

Mambo, localizzatore portatile programmabile

di Davide Scullino

Nuovo protagonista nel mercato dei ricevitori GPS, MAMBO è un dispositivo di localizzazione completamente programmabile, funzionante a batteria e provvisto di cellulare GSM/GPRS integrato, interfaccia Bluetooth e sensore di movimento. Scopriamolo insieme. Prima puntata.



I sistemi di localizzazione e navigazione assistiti da satellite, sono stati e sono argomento che affrontiamo da sempre con vivo interesse, profondendo sforzi ed energie nella ricerca di nuovi prodotti o di applicazioni innovative; un esempio di ciò è stato il localizzatore portatile descritto nel fascicolo n° 100, vera punta di diamante di quella che è la nostra filosofia in materia di GPS: un dispositivo miniaturizzato, alimentabile a batteria e provvisto di cellulare GSM tramite il quale può sia essere interrogato a distanza per conoscere posizione e percorso di chi lo porta con sé,

sia inviare esso stesso, in caso di necessità, e-mail contenenti le coordinate del momento tramite il servizio SMS. Cercando qua e là, di recente ci siamo imbattuti in un prodotto che somiglia molto al nostro localizzatore GPS/GSM, ma che offre qualcosa in più e che certo interesserà chi opera professionalmente in questo settore: si chiama Mambo, è prodotto dalla Falcom ed è anch'esso un localizzatore GPS provvisto di cellulare integrato GSM/GPRS, il quale, anche in questo caso, può essere utilizzato per fare delle normali telefonate. Dispone di una batteria ricaricabile che gli permette di funziona- ➤



Navigatore satellitare tramite PDA con programma di navigazione ed interfaccia Bluetooth.

re autonomamente e che lo rende adatto sia ad essere affidato a una persona (per esempio un escursionista, che può portarlo nello zaino così da essere rintracciato nel caso si perda o abbia un incidente) sia a venire collocato in un autoveicolo, nel quale può funzionare anche senza essere necessariamente collegato all'impianto elettrico: ciò permette di nascondere più facilmente e di impedirne la scoperta misurando la corrente assorbita a vuoto dalla

batteria dell'auto. Il Mambo può essere chiamato e quindi interrogato a distanza da una postazione di controllo, ma non solo: può inviare SMS contenenti la posizione o effettuare autonomamente chiamate dirette a chi tiene sotto controllo i movimenti della persona o del veicolo sul quale si trova. Particolarità molto interessante è la presenza di un sensore di movimento il quale può controllare importanti funzioni; sul coperchio sono presenti tre pul-

santi ai quali possono essere affidati compiti differenti che dipendono da come il dispositivo viene programmato. In tema di programmazione, va detto che il Mambo è provvisto di un'interfaccia Bluetooth (di classe 2 versione 1.1), la quale consente di dialogare con il modulo ed effettuare tutte le impostazioni e la programmazione, ma anche di sfruttare le funzioni tipiche di un cellulare Bluetooth. Il tutto viene controllato da una CPU ad architettura RISC.

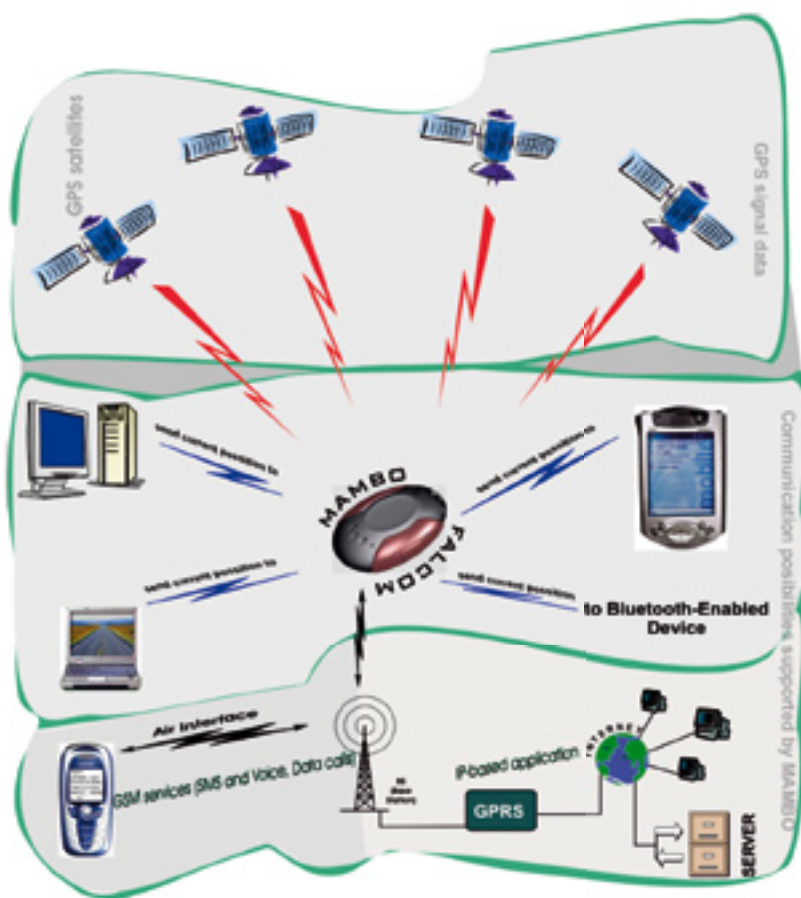
La programmabilità è proprio la caratteristica più importante di questo dispositivo nel senso che tutte le funzionalità disponibili possono essere gestite, correlate o escluse mediante la programmazione iniziale. In pratica possiamo definire, via software, come deve operare l'apparato, escludendo quelle funzionalità che, magari, per il nostro utilizzo, sono superflue e sfruttando al massimo quelle che più ci servono. E' come se potessimo realizzare l'hardware ed il firmware del nostro localizzatore ideale. Se quello che abbiamo detto ha stuzzicato la vostra curiosità, continuate a leggere e scoprirete quali e quante sono effettivamente le potenzialità del Mambo, il primo e forse l'unico localizzatore GPS che integra un vero e proprio telefono cellulare, un'interfaccia Bluetooth, una batteria e un sensore di movimento e che viene fornito completo di software per la programmazione e la gestione di tutte le funzioni.

Di seguito analizzeremo una ad una le varie sezioni del Mambo in modo da poterne sfruttare tutte le potenzialità.

Il localizzatore

La sezione GPS è basata su un ricevitore a 20 canali (ad antenna integrata) capace quindi di agganciare e seguire i segnali di 20 satelliti contemporaneamente.

E qui va fatta una precisazione: se è vero che per localizzare un veicolo



o una persona che porta indosso un dispositivo del genere bastano i segnali di tre satelliti, è anche vero che normalmente oggi non è più così difficile ricevere la portante di 8÷10 satelliti.

Inoltre alla rete tradizionale Navstar sono stati affiancati alcuni satelliti addizionali che hanno lo scopo di aiutare i ricevitori a terra per calcolare la posizione con maggiore precisione. Per cui, anche se oggi è piuttosto difficile riuscire ad utilizzare contemporaneamente tutti i 20 canali, esistono delle situazioni particolari che comunque teoricamente consentirebbero di tracciare anche 14 o 15 satelliti per periodi di tempo variabili nel corso della giornata ed anche dipendenti dalla posizione in cui si trova il GPS.

Non vogliamo ripetere quanto abbiamo già scritto nell'articolo "Nuovi GPS avanzano: l'era dei 32 e dei 50 canali" pubblicato sulla rivista N. 115, ma vi suggeriamo di rileggerlo in quanto gli argomenti tecnici trattati in quella sede sono pertinenti al 100% con questa presentazione. Qui invece vogliamo parlare delle capacità del chipset che governa la sezione ricevente GPS, ovvero il SiRFStarIII prodotto da SiRF Technology, uno dei più importanti produttori di chipset per ricevitori GPS. Il front-

end radio dell'SS3 (abbreviazione di SiRFstarIII) riceve i segnali in banda L1 (1575,42 MHz) con codifica C/A a 1023 MHz.

La parte digitale, invece, che integra Down Converter, Banda Base, DSP, Microprocessore ARM7 e memoria Flash / RAM, gestisce le sentenze GGA, GGL, GSV, RMC secondo il protocollo NMEA0183 (per quanto comunque esso sia in grado di fornire maggiori dettagli tramite il protocollo Binary SiRF, non usato però dal Mambo). Il chipset è molto sensibile: esso presenta infatti un valore di -142 dBm in acquisizione e di -158 dBm in tracking, con tempi di acquisizione di 35 secondi in cold start (effemeridi, almanacchi e ora UTC sono sconosciuti) e meno di 8 secondi in hot start (effemeridi, almanacchi e ora UTC non più vecchi di due ore).

Riguardo alla precisione, il ricevitore ha un'accuratezza migliore di 10 m CEP sulla posizione, di 0,1 m/s sulla velocità e 1 msec rispetto all'orario ricevuto dal sistema GPS.



Sezione cellulare

Il modulo telefonico radio contenuto nel Mambo è un Tri-Band di tipo GSM/GPRS ed è un vero e proprio modem-cellulare con antenna integrata, tant'è che il dispositivo prevede, come accessorio, un auricolare da usare a corredo, se si intende sfruttarne tutte le funzionalità telefoniche in auto. Non manca >



Hardware Setup



Sfilare il coperchio e rimuovere la batteria



Inserire la SIM



Chiudere il porta SIM



Inserire la batteria



Richiudere il coperchio

no microfono, altoparlante e la possibilità di conversare a viva-voce; per la connessione di un auricolare cablati, è previsto un jack da 2,5

mm di diametro a quattro poli. Naturalmente è previsto anche l'abbinamento con un auricolare wireless Bluetooth. Fatte salve le limitazioni imposte dalla Legge, nell'utilizzo del Mambo come dispositivo di localizzatore e ascolto ambientale all'insaputa del conducente, le funzioni vocali permettono di sfruttare l'apparecchio anche per ascoltare da remoto quanto viene detto all'interno del veicolo; infatti, opportunamente programmato, il Mambo può attivare il proprio microfono quando riceve una chiamata e permettere al chiamante di ascoltare quel che avviene nel veicolo sul quale è collocato. Oltre alle funzioni vocali, supporta gli SMS e la ricetrasmmissione di dati; in modalità GPRS, comunica secondo i protocolli TCP/IP e PPP (GPRS classe 10, B).

Sul piano telefonico, va detto che il nostro localizzatore è disponibile in due versioni, adatte ciascuna ad uno specifico mercato: il Mambo 55 e il Mambo 56; il primo è progettato per funzionare a 900/1800/1900 MHz (frequenze utilizzate nel sistema cellulare europeo) mentre il secondo lavora a 850/1800/1900 MHz (sistema radiomobile americano).

Nelle applicazioni che descriveremo nei prossimi articoli faremo sempre riferimento alla versione del Mambo 55. Il connettore per la SIM-Card è accessibile sollevando il coperchio e rimuovendo la batteria.

Interfaccia Bluetooth

Come accennato, Mambo integra un dispositivo di comunicazione Bluetooth utilizzato per collegare un auricolare o altro apparato standard wireless (Bluetooth, ovviamente) ma anche per la comunicazione con il Personal Computer nel quale va installato il programma di configurazione; questo, anche perchè, il dispositivo non ha altra porta per dialogare con l'esterno: il programma di funzionamento ed eventuali modifiche, vengono tutti caricati nella



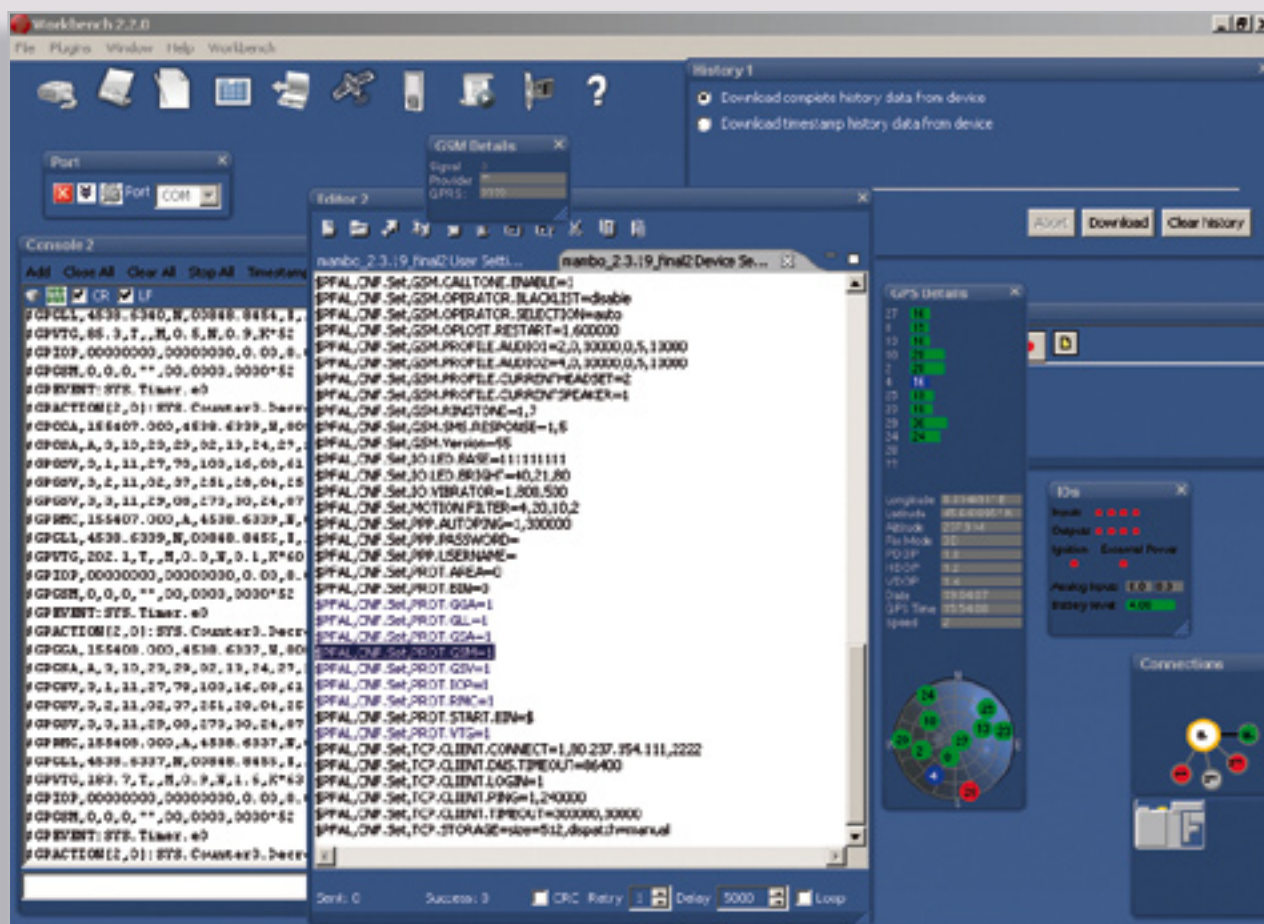
memoria del Mambo tramite l'interfaccia wireless Bluetooth.

Entrando nel dettaglio, l'interfaccia contiene una ricetrasmittente con potenza conforme alla classe 2, versione di protocollo BT 1.1 e profilo SPP (Serial Port Profile) integrato. L'antenna integrata garantisce la copertura entro una distanza di 10 metri circa.

Oltre a garantire la connessione col PC per la configurazione e consentire l'utilizzo di un auricolare wireless, il canale Bluetooth può servire per inviare i dati rilevati dal ricevitore GPS ad un Pocket PC o ad uno Smartphone equipaggiati con interfaccia analogica e dotati di un software di navigazione con la relativa cartografia. In tal modo il Mambo consente di realizzare un vero e proprio navigatore satellitare: basta entrare in auto e attendere che i due apparati si riconoscano, per poi aprire il canale dati col palmare e visualizzare sullo schermo di quest'ultimo la posizione o il percorso corrispondente.

Sensore di movimento

Il dispositivo integra un preciso sensore di movimento basato su un accelerometro capace di rilevare lo spostamento lungo i tre assi (verticale, orizzontale in avanti e di lato); il sensore è completamente configurabile da software, nel senso che nel programma da caricare nel Mambo



Definire semplicemente "programma" l'ambiente di sviluppo, programmazione ed utilizzo di MAMBO è piuttosto riduttivo: esso infatti dispone di tantissime opzioni per mezzo delle quali è possibile plasmare le funzionalità del sistema in relazione alle proprie necessità. Anche il nome scelto dalla stessa Falcom ne è la conferma: infatti la suite di programmi a corredo di MAMBO viene chiamata "Workbench", banco di lavoro.

Nel corso delle prossime puntate daremo ampio spazio a Workbench in quanto tutti gli esempi pratici che realizzeremo verranno progettati, trasferiti e verificati tramite questo tool. In questo riquadro - essendoci occupati a lungo nell'articolo della parte squisitamente hardware - vogliamo anticipare qualche informazione sull'ambiente di

sviluppo allo scopo di fornire una panoramica completa delle potenzialità di questo sistema. Naturalmente, in seguito, approfondiremo queste informazioni con esempi pratici e suggerimenti d'utilizzo.

Nell'immagine in alto abbiamo rappresentato alcune finestre particolarmente significative che andiamo a commentare.

Port: seleziona la COM port BT a cui MAMBO è virtualmente collegato.

GSM Details: fornisce indicazioni sul provider GSM al quale il ricevitore è connesso nonché sulla qualità del segnale.

Console 2: riporta la sequenza di dati e comandi scambiati tra PC e ricevitore. La visualizzazione è in formato testo e non c'è limite al numero di console apribili, sulle quali possono essere applicati filtri di visualizzazione.

Editor 2: permette di gestire la parte Firmware del MAMBO, ovvero leggere / scrive-

re il file di configurazione. Anche in questo caso si possono aprire più finestre di Editor.

History 1: permette di leggere il file che contiene le coordinate del percorso memorizzato quando il MAMBO viene usato come data logger.

GPS Details: rappresenta la visualizzazione grafica dell'immagine astrale della rete satellitare. Permette di ricavare informazioni sulla qualità del segnale, sul numero di satelliti visibili e sulla relativa posizione nella sfera celeste, nonché di indicare lo stato di navigazione in cui il GPS si trova.

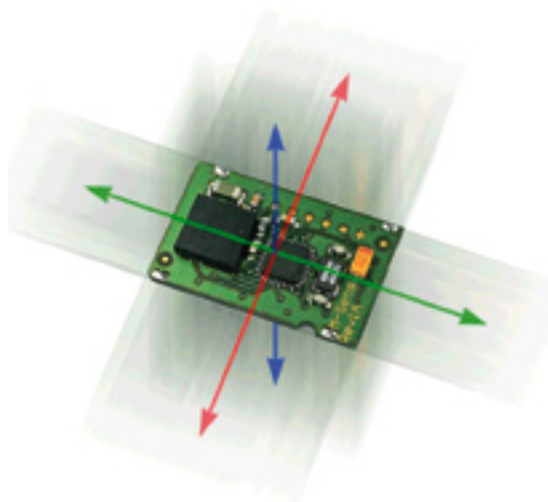
IOs: fornisce alcune indicazioni relative allo stato degli ingressi e delle uscite, oltre allo stato della batteria.

Connections: permette di collegare le finestre sia tra loro sia direttamente alla seriale del MAMBO.

si può definire cosa il dispositivo deve fare quando viene mosso. Ad esempio, si può decidere che l'unità sia normalmente in stand-by e

che, dopo un qualsiasi movimento, si risvegli ed inizi a funzionare normalmente; oppure che in caso di spostamento chiami una persona o

invii, tramite SMS, le coordinate in tempo reale o dopo un ritardo più o meno lungo. L'utilizzo del sensore per rilevare quando il dispositivo si ➤



L'accelerometro integrato a tre assi rileva gli spostamenti permettendo di attivare particolari funzioni come, ad esempio, l'invio di un SMS.

muove e la conseguente possibilità di abilitare molte funzioni solo in tal caso, è di grande aiuto per ottimizzare il consumo dell'energia immagazzinata nella batteria che, in pratica, si traduce nel miglioramento dell'autonomia d'esercizio che può andare ben oltre le ore dichiarate dal costruttore.

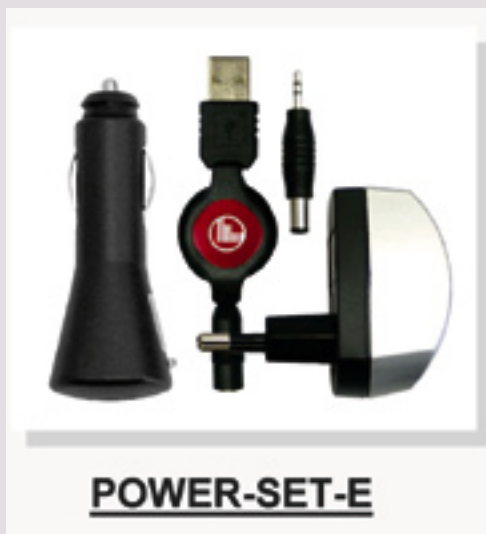
Alimentazione

Mambo è un dispositivo portatile e facilmente installabile a bordo di autoveicoli grazie all'impiego di una batteria ricaricabile a ioni di litio da 3,7 volt fornita di serie; il costruttore offre anche la possibilità di scegliere tra due capacità, ossia 850 mA/h (che garantisce un'autonomia di cinque ore) e 1700 mA/h (dieci



Grazie all'auricolare in dotazione (4), Mambo può essere usato per effettuare normali telefonate: si gestisce con i tasti 1, 2 e 3.

Gli accessori



MAMBO viene fornito con una nutrita dotazione di accessori. Infatti, oltre al ricevitore, disponibile sia in versione americana (la sezione GSM è soggetta a differenti regolamentazioni e l'adattatore di rete è dotato di presa bipolare a passo USA) che europea (nella foto in alto è visibile il POWER-SET-E in cui l'adattatore dispone di spina italiana), e al pacco batterie, la confezione include:

- auricolare con avvolgicavo a molla e pulsante di controllo;
- cordicella dotata di moschettoni, per trasportare al collo il MAMBO;
- cavo adattatore da USB a jack, da utilizzare per ricaricare la batteria;
- adattatore da jack stereo 2,5 mm a jack universale per ricaricare sia il cellulare che il ricevitore GPS/GSM;
- riduttore di tensione veicolare da collegare alla presa dell'accendisigari; fornisce una tensione di uscita di $5\text{ V} \pm 5\%$ con una corrente di 450 mA su un connettore USB e va collegato al MAMBO tramite l'apposito cavo adattatore;
- adattatore di tensione da parete disponibile sia con spina bipolare americana che bipolare europea, a seconda della confezione richiesta; anch'esso fornisce una tensione di uscita di $5\text{ V} \pm 5\%$ con una corrente di 450 mA con connettore di uscita tipo USB;
- CD di installazione della suite Workbench; la versione a corredo è sempre la più recente disponibile nel momento in cui viene preparata la confezione.

ore di funzionamento continuato). L'autonomia d'esercizio dichiarata vale per l'uso continuativo di tutte le funzioni, ma è chiaro che ottimizzando i consumi, ad esempio programmando lo stand-by quando non viene rilevato movimento dall'accelerometro, essa aumenta decisamente.

L'accumulatore è posizionato sotto il coperchio, sopra il lettore per la SIM e viene ricaricato attraverso un regolatore interno collegato al connettore d'alimentazione posto sull'involucro del dispositivo; di serie viene fornito un cavo USB per prelevare l'alimentazione necessaria alla ricarica dall'omonima porta di un computer, ma anche di un alimentatore da auto, provvisto di presa accendisigari, nonché di un alimentatore / caricabatteria da rete universale, funzionante a $100\div 240\text{ Vac}$, sia a 50 che a 60 Hz, utilizzabile quindi in tutto il mondo. Tanto l'adattatore da rete che quello da auto, rendono disponibile l'alimentazione per il Mambo mediante una presa USB della quale sono collegati i soli contatti + e -; ciò permette di introdurre direttamente il cavo in

dotazione al Mambo. Come accennato, la praticità dell'alimentazione a batterie si apprezza sia quando il dispositivo debba "viaggiare" nello zaino di un escursionista che si muove a piedi o in mountain-bike, sia nel caso si intenda tracciare il percorso di un autoveicolo; in una situazione del genere, la capacità di operare in piena autonomia (senza necessità di connessione all'impianto elettrico della vettura) si apprezza maggiormente quando è necessario occultare il localizzatore che così non può essere scoperto neanche andando a verificare il consumo a vuoto della batteria.

Questo metodo è uno dei più usati per scoprire se a bordo di un'auto è stato installato un localizzatore o un sistema di intercettazione ambientale, sebbene sia aggirabile usando dispositivi a batteria o in grado di spegnersi per alcune ore quando la batteria della vettura viene scollegata (operazione necessaria per collegare in serie all'impianto un amperometro per la misura).

Programmazione

Mambo è governato da un microprocessore RISC che esegue i pochi passi necessari a dialogare con il computer per poter caricare il programma di funzionamento prescelto; nella sua Flash EPROM è dun-

que collocato quello che possiamo considerare il suo BIOS, l'insieme delle istruzioni fondamentali per attivare una connessione Bluetooth e accettare le disposizioni che il programma di configurazione gli invie-

rà dal PC, che, evidentemente, dovrà a sua volta disporre di un'interfaccia Bluetooth. In altre parole, una volta acceso, il nostro apparecchio aspetta che gli si carichi il programma da eseguire: i tre pulsanti di cui dispone e i led bicolore posti sul coperchio non fanno inizialmente nulla, però possono essere personalizzati. Il software prevede la possibilità di assegnare a ciascun tasto una specifica funzione oppure un insieme di funzioni, ma inizialmente nessun tasto è attivo.

Più esattamente, va detto che il pulsante centrale serve all'accensione e allo spegnimento, ma per il resto, salvo diversa programmazione, non fa altro.

Oltre alle funzioni di localizzazione, invio delle coordinate da Bluetooth e tracciamento dei percorsi, il programma dà la possibilità di genera-

Sezione GPS

- Chipset SiRFStarIII di SiRF Technology (www.sirf.com);
- Frequenza: L1 a 1.575,42 MHz, C/A code a 1.023 MHz, con Front End ad alta sensibilità;
- Numero canali: 20 paralleli, spread spectrum;
- Sensibilità: -142 dBm in acquisizione; -158 dBm in tracking;
- Tempi di acquisizione: Cold Start 38 s, Warm Start 35 s, Hot Start 1 s;
- Precisione:
10 m CEP per la posizione, 0,1 m/s per la velocità, 1 μ s per il tempo;
- Correzione differenziale: WAAS.

Sezione Telefono

- Bande di frequenze GSM: 900 / 1800 / 1900 MHz
- Sezione GPRS: classe 10
- Modi operativi: Voce, Dati, SMS, TCP/IP, PPP

Sezione Bluetooth

- Versione protocollo: Bluetooth 1.2
- Classe di funzionamento: classe 2, 10 m in campo aperto.

Varie

- Batteria a ioni di litio, disponibile a scelta da 850 o 1600 mA/h
- Dimensioni: 85 x 59 x 25 mm, peso 70 g.
- Microfono ed altoparlante integrati.
- Jack da 2,5 mm a 4 poli per auricolare
- Tre pulsanti e tre led con funzionalità definibile da programma.
- Sensore di movimento (accelerometro a tre assi) integrato.





Per gli sviluppatori è disponibile un'elegante valigetta demo la quale contiene, oltre alla suite Workbench, entrambe le versioni di MAMBO (europea ed americana) nonché tutti gli accessori.

re un allarme se il Mambo esce da una zona definita sulla cartografia; una sorta di allarme che permette di avvertire chi deve sorvegliare una persona o un veicolo se questi si allontanano da un'area predefinita. Tale funzione è di grande importanza in molti ambiti: nel controllo veicolare permette di verificare se l'autista di un camion, di un autobus di linea o di un qualsiasi veicolo aziendale durante la giornata "divaga" ed esce dal proprio percorso; inoltre consente di sapere se l'auto in cui il Mambo è installato viene portata via e/o si allontana dalla zona nella quale dovrebbe trovarsi, ovvero se varca i confini di una regione o dello Stato. Nell'ambito del controllo delle persone, può essere usato per av-

vertire il Soccorso Alpino o chi di dovere se un escursionista o uno scalatore esce dalla zona considerata sicura e si va ad avventurare in luoghi che potrebbero presentare il rischio di frane, valanghe o costituire pericolo per la presenza di animali che possono aggredire l'uomo, e via di seguito. In tutti i casi esposti, il localizzatore, rilevato l'allontanamento dalla zona impostata mediante un apposito software di cui parleremo nella prossima puntata, può chiamare il personale ad-

detto alla sorveglianza del veicolo o della persona che lo porta con sé, ovvero inviare un SMS che avverta della situazione, cosicché si possano prendere i provvedimenti del caso (ad esempio una telefonata all'escursionista per avvertirlo di tornare indietro).

Per impostare i limiti della zona va utilizzato il solito programma di configurazione: questi dati vanno poi trasferiti (via Bluetooth) con l'apposita funzione nella memoria del Mambo.

Il dispositivo è molto versatile e lo dimostra il fatto che di esso è possibile personalizzare qualsiasi funzione, tasti e led di segnalazione compresi; tuttavia va detto che presenta una limitazione, se vogliamo, concettuale: i progettisti che l'hanno sviluppato, forse per un po' di presunzione, hanno realizzato istruzioni rigide; in pratica, i comandi per impostarne il funzionamento sono specifici per ogni condizione, quindi va da sé che il Mambo può essere programmato per eseguire determinate operazioni e solo quelle.

Non è possibile impostare combi-

nazioni tra le varie funzioni che non siano state previste da un apposito comando.

Diciamo presunzione perché in effetti chi ha progettato il Mambo ha probabilmente pensato di aver previsto tutte le possibili situazioni di impiego e in effetti, dando uno sguardo al set di comandi, sembra proprio così; sarebbe stato comunque auspicabile avere a disposizione delle istruzioni più flessibili.

Il discorso sulla programmazione del dispositivo si comprende facendo un paragone con i linguaggi di programmazione dei microprocessori: solitamente ogni istruzione è associata a dei parametri che permettono di usare un comando per fare diverse cose, a seconda dei parametri definiti di volta in volta; invece, nella programmazione del Mambo per ogni azione esiste un comando specifico, quindi due azioni simili non possono essere impostate impartendo la stessa istruzione con un diverso parametro.

Questo spiega perché la lista delle istruzioni è decisamente lunga.

Per fare un esempio, se con la pressione del tasto 1 si desidera far partire una telefonata diretta ad un certo numero, per abbinare al predetto pulsante un'altra funzione oltre alla telefonata (per esempio l'invio di un SMS) non si può utilizzare il medesimo comando ma bisogna impartirne uno a sé stante.

Nella prossima puntata analizzeremo il software di programmazione col quale si sviluppa il programma da caricare nel Mambo e con cui, a stesura ultimata, lo stesso programma viene caricato via Bluetooth; avremo così modo di vedere quanto accennato relativamente alla struttura del set dei comandi.

La programmazione si esegue su qualunque Personal Computer mediante il software fornito direttamente dalla casa costruttrice, che opera in ambiente Microsoft Windows.

Localizzatore Mambo, funzioni e programmazione

di Davide Scullino

Si chiama Mambo ed è stato sviluppato per trovare impiego a 360 gradi nell'ambito della localizzazione e del controllo a distanza di persone e automezzi; le sue funzioni e il nutrito set di istruzioni, permettono di sfruttarlo per svolgere tanti altri compiti di contorno, quali l'abbinamento a un PC palmare per realizzare un navigatore satellitare. Seconda puntata.



Nel fascicolo precedente abbiamo introdotto un dispositivo estremamente versatile, nato per le applicazioni inerenti alla localizzazione e personalizzabile per adattarlo al compito cui si intende destinarlo; si tratta di Mambo, localizzatore GPS (prodotto dalla Falcom) provvisto di cellulare integrato GSM/GPRS, del quale possono essere utilizzate tutte le funzioni vocali (per effettuare una normale conversazione o trasmettere l'audio rilevato nell'ambiente). Mambo dispone di una batteria ricaricabile che gli permette di funzionare autonomamente; può essere chiamato e quindi interrogato

a distanza da una postazione di controllo, ma non solo: può inviare SMS contenenti la posizione o effettuare esso stesso chiamate dirette a chi deve tenere sotto controllo i movimenti di una persona o di un veicolo. Ancora, dispone di un efficace sensore di movimento che può essere abbinato ad alcune delle sue funzioni, per esempio facendo accendere il dispositivo solo quando è in movimento. Sul coperchio sono presenti tre led bicolore e quattro pulsanti, ciascuno dei quali può svolgere compiti dipendenti dalla programmazione.

In tema di programmazione, va detto che Mambo è prov- ➤

visto di un'interfaccia Bluetooth, la quale permette sia di dialogare con il modulo durante le fasi di impostazione e programmazione, sia di fare tutto ciò che un cellulare fa con il Bluetooth.

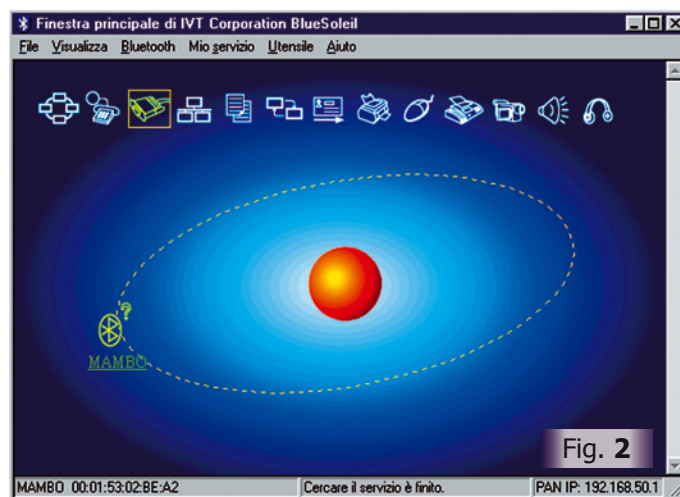
Mambo è governato da un microprocessore RISC che esegue i pochi compiti necessari a dialogare con il computer per caricare il programma di funzionamento prescelto; nella sua Flash EPROM è dunque collocato quello che possiamo considerare il suo BIOS, l'insieme delle istruzioni fondamentali per attivare una connessione Bluetooth.

Spiegate le funzioni di massima, questo mese entriamo nel vivo delle operazioni di programmazione, vedendo come si instaura la comunicazione via Bluetooth, come si interroga il dispositivo, come si scrive in esso un programma o come possono modificare una o più istruzioni: insomma, la gestione da PC.

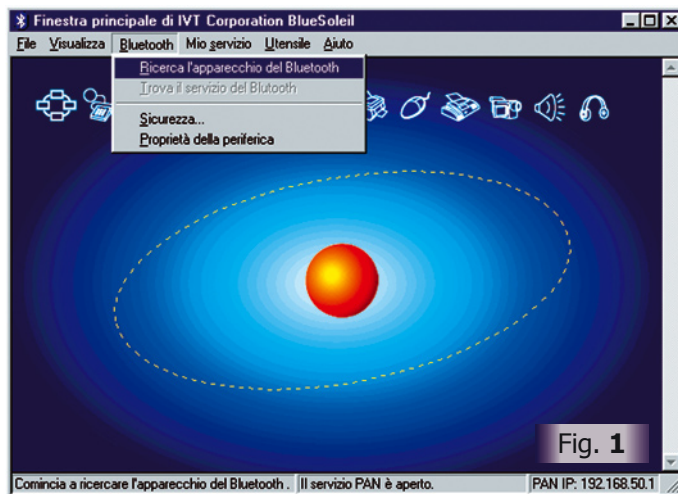
Instaurare la comunicazione

Quando viene messo in funzione per la prima volta, il nostro apparecchio si accende e aspetta di ricevere il programma da eseguire; in esso gira un programma essenziale, implementato in fabbrica, che assegna le funzioni di base ai pulsanti e ai led bicolore di cui Mambo dispone. Tra le operazioni avviate dal software precaricato c'è la possibilità di aprire una sessione di comunicazione via Bluetooth, realizzata attivando l'apposita interfaccia e verificando quando essa comunica la richiesta da parte di un computer presente nel raggio di copertura. Non appena viene rilevata una richiesta di comunicazione, il programma avvia il dialogo. Per comunicare con il dispositivo occorre un computer provvisto di interfaccia Bluetooth standard, ovvero equipaggiato con un "dongle" USB, nel quale sia stato installato un apposito programma di gestione di periferiche Bluetooth;

a riguardo va detto che normalmente i costruttori forniscono, insieme all'hardware e ai driver corrispondenti, anche un semplice software di comunicazione. Nel nostro caso, le prove sono state eseguite usando BluSoleil (versione 1.4.9.5) della IVT Corporation, la cui interfaccia utente è illustrata nella Figura 1.



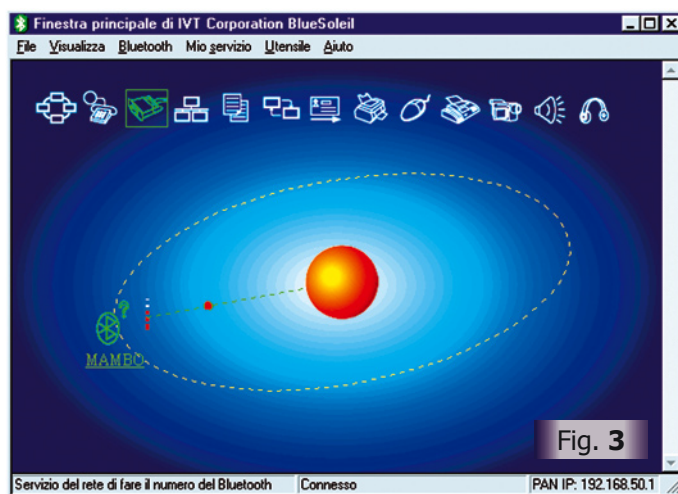
Per cercare i dispositivi attivi nel campo di copertura, bisogna aprire il menu Bluetooth e fare clic sul comando Ricerca dispositivi Bluetooth (Figura 2); se non è stata ancora aperta una COM virtuale, il computer chiede di farlo, altrimenti procede e mostra, al termine della ricerca, le icone corrispondenti agli apparati rilevati. In questa fase,



il programma BlueSoleil esegue una ricerca senza attivare la comunicazione. Per attivarla, bisogna prima

assegnare una COM virtuale al dispositivo, tramite un comando di assegnazione della seriale; quindi si sceglie l'indirizzo di COM (3, 4, 41...) fra quelli liberi. Tale procedura viene effettuata dall'interfaccia utente del programma di dia-

logo Bluetooth. Nel caso di BlueSoleil, l'apertura della COM virtuale si effettua facendo clic, nella finestra di dialogo principale di Figura 1,



sul nome del dispositivo trovato al quale si intende assegnarla; a questo punto risulta evidenziato il pulsante corrispondente al servizio della porta seriale (l'icona con il connettore seriale) sul quale bisogna fare clic per accedere alla finestra di dialogo con cui il computer comunica con la COM aperta e chiede se deve procedere alla connessione. Facendo clic sul pulsante "Sì" viene impartito il comando che avvia la comunicazione: entro alcuni secondi, accanto all'icona dal nome Mambo visibile nella finestra di lavoro del programma (che visualizza i dispositivi trovati nel campo di copertura del Bluetooth), deve apparire un tratteggio rosso verticale e un punto dello stesso colore che si sposta verso la zona centrale dello schermo (Figura 3); ciò significa che la comunicazione è stata instaurata.

Il workbench

Falcom, produttore di Mambo, ha realizzato un software applicativo per ambiente Microsoft Windows (98, 2000, XP) ideale per svolgere tutte le operazioni di programmazione, trasferimento dei dati, acquisizione della memoria riguardante le coordinate del ricevitore GPS, ecc. Il programma in questione si chiama workbench (letteralmente, banco da lavoro) e viene fornito su CD-ROM nella stessa confezione di Mambo. L'interfaccia utente del workbench

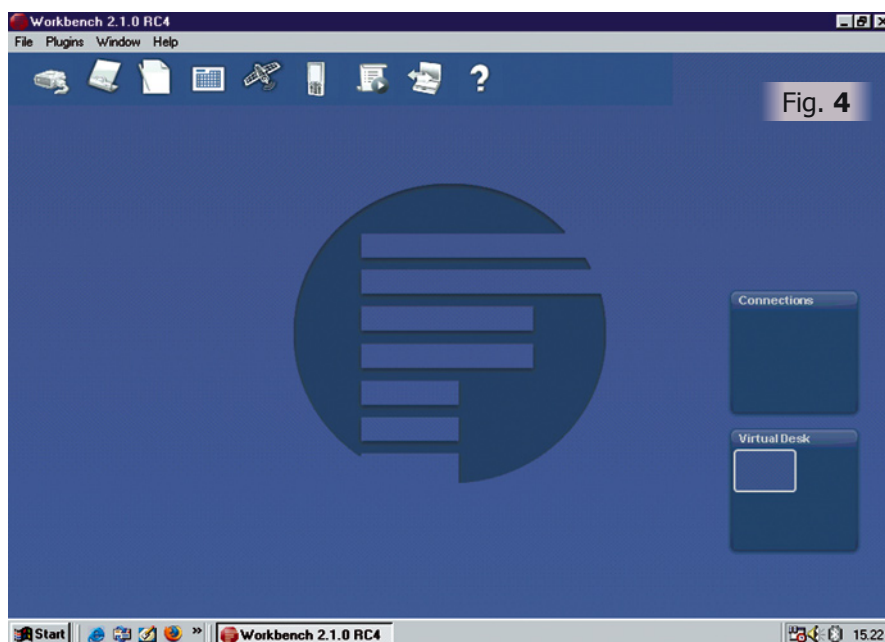


Fig. 4

Falcom (riferita alla versione 2.1.0 RC4) è illustrata nella Figura 4; la finestra di lavoro contiene alcuni menu a tendina e delle icone, corrispondenti ciascuna a una precisa funzione. Le icone sono in realtà i pulsanti della barra degli strumenti, che riassume le principali funzioni; facciamo osservare che i comandi corrispondenti possono essere impartiti anche dal menu *Plugins*.

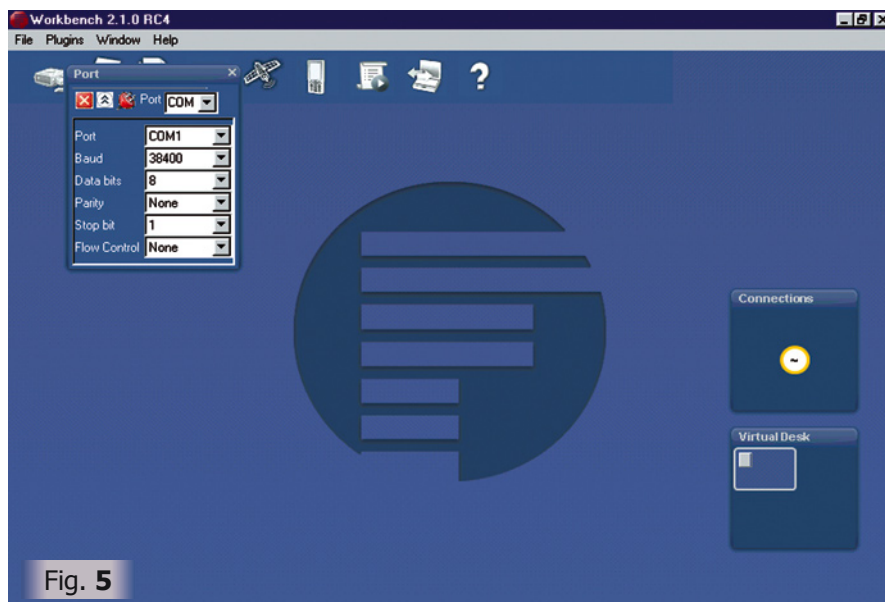
Nella finestra principale del Workbench sono collocate due sezioni chiamate *Virtual Desk* e *Connections*; la prima, situata in basso a destra nella finestra principale, rappresenta un'estensione dell'intero desktop, che si estende ben oltre l'area visualizzata nello schermo del computer.

Virtual Desk è, insomma, una miniatura dell'effettivo piano di lavoro e le finestre aperte (corrispondenti alle funzioni del workbench) vengono visualizzate in essa come icone che seguono la posizione delle finestre vere e

proprie; il rettangolo corrisponde all'area del desktop visualizzata nello schermo. Trascinando con il mouse una finestra da una parte all'altra dello schermo, si vede la sua icona in miniatura che si sposta lungo la finestra del Virtual Desk; aprendo una nuova finestra, ovvero una nuova funzione, la finestra precedente viene spostata fuori dal rettangolo. Insomma, Virtual Desk può essere considerato un monitor di quello che si fa lavorando con il workbench, tuttavia è qualcosa di più: è uno strumento interattivo agendo sul quale, con il puntatore del mouse, si possono spostare le finestre aperte direttamente sullo schermo, senza agire sul desktop reale; sempre in Virtual Desk, si può passare da una finestra all'altra, semplicemente facendo clic sulla miniatura corrispondente. L'altra sezione (*Connections*) è molto più importante perché riassume in un riquadro le connessioni realizzate o realizzabili; in altre parole, riassume le funzioni aperte e i dispositivi Mambo attualmente rintracciati dal programma di comunicazione Bluetooth e in collegamento con il Workbench. È importante perché è in essa che le funzioni si assegnano a uno dei dispositivi trovati, ovvero che le si abbina alla COM virtuale



corrispondente a una sessione di comunicazione aperta con un Mambo. *Connections* serve essenzialmente quando più dispositivi Mambo sono connessi al computer; in tal caso permette di passare agevolmente dall'uno all'altro, ovvero di sconnettere, per esempio, l'editor da un Mambo per assegnarlo ad un altro. Alla luce di quanto detto riguardo alla sezione *Connections*, la si può considerare "la procedura standard di lavoro con Workbench": avviato il programma, per prima cosa bisogna decidere quale delle funzioni attivare, cioè se si desidera operare un editing (con Editor) o inviare comandi direttamente a un Mambo; fatto ciò, si avvia la funzione corrispondente dalla barra degli strumenti. Quando appare sullo schermo la relativa finestra di dialogo, ci si porta nella sezione *Connections*, che deve mostrare un cerchio rosso con il nome della funzione in miniatura. Ora, se esistono COM virtuali aperte e assegnate al programma, sempre nello stesso riquadro le si vede sotto forma di cerchi gialli; in caso contrario, bisogna avviare la procedura di apertura di una porta, definendo in essa il numero della COM usata dal dispositivo Bluetooth per gestire la comunicazione con il Mambo da controllare con Workbench. Il numero della COM in questione si ricava dal programma di gestione



dell'interfaccia Bluetooth (nel nostro caso, BlueSoleil).

Infine, individuata la porta virtuale e atteso che la relativa icona gialla appaia in *Connections*, basta fare clic sul cerchio rosso corrispondente alla funzione da avviare; a questo punto il Workbench conferma l'assegnazione della funzione alla sessione di comunicazione relativa al Mambo desiderato, facendo divenire verde il cerchio. La procedura può sembrare complessa quindi passiamo subito a vedere le funzioni del Workbench, una ad una, iniziando proprio da quella che viene per prima sia nella barra degli strumenti, sia nella sequenza di lavoro con i Mambo.

Selezione porta

Prima di avviare ogni sessione di lavoro in Workbench, bisogna iniziare il dialogo via Bluetooth; pertanto, dopo aver avviato il software Falcom occorre, innanzitutto, accedere alla funzione *Open a new serial port view*, semplicemente facendo clic sulla prima icona di sinistra che appare nella schermata principale. Si accede così alla relativa finestra di dialogo (Figura 5) che riepiloga le porte disponibili e le impostazioni (baud-rate, bit di parità e di dati, ecc.) corrispondenti; in essa l'utente deve scegliere la COM che, dalle

informazioni ricavate dal programma di gestione del Bluetooth, corrisponde alla porta virtuale che il computer ha assegnato all'interfaccia BT del Mambo. La COM virtuale è stata definita nel programma di gestione del BT (nel nostro caso il Blue Soleil) al momento dell'avvio del servizio e della registrazione dei dispositivi.

Definita la porta da usare per interfacciare il Workbench con l'ambiente Bluetooth e impostati i parametri della comunicazione, per stabilire la connessione bisogna fare clic sul terzo (da sinistra) dei pulsanti (*Open Port*) della finestra di dialogo *Port*, quello rosso che somiglia a una spina della corrente. Da questo momento il Workbench si affaccia sulla rete di dispositivi Bluetooth usando l'interfaccia del computer; nella sezione *Connections*, la COM aperta appare come un cerchio giallo. Per chiudere una COM virtuale collegata al Workbench si può procedere in due modi: fare nuovamente clic sul pulsante con la spina o chiudere la finestra di dialogo corrispondente alle impostazioni delle COM. Alla luce di ciò, si badi che una volta aperta la porta, se si intende lavorare con essa non si può chiudere la finestra aperta con *Open a new serial port view*; bisogna lasciarla aperta.



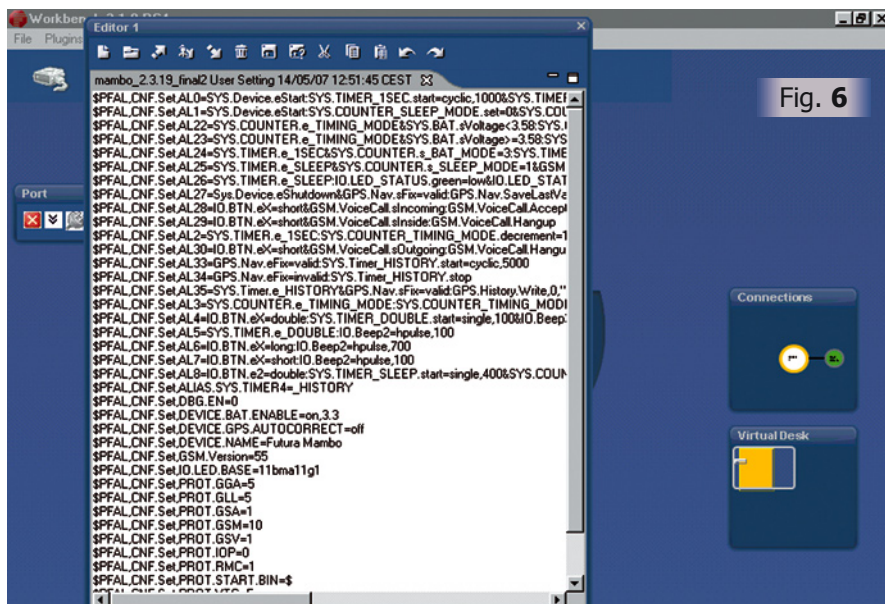


Fig. 6

Editor

Si tratta di una sorta di programma di creazione e modifica di testi, solo che è destinato alla visione, alla scrittura e alla modifica del programma diretto verso il Mambo. Più esattamente, *Editor* permette di vedere il programma acquisito da un dispositivo Mambo, apportare in esso delle modifiche, oppure scrivere ex novo un programma di lavoro.

Per acquisire il programma da un Mambo, oppure inviare a quest'ultimo un nuovo listato o una modifica, si deve passare dalla Console; è tuttavia possibile trasmettere una singola riga di comando semplicemente facendo doppio clic sulla riga corrispondente, quindi evitando di passare dalla Console.

Vediamo i comandi contenuti nella finestra di dialogo dell'*Editor*, analizzando i relativi pulsanti collocati in alto, a partire da sinistra: il primo (*Open New Editor*) si comprende sapendo che *Editor* è una funzione multisessione, nel senso che si possono aprire più sessioni di editing così da poter lavorare su più programmi senza dover necessariamente chiudere un listato. Ciò è di grande utilità quando, ad esempio, si deve scrivere una parte di programma destinata a un Mambo copiandola da quello di un altro dispositivo.

Ebbene, il pulsante in questione permette di aprire una nuova sessione di editing.

Il secondo comando è *Open Configuration File* e serve ad aprire un file di configurazione salvato in una delle periferiche di memorizzazione del computer; appare la finestra di dialogo "Risorse del computer" dalla quale l'utente può indicare qual è il file contenente la configurazione da caricare per editarla ed eventualmente inviarla al Mambo connesso. Il terzo comando è *Start Sending Configuration*: facendo clic sul pulsante corrispondente, il Workbench invia il listato (quindi il programma aperto nell'*Editor*) al Mambo connesso; se l'utente non ha definito alcuna connessione, il programma risponde con una finestra di avviso contenente il messaggio *Editor is not Connected. Please check connections*. È quindi necessario andare nella sezione *Connections* e indicare quale dispositivo va connesso all'*Editor*.

Va notato che impartendo il comando in questione la nuova configurazione inviata prende il posto di quella esistente, nel senso che la memoria di configurazione del Mambo viene cancellata e sostituita con la configurazione scritta nell'editor.

Facendo clic sul pulsante *Read User*

configuration, si impartisce il quarto comando; con esso il software ordina al computer di interrogare il dispositivo Mambo connesso e, ad operazione ultimata, presenta nella finestra dell'*Editor* il listato di configurazione letto dalla memoria di programma del Mambo attualmente connesso (Figura 6).

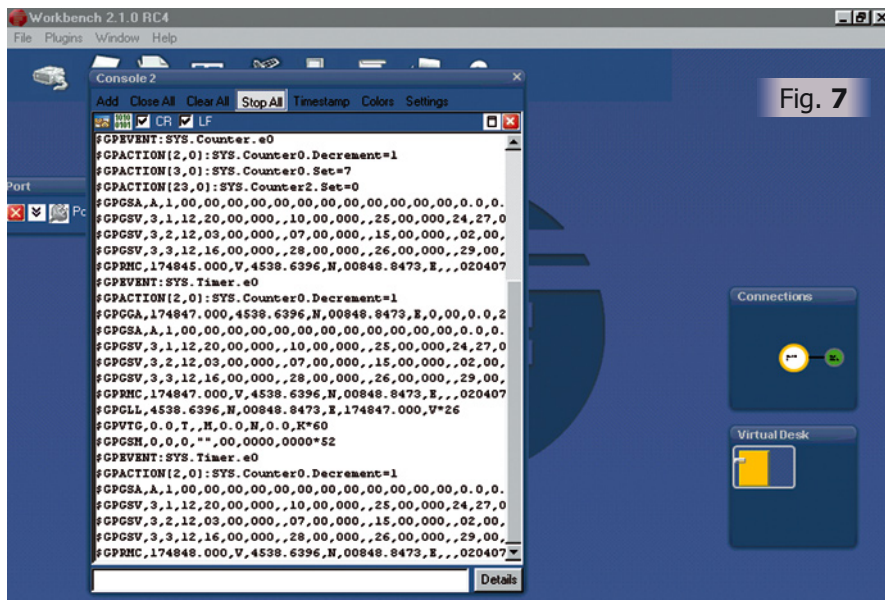
Il pulsante accanto serve a impartire il comando che dà la configurazione del dispositivo attualmente in uso.

Il sesto dei pulsanti (ha un cestino per icona...) serve a cancellare il contenuto della finestra dell'*Editor*; va usato quando si intende rimuovere le righe di comando scritte per ricominciare da capo. Si noti che la rimozione di un listato, caricato con il comando *Open Configuration File*, dalla finestra dell'*Editor* non comporta l'eliminazione del file dall'hard disk.

Il comando *Save Current Editor Content* (settimo pulsante) apre la solita finestra di dialogo "Risorse del computer", e permette di definire in quale posizione (unità a disco, cartella) salvare il listato di configurazione correntemente aperto nell'*Editor*, oltre al nome che si desidera dare al relativo file.

Il pulsante accanto corrisponde alla funzione Windows "salva con nome" e apre una finestra di dialogo uguale a quella che nel sistema operativo corrisponde a tale funzione. ➤





I pulsanti che restano servono a svolgere le funzioni tipiche di un editor di testo, ossia *Taglia*, *Copia*, *Incolla*, *Annulla modifica* e *Annulla l'operazione annullata*.

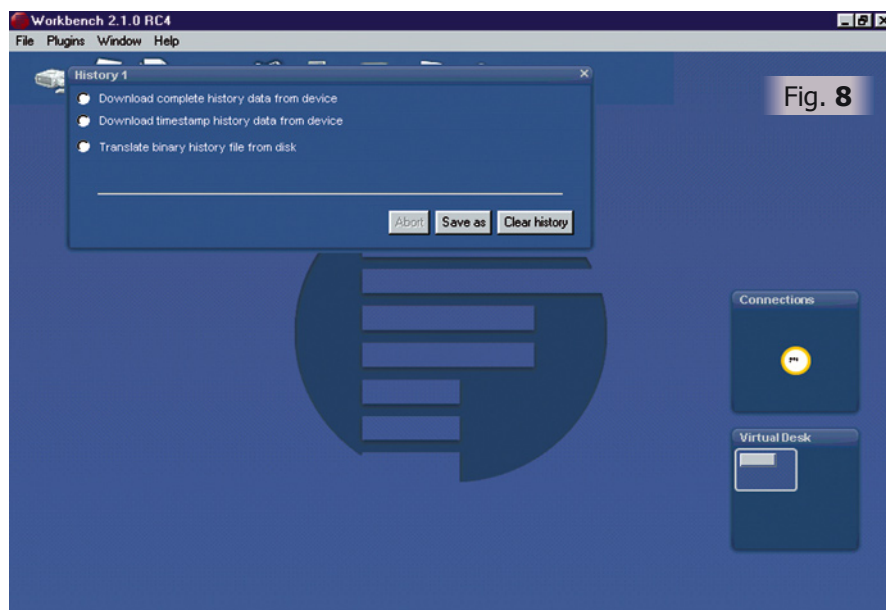
Console

Spiegato l'Editor, passiamo adesso alla funzione *Console*: si attiva facendo clic sul pulsante omonimo del desktop del Workbench e serve per dialogare con i dispositivi Mambo; sostanzialmente, è assimilabile alla finestra di un programma di Terminale, quindi da essa si possono impartire comandi che raggiungeranno direttamente il dispositivo Mambo attualmente connesso. Per comandi si intende non quelli inerenti alla configurazione del funzionamento (le istruzioni), ma quelli riguardanti la comunicazione, l'invio e la lettura di file di configurazione contenuti nella memoria dei Mambo, l'acquisizione della memoria riguardante i punti e le coordinate GPS ed altro ancora. Perché avvenga il dialogo, bisogna prima definire, nella solita sezione *Connections*, a quale Mambo si vuole assegnare la sessione di lavoro corrispondente alla *Console* aperta. Una volta assegnata la funzione, nella finestra scorrono tutti gli eventi trasmessi dalla memoria del Mambo attualmente connesso (Figura 7); è

possibile arrestare lo scorrimento con il pulsante *Stop All*.

History

Il quarto pulsante della barra degli strumenti corrisponde alla funzione *History*, grazie alla quale è possibile operare sulla memoria degli eventi (storico) del dispositivo Mambo attualmente connesso (Figura 8); come per le altre funzioni, l'assegnazione del Mambo con cui si intende lavorare si effettua dalla sezione *Connections*, facendo clic sull'icona corrispondente al dispositivo di interesse. Accedendo a *History* è possibile scaricare l'elenco cronologico completo



degli eventi registrati dal Mambo, oppure un elenco, ancora cronologico, dei soli eventi riguardanti le operazioni di comunicazione, le segnalazioni di stato (livello del segnale, satelliti e via di seguito).

La differenza tra le due operazioni sta nel fatto che una permette di acquisire tutte le situazioni riguardanti l'attività del Mambo attualmente connesso, quindi tutti i dati riguardanti il funzionamento, che possono includere il funzionamento della connessione telefonica GSM/GPRS (gestore, livello del segnale, comunicazioni ecc.) quello del ricevitore satellitare GPS (stato dei satelliti agganciati, livello del segnale, coordinate scaricate, stringhe NMEA) stato della connessione Bluetooth, mentre l'altra estrae solo parte dei dati.

In altre parole, si può definire l'intervallo temporale (da un certo giorno a un altro) da prendere in considerazione, ovvero scaricare tutto quanto è stato registrato, fermi restando i limiti determinati dalla capienza della memoria eventi del Mambo; a tale riguardo si noti che quando la memoria è piena il dispositivo sovrascrive i record iniziando da quelli più vecchi, nel senso che cancella le posizioni più datate per scrivere in esse i nuovi eventi.

Una terza opzione ci consente di

trasferire uno storico degli eventi memorizzato in un'unità a disco del computer nella memoria del Mambo attualmente in connessione, ovvero di salvare un log degli eventi su disco (pulsante *Save as*). Ancora, tramite un apposito pulsante (*Clear History*) si può impartire al Mambo l'ordine di cancellare completamente lo storico degli eventi.

Allo scopo di agevolare la consultazione dell'archivio storico degli eventi, il Workbench prevede la possibilità di applicare dei filtri di vario genere, che consentono di selezionare il tipo di dati riepilogati, ma anche di andare a cercare determinate stringhe o singoli dati specifici.

Una particolarità interessante della funzione *History* è la possibilità di scaricare dalla memoria del Mambo selezionato i dati inerenti alla localizzazione, ovvero le coordinate rilevate, in ordine cronologico; in questo caso l'utente può decidere se salvare nel computer i record corrispondenti in formato binario (formato macchina) o sotto forma di stringhe NMEA0183 intere: nel primo caso il Workbench memorizza tutti i dati contenuti in ciascuna stringa inviata dal Mambo, mentre nel secondo estrae i soli dati riguardanti le coordinate. L'utilizzo del file in formato NMEA0183 è utile quan-

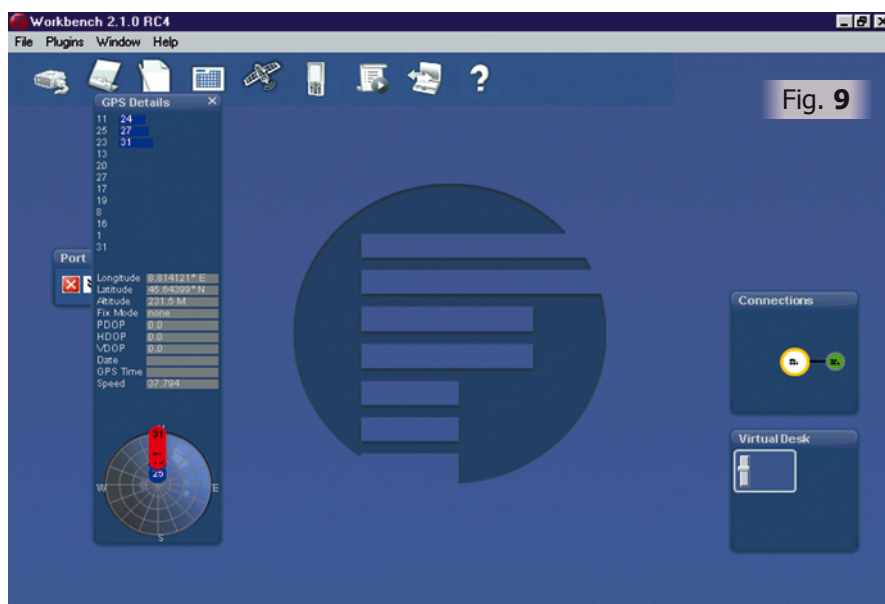


Fig. 9

do si intende inviare subito i dati a un software di navigazione e vedere sulla cartografia del caso il percorso compiuto dal localizzatore. Va comunque detto che ciò è possibile anche scegliendo di salvare il file intero della memoria, in formato binario; in tal caso si sfruttano le funzionalità di alcuni software di navigazione satellitare che permettono di estrarre le coordinate dai dati in formato macchina, ovvero di ricavare le informazioni secondo il protocollo NMEA0183 dal file binario.

Informazioni GPS

La quinta icona della barra degli

strumenti dà accesso alla funzione *GPS Detail* e all'omonima finestra di notifica (Figura 9) in cui il programma riepiloga la situazione dei satelliti dei quali il Mambo riceve il segnale; in essa si trovano informazioni sulla posizione, il livello del segnale, la modalità di funzionamento, lo stato del Fix, l'orario, l'eventuale velocità del veicolo sul quale si trova il Mambo (ammesso che sia in movimento).

Come per tutte le altre funzioni, anche questa si assegna a un determinato dispositivo Mambo selezionandolo nella sezione *Connections*, nel modo consueto, già spiegato in precedenza.

Informazioni sul cellulare

Un comando (*GSM Detail*) analogo al precedente, ma riservato alla sezione GSM/GPRS del Mambo, si impartisce facendo clic sulla sesta icona della barra degli strumenti; permette di accedere a una finestra (Figura 10) che riepiloga le condizioni dell'eventuale connessione telefonica radiomobile del dispositivo selezionato mediante la sezione *Connections*.

La finestra cui si accede contiene le informazioni sul livello del segnale, il nome del gestore (o il suo codice identificativo) della rete cui il di-

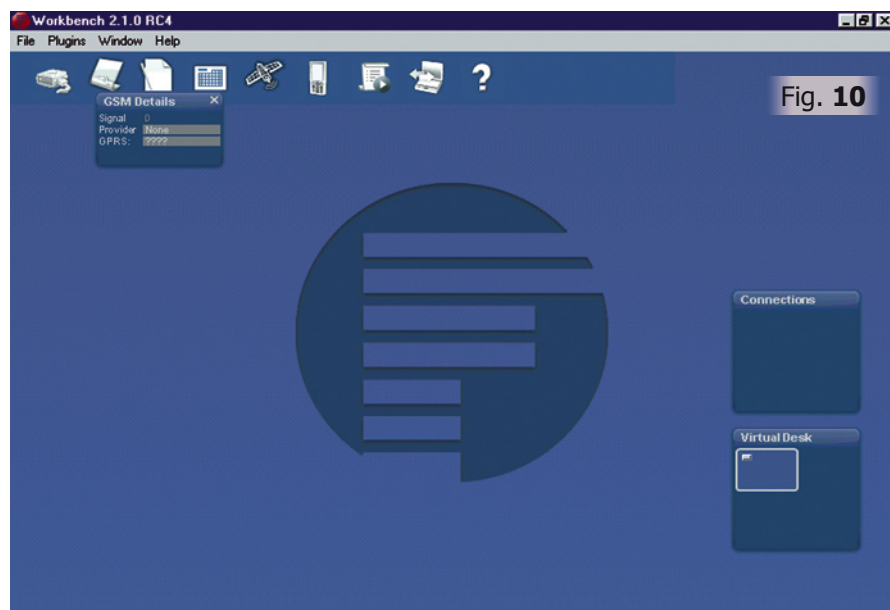


Fig. 10

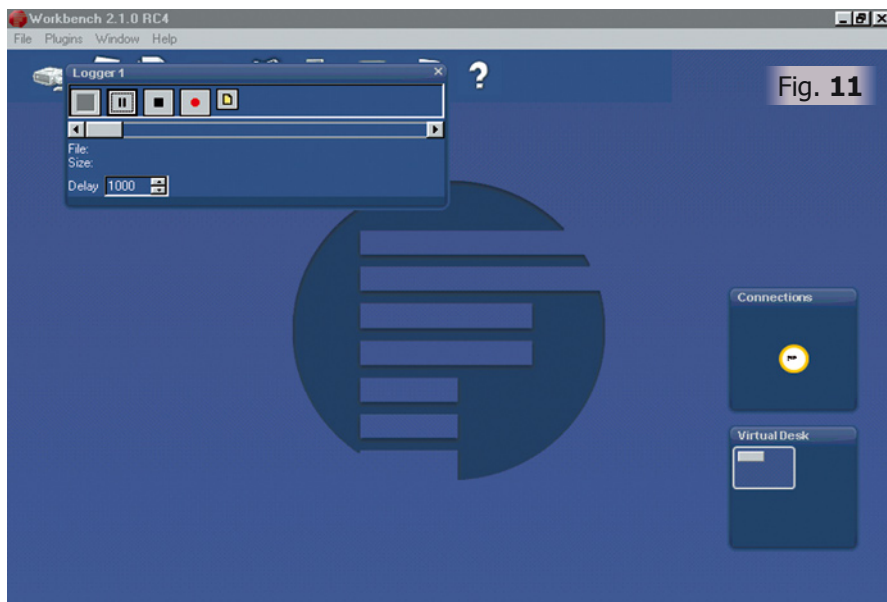


Fig. 11

positivo è agganciato e l'eventuale presenza dell'attività GPRS.

Il Logger

La settima icona della barra degli strumenti corrisponde al comando *Logger* e avvia una sorta di registratore di eventi, i cui parametri di funzionamento possono essere definiti nell'apposita finestra di dialogo cui si accede (Figura 11); il registratore crea un file che memorizza tutto quel che accade durante la connessione con il Mambo abbinato mediante la solita procedura che passa dalla sezione *Connections*.

Informazioni I/O

L'ultima funzione del Workbench che analizziamo è quella chiamata *I/O Detail* alla quale si accede facendo clic sull'ottava icona della barra degli strumenti, ovvero impartendo l'omonimo comando del menu *Plugins*. Dà accesso a una finestra di notifica (Figura 12) che riepiloga le condizioni dei comandi locali e della batteria del Mambo (attualmente connesso mediante la solita sezione *Connections*).

Con questo termina la descrizione delle funzioni del programma Falcom; chi volesse maggiori dettagli, potrà trovarli nell'*Help* in li-

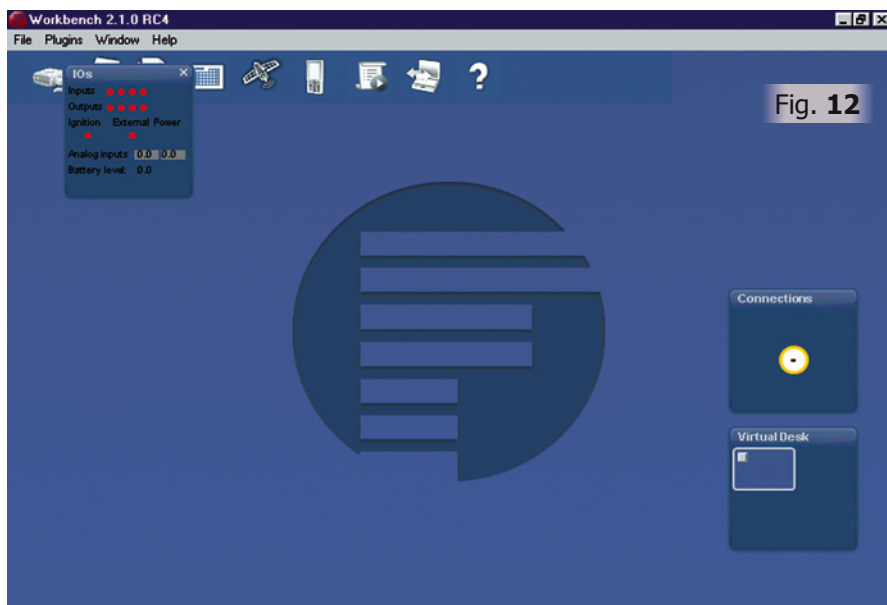


Fig. 12

nea accessibile facendo clic sul pulsante più a destra della barra degli strumenti (quello contraddistinto dal punto interrogativo) che costituisce un valido aiuto all'operatore che, durante una sessione di lavoro, avesse qualche dubbio sull'utilizzo di una certa funzione o sulla definizione di un parametro.

Il file di configurazione

Bene, esaminato il programma Workbench e le sue principali funzioni, possiamo spostarci fisicamente in Mambo e andare a vedere come è strutturato il file di configurazione, cioè quell'insieme di comandi che dice al dispositivo come deve comportarsi.

Partiamo dicendo che inizialmente, ossia all'uscita dalla fabbrica, nella memoria viene scritto un file come quello visibile nel Listato 1; esso contiene le istruzioni di base, che servono a gestire i pulsanti al fine di accendere e spegnere Mambo, instaurare la comunicazione con il computer, gestire le chiamate telefoniche e dare le segnalazioni del caso. Proviamo ad esaminare alcuni punti del listato e vediamo che all'accensione il software inizializza due timer ciclici, uno da 1 secondo e l'altro da 20, oltre ad un contatore da 7 secondi; configura il GPS perché sia acceso con un intervallo on/off di 400/1.000 ms.

I timer servono per valutare e segnalare l'attività del GPS, oltre che per il controllo di altre funzioni quali la connessione telefonica, lo stato di carica della batteria e l'accensione dei led che danno le segnalazioni visive del caso.

I pulsanti servono, oltre che per accendere e spegnere Mambo, alla gestione delle chiamate telefoniche, che il dispositivo, per impostazione predefinita, può sia effettuare che ricevere; in particolare, la pressione breve di uno qualsiasi di essi determina la sospensione di una telefonata in corso, sia essa ricevuta o

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 inizializza un timer (TIMER_1SEC) ciclico della durata di un secondo, un timer dello stesso tipo della durata di 20 sec (TIMER_20SEC), un contatore a 7 e configura il GPS in modo che stia acceso 400 ms su 1000 ms.

$PFAL,CNF.Set,AL1=SYS.Device.eStart:SYS.COUNTER_SLEEP_MODE.set=0&SYS.COUNTER_BAT_MODE.set=0
Inizializza il contatore COUNTER_SLEEP_MODE a 0 e il contatore COUNTER_BAT_MODE a 0

$PFAL,CNF.Set,AL2=SYS.TIMER.e_1SEC:SYS.COUNTER_TIMING_MODE.decrement=1
Se il TIMER_1SEC va a 0 (dopo quindi un secondo) allora (:) decrementa il contatore COUNTER_TIMING_MODE di 1

$PFAL,CNF.Set,AL3=SYS.COUNTER.e_TIMING_MODE:SYS.COUNTER_TIMING_MODE.set=7
Se il contatore COUNTER_TIMING_MODE arriva a 0 resettalo a 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 ms) allora inizializza il TIMER_DOUBLE ad un valore di 100 ms e emetti un beep

$PFAL,CNF.Set,AL5=SYS.TIMER.e_DOUBLE:IO.Beep2=hpulse,100
Se il TIMER_DOUBLE va a 0 (dopo quindi 100ms) allora emetti un beep

... descrizione più approfondita nella prossima puntata.
```

effettuata dall'apparato. Inoltre, se è in arrivo una chiamata, premendo brevemente uno dei tasti si risponde ad essa. Riguardo allo spegnimento, va detto che Mambo, se il comando viene dato quando la sezione GPS è in fix, prima di spegnersi va a salvare nella propria memoria l'ultimo punto valido. Del file di configurazione va detto che è composto da due porzioni, una modificabile dall'utente accedendo dalle apposite funzioni del Workbench e l'altra contenente parametri che non possono essere modificati: la prima è detta *User Configuration* e la seconda *Device Configuration*. Quindi si può andare a sovrascrivere la sola *User Configuration*. Restando in tema di programmazione di Mambo, in questa puntata vogliamo farvi vedere un altro esempio di configurazione mirata a determinate funzioni; per esempio, il Listato 2 è uno *User Configuration* che comprende i comandi da inviare per definire la modalità di funzionamento dei led bicolore e la sensibilità della sezione GPS. Tenete presente che nell'esempio in questione, avente scopo prettamente didattico, sono stati rimossi alcuni parametri predefiniti nello *User Configuration* inserito in fabbrica nella memoria di Mambo.

Nello specifico, con i comandi pre-

senti nel listato si stabiliscono le temporizzazioni inerenti all'invio delle stringhe GGA, GLL, GSA, GSM, GSV, RMC e GTV e la modalità automatica per i led; si definisce altresì il nome con cui il dispositivo deve identificarsi, cosa necessaria quando ci si trova a dover ricevere da una singola postazione i dati di più Mambo collocati su diversi veicoli o indossati da più persone. Per

quel che riguarda la sezione GPS, vengono disattivate la correzione e il modo Power-Save (quindi il timer 400/1.000 ms. che per impostazione predefinita ne controlla attivazione e disattivazione) perciò il ricevitore è "sempre acceso".

Nella prossima puntata analizzeremo in dettaglio i file di configurazione, quindi non perdetevi il prossimo fascicolo.

Listato 2

```
$PFAL,CNF.Set,PROT.GGA=5
Imposta l'invio della stringa GGA ogni 5 secondi
$PFAL,CNF.Set,PROT.GLL=5
Imposta l'invio della stringa GLL ogni 5 secondi
$PFAL,CNF.Set,PROT.GSA=1
Imposta l'invio della stringa GSA ogni secondo
$PFAL,CNF.Set,PROT.GSM=10
Imposta l'invio della stringa GSM ogni 10 secondi
$PFAL,CNF.Set,PROT.GSV=1
Imposta l'invio della stringa GSV ogni secondo
$PFAL,CNF.Set,PROT.RMC=1
Imposta l'invio della stringa RMC ogni secondo
$PFAL,CNF.Set,PROT.VTG=5
Imposta l'invio della stringa VTG ogni 5 secondi
$PFAL,Cnf.Set,IO.LED.BASE=11bma11g1
Imposta i led per funzionare in modo automatico seguendo l'impostazione delle periferiche
$PFAL,CNF.Set,DEVICE.NAME=Elettronica In Mambo
Imposta il nome del Mambo
$PFAL,CNF.Set,DEVICE.GPS.AUTOCORRECT=off
Disabilita la correzione del GPS
$PFAL,CNF.Set,DEVICE.BAT.ENABLE=on,3,3
Abilita il monitoraggio della tensione di ricarica e se supera i 3,7V viene richiamato l'evento Sys.Power.ePlugged
$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=off,900,1000
Disabilita il PowerSave del GPS
$PFAL,CNF.Clear,"AL9"
...
$PFAL,CNF.Clear,"AL21"
Cancella le impostazioni relative all'accensione dei led per l'indicazione dello stato della batteria
```

Vasta gamma di videoregistratori digitali per qualsiasi esigenza, dalla casa al punto vendita, alla piccola o grande azienda.
Da 4 a 16 canali, differenti sistemi di compressione, con interfaccia LAN e video Web Server, con trasferimento dati USB o back-up su DVD: scegli il modello che meglio si adatta alle tue esigenze.

PREZZO IMBATTIBILE
€ 259,00



DVR, la tecnologia digitale per la tua sicurezza

FR329W - DVR / MULTIPLEXER A 4 CANALI MPEG-4/JPEG CON VIDEO WEB SERVER

Videoregistratore digitale real-time a quattro canali con frame rate di 100 IPS e interfaccia Ethernet (10-100 Base-T). Formato di compressione MPEG-4 (CIF), MJPEG (Frame), funzione Motion Detection avanzata per tutti i canali con sensibilità regolabile, possibilità di registrare, riprodurre e visualizzare contemporaneamente le immagini. Visualizzazione real-time da remoto tramite connessione Internet (massimo 5 utenti) con possibilità di effettuare registrazioni e back-up. Grazie all'elevato livello di compressione il DVR è in grado di registrare per 170 giorni consecutivamente (HDD da 500 GB, 4CH, CIF

alta qualità, 15 IPS). Standard video NTSC e PAL • Livello ingressi video 1 Vpp 75 ohm (BNC) • 4 Uscite loop, un'uscita monitor principale e una secondaria, livello 1 Vpp 75 ohm (BNC) • Regolazione qualità immagine • Hard Disk supportato IDE, ATA 66, capacità oltre 400 GB • Refresh video 120 IPS (NTSC) / 100 IPS (PAL) • un ingresso e un'uscita audio (RCA) • Registrazione pre-allarme • Compressione Motion JPEG per trasmissione via Web • Interfaccia Web compatibile IE browser e software AP

in licenza • Invio delle immagini in caso di allarme tramite Email o via FTP • Protocolli supportati TCP/IP, PPPoE, DHCP, DDNS • 4 ingressi e un'uscita d'allarme • Zoom 2 X digitale (solo in modalità real-time) • Alimentazione 19 VDC tramite adattatore di rete 100 ~ 240 Vac (incluso) • Assorbimento < 42 W • Dimensioni 343x223x59 mm • Sistema di ripristino di tutte le funzioni dopo un black-out.



€ 398,00

DVR 4 canali MJPEG con Video Web Server, USB e cassetto estraibile

FR261



€ 610,00

DVR 4 canali MPEG4, Video Web Server, CD-RW e USB

FR337



DVR 8 canali MPEG4, GPRS, Video Web Server, USB, CD-RW e connessione per RAID

€ 1.250,00

FR334



€ 1.750,00

DVR 16 canali MPEG4, GPRS, Video Web Server, CD-RW, USB e connessione per RAID

FR335

I DVR vengono forniti senza Hard Disk.



€ 560,00

DVR 9 canali WAVELET con cassetto estraibile

FR319



DVR 9 canali WAVELET con Video Web Server e cassetto estraibile

€ 670,00

FR319W



€ 1.450,00

DVR 16 canali MPEG4, GPRS, USB, Video Web Server e connessione per RAID

FR322



DVR 16 canali MPEG4, GPRS, USB, Video Web Server, DVD-RW e connessione per RAID

€ 1.670,00

FR322D



€ 280,00

DISK ARRAY BOX per 3 HARD DISK con funzione HUB

FR336

CP343 - SCHEDA USB a 1 CANALE per ACQUISIZIONE A/V

Eccezionale dispositivo dotato di porta USB che consente di trasformare il PC in un vero e proprio DVR. Dispone d'ingresso video RCA e S-VIDEO, ingresso audio stereo e di interfaccia USB 2.0 per collegamento al PC. Completo di software per la gestione delle immagini. Formati video supportati 176x144 (min.) - 1440x1152 (max.); elevata qualità delle immagini; regolazione luminosità, contrasto, tonalità, saturazione. Requisiti minimi di sistema: processore 1,8 GHz



€ 43,00

o superiore; 256 MB di RAM; Hard Disk con almeno 10 GB di spazio libero; Sistema Operativo Win2000, WinXP; porta USB libera; scheda grafica a 64 bit.

DVR4USB - SCHEDA USB a 4 CANALI per ACQUISIZIONE A/V

Trasforma il tuo PC in un vero e proprio DVR con l'ausilio di questa scheda da collegare alla porta USB! Quattro ingressi video e due audio, possibilità di visualizzare le immagini riprese in modalità QUAD o a pieno schermo anche da remoto via LAN o via Internet. Un potente software consente di ottenere tutte le funzioni tipiche di un DVR.



€ 115,00

ACCESSORI per DVR:

- BOXHDD - Cassetto estraibile per DVR FR319 e FR233. € 22,50
- DVRCARTR - Cassetto estraibile per DVR FR190 e FR233. € 32,00

Via Adige, 11 - 21013 Gallarate (VA) •
Tel. 0331/799775 - Fax. 0331/778112

Disponibili presso i nostri Rivenditori o nel nostro punto vendita di Gallarate (VA).
Caratteristiche tecniche e vendita on-line direttamente sul sito www.futurashop.it

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!

DISPOSITIVI GSM PER CONTROLLI REMOTI

Per controllare, attivare e verificare in modalità remota sfruttando le reti GSM che coprono capillarmente tutto il territorio nazionale. Tutti i dispositivi vengono forniti montati e collaudati e sono certificati CE-R&TTE.



TELECONTROLLO GSM CON COMANDI DTMF

Sistema di controllo remoto bidirezionale che sfrutta la rete GSM per le attivazioni ed i controlli. Configurabile con una semplice telefonata, dispone di due uscite a relé (230Vac/10A) con funzionamento monostabile o bistabile e di due ingressi di allarme optoisolati. Possibilità di memorizzare 8 numeri per l'invio degli allarmi. Le operazioni di impostazione e configurazione avvengono tramite SMS; la gestione delle uscite ed il controllo dello stato degli ingressi, oltre che con gli SMS, possono essere effettuate con comandi DTMF, inviati tramite una chiamata voce protetta che prevede il riconoscimento dell'identificativo del chiamante e/o l'inserimento di una password. Alimentazione compresa tra 5 e 32 Vdc, assorbimento massimo 500mA. Antenna GSM bibanda integrata. Il prodotto viene fornito già montato e collaudato.

CARATTERISTICHE TECNICHE: modulo GSM tipo GM862 QUAD Telit; potenza di uscita Classe 4 (2 W @ 850-900 MHz) / Classe 1 (1 W @ 1800-1900 MHz); temperatura di funzionamento -10°C + 55°C; peso 100 grammi circa; dimensioni 98 x 60 x 24 (L x W x H) mm; alimentazione 5 + 32 Vdc; corrente assorbita 20 mA a riposo, 500 mA in trasmissione; corrente massima contatti relé di uscita 10 A; tensione massima contatti relé di uscita 250 Vac; caratteristiche ingressi digitali livello 1 - 5-32 Vdc, livello 0 - 0 Vdc.



TERMOSTATO CON CONTROLLO GSM

Permette di gestire da remoto, tramite rete GSM, il sistema di riscaldamento di abitazioni, uffici ecc... Funziona in parallelo al termostato esistente e consente di stabilire la temperatura desiderata. Può inoltre avvisare, tramite SMS o mediante chiamate, quando la temperatura rilevata si discosta dalla finestra impostata o l'ingresso di allarme è attivo.

CARATTERISTICHE TECNICHE: escursione di temperatura da -55° a +99°C; 2 ingressi a livello logico: uno di allarme optoisolato (allarme caldaia in blocco), uno controllato dal contatto N.O. del termostato esterno; 1 uscita a relé, gestibile in modo automatico, manuale o asservita a termostato esterno; gestione funzionalmetrante SMS, pulsante su scheda o PC dotato di apposita interfaccia RS232 (FT475) o USB (FT635) e software dedicato (scaricabile gratuitamente dal sito www.elettrocin.it). Alimentazione 5 + 32 Vdc, assorbimento 300 mA nominali; dimensioni 98 x 60 x 24 mm.



TELECONTROLLO GSM BIDIREZIONALE CON ANTENNA INTEGRATA

Sistema di controllo remoto bidirezionale che sfrutta la rete GSM per le attivazioni ed i controlli. Configurabile con una semplice telefonata, dispone di due uscite a relé (230Vac/10A) con funzionamento monostabile o bistabile e di due ingressi di allarme optoisolati. Possibilità di memorizzare 8 numeri per l'invio degli allarmi e 200 numeri per la funzionalità apricancello. Tutte le impostazioni avvengono tramite SMS.

CARATTERISTICHE TECNICHE: Alimentazione: 5-32 Vdc • Assorbimento massimo: 500mA • Antenna GSM bibanda integrata • GSM: Dual Band EGSM 900/1800 MHz (compatibile con ETSI GSM Phase 2-Standard) • Dimensioni: 98 x 60 x 24 (L x W x H) mm. Il prodotto viene fornito già montato e collaudato.



APRICANCELLO GSM CON ANTENNA INTEGRATA

Dispositivo di controllo GSM da utilizzare in abbinamento ai sistemi di apertura dei cancelli elettrici. Il funzionamento è molto semplice: il cancello può essere azionato effettuando una chiamata con il proprio cellulare al numero della SIM Card inserita nell'unità GSM. La chiamata non avrà mai risposta (in questo modo non si consuma neppure uno scatto) ma il dispositivo invierà un comando alla centralina di controllo del cancello che provvederà ad aprirlo o chiuderlo. Gestione degli utenti da remoto mediante SMS (è necessario conoscere la password) oppure in locale tramite PC con apposito software di configurazione. Alimentazione 12-24 Vdc selezionabile mediante jumper. Antenna integrata su CS.



MODEM GSM CON INTERFACCIA USB

Compatto modem utilizzabile in tutte le applicazioni nelle quali si ha la necessità di effettuare trasmissioni dati sfruttando la rete mobile GSM. È dotato di porta USB che ne permette l'interfacciamento a qualsiasi PC o Notebook provvisto di tale periferica. L'alimentazione del dispositivo è fornita direttamente dalla connessione USB.

CARATTERISTICHE TECNICHE: Modulo GSM/GPRS Telit; Frequenze 900/1800 MHz • Potenza RF: 2W (900 MHz) • 1W (1800 MHz) • Alimentazione: 5V (tramite porta USB) • Assorbimento a riposo: 30 mA • Assorbimento in connessione: 250mA • Interfaccia dati: USB1.1 e USB2.0 • Antenna: bibanda, integrata su CS.



TELECONTROLLO GSM BIDIREZIONALE CON ANTI-JAMMER

Sistema di teleallarme e controllo remoto GSM dotato di funzione anti-Jammer, da abbinare a qualsiasi impianto antifurto. Il dispositivo può funzionare sia in modalità tradizionale (simile al TDG33) che in modalità anti-Jammer (in questo caso è necessario utilizzare due schede). Il sistema garantisce la massima sicurezza verificando periodicamente il corretto funzionamento dell'impianto radio: in caso di manomissione o disturbo interviene la seconda unità GSM che invia i messaggi di allarme.



TELECONTROLLO GSM BIDIREZIONALE

Unità di controllo GSM con due ingressi fotoaccoppiati e due uscite a relé. Utilizzabile sia per attivare a distanza qualsiasi apparecchiatura che per ricevere messaggi di allarme. In modalità apricancello è in grado di memorizzare fino ad un massimo di 100 utenti. Ideale per realizzare impianti antifurto per abitazioni e attività commerciali, car alarm, controlli di riscaldamento/condizionamento, attivazioni di pompe e sistemi di irrigazione, apertura cancelli, controllo varchi, circuiti di reset, ecc. Fornito montato e collaudato.

CARATTERISTICHE TECNICHE: Frequenza di lavoro: GSM bibanda 900/1.800MHz • Funzione apricancello a costo zero • 2 ingressi optoisolati • 2 uscite a relé (bistabile o astabile) • 5 numeri abbinabili per allarme • 100 numeri abbinabili per apricancello • Carico applicabile alle uscite: 230V, 5A • Alimentazione: 5-32V • Assorbimento massimo: 550mA.

COMBINATORE TELEFONICO GSM CON AUDIO

Compatto combinatore GSM da abbinare a qualsiasi impianto antifurto domestico. Dispone di due canali con messaggi vocali con 8 numeri per canale. Possibilità di invio chiamate vocali o messaggi SMS. Completo di contenitore plastico e antenna integrata su circuito stampato. Alimentazione 12 Vdc.

CARATTERISTICHE TECNICHE: Combinatore telefonico GSM a due canali • 2 messaggi vocali da 10 secondi • 5 cicli di chiamata per canale • 2 Ripetizioni del messaggio • Invio messaggio vocale o SMS • Segnalazione di campo di presenza GSM • Blocco allarme da remoto • Programmazione dei numeri su SIM • Riconoscimento chiamata a buon fine.



TELECONTROLLO GSM ESPANDIBILE

Permette, tramite SMS, di comandare fino a 72 utilizzatori e di leggere altrettanti ingressi digitali, oltre a 2 ingressi analogici (0-3,6V). Può funzionare come teleallarme e inviare, a un massimo di 8 numeri telefonici, un SMS, una telefonata o entrambe le segnalazioni, quando gli ingressi rilevano la condizione di allarme. L'accesso al sistema è protetto da password per impedire l'utilizzo da parte di estranei.

CARATTERISTICHE TECNICHE: Modulo GSM: GM862 QUAD Telit; GSM: Quad band EGSM 850-900/1800-1900 MHz compatibile con ETSI GSM Phase 2-Standard • Potenza di uscita: - Class 4 (2 W @ 850-900 MHz); - Class 1 (1 W @ 1800-1900 MHz) • 8 uscite digitali on-board espandibili a 72 mediante interfaccia FT473 • 8 ingressi digitali on-board espandibili a 72 mediante interfaccia FT488 • Numero massimo espansioni: 8 OUT/IN; SMS di allarme • Alimentazione: 5-32Vdc, 300mA.

Tutti i prezzi sono da intendere IVA inclusa.

FUTURA ELETTRONICA

idee in elettronica

Via Adige, 11
21013 GALLARATE (VA)
Tel. 0331/799775
Fax 0331/778112
www.futurashop.it

Disponibili presso i migliori negozi di elettronica o nel nostro punto vendita di Gallarate (VA). Caratteristiche tecniche e vendita on-line: www.futurashop.it

Mambo, la prima applicazione pratica

di Alessandro Orlando



Ecco il primo esempio di programmazione ed utilizzo: un semplice AVL (Automotive Vehicle Locator) in grado di informare via SMS, ad intervalli prestabiliti, la posizione del veicolo, se questo è fermo oppure in movimento, l'eventuale velocità e se manca il segnale GPS.

Nella precedente puntata abbiamo familiarizzato con la sintassi dei comandi utilizzati dal sistema di localizzazione MAMBO per eseguire specifiche operazioni: durante la spiegazione dei vari comandi, abbiamo anche visto qualche semplice esempio.

In questa puntata entriamo più nel dettaglio, realizzando ed analizzando un eseguibile scritto sulla base di precise specifiche: in altre parole, definiamo le specifiche, scriviamo il programma di configurazione per il Mambo e commentiamo dettagliatamente ogni singola istruzione. Per partire con il piede

giusto, la prima cosa da fare, dopo avere letto il manuale del Mambo in tutta le sue parti ed aver acquisito un minimo di confidenza con i vari esempi di script di cui il manuale stesso è pieno, è quella di stabilire con precisione tanto la modalità di funzionamento del dispositivo quanto come questi si debba comportare in determinate situazioni quali, ad esempio, l'assenza della copertura GSM, la mancanza del segnale GPS, la batteria quasi scarica, le connessioni da utilizzare in funzione dei comandi ricevuti sia in locale che da remoto, eccetera. Più semplicemente, occorre definire "le specifiche di >

progetto” che debbono prendere in considerazione tutte le possibili situazioni che si possono presentare in quella applicazione.

Dopo avere stabilito come il sistema deve funzionare (ossia cosa ci si aspetta dal Mambo durante il suo funzionamento) si deve disegnare, esattamente come si fa, ad esempio, per gli eseguibili scritti per il PC o per i microcontrollori, un diagramma di flusso che sintetizza le istruzioni del programma e determina per l'appunto le azioni da compiere in base alle nostre specifiche ed agli eventi.

A questo punto riprendiamo l'esempio della precedente puntata la cui dichiarazione (listato) - visibile in tabella 1 - svolge la funzione di registro di posizione: in caso di se-

(. **GPS.History.Write,0,""**) quando il Timer ha finito di contare (**Timer.e0**), ovvero ogni 10 secondi.

Il diagramma di flusso del programma appena descritto è visibile in figura 1: anche se abbiamo avuto modo più volte di parlarne, ripetiamo che il listato eseguito dal Mambo viene scritto nel file di configurazione rispettando la sintassi dei comandi; una volta caricato il file di configurazione, l'interprete del localizzatore si occupa di tradurre le istruzioni in azioni. Dal semplice esempio appena esposto, appare evidente che la struttura dell'eseguibile (leggasi “del file di configurazione”) memorizzato all'interno del Mambo deve necessariamente rispondere a delle precise stringhe, la cui sequenza logica tiene conto di tutte le possibili

condizioni di allarme (ricordate che tutti gli eventi controllati dal Mambo sono

gestiti come allarmi) al fine di non incorrere nel blocco improvviso del programma in cicli di loop infiniti, che attendono il verificarsi di un allarme che non si può di fatto verificare in quanto non previsto. Notate che quanto appena esposto rappresenta il punto di partenza per la stesura dell'esempio di

programma vero e proprio, oggetto di questo articolo.

Il programma completo

Il miglior esempio applicativo è, normalmente, quello con il maggior senso logico, quindi abbiamo ipotizzato di voler controllare la posizione di un veicolo ad intervalli regolari e di visualizzarla su un programma cartografico, come ad esempio il Fugawi o altri.

Poiché la stringa NMEA0183 RMC è quella normalmente letta sia dai software cartografici che dai siti che permettono di visualizzare un target all'interno di una mappa (www.mapquest.com), ai fini dell'esempio pratico è necessario capirne la struttura e la composizione in maniera tale da poter estrapolare i dati necessari al programma. La tabella 2 riassume tutti i campi dati della sentenza RMC: essa contiene tutte le informazioni che definiscono la navigazione del sistema, ossia latitudine e longitudine, data e ora, direzione e velocità, validità della posizione calcolata, più altri parametri quali elevazione, variazione magnetica e checksum. Si tratta di una serie di informazioni molto completa anche se, nel nostro caso, i dati utilizzati sono la data, l'ora, la velocità, la latitudine e la longitudine. Di tutti questi dati, quelli da utilizzare per la visualizzazione della posizione

Tabella 1 - Log di posizioni valide

```
$PFAL,Cnf.Set,AL0=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,""
```

gnale GPS valido viene avviato un contatore da 10 secondi allo scadere del quale la posizione rilevata viene scritta in memoria. A questo punto il programma esegue un nuovo ciclo e così via, all'infinito. Se il segnale GPS non è valido, il timer non viene avviato e di conseguenza la posizione non viene memorizzata.

Con la prima dichiarazione (**AL0**) si comunica al localizzatore che esso deve avviare un timer ciclico della durata di 10 sec. (**Timer0.Start=ciclic,10000**), nel caso in cui il segnale sia valido (**GPS.Nav.eFix=valid**).

Con la seconda (**AL1**) viene forzato lo spegnimento del timer ciclico a 10 secondi nel caso in cui il segnale non sia valido (**GPS.Nav.eFix=invalid:Sys.Timer0.stop**).

Con la terza ed ultima dichiarazione, invece, il localizzatore viene forzato a salvare la posizione GPS nello storico della memoria

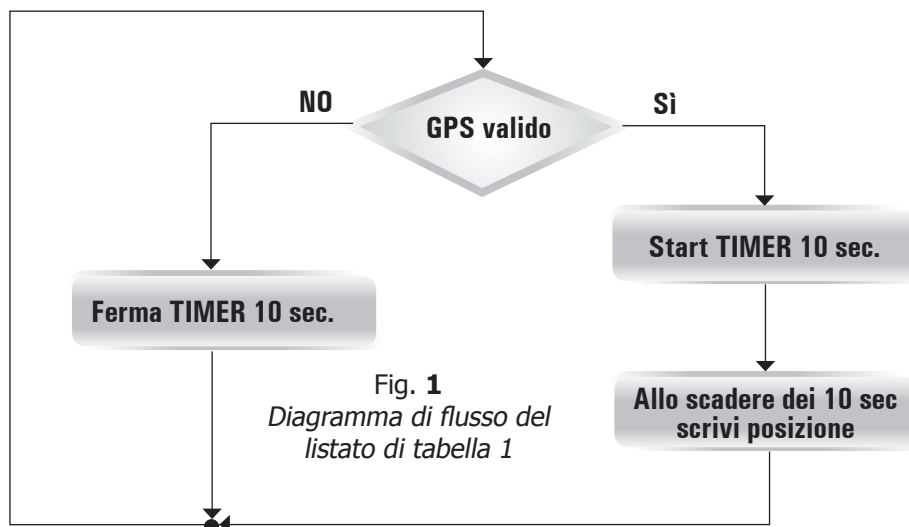


Fig. 1
Diagramma di flusso del
listato di tabella 1

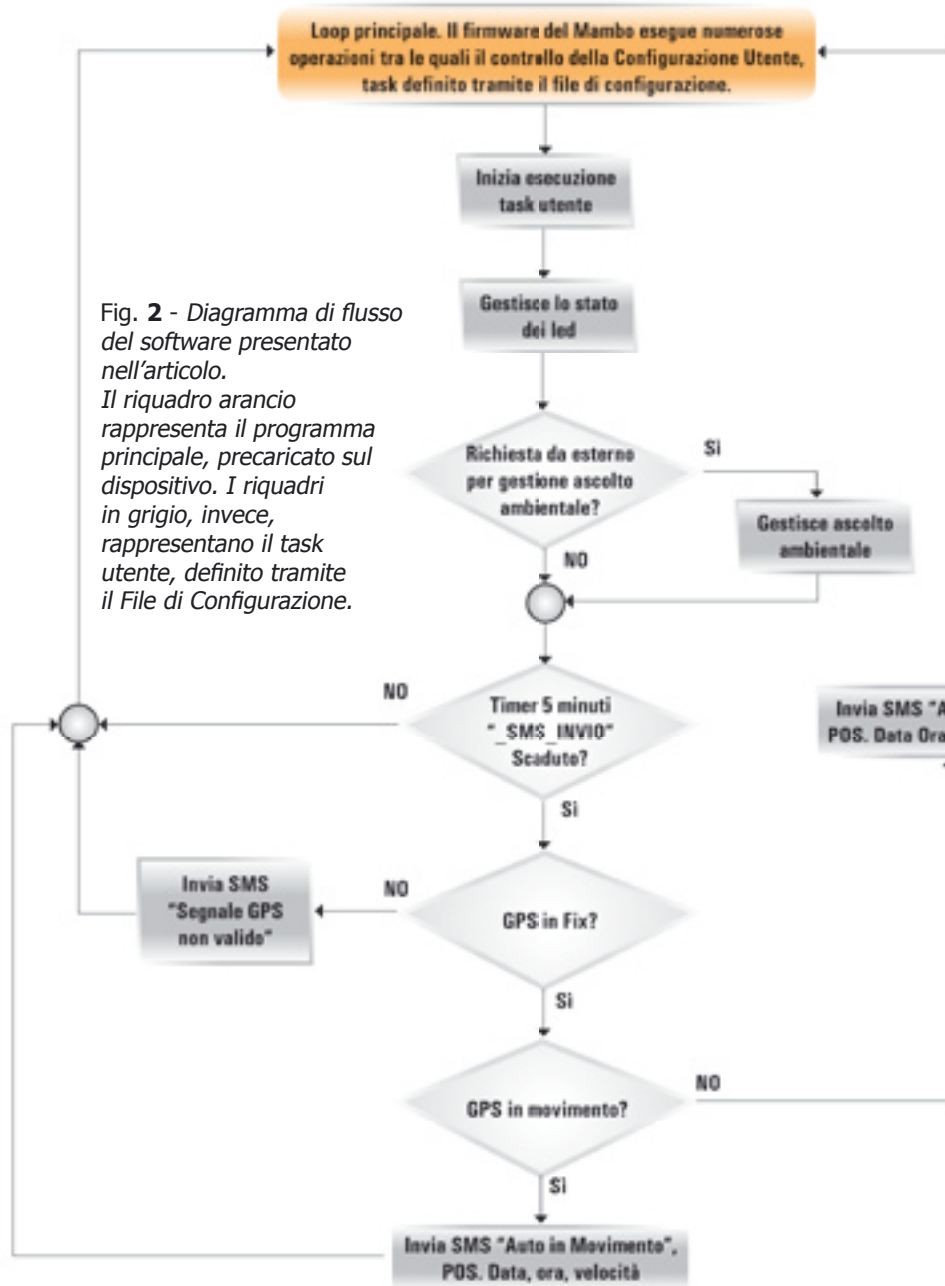


Fig. 2 - Diagramma di flusso del software presentato nell'articolo.

Il riquadro arancione rappresenta il programma principale, precaricato sul dispositivo. I riquadri in grigio, invece, rappresentano il task utente, definito tramite il File di Configurazione.

loop principale del programma ossia quella parte (la maggior parte!) di software che controlla e gestisce lo stato ed il funzionamento di tutte le periferiche (GPS, GSM, LED, pulsanti, interfaccia Bluetooth, codec audio, microfono, altoparlante ecc.) di cui Mambo dispone. Tra le tante cose, il microprocessore dedica un certo tempo alla gestione della Configurazione Utente, legata al file di configurazione ed argomento principale di questo articolo. A tempo debito, il microprocessore avvia il "task utente (file di configurazione)" in cui

la prima operazione effettuata è quella relativa allo stato dei led. Immediatamente dopo, il localizzatore verifica se è in corso una chiamata per l'attivazione ambientale: in caso affermativo, essa viene gestita. In ogni caso, il programma prosegue verificando se il timer che scandisce l'invio degli SMS (`_SMS_invio`, del valore di 5 minuti), è scaduto. Se non lo è, il task utente termina ed il programma rilascia il controllo al loop principale. Se lo è, invece, il programma controlla lo stato del segnale GPS, che può essere valido oppure no. Nel caso di un segnale non valido, il programma invia un SMS contenente tale

informazione. In presenza di segnale valido, invece, il programma esegue un ulteriore controllo, ossia verifica la velocità del localizzatore (o del veicolo in cui esso è installato). In funzione della velocità, il programma decide di informare l'utente remoto del fatto che il veicolo è in sosta oppure in movimento. In ogni caso l'invio di un SMS determina la fine dell'esecuzione del task utente ed il ripristino del controllo da parte del loop principale.

Bene: partendo dalle specifiche di progetto, abbiamo ricavato il dia-

con MapQuest o Fugawi sono solo la latitudine e la longitudine. Vista la sentenza NMEA utilizzata per il programma, stabiliamo ora le specifiche del progetto. Si vuole che il sistema svolga le seguenti funzioni:

1. Invio di un SMS ogni 5 minuti, nel caso di segnale GPS valido; l'SMS contiene data, ora, latitudine, longitudine, velocità.
2. In caso di segnale GPS non valido, invio di un SMS con l'indicazione di segnale non valido.
3. Nel caso in cui il Mambo non sia in movimento, invio di un SMS

contenente data, ora, latitudine, longitudine; la velocità, in questo caso, è inutile.

4. Il Mambo non deve emettere alcun tipo di tono, ma deve segnalare lo stato di funzionamento sia del GPS che del GSM.
5. In caso di chiamata dall'esterno deve rispondere e porsi in modalità di ascolto ambientale.

Avendo definito le specifiche di progetto, a questo punto è possibile realizzare il diagramma di flusso del software, visibile in figura 2. Il primo riquadro in alto rappresenta il

Tabella 2 - Descrizione dei campi della sentenza RMC

Campo	Formato	Esempio	Descrizione
1	\$GPRMC	\$GPRMC	Stringa iniziale
2	hhmmss.ss (Numerico)	133725.569	Ora UTC
3	a (Carattere)	Posizione valida	(A: valida, V: non valida)
4	ddmm.mmmm (Numerico)	5040.4365	Latitudine
5	a (Carattere)	N	Direzione Latitudine (N/S)
6	dddmm.mmmm (Numerico)	01058.5650	Longitudine
7	a (Carattere)	E	Direzione Longitudine (W/E)
8	x.x (Numerico)	0.05	Velocità (Nodi)
9	x.x (Numerico)	302.98	Heading (Gradi)
10	ddmmyy (Numerico)	251004	Data
11	x.x (Numerico)	Non utilizzato	Variazione Magnetica (Gradi)
12	a (Carattere)	Non utilizzato	Variazione direzione magnetica (W/E)
13	*hh (Esadecimale)	*00	Checksum
14	< CR > < LF >		Stringa finale del messaggio

gramma di flusso necessario allo scopo; vediamo ora di scrivere il programma.

Partendo dalla configurazione di base, precaricata nel dispositivo e ricavabile dopo il comando *\$PFAL,Sys.Device.FactoryReset*, abbiamo introdotto delle sostanziali modifiche, fino ad ottenere il programma di configurazione elencato nella tabella 3.

Innanzitutto è stato definito un contatore denominato *_SMS_invio*, il

quale viene decrementato di una unità ogni volta che il *Timer.e_1_sec* va a zero. Una volta raggiunto il valore "0", il contatore *_SMS_invio* viene riprogrammato a 300. Serve per definire l'intervallo di 5 minuti per l'invio degli SMS (per l'appunto 300/60 secondi equivale a 5 minuti). Abbiamo detto che il contatore *_SMS_invio* deve essere riprogrammato a 300 dopo che ha raggiunto il valore 0: questa operazione è ottenuta con la linea di comando de-

scritta in tabella 3 alla linea 1. Il decremento di una unità a cadenza di 1 secondo, invece, viene effettuato con il comando visibile nella linea 2: tramite esso, il Mambo decrementa *_SMS_invio* ogni volta che scade *TIMER_1SEC*.

Nel comando visibile alla linea 3, invece, è stato definito il contatore Counter3 come Alias di *_SMS_INVIO* (vedere il riquadro relativo agli Alias). Quando il contatore *_SMS_INVIO* va a zero, esso genera un evento software riconoscibile dal sistema operativo del localizzatore: questo evento, unitamente al controllo del Segnale GPS ed al fatto che l'autovettura sia in sosta oppure in movimento, fa sì che il sistema generi ulteriori allarmi, dai quali dipendono azioni specifiche verso il mondo esterno. Vediamo ora in dettaglio gli allarmi inseriti per il controllo degli SMS: il primo di essi è riportato nella linea 4.

Quando il contatore *SMS_INVIO* è a zero (significa che sono passati 5 minuti dall'evento precedente) ed il segnale GPS è valido, il localizzatore invia un SMS al numero predefinito, SMS che contiene informazioni su data, ora, latitudine, longitudine e velocità espressa in metri al secondo. Quando invece il segnale GPS non è valido (linea 5), viene inviato un messaggio con la relativa informazione "Segnale GPS non Valido". Il terzo ed ultimo allarme di questa serie, serve per inviare un messaggio per informare che l'autovettura è in sosta, pertanto invia le coordinate, la data e l'ora ma non la velocità, informazione inutile quando il veicolo è fermo. A questo punto, inserendo ad esempio le coordinate nel sito www.mapquest.com (oppure nel software Fugawi), è possibile verificare in tempo reale la posizione del veicolo. Visti i comandi utilizzati per gestire la messaggistica, vediamo ora l'uso degli allarmi nelle segnalazioni luminose tramite i led, segnalazioni effettuate secondo un

Tabella 3
Elenco comandi principali usati nel File di Configurazione di test

1	\$PFAL,CNF.Set,AL4=SYS.COUNTER.e_sms_invio:sys.counter_sms_invio.set=300
2	\$PFAL,CNF.Set,AL5=SYS.TIMER.e_1SEC:sys.counter_sms_invio.decrement=1
3	\$PFAL,CNF.Set,ALIAS.SYS.COUNTER3=_SMS_INVIO
4	\$PFAL,CNF.Set,AL6=SYS.COUNTER.e_SMS_INVIO&GPS.Nav.sFix=valid:GSM.SMS.SEND,"+399991234567",0,"Auto in Movimento: Data &(Date) Ora &(Time) Latitudine &(Lat) Longitudine &(Lon) Velocità &(Speed)"
5	\$PFAL,CNF.Set,AL7=SYS.COUNTER.e_SMS_INVIO&GPS.Nav.sFix=invalid:GSM.SMS.SEND,"+399991234567",0,"Segnale GPS non Valido"
6	\$PFAL,CNF.Set,AL16=SYS.COUNTER.e_SMS_INVIO&GPS.Nav.sFix=valid&IO.MOTION.sStanding:GSM.SMS.SEND,"+399991234567",0,"Auto in Sosta: Data &(Date) Ora &(Time) Latitudine &(Lat) Longitudine &(Lon)"
7	\$PFAL,CNF.Set,AL9=SYS.COUNTER.e_TIMING_MODE&GPS.Nav.sFix=valid:IO.LED_GPS.green=hpulse,400&IO.LED_GPS.red=low&IO.LED_GPS.blue=low
8	\$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
9	\$PFAL,CNF.Set,AL11=SYS.COUNTER.e_TIMING_MODE&GSM.sOpValid:IO.LED_GSM.green=hpulse,400&IO.LED_GSM.red=low&IO.LED_GSM.blue=low
10	\$PFAL,CNF.Set,AL12=SYS.COUNTER.e_TIMING_MODE&GSM.sOpInvalid:IO.LED_GSM.green=low&IO.LED_GSM.red=hpulse,400&IO.LED_GSM.blue=low
11	\$PFAL,CNF.Set,AL14=SYS.COUNTER.e_TIMING_MODE&SYS.BT.sConnected:IO.LED_STATUS.green=low&IO.LED_STATUS.red=low&IO.LED_STATUS.blue=hpulse,400
12	\$PFAL,CNF.Set,AL13=GSM.voiceCall.eIncoming:GSM.VoiceCall.Accept

Aggiornare il file di configurazione

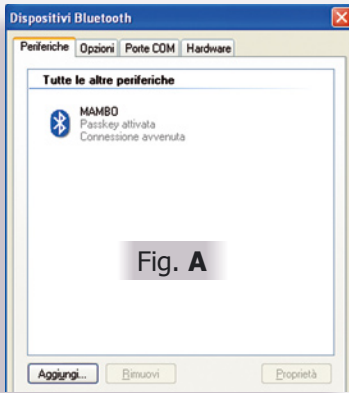


Fig. A

L'aggiornamento del file di configurazione prevede l'uso del software allegato al prodotto, denominato Workbench (tradotto alla lettera: il tavolo da lavoro). Una volta aggiunto al PC il nuovo dispositivo Bluetooth (nel nostro caso il MAMBO) e verificata la porta seriale virtuale che il sistema ha assegnato ad esso (Figura A), è necessario avviare il Workbench e connettere il sistema al Mambo facendo clic sulla prima icona in alto a sinistra (Figura B) denominata PORT (porta della periferica). Per verificare se il Mambo

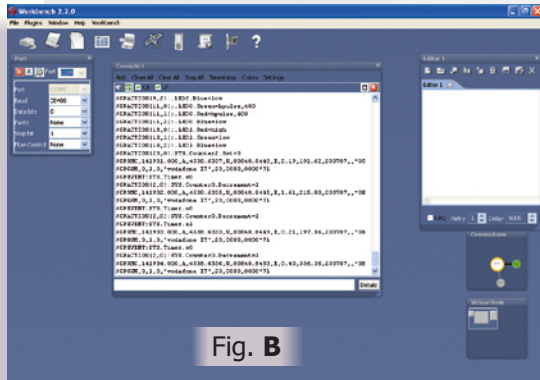


Fig. B

è connesso, si apre la finestra "Console" (dal menu Plugins) quindi la si collega facendo clic sul pallino "Console" della finestra "Connections": il funzionamento è confermato dal flusso di stringhe che appaiono sulla Console (Figura B). A questo punto occorre aprire e connettere la finestra "Editor" di Figura C (tramite il menu Plugins), sulla quale dobbiamo digitare ed inviare (facendo clic sull'icona "freccia in diagonale verso l'alto") la seguente stringa: **\$PFAL,Sys.device.FactoryReset**. Tramite questa dichiarazione, forziamo il MAMBO a ricaricare al suo interno la configurazione di default. Avendo azzerato la periferica, essa riparte dallo stato di reset, pertanto si potrebbe avere qualche problema di riconnessione automatica: conviene quindi abbattere la connessione PORT, togliere la batteria del MAMBO, reinserirla, riavviare il sistema e ristabilire la connessione tramite la procedura già descritta.

Una volta riconnesso il tutto, occorre visualizzare e collegare la finestra di Editor, nella quale apriamo il nostro file di configurazione (tale file è scaricabile dal sito della rivista): inviamo il file al Mambo facendo clic sull'icona già usata in fase di "default reset" (vedere figura D): notate che durante il trasferimento la "freccia" si trasforma in una "X". Alla fine dell'operazione conviene sempre scollegare il programma e rimuovere temporaneamente la batteria del Mambo, in modo da riavviare il sistema con un "Hard Reset". Riconnettendo il localizzatore ed aprendo la finestra Console (Figura E) vediamo fluire la serie di eventi che confermano il funzionamento della nuova configurazione. La procedura può sembrare un po' complessa, ma alla fine scoprirete che basta un po' di pratica.

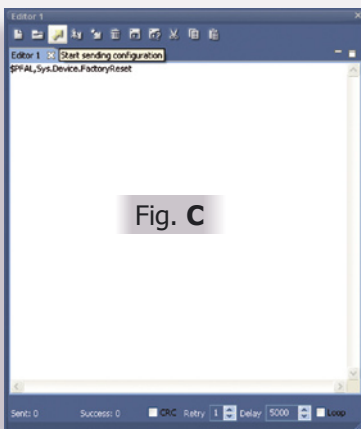


Fig. C

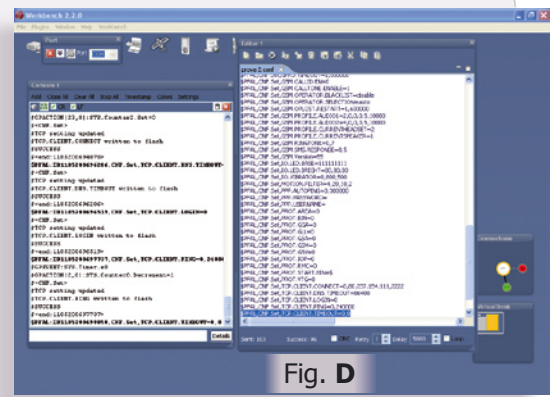


Fig. D

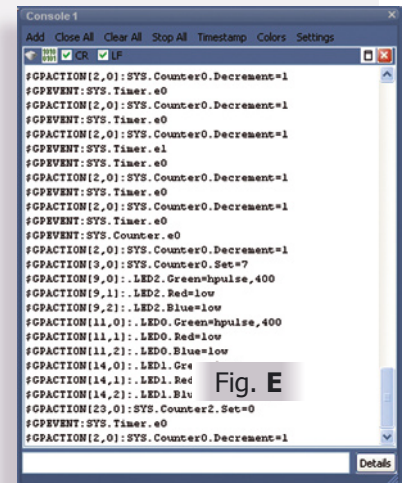


Fig. E

criterio logico e facilmente intuibile ossia facendo lampeggiare il led del GPS e quello del GSM a luce verde nel caso di segnale GPS o copertura GSM validi, e a luce rossa nel caso opposto. La connessione Bluetooth, invece, è indicata dal lampeggio del led blu. Di seguito riportiamo i riferimenti alle linee di comando in tutti i casi previsti:

- segnale GPS valido = led verde a luce pulsante (linea 7);
- segnale GPS non valido = led rosso acceso a luce pulsante (linea 8);

- copertura del servizio GSM = led verde acceso a luce pulsante (linea 9);
- mancanza di copertura del servizio GSM = led rosso acceso a luce pulsante (linea 10);
- Bluetooth connesso = led blu lampeggiante (linea 11).

Come potete notare, le segnalazioni in questo caso sono molto intuitive, al punto che basta il colpo d'occhio per conoscere lo stato del Mambo. Un ulteriore allarme è stato inserito per gestire le chiamate in ingresso, alle quali il Mambo risponde auto-

maticamente avviando la funzione di ascolto ambientale, ossia facendo ascoltare (tramite il cellulare che ha effettuato la chiamata) quanto avviene nell'ambiente in cui il localizzatore è stato posizionato. Il comando con cui si avvia l'ambientale è visibile in linea 12: se nel comando venisse inserito il numero del chiamante, il Mambo accetterebbe solamente le chiamate provenienti da tale numero, ignorando tutte le altre. Prestate attenzione ad un particolare molto importante: il listato che ab-

Cos'è un Alias

Col termine "Alias" si intende un nome scelto dal programmatore ed assegnato ad un particolare contatore, timer, pulsante, Led, eccetera; l'alias permette di ricordare più facilmente ed in modo intuitivo una certa "funzione" software avente un nome generico predefinito (Counter1, Timer2, LED1...). Parlando ad esempio del pulsante centrale del Mambo, si ricorda più facilmente "_MIDDLE" come alias piuttosto che BTN0 come nome predefinito. Pertanto, dichiarando in configurazione **\$PFAL,CNF.Set.Alias.IO.BTN0=_Middle**, il pulsante può essere richiamato tramite _MIDDLE piuttosto che BTN0. Di seguito trovate gli alias utilizzati nel File di Configurazione di questo articolo.

Estratto dal "File di Configurazione"

```
$PFAL,CNF.Set,ALIAS.CNF=Config
$PFAL,CNF.Set,ALIAS.IO.BTN0=_MIDDLE
$PFAL,CNF.Set,ALIAS.IO.BTN1=_UPPER
$PFAL,CNF.Set,ALIAS.IO.BTN2=_LOWER
$PFAL,CNF.Set,ALIAS.IO.BTN3=_HEADSET
$PFAL,CNF.Set,ALIAS.IO.LED0=_GSM
$PFAL,CNF.Set,ALIAS.IO.LED1=_STATUS
$PFAL,CNF.Set,ALIAS.IO.LED2=_GPS
$PFAL,CNF.Set,ALIAS.SYS.COUNTER0=_TIMING_MODE
$PFAL,CNF.Set,ALIAS.SYS.COUNTER1=_SLEEP_MODE
$PFAL,CNF.Set,ALIAS.SYS.COUNTER2=_BAT_MODE
$PFAL,CNF.Set,ALIAS.SYS.COUNTER3=_SMS_INVIO
$PFAL,CNF.Set,ALIAS.SYS.TIMER0=_1SEC
$PFAL,CNF.Set,ALIAS.SYS.TIMER1=_20SEC
$PFAL,CNF.Set,ALIAS.SYS.TIMER2=_SLEEP
$PFAL,CNF.Set,ALIAS.SYS.TIMER3=_DOUBLE
$PFAL,CNF.Set,ALIAS.SYS=System
```

biamo commentato in queste pagine non è quello completo; noi abbiamo voluto focalizzare la vostra attenzione sui comandi più importanti, ossia quelli che hanno un ruolo fondamentale nell'esecuzione delle procedure e che, proprio per tale ragione, presentano una sintassi lunga e complessa. Quello completo (che per motivi di spazio non è stato allegato in toto), contiene anche le istruzioni relative alla cancellazione e modifica di alcuni allarmi contenuti nella configurazione predefinita; esso può essere scaricato direttamente dal sito www.elettronica.in.it.

Conclusione

Con questo articolo abbiamo concluso la presentazione del primo vero programma per il localizzatore Mambo. Vi anticipiamo che ne abbiamo altri in cantiere, e tutti molto interessanti, quindi continuate a seguirci. A titolo puramente informativo, informiamo che la Falcom (casa produttrice del Mambo) ha di recente rilasciato una nuova release del firmware, proprio mentre noi stavamo scrivendo questo articolo. In questa sede, non abbiamo tenuto conto delle eventuali variazioni della nuova release, attualmente in fase di studio da parte nostra, la quale verrà discussa in seguito, in una puntata dedicata esclusivamente alle nuove funzioni introdotte.





comune di scandiano



27/28
OTTOBRE
2007
SCANDIANO
(REGGIO EMILIA)
CENTRO FIERISTICO

4^a FIERA REGIONALE DELL'ELETTRONICA

Orari:
sabato
9,00-18,30
domenica
9,00-18,00

- ◆ TELEFONIA
- ◆ COMPONENTISTICA
- ◆ COMPUTER
- ◆ HI-FI CAR
- ◆ RADIANTISMO
CB e OM
- ◆ VIDEOREGISTRAZIONE
- ◆ MERCATINO
DELLE PULCI
RADIOAMATORIALI

con il patrocinio
di A.R.I. sez. Scandiano

INFO:
0522/857436
www.fierascandiano.it

Biglietti:
Intero euro 7,00
Ridotto euro 4,00
Ragazzi fino a 12 anni gratuito

Aggiorniamo il firmware del Mambo

di *Alessandro Orlando*

Analizziamo in dettaglio la procedura per aggiornare il firmware del localizzatore Mambo che viene effettuata tramite qualsiasi applicativo di emulazione di terminale come il comune HyperTerminal di Windows.



Nelle note conclusive del precedente articolo, pubblicato sulla rivista 119, avevamo anticipato che Falcom, l'azienda che produce il localizzatore Mambo, aveva appena rilasciato una nuova versione del firmware. Nel corso di tale articolo non abbiamo volutamente commentato il nuovo firmware perché volevamo provarlo prima di pubblicare qualsiasi informazione a riguardo. Bene: dopo averlo studiato e provato adeguatamente, è giunto il momento di effettuare l'upgrade del localizzatore. Per aggiornare il dispositivo con il nuovo firmware, esistono teoricamente due vie: il Workbench e l'emulazio-

ne di terminale. La prima consiste nell'uso dell'opzione firmware Uploader, presente nel Workbench (a titolo puramente informativo, vi ricordiamo che tutti i software di gestione nonché i manuali ed i file di aggiornamento sono contenuti nel CD allegato al prodotto); la seconda invece, viene messa in pratica facendo uso del classico HyperTerminal, l'emulatore di terminale presente in ogni versione di Windows. Attingendo al nostro bagaglio di esperienze eravamo sicuri che non avremmo avuto problema alcuno nell'utilizzo del Workbench, mentre HyperTerminal ci avrebbe fatto tribolare, ma di fatto le cose sono andate

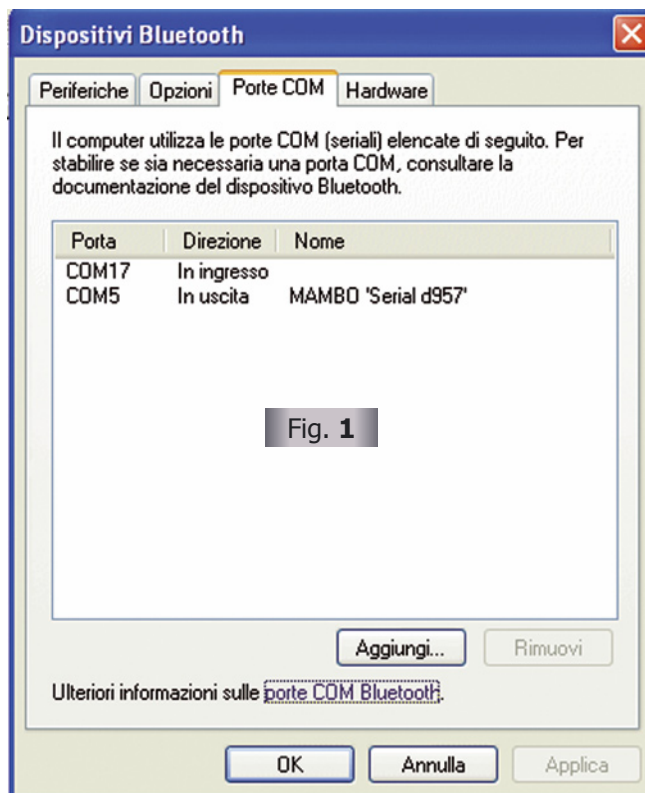


Fig. 1

diversamente da quanto pensavamo. Con il Workbench abbiamo effettuato molteplici tentativi di aggiornamento della flash, ma abbiamo riscontrato una certa difficoltà nel porre il Mambo in modalità di RedBoot (modalità necessaria per effettuare l'aggiornamento firmware) quindi, a fronte dei risultati ottenuti, consigliamo di usare la versione attuale di Workbench (sicuramente Falcom interverrà in seguito, apportando le

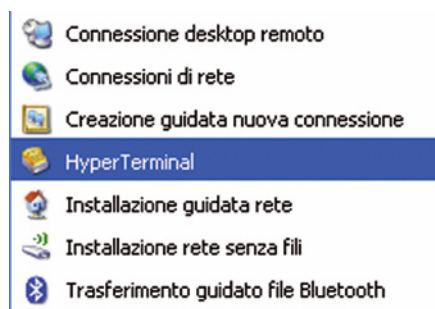
di Mambo è abbastanza laboriosa: ebbene anche l'aggiornamento del firmware tramite terminale non è da meno ma con il vantaggio, tuttavia, di funzionare sempre senza problemi se le istruzioni vengono seguite alla lettera.

Il firmware ufficiale

Le versioni rilasciate sino ad oggi da Falcom sono tre e precisamente la 1.0.1 (versione iniziale), la 1.0.2

dovute correzioni in una nuova versione) mentre consigliamo la seconda soluzione, che fa uso di HyperTerminal. Sappiate che qualsiasi programma di emulazione di terminale è adatto allo scopo: noi abbiamo scelto di utilizzare HyperTerminal perché usato anche da Falcom e perché presente su qualsiasi PC con sistema operativo Windows. Dagli articoli precedenti avete compreso che ogni procedura di configurazione e programmazione

Fig. 2



(che implementa la sostituzione degli indirizzi della Flash Memory) ed infine la 1.0.3 rilasciata in data 04/07/2007, per l'appunto durante la stesura dell'articolo precedente, dove sono state introdotte sostanziali modifiche al linguaggio di programmazione del Mambo. A tal proposito il set di istruzioni riconosciute dal localizzatore può essere definito di "basso livello" e "stand-alone": basso livello perché le istruzioni risultano stringate ed essenziali, e non contengono le classiche locuzioni logiche utilizzate, per esempio, dal BASIC; "stand-alone" perché si tratta di un linguaggio scritto appositamente e che pertanto non può essere compreso (leggasi "interpretato ed eseguito") da altri dispositivi presenti sul mercato.

Configurazione con HyperTerminal

Il punto di partenza della procedura è rappresentato dall'aggiunta del dispositivo all'elenco degli apparati Bluetooth noti al PC che siamo soliti utilizzare, aggiunta che viene effettuata eseguendo la ricerca del localizzatore e il relativo *pairing* (eventualmente fate riferimento a quanto già pubblicato sul fascicolo 117). Dopo aver aggiunto il nuovo dispositivo Bluetooth e identificato la porta che gli è stata assegnata da Windows (Figura 1), si avvia HyperTerminal (Figura 2): a questo punto appare la finestra con cui si procede alla generazione di una "Nuova connessione" (Figura 3), connes-

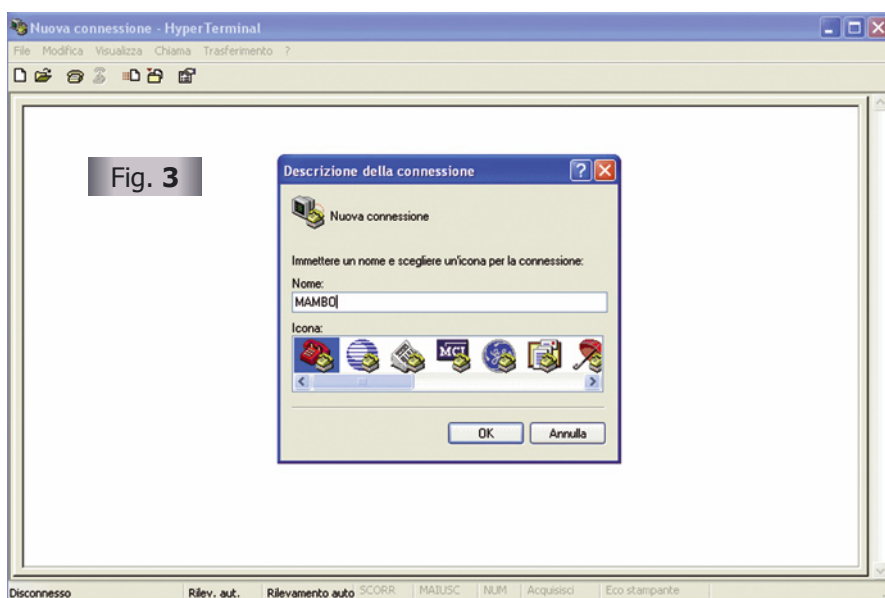


Fig. 3

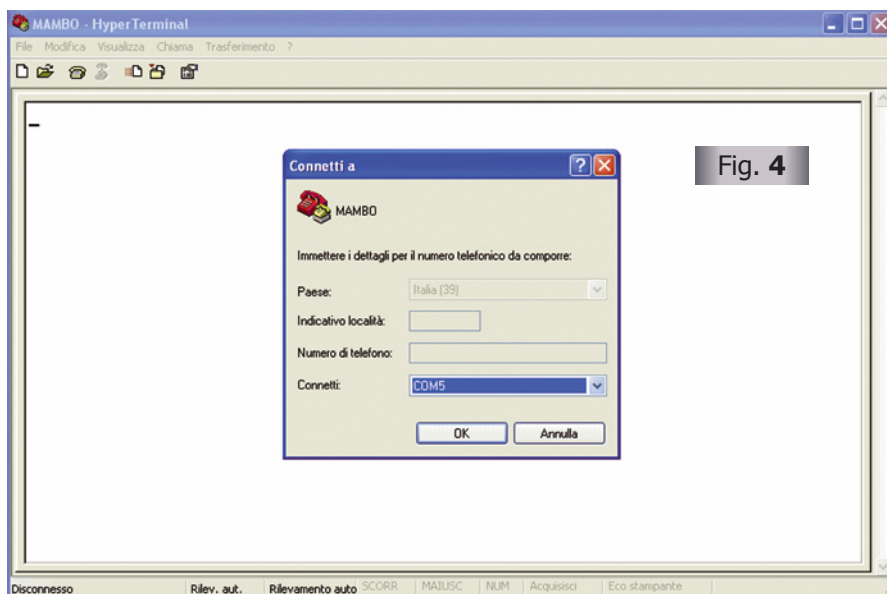


Fig. 4

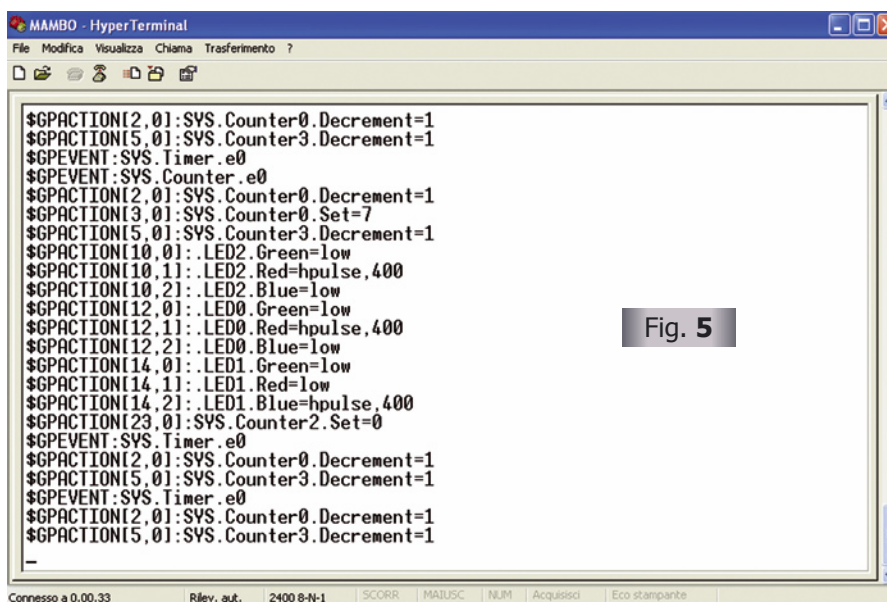


Fig. 5

ne a cui diamo per esempio il nome “Mambo” molto facile da ricordare anche per gli aggiornamenti futuri. Ora l'applicativo fa richiesta della porta COM su cui aprire la connessione (Figura 4): l'identificativo della seriale è quello visto nella Figura 1. Una volta inserita la ComPort e confermato il comando, si apre una nuova finestra in cui scorre il flusso di dati generato dall'esecuzione della “configurazione utente” (leggasi anche “programma”) presente attualmente in memoria (Figura 5).

Il passo successivo consiste nel configurare le opzioni ASCII del terminale: dal menu “File-Proprietà” di HyperTerminal si richiama la finestra delle “Impostazioni”, che va configurata come in Figura 6. Quindi si clicca sul pulsante “Impostazioni ASCII” per caricare la finestra visibile in Figura 7, sulla quale va selezionata la casella “Aggiungi avanzamento riga ad ogni ritorno a capo inviato” (prima casella di spunta).

Modalità RedBoot

Per effettuare l'aggiornamento, il dispositivo deve essere posto in modalità *RedBoot* (questo nome è stato scelto da chi ha scritto il firmware!) ossia quel particolare stato operativo che consente al Mambo di ricevere e ➤

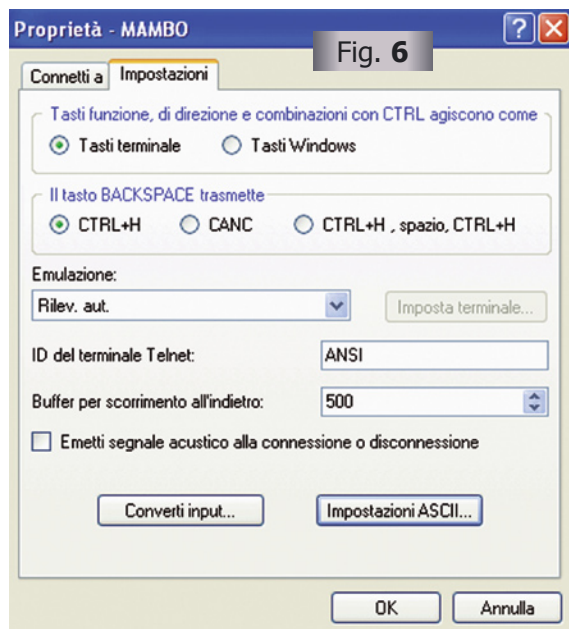


Fig. 6



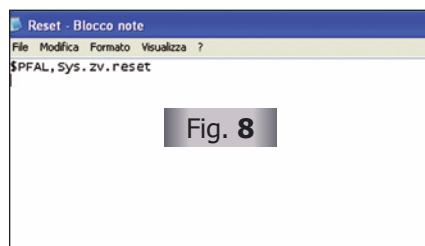
Fig. 7

memorizzare un file Binario. Per prima cosa, si deve preparare e salvare un file di testo (ad esempio *reset.txt*) contenente l'istruzione

\$PFAL,Sys.ZV.Reset<CR-LF>

in cui i caratteri terminatori di stringa CR-LF sono ottenuti premendo il tasto di invio "ENTER" (Figura 8).

Ora è necessario inviare al Mambo il file *reset.txt* appena creato: per fare ciò si utilizza l'opzione di HyperTerminal "Trasferimento - Invia File di testo". Dalla finestra che appare selezionate il file *reset.txt* e confermate l'invio facendo clic sul tasto "Apri".



Nella finestra principale di HyperTerminal appare una serie di dati a segnalare che si è entrati in modalità RedBoot. L'ultima riga comunica che si hanno a disposizione 9 secondi per premere i tasti CTRL+C e terminare (se si vuole) la procedura (Figura 9).

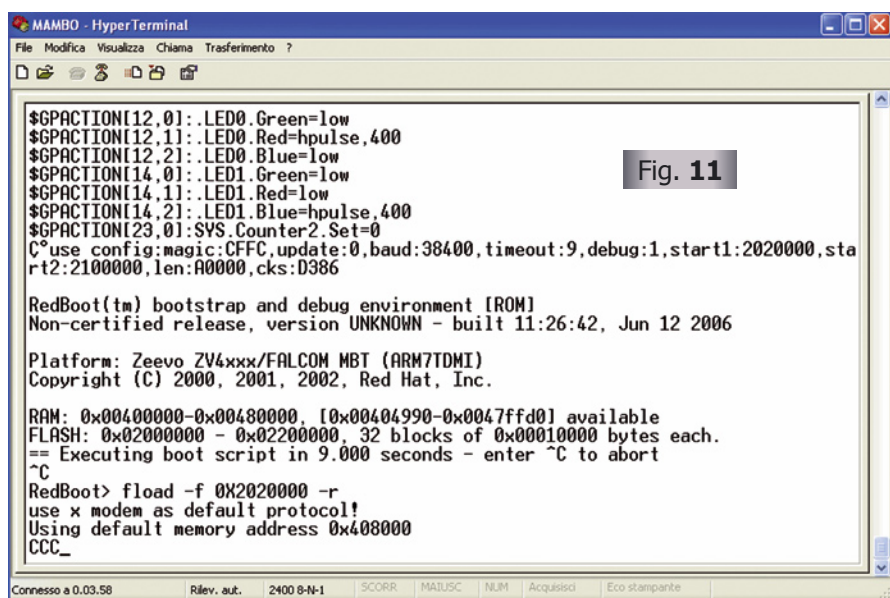
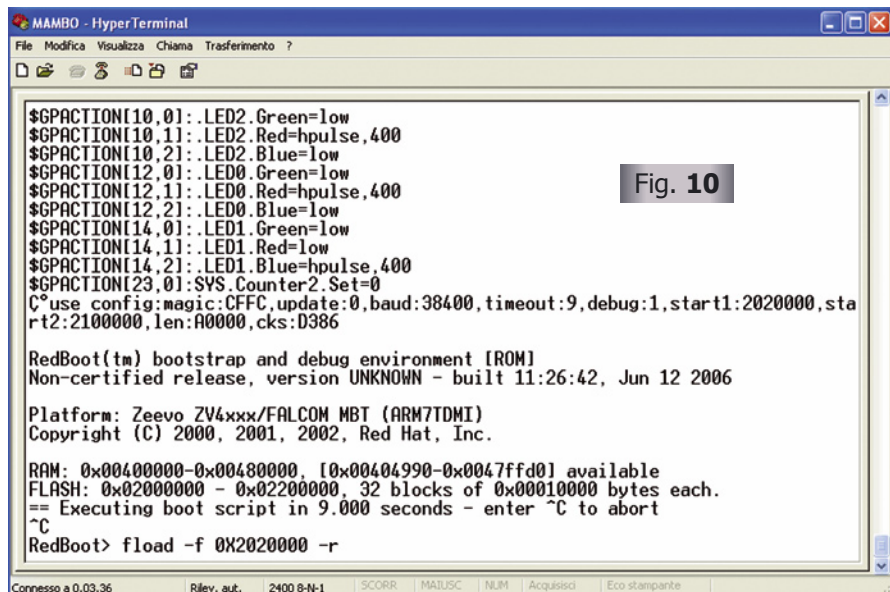
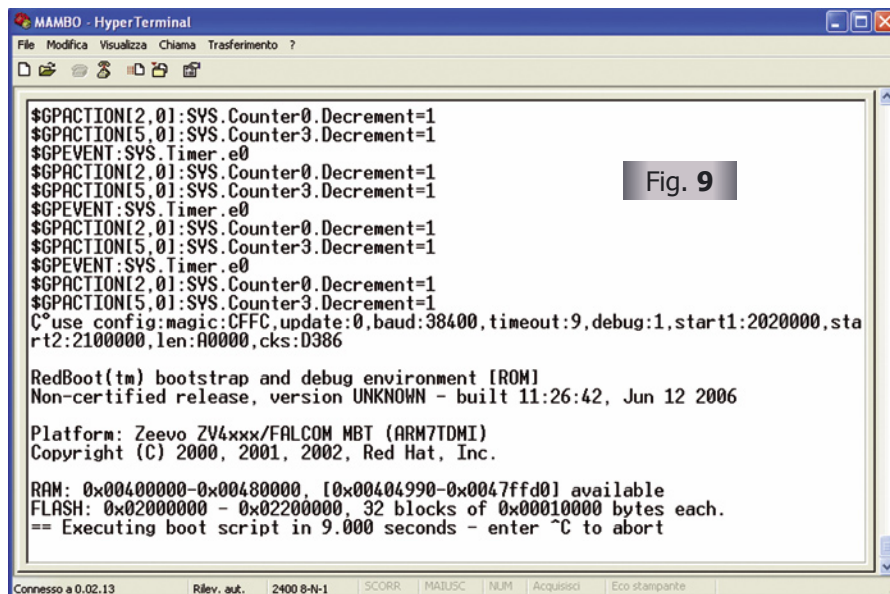
Aggiornamento del firmware

Quanto visto fino ad ora serve esclusivamente per predisporre il Mambo a ricevere l'aggiornamento.

Per inviare il file binario vero e proprio (il nuovo firmware) occorre innanzitutto stabilire l'indirizzo di partenza ossia l'indirizzo fisico di memoria da cui il dispositivo deve iniziare a salvare i byte ricevuti: per fare ciò si deve digitare il comando

fload -f 0X2020000 -r <ENTER>

sulla finestra principale di HyperTerminal (Figura 10); il Mambo risponde avvertendo di inviare il file facendo uso del protocollo XModem,



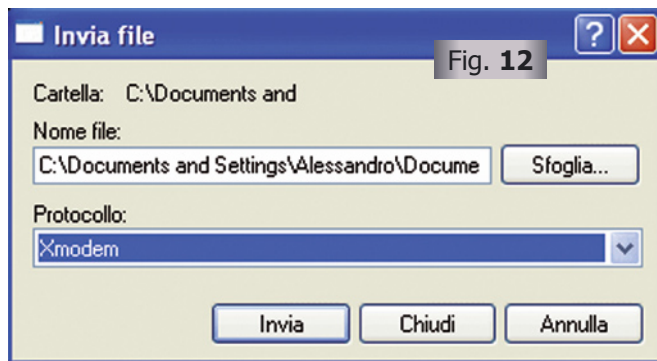


Fig. 12

come visibile in Figura 11. Ora, tramite l'opzione "Trasferimento - Invia File", si carica la finestra visibile in Figura 12, attraverso la quale potete scegliere il file da inviare (nel nostro caso, questi è nominato "Mambo_2.4.3_final") e il protocollo XModem, selezionabile tramite l'apposito menu a tendina. Ora il Mambo è in fase di aggiornamento (Figura 13) al termine della quale occorre digitare il comando

Reset <ENTER>

per effettuare un riavvio completo. In alternativa è possibile scollegare fisicamente la batteria per qualche istante, al fine di essere certi che il reset hardware abbia effetto. Una volta aggiornato il prodotto, le istruzioni usate fino ad ora non sono più valide in quanto il set di comandi utilizzato nel nuovo firmware è differente e quindi, se per qualsiasi motivo doveste effettuare un downgrade (ossia ritornare alla versione precedente) il file di testo *reset.txt* dovrà contenere l'istruzione "\$PFAL,Sys.

Device.Reset" mentre per iniziare l'aggiornamento si dovrà digitare (sempre entro 9 secondi) solo l'istruzione "flood <ENTER>": gli indirizzi verranno definiti automaticamente e si potrà

scegliere indifferentemente tra protocollo *1K XModem* (più veloce e pertanto consigliato dalla stessa Falcom) e XModem.

Il nuovo firmware

Rispetto alla revisione precedente, quella nuova presenta delle differenze sostanziali in quanto sono state aggiunte nuove importanti funzioni.

Per esempio sono stati inseriti il protocollo 3D per il GPS, la funzione "Spy-Call" (che consente di silenziare completamente il dispositivo durante un ascolto ambientale), le possibilità di scaricare i dati GPS scritti nel registro "storico" effettuando una sola lettura e di inviare delle e-mail attraverso il protocollo SMTP e molte altre ancora. Sfortunatamente, i file di configurazione preparati per il firmware precedente non sono più validi, quindi vanno riscritti seguendo le procedure definite nel manuale in Inglese. In un prossimo articolo vedremo come utilizzare le nuove funzioni, compresa quella che consente di inviare i dati GPS attraverso le e-mail, con conseguente notevole risparmio sui costi di gestione telefonica.

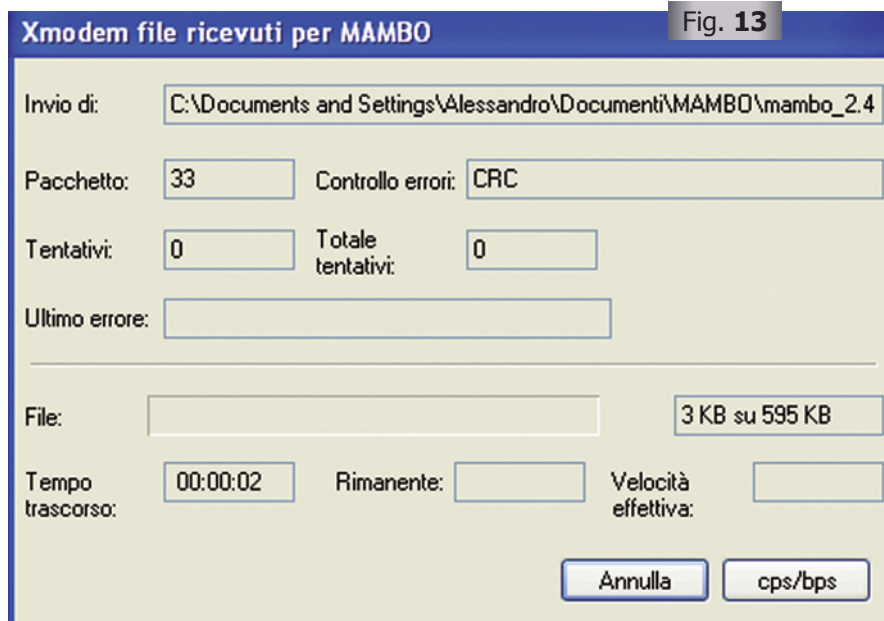


Fig. 13

per il

MATERIALE

Il sistema di localizzazione satellitare GPS/GSM/GPRS Bluetooth Mambo (cod. Mambo55) può essere acquistato al prezzo di 510,00 Euro, IVA inclusa. La confezione include, oltre al localizzatore, il pacco batterie, un auricolare, un caricabatterie con adattatore sia di rete che da accendisigari per auto, un laccio per il trasporto al collo ed il CD contenente i manuali in inglese e l'ultima versione di firmware disponibile al momento dell'acquisto.

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

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

Radiomicrofoni: la gamma dei professionisti

NEW

Valigetta in alluminio per radiomicrofono
 Indicata per modelli MICW40, MICW42 e MICW41-2.
 Dimensioni 500 x 270 x 85 mm.
Cod. MICWAC1 - € 25,00
 Disponibile anche la versione per microfono MICW41.
 Dimensioni 500 x 270 x 130 mm.
Cod. MICWAC2 - € 30,00

Sistemi completi UHF 8 canali - true diversity

Sistema radiomicrofonico professionale operante sulla banda UHF caratterizzato da un'eccellente qualità del suono studiato per offrire con semplicità e immediatezza soluzioni a un'ampia gamma di utilizzi.

È composto da un'unità ricevente con tecnica di ricezione true diversity e tecnologia a microprocessore dotata di display LCD retroilluminato e da un microfono wireless palmare. Estremamente compatto e pratico, è ideale per ap-

plicazioni in spettacoli, conferenze, lezioni, ecc. Il collegamento via radio con l'unità base consente di operare con la massima libertà e autonomia con una portata di circa 100 metri in campo aperto.

La disponibilità di 8 canali consente l'utilizzo di più sistemi in simultanea. Il modello MICW42 dispone anche di funzione "monitor" mentre il modello MICW41 è dotato di base con doppio ricevitore completa di due radiomicrofoni.

Ricevitore True Diversity: frequenza di lavoro 863-865 MHz; 8 canali; modulazione FM (PLL); temperatura d'esercizio da -10°C a +55°C; rapporto F/N max. 80 dB; rapporto P/N max. 80 dB; sensibilità RF 3 µV (52 dB S/N); alimentazione 14 Vac / 500 mA (adattatore di rete incluso).

Trasmettitore con microfono: potenza d'uscita RF @ 50 ohm, 8 mW-30 mW; RF squelch max. 50 dB; alimentazione tramite 2 batterie 1,5 V AA (incluse).

Codice	Rif.	Descrizione	Risp. frequenza	Dimensione base	Peso base	Dimensione microfono	Peso microfono	Prezzo C
MICW40	1	Singolo radiomicrofono	80 Hz - 15 kHz (+3dB)	220 x 180 x 42 mm	1,3 kg	Ø 50 x 275 mm	225 g	292,00
MICW42	2	Singolo radiomicrofono	60 Hz - 18 kHz (+3dB)	210 x 190 x 42 mm	1,15 kg	Ø 50 x 255 mm	410 g	320,00
MICW41	3	Doppio radiomicrofono	60 Hz - 18 kHz (+3dB)	420 x 180 x 42 mm	2,2 kg	Ø 50 x 275 mm	225 g	390,00

Sistema UHF 8 canali con Body Pack - true diversity

Per una mobilità assoluta! Sistema radiomicrofonico professionale, affidabile e competitivo, caratterizzato da un suono di qualità superiore e da una trasmissione sempre nitida. Il set è composto da un'unità ricevente, con tecnica di ricezione true diversity, dotata di display LCD retroilluminato e da un trasmettitore Body Pack con microfono lavalier. Estremamente compatto e pratico, è ideale per applicazioni quali lezioni, spettacoli, conferenze, ecc. La disponibilità di 8 canali, consente l'utilizzo di più sistemi in simultanea.

Ricevitore true diversity: funzione monitor; fre-

quenza di lavoro 863-865 MHz; modulazione FM (PLL); risposta in frequenza 60 Hz-18 kHz (+3 dB); temperatura d'esercizio da -10°C a +55°C; sensibilità RF 3 µV (52 dB S/N); alimentazione 14 Vac/500 mA (adattatore di rete incluso); dimensioni 210 x 190 x 42 mm; peso 1,15 kg. Portata ~100 m (in campo aperto).

Trasmettitore Body Pack: potenza d'uscita RF a 50 ohm, 8 mW-30 mW; RF squelch max. 50 dB; alimentazione tramite 2 batterie da 1,5 V AA (incluse); dimensioni 105 x 65 x 20 mm; peso 82 g.

ACCESSORI:

- **Trasmettitore Body Pack UHF supplementare (cod. MICW43 - € 120,00);**
- **Microfono lavalier per trasmettitore Body Pack MICW43 (cod. MICW45 - € 15,00);**

- **Microfono ad archetto per trasmettitore Body Pack MICW43 (cod. MICW44 - € 45,00).**

NEW

MICW42-2
€ 320,00

Set completo UHF

Sistema monocanale operante alla frequenza di 863,300 MHz composto da un trasmettitore palmare e da un ricevitore, adatto per impieghi professionali. Completo di valigetta per il trasporto.

Ricevitore: modulazione FM (F3E); risposta in frequenza 50 Hz±15 kHz; sensibilità RF 3 µV/20 dB Sinad; squelch controllo esterno; uscita segnale bilanciata 60 mV/600 ohm, sbilanciata 750 mV/5 kohm; alimentazione 12 Vdc/500 mA (adattatore di rete incluso); dimensioni 222 x 114 x 36 mm; peso 550 g.

Microfono con trasmettitore: mo-

UHF

MICW15E
€ 205,00

dulazione FM (F3E); risposta in frequenza 50 Hz±15 kHz; pre-enfasi 50 µs; potenza di uscita RF 10 mW; alimentazione con batteria 9 V; assorbimento 30 mA; autonomia superiore a 8 ore con batteria alcalina; lunghezza 235 mm; peso 210 g.

Sistema completo VHF

Sistema monocanale per impieghi professionali, composto da un'unità ricevente e da un trasmettitore palmare operante alla frequenza di 184,032 MHz.

È caratterizzato da un'elevata stabilità e notevole qualità dell'audio grazie alla modulazione FM.

Completo di valigetta per il trasporto. **Ricevitore:** modulazione FM (F3E); rapporto S/N > 90 dB; risposta in frequenza 50 Hz-15 kHz; sensibilità RF 3 µV per 20 dB Sinad; squelch (controllo esterno); livello uscita AF non bilanciato, 0-300 mV, 10 kohm; connettore d'uscita jack 6,35 mm; alimentazione con adattatore di rete 12 Vdc-18 V / 300 mA; dimensioni 152 x 98 x 36 mm; peso 226 g.

VHF

MICW8D
€ 118,00

NEW

Microfono con trasmettitore: modulazione FM (F3E); deviazione massima ± 40kHz; dinamica 100 dB; risposta in frequenza 50 Hz-15 kHz; range frequenza portante 160-245 MHz; potenza RF 10 mW; alimentazione tramite batteria stilo da 1,5 V; assorbimento 60 mA; autonomia superiore alle 12 ore (con batteria alcalina); lunghezza 233 mm; peso 210 g.

È disponibile anche modello operante alla frequenza di 181,660 MHz (cod. MICW8A - € 110,00).

Mambo, la prova pratica col nuovo firmware

di **Alessandro Orlando**



Analizziamo le nuove funzioni della versione firmware 1.0.3. Ne verifichiamo poi il corretto funzionamento tramite un file di configurazione che, ogni due minuti, invia al servizio di localizzazione on-line www.gpstracer.net un messaggio di posta elettronica contenente data, ora e posizione.

Nel corso dell'articolo pubblicato sul fascicolo precedente abbiamo spiegato in dettaglio la procedura da utilizzare per aggiornare il firmware del Mambo tramite un programma di emulazione terminale, nella fattispecie il classico HyperTerminal. Tipicamente, l'aggiornamento del firmware è necessario quando si vogliono aggiungere nuove funzioni o correggere errori di programmazione in un dispositivo. Nel nostro caso, la nuova release corregge dei bachi (errori di programma) e aggiunge delle nuove funzioni. Vediamo insieme le variazioni più importanti.

La nuova release

Le differenze tra le due release, senza considerare le versioni precedenti o intermedie (chiamate BETA), sono molteplici e alcune di esse sono compatibili solo con l'ultima versione hardware "1E2". Per sapere quale versione di hardware è presente nel vostro Mambo, potete utilizzare il WorkBench attraverso la finestra CONSOLE, tramite la quale è possibile visualizzare il traffico dati da e per il Mambo. Il comando da inviare affinché il localizzatore comunichi la versione HW è

\$PFAL,MSG.Version.Complete.

Nota per l'aggiornamento del firmware



Dopo aver pubblicato il fascicolo precedente, abbiamo continuato a eseguire le nostre prove e verifiche, nel corso delle quali ci siamo resi conto di un fatto importante che non è stato citato nella precedente puntata che, ricordiamo, riguarda l'aggiornamento del firmware del MAMBO. Abbiamo verificato che in alcuni rari casi, il dispositivo si blocca durante la fase di aggiornamento e ciò è dovuto, essenzialmente, a due ragioni: la batteria scarica e la perdita della connessione Bluetooth. Sfortunatamente, quando ciò si verifica, pur riattivando la connessione e riaprendo una sessione di HyperTerminal, non è più possibile inviare il file di reset ne-

cessario per porre il Mambo in modalità RedBoot (indispensabile per avviare l'aggiornamento). In questo caso occorre eseguire una semplice procedura manuale che consta di due fasi: nella prima si apre il collegamento Bluetooth e si avvia la sessione HyperTerminal, mentre nella seconda si devono pigiare contemporaneamente tutti i tre pulsanti del localizzatore per forzare manualmente la modalità RedBoot. Da questo momento è possibile utilizzare la procedura già vista nella puntata precedente, ossia digitare sulla tastiera del PC:

"CRTL-C"

quindi

"fload"

seguito da **ENTER** e inviare il file **"mambo_2.4.3_final.bin"** con il protocollo **1K Xmodem**.

Vediamo ora le più importanti tra le nuove funzioni implementate, tralasciando tutto ciò che concerne la risoluzione dei bachi firmware.

Beep Signal e Spy Call

Questa funzione è utilizzabile solo con hardware 1E2. Come facilmente intuibile dal nome, essa consente al Mambo di porsi costantemente in modalità silenziosa anche durante l'accensione e di disattivare l'altoparlante lasciando però attivo il microfono. Risulta essere molto utile nelle applicazioni in ambito sicurezza per l'ascolto ambientale.

Il protocollo 3D

Tipicamente, tutti i ricevitori GPS calcolano la posizione riferita ad uno spazio invece che a una superficie, pertanto la posizione viene elaborata sui tre assi x-y-z. Semplificando al massimo tutti i concetti che stanno alla base del calcolo, dalle variabili assolute x-y-z vengono estratte Latitudine, Longitudine e Altitudine.

Il nuovo firmware è in grado di fornire anche l'informazione relativa all'Altitudine, espressa in metri. Questa nuova opzione è particolarmente utile nel caso in cui il dispositivo venga uti-

lizzato per esempio da alpinisti, escursionisti o amanti del volo a vela o con aerei ultraleggeri.

Dynamic Protocol

Anche questa funzione è supportata solo dall'hardware 1E2, ed era già presente anche nella precedente versione. Qui sono stati aggiunti due nuovi comandi: **&(SMSNUMBER)** e **&(STARTUPREASON)**. Il primo restituisce il numero del mittente di un SMS; il secondo, invece, il motivo per cui il Mambo si è acceso l'ultima volta.

In entrambi i casi, la comunicazione viene effettuata tramite un SMS di risposta. Le cause di riavvio dopo un stand by possono essere sia l'**Accensione** effettuata manualmente tramite pulsante sia quella automatica; quest'ultima può avvenire per effetto dello scadere di un **Timer**, della ricezione del segnale di **Ring** di una chiamata telefonica in corso, o del cambio di posizione (inteso come movimento fisico, **Motion**), sempre, comunque, durante lo stato di Stand-By.

Geofence Event

Si tratta di una nuova interessante funzione con la quale viene generato un

evento di segnalazione ogni qual volta il dispositivo entra in una delle aree di interesse precedentemente impostate. Tramite il Geofence Event è possibile definire degli SMS di allarme specifici per zona, ossia delle segnalazioni che vengono inviate quando il localizzatore entra ed esce da una delle aree di interesse memorizzate.

History

La funzione History era già presente nella precedente versione. Qui è stato aggiunto il nuovo comando **GPS.History.push** con il quale è possibile ottenere la cronologia completa delle posizioni memorizzate digitando il solo comando anziché più comandi in sequenza come avveniva nella precedente revisione. Il formato dei dati può essere scelto tra **"txt, rmc o bin"**.

TCP

Anche lo stack TCP era già presente. Sono state però implementate rispettivamente:

- la possibilità di inviare delle e-mail utilizzando la connessione Internet tramite il comando **MSG.send.SMTP**,
- la modalità di "invio sicuro" dei pacchetti tramite il comando **TCP.client.sendmode**, comando che permette di inviare i pacchetti dati in maniera veloce (senza controllo di corretta ricezione da parte del destinatario) o sicura (con controllo).

MSG.version.bios

Si tratta di un nuovo comando che consente di verificare la versione del Bios.

Eventi e stati (Events & States)

Facendo qualche passo indietro e ritornando agli articoli in cui avevamo descritto il primo set di comandi e alcuni esempi di script applicativi, ricorderete che tutto ciò che il Mambo può fare (se opportunamente configurato) è costituito da "eventi" e "stati", il verificarsi dei quali genera una serie di azioni definite dallo script di programma. In questa versione FW sono stati

Tabella 1 - Comandi di configurazione GPRS

Comandi	
1	\$PFAL,CNF.Set,TCP.SMTP.CONNECT=1,mail.libero.it,25
2	\$PFAL,CNF.Set,TCP.SMTP.FROM=tuonome@libero.it
3	\$PFAL,CNF.Set,TCP.SMTP.LOGIN="internet.wind",180,"",""
4	\$PFAL,CNF.Set,GPRS.APN=internet.wind
5	\$PFAL,CNF.Set,GPRS.AUTOSTART=1

aggiunti degli eventi nuovi. Il primo è in grado rilevare l'invio o meno delle e-mail via SMTP (**TCP.SMTP.esend** e **TCP.SMTP.failed**); il secondo è il **Sys.power.voltage**, con il quale è possibile stabilire degli stati di allarme della tensione della batteria. Infine **GSM.GPRS.traffic** consente di stabilire la quantità di traffico GPRS rispetto ad un valore stabilito.

Tenete presente un particolare molto importante: noi ci siamo limitati a descrivere a livello generale solo alcuni dei cambiamenti apportati al firmware; in realtà le variazioni sono molte di più e comprendono anche le modifiche, le correzioni dei banchi di programma e le soppressioni di alcune locuzioni utilizzate nella precedente versione come per esempio la soppressione del parametro

sys.batt.charge mode

o la modifica del comando di reset durante la fase di aggiornamento da

sys.ZV.reset a **sys.CPU.reset**.

Sappiate che la lista completa di tutti i cambiamenti apportati è contenuta oltre che nella nuova versione del manuale di circa 250 pagine, nel CD allegato al prodotto.

Alcune prove pratiche

Dopo aver visto per sommi capi quanto offerto dal nuovo firmware, abbiamo voluto verificarne l'efficacia e l'affidabilità. Per fare ciò ci siamo serviti di un nuovo script di configurazione, realizzato proprio per questo scopo, in grado di inviare ogni due minuti un messaggio verso l'indirizzo di posta elettronica **mambo-gps@gpstracer.net**, collegato con il server **www.gpstracer.net**. A quanti non fossero ancora a conoscenza di tale server,

diciamo che si tratta di un servizio che Futura Elettronica mette a disposizione di tutti coloro che acquistano un localizzatore satellitare come il Mambo. Configurando opportunamente il localizzatore, questi può

inviare dei messaggi di posta elettronica, specifici per utente e totalmente riservati, che contengono le informazioni di identificazione dell'utente, dell'ora in cui il messaggio è stato inviato e della posizione geografica del localizzatore al momento dell'invio.

Le e-mail vengono salvate sull'unità a disco del server e, all'occorrenza e previa procedura di riconoscimento utente tramite login e password (richieste nella home page di **www.gpstracer.net**), possono essere richiamati per visualizzare sul cartografico di Google Maps la posizione del localizzatore al momento dell'invio del messaggio.

Rispetto agli SMS, le e-mail sono di sicuro più economiche: basti pensare che la connessione GPRS con l'invio di e-mail può costare (a seconda del tipo di contratto) pochi Euro al giorno. Non potendo verificare ogni singolo contratto e gestore, per comodità noi abbiamo utilizzato un contratto Wind, con l'unica limitazione che ci impongono, ossia di avere un account di posta elettronica associato al dominio **libero.it**. Pertanto tutti i comandi che vedrete in seguito sono validi per il gestore Wind; con gestori differenti, deve essere variata la sola parte finale del comando, che rappresenta il campo "parametro del gestore di telefonia

mobile" e posizionato a destra del carattere "=".

Per la configurazione del localizzatore sono necessari pochi parametri: il primo e più importante è l'indirizzo di posta elettronica, che deve avere una struttura del tipo **tuonome@libero.it** e serve per identificare l'account. Oltre all'account di posta, Mambo necessita anche dei parametri SMTP.

Fate ora riferimento alla tabella 1. Il comando riportato nella riga 1 comunica al localizzatore che si deve connettere al server di posta elettronica di Libero (**mail.libero.it**) attraverso la porta 25.

Il comando riportato sulla seconda linea, invece, istruisce il Mambo relativamente all'indirizzo di posta elettronica del mittente. Si tratta di un comando estremamente importante poiché definisce quel parametro che il server di **www.gpstracer.net** utilizza per riconoscere il mittente della e-mail e per memorizzare, in seguito, il messaggio nella cartella assegnata al mittente. Come già accennato, l'indirizzo di posta al quale inviare le e-mail è uguale per tutti (**mambo-gps@gpstracer.net**) mentre il mittente è definito dall'indirizzo di posta elettronica personale.

L'ultimo parametro SMTP da impostare è il gateway su cui deve essere effettuato il "log in": in altre parole, il trasferimento del messaggio all'indirizzo **mambo-gps@gpstracer.net** al server di posta avviene attraverso il gateway di **internet.wind**.

Il comando preposto per questa operazione è riportato nella terza linea della Tabella 1.

Tabella 2 - Parametri di configurazione GPRS dei gestori italiani

PARAMETRO	TIM	VODAFONE	WIND
APN	ibox.tim.it	web.omnitel.it	internet.wind
Nome connessione	TIM	VODAFONE	WIND
Nome utente	Specifico per l'account di posta elettronica.	Specifico per l'account di posta elettronica.	Specifico per l'account di posta elettronica.
Password	Lasciare vuoto.	Lasciare vuoto.	Lasciare vuoto.
Indirizzo IP	Dinamico.	Dinamico.	Dinamico.
DNS primario	192.120.24.30	10.127.1.41	212.245.255.2
DNS secondario	Lasciare vuoto.	Lasciare vuoto.	Lasciare vuoto.

Quanto visto fino ad ora configura il servizio di posta elettronica, ma non è ancora abbastanza: è necessario configurare anche i parametri GPRS.

In particolare occorre comunicare al Mambo l'APN (Access Point Name, ossia "identificazione, nome") del gestore di telefonia, necessario per effettuare la registrazione in rete e per registrarsi automaticamente all'accensione. L'istruzione visibile nella quarta riga della tabella comunica l'APN mentre quella nella quinta riga impone la registrazione in rete automatica. Ricordiamo che per inviare o ricevere la posta attraverso un collegamento cellulare bisogna connettersi all'APN del provider di telefonia attraverso il suo gateway rispettando i parametri del gestore. È evidente che se cambiassimo SIM Card (per esempio TIM o VODAFONE) dovremmo cambiare i parametri rispettando quelli che vengono assegnati dal gestore stesso per l'accesso ad internet via GPRS e per l'invio di posta elettronica. A titolo informativo, in Tabella 2 abbiamo riportato i parametri dei tre gestori più importanti.

Configurazione del Mambo

Il file di configurazione che abbiamo utilizzato per provare il nuovo firmware deriva da quello già visto nell'articolo pubblicato sul fascicolo 119, in cui la Figura 2 riporta il diagramma di flusso completo: in quella circostanza, il localizzatore inviava ogni 5 minuti, a un cellulare prestabilito, un SMS contenente le informazioni "data, ora, posizione e veicolo fermo oppure in movimento con eventuale velocità". Qui abbiamo eliminato gli allarmi relativi all'invio degli SMS in caso di auto ferma e assenza di segnale GPS, mentre sono stati modificati i parametri relativi alla rete GPRS e ai

vari timer; infine è stato cambiato il mezzo utilizzato per inviare il messaggio, che ora è una e-mail al posto di un SMS. Relativamente ai timer, negli articoli precedenti si è parlato largamente di essi: ricorderete che si tratta di procedure molto importanti per l'esecuzione di routine cicliche. Prestate ora attenzione ai tre comandi riportati in Tabella 3, ognuno dei quali si riferisce a un evento di allarme preciso (AL1, AL4, AL6). La prima riga riporta il comando relativo all'allarme AL1 e, tradotta da "linguaggio Mambo" a italiano, significa che, all'accensione, il localizzatore imposta il TIMER_SLEEP a "0", il TIMER_BAT_MODE a "0" e il TIMER_SMS_INVIO a "120" (secondi). Per questo ultimo Timer abbiamo mantenuto lo stesso nome usato nell'articolo pubblicato sul fascicolo 119, dato che oggi come allora, esso scandisce l'invio di un messaggio (anche se ora si tratta di un messaggio di posta elettronica). La seconda riga è più concisa e definisce l'allarme AL4, che riporta a "120" secondi il TIMER_SMS (AL1) quando questi raggiunge il valore "0". Infine, nella terza riga, trovate il comando relativo all'AL6, allarme che comunica al Mambo di inviare all'indirizzo *mambo-gps@gpstracer.net* la stringa NMEA RMC, stringa che viene definita dal valore numerico "8" dopo l'indirizzo di posta. Tenete presente che il file di configurazione



Fig. 1



Fig. 2

completo, scaricabile gratuitamente dal sito della rivista (www.elettronica.it), è molto lungo ma contiene, in linea di massima, gli stessi parametri di configurazione già visti nel corso degli articoli precedenti. Qui ci siamo limitati a commentare solo i nuovi comandi direttamente coinvolti nella gestione del programma di test, programma che deve essere scaricato a bordo del Mambo e che, dopo il reset all'avvio del localizzatore, invia una e-mail ogni due minuti al server di *www.gpstracer.net*.

Il servizio *www.gpstracer.net*

Bene, dopo aver analizzato le tre stringhe di comando più importanti, utilizzate nel nuovo file di configurazione,

Tabella 3 - Comandi principali nel file di configurazione

Comandi di configurazione del timer e dei messaggi	
1	\$PFAL,CNF.Set,AL1=SYS.Device.eStart:SYS.COUNTER_SLEEP_MODE.set=0&SYS.COUNTER_BAT_MODE.set=0&sys.counter_sms_invio.set=120
2	\$PFAL,CNF.Set,AL4=SYS.COUNTER.e_sms_invio:sys.counter_sms_invio.set=120
3	\$PFAL,CNF.Set,AL6=SYS.COUNTER.e_SMS_INVIO&GPS.Nav.sFix=valid:MSG.Send.SMTP,"mambo-gps@gpstracer.net",8,"



Fig. 3

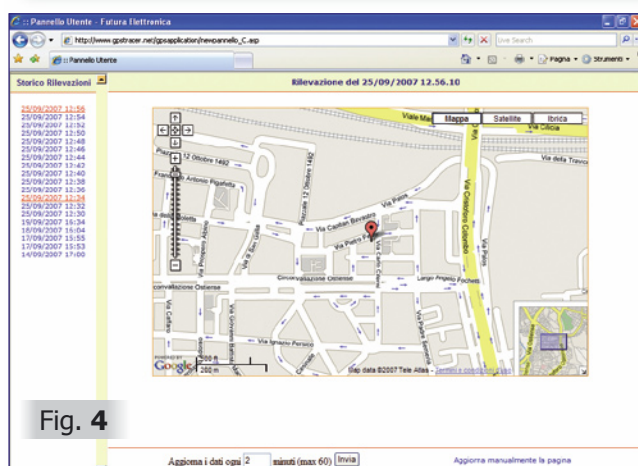


Fig. 4

verifichiamo che Mambo stia funzionando correttamente. Per fare ciò ci avvaliamo di www.gpstracer.net, servizio di localizzazione on-line per accedere al quale è necessario effettuare la registrazione; sappiate che il servizio è gratuito per tutti coloro che acquistano il dispositivo Mambo attraverso il canale di distribuzione ufficiale. La registrazione viene effettuata accedendo alla home page del sito, visibile in Figura 1, alla quale si arriva tramite browser di-

gitando nella barra degli indirizzi www.gpstracer.net. Inserendo nelle due aree di testo i propri dati di riconoscimento (username e password) e cliccando sul pulsante “Accedi”, viene caricata la pagina di Figura 3, in cui è possibile visualizzare e modificare tutti i dati personali. Qualora non siate ancora in possesso dei dati di accesso, li potete richiedere attraverso il collegamento al form di registrazione (Figura 2) presente nella home page (Figura 1). Relativamente alla pagina dei dati personali, notate che è presente una sezione dedicata al localizzatore, sezione in cui è necessario specificare tre parametri fondamentali: il tipo di dispositivo di localizzazione (in questo caso il Mambo),

l'indirizzo di posta elettronica utilizzato per inviare i messaggi al server di www.gpstracer.net, e infine il nome del gestore di telefonia mobile (Wind). Rammentiamo che l'indirizzo di posta elettronica deve coincidere con quello programmato nel Mambo tramite l'istruzione `$PFAL,CNF.Set,TCP.SMTP.FROM=tuonome@libero.it` già vista in tabella 2. A questo punto, non rimane che fare clic sul collega-

mento “Accedi al servizio”: viene visualizzata la pagina di Figura 4, la cui parte di sinistra contiene l'elenco dei messaggi che il localizzatore ha inviato al server, mentre la parte di destra visualizza la mappa. È sufficiente cliccare su uno qualsiasi dei messaggi ricevuti per visualizzare su Google Maps la posizione del localizzatore nel momento in cui il messaggio è stato inviato.

Note conclusive

La volta scorsa, dopo aver terminato l'articolo, ci è stato comunicato la disponibilità di un nuovo firmware. Bene, questa volta è successo di meglio perché ad essere disponibile è un nuovo hardware, la versione 1E2. Nei prossimi giorni effettueremo delle prove quindi vi aggiorneremo sulla situazione. Come anticipazione delle novità, invece, nei prossimi numeri parleremo di “Bolero”, un localizzatore destinato ad applicazioni professionali: esso ripropone tutte le caratteristiche del Mambo, ma dispone anche di ingressi sia digitali che analogici, di uscite digitali, nonché di un telecomando per il controllo a breve distanza e di un modulo di I/O bidirezionale, entrambi comunicanti attraverso un protocollo di trasmissione ISM a 2,4 GHz. Di questi nuovi prodotti, e dei relativi esempi pratici, pubblicheremo articoli specifici. Se volete proseguire il viaggio nell'affascinante mondo del GPS e dei suoi campi di applicazione, continuate a seguirci. Per intanto, arriverci al mese prossimo, puntuali in edicola.

per il

MATERIALE

Il sistema di localizzazione satellitare GPS/GSM/GPRS Bluetooth Mambo (cod. Mambo55) può essere acquistato al prezzo di 510,00 Euro, IVA inclusa. La confezione include, oltre al localizzatore, il pacco batterie, un auricolare, un caricabatterie con adattatore sia di rete che da accendisigari per auto, un laccio per il trasporto al collo ed il CD contenente i manuali in inglese e l'ultima versione di firmware disponibile al momento dell'acquisto.

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

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