LE CHIP CARD

Le chiavi elettroniche diventano sempre più difficili da espugnare, ma è grazie a una nuova tecnologia che si arriva alla sicurezza completa: le chip card infatti assicurano una completa inespugnabilità di quanto protetto

Paola Sbrana - 1ª parte

Recentemente, anche in Italia si vedono sempre più applicazioni delle cosiddette chip-card, ovvero di quelle schedine simili a una carta da visita con sopra un piccolo chip. Una fra le prime fu senza dubbio la security-card che veniva rilasciata in cambio di un abbonamento ad alcuni circuiti di ricezione satellitare, poi ne abbiamo trovate alcune nel settore bancario ed infine ne troviamo sempre più frequentemente nel campo della sicurezza.

Cercheremo, quindi, con una piccola serie di articoli dedicati, di rispondere ai tanti quesiti che queste carte fanno porre ai nostri lettori, spiegando dapprima che cosa sono, come si utilizzano e dove si trovano, per terminare con un circuito di chiave elettronica ed un piccolo terminale portatile per la programmazione di alcune di queste.

I modelli principali

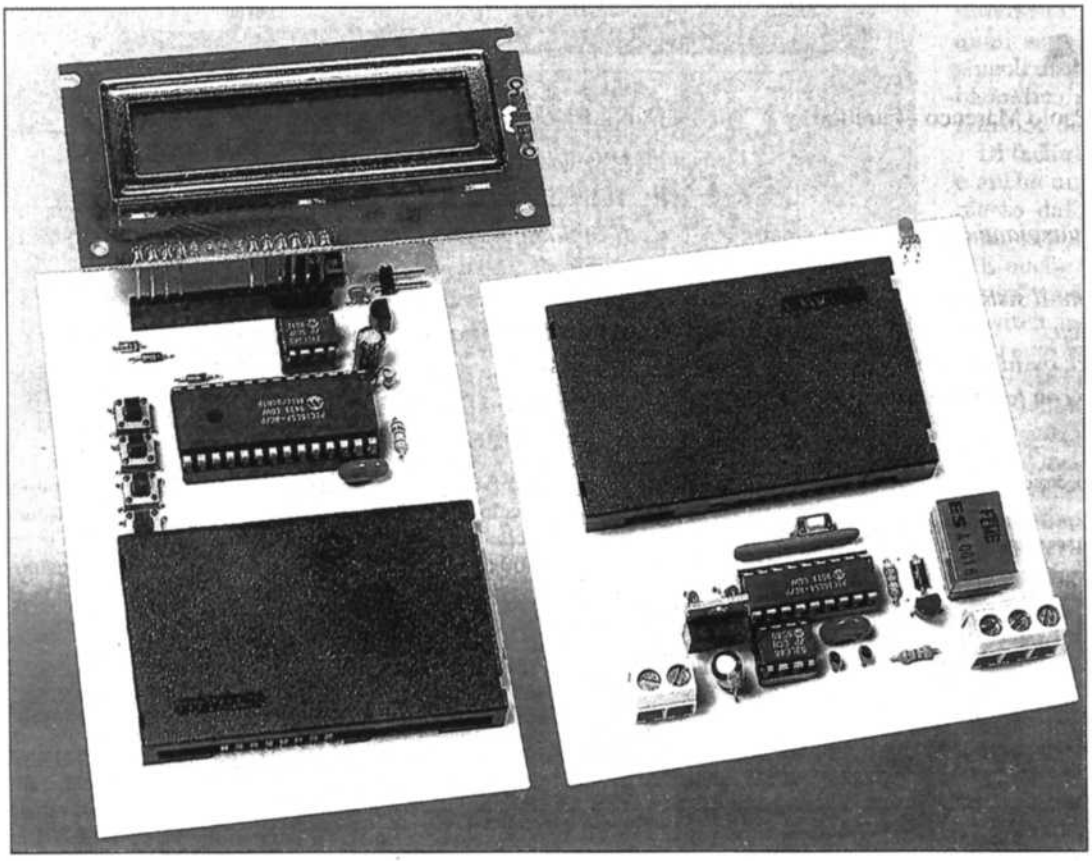
Prima di analizzare dettagliatamente il modello di carta che noi utilizzeremo, vediamo quante altre carte si trovano in commercio: innanzitutto, la prima distinzione che dobbiamo fare è la differenza tra una "vera" chip-card ed un'altra detta erroneamente chip-card, ma che non ha niente del chip inteso come circuito integrato. Esistono infatti carte che hanno al loro interno solamente memorie (di tipo PROM, EPROM o EEPROM, RAM), altre che abbinano alla memoria interna una "intelligenza" (periferiche di dialogo, contatori, PLD) ed infine altre che possiedono veri e propri microcontroller.

Le prime sono le più facili da trattare, ma anche le meno affidabili dal lato della sicurezza. Le altre invece vengono attualmente impiegate nei più svariati settori ed in special modo, le ultime sono prevalentemente sfruttate nel settore bancario.

Noi tratteremo le carte "intermedie", ovvero quelle che hanno al loro interno sia della memoria (di tipo prevalentemente EEPROM) sia alcune periferiche che le rendono intelligenti. Esistono molte carte appartenenti a questo segmento, con capacità di memoria da 256 byte a 8kb e con sistemi di accesso e di codifica da semplici a molto complessi.

Tanto per avere un'idea di che cosa troviamo sul mercato, diciamo che ci sono carte che vengono lette e scritte molte volte senza artifici particolari, mentre ce ne sono altre che possono essere sempre lette ma per essere riscritte è necessario eseguire una procedura iniziale abbastanza complessa con un codice di accesso (il classico PIN) preprogrammato. Dopo n tentativi di scrittura con codice errato, la carta si blocca e non permette più la riscrittura neanche con il codice corretto.

Ce ne sono, infine, alcune che necessitano del codice PIN anche per la sola lettura delle informazioni. Capite quindi quanto vasto sia questo settore e perché noi ci siamo dovuti orientare su di un solo tipo di queste carte, anche per consentire a tutti di comprendere bene il funzionamento e metterlo poi in pratica senza rischio di incomprensioni.



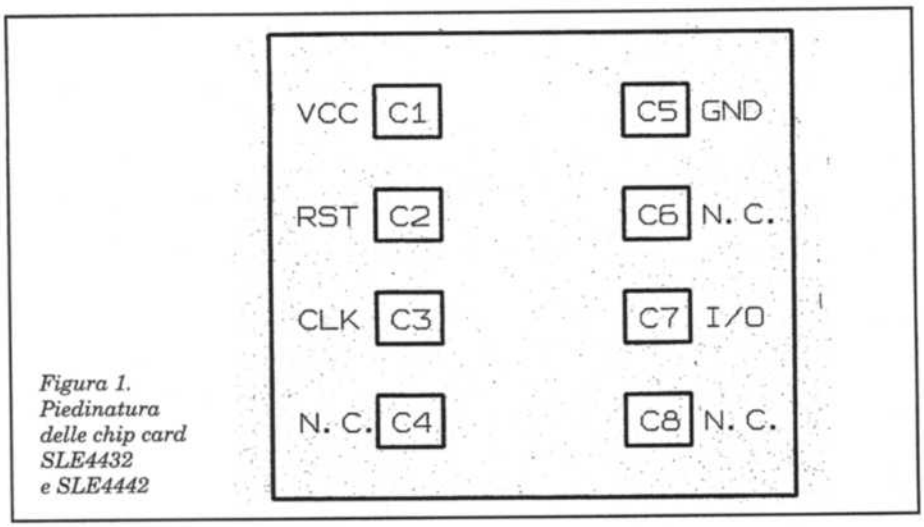


Figura 1. Piedinatura delle chip card SLE4432 e SLE4442

Le SLE4432 e 4443

Le carte che abbiamo individuato per i nostri scopi, sono le SLE4432 e SLE4442 della Siemens, gentilmente forniteci dalla Veron, una compagnia Olivetti Telemedia che si incarica di distribuirle in tutto il territorio nazionale.

Queste carte sono entrambe definite "intelligenti" in quanto hanno a bordo alcune periferiche e una di queste ha la protezione dalla scrittura non autorizzata. Le caratteristiche principali sono:

- Organizzazione di 256 x 8 byte in EEPROM.
- Indirizzamento diretto e sequenziale.
- Protezione irreversibile sulla scrittura dei primi 32 byte.

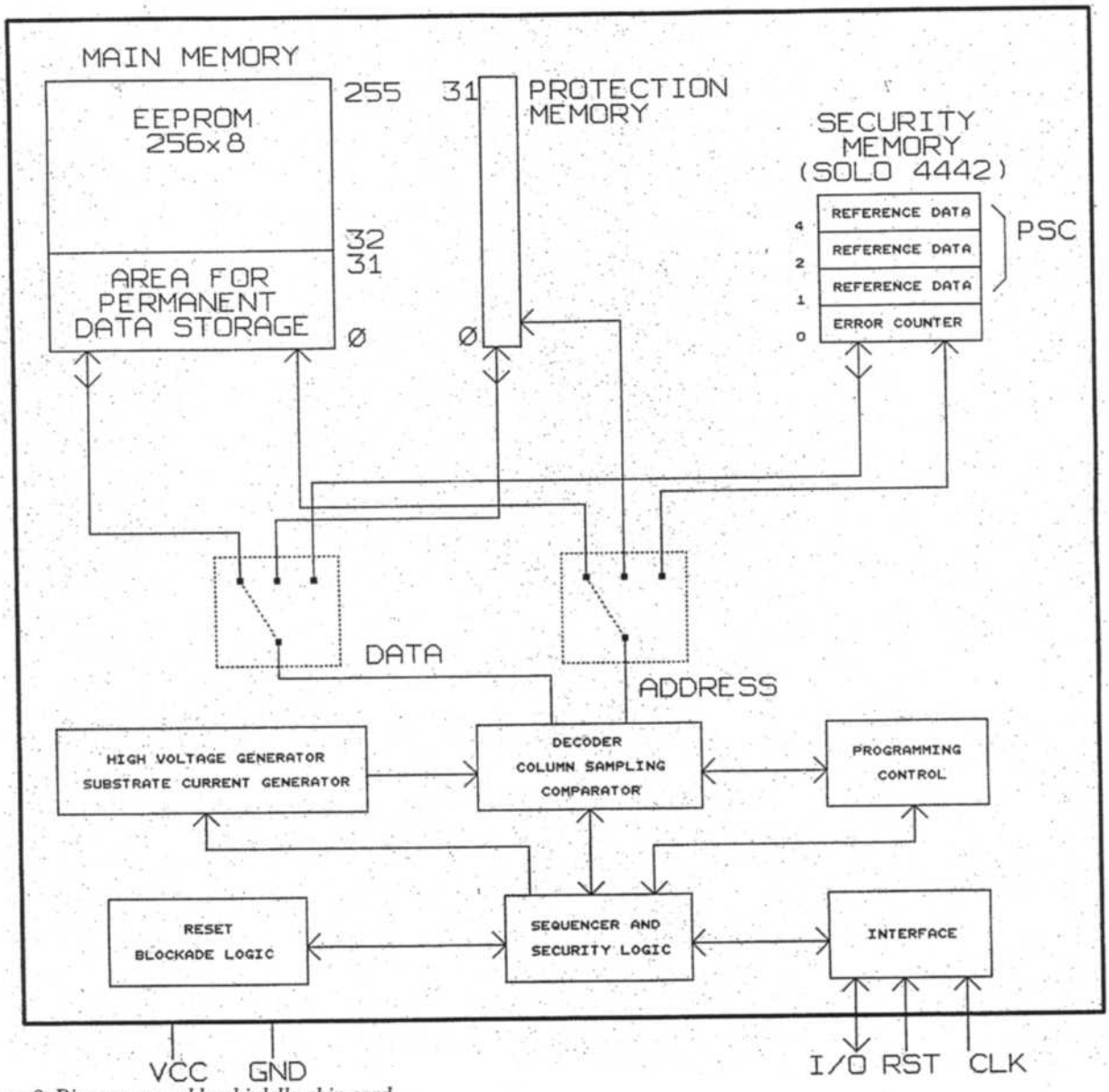


Figura 2. Diagramma a blocchi della chip-card

Figura 3. Schema di accesso alle due memorie

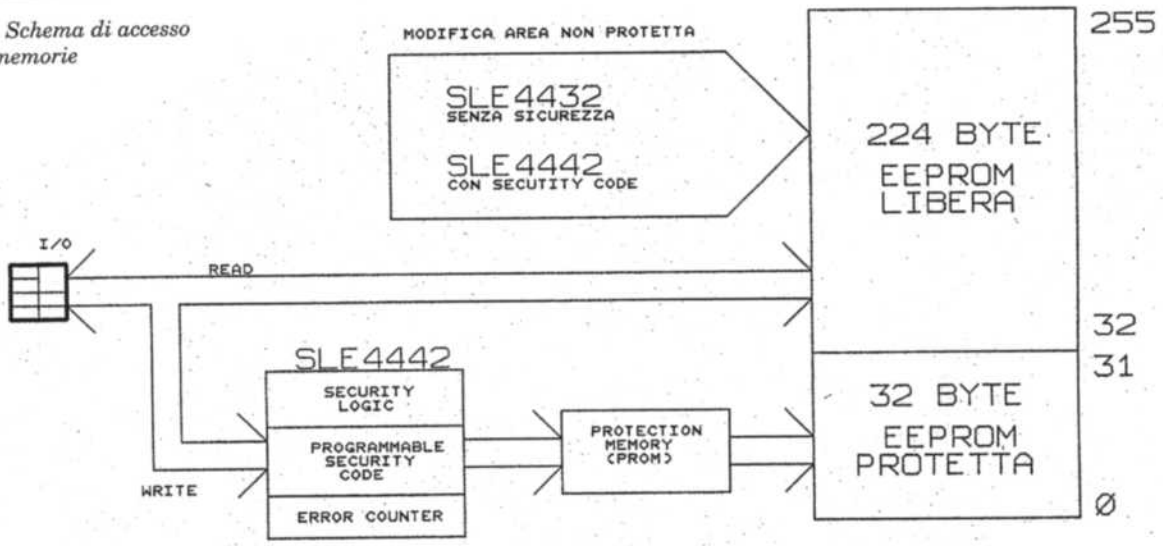
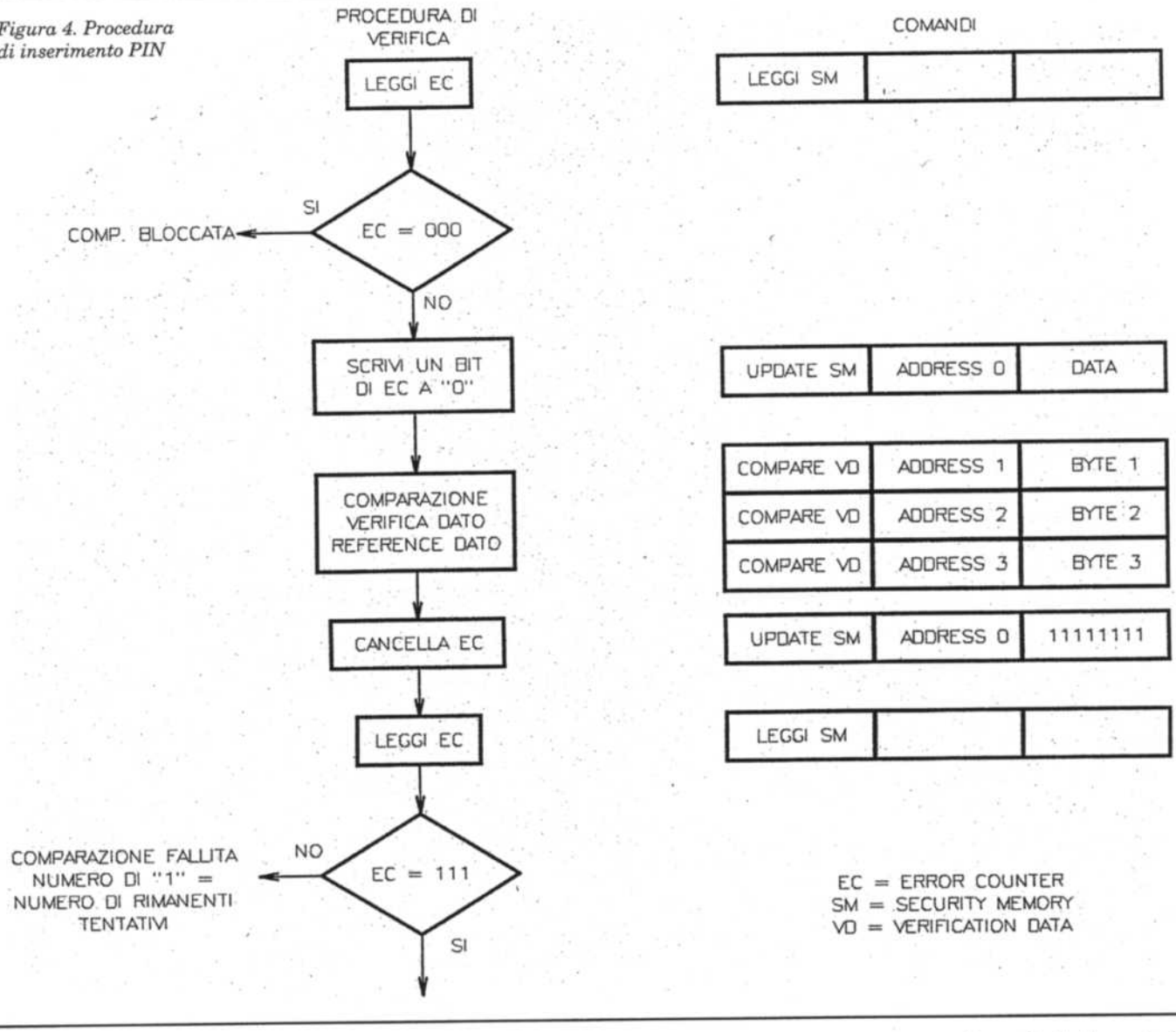


Figura 4. Procedura di inserimento PIN



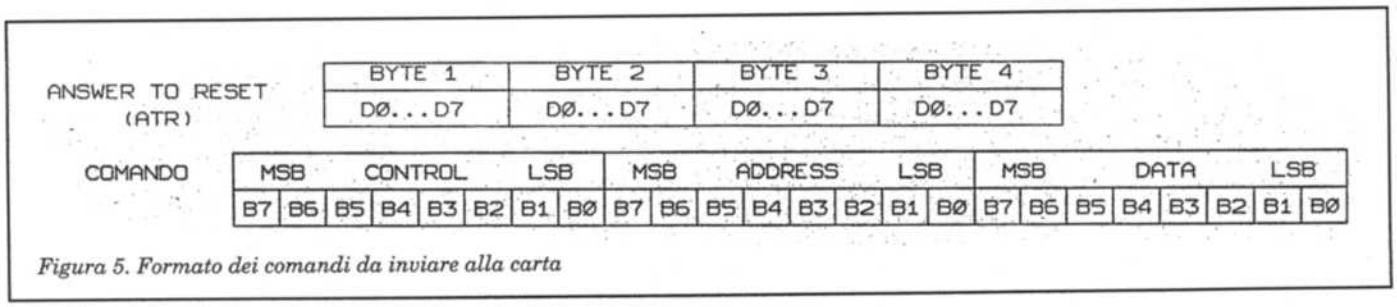


Figura 5. Formato dei comandi da inviare alla carta

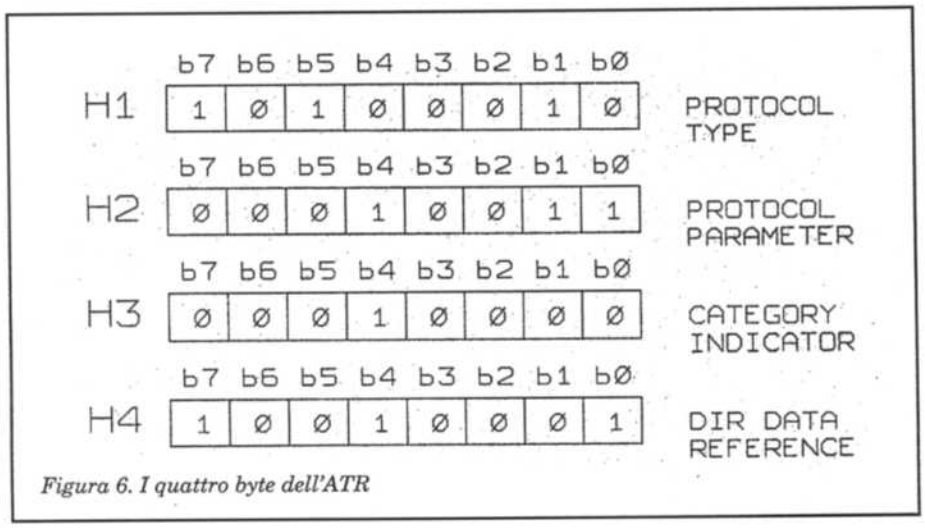


Figura 6. I quattro byte dell'ATR

- Protection memory di 32 byte.
- Protocollo a 2 fili (clock e dato).
- Fine delle operazioni indicato dalla carta.
- Answer-to-reset in accordo allo standard ISO 7816-3.
- Tempo di programmazione per byte: 5ms.
- Minimo numero di scritture: 10.000.
- Ritenzione dei dati per almeno 10 anni.
- Contatti e interfaccia seriale in accordo con lo standard ISO 7816 (trasmissione sincrona).
- Solo per la SLE4442: i dati possono essere variati solamente dopo i 3 byte del security code.

La nostra scelta di impiegare la serie SLE4432 è nata dal fatto che se è possibile leggere una carta è egualmente possibile copiarla, quindi non abbiamo ritenuto opportuno prendere in considerazione le SLE4442.

In accordo con lo standard ISO7816, in Figura 1 vediamo la piedinatura di tali carte. Il contatto C1 (VCC) deve essere connesso ad una tensione di alimentazione di 5 volt precisi.

Il contatto C2 (RST) corrisponde al terminale di reset. Il contatto C3 (CLK)e è la linea di clock e il contatto C7(I/O) quella dei dati.

Il C4, il C6 ed il C8 sono non connessi, mentre il contatto C5 (GND) è la massa.

Analizziamo ora lo schema a blocchi della chip-card aiutandoci con la Figura 2. Le linee di controllo entrano nel blocco INTERFACE e da qui vengono indirizzate ai vari blocchi interessati.

È presente un blocco di reset, uno per la generazione della tensione di programmazione della EEPROM, un controllore di programma, un decoder degli indirizzi ed infine l'area della memoria EEPROM.

Questa area è suddivisa in due blocchi funzionali:

- Il primo (dall'indirizzo 0 al 31 decimale) può essere protetto andando a scrivere sulla PROTECTION MEMORY, mentre sul secondo blocco (dall'indirizzo 32 al 255 decimale) è sempre possibile scrivere o leggere. Da notare che la PROTECTION MEMORY è di tipo PROM, quindi una volta scritta non potrà più essere riscritta, rendendo irrimediabilmente non riscrivibili anche i primi 32 byte della EEPROM. Questa la parte comune alle due carte SLE4432 e SLE4442.
- Ilsecondo, invece, ha in più il blocco detto SECUTITY MEMORY, ovvero un'area di 4 byte dove risiede un numero PIN di tre byte ed un contatore del numero di tentativi di accesso non permesso. Quando questo numero supero i tre consecutivi, la EEPROM non potrà più essere riscritta, ma soltanto letta. Si ricorda che l'inserimento del numero di PIN serve soltanto per l'accesso alla riscrittura della EEPROM e non alle operazioni di lettura.

In Figura 3 vediamo lo schema di accesso alle due memorie.

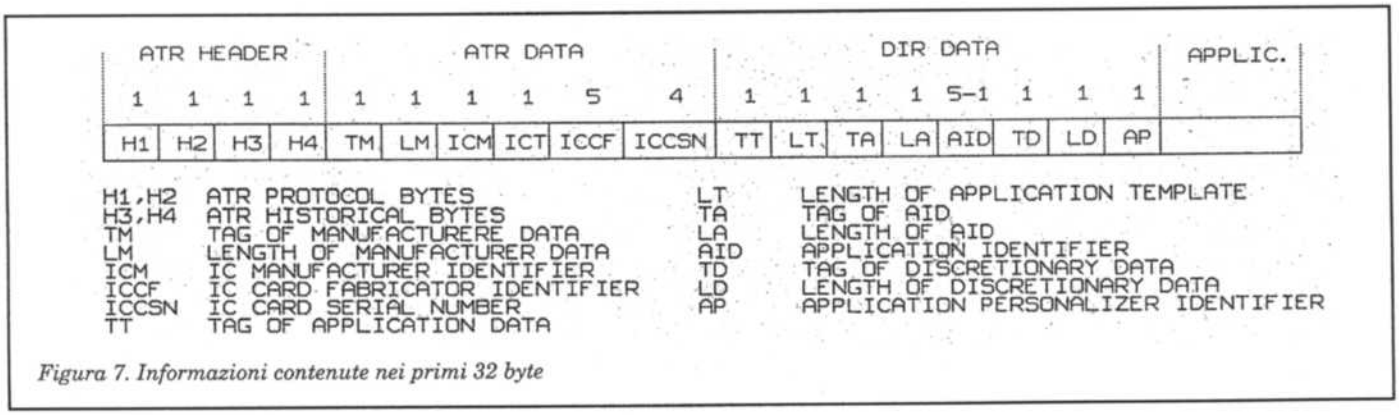


Figura 7. Informazioni contenute nei primi 32 byte

La procedura di verifica PIN

In Figura 4 abbiamo lo schema a blocchi della procedura per l'inserimento del PIN e la conseguente abilitazione della carta SLE4442 alla riscrittura: viene dapprima letto il contatore di errore, poi si va a vedere se questo è uguale a 000.

Se ciò accade, la comparazione è bloccata, viceversa si scrive a 0 un bit dell'error counter. Poi si passa alla comparazione dei tre byte del PIN e si cancella l'error counter. Infine, si legge nuovamente l'error counter e si vede se è uguale a 111. In caso positivo la comparazione è terminata positivamente, viceversa la carta rimane bloccata in riscrittura.

La procedura ATR

In accordo con lo standard ISO7816, le due carte, ad un ben preciso segnale di reset, "rispondono" quattro byte tramite la procedura detta Answer-To-Reset.

In Figura 5 troviamo i quattro byte inviati dalla carta all'ATR ed il formato tipico dei comandi che la carta si aspetta dopo un ATR: i primi otto bit detti CONTROL BYTE identificano il comando vero e proprio che la carta dovrà eseguire, il secondo byte detto ADDRESS BYTE indica l'indirizzo della locazione di memoria dove eseguire il comando ricevuto ed il terzo byte, detto DATA BYTE, consente il passaggio dei dati inerenti la locazione precedentemente indicata.

Tutti i byte sono inviati a partire dal bit meno significativo (LSB first).

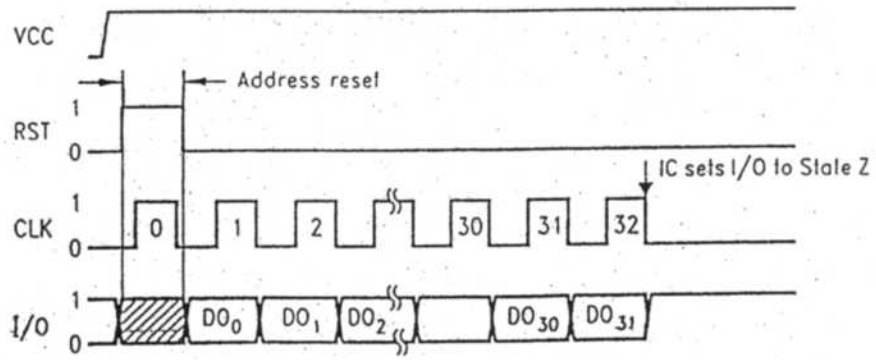


Figura 8. Timing diagram per l'ATR

In Figura 6 sono visibili i 4 byte dell'ATR: il primo (PROTOCOL TYPE) indica il tipo di protocollo impiegato, il secondo (PROTOCOL PARAMETER) indica i parametri per quel tipo di protocollo, il terzo (CATEGORY INDICATOR) indica la categoria di appartenenza ed il quarto (DIR DATA REFERENCE) ci da informazioni sui dati contenuti nella EEPROM.

In Figura 7, invece, vediamo come sono distribuite le informazioni nei primi 32 byte della memoria EEPROM.

Tutte queste informazioni, ricordiamo che sono proteggibili dalla riscrittura tramite la protezione della PROTECTION MEMORY, quindi per fare un esempio, un venditore di carte potrebbe inserire nei byte preposti il suo codice cliente e poi proteggere questa informazione da qualsiasi altra riscrittura (esempio data scadenza garanzia).

L'interfaccia con l'esterno

Per poter dialogare con il mondo esterno, queste carte necessitano di un'alimentazione a 5 volt (connettori VCC e GND), di un terminale di reset (RST), di un terminale di clock (CLK) e di uno per i dati (I/O). Il terminale di reset, viene per lo più usato solo nella fase iniziale di una transazione, per far partire la procedura ATR: in Figura 8 ne vediamo il timing diagram. A fronte di un impulso di reset lungo almeno la durata di un impulso di clock, la carta invia i 4 byte dell'ATR precedentemente visti, ovviamente sincronizzati con la linea del clock.

Una volta terminata la procedura ATR, la carta è pronta a ricevere comandi, nella forma vista prima. A ogni comando, come si vede chiaramente nella Figura 9, seguirà una risposta, sempre gestita dal segnale di clock.

Il prossimo mese vedremo un'applicazione pratica delle carte tipo SLE4432, realizzando una chiave elettronica con oltre 280 miliardi di combinazioni.

Le carte potranno essere acquistate sia vergini che già codificate.

Nel primo caso prossimamente presenteremo un programmatore per le sole SLE4432 della sola memoria EEPROM.

Si ringrazia la Veron Spa - Via Caldera, 21 - Milano per la collaborazione data nell'acquisizione delle informazioni citate. La Veron Spa è disponibile telefonicamente al numero 02/482151 per chiarimenti commerciali.

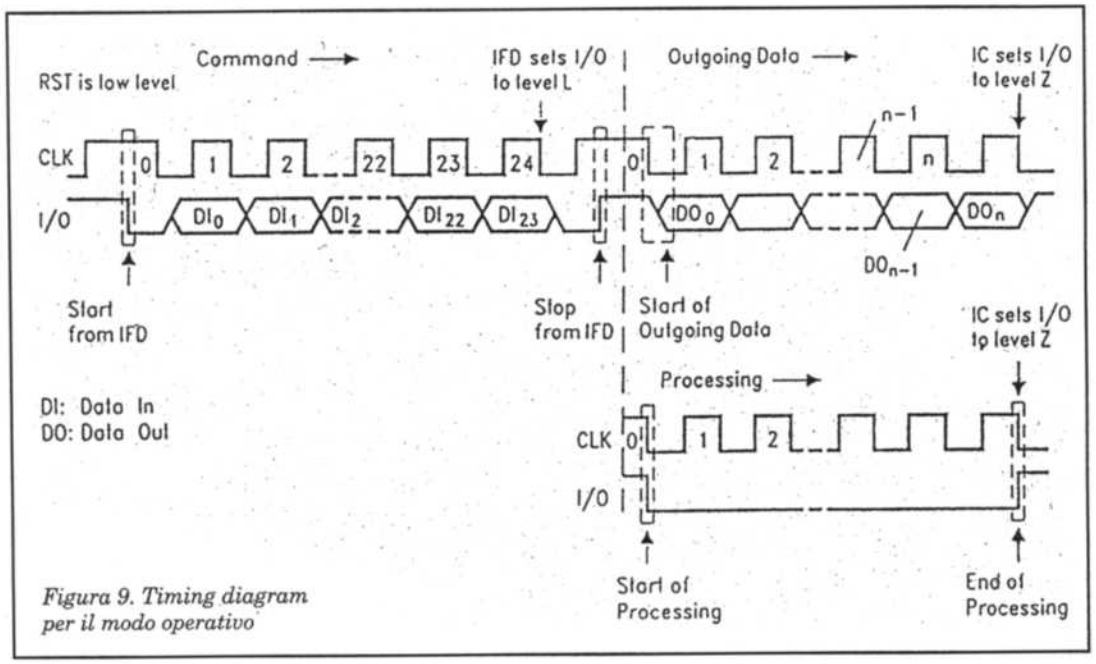


Figura 9. Timing diagram per il modo operativo

continua

LE CHIP-CARD

Le chiavi elettroniche diventano sempre più difficili da espugnare: con le chip card è quasi impossibile! Ecco quindi l'origine di questo progetto, semplice ma allo stesso tempo estremamente pratico e, fattore da non trascurare, decisamente economico

Paola Sbrana - 2ª parte

Continuiamo il nostro viaggio alla scoperta delle chip-card con la presentazione di una chiave elettronica dalle prestazioni eccezionali: oltre 280 miliardi di combinazioni!

La prima applicazione tipica di una chip card è in effetti sempre orientata alla sicurezza, e noi abbiamo voluto proporre una chiave elettronica che possa sostituire egregiamente qualsiasi chiave voi abbiate già in uso, sia questa di tipo meccanico che elettronico. In pratica, lo stesso numero di combinazioni era possibile ottenerlo anche con chiavi dotate di EEPROM seriale tipo quella presentata nel Febbraio '94, ma sicuramente la loro copia e la loro immunità ai tentativi di effrazione sarebbe stata inferiore a quella che proponiamo adesso.

Analizziamone i motivi principali: primo tra tutti la riproducibilità. Oggigiorno, duplicare una EEPROM seriale è diventata un'operazione banale, che è possibile effettuare sia tramite una comune porta parallela da computer, sia attraverso l'impiego di programmatori appositi che costano ormai pochissimo.

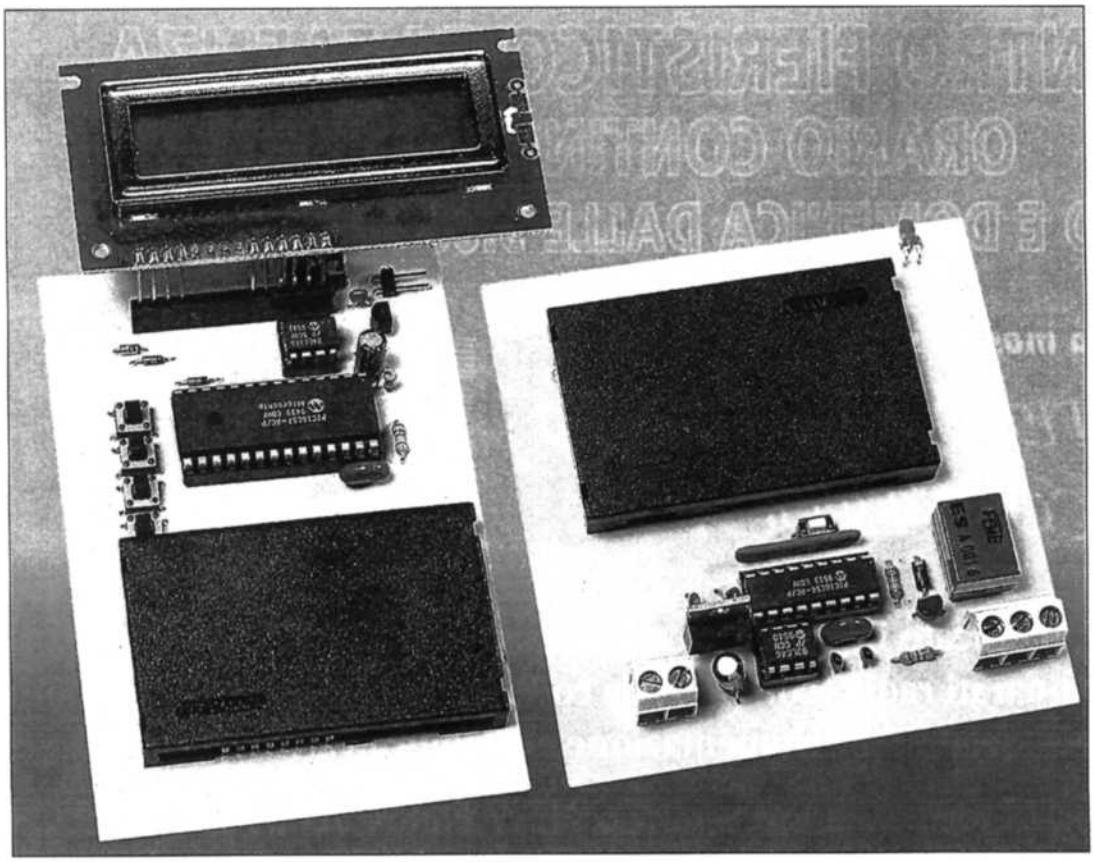
Duplicare una chip card, invece, risulta molto più difficile, specie se non è conosciuto il tipo di carta da copiare (abbiamo già stabilito il mese scorso che ci sono centinaia di chip card diverse sul mercato, ma alla vista sembrano tutte identiche).

Per lo stesso motivo, anche la reperibilità di tali carte non è immediata come può essere per le EEPROM seriali, che si possono trovare presso qualsiasi negozio di componenti elettronici. In alcuni casi, ad esempio, le chip card sono vendute solamente a ditte specializzate dopo stipula di contratti che ne limitano l'impiego e la divulgazione delle particolarità specifiche.

Nel nostro caso, questi problemi non sono reali, perché abbiamo scelto un tipo di carta molto comune agli addetti che non presenta particolari criteri di segretezza.

Un altro blocco "psicologico" che fino ad adesso ha rallentato l'impiego di tali carte era rappresentato dal particolare connettore necessario per l'interfacciamento, connettore che adesso viene venduto anche dalla RS ad un prezzo inferiore alle 15.000 lire.

Tra i molti vantaggi invece che la chip card offre, oltre a quelli visti precedentemente, dobbiamo evidenziare la portabilità di tale carta, che potrà essere riposta vicino alla carta di credito o al bancomat.



La nostra proposta

La chiave che proporremo tra poco, è molto simile per funzioni ad altre già presentate, in quanto consente di acquisire in una EEPROM seriale il codice di una qualsiasi carta e poi di attivare un relè quando rileva la presenza di una carta identica.

In Figura 10 è possibile vedere lo schema elettrico del circuito ideato per la nostra chiave.

Il microcontroller IC1 (un PIC16C54XT appositamente programmato) gestisce tutte le fasi di lavoro della chiave.

La memoria IC2 viene impiegata per il salvataggio del codice acquisito dalla carta, in modo tale da garantirne la con-

sistenza anche dopo un'accidentale assenza di alimentazione. L'alimentazione a 5 volt viene garantita da IC3, un regolatore tipo 7805. L'interfacciamento del microcontroller con la carta è diretto, con l'ausilio di resistenze di pull-up dato che le linee della carta sono tipo open drain.

Le uscite del circuito sono due: la prima comanda un relè, la seconda un diodo Led di segnalazione.

Vediamo allora il funzionamento della chiave attraverso il diagramma a blocchi del software implementato in IC1 e visibile in Figura 11.

Alla locazione di reset, il PIC trova subito una chiamata ad una subroutine che ha il compito di settare correttamente le porte ed i registri interni delle periferiche e della RAM. Poi IC1 si pone in attesa di un evento tra i due seguenti: l'arrivo di una carta oppure la pressione del pulsante di programmazione P1.

Supponiamo che venga premuto il pulsante P1 (acquisizione codice). Allora il Led L1 inizia a lampeggiare con una frequenza di circa 2 Hz e continua così o fino al termine di un tempo fissato in circa 10 secondi, o fino all'arrivo di una chiave.

Se nessuna chiave (chip-card) viene inserita entro i 10 secondi, il Led si spegne ed il programma torna al test iniziale. Al contrario, se viene inserita una carta il microcontroller ne legge 6 byte dislocati in una posizione ben precisa, li memorizza nella EEPROM e poi spegne il Led L1 e torna al test iniziale.

Questo per quanto riguarda la fase di acquisizione del codice. Supponiamo ora di aver già memorizzato un codice e che al test iniziale si presenti una carta.

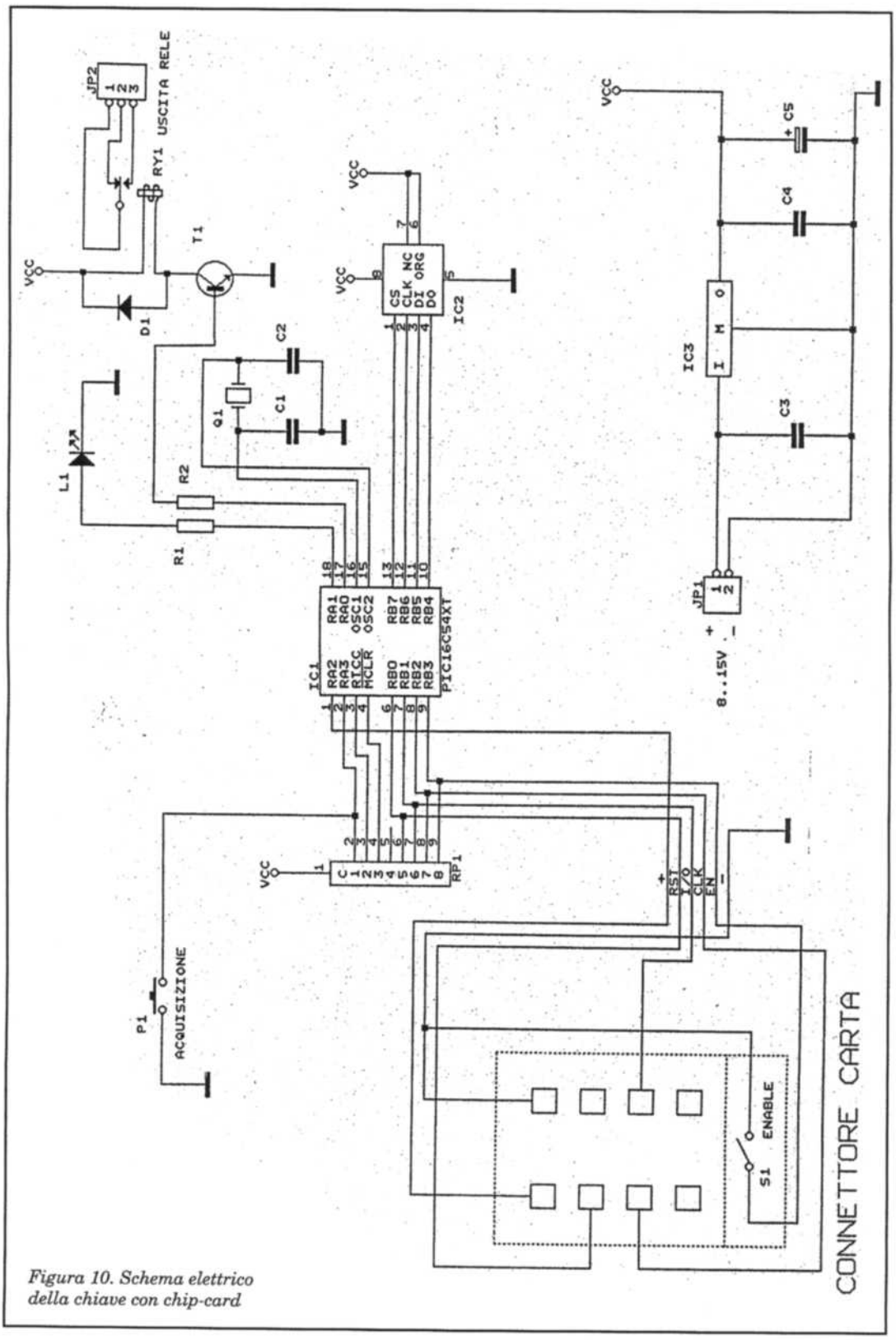


Figura 10. Schema elettrico della chiave con chip-card

Tale carta viene letta all'indirizzo del codice e successivamente viene eseguito un confronto con il codice contenuto in IC2. Se l'esito di questo confronto risulta positivo, il Led L1 ed il relè si attivano e rimangono in tale stato fino a quando la carta rimane presente nell'inseritore.

Viceversa, se il confronto dà esito negativo, il Led L1 inizia a lampeggiare alla frequenza di circa 4 Hz e così rimane fino a quando la carta non venga rimossa dall'inseritore.

Montaggio

Per la realizzazione della scheda, su cui verrà montato anche l'inseritore, potrete fare riferimento alla traccia per il circuito stampato proposta in Figura 12.

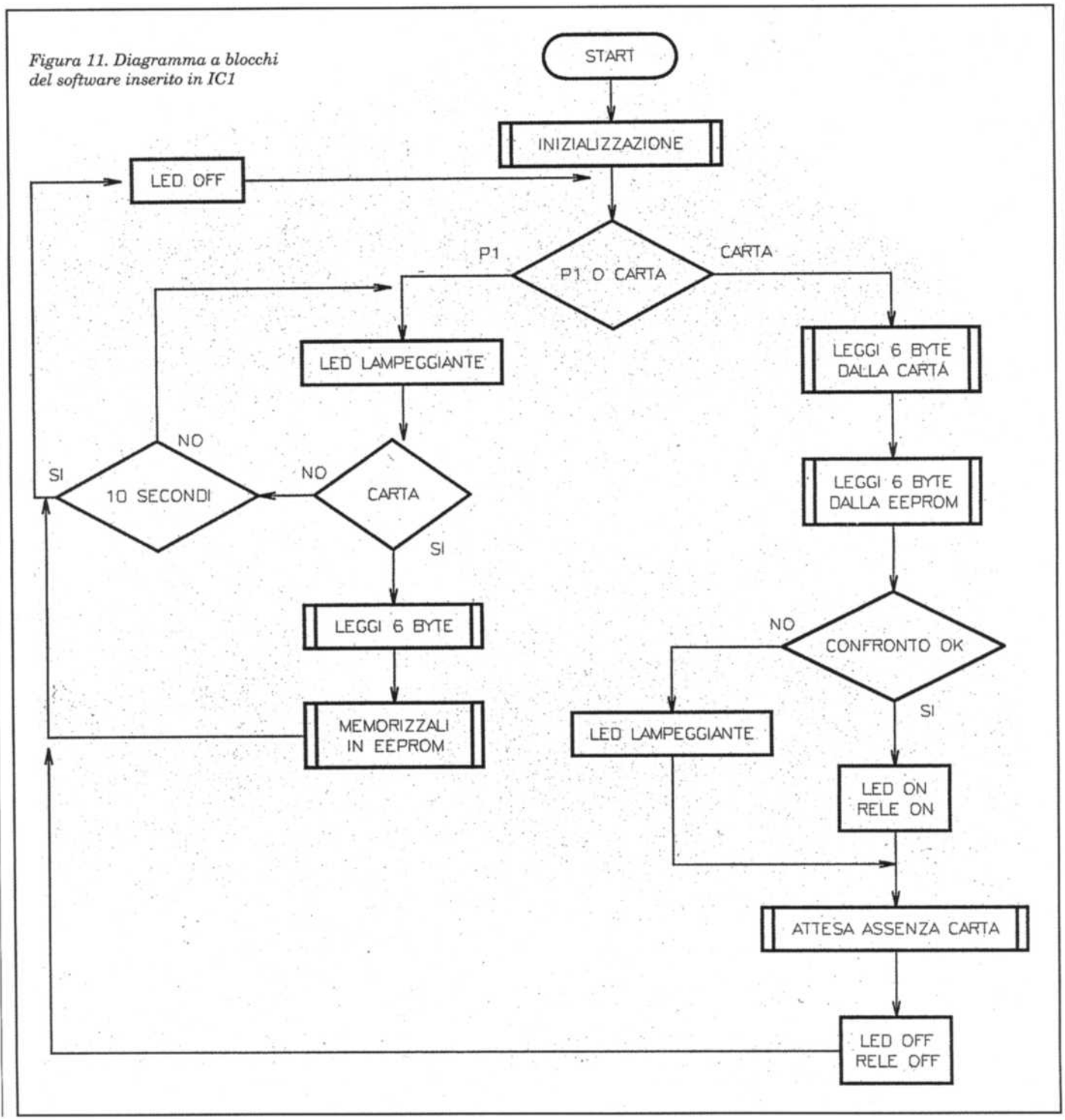
In Figura 13 invece troviamo il piano di cablaggio dei vari componenti, oltre alle connessioni da effettuare con l'esterno.

Fate attenzione ai componenti polarizzati, compresa la rete resistiva: molto spesso infatti, abbiamo occasione di riparare circuiti non funzionanti solo perché sono state inserite erroneamente le reti resistive. Se non riuscite a reperirle dal vostro negoziante di fiducia, potrete autocostruirvele con delle resistenze normali, posizionate verticalmente e con un capo in comune.

Per IC3, potrete sia impiegare un 78L05 che un 7805, a seconda del relè che desiderate utilizzare: un 78L05 sopporta un carico di 100-120 mA al massimo, contro i 700-800 mA del 7805.

Il pulsante P1 potrà anche lasciare il posto a un connettore bipolare senza cortocircuito, in quanto pensiamo che l'acquisizione di un codice avverrà alla prima installazione e poi nel solo caso in cui una carta venga persa e quindi sia necessario riprogrammare tutte le altre con un diverso codice.

Figura 11. Diagramma a blocchi del software inserito in IC1



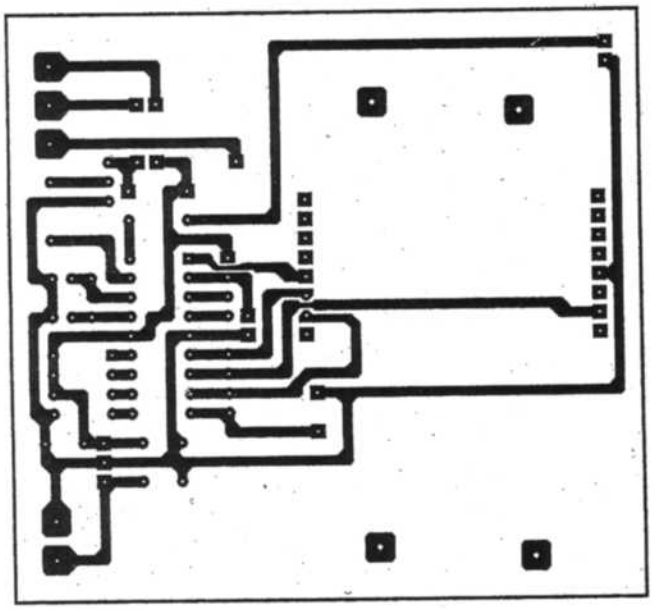


Figura 12. Circuito stampato, scala 1:1

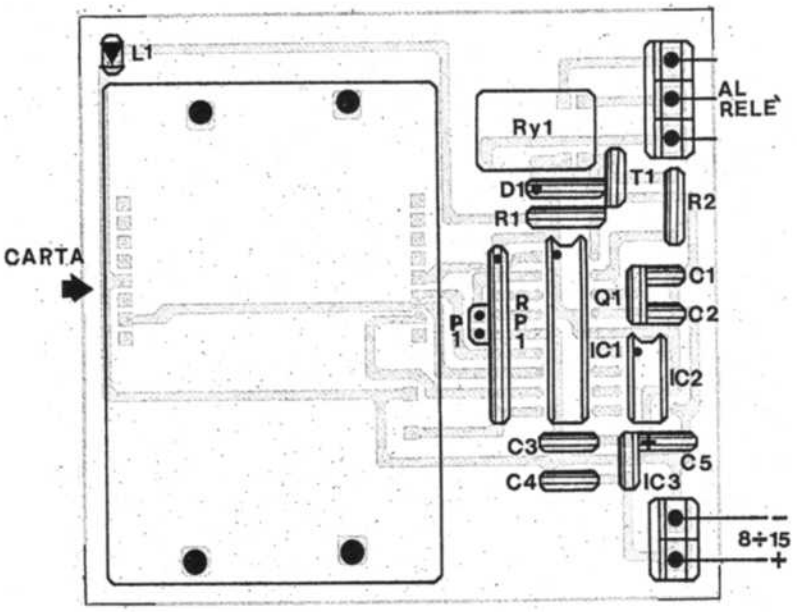


Figura 13. Disposizione dei componenti

A proposito di tali carte, possiamo sia fornirle vergini che già programmate, basterà che lo specifichiate espressamente: sia le une che le altre potranno comunque essere riprogrammate attraverso il programmatore stand-alone (senza uso quindi di computer) che presenteremo il prossimo mese. A tale proposito, occorre specificare che le carte devono avere il codice memorizzato nelle locazioni 84h, 85h, 86h, 87h, 88h, 89h.

Collaudo

Dopo aver effettuato un controllo visivo per essere sicuri di aver posizionato correttamente tutti i componenti, si dovrà dare alimentazione al circuito. Sia il Led che il relè dovranno rimanere spenti. Premiamo ora il pulsante P1 e verifichiamo che il Led comincia a lampeggiare. Attendiamo 10-12 secondi e verifichiamo che il Led si spegne. Premiamo nuovamente il pulsante P1 ed inseriamo una carta programmata. Il Led dovrà spegnersi. Togliamo ora la carta e rinseriamola: sia il Led che il relè dovranno attivarsi per tutto il tempo che la carta rimane inserita nel connettore.

Se adesso proviamo ad inserire una carta con codice diverso, il relè dovrà rimanere disattivato, ma il Led invece dovrà iniziare a lampeggiare.

Per simulare una carta con codice diverso, potrete inserire la stessa carta con il chip rivolto dalla parte opposta dei contatti. La carta verrà rilevata comunque perché la sua presenza è determinata dalla chiusura del contatto S1 posto alla fine del connettore.

Le possibili applicazioni di questo circuito sono infinite e limitate soltanto dalla nostra fantasia. La più classica è quella di sfruttarlo per aprire delle porte con elettroserratura, ma si può inserire anche come chiave d'accesso a cassaforti ed antifurti, come periferica per impianti di controllo accessi, come chiave per macchinette distributrici di bevande e anche come lock per i vostri computer!

L'autore dell'articolo è disponibile per apportare modifiche al software di gestione del microcontroller in funzione delle specifiche esigenze telefonando al numero 0347/2643514.

Nel prossimo numero tratteremo, come già accennato, ad un piccolo programmatore stand-alone, ovvero che non deve essere collegato al computer, ma che avrà come interfaccia utente solamente 4 pulsanti ed un display alfanumerico.

Si ringrazia la Veron S.p.A. - Via Caldera, 21 - Milano - per la collaborazione data nell'acquisizione delle informazioni citate.

La Veron S.p.A. è disponibile telefonicamente al numero 02/482151 per chiarimenti commerciali.

continua

ELENCO COMPONENTI

Semiconduttori

IC1: PIC16C54XT programmato (0347/2643514)
IC2: 9306 o 9346
IC3: 7805
T1: BC337
D1: 1N4001
L1: Led verde 3mm

Resistori

R1: 270 Ω
R2: 4,7 kΩ
RP1: Rete resistiva 10 kΩ

Condensatori

C1, C2: 10 pF
C3, C4: 100 nF
C5: 47 uF

Varie

Q1: Oscillatore o quarzo da 3,579545 MHz
RY1: Relé 5V 1sc.
P1: Pulsante n.a.
Connettore: codice RS 453-785

LE CHIP CARD

Le chiavi elettroniche diventano sempre più difficili da espugnare: le chip card sono sicurissime! Da qui l'idea di realizzare questi tre articoli per illustrare le potenzialità di questo mondo

Paola Sbrana - 3ª parte

Completiamo la trattazione delle chip-card tipo SLE4432 e SLE4442 della Siemens proponendo un programmatore di tipo stand-alone con cui potremo leggere, programmare, modificare e programmare le chip-card del tipo SLE4432, cioè quelle da noi dettagliatamente illustrate e con le quali abbiamo realizzato il mese scorso una chiave elettronica.

Se ben ricordate, le carte potevano essere acquistate già programmate con un codice interno, oppure vergini, in modo tale da poterle programmare e riprogrammare a piacimento.

Con il programmatore che tra poco analizzeremo ciò sarà possibile in modo estremamente semplice, anche per i non esperti di programmazione.

Abbiamo preferito sviluppare un programmatore di tipo stand-alone per due motivi principali: primo perché non tutti coloro che possono avere interesse per le chip-card sono dotati di un computer, secondo perché un computer non è facilmente trasportabile, quindi se ad esempio un installatore di antifurti avesse necessità di programmare carte direttamente al domicilio dei clienti potrebbe incontrare delle difficoltà.

Nella versione stand-alone, invece, il programmatore è di tipo "palmare", alimentato con una batteria da 6 o da 9 volt ed offre un'interfaccia utente molto semplice: 4 pulsanti ed un display alfanumerico per la massima chiarezza possibile.

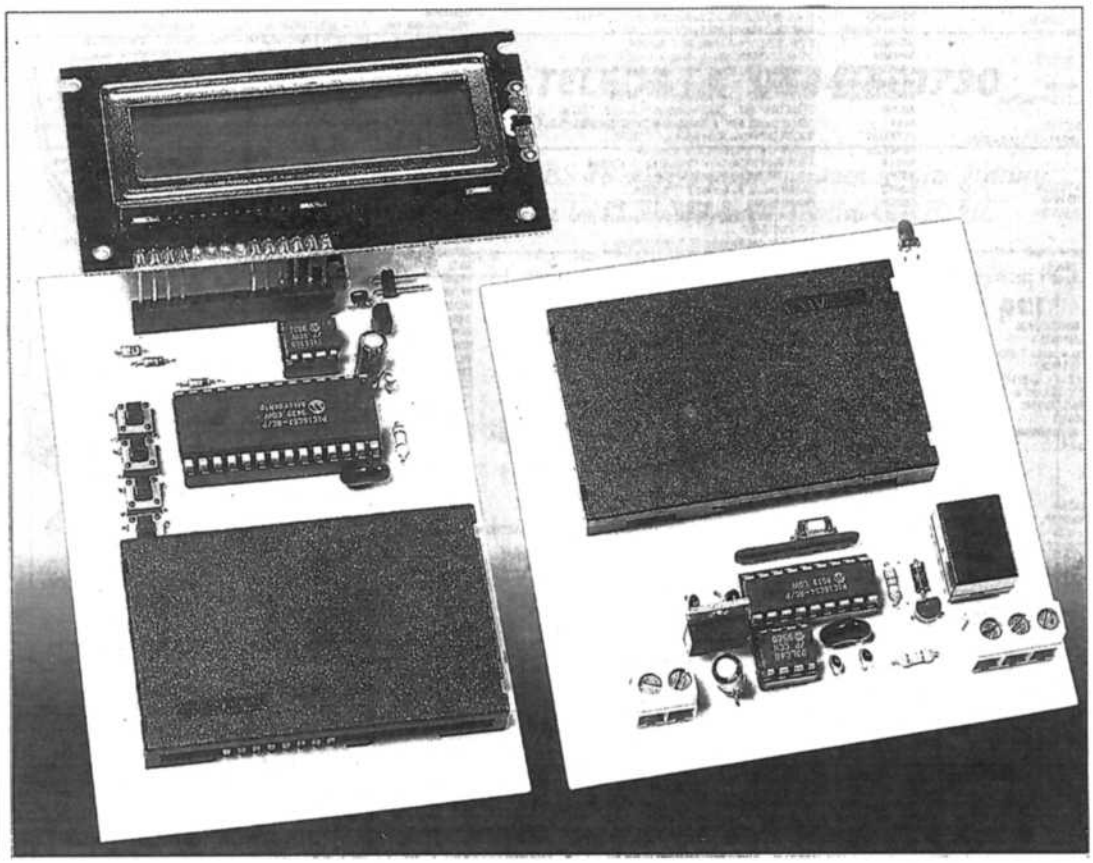
L'unico appunto che si potrebbe fare al nostro programmatore è quello di non essere completo, ovvero di non supportare come vedremo tutte le funzionalità della SLE4432 e parte della 4442.

In pratica, la protezione dei primi 32 byte contro la riscritta non è ammessa in entrambe le carte e così pure la scrittura della sola SLE4442.

Queste limitazioni sono dovute alla difficoltà di gestione software tra PIC, display ed utente: per facilitare le operazioni infatti, è necessario che l'interfaccia utente sia la più chiara e semplice possibile.

Ma per far ciò è implicito che deve corrispondere un grosso lavoro da parte del software di controllo, e questo non era giustificato per l'applicazione che dovevamo ottenere.

Come vedremo poi infatti, il nostro solo interesse è scrivere 6 byte della chip-card, a partire dall'indirizzo 132 decimale.



Tali byte costituiranno il codice della nostra chiave elettronica.

Per completezza però, il nostro programmatore consentirà di leggere interamente la memoria dati di una chip-card di tipo SLE4432 e SLE4442 (256 byte tra cui anche l'header), di memorizzare tale contenuto in una memoria "ausiliaria", di modificare i 256 byte dei dati (ad esclusione di quelli protetti tra i primi 32 byte) sempre della memoria ausiliaria e, successivamente, di riscriverli su una chip card di tipo SLE4432.

A conclusione di ciò sarà anche possibile eseguire una verifica dei dati scritti confrontandoli con quelli della memoria ausiliaria.

In più, anche togliendo alimentazione, il con-

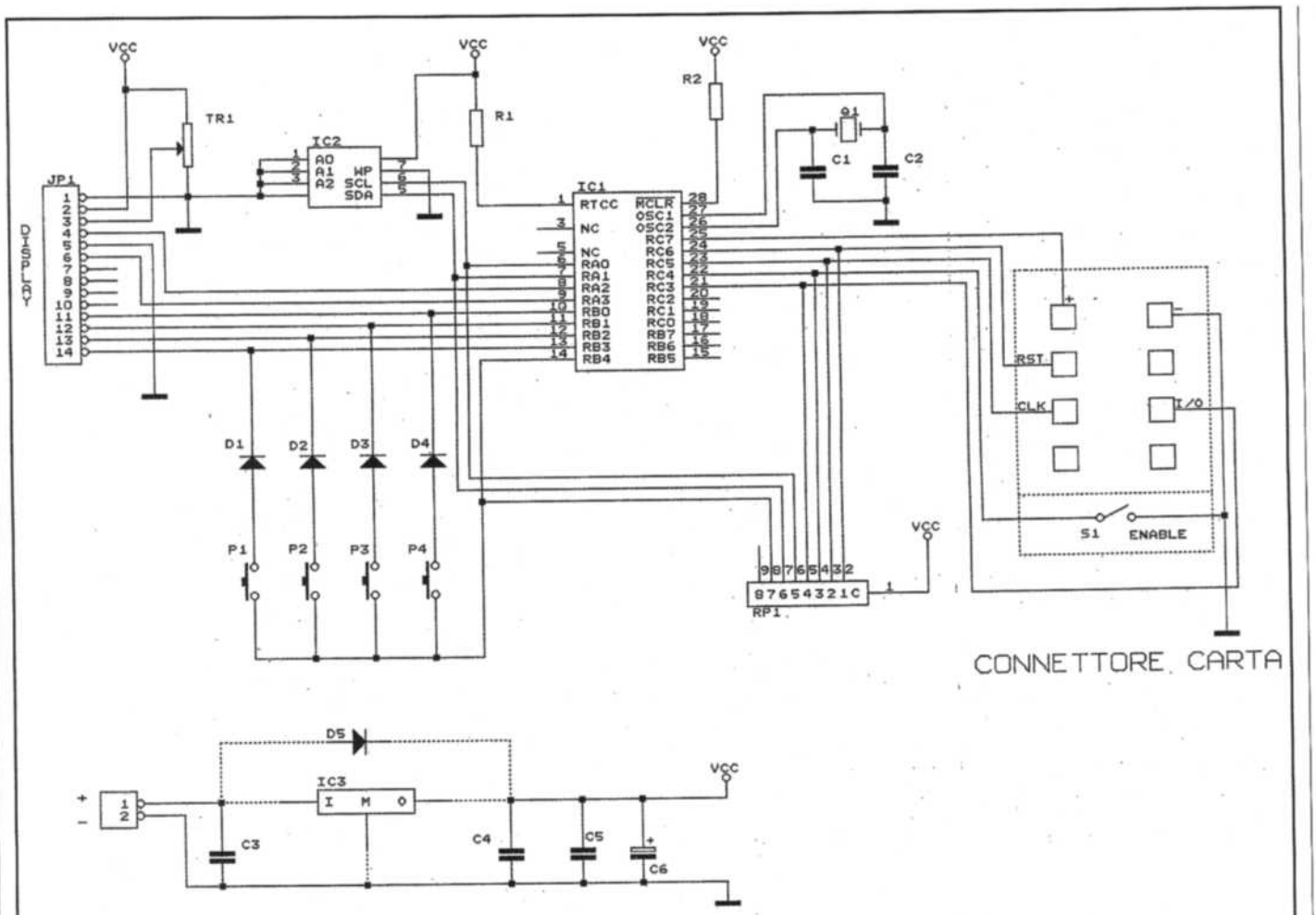


Figura 14. Schema elettrico del programmatore

tenuto della memoria ausiliaria non verrà perso (essendo la memoria in tecnologia EEPROM) e potremo programmare altre carte anche dopo mesi ed anni senza più doverne rileggere una o modificarne i dati.

Come vedete, quindi, il programmatore va al di là delle effettive richieste per i nostri scopi, consentendo una gestione quasi completa delle funzioni richieste per le carte impiegate.

Funziona così

Il circuito elettrico del programmatore è visibile in Figura 14. Il microcontroller IC1 gestisce il display alfanumerico, i quattro pulsanti, la memoria ausiliaria IC2, la carta stessa.

La sezione di alimentazione può essere implementata in due modi diversi: con un diodo (D5) oppure con un 78L05.

La prima soluzione è obbligatoria se si impiegano 4 pile da 1,5 volt cadauna (per un massimo quindi di 6 volt ed un minimo di 5 volt), mentre la seconda va bene per tensioni di alimentazioni comprese tra 8 e 18 volt.

In Figura 15 possiamo vedere un diagramma a blocchi in cui, per semplicità, abbiamo riportato quasi esclusivamente le frasi che vengono visualizzate sul display in funzione dei vari eventi.

Si comincia con la prima "schermata" (da ora in avanti le visualizzazioni sul display saranno indicate così) che visualizza la scritta "Card programmer" sulla prima riga e la scritta "OK 4432" sulla seconda.

In questa fase, l'unico pulsante che verrà riconosciuto sarà P1, ovvero quello corrispondente alla scritta "OK".

Dobbiamo far presente che i quattro pulsanti collocati in verticale, corrispondono alle quattro possibili scritte che possono apparire sulla seconda riga del display.

Premendo il pulsante P1 (OK) avremo sul display le scritte "funzione" sulla prima riga e "RD WR MD VF" sulla seconda.

A seconda di quale dei 4 pulsanti premeremo in questa fase, il software prenderà diverse strade.

Nell'ipotesi di premere il pulsante P3 (MD) che sta per MoDify, appariranno le due scritte "Id:xxx W:YY ZZ" e "Id b1 b2 Fi".

Le xxx relative al prefisso Id indicheranno l'indirizzo della prima delle due celle della memoria il cui valore è riportato in yy ed in zz.

Per far capire meglio, per scrivere il valore esadecimale 59h sull'indirizzo 185h della memoria, dovremo premere il pulsante P1 (Id) fino ad arrivare al numero xxx = 0b9 e poi modificare il secondo byte con il pulsante P3 (b2) il valore zz fino ad ottenere 59.

Si ricorda che la modifica della cella di memoria avverrà solamente sulla memoria ausiliaria del programmatore, quindi per trasferirla poi alla carta sarà necessaria un altro tipo di operazione.

Per terminare la fase delle modifiche, dovremo premere il pulsante P4

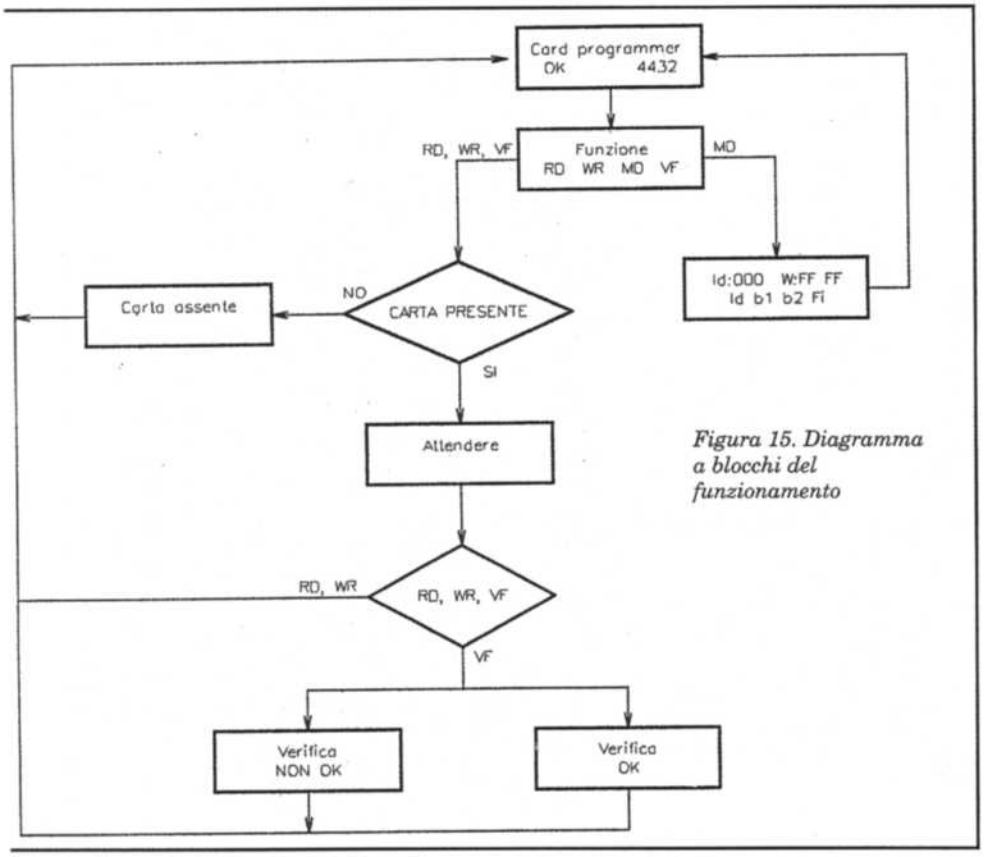


Figura 15. Diagramma a blocchi del funzionamento

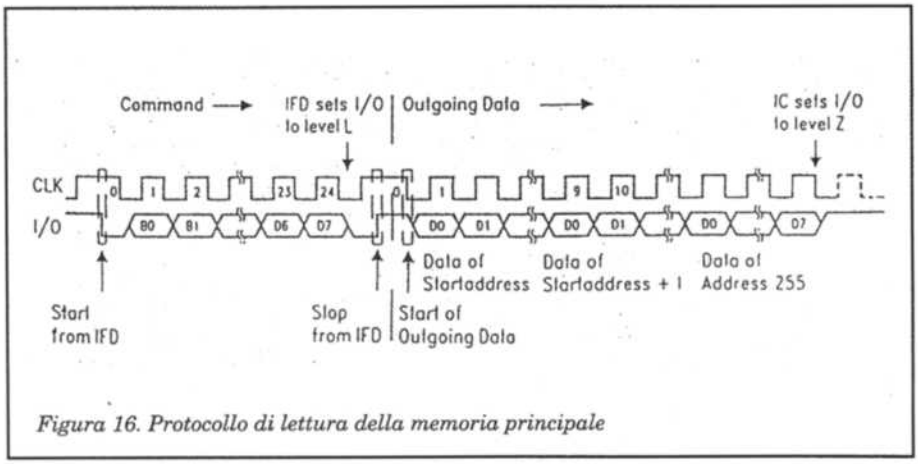


Figura 16. Protocollo di lettura della memoria principale

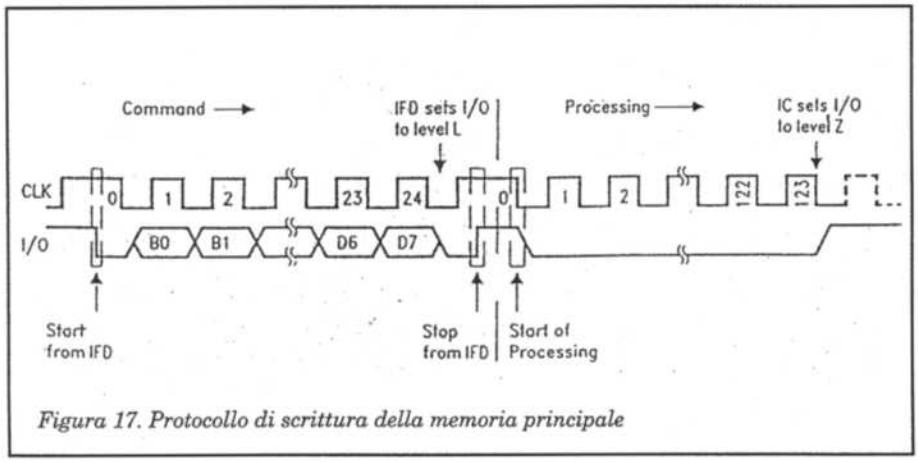


Figura 17. Protocollo di scrittura della memoria principale

(Fi) che sta per Fine modifiche, per poi tornare alla schermata principale.

Se invece, dopo la schermata iniziale, venisse scelta un'operazione diversa dalla modifica, allora il software andrebbe subito a verificare per prima cosa la presenza della carta nell'inseritore.

Se la carta risultasse assente, sul display apparirebbe la scritta "Carta assente" per circa 2 secondi, poi si tornerebbe alla schermata iniziale.

Nel caso in cui la carta venisse trovata, si passerebbe alla scritta "Attendere" per il tempo richiesto dall'operazione prescelta.

Al termine, se tale operazione era una scrittura (WR) oppure una lettura (RD) della carta, il programmatore tornerebbe alla schermata iniziale.

Quando invece avevamo scelto un'operazione di verifica con il pulsante P4 (VF), allora sul display apparirà o la scritta "Verifica " "OK" o la scritta "Verifica" "non OK" a seconda che la verifica abbia dato esito positivo o negativo.

Queste scritte rimarranno per circa due secondi, poi si passerà nuovamente alla schermata iniziale.

Sicuramente avrete compreso che questa volta, data la vastità del programma, non sarebbe stato possibile inserire un diagramma a blocchi più dettagliato.

Interfacciamento

Tornando all'interfacciamento con la carta, nelle Figure 16, 17 e 18 possiamo vedere quale sia il protocollo impiegato per il dialogo tra microcontroller e carta.

In Figura 16 abbiamo il protocollo necessario per leggere il contenuto di una cella di memoria: viene inviato un comando composto da tre byte (comando lettura memoria principale, indirizzo della cella, byte senza effetto) e poi si aspetta che la carta risponda all'interrogazione con un byte per cella.

È possibile anche una lettura di celle contigue senza inviare ogni volta il comando e l'indirizzo.

In Figura 17 invece è visibile il protocollo necessario per la scrittura di una cella: anche qui si inviano tre byte (comando di scrittura della memoria principale, indirizzo della cella, dato da scrivere) e poi si aspetta il tempo che la EEPROM interna alla carta abbia terminato il suo lavoro (mediamente 123 o 255 cicli di clock).

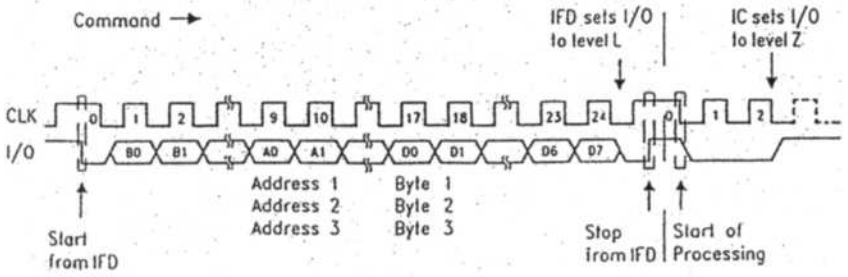


Figura 18. Protocollo di comparazione del PIN e dell'EC

Infine, in Figura 18 abbiamo il protocollo necessario alla verifica del PIN di accesso per la famiglia SLE4442.

Montaggio

Per realizzare il programmatore, è preferibile impiegare un circuito stampato tipo quello proposto in Figura 19. Durante la fase di inserzione dei componenti, fate attenzione a quelli polarizzati, e non dimenticatevi di inserire i componenti che stanno al di sotto di IC1, altrimenti il circuito non funzionerà. Per quanto riguarda il display, potrete montare uno zoccolo per non essere poi legati al tipo di contenitore che vorrete utilizzare.

Allo stesso modo, potrete non inserire anche i 4 pulsanti. L'oscillatore Q1 potrà essere sia di tipo ceramico da 3,58 MHz, sia di tipo quarzato da 3,579545 MHz.

La memoria IC2 invece potrete dimensionarla come vorrete, partendo dalla 2402 fino alla 2416, in quanto il protocollo di lettura/scrittura è il medesimo e varia soltanto la capacità totale.

Anche per IC3 o D5 vale lo stesso ragionamento fatto all'inizio.

Collaudo

Per collaudare il programmatore, occorre fornire alimentazione e verificare che appaia la schermata iniziale descritta in Figura 15.

Se ciò non accade, quasi sicuramente vi siete dimenticati di inserire C1, C2 oppure il quarzo Q1.

Poi dovremo procurarci una carta tipo SLE4432 o vergine o già programmata e, premuto il pulsante P1, ci troveremo nella seconda schermata di Figura 15. A questo punto l'operazione più logica sarà quella di leggere il contenuto della carta, dando il comando di lettura (RD) con P1.

Dopo la scritta "Attendere", la carta sarà stata letta, ovvero il suo contenuto

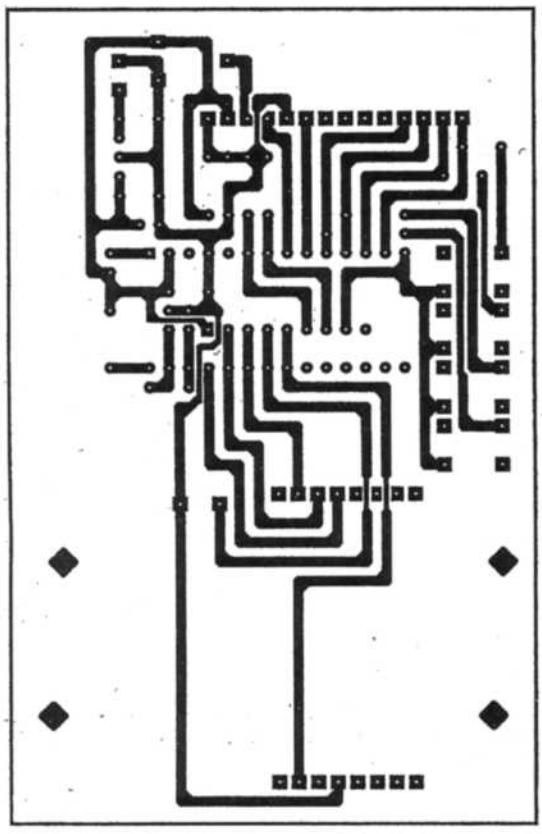


Figura 19. Circuito stampato, scala 1:1

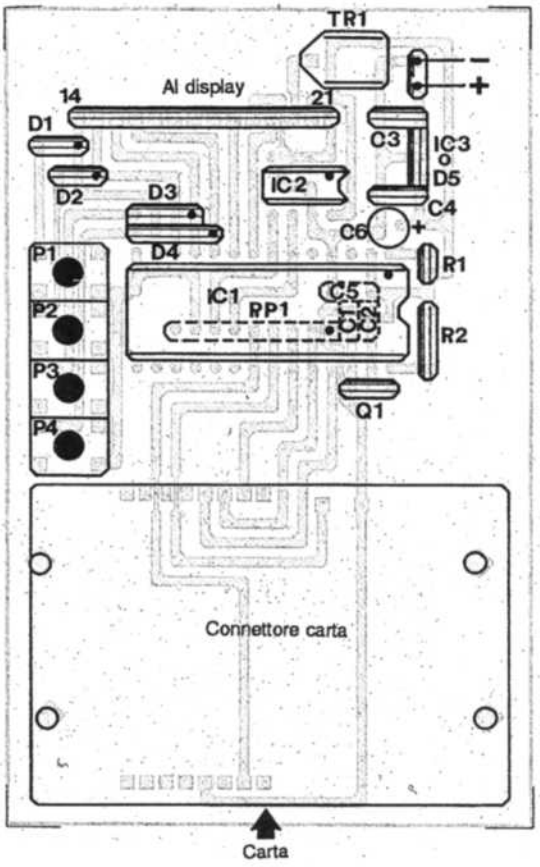


Figura 20. Disposizione dei componenti

ELENCO COMPONENTI

Semiconduttori
IC1: PIC16C57XT programmato (0347/2643514)
IC2: 2402 o 2404 o 2408 o 2416
IC3: 78L05
D1, D2, D3, D4: 1N4148
D5: 1N4001

Resistori
R1, R2: 10 kΩ
RP1: Rete resistiva 10 kΩ
TR1: Trimmer 4,7 kΩ

Condensatori
C1, C2: 10 pF
C3, C4, C5: 100 nF
C6: 22 uF

Varie
Q1: Oscillatore o quarzo da 3,579545 MHz
Display alfanumerico 16x2
Connettore: codice RS 453-785

sarà stato inserito nella memoria IC2. A questo punto, se volessimo programmare la carta per la chiave proposta lo scorso mese, dovremmo entrare nella schermata di modifica ed arrivare all'indirizzo Id:084.

Qui dovranno essere impostati i valori dei primi due byte del codice, poi si passerà all'indirizzo Id:086 e si imposteranno i valori del terzo e quarto byte del codice ed infine si passerà all'indirizzo Id:088 per impostare il quinto e sesto byte del codice.

Al termine si premerà il pulsante P4 (Fi) e le modifiche saranno state memorizzate in IC2.

Per trasferirle alla carta, sarà sufficiente eseguire poi un'operazione di write (WR) corrispondente al pulsante P2 della seconda schermata.

Per essere poi sicuri della corretta memorizzazione, sarà possibile anche eseguire l'operazione di verifica (VF) con il pulsante P4, sempre dalla seconda schermata.

Se poi si volesse programmare altre N carte con lo stesso codice, sarebbe sufficiente ripetere le sole operazioni di write e di verifica per ogni carta.

Si ringrazia la Veron Spa - Via Caldera n. 21 - Milano per la collaborazione data nell'acquisizione delle informazioni citate. La Veron Spa è disponibile telefonicamente al numero 02/482151 per chiarimenti commerciali. ■