

Istituto professionale "MORETTO"

TESINA DI MATURITA'

# Gestione automatizzata di una serra

FIENI STEFANO - PORRINI EMANUELE 5BZ

ANNO SCOLASTICO: 2003/2004

## Sommario:

1. Introduzione.
2. Schema a blocchi.
3. Descrizione del sensore di temperatura.
4. Descrizione del sensore di umidità.
5. Descrizione del sensore di luminosità.
6. Descrizione del sensore di livello.
7. Caratteristiche generali sui convertitori.
8. ADC0808.
9. Porta parallela.
10. Software ( Delphi )

## **Introduzione**

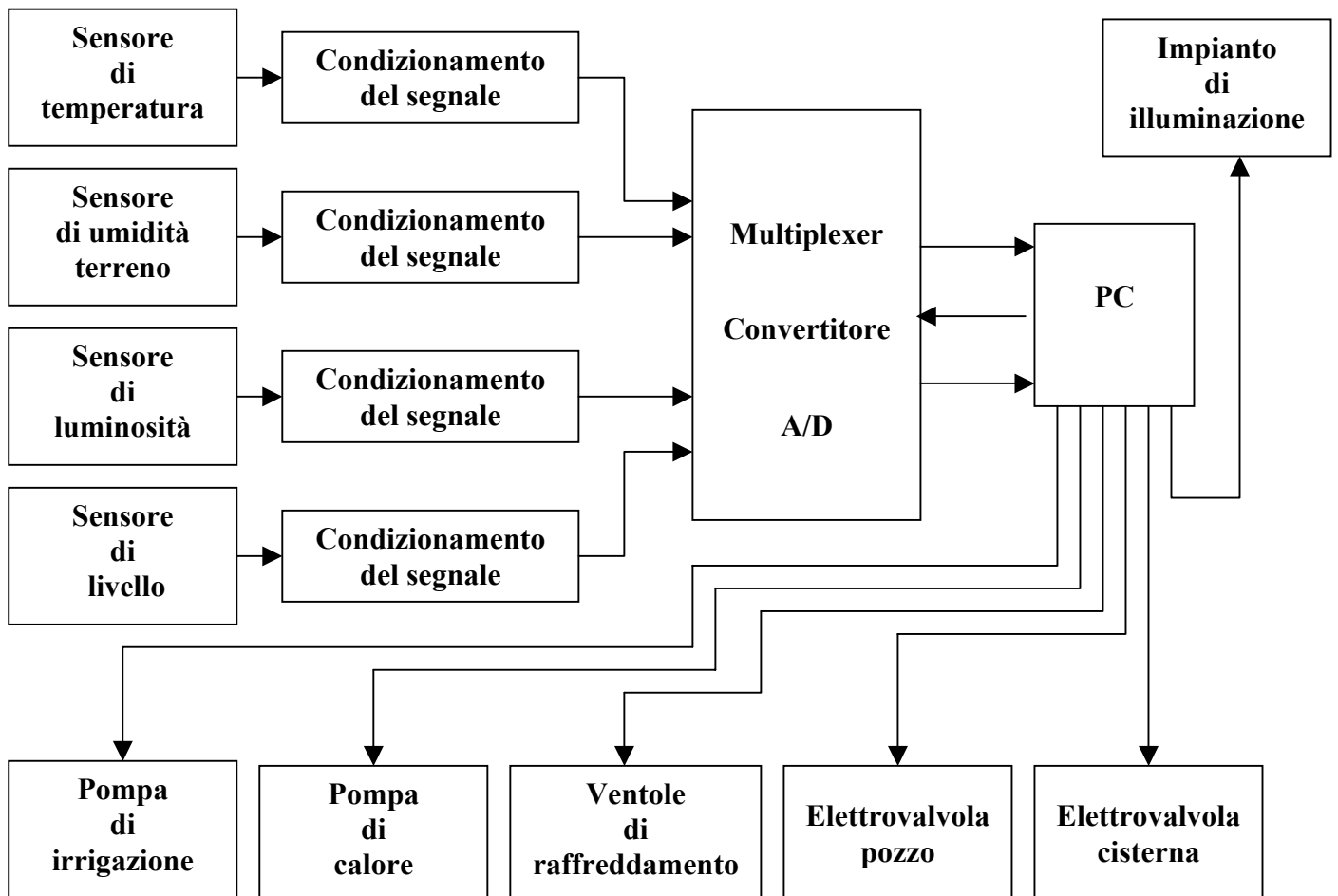
Il progetto che vi presentiamo prevede la gestione tecnologica di una serra che ha permesso l'intensificazione delle produzioni e una maggiore qualità delle colture. Questo metodo di produzione ha riscontrato particolarmente interesse da parte dei lavoratori nel campo agricolo es:(orticoltori, floricoltori ecc ecc...).

Una serra è una struttura concepita per essere trasparente alle radiazioni solari senza però farle disperdere.

I parametri fisici all'interno di essa dipendono dalle caratteristiche strutturali e dalla posizione geografica. Quindi per gestire questi differenti parametri fisici si ricorre a sistemi elettronici attraverso un insieme di trasduttori.

I trasduttori rilevano i diversi parametri all'interno della serra convertendoli in una grandezza elettrica. Per rendere i parametri fisici interni alla serra idonei alla scelta dell'utilizzatore i segnali forniti dai trasduttori vengono convertiti in segnali idonei al trasferimento verso il Pc che a sua volta attraverso un programma pianificato provvede a gestire i vari attuatori (Pompa irrigazione, riscaldamento, luci ecc ecc).

## Schema a blocchi



## **Descrizione dello schema a blocchi**

Questo è lo schema a blocchi fondamentale che descrive il funzionamento dell'intero sistema, riguardante il controllo e il mantenimento dei parametri fisici interni alla nostra serra.

I primi quattro blocchi descrivono complessivamente quattro trasduttori:

- Temperatura;
- Umidità;
- Luminosità;
- Livello;

Questi dispositivi sono in grado di convertire una grandezza fisica in una grandezza elettrica analogica. I diversi circuiti di condizionamento che li seguono adattano il segnale ad una scala idonea alla conversione.

I segnali così condizionati vengono inviati al multiplexer ( Incorporato all'ADC0808 ).

I diversi segnali vengono selezionati uno per volta dal multiplexer attraverso i data select per essere così inviati al convertitore a 8bit di risoluzione. Il convertitore è gestito dalla porta parallela, la quale invia i segnali di inizio e fine conversione.

Una volta terminata la conversione tramite un programma il Pc provvede ad modificare i parametri della serra dove necessita attraverso l'attivazione o no degli attuatori.

## Descrizione dei sensori

### Sensore di temperatura LM35

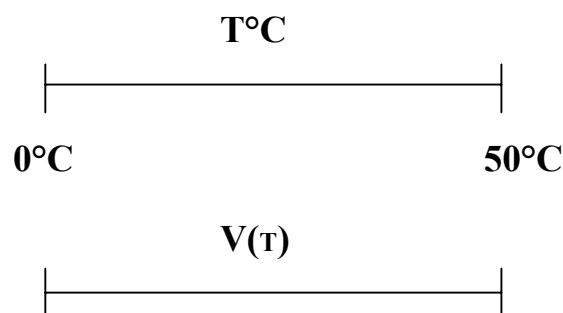
Il controllo della temperatura è un elemento fondamentale per la gestione di una serra: se la temperatura si abbassa troppo, la condensazione sulle foglie delle colture può favorire il propagarsi di funghi e malattie che portano a una non corretta vegetazione.

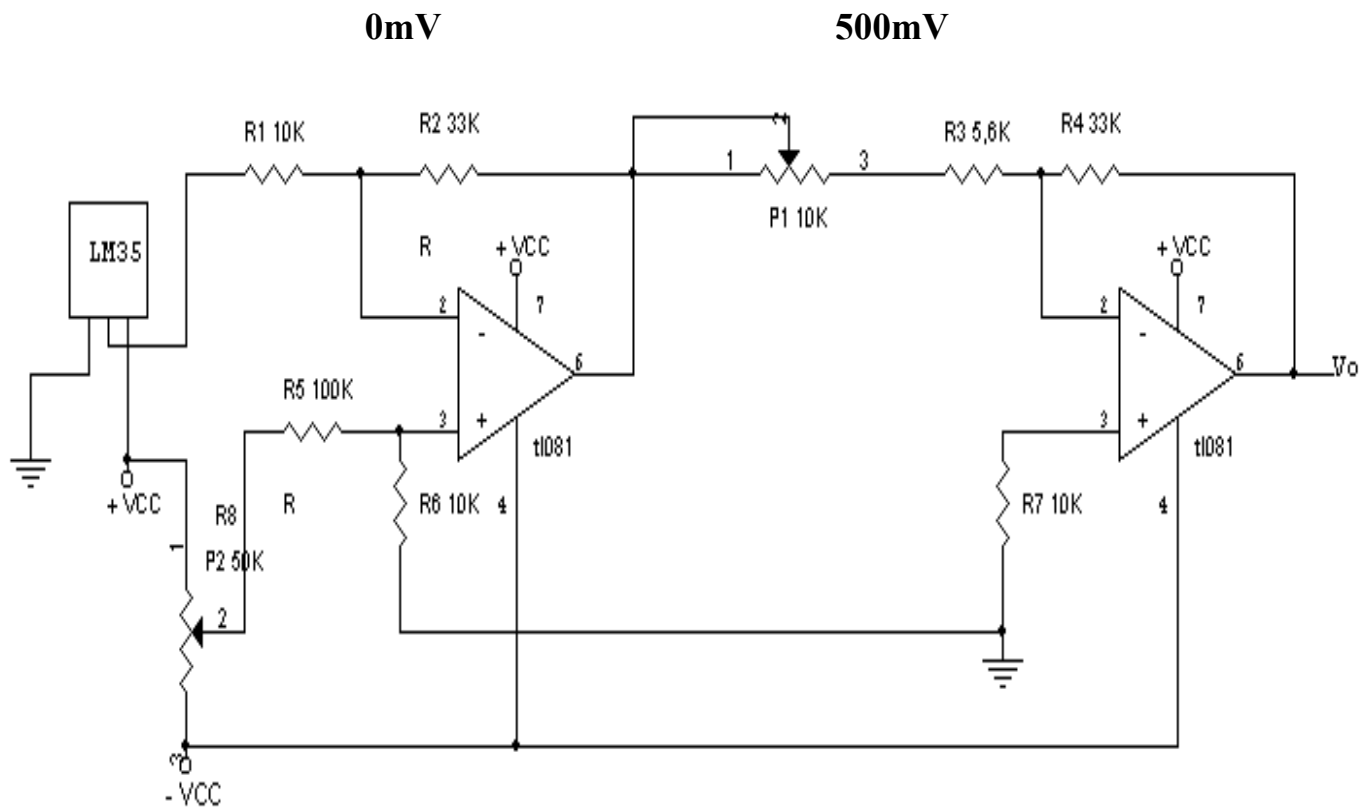
Invece il troppo calore può inibire il processo di traspirazione e quindi provocare un rapido degradamento dell'intera coltura. Quindi con un ottimo sistema di riscaldamento e o raffreddamento metterà in ottime condizioni il clima della serra.

Il sensore utilizzato per il rilevamento della temperatura è LM35 che è un trasduttore che fa passare una tensione in base alla temperatura. Il sensore a 0°C ci da una tensione pari a 0 V, invece quando la temperatura è pari a 50°C il sensore lascia passare una tensione di 500mV. Alla variazione di 1°C all'uscita del dispositivo abbiamo una variazione di 10mV. Per ottenere il segnale su una scala è stato dimensionato in modo da avere 0V per 0°C e 5V per 50°C.

$$V(T) = K \cdot T^{\circ C} \quad K = 10mV/^{\circ C}$$

**T°C =Variazione di temperatura (0°C – 50°C) in gradi centigradi.**





### Spiegazione del circuito

Sapendo che la sonda di temperatura LM35 fornisce 10 mV/°C avremo sull'ingresso dell'amplificatore una tensione di 0V per 0 °C e 500mV per 50 °C.

L'amplificazione totale dovrà essere di 10 di cui 3,3 V dovuto al primo stadio di amplificazione e il rimanente (Circa 3 volte) dovuto al secondo stadio di amplificazione regolabile tramite T1 Da 10K. L'amplificazione del primo stadio è uguale:

$$A1 = -R2 / R1 = 33K / 10K = -3,3$$

L'amplificazione del secondo stadio è uguale:

$A_2 = -R_4 / ( R_3 + T_1 ) = 33K / ( 5,6K + T_1 ) = -3,03$  pertanto T1 dovrà essere regolato affinché valga circa 5,29K mentre T2 dovrà essere regolato al fine di avere il rapporto tra tensione e temperatura rilevata.

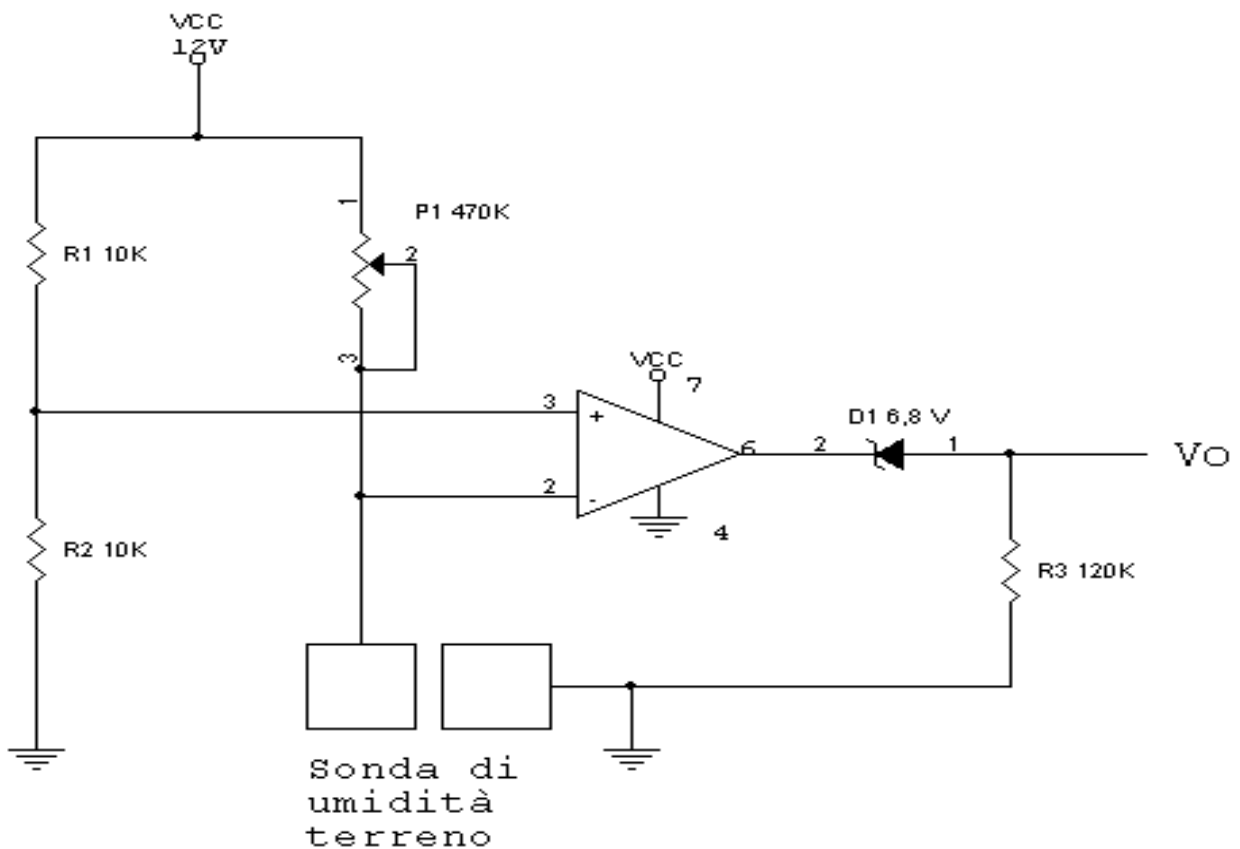
$$A_{tot} = A_1 \times A_2 = -3,3 \times -3,03 = +10$$

### **Sensore di umidità del terreno.**

L'umidità del terreno è un parametro fisico da non dimenticare perché attraverso l'acqua che una coltura si sviluppa. Se si ha un'eccessiva umidità "cioè ristagno d'acqua" si rischia la formazione di marciume che danneggerebbero la coltura. Invece se l'acqua viene a scarseggiare per un periodo di tempo, la coltura rischia la degradazione. Per rilevare e gestire questo parametro si ricorre ad un sensore realizzato attraverso due barrette metalliche poste l'una accanto all'altra ma isolate tra esse. Al variare dell'umidità del terreno varia la resistenza tra le due barrette metalliche:

- Se la terra è secca tra le due barrette e come se avessimo un circuito aperto e la resistenza fornita è infinita;
- Mentre quando la terra è bagnata tra le due barrette e come se avessimo un corto circuito determinato dalla conducibilità dell'acqua.

Queste due condizioni ci permettono di avere in uscita al circuito di condizionamento una tensione che va da 0 a 5 V.



## Spiegazione del circuito

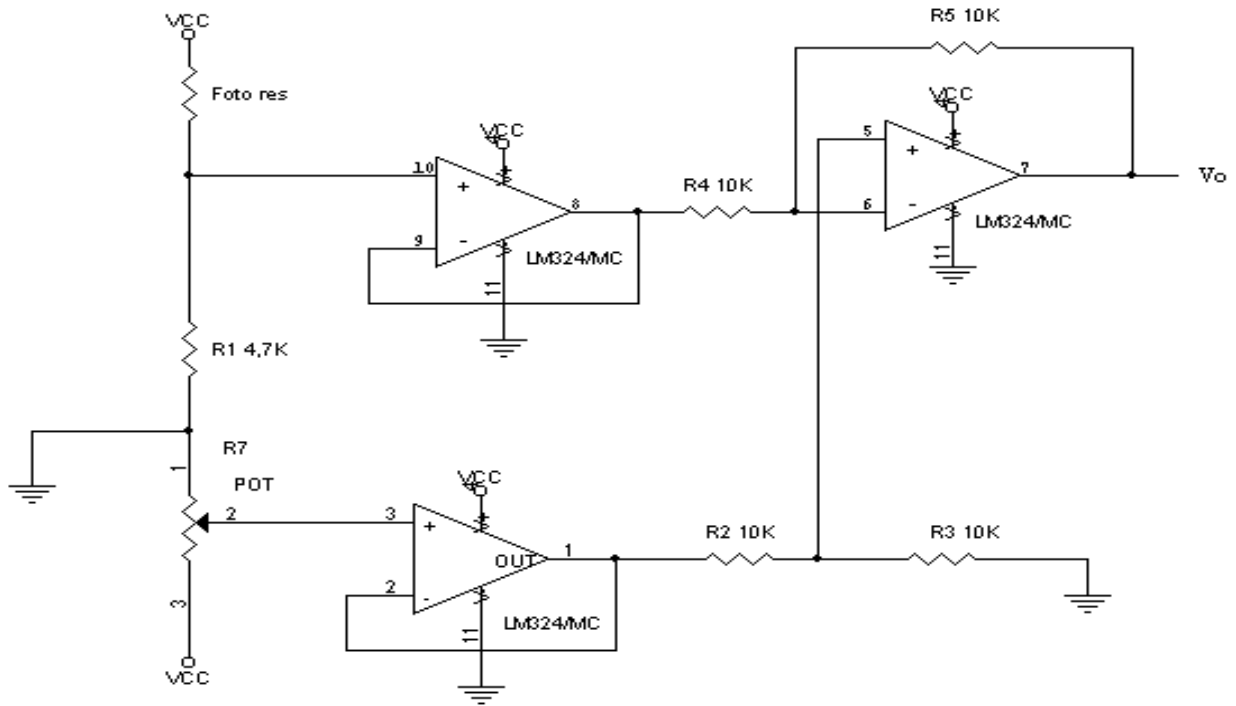
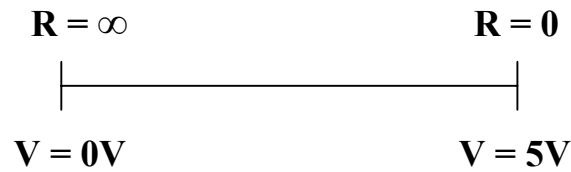
numero 3(non invertente) abbiamo i 6V dati dal partitore di tensione formato da due resistenze da 10K,che è la tensione di riferimento del comparatore; Sul piedino mentre sul piedino numero 2 (invertente) il segnale dipende alla variazione della sonda di umidità:

- Se non si ha presenza di umidità la sonda ci da un valore di tensione maggiore rispetto al segnale di riferimento così l'uscita del comparatore si porta a 0V;
- Se invece si ha la presenza di umidità la sonda ci da un valore di tensione di 0V così da essere minore rispetto alla tensione di riferimento e così all'uscita abbiamo 12V i quali attraverso un diodo zener da 6.8V vengono ridotti ad una grandezza di 5V in uscita al circuito di condizionamento.

## **Sensore di luminosità**

Poiché le condizioni ambientali regolano lo sviluppo della coltura, un'illuminazione appropriata è importante perché è attraverso essa che si ha lo sviluppo. In assenza di luce la pianta non compie il processo della fotosintesi e quindi non si sviluppa. Illuminando l'ambiente quando la luce è assente o scarseggia ci consente di ottenere il raccolto in tempi minori sfruttando tutte le ventiquattro ore a nostra disposizione. Per avere un controllo della luminosità in tutte le ore del giorno abbiamo utilizzato una fotoresistenza.

La fotoresistenza è un dispositivo che sfrutta la proprietà dei semiconduttori sensibili alle grandezze luminose. La resistenza diminuisce all'aumentare della luminosità e aumenta al diminuire della luminosità.



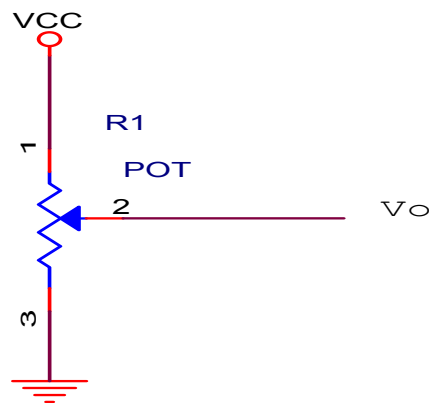
## Spiegazione del circuito

Attraverso il potenziometro fissiamo la tensione di riferimento che viene portata prima su un inseguitore e subito dopo sul piedino non invertente di un amplificatore invertente con guadagno pari a uno; Sul piedino invertente troviamo il segnale proveniente dalla fotoresistenza:

- Quando la fotoresistenza assume un valore elevato di resistenza sul piedino invertente abbiamo zero e quindi in uscita abbiamo 5V.
- Quando la fotoresistenza assume un valore basso di resistenza sul piedino invertente abbiamo una tensione quasi uguale a quella di riferimento di conseguenza l'uscita va a zero.

## Sensore di livello

Abbiamo ipotizzato di sfruttare la superficie del tetto della serra al fine di raccogliere l'acqua piovana all'interno di una cisterna. Per tenere sotto controllo il livello dell'acqua della cisterna abbiamo utilizzato un potenziometro "che potrebbe essere applicato su un galleggiante". Abbiamo sviluppato questo sistema per risparmiare acqua che in questi ultimi anni è diventata una risorsa molto preziosa, quindi quando la serra necessiterà di irrigazione il sistema provvederà a controllare se l'acqua nella cisterna è sufficiente per l'irrigazione e nel caso contrario sfrutterà l'acqua del pozzo.



## **Spiegazione del circuito**

Quando la resistenza è minima l'uscita  $V_o$  assume valore massimo  $V_{cc}$  mentre quando la resistenza è massima l'uscita  $V_o$  assume valore zero.

## **Convertitore Analogico – Digitale (ADC0808)**

Esistono apparecchiature elettroniche che forniscono unicamente segnali di tipo analogico: tra queste, ad esempio, molti trasduttori, che trasformano grandezze fisiche in elettriche e, nella maggior parte dei casi, il segnale prodotto è analogico, ovvero subisce variazioni proporzionali alla grandezza fisica applicata al suo ingresso.

Poiché la manipolazione dei segnali può avvenire attraverso circuiti digitali, è evidente la necessità di convertire la grandezza analogica in digitale mediante i circuiti definiti convertitori analogici digitali.

Viceversa il risultato dell'elaborazione di una rete logica deve comandare attuatori di tipo analogico. Dato che per attuatore si intende un sistema in grado di trasformare un segnale elettrico in una grandezza fisica proporzionale al segnale, è necessario che il risultato dell'elaborazione digitale sia convertito in analogico, per pilotare un attuatore.

Un sistema in grado di trasformare un risultato digitale in un segnale analogico viene definito convertitore digitale analogico.

Nella conversione digitale analogica e analogica digitale si incontra un problema fondamentale: una grandezza analogica è costituita da un insieme continuo di valori,

una grandezza digitale, per la sua natura binaria, e invece costituita da un insieme finito di valori possibili.

Questo condiziona evidentemente la precisione della conversione.

Per aumentare la sensibilità e la precisione si ricorre alla quantizzazione e al campionamento.

I convertitori analogico-digitali trasformano un livello di tensione in un codice binario ad esso corrispondente.

Questa operazione può avvenire in molti modi, tuttavia un criterio di classificazione dei convertitori ADC si può effettuare in base alla velocità di conversione.

- Convertitori veloci (flash): In questo tipo, la codifica insegue continuamente le variazioni dell'ingresso analogico, con precisioni molto spinte.
- L'impiego principale dei convertitori ADC flash si ha in trasduttori che forniscono informazioni a microprocessori; questi componenti possono lavorare infatti a notevoli velocità.
- Convertitore per misure: In questo tipo, la codifica è più lenta, ma la precisione è molto spinta.

Questi tipi di convertitori trovano largo impiego negli strumenti di misura, dove la contemporaneità del dato analogico con il dato digitale ha poca importanza, mentre la precisione è fondamentale.

Convertitore analogico-digitale In informatica, un dispositivo per la conversione dei dati dalla forma analogica alla forma digitale. Tale conversione si rende necessaria per l'elaborazione, la memorizzazione e la trasmissione di grandezze analogiche (che variano con continuità) con dispositivi digitali, che invece funzionano in modo

"discreto". La conversione inversa, da digitale ad analogico, è espletata dai convertitori digitale-analogico.

In un convertitore analogico-digitale (ADC, Analog to Digital Converter), i segnali analogici applicati vengono campionati con cadenza assegnata, cioè se ne prelevano i valori in istanti separati da intervalli di tempo fissi. I valori dei campioni vengono quindi espressi in forma binaria, come combinazioni di 0 e 1. Ne risultano dei codici che possono essere impiegati in diversi sistemi di elaborazione o di comunicazione.

Un convertitore digitale-analogico (DAC, Digital to Analog Converter) è un dispositivo per la conversione di una combinazione digitale in un livello analogico di tensione o di corrente. I DAC sono oggi largamente impiegati nei lettori di compact disc, nei riproduttori di nastri o videocassette e, in generale, negli apparati per l'elaborazione digitale di segnali audio e video.

Il convertitore che abbiamo utilizzato ADC0808 è un dispositivo monolitico C-Mos, che al suo interno è formato da:

- Un multiplexer a otto canali d'ingresso e tre data select che può trasmettere diversi segnali su di un unico canale o dividendolo in bande di frequenza o effettuando un campionamento sequenziale dei dati, e trasferendoli a tempi successivi.
- Un convertitore Analogico – Digitale a otto bit di risoluzione, Che usa la tecnica di conversione ad approssimazione successiva "SAR";

Attraverso il metodo del multiplexaggio dei diversi segnali analogici è stato possibile utilizzare un solo convertitore .

## Quantizzazione

Il processo di digitalizzazione dei segnali analogici introduce il concetto di quantizzazione. Gli infiniti valori del segnale analogico devono essere quantizzati, ovvero raggruppati in un certo numero di fasce delimitate da livelli fissi detti livelli di quantizzazione; a ciascuna fascia di valori analogici corrisponderà un valore digitale. La distanza fra due livelli di quantizzazione costituisce il passo di quantizzazione  $Q$  a cui corrisponde il valore del bit meno significativo LSB.

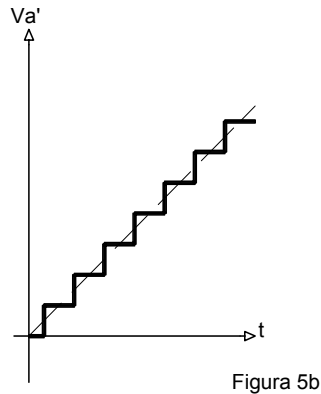
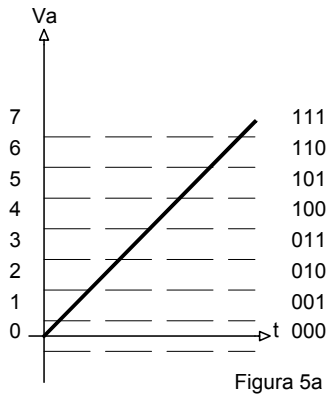
Un dato digitale di  $n$  bit può esprimere  $2^n$  valori, questo valore viene preso come fondo-scala della grandezza analogica.

conseguentemente il valore minimo di scala sarà il rapporto fra il massimo valore di fondo-scala e il numero di valori esprimibili dal dato digitale di  $n$  bit.

In figura 5a è illustrato un segnale a rampa  $V_a'$  variabile da 0 a 7.5 V, con i corrispondenti valori digitali. In figura 5b è riportata la forma d'onda a gradinata  $V_a'$  che si otterrebbe riconvertendo i valori digitali; come si vede, per tutti i valori di  $V_a'$  compresi ad esempio fra 2.5 e 3.5 V, il valore binario corrispondente è 011 che, riconvertito, fornirebbe  $V_a'$  uguale a 3V.

Così, per tutti i valori compresi fra 0 e 0.5V, il valore digitale corrispondente è 000. Pertanto l'errore che si commette nella quantizzazione è sempre minore o uguale a 0.5V, pari cioè al valore di  $\frac{1}{2}$  LSB.

Si noti che in figura 5a i livelli di quantizzazione sono disposti in modo da avere al massimo un errore pari a  $\frac{1}{2}$  LSB.

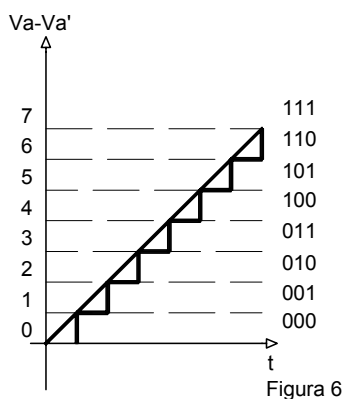


In figura 6 in cui il valore digitale 000 è stata associata la fascia di tensione analogica 0-1V, evidenzia un errore di quantizzazione pari a LSB.

In un ADC i valori digitali di uscita non riproducono dunque

fedelmente il segnale di ingresso ma ne danno una rappresentazione approssimata, tanto più precisa tanto più è piccolo il passo di quantizzazione  $Q$ .

Esistono convertitori a 8, 10, 12 bit, che consentono rispettivamente 256, 1024, 4096 livelli di quantizzazione.



Il numero di bit in uscita di un convertitore A/D, così come il numero dei bit in ingresso di un convertitore D/A, viene generalmente chiamato risoluzione, perchè indica il valore minimo del segnale di ingresso che può essere riportato in uscita (il rapporto fra il valore massimo convertibile e i valori esprimibili da un dato

digitale di  $n$  bit).

## CAMPIONAMENTO

Utilizzato nei convertitori A/D. La conversione consiste nel prelevamento di un campione di un segnale ad un dato istante e nella determinazione del corrispondente valore digitale, che resterà fisso finché non verrà prelevato un altro campione per una nuova conversione. La frequenza con cui il segnale viene prelevato è detta frequenza di campionamento; questa frequenza ha un'importanza fondamentale in riferimento al contenuto del segnale campionato e alle possibilità di ricostruire fedelmente il segnale analogico originario.

Il teorema del campionamento, noto come teorema di Shannon, stabilisce che la frequenza di campionamento deve essere maggiore o uguale al doppio della componente di frequenza più elevata del segnale in esame. Benché la frequenza minima di campionamento è il doppio di quella del segnale analogico, si preferisce campionare a frequenza maggiore in modo da ottenere in uscita una rappresentazione più fedele del segnale di ingresso.

Le irregolarità nella frequenza di campionamento e la mancanza di sincronizzazione fra la frequenza di campionamento e il segnale da convertire provocano talvolta, nel segnale ricostruito, fenomeni di battimento e la comparsa di distorsioni di fase.

Questi errori sono particolarmente evidenti e dannosi quando la frequenza di campionamento è di poco superiore a quella del segnale (da 2 a 8 volte). Risulta invece facile evitarli se la frequenza di campionamento è elevata o sincronizzata con quella del segnale.

Dal momento che i convertitori A/D impiegano un tempo finito per digitalizzare un segnale analogico in ingresso, eventuali variazioni del segnale durante il processo di conversione possono determinare errori significativi. Se la variazione del segnale

di ingresso durante il tempo di conversione è superiore del bit LSB, il dato digitale di uscita può presentare un errore superiore ad 1 LSB; in pratica la risoluzione specificata per il convertitore non viene mantenuta. Questo problema può essere risolto utilizzando circuiti di campionamento e mantenimento.

### **ADC0808.**

Questi tipi di convertitori ADC sono i più usati nel mondo dell'acquisizione dati industriale. Infatti nel nostro caso è stato utilizzato un convertitore analogico-digitale ad approssimazioni successive. ADC0808 è un convertitore che usa una tecnologia monolitica a CMOS. A 8 bit di uscita ed 8 ingressi gestiti da un multiplexer, inoltre è compatibile con i microprocessori più usati.

Questo tipo di convertitore analogico-digitale utilizza il metodo delle approssimazioni successive. Al suo interno c'è un comparatore stabilizzato ad alta impedenza, il 256R divisore di tensione, un interruttore analogico (switch tree) e un registro ad approssimazioni successive.

Questo convertitore offre alte velocità, alta precisione, alta stabilità in temperatura, basso consumo, ed è in grado di mantenere il dato in memoria per un buon periodo di tempo. Il dispositivo è stato progettato per essere inserito in processi o in macchine di controllo e consumo, e applicazioni nell'automazione.

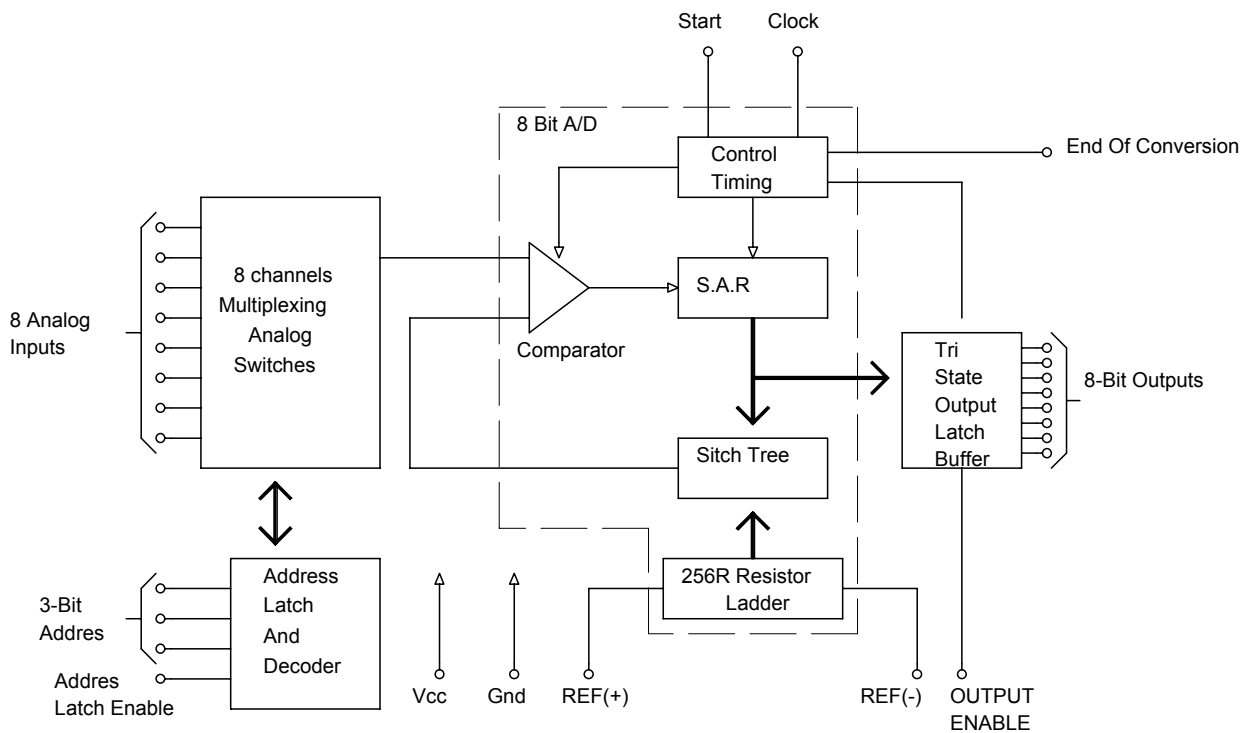


Figura 15

## ADC AD APPROSSIMAZIONI SUCCESSIVE

Il metodo di conversione ad approssimazioni successive è sicuramente il più diffuso in quante consente un buon compromesso fra velocità di conversione e risoluzione.

Nel convertitore a 4 bit in figura 14 il segnale di ingresso  $V_a$  viene comparato con precisi livelli di tensione generati dal convertitore DAC. Dopo l'applicazione del comando di conversione SOC (Start Of Conversion), che azzerà le uscite e inizializza il sistema, il registro ad approssimazioni successive SAR (Successive Approximation Register) si trova nello stato 1000. Questo dato viene presentato in ingresso al DAC che fornisce il primo livello analogico, pari a metà della tensione di fondo scala del convertitore, da confrontare con il segnale  $V_a$ .

Se  $V_a > V_a'$ , l'uscita del comparatore è alta e il bit più significativo del SAR, e quindi anche quello del dato di uscita, si porta a zero. A questo punto, in sincronismo con il clock, viene portato ad 1 il secondo bit più significativo del SAR, cosicché il dato presente sugli ingressi del DAC sarà 1100 oppure 0100 a seconda del risultato

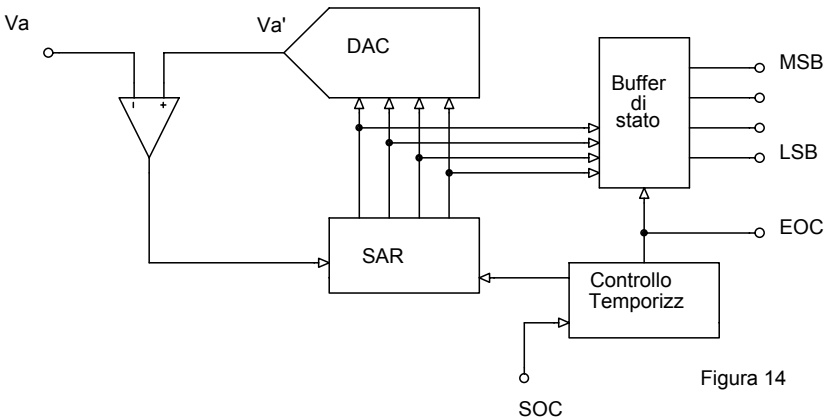


Figura 14

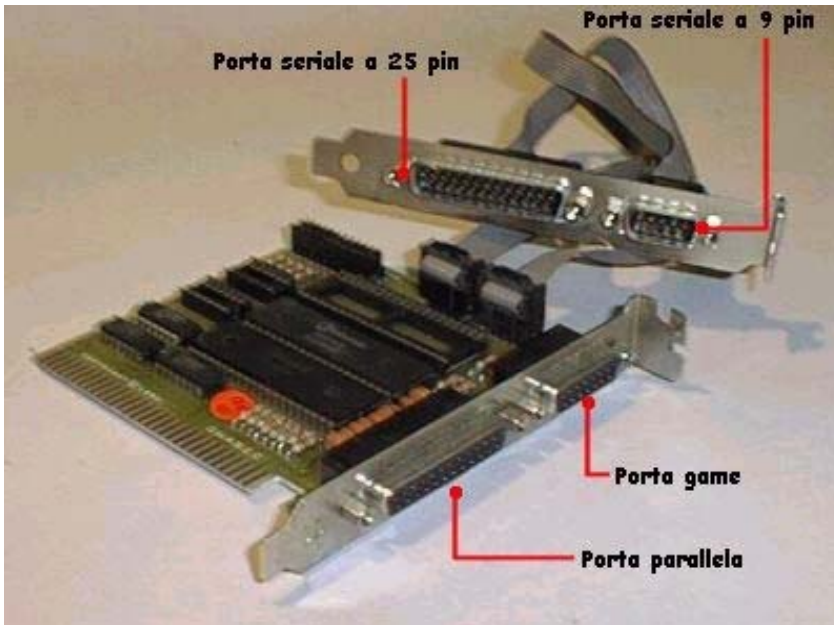
del confronto precedente.

Il secondo confronto porta a 0 o mantiene ad uno il secondo bit del SAR e del buffer di uscita, a seconda che  $V_a$  risulti minore o maggiore di  $V_a'$ . Con procedimento analogo

vengono effettuati il terzo e il quarto confronto. Alla fine della conversione, ovvero dopo quattro confronti successivi, il dato digitale contenuto nel buffer di uscita è pronto e valido; il blocco di temporizzazione segnala la fine della conversione EOC (End Of Conversion) e l'uscita può essere letta.

Utilizzando la tecnica delle approssimazioni successive si richiedono  $n$  interazioni, e quindi  $n$  cicli di clock, per convertire una tensione di ingresso in un dato ad  $n$  bit, indipendentemente dal valore della tensione stessa. Questo fatto unitamente alle buone prestazioni in risoluzione, fa preferire la tecnica ad approssimazioni successive nella realizzazione di convertitori ad media velocità, adatti ad esempio per applicazioni con microprocessori.

## Trasmissione dati Tramite Parallela



### SPP (Standard Parallel Port)

Le porte di questo tipo rispettano lo standard delle prime porte parallele create agli inizi degli anni 80 per i PC IBM e garantiscono quindi una compatibilità verso il basso che permette di usare anche periferiche

"datate".

In questo tipo di parallele, il flusso dei dati è rigidamente monodirezionale, dal PC alla periferica. Vista la velocità, si va da un minimo di 50Kb ad un massimo di 150 Kb al secondo, è possibile connettere solamente stampanti di vecchia generazione come quelle ad aghi ma non quelle laser più moderne che necessitano di prestazione maggiori.

Si hanno a disposizione in questo tipo di porte alcuni registri posti agli indirizzi 0378H e 0278H. Nonostante la rigidità di queste porte monodirezionali, è possibile con delle apposite gestioni software ottenere un flusso bidirezionale (anche se limitato a 4 bit) denominato Nibble mode.

Se questo tipo di porte andavano bene per le prime stampanti ad aghi degli anni 80, col passare del tempo le troppe limitazioni sono affiorate ed hanno portato ad una modifica sostanziale di questa porta che è stata adattata per un flusso di dati

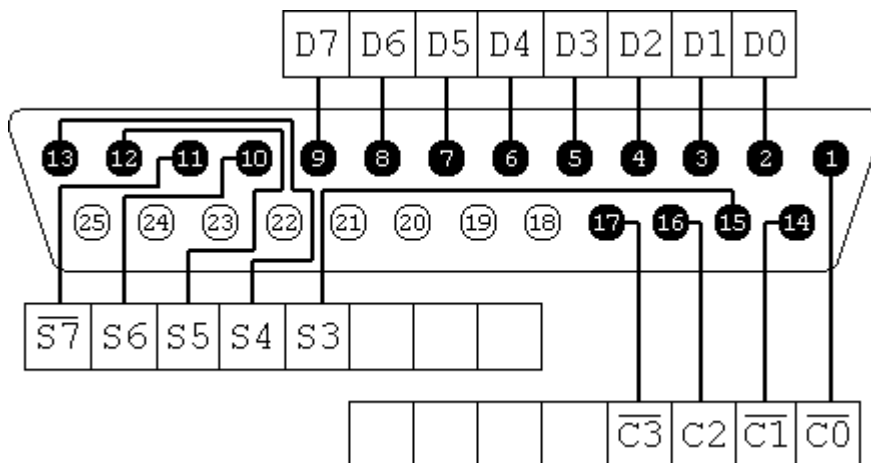
bidirezionale (standard IBM PS/2). In questo modo la connessione non è ristretta alle sole stampanti ma anche ad altre periferiche.

### **EPP - Enhanced Parallel Port**

Introdotta nel 1994 nei computer di tipo AT e ATX, mantiene la compatibilità con il vecchio standard SPP anche essendo di gran lunga differente. Può gestire dati in modo bidirezionale con una velocità che va da 500 Kbytes fino a 2 Mbytes al secondo ed è ideale per connettere periferiche esterne quali hard disk e cd-rom portatili (olte ovviamente a stampanti laser di ogni genere).

### **ECP - Extended Capabilities Port**

Questo tipo di parallela è praticamente identico a quello EPP ma ne migliora alcune funzionalità oltre ad introdurne di nuove quali ad esempio il RLE (Run length Encoding), ovvero la possibilità di codificare e comprimere i dati prima di trasferirli in modo da ottenere velocità di trasmissione dati maggiori. Ideale per scanner, fax e stampanti, può raggiungere livelli di compressione fino a 64:1!



DB25-F	Signal	I/O	Reg.Bit
1	OUT	-STROBE-	C0-
2	OUT	DATA-0	D0
3	OUT	DATA-1	D1
4	OUT	DATA-2	D2
5	OUT	DATA-3	D3
6	OUT	DATA-4	D4
7	OUT	DATA-5	D5
8	OUT	DATA-6	D6
9	OUT	DATA-7	D7
10	IN	- ACK	S6+
11	IN	+ BUSY	S7-
12	IN	+ PaperEnd	S5+
13	IN	+ SELECTIN	S4+
14	OUT	-Auto Fd	C1-
15	IN	- Error	S3+
16	OUT	- Init	C2+
17	OUT	- Select	C3-
18	GND		
19	GND		
20	GND		
21	GND		
22	GND		
23	GND		
24	GND		
25	GND		

## **Gestione dell'intero sistema attraverso un software realizzato con Delphi5.**

Il programma visualizza e gestisce i dati forniti dai trasduttori e convertiti, attraverso la prima porta parallela di indirizzo \$378.

Per leggere i dati in uscita al convertitore è stato necessario settare tramite il bios la porta parallela in modalità EPP+ECP, che consente la bidirezionalità delle linee dati, quindi la porta parallela è in grado di leggere le otto linee dati, che nella modalità SPP sono rigorosamente uscite.

Attraverso la bidirezionalità della porta parallela, è stato possibile leggere il dato proveniente dalle otto uscite digitali del convertitore A/D.

Sulle linee di controllo vengono gestiti gli indirizzi dati del multiplexer che ci permettono di selezionare un canale analogico e inviarlo alla conversione.

Attraverso il canale di Strobe viene gestito lo START del convertitore, un segnale che da l'inizio alla conversione.

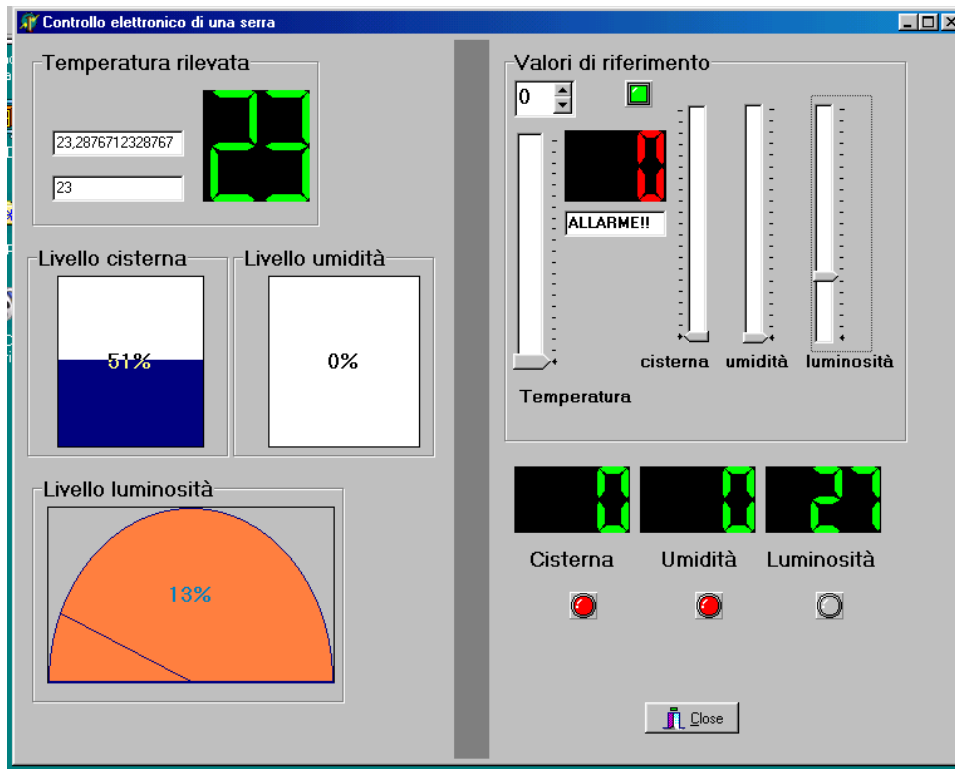
Con il programma faccio in modo di abilitare le uscite da D0 a D7 in modo bidirezionale.

Una volta fatto questo seleziona in modo rigoroso il canale da convertire; quando il canale è selezionato posso inviare l'impulso di start, che da il via alla conversione.

Attraverso l'utilizzo di ritardi (SLEEP) equilibrati abbiamo calibrato i tempi di conversione .

Una volta finito il lavoro del convertitore, viene letto il dato e attraverso una rielaborazione del dato mediante una routine mette a video il valore letto.

I differenti segnali forniti dai trasduttori vengono visualizzati su una schermata del computer attraverso oggetti grafici, che vengono costantemente aggiornati.



Nella schermata a sinistra vengono monitorizzati tutti i differenti segnali provenienti dai trasduttori cioè i parametri fisici all'interno della serra.

In alto si può vedere il valore di temperatura approssimata; al centro troviamo indicatori dell'acqua di una cisterna e il livello d'umidità del terreno e per finire vediamo una barra indicatrice del livello di luminosità.

Nella schermata di destra possiamo notare quattro differenti slider, che ci permettono di impostare differenti valori di riferimento al fine di gestire gli attuatori attraverso una seconda porta parallela.

La seconda porta parallela di indirizzo \$278 è configurata in modalità SPP perché ha lo scopo di pilotare i relè posizionati sulle linee dati da D0 a D5.

## Scrittura software in Delphi

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, ComCtrls, Spin, ExtCtrls, ALed, DBCtrls, Grids, DBGrids, Db,  
DBTables, TeEngine, Series, TeeProcs, Chart, DBChart, ShellApi, Buttons,  
Lcd99, Gauges;
```

```
type
```

```
TForm1 = class(TForm)  
    Timer1: TTimer;  
    Edit1: TEdit;  
    Edit2: TEdit;  
    Timer2: TTimer;  
    LCD991: TLCD99;  
    TrackBar1: TTrackBar;  
    LCD992: TLCD99;  
    Splitter2: TSplitter;  
    Edit6: TEdit;  
    GroupBox1: TGroupBox;  
    GroupBox2: TGroupBox;  
    hhALed1: ThhALed;  
    SpinEdit1: TSpinEdit;  
    GroupBox3: TGroupBox;  
    Gauge1: TGauge;  
    GroupBox4: TGroupBox;  
    Gauge2: TGauge;  
    BitBtn1: TBitBtn;  
    GroupBox5: TGroupBox;  
    Gauge3: TGauge;  
    TrackBar2: TTrackBar;  
    TrackBar3: TTrackBar;  
    TrackBar4: TTrackBar;  
    Label1: TLabel;  
    Label2: TLabel;  
    Label3: TLabel;  
    Label4: TLabel;  
    LCD993: TLCD99;  
    LCD994: TLCD99;
```

```

LCD995: TLCD99;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
hhALed2: ThhALed;
hhALed3: ThhALed;
hhALed4: ThhALed;
Label8: TLabel;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure SpinEdit1Change(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure TrackBar2Change(Sender: TObject);
procedure TrackBar3Change(Sender: TObject);
procedure TrackBar4Change(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  a,al,b,c,f,z,xx,umi,zaz:byte;
  y,yy,ris_um,uau:real;
  w,i:integer;
const control=$378+2;
      data=$378;
      x=5.11;
      ww=38;
implementation
uses zlpportio;
{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);

```

```

begin
  portwriteb(control,$FF); //01
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  portwriteb(control,$FB); //00
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  portwriteb(control,$F7); //11
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
  portwriteb(control,$F3); //10
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  a := PortReadB(Control); //Leggo lo stato del Control
  a := a or $20; //Pongo ad 1 il bit 5 di controllo per far leggere la porta
  PortWriteB(Control, a ); //Immetto il nuovo stato sulla porta
  if lcd991.Value >= lcd992.Value then
    Edit6.text := ('ALLARME!!')
  else Edit6.text := (' TUTTO OK');
  label8.caption := "";
  timer1.Enabled := true;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
// label qui;
var k,p,lh:byte;
begin
  repeat
    begin

      //qui:portwriteb(control,$FB); //00
      //-----
      portwriteb(control,$F7); //11 LIVELLO
      z := PortReadB(Control); //Rileggo il Control
    end
  until false;
end;

```

```

z := z And $Fe; //Pongo ad 1 il bit 1 di controllo per fare leggere il convertitore
PortWriteB(Control, z ); //Immetto il novo stato sulla porta
sleep(30);
b := PortReadB(Control); //Rileggo il Control
b := b or $01; //Pongo ad 0 il bit 1 di controllo per non far leggere il convertitore
PortWriteB(Control, b ); //Immetto il nuovo stato sulla porta
//z := PortReadB(Control); //Rileggo il Control
//z := z And $Fe; //Pongo ad 1 il bit 1 di controllo per fare leggere il convertitore
//PortWriteB(Control, z ); //Immetto il novo stato sulla porta
sleep(30);
xx := PortReadB (Data); //Leggo lo stato dei Data
yy:=(xx*110)/255;
Gauge1.Progress:=round(yy-3);

sleep(30);

//-----
portwriteb(control,$Fd);//01  UMIDITA'

z := PortReadB(Control); //Rileggo il Control
z := z And $Fe; //Pongo ad 1 il bit 1 di controllo per fare leggere il convertitore
PortWriteB(Control, z ); //Immetto il novo stato sulla porta
sleep(30);
b := PortReadB(Control); //Rileggo il Control
b := b or $01; //Pongo ad 0 il bit 1 di controllo per non far leggere il convertitore
PortWriteB(Control, b ); //Immetto il nuovo stato sulla porta
sleep(30);
umi := PortReadB (Data); //Leggo lo stato dei Data
ris_um:=(umi*110)/255;
Gauge3.Progress:=round(ris_um-3);
sleep(30);
//-----
portwriteb(control,$F3);//10 luminosita'

z := PortReadB(Control); //Rileggo il Control
z := z And $Fe; //Pongo ad 1 il bit 1 di controllo per fare leggere il convertitore
PortWriteB(Control, z ); //Immetto il novo stato sulla porta
sleep(30);
b := PortReadB(Control); //Rileggo il Control
b := b or $01; //Pongo ad 0 il bit 1 di controllo per non far leggere il convertitore
PortWriteB(Control, b ); //Immetto il nuovo stato sulla porta
sleep(30);
zaz := PortReadB (Data); //Leggo lo stato dei Data

```

```

uau:=(zaz*110)/255;
Gauge2.Progress:=round(uau-3);
sleep(30);

//-----
portwriteb(control,$FB);//00  TEMPERATURA
z := PortReadB(Control); //Rileggo il Control
z := z And $Fe; //Pongo ad 1 il bit 1 di controllo per fare leggere il convertitore
PortWriteB(Control, z ); //Immetto il nuovo stato sulla porta
sleep(30);
b := PortReadB(Control); //Rileggo il Control
b := b or $01; //Pongo ad 0 il bit 1 di controllo per non far leggere il convertitore
PortWriteB(Control, b ); //Immetto il nuovo stato sulla porta
sleep(30);
//portwriteb($378,64);
//sleep(200);
al := PortReadB (Data); //Leggo lo stato dei Data
sleep(30);
y := al/x; //Lo divido per la costante
Edit1.Text := FloatToStr (y); //Visualizo il risultato
w := round (y); //Approssimo il valore in un intero
Edit2.Text := IntToStr(w); //Visualizzo il valore
lcd991.value:=inttostr(w);

sleep(30);

//-----Accensione lampada quando la luminosità rilevata è maggiore
//-----di quella impostata-----
if Gauge2.Progress>=(100-(trackbar4.Position)) then
begin
  hhAled4.TrueColor:=cllime;
  label8.Caption:='ACCENDI LE LAMPADE';
  p:=PortReadB($278);
  p:=p or 1;
  PortWriteB($278,p);
  beep;
end
else
begin
  hhAled4.TrueColor:=clActiveBorder;
  p:=PortReadB($278);
  p:=p AND $FE;
  PortWriteB($278,p);

```

```

label8.Caption:="";
end;

//-----
if Gauge3.Progress<=(100-(trackbar3.Position)) then
begin
if (100-(trackbar2.Position))>=Gauge1.Progress then
lh:=PortReadB($278);
lh:=lh or $18;
PortWriteB($278,lh);
end
else
begin
lh:=PortReadB($278);
lh:=lh or $30;
PortWriteB($278,lh);
end;
end;
until y>0;

//goto qui;

end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
Timer2.Enabled:=True; //Abilita il timer2
z := PortReadB(Control); //Leggo lo stato del Control
z := z or $01; //Pongo ad 1 il bit 0 di controllo per far leggere il convertitore
PortWriteB(Control, z ); //Immetto il nuovo stato sulla porta
timer2.Enabled:=false;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
portwriteb(data,$16);
end;

procedure TForm1.TrackBar1Change(Sender: TObject);
var bb,pp:byte;
begin

```

```

LCD992.value := IntToStr (100-(Trackbar1.Position));
SpinEdit1.Text := IntToStr (100-(Trackbar1.Position));
c:= strtoint (LCD991.Value); //Il valore rilevata
bb:= strtoint(SpinEdit1.text); //Il valore nello spinedit viene memorizzato in una variabile
i f bb - c >=0 then //Effettuo una sottrazione tra il valore di riferimento e la temperatura
begin //Se il risultato è maggiore di zero
    pp:=PortReadB($278);
    pp:=pp or 4;//accende il riscaldamento rele 3
    PortWriteB($278,pp);
    Edit6.text := (' Tutto OK'); //Scrivi che è tutto ok
    lcd992.OnColor:=clLime;

    hhaLed1.truecolor:=clLime; //Accendi di verde il led
    //label1.Caption:=inttostr(bb);
end
else
begin
    Edit6.text := ('ALLARME!!'); //
    pp:=PortReadB($278);
    pp:=pp AND $FB; //and spegne
    PortWriteB($278,pp);
    lcd992.OnColor:=clred;
    hhaLed1.truecolor:=clRed;

end;
end;

```

```

procedure TForm1.SpinEdit1Change(Sender: TObject);
begin
    if SpinEdit1.Text = " " Then w:=0 //Se nello spinedit non c'è nessun valore allora immetti zero
    Else w:=StrToInt(SpinEdit1.Text);//Altrimenti leggi il valore e memorizzalo in una variabile
    Trackbar1.Position:=(100 -(w));//La trackbar2 assume il valore w

end;

```

```

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    zlpportio.zliostop;
end;

```

```

procedure TForm1.TrackBar2Change(Sender: TObject);

```

begin

```
LCD993.value := IntToStr (100-(Trackbar2.Position));  
if Gauge1.Progress>=(100-(trackbar2.Position)) then hhAled2.TrueColor:=clred  
else hhAled2.TrueColor:=cllime;
```

end;

```
procedure TForm1.TrackBar3Change(Sender: TObject);
```

begin

```
LCD994.value := IntToStr (100-(Trackbar3.Position));  
if Gauge3.Progress>=(100-(trackbar3.Position)) then hhAled3.TrueColor:=clred  
else hhAled3.TrueColor:=cllime;
```

end;

```
procedure TForm1.TrackBar4Change(Sender: TObject);
```

begin

```
LCD995.value := IntToStr (100-(Trackbar4.Position));
```

end;

end.