

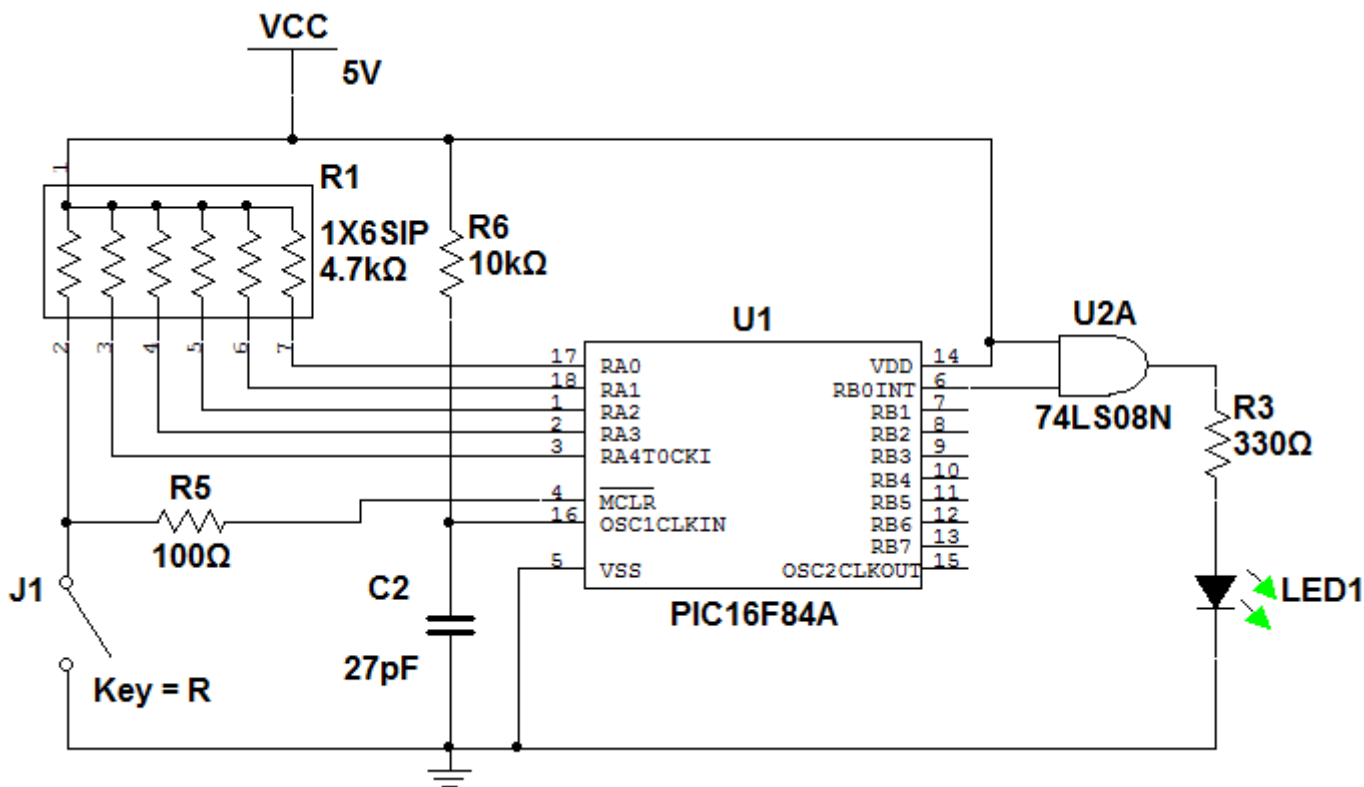
ITIS _____ _____	<h2 style="margin: 0;">LABORATORIO SISTEMI ELETTRONICI</h2>	Esercitazione N° _____ Data: _____
CLASSE IV _____ Gruppo N° _____	Alunno: _____	Anno scolastico 2009/10 Docenti: Prof. _____ Prof. _____

<b>Giudizio</b>	Conoscenza	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D	Frequenza	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D	<b>Giudizio complessivo</b>	<input style="width: 50px; height: 30px;" type="text"/>
	Competenza	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D	Impegno	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D		
	Capacità	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D	Comportamento	<input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D		

*Oggetto dell'esercitazione:*

### MICROCONTROLLORI PROGRAMMABILI PIC 16F84A: LED lampeggiante

Utilizzando MULTISIM, realizzare il circuito riportato in figura, quindi impostare il codice di programmazione e verificarne il funzionamento



Quando viene inserito il modulo MCU PIC16F84A viene chiesto di immettere un nome per il WORKSPACE, inserire Sistemi1 e cliccare su NEXT, per PROGRAMMING LANGUAGE scegliere ASSEMBLY e cliccare su NEXT, selezionare ADD SOURCE FILE (main.asm) e cliccare su FINISH.

Dopo aver realizzato il circuito, andiamo nella cartella Documenti\National Instruments\Circuit Design Suite 10.0\MCU Workspaces\PIC\project1 e apriamo il file main.asm per inserire il codice di programmazione.

```

;*****
; Programma di prova  Alterna 1 e 0 sul primo bit di PORTB ogni 100ms
;*****
#include "P16F84A.inc"      ;include le definizioni del PIC16F84A per l'assemblatore (MPASM)
errorlevel -302           ; will prevent assembler's warning message 302
;direttive
PROCESSOR 16F84A
RADIX     HEX             ;dove non indicato i valori sono HEX
ORG      0x0c            ;assembla a partire dalla locazione della memoria istruzioni
;definizioni
R1 EQU 0x0c
R2 EQU 0x0d
R3 EQU 0x0e
;inizio programma
bsf STATUS,5             ;selezione banco RAM 1
movlw 0x00               ;imposta tutte le linee
movwf TRISB              ;della PORTA B come output
bcf STATUS,5             ;selezione banco RAM 0
loop:
bsf PORTB,0              ;imposta a 1 il primo bit di PORTB
call wait                ;attesa
bcf PORTB,0              ;imposta a 0 il primo bit di PORTB
call wait                ;attesa
goto loop                ;ritorna a loop:
wait:                    ;routine di attesa (circa 100ms)
movlw 0xBE               ;carica nel registro w il valore BE (HEX)
movwf R1                 ;copia il valore di w nella variabile R1
longloop1:
movlw 0xBE               ;carica nel registro w il valore BE (HEX)
movwf R2                 ;copia il valore di w nella variabile R2
longloop2:
decfsz R2,1              ;viene decrementata di 1 la variabile R2
goto longloop2           ;fino a ch  R2 non diventa 0
decfsz R1,1              ;viene decrementata di 1 la variabile R1
goto longloop1           ;fino a ch  R1 non diventa 0
return
END                       ;fine del programma

```

**Relazionare sull'esercitazione svolta descrivendone i risultati ottenuti:**

---



---



---



---



---



---



---



---



---



---

