SELIN s.n.c Elettronica e Informatica via Roata Canale, 41 - CUNEO
Tel. 0171 / 34.65.54 Fax. 0171 / 34.69.70 E-mail selincn@tin.it P.IVA 02188360040

SCHEDA MULTI I/O DI INTERFACCIA PER ATTUATORI/SENSORI SU PORTA PARALLELA (standard 1284)

SCHEDE GUIDA ESERCITAZIONI

SCHEDA MULTI I/O DI INTERFACCIA PER ATTUATORI/SENSORI

SU PORTA PARALLELA EPP (STANDARD 1284)

ESERCITAZIONE N.1

TITOLO: ***DESCRIZIONE E PRINCIPIO DI FUNZIONAMENTO***

OBIETTIVO : *L'ESERCITAZIONE SI PROPONE DI DESCRIVERE LE CARATTERISTICHE E IL PRINCIPIO DI FUNZIONAMENTO PER L'UTILIZZO DEL DISPOSITIVO*

---

1) La scheda di interfaccia per attuatori/sensori (KPRN_IO) permette in modo rapido ed efficace di comandare dispositivi esterni tramite un PC.

2) Lo schema a blocchi si trova nel manuale d'uso e documentazione a pag.3.: a sinistra del connettore della scheda si possono vedere i segnali della porta parallela del PC, mentre sulla destra vi sono gli stessi segnali con le denominazioni tipiche della scheda.

Le uscite digitali sono gestite dall'integrato 74HC374, controllato dal segnale CLK(STROBE), mentre gli ingressi digitali sono gestiti dall'integrato 74HC244, a sua volta abilitato dal segnale GATE(AUTO).

Il convertitore D/A (TLC7524) viene abilitato dal segnale SEL_CS(SEL) negato, mentre il clock è fornito dal segnale WR_DA(INIT); l'uscita del convertitore D/A è anche collegata ad un comparatore, che ricevendo sul piedino invertente un segnale a dente di sega, riesce a produrre il segnale OUT PWM (PWM=Modulazione a larghezza di impulso).

Un secondo comparatore provvede all'abilitazione dell'oscillatore a dente di sega (NE555), ricevendo sul piedino non invertente una tensione di riferimento (RIF) e sul piedino invertente una tensione proporzionale alla corrente circolante nel carico (I SENSE).

Il convertitore A/D (TLC549), di tipo seriale, viene abilitato dal segnale SEL_CS(SEL) e utilizza come clock il bit 0 del bus dati (DB0/SCK), mentre il dato (1 bit) è presente sul segnale DATA_OUT(BUSY); per la completa conversione il dato deve essere letto sequenzialmente 8 volte ed il primo bit letto è il più significativo

3) Analizzando il layout del dispositivo (vedi manuale d'uso e documentazione pag.4) si possono notare :

- Connettore per il cavo da collegare a sua volta alla porta parallela del PC,
- Morsetti della sezione analogica (in A/D, out D/A e relativi terminali gnd),
- 4 relè con i corrispondenti morsetti, i led di segnalzione delle uscite digitali (il led giallo è il bit meno significativo),
- 4 mosfet di potenza con il morsetti per il collegamento della massa e del drain,
- Morsetti per l'input digitale,
- Dip-switch e i pulsanti per simulare ingressi digitali,
- Morsetti di alimentazione della scheda,
- 2 fusibili (fusibile da 500mA per la logica +5v e +12v, fusibile da 2A per relè, mosfet, led, etc.)

4) Passando all'analisi degli schemi elettrici nella prima scheda (titolo:Interfaccia didattica per porta parallela) si trova un complessivo con i blocchi:

PWR_OUT (relè, mosfet e led)
IN_FLAG (pulsanti e dip-switch degli ingressi digitali)
A/D_D/A_1 (convertitori e generazione segnale PWM)
ALIM_1 (generazione +5v e tensione di riferiemento).

5) Nella seconda scheda (stadio di ingresso segnali digitali) si possono vedere gli zener di protezione e le resistenze che assicurano, a dip-switch aperti, il valore logico 0 sugli ingressi dell'integrato 74HC244

6) Nella terza scheda (alimentatore scheda interfaccia printer) vi è l'integrato LM7805 per la tensione +5VCC e l'integrato LM336 per la tensione di riferimento +5 VREF (il trimmer R35 e il test point TP8 servono rispettivamente per la regolazione e la misura e si trovano vicino ai fusibili); i circuiti sono protetti tramite 2 fusibili da 0,5A e da 2A

7) La quarta scheda contiene gli schemi relativi ai relè e ai mosfet di potenza; ogni segnale di uscita (OUT_1/OUT_8) è segnalato tramite un led (D1/D8); il piedino source del mosfet Q6 è collegato a massa tramite una resistenza da 0,1 ohm per la generazione del segnale I_SENSE, mentre il comando del mosfet stesso può essere normale (OUT_8) o provenire dal circuito generatore pwm (PWM_OUT) .

Il ponticello JP1 si trova vicino alla resistenza da 0,1 ohm 5 watt: posizionato sulla destra (sempre mantenendo l'orientamento della scheda come indicato nel layout di pag.4) il mosfet è collegato a PWM_OUT, mentre posizionato sulla sinistra il mosfet viene comandato direttamente dal 74HC374, cioè dal segnale OUT_8. Il test point TP4 è collegato a I_SENSE.

8) La quinta ed ultima scheda del circuito elettrico (sezione analogica/digitale) contiene il comparatore (U11B) che stabilisce la corrente massima ammessa tramite R65 (secondo trimmer a partire da destra).

9) Il circuito di generazione del segnale a dente di sega è costituito dall'integrato 555 (U6) con i comparatori U12B e U11A, i trimmer R44, R45, R63, il ponticello JP3 e i test point TP3 e TP6 per la taratura. Da notare il segnale di abilitazione ENABLE che proviene dall'integrato 74HC374 (OUT_8). Il ponticello JP3, posizionato a destra (sempre mantenendo l'orientamento della scheda come indicato nel layout di pag.4) è collegato a +12VLOG, mentre posto a sinistra è collegato a +5VREF.

L'integrato TLC7524 è il convertitore D/A; i trimmer R55 e R59, il ponticello JP2 e il test point TP2 provvedono alla taratura ed alla regolazione della tensione analogica di uscita.

Il convertitore A/D accetta un campo di tensioni al suo ingresso da 0 a 5 volt positivi, il clock (SCK) viene fonito tramite il bit 0 del bus dati, mentre il dato seriale, 1 bit alla volta, viene letto 8 volte in modo sequenziale, con il primo bit a peso più alto. Sempre sullo stesso foglio si trova il circuito per la generazione del segnale ACK_IRQ (150 microsec)

10) Tutti i trimmer citati nei punti precedenti sono stati preventivamente tarati e non necessitano di ulteriori regolazioni salvo applicazioni particolari.

11) In allegato agli schemi elettrici vi sono due tavole con gli schemi di connessione della porta parallela standard e della porta parallela bidirezionale

SCHEDA MULTI I/O DI INTERFACCIA PER ATTUATORI/SENSORI

SU PORTA PARALLELA EPP (STANDARD 1284)

ESERCITAZIONE N.2

TITOLO: ***FUNZIONAMENTO OUTPUT DIGITALE***

OBIETTIVO : ***L'ESERCITAZIONE SI PROPONE DI DARE LE INDICAZIONI BASILARI PER L'ATTIVAZIONE DELLE USCITE DIGITALI , FORNENDO UN SEMPLICE LISTATO DI PROVA IN LINGUAGGIO C++***

---

1) Prima di alimentare la scheda di interfaccia per attuatori/sensori su porta parallela EPP (chiamata in questa sezione più semplicemente "scheda di interfaccia") leggere e consultare il "Manuale d'uso e documentazione".

2) Collegare la scheda di interfaccia con un alimentatore stabilizzato 12V 2.5A (vedi Caratteristiche e specifiche in "Manuale d'uso e documentazione").

2) Collegare la scheda di interfaccia alla porta parallela del PC (486 o superiori) tramite il cavo fornito. In questo caso la scheda di interfaccia sarà comandata tramite la porta LPT1 (quella normalmente usata per la stampante del PC). Nel caso si volesse installare la scheda EPP (opzionale) sul proprio PC, avendo a disposizione uno slot libero, questa deve essere settata come LPT2, tramite gli appositi ponticelli a bordo della scheda EPP stessa

3) La scheda di interfaccia per attuatori/sensori (KPRN_IO) utilizza due modi di trasferimento dati contenuti nel protocollo standard 1284: il modo Compatibility per l'uscita (PC → scheda) ed il modo Byte per l'ingresso (scheda → PC).

Come si può notare dalle indicazioni fornite nel "Manuale d'uso e documentazione" (pag. 5) l'uscita del dato viene abilitata tramite il segnale STROBE sull'integrato 74HC374 (pin CLK) e deve essere anche attivato il bit 5 per stabilire la direzione (bit 5 H = input, bit 5 L = output).

9) Indipendentemente dal tipo di linguaggio utilizzato per comandare la scheda di interfaccia deve essere rispettata questa sequenza operativa:

Output di un dato sulla scheda di interfaccia

passo	*descrizione*	*operazione*
1	Inizializzazione porta	Registro di controllo=0 (strobe=0, direzione=0)
2	Dato sul bus dati porta	Dato presente in DB0-DB7
3	Impulso di clock per 74HC374	Registro di controllo=1 (strobe=1, direzione=0) Registro di controllo=0 (strobe=0, direzione=0)
4	Dato sulle uscite (led, relè e mosfet)	Dato presente sulle uscite del 74HC374

Indirizzi/bit

	LPT1	*LPT2*	*bit*
Indirizzo porta	378H	278H	DB0-DB7
Indirizzo registro di controllo	37AH	27AH	bit 0=strobe bit 5=direzione

NOTA: Il segnale STROBE (bit 0 del registro di controllo) è negato. L'integrato 74HC374 commuta l'uscita con la transizione positiva (0→1) del clock, per cui la sequenza reale è 1-0-1.

Esempio di programma in C++ per scrittura dato
(output continuo del dato con incremento)

```
//EPP1.CPP
//uscita dato sulla porta con incremento

#include <iostream.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>

main()
{
 unsigned char dato;
 int porta=0x378;
 int reg_c=0x37A;

 outportb(reg_c,0);                    // inizializzazione porta
 clrscr();
 gotoxy(10,8);
 cout<<"Uscita digitale";
 gotoxy(10,10);
 cout<<"dato binario : ";
 for (dato=0;!kbhit();dato++)          // il ciclo è abilitato finchè non si preme un tasto
     {
     outportb(porta,dato);             // il dato è presente sul bus dati porta DB0-DB7
     outportb(reg_c,1);                // impulso di clock strobe=1, direzione=0
     outportb(reg_c,0);                // impulso di clock strobe =0, direzione=0
     gotoxy(25,10);
     printf("%3d",dato);               // visualizzazione dato
     delay(1000);
     }
return 0;
}
```

registro di controllo

bit	*segnale*
0	strobe (negato) abilitazione porta output digitale (0-1-0)
1	auto (negato) abilitazione porta input digitale (0-1-0)
2	write_da convertitore DA 0=attivo
3	sel_cs (negato) selezione AD/DA 1=AD 0=DA
5	direzione bus dati 0=out 1=in

SCHEDA MULTI I/O DI INTERFACCIA PER ATTUATORI/SENSORI

SU PORTA PARALLELA EPP (STANDARD 1284)

ESERCITAZIONE N.3

TITOLO: ***FUNZIONAMENTO INPUT DIGITALE***

OBIETTIVO : ***L'ESERCITAZIONE SI PROPONE DI DARE LE INDICAZIONI BASILARI PER L'ATTIVAZIONE DEGLI INGRESSI DIGITALI , FORNENDO UN SEMPLICE LISTATO DI PROVA IN LINGUAGGIO C++***

---

1) Prima di alimentare la scheda di interfaccia per attuatori/sensori su porta parallela EPP (chiamata in questa sezione più semplicemente "scheda di interfaccia") leggere e consultare il "Manuale d'uso e documentazione".

2) Collegare la scheda di interfaccia con un alimentatore stabilizzato 12V 2.5A (vedi Caratteristiche e specifiche in "Manuale d'uso e documentazione").

2) Collegare la scheda di interfaccia alla porta parallela del PC (486 o superiori) tramite il cavo fornito. In questo caso la scheda di interfaccia sarà comandata tramite la porta LPT1 (quella normalmente usata per la stampante del PC). Nel caso si volesse installare la scheda EPP (opzionale) sul proprio PC, avendo a disposizione uno slot libero, questa deve essere settata come LPT2, tramite gli appositi ponticelli a bordo della scheda EPP stessa

4) La scheda di interfaccia per attuatori/sensori (KPRN_IO) utilizza due modi di trasferimento dati contenuti nel protocollo standard 1284: il modo Compatibility per l'uscita (PC → scheda) ed il modo Byte per l'ingresso (scheda → PC).

5) Come si può notare dalle indicazioni fornite nel "Manuale d'uso e documentazione" (pag. 5) l'ingresso del dato viene abilitato tramite il segnale AUTO sull'integrato 74HC244 (pin GATE o G1+G2); i due segnali si possono trovare nel registro di controllo (indirizzo 37AH) e, nello stesso momento, deve essere anche attivato il bit 5 per stabilire la direzione (bit 5 H = input, bit 5 L = output).

10) Indipendentemente dal tipo di linguaggio utilizzato per comandare la scheda di interfaccia deve essere rispettata questa sequenza operativa:

Input di un dato sulla scheda di interfaccia

passo	*descrizione*	*operazione*
1	Inizializzazione porta	Registro di controllo=20H (auto=0, direzione=1)
2	Impulso di gate per 74HC244	Registro di controllo=22H (auto=1, direzione=1)
3	Lettura del dato sul bus dati	Dato presente in DB0-DB7
4	Disabilitazione gate per 74HC244	Registro di controllo=20H (auto=0, direzione=1)

Indirizzi/bit

	LPT1	*LPT2*	*bit*
Indirizzo porta	378H	278H	DB0-DB7
Indirizzo registro di controllo	37AH	27AH	bit 1=auto bit 5=direzione

SCHEDA MULTI I/O DI INTERFACCIA PER ATTUATORI/SENSORI

SU PORTA PARALLELA EPP (STANDARD 1284)

ESERCITAZIONE N.3

NOTA: Il segnale AUTO (bit 1 del registro di controllo) è negato. L'integrato 74HC244 ha i due segnali di gate (G1 e G2) attivi bassi, per cui la sequenza reale è 1-0-1.

```
//EPP2.CPP
//ingresso continuo dato dalla porta

#include <iostream.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>

main()
{
 unsigned char dato;
 int porta=0x378;
 int reg_c=0x37A;

 int datin;
 outport(reg_c,0x20);                    // inizializzazione porta
clrscr();
 gotoxy(10,8);
 cout<<"Ingresso digitale";
 gotoxy(10,10);
 cout<<"dato in input : ";
                    do
                    {
                    outport(reg_c,0x22);    // segnale di gate auto=1, direzione=1
                    datin=inportb(porta);   //lettura del dato sul bus dati DB0-DB7
                    outport(reg_c,0x20);    // disabilitazione segnale di gate  auto=0, direzione=1
                    gotoxy(27,10);
                    printf("%3d",datin);    // visualizzazione dato
                    } while (!kbhit());     // ciclo abilitato finchè non si preme un tasto
return 0;
}
```

SCHEDA MULTI I/O DI INTERFACCIA PER ATTUATORI/SENSORI

SU PORTA PARALLELA EPP (STANDARD 1284)

ESERCITAZIONE N.4

TITOLO: ***FUNZIONAMENTO OUTPUT ANALOGICO***

OBIETTIVO : ***L'ESERCITAZIONE SI PROPONE DI DARE LE INDICAZIONI BASILARI PER L'ATTIVAZIONE DELL'USCITA ANALOGICA , FORNENDO UN SEMPLICE LISTATO DI PROVA IN LINGUAGGIO C++***

1) Prima di alimentare la scheda di interfaccia per attuatori/sensori su porta parallela EPP (chiamata in questa sezione più semplicemente "scheda di interfaccia") leggere e consultare il "Manuale d'uso e documentazione".

2) Collegare la scheda di interfaccia con un alimentatore stabilizzato 12V 2.5A (vedi Caratteristiche e specifiche in "Manuale d'uso e documentazione").

2) Collegare la scheda di interfaccia alla porta parallela del PC (486 o superiori) tramite il cavo fornito. In questo caso la scheda di interfaccia sarà comandata tramite la porta LPT1 (quella normalmente usata per la stampante del PC). Nel caso si volesse installare la scheda EPP (opzionale) sul proprio PC, avendo a disposizione uno slot libero, questa deve essere settata come LPT2, tramite gli appositi ponticelli a bordo della scheda EPP stessa

6) La scheda di interfaccia per attuatori/sensori (KPRN_IO) utilizza due modi di trasferimento dati contenuti nel protocollo standard 1284: il modo Compatibility per l'uscita (PC → scheda) ed il modo Byte per l'ingresso (scheda → PC).

7) Come si può notare dalle indicazioni fornite nel "Manuale d'uso e documentazione" (pag. 5), il convertitore D/A viene abilitato tramite il segnale SEL_CS (negato); il clock per il convertitoreD/A è il segnale INIT (WR_DA), mentre la tensione di uscita del convertitore viene applicata ad un amplifictore operazionale con guadagno variabile

8) Indipendentemente dal tipo di linguaggio utilizzato per comandare la scheda di interfaccia deve essere rispettata questa sequenza operativa:

Output di un dato analogico sulla scheda di interfaccia

passo	*descrizione*	*operazione*
1	Inizializzazione porta e abilitazione convertitore D/A	Registro di controllo=4H (wr_da=1, sel_cs=0, direzione=1)
2	Dato sul bus dati porta	Dato presente in DB0-DB7
3	Impulso di clock per convertitore D/A	Registro di controllo=0H (wr_da=0, sel_cs=0, direzione=1) Registro di controllo=4H (wr_da=1, sel_cs=0, direzione=1)
4	Dato analogico presente su uscita D/A	Nota: il valore in uscita dipende dalla Vref del convertitore D/A e dal guadagno dell'amplificatore operazionale

SCHEDA MULTI I/O DI INTERFACCIA PER ATTUATORI/SENSORI

SU PORTA PARALLELA EPP (STANDARD 1284)

ESERCITAZIONE N.4

Indirizzi/bit

	LPT1	*LPT2*	*bit*
Indirizzo porta	378H	37AH	DB0-DB7
Indirizzo registro di controllo	278H	27AH	bit 2=clock D/A (wr_da) bit3=abilitazione D/A (sel_cs=0) bit 5=direzione (0=output, 1=input)

NOTA: Il segnale SEL_CS (bit 3 del registro di controllo) è negato: se SEL_CS=0 si trova sul pin 13 dell'integrato 74HC14 un valore pari a 1 e quindi sul pin 12 dello stesso integrato un valore pari a 0. In questo modo il convertitore D/A è abilitato.

Esempio di programma in C++ per uscita analogica
(output continuo del dato con impostazione valore)

```
//EPP3.CPP
//uscita analogica

#include <iostream.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>

main()
{
int porta=0x378;
 int reg_c=0x37A;
 int datout;
 clrscr();
 gotoxy(10,8);
 cout<<"Tensione su uscita analogica";

                do {
                        outport(reg_c,0x4);                 // inizializzazione porta
                        gotoxy(10,10);
                        cout<<"dato decimale (<256) : [   ]";
                        gotoxy(34,10);
                        cin>>datout;
                        outport(porta,datout);              // uscita dato sul bus dati DB0-DB7
                        outport(reg_c,0x00);                // impulso di clock per D/A  wr_da=0, sel_cs=0, direzione=0
                        outport(reg_c,0x4);                 // impulso di clock per D/A  wr_da=1, sel_cs=0, direzione=0
                      } while(datout<256);                  // ripete ciclo se il dato inserito è < di 256

return 0;
}
```

TITOLO: ***FUNZIONAMENTO INPUT ANALOGICO***

OBIETTIVO : ***L'ESERCITAZIONE SI PROPONE DI DARE LE INDICAZIONI BASILARI PER L'ATTIVAZIONE DELL'INGRESSO ANALOGICO , FORNENDO UN SEMPLICE LISTATO DI PROVA IN LINGUAGGIO C++***

---

1) Prima di alimentare la scheda di interfaccia per attuatori/sensori su porta parallela EPP (chiamata in questa sezione più semplicemente "scheda di interfaccia") leggere e consultare il "Manuale d'uso e documentazione".

2) Collegare la scheda di interfaccia con un alimentatore stabilizzato 12V 2.5A (vedi Caratteristiche e specifiche in "Manuale d'uso e documentazione").

2) Collegare la scheda di interfaccia alla porta parallela del PC (486 o superiori) tramite il cavo fornito. In questo caso la scheda di interfaccia sarà comandata tramite la porta LPT1 (quella normalmente usata per la stampante del PC). Nel caso si volesse installare la scheda EPP (opzionale) sul proprio PC, avendo a disposizione uno slot libero, questa deve essere settata come LPT2, tramite gli appositi ponticelli a bordo della scheda EPP stessa

9) La scheda di interfaccia per attuatori/sensori (KPRN_IO) utilizza due modi di trasferimento dati contenuti nel protocollo standard 1284: il modo Compatibility per l'uscita (PC → scheda) ed il modo Byte per l'ingresso (scheda → PC).

10) Come si può notare dalle indicazioni fornite nel "Manuale d'uso e documentazione" (pag. 5), il convertitore A/D viene abilitato tramite il segnale SEL_CS; il clock per il convertitore A/D, che è di tipo seriale, è il bit 0 del bus dati (DB0/SCK), mentre il primo bit è presente sul segnale BUSY(DATA_OUT) e cioè il bit 7 dell'indirizzo 379H(LPT1) o 279H(LPT2); per avere il dato completo sono necessari 8 cicli di lettura

11) Indipendentemente dal tipo di linguaggio utilizzato per comandare la scheda di interfaccia deve essere rispettata questa sequenza operativa:

Output di un dato analogico sulla scheda di interfaccia

passo	*descrizione*	*operazione*
1	Inizializzazione clock A/D	Porta 378H o 278H =1
2	Inizializzazione porta, abilitzione A/D e disabilitazione D/A	Registro di controllo=0CH (sel_cs=1, wr_da=1,direzione=0)
3	Loop di lettura per eseguire la conversione del dato effettivamente presente sull'ingresso del convertitore A/D (la conversione viene effettuata durante lo "scaricamento" del dato precedente)	Porta 378H o 278H =1 Porta 378H o 278H =0 Ripetere per 7 volte
4	Disabilitazione A/D	Registro di controllo=04H (sel_cs=0, wr_da=1,direzione=0)
5	Abilitazione A/D	Registro di controllo=0CH (sel_cs=1,

	Il bit 7 del dato convertito all'ingresso A/D è ora presente sull'uscita del convertitore	wr_da=1,direzione=0)
6	Ciclo di lettura dato (8 volte)	▪ Lettura dato porta all'indirizzo 379H o 279H ▪ Isolamento bit 7 ▪ Assegnazione peso binario corrispondente ▪ Clock per lettura bit successivo
7	disabilitazione convertitore A/D	Registro di controllo=04H (sel_cs=0, wr_da=1,direzione=0)

Indirizzi/bit

	LPT1	*LPT2*	*bit*
Indirizzo porta	378H	278H	bit 0=clock A/D (sck)
Indirizzo registro di controllo	37AH	27AH	bit3=abilitazione A/D (sel_cs=1) bit 5=direzione (0=output, 1=input)
Indirizzo dato	379H	279H	bit 7

NOTA: Il segnale SEL_CS (bit 3 del registro di controllo) è negato: se SEL_CS=1 sul pin 5 del convertitore A/D (TLC549) si trova un valore pari a 0 e quindi il convertitore è abilitato.

```
//EPP4.CPP  (es doc)
//lettura e visualizzazione tensione su ingresso analogico
#include <iostream.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>
int main()
{
 int dato;
 int lettura;
 int offset;
 int cont;
 int porta=0x378;
 int reg_c=0x37A;
 int dato_ad=0x379;
 const float quanto=0.01953;                    // convertitore 8 bit Vref=5volt (Vref/256=0,01953 volt)
 clrscr();
 gotoxy(10,8);
 cout<<"Misura tensione su ingresso analogico";
 gotoxy(10,10);
 cout<<"valore decimale = ";
 gotoxy(10,11);
 cout<<"volt = ";
   do {
          dato=0;
          lettura=0;
          offset=128;
          outportb(reg_c,0xC);                  // inizializzazione porta e abilitzione
          outportb(porta,1);                 // inizializzazione clock
          for(cont=0;cont<7;cont++)             // ciclo di lettura di predisposizione
            {
            outportb(porta,0);
            outportb(porta,1);
            }


          outportb(reg_c,0x4);                  // disabilitazione A/D
```

```
        outportb(reg_c,0xC);                    // abilitazione A/D
        for(cont=0;cont<8;cont++)               // 8 cicli di lettura
                {
                delay(1);
                lettura=inportb(dato_ad);       // lettura dato porta 379H
                lettura=(lettura & 128);        // isolamento bit 7
                if (lettura==0) dato=dato+offset;   // assegnazione peso binario
                offset=offset/2;
                outportb(porta,0);          // clock per lettura successiva
                outportb(porta,1);
                }
        gotoxy(28,10);
        printf("%3d",dato);
        gotoxy(17,11);
        printf("%1.2f",dato*quanto);
        outportb(reg_c,0x4);                    // disabilitazione A/D

    } while(!kbhit());
return 0;
}

        gotoxy(28,10);
        printf("%3d",dato);
        gotoxy(17,11);
        printf("%1.2f",dato*quanto);
        outportb(reg_c,0x0);                    // disabilitazione convertitore A/D sel_cs=0, direzione=0
    } while(!kbhit());
return 0;
}
```

TITOLO: ***PROGRAMMA DI COLLAUDO 1***

OBIETTIVO : ***L'ESERCITAZIONE SI PROPONE LA REALIZZAZIONE DI UN PROGRAMMA DI TEST PER LA SCHEDA DI INTERFACCIA (PARTE DIGITALE E ANALOGICA) , FORNENDO UN LISTATO DI PROVA IN LINGUAGGIO C++***

IL PROGRAMMA PREVEDE UN SEMPLICE MENÙ PER LA SCELTA DELL'OPERAZIONE DESIDERATA.
LE ESERCITAZIONI PRECEDENTI SONO RACCHIUSE IN APPOSITE FUNZIONI.

```
//EPP_IO.CPP
//input digitale,output digitale,input analogico,output analogico
#include <conio.h>
#include <ctype.h>
#include <iostream.h>
#include <stdio.h>
#include <dos.h>
void inp_dig();
void out_dig();
void inp_ana();
void out_ana();
 int porta=0x378;
 int reg_c=0x37A;
int main()
{
   char cmd;
   do  {
   textcolor(3);
   clrscr();
   cout<<"****  Gestione scheda I/O su porta parallela EPP  ****\n\n";
   cout<<"\n\n    (I)nput    digitale";
   cout<<"\n\n    (O)utput   digitale";
   cout<<"\n\n    Input   (A)nalogico";
   cout<<"\n\n    Output  a(N)alogico";
cout<<"\n\n    (E)sci            ";
   cout << "\n\n\n                  Premere la prima lettera: ";
   gotoxy(48,18);
   do {
      cmd = toupper(getch());
      cout << '\n';
      switch (cmd)
      {
            case 'I': { clrscr();
                        gotoxy(5,20);
                        cout<<" premi E per terminare ";
                        inp_dig();
                        break;
                      }
            case 'O': { clrscr();
                        gotoxy(5,20);
                        cout<<" premi E per terminare ";
                        out_dig();
                        break;
                      }
            case 'A': { clrscr();
                        gotoxy(5,20);
                        cout<<" premi E per terminare ";
                        inp_ana();
                        break;
                      }
            case 'N': {
                        clrscr();
```

```
                        out_ana();
                        break;
                       }
            case 'E': break;
            default : gotoxy(48,20);cout<<"(scelta errata)\n";gotoxy(48,18);
      }
   } while (cmd != 'E');
   gotoxy(5,20);
   cout<<"<<   Premi C per continuare o un altro tasto per uscire   >>";
   cmd = toupper(getch());
   } while (cmd == 'C');
   return 0;
}
void out_dig()
{
unsigned char dato;
outportb(reg_c,0);
  gotoxy(10,8);
 cout<<"Uscita digitale";
 gotoxy(10,10);
 cout<<"dato binario : ";
 for (dato=0;!kbhit();dato++)
      {
      outportb(porta,dato);
      outportb(reg_c,1);
      outportb(reg_c,0);
      gotoxy(25,10);
      printf("%3d",dato);
      delay(1000);
      }
}
void inp_dig()
{
int datin;
gotoxy(10,8);
 cout<<"Ingresso digitale";
 gotoxy(10,10);
 cout<<"dato in input : ";
                        do
                        {
                        outport(reg_c,0x20);
                        outport(reg_c,0x22);
                        datin=inportb(porta);
                        outport(reg_c,0x20);
                        gotoxy(27,10);
                        printf("%3d",datin);
                        } while (!kbhit());
}
void out_ana()
{
int datout;
gotoxy(10,8);
 cout<<"Tensione su uscita analogica";
                do {
                        outport(reg_c,0x4);
                        gotoxy(10,10);
                        cout<<"dato decimale (<256) : [   ]\n\n";
                        gotoxy(5,20);
                        cout<<"Per uscire dal ciclo immettere un valore >256";
                        gotoxy(34,10);
                        cin>>datout;
                        outport(porta,datout);
                        outport(reg_c,0x00);
                        outport(reg_c,0x4);
                       } while(datout<256);
gotoxy(5,20);
```

```
cout<<"                 premi E per terminare ";
}
void inp_ana()
{
int dato;
 int lettura;
 int offset;
 int cont;
 int porta=0x378;
 int reg_c=0x37A;
 int dato_ad=0x379;
 const float quanto=0.01953;
 clrscr();
 gotoxy(10,8);
 cout<<"Misura tensione su ingresso analogico";
 gotoxy(10,10);
 cout<<"valore decimale = ";
 gotoxy(10,11);
 cout<<"volt = ";
   do {
            dato=0;
            lettura=0;
            offset=128;
            outportb(reg_c,0xC);
            outportb(porta,1);
            for(cont=0;cont<7;cont++)
              {
              outportb(porta,0);
              outportb(porta,1);
              }
            outportb(reg_c,0x4);
            outportb(reg_c,0xC);
            for(cont=0;cont<8;cont++)
                     {
                     delay(1);
                     lettura=inportb(dato_ad);
                     lettura=(lettura & 128);
                     if (lettura==0) dato=dato+offset;
                     offset=offset/2;
                     outportb(porta,0);
                     outportb(porta,1);
                     }
            gotoxy(28,10);
            printf("%3d",dato);
            gotoxy(17,11);
            printf("%1.2f",dato*quanto);
            outportb(reg_c,0x4);

    } while(!kbhit());
return 0;
}
            gotoxy(28,10);
            printf("%3d",dato);
            gotoxy(17,11);
            printf("%1.2f",dato*quanto);
            outportb(reg_c,0x0);
    } while(!kbhit());
}
```

TITOLO: PROGRAMMA DI COLLAUDO 2

OBIETTIVO : ***L'ESERCITAZIONE SI PROPONE LA REALIZZAZIONE DI UN PROGRAMMA DEMO PER LA SCHEDA DI INTERFACCIA (PARTE DIGITALE E ANALOGICA) , FORNENDO UN LISTATO DI PROVA IN LINGUAGGIO QUICK BASIC***

IL PROGRAMMA PREVEDE, OLTRE ALLE FUNZIONI DELL'ESERCITAZIONE PRECEDENTE, ANCHE LA SCELTA DELLA PORTA DEL PC (LPT1 / LPT2).

```
'
'********** Demo prova scheda printer bidirezionale *******
'
'
'
'
delay& = 50000        'ritardo su ciclo in / out
OUTOUT$ = CHR$(9)
USCITA$ = CHR$(27)    'Tasto di uscita dalle routine: ESC
'
CHIEDI:
          CLS
          LOCATE 5, 20: PRINT "TEST PORTA PRINTER BIDIREZIONALE"
          LOCATE 6, 20: PRINT "--------------------------------"
          LOCATE 10, 20
          INPUT " Selezionare la porta parallela: ", A$
          IF A$ = "1" THEN port.add% = &H378: GOTO CICLO
          IF A$ = "2" THEN port.add% = &H278: GOTO CICLO
          GOTO CHIEDI:
'
CICLO:
sgn.ctrl% = port.add% + 2     'Registro segnali di controllo
'                    bit 0 = caricamento uscite digitali (trans 0->1)
'                    bit 1 = abilitazione lettura ingressi (attivo 0)
'                    bit 2 = scrittura convertitore DA
'                    bit 3 = selezione AD / DA          (0=DA 1=AD)
'                    bit 5 = direzione bus dati         (0=OUT 1=IN)
'
input.ad% = port.add% + 1     'Registro di lettura AD
'                    bit 7 = uscita dato seriale AD
'
CHIEDI2:
                              CLS
                              LOCATE 9, 20: PRINT "1) Scrittura output digitali"
                              LOCATE 10, 20: PRINT "2) Lettura input digitali"
                              LOCATE 11, 20: PRINT "3) Test output D/A"
                              LOCATE 12, 20: PRINT "4) Test input A/D"
                              LOCATE 13, 20: PRINT "5) Loop In/Out analogici"
                              LOCATE 14, 20: PRINT "6) Impostazione ritardo ciclo"
                              LOCATE 18, 20: PRINT "ESC: esce dal programma"
CHIEDI1:
                              RISPO$ = INKEY$
                              IF RISPO$ = "" THEN GOTO CHIEDI1
                              IF RISPO$ = USCITA$ THEN SYSTEM
                              IF RISPO$ = "1" THEN GOSUB OUT.DIG: GOTO CHIEDI2
                              IF RISPO$ = "2" THEN GOSUB IN.DIG: GOTO CHIEDI2
                              IF RISPO$ = "3" THEN GOSUB DA.TEST: GOTO CHIEDI2
                              IF RISPO$ = "4" THEN GOSUB AD.TEST: GOTO CHIEDI2
                              IF RISPO$ = "5" THEN GOSUB LOOP.TEST: GOTO CHIEDI2
                              IF RISPO$ = "6" THEN GOSUB ritardo: GOTO CHIEDI2
                              GOTO CHIEDI1
'
```

```
'------------------------------------------------------------------------------
' Routine OUT.DIG: Attivazione output digitali
'
OUT.DIG:
                        CLS :
                        LOCATE 15, 20: PRINT "ESC: uscita dalla funzione"
                        CNTOUT% = 0
                        OUT sgn.ctrl%, &H4
'
OUT.DIG1:
                        LOCATE 10, 20: PRINT "DATO IN OUTPUT: "; CNTOUT%; "   "
'
                        OUT port.add%, CNTOUT%      'Out dato
                        OUT sgn.ctrl%, &H5          'Caricamento dato
                        OUT sgn.ctrl%, &H4
'
                        CNTOUT% = CNTOUT% + 1
                        IF CNTOUT% = 256 THEN CNTOUT% = 0
'
                        A$ = INKEY$: IF A$ = USCITA$ THEN RETURN
                        FOR I = 1 TO delay&
                        NEXT I
'
                        GOTO OUT.DIG1
'
'------------------------------------------------------------------------------
' Routine IN.DIG: Lettura input digitali
'
IN.DIG:
                        CLS
                        LOCATE 15, 20: PRINT "ESC: uscita dalla funzione"
                        OUT sgn.ctrl%, &H24         'bus dati in input, driver disabilitato
'
IN.DIG1:
                        OUT sgn.ctrl%, &H26                 'abilito driver
                        LOCATE 10, 20:
                        PRINT "DATO IN INPUT: "; INP(port.add%)
                        OUT sgn.ctrl%, &H24                 'disabilito driver
'
                        A$ = INKEY$: IF A$ = USCITA$ THEN RETURN
'
                        FOR I = 1 TO delay&
                        NEXT I
'
                        GOTO IN.DIG1
'
'-------------------------------------
' Routine DA.TEST: test output digitale/analogico
'
DA.TEST:
                        CLS
                        LOCATE 15, 20: PRINT "ESC: uscita dalla funzione"
                        CNTOUT% = 0
'
                        OUT sgn.ctrl%, &H4
'
DA1:
                        LOCATE 10, 20: PRINT "DATO IN OUTPUT: "; CNTOUT%; "   "
'
                        OUT port.add%, CNTOUT%      'Out dato
                        OUT sgn.ctrl%, &H0          'Caricamento dato su DA
                        OUT sgn.ctrl%, &H4
'
                        CNTOUT% = CNTOUT% + 1
                        IF CNTOUT% = 256 THEN CNTOUT% = 0
'
```

```
                        A$ = INKEY$: IF A$ = USCITA$ THEN RETURN
'
                        FOR I = 1 TO delay&
                        NEXT I
'
                        GOTO DA1
'
'-------------------------------------
' Routine AD.TEST: test input analogico/digitale
'
AD.TEST:
                        CLS
                        LOCATE 7, 20: PRINT "Tensione 0ö5 Vcc / Scala 0ö255"
                        LOCATE 8, 20: PRINT "------------------------------"
                        LOCATE 15, 20: PRINT "ESC: uscita dalla funzione"
'
AD1:
                        DATO% = 0:  OFSET% = 128
'
                        OUT sgn.ctrl%, &HC          'Abilito convertitore AD
                        FOR n% = 1 TO 100: NEXT n%   'Delay necessario su PC veloci
'
' Leggo il bit piu' significativo (bit 7) gia' presente in input appena
'  abbassato il CS (SEL.CS).
'
                        LETTURA% = INP(input.ad%)
                        LETTURA% = LETTURA% AND 128
                        IF LETTURA% = 0 THEN DATO% = DATO% + OFSET%
                        OFSET% = OFSET% / 2
'
' Faccio la sequenza per 7 volte sul segnale I/OCLK per leggere gli altri
' bit (sequenza bit 6-->bit 0)
'
                        FOR I% = 1 TO 7
                                OUT port.add%, 0
                                OUT port.add%, 1
                                LETTURA% = INP(input.ad%)
                                LETTURA% = LETTURA% AND 128
                                IF LETTURA% = 0 THEN DATO% = DATO% + OFSET%
                                OFSET% = OFSET% / 2
                        NEXT I%
'
                        LOCATE 10, 20: PRINT "IN ANALOGICO:  "; DATO%; "   "
                        LOCATE 10, 42: PRINT "(valore decimale)"
                        VOLT! = DATO% * .01953      'V/bit
                        LOCATE 12, 20: PRINT "VALORE TENSIONE: "; VOLT!; "Vcc    "
'
                        OUT sgn.ctrl%, &H4
'
                        A$ = INKEY$: IF A$ = USCITA$ THEN RETURN
'
                        FOR I = 1 TO delay&
                        NEXT I
'
                        GOTO AD1
'
'-------------------------------------
' Routine LOOP.TEST: Loop input/output analogici
'
LOOP.TEST:
                        CLS
                        LOCATE 7, 20: PRINT "Tensione 0ö5 Vcc / Scala 0ö255"
                        LOCATE 8, 20: PRINT "------------------------------"
                        LOCATE 17, 25: PRINT "ESC: uscita dalla funzione"
'
LOOP1:
```

```
                        OUT sgn.ctrl%, &H4
                        OUT port.add%, CNTOUT%      'Out dato
                        OUT sgn.ctrl%, &H0          'Carico dato su DA
                        OUT sgn.ctrl%, &H4
'
                        DATO% = 0:  OFSET% = 128
'
                        OUT port.add%, 1
                        OUT sgn.ctrl%, &HC          'Abilito convertitore AD
                        FOR n% = 1 TO 100: NEXT n%  'Delay necessario su PC veloci
'
                        FOR I% = 1 TO 8             'ciclo di conversione
                                OUT port.add%, 0
                                OUT port.add%, 1
                        NEXT I%
'
                        OUT sgn.ctrl%, &H4          'Disabilito convertitore AD
                        OUT sgn.ctrl%, &HC          'Abilito convertitore AD
                        FOR n% = 1 TO 100: NEXT n%
'
' Leggo il bit piu' significativo (bit 7) gia' presente in input appena
' abbassato il CS (SEL.CS).
'
                        LETTURA% = INP(input.ad%)
                        LETTURA% = LETTURA% AND 128
                        IF LETTURA% = 0 THEN DATO% = DATO% + OFSET%
                        OFSET% = OFSET% / 2
'
' Faccio la sequenza per 7 volte sul segnale I/OCLK per leggere gli altri
' bit (sequenza bit 6-->bit 0)
'
                        FOR I% = 1 TO 7
                                OUT port.add%, 0
                                OUT port.add%, 1
                                LETTURA% = INP(input.ad%)
                                LETTURA% = LETTURA% AND 128
                                IF LETTURA% = 0 THEN DATO% = DATO% + OFSET%
                                OFSET% = OFSET% / 2
                        NEXT I%
'
                        LOCATE 10, 20: PRINT "OUT ANALOGICO: "; CNTOUT%; "   "
                        LOCATE 11, 20: PRINT "IN  ANALOGICO: "; DATO%; "   "
                        LOCATE 11, 42: PRINT "(valore decimale)"
                        VOLT! = DATO% * .01953      'V/bit
                        LOCATE 13, 20: PRINT "VALORE TENSIONE: "; VOLT!; "Vcc    "
'
                        OUT sgn.ctrl%, &H4          'Disabilito convertitore AD
'
                        CNTOUT% = CNTOUT% + 1
                        IF CNTOUT% = 256 THEN CNTOUT% = 0
'
                        A$ = INKEY$: IF A$ = USCITA$ THEN RETURN
'
                        FOR I = 1 TO delay&
                        NEXT I
'
                        GOTO LOOP1
'
'-------------------------------------
' Routine RITARDO: impostazione ritardo ciclo di test
'
ritardo:
                        CLS
                        LOCATE 10, 25
                        PRINT "VALORE RITARDO ATTUALE: "; delay&
                        LOCATE 14, 25
```

```
                        LINE INPUT "INSERIRE NUOVO VALORE ---> "; delay1$
'
                        IF delay1$ = "" THEN RETURN
                        IF VAL(delay1$) < 100 OR VAL(delay1$) > 1000000 THEN GOTO ritardo
                        delay& = VAL(delay1$)
'
                        RETURN
'
'-----------------------------------
        END
```