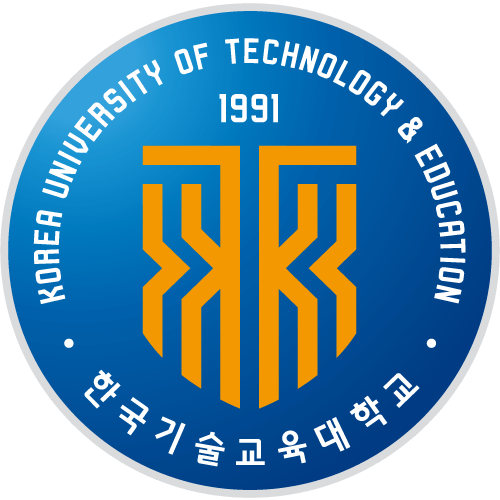
**마이크로프로세서 및 실습**

-비트박스-

|  |  |
| --- | --- |
| 과목 | 마이크로프로세서 및 실습 |
| 분반 | 01 |
| 교수님 | 장경식 교수님 |
| 학번 | 2014136141 |
| 이름 | 송원섭 |



**목차**

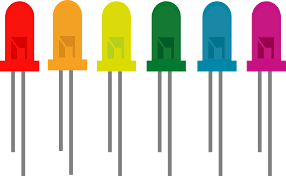
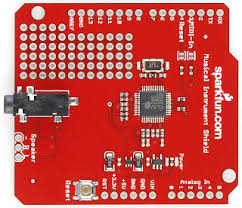
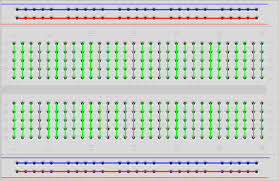
1. 서론--------------------------------------------------------------
2. 작품 설명
3. 기능
4. 본론--------------------------------------------------------------
5. 필요 부품
6. 구성도
7. 핀맵
8. 진행상황
9. 결론--------------------------------------------------------------
10. 소스코드
11. 시연 동영상 링크
12. 참고자료

**서론**

◆작품 설명

-비트박스(BeatBox) 사람의 입으로 디제잉 흉내를 내는 일종의 모사이다.

비트박스를 입으로 박자 맞추면서 배우기가 어려울 때 가벼운 터치로 원하는 비트를 만들고, 따라할 수 있게 녹음해서 재생이 가능하도록 제작.



**출력**



**입력**

◆기능

- 6개 버튼 하나하나에 음을 지정.(1x6)

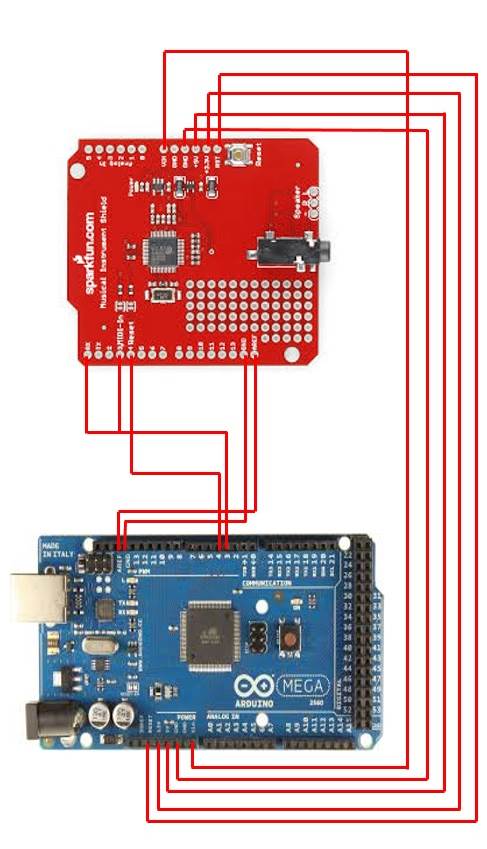
- 녹음, 재생, 악기변경 기능버튼.

- 터치할 때 각 LED를 점등한다.

**본론**

◆필요부품

◆구성도

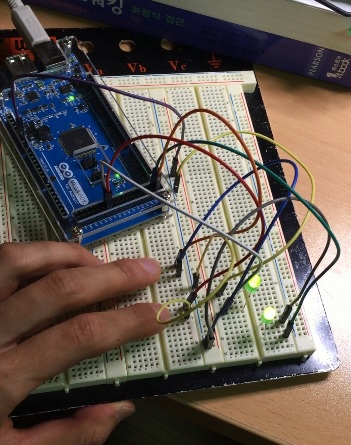
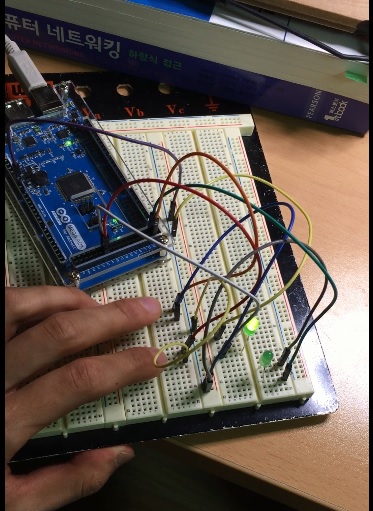


|  |  |  |
| --- | --- | --- |
| **arduino pin** | **module pin** | **Description** |
| A5 | 버튼9 | 악기변경 버튼 |
| A8 | 버튼7 | 녹음 버튼 |
| PWM3 | 뮤직쉴드-3(MIDI IN),RX1 | UART통신 |
| PWM4 | 뮤직쉴드-RESET | 쉴드 초기화 |
| PWM5 | 버튼8 | 느리게 녹음 버튼 |
| PWM6 | 버튼6 | 음재생6 |
| PWM7 | 버튼5 | 음재생5 |
| PWM8 | 버튼4 | 음재생4 |
| PWM9 | 버튼3 | 음재생3 |
| PWM10 | 버튼2 | 음재생2 |
| PWM11 | 버튼1 | 음재생1 |
| 디지털 30 | LED | 풀업 신호에 HIGT출력 |
| 디지털 31 | 음재생버튼 | 풀업신호 |
| 디지털 32 | LED | 풀업 신호에 HIGT출력 |
| 디지털 33 | 음재생버튼 | 풀업신호 |
| 디지털 42 | LED | 풀업 신호에 HIGT출력 |
| 디지털 43 | 음재생버튼 | 풀업신호 |
| 디지털 44 | LED | 풀업 신호에 HIGT출력 |
| 디지털 45 | 음재생버튼 | 풀업신호 |
| 디지털 46 | LED | 풀업 신호에 HIGT출력 |
| 디지털 47 | 음재생버튼 | 풀업신호 |
| 디지털 48 | LED | 풀업 신호에 HIGT출력 |
| 디지털 49 | 음재생버튼 | 풀업신호 |
| 3.3V | 뮤직-3.3V | 아두이노에서 출력되는 전압 |
| 5V | 뮤직-5V | 아두이노에서 출력되는 전압 |
| GND | 뮤직-GND | ground 접지 |
| Reset | 뮤직-RST | 리셋 |
| AREF | 뮤직-AREF | 인터럽트 |
| VIN | 뮤직-VIN | 외부에서 아두이노에 입력전압 |

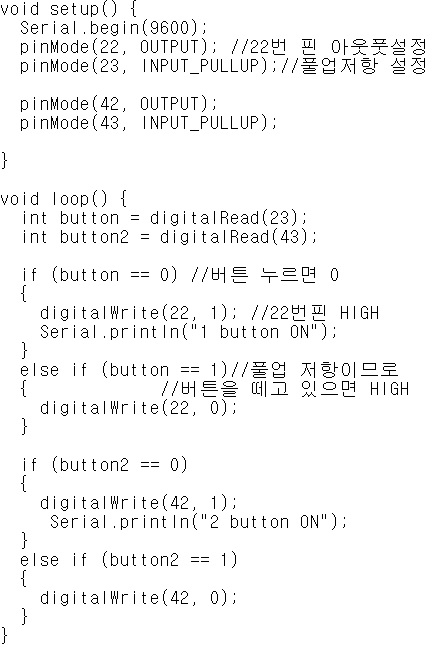
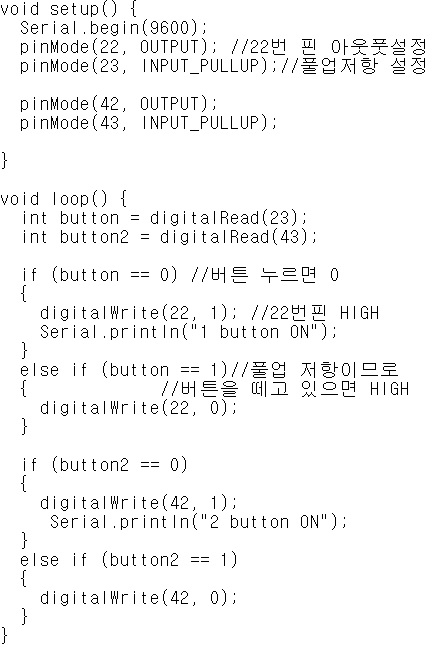
◆핀맵

◆진행상황

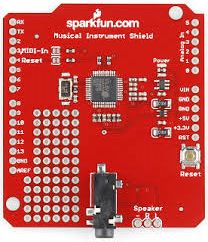
진행(1) Button-LED 입출력 테스트



-풀업저항을 이용해 버튼을 누를시 아웃모드로 설정해 놓은 핀을 LED에 연결해 점등시킴.



진행(2) 뮤직쉴드 연결 및 소스분석



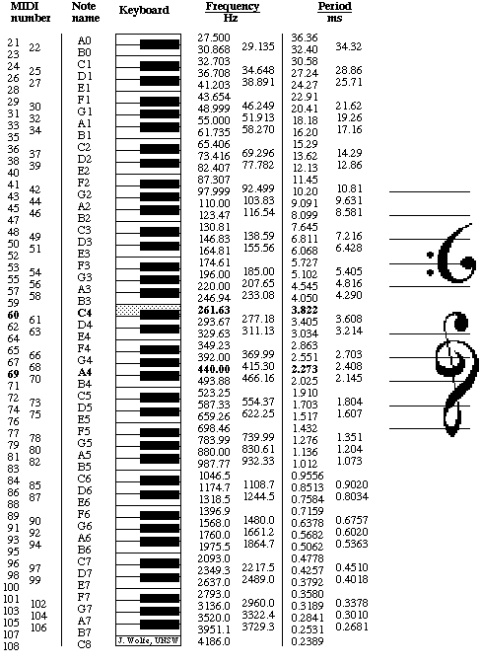
-아두이노와 뮤직쉴드 연결

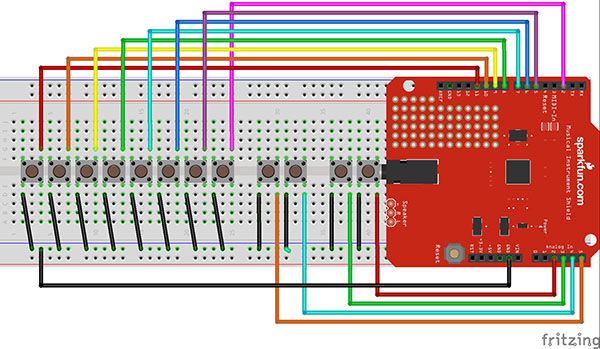
-아두이노와 버튼 연결

뮤직쉴드의 RX와 MIDI-In부분을 아두이노의 TX에 연결을 해줘야 MIDI정보를 받아 출력할 수 있다.

보드는 VS1053 IC로 만들어져 있다.

VS1053은 다양한 음원을 가지고 있어 여러가지 사운드를 출력할 수 있다.





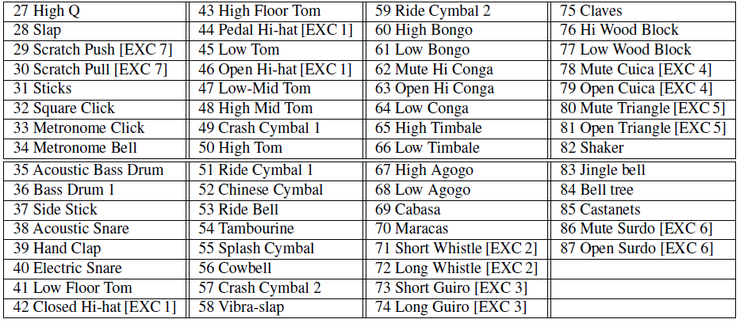
<버튼과 뮤직쉴드 연결 구성도> <음계 MIDI 노트번호>

코코아팹([www.kocoafab.cc](http://www.kocoafab.cc)) 자료참고

진행(3) 뮤직쉴드 소스분석

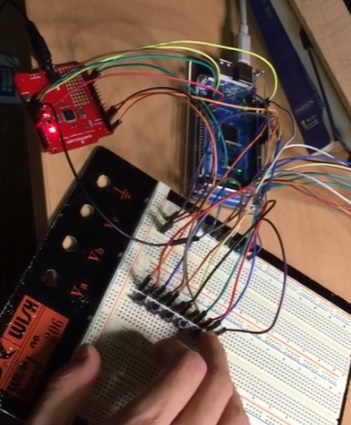
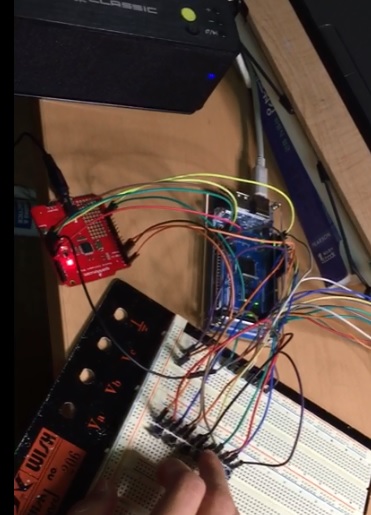


<악기코드표>



<채널9 타악기 코드표>

진행(3) 뮤직쉴드 MIDI출력 테스트



**MIDI프로토콜**

바이트 순서를 가진 프로토콜

Status와 data 2종류

8bit b7=1이면 status b6~b4=command

하위 4bit은 channel

1 000 0000 0x80 Note off 건반 떼다

1 001 0000 0x90 Note on 건반 누르다.

1 100 0000 0xc0 program change 악기변경

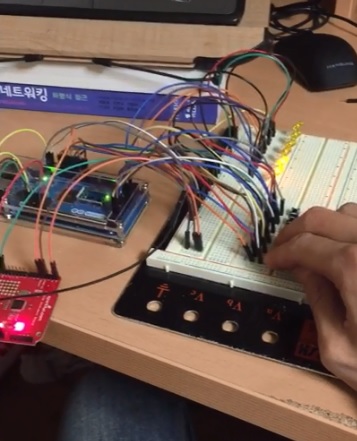
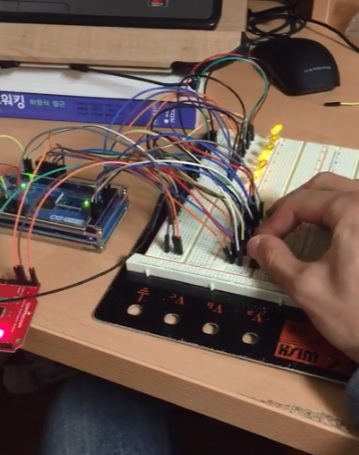
**noteOn(channel, code, velocity)**

**Serial.write(0x90|channel);//command & channel, noteOff는 0x80**

**Serial.write(code);//악기코드**

**Serial.write(velocity);//음의크기**

진행(4) 뮤직쉴드와 LED 출력테스트



- 음 재생을 지정한 버튼에 풀업신호를 LED버튼에 입력해 LED도 동시 출력하게 한다.

(녹음 버튼 구현시 LED 발광도 저장해 같이 출력하고자 한다.)

- 음 재생 버튼(1x6) 과 LED(1x6) 으로 구성.

(녹음, 악기변경 버튼 2개 추가)

주의점: 아두이노의 디지털 핀 이상으로 코드 로직은 맞으나 입출력이 제대로 이루어지지 않을 수 있음.

**-noteOn(byte\_ byte1, byte\_ byte2, byte\_ byte3)**

**해당 음계의 음을 출력하는 함수**

**Byte1 = 상태 명령문과 채널값. 버튼눌림과 타악기 채널9 = 0x90|9**

**Byte2 = 몇번 건반인지(bn1, bn2, bn3…)**

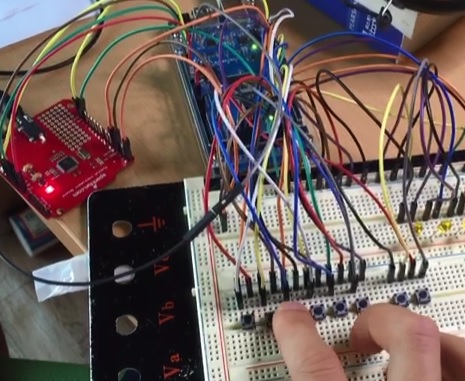
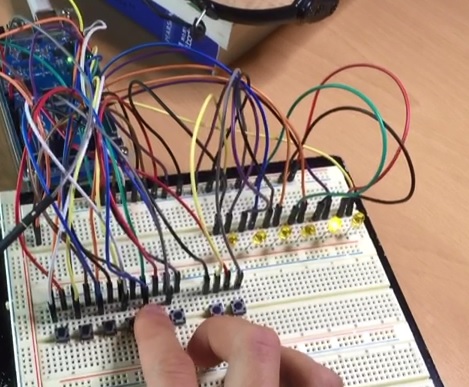
**Byte3 = 소리의 세기를 설정**

**-noteOff(byte\_ byte1, byte\_ byte2, byte\_ byte3)**

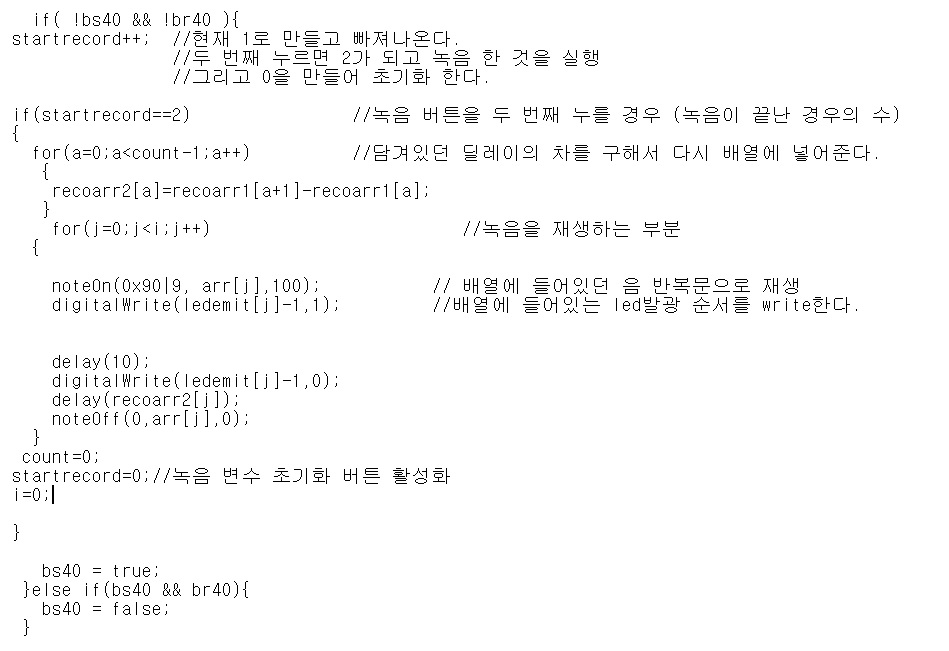
**해당 음계의 음 출력을 중단**

**noteOn과 반대의 역할을 하는 함수**

진행(5) 녹음버튼과 악기변경 버튼 구현



**-녹음버튼 구현**



**-악기변경 버튼 구현:** 단순하게 변수 하나를 두어 악기변경 버튼 누를 시 변수값을 증가시켜 각 버튼 입력시 noteOn 두 번째 파라미터 code에 더해줬다.

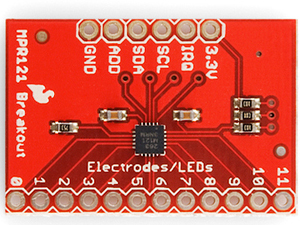
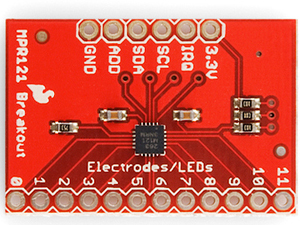
◆구성도

◆소스코드\_버튼과 LED

|  |
| --- |
| void setup() {  Serial.begin(9600);  pinMode(22, OUTPUT); //22번 핀 아웃풋설정  pinMode(23, INPUT\_PULLUP);//풀업저항 설정  pinMode(42, OUTPUT);  pinMode(43, INPUT\_PULLUP);  }  void loop() {  int button = digitalRead(23);  int button2 = digitalRead(43);  if (button == 0) //버튼 누르면 0  {  digitalWrite(22, 1); //22번핀 HIGH  Serial.println("1 button ON");  }  else if (button == 1)//풀업 저항이므로  { //버튼을 떼고 있으면 HIGH  digitalWrite(22, 0);  }    if (button2 == 0)  {  digitalWrite(42, 1);  Serial.println("2 button ON");  }  else if (button2 == 1)  {  digitalWrite(42, 0);  }  } |

◆소스코드\_버튼과 뮤직쉴드

|  |
| --- |
| #include <SoftwareSerial.h>  #include “Adafruit\_VS1053.h”  #define ledbtn1 23 //진동모터 led로 대체 풀업 핀  #define ledbtn2 25 //진동모터 led로 대체 풀업 핀  #define ledbtn3 27 //진동모터 led로 대체 풀업 핀  #define ledbtn4 29 //진동모터 led로 대체 풀업 핀  #define ledbtn5 31 //진동모터 led로 대체 풀업 핀  #define ledbtn6 33 //진동모터 led로 대체 풀업 핀  #define btn1 11 // 버튼1의 아두이노 핀번호 정의  #define btn2 10 // 버튼2의 아두이노 핀번호 정의  #define btn3 9 // 버튼3의 아두이노 핀번호 정의  #define btn4 8 // 버튼4의 아두이노 핀번호 정의  #define btn5 7 // 버튼5의 아두이노 핀번호 정의  #define btn6 6 // 버튼6의 아두이노 핀번호 정의  #define btn7 5 // 버튼7의 아두이노 핀번호 정의  // 3:midi rx , 4:midi reset 아두이노 핀 3번 4번은 이미 사용중  #define btn8 2 // 버튼8의 아두이노 핀번호 정의  //(SoftSerial에서 Rx핀으로 선언되지만 재 세팅 후 버튼용으로 사용)  #define btn9 A5 // 버튼9의 아두이노 핀번호 정의  #define btn10 A4 // 버튼10의 아두이노 핀번호 정의  #define btn11 A3 // 버튼11의 아두이노 핀번호 정의  #define btn12 A2 // 버튼12의 아두이노 핀번호 정의  #define defaultPatch 15 //악기 초기화 버튼 설정 악기번호  SoftwareSerial mySerial(2, 3); //SW시리얼핀 정의 D3이 MIDI신호 전송용, D2는 미사용  byte note = 0; //The MIDI연주될 note(음계)  byte resetMIDI = 4; // VS1053 Reset용 핀  byte ledPin = 13; //MIDI 트래픽 표시용 LED    boolean bs1 = false; // 버튼1의 현재상태(눌림 or 안눌림)  boolean bs2 = false; // 이하, 위와 유사  boolean bs3 = false;  boolean bs4 = false;  boolean bs5 = false;  boolean bs6 = false;  boolean bs7 = false;  boolean bs8 = false;  boolean bs9 = false;  boolean bs10 = false;  boolean bs11 = false;  boolean bs12 = false;  boolean ledbr1;  boolean ledbr2;  boolean ledbr3;  boolean ledbr4;  boolean ledbr5;  boolean ledbr6;  boolean br1; // 버튼1 상태 확인용 입력값 임시저장용  boolean br2; // 이하, 위와 유사  boolean br3;  boolean br4;  boolean br5;  boolean br6;  boolean br7;  boolean br8;  boolean br9;  boolean br10;  boolean br11;  boolean br12;  int patch = 0; //악기 대응, 연주될 악기 종류 (0~127: 기본 128 가지 선택가능)  int bn1 = 60; //버튼1의 note(음계) 가령 "도" 0~127까지 지정가능 (정확한 음계 설정은 MIDI관련정보참고)  int bn2 = 62; //버튼2의 note(음계) 가령 "레"  int bn3 = 64; //버튼3의 note(음계) 가령 "미"  int bn4 = 65; //버튼4의 note(음계) 가령 "파"  int bn5 = 67; //버튼5의 note(음계) 가령 "솔"  int bn6 = 69; //버튼6의 note(음계) 가령 "라"  int bn7 = 71; //버튼7의 note(음계) 가령 "시"  int bn8 = 72; //버튼8의 note(음계) 가령 "도~"  byte byteData;  void setup() {  Serial.begin(31250);  //미디컨트롤을 위한 소프트웨어 시리얼을 준비합니다.  mySerial.begin(31250);  // mySerial2.begin(57600);    //VS1053를 초기화하고 사용할 준비를 합니다.  pinMode(22,OUTPUT);  pinMode(23,INPUT\_PULLUP);  pinMode(24,OUTPUT);  pinMode(25,INPUT\_PULLUP);  pinMode(26,OUTPUT);  pinMode(27,INPUT\_PULLUP);  pinMode(28,OUTPUT);  pinMode(29,INPUT\_PULLUP);  pinMode(30,OUTPUT);  pinMode(31,INPUT\_PULLUP);  pinMode(32,OUTPUT);  pinMode(33,INPUT\_PULLUP);    pinMode(resetMIDI, OUTPUT);  digitalWrite(resetMIDI, LOW);  delay(100);  digitalWrite(resetMIDI, HIGH);  delay(100);    pinMode( btn1, INPUT); // 버튼1 입력용 핀모드를 입력모드로 전환  digitalWrite( btn1, HIGH);  pinMode( btn2, INPUT); // 이하, 위와 유사  digitalWrite( btn2, HIGH);    pinMode( btn3, INPUT);  digitalWrite( btn3, HIGH);  pinMode( btn4, INPUT);  digitalWrite( btn4, HIGH);  pinMode( btn5, INPUT);  digitalWrite( btn5, HIGH);  pinMode( btn6, INPUT);  digitalWrite( btn6, HIGH);  pinMode( btn7, INPUT);  digitalWrite( btn7, HIGH);  pinMode( btn8, INPUT);  digitalWrite( btn8, HIGH);    pinMode( btn9, INPUT);  digitalWrite( btn9, HIGH);  pinMode( btn10, INPUT);  digitalWrite( btn10, HIGH);  pinMode( btn11, INPUT);  digitalWrite( btn11, HIGH);  pinMode( btn12, INPUT);  digitalWrite( btn12, HIGH);  }  void loop() {  br1 = digitalRead(btn1);  ledbr1=digitalRead(ledbtn1);  br2 = digitalRead(btn2);  ledbr2=digitalRead(ledbtn2);  br3 = digitalRead(btn3);  ledbr3=digitalRead(ledbtn3);  br4 = digitalRead(btn4);  ledbr4=digitalRead(ledbtn4);  br5 = digitalRead(btn5);  ledbr5=digitalRead(ledbtn5);  br6 = digitalRead(btn6);  ledbr6=digitalRead(ledbtn6);    br7 = digitalRead(btn7);  br8 = digitalRead(btn8);  br9 = digitalRead(btn9);  br10 = digitalRead(btn10);  br11 = digitalRead(btn11);  br12 = digitalRead(btn12);  if(ledbr1==0) //풀업저항 연결해 버튼누르면 led점등  {  analogWrite(22,100);  }  else digitalWrite(22,0);    if(ledbr2==0) //풀업저항 연결해 버튼누르면 led점등  {  analogWrite(24,100);  }  else digitalWrite(24,0);    if(ledbr3==0) //풀업저항 연결해 버튼누르면 led점등  {  analogWrite(26,100);  }  else digitalWrite(26,0);    if(ledbr4==0) //풀업저항 연결해 버튼누르면 led점등  {  analogWrite(28,100);  }  else digitalWrite(28,0);    if(ledbr5==0) //풀업저항 연결해 버튼누르면 led점등  {  analogWrite(30,200);  }  else digitalWrite(30,0);  if(ledbr6==0) //풀업저항 연결해 버튼누르면 led점등  {  analogWrite(32,200);  }  else digitalWrite(32,0);    if( !bs1 && !br1 ){  noteOn(0x99, 35,100); //북소리  bs1 = true;  }else if(bs1 && br1){  noteOff(0, 35,0);  bs1 = false;  }    if( !bs2 && !br2 ){  noteOn(0x99, 51,100); //북소리  bs2 = true;  }else if(bs2 && br2){  noteOff(0, 51,0);  bs2 = false;  }    if( !bs3 && !br3 ){  noteOn(0x99, 52,100);  bs3 = true;  }else if(bs3 && br3){  noteOff(0, 52,0);  bs3 = false;  }  if( !bs4 && !br4 ){  noteOn(0x99, 54,100);  bs4 = true;  }else if(bs4 && br4){  noteOff(0, 54,0);  bs4 = false;  }  if( !bs5 && !br5 ){  noteOn(0x99, 55,100);  bs5 = true;  }else if(bs5 && br5){  noteOff(0, 55,0);  bs5 = false;  }  if( !bs6 && !br6 ){  noteOn(0x99, 57,100);  bs6 = true;  }else if(bs6 && br6){  noteOff(0, 57,0);  bs6 = false;  }  if( !bs7 && !br7 ){  noteOn(0, bn7,100);  bs7 = true;  }else if(bs7 && br7){  noteOff(0, bn7,0);  bs7 = false;  }  if( !bs8 && !br8 ){  noteOn(0, bn8,100);  bs8 = true;  }else if(bs8 && br8){  noteOff(0, bn8,0);  bs8 = false;  }    if( !bs9 && !br9 ){ //patch up (max:127)  patch++;  if(patch >127) patch = 0;  talkMIDI(0xc0, patch, 0);  bs9 = true;  }else if(bs9 && br9){  bs9 = false;  }  if( !bs10 && !br10 ){ //patch down (min:0)  patch--;  if(patch < 0) patch = 127;  talkMIDI(0xc0, patch, 0);  bs10 = true;  }else if(bs10 && br10){  bs10 = false;  }    if( !bs11 && !br11 ){  //bank 0x78(drum)  talkMIDI(0xb0, 0, 0x78);  talkMIDI(0xb0,20, 0);  talkMIDI(0xc0, patch, 0);  bs11 = true;  }else if(bs11 && br11){  bs11 = false;  }    if( !bs12 && !br12 ){  //bank MSB 0, default instruments  patch = defaultPatch;  talkMIDI(0xb0, 0, 0);  talkMIDI(0xb0,20, 0);  talkMIDI(0xc0, patch, 0);  bs12 = true;  }else if(bs12 && br12){  bs12 = false;  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MIDI LOOPBACK \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  if(Serial.available() > 0)  {  byteData = Serial.read();  mySerial.write( byteData);  }  }  void noteOn(byte channel, byte note, byte attack\_velocity) {  talkMIDI( (0x90 | channel), note, attack\_velocity);  }  //노트 오프 미디 메세지를 송출합니다. 버튼이 눌리지 않은 상태와 같습니다.  void noteOff(byte channel, byte note, byte release\_velocity) {  talkMIDI( (0x80 | channel), note, release\_velocity);  }  void talkMIDI(byte cmd, byte data1, byte data2) {  digitalWrite(ledPin, HIGH);  mySerial.write(cmd );  mySerial.write(data1 );  //모든 명령은 1바이트를 지니며, 모든 cmds는 0xBn보다 2 데이터 바이트를 덜 지닙니다.  //(sort of: http://253.ccarh.org/handout/midiprotocol/)  if( (cmd & 0xF0) <= 0xB0)  mySerial.write(data2 );  digitalWrite(ledPin, LOW);  } |

◆소스코드[](http://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjylbeguq3PAhWDVZQKHTvqDCYQjRwIBw&url=http://m.artrobot.co.kr/product/list.html?cate_no%3D36%26sort_method%3D4%26page%3D6&bvm=bv.133700528,d.dGo&psig=AFQjCNE5HJgFazXr-VXZhvrvlXTD4Aglrg&ust=1474993799704847)\_최종코드

|  |
| --- |
| #include <SoftwareSerial.h>  #include"Adafruit\_VS1053.h"  #define SIZE 100 //녹음용 배열 사이즈  #define ledbtn1 43  #define ledbtn2 45  #define ledbtn3 47  #define ledbtn4 49  #define ledbtn5 31 //led로 대체 풀업 핀  #define ledbtn6 33 //led로 대체 풀업 핀  //악기 3종류  #define vol1 35 //55 75  #define vol2 36 //56 76  #define vol3 42 //62 82  #define vol4 44 //64 84  #define vol5 38 //58 78  #define vol6 40 //60 80  #define btn1 11 // 버튼1의 아두이노 핀번호 정의  #define btn2 10 // 버튼2의 아두이노 핀번호 정의  #define btn3 9 // 버튼3의 아두이노 핀번호 정의  #define btn4 8 // 버튼4의 아두이노 핀번호 정의  #define btn5 7 // 버튼5의 아두이노 핀번호 정의  #define btn6 6 // 버튼6의 아두이노 핀번호 정의  #define btn7 5 // 버튼7의 아두이노 핀번호 정의  #define btn8 2 // 버튼8의 아두이노 핀번호 정의  // 3:midi rx , 4:midi reset 아두이노 핀 3번 4번은 이미 사용중  //(SoftSerial에서 Rx핀으로 선언되지만 재 세팅 후 버튼용으로 사용)  #define btn9 A5 // 버튼9의 아두이노 핀번호 정의  #define btn10 A4 // 버튼10의 아두이노 핀번호 정의  #define btn11 A3 // 버튼11의 아두이노 핀번호 정의  #define btn12 A2 // 버튼12의 아두이노 핀번호 정의  #define defaultPatch 15 //악기 초기화 버튼 설정 악기번호  #define btn40 A8//녹음용  SoftwareSerial mySerial(2, 3); //SW시리얼핀 정의 D3이 MIDI신호 전송용, D2는 미사용  byte note = 0; //The MIDI연주될 note(음계)  byte resetMIDI = 4; // VS1053 Reset용 핀  byte ledPin = 13; //MIDI 트래픽 표시용 LED    boolean bs1 = false; // 버튼1의 현재상태(눌림 or 안눌림)  boolean bs2 = false; // 이하, 위와 유사  boolean bs3 = false;  boolean bs4 = false;  boolean bs5 = false;  boolean bs6 = false;  boolean bs7 = false;  boolean bs8 = false;  boolean bs9 = false;  boolean bs10 = false;  boolean bs11 = false;  boolean bs12 = false;  boolean bs40 = false; //녹음용  boolean ledbr1; //풀업저항에서 오는 값 받을 변수  boolean ledbr2;  boolean ledbr3;  boolean ledbr4;  boolean ledbr5;  boolean ledbr6;  int arr[SIZE]; //녹음용 정수형 배열  long recoarr1[SIZE]; //녹음용 정수형 배열 이곳에 각 버튼이 입력 할 때마다 millis()함수로 값을 넣는다.  long recoarr2[SIZE]; //recoarr1에서 넣고 카운트의 크기가 2 이상이면 버블식으로 앞에부터 차를 구해 이곳에 넣는다. 실질적으로 레코드함수에서 사용할 배열이다.  int count=0;//녹음 버튼 클릭후 각 버튼 누를 때 카운트  int a=0; //녹음 함수 반복문 변수  int i=0;  int j=0;  int startrecord=0;  int ledemit[SIZE]; //녹은된 음을 재생할 때 그 버튼이 보내는 LED발광하게 핀번호를 담는 배열  int tmp=0;  int test=0;//악기변경 변수  boolean br1; // 버튼1 상태 확인용 입력값 임시저장용  boolean br2; // 이하, 위와 유사  boolean br3;  boolean br4;  boolean br5;  boolean br6;  boolean br7;  boolean br8;  boolean br9;  boolean br10;  boolean br11;  boolean br12;  boolean br40; //녹음용  int patch = 0; //악기 대응, 연주될 악기 종류 (0~127: 기본 128 가지 선택가능)  int bn1 = 60; //버튼1의 note(음계) 가령 "도" 0~127까지 지정가능 (정확한 음계 설정은 MIDI 관련정보참고)  int bn2 = 62; //버튼2의 note(음계) 가령 "레"  int bn3 = 64; //버튼3의 note(음계) 가령 "미"  int bn4 = 65; //버튼4의 note(음계) 가령 "파"  int bn5 = 67; //버튼5의 note(음계) 가령 "솔"  int bn6 = 69; //버튼6의 note(음계) 가령 "라"  int bn7 = 71; //버튼7의 note(음계) 가령 "시"  int bn8 = 72; //버튼8의 note(음계) 가령 "도~"  byte byteData;  void setup() {  //PC와 아두이노와의 시리얼통신 준비  Serial.begin(31250);  //미디컨트롤을 위한 소프트웨어 시리얼을 준비합니다.  mySerial.begin(31250);    //VS1053를 초기화하고 사용할 준비를 합니다.  pinMode(42,OUTPUT);  pinMode(43,INPUT\_PULLUP);  pinMode(44,OUTPUT);  pinMode(45,INPUT\_PULLUP);  pinMode(46,OUTPUT);  pinMode(47,INPUT\_PULLUP);  pinMode(48,OUTPUT);  pinMode(49,INPUT\_PULLUP);  pinMode(30,OUTPUT);  pinMode(31,INPUT\_PULLUP);  pinMode(32,OUTPUT);  pinMode(33,INPUT\_PULLUP);    pinMode(resetMIDI, OUTPUT);  digitalWrite(resetMIDI, LOW);  delay(100);  digitalWrite(resetMIDI, HIGH);  delay(100);    pinMode( btn1, INPUT); // 버튼1 입력용 핀모드를 입력모드로 전환  digitalWrite( btn1, HIGH);  pinMode( btn2, INPUT); // 이하, 위와 유사  digitalWrite( btn2, HIGH);  pinMode( btn3, INPUT);  digitalWrite( btn3, HIGH);  pinMode( btn4, INPUT);  digitalWrite( btn4, HIGH);  pinMode( btn5, INPUT);  digitalWrite( btn5, HIGH);  pinMode( btn6, INPUT);  digitalWrite( btn6, HIGH);  pinMode( btn7, INPUT);  digitalWrite( btn7, HIGH);  pinMode( btn8, INPUT);  digitalWrite( btn8, HIGH);  pinMode( btn9, INPUT);  digitalWrite( btn9, HIGH);  pinMode( btn10, INPUT);  digitalWrite( btn10, HIGH);  pinMode( btn11, INPUT);  digitalWrite( btn11, HIGH);  pinMode( btn12, INPUT);  digitalWrite( btn12, HIGH);    pinMode( btn40, INPUT);  digitalWrite( btn40, HIGH);  }  void loop() {  br1 = digitalRead(btn1); //풀업 읽어온다. MIDI출력용  br2 = digitalRead(btn2);  br3 = digitalRead(btn3);  br4 = digitalRead(btn4);  br5 = digitalRead(btn5);  br6 = digitalRead(btn6);  br40=digitalRead(btn40);    ledbr1=digitalRead(ledbtn1); //풀업 읽어온다. LED출력용  ledbr2=digitalRead(ledbtn2);  ledbr3=digitalRead(ledbtn3);  ledbr4=digitalRead(ledbtn4);  ledbr5=digitalRead(ledbtn5);  ledbr6=digitalRead(ledbtn6);    br7 = digitalRead(btn7);  br8 = digitalRead(btn8);  br9 = digitalRead(btn9);  br10 = digitalRead(btn10);  br11 = digitalRead(btn11);  br12 = digitalRead(btn12);  if(ledbr1==0) //풀업저항 연결해 버튼누르면 led점등  {  digitalWrite(42,1);  }  else digitalWrite(42,0);  if(ledbr2==0) //풀업저항 연결해 버튼누르면 led점등  {  digitalWrite(44,1);  }  else digitalWrite(44,0);  if(ledbr3==0) //풀업저항 연결해 버튼누르면 led점등  {  digitalWrite(46,1);  }  else digitalWrite(46,0);  if(ledbr4==0) //풀업저항 연결해 버튼누르면 led점등  {  digitalWrite(48,1);  }  else digitalWrite(48,0);  if(ledbr5==0) //풀업저항 연결해 버튼누르면 led점등  {  digitalWrite(30,1);  }  else digitalWrite(30,0);  if(ledbr6==0) //풀업저항 연결해 버튼누르면 led점등  {  digitalWrite(32,1);  }  else digitalWrite(32,0);      if( !bs1 && !br1 ){  noteOn(0x99, vol1+test,100); // 1번 버튼 북소리  bs1 = true;  if(startrecord==1)// 녹음 버튼이 눌리고 스타트레코드에 1이 들어 있으면 음원 넣기.  {  arr[i]=vol1+test;  i++;  recoarr1[count]=millis();  ledemit[count]=ledbtn6;  count++;  }  }  else if(bs1 && br1){  noteOff(0, vol1+test,0);  bs1 = false;  }  if( !bs2 && !br2 ){  noteOn(0x99, vol2+test,100); //북소리  bs2 = true;  if(startrecord==1)// 녹음 버튼이 눌리고 스타트레코드에 1이 들어 있으면 음원 넣기.  {  arr[i]=vol2+test;  i++;  recoarr1[count]=millis();//이 버튼이 눌렸을 때 시간을 반환한다.  ledemit[count]=ledbtn5;  count++;  }  }else if(bs2 && br2){  noteOff(0, vol2+test,0);  bs2 = false;  }  if( !bs3 && !br3 ){  noteOn(0x99, vol3+test,100);    bs3 = true;  if(startrecord==1)// 녹음 버튼이 눌리고 스타트레코드에 1이 들어 있으면 음원 넣기.  {  arr[i]=vol3+test;  i++;  recoarr1[count]=millis();  ledemit[count]=ledbtn1;  count++;  }  }else if(bs3 && br3){  noteOff(0, vol3+test,0);  bs3 = false;  }  if( !bs4 && !br4 ){  noteOn(0x99, vol4+test,100);  bs4 = true;  if(startrecord==1)// 녹음 버튼이 눌리고 스타트레코드에 1이 들어 있으면 음원 넣기.  {  arr[i]=vol4+test;  i++;  recoarr1[count]=millis();  ledemit[count]=ledbtn2;  count++;  }  }else if(bs4 && br4){  noteOff(0, vol4+test,0);  bs4 = false;  }  if( !bs5 && !br5 ){  noteOn(0x99, vol5,100);  bs5 = true;  if(startrecord==1)// 녹음 버튼이 눌리고 스타트레코드에 1이 들어 있으면 음원 넣기.  {  arr[i]=vol5;  i++;  recoarr1[count]=millis();  ledemit[count]=ledbtn3;  count++;  }  }else if(bs5 && br5){  noteOff(0, vol5+test,0);  bs5 = false;  }    if( !bs6 && !br6 ){  noteOn(0x99, vol6+test,100);  bs6 = true;  if(startrecord==1)// 녹음 버튼이 눌리고 스타트레코드에 1이 들어 있으면 음원 넣기.  {  arr[i]=vol6+test;  i++;  recoarr1[count]=millis();  ledemit[count]=ledbtn4;  count++;  }    }else if(bs6 && br6){  noteOff(0, vol6+test,0);  bs6 = false;  }  /\*  if( !bs7 && !br7 ){  noteOn(0, bn7,100);  bs7 = true;  }else if(bs7 && br7){  noteOff(0, bn7,0);  bs7 = false;  }  if( !bs8 && !br8 ){  noteOn(0, bn8,100);  bs8 = true;  }else if(bs8 && br8){  noteOff(0, bn8,0);  bs8 = false;  }  \*/    if( !bs9 && !br9 ){ //patch up (max:127) //악기변경  if(test!=40)  {  test=test+20;  }  else  test=0;    bs9 = true;  }  else if(bs9 && br9){  bs9 = false;  }  /\*  if( !bs10 && !br10 ){ //patch down (min:0) //악기변경  patch--;  if(patch < 0) patch = 127;  talkMIDI(0xc0, patch, 0);  bs10 = true;  }else if(bs10 && br10){  bs10 = false;  }  \*/  if( !bs11 && !br11 ){  //bank 0x78(drum)  talkMIDI(0xb0, 0, 0x78);  talkMIDI(0xb0,20, 0);  talkMIDI(0xc0, patch, 0);  bs11 = true;  }else if(bs11 && br11){  bs11 = false;  }    if( !bs12 && !br12 ){  //bank MSB 0, default instruments  patch = defaultPatch;  talkMIDI(0xb0, 0, 0);  talkMIDI(0xb0,20, 0);  talkMIDI(0xc0, patch, 0);  bs12 = true;  }else if(bs12 && br12){  bs12 = false;  }  //녹음  if( !bs40 && !br40 ){  startrecord++; //현재 1로 만들고 빠져나온다.  //두 번째 누르면 2가 되고 녹음 한 것을 실행  //그리고 0을 만들어 초기화 한다.    if(startrecord==2) //녹음 버튼을 두 번째 누를 경우 (녹음이 끝난 경우의 수)  {  for(a=0;a<count-1;a++) //담겨있던 딜레이의 차를 구해서 다시 배열에 넣어준다.  {  recoarr2[a]=recoarr1[a+1]-recoarr1[a];  }  for(j=0;j<i;j++) //녹음을 재생하는 부분  {  noteOn(0x90|9, arr[j],100); // 배열에 들어있던 음 반복문으로 재생  digitalWrite(ledemit[j]-1,1); //배열에 들어있는 led발광 순서를 write한다.  delay(10);  digitalWrite(ledemit[j]-1,0);  delay(recoarr2[j]);  noteOff(0,arr[j],0);  }  count=0;  startrecord=0;//녹음 변수 초기화 버튼 활성화  i=0;  }  bs40 = true;  }  else if(bs40 && br40){  bs40 = false;  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MIDI LOOPBACK \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  if(Serial.available() > 0)  {  byteData = Serial.read();  mySerial.write( byteData);  }  }  //노트 온 미디 메세지를 송출합니다. 버튼이 눌린상태와 같습니다.  //채널 범위는 0-15 입니다.  void noteOn(byte channel, byte note, byte attack\_velocity) {  talkMIDI( (0x90 | channel), note, attack\_velocity);  }  //노트 오프 미디 메세지를 송출합니다. 버튼이 눌리지 않은 상태와 같습니다.  void noteOff(byte channel, byte note, byte release\_velocity) {  talkMIDI( (0x80 | channel), note, release\_velocity);  }  void talkMIDI(byte cmd, byte data1, byte data2) {  digitalWrite(ledPin, HIGH);  mySerial.write(cmd );  mySerial.write(data1 );  //모든 명령은 1바이트를 지니며, 모든 cmds는 0xBn보다 2 데이터 바이트를 덜 지닙니다.  //(sort of: http://253.ccarh.org/handout/midiprotocol/)  if( (cmd & 0xF0) <= 0xB0)  mySerial.write(data2 );  digitalWrite(ledPin, LOW);  } |

◆유튜브 시연 동영상

[**https://youtu.be/o\_LfAmR6RoA**](https://youtu.be/o_LfAmR6RoA)

**(버튼LED테스트)**

[**https://youtu.be/gG4p59BLKFI**](https://youtu.be/gG4p59BLKFI)

**(뮤직쉴드 테스트)**

[**https://youtu.be/H0VT1EikWuo**](https://youtu.be/H0VT1EikWuo)

**(뮤직쉴드 LED테스트)**

[**https://youtu.be/kM8YaYWsJkI**](https://youtu.be/kM8YaYWsJkI)

**(뮤직쉴드 기능구현)**

◆참고자료 및 링크

**허경용 저(2016)『따라 하면서 배우는 아두이노 : 아두이노 메가2560으로 마이크로컨트롤러 시작하기』서울: 한빛아카데미**

**김진환, 장성용 저(2016) 『예제로 쉽게 배우는 아두이노』파주: 생능출판**

**http://cafe.naver.com/arduinostory**

**(아두이노 스토리)**

[**https://learn.adafruit.com**](https://learn.adafruit.com)

**(adafruit site)**

[**https://www.sparkfun.com/**](https://www.sparkfun.com/)

**(sparkfun site)**

[**http://blog.naver.com/roboholic84**](http://blog.naver.com/roboholic84)

**(메카솔루션 오픈랩)**

[**http://blog.vctec.co.kr/**](http://blog.vctec.co.kr/)

**(가치창조기술 오픈랩)**

**www.kocoafab.cc/**

**(코코아팹)**

[**www.arduino.cc**](http://www.arduino.cc)

**(아두이노)**