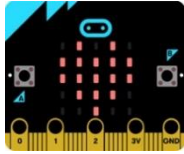
이 외에도 여러가지 블록이 있지만, 자주 사용되는 블록에 대해서 먼저 설명하였습니다. 이 외에 특정상황에서 사용되는 블록들은 예제를 진행하면서 알아보시다.

제2장 마이크로 비트 실습

2.1 LED 실습

✓ 목표: 마이크로 비트의 LED 를 제어를 이해한다.

2.1.1 준비물



마이크로 비트(Micro:bit) x 1



마이크로 USB 케이블 x 1

2.1.2 블록 이해하기

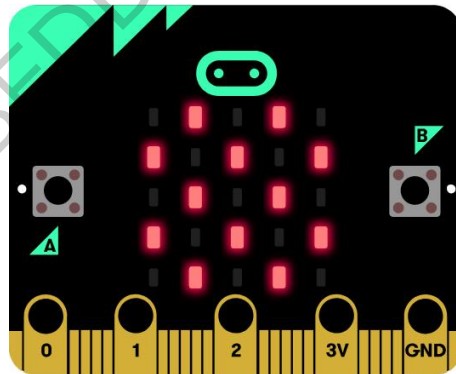
LED 를 제어하는 블록에 대해서 알아보시다.

✓ LED 출력 / [\[기본\]](#)

5x5 로 이루어진 LED 중, 원하는 위치에 LED 를 켜는 블록입니다.



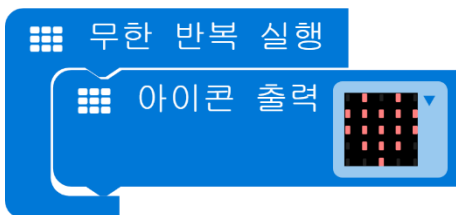
<블록코딩 컴파일러>



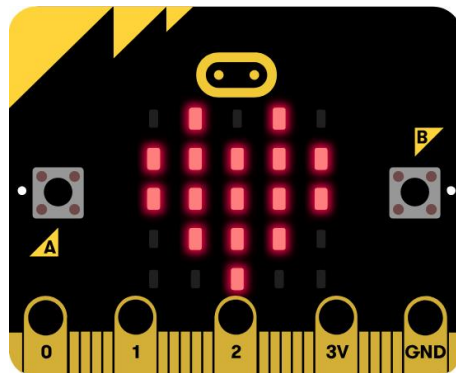
<마이크로비트 실행화면>

✓ 아이콘 출력 / [\[기본\]](#)

정해진 아이콘을 LED 에 출력하는 블록입니다. 총 40 개의 아이콘이 있습니다.



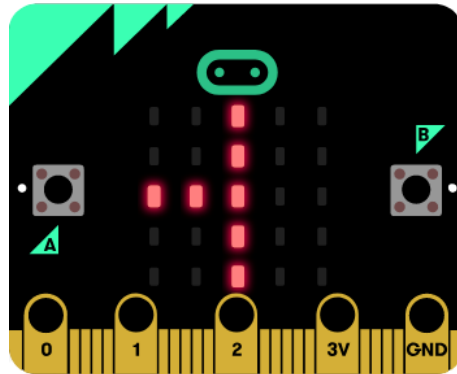
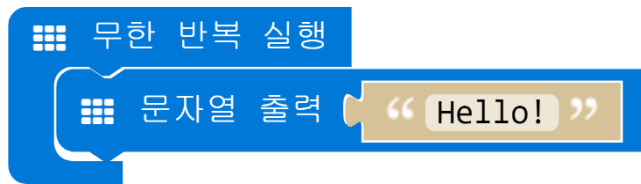
<블록코딩 컴파일러>



<마이크로비트 실행화면>

✓ 문자열 출력 / [기본]

입력한 문자를 출력하는 블록입니다. 알파벳, 숫자, 특수문자 등을 표현할 수 있습니다. 입력한 문자가 왼쪽으로 흘러가면서 표시됩니다.

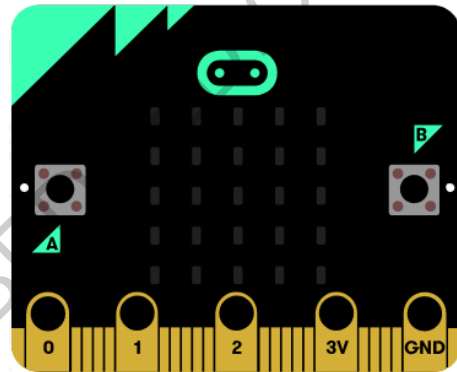
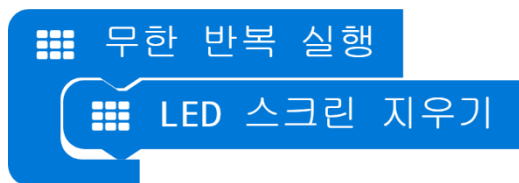


<블록코딩 컴파일러>

<마이크로비트 실행화면>

✓ LED 스크린 지우기 / [기본] → [더 보기]

모든 LED를 끕니다. 문자를 표현한 후, 초기화 시키고 싶을 때 사용할 수 있습니다.

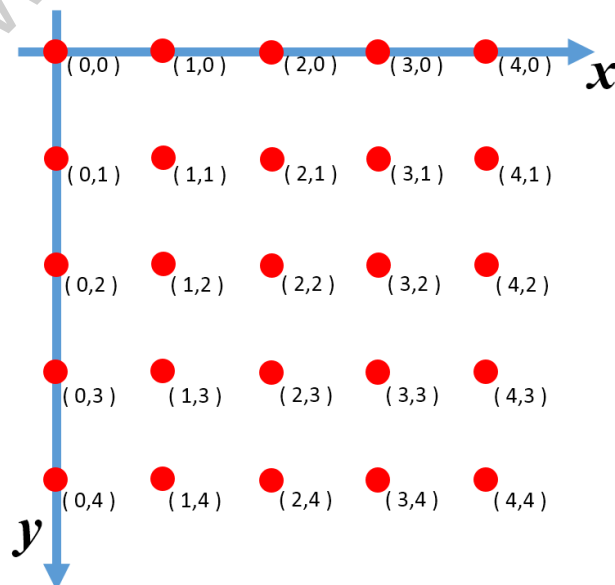


<블록코딩 컴파일러>

<마이크로비트 실행화면>

✓ LED 켜기 / [LED]

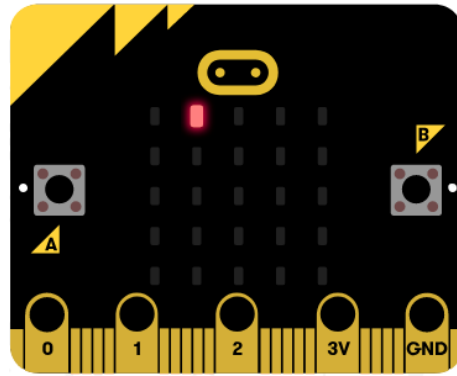
원하는 위치(좌표)의 LED를 켭니다.



<그림 2-1-1> 각 LED 좌표



<블록코딩 컴파일러>



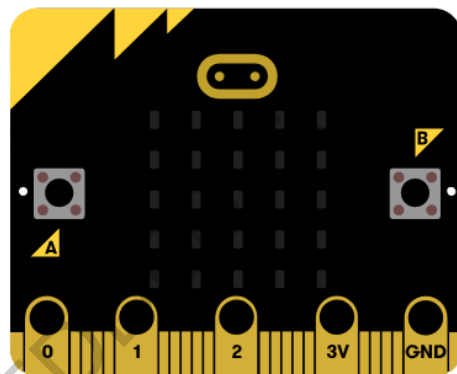
<마이크로비트 실행화면>

✓ LED 끄기 / [LED]

원하는 위치의 LED 를 끕니다.



<블록코딩 컴파일러>



<마이크로비트 실행화면>

2.1.3 만들어보기

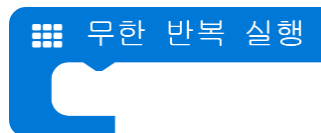
✓ 크기가 변하는 하트 만들기

목표: 점부터 시작해서 크기가 변하는 하트를 만들어 보시다.

원하는 좌표의 LED 를 켜고/끄는 블록을 사용하여도 되지만, 그렇게 되면 많은 블록이 사용되어야 합니다.

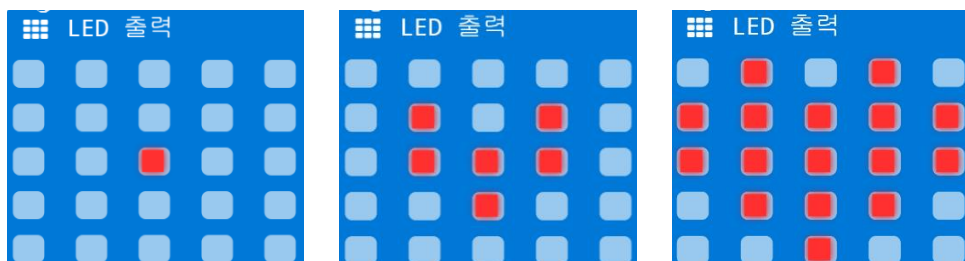
[LED 출력] 블록을 사용하여 만들어 보도록 합니다.

첫 번째, [무한 반복 실행] 블록으로 코딩을 시작합니다.



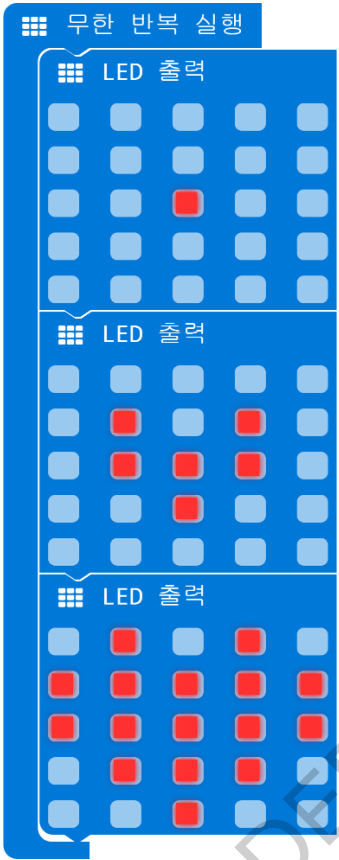
<그림 2-1-2> 무한 반복 실행 블록

두 번째, [LED 출력] 블록을 3 개 가져와 하트가 점점 커지는 3 개의 단계를 그림니다.



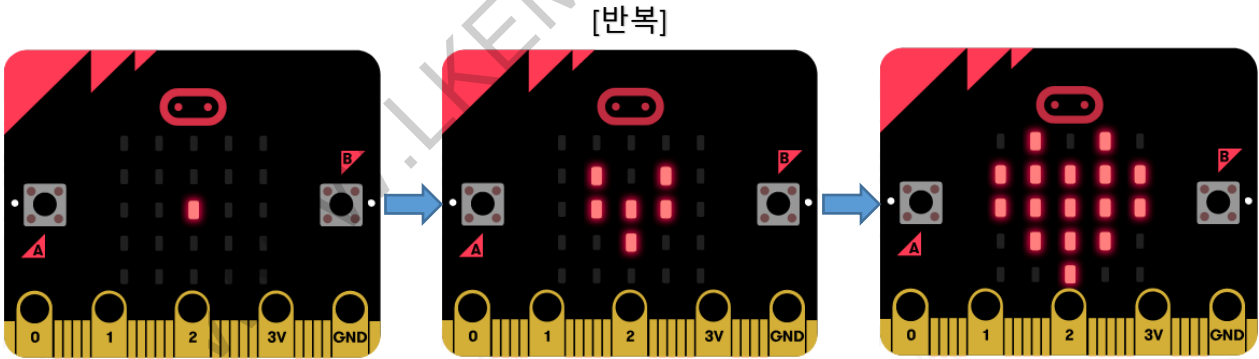
<그림 2-1-3> 점점 커지는 하트를 나타낸 블록

네 번째, 위에서 그린 블록들을 차례대로 붙여주어 완성합니다.



<그림 2-1-4> 블록 완성!

위 블록을 실행시키면, 아래와 같은 결과를 얻을 수 있습니다.

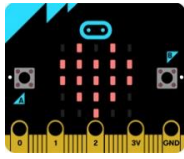


<그림 2-1-5> 실행 이미지

2.2 부저 실습

✓ 목표: 마이크로 비트를 이용해 부저(Buzzer)를 제어한다.

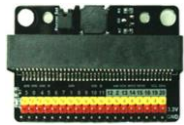
2.2.1 준비물



마이크로 비트(Micro:bit) x 1



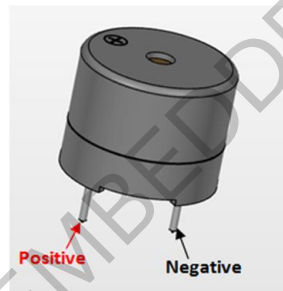
마이크로 USB 케이블 x 1



마이크로 비트 확장 보드 x 1

참고지식

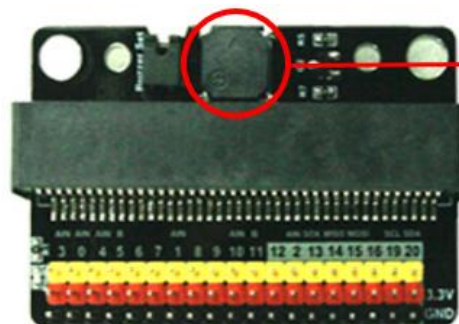
✓ 부저(Buzzer)



<그림 2-2-1> 부저(Buzzer)

부저는 전기적 신호를 소리 신호로 바꿔주는 대표적인 출력 부품입니다. 부저는 크게 능동부저(Active Buzzer)와 수동부저(Passive Buzzer)로 나뉩니다. 능동부저는 내부에 내장된 회로가 있어서 전원만 인가해 주면 소리가 출력됩니다. 수동부저는 능동부저와는 달리 내장된 회로가 없어서 주파수를 이용해 제어해야 합니다. 능동부저는 한가지 음만 낼 수 있는 반면에 수동부저는 주파수를 제어하여 여러가지 음을 출력할 수 있습니다. 이를 이용해 멜로디 출력, 전자피아노 등 다양한 실습을 할 수 있습니다.

마이크로 비트 확장 보드에는 위와 같은 부저가 내장되어 있기 때문에, 별도의 연결없이 부저를 이용할 수 있습니다.



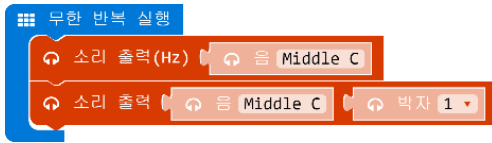
내장되어 있는
부저 (Buzzer)

<그림 2-2-2> 마이크로 비트 확장보드 내부에는 부저가 내장되어 있다.

2.2.2 블록 이해하기

부저를 제어하는 블록에 대해서 알아보시다.

✓ 소리 출력 / [소리]



[소리출력] 블록에는 2 가지 종류가 있습니다.

두 가지 블록은 '음'을 선택하여 출력하고, 한 개의 블록은 추가로 '박자'도 선택하여 출력합니다. 선택한 '음'에 알맞은 주파수를 출력하고, 박자는 해당 주파수를 유지하는 시간을 의미합니다.

✓ 지속 시간 / [소리]



[지속 시간(ms)] 블록은 앞선 챕터에서 설명하였던,

[일시중지(ms)] 블록과 같은 역할을 합니다. 설정한 박자동안은 동작하지 않습니다.

블록에 ms 라는 표기가 되어 있지만, 입력한 bpm 에 따른 박자동안 정지하게 됩니다. 시간이 지난 뒤에는 다음 동작을 계속 합니다.

※ bpm = beats per minutes 의 약자로 '분당 비트 수'를 의미한다.

✓ 소리 출력 (멜로디) / [소리]



[소리 출력 (멜로디)] 블록은 입력되어 있는 멜로디

20 종류 중, 하나를 선택하여 출력하는 기능을 합니다. 설정된 bpm 수치의 영향을 받습니다. 한 번 출력, 반복 출력, 백그라운드 한 번 출력, 백그라운드 반복 출력 중에 선택할 수 있습니다.

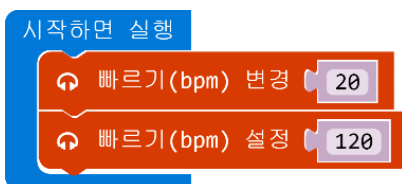
✓ 소리 감지하면 실행 (멜로디) / [소리]



[소리 감지하면 실행 (멜로디)] 블록은 멜로디 출력 상황에 따라서 함수를 실행할 수 있는 블록입니다.

멜로디를 실행시켰을 때, '멜로디 시작'으로 설정하였다면 이 블록 안에 있는 블록들이 실행됩니다.

✓ 빠르기(bpm) 변경, 설정 / [소리]



[빠르기(bpm) 변경 / 설정] 블록은 소리 속도를 설정 및 변경해주는 역할을 합니다. 숫자가 높을수록 속도가 빠르며, 10 ~ 400 까지 설정 가능합니다.

예) 60 bpm 일 경우 → 1 분에 60 박자 → 1 박자 = 1 초

2.2.3 만들어보기

✓ 동요 연주해보기

악보를 보고 동요를 마이크로 비트로 연주해봅니다.

학교종

· 음 표 ·

그림	이름	박자	리듬표시
	온음표	4박자	VVV
	점2분음표	3박자	VVV
	2분음표	2박자	VV
	점4분음표	1박자 반	V\
	4분음표	1박자	V
	점8분음표	반박자 반	V
	8분음표	반박자	\
	16분음표	반의 반박자	\

©EnCyber.com

<그림 2-2-3> 학교종 동요 악보 및 음표 설명

첫 번째, 빠르기(bpm)을 설정해 줍니다. [소리] 카테고리에서 [빠르기(bpm) 설정] 블록을 가지고 옵니다. 이 예제에서는 120 bpm 으로 설정하겠습니다.

시작하면 실행

빠르기(bpm) 설정 120

<그림 2-2-4> 빠르기 bpm 설정

두 번째, 악보에 작성된, 게이름과 박자를 보고 [소리 출력] 블록을 이용하여 코딩을 해줍니다. 반복해서 출력을 할 것이기 때문에, [무한 반복 실행] 안에 작성을 해줍니다.

*3 박자 설정이 없어 2 박자로 대체 하였습니다.

<그림 2-2-5> 게이름, 박자에 맞춰 작성 (학교종이 땡땡땡)

세 번째, 악보에 맞춰서 모두 작성해줍니다. 노래를 반복할 것이기 때문에 마지막에 [지속시간] 블록을 넣어서 곡과 곡사이에 간격을 두도록 합니다.

시작하면 실행

빠르기(bpm) 설정 120

무한 반복 실행

소리 출력	음 Middle G	박자 1
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle A	박자 1
소리 출력	음 Middle A	박자 1
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle E	박자 2
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle E	박자 1
소리 출력	음 Middle E	박자 1
소리 출력	음 Middle D	박자 2

소리 출력	음 Middle G	박자 1
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle A	박자 1
소리 출력	음 Middle A	박자 1
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle E	박자 2
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle G	박자 1
소리 출력	음 Middle D	박자 1
소리 출력	음 Middle E	박자 1
소리 출력	음 Middle C	박자 2
지속 시간(ms)	박자 1	
지속 시간(ms)	박자 4	

<그림 2-2-6> 완성된 블록

네 번째, 마이크로비트를 마이크로 비트 확장 보드에 꽂은 후 입력한 동요가 잘 나오는지 확인합니다.

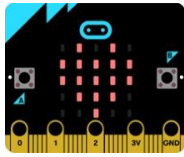


<그림 2-2-7> 마이크로 비트와 확장 보드가 연결된 사진

2.3 버튼(스위치) 실습

✓ 목표: 2 개의 버튼을 이용하여 LED와 부저를 제어한다.

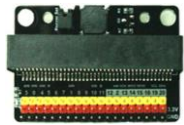
2.3.1 준비물



마이크로 비트(Micro:bit) x 1



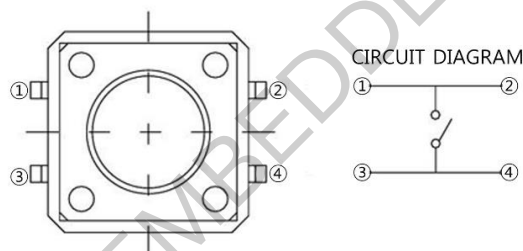
마이크로 USB 케이블 x 1



마이크로 비트 확장 보드 x 1

참고지식

✓택트스위치(Tact-Switch)



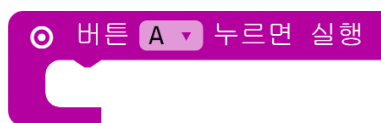
<그림 2-3-1> 택트스위치(Tact-Switch)

택트스위치(Tact-Switch)는 각종 전자제품에 많이 사용되어 일상생활 속에서도 흔히 볼 수 있는 부품입니다. 택트스위치의 1 번, 2 번라인과 3 번, 4 번라인이 평상시에는 연결이 되지 않은 상태로 있다가, 버튼을 누르는 동안은 2 개의 라인이 연결이 되면서 전기적으로 도통(연결)이 됩니다. 이러한 특성을 이용하여 버튼을 누를 때, 프로세서에 입력신호를 주는 입력 부품의 역할을 할 수 있습니다. 1 번, 2 번이 연결된 라인과 3 번, 4 번이 연결된 라인끼리는 항상 연결되어 있으니 회로를 구성할 때 유의해야 합니다.

2.3.2 블록 이해하기

버튼을 이용한 블록에 대해서 알아보시다..

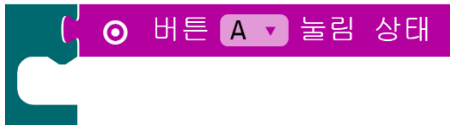
✓ 버튼 누르면 실행



[버튼 누르면 실행] 블록은 버튼을 눌렀을 때, 안에 있는 블록코드를 실행하는 블록입니다.

A 버튼, B 버튼, A+B 버튼 총 세가지 경우를 선택할 수 있습니다.

✓ 버튼 눌림 상태



[버튼 눌림 상태] 블록은 조건문에서 조건으로 사용될 수 있는 블록입니다.

A 버튼, B 버튼, A+B 버튼 총 세가지 경우를 선택할 수 있습니다.

2.3.3 만들어보기

✓ 버튼에 따른 LED, 부저 출력

A 또는 B 버튼을 눌렀을 때 LED를 출력하고, A와 B 버튼을 동시에 눌렀을 때 멜로디가 출력하기로 합니다.

첫 번째, **[입력]**의 **[버튼을 누르면 실행]** 블록 3개를 가지고 와서 각 상황을 설정합니다.



<그림 2-3-2> 각 경우의 블록들

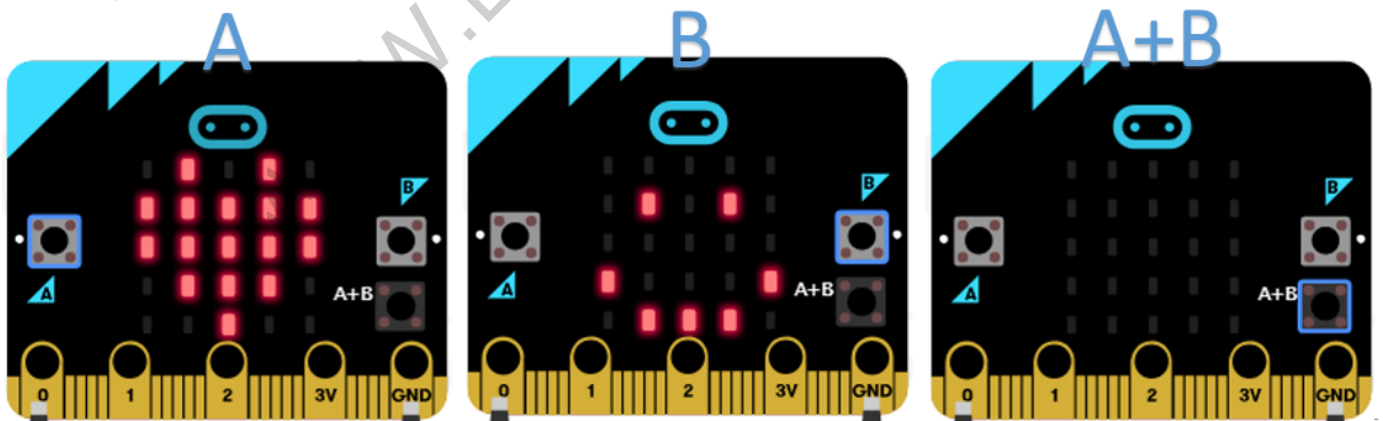
두 번째, 각 버튼을 눌렀을 때 어떤 동작을 할 것인지 생각해보도록 합니다. 앞서서 배운 동요 연주를 넣어도 좋고, 커지는 하트 예제를 넣어도 좋습니다. 필자는 설명의 편리함을 위하여 간단한 블록을 이용하여 코드를 작성하겠습니다.

A: 하트 아이콘 출력 / **B:** 스마일 아이콘 출력 / **A+B:** 화면 지우기, 멜로디 한 번 출력



<그림 2-3-3> 각 버튼에 따른 동작

세 번째, 각 버튼을 눌러서 정상동작하는지 확인합니다.

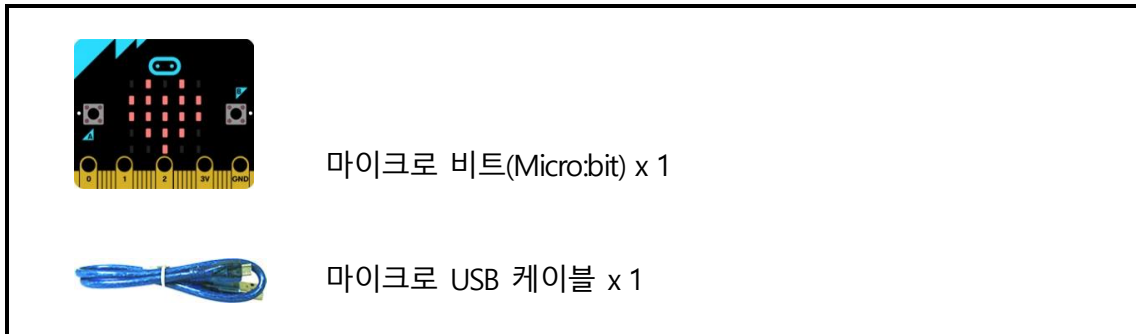


<그림 2-3-4> 각 버튼에 따른 LED 표시 화면

2.4 가속도 센서 실습

✓ 목표: 마이크로 비트를 가속도 센서를 학습한다.

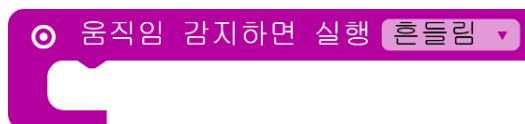
2.4.1 준비물



2.4.2 블록 이해하기

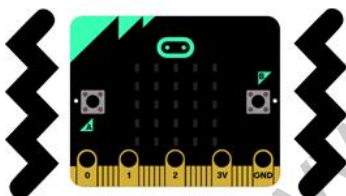
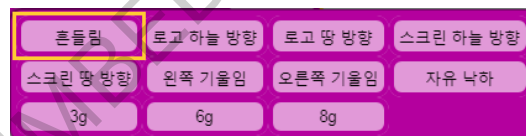
가속도 센서를 이용한 블록에 대해서 알아보시다.

✓ 움직임 감지하면 실행

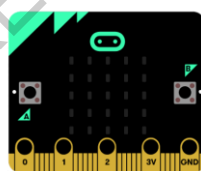


[움직임 감지하면 실행] 블록은 마이크로 비트의 움직임 또는 움직임 방향, 속도 등을 측정하여, 해당 조건에 충족되면 하위 코드를 실행시키는 블록입니다.

총 11 가지의 경우를 선택할 수 있습니다.

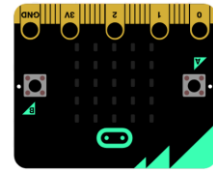


흔들림을 감지



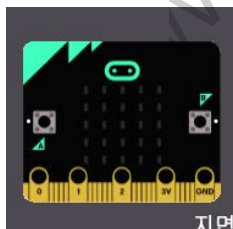
지면

로고 하늘 방향일 때 감지



지면

로고 땅 방향일 때 감지



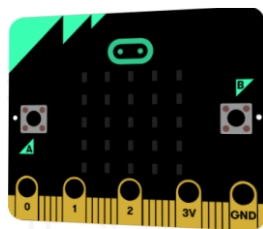
지면

LED 스크린이 하늘 방향일 때 감지

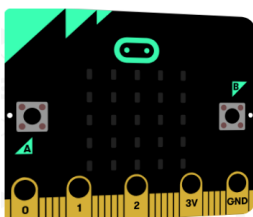


지면

LED 스크린이 지면 방향일 때 감지



왼쪽 기울어졌을 때 감지



오른쪽 기울어졌을 때 감지

자유 낙하: 중력가속도의 속도이상의 움직임을 감지

3g: 중력가속도 x 3 배의 속도이상의 움직임을 감지

6g: 중력가속도 x 6 배의 속도이상의 움직임을 감지

8g: 중력가속도 x 8 배의 속도이상의 움직임을 감지

✓ 가속도 센서 가속도(mg)

행

가속도 센서 가속도(mg)

x축

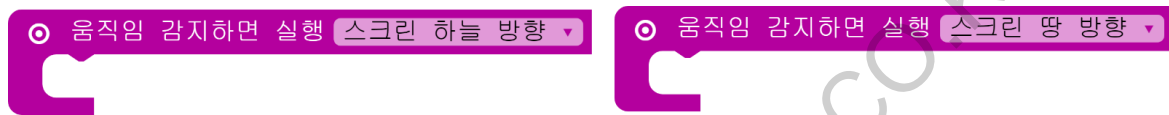
[가속도 센서 가속도(mg)] 블록은 각 축에 대한 가속도 값을 반환하는 블록입니다. 단위는 $mg = g \times 10^{-3} \cong 9.8 \text{ mm/s}^2$ 입니다. - 1023 ~ 1024 까지의 값으로 표현됩니다.
X 축, Y 축, Z 축, 크기(=vector) 선택이 가능합니다.
주로 각 축에 대한 빠르기를 측정하고자 할 때, 사용됩니다.

2.4.3 만들어보기

✓ 마이크로 비트와 가위바위보 하기

마이크로 비트를 뒤집었다가 화면을 위로 했을 때, 가위, 바위, 보 중 하나가 랜덤으로 출력이 되는 코드를 만들어 봅시다.

첫 번째, [움직임을 감지하면 실행] 블록을 가져와 각 상황별 함수를 만듭니다.

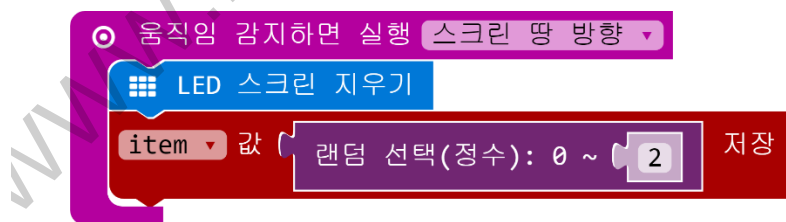


<그림 2-4-1> 각 동작을 하게될 실행 블록

두 번째, 각 동작에 대한 동작을 구현합니다.

1. 스크린이 땅 방향일 때, 화면을 초기화하고, 0 ~ 2 까지 숫자 중 랜덤으로 하나를 뽑는다.
2. 스크린이 하늘 방향일 때, 랜덤으로 뽑은 숫자대로 스크린(LED)을 출력한다.

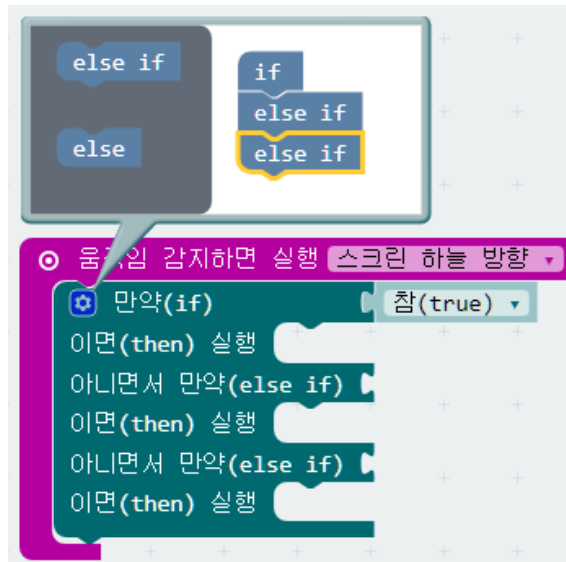
화면 초기화를 위해서 [기본] 카테고리 [더 보기]를 탭에서 [LED 스크린 지우기] 블록을 가지고 옵니다. 그리고 랜덤한 수를 뽑기 위하여, [계산] 카테고리에서 [랜덤 선택(정수)] 블록을 가지고 옵니다. 우리는 총 '가위', '바위', '보' 세 가지의 경우가 필요하니, 0 ~ 2 (0, 1, 2) 세 개의 숫자중 랜덤으로 하나를 가져오도록 설정합니다.



<그림 2-4-2> 스크린이 땅을 향했을 때, 블록 모음 동작

세 번째, 화면을 다시 위로 향했을 때, LED를 통해, 가위바위보를 표현할 수 있도록 합니다.

세가지 사항에 대해서 각각 다른 화면을 출력해야 하므로, [논리] 카테고리에서 [만약(if)] 블록을 가지고 옵니다. 처음에 가지고 오면 조건 블록을 한 가지 조건만 적용이 가능하므로, [설정]을 통하여 들어 줘야 합니다.



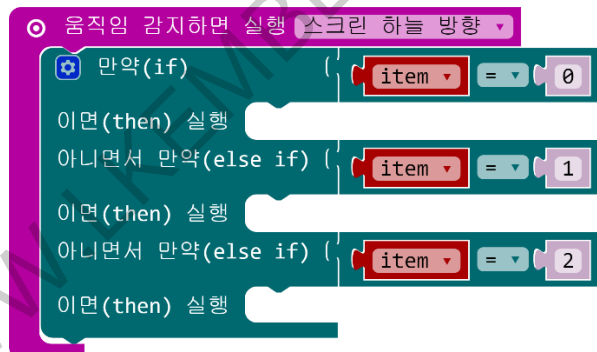
<그림 2-4-3> [만약(if)] 블록 설정 화면

앞서서 스크린이 땅을 향했을 때, 랜덤 정수를 'item'이라는 변수에 저장을 하였기 때문에, **[변수]** 카테고리에서 'item' 변수 블록을 가지고 옵니다. 또한 'item'에 저장된 값을 확인을 할 수 있는 비교연산자가 필요합니다. **[논리]** 카테고리에서 '비교연산자' 블록을 가지고 옵니다.



<그림 2-4-4> 비교연산자와 변수형 블록

변수가 '0 과 같을 때', '1 과 같을 때', '2 와 같을 때' 총 세가지 상황을 만들어 줍니다.

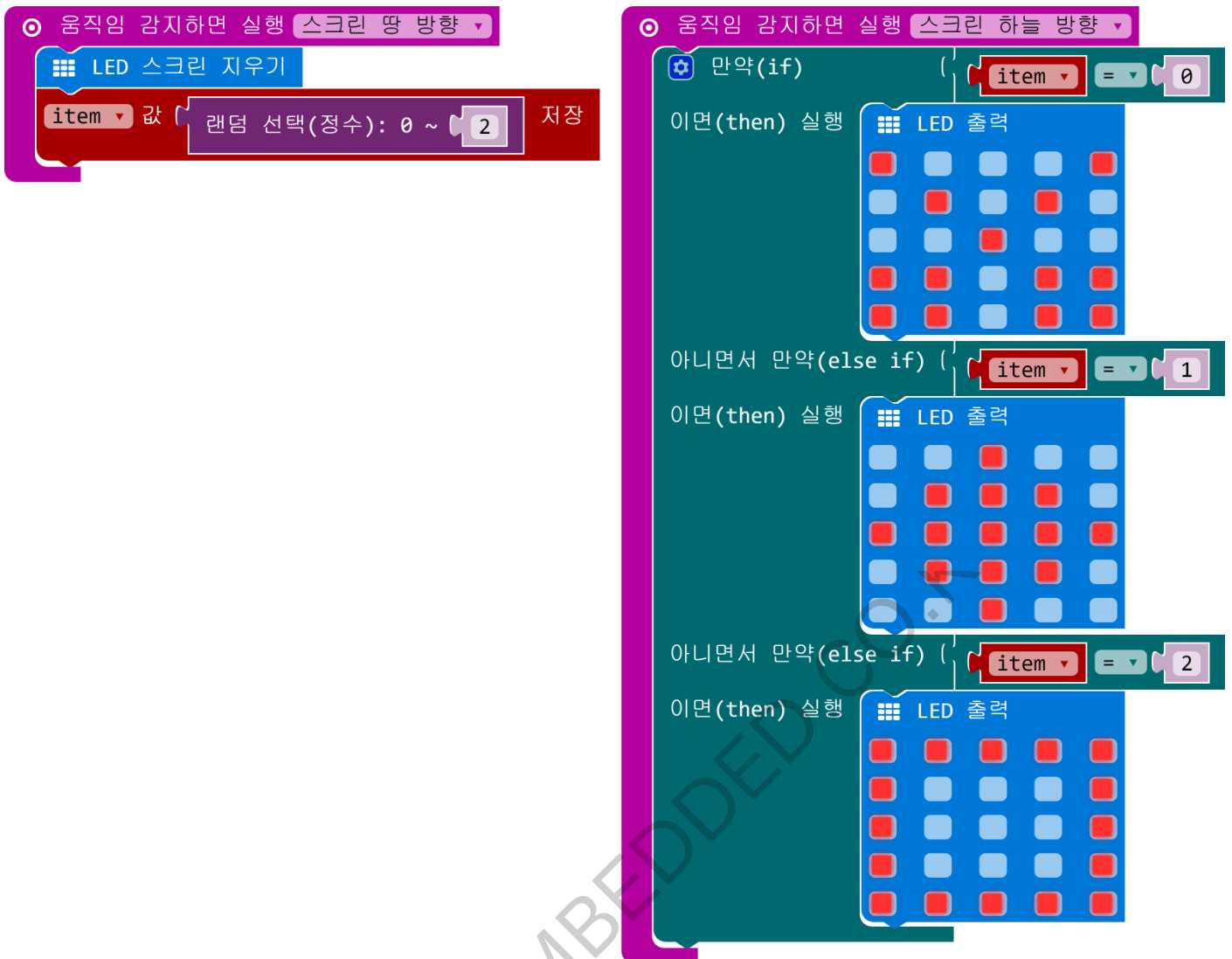


<그림 2-4-5> 세 가지 상황을 만들어준 블록 모습

각 상황에 맞는 [LED 출력]을 가지고 와, 블록작성을 완성해 줍니다. 본 예제에서는 가위, 바위, 보를 아래의 블록과 같이 표현해봤습니다.



<그림 2-4-6> 가위, 바위, 보 LED 출력



<그림 2-4-7> 완성된 블록

위 블록의 동작을 글로 설명하자면, 아래와 같습니다.

!이벤트 – 마이크로 비트의 LED 스크린 부분이 땅을 향했을 때,

- 1) LED 스크린 지우기: 현재 LED 스크린을 모두 OFF 합니다.
- 2) 변수 'item' 에 무작위 수 0 ~ 2 중, 하나를 저장합니다.

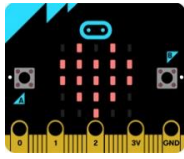
!이벤트 – 마이크로 비트의 LED 스크린 부분이 하늘을 향했을 때,

- 1) 변수 'item'에 저장되어 있는 값이 0 이면 '가위' 모양의 LED 출력
 변수 'item'에 저장되어 있는 값이 1 이면 '주먹' 모양의 LED 출력
 변수 'item'에 저장되어 있는 값이, 0 그리고 1 둘다 아니면 '보' 모양의 LED 출력

2.5 나침반 실습

✓ 목표: 마이크로 비트를 이용하여 나침반 센서에 대해서 학습한다.

2.5.1 준비물



마이크로 비트(Micro:bit) x 1



마이크로 USB 케이블 x 1

2.5.2 블록 이해하기

나침반센서를 이용한 블록에 대해서 알아봅시다..

✓ 자기(나침반) 센서 각도

실행

자기(나침반) 센서 각도(°)

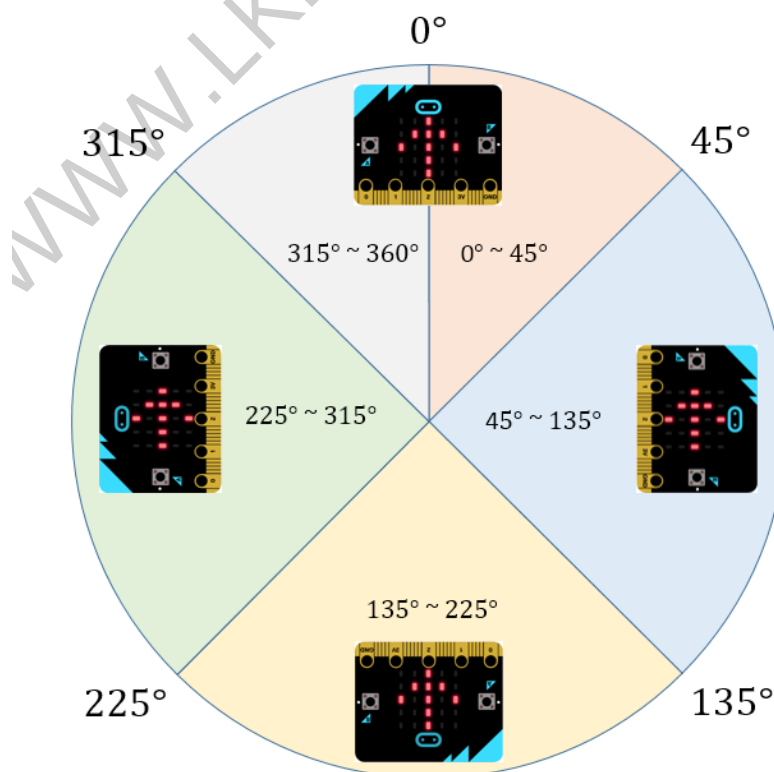
[자기(나침반) 센서 각도] 블록은 지구의 자기를 기준으로 현재 마이크로 비트가 가르키는 방향과의 각도를 숫자로 반환해주는 블록입니다. 0 ~ 360 도 까지의 값을 반환합니다..

2.5.3 만들어보기

✓ 나침반 만들기

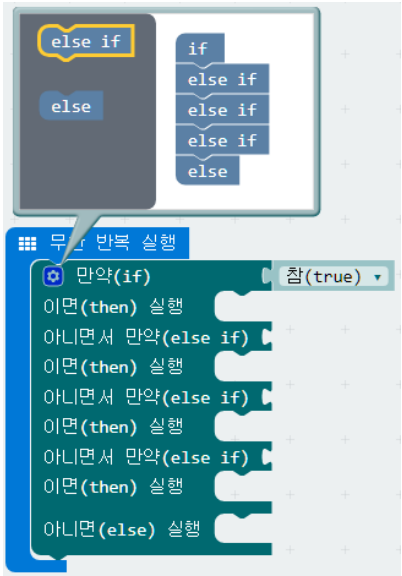
나침반처럼 방향을 바꿔도 계속해서 북쪽을 가르키는 마이크로 비트를 만들어 봅시다.

첫 번째, 먼저 어떤 상황에서 화살표를 어떻게 표현할 것인지 알아보도록 합니다.



<그림 2-5-1> 항상 북쪽을 가르키는 각 위치에 따른 화살표 방향

위 그림은 각 각도에서의 화살표 방향을 정리해둔 그림입니다. 총 5 가지의 경우에 따라서 화살표를 표현해주어야 합니다. **[논리]** 카테고리에서 **[만약(if)]** 블록을 가져와 총 5 가지 상황을 입력할 수 있도록 설정해 줍니다.



<그림 2-5-2> [만약(if)] 블록을 설정한 모습

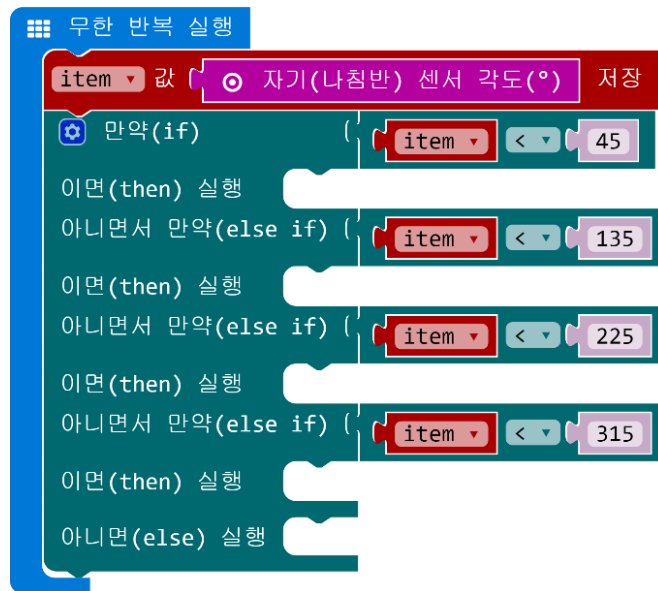
두 번째, **[변수]** 카테고리에서 '변수값 저장' 블록과 **[입력]** 카테고리에서 **[자기(나침반) 센서 각도]** 블록을 가지고 와서 각도 값을 변수에 저장해줍니다.



<그림 2-5-3> 나침반의 값을 변수에 저장한 모습

세번 째, 위에서 제시하였던 그림을 확인 후 비교연산자 블록을 이용하여 각 상황별로 조건을 만들어 줍니다.

- 0° ~ 45°: 북쪽
- 360° ~ 135°: 서쪽
- 136° ~ 225°: 남쪽
- 226° ~ 315°: 동쪽
- 나머지: 북쪽



<그림 2-5-4> 각 상황별로 설정을 해둔 모습

[기본] 카테고리의 [더 보기]에서 [화살표 출력] 블록을 확인하실 수 있습니다. 각 상황에 알맞은 화살표를 출력할 수 있게 설정하여 줍니다.

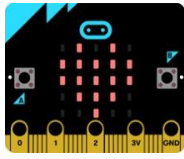


<그림 2-5-5> 완성!

2.6 외부 LED 실습

✓ 목표: 마이크로 비트를 이용하여 외부 LED 제어 방법에 대해서 학습한다.

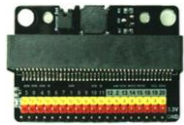
2.6.1 준비물



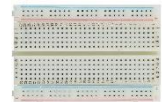
마이크로 비트(Micro:bit) x 1



마이크로 USB 케이블 x 1



마이크로 비트 확장 보드 x 1



400P 브레드보드 x 1



LED(색상무관) x 4



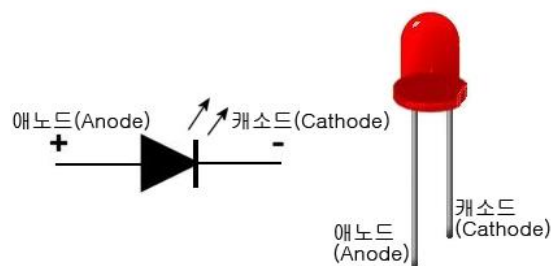
저항 220 옴 x 4



점퍼 케이블(암수) x 6

참고지식

✓ LED(Light Emitting Diode)

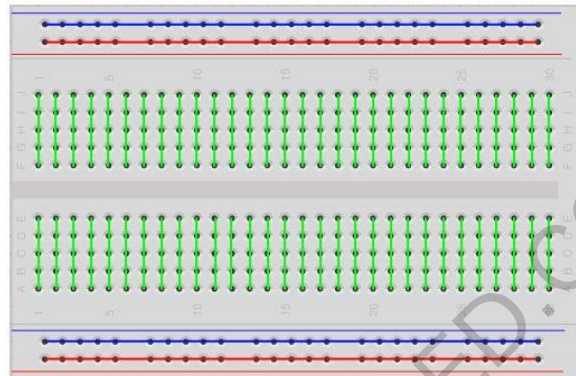


<그림 2-6-1> LED(Light Emitting Diode)

LED(Light Emitting Diode)는 발광 다이오드라고도 합니다. 전류가 흐르면 빛을 방출하는 다이오드로 전기에너지를 빛 에너지로 바꾸어 주는 대표적인 반도체 소자입니다. LED는 허용치보다 높은 전류가 흐르면 쉽게 손상될 수 있어 이를 막기 위해 보호저항을 사용합니다. 본 실습에서는 220Ω을 사용합니다. 또한 LED는 극성이 있어, 연결 시 주의해야 하며 핀이 긴 쪽이 애노드(Anode, +극)이며, 짧은 쪽이 캐소드(Cathode, -극)입니다.

✓ 브레드보드(Breadboard)

아래의 그림을 보면 브레드보드의 배선도를 확인할 수 있습니다. 브레드보드는 핀(Pin)을 꽂는 내부가 서로 연결되어 있어 부품과 점퍼 케이블을 꽂아서 회로를 쉽게 구성할 수 있게 도와주는 부품입니다. 브레드보드가 없이 회로를 구성하면 납땜을 해야 하는 불편함이 있습니다.

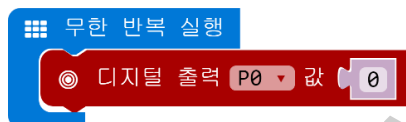


<그림 2-6-2> 브레드보드 배선도

2.6.2 블록 이해하기

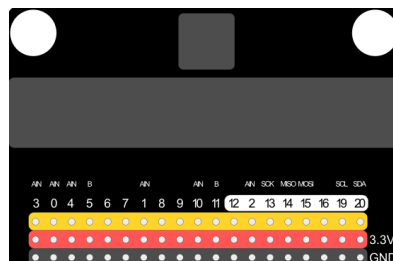
외부 LED를 제어할 수 있는 핀 제어 블록에 대해서 알아보시다.

✓ 디지털 출력



[디지털 출력] 블록은 원하는 핀에서 원하는 신호를 내보내주는 블록입니다. 총 19개의 핀을 선택할 수 있으며, 값은 '0'을 선택하면 해당 핀에서 LOW(0V)가 출력되며, '1'을 선택하면 HIGH(3.3V)가 출력됩니다.

핀 번호는 마이크로 비트 확장 보드에서 확인할 수 있습니다.

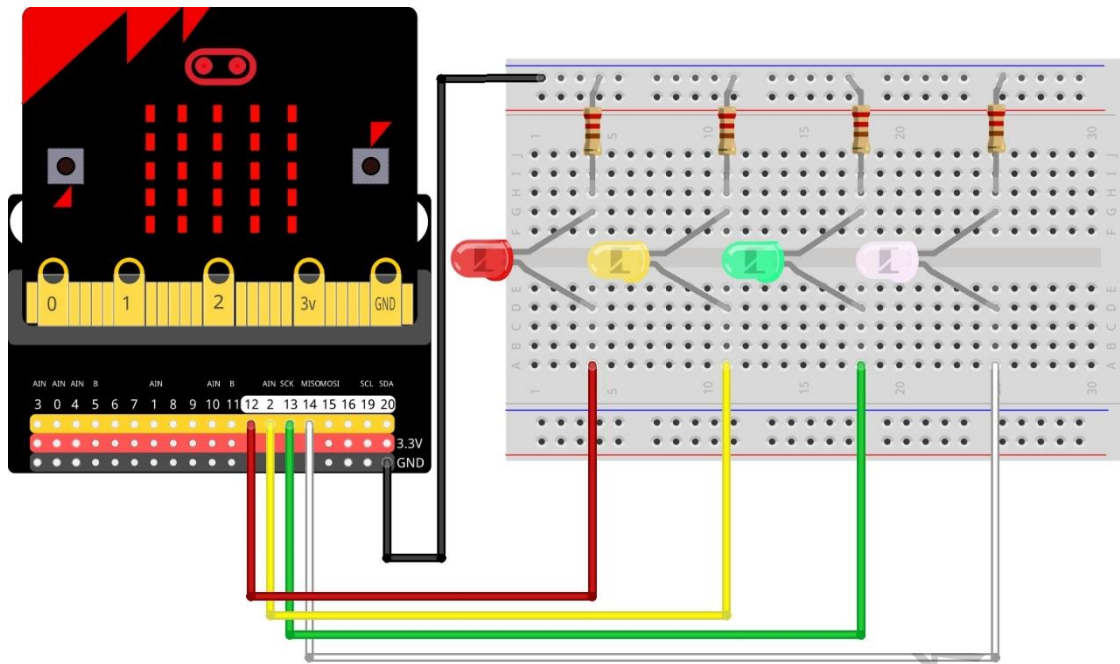


<그림 2-6-3> 마이크로 비트 확장 보드

2.6.3 만들어보기

✓ 외부 LED 제어하기

먼저 아래의 그림과 같이 연결을 해줍니다.



<그림 2-6-4> 외부 LED 사용 결선도

마이크로 비트	LED
12 번 핀	1 번 LED+
2 번 핀	2 번 LED+
13 번 핀	3 번 LED+
14 번 핀	4 번 LED+

위 표와 같이 마이크로 비트와 LED 간 연결을 해줍니다. 각 LED 핀의 캐소드 핀(다리가 짧은)은 저항 220 옴을 통하여 마이크로 비트의 GND 와 연결해줍니다.

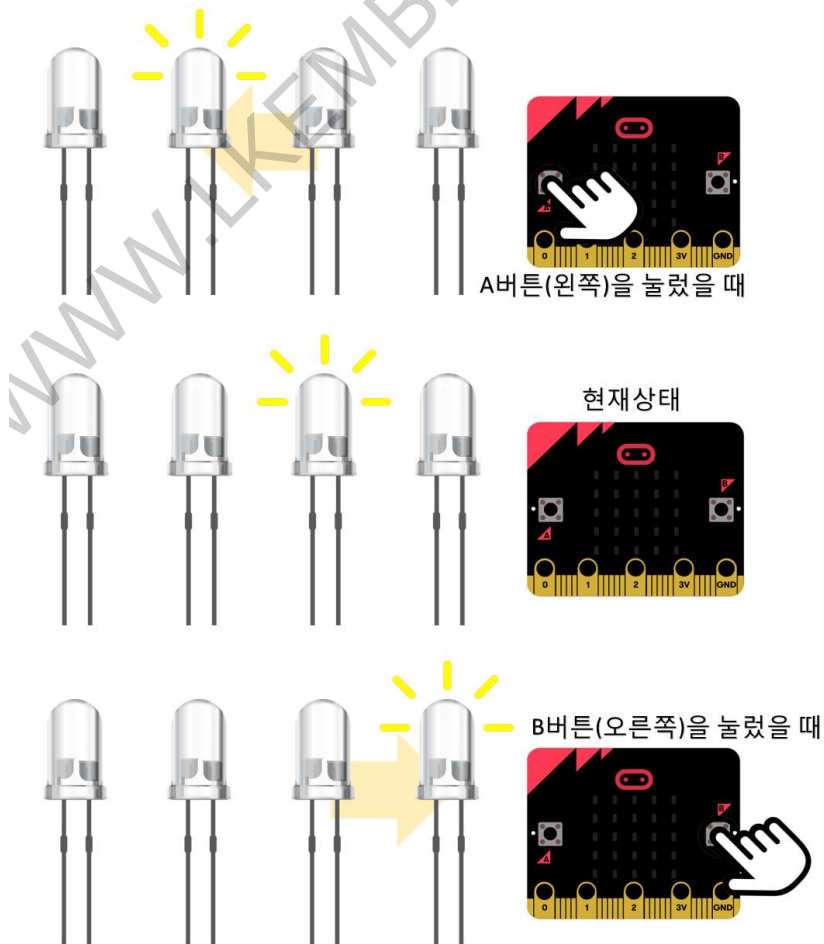
구현할 목표는 마이크로 비트의 버튼을 이용하여, LED on/off 를 제어할 것 입니다.
 먼저 [만약(if)] 블록을 이용하여, 각 LED 를 1 개씩만 켜는 상황을 만들어 줍니다.



<그림 2-6-5> LED 1 개 씩만 ON

이제 버튼 2 개를 이용하여, LED 를 제어하는 코드를 작성합니다.

현재 켜져있는 LED 를 기준으로 왼쪽(A) 버튼을 누르면 왼쪽에 있는 LED 를 켜고, 오른쪽(B) 버튼을 누르면 오른쪽에 있는 LED 를 켜는 동작을 만들어 봅시다.



<그림 2-6-6> 버튼에 따른 LED 동작 구상도



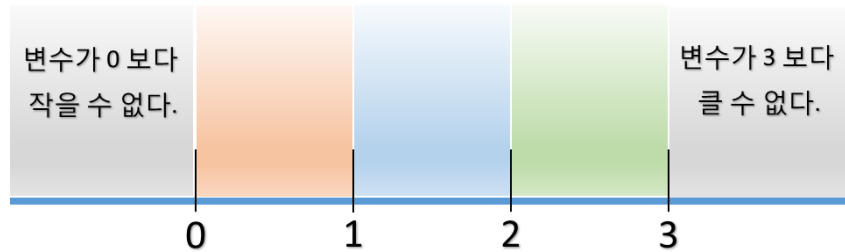
<그림 2-6-7> 변수 값에 따른 각 LED의 동작

이제 버튼을 누를 때마다 동작을 바꾸기 위해서는, 변수의 값도 변경이 되어야 합니다. 왼쪽(A) 버튼을 눌렀을 때 변수에서 1을 감소시켜 주고, 오른쪽(B) 버튼을 눌렀을 때 변수에서 1을 증가시켜 줍니다.



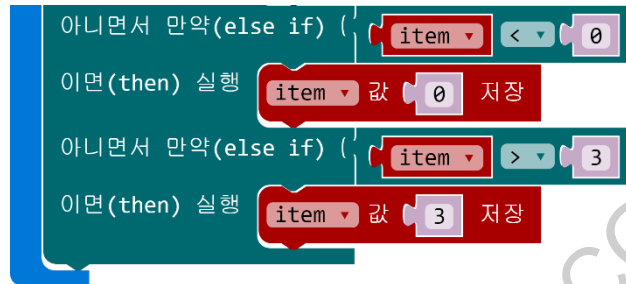
<그림 2-6-8> 각 버튼 동작에 따른 변수값 변화

이제 프로그래밍을 하면 정상적으로 동작이 될까요? 이 과정에서는 한가지 더 짚어야 할 점이 있습니다. 바로 변수는 0보다 작아질 수도 있고, 3보다 더 커질 수 있다는 점이지요. 우리가 원하는 동작을 만들기 위해서는 아래와 같은 조건을 충족해야 합니다.



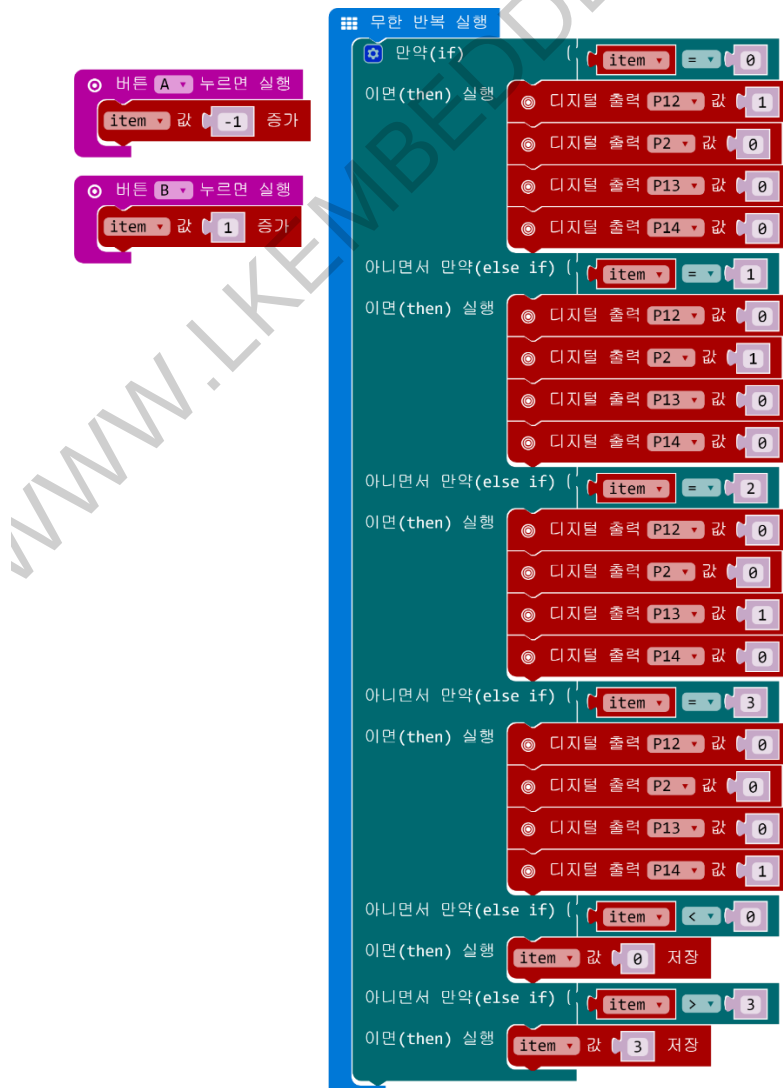
<그림 2-6-9> 변수의 범위 설정

변수가 0 보다 작을 때는 계속해서 0 이 되어야 하고, 3 보다 클때는 3 이 되어야 합니다. 이 조건을 [만약(if)] 블록에 더 추가해주도록 합니다.



<그림 2-6-10> 2 개의 조건 추가

완성시킨 블록을 다운로드 하여 동작을 확인하도록 합니다.



<그림 2-6-11> 완성된 블록

- 맺음말

※기술지원

- 이 책의 예제 소스는 다음의 사이트에서도 다운로드 할 수 있으며, 책에서 궁금한 부분은 상담문의 게시판으로 문의 바랍니다.
- LK 임베디드 홈페이지 문의: WWW.LKEMBEDDED.CO.KR 상담문의 게시판
- 엔지니어 이메일 기술 문의: LKN9270@LKEMBEDDED.CO.KR
- 제품 구입관련 전화문의: Tel. 02-968-8617

※AS 관련 안내

- 제품과 관련된 모든 기술 문의는 LK 임베디드 홈페이지 상담문의 게시판을 이용해주시면 최대한 신속하게 답변을 받아 보실 수 있습니다.
- 제품 초기 불량 시 제품구입일로부터 7 일 이내에 제품 교환 및 환불을 요청할 수 있으며, 학습자 과실로 인하여 하자가 발생하였을 경우에는 수리비가 청구될 수 있고, AS 기간은 6 개월 입니다. 또한 AS 관련 제품은 택배 비용 선불로 보내주셔야 합니다.
- 단순 변심으로 인한 제품 교환 및 환불 요청은 불가하오니 이점 양해바랍니다.
- 문의전화: Tel. 02-968-8617
- LK 홈페이지 주소: WWW.LKEMBEDDED.CO.KR

※감사의 글

LK 임베디드 제품을 구입해 주셔서 감사합니다. 당사는 AVR, PIC, ARM7 을사용하시는 고객님의 편의를 증진시키기 위해서, 마이컴 교육 및 신제품 연구개발을 위해서 항상 노력하고 있습니다. 앞으로도 끊임없는 도전정신을 바탕으로 신제품개발, 완벽한 품질보증 체계 확립, 대 고객 서비스를 통해 고객의 마음을 편하게 하는데 정진할 것입니다. 본 제품을 활용하여 마이컴 학습 및 제품개발에 큰 도움 되시기를 바랍니다.



도서 A/S 안내

당사는 더 좋은 품질의 책을 발간하기 위해서 저자와 출판사의 의견 뿐만 아니라 독자 여러분의 의견도 귀 기울여 듣고 적극 반영하려 노력하고 있습니다

독자 여러분의 좋은 충고나 오타자 또는 레이아웃 및 이미지 등의 오류에 대한 적극적인 의견을 주시면 감사의 마음으로 듣고 신속하게 보완하여 더 좋은 책이 될 수 있도록 최선을 다하겠습니다.

채택된 의견과 최초로 제보된 오자, 오타자, 오답에 대하여는 기념품을 증정하여 드리겠습니다.

- 마이크로 비트 Micro:bit 입문하기 -

마이크로 비트 Micro:bit

저 자 이 경 남

발 행 처 주식회사 LK 임베디드

주 소 서울특별시 노원구 동일로 986 노원 프리미어스 엠코
상가 2층 102 동 215 호, 115 호

전 화 02 - 968 - 8617

팩 스 02 - 968 - 8618

홈페이지/ WWW.LKEMBEDDED.CO.KR

이 메 일 lkn9270@lkembedded.co.kr

■ 이 책을 무단복사, 복제, 전재하는 것은 저작권법에 저촉됩니다.

정가 12,100 원