

Caro Kiwi,

Come promesso eccomi a confutare le tue affermazioni.

Tanto per cominciare, sembra quasi che tu pensi che Laurer un bel giorno si sia svegliato e abbia detto: “Oh che bello, oggi invento il codice a barre”. Chiaramente non è così. Laurer lavorava per l’IBM, e aveva fior di specifiche che gli erano imposte dall’alto. Non solo, ma c’era competizione fra diverse aziende, quindi quelle specifiche non erano in alcun modo modificabili. Altre specifiche gli erano imposte dalle tecnologie dell’epoca.

Inizialmente, il codice a barre era stato pensato per essere letto con un fascio laser passato, meccanicamente o addirittura a mano, sopra al codice. Questo poneva una serie di vincoli legati alla dimensione del “puntino” formato dal laser, della quantità di luce utilizzabile, ecc. Inoltre è chiaro che, se avesse avuto a disposizione la potenza di calcolo che c’è oggi, Laurer avrebbe potuto fare scelte diverse (che sono quelle fatte per i codici più moderni), ma all’epoca, ricordiamocelo, i microcalcolatori praticamente non esistevano ancora: l’8080 è uscito nel ’74, il suo predecessore – l’8008 – era inusabile, e si andava avanti a forza di PLA, di PLS e soprattutto di logiche cablate progettate ad hoc. Questo anche perché non esistevano calcolatori abbastanza veloci da poter elaborare il segnale proveniente dalla fotocellula in tempo reale.

Vediamo allora di ricostruire il processo mentale che ha portato alla definizione del codice così com’è.

Primo: il codice deve essere descritto mediante un alfabeto a due soli simboli, bianco e nero, per ovvie ragioni legate alla semplicità di stampa e a quella del lettore. Questo esclude, per esempio, diverse tonalità di grigio, diversi colori, ecc. Inoltre, per massimizzare il rapporto segnale/disturbo nel segnale fornito dal lettore, bisogna massimizzarne il contrasto: di qui l’imperativa necessità di usare il “no inchiostro” per il bianco e il “tutto inchiostro” per il nero. Ah, dimenticavo: si voleva anche che fosse stampabile con le stesse macchine usate per stampare le etichette dei prodotti, e con le stampanti da calcolatore disponibili all’epoca (ad aghi, termiche o “a catena”).

Secondo: l’etichetta ha dimensioni estremamente variabili e incognite a priori, sia perché variano proprio le sue dimensioni fisiche, sia perché variano la velocità e la direzione con cui il laser passa sul codice, e anche (nei lettori a distanza) varia appunto la distanza a cui viene fatta la lettura. Questo significa che la frequenza base del segnale (il tempo occorrente per scandire un bit) può assumere valori molto diversi e variabili anche durante una singola lettura. Quindi era imperativo usare un codice di tipo sincrono, ma, essendoci un solo canale di trasmissione, il segnale di sincronismo doveva essere conglobato insieme con l’informazione.

Come facciamo per ottenere tutto questo? Beh, innanzitutto dobbiamo capire dove comincia il codice. Per far questo occorre, evidentemente, una zona sufficientemente larga in cui non ci siano transizioni, seguita da una barra nera. La prima zona, detta “quiet zone”, serve a permettere al lettore di distinguere l’inizio del codice da altri pattern. Poi ci deve essere una transizione. Qui le alternative sono due: o facciamo la zona di guardia bianca e poi ci mettiamo una barra nera, o facciamo la zona di guardia nera e poi ci mettiamo una barra bianca. Ma questa seconda ipotesi è chiaramente improponibile: comporterebbe un enorme spreco di inchiostro e sarebbe esteticamente orribile.

Terzo: Abbiamo capito dove comincia il codice. Possiamo cominciare a decodificarlo. Ma c’è ancora un problema: per andare d’accordo con tutte le specifiche, in particolare con quelle concernenti la dimensione massima dell’etichetta, Laurer aveva previsto che il codice fosse diviso in due parti, in modo da poterle leggere anche separatamente. Questo permette di fare, al contrario di tutti i suoi concorrenti, le barre abbastanza larghe da poter essere effettivamente leggibili, ed è quindi il punto vincente della sua realizzazione. Ma per distinguere le due metà del codice occorre un carattere palesemente “sbagliato”, in modo che il lettore se ne accorga. Siccome il concetto di

“carattere” implica quello di “bianco nero bianco nero” (altrimenti il lettore non capisce nulla), e questo carattere deve essere uguale in qualunque senso sia letto, il più piccolo carattere che permette questo è appunto 01010. Per altri dettagli vedi la risposta al tuo punto 5. Ma a questo punto, per mantenere tutto simmetrico, dobbiamo inserire lo stesso pattern anche alle altre estremità delle due metà dell’etichetta.

Alla luce di queste considerazioni, la risposta ai tuoi punti può essere questa:

1) *Il fatto di aver adottato tre codifiche per le cifre anziché due o quattro è arbitrario.*

Non capisco cosa vuoi dire. Le codifiche per ogni cifra sono due: quella a parità dispari e quella a parità pari. La ragione è che il codice doveva poter essere letto con qualunque orientamento, e quindi bisognava poter distinguere le cifre che appartengono alla metà di sinistra da quelle che appartengono alla metà di destra.

2) *Il fatto di aver inserito particolari combinazioni di barre per ogni cifra (alternate da spazi vuoti) è arbitrario.*

No. è arbitraria l’attribuzione delle cifre alle combinazioni di barre, ma le combinazioni non sono libere: vedi più avanti perché. A proposito: se con il termine “arbitrario” intendi “può essere scelto qualunque” non è vero neanche quello: le attribuzioni possibili sono  $10!$ , cioè 3.628.800. Non è un numero piccolo, ma è ben lontano dall’essere infinito. E fra un attimo scoprirai che neppure questo è vero: in realtà sono  $6!$ , cioè solo 720.

3) *Il fatto di aver creato delle cifre con delle barre alternate da altri spazi vuoti per un totale di 7 moduli (e non 8 o 6 o 10) è arbitrario.*

Per nulla. È ben noto che per rappresentare  $n$  cose diverse con un alfabeto di  $k$  simboli occorrono  $\log_k n$  “moduli”, come li chiami tu, approssimato all’intero superiore. Nel nostro caso, le specifiche di progetto richiedevano di poter rappresentare con codifiche distinte le 10 cifre numeriche, per la parte sinistra e per la destra, per un totale quindi di 20 simboli. Siccome  $k$  è pari a 2 (bianco o nero) il numero minimo di “moduli” è  $\log_2 20 \approx 4.32$  che arrotondato all’intero superiore fa 5. Ma attenzione: nelle specifiche di progetto ci sono altre due condizioni importantissime: (a) la codifica di ogni cifra deve cominciare con un bianco e terminare con un nero, per mettere il lettore in grado di ri-sincronizzarsi all’inizio di ogni carattere, e la seconda è che ogni cifra deve contenere esattamente una parte bianca, una nera, un’altra parte bianca e un’altra nera. Ora, se tu provi a scrivere le 32 combinazioni che puoi fare con 5 simboli binari, e le fai precedere da un bianco e seguire da un nero, ottieni questa tabella:

B	B	B	B	B	B	N		B	N	B	B	B	B	N
B	B	B	B	B	N	N		B	N	B	B	B	N	N
B	B	B	B	N	B	N		B	N	B	B	N	B	N
B	B	B	B	N	N	N		B	N	B	B	N	N	N
B	B	B	N	B	B	N		B	N	B	N	B	B	N
B	B	B	N	B	B	N		B	N	B	N	B	B	N
B	B	B	N	N	B	N		B	N	B	N	N	N	N
B	B	B	N	N	N	N		B	N	B	N	N	N	N
B	B	N	B	B	B	N		B	N	N	B	B	B	N
B	B	N	B	B	N	N		B	N	N	B	B	N	N
B	B	N	B	N	B	N		B	N	N	B	N	B	N
B	B	N	B	N	N	N		B	N	N	N	B	N	N
B	B	N	N	B	B	N		B	N	N	N	B	B	N
B	B	N	N	N	B	N		B	N	N	N	N	B	N
B	B	N	N	N	N	N		B	N	N	N	N	N	N

Ho segnato in rosso le combinazioni che non si possono usare perché non contengono esattamente una parte bianca, una nera, una bianca e una nera. Come vedi, ne restano venti, di cui dieci hanno un numero pari di bianchi e dieci un numero dispari. Ho indicato le prime con il colore verde e le seconde con il blu. Queste sono le codifiche che possiamo utilizzare: le verdi per codificare le dieci cifre quando stanno a sinistra e le blu per codificare le stesse cifre quando stanno a destra.

La codifica che è stata scelta è mostrata qui sotto.

B	B	B	B	B	B	N	n.a.		B	N	B	B	B	B	N	3dx
B	B	B	B	B	N	N	n.a.		B	N	B	B	B	N	N	4sx
B	B	B	B	N	B	N	6dx		B	N	B	B	N	B	N	n.a.
B	B	B	B	N	N	N	n.a.		B	N	B	B	N	N	N	0dx
B	B	B	N	B	B	N	8dx		B	N	B	N	B	B	N	n.a.
B	B	B	N	B	N	N	9sx		B	N	B	N	B	N	N	n.a.
B	B	B	N	N	B	N	0sx		B	N	B	N	N	B	N	n.a.
B	B	B	N	N	N	N	n.a.		B	N	B	N	N	N	N	6sx
B	B	N	B	B	B	N	7dx		B	N	N	B	B	B	N	5sx
B	B	N	B	B	N	N	2sx		B	N	N	B	B	N	N	1dx
B	B	N	B	N	B	N	n.a.		B	N	N	B	N	B	N	n.a.
B	B	N	B	N	N	N	9dx		B	N	N	B	N	N	N	8sx
B	B	N	N	B	B	N	1sx		B	N	N	N	B	B	N	5dx
B	B	N	N	B	N	N	2dx		B	N	N	N	B	N	N	7sx
B	B	N	N	N	B	N	4dx		B	N	N	N	N	B	N	3sx
B	B	N	N	N	N	N	n.a.		B	N	N	N	N	N	N	n.a.

Allora, resta dimostrato che meno di sette “moduli” non si potevano usare, perché non avrebbero consentito tutte le configurazioni occorrenti. Se ne sarebbero potuti utilizzare di più, certo, ma questo avrebbe violato un’altra importante specifica di progetto: utilizzare il minimo numero di barre possibile, perché, così facendo, a parità di spazio totale occupato, ogni modulo risulta più largo, aumentando le tolleranze ammissibili in fase di stampa e riducendo la risoluzione necessaria del sensore. In altre parole, minimizzare il numero di moduli equivale a minimizzare il costo del sistema.

In conclusione: meno di sette moduli non si potevano usare per ragioni matematiche, e più di sette sarebbero stati inutilmente ridondanti e dannosi. Come vedi, non è affatto una scelta arbitraria.

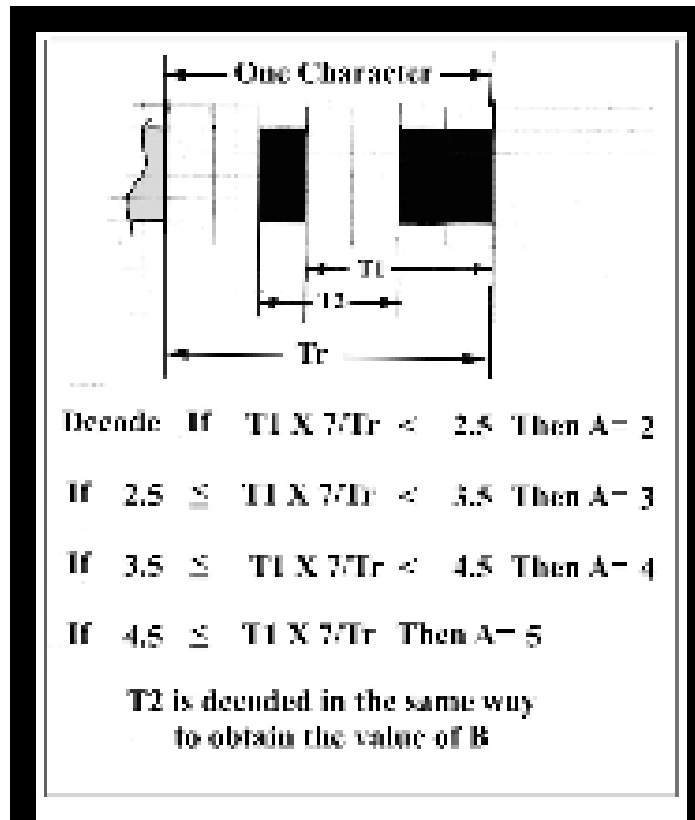
4) Il fatto di "dire" che la cifra è la combinazione di sette moduli particolari e non piuttosto la combinazione di barre nere con spazi vuoti esterni per un totale di sette moduli è arbitrario.

Certo: il lettore non è in grado di distinguere fra spazi esterni e barre bianche. Però ti dimentichi che, secondo la definizione di cifra nel codice UPC, la prima di queste barre deve essere per forza bianca e l’ultima deve essere nera, altrimenti non si può trattare di una cifra. Questo esclude, comunque sia vista, la bar guard centrale.

Ma c’è molto di più. Scolari (e anche tu) crede che l’informazione nel barcone sia data dalla combinazione di barre bianche e nere. Non è vero. L’informazione in realtà è data dai rapporti fra le distanze di transizioni omologhe, e più precisamente:

- L’inizio e la fine del carattere (transizioni nero-bianco);
- La fine della prima barra nera e la fine della seconda (che poi è anche la fine del carattere);
- L’inizio della prima barra nera e l’inizio della seconda.

Lo schema quindi è questo:



quindi T1 e T2 possono assumere ognuno 4 valori diversi, il che dà 16 possibili combinazioni. Di queste, 8 sono relative alle codifiche con parità pari e 8 a quelle con parità dispari, secondo questo schema:

A	5	6 odd	0 even	4 odd	3 even
	4	9 even	2 or 8 odd *	1 or 7 even *	5 odd
	3	9 odd	2 or 8 even *	1 or 7 odd *	5 even
	2	6 even	0 odd	4 even	3 odd
		2	3	4	5
		B			

\* The value of C is required to determine if one of these characters is a 1, 2, 7, or 8

C1=1	2 odd 8 even	1 odd 7 even	C2=1
C1=2	8 odd 2 even	7 odd 1 even	C2=2
	A=3	A=4	

Ci sono otto pattern che danno, a coppie, gli stessi valori di A e di B (3 o 4), e per cui in questo caso occorre calcolare un terzo rapporto per poter discriminare. Queste codifiche sono state riservate alle

cifre che, da una statistica fatta, risultavano le meno utilizzate: 1, 2, 7 e 8. Alle altre 6 cifre (0, 3, 4, 5, 6 e 9), visto che non c'era nessun criterio utilizzabile per semplificare la rete di transcodifica verso il codice BCD che avrebbe dovuto alla fine uscire dal lettore, e che quindi sarebbe stato necessario usare una lookup table, è stata effettivamente data una codifica a caso. E non esiste nessun criterio per dire che un'altra codifica sarebbe stata migliore, anzi, si può dimostrare che sono tutte equivalenti.

5) *Il fatto di aver adottato tre caratteri di controllo anziché due o quattro è pure arbitrario.*

Ancora una volta, no! Una bar guard all'inizio serve per sincronizzare il lettore. Una alla fine serve per sincronizzarlo se la lettura inizia da destra. Quello al centro serve perché... lo spiega Laurer, non c'è bisogno che lo ripeta io:

*The center pattern, which is shared by both the right and left halves of the symbol, serves to tell the logic which direction the beam is crossing the symbol (note that scanners need not read both halves of the symbol on the same pass nor in the same direction). The distance from the starting edge to the end of the second bar is 4 modules. The logic determines that this "character" is much more narrow than the others, thus it is the center. If 0010100 were used the character width would be 5 modules, still quite narrow, but less distinctive. Another reason, perhaps more important, is it reduces the overall width by 2 modules allowing the printing tolerances to be increased ever so slightly. Nothing would be gained by the wider center character, 0010100.*

Quindi tre bar guard occorrono, non se ne può fare a meno se non violando le specifiche di progetto. Metterne di più sarebbe dannoso per quanto detto al punto precedente.

Convinto? Non so se sono stato abbastanza chiaro: ho cercato di condensare in uno spazio decente una materia che in realtà è molto ma molto più complessa di quanto possa apparire a prima vista.

Per approfondire, ti citerei un articolo che avevo letto tempo fa, ma che non ho e non sono riuscito a procurarmi perché la bibliotecaria è in ferie e in biblioteca, guarda caso, ci manca proprio il volume che servirebbe a me:

Pavlidis, T., Swartz, J. E Wang, Y.P.: "Fundamentals of bar code information theory", *IEEE Spectrum*, Vol. 23, N. 4, April 1990.

Ma soprattutto, siccome non riuscivo a capire una cosa relativa ai rapporti di durata di cui ho parlato, ieri ho scritto a Laurer per chiedergli lumi. E lui, gentilissimo, mi ha risposto immediatamente e poi mi ha suggerito di leggere il suo libro autobiografico:

*I have added Appendix "D" to my autobiography describing the development of the U.P.C. The book is "Engineering Was Fun" and can be downloaded for a few bucks or printed for quite a bit more. At [www.lulu.com](http://www.lulu.com) search "laurer".*

*That will give the details as to how the code is decoded.*

*George*

Detto fatto: lo ho comprato immediatamente per la favolosa cifra di \$3.75 (pagabili con Paypal) e sono diventato padrone di un gustosissimo libro che racconta le sue avventure in quell'epoca pionieristica, oltre a un documento tecnico di 30 pagine che spiega per benino tutta la faccenda dell'UPC. Ti consiglio caldissimamente di comprarlo anche tu.

Un'ultima cosa: non ne avevo mai avuto dubbi, ma dalla gentilezza e da alcuni particolari del suo sito, direi che Laurer tutto può essere, tranne che un satanista...