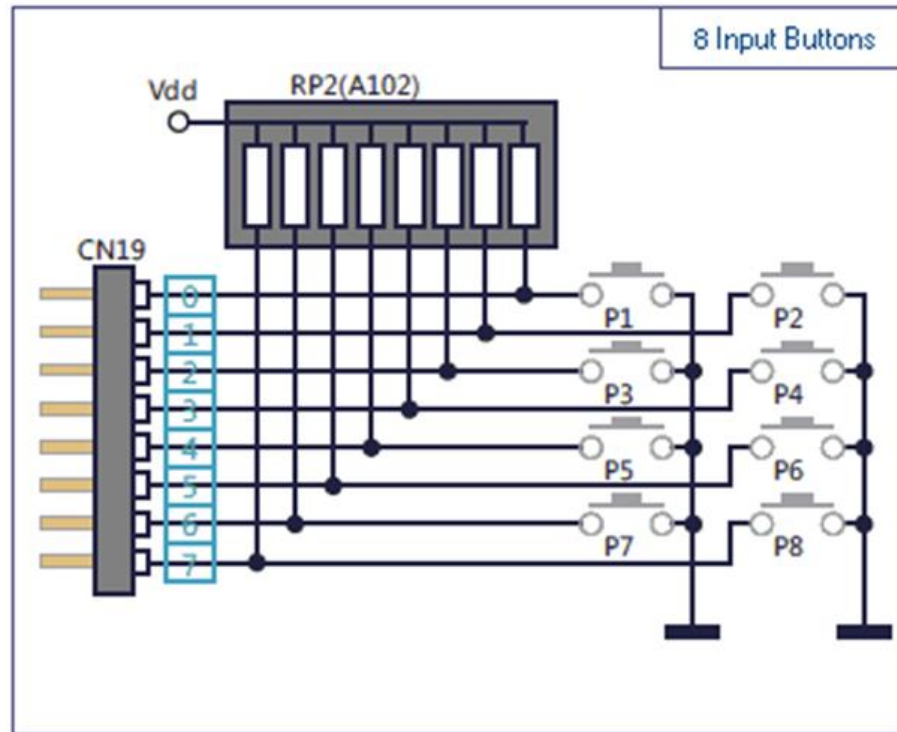


Programming for I/O devices

8 input buttons

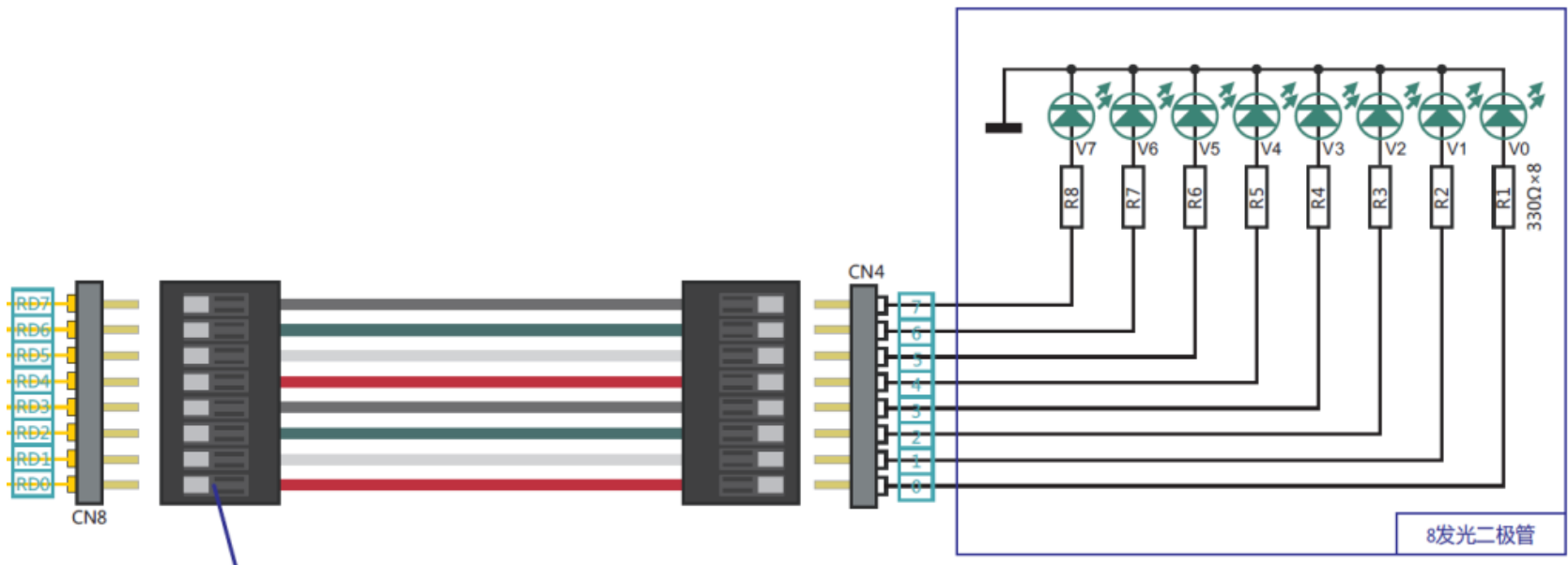
Press => logic 0, otherwise logic 1



Connection

Connect PORT D to the 8 LEDs

Connect PORT C to the 8 input buttons



Debouncing

Suppose the task is to toggle the LEDs once each time the P1 button is pressed.

Do forever

- Wait for button press

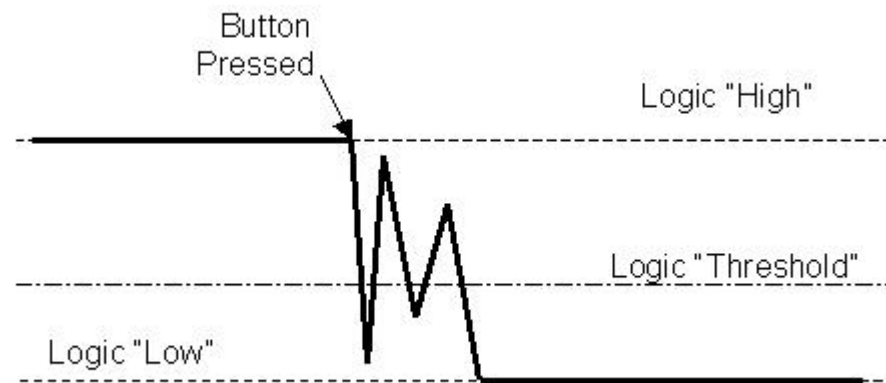
- Toggle LEDs

- Call a Delay (around 20 ms)

- Wait for button up

- Call a Delay (around 20 ms)

End



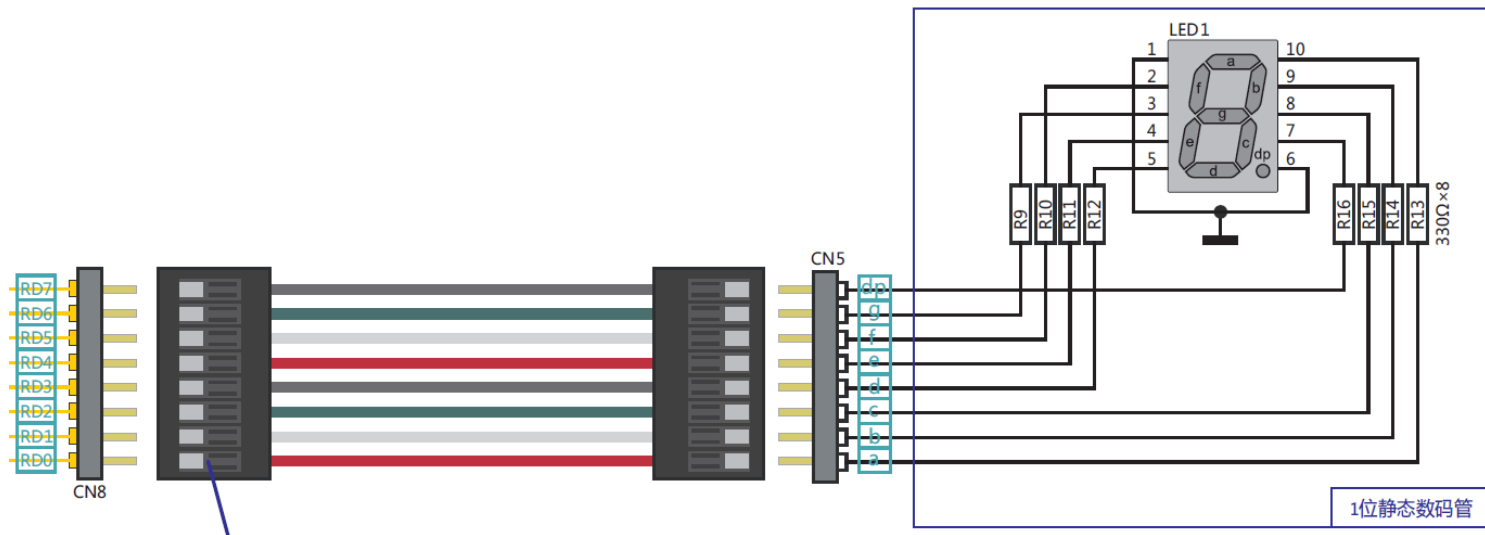
Debouncing example

| | | | |
|-------|----------|------------------|---------------------------|
| Main: | LIST | P=18F4520 | Loop: |
| | #include | <P18F4520.INC> | wait_press: btfsc PORTC,0 |
| | | CONFIG OSC = XT | bra wait_press |
| | | CONFIG WDT = OFF | comf PORTD,F |
| | | CONFIG LVP = OFF | call delay10 |
| | | CBLOCK 0x000 | wait_up: btfss PORTC,0 |
| | | DELAY_H | bra wait_up |
| | | DELAY_L | call delay10 |
| | | ENDC | goto Loop |
| | ORG | 0x0000 | |
| | goto | Main | |
| | ORG | 0x0030 | delay10: movlw 0x7f |
| | | | movwf DELAY_H |
| | movlw | 0x0f | Lop_1: movlw 0 |
| | movwf | ADCON1 | movwf DELAY_L |
| | clrf | TRISD | Lop_2: decf DELAY_L,F |
| | setf | TRISC | bnz Lop_2 |
| | movlw | 0x0aa | decf DELAY_H,F |
| | movwf | PORTD | bnz Lop_1 |
| | | | return |
| | | | END |

Enter and test the example program

Connect PORT D to the single 7-segment LED display

Connect PORT C to the 8 input buttons



Write a program to display a digit when the corresponding button is pressed. For instance, display digit “1” when P1 is pressed.

Pseudo code

Set up I/O ports

Initialize the display (display digit “0”)

Loop:

 if P1 is pressed

 display digit “1”

 delay

 wait for button up

 delay

 goto Loop

 elseif P2 is pressed

 display digit “2”

 delay

 wait for button up

 delay

 goto Loop

....