

# 4X4 KEY MATRIX 읽기

- 기초 학습
  - Random Number
- 4X4 Key matrix 읽기
  - 개 요
  - 예 제
- 응용 실습

Basic learning

---

# 기초 학습

- randomSeed(unsigned int seed)
  - 허위(Pseudo) 난수 생성기를 초기화하는 함수
  - randomSeed(analogRead(0))의 형태로 사용
  - 허위 난수의 시작점이 되는 시드 결정이 중요
    - 시드는 외부 회로가 연결되지 않은 아날로그 핀으로부터 무작위 값을 입력 받아 사용하는 것이 가장 간단한 방법
- random(long min, long max)
  - 의사 난수를 생성하는 함수
  - random(max), random(min, max)의 형태로 사용
  - 난수의 생성 범위는 min ~ max-1까지의 값
  - min 값을 지정하지 않을 경우 최소값은 0으로 설정

---

# 4X4 KEY MATRIX 읽기

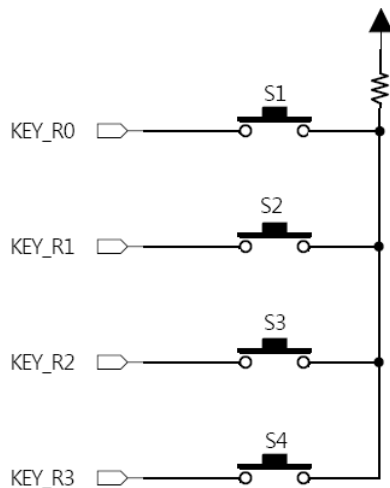
- 목적

- 사용자의 입력을 받아 들이기 위한 장치
- 전화기, 컴퓨터, 핸드폰 등 많은 가전제품에서 사용
- 키 패드 인터페이스 방식은 정적 인터페이스(Static Interface) 방식, 동적 인터페이스(Dynamic Interface) 방식의 두가지로 나뉘어짐

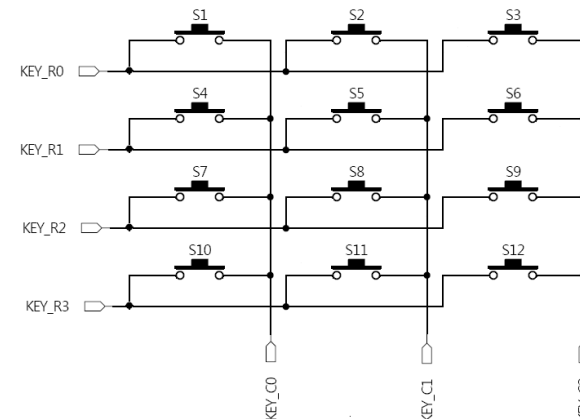
- 관련이론

- 정적 인터페이스 방식
  - 키를 포트에 일대 일로 연결하여 입력을 검사하는 방식
  - 하드웨어 구성이 간편하고, 소프트웨어 구현이 쉬운 장점
  - 많은 키 입력이 필요할 때는 키의 개수 만큼의 포트를 사용하기 때문에 포트 사용이 비효율적
- 동적 인터페이스 방식
  - 키 스위치를 매트릭스 형태로 구성하여 출력상태에 따른 입력 값을 비교하여 키 입력을 판별하는 방법
  - 적은 포트로 여러 개의 키를 제어
  - 눌린 스위치의 검출 방법이 복잡

- 정적 인터페이스 그림에서 S1이 열려있을 때 KEY\_R0 신호는 '0'를 유지하고 있다가, S1을 누르면 신호가 전원과 연결되어 '1'의 값이 됨
- 동적 인터페이스 그림에서 KEY\_R0에 1신호가 가해지고, KEY\_C0이 '1'이라면 우리는 스위치 S1이 눌러졌다는 것을 감지
  - KEY\_R0~3 에 "1000" 이 인가되었는데, KEY\_C0~2의 값이 "100"이라면 스위치 S1이 눌러졌음을 감지
  - KEY\_R0~3=0010 이고, KEY\_C0~2=010이면 S8이 눌러졌음을 감지



정적 인터페이스(static interface)

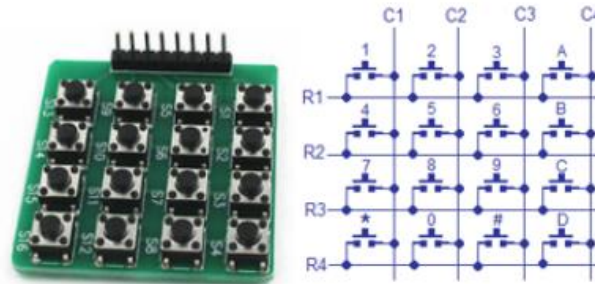


동적 인터페이스(Dynamic interface)

## - 개요

4x4 키패드 모듈에 대해서 알아 보겠습니다.

8개의 디지털 출력, 입력을 사용해서 총 16개의 버튼 입력을 사용할 수 있습니다. 키패드모듈의 모양과 내부 회로도는 아래와 같습니다.

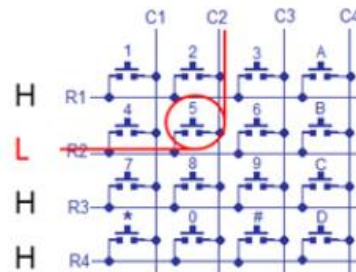


내부는 matrix 형태로 Row열과 Column이 스위치로 연결이 되어있습니다.

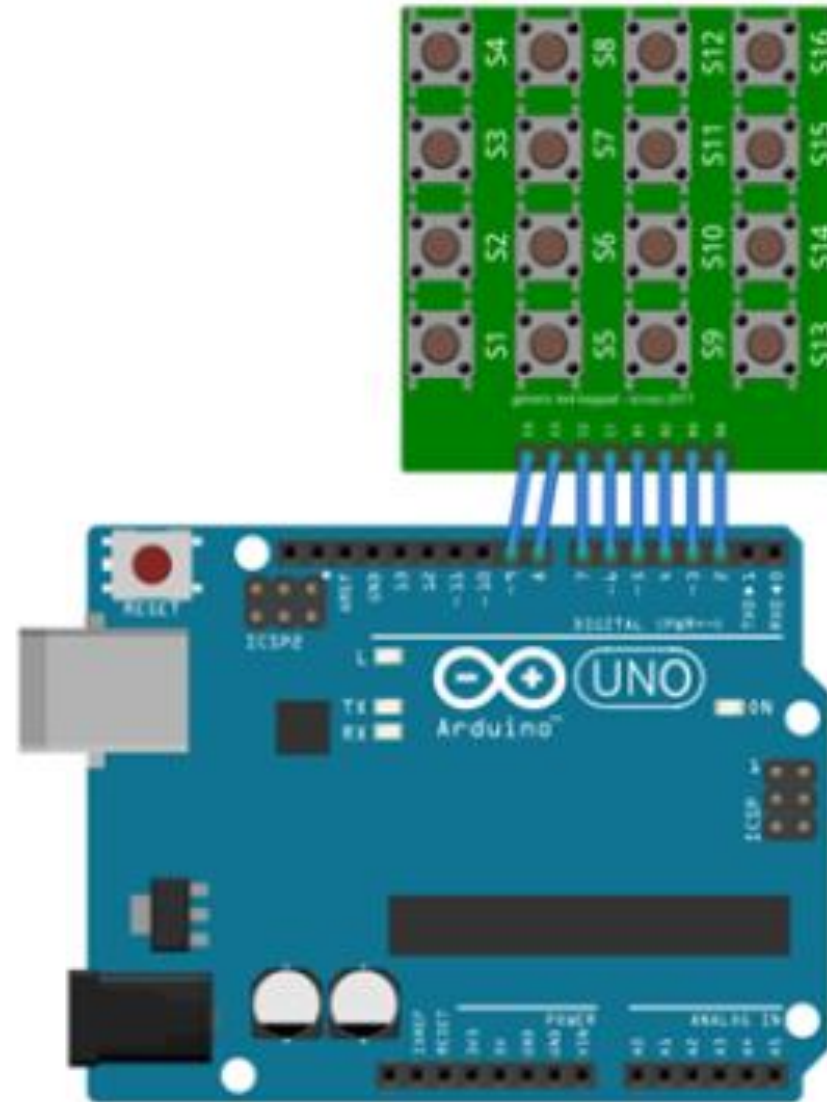
Row 라인을 출력으로 선택하고 Column라인을 입력으로 설정을 하고 출력 포트를 순차적으로 Active Low를 주고 입력라인을 체크하는 방식을 사용합니다.

5번을 인식하는 과정을 그림으로 표시하면 아래와 같습니다.

- 1) R1,R3,R4는 High를 유지하고 R2를 LOW 신호를 출력합니다.
- 2) C1,C2,C3,C4를 체크하여 C2가 Low 인식이 되면 5버튼이 눌렸다고 판단을 합니다.







```
#define Rows 4
#define Cols 4
char keys[Rows][Cols] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'} };

const int RowPinTbl[Rows] = {5, 4, 3, 2};
const int ColPinTbl[Cols] = {6, 7, 8, 9};

void setup() {
    Serial.begin(9600); // init Serial
    for(int i=0; i<Rows; i++)
    {
        pinMode(RowPinTbl[i], INPUT_PULLUP);
    }
    for(int j=0; j<Cols; j++) {
        pinMode(ColPinTbl[j], OUTPUT);
        digitalWrite(ColPinTbl[j], HIGH); // 초기값 HIGH
    }
}
```

```
void loop() { // 버튼 인식
    for(int j=0; j<Cols; j++) {
        digitalWrite(ColPinTbl[j], LOW); // Col 라인 Active Low
        for(int i=0; i<Rows; i++) {
            if( digitalRead(RowPinTbl[i]) == LOW ) { // check Row 라인
                Serial.print("row=");
                Serial.print(i);
                Serial.print(", column=");
                Serial.println(j);
                Serial.print(", Key Number=");
                Serial.println(keys[i][j]);
            }
        }
        digitalWrite(ColPinTbl[j], HIGH); // Col 라인 다시 High
    }
}
```

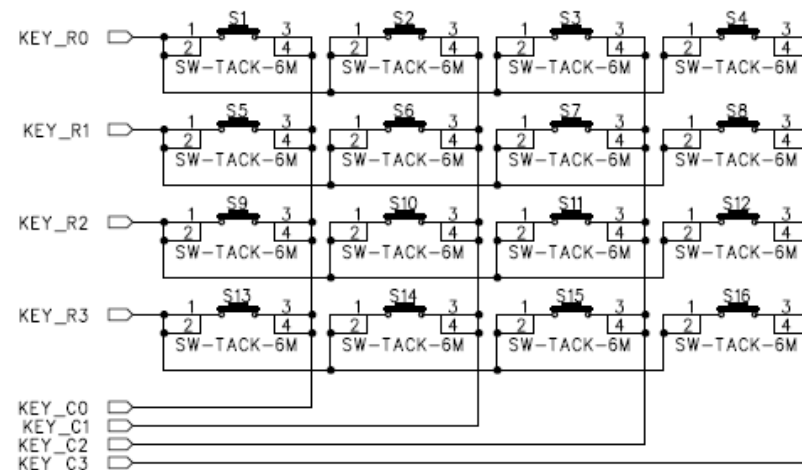
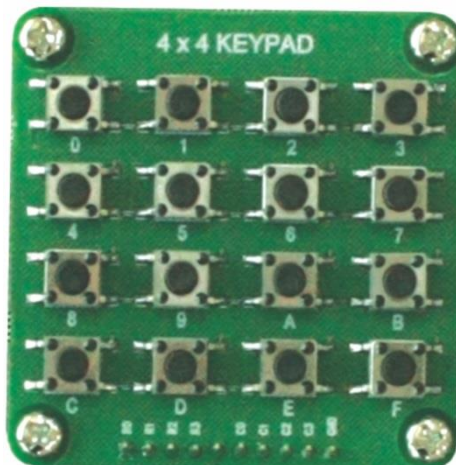
---

# 응용 실습

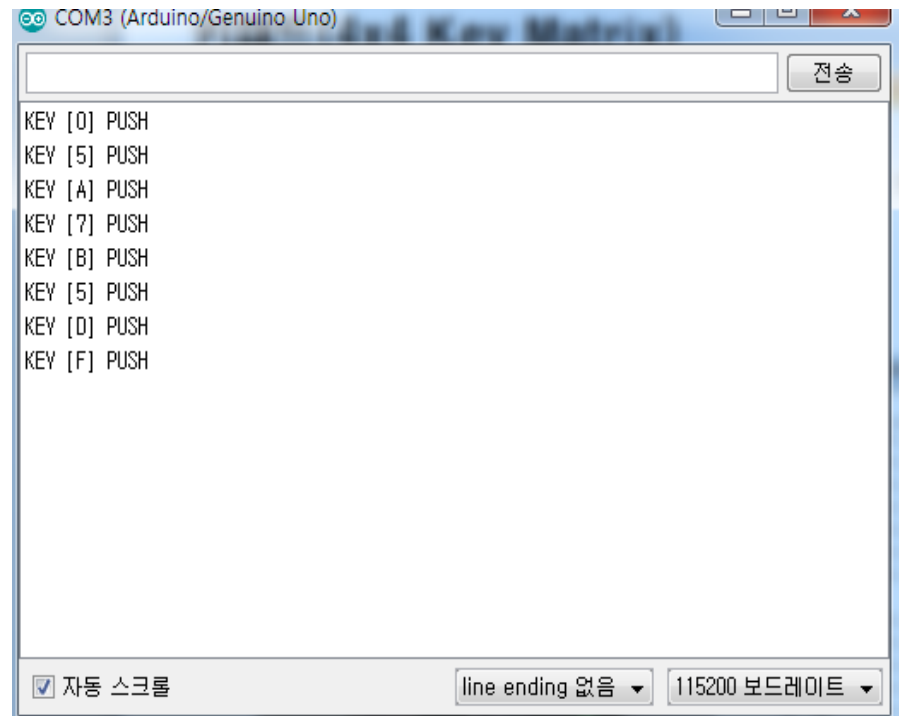
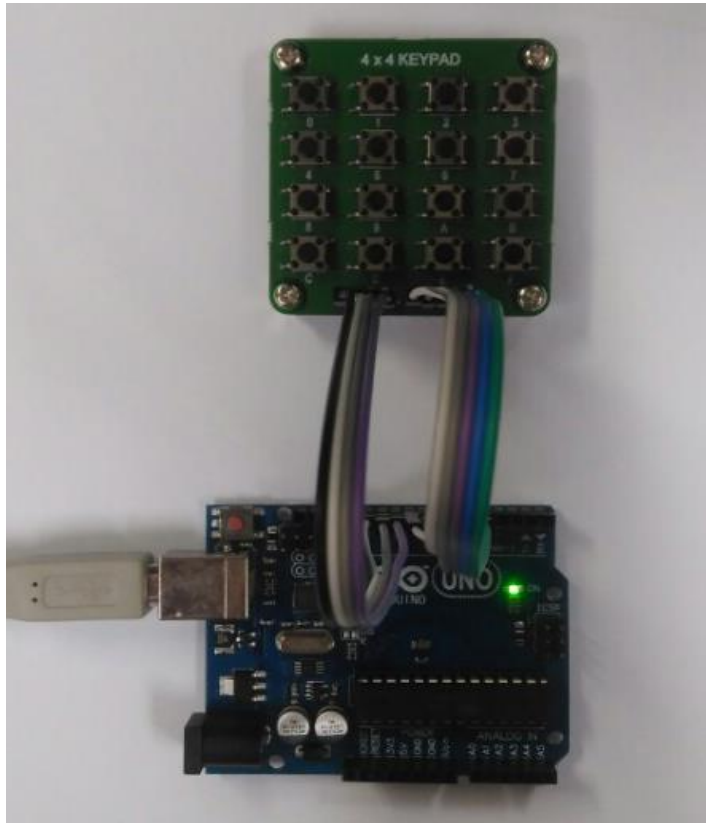
- 버튼을 눌러서 음악연주하기

- 하드웨어 설명
  - 신호에 따른 스위치 눌림 검출표

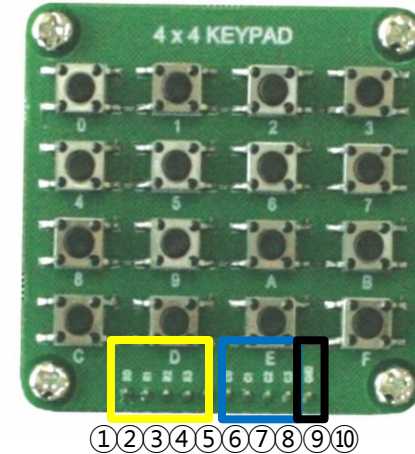
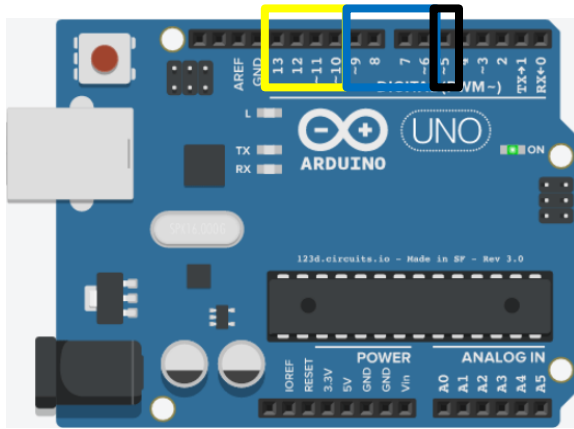
	KEY_C0~3=1000	KEY_C0~3=0100	KEY_C0~3=0010	KEY_C0~3=0001
KEY_R0~3=1000	SW1(0)	SW2(1)	SW3(2)	SW4(3)
KEY_R0~3=0100	SW5(4)	SW6(5)	SW7(6)	SW8(7)
KEY_R0~3=0010	SW9(8)	SW10(9)	SW11(A)	SW12(B)
KEY_R0~3=0001	SW13(C)	SW14(D)	SW15(E)	SW16(F)



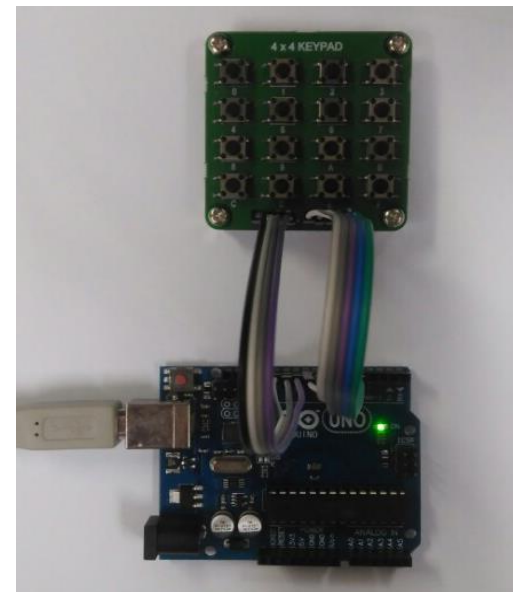
- 예제
  - 임의의 스위치를 클릭하면 시리얼 모니터에 해당 번호가 출력하시오.



## ● Uno Board와 Key Matrix 모듈 연결



Name	Key Matrix Pin Number	Arduino Pin Number
R0	①	13
R1	②	12
R2	③	11
R3	④	10
-	⑤	-
C0	⑥	9
C1	⑦	8
C2	⑧	7
C3	⑨	6
GND	⑩	5





- 프로그램 설명
  - 초기화 구문

```
for(int i=0; i<4; i++)  
{  
    pinMode(aPinRow[i], OUTPUT);  
    pinMode(aPinColumn[i], INPUT);  
}  
pinMode(pinGnd, OUTPUT);  
digitalWrite(pinGnd, LOW);
```

## ● 프로그램 설명

### ▪ loop 구문

이중 for문을 이용해 aPinRow를 HIGH 상태로 두고 aPinColumn을 Read하여 Key의 번호를 읽어옴

```
for(int i=0; i<4; i++) {
    digitalWrite(aPinRow[i], HIGH);
    nRow = 0;
    nRow = i * 4;
    for(int j=0; j<4; j++) {
        nColumn = 0;
        nColumn += j;
        if(digitalRead(aPinColumn[j])) {
            nNumber = nRow + nColumn;
            Serial.print("KEY [");
            Serial.print(nNumber, HEX);
            Serial.println("] PUSH");
        }
    }
    digitalWrite(aPinRow[i], LOW);
    delay(100);
}
```

Column Row	j = 0	j = 1	j = 2	j = 3
i = 0	0	1	2	3
i = 1	4	5	6	7
i = 2	8	9	A(10)	B(11)
i = 3	C(12)	D(13)	E(14)	F(15)

- 전체 소스코드

```
1. int aPinRow[4] = {13, 12, 11, 10};
2. int aPinColumn[4] = {9, 8, 7, 6};
3. int pinGnd = 5;
4.
5. void setup() {
6.     Serial.begin(115200);
7.
8.     for(int i=0; i<4; i++)
9.     {
10.         pinMode(aPinRow[i], OUTPUT);
11.         pinMode(aPinColumn[i], INPUT);
12.     }
13.     pinMode(pinGnd, OUTPUT);
14.     digitalWrite(pinGnd, LOW);
15. }
```

- 전체 소스코드

```
16. void loop() {
17.   int nRow, nColumn, nNumber;
18.   for(int i=0; i<4; i++) {
19.     digitalWrite(aPinRow[i], HIGH);
20.     nRow = 0;
21.     nRow = i*4;
22.     for(int j=0; j<4; j++) {
23.       nColumn = 0;
24.       nColumn += j;
25.       if(digitalRead(aPinColumn[j])) {
26.         nNumber = nRow + nColumn;
27.         Serial.print("KEY [");
28.         Serial.print(nNumber, HEX);
29.         Serial.println("] PUSH");
30.       }
31.     }
32.     digitalWrite(aPinRow[i], LOW);
33.     delay(100);
34.   }
35. }
```

Application practice

---

**응용 실습**

- 응용 문제
  - 도어락과 같이 랜덤함수를 사용하여 임의의 비밀번호를 초기 설정하고, 숫자(비밀번호)를 누르면, 시리얼 모니터에 Success/Fail을 출력하시오. (틀렸을 경우, 숫자야구게임처럼 힌트 제공할 것)
- 구성
  - Arduino Uno
  - 4x4 Key Matrix

