



CORSO ZigBee[®]

in collaborazione con
AUR'EL

di FRANCESCO FICILI

Si sta facendo strada nel panorama dei dispositivi per reti wireless a breve distanza: è ZigBee, un protocollo di comunicazione che ha destato l'interesse di tanti costruttori. Scopriamolo insieme. Prima puntata.

Con il termine ZigBee viene definito il protocollo di comunicazione di alto livello, basato sullo standard IEEE 802.15.4, creato per l'implementazione e la standardizzazione di reti Wireless PAN (Personal Area Network) a bassa velocità di comunicazione. Il protocollo ZigBee (a volte indicato anche come stack ZigBee, o standard ZigBee) prevede lo scambio di dati nelle frequenze radio assegnate per scopi industriali, scientifici e medici (indicate genericamente con il termine ISM) a 433 o 868 MHz in Europa, 315 o

915 MHz negli Stati Uniti e 2,4 GHz a diffusione praticamente mondiale. ZigBee nasce come alternativa, più semplice ed economica, ad altri tipi preesistenti di standard per WPANs, come, per citarne due tra i più diffusi, Bluetooth e Wi-Fi. Lo standard prevede comunicazioni a bassa velocità, realizzate con trasmettitori di bassa potenza, oltre alla creazione di topologie di rete anche molto complesse e ad elevato numero di nodi. Una delle caratteristiche peculiari delle WPAN ZigBee è la possibilità di garantire la comunicazione tra nodi



che non si “vedono” direttamente, sfruttando il mesh networking. Tipici target applicativi sono le applicazioni embedded che richiedano transfer-rate non elevati e consumi ridotti. La finalità dello standard è quella di definire una tipologia di rete wireless non mirata, economica e autogestita, che possa essere utilizzata per scopi quali il monitoraggio e il controllo industriale, le reti di sensori, la domotica, ecc.

L’aggiornamento, la revisione e la pubblicazione dello standard, sono affidati alla ZigBee Alliance, un consorzio di aziende (il cui logo è illustrato nella Fig. 1) del settore telecomunicazioni ed elettronica. La specifica ZigBee 1.0 è stata approvata il 14 dicembre 2004 ed attualmente la ZigBee Alliance ha iniziato a lavorare sulla versione 1.1.

I membri della ZigBee Alliance sono divisi in tre livelli: Promoters, Participants ed Adopters. Attualmente l’adesione a livello Adopters costa a un’azienda circa 3500 dollari all’anno e permette l’accesso alle specifiche e al relativo uso per scopi commerciali. Al giorno d’oggi si contano circa 280 aziende affiliate alla ZigBee Alliance e tra i promotori figurano società del calibro di ST Microelectronics, Samsung, Texas Instruments, Philips, Motorola, Freescale, Siemens ed altre ancora. Tra i Participants figurano aziende del

ZigBee Specification

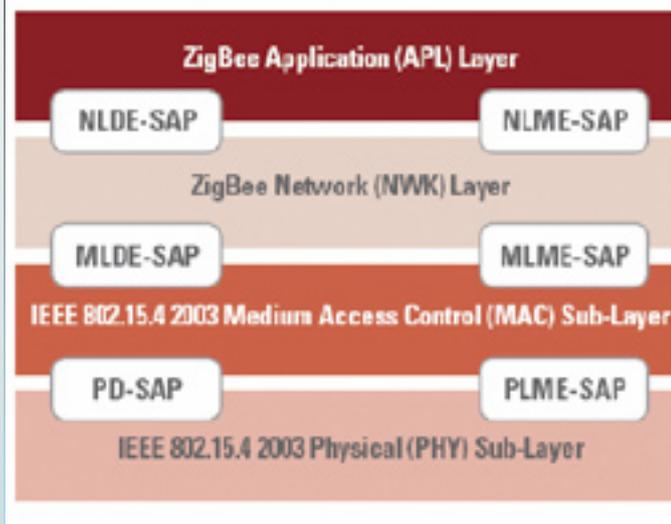


Fig. 2 - Rappresentazione dello standard ZigBee secondo il modello ISO/OSI

settore elettronico e produttori di microcontrollori come Microchip Technology, Atmel, Fujitsu, Renesas, Telit, ma anche aziende di settori non strettamente connessi con quello delle telecomunicazioni, come ad esempio FedEx e Black&Decker.

DESCRIZIONE DELLO STANDARD

Per analizzare lo standard ZigBee partiamo dalla sua “carta d’identità”, ovvero, parlando in termini di protocolli di comunicazione, dalla sua rappresentazione secondo il modello ISO/OSI, una cui schematizzazione semplificata è visibile nella Fig. 2.

Come si vede, il livello Fisico ed il livello MAC fanno parte della specifica 802.15.4; la specifica ZigBee aggiunge altri due livelli, che sono un livello Network (NWK) ed un livello

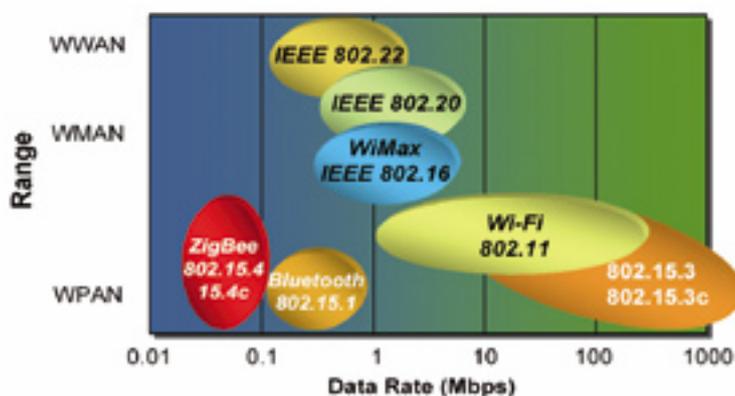
Application (APL). I riquadri in bianco identificano i SAP (Service Access Point), che costituiscono i punti di accesso e comunicazione tra due layer (livelli o strati) contigui.

Prima di addentrarsi nella descrizione dei vari strati che lo compongono, è opportuno fare un piccolo inciso sulle tipologie di dispositivi ammessi dallo standard ZigBee, in modo da poter meglio comprendere i paragrafi che seguiranno. Lo standard ZigBee ammette tre differenti tipologie di dispositivi.



Fig. 1 - Logo della ZigBee Alliance.

Confronto tra le più diffuse tecnologie wireless



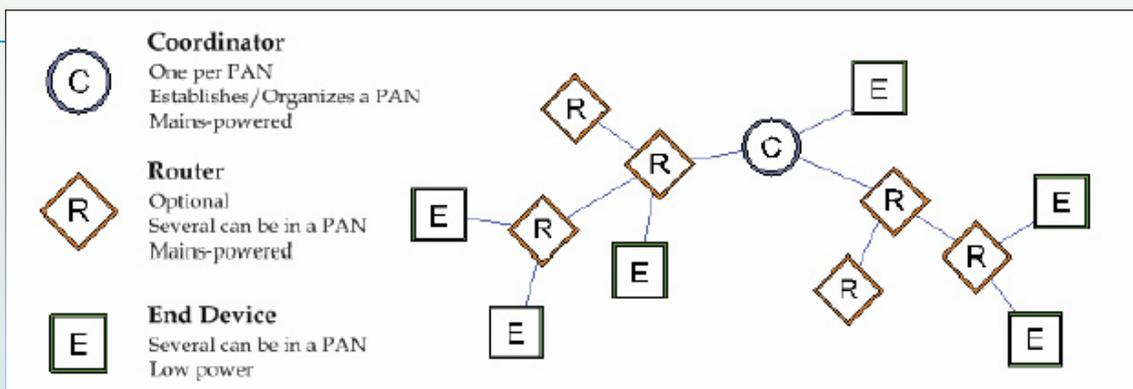


Fig. 3
 Esempio di rete ZigBee.

- **ZigBee Coordinator:** è il dispositivo più "intelligente" tra quelli disponibili, costituisce la radice di una rete ZigBee e può operare da ponte tra più reti. Ci può essere un solo Coordinator in ogni rete. Assolve funzioni vitali, come la creazione della rete stessa, la selezione del canale RF e l'assegnazione degli identificativi. È, inoltre, in grado di memorizzare informazioni sulla rete creata e può agire come deposito per le chiavi di sicurezza. Un malfunzionamento di questo dispositivo può ripercuotersi sul resto della rete o addirittura causarne il blocco.
- **ZigBee Router:** agisce come router intermedio, passando i dati da e verso altri dispositivi. In una PAN può essercene più di uno; esso dispone di un certo livello di intelligenza, in quanto implementa complicati algoritmi di routine per la selezione dei percorsi migliori da e verso un determinato nodo. Un malfunzionamento di uno di questi dispositivi non pregiudica il funzionamento di tutta la rete, ma può renderne inutilizzabile una parte.
- **ZigBee End Device:** questo dispositivo include solo le funzionalità minime per dialogare con il loro nodo parente (Coordinator o Router); può essercene più di uno e non può trasmettere dati provenienti da altri dispositivi. I ZigBee End Device sono i nodi che richiedono il minor quantitativo di memoria e quindi risultano spesso più economici rispetto ai Router ed ai Coordinator.

Un esempio di una semplice rete ZigBee nella quale sono presenti tutti e tre i tipi di dispositivi è illustrato nella Fig. 3.

Passiamo ora alla descrizione dei vari strati che compongono lo stack ZigBee, partendo dal basso verso l'alto e servendoci della Fig. 4, che non è altro se non una rappresentazione più dettagliata, nella quale sono evidenziati anche interfacce e sublayer, dell'architettura dello stack ZigBee.

Livello Fisico

Il livello fisico è definito dallo standard IEEE 802.15.4, ed ha il compito di fornire l'interfaccia tra il canale trasmissivo ed il livello MAC dello standard. Secondo le specifiche, il livello fisico deve essere in grado di fornire i seguenti servizi:

- attivazione e disattivazione del transeiver;
- rilevamento dell'energia del canale in uso;
- rilevamento della qualità del collegamento;

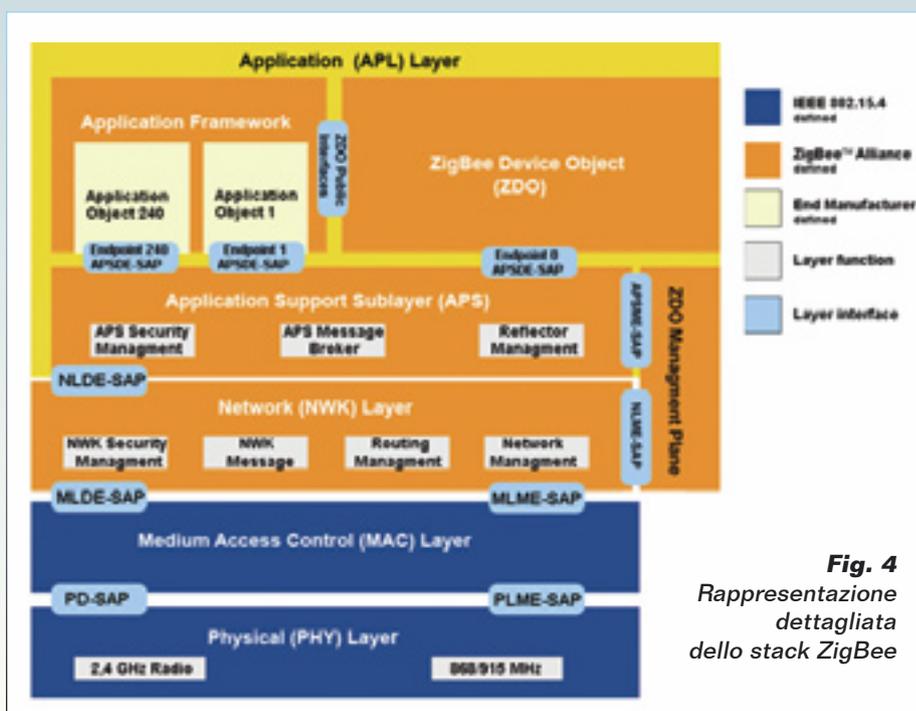


Fig. 4
 Rappresentazione dettagliata dello stack ZigBee

	Frequenza operativa [GHz]	Potenza RF	Interfaccia di comunicazione	Antenne disponibili	Package	Note
Aurel	2,4	1mW (moduli xLI) 20mW (moduli xHI)	UART	Chip o esterna	Pin Strip passo 2mm (modulo)	Software demo dedicato. OCX dedicato
Microchip	2,4	1,26mW	SPI	Chip	QFN 40 pin	Supporto per 802.15.4 MAC e AES encryption
Digi	2,4	1mW (moduli standard) 100mW (moduli PRO)	UART	Filo, chip, U.FL e RPSMA	Pin Strip passo 2mm (modulo)	Supportate modalità di rete avanzate e a basso consumo
Atmel	2,4	2mW	SPI	Chip	QFN 40 pin	
Jennic	2,4	3mW	UART, SPI	Chip	QFN 56 pin	Supporto per 802.15.4 e AES encryption

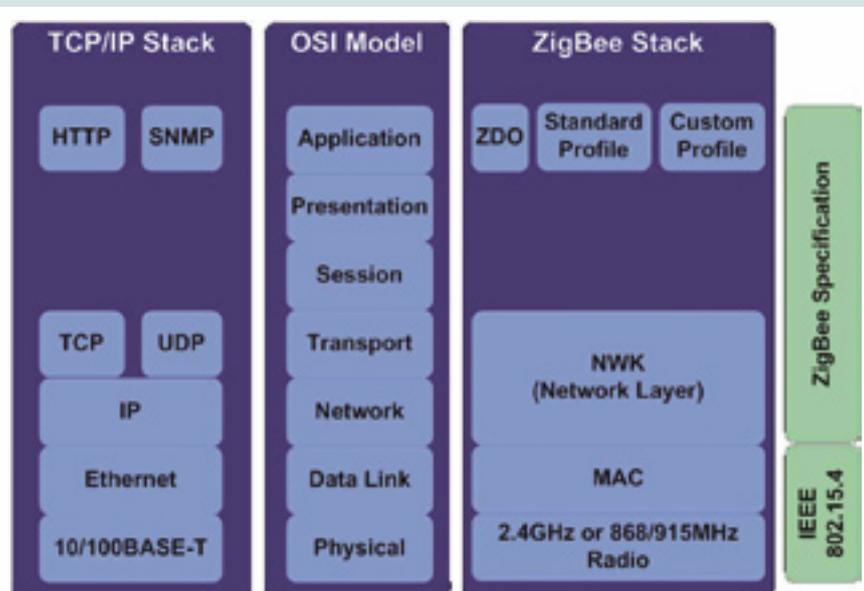
Tabella 1
Cross reference Moduli ZigBee

- selezione della frequenza di comunicazione;
- fornitura dei servizi per l'accesso al canale trasmissivo con la tecnica del CSMA/CA;
- trasmissione e ricezione dei dati.

Il livello fisico è concettualmente costituito da due entità distinte, la prima delle quali è un'entità di comando generalmente chiamata PHY Layer Management Entity (PLME) che costituisce la vera e propria interfaccia con il livello MAC; la seconda è invece un'entità deputata esclusivamente al trasferimento dei dati e viene generalmente indicata con il nome di PHY Layer Data Entity Service (PLDE). Entrambe le entità possiedono un Service Access Point dedicato, che viene utilizzato dal livello MAC per gestire l'intero livello Fisico e comunicare con esso; i due SAP

sono rispettivamente il PHY Layer Management Entity Service Access Point (PLME-SAP) ed il PHY Data Service Access Point (PD-SAP). Un discorso a parte lo merita la tecnica di accesso al canale trasmissivo, che secondo le specifiche deve essere di tipo CSMA/CA. CSMA/CA è l'acronimo inglese di Carrier Sense Multiple Access with Collision Avoidance, ovvero accesso multiplo tramite ascolto del canale con elusione delle collisioni. Nel momento in cui una stazione vuole tentare una trasmissione, ascolta il canale (Listen-before-Transmit). Se lo trova occupato, la stazione attiva un timer di durata casuale (detto tempo di backoff) che viene decrementato solo durante i periodi di inattività del canale. Quando il timer scatta, la stazione fa un altro tentativo.

Rappresentazione OSI degli stack Ethernet e ZigBee



Ciò permette di evitare, su base statistica, che più nodi tentino di accedere al canale contemporaneamente.

Si è scelto di utilizzare questa tecnica perché nelle reti wireless non è possibile rilevare le collisioni, come si fa nel CSMA/CD (Carrier Sense Multiple Access with Collision Detection), motivo per cui le collisioni devono essere evitate piuttosto che rilevate.

Livello MAC

Anche il livello Medium Access Control (MAC) come il livello Fisico, è definito dallo standard IEEE 802.15.4 e secondo le specifiche deve essere in grado di fornire i seguenti servizi:

- supporto all'avvio ed al man-



- tenimento della PAN;
- indirizzamento dei dispositivi;
- esecuzione dell'algoritmo CSMA\CA per l'accesso al canale trasmissivo;
- allocazione dei GTS (Guaranteed Time Slot);
- supporto alla sicurezza dei pacchetti trasmessi.

Come quello Fisico, anche il livello MAC può essere concettualmente diviso in due entità, ossia il MAC Sublayer Management Entity (MLME), che fornisce il servizio di gestione del livello e delle interazioni con il resto dello stack (come il supporto alla creazione di una PAN) ed il MAC Sublayer Data Entity Service (MLDE) che si occupa, invece, del trasporto dei dati. I servizi forniti sono accessibili da due SAP: all'MLDE si accede attraverso il MAC Common Part Sublayer Service Access Point (MCPS-SAP), mentre l'MLME è accessibile tramite il MAC Sublayer Management Entity Service Access Point (MLME-SAP). Relativamente all'indirizzamento, è importante far notare che ogni dispositivo ZigBee è univocamente identificato da un numero di serie a 64 bit, che viene scritto in una ROM interna al chip e che permette così di distinguerlo in tutto il mondo da qualsiasi altro dispositivo della stessa famiglia.

Livello Network

Il livello Network (NWK) del protocollo è definito direttamente dalla ZigBee Alliance. I suoi compiti sono di eseguire il routing dei pacchetti dalla sorgente alla destinazione, eseguire l'associazione e la dissociazione di un nodo da una network, applicare gli algoritmi di sicurezza e, nei coordinatori, inizializzare la rete ed assegnare gli indirizzi. Anche il livello Network, come i precedenti, è formato da due entità: la Network Layer Data Entity (NLDE) e la Network Layer Management Entity (NLME). La NLDE fornisce il servizio di trasporto dei pacchetti ed applica la

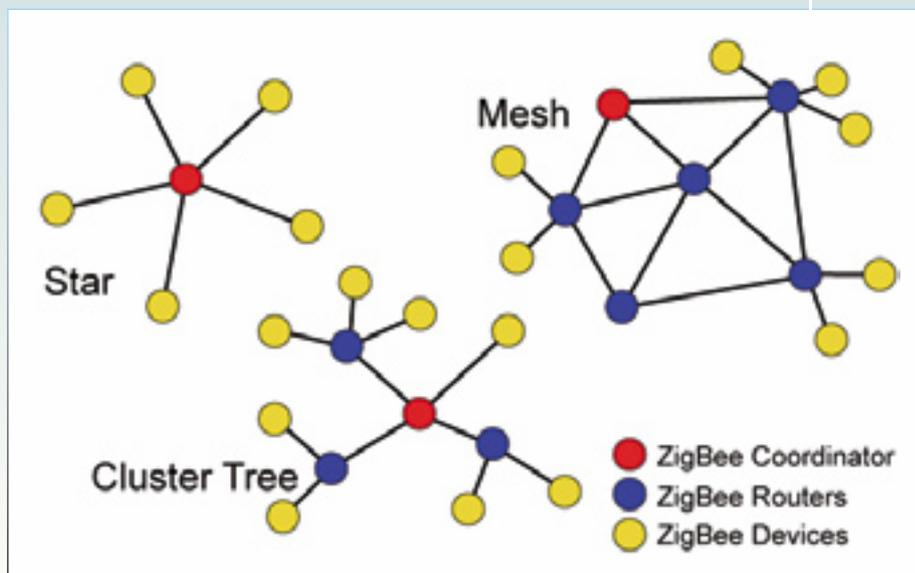
sicurezza; i suoi compiti sono essenzialmente la generazione dei pacchetti dati a livello Network, il routing e l'applicazione della sicurezza sui dati. Il livello Application può accedere ai servizi forniti dalla NLME attraverso il Network Layer Data Entity Service Access Point (NLME-SAP). La NLME fornisce invece servizi volti a garantire ad una applicazione la possibilità di interagire correttamente con lo stack ZigBee. La fornitura di questi servizi al livello superiore è garantita dal Network Layer Management Entity Service Access Point (NLME-SAP).

A livello Network sono definite essenzialmente tre diverse topologie di rete: Star, Mesh e Cluster Tree. La Star Network è costituita attorno ad un PAN Coordinator al quale è collegato un numero variabile di dispositivi figli, i quali possono essere sia Router che End Device. Nel caso di topologia cluster tree il coordinatore è collegato a stella ad una serie di Router ed End Device; i Router, a loro volta, ripropongono una topologia a stella. Infine, nella topologia Mesh ogni nodo è collegato a tutti quelli che si trovano nel suo raggio di copertura. Questa topologia viene anche chiamata rete peer-to-peer. Nella Fig. 5 sono rappresentate graficamente le tre topologie di rete previste dallo standard.

Livello Application

Il livello Application dello standard viene definito direttamente dalla ZigBee Alliance.

Fig. 5
Topologie di rete previste dallo standard ZigBee.



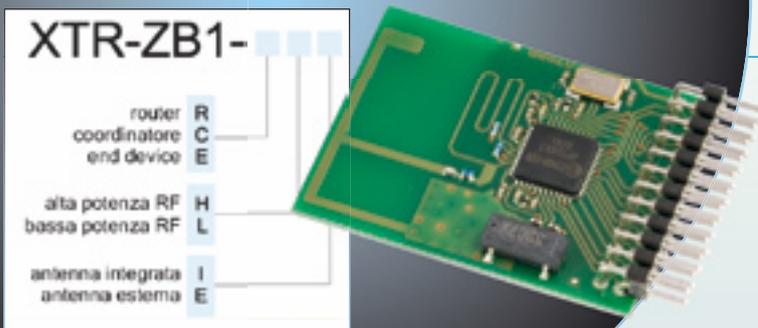


Fig. 6 Modulo ZigBee prodotto da Aurel.

Il layer corrispondente è formato dall'Application Support Sublayer

(APS), dallo ZigBee Device Object (ZDO) e dagli Application Objects implementati dal costruttore del chip in questione.

L'Application Support Sublayer fornisce due interfacce, che sono l'APS Management Entity Service Access Point (APSME-SAP) e l'APS Data Entity Service Access Point (APSDESAP). La prima di queste interfacce viene utilizzata per applicare la sicurezza, mentre la seconda è usata dagli Application Objects e dagli ZDO per la trasmissione dei dati.

Gli Application Object sono definiti dal costruttore del chip e costituiscono le applicazioni utente che si ritrovano al livello più alto dello stack; risiedono su endpoint numerati da 1 a 240. Per essere conformi allo standard, devono aderire a determinati profili approvati dalla ZigBee Alliance. Gli endpoint, insieme all'indirizzo del dispositivo, forniscono un sistema per indirizzare in maniera uniforme ed univoca su tutta la rete i singoli Application Object (ossia le applicazioni finali) all'interno di un dispositivo.

Il dualismo che esiste tra endpoint ed indirizzi ZigBee è simile a quello che viene individuato dalla coppia indirizzo IP del server/numero di porta TCP in una rete TCP/IP.

Gli ZigBee Device Objects (ZDO) forniscono un'interfaccia

verso gli Application Objects usati per scoprire altri dispositivi e servizi forniti da questi. Lo ZDO è un Application Object speciale, che risiede sull'endpoint 0.

SOLUZIONI ESISTENTI

Attualmente, per l'implementazione di reti ZigBee, esistono in commercio diverse soluzioni hardware; vediamo le caratteristiche di alcune tra le più diffuse.

Aurel

Aurel Wireless propone due moduli in grado di operare su rete standard ZigBee in maniera efficiente, sicura e con possibilità di controllo da parte del coordinatore di rete dell'integrità e funzionalità della stessa. La prima serie di moduli è individuata dalla sigla XTR-ZB1 e, come si vede nella Fig. 6, le schede sono differenziate per tipologia di dispositivo (Coordinatore, Router, End Device), per la massima potenza irradiata (High Power e Low Power) e per la tipologia di antenna (interna ed esterna) prevista.

Alcune caratteristiche di questi moduli sono: possibilità di cambio automatico del canale RF in uso in presenza di disturbi, scambio dati tipo mesh tra dispositivi della stessa rete utilizzando stringhe ASCII con indirizzo più dati, trasmissione dati codificata (128 bit di cifratura) automatica e trasparente all'utente, set di comandi ASCII per l'utilizzo a basso livello dei moduli.

Sono basati sul system-on-chip Texas Instruments cc2430, un potente chip che contiene al suo interno un transceiver ZigBee associato ad un microcontrollore 8051, con tagli di flash che arrivano fino a 128 kbyte. La quantità di memoria è sufficiente ad ospitare l'intero stack ed un'applicazione utente firmware. Nel caso specifico, Aurel ha realizzato il firmware per l'implementazione di nodi Coordinator, Router ed End Device, più un application layer che permette di gestire la UART, il convertitore A/D, ingressi ed uscite digitali, interrupt ed altro ancora. Volendo espanderne le funzionalità, può essere associato ad un qualsiasi microcontrollore dotato di porta USART.

Inoltre, per permettere un semplice approccio ai suoi moduli wireless ZigBee, Aurel mette a disposizione un completo kit di sviluppo che

Fig. 7 Starter kit ZigBee prodotto da Aurel



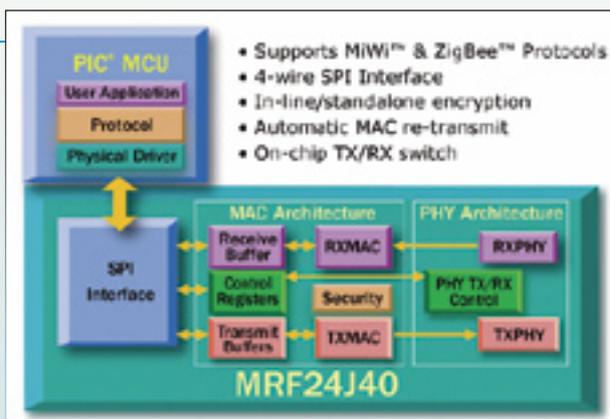


Fig. 8 - Schema a blocchi del modulo MRF24J40.

permette di testare tutte le funzionalità dei dispositivi.

Il kit (visibile nella Fig. 7) comprende una scheda USB per l'interfacciamento del PAN coordinator al PC e 4 demoboard predisposte per la connessione con altrettanti router; sono altresì inclusi tutta la documentazione necessaria, i driver dell'interfaccia USB e una serie di tool software per lo sviluppo di applicazioni.

Microchip

La Microchip's Radio Frequency Products Division, divisione di Microchip dedicata ai prodotti RF, offre diverse soluzioni per lo sviluppo di sistemi wireless a corto raggio, tutti basati sul chip MRF24J40. Questo dispositivo è un transceiver wireless a 2,4 GHz a standard IEEE 802.15.4, pensato per l'utilizzo con protocolli di alto livello come ZigBee.

Nella Fig. 8 è possibile vedere lo schema a blocchi del transceiver MRF24j40 nella sua applicazione tipica, ossia l'interfacciamento con un microcontrollore PIC di Microchip. A supporto dei progettisti, Microchip Technology fornisce una serie di stack software adatti all'implementazione su microcontrollori PIC. Le soluzioni proposte da Microchip sono le seguenti:

- ZigBee;
- MiWi;
- MiWi P2P.

Lo stack ZigBee ha una occupazione di memoria programma che può arrivare sino a 96 kbyte (escluso il codice utente), quindi è limitato a quei microcontrollori PIC aventi tagli di memoria adeguati. Sono supportate le topologie a stella e a mesh, mentre come target si hanno a disposizione i microcontrollori di

Tabella 2 - Caratteristiche degli stack Microchip.

Stack	Topography	Hops	Memory	PIC18	PIC24	PIC32
MiWi™ P2P (Proprietary)	Star/P2P	1	3K-9K	SI	SI	SI
MiWi (Proprietary)	Star/Mesh	4	7K-17K	SI	SI	SI
Zigbee (Certified)	Star/Mesh	Infinite	40K-96K	SI	SI	NO



Fig. 9 - Il modulo MRF24J40MA.

casa Microchip ad 8 e 16 bit (PIC24 e dsPIC); attualmente non sono supportati i PIC32. Lo stack MiWi\MiWi P2P non è altro che una versione semplificata dello stack ZigBee: il footprint del codice si riduce a 7÷17 kbyte (3÷9 Kbyte per MiWi P2P) e risulta quindi adeguato per applicazioni cost-sensitive, che utilizzano microcontrollori con limitate quantità di memoria flash. Sono supportati microcontrollori PIC18, PIC24, dsPIC30/33 e anche i recenti PIC32.

Nella Tabella 2 sono elencate le caratteristiche più importanti dei tre stack.

Il chip MRF24j40 si presenta con un package di tipo QFN a 40 pin, quindi poco adatto alla prototipazione; ecco perché Microchip ha messo in commercio un modulo RF basato su questo chip, con già a bordo tutta l'elettronica necessaria al funzionamento (antenna inclusa) e un agevole collegamento verso l'esterno. Il modulo prende il nome di MRF24j40MA ed è raffigurato nella Fig. 9.

Digi-Maxstream

La Digi-Maxstream, azienda specializzata nella realizzazione di soluzioni embedded, ha sviluppato una serie di moduli radio compatibili con le specifiche dello standard ZigBee. I moduli radio, che sono stati battezzati con il nome di Xbee RF modules, presentano una forma adatta alla prototipazione (due file di pin strip a passo 2 mm) e Digi mette a disposi-

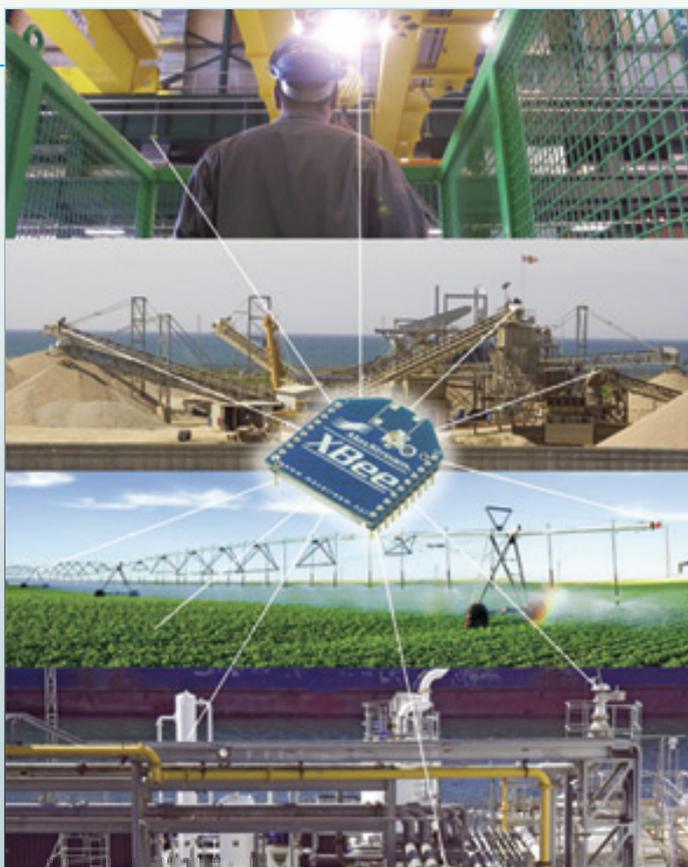


Fig. 10 Scenari applicativi dei moduli Xbee prodotti da Digi.

denominata X-CTU, attraverso cui è possibile programmare i moduli, effettuare il range test e dialogare con i moduli stessi attraverso un terminale dotato di interfaccia seriale.

Sono inoltre a disposizione varie versioni di firmware per l'implementazione di reti ZigBee. In particolare, è possibile realizzare sistemi di comunicazione wireless sia point-to-point che mesh. Per la soluzione point-to-point possono essere utilizzati i moduli denominati "serie 1", più economici ma con prestazioni inferiori. Se invece si vuole realizzare una mesh, è necessario utilizzare i moduli serie 2 (che sono in grado di supportare i nuovi firmware per implementazione di ZigBee mesh, Znet 2.5 e ZB).

Atmel

Anche Atmel, parallelamente alla concorrenza, propone una sua soluzione wireless ZigBee. In questo caso i moduli prendono il nome di Z-Link e sono ottimizzati per l'utilizzo con piattaforme di tipo AVR.

Nella Fig. 12 è riportato un possibile schema a blocchi del sistema.

Un ottimo strumento per lo sviluppo di sistemi wireless basati sulla soluzione proposta da Atmel è lo starter kit RZ200, visibile nella Fig. 13.

Jennic

La Jennic è un'azienda specializzata nella produzione di soluzioni wireless on-chip; la sua vasta esperienza nel settore della connettività senza fili ha portato alla realizzazione di

Whip Antenna



Chip Antenna



U.FL.RF Connector



Fig. 11 Alcune possibilità per la scelta dell'antenna per i moduli prodotti da Digi.

zione una ricca dotazione di librerie firmware sia per soluzioni point-to-point che per l'implementazione di reti mesh.

La soluzione proposta da Digi presenta caratteristiche interessanti per gli hobbisti, in quanto i moduli possono essere facilmente montati su schede elettroniche per prototipazione o PCB. Il passo dei pin strip per la connessione del modulo non è standard, ma esistono in commercio delle schede di adattamento. L'interfaccia di comunicazione è di tipo seriale UART,

quindi facilmente interfacciabile ad un qualsiasi microcontrollore (PIC, AVR, ARM, ecc). Uno dei pregi di questa soluzione è rappresentato dalle varie possibilità nella scelta dell'antenna (Fig. 11).

Per quanto riguarda il supporto offerto ai progettisti, Digi mette a disposizione un'interfaccia software per la programmazione ed il debug,

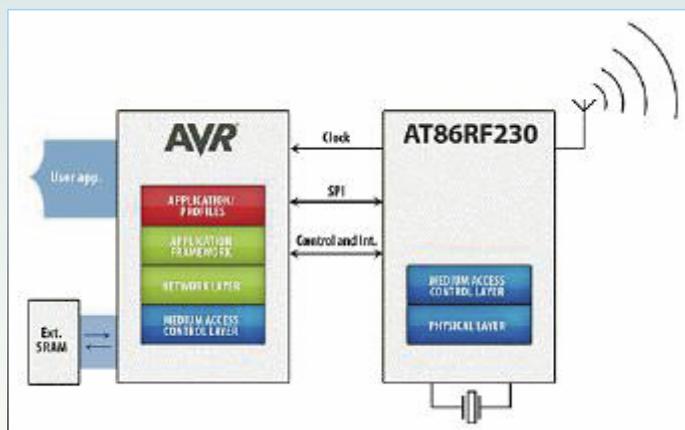
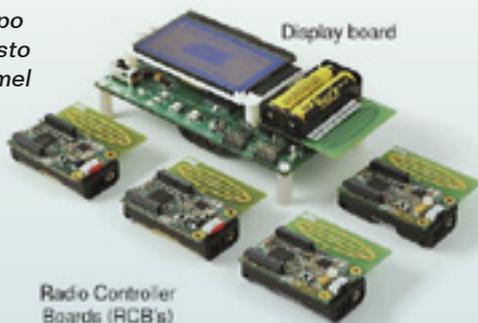


Fig. 12 - Schema a blocchi del modulo AT86RF230.

Fig. 13
Il sistema di sviluppo proposto da Atmel



un microcontrollore dotato di interfaccia IEEE 802.15.4, e quindi adatto a supportare lo stack ZigBee.

Nella Fig. 14 è rappresentato lo schema a blocchi del JN5121.

Questo versatile chip integra un microcontrollore a 32 bit ed una sezione di radiofrequenza operante a 2,4 GHz, fino a 96 kbyte di RAM e 64 kbyte di Flash. La dotazione di periferiche include Timer, ADC, 21 GPIO ed interfacce di comunicazione UART ed SPI.

La soluzione proposta da Jennic fornisce un completo sistema on-chip e può essere una buona scelta nel caso di applicazioni cost-sensitive che non richiedano prestazioni particolarmente spinte.

Un ottimo strumento per iniziare ad utilizzare i moduli realizzati dalla Jennic è l'IEEE 802.15.4 & ZigBee Evaluation Kit, rappresentato nella Fig 15.

Concludiamo qui la prima puntata del corso, abbiamo fatto una panoramica dello stack ZigBee ed analizzato alcune tra le soluzioni più interessanti che propone il mercato. Come si è visto, nonostante lo standard sia di recente concezione, per la sua implementazione in commercio esistono già diverse soluzioni, ognuna con caratteristiche proprie, vantaggi e svantaggi.

Nella prossima puntata concentreremo la nostra attenzione sul kit di sviluppo Aurel, analizzando in dettaglio i vari elementi hardware che lo compongono ed il software demo tramite il quale vi verrà fornita la possibilità di gestire la vostra prima rete ZigBee. ■

Fig. 14 - Schema a blocchi del JN5121

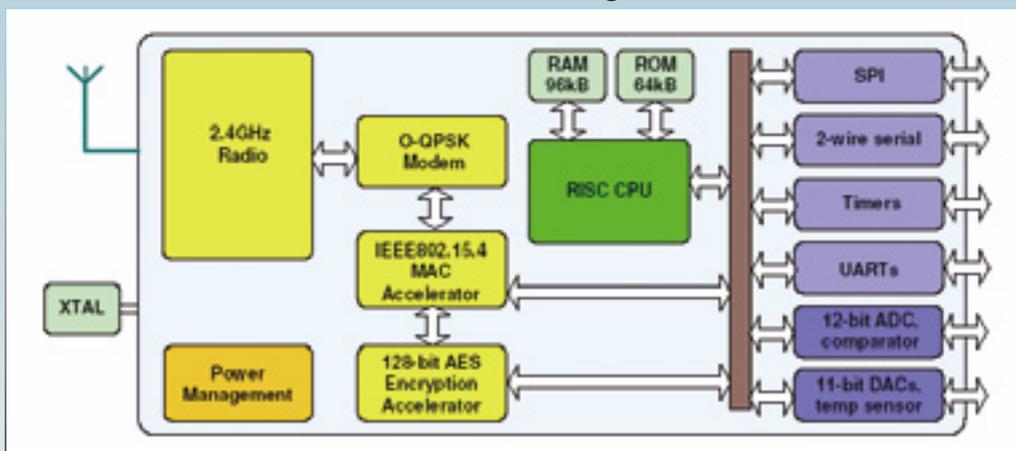


Fig. 15
Evaluation kit della Jennic.





CORSO ZigBee[®]

in collaborazione con
AUREL

di FRANCESCO FICILI

Delineate le principali peculiarità dello standard e viste le specifiche tecniche, approfondiamo l'argomento studiando, tra i dispositivi proposti dalle Case del settore, quelli Aurel. Seconda puntata.

Nella prima puntata abbiamo iniziato a descrivere il protocollo ZigBee, definendone origine e struttura, spiegando come è nato e ad opera di chi e in che ambito trova applicazione; abbiamo dedicato qualche paragrafo anche a una panoramica sui prodotti e i relativi sistemi di sviluppo resi disponibili dai produttori che sono impegnati in prima linea nel settore. Tra tutti, abbiamo scelto quelli dell'Aurel, che analizzeremo in dettaglio, insieme ai componenti dello starter kit che fungerà da base prototipale durante il corso. Verrà

inoltre analizzato il software client fornito a corredo del kit ed il componente OCX, che permette la creazione di applicazioni software personalizzate impieganti i moduli ZigBee Aurel. Prima di descrivere le caratteristiche e i componenti dello starter kit, facciamo un breve riepilogo delle principali caratteristiche della soluzione Aurel.

ZIGBEE AUREL

La soluzione ZigBee proposta da Aurel consiste in una serie di moduli wireless basati sul system-



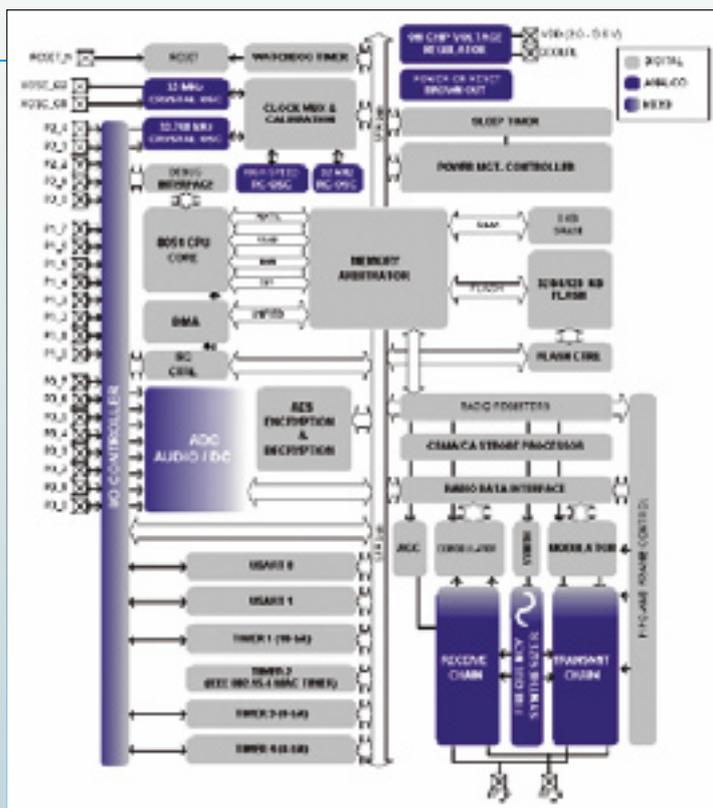


Fig. 1
Schema a blocchi del cc2430.

on-chip cc2430 Texas Instruments. Questo potente e versatile chip integra un transceiver radio compatibile con le specifiche IEEE 802.15.4, un microcontrollore RISC (core 8051) che può avere fino a 128 kbyte di Flash EPROM, 8 kbyte di RAM e una serie di periferiche tra le quali si segnalano un convertitore A/D a 12-bit, un coprocessore per crittografia AES, due porte USART con supporto per svariati protocolli seriali, supporto per MAC address a 64-bit, 21 General Purpose I/O ed un sensore di temperatura integrato. Il tutto è contenuto in un package QLP a 48 pin, di soli 7x7mm. In Fig. 1 vedete lo schema a blocchi del cc2430.

Basandosi sul system-on-chip cc2430, Aurel ha costruito la serie di moduli ZigBee XTR-ZB1, che esistono nelle versioni (a seconda del firmware caricato nella flash del chip) Coordinator, Router ed End Device, e si differenziano per potenza del trasmettitore e tipologia di antenna (esterna o interna), come mostra la Tabella 1. I moduli utilizzati durante questo corso sono quelli a bassa potenza trasmittiva e antenna integrata, quindi CLI, RLI ed ELI. Vediamo adesso quali sono le caratteristiche dell'implementazione di

Aurel dei tre dispositivi che costituiscono una rete ZigBee.

- Coordinator:** anche indicato come PAN Coordinator, è l'elemento a maggiore capacità di calcolo di tutta la rete. Per ogni rete è possibile la presenza di un solo Coordinator, che agisce in fase di creazione della rete svolgendo i compiti di scelta del canale di trasmissione e accettazione dei vari dispositivi che chiedono di associarsi. Il Coordinator definisce anche il PANID, che è un numero identificativo della rete in costruzione e che verrà utilizzato da tutti i membri come unico identificativo valido di appartenenza. Inoltre può essere visto come elemento di supervisione e controllo della rete; è questo il compito per il quale l'Aurel suggerisce di utilizzarlo. Viene pertanto fornito un software di interfaccia con il PC (di tipo OCX, inseribile come componente nel programma utente), tramite il quale si potrà utilizzare il Coordinator come punto di raccolta di tutti i dati trasportati dalla rete. Inoltre l'Aurel ha previsto una funzione di cambio automatico del canale in uso in presenza di disturbi, che agisce esclusivamente in presenza del Coordinator.
- Router:** Il Router è un dispositivo considerato sempre alimentato; in fase di creazione della rete gestisce l'associazione ad essa dei dispositivi che non sono direttamente raggiungibili dal Coordinator, attribuendo indirizzi short (16 bit) a tutte le connessioni che configura. Nelle reti ZigBee. Ad ogni Router (o Coordinator) si può associare un numero massimo di dispositivi, che, nella rete ZigBee Aurel, è stato fissato in 20 (i dispositivi possono essere altri Router).
- End Device:** Gli End Device sono i dispositivi terminali. Per far parte di una determinata rete, devono essere associati ad un dispositivo "padre" (un Router o un Coordinator), tramite il quale comunicano con il resto della rete. Caratteristica peculiare

Tabella 1 - Moduli ZigBee Aurel

Tipo di dispositivo	Antenna integrata	Antenna esterna	Antenna integrata	Antenna esterna
	Bassa Potenza Trasmittiva		Alta Potenza Trasmittiva	
Coordinator	XTR-ZB1-CLI	XTR-ZB1-CLE	XTR-ZB1-CHI	XTR-ZB1-CHE
Router	XTR-ZB1-RLI	XTR-ZB1-RLE	XTR-ZB1-RHI	XTR-ZB1-RHE
End Device	XTR-ZB1-ELI	XTR-ZB1-ELE	XTR-ZB1-EHI	XTR-ZB1-EHE



di questi dispositivi è quella di poter essere posti in "sleep", il che limita il consumo di energia a quello necessario al real-time clock. Questa caratteristica delle reti ZigBee permette di remotizzare facilmente dei sensori che si connettono alla rete con cicli dell'ordine dei secondi, restando in stato attivo per intervalli di tempo dell'ordine dei millisecondi; il tutto permette di alimentare questi dispositivi a batteria anche per lunghi periodi di tempo.

PRESENTAZIONE DELLO STARTER KIT AUREL

Lo starter kit proposto da Aurel è lo strumento ideale per un primo approccio alla connettività wireless ZigBee. Il kit dispone di tutti gli elementi hardware e software necessari alla realizzazione di una piccola WPAN, composta da un Coordinatore e quattro nodi, che possono essere sia Router che End device. Il contenuto dello starter kit commercializzato dall'Aurel è il seguente:

- 1 Demo Board XTR-ZB1 coordinator USB;
- 1 Modulo XTR-ZB1-CLI coordinator;
- 4 Demo Board XTR-ZB1 router;
- 4 Moduli XTR-ZB1-RLI router;
- CD contenente software, driver, e documentazione;
- Manuale d'uso e cavo USB per la connessione;

Vediamo adesso di dare una descrizione più dettagliata dei vari componenti del kit.

La Demoboard USB

La demoboard XTR-ZB1 Coordinator (Figura 2) è stata realizzata per permettere di utilizzare i moduli Radio XTR-ZB1 nella versione coordinatore. La scheda è equipaggiata con un convertitore USB/Seriale prodotto dalla FTDI e tutta la circuiteria necessaria al suo funzionamento. Nella scheda trova posto anche una strip adatta alla connessione di un modulo XTR-ZB1, la quale, come detto nella prima puntata, non è standard.

La demoboard permette di interfacciare il modulo XTR-ZB1 Coordinator ad un qualsiasi PC dotato di una porta USB. Per l'interfacciamento, la Aurel ha deciso di utilizzare un convertitore USB seriale prodotto dalla FTDI: l'FT232R; questo chip sfrutta l'emulazione della seriale su bus USB e in sostanza trasforma l'USB in una porta seriale emulata.

Per il suo funzionamento è necessario installare i driver di emulazione seriale scaricabili dal sito di Aurel o direttamente dal sito del costruttore del chip (FTDI). Dopo l'installazione dei driver la periferica verrà riconosciuta come porta seriale virtuale (VCP - Virtual COM Port) e sarà possibile gestirla come una normale porta seriale.

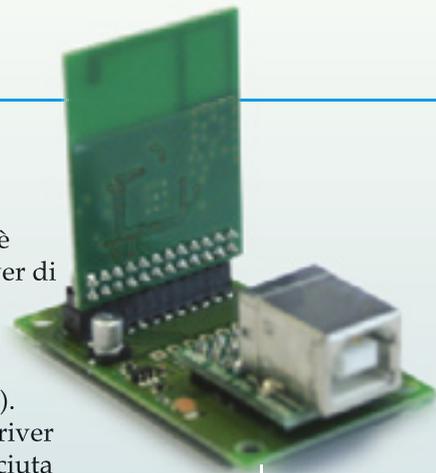


Fig. 2
Demoboard
USB
Coordinator.

La Demoboard Router/End Device

La demoboard XTR-ZB1, una cui foto è visibile nella Fig. 3, è stata realizzata per permettere l'interfacciamento con i moduli Aurel in versione Router o End Device. La scheda è dotata di un pacco batterie formato da due pile stilo per fornire l'alimentazione, di un connettore per il modulo wireless, connettori di espansione ed una serie di periferiche che permettono operazioni di testing e debugging; ci sarà molto utile nel resto del corso per valutare le potenzialità dei moduli prodotti dall'Aurel. Vediamo quindi più in dettaglio come è composta la demoboard Router/End Device, facendo riferimento alla Fig. 4, che rappresenta il layout della scheda vista dal lato componenti. Come si può notare, presenta una serie di componenti di segnalazione e impostazione, oltre ad alcuni connettori di



Fig. 3
Demoboard
Router/
End
Device.

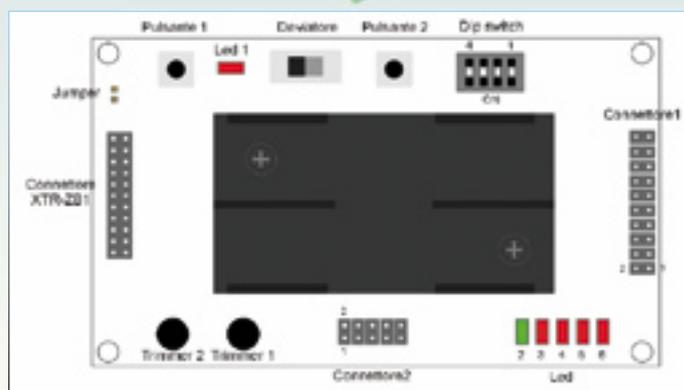


Fig. 4 - Schema della Demoboard Router/End Device.

Tabella 2 - Pinout Connettore 1 della demoboard Router/End Device.

PIN	NOME	DESCRIZIONE
1	Vcc	Connesso al polo positivo del pacco batterie
2	Vcc	Connesso al polo positivo del pacco batterie
3	ADC1	Ingresso 1 del convertitore Analogico/Digitale
4	Input R.	Reset HW del modulo radio (attivo a zero logico). La linea è normalmente a livello alto e quando la si forza a zero avviene un reset simile a quello dopo una perdita di alimentazione.
5	ADC 2	Ingresso 2 del convertitore Analogico/Digitale
6	ADC 3	Ingresso 3 del convertitore Analogico/Digitale
7	OUT 1	Uscita digitale 1
8	ADC 4	Ingresso 4 del convertitore Analogico/Digitale
9	OUT 2	Uscita digitale 2
10	OUT R.	Uscita digitale del modulo radio usata per segnare quando esso è operativo (la linea è a livello alto quando il modulo è operativo e fa parte di una rete, mentre è a zero in fase di start-up e quando il modulo sta cercando di associarsi ad una rete).
11	OUT 4	Uscita digitale 4
12	OUT 3	Uscita digitale 5
13	IN 2	Ingresso digitale 2 (pull-up)
14	IN 1	Ingresso digitale 1 (pull-up)
15	PWD	Pin di disattivazione del power-saving per gli end device.
16	IN 3	Ingresso digitale 3 (pull-up)
17	RESET	Reset Half e Full del modulo radio. La linea è normalmente a livello alto e quando è forzata a zero viene effettuato il reset.
18	INTERRUPT	Interrupt esterno del modulo radio. La linea è normalmente a livello alto e quando viene forzata a livello basso scatta l'interrupt a seguito del quale il modulo radio invia un messaggio al coordinatore.
19	GND	Connesso a massa
20	GND	Connesso a massa

espansione. Il Connettore 1, la cui pinout è riportata nella Tabella 2, è un'espansione composta da due file da 10 pin a passo 2,54 mm, al quale risultano collegate, oltre all'alimentazione della scheda, anche diverse linee del chip cc2430, come i pin del convertitore ADC, gli I/O digitali e le linee di reset ed interrupt. Al connettore 2 sono collegate le linee della seriale (TX, RX, CTS ed RTS), la linea di disattivazione del power-saving per gli end device

Tabella 3 - Pinout Connettore 2 della demoboard Router/End Device.

PIN	NOME	DESCRIZIONE
1	GND	Connesso a massa
2	GND	Connesso a massa
3	PWD	Pin di disattivazione del power saving per gli end device
4	TX	Uscita dati seriali
5	NC	Non Connesso
6	RX	Ingresso dati seriali
7	RTS	Request to send
8	CTS	Clear to send
9	Vcc	Connesso al polo positivo del pacco batterie
10	Vcc	Connesso al polo positivo del pacco batterie

e quelle di alimentazione e massa della demo (vedere pinout in Tabella 3).

Per impostazione predefinita, la porta seriale opera a 38.400 baud, con 8 bit di dati, nessuna parità ed un bit di stop; naturalmente queste impostazioni possono essere modificate con opportuni comandi software, come vedremo più approfonditamente nel seguito del corso. La demoboard dispone di sei led in SMD, utilizzati per fornire indicazioni visive immediate sullo stato di funzionamento e molto utili in fase di debug del sistema. In particolare, abbiamo il LED 1 collegato al positivo dell'alimentazione e utilizzato per segnalare quando la scheda è alimentata. Il led numero 2 è collegato alla linea OUT R e viene utilizzato per segnalare quando il modulo è operativo; risulta acceso quando il modulo è operativo ed è entrato a far parte di una PAN, spento negli altri casi. In fase di startup della rete, questo led potrebbe essere inizialmente spento, perché il dispositivo sta ancora cercando di associarsi alla rete stessa. I led dal 3 al 6 sono collegati alle uscite 1÷4 del modulo, che sono del tipo a logica invertita: quindi quando una di esse è a livello basso il led corrispondente è acceso.

Per testare il convertitore Analogico/Digitale del modulo, sono stati aggiunti due trimmer collegati agli ingressi 1 e 2 dell'ADC, che servono a definire le ampiezze.

La demoboard dispone anche di un dip-switch connesso a tre ingressi digitali del modulo (IN1 ÷ IN3), che può essere utilizzato, ad esempio, per impostare un indirizzo hardware della scheda, nel caso in cui si utilizzi una comunicazione di tipo broadcast e si voglia identificare un particolare dispositivo senza ricorrere al MAC address a 64-bit.

Completano la dotazione della demo, due pulsanti (pulsante 1 e pulsante 2) collegati rispettivamente all'ingresso di reset hardware e alla linea di interrupt esterno.

Procedura di messa in funzione

Per utilizzare la demoboard ZigBee Aurel, dopo aver inserito le due pile AA nell'apposito vano batterie, chiudere il jumper di alimentazione del modulo; oltre che per alimentare o spegnere la scheda, il jumper può essere utile anche per effettuare misure sull'assorbimento del modulo: allo scopo basta collegare un'op-



portuna resistenza di "sense" e misurare la tensione ai suoi capi. Chiuso il jumper, bisogna inserire il modulo XTR-ZB1 nell'apposito zoccolo, con il lato componenti rivolto verso l'esterno della scheda. A questo punto è possibile dare tensione alla demoboard agendo sul deviatore, allorché il LED 1 si accende.

Utilizzo del software demo Aurel

Nel caso di un approccio iniziale alle reti ZigBee (ma spesso anche nelle applicazioni finali) risulta indispensabile utilizzare il coordinatore collegato ad un PC con un'opportuna interfaccia software che permetta la gestione, il monitoraggio e il debug della rete stessa. A corredo del suo kit di sviluppo, Aurel fornisce un applicativo demo, che può essere utilizzato sia come applicativo per il monitoraggio delle reti ZigBee, che come base di partenza per lo sviluppo di applicativi di gestione più complessi.

Vediamo quindi come installare ed utilizzare questo software attraverso un esempio pratico di creazione di una rete ZigBee Aurel.

ZIGBEE IN PRATICA

Per utilizzare un modulo ZigBee, come prima cosa è necessario installare il software ed il componente OCX Aurel che viene utilizzato per la comunicazione. Per l'installazione, lanciare i due programmi di setup e seguire le istruzioni fino al completamento delle due procedure. In caso di difficoltà in fase di installazione, è possibile consultare il manuale dell'applicativo fornito a corredo del kit o scaricato dal sito della Aurel nella sezione relativa alla documentazione ZigBee. L'operazione successiva è l'installazione del driver USB della demoboard coordinator. Collegare la scheda del coordinator al PC con un cavetto USB ed installare la versione del driver appropriata al sistema operativo in uso (esiste la versione per Windows Vista\XP\2000 e quella per i più datati Windows 98\ME). Può essere utilizzata anche la versione autoinstallante del driver (CMD 2.02.04); se utilizzate quest'ultima opzione, si aprirà una finestra di installazione sul prompt dei comandi. Ad installazione ultimata, se tutto sarà andato a buon fine dovrete vedere la finestra di Figura 5. Premete invio per uscire. Per verificare che il driver sia stato correttamente installato, esaminate

Terminologia delle reti ZigBee Aurel

PAN: una Personal Area Network è una rete informatica utilizzata per permettere la comunicazione tra diversi dispositivi vicini a un singolo utente. La PAN può essere utilizzata per collegare i vari dispositivi tra di loro in modo da consentire scambio di informazioni o per consentire la connessione a reti a più alto livello. Una PAN può essere sia cablata che wireless, nel qual caso prende il nome di WPAN (Wireless Personal Area Network).

PAN ID: il PAN ID è un numero su 16 bit che costituisce l'identificativo di una WPAN ZigBee. È quindi possibile discriminare tra loro due WPAN che si trovano nella stessa area.

MAC Address: il MAC Address è un indirizzo a 64 bit che identifica in maniera univoca ogni nodo all'interno di una rete ZigBee. Secondo le specifiche dello standard deve essere memorizzato in una ROM interna e non può mai essere cambiato. Alcuni costruttori lo indicano anche come Serial Number.

Indirizzo Short: gli indirizzi short sono indirizzi a 16 bit che vengono assegnati dinamicamente al momento della formazione di una PAN ZigBee dal Coordinatore o dai Router.

Canale RF: è il canale a radiofrequenza che viene impiegato per la comunicazione. Come è stato detto in precedenza, una rete ZigBee dispone di diverse bande di frequenza utilizzabili, e, all'interno di queste bande di più canali, in modo tale che possa essere cambiato il canale in uso in presenza di disturbi. Le bande disponibili per lo standard ZigBee sono divise per area geografica e, specificatamente, in Europa si può lavorare a 868 MHz negli Stati Uniti a 915 MHz; i 2,4 GHz valgono praticamente in tutto il mondo. La banda più utilizzata è chiaramente quella a 2,4 GHz, ed è in questa che operano i moduli Aurel. All'interno di tale banda sono disponibili 16 canali da 3 MHz ciascuno.

LQI (Link Quality Indicator): è un parametro che nelle reti ZigBee Aurel indica la qualità del collegamento di un determinato nodo con il coordinatore della PAN. Un valore del parametro LQI compreso tra 100 e 200 è indice di un buon collegamento radio.

KA (Keep Alive): il parametro keep alive è utilizzato nelle reti Aurel sui Router e sugli End Device, per ottenere il cambio automatico di canale sulla rete. Quando su un router è attiva la funzione keep alive, esso trasmette un pacchetto ogni 20 secondi al coordinatore, aspettandosi una risposta. Quando non riceve una risposta dopo sei tentativi, esegue un half reset e tenta una nuova associazione, effettuando uno scan su tutti i canali disponibili. Sugli End Device l'operazione è simile, con la differenza che l'End Device non interroga direttamente il coordinatore ma il suo dispositivo "padre" (che potrebbe essere anche il coordinatore stesso), eseguendo la stessa procedura descritta nel caso del Router. L'opzione keep alive consente all'utente di una rete ZigBee Aurel di cambiare canale RF in uso senza perdere il collegamento con i dispositivi associati alla rete, che "seguono" la migrazione del coordinatore verso il nuovo canale.

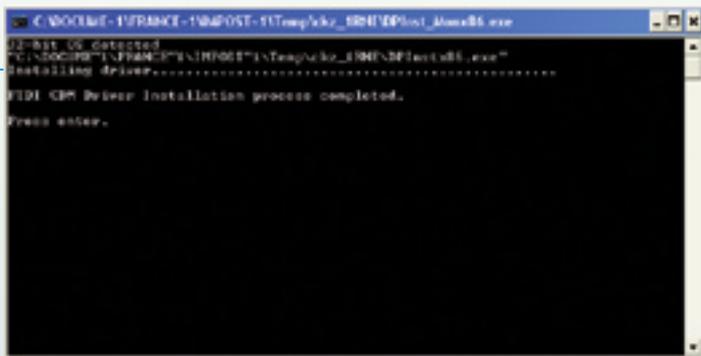


Fig. 5 - Installazione del driver della scheda USB coordinator.

le risorse hardware del sistema (pannello di controllo\systema\hardware\gestione periferiche): sotto l'insieme porte COM e LPT dovreste trovare una voce nominata USB Serial Port, come mostrato nella Fig. 6. Quella è la scheda USB Coordinator.

Una volta completata l'installazione del driver, bisogna predisporre le quattro demoboard per l'accensione (seguendo le istruzioni del paragrafo precedente) e collocarle a piacimento all'interno di una stanza facendo attenzione a non appoggiarle su superfici conduttive, onde evitare che si determinino cortocircuiti tutt'altro che salutari.

Vi ricordiamo che i moduli dello starter kit sono router, quindi con essi è possibile creare diverse topologie di rete. Inoltre, siccome ogni router è in grado di instradare pacchetti dati verso altre destinazioni, è possibile creare delle catene nelle quali la distanza tra il coordinatore e l'ultimo router della catena è pari alla massima distanza che può esistere tra due dispositivi moltiplicata per il numero di dispositivi intermedi. Ora è possibile fornire l'alimentazione alle schede agendo sul deviatore e lanciare, dal menu Programmi del computer, l'applicativo Aurel ZigBee-Programma_Client. Inizialmente comparirà la schermata della Fig. 7, che richiede l'inserimento del numero della porta COM virtuale assegnata al coordinatore.

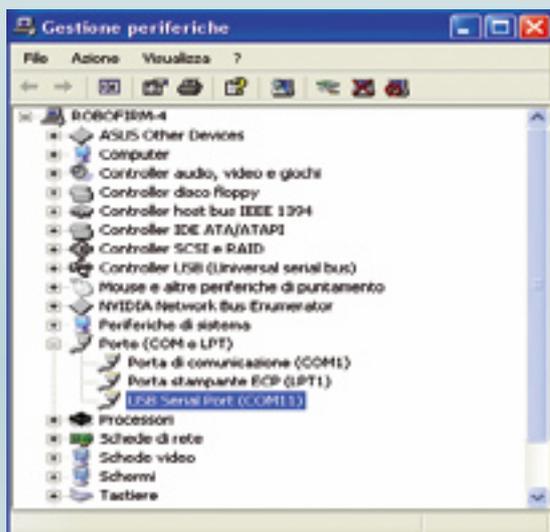


Fig. 6 - Elenco risorse di sistema. Notate la presenza della scheda USB coordinator.

Inserite il numero della porta COM virtuale assegnata dal sistema (è visibile nella schermata delle risorse hardware) e fate clic sul pulsante OK. Eseguita questa operazione, comparirà lo splash screen e successivamente la schermata principale del programma. Non appena il programma viene caricato, il coordinatore inizia la procedura per la creazione di una nuova rete e si mette in attesa di richieste di associazione da parte di altri dispositivi Aurel; in fase di creazione della rete, ad eventuali dispositivi non Aurel non viene concesso di associarsi.

Per associare i dispositivi alla rete, bisogna premere il pulsante UnLock Net (se la rete risultava bloccata) ed attendere che i dispositivi si associno; questa operazione può richiedere alcuni secondi, in funzione della distanza tra i dispositivi e della complessità della rete. Man mano che i dispositivi cominciano ad associarsi, sulle demoboard si accende il LED verde (LED 2) e sulla tabella nella parte centrale della finestra iniziano a comparire i dati relativi ai

singoli dispositivi. Quando tutti i dispositivi sono stati correttamente associati, la schermata del programma client dovrebbe apparire come quella visibile nella Fig. 8. Oltre alla finestra principale, all'apertura del programma viene anche attivata una finestra di log, visibile nella Fig. 9, che contiene lo storico di tutti i messaggi scambiati dall'applicativo.

Quando tutti i dispositivi sono stati correttamente associati, la schermata del programma client dovrebbe apparire come quella visibile nella Fig. 8.



Fig. 7 - Inserimento della porta COM virtuale del coordinatore.



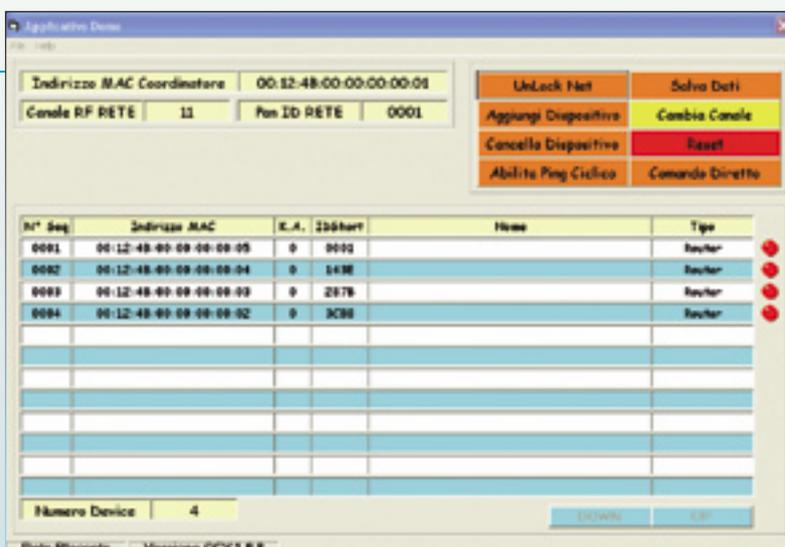


Fig. 8 - Schermata del programma client con quattro dispositivi associati ma non ancora attivi.

cattivo e dal PAN Coordinator. Come si vede, la schermata principale del programma client fornisce una serie di utili informazioni sulla rete appena creata. Nella parte in alto a sinistra viene fornito il MAC address del Coordinatore, il canale RF attualmente in uso e il PAN ID della rete (all'avvio viene eseguito in automatico dal coordinatore uno scan di tutti i canali disponibili e viene scelto quello meno affetto da disturbi). In alto a destra è presente un menu contenente il gruppo pulsanti di comando, tramite i quali possono essere eseguite alcune operazioni base che descriveremo in seguito. La tabella che occupa la parte centrale della finestra contiene invece alcune informazioni relative ai dispositivi associati alle rete. Nel caso di reti con più di 12 dispositivi associati, è possibile utilizzare i pulsanti UP e DOWN, per scorrere l'elenco. Per ogni dispositivo sono riportate le seguenti informazioni:

- numero del dispositivo;
- MAC Address a 64-bit;
- stato del parametro Keep Alive: 0 indica keep alive disattivato, 1 keep alive attivato;
- indirizzo Short a 16-bit;
- nome del dispositivo;
- tipo di dispositivo: può essere Router o End Device;
- stato del dispositivo: il led verde indica dispositivo in stato ON (attivo), mentre il led rosso indica dispositivo in stato OFF (non attivo).

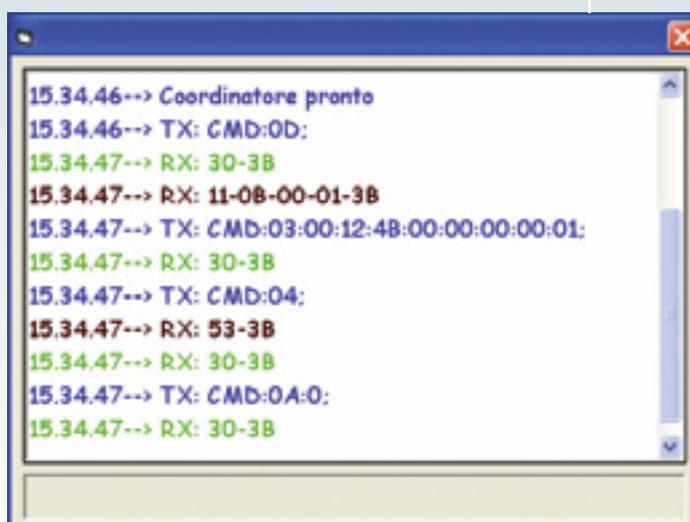
All'avvio del programma, lo stato di tutti i dispositivi è OFF. Diventa ON in caso di risposta al comando ping ciclico o se giunge una risposta dal device quando si entra nella

finestra di gestione dello stesso.

Cliccando due volte nel campo nome della tabella, si entra nella finestra gestione dispositivo, rappresentata in Fig. 10. In questa schermata sono riassunte tutte le informazioni relative al singolo dispositivo contenute nella tabella della schermata principale ed

inoltre si possono cambiare alcuni parametri del dispositivo e controllare ingressi ed uscite della scheda demo. Potete fare alcune prove cambiando lo stato dei quattro led collegati alle linee di uscita, oppure leggendo lo stato degli ingressi digitali (collegati al dip switch) ed analogici (collegati ai trimmer). Se variate lo stato dei trimmer o degli ingressi digitali, per leggere la variazione dovete aggiornare la finestra premendo sul pulsante *Applica* nella parte bassa della finestra stessa. Altre informazioni che vengono visualizzate sono il parametro LQI (vedi riquadro di approfondimento sulla terminologia delle reti Aurel), il valore della tensione presente ai capi della batteria ed il livello della temperatura del sensore interno del system-on-chip cc2430. Quest'ultimo corrisponde al valore della temperatura di esercizio del chip e può variare da modulo a modulo; non è quindi utilizzabile per risalire alla temperatura esterna. Dopo aver fatto alcuni test e verificato che il dispositivo sia collegato e risponda correttamente ai

Fig. 9 Finestra di log.



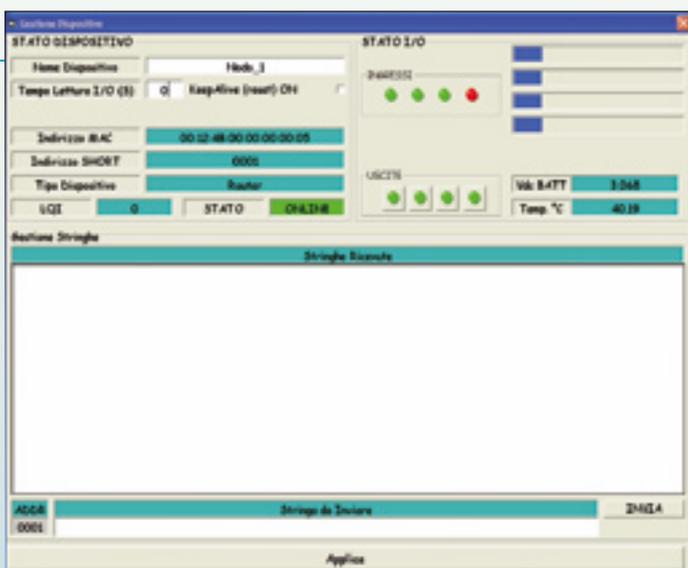


Fig. 10 - Schermata di gestione dispositivo.

Fig. 13, che richiede la conferma del MAC address del dispositivo da eliminare. Se il MAC address corrisponde, si può fare clic sul pulsante OK per continuare; il dispositivo verrà cancellato dalla rete senza ulteriori richieste di conferma.

Riassociazione di un dispositivo precedentemente cancellato

Nel caso invece si voglia riassociare alla rete un dispositivo precedentemente cancellato, è possibile utilizzare il pulsante aggiungi dispositivo.

comandi, potete assegnargli un nome simbolico (nell'esempio *Nodo_1*), premere sul pulsante applica e chiudere la finestra. Ripetete questa operazione per tutti e quattro i nodi. Il risultato finale dovrebbe essere la finestra illustrata nella Fig. 11.

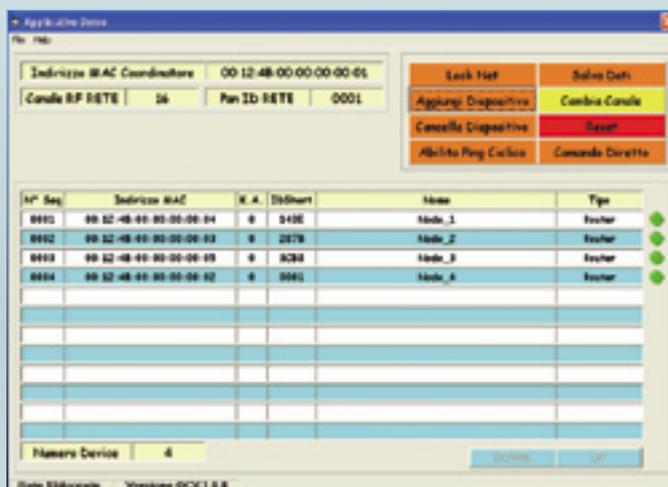


Fig. 11 - Schermata del programma client con quattro dispositivi associati ed attivi.

Come si vede, adesso il campo nome contiene il nome simbolico associato al dispositivo e il led a fianco è diventato verde, ad indicare che il dispositivo è associato alla rete ed è attivo. Vediamo adesso alcune operazioni che è possibile eseguire sulla rete ZigBee Aurel attraverso il software client.

Cancellazione di un dispositivo dalla rete

È possibile eliminare uno specifico dispositivo dalla rete formata (nel caso, ad esempio, non si desidera che il dispositivo ne faccia parte, o nel caso di associazione fortuita), facendo uso del comando cancella dispositivo presente nel gruppo pulsanti di comando. Se viene premuto questo pulsante, comparirà la finestra di dialogo illustrata nella Fig. 12. In essa il computer richiede il numero sequenziale del dispositivo da cancellare.

Dopo aver selezionato il numero e fatto clic sul pulsante OK, comparirà la schermata di



Fig. 12 - Selezione dispositivo da cancellare.

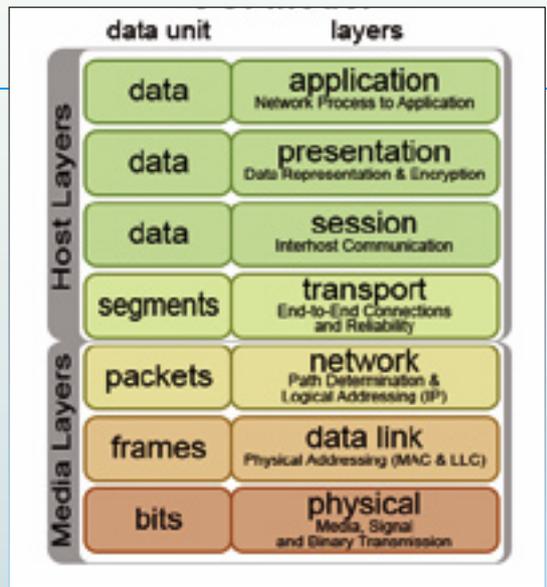


Fig. 13 - Conferma del MAC address da cancellare dalla rete.

Nel caso si esegua questa operazione, comparirà la finestra di dialogo illustrata nella Fig. 14, che richiede il MAC address del dispositivo da associare alla rete. In tal caso dovete inserire l'indirizzo MAC e fare clic sul pulsante OK. Se il dispositivo non si riassocia autonomamente, può essere necessario eseguire un half reset.



Modello ISO-OSI



Cambio del canale attivo e parametro keep alive

Un'altra possibilità offerta dal software demo della Aurel è il cambio manuale del canale RF in uso. Come è stato spiegato in precedenza, all'avvio della rete il Coordinator esegue uno scan su tutti i canali messi a disposizione dallo standard (per trasmissioni a 2,4 GHz i canali disponibili sono quelli che vanno dall'11 al 26) e connette la rete nel canale che ritiene meno affetto da disturbi. Tuttavia se le prestazioni della rete si degradano dopo un

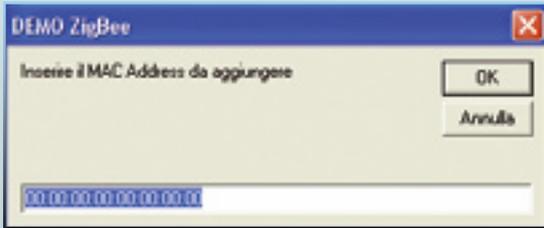


Fig. 14 - Finestra di dialogo Aggiungi Dispositivo.

certo periodo di utilizzo, è possibile che la situazione dei canali attivi nella zona sia mutata e che il canale selezionato sia ora disturbato da interferenze provenienti da reti limitrofe. In questo caso è possibile cambiare il canale attivo via software senza essere costretti ad eseguire un reset dell'intera rete. Per cambiare il canale RF in uso premere il pulsante

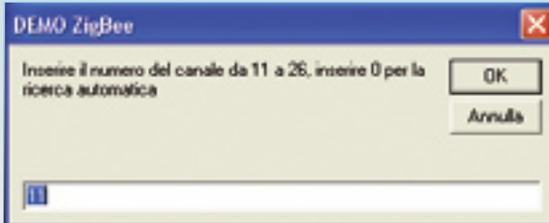


Fig. 15 - Finestra di dialogo Cambio Canale.

Cambia Canale presente nel gruppo pulsanti di comando. Apparirà la finestra di dialogo di Fig. 15, la quale richiede di inserire il canale RF che si desidera utilizzare. L'input ammesso è un numero da 11 a 26; se viene inserito lo 0, il Coordinator provvederà ad eseguire un nuovo scan dei canali disponibili e selezionerà il canale più libero da interferenze. Nel caso venga eseguito il cambio del canale RF via software, tutti i nodi della rete dovranno

seguire la migrazione del Coordinator verso il nuovo canale, altrimenti non saranno possibili ulteriori comunicazioni. Se i nodi non riusciranno a ristabilire la comunicazione sul nuovo canale, bisognerà eseguire un reset locale su ogni nodo, per permettere a ciascun dispositivo di ricollegarsi. Nel caso non si riesca a firmare la rete sul nuovo canale, è possibile eseguire un reset del Coordinator via software (pulsante Reset). In questo caso, se è stato precedentemente attivato il parametro keep alive (KA) sui singoli nodi, questi ultimi riusciranno a ricollegarsi al Coordinator sul canale predefinito (11).

Salvataggio dei dati della rete

Tra le varie funzionalità, il software offre anche la possibilità di salvare i MAC address dei dispositivi associati alla rete. Per farlo, bisogna premere il pulsante Salva Dati nel gruppo pulsanti di comando. Nella directory dell'applicativo (di default: Programma/Demo ZigBee), verrà salvato un file di testo contenente i MAC dei dispositivi associati. Il formato del file è visibile nella Figura 16.

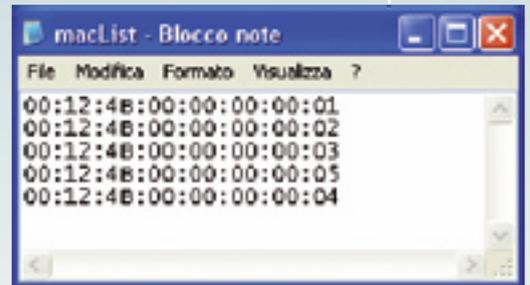


Fig. 16 Formato di salvataggio degli indirizzi MAC.

Il primo MAC address della lista è quello del Coordinator. Concludiamo così questa seconda puntata sulle reti ZigBee Aurel. Nella prossima puntata del corso descriveremo in dettaglio il componente OXC di Aurel (ed il relativo set di comandi di gestione della rete) e inizieremo la scrittura di un nostro software di gestione della rete personalizzato, sfruttando le proprietà, le funzioni e gli eventi forniti da tale componente.



CORSO ZigBee[®]

in collaborazione con
AUREL

di FRANCESCO FICILI

Dopo aver descritto hardware e software di installazione del kit di sviluppo ZigBee Aurel, è giunto il momento di scendere nei dettagli della programmazione. Terza puntata.

Nelle precedenti puntate è stato introdotto il protocollo ZigBee e si è avuto modo di valutare il sistema di sviluppo ZigBee dell'Aurel. Sono stati esaminati in dettaglio le caratteristiche dello standard e i componenti del kit proposto dall'Aurel per la sperimentazione con i moduli XTR-ZB1. È stato inoltre presentato il software demo che permette la gestione ad alto livello di una PAN ZigBee attraverso un'intuitiva interfaccia grafica realizzata in Visual Basic 6. In questa terza puntata del corso analizzeremo in

dettaglio il componente OCX fornito dall'Aurel, che permette di realizzare interfacce di gestione della rete; inizieremo, inoltre, la scrittura di un nostro software di interfacciamento e controllo della rete.

IL COMPONENTE OCX DI AUREL

OCX è l'acronimo di OLE Control eXtension. I controlli OCX sono utilizzati in linguaggi come ad esempio il Visual Basic o il Visual C++ per aggiungere funzionalità ad un programma che si



Tabella 1.a - Proprietà del componente OCX

Proprietà	Descrizione
Public Property Get OCXVersion()	Ritorna una stringa contenente la versione dell'OCX installato.
Public Property Get Commport()	Ritorna un integer contenente la Virtual COM Port utilizzata.
Public Property Let Commport (COMPort byVal as Integer)	Setta la porta COM che si intende utilizzare ed inizializza la seriale con i relativi buffer e timer utilizzati dall'OCX per la gestione della stessa. In particolare, l'OCX utilizza un buffer all'interno del quale vengono memorizzate le stringhe ricevute via seriale dal Coordinator ed un timer a 50ms che verifica la presenza di stringhe nel buffer di ricezione.

Tabella 1.b - Eventi del componente OCX

Eventi	Descrizione
Public Event StringReceived (ByVal RecString as String)	Evento generato quando l'OCX rileva la presenza di una stringa nel buffer di ricezione. L'evento restituisce la stringa così come ricevuta sulla seriale.
Public Event StringSent (ByVal SentString as String)	Evento generato quando l'OCX invia via seriale una stringa. L'evento ritorna la stringa inviata.
Public Event DataReceived (ByVal ShortAddr as String, Data as String)	Evento generato quando viene rilevata la presenza di una stringa inviata da un dispositivo remoto. L'evento ritorna l'indirizzo short del dispositivo remoto (ShortAddr) e i dati utili (Data).
Public Event ErrorDetected (ErrorCode as Integer)	Evento generato quando viene rilevato un errore sulla seriale. Viene restituito il codice di errore corrispondente.

Tabella 1.c - Funzioni del componente OCX

Funzioni	Descrizione
Public Sub StartNet ()	Funzione che invia su seriale il comando CMD:04; (Start Net).
Public Sub StartOPERATIONS ()	Funzione che inizializza la porta seriale, i buffer ed i timer utilizzati dall'OCX. Deve essere utilizzata nell'applicativo durante la fase di inizializzazione della rete, subito dopo aver effettuato i settaggi della porta seriale.
Public Sub StopOPERATIONS ()	Funzione che disabilita il timer utilizzato dall'OCX per la gestione della seriale.
Public Sub LockNet ()	Funzione che invia su seriale il comando CMD:0A:00;
Public Sub UnlockNet ()	Funzione che invia su seriale il comando CMD:0A:FF;
Public Sub PanInfo ()	Funzione che invia su seriale il comando CMD:0D;
Public Sub SendData (ByVal ShortAddr as String, By Val Dati as String)	Funzione che invia su seriale il comando SNT: shortaddr:#dati:Dati; dove #dati è la lunghezza della stringa "Dati" inserita come parametro. Viene eseguito un controllo sui parametri inseriti: se ShortAddr e/o Dati sono stringhe vuote la funzione non fa nulla.
Public Sub AddItem (ByVal MACAddr as String)	Funzione che invia su seriale il comando CMD:03: MACAddr; dove "MACAddr" rappresenta l'indirizzo MAC del dispositivo che si intende possa far parte della rete.
Public Sub SetRfCH (ByVal RfCH as Integer)	Funzione che invia su seriale il comando CMD:07:Rfchannel; dove RfChannel è la conversione in esadecimale del parametro RfCH che rappresenta il nuovo canale nella quale verrà instaurata la rete. Il parametro RfCH può assumere i valori da 11 a 26. Se viene inviato 0 come parametro viene eseguito un nuovo scan su tutti i canali. La funzione esegue un controllo sul parametro RfCH inserito: se è minore di 11 o maggiore di 26 e diverso da zero la funzione non fa nulla.

sta sviluppando. Di fatto si tratta di una collezione di funzioni di libreria, eventi e metodi, che possono essere importati ed utilizzati in

un'applicazione ospite. Il componente OCX di Aurel fornisce appunto una serie di funzioni che permettono la gestione dei moduli XTR-ZB1 (come ad esempio l'inizializzazione della porta seriale dei buffer e dei timer, la ricezione e l'invio di stringhe di dati, ecc.) senza dover passare attraverso i comandi di interfaccia dei moduli stessi. L'uso di questo componente costituisce per lo sviluppatore un ulteriore livello di astrazione (e di semplificazioni) nella scrittura di interfacce di gestione della rete. In tabella 1.a, 1.b ed 1.c sono riassunte rispettivamente le proprietà e gli eventi ed alcune tra le funzioni più significative del componente. Tramite questo ricco assortimento di funzioni, proprietà ed eventi è possibile creare un completo programma di gestione della rete wireless generata con i dispositivi Aurel. Il componente OCX contiene diverse altre funzioni che non sono state, per semplicità, riportate. Per chi desiderasse una guida più esaustiva è possibile scaricare il manuale del componente direttamente dal sito della Aurel (www.aurel.it).

CREAZIONE DI UN SOFTWARE DI GESTIONE PER RETI AUREL

Vediamo adesso come sfruttare il componente OCX per scrivere un nostro software di gestione per reti ZigBee Aurel. Per la scrittura del software di gestione abbiamo scelto Visual Basic 6 e il relativo ambiente di sviluppo, Visual Studio.

La prima cosa da fare è quindi creare un nuovo progetto in questo ambiente. Una volta aperto e salvato il workspace, bisogna, prima di tutto, inserire nel nuovo form il componente OCX. Per far ciò bisogna selezionare, dal menu a tendina Progetto, la voce Componenti. Comparirà una schermata come quella di Fig. 1, all'interno della quale è possibile selezionare il componente che si desidera aggiungere al progetto.

Vi ricordiamo che, per poter essere importato, il componente deve prima essere correttamente installato, come si era fatto nel corso della seconda puntata quando era stato installato il software demo client dell'Aurel.

Una volta importato, il componente diventa accessibile dalla toolbar di Visual Studio, come illustrato in Fig. 2. Non dovete far altro che selezionarlo e trasportarlo sul form,

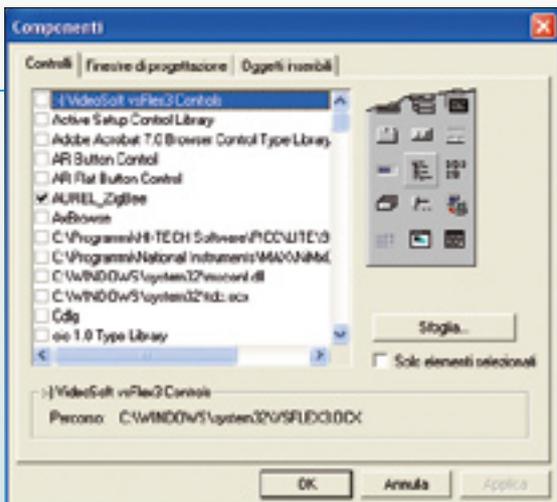


Fig. 1 - Inserimento del componente OCX Aurel.

inserendolo in un punto qualsiasi dell'area di lavoro. Una volta eseguita questa operazione, all'interno del form comparirà il componente ZigBeeNet1, e sarà possibile accedere alle sue funzioni.

Per prima cosa iniziamo a fornire al nostro nuovo software alcune funzioni di base, sfruttando le proprietà dell'OCX, come ad esempio la possibilità di ricavare e quindi visualizzare la versione dell'OCX stesso. L'operazione è resa possibile dalla proprietà Get OCXVersion(), che ritorna una stringa contenente la versione dell'OCX installato. Per la visualizzazione si è scelto di utilizzare l'oggetto status bar, e unitamente alla visualizzazione dell'OCX si è scelto di rendere visibile anche la major e minor revision del software, dichiarate come costanti nella sezione dichiarativa del form. È sempre consigliabile visualizzare e mantenere aggiornate queste informazioni, in quanto molti errori possono derivare dall'uso (a volte inconsapevole) di versioni obsolete, o nel caso di programmi più articolati, che sfruttano componenti diversi, di versioni di tali componenti incompatibili tra loro o con il programma ospite.

Il codice che permette di ricavare e visualizzare la versione dell'OCX e versione del software è rappresentato nel Listato 1.

Il risultato potete vederlo nella Fig. 3: Vediamo adesso qualcosa di più corposo, come l'inizializzazione dell'apparato di comunicazione con il Coordinator (attraverso

Listato 1

```
Visualizzo versioni OCX e SW sulla status bar
StatusBar1.Panels.Add
StatusBar1.Panels(1).Text = "OCX Version " & ZigBeeNet1.OCXVersion
StatusBar1.Panels(2).Text = "SW Version " & SW_Major & "." & SW_Minor
```

Listato 1 - Visualizzazione versione OCX e versione SW.

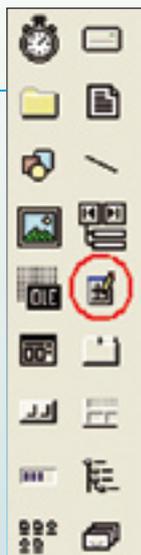


Fig. 2 - Posizione del componente nella toolbar di Visual Studio.



Fig. 3 - Versione OCX e versione SW visualizzate sulla status bar.

la porta seriale emulata su bus USB) e la procedura di avvio della rete.

Dal punto di vista software, per avviare la rete occorre aggiungere nel form due pulsanti, una label ed una textbox. I due pulsanti saranno utilizzati per l'avvio e la chiusura della comunicazione con il Coordinator, la textbox è utilizzata per ricevere in input il numero della porta COM virtuale e la label serve solo come indicazione sullo stato della rete (online ed offline).

Per avviare le comunicazioni bisogna prima di tutto settare i parametri della porta seriale. Per far ciò si usa la proprietà Let Commport (), che riceve

in input il numero della porta seriale sulla quale è collegato il

Coordinator. Il numero della porta seriale da utilizzare viene passato alla proprietà tramite la variabile VCP, che immagazzina il valore ricavato dalla textbox creata appositamente allo scopo. Il Listato 2 riporta il codice relativo a quanto illustrato sopra.

A questo punto si possono utilizzare le due funzioni Sub StartOPERATIONS () e Sub StartNet () per avviare le comunicazioni e comandare al Coordinator di avviare la net, come illustrato nel Listato 3.

La Fig. 4 fa vedere come si presenta la sezione start/stop network del software di controllo quando viene avviata la "net". Una label verde con la scritta "ONLINE" informa l'operatore sullo stato della connes-

Listato 2

```
'Imposto la porta COM virtuale
VCP = txtCOMPortSelection.Text
ZigBeeNet1.CommPort = VCP
```

Listato 2 - Impostazione dei parametri della porta seriale.

Listato 3

```
Creazione rete
ZigBeeNet1.StartOPERATIONS
ZigBeeNet1.StartNet
lblStatus.BackColor = vbGreen
lblStatus.Caption = "ONLINE"
```

Listato 3 Avvio comunicazioni e creazione della rete.

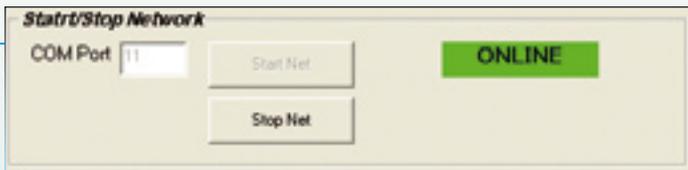


Fig. 4 - Sezione Start/Stop Network del software di controllo.

sione (ONLINE, nel caso della figura). Appena il Coordinator riceve il comando di avvio della rete, esegue uno scan su tutti i canali disponibili (dall'11 al 26) e crea la rete sul canale ritenuto più libero da disturbi. La rete viene identificata tramite un codice a 14 bit, chiamato PAN ID ed uguale ai 14 bit meno significativi del MAC address del Coordinator. All'atto della creazione della rete viene inoltre creata, tramite un algoritmo di generazione di numeri random, la chiave di encryption a 128 bit per la crittografia dei dati. Da questo momento il Coordinator si mette in attesa di associazioni alla rete di altri dispositivi Aurel. Eventuali tentativi di associazione di dispositivi non Aurel vengono rifiutati. Alla sua prima accensione, un dispositivo remoto (Router o End Device) cerca, su tutti

Fig. 5
Finestra di log.



i canali e senza alcun vincolo, una rete cui associarsi. Quando la trova e si associa, gli viene assegnato un indirizzo short a 16 bit, tramite il quale gli verranno indirizzati tutti i successivi pacchetti di rete (per convenzione il Coordinator ha sempre indirizzo short 0x0000). Tale indirizzo short rimane lo stesso per i router, a meno che non venga eseguito un Reset sul dispositivo. L'End Device, invece, cambia indirizzo short a seconda del router cui è associato. Pertanto se l'End Device si muove, uscendo dalla portata radio del suo "padre", e si associa ad un nuovo router, cambia indirizzo short. Allo stesso

Output del comando su coordinatore	
Msg Type ->	1 byte
• 0xCC ->	Nwk update
Indirizzo a 16 bit del nodo ->	2 byte
Indirizzo a 64 bit del nodo ->	8 byte
Tipo di dispositivo interrogato ->	1 byte
Punto e virgola ->	1 byte

Tabella 2
Network update.

modo, se il router padre di un End Device viene spento e l'End Device trova un nuovo router cui associarsi, gli viene assegnato un nuovo indirizzo short.

Per mantenere traccia dei messaggi scambiati sulla seriale tra il software ed il Coordinator, si è deciso di aggiungere al software una textbox che funge da finestra di log contenente tutti i messaggi transitati sulla seriale. Su questa textbox vengono stampati sia i messaggi inviati dal Coordinator al PC sia quelli inviati dal PC al Coordinator, quindi si fa uso sia dell'evento StringReceived che dell'evento StringSent.

All'apertura delle comunicazioni, la finestra si presenta come quella visibile nella Fig. 5. Come si vede vengono visualizzati, oltre all'intestazione della finestra, anche la data e l'ora in cui viene avviato il Coordinator nonché la porta COM virtuale utilizzata.

Vediamo adesso come avviene la fase di associazione vera e propria di un dispositivo alla rete appena creata e come si gestisce il tutto a livello del software di controllo.

Se un dispositivo si associa alla rete, il Coordinator lo notifica sulla seriale con l'invio di un messaggio con codice identificativo 0xCC (Network Update) e contenente l'indirizzo short, l'indirizzo MAC e la tipologia del dispositivo che si sta associando, secondo quanto esposto nella Tabella 2.

Per visualizzare a schermo i dati dei nuovi dispositivi, bisogna prima intercettare le stringhe contenenti i dati che vengono inviati dal Coordinator sulla seriale. Per fare ciò si fa uso dell'evento StringReceived (recString

Listato 4 - Visualizzazione dei messaggi in transito sulla seriale.

Listato 4

```
'Intercetto l'evento scrittura stringa su seriale e plotto su schermo
For i = 1 To Len(recString)
    txtComWindows.Text = txtComWindows.Text & Hex(Asc(Mid(recString, i, 1)))
Next i
txtComWindows.Text = txtComWindows.Text & vbCrLf
```



As String) che restituisce la stringa di dati ricevuta sulla seriale. Prima di prelevare e stampare opportunamente i dati su video, si stampa la stringa così come è stata ricevuta sulla finestra di log, secondo quanto visibile nel Listato 4. Questo piccolo frammento di codice, collocato all'interno dell'evento StringReceived, permette di stampare tutti i messaggi inviati dal Coordinator al PC. Dopo aver stampato la stringa dati sulla textbox bisogna anche decodificarla opportunamente ed aggiornare una tabella dei dispositivi associati con i relativi indirizzi short.

Il Listato 5 illustra il codice che effettua la decodifica della stringa dati e la creazione della tabella di indirizzi short.

L'informazione contenuta nella stringa dati inviata dal Coordinator all'atto dell'associazione di un nuovo dispositivo, a parte il message type (0xCC) ed il punto e virgola finale, è costituita da 11 byte.

I primi due corrispondono all'indirizzo short del dispositivo, i successivi 8 al suo indirizzo MAC e l'ultimo alla tipologia del dispositivo, secondo quanto riportato nella tabella 3. Per riconoscere il Coordinator può essere utilizzato a scelta il suo codice identificativo (0x09) oppure il suo indirizzo short, dal momento che tale indirizzo per il coordinator è sempre 0x0000. Come

Listato 5

```
'Evento 0xCC: Network Update
If (Asc(Mid(recString, 1, 1)) = &HCC) Then

    DevCount = DevCount + 1
    TimerUpdate.Enabled = True

'Aggiorno la lista degli indirizzi short
ShortAddr(DevCount) = Hex(Asc(Mid(recString, 2, 1))) & Hex(Asc(Mid(recString, 3, 1)))

'Numero di sequenza del dispositivo
lblDeviceInfo.Caption = lblDeviceInfo.Caption & "N°:" & DevCount & " "

'Indirizzo short
lblDeviceInfo.Caption = lblDeviceInfo.Caption & "Short:"
If (Asc(Mid(recString, 2, 1)) < 10) Then
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & "0" & Hex(Asc(Mid(recString, 2, 1)))
Else
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 2, 1)))
End If
If (Asc(Mid(recString, 3, 1)) < 10) Then
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & "0" & Hex(Asc(Mid(recString, 3, 1)))
Else
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 3, 1)))
End If
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & " "

'MAC address
lblDeviceInfo.Caption = lblDeviceInfo.Caption & "MAC:"
lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 4, 1))) & ":"
lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 5, 1))) & ":"
lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 6, 1))) & ":"
lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 7, 1))) & ":"
lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 8, 1))) & ":"
lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 9, 1))) & ":"
lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 10, 1))) & ":"
lblDeviceInfo.Caption = lblDeviceInfo.Caption & Hex(Asc(Mid(recString, 11, 1)))
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & " "

'Tipo dispositivo
If (Asc(Mid(recString, 3, 1)) = 0) Then
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & "Type:"
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & "Coordinator"
End If

If (Asc(Mid(recString, 12, 1)) = 7) Then
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & "Type:"
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & "Router"
End If

If (Asc(Mid(recString, 12, 1)) = 6) Then
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & "Type: "
    lblDeviceInfo.Caption = lblDeviceInfo.Caption & "End Device"
End If

    lblDeviceInfo.Caption = lblDeviceInfo.Caption & vbCrLf

End If
```

si può vedere dal Listato 5, al riconoscimento di un message type con codice 0xCC viene per prima cosa aggiornato il contatore del numero di dispositivi (DevCount) ed attivato il TimerUpdate, che determina il lampeggio della spia "UPDATING" all'interno della finestra di comando. Successivamente, per mezzo dell'istruzione:

```
ShortAddr(DevCount) = _Hex(Asc(Mid(recString, 2, 1))) _
& Hex(Asc(Mid(recString, 3, 1)))
```

viene aggiornata la tabella degli indirizzi short dei dispositivi associati alla rete. Questa tabella può essere utilizzata per inviare mes-

Listato 5
Visualizzazione dei dati dei dispositivi e creazione della tabella di indirizzi short.

Tabella 3 - Tipologie dei dispositivi.

Dispositivo	Codice
Coordinatore	0x09
Router	0x07
End Device	0x06
Rejoin End Device	0x04

Caratteristiche del software di gestione

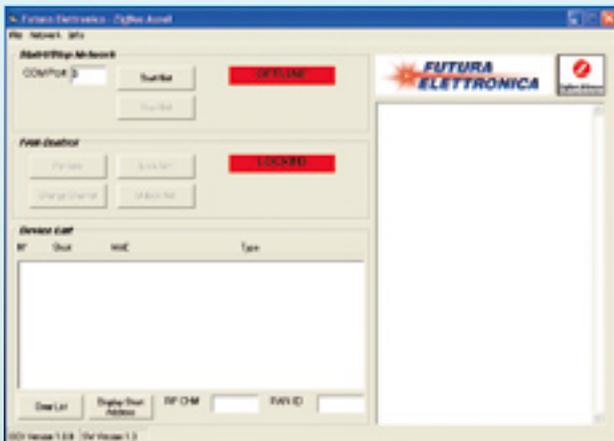
Il software di gestione utilizzato in questa ultima puntata del corso ha lo scopo di mostrare come sia possibile, utilizzando il componente OCX, gestire una rete wireless formata da dispositivi Aurel ZigBee. Il software ha una serie limitata di funzioni, ma permette comunque di ricavare e salvare gli indirizzi MAC e short dei dispositivi collegati, ottenere informazioni sulla rete, bloccare e sbloccare la rete ed anche cambiare il canale RF attivo. Può inoltre essere facilmente ampliato per consentire il controllo remoto dei dispositivi.

Opzioni di visualizzazione e salvataggio

- Monitor messaggi inviati e ricevuti sulla seriale.
- Visualizzazione degli indirizzi short, MAC e tipologia del dispositivo.
- Visualizzazione dei dati della rete (RFchannel e PAN ID).
- Salvataggio dell'elenco degli indirizzi MAC e short.
- Salvataggio Log History File.

Opzioni di controllo

- Blocco e Sblocco della rete.
- Selezione dal canale RF.



saggi ad uno qualsiasi dei dispositivi associati, per mezzo del suo indirizzo short.

La tabella non è altro che un array di stringhe dove l'indice costituisce il numero sequenziale del dispositivo associato e la stringa contiene i due byte dell'indirizzo short.

Le istruzioni successive servono a stampare sulla label DeviceList le informazioni relative ai dispositivi associati, nel seguente ordine: numero di sequenza, indirizzo short, indirizzo MAC e tipologia del dispositivo.

La figura 6 mostra l'output sulla finestra dell'associazione di 4 dispositivi di tipo router (i 4 dispositivi costituenti il kit). È anche possibile salvare le informazioni ottenute su file di testo, utilizzando l'opzione del menu a tendina *File->Save Device List*. In questo

N°	Short	MAC	Type
1	3C00	0:12:4B:0:0:0:2	Router
2	0001	0:12:4B:0:0:0:3	Router
3	143E	0:12:4B:0:0:0:4	Router
4	2B7B	0:12:4B:0:0:0:5	Router

Fig. 6 - Device List.

caso viene aperta una finestra di dialogo che permette di selezionare il percorso in cui salvare il file ed il nome del file stesso. Nella figura 7 è illustrato un esempio di file di testo nel quale è stata memorizzata la device list. Si noti che se alcuni dispositivi vengono associati e poi le comunicazioni vengono

N°	short	MAC	Type
1	2B7B	0:12:4B:0:0:0:5	Router
2	0001	0:12:4B:0:0:0:3	Router
3	143E	0:12:4B:0:0:0:4	Router
4	3C00	0:12:4B:0:0:0:2	Router

Fig. 7 - Salvataggio dettagli dei dispositivi associati.

interrotte (Stop Net) ed il software riavviato, per ottenere nuovamente la Device list e la tabella di corrispondenza degli indirizzi short, bisognerà effettuare un half reset su tutti i nodi, in quanto nel programma non è stato previsto alcun meccanismo per il recupero dei dati della rete attraverso il Coordinator (cosa possibile solo a patto che il Coordinator stesso non venga resettato).

Vediamo adesso come sono stati implementati alcuni controlli e sistemi per ricavare informazioni sulla rete creata: il primo di essi disponibile è il controllo PanInfo, che restituisce una serie di informazioni sulla rete, come il canale RF in uso e l'identificativo univoco della rete (PAN ID). Per ricavare queste informazioni

Tabella 4 - Output del comando PANInfo.

Output del comando
Codice di terminazione --> 1 byte <ul style="list-style-type: none"> • 0x30 --> no_error • 0x33 --> numero di parametri errato • 0x42 --> il dispositivo non è parte della rete
Punto e virgola --> 1 byte
MsgType --> 1 byte <ul style="list-style-type: none"> • 0x11 --> Pan info
Canale su cui è instaurata la rete --> 1 byte
Identificativo della rete (PAN ID) --> 2 byte
Punto e virgola --> 1 byte



Listato 6

```
Evento 0x11: PanInfo
If (Asc(Mid(recString, 1, 1)) = &H11) Then
    lblRFch.Caption = Asc(Mid(recString, 2, 1))
    lblPANid.Caption = Asc(Mid(recString, 3, 1)) & Asc(Mid(recString, 4, 1))
End If
```

Listato 6 - Decodifica del messaggio PanInfo.

viene utilizzata la funzione *Sub PanInfo ()*, disponibile tra le funzioni dell'OCX Aurel. L'output che viene restituito sulla seriale quando è utilizzato tale comando, è riportato nella Tabella 4; per intercettarlo, si utilizza sempre l'evento *StringReceived*. Per distinguerlo da altri tipi di messaggi si usa il suo codice identificativo (*MsgType*). Il codice che decifra il messaggio è riportato nel Listato 6. Come si può vedere dal codice, il numero di canale RF e l'identificativo della PAN vengono scritti, rispettivamente, nelle label *lblRFch* e *lblPANid*. Ciò che si ottiene nella finestra principale del programma è visibile nella figura 8. Uno degli altri controlli implementati è il controllo di blocco e sblocco della rete. Se la rete è bloccata, i dispositivi che non siano già stati precedentemente associati ovvero che non siano stati espressamente autorizzati tramite l'uso del MAC address identificativo, non possono entrare a far parte della rete. Per bloccare e sbloccare la rete si usano i tasti "Lock Net" e "Unlock Net", il cui funzionamento è piuttosto intuitivo. I listati 7 ed 8 illustrano il codice VB6 utilizzato per l'implementazione di tali comandi. Come si vede, in essi viene fatto uso delle funzioni *LockNet* ed *UnlockNet*, rese disponibili dall'OCX.

Listato 7

```
Private Sub cmdLock_Click()
    ZigBeeNet1.LockNet
    lblLock.BackColor = vbRed
    lblLock.Caption = "LOCKED"

    'Comandi associati
    cmdLock.Enabled = False
    cmdUnlock.Enabled = True
End Sub
```

Listato 7 - Blocco della rete.

Listato 8

```
Private Sub cmdUnlock_Click()
    ZigBeeNet1.UnlockNet
    lblLock.BackColor = vbGreen
    lblLock.Caption = "UNLOCKED"

    'Comandi associate
    cmdLock.Enabled = True
    cmdUnlock.Enabled = False
End Sub
```

Listato 8 - Sblocco della rete.

Oltre all'invio del comando verso il Coordinator, viene evidenziato su una label lo stato attuale della rete (LOCKED ed UNLOCKED). È importante tenere sempre presente questa label, in quanto se il Coordinator viene privato della sua sorgente di alimentazione perde i dati relativi ai dispositivi associati, quindi al successivo

ripristino della rete quest'ultima deve essere sbloccata, altrimenti qualsiasi tentativo di associazione da parte di Routers o End Device risulterebbe vano. Infine, il software da noi proposto fornisce anche la possibilità di cambiare il canale RF in uso. Per farlo bisogna fare clic sul pulsante "Change Channel"; dato questo

comando, comparirà la finestra illustrata nella Fig. 9, che richiede l'immissione del nuovo canale sul quale il Coordinator dovrà creare la rete. I valori ammessi vanno da 11 a 26. Se viene inserito il valore 0, il Coordinator effettuerà uno scan su tutti i canali disponibili e sceglierà tra questi quello ritenuto più libero da interferenze. Il Listato 9 mostra il codice utilizzato per l'implementazione di questo comando; in esso viene utilizzata la funzione dell'OCX Aurel "SetRfCh (NewCh)" la quale prende come argomento il numero del nuovo canale RF. Dopo l'impostazione del nuovo canale, viene attivato un timer che genera il lampeggio del led "Updating Nwk", il quale indica un update del sistema; nel contempo, viene ripulita la label contenente le informazioni sui dispositivi, in attesa delle nuove associazioni. Nella Fig. 10 è illustrato come si presenta il frame PAN Control, tramite il quale è possibile accedere alle funzioni appena descritte.

Listato 9

```
NewCh = txtNewCh.Text
ZigBee_Aurel.ZigBeeNet1.SetRfCh (NewCh)
ZigBee_Aurel.TimerUpdate.Enabled = True
ZigBee_Aurel.lblDeviceInfo = ""
```

Listato 9 - Cambio del canale RF attivo.

Fig. 10
Sezione PAN Control.

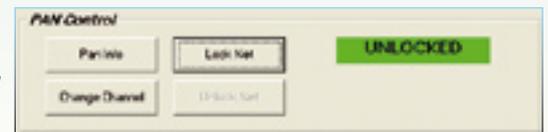
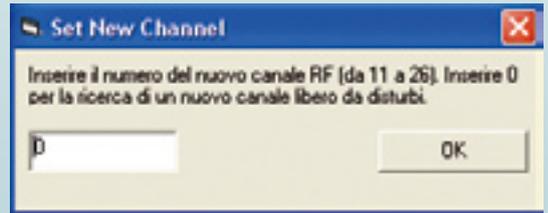


Fig. 8
Visualizzazione RFchannel e PAN ID.



Fig. 9
Cambio del canale RF attivo.





CORSO ZigBee[®]

in collaborazione con
AUR'EL

di FRANCESCO FICILI

Impariamo a gestire lo stato delle linee di uscita dell'XTR-ZB1 usando i comandi aggiuntivi previsti nel nostro software e vediamo il codice in Visual Basic 6 che lo rende possibile. Quarta puntata.

Nella terza puntata di questo corso abbiamo iniziato ad occuparci dello sviluppo di un software di gestione realizzato in Visual Basic 6 che sfrutta il componente OCX fornito da Aurel; le funzioni descritte ed analizzate riguardavano esclusivamente il monitoraggio della rete (ricezione dei MAC address e degli indirizzi short, creazione della tabella degli indirizzi, ricezione del PANid, ecc.) ed il controllo di alcuni parametri particolari (selezione canale RF, blocco/sblocco della rete). In questa quarta e conclusiva puntata

del corso, verrà introdotto ed analizzato un ulteriore set di comandi che permette di gestire sia le linee di output collegate ai led della demoboard sia la ricezione di segnali di allarme.

GESTIONE DELLE LINEE DI OUTPUT

Nelle prime due puntate del corso abbiamo appreso come creare una rete e raccoglierne i dati più significativi (in sostanza gli indirizzi MAC, ma anche gli indirizzi short dei nodi). A questo punto siamo in grado di sfruttare le caratteristi-



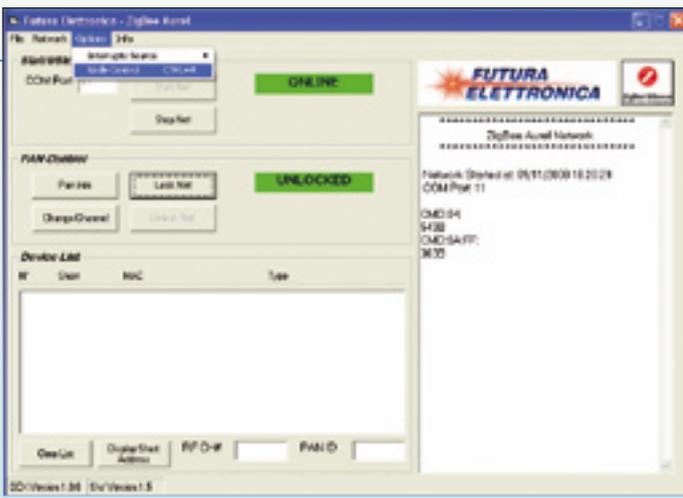


Fig. 1 - Accesso alla finestra di controllo delle uscite GPIO.

che della rete nelle applicazioni che più ci interessano. Applicazioni tipiche sono, ad esempio, il monitoraggio ed il controllo remoto (wireless) di dispositivi elettronici quali relé, sensori, interfacce seriali di microcontrollori, eccetera.

Le demoboard Router/End Device fornita a corredo del kit ZigBee Aurel, dispone di quattro led collegati ad altrettante linee di I/O del modulo XTR-ZB1; naturalmente tali linee sono state impostate come output mediante il firmware, in modo da poter controllare l'accensione o lo spegnimento dei led ad esse collegati. Sfruttando le potenzialità del componente OCX Aurel è stato aggiunto al software di gestione un ulteriore form che permette di controllare lo stato di tali linee. Vediamo innanzitutto come utilizzare questi comandi aggiuntivi; successivamente, affronteremo il codice scritto in Visual Basic 6 che rende possibile il tutto. Allo scopo bisogna, per prima cosa, collegare la demoboard coordinator alla porta USB del PC tramite il cavo fornito in dotazione al kit, accendere una demoboard ed eseguire l'associazione del dispositivo utilizzando il software di gestione (se il dispositivo non compare nella lista di quelli associati, dovete eseguire un half-reset sul nodo locale). Per accedere alla finestra di

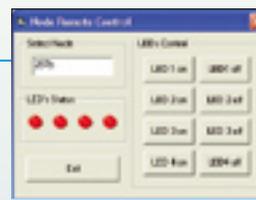


Fig. 2 - Finestra di controllo Node Remote Control.



Fig. 3 - Visualizzazione dello stato dei led con il LED 1 acceso.

controllo delle uscite GPIO di un qualsiasi modulo associato alla rete, bisogna fare clic sull'opzione *Node Control* del menu a tendina *Options*, come illustrato nella Fig. 1; apparirà così la finestra illustrata nella Fig. 2 denominata, appunto, *Node Remote Control*. Qui, per prima cosa bisogna selezionare il nodo della rete che si intende controllare; allo scopo, bisogna inserire nella textbox *Select Node* l'indirizzo short del nodo desiderato (è possibile ricavarlo dalla device list della finestra principale) e, successivamente, agire sui pulsanti di controllo (*LEDx on* e *LEDx off*) per accendere e spegnere i led.

Lo stato del led viene inoltre visualizzato all'interno del riquadro (frame) *LED's Status*, presente nella stessa finestra di dialogo, di fianco ai pulsanti, come evidenziato dalla Fig. 3. A questo punto vediamo il codice che ha permesso l'implementazione del controllo: l'OCX dispone di due funzioni espressamente dedicate alla gestione degli output digitali che però in questo caso non vengono utilizzate in quanto si è deciso di fare uso di una funzione generica per l'invio di comandi; il nostro scopo è avviarsi all'utilizzo di tale funzione che, proprio perché generica, permette l'invio di uno qualsiasi dei comandi supportati dai moduli. La funzione dell'OCX che è stata da noi utilizzata è la **sub SendDirectToCoordinator (By Val Command As String)**, che invia sulla seriale il comando contenuto all'interno della stringa *Command*.

I due comandi utilizzati sono **CMD:0B:addr:#out;** (per settare l'uscita ad 1) e **CMD:0C:addr:#out;** (per settare l'uscita a 0). Le tipologie

Tabella 1 - Parametri dei comandi CMB:0B e CMD:0C

Parametri	Tipo di parametri	Output di comando
<p><i>Addr</i> -> indirizzo a 16 bit del nodo a cui si intende inviare il comando. <i>#out</i> -> uscita che si desidera settare. I valori ammissibili sono compresi nell'intervallo [1-4]</p>	<p><i>Addr</i> -> 4 byte indicanti, in caratteri ASCII, il valore esadecimale dell'indirizzo del nodo destinatario. <i>#out</i> -> 1 byte indicante, in caratteri ASCII, il numero dell'uscita che si intende settare.</p>	<p>Codice di terminazione -> 1 byte</p> <ul style="list-style-type: none"> • 0x30 -> no_error • 0x32 -> error nei parametri • 0x33 -> numero di parametri errato • 0x40 -> system busy • 0x49 -> lost set reset <p>punto e virgola -> 1 byte</p>

Tabella 2 - Output sul nodo locale dei comandi CMB:0B e CMD:0C.

Output del comando su nodo locale
MsgType -> 1 byte
• 0x21 -> set//reset pin
indirizzo a 16bit del nodo remoto -> 2 byte
punto e virgola -> 1 byte

dei parametri dei due comandi sono riportate nella Tabella 1.

L'output del comando sul nodo locale in caso di corretta ricezione, è riportato nella Tabella 2.

La sintassi è molto semplice e prevede che dopo il codice del comando ed il carattere ":" venga inserito l'indirizzo short (in caratteri ASCII) del nodo di destinazione, poi un nuovo carattere ":" ed infine il numero dell'uscita che si intende settare o resettare, seguito dal punto e virgola.

Se il comando viene correttamente ricevuto dal nodo destinazione, quest'ultimo risponderà con un messaggio contenente il codice 0x21 (set/reset pin) il suo indirizzo short ed il carattere di terminazione (punto e virgola).

Il Listato 1.a riporta il codice Visual Basic utilizzato per l'accensione del LED 1, mentre il Listato 1.b contiene il codice che permette lo spegnimento dello stesso led.

Nella Figura 4 sono riportati i messaggi in transito sulla seriale dopo l'invio del suddetto comando per l'accensione del LED 1.

Naturalmente, anziché led, alle linee di output possono essere collegati altri dispositivi, come ad esempio relé (pilotati da transistor) transistor sia BJT che ad effetto di campo, switch allo stato solido, ponti ad H per con-



Fig. 4 - Messaggi in transito sulla seriale dopo l'accensione del LED 1 del nodo con indirizzo short 0x287B.

Listato 1a

```

Private Sub cmdLED1on_Click()

'setta a 0 il valore dell'uscita logica 1 relativa
'al nodo txtNode.text
OutputBuffer = "CMD:0C:" & txtNode.Text & ":1;"
ZigBee_Aure1.ZigBeeNet1.SendDirectToCoordinator (OutputBuffer)
imLED1on.Visible = True
imLED1off.Visible = False
End Sub
    
```

Listato 1a - Codice Visual Basic per l'accensione del LED 1.

Listato 1b

```

Private Sub cmdLED2off_Click()
'setta a 1 il valore dell'uscita logica 1 relativa
'al nodo txtNode.text
OutputBuffer = "CMD:0B:" & txtNode.Text & ":2;"
ZigBee_Aure1.ZigBeeNet1.SendDirectToCoordinator (OutputBuffer)
imLED2on.Visible = False
imLED2off.Visible = True
End Sub
    
```

Listato 1b - Codice Visual Basic per lo spegnimento del LED 1.

trollo di motori elettrici (ad esempio per la movimentazione di tapparelle) e quant'altro. In sostanza, con l'aggiunta dell'opportuno hardware di potenza, si può realizzare un completo sistema di attuazione dotato di controllo remoto wireless.

Ricordiamo inoltre che le caratteristiche intrinseche delle reti ZigBee, come ad esempio il mesh networking, sono gestite dal firmware caricato all'interno dei moduli Aurel in maniera trasparente all'utente. Ad esempio, per sfruttare il mesh networking dei moduli è possibile eseguire qualche semplice esperimento: accendete e collegate alla rete una demoboard equipaggiata con un modulo Router, quindi associate il dispositivo alla rete utilizzando l'apposito software. Ora allontanate il dispositivo dal coordinator fino ad andare al di fuori della sua portata radio; verificate l'assenza del collegamento provando ad accendere e spegnere uno dei led (non ci dovrà essere risposta da parte del router). Adesso posizionate una seconda demoboard a distanza intermedia tra il coordinator e la prima demoboard; associate il secondo modulo (sempre per mezzo del software di controllo) e riprova ad accendere e spegnere un led della prima demoboard, quella che prima era fuori portata. Noterete che adesso il collegamento è presente. Il dispositivo numero 2 sta facendo da "router" tra il coordinator ed il router numero 1.

Appare evidente come posizionando più moduli all'interno, ad esempio, di una abitazione, sia possibile creare una maglia connessa

Il software di gestione

Il programma di gestione utilizzato in questa ultima puntata del corso ha lo scopo di mostrare come sia possibile, sfruttando il componente OCX, gestire una rete wireless formata da dispositivi Aurel ZigBee.

Il software ha una serie limitata di funzioni, ma permette comunque di ricavare e salvare gli indirizzi MAC e short dei dispositivi collegati, di ricavare informazioni sulla rete, di bloccare e sbloccare la rete, di cambiare il canale RF attivo, di controllare alcune linee di output digitale e di interpretare segnali di allarme. Può inoltre essere facilmente ampliato per includere altre caratteristiche.

Le opzioni di visualizzazione e salvataggio sono le seguenti:

- Monitor messaggi inviati e ricevuti sulla seriale;
- Visualizzazione degli indirizzi short, MAC e tipologia del dispositivo;
- Visualizzazione dei dati della rete (RFchannel e PAN ID);
- Salvataggio dell'elenco degli indirizzi MAC e short;
- Salvataggio Log History File.

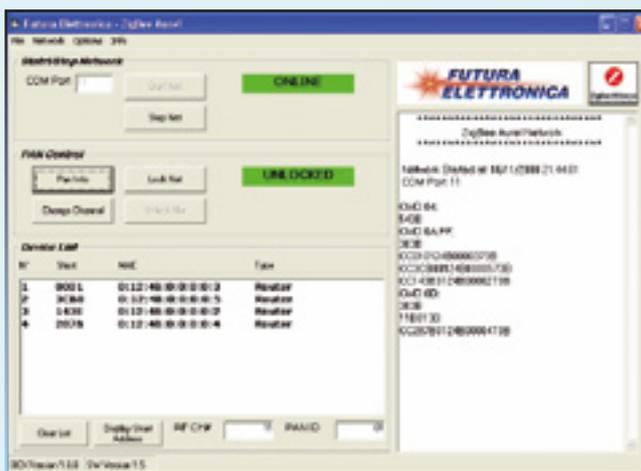


Fig. 7 - Software di gestione.

Le opzioni di controllo sono, invece, le seguenti:

- Blocco e Sblocco della rete;
- Selezione dal canale RF;
- Controllo di 4 linee di output digitale;
- Gestione di una linea di interrupt come segnale di allarme.

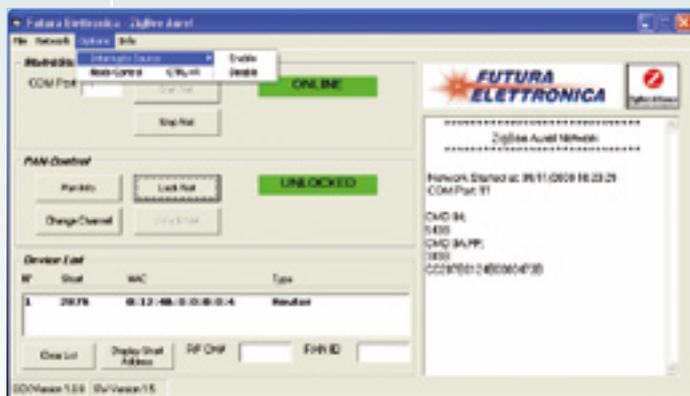
(mesh) che permette di raggiungere dalla base qualsiasi nodo, anche nel caso quest'ultimo si trovi fuori portata, rimbalzando da un nodo all'altro dalla sorgente alla destinazione. Questa importante caratteristica può essere utilizzata nelle reti Aurel ZigBee per la creazione di mesh composte anche da centinaia di nodi (le specifiche fornite da Aurel dichiarano mesh composte da un massimo di 200 nodi).

GESTIONE DEGLI INGRESSI DI ALLARME

Oltre ai quattro led menzionati nel paragrafo precedente la demoboard contiene diverse altre periferiche, tra le quali figura un ingresso di interrupt collegato ad un pulsante. La pressione del suddetto pulsante scatena un interrupt che, con una opportuna gestione a livello firmware, produce l'invio di un pacchetto dal nodo sorgente verso il coordi-

nator contenente una serie di informazioni, come il codice relativo all'evento e l'indirizzo short del nodo che l'ha inviato. Mediante una funzionalità aggiunta al software, il segnale di interrupt generato dalla pressione del relativo tasto sulla demoboard produce un segnale di allarme contenente anche l'informazione circa il nodo di provenienza; per abilitare i segnali di allarme bisogna selezionare la relativa opzione sotto la voce *Interrupt Source* del menu a tendina *Options*. Com'è facilmente intuibile, le due voci *Enable* e *Disable* servono ad abilitare e disabilitare questa funzione. Una volta abilitata la funzione, per verificarne il funzionamento, accendete e collegate alla rete un nodo qualsiasi (nell'esempio il nodo con indirizzo short 0x287B); completata l'operazione, premete il tasto collegato all'ingresso di interrupt (dei due presenti sulla demoboard, è quello situato vicino al dip-switch, mentre l'altro è il reset del modulo radio). La pressione del tasto produrrà il lampeggio di una spia all'interno del frame *PAN Control* e, più precisamente, di quello al disotto del led indicante lo stato della rete (Locked o Unlocked); il lampeggio sarà accompagnato dal classico "ding" (o altro suono di sistema assegnato all'evento) del sistema operativo Windows. All'interno della spia verrà segnalato anche l'indirizzo short del nodo dal quale proviene il segnale di allarme, come illustrato nella Fig. 6. L'output del comando sul coordinatore nel caso di un interrupt sul nodo locale, è ripor-

Fig. 5
Abilitazione
segnali di
allarme.



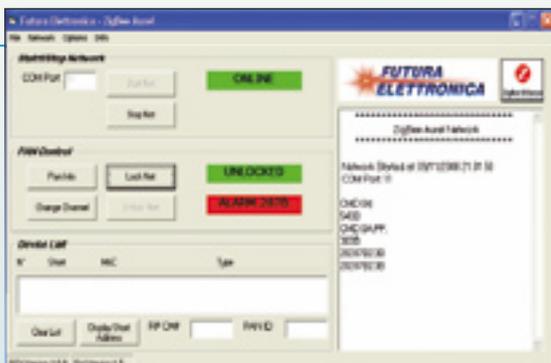


Fig. 6 - Gestione del segnale di allarme.

tato nella Tabella 3. Come si può vedere, il pacchetto contiene il codice identificativo dell'evento (0x20), l'indirizzo short del nodo sorgente ed infine lo stato degli I/O digitali. La gestione dell'evento interrupt è stata inserita insieme alla gestione di altri eventi specifici, come il network update o l'evento PanInfo. L'handler relativo all'evento è riportato all'interno del Listato 2.

Come si vede dal listato, l'evento viene gestito solo se il flag *F_Alarm* vale 1. Questo flag viene settato e resettato dall'opzione *Interrupt Source* del menu *Options* vista in precedenza. Se il flag è settato, si passa alla gestione dell'evento. Per prima cosa, tramite un'istruzione *if*, si riconosce se il codice dell'evento corrisponde a 0x20: in caso positivo si scrive sulla label *lblInterrupt* la scritta *Alarm* accompagnata dall'indirizzo short (in esadecimale) del nodo sorgente.

A questo punto si attiva un timer contenente la sub riportata nel Listato 3, che serve a generare il lampeggio del LED. Con queste semplici istruzioni è stato costruito un sistema che permette di inviare ed intercettare segnali di allarme wireless provenienti dai più disparati dispositivi, dai semplici switch ai sensori di presenza gas o fuoco collegati ad un microcontrollore che ne interpreti l'uscita ed in caso di superamento di una determinata soglia intervenga sull'ingresso di interrupt del modulo. La naturale evoluzione software è, in primo luogo, la creazione di una lista dei segnali di allarme con l'indicazione del nodo di

Tabella 3 - Output del comando sul coordinatore nel caso di interrupt sul nodo locale.

Output del comando su nodo locale
MsgType -> 1 byte
• 0x20 -> Alarm
indirizzo a 16bit del nodo remoto -> 2 byte
stato degli I/O digitali -> 1 byte

Listato 2

```
'Evento 0x20: Interrupt
If (F_alarm = 1) Then
  If (Asc(Mid(recString, 1, 1)) = &H20) Then
    lblInterrupt.Caption = "ALARM " & Hex(Asc(Mid(recString, 2, 1))) &
      Hex(Asc(Mid(recString, 3, 1)))
    TimerInterrupt.Enabled = True
    Beep
  End If
End If
```

Listato 2 - Gestione dell'interrupt.

Listato 3

```
Private Sub TimerInterrupt_Timer()
  F_Interrupt = F_Interrupt + 1
  If ((F_Interrupt Mod 2) = 0) Then
    lblInterrupt.Visible = True
  Else
    lblInterrupt.Visible = False
  End If
  If (F_Interrupt > 10) Then
    TimerInterrupt.Enabled = False
    F_Interrupt = 0
  End If
End Sub
```

Listato 3 - Codice utilizzato per la generazione del segnale di allarme.

provenienza e dell'ora; in un secondo momento, l'affiancamento di un sistema per l'invio di mail o di SMS. Con questa quarta puntata si conclude il corso sui dispositivi ZigBee Aurel XTR-ZB1. Abbiamo analizzato i componenti hardware e software del kit di sviluppo che permettono un facile approccio alla soluzione Aurel per la connettività ZigBee ed un veloce time-to-market nel caso di sviluppo di prodotti basati su questa tecnologia. Abbiamo analizzato in dettaglio il componente OCX ed il software demo fornito da Aurel per la gestione ed il controllo della PAN ZigBee. Infine, nella terza puntata abbiamo iniziato lo sviluppo di un nostro software di gestione scritto in Visual Basic 6, in grado di ottenere la lista degli indirizzi MAC e short dei nodi connessi alla rete, ma anche di ricevere informazioni sulla rete stessa e di operarne eventualmente il blocco. Ancora, in quest'ultima puntata abbiamo visto come è possibile operare per la gestione in remoto delle linee GPIO e di interrupt del nodo stesso. Gli argomenti affrontati dovrebbero essere sufficienti ad avviare il lettore allo sviluppo di proprie applicazioni ZigBee con la famiglia di moduli XTR-ZB1, ma se ciò non fosse sufficiente, nei prossimi numeri di *Elettronica In* presenteremo una serie di progetti pratici volti alla realizzazione di un sistema domotico simile al Velbus ma interamente wireless. ■

