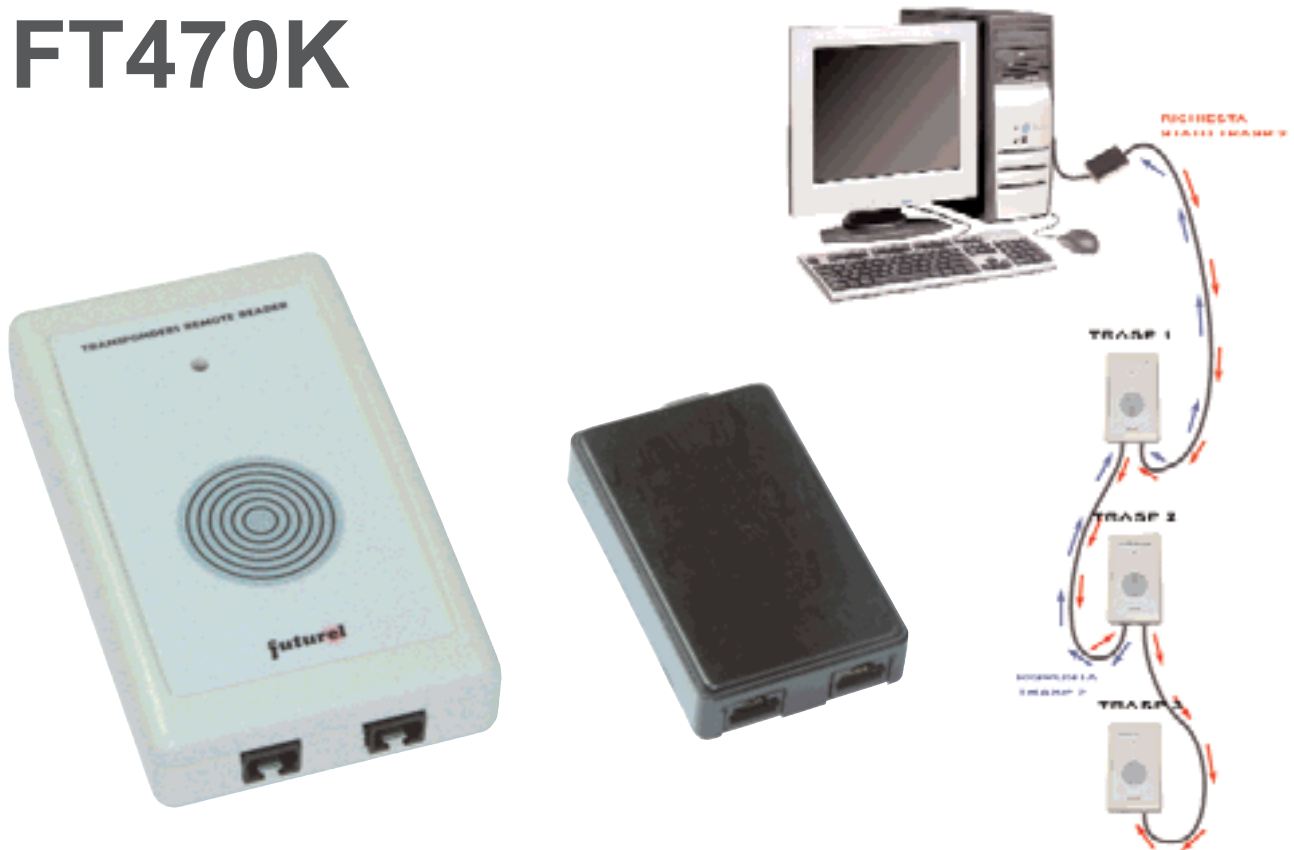


LETTORI DI TRANSPONDER COMANDATI DA UN PC

FT470K

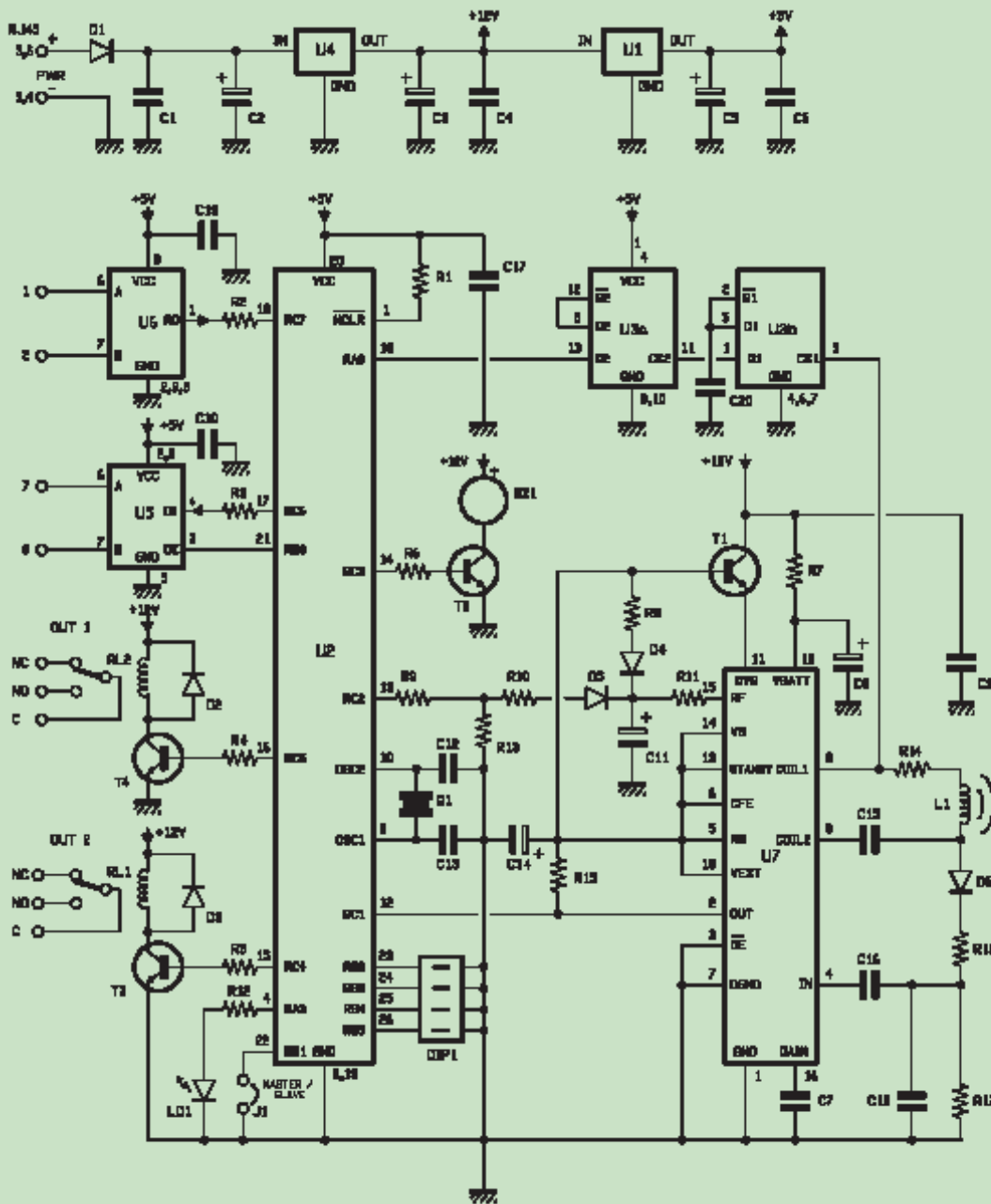


Da parecchi anni i transponder sono divenuti uno dei principali metodi di identificazione e controllo accessi: se ne trova sotto forma di portachiavi, badge a formato ISO7816 (tipo la carta di credito) e oggetti di varia natura. Nelle aziende sostituiscono il tradizionale cartellino orario, grazie a sistemi analoghi a quello da noi proposto nel fascicolo di giugno 2002, mentre in club o locali riservati, palestre, cinema e parcheggi, sono un efficace metodo di selezione e a volte consentono anche il pagamento automatizzato del servizio o l'utilizzo di un qualcosa di prepagato (ad esempio gli erogatori automatici con credito a

scalare). Nelle pagine di Elettronica In abbiamo più volte proposto e descritto dispositivi fatti per leggere i transponder ed elaborarne i dati, sia in maniera autonoma che con l'appoggio di un Personal Computer: si è però sempre trattato di singoli lettori, fatti per controllare un unico punto d'accesso. Ma nella pratica vi sono situazioni in cui occorre verificare chi entra o chi chiede un servizio da più punti, ed occorrerebbe un lettore multiplo: qualcosa come il progetto descritto in questo articolo, che è sostanzialmente un gruppo di lettori di transponder capaci ciascuno di lavorare sia in modo stand-alone, sia inter-

facciato ad un Personal Computer mediante un'interfaccia di comunicazione RS485. Proprio questo tipo di bus consente di gestire, da un solo PC e con un apposito programma scritto in Delphi, fino a 16 lettori collegati tutti sulla stessa linea. La gestione computerizzata estende le possibilità d'uso e la versatilità del sistema perché consente, per ogni lettore, di creare una lista di transponder abilitati (preventivamente appresi dal PC mediante una procedura - autoapprendimento - che richiede il passaggio davanti al sensore del lettore stesso) quindi di assegnare a ciascuno azioni locali quali l'attivazione di uno o più relè

SCHEMA ELETTRICO SCHEDA LETTORE TRANSPONDER



in modo sia impulsivo che bistabile. Queste ed altre funzioni appariranno più chiare dopo aver visto lo schema elettrico di un lettore; prima di analizzarlo è però opportuno premettere che il sistema è modulare e consta, nella sua forma elementare, di un lettore e di un'interfaccia di collegamento al Personal Computer. Quest'ultima è unica per tutti i letto-

ri di transponder eventualmente aggiunti, giacché si tratta di un semplice converter di livelli logici da RS485 a RS232; siccome l'RS485 è un bus, ad esso potranno essere collegati più lettori di transponder (fino ai 16 consentiti dal software di gestione) tutti in parallelo tra loro. Ecco perché basta una sola interfaccia.

FUNZIONAMENTO GENERALE DEL SISTEMA

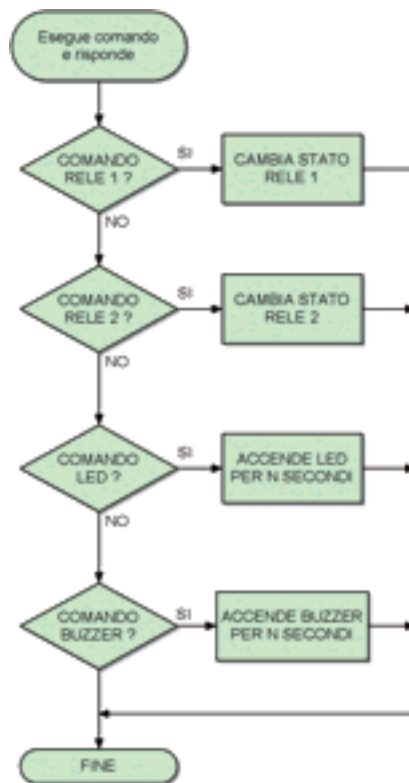
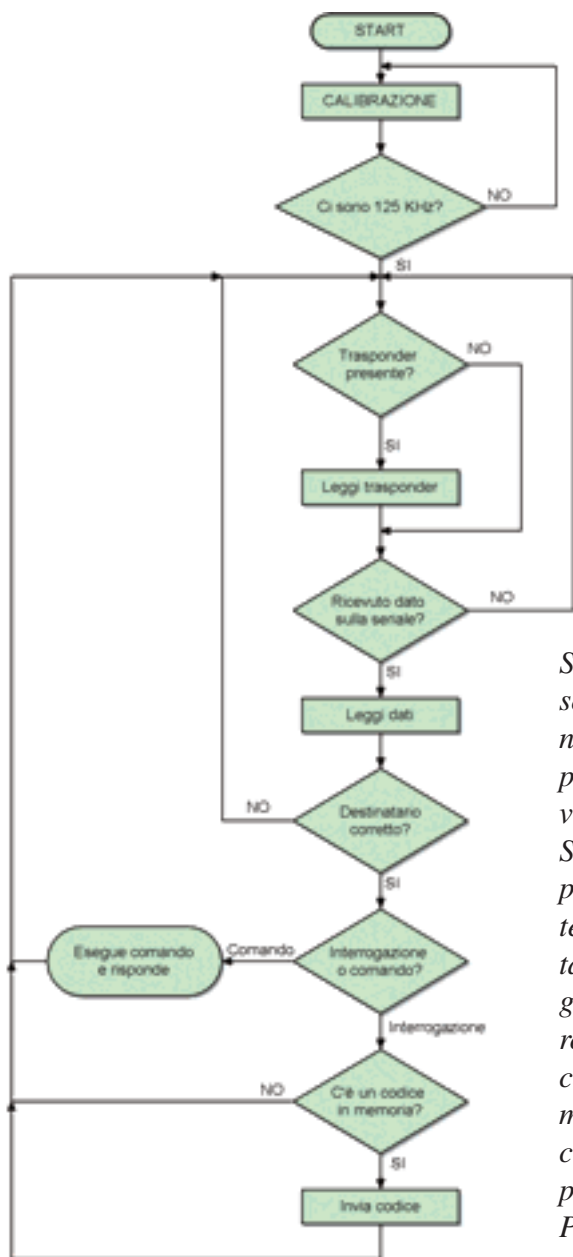
Prima di iniziare l'analisi dei circuiti elettronici, diamo una descrizione di più ampio raggio del progetto generale. Come detto il sistema risulta composto da un PC su cui è installato un software di gestione, da un massimo di 16

diversi lettori di transponder e da un'interfaccia di comunicazione con il compito di collegare il computer verso i lettori transponder. La logica di funzionamento può essere riassunta nel seguente modo: quan-

do viene passata una scheda vicino a un lettore, il codice identificativo della stessa viene letto e quindi trasmesso al PC utilizzando il bus condiviso. Il software presente sul PC, ricevuto tale codice, lo con-

fronta con quelli prememorizzati ed eventualmente comanda al lettore l'esecuzione di alcune azioni. Come si vede tutta la logica di confronto e l'invio dei comandi viene svolta dal PC; i lettori servono solo

FLOWCHART SCHEMA LETTORE DI TRANSPONDER



Schema Flowchart di massima del software presente nel PIC del lettore di transponder. Dopo l'inizializzazione e la calibrazione, viene testata la presenza di un transponder. Se questo è presente, viene letto e memorizzato il codice della tessera. Successivamente si passa a verificare se sono disponibili dati provenienti dal PC, che eventualmente vengono letti. Se il dispositivo non è il destinatario dei dati, si ritorna all'inizio del ciclo di gestione; altrimenti si testa se è presente un'interrogazione o un comando. Nel primo caso si verifica se è presente almeno un codice di una tessera in memoria, ed eventualmente si trasmette tale codice al PC. Se invece è stato ricevuto un comando, si passa all'esecuzione dello stesso: in questo caso il PIC verifica di che tipo di comando si tratta ed eventualmente agisce sui relè.

per identificare i codici scritti nei transponder e eseguire i comandi che provengono dal PC. La gestione della comunicazione avviene in polling: è quindi il PC che sequenzialmente interroga i lettori ed eventualmente gestisce la trasmissione e ricezione dei dati/comandi.

SCHEMA ELETTRICO DEL LETTORE DI TRANSPONDER

Il circuito di ogni lettore può essere suddiviso in 5 blocchi principali: un blocco composto dall'integrato U2270 (chip U7) che gestisce la lettura dei codici dei transponder; un blocco composto dal PIC16F876 (chip U2) che gestisce tutta la logica del dispositivo; un blocco che esegue la gestione di eventuali periferiche esterne, composta dai due relè RL1 e RL2; un blocco che gestisce la comunicazione sul bus RS485, realizzato dai chip U5 e U6 e infine il blocco di alimentazione, composta dai chip U1 e U4.

Il primo blocco è basato essenzialmente sul chip U2270 prodotto dalla Temic, ossia un componente specifico per la realizzazione di lettori di transponder passivi. Questo provvede a generare un campo elettromagnetico a 125 KHz (mediante un oscillatore interno basato su un VCO) e ad irradiarlo, grazie alla bobina L1, nell'ambiente circostante, rivelando poi parte del segnale presente tra il condensatore C15 e la L1 stessa, il quale, come scoprirete tra breve, risente della presenza dei transponder. A riposo, ai capi di C18 è localizzata una tensione continua ricavata dal raddrizzamento, ad opera del diodo D6, dell'onda sinusoidale applicata alla bobina dai piedini 8 e 9 dell'U2270. Se viene avvicinata una scheda transponder, ad una distanza tale da determinare un assorbimento significativo nel circuito a 125 KHz, per

effetto della reazione di indotto viene causata una variazione della corrente che attraversa L1 (dovuta alla commutazione della logica interna al transponder) la quale determina anche un cambiamento dell'ampiezza della tensione applicata tra l'anodo di D6 e massa. Ai capi di C18 si trova quindi un'onda rettangolare a bassa frequenza (poche centinaia di Hz). Questo segnale dipende dal fatto che il transponder, investito dalle linee di flusso del campo elettromagnetico, si attiva e trasmette il proprio codice identificativo. Il segnale che trasporta il codice viene quindi applicato al piedino di ingresso IN (4) tramite il condensatore di accoppiamento C16; all'interno di U7 un amplificatore ed uno squadratore provvedono ad estrarre gli impulsi e a raddrizzarne i fronti di salita e discesa.

Alla fine, in uscita dal piedino 2 di U7, si ottiene un segnale che riproduce il codice identificativo della scheda transponder. Questo segnale viene quindi inviato alla porta di ingresso RC1 del microcontrollore (chip U2). Quest'ultimo è un PIC16F876 al quale è demandato il compito di gestire interamente il lettore, sia per quel che riguarda l'acquisizione dei dati relativi al passaggio dei singoli transponder, sia il dialogo con il computer tramite il bus RS485 (gestito dalle porte RB0, RC6 e RC7 e dai chip U5 e U6), sia l'elaborazione dei comandi in arrivo da esso e riguardanti i relè (RL1 e RL2 comandati dalle porte RC4 e RC5), ed altro ancora.

Dopo l'inizializzazione degli I/O, il main program che gira nel PIC (riferitevi al flowchart) verifica continuamente le condizioni logiche del piedino 12 e del 18: dal primo attende i dati estratti dall'U2270, mentre con il secondo controlla l'arrivo di un'eventuale richiesta di comunicazione da parte

del computer; a riguardo abbiamo già precisato che il sistema funziona in polling, ovvero è il PC che interroga le varie periferiche e non i lettori a trasmettere automaticamente i propri dati. Ciò è logico in quanto, affacciandosi su di un singolo bus, i diversi dispositivi potrebbero generare delle collisioni se trasmettessero contemporaneamente.

Va osservato che all'inizializzazione il programma setta la porta UART interna che, nel nostro caso, provvede a gestire la comunicazione seriale tramite i piedini 17 e 18. L'UART consente una velocità di ritrasmissione di ben 115200 Kbps, quindi più che adatta alle più veloci porte seriali implementate nei Personal Computer. La scelta di affidare la comunicazione ad una specifica periferica piuttosto che ricorrere ad un registro qualsiasi è stata dettata dalla necessità di sgravare di questo compito il microcontrollore, a beneficio della rapidità di risposta.

Quando il micro rileva una commutazione sul pin 12, acquisisce la rispettiva lettura, poi ne verifica il formato ed il checksum per essere certo di aver compiuto un'acquisizione corretta. Riguardo al formato dei dati è utile notare che i transponder da noi previsti inviano 64 bit, dei quali i primi 9 sono un codice di sincronismo (start) per indicare al dispositivo di lettura (il microcontrollore U2) che deve procedere all'acquisizione, 40 sono i dati veri e propri (organizzati in 5 righe x 4 colonne) 10 servono per la parità di riga e 4 per quella di colonna. Se parità di riga e di colonna sono conformi con il checksum ricevuto, il micro memorizza i rispettivi dati in RAM, altrimenti ignora ciò che proviene dall'U2270 ed attende un nuovo segnale di start.

Le informazioni derivanti dalla lettura di un transponder vengono tra-

COLLEGAMENTO DEI LETTORI DI TRANSPONDER

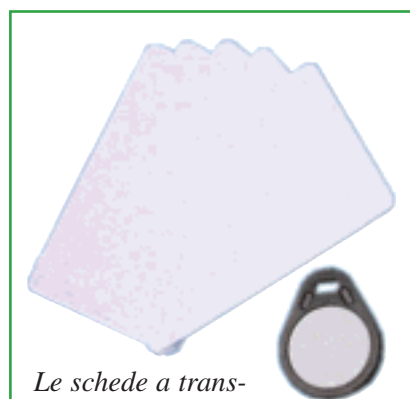


Il collegamento tra il PC e l'interfaccia di comunicazione avviene tramite una porta in formato RS232. La connessione tra l'interfaccia e i diversi 16 lettori di transponder è invece realizzata tramite un bus RS485. Ogni lettore è munito di 2 porte RJ45 che realizzano il bus; le due porte sono collegate tra loro in parallelo. Ciò significa che tutti i dati che arrivano su una porta, vengono trasmessi anche sulla seconda. In questo modo le informazioni trasmesse dal PC vengono inviate all'interfaccia, la quale le converte in formato RS485 e quindi le invia sul bus. I dati giungono solo al lettore di transponder identificato dall'indirizzo specificato, il quale elabora le informazioni e risponde al PC specificando il proprio indirizzo per farsi identificare.



smesse al PC e quindi confrontate con quelle salvate precedentemente nel disco rigido del computer, tramite l'apposita procedura di autoapprendimento: il confronto può avere esito positivo o negativo. Nel primo caso il computer comanda al micro di eseguire alcune azioni locali. Se invece i dati non corrispondono ad alcuna delle stringhe precedentemente memorizzate, la procedura viene abbandonata. Per azioni locali si intende l'attivazione o il cambiamento di stato di uno o di entrambi i relè presenti nella scheda del lettore di transponder. Queste azioni vengono decise dall'apposita finestra di dialogo del programma di gestione, e quindi comunicate al lettore mediante il bus RS485. Ad esempio, possiamo

far sì che a seguito della lettura di un certo transponder scatti il solo RL2, che potrebbe comandare l'apertura di un'elettroserratura; ancora, si può decidere che debbano scattare entrambi i relè, magari con modalità diverse. Siccome il sistema lavora in polling, tutte le azioni vengono attuate solo dopo che la scheda viene interrogata dal computer; ciò vuol dire che la lettura di un transponder abilitato non sempre determina azioni immediate; a volte, prima che RL1 o RL2 si attivino può passare qualche istante. Dal momento della lettura all'interrogazione operata dal PC, i rispettivi dati restano fermi in un buffer del PIC riservato alla comunicazione; quando il computer lo richiede, tali informazioni tran-



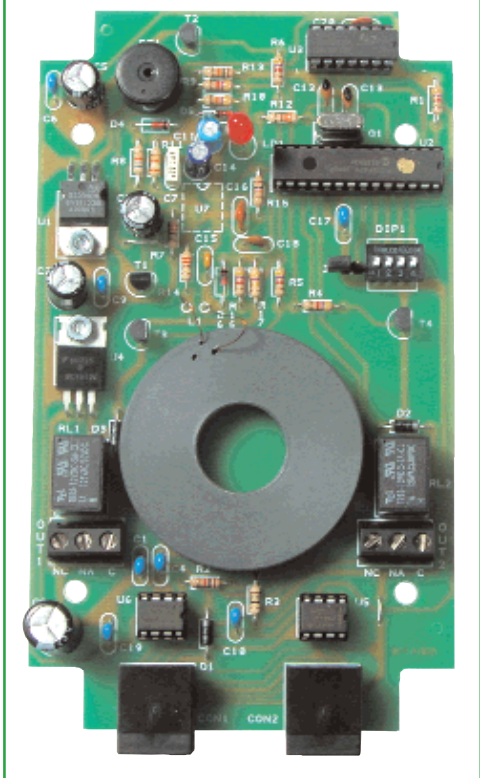
Le schede a transponder sono disponibili nei formati badge ISO7816 (tipo carta di credito) o nel più comodo portachiavi.

sitano attraverso il bus di trasmissione, ossia dal piedino 17 del micro. L'acquisizione è comunque segnalata localmente dall'avvisato-



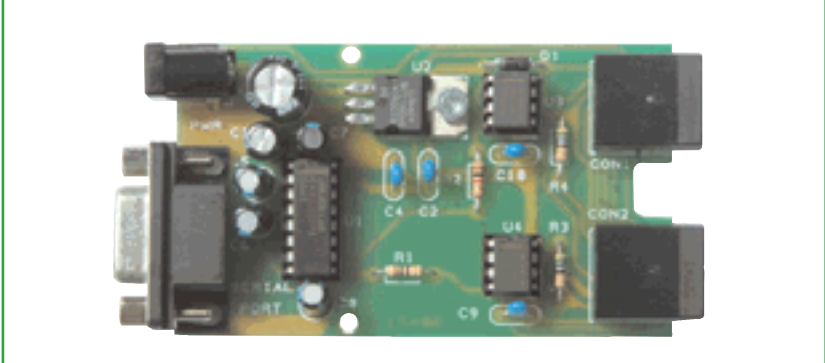
L'interfaccia di comunicazione dispone da un lato della porta in formato RJ45 utilizzata per collegare i lettori transponder; dall'altro lato è invece disponibile una porta seriale in formato RS232 con la quale si esegue il collegamento a un PC. È inoltre disponibile un jack per fornire l'alimentazione di circa 16V.

LETTORE TRANSPONDER



e C11, così da ottenere un potenziale che viene sovrapposto a quello normalmente presente sul pin 15 (RF) dell'U2270. Questa forma di retroazione agisce sul VCO facendo in modo che, se la frequenza tende a salire oltre la tolleranza imposta sopra i 125 KHz il micro riduce il

INTERFACCIA DI COMUNICAZIONE



duty-cycle del segnale PWM prodotto dalla propria linea RC2, facendo così diminuire la tensione applicata al piedino 15 dell'U2270; viceversa, se la frequenza cala eccessivamente, il software del PIC fa aumentare la larghezza degli impulsi PWM e quindi accresce il potenziale che controlla il VCO del chip Temic, intervenendo sul solito piedino 15 e costringendo l'oscillatore controllato a sintetizzare una frequenza maggiore. La compensazione del VCO appena descritta non è costante ma periodica: viene esercitata esclusivamente dopo l'accensione e l'inizializzazione degli I/O e a seguito di ogni lettura operata dall'U2270, per

ricalibrare il sistema. Bene, ora che abbiamo spiegato come avviene la lettura dei transponder e la gestione dei dati in attesa del dialogo con il computer, è il caso di affrontare due dettagli molto importanti: il jumper J1 e la serie di dip-switch DIP1. Il primo, ossia la condizione logica assunta dalla linea RB1 del microcontrollore, consente di optare tra due modalità di funzionamento ben definite: con J1 aperto il lettore acquisisce i dati dei transponder, li passa al computer durante la successiva interrogazione (segnalata all'utente esterno con un lampeggio del led) ed attende istruzioni sull'attivazione di eventuali attuatori locali. Invece se il ponticello viene chiuso

si imposta il funzionamento automatico: in questo caso il led lampeggia a intervalli regolari proprio per indicare questa modalità. Quando una scheda a transponder viene passata, il led rimane acceso per alcuni secondi e viene emesso un beep di conferma; inoltre il codice letto viene trasmesso al PC senza attendere l'interrogazione da parte di quest'ultimo. In questa particolare modalità è possibile acquisire le informazioni sulla lettura mediante un qualsiasi emulatore di terminale: ad esempio, con Hyper Terminal di Windows appare una stringa con il codice del transponder letto e l'ID del lettore. Bisogna però notare che se vengono collegati più lettori su una stessa linea, non essendoci l'interrogazione del PC, si potrebbero verificare dei conflitti se più transponder trasmettessero contemporaneamente. Questa modalità può, per esempio, essere utilizzata per eseguire un test veloce sul funzionamento del dispositivo, o nel caso prevediate di utilizzare un numero limitato di lettori sul singolo bus.

Analizziamo a questo punto la presenza di DIP1: abbiamo detto che il software di gestione da PC può controllare fino a 16 lettori; per farlo deve naturalmente poterli distinguere l'uno dall'altro. Ebbene, il riconoscimento avviene proprio in base a un numero identificativo trasmesso ad ogni interrogazione e ricavato sulla base dell'impostazione delle linee RB2÷RB5 del micro U2. Resta inteso che, in sistemi composti da più periferiche di lettura, il DIP1 deve essere disposto diversamente da una scheda all'altra.

Diamo ora uno sguardo a U5 e U6, due integrati prodotti dalla Maxim con la sigla MAX485: ciascuno di essi è un transceiver a standard RS485, ossia un chip contenente un convertitore TTL/RS485 ed uno RS485/TTL, escludibili a vicenda. Nel nostro caso ne usiamo due per

le motivazioni che saranno espone più avanti nel testo, quando descriveremo l'unità di interfaccia. Per il momento ci basta dire che U5 è impostato per funzionare come solo trasmettitore, ovvero da converter TTL/RS485, mentre U6 è configurato in modo da restare sempre in ricezione e usare la sola sezione convertitrice da RS485 a TTL.

Completiamo l'analisi dello schema con la sezione di alimentazione, che riceve la tensione principale dall'interfaccia di comunicazione mediante lo stesso connettore RJ45 che contiene il canale dati; con esso preleva (vedere lo schema dell'interfaccia) la componente continua in arrivo dal plug d'ingresso che, filtrata da C1 e C2, viene applicata al regolatore U4. Quest'ultimo è un 7812 che serve a ricavare +12V perfettamente stabilizzati necessari al funzionamento del cicalino BZ, dei lettori di transponder e delle bobine dei due relè. I +12V vengono successivamente regolati a +5V da U1 (integrato 7805), con i quali vengono alimentati tutti i dispositivi TTL.

È da notare che ogni scheda monta due connettori RJ45, collegati tra loro in parallelo: ciò è stato realizzato proprio per connettere più lettori in serie. In sistemi multipli basterà collegare con un apposito cavetto un connettore del primo lettore all'RJ45 dell'interfaccia, quindi connettere, con un secondo cavo, l'altro RJ45 del lettore ad uno qualsiasi di quelli del secondo lettore, e così via fino al sedicesimo. La connessione, apparentemente in cascata, è in realtà in parallelo, giacché l'alimentazione ed il bus di ricezione RS485 sono multiplati.

SCHEMA ELETTRICO DELL'INTERFACCIA DI COMUNICAZIONE

Fondamentale importanza riveste il modulo di comunicazione che sostanzialmente provvede alla

conversione dei segnali da standard RS485 usati dal bus dei lettori di transponder in quelli RS232 adottati dalle porte seriali dei PC. Inoltre, il circuito riceve dal plug (PWR) la tensione continua di alimentazione (almeno 16÷17 volt): questa viene portata (tramite i pin 5 e 6 del connettore RJ45) ai moduli a transponder; viene realizzata anche la necessaria connessione di massa (contatti 3 e 4). La tensione di alimentazione viene anche stabilizzata e ridotta a +5V dal regolatore U2, che consente di far funzionare gli integrati TTL.

Tramite lo stesso RJ45, l'unità di interfaccia connette le linee di trasmissione (pin 1 e 2) e ricezione (pin 7 ed 8) del bus RS485 con le schede a transponder.

Il modulo ospita due converter TTL/RS485, usati uno come ricevitore (U3) e l'altro come trasmettitore (U4); questa strana configurazione è dovuta essenzialmente al fatto che vogliamo una comunicazione full-duplex immediata. Infatti gli integrati MAX485 sono effettivamente dei transceiver, dunque contengono ciascuno una sezione ricevente (RS485/TTL) e una trasmittente (TTL/RS485) e per realizzare un link ricetrasmittente potrebbe bastarne uno. Tuttavia per accelerare la comunicazione, rallentata dai limiti degli UART montati nei Personal Computer abbiamo pensato di sdoppiare le linee, riservandone una alla trasmissione e l'altra alla ricezione dei dati. Usando due MAX485, U3 è sempre posto in ricezione, quindi di esso lavora la sola sezione che converte i livelli di corrente RS485 in impulsi TTL da inviare al computer (previa conversione in RS232), mentre in U4 opera sempre e solo la parte trasmittente, che converte i livelli TTL ricevuti dal PC (previa conversione RS232/TTL), in RS485. La linea di trasmissione del bus è dunque il

doppino che esce dai contatti 1 e 2 del connettore RJ45, mentre quella di ricezione parte da 7 e 8.

Come accennato, il modulo d'interfaccia opera una conversione dagli impulsi di corrente della

linea RS485 in impulsi di tensione RS232 e viceversa; l'operazione non è però diretta. Per l'esattezza il blocco U1 (integrato MAX232) trasforma i livelli da RS232 (presenti ai pin 7 e 8) a TTL (forniti ai

pin 9 e 10) e viceversa; mentre i blocchi U3 e U4, oltre a eseguire la trasmissione/ricezione dei dati realizzano anche la conversione dei livelli da TTL (pin 1 e 4) a RS485 (pin 6 e 7).

PROTOCOLLO DI COMUNICAZIONE

Come abbiamo visto all'interno dell'articolo, per eseguire la comunicazione tra il PC e i diversi lettori di transponder viene utilizzato un protocollo.

Questo è composto dai comandi inviati dal computer e dalle risposte inviate invece dai lettori di transponder. Abbiamo già fatto notare che i comandi inviati dal PC iniziano tutti con i simboli “#”, mentre le risposte dei lettori iniziano tutte con “ST”. Successivamente sono presenti due caratteri che identificano il lettore di transponder (indicati con “d u” che rappresentano le decine e le unità dell'indirizzo del lettore).*

Successivamente vengono trasmessi alcuni caratteri che identificano il tipo di comando o di risposta; infine seguono i caratteri di fine pacchetto, che sono 90h per pacchetti inviati dal PC, e “EN” per le risposte inviate dai lettori.

Vediamo ora nel dettaglio quali sono i diversi comandi e risposte che è possibile trasmettere e come questi vengono identificati.

Il primo comando che il PC può trasmettere è l'interrogazione in polling; questo è identificato dai caratteri “01”. Il pacchetto inviato risulta quindi:

***# d u 01 90**

in cui i caratteri 01 indicano appunto l'interrogazione.

A questo punto il lettore può rispondere con il pacchetto:

ST d u 01 EN

per indicare che non ha codici di transponder memorizzati da inviare (i caratteri 01 indicano proprio che non sono presenti dati); se invece tali informazioni sono presenti il transponder risponde con il pacchetto:

ST d u 02 <10 byte che identificano il codice del

transponder letto> EN

in cui i caratteri 02 identificano l'invio del codice letto.

I codici dei transponder vengono trasmessi utilizzando 10 caratteri (a scanso di equivoci, sottolineiamo che i caratteri “<” e “>” non vengono trasmessi nel pacchetto; sono stati inseriti solo per una questione grafica).

Le operazioni che il PC può richiedere di eseguire al lettore di transponder sono l'attivazione o il cambiamento di stato dei relè: il pacchetto che il computer invia è composto da:

***# d u 02 01 90**

in cui i caratteri 02 identificano il comando di attivazione (o cambiamento di stato) del relè; mentre i caratteri 01 identificano il relè numero 1. Se si volesse invece agire sul secondo relè bisognerebbe trasmettere:

***# d u 02 02 90.**

A questo punto il lettore di transponder conferma la ricezione e l'esecuzione del comando inviando il pacchetto:

ST d u 03 01 EN

in cui i caratteri 03 identificano il tipo di risposta, mentre 01 identificano il numero del relè.

Un'ultimo pacchetto riguarda il settaggio dei tempi di attivazione dei due relè. Il PC trasmettendo:

***# d u 03 01 td tu 90**

indica al lettore identificato dai caratteri d u di impostare un tempo di attivazione per il primo relè (caratteri 01) pari a td tu.

Il meccanismo con cui si indicano i secondi è lo stesso dell'indirizzo del lettore: td indica le decine, mentre tu indica le unità.

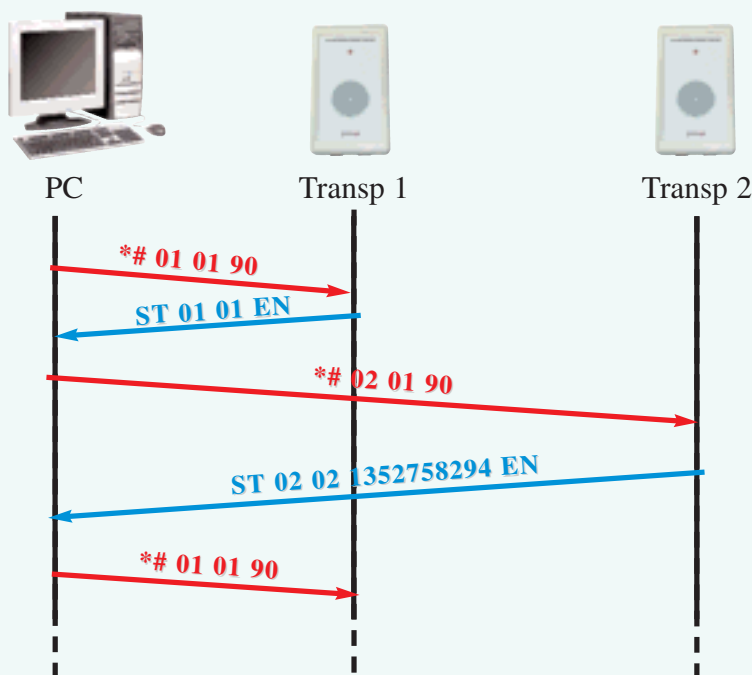
Il tempo di attivazione può quindi essere specifi-

cato nell'intervallo 1÷99 secondi; se invece si trasmette *tdtu=00* si specifica la modalità bistabile.

La risposta del lettore a questo comando risulta:

ST d u 03 01 td tu EN

in cui, ancora una volta, i caratteri **03** identificano il tipo di risposta; **01** indica il numero del relè e **td tu** indicano il tempo di attivazione che è stato impostato. Una volta noto il protocollo di comunicazione è quindi possibile realizzare un opportuno software che, basandosi proprio su questo, gestisce i diversi lettori di transponder e regola quindi l'attivazione dei due relè a seconda di quale utente è stato riconosciuto.



Esempio di interrogazione in polling di 2 lettori di transponder.

Il PC interroga il primo lettore, che risponde che non ha dati.

Il PC passa allora a interrogare il secondo lettore, che invece risponde che ha dei dati e li trasmette. A questo punto il ciclo potrebbe continuare con l'interrogazione anche degli altri lettori presenti, o riprendere dall'inizio con l'interrogazione del lettore 1.

Quanto all'alimentazione, ai morsetti PWR della scheda d'interfaccia dovete collegare un alimentatore in continua capace di erogare una tensione di almeno 16 V ed una corrente di 20 mA più 100 mA per ogni lettore di transponder che equipaggerà il sistema. Per esempio, se prevedete di utilizzare quattro periferiche di lettura, l'alimen-

tatore dovrà essere in grado di erogare non meno di 20+400=420 mA. Per eseguire il collegamento tra l'interfaccia di comunicazione e il Persona Computer è necessario utilizzare un adatto cavo diretto (per esempio quelli che vengono utilizzati per i normali modem telefonici possono andare benissimo), che connetta le porte seriali DB9 dei

due sistemi.

ANALISI DEL SOFTWARE

L'analisi del software verrà svolta in due parti: la prima si occuperà di un livello più basso e mostrerà il protocollo di comunicazione utilizzato tra il PC e i lettori transponder; la seconda parte si occuperà invece di un livello più alto e mostrerà un semplice esempio di programma (scritto in Delphi) che, sfruttando il protocollo di comunicazione, sarà in grado di gestire i diversi lettori transponder e i diversi punti di accesso.

Il protocollo di comunicazione

Il protocollo di comunicazione definisce le regole per l'invio delle interrogazioni, delle impostazioni e dei comandi dal PC verso i lettori a transponder; definisce inoltre le modalità con cui i lettori a transponder dovranno rispondere a tali interrogazioni o comandi.

Iniziamo col dire che ogni pacchetto inviato dal PC inizia con i caratteri ASCII “*#”, utilizzati come simboli di sincronizzazione. Tutte le risposte trasmesse dai lettori iniziano invece con i caratteri “ST”.

Successivamente, sono presenti due caratteri utilizzati per indirizzare uno tra i 16 possibili lettori di transponder collegati al bus RS485. I due caratteri indicano rispettivamente le decine e le unità del codice identificativo: per esempio i caratteri “00” indicano il primo lettore; i caratteri “04” indicano invece il lettore a cui è stato impostato indirizzo 4; infine “15” indicano l'ultimo lettore, ossia il numero 15. È utile notare che l'identificazione è presente sia per le comunicazioni effettuate dal PC,

sia per le risposte trasmesse dai lettori.

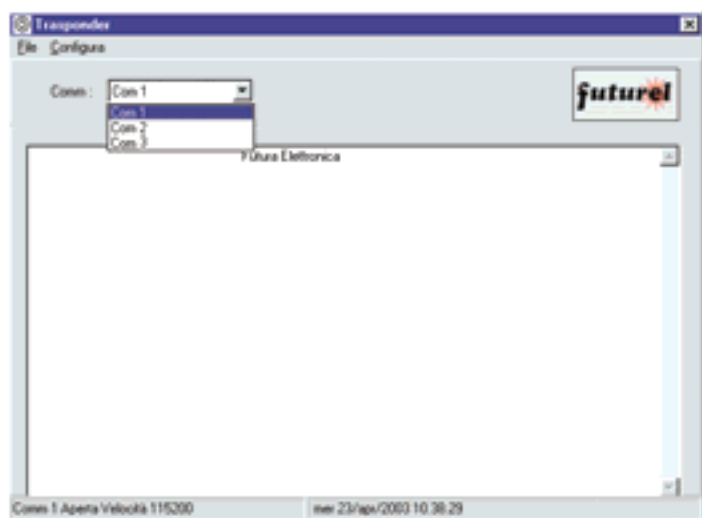
Dopo i simboli di indirizzamento, sono presenti alcune informazioni aggiuntive che identificano il tipo di comando o risposta (per maggiori dettagli e un elenco completo vi rimandiamo al box presente in queste pagine).

Infine, il pacchetto viene terminato dal carattere ASCII 90h se si tratta di un pacchetto inviato dal PC; se invece la trasmissione avviene dai lettori a transponder, il pacchetto viene chiuso dai caratteri "EN".

Il software di gestione

Una volta noto come funziona il protocollo di comunicazione, è possibile realizzare un programma software che, appoggiandosi proprio su questo, realizza il controllo degli utenti e degli accessi. Il software dovrebbe quindi, per prima cosa, andare a interrogare tutti i 16 lettori di transponder per verificare quali sono effettivamente presenti; in seguito dovrebbe andare a richiedere (in modalità polling) la trasmissione dei codici letti, solo ai lettori che sono stati rilevati. Sulla base dei codici ricevuti dovrebbe

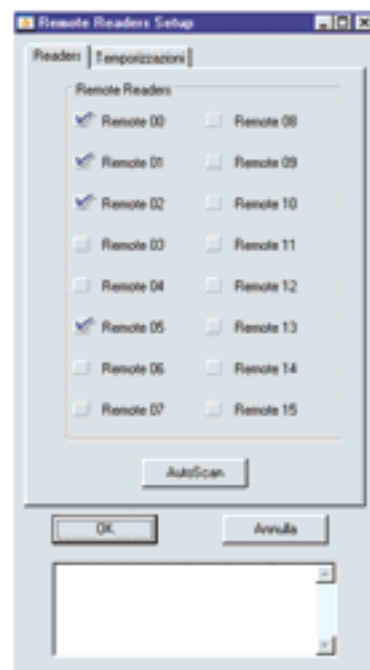
controllare all'interno di un proprio database per verificare se il codice è già presente: in caso affermativo sarebbe possibile identificare l'utente e quindi agire di conseguenza, attivando o meno i relè, eventualmente controllando anche l'orario di accesso e il numero del lettore da cui si sta accedendo, in modo da permettere l'ingresso all'utente solo in determinati orari o da determinate "porte". Se invece il codice non è presente nel database si potrebbe richiedere l'inserimento di un nuovo utente nello stesso (modalità autoapprendimento) oppure eseguire altre operazioni particola-



Lanciato il programma appare la schermata principale.

La prima operazione da compiere è la selezione della porta COM da utilizzare. Dal menu File è possibile avviare le interrogazioni dei lettori o uscire; dal menu Configura è invece possibile entrare nella modalità di configurazione dei lettori transponder o degli utenti.

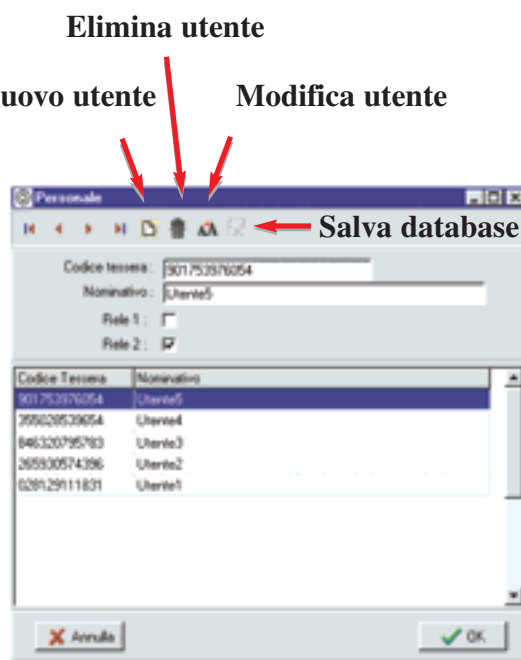
Mediante la funzione Autoscan è possibile attivare la rilevazione automatica di quali lettori transponder risultano collegati al bus RS485. L'impostazione del codice identificativo di ogni lettore avviene tramite il dipswitch a 4 bit presente sul circuito (per i dettagli vi rimandiamo alla scorsa puntata). Come è possibile notare dall'immagine qui a lato non è per forza necessario impostare i codici in modo sequenziale, ma si possono anche effettuare dei "salti" (nel nostro esempio non sono stati impostati i Lettori 03 e 04, mentre è stato inserito il Lettore 05). Premendo OK le impostazioni vengono accettate e quindi salvate.





Schermata relativa alle impostazioni dei Lettori che sono stati individuati mediante la procedura di Autoscan. Per ogni lettore, selezionabile a partire dal codice identificativo, è possibile impostare quale dei 2 relè attivare e il relativo tempo di attivazione. Per ogni relè è possibile specificare un intervallo di attivazione compreso tra 1 e 99 secondi; impostando invece 0 secondi si seleziona la modalità bistabile (un primo comando eccita il relè; un secondo comando lo disattiva, e così via).

Schermata relativa agli utenti memorizzati nel database del programma. Ogni utente è identificato dal codice della tessera transponder e da un nominativo. Per ogni utente è possibile specificare quale dei due relè attivare se si verifica la relativa identificazione (il tempo di attivazione dipende invece dalle impostazioni memorizzate nel singolo lettore). Mediante i pulsanti presenti nella parte superiore è possibile aggiungere, eliminare o modificare i record relativi agli utenti.



ri. Il software dovrebbe inoltre essere in grado di gestire la corretta impostazione dei tempi di attivazione dei relè; vi ricordiamo che questi valori vengono memorizzati all'interno di ogni lettore di transponder, e quindi sono uguali per ogni diverso codice identificato.

Come è facile intuire da questa breve descrizione, le funzionalità che è possibile realizzare via software appoggiandosi al protocollo di comunicazione sono molte e spaziano su di un ampio range di necessità. Chiaramente, per ogni possibile situazione, bisognerebbe realizzare un apposito software; di

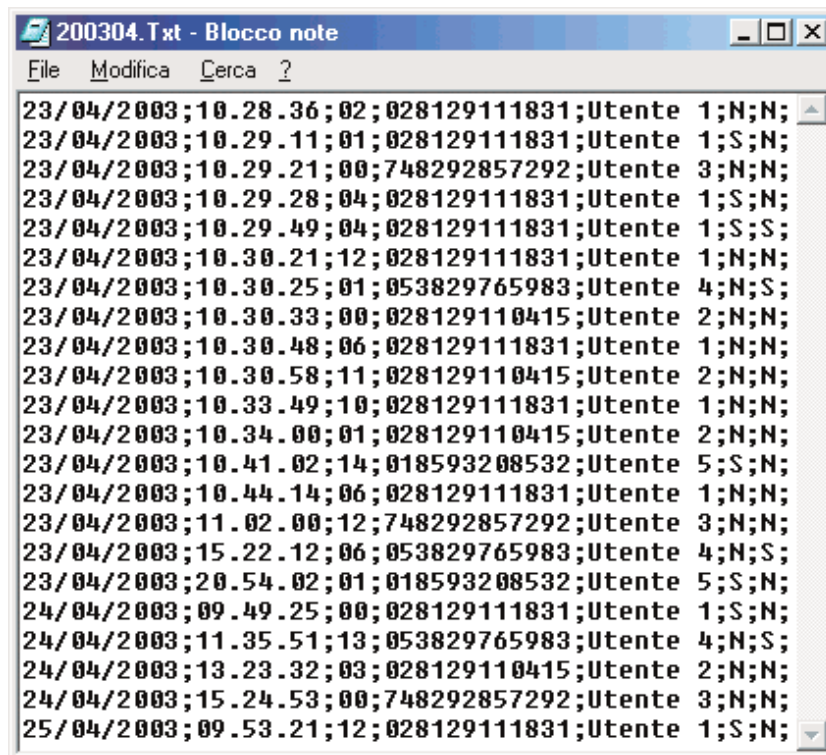
seguito vi mostriamo un esempio di implementazione da noi proposto, che potrebbe andare bene per un utilizzo generale del sistema. Nel caso invece in cui venissero richieste alcune funzioni particolari, sarebbe necessario scrivere un nuovo programma ad hoc per le diverse necessità.

Prima di passare a vedere il programma da noi realizzato, facciamo una considerazione su come andare a eseguire il polling dei diversi lettori: come già detto la comunicazione sulla linea condivisa avviene nel formato RS485. Per questo motivo, nei circuiti elettronici, sono

stati utilizzati degli integrati MAX485 che convertivano i livelli di tensione da TTL a RS485, e viceversa. Prima di eseguire ogni trasmissione, il software presente nel microcontrollore attiva automaticamente il chip MAX485; terminata la trasmissione il chip viene disattivato. Il punto è che le due operazioni richiedono un certo periodo di tempo per essere eseguite; quando quindi si esegue il polling bisogna tenere conto di questo, e introdurre un certo ritardo tra l'interrogazione di un dispositivo e il successivo. Un tempo di alcune decine di millisecondi può andare più che bene;

ESEMPIO FILE DI LOG

Il programma da noi proposto dispone di un'utile e comoda funzionalità di salvataggio dei file di log, in cui vengono scritte tutte le azioni avvenute all'interno di un mese. I dati sono salvati all'interno di file di testo (hanno infatti estensione .txt); ogni azione è scritta su di una riga. Per ogni riga, i campi sono divisi dal carattere ";" in modo da permettere una rapida importazione all'interno di database o fogli elettronici.



```

23/04/2003;10.28.36;02;028129111831;Utente 1;N;N;
23/04/2003;10.29.11;01;028129111831;Utente 1;S;N;
23/04/2003;10.29.21;00;748292857292;Utente 3;N;N;
23/04/2003;10.29.28;04;028129111831;Utente 1;S;N;
23/04/2003;10.29.49;04;028129111831;Utente 1;S;S;
23/04/2003;10.30.21;12;028129111831;Utente 1;N;N;
23/04/2003;10.30.25;01;053829765983;Utente 4;N;S;
23/04/2003;10.30.33;00;028129110415;Utente 2;N;N;
23/04/2003;10.30.48;06;028129111831;Utente 1;N;N;
23/04/2003;10.30.58;11;028129110415;Utente 2;N;N;
23/04/2003;10.33.49;10;028129111831;Utente 1;N;N;
23/04/2003;10.34.00;01;028129110415;Utente 2;N;N;
23/04/2003;10.41.02;14;018593208532;Utente 5;S;N;
23/04/2003;10.44.14;06;028129111831;Utente 1;N;N;
23/04/2003;11.02.00;12;748292857292;Utente 3;N;N;
23/04/2003;15.22.12;06;053829765983;Utente 4;N;S;
23/04/2003;20.54.02;01;018593208532;Utente 5;S;N;
24/04/2003;09.49.25;00;028129111831;Utente 1;S;N;
24/04/2003;11.35.51;13;053829765983;Utente 4;N;S;
24/04/2003;13.23.32;03;028129110415;Utente 2;N;N;
24/04/2003;15.24.53;00;748292857292;Utente 3;N;N;
25/04/2003;09.53.21;12;028129111831;Utente 1;S;N;

```

come si vede è un intervallo molto piccolo che, nell'esecuzione normale di un programma, non dovrebbe presentare nessun problema. A questo punto possiamo iniziare l'analisi del nostro software: il programma è stato scritto in Delphi e gira in ambiente Microsoft Windows 95/98/Me/XP. L'installazione dello stesso risulta semplice; la procedura richiede solo di specificare in quale cartella si desidera copiare i file del programma. Una volta lanciato, appare la schermata principale del programma; questa contiene sostanzialmente un monitor di stato, nel quale appariranno di volta in volta le operazioni eseguite con i comandi manuali dell'utente o con quelli automatici impartiti dal programma stesso. All'interno della stessa finestra è inoltre possibile specificare quale porta COM utilizzare per la trasmissione/ricezione dei dati. La scelta della porta è la prima operazione da compiere, in quanto se il

sistema è collegato su una COM diversa da quella che il programma si aspetta, non è possibile compiere alcuna operazione. Non può invece essere modificata la velocità di comunicazione, in quanto è stato previsto che questa avvenga a 115200 Bit/sec, senza parità, con 8 bit di dati, 1 bit di start ed uno di stop. Comunque non dovrebbero presentarsi problemi, in quanto le COM normalmente montate su tutti i PC costruiti negli ultimi anni sono in grado di gestire una comunicazione a questa velocità e con queste impostazioni. Vi è poi una barra dei menu contenente due voci (*File* e *Configura*), ciascuna delle quali apre un menu a tendina; analizziamole una ad una. La voce *File* contiene due possibilità: *Esci* che permette di terminare il programma e *Avvio* che invece consente di attivare l'interrogazione in polling dei diversi lettori. Se l'interrogazione è attivata, nella finestra principale verranno mostrate alcune informazioni relative alle operazioni in

corso (interrogazione, codice transponder ricevuto, identificazione utente, ecc.). Per esempio apparirà una stringa del tipo *Avvio 07/01/03 15.33.00*; naturalmente data (nel formato gg/mm/aa) e ora (nel formato hh.mm.ss) sono quelle rilevate dall'orologio del PC al momento in cui viene impartito il comando. Per terminare invece la procedura di polling, nel menu *File* la voce *Avvio* viene sostituita da *Stop*. Passiamo adesso al menu più sostanzioso, cioè *Configura*: in esso troviamo le voci *Reader* e *Utenti*, che servono il primo per accedere alla finestra di dialogo in cui si impostano le modalità di funzionamento dei lettori e il secondo per definire ogni utente ed associare al suo rilevamento determinate azioni locali. Aprendo il menu *Configura* e facendo click su *Reader*, si accede alla finestra di dialogo *Remote Readers Setup*, composta da due schede (*Readers* e *Temporizzazioni*) con le quali si possono impostare tutte le funzioni dei lettori. Per l'e-

sattezza, da *Readers* il programma indica quali sono i lettori collegati al sistema. Facendo click sul pulsante *Autoscan* il programma esegue una ricerca sul bus RS485 dei lettori presenti. Tale operazione non è altro che un'interrogazione di tutti gli indirizzi da 0 a 15, completata la quale nella finestra di dialogo appaiono spuntati i lettori rilevati. Si noti che durante l'interrogazione il riquadro testuale in basso mostra lo stato della ricerca. Facendo click sul pulsante *OK* si accetta e si conferma la lista dei lettori rilevati, mentre con *Annulla* si abbandona la procedura e restano valide le ultime impostazioni salvate. Vi facciamo però notare che per permettere il collegamento "a caldo" (ossia senza interrompere l'esecuzione del programma) di un nuovo lettore di transponder, il programma tenta l'interrogazione anche dei lettori che non sono stati identificati mediante la procedura *Autoscan*. Questa procedura può comunque essere utile per verificare che siano stati impostati correttamente i dip-switch di ogni lettore collegato al bus. La scheda *Temporizzazioni* riguarda invece le impostazioni dei due relè di ogni singolo lettore; in essa troviamo due caselle, in ciascuna delle quali si può scrivere il numero di secondi corrispondenti alla durata di attivazione voluta: ad esempio, scrivendo 10 in *Tempo di attivazione Relè 1*, si intende che il relè RL1 del lettore sarà eccitato per circa 10 secondi ogni volta che verrà letto un transponder tra quelli presenti nel database. La selezione del lettore cui si riferisce l'impostazione avviene tramite la casella indicata con *Seleziona Reader*. Dopo avere impostato i diversi tempi di attivazione, fate click sul pulsante *Invia configurazione*; a questo punto il programma invia le impostazioni utilizzando l'opportuno pacchetto di impostazione.

Abbiamo già sottolineato che lasciando a zero il tempo di un relè, questo funzionerà in modalità bistabile: ciò vuol dire che il passaggio di un transponder abilitato determinerà l'inversione dell'attuale stato del relè stesso (si disaccita se è attivo, oppure scatta se è a riposo). È inoltre possibile disattivare uno o entrambi i relè del lettore indicato; in questo modo se in una postazione è necessario comandare un solo dispositivo esterno, si può scegliere di collegarlo al primo relè e di disattivare il secondo. Per esempio, si supponga di voler comandare il solo relè 1 del lettore numero 0 per 2 secondi ogni volta che viene letto un transponder. Ebbene, si dovrà scrivere 2 nella casella *Tempo di attivazione Relè 1*. Bisognerà anche fare click nella casellina accanto alla voce *Tempo di attivazione Relè 1* (così da farvi apparire il segno di spunta). Bisognerà invece deselezionare la casella di spunta accanto alla voce *Tempo di attivazione Relè 2*. Premendo *Invia Configurazione* le impostazioni vengono quindi trasmesse al lettore. Prima di passare alla seconda voce del menu *Configura*, è d'obbligo una precisazione: tutte le impostazioni viste possono essere condotte ed applicate prima che il sistema sia stato avviato con il comando *Avvia* presente nel menu *File*. Queste vengono infatti trasmesse ai rispettivi lettori lungo il bus seriale RS485. Una volta avviato il sistema, il bus viene occupato dalle interrogazioni: non è quindi più possibile trasmettere i comandi di impostazione. A riprova di ciò potete notare che il comando *Reader* risulta non selezionabile (in grigio); per poter modificare le impostazioni occorre bloccare momentaneamente l'interrogazione (menu *File*, voce *Stop*); quindi entrare nel menu *Configura* e selezionare il comando *Reader*, ora disponibile.

Bene, adesso analizziamo la configurazione degli utenti. Questa è accessibile in due modalità: manualmente, aprendo il menu *Configura* e clickando sul comando *Utenti*; in automatico, quando davanti alla bobina di uno dei lettori del sistema viene posto un transponder che non è tra quelli preventivamente appresi.

Nel primo caso si accede alla finestra di dialogo *Configurazione Utenti*, nella quale, per ogni transponder riconosciuto, appare una riga indicante il codice identificativo, il nome assegnato (utile quando un transponder debba identificare una persona) e le eventuali azioni associate (attivazione del relè 1 o del relè 2) al suo passaggio davanti al lettore. Per aggiungere un nuovo utente basta fare click su *Aggiungi* e scrivere manualmente i dati, ossia codice e nome. Vi sono poi due caselle di spunta siglate *Relè 1* e *Relè 2*: la spunta indica che i relativi relè saranno attivati dopo il passaggio del transponder identificato dal codice inserito. Premendo *OK* il nuovo utente viene temporaneamente aggiunto; premendo invece l'icona *Salva* il database degli utenti verrà aggiornato.

Fin qui abbiamo spiegato la definizione manuale; va però detto che il programma è stato realizzato in modo da permettere di aggiungere un transponder diverso da quelli già memorizzati nel database attraverso una tecnica di autoapprendimento. Più precisamente, avvicinando ad un qualsiasi lettore un transponder sconosciuto, viene avviata automaticamente la procedura di definizione di nuovi utenti: a video appare una finestra di richiesta *Aggiungere Nuovo utente?* con la quale il PC chiede all'operatore se vuole creare un nuovo utente; rispondendo *Sì* appare la finestra di dialogo *Configurazione Utenti*, nella quale il campo *Codice* riporta il valore

letto dal transponder.

Se si risponde *No*, il computer ignora il nuovo dispositivo e non determina alcuna azione locale. Resta inteso che se si riavvicina lo stesso transponder a un altro lettore o al medesimo, verrà nuovamente proposta la richiesta di registrazione di un nuovo utente.

Nella finestra *Configurazione Utenti* è inoltre possibile eliminare o modificare i record relativi agli utenti: in entrambi i casi i comandi hanno effetto sulla riga selezionata (per selezionare un utente dalla lista, basta fare click sulla riga che lo contiene).

Un'ultimo aspetto relativo al software di gestione è che questo realizza un file di log in cui memorizza tutte le informazioni relative alle operazioni eseguite; le informazioni sono le stesse che vengono visualizzate nella finestra principa-

le, ma vengono salvate in modo da permetterne una lettura anche in un secondo momento. I dati vengono scritti in file di tipo testo (estensione *.txt*) avente nome nel formato *aaaamm.txt* in cui i primi 4 caratteri identificano l'anno mentre gli ultimi 2 identificano il mese (ogni file contiene infatti le informazioni relative alle operazioni effettuate ogni mese).

Il formato di salvataggio è composto da 7 campi:

- giorno espresso nel formato gg/mm/aaaa;
- ora espressa nel formato hh.mm.ss;
- indirizzo del transponder nel formato d u (il primo carattere per le decine; il secondo per le unità);
- codice identificativo del transponder;
- nome utente associato al codice del transponder;

-due caratteri che identificano se sono stati attivati o meno i rispettivi relè (S indica l'attivazione; N indica la non attivazione).

I dati vengono salvati all'interno del file testuale in cui ogni riga rappresenta un evento; i 7 campi sono inoltre separati dal carattere “;” in modo da consentire una più semplice importazione in database o fogli elettronici, quali per esempio Access o Excel.

**L'articolo completo
del progetto è stato
pubblicato su:**

**Elettronica In n. 79 - 80
Maggio-Giugno 2003**