Hardware V2

Digital I/O

pin

- Logical low = 0
- Logical high = 1

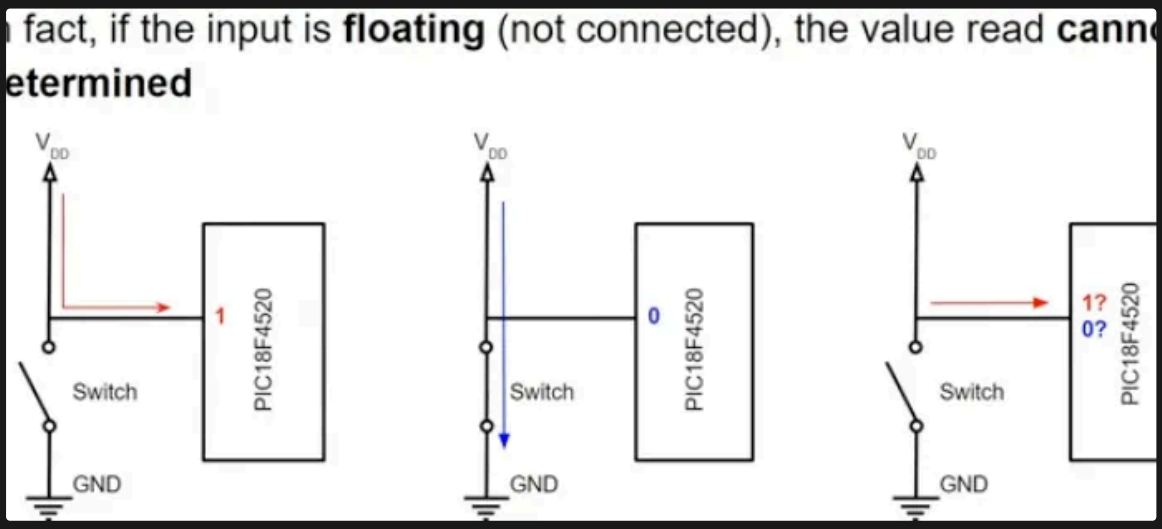
Each line can be programmer as :

- an output (it generate a current and can be used , for example , to lit a LED)
- an input (it receives a current and can be used, for example, to read the push button)

Digital Input : Electrical Consideration

Issue

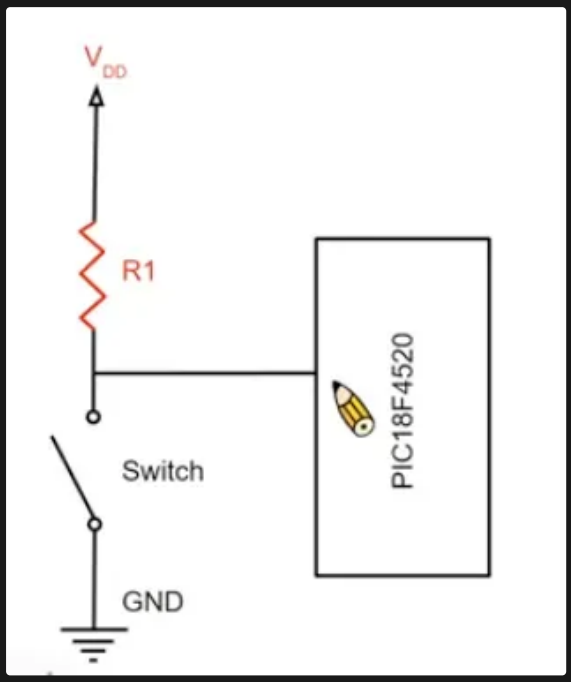
- Ideal condition:
  - An input connected to V_{dd} is read as “1”
  - An input connected to Ground is read as “0”



  - 因為電路上沒有電阻，所以 電流容易受環境的電波所影響

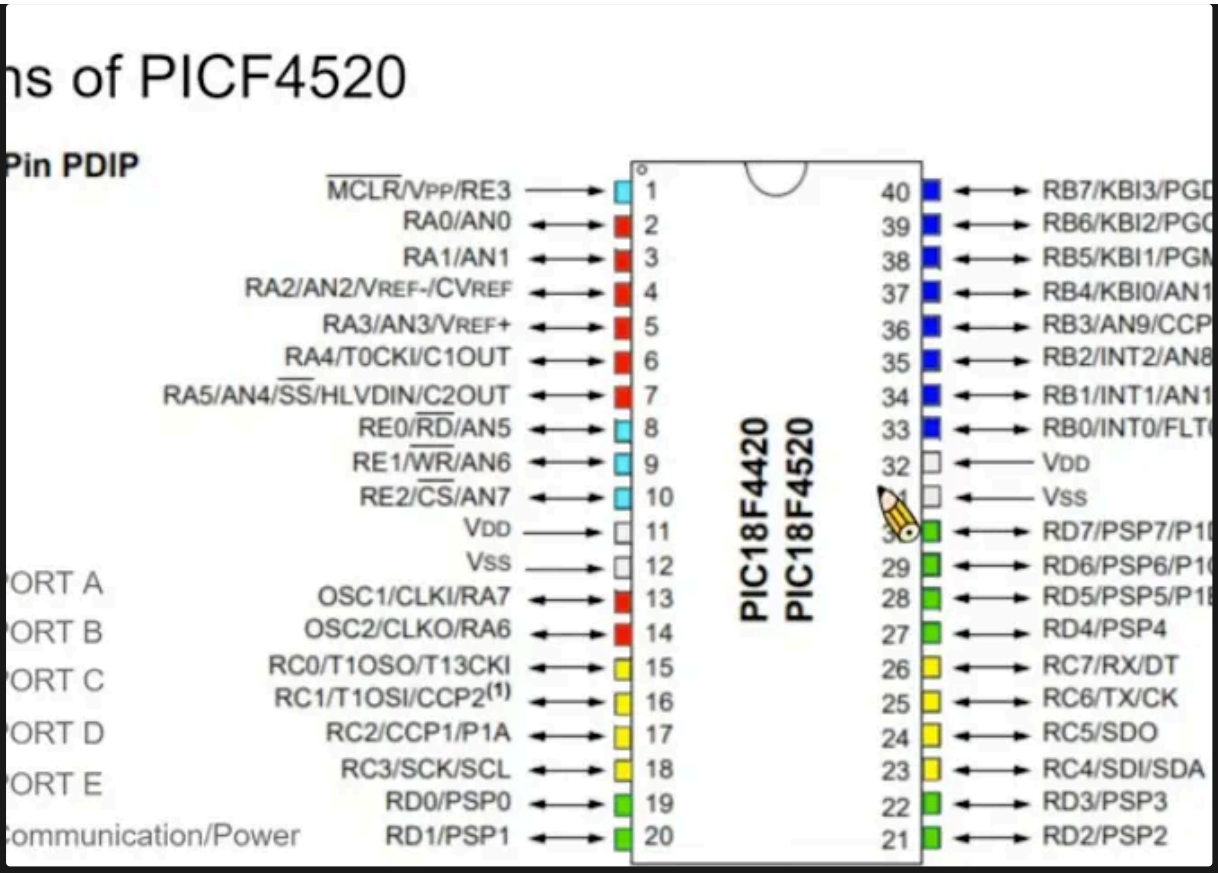
Solution

- Pull-Up : 電阻接input 與 V_{dd} 之間
  - Switched opened = 1 , closed = 0



- Pull-down: 電阻接在 ground 與 input 之間
  - Switched opened = 0 . closed = 1

Port



I/O ports control register:

- Each I/O port has three register for its operation, where x is a letter that denoted
  - TRISx : Control the direction of the PORTx pins
    - Inward (1) : input
    - Outward(0) : output
    - Default : all input (1111 1111)
  - Port: 用來存取pin腳資料
    - port如果是input 就是read date
    - port如果是output 就是要write data
  - LATx :用來eliminate the problem that could occur with read-modify write instructions

	PORTx	LATx
	Reads data value **on the I/O pin**	Reads data value held **in the port l**
	Writes data value to the port latch	Writes data value to the port latc

Initialized

- ADCON1 ：設定為0x0F 才會是Digital I/O

U-0	R/W-0	R/W-0	R/W-0	R/W-q(1)	R/W-q(1)	R/W
—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PC

bit 3-0 **PCFG<3:0>:** A/D Port Configuration Control bits:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7(2)	AN6(2)	AN5(2)	AN4	AN3	AN2	AN1	AN0
0000(1)	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111(1)	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

- RBPU(INTCON2<7> = 0) : turn on all the pull-ups

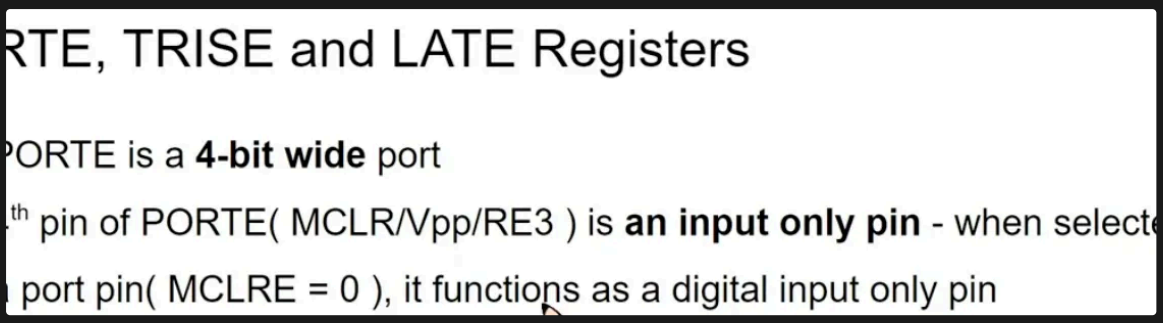
RTB, TRISB and LATB Registers

Each of PORTB pin has a **weak internal pull-up**

- RBPU (INTCON2<7> = 0) can turn on all the pull-ups
- automatically turned off when the port pin is configured as an ou

```
CLRF    PORTB   ; Initialize PORTB by
                ; clearing output
                ; data latches
CLRF    LATB    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0Fh     ; Set RB<4:0> as
MOVWF   ADCON1  ; digital I/O pins
                ; (required if config bit
                ; PBADEN is set)
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISB   ; Set RB<3:0> as inputs
                ; RB<5:4> as outputs
                ; RB<7:6> as inputs
```

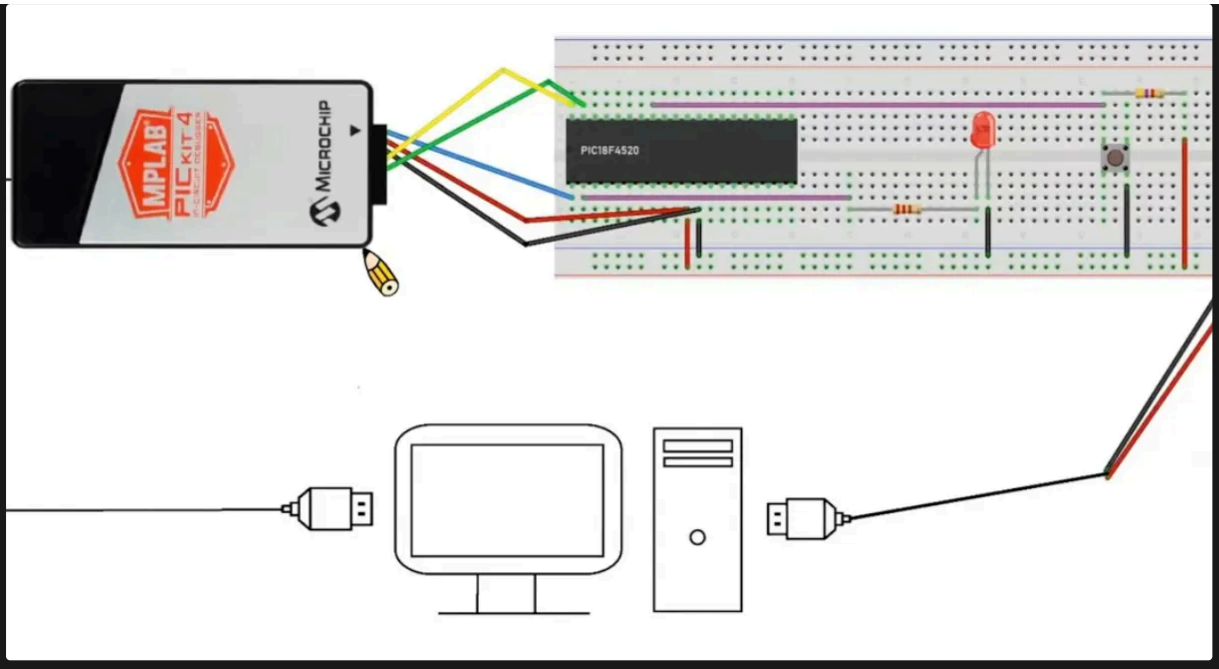
- PORTE : 也要ADCON1 = 0x0A



Bit operation

- Single bit manipulation
  - BCF f ,b : 把f register 的b bits 清空
  - BSF f , b : 把f register 的 b bits 設定
  - BTG f , b : 把f register 的 b bits 反轉
- Multiple bits manipulation:

燒錄器接角



Interrupt

When an interrupt occurs, the processor″

- Global interrupt: GIE(global interrupt enable bit at INTCON<7>) ,要設成1
- Interrupt priority: 有分高低，高priority來低priority就要被打斷，(IPEN bit in RCON<7>)
- Interrupt flag bit: 程式接受到interrupt ，將flag bit 設為1 , 避免遞迴叫interrupt
- ISR結束 : return from interrupt: 會將interrupt routing and 把GIE 設回1

Control Interrupt

Bits to control interrupt:

- Flag bit : 確認interrupt 現在有沒有發生
- Enable bit: allow program executions to branch to the interrupt vector address when the flag bit is set
- Priority bit: to select high priority or low priority

RCON

ON :

REGISTER 9-10: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1(1)	U-0	R/W-1	R-1	R-1	R/W-0(1)	R/W
IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BO}$
bit 7							

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

INTCON

TCON :

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all interrupts

Timer

```
tick_time = prescaler * (1 / (Fosc/4))
```

▼ OSCCON

PIC18F4520 系統時鐘是由OSCCON控制

- IRFC (Internal Oscillator Frequency Select)

```
//常用設定 OSCCONbits.IRCF = xxx; OSCCONbits.SCS = 0b10; //
111 = 8MHz // 110 = 4MHz // 101 = 2MHz // 100 = 1MHz // 011 =
500kHz // 010 = 250kHz // 001 = 125kHz // 000 = 31kHz
```

- Timer 用的 tick 是 Fosc/4 :

  ex: Fosc = 4 MHz → Timer tick = 1MHz = 每 1us 跳一次

▼ Timer knowledge

`Timer0：1:2 ~ 1:256`

`Timer1：1,2,4,8`

`Timer2：1,4,16`

`Timer3：1,2,4,8`

- Timer1 特性
  - 16-bits(TMR1H + TMR1L) →最大65536 ticks
  - 時鐘來源可以來自：Fosc / 4
  - Prescalar : 1 (0b00), 2(0b01) , 4(0b10) , 8(0b11)

```
// timer1 initialize() void Timer1_Init(void){
T1CONbits.RD16 = 1; // 16-bit read/write T1CONbits.T1RUN
= 0; // Internal clock T1CONbits.TMR1CS = 0; // Clock =
Fosc/4 T1CONbits.T1CKPS = 0b00; // Prescale = 1
Timer1_Load_500ms(); PIR1bits.TMR1IF = 0; PIE1bits.TMR1IE
= 1; // Enable Timer1 interrupt IPR1bits.TMR1IP = 0; //
Low priority T1CONbits.TMR1ON = 1; // Start Timer1 }
//button initialize void Button_Init(){ TRISBbits.TRISB0
= 1; // input INTCON2bits.INTEDG0 = 0; // Falling edge
INTCONbits.INT0IF = 0 ; // clear flag INTCONbits.INT0IE =
1; } //initial interrupt void Interrupt_Init(void){
RCONbits.IPEN = 1; // Enable priority levels VERY
IMPORTANT !! INTCONbits.GIEH = 1; // Enable high-priority
interrupts INTCONbits.GIEL = 1; // Enable low-priority
interrupts INTCONbits.INT0IE = 1; // INT0 always high
priority by hardware (fixed) } // button interrupt void
__interrupt(high_priority) HighISR(void){ // ------ INT0
Button Interrupt ------ if(INTCONbits.INT0IF){
__delay_ms(20); INTCONbits.INT0IF = 0; // Clear flag
button_flag = 1; } } //timer1 intterupt void
__interrupt(low_priority) LowISR(void){ // ------ Timer1
Interrupt every 0.5 sec ------ if(PIR1bits.TMR1IF){
PIR1bits.TMR1IF = 0; Timer1_Load_500ms(); timer_flag = 1;
} }
```

```
//常用設定 T1CONbits.TMR1CS = 0; // clock = Fosc/4
T1CONbits.T1CKPS = 0b11; // prescaler 1:8 T1CONbits.RD16
= 1; // 16-bit access TMR1H = 0; TMR1L = 0; // reset
T1CONbits.TMR1ON = 1; // enable //Prescalar // 0b00 1:1
// 0b01 1:2 // 0b10 1:4 // 0b11 1:8
```

- Timer2特性
  - 8 bits TMR2 (0~ 255)
  - 搭配PR2 決定reset 的位置(週期) →可設定計時「週期」
  - 搭配固定週期的PWM ，例如：
    - Servo PWM
    - LED 調光
  - Prescalar = 1 , 4. , 16
  - Postscalar = 1~16 (只影響中斷)
  - Timer2 計時公式

```
T = (PR2 + 1) * Prescalar * 4/Fosc // TMR2 tick =
PRescalar * (4/ Fosc) // Timer2 overflow = (PR2 + 1) ticks
PWM_period = (PR2 + 1) * 4 * prescalar
```

```
T2CONbits.T2CKPS = 0b10; // 1:16 prescale PR2 = 249; // 設
定週期 TMR2 = 0; T2CONbits.TMR2ON = 1; // start
```

用途	原因
CCP 的 PWM（Servo）	CCP PWM 只能用 Timer2！
DC motor PWM	高精度固定週期
LED 漸亮漸暗	很好控制 duty cycle

CCP Module(Capture / Compare / PWM)

- PIC18F4520 的每個CCP 模組都有
  - CCPxCON(控制寄存器)
  - CCPRx(資料寄存器 = CCPRxH + CCPRxL , 16 bits)

- CCP 與Timer 的對應關係

  不同CCP mode 依賴不同Timer

Mode	Timer
Capture	Timer1 或 Timer3
Compare	Timer1 或 Timer3
PWM	**Timer2**

- Capture 模式(量測外部訊號時間)

  Capture mode做什麼 ：外部腳位(如CCP1 = RC2) 偵測到事件時，把當下的timer值「擷取」到CCPRxH : CCPRxL

  可擷取事件：每次falling edge , 每次 rising edge , 每四次 rising edge , 每16次 rising edge

  重點：

  - CCP 腳位要設成input(TRISCbits.TRISC2 = 1)
  - 擷取後會set CCPxIF