

2024 CO-SHOW

MCU 응용 경진대회

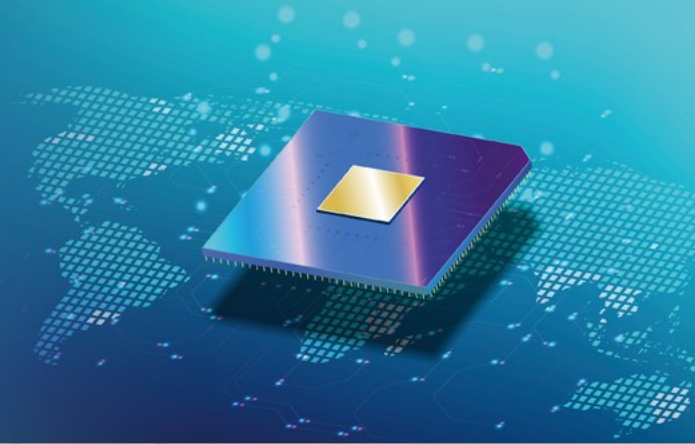
2024. 11. 20 ~ 11. 22

차세대반도체 혁신융합대학 사업단





대회 소개



대회 취지

하드웨어와 소프트웨어의 설계

- 가상 기기의 프로그래밍이 아닌, MCU와 IC로 동작하는 시스템을 구성.
- MCU를 사용함으로써 반도체 설계를 이해.

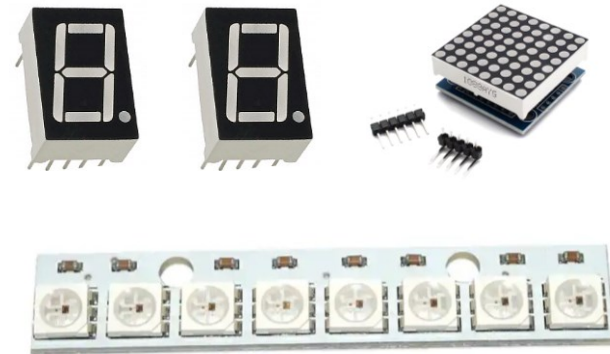
실제 제품의 설계

- 판매 상품의 동작 원리를 가정해 보고, 제품을 분해하여 그 가정을 확인.
- 동작 원리를 이해한 후 제품을 설계.

기본에서 출발

- Datasheet만 참조하여 모든 코드를 직접 작성.
- 최소의 자원으로 최대의 기능을 이끌어 내는 능력 배양.

피아노 LED Visualizer



대회 방식

Round 별 소주제의 작품 제작

	소주제 발표 시각
Round 1	11/20 11:00
Round 2	11/21 10:00
Round 3	11/21 13:00

Round 마다 결과물을 이메일로 제출

- 결과물: 동작 동영상, 작성 코드, 회로도

Round 1, 2 제출은 11/21 18:00 마감

Round 3 제출은 11/22 10:30 마감

제출 시각은 평가에 반영

행사장 운영

	행사장 사용
Round 1	11/20 10:00~16:50
Round 2	11/21 10:00~18:00
Round 3	11/22 10:00~10:30

11/20, 11/21에는 행사장 퇴장 후에도 작업 가능.

11/22 10:30부터 Round 3 대면 심사 진행

과제 구성 및 평가 기준

과제 구성

- 변별력 있는 평가를 위해 주어진 시간에 비해 높은 난이도로 구성.
- 모든 팀원이 참여할 수 있도록 과제 수가 많음.
- 하드웨어 분해 및 제작 포함 (납땜)

평가 기준

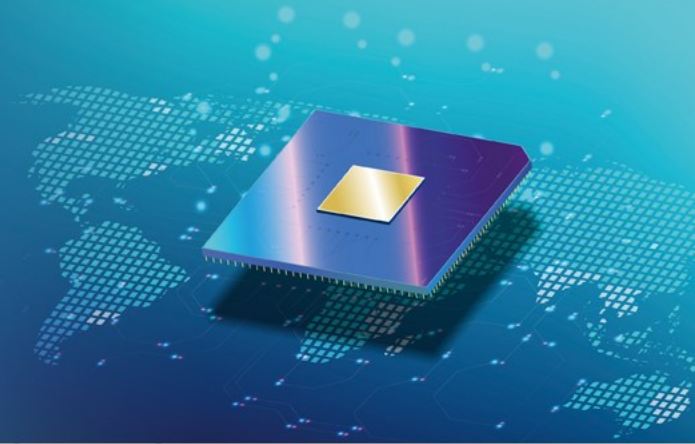
평가 항목	배점
완성도 (기능 구현)	50
난이도 및 창의성	30
소요 시간	20
합계	100

대회 규칙

1. 인터넷 검색은 가능하나, ChatGPT 등 AI가 작성한 코드의 사용은 금지.
2. AVR Libc 라이브러리만 사용 가능 (NonGNU.org).
3. 개인 부품 및 장비 사용은 가능하나, 결과물은 배포한 부품으로만 제작.
4. 부품 불량일 경우 교체 요청 가능.
5. 본인 과실로 부품/장비를 손상하거나 소진한 경우, 교체/추가 제공하나 건당 -5점 감점.
6. 제공함 부품을 모두 사용하지 않아도 됨.
7. 안전 유의 (특히 납땜 인두).
8. 질문 가능. 답변은 공개로 함.

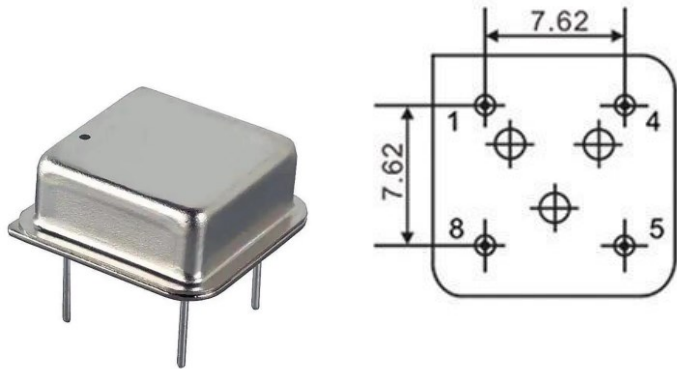


ATmega88PA 사용법



ATmega88PA 회로 구성 (Power, Clock)

Crystal Oscillator (16MHz)



PAD Function:

- 1: Control Voltage
- 4: **GND**
- 5: **Out**
- 8: **Vdd (5V/3.3V)**

ATmega88PA-PU

(PCINT14/ $\overline{\text{RESET}}$) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
	VCC	22	GND
	GND	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 ($\overline{\text{SS}}$ /OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

ATmega Clock Options

Clock Option	Frequency
내부 128kHz RC Oscillator	128kHz 고정
내부 RC Oscillator	8MHz 고정
외부 Crystal	최고 20MHz
외부 Oscillator	최고 20MHz

Clock Source 선택 (Fuse Low Byte)

Table 9-1. Device Clocking Options Select⁽¹⁾

Device Clocking Option	CKSEL3...0
Low Power Crystal Oscillator	1111 - 1000
Full Swing Crystal Oscillator	0111 - 0110
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128kHz RC Oscillator	0011
Calibrated Internal RC Oscillator ~8MHz	0010
External Clock	0000
Reserved	0001

Table 28-9. Fuse Low Byte

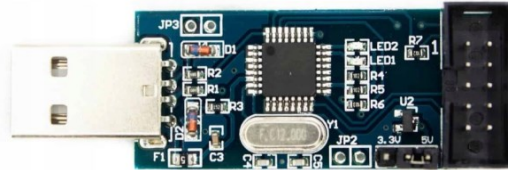
Low Fuse Byte	Bit No	Description	Default Value
CKDIV8 ⁽⁴⁾	7	Divide clock by 8	0 (programmed)
CKOUT ⁽³⁾	6	Clock output	1 (unprogrammed)
SUT1	5	Select start-up time	1 (unprogrammed) ⁽¹⁾
SUT0	4	Select start-up time	0 (programmed) ⁽¹⁾
CKSEL3	3	Select Clock source	0 (programmed) ⁽²⁾
CKSEL2	2	Select Clock source	0 (programmed) ⁽²⁾
CKSEL1	1	Select Clock source	1 (unprogrammed) ⁽²⁾
CKSEL0	0	Select Clock source	0 (programmed) ⁽²⁾

기본 값: 0x62

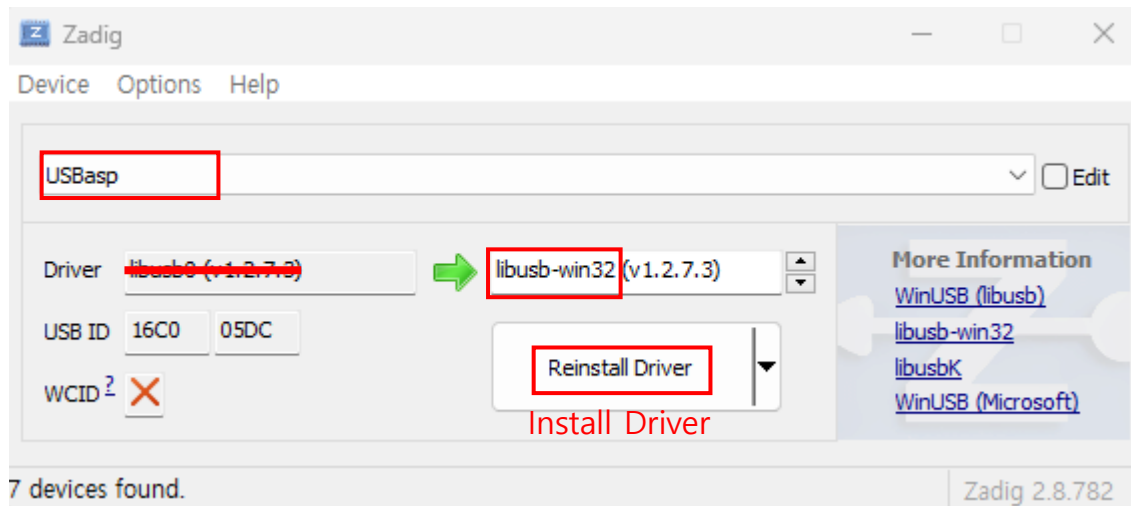
외부 Oscillator 사용 시: 0xE0

USBasp 드라이버 설치

① USBasp를 USB 포트에 삽입

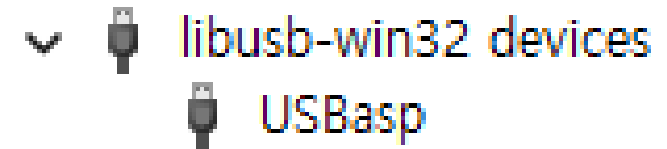


② Zadig를 실행하여 드라이버 설치



위 예제는 이미 설치된 PC에서 실행하였기에 Reinstall로 나오지만, 처음 실행시에는 Install Driver가 표시됨.

③ 윈도우 장치관리자에서 USBasp 드라이버가 설치됨을 확인.



④ 설치 확인 후 USBasp를 USB 포트에서 분리 (커넥터 핀에서 전원 공급을 차단하기 위해).

ATmega88 프로그래밍 준비

① ATmega88을 Breadboard에 삽입

28번 핀

15번 핀

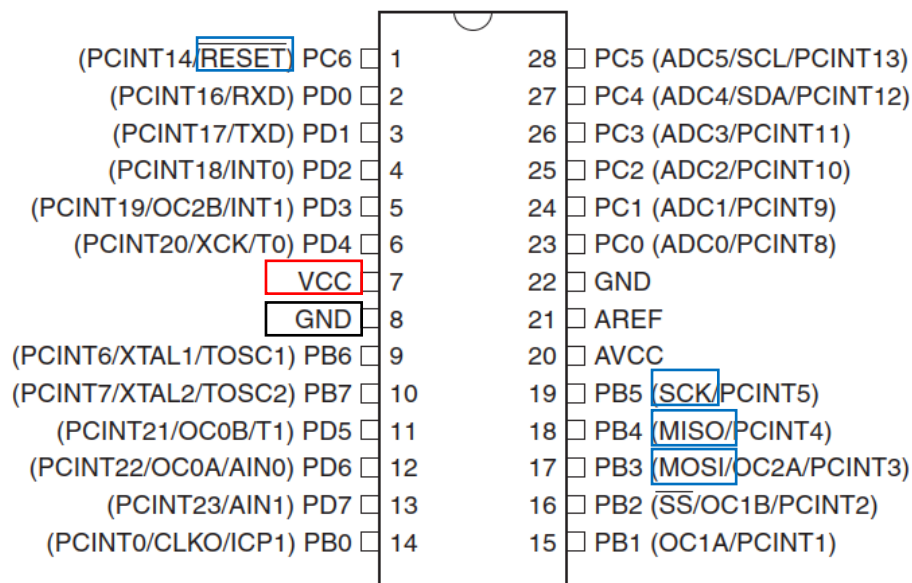
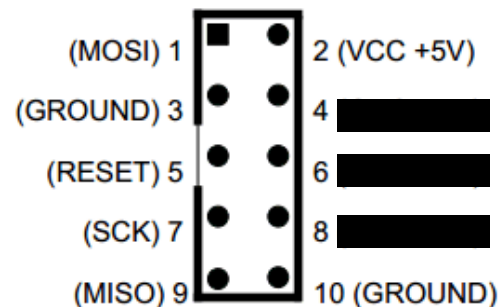
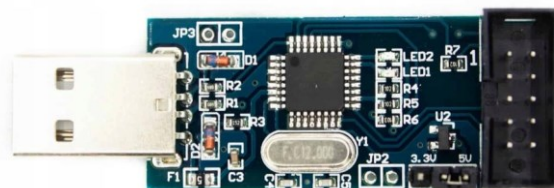


14번 핀

점 (dot)이 있는 핀이 1번 핀

② USBasp의 커넥터와 ATmega88을 연결

USBasp



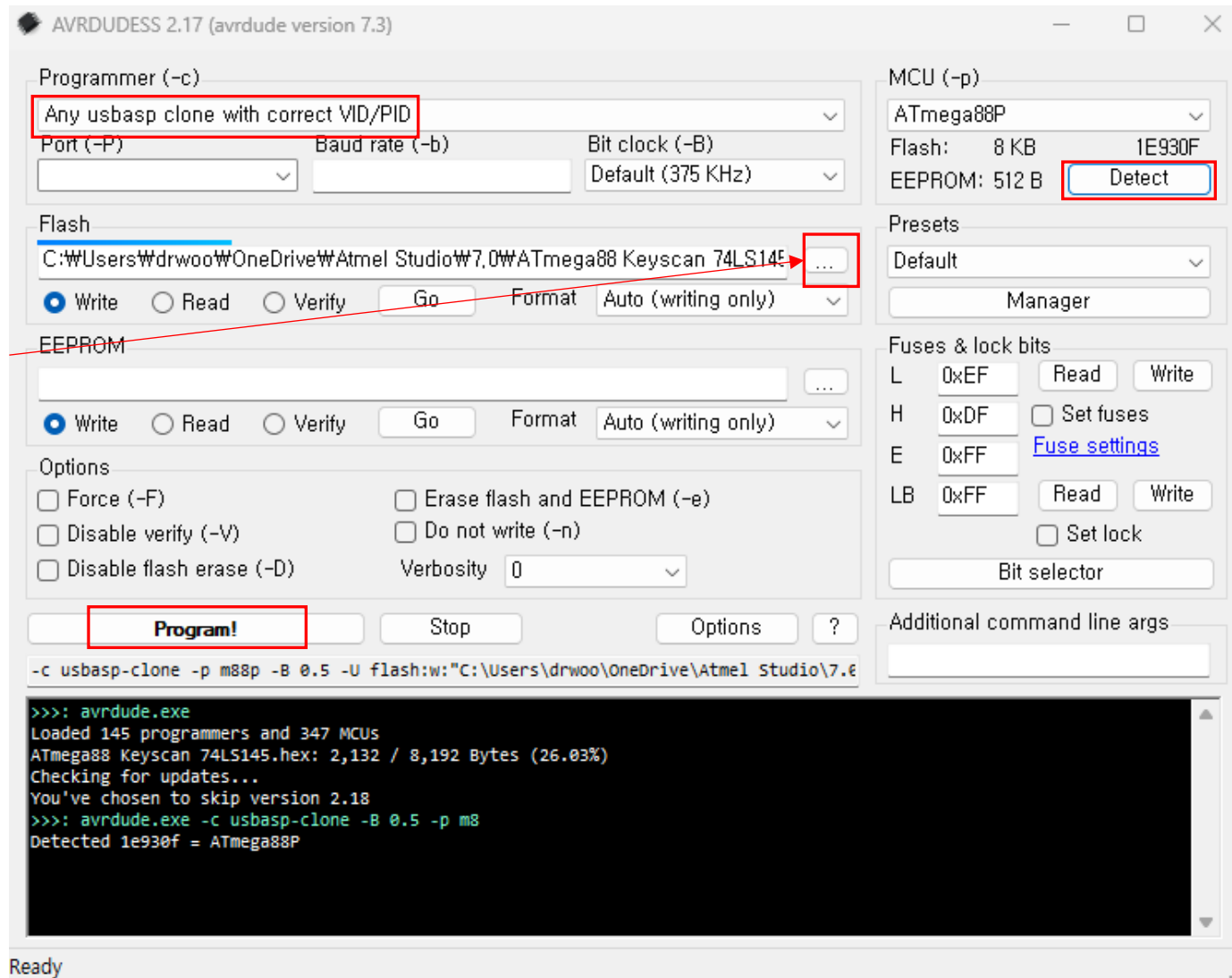
ATmega88

HEX 파일 업로드

① 기본 값으로 선택된 USBasp clone 을 사용.

③ 파일 경로 버튼을 눌러서 업로드할 hex 파일을 지정 (Debug folder).

④ Program 버튼을 누르면 hex 파일의 코드가 ATmega88로 업로드 됨.

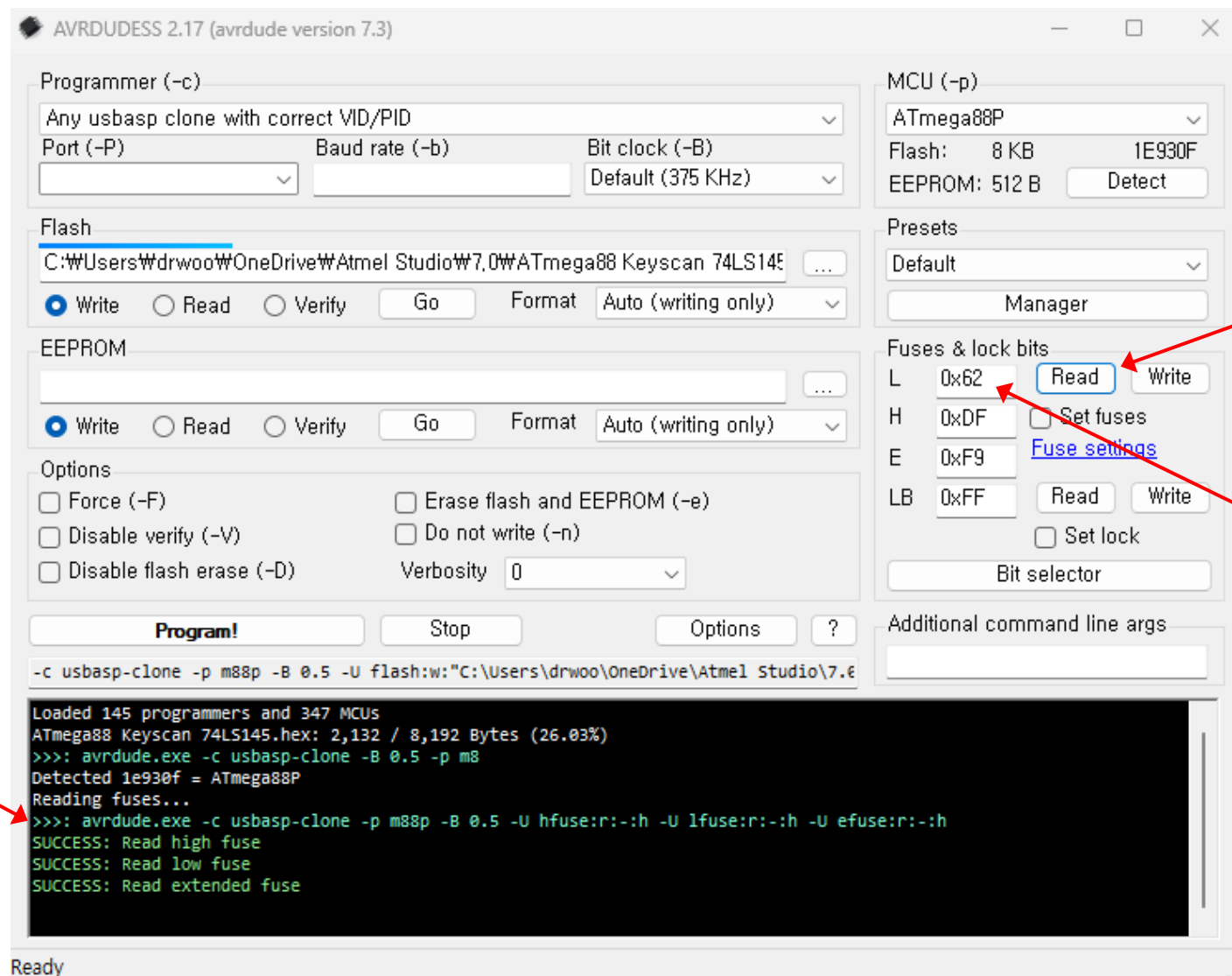


② Detect 버튼을 눌러 USBasp에 연결된 MCU 기종이 ATmega88임을 확인.

★ MCU Detect가 안 되면 AVRDUDESS 2.14 사용.

⑤ 코드 업로드가 끝나면 그 프로그램이 ATmega에서 자동 실행됨.

Fuse Bits의 Read



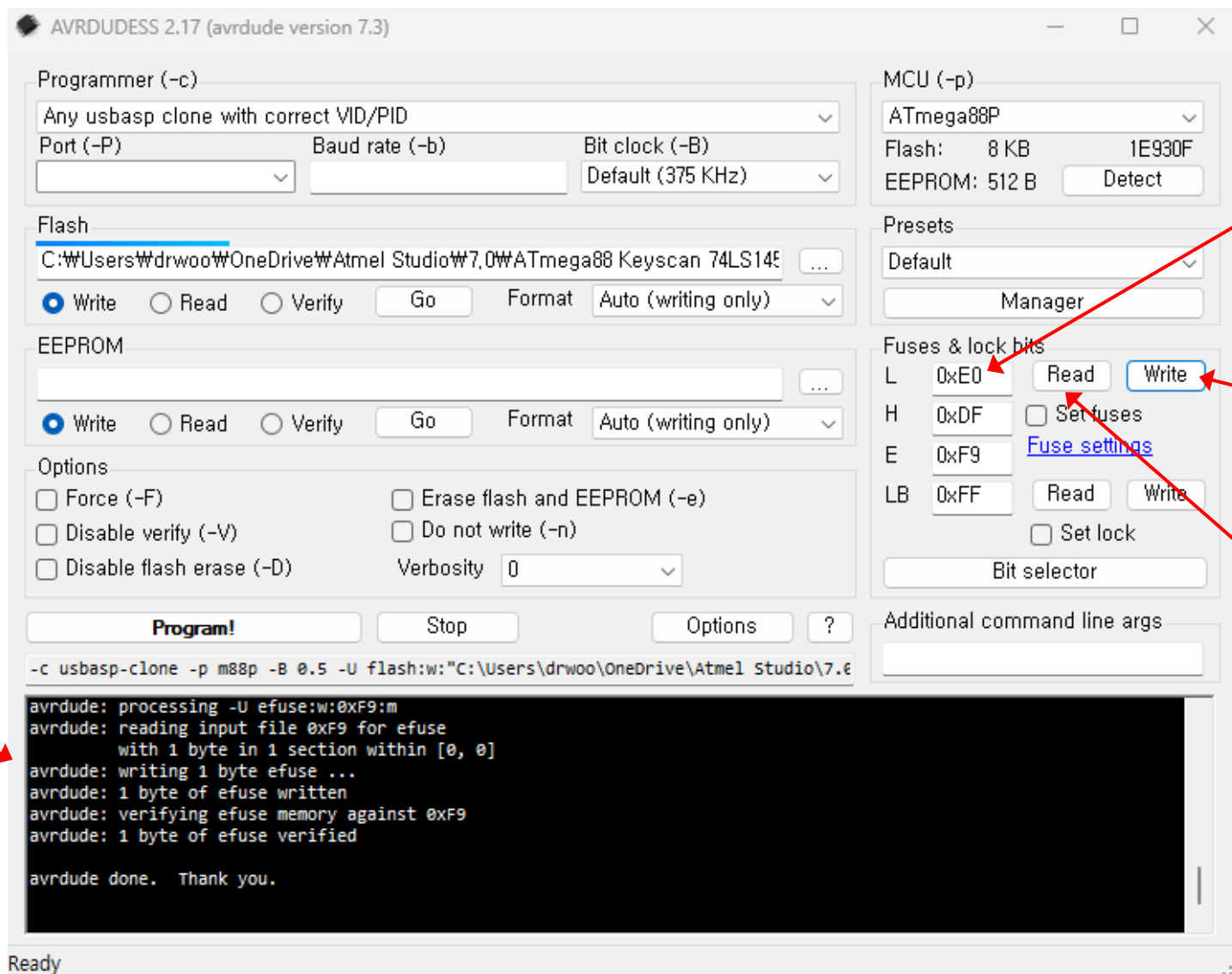
② Fuse Byte의 Read
가 실행됨을 확인.

① Fuse Bits의 Read
버튼을 누름

③ Fuse Bits 의 Low
Byte (L) 가 0x62로
Read 됨을 확인.

0x62 = 1MHz 내부
Clock 사용

외부 Oscillator 사용을 위한 Fuse Bits Write



③ 1 byte의 Fuse Bits가 write 됨을 확인.

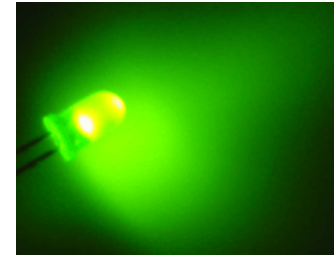
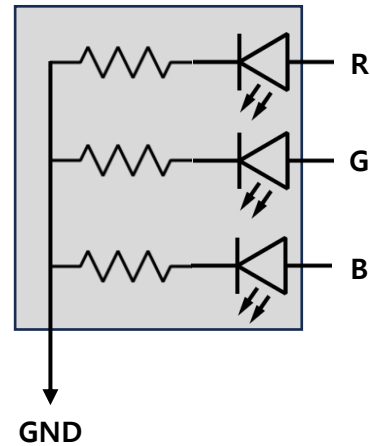
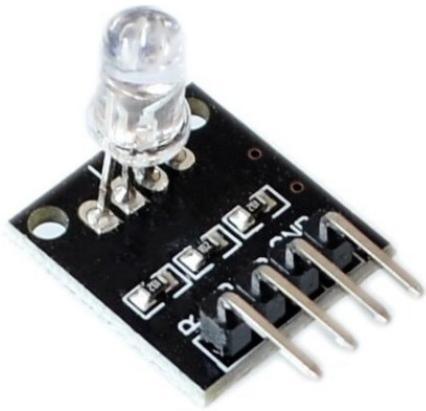
① Fuse Bits의 Low Byte (L) 에 0xE0를 입력

② Write 버튼을 누름

④ Read 버튼을 누르면 Low Byte (L) 이 0xE0로 유지됨을 확인

Fuse Bits 설정은 새 IC에 한번만 하면 됨.

기본 테스트: LED Heartbeat



- Red, Green, Blue LED 가 한 패키지에 통합된 모듈.
- PCB에 전류 제한 저항이 연결되어 있어서 직접 MCU 핀 및 Ground에 연결 가능.
- Active High

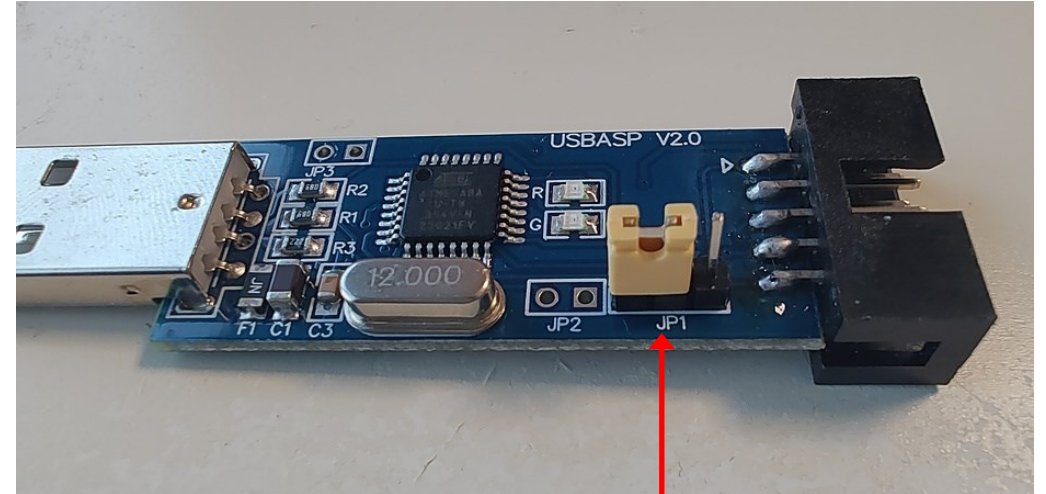
- LED 하나를 1초 주기로 점멸.

★ LED 점멸은 모든 Round 결과물에 포함.

USBasp 점퍼 납땜



정품에는 전원 공급
전압 (5V 또는 3.3V)
를 선택하는 점퍼가
있음.

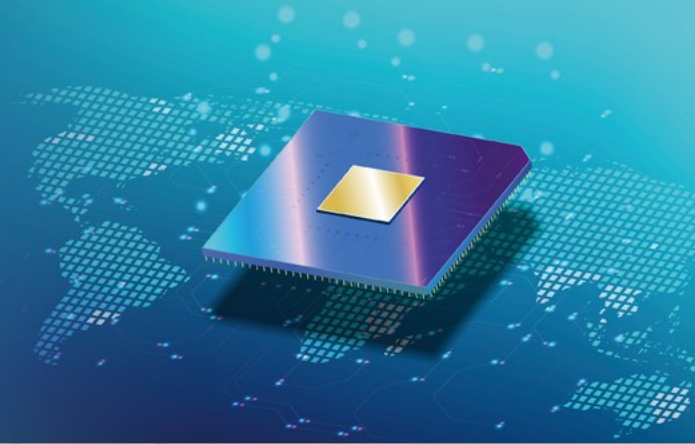


배포 제품에는 점퍼가 없어서
각자 납땜을 해야 함.

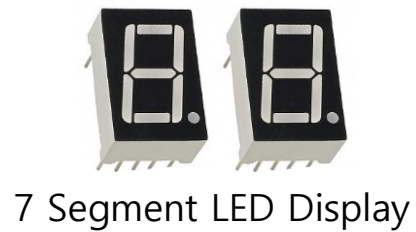
5V / 3.3V 선택 핀도 반대임.



Round 1



Round 1: 건반 인식과 LED 디스플레이

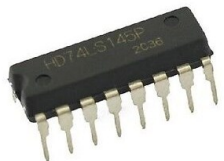


MAX7219를 사용한 8x8
Dot Matrix LED

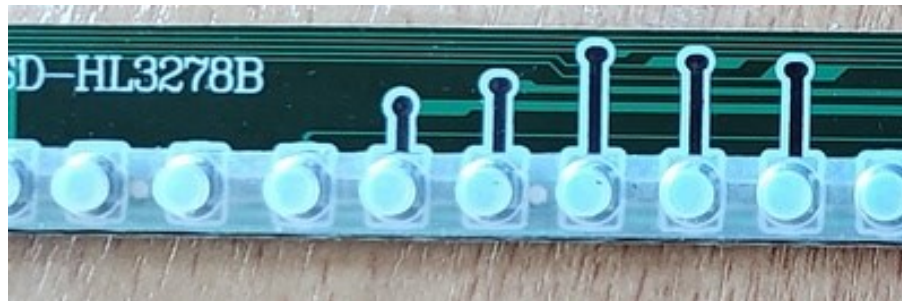


장난감 피아노를 분해하여
건반 스위치 PCB의 회로를
분석.

건반 스위치와 MCU를 연결.



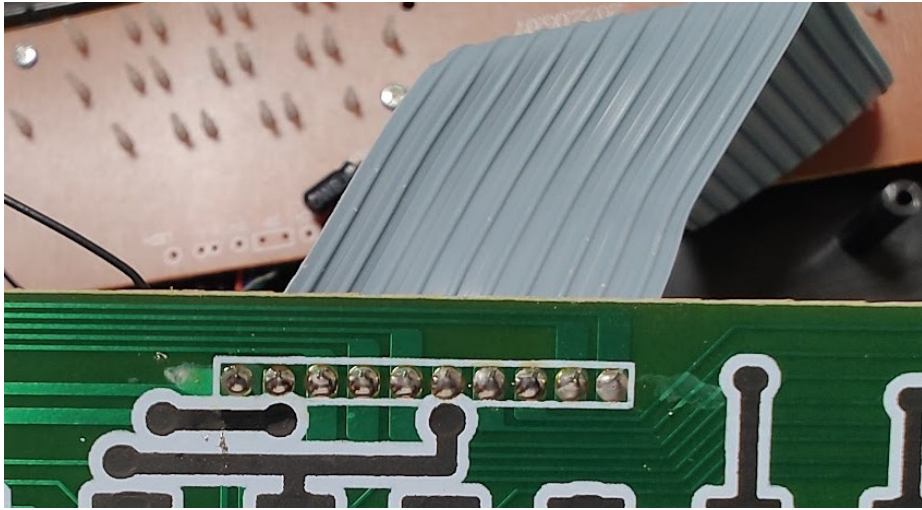
74LS145



건반 스위치의 눌림과 떼임
을 인식하는 코드 작성.

건반을 눌렀을 때와 떼었을 때
건반 번호를 Display에 표시.

피아노 건반과 MCU 연결

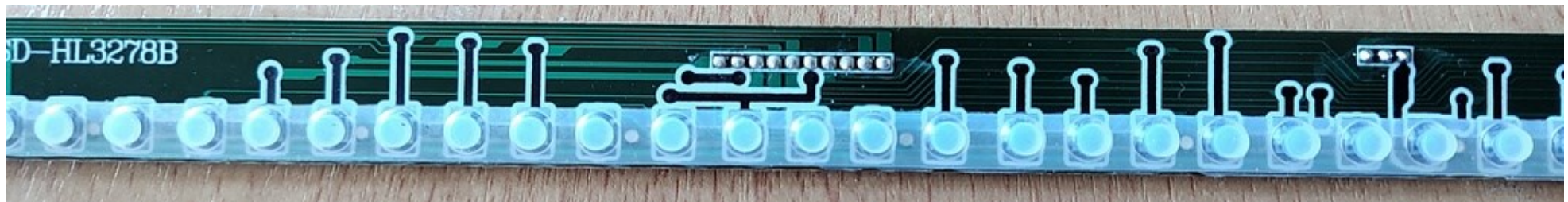


① PCB에 납땜 되어 있는 내부 케이블을 분리하고, Solder Wick을 사용하여 PCB의 납을 제거.

★★주의: PCB를 너무 오래 가열하면 동박이 박리됨. 어려우면 도움 요청.



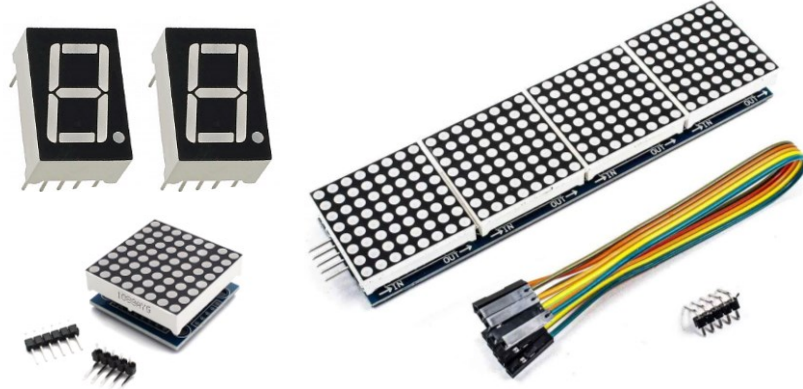
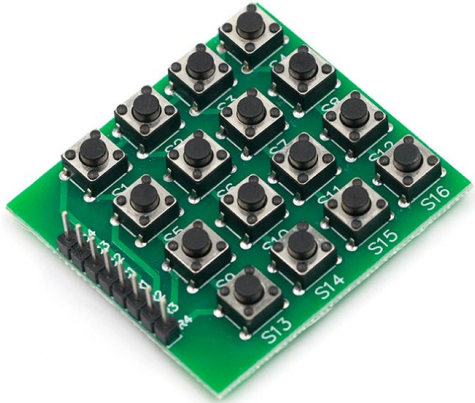
② 납을 제거한 케이블 연결 구멍에 Pin Header를 꽂고 납땜을 함. 그리고 Jumper 선을 통해 Pin Header와 MCU를 연결.



연결 선을 위한 케이스 가공

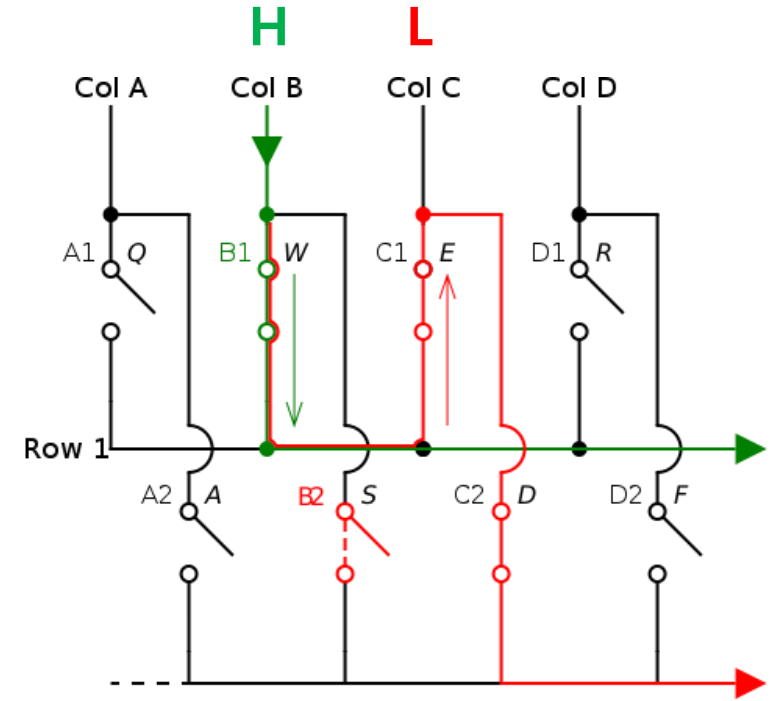


Round 1 참고 사항



- 건반 스위치 대신 4x4 키패드를 사용하여 기본 코드 작성과 테스트가 가능함.

- 어떤 숫자, 기호, 패턴을 표시하여도 상관 없음.
- 디스플레이 기능을 많이 활용하여 창의적인 표현을 할 수록 높은 점수를 부여.



- 동시에 여러 스위치가 눌렸을 때, 출력 핀이 충돌하여 과전류가 흐르는 상황은 피해야 함.

Dot Matrix Display 간의 물리적 연결

Pin Header를 PCB 아래 쪽에서 납땜한 후,
Jumper Pin을 꽂아 연결.

