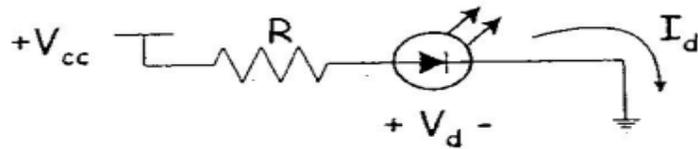


7 Usage of 68HC11 in Controlling System

7.1 interface with LED display

- Basic of one LED connection



$+V_{cc}$ – Supply voltage to LED

R – Resistor used to limit the current in LED

V_d – Minimum voltage needed to light up the LED (~1.5V)

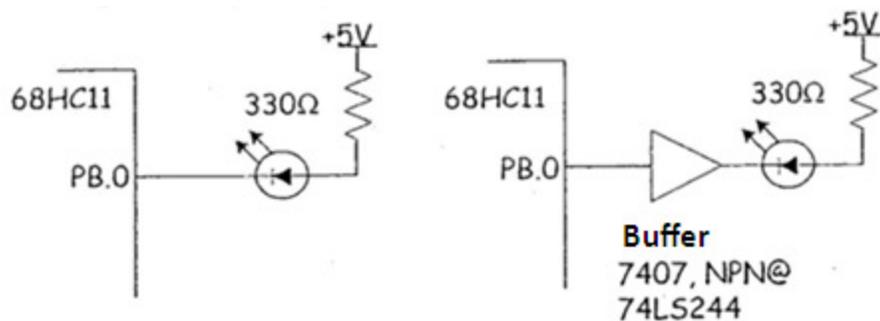
I_d – The current flow in LED (5-25mA)

Example: Estimate the current limited resistor value of LED, if it will connect to the voltage supplier which is +5V.

If said that the medium lightning display is needed. Choose $I_d = 10\text{mA}$.

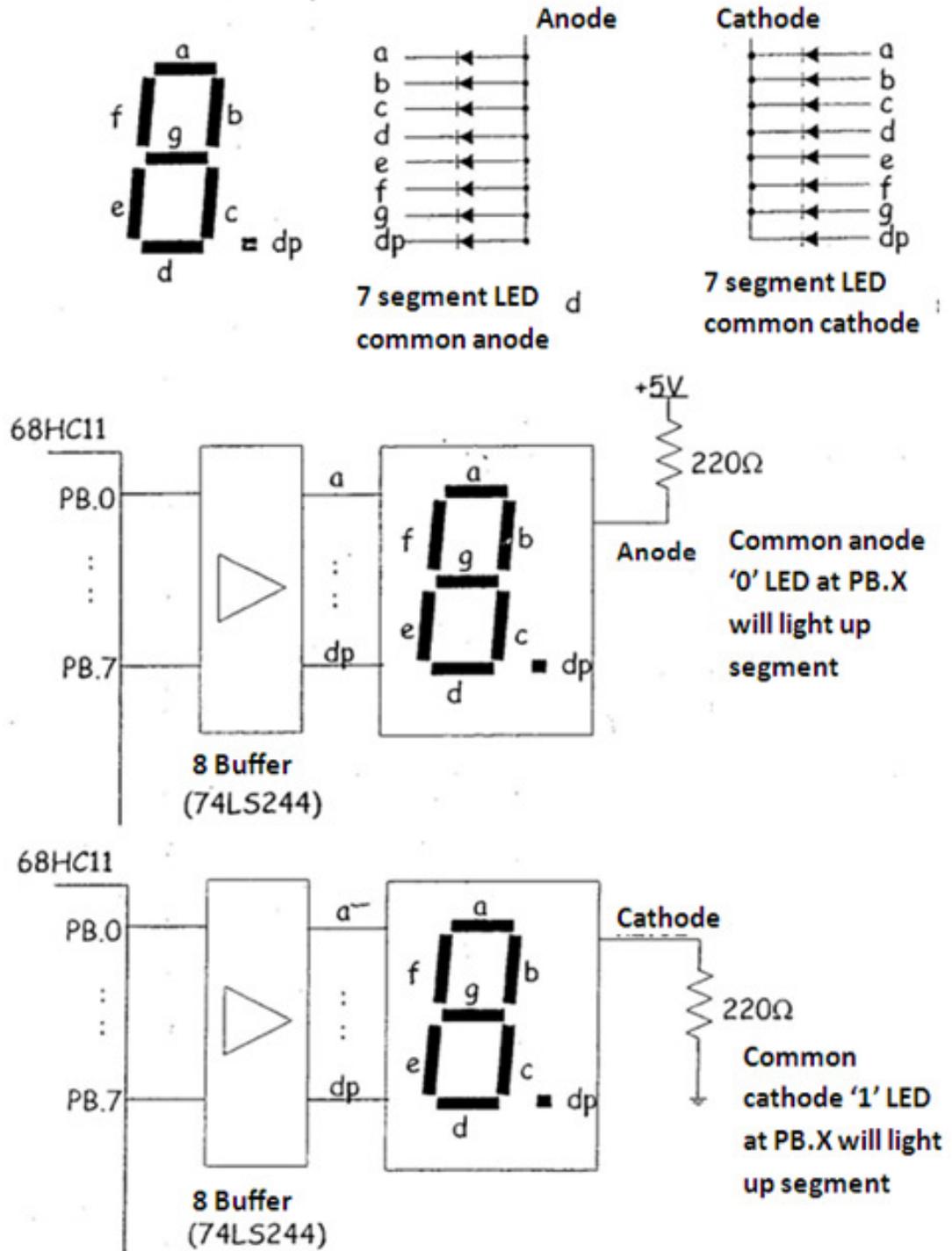
$$\begin{aligned} V_{cc} - V_d &= I_d \times R \\ R &= (V_{cc} - V_d) / I_d \\ R &= (5 - 1.5) / 10 \times 10^{-3} \Omega \\ &\sim 330 \Omega \end{aligned}$$

- Connection of LED to the output port of 68HC11



7.1 interface with LED display (cont...)

- Connection of 7 segment LED to the 68HC11 output port



7.1 interface with LED display (cont...)

Example: write a program to display the word 'rosbi' at a 7 segment LED with continuous common anode. Delay it for 500ms in between letter/character for displaying.

Solution: use the software coding to display each of the letter/character. Assume that the crystal being used is 8MHz.

Letter	Coding ('0' = light on, '1' = light off)								Code
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
	dp	g	f	e	d	c	b	a	
r	1	1	0	0	1	1	1	0	\$CE
o	1	1	0	0	0	0	0	0	\$CO
s	1	0	0	1	0	0	1	0	\$92
b	1	0	0	0	0	0	1	1	\$83
i	1	1	0	0	1	1	1	1	\$CF

```

PORTB      EQU      $1004

REPEAT
AGAIN      LDX      #NAME
           LDAA     0, X      ; get 1 letter
           BEQ      REPEAT   ; finished?
           STAA    PORTB    ; if not, display
           JSR     DELAY5    ; wait for a while
           INX
           BRA     AGAIN

NAME       FCB      $CE, $CO, $92, $83, $CF, 0

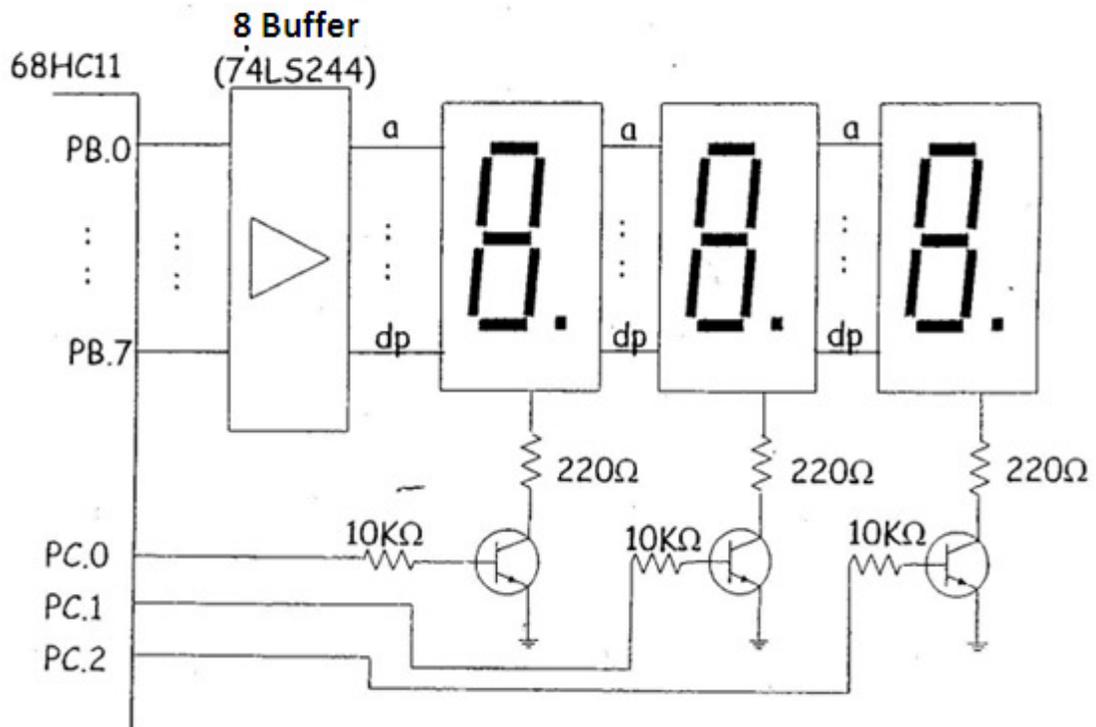
DELAY5     LDAB     #5       ; delay 5x1000ms
DELAY1     LDY     #28571    ; wait 100ms
DELAY      DEY
           BNE     DELAY
           DECB
           BNE     DELAY
           RTS
    
```

7.1 interface with LED display (cont...)

Example: Show how three 7 segment LED common cathode can connect to 68HC11 in the multiplexed display technique. Next, write a program to display '123' at the three LED simultaneously.

Solution: Use the software coding technique to display each of the letter/character. Assume the used crystal is 8MHz.

Letter	Coding ('1' = light on, '0' = light off)								Code
	PB7 dp	PB6 g	PB5 f	PB4 e	PB3 d	PB2 c	PB1 b	PB0 a	
1	0	0	1	1	0	0	0	0	\$30
2	0	1	0	1	1	0	1	1	\$5B
3	0	1	0	0	1	1	1	1	\$4F



7.1 interface with LED display (cont...)

Solution: In the multiplexed technique, only one LED can be activated in a time within rotation while others LED will deactivated. A LED is active awhile (~5ms), then deactivate while the next LED will be light on. Because of the range light on and off is fast, human's eye can only seen it as simultaneous display. Port PC0-PC2 and the NPN transistor will be used to activate ('1') @ deactivate ('0') three of 7 segments LED. Port B (PB0-PB7) used to send the codes for display.

```
LDX      #$1000
LDAA     #%1111000 ; set the output PC0-PC2
STAA     DDRC, X
LDAA     #0
STAA     PORTC, X ; light off all LED

REPEAT   LDAA     #$30 ; '1'
          STAA     PORTB, X ; display at LED 1
          BSET     PORTC, X %00000001
          JSR      DELAY5M ; wait for 5 ms
          BCLR     PORTC, X %00000001 ; light off LED 1

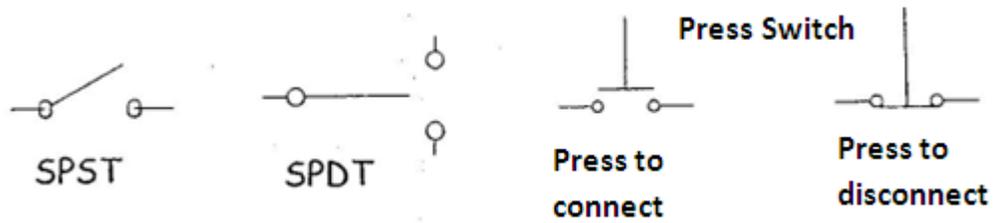
          LDAA     #$5b ; '2'
          STAA     PORTB, X ; display at LED 2
          BSET     PORTC, X %00000010
          JSR      DELAY5M ; wait 5 ms
          BCLR     PORTC, X ; 00000010 ; light off LED 2

          LDAA     #$5B ; '2'
          STAA     PORTB, X ; display at LED 3
          BSET     PORTC, X %00000100
          JSR      DELAY5M ; wait for 5 ms
          BCLR     PORTC, X %00000100 ; light off LED 3

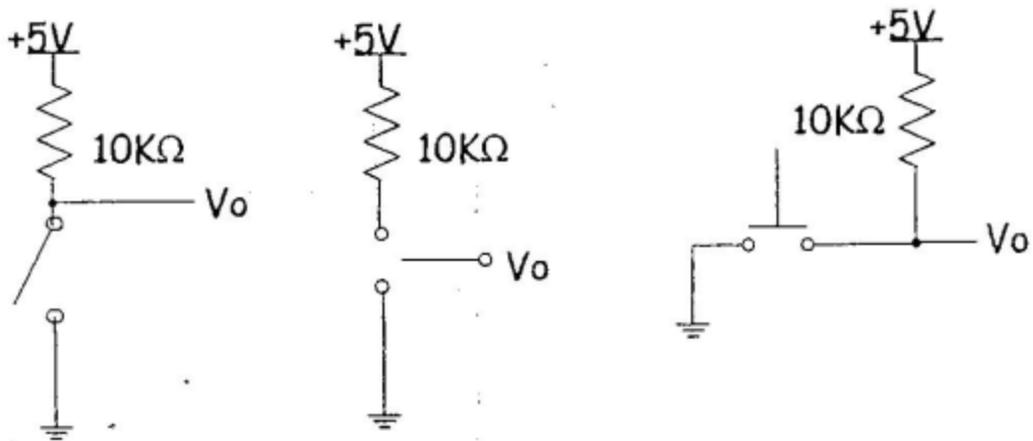
DELAY5M  LDY      #$4F ; delay for 5ms
DELAY    DEY
          BNE     DELAY
          RTS
```

7.2 Interface with mechanical switch

- Mechanical switch symbols



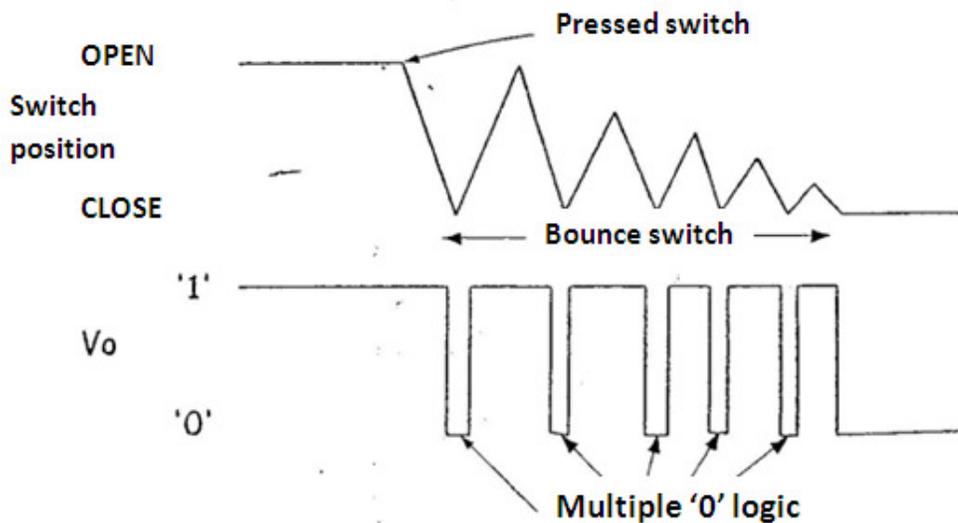
- Switch circuits for logic TTL/CMOS:



When not pressing the switch @ on, $V_o = '1'$

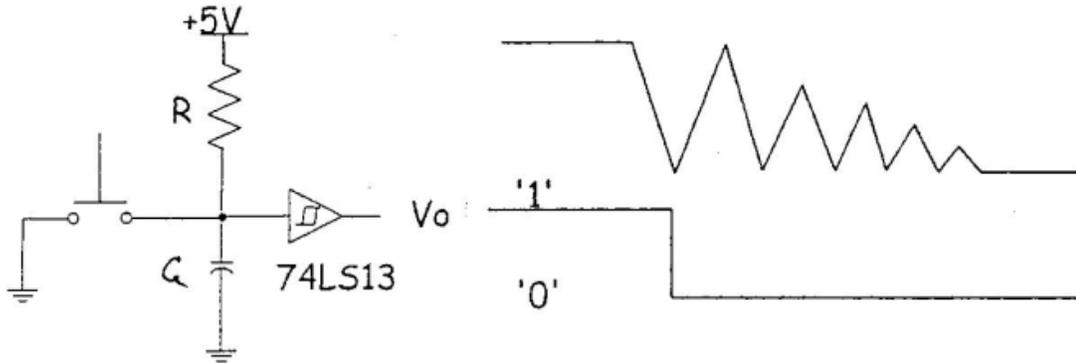
When pressing the switch @ off, $V_o = '0'$

- Bounce of mechanical switch- most mechanical switch bounce when switch off.



7.2 Interface with mechanical switch

- The time of bounce mechanical switch is 5-50ms. Bounce produce multiple '0' logic – giving the false reading to the μ controller!
Therefore, bounce/multiple '0' logic needs to vanish – **DEBOUNCING**
- **Debouncing** use below equipment:



**Choose the time constant RxC larger than bounce time. Example: $RxC = 100ms$ –
 $R = 100k$, $c = 1\mu F$.**

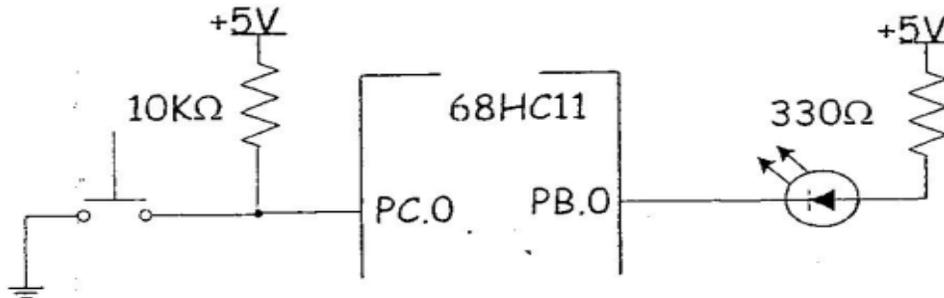
- Debouncing used following software:

Steps:

1. Give effect when pressing the switch and read the condition of logic switch
2. Wait/delay 20-25ms
3. Read the logic switch again
4. If the condition of logic switch \neq with the obtained result in step 1, repeat step 1 again

7.2 Interface with mechanical switch (cont...)

- Example: for circuit below, write a program to blink the LED for 5 times when the switch is pressed and then light off. While remain light off is the switch is not being pressed.



```

LDX      #$1000
LDAA     #%0000001      ; let PC0 to be input
STAA     DDRC, X
BSET     PORTB, X      $01  ; light off the LED

***
1. Effect of pressed switch ('0')
WAIT     BRSET     PORTC, X      %00000001  WAIT
***
2. Delay for 20ms
JSR      DELAY20

***
3. Read the switch again. If bounce, repeat 1.
BRSET     PORTC, X      %00000001  WAIT
***
4. Finish bounce

LDAA     PORTC, X      ; store switch condition in A
                                ; not used in this case

LDAB     #5
WAIT     BCLR     PORTB, X      %00000001  ; LED light on
JSR      DELAY500              ; wait 500ms
BSET     PORTB, X      %00000001  ; LED light off
JSR      DELAY500              ; wait 500ms
DECB
BNE
                                ; is it 5x?
                                ; not yet

BRA      WAIT                  ; repeat

DELAY20  :
RTS

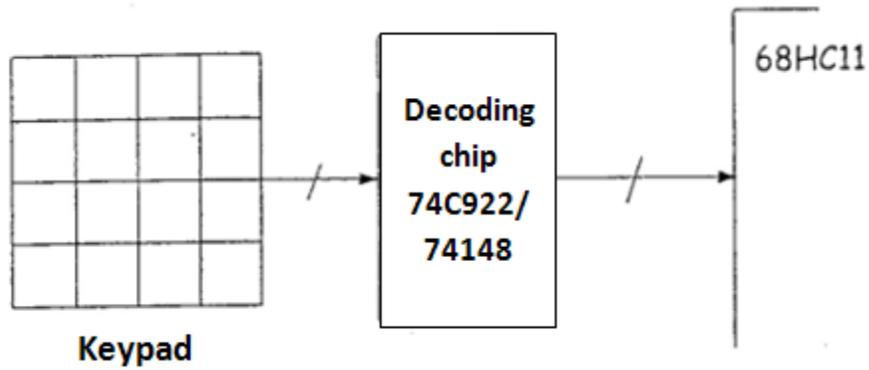
DELAY500 :
RTS

```

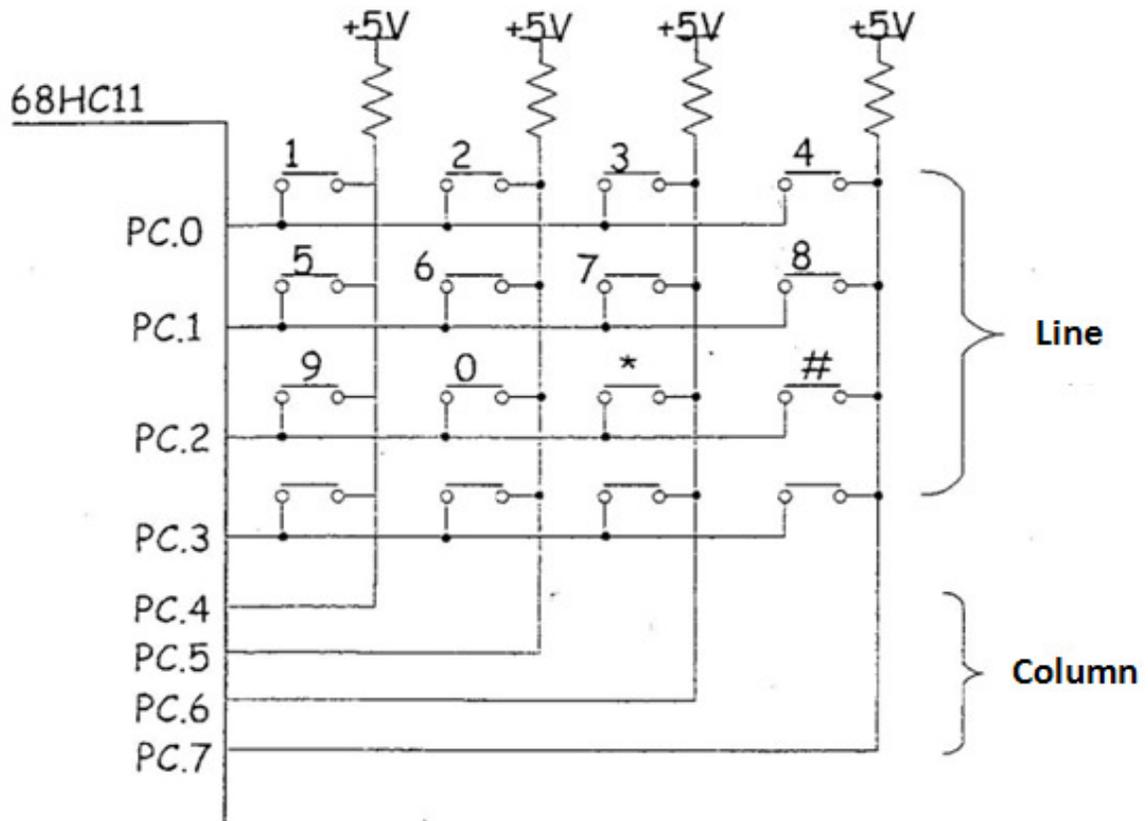
7.2 Interface with mechanical switch (cont...)

- Interface with keypad

Use equipment of decoding keypad:



Using software encoder:



7.2 Interface with Mechanical Switch (cont...)

- Interface with keypad...
Each key is characterized with column and row
Example: key 1 is a column 1 row 1 = L1B1

Key # is a column 3 row 3 = L3B3
Column (PC.4-PC.7) as output
Row (PC.0-PC.3) as input

If one column as logic '0' (other column logic '1'), any key that closed/pressed in that column will be produce logic '0' at row where the key are pressed.

To find the key are pressed:

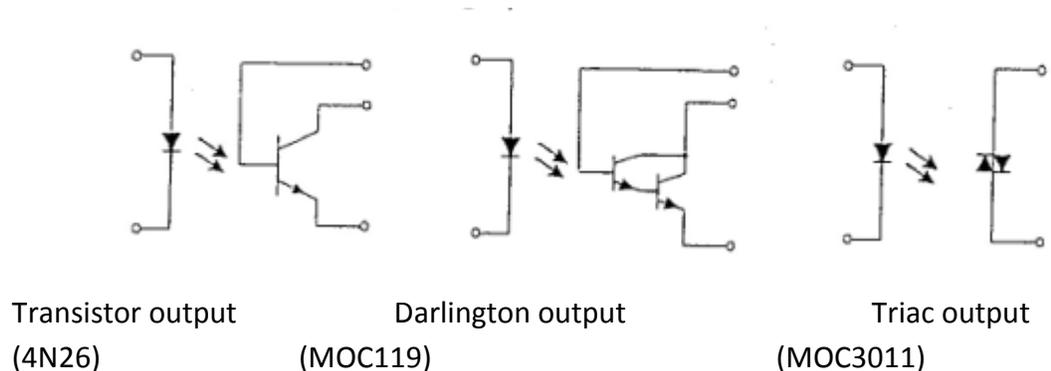
1. Make each column logic '0' rotation ($i = 1 \dots 4$)
2. Read each row rotation ($j = 1 \dots 4$)
3. Combination of column at logic '0' and row at logic '0' are used to determine the key are pressed
4. Do debouncing for that key ($L_i B_j$)

7.3 interface with High Current /Voltage Device

- When high current or voltage device is interfaced to microcontroller system, separation must to do between high current or voltage device and microcontroller system to avoid high current or voltage subject to microcontroller that can damage it.
- Optoisolator / optocoupler
The opto separator that use LED (as the sender) and photo detection (as the receiver) to send a signal through the light. LED is connected to output microcontroller port and output photo transistor is connected to high current or voltage device. Because of the LED and photo detection separate by air insulated, separator between microcontroller and device happen.

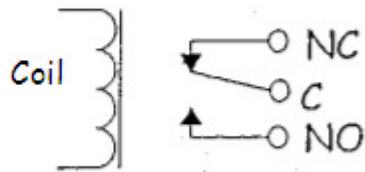
The opto separator useful for controlling device that quite far from microcontroller system and device that have separable power supply from microcontroller system.

Several types of opto separable:



7.3 interface with High Current /Voltage Device (cont...)

- The electromagnetic relay can be used to control high current or voltage device.
Structure of a relay:



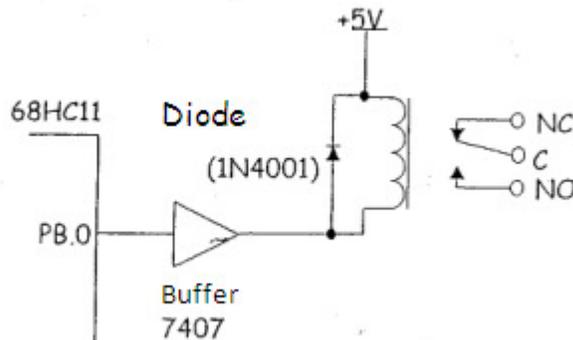
C – common base

NC – normally closed base

NO – normally opened base

Relay can control direct current or alternating current device. Need to select appropriate relay for a specific device.

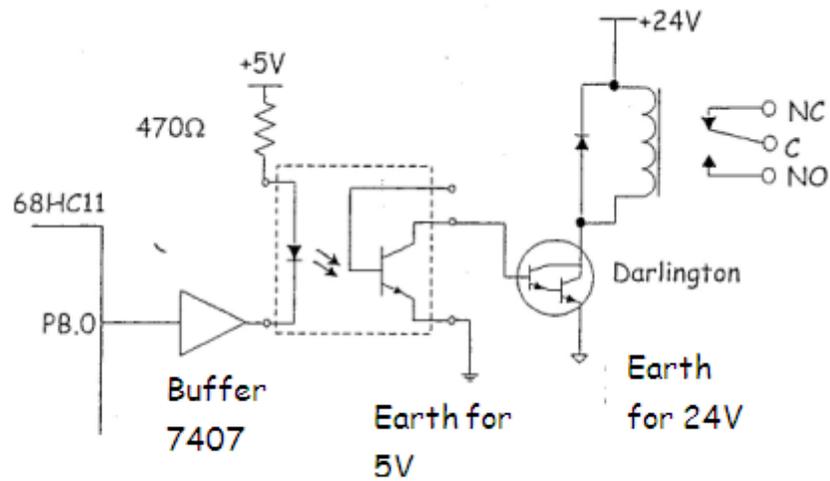
For relay that its coil can be function with low voltage (≤ 5 volt) and smaller current, the following circuit can be used:



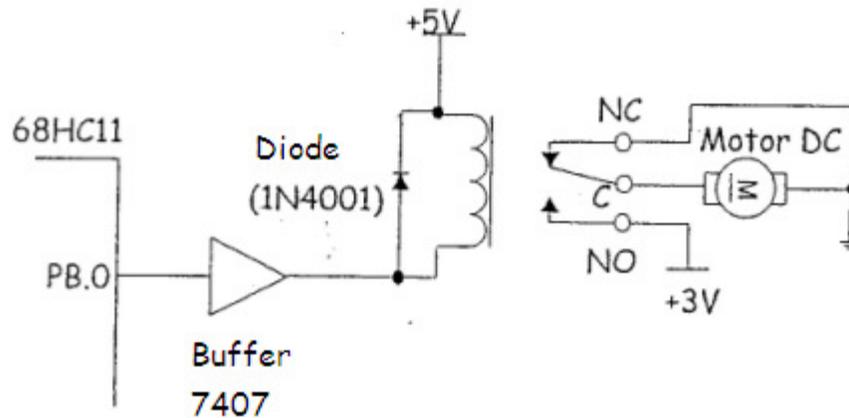
Diode is used to avoid induced negative voltage happen in the coil that can damage the buffer and 68HC11.

7.3 interface with High Current /Voltage Device (cont...)

For relay that its coil require high voltage (> 5 volt) and large current, or when relay output need connected to alternating current device, opto separator with relay need to used. For example, for relay with coil voltage +24V, the following circuit can be used:



Example: interfacing one small DC motor (+3V, 500mA) to 68HC11. DC motor cannot be connected directly to 68HC11 because current 500mA can burn the buffer/68HC11. Enable use relay +5V. When PB.0=’0’, motor stop and when PB.0 = ’1’, motor rotate (only one way!)



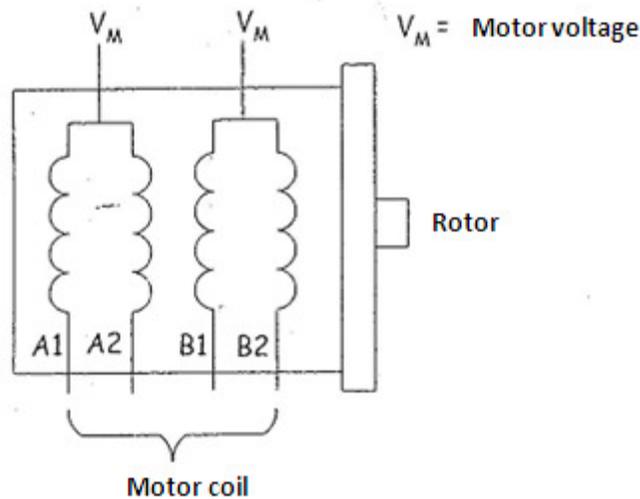
7.4 Interface with stepper motor

- Stepper motor is a motor that move step by step in a certain angle. Angles for a step are 15° and 7.5° . An angle for a step is 15° , one complete rotation (360°) needed 24 steps.
- Stepper motor use as printer, disk drives and robotic.

- The comparison between stepper motor and DC motor:

Stepper motor	DC motor
Move in steps	Rotate continuously
Can control in an open loop and no need of sensor to feedback the position	Controlled by the closed loop. Needed the position detector/velocity to feedback
The control software is easy to write	Its control software is complicated

Symbol:



7.4 Interface with stepper motor (cont...)

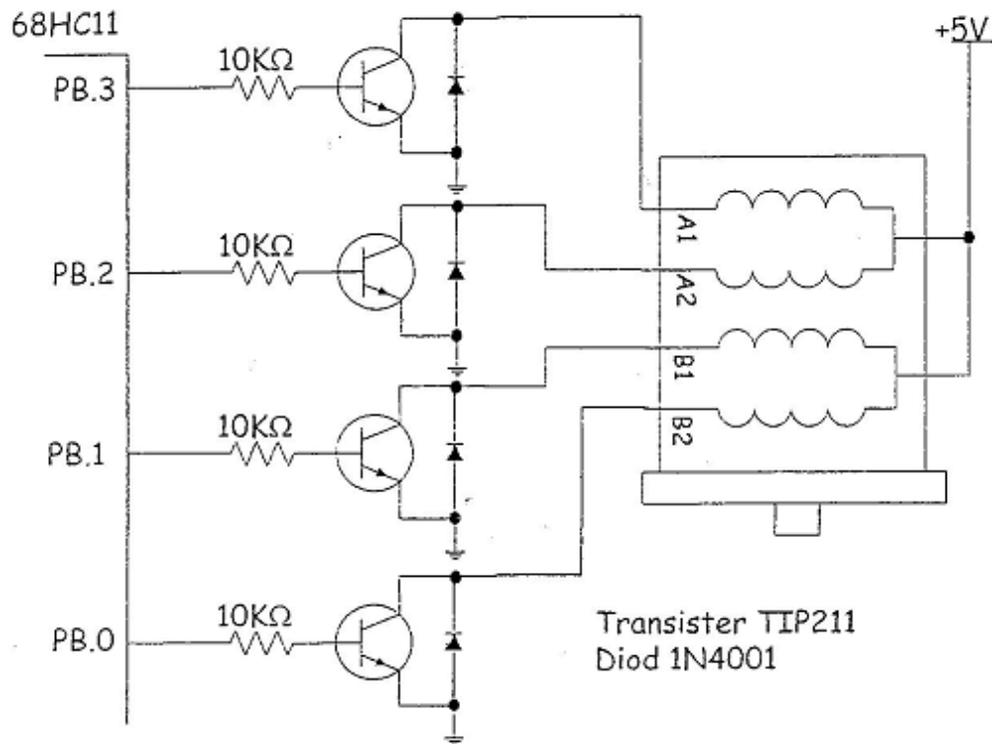
- Driving a stepper motor involved in the implement of voltages in the motor coil (A1, A2, B1 & B2) in the respective sequence. In the full-step mode, the sequence of voltage must apply to the motor coil.

Sequence	A1	A2	B1	B2	Code (Hex)
1	1	0	1	0	\$0A
2	1	0	0	1	\$09
3	0	1	0	1	\$05
4	0	1	1	0	\$06

'1' at A1, A2, B1, B2 means V_M V applied.

'0' at A1, A2, B1, B2 means 0V applied.

- The direct drive circuit: AIRPAX 5V stepper motor with angle of 7.5° per step.



7.4 Interface with Stepper Motor (cont...)

- To drive motor one step in one way, sequence code \$0A, \$09, \$06 was sent to port B (only nibble below). To drive motor one step in opposite direction, sequence code \$06, \$05, \$09, \$0A were send to port B.
- Usually one dawdle is done between changes in code above to allow motor achieve steady state and also to produce rotation in certain velocity.
- There are chip stepper motor driver as SAA1027, L297/298, UCN4204B and so on. Advantage usage of this chips are:
 - Easy interface circuit with microcontroller

- Easy to write program because the sequence code not necessary to know by the programmer.
 - The chips had current /voltage ability to drive stepper motor directly.
-
- Example: write a program to move stepper motor AIRPAX in the circuit before this as 60° in one way and next 15° in opposite way. This motor needs 10 ms to achieve steady state.

To move 60° need 8 steps and 15° need 2 steps. Assume subroutine LENGAH10 to dawdle 10ms still have.

- Example:

*a program example of stepper motor controller for SEM4233

*synopsis:

*sent 8x sequence \$0A, \$09, \$05, \$06 to PORTB

*sent 2x sequence \$06, \$05, \$09, \$0A to PORTB

*Assume nibble in is not used

*programmer: DR. ROSBI MAMAT

*date: 1/10/2000

```

LDX    #$1000    ;DAFTAR I/O BERMULA DI $1000
LDAB   #8        ;60° IKUT JAM
LDAA   #$0A
IKUTJAM STAA  PORTB,X
      JSR   LENGAH10
LDAA   #$09
      STAA PORTB,X
      JSR   LENGAH10
LDAA   #$05
      STAA PORTB,X
      JSR   LENGAH10
LDAA   #$06
      STAA PORTB,X    ;JUJUKAN 1 LANGKAH
      JSR   LENGAH10
DECB
      BNE  IKUTJAM    ;CUKUP 60° ?
LWNJAM LDAB  #2      ;15° LWN JAM
LDAA   #$06
      STAA PORTB,X
      JSR   LENGAH10
LDAA   #$05
      STAA PORTB,X
      JSR   LENGAH10
LDAA   #$09
      STAA PORTB,X
      JSR   LENGAH10
LDAA   #$0A
      STAA PORTB,X    ;JUJUKAN 1 LANGKAH
      JSR   LENGAH10
DECB
      BNE  LWNJAM    ;CUKUP 15° ?
      :
```

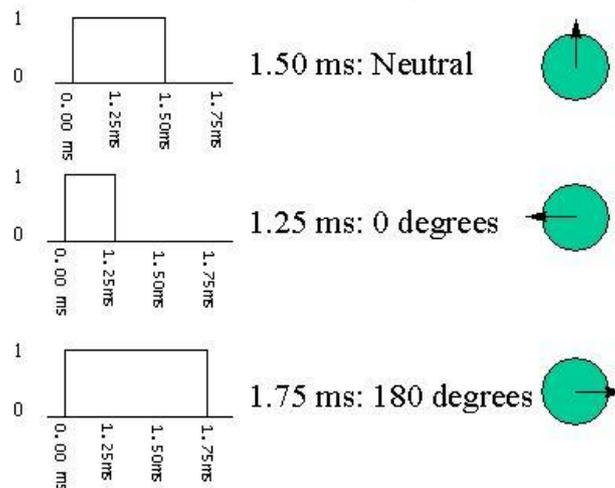
RC Servo motor



- RC Servo motor is normally used in the RC model (car, helicopter, plane)
- Useful for robot because cheap, compact and small. It consists of motor, gear box, electronic and feedback mechanism

Interface with RC servo motor

- RC Servo motor consists 3 pins of interface
 - RED = + 4.8 V
 - WHITE = Control pin
 - BLACK = Ground
- The angle is determined by the duration of a pulse that is applied to the control pin of every 20 milliseconds. This is called Pulse Coded Modulation.
- The length of the pulse will determine how far the motor turns. A 1.5 millisecond pulse, for example, will make the motor turn to the 90 degree position (often called the neutral position). If the pulse is shorter than 1.5 ms, then the motor will turn the shaft to closer to 0 degree. If the pulse is longer than 1.5ms, the shaft turns closer to 180 degrees.



source: <http://www.seattlerobotics.org/guide/servos.html>