

Jurassic News

- hardware: **Exidy Sorcerer**
- software: **Borland Sidekick**
- racconto: **Rimini, Rimini!**
- emulazione: **ENIGMA**
- laboratorio: **ENIGMA con chip AT Mega 238P**
- linguaggi: **SNOBOL**
- storia: **il PC IBM**
- dal passato: **M24, Toshiba T1000, P101**



Retrocomputer Magazine

Anno 10 - Numero 55 - Settembre 2015

Jurassic News

Rivista aperiodica di Retrocomputer

Coordinatore editoriale:

Tullio Nicolussi [Tn]

Redazione:

redazione@jurassicnews.com

Hanno collaborato a questo numero:

Lorenzo [L2]

Salvatore Macomer [Sm]

Sonicher [Sn]

Besdelsec [Bs]

Lorenzo Paolini [Lp]

Giovanni [jb72]

double_wrap

Damiano Cavicchio

Riccardo Franch

Diffusione:

Lettura on-line sul sito o attraverso il servizio Issuu.com; il download è disponibile per gli utenti registrati.

Sito Web:

www.jurassicnews.com

Contatti:

info@jurassicnews.com

Copyright:

I marchi citati sono di copyrights dei rispettivi proprietari.

La riproduzione con qualsiasi mezzo di illustrazioni e di articoli pubblicati sulla rivista, nonché la loro traduzione, è riservata e non può avvenire senza espressa autorizzazione.

Jurassic News

promuove la libera circolazione delle idee

Jurassic News

E' una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte su Internet.

La redazione e gli autori degli articoli non si assumono nessuna responsabilità in merito alla correttezza delle informazioni riportate o nei confronti di eventuali danni derivanti dall'applicazione di quanto appreso sulla rivista.

Il contenuto degli articoli è frutto delle conoscenze, esperienze personali e opinioni dei singoli autori; possono pertanto essere talvolta non precise o differire da fonti "ufficiose" come Wikipedia e siti Web specializzati.

Sono gradite segnalazioni di errori, imprecisioni o errate informazioni che possono, a discrezione della redazione, essere oggetto di errata-corrige in fascicoli successivi.

Scrivere a:

*redazione@jurassicnews.com
dettagliando il più possibile
l'argomentazione.*

Riservatezza o schizofrenia?

In un'epoca dove ci pensiamo pochissimo a mettere un "mi piace" su un elemento di Facebook o postare la nostra opinione su un thread di twitter, dove siamo assillati dall'idea di fare gruppi, cerchie o pagine fan sulla pletora di social-qualcosa, beh, siamo (o vorremmo essere) più di ogni altra epoca sicuri che nessuno riesca a carpire le nostre informazioni.

Si sa che si vive ormai nella contraddizione più palese: da una parte vogliamo farci notare, mentre dall'altra vorremmo che nessuno ci notasse... situazione schizofrenica o giù di lì!

Nascondere informazioni e in particolare comunicazioni strategiche è stato l'assillo di ogni condottiero dagli albori della storia. Peraltro lo stesso Marconi si corrucciava del fatto che le trasmissioni radio potessero essere captate da chiunque e quindi studiava continuamente nuovi metodi per offuscarne il contenuto. Mai gli era balenata l'idea che la sua invenzione avrebbe conosciuto il massimo fulgore proprio nella logica del broadcast, cioè uno trasmette e tutti ricevono. Non era nella cultura dell'epoca e non si capiva a cosa potesse servire una simile diffusione flat di trasmissioni radio.

Periodicamente c'è una celebrazione della macchina Enigma, la critto-decrittatrice in uso alla Wehrmacht tedesca durante il secondo conflitto mondiale. Vogliamo ospitare anche noi due contributi in questo senso: da una parte un simulatore free sul Web che può aiutare a fare piccoli esperimenti e dall'altro, per i più smanettoni, un progetto di replica del famoso meccanismo, fatto questa volta con l'ausilio di un microprocessore di quelli che vanno per la maggiore oggi: un ATmega. Anche se non ci pensate proprio di realizzarlo, capire come funziona una macchina Enigma è comunque un esercizio culturale interessante.

Buona lettura.

L'editoriale

4 *Riservatezza o schizofrenia?*

Come eravamo

8 *Olivetti 101 - una retrospettiva*

Retro programmi

12 *Borland Sidekick*

Retro linguaggi

18 *SNOBOL (parte 2) - Istruzioni principali*

Come eravamo

34 *M24, cane e gatto...*

Emulazione

52 *Emulare la macchina ENIGMA*

Retrocomputing

Espansione massima **6**

Il racconto

Automatik (25) - Rimini, Rimini! **14**

Darwin

Il libro dei perché del PC IBM (parte 2) **20**

Prova hardware

Exidy Sorcerer **24**

Laboratorio

Replicare ENIGMA con il chip ATmega 238P **38**

Mediateca

Trenta candeline per il primo portatile **48**

Espansione massima



di Tullio Nicolussi

E' una tendenza generale o almeno molto diffusa, quella che suggerisce ad ogni collezionista l'ampliamento del sistema che ha recuperato alla sua massima espansione.

Uno Spectrum 16K sembra non abbia dignità di esistenza se non viene subito ampliato a 48K, se non ci si procura l'interfaccia 1 e vi si attaccano due microdrive. Questo impulso soddisfa la naturale propensione di ognuno nel disporre della macchina più potente possibile. E' anche il desiderio di testimoniare le massime possibilità del sistema, come dire -"vedete quali incredibili possibilità aveva questo giocattolino?".

Magari avessimo potuto a suo tempo comprarle quelle espansioni!

La smania di espansione va oltre e talvolta sfocia nello stravolgimento del progetto originale. L'over-clock è il sistema più diffuso per "drogare" l'home computer e gloriarsi di avere "lo Spectrum più veloce del mondo!" Questo non ha molto senso, io credo. Quan-

do si colleziona un reperto storico bisogna come prima regola rispettarne la natura e non approfittare della disponibilità di chip più evoluti in grado di far fare un salto alla tecnologia originale.

Altre volte l'espansionismo è più sottile, più accettabile se volete, ma che merita una riflessione.

Prendiamo per esempio l'Apple II che è un sistema conosciuto da quasi tutti. La configurazione base dell'Apple //e è completa, nel senso che con un doppio floppy e monitor (che sono due opzioni "per modo di dire"), abbiamo tutto quello che serve. Praticamente tutti hanno poi aggiunto la scheda Z80, quella che sui listini veniva chiamata "Z80 Softcard", che da una marcia in più grazie al CP/M e i programmi disponibili, molti di tipo professionale.

Possiamo quindi dire che in un sistema Apple //e da collezione la sua bella Z80 Softcard ci sta benissimo. Spesso si legge sulle schede tecniche proprio la presenza del

secondo processore come una qualità intrinseca del sistema.

Un altro caso che vogliamo portare ad esempio, ma ce ne sarebbero molti altri, è un Olivetti M24 con una scheda che ospita il processore Z8000. Questo rende il diffusissimo computer della Olivetti, che è in pratica un clone MSDOS, in un sistema compatibile con l'M20 che è la macchina della generazione precedente. Questa scheda è stata progettata da Olivetti in fretta e furia, quando ci si rese conto che i programmi sviluppati per PCOS avevano una naturale vita che si estendeva ben oltre alla disponibilità della precedente Linea-1, della quale l'M20 è il "fratellino piccolo".

Come si verifica con la Z80 Softcard di Apple (fra l'altro sviluppata e commercializzata da Microsoft), anche per la scheda Olivetti si parla di estensione delle possibilità del sistema ma senza stravolgimento dell'hardware originale, che infatti si può continuare ad usare in maniera nativa.

Ad un certo punto della sua storia evolutiva l'Apple //e ha avuto un aggiornamento abbastanza significativo che l'ha portato alla configurazione "enhanced". Si tratta della sostituzione del processore dall'originale 6502 al nuovo 65C02, oltre che del firmware di sistema che aggiunge istruzioni, subito sfruttate dalla periferica mouse. Con questa aggiunta un //e diventa compatibile in pratica con il //c che appunto faceva del mouse e dei programmi di interfaccia punta e clicca una sua caratteristica.

La domanda: -"E' sensato aggiungere questo kit al reperto della nostra collezione e continuare a chiamarlo Apple //e?" ha due risposte, secondo me. La prima è che è legittimo se lo si cataloga come "Apple //e espanso con l'Enhancement Kit"; la seconda è che se ne avete uno solo di Apple //e, allora è meglio tenerlo "base" non fosse altro per affibiare al reperto una data più lontana,

seppure di pochi anni.

Va da sé che se ne possediamo due o più, cosa peraltro non rara vista la diffusione del sistema a suo tempo, si può con profitto aggiornare uno e disporre così di due oggetti diversi, entrambi storicamente collocati.

Ci sono però, come si accennava all'inizio, degli aggiornamenti che nulla hanno a che fare con l'originalità del prodotto. Ad esempio si prende un M24 e si mette al posto del processore originale una schedina che ospita un nuovo processore e magari anche della RAM compatibile con l'escalation che si intende fare e cioè pompare il clock del sistema a 20 MHz e oltre. Il gusto è vedere l'MSDOS "volare".

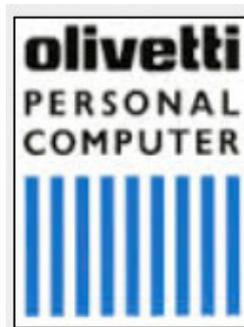
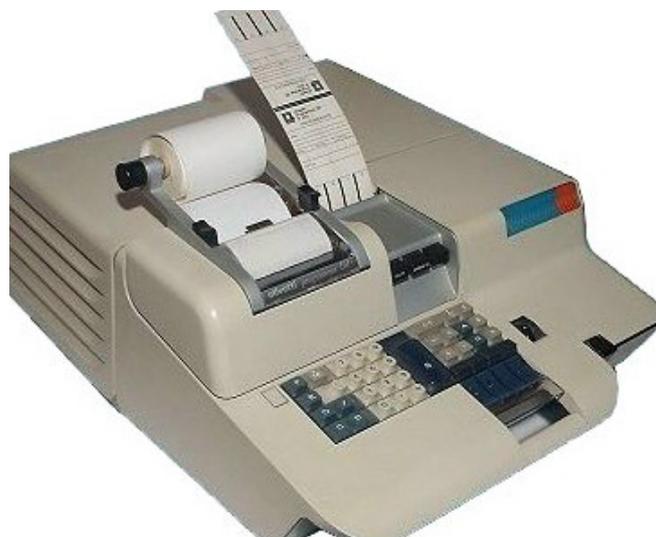
Diciamolo: va bene per uno smanettone che vuole togliersi uno sfizio e dimostrare non so che cosa, ma non ha senso per chi colleziona retro computer.

Quindi il messaggio conclusivo potrebbe essere:

"L'espansione di un retro computer dovrebbe essere limitata ai componenti originali sviluppati nella stessa epoca, possibilmente dallo stesso costruttore o approvati da esso".

(=)

Olivetti 101 - una retrospettiva



di Rodolfo Parisio - IW2BSF

Programma 101, o P101, è stato il primo personal computer al mondo, sviluppato dalla ditta italiana Olivetti negli anni tra il 1962 e il 1964 e prodotto tra il 1965 e il 1971.

Progettata dall'ingegnere Pier Giorgio Perotto (in omaggio al quale assunse il nome di Perottina) insieme a Giovanni De Sandre e Gastone Garziera, la P101 fu presentata per la prima volta nel 1965. Oltre che ad avere un disegno avveniristico, la P101 è stato il primo calcolatore commerciale ad essere digitale e programmabile, piccolo ed economico: il primo personal computer.

L'idea all'origine della progettazione della Programma 101, secondo il racconto di Perotto, muoveva dalla considerazione che all'epoca della sua uscita (presentazione alla Fiera di New York del 1965 come prototipo) si era completamente estranei al concetto di informatica distribuita, che comportava capacità di elaborazione e di immagazzinamento dati su un'unica macchina a disposizione dell'operatore e che, anche nel caso si fosse realizzata una macchina del genere, si riteneva essa dovesse essere più appannaggio di ambienti scientifici che non intesa come un normale strumento di lavoro quotidiano; la Programma 101 voleva quindi essere un prodotto di largo consumo capace di

venire incontro alle necessità operative del più alto numero possibile di persone.

L'azienda, che dopo la morte di Adriano Olivetti aveva puntato più sui sistemi di calcolo meccanici che non su quelli elettronici, aveva presentato quindi la Programma 101 in tono minore; tuttavia quando la mostra newyorkese aprì i battenti il nuovo calcolatore richiamò l'attenzione dei visitatori, i quali trascurarono tutti gli altri prodotti esposti dall'Olivetti nello stand. In aggiunta a ciò, la contemporanea Logos 27-A, calcolatrice elettromeccanica, anch'essa presente a New York, presentava problemi di produzione. La Programma 101 riscuoteva successo anche a Mosca (URSS) e successivamente, nel 1966, alla Fiera campionaria di Milano.

Questo spinse Roberto Olivetti, erede della dinastia, a farsi propugnatore di un tentativo di orientare la strategia aziendale in direzione dell'elettronica, obiettivo solo parzialmente perseguito e mai pienamente consolidato.

Per il lancio fu deciso il mercato americano, nonostante le perplessità circa i problemi eventuali di manutenzione dovuti alla mancanza di tecnici elettronici dell'Olivetti negli Stati Uniti. Essendo l'Olivetti completamente digiuna della fabbricazione in serie di apparecchiature elettroniche, il direttore di produzione dell'epoca pretese specifiche

dettagliate di assemblaggio senza tuttavia impegnarsi a collaudare alcun manufatto completato dalla sua linea di montaggio. In ragione di ciò Perotto, con due suoi collaboratori, si recò in fabbrica quando furono pronti gli imballaggi con i primi esemplari assemblati, e li aprì uno a uno per collaudarli personalmente ed eventualmente correggere errori; fu così possibile far partire per il Nordamerica un lotto di macchine senza problemi di funzionamento.

La produzione ebbe un notevole impulso quando la General Electric, che da qualche anno era in joint-venture con il ramo classico dell'Olivetti in una nuova società chiamata OGE (della quale gli americani detenevano il 75% del capitale sociale), manifestò la sua intenzione di uscire dal mercato dell'informatica. Questo spinse molti progettisti e ingegneri, che erano trasmigrati con tutta la loro struttura nella nuova società, a rientrare in quella parte della Olivetti, quella elettronica appunto, che l'azienda nordamericana a suo tempo non aveva voluto incorporare nella joint-venture e che, dopo l'im-

pennata degli ordini della Programma 101, si stava avviando ad essere il ramo vincente dell'azienda.

Le vendite ebbero talmente successo che alla fine del 1966 la Underwood, ditta americana controllata dalla Olivetti, chiese di poter fabbricare le macchine sul suolo degli Stati Uniti al fine di poter rifornire anche gli uffici delle amministrazioni federali di quel Paese.

Nel frattempo, già dal marzo 1965, era stato depositato presso il competente ufficio statunitense il brevetto sulle soluzioni tecniche adottate del calcolatore la manovra si rivelò essere opportuna perché la Hewlett-Packard produsse, sull'idea costituiva del P101, un analogo dispositivo di largo consumo, l'HP 9100A. Successivamente, quando le fu contestata la violazione di brevetto, la compagnia americana addivenne a un accomodamento extragiudiziale, riconoscendo a Olivetti un compenso a titolo di royalty di 900 000 dollari.

Il brevetto, che negli Stati Uniti, come richiede la legge, deve essere depositato a



nome del progettista e non della sua azienda, era stato ceduto da Perotto alla Olivetti per la simbolica cifra di un dollaro; al riguardo lo stesso Perotto affermò anni più tardi che «mai un dollaro fu speso meglio da un'azienda».

Dei circa 44.000 esemplari venduti, il 90 per cento dei quali sul mercato nordamericano, ne esistono soltanto 8 ancora funzionanti.

Il ruolo dell'Olivetti Programma 101 fu riconosciuto anche in tempi più recenti quando, nel maggio 2012, fu messo a confronto con altri due computer ritenuti pietre miliari della tecnologia informatica: lo statunitense APPLE, uscito nel 1976, primo esemplare fabbricato dalla Apple Computer, e il britannico Amstrad CPC, del 1984, primo home computer europeo di larga diffusione.

Il confronto, tenutosi al Politecnico di Torino, più che mettere in luce le differenze di prestazioni (evidenti per dispositivi nati in tre decenni successivi, e non comparabili con quelle ottenute con i successivi sviluppi dell'elettronica e dell'informatica), fu un omaggio allo spirito imprenditoriale che portò al concepimento di quelle tre macchine, della quale la Olivetti fu riconosciuta unanimemente come la progenitrice!

La denominazione di personal computer non va intesa tanto secondo l'accezione odierna quanto nel significato di macchina da calcolo per uso personale, che possiede un set di istruzioni interne ben definito, che lo classificano come computer, anche se la memoria per i dati temporanei e le costanti è molto limitata e l'informazione minima gestibile non è un valore binario ma un numero a 11 cifre.

Le istruzioni predeterminate erano quelle riguardanti:

- le quattro operazioni matematiche fondamentali (somma, sottrazione, moltiplicazione e divisione);
- la radice quadrata;

- operazioni con i registri: azzeramento, spostamento di dati tra di essi, lettura dall'utente;
- definizioni di etichette di riga, salti condizionati e incondizionati;
- stampa del valore di un registro.

La memoria era organizzata in 10 registri, 3 dei quali di calcolo, 2 di memoria e ulteriori 3 di memoria dati e/o memoria di programma (ripartibili a seconda dell'esigenza). Gli ultimi due erano riservati alla memorizzazione del programma. Uno dei registri, identificato dalla lettera "M", era destinato allo scambio di informazioni con la memoria e da e con le altre periferiche.

La stampa avveniva su un nastro di carta e i programmi potevano essere registrati su schede dalle dimensioni approssimative di 10 centimetri di larghezza per 20 di altezza che recavano due piste magnetiche. Tali piste erano leggibili una alla volta, inserendo la scheda nell'apposito lettore prima in un senso, poi nell'altro.

La memoria di lavoro era a linea di ritardo magnetostrittiva, della capacità di meno di un quarto di kilobyte; l'elettronica era realizzata in componentistica discreta (transistor in package TO-18 e diodi montati su basette in bachelite).

La macchina non era dotata di microprocessore. La Programma 101 lavorava con la precisione di 22 cifre a virgola fissa e 15 cifre decimali. La programmazione era analoga a quella con Assembler, ma più semplice: consentiva fondamentalmente lo scambio fra registri di memoria e registri di calcolo e le operazioni nei registri.

A titolo di esempio, il programma per calcolare i logaritmi occupava entrambe le facce di una scheda magnetica.

Caratteristiche tecniche

- Dimensione: 275 mm (A) x 465 mm (L) x 610 mm (P)
- Peso: 35,5 kg
- Consumo: 0,35 kW
- Dispositivo di output: stampante a 30 colonne su carta di 9 cm
- Precisione: 22 cifre e fino a 15 decimali
- Operazioni: somma, sottrazione, moltiplicazione, divisione e radice quadrata
- Memoria: circa 240 byte
- Archivio: lettore di card magnetiche

Design

Il progetto dello châssis, del quale fu delegato originariamente Marco Zanuso, fu successivamente affidato a Mario Bellini, all'epoca giovane architetto, quando ci si accorse che la soluzione di Zanuso prevedeva un ingombro intollerabile e incompatibile con le esigenze di uno strumento da scrivania. Bellini realizzò una struttura in alluminio profilato al fine di evitare interferenze con altre apparecchiature elettriche, e il peso finale di tutto l'apparato fu di circa 35 chilogrammi.

Alcuni esemplari di P101 sono tuttora esposti in musei come esempi di design innovativo.

Bibliografia:

Storia della Olivetti;
Adriano Olivetti;
Olivetti su Wikipedia;

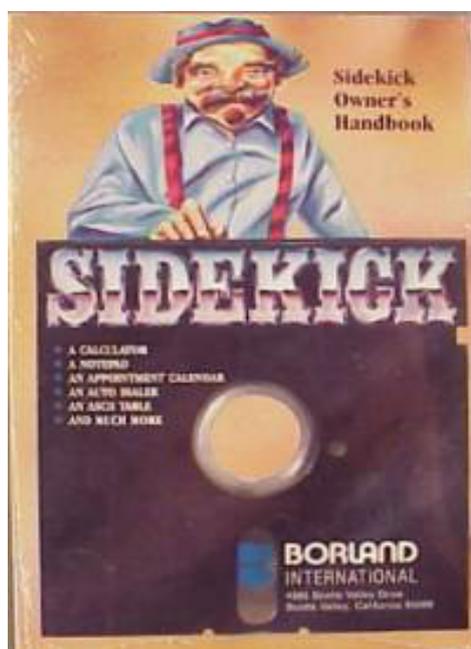
Riferimenti immagini:

- [1] - Wikipedia
- [2] - <http://www.oldcalculatormuseum.com/c-op101-2.jpg>
- [3] - proprietario: AlisonW - Licenza: reuse for non commercial;

(=)



Borland Sidekick



di Tullio Nicolussi

All'epoca facevo il programmatore. Avevo cominciato come tutti "i primigi" con le schede autocostruite che le varie riviste di elettronica piazzavano (e scopi piazzavano) e quindi con il linguaggio macchina. Poi le famose HP e Texas (meno le Texas, perché con le HP ci si poteva "sparare le pose" millantando la conoscenza della notazione polacca inversa ... una delizia!). Poi è stata la volta dell'Apple//e e dei calcolatori dipartimentali dell'Università. Non ero un "sinclairista" o un "64-ista", macchine che possiedo ora come collezionista, ma con il BASIC me la cavavo bene, l'Assembly era il pane quotidiano e Fortran e Pascal (ma anche il LISP) volavano sul Vax 11/760 della facoltà.

Negli anni '90 si andava di PC e quindi DOS e COBOL, visto che il mio lavoro era per clienti business e quando sei su una macchina monotask e stand-alone te la devi cavare senza "aiutini". Quindi a fianco della tastiera: blocco per appunti, listati e out-put di prova ammassati gli uni sugli altri; ovviamente i manuali del linguaggio, la tabella dei codici ASCII e l'immane calcolatrice esadecimale.

Questo set di strumenti era quello che più

o meno tutti usavano e peraltro esistevano anche in versione elettronica, cioè come programmi da far girare sul computer, ma non era pratico: dovevi comunque uscire dall'editor, lanciare il programma per visualizzare la tabella ASCII per esempio, poi lanciare la calcolatrice, poi rilanciare l'editor e ricominciare con il codice,...

Ecco che quei geni della Borland ebbero una idea grandiosa: sfruttare una delle caratteristiche "avanzate" del DOS che era la modalità TSR (Terminate and Stay Resident).

Senza entrare troppo nei particolari tecnici si tratta di sfruttare una caratteristica del DOS che è quella di mantenere aggiornato un puntatore all'area di memoria libera. Nella vecchia concezione di sistema operativo per le macchine home, dotate essenzialmente del BASIC e nulla altro, l'unico puntatore utile è quello che ti dice quanta RAM hai per i programmi dopo aver caricato l'interprete in memoria. La mappatura della memoria è rigida e quindi si sa benissimo (il BASIC lo sa) da dove inizia la RAM, caso mai non sa dove finisce perché la memoria RAM è proprio una delle caratteristiche che normalmente si potevano espandere negli

home.

Nel DOS (MSDOS, PCDOS, DRDOS) si può caricare un programma in memoria e fare in modo che quando termina la memoria stessa non sia liberata per il successivo. Il programma rimane appunto "residente" e questo meccanismo si può ripetere per n-programmi (non troppi! Mica che la memoria sia infinita...).

Va bene, direte voi, ma che me ne faccio di un programma che occupa RAM ma non sta "girando"? Già... Il DOS mica è multitasking, magari lo fosse!

E qui salta fuori un altro "trucco" del DOS che permette di rimandare in esecuzione il codice rimasto inattivo, poi terminarlo di nuovo e così via finché non si decide di spegnere il computer.

La Borland, stimatissima dai programmatori anni '80 e '90 (e peraltro ancora apprezzata da una nicchia di coder) confeziona un TSR che raggruppa le utilities più utilizzate da chi programma. Una combinazione di tasti le attiva in qualsiasi momento (un semplice CTRL-ALT) nel bel mezzo dell'istruzione che stiamo scrivendo e salta fuori un menù dal quale scegliere la calcolatrice piuttosto che la tabella ASCII. Si controlla quello che si deve e poi si riprende con l'editor a lavorare sul sorgente. Una grande comodità e chi l'ha provata questa feature non può che darmi ragione.

Il prodotto si chiama Sidekick e mette a disposizione, letteralmente sotto le dita del programmatore, una calcolatrice, un blocco note, la tabella ASCII, un calendario e, nelle varie revisioni, altre feature più o meno utili. Ad esempio non ho mai usato la funzione Phone-Call che permette, disponendo di un modem, di chiamare automaticamente un numero di telefono registrato nella rubrica. Credo che nemmeno funzionasse perché forse i telefoni non avevano la chiamata a toni... o no?

La Borland ha avuto l'intelligenza di preparare non una unica versione ma varie, tarate sulle necessità dell'utilizzatore. La memoria è infatti preziosa per il program-

matore all'epoca del DOS!

Ogni funzionalità è studiata con la massima attenzione. Il text editor denominato Notepad ad esempio ha i comandi compatibili con WordStar; la calcolatrice quando attivata setta in automatico la funzione Num-Lock del tastierino numerico, etc...

Un'altra caratteristica apprezzabile è la possibilità di spostare le corrispondenti finestre sullo schermo in modo da non coprire con l'attivazione delle aree utili alla funzione stessa.

Infine SideKick si può anche disinstallare dalla memoria, a patto che sia l'ultimo programma residente caricato. Quando devi lanciare un compilatore affamato di spazio è una funzionalità molto apprezzabile!

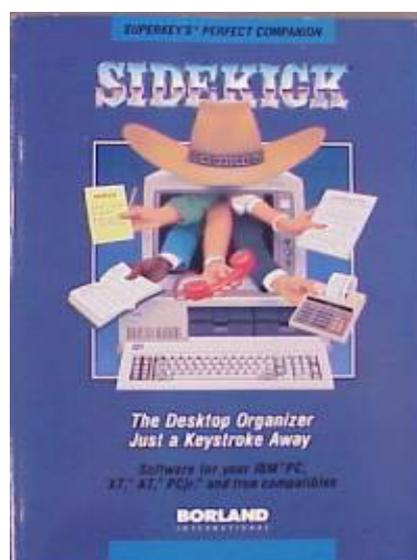
Conclusione.

Sidekick riporta all'epoca pionieristica della programmazione a linea di comando. Il suo esempio sarà seguito anche da altri: qualcuno ricorderà le parimenti utili "Norton Utilities".

Adattandosi poi all'avvento dei sistemi operativi a finestre, ne uscirà una versione per Windows 3.1 e poi una per Windows 95 e 98.

Ma a quel punto non erano più "piccole finestre con piccole funzioni" ma veri e propri ambienti di supporto alle attività quotidiane.

(=)



Automatik (25)

Rimini, Rimini!



Di Lorenzo Paolini

Dove si racconta della gita aziendale in Romagna.

Io e Daniele, il collega “anziano” della ditta Automatik, guadagnammo una vacanza pagata in quel di Rimini alla fine dell'estate 1984. Ufficialmente Romano, il titolare della “premiata ditta” di noleggio videogiochi ed affini, la presentò come regalia per i buoni incassi estivi e l'incremento dei ricavi nell'ultimo anno di lavoro.

In realtà, ma noi due lo sapevamo benissimo, doveva avere un forte sconto su quel pacchetto vacanza, forse lo aveva ricevuto gratis addirittura, perché recentemente aveva fatto qualche affaruccio con una agenzia viaggi che gestiva in proprio anche dei locali di intrattenimento nella zona dei laghi.

Comunque per noi andava bene così. Non è che era quella che si può definire “vacanza da sogno”, ma a caval donato..., come si dice...

Romano mise a disposizione anche il furgone cabinato della ditta per il viaggio. Ci guardammo meravigliati Daniele e io, ma l'arcano fu presto spiegato: il lunedì rien-

trando dovevamo fare una deviazione a Faenza e caricare certi giochi che aveva comprato da un collega conoscente che operava nella zona della Romagna.

Partimmo il venerdì pomeriggio perché, al solito, quando ci presentammo la mattina alle nove con borsa-viaggio, trovammo Romano che ci aspettava sul piazzale perché durante la notte precedente alcuni apparecchi si erano fermati e -“...Tenerli fermi tutto il weekend... Capite bene...”

Rassegnati facemmo il giro delle segnalazioni dividendoci il lavoro. L'appuntamento era alle 12 in ditta, io avrei lasciato il pick-up e Daniele mi avrebbe raggiunto con il furgone, si doveva trasbordare in fretta e poi via...

In realtà quando verso mezzogiorno stavo rientrando, trovai Daniele che mi aspettava in uno slargo poco prima dell'imbocco della stradina che portava in ditta. Mi spiegò che dovevo lasciare giù il pick-up e venire su a piedi senza dire a Romano, casomai lo avessi incontrato. Le cose andarono proprio così: lasciato il pick-up nel parcheggio davanti al laboratorio, mentre scaricavo la

borsa degli attrezzi e due sacchi di monete frutto degli incassi fatti durante il giro, si presentò Romano perché disse che :-"Bisognerebbe passare dal bar Giulia, perché...".

Non era mai finita!

Contrariato senza farlo vedere, assicurai il titolare che l'avrei detto a Daniele. Quindi presi la mia borsa da viaggio che avevo lasciato in laboratorio e mi avviai.

Daniele prese la cosa con filosofia e disse che lo sapeva e che perciò non era rientrato. Concordammo comunque che era meglio onorarla quella consegna, altrimenti saremmo stati inseguiti dalle telefonate di Romano per tutto il viaggio fino all'arrivo a Rimini. Per fortuna che all'epoca non c'erano i cellulari!

Prendere le cose con filosofia era la ricetta per vivere sereni in quella azienda artigiana che ruotava tutta attorno alle esigenze, che qualche volta sfociavano in manie, del titolare.

Comunque ci divertimmo arrivando a Rimini verso le sei di sera, prendendo possesso delle camere e poi via verso la serata che, pur nella stagione che andava scemando, rimaneva brillante e seducente ai nostri occhi di giovani provinciali.

Sabato e Domenica fantastici in pieno Sole che il Padreterno volle regalarci. Ci fu occasione anche per qualche conoscenza dell'universo mondo femminile, ma roba da spiaggia senza approfondimento alcuno. Giusto due chiacchiere di fronte ad un aperitivo.

Daniele si rivelò molto meno chiuso di quanto mi sarei aspettato da un "timidone venuto dal profondo Nord". Io invece cominciamo ad apprezzare la vita con poche preoccupazioni, qualche soldo in tasca e un futuro davanti che non vedevo chiaramente ma che sapevo piano piano si sarebbe rivelato.

La vacanza era spesa ma certo non potevamo presentare in ditta il rimborso spese

per l'ingresso in discoteca! Io avevo portato dei soldi che mi sembravano esagerati quando partii ma che non arrivarono alla Domenica mattina. Daniele invece, previdente e pragmatico giudicò opportuno ridimensionare il fondo cassa "segreto" che gestivamo per le "piccole spese personali extra lavoro".

Non avevamo quindi problemi di soldi ed offrire aperitivo e gelato alle ragazze in spiaggia era un piacere! Fra l'altro capitammo proprio durante una non so che riunione della "Gioventù Cattolica" o qualcosa del genere (non mi ricordo se all'epoca c'era già il meeting di CL), per cui possiamo dire che proprio "non c'era scarsità di materia prima".

Quando rivedo in un film con Totò la scena di lui e Peppino al night che offrono champagne alle ragazze del balletto, beh mi sembra di averla vissuta anch'io quella scena, seppure con le dovute differenze.

Comunque non combinammo "guai", si fa per dire, e il lunedì lasciammo le spiagge assolate e cariche di promesse (ah se avessimo avuto una intera settimana...!) e ci inoltrammo verso l'interno.

Eravamo partiti presto, verso le nove di mattina da Rimini, perché l'idea era di caricare prima di mezzogiorno, fermarsi a mangiare e poi prendercela comoda nel rientro. Daniele disse che voleva passare da Ostiglia dove una sua amica sposata abitava, per salutarla.

Gli chiesi se si era messo d'accordo per la visita ma lui mi rispose semplicemente: -"Che problema c'è? Conosco l'indirizzo...". Non ribattei che mi sembrava pochino...

Ovviamente a Faenza faticammo a trovare il tizio dei giochi (forse qualcuno se la ricorda l'epoca senza i navigatori) e quando cominciammo a caricare scoprimmo che non erano due-tre cabinet come aveva detto Romano, ma almeno dieci-dodici! Cioè sul

furgone non ci stavano a meno di non farli a pezzi... cosa che proposi a Daniele e che lui prese come una presa in giro a suo indirizzo. Era permaloso a volte.

Comunque la sua esperienza nel carico/scarico si rivelò preziosa e dopo due ore buone di lotta riuscimmo a chiudere il portellone del furgone carico come un uovo. Erano le due del pomeriggio passate, addio pranzo tipico in un ristorante romagnolo! Trangugiammo l'indigesto panino da Autogrill mentre si viaggiava verso Nord.

Daniele voleva assolutamente andare a Ostiglia, lo aveva promesso a quella sua amica, sostenne. Io cercai di protestare temendo di dover arrivare alle dieci di sera e dover anche scaricare il furgone. Daniele però si ostinò e la dichiarazione che non saremmo passati in ditta quel giorno mi tranquillizzò e mi predisposi a godermi quella variazione della gita. Non ero mai passato da Ostiglia in vita mia! Riflettevo su quanti posti non si vedono nella vita, magari passandoci accanto per andare altrove. Ovviamente la fantozziana nuvola da impiegato ci mise del suo e beccammo sulla provinciale uno di quegli acquazzoni che ti fanno credere che per quell'anno la bella stagione sia finita.

Ovviamente Daniele non aveva avvisato la sua amica per cui ci presentammo insalutati ospiti accolti da un cane con due fauci enormi che non lasciavano dubbi su chi avrebbe vinto nel caso di una inopinata lotta. Franca, l'amica di Daniele era in casa, una villetta bassa in mezzo a macchie di canneti che mi fecero supporre che il Po non fosse lontano. Non si può dire che l'accoglienza non fu gentile, ma si vedeva chiaramente che era abbastanza imbarazzata dalla visita e non sapeva come comportarsi. C'era anche il marito e avevano una figlia piccola che ricordo graziosa ma petulante e avevano anche un maschio più grande che non era in casa. Io ero l'estraneo fra gli estranei e giudicai opportu-

no non conversare più di tanto ed osservare la situazione. Chissà che relazione era passata fra Daniele e Franca; erano stati amici o qualcosa di più? Boh, non riuscii a capirlo ma certo lei era molto sulle spine e spiava di sottocchi il marito che era cordiale sì, ma forse un tantino guardingo.

In una pausa della conversazione, peraltro banalissima, riuscii a passare al mio collega l'idea di andarcene prima che si sentissero obbligati ad offrirci di restare per la cena. E' stata una delle visite di cortesia peggiore fra quelle che ho mai avuto modo di fare nella vita, comprese quelle per le condoglianze!

Così si concluse la nostra prima e unica vacanza aziendale, con un lavoraccio al mattino seguente a scaricare il furgone e rispondere alle domande del titolare su dove ci eravamo fermati, cosa avevamo fatto a Rimini e via indagando. Da una parte Daniele minimizzava sulle ore passate in Romagna, dall'altro mentiva spudoratamente parlando di una gita a San Marino che mai avevamo fatto ma che prova di veridicità lui esibì uno straccio da cucina stampato con lo stemma della Repubblica e con il calendario dell'anno in corso. Mi disse poi che l'aveva preso su una bancarella in spiaggia per avere qualcosa da raccontare che fosse normale per la famiglia del nostro titolare. Si spinse spudoratamente anche a farne omaggio alla signora che gradì, nonostante fosse proprio uno straccio da pochi soldi. Misteri dell'animo femminile!

La raccolta dei giochi in quel di Faenza non era finita perché venimmo a sapere che il nostro principale stava organizzando l'apertura di una sala giochi in grande stile in una località turistica invernale. Servivano almeno venti cabinet e giochi abbastanza buoni, soprattutto novità. - "Quella è gente che viene da Milano" diceva, vengono a sciare ma vogliono divertirsi e riempiranno le gettoniere di bei soldini".

Il lunedì successivo eravamo nuovamente in viaggio per la pianura del Po per caricare altre macchine. Altra furgonata strapiena e altre ciofeche da pulire, riparare e rendere in qualche modo presentabili. Dovevamo prendere anche dei flipper in questo secondo viaggio ma non c'era posto. Il flipper occupa un sacco di spazio e anche se gli puoi smontare le gambe di acciaio, il pianale è delicato con quel vetro così ampio e la testata ingombrante. Si prospettava un ulteriore viaggio la settimana successiva ma ormai eravamo "di casa" in Romagna e i buoni ristoranti tipici acquistarono due clienti.

Al rientro dall'ultimo viaggio Daniele girò verso Ovest e mentre io credevo mi portasse poco lontano in qualche altro ristorante prossimo all'autostrada, benché fossero appena le quattro del pomeriggio, in realtà uscì addirittura a Piacenza! -"Conosco un posto" fu la sua lapidaria risposta alle mie domande. In effetti il posto c'era e non era cambiato in dieci anni, mi diceva; gli stessi titolari e la stessa cucina squisita. Ricordo il piatto "Pitarè de fasò" (o qualcosa che suona simile) che poi era una pasta e fagioli che meritò un bis da applausi!

Incredibile: avevamo allungato il viaggio di trecento chilometri almeno e impiegato mezza giornata più del previsto ma Romano non parve accorgersene per nulla! Avrebbe potuto forse dalla carta carburante o dai pedaggi autostradali, ma invece niente. Peraltro il pasto fu offerto dalla "Cassa per il Mezzogiorno", nostra personalissima banca, per cui non si dette adito alla sua consorte che teneva i conti e rimborsava le spese, di alcun sospetto.

Io credo che Romano comunque sospettasse le nostre "marachelle" ma forse giudicava opportuno fare finta di nulla. In fondo credo che ci volesse bene a suo modo e senza ombra di dubbio ci giudicava preziosi per l'azienda. In fondo lavoravamo il doppio o quasi delle ore sindacali e gli portavamo a casa dei bei

sacchetti di moneta ogni santo giorno. Che poi sembra che la moneta pesi molto ma valga poco, ma quando cominci a svuotare un sacchetto di moneta da 200 Lire ne conti sui 1500 di pezzi, il che vuole dire 300.000 Lire! Buttale via! ALL'epoca erano metà del mio stipendio!

Le macchine conta-soldi installate in ufficio giravano senza posa ogni mattina e il rumore, che mi sembra di sentire ancora, era il segnale che la giornata lavorativa era cominciata.

Poi tre volte in settimana o il titolare o la moglie partiva con "il bottino" che andava a depositare in banca credo, ma avevano anche un certo giro con negozi ai quali fornivano la moneta ricevendone in cambio valuta cartacea che ovviamente non veniva dichiarata in nessun posto se non forse sulla famosa agenda di Romano.

Io non ho mai visto la cassaforte aperta che stava dietro una specie di libreria in ufficio, ma Daniele mi assicurava che era piena di soldi. Io credo che nessuno, se non forse gli addetti ai lavori, sospettasse il giro di denaro che stava dietro una attività che sembrava "da poveri cristi", altrimenti un "ben intenzionato" avrebbe trovato più soddisfazione nella cassaforte della Automatik piuttosto che nel rischio di assaltare una banca.

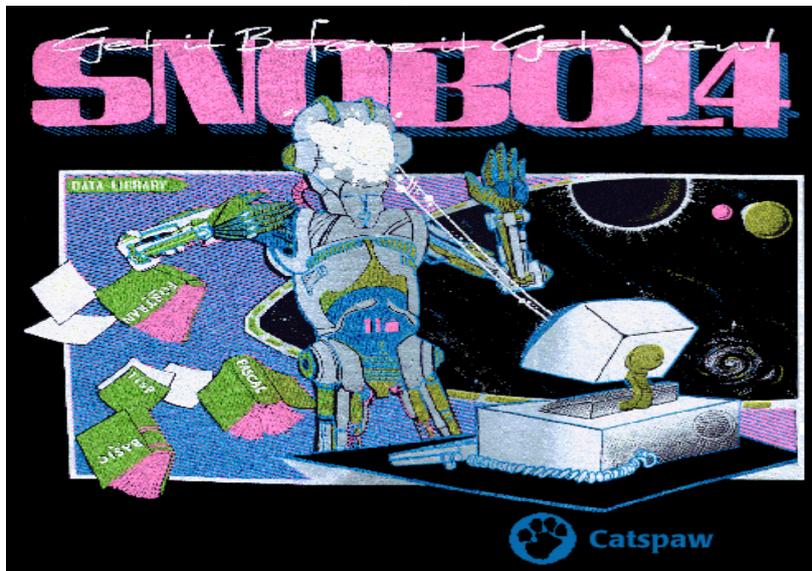
Io ero un po' in apprensione le prime volte che Daniele variava il programma stabilito con il capo o che "pescava" dalla famosa "cassa aperta", ma lui faceva spallucce e mi redarguiva con uno dei suoi moti popolari "Meglio una rampognata che una lavorata!".

Mi manca Daniele e quelle giornate assieme. Si lavorava, ci si arrabbiava e si litigava anche qualche volta, ma le risate che ci siamo fatte sono indimenticabili!

Alla prossima avventura!

SNOBOL (Parte 2)

Istruzioni principali



di Salvatore Macomer

Dopo aver parlato delle origini del linguaggio e averne introdotto gli elementi filosofici e logici, affrontiamo gli aspetti pratici. Faremo riferimento all'implementazione denominata SNOBOL4, che è la più comune, nel senso che gli interpreti o compilatori che si trovano oggi rispettano questo standard.

Un programma in SNOBOL è un insieme di istruzioni con una sintassi ben definita, che vengono eseguite in sequenza. Il programma inizia con la prima istruzione e termina con l'istruzione END.

Le variabili possono essere intere, reali o a stringa (anche se il tipo "stringa" è in realtà una struttura più complessa, ma per analogia comune ad altri linguaggi si può pensare che sia una semplice sequenza di caratteri).

L'assegnazione del valore ad una variabile avviene con il simbolo "=":

`V = 5`

Assegna alla variabile V il valore intero 5.

Il linguaggio supporta anche l'assegnazione calcolata:

`RESULT = 5 * (3 + 11)`

Le variabili reali sono dichiarate tali perché viene assegnato loro un valore reale (cioè con decimali); ad esempio:

`PI = 3.14159`

Alle variabili stringa si assegna il valore racchiuso fra apici singoli:

`STRINGA = 'DOG'`

La dichiarazione di una variabile con nessun valore assegnato si intende essere una stringa vuota:

`STRINGA_NULLA =`

Una caratteristica particolare dello SNOBOL è che se una variabile stringa contiene

un numero (intero o reale), allora può essere usata nelle espressioni matematiche:

```
S = '10.1'  
R = 5 * 2 + S
```

Che assegna alla variabile R il valore 20.1 come risultato del calcolo.

Le funzioni che operano sulle stringhe sono numerose e molto intuitive. Vediamo la concatenazione di due stringhe:

```
S1 = 'TOPOLINO'  
S2 = 'PIPPO'  
S3 = S1 ' E ' S2
```

Che restituisce in S3 la stringa 'TOPOLINO E PIPPO'.

Tre variabili riservate individuano le operazioni di I/O:

INPUT per l'inserimento dei dati da tastiera, OUTPUT per l'emissione a video e PUNCH per la stampa (il nome è un chiaro riferimento vintage ai tempi in cui c'erano i nastri perforati).

Quindi ad esempio:

```
RISULTATO = INPUT  
OUTPUT = RISULTATO  
PUNCH = RISULTATO
```

Il primo statement assegna alla variabile RISULTATO il valore inserito da tastiera che poi viene mostrato a video e stampato.

Lo SNOBOL, come si diceva nella scorsa puntata, nasce per la manipolazione delle stringhe e il suo punto di forza è il principio del pattern matching. In realtà SNOBOL4 implementa due paradigmi: il pattern matching propriamente detto, cioè una determinata stringa è in una qualche relazione con un pattern? L'altro paradigma è quello della sostituzione, cioè sostituire un subset di elementi in una stringa con un determinato pattern.

Vediamo un esempio semplice di pattern matching.

```
PAT = 'GRAM'
```

```
STRINGA = 'PROGRAMMER'  
STRINGA PAT
```

Qui abbiamo le due stringhe PAT e STRINGA contenenti i valori assegnati.

Il terzo statement è la richiesta di pattern matching. Se nel valore di STRINGA è contenuto il valore di PAT, allora lo statement risulterà vero, altrimenti il pattern matching fallirà.

Il replacing è altrettanto semplice come sintassi:

```
WORD = 'PAPERINO'  
WORD 'O' = 'A'
```

dice che nella parola WORD, cioè nella stringa 'PAPERINO' ogni occorrenza della lettera 'O' va sostituita con la lettera 'A'. Al termine dell'esecuzione quindi nella variabile WORD sarà contenuto il valore 'PAPERINA' e l'esecuzione restituirà True per indicare che il replacing è riuscito.

Ogni statement dello SNOBOL restituisce la condizione logica di riuscita o fallimento. E' quindi possibile decidere dopo ogni statement il da farsi, cioè saltare ad una certa label in relazione al risultato dell'operazione.

Ad esempio:

```
PAROLA = INPUT  
PAROLA 'YES' :F(ERR)  
OUTPUT = 'HAI RISPOSTO YES!'  
GOTO FINE  
:ERR OUTPUT = 'HAI RISPOSTO NO!'  
:FINE END
```

con :F(ERR) abbiamo detto al sistema che a fronte del fallimento dell'operazione di pattern matching, si salta alla label che stampa l'errore, altrimenti si continua con lo statement successivo.

Bene, fermiamoci qui e ripartiremo la prossima volta affrontando problemi un po' più complessi ma che grazie alla natura dello SNOBOL saranno risolvibili con poche istruzioni.

(=)

Il libro dei perché del PC IBM

(parte 2)



di Riccardo Franch

Burton Grad e Mike Uretsky stilavano nel 1979 su incarico della IBM uno studio sulle prospettive di mercato relativamente al comparto dei personal computer. Il loro lavoro di consulenti consistette nell'analizzare i prodotti di successo presenti ed isolare i motivi che li avevano resi tali. Considerarono il caso dell'Apple II e dall'analisi risultò evidente che era l'accoppiata software + hardware a costituire l'unicum in grado di realizzare i risultati del marketing. "Prima della disponibilità di Visicalc" - afferma Uretsky - "l'Apple II era un giocattolo. La vera rivoluzione fu il foglio di calcolo che rendeva il sistema utile per le applicazioni business. La gente comprava un Apple II per farci girare sopra Visicalc!".

Era un cambiamento di prospettiva: finora il computer veniva scelto ed acquistato

per la potenza di calcolo che poteva erogare; con l'Apple II si comprava l'hardware solo per la presenza di un certo software.

La IBM doveva quindi uscire con una propria soluzione il più possibile aperta, facilmente programmabile e il più presto possibile!

Ma facciamo un passo indietro.

La IBM stava "giocando" da qualche anno con i microprocessori e con l'idea di costruire un prodotto per l'elaboratore personale. Nel febbraio del 1978 partì il progetto denominato "DataMaster", poi messo a listino con il nome di System/23. L'idea era costruire un elaboratore personale programmabile in BASIC ma che rispettasse una "linea" ben precisa, cioè non si ponesse in alter-

nativa all'elaborazione su mainframe ma piuttosto la affiancasse o andasse a coprire quella fascia bassa del mercato del "piccolo business" che sembrava farsi interessante.

David J. Bradley lavorò nel team del progetto con il compito di scrivere il software di base e di adattare una versione di BASIC che era usata sul Sistema/34. Da lui abbiamo una testimonianza in prima persona di questa "avventura"^[3].

Il DataMaster era un computer basato sul processore Intel 8085 con 64 Kbyte di RAM di base ma che il team riuscì ad estendere fino a 256 Kb con una tecnica di paginazione (che la IBM registrò come brevetto). Due floppy da 8 pollici, schermo e tastiera in un corpo unico costituivano la macchina.

Come si vede dalla foto sotto, si tratta di un sistema "massiccio" e sicuramente ingombrante.

David racconta che si imparò molto dalla realizzazione del DataMaster. Due cose in particolare: tastiera e video dovevano essere separati dal corpo centrale e era meglio prendere un BASIC già pronto piuttosto che mettere in conto un anno di lavoro per adattarne uno esistente come era successo per il DataMaster.

Altro tassello importante era il sistema operativo. Alla IBM ne avevano molta esperienza ma erano realizzazioni sofisticate, poco

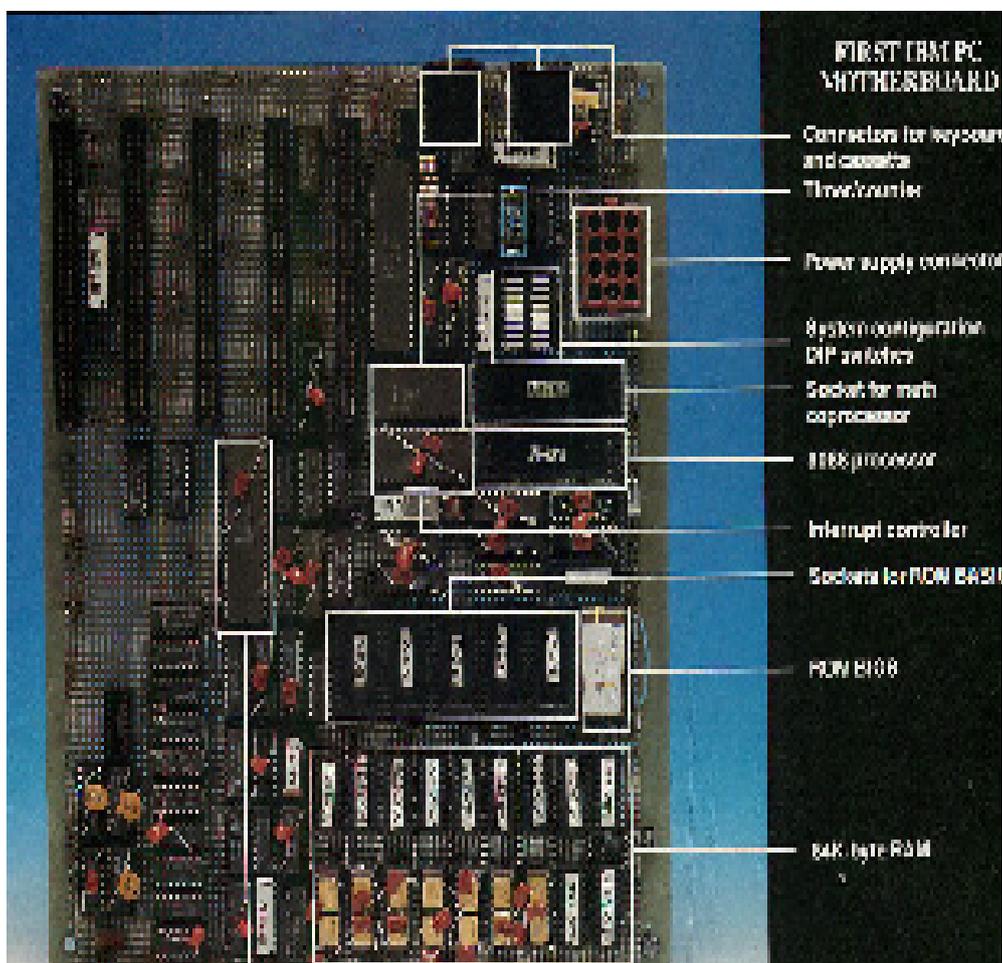
adatte ad un sistema che presumibilmente doveva essere acceso e pronto in cinque minuti.

Dall'agosto 1980 fino all'annuncio del PC 5150, si lavorò alacremente al progetto e David venne impiegato nella scrittura dei driver per le componenti del sistema, a cominciare dal controller floppy da 5,25".

Il prototipo cambiava praticamente ogni giorno e si giunse ad una versione stabile realizzata su piastra millefori dalle dimensioni di quella che poi sarà la motherboard del PC, con i collegamenti realizzati con la tecnica del wrap-up (vedi figure 3 e 4).

Dal suo predecessore il PC ereditò la sezione video "business", realizzata con la scheda MDA (25 righe per 80 caratteri). Il chip di controllo video era un Motorola 6845, preferito all'Intel 8775 usato nel dataMaster. Fu preferito il chip di Motorola perché poteva indirizzare maggiore memoria video e la IBM aveva messo come requisito necessario il supporto ai 256 caratteri necessari per l'internazionalizzazione del sistema. La scheda MDA conteneva inoltre una porta parallela per il collegamento con una





stampante. L'idea dell'home computer era comunque nell'aria, così fu resa disponibile fin da subito la scheda CGA che poteva uscire anche sul TV, viste le basse richieste hardware cui necessitava per visualizzare 25 righe da 40 caratteri o 320x20" pixel in alta risoluzione. L'idea del computer home era in IBM vista non tanto come la macchina per giocare o per imparare la programmazione, quanto una specie di appendice del sistema dell'ufficio: cioè potevi lavorare a casa sugli stessi programmi, ad esempio sul foglio di calcolo che avevi iniziato in ufficio.

La versione "home" non disponeva di floppy ma come in uso allora, si poteva usare il registratore a cassette audio. Il BASIC era in ROM (16 Kb) e altri 16 Kb di RAM costituivano la configurazione minima acquista-

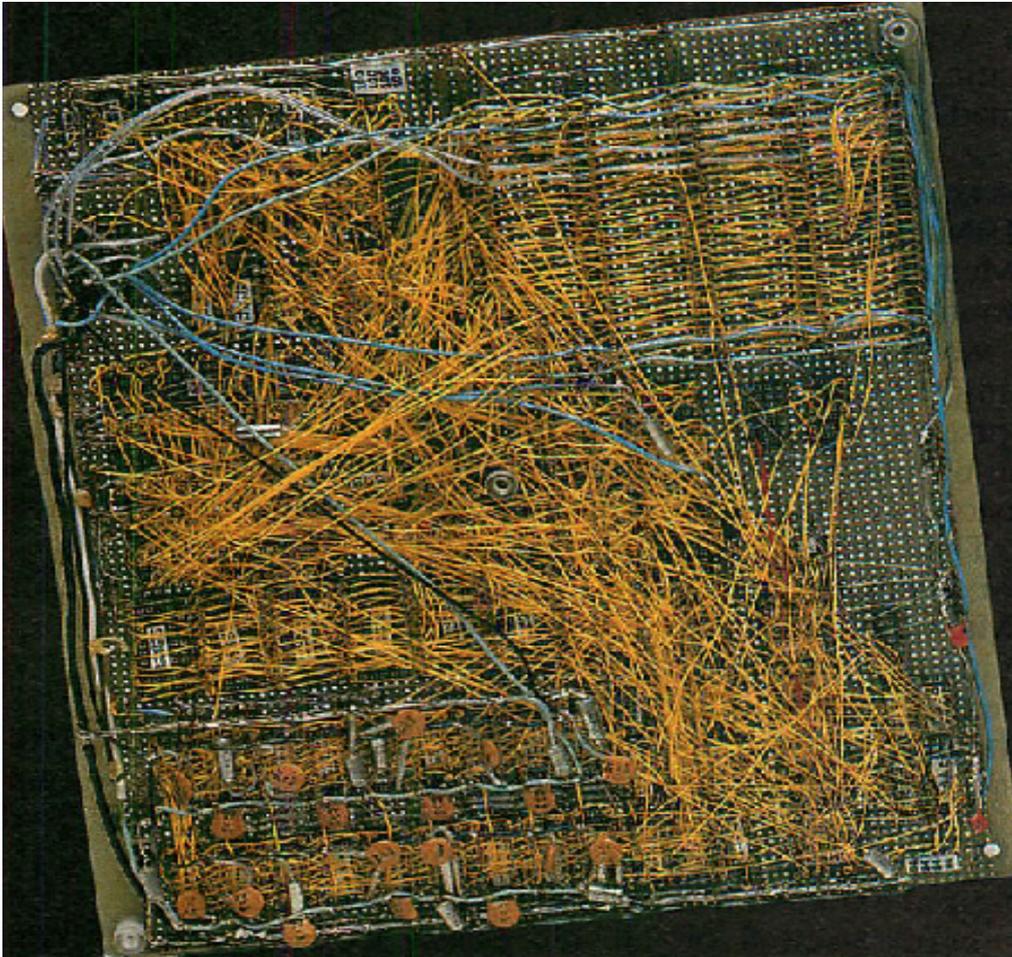
bile.

Dal DataMaster il PC ereditò anche l'architettura del bus e gli stessi slot di espansione, pur modificati come pin-out per supportare il maggior indirizzamento possibile con un 8088 rispetto all'8085.

La questione del limite dei 640 Kb indirizzabili dal DOS è ben nota ed è stata oggetto anche di scherno quando le architetture hanno permesso molto più spazio di indirizzamento. L'8088 poteva indirizzare solo un megabyte ed è stato mappato come meglio si pensava.

La memoria aveva il controllo di parità, cosa indispensabile per costruire all'epoca una macchina affidabile.

La stessa tastiera del PC è quella del precedente progetto con alcune scritte modifi-



cate sui tasti funzione e con il collegamento seriale anziché parallelo presente nel predecessore. La famigerata sequenza CTRL-ALT-DEL venne scelta perché sulla tastiera erano tasti quasi impossibili da premere per errore all'unisono e permisero di rinunciare al pulsante di reset che dava l'impressione di poca stabilità, impressione che la IBM non poteva accettare per le sue macchine!

(...continua...)

Riferimenti bibliografici.

[1] - Edward Bride - *The IBM Personal Computer: A Software-Driven Market* - in *COMPUTER, COVER FEATURE*, Published by the IEEE Computer Society 0018-9162/11 © 2011 IEEE;

[2] - <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/>;

[3] - David J. Bradley - *The creation of the IBM PC* - in *BYTE* Settembre 1990;

Riferimenti alle immagini.

[1] - "IMB PC-IMG 7271" by Rama & Musée Bolo - Own work. Licensed under CC BY-SA 2.0 fr via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:IMB_PC-IMG_7271.jpg#/media/File:IMB_PC-IMG_7271.jpg;

[2] - <http://oldcomputers.net/pics/ibm5322.jpg>;

[3-4] - *BYTE* - Settembre 1990;

Exidy Sorcerer



di Tullio Nicolussi

Prima del 1980, cioè prima che si potesse parlare di “mercato del Personal Computer”, ogni progetto nel settore dell’informatica “casalinga” era una specie di salto nel buio. Le aziende che in qualche modo erano nel settore dell’elettronica digitale (calcolatrici, console da gioco, ...) provavano a realizzare una macchina di questo genere confidenti che qualche pezzo lo si poteva vendere in quantità profittevole. Dalla loro parte anche la relativa facilità di progettazione, per la quale bastava anche solo un ingegnere valido o comunque un team di persone ristrettissimo, purché motivato. Molti progetti sono nati da idee singole maturate da persone curiose di realizzare qualche cosa da soli. Basta pensare a Steve Wozniak, ideatore dei primi Apple per rendersi conto che questo accadeva!

Non si deve però pensare che il progetto di un sistema funzionante fosse banale come ci vogliono far credere i testi che troviamo oggi in commercio. Quei primi esploratori dovevano derimere innumerevoli problemi sia di programmazione che banalmente

hardware, come la disposizione delle piste sul circuito stampato, il dimensionamento di ogni singolo componente, la lotta alle interferenze, etc...

Di molti tentativi rimane poca traccia, alcuni sono assolutamente perduti, altri esistono in pochi esemplari. Ecco un’altra prova che deve essere reso onore a coloro che, appassionati del genere, ne hanno curato la sopravvivenza nelle loro personalissime collezioni.

Nel 1978 il mercato era agli albori ma qualche macchina di calcolo personale c’era e apparentemente era un affare per chi la produceva, anche se forse un “affare in divenire”. C’era il Commodore PET e c’erano il TRS-80 e l’Apple II... macchine oggi ben note e ritenute capostipiti della generazione dei Personal. Uno solo di questi prevedeva una uscita a colori (l’Apple) che in qualche modo poteva sovrastare i concorrenti nel campo dell’intrattenimento, ma l’Apple era particolarmente costoso e la sua diffusione non così ampia come i due concorrenti che potevano disporre di un marketing molto

più carrozzato e di conseguenza erano più conosciuti e diffusi.

Aldilà di meriti e demeriti tecnici, questi sistemi avevano in comune un prezzo alto, sopra i 2000 dollari, che tradotto in Lire di allora si parla di 4-5 milioni: come una utilitaria!

In questo scenario la Exidy propone nel 1978 un sistema a basso costo (circa 800 dollari) basato su uno Z80 con una tastiera estesa, uscita per monitor videocomposito e slot di espansione per cardtridghe ROM oltre la consueta uscita cassette audio. Il nome della macchina è **SORCERER** con chiaro intento di promuoverne le caratteristiche alle future "scoperte" nel nuovo mondo dell'informatica personale.

Quali erano le pecche dei sistemi concorrenti che la Exidy proponeva di superare? Il PET aveva una tastiera da macchina calco-

latrice più che da telescrivente: una tastiera "single finger", cioè da usare con un dito e ciò lo rendeva adatto al data-entry ma poco allo sviluppo alfanumerico delle applicazioni. L'espansione era possibile ma l'interfaccia IEEE488 rendeva la cosa particolarmente costosa e destinata più ai sistemi di laboratorio che all'uso domestico con periferiche a basso costo che venivano proposte con interfaccia parallela o seriale RS232. Il TRS-80 veniva giudicato ingombrante per via del suo video da 12" inglobato nello chassis, la tastiera era ok ma senza tastierino numerico e infine l'uscita video era povera in termini di possibilità di righe di caratteri e conseguentemente poco sfruttabile nella grafica. Per entrambi il BASIC in dotazione, successivamente ampliato, consentiva poche se non nulle possibilità di applicazione nel videoludico e mancava ad esempio nel



TRS la gestione del suono.

Prendendo il meglio dal PET e dal TRS-80 la Exidy assemblò un sistema "best-fit" rispetto ai desiderata dell'utente personale proponendolo ad un prezzo base particolarmente appetibile, anche se con soli 8K di RAM. Il sistema espanso a 32 Kbyte di RAM costava quasi 1400 dollari, sempre comunque significativamente meno della concorrenza.

Notiamo che la pubblicità faceva leva su messaggi rivolti ad un target preciso: i giovani studenti. L'immagine pubblicitaria a fianco è un coacervo di stereotipi: lui che porta la macchina disinvolatamente (in realtà il computer ha un certo peso!) mentre approccia una lei, studentessa sorridente e lusingata che tale "bel figo" le rivolga attenzione. Notate l'ambientazione casual e l'abbigliamento dei due stile "campagnolo", il fatto che il sistema Sorcerer sia in primo piano, lei che porta dei libri, lui il computer... Niente da dire, per quanto me ne intendo di comunicazione una immagine ben fatta!



Primo approccio al sistema.

Il computer **SORCERER** di Exidy si propone in un contenitore-tastiera compatto che gioca sul contrasto nero-beige con sfumatura ocra per i tasti. Ne viene fuori una macchina "gialla" che non sarebbe molto accattivante per il gusto post-2000 ma che era accettabile e addirittura definibile "bello" nel 1980.

La tastiera dà una immagine lineare alla sezione input senza tasti aggiuntivi in giro e con un tastierino numerico a 16 tasti dove è "annegata" anche la croce di movimento del cursore. Le lettere bianche su sfondo giallo/ocra non sono il massimo per visibilità, cosa della quale pare se ne accorga il produttore che nella seconda release del sistema opta per una più classica e visibile tastiera a sfondo nero. Ai tasti "caldi" RESET, CLEAR, RUN





è riservato il colore scuro, così come al tasto GRAPHIC nella parte sinistra, tasto che da accesso ai simboli semigrafici e di punteggiatura stampigliati sui tasti stessi.

Dal punto di vista geometrico e del colore, il contenitore assomiglia molto a quello di un Apple II ma è proporzionalmente più piccolo, eccetto che per la larghezza che è la stessa, dato che è "obbligata" dalla dimensione della tastiera. Gli angoli del cabinet sono smussati



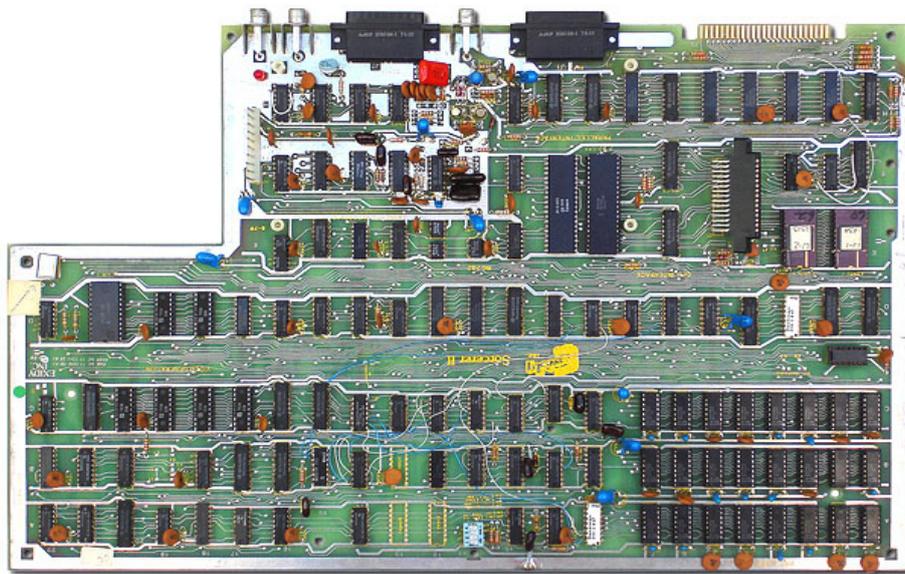
e la plastica del contenitore non è liscia ma ha un effetto "buccia d'arancia" irregolare. Notiamo che anche il colore non è uniforme ma questo potrebbe essere un effetto

dell'invecchiamento. Le foto che si trovano su Internet danno l'impressione che originariamente fosse molto più chiaro, ingiallito dal tempo. Il ripiano superiore non è abbastanza grande da ospitare un monitor in maniera stabile ed è occupato, ma non in tutte le revisioni, dalla stampigliatura del layout della tastiera dove l'utente può volendo scrivere la corrispondenza dei tasti, ad esempio quelli usati nella programmazione BASIC.

Un evidente logo della ditta completa l'aspetto esterno di questa macchina che dà subito una sensazione di robustezza e di equilibrio fra le varie parti.

Esaminandolo da tutti i lati troviamo sul fianco destro la fessura per l'inserimento delle cartridge ROM. Questo slot presenta una particolarità: le cassette si inseriscono obliquamente verso il basso invece che orizzontalmente alla base, una particolarità che non mi pare sia comune ad altri progetti.

Sul fianco sinistro troviamo una finestra protetta da una griglia bucherellata, serve ovviamente per lo smaltimento di calore, così come le fessure nella parte superiore del cabinet (bei tempi quando bastava arieggiare



passivamente per smaltire il calore prodotto dai componenti sulla piastra!).

Sul retro i connettori di espansione. Da sinistra un pettine “Expansion Bus” che scopriremo poi essere compatibile con il Bus-100, l’interfaccia parallela, il connettore coassiale per l’uscita video-composito, la RS-323, due connettori per altrettanti registratori audio ed infine la sezione alimentazione con il tasto di accensione e il cambia tensione.

Hardware.

La mother board nella seconda versione è una piastra unica ben ordinata e razionale. La sezione alimentazione è stata stralciata e posizionata a parte: cosa quantomai opportuna! La prima versione è molto diversa come layout; in comune diciamo che hanno solo la posizione delle uscite (evidentemente il produttore non voleva cambiare il cabinet).

Lo Z80, cuore del sistema, viene cloccato a 2 MHz e con questo settaggio le prestazioni di calcolo sono quelle comuni agli altri personal dell’epoca anche con processore diverso. Del resto non c’erano molte possibilità per ottimizzare le prestazioni di una macchina per battere la concorrenza con eclatanti risultati prestazionali!

Si individuano le componenti principali: il processore, il chip di I/O, la sezione audio/video e l’ordinata matrice dei chip di memoria. Nella prima versione la memoria era montata su una piastra separata da agganciare attraverso un connettore; non è dato saperlo ma probabilmente questa scelta è stata dettata dall’idea di espansione mentre nella seconda versione probabilmente il sistema nasceva con massima disponibilità di RAM, cioè 32 Kb.

Il connettore “User’s Port” è un pettine da 25 pin in doppia faccia, come previsto dallo standard S-100 per il bus di espansione. La

sezione video non supporta l'uscita TV, che può essere comunque collegata con un adattatore esterno, ma soprattutto non supporta i colori.

La disponibilità del bus S-100 e la disponibilità delle altre uscite standard, hanno creato un interessante mercato di terze parti che hanno offerto software e hardware per espandere il sistema base. Alla porta parallela ad esempio può essere collegato un sistema audio che supera le limitazioni dello speaker interno e anche un programmatore di Eprom...

Uno dei punti di forza è il bus di espansione in standard S-100 che consente l'utilizzo di buona parte delle schede di espansione industriale che lo supportano. Questo particolare del progetto ha consentito senza dubbio la pronta uscita dell'unità floppy e relativo CP/M.

La Exidy alle sue origini è una società produttrice di videogiochi, non potevano quindi mancare i ROM-PACK con alcuni titoli. La vocazione professionale del progetto ha limitato la scelta di titoli ludici in favore delle applicazioni più serie. Sono state rese disponibili espansioni ROM con assembler, compilatori per linguaggi diversi come LISP,

FORTRAN e Pascal e programmi di uso personale come il già ricordato Word Processor e alcuni altri programmi applicativi come il gestore di dati CADAS che utilizza le cassette audio come storage.

L'espansione S-100, acquistabile a circa 420 dollari, si può definire ingombrante o quantomeno poco proporzionata alle dimensioni dell'unità centrale. Nell'unità di espansione trovano posto l'alimentatore e sei slot per altrettante schede di espansione.

Decisamente più azzeccata, anche se per forza dimensionata sull'ingombro del monitor, è l'espansione monitor + floppy che costa però parecchio: 3.000 dollari!

Dalla documentazione tecnica si deducono parecchie informazioni sul progetto. Il monitor di sistema, la ROM dei caratteri e la RAM video occupano la parte più alta dei 64K che il processore Z80 indirizza. Subito più in basso gli indirizzi sono riservati alle cartridge ROM che possono essere da 4, 8 o 16 Kbyte. Il monitor di sistema. Il resto della RAM, se implementata viene organizzato dalla ROM-PACK inserita nello slot.

L'hardware è descritto nei minimi dettagli



con indicazioni precise su come modificare la configurazione del sistema, il timing dei segnali, gli schemi logici, funzionali ed elettrici di ogni singola parte, etc...

In generale giudichiamo il progetto SORCERER una sorta di punto di partenza per tantissimi progetti basati sullo Z80 che lo hanno seguito.

Uso del sistema.

L'uso rivela la qualità della tastiera il cui tocco è confortevole e responsivo. Il SORCERER possiede un set di caratteri particolarmente ricco con maiuscole, minuscole e semigrafici. Alcuni tasti, digitati assieme a GRAPHIC quando il sistema è in modalità programmazione, sono associati agli statement BASIC per facilitare la scrittura del codice, anche se oggi è più un impiccio che un aiuto, bisogna pensare alla scarsa familiarità con una tastiera che era tipica delle persone negli anni '80. Per finire l'utente può associare propri simboli ai tasti, ottenibili poi con la combinazione di tasti SHIFT+GRAPHIC; in questo modo un singolo tasto può arrivare ad avere associate ben cinque funzionalità.

La matrice del carattere è di 8x8 pixel e consente l'uso di un set di lettere con discendenti.



Sul video l'output è chiaro anche nella modalità a 64 colonne per 30 righe che è l'unica disponibile al contrario di altri sistemi della stessa classe che possono scegliere modalità semigrafiche diverse. La scelta della Exidy è da un certo punto di vista rigida, ma con il senno di poi la possiamo definire "visionaria", dal momento che l'uso del monitor dedicato al posto di un meno costoso televisore diventerà un indispensabile standard cinque anni dopo, almeno per le macchine considerate "da ufficio".

Il BASIC in dotazione, su licenza Microsoft, fa il suo dovere permettendo uno sfruttamento facilitato delle caratteristiche hardware del sistema, in particolare della sezione grafica che consente l'indirizzo di 122.880 pixel sul video, organizzati in una matrice di 512x240.

Il fatto che il SORCERER possieda due interfacce cassetta e la porta parallela per una stampante a basso costo, lo rende particolarmente fruibile anche da un piccolo ufficio per compiti non troppo "storage intensive", come la gestione di un piccolo magazzino o, utilizzando la cartridge con il Word Processor, disporre di una flessibile macchina da scrivere. L'interfaccia a cassette consente il trasferimento a 1200 baud, una velocità top per l'epoca, ma certo non confrontabile con la flessibilità di un floppy disk. Un cabinet di espansione con la possibilità di gestire due drive floppy sarà rilasciata dopo poco tempo rendendo il sistema più completo e ponendolo nell'ottica di resistere in un mercato semi-professionale dell'elaborazione.

La Exidy ha fatto uscire una successiva versione della macchina con sigla DP1000-2 correggendo dei piccoli bug nel monitor di sistema e altri piccoli particolari ma il SORCERER già dalla prima issue può essere considerato un sistema affidabile e completo, come testimoniano alcune "storie" raccolte sul Web nei siti che lo presentano come com-

ponente della loro collezione.

La documentazione ha la doppia anima comune all'epoca: da una parte le nozioni base per chi inizia e non sa forse nemmeno cosa significhi programmare un computer e dall'altro lato l'esperto che abbisogna di informazioni approfondite sulla struttura hardware e software delle componenti base. La Exidy fornisce anche il listato del monitor per chi vuole leggerselo e scoprire magari gli entry-point che gli abbisognano per qualche suo sviluppo particolare.

Due fascicoli sono i principali veicoli per la conoscenza del sistema: il più tecnico "Sorcerer Technical Manual" e il più orientato all'utente finale "Sorcerer Software Manual". Nel primo tutte le informazioni tecniche desiderabili compresi schemi elettrici, modifiche per adattare tipi diversi di ROM/EPROM e per le specifiche di alimentazione e uscita TV nei vari paesi.

Il "Development Pac User Manual" è invece rivolto agli sviluppatori che intendano confezionare le loro soluzioni sia ludiche che da ufficio. Per questi soggetti è disponibile una cartridge con un monitor avanzato e varie utilities, tutte descritte nel testo a corredo.

Il BASIC è disponibile solo su cartridge ed è un interprete da 12 Kbyte con funzioni e prestazioni nello standard; del resto si tratta di un interprete Microsoft. La macchina di base dispone di 8 Kb di RAM dei quali il BASIC lascia circa 7,5 Kb a disposizione del programma utente. Maggiore flessibilità richiede l'acquisto dell'espansione da 16 Kb o addirittura da 32 Kb che è il massimo indirizzabile dal progetto. L'idea delle cassette ROM rende la macchina appetibile nell'utilizzo di programmi standard, come il Word Processor, che è subito pronto all'accensione e non richiede i lunghi caricamenti da cassetta. Una versione estesa del BASIC era stata promessa su ROM da 16 Kb ma sembra non sia mai stata com-

Figura 5: Organizzazione RAM statica

mercata, mentre sono state realizzate delle versioni estese da terze parti, commercializzate però solo in alcuni paesi.

Senza BASIC il software che fa funzionare il computer è il classico "monitor" che in 4 Kbyte di RAM fornisce la sequenza di comandi per la programmazione in linguaggio macchina usando i codici mnemonici.

Sono disponibili routine per salvare il programma su cassette, richiamarlo in memoria, eseguirlo, etc... La Exidy prevede anche una rudimentale sequenza di automazione chiamata "Monitor Batch System" attraverso la quale è possibile far eseguire in sequenza i programmi registrati su cassetta.

Documentazione e aspetti "social" sono importanti per il successo di un prodotto. Della documentazione ufficiale abbiamo già detto, come andiamo sull'altro fronte, meno ufficiale? Beh, i gruppi, i club e le fanzine ci sono state, sempre proporzionalmente alla diffusione del prodotto. Una interessante è PORT-FE, bollettino del gruppo canadese domiciliato a Toronto. Altre esperienze analoghe si annoverano anche in Europa, ad esempio in Olanda con il gruppo denominato eSAG (Exi-

dy SORCERER User Group).

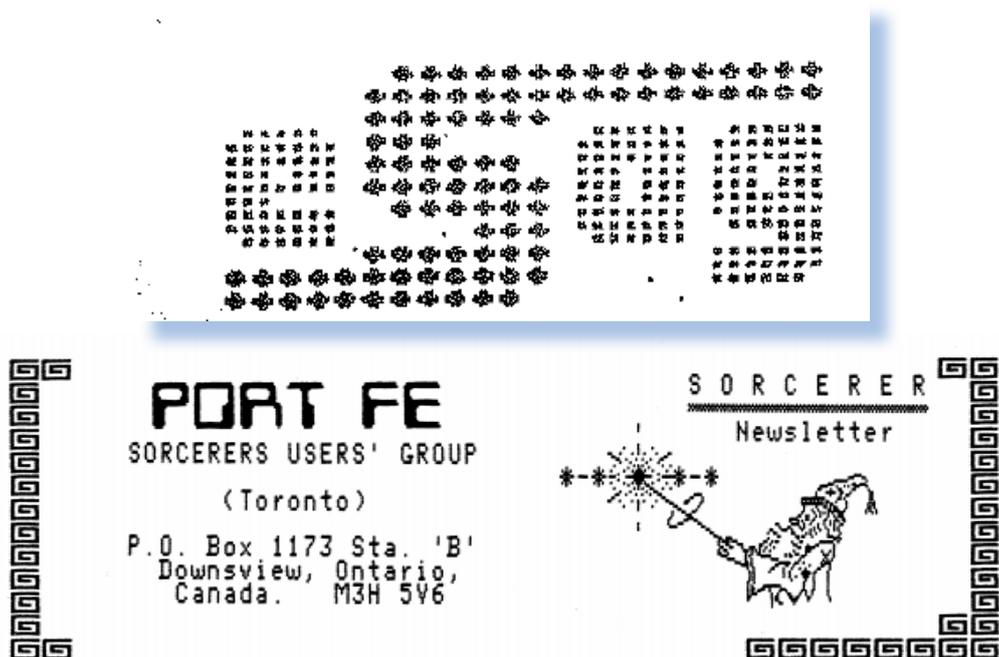
Queste iniziative spesso modificavano il firmware per estendere le funzionalità dei programmi come il monitor o il BASIC.

Conclusioni.

Abbiamo scoperto una macchina che non conoscevo per nulla! Uno dei tanti sistemi home che sgomitando con la concorrenza hanno cercato di crearsi uno spazio in quello che sarebbe diventato una miniera d'oro (per pochi).

In generale ne abbiamo ricavato l'impressione di un sistema discreto (oserei dire "ottimo", per l'epoca!), ben realizzato nel suo complesso e che forse, al pari di tanti altri, avrebbe meritato una vita più longeva. Credo di poter affermare che se nel 1978 mi fossi trovato davanti alla scelta se prendere un Apple II o un Sorcerer sarei stato in imbarazzo e non so proprio se avrei scelto la mela, compagno di anni e anni di studio e gioco!

(=)

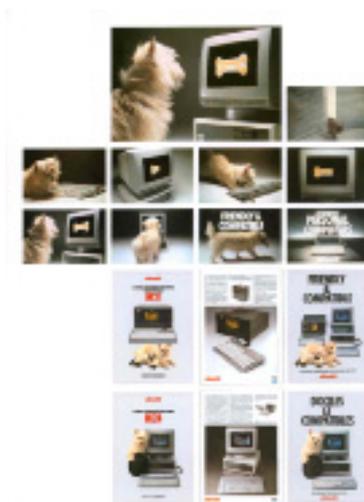




Bibliografia.

- *“Exidy Sorcerer” by Marcin Wichary from San Francisco, U.S.A.*
- *DigiBarn Systems: Exidy Sorcerer*
- *The Trailing Edge Resource Center - Exidy Sorcerer*
- *Exidy Sorcerer computer*
- *Exidy Sorcerer - Wikipedia, the free encyclopedia*
- *The Trailing Edge Related Items*
- *JSorcerer - Exidy Sorcerer Emulator*
- *www.computer-museum.nl*
- *Exidy Sorcerer (espansione bus 100)*
- *Exidy Sorcerer (as seen in Tezza's Vintage Computer Collection)*
- *Welcoming an Exidy Sorcerer*
- *Exidy Sorcerer BASIC ROM-PAC repair (Part 1) - Diagnosis*
- *The Boredom Destroyer:*
- *Exidy's Sorcerer - By Ken Barbier; su Creative Computing, 1978*

M24, cane e gatto...



di Jb72

Non si tratta solo del rientro nel mercato dell'elettronica mondiale, e in posizione di leadership di un'azienda storica dell'industria italiana; con ELEA Olivetti era già stata tra i pionieri nella messa a punto di computers per uso commerciale con le prime macchine al mondo costruite interamente a transistors (ELEA 9003, 1959).

Non si tratta solo dell'ennesimo prodotto di design secondo la più verace e convincente definizione di "made in Italy"; i nomi non sono quelli di giovani designers rampanti che piazzano a segno un colpo, ma piuttosto di interi capitoli della storia dell'arte e del design internazionale: Ettore Sottsass, Giovanni Pintori ed Egidio Bonfante, Carlo Scarpa, Ignazio Gardella, Figini e Pollini... addirittura Kenzo Tange, James Stirling e Louis Kahn... i prodotti, i negozi, gli stabilimenti ed il concetto di industria nella società moderna, per Olivetti (da ben prima che S.Jobs nascesse), comprendono anche la sostanza più sottile dell'opera d'arte.

Come altri big dell'industria elettronica nei primi anni Ottanta, dopo il "rientro" nel mercato internazionale dell'elettronica e dell'informatica alla metà degli anni Settanta e la fondazione a Cupertino dell'"Advanced Technology Center" (1979), anche Olivetti sente la necessità di affiancare

ai propri mainframe dei prodotti di piccole dimensioni. A differenza di come molti potrebbero pensare, non si tratta necessariamente di un tentativo maldestro di penetrare per arricchirsi velocemente nel mondo del personal computer, magari per contrastare piccoli produttori allora appena emergenti. L'idea del computer personale a diffusione di massa, il concetto di multimedialità e possibile uso domestico di un computer era allora solo da poco trascinata dall'universo nebuloso delle fantasie hackers; queste aziende piuttosto, pensavano a terminali intelligenti che dialogassero con i fratelli maggiori presenti in catalogo, macchine da laboratorio "general purpose", al limite, piccole reti distribuite con funzionalità di base d'ufficio: elaborazione testi, piccoli database, gestione di magazzino. Ciò che deve essere evidenziato è che computer come l'Olivetti M24 non possono essere direttamente rapportati a contemporanei prodotti Apple o Commodore, i quali appartengono a tutt'altre categorie e cosa che, nel momento in cui viene fatta, costituisce un banale termine di raffronto possibile solo in una visione retrospettiva.

HP, che aveva un catalogo zeppo di apparecchiature elettroniche per ogni esigenza comprendente ottime calcolatrici tascabili, disponeva di piccoli computers portatili;

negli stessi anni l'incursione nel settore delle macchine di fascia più bassa la avevano fatta concorrenti diretti come i produttori degli UNIVAC: Sperry Rand (poi UNISYS)... ma soprattutto IBM ci era riuscita con un considerevole successo determinato soprattutto dall'autorità indiscussa del suo nome! In realtà anche Olivetti lo aveva già fatto appena dopo IBM con un vero gioiellino di tecnologia, l'M20. Una piccola macchina multisistema ma anche compatibile CP/M, basata su un microprocessore sfortunato (Z8001) ma di ottima fattura, ideale per gli uffici e per i laboratori scientifici a cui veniva proposto nelle due vesti commerciali di strumento professionale. Ebbe anche un modesto riscontro sul fronte degli appassionati perché si trattava di un prodotto veramente di alta qualità seppure in nessuna delle intenzioni destinato ad essi. Non solo, ma proprio Olivetti, come e più degli stessi colossi americani, disponeva già per i suoi clienti di prodotti informatici per singoli utenti. Adirittura, siamo nel 1965 con la mitica P101 ("perottina") a Olivetti era già stato riconosciuto il merito in anticipo su tutti i concorrenti di aver definito all'interno degli ambienti aziendali e professionali, il concetto stesso di Personal Computer, ancor prima che questo venisse carpito dall'ideologia hippie prima, e del mercato consumistico poi. Anche verso la fine degli anni Settanta e prima dell'M20, Olivetti aveva introdotto negli uffici le primissime macchine da scrivere elettroniche (ET) e microcomputer, dei veri e propri personal

computer a tutti gli effetti (con software totalmente proprietario) come la serie P6060 del 1976. Insomma, "nulla di nuovo sotto il sole" e soprattutto bisognerebbe ben distinguere tra gli autodidatti studiosi di storia della scienza che vorrebbero l'uomo privo di qualsiasi tecnologia informatica prima dell'avvento nel mercato di tale Steve Jobs!

Nel 1984, per Olivetti, è la volta di M24: quasi la naturale evoluzione del precedente tentativo ma che a differenza di questo rappresenta un successo commerciale pienamente giustificato. Questo Personal Computer si ispira al PC-IBM di cui riprende l'architettura ISA e di cui risulta avere un BIOS pienamente compatibile in grado di far funzionare il computer con il sistema operativo DOS di Microsoft. E' però da sottolineare la sostanziale differenza che esiste tra un "compatibile MS-DOS" come è l'M24 (e come ce n'erano altri al tempo) ed i cosiddetti "cloni" del PC-IBM (cinesi e non), brut-





te copie a tutti gli effetti del personal computer di IBM e che proprio allora iniziavano a fare la loro comparsa sul mercato. Rispetto al personal IBM, che allora compariva nella veste definitiva di PC-XT, la soluzione Olivetti lo supera praticamente su tutta la linea.

Dal punto di vista costruttivo nulla da eccepire per entrambi: la qualità costruttiva delle macchine del tempo non è nemmeno lontanamente equiparabile con qualsiasi computer prodotto attualmente, nemmeno se confrontata con workstation o con i marchi più blasonati. Un PC della metà degli anni Ottanta era costruito senza tener presente l'obsolescenza in cui sarebbe incorso entro meno di un lustro, ma sembrava costruito per dover durare in eterno. L'unità centrale è pesante e quasi al limite della trasportabilità manuale, la piastra madre è contenuta in un sottofondo metallico arieggiato indipendentemente e dal quale emergono solo gli slot per installazione di schede di espansione (il computer era proget-

tato per usi industriali e la varietà di schede esistente era notevole), il trasformatore è poderoso, la ventola posteriore ha le dimensioni e la consistenza di un buon ventilatore. Forse il prodotto Olivetti poteva sembrare un po' più "plastico" rispetto a quello IBM, ma dal punto di vista dell'estetica quello americano sembrava una "macchina dell'Ottocento" (!) mentre l'italiano, con il suo schermo orientabile, la tastiera elegantissima, il case essenziale, sembrava un oggetto proveniente dal futuro: "il computer del 2000"!

Anche tecnologicamente l'M24 era superiore all'IBM-XT: processore 8086 full 16bit, maggiore dotazione di RAM, migliori prestazioni in generale, video con fosfori ad alta persistenza di qualità formidabile (la grafica sembrava aderente allo strato superiore del monitor) e una scheda grafica migliorata che introduceva la risoluzione di 640x400 punti in 16 colori su palette di 64 (superiore ad Hercules e ad EGA quando le potenzialità grafiche si misuravano in numero di pixels



per numero di colori!). Grafica, tra l'altro, pienamente usufruibile perché Olivetti, essendo un produttore di primissimo piano, poteva fornire driver per i software più importanti come pure

metteva software di grande diffusione (come l'elegante pacchetto integrato a finestre Frameworks di Aston-Tate) oppure pacchetti completi di una vastissima biblioteca realizzata dal settore software dell'azienda e distribuita con una rete capillare di negozi e concessionari altamente qualificati.

In quel periodo Olivetti stava raggiungendo l'acme del suo successo. Prima in Europa, leader nel mercato mainframe, primo produttore di hard-disk con Olivetti Peripheral Equipment (OPE) e stampanti a getto d'inchiostro, acquisizioni eccellenti come Triumph-Adler in Germania e Acorn in Gran Bretagna. Negli Stati Uniti grazie alla partnership con AT&T produce insieme a questa una versione U.S. di M24 denominata PC6300. Il grande successo di questa macchina (e altre precedenti come l'ultraportatile M10, oppure contemporanee come il bellissimo portatile M15; un po' meno magari l'originale trasportabile M21 ma abbastanza il successivo personal compatibile di seconda generazione e più piccolo M19) avvia una serie di aggiornamenti e restyling come gli M240, M280 (PC-AT) e M380 (server con Intel 80386). Poi macchine ancora più potenti: server e workstation intermedie tra il personal computer



ed il mainframe; quando qualche anno dopo IBM lancia la linea PS/2, Olivetti rinnova la fiducia lanciando in anteprima la produzione dei primi compatibili Microchannel (MCA) come il P500. Ma ormai è tardi e la terra sta cedendo sotto ai piedi di questo gigante: quando ormai la formidabile avventura industriale è stata completamente trasformata in "polpettone finanziario"... ecco il tracollo!

Al tempo il mio primo PC - non proprio un Olivetti M24 originale ma una versione "esotica" e del tutto analoga tranne per il colore "beige" anziché grigio, e marchiata da una nota acquisizione Svizzera di Olivetti: la Hermes - era dotato di due drive, un winchester da 10 Megabyte e un overclocking "sperimentale" del chip 8086 a quasi 8 Mhz (che in realtà a volte si dimostrava un po' problematico). Ricordo quando mi capitava di descriverlo a qualcuno di mestiere, la puntuale constatazione laconica di questi era inevitabile: "beh... praticamente il migliore!"

(=)

Replicare ENIGMA con il chip ATMega 238P



di [double_wrap](#)

*T*empo fa ho visto al cinema il film *The Imitation Game* che racconta la storia della forzatura di Enigma, la macchina cifrante tedesca. E' un gran bel film e mi ha fatto nascere l'idea di realizzare una replica della macchina. In rete c'è molta documentazione ed anche ottimi simulatori per PC (alcuni link sono riportati in calce all'articolo).

Io volevo un oggetto portatile ed ho trovato una realizzazione che usa un **Atmega238P** con uno schermo touch per la configurazione dei rotori [<http://arduinoenigma.blogspot.it>].

Oltre alla vendita sono anche disponibili liberamente i codici sorgente in pieno spirito open source [<https://drive.google.com/folderview?id=oBonObKKDtir6Wlk2dzh2NC1vTjg&usp=sharing#list>]; in particolare è possibile realizzare una versione senza schermo

touch impiegando un normale terminale seriale come hyperterminal o putty [<http://arduinoenigma.blogspot.it/2014/10/source-code-for-implementation-of.html>].

La mia Enigma è come il Chianti, un assemblaggio di HW e SW costruiti intorno al core di codifica open source: ho aggiunto una tastiera ed un display per digitare e visualizzare il testo co-decodificato ed una scheda SD per gestire la configurazione interna dei rotori. Lo schema a blocchi, visibile nel disegno della pagina a fronte, è composto dai seguenti elementi:

- il telecomando ir invia ad un ricevitore ir TSOP a 36 KHz il testo ed i comandi di impostazione;
- la scheda SD ospita un file con la configurazione dei rotori e degli steckers;

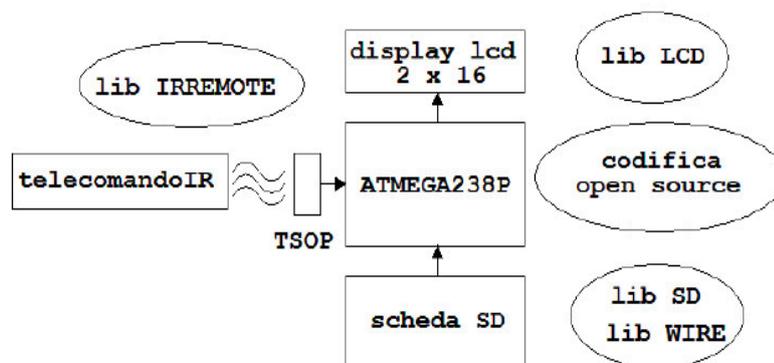


Fig. 1 -

Schema a blocchi della realizzazione.

- il display lcd visualizza il testo e la posizione dei rotori che avanzano ad ogni battuta di tasto;

- l'Atmega238P usa il core di codifica open source e le librerie per gestire i componenti.

Per realizzare la mia Enigma replica e per configurarla in modo da decodificare messaggi reali ho dovuto studiare come è fatta dentro la vera Enigma: vale la pena di spenderci qualche parola...

Come è fatta Enigma

Enigma è una macchina codificatrice meccanica, può essere usata sia per codificare che per decodificare un messaggio essendo una macchina simmetrica. Tale caratteristica ne semplifica moltissimo l'uso pur mantenendo una elevata capacità crittografica.

Fu inventata nel 1918 dal tedesco Arthur Scherbius col supporto dell'olandese Hugo Alexander Koch, una accoppiata perfetta fra una nazione di tecnici ed una di mercanti: era destinata all'ambito finanziario ed ottenne parecchi bre-

Fig. 3 -

Enigma versione M4



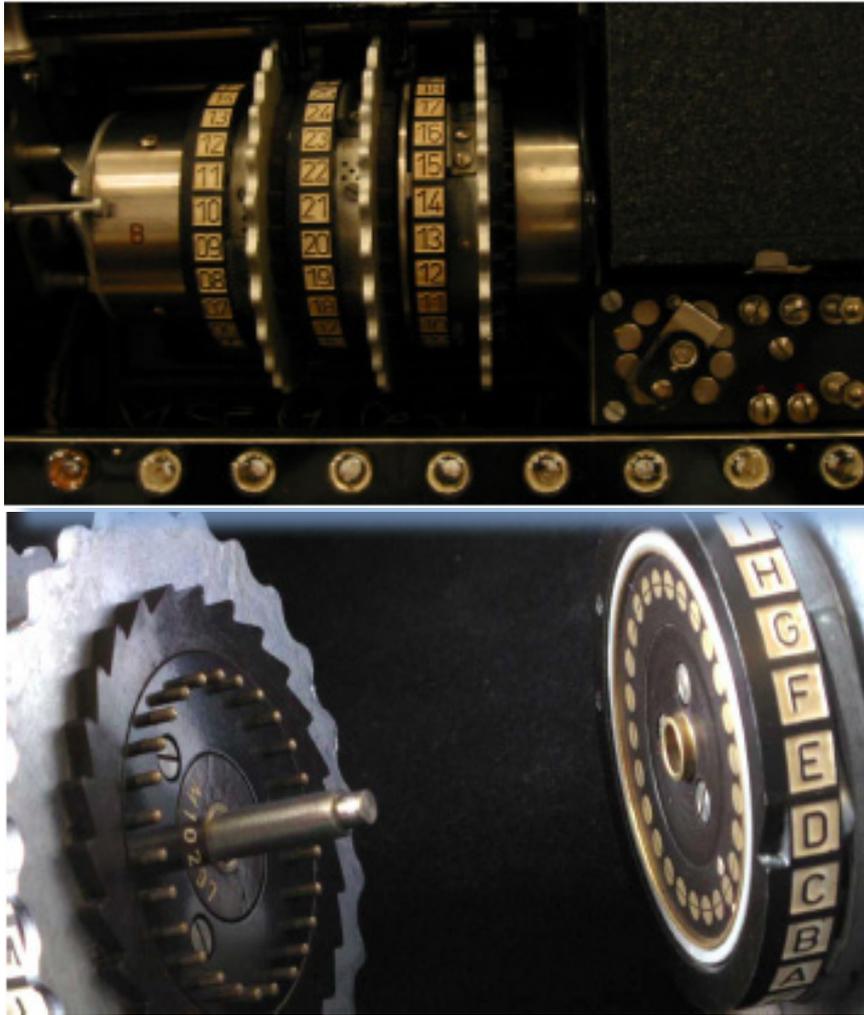
Fig. 2 -

Enigma versione I



Fig. 4 -

I rotori di Enigma I e sotto quelli della versione M3



Enigma usa una tastiera a 26 caratteri, solo lettere senza numeri: quando si preme un tasto si accende la lampadina corrispondente al carattere codificato. La macchina opera una crittografia a sostituzione di carattere usando una serie di rotori (walzen) ciascuno dei quali scambia uno dei 26 caratteri.

Ogni rotore ha quindi 26 entrate e 26 uscite connesse in modo fisso e diverso da rotore a rotore. Sono anche presenti 10 plug (stecker) ad inserimento libero: ciascuno inverte due caratteri della tastiera prima che

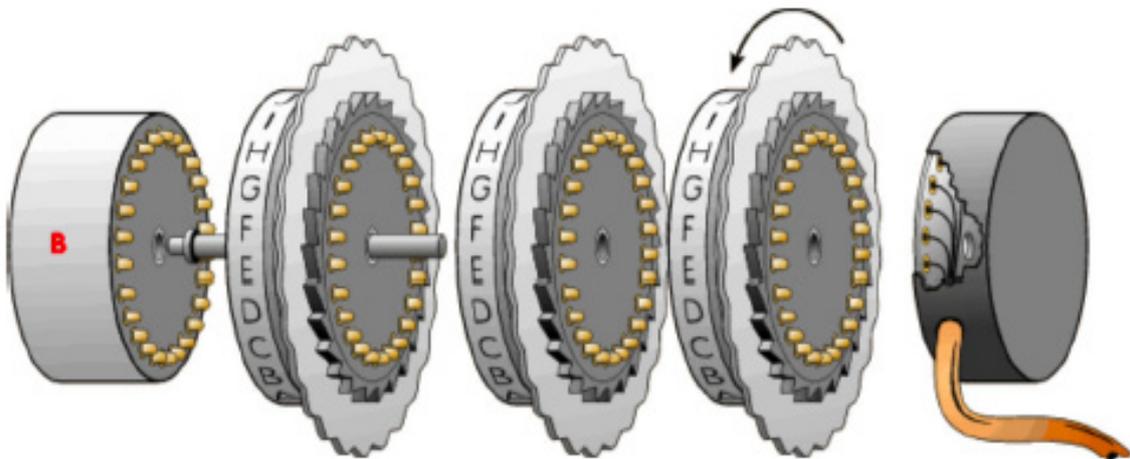
vetti finchè con la seconda guerra mondiale non fu acquisita dall'apparato militare tedesco.

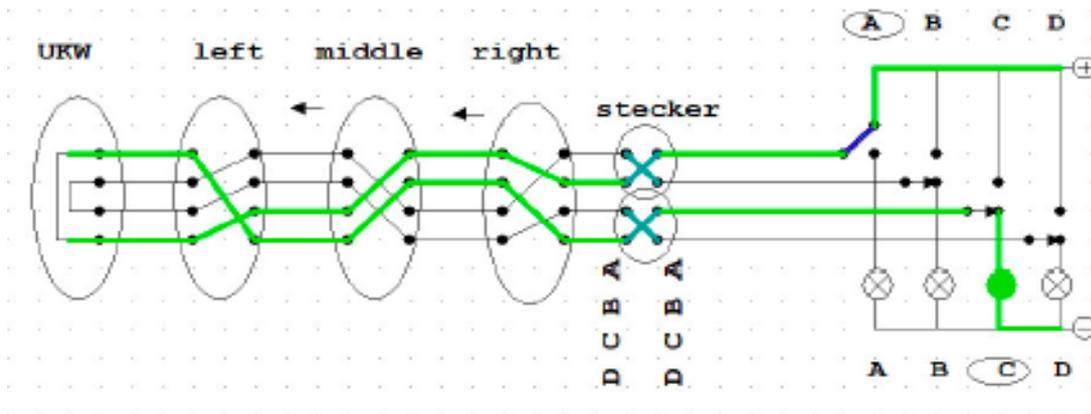
Fu costruita in molte versioni conservando però le stesse funzionalità di crittazione e la caratteristica di portabilità che la rendeva utilissima nelle campagne militari.

siano inviati ai rotori. Nello schema semplificato (nella pagina a fronte) sono rappresentati solo 4 tasti e le connessioni dei rotori sono ridotte a 4. Il colore verde mostra il percorso della corrente dal tasto premuto alla lampadina illuminata attraverso i plug

Fig. 5 -

Come sono assemblati meccanicamente i rotori.





ed i rotori.

Il tasto A della tastiera commuta il deviatore associato: la corrente passa prima attraverso un plug che inverte la A con la B, poi attraverso lo statore (Entrittswalze o ETW) entra nel rotore di destra che inverte la B con la A, passa nel rotore centrale che inverte la A con la C e nel rotore di sinistra che inverte la C con la D, infine arriva al riflettore (umkehrwalze o UKW) che inverte la D con la A. A questo punto la corrente riattraversa i tre rotori ed un ulteriore plug e finalmente accende la lampadina C. La sequenza degli scambi di carattere operati è la seguente:

$A > B > A > C > D > A > D > B > D > C$
e quindi la lettera A è codificata in C.

Nella macchina reale con i 26 caratteri, i rotori completi delle 26 connessioni ed i 10 plug la codifica è assai più articolata.

Il bello viene adesso: ogni volta che si pigia un tasto il rotore destro avanza di una posizione modificando la sequenza di scambio e, quando viene completato un giro, fa avanzare il rotore centrale, lo stesso vale per gli altri rotori. La sequenza di codifica

è quindi diversa ad ogni tasto premuto ma non basta. Ogni rotore può essere impostato su 26 posizioni diverse (ringstellung) per i collegamenti interni ed è anche possibile inserire nella plugboard (steckerbrett) 10 plug (stecker) che invertono a coppie 20 delle 26 lettere. Non è mica finita. Nella macchina Enigma usata dall'esercito e dall'aviazione è possibile scegliere fra 2 riflettori e 5 rotori posizionandoli in vario ordine (walzenlage) ma la marina prima aggiunse tre nuovi rotori a doppio avanzamento portando ad otto la scelta e poi portò a quattro i rotori codificatori aggiungendo i rotori beta e gamma per un totale di 10 rotori disponibili.

Finita la configurazione interna (variabile su base giornaliera) si passa alla configurazione esterna (variabile per ogni singolo messaggio) posizionando ciascun rotore in modo da impostarne la posizione iniziale (grundstellung) fra le 26 possibili.

La Enigma I dell'esercito e la Enigma M3 della marina differiscono essenzialmente per il fatto che la prima dispone di 5 rotori ed usa numeri da 01 a 26 mentre la seconda



Fig. 6 -
i particolari Steckerbrett (a sinistra) e grundstellung (a destra).

dispone di 8 rotori ed usa le lettere da A a Z. Un aspetto interessante è la retrocompatibilità fra le macchine Enigma M4 a 4 rotori con le Enigma I ed M3 a tre rotori: occorre solo usare una opportuna combinazione di riflettore e di quarto rotore.

Se non ci fosse stato il genio di Turing a Bletchley Park nessuno avrebbe mai decifrato il codice Enigma. Inizialmente i polacchi decifrarono le prime Enigma ma commisero l'errore imperdonabile di farlo sapere ed i tedeschi portarono a cinque il numero dei rotori impedendo ulteriori decrittazioni per parecchi anni. Non a caso gli inglesi annunciarono di avere forzato Enigma solo nel 1970, ben 25 anni dopo la fine della guerra, per intercettare eventuali messaggi codificati con le Enigma ancora in giro ed usate in ambito finanziario.

Enigma è una macchina strutturalmente "semplice" oltre che meccanicamente robusta come tutto quanto prodotto in Germania: la sua forza sta nella cifratura variabile giornaliera integrata dalla chiave di messaggio di tre lettere (quattro per la M4 della marina) che veniva decisa dall'operatore "a sentimento" messaggio per messaggio. Si è calcolato che la codifica di una Enigma I sia equivalente ad una chiave di 77 bit ($\sim 1.1 \times 10^{23}$ combinazioni) mentre una M4 ha una chiave equivalente di 84 bit ($\sim 3.1 \times 10^{25}$ combinazioni). Sembra assurdo ma la chiave del singolo messaggio veniva codificata ed inviata "assieme al messaggio" tanto era forte il sistema di cifratura. Per la cronaca anche PGP invia "assieme al messaggio" codificato in 3DES la relativa chiave di cifratura codificata in RSA,

speriamo che non si sia già ripetuta la storia di Enigma.

La mia Enigma.

L'HW è minimo (schema nella pagina a fronte): un Atmega238P-PU (con bootloader precaricato) col quarzo a 16 MHz e due condensatori dialoga con un display lcd, con una scheda SD e con un TSOP36 che riceve i codici generati da un normale telecomando ad infrarossi. Il tutto è alimentato a 5 volt e l'assorbimento è di circa 30 mA. La scheda SD lavora a 3,3 V e per collegarla all'Atmega ho usato una vecchia board di NE che avevo nel cassetto: c'è uno slot per SD ed un buffer 74HC244 (8 buffer) usato come adattatore di livello che si può sostituire con un CD4050 (6 buffer); i 3,3 Volt sono forniti da un regolatore ldo LM3940. L'uso dei buffer è sempre preferibile al partitore con resistenze, oltretutto nel mio caso Atmega e SD sono collegati con cavetti volanti di una decina di centimetri.

Sono collegati i pin da 1 a 7 della scheda SD (o di un adattatore per microSD) che usa il protocollo SPI sui pin 11,12, e 13 dell'Atmega, come chip select ho usato il pin 10. Se si usa un altro pin è fondamentale impostare il pin 10 come output con `pinMode(10, OUTPUT)` se il pin 10 fosse configurato come input ed andasse a livello basso l'Atmega entrerebbe in modalità SPI slave: occorre sempre ricordarsene.

Come sempre accade il realizzare l'HW ed il SW è stata la parte più semplice del lavoro, quella laboriosa è stata trovare un metodo per impostare la configurazione interna che fosse semplice e che richiamasse l'Enigma originale: la scelta è caduta su una SD che ospita un file con la configurazione.

Fig. 7 -

Piedinatura di una SD Card

Pin	Descrizione	Modalità SPI	Direzione
1	CD/DAT3	CS	Ingresso
2	CMD	SDI	Ingresso
3	GND		
4	VDD 3,3V		
5	CLK	SCK	Ingresso
6	GND		
7	DAT0	SDO	Uscita
8	DAT1		
9	DAT2		

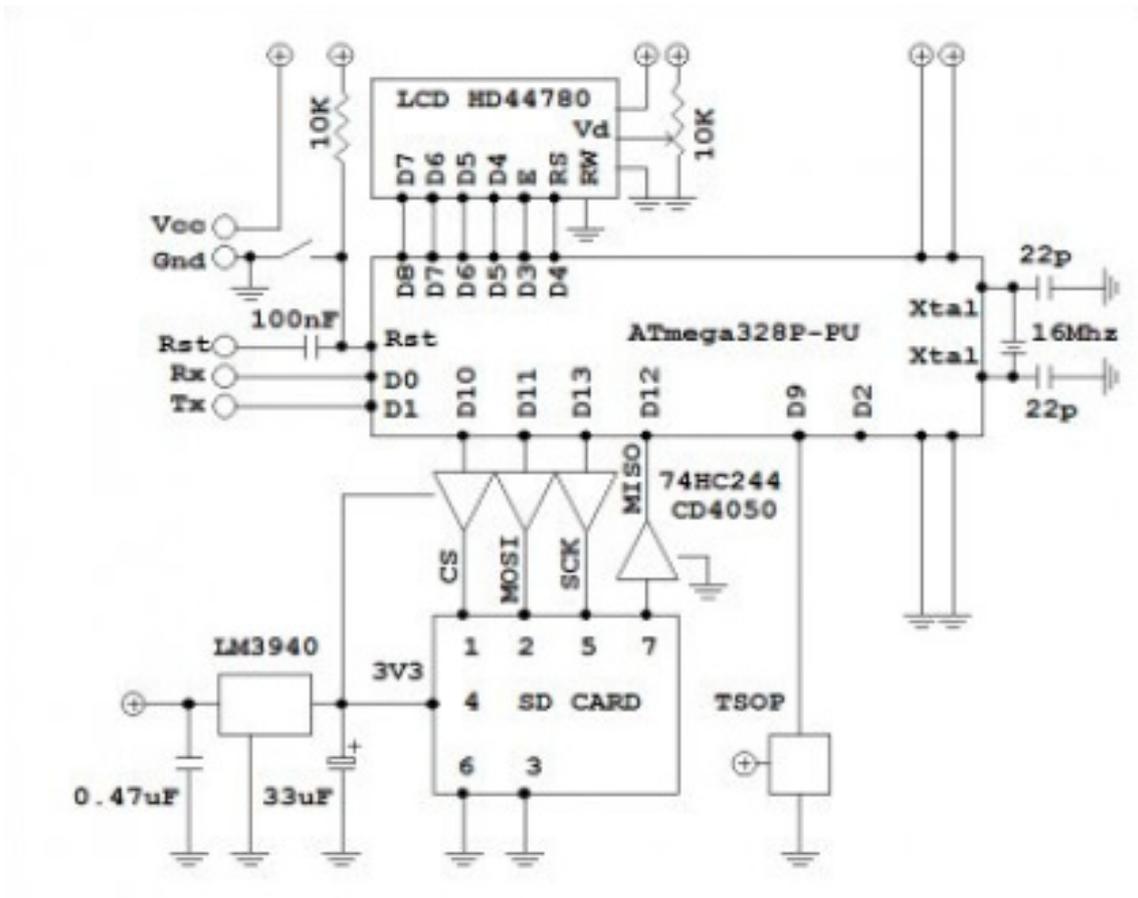


Fig. 8 -

Schema elettrico della realizzazione.

Il programma di emulazione.

[Il sorgente del programma è disponibile ai link riportati in fondo all'articolo; Il codice va compilato con la IDE 1.0X di Arduino e caricato nell'atmega238P con un convertitore USB/TTL]

Nei commenti ho evidenziato chiaramente la parte open source utilizzata, quindi sia dato a Cesare il molto già fatto ed a me quello che rimaneva da fare.

Le istruzioni nel loop() decodificano il codice RCS ricevuto del telecomando con la libreria IrRemote [disponibile al link: "https://github.com/shirriff/Arduino-IRremote"] e quindi:

se è stato ricevuto un carattere si muovono i rotori e si visualizza la loro nuova posizione si co-decodifica il carattere e lo si visualizza
 se è stato ricevuto un comando si imposta la posizione iniziale dei rotori (la chiave di cifratura del messaggio) si eseguono altre fun-

zioni di supporto spiegate più avanti.

Partendo dal codice open source Enigma-Serial ho integrato il setup() con l'inizializzazione del display lcd, della scheda SD e del ricevitore IR, ho aggiunto la funzione leggi_sd() che legge la configurazione dalla scheda SD (sostituisce la funzione initEnigma() originale) e nel loop() ho richiamato le due funzioni "core" che fanno funzionare Enigma:

MoveWheels(); fa avanzare i rotori con le giuste (ed articolate) regole;

tasto= EncodeKey(tasto); co-decodifica il carattere ricevuto.

La ricezione è in background con un interrupt, questo è l'uso di base

```

if (irrecv.decode(&results)) // se è stato ricevuto un codice
{
    code_value = results.value; // leggi il codice ricevuto
    // fai qualcosa
    irrecv.resume(); // riavvia la
    
```

Fig. 10 -

Il telecomando BRAVO, prima e dopo l'adattamento.



ricezione

}

La libreria decodifica una decina di protocolli: il telecomando BRAVO B308 (trovato usato a 5 Euro) ha la bellezza di 50 tasti allineati ed usa la codifica Philips RC5. Tale protocollo genera un codice di 14 bit, di cui solo 11 utili, 5 per l'indirizzo (0x00 per TV) e 6 per i comandi. Ad ogni pressione viene invertito il bit 12 (=2048): premendo ripetutamente lo stesso tasto si ha 2-2050-2-2050 mentre se si tiene premuto il tasto si ha 2-2-2 o 2050-2050-2050. Occorre crearsi una mappa tasti premuti - codici inviati - caratteri assegnati in funzione del telecomando usato, tale mappa va inserita nel sorgente prima di compilarlo. Usando telecomandi con codifiche diverse dalla RCS occorre fare qualche adattamento.

Montaggio e uso.

Fig. 11 -

Il montaggio delle schede elettroniche.

In una scatola di legno verniciata ho inserito la board con l'Atmega e quella con la SD, il display e l'interruttore di accensione.

Aprendo uno sportello si accede allo slot SD per inserire la scheda e configurare la macchina, proprio come nelle vecchie Enigma.

Per l'utilizzo consiglio vivamente di impraticarsi sul simulatore per PC <http://users.telenet.be/d.rijmenants/en/enigmasim.htm> e soprattutto di leggere l'ottimo manuale <http://users.telenet.be/d.rijmenants/Enigma%20Sim%20Manual.pdf> che fornisce moltissime informazioni tecniche, storiche e procedurali per prendere confidenza sulla impostazione dei rotori, dei plug e della chiave di cifratura.

Riporto un glossario dei termini tedeschi usati:

Eintrittswalze (ETW) = statore di ingresso;

Umkehrwalze (UKW) = riflettore

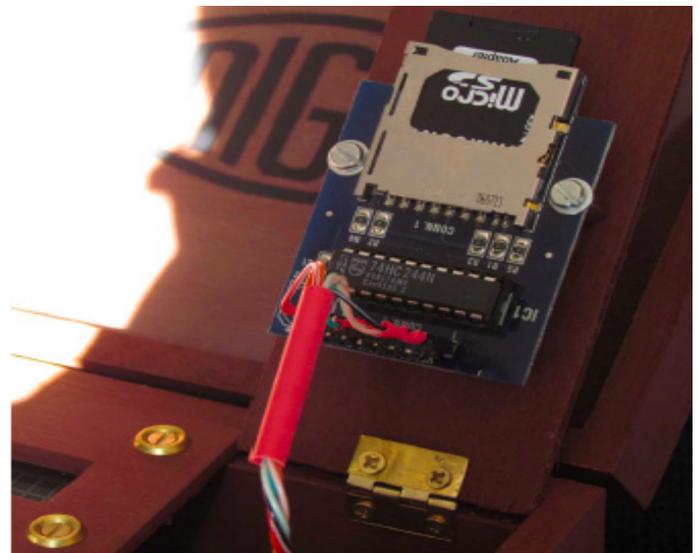
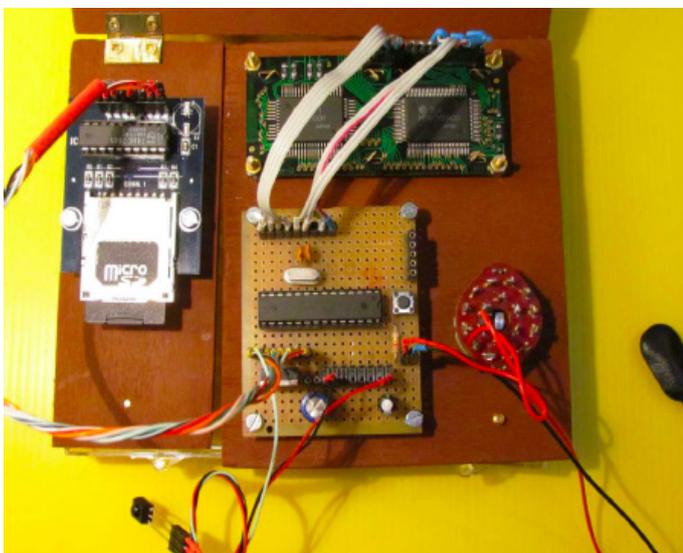
Walzen = rotore di codifica;

Walzenlage = selezione dei rotori;

Ringstellung = impostazione interna dei rotori;

Grundstellung = impostazione esterna dei rotori;

Steckerbrett = pannello dei plug;





Per decodificare un testo cifrato occorre avere le stesse impostazioni della macchina cifrante, come ho detto Enigma è una macchina simmetrica. Per prima cosa occorre impostare la configurazione interna giornaliera

- 1) Umkehrwalze: riflettore B o C per Enigma I e M3, B_{thin} o C_{thin} per Enigma M4;
- 2) Walzenlage: 3 rotori scelti fra i 5 per I o fra gli 8 per M3 e M4, quarto rotore "beta" o "gamma" per M4;
- 3) Ringstellung: lettera del ring per ciascuno dei tre o quattro rotori scelti;
- 4) Steckerbrett: di norma 10 plug, se ne possono usare meno ma non di più;

Nella mia Enigma si scrivono i 44 caratteri che descrivono la configurazione interna nel file `enigma.txt`. Si copia il file sulla SD card, si apre lo sportello, si inserisce la scheda nello slot e si richiude lo sportello e si accende Enigma. Per semplificare la configurazione ho sviluppato un programma in RapidQ che genera il file `enigma.txt`.

Ora che abbiamo impostato la configurazione interna giornaliera possiamo introdurre

la chiave del messaggio. All'accensione appare sul display la chiave AAA (o AAAA per la M4 (il riconoscimento è automatico in base ai rotori scelti) e col telecomando si preme il tasto UP (A) o DOWN (v) per cambiare la lettera e il tasto SET(>) per passare alla lettera successiva. Quando il display non lampeggia più la chiave è completa.

Qui è doveroso un riconoscimento a Gvsoft: ho preso spunto dal suo ottimo articolo ARDU_CLOCK [<http://www.grix.it/viewer.php?page=11974>] per impostare la chiave sul display.

Come esempio inseriamo la chiave BLA e battiamo il nostro messaggio: sul display apparirà il testo crittografato a gruppi di 4 lettere (convenzione della marina), occorre copiarlo su carta perché i caratteri scorrono sulla riga inferiore del display. Vedremo anche che ad ogni tasto le lettere dei rotori visualizzate sulla riga superiore cambiano: i rotori ruotano ad ogni tasto premuto.

Per decodificare il testo che abbiamo appena crittografato occorre reintrodurre la chiave di codifica BLA come sopra, digitiamo il testo crittografato e vedremo riapparire il testo in chiaro.

Ci sono alcune funzioni per agevolare l'uso:

Fig. 12 -

La realizzazione pratica del contenitore.

Fig. 13 -

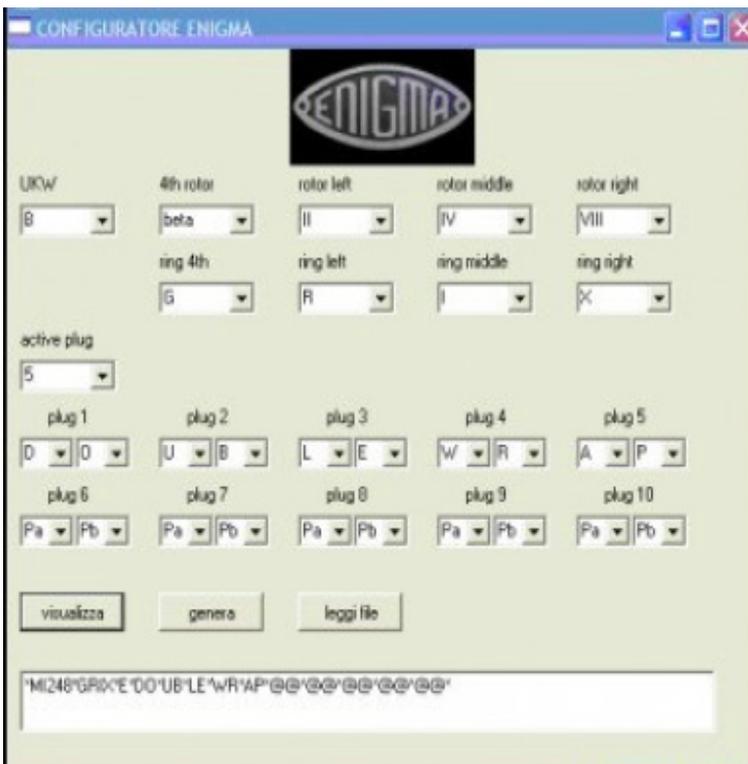
Il display mostra la fase di codifica (in alto) e la configurazione interna (in basso).



- save*: Salva la chiave di messaggio in memoria;
- load*: Carica la chiave di messaggio dalla memoria;
- read*: Ricarica la configurazione dal file sulla SD;
- displ*: Visualizza la configurazione interna letta dalla SD;

Fig. 14 -

Il programma di supporto per generare il file di testo della configurazione.



Conclusione.

Si potrebbe dire molto di più ma è meglio non eccedere. Ci sono bellissimi dettagli tecnici come il doppio salto del rotore centrale e procedurali come la scrittura dei numeri o la doppia codifica di chiave e messaggio o ancora le rotazioni manuali dei rotori durante il messaggio stesso: chi fosse interessato in rete troverà moltissima documentazione.

Realizzare la replica di Enigma è stato un pò rivivere un pezzo della nostra storia e della nostra tecnica anche se troppo spesso l'ingegno è stato finalizzato alla distruzione piuttosto che al miglioramento dell'umanità.

(=)

Reference.

- Sito dell'autore: <http://adamatj.altervista.org>
- Blechley Park - <http://www.bletchleypark.org.uk/>;
- Enigma touch screen - <http://arduinoenigma.blogspot.it>
- Codice open per Enigma su seriale - <http://arduinoenigma.blogspot.it/2014/10/source-code-for-implementation-of.html>;
- Codice per aggiornamento da ir su LCD scritto da Gvsoft su grix - <http://www.grix.it/viewer.php?page=11974>;
- Enigma - <http://www.cryptomuseum.com/crypto/enigma/index.htm>;
- Enigma - <http://users.telenet.be/d.rijmenants/index.htm>;
- Simulatore Enigma 7.0.6 - <http://users.telenet.be/d.rijmenants/en/enigmasim.htm>;
- Manuale del simulatore Enigma 7.0.6 - <http://users.telenet.be/d.rijmenants/Enigma%20Sim%20Manual.pdf>;
- Simulatore enigma on line - <http://enigma.louisedade.co.uk/index.html>;
- Libreria IrRemote -<http://www.sbprojects.com/knowledge/ir/rc5.php> -
<https://github.com/z3to/Arduino-IRremote>
- Protocollo RCS - <http://www.sbprojects.com/knowledge/ir/rcS.php>
- RapidQ - <http://rapidq.phatcode.net/>;

Trenta candeline per il primo portatile



di Damiano Cavicchio

Il Corriere della sera in un recentissimo articolo ricordava l'uscita del primo portatile di massa, anche se in effetti non si trattava per l'epoca di una novità assoluta, in quanto esistevano già sul mercato altri modelli di Pc portatili o come si diceva "trasportabili", realizzati da Osborne, Radio Shack, Kaypro, ecc... Ma il **Toshiba T1100** per il suo carico di novità tecnologiche e per il suo design, lo si può ritenere un capostipite per i successivi portatili usciti sul mercato mondiale.

Quindi esattamente 30 anni fa, nell'aprile 1985, abbiamo assistito ad una primavera tecnologica che ci ha portato nel mondo della mobilità e con il lancio del notebook Toshiba T1100 sul mercato europeo l'azienda giapponese ha segnato una svolta nella tecnologia dei PC mobile.

Il Toshiba T1100 è il primo notebook al mondo pensato per il mercato di massa, in-

tegrava al suo interno un processore Intel 80C88 da 4.77 MHz, il sistema operativo MS-DOS 2.11, un drive per floppy disk da 3,5 pollici in grado di archiviare 720 kilobyte – per la prima volta integrato in un PC di massa – e 512 kilobyte di RAM. Il notebook misurava 31 cm di larghezza, 6,6 cm di altezza e 30 cm di profondità. Con un peso di 4 kg, rappresentava un trionfo di miniaturizzazione tecnologica per il suo tempo.

Era dotato di un nitido e leggibile display in bianco e nero (con schermo 9,1"x4,7" e una risoluzione di 640 x 200 pixel), connessione via modem opzionale a 14,4 kbps.

Questo notebook nel 1985 costava più di 4000 dollari.

Con il sistema operativo Microsoft MS-DOS, il T1100 era compatibile con il software per PC IBM, aspetto fondamentale per essere utilizzato in ambiente business.

La nascita.

Il responsabile del progetto era Atsutoshi Nishida, che riuscì a convincere i vertici dell'azienda giapponese della bontà del prodotto e con la promessa di venderne almeno 10.000 esemplari. Inizia così a lavorare sul T1100 nell'aprile 1984 e il primo prototipo è pronto in agosto. Nonostante il T1100 fosse IBM-compatibile il problema è che nessuno avrebbe comprato un computer senza software e il nuovo drive da 3,5" che il sistema montava come unità di massa era troppo avanzato e non esistevano ancora programmi per quel formato, ma solo su dischi da 5,25".

Nishida riuscì a convincere Lotus a far migrare Lotus 1-2-3 al formato da 3,5 pollici e, in seguito, anche Ashton-Tate Corp (poi acquisita da Borland) a far migrare dBase II su dischi più piccoli, successivamente ancora Lotus ha fatto migrare una versione della sua suite Symphony al nuovo formato. Infine, il team Toshiba è riuscito ad ottenere una versione di Microsoft Flight Simulator che girasse sui floppy da 3,5 pollici.

Il Toshiba T1100 è stato lanciato in Europa alla fiera di Hannover nel mese di aprile 1985 e messo in vendita subito dopo in Germania e poi negli Stati Uniti, agli inizi del 1986, ne furono venduti più di 10.000 unità.

... in seguito.

Nel 1987 Toshiba propone una versione aggiornata del Toshiba T1100 con processore Intel 80C86 a 9,54 MHz, 1 MB di RAM di cui 384KB utilizzabile come RAMdisk, scheda grafica CGA, unità floppy con 720KB 3,5" e un hard disk da 20 MB (alcuni modelli hanno avuto due unità floppy) e sistema operativo MS-DOS 3.30, incluso con il computer portatile. È stato il primo computer portatile con una batteria sostituibile. Il suo prezzo originale era di 6.499 dollari.

Il mercato.

Molto intelligentemente la Toshiba presenta il suo prodotto con la seguente pubblicità:

Testo originale: *L'intelligenza portatile, Toshiba T1100.*

T1100 Portatile "Made in Japan": leggero anche nel prezzo, ovunque con voi, quotidianamente.

Da sempre, il sogno di chi aspira a lavora-

**L'intelligenza portatile
Toshiba
T1100**

T1100 Portatile "Made in Japan": leggero anche nel prezzo, ovunque con voi, quotidianamente.

Da sempre, il sogno di chi aspira a lavorare con un Personal Computer, è di possederne uno che sia realmente all'avanguardia, totalmente compatibile ed in grado di comunicare con lo standard più diffuso, leggero e maneggevole, silenzioso e di piccole dimensioni.

La Toshiba, sempre attenta alle esigenze del mercato, ha compreso tutto questo per prima ed è così l'unica casa al mondo che rivoluzionando il mercato del computer presenta tre portatili in altrettante classi diverse.

Il Toshiba T1100, per esempio, anche grazie alla applicabilità del virtual disk, è il più potente di altri personal esclusivamente da tavolo di prezzo notevole anche rispetto.

La capacità di memoria, il nuovo video LCD ad altissima leggibilità ed estremamente spazioso, l'elevata risoluzione grafica, sono solo alcune delle caratteristiche che contraddistinguono questo piccolo "supercorpo" dalle grandi prestazioni, rendendolo inimitabile nel lavoro quotidiano.

Caratteristiche tecniche

Processore	80C86
Alimentazione	3 batterie ricaricabili (3 ore lavorative) e a 220 volt
Video	LCD ad alta risoluzione (640 x 200 pixels)
Sistema operativo	MS-DOS 3.11
Memoria	controllata da 256 KB, espandibile a 512 KB
Interfaccia	standard ECP8 - Video compatto
Dischi	due drive floppy da 720 KB
Porte	Seriale Parallela, Orologio/Calendario, Modem
Peso	kg. 4,1

**TOSHIBA
COMPUTER**

TIBER

Distributore maggior informazioni sul portatile T1100	
Nome	
Cognome	
Via	
C.A.P.	
Città	
Prov.	
Telefono	
14C	

re con un Personal Computer, è di possederne uno che sia realmente all'avanguardia, totalmente compatibile ed in grado di comunicare con lo standard più diffuso, leggero e maneggevole, silenzioso e di piccole dimensioni.

La Toshiba, sempre attenta alle esigenze del mercato, ha compreso tutto questo per prima ed è così l'unica casa al mondo che rivoluzionando il mercato dei computers presenta tre portatili in altrettante classi diverse.

Il Toshiba T.1100, per esempio, anche grazie alla applicabilità del virtual disk, è più potente di altri personal esclusivamente da tavolo di prezzo notevolmente superiore.

La capacità di memoria, il nuovo video LCD ad altissima leggibilità ed estremamente riposante, l'elevata risoluzione grafica, sono solo alcune delle caratteristiche che

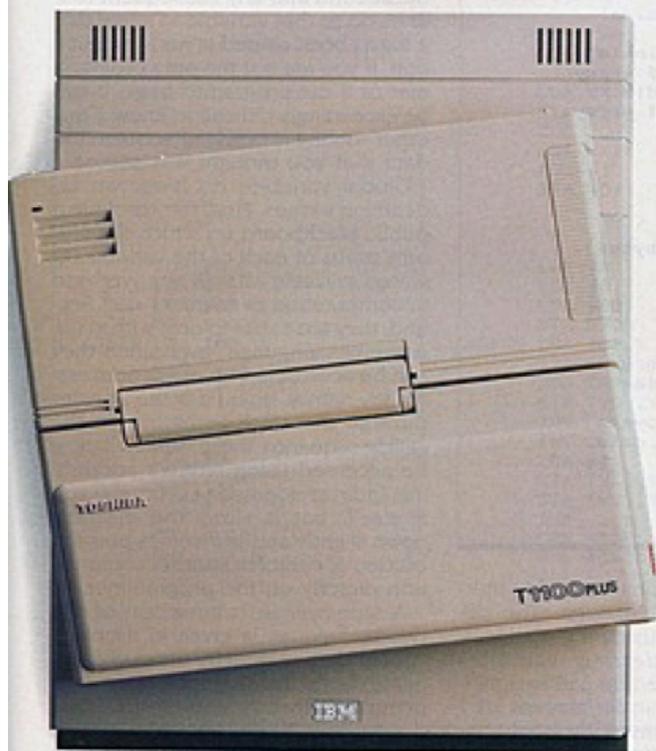
contraddistinguono questo piccolo "giapponese" dalle grandi prestazioni, rendendolo insostituibile nel lavoro quotidiano

Un anno dopo, Toshiba ha presentato il T3100 con hard disk integrato, primo al mondo tra i PC portatili, basato su un processore con performance precedentemente raggiunte solamente dai PC desktop. Grazie a questi risultati, Toshiba ha continuato ad ampliare il proprio portfolio di computer portatili e ha cominciato a venderli anche negli Stati Uniti e in Giappone. I notebook Toshiba, compreso il famoso T1100, hanno continuato a espandere il mercato dei PC, incoraggiando lo sviluppo del settore dei componenti per dispositivi portatili, dando un significativo contributo alla società dell'informazione.

(=)



This Is One Time IBM's Size Is Not An Advantage.



The T1100 PLUS has other things the IBM doesn't. Like serial, parallel and CRT ports that are built-in standards. Not add-on extras. And a high resolution LCD display screen that tilts to many convenient viewing angles. Not just a few.

Of course, the Toshiba T1100 PLUS does have a few things in common with the IBM portable. Like complete IBM compatibility. And two built-in 720K 3.5" disk drives. It's just that it has these things in a smaller, lighter, faster, more portable, more affordable package.

To find out about the T1100 PLUS, call 1-800-457-7777 for the name of your Toshiba dealer.

And see why being smaller than IBM is going to make us one of the biggest names in PCs.

1. IBM PC Convertible is a registered trademark of International Business Machines Corporation.

When it comes to a portable computer, smaller is better. That's one reason the new dual disk drive Toshiba T1100 PLUS has it all over the IBM PC Convertible!

But our portable PC is not only smaller than the IBM, it's lighter. And more powerful. Our 80C86 microprocessor lets you zip through work at up to twice the speed. And our maximum memory capacity is 640K of RAM. Theirs is only 512K.



In Touch with Tomorrow

TOSHIBA

Toshiba America, Inc., Information Systems Division

Inquiry 354

Emulare la macchina ENIGMA



di Lorenzo/2

C'è un po' di confusione, come sappiamo, fra i verbi "emulare" e "simulare". In realtà preferiamo chiamare "emulatore" il sistema hardware/software che replichi il più possibile, eventualmente solo virtualmente, il sistema originale, mentre al sistema che si limita a "funzionare come..." riserviamo il termine di simulatore.

Con questa premessa dovremmo chiamare "simulatore" quello descritto nell'articolo "Enigma Cipher" di Ian Neill pubblicato sul numero di luglio 2014 (issue 25) della rivista "The MagPi".

La rivista è una pubblicazione free che si può scaricare dal sito <https://www.raspberrypi.org/magpi/>.

L'autore, lavorando su un RaspberryPi, ovviamente, costruisce in Python un sorgente che permette di capire i principi che stanno sotto il funzionamento della famosa macchina crittografica Enigma.

Non ci dilungheremo sul funzionamento della Enigma originale, basterà accennare che il suo funzionamento si basava su una serie di rotori meccanici uniti ad altri accoppiatori, sempre meccanici. Il tutto permetteva di inserire un messaggio e avere come output lo stesso messaggio "oscurato" da una chiave e quindi decifrabile solo da chi sia in possesso della chiave o meglio delle regole "meccaniche" che permettono di predisporre la macchina ricevente in grado di decifrare la stringa ricevuta.

Anche la storia di come il team di scienziati inglesi, capitanata dal famoso Alan Turing, sia riuscito a scardinare i segreti del sistema è abbastanza nota, soprattutto oggi che è uscito anche un film dove la storia della scoperta del codice è parte importante.

Come ogni sistema di crittazione di dati anche Enigma aveva i suoi punti deboli... Quello che poi si è rivelato fatale, è che il sistema non permette di annullare il fattore

“stupidità umana”. Solo grazie alla pigrizia di un operatore che “non aveva voglia” di cambiare i codici di base prima di rispedito lo stesso messaggio due volte, si è potuto imboccare la strada della decifrazione.

Il codice messo a punto dall'autore non si propone nulla di particolarmente innovativo, si tratta di stabilire una rappresentazione degli elementi meccanici della macchina e dell'algoritmo di funzionamento. Quello che viene simulato dal programma in realtà è una versione “celebrativa” della macchina crittografica, realizzata come esempio celebrativo e disponibile in un case da cd-rom. Inutile dire che è un giochino, nulla d'altro!

Quella predisposta da Neill è una simulazione di una Pocket Enigma con in dotazione due dischi crittografici. La valenza è didattica e potrebbe essere uno spunto per un esercizio di programmazione per principianti programmatori.

La scelta dell'autore è quella di non inserire direttamente i codici ASCII dei 26 caratteri alfabetici, ma di usare due liste con 26 elementi ognuno dei quali contiene lo spostamento relativo del carattere originale rispetto a quello crittato.

Wheel definitions

```
wheel1 = [-1, 3, -5, 7, -3, 2, 3, -2, 4, -3, -7, 6, -4, 1, -1, 6, 3, -6, 2, -3, -2, -6, 2, 5, -2, 1]
wheel2 = [2, 2, -2, -2, -8, 3, 9, 5, -3, 1, -1, 2, -5, -2, 2, -9, -2, 8, 2, 2, -2, -2, 8, 1, -1, -8]
```

Ad esempio il primo rotore parte con i valori [-1, 3, -5,...]; ipotizzando una configurazione iniziale della sequenza di caratteri (perché anche questa poteva cambiare, anzi la regola era cambiarla ad ogni messaggio) “normale”: “ABCDEF...”, la regola implementata nel primo rotore sarebbe: “se incontri il carattere ‘A’ --> sostituiscilo con ‘Z’

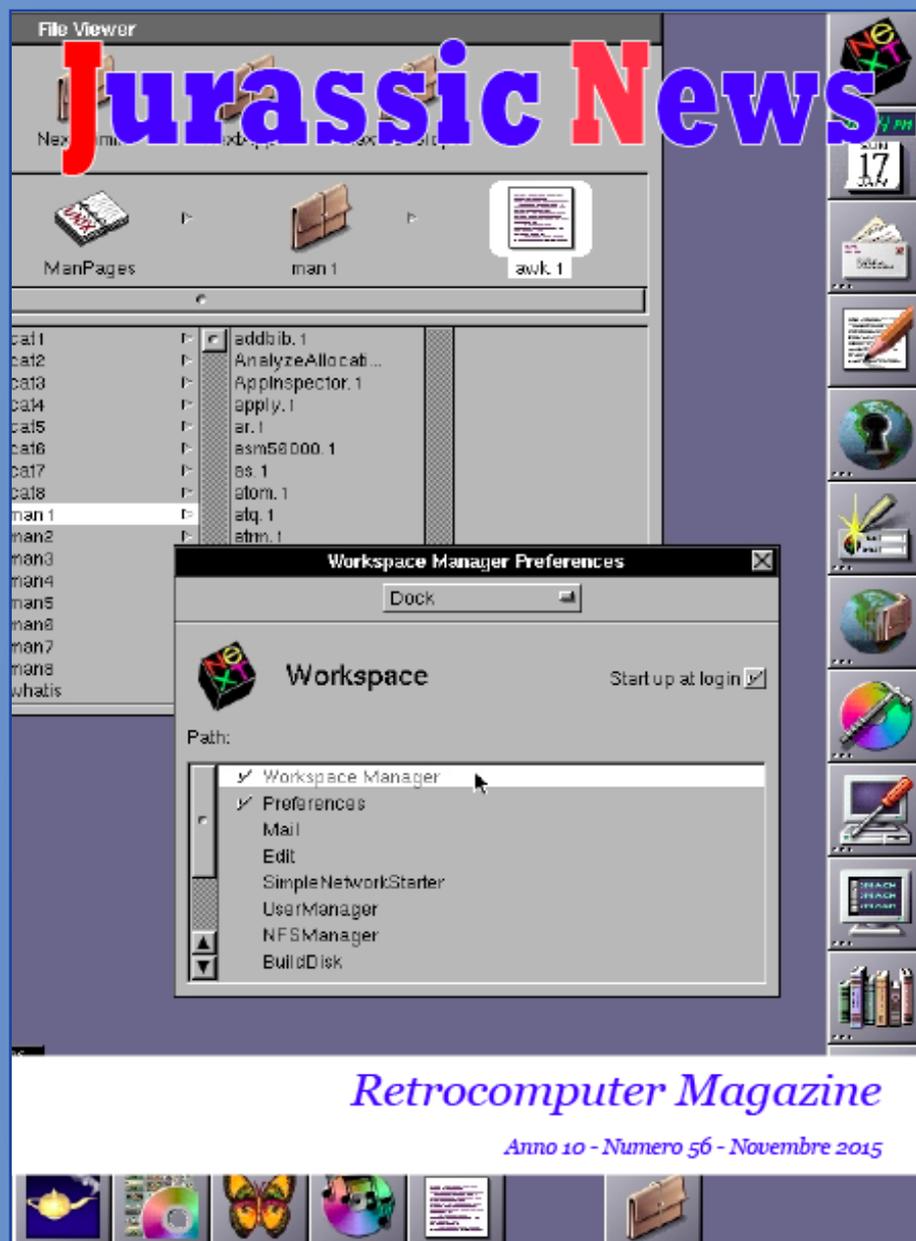


(vai indietro di uno, appunto); se incontri una ‘B’ --> diventa ‘E’ (tre passi in avanti), e così via.

Con la configurazione di esempio il messaggio “Attack at dawn.” viene crittografato nella sequenza: “M UQXZI MGAZE DKS” che a sua volta decifrato da un'altra macchina Enigma viene ricevuto come “ATTACKATDAWNX”.

Aldilà della valenza come simulatore di una vera macchina Enigma, il codice deve essere preso come esempio di introduzione alla simulazione, null'altro. Il fatto di implementarlo su un Ras-Pi se vogliamo apre la strada per chi lo volesse per sviluppare un sistema meccanico di input/output che lo renda più simile ad un vero oggetto fisico.

(=)



**Indovinate quale sistema
sarà protagonista della prova
sul prossimo numero...**