

Ghép nối với thiết bị ngoại vi Inputs

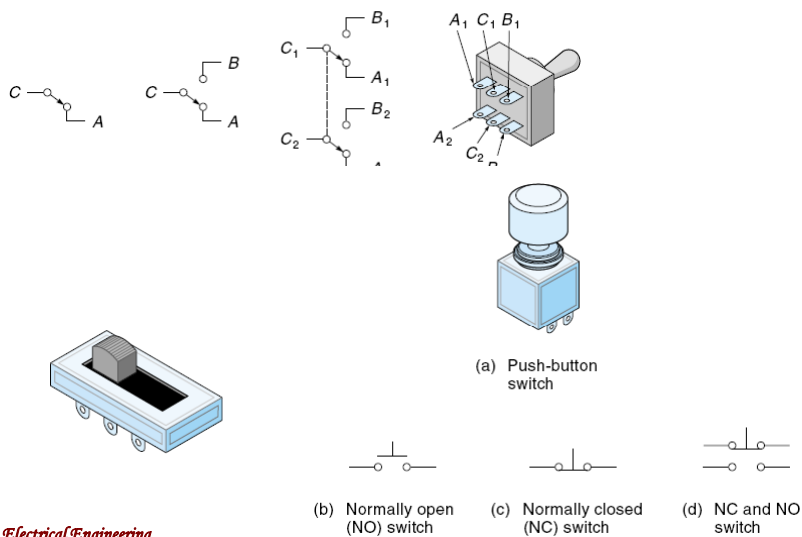
TS Nguyễn Hồng Quang



Electrical Engineering

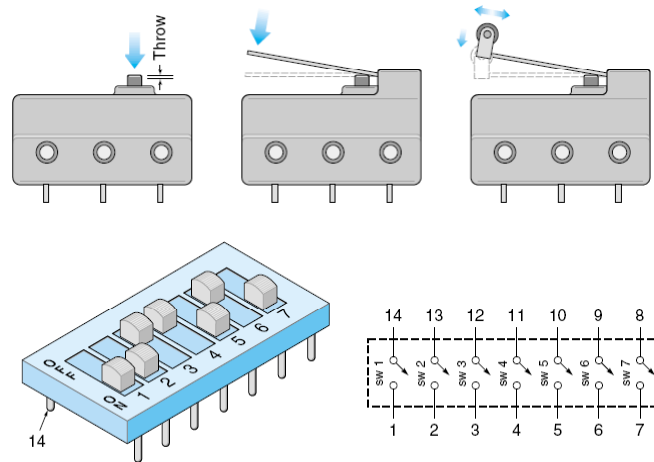
1

Các loại đầu vào thông dụng



Electrical Engineering

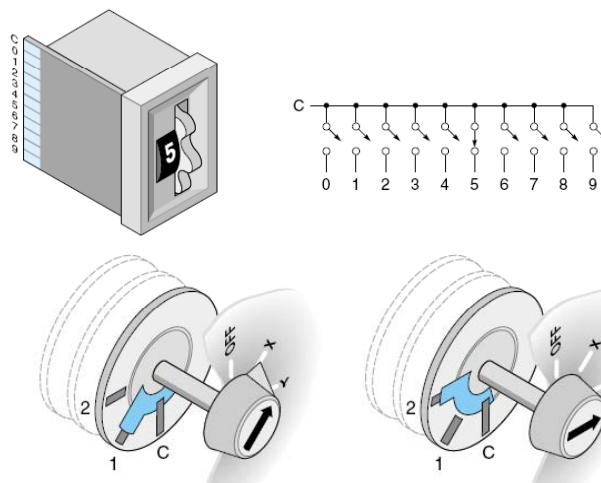
Công tắc hành trình



Electrical Engineering

3

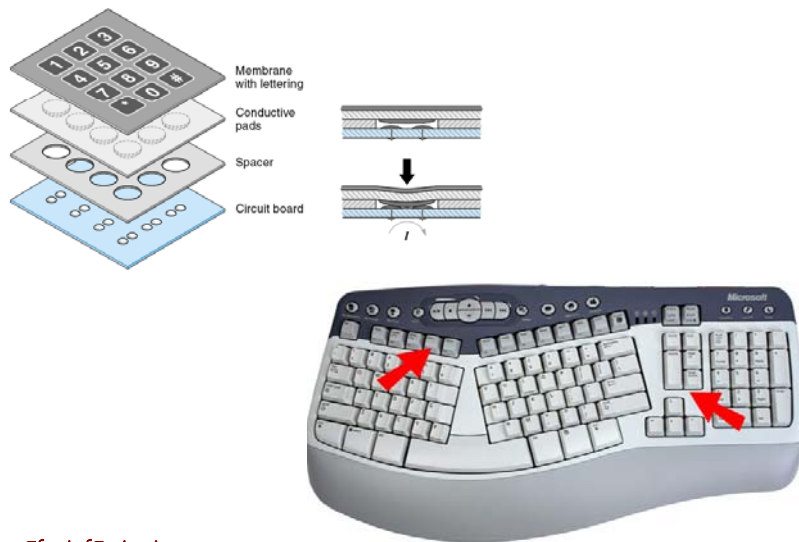
Công tắc khác



Electrical Engineering

4

Bàn phím



Electrical Engineering

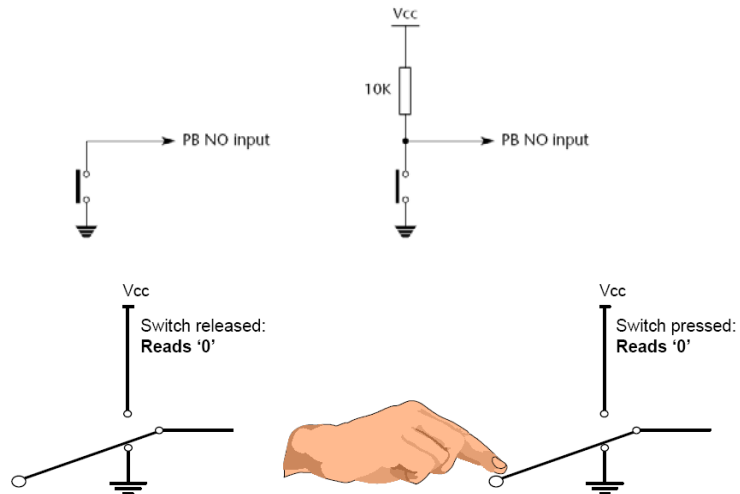
Giao tiếp hiện đại



Electrical Engineering

6

Phím ấn với điện trở treo



Electrical Engineering

7

Đọc tín hiệu đầu vào

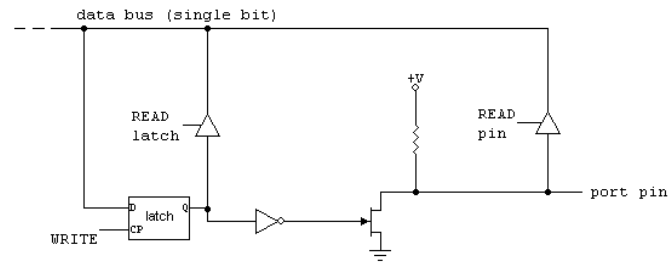
- Luôn đặt giá trị 1 vào cổng tương ứng nếu muốn cổng đó là cổng vào
- Ví dụ
 - SETB P1.0
 - MOV P2, #FFH
 - MOV C, P1.0
 - MOV R3, P2



Electrical Engineering

8

Giải thích

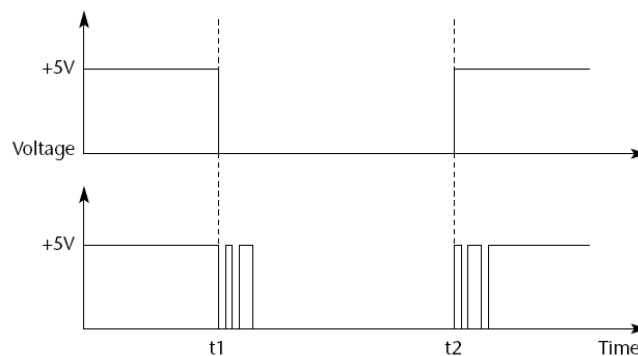


8051 single port pin

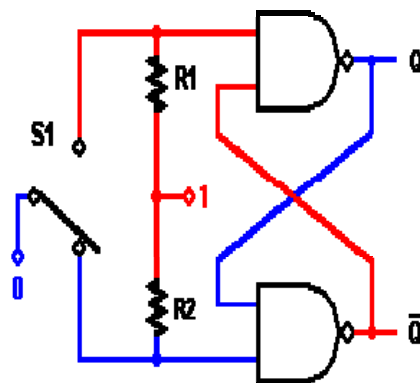
- Logic 0 đưa vào D sẽ mở FET, ghim port pin xuống GND
- Logic 1 khóa FET lại



Hiện tượng rung phím



Chống rung phím bằng phần cứng



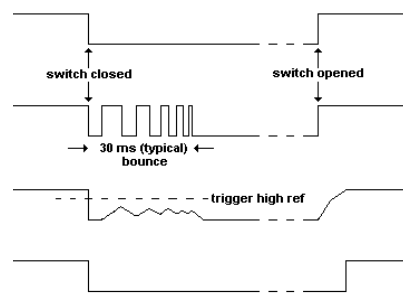
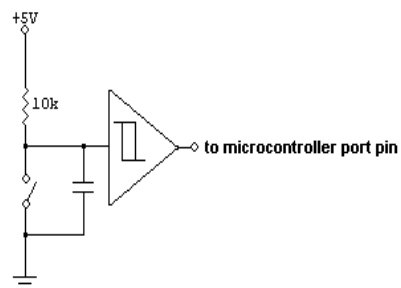
- Sử dụng trigger D, J-K



Electrical Engineering

11

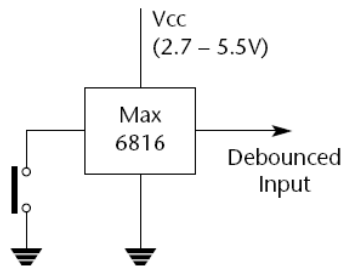
Sử dụng tụ



Electrical Engineering

12

Sử dụng IC chuyên dụng



- Đơn (Max6816), đôi (Max6817) và 8 chân (Max6818)
- Điện áp làm việc 2.5-5V
- Môi trường làm việc khắc nghiệt
- Chống rung cả lúc ấn và nhả phím



Sử dụng phần mềm

- Đọc phím ấn, kiểm tra xem trạng thái khác với trước không.
 - Trễ trong khoảng 20ms
 - Đọc lại phím ấn
 - Nếu giữ nguyên thì chuyển trạng thái



Minh họa phần mềm

```
bit SWITCH_Get_Input(const unsigned char DEBOUNCE_PERIOD)
{
    bit Return_value = SWITCH_NOT_PRESSED;

    if (Switch_pin == 0)
    {
        /* Switch is pressed */

        DELAY_LOOP_Wait(DEBOUNCE_PERIOD);

        if (Switch_pin == 0)
        {
            Return_value = SWITCH_PRESSED;
        }
    }

    return Return_value;
}
```

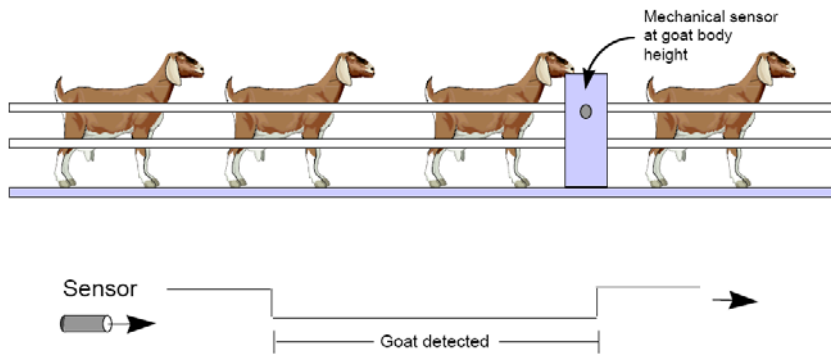


Nhược điểm phần mềm

- Phụ thuộc vào thời gian trễ, phụ thuộc vào từng loại phím
- Nếu người dùng nhấn phím quá lâu (>500ms) thì chương trình sẽ trả về N lần
- Không phù hợp với chương trình cần đếm số lần nhấn và nhả phím



Ví dụ đếm dê



Electrical Engineering

17

Cấu trúc phần mềm

```
bit SWITCH_Get_Input(const unsigned char DEBOUNCE_PERIOD)
{
    bit Return_value = SWITCH_NOT_PRESSED;

    if (Switch_pin == 0)
    {

        DELAY_LOOP_Wait(DEBOUNCE_PERIOD);

        /* Check switch again */
        if (Switch_pin == 0)
        {
            while (Switch_pin == 0);
            Return_value = SWITCH_PRESSED;
        }
    }

    return Return_value;
}
```



Electrical Engineering

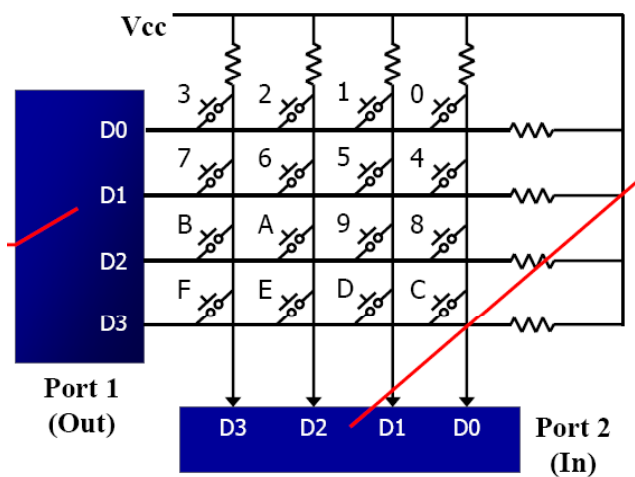
18

Các trường hợp tổ hợp phím khác

- Phím nhấn không giữ
- Phím nhấn và giữ nguyên trạng thái
- Phím nhấn và lần lượt chuyển trạng thái khác nhau tùy thời gian nhấn
- Tổ hợp nhiều phím



Ma trận phím



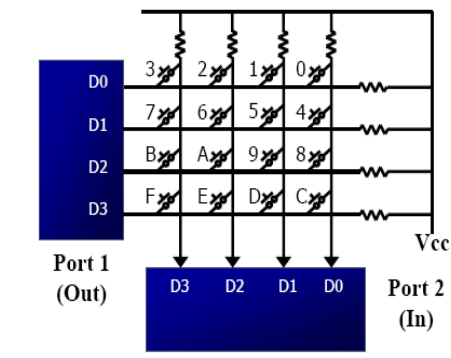
Phương pháp xác định phím ấn

- Phím được xác định bởi tổ hợp hàng và cột
- Các cột nối với đầu vào
- Các hàng nối với đầu ra
- Việc xác định phím được bắt đầu từ đọc trạng thái vào
 - Nếu trạng thái này bằng 1 cả thì không có phím ấn
 - Nếu trạng thái này khác 1 thì tìm xem cột nào được nhấn
- Việc đặt trạng thái của cột quyết định bởi lần lượt đặt hàng bằng 0 ở đầu ra



Ví dụ

- D3 – D0 = 1110 cho hàng, D3 – D0 = 1011 cho cột
- D3 – D0 = 1101 cho hàng, D3 – D0 = 0111 cho cột



Ví dụ phần mềm

;P1.0-P1.3 connected to rows, P2.0-P2.3 to column

```

MOV P2,#0FFH ;make P2 an input port
K1: MOV P1,#0 ;ground all rows at once
    MOV A,P2 ;read all col
        ;(ensure keys open)
    ANL A,00001111B ;masked unused bits
    CJNE A,#00001111B,K1 ;till all keys release
K2: ACALL DELAY ;call 20 msec delay
    MOV A,P2 ;see if any key is pressed
    ANL A,00001111B ;mask unused bits
    CJNE A,#00001111B,OVER;key pressed, find row
    SJMP K2 ;check till key pressed
OVER: ACALL DELAY ;wait 20 msec debounce time
    MOV A,P2 ;check key closure
    ANL A,00001111B ;mask unused bits
    CJNE A,#00001111B,OVER1;key pressed, find row
    SJMP K2 ;if none, keep polling

```



.....
Electrical Engineering

23

Tiếp

```

OVER1: MOV P1, #11111110B ;ground row 0
    MOV A,P2 ;read all columns
    ANL A,#00001111B ;mask unused bits
    CJNE A,#00001111B,ROW_0 ;key row 0, find col.
    MOV P1,#11111101B ;ground row 1
    MOV A,P2 ;read all columns
    ANL A,#00001111B ;mask unused bits
    CJNE A,#00001111B,ROW_1 ;key row 1, find col.
    MOV P1,#11111011B ;ground row 2
    MOV A,P2 ;read all columns
    ANL A,#00001111B ;mask unused bits
    CJNE A,#00001111B,ROW_2 ;key row 2, find col.
    MOV P1,#11110111B ;ground row 3
    MOV A,P2 ;read all columns
    ANL A,#00001111B ;mask unused bits
    CJNE A,#00001111B,ROW_3 ;key row 3, find col.
    LJMP K2 ;if none, false input,
        ;repeat

```



.....
Electrical Engineering

24

Tiếp

```

ROW_0: MOV  DPTR,#KCODE0    ;set DPTR=start of row 0
        SJMP FIND           ;find col. Key belongs to
ROW_1: MOV  DPTR,#KCODE1    ;set DPTR=start of row
        SJMP FIND           ;find col. Key belongs to
ROW_2: MOV  DPTR,#KCODE2    ;set DPTR=start of row 2
        SJMP FIND           ;find col. Key belongs to
ROW_3: MOV  DPTR,#KCODE3    ;set DPTR=start of row 3
FIND:   RRC  A               ;see if any CY bit low
        JNC  MATCH          ;if zero, get ASCII code
        INC  DPTR            ;point to next col. addr
        SJMP FIND           ;keep searching
MATCH:  CLR  A               ;set A=0 (match is found)
        MOVC A,@A+DPTR      ;get ASCII from table
        MOV  P0,A            ;display pressed key
        LJMP K1
;ASCII LOOK-UP TABLE FOR EACH ROW
ORG 300H
KCODE0: DB  '0','1','2','3' ;ROW 0
KCODE1: DB  '4','5','6','7' ;ROW 1
KCODE2: DB  '8','9','A','B' ;ROW 2
KCODE3: DB  'C','D','E','F' ;ROW 3
        END

```



Vấn đề xử lý bàn phím

- Chống rung triệt để
- Bàn phím chưa được xử lý khi có phím chức năng (tổ hợp nhiều phím)
- Phím CAPLOCK
- Xử lý phím giữ trong thời gian dài



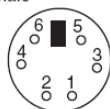
Sử dụng bàn phím IBM/PC

Male



(Plug)

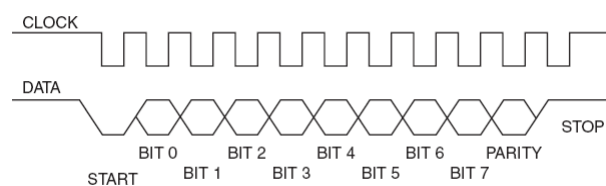
Female



(Socket)

6-pin mini-DIN (PS/2):

- 1 – Data
- 2 – NC
- 3 – Ground
- 4 – V_{cc} (+5 V)
- 5 – Clock
- 6 – NC



Electrical Engineering

27