임베디드시스템 기초(#514115)

#2. GPIO & Matrix Keypad

한림대학교
전자공학과 이선우

한림대학교
HALLYM UNIVERSITY

Short Review #1

- General Purpose Input Output (GPIO)
- Output port
- Input port
- Switch 사용 방법

General Purpose Input Output(GPIO) port

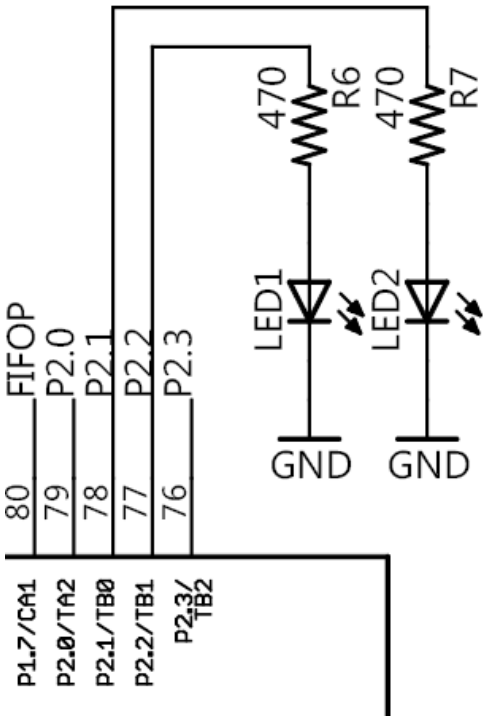
- 모든 MCU의 가장 기본적이고 중요한 주변장치(Peripheral device)
- 줄여서 Digital I/O or digital port라고도 함.
- MCU 칩의 각 핀(pin)이 수행하는 기능의 하나.
- 대개 핀 하나로 입출력 기능 수행 (I/O 기능 선택→direction 설정)
- Port
  - 복수개의 디지털 입출력 핀을 묶어서(보통 8개, 8bit) 지칭하는 용어
- Digital Output 기능
  - 스위치 기능: ON (Vcc 연결), OFF (GND 연결)
- Digital Input 기능
  - 핀의 상태(H/L)를 아는 기능

Digital output 기능

- 일반적인 logical gate의 출력과 동일 기능
  - 즉, 논리값 1 = Vcc 가 연결 → 3V/5V 출력
  - 논리값 0 = GND가 연결

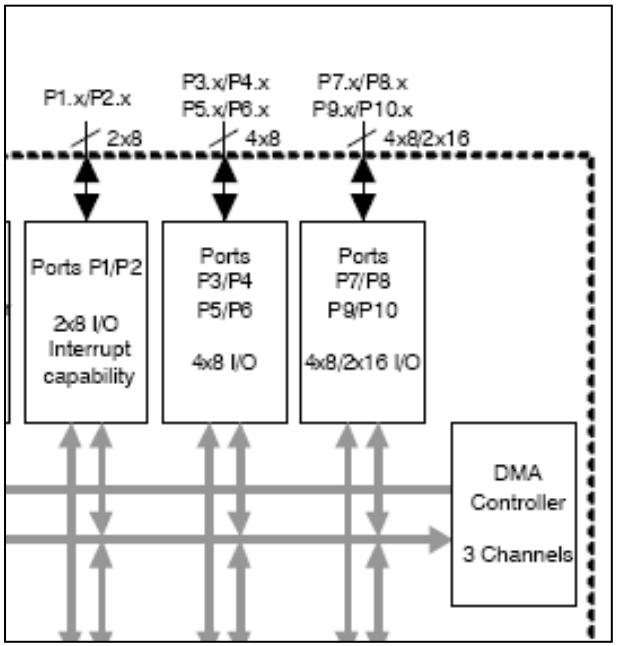
- 모든 외부 장치 ON/OFF 제어에 사용
  - Switch와 동일 기능 (회로적으론 다름)

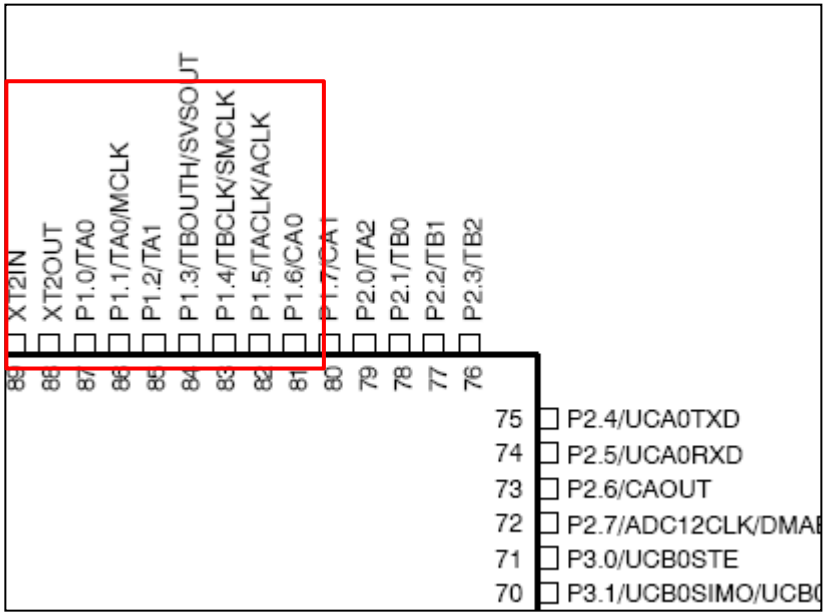
- 전기적 특성(Electrical characteristics)
  - Vcc=3V, $I_{OH(max)} < 48mA$
  - Vcc=2.2V, $I_{OH(max)} < 12mA$



MSP430FG461x Digital I/O ports

- MSP430FG4618
  - P1/P2: 2x8 I/O, 인터럽트 발생 가능
  - P3/P4/P5/P6: 4X8 I/O
  - P7/P8/P9/P10: 4X8/2X16 I/O
  - 각 핀별로 Direction(I/O) 설정 가능





한림대학교 HALLYM UNIVERSITY

Digital I/O Registers

Table 11–1. Digital I/O Registers, P1-P6

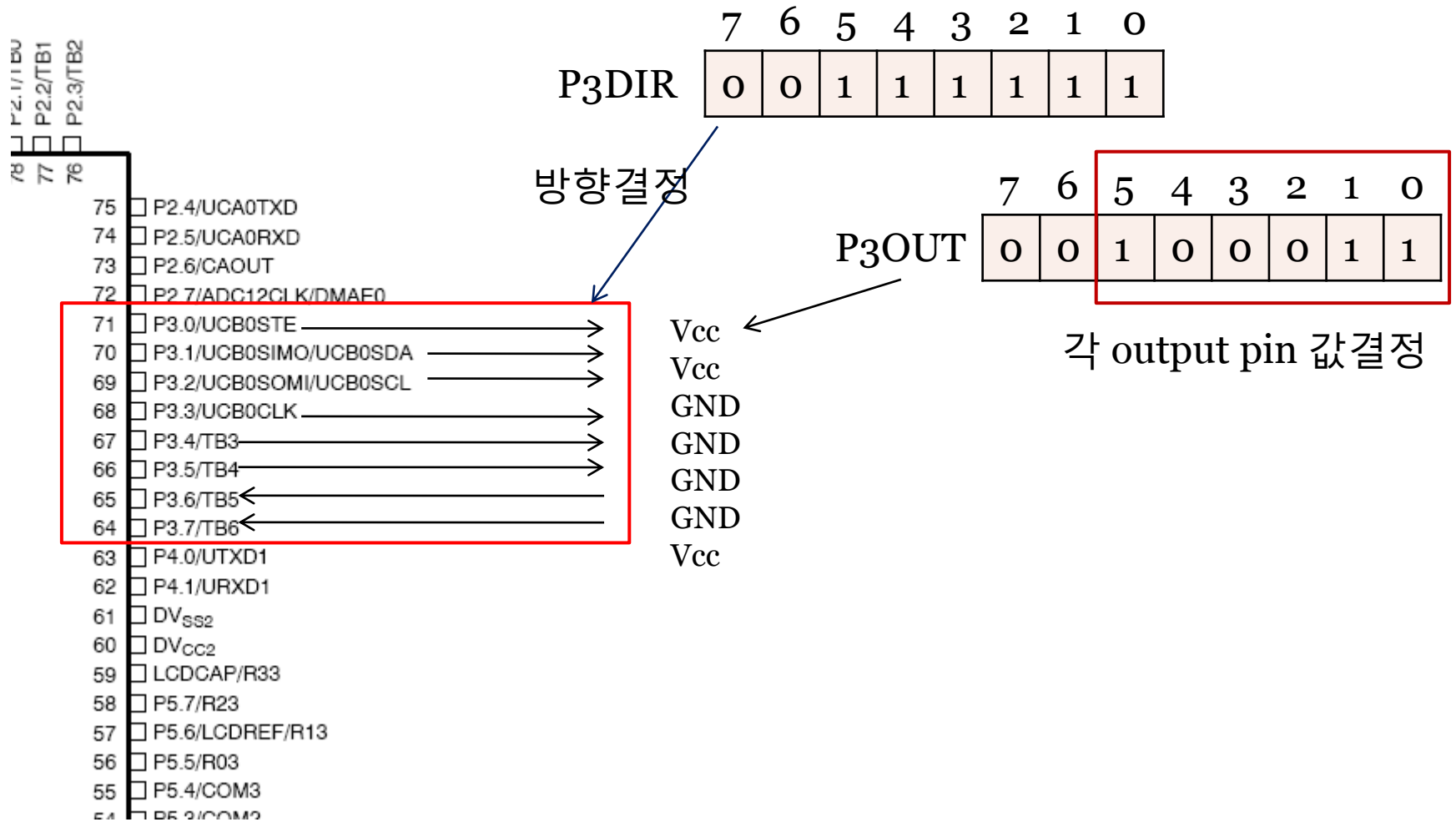
Port	Register	Short Form	Address	Register Type	Initial State
P1	Input	P1IN	020h	Read only	–
	Output	P1OUT	021h	Read/write	Unchanged
	Direction	P1DIR	022h	Read/write	Reset with PUC
	Interrupt Flag	P1IFG	023h	Read/write	Reset with PUC
	Interrupt Edge Select	P1IES	024h	Read/write	Unchanged
	Interrupt Enable	P1IE	025h	Read/write	Reset with PUC
	Port Select	P1SEL	026h	Read/write	Reset with PUC
	Resistor Enable	P1REN	027h	Read/write	Reset with PUC
P2	Input	P2IN	028h	Read only	–
	Output	P2OUT	029h	Read/write	Unchanged
	Direction	P2DIR	02Ah	Read/write	Reset with PUC
	Interrupt Flag	P2IFG	02Bh	Read/write	Reset with PUC
	Interrupt Edge Select	P2IES	02Ch	Read/write	Unchanged
	Interrupt Enable	P2IE	02Dh	Read/write	Reset with PUC
	Port Select	P2SEL	02Eh	Read/write	0C0h with PUC
	Resistor Enable	P2REN	02Fh	Read/write	Reset with PUC
P3	Input	P3IN	018h	Read only	–
	Output	P3OUT	019h	Read/write	Unchanged

PxOUT: Output으로 설정되었을 때 각 핀의 논리값(1/0) 결정
P1OUT.1 =1 ➔ P1.1 pin=Vcc

PxDIR: Direction을 설정하는 SFR
P1DIR.1 =1 ➔ P1.1 pin: output direction

Special Function Reg. (SFR)
내장 주변 장치를 제어하는 조정기

Digital Output 관련 SFR 역할



실습보드 장착 LED ON/OFF 제어

```
#include "msp430.h"

int main(void)
{
  unsigned int i;
  WDTCTL = WDTPW + WDTHOLD;
       // Stop watchdog timer
  P2DIR |= 0x06;
  P2OUT = 0x02;
  for (;;)
  {
    P2OUT ^= 0x06;
    i = 30000;
    do i--;
    while (i != 0);
  }
}
```

PORT의 입출력 방향 설정

- P2DIR ←0000 0110 B
  (P2.2=LED1, P2.1=LED2)

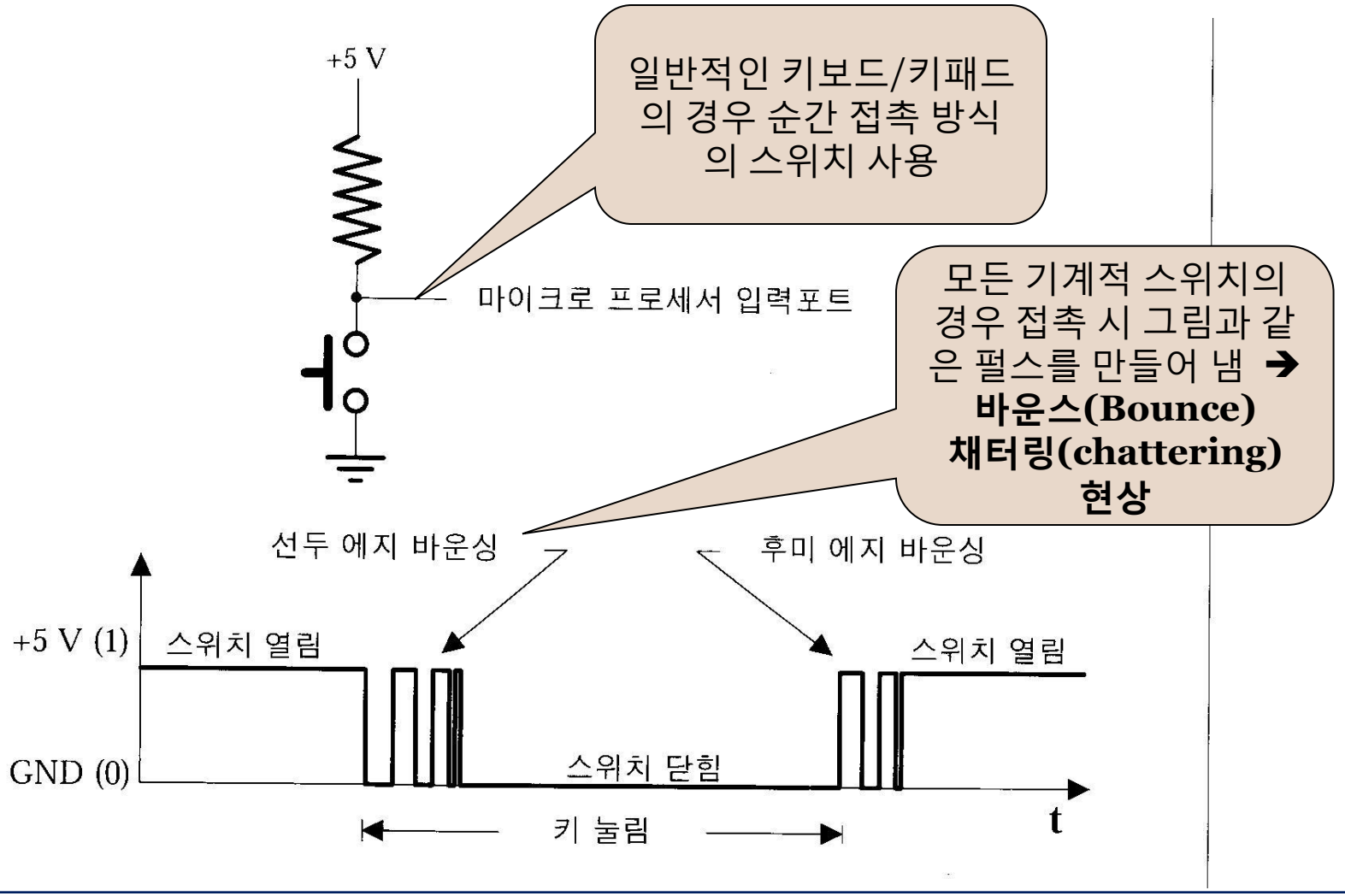
각 핀의 출력 설정

- 각 핀의 논리값을 결정.
- 1: H(ON,3.3V)
- 0: L(OFF,0V)

Toggle

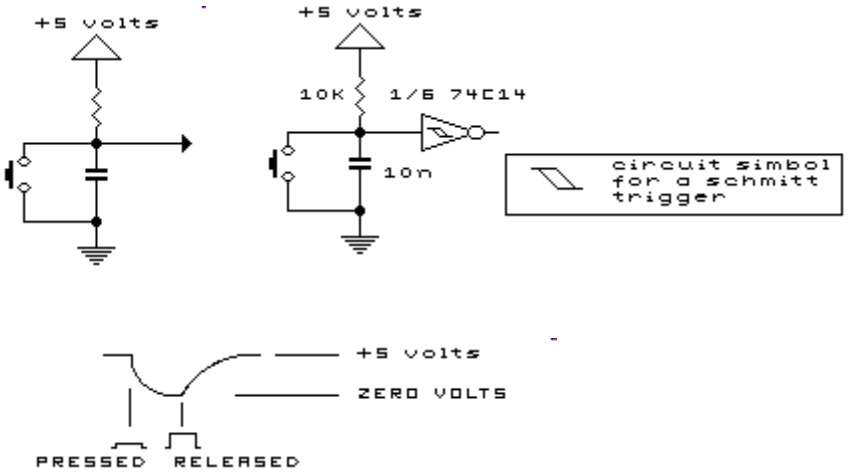
- ^= → P2OUT ^ 0000 0110

Mechanical switch 관련 특징/용어

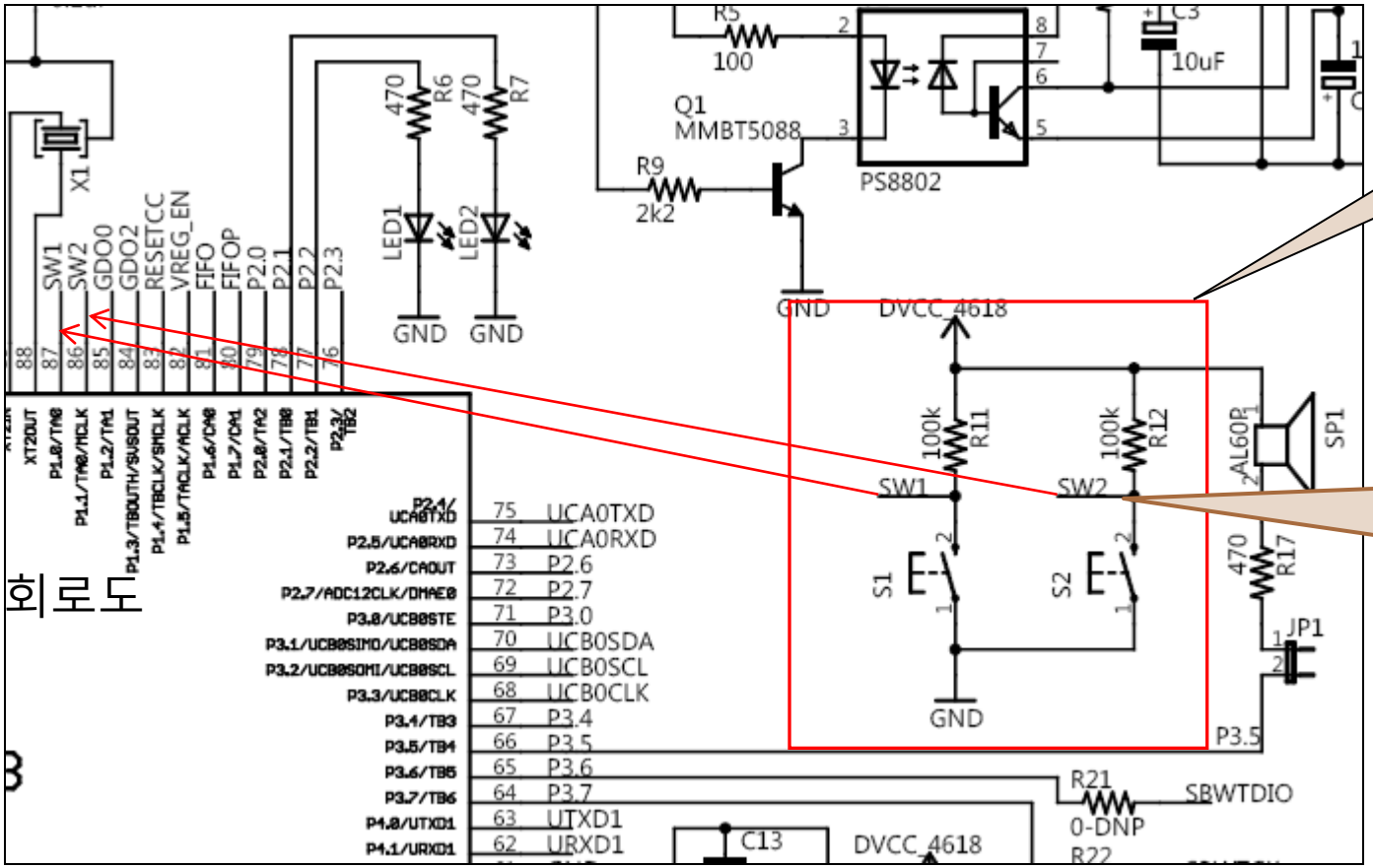


Debouncing (chattering 방지) 기법

- Contact bounce(chatter)에 의한 read error를 방지하기 위한 방법
  - Hardware적인 방법
    - RC Low-pass filter 회로 이용
      - 장점: 간단한 회로
      - 단점: 느린 응답성
    - 별도 전용 칩을 사용
      ELM410
  - Software적인 방법
    - 일정 시간 간격(약 수백ms)을 두고 2번 읽어 동일한 값이면 인정하는 방법



입력 포트 사용하기 (외부 상태 알기)



입력 포트 사용하기 (외부 상태 알기)

```
void main(void)
{
   WDTCTL = WDTPW + WDTHOLD;
   P2DIR |= 0x06;
   P1DIR &= 0xFC;

 while(1)
 {
  unsigned char swstatus;

   swstatus = P1IN;
   swstatus &= 0x03;
   swstatus = swstatus<<1;
   P2OUT = swstatus;

  delay(10000);
  }
}
```

PORT의 입출력 방향 설정

- output: P2.2=LED1, P2.1=LED2
- input: P1.0=SW1, P1.1=SW2

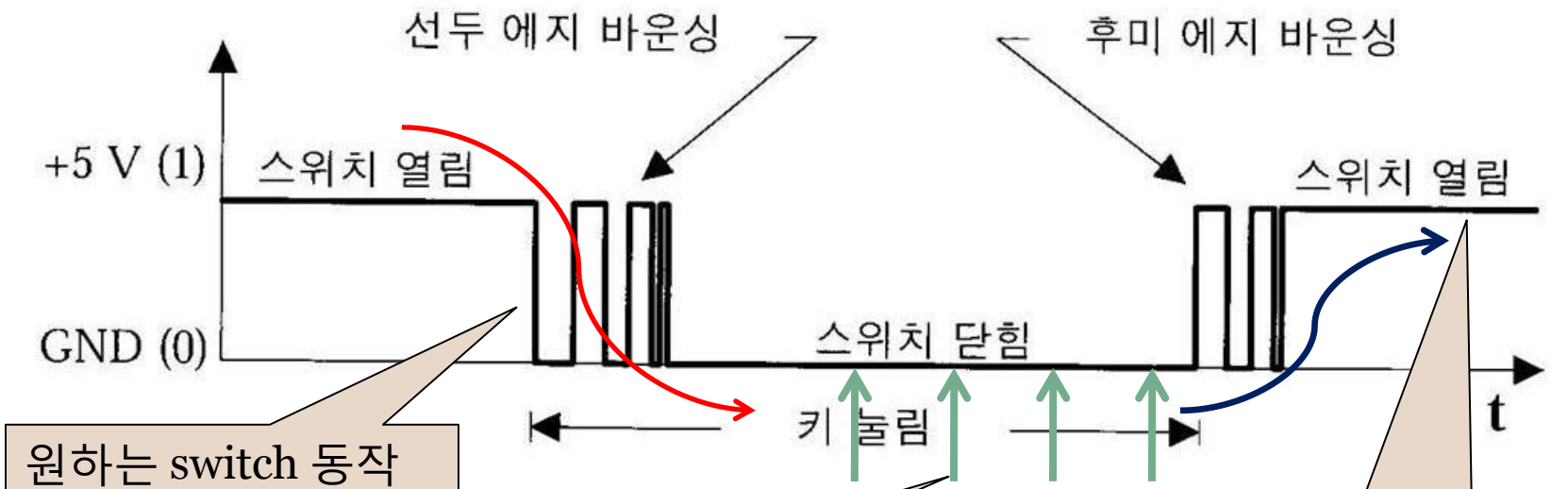
핀 상태값을 읽음→INPUT

- 1: H(ON,3.3V), 0: L(OFF,0V)
- 만약 SW가 눌려있다면 L의 값을 가짐

신호 처리

- &= 0x03; bit1,bit0만 값을 유지시키고 나머지 비트는 0으로.
- <<1: shift left 1bit
- SW 상태 반대로 LED로 표시, 즉 누르면 꺼짐.

연속 입력 문제



원하는 switch 동작
즉, H→L 만을 감지!

무한루프에 의한 연속 입력 발생➔ 원하지 않는 auto-repeat

*문제해결방법
스위치가 원상태로 돌아온 것을 감지하여 H→L을 enable시킴.

의도하지 않은 repeat 방지법

```
unsigned char rd_flag,  sw1, sw2;
//최초 설정, 보통 initial condition이라 함
rd_flag = 1;

while(1)
{
    sw1 = P1IN & 0x01;
    delay(10000);
    sw2 = P1IN & 0x01;
    if (  !sw1 && !sw2)   {
      if( rd_flag) {
                P2OUT  ^= 0x02;   //LED1 toggle
                rd_flag =0;
      }
    }
    else if ( sw1 && sw2 ) {
      rd_flag = 1;
    }

}
```

IF 조건문 의미

• 2번 읽어 두 값 모두 0일 때만 TRUE. 즉, 스위치 눌렸을 때

문제 해결 방법

• rd_flag가 1일 때만 응답

• rd_flag=0으로 리셋시킴으로 오직 한번만 실행됨.

SW1 Read enable

• SW1이 다시 원래 상태(H)를 감지하여 rd_flag=1 시킴.

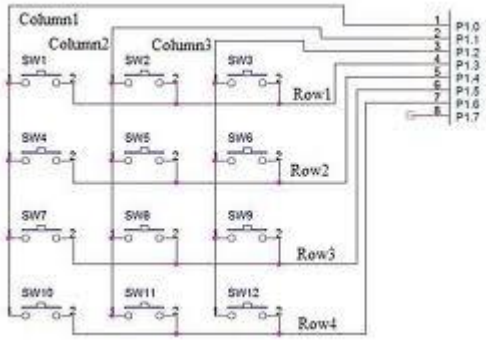
Matrix Keypad

- What ?
- How to use ?
- How to make a code ?

Keypad란?

- Keypad
  - A Keypad is a set of buttons arranged in a block which usually bear digits, symbols and usually a complete set of alphabetical letters. (http://en.wikipedia.org/wiki/Keypad)
  - Numeric keypad, telephone keypad, etc.

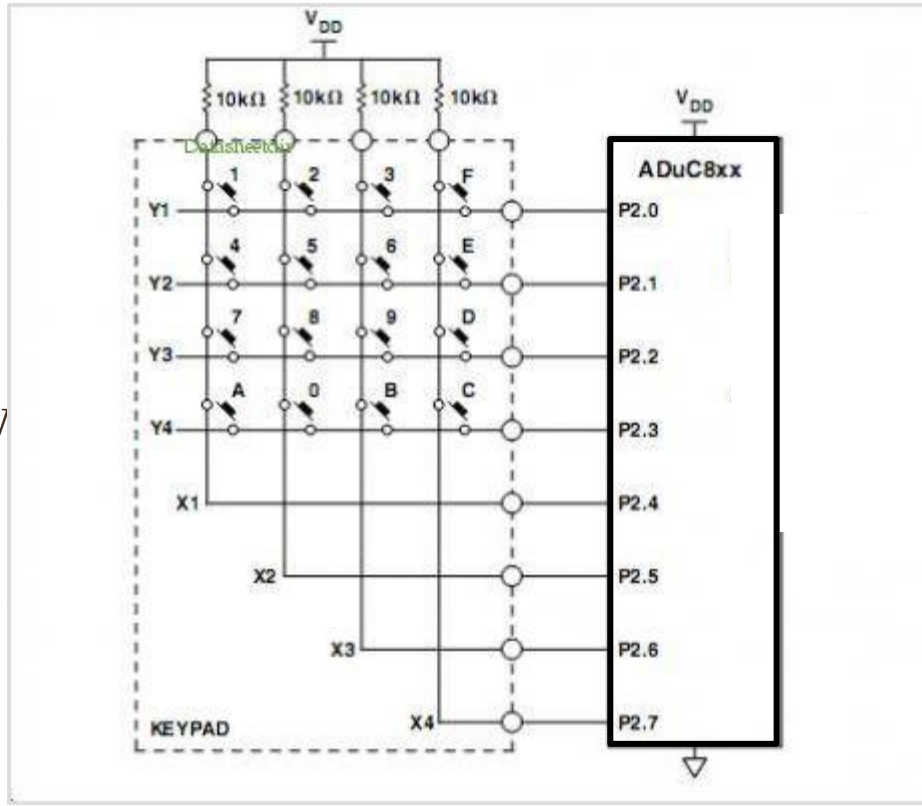
3X4 keypad



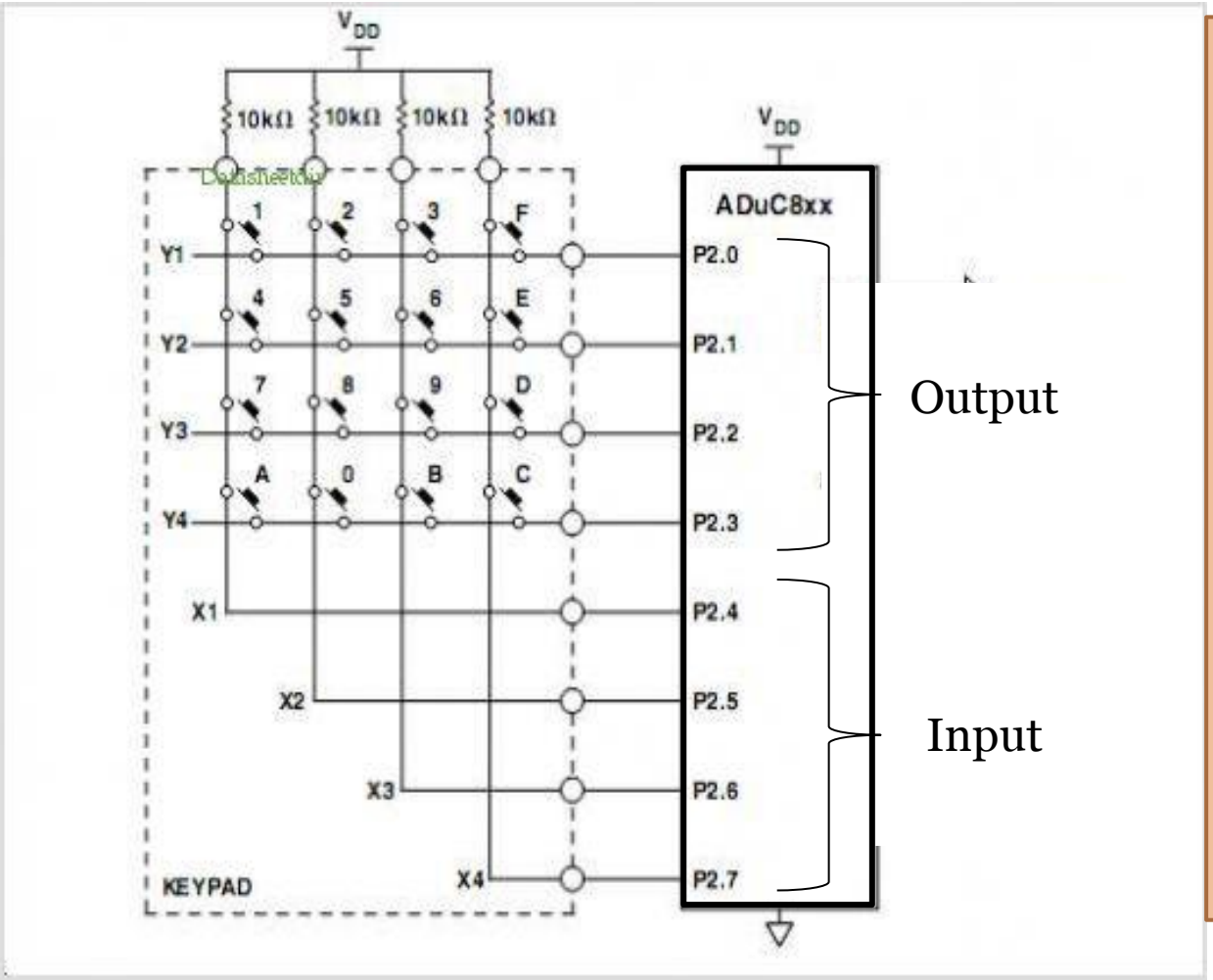
4X4 keypad

Keypad 사용 방법

- 사용 이유
  - 적은 수의 포트로 많은 키 입력 가능 (Ex: 16개 스위치→ 4x4→8 DIO 필요)
- 동작 원리
  - Pull-up된 X1~X4와 스위치로 연결된 Y1~Y4 라인 중 하나씩 Low로 만들면서 P2.4~P2.7 핀의 상태를 읽어 4개 스위치 중 하나의 눌림을 판단하는 방식➔ **scanning**
  - Ex.:P2.0=L, 나머지=H 상태에서 P2.4~P2.7 READ. 만약 1번 스위치가 눌렸다면 P2IN=1110 0000 (즉 P2.4=L)됨.



Example: 4X4 Keypad



- GPIO 설정
- P2.0~P2.3: output
- P2.4~P2.7:input

- Key mapping table

- If Y1=L, read [P2.7-4]
  4bit [1110] = 1
  4bit [1101] = 2
  4bit [1011] = 3
  4bit [0111] = F
- If Y2=L, read [P2.7-4]
  4bit [1110] = 4
  4bit [1101] = 5
  4bit [1011] = 6
  4bit [0111] = E

.....

Code for 4X4 keypad reading

```
void main()
{
 int num = 0;
 char rd;
 while(1)
 {
   do {
     num = 0xff;
     P2OUT &= ~(0x01); //P2.0=L
     rd = P2IN & 0xf0;
     if (rd != 0xf0)
         switch(rd) {
           case 0x0e: num=1; break;
           case 0x0d: num=2; break;
           case 0x0b: num=3; break;
           case 0x07: num=0xf;
         }
     P2OUT |= (0x01); //Y1 disable
    P2OUT &= ~(0x02); //P2.1=L
  //위와 비슷한 루틴 수행하여 Y2 line을
  //enable시켜 2~E 4개 스위치의 눌림을 감
   지함..
     P2OUT != (0x02); //Y2 disable
     //Y3 enable & check 4 SWs (7~D)
     //Y4 enable & check 4 SWs (A~C)
     } while(num == 0xff);


//16개 스위치 중 하나가 눌렸고 이를 읽은 상
   태.. 이 곳에서 눌린 스위치에 따른 작업을
   수행하면 된다.
}
```