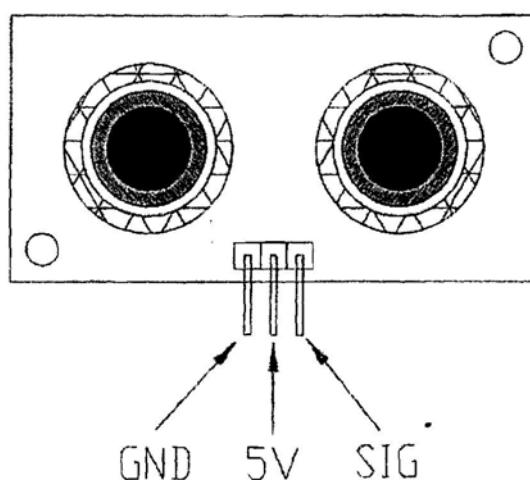



Ministero dell'Istruzione, dell'Università e della Ricerca
M272 – ESAME DI STATO DI ISTITUTO TECNICO INDUSTRIALE

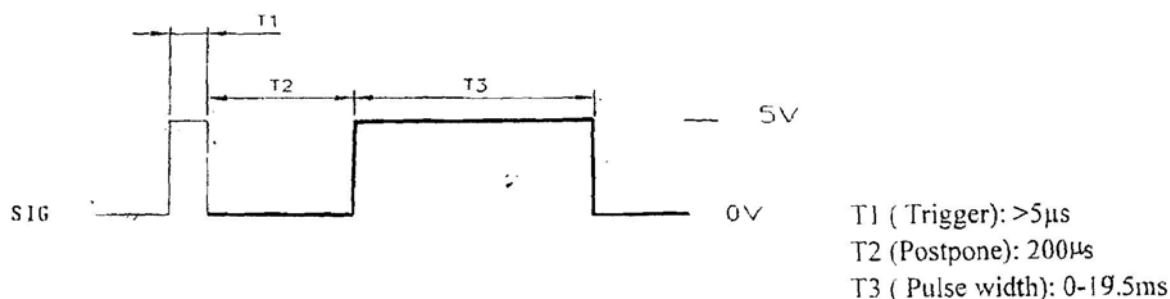
CORSO DI ORDINAMENTO

Indirizzo: ELETTRONICA E TELECOMUNICAZIONI
Tema di: SISTEMI ELETTRONICI AUTOMATICI
(Testo valevole per i corsi di ordinamento e per i corsi del progetto sperimentale "Sirio")

Si vuole realizzare un dispositivo elettronico che sia in grado di misurare la distanza da un oggetto, utilizzando una coppia di capsule ultrasoniche funzionanti da trasmettitore Tx e da ricevitore Rx opportunamente interfacciate (vedi figura Fig. 1).


Fig. 1

Il trasduttore ultrasonico contiene un circuito elettronico di controllo che permette di attivare il dispositivo (il segnale SIG del trasduttore ultrasonico è di tipo bidirezionale) e che produce in uscita un impulso (T_3) di durata proporzionale alla distanza (vedi figura Fig. 2).


Fig. 2

La distanza D [m] può essere calcolata, conoscendo la durata di T_3 e la velocità V del suono nell'aria attraverso la seguente formula:

$$D = \frac{VT_3}{2}$$



Ministero dell'Istruzione, dell'Università e della Ricerca

M272 – ESAME DI STATO DI ISTITUTO TECNICO INDUSTRIALE

CORSO DI ORDINAMENTO

Indirizzo: ELETTRONICA E TELECOMUNICAZIONI

Tema di: SISTEMI ELETTRONICI AUTOMATICI

(Testo valevole per i corsi di ordinamento e per i corsi del progetto sperimentale “Sirio”)

La velocità del suono dipende dalla temperatura secondo la seguente equazione:

$$V = a + b \text{ Temp [m/s]}$$

Dove le costanti a e b valgono rispettivamente: $a = 331.5$ [m/s] e $b = 0.62$ [m/s °C]

T_{emp} rappresenta la temperatura in gradi centigradi [°C].

Per misurare la temperatura T_{emp} è utilizzato un sensore con uscita digitale a 9 bit in complemento a due di tipo proporzionale (vedi tabella Tab. 1):

Temperature	Digital Output	
	Binary	Hex
+ 125°C	0 1111 1010	0FAh
+ 25°C	0 0011 0010	032h
+ 0.5°C	0 0000 0001	001h
0°C	0 0000 0000	000h
- 0.5°C	1 1111 1111	1FFh
- 25°C	1 1100 1110	1CEh
- 55°C	1 10010010	192h

Tab. 1

Il candidato, formulate le eventuali ipotesi aggiuntive:

- descriva attraverso quali periferiche del microcontrollore o microprocessore intende misurare la durata di T_3 e il valore di temperatura T_{emp} ;
- disegni il diagramma di flusso per la gestione dei sensori, evidenziando le modalità di interfacciamento se in “polling”, in “interrupt” o in modalità “mista”;
- descriva almeno una metodologia software e/o hardware per rilevare il periodo T_3 ;
- indichi una metodologia per visualizzare le grandezze rilevate dai sensori;
- progetti il codice di gestione di almeno uno dei due sensori utilizzati;
- valuti la distanza massima rilevabile dal sensore ultrasonico supponendo la temperatura T costante e pari a +20°C.

Inoltre il candidato ipotizzando una variazione di temperatura da +20°C a +40°C, valuti la variazione relativa della distanza rilevata.

Durata massima della prova: 6 ore.

È consentito l'uso di manuali tecnici e di calcolatrici non programmabili.

Non è consentito lasciare l'Istituto prima che siano trascorse 3 ore dalla dettatura del tema.

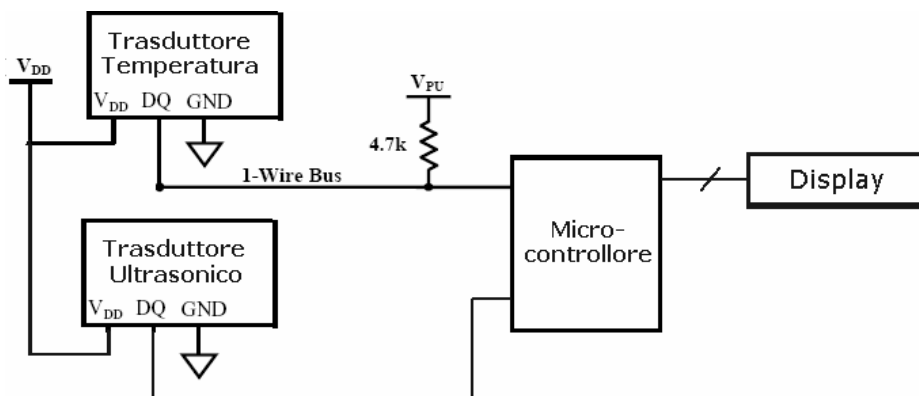
Svolgimento

(a)

Il sensore di temperatura ha un'uscita digitale a 9 bit, emettendo un codice binario proporzionale al valore Temp richiesto. Possiamo pensare si tratti di un dispositivo 1-Wire. Un sistema di comunicazione 1-Wire (progettato e sviluppato da Dallas Semiconductors) si basa su un dispositivo, detto master (il microcontrollore), che gestisce la comunicazione, scambiando informazioni e fornendo alimentazione a più dispositivi, detti slave (il sensore di temperatura), mediante un singolo cavo twisted-pair: un singolo filo (oltre quello di massa) permette la comunicazione tra più dispositivi e contemporaneamente fornisce l'alimentazione necessaria al loro funzionamento.

Il master ha una uscita open-drain o tri-state, con un resistore di pull-up a V_{PU} , gli slave hanno un'uscita open-drain con cui possono portare a 0 il segnale sul bus.

I dati sulla rete 1-Wire vengono trasferiti in "time slots", cioè porzioni di tempo di lunghezza prestabilita, tra 60 e 120 μ s, in cui è possibile scambiare informazioni sul bus. Il sistema non richiede clock, poiché ogni dispositivo slave 1-Wire contiene un oscillatore interno che si sincronizza durante i fronti di discesa del clock del master.



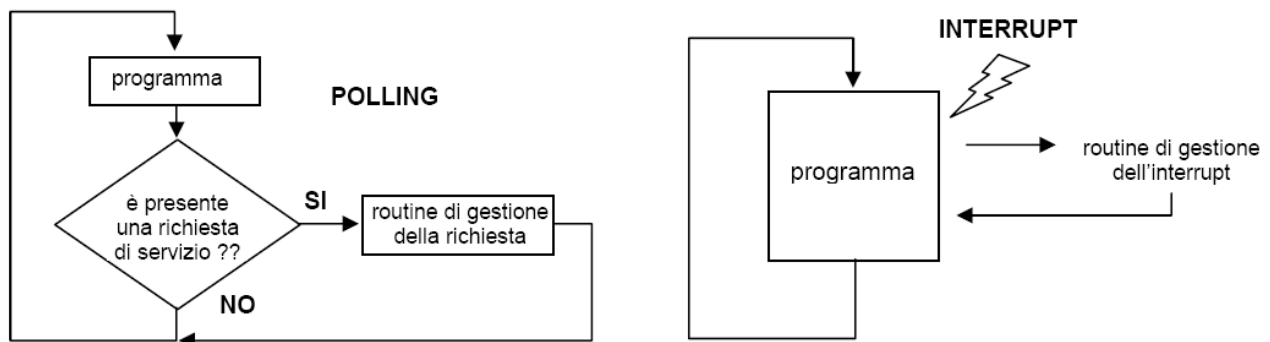
Il dato proveniente dal traduttore di temperatura viene acquisito mediante opportuna routine di acquisizione sul pin RB1 e memorizzato in questo modo:

- MSB nel file register tempsegno,
- I rimanenti 8 bit in tempvalore

La forma d'onda proveniente dal trasduttore ultrasonico trattandosi di segnale digitale 0-5V viene mandata ad un pin RB0 in ingresso al PIC.

(b)

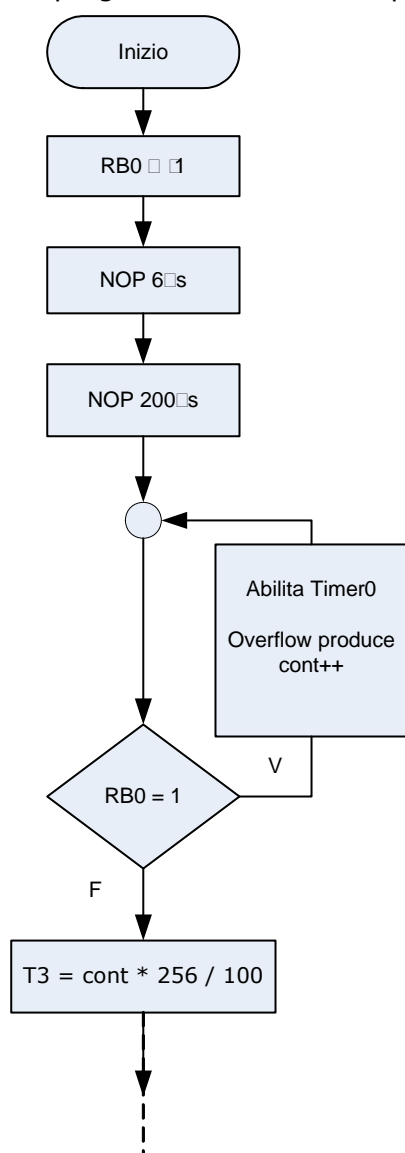
Un microcontrollore che ha in carico l'esecuzione di un programma, svolge in sequenza le istruzioni programmate secondo un ordine predeterminato e inoltre deve anche rispondere a richieste che gli giungono in modo casuale dai dispositivi con i quali interagisce.



A fronte di una richiesta di servizio casuale, il microprocessore deve abbandonare il regime corrente e avviare una procedura di gestione della richiesta di servizio. Due sono le modalità

con le quali il microcontrollore si dispone ad accogliere la richiesta di servizio proveniente da un dispositivo esterno:

- con il Polling, che significa interrogazione, il controllo delle richieste segue uno schema prefissato, ed è lo stesso programma in esecuzione sul microcontrollore a prevedere un controllo periodico delle periferiche per intercettare quelle con richieste di servizio pendenti. Il polling è quindi un metodo software: il controllo è tutto a carico del programma in esecuzione che periodicamente deve interrogare le periferiche per vedere se è presente una richiesta di servizio.
- Con l'interrupt è il dispositivo a lanciare la richiesta di servizio, la quale viene ricevuta dall'hardware del microcontrollore (mentre il software in esecuzione non ha il compito di sondare le richieste di servizio, come nel polling). Non appena un dispositivo lancia sulla linea una richiesta di interrupt (mediante un opportuno segnale: settando un pin al livello logico alto, oppure effettuando un cambio di stato su un pin), il microcontrollore sospende il programma corrente e passa ad eseguire la routine di gestione dell'interrupt memorizzata all'interno del programma stesso. Dopodichè il microcontrollore torna a eseguire il programma corrente dal punto dove era stato interrotto.



Dato che per attivare il trasduttore ultrasonico e quindi per produrre in uscita un impulso di durata $T_2 + T_3$ è necessario aver inviato prima un impulso $> 5\mu s$ si preferisce un interfacciamento in polling (a meno di non dover far altro con il microcontrollore durante il tempo T_2 , usando in tal caso l'interrupt per non tenerlo impegnato durante quei $200\mu s$).

Un diagramma di flusso per rilevare il tempo T_3 è dato ponendo RB0 in uscita, emettendo un segnale alto verso il SIG del trasduttore della durata di $6\mu s$, ottenuto con 6 No Operation (NOP). Ponendo quindi RB0 in ingresso per ricevere un segnale dal trasduttore, dopo avere aspettato un tempo pari a $200\mu s$ è noto che il livello del segnale si porti alto, così, mediante un ciclo e finché il segnale rimane alto, si valuta il tempo T_3 usando il timer del microcontrollore e una variabile di conteggio.

(c) (e)

La codifica del programma avviene in PICC per PIC16. Per misurare il tempo T_3 si collega il SIG del trasduttore ultrasonico al piedino RB0 del microcontrollore (si considera un PIC16) il quale viene dapprima settato in uscita ($RB0 = 0$) in modo da inviare sul pin SIG verso il trasduttore un segnale di 5V della durata $T_1 > 5\mu s$ usando una funzione da noi realizzata DelayUs(n) e definita in una libreria.h importata nel programma. Utilizziamo quindi DelayUs(6) per realizzare un ritardo di $6\mu s$.

Successivamente, posto in ingresso RB0 ($RB0 = 1$) si attende $200\mu s$ usando DelayUs(200), quindi si controlla che dall'esterno il trasduttore ultrasonico di prossimità abbia portato RB0 a livello logico alto. A questo punto si avvia un conteggio con timer, finché il segnale su RB0 non si sia portato sul livello basso. La durata dell'impulso T_3 va da 0 a $19,5ms$ in base alla distanza rilevata. Per il conteggio viene impiegato il Timer0 del microcontrollore, che fa capo al registro TMR0.

Il Timer è un semplice contatore ad 8 bit e ad ogni quarto di periodo del clock esso incrementa il suo valore (da 0 a 255). Raggiunto il valore di 255, al clock successivo, il registro effettua un overflow e riprende il suo conteggio da 0. L'evento di overflow influisce sul bit TOIF del registro INTCON, il quale subisce una transizione da 0 a 1.

Il clock del PIC16 si considera impostato a $f_{osc} = 4MHz$, così che la frequenza del ritardo base del registro TMR0 vale: $f_{osc}/4 = 1MHz$ (ossia il ritardo base vale $1\mu s$). Si farà incrementare una variabile di conteggio cont ad ogni overflow:

$$t_{\text{ritardo overflow}} = 1\mu\text{s} \cdot 256 = 256\mu\text{s}$$

Il tempo di ritardo può essere eventualmente amplificato mediante il prescaler che è un divisore di frequenza: riceve in ingresso la frequenza $f_{\text{osc}}/4$ e la restituisce divisa per 2, 4, 8, ecc. fino a 256. Il fattore di divisione può essere programmato tramite la combinazione di tre bit PS2, PS1, PS0 del registro OPTION_REG. Settiamo il prescaler come di default a 1:2.

```
OPTION=0b10000000;
// Impostazione Registro Opzioni
// bit 0 -> PS0 Prescaler Rate Select bit 0
// bit 1 -> PS1 Prescaler Rate Select bit 1
// bit 2 -> PS2 Prescaler Rate Select bit 2
// bit 3 -> PSA Prescaler assegnato al Timer0 (1=al Watchdog Timer)
// bit 4 -> T0SE Incremento Timer0 su transizione: 0=low->high 1=high->low
// bit 5 -> T0CS Clock Timer0 da clock interno (1=su transizione pin T0CKI)
// bit 6 -> INTEDG Interrupt (INT) edge (1=salita 0=discesa)
// bit 7 -> RBPU Resistenze pull-up su porta B (1=off 0=on)
```

Il valore di partenza di conteggio del Timer0 si imposta a zero:

```
TMR0=0
```

Noto che T3 vale al massimo 19,5ms, si avrà in luogo di tale valore un conteggio massimo pari a:

$$\text{cont} = 19,5\text{ms}/256\mu\text{s} = 76$$

e in ogni caso un conteggio variabile tra 0 e 76 per ampiezze di impulso T3 variabili tra 0 a 19,5ms.

```
int cont; // Contatore per il Timer
long T3, Temp, tempsegno, tempvalore, V, D;
// T3: Tempo di misura T3 espresso in decine di ms
// Temp: Temperatura
// tempsegno: MSB del codice a 9 bit fornito dal sensore di temperatura (bit di segno)
// tempvalore: Rimanti 8 bit della misura di temperatura
// V: Velocità del suono
// D: Distanza espressa in cm

void main (void) {
    TRISB=0b00000000; // RB0 in uscita
    RB0 = 1; // RB0 a 5V
    DelayUs(6); // Attende 6µs, produce impulso T1
    RB0 = 0; // RB0 a 0V, fine dell'impulso di attivazione del trasduttore
    TRISB = 0b00000001; // RB0 in ingresso
    DelayUs(200); // Attende 200µs
    while (RB0 == 1) {
        TMR0 = 0; // Timer0 = 0
        if (TOIF) { // Overflow del timer0
            cont++; // Incremento il contatore
            TOIF = 0; // Resetta il flag del timer0
        }
    }
    T3 = cont * 256 / 100; // Calcolo il tempo T3 espresso in decine di ms: T3 * 10^-4
```

Continuando il programma, per ricavare il valore della distanza, programmiamo la parte inerente la temperatura. Il bit di segno della temperatura è il MSB fornito in seriale dal trasduttore che abbiamo memorizzato nel LSB del file register tempsegno.

Se tempsegno = 0 allora consideriamo i rimanenti 8 bit, convertendoli in decimale e ottenendo un numero pari al valore della temperatura moltiplicato per due (es: 125 °C -> 250) quindi dividiamo per 2. Se tempsegno = 1 allora per recuperare il valore di temperatura in decimale dobbiamo effettuare prima l'operazione inversa del complemento a 2, ossia: 256 - tempvalore e poi dividere per 2. L'equazione che useremo per conoscere la velocità del suono in quest'ultimo caso sarà sottrattiva. In ultimo ricaviamo la distanza D espressa in centimetri. Successivamente mediante una opportuna funzione sarà possibile rappresentare il valore ottenuto su display.

```

if (tempsegno == 1) {
    Temp = (256 - tempvalore) / 2;
    V = (33150 - 62 * Temp) / 100 ;
}
else
{
    Temp = tempvalore / 2;
    V = (33150 + 62 * Temp) / 100 ;
}

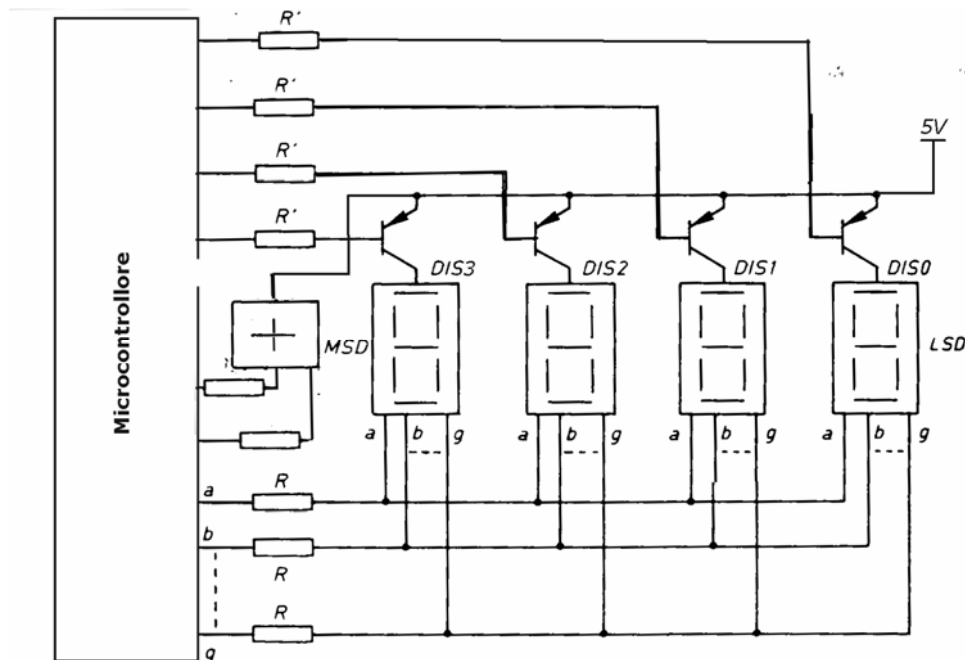
D = V * T3 / 2 / 100; // risultato in centimetri

```

(d)

Per visualizzare le grandezze rilevate dai sensori è possibile utilizzare un display LCD oppure realizzarlo mediante 4 display a 7 segmenti con dot pitch (necessario per separare le cifre decimali) e un display di segno.

Per visualizzare la distanza fino all'ordine del centimetro, basterebbero 3 display 7 segmenti, mentre per quanto riguarda la temperatura, necessita di un display di segno ± e per quanto riguarda le cifre, di dover rappresentare le centinaia ma anche i decimi di grado centigrado.



I 4 display 7 segmenti con dot pitch, del tipo anodo comune, sono collegati in serie (con i rispettivi catodi collegati tra loro) e vengono pilotati usando gli stessi pin del PIC, il quale mediante un altro pin agisce di volta in volta abilitando l'anodo del display destinato a ricevere il numero da visualizzare. Così facendo si sfrutta la persistenza dell'immagine sulla retina dell'occhio, per creare l'illusione che i

numeri riportati sui display siano fissi, mentre in realtà vengono aggiornati rapidamente in sequenza. Usiamo 8 piedini di PORTC vanno collegati ai catodi dei display: RC0 con dp, RC1 con c, RC2 con d, RC3 con e, RC4 con b, RC5 con a, RC6 con f, RC7 con g.

Per illuminare di volta in volta un display il PIC dovrà mandare il rispettivo anodo a 5V. La corrente inviata dal microcontrollore però non è sufficiente ad accendere nitidamente i LED dei 3 display, pertanto è conveniente pilotarli mediante tre transistor PNP in configurazione ON/OFF, che separano il microcontrollore dai display; si è scelto di utilizzare i BC307, di tipo general purpose. I piedini del PIC che pilotano gli anodi dei display sono RB4, RB5, RB6 e RB7 e sono collegati in base ai rispettivi transistor, mediante una resistenza da 2,2 k Ω . Gli emettitori dei PNP vanno all'alimentazione di 5V, i collettori agli anodi dei display a LED.

Il display di segno è alimentato da 5V sull'anodo e pilotato da RB3 e RB2.

(f)

Nel caso $T = 20^{\circ}\text{C}$:

$$V = a + (b \cdot \text{Temp}) \rightarrow V_{20} = 331,5 + (0,62 \cdot 20) = 343,9 \text{ m/s}$$

$$D = V \cdot T_3 / 2 \rightarrow D_{20} = 343,9 \cdot 19,5 \cdot 10^{-3} / 2 = 3,35 \text{ m}$$

Nel caso $T = 40^{\circ}\text{C}$

$$V_{40} = 331,5 + (0,62 \cdot 40) = 356,3 \text{ m/s}$$

$$D_{40} = 343,9 \cdot 19,5 \cdot 10^{-3} / 2 = 3,47 \text{ m}$$

Nel passaggio della temperatura da 20°C a 40°C la misura della distanza subisce una variazione:

$$\Delta D = D_{40} - D_{20} = 12 \text{ cm}$$