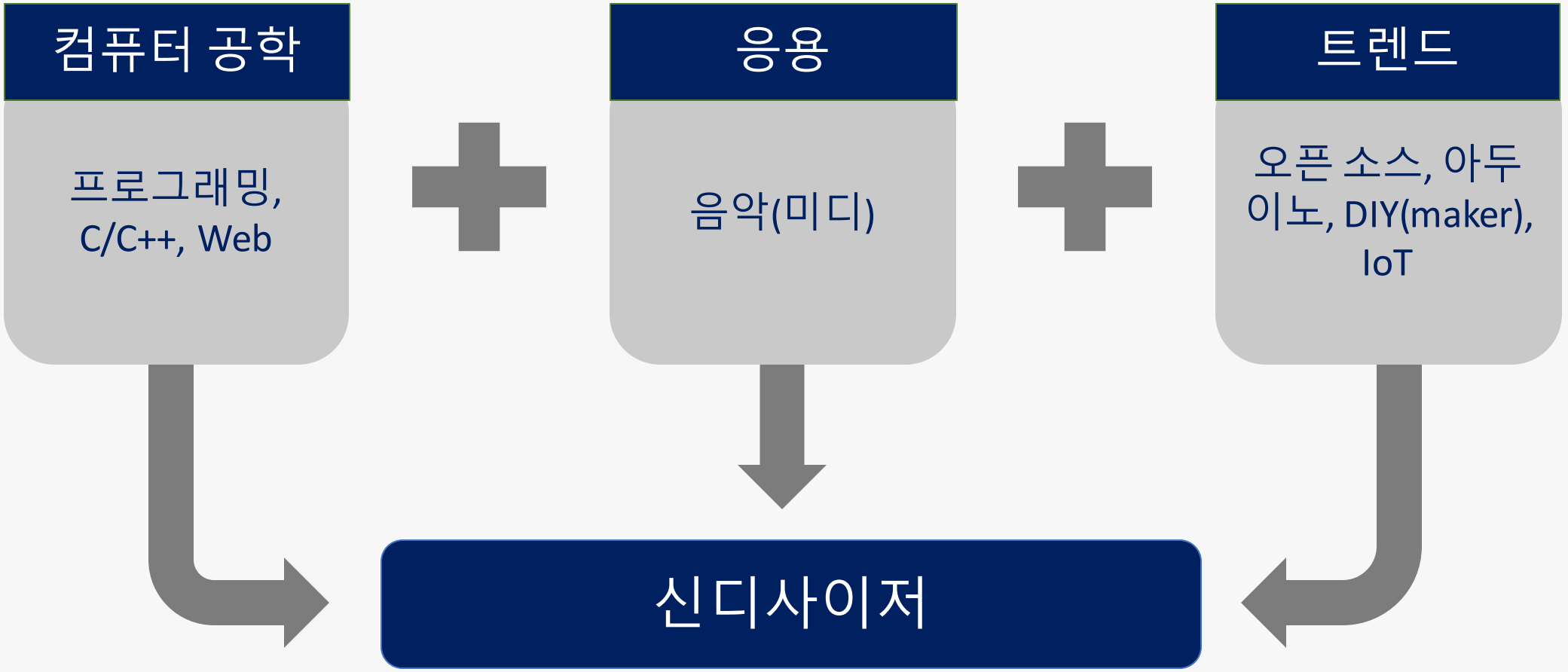
신디사이저와 IoT

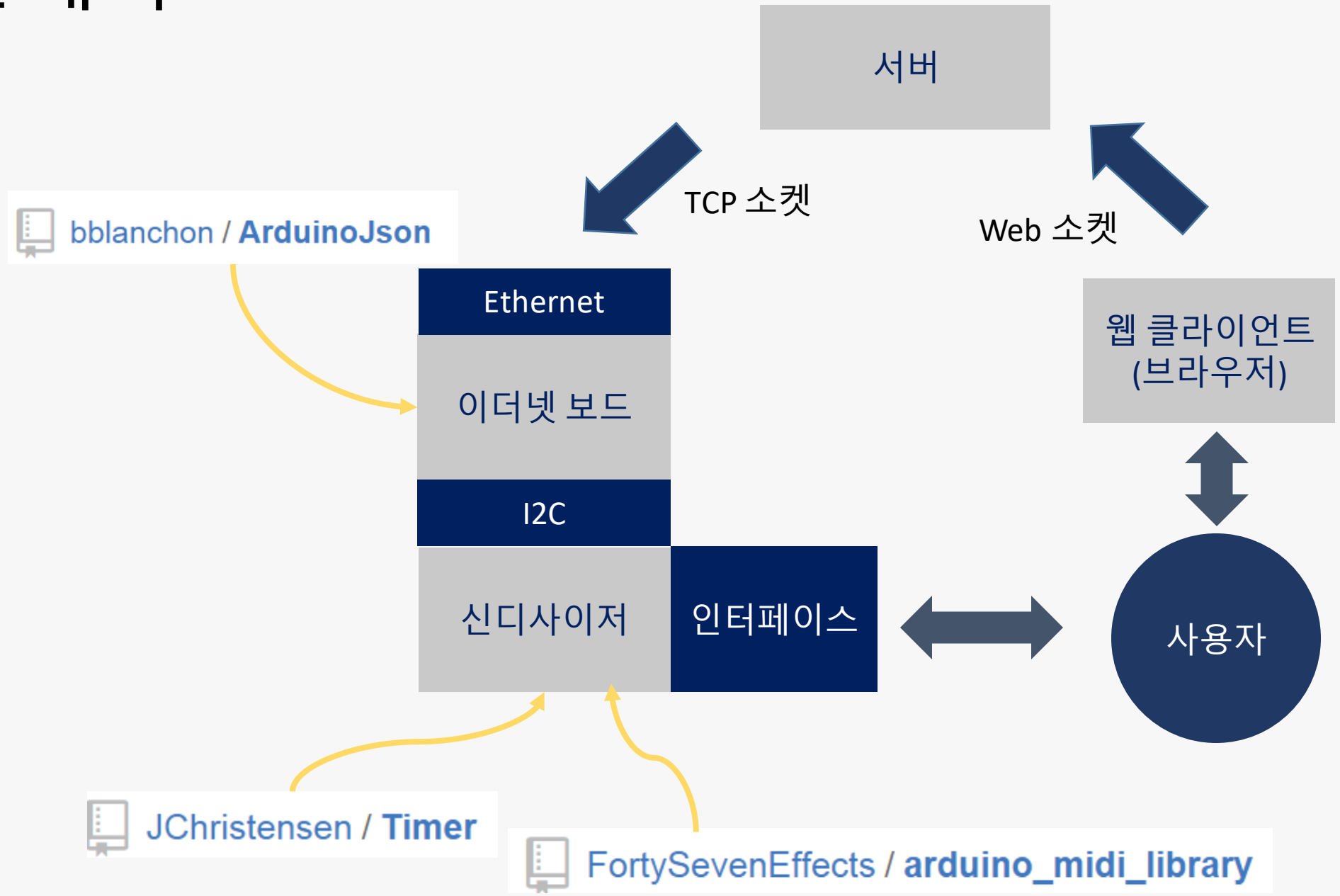
하프시코드
201224540 조용래

목차

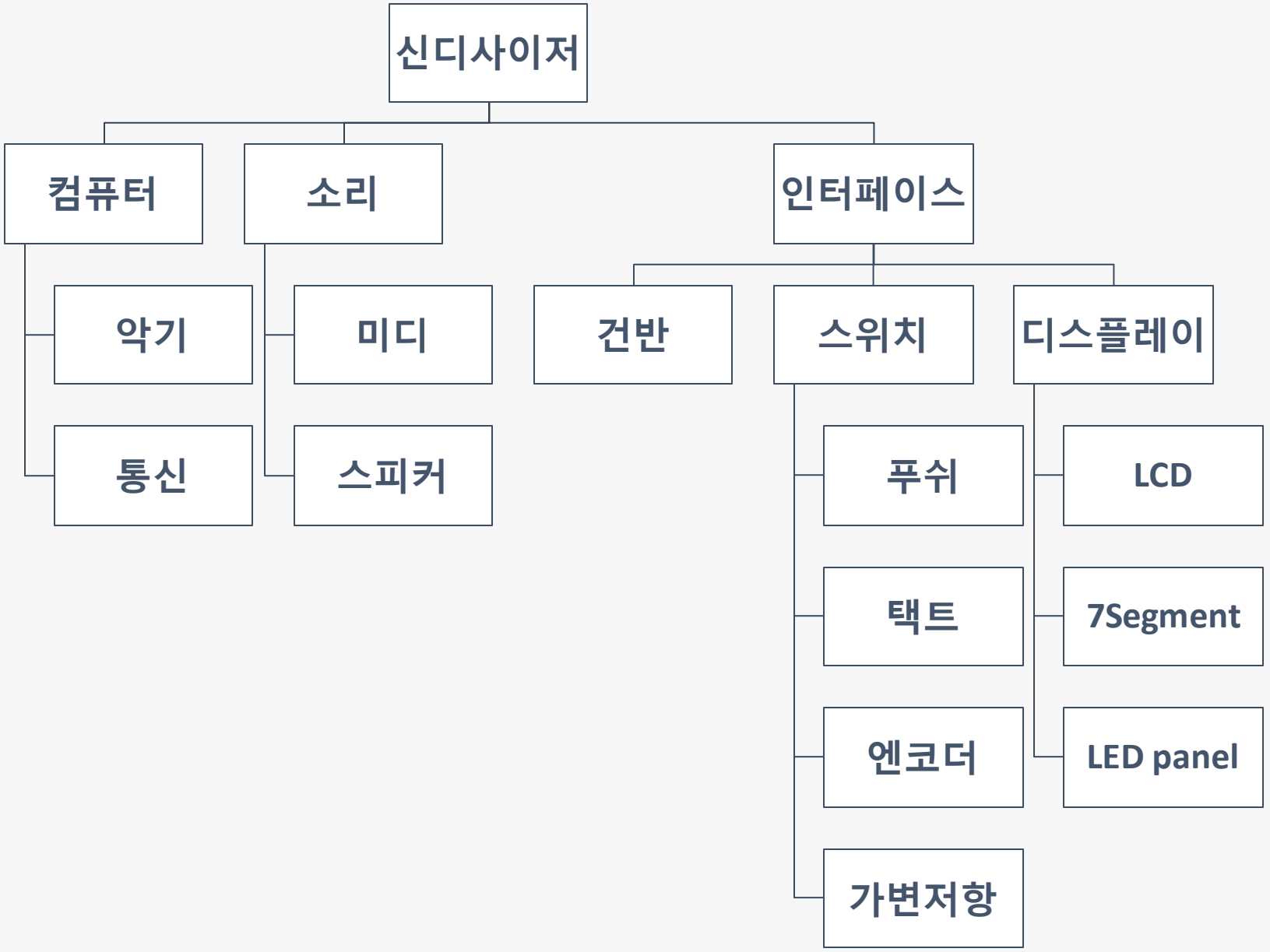
동기



전체 구조


서버
TCP 소켓
Web 소켓
bblanchon / ArduinoJson
Ethernet
이더넷 보드
I2C
신디사이저
인터페이스
웹 클라이언트
(브라우저)
사용자
JChristensen / Timer
FortySevenEffects / arduino_midi_library

신디사이저 구성

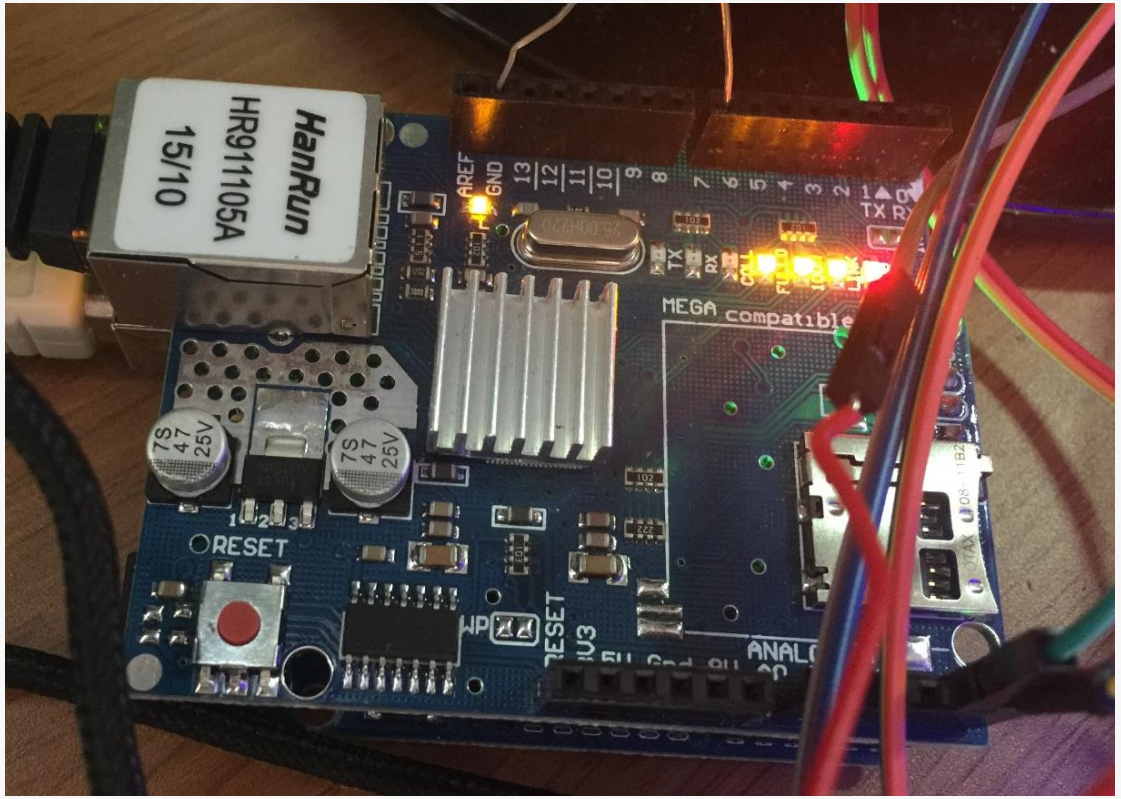

신디사이저
컴퓨터
소리
인터페이스
악기
통신
미디
스피커
건반
스위치
디스플레이
푸쉬
택트
엔코더
가변저항
LCD
7Segment
LED panel

신디사이저 전체 사진

부품 별 설명

마이크로컨트롤러

악기(아두에노 메가)
- 악기 내부의 소프트웨어 구현



통신(아두이노 우노 + 이더넷 쉴드)
- 서버에서 받은 정보를 우노로 전달

미디 프로토콜

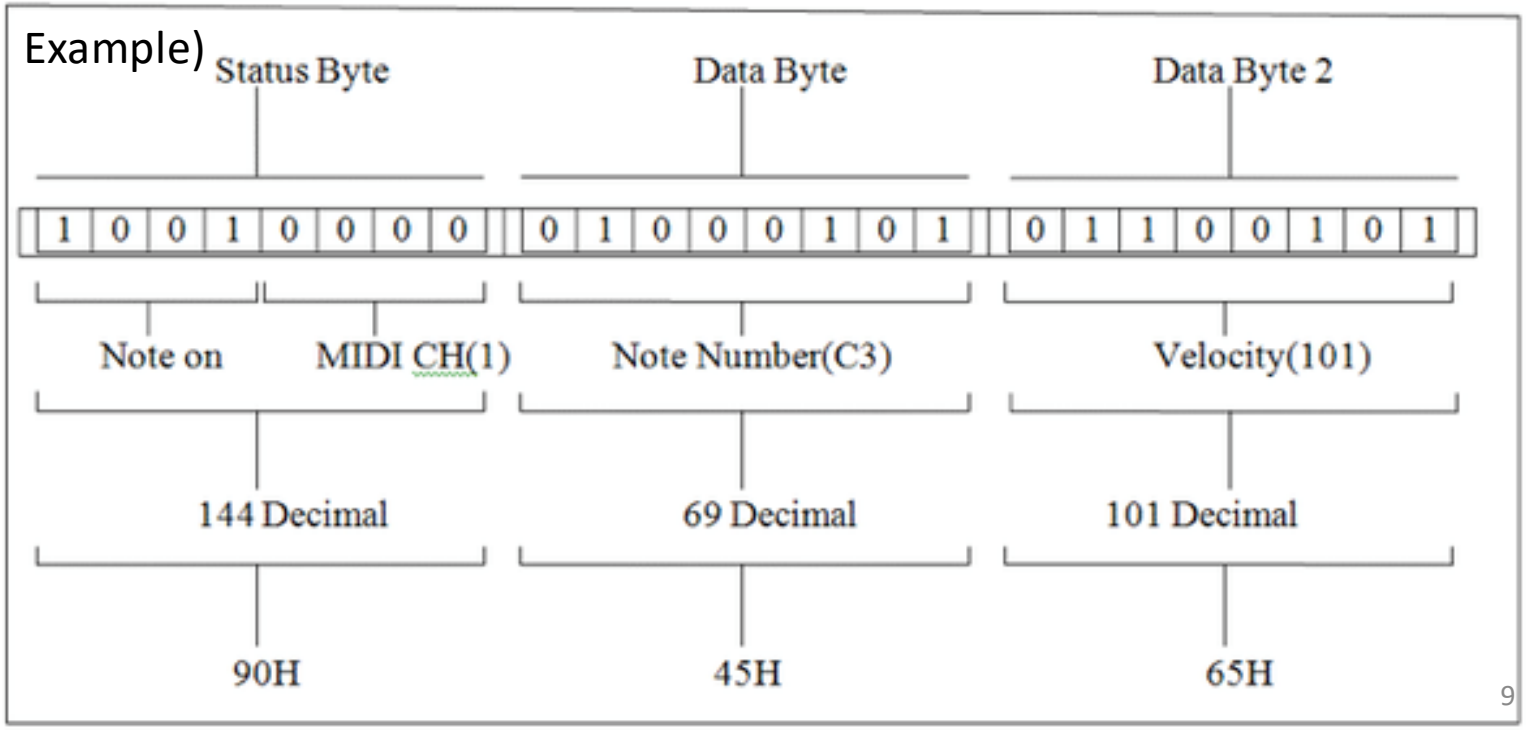
미디(Musical Instrument Digital Interface)

- 전자 악기 간 통신 규약
- 16개의 채널
- 명령 바이트(MIDI Messages)로 수행할 명령과 채널 명시
- 데이터 바이트로 명령 바이트에 맞는 인자 전달

명령어 형식

MIDI commands

0x**8**0	Note Off
0x**9**0	Note On
0x**A**0	Aftertouch
0x**B**0	Continuous controller
0x**C**0	Patch change
0x**D**0	Channel Pressure
0x**E**0	Pitch bend
0x**F**0	(non-musical commands)

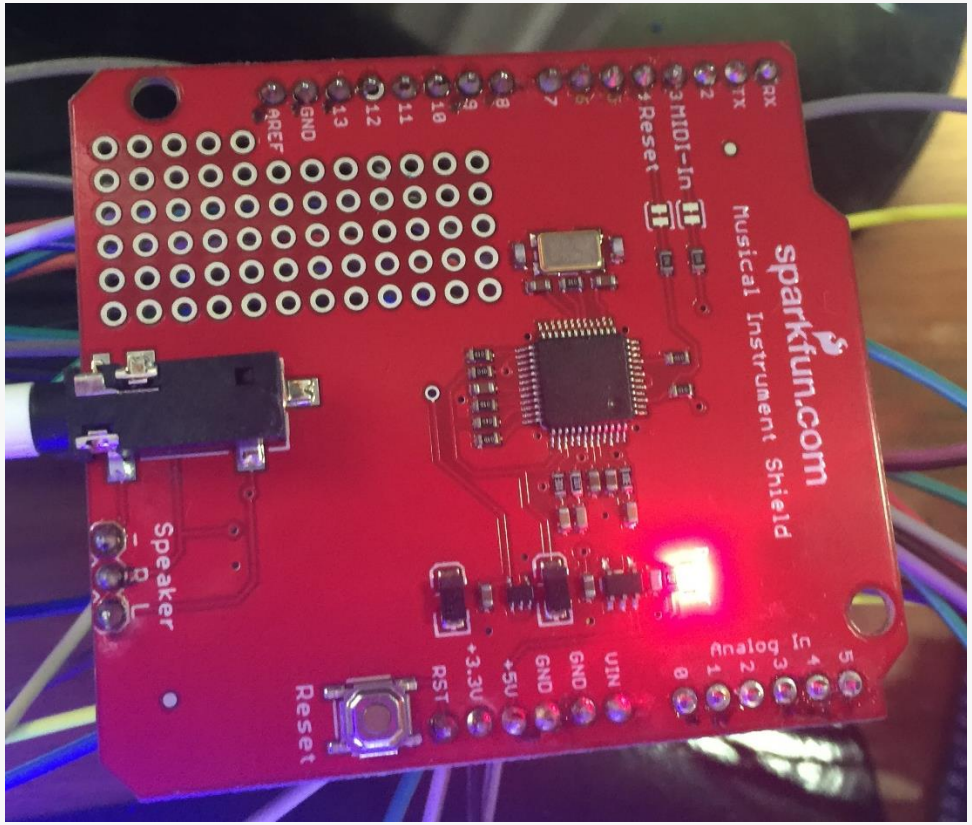


미디 보드

- 디지털 신호를 받아 소리를 만들어 냄
- VS1053b 칩 사용
- 미디 명령어를 받아 음악적 정보를 실제로 처리
- 미디 메시지 일부 지원

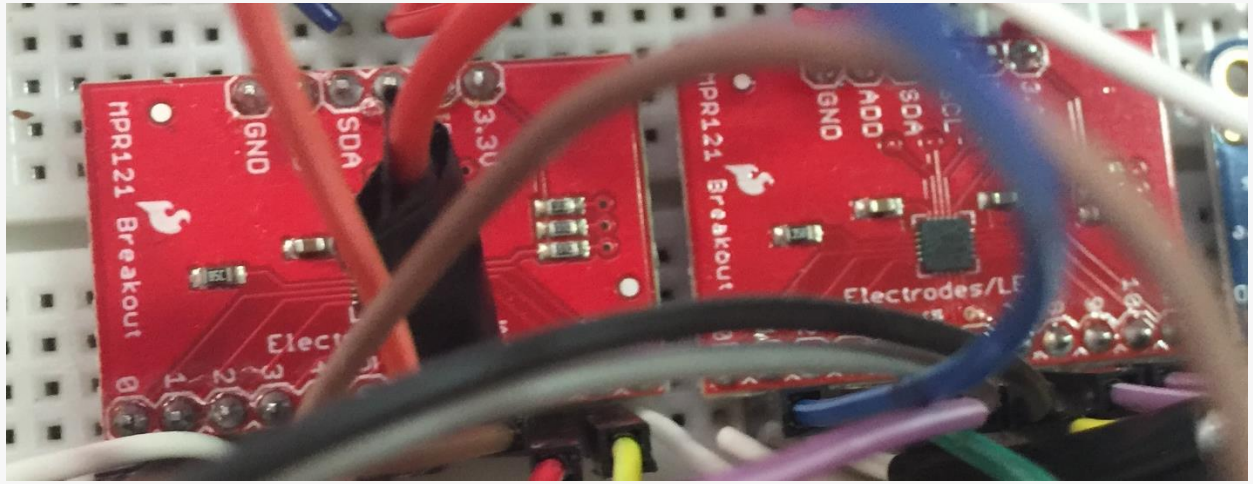
Supported MIDI messages:

- meta: 0x51 : set tempo
- other meta: MidiMeta() called
- device control: 0x01 : master volume
- channel message: 0x80 note off, 0x90 note on, 0xc0 program, 0xe0 pitch wheel
- channel message 0xb0: parameter
  - 0x00: bank select (0 is default, 0x78 and 0x7f is drums, 0x79 melodic)
  - 0x06: RPN MSB: 0 = bend range, 2 = coarse tune
  - 0x07: channel volume
  - 0x0a: pan control
  - 0x0b: expression (changes volume)
  - 0x0c: effect control 1 (sets global reverb decay)
  - 0x26: RPN LSB: 0 = bend range
  - 0x40: hold1
  - 0x42: sustenuto
  - 0x5b effects level (channel reverb level)
  - 0x62,0x63,0x64,0x65: NRPN and RPN selects
  - 0x78: all sound off
  - 0x79: reset all controllers
  - 0x7b, 0x7c, 0x7d: all notes off



SparkFun Musical Instrument Shield

건반 만들기
– MPR121(정전식 터치 센서)



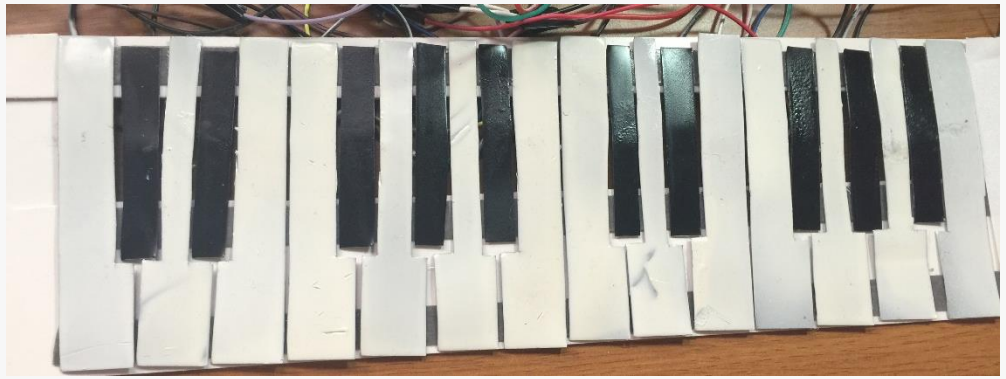
MPR121 Capacitive Touch Sensor Breakout Board

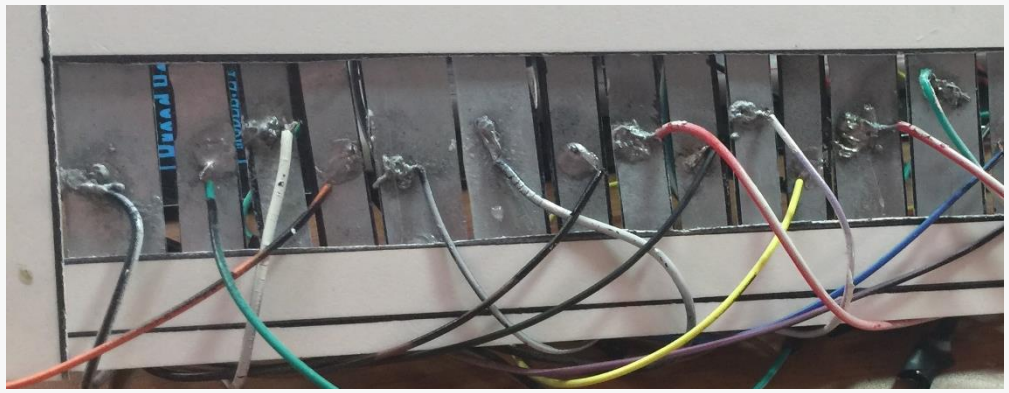
- 12개의 정전식 터치 접점
- 피아노 한 옥타브가 12개임을 고려해 센서 하나당 한 옥타브 할당
- 금속판으로 된 건반의 입력 담당
- I2C 프로토콜

Pin No.	Pin Name	Description
1	$\overline{\text{IRQ}}$	Active Low Open-drain Interrupt Output
2	SCL	I^2C Serial Clock
3	SDA	I^2C Serial Data
4	ADDR	I^2C Slave Address Pin Selects. Connect to VSS, VDD, SDA, SCL to choose address 0x5A, 0x5B, 0x5C, 0x5D respectively.

건반 만들기
– 재료 선택

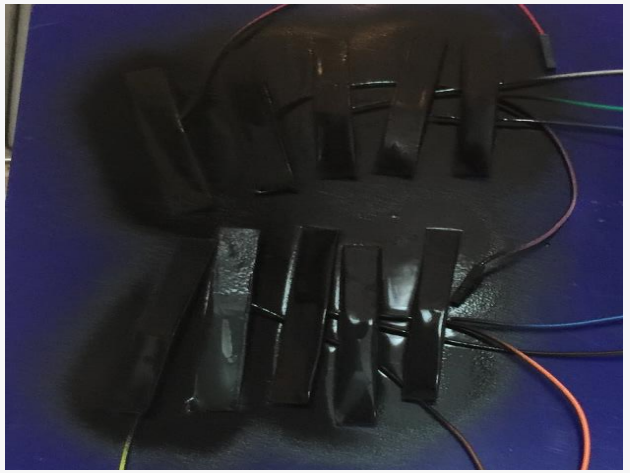
건반 모습





2옥타브 건반

아연판

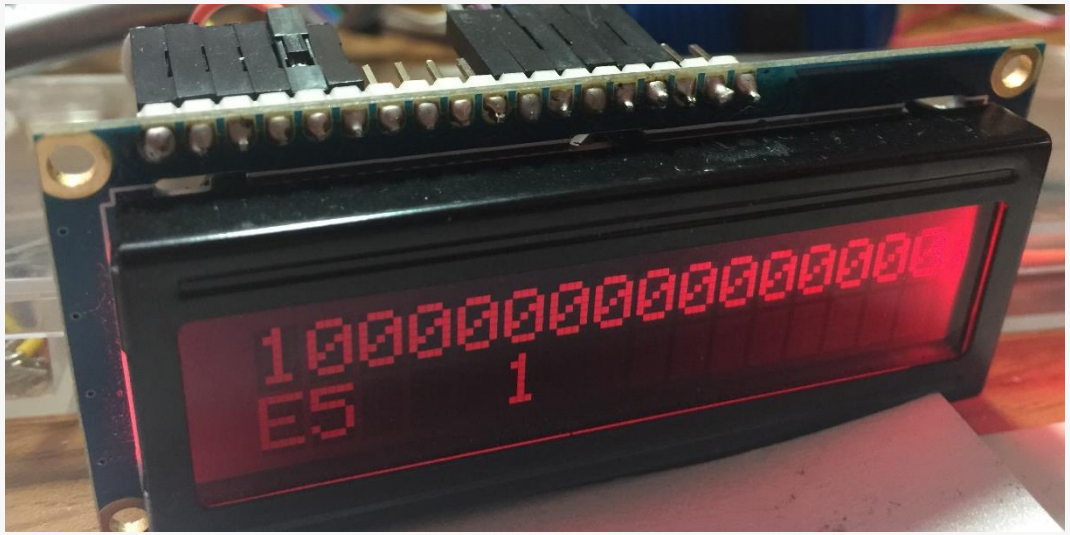


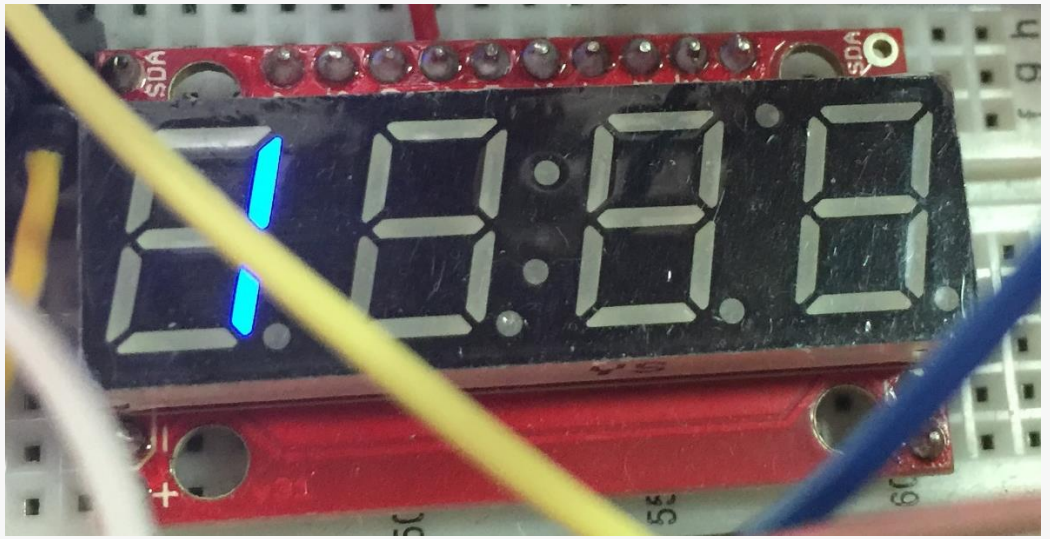
스프레이 뿌린 후 모습

몬태나 스프레이 – 흰, 검

디스플레이

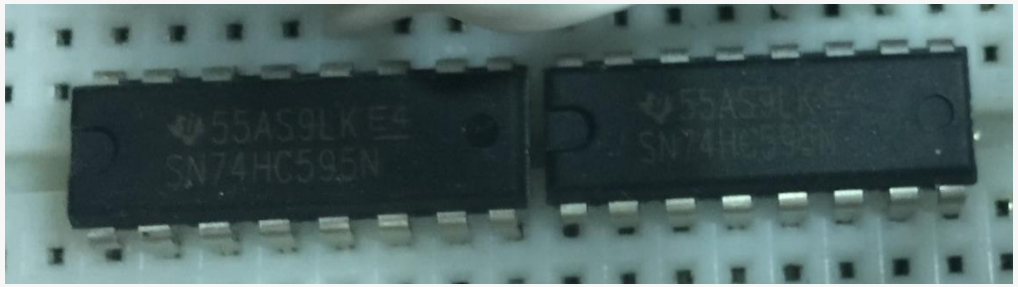
- LCD, 7segment





악기가 사용자에게 보여주는 각종 음악적 수치들(템포, 볼륨, 노트 값, 노트 위치, 옥타브 등등) + 간단한 메시지

디스플레이
- LED 패널(쉬프트 레지스터 + LED)

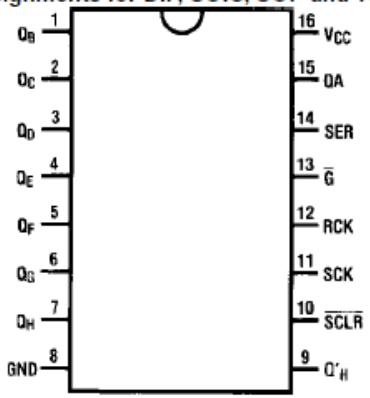


74HC595(쉬프트 레지스터)

Led 라인

Connection Diagram

Pin Assignments for DIP, SOIC, SOP and TSSOP



Top View

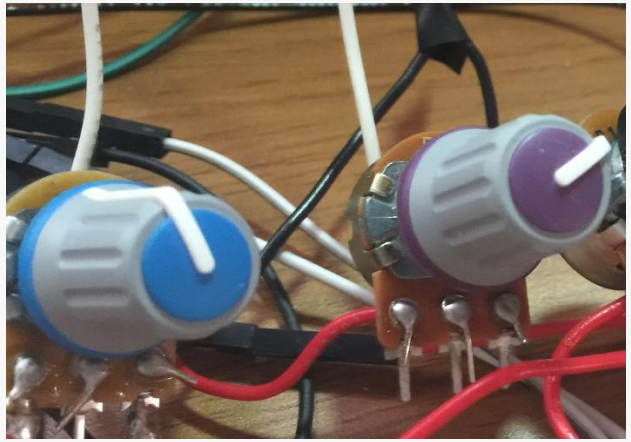
Truth Table

RCK	SCK	$\overline{SCLR}$	$\overline{G}$	Function
X	X	X	H	Q_A thru Q_H = 3-STATE
X	X	L	L	Shift Register cleared $Q_H = 0$
X	↑	H	L	Shift Register clocked $Q_N = Q_{n-1}$, Q_0 = SER
↑	X	H	L	Contents of Shift Register transferred to output latches

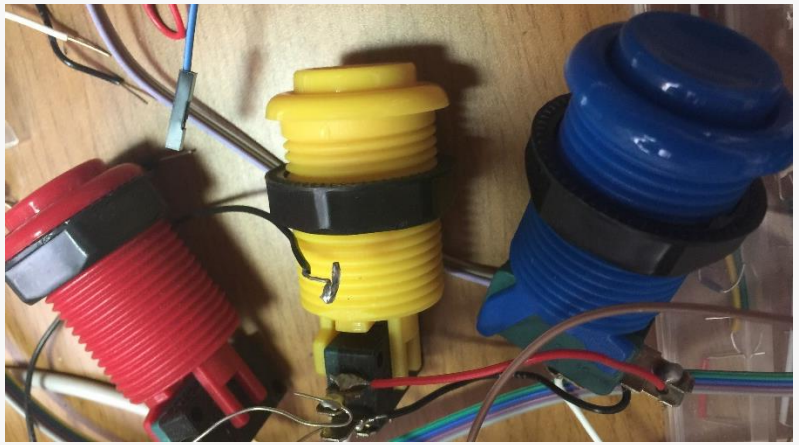
- 템포에 맞춰 현재 재생 위치 표현
- 노트 위치 조절 시 현재 위치 표현

- 다수의 LED를 쉬프트 레지스터를 이용하여 제어
- 래치 핀이 상승 에지일 때 쉬프트 레지스터 값 출력

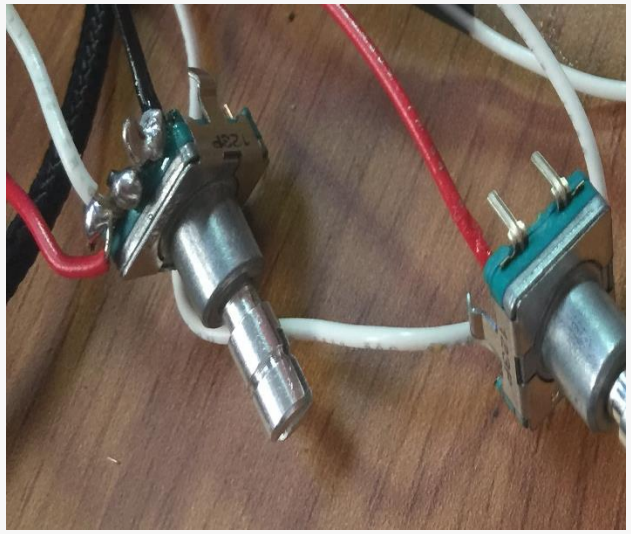
스위치



가변 저항 -> 볼륨, 템포, 리버브, 팬포트 (연속적인 변화에 민감하지 않은 값들)



게임 스위치 -> 채널 선택, 재생, 정지, 통신/솔로 모드 전환



로타리 엔코더 스위치 -> 마디 크기 조절

옥타브 쉬프트, 노트 위치 조절, 음색 선택 등

사용된 연장들

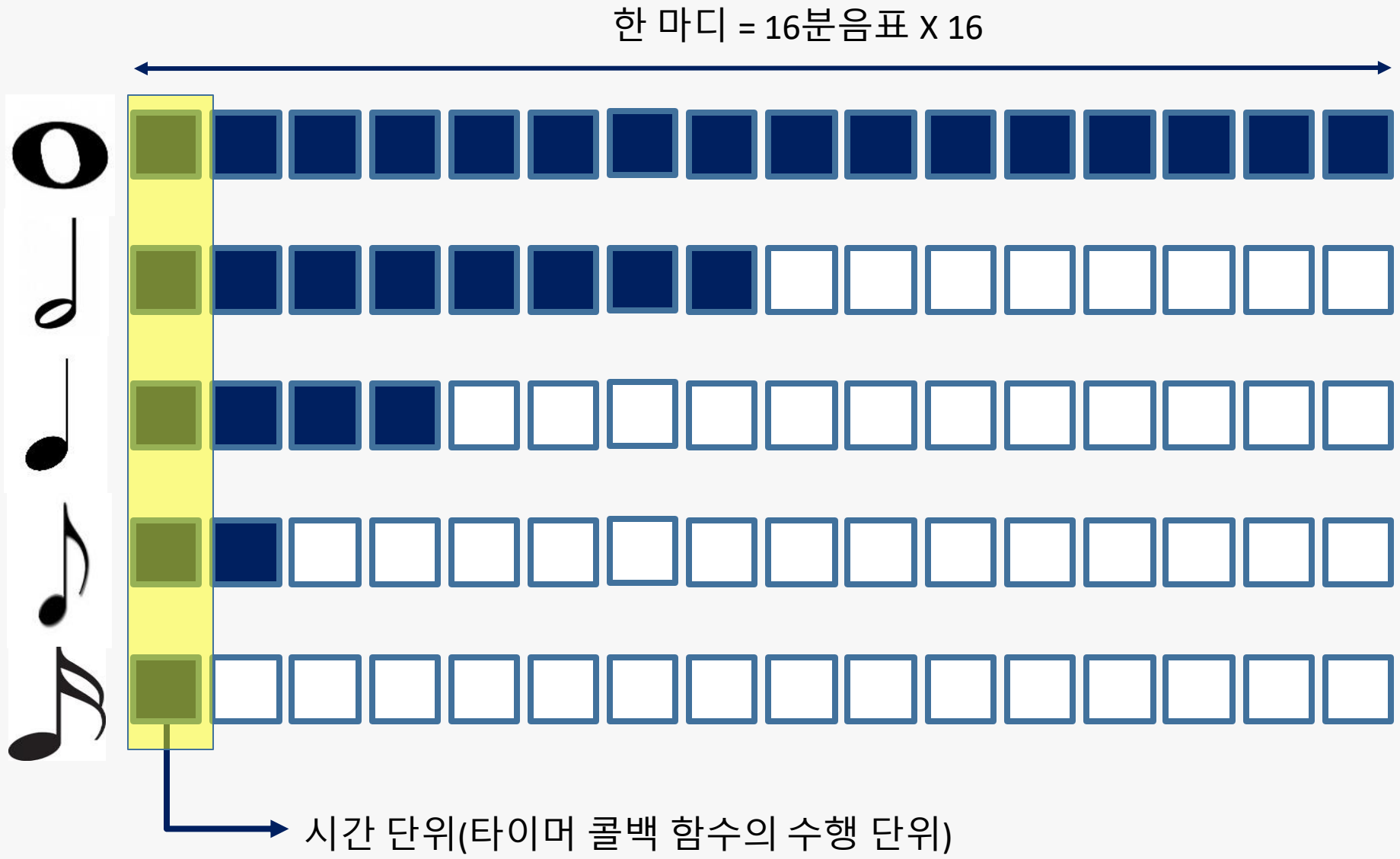
드릴, 글루건, 절연테이프, 인두기, 페이스트, 칼, 가위, 니퍼, 와이어 스트리퍼, 롱노즈 플라이어, 톱..

건반 시연 영상

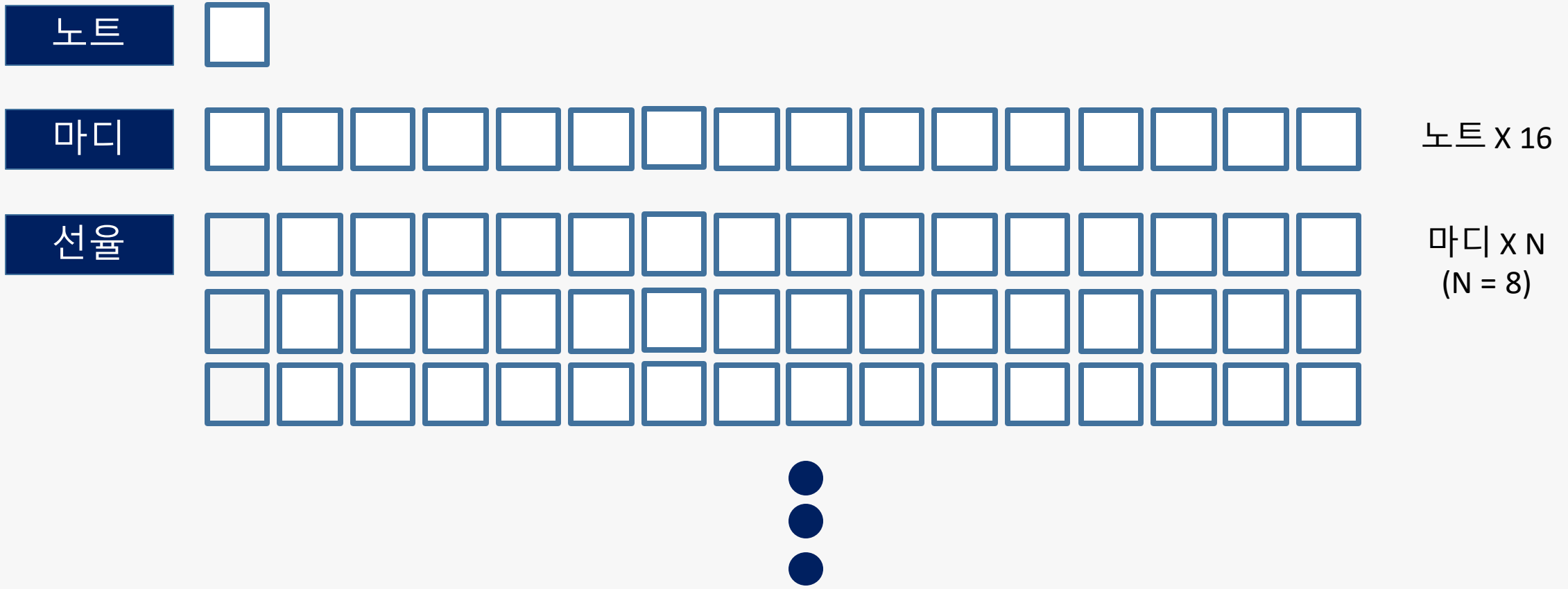
음악 프로그래밍

음악을 프로그래밍 하려면?

노트의 길이



노트(note), 마디(measure), 선율(melody)



미디 라이브러리

미디 보드를 수월하게 다룰 수 있게 만든 C++ 라이브러리

FortySevenEffects / **arduino_midi_library**

```
53  public:
54      inline void sendNoteOn(DataByte inNoteNumber,
55                             DataByte inVelocity,
56                             Channel inChannel);
57
58      inline void sendNoteOff(DataByte inNoteNumber,
59                              DataByte inVelocity,
60                              Channel inChannel);
61
62      inline void sendProgramChange(DataByte inProgramNumber,
63                                    Channel inChannel);
64
65      inline void sendControlChange(DataByte inControlNumber,
66                                    DataByte inControlValue,
67                                    Channel inChannel);
```

Note On 메시지

Note Off 메시지

음색 변경

소리 효과 조절
- 볼륨, 리버브, 팬포트 등

템포와 타이머

- 곡의 빠르기를 어떻게 구현할 것인가?
- 음악은 타이머 콜백 함수가 일정한 주기로 정해진 노트를 재생하는 것
- 이때, 타이머 호출 주기를 변경하여 재생 속도 조절 가능
- 아두이노 소프트웨어 타이머 사용

주기적으로 일을 할 때는 타이머

음색 리스트

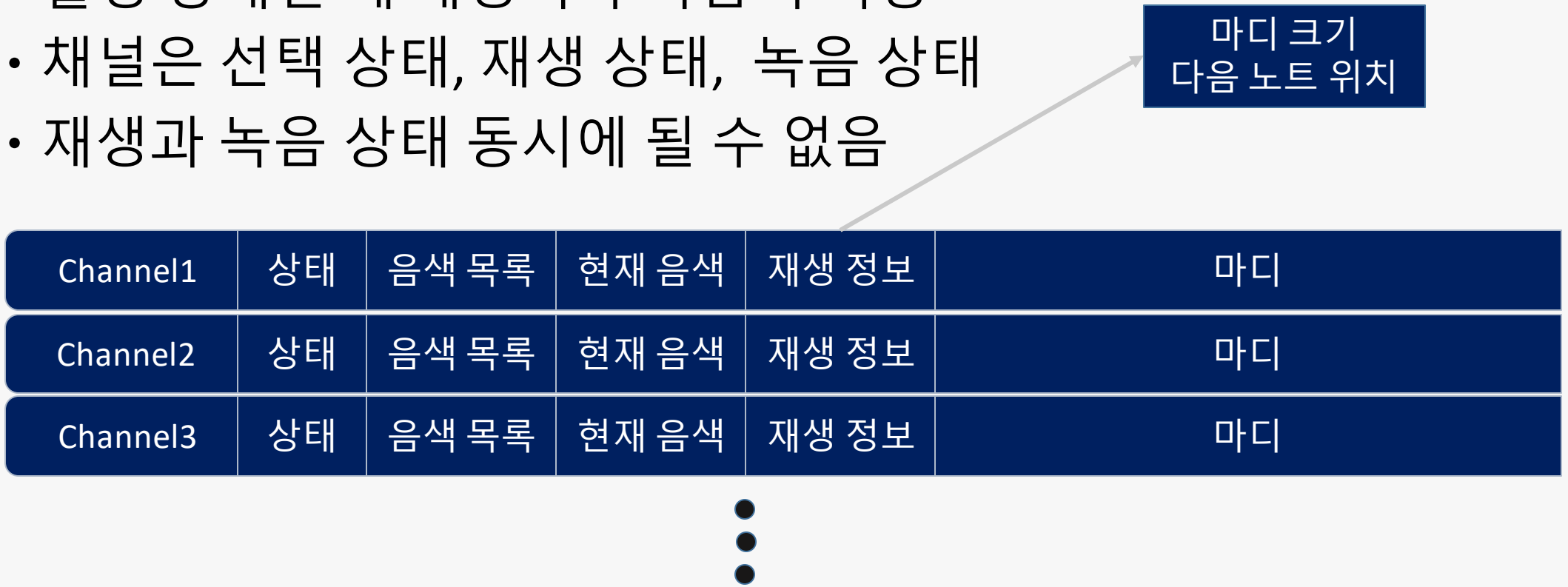
- General MIDI 프로토콜
- 128가지 음색
- 채널 당 8개의 음색 할당

Piano Timbres:	Chromatic Percussion:	Organ Timbres:
1 Acoustic Grand Piano	9 Celesta	17 Hammond Organ
2 Bright Acoustic Piano	10 Glockenspiel	18 Percussive Organ
3 Electric Grand Piano	11 Music Box	19 Rock Organ
4 Honky-tonk Piano	12 Vibraphone	20 Church Organ
5 Rhodes Piano	13 Marimba	21 Reed Organ
6 Chorused Piano	14 Xylophone	22 Accordion
7 Harpsichord	15 Tubular Bells	23 Harmonica
8 Clavinet	16 Dulcimer	24 Tango Accordion

Guitar Timbres:	Bass Timbres:	String Timbres:
25 Acoustic Nylon Guitar	33 Acoustic Bass	41 Violin
26 Acoustic Steel Guitar	34 Fingered Electric Bass	42 Viola
27 Electric Jazz Guitar	35 Plucked Electric Bass	43 Cello
28 Electric Clean Guitar	36 Fretless Bass	44 Contrabass
29 Electric Muted Guitar	37 Slap Bass 1	45 Tremolo Strings
30 Overdriven Guitar	38 Slap Bass 2	46 Pizzicato Strings
31 Distortion Guitar	39 Synth Bass 1	47 Orchestral Harp
32 Guitar Harmonics	40 Synth Bass 2	48 Timpani

내부 자료 구조 – 채널

- 채널 별 음색 할당
- 채널 별 마디 크기
- 채널은 크게 활성 상태, 비활성 상태로 구분
- 활성 상태일 때 재생이나 녹음이 가능
- 채널은 선택 상태, 재생 상태, 녹음 상태
- 재생과 녹음 상태 동시에 될 수 없음



내부 자료 구조 – 드럼 시퀀싱

- 드럼 전용 채널 10(미디 프로토콜 고정)
- 2차원 배열로 행은 드럼 음색 종류를 구분하고 열은 현재 재생 중인 위치를 나타냄
- 타이머 콜백 함수는 호출 될 때마다 현재 칼럼을 재생하고 다음 위치로 이동

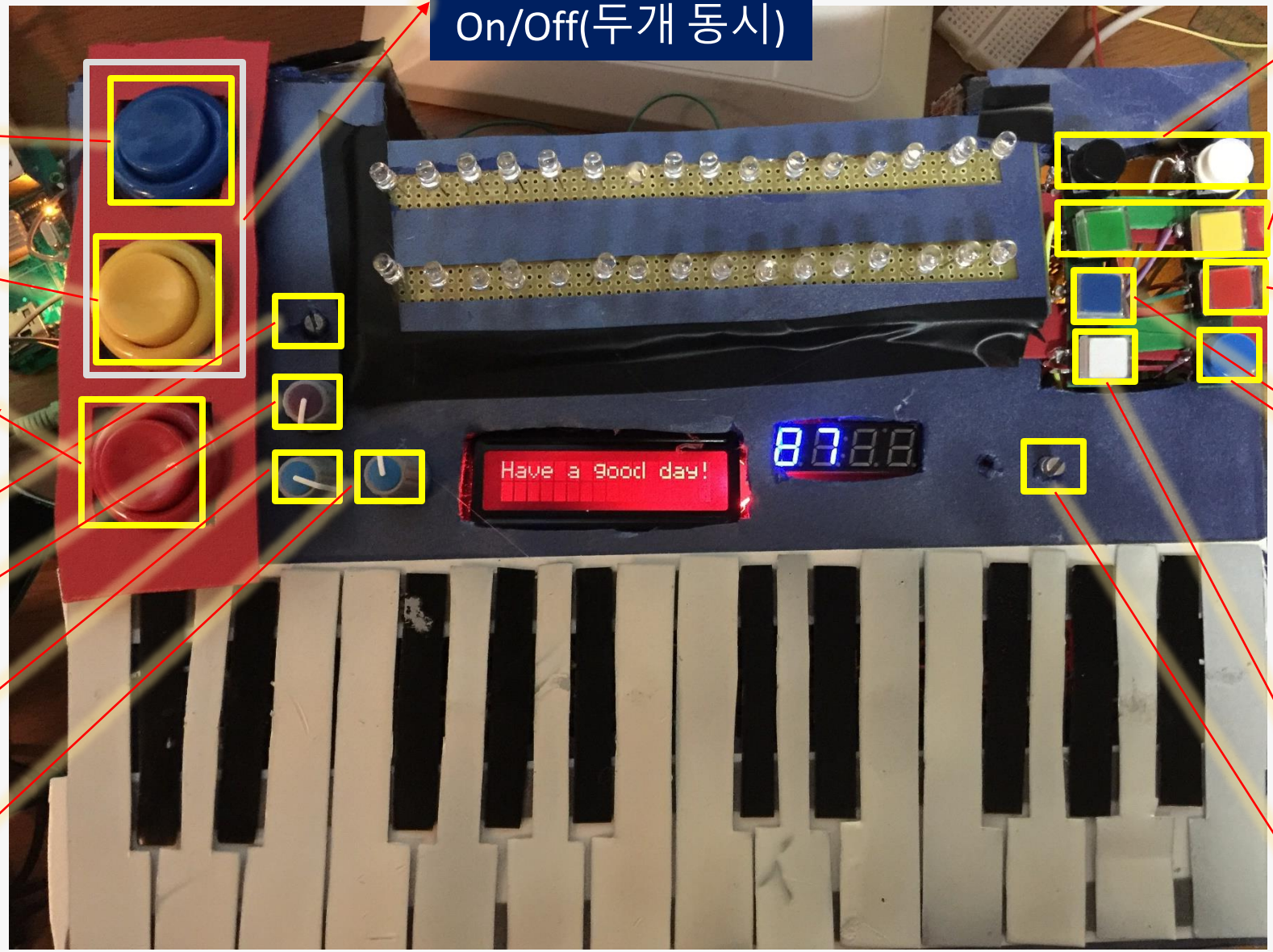
드럼 비트 패턴 예시

드럼 시퀀싱 시연 영상

신디사이저 연주

자유 연주	피아노 처럼 건반 두드리며 연주 가능
녹음	템포에 맞춰 정해진 크기의 마디를 녹음하거나 노트 위치를 조절하며 손으로 직접 수정가능
재생/중지	채널에 있는 마디들 재생/중지
시퀀싱	5개 채널에 있는 각 마디 재생 도중에 마디의 음을 바꾸거나 음색 변경 가능 통신 모드일 때 드럼 머신의 역할을 함

신디사이저 기능



통신 모드
On/Off(두개 동시)

옥타브 쉬프트(증가/감소)

채널 변경

음색 변경(왼쪽/오른쪽)
– 연주
노트 위치 조절 - 녹음

중지

녹음 모드

재생
녹음 방법 변경

악기 목록 수정,
드럼 라인 변경

볼륨 조절

채널 활성화/
비활성화,
노트 제거

템포 조절

마디 초기화

리버브 조절

팬포트 조절

마디 길이 조절

몇가지 난점들 – 건반 갯수

부족한 건반 갯수
(24개-2옥타브)

Roland - JUPITER-50

보통 48~88개의 건반

옥타브 쉬프트
(최소 옥타브 조절)

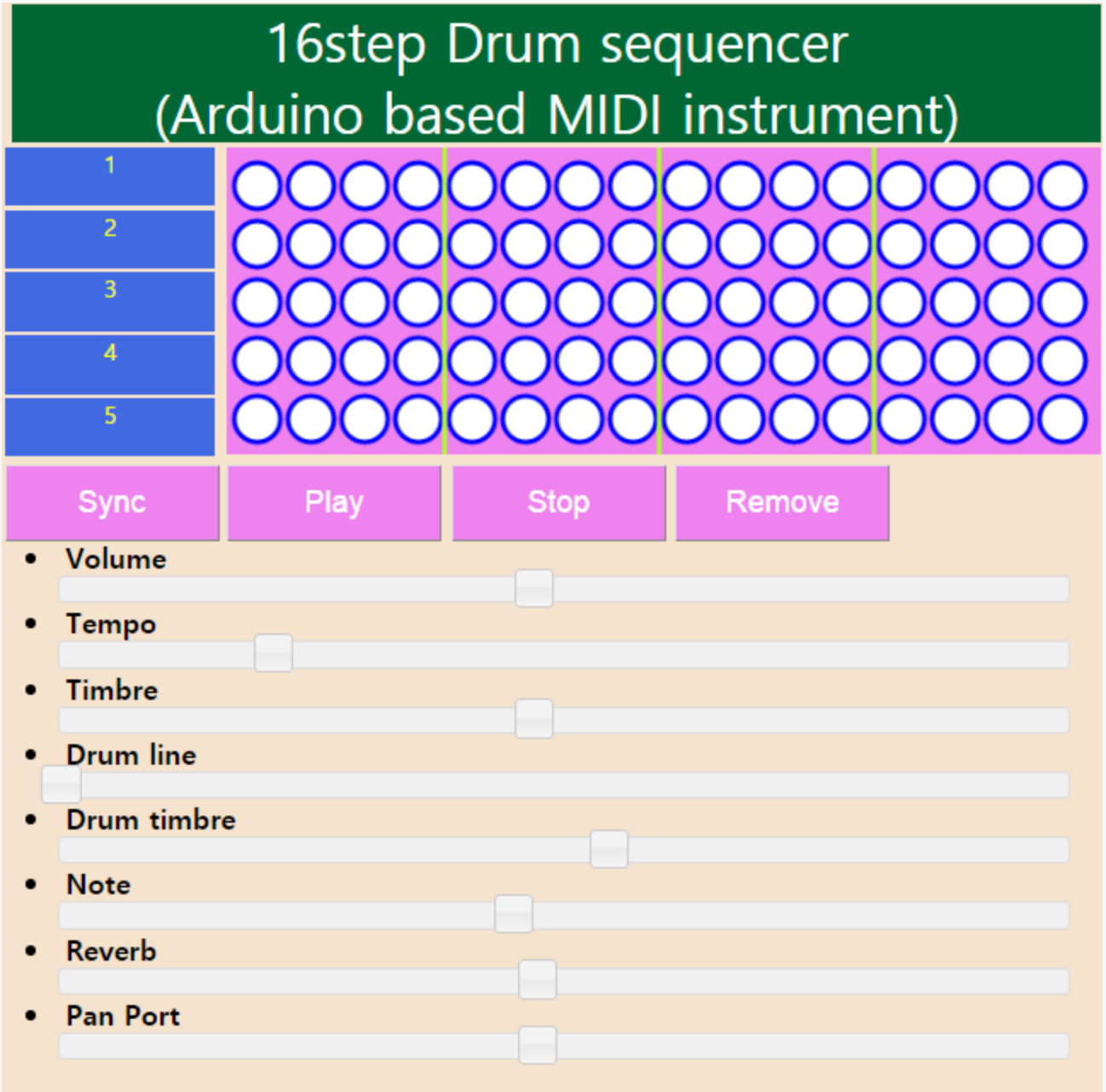
몇가지 난점들 – 버튼 갯수

종류/기능에 따른 별개 버튼 할당 → 핀 부족/배선 부담 → 상황에 따른 서로 다른 기능

novation - nova

웹 클라이언트 & 통신

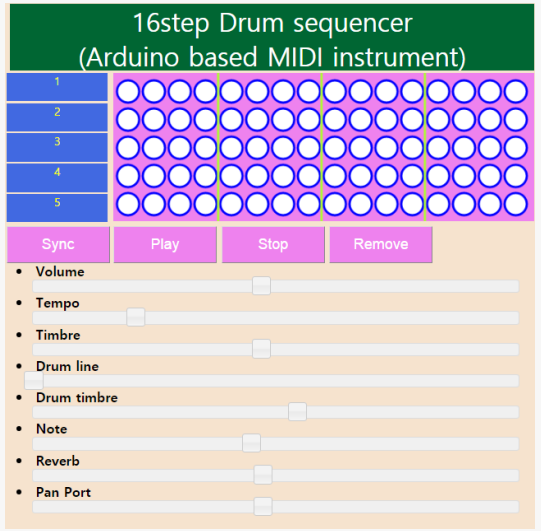
브라우저에서 악기 연주 하기



- HTML5, CSS3, Canvas, Javascript, Jquery 활용
- 웹 소켓 방식으로 값 변화시 실시간 전송
- 원격 16스텝 드럼 시퀀서 구현
- 드럼 라인과 드럼 음색을 선택 후 원하는 위치를 클릭하면 악기 쪽으로 재생 정보 전달
- 볼륨, 리버브, 팬 포트는 값 변화시(슬라이더 움직이는 도중) 즉시 악기에 반영
- 나머지는 슬라이더 움직임이 끝난 후 반영 (통신 부하를 줄이기 위함)

웹 기반 드럼 머신 연주

통신 흐름



브라우저

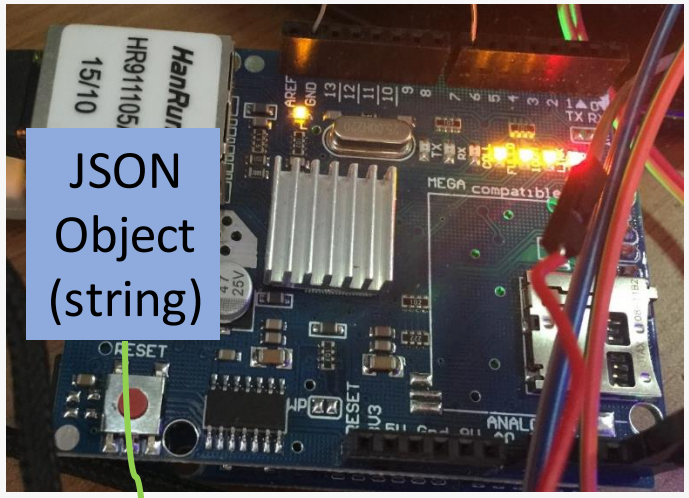
웹 소켓

소켓 이벤트 방출

node

서버

TCP 소켓



이더넷 보드

bblanchon / ArduinoJson

파싱된 JSON Object
{name, value}

I2C

악기

덧붙임

IoT의 정의?

사물들의 소유주의 필요와 이익에 근거하여
점차적으로 인터넷에 연결

어느 정도 인지도 있는 오픈 소스만 표기
->이외에도 MPR121 관련 레지스터 세팅하고 읽고 쓰는 코드나 아두이노 – 노드js – 웹 브라우저 간의 간략한 웹 소켓 통신 구조 코드도 이용

Q&A