

7주차 - 디지털 IO 인터페이스 활용할 수 있는 센서

시각 표현 소자

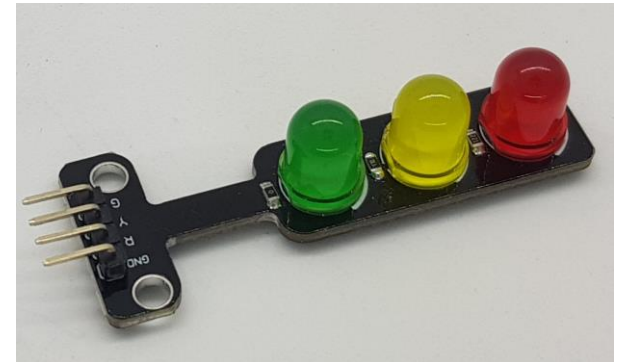
신호등 모듈 사용해보기

■ 모듈이란?

- ❖ 회로를 구성한 보드
- ❖ 기판위에 부품을 배치하고 결선하여 인터페이스를 간단하게 만든 전자 보드를 모듈이라 한다

■ 신호등 모듈

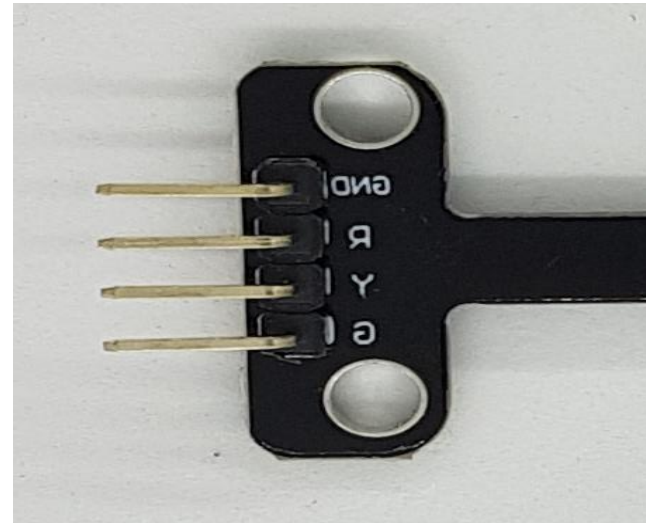
- ❖ 빨강, 노랑 초록 LED 모듈
- ❖ GND 공통에 각 LED에 HIGH 공급하여 켤 수 있다
- ❖ 핀이름은 R = 빨강, Y = 노랑, G = 초록, GND



신호등 모듈 핀

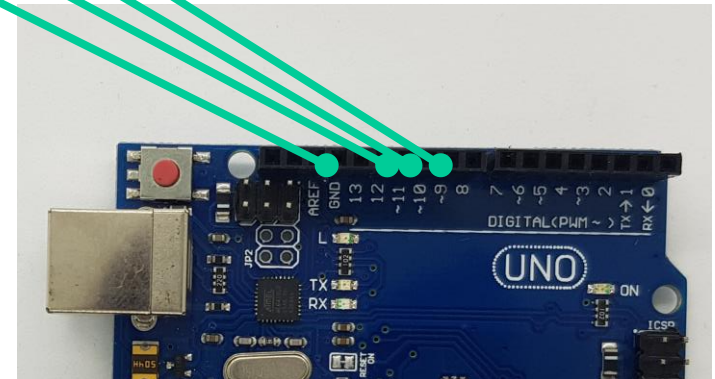
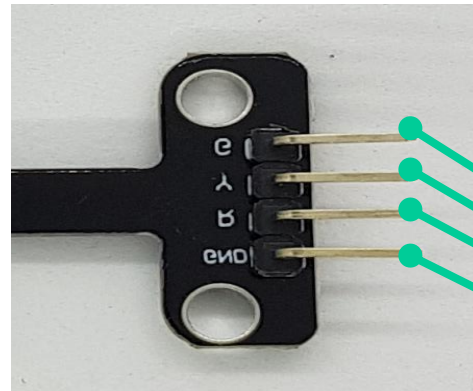
■ 핀이름은 다음과 같이 연결

- ❖ GND – 0V or GND
- ❖ R – Red LED
- ❖ Y – Yellow LED
- ❖ G – Green LED



신호등 연결

신호등모듈	아두이노
GND	GND
R	11
Y	10
G	9



신호등 모듈 - 코드(1)

```
#define RED_LED 11
#define YELLOW_LED 10
#define GREEN_LED 9

void setup()
{
  pinMode(RED_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
}
```

신호등 모듈 - 코드(2)

```
void loop()
{
    digitalWrite(RED_LED, HIGH);
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(GREEN_LED, LOW);
    delay(1000);
    digitalWrite(RED_LED, LOW);
    digitalWrite(YELLOW_LED, HIGH);
    digitalWrite(GREEN_LED, LOW);
    delay(1000);
    digitalWrite(RED_LED, LOW);
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(GREEN_LED, HIGH);
    delay(1000);
}
```

과제 – 신호등 모듈 수정

```
static int status;

void setup()
{
    status = 0;
    pinMode(RED_LED, OUTPUT);
    pinMode(YELLOW_LED, OUTPUT);
    pinMode(GREEN_LED, OUTPUT);
}

void loop()
{
    digitalWrite(RED_LED, LOW);
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(GREEN_LED, LOW);
```

```
    switch(status){
        case 0:
            status = 1;
            _____ BLANK 1 _____
            break;
        case 1:
            status = 2;
            _____ BLANK 2 _____
            break;
        case 2:
            status = 0;
            _____ BLANK 3 _____
            break;
        default:
            break;
    };
    delay(1000);
}
```

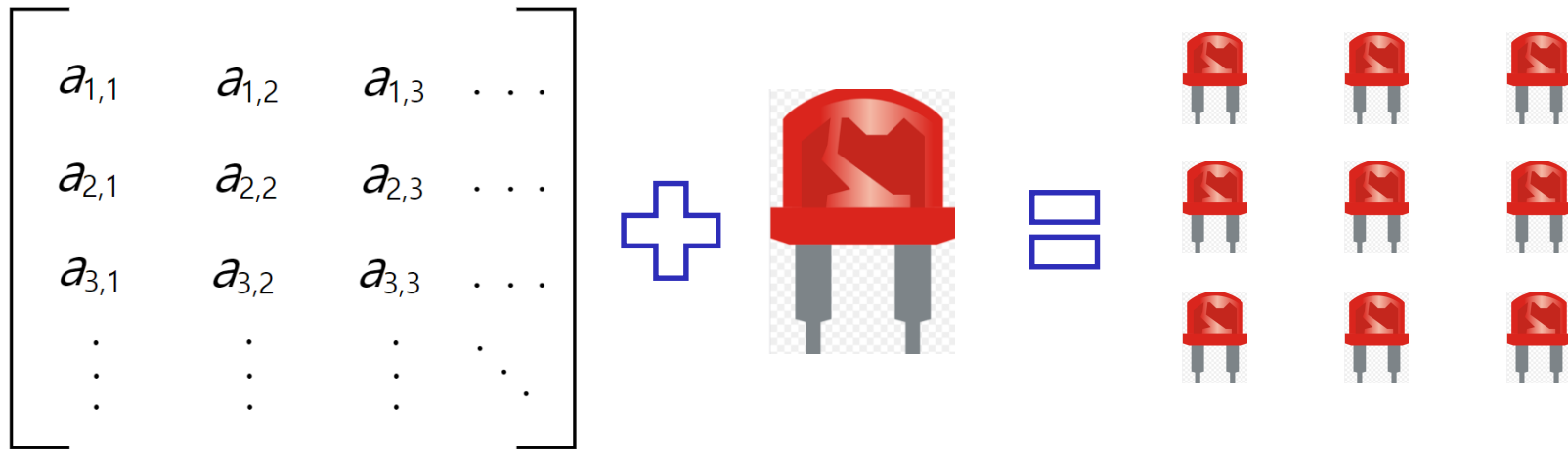
과제 – 기존 코드의 동작과 같이 1초에 한 개씩
켜지도록 **BLANK1, 2, 3** 을 채워주세요

시각 표현 소자

도트메트릭스(DotMatrix)

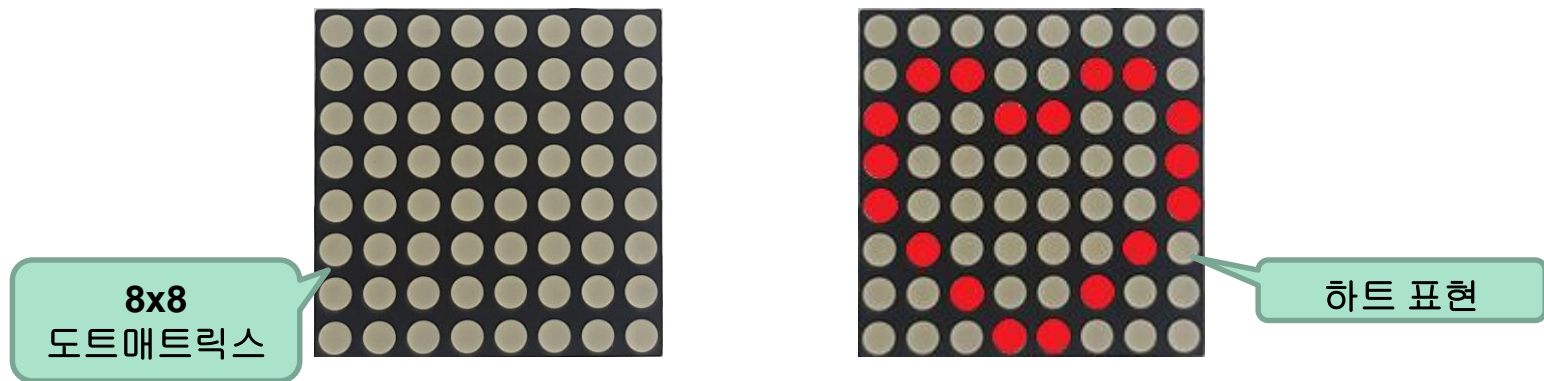
■ Dot + Matrix

- ❖ Dot는 LED를 의미한다.
- ❖ Matrix 는 행렬 배열 모양이다.
- ❖ LED 배열을 DotMatrix 라고 한다.



도트매트릭스(DotMatrix)

- 도트매트릭스는 행렬 LED 개수에 따라 사이즈가 가변이다
 - ❖ 8x8 또는 16x16. 그 이상의 배열도 있다.
- 문자, 숫자, 그림 등 Display 목적으로 사용 된다
- LED는 휘도가 높아 멀리서도 정확하게 보인다
- 전광판 또는 안내판 등에 사용되며, 일상 생활에서 많이 접할 수 있다

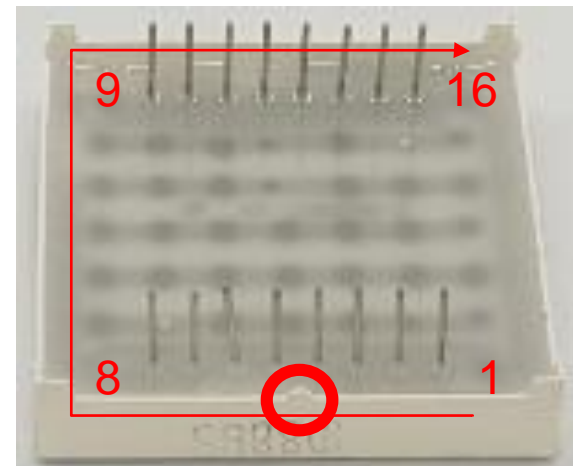


도트매트릭스 전광판



도트매트릭스 사용 방법

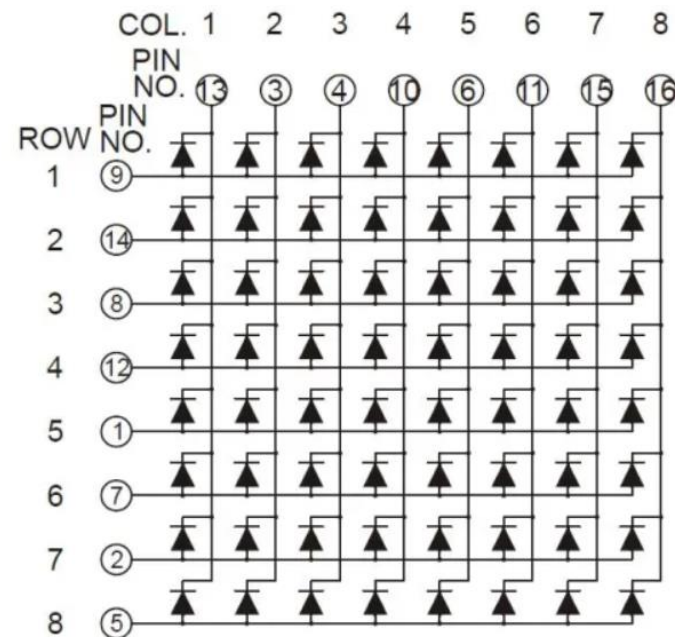
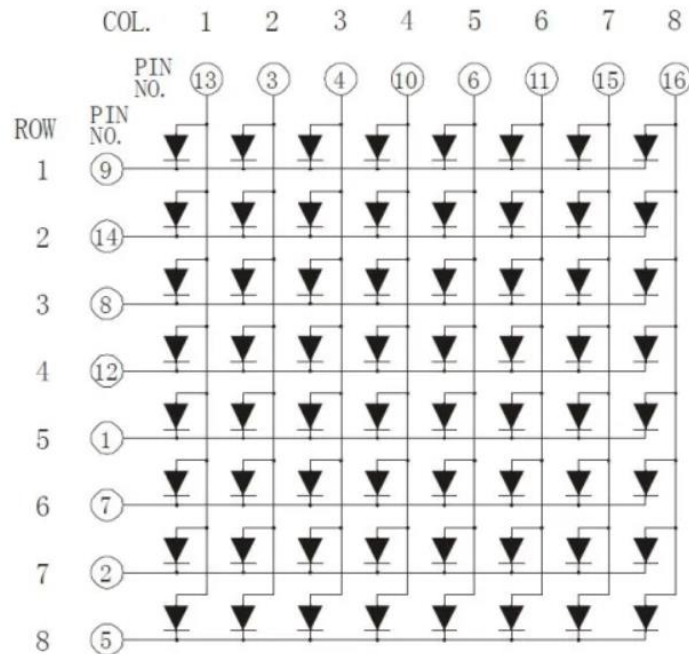
- 행렬의 핀이 있으며, 교차해서 선택할 수 있다
- 8x8 도트매트릭스는 16개의 핀이 있다
- 반시계 방향으로 핀이 카운트 된다
- COL과 ROW 핀이 섞여 있다
 - ❖ COL 13,3,4,10,6,11,15,16
 - ❖ ROW 9, 14, 8, 12, 1, 7, 2, 5



도트매트릭스 사용 방법

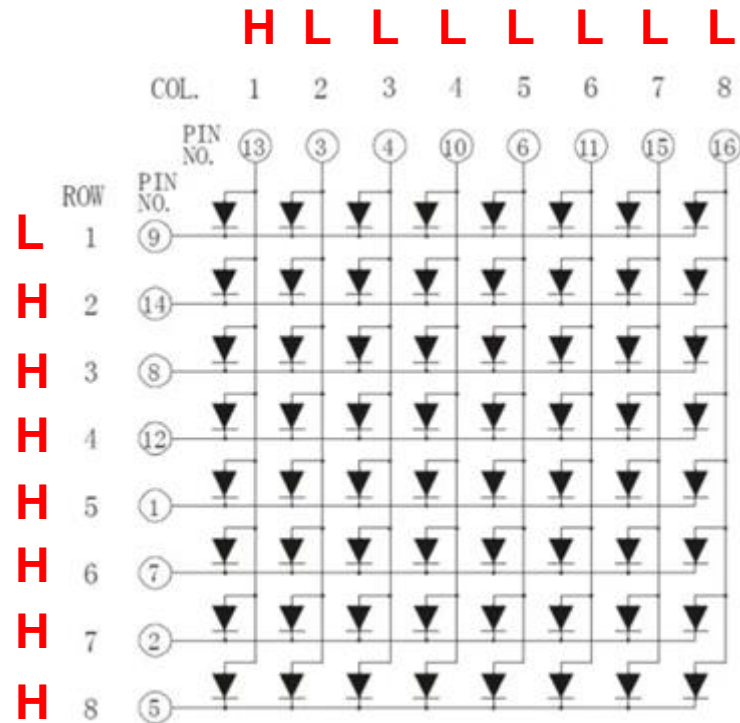
■ LED의 극성에 따라 두 가지 형태가 있다

❖ Common Cathode/Common Anode



도트매트릭스 사용 방법

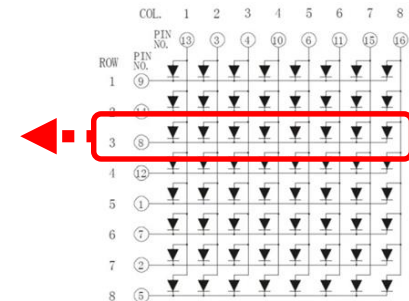
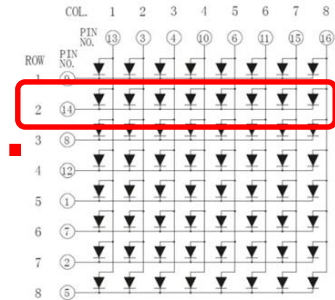
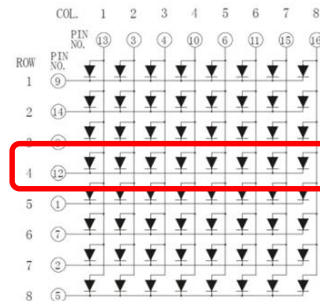
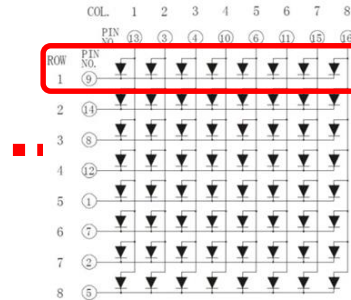
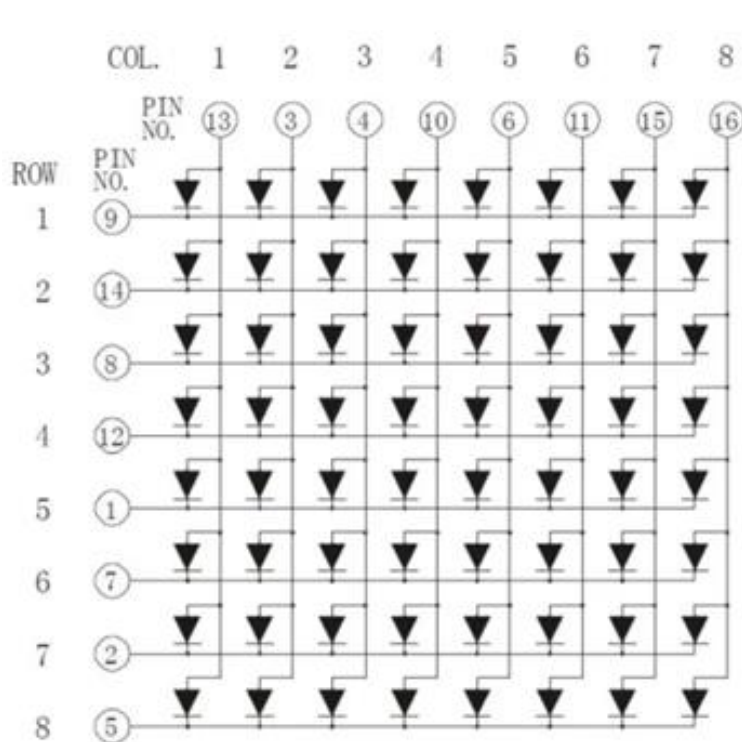
■ 한번에 16핀을 모두 제어 해야한다



도트매트릭스 사용 방법

■ 착시 현상을 이용하여 행 단위로 빠르게 제어한다

❖ 사람의 눈은 잔상이 사라지기전에 다시 비추면 계속 켜있다고 느낀다



도트매트릭스 연결도(Common Cathod)

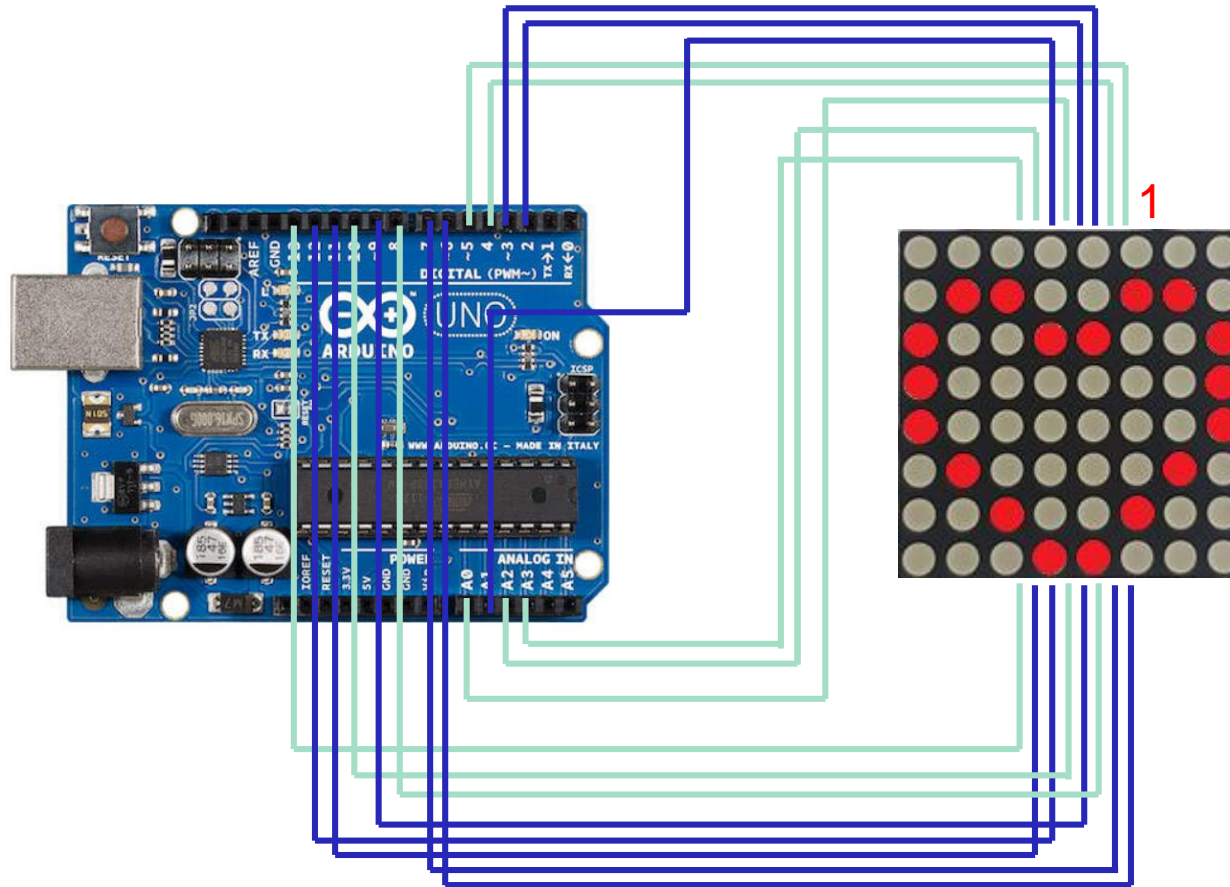
■ COL : 9,3,2,12,15,11,7,6

■ ROW: 13,8,17,10,5,16,4,14

COL	PIN	Arduino
1	13	9
2	3	3
3	4	2
4	10	12
5	6	15
6	11	11
7	15	7
8	16	6

ROW	PIN	Arduino
1	9	13
2	14	8
3	8	17
4	12	10
5	1	5
6	7	16
7	2	4
8	5	14

도트매트릭스 연결도(Common Cathode)



Common Anode Dotmatrix 켜기

```
void setup() {  
    pinMode( 9,OUTPUT);  
    pinMode( 3,OUTPUT);  
    pinMode( 2,OUTPUT);  
    pinMode(12,OUTPUT);  
    pinMode(15,OUTPUT);  
    pinMode(11,OUTPUT);  
    pinMode( 7,OUTPUT);  
    pinMode( 6,OUTPUT);  
  
    pinMode(13,OUTPUT);  
    pinMode( 8,OUTPUT);  
    pinMode(17,OUTPUT);  
    pinMode(10,OUTPUT);  
    pinMode( 5,OUTPUT);  
    pinMode(16,OUTPUT);  
    pinMode( 4,OUTPUT);  
    pinMode(14,OUTPUT);  
}
```

```
void loop() {  
    // Col  
    digitalWrite( 9,HIGH);  
    digitalWrite( 3,LOW);  
    digitalWrite( 2, LOW);  
    digitalWrite(12, LOW);  
    digitalWrite(15, LOW);  
    digitalWrite(11, LOW);  
    digitalWrite( 7, LOW);  
    digitalWrite( 6, LOW);  
  
    // Row  
    digitalWrite(13, LOW);  
    digitalWrite( 8, HIGH);  
    digitalWrite(17, HIGH);  
    digitalWrite(10, HIGH);  
    digitalWrite( 5, HIGH);  
    digitalWrite(16, HIGH);  
    digitalWrite( 4, HIGH);  
    digitalWrite(14, HIGH);  
}
```

스마일 예제 (1/5)

//표현할 문자를 2진수로 정의.

```
#define DOT { \  
    {0,0,1,1,1,1,0,0}, \  
    {0,1,0,0,0,0,1,0}, \  
    {1,0,1,0,0,1,0,1}, \  
    {1,0,0,0,0,0,0,1}, \  
    {1,0,1,0,0,1,0,1}, \  
    {1,0,0,1,1,0,0,1}, \  
    {0,1,0,0,0,0,1,0}, \  
    {0,0,1,1,1,1,0,0} \  
}
```

```
byte col = 0;  
byte leds[8][8];
```

스마일 예제 (2/5)

```
int pins[17] = {-1, 5, 4, 3, 2, 14, 15, 16, 17, 13, 12, 11, 10, 9, 8, 7, 6};
```

```
int cols[8] = {pins[13], pins[3], pins[4], pins[10],  
pins[6], pins[11], pins[15], pins[16]};
```

```
int rows[8] = {pins[9], pins[14], pins[8], pins[12], pins[1], pins[7],  
pins[2], pins[5]};
```

```
int pattern[8][8] = DOT;
```

스마일 예제 (3/5)

```
void setup() {  
    //아두이노 디지털 핀 출력 설정  
    for (int i = 1; i <= 16; i++) {  
        pinMode(pins[i], OUTPUT);  
    }  
  
    //LED를 켜기 전에 모두 끄기  
    for (int i = 1; i <= 8; i++) {  
        digitalWrite(cols[i - 1], LOW);  
    }  
    for (int i = 1; i <= 8; i++) {  
        digitalWrite(rows[i - 1], LOW);  
    }  
}
```

스마일 예제 (4/5)

```
void loop() {  
    //출력  
    display();  
    delay(2);  
}
```

```
void display() {  
    //다음 column으로 넘어가기 전에 이전led를 끈다  
    digitalWrite(cols[col], LOW);  
  
    //현재 column  
    col++;  
    if (col == 8) {  
        col = 0;  
    }  
}
```

스마일 예제 (5/5)

```
for (int row = 0; row < 8; row++) {  
    if (pattern[col][7 - row] == 1) {  
        digitalWrite(rows[row], LOW); // LED ON  
    }  
    else {  
        digitalWrite(rows[row], HIGH); // LED OFF  
    }  
}  
  
digitalWrite(cols[col], HIGH);  
} //display() 함수 종료
```


마무리

■ Delay를 1초로 늘려보자

```
void loop() {  
  //출력  
  display();  
  delay(2);           → delay(1000);  
}
```

❖ 행렬의 그리는 동작이 정확히 보이게 된다.

■ Tinkercad 를 사용하여 확인해보자

❖ Led 를 8x8로 배열하여 시뮬레이션할 수 있다.

❖ 코드 검증 및 이해하는데 효과적이다.