

Temi d'esame

Esame di Stato di istruzione secondaria superiore

Indirizzo: ITAT – Elettronica ed elettrotecnica

Articolazione Elettronica

Tema di: Sistemi Automatici 2015

Esame di Stato di istruzione secondaria superiore

Indirizzo: ITAT – Elettronica ed elettrotecnica

Articolazione Elettronica

Tema di: Sistemi Automatici 2012

Esame di Stato di istruzione secondaria superiore

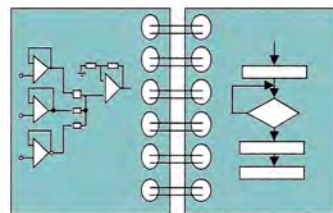
Indirizzo: ITAT – Elettronica ed elettrotecnica

Articolazione Elettronica

Tema di: Sistemi Automatici 2009

Risolvere il tema d'esame di Sistemi Automatici – Elettronica

In questo modulo vengono forniti i fondamenti per potersi cimentare nella risoluzione del tema d'esame di Sistemi Automatici. Nella prima pagina si danno indicazioni generali sulla struttura della prova. A seguire una tabella riporta uno schema di ripasso dei principali argomenti afferenti il Tema d'Esame, con la specificazione delle unità di riferimento. Ognuno di questi argomenti viene poi riassunto e ripreso, avvalendosi di esempi tratti dai temi del passato. Infine vengono risolti alcuni tra i più recenti temi d'esame.



Generalmente l'elaborato di Sistemi Automatici del nuovo ordinamento è composto da due parti.

La prima parte è un progetto **pluridisciplinare** che richiede conoscenze di programmazione dei sistemi a **microcontrollore** proprie della materia di Sistemi Automatici, capacità di schematizzare circuiti elettronici mutuata dalla materia Elettronica ed Elettrotecnica, nonché conoscenze tecnologiche costruttive derivate dalla materia TPSEE.

Nella seconda parte vengono proposti dei quesiti e viene richiesto al candidato di risolverne solo un sottoinsieme. Alcuni quesiti sono una espansione del progetto della prima parte, altri si riferiscono al filone dei **sistemi lineari**, concernente la stabilizzazione, il calcolo delle funzioni di trasferimento e delle risposte.

Il tema della prima parte ha aspetti hardware e software. Per l'hardware è richiesta la schematizzazione a blocchi del sistema, con la descrizione delle funzioni di ciascun blocco e l'interrelazione tra i blocchi, in funzione dell'operatività complessiva.

Viene richiesto inoltre di progettare alcuni blocchi utilizzando classiche configurazioni elettroniche e di dimensionare i componenti in funzione delle specifiche.

Dal punto di vista software è richiesto il diagramma di flusso per la gestione del sistema e la codifica di parte dello stesso con un linguaggio di propria conoscenza.

TABELLA ARGOMENTI PER IL TEMA D'ESAME

Argomento	Unità	Volume
Input e Condizionamento	A2, A4, B1, E1+12	3
Conversione Analogico-Digitale	A5, G1, G2, G3	3
Attuazione	A2, H2	3
Visualizzazione	F1, F2, F3	3
Temporizzazione	D5, D6, E5, E6, F5, F6	2
Elaborazione	F2, F3, G1, G3	3
Memorizzazione	D2, D3	2
Sistemi lineari	C1, C2, C3, D1, D2, D3, D4	3
Sistemi lineari	I1, I2, I3, L1, L2, L3, L4, M1, M2, M3, M4, N1, N2, N3, N4	2

Input e condizionamento

L'input può essere di tipo **digitale** oppure **analogico**.

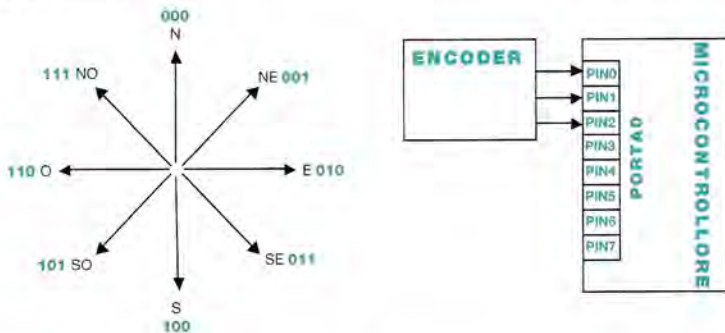
L'ingresso di tipo digitale, proveniente tipicamente da trasduttori con uscite digitali, come l'encoder per la misura di rotazioni angolari, può essere direttamente acquisito da una porta e gestito a livello software.

Input Digitale

Un centro di previsioni atmosferiche deve rilevare la direzione del vento.

Il dispositivo per la rilevazione della direzione è un codificatore angolare digitale con codifica a tre bit, secondo l'associazione descritta dalla seguente rosa dei venti. Lo schema a fianco visualizza il collegamento tra le uscite dell'encoder e i tre pin PORTA0.PIN0, PORTA0.PIN1, PORTA0.PIN2 utilizzati.

ESEMPIO 1



Il seguente segmento di programma si occupa di trasformare a livello software il valore digitale N acquisito nel corrispondente valore angolare.

```

if (PORTA.PIN0==0)      if (PORTA.PIN2==0)
B0=0                    B2=0
else                    else
B0=1;                  B2=1;
if (PORTA.PIN1==0)      N=B0*2^0 + B1*2^1 + B2*2^2;
B1=0                    ANGOLO=N*45;
else
B1=1;

```

L'ingresso di tipo analogico necessita di una combinazione di azioni fornite da una catena di dispositivi:

Input Analogico

- **trasduttore** per la conversione del dato fisico in grandezza elettrica;
- **condizionatore** per l'adattamento dei segnali uscenti dal trasduttore alla gamma di ingresso dell'ADC;
- **convertitore analogico-digitale** ADC.

Trasduttore

La grandezza elettrica all'uscita del trasduttore può essere o direttamente una tensione, oppure una corrente o una resistenza. In questi ultimi casi è necessario convertire corrente e resistenza in tensione, per mezzo di **convertitori I→V** e **R→V** a operazionale.

Il circuito in figura è un convertitore **R→V**.

ESEMPIO 2

La $R(T)$ è un sensore RTD (Resistive Temperature Detector) con la seguente dipendenza dalla temperatura:

$$R(T) = R(0) \cdot (1 + \alpha T)$$

Nell'ipotesi che la temperatura vari da $T_{MIN} = 50^\circ\text{C}$ a $T_{MAX} = 200^\circ\text{C}$ la resistenza subirà le seguenti variazioni, nell'ipotesi $R(0) = 100\ \Omega$, $\alpha = 3,675 \cdot 10^{-3}$.

$$\begin{aligned}
T_{MIN} = 50^\circ &\rightarrow R(T_{MIN}) = 100 \cdot (1 + 3,675 \cdot 10^{-3} \cdot 50) = 118\ \Omega \\
T_{MAX} = 200^\circ &\rightarrow R(T_{MAX}) = 100 \cdot (1 + 3,675 \cdot 10^{-3} \cdot 200) = 173\ \Omega
\end{aligned}$$

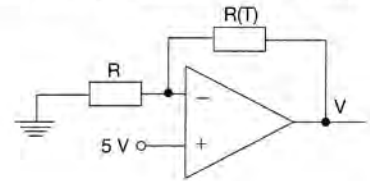
L'operazionale, in configurazione non invertente, applica una corrispondenza lineare tra valore della resistenza e uscita:

$$V = \left(1 + \frac{R(T)}{R}\right) \cdot 5$$

Ponendo $R = R(T_{MAX}) = 173 \, \Omega$ risulta:

per $T = T_{MIN} \rightarrow V_{MIN} = (1 + 118/173) \cdot 5 = 8,4 \, \text{V}$

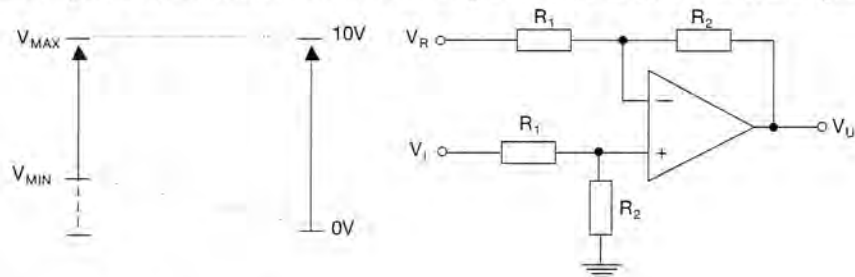
per $T = T_{MAX} \rightarrow V_{MAX} = (1 + 173/173) \cdot 5 = 10 \, \text{V}$



Condizionatore

Il circuito di condizionamento deve adattare il valore minimo V_{MIN} e massimo V_{MAX} della tensione uscente dal traduttore ai rispettivi valori accettati dall'ADC, che normalmente sono $0 \div 5 \, \text{V}$ oppure $0 \div 10 \, \text{V}$.

Il compito è comunemente affidato all'amplificatore differenziale nella figura.



La formula del circuito è:

$$V_u = \frac{R_2}{R_1} \cdot (V_I - V_R)$$

Imponendo le condizioni:

$$V = V_{MIN} \rightarrow V_u = 0 \, \text{V}$$

$$V = V_{MAX} \rightarrow V_u = 10 \, \text{V}$$

si ricava il sistema a fianco.

$$\begin{cases} \frac{R_2}{R_1} \cdot (V_{MIN} - V_R) = 0 \\ \frac{R_2}{R_1} \cdot (V_{MAX} - V_R) = 10 \end{cases}$$

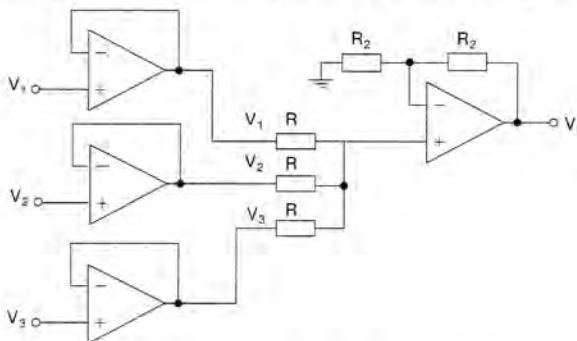
Dalla prima equazione si ricava $V_R = V_{MIN}$ e dalla seconda si ricava il rapporto R_2/R_1 .

In taluni casi una prima elaborazione numerica è richiesta già a livello hardware, prima ancora che a livello software. Il caso tipico è la determinazione della media di più fonti analogiche con l'ausilio dell'operazionale in veste di sommatore.

Elaborazione
Hardware

ESEMPIO 3

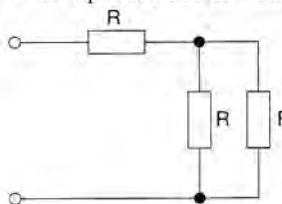
Si debba ricavare un segnale variabile tra $0 \, \text{V}$ e $5 \, \text{V}$ mediando tre sorgenti variabili tra 0 e $2,5 \, \text{V}$. Si utilizza allo scopo un sommatore non invertente, preceduto da tre operazionali con guadagno unitario, per il disaccoppiamento con i sensori, come descritto dalla seguente figura.



$$\text{Risulta dall'analisi circuitale: } V_u = \left(1 + \frac{R_2}{R_1}\right) \cdot \frac{V_1 + V_2 + V_3}{3} = 2 \cdot \frac{V_1 + V_2 + V_3}{3}$$

Infatti per la sovrapposizione degli effetti la tensione presente sul + dell'operazionale dovuta a un singolo ingresso vale:

$$V_+ = V_i \cdot \frac{R/2}{R + R/2} = \frac{R/2}{3R/2} = \frac{1}{3}$$



Per collaudare la formula immaginiamo che tutti gli ingressi valgano 2,5 V.

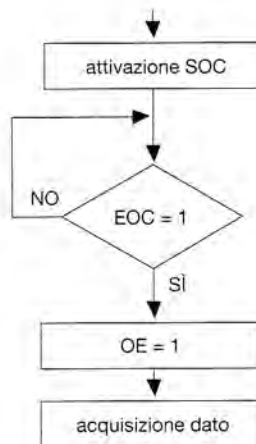
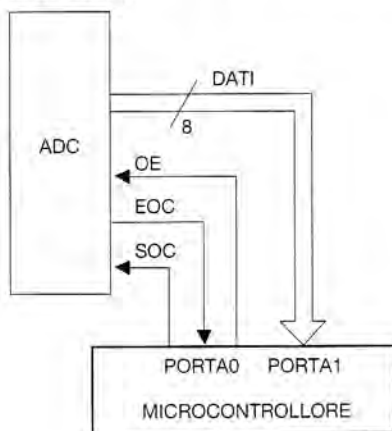
Si verifica allora che $V_U = 5$ V infatti: $2 \cdot (V_1 + V_2 + V_3) / 3 = 2 \cdot (2,5 + 2,5 + 2,5) / 3 = 5$

Conversione analogico-digitale

Per la conversione analogico-digitale si può utilizzare un modulo interno al microcontrollore, nel qual caso è sufficiente conoscere le modalità software per la sua gestione, oppure un **ADC** esterno. In questo caso la comunicazione tra microcontrollore è garantita dai seguenti segnali di temporizzazione e sincronizzazione:

- **SOC** (Start Of Conversion) per l'avvio della conversione;
- **EOC** (End Of Conversion) per la notifica di fine conversione;
- **OE** (Output Enable) per l'abilitazione delle uscite dell'ADC.

Le due figure seguenti mostrano lo schema di interfacciamento tra centrale di elaborazione e componente ADC tramite due porte di input-output, e il corrispondente diagramma di flusso.



ADC Esterno

Il diagramma di flusso può essere implementato in un qualsiasi linguaggio di programmazione, purché si conoscano le modalità per referenziare gli ingressi e le uscite del sistema.

In questo esempio i tre pin della porta 0 coinvolti nel protocollo di gestione dell'ADC si referenziano come PORTA0.PIN0, PORTA0.PIN1, PORTA0.PIN2.

```

PORTA0.PIN0=1;           //Impulso di SOC
PORTA0.PIN0=0;
do
{
}
while(!PORTA0.PIN1);     //Attesa EOC=1
PORTA0.PIN2=1;           //OE=1
DATO=PORTA1;             //Lettura DATO
  
```

ESEMPIO 4

Se l'ADC ha otto uscite il dato convertito è pronto per essere acquisito attraverso una porta a 8 bit.

Nel caso di ADC a 12 oppure 16 uscite è necessario utilizzare due porte, la prima per l'acquisizione del byte meno significativo, la seconda per l'acquisizione del nibble (caso 12 bit) o del byte (caso 16 bit) più significativo. In questi casi deve essere attribuito il giusto peso al termine più significativo.

ESEMPIO 5

Il dato a 12 bit spazia da 0 a $2^{12} - 1 = 4095$. La figura mostra uno stralcio di tutti i valori possibili.

	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1

Per convertire il dato digitale a 12 bit si deve tenere conto che:

- il primo byte $b_7 \div b_0$ ha una gamma $0 \div 255$;
- il secondo byte (nibble basso) $b_{11} \div b_8$ ha una gamma $256 \div 4095$.

Pertanto il secondo byte pesa 256 volte il primo.

Nell'ipotesi che il primo e secondo byte vengano acquisiti rispettivamente da due porte di nome PORTA0 e PORTA1 la formula per il calcolo del valore digitale letto è:

$$DATO = PORTA0 + PORTA1 * 256$$

Attuazione

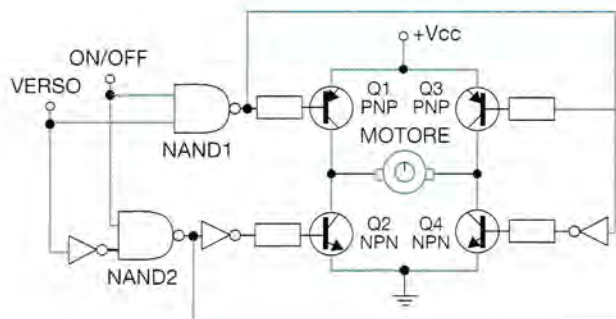
Le uscite delle porte di input-output sono tutte digitali, pertanto sono utili per comandare dispositivi in modalità **ON/OFF**. Tuttavia la potenza disponibile spesso non è adeguata, dovendo comandare apparati che assorbono correnti elevate e pertanto dissipano molta potenza.

Per ovviare a questo si può impiegare il transistor in funzione di amplificatore o fare ricorso al relè.

ESEMPIO 6

Il circuito è un comando a transistori a ponte per l'avviamento di motori a corrente continua con possibilità di inversione del verso di rotazione.

L'ingresso ON/OFF attiva il motore, mentre l'ingresso VERSO ne comanda il verso di rotazione.



Se $ON/OFF = 0$ le uscite di NAND1 e NAND2 valgono entrambe 1 a prescindere dal livello dell'ingresso VERSO.

Di conseguenza tanto la coppia di transistori Q_1Q_4 quanto la coppia Q_2Q_3 sono interdette.

Se invece $ON/OFF = 1$ il motore è alimentato e il suo verso dipende dal livello dell'ingresso VERSO:

- $VERSO = 0 \rightarrow NAND1 = 1, NAND2 = 0 \rightarrow Q_3Q_2$ in conduzione
- $VERSO = 1 \rightarrow NAND1 = 0, NAND2 = 1 \rightarrow Q_1Q_4$ in conduzione

La direzione di rotazione cambia a seconda della coppia complementare di transistor in conduzione:

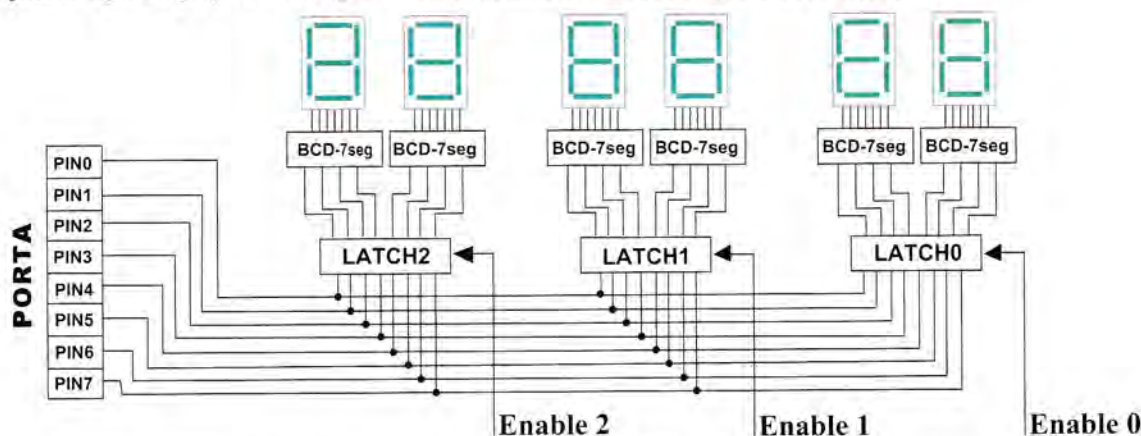
- se conducono Q_1Q_4 la corrente scorre in un senso nel motore che ruota in una data direzione;
- se conducono Q_3Q_2 la corrente scorre in un senso opposto nel motore che inverte il senso di rotazione.

Visualizzazione

Nel caso di visualizzazione mediante display a sette segmenti spesso i digit implicati sono tanti e il numero di porte a disposizione è insufficiente per comandarli tutti.

In questi casi si fa ricorso alla **multiplexazione**, inviando a una sola porta in sequenza i dati per ciascun display, facendo seguire la memorizzazione del dato sul latch attestato al display.

La figura seguente riguarda la visualizzazione su tre indicatori di tre numeri a due cifre. Ogni indicatore è formato da due display. Per ogni coppia di display basta un byte, poiché ogni display richiede quattro bit di codifica della cifra numerica rappresentata.



Il byte uscente dalla porta deve essere predisposto a livello software in funzione della configurazione hardware della sezione di visualizzazione.

Si supponga di dover visualizzare su due display una temperatura T compresa tra 0°C e 99°C .

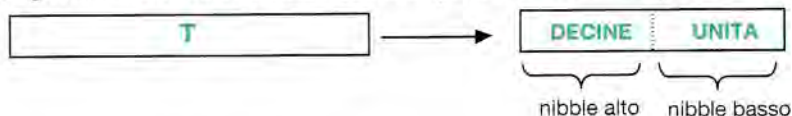
Innanzitutto serve scomporre T nelle sue cifre delle unità UNITA e delle decine DECINE.

L'operazione viene compiuta in due fasi:

- si calcola il **resto** della divisione $T/10$ per il calcolo dell'unità
- si calcola il **valore intero** della divisione $T/10$ per il calcolo della decina

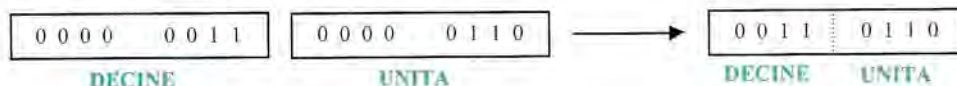
Successivamente per disporre il tutto all'interno di un singolo byte si deve:

- portare DECINE nel nibble alto, traslando di quattro bit a sinistra oppure moltiplicando per 16
- aggiungere nel nibble basso UNITA, con una operazione di somma oppure un OR



Supponendo ad esempio che $T = 36$ i passi sono:

1. UNITA = resto($T/10$) = 6 = $0110_{(2)}$
2. DECINE = intero($T/10$) = 3 = $0011_{(2)}$
3. DECINE = DECINE * 16 = 48 = $00110110_{(2)}$
4. DECINE + UNITA = 48 + 6 = 54 = $00110100_{(2)}$



ESEMPIO 7

Temporizzazione

Nei sistemi automatici la **temporizzazione** svolge un ruolo fondamentale, disciplinando la scansione delle operazioni automatiche.

I sistemi a microcontrollore sono solitamente dotati di uno o più **timer** che oltre a supportare a livello hardware la temporizzazione, si interfacciano con il software per mezzo di opportuni registri, garantendo al programmatore la massima flessibilità progettuale.

Nei casi più semplici tuttavia la temporizzazione è fornita da un clock esterno, che è costituito da una sequenza ininterrotta di **impulsi** di periodo T noto.

La figura visualizza il treno di impulsi di periodo T , che va collegato a un ingresso del microcontrollore.

In questo caso il compito primario del software è quello di **intercettare** l'arrivo di un impulso.

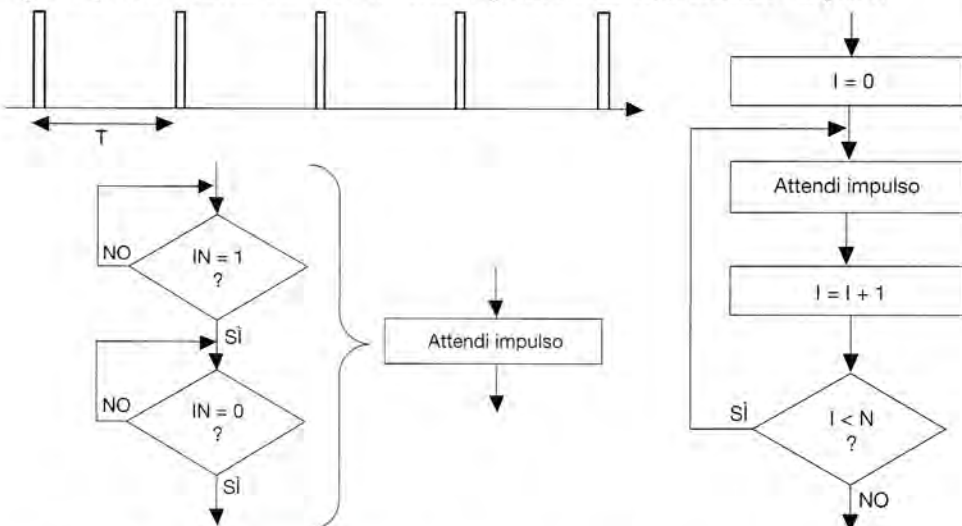
Come descritto dal diagramma di flusso "Attendi impulso" la rilevazione dell'impulso avviene mediante due test in sequenza:

- nel primo test il sistema rimane in attesa di un livello 1;
- nel secondo test il sistema attende il ritorno al livello 0.

Per calcolare il **tempo di ritardo** si deve solo moltiplicare il numero di impulsi N per il periodo T .

Il diagramma di flusso a destra consente di dosare il ritardo in base a N .

Utilizza una variabile di ciclo I che viene inizializzata a zero e viene incrementata a ogni impulso. L'uscita dal ciclo si verifica quando sono stati contati N impulsi.

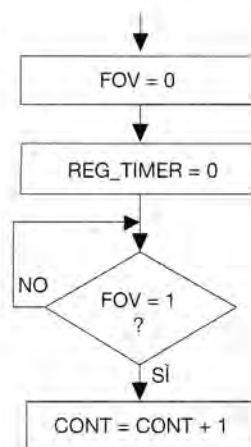


Nei timer inclusi all'interno del microcontrollore è presente un **registro di conteggio** REG_TIMER e un flag che ne segnala l'**overflow**. Il tempo di overflow è noto perché gli impulsi di conteggio provengono dal clock interno.

Ogni unità di attesa elementare viene implementata mediante un ciclo while, che attende appunto la notifica di overflow con il passaggio a 1 di FOV.

L'attesa può essere amplificata nidificando l'attesa elementare all'interno di un nuovo ciclo nel quale contare il numero CONT di attese elementari.

```
REG_TIMER=0;           //resetta registro timer
FOV=0                  //resetta flag di overflow
do {} while(FOV=0);    //attendi overflow
CONT = CONT + 1;
```



Temporizzazione
mediante timer
esterno

Temporizzazione
mediante timer
interno

Elaborazione

Quando si hanno a disposizione tutti i dati sotto forma di variabili, diventa molto agevole l'applicazione di formule, funzioni e algoritmi, grazie alla immediatezza e flessibilità del software.

Una eccezione si verifica se il trasduttore impiegato non è lineare, in questo caso infatti non essendoci una corrispondenza lineare tra il valore nativo e il corrispondente valore digitale, non è possibile determinare matematicamente il dato originale. Si ricorre in questi casi a una look-up table, ovvero una tabella di associazione, nella quale sono registrate tutte le corrispondenze necessarie.

Una **look-up table** è una tabella di correlazione tra dati provenienti da una sorgente e dati prodotti in uscita. Si usano tabelle simili nei campi della visione, qualora si voglia operare una correzione dei colori per conferire a una immagine stili diversi. In questi casi si associa a ogni colore originale una diversa gradazione. Un altro caso lampante è la serie delle tabelle impresse nella memoria dei computer per le funzioni trigonometriche: anziché affidare il calcolo a un algoritmo, che risulta troppo lento, si ricava il valore indicizzando una tabella dove vengono preventivamente registrati tutti i risultati.

Si supponga che un DATO possa variare tra 1 e 15.

Potrebbe ad esempio essere una temperatura variabile tra 1 °C e 15 °C.

Si supponga che il trasduttore che rileva il dato sia di tipo logaritmico, ovvero faccia corrispondere alla temperatura T una tensione V data da: $V = \ln(T)$

ESEMPIO 8

Nel foglio di lavoro si possono leggere i risultati della formula nella colonna LN.

Il dato di tensione una volta digitalizzato e acquisito dovrà essere riconvertito a livello software nel dato originario di temperatura.

Questa operazione può essere compiuta in due modi:

- mediante un algoritmo matematico
- mediante una tabella di look-up.

L'algoritmo dovrebbe computare il calcolo dell'antilogaritmo, per elidere l'effetto del logaritmo.

L'antilogaritmo naturale equivale all'esponenziale:

$$e^{\ln(\text{DATO})} = \text{DATO}$$

Nella colonna ANTILN si può osservare che l'operazione restituisce i valori originali DATO.

	A	B	C	D
1		DATO	LN	ANTILN
2		1	0,00	1
3		2	0,69	2
4		3	1,10	3
5		4	1,39	4
6		5	1,61	5
7		6	1,79	6
8		7	1,95	7
9		8	2,08	8
10		9	2,20	9
11		10	2,30	10
12		11	2,40	11
13		12	2,48	12
14		13	2,56	13
15		14	2,64	14
16		15	2,71	15

Se è richiesta una operatività immediata in tempo reale non è tuttavia possibile utilizzare l'algoritmo.

Conviene allora considerare direttamente i valori calcolati registrati nella tabella e leggere direttamente sulla stessa il valore convertito.

Memorizzazione

Tutti i microcontrollori hanno a disposizione una memoria interna. Tuttavia può capitare la richiesta di memorizzare dei set di dati entro una **memoria esterna**.

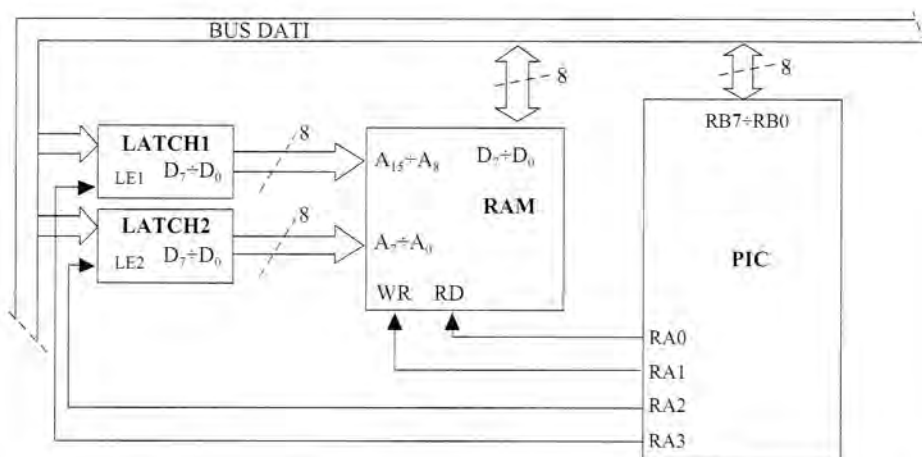
In questi casi si deve prevedere un collegamento di interfaccia tra la memoria e il dispositivo di elaborazione e un algoritmo per gestire le operazioni di scrittura e lettura dei dati.

ESEMPIO 9

Si ipotizzi di avere a disposizione una memoria RAM di 64 KByte, con 16 linee di indirizzo $A_{15} \div A_0$ e 8 linee dati $D_7 \div D_0$. La si interfacci con un PIC attraverso due porte di input-output da 8 bit: una PORTA e una PORTB. Si stabilisce di utilizzare PORTB per i dati e PORTA per i comandi.

Per gli indirizzi sono necessari ulteriori 16 bit. Poiché non ci sono altre porte di input-output a disposizione, si stabilisce di utilizzare il bus dati anche per il transito degli indirizzi. Questi vengono memorizzati su due latch a 8 bit in due fasi di transito distinte.

Nella prima vengono inviati i bit $A_{15} \div A_8$ e attivato il primo latch, successivamente vengono inviati $A_7 \div A_0$ e viene attivato il secondo latch. I pin della porta A comandano la lettura e la scrittura sulla RAM e i due latch.



Si riporta di seguito un programma in pseudocodice per la scrittura dell'intera memoria.

Si indicano con il nome PUNTATORE_LOW e PUNTATORE_HIGH le due variabili che contengono gli indirizzi $A_7 \div A_0$ e $A_{15} \div A_8$.

```

ripeti 65535 volte
{
  PUNTATORE_LO → PORTB
  impulso su RA2 (LE2)
  PUNTATORE_HI → PORTB
  impulso su RA3 (LE1)
  poni DATO su PORTB
  impulso su RA1 per scrittura (WR)
  incrementa puntatore
}

```

Sistemi lineari

In questa sezione è richiesta la capacità di risolvere schemi a blocchi.

Viene richiesto anche il calcolo delle funzioni di trasferimento e la determinazione della risposta della f.d.t a un dato segnale di ingresso.

Fondamentale è la capacità di progettare un sistema di controllo nei riguardi della stabilità, dimensionando il guadagno statico per assolvere le specifiche riguardanti il **margine di fase** φ_M .

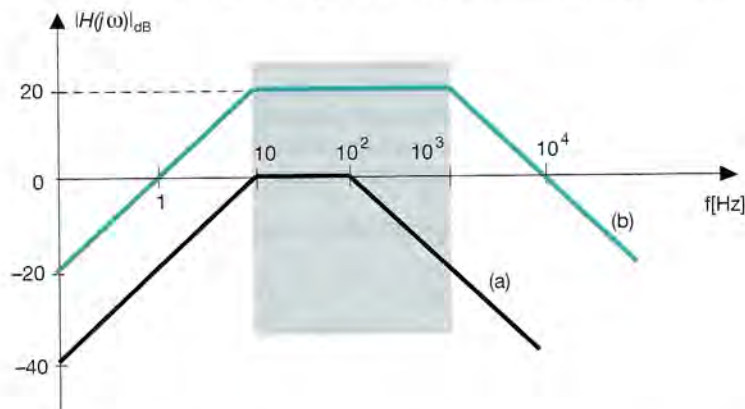
È richiesto a questo proposito di calcolare di quanto attenuare o amplificare il guadagno statico per assolvere la specifica su φ_M .

Per rispondere si devono tracciare i diagrammi di Bode del modulo e della fase e valutare come si modificano ω_C , φ_C , φ_M traslando il diagramma verso l'alto o verso il basso.

Il diagramma in figura rappresenta la risposta in frequenza di un amplificatore.

Determinare la funzione di trasferimento dell'equalizzatore di ampiezza, utilizzando i diagrammi asintotici forniti della risposta originale in nero e della risposta equalizzata in blu.

ESEMPIO 10



La f.d.t $H(j\omega)$ scelta per l'equalizzatore è costituita da una costante moltiplicativa, uno zero e un polo.

$$H(j\omega) = K \cdot \frac{1 + sT}{1 + s\tau}$$

1. per implementare la curva b l'equalizzatore deve cancellare il polo della f.d.t originale ($f = 10^2$ Hz) e inserire un nuovo polo a frequenza maggiore ($f = 10^3$ Hz);
2. per traslare verso l'alto il guadagno nella banda passante, l'equalizzatore deve introdurre un guadagno di 20 dB.

Per soddisfare il secondo requisito si deve porre $K = 10$ affinché $K_{dB} = 20 \log 10 = 20$ dB

Per soddisfare il primo requisito deve risultare:

$$z = 10^2 \cdot 2\pi \quad p = 10^3 \cdot 2\pi$$

ESAME DI STATO DI ISTRUZIONE SECONDARIA SUPERIORE
Indirizzo: ITAT - ELETTRONICA ED Elettrotecnica
ARTICOLAZIONE ELETTRONICA
Tema di: SISTEMI AUTOMATICI 2015

PRIMA PARTE

Una ditta farmaceutica, per testare gli effetti di un farmaco in studio, commissiona la progettazione di uno strumento che effettui diversi controlli di qualità del prodotto in relazione a due diversi reagenti. Le analisi si effettuano su un campione di 10 ml del prodotto inserendo nel contenitore, in due fasi successive, due reagenti A e B mediante appositi micro dosatori e rilevando la risposta alla reazione mediante due sensori caratterizzati dai parametri funzionali di seguito descritti.

Sensore SPH di acidità: fornisce una uscita digitale a 4 bit corrispondente ai valori di PH compresi tra 1 e 14 secondo la tabella riportata:

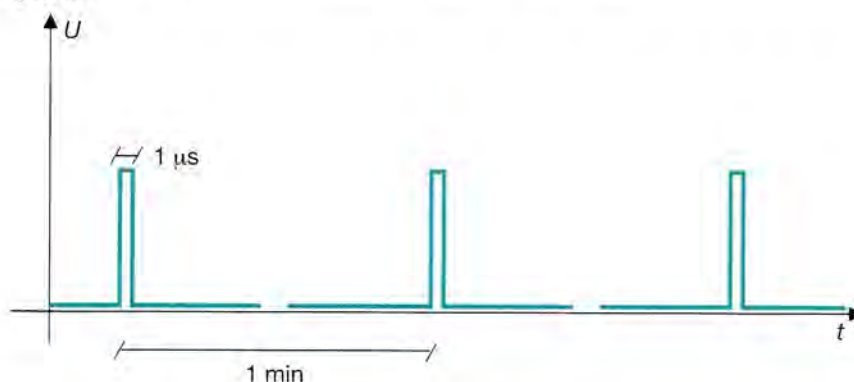
Valore in uscita (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Misura corrispondente	error	1	2	3	4	5	6	7	8	9	10	11	12	13	14	error

Biosensore SC: in grado di rilevare concentrazioni comprese 100 µg/ml e 2000 µg/ml fornendo in uscita una corrente dipendente dalla misura di concentrazione secondo la relazione:

$$I(C) = -\frac{1}{10 * \ln(C)}$$

Il sistema si completa con:

- un pulsante di START del tipo relè ad autoritenuta con ritorno alla posizione di riposo dopo 10 secondi;
- due micro erogatori per il rilascio dei reagenti A e B in dosaggi già stabiliti che possono essere assimilati a dispositivi ON/OFF;
- un timer per la scansione dei tempi che fornisce un impulso TTL di durata 1 µs ad intervalli di 1 minuto come riportato nel grafico.



Il processo da controllare è il seguente: l'avvio è determinato dall'operatore attraverso il pulsante START che attiva il timer. Al primo impulso utile dopo l'attivazione il sistema deve consentire il rilascio del reagente A mediante l'abilitazione del corrispondente micro erogatore. Dopo 1 minuto deve essere acquisito il valore presente sul sensore SPH e contestualmente deve essere consentito il rilascio del reagente B mediante abilitazione del corrispondente micro erogatore. Trascorso un altro minuto si effettua la seconda acquisizione del valore di PH mediante il medesimo sensore SPH ed entrambi i micro erogatori vengono posti nella posizione OFF.

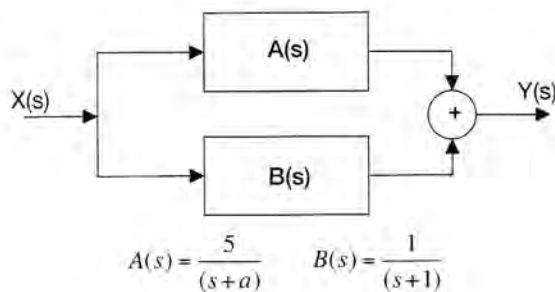
Dopo una attesa di altri 2 minuti deve essere acquisito il valore della concentrazione del principio attivo mediante il biosensore SC e il processo ha termine.

Il candidato, fatte le ipotesi aggiuntive che ritiene più opportune, deve:

1. rappresentare lo schema a blocchi che realizza il sistema di controllo, utilizzando un dispositivo programmabile di sua conoscenza, descrivendo i singoli blocchi dal punto di vista funzionale;
2. descrivere la logica di controllo che si intende utilizzare con riferimento alle specifiche tecniche dei dispositivi scelti;
3. progettare un algoritmo di gestione del processo che consenta l'attuazione delle singole fasi e la memorizzazione dei dati rilevati in apposita area di memoria, nell'ipotesi che si vogliano effettuare prove su 100 campioni di prodotto.

SECONDA PARTE

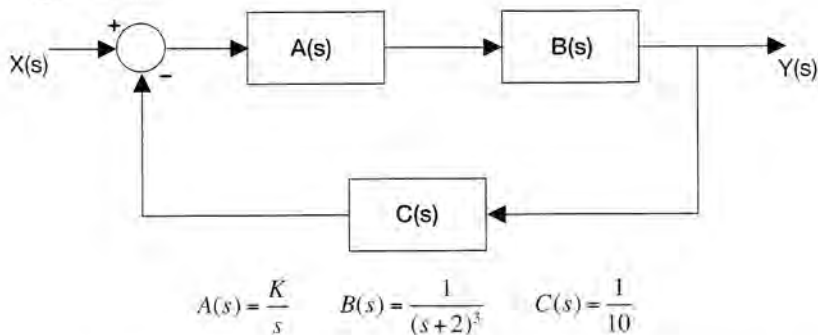
1. In relazione allo schema sviluppato nella prima parte si progetti l'interfaccia HW tra il sensore SC e il sistema programmabile scegliendo un dispositivo di conversione A/D che garantisca un errore massimo di misura rilevata pari a $1 \mu\text{g/ml}$; inoltre si modifichi il software di gestione in modo da calcolare, al termine del processo, il valore medio di concentrazione ottenuto nelle prove e il numero di prove in cui tale concentrazione è inferiore a $500 \mu\text{g/ml}$.
2. In relazione allo schema sviluppato nella prima parte si proponga una soluzione atta a rilevare il malfunzionamento del sensore SPH. In particolare, nel caso in cui tale dispositivo fornisca in uscita i valori 0HEX oppure FHEX il sistema deve provvedere all'accensione di un led di segnalazione, i dati memorizzati fino a quel momento devono essere annullati e il processo deve essere sospeso per riprendere solo dopo il reset di sistema.
3. Dalle prove effettuate su un sistema continuo, opportunamente sollecitato, si è ottenuto il modello rappresentato in figura.



Dopo aver ricavato la funzione di trasferimento complessiva si determini il valore di $a > 0$ per il quale la risposta a regime $y(t)$ del sistema sottoposto a sollecitazione $u(t)$ gradino unitario abbia valore 6.

Il candidato inoltre, sulla base delle proprie competenze in materia di controlli analogici, proponga un esempio applicativo di sistema reale cui possa corrispondere il modello dato.

4. Per il sistema di figura siano:



Si determini la funzione di trasferimento complessiva del sistema e si calcoli il margine di fase e di guadagno del sistema ad anello aperto nel caso in cui $K = 50$.

Per quale campo di variazione di K il sistema in questione risulta stabile?

SOLUZIONI PRIMA PARTE

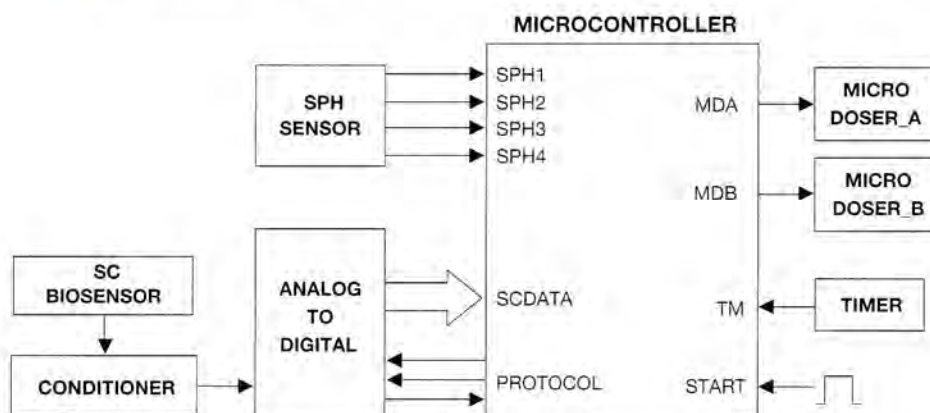
1. Il candidato rappresenti lo schema a blocchi che realizza il sistema di controllo, utilizzando un dispositivo programmabile di sua conoscenza, descrivendo i singoli blocchi dal punto di vista funzionale

Il sistema è controllato da un microcontrollore attraverso le porte di ingresso-uscita.

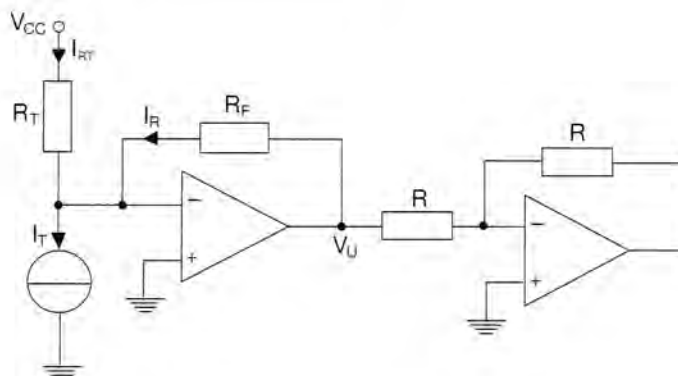
Una porta è dedicata interamente all'acquisizione del dato digitale SCDATA proveniente dal convertitore ADC che a sua volta riceve il segnale condizionato proveniente dal sensore SC.

Tre bit di una seconda porta inoltre sono dedicati ai segnali di protocollo SOC, EOC, OE per la gestione dell'ADC. Altri quattro bit sono dedicati all'acquisizione delle uscite del sensore SPH.

Un bit riceve gli impulsi del timer e un ultimo bit riceve il segnale di START proveniente dal relè ad autoritenuta. Sono presenti solo due bit d'uscita per il comando dei due dosatori.



Lo schema del blocco di condizionamento visualizzato a fianco è composto da una prima sezione che converte in tensione la corrente fornita dal sensore SC seguito da un blocco invertitore di tensione. Si procede di seguito al dimensionamento dei componenti.



Si calcolano le correnti erogate agli estremi di concentrazione $C = 100 \mu\text{g/ml}$ e $C = 2000 \mu\text{g/ml}$.

$$I_T(100) = -\frac{1}{10 \cdot \ln(100)} = -0,0217 \quad I_T(2000) = -\frac{1}{10 \cdot \ln(2000)} = -0,0132$$

Si calcola la differenza tra le due correnti:

$$\Delta I = 0,0217 - 0,0132 = 0,0085$$

Scegliendo un ADC con dinamica di ingresso $0 \div 5 \text{ V}$ alla ΔI deve corrispondere $\Delta V = 5 \text{ V}$.

La resistenza R_F di amplificazione vale allora:

$$R = \frac{\Delta V_U}{\Delta I} = \frac{5}{0,0085} = 588 \Omega$$

I valori della tensione d'uscita corrispondenti ai limiti di concentrazione diventano allora:

$$V_U(100) = -0,0217 \times 588 = -12,76 \text{ V}$$

$$V_U(2000) = -0,0132 \times 588 = -7,76 \text{ V}$$

Come previsto la variazione della tensione d'uscita vale $\Delta V_U = 12,76 - 7,76 = 5 \text{ V}$.

Ora è necessario adattare i livelli oltre all'intervallo. Per garantire una tensione $V_U = 0$ in corrispondenza di $I_T = 0,0132$ si deve annullare I_R forzando una $I_{RT} = I_T$.

A questo scopo la R_T deve valere:

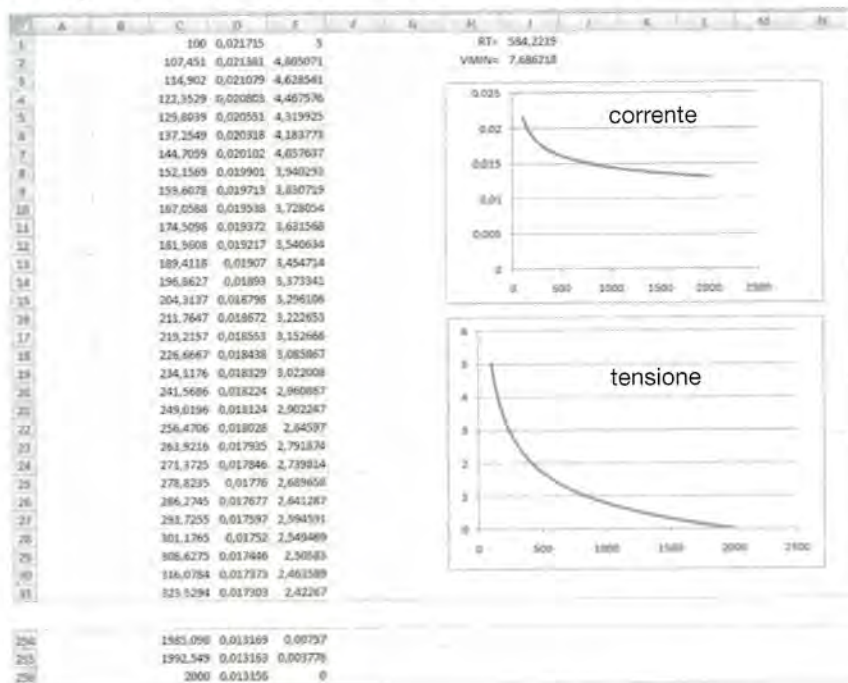
$$R_T = \frac{V_{CC}}{I(2000)} = \frac{15}{0,0132} = 1,14 \text{ k}\Omega$$

Il secondo operazionale deve solamente invertire il segno della tensione, per cui è sufficiente disporre due resistenze R di uguale valore.

Il sensore di concentrazione C non è lineare bensì logaritmico, come si evince dalla sua formula e come si può osservare nei grafici tracciati entro un foglio di lavoro Excel con le seguenti colonne:

- Colonna C concentrazioni da 100 a 2000 $C2 = C1 + ((2000-100)/255)$
- Colonna D corrispondenti valori di corrente $D1 = 1/(10 \cdot \ln(C1))$
- Colonna E corrispondenti valori di tensione $E1 = D1 \cdot \$I\$1 - \$I\2

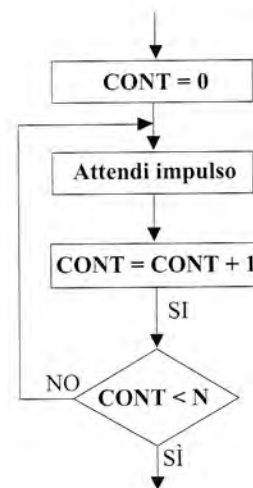
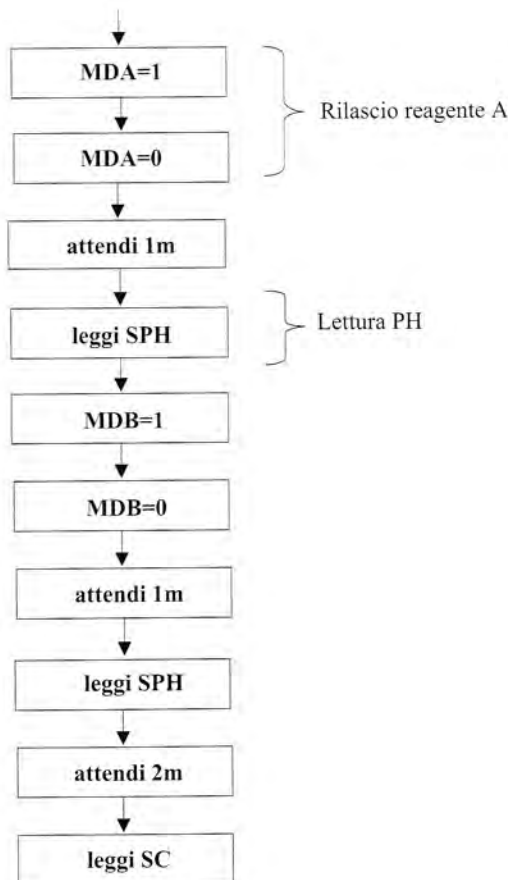
Pertanto, noti i valori della tensione all'uscita del condizionatore e i corrispondenti valori digitali, per risalire ai veri valori di concentrazione si dovrà utilizzare il foglio di lavoro come tabella di lookup, intercettare la tensione nella colonna E e leggere in corrispondenza la concentrazione nella colonna C.



2. Il candidato descriva la logica di controllo che si intende utilizzare con riferimento alle specifiche tecniche dei dispositivi scelti

Il flusso è puramente sequenziale e prevede l'attivazione degli erogatori, le attese e la lettura delle concentrazioni.

Per i cicli di attesa è necessario invece impiegare dei test. Il criterio è quello di attendere ripetutamente l'arrivo degli impulsi del timer. Conoscendo il periodo T del timer il sistema dosa il ritardo DURATA, contando con la variabile CONT il numero di impulsi in base alla relazione: $DURATA = CONT * T$



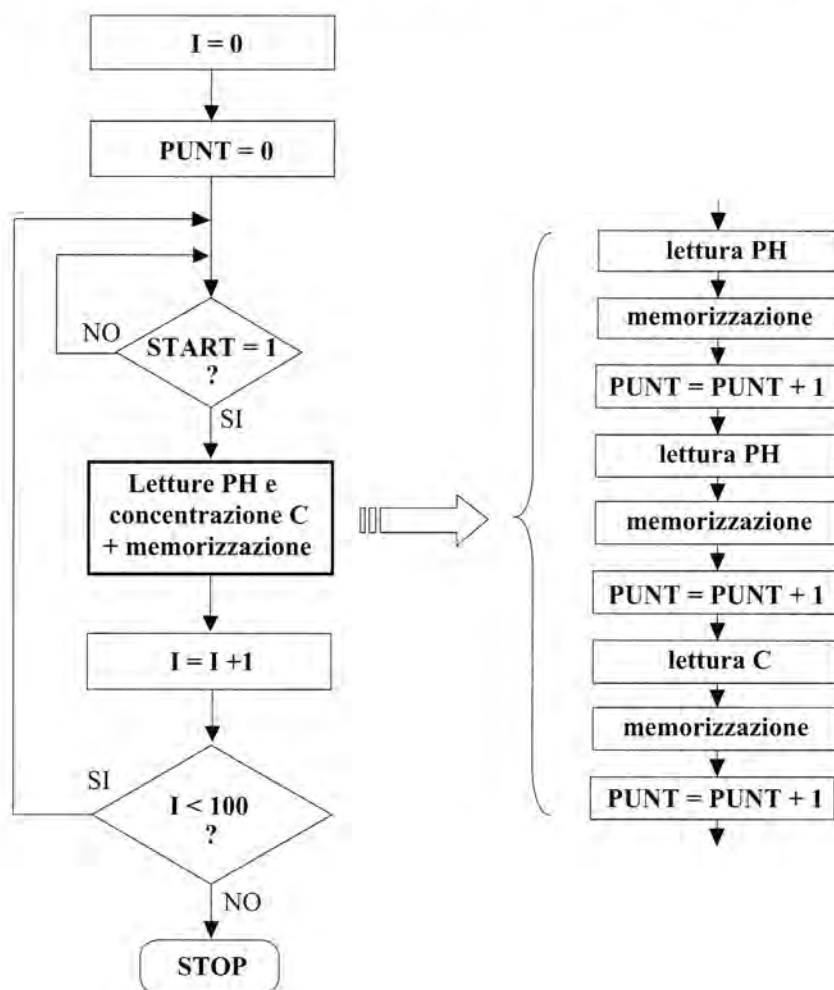
3. Il candidato progetti un algoritmo di gestione del processo che consenta l'attuazione delle singole fasi e la memorizzazione dei dati rilevati in apposita area di memoria, nell'ipotesi che si vogliano effettuare prove su 100 campioni di prodotto

L'algoritmo impiega una variabile puntatore PUNT per conservare l'indirizzo della locazione di memoria dove memorizzare il dato attuale.

A ogni ciclo PUNT viene incrementato tre volte per memorizzare i due PH e la concentrazione C.

La variabile I conserva il numero di ripetizioni di ogni singolo processo; il ciclo finisce quando $I = 100$.

Prima di ogni ciclo il sistema rimane in attesa del segnale START, per consentire la predisposizione del nuovo campione di prodotto. Con questo algoritmo le letture vengono memorizzate sequenzialmente. Per memorizzare in tre regioni di memoria distinte i valori del PH e di C basta modificare il programma utilizzando tre puntatori.



SOLUZIONI SECONDA PARTE

1. In relazione allo schema sviluppato nella prima parte si progetti l'interfaccia HW tra il sensore SC e il sistema programmabile scegliendo un dispositivo di conversione A/D che garantisca un errore massimo di misura rilevata pari a $1 \mu\text{g/ml}$; inoltre si modifichi il software di gestione in modo da calcolare, al termine del processo, il valore medio di concentrazione ottenuto nelle prove e il numero di prove in cui tale concentrazione è inferiore a $500 \mu\text{g/ml}$

La relazione che lega la concentrazione C alla tensione V non è lineare bensì logaritmica.

Per garantire la migliore risoluzione si deve fare riferimento alla minima variazione della tensione, che si verifica nel passaggio da $C = 2000$ a $C = 1999$ dove la variazione di tensione vale:

$$\Delta V = 0,003776 - 0 = 0,003776$$

Il numero di livelli deve essere quindi: $5/0,003776 = 1324$ e il numero di bit 11 perché sia $2^{11} > 1324$.

I 2048 livelli assolvono nel contempo alla specifica sull'errore di misura di 1 microg/ml, perché da 100 a 2000 i passi sono solo 1900. Per quanto riguarda il software di gestione si sceglie di formulare la codifica in linguaggio C. Viene utilizzato un vettore di 100 caselle nelle quali si presuppone sia stato memorizzato l'insieme delle concentrazioni.

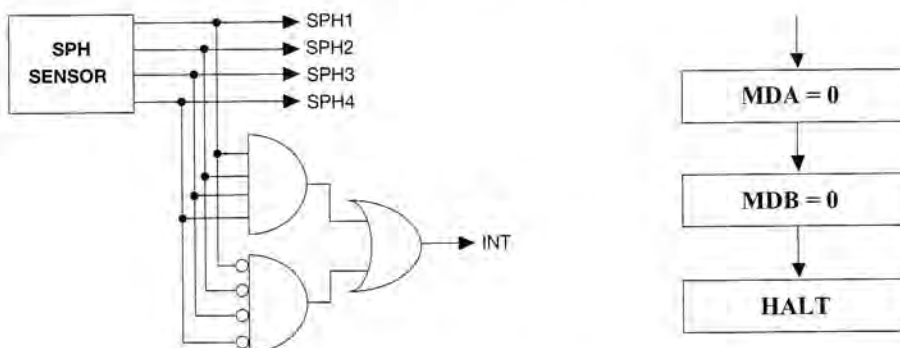
```
float CONCENTRAZIONI[100], TOTALE = 0, MEDIA; int i, NUMPROVE = 0;
for(i = 0; i < 100; i = i + 1)
    TOTALE = TOTALE + CONCENTRAZIONI[i];
MEDIA = TOTALE / 100;
for(i = 0; i < 100; i = i + 1)
    if(CONCENTRAZIONI[i] < 500)
        NUMPROVE = NUMPROVE + 1;
```

2. In relazione allo schema sviluppato nella prima parte si proponga una soluzione atta a rilevare il malfunzionamento del sensore SPH. In particolare, nel caso in cui tale dispositivo fornisca in uscita i valori 0HEX oppure FHEX il sistema deve provvedere all'accensione di un led di segnalazione, i dati memorizzati fino a quel momento devono essere annullati e il processo deve essere sospeso per riprendere solo dopo il reset di sistema.

L'errore avviene se le combinazioni binarie sono 0000 = 0 Hex oppure 1111 = F Hex.

Due porte AND intercettano gli errori e una OR li propaga verso l'ingresso di interrupt INT del microcontrollore.

Il diagramma di flusso descrive la routine di gestione dell'interrupt: i due microdosatori devono essere inibiti per sospendere il processo e il sistema deve essere posto in modalità HALT.



3. Dopo aver ricavato la funzione di trasferimento complessiva si determini il valore di $a > 0$ per il quale la risposta a regime $y(t)$ del sistema sottoposto a sollecitazione $u(t)$ gradino unitario abbia valore 6.

Il candidato inoltre, sulla base delle proprie competenze in materia di controlli analogici, proponga un esempio applicativo di sistema reale cui possa corrispondere il modello dato.

I due blocchi sono in cascata quindi le due f.d.t si sommano.

$$G(s) = A(s) + B(s) = \frac{5}{s+a} + \frac{1}{s+1}$$

Si calcola la risposta allo scalino.

$$X(s) = \frac{1}{s} \quad Y(s) = X(s) \cdot \frac{5}{s+a} + X(s) \cdot \frac{1}{s+1} = \frac{5}{s(s+a)} + \frac{1}{s(s+1)}$$

Si antitraforma la prima risposta con il metodo dei residui.

$$\frac{5}{s(s+a)} = \frac{A}{s} + \frac{B}{s+a}$$

$$A = \lim_{s \rightarrow 0} \frac{5}{s(s+a)} \cdot s = \frac{5}{a}$$

$$B = \lim_{s \rightarrow -a} \frac{5}{s(s+a)} \cdot (s+a) = -\frac{5}{a}$$

$$\frac{5}{s(s+a)} = \frac{5}{a} \cdot \left[\frac{1}{s} - \frac{1}{s+a} \right]$$

$$y'(t) = L^{-1} \left[\frac{5}{s(s+a)} \right] = \frac{5}{a} \cdot L^{-1} \left[\frac{1}{s} \right] - \frac{5}{a} \cdot L^{-1} \left[\frac{1}{s+a} \right] = \frac{5}{a} (1 - e^{-at})$$

Si calcola la seconda risposta considerando che al posto di a è presente 1.

$$y''(t) = (1 - e^{-t})$$

Si sommano le risposte e si impone il valore di regime per calcolare a .

$$y(t) = y'(t) + y''(t) = \left(\frac{5}{a} \right) \cdot (1 - e^{-at}) + (1 - e^{-t})$$

$$\lim_{t \rightarrow \infty} y(t) = \left(\frac{5}{a} + 1 \right) = 6 \rightarrow a = 1$$

È possibile risolvere il problema direttamente, senza passare attraverso il calcolo della antitrasformata, applicando il teorema del valore finale. Si lascia al lettore l'individuazione dell'esempio applicativo.

4. Si determini la funzione di trasferimento complessiva del sistema e si calcoli il margine di fase e di guadagno del sistema ad anello aperto nel caso in cui $K = 50$.

Per quale campo di variazione di K il sistema in questione risulta stabile?

Applicando la formula della retroazione si ricava:

$$G_R(s) = \frac{Y(s)}{X(s)} = \frac{A(s) \cdot B(s)}{1 + A(s) \cdot B(s) \cdot C(s)} = \frac{\frac{K}{s} \cdot \frac{1}{(s+2)^3}}{1 + \frac{K}{s} \cdot \frac{1}{(s+2)^3} \cdot \frac{1}{10}}$$

Sviluppando risulta:

$$G_R(s) = \frac{\frac{K}{s} \cdot \frac{1}{(s+2)^3}}{1 + \frac{K}{s} \cdot \frac{1}{(s+2)^3} \cdot \frac{1}{10}} = \frac{\frac{K}{s} \cdot \frac{1}{(s+2)^3}}{\frac{10s(s+2)^3 + K}{10s(s+2)^3}} = \frac{10K}{10s(s+2)^3 + K}$$

La f.d.t. ad anello aperto vale:

$$G(s) = A(s) \cdot B(s) \cdot C(s) = \frac{50}{s} \cdot \frac{1}{(s+2)^3} \cdot \frac{1}{10} = \frac{5}{s(s+2)^3}$$

Si pone la f.d.t. nella forma costanti di tempo e si determina K_{dB} e i poli.

$$G(s) = \frac{5}{s(s+2)^3} = \frac{5}{8} \cdot \frac{1}{s(0.5s+1)^3}$$

$$K = 5/8 \quad 20 \log K = -4 \quad p_1 = 0, p_2 = 2$$

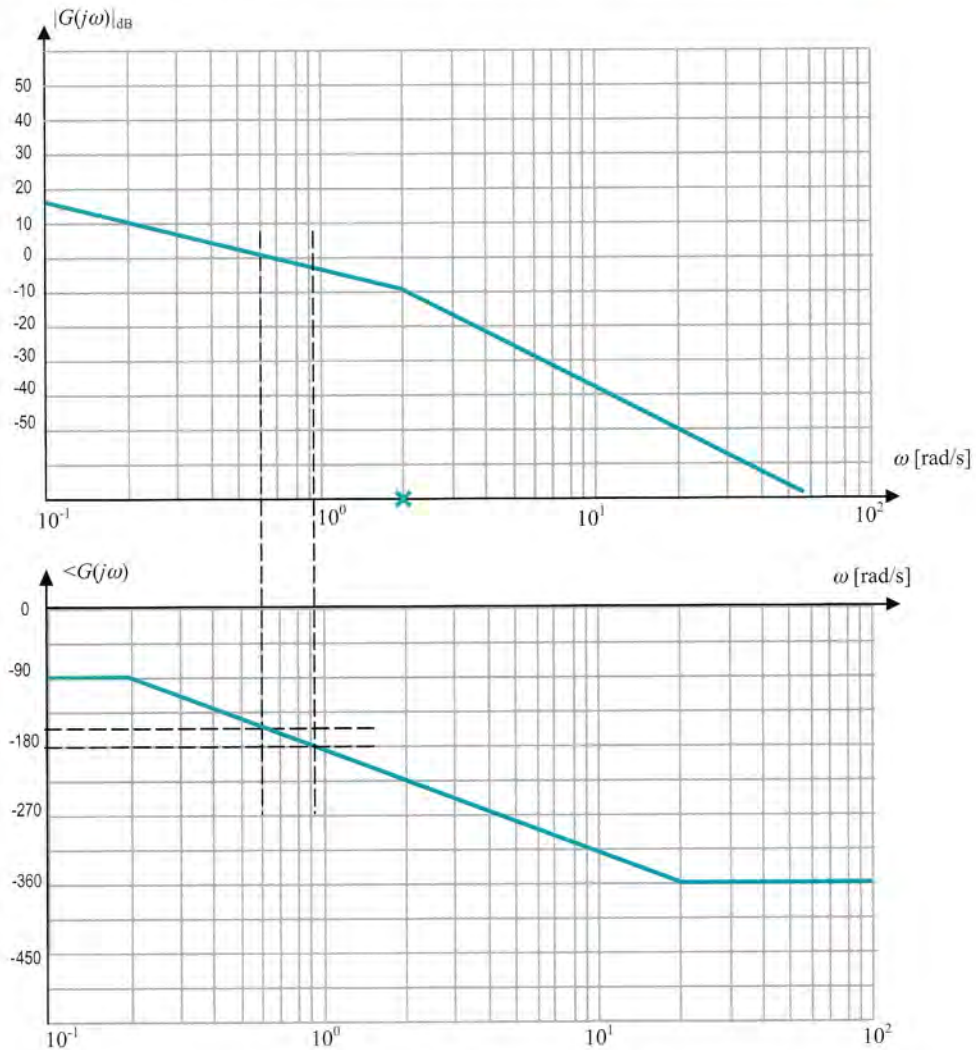
Si disegnano i diagrammi di Bode e si osserva che il grafico del modulo interseca l'asse delle ω a una pulsazione che vale circa 0,6 rad/sec.

In corrispondenza si legge un sfasamento critico $\varphi_C \approx 155^\circ$ e di conseguenza un margine di fase $\varphi_M \approx 25^\circ$.

Per ricavare il campo di variazione di K si deve tracciare una retta orizzontale a -180° fino a incrociare il diagramma delle fasi; da qui si traccia una retta verticale fino a incontrare la pulsazione.

Ora si trasla il diagramma dei moduli verso l'alto in modo che passi per 0 dB a questa pulsazione e si va a leggere il K nel punto $\omega = 10^0$. Per tutti i valori inferiori a questo K il sistema sarà stabile.

In caso contrario infatti il sistema avrà uno sfasamento di 180° con un guadagno superiore a 1 e quindi sarà instabile per il criterio di Bode.



ESAME DI STATO DI ISTRUZIONE SECONDARIA SUPERIORE
Indirizzo: ITAT - ELETTRONICA ED ELETTROTECNICA
ARTICOLAZIONE ELETTRONICA
Tema di: SISTEMI AUTOMATICI 2012

Si vuole realizzare un sistema 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 1).

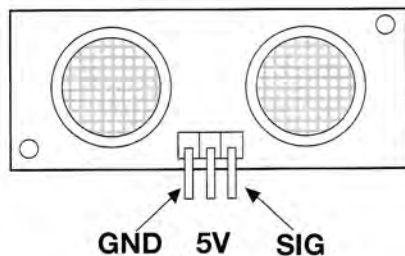
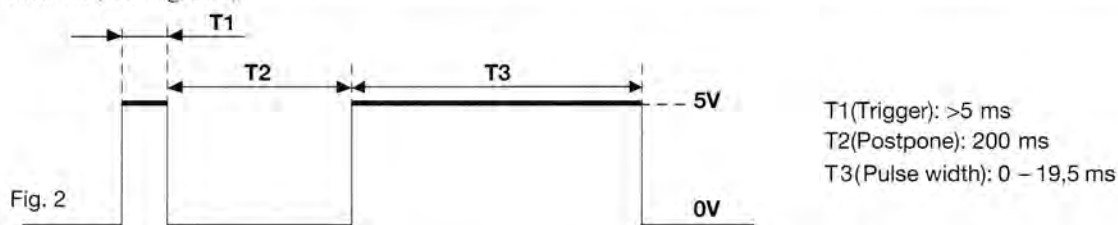


Fig. 1

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 (T3) di durata proporzionale alla distanza (vedi figura 2).



La distanza D [m] può essere calcolata, conoscendo la durata di T3 e la velocità V del suono nell'aria, attraverso la seguente formula:

$$D = \frac{V \cdot T_3}{2}$$

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

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

Dove le costanti a e b valgono rispettivamente: $a = 331,5$ [m/s] e $b = 0,62$ [m/s °C].

TEMP rappresenta la temperatura in gradi centigradi [°C].

Per misurare la temperatura Temp è utilizzato un sensore con uscita digitale a 9 bit in complemento a due di tipo proporzionale (vedi tabella 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 1001 0010	192h

Il candidato, formulate le eventuali ipotesi aggiuntive:

1. Descriva attraverso quali periferiche del microcontrollore o microprocessore intende misurare la durata di T_3 e il valore di temperatura Temp;
2. Disegni il diagramma di flusso per la gestione dei sensori, evidenziando le modalità di interfacciamento se in "polling", in "interrupt" o in modalità "mista";
3. Descriva almeno una metodologia software e/o hardware per rilevare il periodo T_3 ;
4. Indichi una metodologia per visualizzare le grandezze misurate dai sensori;
5. Progetti il codice di gestione di almeno uno dei sensori utilizzati;
6. 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.

SOLUZIONE

Colori utilizzati nella soluzione:

Arancio: Variabili di Programmazione

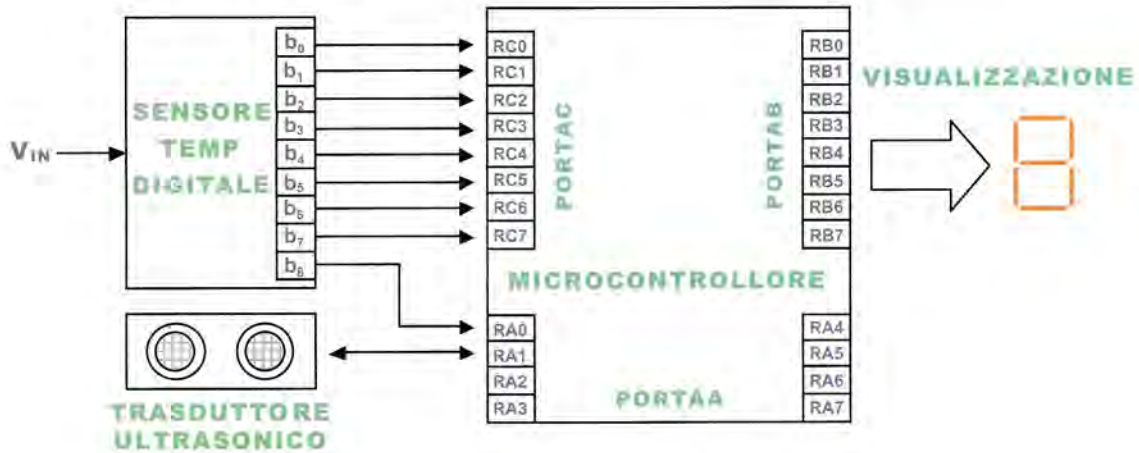
Blu: Segnali Circuitali

Verde: Dispositivi

1. Il candidato descriva attraverso quali periferiche del microcontrollore o microprocessore intende misurare la durata di T_3 e il valore di temperatura Temp.

Il sensore di temperatura digitale viene collegato alla **PORTAC** con i primi 8 bit; il nono bit di segno è collegato al pin **RA0** della **PORTAA**.

Il trasduttore ultrasonico è collegato bidirezionalmente al pin **RA1**.



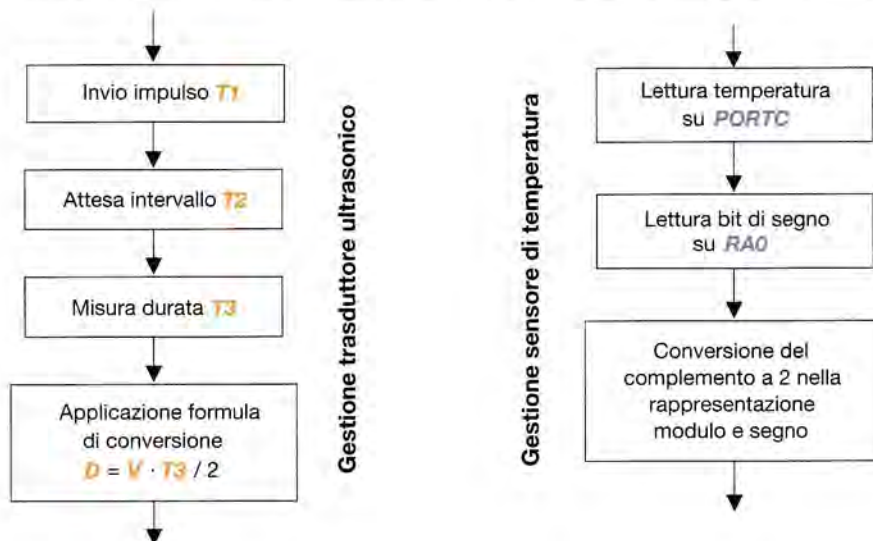
2. Il candidato disegni il diagramma di flusso per la gestione dei sensori, evidenziando le modalità di interfacciamento se in "polling", in "interrupt" o in modalità "mista".

Il sistema deve svolgere ciclicamente in sequenza i seguenti compiti, come specificato in dettaglio nei diagrammi di flusso:

1. lettura del sensore digitale e calcolo del valore assoluto e del segno della temperatura;
2. lettura del trasduttore ultrasonico e calcolo della distanza;
3. visualizzazione dei dati.

Anche se ciascuno di questi compiti richiede un dato tempo, questo non incide la funzionalità e la precisione del sistema, in quanto il microcontrollore non è impegnato in altri compiti.

È quindi possibile utilizzare il polling per interrogare la durata T_3 dell'impulso. La scelta è suffragata anche dalla considerazione che è il microcontrollore a inviare per primo l'impulso di attivazione al trasduttore ultrasonico e non viceversa.



3. Il candidato descriva almeno una metodologia software e/o hardware per rilevare il periodo T_3

Una possibile metodologia software connessa con l'impiego del timer consiste nel contare con una variabile **CONT** il numero di periodi di overflow del timer. Nota la durata T di ciascun periodo, si può calcolare la durata T_3 mediante la formula: $T_3 = T \cdot \text{CONT}$.

Noto il valore massimo di T_3 di 19,5 ms e sapendo che **CONT** non può oltrepassare 255 (è un byte) si deduce che il periodo del timer deve essere almeno: $T = T_3 / \text{CONT} = 19,5 / 255 \approx 76 \mu\text{s}$.

Scegliendo un valore maggiore come $T = 256 \mu\text{s}$ si ricava: $\text{CONT} = T_3 / T = 19,5 \text{ ms} / 256 \mu\text{s} = 76$.

Tutto ciò significa che l'intervallo $0 < T_3 < 19,5 \text{ ms}$ genera dei conteggi compresi nell'intervallo $0 < \text{CONT} < 76$.

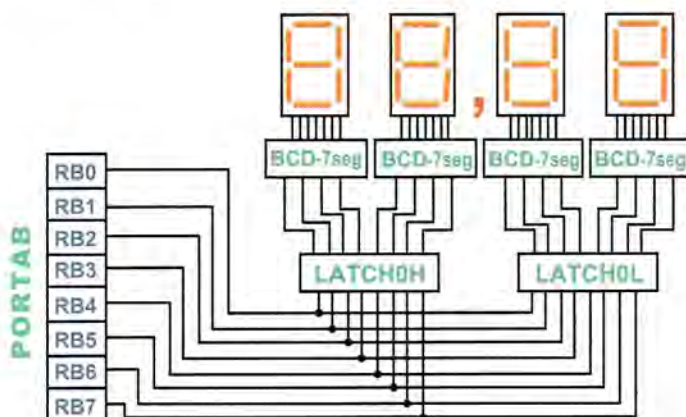
4. Il candidato indichi una metodologia per visualizzare le grandezze misurate dai sensori

Ipotizzando la precisione di due cifre decimali, la visualizzazione richiede quattro display separati da una virgola.

Ogni display viene pilotato da un decodificatore per display a 7 segmenti.

Il dato all'ingresso di ciascun decodificatore è di quattro bit, pertanto ogni coppia di decodificatori fa capo a un byte.

Per limitare l'impiego a una sola porta si possono moltiplicare i due byte mediante l'impiego di latch, opportunamente temporizzati dal microcontrollore.



5. Il candidato progetti il codice di gestione di almeno uno dei sensori utilizzati

Si riporta di seguito il codice in linguaggio C della misura di **T3** e il relativo diagramma di flusso.

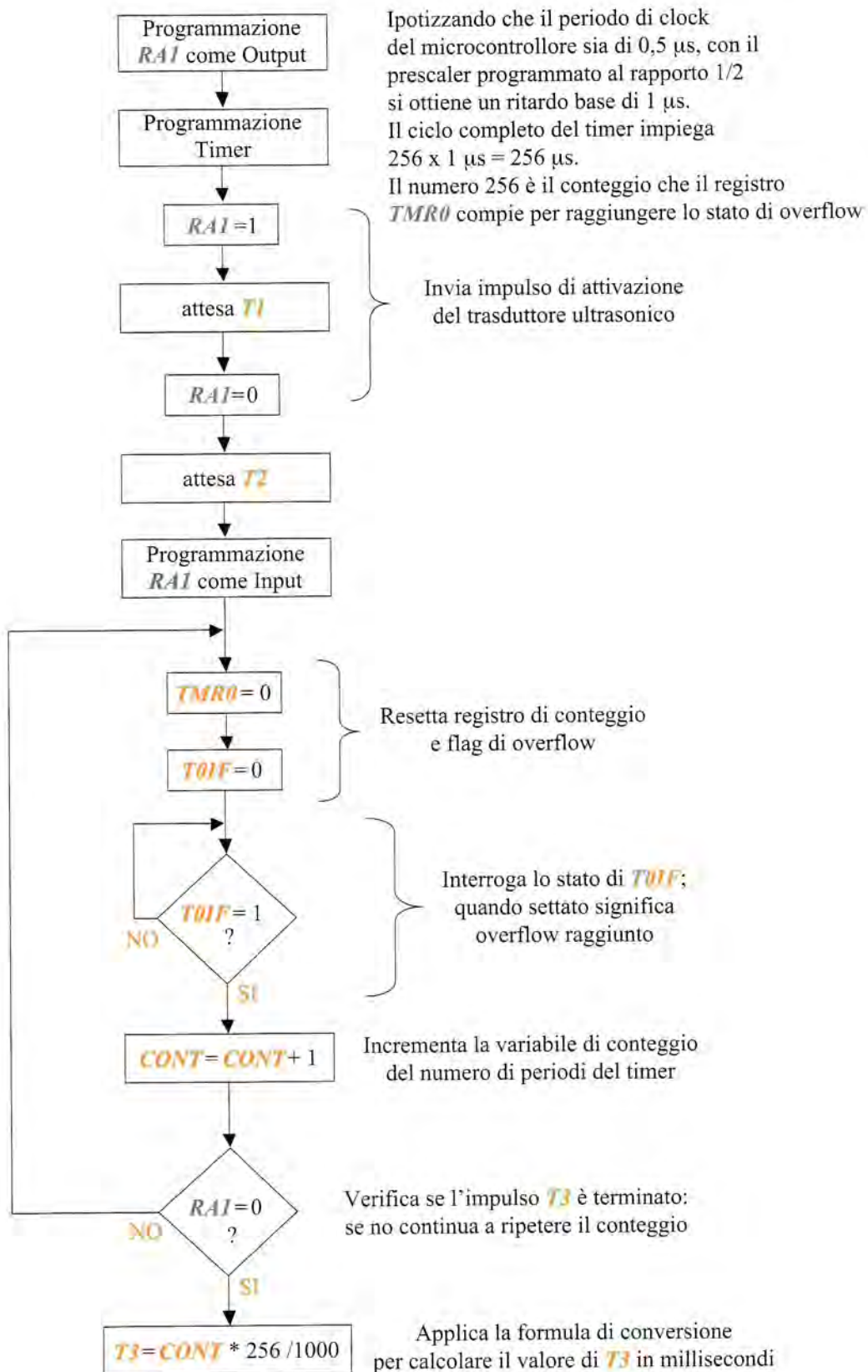
Il nucleo del programma risiede, come descritto nella risposta 3, nel conteggio delle ripetizioni di un ciclo di timer di durata nota e pari a 256 μ s.

```
#include <htc.h>
char CONT;           //CONT contatore numero di overflow
int T3;              //T3 durata dell'impulso T3
void main(void)
{
    OPTION_REG=0B10000000; //imposta timer con prescaler 1/2

    TRISA = 0B00000000;    //imposta porta A output
    while(1){
        RA1 = 1;           //impulso di 5µs per avvio sensore
        Delay_us(5);
        RA1 = 0;
        Delay_us(200);      //attesa di 200µs

        TRISA = 0B11111111; //imposta porta A input
        while(RA1==1)
        {
            TMRO=0;         //resetta registro timer
            TOIF=0;         //resetta flag overflow
            do {} while(TOIF==0); //attendi overflow
            CONT = CONT + 1; //incrementa conteggio degli overflow
        }

        T3 = CONT * 256 /1000 //calcolo di T3 in millisecondi
    }
}
```



6. Il candidato valuti la distanza massima rilevabile dal sensore ultrasonico supponendo la temperatura T costante e pari a $+20^\circ\text{C}$.

È sufficiente calcolare la velocità del suono alla temperatura di 20°C e da questa risalire alla distanza, imponendo nella relativa formula la durata massima dell'impulso $T_3 = 19,5 \times 10^{-3} \text{ s}$

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

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

Per determinare la variazione relativa della distanza rilevata è sufficiente ripetere il calcolo di D alla temperatura di 40°C .

$$V = a + b \cdot \text{Temp} = 331,5 + 0,62 \times 40 = 356,3 \text{ m/s}$$

$$D = V \cdot T_3 / 2 = 356,3 \times 19,5 \times 10^{-3} / 2 = 3,47 \text{ m}$$

La variazione subita vale: $D(40) - D(20) = 3,47 - 3,35 = 0,12 \text{ m} = 12 \text{ cm}$

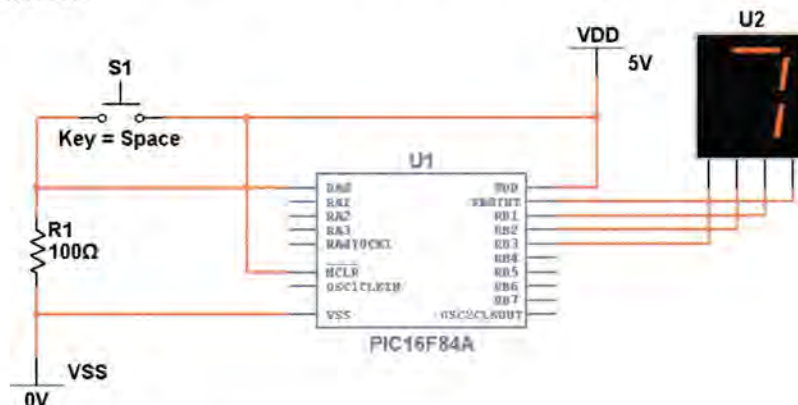
Extra: Implementazione pratica relativa alla misura della durata del livello alto di un segnale con il timer

L'applicazione impiega la stessa metodologia della risoluzione per misurare per quanti secondi l'utente mantiene premuto il pulsante $S1$.

Ipotizzando un periodo di clock di $1 \mu\text{s}$ e il prescaler $1/32$ si ricava un tempo di ritardo base uguale a $1 \mu\text{s} \times 256 \times 32 = 8,192 \text{ ms}$

Per la misura di un secondo si deve contare un numero di overflow uguali a $1/8,192 \times 10^{-3} = 122$.

L'evento di overflow viene rilevato mediante lo stato di $TOIF$, mentre il conteggio delle 122 ripetizioni viene effettuato mediante un ciclo for.



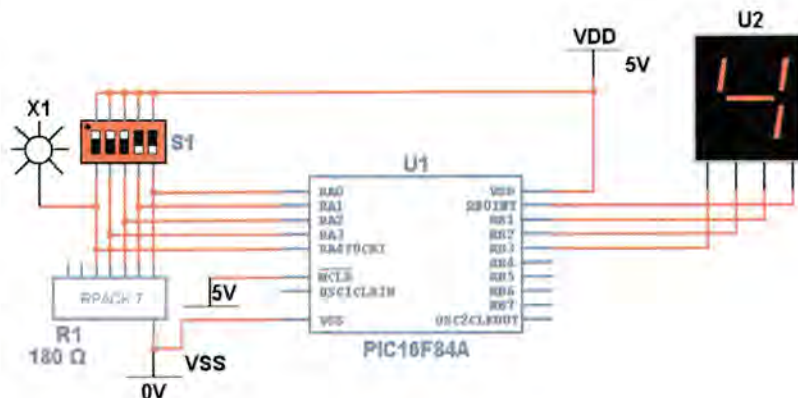
```
#include <htc.h>
char SECONDI=0, CONTATORE, i;
void main(void)
{
    TRISA = 0B11111111;           //imposta porta A input
    TRISB = 0B00000000;           //imposta porta B output
    OPTION = 0B10000100;           //imposta prescaler 1/32

    while(1){
        CONTATORE=122;             //imposta conteggio overflow
        while(RA0==1)               //ripeti se tasto premuto
        {
            for(i=0; i<CONTATORE; i=i+1)
            {
                TMRO=0;             //resetta registro timer
                TOIF=0;             //resetta flag di overflow
                do {} while(TOIF==0); //attendi evento di overflow
            }
            SECONDI=SECONDI+1;       //conta i secondi
            PORTB=SECONDI;          //visualizza i secondi
        }
    }
}
```

Extra: Implementazione pratica relativa al calcolo del complemento a due di un numero a 5 bit

Il circuito contempla uno switch a 5 posizioni, quattro per l'impostazione del numero binario a 4 bit e un quinto per l'impostazione del segno.

Nel programma sotto riportato se il bit di segno presente sul pin RA4 vale 0 viene trasferito in uscita il numero. Viceversa se il bit di segno vale 1 il numero viene complementato (complemento a 1) e ad esso viene sommato 1 (complemento a 2).



```
#include <htc.h>
unsigned int NUMERO;
void main(void)
{
    TRISA = 0B11111111; //imposta porta A input
    TRISB = 0B00000000; //imposta porta B output
    while(1) {
        if (RA4==0)
        {
            PORTB=PORTA;
        }
        else
        {
            NUMERO=PORTA;
            NUMERO=~NUMERO;
            NUMERO=NUMERO+1;
            PORTB=NUMERO;
        }
    }
}
```

ESAME DI STATO DI ISTRUZIONE SECONDARIA SUPERIORE
Indirizzo: ITAT - ELETTRONICA ED ELETTROTECNICA
ARTICOLAZIONE ELETTRONICA
Tema di: SISTEMI AUTOMATICI 2009

Si vuole monitorare, con cadenza di acquisizione pari a 10 secondi, la temperatura durante il processo di produzione di una particolare fibra plastica. Sapendo che:

- la temperatura è compresa tra 0 °C e 100 °C e a ogni variazione di 1 °C corrisponde una variazione di 10 mV,
- il numero di sensori di temperatura utilizzati nell'impianto è pari a 8,
- è necessario fornire in uscita, oltre alle temperature dei sensori, anche la temperatura media,
- le specifiche a cui lo strumento deve soddisfare sono:
 - il rapporto segnale/rumore del convertitore A/D non deve essere inferiore a ~ 72 dB,
 - la tensione di riferimento del convertitore A/D sia pari a $V_{ref} = + 5 \text{ V}$.

Il candidato, formulate le ipotesi aggiuntive che ritiene opportune:

1. individui uno schema a blocchi dell'impianto utilizzando un microcontrollore o un microprocessore conosciuto;

- sviluppi il circuito di condizionamento tra un sensore di temperatura ed un canale del convertitore analogico-digitale;
- indichi le risorse software in termini di registri e/o variabili utilizzate per realizzare l'algoritmo di acquisizione e di calcolo della temperatura media;
- disegni il diagramma di flusso dell'algoritmo per il calcolo della temperatura media;
- fornisca una porzione di codice significativa dell'algoritmo utilizzato;
- indichi almeno un tipo di circuito per visualizzare le grandezze acquisite ed elaborate.

SOLUZIONE

Colori utilizzati nella soluzione:

Arancio: Variabili di Programmazione

Blu: Segnali Circuitali

Verde: Dispositivi

1. Il candidato individui uno schema a blocchi dell'impianto utilizzando un microcontrollore o un microprocessore conosciuto

Nello schema riportato nella pagina a fianco i dispositivi sono organizzati secondo la classica **catena di acquisizione** e disciplinati dal **microcontrollore** il quale, opportunamente programmato, rileva le grandezze in input e impartisce in output gli opportuni comandi.

All'ingresso della catena sono presenti le otto fonti di temperatura, che vengono opportunamente **condizionate**, per garantirne l'adattamento con l'intervallo di tensione accettato dall'**ADC**.

Un **selettore analogico AMUX** a 8 ingressi-1 uscita, grazie ai segnali di indirizzo $A_2A_1A_0$ che giungono dal microcontrollore, seleziona in sequenza i segnali $I_0 + I_7$, provenienti dagli otto canali e li applica all'ingresso dell'ADC.

L'ADC compie la conversione con la supervisione del microcontrollore, che ciclicamente invia il segnale di inizio conversione **SOC** e ne attende la conclusione testando la linea **EOC**.

Considerato che nell'intervallo di 10 s vengono convertite 8 grandezze, per la conversione di una grandezza l'ADC dispone di $10/8 = 1,25$ s; la velocità di conversione non è pertanto assolutamente un dato critico per la scelta dell'ADC.

Inoltre, data la bassa velocità di variazione della temperatura, non è necessaria la presenza del modulo Sample Hold.

Per soddisfare la specifica di 72 dB del **rapporto segnale/rumore** la dimensione della parola digitale deve essere di **12 bit**, come di seguito dimostrato.

L'errore di quantizzazione fisiologico dovuto alla quantizzazione dei campioni analogici è definito, per un segnale sinusoidale di ampiezza V_{MAX} , come il rapporto tra la potenza S del segnale utile e la potenza N_Q del rumore di quantizzazione. La potenza del segnale utile è il quadrato del suo valore efficace

$$S = (V_{eff})^2 = \left(\frac{V_{MAX}}{\sqrt{2}} \right)^2 = \frac{V_{MAX}^2}{2}$$

Per un convertitore a n bit il range $-V_{MAX} + +V_{MAX}$ della sinusoide è codificato con 2^n livelli di campionamento, pertanto il quanto vale: $Q = 2 \cdot V_{MAX}/2^n$.

Si dimostra inoltre che il corrispondente rumore di quantizzazione, legato appunto all'ampiezza del quanto, vale: $N_Q = Q^2/12$. Si può quindi ricavare la formula del rapporto segnale/rumore:

$$\left(\frac{S}{N_Q} \right) = 10 \log \frac{S}{N_Q} = 10 \log \frac{\frac{V_{MAX}^2}{2}}{\frac{Q^2}{12}} = 10 \log 6 \cdot \frac{V_{MAX}^2}{Q^2} = 10 \log 6 \cdot \frac{V_{MAX}^2}{\frac{2^2 \cdot V_{MAX}^2}{(2^n)^2}} = 10 \log \frac{6}{4} \cdot 2^{2n}$$

$$\left(\frac{S}{N_Q} \right) = 10 \log \frac{3}{2} + 20 \cdot n \cdot \log 2 \approx 1,76 + 6n$$

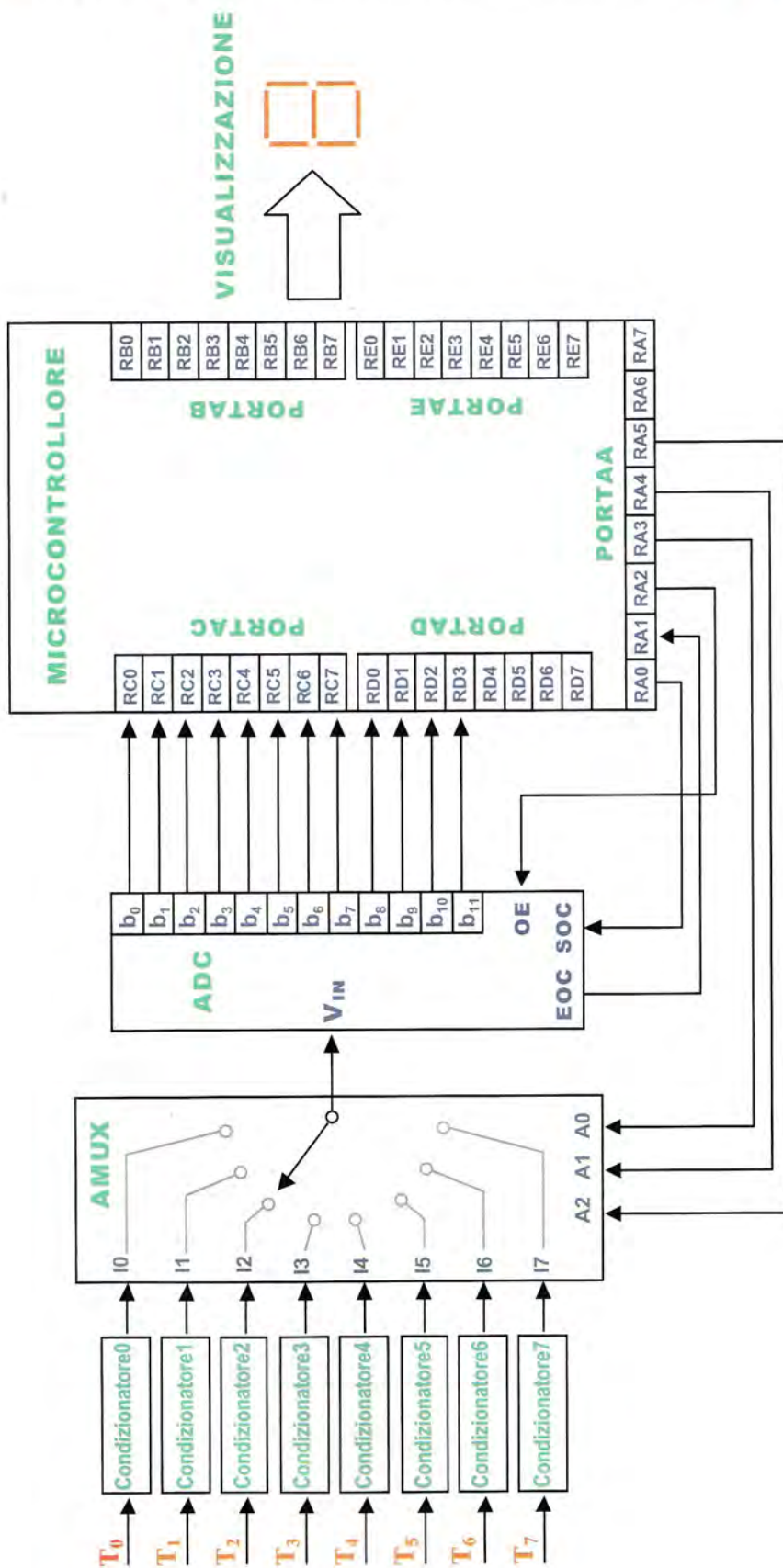
Imponendo $1,76 + 6n = 72$ si ricava: $6n = 72 - 1,76$ e quindi $n = 11,7$, che viene approssimato per eccesso a **$n = 12$** .

Con 12 bit la risoluzione risulta $R = 5V/2^{12} = 5/4096 \approx 0,001$ a tre cifre decimali, questa richiederebbe 5 display, due per le cifre prima della virgola e tre dopo, da 0,001 a 99,999.

Si sceglie per semplicità di utilizzare quattro display, con una risoluzione di due decimali, da 0,01 a 99,99.

Nello schema le porte di I/O del microcontrollore sono state così assegnate:

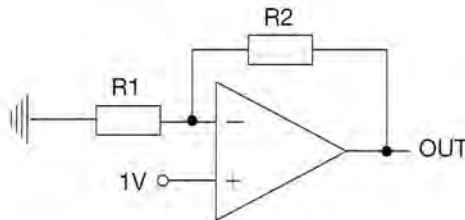
- bit0 della **PORTAA** \leftrightarrow **SOC**; bit1 della **PORTAA** \leftrightarrow **EOC**; bit2 della **PORTAA** \leftrightarrow **OE**
- bit3, bit4, bit5 della **PORTAA** \leftrightarrow uscite di selezione $A_0A_1A_2$
- PORTAC** \leftrightarrow byte basso della parola convertita
- PORTAD** \leftrightarrow nibble alto della parola convertita
- PORTAB** e **PORTAE** \leftrightarrow pilotaggio sezione di visualizzazione



2. Il candidato sviluppi il circuito di condizionamento tra un sensore di temperatura e un canale del convertitore analogico-digitale

Considerato che a una variazione di un grado corrisponde una variazione di 10 mV nella retta di trasduzione, la tensione massima fornita dai sensori è pari a: $10 \text{ mV} \cdot 100 \text{ }^{\circ}\text{C} = 10 \cdot 10^{-3} \cdot 100 = 1 \text{ V}$.

Poiché $V_{REF} = +5 \text{ V}$, questo valore deve essere amplificato di un fattore 5, utilizzando il seguente **amplificatore non invertente**:



Dalla formula dell'amplificatore si ha:

$$A = 1 + R_2/R_1 = 5 \rightarrow R_2/R_1 = 4$$

Fissato ad esempio $R_1 = 1 \text{ K}\Omega$ si ricava $R_2 = 4 \text{ K}\Omega$

3. Il candidato indichi le risorse software in termini di registri e/o variabili utilizzate per realizzare l'algoritmo di acquisizione e di calcolo della temperatura media

Ogni 10 secondi vengono raccolti gli 8 valori di temperatura entro altrettante locazioni di memoria.

Se si programma con un linguaggio di alto livello si deve predisporre pertanto una variabile di tipo **vettore** di 8 caselle:

0	1	2	3	4	5	6	7
T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7

Se si utilizza un linguaggio di alto livello la **MEDIA** viene ricavata semplicemente sommando con un ciclo for il contenuto delle 8 caselle e dividendo successivamente per 8.

Per l'algoritmo di acquisizione vengono utilizzate due variabili di tipo vettore:

- **DATO[8]** per memorizzare gli 8 valori digitali a 12 bit;
- **T[8]** per memorizzare i corrispondenti valori di temperatura reali.

4. Il candidato disegni il diagramma di flusso dell'algoritmo per il calcolo della temperatura media

Per calcolare la temperatura media il microcontrollore deve innanzitutto:

- acquisire i dati di temperatura
- convertire ogni dato digitale a 12 bit nel valore reale di temperatura.
- sommare gli 8 dati reali di temperatura e dividere per 8

Il dato presente su PORTD ha peso maggiorato perché corrispondente ai quattro bit più significativi del convertitore ADC. Il valore presente deve essere moltiplicato per $2^8 = 256$.

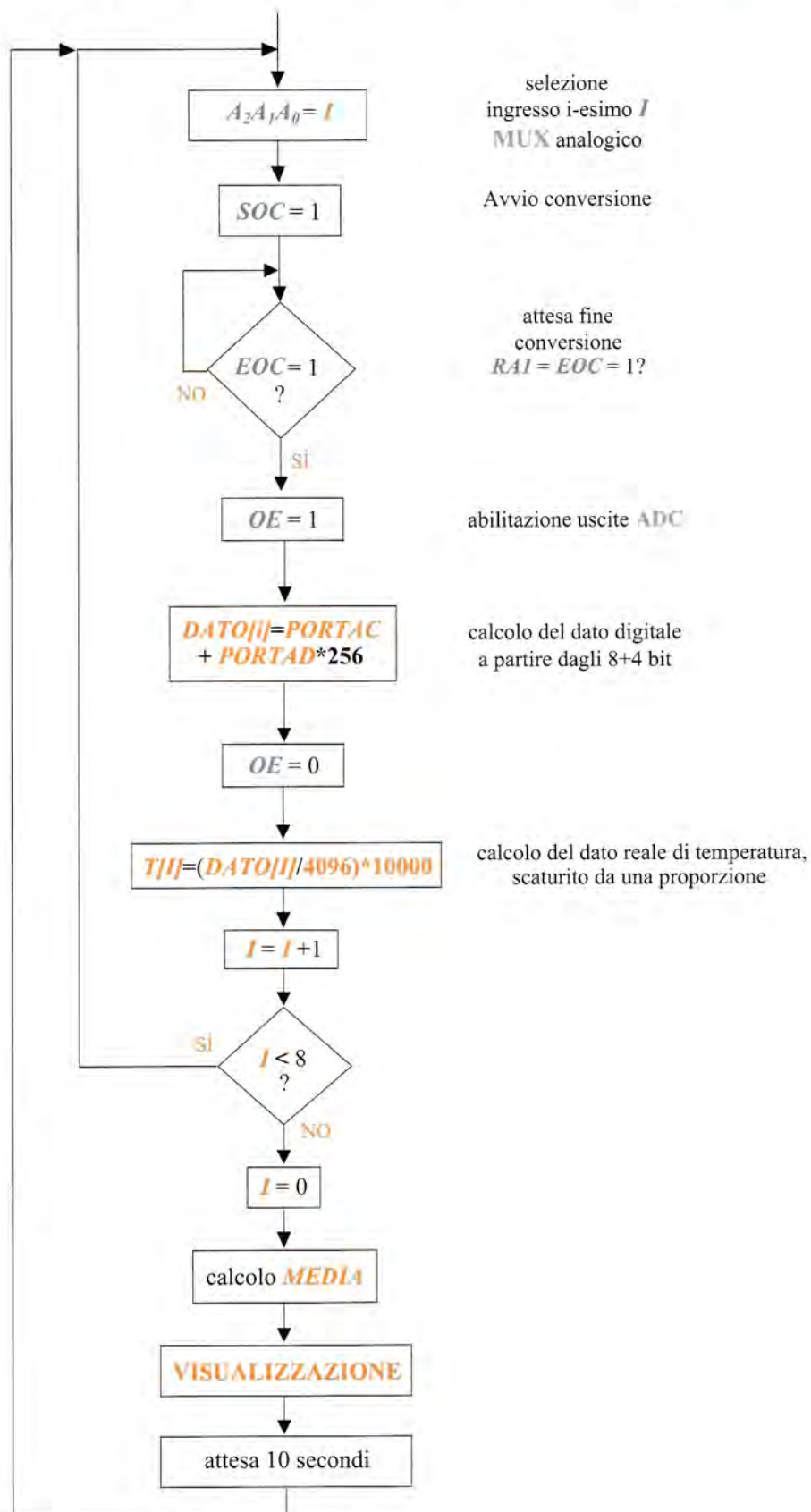
Di conseguenza la formula per il calcolo del dato è: $DATO = PORTAC + PORTAD * 256$.

Al dato digitale corrisponde la temperatura reale in base alla seguente proporzione (si tenga presente che 10000 è da intendersi visualizzato come 100,00):

$$DATO: 4096 = T : 10000$$

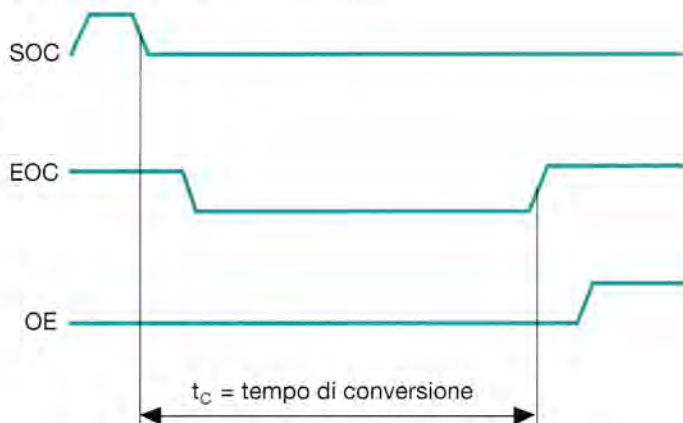
La formula di conversione del dato digitale in temperatura reale a 4 cifre (2 decimali) è dunque:

$$T = DATO \cdot \frac{10000}{4096}$$



5. Il candidato fornisca una porzione di codice significativa dell'algoritmo utilizzato

Si sceglie di considerare l'algoritmo di **conversione**. Per definirlo è necessario conoscere le relazioni che sussistono a livello temporale tra i vari segnali coinvolti nel processo di conversione. Per questo si riporta di seguito il diagramma di temporizzazione generico di un ADC:



In esso compaiono i seguenti segnali:

SOC = Start Of Conversion = Inizio Conversione

EOC = End Of Conversion = Fine Conversione

OE = Output Enable = Abilitazione Uscite

Come noto il segnale **SOC** avvia la conversione. Dopo un certo ritardo la linea **EOC** viene portata bassa e si mantiene tale fino alla fine della conversione, allorchè ritorna alta.

Il microprocessore può pertanto operare in questo modo

1. trasmette un impulso di inizio conversione **SOC** all'ADC; dopo un certo ritardo l'ADC porta bassa la linea **EOC** iniziando il processo di conversione; la sua conclusione viene notificata dal livello alto sulla linea **EOC**
2. attende fine conversione **EOC=1**
3. porta alta la linea **OE** per abilitare la lettura del dato
4. successivamente legge il dato dalle porte **PORTC** e **PORTD**
5. assegna i giusti pesi alle due porzioni lette per il calcolo del **DATO** digitale
6. calcola il valore reale **T** della temperatura, applicando la proporzione
7. arrotonda il risultato per il rendering digitale

```

PORTA.F0=1;           1) impulso SOC
PORTA.F0=0;
do
{
}
while (!PORTA.F1);    2) attesa EOC=1
PORTA.F2=1;          3) OE=1
BASSO=PORTC;         4) lettura DATO
ALTO=PORTD;
PORTA.F2=0;
DATO=BASSO+ALTO*256;  5) peso 256 per dato su PORTAD
T=(T/4096)*10000;    6) proporzione per valore reale T
T=floor(T);          7) arrotondamento mediante funzione floor

```

6. Il candidato indichi almeno un tipo di circuito per visualizzare le grandezze acquisite ed elaborate

L'implementazione qui fornita impiega 8 gruppi di 4 display a 7 segmenti per l'indicazione a quattro cifre degli otto dati di temperatura, più un gruppo di 4 per l'indicazione della media.

Ogni gruppo di 4 display è pilotato:

- da 4 decoder per display 7 segmenti
 - da due latch per la memorizzazione dei bit destinati al gruppo
- I latch sono necessari perché i gruppi sono tutti multiplati sulla **PORTAB**. Per ogni gruppo la **PORTAB** trasporta infatti:

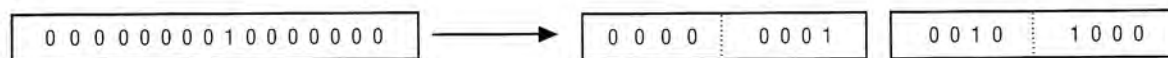
- un primo byte delle **MIGLIAIA-CENTINAIA** latchato su **LATCHXH**
- un secondo byte delle **DECINE-UNITA** latchato su **LATCHXL**

Gli impulsi destinati ai diversi latch sono impartiti dal programma e diramati da un **DECODIFICATORE 4/16** che riceve sulle linee $A_3A_2A_1A_0$ l'indirizzo di quale delle uscite $U0 \div U15$ attivare.

Il valore reale $T[i]$ della temperatura, una volta determinato, per poter essere visualizzato sui quattro display delle migliaia, centinaia, decine e unità, deve essere convertito in BCD e spalmato su due byte



Ad esempio il valore digitale $0000000010000000_{(2)} = 0128_{(10)}$ deve essere così rappresentato:



Per questo, con riferimento al byte *DECINE-UNITA*, si deve operare così:

- ricavare da $T[i]$ le singole cifre componenti
- collocare *UNITA* sul nibble basso
- collocare *DECINE* sul nibble alto

Per collocare *DECINE* sul nibble alto e *UNITA* su quello basso è necessario:

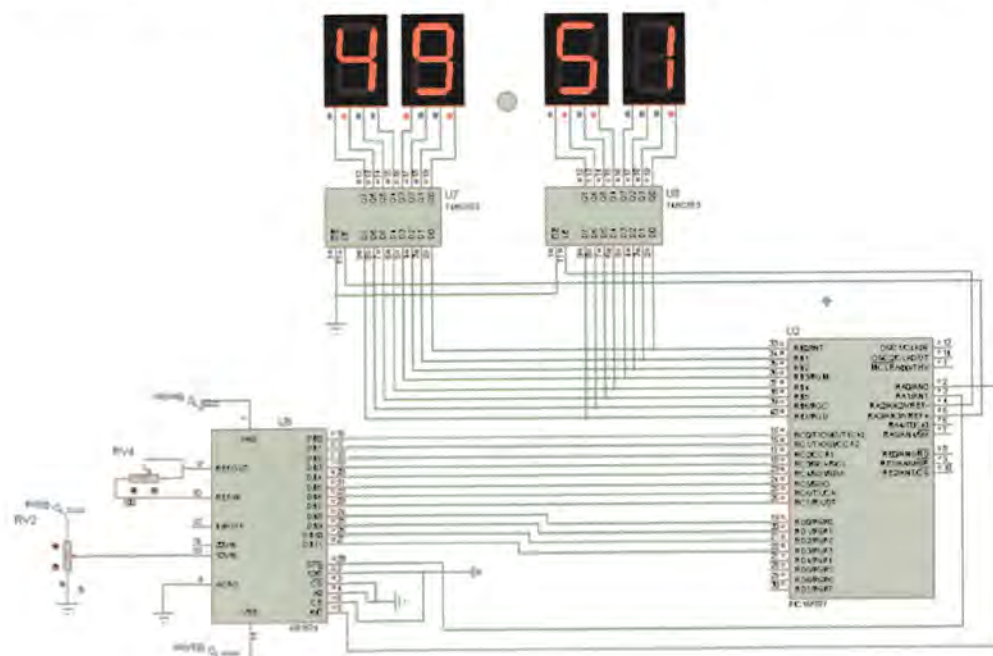
- portare *DECINE* nel nibble alto, traslando di quattro bit a sinistra oppure moltiplicando per 16
- aggiungere nel nibble basso le *UNITA*, con una operazione di somma oppure un OR

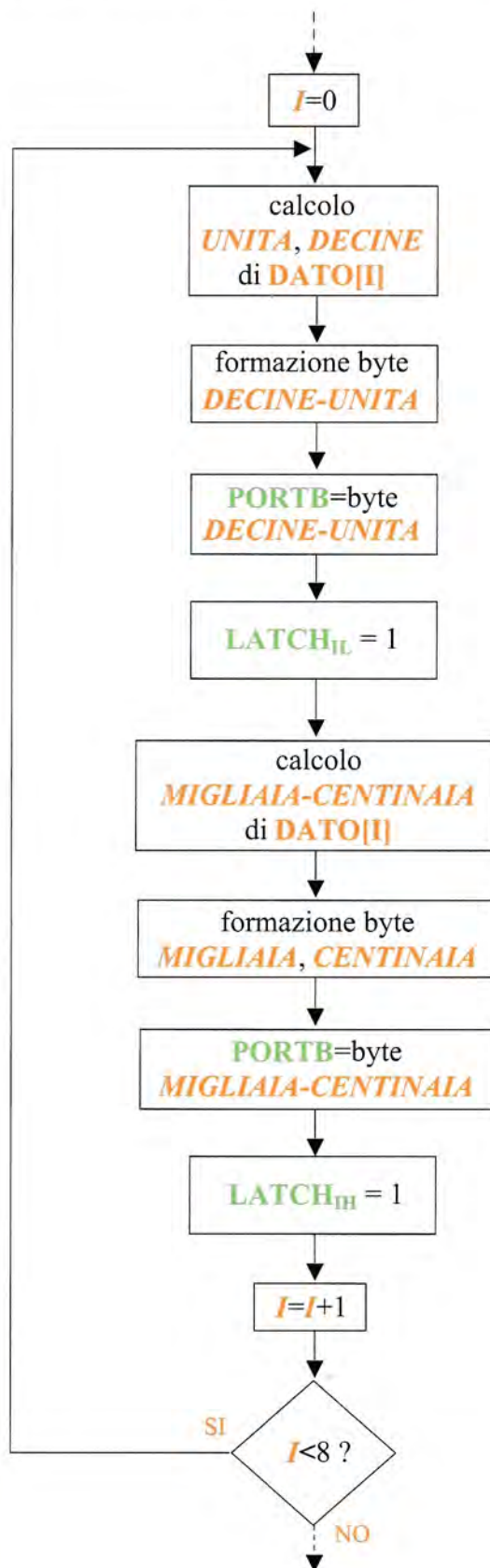
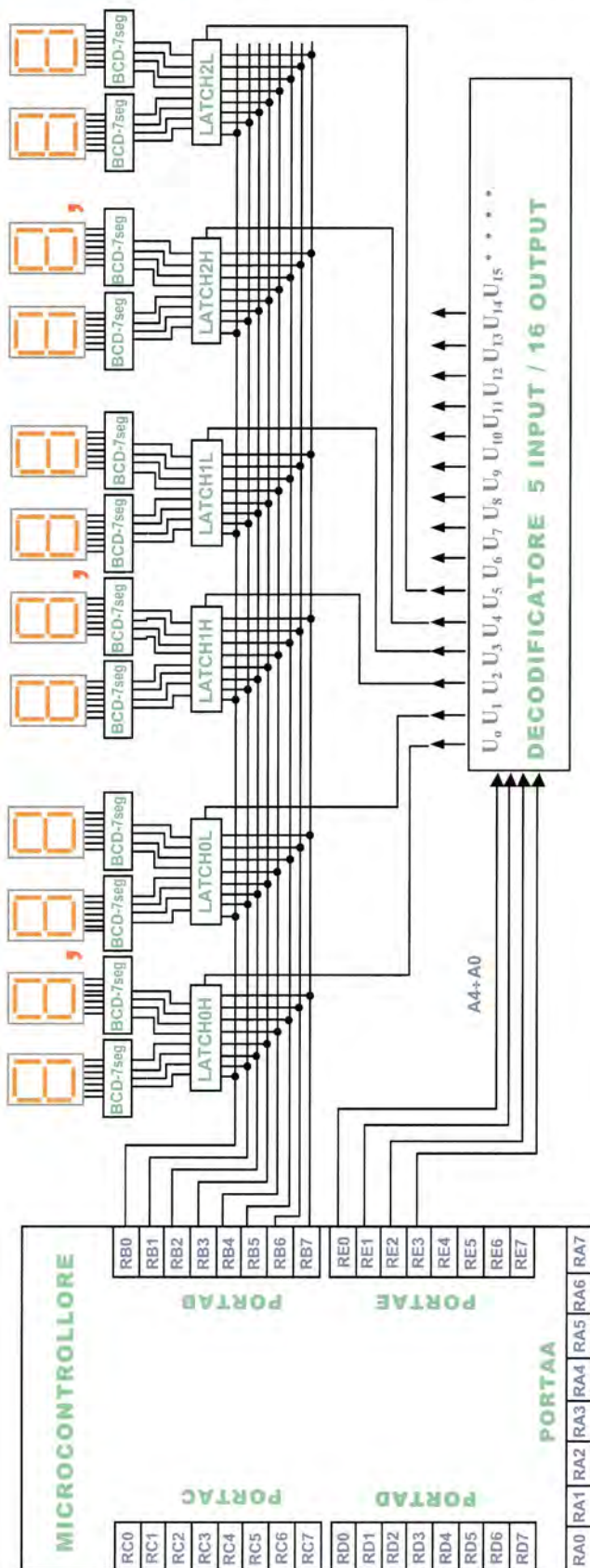
Ad esempio:



Per una comprensione precisa della visualizzazione delle otto temperature, si veda il diagramma di flusso riportato a fianco dello schema. La visualizzazione della media è analoga.

Extra: implementazione pratica relativa alla visualizzazione di un valore di temperatura





La simulazione è condotta mediante un microcontrollore PIC16F877 e il convertitore a 12 bit AD1674.

In questo integrato la conversione viene gestita attraverso i terminali *R/C* (Read/Convert) e *STS* in questo modo:

- *R/C* = 0 avvio conversione
- *R/C* = 1 lettura dato convertito
- *STS* = 1 conversione in corso
- *STS* = 0 conversione completata

Il programma è scritto in C.

```
int MIGLIAIA, CENTINAIA, DECINE, UNITA, T;
unsigned char BASSO, ALTO;
float DATO;
void main() {
    ADCON1 = 0B00000110;
    TRISA = 0B00000010;
    TRISB = 0;
    TRISC = 255;
    TRISD = 255;
    TRISE = 0;

    do {
        PORTA.F0=0; //inizio conversione
    }
    while (PORTA.F1); //attesa fine conversione
    PORTA.F0=1; //abilitazione uscite
    BASSO=PORTC;
    ALTO=PORTD;
    DATO=BASSO+ALTO*256; //dato digitale a 12 bit
    DATO=(DATO/4096)*10000; //valore di temperatura
    T=floor(DATO);
    UNITA=T%10; //visualizzazione
    T=floor(T/10);
    DECINE=T%10;
    T=floor(T/10);
    CENTINAIA=T%10;
    MIGLIAIA=floor(T/10);
    PORTB=UNITA+DECINE*16;
    PORTA.F2=1;
    PORTA.F2=0;
    PORTB=CENTINAIA+MIGLIAIA*16;
    PORTA.F3=1;
    PORTA.F3=0;
    Delay_ms(40);
} while(1);
} //~
```