

<i>INTRODUZIONE</i>	2
<i>Segnali di Controllo provenienti dal Bus del PC</i>	3
<i>Segnali di Controllo presenti sulla scheda</i>	3
<i>Altri segnali presenti sulla scheda</i>	3
<i>ESEMPIO DI PROGETTO DI INTERFACCE</i>	4
<i>Interfaccia relè</i>	4
<i>Interfaccia display a sette segmenti decodificati</i>	5
<i>Interfaccia Led e Dip Switch</i>	5
<i>CONTROLLO E SINCRONIZZAZIONE DELLE VARIE INTERFACCE</i>	6
<i>Assegnazione degli indirizzi</i>	6
<i>SOFTWARE DI GESTIONE</i>	6
<i>Bibliografia</i>	8
<i>Appendice A : Bus di Espansione PC XT (8 bit)</i>	9

Interfacciamento con PC/XT/AT progetto HDW e SFTW

prof. Cleto Azzani
 IPSIA Moretto Brescia

Febbraio 1995

Introduzione

INTERFACCIA SPERIMENTALE PER PC/XT/AT

NUOVA ELETTRONICA LX833

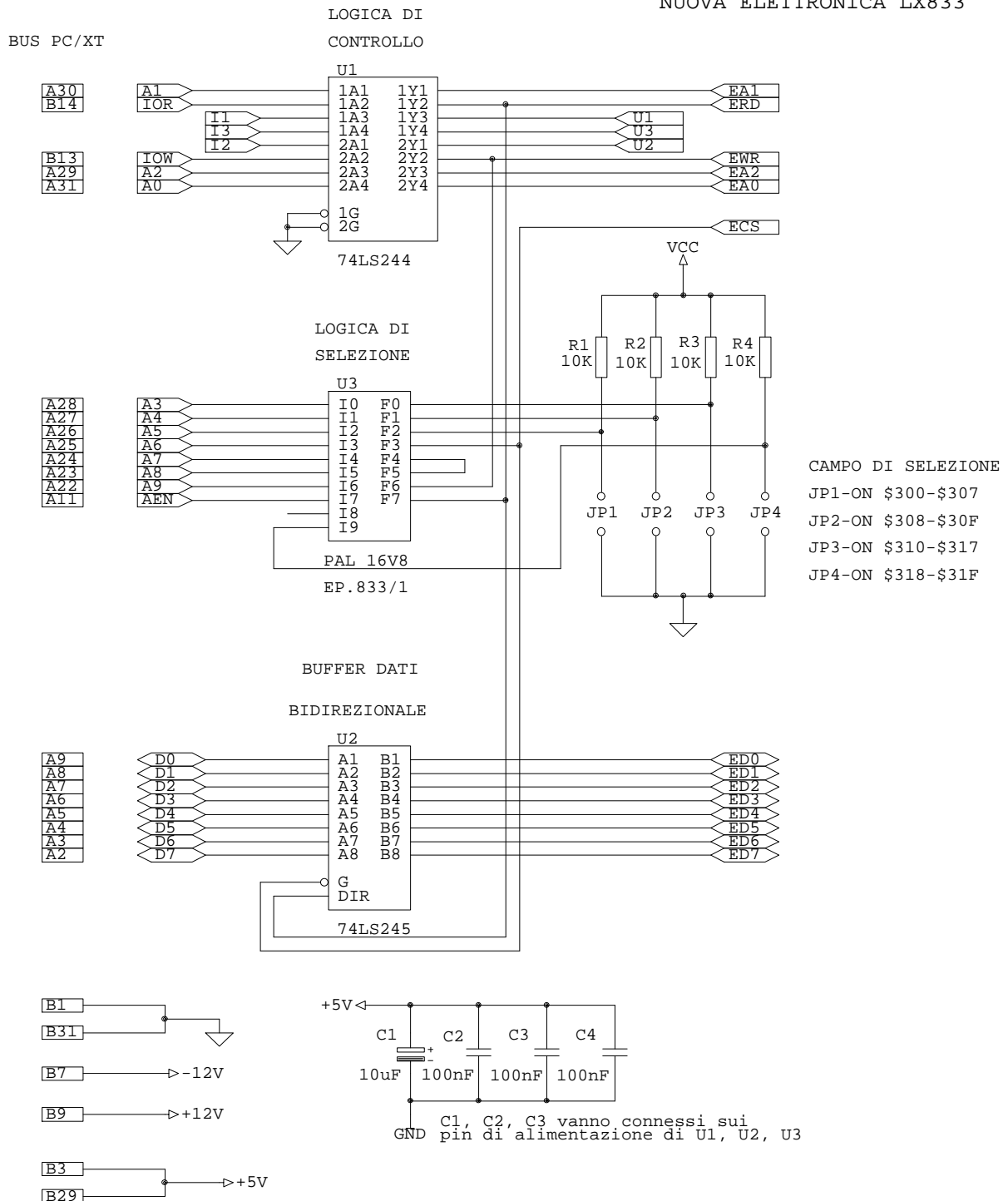


fig. 1 Circuito scheda LX833 Nuova Elettronica

In fig.1 è riportato lo schema elettrico dell'interfaccia sperimentale LX833 realizzata da Nuova Elettronica e adatta per essere inserita negli slot a 8 bit di un Personal Computer IBM compatibile classe XT oppure AT. Il circuito LX833 utilizza tre circuiti integrati cui sono affidati i seguenti compiti:

U1 - 74LS244 buffer per amplificazione di indirizzi e segnali di controllo

Cleto Azzani

U3 - 74LS245 buffer bidirezionale per dati

U2 - PAL 16V8 (programmable array logic) decodifica indirizzi

Il circuito preleva direttamente dal BUS del PC i segnali presenti sul Data-Bus (D0-D7) sull'Address-Bus (A0-A9) i segnali di controllo: IOR, IOW, AEN e le tensioni di alimentazione (+5V, +12V, -12V) necessarie per la alimentazione sia dei circuiti integrati già presenti nel circuito, sia dei circuiti integrati da collocare in area wrap (area a disposizione dell'utente). La scheda è provvista anche da una serie di jumper (ponticelli) attraverso i quali si stabilisce l'area di indirizzamento più opportuna della scheda LX833.

Un solo jumper alla volta può essere chiuso. Poichè in ingresso alla PAL (U2) giungono tutti gli indirizzi meno i tre meno significativi (A0, A1, A2), ne deriva che l'abilitazione si attiverà su 8 indirizzi di I/O contigui come riportato in tab. 1.

Jumper	CAMPO
JP1	\$300-\$307
JP2	\$308-\$30F
JP3	\$310-\$317
JP4	\$318-\$31F

Segnali di Controllo provenienti dal Bus del PC

AO-A1-A2: rappresentano i primi tre bit del bus degli indirizzi.

IOR (Input Output Read): IOR, significa "lettura da un dispositivo di ingresso/uscita". Quando la CPU presente nel PC intende effettuare una operazione READ da un dispositivo di I/O pone a livello logico basso il segnale IOR.

IOW (Input Output Write): IOW, significa "scrittura verso un dispositivo di ingresso/uscita". Quando la CPU presente nel PC intende effettuare una operazione WRITE verso un dispositivo di I/O pone a livello logico basso il segnale IOW.

AEN (Address Enable): AEN significa "abilitazione all'Address Bus". Quando la CPU presente nel PC emette un livello logico basso su AEN ciò significa che l'indirizzo esistente in quell'istante sull'Address Bus è da considerarsi valido.

Segnali di Controllo presenti sulla scheda

ECS (Expansion chip select): ECS significa "selezione della scheda di espansione". Lo scopo di questo segnale è di fornire l'abilitazione agli elementi circuitali presenti sulla scheda, quando sul bus degli indirizzi è presente uno degli indirizzi contenuto entro il gruppo selezionato dal jumper chiuso (vedi tab. 1). Il segnale ECS è attivo a livello logico basso.

Altri segnali presenti sulla scheda

ED0-ED7 rappresentano il Data-Bus bufferizzato D0-D7 a valle di U3 SN74LS245 (transceiver o buffer bidirezionale).

EA0-EA2 versione bufferizzata degli indirizzi bassi A0-A3 a valle di U1 SN74LS244

ERD versione bufferizzata del segnale IOR a valle di U1 SN74LS244

EWR versione bufferizzata del segnale IOW a valle di U1 SN74LS244

Esempio di progetto di interfacce

Il circuito si propone l'obiettivo di realizzare un'interfaccia sperimentale in grado di gestire e comandare una serie di dispositivi elettronici mediante l'utilizzo di un Personal Computer IBM compatibile fatto funzionare con un software di gestione redatto in TURBO PASCAL.

I dispositivi prescelti da interfacciare sono:

- gruppo di 4 relè reed,
- gruppo di 2 display a sette segmenti decodificati e pilotati da decoder driver 9368,
- gruppo di 8 diodi led,
- gruppo di 8 dip-switch.

Data la complessità dell'interfaccia si è deciso di montare su un apposito circuito stampato a parte (diverso dal circuito LX833) l'hardware relativo ai dispositivi elencati da a) a d).

Esaminiamo uno ad uno i circuiti relativi alle quattro interfacce da realizzare.

Interfaccia relè

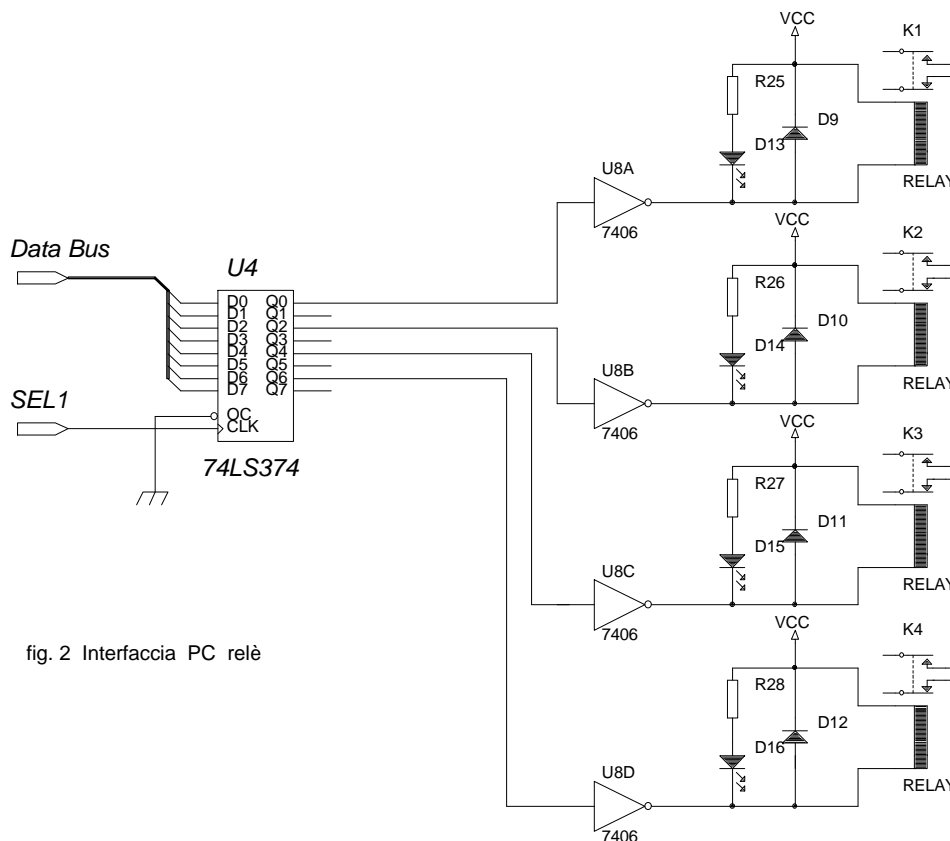


fig. 2 Interfaccia PC relè

In fig. 2 è riportato lo schema elettrico dell'interfaccia verso relè reed. In essa si nota la presenza di un registro a 8 bit 74LS374 U4 che pilota sulle uscite Q0, Q2, Q4, Q6 quattro buffer invertenti 7406 che a loro volta pilotano altrettanti relè reed a 5V. Si osservi che ogni relè reed è dotato di diodo volano per i noti problemi esistenti nel pilotaggio dei carichi induttivi. Si noti che l'attivazione di un relè è pure visualizzata attraverso un diodo led con relativa resistenza di limitazione posto in parallelo alla bobina di eccitazione. Il circuito integrato 74LS374 riceve i dati provenienti dal Data Bus della scheda LX833 ED0-ED7 e li memorizza sul fronte di salita del suo ingresso di clock alimentato dal segnale SEL1.

Per attivare il relè K1 (connesso alla uscita Q0) il PC dovrà inviare al registro U4 il codice binario %00000001 (\$01), per attivare K2 il PC dovrà inviare il codice binario %00000010 (\$02), per attivare contemporaneamente i relè K1, K3 e K4 il PC dovrà inviare il codice binario %01010001 (\$51). Il segnale

SEL1 dovrà essere generato da una operazione WRITE eseguita all'indirizzo attribuito all'interfaccia nell'ambito di quelli previsti dalla tabella 1.

Interfaccia display a sette segmenti decodificati

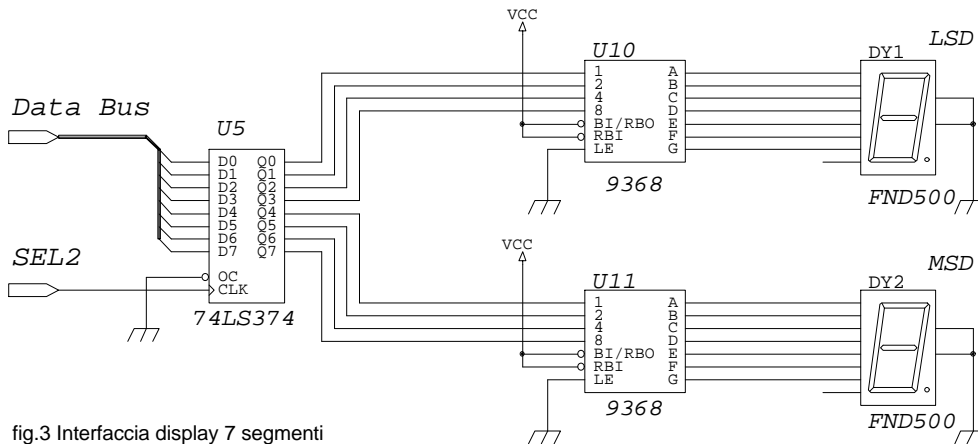


fig.3 Interfaccia display 7 segmenti

In fig. 3 è riportato lo schema elettrico dell'interfaccia verso display a 7 segmenti decodificati. In essa si nota la presenza di un registro a 8 bit 74LS374 U5 che pilota due decoder TTL 9368 connessi a due display a 7 segmenti catodo comune del tipo FND500. Il codice binario relativo ai 4 bit D0-D3 attiva il display LSD (cifra meno significativa) il codice relativo ai 4 bit D4-D7 attiva il display MSD (cifra più significativa). Le due decodifiche 9368 sono mantenute trasparenti (LE connesso a 0); la funzione di memorizzazione è affidata al circuito integrato U5 che come è noto memorizza i dati presenti sugli ingressi D0-D7 sul fronte di salita dell'ingresso di clock alimentato dal segnale SEL2.

Per trasferire al gruppo display il dato binario %01010010 (\$52) il PC dovrà inviare al registro U5 il codice binario %01010010 o esadecimale \$52. Sul display MSD apparirà la cifra 5 sul display LSD apparirà la cifra. Il segnale SEL2 dovrà essere generato da una operazione WRITE eseguita all'indirizzo attribuito all'interfaccia nell'ambito di quelli previsti dalla tabella 1.

Interfaccia Led e Dip Switch

In fig. 4 è riportato lo schema elettrico dell'interfaccia verso 8 led e 8 dip switch. In essa si nota la presenza di un registro a 8 bit 74LS374 U6 che pilota 8 led corredati della appropriata resistenza di limitazione della corrente. Nel circuito è pure presente un transceiver 74LS245 U7 che consente la lettura dello stato logico predisposto dal gruppo di dip switch P1-P8 alimentati dalle otto resistenze di pull-up. Si osservi che avendo posto l'ingresso DIR di U7 a livello 0, il trasferimento dati avviene nel transceiver da B verso A.

Il segnale SEL3 (fronte di salita per clock di U6) dovrà essere generato da una

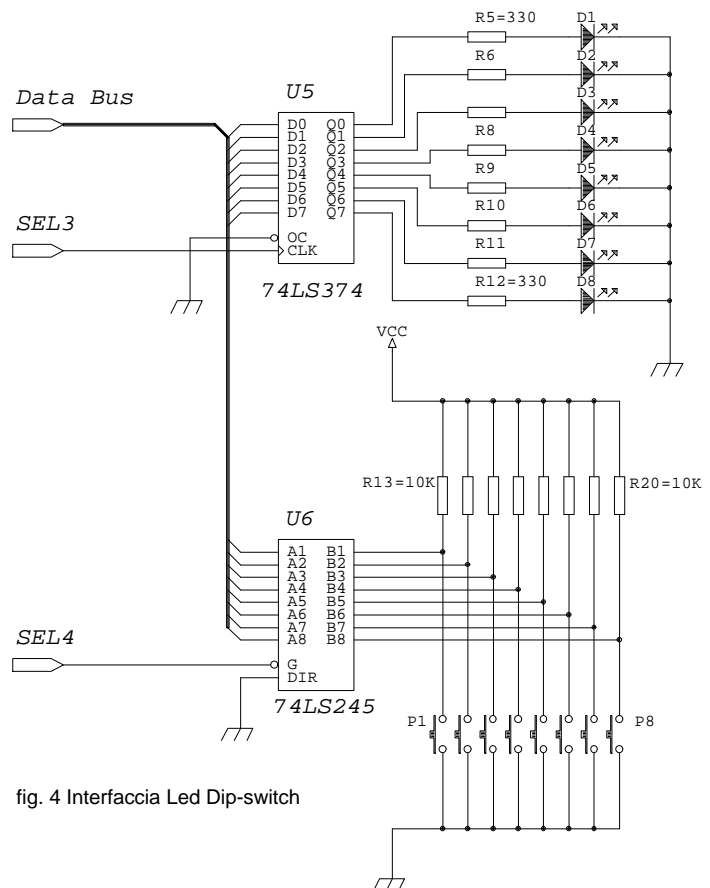


fig. 4 Interfaccia Led Dip-switch

operazione WRITE eseguita all'indirizzo attribuito all'interfaccia Led nell'ambito di quelli previsti dalla tabella 1. Il segnale SEL4 (livello basso per ingresso G di U7) dovrà essere generato da una operazione READ eseguita all'indirizzo attribuito all'interfaccia Dip-Switch nell'ambito di quelli previsti dalla tabella 1.

Controllo e sincronizzazione delle varie interfacce

La necessità di interconnettere i circuiti di fig. 2, 3 e 4 al circuito di fig. 1 (LX833) impone di suddividere i circuiti in due gruppi:

1) dispositivi verso i quali il PC effettua operazioni di tipo READ (interfaccia dip-switch)

2) dispositivi verso i quali il PC effettua operazioni di tipo WRITE (gruppo relè, gruppo display, gruppo led).

Una soluzione al problema può, in linea di principio essere quella di fig. 6. In essa sono presenti due decoder 74LS139 uno dedicato a gestire fino ad un massimo di otto dispositivi funzionanti in modalità READ e l'altro dedicato a gestire fino ad un massimo di otto dispositivi funzionanti in modalità WRITE. Ciascun decoder ha l'ingresso di abilitazione G2B connesso al segnale ECS (abilitazione scheda), l'ingresso di abilitazione G2A è connesso al segnale ERD nel caso del decoder U8 (gestione interfacce READ) al segnale EWR nel caso del decoder U9 (gestione interfacce WRITE).

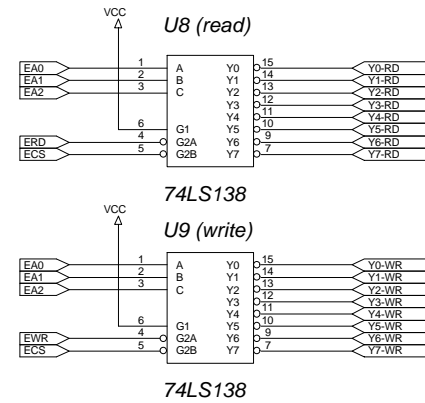


fig. 6 Circuitria di selezione

Assegnazione degli indirizzi

Ipotizzando ora di assegnare gli indirizzi come presentato in tabella bisognerà da ultimo rendere la logica di funzionamento dell'interfaccia compatibile con i segnali forniti dal sistema di decodifiche U8 e U9. Le uscite dei decoder 74LS138 sono attive a livello basso per cui date le premesse SEL1, SEL2 e SEL3 dovranno essere ricavati negando con una porta 74LS04 i segnali rispettivi Y2-WR, Y3-WR e Y0-WR mentre SEL4 dovrà essere collegato direttamente a Y1-RD.

Un altro esempio di circuitria di assegnazione degli indirizzi viene presentata in fig. 7 in tal caso anziché utilizzare due circuiti integrati 74LS138 si utilizza un solo circuito integrato 74LS139 (decodifica doppia) con ovvia conseguente riduzione dei costi.

Interfaccia	Selezione	Indirizzo	Modo	Connessione
Relè	SEL1	302	WR	Y2-WR
Display	SEL2	303	WR	Y3-WR
Led	SEL3	300	WR	Y0-WR
Dip-Switch	SEL4	301	RD	Y1-RD

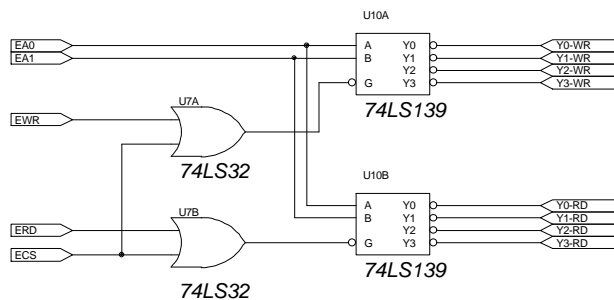


fig. 7 Altro circuito di selezione

Software di Gestione

Viene ora riportato il listato sorgente del programma Turbo Pascal redatto dagli alunni Consadori, Giusti, Romano, Scalvini della 5AI 1990-91 IPSIA Moretto Brescia con il quale è stato concretamente collaudata la funzionalità delle interfacce descritte. Il listato è sufficientemente commentato e non dovrebbe dare grossi problemi di comprensione.

```

PROGRAM INTERFACCIA(INPUT,OUTPUT);
USES CRT;
VAR
    K : INTEGER;
    ADDR, LED, DIP, RELE, DISPLAY:WORD;

Procedure PLED; { Programma di gestione diodi led }
var
    N:array[1..41] of byte; { Apertura di un vettore N }
    K: integer;           { Dichiarazione variabile contatore }
Begin
    { I Led si accendono convergendo dagli estremi verso il centro }
    N[1]:=$00;           N[2]:=$81;           N[3]:=$42;           N[4]:=$24;
    N[5]:=$18;           N[6]:=$24;           N[7]:=$42;           N[8]:=$81;
    N[9]:=$00;

    { Un solo Led acceso partendo da L8 verso L1 }
    N[10]:=$80;          N[11]:=$40;          N[12]:=$20;          N[13]:=$10;
    N[14]:=$08;          N[15]:=$04;          N[16]:=$02;

    { Accensione progressiva dei Led da destra verso sinistra riempimento progressivo del registro }
    N[17]:=$01;          N[18]:=$03;          N[19]:=$07;          N[20]:=$0F;
    N[21]:=$1F;          N[22]:=$3F;          N[23]:=$7F;          N[24]:=$FF;

    { Un solo Led acceso alternativamente prima L8 quindi L1 poi L7 e quindi L2 e così via }
    N[25]:=$00;          N[26]:=$80;          N[27]:=$01;          N[28]:=$40;
    N[29]:=$02;          N[30]:=$20;          N[31]:=$04;          N[32]:=$10;
    N[33]:=$08;          N[34]:=$10;          N[35]:=$04;          N[36]:=$20;
    N[37]:=$02;          N[38]:=$40;          N[39]:=$01;          N[40]:=$80;
    N[41]:=$FF;

    port[display]:=$00;
    port[rele]:=$00;
    For K:=1 to 41 do
        Begin
            Port[LED]:=N[K];           { Operazione di scrittura sui led }
            delay(200);                { Ritardo 200ms }
            Gotoxy(11,19); write(K);{ Posizionamento del cursore coordinate 11,19 }
            Gotoxy(7,15); write('Questa procedura è in grado di fare accendere');
            Gotoxy(7,16); write('8 diodi led in vari tipi di sequenza');
        End;
    End;

Procedure PDIP; { Programma di gestione dip-switch
                legge il dato presente sul dip-switch e poi lo trasferisce sul display a 7 segmenti }
var
    J:byte;
Begin
    Gotoxy(7,15); write('Questa procedura permette di visualizzare');
    Gotoxy(7,16); write('sul display il valore impostato sul dip-switch');
    repeat
        { Inizio ciclo ripetitivo }
        J:=port[dip]; { Operazione di lettura dai dip-switch }
        J:=255-J;     { Conversione da logica negativa a positiva }
        port[display]:=J; { Operazione di scrittura sui display }
        Gotoxy(11,19); write(J:4);
    until false; { Condizione di loop infinito }
End;

Procedure PRELE; { Programma di gestione dei relè }
var
    dato:byte;
    H:byte;
Begin
    dato:=0;
    port[rele]:=$00; { Spegnimento dei rele }
    port[led]:=$00;  { Spegnimento dei led }
    K:=port[dip];    { Lettura del valore impostato sul dip-switch }
    K:=255-K;
    Gotoxy(11,19); write(K);
    Gotoxy(7,15); write('Questa procedura permette di azionare i rele');
    Gotoxy(7,16); write('in sequenza ogni 16 conteggi visualizzati dal');
    Gotoxy(7,17); write('display partendo dal valore 03');
    readln;
    repeat

```

```

Gotoxy(19,19);write(dato:4);
port[display]:=dato; { Scrittura del dato sul display }
If (dato mod 16)=3 then begin { Questa istruzione permette di attivare } { un rele ogni 16 impulsi }
    H:=(((dato div 16)+1) mod 4)+1; { Conversione del risultato intero }
                                { di dato/16 in base 4 per attivare il rele corrispondente }
    Gotoxy(27,19);write('rele: ',(((h+2) mod 4)+1):2);
    case H of
        1 : port[rele]:=$40; { Indirizzo corrispondente al rele 1 }
        2 : port[rele]:=$01;
        3 : port[rele]:=$10;
        4 : port[rele]:=$04;
    end;
end;
dato:=dato+1;
delay(500);
until dato=K+1;
End;

```

Procedure PDISPLAY;

```

var
    dato,K : byte;
Begin
    dato:=0;
    K:=port[dip];
    K:=255-K;
    Gotoxy(11,19); write(K);
    Gotoxy(7,15); write(' Questa procedura permette di fare ');
    Gotoxy(7,16); write(' avanzare i display fino al numero ');
    Gotoxy(7,17); write(' fissato sui dip-switch ');
    readln;
    repeat
        Gotoxy(19,19);write(dato);
        port[display]:=dato;
        dato:=dato +1;
        delay(500);
    until dato=K+1;
End;

```

```

BEGIN { Programma Principale MAIN }
ClrScr;
ADDR:=$300; { $300-$307 campo di indirizzamento della scheda }
LED:=ADDR;
DIP:=ADDR+$1;
RELE:=ADDR+$2;
DISPLAY:=ADDR+$3;
Repeat
    ClrScr;
    Gotoxy(10,3); write(' 1- GESTIONE LED ');
    Gotoxy(10,5); write(' 2- GESTIONE DISPLAY');
    Gotoxy(10,7); write(' 3- GESTIONE RELE');
    Gotoxy(10,9); write(' 4- GESTIONE DIP-SWITCH'); Gotoxy(10,11); write(' 0- FINE');
    Gotoxy(11,13); readln(K);
    Case K of
        1:PLED;
        2:PDISPLAY;
        3:PRELE;
        4:PDIP;
    End;
until K=0;
END.

```

Bibliografia

Interfaccia sperimentale per PC/XT/AT NE LX833 (tesina per esame di maturità)
redatta da Consadori,Giusti, Romano,Scalvini. 5AI 1990-91 IPSIA Moretto Brescia

Appendice A : Bus di Espansione PC XT (8 bit)

Massa	GND		B1	A1	I/O CH CK	L I/O Channel check
Reset	RESET DRV		B2	A2	D7	MSB Dati
Alimentazione	+5V		B3	A3	D6	
Richiesta di Interrupt 2	IRQ2		B4	A4	D5	
Alimentazione	-5V		B5	A5	D4	
Richiesta DMA2	DRQ2		B6	A6	D3	
Alimentazione	-12V		B7	A7	D2	
0 stati di attesa	OVS		B8	A8	D1	
Alimentazione	+12V		B9	A9	D0	LSB Dati
Massa	GND		B10	A10	I/O CH RDY	
Scrittura in Memoria	MEMW	L	B11	A11	AEN	L Abilitazione Indirizzi
Lettura in Memoria	MEMR	L	B12	A12	A19	MSB Indirizzi
Scrittura I/O	IOW	L	B13	A13	A18	
Lettura I/O	IOR	L	B14	A14	A17	
Riconoscimento richiesta DMA3	DACK3	L	B15	A15	A16	
Richiesta DMA3	DRQ3		B16	A16	A15	
Riconoscimento richiesta DMA1	DACK1	L	B17	A17	A14	
Richiesta DMA1	DRQ1		B18	A18	A13	
Refresh memoria	REFRESH	L	B19	A19	A12	
Clock da scheda madre	CLK		B20	A20	A11	
Richiesta di Interrupt 7	IRQ7		B21	A21	A10	
Richiesta di Interrupt 6	IRQ6		B22	A22	A9	
Richiesta di Interrupt 5	IRQ5		B23	A23	A8	
Richiesta di Interrupt 4	IRQ4		B24	A24	A7	
Richiesta di Interrupt 3	IRQ3		B25	A25	A6	
Riconoscimento richiesta DMA2	DACK2	L	B26	A26	A5	
Impulso cont. fine DMA	T/C		B27	A27	A4	
Buffer Address Latch Enable	BALE		B28	A28	A3	
Alimentazione	+5V		B29	A29	A2	
Clock a 14.3181 MHz	OSC		B30	A30	A1	
Massa	GND		B31	A31	A0	LSB Indirizzi