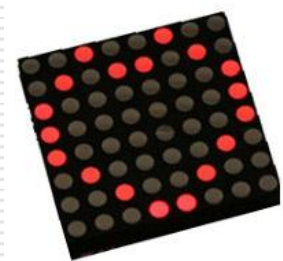


15 장 : 외부에 Dot Matrix 연결하여 사랑 고백하기

ATmega128 마이크로컨트롤러를 이용한 임베디드시스템 구현



JCnet
제이씨넷

신 상 석

목차

1. 도트 매트릭스 (Dot Matrix)
2. 실습 EXT-1 :

<Mission Impossible 2 !>

Dot Matrix로 ♥ 고백하기

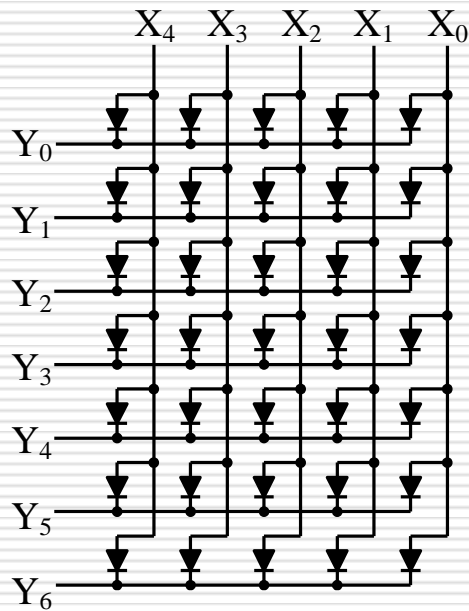
도트매트릭스 (Dot Matrix)

□ 도트매트릭스 (Dot Matrix)

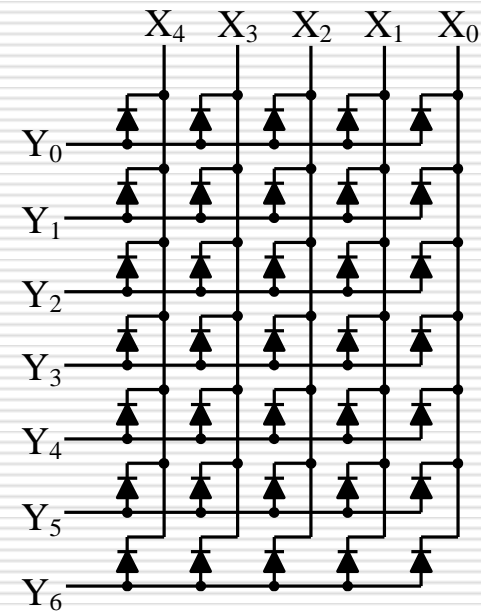
- LED를 행렬 형태로 배열하여 숫자나 문자 등을 디스플레이하는 소자
- LED 개수에 따라 실습용으로는 5×7 , 8×8 , 16×16 등이 있고, 상업용은 개수가 많고, 전력소모도 많은 임의 형태임
- 색깔은 단색은 노랑, 오렌지, 빨강, 녹색, 파랑, 흰색 등이 있으며, 다색(3색, 7색)의 경우도 존재
- 눈의 잔상 효과를 이용한 동적 드라이브 방식으로 제어 (FND 드라이브와 비슷함)
- 광고판, 전광판 등에 널리 사용

도트매트릭스 (Dot Matrix)

□ 도트매트릭스 (Dot Matrix)



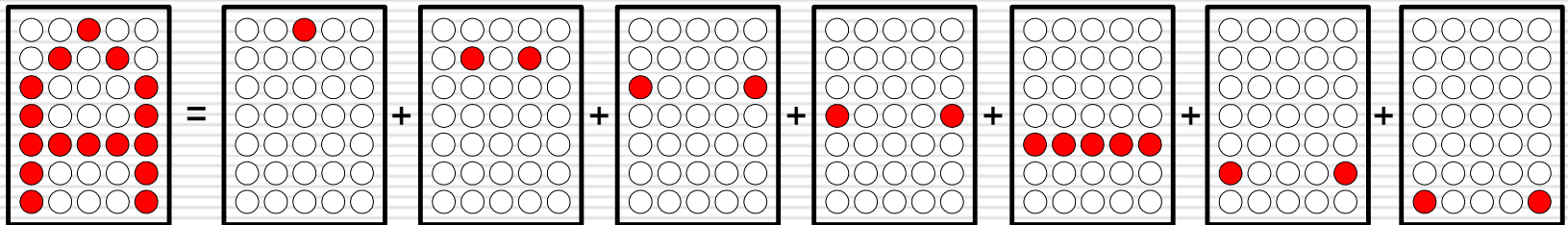
공통 캐소드(5×7)



공통 애노드(5×7)

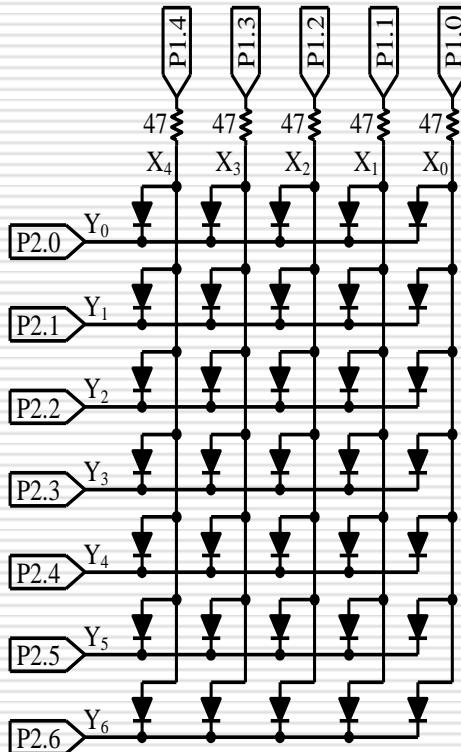
도트매트릭스 (Dot Matrix)

- 도트매트릭스 구동 방법 (5 x 7 도트매트릭스 경우)
 - X축과 Y축이 만나는 점에 LED가 있으므로 X축을 데이터 라인으로 하고 Y축을 스캔 라인으로 함
 - 즉, 첫번째 라인에 표시할 데이터 값을 X축에 보내고, Y축에는 맨 윗줄만 'ON' 시키는 데이터 값(예 : 0x01)을 보내면 첫번째 줄 표시 가능하므로, 이후 시분할 방법으로 두번째 줄 ~ 7번째 줄까지 Y축 값을 변경하면서 스캔하면 7개 라인 모두 표시할 수 있음 (단, FND와 비슷하게 시분할하여 한 줄씩 반복적으로 디스플레이하여야 하며, 한 줄 디스플레이에 4ms 이상 시간을 사용하면 안됨)

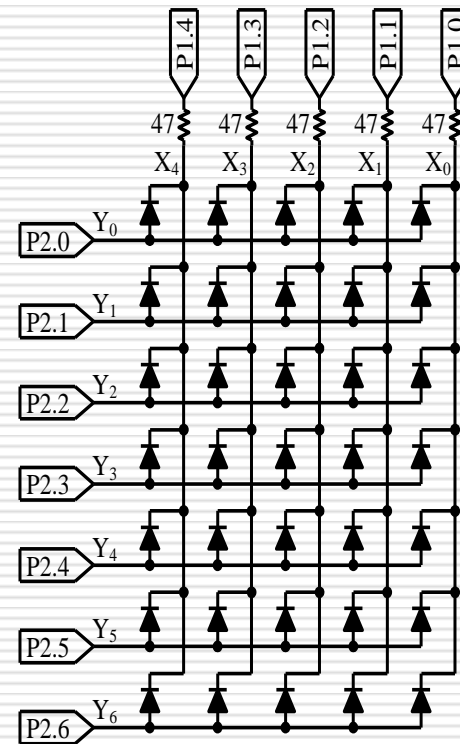


도트매트릭스 (Dot Matrix)

□ 간단한 도트매트릭스 연결 회로



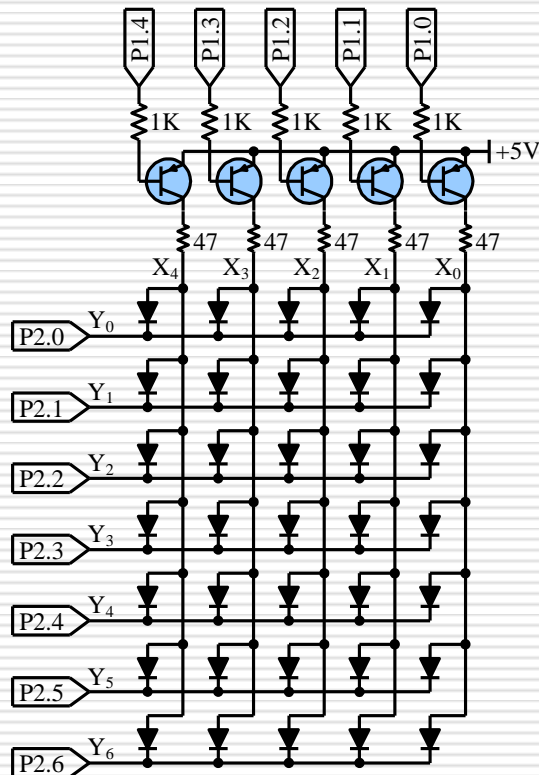
공통 캐소드인 경우



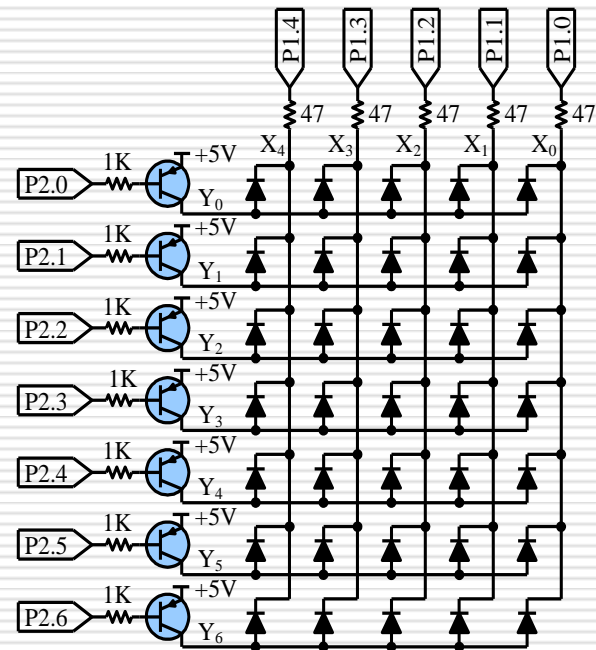
공통 애노드인 경우

도트매트릭스 (Dot Matrix)

□ 간단한 도트매트릭스-TR 연결 회로



공통 캐소드인 경우

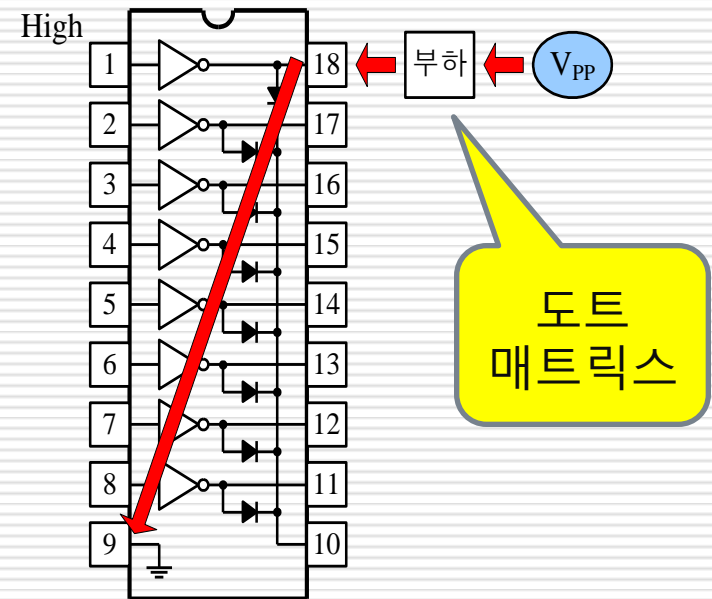
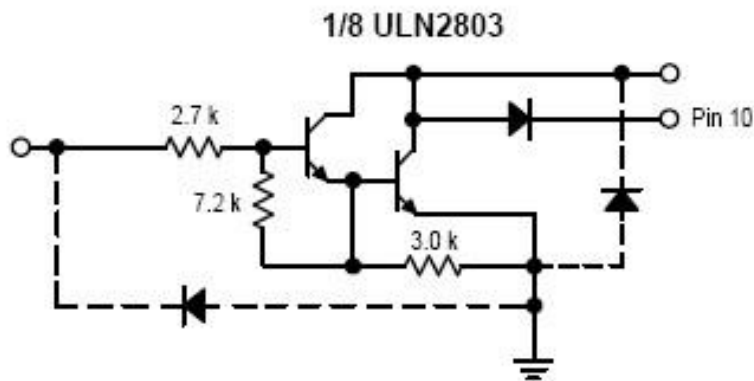


공통 애노드인 경우

도트 매트릭스 (Dot Matrix)

□ ULN2803

- 달링톤 회로 : 신호로 증폭하거나 제어할 때 사용할 목적으로 TR를 여러단으로 결합하여 만든 회로
- ULN2803 : NPN 타입의 달링톤 회로를 8개 포함하고 있는 IC



ULN2803인 경우

실습 EXT-1 : 도트 매트릭스로 ♥ 고백하기

□ Mission Impossible 2

1. 임무

(1) SZ420788K-16P 8x8 도트매트릭스 JKIT-128-1에
연결하여 ♥ 모양 나타내기

(2) "I ♥ You!"를 물 흘러가듯이 표현하기

2. 힌트

- ✓ SZ420788K-16P 는 엘레파츠(www.eleparts.co.kr)에서 구입
- ✓ 일반적인 대부분의 IC에 대한 데이터시트는 올데이터시트(www.alldatasheet.co.kr)에서 확보 가능
- ✓ JKIT-128-1 사용자매뉴얼과 회로도는 엘레파츠(www.eleparts.co.kr)나 디바이스마트(www.devicemart.co.kr)에서 확보 가능
- ✓ 무작정 달려들기 보다는, 일의 효율성을 위한 체계적인 방식을 채택하는 것이 유리함 (Top Down, Divide & Conquer)

실습 EXT-1 : 도트 매트릭스로 ♥ 고백하기

□ Mission Impossible 2 추진 절차 (예시)

1. SZ420788K, ULN2803 데이터시트, JKIT-128-1 회로도 확보



2. 위 2 부품과 1K 및 47옴 저항으로 JKIT-128-1 과 연결도 작성



3. 브래드보드에 편리한 형태로 잘~ 배선



4. 프로그램 작성 및 실행 (디버깅 포함)

실습 EXT-1 : 도트 매트릭스로 ♥ 고백하기

□ 생각해 볼 항목



실습 EXT-1 : 도트 매트릭스로 ♥ 고백하기

□ 구동프로그램 코딩 : 도트매트릭스 (dotmatrices-1-1.c)

```
#include <avr/io.h>
#define F_CPU 16000000UL
#include <util/delay.h>

int main()
{
    unsigned char data[] =
    {0x22, 0x77, 0xff, 0xff, 0xfe, 0x7c, 0x38, 0x10, 0x00}; // Heart
    int i;
    DDRA = 0xff;
    DDRC = 0xff;
```

실습 EXT-1 : 도트 매트릭스로 ♥ 고백하기

□ 구동프로그램 코딩 : 도트매트릭스 (dotmatrices-1-1.c)

```
while(1)
{
    for (i=0; i<8; i++)
    {
        PORTA = data[i];
        PORTC = 1 << i;
        _delay_ms(1);
    }
}
```

실습 EXT-1 : 도트 매트릭스로 ♥ 고백하기

□ 구동프로그램 코딩 : ♥ 고백하기 (dotmatrices-1-2.c)

```
#include <avr/io.h>
#define F_CPU 16000000UL
#include <util/delay.h>

int main()
{
    unsigned char data[] =
    {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,      // Blank
    0x3c, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x3c, 0x00, // I
    0x22, 0x77, 0xff, 0xff, 0xfe, 0x7c, 0x38, 0x10, 0x00,  // Love
    0x66, 0x66, 0x7e, 0x3c, 0x18, 0x18, 0x18, 0x3c, 0x00, // Y
    0x3c, 0x7e, 0x66, 0x66, 0x66, 0x66, 0x7e, 0x3c, 0x00, // O
    0x66, 0x66, 0x66, 0x66, 0x66, 0x66, 0x7e, 0x3e, 0x00, // U
    0x08, 0x18, 0x18, 0x18, 0x10, 0x00, 0x30, 0x30, 0x30}; // !
```

실습 EXT-1 : 도트 매트릭스로 ♥ 고백하기

□ 구동프로그램 코딩 : ♥ 고백하기 (dotmatrices-1-2.c)

```
int i, j, k=0;
DDRA = 0xff;
DDRC = 0xff;
while(1)
{
    for (k=0; k<55; k++)
        for (j=0; j<30; j++)
            for (i=0; i<8; i++)
            {
                PORTA = data[i+k];
                PORTC = 1 << i;
                _delay_ms(1);
            }
}
```

문고 답하기

Q & A

