

Cognoms i Nom: _____ UNA POSSIBLE SOLUCIÓ _____ Doc. Identitat: _____

Totes les respostes han d'estar degudament justificades

1. Amb el propòsit de disposar d'una entrada digital i el seu complementari, un estudiant proposa el circuit adjunt, on A està connectat al pin RA0 i B al pin RA1.

Indiqueu en una taula la tensió (V_A i V_B) i el valor lògic en A i B en funció de l'estat (obert/tancat) del pulsador –Els pins RA0 i RA1 estan configurats com a entrada digital–.

El circuit proposat és una opció adequada ?

Sinó és adequada, proposeu una alternativa vàlida.

(1,5 punts)

Considereu els següents valors típics:

$$V_{DD} = 5 \text{ V}, \quad V_{SS} = 0 \text{ V}, \quad V_{ILmax} = 0.8 \text{ V}, \quad V_{IHmin} = 2 \text{ V}, \quad I_{Lmax} = \pm 200 \text{ nA}$$

$$V_{OLmax} = 0.6 \text{ V}, \quad V_{OHmin} = 4.3 \text{ V}, \quad I_{Omax} = \pm 25 \text{ mA}$$

Pulsador Obert

$V_A \approx 5 \text{ V} - 7 \text{ k} \cdot 200 \text{ nA} = 4,98 \text{ V}$. Com $4,98 \text{ V} \geq V_{IHmin}$ tenim un "1" lògic a A.

$V_B \approx 3 \text{ k} \cdot 200 \text{ nA} = 0,6 \text{ mV}$. Com $0,6 \text{ mV} \leq V_{ILmax}$ tenim un "0" lògic a B.

Pulsador Tancat

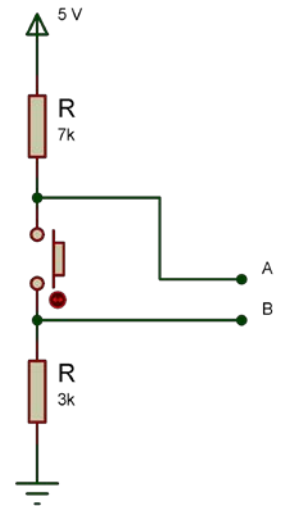
A i B són el mateix punt elèctric. Tenim les dues resistències

formant un divisor de tensió, per tant $V_A = V_B = (3 \text{ k} / (7 \text{ k} + 3 \text{ k})) \cdot 5 \text{ V} = 1,5 \text{ V}$

Com $V_{ILmax} < 1,5 < V_{IHmin}$, ens trobem en el cas que aquesta tensió no correspon ni a un valor vàlid per un "0" lògic, ni a un valor vàlid per un "1".

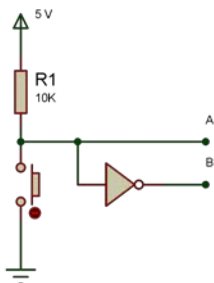
Resum

Pulsador	V_A	A	V_B	B
Obert	$\approx 5 \text{ V}$	H	$\approx 0 \text{ V}$	L
Tancat	1,5 V	Indefinit	1,5 V	Indefinit



Aquest circuit no funciona correctament per generar un senyal digital, ja que quan el pulsador està tancat, les tensions V_A i V_B no corresponen ni a '0' lògic ni '1' lògic i per tant el comportament del circuit serà indeterminat (no es pot establir si els 1,5 V seran interpretats pel microcontrolador com un '0' o com '1').

Les alternatives possibles són: 1) Afegir una porta NOT o 2) Generar el valor complementari per software ($A = !B$)

Opció 1Opció 2

$A = \text{PORTAbits.RA0};$

$B = !A;$

En l'opció 1, en els moments de canvi de A, els dos senyal (A i B) tenen el mateix valor lògic degut al temps de propagació de la porta NOT (t_{pNOT}). L'opció 2 no presenta aquest problema, però implica afegir algunes línies de codi addicionals al programa.

2) Descriu detalladament com funciona la jerarquia d'interrupcions en el PIC18F4550 en funció del bit IPEN. (1 punt)

El bit IPEN activa o desactiva el sistema de prioritats de interrupcions.

Si $IPEN = 0$ no s'aplica cap sistema de prioritats, és a dir, cap interrupció pot interrompre l'execució de la rutina d'atenció a la interrupció. Amb $IPEN = 0$ totes les interrupcions es tracten en la rutina d'atenció a la interrupció d'alta prioritat. Les interrupcions es poden habilitar/deshabilitar de forma global amb el bit GIE, i les interrupcions provinents dels perifèrics es poden habilitar/deshabilitar amb el bit PIE.

En el cas que $IPEN = 1$, una interrupció d'alta prioritat pot interrompre l'execució d'una rutina d'atenció a una interrupció de baixa prioritat. En aquest cas, dues interrupcions amb la mateixa prioritat tampoc es podran interrompre. El nivell de prioritat de cada interfície queda definit pel bit associat IP (Interrupt Priority). Cada nivell de prioritat és pot habilitar o deshabilitar de forma independent amb els bits GIEH i GIEL.

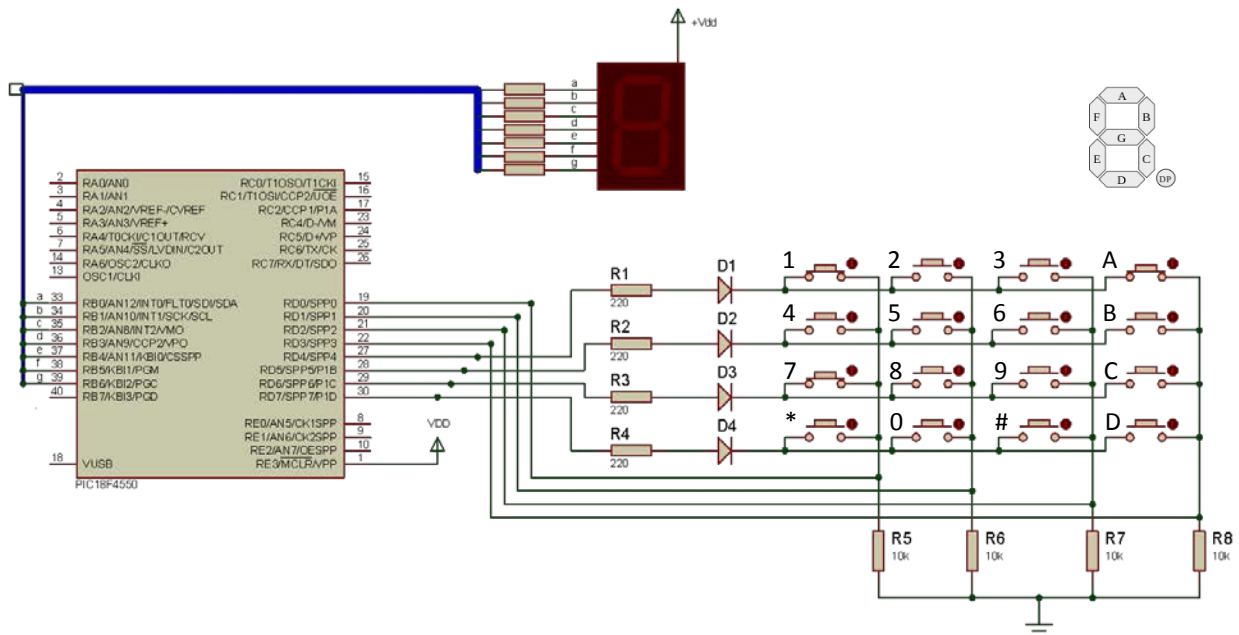
3) A partir del conjunt teclat – display set-segments utilitzat al laboratori:

(3 punts)

a) Configureu els registres necessaris per a que els bits dels ports B i D d'E/S realitzin la funció requerida.

b) Escriu el codi d'una rutina que retorni un valor de 0 a 15 en funció de la tecla pulsada. Si no es polsa cap tecla, la rutina retorna el valor 255 (considereu que hi ha com a màxim una tecla pulsada).

c) Escriu el codi necessari per mostrar el dígit 2 en el display set segments.



a)

El port B ha d'estar configurat com sortida $TRISB = 0x00$

El 4 bits més baixos de port D han d'estar configurats com entrada, i els 4 bits més alts com a sortida $TRISD = 0x0F$

b) Programa per fer Scan matrix del teclat 4x4 de la figura

`char mapa_tecles[16]={'1','2','3','A','4','5','6','B','7','8','9','C','*','0','#','D'};`

```
void llegir_teclat() {
    unsigned char i, columna;
    i = 0;
    while (i < 4) {
        PORTD = (0x10 << i); // Activem fila
        delay_ms(5); // esperem la propagació del senyal i que les entrades RD0 .. RD3 siguin estables
        columna = PORTD & 0x0F; // Llegim columna
        if (columna) break; // Tecla pulsada
        i++; // Seguent fila
    }
    if (columna) return mapa_tecles[i*4+columna]; // Si fila <> 0 retornem el char associat a la tecla pulsada
    else return 255; // No hi ha tecla pulsada
}
```

c) Per mostrar un '2' al 7-segments cal encendre els LEDs a, b, d, e i g, i per tant cal fixar un '0' en els pins de sortida associats (el 7-segments és d' ànode comú), i '1' en els pins de sortida associats als LEDs c i f.

$PORTB = 0b10100100 = 0xA4 = 164$ (el bit 7 és indiferent ... ho deixem a '1' –no actiu en aquest cas-)

Cognoms i Nom: _____ UNA POSSIBLE SOLUCIÓ _____ Doc. Identitat: _____

Totes les respostes han d'estar degudament justificades

4) Explica el motiu pel qual per llegir el valor del TMR0 primer s'ha de llegir la part baixa (TMR0L) i després la part alta (TMR0H) i quan volem escriure un valor en el TMR0 primer hem d'escriure la part alta (TMR0H) i després la part baixa (TMR0L). (1 punt)

El valor del TMR0 no és accessible directament. S'hi accedeix a través del TMR0H que actua com a buffer. Quan es llegeix el valor del TMR0L el valor de la part alta del registre passa al registre TMR0H i és per això que en una lectura primer hem d'accedir a la part baixa.

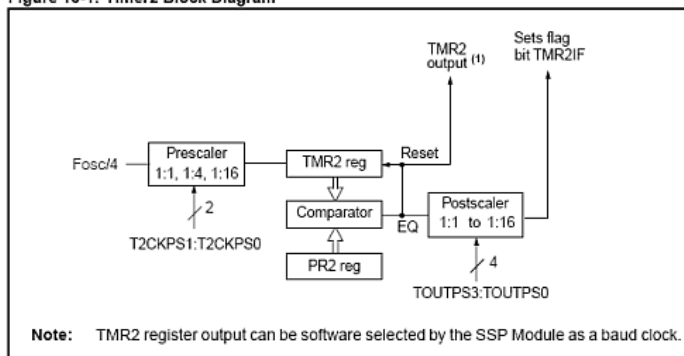
Quan fem una escriptura del valor TMR0L també es carrega la part alta del registre TMR0 amb el valor de TMR0H. Per tant primer hem d'estar segurs de que el valor de TMR0H és el correcte abans de fer l'escriptura del TMR0L.

5) En referència als Timers del PIC18F (3,5 punts)

a) Volem generar un senyal quadrat periòdic de 2Khz mitjançant el Timer2 d'un PIC18F4550 que treballa amb un oscil·lador a 8MHz. Considerem que ja hem configurat i programat una rutina d'interrupció que complementa un pin d'un port d'E/S pel que volem generar aquest senyal.

Quins valors hem de fixar al prescaler, al registre PR2 i al postscaler per tal de generar aquest senyal amb la màxima exactitud?

Figure 13-1: Timer2 Block Diagram



REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0
bit 6-3	T2OUTPS3:T2OUTPS0: Timer2 Output Postscale Select bits						
	0000 = 1:1 Postscale						
	0001 = 1:2 Postscale						
	•						
	•						
	1111 = 1:16 Postscale						
bit 2	TMR2ON: Timer2 On bit						
	1 = Timer2 is on						
	0 = Timer2 is off						
bit 1-0	T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits						
	00 = Prescaler is 1						
	01 = Prescaler is 4						
	1x = Prescaler is 16						

La freqüència del senyal que volem generar és de 2 KHz i per tant el seu període és de $1/2000 = 0,0005$ segons.

Com que haurem de canviar el valor del pin de sortida cada mig període haurem de provocar un canvi (una interrupció) cada $0,00025$ segons.

La freqüència d'entrada del timer2 és de $F_{osc}/4$ per tant té un període de $0,5$ microsegons.

Per tal de poder mesurar $0,00025$ segons necessitem comptar fins a:

$$0,00025 / (0,000005) = 500;$$

Com que el valor 500 supera el valor màxim que podem guardar en 8 bits necessitem utilitzar un prescaler. Podem fer servir el prescaler de 4 i llavors el registre PR2 haurà de valdre 125 o bé utilitzar el prescaler de 16, i en aquest cas caldria que el registre PR2 valgués 31 (valor exacte seria 31,25, i per tant tenim més exactitud amb el prescaler de 4).

b) Configura adequadament els bits necessaris per a que cada cop que s'activi el flag TRM2IF s'executi la rutina d'atenció a la interrupció d'alta prioritat

Registre RCON - IPEN = 1 activem jerarquia d'interrupcions

Registre INTCON

GIE = 1 activem a nivell general les interrupcions

PEIE = 1 activem les interrupcions del grup de perifèrics

Registre PIE1 - TMR2IE = 1 activem la interrupció del TIMER2

Registre IPR1 - TMR2IP = 1 configurem la interrupció del TIMER2 com d'alta prioritat

c) Escriu en llenguatge C el codi de la rutina d'atenció a la interrupció que implementa la part A del problema suposant que hem de treure el senyal pel pin RC3 que ja ha estat configurat com pin digital de sortida.

```
void interrupt RSI_INT_TIMER2() {
    if (TMR2IF && TMR2IE) {        // Cal tractar la interrupció del Timer2 ?
        TMR2IF = 0;                // baixem el flag
        RC3 = !RC3;                // toggle del bit

    }                               // el valor del Timer es reseteja a 0 automàticament
}                                   // el valor del registre PR2 no varia ni cal reprogramar-lo
```

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⁾

REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF

REGISTER 9-6: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

REGISTER 9-8: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
SPPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP

REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP

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 ⁽²⁾	R/W-0
IPEN	SBOREN	—	R _i	T _O	P _D	P _{OR}	B _{OR}

REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
SPPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF