

KIRKEGAARD WINKLER
L'enigma dell'ora esatta

- Rumori dalla terra -



ebook speciale a tutti gli Amici di Electroyou.

Carlomariamamenti, Paolino, Tuxology, Tardo-Freak, come non ricordare Attilio, Berello, il supporto di Carloc (ma non dovevi passare da queste parti ;) , Demos81, e poi ancora GrandeGiove, Lettrico, Oberon, Scorpion92, IsidoroKz, Guerra, GuidoB, Clavicordo, VRI, DarwiNe, FrancescoFP, Marc96, Dimaios, Demos81, morettigiorgio per ultimo ma non per importanza admin Zeno.

Un abbraccio a tutti...

La mela mancante

La logica booleana (George Boole 1815 - 1864) è la formalizzazione matematica della logica Aristotelica ed è divenuta la base della programmazione dei sistemi embedded e di tutti i computer in generale . Il presupposto fondamentale è che esistano solo due livelli di verità: 1 e 0.

Un computer, dal punto di vista concettuale, può essere progettato su qualsiasi rappresentazione di dato. Storicamente però i vantaggi dati dalla memorizzazione e riproduzione delle memorie bistabili hanno fatto in modo che i computer con questa tecnologia potessero sfruttare velocemente le congiunzioni logiche dell'algebra booleana. Questa base fu il primo immobile che avrebbe permesso quell'evoluzione disorganica della tecnologia digitale che stiamo vivendo. Ma la logica come i computer ha i suoi bachi e questa sarà una piccola storia di chi un giorno ha tentato di risolverli...

3-09-1995 *Milano Centro direzionale Colleoni*
Meeting giornaliero Microchip.

I **meeting** della Microchip erano diciamo quelli un po' più blasonati rispetto alle case concorrenti. Il successo planetario del microcontrollore PIC nei sistemi embedded era diventata una certezza e molti volevano partecipare, per acquisire segreti che non esistevano. Rag-

giunsi il palazzone da lontano piuttosto anonimo, poi la hall e il suo via vai senza fretta di Field Application, Vertical Segment Manager e altri variegati portaborse che parodiavano amministratori delegati di ditte fantasma.

Arrivai con mezzora di anticipo e la segretaria discretoccia mi riempì con le solite brochures e con i soliti gadget previsti per la presentazione. Chiedo per un bagno, il viaggio è stato un po' lungo e la tangenziale come solito mi ha messo alla prova. Nel frattempo arrivò altra gente che si dispose disseminandosi in gruppi omogenei per tutta la sala. Mi sedetti fra le ultime file e un po' alla volta intorno a me si formò un gruppo altrettanto omogeneo di single.

Alzai il leggio della sedia e aprii il sacchetto, tirando fuori il contenuto, intanto arrivò un "Field non so che" e iniziò la presentazione dei prodotti. Intanto guardavo il contenuto ed erano un po' di ciclostilati sulle nuove famiglie e poi, anonimo, c'era un foglio di presenatazione per un nuovo tools di sviluppo. Il fuzzyTech.

Guardai un po' incuriosito e sfogliando fra le pagine, notai subito i riferimenti di un **linguaggio naturale** che poteva essere utilizzato nei nostri sistemi embedded. Tutta la giornata la passai fra variabili fuzzificate, funzioni di appartenenza, grafici, distraendomi così spesso dalla presentazione fastidiosamente noiosa. Il mese dopo ne parlai, in laboratorio con il mio capo. Se ne discusse così un po', a grandi linee, ma alla fine ci convincemmo e titubanti iniziammo lo sviluppo di un prototipo.

Nell'azienda dove lavoravo, producevamo elettronica nel settore consumer e automotive; i clienti erano la

Piaggio, San Giorgio, Ferroli, York International, Olimpia, Ariete, De Longhi e così via. Avevamo come clienti un piccolo spaccato della traballante industria del tempo alle prese con il difficile mercato globale e con i cinesi già alle porte che da lì a poco avrebbero rimescolato tutte le carte. Tutte le volte che gli amministratori si incontravano si abbracciavano e si dicevano "anche quest'anno ce l'abbiamo fatta " (we did it!). Ora i problemi erano così tanti che venivano prese in considerazione tutte le soluzioni alternative e quella Fuzzy era una di quelle.

I precursori del Fuzzy

La storia moderna della logica fuzzy nasce quando si mise in **discussione** i fondamenti della logica convenzionale. Il primo esempio storico è dato dal matematico Russel quando nel 1901 in piena belle epoque enunciò il paradosso del barbiere. Paradosso che per i puristi in realtà è una **antinomia** ma che, comunque la si chiami, mise in crisi le basi scientifiche del tempo o meglio dire mise in crisi il significato del vero scientifico. Il paradosso si enuncia così:

« In un villaggio vi è un solo barbiere, un uomo ben sbarbato, che rade tutti e soli gli uomini del villaggio che non si radono da soli. Il barbiere rade se stesso[1]? »

Una delle più celebri antinomie è presente anche nella bibbia e viene definita con il paradosso del mentitore. Nella lettera a Tito 1:12-13 l'apostolo Paolo nel I secolo DC cade in questo paradosso quando appunto scriven-

do al discepolo Tito disse:

“Uno di loro, proprio un loro profeta, ha detto: «I Cretesi sono sempre bugiardi, male bestie, ventri pigri». Questa testimonianza è vera, per questo motivo riprendili severamente, affinché siano sani nella fede”

Se il Cretese dice che i Cretesi sono sempre bugiardi dice la verità o dice una bugia ? Questo paradosso però non è "originale" del biblico santo, ma ha origini ancora più lontane e proviene dal filosofo Epimenide del VII secolo AC.

Ma ritorniamo al problema di Russel. Con il suo paradosso, abbiamo detto, mise in crisi tutto il sistema logico e matematico del tempo. C'era chi cercò di arginare la pericolosa falla che si era creata e che stava per portare via tutte le basi della logica su cui si posava il progresso scientifico. I grandi filosofi e matematici del tempo come Whithead, Luitzen Brouwer, David Hilbert ed infine Russel stesso, cercarono teorie alternative che riuscissero a giustificare in modo convincente il paradosso come soluzione dei paradigmi della logica.

Non ci fu nulla da fare, alla fine tutti gli sforzi naufragarono con Kurt Gödel e il suo teorema dell'incompletezza del 1931 dove dimostrò che all'interno di una teoria matematica esisteva almeno un assioma che non era in grado di essere dimostrato. Con il crollo del vero scientifico venne alla ribalta il vero trascendente e Godel stesso si distinse per la famosa prova ontologica sull'esistenza di Dio che non pubblicò mai e wikipedia sostiene che non l'ha fatto per non essere frainteso (mi sono

chiesto da chi: da Dio, dall'umanità, dai colleghi atei ?).

Il primo teorema dell'incompletezza di Godel sostiene

In ogni teoria matematica T sufficientemente espressiva da contenere l'aritmetica, esiste una formula tale che, se T è coerente, allora né la sua verità né la sua negazione sono dimostrabili in T .

La logica convenzionale non permette l'esistenza del **terzo escluso** (*Tertium non datur* a chi piacciono i latinismi) e cioè , in parole povere, i presupposti di tutti gli assiomi enunciati o sono veri o sono falsi e sono impossibili transizioni intermedie. Ma secondo Godel questo non è dimostrabile e così si dovevano cercare alternative che andassero oltre la logica del complemento fra 2 stati.

A ben vedere però si era già parlato di implementazioni di questo tipo chiamate **logiche polivalenti** che stavano piano piano prendendo piede.

Questi nuovi sviluppi erano leggermente antecedenti rispetto a Godel ed erano stati sviluppati da Jan Łukasiewicz nel 1920. Al concetto di vero e falso il filosofo e matematico polacco aggiunse il poetico terzo escluso chiamato "problematico". Comunque sia, Jan ha spianato la via ad un matematico sistemista per giunta ingegnere di origini iraniane chiamato Lotfi Zadeh, che ha trovato nelle logiche polivalenti un valido supporto su cui concretizzare la sua rivoluzionaria idea chiamata "logica Fuzzy".

Il nome Fuzzy si è indissolubilmente legato al suo autore. Pare che il suo percorso sia anche dovuto ai suoi studi pregressi relativi alla **teoria dei sistemi**. Questa è anch'essa una cosmogonica ipotesi che abbraccia tutto lo scibile umano. Un sistema nella sua definizione potrebbe essere un gruppo di persone che lavorano in una azienda, oppure un gruppo di atomi di giunzione di un mosfet o anche le persone di un sistema politico che governano in un paese democratico. La teoria dei sistemi è ad esempio la base analitica con cui si risolvono tutti i loop di controllo con PID.

Da questa teoria Zadeh ha cercato di modellizzare il concetto di **“grado di appartenenza”**. Fino a quel momento tutto il mondo occidentale aveva considerato i sistemi come insiemi di elementi dicotomici. Tutte le parti erano progettate in modo tale da considerarle come macchine a stati sequenziali che facevano corrispondere a noti sistemi di segnali in ingresso altrettanto noti sistemi di segnali in uscite. La relazione di causa effetto era evidente ma non bastava più. Con il grado di appartenenza si eliminò l'ideale gradino di passaggio fra lo stato di vero e lo stato di falso per implementare una funzione che tipicamente è lineare ma potrebbe essere benissimo una curva di qualsiasi tipo in grado di unire i due stati.

Per spiegare meglio questo concetto prendo la definizione di logica Fuzzy dalla Treccani che non è proprio il massimo però ci accontentiamo...

La logica fuzzy ha l'obiettivo di rappresentare il più fedelmente possibile, su sistemi a logica binaria, il modo

di ragionare tipico della mente umana. Il ragionamento umano ammette, infatti, la possibilità di passare da un concetto a un altro gradualmente, mentre ciò non è possibile nella logica booleana, in cui la definizione stessa di insieme ordinario (ordinary set) comporta che l'appartenenza di un elemento a un insieme corrisponda a un concetto binario, potendo assumere i due soli valori $\{1,0\}$.

Un piccolo accenno di teoria

Nel 1965 Zadeh sviluppò le basi della teoria che pubblicò su «Information and Control» con il titolo Fuzzy Sets. La logica fuzzy è una estensione della logica booleana; ora prendiamo la definizione ufficiale di Zadeh

Let X be a space of points, with a generic element of X denoted by x .

Thus $X = \{x\}$. A fuzzy set A in X is characterized by a membership function $f_A(x)$ which associates with each point in X a real number in the interval $[0,1]$, with the values of $f_A(x)$ at x representing the "grade of membership" of x in A .

Thus, the nearer the value of $f_A(x)$ to unity, the higher the grade of membership of x in A .

Si definisca un insieme fuzzy A in x a cui associare il grado di appartenenza. Dove X viene definito all'interno dell'intervallo reale $U=[0.0,1.0]$. In senso stretto un insieme Fuzzy è composto da ordinate coppie di valori

$$A = (x, \mu(x)) \parallel x \in U$$

se $\mu(x)$ si avvicina a 1.0, il "grado di appartenenza" di x in A aumenta. Come abbiamo già detto, questa funzio-

ne in genere è lineare, ma volendo complicarci la vita possiamo utilizzare anche curve alternative.

Nel fuzzy vengono implementate le **variabili linguistiche** che sono un po' la chiave della logica. Queste variabili linguistiche sono a loro volta associate a delle variabili fuzzy che infine sono associate a delle variabili di base. Le variabili linguistiche sono quelle che descrivono le sensazioni Oggettive/Soggettive umane. Caldo, Freddo, Vecchio, Giovane, Molto alto, Abbastanza basso... e così via.

Le operazioni fuzzy richiamano quelle della logica booleana, ma l'implementazione è diversa. Definiamo A e B come due insiemi Fuzzy.

L'operando OR o di unione

$$A \cup B \Rightarrow \mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

Questo operando in logica booleana è equivalente alla somma degli operandi

Operando AND o di intersezione

$$A \cap B \Rightarrow \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

Operando NOT o negazione

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

L'alternativa

La logica convenzionale non è in grado di giustificare un evento che sia vero e falso nello stesso momento, però è in grado di giustificarlo quando, questo evento, sia vero o falso in momenti diversi. Questa miracolosa scienza viene chiamata **teoria della probabilità**. Gli eventi bivalenti, vengono messi all'interno di uno spazio campionario e prendono un valore che è molto simile al grado di appartenenza di una variabile fuzzy, idealizzato da Zadeh. Ma nella definizione di queste operazioni booleane si evidenziano le differenze più importanti.

Uno degli assiomi nella teoria della probabilità dice che se ho un insieme di due eventi incompatibili, la probabilità che accada un evento dell'insieme è pari alla somma delle proprietà degli eventi.

$$P(A \cup B) = \frac{n_{A \cup B}}{n} = \frac{n_A}{n} + \frac{n_B}{n}$$

Nella logica fuzzy, come abbiamo visto pocanzi, se ho un insieme di due variabili fuzzy, il grado di appartenenza dell'insieme fuzzy è pari al massimo grado delle funzioni di appartenenza delle variabili.

Un'altra importante differenza fra la **logica fuzzy** e la probabilità è che la probabilità fa riferimento a eventi che devono accadere mentre la logica fuzzy a quelli che stanno accadendo.

Parlando dei limiti del pensiero umano faccio un piccolo excursus storico. Nell'antico Israele (X secolo AC) c'era la consapevolezza che la sola logica umana non sarebbe stata in grado di arrivare alla verità in tutti gli affari umani. Queste difficoltà divennero evidenti specialmente nei casi di giudizio. Alcune cause che venivano portate a giudizio dal giudice e sommo sacerdote non avevano soluzioni possibili.

Partendo dall'assunto che una **non soluzione** è peggio di una **soluzione sbagliata**, per divina provvidenza inserirono nella legislazione un metodo che gli avrebbe tirato di impiccio nei casi più difficili) Questo metodo era chiamato degli Urim e i Tummim.

Non si sa con precisione cosa fossero, si sa che erano celate all'interno dell'Efod, un prezioso pettorale che faceva parte del cerimoniale nel tempio, indossato dal sommo sacerdote e utilizzati quando era impossibile riuscire a dare un giudizio equo. Gli Urim e Tummim erano probabilmente pietre colorate, solo da un lato, che venivano lanciate lasciando con quel gesto il giudizio a Yahweh (si noti quale importanza aveva assunto la probabilità del lancio che avrebbe dato la soluzione dell'evento contestato.

Un po' più difficoltosa era invece la "scelta a Dio" effettuata ad esempio dai Longobardi. Questi infatti nel caso in cui ci fosse l'impossibilità di dimostrare la colpa, il presunto reo veniva assolto con l'ordalia. Il poveretto, che magari era lui a non poter dimostrare la sua innocenza, veniva costretto a prendere con le mani nude una spada incandescente o un altro ferro rovente lasciando a Dio l'incombenza di proteggerlo dal dolore e dall'ingi-

stizia.

Il principio fuzzy

Riporto il testo preso da Web di Antonella Giulia Pizzaleo che ha scritto nel 2004 un libro dedicato a questa tecnologia, intitolato banalmente "Fuzzy Logic".

"Prendiamo in mano una mela. È una mela? Sì. L'oggetto nelle nostre mani appartiene agli agglomerati spazio-temporali che chiamiamo l'insieme delle mele, di tutte le mele di ogni tempo e luogo.

Ora stacciamone un boccone, mastichiamolo e inghiottiamolo. Lasciamo che il nostro apparato digerente separi le molecole della mela." L'oggetto che abbiamo in mano è ancora una mela?

O no? Stacciamone un altro boccone. L'oggetto è ancora una mela? Diamo ancora un altro morso, e così via fino a finirla. La mela è mutata da una cosa in una non-cosa, in nulla. Ma dove ha oltrepassato la linea di demarcazione fra mela e non-mela?

Con questa metaforica descrizione Bart Kosko introduce nel suo libro la logica fuzzy. Gli oggetti che ci circondano, ma anche i nostri corpi e gli eventi che viviamo, mutano in maniera fluida e disinvolta, impedendoci di caratterizzarli in modo rigido, dicotomico.

Antonella fa un riferimento esplicito alla citazione di Bart Kosko che è uno dei più celebri discepoli di Zadeh probabilmente citando il suo libro "Il Fuzzy pensiero". L'argomento è relativo al concetto di mela e non mela che tutti citano pedissequamente spesso senza nominare l'autore. Comunque sia ,tutte le volte che mi presentano

questo esempio, se faccio riferimento al linguaggio naturale utilizzato, mi si pongono molte perplessità, tanto da chiedermi se ci fossero delle differenze indotte, magari dalla traduzione in Inglese, che potevano variare il senso dell'insieme mela e del suo significato semiologico.

Dopo averne staccato un boccone abbiamo ancora una mela ? Io direi di no. Diciamo che sopra un tavolo ho accostate una mela intera e 3 mele morsicate. Se chiedo a voi di portarmi una mela che tipo di scelta farete? Per di più è evidente che la demarcazione fra mela e non mela non è data solo dalla quantità mancante di mela ma anche dalla qualità di mela che voglio ottenere.

Se sopra il tavolo metto accostate una mela integra e sana e 3 mele non morsicate ma palesemente ammaccate, marcite o lesionate quale scelta farete ? Nella definizione di una mela c'è quindi qualcosa di più di mela completa. Il senso di melinità che ho è che voi inferite non mi sembra così fuzzy come l'esempio vuol fare intendere.

Quello che si crede

Riprendo la citazione dal Treccani

...nella logica booleana, in cui la definizione stessa di insieme ordinario (ordinary set), comporta che l'appartenenza di un elemento a un insieme corrisponda a un concetto binario, potendo assumere i due soli valori

$\{1,0\}$.

Questo è il classico esempio dove la logica convenzionale o del terzo escluso sembrano inadeguati, infatti il **limite** della logica booleana è quello di non poter descrivere, valori intermedi agli stati. Ma anche qui ho delle perplessità. Il concetto di vero e falso è un presupposto della logica e del pensiero universale (e non solo occidentale) e il problema di includere i passaggi intermedi fra due stati in antitesi c'è sempre stato.

Fra uomo e donna c'è un universo di gay, così come fra il bene e il male c'è un mare di stupidità od anche fra il bianco e il nero tanti livelli di grigio (o forse meglio un arcobaleno di colori..) Il concetto di mela con racchiuse tutte le caratteristiche semiologiche del termine è un concetto primordiale, basti pensare ad Adamo ed Eva e il morso fatale, o anche ad esempio, il quantomeno maledetto morso di Biancaneve sulla mela donata dalla strega cattiva. La mela morsicata divenne invece l'ispirazione vincente di Steve Jobs quando creò il logo della Apple includendo su di esso tutti i significati subliminali che il logo richiama e che in parte abbiamo citato.

Prendo ancora una citazione di Antonella ripresa sempre da una sintesi su web del suo libro.

“A proposito delle applicazioni fuzzy Cammarata ha scritto:

Con l'inserimento della logica fuzzy nei normali prodotti industriali si possono ottenere, a costi relativamente

bassi, molti vantaggi, come un loro maggiore quoziente d'intelligenza, uno sviluppo più rapido dei prodotti stessi (con minimo time to market), una maggiore affidabilità di funzionamento, una minore necessità di manutenzione, eventuali capacità autodiagnostiche, minore inquinamento e minor consumo di energia.”

Silvio Cammarata è un po' il guru italiano della tecnologia Fuzzy e delle applicazioni di **reti neurali**. Ora la citazione di una citazione è notoriamente una cosa indelicata perché si decontestualizza ciò che è già decontestualizzato, ma la specificità dell'argomento mi permette di prendere in considerazione un aspetto.

Infatti in questo caso siamo molto più vicini ad un partito preso che non a un rigoroso assioma scientifico ed è comico notare come essa stessa abbia l'indiscutibile proprietà di essere molto fumosa. “Costi relativamente bassi”, “molti vantaggi” “più rapido” sembra davvero la pubblicità di un'automobile.

Senza considerare poi uno spiccato senso esoterico dei sistemi embedded che grazie al Fuzzy riescono ad aumentarne il “quoziente di intelligenza”.

Associare il significato di intelligenza ad un elettrodomestico è un tentativo di illudere il consumatore a credere che il Fuzzy sia in grado di dare vita e coscienza agli oggetti così come faceva l'equivalenete apprendista stregone di disneyana memoria e che diventò il vero sogno di tutte le massaie. Il QIM esiste davvero, ma le applicazioni sono di intelligenza artificiale vera e non spe-

culative selvagge operazioni di marketing. Sono convinto che se dovessi applicare il QIM ai miei sistemi embedded questo bene che vada potrebbe raggiungere un punteggio di qualche punto superiore ad un sasso, forse col fuzzy diverrebbe come quello di una patata ma non molto di più.

In un altro sito web di tipo ingegneristico un blogger, dopo aver spiegato sommariamente alcuni esempi di logica fuzzy, si è posto la domanda chiave “ma a cosa serve ?” e così si è dato la risposta:

“A qualcosa servono, visto che se ne parla tanto, ed e' nato un mercato di sistemi fuzzy.”

mi astengo da ulteriori commenti perchè mi sembra abbondantemente autoesplicante.

Prendiamo ancora Allen Bonde dove nelle sue pagine web scrive:

Fuzzy rule-based systems apply these methods to solve many types of "real-world" problems, especially where a system is difficult to model, is controlled by a human operator or expert, or where ambiguity or vagueness is common.

Per i non anglofoni la citazione in sintesi dice che con Regole Fuzzy possono risolvere molti tipi di problemi del mondo reale specialmente in sistemi **difficili** da mo-

delizzare. Tralasciamo il fatto che ovviamente tutti i sistemi embedded risolvono problemi del mondo reale e per quello trascendente abbiamo ancora molta strada da percorrere. Ora però mi chiedo: se un sistema è difficile da modellizzare per i numerosi stati di input e magari i molteplici attuatori di uscita, come può un sistema Fuzzy renderlo più semplice ? Con l'inclusione del terzo escluso ?

Esempio Embedded

Siccome siamo in un forum ingegneristico facciamo l'esempio di una semplice applicazione embedded. Diciamo che devo fare un termostato e devo controllare la temperatura di un frigo. Secondo la logica booleana per implementare un termostato deve definire una temperatura di setpoint che il mio termostato dovrà regolare. Siccome sono un sistemista embedded vi aggiunto mezzo grado di isteresi per evitare condizioni di isterismo del relè. La logica quindi diventerà in un ipotetico linguaggio pseudo informatico:

```
T_SETPOINT = 8
```

```
T_ISTERESI = 0.5
```

```
se Tmis < T_SETPOINT allora
```

```
    Out_Compressore = ON
```

```
se Tmis > T_SETPOINT + T_ISTERESI allora
```

```
    Out_Compressore = OFF
```

Se dovessi farlo in logica fuzzy sostituisco la misura di

temperatura con una serie di variabili linguistiche fuzzy , in grado di descrivere la sensazione di temperatura. In caso di un sistema di questo tipo potrei dichiararle così:

Molto Caldo $+FS^{\circ}$ a 12°

Caldo 13° a 8°

Freddo 9° a 3°

Molto Freddo 4° a $-FS^{\circ}$

Innanzitutto come avete potuto notare la sensazione di freddo ovviamente non è la mia ma è del sistema. Le funzioni che esprimono il grado di appartenenza potrebbero essere ad esempio aree trapezoidali con lati di ampiezza un grado. Il grado di appartenenza, ad esempio della variabile Molto Caldo, potrebbe valere "0.0" a 12 gradi e "1.0" da 13 in poi. Da questo si deduce che il grado di appartenenza a 12.5 sarà pari a "0.5" perchè la funzione è di tipo lineare. Dopo aver dichiarato le variabili di ingresso bisogna dichiarare quelle di uscita che descrivano lo stato dell'attuatore.

Compressore_Activo

Out_Compressore = ON

Compressore_Disattivo

Out_Compressore = OFF

Costruisco quindi la regola fuzzy:

Se è Caldo e Molto Caldo allora
Compressore_Activo

Se è Freddo e Molto Freddo allora
Compressore_Disattivo

Ora mi chiedo è cambiato qualcosa ? Così a vista mi sembra di no... Ahhh si dimenticavo ... è vero. Qualcosa è cambiato. Le 4 istruzioni per eseguire il controllo sono state sostituite dai 100K di codice necessario ad implementare il motore fuzzy. (NDR: 100 k ovviamente sono una iperbole...).

L'applicazione embedded di un termostato è semplice... ma non così semplice. Tutti i sistemi embedded (compresi quindi anche quelli sviluppati con la logica Fuzzy) hanno bisogno di una parte oscura di codice che "noi" chiamiamo firmware proprio per distinguerla dal senso comune che viene dato alla parola software.

Questa parte oscura è fatta di impostazioni del microprocessore, della mappatura della memoria, della progettazione di una base tempi interna fondamentale per i sincronismi del sistema, della modellizzazione della sonda di temperatura e delle funzioni di linearizzazione di misura e di filtro per il rumore, di funzioni per il controllo delle uscite e così via. Insomma la parte intelligente di un termostato la possiamo racchiudere in 4 righe

che diventano due col Fuzzy...

Certo alcuni di voi obietteranno considerando che per evitare proprio l'handicap dato dall'includere una libreria fuzzy , molte case hanno pensato bene di sviluppare processori con dentro l'hardware necessario per eseguire le regole fuzzy. Anche in questo caso i problemi rimangono e forse sono anche più gravi, ad esempio la linea di prodotti deve essere garantita per almeno la durata di produzione, in caso di grosse numeri difficilmente i costi possono essere comparabili rispetto ad un micro tradizionale e così via.

Tutta questa fatica e questo rischio progettuale poi per ottenere cosa ? Prestazioni progettuali inferiori e rischi di gestione del progetto superiori ? Sapete già quale sarà la risposta...

Sono nel laboratorio e ho appena installato il software fuzzyTECH. Che bello lavorare con le interfacce grafiche. E poi il diagramma 3D con visualizzato lo stato di tutte le variabili rendeva evidente gli eventuali buchi della logica. Dopo aver ultimato tutte le regole il modulo creava un codice oggetto che poi dovevo linkare nel mio progetto assembly.

Le regole assomigliavano molto al livello di supporto logico che volevamo implementare. Questa era un'interfaccia ideale per i nostri clienti che avrebbero potuto utilizzare autonomamente, implementando così le loro malsane idee su un supporto che poteva diventare un simulatore logico del sistema. A me sarebbe bastato l'onere di includere all'interno del micro questo motore fuzzy. Che dire, inizialmente un sogno, ma poi mi risvegliai.

Una delle caratteristiche del software è che non può disallinearsi dal modello di pensiero del programmatore. C'è un'importante legge della programmazione di Turinghiana memoria che in sintesi sostiene questo fatto: se sei in grado di descrivere la soluzione di un problema in un qualsiasi linguaggio naturale sei in grado anche di descrivere un algoritmo in grado di risolvere il problema. Da questo si deduce, in generale, che più è difficile la descrizione della soluzione è più complicato sarà il nostro algoritmo. Così mi risvegliai dal sogno quando mi accorsi che l'implementazione della logica Fuzzy non mi semplificava le cose ma anzi veicolava pericolosi elementi di instabilità nel micro cosmo embedded dove vivo

In questo altro sito trovato su web si descrivono le proprietà taumaturgiche del Fuzzy :

"FL was conceived as a better method for sorting and handling data but has proven to be an excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning."

In questo caso non posso che ammettere dei limiti per-

chè davvero questa proprietà di ordinamento davvero mi è sfuggita. Ma comunque sono ancora evidenti declamate proprietà fumose di robustezza che permettono addirittura di perdonare errori all'operatore. Ora se riprendo l'esempio del precedente termostato, visto che l'operatore vedrà solo variabili Fuzzy come può perdonarne gli errori ? Alla fine di questo commento, si raggiunge il massimo, quando sostanzialmente si dice che un sistema Fuzzy funziona alla **prima** implementazione e senza ritocchi. Devo solo sperare che Mangusta non legga questi articoli altrimenti sono davvero spacciato.

Le critiche feroci

Susan Haack un acclamato dottore di filosofia dell'università di Miami seguace di Peirce e del pragmatismo universale ha scritto un libello "Do I need fuzzy logic ?" dove sostanzialmente spara ad alzo zero sulla logica creata da Zadeh. In sintesi la Haack parla della logica fuzzy come di una "metodologia molto stravagante e linguisticamente incorretta" per poi dire che al contrario di quello che predica la logica non semplifica le cose ma le complica pesantemente a partire dal come vengono espressi i concetti di vero e falso e delle variabili. Questo un elenco di citazioni dirette che non hanno bisogno di traduzioni.

"Fuzzy theory is wrong, wrong, and pernicious. What we need is more logical thinking, not less. The danger of fuzzy logic is that it will encourage the sort of imprecise thinking that has brought us so much trouble. Fuzzy logic is the cocaine of science."

- Professor William Kahan UC Berkeley

"'Fuzzification' is a kind of scientific permissiveness. It tends to result in socially appealing slogans unaccompanied by the discipline of hard scientific work and patient observation."

- Professor Rudolf Kalman UFlorida

"Fuzziness is probability in disguise. I can design a controller with probability that could do the same thing that you could do with fuzzy logic."

- Professor Myron Tribus, on hearing of the fuzzy-logic control of the Sendai subway system IEEE Institute, may 1988.

Quello che tipicamente si obietta alla logica fuzzy e che pur essendo vera l'idea che esistono livelli intermedi di verità e falsità è altrettanto vero che esistono comunque gli stati di verità e falsità **assoluti** che sono il riferimento di tutti gli assiomi e le ipotesi che facciamo sulla realtà che ci circonda. Questa sorta di distrazione dai concetti di vero e falso alla fin fine anziché irrobustire il pensiero lo indeboliscono rendendo più plusibili le eccezioni che la regola.

Avevo appena finito di mettere su la piccola etichetta rifrangente su un lato della lunga ventola tangenziale. Stavo monitorando la stabilità della velocità ventola con il tachimetro ad infrarossi prestato dalla S.Giorgio per le prove. Allen si avvicinò guardando il condizionatore. Ad un certo punto mi chiese "Ma davvero hai implementato il controllo fuzzy ?" "Controllo... ho messo su la logica per la gestione delle velocità ventola...". Lui con la testa

fece un cenno di consenso.

Rimase un attimo fermo guardando quel condizionatore in formato Monster. Si girò per ritornare al suo posto e disse senza guardarmi "Sai... non sembra...". Non gli risposi e rimanendo lì con la tachimetrica in mano che visualizzava sulle 4 cifre gli RPM del motore pensai alla risposta. "Non sembra ? Certo che non sembra..." Ripensai allo sviluppo alle regole implementate che furono circa una ventina e che facevano tutto quello che promettevano.

La velocità ventola aumentava e diminuiva a seconda delle temperature impostate così come la matrice gli comandava di fare. Tutta l'implementazione mi aveva tenuto impegnato per circa una settimana ma ovviamente il resto del firmware era già stato implementato. I risultati furono buoni, ma alla fine il condizionatore faceva **le stesse** cose di prima. La settimana dopo partì un nuovo progetto importante per una nota ditta nazionale e da lì disinstallai il programma e non ne facemmo più niente...

Ma... e c'è sempre un ma...

Con la Logica Fuzzy non è tutto oro quello che luccica Zadeh lo sa bene. Quello che credo e che in fondo Zadeh abbia ragione. Cioè credo che la dicotomia mutuamente esclusiva tra vero e falso sia discutibile e per questo da tenere conto, ma credo anche che così come è stata dichiarata non possa essere digitalizzata attraverso la logica Fuzzy. Questa logica dovrebbe trovare un senso nello sviluppo del pensiero umano e non pretendere che sia la base algoritmica del mondo digitale.

La logica fuzzy nei sistemi embedded così come sono attualmente sviluppati è naturalmente inefficiente e quindi destinata ad avere solo gruppi di aficionados. Magari può riuscire a cavalcare alcune ondate commerciali che però si esauriranno precocemente per ovvi motivi. Magari con un nuova concezione di microcontrollori, si possono ipotizzare altri scenari.

Parziale bibliografia web

http://it.wikipedia.org/wiki/Logica_fuzzy
<http://www.jimbrule.com/fuzzytutorial.html>
http://www.castelvechchietore.com/spirale/mente_anim_a/estratti/fuzzylogic.html
<http://www.fuzzytech.com/>
<http://www.cs.bilkent.edu.tr/~bulbul/depth/fuzzy.pdf>
<http://www.austinlinks.com/Fuzzy/>
<http://blog.peltarion.com/2006/10/25/fuzzy-math-part-1-the-theory/>
<http://www.calvin.edu/~pribeiro/othrlnks/Fuzzy/home.htm>
http://www.cs.ucsb.edu/~foschini/files/P0406_preprint.pdf
<http://home.howstuffworks.com/rice-cooker2.htm>
http://www.istituto-meme.it/pdf/pubblicazioni/fuzzy_logic.pdf
http://tecalibri.altervista.org/K/KOSKO_fuzzy.htm#p009
http://sipi.usc.edu/~mendel/publications/FLS_Engr_Tutorial_Errata.pdf
<http://www.bu.edu/wcp/Papers/Logi/LogiGrun.htm>

<http://www.ce.unipr.it/people/bianchi/Research/ProgettoFuzzy/home.html>

[http://www.webalice.it/capaso/DOCS/RETI_NEURALI/Logica](http://www.webalice.it/capaso/DOCS/RETI_NEURALI/Logica%20Fuzzy.pdf)

[%20Fuzzy.pdfhttp://www.ildiogene.it/EncyPages/Opere/DeSimoneMatteo-La%20tecnologia%20Fuzzy.doc](http://www.ildiogene.it/EncyPages/Opere/DeSimoneMatteo-La%20tecnologia%20Fuzzy.doc)

Under the Bug

Sovente accade, nella traduzione della lingua anglo sassone, che non esistano parole italiane perfettamente sovrapponibili nel significato. Bug viene tradotto letteralmente con Baco ma questo è sempre un termine all'appannaggio di chi lavora nel settore, infatti come il termine inglese, ne riprende l'eccezione.

Le prime tre righe di wikipedia spiegano dicendo: "Nell'informatica il termine bug o baco identifica un errore nella scrittura di un programma software. Meno comunemente, il termine bug può indicare un difetto di progettazione in un componente hardware, che ne causa un comportamento imprevisto o comunque diverso da quello specificato dal produttore."

Scrivo funzioni in modo che non siano più grandi di 40 di righe, questo mi permette di avere una sezione di codice completa a vista d'occhio.

Quaranta erano le righe che il mio IDE visualizzava (senza spostare la barra laterale) e sono abbastanza per scrivere il 90% delle procedure di un software (ecco .. se siete capaci ditemi come si fa). Con l'avvento di Windows e dei font true type che diedero inizio all'era del **WISIWYG** (acronimo di oltreoceano *What you see is what you get* usato per identificare tipicamente quelle applicazioni che ti permettevano di vedere su schermo quello che poi si sarebbe stampato su carta) uso sempre l'Arial 10 di colore nero con sfondo bianco.

Un tempo il contrasto era sempre rovesciato. I caratteri

di colore verde fosforescenti erano fatti con tubi catodici a fosfori ad alta persistenza. Le frequenze di aggiornamento dei trasformatori di riga erano un po' bassine così pensarono bene di aumentare il tempo di persistenza dei fosfori sul tubo, per diminuire il tremolio dell'immagine. Come dire, parlando metaforicamente, se Maometto non va alla montagna, la montagna va a Maometto (n.b. il proverbio vero ha i termini invertiti...)

Se avevi così l'ardire, di modificare le impostazioni del programma e rovesciare il contrasto, impostando il carattere nero su uno sfondo verde fosforescente, dopo poco, inevitabilmente, subivi pesanti effetti fisici collaterali difficilmente reversibili tipo: arresto psicomotorio, perdita di identità, disturbi bipolari, attacchi epilettici e via di seguito. Per qualche periodo, tentai anche sotto Windows di mantenere questo tipo di contrasto (carattere bianco su fondo nero) ma la debole luminosità del carattere grafico rendeva difficile la visualizzazione del testo e alla lunga ebbi gli stessi effetti di prima (arresto psicomotorio, perdita di identità...).

Mi giocai la carta della pensione per lavoro usurante (alla stessa stregua del lavoro in miniera, catena di montaggio...) ma non ci riuscii e così sono ancora qui a scrivere software...

Un tempo c'era un editor chiamato **Brief**. I poveracci scrivevano il codice con l'edit del dos, chi lavorava su Unix usava il **Vicredendo** di lavorare con una seria applicazione di testo. I programmatori professionisti usavano il Brief. Se non l'avevi lo compravi... se non lo compravi lo copiavi. Aveva le finestre dimensionabili e

potevi nello stesso schermo, vedere più sezioni di codice di diversi moduli contemporaneamente.

Unico nel suo genere aveva la possibilità di selezionare colonne di testo. Diventò uno standard fra i programmatori tanto che, ad esempio, Visual Studio 6 permetteva di impostare l'editor con gli stessi **shortcut key** (combinazioni di tasti) del Brief.

Veniamo al nostro post. Questa volta diversamente dagli altri racconti prenderemo in considerazione tre tipologie di baco software/firmware che possono capitare durante lo sviluppo di un progetto. Questi bachi li chiameremo:

- baco sulla fiducia
- baco presunto
- baco specialistico

Baco sulla fiducia

Franck ... se ti chiedo "gli errori" e tu dici "non funziona", non andiamo molto d'accordo ... (nessuno)

Il progetto embedded è stato sviluppato con un compilatore C della Microchip su un micro st32. Il firmwarista doveva travasare il software (e fare qualche modifica), aggiungendo dei moduli per alcune interfacce hardware che erano state aggiunte. Il modulo software in questione quindi era già provato, testato e perfettamente funzionante. Ad un certo punto durante le fasi di download, del firmware, doveva essere calcolata la checksum. Questo calcolo veniva fatto all'interno di un ciclo for dichiarato così:

```

for (i=0; i<=fwsiz; i+=2)
{
    Crc = chBufferFw[i];
    calcolo..
}

```

Con il travaso sul nuovo hardware, il calcolo della CRC non risultava più corretto. Il baco consisteva in un effetto collaterale ad un problema notorio del C che è l'inesistente controllo del tipo di dato. A seguito di questo mancato controllo il C consente il confronto fra variabili di tipo diverso. Per la precisione erano dichiarate così:

```

int i = 0;
unsigned long fwsiz=0 ;

```

Il ciclo, come si può ben capire, funziona correttamente fintanto che i valori della variabile fwsiz sono inferiori a 32767. I confronti con tipi di dato diversi è uno dei “problemi” impliciti del C. Diciamo che tipicamente un compilatore, un warning su una linea di codice, di questo tipo, dovrebbe sempre darlo. Purtroppo questa cortesia non è accaduta e così lo sventurato ha perso una buona mezza giornata, diagnosticando improbabili problemi sul calcolo CRC, perso fra tabelle checksum e relative criptiche funzioni.

Il round tip di questo bug potrebbe essere quello di acquisire la **notazione ungara**. Questa notazione è una specie di metodo per la stesura del codice. Applicandolo il programmatore si obbliga ad usare una nomenclatura delle variabili un po' più rigorosa. In pratica, questo

metodo consiste nell'aggiungere un suffisso che ne descriva il tipo. Facciamo qualche esempio ripreso da wikipedia:

nSize: numero intero
cApples: conteggio degli elementi
dwLightYears: double Word
bBusy: booleano
fBusy: flag booleano
pFoo: puntatore
szLastName: Stringa terminata con zero
rgStudents: Array o range

Ora io mi sono preso delle licenze poetiche storiche tanto da aver sviluppato un dialetto della notazione ungara che è diventato col tempo così:

iSize: numero intero
ulDimX: long senza segno
dwLightYears: double Word
bBusy: booleano
fBusy: float
pFoo: puntatore

etc... etc...

il codice suddetto seguendo la mia licenza poetica sarebbe diventato:

```
for ( iNdx = 0 ; iNdx <= ulFwSize ; iNdx += 2)
{
    iCrc = chBufferFw[iNdx];
    calcolo...
}
```


Faccio una piccola parentesi sulla scrittura dei programmi. Quando scrivo software indento ad ogni test condizionale, mettendo la parentesi graffa allineata con il test e il codice avanzato di un tab in questo modo:

```
if(<test condizionale>)  
{  
  istr1;  
  istr2;  
  if(<test condizionale>)  
    {  
      istr1;  
      istr2;  
    }  
}
```

Il termine Indento non è un errore di battitura ma deriva dal verbo indentare, che significa mettere i denti (cit. Treccani) ma in questo caso ha il senso di allineamento.

Dicevamo che questa è una regola nella mia testa, tutto il mio codice è scritto così. Quando lo guardo mi sembra più ordinato, pulito, quasi a scongiurare qualsiasi evento degenerare.

Spesso, quando l'algoritmo diventa complicato, lo sviluppo del software ci potrebbe costringere ad annidare diversi livelli di parentesi.

Durante le fasi di debugging, aggiungendo e togliendo codice in stato epilettico compulsivo da baco, se si per-

de cancellando per sbaglio una parentesi graffa, si creano piccoli prostranti incubi. Questo accade perché il C, proprio per come è strutturato il linguaggio, non riesce a rilevare la condizione di inizio o termine del ciclo, la dove si è creato. Gli errori, spesso criptici, potrebbero avvenire centinaia di righe dopo la mancata parentesi o addirittura anche in moduli diversi. Con questo allineamento, riesco agevolmente ad associare la loro corretta apertura e chiusura seguendo semplicemente l'indentazione.

I programmatori della Microsoft in tutti i loro esempi su web, invece la prima parentesi la mettono al termine del test condizionale e il test di prima per loro diventa così:

```
if(<test condizionale>) {  
    istr1;  
    istr2;  
    if(<test condizionale>) {  
        istr1;  
        istr2;  
    }  
}
```

Dopo essermi giocato il posto alla Microsoft, riprendiamo il discorso della **notazione unghera** allargandolo alla dichiarazione delle variabili. Queste dovrebbero essere fatte in modo da richiamare l'uso a cui sono previste. Se la variabile del ciclo for la utilizzo come indice di un vettore, la chiamerò in un modo che mi ricordi per quale motivo mi serve, nel nostro esempio potrebbe essere chiamata così *ilIndexVect*.

Le dichiarazioni minimaliste tipo 'i' 'b' o 'xy' sono troppo essenziali e vanno bene solo per dei registri di un micro però, dal lato opposto, bisognerebbe evitare anche di essere troppo prolissi. Nomi di variabile del tipo **"NumerdoDiConteggiMensili"** sono in genere inutili e alla fine si ottiene l'effetto contrario, perché rendono il codice pesante da leggere e per questo in genere sconsigliati.

Il pregio della notazione ungara è quello di rendere evidente al firmwareista il tipo di variabili utilizzate, e quindi il loro uso improprio. Con un colpo d'occhio, diventa chiara la porcata che il C rende legittima, quella appunto di confrontare impunemente, dati di tipo diverso.

Una delle massime regole, del buon debugger è quella di non dare mai troppa fiducia ai compilatori.

Baco presunto

- ciao a tutti,so che si può sviluppare in iOS solo con una macchina apple,c'è qualche modo per ovviare a questo problema?
 - link con una macchina virtuale su windows
 - cioè devo virtualizzare iOS sul pc!? una delle procedure più complicate! non c'è un modo più semplice?
 - Comprati un Mac
- (forum Html.it)

Prima di parlare del baco presunto volevo raccontare un piccolo aneddoto perché, come qualcuno ha già capito, sono un tipo che ogni spesso je rode...

leri sono andato in banca per le pulizie di primavera, togliere le ragnatele dal conto, sentire l'eco del vuoto creato dall'inflazione e così via. La direttrice ha chiesto di incontrarmi per capire chi fossi, vista la mia naturale refrattarietà ai contatti sociali, di qualsiasi tipo. Dopo i primi convenevoli si parlottò del più e del meno con discorsi alla deriva senza un particolare filo logico. Parlai di una recente avventura negativa avuta con le poste.

Lei annui piuttosto compiaciuta, lamentandosi a ruota, che l'istituto dovrebbe fare il mestiere di postini o portaflettere e non di gestire il denaro della gente (e fregare i clienti alle banche). Coerentemente con quanto affermato, di lì a poco tentò di rifilarmi con tecniche coercitive, una nuovissima e convenientissima, polizza di assicurazione sulla casa.

Diciamo che ebbi una sorta di disagio però non ebbi il coraggio di stroncarle l'entusiasmo che mise nel propor-mi questo prodotto finanziario. Debolemente incuriosito, mi sono fatto dare il prospetto informativo e la sera anziché leggermi un buon libro, ho dato un'occhiata agli articoli e le varie clausole.

Stentavo a credere ai miei occhi e a quello che c'era e c'è scritto, così cito il documento che ufficialmente è stato redatto dalla Avipop assicurazioni s.p.a. modello PN 026627/B ed. 04/2012 pagina 6 di 22., al suo interno infatti si legge:

Articolo 1.8

Sono esclusi dalla assicurazione...

f) danni causati da guerra civile con o senza dichiarazione, insurrezioni, invasioni e ostilità... trasmutazione

del nucleo di atomo nonché da radiazioni provocate dall'accelerazione artificiale di particelle atomiche e da esposizione a radiazioni ionizzanti;

e fin qui inevitabilmente sorrido, perché chi ha scritto questo rimando è uno che davvero crede in quello che fa... poi il testo continua

g) danni causati da terremoti, eruzioni vulcaniche, inondazioni, alluvioni, allagamenti, mareggiate, frane e/o cedimenti del terreno valanghe e slavine...

e anche qui nessuno replicherebbe alla legittimità del mancato indennizzo, se non che al prossimo articolo

h) i danni di qualsiasi natura direttamente od indirettamente derivanti da mancato, errato, inadeguato funzionamento e/o elaborazione del sistema informativo e/o di qualsiasi impianto apparecchiatura, componente elettronica, firmware, hardware.

Questa sorta di malcelata associazione che accomuna il lavoro da firmwareista a quello di una calamità naturale, è veramente sintomatica. Il lavoro da programmatore, nell'immaginario collettivo, è considerato alla stessa stregua di un pianeta che entra in collisione con la terra o ad un'esplosione di un super vulcano che sommerge di lava l'intero globo terraqueo.

Mi chiedo, ma tutte quelle storie relative al controllo

qualitativo del software dove sono andate a finire ? Le ISO sul software ? La documentazione di progetto ? Evidentemente non sono bastate e la compagnia di assicurazione, che di previdenza se ne intende, ha pensato bene di risolvere la questione alle origini del problema. Se la casa ti piglia fuoco perché nella caldaia c'è un baco, semplicemente ti arrangi e forse, in questo caso, si potrebbe anche usare un altro intercalare, ma siamo dei gentiluomini...

Dopo essermi fatto la mia quotidiana rosicata... riprendiamo la storia del Baco presunto. Nella spiegazione di baco citata da Wikipedia si includeva anche una estensione del significato meno comune, che riprendo:

“...Meno comunemente, il termine bug può indicare un difetto di progettazione in un componente hardware, che ne causa un comportamento imprevisto o comunque diverso da quello specificato dal produttore.”

In questo esempio, considereremo un tipo di baco in questa eccezione migliore. Il cliente era una nota azienda di elettrodomestici ed aveva richiesto un sistema embedded per il controllo del suo condizionatore domestico. Il progetto firmware era già consolidato da tempo e fu inserito in questa nuova piattaforma, che però aveva lo stesso microcontrollore dei precedenti progetti.

Le funzioni più o meno erano le solite: un controllo ventola con pid a taglio di fase, un ingresso tachimetrica con frequenza 20Hz -2 KHz, un controllo per un motore passo passo dedicato al deflettore, una interfaccia seriale ad infrarossi per il telecomando, un paio di ingressi

analogici per la lettura di due sonde temperatura **NTC**, un uscita relè per il compressore e poco altro...

Il cliente installò la scheda su un nuovo prototipo, ma dopo 6 mesi di funzionamento corretto, la macchina andava sistematicamente allarme di NTC in corto. In questa condizione il firmware si suicidava disattivando il condizionatore senza possibilità di ripristino. In realtà le possibilità di ripristino c'erano; spegnendo e riaccendendo, la macchina, l'utente riusciva infatti a far riprendere il suo funzionamento corretto.

Ma in questa condizione però poteva durare solo una decina di minuti, passato questo tempo si ripeteva lo stato di errore sulla NTC che provocava il nuovo arresto.

Il modulo Ntc, era composto da due parti di codice differenziate. Una parte, era dedicata alla misura, a cui veniva associata una funzione di linearizzazione con spezzate. Nella seconda parte, venivano invece, attivati e disattivati i flag di sistema a seconda della temperatura raggiunta. Sui margini del range operativo, avevamo quindi implementato gli allarmi di Ntc Aperta e Ntc in corto.

Questi margini, previsti, erano -20°C e +60 °C, limiti sovrabbondanti per una temperatura interna ad una tipica stanza italiana. Il progetto iniziale, che prevedeva inizialmente l'utilizzo della sola temperatura ambiente, in corso d'opera fu soggetto ad una modifica che rendeva necessaria l'implementazione di una sonda in grado di misurare anche la temperatura della batteria esterna.

I margini dichiarati di utilizzo della batteria erano di -10°C e 50°C **range** che comunque garantiva l'utilizzo dello stesso modulo per il secondo tipo di sonda.

Il modulo nuovo però doveva essere utilizzato per una nuova serie di condizionatori di potenza frigorifera superiore. Nei condizionatori d'aria c'è un parametro che definisce appunto la potenza frigorifera dell'elettrodomestico, questo parametro è definito dalla sigla BTU (British Thermal Unit).

“Una BTU è definita dalla quantità di calore richiesta per alzare la temperatura di 1 libbra (ovvero 453,59237 grammi) di acqua da 39 °F a 40 °F (3.8 °C a 4.4 °C) il BTU all'ora (BTU/h) è l'unità di potenza associata alla BTU: 1 cavallo vapore è approssimativamente pari a 2540 BTU/h 1 watt è approssimativamente pari a 3,412 BTU/h 1000 BTU/h sono quindi approssimativamente pari a 293 W 1 Frigorio/h = 3,97 BTU/h” (wikipedia)

In Italia esiste la misura definita frigoria molto più efficiente e pratica, però in questo sistema sfacciamo dove si globalizza considerando le cose non con la testa ma con altre parti del corpo, si preferiscono i cavalli fatti a vapore, le iarde ai metri le libbre ai chilogrammi.

“la frigoria è la quantità di calore che deve essere sottratta da un chilogrammo d'acqua per abbassarne la temperatura da 15,5 °C a 14,5 °C alla pressione di 1 atmosfera”. (wikipedia)

Il sistema economico prendendo il sopravvento sulle

persone ha stabilito di imperio che il parametro BTU venga utilizzato come misura termica.

I condizionatori tipicamente avevano poche migliaia di **BTU** e i modelli di punta a quel tempo, vantavano l'astronomico target da 9000 btu. Il miglior rendimento dei compressori, associato a nuove ingegnerizzazioni delle batterie, utilizzate per la dissipazione del calore, permise lo sviluppo industriale di nuovi condizionatori con potenze frigorifere (?) impensabili che raggiunsero così i target attuali da 12000 BTU, ampiamente sufficienti per appartamenti di medio piccole dimensioni.

Ritorniamo al nostro racconto. Cosa poteva causare un problema sulla lettura di temperatura dopo sei mesi di funzionamento corretto ? Quale problema logico nel firmware poteva causare questa anomalia ? Dove era il baco ?

In questo caso il firmware faceva quello che doveva fare, il vero problema nasceva da una diversa implementazione del sistema di controllo. Così come spiega la citazione iniziale, il baco nella sua estensione è qualcosa che non deve essere necessariamente legato ad un problema software. L'apparato era stato inserito sopra una macchina più performante con diverse specifiche o per meglio dire con differenti limiti di targa. Il calore della batteria esterna del nuovo condizionatore, durante il periodo invernale, non aveva dato problemi.

Durante il primo periodo estivo le cose andarono bene sino a quando, verso luglio nei giorni più caldi dell'alta Brianza, le temperature della batteria esterna aumentarono, superando il limite tale per cui raggiunto un valore

di resistenza limite, veniva rilevata la condizione di NTC in corto (60 °C).

Per questo tipo di bachi, che potremo appunto chiamare presunti, non ci sono round tips. Sono bachi di progetto e non di firmware che accadono tipicamente quando non si conosce bene il sistema e le sue risposte funzionali. Questi bachi sono in genere direttamente proporzionali al numero di persone che mettono bocca sul progetto. In questo caso per risolverlo, si è semplicemente reingegnerizzato il modulo modificando la lettura analogica per implementare l'allarme NTC in corto, in modo più efficace con temperature equivalenti molto più alte.

Baco specialistico

drew99: Dichiaro un array dinamico multidimensionale grande 65536 * 65536, ma il programma non riesce ad inizializzare l'intero array....

ixamit: $65536 * 65536 * 4$ (32bit) = 17.179.869.184 bytes - sono

un *po'* troppi in RAM

(forum iprogamatori.it)

Le sezioni di codice implementato hanno peculiari punti deboli. Alcuni dovuti al linguaggio altri al firmwareista. Direi che un firmware strutturato di un sistema embedded ha in genere quattro sezioni di codice.

- Sezione interrupt
- Sezione ciclica
- Sezione procedurale di ingresso
- Sezione procedurale di uscita.

La sezione di interrupt interagisce con le altre parti del software condividendo le risorse. Questa condivisione è anche la parte debole di questa sezione, perché chi legge è naturalmente desincronizzato rispetto da chi scrive. Per sapere quando ci sono stati gli aggiornamenti delle strutture dati condivise, si devono implementare semplici sistemi di semafori booleani. La sezione ciclica è quella che contiene meno bachi, perché piuttosto semplice da implementare.

Tipicamente in questo caso le parti deboli, sono quelle dove il firmwareista si trova più disagio. Se ad esempio abbiamo perplessità con i puntatori si cercherà di usarne il meno possibile.

Ovviamente questa è un esempio iperbole che possiamo associare solamente ad un novizio. Un firmwarista vero che lavora in C semplicemente non deve avere problemi.

Nelle sezioni di ingresso e uscita, risiede il 90% del codice, qui troveremo quasi tutti i bachi nelle loro forme più variegate e differenziate. Quello che considereremo però non è un progetto firmware ma bensì un progetto software su PC, scritto in Visual Studio 6.0. Si lo so il sistema è un po' datato, i progetti in C++ dopo anni di indiscusso dominio stanno piano piano lasciando il passo, al più recente C#.

A chi interessasse in questa pagina della Microsoft vengono spiegate le differenze più importanti fra i due linguaggi ([link](#)). In questa pagina invece un elenco con il rating di popolarità dei linguaggi ([link](#)).

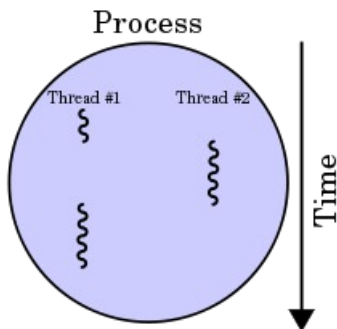
Noi invece parleremo, sì di un software C++ sviluppato in modo particolare non ad oggetti ma a Thread, un modulo monolitico di circa 250.000 righe di codice. Per questo lo possiamo considerare come un abnorme sistema embedded degenerato.

Questo software era una specie di centro di raccolta dati, sviluppato per una delle tante aziende di servizi. Con le dovute proporzioni era un piccolo sistema di acquisizione di misure che dovevano essere prese dalle periferiche installate sul campo e **“semplicemente”** salvate in un database; il peggiore di tutti, Access. La semplicità della descrizione appena data è diventata inversamente proporzionale alla complessità algoritmica raggiunta.

Le migliaia di righe di codice, che con il tempo si sono aggiunte, sono diventate prima decine e poi centinaia di migliaia rendendo il software una sorta di mammut impazzito che sculetta dentro un negozietto di cristalli Swarovski.

Ma veniamo un attimo all'elemento importante di questo software, il thread.

La Microsoft aveva implementato nel sistema operativo di famiglia, Windows, la possibilità di creare queste fantomatiche strutture chiamate thread. Queste strutture erano riprese sempre dagli antesignani sistemi Unix che erano, al tempo, il riferimento principale delle applicazioni multitasking. In un sistema mono processore due thread funzionano così:



220px-Multithreaded_process.svg.png (immagine presa da Wikipedia)

Il thread ha 4 stati definiti

- Creazione
- Blocco
- Sblocco
- Terminazione

Lo stato di creazione e di terminazione non necessitano di particolari spiegazioni. Gli stati di Blocco e Sblocco sono invece più complessi perché sono un po' il cuore del thread. Vengono gestiti da strutture dati che in gergo si chiamano semafori e che sono associati agli eventi di sistema. Tipicamente, possono essere di due tipi:

- asincroni
- sincroni

Quelli asincroni sono eventi non predicibili, quelli sincroni invece sono tipicamente associati ad un timer e quindi avvengono in un tempo noto. Un thread, ricorda molto la **features** degli interrupt nei

microcontrollori più complessi perché hanno la stessa tipologia degli eventi di sistema. Dopo questo stringato excursus sui thread veniamo al nostro software. Diciamo che nonostante il software soffrisse di questa Filariasi linfatica cronica, tutte le installazioni fino a quel tempo non avevano dato grossi problemi.

Il vero problema nasceva dal fatto che saltuariamente il programma andava in crash con una frequenza, però, talmente bassa, da rendere particolarmente difficile le operazioni di debug. Nell'ultima installazione le cose andarono molto diversamente, il **crash** era sistematico e quotidiano. La frequenza era tale che il softwareista poteva verificarlo direttamente. Benissimo !

I centri di raccolta dati hanno dei sistemi hardware un po' particolari. Le installazioni si basano tutte su una piattaforma che ha alla base un ambiente definito con il termine tecnico di **macchina virtuale**. Tipicamente i desktop di casa, come quello che usiamo tutti noi, sono monolitici; intendo dire che ad una unica base hardware, corrisponde un unico sistema operativo che si interfaccia con l'utente. Alcune case come la VMWare hanno sviluppato complessi software di virtualizzazione di macchine e reti.

Semplificando molto, questi software permettono appunto di creare più istanze di interfaccia utente, cioè ad un'unica base hardware puoi avere più sistemi operativi che possono essere utilizzati da più utenti.

I thread implementati erano diversi, ma per essere sicuri che questi non interagissero maleficamente fra di loro,

era stata inserita una gestione a semafori in modo tale che, girassero solo due thread alla volta: quello della dialog principale (sempre), e uno (e solo uno) degli altri thread disponibili. Ora il softwareista con un colpo d'occhio notò che la macchina virtuale prevedeva l'implementazione di due processori di tipo **Xeon**.

Gli Xeon sono storicamente la variante più ricca dei processori Pentium erano quelli diciamo utilizzati dai professionisti e da chi aveva le palanche... . L'amministratore di rete a seguito delle nostre perplessità ci disse pacifico.

“Si certo...i due Xeon li ho configurati io. Fanno parte delle impostazioni di virtualizzazione della macchina, se volevo ne potevo mettere anche 4 di processori... ma due mi sembravano sovrabbondanti ”

Il sospetto era che sin'ora tutte le installazioni era state fatte su PC da quattro soldi che sono di fatto mono processore. In questo caso, l'implementazione dei due processori era vera (anche se virtuale) e il **VMWare** stava funzionando alla grande. I thread dell'applicativo in questione sulla macchina giravano d'avvero contemporaneamente e questo rendeva i crash da saltuari a sistematici. In pratica il softwareista pensò che ci fosse un problema di **Mutex** fra i thread.

Su wikipedia da grandi inconsapevoli feticchie alla voce thread scrivono con grande solerzia:

La programmazione con diversi thread ha reputazione di essere una tecnica abbastanza difficile. In realtà, usare un thread non comporta particolari difficoltà di

per sé: la difficoltà consiste usualmente nel sincronizzare questi diversi thread, perché non è mai possibile sapere, al momento della programmazione, a che punto dell'esecuzione sarà un thread specifico: la loro progressione dipende infatti dalle priorità decise dal sistema operativo. In particolare la caratteristica tipica dei thread di condividere risorse tra loro, come appunto la sezione di dati, può portare

ad alcuni problemi. Più thread infatti possono accedere ad una stessa variabile e modificarne il valore.

Sebbene questo non accada nello stesso momento, perché l'accesso ad una variabile, che di fatto è memorizzata in memoria RAM, è intrinsecamente limitata ad un'unità al massimo, può accadere che un thread modifichi il valore di una variabile, mentre un altro thread necessita del vecchio valore memorizzato in essa. Si ricorre pertanto all'uso di tecniche di sincronizzazione come la mutua esclusione per risolvere il problema.

Ne consegue che, idealmente, un thread dovrebbe eseguire del codice quanto più possibile indipendente dal resto del programma. Inoltre, gli errori nella sincronizzazione tra thread sono spesso molto difficili da individuare, perché la loro occorrenza dipende essenzialmente dall'ambiente in cui il programma viene eseguito. La sincronizzazione di un thread con un altro è normalmente necessaria per permettere a questi di comunicare tra di loro e di restituire i risultati di una funzione al processo principale; essa viene normalmente realizzata tramite mutex.

(wikipedia)

Come si può notare, il problema della condivisione delle

risorse fra thread è una cosa abbastanza nota nell'ambiente, tanto che mamma Microsoft, per ovviare ai problemi di sincronizzazione, un round tips lo aveva già implementato nella notte dei tempi chiamandolo

Critical Section (d'ora in avanti **CS**)

quella che chiamano Mutex e cioè la mutua esclusione dei thread, deve avvenire seguendo una serie di precise condizioni che riprendo sempre da wikipedia...

- un solo processo o thread accede alla regione critica;
- nessun processo può bloccarsi in sezione critica
- evitare deadlock e starvation
- nessuna assunzione sulla velocità dei processi
- nessun processo fuori dalla sezione critica può impedire ad un altro di entrare
- accesso alla sezione critica consentito in un tempo definito.

(wikipedia)

Nel punto tre, quando dicono di evitare che accadano altri devastanti bachi come il deadlock o lo starvation, si potrebbe rispondere con altri pesanti intercalari, ma siccome abbiamo imparato ad essere dei signori, evitiamo le provocazioni. Riprendiamo invece la **CS** e la sua implementazione. Nel costruttore dell'oggetto dichiariamo l'inizializzazione

```
Class::TestClass(void)
{
    InitializeCriticalSection (&m_criticalSection);
}
```

Nel distruttore invece la disabilitiamo

```

Class::~~TestClass(void)
{
DeleteCriticalSection(&m_criticalSection);
}

```

Le sezioni di codice da proteggere devono quindi essere scritte così:

```

EnterCriticalSection (&m_criticalSection);
AddItem(p);
LeaveCriticalSection (&m_criticalSection);

```

Per associazione con i microprocessori la CS ricorda molto le istruzioni CLI e SEI di alcuni microcontrollori (Atmel su tutti) che abilitano e disabilitano gli interrupt globali. Spesso vengono utilizzati appunto in contesti simili e cioè quando abbiamo sezioni di codice che gestiscono parti critiche che non possono essere interrotte da nessuno.

Il crash svanì d'incanto e come per magia il softwareista si risollevò dalla pressione continua che quel baco aveva causato da tempo. Quella sera i suoi colleghi lo salutarono e lui rimase per un momento interdetto ma quella volta diversamente dalle altre alzò la testa, allungò il dito sul tasto di on/off del computer spegnendolo e nello stesso tempo disse “aspettatemi ... esco con voi”.

Come scrivere codice

ingestibile

In questo post, darò una serie (limitata per altro) di indicazioni sullo sviluppo del codice che se applicate, lo renderanno impossibile da gestire da qualsiasi persona. Nella loro pedissequa applicazione il codice manipolato diventerà ben presto inutilizzabile anche allo stesso autore.

Diciamo subito che è un post, dai tratti spiccatamente goliardici, ed è ripreso da una pagina web (link) scritta in modo magistrale da un esperto softwarista d'oltreoceano. L'intento esplicito dell'autore, rivolto ai soli softwareisiti, (java per altro) è quello di contrastare il più possibile l'eventuale scelta aziendale di silurarti. Ne riprenderò le struttura con qualche personale aggiunta e modifica, in modo da integrarli in alcune parti a mio avviso mancanti. Per essere dei veri cani nella programmazione ci vuole metodo e oggi impareremo come.

In verità però c'è da dire anche che, se i presupposti del post sono delle poco etiche rivalse prototerroistiche aziendali, questa panoramica di indicazioni, sono comunque un ausilio per chi ha intenzione di intraprendere la carriera da programmatore. Basta infatti mettere una **negazione** ad ogni post e otterremo esattamente l'opposto di quello che il nostro amico voleva ottenere, a voi la scelta.

Durante lo sviluppo del codice, il programmatore serio, deve pensare anche quando dovrà gestire il software nei tempi futuri. Questo perchè lo sviluppo di un qualsiasi software non è mai un progetto a *tempo determinato*. Per tanti motivi, saranno applicate eccezioni progettuali che necessiteranno l'integrazione di nuove sezioni di codice o ci saranno comunque attività di manutenzione che sono inevitabilmente necessarie. Noi dovremmo fare in modo di rendere il codice non mantenibile, rendendo il più difficile possibile qualsiasi postumo tentativo di modifica.

Ma prima di parlare delle tecniche di non mantenibilità, parliamo un attimo di cosa vuol dire la parola mantenibilità. Per fare questo su web ho trovato questa rigorosa definizione:

Mantenibilità

La mantenibilità è una misura della facilità con la quale un sistema può essere riparato una volta manifestatosi il malfunzionamento. In maniera più specifica, la mantenibilità è la probabilità $M(t)$ che il sistema malfunzionante possa essere riportato al suo corretto funzionamento entro il periodo t .

Essa è strettamente correlata con la disponibilità poiché tanto più è breve l'intervallo di ripristino del corretto funzionamento, tanto più elevata sarà la probabilità di trovare il sistema funzionante ad un dato istante temporale.

Per il valore estremo $M(0) = 1.0$, il sistema in oggetto sarà sempre disponibile.

Ora per rendere codice **ingestibile** bisogna pensare come il futuro programmatore tenterà di approcciarsi alle migliaia di righe di codice che hai scritto. In pratica devi essere malvagiamente empatico. Si parte dalla premessa che, se tu fossi lo sventurato successore di qualche prolifico softwareista, non potrai sicuramente leggerti tutti i moduli sorgente cercando, magari di capirne il "senso" trascrivendone i diagrammi di flusso. L'unica cosa che potrai fare è quella di interpretare le sezioni di codice che ti servono per risolvere il problema contingente. Così il nostro target sarà quello di diminuire il più possibile il grado di comprensione del nostro codice o per esserè più rigorosi tenderemo di azzerare l'indice di mantenibilità

Nomenclatura delle variabili

Quidquid latine dictum sit, altum sonatur. - (qualsiasi cosa detta in latino, sembrerà importante)

Le variabili vengono utilizzate nel 99% del nostro codice. La gestione e manipolazione delle variabili aumenta o diminuisce il grado di comprensione del software. In questa sezione consiederemo nel dettaglio come dare un nome specifico alle variabili.

Nomenclatura onomastica

Prendete un libro di nomi per bambini. Se sono stranieri è ancora meglio. Cercate quelli più caratteristici e applicateli alle variabili specifiche del vostro programma. Fred per esempio è un nome di variabile perfetto. Ci sono altri nomi particolarmente belli e facili da scrivere perchè sono sequenze di tasti come ad esempio asdf e

qwerty e che vengono usati come nick dei forum...

Minimalista

Dichiarate le variabili con singole lettere x, y, vanno bene anche le accoppiate xy xz vanno benissimo anche i dittonghi ia ie etc. etc.

Scrittura erronea creativa

In questo caso le variabili vanno dichiarate con errori sintattici come, ad esempio, ldnice, Miusra o ad esempio gms o comnado. Possono essere chiamate anche dichiarazioni da refuso (un mio collega ha scritto un modulo gms.cpp utilizzandolo su una decina di progetti contemporaneamente e lo maledico tutte le volte che ci penso...).

Concetti astratti

Utilizzare termini il più generici possibile, tipo appunto: generico, esso, data, routine, o ad esempio anche dare il nome variabile è bellissimo

Acronimi

Definire il numero più ampio di acronimi possibile, ricordandosi ovviamente di evitare spiegazioni.

es:

ABPP SCRTD

Lingue straniere

Utilizzare definizioni di variabili in lingua straniera. Nel dettaglio quelle tipo Klingon, Hobbiteese e Esperanto vanno benissimo.

Maisucolo randomico

In questo caso le variabili sono molto lunghe e approfittando che i parser dei compilatori sono sempre case

sensitive si inserisce una maiuscola a caso, tipo VariabileCondaria()

Accenti

Mettiamo subito l'esempio così si capisce...

```
typedef struct { char ch; } int;  
int iPippo;
```

Lunghezza

I parser dei compilatori hanno un numero massimo di lettere verificate tipicamente 8. Francamente nei compilatori Microsoft non lo so se sono davvero 8. Comunque sia una volta dichiarate le costanti posso ottenere una cosa di questo tipo:

Numero_Massimo_diCaratteri

Numero_Massimo_diLettere

In entrambi i casi il compilatore le tratterà allo stesso modo "Numero_M"

ASCII esteso

Utilizzare nella dichiarazione delle variabili i caratteri estesi ASCII (ma questa è proprio un suicidio istantaneo dopo una decina di variabili scritte così, come farete davvero non lo so... auguri!)

Riferimenti matematici

Utilizzare nomi di variabili tipo barra/asterisco=addendo... .

Nomi accecanti

Utilizzare nomenclature non convenzionali

esempio:

Freud *superman=robocop+startrek

Lower Case l Looks a Lot Like the Digit 1

se non sei un anglofono copia il sottotitolo e traducilo con google capirai... subito ;)

Riusa le variabili globali come private o locali

Questa è una tipica possibilità del C che viene spacciata per un pregio, invece è un difetto. Dichiarare ad esempio una variabile globale `Spare` e una uguale all'interno delle funzioni rende difficoltoso seguire il flusso della variabile. Diciamo che, tendenzialmente, più variabili globali usi e più il codice diventa illeggibile e meno sicuro.

Cd wrtn wtht vwls s mch trsr

Anche le abbreviazioni abbassano notevolmente il grado di capibilità del codice, per ovvi motivi.

HOT Tips di offuscamento del codice

The longer it takes for a bug to surface, the harder it is to find. - Roedy Green

Una delle indicazioni più gettonate da parte dei team manager è quella di richiedere lo sviluppo di codice **autoesplicante**. Questo perchè si sono accorti che le modifiche a posteriori hanno l'effetto nefasto di disallineare i commenti dal codice. Come abbiamo già chiarito il codice autoesplicante è esattamente l'opposto di quello che stiamo facendo noi quindi....

Aggiungi tonnellate commenti

Il vero motivo per richiedere codice autoesplicante è

che i commenti oltre a disallinearsi con il codice se scritti in in quantità diventano comunque difficili da leggere, spesso si ottiene l'effetto contrario a quello sperato. I commenti sono per definizione ambigui e fuorvianti. Così, scrivi commenti come se fossi il Proust degli *schiacciatasti* (suffisso dato agli ingegneri informatici di origine ignota, l'equivalente per gli ingegneri elettronici è di *spellafili* per i maccanici *girabulloni*... e così via ognuno ha il suo). Scrivi, su tutte le linee di codice Includendo tutorial ed esempi di utilizzo del sorgente proprio come se fosse un manuale di istruzioni. Descrivi tutte le funzioni anche le più stupide.

es:

```
i++; // incrementa di uno
```

L'effetto minimo è quello di far perdere tempo a chi deve modificare il tuo codice, se poi hai la premura di inserire anche qualche minchiata quà e là si possono ottenere spiacevoli effetti collaterali.

Togli gli spazi

Elimina tutti gli spazi del codice in modo che le linee diventino cluster di caratteri. Queste possono diventare espressioni artistiche di pregievole livello.es: l'effetto del codice senza spazi... è questo...

```
@ifdef PUMP_TIMEOUT // Verifico che il tempo trascorso dall'ultima attivazione // sia superiore a Tmin_Partenze ( campo K_Tmin_Partenze )
```

```
timeCurrTimeDate=COleDateTime::GetCurrentTime();
```

```
timeCurrTimeDate = timeCurrTimeDate – timeTime_Ac-
```

```

tive_Pump;
timeDelta.SetTime(0,IKTimer,0);
if ( timeCurrTimeDate > timeDelta )
{
    if (ITimerHold <= ITimerEs )
    {
#endif
        ITimerHold = ITimerEs;
        iCurrIdx =(int) lld;
#ifdef PUMP_TIMEOUT
    }
}
#endif

```

Inverti il senso

Se dalla descrizione della funzione deve tornare il valore vero, invertite il senso.

Es:

```
If ( isNegative( iValue ) )
```

isNegative è una funzione che ritorna uno stato booleano. In questo caso il softwareista si aspetta che il parametro iValue sia vero quando è un valore negativo...

Mischia i linguaggi

Questo nella stesura del codice è davvero spazzante.

Se riuscite a mischiare le applicazioni fra Delphi, C++, Python, Php renderete il programma mooolto illeggibili.

La notazione Ungara

In questo caso la *notazione ungarica* è una vera arma nucleare nella guerra di offuscamento del codice. Basta mischiare i tipi di dato con la notazione in modo casuale.

es:

```
unsigned char pPointer;
```

Mascherare il codice

Any fool can tell the truth, but it requires a man of some sense to know how to lie well.- Samuel Butler (1835 – 1902)

Incorrect documentation is often worse than no documentation.- Bertrand Meyer

Includere in una sezione di codice una parte di commento nel seguente modo

```
for(j=0; j<array_len; j+=8)
{
    total += array[j+0 ];
    total += array[j+1 ];
    total += array[j+2 ];/* Main body of
    total += array[j+3]; * loop is unrolled
    total += array[j+4]; * for greater speed.
    total += array[j+5]; */
    total += array[j+6 ]; total += array[j+7 ];
}
```

Definizione di macro

Dichiarare una macro in modo che l'operazione vista non sia quella effettuata così come descritto nell'esempio seguente.

```
@define a=b a=0-b
```

Prolungamento di linea

la seguente dichiarazione di macro

```
@define local_var xy_z
```

potrebbe diventare

```
@define local_var xy_z // local_var OK
```

l'effetto è quello che il *back slash* estende la linea senza aggiungere *line feed* o *carriage return*. Il pregio di questa dichiarazione è che la variabile non può più essere trovata con uno strumento di ricerca come il *find* o il *grep*.

Overloading

Anche qui possiamo mischiare le operazioni approfittando di questa potente feature del C++. Ovviamente l'overloading migliore è quello che ti permette di differenziare le operazioni in modo tale che l'operazione ad esempio `i = i + 5`; dia un diverso risultato rispetto all'istruzione `i += 5`. L'overloading è lo stato dell'arte fra le operazioni di offuscamento del codice perchè agisce nel tempo, proprio mentre il nostro ignaro succedaneo continuerà a lavorare con il nostro software. Overloadare funzioni come il not "!" o la "new" è una vera libidine....

Doppio puntatore

Dal punto di vista algoritmico il doppio puntatore è una struttura dati che non serve a nulla. Ma fa tanto Nerd...

es:

```
int **ipp;
```

```
int i = 5;
```

```
int *ip1 = &i, ipp = &ip1;
```

Callback

Questo è un tipo di riferimento a funzione, che proviene da una notoria sega mentale algoritmica informatica. Sostanzialmente le *callback* sono riferimenti a funzioni. Una tipica implementazione è quella di permettere ad una stessa funzione di gestire diverse strutture dati .

es: (ripreso da Wikip)

```
@include <stdlib.h>
```

```
@include <string.h>
```

```
@define QUANTI_SOLDATI 100
```

```
struct TSoldato  
{  
    int grado;  
    char nome[80];  
}
```

```
Esercito[QUANTI_SOLDATI];
```

```
int ConfrontaSoldati(const void *ptr1, const void *ptr2)
{
    const TSoldato *s1=(const TSoldato*)ptr1;
    // GNU GCC Compiler necessita della parola chiave
    "struct" prima di TSoldato
    // sia a sinistra che a destra dell'operatore di assegna-
    zione
    const TSoldato *s2=(const TSoldato*)ptr2;
    // come sopra
    if(s1->grado!=s2->grado)
        return s1->grado - s2->grado;
    return strcmp(s1->nome, s2->nome);
}
```

```
void OrdinaEsercito()
```

```
{
    qsort(Esercito, QUANTI_SOLDATI, sizeof(TSoldato),
    ConfrontaSoldati); // come sopra, all'interno della 'sizeof'
    prima di TSoldato

    // scrivere la parola chiave "struct"
}
```

La funzione Ordina esercito richiama una funzione standard del C la qsort che ha bisogno di una funzione di callback in grado di eseguire il confronto di un tipo di dato creato dall'utente. Come potete notare non ci sono

riferimenti alla procedura e la ConfrontaSoldati ha un uso ambiguo come parametro.

Codice ricorsivo

Più codice ricorsivo implementi e meno leggibile diventa il codice. Ad esempio prendete un cronometro e misurate il tempo che impiegate a comprendere che tipo di operazione matematica esegue questa funzione ricorsiva.

```
int x (int n)
{
    if (n <= 1) return 1;
    else
        return n * x (n-1);
}
```

Certo è, che se al nome della funzione avessi utilizzato un altro nome tipo, anzichè x che ne so “fatt”, allora il tempo di comprensione sarebbe stato nullo.

Funzioni monotone

Le funzioni dovrebbero essere tendenzialmente **endofunzioni idempotenti**. Prendo a piene mani dalla terminologia matematica forzando un po' il significato. Più parametri complessi vengono utilizzati e più l'indice di mantenibilità diminuisce.

[modifica]Namespace

Le strutture/unioni e la definizione di tipo struttura/unione hanno diverse dichiarazioni. Puoi usarle insieme dichiarandole con lo stesso nome. Se poi sono anche simili (ma non uguali) è meglio.

```
typedef struct { char* pTr; size_t lEn; } snafu;  
struct snafu { unsigned cNt char* pTr; size_t lEn; } A;
```

Macro inutili

Definisci funzioni che sembrano macro ma che non fanno niente. Magari commentando gli argomenti

```
1.define fastcopy(x,y,z) /*xyz*/  
fastcopy(array1, array2, size); /* does nothing */
```

Nascondi le istanze

Sempre con la define dichiara variabili del preprocessore cambiandole fra un header e l'altro

```
@define xxx global_var // in file std.h  
@define xy_z xxx // in file ..\other\substd.h  
@define local_var xy_z // in file ..\codestd\inst.h Define  
@ifndef DONE
```

```
@ifdef TWICE // put stuff here to declare 3rd time  
around
```

```
void g(char* str);
```

```
@define DONE  
@else // TWICE
```

```
@ifdef ONCE // put stuff here to declare 2nd time  
around
```

```
void g(void* str);
```



```
@define TWICE
```

```
@else // ONCE // put stuff here to declare 1st time  
around
```

```
void g(std::string str);
```

```
@define                                ONCE  
@endif                                //  
@endif                                //  
@endif // DONE
```

Ritorni di funzione

Nella stesura di una funzione canonica ci si aspetta che ci sia un unico punto di ritorno dalla funzione. Aumentare il numero di ritorni diminuisce il livello di mantenibilità

Unità di misura

Mai spiegare l'unità di misura delle variabili numeriche

Gotchas

Il termine gotchas è una forma pecoreccia ritengo di origine americana della affermazione “I’ve got you”. Si traduce in un “Ho te” o anche “Ti ho preso” o “Ti tengo d’occhio” che nell’ambito della programmazione è evidentemente riferito al grande demone del programmatore che è il baco. Mai commentare una sezione di codice che potrebbe presentare problemi o potenziali bachi, perchè favorire così tanto il nostro successore ?

Documentare variabili

Mai documentare la dichiarazione delle variabili. Queste infatti sono il cuore della logica e gestiscono il flusso dei dati. Le variabili numeriche hanno range di valori legittimi, decimali, base tempi di aggiornamento, formattazione di visualizzazione, valore di inizializzazione. Spiegare una variabile significa spiegare l'algoritmo e il programma, state alla larga!

GOTO riesumato

La programmazione a GOTO contrastata dai puristi informatici è il pane quotidiano di chi programma in assembler. Si può scrivere codice strutturato anche utilizzando i goto ma se si abusa allora è tutta un'altra storia.

```
x ( char * pointer )
```

```
{
```

```
loop:
```

```
    if( x++ < 10)
```

```
        goto loop;
```

```
}
```

Offusca un sistema embedded

Nello sviluppo di sistemi embedded in **real-time** il controllo delle temporizzazioni è fondamentale. Più il sistema diventa complesso e più le sezioni di codice aumenteranno. Il tempo di esecuzione diventerà sempre meno

marginale.

Loop bloccanti

Inserisci nel codice cicli operativi bloccanti. Ad esempio evita di utilizzare gli interrupt per definire i timer che utilizzerai nel ciclo e usa sistematicamente delle funzioni di delay. Spostando o copiando sezioni di codice il software/firmware avrà risposte diverse.

```
es:delay(int sec){for ( i=0;i<sec;i++);}
```

Watchdog fantasma

Inserisci il watchdog in un interrupt timer. Se il loop principale del firmware per qualche motivo entra in dead lock il programma non se ne accorgerà.

Watchdog a schiovere

Anzichè avere un unico reset watchdog come sarebbe auspicabile, sparpagliali su tutto il codice a casaccio. Chi viene dopo di te, difficilmente prenderà la sonda dell'oscilloscopio per verificare i tempi. Se si convince che sei molto stretto con i tempi le modifiche diventano più complicate.

Awards dell'offuscamento del Codice

Se siete dei programmatori Java indubbiamente conoscerete questo link. Ci sono molti programmi commerciali che adottano tecniche varie di offuscamento del codice. Nel mondo è in atto una vera e propria guerra del

software. Il fenomeno della pirateria informatica è poca cosa in confronto allo scontro fra le case madri come Google, Apple e Microsoft. Tutte indistintamente si copiano programmi compreso i banchi... Per limitare questi furti esistono una serie di tecniche e di programmi che hanno lo scopo di modificare il codice sorgente generando rendendolo difficile da gestire. (link) Ma c'è anche chi ne ha fatto diventare una vera e propria arte. In questo sito (link) c'è la possibilità di partecipare ad un concorso annuale dove i programmatori si sfidano nella gara di offuscamento del codice. Alcuni sono veramente incredibili...

Vertigo

Max Eisenhardt è un personaggio fantastico che Stan Lee aveva creato mettendo insieme nella stessa figura un anti eroe e un super eroe creando l'ennesimo personaggio della famiglia Marvel con evidenti disturbi bipolari. Al nome di battaglia **Magneto** ben si addicono i super poteri posseduti. Lui è infatti il signore del magnetismo in grado di manipolare il metallo e di generare campi magnetici dalla incredibile forza ed intensità. Probabilmente non sapeva che il mediatore dell'interazione elettromagnetica è il fotone e io invece ancora adesso mi chiedo come si possa giustificare con una questione di prospettiva, l'antitesi logica della sua natura.

In verità i **signori** del magnetismo esistono davvero anche se non sono dei super eroi. Sono persone schive e poco socievoli, vivono all'interno dei centri di ricerca internazionali passando il tempo dentro stanze impegnati in studi di altissima ingegneria. Vivono anonimamente sul debole confine che delimita la materia dall'energia in quel universo parallelo che si chiama meccanica quantistica. Questa che leggerete sarà così una storia sull'elettromagnetismo è su un ingegnere alle prese con una manciata di bit di errore.

L'azienda

13 Maggio 2003

Milano centro direzionale Exenia

Era un giorno umido e sono all'interno di una stanza di attesa, in una palazzina anonima nel hinterland milanese. Una Milano strana a dir la verità perennemente depressa dalla crisi, ma sempre vivida e tumultuosa.... Attendevamo noiosamente seduti su sedie che io chiamavo di appoggio temporaneo.

Le sedie di appoggio temporaneo non hanno una forma definita ma, per intenderci, sono quel tipo di sedia che dopo due minuti ti provocano un dolore spontaneo che ti parte dalla seconda vertebra della spina dorsale e arriva su per la colonna sino ai nervi della cervicale. In quel momento stavano per scadere i due minuti, così mi alzai piuttosto nervosamente, mentre il mio collega Limite **stoicamente** tentò di rimanere lì seduto, pensando di resistere cambiando posizione.

“Ma quando arrivano ?” dissi

“E non so... aspettiamo ancora un attimo poi ... torniamo dalla segretaria.... che ti devo dire”

Nella stanza c'era un tavolo in formica, centrale, grigio, che poteva ospitare una ventina di persone. Aveva le gambe cromate che inutilmente specchiavano il grigio generale della stanza. L'unica nota di colore era il pavimento in linoleum, color alga fangosa che nelle parti di

passaggio prendeva un tenue color sabbia. Sul muro una lavagna bianca dove si intravedevano le **scritte labili** di un solerte tutor che evidentemente non aveva molte cose da dire, per la quantità di segni o invece, al contrario, ne aveva tante da non aver tempo per scrivere.... chi lo sa ?

Un po' impaziente tirai fuori il portatile aziendale, un Acer con Windows Me, che avevo battezzato Windows stramuort. Tirai fuori anche l'alimentatore esterno perché la batteria aveva già da tempo esalato gli ultimi elettroni. Lo misi sopra il tavolo solo dopo aver trovato una presa comoda per accenderlo. Dopo poco sentii bussare e alzai lo sguardo sopra l'lcd del portatile.

Finalmente erano arrivati. Vidi Firelli insieme ad altre tre persone che erano a me sconosciuti in quel momento.

"Dott. Kirkegard, dott. Limite vi presento il responsabile aziendale Loriani e l'amministratore delegato Exenia ing Rodari...il dott Spalletti e il dott Rubini" -dottore ? ma che sta dicendo... e poi cosa è questa storia- Pensai stringendo le mani agli sconosciuti, non più sconosciuti. Avevo già un inequivocabile sensazione di nevralgia del trigemino, che mi viene a tutti gli incontri imprevisti di questo tipo (alieni inclusi).

Che cosa era successo perché arrivasse l'amministratore delegato? Ora avevamo alcuni problemi nel progetto ma ero convinto di risolverli in sedi opportune...

"ci seguite ?" Firelli allargò il braccio mentre sembrava indicare una stanza con il patibolo.

"certo..." dicemmo corali all'unisono.

Camminammo lungo lo stretto corridoio interno. Di lato, sulla sinistra una sequenza interminabile di uffici amministrativi che avevano la finestra illuminante sull'esterno. Sulla destra altri uffici vari, ma questa volta senza finestra. Attraversammo un piccolo andito e sulla sinistra c'era un salone molto grande con decine di **tecnigrafi** allineati, molti erano coperti da un telo ed erano evidentemente in disuso. Mi sembrava però, una stanza molto vissuta e in quel mentre, immaginavo il via vai che ci doveva essere stato fra quei disegnatori meccanici che muovevano con la destrezza di un acrobata i leggeri pantografi sul tavolo da disegno.

C'erano anche dei **regoli** calcolatori in una teca laterale, che testimoniavano la presenza di persone diciamo non proprio di primo pelo. Sulla destra c'era invece una scala circolare ampia e noi evidentemente dovevamo salire ai piani superiori. Raggiungemmo il piano e c'era ancora un corridoio sulla sinistra molto lungo e buio, ma noi li ci fermammo.

Firelli fece un cenno a Rodari a cui rispose annuendo. Firelli allora mise le mani nella tasca per prendere un corposo mazzo di chiavi ed entrò nella seconda porta. La stanza era buia e Firelli accese le luci. Dall'oscurità venne fuori una sala molto ampia con un tavolo circolare da convegno del G8 per intenderci. C'erano una serie di stanze con i vetri oscurati e ci accomodammo in una di queste, intorno al tavolo centrale.

Firelli prese l'iniziativa

L'errore

“Bene dott Kirkegard , come può immaginare le cose sono **leggermente** cambiate...” aridaje dottore ?! Lo guardavo perplesso...

“Cioè in che senso...”

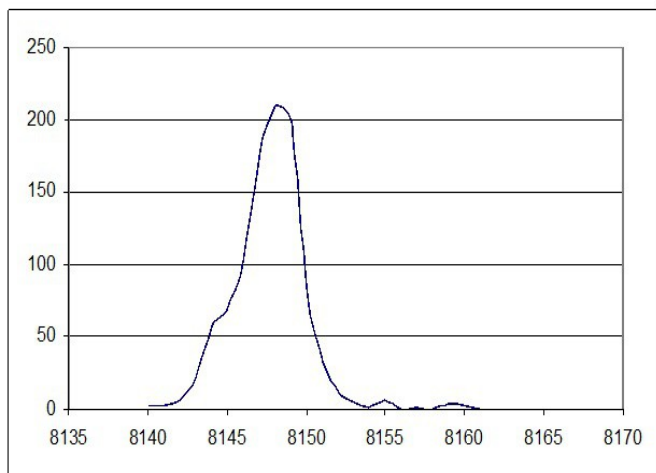
“Nel senso che la relazione presentata sulle schede non supera i nostri requisiti...”

“Non supera i requisiti? Ma che sta dicendo, Firelli, guardi che le specifiche progettuali avevano definito delle caratteristiche che le schede stanno rispettando ...e questo documento lo dimostra...”

“sì ... sì ... questo ce lo siamo già detto via telefono. Il problema è che vede... non so come spiegarglielo... questo è un sistema piuttosto complesso e diciamo che.... con il tempo abbiamo avuto la necessità di confermare i vincoli, che non possono essere oltrepassati.” Intanto si attivò il tritatore gonadico tipico nelle situazioni di questo tipo.

Mentre parlava tirò fuori una cartellina e l'appoggiò sul tavolo. La aprì e incominciò a sfogliare alcune pagine di un documento. Era la contro relazione di un **consulente** esterno che analizzava le caratteristiche della mia scheda e la sua applicabilità nel sistema. Guardavo basito e molto infuriato perché pensavo che stavano semplicemente tentando di scaricarmi con dei pretesti, poi Firelli tirò fuori la mia relazione sugli errori del sistema elettronico.

Nella relazione fatta, avevo incluso una piccola sezione con le curve degli errori sulla misura del canale analogico. Le curve erano delle **curve gaussiane** prese da dati sperimentali (sul campo) dove si notava che il 90 per cento degli errori erano in valore assoluto di due bit al massimo e considerando che il convertitore era un delta sigma a 16 bit mi sembrava un risultato eccezionale.



Error.jpg

“Vede ingegner Kirkegaard, guardi noi possiamo essere disponibili a venirci incontro su tutto, c'è però un elemento bloccante che è diventato invalidante nella sua scheda ...”

“... e cioè?” dissi mentre nello stesso tempo le certezze che mi avevano convinto di portare avanti il progetto sino a quel punto si sbriciolarono sotto i miei piedi e adesso ero in piena caduta libera...
“vede...” indicò la **base** della curva gaussiana
“sono questi quattro bit di errore”

“Mi scusi dott Firelli ma questa è una condizione statistica **improbabile**... ora mi sembra che” non feci in tempo a concludere l'arringa

“Ingegnere... mi permetta.. improbabile, significa poco probabile e quindi non impossibile e noi semplicemente le dico, questo evento non possiamo **tollerare** che accada.. in nessun caso... qui se vuole troverà le risposte del **perché**...” Così prese un altro piccolo plico di una decina di pagine e me lo passò. Il sangue smise di defluire agli organi un po' per la caduta libera un po' per un inizio di arresto cardiaco.

“Sì però mi scusi.. parliamo di cose concrete... ma allora il progetto termina qui ?”

“Diciamo di sì ” prese la parola Rodari con voce stentorea.

“però, mettiamola in questo modo, io oggi...” intanto guardò l'orologio

“...darò il benestare per la chiusura della commessa con la vostra azienda. Ma come può immaginare c'è tutto un iter burocratico da gestire che confermerà la chiusura solo.... vediamo....la prossima settimana.”
e aprì un moleskine vissuto sulla pagina segnata centrale.

“.. In via eccezionale posso darle questa come dire....

chance. Se lei entro martedì prossimo, riuscirà a portarmi questa curva con gli errori di misura... dimezzati allora diciamo che possiamo... riprendere in considerazione il discorso... ecco di più non posso fare...”

Disse queste cose con una naturalezza di chi capisce che non è possibile ottenere certi risultati ma che però sa che ogni 2000 anni qualcuno fa miracoli...

“ora però vi chiedo scusa ma ho un impegno urgente...”
Si alzò velocemente assieme ai silenziosi Spalletti e Rubini che si tennero ben alla larga dal fare commenti che avrei potuto interpretare come ostili
Avvicinandosi mi strinse la mano dicendomi
“Buona fortuna”.

Non sapevo cosa dire, sapevo solo che in quel momento mi chiedevo come avevo fatto ad arrivare lì in quel posto mentre invece non fosse stato meglio risiedere in un'isola caraibica a fare il barman in un villaggio turistico di ricchi americani bevendo latte di cocco e ballando il merenge.
Così mentre tornavamo a casa con il mio socio Limite parlammo molto sulle cose da fare. C'erano un paio di commesse in sospeso e l'inizio di un paio di produzioni. Io avevo da ultimare due attrezzature di collaudo.

Ritorno a casa

14 Maggio 2003
Roma(provincia).

Laboratorio elettronico Azienda ELMel del gruppo Selex (Finmeccanica).

La mattina dopo entrai in ufficio portai in amministrazione le spese per la trasferta esterna, due chiacchiere con la segretaria poi ritornai al laboratorio.

Avevo comprato un bancone nuovo con due scaffalature medie che avrebbero sostenuto gli strumenti di misura.

Un **frequenzimetro** di origine militare, un generatore di funzioni.

Un oscilloscopio digitale della Tektronix. Un oscilloscopio digitale della LeCroy prestato da una azienda di un amico, -non abbiamo abbastanza finanze per comprarne uno ma provarlo è come guidare una Ferrari - .

Un tester della Fluke con i puntali segnati (perché c'è sempre qualcuno che in azienda te li frega...). Un alimentatore variabile di origine cinese con il display a lancetta che non ha mai funzionato. Un doppio alimentatore un po' più serio, un alimentatore in corrente da 20 ampere e un simulatore di carico.

Una cassettera con due cacciaviti a stella due a taglio fine, una tronchesina, una spellafili, entrambi della Pastorino e anche questi segnati per i soliti motivi di prima. A fianco la sezione di **reworking** con uno sverniciatore, il saldatore della Weber anche quello prestato ma poi precettato per uso capione. Il saldatore era di tipo termostato ma anche quelli a temperatura fissa non andavano male.

Stagno fine, stagno grosso, una boccetta di solder, a

fianco una supporto per il montaggio di componenti TH e una lente con luce al neon per faticare meno con le prime schede smd. Nella cassettera più grossa 4 pine professionali per la crimpatura con sopra un paio di tubi con una serie di ruzzole di cavi per i cablaggi.

Di lato avevo la parete con i gruppi di cassettiere per contenere la serie di resistenze TH, la serie di condensatori ceramici, elettrolitici, la serie 74HC ma non solo. Con i porta rullini delle macchine fotografiche oramai in disuso avevo fatto una cassettera per i componenti smd. Un gsm della Siemens TC35, per le prove di trasmissione dati.

Un fornello ad aria calda termostatoato, per piccole produzioni e per il **baking** dei componenti.

Accesi il computer e scaricai le email, entrai nella cartella della Exenia per cercare tutti i messaggi che parlassero di precisione delle misure analogiche. Dovevo fare delle find dentro un centinaio di email.

Poi ripresi un po di lavoro pregresso implementando la calibrazione a 10 punti con spezzata della ntc

$$T = P[i][1] + \frac{(P[i-1][1] - P[i][1])}{P[i-1][0] - P[Idx][0]} * (T - P[i-1][0])$$

Tornai a casa dove vivevo da single oramai da un paio d'anni. Accesi la televisione con molta curiosità, perchè la guerra nel golfo era davvero iniziata così come aveva previsto Firelli e noi evidentemente eravamo davvero in ritardo... Saddam era stato deposto da poco e le operazioni di invasione erano terminate. Ascoltavo le notizie

delle operazioni militari sempre con una certa apprensione ma nel mentre però ripensai a quel giorno, quando tutto ebbe inizio ... mesi prima, molti mesi prima...

Vertigo II

Mit der Dummheit kämpfen Götter selbst vergebens.

Friedrich Schiller (1759 - 1805) e' stato un importante preromantico poeta e filosofo, tedesco. Ha prodotto significative opere teatrali (La pulzella di Orlean's) ed e' anche notoriamente additato come l'autore della frase citata all'inizio di questo post. La traduzione più' o meno dice così' "Contro la stupidità' neanche gli dei possono nulla" la frase evidenzia lo stato di frustrazione che l'autore a trasferito da se stesso agli dei, di fronte alla devastante stupidità' che l'uomo e' in grado di dimostrare. Isaac Asimov (1920 - 1992) nel 1972 ha scritto un romanzo **fantascientifico**, da lui reputato il migliore mai scritto, con il titolo "Neanche gli dei" (The Gods Themselves)che ha vinto numerosi premi letterari (Hugo, Nebula). Il libro e' diviso in tre parti principali, originalmente pubblicati in una rivista come tre storie consecutive. La trama principale narra di una cospirazione da parte di alieni che abitano un Universo parallelo, con leggi fisiche diverse da quelle note, che avevano l'intento di ottenere energia trasformando il sole in una supernova. I titoli delle tre parti formano, tutti insieme, una frase unica.

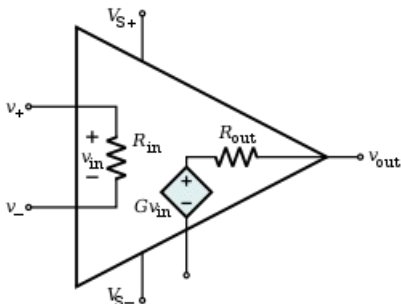
1 -Contro la stupidità.....

2 -Neanche gli Dei.....

3 -Possono nulla?

Gli Operazionali

Gli operazionali sono un po' la base dell'elettronica analogica "tradizionale".



250px-Op-Amp_Internal.svg.png

Le caratteristiche degli operazionali sono essenzialmente:

- guadagno in tensione infinito
- resistenza di ingresso infinita
- resistenza di uscita nulla

la premessa di queste ipotesi sono:

- lo stato equipotenziale degli ingressi
- le correnti di ingresso nulle

Ma le regole d'oro degli amplificatori operazionali sono queste:

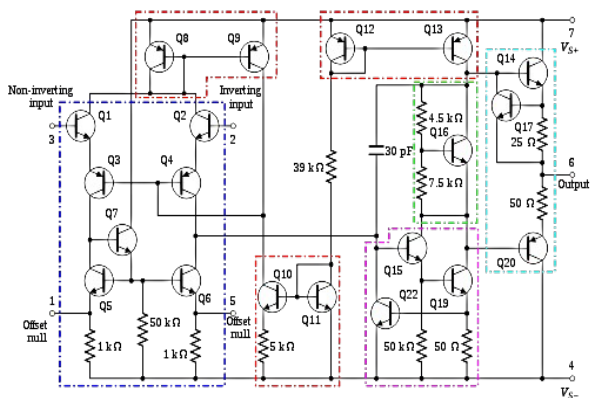
1 - L'uscita farà di tutto per cercare di portare la differen-

za di **tensione** degli ingressi pari a zero.
2 - Gli ingressi non pilotano nessun carico.
Queste cose si sveleranno con un po' di esperienza.

L'amplificatore operativo si chiama così perchè è in grado di eseguire le operazioni logiche:

- sottrattore
- invertente
- non invertente
- inseguitore
- sommatore
- integratore
- derivatore
- comparatore
- strumentale (nelle applicazioni degli strumenti di misura..)
- Trigger di Schmitt
- Gyration di induttanza
- Zero voltage detector
- convertitore di impedenza negativa
- raddrizzatore
- logaritmo
- esponenziale
- separatore

Ma noi che sviluppiamo applicazioni embedded con questa mania di capire come sono fatte le cose non ci accontentiamo delle applicazioni di un operativo; così muniti di micro fresa seghiamo l'integrato e andiamo a vedere come è fatto il silicio



e così con una **manciata** di resistenze e transistor messo su una basettina e mi monto il circuito integrato... per scoprire dopo una giornata che il circuito non mi funziona... acc

...noi però avevamo una storia da portare avanti, nel post precedente eravamo quasi alla fine, ma in questo faremo un flashback per raccontare come sono iniziate le cose e per spiegare alcuni punti sospesi. Così ritorneremo indietro nel tempo ...

11 Novembre 2002

Misi a posto l'ufficio come solito cercando di distribuire sparso **caos** ordinato, necessario ad evitare la fastidiosa impressione che abbiamo quando guardando l'ordine in una stanza pensiamo che non vi lavori nessuno. Quella era la prima visita degli ospiti che faceva seguito

ad un carteggio via e-mail per un progetto che sembrava essere piuttosto importante. Nel frattempo stavo leggendo delle documentazione per un altro progetto che stavo ultimando. Era il controllo del **flusso magnetico** di un magnete in un apparato medicale.

In un post precedente abbiamo trattato i limiti della logica umana

considerando la capacità della mente stessa di creare paradossi senza poterli capire. Per giustificare questi **buchi** della logica

si è cercato di creare alternative chiamate logiche polivalenti

molto complicate. Un giorno feci un'intervista ad un fisico

parlammo di tante cose e alcune mi rimasero impressione...

Domanda - Quale difficoltà hanno dovuto affrontare i fisici quando hanno scoperto che esiste un **universo** dentro l'atomo ?

Risposta - Guardi, il problema è complicato per le diverse

sfaccettature che presenta. Quando dalla fisica degli atomi

si è passati a quella delle componenti subatomiche, oltre

alla difficoltà di trovare dei modelli coerenti,

("la teoria del tutto" chiarita da Borrow nell'omonimo libro) si sono aggiunte problematiche relative agli stessi limiti della logica.

Il panorama scientifico a quel tempo era come dire, an-

cora, profondamente condizionato dal meccanicismo cartesiano e dal positivismo francese. **L'elettromagnetismo** in verità era considerato più una scienza esoterica utile a risvegliare i morti che non ad un fenomeno fisico. La società umana aveva aspettative sovradimensionate per queste scoperte e sprofondò in un medio evo dal quale non ne è più uscita. Nell'immaginario collettivo fu un proliferare di idee alchemiche dove tutto poteva diventare possibile; il successo dato dalla visione di una realtà alternativa come quella creata nella letteratura fantascientifica ne fu la dimostrazione.

Per contro nella realtà quotidiana all'interno dei laboratori di ricerca i fisici dovettero affrontare problematiche di difficoltà estrema a partire dagli stati plurimi delle componenti atomiche. Per questo nacque il caos nel tentativo di descrivere quel mondo.

A ben vedere con questi presupposti sembra sia legittimata la naturale refrattarietà che riscontriamo quando si chiede ad un fisico di **rispondere** ad imbarazzanti domande sulla natura del fotone o dei campi elettromagnetici.

Domanda - diciamo che intuisco... può farmi un esempio ?

Risposta - Prendiamo la **legge di Ohm**. Questa formula è un efficacissimo mezzo che ci permette di descrivere con una semplicità infantile (tanto che posso capirla anche io che sono un fisico), i fenomeni elettrici che spesso trattate nel forum di questo portale, aiutati da sommi e valenti guru nella disciplina dell'elettronica. Questi fenomeni fisici, possiedono tutto tranne che la semplicità della legge.

Cerco di spiegarmi...La **corrente** che viene vista come parametro di una funzione lineare nella legge appena citata, descrive l'effetto di un movimento di elettroni che saltano di atomo in atomo, lasciandosi dietro delle buche quantici chiamati lacune. Agli occhi di uno spettatore molecolare si vedranno lacune ed elettroni che si muovono con versi opposti...

Domanda - sin quì penso di esserci...

Risposta - Questo passaggio di elettroni ha però anche altri effetti importanti. Sempre per semplificare, a livello subatomico accade che questo movimento crea una sorta di vibrazione nell orbitale dell'atomo. Se aumenta questa vibrazione l'effetto viene percepito con l'aumento di temperatura dell'oggetto.

Domanda - Il calore è un fenomeno quantico ?

Risposta - Vede le dico una cosa che la sorprenderà. In realtà il calore potrebbe essere inteso meglio come fenomeno ottico che non come fenomeno "meccanico". Perchè quello che accade a livello molecolare e che, la vibrazione atomica descritta pocanzi, provoca l'emissione nello spazio di un componente base dell'iterazione elettromagnetica e della luce che è chiamato "fotone".

Domanda - ... quindi ?

Risposta - Raggiunto questo livello incominciarono problemi logici suddetti, perchè il fotone possiede caratteristiche che la logica così come la intendiamo, non riesce a gestire. Il fotone infatti è definito a massa nulla,

indistruttibile e ha caratteristiche particellari o di onda elettromagnetica a seconda di come lo prendiamo in considerazione, e da qui

il paradosso apparente. Consideri che questa antitesi ha origini molto lontane, la teoria corpuscolare è stata data da Newton mentre quella

ondulare da Christiaan Huygens olandese contemporaneo di Newton nel 1600, un esimio e sconosciuto scienziato fra l'altro molto vicino a Cartesio.

Domanda - e come ne sono venuti fuori ?

Risposta - Quello che ha permesso di interpretare la natura del fotone per semplicità ed efficacia è stata la rivoluzione di pensiero creata da Einstein, che insieme a Plank definirono per primi le caratteristiche principali di questo nuovo elemento della materia. La teoria della Relatività fu la chiave...

Domanda - ... ha detto semplicità ?

Risposta - ... be diciamo semplicità formale per i fisici... per l'uomo comune bastava la parola relatività...

Arrivarono due persone distinte, una era l'ingegner Firrelli e l'altra il dott Cortopassi, uno era un ingegnere elettronico l'altro un fisico. Notoriamente fra ingegneri e fisici non scorre buon sangue ma evidentemente la causa comune aveva permesso loro di lasciare indietro le antiche schermaglie universitarie.

“Buon giorno, Kirkegaard” dissi allungando la mano

“Buon giorno, Firelli e questo è il mio collega Cortopassi”

“Bene ... piacere... ma.. accomodatevi prego... posso offrire un caffè ?”

“No grazie ... abbiamo già fatto colazione... troppi caffè sa' mi rendono nervoso...” disse lui con una banalità disarmante...

mi rivolsi a Cortopassi che evidentemente aveva meno pudore

“Io... lo accetto volentieri” disse lusingandomi.
“Avete trovato traffico ?” nel mentre andai nella stanza contigua per preparare due caffè... perchè a me non rende nervoso. Diciamo che questo è quello che credo ma non sono mica sicuro.
“A dir la verità non troppo, questa mattina siamo partiti un po' più tardi e così il traffico della tangenziale l'abbiamo saltato”

“Veniamo a noi ?”

“Sì certamente...”

“Bene.. il progetto come le ho anticipato via mail fa parte di una commessa europea di protezione passiva navale dalle mine marine.”

“ Roba militare quindi ?..”. dissi con un tono di uno che

evidentemente non gradiva molto la cosa. “Si in un certo senso... diciamo però che sino ad ora era un problema prettamente militare ma in questo caso l'applicazione è diventata invece **civile**. Insomma la “prossima” guerra nel Golfo (Golfo Persico ndr) è un po’ la causa di questa commessa. Ora diciamo che la situazione attuale mondiale ha creato una domanda maggiore di petrolio e questo ha creato di conseguenza l’inevitabile intensificarsi del traffico petrolifero. Le vie commerciali marittime verso il medio oriente sono diventate strategiche e soggette a pericoli attacchi. Mi spiego meglio....”

“Exenia ha un brevetto internazionale navale per la protezione passiva delle mine magnetiche. Circa sei mesi fa è stata contattata da una multinazionale navale Olandese per soddisfare l’appalto di costruzione per una decina di supepetroliere, che dovrebbero trasportare il greggio in tutto il mondo. Ora consideri però che il 25 per cento del traffico petrolifero mondiale passa dal golfo Persico allo stretto di Ormuz e il golfo di Oman, per arrivare poi sull’oceano indiano... e queste sono quindi percorsi diciamo obbligati da percorrere...”

“25 per cento ?” chiesi con curiosità.

“Si diciamo che nello stretto, barile più barile meno, ci passano.... intorno ai 5/6 miliardi di barili l’anno.... considerando che attualmente abbiamo un corrispettivo di circa 50 dollari a barile... più o meno siamo intorno 250/300 miliardi di dollaril’anno... per farle capire di cosa stiamo parlando consideri che questo petrolio è

circa il 20% del fabbisogno mondiale.. Valuti un attimo lei quanto potrebbe essere ritenuto **importante** quel piccolo tratto di mare visto che ogni giorno ci passano quasi un miliardino di dollari di petrolio..”

- Il prodotto interno lordo americano è di circa diecimila miliardi di dollari l'anno e quello italiano è di circa mille miliardi di dollari (dati 2002)-

Aprì una cartina mondiale nella agenda e indicò il budello di acqua a sinistra della penisola arabica davvero molto simile ad un sifone di scarico. “Come può immaginare il disastroso scenario politico attuale ha reso queste tratte navali estremamente pericolose per tanti fattori e uno dei tanti è proprio quello delle mine magnetiche.” “non sono ferrato però immagino il problema...” “Questa situazione ha fatto tremare un po’ di tavolini in giro per il mondo e ... diciamo così... il risultato è che sono partite un paio di commesse **internazionali** per migliorare la protezione passiva delle navi. Questa commessa avrebbe portato altri vantaggi come lo svecchiamento delle petroliere che sono diventate davvero dei giganteschi bidoni galleggianti... ma la cosa poco interessa a noi in questo momento...”

Le mine in sintesi

“Ora le spiego cosa sono le mine...”

“Le mine nascono come efficace strumento militare negli anni della seconda guerra mondiale. Si suddividono

in due tipologie principali: le mine terrestri e quelle marine. Le mine in entrambi i casi erano, di norma, innescate a contatto (urto) da piccoli compressori che rompevano fialette di acido che attivavano il composto esplosivo. La maggior parte delle mine ad ormeggio marine erano di questo tipo. Nel tempo, sono state prodotte mine di diverso tipo, le più efficienti venivano denominate ad **influenza magnetica**.

Questo tipo di mine, erano innescate non più ad urto ma dalla presenza di una massa metallica, di un rumore o una variazione della pressione dell'acqua. La Marina poi impiegò varianti di queste mine, definite "a temporaneo galleggiamento", le quali, dopo un certo periodo d'operazione, si autodistruggevano. Questo perchè le mine sono un efficacissimo strumento di offesa, ma hanno pesanti effetti collaterali, anche su chi le ha messe."

"Le mine magnetiche erano innescate dalla **forza** magnetica generata dallo scafo ferroso di una nave. In cosa consiste questa forza magnetica ?"

...

" Lei immagini lo scafo di una imbarcazione come se fosse un enorme conduttore immerso nel grande campo gravitazionale terrestre, che, più o meno dovrebbe, corrispondere a circa 40 uTesla. Il movimento del conduttore crea un campo magnetico che si interseca con quello terrestre e che, sommandosi, genera correnti nelle bobine di innesco della mina. Questa minuscola forza magnetica, tanto per farle capire è normalmente di circa 50 milliGauss (consideri che 1 Gauss = 100 μ Tesla) e produce una corrente sulle bobine di qualche microam-

pere, resi sufficienti all'attivazione dell'innesco e della esplosione.”

“Ora una mina magnetica è di diversi fattori più distruttiva di una mina da contatto, perchè l'esplosione non avviene infatti - mi scusi per la ripetizione - ... a **contatto** dello scafo, ma con questa tecnologia potrebbe avvenire anche a diverse decine di metri. Ora considerando che una delle proprietà dell'acqua e l'incomprimibilità, lo spostamento violento della massa d'acqua che intercorre fra la mina e lo scafo, amplifica l'energia emessa dall'esplosione. L'energia si trasforma in pressione, distribuendosi sulla superficie dello scafo presente sotto la linea di galleggiamento, causando danni permanenti a tutta la struttura interna compromettendo la stabilità. Ma consideri che anche solo la semplice oscillazione, potrebbe affondare l'imbarcazione grazie al violento spostamento del baricentro della nave.”

“Molto affascinante...” dissi tentando di stargli dietro... ma avevo ancora poche risorse...

“Questo vuol dire che una mina con questa tecnologia potrebbe avere una carica di tritolo relativamente piccola per affondare un incrociatore...I vantaggi di queste mine erano enormi rispetto quelle a contatto, ad esempio, un altro vantaggio era che mentre le mine a contatto dovevano trovarsi esattamente nella direzione della nave, quelle magnetiche solo in prossimità, allargando così il loro fatale raggio operativo d'azione”

Firelli continuò l'escursus storico...

“Ora torniamo un attimo alla seconda guerra mondiale. La Royal Navy, nel tentativo di contrastare questa tecnologia (di origine tedesca), sperimentò varie apparecchiature, tra le quali un sistema detto "degaussing" e fece una piccola pausa da attore... che si imparano nelle tecniche di marketing

“... il quale fu prima sperimentato sull'incrociatore Manchester e poi utilizzato sulla nave di linea la Queen Elisabeth sin dal 1940. Da lì poi la US Navy ha fatto le sue prove e i suoi prodotti installandoli sul suo parco macchine.... In Italia, siamo partiti in sordina ma piano piano siamo riusciti a dire la nostra perchè abbiamo sviluppato una serie di avanzatissimi prodotti con questa tecnologia grazie ad alcuni brevetti esclusivi... Tutto chiaro ?”

“Tutto chiaro è un po' troppo....mi sembra di capire allora che la Exenia produce questo sistema di ... di degaussing...” Facendo finta di capire che cosa è quando a malapena conoscevo la funzione presente in qualche qualche monitor (a quel tempo erano ancora prevalenti quelli con il tubo catodico) che aveva un pulsantino con sopra scritto quel termine.

Degaussing

“Esattamente dottor Kirkegaard...Esattamente... Bene ora un accenno alla tecnica di Degaussing.” -non era ancora finita - "La nave, come abbiamo detto poc'anzi,

può essere modellizzata come con un grosso conduttore polarizzato che crea un campo magnetico. Il nostro scopo è quello di annullare questo campo magnetico della nave con delle correnti chiamiamole inverse per far diventare diciamo “invisibile” lo scafo ai sensori delle mine.”

-un incorciatore invisibile... perchè non chiedono a Copperfiled ?-

“Questo sistema detta in parole povere consiste “semplicemente” nel cinturare lo scafo in una gabbia di **Faraday**, con una serie di vere e proprie putrelle di ferro dove si sarebbe fatta scorrere della correntina.... Le applicazioni della gabbia di Farady sono di due tipi una statica e l'altra elettromagnetica. Se noi induciamo un passaggio di cariche controllato, sulla superficie dello scafo creiamo non solo un filtro magnetico dall'esterno verso l'interno ma anche dall'interno verso l'esterno, annullando quindi il campo creato dalla nave stessa, che è la causa di innesco delle mine.”

Questo effetto fisico è possibile grazie anche ad altri fattori come ad esempio l'effetto **pelle** dei conduttori che è fondamentale per capire la distribuzione delle cariche sulla superficie dello scafo.

-...urca-

“Ora non sono molto ferrato in materia ma viste le masse in gioco le correntine ... ritengo non siano proprio correntine...” chiesi senza più fiato

“ Dice bene... dott Riva.. dice bene ... Diciamo che le tecnologie tradizionali sono intorno al migliaio di amperre... però stia tranquillo ... con la nostra tecnologia sia-

mo abbondantemente sotto i cento...”

“aaa...” adesso sì che sono tranquillo... e Firelli continuò ignorandomi....

“Il problema di questo sistema è che le correntine devono creare un campo magnetico di segno opposto alla differenza di campo che verrà sentita dalla mina...e quindi le piccole discrepanze del controllo potrebbero essere fatali.. e insomma ... per quello che ci riguarda.... nel sistema ci sono ipotesi di guasto o errori che devono essere resi a probabilità nulla....”

Ero davvero curioso di sapere quale era la tecnologia misteriosa...

“se può dirlo ... mi chiedevo quale tecnologia....?” chiesi con misurata curiosità

“a certo.. non è un segreto... questa tecnologia diciamo non ha un nome specifico diciamo che si basa su nuovi materiali chiamati **superconduttori**. Questi nuovi materiali permettono cose.... prima impensabili... anche se diciamo si dicono molte storie un po' esagerate sull'argomento...”

“Cioè scusi, in che senso...” Rimase un po' perplesso perchè evidentemente non voleva parlare troppo comunque visto l'accento e la forzata domanda... “...be ... allora le dico solo che questo tipo di esperimenti associati a queste tecnologie diciamo sono un po' particolari. Non voglio spaventarla...”

“Fantasie?... scusi ma io sono proprio digiuno... di cosa

sta parlando?”Fino ad ora ero solo preoccupato... ma francamente adesso incominciavo a non capire se era solo preoccupazione...

“... non ... non capisco” dissi ancora.

Philadelphia Experiment

Firelli si girò verso Cortopassi “Allora guardi tanto evidentemente prima o poi lo verrà a sapere. Questo sviluppo tecnologico ha diciamo avuto un impulso da un celebre... test. ... un test navale americano avvenuto 28 ottobre del 1943 chiamato Philadelphia Experiment...”

Bene a breve parleremo anche dello sbarco degli Ufo.. l'area 51... i cerchi nel grano.

“... ma mi scusi ... ma quello se non ricordo male ma non era un'esperimento per teletrasportare le navi - avevo visto superquark l'altro giorno - ... che c'entrano le mine ?!?!”

“Si .. si c'entrano c'entrano...guardi come immaginavo anche lei è stato un po' condizionato da quel delirio collettivo. Quello del **teletrasporto** è una ovvia stupidaggine presa in prestito dai telefilm di star trek. C'erano teorie meno fantasiose ma ugualmente sbagliate che sostenevano fossero in atto tentativi di applicare studi sui fenomeni elettromagnetici e la loro interferenza sul campo gravitazionale. Questo faceva seguito alla teoria appena sfornata di Einstein sul **campo unificato** e sulla sua capacità di curvare la luce.

Questi studi sembrano dimostrare che qualsiasi oggetto immerso in campo magnetico di una particolare intensità potrebbe svanire alla vista di un osservatore perchè la traiettoria della luce intorno al campo non sarebbe più rettilinea ma si incurverebbe. Fu montata l'ennesima grossa bufala mediatica utile solo a creare l'ennesimo fenomeno allucinatorio collettivo, come il riuscito sbarco degli alieni nell'area 51."

- *eccolo li... che ti avevo detto...* - ma non aveva finito...

"In realtà la marina americana stava facendo solo delle prove di degaussing e stavano semplicemente smagnetizzando l'incrociatore militare... (Nell'ambiente noi tutti sappiamo cosa è successo ma ad esempio i ragazzi di Wikipedia ancora adesso non lo sanno) le nebbie nel porto di Philadelphia fecero il resto creando un finto alone di mistero. Qualcuno pensò bene però di sfruttare questa sorta di delirio schizofrenico collettivo in altri modi con storie di altre dimensioni e di marinai entrati in simbiosi con la nave stessa..." disse sorridendo...

La proposta

Si chinò su una Samsonite molto lussuosa, solo dopo capii che era una rara valigia diplomatica e l'apri. Tirò fuori uno schema in formato A0 che copri tutta la scrivania. Era pieno di blocchi rimandi con tanti quadrati interconnessi fra di loro e molte correzioni manuali che evidentemente dovevano descrivere il sistema di degaussing della Exenia. "Questo grafo è L'Exenia experiment e lei si occuperà di questa sezione" disse sorridendo. Indicò con un dito un piccolo quadratino attaccato diret-

tamente alla barra della cintura dello scafo. “Lei dovrà leggere lo stato di polarizzazione della nave e inviare i dati al centro operativo... tutto quà...”

Pensa te... tanti discorsi e poi per descrivere il lavoro di sei mesi, ha impiegato una frase di 3 secondi. In quel momento qualsiasi fossero stati i numeri in gioco pensavo seriamente di ringraziarli e di mandarli via.

“Insomma la sua scheda è più o meno questa...” e mi passò lo schema elettrico di qualche sfigato che prima di me si era cimentato nel progetto... e già lo immaginavo con gli alluci che uscivano fuori da qualche pilone di cemento del cantiere navale...

“no è che....” dissi con poca convinzione

“bene i costi della scheda vanno dai 5000 ai 7000 euri a pezzo. Ogni scafo ha circa 15 schede ... le petroliere sono circa 15 diciamo che siamo intorno al milione e mezzo di euro... dico... di commessa” disse con un sorriso appena accennato.

“ eeee.. si capisco ...” cercai di prendere tempo e riprendermi un attimo dalla giostra di cifre pressante.

“a parte che devo sentire il mio socio adesso non posso davvero darle delle conferme... io avrei bisogno di qualche giorno per pensarci su... ?”

“ovviamente no non è possibile... io al massimo le posso dare questa notte per pensarci su... domattina mi deve dare delle risposte .. come può immaginare siamo in ritardo con la commessa...”

“ritardo di quanto...”

“da contratto dovevamo consegnare la settimana scorsa... e per finire il progetto se ci va bene ci vorranno tre mesi...”

Tre mesi ?

“Tre mesi sono troppo pochi....”

Vertigo III

Il rumore disturba, confonde, disperde. Un vecchio adagio cinese dice più o meno così: se un albero cade in una foresta sconfinata e non vi è nessuno ad assistervi, la caduta farà o no rumore? Nei sistemi digitali il rumore delle misure analogiche viene convertito in numeri. Il rumore come difetto mangia preziosi bit incrementando il suo stato di appartenenza al progetto. Non è più un generico problema che si potrebbe risolvere ma diventa "il" problema da risolvere. Nel campo della gestione dei segnali è una vera bestia nera e la prima regola del progettista elettronico è che la catena di trasduzione del segnale deve essere la più breve possibile. I tecnici audio aprivano gli amplificatori e contavano i transistor, i migliori erano quelli che ne avevano meno di tutti...

I problemi della vita sono ostacoli che diventano una risorsa solo se vengono superati (anonimo)

I convertitori si suddividono:

- Conversione diretta
- Approssimazioni successive
- Ad Inseguimento
- A doppia rampa
- A pipeline

Ci sono i convertitori definiti **Sigma Delta** ma i puristi dicono si chiami **Delta Sigma** in tutti i casi è meglio identificarlo come sistema o ancora meglio come **metodo di conversione**. Ho notato una differenza buffa nelle pagine di Wikipedia: quella in italiano viene definito metodo

Sigma Delta, mentre in quella inglese Delta Sigma method.

Gli ADC sono essenzialmente lineari ma ne esistono anche di logaritmici ed hanno applicazioni particolari nella sintesi vocale.

Un giorno tentai di implementare un convertitore ad inseguimento pensando di ottenere le risoluzioni che mi servivano con un micro senza convertitori AD. La cosa era più un bell'esercizio ingegneristico che una necessità di lavoro; pensavo però che magari poteva tornarmi utile nel caso ci fossero esigenze di costi...

Nella citazione fatta prima come tutti i problemi quando sono capiti e affrontati diventano una risorsa, una opportunità. Il rumore non fa eccezioni e il Dithering è una conferma ...

Il dithering consiste nell'introdurre artificialmente del rumore nel segnale di ingresso al fine di migliorare la qualità di conversione superando la limitazione di una risoluzione finita. Anche se può sembrare assurdo che del rumore possa migliorare la qualità si può mostrare come questo sia vero con un semplice esempio numerico. Supponiamo che il segnale di ingresso sia sempre pari e costante al valore di 0,34 Volt e che il nostro convertitore abbia una risoluzione di 0,1 volt.

In assenza di rumore il segnale sarà campionato e approssimato come una sequenza di valori pari a 0,3 V, il livello più vicino del quantizzatore. Se invece sommiamo del rumore bianco, cioè un segnale con valore medio nullo, con una varianza pari a 0,1 V (pari alla ri-

soluzione del convertitore, 1LSB) avremo che il segnale oscillerà ora tra 0,24 V e 0,44 V con il risultato che i campioni avranno i valori di 0,2, 0,3 o 0,4 Volt. Per le proprietà statistiche del rumore, il valor medio dei campioni, invece di 0,3 volt, sarà di 0,34 V: in pratica il rumore ha annullato l'errore medio.

Riprendo la scheda, il circuito di disaccoppiamento della seriale doveva essere verificato, dovevo garantire che l'hardware non avesse problemi con le velocità fantasmagoriche di 115 kBaud... .. un fotoaccoppiatore è stato montato su bread-board, alimentato a 5 volt, con resistenza di pull-up sul fotodiodo di 220 Ohm e resistenza di pull-up sul fototransistore di 1 kohm.

Il catodo del fotodiodo è stato connesso all'uscita di un invertitore 74HC14, ed il collettore del fototransistor è stato connesso all'ingresso di un altro invertitore, in questo modo si è inteso simulare una condizione di funzionamento il più possibile vicina a quella **reale**. Con un generatore di funzioni è stato iniettato un segnale ad onda quadra con ampiezza di 5 volt sull'ingresso dell'invertitore-pilota, e si è osservato con un oscilloscopio il segnale sul collettore del fototransistor.

Considerato che la velocità di comunicazione massima alla quale dovranno funzionare le porte seriali della scheda di controllo è 115.2 kB/sec, la frequenza di riferimento per verificare il corretto funzionamento dell'accoppiatore è di 57.6 kHz (alla velocità di 115.2 kB/sec, il

tempo di **durata** di un bit è di 8.68 s, considerando una sequenza di dieci bit <start-bit 10101010 stop.bit>, ne risulta un tratto di onda quadra con frequenza di 57.6 kHz). A questa frequenza ho verificato che il segnale di uscita è praticamente un'onda quadra.

La forma del segnale di uscita comincia a degradare verso frequenze 7 volte superiore a quella di riferimento, ma rimane accettabile fino alla frequenza dieci volte superiore. Inoltre, se si osserva il segnale in uscita all'invertitore d'uscita, il circuito funziona **correttamente** almeno fino alla frequenza di 2 MHz (frequenza massima del generatore di segnale utilizzato).

E' stato provato anche il circuito del segnale DIR della porta RS485, che ha un transistor per invertire il segnale (la sua naturale applicazione...). Questo segnale, nel funzionamento reale della scheda di controllo, avrà una frequenza almeno dieci volte inferiore alla frequenza di riferimento considerata per le prove al punto precedente. Alla frequenza di 5.7 kHz, il segnale di uscita, preso sulla resistenza di pull-down ovvero sul collettore del transistor invertitore, è **praticamente** un'onda quadra. La forma del segnale comincia a degradare per frequenze 6-7 volte superiori, ma rimane accettabile fino alla frequenza di 60 kHz. Volendo, la banda passante può essere aumentata diminuendo il valore della resistenza di pull-down.

Sfruttando l'uscita counter a 16 bit dei microprocessori siamo in grado di creare un treno di impulsi, in cui ogni

impulso ha un'ampiezza conosciuta e costante pari a V , e durata dt ;
ognuno di essi ha pertanto un integrale noto pari a Vdt

Gli errori nella manipolazione del segnale.

La gestione del segnale analogico in particolar modo se è uno strumento di misura deve prevedere in sede progettuale di una stima degli errori. Negli amplificatori operazionali si dovranno considerare ad esempio i parametri V_{offset} e V_{drift} .

La V_{offset} è in pratica la tensione minima che dobbiamo applicare agli ingressi dell'operazionale per portare l'uscita a Zero, tipicamente è espressa in μV . La V_{drift} è invece un parametro che identifica la deriva termica dell'operazionale ed è tipicamente espressa in $\mu V/^{\circ}C$.

L'uso degli amplificatori operazionali deve prevedere ad esempio un altro tipo di errore che potremmo definire deriva massima del guadagno. Questo tipo di errore però in realtà coinvolge le reti resistive utilizzate dall'operazionale stesso. In questo caso si deve considerare che le resistenze, come componente elettronico, sono **termovariabili**. In pratica il valore resistivo ha una "naturale" deriva termica ($\sim 15 \text{ ppm}/^{\circ}C$) e in certe applicazioni è quindi importante sapere la temperatura di lavoro della scheda. Una differenza di $50^{\circ}C$ corrisponde ad una differenza di 750 ppm che per alcuni strumenti di misura potrebbero diventare significativi compromettendone le prestazioni.

14 Novembre 2002

La scheda aveva un connettore amp con i fili di sezione 1,5 che arrivavano attraverso un passacavo della scatola, alle barre della "gabbia di Faraday". Io non sapevo se ci fosse altro, sapevo semplicemente che la misura ballava di quattro bit e che dovevo dimezzare quell'errore per rientrare nelle specifiche. Gli errori sulle misure analogiche si eliminano con i filtri. Nel bignami del sistema embedded sul capitolo misure analogiche si identificano due grandi categorie di errori:

- sincroni
- asincroni

Fra quelli sincroni ci sono errori in bassa frequenza ed errori in alta frequenza. Questo francamente mi sembra essere un errore a frequenza zero...

Rapporto segnale/rumore

La risoluzione di un convertitore è limitata dal rapporto **Segnale/Rumore**. Questo significa che i bit meno significativi diventano funzione non più del segnale ma del rumore. E se mettessi una finestra clamp di quattro bit ? Quando compare quello stato di errore verrebbe segato e loro non si accorgerebbero di nulla. No, che fesseria un clamp a quattro bit sono davvero troppi il convertitore da 16 bit diventerebbe di schianto un 12 bit.

Ci potevano essere sezioni del circuito che potevano influenzare la misura ? Potevo risolvere la cosa implementando un filtro matematico che non fosse il clamp.

Forse sì, ed è quello che feci. La misura analogica veniva implementata una volta ogni 2 ms. Il tempo minimo richiesto dal mio DeltaSigma per terminare la misura era di circa 1,5 ms, 100 us più o meno a seconda del ciclo. Dovevo rispondere alla query del centro che veniva effettuata ogni 10 ms. Così eseguivo le tre misure, dopo di che mi rimanevano 4 millisecondi per eseguire il formulone di conversione.

Nei sistemi dove si hanno più schede su bus interrogati da un centro, di solito è il centro che si sobbarca i calcoli di conversione delle misure. Ma evidentemente in questo caso il volume di dati letti, poi da convertire, avrebbe richiesto una potenza di calcolo inusuale per il computer.

Così il sistemista pensò bene di trasferire parte del calcolo alle periferiche. Queste sono scelte che modificano pesantemente le prestazioni e la qualità del sistema stesso. Io semplicemente ne prendevo atto consapevole che se era sbagliata io sarei diventato un **effetto collaterale** di piccola entità.

Lo spettrofotometro

Mi fermo un attimo. Mi scarico le e-mail e vedo quella del Martini

Ciao Kirkegaard

Sai riguardo la scheda ti volevo dire che in azienda abbiamo....

Martini è un gran curiosone, un giorno lo incontrai al bar dove andavo a mangiare quotidianamente e gli raccontai dei problemi che avevo avuto su una scheda low power che era troppo poco low. I consumi da raggiungere erano intorno ai 50 uA/ora in quella scheda prototipo, avevamo invece consumi superiori ai 300uA/h.

Era un paio di giorni che ci stavo lavorando sopra, ma non ero ancora riuscito a trovare quale sezione di circuito consumasse di più. Lui mi propose di utilizzare un **spettrofotometro** ad infrarossi. In pratica mi consigliava di fare una foto termica alla scheda, partendo dal presupposto che le parti dove era maggiore il flusso di corrente sarebbero diventate più "calde" e avrebbero emesso fotoni visibili agli infrarossi.

Le foto agli infrarossi sono in realtà delle traslazioni di frequenza che consentono appunto di spostare le frequenze dell'infrarosso all'interno dello spettro visibile dall'occhio umano (400-800 Thz). L'occhio come tutti i trasduttori ha una **banda passante** definita. Questo tipo di fotografie rendono evidenti le differenze termiche degli oggetti. Mi lasciano sempre un po' perplesso questi escamotage e così preferii continuare la ricerca dell'ottimizzazione dei consumi manualmente. Le foto per definizione sono comunque istantanee nel tempo mentre i consumi invece sono medie orarie e a loro volta sono la somma di microeventi creati in un sistema embedded

velocissimo.

Guardai un po' il sito di repubblica... distraendomi altri cinque minuti...

Il formulone erano una serie di rapporti in floating point che da prove su banco avrebbero impegnato il micro per circa un millisecondo (in realtà erano circa 780 us). Non poco se si considera che il micro era un aggressivo AT Mega 128 da 16 Mhz.

float

perm,indx,indy,indz,eccx,eccy,eccz,hx,hy,hz,dx,dy,dz;

float res;

void main(void)

{

while(1)

{

res= 5000.0 / 10000.0 * (perm (hx * indx + hy * indy
+ hz * indz + dx * eccx + dy * eccy + dz * eccz)
/ 1000.0);

}

}

E se ci fosse un baco nella sezione di codice che include il formulone ? Non potevo permettermelo per scongiurare ogni problema presi il generatore di funzioni e impostai una **rampa** triangolare a frequenza bassa e con un circuito di conversione simulai la corrente vista dal convertitore. Per un giorno intero feci continue misurare giornaliere con tanto di verifica serale su excel dei grafici.

Provai una serie di filtri a medie di medie, alfa beta, Kal-

man... c'era semplicemente un problema di fondo in questa scelta. Sono tutti filtri che tagliano le frequenze alte... ma non quelle basse...

Implementazione del filtro alfa/beta

$$\alpha + \beta = 1$$

$$V_n = \alpha V(n) + \beta V(n-1)$$

In via preliminare mettiamo i parametri $\alpha = 0.7$ $\beta = 0.3$

Come diminuire un errore evidentemente a bassissima frequenza in un circuito analogico?

La scheda prevedeva anche un circuito di rilevazione del ripple che in realtà era un rilevatore di picco audio degli anni 80. Qualcuno più anziano si ricorderà quegli avveniristici display dei vecchi stereo che visualizzavano il picco di frequenza più alto. E' stato provato su bread-board, con alimentazione di 15 volt e con un segnale sinusoidale ricavato dal generatore di funzioni.

Il circuito si è dimostrato preciso per segnali di ingresso con ampiezza da 100mVpp a 10 Vpp, e con frequenza da 100 Hz a 5 kHz. Il limite di banda inferiore è dato dalla rete RC di ingresso. Il limite di banda superiore è dato dal condensatore da 100pF. Se si elimina il condensatore da 100pF, la banda sale fino a circa 50 kHz.

Guardavo perplesso. La scheda l'altra settimana aveva

superato le prove EMC di compatibilità elettromagnetica, avevamo semplicemente dovuto aggiungere una piccola ferrite sulla alimentazione, per altro isolata, da un doppio trasformatore. La ferrite. Guardavo così quel piccolo anello metallico per fare mente locale. La ferrite rende immuni dai disturbi elettromagnetici, e se questo errore era davvero dovuto ai disturbi elettromagnetici del sistema? Un piccolo **brain storming** mi prese nel cervello. La scheda era montata sopra un alimentatore che aveva quattro condensatori grossi come una damigiana da venti litri... Ora la schermatura dall'alimentatore l'avevano fatta isolando il supporto metallico ma si erano dimenticati il cavo. E se il rumore passava proprio di lì? Chiamai istantaneamente la Exenia avevo bisogno del simulatore per domani mattina libero.

15 Novembre 2002

C'è un famoso aneddoto che racconta di un General Manager della IBM che era stato messo a capo di un progetto internazionale piuttosto importante. Dopo diversi anni il progetto fallì per una serie di motivi, che il nostro ingegnere imputò a se stesso.

Alla riunione con i principali soci azionari, oltre al resoconto del progetto, presentò la lettera di dimissioni ritenendo di non essere più in grado di mantenere l'incarico. Il CEO guardò la lettera, disse qualcosa del genere "Lei con questa esperienza ci è costato xx milioni di dollari se lei se ne va saranno perduti definitivamente. La invito a riprendere il suo posto, le dimissioni non sono accettate...."

Montai la scheda sul supporto mentre Firelli mi stava guardando un po' divertito nel vedere uno che tenta il tutto per tutto prima di finire sul patibolo. Una vite due viti tre viti.. adesso il cavo del connettore della misure e la ferrite. Dubbio atroce, la monto fuori o dentro? Se la monto dentro nessuno si accorge di niente, ma cosa vuol dire davvero disturbo elettromagnetico ? Non lo so, roba da meccanica quantistica ed io in questo momento non sono un fisico ma uno sfigato. Per istinto la montai fuori poi impostammo la macchina in modalità simulazione e accendemmo l'alimentatore sottostante.

Due giorni di misure e l'errore rimase quello che era.

Niente da fare, ero davvero spacciato sino a quando non ripresi il contratto e ricontrollai bene le richieste di progetto...

“la misura A1 dovrà rispettare la precisione pari a 3 bit di errore sull'intera scala.”

l'intera scala, cercai meglio sino a quando ritrovai tutti i documenti... forse avevo un'altra carta da giocare...

La resa dei conti

22 Novembre 2002

I problemi sono come i viaggi, non diventano poi così tanto importanti i risultati quanto il percorso che abbiamo fatto per cercare di risolverli (l'anonimo di prima)

Salii in macchina, un audi a3 prima serie comprata usata cinque anni prima. Partii la mattina presto e raggiunsi dopo un paio d'ore il centro Exenia. All'entrata ricevetti il solito pass in cambio di un documento personale. Raggiunsi la reception che in realtà era il solito ufficio che fungeva da: segreteria, contabilità, ricezione clienti, tutto riunito nella solita stanza sotto la responsabilità di un paio di segretarie.

“Buon giorno Firelli”

“buon giorno Kirkgaard tutto bene?... ha trovato traffico ?

“tutto bene ... tutto bene... non più delle altre volte”

soliti convenevoli

“Bene allora come sono andate le prove...”

“Diciamo che sono andate bene anche se hanno confermato gli errori ...”

Aprii la mia valigetta e tirai fuori la cartellina bristol con dentro la documentazione sulle misure. Firelli si avvicinò allungando la mano. La giacca si sollevò e notai al polso un inconfondibile rolex d'acciaio che stava sopra il

polsino. Dal taschino della giacca usciva invece fuori un corposo cappuccio di penna innervato della mont blank. Guardò le misure e rimase impassibile, cercai di interpretare i movimenti aspettandomi qualche segno di nervosismo in fin fine adesso io e lui eravamo alla resa dei conti.

“Be... allora non abbiamo risolto la questione... mi pare di capire...”

“Ma io direi invece che abbiamo risolto...”

Mentre dissi questo, gli passai le curve gaussiane e la fattura del lavoro fatto sino a quel momento. Avevo messo su una cifra che era vicino ai 50000 euri (considerando circa 25 mila euri di materiale vario). “Mi pare che questa cosa sia difficilmente richiedibile...”
“Io penso di no... Firelli i contratti parlano chiaro...”
“Mi scusi ci eravamo detti...”

Questa volta non gli lasciai finire l'arringa
“Certo c'eravamo detti di rispettare il contratto... e vede il **contratto** parla chiaro...”

Tirai fuori la fotocopia delle richieste progettuali, con movimento scenografico... spostai il documento che aveva sott'occhio e vi misi il ciclostilato...
“non capisco...” disse fra i denti, ma non fece in tempo ad aggiungere altro perché veloce presi l'evidenziatore che mi ero portato sulla tasca destra (in offerta al Conad 7,5 € la scatola da 10) lo tirai fuori e lo stappai davanti ai suoi occhi... “Vede Firelli questa è la copia del contratto che mi ha portato in data xx xx xx e che è sta-

ta firmata e riconsegnata in data xx xx xx ... vede qui in fondo ?”

“ ...”

Nessuna risposta, ma io non avevo bisogno di risposte... “bene in questo contratto c'è scritto bene la dicitura errore sulla misura massimo 3 bit sul range di misura...”

“infatti sono proprio i tre bit...”

tentò di balbettare ma non lo lasciai finire...

“già ma il range di misura citato va da 0 a 8120 considerando un convertitore a 12 bit... vede qua in fondo ?”

indicai la nota relativa alla sezione analogica con la richiesta.

“Ma noi in data xx xx xx abbiamo aggiornato la scheda e come da accordi intercorsi ed unanimi abbiamo modificato la scheda e messo un convertitore a 16 bit...” alla parola unanimi feci il segno delle virgolette con le dita alzate. L'avevo imparato in un film con Gene Hackman. Firelli non era uno stupido e si accorse subito dove volevo andare a parare.

“Da questi grafici si evidenzia la natura **caotica** del rumore.. attraverso la serie di campane di Gauss, se facciamo un rapporto percentuale sui dati rilevati si vede chiaramente che i 3 bit di errore sulla misura a 12 bit hanno un peso maggiore rispetto ai 4 bit di errore rilevati con il convertitore a 16 bit...”

Tentò una disperata difesa.. “Sì ma l'errore è dentro il

range a 12 bit..."

"Firelli... guardi non è difficile... come le ho già detto il bit di un convertitore a 12 bit non è il solito di un convertitore a 16 bit... posso dimostrarglielo velocemente se crede"

era esausto...

"Va bene .. va bene. Quindi mi faccia capire per lei la cosa va bene così?"

"Certo che va bene così l'errore sulla misura è inferiore alla richiesta contrattuale. Veda lei..."

Un attimo di empasse... Firelli non aveva più argomenti nessun commento, solo uno sguardo perso fra tabelle e grafici di analisi statistica. Ci salutammo cordialmente, chiusi la mia 24 ore comprata da Stefan a 17,50 euro il mese scorso... ma prima di andar via

"Dimenticavo questo .. è il conto... Firelli io... devo andare ho un impegno"

Me ne andai ma dopo una settimana arrivò il bonifico non fecero una piega. La commessa saltò comunque, ma ancora adesso non sono riuscito a sapere se fu quel bit in più di errore a far decadere il progetto. Forse era meglio così... anzi tutte le volte che ci penso ne sono convinto, anche perché da lì a poco con la morte di Saddam la guerra terminò e quella storia ... finì per sempre.

Indice generale

La mela mancante.....	3
Under the Bug.....	28
Baco sulla fiducia.....	30
Baco presunto.....	35
Baco specialistico.....	42
Come scrivere codice.....	51
ingestibile.....	51
Nomenclatura delle variabili.....	53
Nomenclatura onomastica.....	53
Minimalista.....	54
Scrittura erronea creativa.....	54
Concetti astratti.....	54
Acronimi.....	54
Lingue straniere.....	54
Maisucolo randomico.....	54
Accenti.....	55
Lunghezza.....	55
ASCII esteso.....	55
Riferimenti matematici.....	55
Nomi accecanti.....	55
Lower Case l Looks a Lot Like the Di- git 1.....	56
Riusa le variabili globali come private o locali.....	56
Cd wrttn wtht vwls s mch trsr.....	56
HOT Tips di offuscamento del codice.....	56

Aggiungi tonnellate commenti.....	56
Togli gli spazi.....	57
Inverti il senso.....	58
Mischia i linguaggi.....	58
La notazione Ungara.....	59
Mascherare il codice.....	59
Definizione di macro.....	60
Prolungamento di linea.....	60
Overloading.....	60
Doppio puntatore.....	61
CallBack.....	61
Codice ricorsivo.....	63
Funzioni monotone.....	63
[modifica]Namespace.....	63
Macro inutili.....	64
Nascondi le istanze.....	64
Ritorni di funzione.....	65
Gotchas.....	65
Documentare variabili.....	66
GOTO riesumato.....	66
Offusca un sistema embedded.....	66
Loop bloccanti.....	67
Watchdog fantasma.....	67
Watchdog a schiovere.....	67
Awards dell'offuscamento del Codice.....	67
Vertigo.....	69
L'azienda.....	70
L'errore.....	73
Ritorno a casa.....	76

Vertigo II.....	80
Gli Operazionali.....	81
Le mine in sintesi.....	90
Degaussing.....	93
Philadelphia Experiment.....	96
La proposta.....	97
Vertigo III.....	100
Rapporto segnale/rumore.....	105
Lo spettrofotometro.....	106
La resa dei conti	111