

SPEDIAMO I BIT

di Tommaso Pantuso

Protocolli e comunicazioni

Una volta stabilite le regole grammaticali (codici) che permettono la reciproca comprensione tra macchine, bisogna stabilire il modo in cui inviare tali regole lungo le linee di trasmissione, cioè come strutturare le varie forme grammaticali in modo che una macchina che riceve un pacchetto di "zero" e di "uno" possa semplicemente decodificarlo traendone tutte le informazioni necessarie alla comprensione del messaggio. In qualunque trasmissione di dati digitali, ogni informazione viene codificata sotto forma di un insieme di "0" e di "1" ed inviata a "pacchetti" verso l'unità ricevente. Le trasmissioni avvengono sotto il governo di leggi e regole ben precise (e non potrebbe essere altrimenti) il cui insieme prende il nome di *protocollo*: esso stabilisce come debba essere strutturata l'informazione prima di essere inviata sulla linea di comunicazione ed il modo in cui abilitarne la ricezione e la trasmissione in maniera comprensibile. Strutturare un'informazione significa costituirla in modo tale che chi la riceve possa utilmente utilizzarne le parti componenti comprendendo semplicemente le varie suddivisioni tra esse. Detto così può sembrare complicato ma un esempio chiarirà tutto.

Cominciamo col dire che una unità strutturata di informazioni viene denominata *frame* e possiede la generica struttura indicata in figura 1. Come si può verificare dall'illustrazione un frame è composto da tre blocchi fondamentali:

- controllo di trasmissione o *header*;
- corpo del testo da trasmettere o *body*;

- parte terminale di controllo o *trailer*. L'*header* è quella parte del "pacchetto di bit" trasmessi che contiene tutte le informazioni che permetteranno all'apparecchio ricevente di percepire correttamente il messaggio (ad esempio il numero di bit che compongono l'informazione). Il *body* rappresenta il messaggio vero e proprio. Il *trailer* è la parte che avvia il controllo sui dati ricevuti per verificare la presenza di eventuali errori di trasmissione. Supponendo allora di voler inviare dei dati da una stazione ad un'altra, potremmo costituire il frame di comunicazione nella maniera indicata in figura 2a, inviando per prima cosa un insieme di bit che la macchina ricevente interpreterà come "attenzione, sta per arrivare l'*header*" e quindi si predisporrà a sfruttare tutte le informazioni sulla ricezione contenute in esso. Operando in

codice ASCII, questa informazione preliminare sarebbe inviata tramite il carattere SOH (Start Off Header) codificato dal blocco "0000001". Dopo il primo carattere di controllo, possiamo inviare l'*header* ed in coda a quest'ultimo un ulteriore insieme di bit che informi il sistema ricevente che: "è terminato l'*header* e quindi da questo punto in poi preparati a ricevere il testo".

Sempre in codice ASCII l'insieme di bit che separa *header* e testo è rappresentato dal carattere STX (Start Off Text) codificato da "0000010". A questo punto possiamo inviare il testo, anche esso seguito da un blocco che ne indica la fine. Tale blocco corrisponderebbe al carattere ASCII ETX (End Off Text) codificato dal pattern "0000100" e la macchina ricevedendolo lo interpreterà come "il testo è finito, puoi sconnetterti dal trasmettitore". L'ultimo blocco, Block Check Characters, avvia un controllo sui dati effettuato a carico del ricevitore. La figura 2b mostra la codifica del messaggio mediante i caratteri di controllo ASCII citati utilizzati come divisori di zone. Si tenga presente che la definizione di un protocollo fa sì che l'utente possa ignorare tutta la sequenza di operazioni descritte poiché il sistema che, via hardware o software, si occupa sia della costruzione del frame, una volta in possesso del messaggio da inviare, sia della generazione dei debiti segnali elettrici che stabiliscono le modalità con cui deve avvenire lo scambio delle informazioni.

Un esempio di controllo degli errori: la parità

Uno dei problemi che affligge le trasmissioni di dati è il verificarsi di errori nel passaggio di un'informazione da un'unità trasmittente ad una ricevente. Esiste infatti la probabilità che in un blocco di bit passante attraverso una linea di trasmissione uno "0" si trasformi in un "1", o viceversa, a causa della presenza di *fattori disturbanti*, modificando così il significato dell'informazione trasmessa. Ad esempio codificando il carattere A con la parola binaria "1000001" (ASCII), se mentre tale carattere veicola lungo la linea, un disturbo modi-

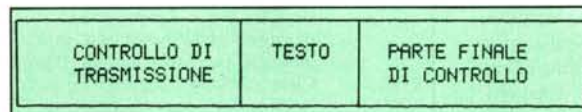


Figura 1
Struttura generica di un frame.

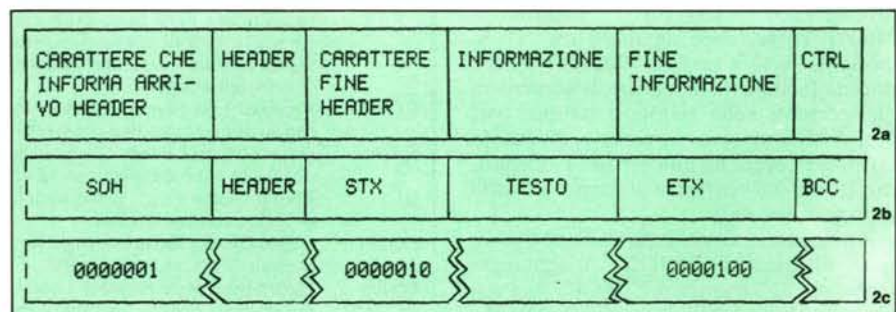


Figura 2 - Ipotesica costruzione di un frame.

fica il secondo bit *una sola volta*, esso diventa un "1" trasformando il carattere in "100011" che non rappresenta più una A ma una C.

Il disturbo su una linea di trasmissione, che possiamo anche chiamare *rumore*, può essere costituzionalmente presente nel sistema o può essere dovuto a fattori accidentali. Un classico esempio di rumore del primo tipo è il cosiddetto *rumore bianco* o gaussian noise che è dovuto alla costituzione stessa della materia essendo introdotto dal moto degli elettroni al suo interno; esso è inoltre un fattore che limita la velocità di trasmissione e la sua influenza cresce con la lunghezza della linea.

Disturbi più fastidiosi perché meno prevedibili sono i *disturbi impulsivi* (impulse noise), introdotti nel sistema da fonti esterne. Ad esempio una lavatrice domestica in funzione introduce sulla linea elettrica degli impulsi causati dallo scintillio delle spazzole del motore; un contatto che si apre e si chiude sulla linea elettrica, ad esempio un interruttore, introduce su di essa del rumore che può andare ad interferire con qualunque apparecchiatura collegata sulla stessa linea. La fonte principale degli errori sui dati sono i disturbi di tipo impulsivo ed in virtù di questo fatto si cerca di realizzare le linee di trasmissione in modo tale da renderle il più possibile immuni all'impulse noise e nello stesso tempo si studiano delle tecniche, più o meno empiriche, per la rivelazione di eventuali errori da cui può essere affetta l'informazione. Tra queste tecniche vogliamo illustrarne una molto semplice e molto diffusa: il *controllo di parità* (fig. 3). Essa consiste nell'aggiungere ad ogni carattere, al momento della trasmissione, una informazione supplementare tramite la quale, in ricezione, è possibile accorgersi se il carattere inviato è stato modificato. In pratica, l'apparecchio ricevente sa se il numero di "1" contenuti nella parola ricevuta deve essere pari o dispari e se esso rileva delle discordanze tra il numeri di "1" ricevuti e quelli che avrebbe dovuto ricevere segnala la condizione di errore. In altre parole, al momento della trasmissione viene aggiunto al carattere da trasmettere un bit in modo che il numero degli "1" in esso contenuti sia sempre pari (*parità pari*) o sia sempre dispari (*parità dispari*). Ad esempio nel caso della parità pari, al carattere "1001101" verrebbe aggiunto alla fine uno "0" ed esso diventerebbe "10011010" mentre a "1000101" verrebbe aggiunto un "1" in modo da farlo diventare 10001011. Come potrete facilmente osservare, il numero di "1" presenti nella parola è sempre pari quindi il ricevitore, informato di questo fatto, se riceverà un numero di "1" dispari, capirà che si è verificato un errore durante il tragitto del dato.

Analogamente, nel caso della parità dispari, al carattere "1001101" si aggiungerebbe un "1" mentre a "1000101" si aggiungerebbe uno "0". Si tenga presente che con tale tecnica si possono correggere solo

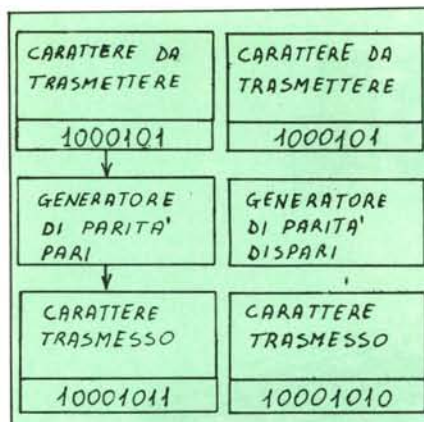


Figura 3 - Esempi di generazione di parità pari e dispari.

errori singoli, cioè se il numero di bit che, ad esempio, passa da zero ad uno è *pari*, si verifica un errore che il ricevitore non può rilevare. Se ad esempio la parola "10011010" contenente un bit di *parità pari*, diventa "1000010", il numero di "1" in essa contenuti è sempre pari quindi per la macchina che riceve non si è verificato nessun errore lungo la linea. Analogamente per la parità dispari. Nonostante questo problema, la tecnica della parità è di gran lunga utilizzata, specialmente quando si trasmettono dati con una *ridondanza intrinseca* (ad esempio dei testi) in cui ci si può facilmente accorgere degli errori che non sono stati rivelati e correggerli. Le cose cambiano quando si ha la necessità di una elevata affidabilità, ad esempio quando si trasmettono dati numerici, nel qual caso si adottano tecniche più sofisticate, basate soprattutto su codici di trasmissione particolari (ad es. Hamming) che permettono, oltre alla rivelazione dell'errore, anche la correzione automatica. Si tenga presente che l'introduzione di bit per il controllo

degli errori rende la parola ridondante, cioè formata da più bit di quelli che in realtà necessiterebbero. Diventando più lunga l'informazione da trasmettere si ha che, a parità di velocità di trasmissione, il numero di parole inviabili diminuisce al crescere della sofisticazione del metodo: quindi, in fase di progetto, bisogna cercare il giusto equilibrio tra la velocità, ridondanza ed affidabilità.

Il numero di bit al secondo trasmessi prende nome di *baud*.

Un esempio di standard

Quando si realizzano dei protocolli di comunicazione si cerca di attenersi ad alcune regole definite da uno *standard di comunicazione*. Uno standard definisce le caratteristiche elettriche di un circuito di interfaccia tra dispositivi ed il tipo di segnali che essi devono scambiare per la corretta ricezione dei dati. Esso è studiato appositamente per favorire la diffusione di periferiche per computer. In altre parole, quando una casa costruttrice realizza un sistema *comodo* di comunicazione, esso rappresenta un potenziale standard ed ha tanto più successo quanto maggiore è il numero di periferiche che vengono poste sul mercato seguendo tale sistema e quanto più grande è il numero dei consensi che incontra tra gli utenti. Tale consenso è molto più imputabile alla praticità dello standard piuttosto che alla perfezione teorica od ai principi avanzati con cui esso è realizzato. A questo punto ad uno standard, per essere considerato tale, non rimane che essere approvato ufficialmente da apposite commissioni (ad esempio l'IEE, The Institute of Electrical and Electronics Engineers).

Aggiungiamo che esistono degli stan-

Principali linee RS 232 C

PG -	Protective Ground - massa che va collegata alla massa delle apparecchiature.
TD -	Transmitted Data - uscita; linea su cui vengono trasmessi i dati.
RD -	Received Data - ingresso; linea su cui vengono ricevuti i dati.
RTS -	Request to Send - uscita; questa linea può essere utilizzata con diverse funzioni. Durante una trasmissione monodirezionale tra una unità trasmittente ed una di sola ricezione, un livello alto su questa linea permette la trasmissione mentre un livello basso la inibisce. Durante una trasmissione bidirezionale su una linea, un livello alto su RTS pone l'apparecchio in trasmissione mentre un livello basso lo pone in ricezione. In altri casi una transizione di questa linea da un livello basso ad uno alto indica al ricevitore che chi trasmette ha i dati pronti.
CTS -	Clear to Send - ingresso; una condizione alta su CTS indica la disponibilità alla ricezione dei dati trasmessi su TD.
DSR -	Data Set Ready - ingresso; in condizione alta è analoga alla precedente; se va a livello basso prima della linea DTR (vedi più avanti) la macchina interpreta tale evento come "perdita di connessione tra ricevitore e trasmettitore".
SG -	Signal Ground - massa del segnale; punto rispetto a cui si misurano i livelli dei segnali di comunicazione.
RLSD -	Received Line Signal Detector - ingresso; una condizione alta su questa linea indica che si stanno ricevendo i corretti segnali di comunicazione.
DTR -	Data Terminal Ready - uscita; quando questa linea è attiva (alta), trasmettitore e ricevitore sono connessi per la comunicazione.
RI -	Ring Indicator - ingresso; questa linea è collegata al campanello telefono quando si è collegati con un modem.
SQD -	Signal Quality Detector - ingresso; quando questa linea è attiva, vuol dire che nel dato ricevuto è presente un errore.
DSRS -	Data Signal Rate Selector - questa linea, nei sistemi a due velocità, seleziona quella più o meno elevata a seconda che si trovi a livello alto od a livello basso.

dard di fatto, cioè non ufficialmente definiti, che sono però stati resi popolari dal vasto impiego su alcuni tipi di periferiche; tra questi ricordiamo il sistema Centronics, utilizzato spessissimo per la comunicazione di dati da parte di un elaboratore ad una stampante ed il sistema a Current Loop utilizzato su teletype.

Esaminiamo di seguito, per fornire un esempio concreto delle questioni trattate, un noto standard, l'RS 232C che definisce le modalità di trasmissione seriale di informazioni, descrivendo alcune delle più importanti linee e la loro utilizzazione nell'ambito di una interfaccia elettrica.

In figura 4 forniamo l'elenco completo dei segnali RS 232C. Il connettore da utilizzare su un'interfaccia RS 232 è un Cannon a 25 poli (figura 5) ed ogni piedino di tale connettore ha, in qualunque periferica od unità centrale, lo stesso significato (figura 6): è questo uno dei vantaggi di uno standard. Delle 25 linee citate, per realizzare un efficace collegamento tra unità centrale ed unità periferica ne bastano da tre a

cinque. Analizziamo le linee principali che permettono un potenziale collegamento illustrandone sia la funzione nell'ambito del protocollo, sia il modo in cui i segnali di controllo concorrono al coordinamento della trasmissione.

Supponiamo di voler trasmettere dei dati da un computer centrale ad un terminale utilizzando lo standard di cui stiamo parlando. Per prima cosa avremo bisogno di una linea su cui inviare i dati e sceglieremo la linea che fa capo al piedino 2 del connettore, cioè quella di *Trasmissione Dati*. I dati non possono essere inviati in maniera casuale ma devono essere controllati dai segnali generati su delle linee particolari. Supponendo di avere i dati pronti nel computer, dovremo per prima cosa inviare da esso un segnale verso l'unità periferica che la avvisi della disponibilità degli stessi. Tale segnale viene generato sulla linea *Request To Send*: quando essa passa da un livello basso ad uno alto la periferica capisce che l'unità centrale possiede i *dati pronti*. Per livello alto si intende in questo stan-

dard un valore di tensione superiore a +3 volt (tipicamente +12 volt) e per livello basso una tensione inferiore a -3 volt (tipicamente -12 volt). Aggiungiamo che un livello alto viene qualificato in questa sede come uno "0" binario mentre un livello basso corrisponde ad un "1".

A questo punto se la periferica è libera, cioè se ha completato la ricezione di dati precedenti, invia un segnale sulla sua linea *Clear to Send* (mandandola da un livello alto ad uno basso di tensione) ed avverte il computer che è disponibile a ricevere i dati.

La possibile procedura descritta, che fa uso di linee di controllo per la coordinazione dell'invio dei dati, prende il nome di *handshake*. L'esempio fornito serve per rendere l'idea di come possa verificarsi una trasmissione in maniera molto semplice; per trasmissioni che richiedono più segnali di controllo si usa un numero maggiore di linee. Per finire, presentiamo nel riquadro alcuni fra i più importanti segnali previsti dallo standard con una sommaria descrizione delle funzioni svolte. MC

LINEE RS 232 C

CIRCUITI DI MASSA

AA PROTECTIVE GROUND
AB SIGNAL GROUND

CIRCUITI DATI

BA TRANSMITTED DATA
BB RECEIVED DATA

CIRCUITI DI CONTROLLO

CA REQUEST TO SEND
CB CLEAR TO SEND
CC DATA SET READY
CE RING INDICATOR
CF RECEIVED LINE SIGNAL DETECTOR
CG SIGNAL QUALITY DETECTOR
CH DATA SIGNAL RATE SELECTOR (OUT)
CI DATA SIGNAL RATE SELECTOR (IN)

CIRCUITI DI TEMPORIZZAZIONE

DA TRANSMITTER SIGNAL ELEMENT TIMING (OUT)
DB TRANSMITTER SIGNAL ELEMENT TIMING (IN)
DD RECEIVER SIGNAL ELEMENT TIMING (IN)

CIRCUITI SECONDARI

SBA SECONDARY TRANSMITTED DATA
SBB SECONDARY RECEIVED DATA
SCA SECONDARY REQUEST TO SEND
SCB SECONDARY CLEAR TO SEND
SCC SECONDARY RECEIVED LINE SIGNAL DETECTOR

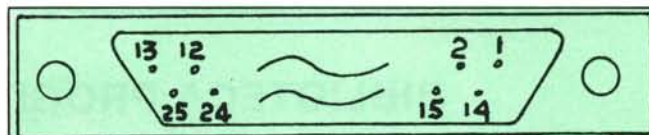


Figura 5 - Disposizione dei pin di un connettore d'interfaccia RS 232 femmina visto di fronte.

NUMERO	DESCRIZIONE
1	PROTECTIVE GROUND
2	TRANSMITTED DATA
3	RECEIVED DATA
4	REQUEST TO SEND
5	CLEAR TO SEND
6	DATA SET READY
7	SIGNAL GROUND
8	RECEIVED LINE SIGNAL DETECTOR
9	(RESERVED FOR DATA SET TESTING)
10	(RESERVED FOR DATA SET TESTING)
11	UNASSIGNED
12	SECONDARY RECEIVED LINE SIGNAL DETECTOR
13	SECONDARY CLEAR TO SEND
14	SECONDARY TRANSMITTED DATA
15	TRANSMISSION SIGNAL ELEMENT TIMING (IN)
16	SECONDARY RECEIVED DATA
17	RECEIVER SIGNAL ELEMENT TIMING (IN)
18	UNASSIGNED
19	SECONDARY REQUEST TO SEND
20	DATA TERMINAL READY
21	SIGNAL QUALITY DETECTOR
22	RING INDICATOR
23	DATA SIGNAL RATE SELECTOR
24	TRANSMIT SIGNAL ELEMENT TIMING (OUT)
25	UNASSIGNED

Figura 6 - Descrizione dei terminali del connettore RS 232.

TA

TRIUMPH-ADLER

- Modelli: PC -P3 - P4
- Minifloppy: da 320 Kbytes a 785 Kbytes
- Hard disk: da 5 Mbytes
- Video a fosf. verdi: 24 x 80 car.
- Stampanti:
ad aghi DRH80-DRH136-DRS250;
a margherita TRD170S
- Macchine da scrivere interfacciabili
- Linguaggi: BASIC - PASCAL
FORTRAN IV - COBOL
- Prezzi da L. 1.050.000
a L. 9.600.000



BIBLIOTECA PROGRAMMI ALPHATRONIC

Contabilità generale: Partitari - Situazione contabile - Registri IVA - Denunce e allegati annuali IVA.

Contabilità semplificata: Registri IVA - Riepiloghi periodici - Situazione contabile - Elenco clienti e fornitori.

Paghe e stipendi: Cedolino - Quadrature - Elaborazioni mensili - Servizi annuali.

Amministrazione condominiale: Ripartizione - Acconti - Spese - Fornitori.

Magazzino e fatturazione: Gestione preventivi; conferma d'ordine, bolle in automatico con fatturazione (immediata, differita, accompagnatoria) in codice alfanumerico - Listino - Giornale - Inventari valorizzati: prezzi di acquisto - Inventari valorizzati: prezzi di vendita - tratte e ricevute bancarie - Statistica di vendita.

Legge 373: Calcolo e progettazione delle dispersioni termiche di un edificio.

Programmi di utilità: Cross-reference - Dump memoria/disco - Routine in assembler - Auto-index.

Ingegneria civile/2: Strutture semplici e frequenti.

Medicaldata: Visite mediche - Analisi - Scheda sanitaria - Controllo economico.

Mercati Ortofrutticoli: Gestione partite, ordini, fatture, clienti, fornitori, stampe riepilogative di fine giornata, trasmissione dati per studi commerciali.

Ottici e Lenti Corneali: Gestione archivio clienti con prima e successive prescrizioni. Gestione magazzini: Lenti corneali - Ottica - Materiali - Sbozzi e semilavorati. Statistiche.

Ingegneria in regime sismico - Data Base - Text-editor - Mailing list - Alberghi - Case di spedizionieri e trasporti - Controlli numerici - Laboratori analisi - Collegamento HP-3000 come terminale intelligente - Gestione assicurazioni - Word-processing (utilizzabile con Triumph-Adler SE 1010 o SE 1005)

Disponibili schede di tipo: Parallela - Grafica ad alta risoluzione - Seriale - Orologio - Eprom Programmer.

RIVENDITORI AUTORIZZATI TRIUMPH-ADLER:

MILANO
FIRENZE
VIAREGGIO (LU)
PESCARA
L'AQUILA
ROMA
ROMA
ROMA
ROMA
MONTEROTONDO (RM)
NAPOLI

S.C.E.D. ELECTRONIC s.r.l.
SOFTLAB s.r.l.
LOGICA COMPUTER s.a.s.
ITALDATA s.r.l.
S.E.A.D. s.r.l.
EMMEPI s.n.c.
PRIMAPRINT s.r.l.
SIPEP s.n.c.
TECNOSINTESI s.n.c.
LINEA UFFICIO s.n.c.
C.I.E.D. s.r.l.

Via Meda 15 - Tel. 02/8394486
V.le Europa 101 - Tel. 055/683859
Via Fratti 165 - Tel. 0584/42704 - 3532352
Via Tiburtina 75 - Tel. 085/50843 - 54800
Via dei Giardini 16/a - Tel. 0862/315754
Via Accademia dei Virtuosi 7 - Tel. 06/5410273
Via Ulisse Dini 10 - Tel. 06/5578338
Via Amelia 32 - Tel. 06/787013
Via G. Ricci Curbastro 34/b - Tel. 06/5563676
Via S. Martino 36 - Tel. 06/9001082
P.zza Garibaldi 82, S. Giuseppe Vesuviano
Tel. 081/8282545 - 8273625