# Coffee Making

Fotis Georgatos < *gef@ceid.upatras.gr* >
V1.0 2004-08-29

Uno dei più memorabili commenti mai fatti sul software è se questo o quel pezzo di codice possa fare il caffè. Il caffè è un bene del mondo secondo solo al petrolio. Linux FA il caffè; ed è anche buono! Traduzione fino al 1998 a cura di Christopher R. Gabriel, dopo il 1998 a cura di Claudio Cattazzo, claudio(at)pluto.it, e Giuseppe Briotti, g.briotti(at)mclink.it.

## Indice

# 1 Introduzione

Per lungo tempo l'umanità si è chiesta come un computer potesse fare il caffè...

Le persone hanno bisogno di caffè per svegliarsi e restare sveglie per lungo tempo di fronte al computer. Tutti sanno che per programmare è meglio la notte!

Il trucco principale è interfacciare la macchina da caffè con il computer, in modo da poterla controllare via software. Questo HOWTO mostrerà come farlo.

Prima, descrive un interruttore ACCESO/SPENTO implementato come un circuito che controlla l'alimentazione della macchina da caffè. Un altro capitolo spiegherà i segreti della costruzione di macchine di Turing da caffè, affidabili e intelligenti!

Questo HOWTO è stato scritto inizialmente come parte di un dibattito nella mailing list linux-greek-users sul fatto che linux potesse o meno fare il caffè. È divenuto poi un articolo sul nostro magazine online chiamato *magaz* . Nel caso ve lo chiediate, magaz è in greco e sicuramente vi apparirà così!

Buon divertimento.

## 1.1 Copyright

## 1.2 Disclaimer

Si usino le informazioni presenti in questo documento a proprio rischio. Rinnego ogni potenziale responsabilità per i contenuti di questo documento. L'utilizzo di concetti, esempi e/o altro contenuto di questo documento è interamente a proprio rischio.

Tutti i copyright sono dei rispettivi possessori, a meno che non sia specificato diversamente. L'uso di un termine in questo documento non dovrebbe interessare la validità dei marchi.

Il fatto di nominare un particolare prodotto o marca non dovrebbe venire visto come promozione.

Si consiglia fortemente di fare un backup del proprio sistema prima di installazioni di grande entità e di effettuare backup ad intervalli regolari.

## 1.3 Versione

Il Coffee HOWTO ora è chiamato Coffee Making HOWTO e si avvicina alla release v1.0, che apparirà in prima edizione da qualche parte qui:

*http://fotis.home.cern.ch/fotis/Coffee.html* .

È tempo per ognuno di sapere che fare il caffè è solo una delle funzionalità standard presenti gratuitamente in *ogni* distribuzione Linux. O SCO ha un brevetto anche su quello?

## 1.4 Traduzioni

Si dovrebbe essere in grado di trovare facilmente una traduzione di questa versione o delle precedenti del Coffee Making HOWTO nelle seguenti lingue:

- Cinese
- Giapponese
- Russo
- Indonesiano
- Italiano
- Polacco

e si spera molte altre... (nessuno spagnolo o portoghese?)

## 1.5 Ringraziamenti

- Etiopia: identificata come il paese di origine del caffè.
- Il caffè è popolare nel Medio Oriente da secoli, finché un fallito assedio dei Turchi alla città di Vienna nel 1683 si lasciò dietro sacchi pieni di strani chicchi marroni che nessuno voleva: *http://www.vienna.cc/ekaffeeh.htm/* .
- Kostas Lialiambis è quello che osò dire che non era in grado di fare il caffè con il suo sistema Linux, ancora nel 1997.
- Panagiotis Vrionis per avermi fornito note interessanti ed umoristiche ed aver avviato tutto nei primi tempi.

- NUMEROSE persone in internet con aggiunte ed osservazioni. Grazie a tutti, veramente! Anche se posso non aver risposto alle vostre email!

### 1.6 Cosa bevi o fumi mentre scrivi questo documento?

Per quanto mi risulta, questo è un lavoro svolto senza uso di doping.

Ma posso dirvi il segreto della musica suonata in sottofondo: quasi tutte le canzoni di Zampetas o Mpithikotsis (bouzouki greco e affini).

### 1.7 Commenti

Per i commenti online sulla propria macchina da caffè rivolgersi a *http://coffee.sf.net/* .

Se avete ancora commenti da fare, le email in questi giorni vanno perse, quindi perché non inviarmi semplicemente una cartolina con una immagine della vostra magnifica città, aggiungendo una raccomandazione per il caffè preferito dalle vostre parti?

```
Fotis Georgatos,
Aliartou 32,
TK 11142 Athens,
GREECE
```

PS.

- I vostri suggerimenti non andranno pers, in questi giorni tendo a viaggiare parecchio in giro per il mondo e potrei essere da voi prima che ve ne rendiate conto.
- Certo, sono stato ad Amsterdam ed ho anche vissuto lì per tre anni; i coffee shop generalmente non riguardano il caffè!
- Sorpresa: al momento preferisco la cioccolata o il tè al caffè. ;-)

## 2 Menu

### 2.1 Alla francese

Caffè molto popolare tra i programmatori perché non necessita di molta cura e la sua cottura è ridotta all'essenziale; come il software commerciale. Il suo gusto stimolante ha ispirato migliaia di programmatori nella scrittura di software incredibile, scritto nelle primissime ore del giorno. Windows per esempio è stato scritto alle 5 esatte del mattino, solo grazie al caffè! Un risultato simole è garantito.

### 2.2 Nescafè

Nescafè è un caffè piuttosto forte, fatto mescolando acqua calda con una mistura di caffè, zucchero e un po' d'acqua. Solitamente si versano un cucchiaio di caffè e uno di zucchero, con pochissima acqua per miscelarli. Nel frattempo l'acqua dovrebbe bollire. Non appena l'acqua è calda a sufficienza, mescolare tutto assieme e preferibilmente aggiungere latte condensato. Sebbene si possa usare qualcosa di più semplice di una macchina da caffè per scaldare l'acqua, l'ho visto fare così molte volte...

## 2.3 Frappè

Una popolare variante del caffè appena menzionato. In realtà non è necessaria una macchina da caffè, ma un congelatore per avere acqua fredda e cubetti di ghiaccio. È molto popolare nel sud-est dell'Europa durante i caldi mesi estivi.

## 2.4 Freddo

Questo è complicato, si leggano le Coffee FAQ (si vedano i riferimenti).

## 2.5 Espresso

L'espresso è un tipo di caffè italiano molto forte. Lo si serve in piccole tazze (Perché? Si veda il capitolo Sintomi da overdose) con una o due zollette di zucchero. Per produrre un buon espresso c'è bisogno di grani di caffè appena macinati, acqua, zollette di zucchero e una macchina speciale. Queste macchine fanno bollire l'acqua e spingono il vapore bollente attraverso i grani di caffè macinati. Si può comprare una macchina super-mega-automatica con un bel po' di denaro. Ma si può usare anche una macchina a basso prezzo.

Ok, iniziamo. Riempire d'acqua la propria macchina. Lasciarla diventare calda. Nel frattempo, riempire con circa un cucchiaio da tè di polvere di caffè il porta filtro della propria macchina. Pressare la polvere di caffè. Non troppo. Ora l'acqua è alla giusta temperatura. Fissare il porta filtro alla macchina e lasciarla lavorare. Dopo circa 30 secondi si può servire un delicato, caldo espresso. È eccellente dopo un buon pasto. Ci si sente bene e si può programmare per qualche ora in più.

## 2.6 Cappuccino

(Si veda anche il capitolo Espresso) Se si possiede una macchina più professionale, la si può usare per montare il latte. C'è bisogno di questa funzionalità per fare una variante morbida del caffè. È facile da preparare. Mettere del latte montato in una tazza da caffè e riempirla con l'espresso. Decorarla poi con qualche fiocco di cioccolata. Tutto qua.

# 3 Hardware

Un diagramma generale potrebbe essere come questo:

```
---------    0-5V  ----------  ~220V   -------------------
|  PC   |====>====|Circuito|==========|Macchina da caffè|
---------          ----------          -------------------
```

Il concetto è quello di prendere una tensione di controllo dal computer, che guida un circuito elettrico isolato con un Relè o un Triac.

Si deve scegliere un circuito a Relè se si ha una macchina da caffè superiore ai 200W. Se ne può usare uno basato su Triac se la propria macchina non è ad alta potenza.

Tutti i circuiti presentati sono stati sperimentati, ma i risultati e i rischi sono di VOSTRA RESPONSABILITÀ. Se non si ha esperienza con l'elettronica, NON SI DEVE cercare di farsela basandosi su questi circuiti, altrimenti se ne può ottenere una pessima...

Si deve stare molto attenti mentre si sperimenta con i 220V e si consiglia caldamente di usare un fusibile giusto.

## 3.1 Pilotare la tensione 0-5V dal computer

Ecco un semplice esempio di come ottenere una tensione 0-5V dalla porta parallela del computer.

```
          Vista posteriore    -----    Pin 10 - ACK
          Connettore DB-25    |   |    Pin  9 - D7
          Maschio             |   |                         Pin 2 - D0
                              v   v                         v   Pin 1 -
     ~Strobe
           -----------------------------------------------------------
          /                                                           \
          \     13   12   11   10    9    8    7    6    5    4    3    2    1     /
           \                                                         /
            \     25   24   23   22   21   20   19   18   17   16   15   14     /
             \_______________________________________________________/
```

Pin 1 è lo Strobe (logica inversa).

Pin 2-9 sono i segnali del BUS DATI, esattamente quello che è stato scritto nei latch della porta parallela con un comando OUTB.

Pin 10 è il segnale di acknowledge (ACK), controllato da voi, in modo da poter produrre un interrupt nella CPU.

Pin 18-25 sono corto-circuitati e rappresentano la massa (GND).

In dettaglio:

| <= in<br>=> out | DB25<br>pin | Cent<br>pin | Nome del<br>Segnale | Reg<br>Bit | Note sulla funzione |
|---|---|---|---|---|---|
| => | 1 | 1 | -Strobe | C0- | Imposta l'impulso basso >0.5uS per permettere la scrittura |
| => | 2 | 2 | Data 0 | D0 | Imposta il dato meno significativo |
| => | 3 | 3 | Data 1 | D1 | ... |
| => | 4 | 4 | Data 2 | D2 | ... |
| => | 5 | 5 | Data 3 | D3 | ... |
| => | 6 | 6 | Data 4 | D4 | ... |
| => | 7 | 7 | Data 5 | D5 | ... |
| => | 8 | 8 | Data 6 | D6 | ... |
| => | 9 | 9 | Data 7 | D7 | Imposta il dato più significativo |
| <= | 10 | 10 | -Ack | S6+ IRQ | Impulso basso ~ 5uS per conferma ricezione |
| <= | 11 | 11 | +Busy | S7- | Alto per Occupato/Spento/Errore |
| <= | 12 | 12 | +PaperEnd | S5+ | Alto per fine carta |
| <= | 13 | 13 | +SelectIn | S4+ | Alto per stampante selezionata |
| => | 14 | 14 | -AutoFd | C1- | Imposta basso per autoavanzamento di una riga |
| <= | 15 | 32 | -Error | S3+ | Basso per Errore/Spento/Fine carta |
| => | 16 | 31 | -Init | C2+ | Imposta l'impulso basso > 50uS per inizializzare |
| => | 17 | 36 | -Select | C3- | Imposta basso per selezionare la stampante |
| == | 18-25 | 19-30, | Ground | | |

## 3.2 Controllo tramite relè

Il circuito più semplice che si possa costruire è:

```
                              Vcc
                               |
                               +------+
                               |    __|__
                           Relè con  /^\  Diodo 1N4002
                             bobina /---\
                               |      |
                               +------+
                               |
                            | /
                    4.7K    B |/  C
porta parallela >-\/\/\/\/---|        Transistor NPN: BC547A o 2N2222A
pin di dati                  |\  E
                            | \
                              V
                              |
porta parallela >--------------+
pin di massa                  |
                           Massa
```

Si colleghi Vcc alla stessa tensione del tipo di relè (solitamente 5 o 12V). Ovviamente le specifiche del relè devono essere adatte alla propria macchina da caffè.

I baristi hanno la tendenza a mettere il relè DOPO il transistor, al pin emettitore (E) invece che al collettore (C). È una cattiva abitudine perché influenza malamente il transistor e può risultarne un pessimo caffè :-). Il diodo 1N4002 è utile per proteggere il transistor dalla corrente del relè. Se non lo si utilizza, il transistor diventerà ben presto scuro e puzzolente...

## 3.3 Controllo con un TRIAC #1

Se si desidera un circuito semplice si può usare il driver triac MOC301[012] della Motorola, assieme ad un TRIAC multi-funzione come l'SC141D. Questo metodo ha il vantaggio che non necessita di alimentazione supplementare.

Per un carico non induttivo si usi questa circuiteria:

```
       270     1 +-------+ 6    180
+5v -VAVAVA-----+       +----VAVAVA-----+-------------- Linea carica
             2 |  MOC  |               |
TTL in ---------+ 3012  +nc            VA  SC141D
                |       | 4          / |
             nc+       +------------/  |
                +-------+              +----\/\/\/---- Linea non carica
                                            CARICO
```

Se si sta lavorando con 220V si cerchi di ottenere un 3021. Carichi induttivi dovrebbero essere utilizzati assieme a dei condensatori di bypass, meglio consultare la *Motorola Application Note AN-780*. Le macchine da caffè sono per lo più carichi resistivi e non induttivi (come un motore), ma chissà come è la vostra?

## 3.4 Controllo con un TRIAC #2

```
+5VDC
|    180                      180           2.2k
+---/\/\/\----+-----+   +----/\/\/-+--/\/\/\---+-------> 120V
              |    1|   |6         |           |         Carico
              |    +=====+         |           | MT1
              |    | MC  | Driver  |          +-+
              |    | 3032| TRIAC   |        G | | TRIAC
              |    +=====+         |         /| |
              \    2|   |4         |        / +-+
       2N3904  |----+   |          |        |  | MT2
              /     |   +--------- | -------+  |
             V      \              |        |  |
             |      /              |        \  |
             |      \ 43    .01u  ---   10k /  |
             |      /       500V  ---       \  |
             |      |              |        /  |
             +------+              |        |  |            Non carico
             |                     +--------+--+---o    o--> 120V
             /                                     carico
 >-/\/\--|  2N3904
             \
              V
              |
             ---
            ///
```

```
Il progetto è per 120V. Si dovrebbero cambiare le resistenze in funzione dei
220V.
```

Descrizione del circuito:

L'MC3032 è un driver TRIAC optoisolatore. La resistenza da 180 Ohm imposta la corrente per il LED dentro l'optoisolatore. Si cambi il valore di questa resistenza, se necessario, per ottenere una corrente accettabile (es. 15 mA).

Si noti che non si può controllare questo circuito senza un carico. Il TRIAC non cambierà a meno che non sia connesso ad una sorgente di tensione AC, quindi non lo si può controllare per un semplice interruttore senza applicare AC e un carico. Si noti la valutazione di 500V sul condensatore da .01.

# 4 Software

## 4.1 Programmazione

Si dovrà creare un eseguibile che farà i seguenti passi:

- Ottenere i permessi per usare lo spazio degli indirizzi di I/O, chiamando il kernel, con la chiamata **ioperm**: es. *ioperm(BASE, intervallo, 1);*
- Eseguire un'istruzione di richiesta in uscita per impostare la tensione 0.5V sulla porta parallela, es. *outb(1, BASE);*
- Attendere il tempo necessario alla preparazione del caffè. Sarebbe bello se questo tempo fosse mostrato nella riga di comando.
- Quindi dovrà spegnere la macchina del caffè: *outb(0, BASE)*;

- Prima di terminare deve restituire la porta parallela con un *ioperm(BASE, intervallo, 0);*

Si cambi BASE = 0x3bc per /dev/lp0, 0x378 per /dev/lp1 e 0x278 per /dev/lp2; intervallo = 8.

Sarebbe utile avere questo programma setuid in modo che tutti possano bere il caffè! BOFH!

### 4.2 Driver di periferica

Basta leggere la *kernel hacker's guide* per implementare un driver di periferica; lo si potrebbe anche fare in user space. Per favore lo si compili come modulo, in modo che non si debba compilare il kernel ad ogni aggiornamento. Quindi si scriva:

```
echo cappuccino > /dev/coffee
```

E si avrà una calda tazza di caffè in pochi minuti! Ci si ricordi di assegnare i giusti permessi a /dev/coffee, a seconda che solo root possa farsi il caffè o meno.

Il vantaggio di questo metodo è che supporta il feedback dalla macchina da caffè utilizzando l'ACK della porta parallela, quindi macchine da caffè intelligenti potrebbero produrre un interrupt quando sono pronte.

Lo si faccia da soli, dopo aver letto l'eccellente libro di Alessandro Rubini e Jonathan Corbet *Linux Device Drivers* e studiato il repository del codice sorgente di *Cross Reference Linux* .

### 4.3 Connessione via Internet

Se si è implementato il programma di controllo in C (si veda sopra), si deve soltanto scrivere uno script CGI che accende e spegne la macchina da caffè o che passi attraverso istruzioni più complesse. Si dovrebbero scrivere alcune pagine web carine, dove si spiega come fare il caffè, e metterle su un server web **Apache**...

...la tecnologia LAMP (Linux, Apache, MySQL, [Perl|Python|PHP]), vi aiuterà a costruire un sistema per il caffè completamente configurabile!

In futuro, quando le applicazioni tenderanno a divenire più complesse, potrete volerle estendere sulla base della programmazione Flow-Based: *http://www.jpaulmorrison.com/fbp/* . Che accoppiata per una grande macchina da caffè!

## 5 Costruire la Macchina di Turing da Caffè

Vi struggete in memoria dei bei vecchi tempi, quando gli uomini erano uomini e si costruivano da soli la propria macchina da caffè?

Questo capitolo spiega il modo di assemblare una autentica, intelligente, macchina da caffè. Sarà un computer progettato con architettura di von Neumann, comprensivo di una CPU, ROM/RAM, I/O e sarà anche adatto ad impieghi più generici, una cosiddetta *Macchina Universale di Turing* .

### 5.1 Un linguaggio assembly adatto

Contrariamente ad altri complessi ma popolari sistemi quali CISC o RISC, la nostra macchina sarà semplicemente MISC: Mono-Instruction Set Computer!

Il nostro processore comprenderà soltanto un comando, ma ciononostante, con memoria e tempo sufficiente, sarà in grado di eseguire qualunque azione che possa essere compiuta dal vostro Pentium IV 3GHz, o simularla completamente. Può risolvere un qualunque problema computabile eseguendo del semplice codice come questo:

```
SBN     $mem1, $addr1
SBN     $mem2, $addr2
SBN     $mem3, $addr3
SBN     $mem4, $addr4
SBN     $mem5, $addr5
SBN     $mem6, $addr6
[...]
```

Il comando magico è chiamato SBN $mem, $addr (Subtract and Branch if Negative), che prende il valore dalla cella di memoria $mem, lo sottrae all'accumulatore (A è l'unico registro disponibile in questa architettura) e lo memorizza nuovamente nell'accumulatore e in memoria $mem: *[mem]* $<= A <= A$-*[mem]*. Se e solo se il risultato è negativo, salta all'indirizzo specificato da $addr. Se $addr punta al comando successivo, non vi è alcun salto condizionato. Ora, con questo comando a disposizione, è possibile sottrarre, sommare, azzerare gli indirizzi di memoria, spostare byte, moltiplicare, comparare e così via. Cosa migliore di tutte, ci si può costruire facilmente un compilatore ottimizzato.

Voila. Questo è un grande sistema per qualunque problema di Turing Completo, inoltre, è sicuramente più facile da programmare della Macchina di Turing originale!

## 5.2 Hardware ed interfacciamento

La gran cosa di questo innovativo processore MISC è che si ha bisogno di 0 bit per memorizzare l'opcode dei comandi. Questo rende la CPU molto, molto più semplice: si ha bisogno di leggere solo una coppia di operandi ogni volta. Si potrebbe voler estendere le possibilità del processore estendendo l'istruzione SBN a 3 o 4 operandi, così sarebbe in grado di caricare e memorizzare i dati dalla memoria principale. Questo è un esercizio lasciato al lettore; onore a *google* .

Il diagramma della CPU risulterà come segue:

```
<========= ADDRESS BUS ==============>
       =                =
       =  +---------+   =
       =  | CONTROL |   =
   +---------+  +-----------------+
   | ALU & A |  | Program Counter |
   +---------+  +-----------------+
       =  |  LOGIC  |   =
       =  +---------+   =
       =                =
<=========== DATA BUS ===============>
```

Ora, tutto quello che si deve fare è mettere insieme alcuni chip di memoria, ad esempio riciclando una RAM di cache statica da vecchi PC 386, una ALU e pochi componenti glue. Si può scegliere tra TTL o CMOS per le porte logiche e i latch; io sono un sostenitore di CMOS, ma questo dipende in realtà dalle proprie preferenze. Si può costruire un sistema ad 8, 16, 32, 64 bit o della dimensione di cui si ha bisogno. Nel caso servisse, per word più lunghe ho trovato preferibile costruire l'ALU con EPROM 27128 pre-programmate piuttosto dei chip 74x181s, più difficili da trovare. Ci si procuri anche un'unità di propagazione del carry.

La natura monolitica di questo sistema consente solo I/O mappato in memoria e richiede di provvedere ad un progetto particolare per l'interfacciamento bidirezionale, ma niente di più particolare di quanto già visto in sistemi di più vecchia generazione. *AGC* , il computer che ha guidato la missione Apollo 11 sulla Luna, è stato realizzato utilizzando questa tecnica, quindi dovrebbe essere sufficiente anche per i nostri scopi.

Si osservi che il bus dei dati deve essere ampio esattamente come il bus degli indirizzi; questo implica che la nozione di un byte è applicabile solo alle macchine da caffè ad 8 bit, il che risula essere più una funzionalità che un baco. Si resterà sorpresi da quale caffè si potrà ottenere con un bus a 8 o 16 bit! È davvero un hardware di impiego generale.

## 5.3 Software

Un sistema così puro si accoppierà bene con il linguaggio di programmazione *FORTH* , famoso per il controllo di sistemi embedded. Il più importante prerequisito per fare questo è di avere un meccanismo di stack, che in questo caso può essere realizzato con un contatore combinato con un pool di memoria.

Se si aspira ad una seria piattaforma di sviluppo per il caffè, la portabilità del C è una scelta obbligatoria. La propria scelta potrebbe essere quella di modificare gcc, lcc o sdcc, che con i giusti aggiustamenti ai back-end saranno in grado di produrre lo speciale codice assembly MISC. Un giorno si potrebbe anche voler riscrivere un altro linguaggio come il C; ci si scordi la lettera D, è già stata presa, quindi non si facciano gli stessi errori con il compilatore: http://www.gnu.org/software/gcc/projects/beginner.html

Qualora si stia pensando di scriversi il proprio compilatore, ci si documenti prima su flex, yacc e solo un altro po' di teoria. In particolare, si apprezzerà subito la tassonomia di Noam Chomsky sui linguaggi:

- grammatica regolare (formalismi generali per le espressioni regolari),
- grammatica libera dal contesto (qualunque linguaggio BNF rientra in questa categoria)

e così via. Si legga anche *Computability Theory* .

## 5.4 Una criticità minore della Macchina di Turing

A causa di come una Macchina di Turing opera (si guardi a proposito *http://plato.stanford.edu/entries/turing-machine/* ), è un dispositivo molto complicato da programmare e su cui fare il debug una volta fatto tutto. La ragione è che il suo comportamento è un processo sequenziale completamente determinato dai seguenti parametri:

- (1) lo Stato in cui si trova la macchina;
- (2) il Simbolo (o Numero) nella casella che sta analizzando;
- (3) una Tabella di Istruzioni.

Contemporaneamente, il maggiore svantaggio della Macchina di Turing (MT) è che la sua natura sequenziale comporta che può essere efficacemente mappata su di essa solo una particolare classe di problemi. Le MT sono adatte per problemi che sono descritti su un supporto di immagazzinamento seriale (nastri) e che non fanno utilizzo di indici per referenziare i dati. Questo è in contrasto con la Macchina da Caffè (MC), che può gestire anche algoritmi ad Accesso Casuale (senza perdita di semplicità).

In aggiunta a questo, le MT costringono ad una notevole e non necessaria complessità sul punto (3) con l'idea di mantenere semplici i punti (1) e (2). E qualora non si ritenga che la così chiamata Tabella delle Istruzioni non sia di una complessità schiacciante, mai provato a scrivere un compilatore per una Macchina di Turing? Un sistema che non sia facilmente programmabile e sia di difficile debug, dovrebbe essere considerato un sistema seriamente discutibile, almeno per quanto riguarda l'Ingegneria dei Computer. Per esempio, si provi a simulare la Macchina da Caffè con una Macchina di Turing e viceversa. Beh, se ancora si è in disaccordo, mostratemi il codice!

Ultima annotazione: la Macchina da Caffè (MC) è un modello per l'architettura di von Neumann molto migliore ed ha una relazione O(1) con quella che è la pratica diffusa degli algoritmi pesati nell'attuale teoria della complessità.

## 6 Sintomi da overdose

- eccitazione
- nervosismo
- insonnia
- tachicardia o aritmia cardiaca
- inquietudine
- ipersensibilità alla luce
- fastidio rispetto a stimoli sonori
- disturbi gastrointestinali

## 7 Espansioni

- Tutto l'hardware e il software descritti in questo documento possono essere espansi fino a supportare tostapane, bistecchiera, forno a microonde, ecc.
- Cluster di 8 macchine da caffè. Questo permetterà di avere del caffè anche se la prima macchina si spegne. Ovviamente ci sarà un aumento delle prestazioni.
- Macchine da caffè a vettore parallelo saranno una futura release.
- Se si desidera la massima automazione si avrà bisogno di più circuiti e sensori, in modo da poter controllare il flusso dell'acqua, la temperatura, la quantità di caffè, ecc.
- Nel prossimo futuro verranno implementate funzionalità SNMP.
- Macchina da caffè seriale a 115Kbps.

## 8 Maggiori informazioni

- *http://db.uwaterloo.ca/~alopez-o/Coffee/coffaq.html*
  Questa è la **Coffee-FAQ** più conosciuta di Internet
- *http://faculty.washington.edu/chudler/caff.html*
  Caffeina ed effetti sul Sistema Nervoso

- *http://www.gardfoods.com/coffee/coffee.coffee.htm*

  Una storia completa, anche se informale, dell'interazione dell'uomo con la pianta di caffè. La gente ha masticato bacche di caffè in Africa per circa 100.000 anni. Il caffè si è sviluppato nello Yemen, dove non è nativo (deve essere seminato) nel 525 dC.

- *http://www.faqs.org/rfcs/rfc2324.html*

  RFC2324: Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)

- *http://www.faqs.org/rfcs/rfc2325.html*

  RFC2325: Definitions of Managed Objects for Drip-Type Heated Beverage Hardware Devices using SMIv2

- *http://www.tldp.org/HOWTO/mini/IO-Port-Programming*

  Programmazione della porte di I/O sotto il popolare sistema operativo Linux.

- *http://www.ee.washington.edu/circuit_archive/circuits/F_ASCII_Schem_PC.html*

  Tantissimi circuiti in ASCII. Alcuni di questi sono per la porta parallela.

- *http://www.citr.auckland.ac.nz/~james/parport.html*

  Tutto quello che avreste voluto sapere sulla porta parallela e non avete mai osato chiedere.

- *http://en.tldp.org/LDP/khg/HyperNews/get/khg.html*

  Come scrivere i propri driver di periferiche. Avanti, fatelo anche voi!

- *http://www.hut.fi/Misc/Electronics/circuits/parallel_output.html*

  La pagina web di Tomi Engdahl; è *obbligatorio* vederla per chiunque si interessi di elettronica.

- *http://dir.yahoo.com/Computers_and_Internet/Internet/Devices_Connected_to_the_Internet/Coffee_Machines*

  Macchine da caffè online. Sfortunamente non ci sono benchmark.

- *http://www.cs.su.oz.au/~bob/Coffee/index.html*

  Questa macchina da caffè fornisce solo cappuccino. Deve essere aggiornata!

- *http://einstein.et.tudelft.nl/~janssen/*

  Caffè caldo dai Paesi Bassi.

- *http://www.cl.cam.ac.uk/coffee/coffee.html*

  The Trojan Room Coffee Machine.

- *http://www.menet.umn.edu/coffeecam/*

  CoffeeCAM.

L'elenco di link in questo capitolo spesso diventa superato, per questo si può voler utilizzare l'eccellente Way-Back Machine per ritrovarli nuovamente: *http://www.archive.org/*