



Kirkegaard Winkler (Kirkegaard)

A SILICON BUG

6 November 2011

Al mondo non ci sono che due modi per fare carriera: o grazie alla propria ingegnosit  o grazie all'imbecillit  altrui. (Jean de La Bruy re, I caratteri, 1688)

Si supponga che, nel futuro, si possa costruire un computer che si comporti come se capisse il cinese.

In altre parole, il computer prenderebbe dei simboli cinesi in ingresso, eseguirebbe un programma e produrrebbe altri simboli cinesi in uscita. Si supponga che il comportamento di questo computer sia cos  convincente da poter facilmente superare il **test di Turing**.

In altre parole, il computer possa convincere un uomo che parla correttamente cinese (per esempio un cinese) di parlare con un altro uomo che parla correttamente cinese, mentre in realt  sta parlando con un calcolatore. A tutte le domande dell'umano il computer risponderebbe appropriatamente, in modo che l'umano si convinca di parlare con un altro umano che parla correttamente cinese. I sostenitori dell'**intelligenza artificiale forte** concludono che il computer

capisce la lingua cinese, come farebbe una persona, in quanto non c'  nessuna differenza tra il comportamento della macchina e di un uomo che conosce il cinese. Questa   la "famosa" **ipotesi della stanza cinese** di Searle ma veniamo alla nostra storia e al pi  terrificante e devastante bug che un firmwareista pu  incontrare, il **silicon bug**....

Peter Solonico era il buyer della azienda dove lavoravo e il suo ufficio aveva posto vicino l'amministrazione. Ricevette una telefonata da Simon che invece era il responsabile commerciale nella nostra zona della ditta fornitrice di componenti elettronici. La chiamata di routine era stata fatta per alcuni aggiornamenti che Simon aveva ricevuto qualche giorno prima ad un meeting aziendale. Parlarono di molte cose, come gran parte delle telefonate di quel tipo, e quindi parlarono della evidente crisi, dei mercati finanziari, di come le aziende non fanno pi  investimenti, un po' di retorica dei disastri e fra le varie lamentele anche del micro usato negli ultimi progetti aziendali . Il microprocessore era un Texas TMS430F149 un micro molto interessante e particolarmente performante nelle applicazioni a basso consumo energetico.

“Ti dicevo che abbiamo un aggiornamento per quel micro. C’è infatti il modello successivo il TMS430F249 che potete tranquillamente sostituire nella vostra produzione corrente. Costa un euro di meno del precedente, consuma la metà... così è perfetto per le vostre applicazioni low power ”

“mmm ... Ma sei sicuro che non ci sono problemi, e posso sostituirlo ?”

“Vai tranquillo il nostro FAE ha detto che sono equivalenti sia hardware che software”

“Va bene mi hai convinto ... allora mandamene 170 pezzi così il prossimo lotto lo facciamo con questo micro”

“Perfetto. Te li mando subito...”

“Benissimo .. allora ci sentiamo a presto”

“Ciao..”

“Ciao..”

Arrivai in sede piuttosto trafelato e affaticato dalla lunga settimana di lavoro che finalmente sembrava finire. Gli uffici dove lavoravo, questa volta, erano dei luminosissimi open space con soffitti molto alti e ampie finestre addobbate da monotoniche veneziane che celavano la vista ad un angosciante parcheggio esterno. Le postazioni di lavoro erano suddivise da separatori che erano poco più alti delle scrivanie, poco colorati. A fianco a me c'era Sasha che era l'elettronico masterista addetto alla progettazione di tutti i disegni sia meccanici che elettrici, assemblava i prototipi, eseguiva test preliminari, fumava sigari toscani, e le solite malelingue sostenevano che aveva anche una omosessualità pesantemente occultata.

Arrivai con qualche minuto di ritardo e subito accesi il pc mettendo a posto il perenne disordine della scrivania, quando subito, mi accorsi della presenza inquietante di Malcom che era il terribile responsabile della produzione. Malcom era tanto brutto quanto odioso ma altremodo indispensabile per la vita aziendale; constatare che era odioso al sottoscritto era davvero di poco conto e anzi per molti colleghi diventava un pregio. Con la coda dell'occhio vidi che Malcom arrivava con il solito passo trafelato, passo che tipicamente adottava quando si creavano situazioni critiche che figurativamente rappresentavamo con una goffa pantomima facendo l'aereoplanino come se fossimo dei cetrioli volanti ... a bassa quota.

“Kirk senti ... ti ricordi quando l'altra settimana abbiamo comprato il programmatore della Texas.”

“Si ...” ma durante i puntini di sospensione credo di aver assunto una espressione inequivocabilmente inquietante che era una via di mezzo fra l’angoscia kierkegaardiana e la dolorosa sofferenza di un dito schiacciato da una sportellata.

“Quando hai finito con le faccette da ebete, considera anche che ieri sera abbiamo iniziato la programmazione delle schede ma il micro ... non si è nemmeno acceso. Non va.... ” una pausa di riflessione dove uno volentieri si sparerebbe sui maroni e poi la conclusione scontata dell’epitaffio “...potresti darci un occhiata ?”

Un occhiata ? Perché non gliela dai te l'occhiata sacco di m...

Mi concentrai per cercare di occultare quello che pensavo poi alla fine come sempre cedetti “Si dammi un attimo che mi riprendo dallo shock per averti visto così presto.... arrivo subito”

Parlando di cose serie come dicevo il Texas MSP430 e' un micro particolarmente interessante. Si distingue per essere un classico Von Neumann ma a 16 bit che però essendo di nuova generazione ne hanno reso l'architettura interna sotto molti aspetti particolarmente efficiente. È un 16 bit nativo e quindi la gestione dei registri è stata tutta progettata per lavorare con questa struttura dati. Il micro è un **RISC** con un ciclo macchina da 125 ns che garantisce efficienza e versatilità, in particolar modo nella gestione dei registri interni e del salvataggio su ram; per i dettagli potete vedere il seguente [link](#). Abbiamo un convertitore a 12 bit, due timer a 16 bit con una decina di configurazioni caputure/compare. E' anche un lowpower con diverse configurazioni per la gestione del basso consumo. In standby con risveglio da interrupt consuma meno di 2 uA. (avvicinandoci molto alle correnti di scarica di una batteria...).

Un elemento fondamentale nello sviluppo di un firmware è quello di avere a disposizione nel codice una struttura dati che ci possa indicare il trascorrere del tempo. Questa struttura dati è, in realtà, semplicemnte una serie di registri che vengono aggiornati attraverso un interrupt di un timer interno del nostro uC. Per la gestione dei timer di sistema si può implementare un routine assembly tipo quella sottostante. La routine è tutto sommato molto semplice grazie anche agli opcode RISC che il micro mette a disposizione.

```
#include "msp430x14x.h"
```

```
#define MAX_TIMERS    8
```

```
rseg CODE

PUBLIC __timer_handler

EXTERN glbTimers

__timer_handler:

    ; push.w R2
    push.w R4
    push.w R5
    mov.b #MAX_TIMERS,R4
    mov.w #glbTimers,R5
Loop:
    cmp.w #0,0(R5)
    jz IsZero
    dec.w 0(R5)
IsZero:
    add #2,R5
    dec.b r4
    jnz Loop
    pop.w R5
    pop.w R4
    ; pop.w R2
    ret
end
```

Se questa routine la associamo ad un interrupt di un timer interno che viene richiamato a cadenza costante ecco pronta una struttura dati che ci permetterà di verificare il trascorrere del tempo; ma ora descriviamo velocemente il nostro progetto.

Questa volta la ditta cliente era l'[Italgas](#) e l'applicazione era per la loro rete di distribuzione nazionale. Il piccolo sistemino veniva e viene utilizzato per la lettura delle correnti catodiche dei metanodotti. Tento di descriverlo meglio. Il gas di Putin arriva in Italia attraverso condutture da 50 / 60 bar di pressione. Tipicamente nei pressi di grandi svincoli di distribuzione del gas e in mezzo a sperdute campagne, ci sono delle cabine chiamate di **primo salto** che diminuiscono drasticamente la

pressione del gas dai 50/60 bar ad una decina di bar. Queste condutture raggiungono poi le periferie cittadine dove ci saranno altre cabine dette di **secondo salto** che porteranno il gas sino alla pressione di qualche bar (4.5 tipico). Queste poi raggiungeranno le ultime cabine di **terzo salto** dove il gas viene portato alla pressione di esercizio cittadino che è di qualche millibar (tipicamente 20). Per ulteriori dettagli la pagina di wikip [link](#)

Dopo questa velocissima descrizione adesso addentriamoci in un dettaglio che ci interessa. Tutte queste condutture soffrono di un problema peculiare che è tipico di questi sistemi. Un tubo metallico, a contatto con il terreno si deteriora corrodendosi. Questa fenomeno fisico di corrosione non è quello che conosciamo di ossidazione del metallo chiamato ruggine; questa corrosione è causata invece dalla meno ovvia differenza di potenziale che nasce fra la struttura metallica e il terreno circostante. Questa differenza di potenziale crea le correnti catodiche. Il problema è globalmente riconosciuto ed è stata sviluppata tutta una tecnologia che si chiama sistema di [protezione catodica](#). Il continuo monitoraggio di queste correnti consente di prendere delle contromisure che attenuano e rallentano i pericolosi effetti di corrosione che se degenerassero incontrollabilmente potrebbero provocare pericolosissimi fori nella tubazione. Queste contromisure “semplificando” consistono nel rendere **immune** il metallo della tubazione cercando di annullare queste corrosive correnti. L'operazione di immunizzazione viene resa possibile “alimentando” la tratta con potenti alimentatori che mettendo sotto tensione il tubo creano correnti inverse che annullano quelle rilevate... ma ci sono altre tecniche. Questo è il [corso](#) (a pagamento ?) Italgas dove potrete vedere probabilemnte anche i nostri apparati...

Il progetto firmware non era il mio ma lo avevo ereditato dal mio predecessore. Premesso che era ed è un amico, questo fatto dell'eredità del firmware rendeva comunque le cose più complicate, perchè progetti di questo tipo sono particolarmente sofisticati dal punto di vista ingegneristico e il mio predecessore non era niente male come firmwarista; ma lo capiremo meglio fra un po'....

Raggiunsi la produzione come un cammello che sta per entrare nel deserto. Ora pensavo che la volta precedente avevamo scoperto che il programmatore non supportava il nuovo modello di microcontrollore, ma questa volta era chiaro che c'era qualcos'altro. Verificai che effettivamente tutta la procedura di programmazione era stata effettuata correttamente. Dopo pochi minuti feci una telefonata al Chiatti che era il FAE italiano per quella famiglia di microcontrollori. "No Kirk avevamo già parlato di questo.. ."

"Si... ma non con me..."

"Il progetto lo devi ricompilare però non basta ricompilarlo... ci sono delle piccole modifiche da fare... "

"si ma perché non ne parliamo prima... e poi piccole... ho già capito tutto...."

"ti mando il pdf..."

"si ... manda manda... il pdf..."

e da quel momento niente fu come prima...

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Kirkegaard:a-silicon-bug>"