

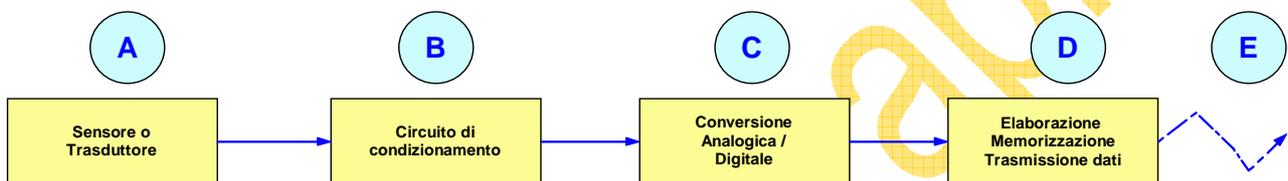
# PROGETTAZIONE DI UN “SISTEMA” ELETTRONICO

Partiamo, a solo scopo esemplificativo, dal **tema di esame** dell’A.S. 2001/2002.

L’importante a mio avviso è, prima, leggere attentamente il testo e cercare di comprendere a grandi linee tutto il “sistema” che verrà progettato, cercando di comprendere bene le richieste del “cliente”.

Nel caso in questione abbiamo un piccolo albergo di 8 stanze, è chiaro che se vogliamo regolare temperatura ed umidità in ciascuna stanza dovremo avere una coppia di sensori in ciascuna di esse. Prima esemplificazione mentale, consideriamo una sola stanza e poi moltiplicheremo per 8 volte il circuito progettato. Sembra banale pensare così, ma spesso non lo è, la soluzione semplice è sempre la migliore.

Allora posso iniziare a “*progettare*” il dispositivo di rilevazione di temperatura ed umidità. Prima disegniamo uno schema a blocchi molto semplificato che serve anche solo a chiarirci le idee e, soprattutto, a far capire a chi dovrà correggere il compito che noi abbiamo le idee chiare.



Teniamo anche presente che il dato rilevato dal sensore dovrà essere inviato ad una centralina di raccolta ed elaborazione che deciderà poi se accendere la caldaia, l’umidificatore, ecc.

Abbiamo quindi una linea di “*comunicazione*”, vedremo in seguito quale tecnologia usare per la trasmissione dei dati; a livello di partenza può bastare la semplice dicitura di “*linea di trasmissione*”.

Il sensore o trasduttore è un dispositivo appositamente costruito che esegue la trasformazione di una qualsiasi grandezza fisica ( peso, temperatura, massa, lunghezza, velocità, ecc. ecc. ) in una grandezza elettrica che potrà essere elaborata successivamente con circuiti elettronici.

- A)** Il primo blocco è sempre il sensore, di esso in genere viene fornita sempre la caratteristica di risposta. In uscita da esso avremo sempre un segnale elettrico ( in genere tensione o corrente, ma a volte anche resistenza, capacità, ecc. ).
- B)** Quasi mai l’uscita fornita dal sensore è adatta ai nostri scopi, per cui seguirà sempre un blocco di “*condizionamento*” del segnale. Un circuito, cioè, che ha il compito di riportare i parametri elettrici del sensore a livelli a noi più utili. Ad esempio il sensore di temperatura del tema di esame ha un’uscita in tensione tra 20 mV e 150mV (rispettivamente ad inizio e a fine scala), mentre per i nostri scopi occorre che i due limiti inferiore e superiore della scala siano compresi, ad esempio, tra 0V e 5V. Normalmente i valori di uscita normalizzati sono sempre questi, corrispondenti ai limiti dei più diffusi convertitori A/D. In ogni caso c’è sempre un problema di scalatura dei valori (**range**) che si può risolvere con un amplificatore, od un problema di azzeramento (**offset**) che si può risolvere con un circuito sommatore.
- C)** In genere l’elaborazione di un dato letto da un sensore è più efficiente e flessibile se viene realizzata con segnali digitali. Con un microprocessore possiamo risolvere praticamente tutti i problemi che si presentano in maniera software ovvero modificando solo un programma e

non ricalcolando e riprogettando circuiti elettronici (dovendo quindi ad esempio rifare anche il circuito stampato). Occorre quindi convertire il segnale analogico in digitale.

- D)** Il segnale così digitalizzato può essere facilmente elaborato, memorizzato e, soprattutto, inviato ad una centralina di elaborazione dati (anch'essa gestita da un microprocessore).
- E)** Il segnale digitalizzato viene inviato su una linea di trasmissione dati alla centralina di raccolta dati. Della tecnologia usata per la trasmissione dei dati parleremo in seguito, può essere ad esempio una trasmissione seriale via cavo (parlerò ad esempio del protocollo RS-232), una trasmissione via radio (parlerò quindi della modulazione AM, FM, FSK, ecc.) oppure una trasmissione su fibra ottica (parlerò ad esempio delle fibre ottiche).

A questo punto facciamo una riflessione: in 6 (sei) ore non possiamo progettare l'intero sistema di rilevazione e controllo di cui ci è stato richiesto nel tema. A mio avviso in una prova di maturità dobbiamo dimostrare a chi ci sta di fronte che siamo "maturi", ovvero che di fronte ad un problema, anche complesso, siamo in grado di ragionare, di analizzare, di prendere in esame tante variabili, di avere un'idea "generale" del sistema e quindi di cominciare a sviluppare il lavoro. Quindi la parte generale ovvero la discussione iniziale in cui si analizza il problema e si schematizza a "blocchi" il sistema è la più importante e va sicuramente fatta.

Si tratta, poi, di decidere tra tutti i blocchi presenti nel sistema quale andrà sviluppato in dettaglio. Chiaramente io cercherei di sviluppare un argomento in cui sono più "ferrato". Per esempio se sono "bravo" nel progetto di semplici amplificatori operazionali, cercherò di sviluppare il blocco (B) composto da amplificatori e sommatore. Se sono "bravo" nel digitale cercherò di sviluppare il blocco (C) relativo al convertitore A/D, magari con la circuiteria di **sample & hold**, magari con un timer od un contatore per effettuare un campionamento ogni 30 minuti (come richiesto dal cliente). Se sono invece "bravo" in telecomunicazioni svilupperò la parte relativa alla trasmissione dei dati nei blocchi (D) ed (E); potrò scegliere un tipo di comunicazione seriale sincrona o asincrona, una comunicazione a bus o ad anello, una comunicazione via radio, via fibra ottica, ecc. (posso cioè scegliere quello che voglio!!). Se invece sono più esperto nel software o nei circuiti hardware relativi ai microprocessori posso sviluppare il progetto della centralina di raccolta dati (o a livello circuitale: microprocessore, memorie, circuiti di clock, circuiti di In/Out oppure a livello software parlando cioè di quello che dovrà fare il programma, magari con uno schema di diagramma di flusso a blocchi o addirittura con porzioni di codice in uno qualsiasi dei linguaggi conosciuti: assembler, C, pascal, basic, ecc. ).

Vediamo in dettaglio i vari blocchi.

## Sensore o trasduttore (A)

Per quello che riguarda questo blocco in realtà non c'è niente da calcolare poiché in genere viene fornito come dato di progetto e difficilmente viene chiesto di scegliere un sensore o di dimensionarne alcune caratteristiche. Per fare questo occorrerebbe ad esempio consultare cataloghi cartacei od eseguire ricerche su Internet e questo durante l'esame non è possibile farlo.

## Blocco di condizionamento (B)

La parte che cercherei di preparare meglio è sicuramente la prima: quella relativa al blocco di condizionamento (B) che in realtà si riferisce a pura Elettronica; così intanto preparo meglio una materia sola invece che due e posso ricavare spunti anche per la successiva prova orale di Elettronica che in genere parte con il compito scritto di Sistemi davanti agli occhi...

1

Prendiamo in esame il sensore di temperatura che ha una risposta del tipo:

$$V_{(t)} = V_o + k \cdot T \quad \text{dove } T \text{ è espresso in } ^\circ\text{C}, V_o = 0,45 \text{ V e } k = 15 \cdot 10^{-3}$$

ha quindi una risposta in tensione. La formula si legge così: La tensione di uscita (alla temperatura  $t$ ) dipende da un valore iniziale  $V_o$  (ovvero tensione di uscita alla temperatura di riferimento che in genere è  $0^\circ\text{C}$  ma non è detto) a cui viene sommato un valore dato dal prodotto di una costante  $k$  (che dipende dal sensore stesso) e dalla differenza di temperatura che c'è fra quella di riferimento ( $t_o$ ) e quella in esame ( $t$ ). Se la temperatura  $t$  è minore di quella di riferimento  $t_o$ , allora questa somma diventa una differenza (perché la  $\Delta t$  diventa negativa), cioè la tensione di uscita scende al di sotto della tensione  $V_o$ . Nel testo non è indicato quale è la temperatura di riferimento, sembra che sia da intendere  $0^\circ\text{C}$  anche perché viene indicato che  $T$  viene espressa in  $^\circ\text{C}$  e non ad esempio in K (kelvin). In ogni caso se esiste qualche dubbio prendi tu una decisione, ovvero scrivi chiaramente nel testo che da quello che hai capito la temperatura di riferimento è  $0^\circ\text{C}$  e da lì parti per fare i tuoi calcoli.

Il sensore ha una risposta di tipo lineare tra  $-15^\circ\text{C}$  e  $+50^\circ\text{C}$  (lo dice il testo e si vede anche dalla formula che descrivere la risposta  $\rightarrow$  è una equazione di primo grado e quindi rappresentabile con una retta) e siccome noi lo utilizzeremo nel range  $20^\circ\text{C} \dots 28^\circ\text{C}$  la risposta sarà sicuramente lineare (altrimenti dovremo prevedere un circuito di linearizzazione, cosa sicuramente complicata). Anche questa annotazione va messa nel testo, non costa nulla in termine di calcoli ma fa vedere a chi legge che stai considerando tutti i particolari.

Troviamo il range di uscita del sensore ovvero la differenza di tensione che fornisce in uscita tra inizio scala (temperatura minima) e fine scala (temperatura massima). Nota che il range si ottiene sempre come differenza tra valore di fondo scala e valore di inizio scala.

$$V_{(20^\circ\text{C})} = V_o + k \cdot T = 0,45\text{V} + 15\text{mV}/^\circ\text{C} \cdot 20^\circ\text{C} = 0,45 + 15 \cdot 10^{-3} \cdot 20 = 0,75\text{V}$$

$$V_{(28^\circ\text{C})} = V_o + k \cdot T = 0,45\text{V} + 15\text{mV}/^\circ\text{C} \cdot 28^\circ\text{C} = 0,45 + 15 \cdot 10^{-3} \cdot 28 = 0,87\text{V}$$

$$\text{range} \Rightarrow \Delta V = V_{\text{MAX}} - V_{\text{MIN}} = 0,87\text{V} - 0,75\text{V} = 0,12\text{V} = 120\text{mV}$$

Come si vede la differenza di tensione tra gli estremi della scala è piuttosto piccola ed in genere è sempre così.

2

A questo punto comincio a prendere altre decisioni: il convertitore A/D che utilizzerò ha un range di ingresso tra 0V e 5V. Posso scegliere io, dato che non mi sono stati imposti dei vincoli di progetto. Ovviamente qui starebbe all'esperienza del progettista conoscere quali tipi di convertitori A/D esistono sul mercato.

Come dato generale prendiamo questo: dato che utilizzerò un circuito a microprocessore, alimentato in genere a 5V, posso tranquillamente presumere che anche il convertitore A/D sia alimentato a 5V e quindi utilizzi un range di ingresso tra 0V e 5V. Ricorda che sei tu a scegliere e, soprattutto, scrivi chiaramente nel testo perché sei arrivato a questa scelta.

3

Cominciamo ad avere tutti i dati di progetto del nostro circuito di condizionamento. Abbiamo in ingresso un segnale che ha un range di 0,12V e dovremo avere in uscita un segnale che abbia un range di 5V.

Nota bene che l'uscita del circuito di condizionamento rappresenta il segnale di ingresso dello stadio seguente cioè del convertitore A/D.

Inoltre conviene in ogni caso sfruttare tutto il range di ingresso del circuito seguente per avere la massima sensibilità per apprezzare anche le più piccole variazioni di temperatura.

$$\Delta V_{OUT} = V_{OUT_{MAX}} - V_{OUT_{MIN}} = 5V - 0V = 5V$$

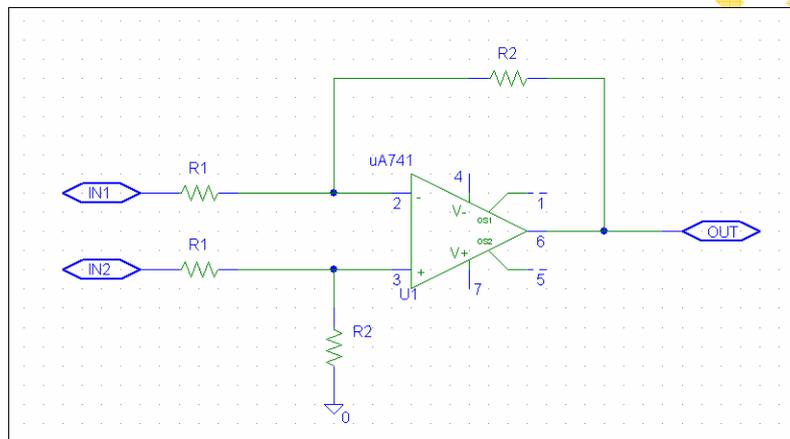
Il nostro circuito di condizionamento dovrà essere quindi un amplificatore (deve cioè “aumentare” il range di tensione). Se dovevamo diminuire il range occorreva utilizzare un attenuatore.

Quanto dovrà “amplificare” il nostro circuito di condizionamento?

Calcoliamo il guadagno o amplificazione in tensione:

$$A_V = \frac{\Delta V_{OUT}}{\Delta V_{IN}} = \frac{5V}{0,12V} = 41,67$$

Progettiamo quindi un amplificatore con un guadagno di 41,67 volte. Realizziamolo con gli Amp.Op. che sono facilmente gestibili, basta modificare il valore di alcune resistenze per modificare il guadagno!!



Questo è lo schema di un generico amplificatore differenziale, ricordo a memoria solo questo schema e lo utilizzo per tutti i casi che mi si presentano. Tutte le altre configurazioni derivano da questa e risalgo ad esse molto facilmente. Ne so progettare uno solo ma in realtà li so fare tutti con una sola formula!!

$$A_V = \frac{R_2}{R_1} \quad (\text{semplice da ricordare!!})$$

**Attenzione** che  $R_1$  ed  $R_2$  posso avere anche le sigle scambiate tra loro oppure avere anche altre sigle. Io questa formula la ricordo in modo “mentale-visuale” in questo modo: l’amplificazione è uguale al rapporto tra la resistenza di “sopra” e la resistenza posta sull’ingresso.

Tornando al nostro circuito che è un amplificatore differenziale, ma che si può usare anche come normale amplificatore invertente oppure non invertente.

Se uso entrambi gli ingressi applicandovi due segnali diversi otterrò in uscita la differenza tra  $IN_2$  ed  $IN_1$  amplificata di  $R_2/R_1$  volte:

$$V_{OUT} = (V_{IN2} - V_{IN1}) \times \frac{R_2}{R_1}$$

Se mi serve un amplificatore invertente uso l’ingresso  $IN_1$  e collego  $IN_2$  a massa. In altre parole l’ingresso non invertente è collegato a massa con una  $R_3=R_1//R_2$ .

La tensione di uscita sarà uguale a:

$$V_{OUT} = (0 - V_{IN1}) \times \frac{R_2}{R_1} = -V_{IN1} \times \frac{R_2}{R_1}$$

se mi serve non invertente faccio l’esatto contrario. Il segnale utile viene applicato su  $IN_2$ , mentre sull’altro ingresso è presente la coppia di resistenze  $R_1$  ed  $R_2$  che determinano il guadagno dell’amplificatore.

La tensione di uscita sarà uguale a:

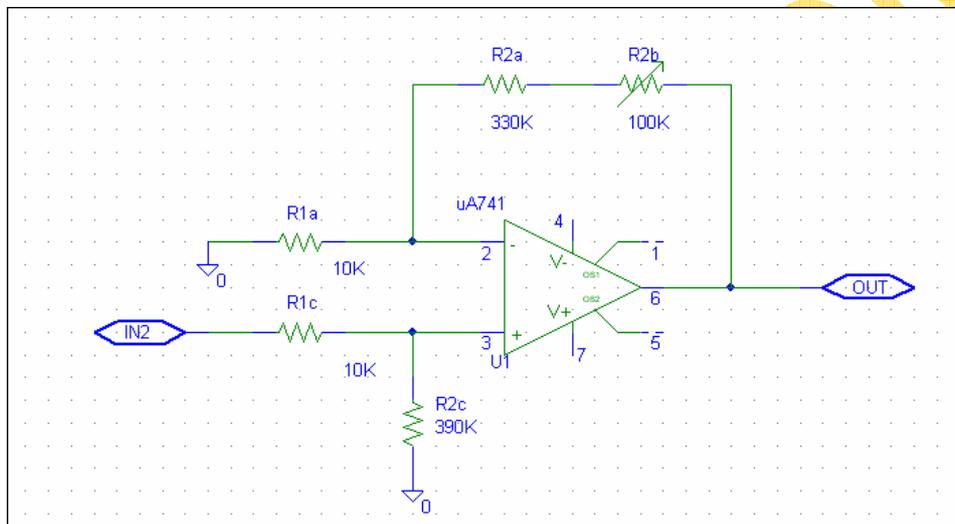
$$V_{OUT} = (V_{IN2} - 0) \times \frac{R_2}{R_1} = +V_{IN2} \times \frac{R_2}{R_1}$$

Posso a questo punto, conoscendo il guadagno da ottenere, calcolare  $R_1$  ed  $R_2$ . Siccome si tratta di due incognite di cui conosco il prodotto ho infinite soluzioni a questa equazione. Si procede imponendo noi un valore a piacere, ricavando l'altro. In realtà qualsiasi valore non va bene, usando dei comuni Amp.Op. in commercio ( $\mu A741$ , LM314, ecc.) in pratica non vanno usate resistenze al di sotto di alcuni  $K\Omega$  o al di sopra delle centinaia di  $K\Omega$ .

Ad esempio non sceglierei  $R_1 = 1 K\Omega$  da cui otterrei  $R_2=41,67 K\Omega$ , ma proverei con  $R_1=10 K\Omega$  da cui ottengo  $R_2=416,7 K\Omega$ .

Ricordiamoci che voglio ottenere  $A_V = 41,67$  e che  $AV = R_2/R_1$ .

Siccome non esistono in commercio resistenze da  $416,7 K\Omega$  otterrò questo valore con una resistenza fissa da  $330 K\Omega$  ed una variabile in serie da  $100 K\Omega$  che regolerò in fase di taratura del circuito.



$R_{2C}$  non è così critica come quella del ramo superiore che invece determina l'amplificazione. Il criterio generale è quello per cui le impedenze "viste" dai morsetti di ingresso dell'Amp.Op. verso l'esterno devono essere il più possibile simili perché le correnti parassite su questi ingressi, se sono uguali, si annullano esattamente fra di loro poiché un ingresso è di polarità opposta all'altro. Quindi in uscita avrò questa differenza fra le tensioni parassite amplificata  $A_V$  volte ma, dato che si tratta quasi sempre di una differenza tra valori simili, tale valore è sempre prossimo a  $0V$  e di conseguenza anche l'effetto in uscita sarà prossimo allo zero.

4

Adesso salta fuori il secondo problema da risolvere con il circuito di condizionamento. Facciamo queste due serie di calcoli: troviamo la tensione di ingresso a  $20^\circ C$  e a  $28^\circ C$  (ai due limiti della scala) e di conseguenza anche la tensione di uscita a queste due temperature.

Temperatura	Uscita sensore Ingresso c.d.c.	Range di ingresso c.d.c.	Amplificazione $A_V=41,67$	Uscita Amp.Op. (c.d.c.)	Range di uscita c.d.c.
$20^\circ C$	0,75V	0,12V	$0,75V * 41,67$	31,25V	5,0V
$28^\circ C$	0,87V		$0,87V * 41,67$	36,25V	

Il range desiderato in uscita è corretto: tra tensione massima e minima ci sono  $5V$  di differenza. Il fatto è che ad inizio scala non ci sono  $0V$ , bensì  $31,25V$ !! Il convertitore A/D nella migliore delle ipotesi è già andato!! Inoltre all'uscita di un Amp.Op. non ci può essere una tensione superiore a

quella di alimentazione, anzi sarà al massimo leggermente inferiore perché l'Amp.Op. sarà in saturazione. Siccome l'Amp.Op. è alimentato probabilmente con una tensione differenziale di  $\pm 12V$  o  $\pm 15V$  (sono io a decidere) in uscita non potrò ottenere tensioni superiori, ad esempio, a circa  $+11V$  nel primo caso e circa  $+14V$  nel secondo caso.

Nel nostro caso dovendo ottenere in uscita una tensione massima di  $5V$  è più che sufficiente stabilire una tensione duale di alimentazione di  $\pm 12V$ .

Quello che si presenta è il classico problema di *offset* (azzeramento), che si risolve utilizzando un circuito sommatore o sottrattore: si tratta cioè di "spostare" o traslare verso l'alto (somma) o verso il basso (sottrazione) un livello di tensione.

Nel nostro caso di tratterebbe di sottrarre all'uscita una tensione fissa di  $31,25V$ , in modo da avere una tensione minima in uscita di  $0$  volt, ovvero l'inizio della scala è fissato a zero volt. In questo modo i due limiti di scala diventeranno di  $0$  volt e di  $36,25 - 31,25 = 5$  volt. Il range rimane lo stesso di prima poiché ho sottratto ad inizio e fine scala la stessa quantità.

Il problema però è sempre meglio risolverlo alla fonte, ovvero invece di risolverlo in uscita quando ormai l'amplificatore è già in saturazione, conviene risolverlo all'ingresso prima che il problema si presenti.

All'ingresso dobbiamo però sottrarre  $31,25V / A_v$  perché c'è di mezzo un amplificatore!!!

In ingresso dovremo quindi sottrarre una tensione di  $V_{offset} = 31,25 / A_v = 31,25 / 41,67 = 0,75 V$ , valore molto più basso e che, soprattutto, fa in modo di non mandare in saturazione l'amplificatore operazionale.

Guarda caso il valore da sottrarre coincide con la tensione minima di ingresso!!!

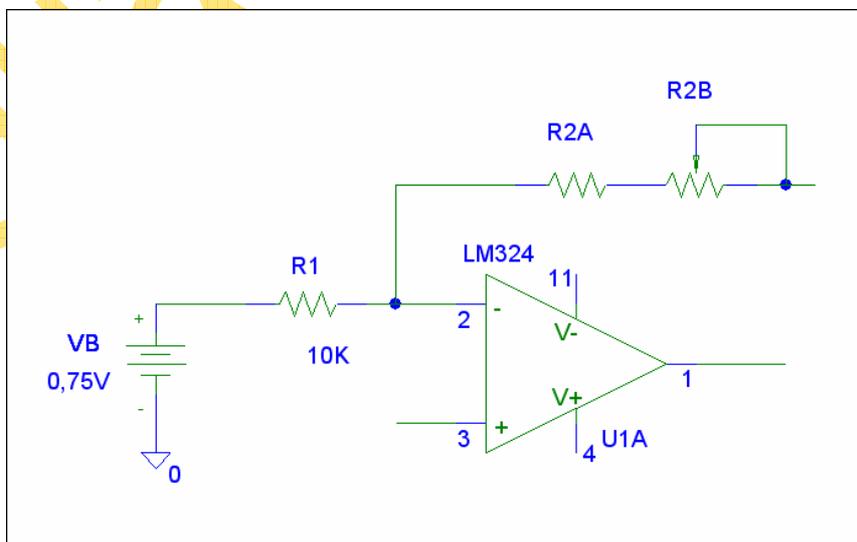
Non è però un caso!!!

Poiché alla tensione minima di ingresso ( $0,75V$ ) dovrà corrispondere la tensione minima di uscita che desidero sia  $0V$ , quale sarà quel numero che moltiplicato per  $41,67$  ( $A_v$ ) darà come risultato  $0$  (zero)? Naturalmente potrà essere solo lo zero. Per cui devo fare in modo che la tensione di ingresso al minimo sia zero, quindi quale numero sottratto da  $0,75V$  mi dà come risultato zero? Naturalmente solo se stesso!!! La tensione di offset da annullare sarà quindi di  $0,75V$  che andranno sottratti al segnale presente in ingresso.

Devo sottrarre quindi  $0,75V$  dal segnale di ingresso, come posso fare?

Ho bisogno di un circuito sottrattore. Un altro circuito? No! Se ho usato un Amp.Op. che ha due ingressi e ne ho usato uno solo per il sensore, rimane un ingresso libero che attualmente è stato collegato a massa perché inutilizzato.

Se invece di collegare a massa  $IN_1$  lo collego ad una sorgente di tensione continua (ad es.: una batteria) di  $0,75V$  ho risolto il tutto.



Se riflettiamo bene l'ingresso  $IN_1$  era collegato a massa ovvero ad una sorgente di 0V e quindi sottraeva 0 volt alla tensione che giungeva dal sensore sull'altro ingresso dell'Amp.Op., quindi il suo effetto era nullo.

Adesso l'uscita del circuito di condizionamento fornisce 0V a 20°C e 5V a 28°C !!

Per quello che riguarda la tensione di offset ho esemplificato con l'uso di una batteria.

In realtà non esistono batterie da 0,75V, ma anche se esistessero avrebbero senz'altro un costo non indifferente. Come si può ovviare? Semplice! Dato che abbiamo a disposizione una tensione di riferimento termo-stabilizzata a 5V che viene usata anche dal convertitore A/D (ne ha bisogno come riferimento per avere una conversione il più possibile precisa), oppure in ogni caso esiste l'alimentazione a 5V per il microprocessore (un po' meno precisa), da questa tensione, utilizzando un semplice partitore resistivo, con resistenze di precisione a tolleranza  $\pm 1\%$ , eventualmente con una resistenza variabile in serie, posso ottenere gli 0,75V desiderati.

L'unica accortezza è di stare attenti che la  $I_1$  sia molto piccola rispetto alla  $I_{B2}$ , questo si ottiene con una semplice regola:  $R_1 \gg R_{B2}$  (si legge  $R_1$  molto maggiore di  $R_{B2}$ , dove per molto maggiore si intende almeno 10 volte maggiore di), oppure leggendolo a rovescio  $R_{B2} \ll R_1$  (dato che  $R_1$  è già stata dimensionata), quindi se  $R_1 = 10 \text{ Kohm}$  posso tranquillamente fissare  $R_{B2} = 1 \text{ Kohm}$  e calcolarmi l'altra resistenza del partitore per ottenere la tensione desiderata.

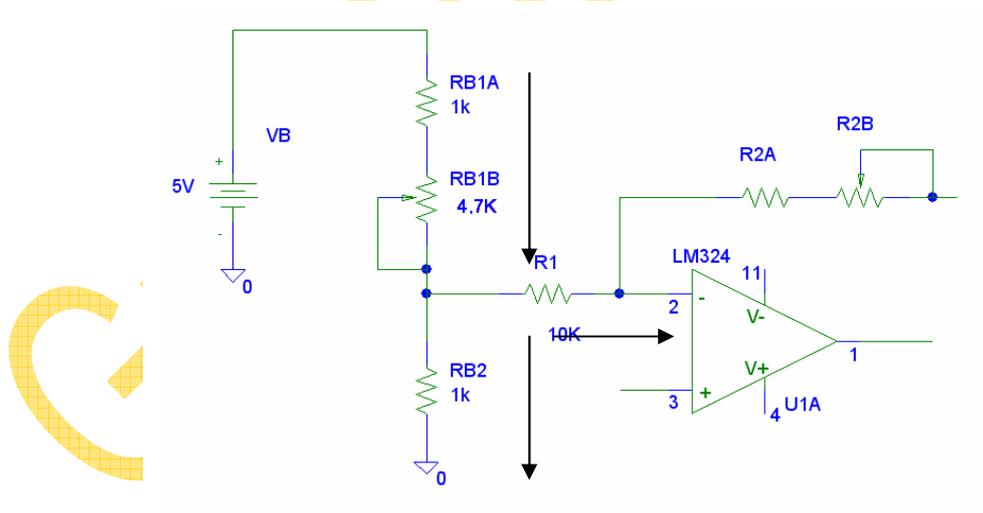
$V_{RB2} = \frac{V_{BATT}}{R_{TOT}} \times R_{B2}$  dove si può trascurare tranquillamente l'effetto del ramo con  $R_1$  !!

$V_{RB2} = \frac{V_{BATT}}{R_{B1} + R_{B2}} \times R_{B2} = \frac{5V}{R_{B1} + 1K\Omega} \times 1K\Omega = 0,75V$  in cui ho una sola incognita e quindi posso

risolvere l'equazione con una semplice formula inversa

$R_{B1} = \frac{V_{BATT} \times R_{B2}}{V_{RB2}} - R_{B2} = \frac{5 \times 1K}{0,75} - 1K = 5,67K\Omega$ , valore non esistente in commercio ma facilmente

realizzabile con un resistore fisso ed uno variabile in serie come si può vedere in figura.



Sono stato fortunato perché dovevo sottrarre questa tensione ed avevo a disposizione proprio l'ingresso invertente dell'Amp.Op.? No, in realtà ero a posto anche nell'altro caso.

Se avevo bisogno di sommare una tensione positiva per correggere l'offset in uscita (ad esempio uscita con tensione negativa da riportare a zero) potevo benissimo usare l'ingresso invertente per sottrarre una tensione negativa (batteria rovesciata) per ottenere l'effetto finale desiderato.

Dove posso trovare una tensione negativa in questo caso?

Semplicemente posso utilizzare la sezione con uscita negativa dell'alimentatore duale che è quasi sempre presente quando utilizzo degli Amp.Op.. Nel caso in cui non fosse disponibile una tensione negativa di precisione posso sempre aggiungere uno stadio amplificatore/invertente a guadagno unitario per ribaltare la tensione di offset. Sono costretto ad aggiungere uno stadio però i calcoli da effettuare sono sempre i soliti e se ho studiato la parte riguardante gli AmpOp non è un problema.

## Conversione Analogico/Digitale (C)

A questo punto possiamo fare alcune considerazioni sulla conversione analogico-digitale.

Il trattamento di un segnale è molto più semplice se esso si presenta in forma digitale: ad esempio la memorizzazione di una misura effettuata da un sensore è molto più semplice se tale misura la possiamo manipolare come una combinazione di bit → una cella di memoria contiene in realtà una combinazione di bit, un file su disco contiene una sequenza di byte (composti da 8 bit) → una serie di misure possono essere memorizzate su di un floppy disk, la trasmissione via radio / cavo / fibra ottica per grandi distanze è più conveniente farla in digitale che non in analogico → viene evitato il degradamento del segnale nel suo percorso, cosa quasi impossibile nelle modulazioni analogiche.

Con l'avvento di microprocessori sempre più veloci, inoltre, attualmente abbiamo una notevole facilità ad eseguire calcoli su questi segnali nel momento stesso in cui vengono "letti", si tratta della cosiddetta elaborazione in "tempo reale" → si possono cioè effettuare, ad esempio, operazioni con l'uso di somme, sottrazioni, moltiplicazioni, divisioni, integrali e derivate istante per istante, riuscendo in un certo senso a "prevedere" come evolverà un segnale nel tempo futuro analizzandolo in questo momento. Per esempio ci sono tecniche di calcolo sui segnali audio nei quali, in tempo reale, viene analizzato il segnale stesso e si riesce ad estrarre od eliminare il rumore di fondo in una conversazione, oppure togliere la voce di un cantante lasciando solo la musica di accompagnamento; queste prestazioni sarebbero state impensabili con l'elaborazione del segnale analogico.

Le cose da tenere a mente sulla conversione A/D sono essenzialmente ben poche.

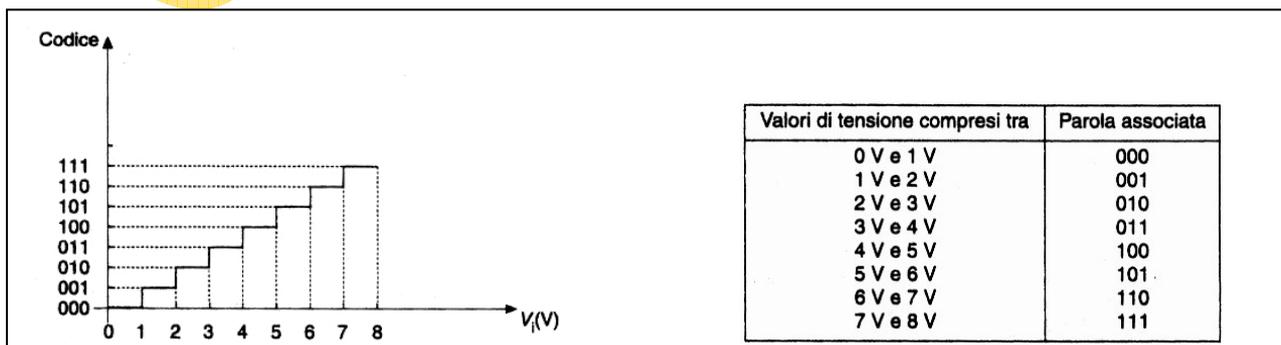
Ad esempio occorre conoscere il **range di ingresso** del convertitore (questo valore lo abbiamo già utilizzato per dimensionare il blocco precedente).

Occorre conoscere la **precisione** o **risoluzione** del convertitore, generalmente viene espressa in **bit** ovvero con quanti bit viene scomposto il segnale in uscita.

Ad esempio un convertitore ad 8 bit restituisce in uscita 8 segnali digitali (che vanno però sempre letti in blocco), corrispondenti a  $2^8$  diversi livelli della tensione di ingresso. Nel caso specifico, avendo un range di input di 5V ed una precisione di 8 bit, il nostro convertitore riuscirà a discriminare (a diversificare) fino a 256 diversi livelli di tensione, corrispondenti ognuno a:

$$\Delta V = \frac{V_{range}}{2^n} = \frac{5V}{2^8} = \frac{5V}{256} \cong 19mV$$

Il convertitore non può cioè riuscire a distinguere differenze inferiori a 19mV, questa è la sua "precisione". In genere questo viene rappresentato con un grafico a "gradinata" nel quale ad ogni



variazione di tensione in ingresso corrisponde una variazione del valore (numerico) presente in uscita.

In altre parole il convertitore “*non sente*” variazioni inferiori ad un gradino. Come posso fare quindi se ho bisogno di una precisione maggiore? Semplicemente scelgo convertitori con un numero maggiore di bit. Ad esempio, con il solito range di ingresso, ma utilizzando un convertitore con “parola” a 10 bit otterrò una precisione di:

$$\Delta V = \frac{V_{range}}{2^n} = \frac{5V}{2^{10}} = \frac{5V}{1024} < 5mV$$

*La tensione di ingresso viene scomposta in 1024 gradini !!!*

Si trovano in commercio convertitori a 10, 12, 14, 16 ed anche più bit. Un convertitore a 16 bit corrisponde alla possibilità di “*quantizzare*” (dividere in quanti, step o passi o gradini) fino a 65.536 diversi valori della tensione di ingresso, corrispondente nel nostro caso a “*sentire*” differenze molto inferiori al millivolt su 5V di fondo scala.

Per rendersi conto di cosa significa questo valore si può considerare che un buon strumento di misura dei nostri laboratori ha una precisione di circa lo 0,5% ovvero 5‰ sul fondo scala, ebbene un convertitore a 10 bit (non il massimo quindi) ha una precisione di circa 1‰ sul fondo scala!!

Nei normali lettori CD in commercio vengono utilizzati convertitori a 14 bit e, nel suono riprodotto, riusciamo a distinguere dai sussurri fino ai colpi sui tamburi, quindi livelli sonori diversissimi fra loro come intensità.

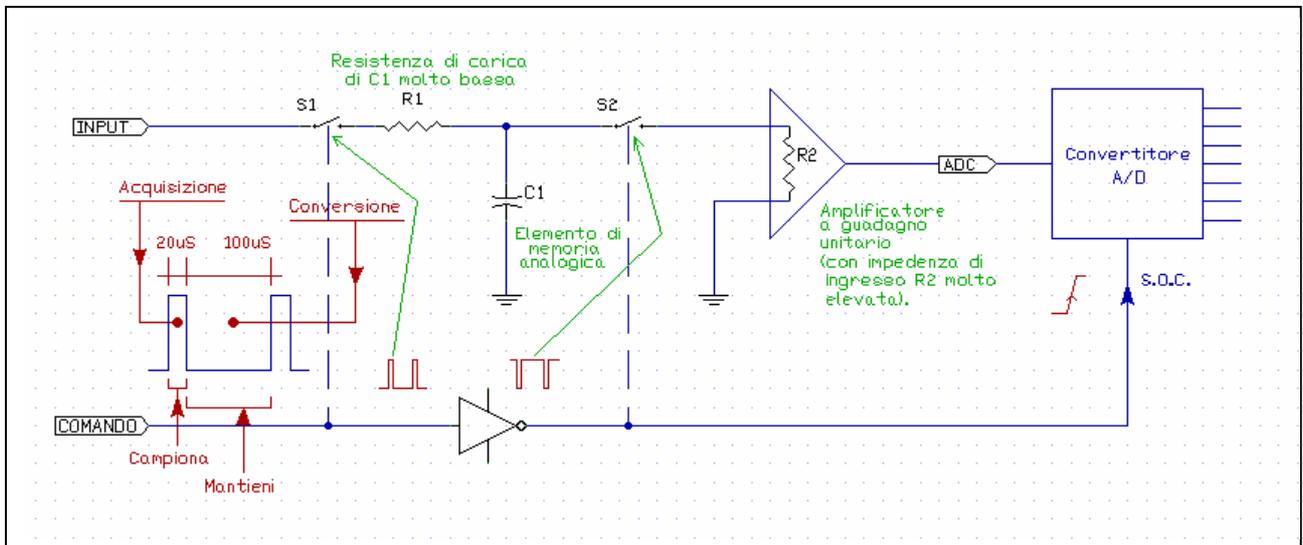
Un'altra cosa da valutare nei convertitori A/D è la cosiddetta **frequenza di campionamento**, cioè ogni quanto tempo viene “*letto*” l'ingresso, ovvero effettuata una conversione. Qui occorre ricordarsi il teorema di **Shannon**: “*la frequenza di campionamento di un segnale deve essere perlomeno il doppio rispetto al segnale che si sta campionando*”. Ad esempio, tornando al nostro tema di esame, misurando una temperatura di un ambiente (che ha una grande inerzia, ovvero reagisce lentamente alle variazioni) posso supporre che essa si modifichi molto lentamente, in tempi cioè dell'ordine dei minuti, in questo caso un campionamento ogni 60 secondi è più che sufficiente. Ma come? Si stava parlando di frequenza e siamo passati a calcoli sui tempi? Ricordiamoci che tempo e frequenza sono legati fra loro!! Ricordate come?

Se però devo misurare la temperatura di un liquido contenuto in un piccolo pentolino (bassa inerzia termica perché la massa in gioco è ben poca cosa) dovrò campionare in tempi dell'ordine del secondo. Ad esempio, tornando a parlare dei CD musicali (digitali), siccome si deve campionare un segnale audio e quindi nel range 20Hz...20KHz (risposta media dell'orecchio umano) dovrò utilizzare frequenze di campionamento di oltre 40KHz (il doppio della massima frequenza in ingresso), infatti lo standard per la registrazione dei CD musicali fissa questo valore a 44KHz.

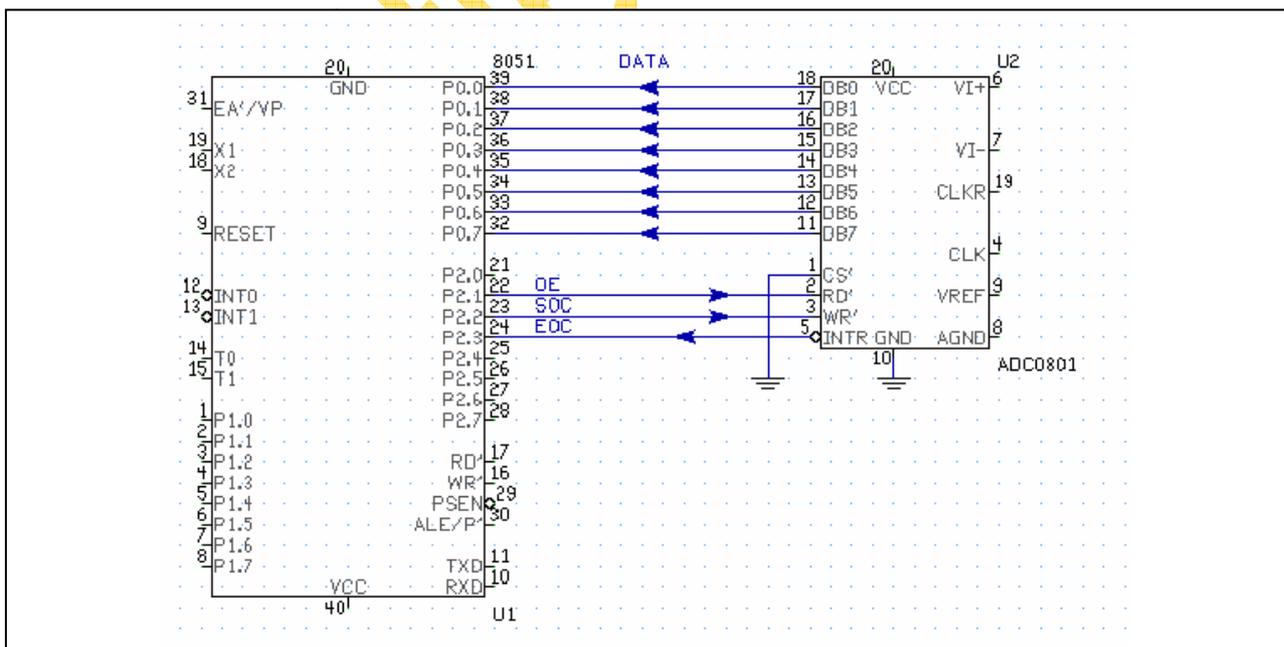
Un'altra cosa di cui si deve tenere conto è che, durante la fase di conversione A/D (che non è istantanea, ma dura a volte anche centinaia di μsec), il segnale di ingresso non si deve modificare, deve cioè rimanere “*fermo*”. Per la stessa ragione, se devo misurare il livello di una vasca contenente acqua, difficilmente riuscirò a farlo se nella vasca continua ad entrare acqua (o ad uscire). Siccome però non possiamo “*fermare*” il segnale di ingresso perché non dipende da noi si ricorre ad un artificio circuitale, così come potrei fare una foto istantanea alla vasca contenente acqua e poi misurare comodamente il livello del liquido sulla foto, così posso fare con un circuito elettronico di memoria analogica (nei casi più semplici basta una capacità, un condensatore), “*congelare*” il livello del segnale di ingresso, dopo di che il convertitore ha il tempo per effettuare la conversione senza che il segnale di ingresso si modifichi. In realtà esso nel frattempo si modifica ma questo sarà un problema risolvibile con altre soluzioni.

Nello schema esemplificativo si ha un segnale impulsivo che determina il flusso degli eventi, la loro successione temporale. L'impulso che rimane alto per 20 μsec fa chiudere l'interruttore elettronico S1 che, tramite una R molto bassa, carica la capacità C di memoria (la carica quindi in un tempo molto piccolo → costante tempo RC molto piccola). Il valore molto basso di R fa sì che la tensione ai capi di C riesca a seguire molto velocemente qualsiasi variazioni di ingresso, riesce cioè a

“*seguire*” ogni tipo di segnale in ingresso. Quando cessa l’impulso l’interruttore S1 si riapre, scollegando la capacità C dal segnale di ingresso. Nel frattempo si chiuderà invece l’interruttore S2, comandato dal segnale impulsivo di comando negato tramite una porta NOT. La capacità C ha cioè “*campionato*” (sample) il segnale di ingresso nella prima fase e adesso, nella seconda fase, lo “*mantiene*” (hold) poiché è collegata ad un AmpOp con impedenza di ingresso elevatissima, per cui non si scarica praticamente per niente nei 100  $\mu$ sec di durata della conversione A/D. Il medesimo segnale impulsivo fornisce anche lo start alla conversione vera e propria (segnale SOC).



L’ultimo passo della catena della conversione analogico/digitale è il trasferimento dei dati al  $\mu$ P. Attualmente svariati tipi di  $\mu$ P single-chip contengono già “*on-board*” (a bordo, cioè interno) il convertitore A/D per cui basta applicare ad un piedino di ingresso il segnale da convertire e tutto il resto avviene tramite istruzioni software che agiscono su registri interni appositamente progettati. L’intero procedimento però si riesce a comprendere meglio se utilizziamo uno schema a blocchi che si può riferire all’uso di un convertitore A/D esterno al  $\mu$ P.



La sequenza, guidata dal  $\mu\text{P}$ , inizia con la generazione del segnale **SOC** (start of conversion) che dà il via alla conversione (il sample&hold è automatico ed incorporato nella circuiteria dell'ADC). Quando l'ADC ha terminato la fase di conversione lo comunica al  $\mu\text{P}$  tramite il segnale di **EOC** (end of conversion), a questo punto il  $\mu\text{P}$  può leggere dal databus, in genere parallelo, il risultato (digitale) della conversione, a volte è presente un segnale **OE** (output enable) per far emettere all'ADC i segnali sul bus (altrimenti resta in stato di alta impedenza, cioè "isolato" dal bus).

Da questo momento in poi il segnale analogico che è stato convertito in un dato numerico (binario) è in un registro all'interno del  $\mu\text{P}$  e può essere "trattato" (elaborato) dal programma, in base alle istruzioni predeterminate dal programmatore.

Oggigiorno diversi modelli di  $\mu\text{P}$  hanno ADC a 8 o 10 bit "on-board", per precisioni maggiori, da 12 bit in su, vengono utilizzati chip esterni altrimenti il costo dei  $\mu\text{P}$  single-chip sarebbe troppo elevato. Nel caso di ADC esterni fino ad 8 bit le uscite spesso sono del tipo parallelo, ovvero 8 fili per trasferire il risultato della conversione. Siccome lo standard è normalmente con bus ad 8 bit, quando si utilizzano ADC a 10 o più bit la lettura del risultato deve essere effettuata in 2 passaggi (ad 8 bit per volta).

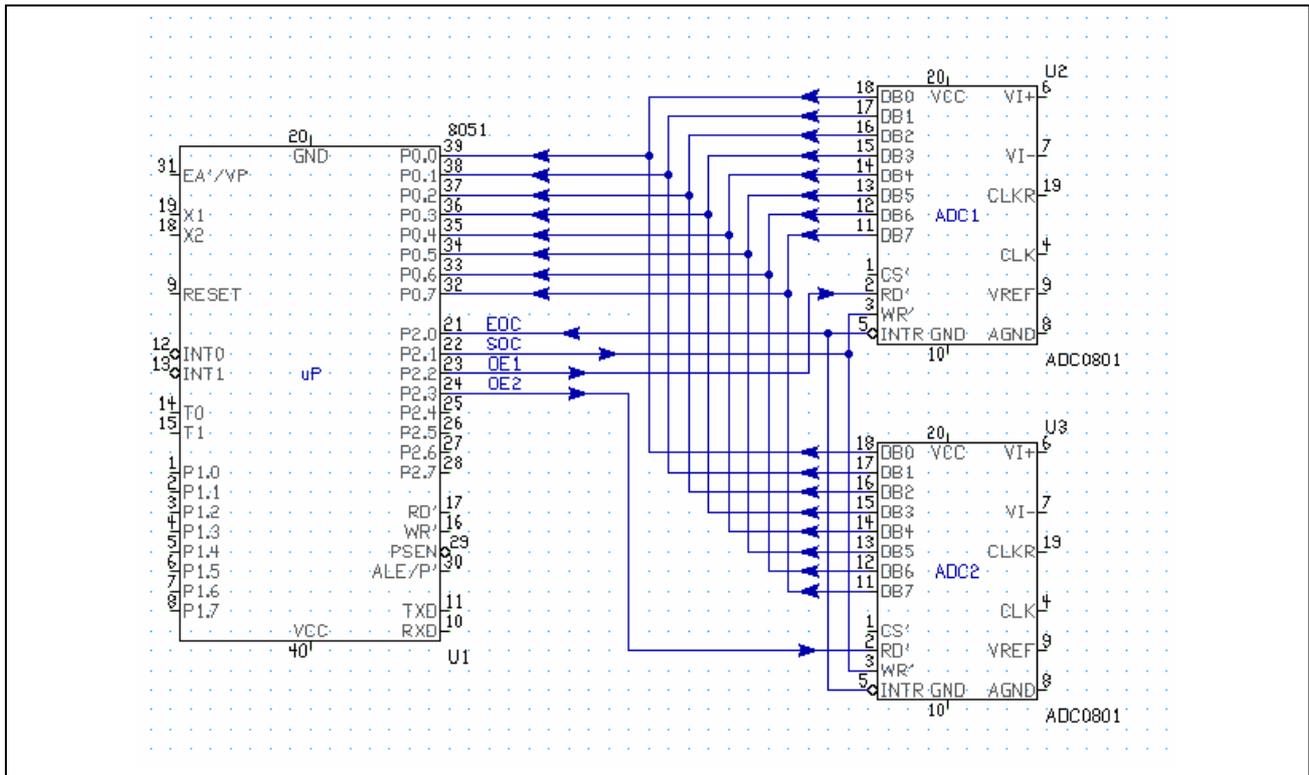
Utilizzare 8 piedini del  $\mu\text{P}$  è però troppo dispendioso (in realtà poi ne servono almeno 10-11  $\rightarrow$  8 bit di dati + SOC + EOC + OE !!), in un microprocessore a 20 piedini ne servirebbero oltre la metà solo per l'uso di un convertitore A/D.

Si utilizzano perciò, sempre più spesso, i convertitori A/D con linee di dati seriali: con solo 3 pin (clock, dati ingresso e dati uscita, oltre ovviamente al quarto filo che è la massa o ground o riferimento ma che è comunque già presente) si possono usare ADC a 14 bit !!

La comunicazione è di tipo seriale sincrona e con questo bus si inviano all'ADC i comandi (SOC) e si ricevono le letture, anzi, molto spesso nel solito chip ci sono anche più convertitori che, a rotazione, possono essere letti con i soliti 3 collegamenti. Si possono, inoltre, collegare più chip in parallelo sempre sul solito bus a 3 fili e, tramite codici, interrogare uno solo di questi chip e quindi esso sarà l'unico che mi risponderà sul bus con il risultato della conversione, mentre gli altri continueranno a "dormire" rimanendo in stato di quiete (stand-by).

Nel testo dell'esame (a cui ogni tanto dobbiamo tornare per ripensare il tutto, ricontrollare se stiamo andando "fuori traccia" o comunque se ci stiamo dimenticando di qualcosa) viene chiesto di controllare anche l'umidità dell'ambiente oltre alla sua temperatura. Servirà quindi un altro sensore (e questo lo avevamo già scritto all'inizio e, se non lo abbiamo fatto, sarebbe il momento di aggiungerlo all'inizio dello svolgimento, tanto stiamo ancora scrivendo la brutta copia). Aggiungendo un nuovo sensore occorre tutta la catena di condizionamento e, soprattutto, servirà un altro convertitore analogico-digitale. Dobbiamo quindi prevedere di raddoppiare gli 11 collegamenti che abbiamo già visto sono necessari per il pilotaggio di un convertitore. Studiando però meglio il problema possiamo sfruttare la cosiddetta tecnica a "bus". Questo perché non abbiamo la necessità di leggere i due convertitori in contemporanea, possiamo benissimo leggere prima un sensore e poi l'altro a distanza di tempo, tempo che poi si rivela brevissimo dell'ordine di 1 msec al massimo, trattandosi di microprocessori. Quando utilizzo un convertitore l'altro può essere disabilitato, in questo modo posso collegarli semplicemente in parallelo, abilitando ora l'uno ora l'altro tramite il segnale di **OE** (Output Enable). A volte tale segnale si chiama **CS** (Chip Select) oppure più semplicemente **RD** (Read) come nello schema illustrato qui di seguito. La barra dopo la sigla è da intendere come una soprilineatura (negazione booleana) che indica che il segnale stesso è attivo a livello basso.

Quando il convertitore non risulta abilitato tutte le sue uscite restano in stato di altissima impedenza, indicata in genere con **HiZ** o **three-state** o "terzo stato", quindi virtualmente scollegato (come se non ci fosse). Si preoccuperà il microprocessore di abilitare in sequenza, alternandoli, i due convertitori. Se per errore vengono abilitati entrambi i convertitori, avendo le loro uscite collegate in parallelo, molto probabilmente si avrà un cortocircuito con conseguente probabile distruzione del circuito integrato stesso.



Come si può vedere con questa soluzione invece di utilizzare 11 collegamenti (necessari ad un solo convertitore) moltiplicati per due e cioè un totale di 22 collegamenti, ne sono necessari solamente 12 ovvero **solo uno in più** rispetto all'uso di un solo convertitore!!!

Per chiarire meglio, se il  $\mu P$  lancia il comando di SOC questo comando giunge ad entrambi gli ADC poiché sono collegati in parallelo, però solo l'ADC attivato in quel momento (tramite il segnale OE1 od OE2) farà partire la conversione. Al termine della stessa il convertitore lancerà il segnale EOC verso il  $\mu P$  che saprà che giunge da quel convertitore e non dall'altro per il semplice fatto che il  $\mu P$  sa quale convertitore aveva attivato. L'altro convertitore, siccome non è abilitato, sarà in stato di altissima impedenza e "dormirà" senza rispondere ad alcun comando.

## Elaborazione Dati (D)

Una volta che il segnale analogico, convertito in un valore numerico (digitale), si trova in un registro all'interno del microprocessore, può venire finalmente elaborato tramite le istruzioni di un apposito programma software.

Sono possibili, ad esempio, calcoli per ottenere la media dei valori nel tempo (usando un "orologio" interno al  $\mu P$ , molto preciso perché legato al clock, quartzato, del  $\mu P$  stesso).

Possono essere ricavati i valori massimi e minimi nel tempo, le eventuali letture "sbaltate" dovute a disturbi possono essere scartate (pensate un po' a come potrebbe fare il  $\mu P$  per ottenere questa funzione...).

Il valore letto, confrontato con soglie predeterminate, può far attivare/disattivare delle uscite: pensiamo ad esempio ad un termostato.

Il microprocessore, cioè, grazie ad un programma può in sostanza effettuare delle "scelte" in base a criteri predefiniti.

Il valore letto ed elaborato può essere memorizzato per creare uno "storico" delle misure o per utilizzarlo per confronti successivi, ecc. ecc.

Concludendo, tramite l'uso di un microprocessore opportunamente "istruito" si può fare qualsiasi cosa!!

Il valore letto ed eventualmente elaborato può essere infine “*inviato*” a stazioni remote con vari tipi di comunicazione, principalmente basate su protocolli seriali di trasmissione dati (sempre per il notevole risparmio di connessioni elettriche), scelta comunque obbligata con la trasmissione su canali radio o in fibra ottica.

Ritornando al tema di esame proviamo a definire nelle linee generali “*cosa dovrà fare*” il programma contenuto nella EPROM del microprocessore. Non infognatevi nella scrittura del programma vero e proprio (istruzioni in linguaggio assembler, basic o C), vi mancherà il tempo ed è una cosa che i professionisti non fanno in qualche ora...

Disegniamo a grandi linee la sequenza di istruzioni necessarie a risolvere il nostro tema, ovvero le istruzioni necessarie a “*controllare*” l’intero sistema. Dovete cioè, secondo me, solamente dimostrare che avete le idee chiare su come va fatto funzionare il sistema, dare una idea generale di cosa dovrà fare il programma e quindi *far capire* al vostro esaminatore che queste cose le sapete fare e che, se ne aveste il tempo sufficiente (alcuni giorni), riuscireste anche a scrivere l’intero programma. In genere nel tema di esame c’è sempre una riga che dice qualcosa di simile a questo: *il candidato scriva, in un linguaggio a piacere, una parte del programma di controllo*. Questa parte, assolutamente generica, non ha molto senso se non quello di aiutare un candidato che conosce molto bene la programmazione e riuscirà perciò a scrivere alcune righe di codice. A mio avviso, però, è molto più esplicativo un diagramma di flusso, anche se abbastanza semplificato, dell’intero programma.

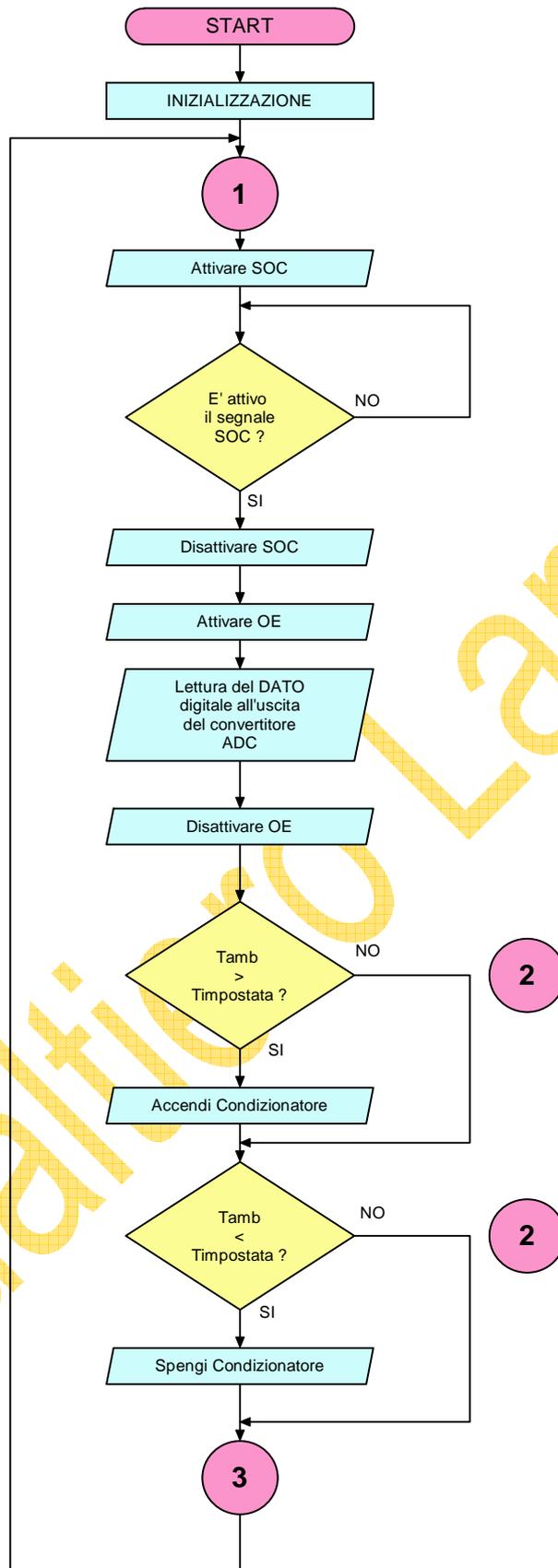
Aiutiamoci con uno schema a blocchi detto “*diagramma di flusso*” o “*flow-chart*”, utilizziamo i simboli standard e cerchiamo di essere più chiari possibile, in questa fase non occorre entrare nei più piccoli dettagli, basta sicuramente dare una chiara visione di insieme dell’intero flusso.

Un’idea potrebbe essere quella di farsi uno schema mentale dell’intera sequenza di passi e scriverla su carta come elenco e poi “*tradurla*” in simboli grafici:

1. Misurare la temperatura ambiente
  - a. Mandare il segnale **SOC** (start of conversion) al convertitore A/D.
  - b. Attendere il segnale **EOC** (end of conversion) che indica che il convertitore ha terminato il suo lavoro.
  - c. Leggere il dato binario (lettura) dal convertitore
2. Confrontare la temperatura ambiente con quella impostata
  - a. Se è minore tenere spento il condizionatore
  - b. Se è uguale non fare nulla
  - c. Se è maggiore tenere acceso il condizionatore
3. Ripetere tutto dal principio

Questa sequenza scritta, un po’ ampliata, può servire anche come descrizione del funzionamento del programma e con questa potete riempire almeno una pagina scritta !!!

Ho volutamente *dimenticato* di analizzare la parte riguardante il controllo di umidità dell’ambiente: ci torneremo sopra più avanti. Costruiamo il nostro muro mattoncino dopo mattoncino.



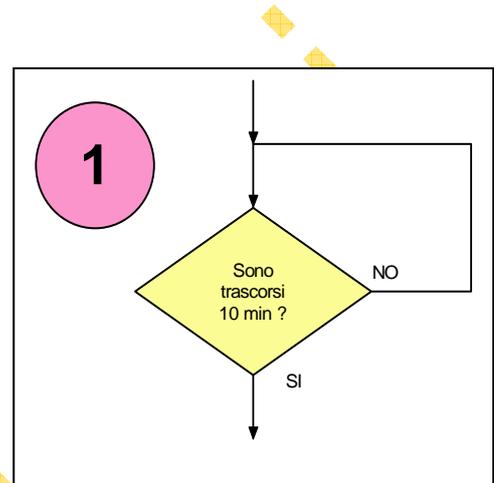
Un programma in questo modo a prima vista sembra funzionare e come soluzione “rapida” ad un tema di esame può andare più che bene, andando però a installare la scheda con il microprocessore pilotato da questo programma nel mondo “reale” ci accorgeremo di una cosa:

siccome la temperatura nell'ambiente varia molto lentamente, si tratta pur sempre di una massa di aria notevole come può essere quella contenuta in una stanza di albergo (sono vari metri cubi!!). Avrò perciò un momento in cui la temperatura "ballerà" per un certo periodo di tempo intorno alla temperatura di soglia ed avremo quindi accensioni/spengimenti continui del condizionatore nel giro di pochi secondi. Queste continue manovre di accensione/spengimento possono danneggiare l'apparecchio. Si tratta di un problema simile a quello di un circuito elettrico a soglia quando la tensione di ingresso si muove nell'intorno della soglia di intervento, vi ricordate con quale circuito elettronico si evitava questo problema? Vediamo qui di seguito.

Ci sono due possibili soluzioni per risolvere questo problema:

1. Ritardare con un tempo abbastanza lungo (ad esempio 10 minuti) le misure di temperatura con la relativa decisione di accensione oppure di spengimento dell'apparecchio.

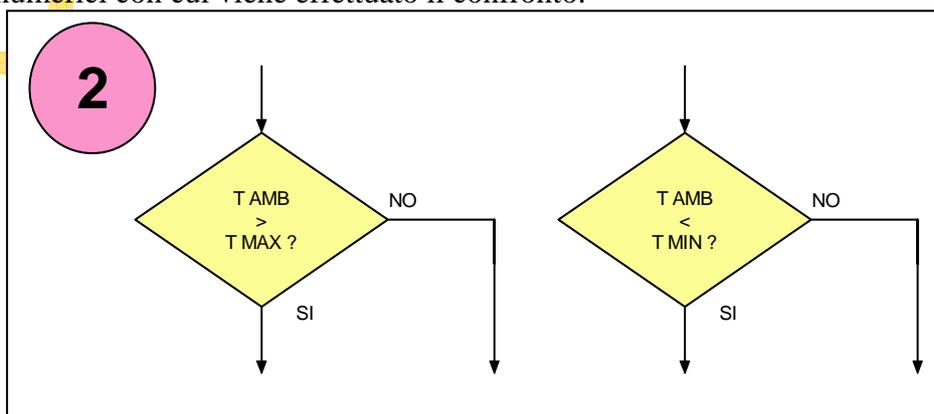
In questo modo tra una attivazione/disattivazione e la successiva passeranno almeno 10 minuti (sarà comunque sempre un tempo multiplo di 10 minuti). Questa ad esempio è la soluzione più utilizzata nei termostati ambiente di tipo elettronico che si installano nelle abitazioni. Se 10 minuti di tempo vi sembrano troppi provate a pensare a quanto tempo impiega il condizionatore a raffreddare una stanza o quanto tempo impiega la stessa stanza a riscaldarsi quando fuori è più caldo. Basta quindi inserire il controllo visualizzato a lato nel punto contrassegnato con ❶ sul diagramma di flusso.



2. Stabilire due soglie di intervento sufficientemente "distanti" tra loro. Un intervallo di 1°C è più che sufficiente, avremo cioè una "isteresi" tra i due interventi (accensione e spengimento). L'isteresi è l'intervallo di non-intervento. Chiamando  $T_{MIN}$  e  $T_{MAX}$  questi due valori all'intorno della temperatura impostata, il nuovo criterio di intervento che utilizzerà il microprocessore sarà:

- a. Se  $T_{AMB} > T_{MAX} \rightarrow$  accendi il condizionatore
- b. Se  $T_{AMB} < T_{MIN} \rightarrow$  spengi il condizionatore
- c. Se  $T_{MIN} < T_{AMB} < T_{MAX} \rightarrow$  non fare nulla, ovvero se il condizionatore era acceso rimane tale e se era spento lo stesso, cioè il condizionatore **NON** cambia lo stato in cui si trova.

Per questo tipo di soluzione basta modificare i due blocchi di controlli contrassegnati da ❷ nel diagramma di flusso. La modifica è minima perché si tratta solo di modificare i due valori numerici con cui viene effettuato il confronto.



Siete riusciti a ricordare quale era il circuito elettronico che realizza la stessa funzione, cioè un circuito a scatto (commutazione) con una certa isteresi che fa sì che la tensione di commutazione in ingresso non sia identica, a seconda della direzione della commutazione? Avete pensato anche voi al **Trigger di Schmitt**? Sì? Bravi, vuol dire che siete ben preparati e potrete rispondere a queste domande a *bruciapelo*!!

Concludendo si potrebbe anche dire che le due soluzioni proposte sono “*quasi uguali*” nel senso che, mentre la prima “*ritarda*” l’intervento del cambiamento di stato del condizionatore, la seconda soluzione si basa sulla temperatura ma, data l’inerzia con cui si modifica la temperatura della massa d’aria contenuta nell’ambiente, corrisponde comunque ad un “*ritardo*” di vari minuti tra una attivazione e una disattivazione.

Tornando nuovamente al tema di esame dobbiamo sottolineare a questo punto che il programma software illustrato fino ad ora gestisce solamente il controllo di temperatura dell’ambiente. In coda a questo controllo, cioè nel punto contrassegnato dal simbolo ③, dovremo mettere la sequenza di istruzioni necessarie al controllo di umidità dell’ambiente.

Il diagramma di flusso si può dire che sia identico a quello proposto per il controllo di temperatura, non ha senso ripetere le stesse cose (è noioso ed inutile), però potreste scrivere qualcosa del genere: “... *Il controllo di umidità dell’ambiente viene effettuato in maniera analoga a quella proposta per il controllo di temperatura, ovviamente in questo caso il segnale di ingresso di tutta la catena di conversione A/D e di controllo del software successivo sarà quello proveniente dal secondo sensore. Non ritengo necessario, per ragioni di tempo e di spazio, illustrare questa parte che è quasi identica alla precedente.*” In questo modo dimostrate di aver tenuto conto anche di questa parte di problema, fate bella figura e non annoiate l’insegnante che dovrà correggere il tema, dimostrate cioè di essere maturi. Però ricordatevi di scriverlo!! Scrivete sempre tutto, motivate tutte le vostre scelte scrivendone le ragioni.

## **Trasmissione Dati (E)**

Nel modo con cui ho sviluppato fino ad ora questo tema di esame e cioè: misure su sensori locali (cioè in ognuna delle otto stanze) e controllo locale (cioè su di una scheda con microprocessore che controlla un condizionatore/umidificatore presente in ogni stanza) in realtà non c’è bisogno di far convergere i dati ad una centralina di raccolta (e quindi sviluppare una parte riguardante la trasmissione dati) per effettuare poi in quel luogo il controllo totale di tutti gli ambienti.

Diversa quindi sarebbe stata la situazione se avessi scelto di utilizzare solo i sensori locali per le misure, trasmettendo poi tutti i dati ad una centralina che gestiva in un unico punto la climatizzazione degli ambienti, provvedendo quindi ad inviare alle varie stanze l’aria già climatizzata tramite condutture e tubazioni. In effetti nei grandi alberghi il sistema viene realizzato in questo modo perché permette di avere grandi risparmi nella realizzazione dell’impianto e, soprattutto, grandi risparmi nella manutenzione e nel consumo energetico durante l’esercizio.

Però non tutto è perduto, se proprio volete parlare della comunicazione dati (perché ad esempio siete molto preparati su questa parte e non volete assolutamente lasciarvi sfuggire l’occasione di aumentare di qualche punto la votazione finale) potrete farlo comunque con questa semplice premessa: “*la regolazione di temperatura ed umidità, nella soluzione da me proposta, avviene localmente, però la situazione delle stanze, in tempo reale, viene inviata ad una centralina di raccolta dati che provvederà alla supervisione di tutto l’albergo*”.

Nella centralina di raccolta un opportuno programma (magari realizzato con VisualBasic e quindi con interfaccia grafica stile Windows, molto gradevole e soprattutto chiara) si occuperà di visualizzare la situazione attuale in tempo reale (stato dei condizionatori: accessi / spenti ), temperatura ed umidità presente nelle varie stanze, evidenziando eventuali anomalie (allarmi di malfunzionamento).

Per realizzare questo da ogni stanza dovrò inviare le letture dei sensori e lo stato del condizionatore o dell'umidificatore alla centrale di raccolta (dovrò cioè inviare dei dati numerici ovvero digitali).

Scelgo la modalità di **trasmissione** dati **via radio** perché mi permette di non effettuare lavori murari durante l'installazione dell'impianto, evito cioè di fare tracce nei muri delle stanze o dei corridoi per incassare le tubazioni. Faccio la supposizione che l'albergo sia già esistente ed in attività e questa è una piccola ristrutturazione per modernizzarlo. In ogni caso queste premesse fatele e, soprattutto, scrivetele !!! Farete comunque una bella figura !!!

Per trasmettere dei dati via radio evitiamo di scegliere la modulazione di ampiezza (**AM**) o di frequenza (**FM**) che servono a trasmettere dati analogici (voce, musica, ecc.). Scegliamo ad esempio delle modulazioni digitali su portante analogica tipo la ASK o la FSK.

In realtà la modulazione **ASK** è una modulazione di ampiezza in cui la portante assume solo due valori, due livelli diversi, corrispondenti alla codifica degli stati binari 0 ed 1, mentre la frequenza della portante resta fissa. Infatti Amplitude Shifting Keying (ASK) significa "Codifica a spostamento o variazione di Ampiezza".

La **FSK** invece è una modulazione di frequenza in cui la portante può assumere solo due diversi valori di frequenza (mentre l'ampiezza rimane costante) corrispondenti anche qui alla codifica degli stati binari 0 od 1. La sigla FSK è l'acronimo di Frequency Shifting Keying ovvero "Codifica a spostamento o variazione di Frequenza".

Ricapitolando vi avevo detto di non usare la modulazione di frequenza o di ampiezza e poi in realtà ve la faccio usare !!! Questo serviva solo per chiarire la diversità fra le sigle AM/FM ed ASK/FSK.

Tra le due modulazioni (ASK ed FSK) la differenza è veramente minima: la ASK è circuitalmente più facile da realizzare ma è molto più sensibile ad eventuali disturbi elettromagnetici sempre presenti nell'etere; la FSK è circuitalmente più complessa da realizzare (più costosa) ma ha una più elevata immunità ai disturbi elettromagnetici e quindi sarebbe preferibile alla precedente.

Oggigiorno si trovano in commercio (a basso costo) dei moduli già montanti che realizzano la ricetrasmisione dati (**RX/TX**), dobbiamo solo collegare in ingresso il segnale digitale (seriale) da trasmettere e collegare l'antenna all'altro capo del modulo. Le tecniche utilizzate in questi moduli sono le più varie, tra di esse ci sono sia la ASK che la FSK, ma anche modulazioni più complesse tipo la **PSK** (modulazione di fase) e tecniche miste come la **QAM** (che combina ASK, FSK e PSK ottenendo velocità di trasmissioni elevatissime rimanendo comunque affidabile).

L'importante comunque è di introdurre questo argomento per poi poter riuscire a parlare, ad esempio, della ASK (su cui magari sono preparatissimo) oppure anche delle antenne, ecc. in questo tema scritto ma, soprattutto, per prepararmi il terreno per la successiva prova orale in cui potrò fare "agganci" con quello che ho illustrato in questo compito scritto. I collegamenti tra vari argomenti sono cose che fanno sempre una buona impressione sulla commissione che vi dovrà esaminare !!!!

A proposito, ho parlato di trasmissione dei dati ed ho dato per scontato che si tratta di una **trasmissione seriale** e non parallela dei dati. Eh sì, stiamo trasmettendo dei dati su di un canale radio, cioè su di un solo canale, come quello di un fiume, su di un canale passa un solo dato alla volta (è come se usassi un solo filo).

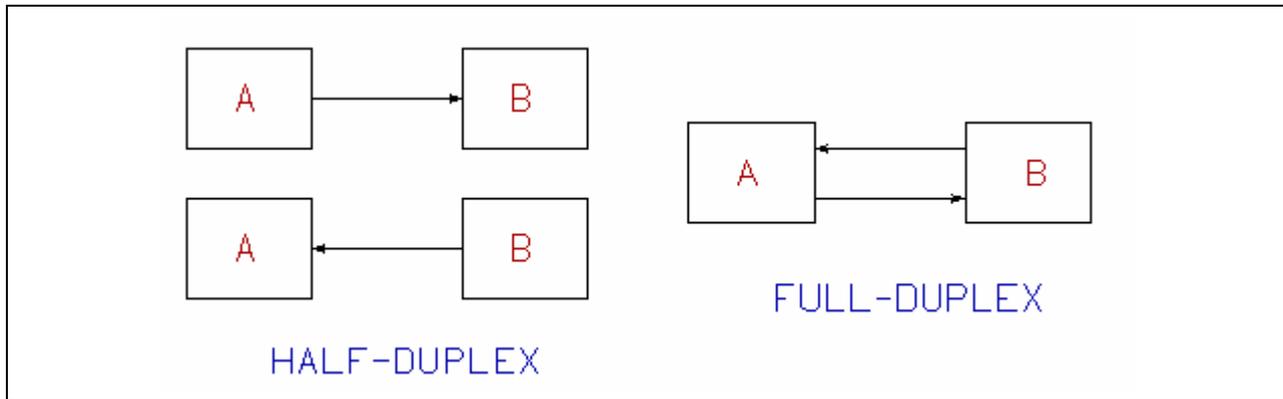
Su di un solo filo conduttore può passare un solo segnale alla volta; in realtà con un solo filo non posso fare nulla, ci vuole sempre una differenza di potenziale (d.d.p.), l'altro filo esiste sempre anche se è sottointeso, è il "riferimento" ovvero l'onnipresente "massa". Ma anche per i segnali radio di sono in realtà due conduttori, uno è il lato "caldo" (l'antenna e quindi l'etere), l'altro è il lato "freddo" (ovvero il suolo, il terreno, l'intero globo terrestre).

Se devo trasmettere dei dati su di un solo canale (radio o cablato che sia) li dovrò suddividere in una successione di bit (in genere 8) che però in ogni caso non posso "spedire" tutti insieme, li dovrò inviare uno dopo l'altro, cioè in "serie".

Anzi, siccome ho parlato di Tx/Rx, il medesimo canale radio lo posso usare in una sola direzione alla volta, come si vede in figura o trasmette A e riceve B oppure trasmette B e riceve A; ovvero A e B non possono trasmettere insieme o meglio lo possono anche fare ma nessuno dei due riceverà

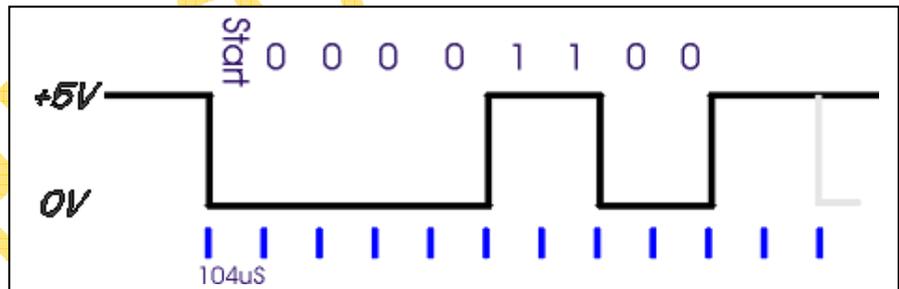
niente e quindi la trasmissione non ha senso. Si tratta cioè di una trasmissione *“half duplex”* come la classica telefonata: i due clienti che parlano tra loro non lo fanno insieme, ma a turno, altrimenti non capirebbero nulla della conversazione. Usando dei walkie-talkie (i radiotelefoni) si usa la parola *“passo”* per concludere la trasmissione e passare in ricezione, potrà così parlare l’altro interlocutore che, quando avrà finito *“passerà”* nuovamente la comunicazione all’altro.

Posso sfruttare quindi questo aggancio per parlare ad esempio di half-duplex e di full-duplex, come si vede le possibilità sono molto ampie, posso parlare e scrivere di ciò che voglio.



Dovendo comunicare tra due schede con dati seriali il primo problema che si presenta è quello della *“sincronizzazione”*: come fa a sapere il ricevente (che sta ricevendo un *flusso* continuo di bit uno dopo l’altro) dove è l’inizio e dove la fine di ogni gruppo di 8 bit? Occorre stabilire in anticipo una serie di regole, comuni e conosciute da entrambi i partecipanti, che definiscano questa ed altre cose ancora: questa serie di regole si chiama *“protocollo”*. Uno dei protocolli seriali più conosciuti e diffusi è il cosiddetto **RS-232**.

Brevemente: la linea di comunicazione normalmente è a livello alto (TTL → 5 volt) e ciò significa che la linea dati è libera. Il primo fronte di discesa rappresenta l’inizio dello *“start bit”* (che di conseguenza è sempre a livello basso o 0), dopo di



esso seguono 8 bit di dati partendo dal meno significativo ed infine avremo uno *“stop bit”* (sempre a livello alto o 1) che indica che la trasmissione del dato è terminata. Il prossimo fronte di discesa (che potrà giungere anche dopo un tempo lunghissimo) rappresenterà lo start di una nuova sequenza di bit. La larghezza di ogni bit è costante nel tempo ed il ricevente la deve conoscere a priori (ovvero prima di cominciare il suo lavoro). In realtà lo standard RS-232 prevedeva altri livelli di tensione ossia -12V per il valore 1 (alto) e +12V per il valore 0 (basso), il grafico quindi in realtà risulterebbe ribaltato, però stiamo usando il microprocessore che lavora con livelli TTL che rispettano uno standard leggermente diverso.

Nello standard RS-232, inoltre, sono comprese varie altre possibilità, ad esempio la possibilità di trasmettere i dati a gruppi di 7 bit (era usato dalle telescriventi ma è ormai in disuso), di inserire un bit di parità dopo i bit di dati (per migliorare il controllo sugli errori), di inserire due bit di stop per distanziare la trasmissione di un dato con il successivo (e dare modo al ricevente di memorizzare il dato appena ricevuto prima di apprestarsi a riceverne un altro), ecc. Tutte queste impostazioni devono essere conosciute in anticipo altrimenti il ricevente avrà difficoltà o impossibilità a ricevere correttamente i dati.

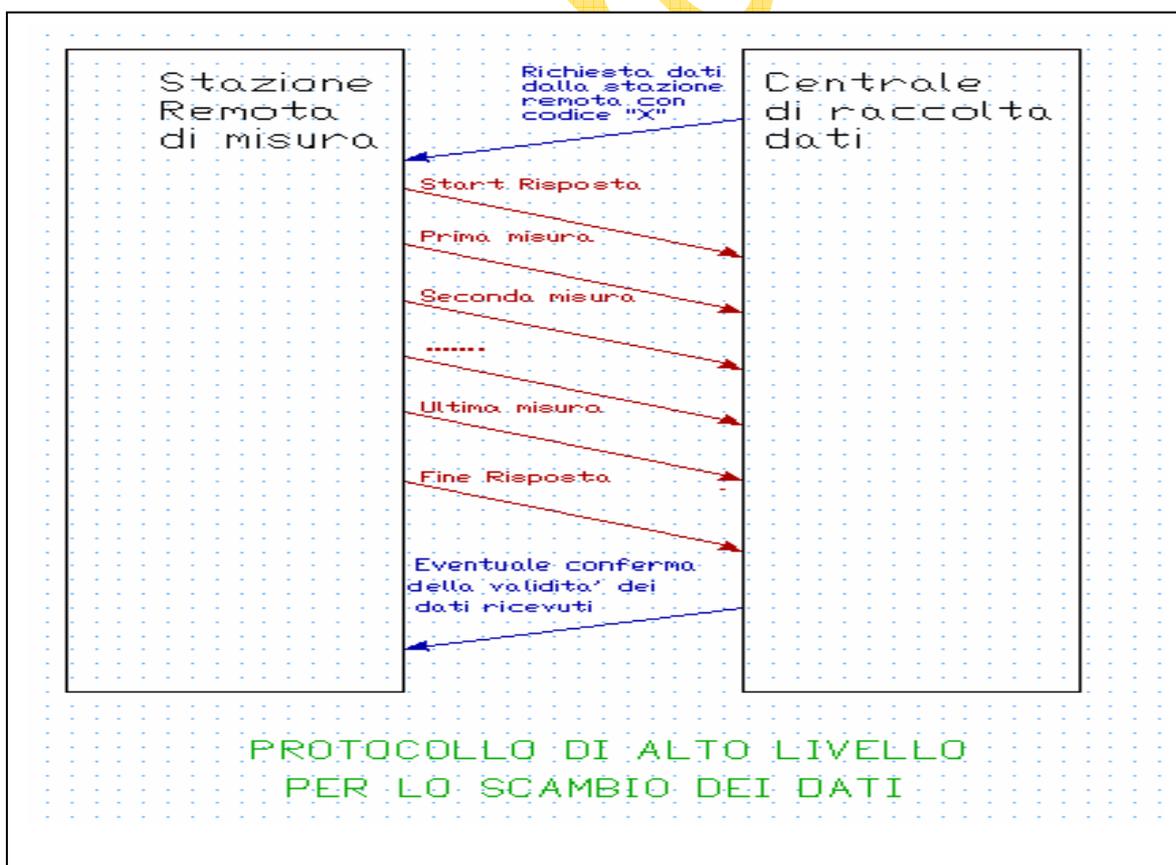
Le velocità di trasmissione dati spaziano dalle centinaia di **bit al secondo (bps)** fino alle centinaia di migliaia. In genere si parla di **baud** dicendo che è equivalente al **bps**, in realtà la velocità

espressa in baud si riferisce al “numero di simboli trasmessi in un secondo”, nel nostro caso si intende il carattere di 8 bit che per venir trasmesso ha però bisogno di 10 bit. In sostanza una velocità di 1.000 bps non corrisponde a 1.000 caratteri trasmessi in un secondo, sono circa il 20% in meno. La velocità più utilizzata è forse la 9.600 bps che indica che ogni secondo possono essere trasmessi 9.600 bit, cioè la larghezza di un bit è di  $1\text{sec}/9600 = 104\mu\text{sec}$ , da cui ne consegue che per trasmettere un byte di 8 bit (con lo start e lo stop sono in realtà 10 bit, a volte 11 se è attivato il bit di parità) occorre poco più di 1 msec. Nel nostro caso, dovendo trasmettere pochissimi dati, un paio di byte per le misure dei sensori e un altro, al massimo due byte, per lo stato del sistema, posso utilizzare anche velocità molto più basse, ad esempio 1.200 bps, riuscendo comunque a spedire tutte le misure in tempi molto inferiori al secondo.

Il protocollo utilizzato a “basso livello” è quindi l’ RS-232. Per basso livello si intende il livello più vicino all’hardware ovvero al circuito elettronico stesso. Questo protocollo però non prevede un efficiente controllo degli errori, prevede un rudimentale controllo di parità che evidenzia se c’è stato un errore sui bit ricevuti. Non esiste però il modo di ricostruire il segnale originale dato che non c’è modo di sapere quale degli otto bit è giunto errato, anzi questo controllo di parità nel caso di due bit errati nel solito byte può anche essere completamente annullato: il bit di parità mi può dire che il dato giunto è corretto, in realtà all’interno di esso ci possono essere due o più errori !!! In effetti il controllo di parità a questo livello non viene praticamente mai utilizzato.

Occorre utilizzare un protocollo ad “alto livello” che si può facilmente costruire anche per conto nostro. Si stabiliscono cioè alcune regole comuni tra stazione trasmittente e stazione ricevente e tutto quello che ne consegue e che dovrò scrivere un programma software per “implementare” questo protocollo ovvero per realizzarlo. (Implementare significa realizzare).

A livello di esempio, molto schematico (perché in fase di sviluppo del tema di esame, ed è sempre il caso di puntualizzarlo, devo toccare tutti i punti ma non posso per ragioni di tempo approfondire troppo il tutto, sei ore sono poche!!!), posso utilizzare un protocollo di questo tipo, come si può vedere in figura.



Passando quindi alla sua descrizione scritta, in cui si può ovviamente rimanere abbastanza generici. Scegliamo un protocollo bidirezionale

onale che permette quindi un completo controllo del flusso dei dati nei due sensi, scelgo di utilizzare solo caratteri alfabetici riuscendo in questo modo ad effettuare un semplice debug (test del funzionamento ed eventuali correzioni in maniera più semplice), per controllare il flusso dei dati basta il software HyperTerminal di Windows. In questo protocollo esiste un “*master*” che dirige tutto il traffico e questo sarà il microprocessore residente sulla centralina di raccolta dei dati e vari “*slave*” rappresentati dai processori residenti nelle schede di controllo delle varie stanze dell'albergo. Un master trasmette e tutti gli slave ricevono il comando, solo lo slave interpellato risponde alla chiamata, per fare questo ho bisogno di codificare ogni scheda remota, ad esempio ogni scheda è codificata con il numero della stanza dell'albergo. Quindi il flusso dei dati si potrebbe riassumere nei due punti della descrizione che segue.

1. La centrale di controllo invia nell'etere questa sequenza di caratteri: **S101?** che significa *interrogo la stanza n° 101*, nello specifico il carattere “S” può essere inteso come start della trasmissione (tutte le schede sono in ricezione e a questo punto si “*risvegliano*”), seguono sempre 3 caratteri numerici che indicano il numero della stanza interpellata (tutte le schede “*leggono*” questi 3 caratteri ed a questo punto rimane “*sveglia*” solo quella interpellata ovvero solo quella il cui codice interno coincide con quello ricevuto dall'etere, tutte le altre schede si “*riaddormentano*”). Il carattere “punto interrogativo” indica la fine della trasmissione e solo dopo aver ricevuto questo carattere la scheda interpellata (cioè l'unica rimasta “*sveglia*” risponderà). Questo protocollo mi garantisce che se cerco di interpellare una scheda “*inesistente*” questa ovviamente non risponderà perché non esiste, ma neanche le altre risponderanno perché il codice non coincide. In caso di errori dovuti a disturbi elettromagnetici nessuna scheda mi risponderà, ma anche se mi dovesse rispondere una scheda... vedremo dopo cosa succede. Terminata la trasmissione il master stesso si pone in “*ascolto*” della risposta.
2. L'unica scheda il cui codice coincide con quello trasmesso dalla centrale di controllo risponderà ad esempio con una serie di caratteri in questo modo: **R101,T=124,U=203,C=0,H=1\*** che significa: risponde la scheda 101 (R101), la virgola separa il dato seguente che è la misura di temperatura T=124 indica che il sensore di temperatura rileva un valore di 124 step su 255 di fondo scala (stiamo lavorando con ADC ad 8 bit), la misura di umidità U=203 è di 203 step, C=0 indica che il condizionatore è spento ed H=1 che l'umidificatore è acceso, mentre il carattere “asterisco” indica la fine della sequenza di dati. Da notare che ho utilizzato lettere (caratteri) diverse dal punto 1, questo perché quando risponde la scheda interpellata oltre alla centrale di raccolta sono comunque in perenne ascolto anche le “*altre*” schede che non devono venire “*risvegliate*” ad esempio dall'uso del carattere “S” altrimenti si avrebbero malfunzionamenti nel sistema.
3. Se, entro un tempo prestabilito, ad esempio 10 secondi, nessuna scheda risponde il master provvede a segnalare l'anomalia e riprova, ripartendo dal punto 1, con la scheda di un'altra stanza.

Un protocollo di questo tipo è molto semplice, non ha correzione di errori, questo lo dobbiamo scrivere, magari dicendo che avendo a disposizione un tempo maggiore avreste potuto implementare anche questa funzione, però semplici errori vengono rilevati ed automaticamente scartati.

Ad esempio se ricevo una sequenza in cui invece di T=124 ricevevo F=124 perché al posto di T, a causa di un disturbo, viene ricevuta la lettera F, posso automaticamente scartare tutta la sequenza, tanto a breve distanza di tempo (si parla di secondi) posso richiedere nuovamente la lettura dei sensori e, statisticamente, sarà poco probabile un nuovo errore. Eventuali errori sul numero ricevuto sono di più difficile rilevazione, però sapendo che numeri superiori a 255 sono errati ne elimino tanti di errori. Inoltre in fase di ricezione posso fare un controllo di questo tipo: siccome a distanza di alcuni secondi la temperatura ed anche l'umidità non possono essere molto diverse (abbiamo una

notevole inerzia nel sistema!!!) posso scartare le misure che divergono di molto dalla media che via via accumulo nel tempo, ecc. ecc.

Questo protocollo mi garantisce anche su un errore dovuto a disturbi elettromagnetici sulla richiesta iniziale di lettura. Ad esempio se la centrale trasmette S101? per interrogare la scheda della stanza 101 e per errore la S viene trasformata in una Q nessuna scheda si “risveglierà” perché la lettera Q non è il codice di start della trasmissione. Se invece viene corrotto il numero di stanza e risulta il numero di una stanza inesistente nessuna scheda mi risponderà, se invece il numero corrotto corrisponde davvero al codice di un'altra scheda mi risponderà quest'ultima, ovvero interrogo la stanza 101 ma, per un errore avvenuto mi risponde la stanza 102, poco male perché la centrale riceve le misure della stanza 102 e le può comunque utilizzare, tanto nel giro di pochi secondi interrogherà nuovamente la stanza 101 e difficilmente si ripeterà lo stesso errore.

## Conclusioni aggiuntive

Con questo possiamo dire di aver concluso in maniera sufficientemente ampia la trattazione del sistema di misura, controllo e regolazione richiesto dal tema di esame. Rimangono fuori alcune considerazioni che farò qui di seguito.

- Il sensore di umidità indicato nel testo dell'esercizio ha una uscita in corrente: come faccio a progettare il circuito di condizionamento?
- Il sensore proposto (non si riferisce a questo testo di esame) ha una risposta in variazione di resistenza.
- Il sensore proposto ha una risposta in variazione di capacità, ecc.

In realtà non è un grandissimo problema: la catena a partire dal circuito di condizionamento, convertitore A/D in poi rimane valida, basta solo convertire la grandezza elettrica proposta in una variazione di tensione, quindi la posso applicare all'ingresso del circuito di condizionamento, avendo le stesse regole di progettazione e cioè **range d'ingresso** → **range d'uscita** e da qui si parte con i calcoli della progettazione dei vari circuiti.

Nelle pagine seguenti farò quindi un riepilogo dei principali circuiti che realizzano queste funzioni, ricordate però che se qualcosa non sapete (e questo è sempre molto probabile anche se siete preparatissimi) potrete sempre cercarlo sul manuale che è sempre ammesso in sede di esame.