

# DISPOSITIVI DI COLLEGAMENTO AL MICROPROCESSORE

## Capitolo IX del Testo

### Corso di Elettronica 3

Ettore Panella – Giuseppe Spalierno

Edizioni Cupido

## 1. Generalità

Quando si vogliono collegare tra loro apparecchi di diversa natura, ad esempio dispositivi logici di famiglie diverse, trasduttori a dispositivi digitali, amplificatori ed attuatori, ecc. è necessario prevedere opportune soluzioni *hardware* e *software* in grado di ottimizzare il collegamento. Ciò avviene attraverso le *interfacce*.

Col termine interfaccia si intende l'architettura di collegamento tra le parti di un sistema che consente il trasferimento delle informazioni.

L'interfaccia si può realizzare:

- 1) tra componenti di una stessa scheda;
- 2) tra scheda e scheda;
- 3) tra apparecchiature differenti.

Nel primo caso bisogna assicurare la compatibilità tra i livelli di tensione, di corrente e le frequenze in gioco. Nel secondo caso bisogna minimizzare i fenomeni transitori negativi prodotti dalle induttanze e capacità parassite a causa del collegamento con connettori e cavi. Nel terzo caso i problemi da affrontare riguardano la diversa filosofia di funzionamento delle apparecchiature, diverso significato logico dei bit, linee di clock aggiuntive, ecc.

Un'interfaccia deve essere:

- 1) *flessibile*, cioè deve essere facilmente riconfigurabile;
- 2) *compatibile* in modo da poter collegare direttamente le apparecchiature;
- 3) *affidabile* nei confronti del funzionamento hardware, della bontà del software e della corretta trasmissione dei dati.

## Esempio 1

Si vuole pilotare un diodo LED in logica negativa tramite un NOT TTL LS come in fig.1. Sapendo che per la corrente:  $I_{OL} = 10\text{mA}$  il LED presenta una caduta di

tensione ai suoi capi:  $V_L = 1.8V$  e la porta logica:  $V_{OL} = 0.2V$ , determinare il valore della resistenza  $R$ .

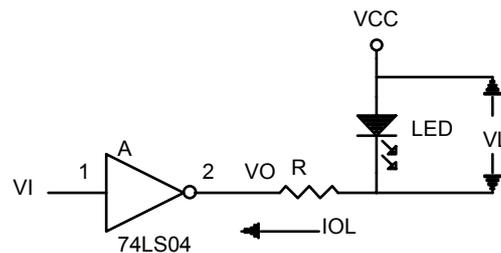


Fig. 1. - Pilotaggio di un diodo LED in logica negativa.

(Esempio di interfacciamento tra componenti di una stessa scheda e cioè tra la porta logica ed il diodo LED. L'elemento di interfaccia, in questo caso, è la resistenza  $R$ ).

### Risoluzione

Per  $V_O = V_{OL}$  si ha:

$$V_{CC} = V_L + R \cdot I_{OL} + V_{OL}$$

da cui:

$$R = \frac{V_{CC} - V_L - V_{OL}}{I_{OL}} = \frac{5 - 1.8 - 0.2}{10 \cdot 10^{-3}} = 300\Omega$$

Se  $V_I = 0$  si ha:  $V_O = V_{OH}$ . Nel circuito di uscita non circola corrente per cui il LED è spento.

Se  $V_I = V_{IH} > 2V$  si ha:  $V_O = V_{OL}$ . Il terminale di uscita assorbe la corrente  $I_{OL}$  proveniente dal circuito esterno per cui il LED è acceso.

### Esempio 2

Si vuole comandare da due punti distinti, tramite gli interruttori  $I1$  e  $I2$ , una lampadina da 100W/220V secondo lo schema di fig.2 che fa uso di un OR esclusivo CMOS, di un BJT e di un relè.

Descrivere il funzionamento del circuito. Sapendo che la bobina del relè assorbe la corrente  $I_C = 50mA$  a 12V circa e che per il BJT si ha:  $V_{CEsat} = 0.3V$ ,  $V_{BE} = 0.7V$ ,  $h_{FE} = 100$ , dimensionare la resistenza  $R_B$  affinché risulti:  $I_B = 5I_{Bmin}$ .

Si ponga:  $V_{CC} = +12V$  e  $V_{OH} = 11.5V$ .

[ R.  $R=4.32K\Omega$  ]

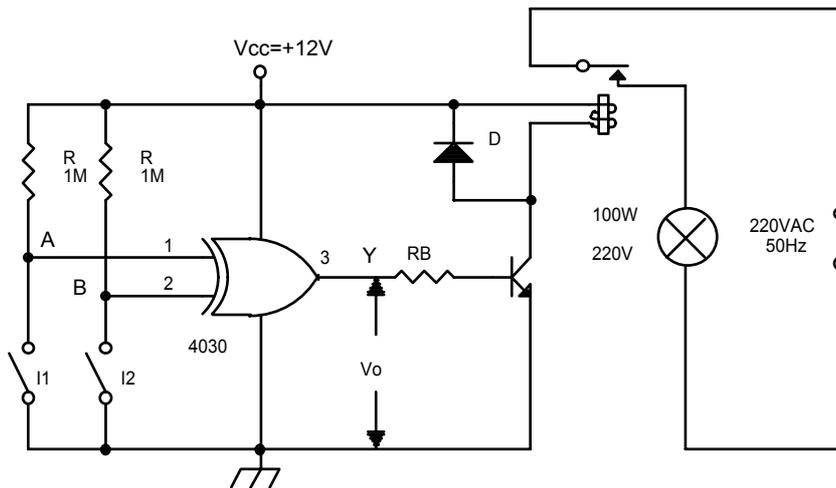


Fig.2. – Pilotaggio di un relè secondo le specifiche dell'Esempio 2.

In questo capitolo sarà analizzato il collegamento tra il microprocessore e le periferiche di I/O utilizzando circuiti integrati dedicati programmabili e non programmabili o schede di interfacce da inserire in opportune slot del computer.

Le interfacce, inoltre, si suddividono in *interfacce standard* e *non standard*.

Le interfacce standard sono quelle che rendono la periferica compatibile al computer a cui collegarsi. Ad esempio l'interfaccia *Centronics* presente nel PC è di tipo standard perché consente il collegamento al PC da parte di qualsiasi stampante in grado di riconoscere l'interfaccia Centronics. La compatibilità è di tipo meccanico (spina e presa compatibili), elettrico (il PC e la periferica generano e riconoscono tensioni e correnti di valori prestabiliti) e software (gestione del funzionamento della periferica).

Le interfacce standard più comuni, oltre alla Centronics, sono l'interfaccia seriale *RS-232* e la più recente interfaccia *USB* (Universal Serial Bus).

La prima utilizza il *collegamento parallelo* mentre le altre due sono *seriali*.

Il collegamento parallelo a  $n$  bit consente di trasferire simultaneamente tutti i bit ed impiega, pertanto, almeno  $n$  fili.

Il collegamento seriale trasferisce un bit per volta su uno stesso filo rispettando particolari protocolli per il sincronismo della trasmissione.

A parità di frequenza di sincronizzazione il collegamento parallelo è  $n$  volte più veloce di quello seriale. Quest'ultimo, però, ha il vantaggio di impiegare, in teoria, un solo filo per il trasferimento dei dati, condizione che torna particolarmente utile quando si devono collegare due apparecchiature poste a grande distanza come, ad esempio, due modem.

Le interfacce non standard sono quelle che non rispettano delle specifiche riconosciute da tutte le apparecchiature. Si utilizzano per collegare particolari dispositivi e spesso occorre prestare attenzione alle linee a cui effettuare il collegamento.

## 2. Slot di espansione di un personal computer

Affinché un dispositivo periferico possa scambiare dati con il microprocessore di un computer, è necessaria la presenza di un circuito di interfaccia che colleghi il BUS del microprocessore al periferico.

Il microprocessore alloggia su una scheda nota come *scheda madre*. Su di essa prendono posto le parti più importanti di un computer: memoria centrale di lettura e scrittura RAM e memoria a sola lettura ROM, clock e circuiti di supporto del microprocessore ed un certo numero di connettori, tra loro collegati in parallelo, in grado di ospitare le schede di interfaccia che consentono di collegare al computer stampante, modem, mouse, monitor, scanner, webcam, amplificatori audio, casse acustiche e microfoni, ecc.

Questi connettori possono essere assai diversi tra loro. Si va dai vecchi connettori ISA a 16 bit con velocità di trasferimento dati non superiore a 8Mbyte/s., ai più recenti connettori PCI a 32 o 64 bit con velocità massima di 264Mbyte/s. Le nuove schede grafiche che interfacciano il monitor al PC alloggiano su particolari connettori noti come AGP che consentono velocità di trasferimento dati fino a 1Gbyte/s. e, comunque, dipendenti dalla frequenza di clock della scheda madre.

### 2.1. ISA

Il connettore ISA (Industry Standard Architecture) implementa BUS a 16 bit. E' costituito da due parti: il connettore XT (Extended Tecnology) con 62 contatti da A1..A31, B1..B31 ed il connettore esteso con 36 contatti C1..C18, D1..D18.

Si riporta, in fig.3, la sagoma del connettore.

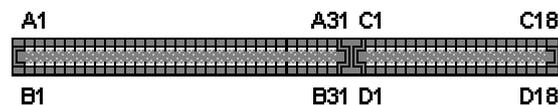


Fig.3. – Connettore ISA a 16 bit. Si nota a sinistra la parte corrispondente al connettore XT a 62 contatti per BUS dati ad 8 bit.

In tabella 1 si riporta la piedinatura del connettore ISA, il nome delle linee, le loro direzioni e la descrizione.

Tabella 1

Pin	Nome	Direz.	Descrizione	Pin	Nome	Direz.	Descrizione
A1	/I/O CH CK	←	I/O channel check; active low=parity error	B1	GND		Ground
A2	D7	↔	Data bit 7	B2	RESET	→	Active high to reset or initialize system logic
A3	D6	↔	Data bit 6	B3	+5V		+5 VDC
A4	D5	↔	Data bit 5	B4	IRQ2	←	Interrupt Request 2
A5	D4	↔	Data bit 4	B5	-5VDC		-5 VDC
A6	D3	↔	Data bit 3	B6	DRQ2	←	DMA Request 2
A7	D2	↔	Data bit 2	B7	-12VDC		-12 VDC
A8	D1	↔	Data bit 1	B8	/CARD SLCTD	←	Card selected; activated by cards in XT's slot J8
A9	D0	↔	Data bit 0	B9	+12VDC		+12 VDC
A10	I/O CH RDY	←	I/O Channel ready, pulled low to lengthen memory cycles	B10	GND		Ground
A11	AEN	→	Address enable; active high when DMA controls bus	B11	/SMEMW	→	System Memory Write
A12	A19	→	Address bit 19	B12	/SMEMR	→	System Memory Read
A13	A18	→	Address bit 18	B13	/IOW	→	I/O Write
A14	A17	→	Address bit 17	B14	/IOR	→	I/O Read
A15	A16	→	Address bit 16	B15	/DACK3	→	DMA Acknowledge 3
A16	A15	→	Address bit 15	B16	DRQ3	←	DMA Request 3
A17	A14	→	Address bit 14	B17	/DACK1	→	DMA Acknowledge 1
A18	A13	→	Address bit 13	B18	DRQ1	←	DMA Request 1
A19	A12	→	Address bit 12	B19	/REFRESH	↔	Refresh
A20	A11	→	Address bit 11	B20	CLOCK	→	System Clock (67 ns, 8-8.33 MHz, 50% duty cycle)
A21	A10	→	Address bit 10	B21	IRQ7	←	Interrupt Request 7
A22	A9	→	Address bit 9	B22	IRQ6	←	Interrupt Request 6
A23	A8	→	Address bit 8	B23	IRQ5	←	Interrupt Request 5
A24	A7	→	Address bit 7	B24	IRQ4	←	Interrupt Request 4
A25	A6	→	Address bit 6	B25	IRQ3	←	Interrupt Request 3
A26	A5	→	Address bit 5	B26	/DACK2	→	DMA Acknowledge 2
A27	A4	→	Address bit 4	B27	T/C	→	Terminal count; pulses high when DMA term. count reached
A28	A3	→	Address bit 3	B28	ALE	→	Address Latch Enable
A29	A2	→	Address bit 2	B29	+5V		+5 VDC
A30	A1	→	Address bit 1	B30	OSC	→	High-speed Clock (70 ns, 1431818 MHz, 50% duty cycle)
A31	A0	→	Address bit 0	B31	GND		Ground
C1	SBHE	↔	System bus high enable (data available on SD8-15)	D1	/MEMCS16	←	Memory 16-bit chip select (1 wait, 16-bit memory cycle)
C2	LA23	↔	Address bit 23	D2	/IOCS16	←	I/O 16-bit chip select (1 wait, 16-bit I/O cycle)
C3	LA22	↔	Address bit 22	D3	IRQ10	←	Interrupt Request 10
C4	LA21	↔	Address bit 21	D4	IRQ11	←	Interrupt Request 11
C5	LA20	↔	Address bit 20	D5	IRQ12	←	Interrupt Request 12
C6	LA18	↔	Address bit 19	D6	IRQ15	←	Interrupt Request 15
C7	LA17	↔	Address bit 18	D7	IRQ14	←	Interrupt Request 14
C8	LA16	↔	Address bit 17	D8	/DACK0	→	DMA Acknowledge 0
C9	/MEMR	↔	Memory Read (Active on all memory read cycles)	D9	DRQ0	←	DMA Request 0
C10	/MEMW	↔	Memory Write (Active on all memory write cycles)	D10	/DACK5	→	DMA Acknowledge 5
C11	SD08	↔	Data bit 8	D11	DRQ5	←	DMA Request 5
C12	SD09	↔	Data bit 9	D12	/DACK6	→	DMA Acknowledge 6
C13	SD10	↔	Data bit 10	D13	DRQ6	←	DMA Request 6

C14	SD11		Data bit 11	D14	/DACK7		DMA Acknowledge 7
C15	SD12		Data bit 12	D15	DRQ7		DMA Request 7
C16	SD13		Data bit 13	D16	+5 V		
C17	SD14		Data bit 14	D17	/MASTER		Used with DRQ to gain control of system
C18	SD15		Data bit 15	D18	GND		Ground

## 2.2. PCI

Il connettore PCI (Peripheral Component Interconnect) implementa BUS a 32 o 64 bit. Nel primo caso si avvale di due file, A e B, di 62 contatti ciascuna; la massima velocità di trasferimento dati è 133 Mbyte/s. Nel secondo caso i contatti sono 94 per ciascuna fila e la massima velocità di trasferimento dati è 266 Mbyte/s.

Si mostra, nella Tabella 2, la descrizione delle linee del bus PCI.

Tabella 2

Pin	+5V	+3.3V	Universal	Description	Pin	+5V	+3.3V	Universal	Description
A1	TRST			Test Logic Reset	B1	-12V			-12 VDC
A2	+12V			+12 VDC	B2	TCK			Test Clock
A3	TMS			Test Mode Select	B3	GND			Ground
A4	TDI			Test Data Input	B4	TDO			Test Data Output
A5	+5V			+5 VDC	B5	+5V			+5 VDC
A6	INTA			Interrupt A	B6	+5V			+5 VDC
A7	INTC			Interrupt C	B7	INTB			Interrupt B
A8	+5V			+5 VDC	B8	INTD			Interrupt D
A9	RESV01			Reserved VDC	B9	PRSNT1			Reserved
A10	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)	B10	RES			+V I/O (+5 V or +3.3 V)
A11	RESV03			Reserved VDC	B11	PRSNT2			??
A12	GND03	(OPEN)	(OPEN)	Ground or Open (Key)	B12	GND	(OPEN)	(OPEN)	Ground or Open (Key)
A13	GND05	(OPEN)	(OPEN)	Ground or Open (Key)	B13	GND	(OPEN)	(OPEN)	Ground or Open (Key)
A14	RESV05			Reserved VDC	B14	RES			Reserved VDC
A15	RESET			Reset	B15	GND			Reset
A16	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)	B16	CLK			Clock
A17	GNT			Grant PCI use	B17	GND			Ground
A18	GND08			Ground	B18	REQ			Request
A19	RESV06			Reserved VDC	B19	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)
A20	AD30			Address/Data 30	B20	AD31			Address/Data 31
A21	+3.3V01			+3.3 VDC	B21	AD29			Address/Data 29
A22	AD28			Address/Data 28	B22	GND			Ground
A23	AD26			Address/Data 26	B23	AD27			Address/Data 27
A24	GND10			Ground	B24	AD25			Address/Data 25
A25	AD24			Address/Data 24	B25	+3.3V			+3.3VDC
A26	IDSEL			Initialization Device Select	B26	C/BE3			Command, Byte Enable 3
A27	+3.3V03			+3.3 VDC	B27	AD23			Address/Data 23
A28	AD22			Address/Data 22	B28	GND			Ground
A29	AD20			Address/Data 20	B29	AD21			Address/Data 21
A30	GND12			Ground	B30	AD19			Address/Data 19
A31	AD18			Address/Data 18	B31	+3.3V			+3.3 VDC
A32	AD16			Address/Data 16	B32	AD17			Address/Data 17
A33	+3.3V05			+3.3 VDC	B33	C/BE2			Command, Byte Enable 2
A34	FRAME			Address or Data phase	B34	GND13			Ground
A35	GND14			Ground	B35	IRDY			Initiator Ready

A36	TRDY			Target Ready	B36	+3.3V06			+3.3 VDC
A37	GND15			Ground	B37	DEVSEL			Device Select
A38	STOP			Stop Transfer Cycle	B38	GND16			Ground
A39	+3.3V07			+3.3 VDC	B39	LOCK			Lock bus
A40	SDONE			Snoop Done	B40	PERR			Parity Error
A41	SBO			Snoop Backoff	B41	+3.3V08			+3.3 VDC
A42	GND17			Ground	B42	SERR			System Error
A43	PAR			Parity	B43	+3.3V09			+3.3 VDC
A44	AD15			Address/Data 15	B44	C/BE1			Command, Byte Enable 1
A45	+3.3V10			+3.3 VDC	B45	AD14			Address/Data 14
A46	AD13			Address/Data 13	B46	GND18			Ground
A47	AD11			Address/Data 11	B47	AD12			Address/Data 12
A48	GND19			Ground	B48	AD10			Address/Data 10
A49	AD9			Address/Data 9	B49	GND20			Ground
A50					B50	(OPEN) GND	(OPEN)		Ground or Open (Key)
A51					B51	(OPEN) GND	(OPEN)		Ground or Open (Key)
A52	C/BE0			Command, Byte Enable 0	B52	AD8			Address/Data 8
A53	+3.3V11			+3.3 VDC	B53	AD7			Address/Data 7
A54	AD6			Address/Data 6	B54	+3.3V12			+3.3 VDC
A55	AD4			Address/Data 4	B55	AD5			Address/Data 5
A56	GND21			Ground	B56	AD3			Address/Data 3
A57	AD2			Address/Data 2	B57	GND22			Ground
A58	AD0			Address/Data 0	B58	AD1			Address/Data 1
A59	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)	B59	VCC08			+5 VDC
A60	REQ64			Request 64 bit ?	B60	ACK64			Acknowledge 64 bit ?
A61	VCC11			+5 VDC	B61	VCC10			+5 VDC
A62	VCC13			+5 VDC	B62	VCC12			+5 VDC
A63	GND			Ground	B63	RES			Reserved
A64	C/BE[7]#			Command, Byte Enable 7	B64	GND			Ground
A65	C/BE[5]#			Command, Byte Enable 5	B65	C/BE[6]#			Command, Byte Enable 6
A66	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)	B66	C/BE[4]#			Command, Byte Enable 4
A67	PAR64			Parity 64 ?	B67	GND			Ground
A68	AD62			Address/Data 62	B68	AD63			Address/Data 63
A69	GND			Ground	B69	AD61			Address/Data 61
A70	AD60			Address/Data 60	B70	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)
A71	AD58			Address/Data 58	B71	AD59			Address/Data 59
A72	GND			Ground	B72	AD57			Address/Data 57
A73	AD56			Address/Data 56	B73	GND			Ground
A74	AD54			Address/Data 54	B74	AD55			Address/Data 55
A75	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)	B75	AD53			Address/Data 53
A76	AD52			Address/Data 52	B76	GND			Ground
A77	AD50			Address/Data 50	B77	AD51			Address/Data 51
A78	GND			Ground	B78	AD49			Address/Data 49
A79	AD48			Address/Data 48	B79	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)
A80	AD46			Address/Data 46	B80	AD47			Address/Data 47
A81	GND			Ground	B81	AD45			Address/Data 45
A82	AD44			Address/Data 44	B82	GND			Ground
A83	AD42			Address/Data 42	B83	AD43			Address/Data 43
A84	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)	B84	AD41			Address/Data 41
A85	AD40			Address/Data 40	B85	GND			Ground
A86	AD38			Address/Data 38	B86	AD39			Address/Data 39
A87	GND			Ground	B87	AD37			Address/Data 37
A88	AD36			Address/Data 36	B88	+5V	+3.3V	Signal Rail	+V I/O (+5 V or +3.3 V)
A89	AD34			Address/Data 34	B89	AD35			Address/Data 35
A90	GND			Ground	B90	AD33			Address/Data 33

A91	AD32		Address/Data 32	B91	GND		Ground
A92	RES		Reserved	B92	RES		Reserved
A93	GND		Ground	B93	RES		Reserved
A94	RES		Reserved	B94	GND		Ground

### 2.3. AGP

Acronimo di *Accelerated Graphics Port*, è un bus che collega direttamente scheda video, processore e memoria, e permette di accelerare lo scambio di dati fra tali componenti.

L'AGP è stato creato da Intel nel 1997 e consente di utilizzare la memoria di sistema quando quella della scheda grafica si esaurisce. Tale possibilità viene però utilizzata solo nelle schede video economiche, a causa della maggiore lentezza di questa soluzione.

Il vero vantaggio dell'AGP sta nella maggiore banda passante (266 MByte/s. nella versione 1X, oltre un gigabyte al secondo nella versione 4x) rispetto al bus PCI e nel fatto che quest'ultimo viene liberato dal traffico dati della scheda video.

La velocità dipende sia dalla frequenza di funzionamento che dalla lunghezza della parola da trasferire. Nata per funzionare a 66.6MHz per parole a 32bit (4 byte) la velocità di trasferimento, in tal caso, vale:  $66.6\text{MHz} \times 4\text{Byte} = 266\text{Mbyte/s}$ .

La versione 2X, per ogni ciclo di clock, sfrutta sia il fronte di salita che quello di discesa consentendo, così, il raddoppio della massima velocità: 532Mbyte/s. Nella versione 4X la velocità supera il Gbyte/s. poiché il bus prevede il trasferimento di dati a 64bit. La velocità può ulteriormente aumentare aumentando la frequenza della CPU a bordo delle scheda grafica e passando a 128 il numero di bit trasferiti localmente sulla scheda.

Poiché il bus AGP prevede attualmente la presenza di un solo dispositivo, non possono essere presenti più slot AGP sulla motherboard.

## 3. Interfacce standard

Come già detto nel paragrafo 1, le interfacce standard sono quelle che rendono compatibile il collegamento tra un dispositivo periferico ed il computer.

Le interfacce standard, normalmente, sono costituite da schede elettroniche inserite in una slot di espansione di un computer: ISA, PCI, AGP. In quest'ultima, in realtà, può trovar posto solo una scheda grafica in grado di pilotare un monitor attraverso il connettore VGA disponibile sul retro del PC. Le schede grafiche più recenti presentano anche una uscita TV in grado di comandare un normale ricevitore televisivo oppure un videoregistratore.

Le interfacce standard su schede più recenti sono disponibili solo su connettori PCI. Le interfacce standard più note sono:

1. Interfaccia Centronics
2. Interfaccia SCSI
3. Interfaccia RS-232
4. Interfaccia USB
5. Interfaccia IEEE1394

La prima è un'interfaccia parallela a 8 bit. Il suo nome è quello della ditta di stampanti che per prima diffuse questa interfaccia per poter, appunto, collegare una stampante al computer. Utilizzata su ogni computer per oltre un ventennio per connettere stampanti e dispositivi periferici come scanner, webcam, ecc., è attualmente in disuso a causa della limitata velocità di funzionamento.

L'interfaccia SCSI (Small Computer System Interface), anch'essa parallela ad 8 bit e ratificata nel 1986, ha subito nel tempo numerose evoluzioni ed è utilizzata per collegare dispositivi destinati ad essere particolarmente veloci: hard-disk, scanner, masterizzatori.

L'interfaccia RS-232 è di tipo seriale, ha subito nel tempo delle evoluzioni che ne hanno consentito di aumentare la velocità di trasferimento dati che, comunque, rimane abbastanza basso (256Kbit/s.) e ridotto il numero delle linee effettivamente utilizzate.

L'interfaccia USB (Universal Serial Bus), apparsa nel 1995, è destinata a soppiantare la RS-232 per alcune altre notevoli caratteristiche e per la sua elevata velocità che può raggiungere il valore di 12Mbit/s e di 420Mbps per la versione 2.0.

L'interfaccia IEEE1394, anch'essa seriale, consente un flusso massimo di dati pari a 400Mbps con la possibilità di collegare fino a 63 dispositivi in serie o a stella.

L'inserzione di nuovi dispositivi collegati a tale interfaccia può avvenire a caldo.

Sulla maggior parte delle moderne schede madri di PC sono già presenti le interfacce Centronics, RS-232 e USB. In tal modo restano disponibili i connettori ISA e PCI per altre interfacce standard e non standard.

Alcune schede madri, oltre a contenere le citate interfacce, contengono anche alcune altre interfacce, normalmente presenti su slot e, addirittura esterne. Tali schede madri sono denominate "all-in-one" e contengono:

- 1 Centronics;
- 2 Seriali RS-232;
- 2 USB;
- 1 Modem interno;
- 1 interfaccia grafica;
- 1 interfaccia audio.

Tale soluzione risulta particolarmente economica ed interessante ma poco si adatta ad esigenze legate all'aggiornamento delle citate interfacce già integrate su scheda.

### 3.1. Interfaccia Centronics

E' una interfaccia parallela ad 8 bit di tipo asincrona usata soprattutto per collegare un computer ad una stampante parallela. Il flusso di dati è tipicamente monodirezionale e va, naturalmente, dal computer alla stampante. Recentemente, grazie alla evoluzione della interfaccia, è possibile anche inviare sulle stesse linee dati che vanno dal dispositivo periferico al computer e ciò consente l'utilizzo di tale interfaccia anche per il collegamento di dispositivi di input come, ad esempio, lo scanner o la webcam. Il connettore sul retro del computer è di tipo D a 25 poli femmina. Su un PC possono prendere posto fino a 3 interfacce parallele denominate LPT1, LPT2 e LPT3 (Line Printer Terminal). Ciascuna delle 3 LPT presenta 3 indirizzi contigui destinate alle periferiche di I/O.

L'indirizzo base della LPT1 è 888 (in esadecimale: 378), gli altri due indirizzi sono 889 e 890. L'indirizzo base della LPT2 è 632 (in esadecimale: 278), gli altri due indirizzi sono 633 e 634. L'indirizzo base della LPT3 è 956 (in esadecimale: 3BC), gli altri due indirizzi sono 957 e 958. Si riassume la situazione nella seguente tabella 3.

Tabella 3

	Ind. base	Ind.base+1	Ind.base+2
<b>LTP1</b>	888	889	890
<b>LPT2</b>	632	633	634
<b>LPT3</b>	956	957	958

Alla LPT1 viene riservato l'interrupt IRQ7. Il registro base di indirizzo 888, denominato *registro dati*, contiene 8 bit di uscita dal PC.

Il registro di indirizzo successivo 889, noto come *registro di stato*, è accessibile solo dall'esterno e solamente per 5 dei suoi 8 bit. È detto registro di stato perché ciascuna delle 5 linee individua un particolare stato in cui si trova la stampante (occupata, carta esaurita, errore, ecc.). Il registro di indirizzo 890, noto come *registro di controllo*, rende disponibili solo 4 bit di uscita.

In fig.4 si mostra il connettore a 25 poli posto sul retro del computer.

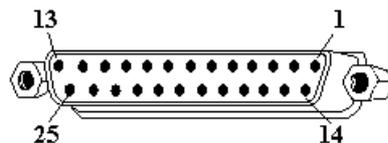


Fig. 4 – Interfaccia Centronics. Connettore a 25 poli femmina posto sul retro del PC

In tabella 4 si riporta la piedinatura, la denominazione, la direzione, l'indirizzo e l'uso delle linee che l'interfaccia Centronics mette a disposizione sul connettore D a 25 poli agli indirizzi 888, 889 e 890 della LPT1.

Tabella 4

PIN	NOME	DIREZIONE	INDIRIZZO ED USO
1	$\overline{\text{STROBE}}$	USCITA	890 OUT 890,1 0V 890 OUT 890,0 5V
2	DATA1	USCITA	OUT 888,N Per $0 \leq N \leq 255$
3	DATA2	USCITA	
4	DATA3	USCITA	
5	DATA4	USCITA	
6	DATA5	USCITA	
7	DATA6	USCITA	
8	DATA7	USCITA	
9	DATA8	USCITA	
10	ACK	INGRESSO	B3 (64) SE ALTO
11	$\overline{\text{BUSY}}$	INGRESSO	B4 (128) SE BASSO
12	PAPER OUT	INGRESSO	B2 (32) SE ALTO
13	SELECTED	INGRESSO	B1 (16) SE ALTO
14	$\overline{\text{AUTOFEED}}$	USCITA	890 OUT 890,2 0V
15	ERROR	INGRESSO	B0 ( 8 ) SE ALTO
16	INIZIALIZE PRINTER	USCITA	890 OUT 890,4 5V
17	$\overline{\text{SELECT INPUT}}$	USCITA	890 OUT 890,8 0V
18...25	GND		

**Registro dati (indirizzo 888) :**

Le linee di uscita DATA8...DATA1, di indirizzo 888 (378 esadecimale), situate tra i pin 9...2, sono memorizzate, cioè rappresentano i bit di uscita di altrettanti flip-flop.

LINEE	DATA8	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1
PIN	9	8	7	6	5	4	3	2

**Registro di stato (indirizzo 889):**

L'interfaccia Centronics possiede 5 linee di ingresso all'indirizzo 889 (379 esadecimale) con i seguenti valori e logiche di funzionamento:

$\overline{\text{BUSY}}$	(pin 11)	vale 128 se è al livello basso	(logica negativa).
ACK	(pin 10)	vale 64 se è al livello alto	(logica positiva).
PAPER OUT	(pin 12)	vale 32 se è al livello alto	(logica positiva).
SELECTED	(pin 13)	vale 16 se è al livello alto	(logica positiva).
ERROR	(pin 15)	vale 8 se è al livello alto	(logica positiva).

LINEE	$\overline{\text{BUSY}}$	ACK	PAPER OUT	SELECTED	ERROR	1	1	1
PIN	11	10	12	13	15			

I rimanenti 3 bit meno significativi del byte di ingresso non sono accessibili sul connettore e sono poste, internamente all'interfaccia, al livello alto ( $111_2 = 7_{10}$ ).

Avendo a disposizione 5 bit (32 combinazioni) è possibile acquisire, via software in una variabile A, un valore compreso tra 0 e 31.

Ciò si realizza con i seguenti passaggi:

$A = A \text{ XOR } 128$  (per complementare il bit applicato su  $\overline{\text{BUSY}}$ );

$A = A - 7$  (per azzerare i tre bit meno significativi : operazione, comunque, non necessaria);

$A = A / 8$  (per lo scorrimento a destra di 3 posizioni).

L'algoritmo di acquisizione in linguaggio QBASIC risulta, pertanto:

$$A = ((\text{INP}(889) \text{ XOR } 128) - 7) / 8$$

In questo modo la parola di ingresso A assume un valore compreso tra 0 e 31.

#### Registro di controllo (indirizzo 890):

L'indirizzo Centronics 890 (37A esadecimale) rende disponibile in uscita altri 4 bit di cui tre attivi in logica negativa ed una in logica positiva :

$\overline{\text{STB}} = \overline{\text{STROBE}}$  (pin 1) vale 1 se è al livello basso (logica negativa).

$\overline{\text{AF}} = \overline{\text{AUTOFEED}}$  (pin 14) vale 2 se è al livello basso (logica negativa).

IP = INIZIALIZE PRINTER (pin 16) vale 4 se è al livello alto (logica positiva).

$\overline{\text{SI}} = \overline{\text{SELECT INPUT}}$  (pin 17) vale 8 se è al livello basso (logica negativa).

IRQE = ABILITA INTERRUPT vale 16 se è al livello alto (logica positiva) ma non è disponibile fisicamente.

indirizzo 890	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LINEE				IRQE	$\overline{\text{SI}}$	IP	$\overline{\text{AF}}$	$\overline{\text{STB}}$
PIN					17	16	14	1

Se si intende trasmettere una parola a 4 bit con valore compreso tra 0 e 15 e con i bit attivi in logica positiva, occorre commutare (XOR con 1 logico) i bit **STB**, **AF** e **SI** e lasciare inalterato (XOR con zero logico) il bit IP :

#### OUT 890, A XOR 11

<b>A3</b>	<b>A2</b>	<b>A1</b>	<b>A0</b>	<b>A nel PC</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>11</b>
<b><math>\overline{A3}</math></b>	<b>A2</b>	<b><math>\overline{A1}</math></b>	<b><math>\overline{A0}</math></b>	<b>A XOR 11</b>
<b>A3</b>	<b>A2</b>	<b>A1</b>	<b>A0</b>	<b>A in uscita</b>

#### Procedura per dichiarare di ingresso la porta all'indirizzo 888.

Se l'interfaccia Centronics è impostata come EPP (Enhanced Parallel Port) è possibile effettuare l'input di un byte all'indirizzo 888 purché si imposti ad 1 il bit 5 del registro di controllo posto all'indirizzo 890.

Impostando a 0 il bit 5 dell'indirizzo 890, il byte DATA8...DATA1 del registro dati all'indirizzo 888 ritorna a funzionare come byte di output.

Si riporta un semplice listato di test in QBASIC (facilmente modificabile in un altro linguaggio di programmazione) per provare la Centronics all'indirizzo 888 come canale di input a 8 bit.

Con un semplice spezzone di cavo telefonico (quello usato per collegamenti su bread-board) si collega il pin 2 (DATA1) al pin 25 (GND) del connettore (femmina) a 25 poli posto sul retro del PC e si prova il programma ottenendo sul monitor la stampa continua del numero 254 fino alla pressione di un tasto qualsiasi sulla tastiera. Portando il filo sul pin 3, 4, ecc. tenendo l'altro capo del filo sempre a massa, si può verificare che effettivamente il dato in lettura si modifica in funzione del pin collegato a massa. È superfluo dire che le linee DATA8 ... DATA1 posti sui pin 9...2 sono da ritenersi allo stato 1 se tenute aperte.

Per impostare l'interfaccia Centronics come EPP, infine, si va nel BIOS SETUP e si imposta la stampante come EPP anziché come SPP.

```

OUT 890, 32
DO
PRINT INP(888)
LOOP WHILE
INKEY$=""
END

```

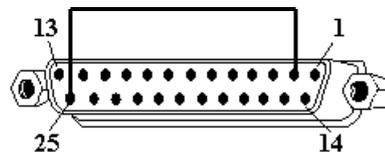


Fig.5. – Programma e collegamenti per provare la Centronics come porta di ingresso all'indirizzo 888.

### 3.2. Interfaccia seriale EIA RS232-C

L'interfaccia seriale americana EIA RS232-C (EIA= Electronic Industries Associates), corrispondente alla V.24/V.28 dell'ITU-T, è uno standard di collegamento seriale che può essere di tipo sincrono o asincrono tra un dispositivo di comunicazione DCE come, ad esempio, il modem (Data Communication Equipment) e un dispositivo terminale DTE, come, ad esempio, il computer (Data Terminal Equipment) con velocità di trasmissione inferiore o uguale a 19.2Kbps (questo limite è oramai superato).

L'interfaccia è costituita da un insieme di 25 linee, non tutte indispensabili, che trasportano i bit di dati, segnali di controllo e la massa.

Nel collegamento tra un computer ed un dispositivo periferico vengono adoperati dei connettori miniatura tipo D a 25 poli. Sul DTE (computer, ad esempio) si trova la spina (connettore maschio) mentre sul DCE (modem) si trova la presa (connettore femmina). In alcuni DCE (ad esempio, il mouse seriale) manca la presa esterna poiché il cavo di collegamento entra direttamente nell'apparecchiatura.

I tipici dispositivi periferici che si possono collegare ad un computer via RS232 sono: stampante, modem, mouse, ecc.

#### 3.2.1. Caratteristiche meccaniche ed elettriche

In fig.6a si mostra il connettore a 25 poli per la RS-232C le cui caratteristiche meccaniche sono normalizzate secondo lo standard ISO 2110 della International Standard Organization. In molte applicazioni pratiche non si utilizzano tutte le linee ma solo una piccola parte di esse. In tal caso si fa uso di un connettore ridotto a 9 poli come quello in fig.6b.

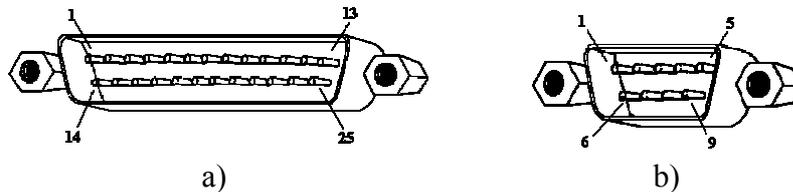


Fig. 6. - Connettore per la RS-232C: a) di tipo a 25 poli; b) di tipo a 9 poli.

Il significato di tali linee è descritto nella Tabella 5.

Qualunque sia la linea (dati, clock o controlli), il circuito elettrico equivalente di tale interfaccia tra il trasmettitore e il ricevitore è quello indicato in fig.7.

In tale standard si definisce *mark* la tensione  $V_i$  di valore inferiore a  $-3V$  e si definisce *space* quella superiore a  $+3V$ . Durante la trasmissione si associa il livello logico 1 a *mark* e 0 a *space*. Si osserva subito che i livelli logici sono bipolari e in logica negativa (tensione positiva=0; tensione negativa=1). Tipicamente i valori di tensione assunti sono  $\pm 12V$ .

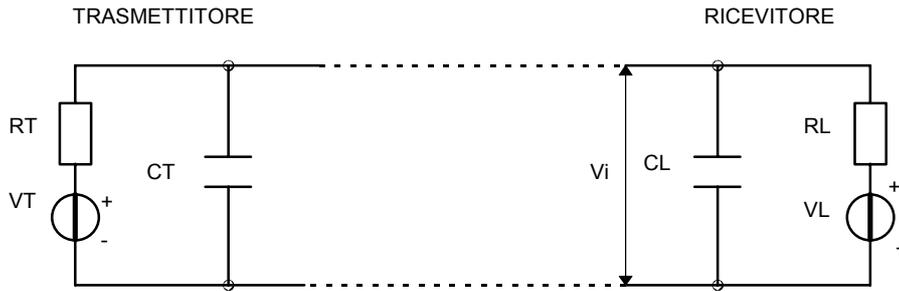


Fig. 7. - Circuito equivalente tra trasmettitore e ricevitore nello standard RS232-C.

$V_T$  = f.e.m. del trasmettitore a circuito aperto;       $R_T$  = Resistenza interna del trasmettitore;  
 $C_T$  = Capacità equivalente del trasmettitore;       $C_L$  = Capacità equivalente del ricevitore;  
 $R_L$  = Resistenza d'ingresso del ricevitore;       $V_L$  = f.c.e.m. del ricevitore a circuito aperto;  
 $V_i$  = d.d.p. all'interfaccia.

La resistenza di carico del ricevitore  $R_L$  deve essere compresa tra  $3K\Omega$  e  $7K\Omega$ , la capacità  $C_L$  in parallelo al carico deve essere inferiore a  $2.5nF$  e la f.c.e.m.  $V_L$  non deve superare i  $2V$ .

La f.e.m.  $V_T$  del driver del trasmettitore non deve superare i  $25V$ ,  $R_T$  e  $C_T$  non sono specificati ma devono essere tali da evitare una corrente di corto circuito superiore a  $0.5A$  e da consentire una  $V_i$  compresa tra  $5V$  e  $15V$ .

Poiché la capacità per unità di lunghezza di un cavo è di circa  $200pF/metro$  si evince che la massima distanza tra i dispositivi collegati in tale standard non deve superare i  $12-15$  metri. In tabella 5 si descrive la piedinatura del connettore a 9 e 25 poli, il nome e la descrizione delle linee della RS-232C.

Tabella 5

PIN (9)	PIN (25)	NOME V.24 ITU	NOME RS-232	DESCRIZIONE
	1	C101	FG	Frame ground = Massa di protezione
3	2	C103	TxD	Transmitted data = Dati in trasmissione
2	3	C104	RxD	Received data = Dati in ricezione
7	4	C105	RTS	Request to send = Richiesta di trasmissione
8	5	C106	CTS	Clear to send = Pronto a trasmettere
6	6	C107	DSR	Data set ready = DCE pronto
5	7	C102	GND	Ground = Massa dei segnali
1	8	C109	DCD	Data carrier detector = Portante in ricezione presente
	9			Riservato per apparecchi di collaudo
	10			Riservato per apparecchi di collaudo
	11	C126	CK	Scelta frequenza in trasmissione
	12	C122	SCF	Segnale di ricezione presente sul canale ausiliario
	13	C121	SCB	Pronto per la trasmissione sul canale ausiliario
	14	C118	SBA	Dati in trasmissione del canale ausiliario
	15	C114	TC	Transmit clock = Clock di trasmissione dal modem
	16	C119	SBB	Dati in ricezione del canale ausiliario
	17	C115	RC	Received clock = Clock di ricezione
	18			Non connesso
	19	C120	SCA	Richiesta di trasmissione del canale ausiliario
4	20	C108	DTR	Data terminal ready = DTE pronto
	21	C110	CG	Rivelatore della qualità del segnale
9	22	C125	RI	Ring indicator = Chiamata in arrivo
	23	C111	CI	Selezione velocità di trasmissione da DTE
	24	C113	DA	Clock di trasmissione da DTE
	25			Non connesso

### 3.2.2. Caratteristiche funzionali dell'interfaccia ITU-T V.24

Si descrivono le caratteristiche funzionali delle linee (circuiti) dell'interfaccia V.24 esaminate nella precedente tab.5. Nella trasmissione dati tra un DTE e un DCE il numero di linee utilizzate dipende dal tipo di collegamento e dai modem impiegati.

C101= Massa di protezione.

Linea collegata alla massa dei segnali C102 all'interno del modem.

C102=Massa dei segnali.

Linea comune di riferimento per tutti i circuiti di interfaccia.

C103=Dati in trasmissione (DTE → DCE)

I dati binari in forma seriale generati dal DTE viaggiano verso il DCE. In assenza di trasmissione tale linea è nello stato MARK (tensione negativa).

C104=Dati in ricezione (DCE → DTE)

I dati binari in forma seriale generati dal DCE viaggiano verso il DTE.

C105=Richiesta di trasmissione (DTE → DCE)

Tale segnale obbliga il modem a trasmettere la portante in linea entro 2ms. Se la linea è nello stato di riposo il modem interrompe la trasmissione entro 2ms.

C106=Pronto a trasmettere (DCE → DTE)

Tale segnale indica che il DCE è pronto a trasmettere in linea e rappresenta la risposta alla linea C105 purché il modem sia connesso alla linea telefonica.

Il tempo di risposta in corrispondenza della prima richiesta di trasmettere è compreso tra 750 e 1400ms. ed è compreso tra 22 e 40ms. nelle successive richieste di trasmissione.

C107=Modem pronto (DCE → DTE)

Questo segnale indica al DTE che il modem è collegato alla linea telefonica e che non è in condizioni di prova. Nei modem in banda base tale linea è sempre attiva.

C108=DTE pronto (DTE → DCE)

L'attivazione di questa linea provoca il collegamento del modem alla linea telefonica qualunque sia la condizione delle altre linee. Questa linea, inoltre, indica che il DTE è pronto sia a trasmettere che a ricevere i dati.

C109=Portante in ricezione (DCE → DTE)

Il DCE informa il DTE che la portante in linea è ad un livello superiore alla soglia di ricezione.

C110=Rivelatore della qualità del segnale (DCE → DTE)

Il DCE informa il DTE che non ci sono ragioni per credere che i dati ricevuti siano errati.

C111=Selezione di velocità (DTE → DCE)

L'attivazione di tale linea obbliga il modem a scegliere la velocità più elevata. Il livello logico opposto imposta la velocità più bassa (per modem 1200bps / 600bps).

C112=Selezione velocità (DCE → DTE)

Se la scelta della velocità è fatta dal modem, quest'ultimo avverte il DTE della velocità selezionata : lo stato ON sceglie la velocità più alta, lo stato OFF quella più bassa. Questa linea utilizza lo stesso pin del C111 ed è ad esso alternativa.

C113=Clock di trasmissione da DTE (DTE → DCE)

Questa linea rappresenta il clock generato dal DTE. Le transizioni ON-OFF devono coincidere con la posizione centrale su ciascun bit della linea C103.

C114=Clock di trasmissione da DCE (DCE → DTE)

Questa linea rappresenta il clock generato dal DCE. Durante le transizioni OFF-ON il DTE deve generare il bit successivo su C103.

C115=Clock in ricezione (DCE → DTE)

Questa linea rappresenta il clock per i dati che il DTE riceve su C104. La forma d'onda è quadra e la transizione ON-OFF indica il centro del bit ricevuto su C104.

C118=Trasmissione dati sul supervisore (DTE → DCE)

Questa linea svolge le stesse funzioni del C103 ma si riferisce al canale supervisore. Per i modem a 1200bps il canale supervisore è a 75 bps.

C119=Ricezione dati sul supervisore (DCE → DTE)

Questa linea svolge le stesse funzioni del C104 ma si riferisce al canale supervisore. Per i modem a 1200bps il canale supervisore è a 75 bps.

C120=Richiesta di trasmissione sul canale supervisore (DTE → DCE)

Questa linea svolge le stesse funzioni del C105 ma si riferisce al canale supervisore.

C121=Pronto a trasmettere sul canale supervisore (DCE → DTE)

Questa linea svolge le stesse funzioni del C106 ma si riferisce al canale supervisore.

C122=Rivelatore segnale di linea dati supervisore (DCE → DTE)

Questa linea svolge le stesse funzioni del C109 ma si riferisce al canale supervisore.

C123=Rivelazione qualità segnale (DCE → DTE)

Questa linea svolge le stesse funzioni del C110 ma si riferisce al canale supervisore.

C125=Chiamata in arrivo (DCE → DTE)

L'attivazione di questa linea indica al DTE che il modem ha ricevuto una chiamata telefonica. La risposta del DTE sulla linea C108 provoca la connessione del modem alla linea (risposta automatica).

C126=Scelta della frequenza in trasmissione (DTE → DCE)

Linea che serve alla scelta della frequenza del modem quando questo prevede due canali.

Le linee dei segnali sono dette *sbilanciate* perché sono tutte riferite ad una massa comune (circuito C102).

### 3.2.3. Conversione TTL RS-232 e viceversa

Per trasformare un livello logico TTL in RS-232 compatibile occorre un traslatore di livello in modo che il valore di tensione compreso tra +2V e +5V (1 logico TTL) si trasformi in -12V mentre un valore di tensione compreso tra 0 e 0.8V (0 logico TTL) si trasformi in +12V.

Vi sono integrati in grado di convertire un livello TTL in RS-232C e viceversa come, ad esempio, il 1448 e 1449 (fig.8).

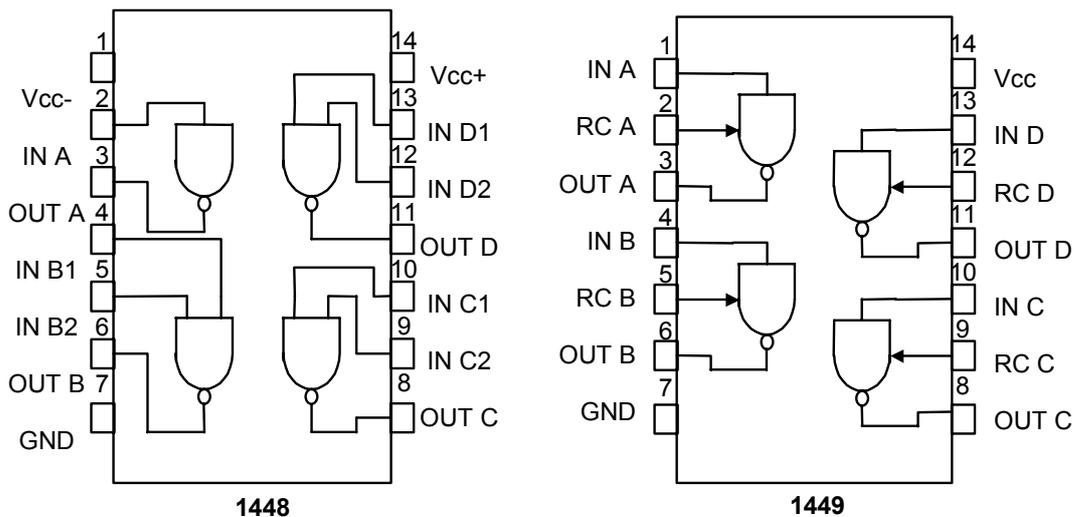


Fig. 8. - Convertitori di livello integrati TTL/RS-232C e viceversa.

Il primo converte un livello TTL in RS-232C. Ha bisogno di doppia alimentazione (da  $\pm 12V$  a  $\pm 15V$ ) e contiene 4 dispositivi di tipo invertente. Il secondo, invece, converte un livello RS-232 in TTL. È alimentato con una tensione di alimentazione a +5V e contiene 4 elementi invertenti. Il collegamento bidirezionale seriale tra due dispositivi che lavorano con segnali TTL si può effettuare come in fig.9.

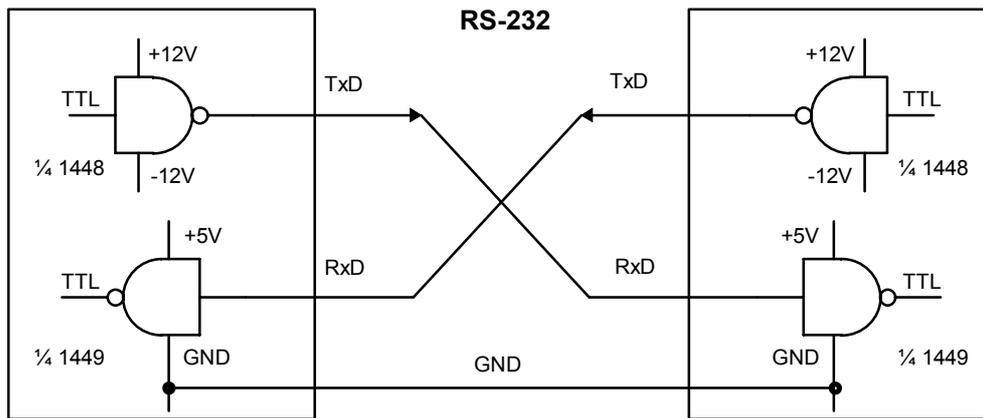


Fig. 9. - Collegamento seriale con segnali nello standard RS-232C tra dispositivi TTL.

I valori di tensione assegnati allo standard RS-232C consentono di ottenere una elevata immunità al rumore che aumenta all'aumentare della "distanza" tra i due livelli logici.

### 3.2.4. Collegamento null-modem

Per collegare tra di loro due computer con l'interfaccia seriale RS-232 si deve preparare un cavo, noto come cavo **null modem**, con almeno tre fili (RxD, TxD e GND) e due connettori a 25 poli femmine da porre alle due estremità del cavo. Occorre, però, fornire i corretti potenziali alle linee di ingresso che non si intendono utilizzare.

In fig.10 si mostra un tipico collegamento ove il pin 5 (linea di ingresso CTS) è pilotato dallo stesso DTE attraverso il pin 4 (linea di uscita RTS). Analogamente la linea di uscita DTR (pin 20) va collegata ai pin 6 e 8 di ingresso (DSR e DCD, rispettivamente). Ciò vale per entrambi i computer.

In questo modo si "imbrogia" il computer ma si deve rinunciare al collegamento in *handshake*. La tecnica dell'*handshake* (letteralmente stretta di mano) consiste nel sincronizzare il collegamento attraverso due linee di controllo: il trasmettitore invia un livello logico di richiesta di invio (uscita RTS, pin 4) ed il ricevitore risponde con un livello logico di pronto a trasmettere (ingresso CTS pin 5); in pratica si autorizza il trasmettitore ad inviare i dati.

L'imbroglio (collegamento pin 4 e 5 dello stesso connettore) consiste nel fatto che il trasmettitore "autorizza se stesso" ad inviare dati.

Il collegamento dei pin 6, 8 e 20 dello stesso connettore realizza un "imbroglio" simile.

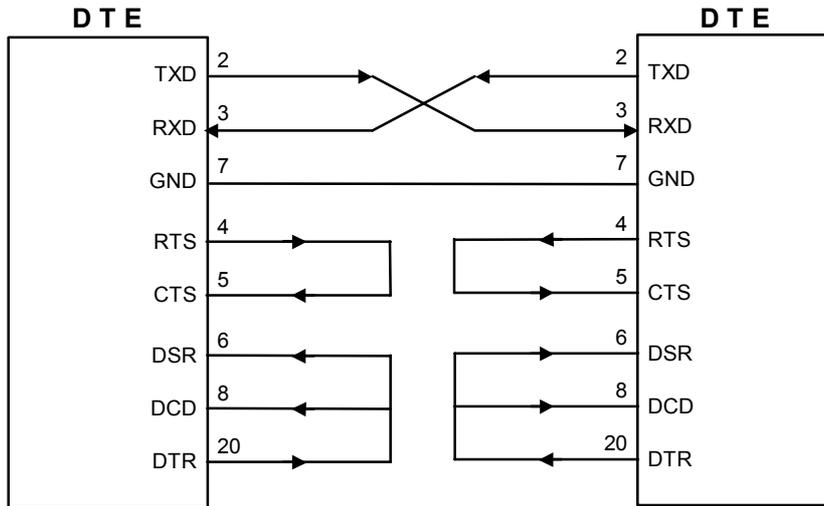


Fig. 10. - Collegamento tra due computer via RS-232C con cavo NULL-MODEM.

Volendo un collegamento con handshake si devono utilizzare almeno 5 fili. In fig.11 si mostra un possibile collegamento. I due computer, essendo collegati direttamente e non via modem, possono comunicare ad elevata velocità.

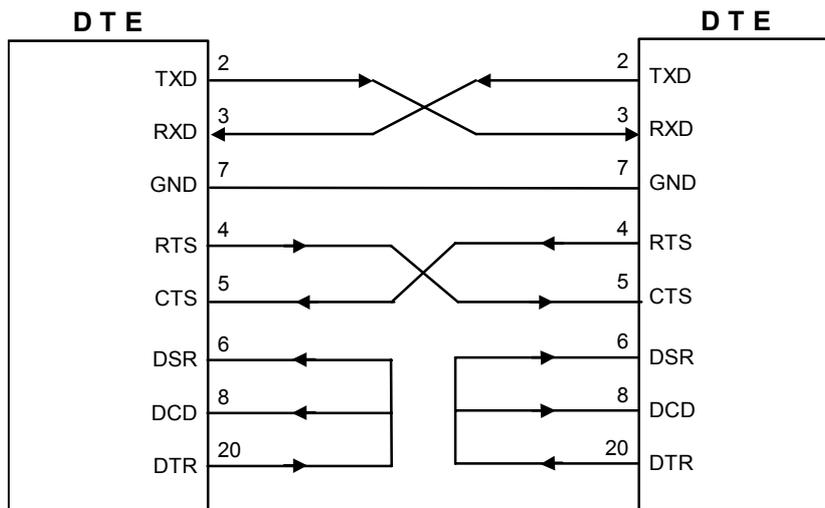


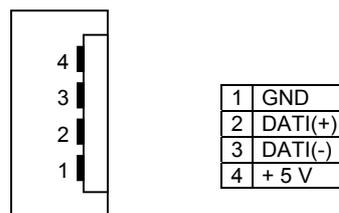
Fig. 11. - Collegamento tra due computer via RS-232C con cavo NULL-MODEM con possibilità di handshake.

### 3.3. USB

L'interfaccia USB (Universal Serial Bus - Bus seriale universale) fu creata nel 1995 da un consorzio di costruttori fra cui Intel, Compaq, Digital e Microsoft, con l'intento di sostituire le attuali porte seriali e parallele per il collegamento di periferiche al PC. Raggiunge la velocità di 12 MBit/s. e si possono collegare fino a 127 periferiche in cascata. La versione 2.0 consente velocità fino a 480Mbps.

L'USB può fornire direttamente l'alimentazione alle periferiche a basso consumo (come tastiere e mouse). E' inoltre completamente Plug and Play: le periferiche possono essere collegate "a caldo" (senza spegnere la macchina) e vengono automaticamente riconosciute.

Ciascuno dei due connettori disponibili sul retro del PC è a 4 poli disposti secondo la fig.12. La massima corrente fornita dai due connettori è di 1A.



**Fig.12. – Uno dei due connettori USB disponibili sul retro del PC.**

In figura 13 si mostra il cavo USB che collega la presa posta sul retro del PC al dispositivo periferico USB come, ad esempio, stampante, scanner, modem, webcam.



**Fig.13. – Terminali del cavo USB. A sinistra si mostra il terminale che si applica al periferico, a destra quello che va al PC.**

Volendo collegare più dispositivi USB si può far uso di HUB a più porte, una specie di spina multipla che si collega al PC che presenta più connettori simili a quello mostrato in fig.12. In fig.14, infine, si mostra un HUB USB a 4 porte.



**Fig.14. – Hub USB a 4 porte.**

### 3.4. Interfaccia IEEE1394

Lo standard IEEE1394 fu adottato nel 1995 dalle specifiche fornite da un gruppo di aziende elettroniche tra cui Apple, Compaq e Sony.

Questo tipo di interfaccia, di tipo seriale e bidirezionale, fu concepita nel 1986 dalla Apple che la denominò "FireWire" per le sue particolari caratteristiche di velocità. Successivamente la Sony introdusse delle varianti denominando il collegamento "iLink". Lo standard IEEE1394 comprende sia le specifiche "Fire Wire" che "iLink".

Attualmente sono in commercio numerosi prodotti che si avvalgono dello standard IEEE1394 per la trasmissione dei dati digitali: sistemi di elaborazione digitale video, videocamere digitali, riproduttori audio digitali, personal computer, hard-disk, stampanti, ecc.

Tale standard consente velocità di trasferimento fino a 400Mbps, corrispondente a 50Mbyte/s, seconda solamente all'interfaccia ultrawide SCSI 3 ed alla nuova USB 2.0 a 480Mbps.

Tra le principali caratteristiche ricordiamo:

- possibilità di collegamento e scollegamento fisico a caldo;
- capacità di miscelare su una singola linea dati in tempo reale sincroni ed asincroni;
- capacità di utilizzare simultaneamente dispositivi sia a bassa che alta velocità sulla stessa connessione fino ad un numero massimo pari a 63;
- i sistemi operativi Windows 98SE, Windows ME, Windows 2000, Linux, Unix e Sun contengono i driver per la gestione dell'IEEE1394.

Alcuni costruttori stanno elaborando soluzioni di reti di computer "peer to peer" sfruttando tale interfaccia per le comunicazioni bidirezionali ad alta velocità. Presto saranno disponibili velocità a 800-1600-3200 Mbps su cavo, fibre ottiche e radiofrequenza.

Per i PC l'interfaccia IEEE1394 è disponibile sotto forma di schede su PCI con connettori del seguente tipo:

- a 4 pin senza alimentazione: due coppie di cavi intrecciati, una per la trasmissione ed una per la ricezione;
- a 6 pin con alimentazione: i due pin in più forniscono la massa e l'alimentazione da 8V a 30V con capacità di pilotaggio in corrente fino a 1.5A.

La lunghezza massima del cavo di collegamento è di 4 metri; nel caso di collegamento in serie di più dispositivi si può arrivare ad una distanza massima pari a 72 metri poiché i dispositivi collegati in "daisy chain" sono dei ripetitori che acquisiscono il segnale in arrivo e lo ritrasmettono rigenerato al dispositivo collegato in catena.

In fig.15 si mostra il tipico collegamento tra vari dispositivi "FireWire".

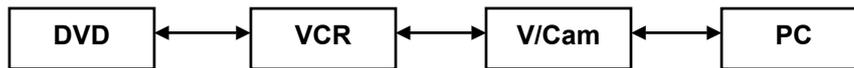


Fig.15. Collegamento fra vari dispositivi che utilizzano lo standard IEEE1394.

Una tipica scheda PCI IEEE1394 presenta 3 attacchi esterni ed uno interno, è di tipo “plug & play”, supporta fino a 15W di potenza e monta due circuiti integrati della Texas Instruments TSB41LV03A e TSB12LV23.

Si mostra in fig.16a una tipica scheda IEEE1394 e in fig.16b il connettore a 4 poli del cavo di collegamento.



a)



b)

Fig.16. – a) Scheda IEEE1394 con tre attacchi esterni; b) connettore a 4 poli.

### 3.5. Confronti

In tabella 6 si mostra il confronto tra le precedenti ed altre interfacce rispetto solo alla velocità di funzionamento e non tiene conto di altre caratteristiche.

Tabella 6

Tipo	MByte/secondo	Tipo	MByte/secondo
Modem 56K	0.007	Ultra Wide SCSI	40
Seriale	0.0137	IEEE1394	50
ISDN 128K	0.016	USB 2.0	60
ADSL	1	IDE ATA-66	66
Ethernet	1.25	Ultra 3 Wide SCSI	80
USB	1.5	PCI (32bit/33MHz)	133
Parallela (EPP)	2	Ultra 160 Wide SCSI	160
Fast Ethernet	12.5	PCI (64bit/33MHz)	266
BUS ISA	16	AGP 2X	533
Wide SCSI 2	20	AGP 4X	1064
IDE ATA-33	33	DDR SRAM 200MHz	1600

## 4. Circuiti impiegati nelle interfacce

Come si è detto all'inizio del capitolo, una interfaccia deve garantire una compatibilità elettrica e per questo sono necessari circuiti adattatori di impedenza e traslatori di livello. Per quanto riguarda, in particolare, il collegamento al bus di un microprocessore, l'interfaccia deve garantire la possibilità che un dispositivo collegato al bus sia posto in "tri-state" cioè in condizioni tali da non influenzare altri dispositivi che in quel momento stanno utilizzando il bus. Le interfacce dei dispositivi di uscita devono prevedere degli elementi di memoria, realizzabili con flip-flop, in grado di conservare il dato che il microprocessore ha depositato sul bus. In questo modo si fornisce la possibilità al dispositivo di uscita di poter leggere il dato in un momento successivo a quello in cui la CPU lo ha inserito sul bus dati.

La maggior parte delle interfacce contengono all'interno dei circuiti integrati programmabili per aumentare la versatilità e diminuire la complessità circuitale.

### 4.1. Traslatori di livello

Sono utilizzati quando si deve modificare un valore di tensione per la rappresentazione dello 0 logico e dell'1 logico.

Nell'interfaccia seriale RS-232, ad esempio, si deve modificare il livello logico 1 TTL, tipicamente compreso tra 2V e 5V, in un valore negativo di tensione compreso tra -5V e -15V e, viceversa, il livello logico 0 TTL, tipicamente compreso tra 0V e 0.8V, in un valore positivo di tensione compreso tra +5V e +15V. Una semplice soluzione alternativa a quella esaminata nel paragrafo 3.2.3. consiste nell'impiego di un comparatore invertente, a doppia alimentazione  $\pm 12V$  o  $\pm 15V$ , con tensione di riferimento pari a 1.5V come si mostra in fig.17.

$R_1$  assicura una corrente di cortocircuito pari a  $15V/47\text{ Ohm} < 0.5A$ .

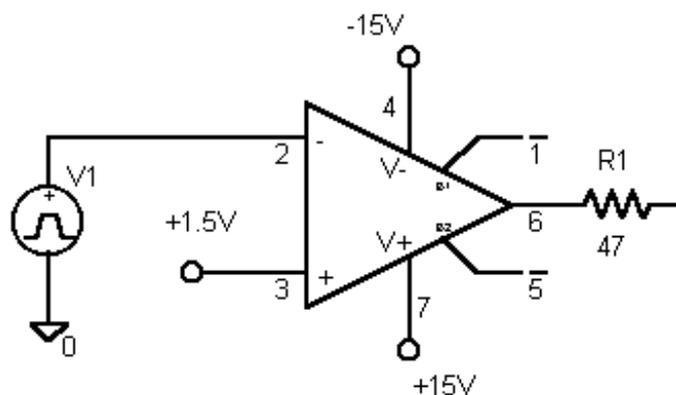


Fig.17. – Schema di principio di un traslatore di livello TTL – RS232.

Per trasformare, invece, un livello RS-232 in TTL si può impiegare il circuito di fig. 18.

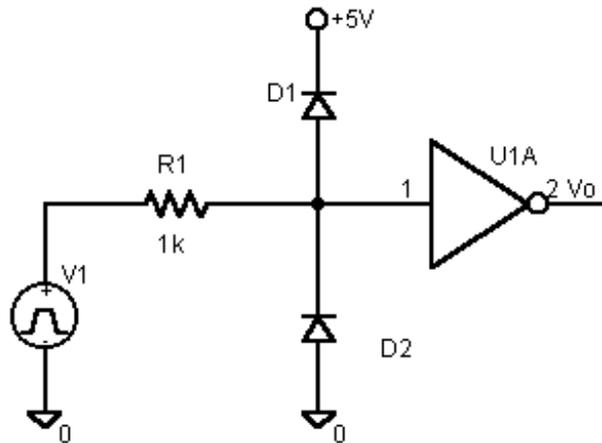


Fig.18. – Circuito per traslare un livello RS-232 in TTL.

Se  $V_1 = 12V$  entra in conduzione  $D_1$  assicurando all'ingresso della porta NOT il valore  $V_1 = 5.6V$  e quindi  $V_O = 0$  (livello 0 TTL).

Se  $V_1 = -12V$  entra in conduzione  $D_2$  e all'ingresso della porta NOT si ha:  $V_1 = -0.6V$  e quindi  $V_O = 3.5V$  (livello 1 TTL).

#### 4.2. Buffer TRI-STATE

Il collegamento di un dispositivo periferico di input al bus dati di un sistema a microprocessore deve avvenire interponendo dei buffer tri-state non invertenti. L'ingresso di abilitazione di ciascun buffer consente il collegamento dei dati del periferico di input al bus dati del microprocessore. Ponendo l'ingresso di abilitazione al livello logico opposto i buffer portano le loro uscite nello stato tri-state e quindi, in pratica, isolano elettricamente il dispositivo di input dal bus. In fig. 19 si mostra un tipico collegamento.

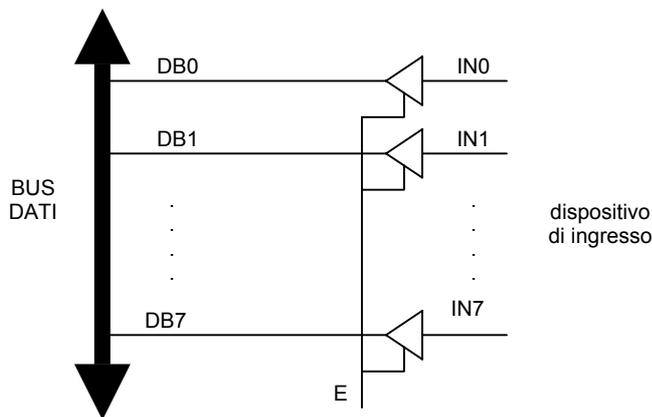


Fig.19. – Collegamento di un dispositivo di ingresso al BUS dati di un microprocessore.

Per questo tipo di collegamento è sufficiente usare l'integrato 74LS244 che contiene due gruppi indipendenti di 4 buffer tri-state a trigger di Schmitt. Se gli ingressi di abilitazione di ciascun gruppo:  $\overline{1G}$  e  $\overline{2G}$  sono tenuti al livello alto l'uscita di ciascun buffer ha lo stesso livello logico del proprio ingresso; se, invece gli ingressi di abilitazione sono al livello basso, le uscite dei buffer si portano in tri-state (stato di alta impedenza) indipendentemente dal livello logico applicato in ingresso. Si riporta, in fig.20, la piedinatura e lo schema interno dell'integrato 74LS244.

SN54LS', SN54S' . . . J OR W PACKAGE  
SN74LS', SN74S' . . . DW OR N PACKAGE  
(TOP VIEW)

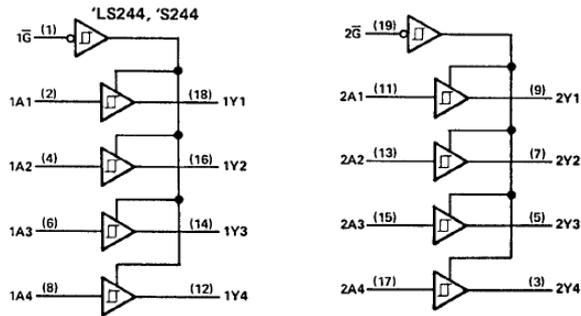
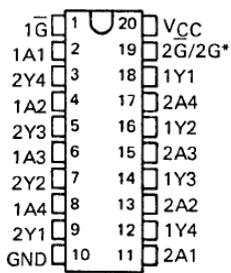
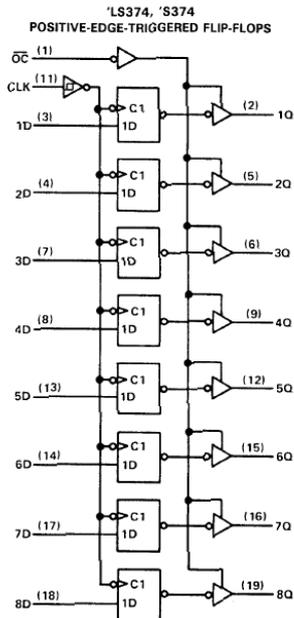


Fig.20. – Piedinatura e schema interno del 74LS244.

### 4.3. Flip-flop D con uscite tri-state



Il collegamento di un dispositivo di uscita si realizza interponendo dei flip-flop D che, memorizzando il dato, consentono al periferico la lettura anche in un momento diverso da quello di invio da parte del bus dati.

Il circuito integrato 74LS374 contiene al suo interno 8 flip-flop D PET con uscita tri-state. Il clock è applicato agli omonimi ingressi di tutti i flip-flop.

La linea  $\overline{OC}$  (OUTPUT CONTROL), se alta, tiene in tri-state le uscite del dispositivo.

Si mostra in fig.21 la piedinatura e lo schema interno del 74LS374.

Fig.21. – Piedinatura e schema interno del 74LS374.

#### 4.4. Porte di I/O programmabili di tipo parallelo

Sono circuiti integrati a larga scala di integrazione (LSI) costituiti da due o più porte ad 8 bit, indicate, generalmente, porta A, porta B, ecc..

Ogni bit di ciascuna porta è programmabile via software come bit di uscita o di ingresso.

L'integrato si inserisce tra il bus del microprocessore ed il dispositivo periferico a cui collegarsi.

In fig.22 si mostra lo schema funzionale semplificato di un integrato contenente tre porte parallele programmabili del tipo PPI8255 (Programmable Peripheral Interface) della Intel.

Esso trova impiego nei circuiti di interfaccia parallela, come ad esempio, la Centronics, utilizzata ampiamente per collegare una stampante.

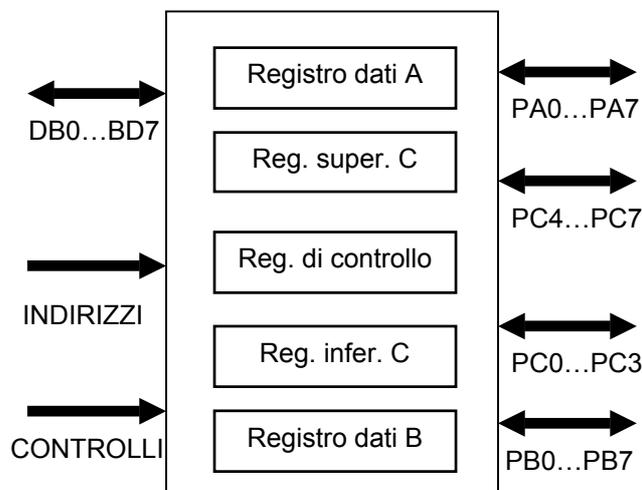


Fig.22. – Schema a blocchi semplificato del dispositivo di I/O programmabile parallelo PPI8255.

Le linee mostrate a sinistra consentono di interfacciare il dispositivo al BUS di un microprocessore. In particolare le linee che si collegano al BUS dati sono bidirezionali e di tipo tri-state. Le linee collegate al BUS indirizzi sono solo due: A1 e A0 che consentono di individuare quale delle tre porte si sta utilizzando. Le linee collegate al BUS controlli sono quella di abilitazione, di scrittura, lettura e reset. L'integrato, a 40 pin, è alimentato a +5V.

Il *registro dati* contiene il dato effettivo da scambiare tra periferico e microprocessore. Il *registro di controllo*, a sola scrittura e dotato di 8 bit, permette di stabilire le modalità di funzionamento delle porte.

La porta C è suddivisa in due parti: la porta C superiore è configurata come la porta A mentre la porta C inferiore allo stesso modo della porta B.

Si mostra, in fig.23, la piedinatura del PPI8255.

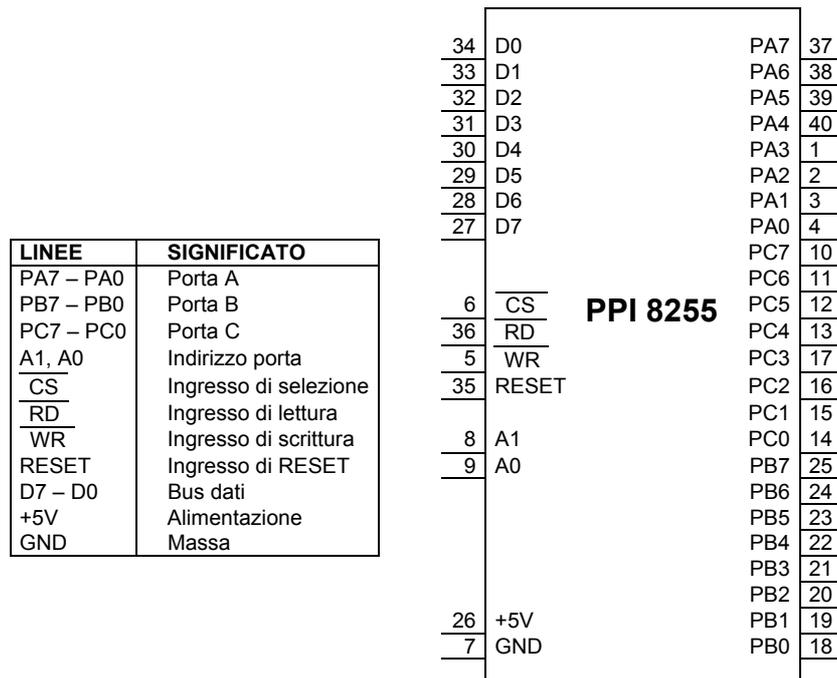


Fig.23. – Piedinatura e significato delle linee del PPI8255.

Esaminiamo le modalità di funzionamento delle porte che l'impostazione dei bit del registro di controllo consente di realizzare.

D7 definisce il modo di funzionamento dell'8255; in particolare se:

D7 = 1, l'integrato funziona in modalità 0, 1 o 2;

D7 = 0, l'integrato funziona in modalità set/reset.

La coppia D6, D5 individua la modalità operativa della porta A e dei 4 bit superiori della porta C; in particolare si ha:

D6 D5 = 00 Modo 0 Ingresso/Uscita;

D6 D5 = 01 Modo 1 Ingresso/Uscita con strobe;

D6 D5 = 1X Modo 2 Bidirezionale con strobe solo per la porta A.

D4 definisce il modo di funzionamento della porta A; in particolare:

D4 = 1 Ingresso;

D4 = 0 Uscita.

D3 definisce il funzionamento della porta C superiore; in particolare:

D3 = 1 Ingresso;

D3 = 0 Uscita.

D2 definisce il modo operativo della porta B; in particolare:

D2 = 1 Modo 1 Ingresso/Uscita con strobe;

D2 = 0 Modo 0 Ingresso/Uscita.

D1 definisce il funzionamento della porta B; in particolare:

D1 = 1 Ingresso;

D1 = 0 Uscita.

D0 definisce il modo di funzionamento della porta C inferiore; in particolare:

D0 = 1 Ingresso;

D0 = 0 Uscita.

Nel modo 0 ciascuna porta può essere programmata come porta di uscita o come porta di ingresso. Le porte di ingresso si comportano come buffer tri-state mentre le porte di uscita si comportano come dei flip-flop D.

Nel modo 1 il trasferimento dei dati avviene sotto il controllo dei segnali di strobe emessi o ricevuti dagli 8 bit della porta C.

Il modo 2 funziona solo per la porta A programmata per funzionare come porta di ingresso e di uscita, ovviamente in tempi diversi. Il trasferimento dei dati avviene sotto il controllo dei segnali di strobe emessi od accettati attraverso 5 bit della porta C.

#### **4.5. Porte di I/O programmabili di tipo seriale**

Sono circuiti integrati a larga scala di integrazione (LSI) noti con le sigle UART e USART (Trasmettitore Ricevitore Universale Asincrono il primo e Sincrono e Asincrono il secondo) adatti ad interfacciare i dati paralleli di un microprocessore con quelli seriali di apparecchiature come, ad esempio, il modem, il mouse o la stampante seriale secondo lo standard RS232.

Questi integrati posseggono al loro interno un registro a scorrimento PISO (Ingresso Parallelo ed Uscita Seriale) per serializzare i dati emessi dal microprocessore e da un registro a scorrimento SIPO (Ingresso Seriale ed Uscita Parallela) per parallelizzare i dati provenienti da un dispositivo seriale come, ad esempio, il modem.

Alcuni UART posseggono un unico registro PIPO universale (Ingresso ed Uscita Paralleli) in grado di comportarsi, all'occorrenza, come PISO o come SIPO.

Come è noto, la trasmissione seriale può essere sincrona o asincrona. La trasmissione sincrona può avvenire ad un filo o due fili oltre a quello di massa.

Nel primo caso, oltre ai bit di dati, si inviano particolari bit di sincronismo che consentono al ricevitore l'estrazione del clock. Nel secondo caso il secondo filo è utilizzato per la trasmissione del clock.

Nella trasmissione asincrona si trasmette un blocco di bit di 5, 7 o 8 bit (spesso un byte corrispondente ad un carattere nel codice ASCII) preceduti da un bit di start e seguiti da uno o più bit di stop. È possibile aggiungere, per ogni

blocco, un bit supplementare di parità pari o dispari per consentire al ricevitore il controllo della correttezza dei bit ricevuti. I bit sono trasmessi con una frequenza che deve essere la stessa con la quale il ricevitore esegue la lettura del segnale ricevuto.

Tutte queste informazioni, ovviamente, devono essere note prima di iniziare la trasmissione vera e propria. Il trasmettitore ed il ricevitore, pertanto, preliminarmente avviano una fase di “negoziazione” in cui si scambiano le informazioni delle modalità di trasmissione.

Le regole che governano la trasmissione e la ricezione sono standardizzate e sono note come *protocolli di comunicazione*.

Un UART è costituito, fondamentalmente, da tre sezioni:

- ❑ trasmettitore
- ❑ ricevitore
- ❑ controllo

Il trasmettitore riceve dal microprocessore il dato parallelo e lo trasforma, grazie al registro PISO, in forma seriale. Il ricevitore esplica la funzione opposta: riceve il dato seriale e lo parallelizza grazie al registro SIPO. La sezione di controllo è collegata al BUS di controllo del microprocessore e fornisce i segnali di stato. Le altre funzioni svolte sono: controllo del bit di start, di stop e parità. La massima velocità di funzionamento è, però, limitata a 115Kbps.

In fig.24 si mostra lo schema a blocchi semplificato di un UART.

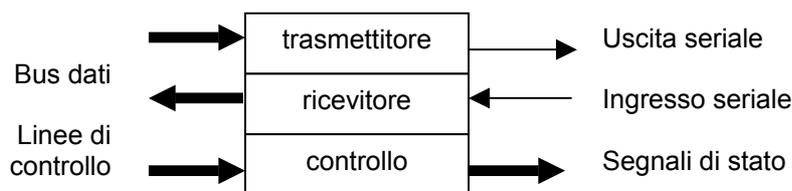


Fig.24. – Schema a blocchi semplificato di un UART.

Gli USART utilizzano una sequenza di bit che individua l’inizio della trasmissione (stringa di sincronismo). Il blocco di byte di dati viene trasmesso di seguito senza utilizzare il bit di start, i bit di stop ed il bit di parità per ciascun byte.

In coda al blocco vengono trasmesso 16 o 32 bit di controllo per consentire al ricevitore di appurare la correttezza della trasmissione. I bit di controllo costituiscono il CRC (Codice Ciclico di Ridondanza) e si genera prelevando i bit di resto del rapporto tra la stringa di bit da trasmettere ed una particolare stringa di bit a 16 o 32 bit, definita *polinomio generatore*. Il ricevitore, dopo aver acquisito il blocco di dati, esegue la stessa operazione del trasmettitore e confronta il CRC ricevuto con quello che ha generato. Se essi sono identici la trasmissione è avvenuta correttamente. In realtà si ottiene lo stesso resto per tutti i numeratori che differiscono da quello dato per un multiplo intero del denominatore. Poiché è assai improbabile che un eventuale errore di trasmissione fornisca una configurazione tale da rendere il resto del rapporto identico a quello ottenuto in assenza di errore, il



La sezione "modem control" è utilizzata quando l'integrato è collegato ad un modem. Le linee indicate in tale sezione si connettono alle omologhe del modem.

L'8251A presenta una sezione "trasmettitore" distinta da quella "ricevitore" ed esse possono lavorare ad una frequenza di clock diverse pari a 1, 1/16, 1/64 della frequenza applicata sul piedino TxC (Trasmitter Clock) e RxC (Receiver Clock). La selezione della frequenza di operazione viene programmata prima di avviare la comunicazione.

Il trasmettitore presenta due registri: il primo serve per la memorizzazione del dato da trasmettere; il secondo è un registro a scorrimento di tipo PISO la cui uscita seriale è rappresentata dalla linea TxD (Trasmitter Data). Alla fine della trasmissione di un carattere, la linea TxRDY (Trasmitter Ready) si porta al livello alto. La linea TxE (Trasmitter Empty) si porta al livello alto nel caso in cui il microprocessore tarda ad inviare un successivo dato.

Il ricevitore acquisisce il dato seriale dalla linea RxD (Receiver data) e lo trasmette in un registro a scorrimento di tipo SIPO che parallelizza il dato e successivamente lo trasferisce in un registro buffer. La linea SYNDET può essere utilizzata sia come ingresso che uscita durante la ricezione sincrona e serve a sincronizzare l'USART come ricevitore.

L'inizializzazione dell'USART è realizzata da un'istruzione di "modo" scritta nel registro di comando. Tale istruzione indica il tipo di comunicazione: sincrona o asincrona, lunghezza del carattere (5, 7 o 8 bit), presenza o assenza del bit di parità pari o dispari, numero di bit di stop (1, 1 e 1/2, 2), caratteri di sincronismo (nella trasmissione sincrona), ecc.

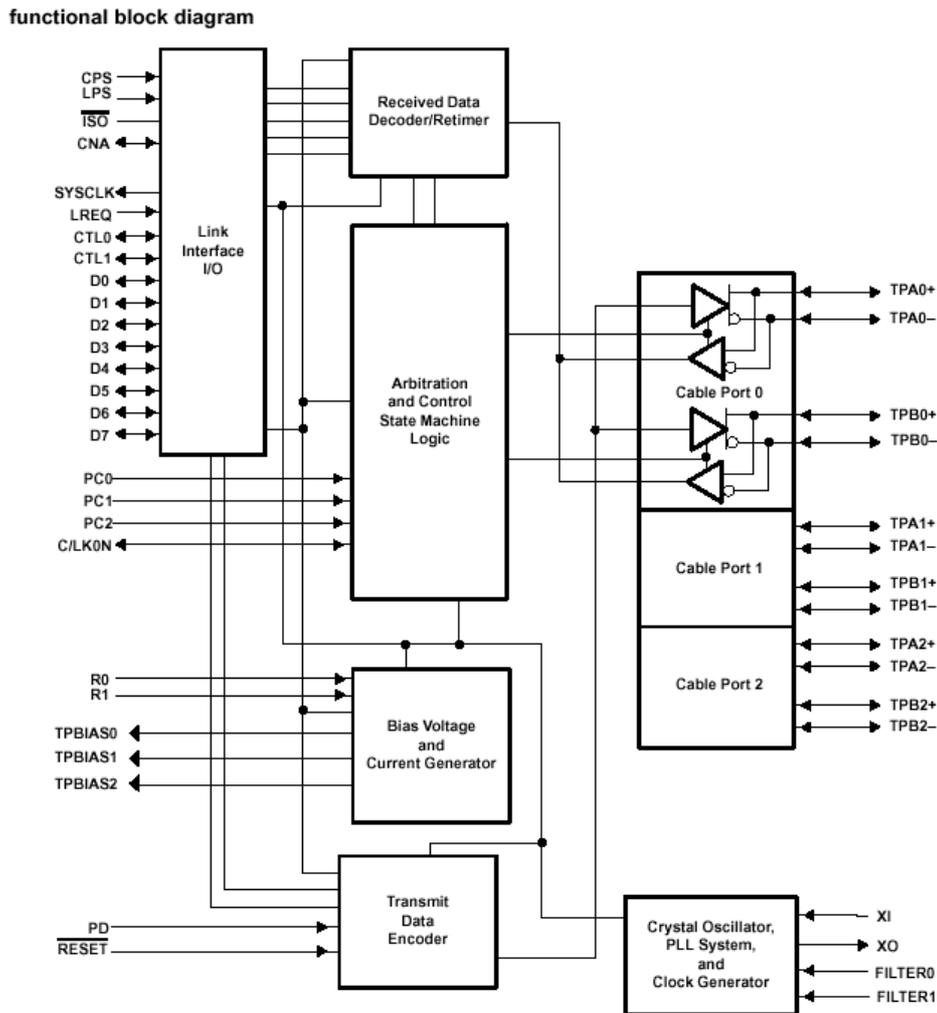
Il dispositivo presenta un registro di stato a 8 bit il cui contenuto può essere letto dal microprocessore ed indica con i suoi bit lo stato di funzionamento dell'USART.

Nei moderni PC è utilizzato l'USART 16550 che consente le comunicazioni veloci potendo supportare velocità fino a 115200bps. Esso, inoltre, contiene un buffer di 16 byte strutturati a coda (FIFO= First In First Out) che funziona da "cuscinetto" tra il computer e il modem consentendo di adeguare la velocità all'apparato più lento.

#### 4.6. TBS41LV03A

È un circuito integrato a 80 pin, disposti sui quattro lati di un quadrato, alimentato a 3.3V, che svolge la funzione di trasmettitore e ricevitore per l'implementazione di un nodo a tre porte ciascuna delle quali è in grado di pilotare un cavo secondo lo standard IEEE1394. Ciascuna porta contiene due "line driver" di tipo differenziale per cui il cavo che si collega a ciascuna porta è costituito da 4 fili. I *transricevitori* contengono circuiti per il monitoraggio delle condizioni della linea, per l'*inizializzazione*, per l'*arbitraggio* e per la *ricezione* e *trasmissione* di pacchetti. Il circuito integrato è progettato per interfacciarsi ad un "line layer controller" (LLC) come, ad esempio, il TSB12LV23.

In fig. 26 si mostra lo schema funzionale del circuito integrato.



**Fig.26. – Schema funzionale del TSB41LV03A della Texas Instruments.**

Per il funzionamento richiede un cristallo di quarzo funzionante a 24.576MHz. Un oscillatore interno pilota un PLL interno che genera un clock a 393.216MHz. Due divisori di frequenza per due determinano i clock a 196.608MHz e 98.304MHz.

Questi tre valori sono quelli che determinano il funzionamento dell'interfaccia IEEE1394 a 400Mbps, 200Mbps e 100Mbps.

Per ulteriori dettagli tecnici si rimanda alla consultazione dell'esauriente datasheet.

Si riporta in fig.27 il collegamento tra una delle tre porte ed il cavo esterno contenente al suo interno due coppie di fili A e B.

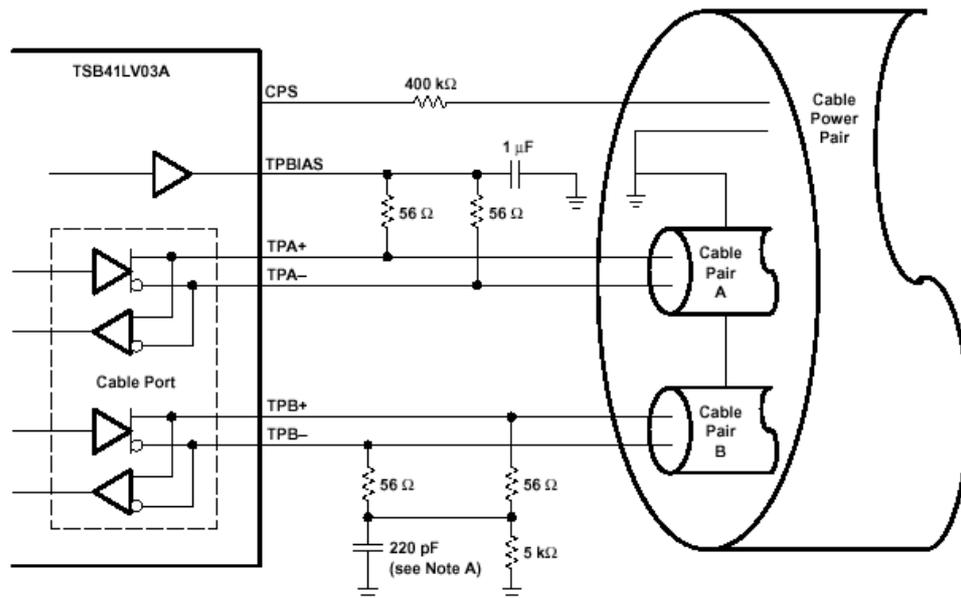


Fig. 27. – Collegamento di una porta dell'integrato ad un cavo esterno contenente due coppie di fili.

## 5. Schede grafiche

Hanno il compito di inviare al monitor l'immagine generata dal computer.

Le parti costitutive fondamentali sono 3:

- la RAM che contiene i dati digitali da presentare al monitor;
- il convertitore digitale/analogico (RAMDAC) per trasformare le informazioni custodite nella RAM presente su scheda grafica dalla forma digitale alla forma analogica;
- la GPU (Graphics Processing Unit) un particolare circuito integrato che governa le complesse operazioni da svolgere: è, in pratica, un microprocessore dedicato alla gestione della grafica.

La maggior parte delle schede grafiche odierne eseguono speciali e potenti istruzioni grafiche in grado di elaborare effetti tridimensionali necessari nei recenti videogiochi e nelle simulazioni grafiche.

Nei primi personal computer il monitor era monocromatico e funzionava esclusivamente in modalità testo. La gestione del monitor era affidato ad un circuito integrato controllore di CRT (Tubo a Raggi Catodici) e la memoria video, di pochi kilobyte, risiedeva in memoria centrale.

Con l'avvento della grafica (possibilità di suddividere il monitor in una fitta matrice bidimensionali di puntini luminosi) e del colore, i computers sono stati dotati di schede video sempre più potenti, cioè in grado di rappresentare sul monitor un numero maggiore di punti (pixel) e di colori con velocità di refresh sempre più elevate.

Si pensi che la prima scheda grafica a colori, la CGA (Color Graphic Adapter) supportava una risoluzione grafica di 300x200 pixel a 4 colori e quindi un'occupazione di memoria video di soli 16Kbyte.

La scheda grafica VGA standard (Video Graphics Adapter), anch'essa ormai obsoleta, gestisce una risoluzione grafica di 640x480 pixel a 16 colori per cui sono necessari 150Kbyte a bordo della scheda. Infatti per rappresentare uno dei 16 colori sono necessari 4 bit per cui il numero di bit necessari sono:

$640 \times 480 \times 4 = 1.228.800$  che corrispondono a 153.600byte e quindi a 150Kbyte.

Con l'avvento della SVGA (Super VGA) si possono gestire risoluzioni grafiche spaziali e cromatiche notevolmente superiori per cui la memoria a bordo della scheda grafica risulta sicuramente superiore a 1Mbyte. Attualmente sono disponibili schede grafiche a 8 e più Mbyte che montano potentissimi processori grafici che consentono una corretta visione dei filmati e di immagini 3D.

Per ottenere ciò non basta soltanto disporre di elevata memoria a bordo della scheda grafica: è necessario che i dati transitino dal microprocessore al monitor, via scheda grafica, ad elevatissima velocità.

Un tempo l'adattatore video si innestava in uno slot d'espansione ISA a 16 bit. Successivamente, per accelerare il trasferimento dei dati dalla scheda madre al monitor, sono stati introdotti connettori PCI capaci di far transitare dati a 32 bit con velocità di 33MHz (33.3MHz, per la precisione) corrispondente ad un trasferimento dati pari a 133Mbyte/s. (infatti:  $32 \cdot 33.3 / 8 = 133$ ).

Sui recenti personal computer è presente il connettore AGP (Advanced Graphic Port) che lavora alla frequenza di 66MHz su dati a 32 o 64bit.

In tal modo il suo throughput è da 2 a 4 volte più elevato rispetto a quello della PCI perché risulta raddoppiata la velocità ed, eventualmente, anche la capacità di trasferimento.

Per comprendere meglio i concetti di *risoluzione spaziale*, *risoluzione cromatica* e *refresh*, nei successivi paragrafi si tratterà di tali argomenti.

## 5.1. Risoluzione spaziale

La risoluzione spaziale rappresenta il numero di punti che una immagine può contenere. Una immagine ha una struttura a matrice di punti ed è organizzata in righe e colonne. Normalmente il rapporto tra il numero di colonne e il numero delle righe è 4/3.

Le risoluzioni grafiche tipiche sono: 640x480 (480 righe e 640 colonne; ovvero, ogni riga è costituita da 640 punti detti pixel), 800x600, 1024x768, 1280x1024, 1600x1200.

Osserviamo che il passaggio da una risoluzione a quella più elevata si ottiene moltiplicando ciascuna dimensione per 1.25, cioè aumentando la dimensione del 25%.

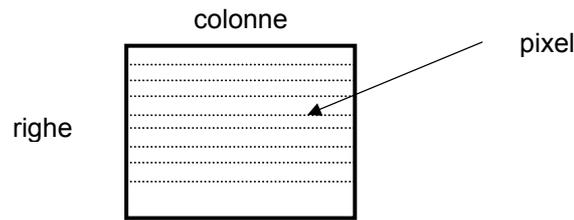


Fig.28. - Rappresentazione semplificata dei pixel sullo schermo di un monitor.

## 5.2. Risoluzione cromatica

La risoluzione cromatica rappresenta la quantità di informazione associata al pixel.

Il pixel rappresentato con un solo bit può assumere due soli valori: 0 o 1 ed il punto sullo schermo potrà essere solamente acceso o spento.

Se ad ogni pixel si associano 8 bit, ogni punto potrà assumere uno dei  $2^8$  livelli di grigio compresi tra il bianco e il nero. Se si utilizza un monitor a colori i 256 livelli possono essere rappresentati da altrettanti colori.

Con 16 bit un pixel può assumere uno tra  $2^{16}=65536$  colori. In tal modo l'immagine avrà una colorazione più definita e quindi più reale rispetto al caso di soli 256 colori.

Con 24 bit, infine, sono possibili oltre 16 milioni di colori ( $2^{24}$ ) e, considerando che i colori fondamentali sono il rosso (red R), verde (green G) e blu (blue B), i 24 bit si possono pensare organizzati in 3 byte, uno per ciascun canale di colore.

Ogni colore fondamentale, quindi, possiede 256 livelli: dall'assenza di colore al colore saturo. La composizione dei 3 colori fondamentali, in opportuni valori percentuali, fornisce uno dei 16 milioni di colori.

Evidentemente l'elevata risoluzione spaziale e cromatica richiedono ingenti quantità di memoria e impegnano notevolmente il sistema poiché si devono "movimentare" grandi quantità di dati.

Si riporta, nella seguente tabella 8, la quantità di memoria, in byte, necessaria per la rappresentazione di immagini nelle varie risoluzioni spaziali e cromatiche.

Tabella 8 - Dimensionamento delle varie risoluzioni grafiche

Risoluzione	n° livelli n° colori	Calcoli	Byte richiesti
Monocromatica 640x480	1 livello	$(640 \cdot 480 \cdot 1) / 8$	38.4 Kbyte
Monocromatica 640x480	256 livelli di grigio	$(640 \cdot 480 \cdot 8) / 8$	300 Kbyte
Colori 640x480	16 colori	$(640 \cdot 480 \cdot 4) / 8$	150 Kbyte
Colori 640x480	256 colori	$(640 \cdot 480 \cdot 8) / 8$	300 Kbyte
Colori 640x480	65536 colori	$(640 \cdot 480 \cdot 16) / 8$	600 Kbyte
Colori 640x480	16M colori	$(640 \cdot 480 \cdot 24) / 8$	1.2 Mbyte
Colori 800x600	16 colori	$(800 \cdot 600 \cdot 4) / 8$	235 Kbyte
Colori 800x600	256 colori	$(800 \cdot 600 \cdot 8) / 8$	469 Kbyte
Colori 800x600	65536 colori	$(800 \cdot 600 \cdot 16) / 8$	938 Kbyte
Colori 800x600	16M colori	$(800 \cdot 600 \cdot 24) / 8$	1.4 Mbyte
Colori 1024x768	16 colori	$(1024 \cdot 768 \cdot 4) / 8$	384 Kbyte
Colori 1024x768	256 colori	$(1024 \cdot 768 \cdot 8) / 8$	768 Kbyte
Colori 1024x768	65536 colori	$(1024 \cdot 768 \cdot 16) / 8$	1536 Kbyte
Colori 1024x768	16M colori	$(1024 \cdot 768 \cdot 24) / 8$	2304 Kbyte

### 5.3. Monitor

I monitor utilizzati come terminali di *output* di un personal computer sono qualitativamente migliori dei comuni monitor televisivi.

Per evitare il fastidioso fenomeno dello sfarfallio dell'immagine, i monitor per PC presentano una frequenza di scansione verticale superiore a 70Hz.

Valori per i quali lo sfarfallio non si avverte sono 75Hz, 85Hz e 100Hz. Valori più bassi affaticano la vista.

La scelta della risoluzione spaziale ottimale dipende soprattutto dalle dimensioni in pollici del monitor e dalla risoluzione spaziale di quest'ultimo.

La misura in pollici è riferita alla lunghezza utile della diagonale di un monitor. Poiché un pollice equivale a 2.54cm, i monitor da 14, 15, 17 e 19 pollici hanno le dimensioni riportate in tabella 9.

Tabella 9 – dimensioni in cm. di vari monitor

Pollici	Diagonale (cm.)	Base (cm.)	Altezza (cm.)
14	35.56	28.45	21.34
15	38.10	30.48	22.86
17	43.18	34.55	25.91
19	48.26	38.61	28.96

Per un monitor a 14 pollici la risoluzione spaziale ottimale, nel caso di visualizzazione di una pagina di testo, è 640x480; quella a 800x600 è ritenuta da molti ai limiti della leggibilità e quella a 1024x768 non consente una buona lettura.

Per un monitor a 17 pollici la risoluzione spaziale ottimale è 800x600.

Di difficile leggibilità appare la schermata di testo con la risoluzione a 1024x768.

All'aumentare della risoluzione spaziale diminuiscono le dimensioni di ciascun carattere sullo schermo. Molti software, in tal caso, consentono di intervenire aumentando il fattore di zoom di visualizzazione dello schermo.

La risoluzione del monitor si misura in *dot pitch*, ovvero la dimensione di un pixel sullo schermo. Valori tipici sono 0.28mm ma sono in commercio monitor di ottima qualità con dot pitch di 0.26mm e 0.25mm.

In figura 29 si mostra il metodo di misura del dot pitch sugli schermi dei monitor tradizionali a triade di fosfori.

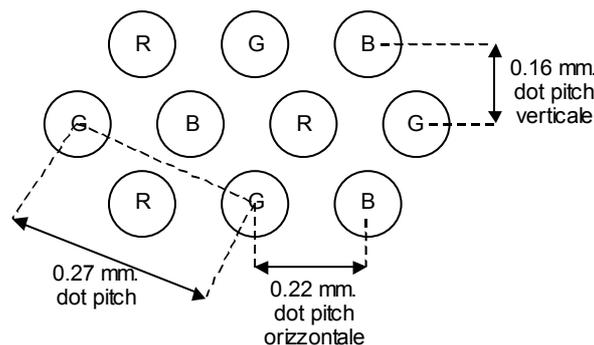


Fig.29. - Metodo per la misura del *dot pitch* di un monitor. Tre cerchi colorati, uno rosso, uno verde e uno blu, posti ai vertici di un triangolo equilatero, rappresentano la triade dei colori fondamentali e quindi un pixel sullo schermo. I cerchi indicati con R G e B, rispettivamente rappresentano il puntino rosso, verde (green), e blu.

#### 5.4. Caratteristiche di una scheda grafica

Una moderna scheda video presenta al suo interno un processore grafico (GPU= Graphics Processor Unit), funzionante ad una frequenza di clock compresa tra 100 e 200MHz che svolge i calcoli matematici che consentono la generazione dell'immagine. La GPU contiene la RAMDAC che converte l'immagine digitale in formato analogico per il monitor. Alcune GPU, come la Matrox G450, contiene al suo interno due RAMDAC. La seconda serve per gestire un secondo monitor collegato alla scheda video.

Alcune sigle di recenti GPU: Nvidia Ge Force 3, ST Kyro II, Matrox G450.

Le GPU sopra menzionate sono acceleratori grafici 3D che consentono l'accelerazione del processo di elaborazione delle immagini tridimensionali mediante l'implementazione di particolari funzioni geometriche che eliminano pixel ridondanti o invisibili all'occhio umano, tecniche che velocizzano il trasferimento dei dati e che forniscono rese più brillanti e quindi immagini che si susseguono in modo fluido.

La scheda contiene anche la RAM video di tipo SDRAM (RAM Dinamica Sincrona), SGRAM (RAM Grafica Sincrona), più adatta per le operazioni di memorizzazioni grafiche, o le più moderne DDR (Double Data Rate). Le SDRAM hanno tempo d'accesso di 7ns o meno e possono funzionare con la stessa frequenza del processore grafico e cioè con un valore compreso tra 100 e 200MHz. Normalmente sono installate con capacità di 32Mbyte attraverso 4 schede da 8Mbyte ciascuna.

Le DDR tipiche hanno tempo d'accesso di 3.8ns e possono funzionare a 400MHz poiché sfruttano i due fronti del segnale di clock. Normalmente sono installate con capacità di 64Mbyte attraverso 8 schede da 8Mbyte ciascuna. Data l'elevata velocità di funzionamento, le schede grafiche con memorie DDR devono essere utilizzate su computer che implementano connettori AGP a 4X.

Anche la potenza dissipata è notevole per cui quasi tutte le schede grafiche montano uno o più dissipatori termici per facilitare lo smaltimento di calore prodotto per effetto Joule.

La modalità di rappresentazione dell'immagine sul monitor è la seguente:

- 1) Il software chiede la rappresentazione di una data immagine sul monitor;
- 2) I driver video controllano se tale immagine può essere accelerata dalla GPU; in caso affermativo l'immagine intera viene processata dalla GPU, viceversa tale funzione viene eseguita dalla CPU, comunque più lenta. l'immagine viene calcolata e memorizzata nella memoria della scheda video (*frame buffer*);
- 3) l'immagine viene prelevata dal *frame buffer* e convertita nella RAMDAC da formato digitale in quello analogico per essere così inviata al monitor; il monitor riceve il segnale analogico e l'immagine appare sullo schermo.

Molte schede grafiche, oltre al tipico connettore verso il monitor del PC, presentano l'uscita TV per un monitor televisivo; alcune presentano anche un ingresso TV a videofrequenza e/o radiofrequenza. In questo ultimo caso sono dotati di un sintonizzatore interno.

## 6. Schede audio

Per poter utilizzare le prestazioni audio è necessario possedere, all'interno del PC, una scheda audio che ha il compito, durante la fase di registrazione, di convertire il segnale elettrico analogico, proporzionale all'intensità del suono, in segnale elettrico digitale in grado di poter essere memorizzato nel PC.

In fase di riproduzione la scheda audio deve effettuare la conversione opposta, ossia deve trasformare i dati digitali in segnale analogico che, opportunamente amplificato dalla scheda audio o da un amplificatore esterno, viene inviato agli altoparlanti.

La scheda audio si inserisce in uno slot ISA a 16 bit (connettore nero) o di tipo PCI (connettore bianco).

Per il suo riconoscimento, la scheda audio utilizza indirizzi di memoria, indirizzi interrupt e canali DMA.

Presenta, nella parte posteriore del PC, una serie di prese Jack ed un connettore a D a 15 poli a cui collegare un Joystick o un dispositivo MIDI.

Si mostra in fig.30 lo schema a blocchi di una scheda audio (Sound Blaster a 16 bit).

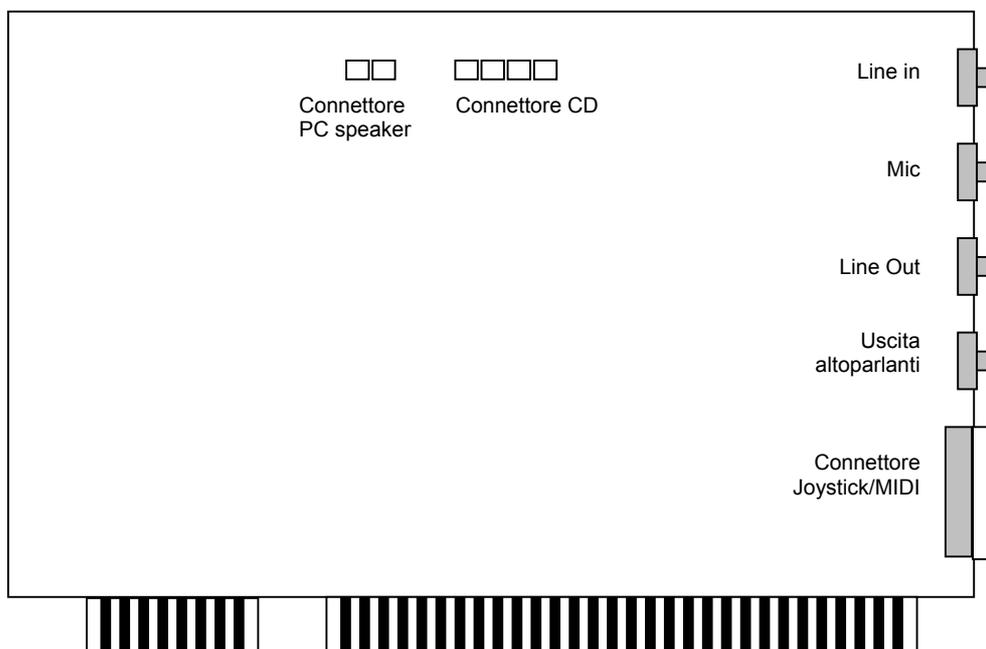


Fig.30 - Le prese jack ed i connettori della scheda audio.

Di seguito si indicano le funzioni dei connettori e delle prese Jack della scheda audio.

- ❑ **Line In:** permette di collegare alla scheda audio dispositivi quali riproduttori di nastri, ecc.;
- ❑ **Mic In :** consente di collegare un microfono per l'ingresso vocale;
- ❑ **Line Out:** consente di collegare amplificatori esterni;
- ❑ **Uscita altoparlanti:** consente di collegare altoparlanti. L'amplificatore interno ha una potenza massima di 4 Watt per canale su un altoparlanti da 4 Ohm (2 Watt su altoparlanti da 8 Ohm);
- ❑ **Connettore Joystick/MIDI:** permette di collegare un Joystick o un MIDI kit;
- ❑ **Connettore PC speaker:** consente di inviare agli altoparlanti esterni collegati alla scheda audio il suono che normalmente va allo speaker del PC;
- ❑ **Connettore CD:** consente l'ascolto, dagli altoparlanti collegati alla scheda audio, dei brani musicali contenuti nel CD ROM.

Una scheda audio è perfettamente in grado di digitalizzare un suono analogico, copiarlo in memoria o su disco rigido, leggerlo e riconvertirlo per l'ascolto. Per queste operazioni sono necessarie le prese jack precedentemente indicate.

Il connettore per Joystick non serve solo per i videogiochi ma permette di collegare tra loro vari strumenti musicali e di controllarli da un'unica tastiera-master secondo lo standard MIDI.

Lo standard MIDI, in riproduzione, consente di "suonare" simultaneamente un certo numero di strumenti, indicati col termine *voci*, grazie ad un circuito integrato interno alla scheda audio che funge da sintetizzatore FM (Modulazione di Frequenza) secondo la tecnologia Yamaha OPL2 e OPL3.

Fra le tante caratteristiche che determinano la buona qualità di una scheda audio occorre ricordare:

- ❑ **il numero di bit** nei quali viene convertito un valore analogico campionato; le prime schede audio erano a 8 bit ma ben presto apparvero quelle a 16 bit, sufficienti per una buona resa acustica, quelle a 32 bit e a 64 bit; maggiore è il numero di bit minore è l'errore di *quantizzazione* commesso nella conversione A/D e più la riproduzione risulta esente da fruscio;
- ❑ **la frequenza di campionamento** che rappresenta il numero di campioni convertiti da analogico a digitale in un secondo; maggiore è la frequenza di campionamento migliore è l'approssimazione della forma d'onda originale; tipicamente i valori sono 11KHz, 22KHz e 44.1KHz;
- ❑ gestione degli effetti tridimensionali, *surround* e *riverbero* in tempo reale, da 20 a 48 canali MIDI, capacità di compressione sonora.

Si riassumono nella tabella 10 le caratteristiche tecniche salienti delle comuni schede audio.

Tabella 10

Produttore	Aztech	Creative Lab	Creativ Lab	Gravis	TerraTec
Modello	Waverider pro 32-3D	SB AWE 64 Gold	Soundblaster 16	Ultrasound PnP Pro	Maestro 32/96
Plug & Play	NO	SI	SI	SI	SI
Sintesi Wav.	SI	SI	NO	SI	SI
ROM Wav.	1Mbyte	1Mbyte	NO	1Mbyte	4Mbyte
RAM standard	NO	4Mbyte	NO	512Kbyte	NO
Max RAM	NO	28Mbyte	NO	NO	NO
Slot per RAM	NO	Proprietarie	NO	2x30pin	NO
Voci HD	32	32	NO	32	32
Voci SW	NO	32	NO	NO	NO
Frequenza campionam.	48KHz	44.1KHz	44.1KHz	48KHz	48KHz
Effetti 3D	SI	SI	SI	SI	SI
Full duplex	SI	SI	SI	SI	SI
Soundblaster	SI	SI	SI	SI	SI
General MIDI	SI	SI	NO	SI	SI

## 6.1. Campionamento del segnale audio

Per convertire un'onda sonora analogica in segnale digitale, il computer deve essere in grado di misurarne l'ampiezza in istanti successivi periodicamente.

Ogni misura si chiama campione per cui la conversione A/D prende il nome di *campionamento* del suono.

Per la riproduzione di un suono digitale memorizzato nel PC è necessario effettuare la conversione D/A, cioè dalla forma digitale alla forma analogica.

### 6.1.1. Frequenza di campionamento

Più elevata è la frequenza di campionamento tanto maggiore è la rassomiglianza tra il segnale campionato e quello originario.

Spingere la frequenza di campionamento all'infinito pone un problema sul numero di campioni generati in un secondo. Se, ad esempio, la frequenza di campionamento fosse di 1 MHz, avremmo un milione di campioni al secondo per cui per un brano di 5 minuti (300 secondi) sarebbe convertito in 300 milioni di campioni. Ciascun campione, inoltre, utilizza uno o più byte per cui il citato brano potrebbe avere una dimensione anche superiore a 1G byte !!

Riducendo la frequenza di campionamento oltre un certo limite, però, non saremmo in grado di ricostruire fedelmente la forma d'onda di partenza.

Come fare ? Ci viene incontro il teorema del campionamento di Shannon che dice che *un segnale complesso avente massima frequenza  $f_{max}$  può essere completamente ricostruito se si effettua il campionamento ad una frequenza  $f_c$  almeno pari al doppio di  $f_{max}$* . In formula:

$$f_c \geq 2 \cdot f_{max}$$

Il limite  $f_c = 2 \cdot f_{\max}$  è puramente teorico perché per la ricostruzione del segnale di partenza è necessario un filtro passa-basso di ordine molto elevato, praticamente impossibile da realizzare.

In genere si campiona ad una frequenza maggiore di  $2 \cdot f_{\max}$ .

In telefonia viene destinata alla voce una banda di frequenza da 300Hz a 3.4KHz e gli impianti telefonici, per la conversione A/D, campionano alla frequenza di 8KHz (maggiore del doppio di 3.4KHz).

In campo telefonico non ha importanza la qualità audio ma la intelligibilità del messaggio.

In campo radiofonico, invece, è importante anche la qualità dell'audio. Infatti nella modulazione di ampiezza AM viene destinata all'audio una banda di 10KHz per cui il campionamento per la conversione A/D deve avvenire ad una frequenza superiore a 20KHz (in FM la banda è di 180KHz).

I brani audio incisi su CD, infine, per poter essere fedelmente riprodotti devono essere campionati ad una frequenza superiore a 40KHz poiché la banda acustica del nostro orecchio va 20Hz a 20KHz.

Si riassumono, nella seguente tabella 11, le tre frequenze di campionamento tipiche utilizzate per l'acquisizione digitale di segnali audio.

**Tabella 11**

Denominazione	Frequenza di campionamento	Numero di campioni per la durata di un minuto
Qualità telefono	11 KHz	660.000 campioni
Qualità radio	22 KHz	1.320.000 campioni
Qualità CD	44.1 KHz	2.646.000 campioni

### 6.1.2. Lunghezza del campione

Il dato campionato deve essere convertito in forma digitale. Se si destinano 8 bit ad ogni campione, la dinamica del segnale analogico viene suddivisa in:  $2^8 = 256$  livelli.

Nella ricostruzione si ottiene un tipico andamento a scalinata come si mostra in fig.31.

Assegnando ad ogni campione non 8 bit ma 16 bit la dinamica viene divisa in:  $2^{16} = 65536$  livelli.

La conversione è più precisa ma il file prodotto ha dimensione doppia. In genere, tuttavia, si preferisce un campionamento a 16 bit rispetto a 8 bit per ridurre il rumore di fondo. In caso di campionamento stereo si generano due campioni per volta.

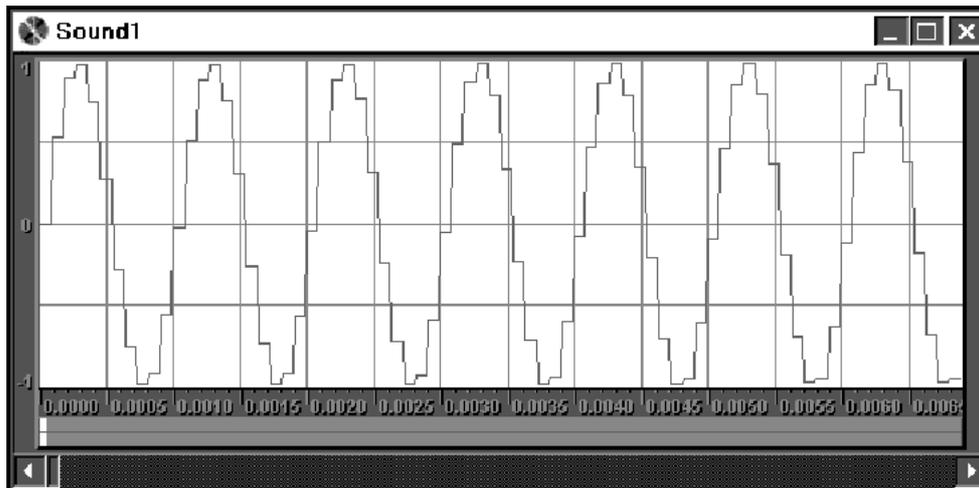


Fig.31. - Onda sinusoidale di ampiezza 1V e frequenza 1KHz campionata a 11KHz con campioni a 8 bit in formato mono.

Per la produzione di brani audio in formato CDA (CD Audio) da registrare su CD-ROM, la scelta obbligata è 44.1KHz a 16bit stereo. In questa modalità di funzionamento un minuto di riproduzione occupa  $2.646.000 \times 4 = 10.584.000$  byte cioè poco più di 10Mbyte. Un brano di durata 5 minuti, perciò, viene memorizzato in 50Mbyte. Se il brano è memorizzato su CD-ROM, il driver deve leggere il file ad una velocità:

$$44.1 \times 4 = 176.400 \text{ byte/secondo.}$$

Questo valore rappresenta la *velocità base* dei lettori di CD-ROM. Un lettore di CD-ROM si dice a *doppia velocità*, o 2x, se è in grado di leggere i dati ad una velocità doppia del valore 176.400 (approssimato a 150Kbyte). I lettori di CD-ROM più recenti sono del tipo 50x ed oltre, cioè possono leggere dati ad una velocità 50 volte superiore a quella necessaria per la riproduzione di un brano audio campionato a 44.1KHz, 16 bit, stereo. In realtà un sistema automatico calibra la massima velocità in funzione della leggibilità del CD-ROM.

## 6.2. Formati dei files audio

Di seguito si elencano alcuni formati audio di maggior utilizzo e riconosciuti dalla maggior parte dei programmi di elaborazione del suono.

### 6.2.1. WAV

Il formato WAV è stato sviluppato da IBM e MicroSoft e può contenere campioni ottenuti con qualsiasi frequenza, risoluzione e numero di canali e può includere uno dei tanti stili di compressione. Quando si esegue il salvataggio in formato WAV è possibile impostare gli attributi del file consistenti nella frequenza di campionamento, numero di bit, mono o stereo, legge  $\mu$  e legge A, MPEG layer-3, Creative ADPCM, ecc. Nel dubbio conviene non modificare l'impostazione automatica.

### 6.2.2. AU

Il formato AU è molto diffuso ed utilizzabile su varie piattaforme operative. Fu sviluppato dall'industria telefonica degli Stati Uniti col nome  $\mu$ -law, campiona un canale mono a 8 bit a 8KHz e consente una gamma dinamica più ampia rispetto ad altri suoni codificati a 8 bit e a 8 KHz. Presenta un fruscio di sottofondo piuttosto fastidioso.

### 6.2.3. RA

Il formato RealAudio RA, ascoltabile facendo uso direttamente del software riproduttore di tale formato, è molto utilizzato in Internet perché consente l'ascolto in tempo reale cioè durante la ricezione dei dati e non dopo la memorizzazione su disco dell'intero file. La sua qualità è di tipo radio monofonico.

Presenta un algoritmo molto complesso di tipo lossy (a perdita di informazioni), è molto più compatto del file WAV ma la qualità audio non è altrettanto buona.

### 6.2.4. MP3

È il formato audio più conveniente. Esso, infatti, consente di ridurre a 5Mbyte un file audio di 50Mbyte (corrispondente ad un brano di circa 5 minuti) con trascurabile decadimento delle prestazioni (si tratta di un formato lossy).

Il modello teorico è stato messo a punto da un italiano, l'ing. Leonardo Chiariglione della Telecom di Torino e successivamente le prime applicazioni pratiche sono state sperimentate in Germania presso il Fraunhofer Institut.

Il principio di funzionamento dell'algoritmo di compressione consiste nell'eliminare le informazioni ritenute non importanti ai fini della qualità del suono.

Per stabilire quali siano le informazioni da sacrificare si ricorre allo studio delle capacità percettive dell'orecchio umano. La banda passante dell'orecchio umano percepisce suoni con frequenza compresa tra 20Hz e 20KHz ma con sensibilità che dipende dalla frequenza e quindi non uniforme. L'orecchio è più sensibile alle frequenze comprese tra 700Hz e 6000Hz nelle quali si concentra la maggior parte delle informazioni. La sensibilità, inoltre, varia anche al variare dell'intensità del segnale. Più è bassa l'intensità sonora e più si accentua la differenza di sensibilità dell'orecchio alle diverse frequenze.

L'orecchio, inoltre, è sensibile anche all'*effetto di mascheramento* che consiste nel seguente fenomeno: se abbiano due suoni puri con frequenza poco diversa tra loro ma ampiezza differente, quello con intensità minore non viene percepito essendo mascherato dall'altro. Questo effetto viene sfruttato per ridurre tantissimo le dimensioni del file MP3 rispetto a quello originale. Anche l'effetto di mascheramento, come la sensibilità, dipende dalla frequenza.

La codifica MP3 funziona nel seguente modo. In primo luogo viene eseguita la FFT (Fast Fourier Transform) sul segnale digitale originale in modo da poterlo trattare nel dominio della frequenza piuttosto che in quello del tempo.

Il segnale così prodotto viene elaborato da un algoritmo che tiene conto del comportamento dell'orecchio umano in funzione della frequenza e viene suddiviso in 32 sottogamme di ugual ampiezza. In ciascuna sottogamma il segnale viene codificato con un numero diverso di bit a seconda dell'importanza della

sottogamma. Tutti i dati codificati e le informazioni relative al canale di provenienza vengono uniti e formano un unico flusso costante di dati. Nei punti ove il livello sonoro è elevato la banda audio viene riprodotta quasi per intero con l'esclusione delle frequenze più basse e più alte. Ove, invece, il livello sonoro è ridotto la banda viene ristretta attraverso un maggior taglio di frequenze basse ed alte. Le frequenze eliminate non sarebbero state percepite o lo sarebbero state con difficoltà. L'algoritmo ha causato, pertanto, perdita di informazione, riducendo in tal caso, l'entità del file codificato senza apprezzabile perdite della resa acustica.

La decodifica, al fine della riproduzione sonora, consiste nel separare le informazioni musicali da quelle di servizio e nella ricostruzione del segnale originale. Per far ciò occorre un opportuno software decodificatore e riproduttore di suoni. Uno dei più famosi a livello mondiale è WinAmp.

Occorre fare, a questo punto, una precisazione. Se il file MP3 viene ascoltato attraverso le casse acustiche del PC o attraverso un impianto di amplificazione di qualità non elevata, la differenza tra il file originale e quello MP3 non è percepibile. Se, invece, si utilizza un impianto HI-FI di buona qualità e ci si mette nelle condizioni di ricreare effetti realistici, si riscontra facilmente la differenza tra il suono originale e quello compresso. Per ovviare, almeno in parte, a questo inconveniente si possono utilizzare gli equalizzatori grafici (software o hardware) per adattare il suono al gusto dell'ascoltatore.

Campionando il segnale a frequenza più elevata, ad esempio a 44.100Hz, il fattore di compressione da 10:1 può passare a circa 4:1. Il file prodotto sarà, pertanto, meno compresso ma avrà una qualità prossima a quella del file originale.

### 6.3. Lo standard MIDI

La sigla MIDI (Musical Instrument Digital Interface) è riferita ad un protocollo di comunicazione studiato per consentire a due o più strumenti musicali di dialogare tra loro. Il protocollo MIDI *non trasmette suoni* ma *istruzioni* che pilotano un sintetizzatore.

Su tutte le schede audio, la porta Joystick è anche l'interfaccia MIDI che consente di collegare al PC strumenti musicali.

In assenza di strumenti musicali si può utilizzare il sintetizzatore FM della propria scheda audio che consente di simulare un numero abbastanza elevato di tali strumenti.

Affinché il file MIDI possa essere riprodotto correttamente su qualsiasi PC multimediale è necessario che vi sia una precisa corrispondenza tra numerazione e strumenti musicali.

Lo standard General MIDI, o GM, definisce 128 strumenti secondo un ordine prestabilito. I file prodotti hanno il suffisso MID, sono costituiti da suoni perfetti con riproduzione simultanea di uno o più di essi. Poiché, come si è detto, il file è costituito da istruzioni piuttosto che da campioni di suoni, la sua dimensione è estremamente contenuta: 40-50Kbyte per brani della durata di diversi minuti primi.

Lo svantaggio principale consiste nel fatto che non è possibile inserire la voce.

Per meglio comprendere la differenza tra file WAV e file MID è sufficiente fare riferimento al confronto, a proposito delle immagini, al formato bitmap BMP rispetto al formato vettoriale.

Utilizzando programmi di elaborazione del suono è possibile registrare un file MID, magari miscelato con l'audio che proviene da un microfono o da un CD-ROM, in un file in formato WAV o MP3, più compresso.

#### 6.4. Software per l'audio

Il sistema operativo Windows mette a disposizione un certo numero di programmi per la produzione e riproduzione del suono. Per l'elaborazione del suono si dovrà ricorrere ad opportuni programmi software disponibili a basso costo o di tipo professionale.

Citiamo subito il programma *Registratore di suoni* attivabile con: Avvio/Programmi/Accessori/Svago/Registratore di suoni (fig.32).



Fig.32. - Schermata del programma *Registratore di suoni*.

La barra del menù presenta 4 voci: File, Modifica, Effetti, ?

Il menu *File* consente le solite scelte: nuovo, apri, salva, salva con nome, proprietà, esci. I file trattabili sono solo quelli in formato WAV. Per modificare gli attributi del file è sufficiente attivare File/Proprietà e specificare il formato desiderato.

Il menù *Modifica* consente di copiare, incollare più qualche altra funzione di scarso utilizzo.

Il menù *Effetti* consente di aumentare o diminuire il volume, la velocità, di inserire l'eco e di riprodurre al contrario.

Il menù ? presenta una guida in linea.

I cinque pulsanti che scorgiamo in basso in fig.32 consentono di andare all'inizio del brano, alla fine, di avviare la riproduzione, lo stop ed infine la registrazione. Cliccando sul pulsante di *registrazione* (quello a destra) si dà avvio a tale funzione che avviene aggiungendo in coda i valori campionati finché si decide di interrompere la registrazione, cosa che avviene cliccando sul pulsante *Ferma*.

La registrazione è un'operazione semplice ma preventivamente dobbiamo controllare se sono stati selezionati i canali di input desiderati e dobbiamo impostare i livelli di registrazione. Quest'ultima operazione comporta l'esecuzione di un certo numero di prove preliminari prima che la vera registrazione possa ritenersi soddisfacente.

Per far ciò si deve attivare il programma *Controllo volume* disponibile negli accessori di Windows.

In fig.33 si vede come è possibile selezionare il volume in fase di riproduzione delle diverse sorgenti di audio.

In realtà interessa effettuare il controllo della registrazione e non della riproduzione per cui dal menù *Opzioni* si sceglie la voce *Proprietà* e si seleziona *Regola il volume di registrazione*.

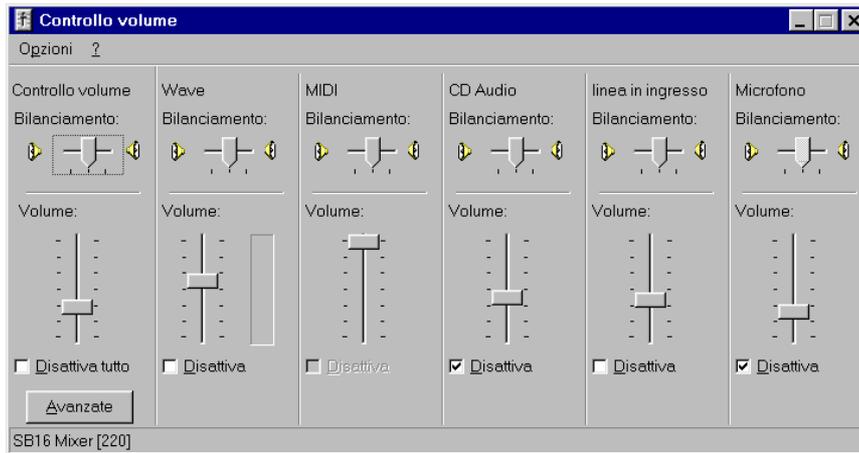


Fig.33. - Controllo volume.

I canali di ingresso per la registrazione sono il microfono, la linea di ingresso, il lettore di CD, i file MIDI ed altri eventuali dispositivi audio installati nel proprio PC.

Tali canali possono essere selezionati o deselezionati. Per tale motivo questo programma prende il nome di *mixer audio*.

Per la sola riproduzione di file di altro formato il Windows mette a disposizione un altro programma che si chiama *Lettore multimediale (Windows Media Player)*.

Esso è in grado di riprodurre file WAV, MIDI, AVI, MP3 (se si installa un driver apposito), CD audio; in poche parole i tipi più importanti di file audio ed i filmati.

I programmi di elaborazione audio presentano funzioni più o meno sofisticate di elaborazione audio e l'inserzione di effetti speciali come: effetto doppler, eco, filtraggi, modifica volume e velocità, fader, ecc.

Generalmente mostrano la forma d'onda del file in esame, mono o stereo, una pulsantiera per la riproduzione, la registrazione, il riavvolgimento veloce ed il mixer del tipo in dotazione al Windows.

## 6.5. Caratteristiche di una moderna scheda audio

Le moderne schede audio consentono la riproduzione di effetti sonori particolari come, ad esempio, il *surround sound*, il *suono 3D*, ecc.

Il cuore della scheda audio è un unico processore che contiene sia il convertitore AD, per l'acquisizione dell'audio, che il convertitore DA per la sua riproduzione. Presenta al suo interno, altresì, tutta la circuitistica per la gestione dei file midi.

A titolo d'esempio, la recente scheda audio *Sound Blaster Live!* utilizza il processore EMU10K1 sviluppato da E-mu Systems (azienda acquisita da Creative), contenente al suo interno 2 milioni di transistor e capace di 1000 MIPS (Milioni di istruzioni al secondo).

È un potente sintetizzatore musicale in grado di gestire numerosi effetti come *riverbero*, *chorus*, *flanger*, *pitch shifter*, ecc. applicabili in tempo reale sulla sorgente sonora.

Dispone di 64 voci in hardware, che unite alle 192 messe a disposizione dalla sintesi *wavetable*, forniscono un totale di 256 voci di sintesi musicale.

Sono presenti 48 canali MIDI con 128 strumenti compatibili GM - GS e 10 set di percussioni. Sono disponibili configurazioni con banchi di memoria da 2, 4 o 8 MByte; inoltre è possibile usare fino a 32 MByte della Ram di sistema per immagazzinare i propri campioni audio tramite la tecnologia *SoundFont*.

Il campionamento può avvenire a 8 o 16 bit con frequenze di 8, 11.025, 16, 22.050, 24, 32, 44.100 o 48 KHz, utilizzando un algoritmo di interpolazione a 8 punti. Il segnale viene processato con una profondità di 32bit per poi essere reso a 8 o 16 bit tramite *dithering*.

Una delle caratteristiche esclusive di questo processore è la tecnologia "E-mu Environmental Modeling", che, oltre a consentire l'immersione totale nell'ambiente, permette ai suoni di interagire tra di loro secondo un rapporto causa-effetto, consentendo la riflessione e il riverbero dei suoni.

È possibile, ad esempio, simulare una grande sala o un teatro con buon realismo; chiudendo gli occhi, si può determinare con buona approssimazione la direzione di provenienza di un suono. Per ottenere questo è necessario disporre di almeno 4 diffusori acustici; la *Sound Blaster Live* può gestirne fino a 8.

## 7. Interfacce non standard

Le interfacce non standard sono quelle non riconosciute a livello internazionale e di cui, pertanto, devono essere note le caratteristiche meccaniche, elettriche e funzionali affinché si possa effettuare il collegamento tra una data apparecchiatura ed il computer.

Nei successivi sottoparagrafi si descrivono alcune interfacce non standard di facile realizzazione per collegare al computer circuiti digitali, circuiti analogici. Tali interfacce non standard possono essere interne o esterne al PC e possono essere seriali o parallele.

Per la loro gestione si utilizzano le istruzioni di input ed output dei linguaggi di programmazione più noti, come sarà descritto nel paragrafo 7.6. Nel seguito si farà riferimento al linguaggio Q-Basic e Visual Basic.

### 7.1 Interfaccia parallela interna per l'I/O digitale

Si mostra in fig.34 lo schema funzionale di una interfaccia parallela da realizzare su scheda da inserire in uno slot ISA disponibile di un personal computer.

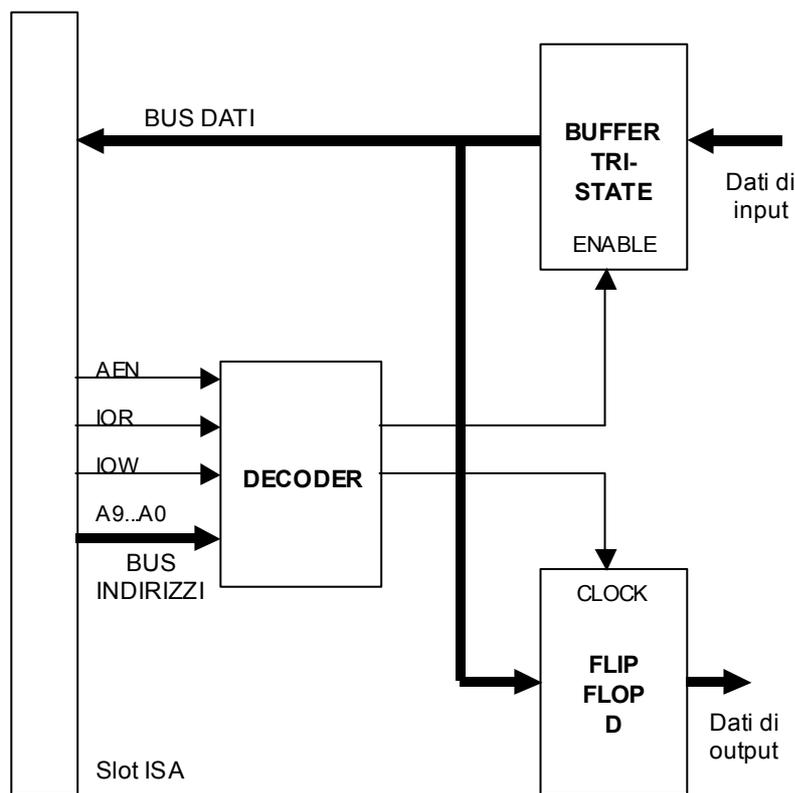


Fig.34. – Schema funzionale dell'interfaccia parallela proposta.

L'interfaccia deve poter risolvere le seguenti questioni:

- ❑ sdoppiare il BUS dati a 8 bit bidirezionali in 8 bit di ingresso ed 8 bit di uscita;
- ❑ attribuire alla scheda un indirizzo specifico, attraverso una logica di decodifica.

Il primo requisito è soddisfatto utilizzando 8 buffer tri-state contenuti nel chip 74LS244 per la gestione degli 8 bit di ingresso e gli 8 flip-flop D contenuti nel 74LS374 per la gestione degli 8 bit di uscita.

Il secondo requisito è soddisfatto grazie all'impiego di un comparatore digitale di uguaglianza per parole a 8 bit contenuto nel chip 74LS688: quando gli 8 bit utilizzati del bus indirizzi coincidono con quelli impostati dall'utente, una volta per tutte tramite microswitch, la scheda è abilitata al funzionamento ed è, pertanto, possibile svolgere operazioni di ingresso oppure operazioni di uscita.

In fig.35 si mostra il circuito dettagliato.

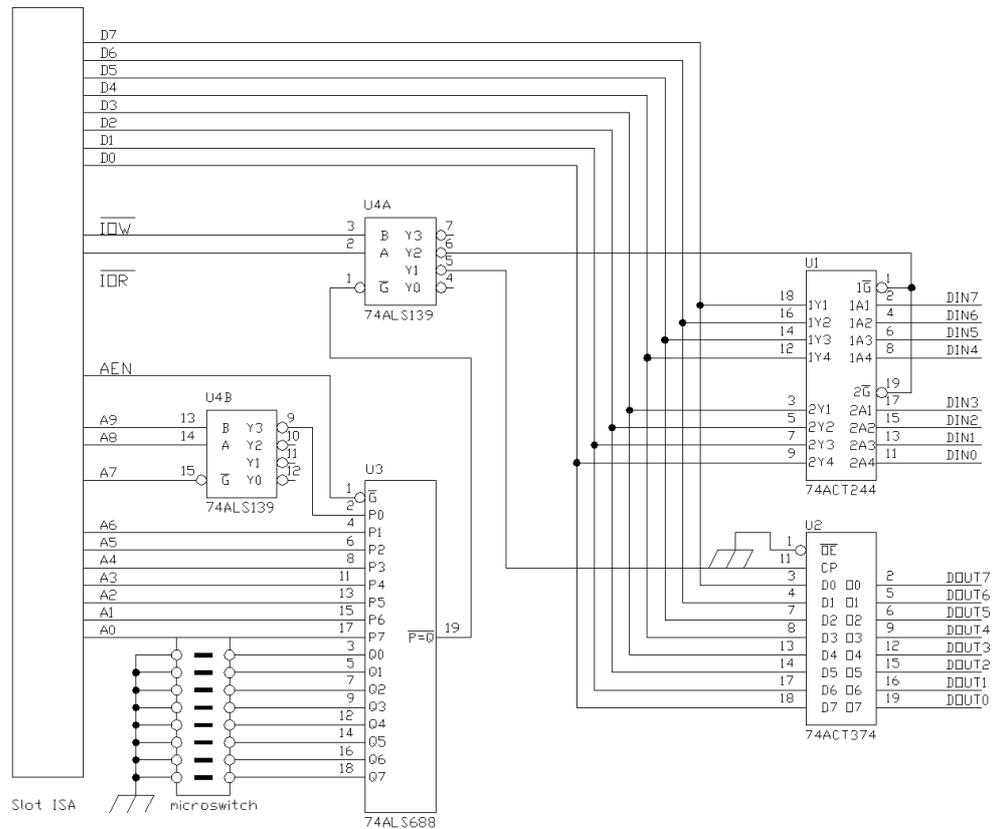


Fig.35. – Schema elettrico dell'interfaccia parallela.

I bit D7..D0 rappresentano il BUS dati bidirezionale attestato sullo slot ISA. Tali bit sono collegati agli ingressi del 74ACT374 che contiene 8 flip-flop D ed alle uscite del 74ACT244 che contiene 8 buffer tri-state.

Il bit AEN (address enable) è collegato all'ingresso di abilitazione del comparatore di uguaglianza per cui quest'ultimo si abilita durante la fase di indirizzamento di una istruzione.

L'integrato 74ALS139 contiene due decodificatori indipendenti a 2 bit con ingresso di abilitazione. La porzione U4B fornisce sull'uscita Y3 un livello basso se i bit del BUS indirizzi A9 ed A8 sono uguali a 1 e l'ingresso di abilitazione è attivato: A7=0. Tale linea Y3 pilota l'ingresso P0 del comparatore.

Il comparatore confronta il byte P, collegato al BUS indirizzi, col byte Q generato agendo sugli 8 microswitch. Se, ad esempio, tutti i microswitch sono collegati a massa, l'uscita del comparatore  $\overline{P=Q}$  va al livello basso se i 10 bit meno significativi del BUS indirizzi assume il valore esadecimale 300H.

Il livello logico basso all'uscita del comparatore abilita il decodificatore U4A i cui ingressi sono:

$$B = \overline{IOW} \text{ e } A = \overline{IOR}$$

Durante una istruzione di output su periferica di indirizzo 300H l'ingresso B è forzato al livello basso e l'ingresso A rimane al livello alto, cioè si ottiene: BA=01 e quindi l'uscita Y1 si attiva portandosi al livello logico basso (uscite del decoder in logica negativa). L'uscita Y1 pilota l'ingresso CP (Clock Pulse) del 374. Al termine dell'istruzione corrente Y1 ritorna al livello alto e questa transizione positiva consente la memorizzazione nei flip-flop del byte presente sul BUS dati.

Durante una istruzione di input su periferica di indirizzo 300H, invece, si ottiene BA=10 e quindi sarà Y2 a portarsi al livello logico basso. In tale circostanza si abilitano gli 8 buffer tri-state contenuti nel 244.

È consigliabile inserire delle resistenze di pull-up da 10KΩ tra ciascuna linea del microswitch ed i relativi ingressi Q del comparatore. Tale accorgimento è indispensabile se si desidera attribuire alla scheda un indirizzo diverso da 300H per non lasciare flottanti gli ingressi Q non collegati a massa considerando che uno o più ingressi devono essere tenuti al livello logico alto.

Si osservi, infine, che il decodificatore U4B impone che la configurazione dei bit A9A8A7 del BUS indirizzi sia 110 considerato che si è utilizzata l'uscita Y3 di U4B. In tal caso il microswitch Q0 dovrà assumere lo stesso valore di P0 e cioè 0. Imponendo: Q0=1 la scheda sarà disabilitata in tutti i casi.

Gli indirizzi possibili sono, pertanto, quelli che vanno da 300H a 37FH. Ovviamente si eviterà l'impostazione di indirizzi coincidenti con quelli della LTP1 (378H, 379H, 37AH) e della COM1 (3F8H...3FFH).

I 4 circuiti integrati utilizzati devono essere alimentati con una tensione costante di +5V prelevabile dal contatto B3 oppure dal contatto B29 del connettore ISA (vedi tabella 1 del paragrafo 2.1). La massa dello slot ISA è disponibile ai contatti B1, B10 e B31.

## 7.2. Interfaccia parallela esterna per l'I/O digitale

L'interfaccia interna esaminata nella precedente fig.35 presenta l'inconveniente di dover aprire il computer per il suo inserimento nello slot ISA, operazione abbastanza semplice ma che non trova il consenso di molti.

Una possibile soluzione per interfacciare propri circuiti digitali con 8 bit di ingresso ed 8 bit di uscita al personal computer è quella di utilizzare l'interfaccia standard Centronics già disponibile sul retro del PC e descritta nel paragrafo 3.1.

Poiché l'interfaccia Centronics fornisce un byte di uscita ma accetta solo 5 bit in ingresso si ricorre, per l'acquisizione di un byte, alla tecnica del multiplex facendo acquisire al PC prima 4 bit e successivamente gli altri 4 bit del byte. L'acquisizione di un byte, pertanto, avverrà attraverso due fasi successive di acquisizione a 4 bit. Il software di gestione provvederà alla opportuna fusione dei due semibyte.

In fig.36 si propone uno schema elettrico molto semplice che impiega solamente un 74LS244 che, come è noto, contiene due gruppi indipendenti di buffer tri-state.

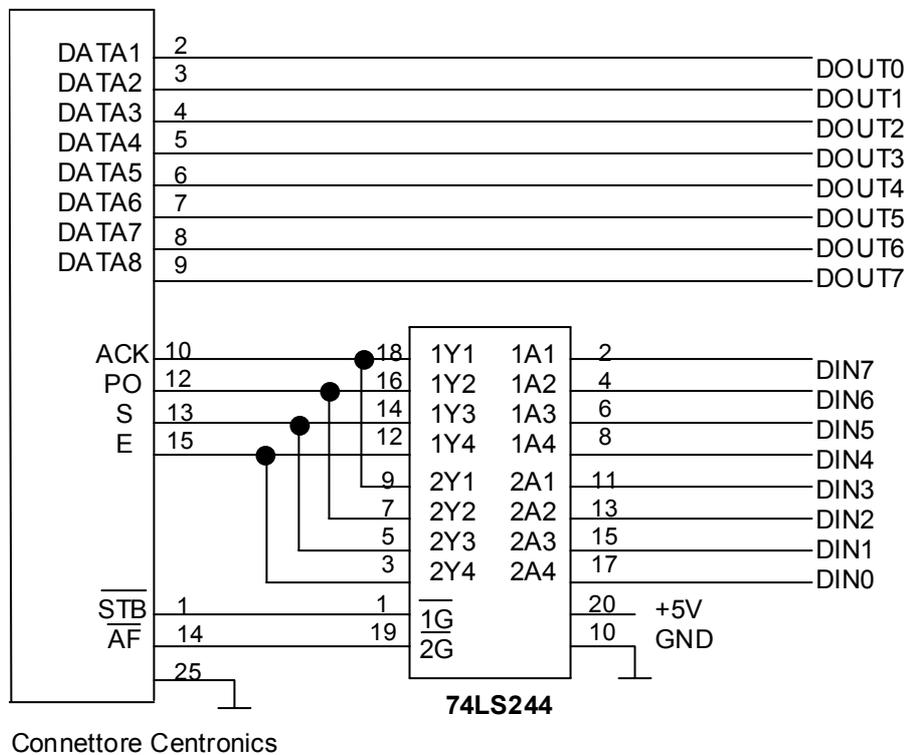


Fig.36. - Interfaccia parallela esterna per l'I/O digitale.

Supponendo di utilizzare il linguaggio QBasic, l'algoritmo di acquisizione del byte applicato alle linee DIN7...DIN0 è il seguente:

<b>OUT 890, 1</b>	Generazione del livello basso sul bit $\overline{STB}$ ed alto sul bit $\overline{AF}$ per cui i bit 1Y1 1Y2 1Y3 1Y4 sono abilitati mentre i bit 2Y1 2Y2 2Y3 2Y4 sono in tri-state.
<b>AH = (INP(889) AND 120) * 2</b>	Acquisizione e sistemazione dei 4 bit di ingresso più significativi nella parte alta del byte A.
<b>OUT 890, 2</b>	Generazione del livello alto sul bit $\overline{STB}$ e basso sul bit $\overline{AF}$ per cui i bit 1Y1 1Y2 1Y3 1Y4 sono in tri-state mentre i bit 2Y1 2Y2 2Y3 2Y4 sono abilitati.
<b>AL = (INP(889) AND 120) / 8</b>	Acquisizione e sistemazione dei 4 bit di ingresso meno significativi nella parte bassa del byte A.
<b>A = AH + AL</b>	Composizione del byte A.
<b>OUT 890, 0</b>	Generazione del livello alto sia su $\overline{STB}$ che su $\overline{AF}$ per cui sono in tri-state i bit 1Y1 1Y2 1Y3 1Y4 ed i bit 2Y1 2Y2 2Y3 2Y4.

L'AND con 120 (01111000) consente di estrarre solo i 4 bit utilizzati dall'interfaccia forzando gli altri a 0. La moltiplicazione per 2 produce lo scorrimento a sinistra di un bit mentre la divisione per 8 produce lo scorrimento a destra di 3 bit. Pertanto si ha:

<b>AH=</b>	DIN7	DIN6	DIN5	DIN4	0	0	0	0
<b>AL=</b>	0	0	0	0	DIN3	DIN2	DIN1	DIN0
<b>A=AH+AL</b>	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0

L'algoritmo per la generazione del byte B di uscita sulle linee DOUT7...DOUT0 è estremamente semplice:

#### **OUT 888, B**

Per poter funzionare correttamente, il 74LS244 deve essere alimentato con una tensione costante di +5V prelevata da un alimentatore esterno.

### **7.3. Interfaccia parallela esterna per l'I/O digitale n. 2**

E' possibile utilizzare il registro dati posto all'indirizzo 888 dell'interfaccia Centronics come porta bidirezionale per generare un byte di uscita o per acquisire un byte di ingresso.

Come già illustrato nel paragrafo 3.1, per rendere bidirezionale il registro dati dell'interfaccia Centronics è necessario che quest'ultima sia impostata preliminarmente come EPP dal setup del BIOS del PC. Per farla funzionare come registro di ingresso si dovrà impostare ad 1 il bit 5 del registro di controllo posto all'indirizzo 890. In QBasic l'istruzione è: OUT 890, 32.

Per rendere il registro dati nuovamente di uscita è sufficiente riportare a 0 il bit 5 del registro di controllo. In Qbasic l'istruzione è: OUT 890, 0.

Per sdoppiare le linee di ingresso da quelle di uscita si propone l'utilizzo del circuito integrato 74LS374 per memorizzare il byte di uscita e l'integrato 74LS244 per bufferizzare il byte di ingresso secondo lo schema di fig.37.

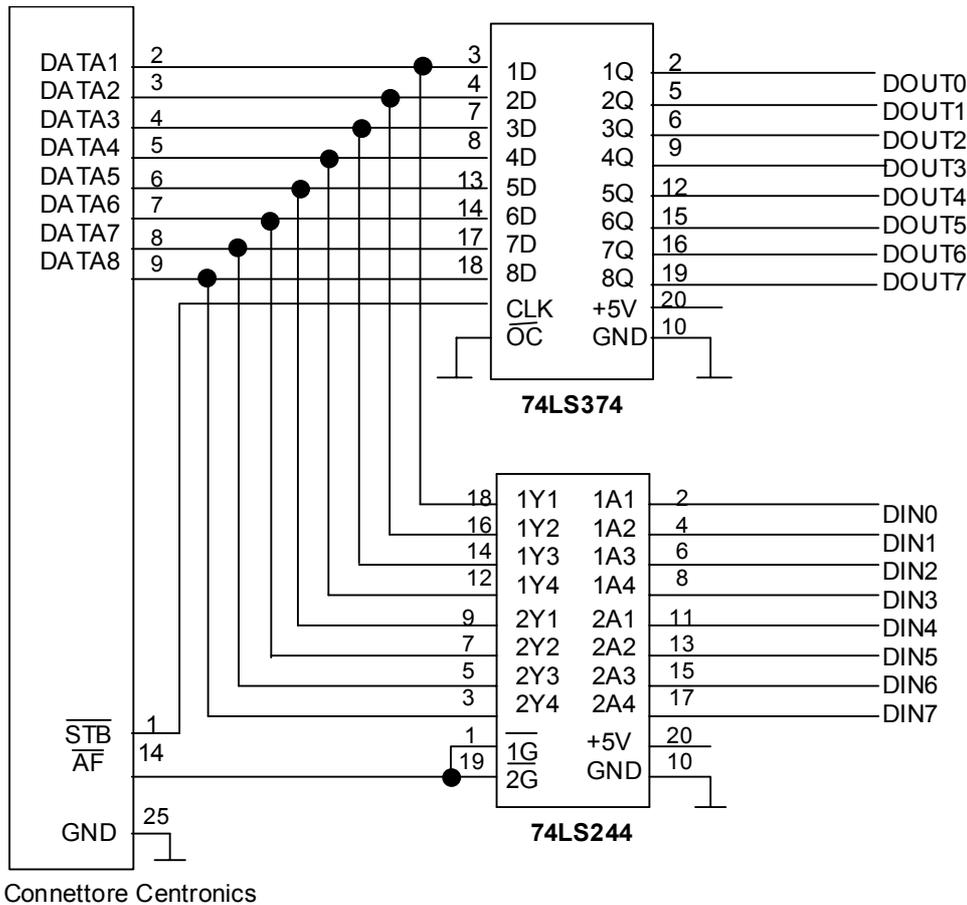


Fig.37. – Interfaccia parallela per l'I/O digitale (n.2).

Si riporta, in linguaggio QBasic, l'algoritmo di acquisizione del byte DIN7..DIN0 nella locazione di memoria A:

- OUT 890, 32** Impostazione del registro dati come registro di ingresso
- OUT 890, 34** Abilitazione del 244
- A=INP(888)** Acquisizione del byte DIN7...DIN0 nella locazione A
- OUT 890,0** Ripristino registro dati come registro di uscita

Si riporta, infine, l'algoritmo di emissione del byte B:

- OUT 890,0** Impostazione del registro dati come registro di uscita
- OUT 888, B** Emissione del byte B attraverso il registro dati
- OUT 890, 1** Clock al livello basso
- OUT 890, 0** Clock al livello alto

### 7.4. Interfaccia parallela esterna per l'I/O analogico e digitale

Se colleghiamo un DAC a 8 bit alle uscite DOUT0...DOUT7 del 74LS374 della precedente figura 37 si ottiene una tensione di uscita analogica proporzionale al numero emesso dal PC.

Se, inoltre, colleghiamo le uscite D0...D7 di un ADC a 8 bit agli ingressi DIN0...DIN7 del 74LS244 della precedente figura 37, è possibile far acquisire al PC un valore numerico proporzionale alla tensione analogica applicata all'ingresso dell'ADC.

In fig.38 si mostra la porzione di circuito interessata alla conversione DA e AD.

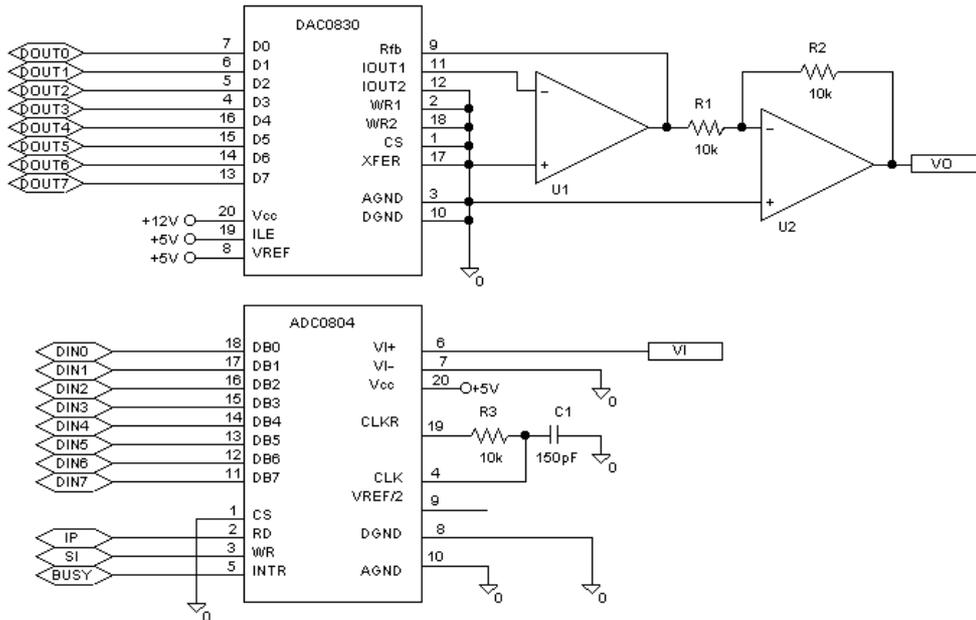


Fig.38. – Circuito da collegare allo schema elettrico di fig.37 per realizzare l'interfaccia al PC sia digitale che analogica.

Il convertitore digitale analogico utilizzato è il DAC0830 che presenta uscita in corrente. L'operazionale U1 si comporta da convertitore corrente-tensione e l'operazionale U2 è in configurazione invertente a guadagno unitario. In tal caso la tensione di uscita  $V_o$  che si ottiene è compresa tra 0 e 5V secondo la formula:

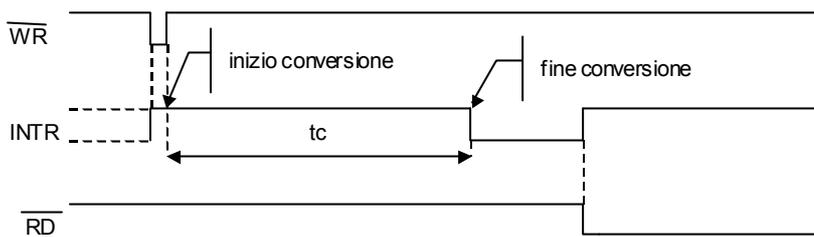
$$V_o = \frac{5}{256} \cdot N$$

ove  $5V$  è la tensione di fondo scala coincidente con  $V_{REF}$  applicato al piedino 8,  $256 = 2^8$  ove 8 è il numero di bit di ingresso del DAC ed N rappresenta il codice numerico applicato.

Per la conversione analogico-digitale si utilizza l'ADC0804 ad approssimazioni successive ad 8 bit alimentato a  $5V$ . Con i valori di  $R3$  e  $C1$  inseriti il tempo di conversione è di circa  $100\mu s$ . Lasciando flottante l'ingresso  $V_{REF}/2$  al piedino 9, quest'ultimo si polarizza automaticamente al valore  $2.5V$ .

Il funzionamento è il seguente:

Poiché la linea  $CS$  è collegata a massa, il convertitore risulta sempre abilitato al funzionamento.



**Fig.39. - Tempificazioni per l'ADC0804: Start conversion (impulso  $\overline{WR}$ ), stato di occupato per conversione in corso (livello alto di  $INTR$ ), lettura dei bit di uscita (livello basso di  $\overline{RD}$ ).**

Per attivare un ciclo di conversione si deve applicare un livello logico basso seguito immediatamente da un livello logico alto alla linea  $WR$  posta al piedino 3 come si mostra in fig.39. A ciò provvede la linea  $SI$  (Select Input), attiva bassa, del registro della Centronics sito all'indirizzo 890. Qualunque era il livello logico di  $INTR$ , il passaggio al livello basso di  $WR$  porta la linea  $INTR$  al livello logico alto.

Durante la conversione  $INTR$  è al livello logico alto e va automaticamente al livello logico basso al termine della conversione. Collegando tale linea a  $BUSY$ , attiva bassa, del registro di ingresso della Centronics posto all'indirizzo 889, è possibile consentire l'acquisizione al computer del dato digitale convertito.

Prima di effettuare la lettura si devono abilitare le linee di uscita dell'ADC applicando un livello logico basso alla linea  $RD$  posta al piedino 2. A ciò provvede la linea  $IP$  (Initialize Printer), attiva bassa, del registro della Centronics sito all'indirizzo 890. Quando si attiva  $RD$ , la linea  $INTR$  ritorna al livello alto.

In tabella 12 si mostrano i bit del registro dell'interfaccia Centronics sita all'indirizzo 890, la piedinatura rispetto al connettore a 25 poli, i valori dei bit, lo stato per le operazioni di "start conversion", di abilitazione alla lettura e le relative istruzioni supponendo di utilizzare il linguaggio QBasic.

**Tabella 12 – istruzioni per l'acquisizione e l'emissione analogica e digitale col circuito di fig.38 collegato a quello di fig.37**

Indirizzo 890	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Istruzione Qbasic
LINEE				IRQE	— SI	IP	— AF	— STB	
PIN					17	16	14	1	
Funzione svolta			888 IN (H) OUT (L)		— WR ADC	— RD ADC	— G 244	CLK 374	
Valore bit	128	64	32	16	8	4	2	1	
Avvio conversione			H		L	H	H	H	OUT 890, 44
			H		H	H	H	H	OUT 890, 36
Abilitazione lettura ADC			H		H	L	H	H	OUT 890, 32
Abilitazione 244			H		H	H	L	H	OUT 890, 38
Acquisizione al PC di un dato dall'ADC			H		H	L	L	H	OUT 890,34
CLOCK del 374			L		H	H	H	L	OUT 890, 5
			L		H	H	H	H	OUT 890, 4

Prima di acquisire il dato digitale, il PC deve saggiare la linea INTR per controllare se la conversione è terminata. Quando ciò avviene la linea INTR va al livello basso. Poiché la linea INTR è collegata alla linea BUSY, la variabile QBasic EOC, definita dalla successiva istruzione, vale 0 se la conversione è ancora in corso e vale 1 al termine della conversione:  $EOC = (INP(889) \text{ AND } 128) / 128$ .

Per l'acquisizione del dato digitale dell'ADC sono necessarie le seguenti due istruzioni: OUT 890, 34 e  $DATO\_ACQUISITO = INP(888)$ .

La prima abilita sia le uscite dell'ADC che quelle del 244. La seconda trasferisce, nella variabile  $DATO\_ACQUISITO$  del programma che gestisce l'acquisizione, il valore numerico applicato sulle linee DATA8...DATA1 dell'interfaccia Centronics di indirizzo 888.

Si riporta, in linguaggio QBasic, l'intero l'algoritmo di acquisizione del byte DB7..DB0 delle linee di uscita dell'ADC0804 nella locazione di memoria  $DATO\_ACQUISITO$ :

<b>OUT 890, 44</b>	
<b>OUT 890, 36</b>	Start conversion
<b>DO</b>	
<b>EOC = (INP(889)AND128)/128</b>	Controllo ripetuto finché la conversione ha termine (EOC=1)
<b>LOOP UNTIL EOC=1</b>	
<b>OUT 890, 34</b>	Abilitazione lettura ADC e 244
<b>DATO_ACQUISITO=INP(888)</b>	Acquisizione del byte
<b>OUT 890, 4</b>	Disabilitazione dell'ADC, del 244 e ripristino porta 888 come porta di uscita

## 7.5. Istruzioni di input ed output con Visual Basic

Il linguaggio di programmazione Visual Basic consente di mettere a punto interfacce grafiche estremamente elaborate ed accattivanti con poca fatica ma purtroppo, in forma nativa, non supporta le istruzioni di input e di output direttamente da unità periferiche.

Per ovviare a ciò è possibile utilizzare un particolare file di libreria DLL, come ad esempio INPOUT32.DLL, funzionante su computer con Windows 95, 98, ME ma non con Windows 2000. Tale DLL va inserita nella cartella C:\Windows del proprio hard-disk.

File di questo genere, implementabili in linguaggio C o Pascal, sono numerosi e disponibili su Internet con le istruzioni d'uso. Si forniscono due indirizzi Web:

[www.lvr.com/files/inpout32.zip](http://www.lvr.com/files/inpout32.zip) e [www.driverlinx.com/](http://www.driverlinx.com/).

In questo modo è possibile utilizzare in Visual Basic 4 e versioni successive le istruzioni OUT e INP che abbiamo utilizzato in QBasic con lo stesso tipo di sintassi.

Affinché si possa invocare, nel nostro applicativo, il file di libreria INPOUT32.DLL, si deve includere nel progetto Visual Basic da noi generato il modulo INPOUT32.BAS, presente anch'esso nel file ZIP, di seguito riportato:

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

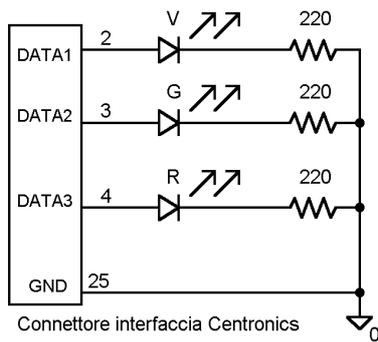
Tale inclusione si può realizzare attivando il comando *Inserisci modulo / Esistente* dal menù *Progetto* di Visual Basic.

### Esempio n.1

Si vuole comandare un semplice semaforo, simulato da 3 diodi LED con software in Visual Basic. I 3 diodi LED sono, ovviamente, di colore verde, giallo e rosso e collegati, tramite resistenza da 220Ω, rispettivamente alle uscite DATA1, DATA2 e DATA3 del registro dati di indirizzo 888 dell'interfaccia Centronics.

L'interfaccia grafica deve mostrare il disegno del semaforo che, ad intervalli regolari di tempo, aggiorna la lampada che deve essere accesa. Sul monitor devono, inoltre, essere disponibili 3 caselle di testo in cui inserire, in secondi, la durata del verde, del giallo e del rosso.

### Risoluzione

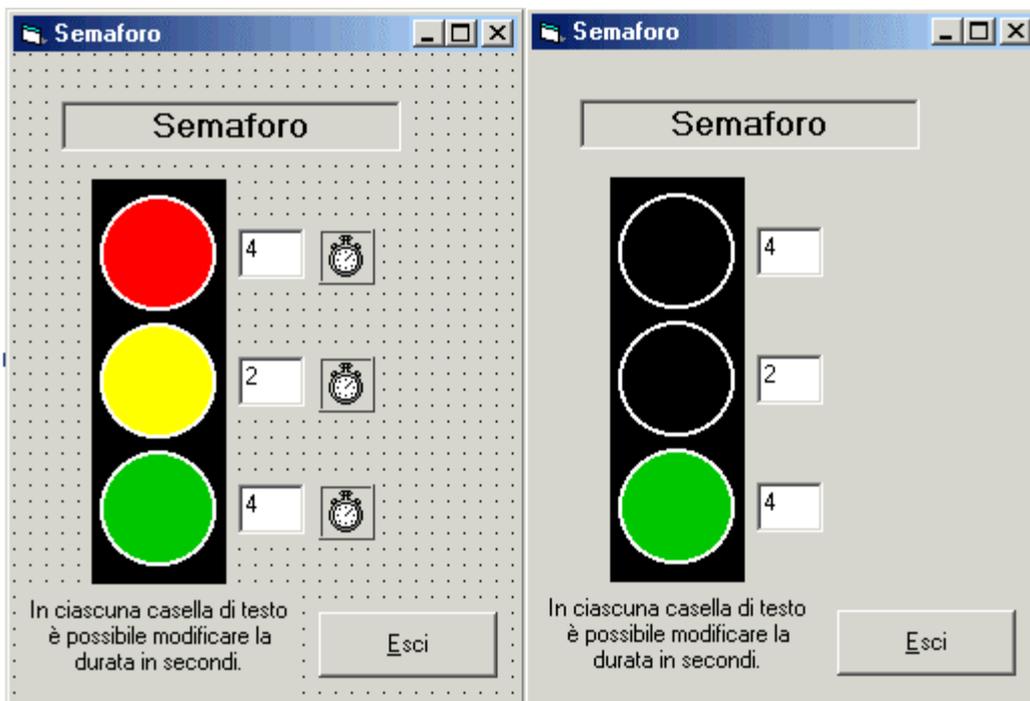


Si mostra in fig.40 il circuito da collegare direttamente all'uscita del connettore dell'interfaccia Centronics.

L'istruzione per l'attivazione del verde è: OUT 888, 1  
L'istruzione per l'attivazione del giallo è: OUT 888, 2  
L'istruzione per l'attivazione del rosso è: OUT 888, 4

**Fig.40.** – Diodi LED verde, giallo e rosso collegati alle linee di uscita del registro dati, indirizzo 888, dell'interfaccia Centronics.

In fig.41 si mostra, invece, l'interfaccia grafica realizzata e la schermata ottenuta durante l'esecuzione del programma.



**Fig.41.** – A sinistra: interfaccia grafica realizzata in ambiente Visual basic. A destra: schermata durante l'esecuzione del programma.

Di seguito si riportano i listati delle sub del programma realizzato in Visual Basic.

```
Private Sub Form_Load()
Shape_verde.BackStyle = 1
Shape_giallo.BackStyle = 0
Shape_rosso.BackStyle = 0
Timer_verde.Interval = Val(Text_verde) * 1000
Timer_verde.Enabled = True
Out 888, 1
End Sub

Private Sub Timer_verde_Timer()
Timer_verde.Enabled = False
Shape_verde.BackStyle = 0
Shape_giallo.BackStyle = 1
Timer_giallo.Interval = Val(Text_giallo) * 1000
Timer_giallo.Enabled = True
Out 888, 2
End Sub

Private Sub Timer_giallo_Timer()
Timer_giallo.Enabled = False
Shape_giallo.BackStyle = 0
Shape_rosso.BackStyle = 1
Timer_rosso.Interval = Val(Text_rosso) * 1000
Timer_rosso.Enabled = True
Out 888, 4
End Sub

Private Sub Timer_rosso_Timer()
Timer_rosso.Enabled = False
Shape_rosso.BackStyle = 0
Shape_verde.BackStyle = 1
Timer_verde.Interval = Val(Text_rosso) * 1000
Timer_verde.Enabled = True
Out 888, 1
End Sub

Private Sub Command1_Click()
Out 888, 0
End
End Sub
```

Il programma consiste in 5 subroutine. La prima, **Form\_Load**, si attiva al caricamento del programma e consente di visualizzare acceso il cerchio verde tenendo spenti, invece, i cerchi giallo e rosso. Consente, inoltre, di fissare la durata del timer verde, in millisecondi, in base al contenuto della relativa casella di testo moltiplicato per 1000 e successivamente attiva il timer verde ed emette sulla Centronics il codice 1 che illumina il diodo verde collegato al pin2. Trascorso il tempo previsto, il timer verde richiama la subroutine **Sub Timer\_verde** che ha il compito di disabilitare il timer verde, di “spegnere” il

cerchio verde, di “accendere” quello giallo, di caricare nel timer giallo il valore inserito nella relativa casella di testo, di attivare lo stesso timer giallo e di emettere sulla Centronics il dato 2 che illumina solo il diodo giallo.

In sequenza temporale, quindi, si attivano le subroutine **Sub Timer\_Giallo** e **Sub Timer\_Rosso**. Quest’ultima, a sua volta, consente il caricamento della subroutine **Sub Timer\_Verde** ed il ciclo si ripete.

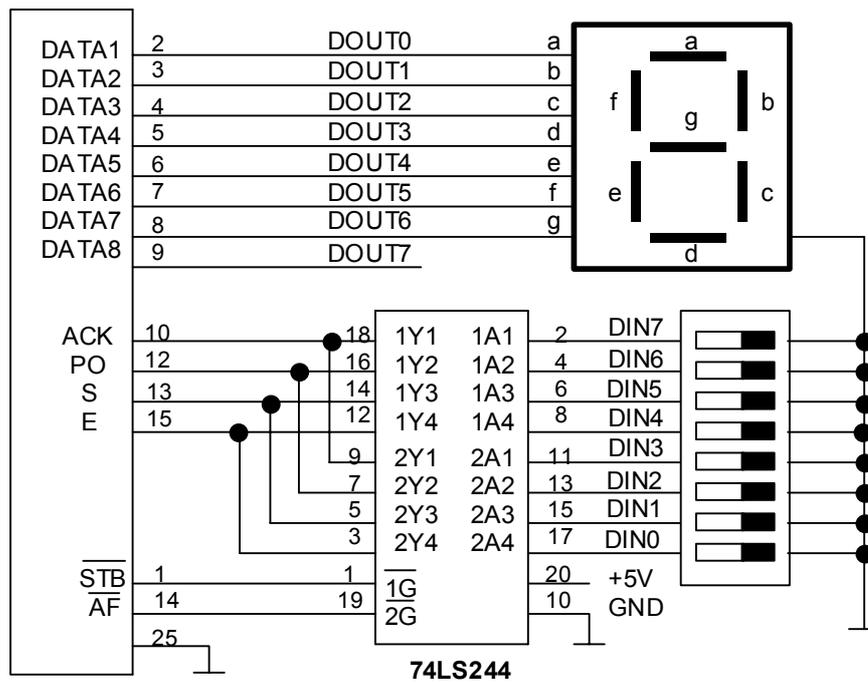
La subroutine **End** consente di interrompere il ciclo e di uscire dal programma dopo aver spento tutti i LED con l’istruzione OUT 888, 0.

**Esempio n.2**

Si vuole realizzare un circuito codificatore con priorità in grado di visualizzare, su un display a sette segmenti a catodo comune, il numero corrispondente all’ingresso attivato, tramite microswitch ad 8 microinterruttori, avente maggior priorità.

Il sistema codificatore è realizzato dal circuito di interfaccia parallela di fig.36, discusso nel precedente paragrafo 7.2, che per l’acquisizione di un byte utilizza il 74LS244 che contiene 8 buffer tri-state utilizzato come multiplexer che consente il passaggio alternato di due coppie di 4 bit. Il computer, oltre a svolgere la funzione di codificatore, si comporta anche da decodificatore per il pilotaggio del display.

In fig.42 si mostra il circuito da realizzare.



Connettore Centronics

**Fig. 42. – Circuito di interfaccia parallela che utilizza il registro di stato dell’interfaccia Centronics per l’acquisizione del dato digitale.**

Spiegare l'algoritmo risolutivo e produrre un programma in Visual Basic in grado di codificare il dato di ingresso con priorità e di decodificarlo per la visualizzazione sul display a sette segmenti a catodo comune.

### Risoluzione

Sia DIN7 l'ingresso a maggior priorità e DIN0 quello a priorità più bassa. Supponiamo, inoltre, di voler lavorare in logica positiva: un livello logico alto applicato ad una linea di ingresso viene riconosciuto come ingresso abilitato. Si consiglia, infine, di inserire 8 resistenze di pull-up, ciascuna da 10K $\Omega$ , tra l'alimentazione +5V e ciascun ingresso DIN7...DIN0.

Anche le linee di uscita lavorano in logica positiva poiché si è detto di utilizzare un display a catodo comune.

Si riportano in tab. 13 i possibili stati delle linee di ingresso, i codici numerici acquisiti e quelli di uscita che consentono il pilotaggio del display.

Tab.13

DIN7	DIN6	DIN5	DIN5	DIN3	DIN2	DIN1	DIN0	DIN	N1	g	f	e	d	c	b	a	N2
1	X	X	X	X	X	X	X	>127	7	0	0	0	0	1	1	1	7
0	1	X	X	X	X	X	X	>63	6	1	1	1	1	1	0	1	125
0	0	1	X	X	X	X	X	>31	5	1	1	0	1	1	0	1	109
0	0	0	1	X	X	X	X	>15	4	1	1	0	0	1	1	0	102
0	0	0	0	1	X	X	X	>7	3	1	0	0	1	1	1	1	79
0	0	0	0	0	1	X	X	>3	2	1	0	1	1	0	1	1	91
0	0	0	0	0	0	1	X	>1	1	0	0	0	0	1	1	0	3
0	0	0	0	0	0	0	1	>0	0	0	1	1	1	1	1	1	63
0	0	0	0	0	0	0	0	=0	8	0	0	0	0	0	0	0	0

Il software, dopo aver acquisito il dato DIN, deve determinare il numero N1 secondo il seguente ragionamento:

Se  $DIN > 127$  ( $2^7 - 1$ ) allora sicuramente  $DIN7 = 1$  e, indipendentemente dallo stato degli altri ingressi, si deve porre  $N1 = 7$  poiché  $DIN7$  è l'ingresso a maggior priorità. In caso negativo si deve controllare se  $DIN > 63$  ( $2^6 - 1$ ). In caso affermativo si ha:  $N1 = 6$  altrimenti si procede controllando  $DIN$  con 15, 7, ecc.

È sufficiente attivare un ciclo definendo una variabile contatore I da 7 a 0 ed imporre i controlli visti prima fermandosi quando questo è soddisfatto. In tal caso  $N1$  è uguale al valore corrente di I.

Se il controllo non è mai soddisfatto vuol dire che non è attivato alcun ingresso per cui  $DIN = 0$  e per  $N1$  attribuiremo convenzionalmente il valore 8 (display spento).

Il valore  $N2$  da fornire in uscita per il pilotaggio del display è uno dei 9 valori memorizzati in un vettore.

In fig.43 si fornisce l'interfaccia grafica realizzata in linguaggio Visual Basic.



**Fig.43. – Schermata del programma supponendo di aver attivato gli ingressi 5, 4, 3, 1. Quello con maggior priorità è l'ingresso 5. Dall'interfaccia Centronics si attiva il codice binario a 7 bit 109 che comanda il display a sette segmenti in modo da visualizzare il numero 5.**

Si mostra, di seguito, il listato del programma in Visual Basic che gestisce il codificatore. Il programma consente la visualizzazione del formato binario e dell'ingresso attivato con maggior priorità anche inserendo manualmente un valore nella casella di testo text3, etichettata DIN.

```

Dim n2(10) As Integer

Private Sub Form_Load()
n2(7) = 7
n2(6) = 125
n2(5) = 109
n2(4) = 102
n2(3) = 79
n2(2) = 91
n2(1) = 6
n2(0) = 63
n2(8) = 0
End Sub

Private Sub Text3_Change()
visualizzazione
End Sub

Private Sub CmdCodif_Click()
acquisizione
visualizzazione
End Sub

Private Sub CmdEsci_Click()
End
End Sub

Private Sub acquisizione()
Out 890, 1
ah = (Inp(889) And 120) * 2
Out 890, 2
al = (Inp(889) And 120) / 8
a = ah + al
Out 890, 0
Text3 = a
End Sub

Private Sub visualizzazione()
din = Val(Text3.Text)
If din > 255 Then
din = 0
Text3 = 0
End If
n1 = 8
For i = 0 To 7
If din > 2^i - 1 Then n1 = i
Next i
Text2 = ""
If Not n1 = 8 Then Text2 = n1
Out 888, n2(n1)
For i = 0 To 7
Text1(i) = din Mod 2
din = Int(din / 2)
Next i
End Sub

```

La routine **Form\_Load()** si attiva al caricamento del programma e consiste nell'assegnare al vettore n2 i relativi 9 valori come si è visto nella tabella 13.

La routine **Text3\_Change()** si attiva quando il contenuto della casella di testo denominata DIN cambia di valore. Il suo compito è quello di attivare la routine *visualizzazione* che consente di materializzare il numero decimale DIN in forma binaria e nella casella di testo a destra compare il numero di posizione del bit più significativo posto ad 1.

La routine **CmdCodif\_Click()** si attiva facendo click sul pulsante Codifica. Il suo compito è quello di eseguire l'*acquisizione* di un byte dagli otto microswitch e la visualizzazione su monitor e sul display.

La routine **CmdEsci\_Click()** si attiva facendo click sul pulsante Esci. Il suo compito è quello di interrompere l'esecuzione del programma.

La routine **acquisizione** cattura il byte prodotto tramite microswitch in due fasi da 4 bit. In ciascuna di tali fasi, nel registro di stato devono essere presenti solo i 4 bit acquisiti che occupano le posizioni 6, 5, 4 e 3 del registro. Per essere sicuri che gli altri bit del registro non influenzino l'acquisizione, si procede all'azzeramento del bit 7 e dei bit 2, 1 e 0. Questo azzeramento si ottiene eseguendo il prodotto logico, bit a bit, tra il contenuto del registro e il numero corrispondente alla configurazione binaria: 0 1 1 1 0 0 0 che corrisponde al decimale 120. Infatti:

Registro di stato	A7	A6	A5	A4	A3	A2	A1	A0
Byte di maschera	0	1	1	1	1	0	0	0
Risultato prodotto logico	0	A6	A5	A4	A3	0	0	0

La moltiplicazione per due consente lo scorrimento a sinistra di una posizione. Il risultato è posto nella variabile denominata AH. La divisione per  $8=2^3$  consente lo scorrimento a destra di tre posizioni. Il risultato è posto nella variabile denominata AL. La somma tra AH ed AL è posta nella variabile A che viene poi inviata nella casella di testo DIN.

AH	DIN7	DIN6	DIN5	DIN4	0	0	0	0
AL	0	0	0	0	DIN3	DIN2	DIN1	DIN0
A	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0

L'istruzione OUT 890, 1 porta al livello basso  $\overline{STB}$  che abilita i primi 4 buffer tri-state a cui sono collegate le linee DIN7 DIN6 DIN5 DIN4.

L'istruzione OUT 890, 2 porta al livello basso  $\overline{AF}$  che abilita gli ultimi 4 buffer tri-state a cui sono collegate le linee DIN3 DIN2 DIN1 DIN0.

La routine **visualizzazione** individua l'ingresso abilitato avente maggior priorità da visualizzare nella casella di testo text2, la più grande, e poi converte il numero acquisito in binario da visualizzare su monitor.

**Esempio n.3**

Prova scritta di Sistemi agli Esami di Stato 2001 per l'indirizzo "Elettronica e Telecomunicazioni" - Traccia 2.

Si vuole sviluppare un sistema di monitoraggio dello stato di occupazione in un parcheggio con la capacità di 512 autoveicoli suddivisi in otto settori. Ciascun settore del parcheggio ha un unico varco di ingresso e un unico varco di uscita. Ogni varco può essere attraversato da una sola macchina alla volta ed è dotato di un opportuno sensore che segnala il passaggio dell'auto.

Un sistema a microprocessore deve essere in grado di acquisire l'impegno del parcheggio sia in relazione al numero di auto in sosta che alla loro distribuzione nei vari settori. Un opportuno sistema di segnalazione all'ingresso del parcheggio informa gli automobilisti in arrivo indicando per ogni settore il numero di posti liberi. All'ingresso di ogni settore c'è una sbarra che si chiude automaticamente quando il settore è pieno. All'ingresso principale c'è una sbarra che si chiude quando i posti liberi scendono a 10.

Il candidato, dopo aver effettuato tutte le ipotesi aggiuntive ritenute necessarie:

1. individui gli attuatori e i sensori necessari al sistema di controllo;
2. descriva in modo motivato in quali punti il sistema necessita di blocchi di condizionamento del segnale;
3. disegni lo schema a blocchi del sistema di controllo evidenziando i problemi relativi all'interfacciamento con il sistema a  $\mu P$ , proponendo soluzioni adeguate;
4. descriva, con uno strumento di sua conoscenza, l'algoritmo dell'intera gestione impostandolo in modo modulare;
5. dettagli in maniera particolareggiata, in relazione alle soluzioni hardware proposte, l'algoritmo di conteggio delle macchine, settore per settore, e lo codifichi in un linguaggio di sua conoscenza.

**Soluzione**

Si ipotizza che tutti i settori abbiano la stessa capacità di ospitare autovetture. In tal caso la capienza di ogni settore è:  $512/8 = 64$  autovetture.

**Risposta al quesito 1.**

Gli attuatori sono i motori che comandano la chiusura o l'apertura della sbarra. Occorrono 9 motori: uno per l'ingresso principale ed uno per ciascuno degli 8 settori.

Si preferisce far cadere la scelta su motori in corrente continua per consentire un comodo sistema di inversione del moto per la chiusura o l'apertura della sbarra costituita da una struttura leggera. Due contatti di fine corsa, uno di apertura ed uno di chiusura, consentiranno l'arresto del moto quando la sbarra, aprendosi, raggiungerà la posizione verticale e quando, chiudendosi, raggiungerà la posizione orizzontale.

L'inversione della rotazione si può ottenere inserendo il motore in un circuito con due transistor PNP e due NPN in configurazione a ponte come in fig.44.

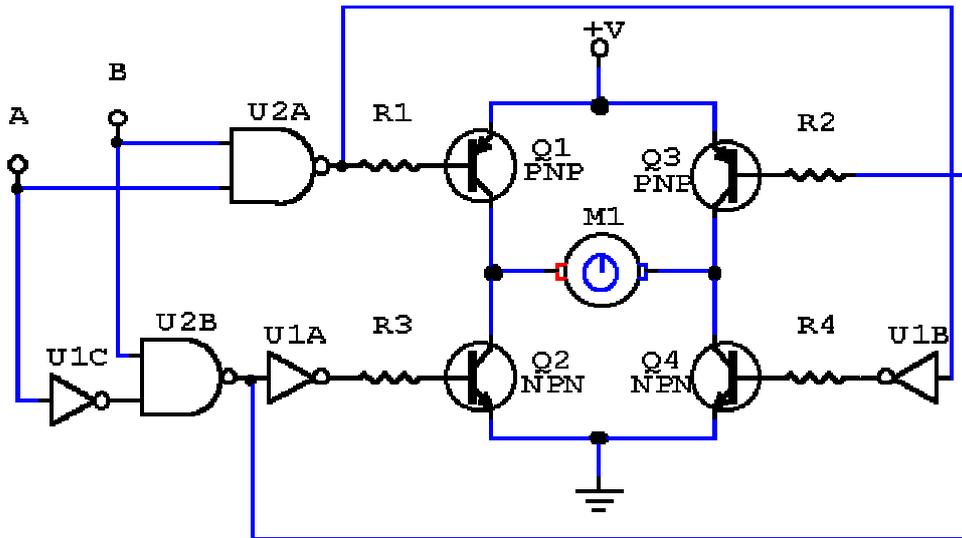


Fig.44. - Circuito per l'inversione del moto di un motore in corrente continua.  
 B = 0 arresta il moto. Il valore logico applicato su A individua il verso di rotazione.

Un bit deciderà la coppia di BJT complementari da porre in conduzione e quindi le polarità di alimentazione del motore. In commercio esistono, tuttavia, opportuni circuiti integrati che svolgono tale funzione. Sono disponibili in commercio sbarre già dotate di sistemi elettromeccanici con finecorsa per l'automazione dell'apertura e chiusura.

I sensori, invece, sono sedici, due per ogni settore. Il sensore posto all'ingresso del settore deve segnalare l'entrata dell'auto, quello posto all'uscita, invece, deve segnalare, appunto, l'uscita dell'auto. Ciascuno di essi può essere realizzato attraverso un fascio di luce generato da una lampadina e rilevato da un trasduttore fotoelettrico come una fotoresistenza, fotodiode o fototransistor. Un semplice circuito elettronico potrà associare alla luce incidente sul trasduttore il livello logico 0 e all'assenza di luce il livello logico 1, cioè il livello logico 0 in assenza di auto ed 1 al passaggio dell'auto. Si decide, tuttavia, di utilizzare degli interruttori a strisciamento come quelli utilizzati dai sistemi automatici di autolavaggio. A riposo l'interruttore è in una posizione fissa e al passaggio dell'auto, l'interruttore viene portato nella posizione estrema opposta. Per evitare rimbalzi meccanici che porterebbero ad un falso conteggio si preferisce un interruttore a deviatore a due posizioni che pilota un circuito soppressore di rimbalzi meccanici come, ad esempio, un semplice flip flop asincrono a porte NAND come mostrato in fig. 45.

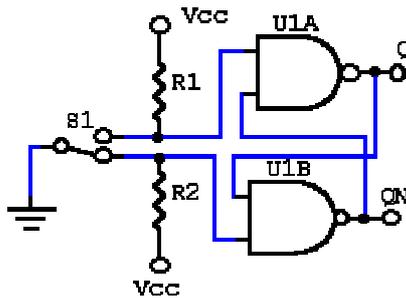


Figura 45. - Deviatore a due posizioni che comanda un antirimbazzo a porte NAND.

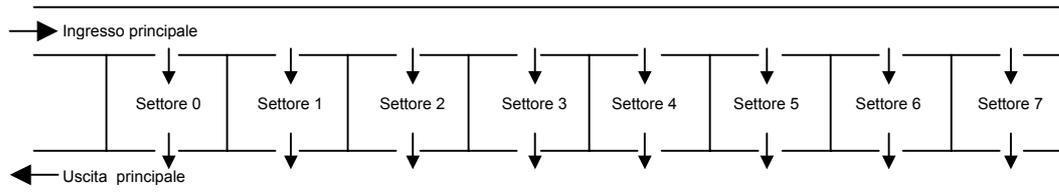
**Risposta al quesito 2.**

È evidente che i sensori utilizzati necessitano di blocchi di condizionamento del segnale perché devono trasformare la presenza/assenza di un'auto in transito in un bit, segnale digitale a due livelli.

Il circuito antirimbando da applicare su ciascun deviatore assume una importanza fondamentale per la corretta valutazione delle auto in transito.

Il particolare sistema ipotizzato non utilizza grandezze fisiche analogiche per cui non si prevede l'impiego di circuiti amplificatori e traslatori di livello.

I sensori saranno disposti all'ingresso e all'uscita di ciascun settore, come si mostra in fig.46.



**Fig.46. - Disposizione del parcheggio. Ogni settore presenta un sensore all'ingresso ed uno all'uscita. L'ingresso di ciascun settore e l'ingresso principale sono dotate di sbarre automatiche comandate dal sistema a microprocessore.**

**Risposta al quesito 3.**

Il sistema di acquisizione al microprocessore deve poter accettare i 16 bit provenienti dai sensori e deve poter comandare le nove sbarre. Si potrebbe far uso di un personal computer e di una scheda di interfacciamento specifica in grado di accettare 16 bit simultaneamente e di comandare, in uscita, le nove sbarre con altrettanti bit. Il sistema di segnalazione all'ingresso può comodamente essere sostituito da un monitor collegato al PC che visualizza in tempo reale i posti liberi di ciascun settore desumendoli dall'elaborazione software dei segnali provenienti dai sensori. In questo modo si evita l'utilizzo di display e relativi decoder pilotati dal PC.

Per l'acquisizione delle 16 linee si decide di utilizzare l'interfaccia standard Centronics, normalmente utilizzata per pilotare le stampanti. Come è noto, il connettore Centronics sul retro del PC è a 25 poli e mette a disposizione 8 linee, normalmente di uscita, comprese tra i pin 2 e 9 ed allocata all'indirizzo di periferica 888. Configurandola opportunamente, tale byte potrà essere programmato di ingresso o di uscita. L'interfaccia presenta, inoltre, 5 bit di ingresso all'indirizzo 889 ed altri 4 bit di uscita all'indirizzo 890.

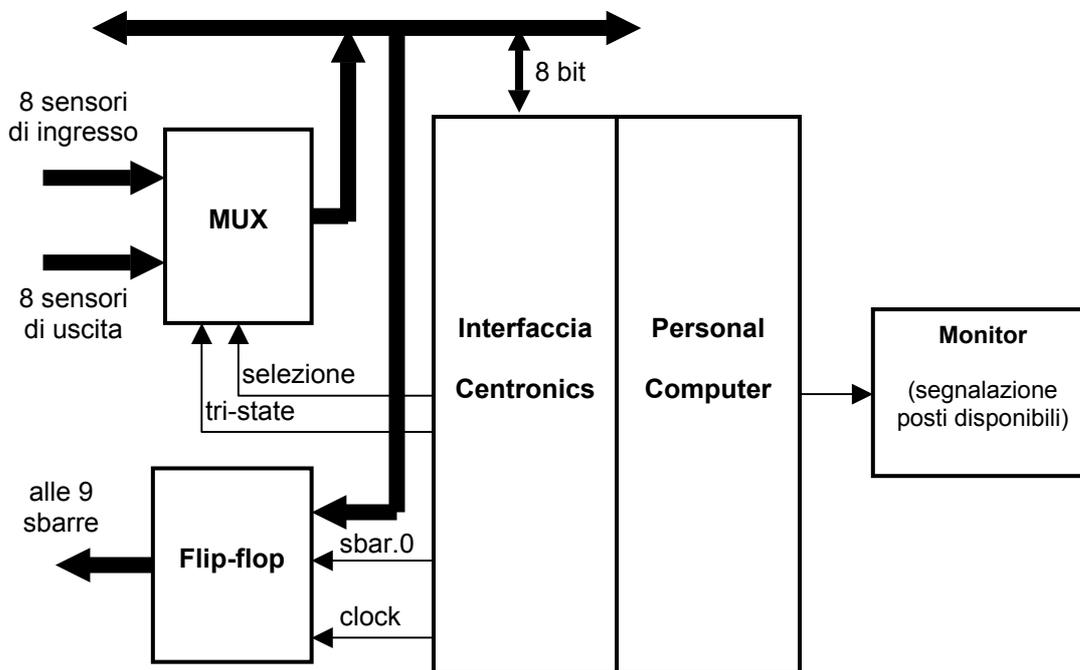
Si decide di realizzare l'acquisizione dei 16 sensori inviando sulla porta di indirizzo 888 programmata in lettura prima 8 bit e successivamente gli altri 8. In questo modo il processo di acquisizione avviene in due fasi distinte avvalendosi di 8 multiplexer digitali a due ingressi. Un bit di controllo, emesso da una linea di uscita della porta di indirizzo 890 della Centronics deciderà quale gruppo di 8 bit

dovrà essere acquisito; un altro bit di controllo porrà le uscite del multiplexer in tri-state.

Il byte presente sulla porta di indirizzo 888 potrà essere programmato come linee di uscita per pilotare 8 delle 9 sbarre. La sbarra dell'ingresso principale sarà pilotata da uno dei 4 bit di uscita della porta di indirizzo 890. In tal caso l'interfaccia hardware esterna al microprocessore è minimizzata all'impiego del solo multiplexer e da 9 flip – flop per memorizzare il pilotaggio delle sbarre.

Il multiplexer potrà essere implementato da due circuiti integrati 74LS257 ognuno dei quali presenta 4 multiplexer a 2 ingressi. Collegandoli in parallelo si ottengono, così, 8 multiplexer a due ingressi. Ciascun integrato, a differenza del normale 74LS157, ha la possibilità, grazie alla linea di controllo OE (Output Enable, attiva bassa), di porre in tri-state le relative linee di uscita per cui, quando il PC comanda le sbarre, le linee di uscita della porta 888 non interferiscono con le uscite dei multiplexer poiché questo sono nello stato di alta impedenza.

Si mostra in fig.47 lo schema a blocchi dell'interfaccia.



**Fig.47. - Schema a blocchi del sistema. Gli 8 bit bidirezionali dell'interfaccia Centronics sono quelli della porta di indirizzo 888. I 4 bit di uscita per la selezione ed il tri-state del multiplexer, il comando della sbarra 0 ed il comando del clock dei nove flip-flop sono prelevati dalla porta di indirizzo 890.**

Scegliamo 4 le linee dell'interfaccia Centronics allocate all'indirizzo 890 per il pilotaggio del multiplexer e dei flip-flop.

Tabella 14

Denominazione della linea nello schema	Bit della porta 890	Denominazione della linea rispetto alla Centronics	Pin
Sbarra 0	A0	STB	1
Clock	A1	AF	14
Tri-state	A2	IP	16
Selezione	A3	SI	17

Per rendere bidirezionale il byte allocato all'indirizzo 888 si deve impostare la Centronics come EPP da setup all'accensione del computer. In particolare:

### 1) 888 di ingresso

il bit 5 della porta 890 (non accessibile sul connettore) deve essere posto a 1. Se A è la parola di uscita impostata sulla 890, per settare a 1 il bit 5 della porta 890 è sufficiente applicare le seguenti istruzioni, valide in Q-Basic:

**B = A OR 32**  
**OUT 890, B**

32 rappresenta il byte 0010000. L'operazione OR viene eseguita bit a bit tra la variabile A e la costante 32. I bit del risultato B coincidono con quelli di A se l'addendo omologo vale 0 mentre si porta a 1 se il bit omologo vale 1.

<b>A</b>	A7	A6	A5	A4	A3	A2	A1	A0	OR
<b>32</b>	0	0	1	0	0	0	0	0	=
<b>B</b>	A7	A6	1	A4	A3	A2	A1	A0	

### 2) 888 di uscita

il bit 5 della porta 890 (non accessibile sul connettore) deve essere posto a 0. Se A è la parola di uscita impostata sulla 890, per settare a 0 il bit 5 della porta 890 è sufficiente applicare le seguenti istruzioni, valide in Q-Basic:

**B = A AND 223**  
**OUT 890, B**

223 rappresenta il byte 11011111. L'operazione AND viene eseguita bit a bit tra la variabile A e la costante 223. I bit del risultato B coincidono con quelli di A se il bit omologo vale 1 mentre si porta a 0 se il bit omologo vale 0.

<b>A</b>	A7	A6	A5	A4	A3	A2	A1	A0	AND
<b>223</b>	1	1	0	1	1	1	1	1	=
<b>B</b>	A7	A6	0	A4	A3	A2	A1	A0	

**Risposta al quesito 4.**

La descrizione dell'algoritmo dell'intera gestione sarà eseguita in modo testuale e sequenziale.

Le funzioni principali sono:

- a) l'acquisizione degli 8 sensori di ingresso;
- b) l'acquisizione degli 8 sensori di uscita;
- c) l'aggiornamento del numero dei posti disponibili e stampe su monitor;
- d) il comando delle sbarre.

Per l'acquisizione degli 8 sensori di ingresso si pone la linea A3 del registro di comando della Centronics di indirizzo 890 al valore: A3=1 (selezione del multiplexer) e la linea A2 al valore: A2=0 (uscite del multiplexer abilitate), si predispose la porta di indirizzo 888 come di ingresso, si esegue l'acquisizione del byte che viene memorizzato nella variabile S\_IN (sensori di ingresso).

Per l'acquisizione degli 8 sensori di uscita si pone la linea della Centronics di indirizzo 890 A3=0 (selezione del multiplexer) e si lascia A2=0, si predispose la porta di indirizzo 888 come di ingresso e si esegue l'acquisizione del byte che viene memorizzato nella variabile S\_OUT (sensori di uscita).

Il software deve poter confrontare i bit della variabile acquisita S\_IN con il valore che la stessa variabile aveva acquisito in precedenza. Se il bit i-esimo di S\_IN è passato da 1 a 0 vuol dire che un'auto è entrata completamente nel settore i-esimo ed il relativo contatore di posti liberi LIBERI(i) viene decrementato di uno. Se, invece, non c'è stata la transizione da 1 a 0, la variabile LIBERI(i) non viene decrementata.

Analogamente il software incrementa LIBERI(i) solo se il bit i-esimo della variabile acquisita S\_OUT ha compiuto, rispetto alla precedente acquisizione, una transizione da 1 a 0.

Se LIBERI(i) = 0 si deve chiudere la sbarra i-esima.

Per determinare il numero totale di posti liberi nell'autorimessa è sufficiente applicare la formula:

$LIBERITOT = \sum LIBERI(i)$  per i che va da 0 a 7.

Se la variabile LIBERITOT diventa minore o uguale a 10 si deve chiudere la sbarra 0 portando a 1 il bit A0 della porta 890 dell'interfaccia Centronics.

L'eventuale comando di chiusura o apertura delle sbarre deve avvenire definendo la variabile SBARRA che contiene un byte da inviare sulla porta 888 della Centronics dopo averla ridefinita di uscita settando a 0 il bit 5 della porta 890 della Centronics.

Dopo tali sequenze di acquisizione, elaborazione e comando sbarre, il software stampa su monitor il resoconto dei posti liberi di ciascun settore. Il monitor è posto all'ingresso principale dell'autorimessa.

**Risposta al quesito 5**

Si sceglie il linguaggio di programmazione Visual Basic.

Si riporta, in fig.48, l'interfaccia grafica di un programma realizzato per simulare il funzionamento dell'autorimessa. Il programma reale deve poter disporre di una

libreria **DLL** che consente di poter utilizzare le istruzioni di I/O verso le periferiche IN e OUT che il Visual Basic non possiede.

Per simulare l'ingresso o l'uscita di un'autovettura in uno degli 8 settori è sufficiente fare click sulla relativa *check box* e poi fare click sul pulsante *Acquisizione*.



Fig.48. - Interfaccia grafica in Visual Basic del programma di simulazione della gestione dell'autorimessa.

L'algoritmo del conteggio delle macchine si attiva dopo l'acquisizione dello stato dei 16 sensori. Di seguito si riporta il listato e relativo commento:

```

Private Sub cmd_acq_Click()
For i = 0 To 7
a = Ck_in(i).Value
b = Ck_out(i).Value
txt_liberi(i) = txt_liberi(i) - a + b
If txt_liberi(i) > 64 Then
txt_liberi(i) = 64
Ck_out(i).Value = 0
End If
If txt_liberi(i) <= 0 Then
txt_liberi(i) = 0
Ck_in(i).Value = 0

```

```

    End If
Next i
liberitot = 0
For i = 0 To 7
    liberitot = liberitot + txt_liberi(i)
Next i
txt_libtot.Text = liberitot
End Sub

```

Alla pressione del pulsante *Acquisizione* per ogni settore da 0 a 7 si indica con a=1 l'entrata di una autovettura e con b=1 l'uscita. Se il sensore di ingresso non rileva l'entrata dell'autovettura sarà: a=0. Analogamente si avrà: b=0 se il sensore di uscita non rileva l'uscita di una macchina dal settore.

Nel listato proposto i posti liberi del settore "i" si inseriscono nella casella di testo **txt\_liberi(i)**.

L'aggiornamento dei posti liberi nel settore "i" sarà, pertanto:

**txt\_liberi(i) = txt\_liberi(i) - a + b**

Cioè il contatore dei posti liberi diminuisce di uno se entra un'autovettura e si incrementa di uno se esce un'autovettura.

Un algoritmo di controllo impedisce al contatore dei posti liberi di assumere valori maggiori di 64 o minori di 0.

L'ultima parte dell'algoritmo ricalcola il valore dei posti liberi complessivi dell'autorimessa semplicemente sommando i posti liberi di ciascun settore. Il risultato *liberitot* viene posto nella casella di testo **txt\_libtot**.

Il programma, infine, deve poter controllare se abbassare una o più sbarre. Quando si aggiorna la casella di testo del numero di posti liberi di ciascun settore e il numero di posti liberi complessivo dell'autorimessa, si attivano due subroutine.

La prima controlla se i posti liberi di un dato settore sono uguali a 0. In tal caso si visualizza sul monitor il disco rosso, **shape1(i)**. Il disco i-esimo torna verde quando si creano disponibilità di posti in quel settore.

La seconda subroutine controlla se il numero totale dei posti liberi dell'autorimessa è uguale o inferiore a 10. In tal caso il disco che segnala il numero di posti totale, **shape2**, si colora di rosso. Se i posti sono in numero superiore a 10 il disco si colora di verde.

Il programma operativo deve, oltre a colorare i dischi sul monitor, comandare contemporaneamente le sbarre in apertura o chiusura. I due listati proposti gestiscono solo l'aggiornamento dei colori dei dischi sul monitor quando si modifica il contenuto delle caselle di testo che contengono il numero di posti liberi.

```

Private Sub txt_liberi_Change(Index As Integer)
For i = 0 To 7
    If txt_liberi(i).Text = 0 Then
        Shape1(i).BackColor = vbRed
    Else
        Shape1(i).BackColor = vbGreen
    End If
Next i
End Sub

```

```
Private Sub txt_libtot_Change()  
If txt_libtot < 11 Then  
    Shape2.BackColor = vbRed  
Else  
    Shape2.BackColor = vbGreen  
End If  
End Sub
```

La pressione del pulsante *autorimessa vuota* consente di inserire nei posti liberi il valore 64 e nei posti liberi totale il numero 512. Tutti i dischi diventano verdi.

```
Private Sub cmd_vuota_Click()  
For i = 0 To 7  
    txt_liberi(i) = 64  
Next i  
txt_libtot = 512  
End Sub
```

Viceversa la pressione del pulsante *autorimessa piena* consente di inserire nei posti liberi il valore 0 e nei posti liberi totale il numero 0. Tutti i dischi diventano rossi.

```
Private Sub cmd_piena_Click()  
For i = 0 To 7  
    txt_liberi(i) = 0  
Next i  
txt_libtot = 0  
End Sub
```

Il pulsante *inizializza* consente di determinare il numero totale dei posti liberi se si decide di inserire manualmente il numero di posti liberi di uno o più settori. Ogni variazione numerica nelle caselle di testo, si ricorda, consente l'aggiornamento del colore del relativo disco e nel programma reale il comando di chiusura o apertura della sbarra.

```
Private Sub cmd_ini_Click()  
liberitot = 0  
For i = 0 To 7  
    liberitot = liberitot + txt_liberi(i)  
Next i  
txt_libtot.Text = liberitot  
End Sub
```

N.B. La traccia del compito non chiede di fornire la soluzione completa ed operativa del problema per cui lo svolgimento proposto non dettaglia l'hardware necessario, sebbene in fig.47 lo schema a blocchi fornito si possa ritenere esauriente. Si lascia allo studente desideroso di realizzare l'apparecchiatura il compito di completare sia l'hardware che il software, corredandolo di istruzioni che eseguono l'acquisizione dai sensori ed il comando sulle sbarre.

## 7.6. Interfaccia parallela esterna per l'I/O analogico e digitale n.2

Si fornisce, di seguito, un'altra interfaccia esterna non standard di tipo parallela per l'esecuzione dell'input e dell'output di un segnale analogico e di dati digitali ad 8 bit. La soluzione qui fornita si avvale di un decodificatore a due bit di ingresso per la selezione di una delle possibili 4 modalità di funzionamento.

Anche in questo caso utilizziamo il registro dati dell'interfaccia Centronics allocata all'indirizzo  $888_{10}$  corrispondente a  $378_{16}$  funzionante come porta di uscita e come porta di ingresso. Si ricordi che, per far funzionare il registro dati come porta di ingresso, si deve innanzitutto settare l'interfaccia parallela da BIOS come porta EPP oppure ECP e successivamente si deve settare ad 1 il bit 5 del registro di controllo allocato all'indirizzo 890. In QBasic tale istruzione è: OUT 890,32.

Anche in Visual Basic è possibile impartire lo stesso comando con la medesima sintassi del QBasic purché si inserisca in C:\Windows oppure C:\Windows\System la libreria INPOUT32.DLL e si includa nel progetto Visual Basic da noi generato il modulo INPOUT32.BAS come si è ampiamente discusso all'inizio del paragrafo 7.5.

Si ripete nella seguente tabella 15 la sintassi da utilizzare per effettuare l'I/O con i più comuni linguaggi di programmazione. Si suppone che il dato da trasferire dal computer al mondo esterno o viceversa sia una variabile in formato byte definita col nome DATO, che la porta di uscita abbia indirizzo memorizzato nella variabile PORTA\_OUT e che la porta di ingresso abbia indirizzo memorizzato nella variabile PORTA\_IN.

Tabella 15

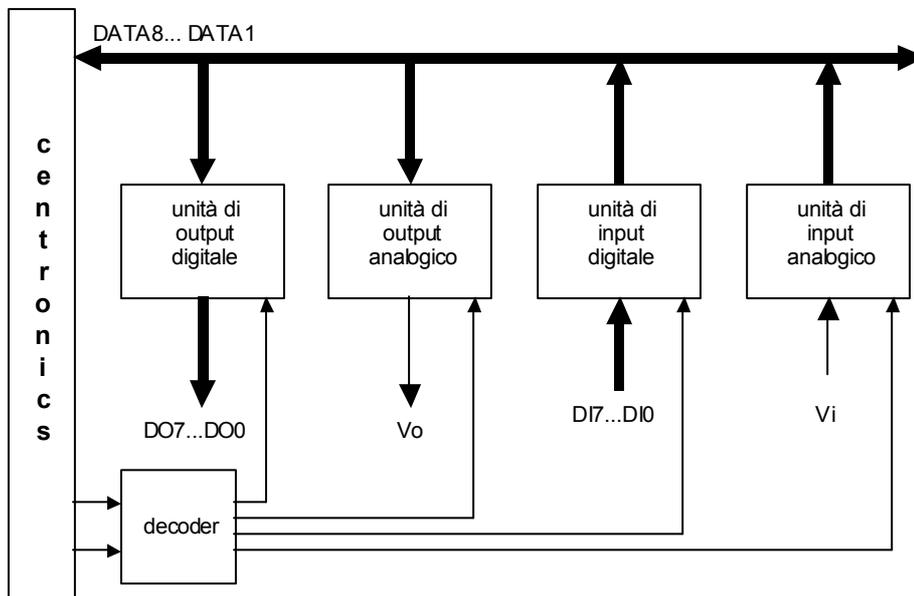
Linguaggio di programmazione	sintassi	commento
C	Outport(PORTA_OUT, DATO)	Istruzione di output. Il contenuto della variabile DATO viene trasferito sulla periferica di indirizzo PORTA_OUT.
	DATO = Inport(PORTA_IN)	Istruzione di input. Nella variabile DATO si memorizza il valore proveniente dalla periferica di indirizzo PORTA_IN
Pascal	Port[PORTA_OUT] := DATO	Istruzione di output. Il contenuto della variabile DATO viene trasferito sulla periferica di indirizzo PORTA_OUT.
	DATO := Port[PORTA_IN]	Istruzione di input. Nella variabile DATO si memorizza il valore proveniente dalla periferica di indirizzo PORTA_IN
QBasic e Visual Basic	Out PORTA_OUT, DATO	Istruzione di output. Il contenuto della variabile DATO viene trasferito sulla periferica di indirizzo PORTA_OUT.
	DATO = In(PORTA_IN)	Istruzione di input. Nella variabile DATO si memorizza il valore proveniente dalla periferica di indirizzo PORTA_IN
assembly	mov DX, PORTA_OUT out DX, AL	Trasferisce in DX l'indirizzo della porta di uscita. Trasferisce il byte contenuto in AL sulla periferica di indirizzo PORTA_OUT.
	mov DX, PORTA_IN in AL, DX	Trasferisce in DX l'indirizzo della porta di ingresso. Trasferisce in AL il byte proveniente dalla periferica di indirizzo PORTA_IN

Nella rappresentazione a blocchi tale interfaccia si può schematizzare con quattro blocchi collegati al bus dati bidirezionale ad 8 bit della porta dati della Centronics. Tali blocchi sono:

- ❑ unità di output digitale costituito da 8 flip-flop
- ❑ unità di output analogico costituito da un DAC ad 8 bit ed uscita in tensione
- ❑ unità di input digitale costituito da 8 buffer tri-state
- ❑ unità di input analogico costituito da un ADC ad 8 bit di uscita ed ingresso in tensione

Un decodificatore a 2 bit e 4 uscite è in grado di abilitare una sola delle 4 unità. I 2 bit di ingresso del decodificatore sono generati da altrettante linee del registro di comando della Centronics.

Si mostra, in fig.49, lo schema a blocchi dell'interfaccia bidirezionale per segnali analogici e digitali.



**Fig.49. – Schema a blocchi del circuito di interfaccia parallela per l'input e per l'output sia digitale che analogico che fa uso di un decodificatore a 4 uscite per la selezione dell'unità periferica da utilizzare.**

In fig.50 si fornisce lo schema dettagliato.

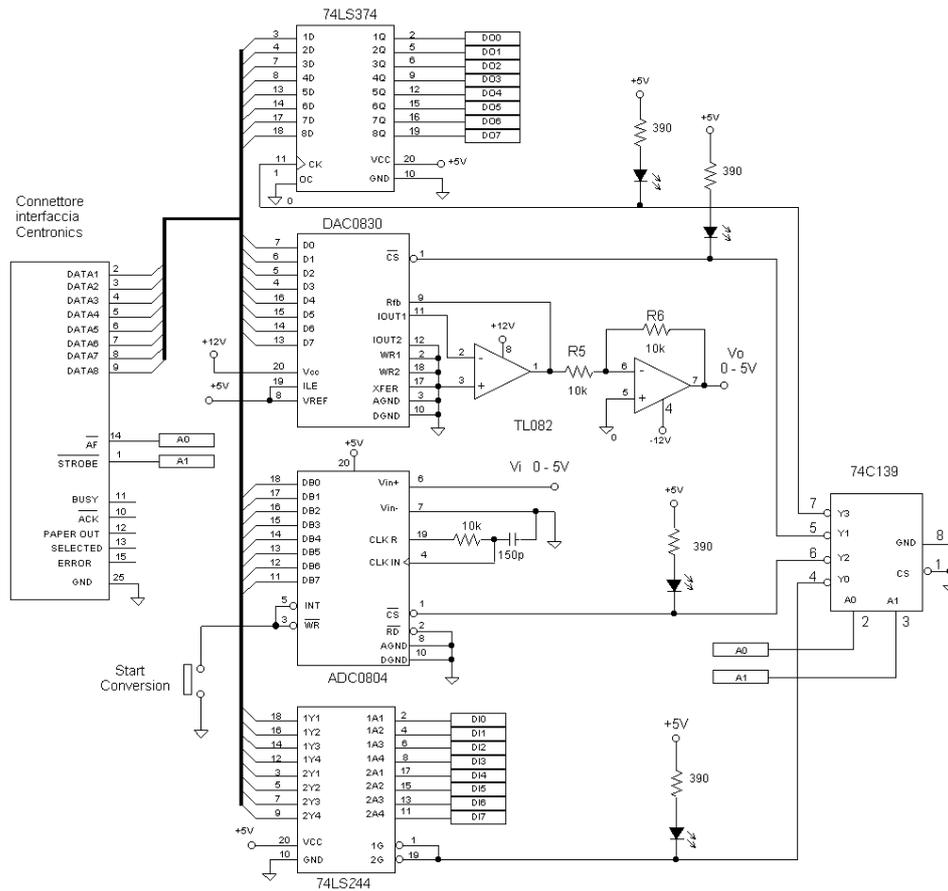


Fig.50. – Interfaccia bidirezionale per l'I/O analogico e digitale.

**Selezione del tipo di operazione**

La selezione del tipo di operazione da effettuare viene affidata al decodificatore 74C139.

Il 139 è un circuito integrato contenente due decoder indipendenti ciascuno a due ingressi (A1 e A0) e quattro uscite (Y0,Y1,Y2,Y3) forniti di una linea di ENABLE indicata con CS (Chip Select) attiva bassa.

In tabella 16 si mostra la tabella della verità dell'integrato 74C139.

Tabella 16

INPUTS			OUTPUTS			
CS	A1	A0	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	0	0	L	H	H	H
L	0	1	H	L	H	H
L	1	0	H	H	L	H
L	1	1	H	H	H	L

Considerando i collegamenti in fig.50, la tabella 16 assume il significato indicato in tab. 17.

Tabella 17

$\overline{\text{STROBE}}$	$\overline{\text{AF}}$	CODICE n	DISPOSITIVO ABILITATO
L	L	3	74LS244 ingresso digitale
L	H	1	DAC0830 Uscita analogica
H	L	2	ADC0804 ingresso analogico
H	H	0	74LS374 uscita digitale

Nella colonna “codice” si è posto il valore n da assegnare alla istruzione out 890, n al fine di attivare una delle quattro modalità operative.

Per abilitare l'ingresso digitale si dovrà utilizzare, ad esempio in QBasic, l'istruzione OUT 890, 3. In tal modo sia  $\overline{\text{STROBE}}$  che  $\overline{\text{AF}}$  si portano al livello basso per cui, del decoder, si attiva al livello logico basso l'uscita Y0 che abilita gli ingressi di gate del buffer tri-state 74LS244. Le uscite Y1, Y2 e Y3 del decoder sono al livello alto per cui gli altri canali sono disabilitati.

Per abilitare l'uscita analogica si dovrà utilizzare, invece, l'istruzione OUT 890, 1 che porta  $\overline{\text{STROBE}}$  al livello basso e  $\overline{\text{AF}}$  al livello alto. Si attiva la combinazione di ingresso 01 del decoder che porta al livello logico basso l'uscita Y1 che, a sua volta, abilita il DAC0830.

Per abilitare l'ingresso analogico si dovrà utilizzare l'istruzione OUT 890, 2 che porta  $\overline{\text{STROBE}}$  al livello alto e  $\overline{\text{AF}}$  al livello basso. Si attiva la combinazione di ingresso 10 del decoder che porta al livello logico basso l'uscita Y2 che, a sua volta, abilita l'ADC0804.

Per abilitare l'uscita digitale, infine, si dovrà utilizzare l'istruzione OUT 890, 0 che porta sia  $\overline{\text{STROBE}}$  che  $\overline{\text{AF}}$  al livello alto. Si attiva la combinazione di ingresso 11 del decoder che porta al livello logico basso l'uscita Y3 che, a sua volta, forza al livello basso l'ingresso di clock del 74LS374. Per attivare la memorizzazione del byte di uscita dell'interfaccia Centronics sulle linee Q del 374 è necessario che il clock, dopo essere passato al livello basso, ritorni al livello alto.

Per far ciò è conveniente attivare il codice che abilita l'uscita analogica: OUT 890, 1. In questo modo Y3 ritorna al livello alto e il 374 memorizza nei suoi flip-flop il byte di uscita e, contemporaneamente, si rende disponibile anche l'uscita analogica Vo con valori compresi tra 0 e 5V.

Per quanto riguarda l'acquisizione analogica, occorre ricordare che, tenendo conto dello schema fornito, il range di ingresso è compreso tra 0 e 5V, il tempo di conversione è di circa 100 $\mu$ s avendo inserito, come rete di timing, una resistenza da 10K $\Omega$  e un condensatore da 150pF. L'ADC0804 funziona in “free

mode” avendo collegato insieme le linee INT e WR ai piedini 3 e 5. In tal caso quando termina una conversione l’uscita INT comanda un nuovo inizio conversione all’ingresso WR. La prima volta, dopo aver dato alimentazione al circuito, potrebbe essere necessario generare manualmente un impulso su WR ed è per questo che si è introdotto un pulsante normalmente aperto, denominato “Start conversion”, in grado di fornire, alla chiusura, un livello logico basso sulla linea WR.

L’uscita analogica è affidata al convertitore D/A avente sigla DAC0830.

Quest’ultimo presenta uscita in corrente per cui è necessario un *convertitore corrente tensione* invertente, realizzato con uno dei due operazionali contenuto nel chip TL082, che fornisce una tensione di uscita compresa tra  $-5V$  e  $0$ . Per rendere positiva tale tensione è sufficiente un amplificatore *invertente a guadagno unitario* che si realizza col secondo operazionale contenuto nello stesso chip TL082 esaminato in precedenza.

I diodi LED inseriti sulle uscite del decoder consentono di osservare visivamente il canale che è stato abilitato sui quattro disponibili.

Il circuito complessivo necessita di un sistema di alimentazione in grado di fornire  $+5V$  agli integrati digitali e  $+12V$  e  $-12V$  agli operazionali e al DAC0830.

## 7.7. Acquisizione analogica tramite scheda audio

È noto che la scheda audio del personal computer è in grado di memorizzare segnali elettrici provenienti da una linea esterna o da un microfono. La scheda presenta un convertitore A/D in grado di trasformare il segnale analogico di ingresso in forma numerica per la sua memorizzazione su hard-disk in formato WAV od altro formato.

Per l’acquisizione di segnali analogici al computer, pertanto, potremmo utilizzare, come canale di ingresso, quello della scheda audio.

Per una corretta acquisizione al fine di una successiva elaborazione dei dati, dobbiamo tenere presente le seguenti condizioni:

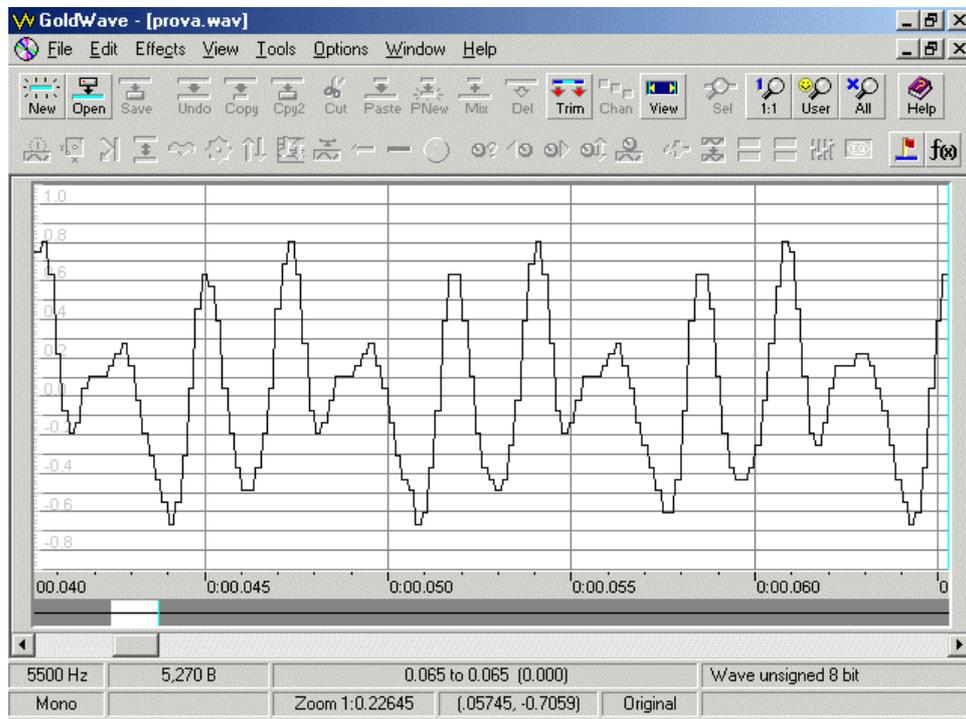
1. l’ingresso è accoppiato in “alternata” con banda passante compresa tra  $20Hz$  e  $20KHz$  circa;
2. l’ampiezza del segnale di ingresso non deve superare  $2V$  picco-picco per evitare distorsioni;
3. l’ingresso può essere mono o stereo e quindi possiamo introdurre due segnali analogici distinti contemporaneamente.

La prima condizione ci impedisce di acquisire segnali costanti o lentamente variabili nel tempo.

La seconda condizione ci impone l’utilizzo di un circuito di condizionamento dell’ampiezza.

La terza condizione ci consente di acquisire simultaneamente anche due segnali analogici alla volta.

Per il software di acquisizione potremmo far uso di quelli di elaborazione audio esistenti in commercio, di tipo “freeware” o di tipo “shareware” disponibile su Internet come, ad esempio, il noto programma Goldwave ([www.goldwave.com](http://www.goldwave.com)) di cui si mostra una schermata in fig.51.



**Fig.51. – Schermata tipica del programma Goldwave per l’acquisizione, l’elaborazione, il salvataggio e la riproduzione di suoni.**

Il programma presenta una serie di funzioni che consentono l’editing del segnale acquisito: selezione, cancellazione, taglia, copia incolla, filtri, effetto doppler, fade in e out, ricerca della massima ampiezza, normalizzazione del segnale ad un valore assegnato, ecc.

Per l’acquisizione del segnale dalla sorgente microfonica, ad esempio, occorre innanzitutto specificare al programma i seguenti dati col comando *File/New*:

- Mono o stereo;
- La frequenza di campionamento;
- La durata massima di registrazione.

Se il segnale da acquisire è uno solo si sceglierà Mono. Se la massima frequenza del segnale da campionare è, ad esempio 1KHz, per il teorema di Shannon la frequenza di campionamento dovrà essere almeno pari al doppio della massima frequenza del segnale. Conviene scegliere come “sampling rate” il valore

5000Hz. Volendo acquisire per mezzo minuto, sceglieremo, come “lunghezza”, 30 secondi.

Il programma Goldwave consente, anche, di generare forme d’onda a partire direttamente da una qualsiasi formula matematica che possiamo inserire attraverso il comando *Tool/Expression Evaluator...* che rimanda ad una finestra in cui applicare la formula o scegliere le funzioni standard disponibili.

La forma d’onda generata può essere inviata in uscita alla scheda audio attivando la funzione *Play* dalla finestra *Device Control* che si attiva dal menù *Tools*.

È ovvio che il segnale di uscita può essere inviato sull’oscilloscopio per la sua visualizzazione.

### 7.7.1. Acquisizione di segnali costanti

Per acquisire segnali costanti o lentamente variabili nel tempo  $V_i$ , come quelli provenienti da un trasduttore di temperatura, è necessario trasformare tale segnale in uno impulsivo con frequenza compresa nella banda di lavoro della scheda audio. Ciò si può realizzare, ad esempio, utilizzando un circuito campionatore che fornisca sull’uscita  $V_o$  un’ampiezza proporzionale a quella del segnale da acquisire.

In fig.52 si mostra lo schema a blocchi di un circuito campionatore.

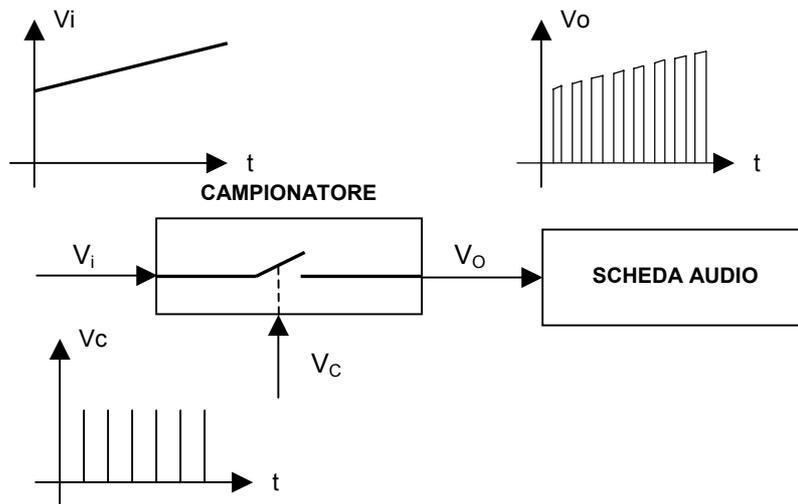


Fig. 52. – Circuito campionatore.

Esso è costituito da un interruttore analogico la cui apertura o chiusura è comandata da un treno di impulsi applicati sulla linea  $V_c$ .

L’involuppo del segnale  $V_o$  consente di ricostruire la forma d’onda del segnale  $V_i$ . Come interruttore analogico si può utilizzare l’integrato CD4046, per segnali unipolari, o l’LF13201 per segnali bipolari.

### 7.7.2. Elaborazione col foglio elettronico

Il file acquisito ed elaborato può essere salvato su disco nei vari formati audio oppure nel formato di testo TXT. Per far ciò è sufficiente aprire *File/Save as...*, assegnare il nome ed impostare il formato TXT dalla casella a discesa *Salva come*.

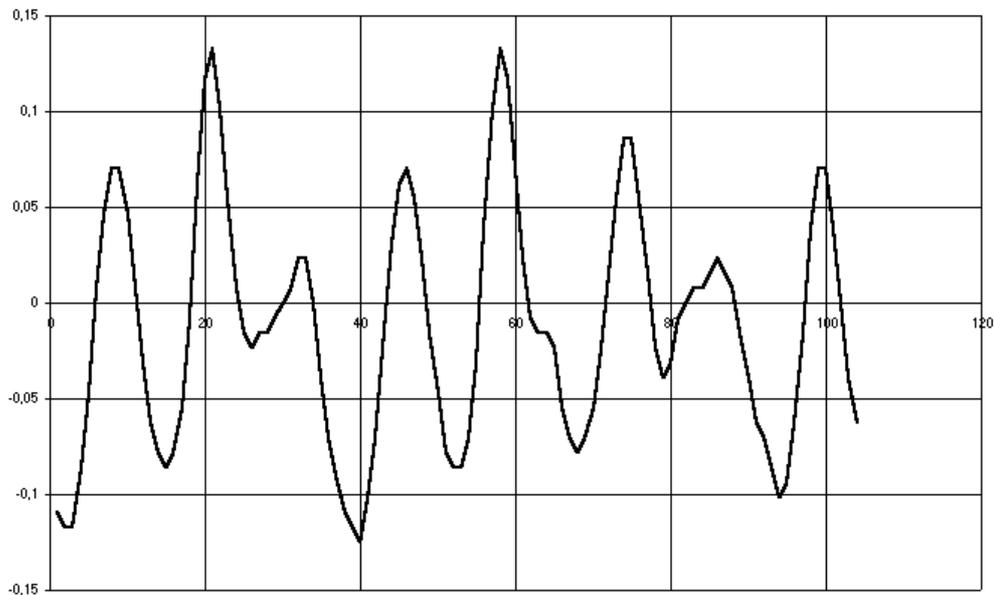
Dopo l'acquisizione ed il salvataggio dei dati nel file di testo è interessante eseguire delle elaborazioni con un foglio elettronico, ad esempio Excel.

Occorre prestare attenzione alla fase di lettura del file di testo da parte del foglio elettronico. In particolare i dati numerici inseriti nel file hanno il punto come delimitatore dei decimali, pertanto, per cambiare il punto decimale in virgola si può procedere in due modi:

1. dopo aver inserito in una colonna i dati numerici, si seleziona la colonna e col comando *Modifica/Sostituisci* è possibile la sostituzione;
2. in fase di lettura del file di testo da parte di Excel, si attiva una procedura automatica di *importazione guidata di testo* che, tra le altre opzioni, consente di indicare il separatore dei decimali.

Si potrà, poi, procedere a tutte le elaborazioni che desideriamo attivare, ad esempio, ricerca del massimo, del minimo, del valor medio, media geometrica, deviazione standard, tendenza, ecc.

Inoltre, com'è ovvio, è possibile la rappresentazione grafica come si mostra nella seguente figura 53.



**Fig.53.** – Grafico ottenuto con Excel relativo ad segnale elettrico acquisito dalla scheda audio ed esportato in formato testo con Goldwave.