

Electronica Innovativa

Electronica In

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

118

**Radiocontrollo
16 canali per Velbus**

**Anti-Jammer per
teleallarme GSM**

**Lettore / scrittore
per Smart-Card ACOS2**

Localizzatore Mambo, il set di comandi

Radiocomando 4 CH Rolling Code

Oscilloscopio per PC: software e taratura

Controllo di toni e volume stereofonico

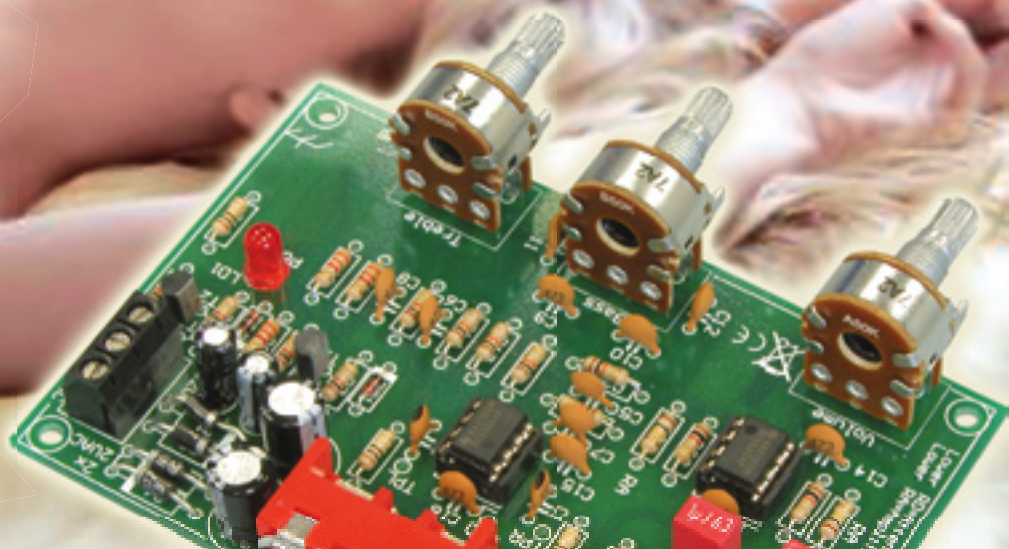


*** **ESCLUSIVO** ***
CORSO LCD
(Liquid Crystal Display)

Controllo toni e volume stereofonico

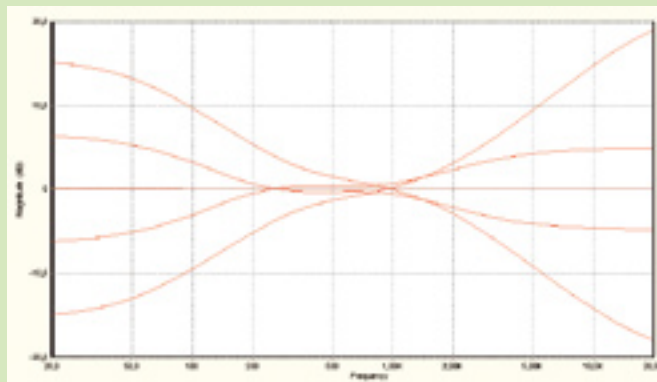
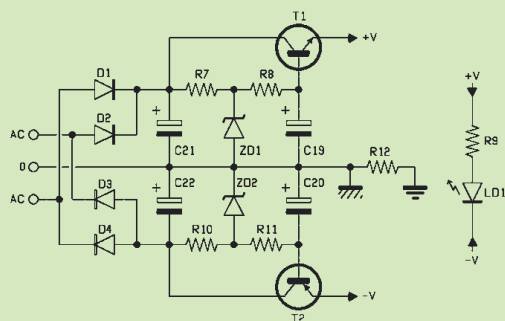
di *Cristiano Ruggeri*

Consente di dotare qualsiasi amplificatore audio stereo di un efficace controllo di toni e volume. La larghezza di banda di 500 kHz, la distorsione inferiore allo 0,005 % e il rapporto segnale/rumore di 98 dB gli conferiscono prestazioni hi-fi. Guadagno unitario con potenziometri in posizione centrale. Completo di stadio di alimentazione.

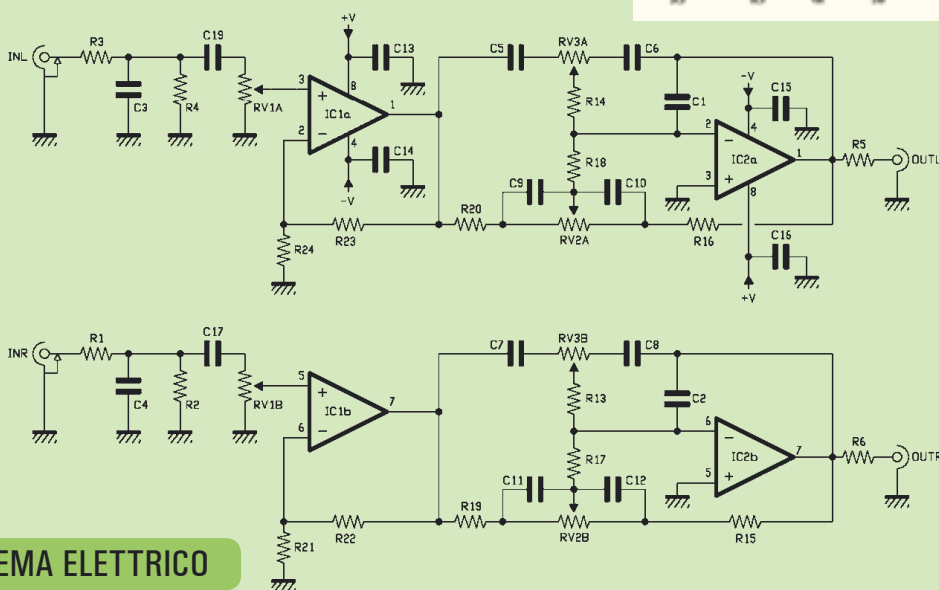


La maggior parte dei progetti audio che compaiono sulle riviste specializzate e nei libri dedicati a questo argomento riguarda stadi amplificatori di potenza, spesso molto semplici dal punto di vista circuitale e costruttivo in quanto realizzati con un singolo circuito integrato monolitico. Questo imperversare di moduli di potenza ha molteplici ragioni: probabilmente la richiesta di sistemi di potenza è maggiore o, forse, c'è una maggior disponibilità di integrati per questo scopo coi quali sbizzarrirsi nelle applicazioni. È anche vero che è molto più semplice realizzare un finale piuttosto che uno stadio d'ingresso

così come risulta più attraente un titolo in copertina che propone la costruzione di un amplificatore da qualche centinaia o migliaia di watt. Anche se i preamplificatori, al contrario degli stadi finali, non possono “esplodere”, realizzare un buon preampli risulta per molti versi più difficile che costruire un finale di potenza. Rumore di fondo, ronzii ed elevata diafonia sono difficili da scovare ed eliminare. D'altra parte un preamplificatore con controllo toni ci vuole sempre, quale che sia il finale utilizzato. Per colmare questa lacuna, ecco il progetto di un semplice ma valido circuito stereo in grado di effettuare la regolazione ➤



In alto a sinistra: l'alimentatore integrato che, all'accensione, fornisce gradualmente la tensione agli amplificatori operazionali. Ciò è reso possibile dalle reti di ritardo e filtraggio costituite da R8-C19 e R11-C20.



In altro a destra: la risposta in frequenza del controllo di toni, realizzato con la classica rete "Baxandall" a due bande. Le curve sono riferite ai due potenziometri in posizione 0%, 25%, 50%, 75% e 100%.

SCHEMA ELETTRICO

dei toni alti, bassi e del volume di una qualsiasi catena di amplificazione di bassa frequenza. La catena di amplificazione consiste generalmente in un preamplificatore seguito da un controllo di tonalità e un finale di potenza; il controllo consente la correzione della tonalità in due zone della gamma audio: i toni bassi (intorno ai 100 Hz) e quelli acuti (tipicamente intorno a 10 kHz).

Il nostro circuito

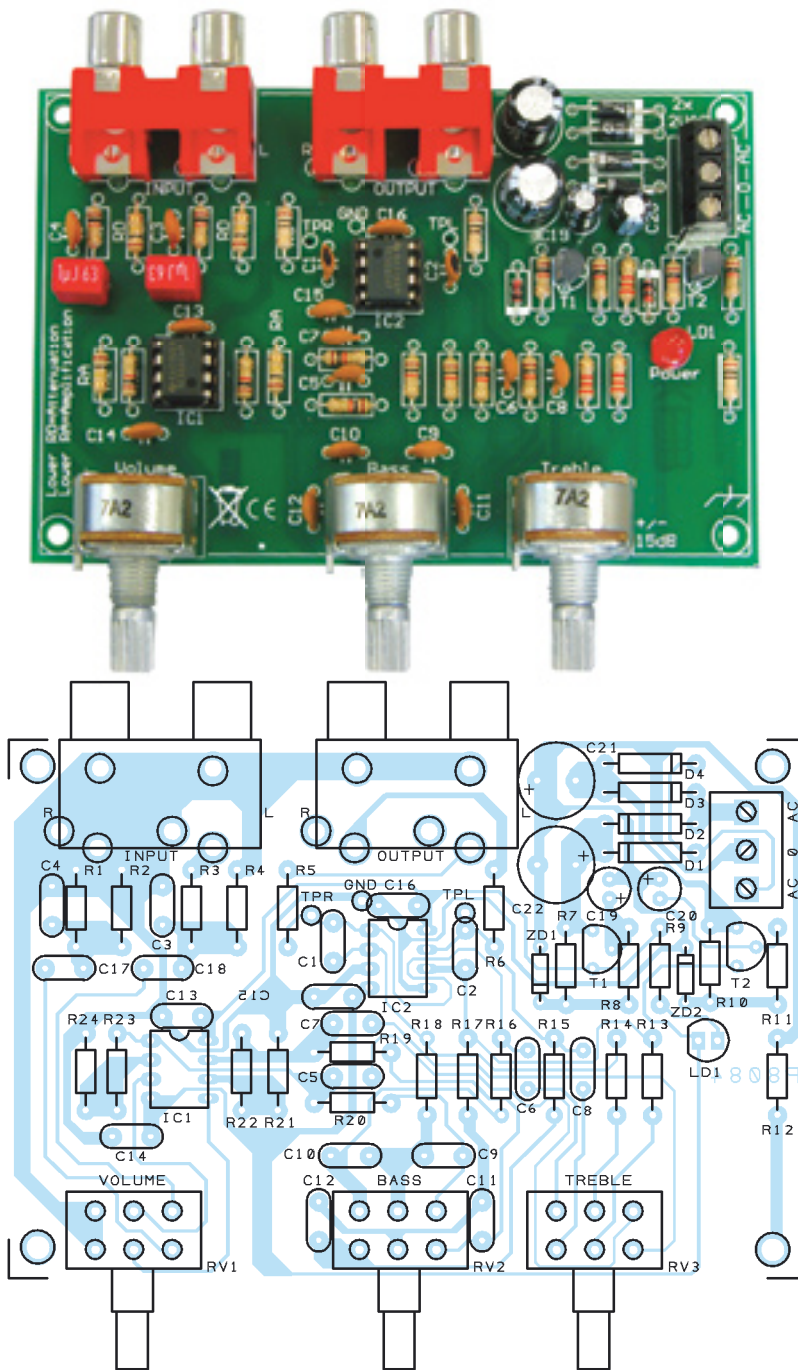
Il progetto descritto in queste pagine è un controllo stereofonico a due vie basato sul classico *ponte di Baxandall attivo*, che permette di operare la correzione sui toni bassi e alti. Il circuito può amplificare o attenuare il segnale audio di 15 dB in corrispondenza delle frequenze di 20 Hz e 15 kHz. L'escursione di $\pm 15\text{dB}$ diminuisce progressivamente fino

ad annullarsi a frequenze dell'ordine di 1 kHz, alle quali il controllo non esercita alcuna influenza. Diamo subito uno sguardo allo schema elettrico, dal quale appare come il circuito sia composto da tre parti: un

alimentatore stabilizzato duale e due sezioni BF (uguali tra loro), destinate a trattare ciascuna un canale della stereofonia. Analizziamo per primo l'alimentatore, che prende tensione da un trasformatore (avente primario da rete e secondario da 2×12 volt, 100 mA) e fornisce $\pm 8,5$ V ben stabilizzati. Ovviamente, al posto del trasformatore di alimentazione con doppio secondario, potremo utilizzare un alimentatore duale in corrente continua in grado di erogare dai 12 ai 15 volt per ramo.

Dal ponte raddrizzatore giungono gli impulsi sinusoidali, con i quali vengono caricati gli elettrolitici C21 e C22 fino ad ottenere due componenti continue, una positiva e l'altra negativa rispetto alla massa di riferimento; la prima raggiunge il collettore del T1 e la seconda quello del T2. I transistor realizzano due regolatori di tensione

- Risposta in frequenza: 5 Hz ÷ 500 kHz;
- Distorsione armonica (@ 1 kHz): 0,005 %;
- Attenuazione/esaltazione a 20 Hz: ± 15 dB;
- Attenuazione/esaltazione a 15 kHz: ± 15 dB;
- Rapporto segnale/rumore: 98 dB;
- Impedenza d'ingresso: 50 kohm;
- Massima tensione d'uscita: 5 V_{eff.};
- Tensione d'alimentazione: 2×12 Vac;
- Corrente assorbita: 100 mA.



ELENCO COMPONENTI:

R1, R3, R7, R8: 1 kohm
 R10, R11, R22, R23: 1 kohm
 R2, R4, R21, R24: 100 kohm
 R5, R6, R12: 10 ohm
 R9, R13, R14: 3,3 kohm
 R15÷R20: 10 kohm
 RV1: Doppio pot. 50 kohm
 RV2, RV3: Doppio pot. 50 kohm, zero centrale
 C1, C2: 15 pF ceramico
 C3, C4: 100 pF ceramico
 C5÷C8: 4,7 nF multistrato
 C9÷C16: 47 nF multistrato
 C17, C18: 1 µF 63V poliestere

C19, C20: 47 µF 25V elt.
 C21, C22: 220 µF 25V elt.
 IC1, IC2: TL072
 ZD1, ZD2: zener 9V1 500mW
 D1÷D4: 1N4007
 T1: BC547
 T2: BC557
 LD1: led 5 mm rosso

Varie:
 - Morsettiera 3 poli
 - Zoccolo 4+4 pin (2 pz.)
 - Doppia presa RCA da cs (2 pz.)
 - Circuito stampato

Il montaggio del controllo di toni è piuttosto semplice. Occorre però fare attenzione alle polarità di diodi e condensatori elettrolitici. Badate inoltre ai potenziometri: quelli dei toni sono differenti da quello del volume in quanto hanno la posizione di zero a metà corsa.

lineari a BJT, nei quali essi operano come emitter-follower portando sul proprio emettitore un potenziale pari a quello del diodo Zener che ne polarizza la base, diminuito della caduta V_{be} (tipicamente 0,6 V). Quindi T1 restituisce circa 8,5 V positivi e T2 altrettanti volt, ma negativi; tali tensioni servono ad alimentare gli amplificatori operazionali utilizzati nel circuito. E veniamo a questi ultimi, che analizziamo con una premessa: trattandosi di un circuito stereo, le sezioni di controllo sono due uguali, una per canale; pertanto limitiamo l'esame ad una sola di esse (L) fermo restando che quanto detto per l'una varrà per l'altra. Ogni blocco di controllo della tonalità è composto da un ponte di Baxandall messo in retroazione a un amplificatore operazionale (IC2) funzionante in configurazione invertente, l'uscita del quale manda il segnale al connettore di uscita (OUTL); il tutto è preceduto da un altro operazionale (IC1) in configurazione non-invertente, che funge da buffer e serve a separare dal ponte lo stadio di uscita dell'apparato BF dal quale si preleva l'audio (INL).

Per comprendere il perché del buffer, bisogna capire come lavora la sezione che controlla effettivamente la tonalità.

Il ponte di Baxandall

Facendo riferimento anche alla figura 1, il ponte Baxandall è la rete di retroazione dell'operazionale IC2a, rete costituita da un'impedenza posta tra uscita e input invertente e un'altra situata fra quest'ultimo e l'ingresso del segnale da amplificare (amplificatore operazionale in configurazione invertente). La prima (X_f) è formata dalla metà del ponte a destra dei cursori dei potenziometri RV3a e RV2a, mentre la seconda (X_i) consta della metà di sinistra. Le formule che descrivono il funzionamento di questo ponte sono abbastanza complesse e una descrizione dettagliata esulereb-

be dai fini dell'articolo. Comunque, assumendo che il simbolo “//” significa “parallelo elettrico”, che “DX” e “SX” indicano la parte di potenziometro a destra o sinistra del cursore, e che la presenza di R18 ed R14 è ininfluente in quanto l'amplificatore operazionale, tipicamente, non assorbe corrente, possiamo affermare con sufficiente approssimazione che l'impedenza Xf è data dalla formula

$$Xf = (RV3a_{DX} + X_{C6}) // [(RV2a_{DX} // X_{C10}) + R16]$$

e, analogamente, per l'impedenza d'ingresso Xi possiamo scrivere

$$Xi = (RV3a_{SX} + X_{C5}) // [(RV2a_{SX} // X_{C9}) + R20]$$

Da quanto sopra, si deduce che Xf e Xi dipendono sia dalla posizione del cursore dei potenziometri sia dalle reattanze dei condensatori, a loro volta legate alla frequenza. Quando i potenziometri hanno i cursori a metà corsa, le impedenze si eguagliano (indipendentemente dalla frequenza) e l'IC2a presenta guadagno unitario su tutta la banda audio. Se si sposta un cursore verso l'ingresso, aumenta l'impedenza di reazione a discapito di quella d'ingresso, il che determina un incremento dell'amplificazione nella banda corrispondente al potenziometro in questione (bassi se si parla di RV2a o acuti se si tratta di RV3a). Viceversa, spostando un cursore verso l'uscita dell'operazionale, la resistenza d'ingresso cresce e quella di retroazione diminuisce: in questo caso si ottiene l'attenuazione della banda corrispondente. Per come è strutturata, la parte

comprendente RV2a opera sulle note gravi, in quanto i condensatori C9 e C10 bypassano il potenziometro sulle note medio-alte e alte; pertanto l'effetto della variazione della resistenza si sente solo sui toni bassi. Quella facente capo a

RV3a interviene sui toni acuti perché C5 e C6 presentano reattanza trascurabile solamente a frequenze medio/alte e acute, quindi neutralizzano l'effetto dell'RV3a alle basse frequenze, dato che in tale gamma la loro impedenza è tanto alta da rendere poco apprezzabile la variazione di quella dello stesso RV3a. Sempre con sufficiente approssimazione, possiamo dire che la frequenza di accordo della sezione degli acuti sta attorno a 1,4 kHz (la costante di tempo corrispondente è data all'incirca dal prodotto $C5 \times RV3a/2$) mentre quella della parte dei bassi si trova nell'intorno dei 135 Hz (circa $C9 \times RV2a/2$). L'operazionale IC1a posto all'ingresso, infine,

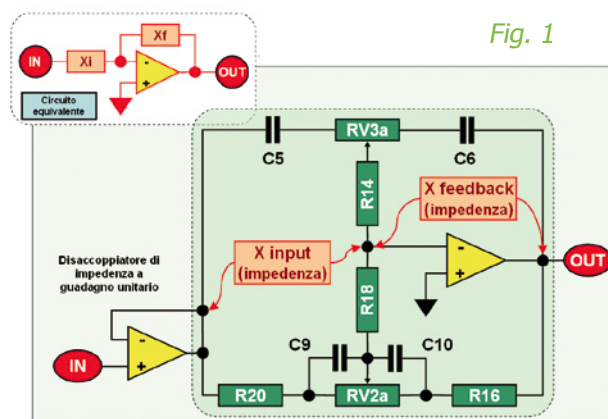
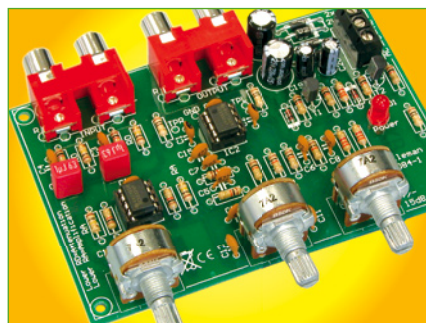


Fig. 1

ha guadagno in tensione praticamente unitario, quindi non amplifica; serve però a dare un'impedenza d'ingresso costante e indipendente dalle variazioni dovute alle diverse posizioni del cursore del potenziometro di volume RV1a, che altererebbero l'impedenza in serie a quella d'ingresso e perciò il guadagno dell'IC2a.

La costruzione

Avendo a disposizione la basetta, saldate per primi resistenze, diodi al silicio (attenzione al catodo, che è l'elettrodo segnato dalla fascetta colorata sul corpo) e zoccoli per gli integrati (inserirete gli operazionali solo finite le saldature); procedete con i condensatori, partendo da quelli non polarizzati e rispettando la polarità degli elettrolitici. Collocate, quindi, il led (il suo catodo è l'elettrodo dalla parte smussata del corpo) e i tre doppi potenziometri; per gli ingressi e le uscite montate le prese RCA doppie, mentre per collegare il secondario del trasformatore usate la morsettiera tripolare.

per il

MATERIALE

Il progetto descritto in queste pagine è disponibile in scatola di montaggio (cod. K8084) al prezzo di 20,00 Euro. Il kit comprende tutti i componenti, la basetta forata e serigrafata, i connettori, i tre potenziometri e la morsettiera. Non è compreso il trasformatore di alimentazione. Il circuito può essere alimentato anche con una sorgente duale in corrente continua in grado di fornire 12 o 15 volt per ramo. Tutti i prezzi si intendono IVA inclusa.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futuranet.it>

Lettore / scrittore per Smart Card ACOS2

di Fabio Riscica

Specificamente progettato per semplici ma sicure operazioni di monetica quali gettoniere, borsellini elettronici o contatori di crediti virtuali, è in grado di leggere e scrivere le smart card di tipo ACOS2. È dotato di tastiera e display con cui modificare e visualizzare il saldo crediti contenuto nella smart card. Alimentabile a batteria per un impiego portatile.

Senza dubbio quello delle smart card è oggi un settore destinato a crescere in modo esponenziale, sia a causa dell'altissimo livello di sicurezza ormai raggiungibile utilizzando queste carte intelligenti, sia per l'affidabilità del sistema di memorizzazione dell'informazione e sia anche per i costi di produzione che, nonostante la maggior complessità dei chip utilizzati, sono scesi in maniera sensibile tanto che questo genere di carte vengono utilizzate anche per promozioni su larga scala con milioni e milioni di pezzi distribuiti. Per chi avesse perduto il corso sulle smart card presentato nei preceden-

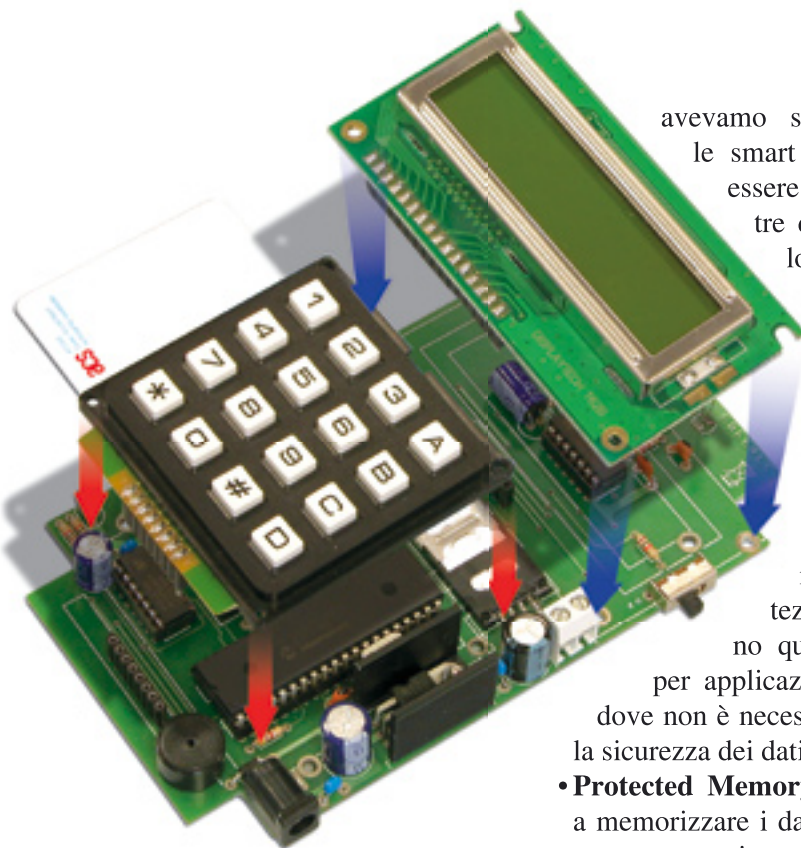
ti numeri della rivista (basato sulle carte ACOS2 della ACS), ricordiamo che queste sono carte a microchip che consentono di memorizzare dati e di elaborare in maniera autonoma operazioni di crittografia su di esse, piuttosto che su di un elaboratore centrale. Di conseguenza i sistemi di autenticazione basati su smart card sono più affidabili di quelli a carta a banda magnetica, in quanto resistenti a manomissione ed immuni ai campi elettromagnetici. Inoltre la capacità di memorizzazione dei dati è notevolmente superiore a quella delle carte a banda magnetica che, come sappiamo, è praticamen-



te inesistente. Per gestire le smart card, esistono in commercio svariati tipi di terminali, dal costo non sempre contenuto. Questi, inoltre, per poter funzionare, spesso devono essere collegati a personal computer o registratori di cassa mentre si interfacciano difficilmente con apparecchiature custom come distributori automatici e macchinari. Da quanto appena esposto si intuisce

che i lettori commerciali, cercando di indirizzare tutte le possibili applicazioni, alla fine peccano un po' in flessibilità (e ciò è strano!). Per questa ragione abbiamo deciso di presentare il progetto di un lettore/scrittore portatile offline di smart card a CPU, il quale, per funzionare, non necessita di un personal computer; abbiamo anche dotato il nostro progetto di uno slot per accogliere

un modulo SAM (Secure Access Module), tramite il quale è possibile incrementare il livello di sicurezza della nostra applicazione. Non si tratta certo di un sistema altamente sofisticato per applicazioni bancarie, ma può essere facilmente impiegato come metodo di pagamento in un'attività commerciale, ad esempio per gestire le consumazioni dei clienti di un esercizio commerciale ➤



avevamo spiegato come le smart card possono essere suddivise in tre differenti tipologie:

•Memory Card.

Permettono di memorizzare dei dati sui quali non viene attivato alcun meccanismo di protezione. Vengono quindi utilizzate

per applicazioni generiche dove non è necessario garantire la sicurezza dei dati.

•**Protected Memory Card.** Oltre a memorizzare i dati, queste card sono protette in scrittura da password. Vengono utilizzate come fidelity card e sistemi di identificazione, ma non è consigliabile utilizzarle come sistemi di pagamento a causa del ridotto livello di sicurezza. Una delle più diffuse protected memory card è la SLE4442, prodotta da Siemens e dotata di una memoria EEPROM di 2 kbit. Viene generalmente utilizzata per la fidelizzazione dei clienti (carte punti, benzina, ecc.).

•**CPU Card.** È la famiglia che raggiunge il livello di sicurezza più elevato. Oltre a memorizzare dati (anche in modo strutturato), le CPU card permettono di eseguire internamente operazioni di crittografia. Vengono utilizzate per implementare la firma digitale, per il commercio elettronico e come sistemi di pagamento. Esistono molti modelli di carte a CPU, che si differenziano sostanzialmente per il tipo di sistema operativo, la capacità di memoria e le prestazioni crittografiche.

La carta cui faremo riferimento è la ACOS2 (ampiamente trattata nel nostro Corso): prodotta da ACS, essa è

dotata di potenzialità crittografiche DES (Digital Encryption Standard), MAC (Message Authentication Capability) e funzionalità ATP (Account Transaction Processing). Fra le possibili applicazioni di queste carte possiamo elencare borsellini elettronici, gestione degli accessi nei centri sportivi, nelle discoteche e negli alberghi, chiavi elettroniche, sistemi di pagamento per i videogiochi e per i parchi di divertimenti. Vediamo ora come è strutturato lo schema elettrico da noi messo a punto.

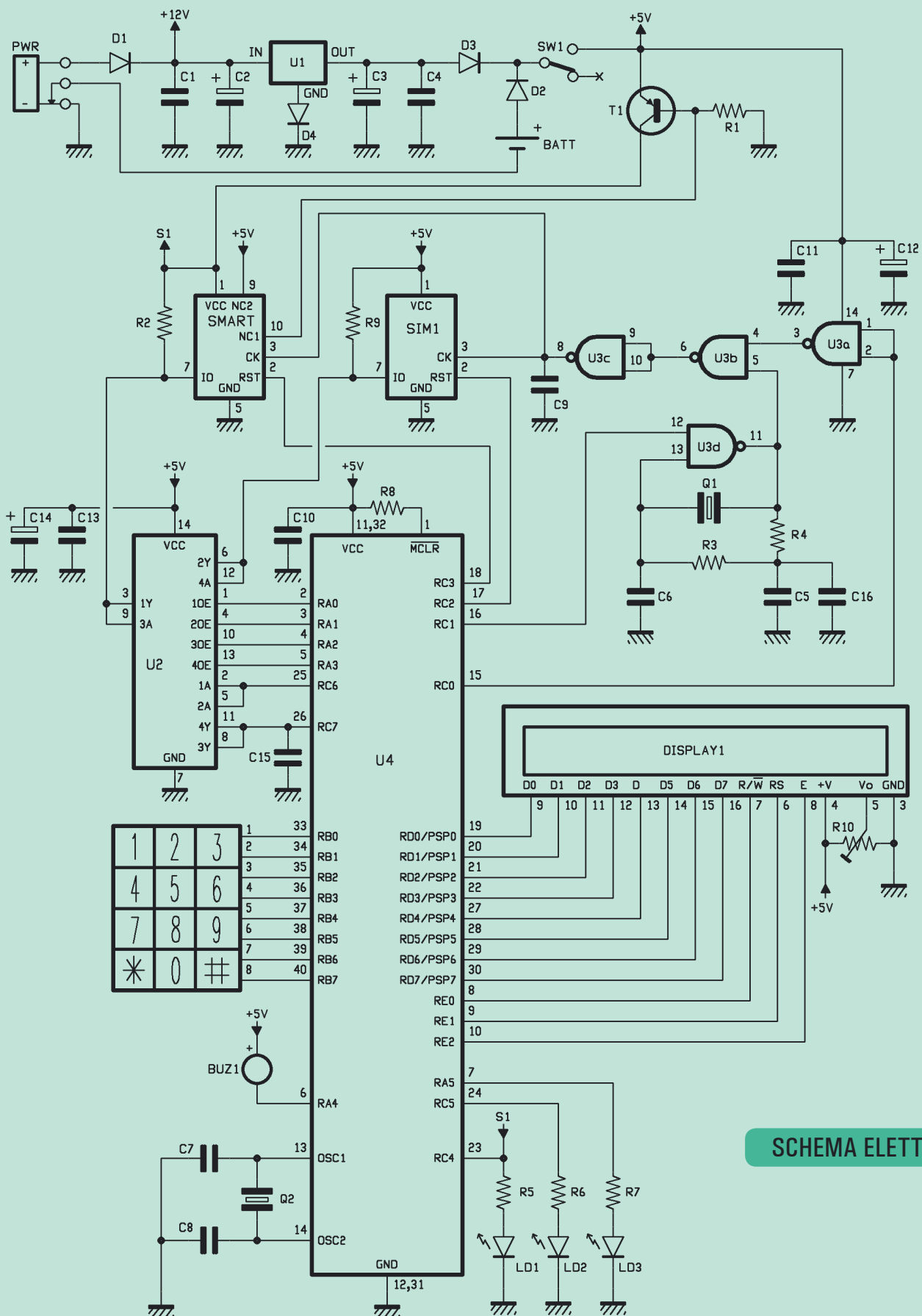
Schema elettrico

Il nostro lettore di smart card presenta un'architettura molto semplice: il cuore del circuito è il microcontrollore U4, un PIC16F877 dotato di 8 kword di memoria programma, 33 linee di I/O ed interfaccia seriale UART. In realtà esso offre molte altre funzioni, ma nel nostro progetto non sono utilizzate. Il programma eseguito dal PIC è stato semplificato nel diagramma di flusso riportato nel riquadro di pagina 28, al quale rimandiamo per ulteriori dettagli; qui procediamo con la descrizione dello schema elettrico.

Nella parte superiore potete trovare il circuito di alimentazione, basato sul classico regolatore 7805, che fornisce la corretta tensione di alimentazione di 5 V a tutti gli integrati del circuito. La tensione continua applicata all'ingresso può variare entro certi limiti, ma noi consigliamo di fornire 9÷12 Vcc, gamma che permette di evitare l'eccessivo surriscaldamento del regolatore, il quale deve comunque essere munito di un adeguato dissipatore. Il diodo D1 ha la funzione di proteggere il circuito dall'inversione di polarità dell'alimentazione, mentre i diodi D2 e D3 consentono di alimentare il circuito a batteria: quest'ultima viene esclusa automaticamente quando si inserisce la spina dell'ali- ➤

che hanno fatto un abbonamento, oppure un sistema di controllo di crediti virtuali, con cui i gestori di negozi possono fidelizzare i propri clienti. In linea di massima, questo dispositivo può trovare impiego in tutte le applicazioni in cui viene erogato un servizio previo pagamento di una quota anticipata, o dove viene effettuato un conteggio di crediti virtuali con cui avere diritto a premi in funzione dei crediti ricevuti.

Questo progetto ha anche una valenza didattica nel senso che può essere considerato un primo approccio alla tecnologia delle smart card: acquisite le informazioni necessarie potrete personalizzare il circuito realizzando magari un sistema di sola lettura delle informazioni oppure, ad esempio, un sistema più complesso con stampante o altre periferiche. In ogni caso non dimenticatevi di leggere il Corso sulle smart card presentato sui fascicoli 108-114. A quanti avessero smarrito uno o più numeri, ricordiamo che questo Corso - così come tutti gli altri Corsi proposti dalla rivista - può essere scaricato in formato digitale dal nostro sito (www.elettronica.in.it). Nel Corso



SCHEMA ELETTRICO



mentatore stabilizzato nella presa PWR. Notate, inoltre, la presenza del diodo D4 sul terminale di massa dello stabilizzatore: in questo modo la tensione di uscita viene elevata a 5,6 V circa, al fine di compensare la caduta di tensione sul diodo D3 (0,6 V) che protegge l'uscita dello stabilizzatore dai rientri di corrente quando il circuito è alimentato a batteria. Le coppie di condensatori C1-C2 e C3-C4 svolgono la funzione di filtro di ingresso e di uscita.

Procedendo verso il basso nell'analisi dello schema, possiamo notare i due connettori per le schede: SMART, in cui viene inserita la smart card utente, e SIM1, che contiene la SAM, necessaria per il corretto funzionamento del circuito (ha la funzione di calcolare il certificato di credito, come verrà illustrato più avanti). La smart card riceve l'alimentazione attraverso il transistor PNP T1, ma solo quando essa è inserita nel connettore SMART. Infatti il contatto di presenza carta, integrato nel connettore è di tipo normalmente chiuso, mantiene la base di T1 al positivo di alimentazione: in questo modo il transistor è interdetto. Una

volta inserita la scheda, il contatto si apre e scollega la base di T1 dal positivo: ora la base del transistor è polarizzata tramite la resistenza R1 verso massa, quindi T1 passa in saturazione alimentando così la smart card.

Entrambe le schede necessitano di un segnale di clock che viene generato da un particolare circuito formato da quattro porte NAND (l'integrato U3, visibile nella parte destra dello schema), un quarzo ed alcuni componenti passivi (i tre condensatori C5, C6, C16 e le due resistenze R3, R4). Questa configurazione ricorda il classico oscillatore a pi-greco, ma è leggermente più complessa: abbiamo infatti voluto prevedere la possibilità di disabilitare l'oscillatore quarzato (tramite il pin RC1 del micro) e di pilotare il terminale di clock della smart card direttamente dal pin RC0 del microcontrollore. In questo modo è possibile utilizzare lo stesso circuito anche per gestire carte sincrone, come quelle a memoria protetta, a patto che si modifichi in modo opportuno il firmware. Sempre relativamente all'oscillatore, occorre prestare attenzione al valo-

re dei componenti passivi, piuttosto critici per l'oscillazione. Nel caso in cui si rilevassero problemi di funzionamento dovuti alle tolleranze di questi ultimi, si può provare ad incrementare di qualche pF il valore di C5, in modo da sbilanciare il ponte ed innescare più facilmente le oscillazioni. Inoltre non bisogna toccare con le mani le piste del circuito stampato durante il funzionamento dell'apparecchio; si deve anche evitare di appoggiare il circuito su supporti capacitivi che possono causare ancora il blocco dell'oscillatore. È pertanto consigliabile inserire il circuito in un contenitore plastico.

Lo scambio di informazioni con le smart card avviene secondo il protocollo seriale standard (9.600 bps ricavati da un clock di 3,579 MHz): il microcontrollore scelto integra un'interfaccia UART, utilizzata per comunicare con entrambe le carte tramite il quadruplo buffer three-state 74HC125 (U2). Abilitando in modo opportuno i singoli buffer di U2, è possibile indirizzare il flusso di dati e la relativa direzione verso la smart card utente o il modulo SAM. La tastiera a matrice collegata alla porta PORTB del microcontrollore consente di digitare gli importi relativi alle transazioni e di selezionare le varie operazioni che si vogliono effettuare. Si noti che non sono presenti le resistenze esterne di pull-up, necessarie per il corretto funzionamento della tastiera a matrice, in quanto sono state utilizzate quelle di cui la PORTB del microcontrollore è dotata (nota relativa al PIC: le resistenze di pull-up della PORTB vengono attivate settando il bit RBPU del registro OPTION). Il display LCD (DISPLAY1), dotato di controller integrato e gestito in modalità parallela a 8 bit, serve per visualizzare sia i dati (letti e scritti) sia il risultato delle operazioni eseguite sulla carta. Il trimmer R10 permette di regolare il contrasto della visualizzazione. Abbiamo previsto

anche tre diodi led ed un buzzer: vedremo più avanti le funzioni ad essi assegnate. Si noti che il led LD1 segnala la presenza della tessera, essendo direttamente alimentato dal transistor T1 quando viene inserita la smart card utente nel connettore SMART.

La smart card ACOS-SAM

Per rendere più sicure le operazioni di lettura/scrittura sulla chip card, il nostro circuito prevede l'impiego di un modulo di autenticazione ACOS-SAM il quale dispone di un set di comandi più esteso rispetto alla tradizionale ACOS di cui ci siamo occupati nel nostro Corso. In questo capitolo vogliamo illustrare i comandi che caratterizzano la creazione del file system e la generazione del certificato di credito, la cui conoscenza è indispensabile per l'operazione di incremento dell'account della tessera utente. Infatti è già stato accennato che, per generare il certificato di credito della smart card ACOS, è necessario effettuare un'elaborazione basata sull'algoritmo DES. La funzione del modulo SAM è quello di implementare un motore crittografico hardware ed alleggerire il firmware da complesse elaborazioni. Con tale scelta si evita inoltre la memorizzazione della chiave di credito direttamente nel codice del microcontrollore, operazione che avrebbe compromesso la sicurezza del nostro circuito di pagamento. Il comando che consente di operare sul file system e di creare files e directory conforme allo standard ISO7816-4, a cui si rimanda per ulteriori approfondimenti, è CREATE FILE (visibile in Tabella 1):

CREATE FILE: 00 E0 00 00
<LENGTH> <DATA>

in cui <LENGTH> è la lunghezza dei dati, dipendente dal tipo di file che si vuole creare mentre <DATA>

Tabella 1

CLA	00 - Clear Mode 04 - SM Mode
INS	E0
P1	00
P2	00
P3	Length of Data
Data	FCP TLV of File to be created

è la struttura FCP TLV dello stesso. Analogamente a quanto già visto per la smart card ACOS, è possibile scrivere i records di un file mediante il comando UPDATE RECORD:

UPDATE RECORD: 00 DC
<NUM> 00
<LENGTH> <DATA>

dove <NUM> è il numero del record che si vuole scrivere, <LENGTH> è la lunghezza dei dati e <DATA> i byte da scrivere. Il comando che consente di generare il certificato necessario per la transazione sull'account è PREPARE ACOS ACCOUNT TRANSACTION ed ha la seguente struttura:

PREPARE ACOS ACCOUNT TRANSACTION: 80 7E
<P1> <P2> 0D <DATA>

<P1> è un byte di cui sono significativi soltanto il bit 0 ed il bit 1. Il bit meno significativo di <P1> caratterizza il tipo di crittografia (LSb=0: triplo DES; LSb=1: singolo DES); il bit 1 indica, se pari ad 1, che la crittografia deve essere eseguita basandosi sulla Session Key. <P2> deve essere pari ad E2 se si vuole calcolare il certificato di credito mentre deve essere pari ad E6 per calcolare il certificato di debito. Il successivo blocco di 13 bytes DATA inviato alla SAM ha la struttura relativa alla transazione <AMOUNT><TTREF><ATREF>, vista nel corso.

Dopo l'esecuzione del comando è possibile richiedere alla SAM il MAC generato mediante il GET RESPONSE, che restituisce un bloc-

co di 11 bytes secondo la struttura <MAC><AMOUNT><TTREF>.

La chiave su cui si basa il comando per la generazione del MAC deve essere preventivamente generata tramite il comando DIVERSIFY KEY, che ha la struttura

DIVERSIFY KEY: 80 72
<P1> <P2> 08 <DATA>

Il valore di <P1> indica il tipo di chiave che deve essere generata (Secret Code=1; Account Key=2; Terminal Key=3; Card Key=4).

I bit di <P2> specificano l'indice della Master Key nel file selezionato. I bit 0-4 identificano l'indice della chiave, il bit 7 indica se la chiave è locale (bit 7 uguale ad 1) o globale (bit 7 uguale a 0).

Gli otto byte <DATA> possono essere utilizzati per trasmettere alla SAM il numero seriale della carta utente a partire dal quale verrà generato un certificato personalizzato. Se la chiave da derivare è ad 8 bit, la formula utilizzata nell'esecuzione del comando per la sua generazione è la seguente:

DERIVED KEY =
3DES(KEY, DATA)

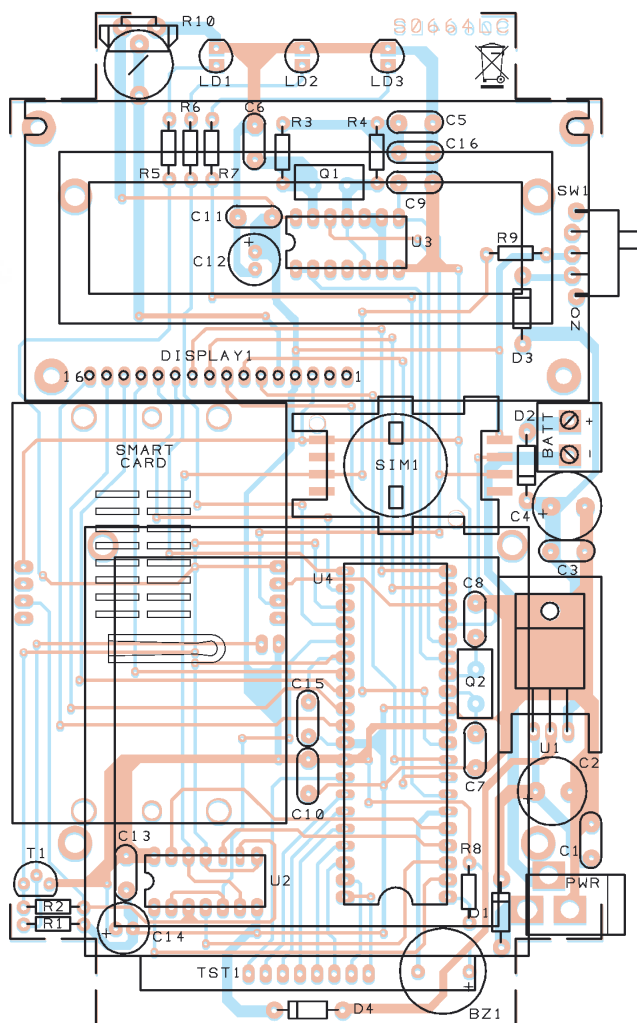
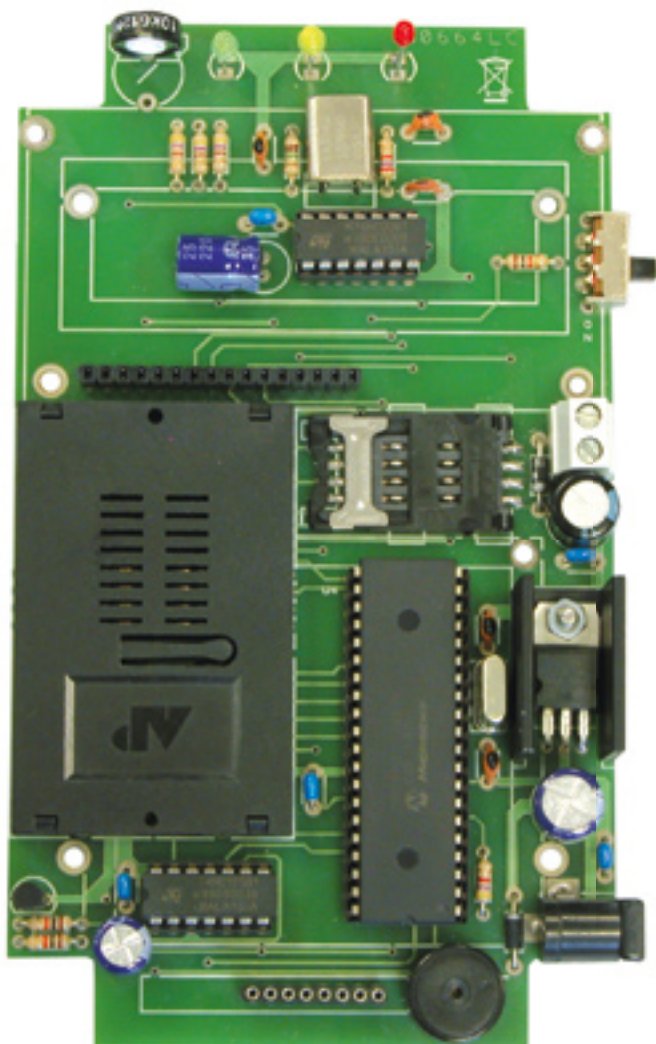
Inizializzazione della ACOS-SAM e delle smart card utente

A questo punto è evidente che, per poter operare con la smart card utente in maniera corretta, è necessario inizializzare opportunamente la struttura del modulo SAM.

Una volta definita arbitrariamente una chiave principale di 16 bytes, relativa alla nostra applicazione, bisogna derivare la Credit Key ad 8 bytes, crittografando il dato

00 00 00 00 00 00 00 00

con la chiave principale mediante l'algoritmo 3DES. A tale scopo è possibile scaricare un apposito software dal sito della rivista. La Credit Key generata dal software 3DES ➤



In questo caso il circuito stampato presenta dimensioni tali da rendere abbastanza semplice il montaggio, anche perchè tutti i componenti utilizzati sono di tipo tradizionale. Occorre solo prestare attenzione alle polarità di diodi e condensatori elettrolitici.

ELENCO COMPONENTI:

R1: 10 kohm
 R2: 10 kohm
 R3: 1 Mohm
 R4: 2,7 kohm
 R5÷R7: 470 ohm
 R8: 4,7 kohm
 R9: 10 kohm
 R10: trimmer 10 kohm
 C1, C3: 100 nF multistrato
 C2: 470 µF 25 VL elettrolitico
 C4: 470 µF 25 VL elettrolitico
 C5÷C8: 22 pF ceramico
 C9: 47 pF ceramico
 C11: 100 nF multistrato
 C12: 220 µF 25 VL elettrolitico
 C13: 100 nF multistrato
 C14: 220 µF 25 VL elettrolitico
 C15, C16: 100 nF multistrato
 U1: 7805

U2: 74HC125
 U3: 74HC00
 U4: PIC16F877-20P (MF664)
 T1: BC557
 Q1: Quarzo 3,579545 MHz
 Q2: Quarzo 10 MHz
 D1÷D4: 1N4007
 LD1: led 3 mm verde
 LD2: led 3 mm giallo
 LD3: led 3 mm rosso
 SW1: deviatore slitta
 BZ1: Buzzer 12V con elettronica
 TST1: tastiera 16 tasti a matrice
 DISPLAY1: display LCD 16x2

Varie:
 - Plug alimentazione

- Porta SIM a libro
 - Connettore smart card
 - Morsettiera 2 poli
 - Zoccolo 7+7 (2 pz.)
 - Zoccolo 20+20
 - Dissipatore (ML26)
 - Vite 10 mm 3 MA
 - Vite 6 mm 3 MA (3 PZ.)
 - Dado 3 MA (4 pz.)
 - Strip femmina 8 pin (2 pz.)
 - Strip maschio 8 pin
 - Strip femmina 16 pin (2 pz.)
 - Strip maschio 16 pin
 - Torretta M/F 5 mm
 - Torretta M/F 10 mm
 - Torretta F/F 10 mm (2 PZ.)
 - Circuito stampato codice S664

deve poi essere memorizzata nelle smart card utente utilizzando il lettore MINILECTOR ed il software ACOS2-INIT, descritto nella settima parte del nostro Corso sulle smart card, presentato sul numero 114. Ad esempio, supponendo di fissare la chiave principale pari a

11 22 33 44 55 66 77 88 99 AA BB
CC DD EE FF 00

la Credit Key generata dal software 3-DES che deve essere memorizzata in tutte le carte utente del circuito di pagamento è pari a:

11 0D 81 FC C8 94 55 06.

Per quanto riguarda l'architettura del modulo SAM, al suo interno è stato previsto un solo file avente identificativo:

40 02

che deve contenere nel suo primo record la chiave principale di 16 bytes. I comandi che devono essere inviati al modulo SAM per l'inizializzazione, tramite un lettore standard di smart card, sono i seguenti:

;Creazione Master File

00 E0 00 00 0A 62 08 82 02 3F FF
83 02 3F 00

;Creazione Key File

00 E0 00 00 1B 62 19 02 05 0C 01
00 15 04 83 02 40 02 88 01 02 8A 01
01 8C 06 6B 03 FF FF FF FF

;Inizializzazione della chiave principale (da personalizzare)

00 DC 00 02 15 81 02 FF FF 00 11
22 33 44 55 66 77 88 99 AA BB CC
DD EE FF 00

Se i comandi vengono eseguiti correttamente, la SAM risponde con la canonica coppia di bytes:

90 00

a ciascuno di essi, segnalando che il modulo è stato correttamente ini-

zializzato e può essere inserito nell'apposito alloggiamento posto sul circuito stampato.

Si ricordi che la presenza del modulo SAM è necessaria per poter effettuare operazioni di accredito dell'account delle carte utente, ma il nostro lettore può comunque eseguire operazioni di addebito anche se tale modulo non è presente. Una particolare applicazione potrebbe infatti prevedere di effettuare soltanto addebiti sulle carte utente con il nostro lettore portatile e gli accrediti mediante il software ACOS2-PURSE presentato nell'ultima puntata del nostro Corso sulle smart card. In tal caso la presenza del modulo SAM non è richiesta, perché la sua funzione verrebbe svolta dal software ACOS2-PURSE installato nel personal computer del gestore.

Realizzazione pratica

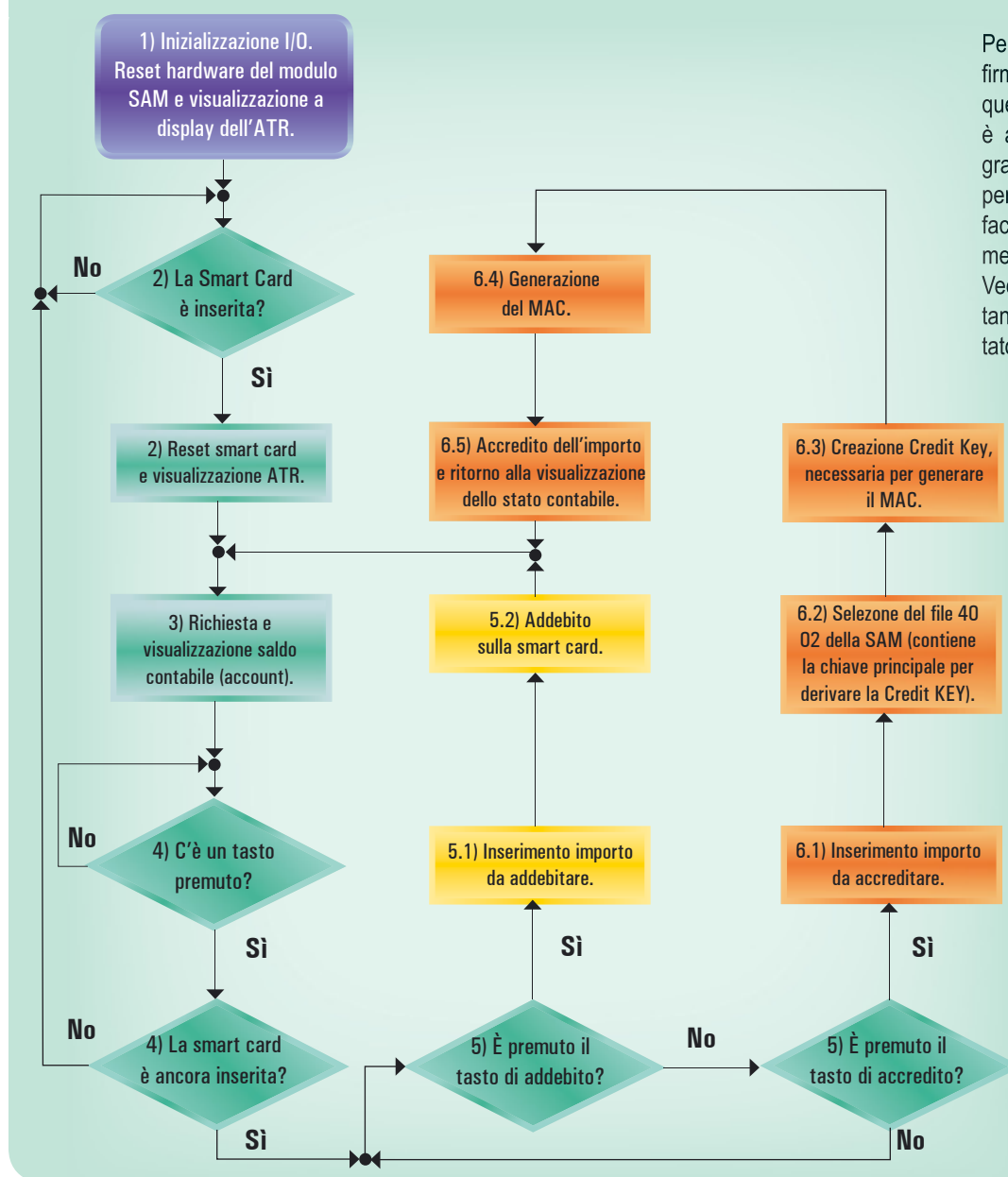
La realizzazione di questo progetto non comporta grosse difficoltà in quanto vengono impiegati componenti ad inserzione facilmente reperibili: l'unico componente a montaggio superficiale è il socket per la SAM (SIM1). Il circuito stampato, di dimensioni generose, non è affollato di componenti e la maggior parte di questi ultimi è piuttosto ingombrante; riteniamo quindi che la realizzazione di questo progetto sia alla portata di chiunque.

Procuratevi tutti i componenti necessari e scaricate dal sito della rivista il firmware per il microcontrollore ed i

master in scala 1:1 per la realizzazione del circuito stampato. Qualora doveste avere difficoltà nella realizzazione pratica del PCB, potete rivolgervi direttamente all'azienda che realizza i prototipi dei nostri circuiti stampati (www.mdsrl.it), indicando il codice del circuito in questione (S0664).

Diamo per assodato che siate in possesso di un circuito stampato e procediamo con il montaggio dei componenti. Il primo è sicuramente il socket per la SIM card, unico a montaggio superficiale. Prendete il circuito stampato e posizionate con il lato componenti di fronte a voi e le piazzole dei tre led in alto (vedrete il codice dello stampato in alto a destra). Posizionate il socket prestando attenzione al verso: lo sportello si deve aprire verso destra e i due perni su cui esso ruota devono essere posti in prossimità del diodo D2. Saldare uno degli otto pin, prestando attenzione all'allineamento di tutti gli altri rispetto alle piazzole. Verificato che l'allineamento è perfetto, procedete saldando tutti gli altri pin: suggeriamo di effettuare la seconda saldatura sul pin diagonalmente opposto al primo, in modo da rendere stabile l'allineamento. A questo punto potete montare tutti i componenti a basso >





Per quanto le funzioni svolte dal firmware siano semplici, la sequenza di operazioni eseguite è abbastanza articolata. Il diagramma di flusso qui a fianco permette di comprendere più facilmente la logica di funzionamento del firmware.

Vediamole più in dettaglio, aiutandoci con il commento riportato di seguito. Immediatamente dopo l'accensione, il programma inizializza le periferiche del micro, esegue il reset hardware del modulo SAM e visualizza sul display l'ATR (Answer To Reset) restituito dalla carta (1). Si ricorda che ogni tipo di smart card è caratterizzato da uno specifico ATR. Quindi inizia un loop di attesa con cui il programma verifica l'inserimento della smart card utente: quando questa viene inserita, il micro esegue un reset hardware su di essa e visualizza il relativo ATR sul display (2). Quindi effettua l'interrogazione dell'account della smart card utente tramite i comandi IN-

profilo, quali le resistenze, i diodi (attenzione al verso), il quarzo Q1 (deve essere montato sdraiato sul PCB), tutti i condensatori non elettrolitici e gli zoccoli dei circuiti integrati (rispettate il verso della tacca di riferimento). Procedete montando l'interruttore SW1, i condensatori elettrolitici (ancora una volta rispettate la polarità), i tre led (LD1, LD2 e LD3, rispettivamente verde, giallo e rosso; il lato piatto è rivolto all'interno dello stampato), il quarzo Q2 (verticale), il buzzer, il transistor T1, il connettore PWR, il trimmer verticale R10, i connettori pin-strip

femmina a passo 2,54 mm e la morsettiera per la batteria (BATT). Montate a questo punto il connettore per la smart card (non c'è rischio di sbagliare l'orientamento: può essere posizionato in un solo verso). Terminate quindi con lo stabilizzatore di tensione U5: per quest'ultimo, piegate a 90° i suoi pin in modo che possa essere fissato meccanicamente allo stampato anche con la vite da 3 MA oltre che con la saldatura dei terminali. Posizionatelo sullo stampato interponendo l'aletta di raffreddamento a U tra il suo dissipatore e il PCB, quindi serrate il pacchetto

stabilizzatore-aletta-PCB con la vite da 3 MA. Saldate infine i tre terminali dello stabilizzatore. Potete quindi innestare i tre chip e la SIM card sui rispettivi connettori (nota: il micro e la SIM devono essere stati programmati). Prendete ora la tastiera e saldate sul lato opposto ai tasti un connettore pin-strip a passo 2,54 mm ad otto contatti. Infine il display, sul quale dovete saldare un connettore pin-strip (sempre a passo 2,54 mm) a 16 contatti dal lato opposto al vetrino. Inserite ora la tastiera ed il display sui rispettivi connettori (aiutatevi con la figura visibile nel

QUIRE ACCOUNT e GET RESPONSE; l'account corrente viene quindi visualizzato sul display (3). Ora il micro attende la pressione del tasto relativo all'addebito o all'accredito. Se la smart card utente viene estratta (4), non appena viene premuto uno dei due tasti, il programma riprende il ciclo di controllo presenza tessera (2).

Nell'ipotesi in cui la tessera sia ancora inserita quando viene premuto un pulsante, il programma controlla quale pulsante è premuto.

Se viene premuto il pulsante di addebito (5) il programma richiede l'importo da addebitare (5.1), effettua l'addebito sulla smart card utente mediante il comando DEBIT (5.2) quindi ritorna al punto 3, qualora si volesse effettuare una nuova operazione.

Se invece viene premuto il pulsante di accredito (5), il programma, richiede l'importo (6.1), seleziona il file 40 02 della SAM, contenente la chiave principale utilizzata per derivare la Credit Key (6.2), crea sulla SAM la Credit Key, necessaria per generare il MAC, mediante il comando DIVERSIFY KEY (6.3), genera il MAC della SAM mediante i comandi PREPARE ACOS ACCOUNT TRANSACTION e GET RESPONSE (6.4), effettua l'accredito dell'importo mediante il comando CREDIT (che utilizza il MAC generato dalla SAM nel passo precedente), quindi ritorna ad attendere eventuali altre istruzioni per nuove operazioni di lettura o scrittura (3).

piano di montaggio) e preparatevi "psicologicamente" ad alimentare il vostro lettore di smart card.

Utilizzo del lettore

Dopo aver montato correttamente tutti i componenti secondo le indicazioni riportate nel capitolo precedente, inizializzato ed installato il modulo SAM (se tale funzionalità è richiesta) e le carte utente con il software ACOS2-INIT, seguendo le indicazioni riportate nel paragrafo precedente, siamo pronti per testare il nostro lettore portatile.

Alimentando il lettore ed agendo sull'interruttore di accensione SW1 appare un messaggio relativo alla versione del firmware, una porzione dell'ATR della SAM (3BBE950000..9000) e la richiesta di inserimento della carta utente. Nel caso in cui l'ATR visualizzato fosse diverso da quello riportato, bisogna verificare se il modulo SAM è presente e se l'oscillatore a 3,579 MHz opera correttamente. È possibile modificare il contrasto di visualizzazione del display agendo sul trimmer R10.

Inserendo la carta utente, appare una porzione dell'ATR della carta (3BBE110000..9000) ed il contenuto dell'account. Anche qui bisogna prestare attenzione al valore dell'ATR.

Le due possibili operazioni (di addebito o accredito) sono selezionabili agendo rispettivamente sui pulsanti A e B. Viene quindi richiesto l'importo relativo, che è composto da cinque cifre numeriche: è necessario digitare tutte le cinque cifre (ad esempio, se si vuole inserire

un importo pari ad 1 unità, bisogna digitare 00001). Il sistema effettua l'operazione richiesta e quindi visualizza il nuovo saldo contabile della scheda. L'accensione dei led LD2 o LD3 indica che è in corso un'operazione di addebito (LD2, giallo) o di accredito (LD3, rosso). Quando una smart card è inserita, il led verde LD1 è acceso. Una volta estratta la carta utente, possiamo aggiornare la visualizzazione del display (che richiede l'inserimento di una nuova carta) premendo un pulsante qualsiasi. Si ricordi infine quanto già detto durante il corso sulle operazioni di addebito o accredito delle smart card ACOS:

- non è possibile eseguire un'operazione di addebito superiore all'account corrente della carta;
- non è possibile eseguire un'operazione di accredito che oltrepassi il MAXBALANCE impostato sulla carta;
- se il MAC non è corretto (ad esempio se la Credit Key memorizzata sulla carta non è coerente con la chiave principale memorizzata nella SAM o se il modulo SAM non è presente o non sta funzionando), l'accredito non viene eseguito.

Dopo aver effettuato l'operazione richiesta, il lettore visualizza il contenuto dell'account: se ciò non dovesse accadere, il motivo è certamente da imputare al verificarsi ad una delle situazioni appena descritte.

per il

MATERIALE

I componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato, il firmware in formato esadecimale e i programmi necessari per la programmazione delle schede ACOS2 e SAM sono scaricabili dal sito della rivista (www.elettronica.in). Il microcontrollore programmato (cod. MF664, 18,00 Euro), le schede cod. ACOS3SAM (22,00 Euro) e ACOS2-8K (3,90 Euro) ed il sistema di sviluppo (cod. SDK-ACR38, 119,00 Euro) sono disponibili presso Futura Elettronica. I prezzi sono comprensivi d'IVA.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futuranet.it>

Radiocontrollo 16 canali per impianti Velbus

di Corrado Rossi



Permette di comandare via radio 16 carichi collegati a dispositivi Velbus. Si comporta esattamente come il controllo a pulsanti VMB8PB, ma il comando avviene tramite uno o più trasmettitori. Utilizza una scheda seriale VMB1RS per convertire i comandi ricevuti in stringhe di dati compatibili con gli impianti Velbus.

Nel corso dell'articolo presentato sul fascicolo 117 avevamo anticipato che stavamo lavorando ad un progetto pratico in grado di interfacciarsi direttamente alla rete domestica Velbus; nelle conclusioni finali avevamo anche detto che stavamo realizzando un sistema di controllo remoto a 2 o 4 canali, con il quale comandare via radio i vari automatismi presenti in una casa. In corso d'opera ci siamo resi conto che, semplicemente apportando una piccola modifica al circuito, potevamo realizzare facilmente un sistema idoneo a gestire 16 canali anziché solamente 4; questa evoluzione ha permes-

so di realizzare un dispositivo più versatile e addirittura più economico dello stesso modulo Velbus che il nostro circuito emula. Nella fattispecie il circuito descritto in queste pagine è in grado di funzionare come due moduli pulsanti a otto ingressi VMB8PB. Ripensate per un istante all'impianto dimostrativo visto nelle puntate precedenti: in esso sono presenti tre moduli di ingresso, due a 8 pulsanti (VMB8PB) e uno a 6 pulsanti (VMB6IN), per un totale di 22 ingressi, dei quali se ne utilizzano solamente 15. Il progetto proposto in queste pagine, quindi, è più che sufficiente per controllare l'intero impianto. ➤

La stringa di controllo compatibile col protocollo Velbus viene generata da un modulo seriale VMB1RS controllato da un micro PIC al quale giungono i dati provenienti dal modulo ricevitore RF a 433 MHz. In estrema sintesi il sistema simula la pressione di uno dei pulsanti del modulo VMB8PB ogni volta che viene premuto un tasto del telecomando, inviando lungo la linea CAN-Bus la stringa di dati corrispondente.

Un ulteriore punto di forza del nostro sistema è rappresentato dal fatto di poter utilizzare telecomandi a 1, 2, e 4 canali (eventualmente anche 8 e 16) a patto che abbiano la stessa codifica base (nel nostro caso si tratta di codifica a base UM3750); in pratica potremmo tenere in macchina un telecomando a due canali per attivare solo il cancello e la porta basculante ed in ogni locale il telecomando necessario per le funzioni che vogliamo comandare da quella zona. Inoltre questo dispositivo può operare con tutti i controlli già presenti in rete senza interferire con essi, a ulteriore conferma della sua versatilità. Se quanto esposto finora relativamente alla famiglia Velbus vi ha convinti della validità della soluzione, questo progetto non mancherà di convincervi: scopriamo insieme tutti i suoi segreti.

Struttura del dispositivo

Prima di analizzare in dettaglio lo schema elettrico, vediamo per som-

mi capi come è stato strutturato il progetto aiutandoci con la figura 1, in cui appare evidente che il dispositivo sfrutta le capacità di conversione della scheda VMB1RS delle informazioni inviate al suo ingresso in stringhe di dati compatibili col protocollo Velbus. Questa scelta offre due vantaggi: consente di collegare facilmente alla rete qualsiasi microcontrollore a condizione che esso sia dotato di una interfaccia seriale con livelli RS232; in secondo luogo i comandi che il micro deve inviare sono strutturati in formato ASCII quindi molto semplici dal punto di vista software. Il micro non deve far altro che trasmettere stringhe di dati asincroni in formato 38.400-8-N-1 rispettando la sintassi vista nel fascicolo precedente; tale sintassi è facilmente interpretabile da qualsiasi micro, anche di prestazioni limitate.

Vediamo dunque come è costituita la scheda in oggetto. Poiché deve far

parte della catena di prodotti Velbus, essa è stata studiata per poter essere collegata direttamente alla linea a 4 fili CAN-Bus dalla quale preleva la tensione di alimentazione. Le linee GND e +12Vcc alimentano sia il radiocontrollo che la scheda seriale VMB1RS mentre le due linee dati sono connesse ai corrispondenti terminali di questa stessa scheda.

All'interno del circuito da noi messo a punto possiamo distinguere due elementi fondamentali: un ricevitore RF, al quale è affidato il compito di convertire le informazioni trasmesse dal telecomando in una serie di dati digitali; un microcontrollore, che legge i dati in arrivo dalla sezione a radiofrequenza e, una volta riconosciuto quale pulsante è stato premuto, ne converte il comando logico in uno stream di dati da inviare alla scheda VMB1RS. Il nostro dispositivo, durante il normale funzionamento, gestisce due indirizzi destinati a due scopi differenti: il primo

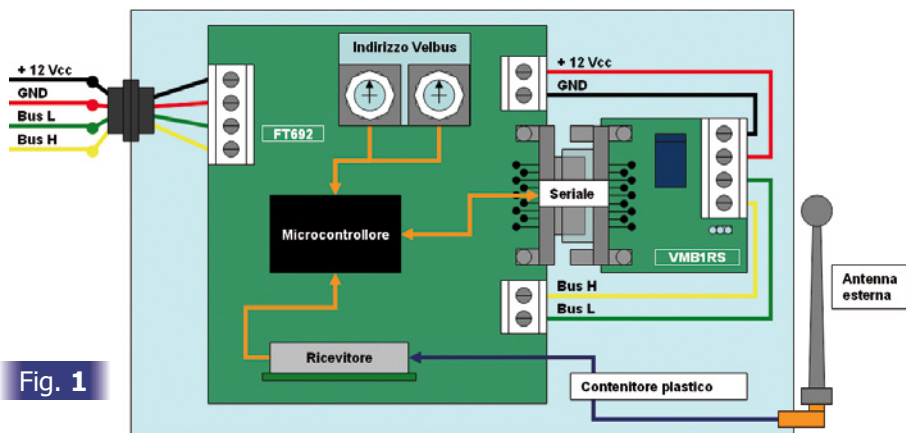


Fig. 1

RM ELETTRONICA SNC

vendita componenti elettronici

rivenditore autorizzato:

**FUTURA
ELETTRONICA**

ELETTRONICA

G.P.E.

**ELSE
Kit**

Via Val Sillaro, 38 - 00141 ROMA - tel. 06/8104753

definisce il valore della codifica a radiofrequenza che, lo ricordiamo, utilizza un sistema tipo UM3750 a 12 bit. Il codice utilizzato viene autoappreso con la procedura che vedremo tra breve. Il secondo indirizzo è quello che identifica la nostra scheda all'interno di una rete Velbus. Come sappiamo, tutti i circuiti collegati alla rete devono disporre di un proprio indirizzo tra quelli liberi (non occupati da altri dispositivi e diversi da quelli riservati). Tale indirizzo, scelto tra 222 possibili combinazioni (256 meno 34 riservate), va impostato con i due rotary switch (HR1 e HR2) impiegati nel circuito. Gli indirizzi vanno impostati in esadecimale, ovvero da 00 a FF. Il primo switch (HR1) definisce il nibble più significativo (il simbolo a sinistra) mentre HR2 definisce il nibble meno significativo (quello di destra). Ad esempio, se la scheda deve presentare l'indirizzo 6B, dobbiamo posizionare HR1 su 6 e HR2 su B. Ricordiamo che il dispositivo necessita di due indirizzi in quanto emula due moduli pulsanti VMB8PB; il primo indirizzo è quello impostato con gli switch mentre il secondo viene definito automaticamente e corrisponde a quello successivo (nell'esempio il secondo indirizzo corrisponde a 6C).

Per quanto riguarda la codifica base del telecomando, questa viene acquisita automaticamente alla pressione di un pulsante su un telecomando qualora i preselettori vengano impostati su F0.

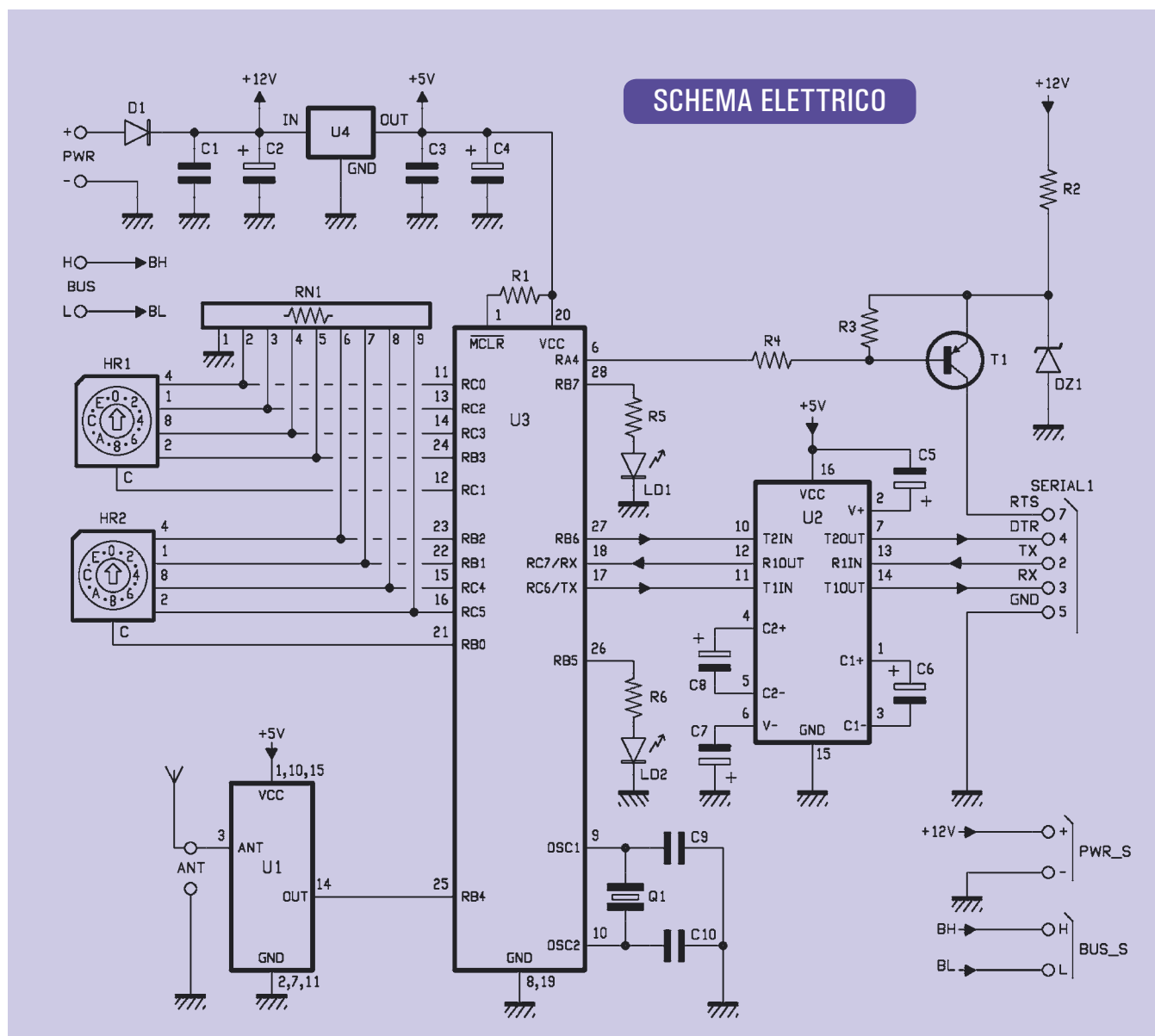
Come viene gestita la codifica radio

Per comprendere chiaramente come viene effettuato il riconoscimento del canale attivo, è necessario ricordare come funziona la codifica UM3750 (questo chip può essere usato sia come encoder che come decoder) utilizzata nel telecomando. L'encoder genera in uscita un flusso seriale di dati che contiene,

Pulsante emulato	Impostazione dip-switch				
	A1 ÷ A8	A9	A10	A11	A12
P1	Indirizzo base	OFF	OFF	OFF	OFF
P2	Indirizzo base	OFF	OFF	OFF	ON
P3	Indirizzo base	OFF	OFF	ON	OFF
P4	Indirizzo base	OFF	OFF	ON	ON
P5	Indirizzo base	OFF	ON	OFF	OFF
P6	Indirizzo base	OFF	ON	OFF	ON
P7	Indirizzo base	OFF	ON	ON	OFF
P8	Indirizzo base	OFF	ON	ON	ON
P9	Indirizzo base	ON	OFF	OFF	OFF
P10	Indirizzo base	ON	OFF	OFF	ON
P11	Indirizzo base	ON	OFF	ON	OFF
P12	Indirizzo base	ON	OFF	ON	ON
P13	Indirizzo base	ON	ON	OFF	OFF
P14	Indirizzo base	ON	ON	OFF	ON
P15	Indirizzo base	ON	ON	ON	OFF
P16	Indirizzo base	ON	ON	ON	ON

oltre ad alcuni campi di controllo e sincronismo, anche la “chiave di riconoscimento” (programmabile tramite dip switch e composta dai 12 bit da A1 a A12), chiave che identifica sia il telecomando che il tasto premuto su di esso. Tipicamente, un telecomando monocanale utilizza tutti i bit di codifica. Un bicanale utilizza da A1 a A11, delegando ad A12 la definizione del pulsante premuto; un quadricanale ha la codifica base definita da A1÷A10 mentre A11 e A12 definiscono i quattro pulsanti, e così via. Mentre la perdita di un bit ad ogni raddoppio di pulsanti è una condizione obbligatoria, non è invece obbligatorio che i bit dedicati ai pulsanti siano sequenziali. Generalizzando, esiste sempre una parte di dati fissa, che possiamo chiamare “codice base”, ed una parte variabile che rappresenta il “codice pulsante”. Ritornando ai dati in uscita, il flusso seriale viene utilizzato per modulare una portante RF; questo segnale viene captato e decodificato dal nostro circuito. Il microcontrollore identifica il tasto premuto e genera la stringa di comando per il sistema CAN-

Bus, esattamente come se fosse stato premuto un pulsante del modulo VMB8PB. Per comprendere come dobbiamo impostare il dip-switch del telecomando (o dei telecomandi), osserviamo la Tabella 1 riportata in questa pagina. I primi otto bit sono comuni a tutti i telecomandi e definiscono il “codice base”. Se utilizziamo un telecomando a 16 canali la programmazione è molto semplice in quanto questo dispositivo utilizza un dip-switch a 8 bit che va impostato col codice base. Ogni volta che viene premuto un tasto, la matrice interna genera automaticamente la parte della stringa mancante ed il ricevitore identifica immediatamente quale pulsante è stato premuto. Se però, come avviene in pratica, utilizziamo dei telecomandi a 4 canali, dobbiamo impostare il dip-switch interno in modo da ottenere l'emulazione dei pulsanti da P1 a P4 o da P5 a P8 o, ancora, da P13 a P16: in pratica dobbiamo definire quali pulsanti debbono simulare i nostri 4 tasti. Da questo punto di vista la tabella è intuitiva. Dato per scontato che i primi otto bit debbono essere ➤



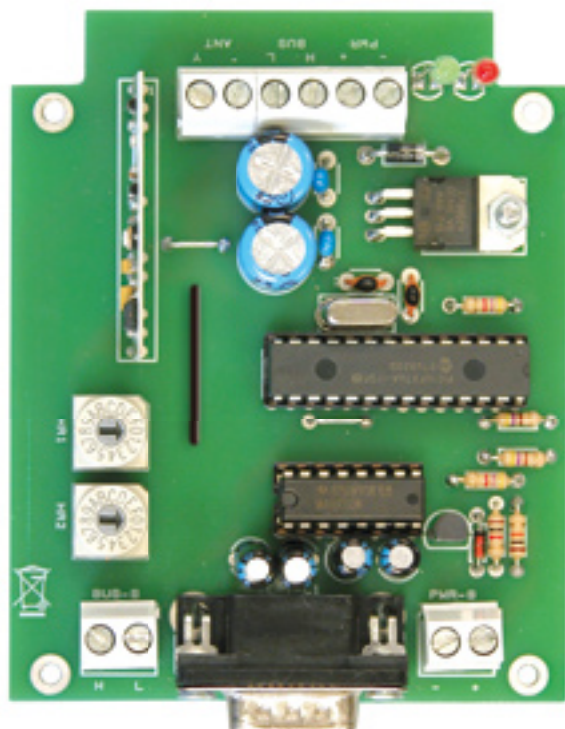
impostati nello stesso modo in tutti i telecomandi, la cosa che cambia riguarda l'impostazione dei bit A9 e A10. Quindi, per emulare, ad esempio, i pulsanti P1-P4 dovremo porre entrambi i bit relativi ad A9 e A10 in posizione OFF; per emulare P9-P12 dovremo porre in ON il dip A9 ed in OFF il dip A10 e così via. Semplicissimo! Ma torniamo brevemente all'impostazione del codice base. Nei dispositivi standard, sia il telecomando che il ricevitore sono dotati di un dip switch e, affinché la coppia TX/RX funzioni correttamente, essi devono presentare la medesima sequenza di ON e OFF su tutti i microinterruttori. Però, come visibile nello schema, il nostro ricevitore

non è dotato di dip switch e quindi il codice va impostato in maniera differente. Come? Presto detto. Attraverso il primo telecomando (che pertanto diventa la "matrice di codice") si programma sul dip switch la base a 8 bit (A1÷A8); quindi, a ricevitore spento, si imposta il valore F0 sui due preselettori binari rotativi. Infine si accende il ricevitore che resta in attesa della prima trasmissione del TX. Ora si preme un pulsante qualsiasi del telecomando matrice e lo si mantiene premuto fino a quando il led verde non inizia a lampeggiare. In questo momento il microcontrollore acquisisce i primi 8 bit relativi al codice base (A1÷A8); a questo punto si possono riportare i

due preselettori al valore precedente che rappresenta l'indirizzo Velbus. Tutti i telecomandi aggiuntivi devono avere lo stesso codice base impostato sul telecomando matrice: ciò significa che i dip switch A1÷A8 devono essere impostati in maniera identica. Ritourneremo, comunque, sull'argomento nel paragrafo riguardante l'utilizzo pratico.

Schema elettrico

Una volta compreso come è strutturato questo radiocontrollo e quale logica segue, la comprensione dello schema elettrico è abbastanza intuitiva. Partiamo da U1: si tratta di un ricevitore RF che converte quanto viene ricevuto in antenna (pin 3) in



ELENCO COMPONENTI:

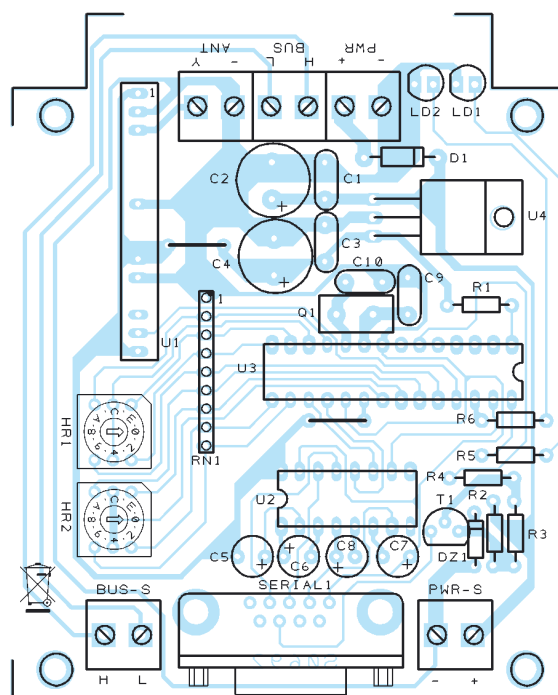
R1: 4,7 kohm
R2: 1 kohm
R3: 10 kohm
R4: 4,7 kohm
R5: 470 ohm
R6: 470 ohm
RN1: rete resistiva 8x10 kohm + C
C1, C3: 100 nF multistrato
C2: 470 μ F 25 VL elettrolitico
C4: 470 μ F 16 VL elettrolitico

C5÷C8: 1 μ F 100 VL elettrolitico
C9, C10: 15 pF ceramico
D1: 1N4007
DZ1: Zener 7,5 V 400mW
T1: BC557
Q1: Quarzo 16 MHz
U1: AC RX2
U2: MAX232
U3: PIC16F876A
U4: 7805

LD1: led 3 mm rosso
LD2: led 3 mm verde
HR1, HR2: Selettore rotativo esadecimale

Varie:

- Morsettiere 2 poli (5 pz.)
- Connettore DB9 maschio
- Zoccolo 8+8
- Zoccolo 14+14
- circuito stampato codice S692



una serie di dati digitali disponibili sul suo piedino 14. Tale serie di dati viene trasferita al microcontrollore (U3) per mezzo del piedino 25 (RB4). Il firmware confronta l'indirizzo base ricevuto con quello memorizzato e, una volta verificata la corrispondenza, estrae gli ultimi quattro bit che, in pratica, rappresentano il pulsante attivato. Infine genera una stringa di comando da inviare alla linea Velbus.

Per fare ciò il PIC comunica con la scheda VMB1RS (che necessita delle linee RX, TX, RTS e DTR) attraverso il convertitore di livello U2, il classico e arcinoto MAX232, associato al transistor T1 che, in questa particolare configurazione, si

comporta come un driver RS232. Il micro, inoltre, controlla lo stato dei due preselettori binari rotativi HR1 e HR2, utilizzati per definire l'indirizzo Velbus assegnato alla scheda. Poiché questo dispositivo controlla 16 canali (leggasi "pulsanti"), esso si comporta come due schede VMB8PB a otto canali, quindi, occupa due indirizzi Velbus ed in particolare quello impostato sui rotary switch e quello immediatamente successivo.

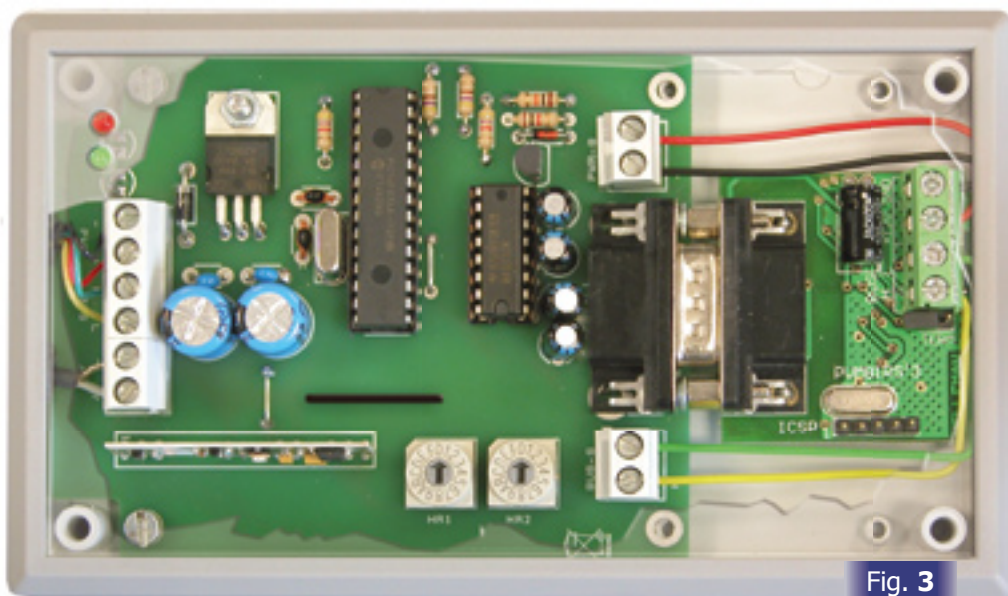
Il micro gestisce anche lo stato dei due led LD2 (verde) e LD1 (rosso), che comunicano uno la ricezione di un comando valido da uno dei telecomandi, e l'altro l'avvenuta trasmissione del comando sul bus

seriale. Completa il circuito l'alimentatore, realizzato con il classico 7805. Come sempre, D1 protegge il circuito da inversioni di polarità accidentali, i condensatori di ingresso C1 e C2 realizzano un filtro sulla linea a 12 V, mentre i condensatori C3 e C4 filtrano e ripuliscono da eventuali rumori provenienti dal bus la linea a 5 V con cui viene alimentata tutta l'elettronica.

Realizzazione pratica

La realizzazione di questo circuito non presenta grosse difficoltà in quanto vengono utilizzati solamente componenti a montaggio tradizionale ed il CS è di tipo monofaccia. Come sempre, il master può essere ➤

scaricato in formato 1:1 dal sito della rivista. Una volta realizzato il PCB, potete iniziare il montaggio partendo dai componenti a basso profilo, ossia tutte le resistenze, i ponticelli, i condensatori non polarizzati, gli elettrolitici C5, C6, C7 e C8. Procedete quindi montando il transistor T1 (rivolgete il lato piatto verso l'esterno del circuito stampato), tutti i diodi zener e tradizionali, prestando attenzione alla polarità. Montate ora gli zoccoli dei due circuiti integrati (attenzione al verso della tacca di riferimento) e i due commutatori binari rotativi (HR1 e HR2). Posizionate e saldate, ora, uno alla volta i componenti ad alto profilo, cioè il quarzo Q1, il connettore della seriale, le morsettiere ed i condensatori elettrolitici C2 e C4. Si prosegue montando lo stabilizzatore di tensione il cui corpo va fissato allo stampato. Montate ora il modulo RF Aurel U1 (AC-RX2): a vostra discrezione, potete sia saldarlo direttamente sullo stampato sia innestarlo su dei connettori a tulipano. Non potete rischiare di montare il ricevitore RF al contrario poiché



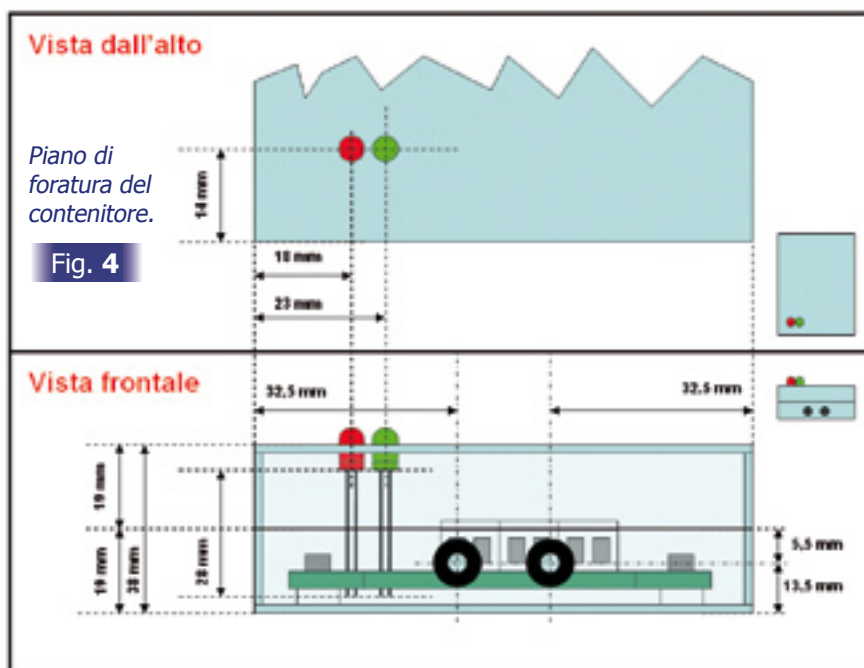
Il controllo via radio per reti Velbus è stato inserito all'interno di un contenitore plastico Teko mod. 660.

esso può essere posizionato in un solo modo. Rimangono da montare i due led: in questo caso dovete avere a disposizione il contenitore plastico, in modo da poter verificare con esattezza la lunghezza dei terminali affinché le teste possano fuoriuscire dal coperchio del contenitore: nel nostro prototipo i terminali dei led sono stati tagliati alla lunghezza di 28 mm. Innestate ora la scheda VMB1RS sul connettore DB9, quindi effettuate i quattro collegamenti filati necessari. La realizzazione della parte elettronica è terminata.

Ora occorre inserire il circuito nel contenitore, sul quale devono essere realizzati i quattro fori, due per far fuoriuscire i fili dalla base e due per far sporgere la testa dei led dal coperchio. Prendete ora la base del contenitore (quella da cui si inseriscono le quattro viti di fissaggio del coperchio) e posizionate al suo interno il circuito, facendo corrispondere i quattro fori di fissaggio con le quattro torrette plastiche, quindi identificate il lato corto in prossimità della morsettiera a sei contatti, lato sul quale dovete effettuare i due fori per il passaggio dei cavi. Rimuovete il circuito e tracciate i riferimenti dei due fori, i quali si trovano rispettivamente a 32,5 mm di distanza dal bordo laterale e a 5,5 mm dal profilo superiore.

Prendete ora il coperchio: sulla sua parte superiore dovete praticare i due fori per permettere ai led di fuoriuscire; tracciate i punti esatti in cui effettuare i fori: il centro dei led si trova a 14 mm dalla base (intesa come lato corto) e a 18 e 23 mm rispetto al lato sinistro (lato lungo).

A questo punto, avendo tracciato tutti i centri, potete iniziare a forare (raccomandiamo di prestare la dovuta attenzione). Per i led da 3 mm



Riconoscimento pulsante e trasmissione comando

Il protocollo seriale Velbus è piuttosto complesso ma grazie alla scheda seriale VMB1RS, che converte da seriale a Velbus e viceversa, inviare dei comandi sul bus è molto facile. Dal listato riportato di seguito, notate come poche righe di programma (localizzate nella routine TXVEL) consentono di comporre stringhe di comando e di inviarle direttamente ai dispositivi in rete. Per esempio, la pressione di un pulsante genera una stringa simile a "0F F8 05 04 00 01 00 00 EF 04", in cui, da sinistra a destra, troviamo:

"inizio stringa" (0F), "priorità" (F8), "indirizzo" (05), "stato RTR e numero di databyte" (04), "comunicazione stato pulsanti" (00), "puls. premuto" (01), "puls. rilasciato" (00), "puls. premuto a lungo" (00), "checksum" (EF), "terminatore stringa" (04).

```
*****
TRASMETTI:
  IF VALIDO=1 THEN
    GOSUB LEGGIADDRESS
    GOSUB RESETBUFF
    TMP=1
    IF CH>=8 THEN
      CH=CH-8
      ADDRESS=ADDRESS+1
    ENDIF
    TMP=TMP << CH
    PRIORITA=$F8
    NBYTE=4
    SELECT CASE PRESSIONE
      CASE 0
        BUFF[2]=TMP
      CASE 1
        BUFF[1]=TMP
      CASE 2
        BUFF[3]=TMP
    END SELECT
    GOSUB TXVEL
  ENDIF
RETURN
*****
TXVEL:
  HIGH LEDR
  IF BYTE2=1 THEN
    ADDRESS=ADDRESS+1
  ENDIF
  GOSUB CALCOLACHK
  HSEROUT [$0F]
  HSEROUT [PRIORITA,ADDRESS]
  HSEROUT [NBYTE]
  FOR TMP=0 TO NBYTE-1
    HSEROUT [BUFF[TMP]]
  NEXT TMP
  HSEROUT [CHK,$04]
  GOSUB LEGGIADDRESS
  LOW LEDR
RETURN
*****
CALCOLACHK:
  VALORE=$0F+PRIORITA+ADDRESS+NBYTE+BUFF[0]+_
  BUFF[1]+BUFF[2]+BUFF[3]+BUFF[4]+BUFF[5]+_
  BUFF[6]+BUFF[7]
  CHK=VALORE ^ %11111111
  CHK=CHK+1
RETURN
*****
```

```
*****
' Questa routine viene chiamata dalla routine che rileva
' la pressione di un tasto del telecomando.
' Se la base dell'indirizzo ricevuto è valida
' legge l'indirizzo impostato sui rotary switch e
' cancella BUFF (array che viene impostato in base
' alla pressione del pulsante).
' Se il tasto premuto è associato ai canali 9÷16 allora
' incrementa l'indirizzo Velbus di una unità.
'
' Shift (sinistra) per identificare il pulsante premuto.
' Imposta messaggio a priorità alta.
' Definisce che la stringa contiene quattro databyte.
' In base alla durata della pressione, imposta databyte.
'
' Pulsante rilasciato
' Pulsante premuto brevemente
' Pulsante premuto a lungo.
' Chiama la routine di trasmissione su seriale.
' Fine routine.
*****
' Routine che compone e invia la stringa seriale.
' Attiva il led rosso (trasmissione seriale in corso)
' Se il pulsante è associato ad un canale da 9 a 16
' incrementa l'indirizzo Velbus di una unità.
'
' Richiama la routine per il calcolo del Checksum.
' Inizia trasmissione. Invia carattere inizio stringa.
' INVIA priorità ed indirizzo.
' INVIA il numero di databyte che seguiranno.
' NBYTE contiene anche lo stato dell'RTR.
' INVIA i databyte definiti precedentemente.
'
' INVIA checksum e terminatore di stringa.
' Torna a rileggere l'indirizzo dei rotary switch.
' Spegne il led rosso (trasmissione seriale terminata)
' Fine routine.
*****
' Routine per il calcolo del checksum.
' Esegue una somma numerica di tutti i byte partendo dal
' delimitatore di inizio stringa e terminando all'ultimo
' byte dati valido.
' Prende in considerazione il byte meno significativo,
' lo inverte bit a bit e lo incrementa di una unità.
' Fine routine.
*****
```

potete forare il coperchio con una punta da 3,5 mm, magari facendo prima un foro guida da 2 mm.

Posizionate ora il coperchio sulla base: i led fuoriescono quanto basta perché si possano vedere, mentre dai fori sul fianco è possibile vedere la morsettiera.

A montaggio ultimato, il radiocontrollo si presenta come in figura 3.

Utilizzo

Abbiamo già detto che questo dispositivo necessita di due indirizzi: uno per far sì che la rete Velbus lo riconosca (selezionabile tramite i commutatori binari rotativi HR1 e HR2)

ed uno che definisce la codifica base del telecomando (UM3750). Occorre inoltre che abbiate a disposizione una rete Velbus cablata e funzionante a cui dovete collegare il radiocontrollo: vi suggeriamo di porre un interruttore in serie alla linea +12Vcc. Se non avete una rete completa a disposizione, è necessario che disponiate almeno di un apparato di comando, ad esempio il VMB4RY, alimentato e collegato via bus a questo circuito. Vi occorre, inoltre, un telecomando con almeno 4 pulsanti. Programmate sul telecomando la codifica base (Bit A1÷A8), quindi collegate alla morsettiera del circui-

to uno spezzone di filo lungo 17 cm che funge da antenna. Eventualmente, al posto dello spezzone di filo, potete utilizzare una antenna a stilo con corpo in gomma, collegandola tramite un cavetto RF adeguato (per esempio RG178): questa soluzione, migliorando l'efficienza in ricezione, aumenta la portata in trasmissione. Inoltre l'antenna a stilo può essere posizionata anche all'esterno in quanto resistente alle intemperie. A questo punto potete accendere il radiocontrollo e impostare i pre-selettori binari rotativi HR1 e HR2 a F0: in questa condizione nessun led è acceso (vedere il riquadro ➤

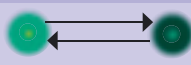
“Segnalazioni visive”). Premete un pulsante del telecomando e mantenete premuto fino a quando il led verde non inizia a lampeggiare velocemente, segnalando così di avere acquisito l’indirizzo base. Potete ora riposizionare HR1 e HR2 all’indirizzo Velbus assegnato in origine. Il led verde segnala la ricerca di un segnale valido lampeggiando velocemente. Premete un pulsante sul telecomando ed attendete: entro mezzo secondo circa il sistema riconosce il comando, accende il led verde a luce fissa ed attiva brevemente il led rosso per comunicare che ha inviato alla scheda VMB1RS (quindi al CAN-Bus) una stringa di comando dipendente dal pulsante premuto.

Per avere la certezza che il comando è stato inviato al bus, il dispositivo deve essere anche accoppiato alla rete o all’apparato, oltre che collegato ad essi.

Ci spieghiamo meglio. Nel caso abbiate realizzato il semplice impianto descritto nei precedenti fascicoli, potete disattivare temporaneamente le due schede VMB8PB, aventi indirizzo 05 e 06 e assegnare quindi indirizzo 05 al nostro radiocontrollo (in questo modo esso occupa sia l’indirizzo base che l’indirizzo base+1, quindi 06).

Se ora premete i vari pulsanti del telecomando, gli attuatori Velbus devono reagire esattamente come reagirebbero in presenza di istruzioni provenienti dalle due schede escluse in precedenza.

Il radiocontrollo comunica i vari stati operativi per mezzo di segnalazioni luminose effettuate tramite due led, uno verde ed uno rosso. Vediamo di seguito le varie situazioni che si possono presentare.

Segnalazione	Descrizione
	Lampeggio alternato verde/rosso: il radiocontrollo è in attesa dell’indirizzo base (indirizzo Velbus F0), che viene fornito dal primo telecomando attivato.
	Led verde acceso a luce fissa: è stato riconosciuto l’indirizzo base del telecomando (sia in apprendimento che in scansione).
	Led verde che lampeggia velocemente: il radiocontrollo attende un segnale valido da un telecomando.
	Il led verde è acceso a luce fissa. Il led rosso, invece, si accende brevemente. Un telecomando ha trasmesso un comando valido e il radiocontrollo ha generato e trasferito tale comando alla rete Velbus.
	Rosso acceso a luce fissa: manca la codifica base. Si verifica all’accensione del dispositivo. L’apprendimento si effettua tramite l’indirizzo Velbus F0.

Tenete presente che il radiocontrollo emula la scheda avente indirizzo 05 sui canali 1÷8 e quella con indirizzo 06 sui canali 9÷16. Se invece state operando solo con un attuttore (il VMB4RY, comando a relé a quattro canali) potete effettuare l’accoppiamento “pulsante per canale” secondo la procedura descritta sul fascicolo 115. Ipotizzando di voler associare ognuno dei quattro pulsanti ad un canale operante in modalità “bistabile” (una pressione del pulsante attiva il relé e la successiva lo disattiva), impostate sui preselettori rotativi di indirizzo del VMB4RY il valore “D1”, con cui il canale 1 apprende quali dispositivi lo attivano in bistabile. Premete il pulsante del telecomando che volete associare al canale 1 e mantenete premuto fino a quando il relé non si attiva, segnalando

di avere appreso la programmazione. Ripetete la stessa procedura con gli altri tre canali, impostando sui preselettori i valori D2, D3 e D4. Una volta terminata la programmazione, reimpostate sul VMB4RY l’indirizzo Velbus e verificate che ogni pulsante del telecomando attivi e disattivi un solo relé.

Conclusioni

Abbiamo terminato la presentazione del nostro primo progetto pratico operante con dispositivi Velbus, progetto che, non lo nascondiamo, ci ha dato grande soddisfazione. Nelle prossime uscite presenteremo due nuovi progetti per interagire da remoto con la rete tramite messaggi SMS e connessione Bluetooth nonché, in anteprima, una serie di nuovi moduli Velbus tra i quali un controllo pulsanti con display.

per il

MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato monofaccia può essere scaricato gratuitamente dal sito della rivista (www.elettronica.in.it) così come il firmware implementato nel micro. Quest’ultimo è disponibile anche già programmato al prezzo di 18,00 Euro (cod. MF692). Il telecomando a 4 canali viene fornito già montato (cod. TX3750-4CS) al prezzo di 28,00 Euro. Tutti i prezzi sono comprensivi d’IVA.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futuranet.it>

Anti Jammer per teleallarme GSM

di Boris Landoni

A Rottweiler dog is shown from the chest up, holding a green printed circuit board (PCB) in its mouth. The PCB is a GSM alarm module, featuring a gold-colored SIM card slot with a white label that includes a barcode and the text 'CE0168'. The module has several electronic components, including capacitors and integrated circuits. The background is a blurred outdoor scene with a house and a car.

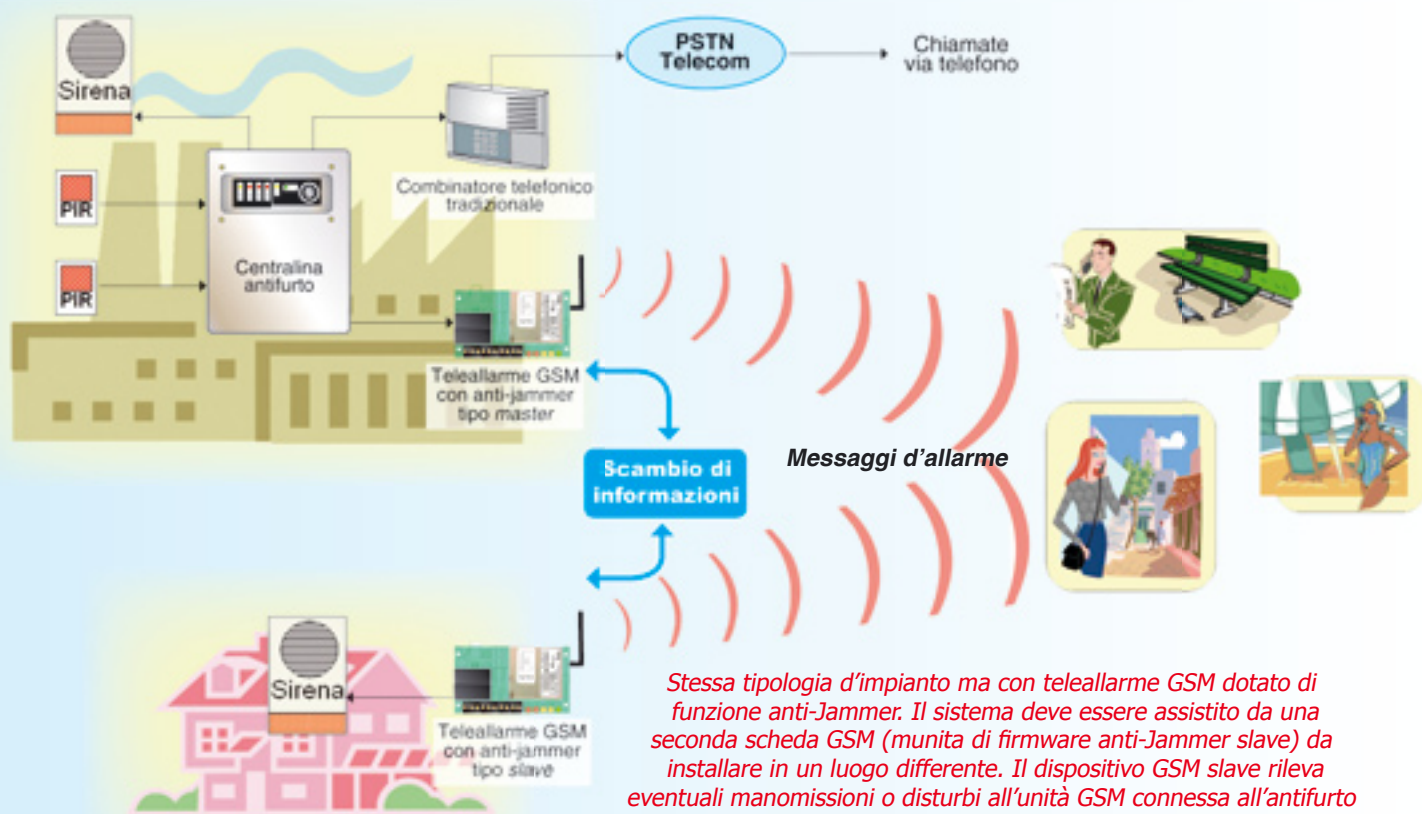
Teleallarme GSM, dotato di funzione anti-Jammer, da abbinare a qualsiasi impianto antifurto. Il sistema, composto da due unità GSM, garantisce la massima sicurezza verificando periodicamente il corretto funzionamento dell'impianto radio. In caso di manomissione o disturbo interviene una seconda unità GSM che invia i messaggi di allarme.

Da parecchi anni si è affermata la tendenza dei costruttori e installatori a dotare i propri sistemi anti intrusione di un'interfaccia di teleallarme: solitamente un combinatore telefonico per linea tradizionale o cellulare, capace di avvertire una o più persone, in caso di allarme, mediante chiamate voce o invio di SMS. Purtroppo i malviventi ("ladri di polli" a parte) sono tutt'altro che sprovveduti, e nel tempo hanno scoperto come fare per rendere inattive centraline d'allarme wireless e combinatori GSM; basta mettere in funzione, nelle loro vicinanze, un trasmettitore ra-

dio operante sulla gamma di frequenza alle quali essi funzionano: la sovrapposizione di una portante RF sufficientemente intensa disturba o copre la ricezione, quindi impedisce il corretto funzionamento degli apparati che, si dice, vengono "accecati" da onde radio intense. I dispositivi usati per questo scopo vengono detti Jammer. Fortunatamente esiste una maniera per aggirare l'ostacolo: adottare un sistema di interrogazione ciclica come quello descritto in queste pagine, composto da due moduli GSM identici ma programmati per funzionare in maniera differente, ossia uno



Sistema antifurto tradizionale con due combinatori telefonici, uno via filo a l'altro di tipo GSM. Un impianto del genere può essere messo fuori uso tagliando i fili del telefono e disturbando il GSM con un Jammer.



Stessa tipologia d'impianto ma con teleallarme GSM dotato di funzione anti-Jammer. Il sistema deve essere assistito da una seconda scheda GSM (munita di firmware anti-Jammer slave) da installare in un luogo differente. Il dispositivo GSM slave rileva eventuali manomissioni o disturbi all'unità GSM connessa all'antifurto inviando, in questo caso, i messaggi di allarme.

da master e l'altro da slave. Nel nostro caso, il master effettua delle chiamate verso lo slave che, dopo aver ricevuto alcuni squilli, le rifiuta. La rete GSM informa del rifiuto il master che, in questo modo, ha anche la conferma del buon esito delle chiamate: la situazione viene considerata normale fino a che lo slave riceve chiamate. Quest'ultimo comincia ad "insospettirsi"

quando, scaduto un certo intervallo di tempo, non riceve più chiamate; allora si porta in una situazione di preallarme e chiama automaticamente il master per verificare se va tutto bene: effettua alcuni squilli e poi interrompe la chiamata, quindi si dispone in attesa di una telefonata. Se il master non risponde entro un certo tempo, scatta l'allarme: il dispositivo slave attiva una delle

sue uscite a relé, con le quali può dare svariate segnalazioni locali, ovvero attivare l'ingresso di un altro sistema d'allarme o un segnalatore di eventi su una console di un istituto di vigilanza. Tuttavia, la cosa più importante è sicuramente costituita dal fatto che anche questa unità, così come quella collegata all'impianto di allarme, è in grado di effettuare chiamate o inviare SMS ➤

ai numeri telefonici memorizzati in un'apposita lista, per comunicare a distanza l'allarme a una o più persone. Ricapitolando, possiamo dire che il teleallarme GSM principale continua a funzionare in modo tradizionale, inviando i messaggi di allarme agli utenti memorizzati nel caso in cui l'antifurto entri in

tradizionale telecontrollo e teleallarme ma ha anche la possibilità di integrare questa funzione con la modalità anti-Jammer; in quest'ultimo caso è tuttavia necessario utilizzare due unità, una da adibire come master e l'altra come slave. Tutte le funzionalità possono essere definite mediante comandi di

so fra 5 e 32 V, che viene filtrata a valle del diodo di protezione (D1) mediante i condensatori C1 e C2; dalla differenza di potenziale ai capi di questi ultimi, il regolatore switching U1 ricava 3,6 V stabilizzati, con i quali alimenta tutto il resto del circuito. U1 è un MC34063 che incorpora un regolatore PWM

Jammer: il disturbatore



Il termine inglese "jam" assume significati differenti a seconda di come viene utilizzato: può voler dire "pasticcio, confusione, inceppamento" oppure "marmellata". In generale, il comune denominatore è rappresentato dal concetto di "pasticcio, disturbo". In pratica il jammer, simile al dispositivo visibile nella fotografia qui a sinistra, è in grado di generare dei segnali radio molto forti, l'effetto dei quali è appunto il disturbo ed il conseguente

inceppamento dei sistemi di telecomunicazione, quali ad esempio le reti telefoniche GSM e GPRS. Più in generale, un jammer rende impossibile l'utilizzo di tutti i sistemi di trasmissione le cui frequenze sono localizzate nella banda radio in cui opera il jammer stesso. Nell'articolo descriviamo un circuito in grado di prevenire le conseguenze di un disturbo indotto con intenzioni malevole a scopo di furto. Ma i Jammer possono anche essere molto utili come dimostra l'esempio riportato in questo riquadro: un convoglio di veicoli, in cui due o più di essi rappresentano la scorta ad



un diplomatico, un'autorità religiosa o un'alta carica di governo o militare. Che un veicolo e un convoglio in movimento su strada siano a rischio di attentato è abbastanza evidente (ricordiamo ciò che successe a Falcone e Borsellino). Se però uno dei veicoli, meglio se quello dell'autorità, fosse dotato di un dispositivo di jamming, eventuali bombe comandate a distanza tramite cellulare (o comunque via radio) verrebbero inibite al passaggio del convoglio evitando tragiche conseguenze.



funzione; in più esso comunica continuamente (senza addebito di traffico telefonico) con una seconda scheda GSM installata in un luogo differente (magari la vostra abitazione se l'impianto principale si trova in un capannone o in un negozio). Se l'unità GSM connessa all'impianto antifurto viene messa fuori uso o disturbata dall'esterno, il "dialogo" con l'unità slave si interrompe e quest'ultima invia i messaggi di allarme.

L'unità master/slave

La scheda GSM da noi realizzata è in grado di funzionare come un

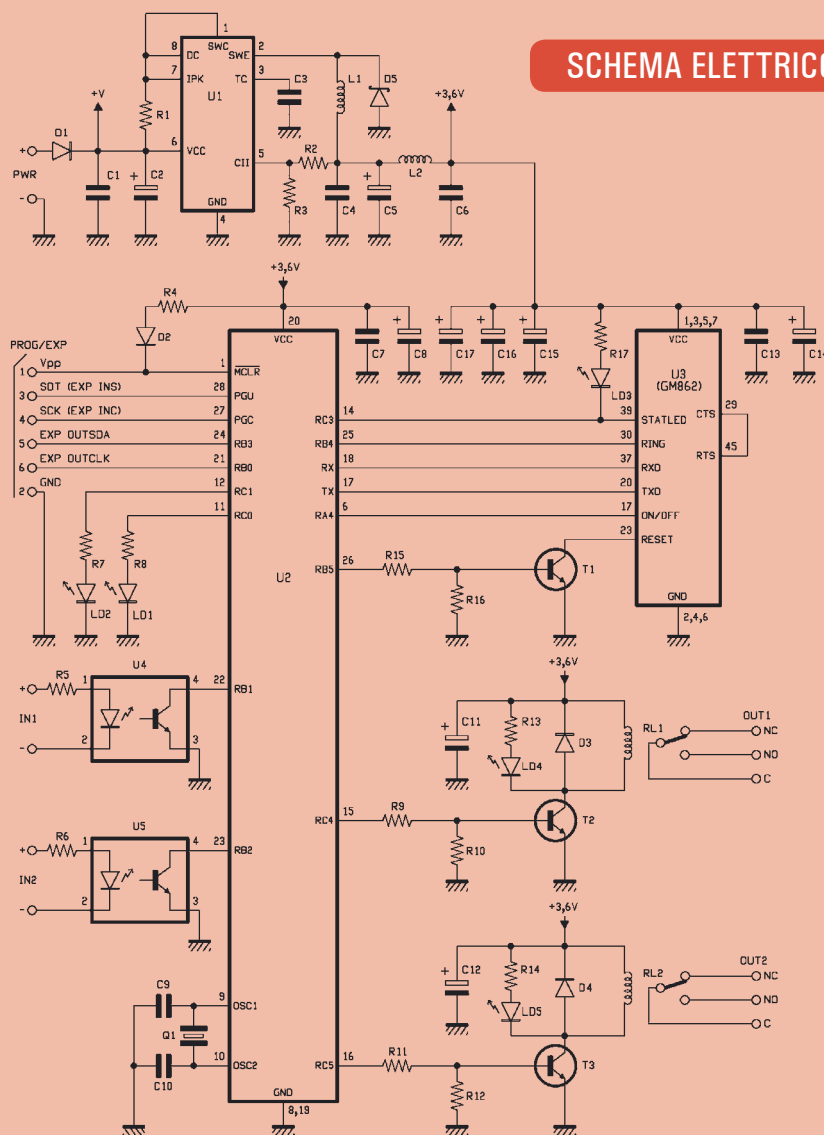
configurazione inviati tramite SMS da un telefono cellulare.

Prima di vedere come funziona il modulo nelle due modalità, analizziamone lo schema elettrico: il circuito utilizzato è molto simile al telecontrollo GSM cod. TDG33 presentato nel fascicolo 105 del febbraio 2006. In questo caso, tuttavia, cambia completamente il firmware, nel senso che le funzioni tradizionali sono integrate con la modalità anti-jammer (master o slave).

Lo schema elettrico è visibile nella pagina accanto. Il circuito funziona con una tensione continua, anche non stabilizzata, di valore compre-

di tipo step-down, il cui transistor d'uscita carica l'induttanza L1 con impulsi la cui larghezza dipende dalla tensione livellata da C4 e C5 e retrocessa dal partitore resistivo R2/R3: più la tensione scende per effetto del carico, più si allargano gli impulsi e viceversa. La retroazione serve, insomma, a stabilizzare la tensione d'uscita, che, con gli attuali valori di R2 ed R3, ammonta a 3,6 volt. Gli impulsi vengono ottenuti da un modulatore PWM interno che pilota la base di un transistor NPN (sempre interno al chip) il cui emettitore eroga impulsi di corrente che attraversano L1

SCHEMA ELETTRICO



di 0,5 secondi seguito da una pausa di 2 secondi quando lo stesso ha agganciato il segnale.

Dalla frequenza e durata degli impulsi con cui il Telit fa pulsare il led di "campo", il PIC18F2620 comprende le condizioni di lavoro; più esattamente, se rileva che il cellulare è fuori campo, riavvia il modulo Telit in modo che si riconnetta alla rete GSM/GPRS.

Per comunicare con il modulo radio, il micro utilizza la UART di cui è provvisto. Essa è accessibile dai pin 17 (trasmissione, TX) e 18 (ricezione, RX). Mediante il TX, il PIC esegue due operazioni verso il modulo Telit: lo interroga ciclicamente per verificare l'arrivo di SMS o segnalazioni di centrale ("numero occupato", "chiamata rifiutata...") e lo istruisce per effettuare le chiamate in caso di allarme. Relativamente a queste ultime, indipendentemente dalla modalità in cui il circuito opera, se il PIC rileva che la chiamata non va a buon fine, ritenta per tre volte.

Come accennato, nella modalità teleallarme il nostro apparato permette di monitorare lo stato dei due ingressi (isolati galvanicamente dal micro tramite i fotoaccoppiatori U4 e U5) in grado di rilevare come stato logico attivo tensioni di valore compreso fra 3 e 30 V.

L'isolamento galvanico è molto importante in quanto protegge il sistema da rotture causate da differenze di potenziale tra massa del circuito e sensore/uscita da controllare. Spesso, infatti, gli ingressi sono collegati a sensori tramite cavi molto lunghi che possono sia captare forti disturbi di alternata sia spostare sensibilmente il potenziale di "massa".

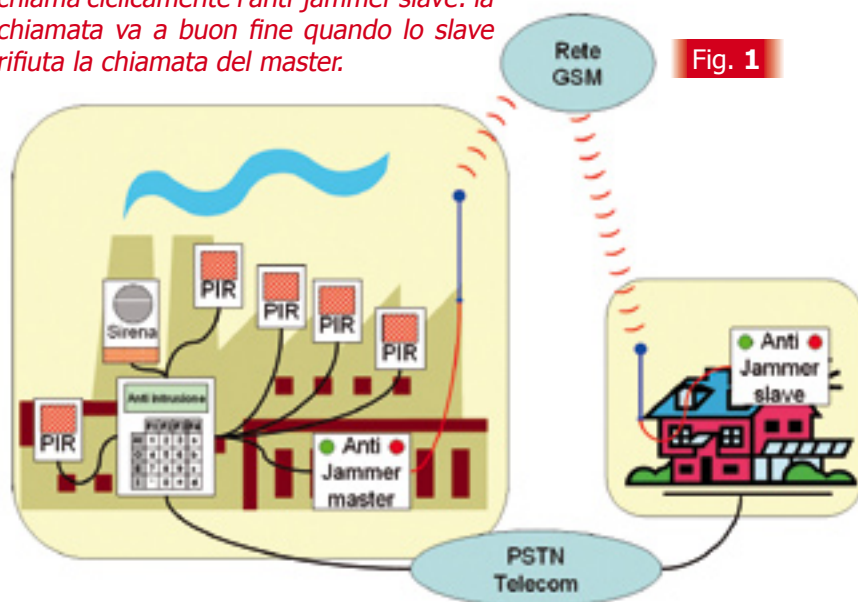
Solitamente ad uno di questi ingressi va collegata l'uscita dell'impianto antifurto o antintrusione la cui attivazione determina l'invio da parte del dispositivo GSM delle chiamate di allarme.

dirigendosi verso C4 e C5; quando il transistor è interdetto, l'induttore restituisce l'energia immagazzinata forzando lo scorrimento di corrente nei condensatori attraverso il diodo Schottky D5. Quest'ultimo serve anche per proteggere il transistor interno all'integrato che, altrimenti, verrebbe danneggiato dall'extratensione inversa generata dalla L1 nel momento dell'interdizione. I condensatori di filtro posti sulla linea dei 3,6 volt servono a sopprimere i disturbi originati dalla sezione RF del modulo cellulare quando trasmette. La gestione dell'intero telecontrollo è affidata

al microcontrollore PIC18F2620 Microchip che, inizializzate le linee di I/O, provvede a monitorare la condizione logica degli ingressi a livello di tensione (RB1 ed RB2) ma anche le linee RB4, RC3, RX, che servono a ricevere dal modulo cellulare le principali segnalazioni; precisamente, RB4 viene usata per rilevare l'arrivo delle chiamate entranti mentre RC3 controlla il led di stato collegato all'uscita STATLED del GM862. Tale uscita pulsa alla frequenza di 1 Hz quando il cellulare sta cercando la rete radiomobile, mentre fornisce un impulso a zero logico della durata

Fase 1.

Nessun allarme. L'antifurto ha a disposizione la linea telefonica cablata e la rete GSM. L'anti-jammer master, che funziona anche come teleallarme GSM, chiama ciclicamente l'anti-jammer slave: la chiamata va a buon fine quando lo slave rifiuta la chiamata del master.

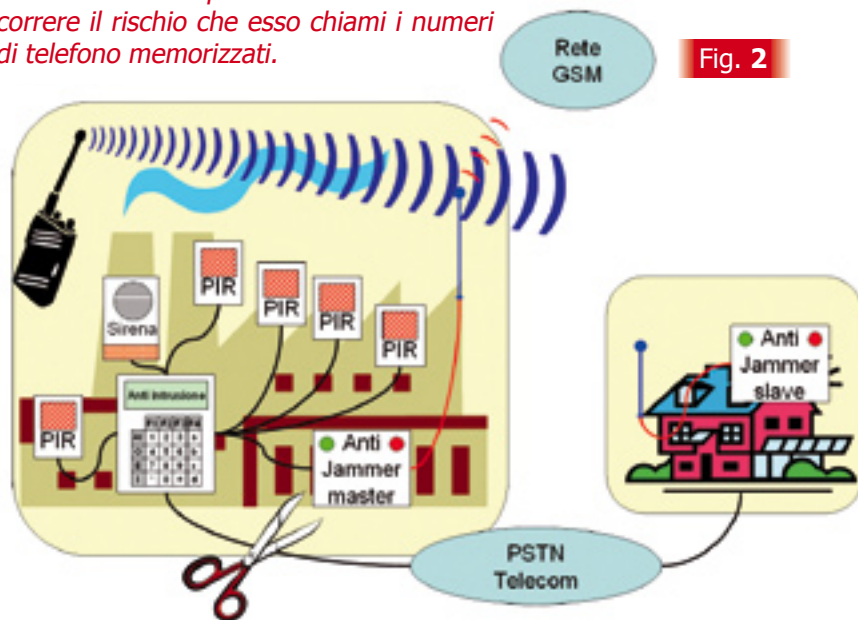


reso disponibile l'intero scambio, così da poter gestire sia circuiti che richiedono un contatto normalmente chiuso, sia che abbiano bisogno di un contatto normalmente aperto. Rimangono da descrivere le linee /MCLR, PGU, PGC, RB0 ed RB3: esse servono sia per la programmazione in-circuit del PIC18F2620 che per l'eventuale aggiunta di schede di espansione. Più esattamente, PGU e PGC sono, rispettivamente, il Serial Data (SDA) e il Serial Clock (SCL) del bus I²C per la comunicazione con il programmatore, mentre RB3 e RB0 realizzano un bus di espansione (attualmente non utilizzato); tutte le linee sono riferite alla massa comune GND. Bene, spiegato lo schema elettrico

Gli ingressi vengono considerati attivi quando ad uno di essi è applicata una tensione di almeno 3 volt e quindi il fototransistor interno al fotoaccoppiatore è in conduzione; in tal caso il piedino 5 è a circa zero volt e così il piedino corrispondente del micro; sono ritenuti a riposo nel caso contrario, ossia in assenza di tensione o quando questa non superi la soglia di conduzione dell'optoisolatore; in tal caso il relativo I/O del microcontrollore, per effetto del resistore di pull-up configurato durante l'inizializzazione, è forzato a livello alto. LD1 ed LD2 segnalano l'attivazione, rispettivamente degli ingressi 1 e 2; però, a questo punto bisogna fare una precisazione: per attivazione si intende il raggiungimento, sulle linee RB1 ed RB2, delle condizioni logiche definite in fase di configurazione del telecomando, mediante apposite istruzioni da SMS. In altre parole, per il sistema un ingresso può essere attivo quando riceve una tensione superiore a quella di soglia (3 V) o, viceversa, quando non viene polarizzato. Per il comando di utilizzatori ester-

Fase 2.

I ladri tagliano la linea telefonica e disturbano la trasmissione GSM con un dispositivo jammer. Il sistema antifurto è isolato: i ladri hanno il tempo di manomettere l'impianto di allarme senza correre il rischio che esso chiami i numeri di telefono memorizzati.



ni, il microcontrollore si avvale dei relé RL1 e RL2 (alimentati a 3,6 V), pilotati mediante le linee RC4/RC5 ed i transistor T2/T3, usati come amplificatori di corrente. I due led LD4/LD5 collegati in parallelo alle bobine di RL1 e RL2, servono per segnalare l'attivazione del relè relativi. Delle uscite di RL1/RL2 viene

andiamo ad analizzare una ad una le modalità di funzionamento del nostro apparato.

Funzione anti-jammer

Per realizzarla occorrono due unità: una impostata come master e l'altra funzionante da slave. Il master chiama lo slave sia ad intervalli re-

L'impostazione del tipo di periferica si effettua mediante l'in-

Non avendo ricevuto risposta dopo il sollecito di chiamata, l'anti-jammer slave va in allarme quindi attiva un relé (con cui pilotare ad esempio una sirena) ed invia un SMS o una chiamata voce a tutti i numeri di telefono memorizzati, per informare di aver perso i contatti con l'impianto antifurto al quale è collegato il master.



Fig. 4

L'anti-jammer slave non riceve più chiamate dal master, pertanto va in preallarme effettuando una chiamata verso di esso e rimanendo in attesa di una sua risposta. Essendo questi disturbato dal jammer, non può ricevere la chiamata dello slave e quindi nemmeno rispondere.

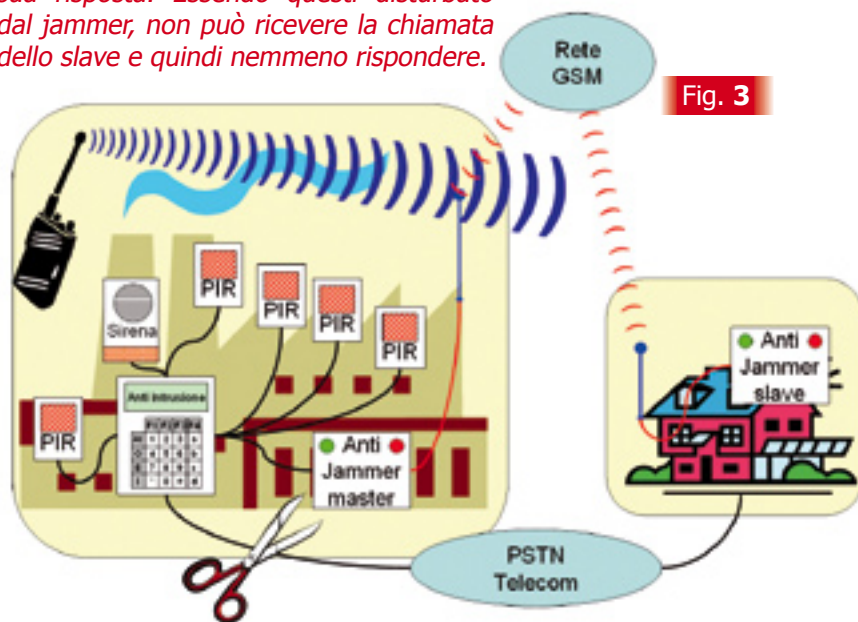


Fig. 3

di accesso al sistema (per impostazione iniziale o dopo un reset totale è 12345) senza la quale non sarà possibile operare alcuna impostazione. In ogni momento, con un apposito comando è possibile interrogare il modulo per sapere se è master o slave: basta inviare al suo numero telefonico un SMS conte-

Bene, detto ciò analizzando il comportamento, nella funzione anti-jammer, del dispositivo master, con una premessa: il numero telefonico corrispondente alla SIM Card inserita nell'unità slave deve essere sempre memorizzato in prima posizione; le modalità di inserimento dei numeri verranno descritte più avanti, a proposito delle funzioni di teleallarme.

Dopo l'accensione l'unità effettua, periodicamente e con cadenza definita da un'apposita configurazione, telefonate dirette al numero telefonico che ha nella prima posizione della sua lista di otto (vedere Fig. 5); ciò significa che bisogna memorizzare in posizione 1 il numero dell'unità slave. Durante le chiamate non trasmette alcunché, ma si limita ad attendere qualche istante e ad accendere LD1; a questo punto aspetta che lo slave respinga la telefonata e mandi il segnale di occupato, allorché spegne LD1 e fa illuminare LD2 per un istante. Va detto che il dispositivo master telefona

Diagramma di flusso modulo master

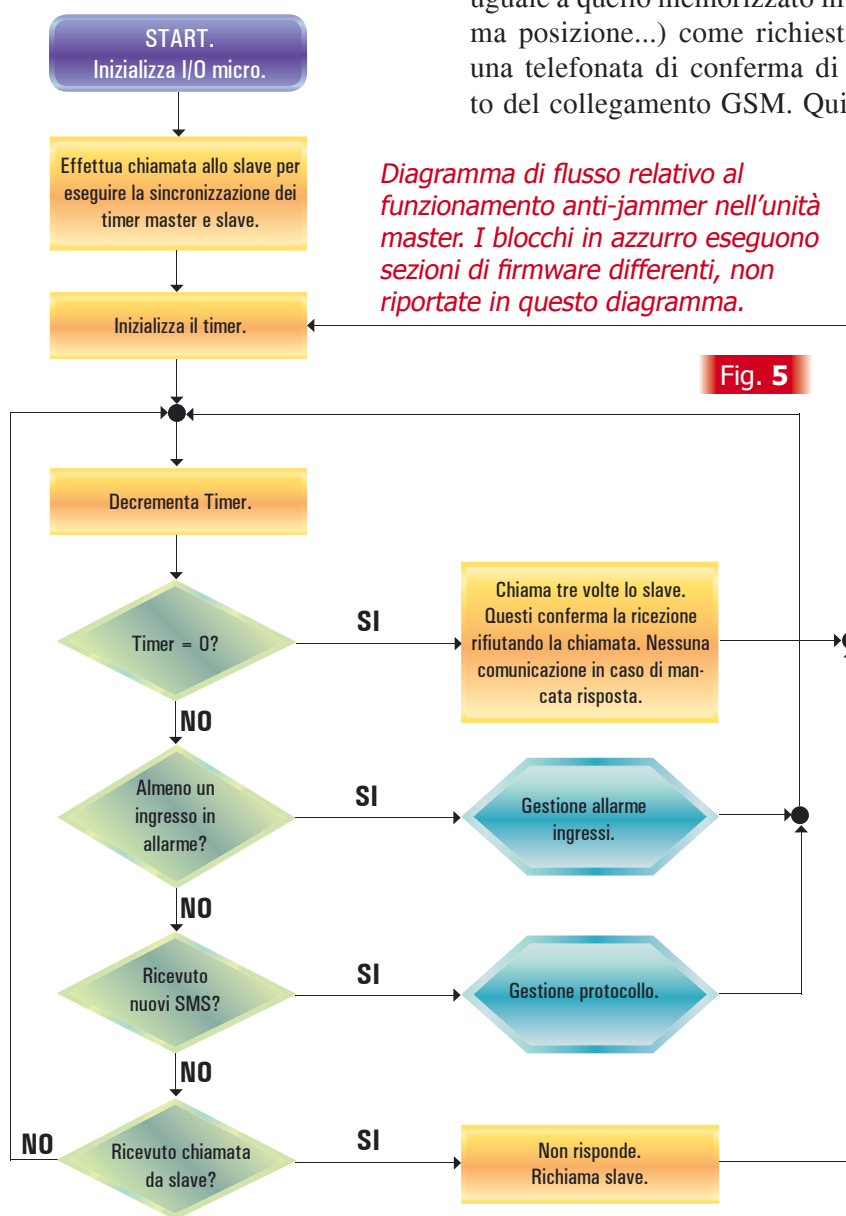


Diagramma di flusso relativo al funzionamento anti-jammer nell'unità master. I blocchi in azzurro eseguono sezioni di firmware differenti, non riportate in questo diagramma.

Fig. 5

che gestisce l'unità interpreta l'arrivo di una chiamata dallo slave (lo riconosce perché ha un numero uguale a quello memorizzato in prima posizione...) come richiesta di una telefonata di conferma di stato del collegamento GSM. Quindi,

un SMS contenente il testo: *ATT:tt;attpwd*. In esso *tt* sta per il tempo, espresso in minuti, compreso tra 01 e 59, mentre *attpwd* è la password attualmente in vigore. Con un messaggio analogo è possibile richiamare il periodo impostato qualora lo si sia dimenticato; il testo del messaggio è *ATT?*. Può essere necessario effettuare tale richiesta se ad esempio si sostituisce l'unità slave o semplicemente la si deve riconfigurare molto tempo dopo la configurazione precedente. Sape- te... le macchine ricordano sempre mentre le persone a volte dimenticano, specialmente date e numeri...

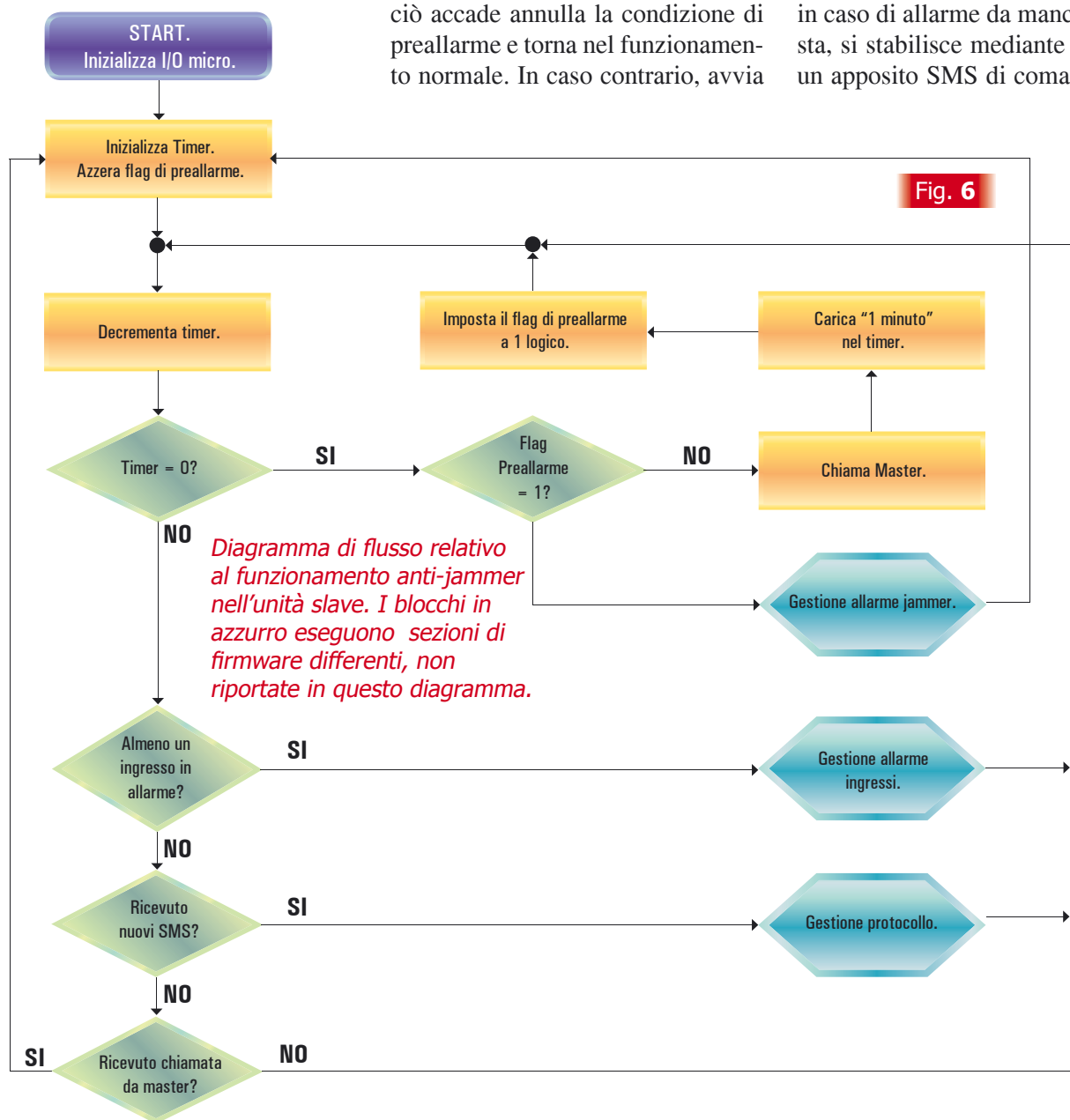
Come funziona lo slave

Vediamo adesso il funzionamento del modulo configurato come slave (Fig. 6), con una premessa: le chiamate provenienti dal master sono, per definizione, quelle in arrivo dal numero memorizzato in prima posizione nella lista di otto indicativi. Quindi, affinché possano essere riconosciute, lo slave deve essere programmato in modo che nella prima posizione vi sia il numero del master. Ad ogni accensione, l'unità slave effettua una telefonata diretta al master e attende un paio di squilli; poi riaggancia e si dispone ad aspettare che questo risponda con una propria chiamata, l'arrivo della quale viene segnalato localmente dall'illuminazione simultanea dei led LD1 ed LD2 (vedere tabella 2). Questi si spengono quando l'unità slave respinge la chiamata, ossia al termine della telefonata da parte della master. A regime, il microcontrollore del modulo funzionante da slave avvia un timer, la cui durata è definibile dall'utente con un SMS analogo a quello visto per il modulo master: *ATT:tt;attpwd*. In esso *tt* sta per il tempo, espresso in minuti, compreso tra 01 e 59, mentre *attpwd* è la solita password. Con un messaggio analogo è possibile variare il periodo attualmente impostato;

non solo periodicamente, ma anche dopo ogni accensione: ciò per bloccare eventuali sequenze di allarme partite sullo slave qualora, causa un black-out (riguardante il master) per qualche tempo non sia riuscito ad effettuare la chiamata periodica. Oltre a ciò, il modulo master è sempre pronto a ricevere chiamate dallo slave alle quali non risponde, limitandosi ad attendere che esso faccia il canonico squillo e poi riagganci; quando ciò accade, viene acceso per un istante LD2. Il microcontrollore

appena liberata la linea, compone il numero telefonico dell'unità slave e fa squillare un paio di volte la suoneria per confermare che tutto va bene. Al solito, accende LD1 e lo spegne, facendo illuminare LD2 quando lo slave respinge la chiamata. La tabella 2, riportata a pagina 53, riassume tutte le possibili segnalazioni luminose relative alla modalità anti-jammer. L'intervallo delle chiamate automatiche periodiche può essere definito inviando al modulo master

Diagramma di flusso modulo slave



cia. A questo punto aspetta che, entro un minuto, il master lo chiami; se ciò accade annulla la condizione di preallarme e torna nel funzionamento normale. In caso contrario, avvia

contenente il testo *OUT1:OFF*.

Il modo in cui RL1 deve funzionare in caso di allarme da mancata risposta, si stabilisce mediante l'invio di un apposito SMS di comando il cui

tramite il comando *ATT?* è possibile richiedere un SMS che contiene il tempo di attesa impostato. L'intervallo deve essere sempre superiore a quello definito nel master, in modo che lo slave attenda un po', prima di preoccuparsi. Se, trascorso l'intervallo impostato, lo slave non riceve la chiamata dal master, esso entra in preallarme: effettua una chiamata al numero memorizzato in prima posizione nella lista, lascia trascorrere qualche istante (quel che serve a fare un paio di squilli) poi riaggan-

la sequenza di allarme: effettua le chiamate ai numeri in posizione da 2 a 7 della lista e invia agli stessi numeri (in alternativa) dei messaggi contenenti il testo memorizzato tramite l'apposito comando (sempre via SMS).

Localmente, attiva il relé 1 secondo la modalità definita con l'apposita programmazione, modalità che può essere bistabile o impulsiva; nel primo caso, il relé può essere disattivato solo manualmente dall'utente, che deve inviare al modulo un SMS

testo è *TAC:ss*, dove al posto di *ss* va scritto l'intervallo, che può essere scelto fra 00 e 59 secondi; se il valore è 00, RL1 lavorerà in modalità bistabile, mentre se si sceglie un tempo tra 01 e 59 s. lo stesso funzionerà ad impulso e, a seguito di ogni attivazione in caso di allarme, scaduto l'intervallo impostato tornerà a riposo da solo.

Infine, riguardo ad RL1, si noti che quando è attivo in modo bistabile, con un apposito comando SMS è possibile disattivarlo a tempo, se-

condo la modalità di inversione di stato disponibile nella funzione telecontrollo descritta di seguito.

L'utilizzo da telecontrollo

Sebbene non sia l'uso principe, il dispositivo master/slave può anche funzionare da telecontrollo, dato che incorpora un set di funzioni preposte allo scopo; in particolare per l'esattezza, può essere interrogato a distanza mediante comandi SMS per acquisire lo stato degli ingressi e le impostazioni vigenti. Inoltre, sempre su comando da messaggi di testo, può impostare come richiesto le due uscite a relé di cui dispone.

All'attività degli input possono essere associate azioni locali o remote, quali l'invio di chiamate o SMS diretti ai numeri della lista quando uno o entrambi divengono attivi, intendendo con attivi che raggiungono lo stato definito in fase di impostazione. Riguardo a tale funzione occorre fare una precisazione: vengono avvertiti tramite telefonata o SMS i numeri dalla seconda alla settima posizione; la prima è riservata alla funzione anti-jammer.

Durante ogni chiamata, il circuito non trasmette alcunché, ma si limita ad effettuare qualche squillo al telefono della persona chiamata: scopo di ciò è richiamare l'attenzione e segnalare il verificarsi di un'anomalia con maggiore tempestività rispetto all'SMS; non va infatti dimenticato che i messaggi di testo non sempre vengono recapitati in tempo reale.

Sempre in tema di attività degli ingressi, è possibile definire un intervallo di osservazione, ossia stabilire per quanto tempo la situazione di attivazione deve permanere prima di essere considerata allarme e determinare l'invio del messaggio di notifica e l'effettuazione della chiamata ai numeri memorizzati; ciò si imposta ingresso per ingresso.

I relé di uscita possono essere attivati in modo sia impulsivo che

L'anti-jammer qui descritto realizza una connessione GSM punto-punto tra due dispositivi che si tengono in contatto con chiamate periodiche: la perdita di comunicazione viene considerata come situazione di allarme. In tal modo, se al master viene impedito di chiamare, lo slave inizia ad allertare tutti i numeri di telefono memorizzati.

Per mantenere la comunicazione, il master chiama lo slave, e quest'ultimo rifiuta la chiamata. La centrale GSM, quindi, informa il master che la chiamata non è stata accettata, situazione considerata come "comunicazione con esito positivo", dato che solo lo slave avrebbe potuto rifiutare la chiamata. Poiché non viene stabilita alcuna connessione, questa procedura ha costo di gestione zero, a patto che nella SIM del GSM slave siano disabilitate tutte le deviazioni di chiamata ad altro numero o segreteria telefonica.

bistabile: il comando è unico e per ottenere le due modalità basta definire opportunamente il parametro tempo; se si invia un SMS per forzare l'attivazione monostabile di un relé già attivato in modo bistabile, il comando ha l'effetto di far tornare a riposo il relé stesso per il tempo definito nel messaggio. Scaduto l'intervallo, il relé torna eccitato, ossia ripristina la modalità bistabile, modalità che può essere disattivata solo con un SMS che contenga la disattivazione bistabile, oppure privando il circuito dell'alimentazione quando non sia attivata la funzione di ripristino dopo black-out; quest'ultima, alla successiva accensione riporta i relé nelle condizioni in cui si trovavano prima dell'evento. Resta inteso che abilitando tale funzione, un relé attivo in modo bistabile può essere interdetto solo con il comando di disattivazione.

Per gestire le uscite a relé il circuito deve ricevere SMS, contenenti specifici comandi, da numeri telefonici abilitati a farlo (quelli memorizzati nell'apposita lista, eccetto il primo).

Quanto ai comandi SMS di configurazione del sistema, possono essere impartiti in due modi: da un qualsiasi cellulare, a patto di inserire la password richiesta (attpwd); da un telefonino il cui numero sia già presente in lista, eccezion fatta per alcune impostazioni che richiedono entrambe le condizioni, ossia che gli SMS siano inviati da un te-

lefono presente in lista e corredato dalla password.

Comandi di sistema

Passiamo dunque alla descrizione dei comandi e alle relative sintassi, con la premessa che il sistema può accettare SMS multipli, ossia contenenti più di un comando o comandi riguardanti uno o più numeri telefonici; i comandi devono essere separati ognuno da quello successivo con una virgola. Premettiamo altresì che tutti i comandi per i quali non è espressamente prevista la password, hanno effetto solamente se provengono da un telefono riconosciuto, ossia il cui numero sia nella lista di quelli memorizzati nel dispositivo. Il fatto che un comando possa essere inviato senza password va inteso nel senso che lo può inviare un telefono della lista; uno estraneo deve metterla comunque. Il primo comando che esaminiamo è quello che riguarda la modifica della password e consta di un SMS del tipo *PWDxxxxx;attpwd*, dove al posto di *xxxxx* si deve scrivere la nuova password (numerica, di cinque cifre...) e *attpwd* indica la password attuale. A tale riguardo facciamo notare che la password predefinita nel microcontrollore del circuito è 12345.

La memorizzazione di uno degli otto numeri si effettua inviando un SMS contenente il testo *NUMx+n nnnnnnnnnnnn;attpwd*, dove al posto di *x* deve essere scritto quale dei numeri (posizione) si sta memoriz-

Tabella 1

FUNZIONE	COMANDO SMS	VALORE DI DEFAULT	PWD OBBLIGATORIA
PASSWORD		12345	
CAMBIO PASSWORD	PWDxxxxx; 12345	12345	*
MEMORIZZARE UN NUMERO (massimo 8 numeri) (MAX 19 CARATTERI PER NUMERO)	NUMx + 393355760937; 12345	.	SE LA POSIZIONE x NON È VUOTA
CANCELLARE UN NUMERO	NUMx; 12345	-	*
VERIFICARE I NUMERI MEMORIZZATI	NUM?; 12345	-	*
RESET COMPLETO	RES	-	*
IMPOSTARE I NUMERI A CUI VERRANNO INVIATI GLI SMS	SMSxxxxxxx: ON/OFF	tutti	*
IMPOSTARE I NUMERI A CUI VERRÀ FATTO UNO SQUILLO	VOCxxxxxxx: ON/OFF	tutti	*
LIVELLO LOGICO DI ALLARME ALTO - PRESENZA TENSIONE IN INGRESSO (x Può VALERE 1 O 2)	LIVx:A	ALTO	
LIVELLO LOGICO DI ALLARME BASSO - ASSENZA TENSIONE IN INGRESSO (x Può VALERE 1 O 2)	LIVx:B	ALTO	
LIVELLO LOGICO DI ALLARME VARIAZIONE	LIVx:V	STATO INIZIALE	
RICHIESTA LIVELLO	LIV?		
TEMPO INIBIZIONE INGRESSO1 (DA 00 A 59 MIN)	INI1:mm	5	
TEMPO INIBIZIONE INGRESSO2 (DA 00 A 59 MIN)	INI2:mm	5	
INTERROGA IL TEMPO DI INIBIZIONE	INI?		
ATTIVAZIONE RELÉ IN MODALITÀ BISTABILE	OUTx:ON	OFF	
DISATTIVAZIONE RELÉ IN MODALITÀ BISTABILE	OUTx:OFF	OFF	
CAMBIO STATO RELÉ MONOSTABILE (TEMPO IN SECONDI DA 01 ÷ 59)	OUTx:ss	OFF	
RIPRISTINO RELÉ x VALE 1 PER AVERE IL RIPRISTINO O PER AVERE I RELÉ DISATTIVATI	RIPx	1	
INTERROGAZIONE RIPRISTINO	RIP?	0	
RICHIESTA STATO INGRESSI E USCITE	STA?		
TEMPO DI ATTIVAZIONE RELÉ1 IN CASO DI ALLARME DA 00 (BISTABILE) ÷ 59 SECONDI	TAC:ss	5	
TESTO ALLARME INGRESSO1 QUANDO È PRESENTE TENSIONE MAX 100 CARATTERI	TIN1A:xxxxxxxxxx	ALLARME 1 ALTO	NO MULTI
TESTO ALLARME INGRESSO1 QUANDO È ASSENTE TENSIONE MAX 100 CARATTERI	TIN1B:xxxxxxxxxx	ALLARME 1 BASSO	NO MULTI
TESTO ALLARME INGRESSO2 QUANDO È PRESENTE TENSIONE MAX 100 CARATTERI	TIN2A:xxxxxxxxxx	ALLARME 2 ALTO	NO MULTI
TESTO ALLARME INGRESSO1 QUANDO È ASSENTE TENSIONE MAX 100 CARATTERI	TIN2B:xxxxxxxxxx	ALLARME 2 BASSO	NO MULTI
TESTO MESSAGGIO MANCATA RICEZIONE SQUILLO MASTER MAX 100 CARATTERI	TAL:xxxxxxxxxx	ALLARME MASTER	NO MULTI
IMPOSTA SE SI TRATTA DI UN DISPOSITIVO MASTER (0) O SLAVE (1)	DEV:0	0	
RICHIEDI SE SI TRATTA DI UN MASTER O SLAVE	DEV:?	0	
IMPOSTA IL TEMPO DI ATTESA TRA UNA CHIAMATA E L'ALTRA 01 ÷ 59 MIN	ATT:05	50	
RICHIEDI IL TEMPO DI ATTESA	ATT?	50	
DISABILITA LA RISPOSTA PER QUEL MULTIMESSAGGIO	RISP	-	

zando, al posto delle *n* va il numero e *attpwd* è l'attuale password. Il tutto va scritto senza spazi. Sono ammessi numeri di 19 cifre; notate che il + è il simbolo che, nel prefisso internazionale composto dai cellulari, sostituisce lo 00. Per esempio, l'inserimento del numero 003911512000 in ottava posizione

si opera con un comando del tipo: *NUM8+3911512000;attpwd*.

Va notato che per memorizzare un numero è necessario introdurre la password solamente quando si tenta di farlo in una posizione già occupata da un altro; invece, se si deve inserire un numero in una posizione vuota della lista basta inviare un

SMS con *NUMx+nnnnnnnnnnnnnnnnnnnn*. Ovviamente quanto appena detto vale nel caso in cui il comando venga inviato da un telefono il cui numero sia già stato fatto apprendere al telecontrollo, altrimenti la password è obbligatoria.

La rimozione di un numero si effettua con un SMS che contiene il ➤

Riassunto configurazioni per master e slave

Il sistema va configurato in base all'uso cui si intende destinarlo: nella funzione anti-jammer, ogni unità deve essere programmata almeno per definire i numeri da chiamare; sullo slave, occorre anche definire l'attività del relé 1. Nell'uso come telecomando, basta un modulo, mentre

per realizzare la funzione anti-jammer è necessario utilizzare due moduli, di cui uno funzionante come master e l'altro come slave. La seguente tabella illustra le configurazioni minime da impostare in funzione dell'impiego cui sono destinate le apparecchiature.

Tabella 1

UNITÀ	Anti-jammer	TELECONTROLLO
Master	n° telefonico dell'unità Slave e intervallo di chiamata	Tutte le altre impostazioni, escluse quelle dell'anti-jammer (numero in prima posizione, intervallo d'attesa, testo SMS di allarme per mancata ricezione)
Slave	n° master e intervallo di attesa, numeri da chiamare e testi SMS d'allarme, eventuale tempo e modo di attivazione di RL1	Tutte le altre impostazioni, escluse quelle dell'anti-jammer (numero in prima posizione, intervallo d'attesa, testo SMS di allarme per mancata ricezione chiamata da master)

testo *NUMx;attpwd*; al posto della x va scritto il numero della posizione nella quale si trova l'indicativo da cancellare, mentre *attpwd* è la solita password. Ad esempio, per rimuovere dalla lista memorizzata il quarto numero, occorre inviare un messaggio che contiene *NUM4;attpwd*.

Per richiedere la lista dei numeri attualmente memorizzati nel telecomando occorre inviare un SMS con il seguente testo: *NUM?;attpwd*. Il circuito risponde al numero del telefono dal quale proviene l'interrogazione. Vediamo adesso i comandi relativi alla lista dei numeri cui, sia in anti-jammer che teleallarme, il dispositivo invia i messaggi di allerta. Abbiamo già detto che le segnalazioni possono essere effettuate inviando degli SMS. Il comando *SMSxxxxxxx:yy* consente di scegliere a quali numeri memorizzati deve essere inviato un messaggio in caso di necessità. La serie di x rappresenta la posizione di memoria occupata dai numeri (quindi 1 per il primo, 2 per il secondo e così via fino all'ottavo) mentre yy può assumere valore ON (per abilitare l'SMS sui numeri prescelti) oppure OFF (per disattivare l'emissione dell'SMS). Ad esempio, se volessimo annullare l'emissione degli SMS a tutti i numeri tranne che al quinto, senza richiedere un messaggio di conferma, potremmo inviare il comando *SMS12345678:OFF,SMS5:*

ON,RISP. Oltre che con gli SMS, il telecomando può allertare le persone anche con degli squilli; scopo dello squillo è richiamare l'attenzione sul fatto che si è verificato un evento e ciò più rapidamente di quanto non si otterrebbe dagli SMS, i quali possono giungere con un certo ritardo. Il comando per configurare i numeri che devono ricevere lo squillo è molto simile a quello visto per gli SMS:

VOCxxxxxxx:yy

Anche in questo caso, al posto delle x devono essere scritte le posizioni occupate dai numeri memorizzati (1 per il primo, 2 per il secondo...) mentre le yy possono assumere ancora valore ON o OFF a seconda che si vogliano attivare o disattivare chiamate vocali specifiche per ogni numero. Applicando l'esempio precedente alle chiamate vocali otteniamo il comando

VOC12345678:OFF,VOC5:ON,RISP

che disattiva la chiamata vocale su tutti i numeri tranne il quinto della rubrica, e ancora una volta non richiede l'SMS di conferma per il comando eseguito.

Per impostazione predefinita, il software prevede che tutti i numeri telefonici nelle posizioni da 2 a 7 ricevano la notifica di allarme ingressi, sia tramite SMS che mediante breve chiamata vocale, impostazione che viene anche ripristinata con il comando di reset totale del sistema. La posizione 1,

invece, è destinata alla funzione anti-jammer: il master memorizza in posizione 1 il numero dello slave e viceversa. Questa condizione è obbligatoria per il funzionamento in tale modalità.

Restando in tema di ingressi del telecomando, il comando *LIVx:y* permette di definire il livello da considerare come allarme. La x definisce l'ingresso a cui si riferisce il comando (1 per il primo ingresso e 2 per il secondo) mentre la y può assumere i valori A (la condizione di allarme è ALTO, presenza di tensione), B (la condizione di allarme è BASSO, mancanza di tensione) e V (la condizione di allarme è una VARIAZIONE dello stato dell'ingresso). Ad esempio:

- *LIV1:A*, l'input 1 è considerato in allarme in presenza di tensione;
- *LIV2:B*, l'input 2 è considerato in allarme in mancanza di tensione;
- *LIV1:V*, l'input 1 è considerato in allarme quando cambia di stato.

Se non ci si ricorda come sono impostati i due input, basta interrogare l'unità mediante il comando *LIV?*; essa risponde con un SMS contenente l'impostazione attuale.

Appena programmato, il microcontrollore, per impostazione predefinita, prevede che gli ingressi siano attivi in presenza di tensione (livello logico ALTO).

Per quel che riguarda il rilevamento dell'attivazione degli ingressi, è possibile definire un periodo di inibizione dal momento in cui si

verifica un'attivazione, prima che se ne possa leggere un'altra. Il tempo può essere impostato tra 0 e 59 minuti, con i messaggi: *INI1:mm* per l'input 1 e *INI2:mm* per l'input 2. Ad esempio, volendo stabilire che, a seguito di allarme, l'IN1 non possa determinare altri allarmi per due minuti, si deve configurare il telecomando inviandogli un SMS contenente *INI1:02*. Se non ci si ricorda l'impostazione degli ingressi riguardo al tempo di inibizione, con il messaggio *INI?* è possibile richiederlo; il telecomando risponde inviando, al cellulare che ha mandato l'SMS, un messaggio contenente l'attuale impostazione.

Quando agli ingressi sono collegati sensori che monitorizzano fenomeni frequentemente variabili, può essere necessario ignorare il tempo di inibizione, così da dare all'utente remoto avvisi realistici circa gli avvenimenti in corso; il nostro telecomando prevede la possibilità di disattivare, ingresso per ingresso, il tempo di inibizione impostato; basta definire un intervallo uguale a 0, con messaggi di comando del tipo: *INI1:00* per l'input 1 e *INI2:00* per l'ingresso 2.

Si noti che il tempo di inibizione, per impostazione predefinita o dopo un reset totale, è fissato in 5 minuti.

Passiamo adesso ai comandi di controllo dei relé: il primo definisce l'attivazione bistabile ed ha una sintassi del tipo *OUTx:stato*, dove al posto della *x* va inserito il numero del relé (1 o 2) e *stato* può valere *ON* o *OFF*. Se ad esempio dovessimo attivare il relé 2, invieremmo il comando *OUT2:ON*, mentre per disattivarlo invieremmo *OUT2:OFF*. Si può conoscere lo stato dei relé inviando il comando *STA?* a cui il modulo risponde con un SMS contenente l'attuale situazione di relé ed ingressi. Il messaggio viene inviato al telefono che ha impartito il comando. Ferma restando la possi-

- Frequenza operativa: banda GSM 900/1.800 MHz;
- Numeri memorizzabili: 8; il primo è assegnato all'anti-jammer;
- Segnalazione: SMS e chiamata voce in caso di attivazione ingressi e entrata in funzione del controllo anti jamming;
- Uscite: 2, dotate di relé in grado di sopportare fino a 250 V a 10 A;
- Ingressi: 2 optoisolati a livello di tensione (3 ÷ 30 V);
- Modi operativi: anti-jammer master/slave; teleallarme; combinatore telefonico; unità di telecomando;
- Alimentazione: 5 ÷ 32 Vcc, 300 mA max.

bilità di attivare in maniera stabile le due uscite, il telecomando prevede l'opzione di comando monostabile; inviando un SMS contenente il testo *OUTx:ss* il relé *x* (quindi 1 o 2) inverte la propria condizione per l'intervallo di tempo definito da *ss* (da 1 a 59 secondi). Per esempio se, dopo aver ordinato l'attivazione bistabile di relé 2, inviamo il comando *OUT2:03*, lo stesso relé si disattiva per tre secondi, per poi tornare attivo. Ovviamente lo si può diseccitare definitivamente solo con l'apposito comando (*OUT2:OFF*) o togliendo al telecomando GSM l'alimentazione, a patto che non sia attivo il modo di ripristino delle uscite.

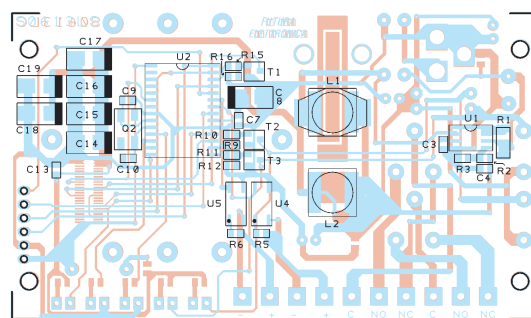
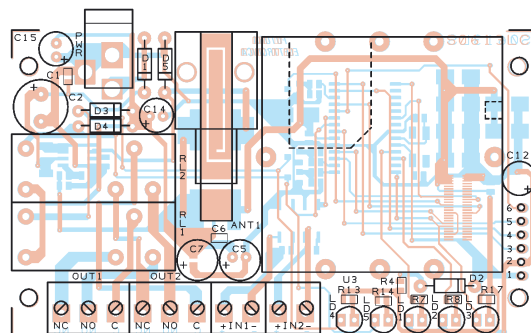
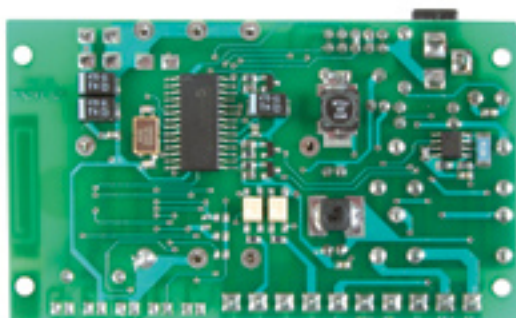
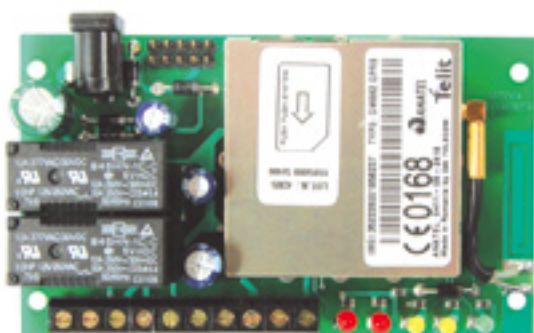
Quest'ultima funzione consente, in caso di black-out, di memorizzare lo stato dei relé e ripristinarlo al ritorno della tensione di alimentazione; già attiva per impostazione predefinita del software caricato nel microcontrollore, la si disattiva o riattiva con i comandi SMS, rispettivamente, *RIP0* e *RIP1*. Per sapere in ogni momento se il ripristino è attivo o meno, è a disposizione il comando *RIP?*, che restituisce un SMS indicante l'attuale condizione.

Bene, giunti a questo punto parliamo dell'ultima classe di comandi: quella che permette di definire i messaggi di notifica inviati dal teleallarme. Partiamo con l'SMS inviato ai numeri della lista (dal 2 al 7) ogni volta che scatta l'allarme

dovuto alla mancata risposta del dispositivo master con la quale lo slave, atteso invano l'intervallo impostato, chiama per interrogare il master sulla sua condizione; il testo (max 100 caratteri, spazi compresi) si definisce inviando l'SMS *TAL:xxxxxxxxxx*, nel quale al posto delle *x* bisogna scrivere il testo. La frase predefinita nel software del microcontrollore è *ALLARME MASTER*.

Riguardo all'attività degli ingressi (telecontrollo/teleallarme) è possibile definire, per ciascuno, sia il messaggio di notifica corrispondente alla presenza di tensione, sia quello relativo all'assenza di tensione. La sintassi del comando è *TINyz:xxxxxxxxxx* in cui *y* rappresenta l'ingresso (1 o 2), *z* il livello di tensione (*A*=alto, *B*=basso) e le *x* rappresentano il testo da introdurre dopo il simbolo "due punti". Come per il precedente messaggio, il testo deve avere al massimo 100 caratteri, spazi compresi. Resta inteso che i messaggi vengono inviati dal sistema in funzione delle impostazioni del livello logico inteso come allarme. In ogni momento l'utente in possesso della password può ripristinare le impostazioni iniziali (predefinite) del sistema e cancellare in un sol colpo tutti i numeri telefonici memorizzati, inviando il comando *RES:pwd*, nel quale *pwd* è la password corrente. Tale opzione comporta la possibilità che il si-





ELENCO COMPONENTI:

R1: 0,1 ohm 1W
 R2: 2,2 kohm
 R3: 1,2 kohm
 R4+R5, R9: 4,7 kohm
 R7, R8, R17: 470 ohm
 R10, R12, R16: 10 kohm
 R11: 4,7 kohm
 R13, R14: 1 kohm
 R15: 4,7 kohm
 C1, C4: 100 nF multistrato SMD
 C2: 470 µF 35 VL elettrolitico
 C3: 100 pF ceramico SMD
 C5: 220 µF 16 VL elettrolitico
 C6, C7, C13: 100 nF multistrato SMD
 C8: 470 µF 6,3 VL tantalio SMD
 C9, C10: 10 pF ceramico SMD

La realizzazione pratica dell'anti-jammer è abbastanza laboriosa, poiché i componenti, sia ad inserzione che a montaggio superficiale, sono montati su entrambi i lati del circuito stampato. Occorre pertanto prestare la massima attenzione montando i componenti SMD, specialmente il connettore per il modulo GSM.

C11, C12: 100 µF 16 VL elettrolitico
 C14+ C19: 470 µF 6,3 VL tantalio SMD
 Q1: quarzo 20 MHz SMD
 U1: MC34063
 U2: PIC18F2620 (MF613)
 U3: Modulo Telit GM862-GPRS
 U4, U5: TLP181
 D1+D4: 1N4007
 D5: 1N5819
 T1+T3: BC817
 LD1, LD2: led 3 mm giallo
 LD3: led 3 mm verde
 LD4, LD5: led 3 mm rosso

L1: Bobina 22 µH - 2A
 L2: Bobina 1 µH - 3A
 RL1: Relé 5V 1 scambio
 RL2: Relé 5V 1 scambio

Varie:
 - Morsettiere 2 poli (2 pz.)
 - Morsettiere 3 poli (2 pz.)
 - plug alimentazione
 - Strip maschio 6 pin
 - Connettore da c.s. per GM862
 - Connettore per antenna GSM
 - Circuito stampato codice S613

stema risponda a taluni comandi con più di un SMS; per evitare ciò, possiamo terminare un messaggio a più comandi con il testo *RISP*. Così facendo, il telecomando non genererà gli SMS di risposta.

Realizzazione pratica

Vediamo, infine, qualche nota pratica: una volta procurata la basetta, inserite e montate i componenti usando un saldatore a punta sottile

da 20÷25 watt di potenza; dispo-
 nete i componenti iniziando con il
 connettore miniatura per il modulo
 Telit, quindi sistemate gli altri ele-
 menti, quasi tutti SMD, prestando
 attenzione soprattutto ai componen-
 ti polarizzati (diodi, condensatori
 elettrolitici, transistor...). Poiché il
 micro è a montaggio superficiale,
 deve essere programmato in-circuit;
 il firmware è scaricabile dal sito
www.elettronica.in già compilato

e può essere caricato sul micro tra-
 mite l'apposito connettore a passo
 2,54 mm previsto allo scopo, che
 va collegato a un programmatore
 adatto.

Prima di inserire la SIM nel
 GM862, occorre disattivarne il PIN
 tramite un comune cellulare, per-
 ché il firmware del PIC18F2620
 non gestisce il codice d'accesso.
 Un discorso particolare lo merita
 l'antenna: siccome è costituita da

una pista dello stampato, bisogna collegarla al modulo con un cavetto coassiale, l'anima del quale va stagnata sull'esterno della pista.

Se necessario potete anche usare un'antenna esterna: basta montare un connettore sul circuito stampato e collegare ad esso il coassiale che giunge dal modulo. Ovviamente qualora il dispositivo venga inserito in un contenitore, quest'ultimo dovrà essere necessariamente plastico; utilizzando un contenitore metallico dovrete impiegare un cavo adattatore d'antenna con presa da pannello FME ed un'antenna bibanda.

Utilizzo pratico

Una volta completato il montaggio dei due apparati (stiamo parlando dell'impiego come anti-jammer), essi vanno configurati: ciò significa che, mediante SMS, bisogna memorizzare i numeri del master nel modulo slave e viceversa, nonché tutti quelli da chiamare; per l'esattezza, nella lista dello slave, a partire dalla seconda posizione vanno scritti gli indicativi telefonici da contattare (mediante telefonata o SMS) quando scatta l'allarme dovuto alla mancata risposta dell'unità master. In quest'ultima, le posizioni dalla seconda alla settima vanno usate per memorizzare i numeri di telefono degli utenti da allertare in caso di allarme. Oltre alla lista dei numeri da contattare, nei moduli vanno definiti anche i parametri di

Segnalazioni luminose			
Legenda			
● LED spento ● LED acceso			
LD1	LD2	Master	Slave
On ●	On ●	Chiamata in arrivo	Chiamata in arrivo
Off ●	Off ●	Chiamata terminata	Chiamata terminata
Off ●	On ●	Non previsto	Il dispositivo è slave
On ●	Off ●	Il dispositivo è master	Non previsto
5 lampeggi	Mantiene lo stato precedente	Reset timer	Reset timer
5 lampeggi	5 lampeggi	SMS in arrivo	SMS in arrivo
 1 lampeggio	Mantiene lo stato precedente	Inizio chiamata	Inizio chiamata

funzionamento quali i testi degli SMS da inviare in caso di allarme, le modalità di funzionamento di ingressi e relé e il tempo di attesa tra una trasmissione/ricezione e l'altra. Relativamente a quest'ultimo, suggeriamo di effettuarne la programmazione solo dopo aver ultimato tutte le altre e ciò per una ragione: per impostazione predefinita, dopo la prima accensione e a seguito di ogni reset, in entrambe le unità è impostato un intervallo di 50 minuti. Ciò significa che l'unità slave, se non riceve chiamate dal master, non fa partire la sequenza di allarme almeno per oltre 52 minuti dall'accensione, un tempo più che sufficiente a configurare entrambi i circuiti. Se si va a ridurre tale intervallo prima di avere ultimato tutte le altre impostazioni, si rischia di avere delle segnalazioni di allarme sullo slave prima di avere ultimato la sua configurazione. Dunque, il consiglio è configurare prima il di-

positivo master e lasciarlo spento, poi impostare tutti i parametri dello slave, modificando per ultimo il tempo di attesa; solo alla fine delle operazioni, alimentate il master.

Un'ultima nota riguarda la possibilità di inviare, durante la configurazione, i cosiddetti multimesaggi (SMS pluricomando): le unità del sistema accettano SMS contenenti più comandi, anche di diverso tipo; in tal modo, chi deve configurare i moduli può risparmiare tempo e denaro, perché invece di mandare un SMS per ogni impostazione può, con uno solo, definire più parametri alla volta. I messaggi pluricomando non possono essere usati per i parametri riguardanti i testi degli SMS da trasmettere in caso di allarme; ciò perché, potendo scrivere testi di 100 caratteri, ciascuno risulterebbe troppo lungo e non si riuscirebbe a metterne due o più in un solo SMS.

per il

MATERIALE

Il firmware in formato esadecimale, necessario per il funzionamento come anti-jammer, è disponibile gratuitamente e può essere scaricato dal sito della rivista (www.elettronica.in.it). Ricordiamo che il micro va programmato direttamente sulla basetta con la modalità in-circuit. Il progetto descritto in queste pagine (cod. TDG38) può anche essere acquistato già montato e collaudato al prezzo di 192,00 Euro IVA compresa.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futuranet.it>

Radiocomando 4 canali Rolling Code

di Arsenio Spadoni

Dispone di uscite a relé, attivabili ad impulso o a livello, con le quali comandare l'attivazione di utilizzatori funzionanti a bassa tensione; particolarmente sicuro grazie all'adozione di un innovativo modulo Aurel che contiene sia la sezione a radiofrequenza che la decodifica Keeloq a codice variabile.

Sono passati molti anni da quando, descrivendo e proponendo progetti di radiocomandi con codifica variabile, abbiamo dato il via ad un nuovo modo di progettare questo genere di apparati, aumentandone enormemente il grado di sicurezza. Dai primi timidi passi mossi alla scoperta dei Dynacoder e dell'algoritmo Keeloq, sono nati infatti numerosi schemi di comandi a distanza via radio basati su un sistema di codifica ad elevata sicurezza utilizzabile per l'attivazione e disattivazione dei più sofisticati impianti d'allarme ma anche dei comuni impianti apricancello o apriporta. Tale siste-

ma di codifica è profondamente diverso dai canonici codici fissi ottenibili da integrati pur ancora utili, quali gli MM53200 (e i suoi derivati, UM3750, UM86409, ecc.) e la famiglia MC1450xx, in quanto basa il suo funzionamento sulla generazione di codici variabili dipendenti da algoritmi matematici. La recente disponibilità di un innovativo modulo ibrido Aurel ci permette di parlare nuovamente di questa tecnologia in quanto il modulo in questione (contraddistinto dal codice RX 4M-HCS) comprende, oltre alla sezione a radiofrequenza, anche un decodificatore Keeloq: con questo modulo risulta >

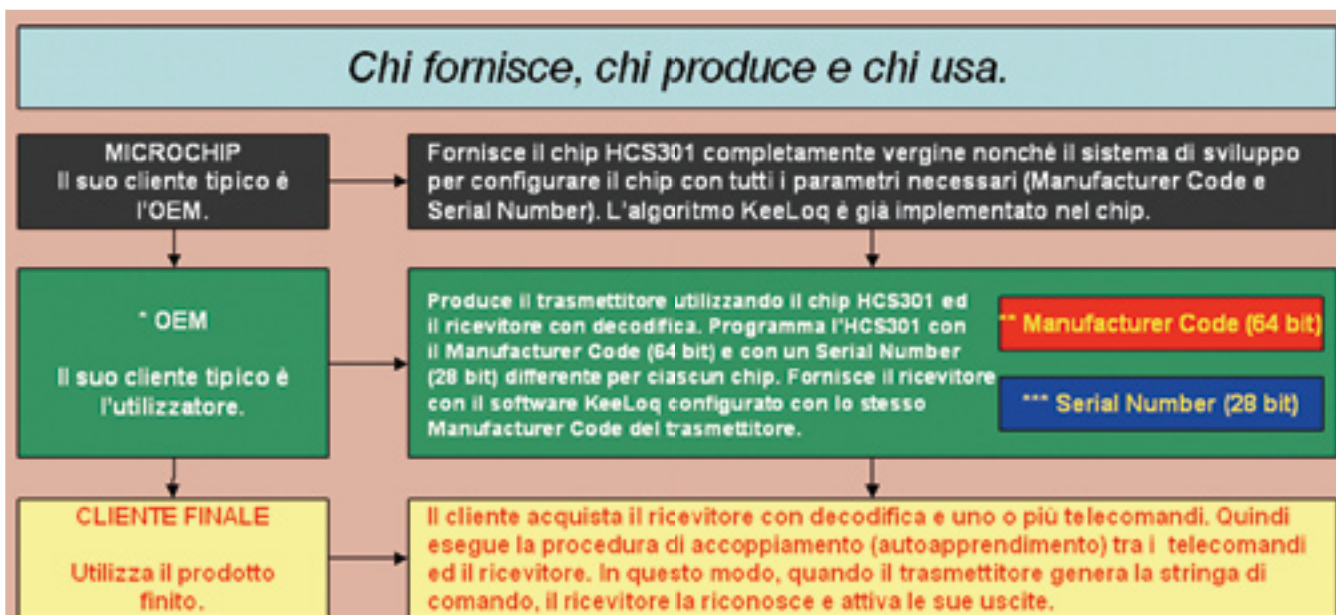


Fig. 1

* OEM: Original Equipment Manufacturer. Si intende l'azienda che acquista i componenti base (chip, resistenze, condensatori...) e realizza i prodotti finiti (il telecomando, la televisione, il forno a microonde).

** Manufacturer Code: è un codice a 64 bit specifico di ciascun produttore che viene memorizzato sia nell'EEPROM dell'encoder HCS301 che nella memoria del microcontrollore utilizzato come decoder e nel quale è implementato il software KeeLoq. Per poter funzionare, un Encoder e un Decoder devono avere lo stesso Manufacturer Code.

*** Serial Number: è un codice progressivo a 28 bit col quale vengono differenziati i chip HCS in fase di programmazione. Il Serial Number consente dunque di distinguere un trasmettitore da un altro. In fase di accoppiamento tra TX e RX, questo codice viene appreso dal ricevitore; durante il normale funzionamento l'encoder verifica come prima cosa che il Serial Number del TX che trasmette sia tra quelli memorizzati e solo in questo caso procede a decodificare e verificare gli altri dati contenuti nella stringa in arrivo.

veramente semplice realizzare un radiocomando a quattro canali con possibilità di funzionamento dei relé in modalità bistabile o impulsiva.

Al ricevitore possono essere accoppiati fino ad un massimo di 10 trasmettitori a quattro canali tipo TX-4M-HCS, prodotti dalla stessa Aurel e forniti già montati e collaudati.

La novità introdotta dal sistema sta nel fatto che l'intera unità ricevente del radiocomando è praticamente contenuta in un unico ibrido SMD Aurel che contiene, quindi, il radio-ricevitore, la decodifica rolling-code a base HCS301 Microchip ed anche i driver per i relé. In pratica, per realizzare il ricevitore, è sufficiente fornire alimentazione al modulo e collegare quattro relé alle quattro uscite!

Prima di analizzare nei dettagli lo schema elettrico, proviamo a "risolvere" le nozioni fondamentali del sistema rolling-code e della co-

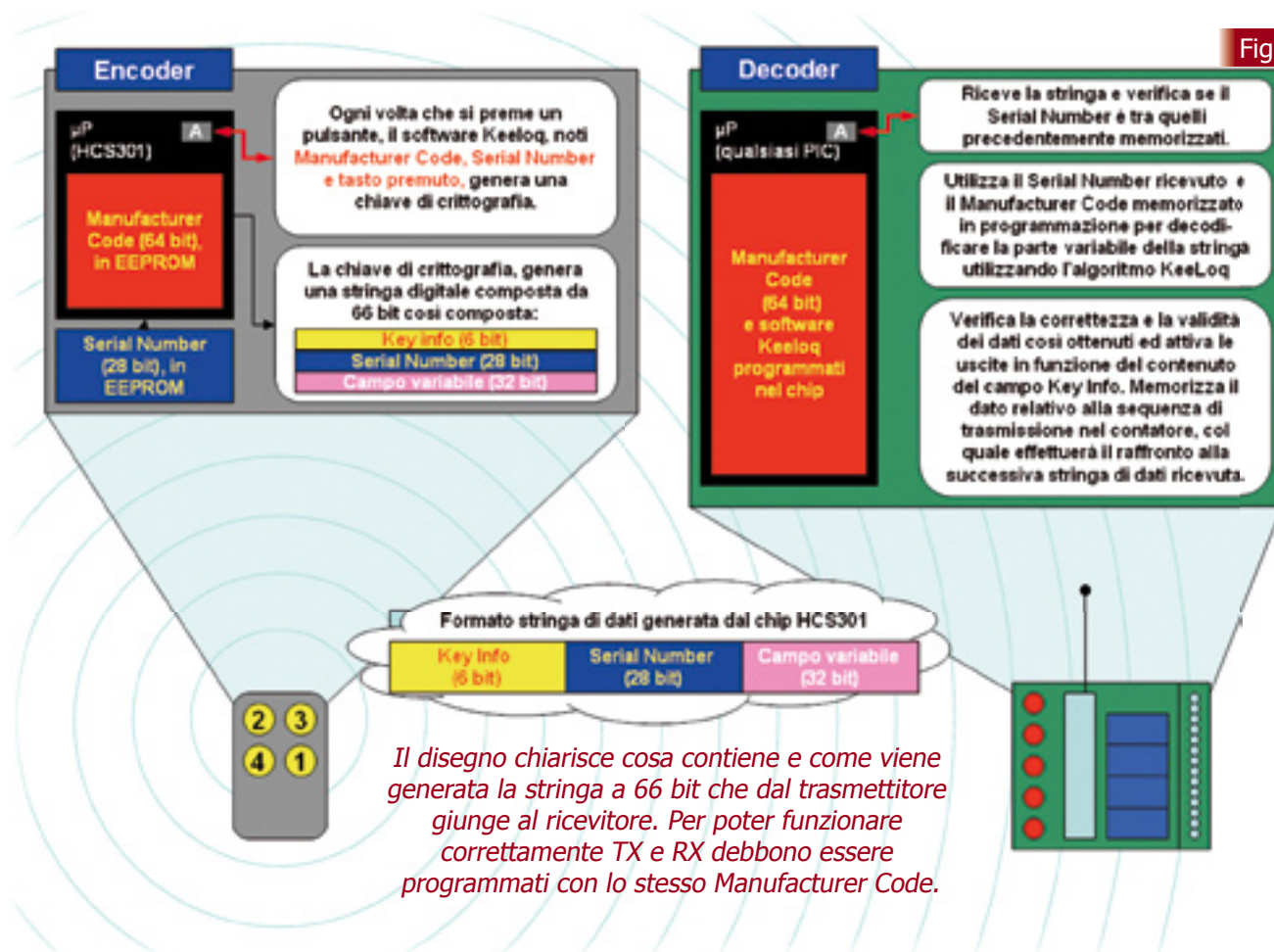
difica Microchip che utilizza l'algoritmo chiamato KeeLoq.

La codifica KeeLoq

Contrariamente ai sistemi a codifica fissa (UM3750, MM53200, ecc.) in cui sia il trasmettitore che il ricevitore devono essere programmati con un indirizzo comune ad entrambi (normalmente la programmazione si effettua tramite dip switch), la codifica KeeLoq è di tipo rolling-code (alla lettera "codice rotolante"), ossia non fissa ma variabile; per l'esattezza, ad ogni trasmissione l'encoder dell'unità trasmittente genera e invia una stringa che differisce notevolmente da quella trasmessa l'ultima volta. Affinché il decoder posto sul ricevitore la riconosca, deve operare con lo stesso algoritmo, nel senso che deve conoscere la "formula matematica" secondo cui la codifica varia ad ogni trasmissione, nonché alcuni dati comuni sia all'encoder

che al decoder. Una procedura di questo tipo consente di superare il principale limite dei telecomandi a codifica fissa: la possibilità di ripetere il codice, per esempio intercettando la trasmissione e memorizzando la stringa di dati, per poi riutilizzarla allo scopo di ottenere l'accesso al comando. Nel caso dei sistemi a codifica variabile, chi intercettasse e decifrasse la comunicazione non riuscirebbe a sapere cosa trasmettere per attivare il decodificatore, perché per farlo dovrebbe conoscere la sequenza di variazione del codice, oltre che i dati presi a riferimento per generare la chiave di crittografia della stringa.

Quello che proponiamo in queste pagine è, appunto, un radiocomando rolling-code, sviluppato usando l'encoder HCS301: il componente Microchip genera ad ogni attivazione una stringa di 66 bit, dei quali i primi 28 corrispondono al Serial



Number e sono trasmessi in chiaro, i successivi 32 rappresentano la parte variabile generata dalla chiave crittografica Keeloq, mentre gli ultimi 6 bit trasportano le informazioni relative al pulsante premuto ed allo stato della batteria. Per comprendere più facilmente quanto stiamo per analizzare, consigliamo di osservare le figure 1 e 2. Tutte le volte che viene attivato, l'encoder emette una sequenza di bit che costituisce il suo codice digitale, cioè l'insieme di tre gruppi di dati, dei quali il primo a 28 bit è univoco per ciascun chip (e quindi per ciascun telecomando) e viene memorizzato dal produttore dei telecomandi durante la fase di programmazione nell'HCS301; il secondo blocco è formato da 32 bit, diversi ad ogni trasmissione. Il cambiamento non è affatto casuale (altrimenti il ricevitore non saprebbe cosa aspettarsi) ma segue un preciso algoritmo determinato dall'uni-

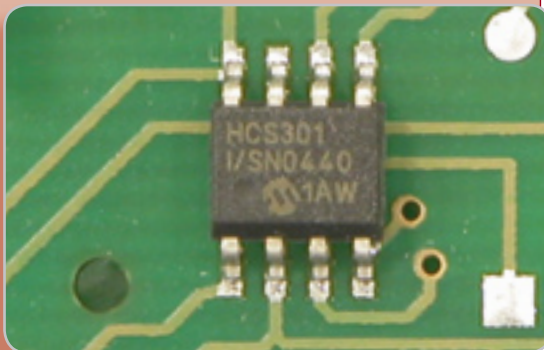
tà di elaborazione interna sulla base del codice fisso di 28 bit, nonché sulla base del Manufacturer Code a 64 bit, anch'esso memorizzato nel chip durante la fase di programmazione. Quando viene premuto un pulsante ad attivata la trasmissione, il firmware dell'HCS301 prende in considerazione il Serial Number da 28 bit, il Manufacturer Code da 64 bit, i dati relativi al pulsante e quelli relativi all'ultima attivazione generando una chiave di crittografia che, a sua volta, produce la sequenza dei 32 bit, completamente differenti da quella precedente e da quella successiva. Questi 32 bit vengono trasmessi unitamente ai 28 bit del Serial Number e ad altri 6 bit contenenti le informazioni del pulsante. In tutto, quindi, 66 bit di cui 32 che cambiano completamente ad ogni trasmissione. Alla luce di ciò, un simile sistema di codifica si può davvero definire univoco e invio-

labile. Riepilogando, il costruttore dei telecomandi scrive nel chip un Manufacturer Code a 64 bit uguale per ciascun lotto di produzione e un dato a 28 bit (Serial Number) che cambia (solitamente in maniera incrementale) da chip a chip. La chiave di crittografia, una volta generata, va a finire in una EEPROM riservata allo scopo e viene richiamata quando l'HCS301 deve sintetizzare il proprio algoritmo di variazione dei 32 bit variabili della stringa di dati emessa ad ogni trasmissione.

A questo punto vediamo di capire cosa succede in fase di ricezione, ovvero come funziona il decoder. Quest'ultimo è solitamente costituito da un PIC nel quale è stato caricato sia il software Keeloq sia l'altro dato di cui il chip ha tassativamente bisogno: lo stesso Manufacturer Code a 64 bit utilizzato per programmare l'encoder. Non è possibile fare funzionare due dispositivi (encoder e decoder) ➤

La codifica HCS301

L'encoder Microchip è tra i più utilizzati nei radiocomandi ad elevata sicurezza perché implementa un algoritmo di variazione della codifica praticamente non ripetibile e inviolabile; costruire il decoder è cosa abbastanza semplice, perché la Casa rende disponibile il software da implementare nel micro che svolge questa funzione. Sebbene venga utilizzato da quasi tutti i fabbricanti di radiocomandi, l'HCS301 garantisce un'elevatissima sicurezza d'uso perché permette di realizzare prodotti personalizzati: ogni costruttore ha la possibilità di personalizzare la propria produzione con un Manufacturer Code a 64 bit, che rappresenta uno dei dati presi in considerazione per generare la chiave di crittografia con la quale viene generata la parte variabile del treno d'impulsi a 66 bit prodotto dall'encoder HCS301. Ogni singolo chip viene inoltre identificato da un codice univoco a 28 bit denominato, appunto, Serial Number. Anche questo dato viene memorizzato in fase di programmazione del chip da parte del costruttore del trasmettitore. In ultima analisi ciascun costruttore programma i chip con un proprio Manufacturer Code a 64 bit (lo stesso utilizzato nel software di decodifica del ricevitore) ed un Serial Number a 28 bit, univoco per ciascun chip HCS301.



con Manufacturer Code differenti. Ciò, tuttavia, non è sufficiente. E' necessario infatti accoppiare uno o più trasmettitori al ricevitore che si vuole controllare. Questa fase, detta di autoapprendimento, consiste nell'inviare una stringa completa al ricevitore il quale estrarrà la stringa a 28 bit (trasmessa in chiaro) e la memorizzerà. A seconda della capacità di memoria, ciascun decoder potrà memorizzare i dati di uno, dieci o cento trasmettitori. I relativi Serial Number, dunque, risulteranno presenti in EEPROM. Durante il normale funzionamento, quando il decoder riceve una stringa di dati dal TX, effettua preliminarmente una verifica sul Serial Number in arrivo: se questo corrisponde ad uno di quelli presenti in EEPROM, prosegue nell'analisi della stringa, in caso contrario ignora completamente la trasmissione. Risulta evidente, dunque, che TX e RX vengono accoppiati sulla base del Serial Number di ciascun TX. Nel caso il "match" dia risultato positivo, il decoder ef-

fettua l'analisi della parte variabile della stringa (i 32 bit) sulla base di una chiave crittografica generata dal firmware Keeloq che prende a riferimento sia il Manufacturer Code a 64 bit che il Serial Number a 28 bit. In pratica, semplificando, l'encoder effettua l'operazione inversa del decoder (che utilizza gli stessi dati e lo stesso software per generare la stringa variabile a 32 bit). In realtà la cosa non è così semplice in quanto nel codice a 32 bit sono contenute numerose altre informazioni crittografate la più importante delle quali riguarda lo stato del contatore relativo al sincronismo tra TX e RX. Dopo l'autoapprendimento e la sincronizzazione iniziale, il decoder conosce la modalità di variazione dei 32 bit, ovvero sa cosa aspettarsi dall'encoder ad ogni trasmissione; tuttavia, se il trasmettitore viene attivato più volte senza che il ricevitore possa captarne il segnale, alla prima ricezione l'encoder si blocca. Una simile evenienza non è affatto inverosimile: ad esempio, può acca-

dere che, mentre ci si trova in vacanza, lasciando in tasca il trasmettitore per sbaglio si preme più volte il tasto di trasmissione, magari sedendosi in una certa posizione. E, perché no, che un bambino, trovato il TX, per gioco preme continuamente il tasto di attivazione. In tutti i casi, al rientro la trasmittente si sarà portata fuori passo rispetto alla ricevente e il radiocomando non funzionerà.

Se si perde il sincronismo, teoricamente bisogna procedere alla risincronizzazione manuale (una nuova fase di apprendimento) tuttavia l'HCS301 prevede due metodi di ripristino automatico: l'algoritmo Keeloq ammette sempre una tolleranza di 64 tentativi, nel senso che è possibile risincronizzare il decoder con l'encoder anche se quest'ultimo ha trasmesso fino a 64 volte senza che il ricevitore lo abbia captato; ciò perché il programma del decodificatore conosce i passi ammessi in tolleranza. In altre parole, quando riceve un segnale, il decoder controlla se il valore finale è uno di quelli rientranti nel margine di 64 tentativi.

Dopo eventuali trasmissioni a vuoto, al ripristino del collegamento il decoder legge il risultato ricevuto, quindi se non combacia con il passo successivo all'ultimo identificato effettua il confronto con tutte le 64 possibilità ammesse; nel caso trovi che il dato corrisponde ad una di esse si rimette in passo con l'encoder cosicché al prossimo arrivo del segnale ripartirà dal valore seguente. Quanto detto riguarda il primo metodo.

Il secondo consente al decoder di rimettersi in passo con il codificatore quando quest'ultimo è stato attivato per più di 64 volte al di fuori del campo di copertura del collegamento radio; in tale evenienza bisogna effettuare due trasmissioni consecutive e i due dispositivi si sincronizzano nuovamente. Chiaramente ciò non accade con un trasmettitore qualunque, altrimenti sarebbe fin

Il modulo RX 4M-HCS

troppo facile avere l'accesso con un dispositivo analogo: può essere fatto solo se il TX è stato inizialmente abbinato con la procedura di autoapprendimento, ovvero se il decodificatore ha già memorizzato i 28 bit fissi del Serial Number del trasmettitore in oggetto.

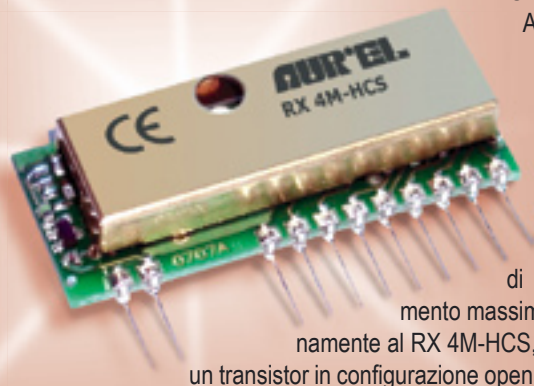
Radiocomando e decoder

Spiegato come funziona il nostro sistema rolling-code, possiamo analizzare il progetto descritto in queste pagine, che consta di un'unità trasmittente e una ricevente.

La prima è un microTX palmare, quadricanale, funzionante con un'apposita micropila a 12 volt, che, supponendo un uso normale (tre o quattro attivazioni al giorno) gli garantisce un'autonomia di qualche anno; al suo interno troviamo un oscillatore da 1 mW stabilizzato con risuonatore SAW accordato a 433,92 MHz e modulato dagli impulsi che l'encoder Microchip HCS301 produce ogni volta che viene premuto un pulsante. Il trasmettitore è prodotto dall'Aurel e siglato TX-4M-HCS.

Quanto al ricevitore, è la parte più interessante del progetto, in quanto adotta un nuovo ibrido che contiene un radoricevitore con stadio d'ingresso superrigenerativo (dotato di filtro ceramico per restringere la banda) accordato a 433,92 MHz e demodulatore d'ampiezza, seguito da un comparatore in funzione di squadratore dei fronti del segnale demodulato; all'uscita dello squadratore si trova un microcontrollore programmato che gestisce le quattro uscite pilotando opportunamente altrettanti transistor NPN configurati in modo open-collector.

Il micro è programmato con il software necessario a decifrare i segnali inviati dal minitrasmettitore con HCS301, e contiene perciò l'algoritmo Keeloq. Trasmettitore palmare e ricevitore utilizzano ovviamente lo stesso Manufacturer Code a 64 bit. ➤

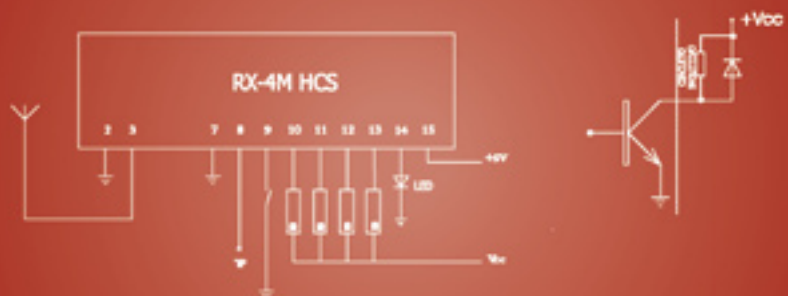


Si tratta di un ibrido in tecnologia SMD contenente un ricevitore AM, sintonizzato sui 433,92 MHz, che integra una decodifica con algoritmo Keeloq di Microchip con quattro canali di uscita completi di altrettanti driver a transistor NPN open-collector, capaci di pilotare carichi con assorbimento massimo di 100 mA ciascuno. Internamente al RX 4M-HCS, ogni uscita è comandata da

un transistor in configurazione open collector, in grado di erogare una corrente massima di 100mA. Lo stato non attivo è con il transistor in interdizione, mentre lo stato attivo è con il transistor in saturazione a GND. Le uscite possono essere programmate per funzionare in modalità monostabile o bistabile, ognuna delle quali in modo indipendente. Nel modo di funzionamento monostabile l'uscita è attiva per tutto il tempo che viene premuto il relativo tasto nel trasmettitore, rilasciando il tasto l'uscita si porta nello stato non attivo. Nel modo di funzionamento bistabile l'uscita cambia stato ad ogni pressione del tasto (da non attiva ad attiva e viceversa). I due modi di funzionamento sono indipendenti, cioè è possibile programmare alcune uscite per il funzionamento monostabile ed altre per il funzionamento bistabile. La pin-out dell'ibrido è riportata di seguito:

- 1) Non presente
- 2) GND
- 3) Antenna
- 4) Non presente
- 5) Non presente
- 6) Non presente
- 7) GND
- 8) Test Point – Uscita analogica RX
- 9) Pulsante di programmazione connesso a massa
- 10) Uscita CH1 - Open collector
(Attivata dalla pressione del pulsante 1 nel trasmettitore)
- 11) Uscita CH2 - Open collector
(Attivata dalla pressione del pulsante 2 nel trasmettitore)
- 12) Uscita CH3 - Open collector
(Attivata dalla pressione del pulsante 3 nel trasmettitore)
- 13) Uscita CH4 - Open collector
(Attivata dalla pressione del pulsante 4 nel trasmettitore)
- 14) Uscita LED – Connessa all'anodo del LED, catodo a massa
- 15) Vcc (+5 volt)

Notate che i piedini 1, 4, 5, 6 non sono presenti.



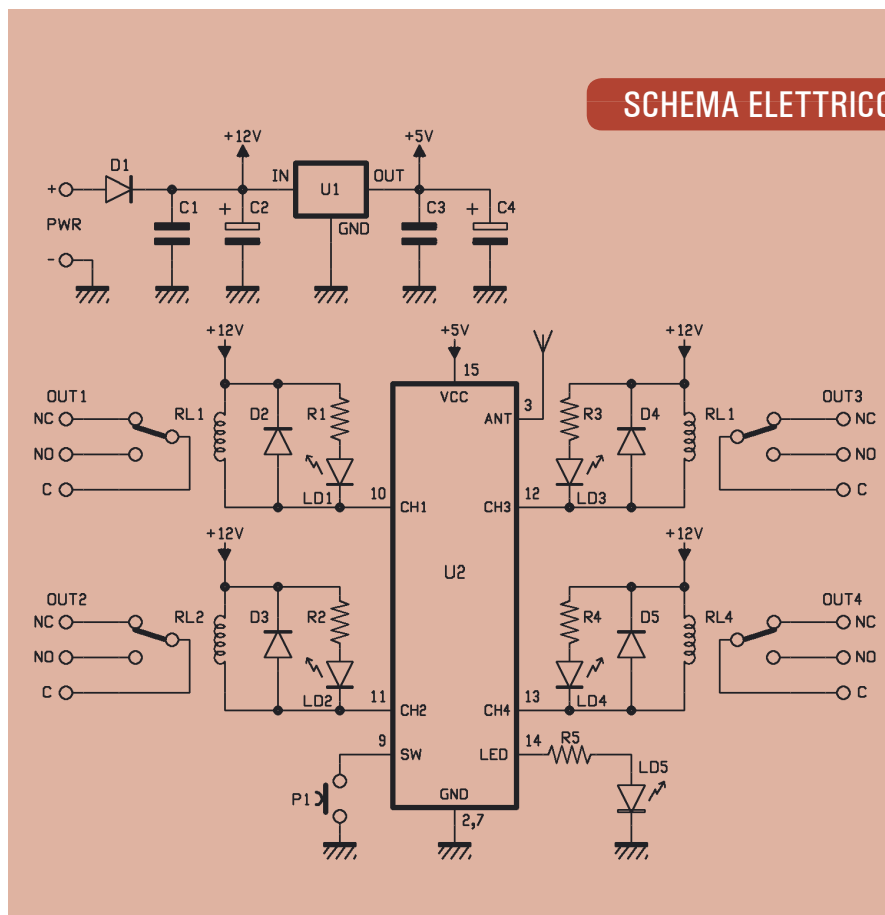
Schema del ricevitore

Dato che quasi tutto è contenuto nell'ibrido U2, il circuito è poca cosa: si riduce a quattro relé a 12 V, ciascuno provvisto di un led posto in parallelo alla bobina e di un diodo al silicio necessario ad eliminare le extratensioni generate dalla bobina stessa; i diodi di protezione evitano il danneggiamento dei transistor driver situati nel modulo Aurel.

La tensione di alimentazione dell'intero ricevitore deve essere in continua, di valore compreso fra 12 e 15 V, e si applica ai punti + e - PWR; il diodo D1 serve a proteggere l'insieme dai danni che potrebbero verificarsi se l'alimentazione venisse applicata con polarità invertita.

A valle del diodo si preleva la tensione che serve a far funzionare le bobine dei quattro relé, ciascuno dei quali è pilotato in modo sink, ossia ad assorbimento di corrente (il transistor corrispondente chiude a massa l'estremo inferiore della bobina, permettendo lo scorrimento di corrente dal positivo); sempre sul catodo del D1, il regolatore integrato U1 (un comune 7805) prende ciò che gli serve a ricavare 5 volt ben stabilizzati occorrenti al funzionamento del modulo ibrido.

Il tutto assorbe un massimo di 150 milliampere, quando i quattro relé sono eccitati e tutti i led illuminati. Gli scambi di RL1, RL2, RL3 e RL4 sono resi completamente disponibili, così da poterne utilizzare sia la parte normalmente chiusa che il contatto normalmente aperto; ogni scambio può commutare tensioni fino a 60V e correnti dell'ordine di 500 milliampere, quindi può essere utilizzato per accendere e spegnere utilizzatori funzionanti in bassa tensione e di piccola potenza, ma anche per comandare centraline apricancello, impianti d'allarme per auto e casa, moduli di controllo per tapparelle motorizzate ecc.

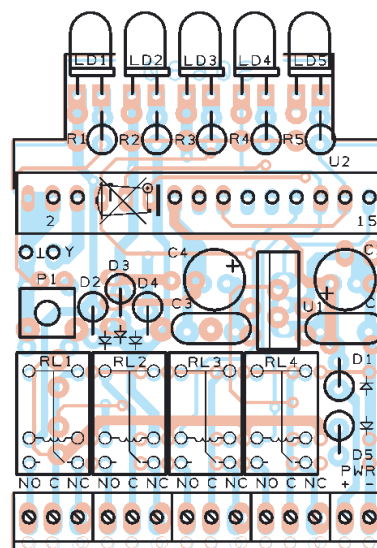
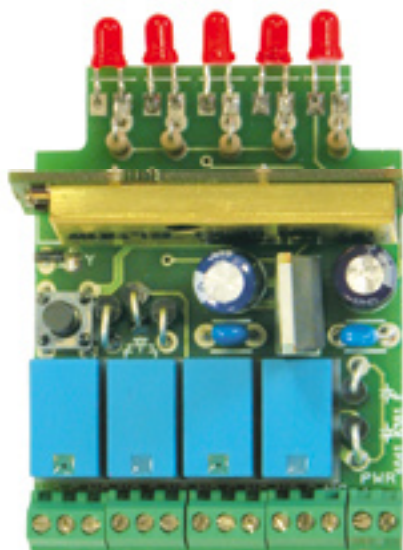


Realizzazione pratica

Bene, commentato lo schema elettrico possiamo passare alle note costruttive; per prima cosa diciamo che l'unica unità da costruire è la ricevente, in quanto la trasmittente si compera già pronta, tarata e collaudata. Quanto al ricevitore, si realizza partendo da un piccolo circuito stampato facilmente ottenibile per fotoincisione (come di consueto, le tracce dei lati ramati si possono scaricare dal nostro sito web (www.elettronica.in.it) con un po' di attenzione, visto che è a doppia faccia; quindi, per prepararlo esponete nel bromografo prima un lato (con la rispettiva pellicola) e poi, fatti un paio di fori, l'altro, sovrapponendo il master corrispondente ben centrato prendendo a riferimento i fori fatti. Incise le due facce, dopo aver realizzato tutti i fori iniziate il montaggio disponendo, nell'ordine, i condensatori non polarizzati, le resistenze e i diodi (per ottenere una

basetta poco ingombrante, è stato previsto che tutti siano montati verticalmente) quindi il pulsante da c.s. e gli elettrolitici, poi i quattro relé e il regolatore 7805, che deve essere posizionato verticalmente e con il lato metallico (quello che normalmente verrebbe posto a contatto di un dissipatore) verso l'interno del PCB. Fatto ciò, inserite e saldate i cinque diodi luminosi, dopo averne piegato i terminali a 90° in modo da poterli far sporgere dal bordo dello stampato, quindi sistemate l'ibrido, ponendolo in verticale e ben parallelo alla superficie della basetta; non preoccupatevi della disposizione, perché per come sono disposti i suoi piedini entra solo nel verso corretto. Per agevolare le connessioni riguardanti l'alimentazione e gli scambi dei relé sono state previste delle morsettiere miniatura a passo dimezzato (1,27 mm) rispetto a quelle tradizionali. Il montaggio si completa con l'antenna, per realizzare la quale è

Avendo a disposizione il circuito stampato, la realizzazione del ricevitore è piuttosto semplice poiché il modulo Aurel integra tutti i componenti elettronici attivi necessari. All'esterno non rimane che collegare pochi componenti quali relé, diodi, condensatori e poco altro ancora. La parte più complessa è il PCB, un doppia faccia a fori metallizzati molto compatto.



ELENCO COMPONENTI:

R1÷R4: 1 kohm
R5: 470 ohm
C1: 100 nF multistrato
C2: 100 µF 25 VL elettrolitico
C3: 100 nF multistrato
C4: 220 µF 16 VL elettrolitico

D1÷D5: 1N4007
U1: 7805
U2: RX-4MHCS
LD1÷LD5: led 3 mm rosso
P1: microswitch
RL1÷RL5: Relé miniatura 12V

Varie:

- Morsettiere 3 poli passo 2 mm (4 pz.)
- Morsettiere 2 poli passo 2 mm
- circuito stampato codice S680
- antenna a filo da 17 cm oppure a stilo con corpo in gomma

sufficiente tagliare uno spezzone di filo in rame smaltato o nudo, lungo 17 o 18 cm da infilare e saldare nell'apposita piazzola. Se adottate del filo smaltato, rimuovete lo smalto in prossimità della saldatura, altrimenti lo stagno non può aderire.

In alternativa, è possibile usare un'antenna esterna composta da uno stilo lungo 17 o 35 cm, inserito al centro di un piano di massa (dal quale deve essere elettricamente isolato) composto da una piastra ramata per circuiti stampati o da ferro dolce: si realizza, così, una ground-plane, che deve essere collegata connettendo il piano alla massa del ricevitore (vicino alla piazzola d'antenna) e lo stilo al piedino 3 dell'ibrido Aurel.

Come si usa

Ora che abbiamo terminato il montaggio, dobbiamo imparare a programmare ed utilizzare. La fase di autoapprendimento si effettua agendo sul pulsante P1. Riguardo

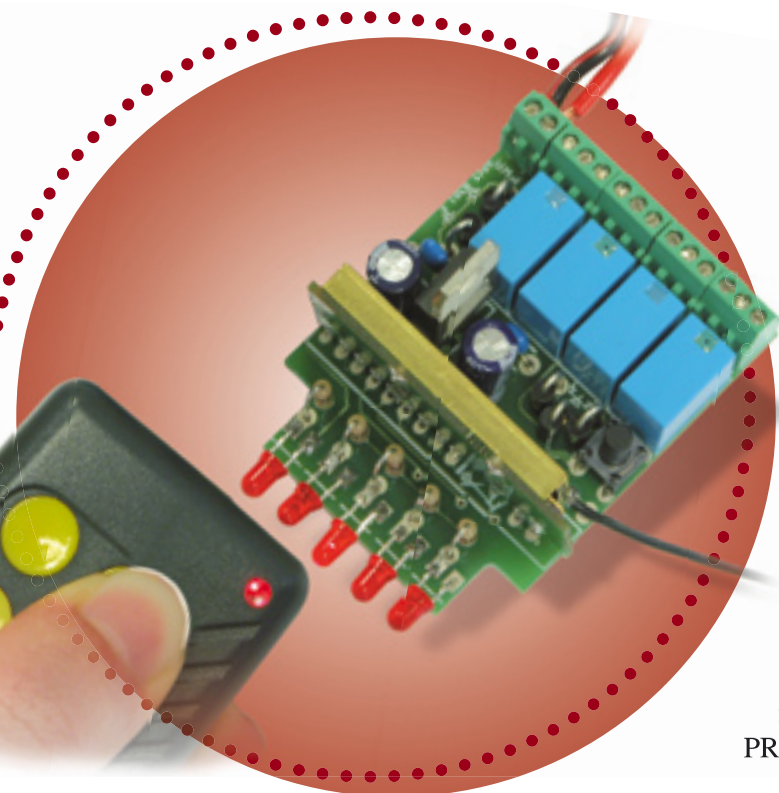
tale procedura, notate che ad un ricevitore possono essere abbinate fino a 10 trasmettenti, ciascuna delle quali sarà abilitata a comandare lo stato dei relé; resta inteso che l'abbinamento è possibile solo con TX caratterizzati da encoder Microchip HCS301 della stessa partita, cioè aventi il medesimo Manufacturer Code a 64 bit.

Oltre alla procedura di apprendimento, analizziamo quella inerente alla programmazione della modalità di funzionamento delle uscite, che deve essere definita canale per canale; i quattro relé possono essere programmati individualmente per funzionare in modalità monostabile o bistabile. Nel modo di funzionamento monostabile, l'uscita è attiva per tutto il tempo che viene premuto il relativo tasto nel trasmettitore; rilasciando il tasto l'uscita si porta nello stato non attivo. Nel modo di funzionamento bistabile, il relé cambia stato ad ogni pressione del tasto

(da non attiva ad attiva e viceversa). La programmazione della modalità di funzionamento delle uscite è individuale, intendendo con ciò che ciascun canale può essere impostato diversamente dagli altri.

Vediamo dunque, per prima, la fase di abbinamento: in ogni istante, salvo che non ci si trovi all'interno di un'altra procedura, premendo e rilasciando il pulsante di programmazione si entra nella fase di apprendimento, segnalata dal fatto che il led LD5 lampeggia rapidamente per 10 secondi; durante questo tempo, operando entro il campo di copertura radio del ricevitore, la pressione di un qualsiasi tasto del trasmettitore determinerà l'apprendimento del codice corrispondente al TX. L'avvenuta memorizzazione e il conseguente abbinamento trasmettente/ricevente vengono indicati dall'LD5 con l'immediata interruzione del rapido lampeggio e l'accensione a luce fissa per qualche istante; poi il led





apprendimento: basta premere nuovamente il pulsante e rilasciarlo non appena si illumina il led rosso; attenzione a non premerlo per troppo tempo, perché altrimenti si procede alla cancellazione dei dati in EEPROM.

Modo operativo delle uscite

Dopo l'abbinamento a un trasmettitore, il ricevitore è predisposto per far funzionare tutte le sue uscite in modalità monostabile. Per ottenere il funzionamento bistabile, bisogna premere e rilasciare il pulsante P1 del ricevitore. Quando il led LD5 inizia a lampeggiare, si deve premere nuovamente il pulsante P1: a questo punto il led rimane acceso a luce fissa per 10 secondi, periodo entro il quale deve essere premuto il pulsante del telecomando che controlla l'uscita da impostare come bistabile. Il ricevitore conferma con tre lampeggi del led di avere applicato la nuova modalità

all'uscita. Per ritornare al modo di funzionamento monostabile si ripete la procedura sopra; in questo caso l'avvenuto ripristino della modalità impulsiva per l'uscita in questione viene segnalata con due lampeggi del led. Il cambiamento del modo di funzionamento delle uscite può essere fatto solo dopo che il ricevitore ha appreso il codice del trasmettitore usato per la configurazione.

Cancellazione della memoria

Il modulo RX 4M-HCS dispone di una EEPROM nella quale memorizza i codici dei trasmettitori abbinati durante la fase di apprendimento e le modalità di esercizio delle uscite open-collector. Volendo rimuovere tutti i dati, ossia cancellare interamente la memoria, bisogna premere il pulsante P1 e rilasciarlo dopo che LD5 inizia a lampeggiare. Ora si deve ripremere nuovamente il pulsante tenendolo premuto per circa 5 secondi fino a quando non si spegnerà nuovamente. Al rilascio del pulsante il LED lampeggerà 5 volte, informando che la memoria è stata cancellata. Dopo il reset, nessun trasmettitore con codifica HCS viene riconosciuto, e tutte le uscite sono impostate per il modo di funzionamento monostabile. L'operazione di cancellazione totale della memoria va eseguita sempre alla prima installazione del ricevitore, per rimuovere i dati casuali che spesso si trovano in una memoria appena uscita di fabbrica.

per il

MATERIALE

Il progetto pubblicato in queste pagine è disponibile in scatola di montaggio (cod. FT680K) al prezzo di 32,00 Euro. Il kit comprende la basetta metallizzata a doppia faccia, tutti i componenti ed il modulo ricevitore/decodificatore Aurel. Quest'ultimo (cod. RX-4M-HCS) è anche disponibile separatamente al prezzo di 17,00 Euro. I telecomandi quadricanale sono disponibili già montati e collaudati (cod. TX-4M-HCS) al prezzo di 20,00 Euro cadauno, compresa IVA.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futuranet.it>

Oscilloscopio con PC, software e taratura

di *Davide Scullino*

Dopo la descrizione dell'hardware, dedichiamo questa puntata alle operazioni di taratura necessarie per rendere operativo il nostro strumento. Presentiamo anche il software per PC fornito a corredo dello strumento e iniziamo la descrizione delle DLL con cui creare programmi personalizzati.

La strumentazione per le misure di laboratorio presenta un costo ancora piuttosto elevato, specie se parliamo di strumenti di qualità. La disponibilità di prodotti orientali che in altri settori ha portato all'abbassamento dei prezzi di vendita, in questo caso non ha portato ad alcuna significativa riduzione dei prezzi. Fortunatamente, per contenere i costi, in questo campo esiste la possibilità di utilizzare delle interfacce da abbinare all'ormai immancabile Personal Computer: questa soluzione consente di realizzare strumenti con un elevato standard qualitativo con una spesa relativamente conte-

nuta. Un esempio è l'oscilloscopio che abbiamo iniziato a descrivere il mese scorso, un dispositivo da collegare alla porta parallela del quale, in questa puntata, presentiamo il software applicativo; si tratta di un programma sviluppato per funzionare in ambiente Microsoft Windows (adattandosi praticamente a tutte le versioni: 95, 98, NT, Me, 2000 e XP, sia Home che Professional Edition) che permette di realizzare, oltre alle funzioni di oscilloscopio, anche quelle di analizzatore di spettro e registratore di eventi o transienti. Un prodotto software completo e, soprattutto, del tutto gratuito (lo si scarica dal sito >

www.velleman.be). L'applicativo si installa senza difficoltà e si usa ancor più facilmente. Ma non è tutto qui: dallo stesso sito è possibile ottenere anche una DLL (Dynamic Link Library, ossia una libreria di Windows) chiamata DSOLink.dll, che permette al programma di inviare i dati direttamente ad un foglio di lavoro di Microsoft Excel. Ancora più importante è la seconda libreria (PCS100.dll) che consente ai più esperti di realizzare programmi personalizzati utilizzando i tool di sviluppo più disparati, da Delphi a Visual Basic a C++ Builder.

E allora, senza perdere altro tempo andiamo subito a vedere di cosa si tratta, esaminando prima di tutto il software di gestione che permette di ottenere dall'interfaccia le informazioni sul segnale prelevato dalla sonda connessa al BNC.

Il software di gestione

È l'interfaccia utente dello strumento di misura, l'elemento che permette di operare effettivamente sull'oscilloscopio; se preferite, potremmo definire il programma di gestione come il pannello di comando di ciascun apparato, provvisto di tante manopole, levette, pulsanti e selettori virtuali. Il software si chiama PCLab2000 e può essere scaricato dal Web come file "zippato" da scompattare e installare nell'hard-disk del computer, oppure installato dal CD-ROM fornito insieme al kit di montaggio K8031.

Una volta completata la procedura d'installazione, l'utente ha a disposizione il pannello di comando di più strumenti: oscilloscopio, generatore di funzioni, tracciante dei diagrammi di Bode, analizzatore di spettro e memoria per la registrazione di segnali e transienti.

Il software è integrato ed è stato messo a punto non solo per l'interfaccia descritta nella prima puntata di questo articolo, ma anche per un'altra (PCG10) eventualmente collegabile

insieme ad essa e che serve a generare forme d'onda. Usando solo il K8031, sono accessibili le funzioni di oscilloscopio, registratore di transienti e analizzatore di spettro nonché le relative finestre di dialogo.

Partiamo dall'inizio e, prima di descrivere l'installazione del software, elenchiamo i requisiti di sistema: il PCLab2000 gira su qualsiasi PC basato almeno su processore Intel Pentium o compatibile. Il computer deve avere un monitor e una scheda grafica capaci di garantire una risoluzione di 800x600 (SVGA) oltre al lettore CD, 3 Mb di spazio libero sull'hard-disk e, ovviamente, una porta parallela disponibile: quest'ultimo requisito non dovrebbe essere un problema, ma sui nuovi computer, in special modo i notebook, si potrebbe fare i conti con la sgradita sorpresa di non trovare la parallela; infatti i più moderni PC hanno talvolta le sole USB e firewire. In quest'ultima evenienza si può ricorrere ad un convertitore USB/centronics. L'installazione è molto semplice: inserite il disco nel lettore di CD-ROM e, se non parte automaticamente la schermata di avvio dell'installazione, aprite Risorse del computer, fate doppio clic sull'icona del CD-ROM, quindi su quella chiamata Setup della finestra che si apre. Se iniziate dal file .zip, fate doppio clic e scompattatelo sull'hard-disk; la scompattazione richiede un apposito programma decompressore, che può essere WinZip (www.winzip.com) o Stuffit (www.stuffit.com) a meno che non stiate lavorando sotto Windows XP, che è già in grado di aprire i file compressi ad estensione .zip. Decompresso il file, si può fare doppio-clic sull'icona di Setup della cartella creata e avviare la procedura di installazione.

In tutti i casi, vi verrà chiesto di accettare l'accordo di licenza e di confermare o modificare il percorso in cui il programma di installazione copierà i file; al termine, verranno automaticamente create le icone e, nel menu Programmi di Windows, sarà aggiunta la voce PCLab2000. A questo punto il programma potrà essere eseguito.

Riguardo all'installazione, ricordiamo che per installare il software su Windows NT e 2000 occorre disporre dei privilegi dell'amministratore di sistema, senza i quali non è possibile portare a termine l'installazione; in altre parole, bisogna lanciare il setup dopo aver ottenuto l'accesso a Windows come utente amministratore.

Avviando il programma appare la finestra di dialogo principale, suddivisa in tre riquadri e una sezione contenente tre pulsanti; in essa l'utente può definire alcuni parametri di funzionamento, tra i quali il modello dell'oscilloscopio e del generatore di funzioni collegati (l'impostazione predefinita riguarda PCS500/K8031 e PCG10) e la porta parallela usata. A riguardo, notate che è possibile specificare fino a tre indirizzi, cor-

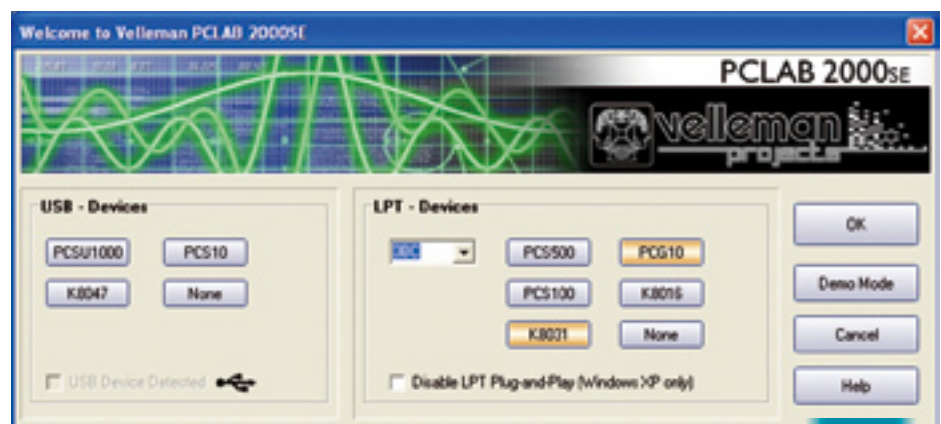


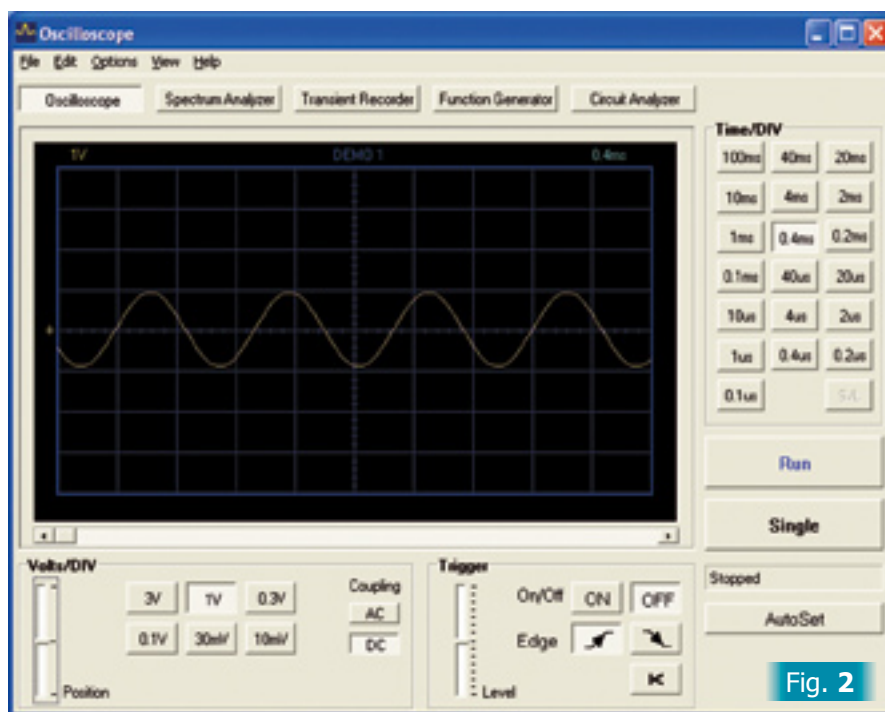
Fig. 1

rispondenti rispettivamente a LPT1, LPT2 e LPT3; chiaramente, bisogna fare clic sul pulsante di opzione corrispondente a una delle porte presenti nel PC (di solito LPT1, ovvero 378 H).

Confermando con OK si passa alla schermata di controllo dell'oscilloscopio; notate che se non è stato collegato alcuno strumento alla porta parallela, all'avvio verrà segnalato che non è possibile utilizzare alcun dispositivo: una finestra di avviso vi inviterà a controllare le connessioni e l'alimentazione. Ciò non accade se, nel terzo riquadro della finestra di dialogo principale, fate clic su Demo Mode. Selezionando quest'ultima modalità potrete simulare l'uso del programma senza comandare alcunché, quindi senza dover collegare alcuno strumento alla porta parallela. Una finestra informa che facendo clic su RUN si può iniziare la dimostrazione. La modalità Demo serve a prendere dimestichezza con le funzioni dell'oscilloscopio e dell'eventuale generatore di funzioni.

Il pannello di controllo

Visto che qui si parla dell'oscilloscopio, diamo uno sguardo alle sole funzioni inerenti l'interfaccia K8031. Come appena detto, quello che appare dopo la conferma delle impostazioni è il pannello di comando dell'oscilloscopio (Fig. 2): la schermata corrispondente rappresenta uno strumento come quelli tradizionali, con tanto di schermo



graduato, pulsanti di selezione della base dei tempi, due riquadri per l'ampiezza (volt/divisione) dei canali CH1 e CH2 e i selettori per il trigger (ON/OFF e fronte di aggancio). Due cursori graduati, ciascuno collocato nel riquadro di uno dei canali di ingresso, permettono di spostare la traccia verticalmente: basta farvi clic e muovere il puntatore in alto o in basso, rilasciando il pulsante sinistro del mouse nel punto voluto. Allo stesso modo si muove il terzo cursore, quello nella zona del trigger (Trigger Level). A riguardo, si noti che, essendo il nostro circuito ad un solo canale, i comandi relativi a CH2 non vanno utilizzati.

In alto, la videata riporta i pulsanti

con i quali attivare le altre modalità: Spectrum Analyzer, Transient Recorder, Function Generator, Circuit Analyzer.

È quasi superfluo dire che la forma d'onda del segnale applicato all'ingresso appare nello schermo con il colore della rispettiva scritta: la traccia gialla è quella del CH1, mentre in verde appare quella del CH2 (non usata, perché il K8031 è a singolo canale).

Se l'immagine appare instabile dovrete agire sul livello di trigger, ovvero provare a vedere se la situazione migliora facendo clic su INT o tornando su CH1. Per aumentare l'ampiezza basta selezionare un valore più basso di volt/divisione; per

>

per il

MATERIALE

L'oscilloscopio per PC presentato in queste pagine è disponibile in scatola di montaggio (cod. K8031) al prezzo di 144,00 Euro. Il kit comprende tutti i componenti, la basetta forata e serigrafata, il contenitore e le minuterie. Non sono compresi l'alimentatore e la sonda. L'oscilloscopio è anche disponibile già montato e collaudato (cod. PCS100A) al prezzo di 170,00 Euro. In questo caso la fornitura comprende anche l'alimentatore e una sonda.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futuranet.it>

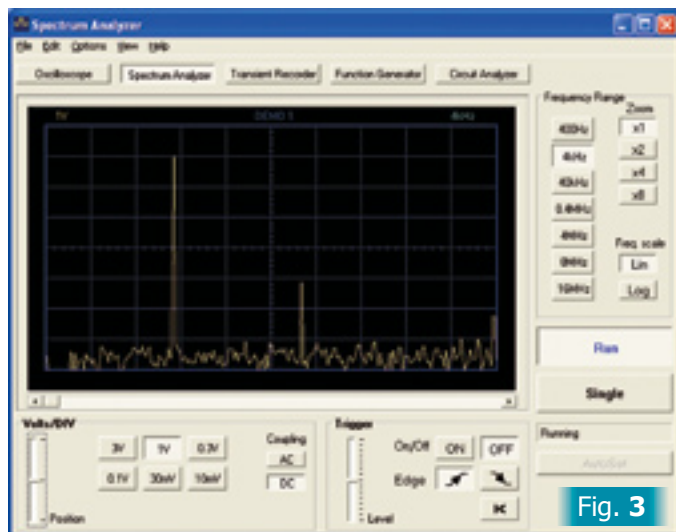


Fig. 3

estendere la forma d'onda orizzontalmente scegliete un appropriato valore nella sezione time/div. Insomma, come con un normale oscilloscopio.

L'analizzatore di spettro

Questa modalità si attiva facendo clic sul pulsante Spectrum Analyzer, che mostra in primo piano la relativa schermata: sostanzialmente il pannello di un classico analizzatore di spettro con tanto di schermo graduato (Fig.3).

Il pannello somiglia molto a quello dell'oscilloscopio, in cui risulta utilizzabile un solo canale (CH1); al posto della base dei tempi, però, trovate una serie di pulsanti con i quali è possibile impostare la gamma di frequenza visualizzata. Il va-

di banda, più elevata risulta la compressione di quanto visualizzato con conseguente minore leggibilità della schermata.

Particolarmente interessanti sono le funzioni di zoom e impostazione della scala graduata: la prima permette di andare ad analizzare dettagli anche molto piccoli della curva di risposta; la seconda consente all'utente di decidere se rappresentare il risultato dell'analisi in formato lineare oppure logaritmico.

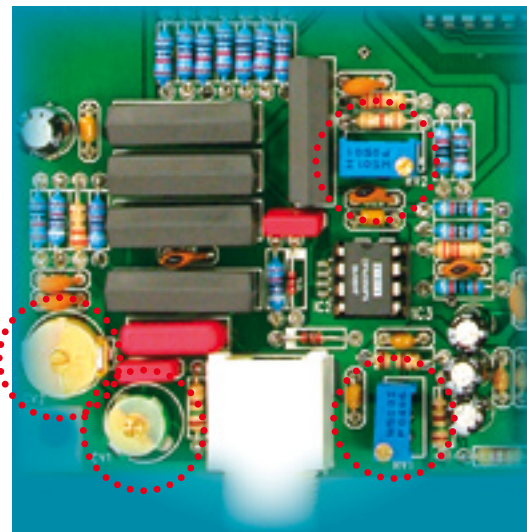
Quest'ultimo è quello usato per il tracciamento dei diagrammi di Bode e, in generale, per le curve di risposta in frequenza di amplificatori, filtri e dispositivi audio o di media frequenza. L'analisi di spettro permette altresì di visualizzare il livello delle armoniche componenti una forma d'onda non sinusoidale, ovvero di verificare la presenza e l'ampiezza delle sinusoidi elementari di frequenza multipla rispetto a quella del segnale periodico in analisi.

lore massimo riportato nella schermata è di 25 MHz, ma di fatto, presentando la nostra interfaccia un valore massimo di 12 MHz, il valore più alto non può superare questo limite. Maggiore è la larghezza

Il registratore di transienti

Viene attivato facendo clic sul pulsante funzione Transient Recorder: la relativa schermata (Fig.4) mostra ancora lo schermo graduato, le sezioni volt/divisione del canale CH1

Particolare del circuito stampato in cui sono visibili i due condensatori variabili ed i due trimmer multigiri che vanno regolati per tarare l'oscilloscopio.



(traccia gialla) e, in alto a destra, una serie di pulsanti con i quali scegliere il tempo di registrazione.

Lo strumento così attivato permette di memorizzare l'andamento delle tensioni dei due canali (o anche di uno solo di essi) entro l'intervallo di tempo corrispondente al pulsante sul quale avete fatto clic: ad esempio, premendo 10s la registrazione riguarda un arco di 10 secondi. Chiaramente l'intervallo viene conteggiato a partire da quando fate clic sul pulsante RUN.

Il registratore di transienti è una sorta di oscilloscopio a memoria, utile quando si deve monitorare quel che accade in un circuito soggetto a variazioni casuali, difficilmente riscontrabili in una misura normale: ad esempio caricabatterie, alimentatori switching, driver di carichi induttivi, linee elettriche, alimentazioni dei computer ecc.

Nell'uso del registratore ricordate

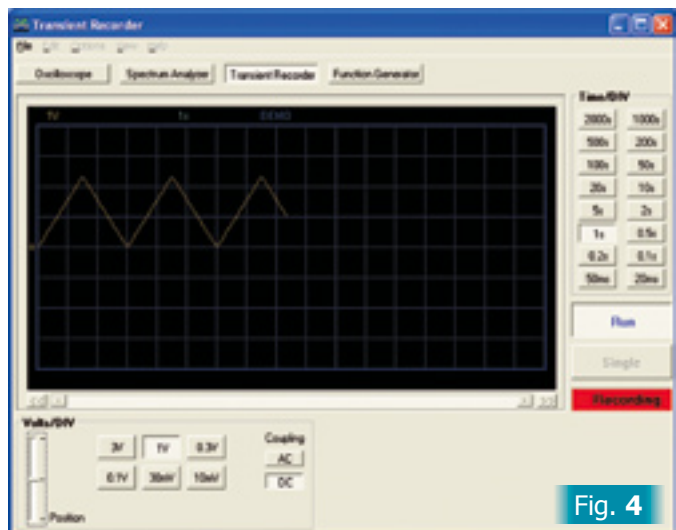


Fig. 4

Fig. B

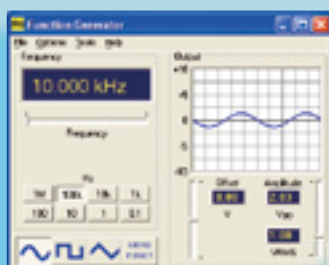


Fig. A

L'interfaccia grafica PCLAB 2000 è in grado di controllare due dispositivi collegati contemporaneamente alla porta parallela. Un accoppiamento molto interessante è costituito da un oscilloscopio, come il nostro K8031, e da un generatore di funzioni, come il PCG10 (presentato sul fascicolo n. 68 di Elettronica). Attivando dal menu principale di PCLab2000 (sezione *LPT - Devices*) i dispositivi K8031 e PCG10 ed entrando nella schermata principale facendo clic su OK, si renderanno disponibili due nuovi pulsanti di scelta (Fig. C, zona in alto a destra): "Function Generator" e "Circuit Analyzer". Il primo attiva la finestra di controllo del PCG10 (Fig. A), mentre il secondo apre la finestra "Bode Plotter" (Fig. D), molto utile per verificare la risposta in frequenza dei circuiti analogici. In questa modalità, il generatore di funzioni e la sezione "Analizzatore di spettro" lavorano sinergicamente in un ciclo in cui il generatore di funzioni emette una frequenza nota e l'analizzatore di spettro la controlla e la visualizza sul diagramma "Bode Plotter". In Fig. D è visibile la risposta in frequenza del filtro passa-banda passivo di Fig. B, analizzato nel

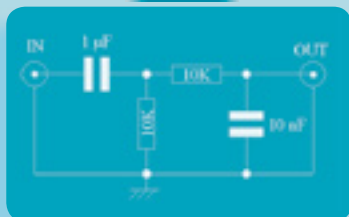


Fig. D



Fig. C

range di frequenze 1 Hz ÷ 10 kHz con 10 misurazioni per ogni decade e visualizzato su scala logaritmica. Questa funzione è particolarmente utile per verificare la banda passante dei filtri, il guadagno di amplificatori, preamplificatori e controlli di toni, ed in generale per verificare il comportamento di tutti i dispositivi che operano solo con segnali analogici.

A coloro che utilizzeranno la funzione "Bode Plotter" vogliamo ricordare alcune particolarità delle misure riguardanti la risposta in frequenza:

- Una decade è una gamma di frequenze in cui il valore massimo vale 10 volte quello minimo, ad esempio 10÷100 Hz, 10÷100 kHz, 20÷200 MHz;
- Normalmente (ma non è una regola), la misurazione viene effettuata con 10 frequenze per decade; tali frequenze si ottengono impostando incrementi uguali alla frequenza minima: ad esempio, la decade 10÷100 Hz è stata analizzata a salti di 10 Hz partendo da 10 Hz: la 1÷10 kHz a salti di 1 kHz...
- Nella rappresentazione su scala logaritmica, le decadi hanno sempre la stessa ampiezza (vedere Fig. D) e ciò rende il grafico disegnato su di esse molto più comprensibile di come lo sarebbe su scala lineare.

che maggiore è il tempo scelto, più è lo spazio occupato dai rispettivi dati nel disco rigido del computer.

Taratura dell'oscilloscopio

Bene, adesso che conoscete il programma di gestione, potete alimentare lo strumento e connetterlo al

computer: accendete quest'ultimo ed avviate PCLab2000. Scelta la funzione Oscilloscopio, procedete con la taratura, operazione indispensabile a garantire il miglior funzionamento possibile dell'interfaccia K8031. Le regolazioni sono suddivise in più fasi, la prima delle quali riguarda l'offset; per effettuarla

bisogna cortocircuitare l'ingresso (BNC) e fare clic sul pulsante RUN nella finestra di dialogo del programma, dopo aver selezionato l'accoppiamento in DC e il Trigger in OFF. Fatto ciò, portate il cursore della posizione verticale (Y) in posizione centrale, impostate come sensibilità verticale un valore compreso tra 1 e ➤

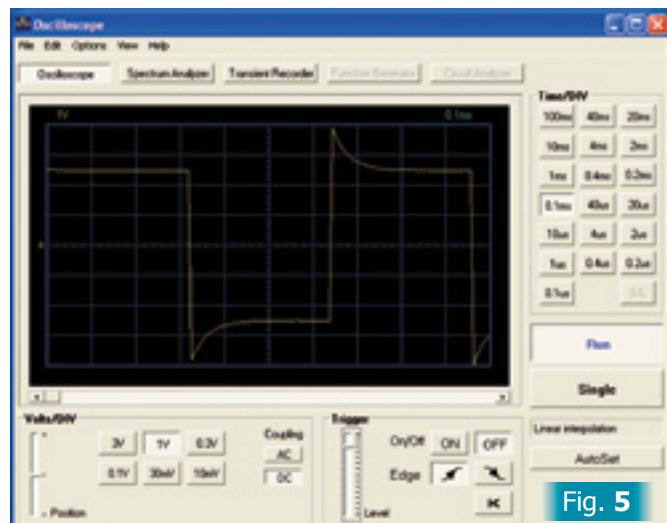


Fig. 5

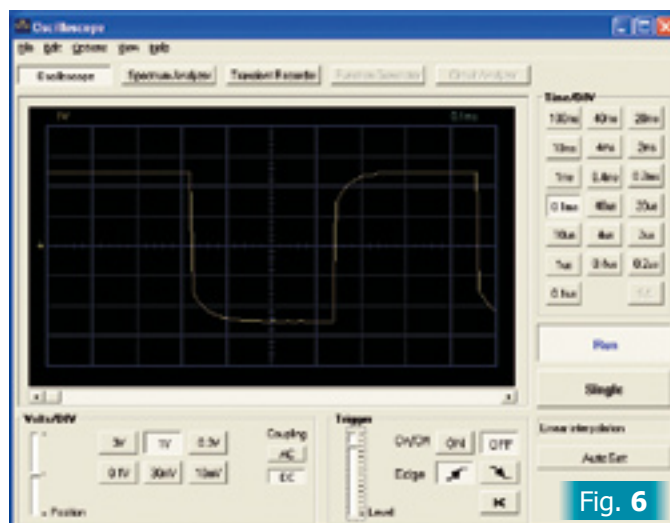


Fig. 6

A completamento del software di gestione è disponibile una DLL che permette, quando lo strumento è in funzione, di trasferire direttamente in un foglio di lavoro di Microsoft Excel i dati corrispondenti all'acquisizione del segnale; più esattamente, la DLL è compresa nel programma PCLab2000 e viene da esso installata nella cartella di sistema di Windows (Windows\System per Windows 95, 98, Me oppure System32 per Windows NT, 2000, XP). Con essa è possibile creare, in Excel, una macro che funziona da emulatore di tastiera: ogni volta che il programma acquisisce dall'interfaccia una lettura, ne trasferisce i dati in una cella del foglio di lavoro.

Per creare la macro, dopo aver installato PCLab2000, aprite Excel e, dal menu File impartite il comando Nuovo (potete fare lo stesso, mediante la scorciatoia da tastiera Ctrl+n); dal sottomenu Barre degli strumenti del menu Visualizza, impartite il comando Forme.

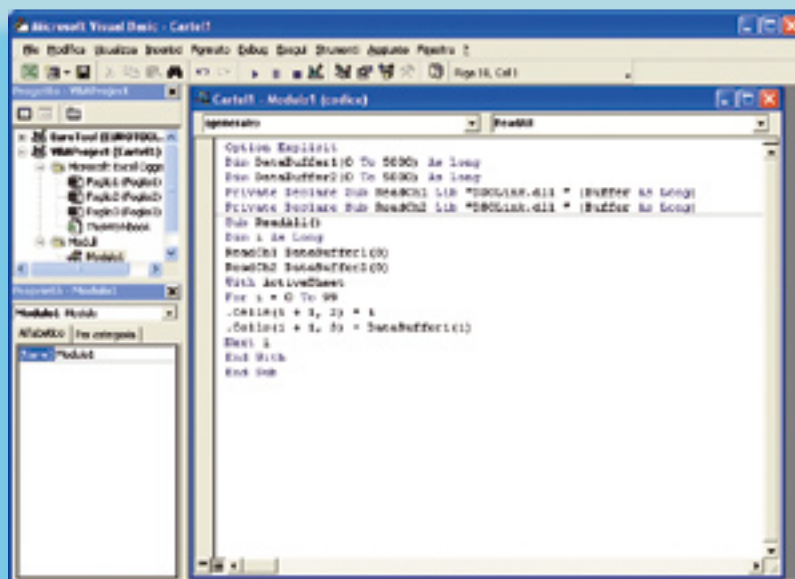
Appare così l'omonima barra degli strumenti; in essa fate clic sul pulsante (Pulsante) e vedrete il puntatore del mouse diventare una piccola croce. Automaticamente il programma passa nel foglio di lavoro da voi creato, dove dovete fare clic e trascinare il puntatore fino a costruire il rettangolo nel quale volete che, nel normale utilizzo del foglio di lavoro, il pulsante appaia; rilasciando il pulsante sinistro del mouse appare la finestra di dialogo Assegna Macro, con la quale assegnerete la macro al pulsante.

Fate clic nella casella Nome macro e scrivete il nome ReadAll, quindi fate clic sul pulsante New.

Così facendo accedete alla finestra di dialogo di Microsoft Visual Basic Editor: in essa vedete che è stata creata una subroutine chiamata ReadAll.

Nel riquadro contenente le righe di comando della subroutine sostituite il listato predefinito:

```
Sub ReadAll()  
End Sub
```



con quanto riportato nel riquadro seguente:

```
Option Explicit  
Dim DataBuffer1(0 To 5000) As Long  
Dim DataBuffer2(0 To 5000) As Long  
Private Declare Sub ReadCh1 Lib "DSOLink.dll" (Buffer As Long)  
Private Declare Sub ReadCh2 Lib "DSOLink.dll" (Buffer As Long)  
Sub ReadAll()  
Dim i As Long  
ReadCh1 DataBuffer1(0)  
ReadCh2 DataBuffer2(0)  
With ActiveSheet  
For i = 0 To 99  
.Cells(i + 1, 2) = DataBuffer1(i)  
.Cells(i + 1, 3) = DataBuffer2(i)  
Next i  
End With  
End Sub
```

3 V/div., quindi ruotate in un verso o nell'altro il cursore del trimmer RV1 fino a portare la traccia esatta-

mente al centro dello schermo (sulla linea mediana dell'altezza).

Regolato l'offset, si può passare alla calibrazione dei fronti di salita, ossia dell'accoppiamento capacitivo dell'ingresso; allo scopo bisogna rimuovere il cortocircuito dall'input, quindi portare la sonda (x1) sul test-point J4 dove viene reso disponi-

bile l'apposito segnale di calibrazione. Dopo aver scelto una base dei tempi di 0,5 μ s/div. e una sensibilità verticale di 1 V/div. agite sul compensatore CV1 al fine di ottenere fronti di salita dritti e angoli ben squadri; a tale riguardo va detto che ruotando il cursore in un verso la forma d'onda presenterà gli angoli smussati, mentre nel verso opposto si vedranno apparire delle sovraelongazioni in corrispondenza degli stessi angoli, ovvero dopo la salita e la discesa della tensione (Figura 1).

Fatto ciò, nel menu VIEW del programma di gestione impartite il comando RMS value; regolate, ora, RV2 fin quando il segnale non rag-

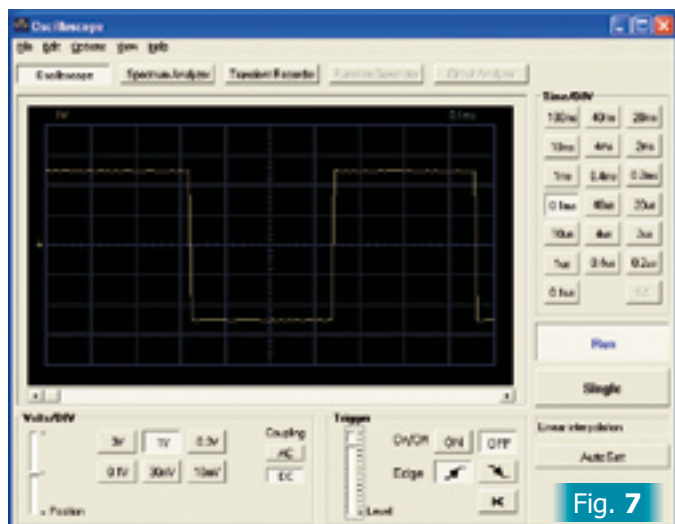


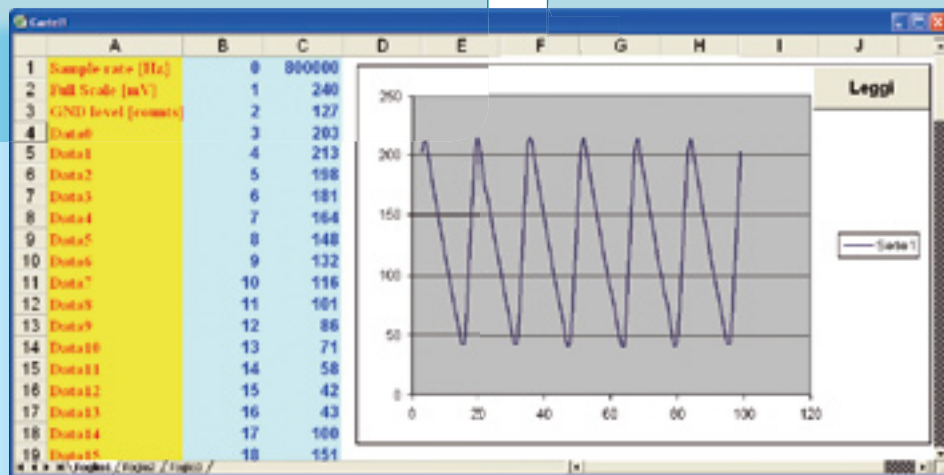
Fig. 7

Premete insieme i tasti Alt ed F11 per tornare ad Excel e scrivete il seguente testo nella colonna A, partendo dalla prima riga disponibile (una riga per casella):

Sample rate [Hz]
Full Scale [mV]
GND level [counts]
Data0
Data1
Data2
 ...

Nel programma PCLab2000, avviate la funzione Oscilloscopio e fate clic su RUN o SINGLE; ora andate al foglio di lavoro di Excel da voi creato e in esso fate clic sul pulsante che avete costruito: la macro corrispondente verrà eseguita e nelle colonne B e C appariranno i dati descritti nella A.

Notate che le prime tre righe esprimeranno le impostazioni dello strumento e le seguenti le letture dell'A/D converter dell'interfaccia corrispondenti ciascuna ad uno dei punti costituenti la forma d'onda visualizzata nello schermo virtuale del PCLab2000; le letture sono sempre riportate come valore decimale (da 0 a 255) del dato binario formato dagli otto bit d'uscita del convertitore A/D.



giunge 2,5 V RMS, quindi impostare 0,3 V/div. e connettere la sonda (sempre in x1) al test-point J3. Agendo sul cursore della posizione Y, portate la traccia esattamente a metà altezza dello schermo virtuale dell'oscilloscopio; ora bisogna agire sul compensatore CV2, ruotandone la vite fin quando il segnale non tor-

na netto come quando avete regolato lo strumento mediante CV1, ovvero con gli angoli ben squadrati, senza arrotondamenti o sovraelongazioni (Fig.7).

La procedura deve essere ripetuta almeno una volta: quindi tornate alla funzione oscilloscopio, portate la sonda su J4 e verificate che la forma

PCLab2000 offre agli utenti esperti la possibilità di realizzare interfacce software personalizzate adatte alle più disparate esigenze; la cosa può essere fatta grazie ad una DLL chiamata PCS100D.DLL scaricabile gratuitamente dal sito Web www.velleman.be. PCS100D.DLL è una DLL a 32 bit di Windows. Richiamando le funzioni e le procedure esportate da essa, si possono scrivere, per Windows 98SE, 2000, Me, XP, applicazioni in Delphi, Visual Basic, C++ Builder o per qualsiasi sistema di sviluppo a 32 bit per Windows. Di seguito riportiamo una breve panoramica delle procedure e delle funzioni esportate dalla PCS100D.DLL follows; gli esempi e le procedure sono scritti in Delphi. Il mese prossimo dedicheremo ampio spazio allo sviluppo di programmi personalizzati con questi strumenti.

Si parte con le impostazioni generali di comunicazione tra interfaccia e PC:

```
Start_PCS100 Apre il link di comunicazione con il dispositivo K8031
Stop_PCS100 Chiude il predetto link
RunOn(Run: Boolean) Set PCS100 Avvia la modalità on o off
SingleOn(Single: Boolean) Imposta la modalità Single on o off per il K8031
SelectPrinterPort(Printer_no: Longint) Definisce l'indirizzo della porta LPT
Show_PCS100(Visible: Boolean); Mostra o nasconde l'interfaccia utente del K8031
```

Quelle che seguono sono, invece, le impostazioni per le misure:

```
Voltage(Volts:Longint) Imposta la scala V/div
SampleRate(Rate:Longint) Imposta la scala Time/div
TriggerLevel(TrgLevel:Longint) Imposta il livello di trigger
TrgEdge(Positive_Negative:Longint) Imposta il fronte del trigger
TrgOn(trg_on: Boolean); Trigger on/off
YPosition(y_pos:Longint); Imposta Y-position
TriggerLevel(TrgLevel:Longint) Imposta il livello di trigger
TriggerLevel(TrgLevel:Longint) Imposta il livello di trigger
Coupling(AC_DC_GND:Longint) Imposta l'accoppiamento dell'ingresso
```

Infine, ecco la parte riguardante la lettura dei dati:

```
ReadData(Buffer: Pointer; Count: Longint): Boolean; legge i dati dal K8031
```

d'onda rettangolare abbia gli angoli squadrati; in caso contrario agite nuovamente sul compensatore CV1. Tornate alla funzione RMS value e, se occorre, regolate nuovamente CV2. A questo punto aprite il menu File e in esso impartite il comando Calibrate and exit; l'interfaccia da PC è tarata ed è pronta per l'uso.

Conclusioni

Abbiamo così ultimato la presentazione dell'interfaccia grafica che, come avete potuto verificare, è molto semplice ed intuitiva. Per i lettori più preparati sono anche disponibili specifiche librerie che permettono di realizzare interfacce software personalizzate, differenti dal PCLAB2000. Nel prossimo numero ce ne occuperemo in dettaglio. Appuntamento, dunque, al mese prossimo.

Localizzatore Mambo, il set di comandi

di Alessandro Orlando

Il localizzatore GPS portatile Mambo dispone di un set di comandi di configurazione molto completo e versatile. Iniziamo a conoscere ed utilizzare tali comandi proponendo una serie di esempi pratici che consentono di prendere confidenza sia con la sintassi che con le modalità di utilizzo.

Chi ci segue da qualche tempo sa che la tecnologia GPS ci ha sempre affascinato, al punto che in quasi tutti i numeri c'è un articolo o un progetto dedicato a questo argomento. Tra i tanti sistemi di cui ci siamo occupati uno in particolare ci ha colpito per potenza, versatilità e prestazioni: è il Mambo, sistema di localizzazione integrato prodotto dalla tedesca Falcom che, in un compatto contenitore, accoglie un ricevitore GPS a 20 canali, un modem/cellulare GSM e GPRS, un'interfaccia Bluetooth, la pulsantiera e la batteria; si tratta di un dispositivo completamente programmabile tramite un

set di comandi molto completo ed evoluto, set utilizzabile per mezzo di un ambiente di lavoro grafico (Workbench) fornito a corredo del localizzatore. Nel primo articolo dedicato al Mambo, pubblicato sul fascicolo 116, abbiamo presentato il prodotto descrivendone le caratteristiche e le possibilità di impiego a livello generale; nel fascicolo successivo (rivista 117), invece, abbiamo analizzato in dettaglio il Workbench, ossia il software allegato al prodotto, col quale è possibile (con l'ausilio di un PC) programmare il dispositivo in funzione delle nostre esigenze nonché visualizzare i dati che esso può fornir-

Tabella 1 - Struttura base del comando

Stringa iniziale	Comando	Parametro	Checksum	Fine sequenza
\$PFAL	< Comando >	< Parametro >	< *CKSUM >	< CR > < LF >
\$PFAL	< Comando >	< Parametro >	Nessuno	< CR > < LF >
PFAL	< Comando >	< Parametro >	< *CKSUM >	< CR > < LF >
PFAL	< Comando >	< Parametro >	Nessuno	< CR > < LF >

re, quali la posizione calcolata dalla sezione GPS (comunicata in formato NMEA0183) oppure le informazioni relative al gestore di telefonia mobile. In questo articolo forniamo invece una panoramica dei comandi e della relativa sintassi, presentando anche alcuni semplici esempi. Scopriamo insieme il DNA del sistema.

<c_type>,<c_index>,<c_subindex> oppure
 <c_type>,<c_index>=<value> oppure
 <c_type>,<c_index>,<c_subindex>=<value>.

In ognuna delle tre forme, *c_type* specifica il tipo di comando, *c_index* e *c_subindex* indicano rispettiva-

ce rappresentato dal punto (“.”). Il parametro “*c_type*”, come abbiamo già anticipato, specifica il tipo di comando cui sono destinati i parametri. In realtà, sarebbe più corretto dire che “*c_type*” specifica la periferica a cui il comando è destinato, a patto che vengano considerati come periferiche anche il file System (SYS) e la pagina di memoria in cui viene memorizzato il file di configurazione (CFG). Nella tabella 2 sono riportati tutti i possibili valori che “*c_type*” può assumere: ad esempio, se esso vale “GSM” significa che il comando è destinato a programmare un’azione o a leggere uno stato della sezione telefonica. Analogamente, se fosse “GPS”, il destinatario del comando sarebbe il ricevitore satellitare.

Sia dalle righe precedenti che dai capitoli pubblicati sulle riviste 116 e 117, si è compreso come il set di comandi sia molto nutrito, viste le innumerevoli possibilità operative fornite dal nostro versatile dispositivo. Nel corso di questo articolo, ci proponiamo di fornire, come già anticipato, un’anteprima di tutti i comandi, fornendo anche alcuni esempi molto semplici. Maggiori dettagli saranno invece forniti con le prossime puntate.

Tabella 2 - Descrizione campi

c_type	Specifica
Sys	- Determina la programmazione predefinita del file System in: - Operazioni di sistema quali: - Reset - Accensione / Spegnimento, ecc. - Inizializzazione / Interruzione dei processi di sistema quali: - Timer - contatori, ecc.
Cnf	- Determina la programmazione dei parametri che possono essere letti/scritti o cambiati ed è basato sugli eventi.
I/O	- Determina l’uso e la programmazione dei Led, Toni di avviso, Vibrazione, ecc.
GPS	- Determina la programmazione della parte GPS inclusa la lettura delle informazioni e dello storico contenuto nel logger interno, ecc
GSM	- Determina la programmazione della parte GSM, GPRS, SMS, VOCE, DATI ecc
TCP	- Determina la programmazione della connessione TCP inclusa connessione / disconnessione, ecc
MSG	- Determina la programmazione dei messaggi in uscita con protocollo GPS che devono essere inviati attraverso la seriale, la linea CSD oppure il TCP.

Struttura base dei comandi

Il localizzatore Mambo, per diventare operativo, deve essere precedentemente programmato e configurato tramite una serie di comandi la cui sintassi di base è descritta in Tabella 1. Il comando, che per semplicità possiamo abbreviare in “Cmd” (per inciso, è proprio ciò che Falcom ha fatto...) specifica in maniera dettagliata la destinazione, il tipo ed il valore di ogni istruzione. Analizzando il comando più in dettaglio, osserviamo che esso può presentare aspetti differenti:

mente pagina e sottopagina ed infine <value> specifica il valore numerico assegnato al comando. Sulla stessa riga di programma possono essere scritti più comandi “Cmd”, purché vengano separati dal simbolo “;”. Il separatore dei campi “pagina” e “sottopagina” è inve-

Comandi di Sistema (Sys)

Riportati nella Tabella 4, sono quelli più numerosi: tramite essi si interagisce con il kernel di sistema o, in parole povere, si programma abbastanza pesantemente il modo in cui il microprocessore principale di Mambo deve operare.

Utilizzando questi comandi è possibile istruire la CPU per eseguire un

Tabella 3 - Comandi di configurazione

\$PFAL,Cnf.Set, <parameter_name = value >	Setta o cambia la configurazione del Mambo.
\$PFAL,Cnf.Get, <parameter_name >	Controlla la configurazione del parametro specificato.
\$PFAL,Cnf.Clear, <parameter_name >	Azzerla la configurazione del parametro specificato.
\$PFAL,Cnf.ShowUser	Controlla la configurazione Utente.
\$PFAL,Cnf.ShowDefault	Controlla la configurazione di Default.
\$PFAL,Cnf.Show	Controlla la configurazione di tutti i parametri usati.
\$PFAL,Cnf.Search, <parameter_name >	Cerca la configurazione del parametro specificato.

Tabella 4 - Comandi di sistema

\$PFAL,Sys.Security.Lock,"password"	Blocca il sistema.
\$PFAL,Sys.Security.Unlock,"password"	Sblocca il sistema.
\$PFAL,Sys.Security.RemoveLock,"password"	Rimuove il blocco di sistema.
\$PFAL,Sys.RUpdate.Select.Sectors	Seleziona il settore da aggiornare.
\$PFAL,Sys.RUpdate.Select.Blocks	Seleziona il blocco all'interno del settore da aggiornare.
\$PFAL,Sys.RUpdate.Write, < mode >	Invia il blocco di comandi PFAL attraverso il sistema di comunicazione selezionato.
\$PFAL,Sys.RUpdate.Show.Progress	Mostra l'avanzamento dell'aggiornamento del Firmware.
\$PFAL,Sys.RUpdate.Show.CRC	Mostra il checksum di ogni settore appena aggiornato.
\$PFAL,Sys.RUpdate.Start	Esegue l'aggiornamento del firmware da remoto.
\$PFAL,Sys.Device.Reset	Resetta il sistema.
\$PFAL,Sys.Device.Shutdown	Spegne il sistema.
\$PFAL,Sys.Device.FactoryReset	Azzerla la configurazione utente ai valori di default.
\$PFAL,Sys.Device.Sleep = < value >	Pone il sistema a riposo sino al verificarsi dell'evento specificato.
\$PFAL,Sys.Device.WDOG = < value >	Abilita/disabilita il reset del sistema dopo lo sgancio del sistema o lo stop alla risposta.
\$PFAL,Sys.Device.ClearAlarm	Azzerla i timer utente e riavvia il sistema.
\$PFAL,Sys.Device.ClearConfig	Azzerla la configurazione utente e riavvia il sistema.
\$PFAL,Sys.Device.CfgUpdateMode	Pone il sistema in un particolare modo di aggiornamento per cambiare la configurazione degli allarmi.
\$PFAL,Sys.Timer< index > .Configure = < mode > , < timeout >	Configura il timer del sistema.
\$PFAL,Sys.Timer< index > .Start = < timer_settings >	Avvia/riavvia il timer del sistema.
\$PFAL,Sys.Timer< index > .Stop	Blocca il timer in corso.
\$PFAL,Sys.Timer< index > .Pause	Mette in pausa il timer in corso.
\$PFAL,Sys.Timer< index > .Resume	Riavvia l'esecuzione del timer messo in pausa.
\$PFAL,Sys.Timer< index > .Arm	Allarma un timer inizializzato.
\$PFAL,Sys.Timer< index > .Disarm	Disallarma un timer inizializzato ed allarmato.
\$PFAL,Sys.Timer< index > .Erase	Cancella la configurazione dei Timer.
\$PFAL,Sys.Timer< index > .Save< storage_index >	Salva lo stato di un Timer nell'indice della memoria.
\$PFAL,Sys.Timer< index > .Load< storage_index >	Carica lo stato di un Timer dall'indice della memoria.
\$PFAL,Sys.Timer< index > .State	Legge lo stato del timer usato.
\$PFAL,Sys.Trigger< index > = < state_type >	Attiva/disattiva un Trigger di sistema.
\$PFAL,Sys.Trigger< index >	Recupera lo stato del Trigger corrente.
\$PFAL,Sys.Trigger< index > .Save< storage_index >	Salva lo stato del Trigger nell'indice della memoria.
\$PFAL,Sys.Trigger< index > .Load< storage_index >	Carica lo stato del Trigger dall'indice della memoria.
\$PFAL,Sys.Counter< index > .Set = < value >	Setta il valore di un contatore.
\$PFAL,Sys.Counter< index > .Increment = < inc_value >	Incrementa il valore esistente di un contatore.
\$PFAL,Sys.Counter< index > .Decrement = < dec_value >	Decrementa il valore esistente di un contatore.
\$PFAL,Sys.Counter< index > .State	Legge lo stato di un contatore usato.
\$PFAL,Sys.Counter< index > .Save< storage_index >	Salva lo stato di un contatore nell'indice della memoria.
\$PFAL,Sys.Counter< index > .Load< storage_index >	Carica lo stato di un contatore dall'indice della memoria.
\$PFAL,Sys.Counter< index > .Clear	Setta uno specificato contatore a 0.
\$PFAL,Sys.Macro< index >	Attiva la macro.
\$PFAL,Sys.GSM.Enable	Accendi il modulo GSM.
\$PFAL,Sys.GSM.Disable	Spegni il modulo GSM.
\$PFAL,Sys.GSM.Reset	Resetta il modulo GSM.
\$PFAL,Sys.GPS.Enable	Accendi il modulo GPS.
\$PFAL,Sys.GPS.Disable	Spegni il modulo GPS.
\$PFAL,Sys.GPS.Reset	Resetta il modulo GPS.
\$PFAL,Sys.UserEvent< index >	Crea un evento per una specifica applicazione.
\$PFAL,Sys.Bat.Voltage	Richiede la tensione batteria.
\$PFAL,Sys.Bat.ChargeState	Controlla se la batteria è in carica.
\$PFAL,Sys.BT.Enable	Accendi il modulo Bluetooth.
\$PFAL,Sys.BT.Disable	Spegni il modulo Bluetooth.
\$PFAL,Sys.BT.Reset	Resetta il modulo Bluetooth.
\$PFAL,Sys.ZV.Reset	Resetta l'applicazione e pone il Mambo in Software Update.

reset, una inizializzazione, attivare e controllare il sensore di movimento, eseguire una sequenza di azioni predefinita con la quale si può personalizzare il modo di funzionamento del dispositivo. Sono i comandi più

potenti e flessibili nonché i più complessi da utilizzare.

Comandi di configurazione (Cnf)

Meno numerosi dei precedenti, essi sono stati riassunti in Tabella 3. De-

terminano la programmazione di tutti i parametri che possono essere letti/scritti o cambiati durante il normale funzionamento del sistema. L'utilizzo di questi comandi determina il comportamento del Mambo ➤

Tabella 5 - Comandi di gestione I/O

\$PFAL,IO.LED <index> , <color> = <config_type>	Setta e controlla il lampeggio dei Led.
\$PFAL,IO.Beep <index> = <config_type>	Setta e controlla i toni da generare.
\$PFAL,IO.Vibration = <config_type>	Setta il tipo di avviso a vibrazione.

Tabella 6 - Comandi GPS

\$PFAL,GPS.Nav.Position <buffer_index>	Controlla il percorso effettuato da un dato punto memorizzato nel buffer.
\$PFAL,GPS.Nav.Position <buffer_index> = <type>	Salva temporaneamente la posizione attuale o azzeri i dati esistenti dal buffer.
\$PFAL,GPS.Nav.Position <buffer_index> = save <storage_index>	Muovi i dati GPS dal buffer all'indice della memoria.
\$PFAL,GPS.Nav.Position <buffer_index> = load <storage_index>	Carica i dati GPS dall'indice della memoria al buffer per utilizzarli temporaneamente.
\$PFAL,GPS.Nav.Distance	Controlla la distanza dal punto di partenza.
\$PFAL,GPS.Nav.Distance = <value>	Setta/resetta il valore utente della distanza.
\$PFAL,GPS.Nav.SaveLastValid	Salva l'ultima posizione utile nel caso il GPS abbia un segnale non valido.
\$PFAL,GPS.History.Write, <add_prot_to_memory> , <"text">	Salva i dati della posizione GPS nello storico della memoria.
\$PFAL,GPS.History.Clear	Azzeri lo storico della memoria.
\$PFAL,GPS.History.GetStart	Controlla i vecchi dati memorizzati dallo storico della memoria.
\$PFAL,GPS.History.SetRead, <s_date> , <s_time> , <e_date> , <e_time>	Seleziona per data/ora i dati da leggere dallo storico della memoria.
\$PFAL,GPS.History.Read	Scarica il record selezionato dallo storico della memoria.
\$PFAL,GPS.Geofence.Park.Set	Poni/ Attiva il Parking.
\$PFAL,GPS.Geofence.Park.Remove	Disabilita il Parking.
\$PFAL,GPS.Geofence.GeoState, <geo_id>	Controlla lo stato del GEO Fence.
\$PFAL,GPS.Geofence.AreaState, <area_id>	Controlla lo stato di un'area specifica.

in funzione della rilevazione degli eventi (perdita del fix, rilevazione di un movimento, ricezione di un SMS...). Non agiscono, però, direttamente su tali eventi, bensì solo sulla locazione di memoria di sistema in cui vengono definiti i "modelli di comportamento" del localizzatore, ossia il modo in cui il sistema deve reagire.

Comandi di gestione I/O (IO)

Come da Tabella 5, sono i meno numerosi. Essi agiscono solo ed esclusivamente sulle segnalazioni che l'utente può percepire ossia i led, i toni emessi dal buzzer e l'avviso a vibrazione.

Comandi GPS (GPS)

Dal nome si intuisce facilmente che essi sono dedicati all'interazione con il ricevitore satellitare. Poiché le informazioni che il ricevitore GPS è in grado di fornire sono numerose, ne consegue che anche il numero di comandi è piuttosto alto (Tabella 6). Va detto che sono tutti comandi molto specializzati, che permettono

di eseguire funzioni più riconducibili, come concetto, a delle macro

Tabella 7 - Comandi GSM

\$PFAL,GSM.PIN = <"pin">	Immetti il PIN della SIM usata.
\$PFAL,GSM.PUK = <"puk"> , <"pin">	Immetti il PIN ed il PUK .
\$PFAL,GSM.IMEI	Controlla il seriale del dispositivo.
\$PFAL,GSM.SIMID	Controlla l'ID della SIM.
\$PFAL,GSM.OwnNumber	Controlla il numero di telefono del chiamante.
\$PFAL,GSM.Balance	Controlla le informazioni della SIM.
\$PFAL,GSM.VoiceCall.Dial, <"p_number">	Effettua una chiamata Voce.
\$PFAL,GSM.VoiceCall.Accept	Rispondi ad una chiamata voce.
\$PFAL,GSM.VoiceCall.Hangup	Attacca ad una chiamata voce.
\$PFAL,GSM.VoiceCall.Channel	Setta il percorso Audio.
\$PFAL,GSM.VoiceCall.Ringtone = <type> , <volume>	Setta o controlla le impostazioni della suoneria.
\$PFAL,GSM.VoiceCall.Volume.Speaker = <n_gain> , <volume> , <sidetone>	Setta o controlla le impostazioni dell'altoparlante.
\$PFAL,GSM.VoiceCall.Volume.Microphone = <gain> , <n_gain>	Setta o controlla le impostazioni del microfono.
\$PFAL,GSM.SMS.Send, <"p_number"> , <protocols> , <"text">	Manda un SMS al numero telefonico predefinito.
\$PFAL,GSM.SMS.Inbox.Clear	Pulisci la cartella degli SMS in arrivo.
\$PFAL,GSM.SMS.Inbox.State	Controlla tutti gli SMS in arrivo.
\$PFAL,GSM.SMS.Outbox.Clear	Pulisci la cartella degli SMS spediti.
\$PFAL,GSM.SMS.Outbox.State	Controlla gli SMS spediti.
\$PFAL,GSM.DataCall.Sent, <protocols> , <"text">	Manda un messaggio ad un Modem GSM attraverso la chiamata Dati stabilita.
\$PFAL,GSM.DataCall.Accept	Rispondi ad una chiamata dati.
\$PFAL,GSM.DataCall.Hangup	Aggancia ad un chiamata dati.
\$PFAL,GSM.GPRS.Connect	Effettua una connessione GPRS.
\$PFAL,GSM.GPRS.Disconnect	Disconnetti GPRS.
\$PFAL,GSM.GPRS.State	Controlla lo stato del GPRS.
\$PFAL,GSM.GPRS.Traffic = <complete> , <incoming> , <outgoing>	Setta o controlla il contatore GPRS.

(elenchi di istruzioni) più che a dei comandi veri e propri: ad esempio, una sola istruzione permette di cancellare lo storico della navigazione, piuttosto che calcolare e rendere la distanza rispetto ad un punto di partenza.

Comandi GSM (GSM)

Analogamente a quelli per i GPS, essi operano sulla sezione GSM, sia a livello di configurazione (ad esempio l'inserimento del PIN della SIM) che di gestione delle chiamate entranti ed uscenti o di generazione e ricezione di messaggi SMS. Anche in questo caso, come da Tabella 7, siamo in presenza di un congruo numero di opzioni.

Comandi TCP (TCP)

Sono riportati in Tabella 8; come si intuisce dal nome, essi consentono di operare a livello TCP su reti dati: mediante il loro utilizzo, è possibile effettuare connessioni e file transfer

Tabella 8 - Comandi TCP

\$PFAL,TCP.Client.Connect	Effettua una connessione TCP verso un server remoto.
\$PFAL,TCP.Client.Disconnect	Disconnetti il TCP da un server remoto
\$PFAL,TCP.Client.State	Controlla lo stato della connessione TCP.
\$PFAL,TCP.Client.Send,< protocols >,< "text" >	Invia un pacchetto TCP al server remoto connesso.
\$PFAL,TCP.Storage.Dispatch	Sposta le informazioni correnti dalla memoria TCP al Buffer in uscita.
\$PFAL,TCP.Storage.Clear	Azzerà il contenuto della memoria TCP creata.
\$PFAL,TCP.Storage.AddProtocol,< protocol >,< "text" >	Scrivi il protocollo specificato e/o un testo utente nella memoria TCP.
\$PFAL,TCP.Storage.AddRecord,< protocol >,< "text" >	Aggiungi un record alla memoria TCP.

verso i server remoti, piuttosto che inviare o ricevere pacchetti di dati. Ad esempio, utilizzando il Mambo come sistema per la gestione ed il controllo di flotte di autoveicoli aziendali, probabilmente l'utilizzo delle connessioni TCP sarebbe molto più efficace ed economicamente conveniente rispetto all'utilizzo degli SMS o delle chiamate dati.

Comandi di comunicazione e gestione messaggi (MSG)

Anche in questo caso si tratta di una lunga lista di comandi (Tabella 9), utilizzabili per comporre, manipolare e controllare le segnalazioni sotto forma di messaggi di testo. Oltre alle suddette funzioni, essi agiscono su elementi secondari, ma non per questo meno utili, quali data e ora di sistema piuttosto che sull'identificativo del software.

Alcuni facili esempi

Dalla lista dei comandi appare evidente che per ottenere un determinato comportamento da parte del Mambo occorre impartire una specifica istruzione. Alcune istruzioni sono piuttosto semplici e la loro funzione è intuitiva. Per esempio per accendere il modulo GSM del Mambo l'istruzione relativa è:

\$PFAL,Sys.GSM.Enable

Come visibile in tabella, l'accensione e lo spegnimento del modulo GSM fa parte dei comandi di sistema (Sys) quindi occorre utilizzare il c_type Sys per accedere alla sezione del sistema, GSM per accedere alla

Tabella 9 - Comandi di comunicazione e gestione messaggi

\$PFAL,MSG.Send.Serial,< protocols >,< "text" >	Invia il protocollo specificato + le informazioni di sistema attraverso la seriale.
\$PFAL,MSG.Send.CSD,< protocols >,< "text" >	Invia il protocollo specificato più le informazioni di sistema attraverso la connessione dati stabilita.
\$PFAL,MSG.Send.TCP,< protocols >,< "text" >	Invia il protocollo specificato più le informazioni di sistema via TCP attraverso il server remoto connesso.
\$PFAL,MSG.Mode.Serial=< out_sys_messages >,< mode >	Controlla o reinoltra i messaggi di sistema via SERIALE.
\$PFAL,MSG.Mode.CSD=< out_sys_messages >,< mode >	Controlla o reinoltra i messaggi di sistema via CSD.
\$PFAL,MSG.Mode.TCP=< out_sys_messages >,< mode >	Controlla o reinoltra i messaggi di sistema via TCP.
\$PFAL,MSG.Version.Complete	Controlla tutte le versioni della periferica.
\$PFAL,MSG.Version.Modules	Controlla la versione dei moduli della periferica.
\$PFAL,MSG.Version.InternalComm	Controlla la versione della DSP della periferica.
\$PFAL,MSG.Version.HardwareRev	Controlla la revisione Hardware della periferica.
\$PFAL,MSG.Version.Hardware	Controlla la versione Hardware della periferica.
\$PFAL,MSG.Version.Software	Controlla la versione Software della periferica.
\$PFAL,MSG.Version.SoftwareID	Controlla l'ID del Software.
\$PFAL,MSG.Info.ServerLogin	Identifica la periferica al server Falcom.
\$PFAL,MSG.Info.Protocol,< protocols >,< "text" >	Invia il protocollo specificato al mittente.
\$PFAL,MSG.Info.Time	Controlla l'orario, data, ecc. in formato UTC.
\$PFAL,MSG.Info.Alarm,< alarm_index >	Invia le informazioni dell'allarme specificato al mittente.
\$PFAL,MSG.Channel.SerialGSM	Indirizza i comandi in ingresso direttamente dalla seriale al modulo GSM.

sottopagina riservata alla sezione GSM ed infine il parametro *Enable* per l'accensione.

Allo stesso modo se nell'esecuzione di un comando, eventualmente collegato ad un evento (AL) che vedremo più in dettaglio in seguito, si dovesse rispondere ad una chiamata in maniera automatica e fare anche altre operazioni, la stringa di comando che ne consegue sarebbe:

\$PFAL,GSM.< Tipo_chiamata >.accept

dove *tipo-chiamata* specifica se è voce (Voice_call) o dati (Data_call). Anche in quest'ultimo caso il comando è molto semplice, ma sfortunatamente la sintassi non è sempre così facile e intuitiva, soprat-

tutto quando bisogna distinguere i comandi impartiti ed eseguiti una sola volta da quelli che fanno parte di un listato, che quindi vengono eseguiti ogni volta che si accende il localizzatore.

Un altro importante comando, basato sul verificarsi degli eventi, è "AL". Questa particolare funzione

programmata nella configurazione utente è disponibile ad ogni accensione del Mambo ed attiva una specifica azione quando si verifica una condizione predefinita.

La sintassi del parametro è:

**AL < indice > = < Condizione > :
< Azione > .**

Se, ad esempio, volessimo trasmettere un messaggio ad un numero prefissato nel caso in cui il segnale GPS non riuscisse a fare il fix, dovremmo inserire all'interno del listato la stringa o dichiarazione di allarme di Tabella 10, riportata nella pagina seguente.

Sempre a titolo di esempio, per ottenere l'invio di una chiamata voce ➤

Tabella 10

\$PFAL,Cnf.Set,ALO=GPS.eFix=invalid:GSM.SMS.Send,"+39328.....",0,"Segnale GPS non Valido"	
ALO	Allarme n°0
GPS.eFix=invalid	Se il segnale GPS non è valido
GSM.SMS.Send,"+39328.....",0,"Segnale GPS non Valido"	Spedisci al numero +39328..... Il messaggio "Segnale GPS non Valido".

Tabella 11

\$PFAL,Cnf.Set,AL1=IO.BTN.e0=short:GSM.VoiceCall.Dial,"+39328....."	
AL1	Allarme n°1
IO.BTN.e0=short	Se premo brevemente il tasto centrale
GSM.VoiceCall.Dial,"+39328....."	Invia una chiamata al numero +39328.....

di emergenza mediante la pressione del tasto centrale del dispositivo (quello grande), si deve utilizzare il comando riportato in tabella 11. Relativamente a questo, se anziché *short* avessimo scritto *long*, oppure *double*, avremmo dovuto premere il pulsante per oltre 2 secondi per *long* e due volte nell'arco di 500 msec nel caso di *double*. Ne consegue che la funzione "AL" è molto importante poiché consente di effettuare azioni predefinite in funzione di eventi predefiniti.

Nello stesso modo, per fare un altro esempio, si può dire al localizzatore di memorizzare tutte le posizioni GPS in mancanza di segnale GSM valido oppure, ancora, di attivare ed utilizzare il microfono interno per l'ascolto ambientale nel caso in cui la periferica venga chiamata da un numero telefonico conosciuto.

Con la programmazione è ovviamente possibile stabilire la funzione precisa di ciascun tasto e dei vari led, oppure di impostare il modo di funzionamento del ricevitore GPS. Per esempio, al fine di limitare l'assorbimento di corrente, si può programmare l'intervallo di accensione e spegnimento del modulo GPS.

Anche i Timer trovano numerosi impieghi, ad esempio per fare lampeggiare i led oppure, con l'ausilio dei contatori, per particolari funzioni come la scrittura di un dato GPS ad intervalli prestabiliti.

I contatori possono essere sia *ciclici* che *singoli*. Ciclico significa che il

conteggio si ripete periodicamente, mentre singolo significa che la funzione di conteggio viene eseguita una sola volta. Ad esempio:

```
$PFAL,Sys.Timer0.Start=ciclic,2000
```

significa che l'evento Timer "0" dura 2 secondi, allo scadere dei quali si ripete; se avessimo scritto

```
$PFAL,Sys.Timer0.Start=single,2000
```

avremmo avuto lo stesso evento della durata di due secondi ma per una sola volta. Nell'esempio seguente

```
$PFAL,Cnf.Set,ALO=GPS.Nav.eFix=valid:Sys.Timer0.
```

```
Start=ciclic,10000
```

```
$PFAL,Cnf.Set,AL1=GPS.Nav.eFix=invalid:Sys.Timer0.stop
```

```
$PFAL,Cnf.Set,AL2=Timer.e0:GPS.History.Write,0,""
```

con la prima dichiarazione ALO diciamo al Mambo che deve far partire un timer della durata di 10 sec nel caso in cui il segnale GPS sia valido (si ricordi a tal proposito che l'unità di tempo riconosciuta dal localizzatore è il millisecondo, quindi 10.000 msec = 10 sec). Con la seconda riga definiamo che il timer deve fermarsi quando il segnale non è valido. Con la terza ed ultima, quando il Timer ha finito di contare (Timer.e0), ossia ogni 10 secondi, si deve scrivere la posizione GPS nello storico della memoria.

Tutte le programmazioni effettuate possono essere cancellate in qualsiasi momento forzando il Mambo a

caricare tutti i parametri di default con l'istruzione:

```
$PFAL,Sys.Device.FactoryReset
```

Con questo comando in realtà si ottiene il ripristino della configurazione di fabbrica una parte della quale è riportata nel Listato 1.

Fino ad ora abbiamo visto esempi che generano un'azione in funzione di un evento. Personalizzando i vari programmi, però, sicuramente può essere necessario comunicare all'utente alcune informazioni relative al funzionamento, tramite segnalazioni luminose, rese possibili grazie ai tre led di cui Mambo è dotato.

I led di cui il dispositivo è fornito possono assumere tre colori (verde, rosso e blu) e vengono controllati tramite la seguente linea:

```
$PFAL,IO.LED <indice> . <colore>
=< conf. >
```

in cui il parametro *indice* seleziona il led (LED1, LED2 oppure LED0); a titolo informativo, il LED0 è quello vicino ai pulsanti, il LED1 è quel-

lo centrale mentre il LED2 è quello esterno. Il parametro *colore* serve per cambiare, appunto, il colore del led:

sono ammesse le opzioni *Green*, *Red* e *Blue*. Infine, il parametro *conf.* definisce se il led deve rimanere acceso a luce fissa, spento oppure lampeggiare ciclicamente. Per esempio,

```
$PFAL,IO.LED1.Blue=cyclic,500,1000
```

significa che il led centrale deve lampeggiare rimanendo acceso per 500 msec e spento per 1 secondo.

In quest'altro esempio

```
$PFAL,IO.LED1.Green=low
```

si stabilisce che il colore verde del led centrale deve essere spento.

L'uso dei led è importante perché con-

Listato 1

```
$PFAL,CNF.Set,AL0=SYS.Device.eStart=SYS.TIMER_1SEC.start=cyclic,1000&SYS.TIMER_20SEC.start=cyclic,20000&SYS.COUNTER_TIMING_MODE.set=7&GPS.Nav.PowerSave=on,400,1000
All'accensione (Device.eStart) inizializza un Timer ciclico di 1 sec (TIMER_1SEC.start=cyclic,1000) un timer dello stesso tipo della durata di 20 sec (TIMER_20SEC.start=cyclic,20000), un contatore a 7 (COUNTER_TIMING_MODE.set=7) e configura il GPS in modo che stia acceso 400 msec. su 1000 msec. (GPS.Nav.PowerSave=on,400,1000).

$PFAL,CNF.Set,AL1=SYS.Device.eStart=SYS.COUNTER_SLEEP_MODE.set=0&SYS.COUNTER_BAT_MODE.set=0
Sempre all'accensione (Device.eStart) inizializza il contatore (COUNTER_SLEEP_MODE.set=0) a 0 e il contatore (COUNTER_BAT_MODE.set=0) a 0

$PFAL,CNF.Set,AL2=SYS.TIMER.e_1SEC=SYS.COUNTER_TIMING_MODE.decrement=1
Se il (TIMER.e_1SEC) finisce di contare (dopo quindi un secondo) allora (:) decrementa il contatore (COUNTER_TIMING_MODE.decrement=1) di 1

$PFAL,CNF.Set,AL3=SYS.COUNTER.e_TIMING_MODE=SYS.COUNTER_TIMING_MODE.set=7
Se il contatore (COUNTER.e_TIMING_MODE) finisce di contare riportalo a 7 (SYS.COUNTER_TIMING_MODE.set=7).

$PFAL,CNF.Set,AL4=IO.BTN.eX=double=SYS.TIMER.DOUBLE.start=single,100&IO.Beep3=hpulse,200
Se uno qualsiasi dei pulsanti è premuto due volte (entro 500 msec.) (IO.BTN.eX=double) allora (:) inizializza il (TIMER.DOUBLE.start=single,100) ad un valore di 100 msec. ed emetti un beep singolo della durata di 200 msec. (IO.Beep3=hpulse,200).

$PFAL,CNF.Set,AL5=SYS.TIMER.e_DOUBLE=IO.Beep2=hpulse,100
Se il (TIMER.e_DOUBLE) finisce di contare (dopo quindi 100ms) allora emetti un beep singolo della durata di 100 msec. (IO.Beep2=hpulse,100).

$PFAL,CNF.Set,AL6=IO.BTN.eX=long:IO.Beep2=hpulse,600

$PFAL,CNF.Set,AL7=IO.BTN.eX=short:IO.Beep2=hpulse,100

$PFAL,CNF.Set,AL8=IO.BTN.e2=double=SYS.TIMER_SLEEP.start=single,400&SYS.COUNTER_SLEEP_MODE.set=2

$PFAL,CNF.Set,AL9=SYS.COUNTER.e_TIMING_MODE&GPS.Nav.sFix=valid:IO.LED_GPS.green=low&IO.LED_GPS.red=low&IO.LED_GPS.blue=low
Se il contatore (COUNTER.e_TIMING_MODE) finisce di contare (grazie ad AL3 ogni 7 sec) e il segnale GPS è valido (GPS.NAV.sFix=valid) allora spegni completamente il led del GPS (si noti che essendo il led di tre colori la dichiarazione va specificata per ogni colore).

$PFAL,CNF.Set,AL10=SYS.COUNTER.e_TIMING_MODE&GPS.Nav.sFix=invalid:IO.LED_GPS.green=low&IO.LED_GPS.red=hpulse,400&IO.LED_GPS.blue=low
Se il contatore (COUNTER.e_TIMING_MODE) finisce di contare (grazie ad AL3 ogni 7 sec) e il segnale GPS non è valido allora spegni il led verde e blu e fai lampeggiare il rosso del GPS

$PFAL,CNF.Set,AL27=Sys.Device.eShutdown&GPS.Nav.sFix=valid:GPS.Nav.SaveLastValid
Se il Mambo sta per essere spento (sys.Device.eShutdown) e (&) il Gps ha un segnale valido (GPS.Nav.sFix=valid) allora (:) salva in memoria l'ultimo punto valido (GPS.NAV.SaveLastValid).

$PFAL,CNF.Set,AL28=IO.BTN.eX=short&GSM.VoiceCall.sIncoming=GSM.VoiceCall.Accept
Se viene premuto brevemente un qualsiasi pulsante (IO.BTN.eX=Short) ed (&) è in arrivo una chiamata (GSM.VoiceCall.sIncoming) allora (:) accetta la comunicazione (GSM.VoiceCall.Accept).

$PFAL,CNF.Set,AL29=IO.BTN.eX=short&GSM.VoiceCall.sInside=GSM.VoiceCall.Hangup
Se viene premuto brevemente un qualsiasi pulsante (IO.BTN.eX=short) e (&) se si è attualmente impegnati in una conversazione (GSM.VoiceCall.sInside) allora (:) aggancia la comunicazione (GSM.VoiceCall.Hangup).

$PFAL,CNF.Set,AL30=IO.BTN.eX=short&GSM.VoiceCall.sOutgoing=GSM.VoiceCall.Hangup
Se viene premuto brevemente un qualsiasi pulsante (IO.BTN.eX=short) ed (&) stata effettuata una chiamata dal MAMBO (GSM.VoiceCall.sOutgoing) allora (:) chiudi la comunicazione (GSM.VoiceCall.Hangup).

$PFAL,CNF.Set,GSM.Version=55
```

sente di associare segnalazioni visive a precisi stati di funzionamento, ma anche il buzzer può servire allo scopo. Quest'ultimo è gestito con la seguente stringa:

```
$PFAL,IO.Beep <indice> = <conf.>
```

in cui il campo *indice*, in questo caso, può assumere un valore numerico compreso tra 0 e 9 ed ogni valore determina un tipo di suono differente. Il campo *conf.*, analogamente a quanto avviene per i led, definisce se il buzzer deve essere attivo, spento o attivo per un tempo stabilito, anche in modalità ciclica. Un altro esempio consente di meglio comprendere come va pro-

grammato il buzzer:

```
$PFAL,IO.Beep1=cyclic,500,1000
```

in questo modo il suono con tonalità "1" deve essere emesso ciclicamente per 500 msec, seguito da una pausa di un secondo.

Conclusioni

Da quanto visto fino ad ora, avrete certamente intuito che Mambo è un dispositivo molto versatile e completamente adattabile alle proprie esigenze. All'acquisto, esso è predisposto per garantire una serie di funzionalità base, descritte dal file di configurazione che, per motivi di

spazio, non è stato possibile inserire e commentare in toto: sappiate che il listato di default comprende, oltre alla Configurazione Utente, anche la Configurazione di Sistema e pertanto è molto lungo e complesso. Si è comunque cercato di spiegare in linea di massima, i comandi della User Configuration più frequentemente utilizzati per compiere azioni. Ricordiamo che Mambo dispone di un completo manuale comprendente anche numerosi esempi di base che andremo ad utilizzare nelle prossime puntate per presentare alcuni esempi pratici piuttosto complessi. Non perdetevi quindi le prossime uscite!



Impariamo a conoscere e utilizzare uno dei componenti elettronici più interessanti: il display LCD. Scopriamo tutti i segreti del controller grafico Samsung KS0713, un chip potente e completo, in grado di gestire sia le comunicazioni con il microcontrollore che la visualizzazione grafica.

A CURA DI *Matteo Destro*

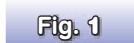
6

Proseguiamo il viaggio nell'affascinante mondo dei display grafici. In questa puntata parliamo del display 64128EFCBC-3LP e del suo controller grafico KS0713 prodotto da Samsung: questo controllore è largamente utilizzato sia dai produttori di display, sia dalle aziende che realizzano prodotti specializzati (di qualsiasi tipo) dotati di un display grafico. Ciò è possibile per via di un grande vantaggio offerto dal KS0713: questo chip è infatti in grado di occuparsi contemporaneamente tanto della comunicazione con il microcontrollore quanto della gestione della parte grafica, contrariamente a quanto visto per il KS0108B che, per poter realizzare la funzione di display grafico completo, necessita di un "companion chip" chiamato KS0107B. Vediamo insieme come funziona il 713, quali comandi mette a disposizione e, infine, come si

utilizza: per quest'ultimo punto ci aiuteremo con delle routine d'esempio implementate nella demoboard.

L'architettura del controller grafico KS0713

Il KS0713 è un driver per display LCD realizzato in tecnologia LSI (Large Scale Integration = integrazione a larga scala), tecnologia che ha permesso di integrare in un unico dispositivo sia l'elettronica di gestione dei segnali necessari a pilotare il vetrino LCD, sia la logica di controllo e interfacciamento con un microcontrollore; contiene, inoltre, la memoria RAM necessaria per immagazzinare le informazioni (immagini o altro) da visualizzare sul display. La dimensione della RAM interna è di 1072 byte, taglio che consente di gestire un display con risoluzione massi- ➤



Il blocco successivo è il “Display Data RAM o DDRAM”, ossia il banco di memorie che contiene l’immagine da visualizzare sull’LCD. Tale memoria RAM è un “array” bidimensionale composto da 65 righe per 132 colonne, per un totale di 8580 celle da un bit per pixel: per accendere un singolo pixel è necessario individuare la riga e la colonna appropriati e settare il bit relativo. Le 65 righe dell’array sono suddivise in 8 pagine da 8 linee ciascuna, più una pagina composta da una sola riga; ogni pagina, infine, è formata da 64 co-

Un altro circuito interno molto importante è lo “Status Register”, che memorizza e comunica le informazioni di stato del display tramite il bit “Busy

lonne (64 byte). Per gestire la memoria DDRAM sono necessari i tre blocchi funzionali denominati “Page Address Circuit”, “Column Address Circuit” e “Line Address Circuit”. Il primo gestisce ed indirizza le nove pagine della memoria DDRAM. Per un corretto indirizzamento è necessario impostare 4 bit di selezione pagina, utilizzando un apposito comando. La Figura 2 mostra come sono distribuite le nove pagine della memoria DDRAM e le relative colonne di indirizzamento (SEGx). Per scrivere correttamente sulla memoria del controller è necessario prima impostare l'indirizzo di pagina su cui si vuole scrivere e successivamente l'indirizzo di colonna. Entrambe le operazioni necessitano di appositi comandi, spiegati un po' più avanti.

Il blocco “Column Address Circuit” serve per gestire l'indirizzo di colonna nella DDRAM. Le colonne sono 131 e quindi, per poter indirizzare tutte le celle della memoria, servono 8 bit. L'indirizzamento di colonna è dotato di autoincremento dopo ogni operazione di lettura/scrittura.

Il terzo e ultimo registro di gestione della DDRAM, il “Line Address Circuit”, serve per spostare l'immagine visualizzata sul display: a seconda del suo valore l'immagine è più o meno spostata rispetto all'asse Y. In altre parole serve per decidere da quale riga della memoria DDRAM si inizia la visualizzazione sul display LCD.

Il circuito successivo è il “Segment Controller”, che viene pilotato da tre istruzioni (facilmente intuibili) ininfluenti sul contenuto della memoria RAM, ma che hanno effetto sulla visualizzazione dell'immagine sul cristallo liquido. Le istruzioni sono “Display ON/OFF”, “Reverse Display ON/OFF” e “Entire Display ON/OFF”.

Il blocco “Common Output Control Circuit” è

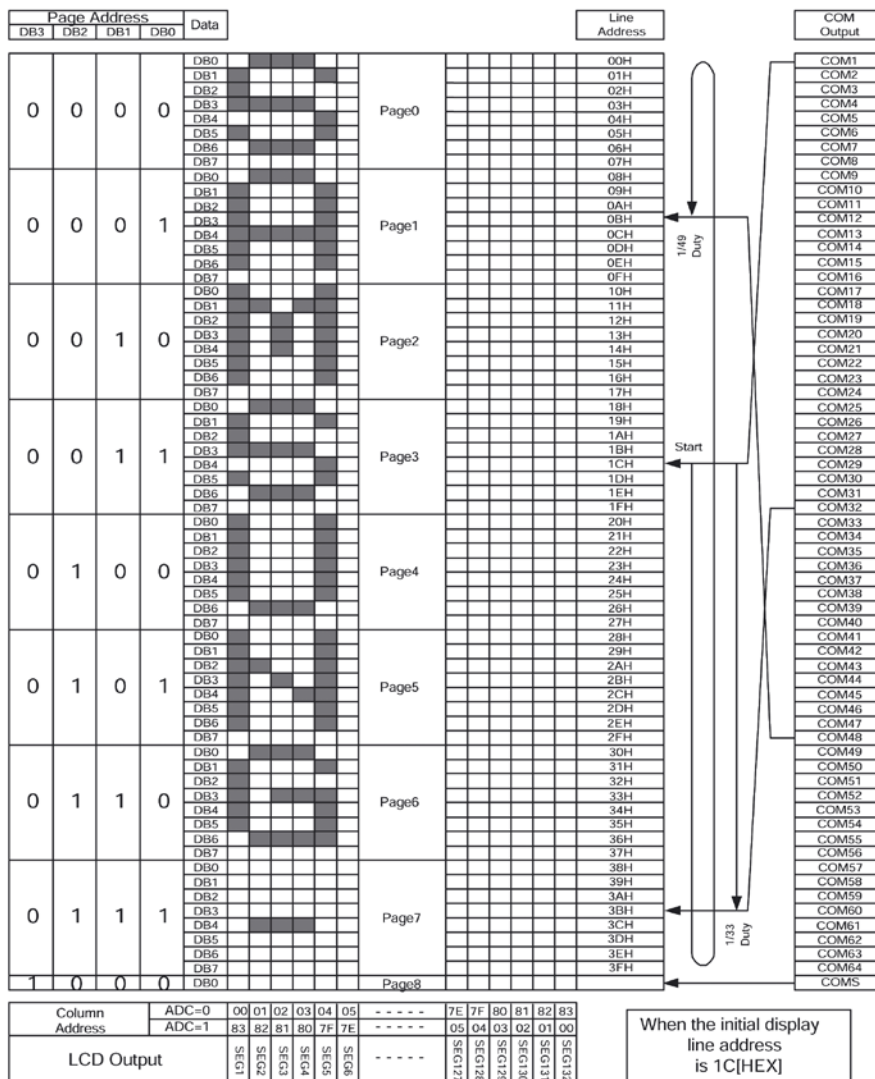


Fig. 2

piuttosto complesso perché è composto, in pratica, da tre blocchi, ossia il “Common Controller” e due “33 Common Driver Circuit”. Spiegare come funziona esattamente sarebbe piuttosto complesso e pesante, perciò vogliamo semplificare al massimo il concetto: i pixel del display sono collegati a matrice e ciò significa che per accendere un pixel in particolare occorre fornire sulla “linea” e sulla “colonna” (coordinate x-y) di tale pixel due livelli di tensione opposti e variabili alternativamente (come necessario agli LCD) ad una certa frequenza. I pixel, inoltre, vengono indirizzati una riga (132 pixel) alla volta. Da ciò si deduce che le righe che compongono l'immagine vengono disegnate in 65 fasi successive ad ognuna delle quali viene assegnato 1/65 del tempo necessario per costruire tutta l'immagine. Si può pertanto affermare che il duty cycle assegnato ad ogni riga è appunto 1/65. Bene, il “Common Output Control Circuit” controlla la relazione che intercorre

tra il duty cycle impostato (in questo caso è 1/65, dato che il nostro display ha 64 righe) e il numero di “uscite comuni” (leggasi “righe”) usate per costruire l’immagine, tenendo conto anche delle esigenze di polarizzazione di ogni singolo pixel. Questo blocco funzionale viene condizionato dall’istruzione “SHL Select”.

Consideriamo infine il circuito di controllo dell’alimentazione il quale comprende tre blocchi funzionali denominati “V/C circuit”, “V/R Circuit” e “V/F Circuit”.

Il “V/C circuit” necessita di una tensione di ingresso che viene riportata in uscita moltiplicata per un fattore K. A seconda di come viene configurato il chip, il fattore K può assumere i valori 2, 3, 4 o 5.

Il “V/R Circuit” gestisce e determina la tensione di controllo (chiamata “Vo”) del cristallo liquido. Tale tensione, usata per regolare il contrasto, viene calcolata con la seguente funzione:

$$V_o = (1 + R_b/R_a) * [1 - (63 - \alpha)] * V_{ref}$$

Il valore di riferimento V_{ref} è fisso a 2,0 Vdc mentre la costante α viene impostata al valore desiderato utilizzando le due istruzioni “Set Reference Voltage Mode” e “Set Reference Voltage Register”, spiegate tra qualche pagina. Il fattore di moltiplicazione viene ricavato utilizzando l’istruzione “Regulator Resistor Select” la quale agisce sulla sezione di formula “ $1 + R_b/R_a$ ” (R_b ed R_a sono due resistenze di riferimento interne al chip) e permette di selezionare fino a 8 differenti valori.

Utilizzando correttamente le istruzioni sopra esposte si ottiene il giusto valore di contrasto da utilizzare con il display LCD.

Per ultimo abbiamo il “V/F Circuit” che suddivide la tensione “Vo” in quattro livelli di tensione distinti V1, V2, V3 e V4. Questi valori dipendono dall’impostazione di due pin del controller denominati “DUTY1” e “DUTY0” i quali, nei display commerciali, non sono accessibili dall’esterno e vengono quindi configurati in fabbrica. L’unica variabile del V/F Circuit su cui si può agire è il flag di “Bias” il quale viene impostato da un apposito comando denominato “LCD Bias Select”.

Tabella 1

Duty Ratio	Duty 1	Duty 0	LCD Bias	V1	V2	V3	V4
1/65	fisso	fisso	1/7 (0)	(6/7)*Vo	(5/7)*Vo	(2/7)*Vo	(1/7)*Vo
			1/9 (1)	(8/9)*Vo	(7/9)*Vo	(2/9)*Vo	(1/9)*Vo

Con le impostazioni fisse dei pin “DUTY1” e “DUTY0” si possono solo ottenere (nel nostro caso) le due configurazioni visibili in tabella 1. Le caselle evidenziate in giallo mostrano la configurazione utilizzata durante l’inizializzazione del display grafico. A questo punto, possiamo ritenere conclusa la spiegazione dello schema di principio del controller KS0713, quindi iniziamo a vederne i comandi.

Il set di comandi del KS0713

Abbiamo analizzato dettagliatamente la struttura hardware del controllore grafico, per cui ora dobbiamo prendere confidenza con il set di comandi necessari per interagire con un’architettura così complessa. La tabella 4 riassume tutti i comandi supportati dal KS0713. Poiché questo controllore dispone di molti comandi, taluni piuttosto complessi, ancora una volta abbiamo semplificato al massimo la spiegazione.

Read Display Data

Tabella 2 - Read Display Data

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	Data							

Questo comando (Tabella 2) serve per leggere un byte dalla memoria DDRAM del controller dopo avere inizializzato l’indirizzo di pagina e l’indirizzo di colonna. L’indirizzo di colonna viene incrementato automaticamente dopo la lettura: in questo modo è possibile eseguire una lettura sequenziale dei dati. Prima di leggere il dato richiesto è necessario eseguire una lettura “Dummy” (“fittizia”), cioè una lettura a vuoto.

Write Display Data

Tabella 3 - Write Display Data

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	Data							

Come suggerisce il nome, serve per scrivere un byte nella memoria DDRAM. Prima di eseguire l’operazione di scrittura è necessario inizializzare l’indirizzo di pagina e l’indirizzo di colonna identificando così la locazione di memoria richiesta (Tabella 3).

Analogamente alla lettura, l’indirizzo di colonna viene incrementato automaticamente ad ogni scrittura in RAM, permettendo di eseguire una scrittura sequenziale dei dati.

Tabella 4 - Riassunto dei comandi

Comando	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Funzione
Read Display Data	1	1	Dato letto dalla DDRAM								Leggi dato dalla DDRAM
Write Display Data	1	0	Data scritto in DDRAM								Scrivi dato nella DDRAM
Read Status	0	1	Busy	ADC	ON/OFF	RST	0	0	0	0	Leggi lo stato del controller
Display ON/OFF	0	0	1	0	1	0	1	1	1	DON	Accende il display LCD Se DON = 0; Display OFF Se DON = 1; Display ON
Initial Display Line	0	0	0	1	ST5	ST4	ST3	ST2	ST1	ST0	Specifica la riga di memoria DDRAM per COM1
Set Reference Voltage Mode	0	0	1	0	0	0	0	0	0	1	Comando per impostazione tensione di riferimento. Necessita di parametro
Set Reference Voltage Register	0	0	x	x	SV5	SV4	SV3	SV2	SV1	SV0	Parametro necessario durante impostazione tensione di riferimento
Set Page Address	0	0	1	0	1	1	P3	P2	P1	P0	Impostazione indirizzo di pagina della memoria DDRAM su cui leggere o scrivere
Set Colum Address MSB	0	0	0	0	0	1	Y7	Y6	Y5	Y4	Impostazione indirizzo di colonna, nibble più significativo (MSB)
Set Colum Address LSB	0	0	0	0	0	0	Y3	Y2	Y1	Y0	Impostazione indirizzo di colonna, nibble meno significativo (LSB)
ADC Select	0	0	1	0	1	0	0	0	0	ADC	Se ADC = 0; impostata direzione normale di visualizzazione rispetto all'asse X cioè da SEG1 → SEG132 Se ADC = 1; immagine specchiata rispetto all'asse Y. Visualizzazione invertita rispetto all'asse X cioè da SEG132 → SEG1
Reverse Display ON/OFF	0	0	1	0	1	0	0	0	0	ADC	Se REV = 0; visualizzazione normale Se REV = 1; visualizzazione in modalità reverse
Entire Display ON/OFF	0	0	1	0	1	0	0	1	0	EON	Se EON = 0; visualizzazione normale Se EON = 1; visualizza tutti i pixel del display a "1" logico
LCD Bias Select	0	0	1	0	1	0	0	1	0	BIAS	Se BIAS = 0; selezionato fattore di moltiplicazione 1/7 Se BIAS = 1; selezionato fattore di moltiplicazione a 1/9
Set-modify-read	0	0	1	1	1	0	0	0	0	0	Imposta modalità modify-read
Reset-modify-read	0	0	1	1	1	0	1	1	1	0	Annulla modalità modify-read
Reset	0	0	1	1	1	0	0	0	1	0	Comando di inizializzazione funzioni interne
SHL Select	0	0	1	1	0	0	SHL	X	X	X	Se SHL = 0; impostata direzione normale di visualizzazione rispetto all'asse Y cioè da COM1 → COM132 Se SHL = 1; immagine specchiata rispetto all'asse X. Visualizzazione invertita rispetto all'asse Y cioè da COM64 → COM1
Power control	0	0	0	0	1	0	1	VC	VR	VF	Attivazione blocchi di controllo alimentazione. Attivi se a livello logico alto.
Regulator Resistor select	0	0	0	0	1	0	0	R2	R1	R0	Impostazione rapporto di resistenza per regolazione contrasto LCD
Set Static Indicator Mode	0	0	1	0	1	0	1	1	0	SM	Comando per impostazione indicatore statico
Set Static Indicator Register	0	0	X	X	X	X	X	X	S1	S0	Parametro necessario durante impostazione indicatore statico
Power Safe	-	-	-	-	-	-	-	-	-	-	X
Test Instruction	0	0	1	1	1	1	X	X	X	X	Non usare questa istruzione

Read Status

Serve per leggere lo stato del controller. Degli 8 bit disponibili solo quattro contengono delle informazioni, in particolare i quattro bit più significativi (tabella 5). Per la lettura dello stato non è

Tabella 5 - Read Status

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	Busy	ADC	ON/OFF	RST	0	0	0	1

necessario eseguire una lettura Dummy. Vediamo ora il significato dei quattro flag.

BUSY: Indica se il controller è occupato e quindi non disponibile a ricevere comandi e dati. "0" significa "controller disponibile"; "1", viceversa, "controller non disponibile".

ADC: Indica la relazione tra l'indirizzo di colonna della memoria DDRAM e il driver per la gestione dei segmenti. "0" definisce la direzione ➤

invertita “SEG132 → SEG1”; “1”, invece, determina la direzione normale SEG1 → SEG132
 ON/OFF: Indica se il display è acceso o spento. “0” = display acceso; “1” = display spento
 RST: Indica se è in corso l’inizializzazione interna del controller. “0” = nessun reset in corso; “1” = reset in corso.

Display ON/OFF

Tabella 6 - Display ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	0	1	1	1	DON

Questo comando (Tabella 6) accende e spegne il display: se “DON” viene posto a livello logico alto il display è acceso; viceversa è spento.

Initial Display Line

Tabella 7 - Initial Display Line

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	0	ST3	ST2	ST1	ST0

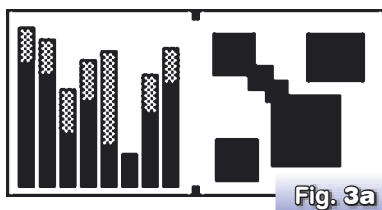


Fig. 3a



Fig. 3b

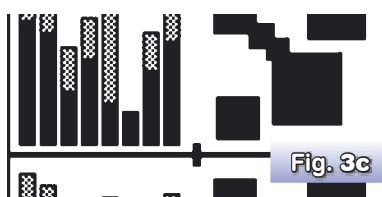


Fig. 3c

Questo registro, dettagliato in Tabella 7, serve per indicare da quale linea si inizia a visualizzare l'immagine contenuta nella memoria DDRAM. Può essere utilizzato per eseguire uno scroll del display rispetto all'asse Y. L'esempio a sinistra evidenzia in modo più chiaro quanto detto: L'immagine 3a ha il registro “Initial Display Line” impostato a 0x00 quindi essa

non appare “spostata” rispetto all'asse Y. L'immagine 3b ha subito uno scroll, in particolare il registro “Initial Display Line” è impostato a 0x0A. Come si può osservare, la parte superiore dell'immagine ricompare nella parte inferiore del display. Accentuando ancora il valore del registro “Initial Display Line” si nota ancora di più lo spostamento dell'immagine rispetto alla sua posizione originale. Nella figura 3c il valore del registro “Initial Display Line” è pari a 0x0F.

Set Reference Voltage Mode e Set Reference Voltage Register

Per impostare la tensione di riferimento è necessario inviare prima un comando (Tabella 8a) e poi il suo parametro di configurazione (Tabella 8b).

Tabella 8a - Set Reference Voltage Mode

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	0	0	0	0	1

Tabella 8b - Set Reference Voltage Register

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	X	X	SV5	SV4	SV3	SV2	SV1	SV0

Questa coppia di istruzioni imposta il blocco funzionale “V/R Circuit” descritto in precedenza. Il parametro agisce sulla variabile α dell'equazione

$$V_o = (1 + R_b / R_a) * [1 - (63 - \alpha)] * V_{ref}$$

variando il valore di α si varia il contrasto. Tale equazione viene influenzata anche da un altro comando, il “Regulator Resistor Select” descritto nel paragrafo successivo, che agisce sul rapporto di resistenza

$$1 + R_b / R_a.$$

Regulator Resistor Select

Questo comando (Tabella 9), in abbinamento al comando precedente, serve per completare la configurazione del blocco funzionale “V/R Circuit”.

Tabella 9 - Regular Resistor Select

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	0	0	R2	R1	R0

In particolare agisce sul rapporto delle resistenze i cui possibili valori sono riportati in Tabella 10. Grazie a questo ed al precedente comando, il contrasto può essere regolato via software anziché usando dei trimmer esterni.

Tabella 10

R2	R1	R0	1+(Rb/Ra)	R2	R1	R0	1+(Rb/Ra)
0	0	0	1,90	1	0	0	3,61
0	0	1	2,19	1	0	1	4,35
0	1	0	2,55	1	1	0	5,29
0	1	1	3,02	1	1	1	6,48

Set Page Address

Esattamente come il nome lascia intuire, esso serve per selezionare la pagina di memoria DDRAM su cui si vuole scrivere. Oltre all'indirizzo di pagina, è necessario impostare anche l'indirizzo di colonna. In totale, le pagine sono 9, indirizzate da "0" a "8" (vedere Tabella 11, bit P3÷P0).

Tabella 11 - Set Page Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	1	1	P3	P2	P1	P0

Set Colum Address MSB e LSB

Tramite questo comando si imposta l'indirizzo di colonna e, poiché le colonne sono 132, esso necessita di 8 bit per indirizzarle tutte. Tale registro deve essere programmato inviando al controller due comandi distinti (Tabella 12), contenenti

Tabella 12 - Set Colum Address MSB e LSB

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	Y7	Y6	Y5	Y4
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	Y3	Y2	Y1	Y0

ognuno ½ byte, ossia 1 nibble. Prima si invia il nibble più significativo (Y7÷Y4) poi il meno significativo (Y3÷Y0).

ADC Select

Serve per impostare la relazione tra l'indirizzo di colonna della memoria DDRAM e il driver di segmento del controller.

Tramite esso, è possibile specchiare l'immagine

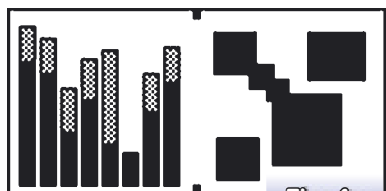


Fig. 4a



Fig. 4b

Tabella 13 - ADC Select

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	0	0	0	0	ADC

(specchiato) rispetto all'asse Y e quindi ADC = 1. Ricordiamo che il contenuto della memoria non è stato alterato in alcun modo in quanto questo comando agisce solo sulla visualizzazione della memoria e non sul suo contenuto.

Reverse Display ON/OFF

Tabella 14 - Reverse Display ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	0	0	1	1	REV

Visibile in tabella 14, serve per invertire il valore logico dei pixel contenuti nella memoria DDRAM del controller senza corromperne il contenuto (immagine negativa). Se REV = 0 non viene effettuata alcuna inversione (negazione) del valore logico dei pixel cosa che avviene con REV = 1.

Entire Display ON/OFF

Tabella 15 - Entire Display ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	0	0	1	0	EON

Forza tutti i pixel del display LCD a "1" logico senza tenere conto del contenuto della memoria DDRAM. Questo comando (tabella 15) ha priorità sul comando "Reverse Display ON/OFF".

Select LCD Bias

Tabella 16 - Select LCD Bias

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	0	0	0	1	Bias

Agisce sul blocco funzionale "V/F Circuit". Nel nostro caso il valore di Bias è impostato a "1", quindi il fattore di divisione per Vo è 1/9.

Set Modify-Read e Reset Modify-Read

Il primo (tabella 17a) blocca l'auto-incremento dell'indirizzo di colonna solo durante la lettura

Tabella 17a - Set Modify-Read

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	1	0	1	1	1	0

ra della memoria DDRAM. Durante la scrittura l'auto-incremento rimane ancora attivo.

Per annullare questa impostazione è necessario utilizzare l'istruzione "Reset Modify-Read" descritta in Tabella 17b.



Tabella 17b - Reset Modify-Read

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	1	0	0	0	1	0

Il comando Reset Modify-Read serve per azzerare alcuni parametri del controller precedentemente impostati (Reset Software).

I parametri controllati dal comando sono:

- Display Line
- Column Address
- Page Address
- Status

Tale comando non influenza il contenuto della memoria DDRAM.

SHL Select

Tabella 18 - SHL Select

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	0	0	SHL	X	X	X

Permette di impostare la relazione tra le righe dell'immagine memorizzata in DDRAM e le

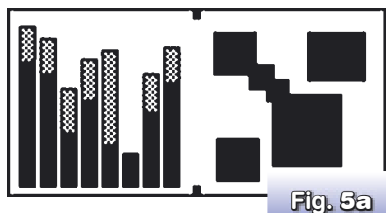


Fig. 5a

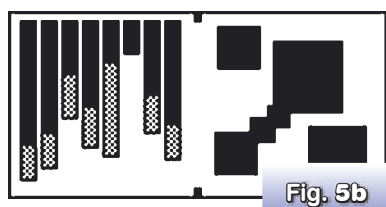


Fig. 5b

linee COMx del controller. Ad esempio, utilizzando correttamente il comando (visibile in tabella 18) è possibile ribaltare l'immagine, contenuta in memoria, rispetto all'asse X senza modificarla. La figura 5a visualizza il contenuto della memoria DDRAM in modo normale

(SHL = 0) mentre la figura 5b visualizza il contenuto della memoria ribaltato rispetto all'asse X (SHL = 1). Come per il comando ADC Select, il contenuto della memoria RAM non è stato alterato in alcun modo.

Power Control

Questo comando (Tab. 19) attiva i tre blocchi funzionali di controllo dell'alimentazione del display LCD. Tali blocchi, descritti precedentemente, sono

Tabella 19 - Power Control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	0	1	VC	VR	VF

“V/C Circuit”, “V/R Circuit” e “V/F Circuit”.

Quando i bit VC, VR e VF sono a livello logico alto significa che il relativo blocco è attivo, in caso contrario esso è spento.

Set Static Indicator Mode

Tabella 20 - Set Static Indicator Mode

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	0	1	1	0	SM

Serve per attivare/disattivare il cursore (Set Static Indicator). Se SM = 0 il cursore è spento, viceversa è acceso. Questo comando (visibile in tabella 20) coopera col comando di controllo dello “Static Indicator Register”, spiegato di seguito.

Set Static Indicator Register

Tabella 21 - Set Static Indicator Register

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	X	X	X	X	X	X	S1	S0

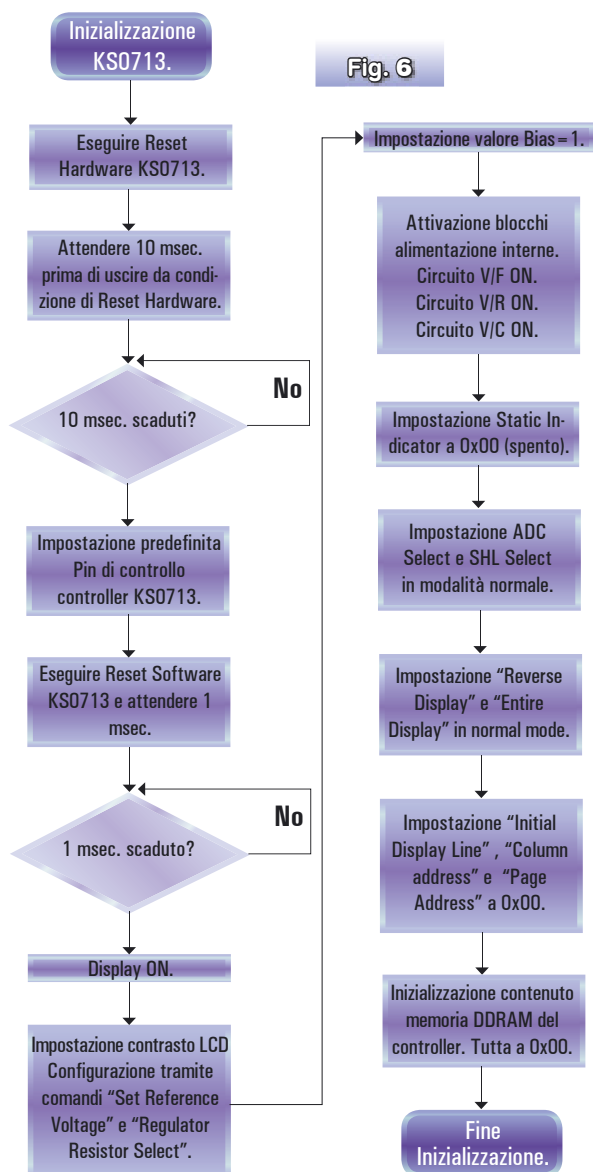
Permette di impostare “Static Indicator Register”. Nella nostra applicazione, SM corrisponde a 0 e quindi il valore da passare al registro “Static Indicator Register” non ha significato.

Il firmware per gestire il KS0713

Giunti a questo punto, abbiamo visto sia la struttura hardware che il set di comandi del controller in oggetto. Siete quindi pronti per analizzare il firmware di inizializzazione che, come da diagramma di flusso in figura 6, necessita di diversi passi.

Per prima cosa è necessario eseguire un reset hardware, agendo sull'apposito pin di controllo denominato “RST” (vedere lo schema elettrico sul fascicolo 116): tale pin va mantenuto a livello logico basso per almeno 10 millisecondi. Conclusa la fase di reset hardware è necessario impostare i pin di controllo del KS0713: in particolare occorre settare il “CS1” che serve per abilitare il controller. Fatto ciò si esegue un reset software (comando “Reset”) in modo da impostare alcuni blocchi funzionali a dei valori predefiniti, necessari per il corretto funzionamento.

Conclusa questa prima fase di init, è possibile accendere il display e configurare il contrasto ad un valore ottimale per la visualizzazione delle immagini. Si procede impostando il valore di “Bias” (comando Select LCD Bias) per passare



successivamente all'accensione dei tre blocchi di gestione dell'alimentazione interna (i circuiti "V/F Circuit", "V/R Circuit" e "V/C Circuit") tramite il comando Power Control, in cui i bit DB2-DB1-DB0 sono tutti a 1. Ora si imposta il blocco "Static Indicator" a 0x00 e successivamente si programmano i registri "ADC Select" e "SHL Select" in modalità normale. Seguono le impostazioni di "Reverse Display" e "Entire Display" anch'essi in modalità normale.

Non rimane che definire l'indirizzo di pagina, l'indirizzo di colonna e la linea di inizio visualizzazione. Se questi tre parametri vengono impostati tutti a zero si può fare a meno di iniziarli perché il comando di "Reset" dato in precedenza ha già riportato tali registri al valore di 0x00. Per ultimo si esegue l'inizializzazione della memoria DDRAM perché, al termine del reset e del power up, non si conosce lo stato che le celle possono assumere, quindi conviene scrivere un valore

Listato 1

```

void Init_KS0713(void)
{
  RST_LCD_M(LOW);
  Wait_mSec(10);
  RST_LCD_M(HIGH);

  Set_Pin_KS0713();
  Reset_SW_KS0713();
  Wait_mSec(1);

  Set_ON_KS0713();
  Set_RES_KS0713(0x06);
  Set_REF_Voltage_KS0713();
  Set_REF_Voltage_Value_KS0713(0x07);
  Set_BIAS_KS0713();
  Set_POWERC_KS0713();
  Set_SIR_KS0713();
  Set_ADC_KS0713(KS0713_ADC_SELECT_0);
  Set_SHL_KS0713(KS0713_SHL_SELECT_0);
  Set_OFF_All_LCD_Pixel();
  Set_REV_Disp_OFF();
  Set_Start_Line_KS0713();
  Set_Address_KS0713_MSB();
  Set_Address_KS0713_LSB();
  Set_Page_KS0713();
  Clear_RAM_KS0713();
}
  
```

- Reset Hardware.
- Impostazione pin di controllo.
- Display LCD ON.
- Impostazione contrasto.
- Impostazione BIAS.
- Impostazione alimentazione interna.
- Impostazione Static Indicator.
- Impostazione ADC e SHL.
- Impostazione "Reverse Display" e "Entire Display" in mod. normale.
- Impostazione indirizzo di pagina e indirizzo di colonna. Inizio visualizzazione da linea 0x00.
- Impostazione DDRAM.

noto, ad esempio 0x00, in ognuna di esse. In questo modo si evita che compaiano dei pixel accesi indesiderati sullo schermo.

Il codice della routine di inizializzazione, visibile nel listato 1, segue esattamente la flow chart di figura 6. Nonostante il diagramma di flusso sia piuttosto complesso, la routine di inizializzazione dei pin di controllo dell'integrato KS0713 è molto semplice: dal Listato 2, infatti, si evincono sei

Listato 2

```

void Set_Pin_KS0713(void)
{
  MI_LCD_M(HIGH); // 6800 Microcontroller
  CSI_LCD_M(LOW); // Abilita KS0713
  RS_LCD_M(LOW); // Selezionato invio comando
  RW_LCD_M(LOW); // Abilitata scrittura
  E_LCD_M(LOW);
  BCKL_2_M(HIGH); // Accende retroilluminazione
}
  
```

linee di comando; prima viene definita l'interfaccia di comunicazione parallela di tipo 6800, poi si abilita il controller KS0713 e infine i pin "RS", "R/W" ed "E". Per una migliore visualizzazione accendiamo la retroilluminazione del display.

Dopo aver configurato correttamente l'interfaccia hardware, è necessario inviare dei comandi specifici al driver: per facilitare l'invio sia dei comandi che dei dati, sono state realizzate due routine apposite. La prima, chiamata "Write_Comm_KS0713" (Listato 3), come intuibile dal nome, serve per inviare i comandi e quindi controlla i pin "R/W", "RS" e "E" secondo il diagramma temporale visibile in figura 7. Analogamente, la

Listato 3

```
void Write_Comm_KS0713(uchar Command)
{
    RW_LCD_M(LOW);
    Nop();
    RS_LCD_M(LOW);
    Nop();
    E_LCD_M(HIGH);
    LATD = Command;
    Nop();
    E_LCD_M(LOW);
}
```

Listato 4

```
void Write_Data_KS0713(void)
{
    RW_LCD_M(LOW);
    Nop();
    RS_LCD_M(HIGH);
    Nop();
    E_LCD_M(HIGH);
    Nop();
    LATD = KS0713_Data;
    Nop();
    E_LCD_M(LOW);
}
```

routine “Write_Data_KS0713” (Listato 4) invia i dati e i parametri al controller. Osservando le routine di scrittura si nota che sono state inserite delle istruzioni di Nop() che servono per introdurre dei ritardi voluti, tramite i quali si garantisce il rispetto dei diagrammi temporali.

Nella tabella 22 sono riportati i tempi da rispettare quando il controller viene alimentato a +3,3 Vdc.

Come già anticipato, il diagramma temporale di figura 7 si riferisce sia alle operazioni di lettura che a quelle di scrittura, attraverso un'interfaccia parallela di tipo 6800. Anche se è abbastanza scontato, ricordiamo sempre quanto sia importante il rispetto delle tempistiche.

Analizziamo ora la routine di lettura dello stato del controller.

Come si può notare dal listato 5, è necessario impostare la Porta D del PIC18F4620 in lettura forzando il valore 0xFF nel registro TRISD.

Dopodiché occorre gestire opportunamente i pin di controllo, rendendo così disponibile lo stato del controller sul bus dati.

Tale valore viene letto e trasferito nella variabile “KS0713_Status”. Conclusa la lettura si esegue un test mascherato verificando che il flag di “Busy” non sia a livello logico alto. Nel caso in cui tale flag sia a 1 logico, il programma rimane in loop in attesa che il controller esca dalla condizione di “occupato”. Prima di uscire dalla routine è necessario reimpostare la Porta D come uscita,

Tabella 22 - Tempistiche assolute

Item	Signal	Symbol	Min	Typ	Max	Unit	Remark
Address setup time	RS	t_{AS68}	10	—	—	ns	
Address hold time		t_{AH68}	10	—	—	ns	
System cycle time	RS	t_{CY68}	150	—	—	ns	
Data setup time	DB0-DB7	t_{DS68}	18	—	—	ns	
Data hold time		t_{DH68}	10	—	—	ns	
Access time	DB0-DB7	t_{ACC68}	—	—	65	ns	$C_L = 100pF$
Output disable time		t_{OD68}	10	—	45	ns	
Enable pulse width	Read	E_RD	$t_{PW68(R)}$	65	—	—	
	Write		$t_{PW68(W)}$	25	—	—	

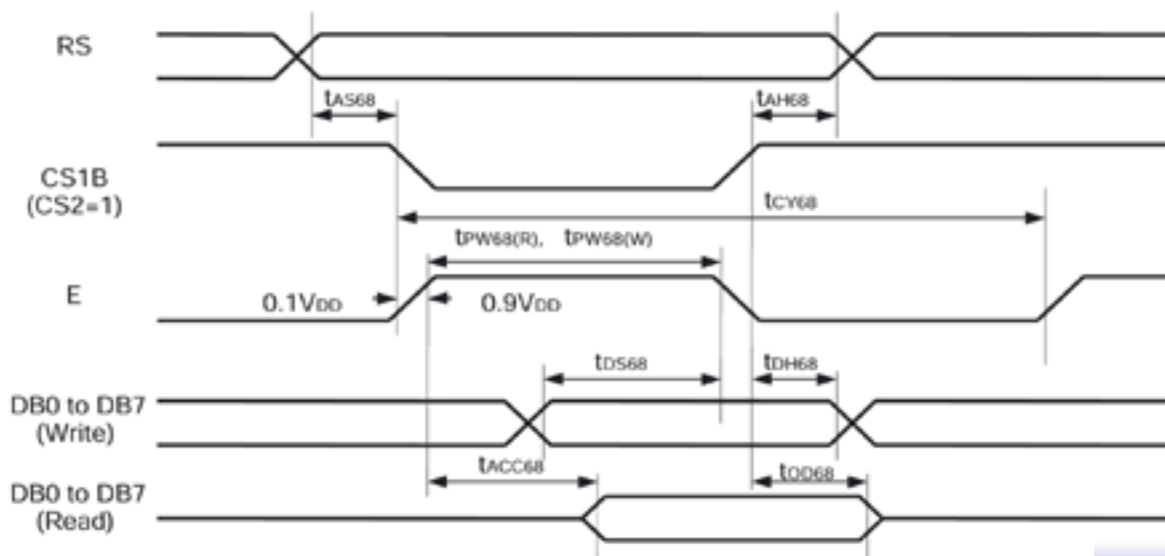


Fig. 7

Listato 5

```

void Test_KS0713_Busy(void)
{
    TRISD = 0b11111111;
    Wait_Busy:
    RW_LCD_M(HIGH);
    Nop();
    RS_LCD_M(LOW);
    Nop();
    E_LCD_M(HIGH);
    Nop();
    Nop();
    E_LCD_M(LOW);

    KS0713_Status = PORTD & 0x80;
    if (KS0713_Status != 0x00)
    {
        goto Wait_Busy;
    }
    TRISD = 0b00000000;
}

```

Porta D impostata come ingresso.

Gestione segnali di controllo per lettura stato controller.

Mascheratura per isolare il FLAG di Busy.

Impostazione indirizzo di pagina e indirizzo di colonna. Inizio visualizzazione da linea 0x00.

Porta D impostata come uscita.

forzando il valore 0x00 nel registro TRISD.

Fino ad ora abbiamo eseguito solamente delle configurazioni hardware: l'unica operazione effettiva fatta sul display è il controllo dello stato di busy. Ora iniziamo a far comunicare il micro con il display. Ricorderete che i comandi supportati dal KS0713 (largamente utilizzati da questo punto in poi) sono parecchi, quindi, per semplificare l'utilizzo del codice, abbiamo realizzato una routine specifica per ogni singolo comando: prestate attenzione al fatto che alcune di esse

Listato 6

```

void Clear_RAM_KS0713(void)
{
    uchar temp_count;
    KS0713_Page = KS0713_Page_0;
    KS0713_Y_Address_MSB = 0;
    KS0713_Y_Address_LSB = 0;
    KS0713_Data = 0x00;
    do
    {
        Set_Address_KS0713_MSB();
        Set_Address_KS0713_LSB();
        Set_Page_KS0713();
        temp_count = 128;
        do
        {
            Write_Data_KS0713();
        } while (--temp_count != 0);
    } while(++KS0713_Page != 8);
}

```

Inizializzazione delle variabili necessarie per l'indirizzamento della memoria DDRAM prima della sua cancellazione.

Valore da scrivere in RAM.

Indirizzamento memoria DDRAM del controller. L'indirizzamento dipende dai valori assunti dalle variabili descritte prima del ciclo "do While".

Scrittura di 128 byte nella pagina di memoria "n". Conclusa la scrittura sui 128 byte della pagina "n" si passa all'iniziale della pagina "n+1".

Il loop "Do While" si conclude dopo avere inizializzato.

Listato 7

```

void Write_Bitmap_RAM_KS0713(uint Address_EEPROM, uchar Offset)
{
    uchar temp_count;
    KS0713_Page = KS0713_Page_0;
    KS0713_Y_Address_MSB = 0;
    KS0713_Y_Address_LSB = Offset;
    I2C_EEPROM_Add.u1 = Address_EEPROM;
    do
    {
        Set_Address_KS0713_MSB();
        Set_Address_KS0713_LSB();
        Set_Page_KS0713();
        temp_count = 128;
        do
        {
            I2C_Process();
            KS0713_Data = I2C_EEPROM_Data[0];
            I2C_EEPROM_Add.u1 += 1;
            Write_Data_KS0713();
        } while (--temp_count != 0);
    } while(++KS0713_Page != 8);
}

```

Inizializzazione delle variabili necessarie per l'indirizzamento della memoria DDRAM.

Indirizzamento EEPROM per lettura immagine.

Indirizzamento memoria DDRAM del controller. L'indirizzamento dipende dai valori assunti dalle variabili descritte prima del ciclo "do While".

Routine per la lettura dei dati dalla EEPROM.

Valore da scrivere in DDRAM e letto dalla EEPROM.

Routine per la scrittura nella memoria DDRAM.

necessitano di parametri aggiuntivi. Tali routine sono visibili (e commentate) nel riquadro di pagina 93. Si ricorda che le costanti sono memorizzate nel file "KS0713.h" e che queste sono sempre scritte in maiuscolo, al fine di distinguerle facilmente dalle variabili, sempre scritte in minuscolo. Abbiamo voluto raggruppare tutte le routine di comando in un riquadro specifico per non appesantire la descrizione dei vari listati.

Ora non ci resta che descrivere le ultime tre routine, le quali per-

mettono di eseguire le seguenti funzioni:

- pulizia memoria DDRAM del controller KS0713
- visualizzazione di un'immagine o un font, prelevando i byte necessari dalla memoria EEPROM esterna.

Iniziamo col descrivere la routine che si occupa di "cancellare" la memoria DDRAM del controller: essa, in pratica, scrive il valore 0x00 su tutta la memoria. La "pulizia" inizia dalla pagina "0x00" all'indirizzo di colonna "0x00" e prosegue fino alla pagina "0x07" all'indirizzo di colonna "0x80": viene ➤

Listato 8

```

void Write_Font_RAM_KS0713(uint Address_EEPROM, uchar Page, uchar
Col_Address, uchar Width, uchar Height)
{
    uchar temp_count;

    KS0713_Page = Page;
    KS0713_Y_Address_MSB = Col_Address >> 4;
    KS0713_Y_Address_LSB = Col_Address & 0x0F;

    I2C_EEPROM_Add.u1 = Address_EEPROM;
    temp_count = Width;

    do
    {
        Set_Address_KS0713_MSB();
        Set_Address_KS0713_LSB();
        Set_Page_KS0713();

        do
        {
            I2C_Process();
            KS0713_Data = I2C_EEPROM_Data[0];
            I2C_EEPROM_Add.u1 += 1;
            Write_Data_KS0713();
        } while (--temp_count != 0);

        if (Height > 1)
        {
            temp_count = Width;
            KS0713_Page = KS0713_Page + 1;
            KS0713_Y_Address_MSB = Col_Address >> 4;
            KS0713_Y_Address_LSB = Col_Address & 0x0F;
        }
    } while (--Height != 0);
}

```

Inizializzazione delle variabili necessarie per l'indirizzamento della memoria DDRAM.

Indirizzamento EEPROM per lettura immagine.

Indirizzamento memoria DDRAM del controller. L'indirizzamento dipende dai valori assunti dalle variabili descritte prima del ciclo "do While".

Routine per la lettura dei dati dalla EEPROM.

Valore da scrivere in DDRAM e letto dalla EEPROM.

Routine per la scrittura nella memoria DDRAM.

Se l'altezza del font è multiplo di 8 bit allora è necessario reindirizzare la memoria DDRAM del controller. Ciò vale sia per l'indirizzo di pagina che di colonna.

come nel caso precedente. Il secondo parametro, "Page", identifica la pagina della memoria DDRAM su cui si vuole scrivere, il terzo parametro è l'indirizzo di colonna (da 0 a 128). Seguono altri due parametri, corrispondenti a larghezza e altezza del carattere da visualizzare. La larghezza è espressa in pixel mentre l'altezza in numero di byte, poiché deve essere sempre un multiplo di 8 bit. A questo punto, abbiamo concluso la spiegazione delle routine software facenti uso dei comandi principali del controllore.

Relativamente ai comandi non citati...

Fin dalle prime pagine di que-

sto articolo, si è potuto constatare che il KS0713 è un dispositivo molto flessibile: però, tale flessibilità è resa possibile solo grazie al gran numero di comandi di cui esso dispone. Noi abbiamo commentato, per quanto possibile, quelli utilizzati più frequentemente. Comunque, nel riquadro alla pagina seguente, trovate l'elenco di tutte le funzioni software richiamabili (ognuna completata da una piccola spiegazione) tramite le quali è possibile gestire tutto il set di istruzioni di cui il display (legga-si anche "il driver KS0713") è dotato.

Funzione dei pulsanti sulla demoboard

Il display descritto in questo articolo può essere inserito e collaudato sulla demoboard presentata nelle puntate precedenti.

Diamo una brevissima descrizione delle funzioni associate ai cinque pulsanti di cui essa dispone, rammentando nel contempo che la capacità di memoria della EEPROM scelta permette di memorizzare fino a quattro immagini e due font.

- Pulsanti "RIGHT" e "LEFT": richiamano l'immagine successiva o precedente
- Pulsanti "UP" e "DOWN": richiamano la funzione "Set_Start_Line_KS0713", quindi eseguono uno scorrimento dell'immagine verso l'alto o verso il basso.
- Pulsante "ENTER" richiama la funzione "Mirror_Bitmap" la quale, a seconda del nu-

effettuata attraverso due cicli "do-loop" ricorsivi, in cui il primo - `do {...} while(--temp_count != 0)` - esegue un conteggio delle scritture all'interno di una pagina, mentre il secondo - `do {...} while(++KS0713_Page != 8)` - conta le pagine scritte. La routine riportata nel listato 7 legge l'immagine da visualizzare sul display LCD dalla memoria EEPROM esterna, ossia quella del microcontrollore. Quindi la trasferisce byte per byte dalla EEPROM alla DDRAM del controller. Tale funzione necessita di due parametri: il primo, "Address_EEPROM", serve per indirizzare la memoria EEPROM esterna e quindi identificare l'indirizzo di partenza da cui leggere; più in generale, determina il punto esatto in cui inizia una sequenza di byte che definisce un'immagine. Il secondo parametro, "Offset", serve per visualizzare correttamente l'immagine nel caso in cui la stessa sia visualizzata specchiata rispetto all'asse Y. Ricordatevi, infatti, che la memoria DDRAM del controller è di 132x65 pixel mentre la nostra immagine è da 128x64 pixel, di conseguenza quando si visualizza l'immagine riflessa rispetto all'asse Y è necessario impostare un offset di 4 pixel. Se, invece, l'immagine non è specchiata l'offset deve essere zero. L'ultima routine realizzata (listato 8) si occupa di scrivere un font sul display. Come la precedente, anche questa necessita di alcuni parametri: il primo, "Address_EEPROM" serve per indirizzare la memoria EEPROM, esattamente

L'elenco dei comandi

```

Funzione "Reset". Codice comando 0xE2.
void Reset_SW_KS0713(void)
{
    Write_Comm_KS0713(KS0713_RESET);
}

Funzione "Display ON". Codice comando 0xAF.
void Set_ON_KS0713(void)
{
    Write_Comm_KS0713(KS0713_DISPLAY_ON);
}

Funzione "Display OFF". Codice comando 0xAE.
void Set_OFF_KS0713(void)
{
    Write_Comm_KS0713(KS0713_DISPLAY_OFF);
}

Funzione "Regulator Resistor Select". Codice
comando 0x20 in "OR" con il valore adeguato di
resistenza, nel nostro caso 0x06 (Vedere valore
"Value").
void Set_RES_KS0713(uchar Value)
{
    Write_Comm_KS0713(KS0713_RES | Value);
}

Funzione "Set Reference Voltage Mode". Codice
comando 0x81.
void Set_REF_Voltage_KS0713(void)
{
    Write_Comm_KS0713(SET_REF_VOLTAGE);
}

Funzione "Set Reference Voltage Register". Codic-
e parametro 0x07 (Vedere valore "Value").
void Set_REF_Voltage_Value_KS0713(uchar Value)
{
    Write_Comm_KS0713(SET_REF_VALUE | Value);
}

Funzione "Select LCD Bias". Codice comando 0xA3.
void Set_BIAS_KS0713(void)
{
    Write_Comm_KS0713(KS0713_BIAS);
}

Funzione "Power Control". Codice comando 0x2F.
void Set_POWER_KS0713(void)
{
    Write_Comm_KS0713(KS0713_POWER_CONTROL);
}

Funzione "Set Static Indicator Mode". Codice
comando 0xAC.
void Set_SIR_KS0713(void)
{
    Write_Comm_KS0713(KS0713_SIR);
}

Funzione "Set Static Indicator Register". Codice
parametro 0x00 (Vedere valore "Value").
void Set_SIR_Value_KS0713(uchar Value)
{
    Write_Comm_KS0713(Value);
}

Funzione "ADC Select". I valori possibili che
può assumere il parametro ADC sono i seguenti:
• 0xA0, quindi direzione normale SEG1 → SEG132
• 0xA1, quindi direzione normale SEG132 → SEG1
void Set_ADC_KS0713(uchar Value)
{
    Write_Comm_KS0713(Value);
}

```

mero di volte che si è premuto il pulsante, esegue delle rotazioni dell'immagine specchiandole rispetto all'asse Y o all'asse X. Dopo quattro pressioni consecutive, l'immagine ritorna alla condizione originale, priva cioè di rotazioni.

Conclusioni

Abbiamo terminato lo studio del KS0713.

```

Funzione "SHL Select". I valori possibili che può
assumere il parametro SHL sono i seguenti:
• 0xC0, quindi direzione normale COM1 → COM64
• 0xC8, quindi direzione normale COM64 → COM1
void Set_SHL_KS0713(uchar Value)
{
    Write_Comm_KS0713(Value);
}

```

Funzione "Entire Display ON". Codice comando 0xA5.

```

void Set_ON_All_LCD_Pixel(void)
{
    Write_Comm_KS0713(KS0713_ON_ALL_LCD);
}

```

Funzione "Entire Display OFF". Codice comando 0xA4.

```

void Set_OFF_All_LCD_Pixel(void)
{
    Write_Comm_KS0713(KS0713_OFF_ALL_LCD);
}

```

Funzione "Reverse Display ON". Codice comando 0xA6.

```

void Set_REV_Displ_ON(void)
{
    Write_Comm_KS0713(KS0713_REVERSE_DISP_ON);
}

```

Funzione "Reverse Display OFF". Codice comando 0xA7.

```

void Set_REV_Displ_OFF(void)
{
    Write_Comm_KS0713(KS0713_REVERSE_DISP_OFF);
}

```

Funzione "Initial Display Line". Codice comando 0x40 in "OR" con il valore di riga desiderato (Vedere valore "Value").

```

void Set_Start_Line_KS0713(void)
{
    Write_Comm_KS0713(KS0713_SET_START_LINE |
    KS0713_Start_Line);
}

```

Funzione "Set Page Address". Codice comando 0xB0 in "OR" con il valore di pagina desiderato (Vedere valore "Value").

```

void Set_Page_KS0713(void)
{
    Write_Comm_KS0713(KS0713_SET_PAGE_ADD |
    KS0713_Page);
}

```

Funzione "Set Colum Address MSB". Codice comando 0x10 in "OR" con il valore di colonna (Nibble più significativo) desiderato (Vedere valore "Value").

```

void Set_Address_KS0713_MSB(void)
{
    Write_Comm_KS0713(KS0713_SET_COL_ADD_MSB |
    KS0713_Y_Address_MSB);
}

```

Funzione "Set Colum Address LSB". Codice comando 0x10 in "OR" con il valore di colonna (Nibble meno significativo) desiderato (Vedere valore "Value").

```

void Set_Address_KS0713_LSB(void)
{
    Write_Comm_KS0713(KS0713_SET_COL_ADD_LSB |
    KS0713_Y_Address_LSB);
}

```

Come per gli altri controllori, il firmware in codice sorgente può essere scaricato dal sito della rivista (www.elettronica.in.it) ed è liberamente utilizzabile come punto di partenza per eventuali applicazioni differenti dalla nostra.

Appuntamento quindi alla prossima uscita, in cui analizzeremo un altro display dotato di un controllore grafico molto potente: il T6963C.