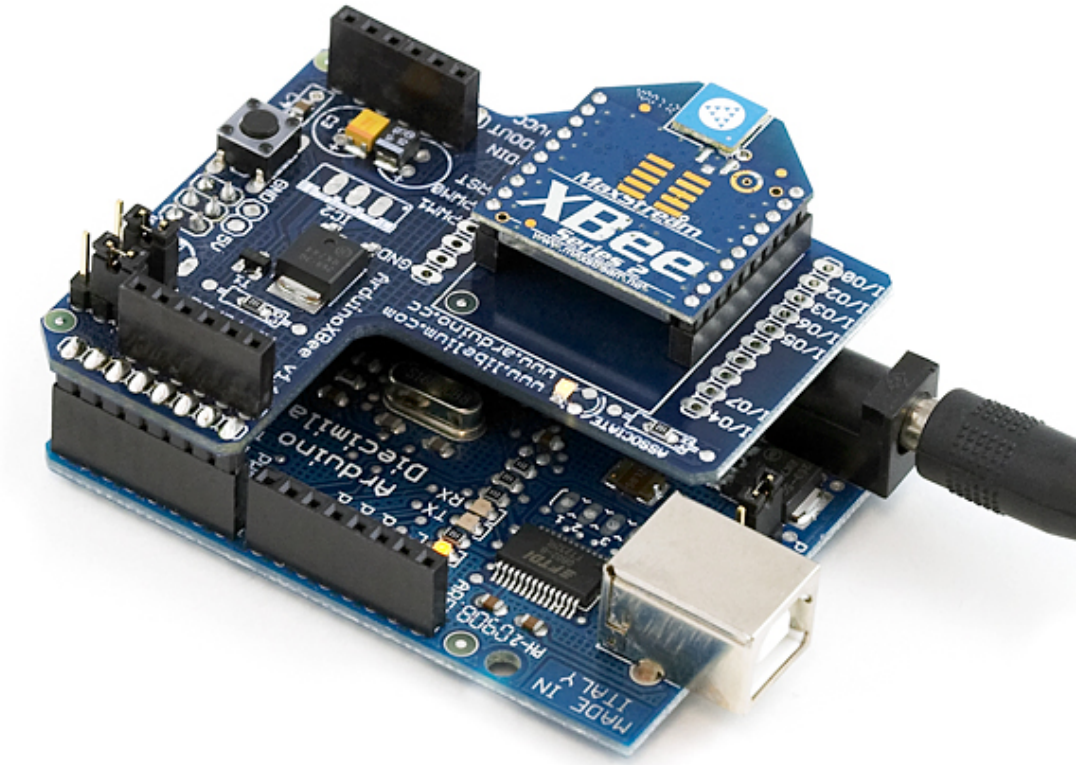
RETI MESH E ARDUINO

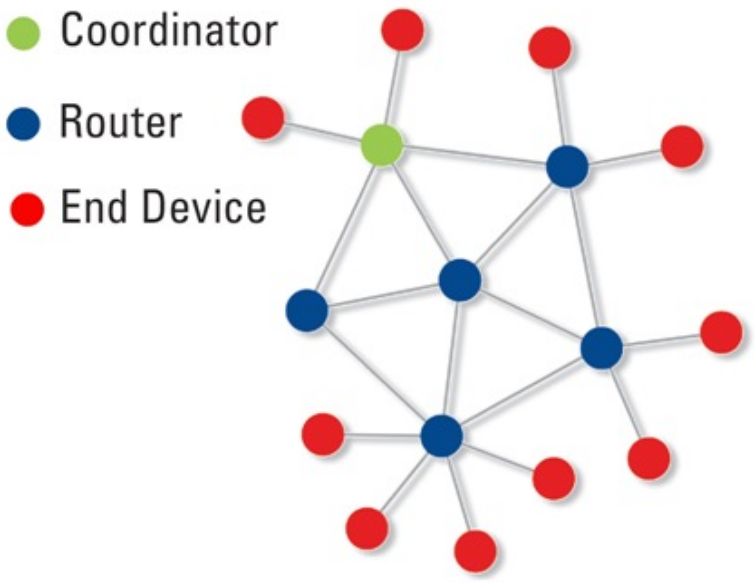


Indice:

1 Reti MESH

1.1 Cosa sono?

In telecomunicazioni con il termine “MESH” si intende una rete wireless che segue il modello delle LAN. A differenza però di una rete come possiamo definire quella di internet in questo caso parliamo di rete decentralizzata ovvero all’interno della stessa non abbiamo risorse centralizzate ma ben si un numero infinito di nodi che si comportano come ricevitori, ripetitori o trasmettitori. Ogni nodo quindi è equiparato ad un altro. Questa soluzione risulta quindi assai economica ma soprattutto molto resistente in quanto ogni nodo ha il solo compito di ripetere o trasmettere il segnale al nodo successivo. Qualora un nodo dovesse risultare inoperante la rete verrebbe ridefinita all’istante senza che ci siano delle interruzioni del servizio offerto. All’interno di queste reti troviamo dunque necessariamente uno e un solo nodo che si comporterà da coordinator e avrà il compito di istanziare l’intera rete. Poi sarà composta da nodi che si comporteranno come router o end device. Anch’essi possono operare sia in trasmissione che in ricezione ma anche come semplici ripetitori. L’intelligenza quindi delle reti mesh sta nel fatto che sono in grado in maniera del tutto automatico di determinare quale percorso dovrà compiere un pacchetto che deve andare dal nodo A al nodo C anche se il nodo A e il nodo C non possono comunicare direttamente in quanto sono fuori portata dei moduli radio utilizzati. E’ altresì estremamente facile espandere queste tipologie di reti poiché basterà inserire un nuovo modulo radio opportunamente configurato per operare sulla specifica rete mesh. Attualmente esistono diversi progetti di reti mesh che spaziano in diversi settori. Sicuramente uno dei progetti più in vista in Italia è Ninux. Progetto nato a Roma con l’idea di creare una rete comune a tutti, libera da canoni di sottoscrizione in cui ogni utente può condividere le risorse che meglio ritiene opportune. Questo progetto si basa proprio sulle reti mesh. Vale quindi tutto quello che abbiamo visto fino ad ora su dette reti.



1.2 Esempi applicativi

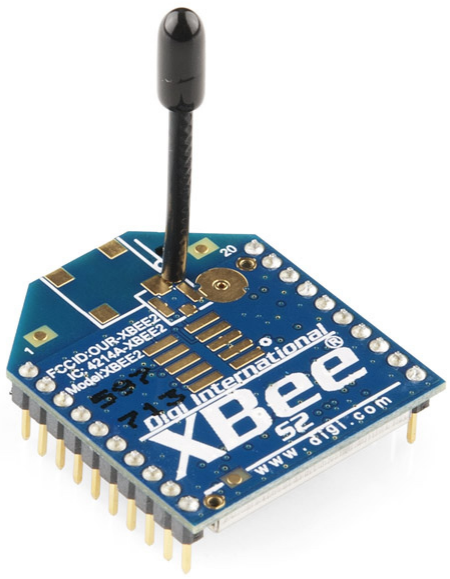
Nel nostro contesto vogliamo pensare un utilizzo pratico di tale rete per raccogliere i risultati di diversi sensori dislocati in un territorio. Si voglia per tanto considerare come ipotesi la seguente: Realizzazione di un applicativo in grado di mostrare le informazioni meteo relative ad una città. La città sarà suddivisa in quartieri e ogni quartiere conterrà un diverso numero di nodi aventi il compito di leggere dall'ambiente i risultati dei sensori.

A questa ipotesi si possono trovare diverse soluzioni applicative. Si potrebbe decidere di usare un servizio web in grado di raccogliere tramite l'uso di specifiche API i valori rilevati dai sensori dislocati geograficamente ma dotati di una connessione internet. Altra soluzione di certo un po' più economica ma anche più robusta è l'impiego di una rete mesh. In questo modo tutti i nodi del sistema sono in grado di comunicare tra di loro ma soprattutto decade il vincolo dal sistema centralizzato rappresentato dal servizio web citato nella soluzione precedente. Sarà quindi sufficiente dotare i nostri sensori di appositi moduli radio opportunamente configurati e sviluppare il portale di front-end destinato al pubblico. Questa soluzione risulterà quindi sicuramente più solida in quanto non compare nessuna dipendenza tra i vari nodi ed il servizio web. Qualora una delle parti che compone il sistema dovesse decadere la rete continua in ogni caso il suo lavoro.

2 Protocollo ZigBee

2.1 Standard del protocollo

ZigBee è uno dei protocolli più diffusi per la trasmissione di informazioni nel campo del Wireless Sensor Network. Attraverso l'uso di piccole antenne digitali a bassa potenza e a basso consumo il protocollo specifica una serie di profili applicativi che permettono la realizzazione di comunicazioni specifiche nel campo sensoriale. Tutte le antenne devono necessariamente rispettare lo standard IEEE 802.15.4 per le WPAN (wireless personal area network) La prima versione viene rilasciata nel 2004 e prende il nome di ZigBee 1.0 . A questo link è possibile trovare la documentazione rilasciata dalla ZigBee Alliance. Questo standard nasce con l'intento di voler essere più economico e più semplice rispetto alle altre tecnologie già affermata nel campo delle WPAN come per esempio il Bluetooth. Quest'ultimo citato è assai dispendioso in termini di energia elettrica assorbita rispetto ad un'antenna ZigBee. In seguito anche lo standard Bluetooth ha definito moduli low energy per cercare di equipararsi alle antenne ZigBee. L'unica implementazione attualmente in uso dello standard ZigBee si base su antenne operanti a 2.4 GHz. Si noti come un nodo ZigBee costi appena $ 1.10 rispetto ai $ 3 richiesti per un nodo Bluetooth. Salta quindi subito all'occhio come un nodo ZigBee sia più conveniente. Purtroppo però la maggior parte dei nodi ZigBee ha bisogno di una MCU che fa innalzare notevolmente il prezzo finale. Questo protocollo è quindi vincente quando si vuole inserire un nodo applicativo di tipo embedded in un wireless mesh network. Si noti però che tale adozione è consigliata solo quando si vuole ottenere un basso transfer-rate (velocità di trasferimento) e un basso consumo di energia elettrica. Si pensi che con la batteria di un singolo nodo si riesca ad alimentare ininterrottamente per due anni un nodo basato su standard ZigBee.



2.2 Xbee e protocollo ZigBee

Xbee è il nome scelto dall'azienda Digi per la loro serie di antenne che seguono lo standard ZigBee. In commercio esistono diversi produttori di antenne digitali ma noi soffermiamoci su quelle prodotte dalla Digi. Sono in commercio in due form-factor diverse. La più comune è la SMT (Surface Mount) ma esiste anche nella versione thru-hole in cui ogni componente sarà in vista e facilmente sostituibile. Attualmente in commercio esistono tre diverse serie di moduli Xbee. La S1 e la S2 sono principalmente ad uso domestico mentre la PRO principalmente è per uso professionale. S1 ed S2 destinati quindi ad un uso domestico hanno un range inferiore rispetto alla serie PRO ma hanno anche un costo inferiore. Indifferentemente dalla serie di moduli usati essi sono in grado di lavorare principalmente in due modalità. AT, meglio conosciuta come Transparent Mode, il dato catturato dal nostro sensore viene trasmesso direttamente sul pin di TX dell'antenna. Altra modalità è l'API mode in cui il dato trasmesso viene prima inglobato in un pacchetto. Possiamo quindi paragonare l'API mode al protocollo TCP che sappiamo essere connection oriented. L'AT mode invece è paragonabile al protocollo UDP che invece è connectionless. Ora riferendoci all'esempio descritto precedentemente se supponiamo di avere una rete mesh di sensori sicuramente sarà più sicuro utilizzare le antenne in modalità API in quanto ci garantisce che l'informazione viene inglobata in un pacchetto e che il pacchetto arrivi al nodo di destinazione in quanto il nodo trasmettitore chiuderà il canale di comunicazione solo ed esclusivamente quando riceverà l'ACK. Va però specificato che in entrambi le modalità è possibile instaurare comunicazioni di tipo point-to-point o point-to-multipoint. Siamo così in grado di controllare remotamente i pin di I/O di qualunque device che sia in grado di interfacciarsi con i moduli radio.

2.3 Vantaggi utilizzo del protocollo ZigBee

Dopo aver definito quindi le caratteristiche principali del protocollo siamo in grado di affermare che l'utilizzo del protocollo ZigBee nella realtà comporta diversi vantaggi. Il primo punto di cui ogni sviluppatore deve tener conto è il consumo energetico totale del nodo. Nel caso specifico in cui si utilizzino nodi che adottano antenne ZigBee notiamo come il consumo totale sia veramente ridotto al minimo. Altro punto fondamentale sta nella stabilità del protocollo. Supponiamo quindi di voler realizzare una comunicazione tra due nodi quindi di tipo point to point. Possiamo adottare diverse soluzione. La prima che sicuramente ci verrà in mente perché più conosciuta adotta il Bluetooth. Questi moduli sfruttano una semplice comunicazione seriale e quindi priva di standard di comunicazione. Possiamo definirla connectionless oltre che sicuramente più dispendiosa in termini di energia elettrica. Un'altra soluzione che possiamo adottare sono le antenne ZigBee, più economiche in termini di energia elettrica, più sicure in termine di trasmissione in

quanto nel protocollo sono implementati controllo sulla trasmissione ma più dispendiosi in termini economici poiché le antenne ZigBee rispetto a quelle Bluetooth necessitano di una MCU dedicata che innalza il costo finale del modulo ZigBee. Inoltre usando il protocollo ZigBee siamo in grado di instaurare connessione broadcast o point to multipoint. Questa soluzione non è possibile invece adottando moduli Bluetooth. E' quindi evidente che lo standard ZigBee rispecchi un po' il funzionamento delle reti LAN.

3 Xbee e configurazione

3.1 Software X-CTU

Fino ad ora abbiamo parlato in maniera astratta di cosa sono le reti mesh e di un possibile protocollo da adottare in tale rete. Abbiamo inoltre visto varie tipologie di antenne digitali. Abbiamo citato quindi le antenne Xbee prodotte dalla Digi. Inoltre abbiamo sostenuto che esistono varie modalità in cui queste antenne possono operare. Con quest'ultima affermazione sorge spontanea la domanda: Come si procede alla configurazione della modalità operativa di tali antenne? Abbiamo anche sostenuto che le antenne usanti il protocollo ZigBee devono avere una MCU di controllo. Le antenne che andremo ad analizzare sono quindi già dotate della MCU ed hanno un costo di $25. Dobbiamo quindi solo procedere alla configurazione di tali moduli. Per far ciò sfruttiamo il software che la casa produttrice di tali moduli ci mette a disposizione. Colleghiamoci al link della Digi ed effettuiamo il download del software. Una volte avviato ci troveremo di fronte alla main page (Fig 1.1) del software. Ora dobbiamo procedere a collegare il Dongle USB (Fig 1.2) con agganciato al di sopra il moduli Xbee da configurare. Il modulo può essere collegato al PC sia mediante il dongle, dispositivo che si vede in foto, oppure tramite un Arduino. In quest'ultimo caso occorre prima di tutto caricare sulla board uno sketch vuoto. Scolleghiamo dunque l'Arduino dal PC, agganciamo l'xbee shield con i ponticelli rivolti verso USB. Inseriamo il modulo Xbee e ricolleghiamo Arduino. A questo punto possiamo tranquillamente usare X-CT.

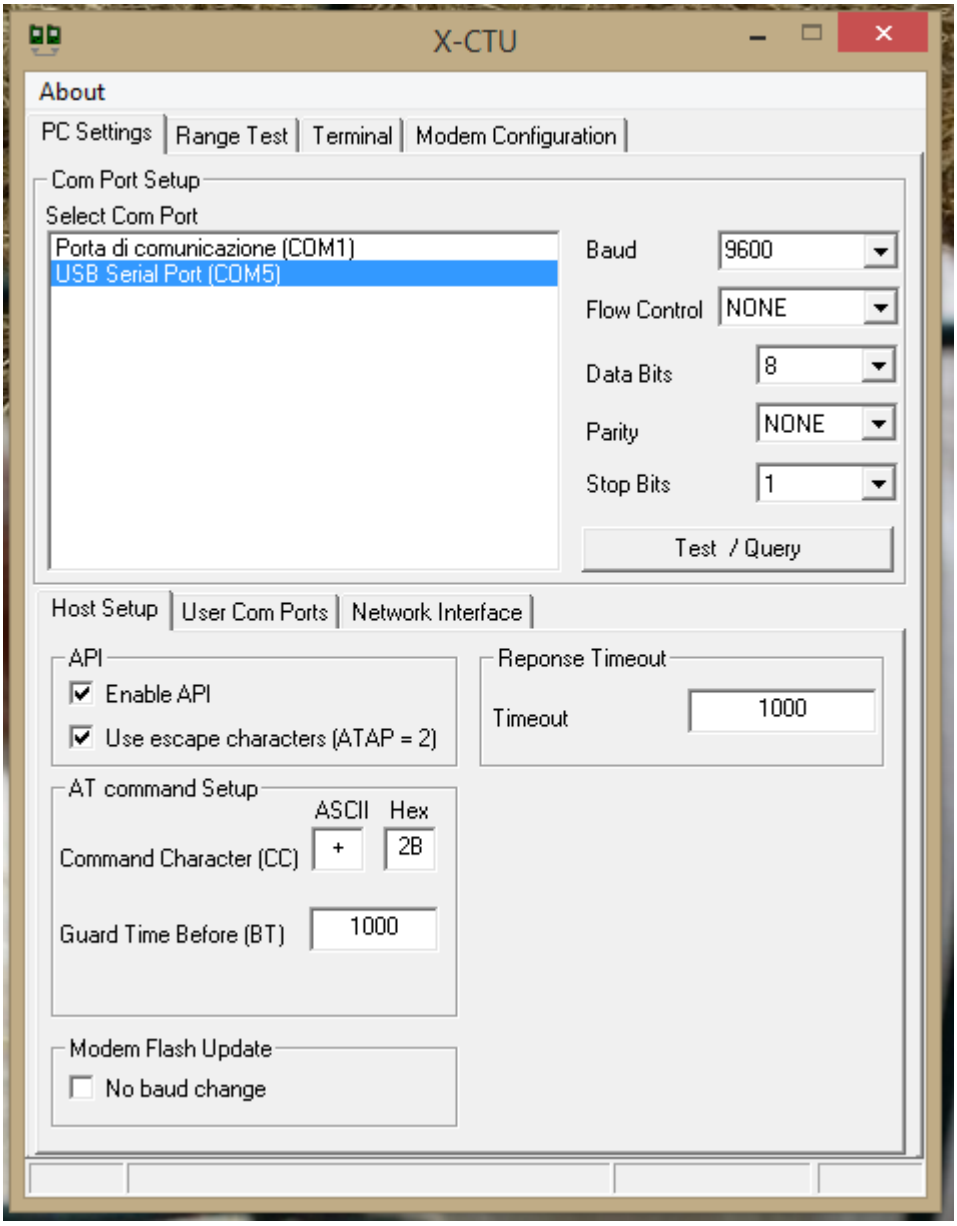


Fig 1.1 – Main Page software X-CTU

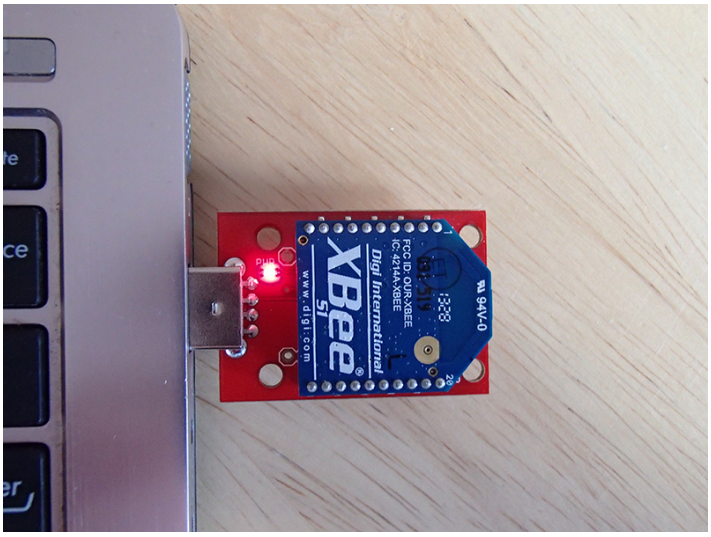


Fig 1.2 Dongle USB con modulo Xbee

Una volta selezionata la porta seriale con il quale comunicheremo con il device fisico , spostiamoci nella scheda "Modem Configuration". Da questa schermata è possibile leggere e scrivere tutti i parametri di configurazione del nostro modulo. Come prima cosa occorre scegliere quale firmware utilizzare. Il firmware è un insieme di istruzione che permette ad un componente hardware di dialogare con altri componenti hardware. E' proprio nella scelta del firmware che possiamo esplicitare se il device opererà in modalità AT o API. Inoltre sempre in questa fase dobbiamo stabilire se quel modulo sarà coordinator, router o end device. La definizione di queste tre modalità la rimando nel prossimo capitolo. Una volta scelto il firmware specifico per il nostro device possiamo andare a settare i valori specifici della rete che si vuol creare. Particolare attenzione va quindi prestata al campo PAN ID, DH, DL e AP (Fig 1.3). Quest'ultimo campo lo si trova qualora lavoriamo in modalità API. Il campo PAN ID corrisponde al nome della rete. E' quindi necessario che tutti i nodi che vogliono comunicare tra loro abbiano lo stesso ID. DH invece è la parte alta dell'indirizzo di destinazione. L'indirizzo è formata dal serial number del modulo. Esso però viene suddiviso in due parti che prendono il nome rispettivamente di Destination Address High (SH remoto) e Destination Address Low (SL remoto). Se vogliamo effettuare una comunicazione point to point sarà quindi necessario inserire in questo campo l'SH e l'SL del modulo remoto con cui vogliamo comuniare. Se invece vogliamo creare una rete point to multipoint in cui ogni singolo modulo è in grado di comunicare con molteplici altri nodi sarà necessario settare DH e DL rispettivamente a 0 e FFFF. Infine ultimo parametro che andremo ad analizzare è AP. Definisce se il modulo fa

uso o no della modalità API. Attenzione che qualora venga utilizzata la libreria Xbee è necessario settare tale valore a 2. Espressi tutti i parametri e scelto il firmware clicchiamo sul tasto Write. A questo punto verranno scritti i parametri ed il firmware sul modulo.

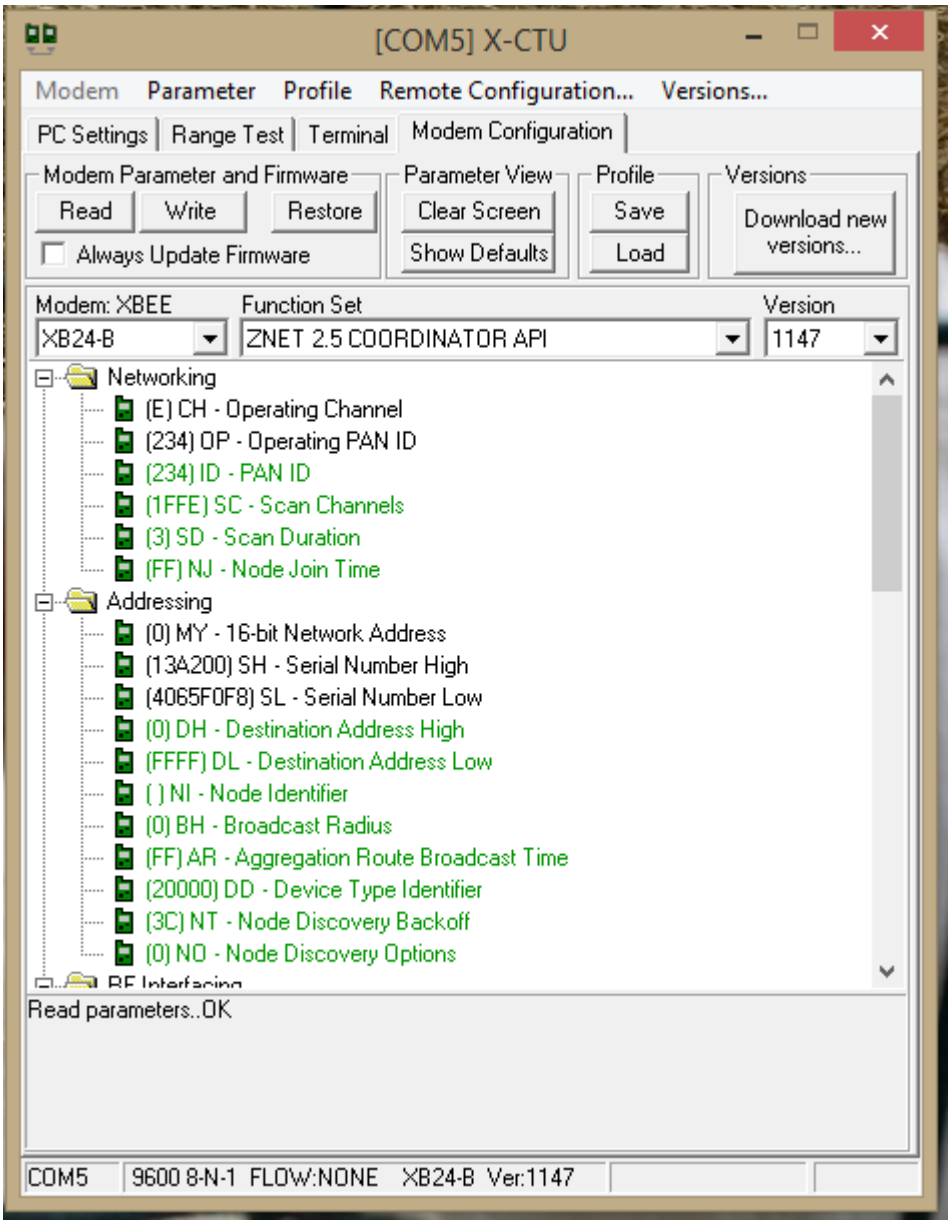


Fig 1.3 Dettagli Pan ID, DH, DL, AP

3.2 Definizione di coordinator, router ed end device

Abbiamo più volte citato questi tre nomi senza però mai analizzarli. Ma vediamo ora che significano. Un modulo a secondo del firmware che li viene uploadato al suo

interno può svolgere la funzione di coordinator, di router o di end device. Il coordinator come ci ricorda il nome avrà funzioni di coordinamento. In realtà possiamo qualificarlo come il responsabile della rete. E' colui che ha il compito di instaurare la rete. Senza questo modulo la rete non può esistere in quanto ogni altro modulo sia router o end device indifferentemente quando entrano in funzione, si presentano alla rete cercando il coordinator.. Oltre a queste funzioni si comporta anche come un normale router. Per router si intende che il modulo è in grado di ricevere/trasmettere ma anche ripetere le informazioni che sono giunte a lui. Infine abbiamo gli end device. Questi sono nodi della rete considerati più semplici poiché non ripetono nulla. Quello che ricevono se lo tengono se è indirizzato al broadcast altrimenti se contiene un indirizzo specifico e non corrisponde al suo viene scartato. Se ricordiamo quindi la capacità della rete mesh di disegnarsi da sola la strada che il pacchetto deve compiere per raggiungere due nodi distinti con questa configurazione notiamo come il modulo configurato in end device non sia in grado di fare da "ponte" tra due nodi. Gli end device di fatto nella pratica reale sono posti agli estremi delle reti.

3.3 Modalità operative (AT, API)

All'interno del manuale più volte sono state citate le due modalità operative. Abbiamo specificato che la scelta va effettuata in fase di upload del firmware nel modulo Xbee. Ma ora vediamo nel dettaglio le differenze sostanziali tra queste due modalità. AT viene utilizzata quando quel che si vuole trasmettere è di scarsa grandezza. Questa modalità è priva di interfacciamento e di fatto si opera con comandi veri e propri. AT precede il comando che si vuol far eseguire al modulo e sta a significare "Attention". E' una modalità quindi sicuramente molto potente in quanto è possibile personalizzare nel dettaglio ogni singolo parametro del modulo ma è anche più complessa in quanto si opera con comandi "primitivi". Un paragona potrebbe essere l'interfaccia grafica del sistema operativo. Se la si esclude il PC continua a funzionare però per una persona che non ha basi informatiche le cose si complicano. Nel caso invece dell'API mode anche qui parliamo di una modalità operativa per l'invio/ricezioni di dati. A differenza della precedente però quando si vogliono far eseguire dei comandi al modulo non importa entrare in modalità Command. Inoltre si dispone di API in grado di restituire la qualità del segnale, l'indirizzo locale e l'indirizzo remoto. Infine sempre questa modalità dispone di librerie studiate ad hoc per la trasmissione di dati sotto forma di pacchetti. Questa modalità ci assicura quindi un controllo sulla trasmissione, siamo quindi in grado di avere la certezze che il nostro dato è giunto a destinazione.

3.4 Configurazione modulo in modalità API

Il primo approccio a questi moduli radio conviene quindi farlo usando la modalità API. Come già detto mediante l'uso del software X-CTU andremo ad installare e configurare le antenne radio. Per una convenzione qualora vogliate seguire questa guida adottiamo i seguenti parametri:
-PAN ID: 234
-DH: 0
-DL: 0
-AP: 2
Ricordo che tali parametri valgono sia per il modulo coordinator sia per i moduli router. Scegliete la giusta versione del firmware, settate i parametri e cliccate sul tasto Write. Attendete l'ok dell'applicazione. A questo punto possiamo scollegare il modulo dal PC ed integrarlo nella nostra applicazione reale.

4 Presentazione ambiente applicativo

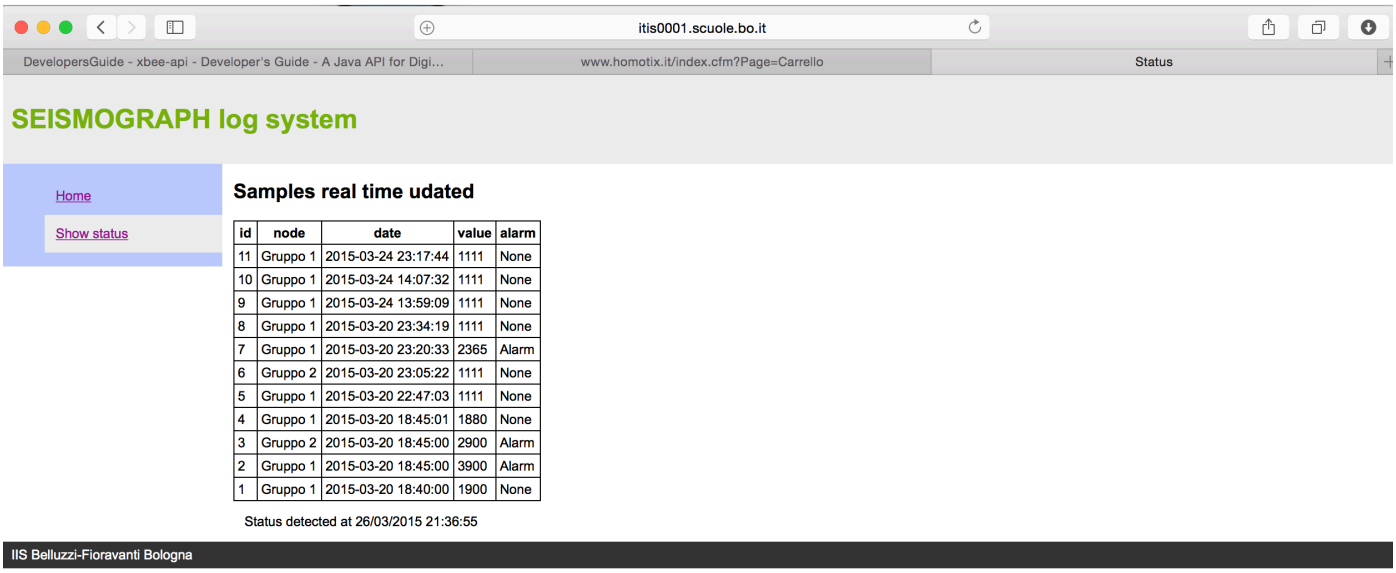
4.1 Descrizione dell'ambiente sismografo

Fino ad ora abbiamo intavolato diversi argomenti ma tutti teorici. Vediamo di rendere in maniera concreta quanto fino ad adesso appreso. Vogliamo simulare una situazione di analisi e lettura di valori raccolti da sensori sismografici posti per esempio su di un vulcano. Questo sistema ci garantisce di tener traccia degli avvenimenti sismografici della terra, ed in particolar modo in zone più rischiose di lanciare un allarme. Fatta questa prima premessa, si noti come necessariamente in una vasta zona di terreno sia necessario dislocare dei nodi di controllo dotati di sensori per la rilevazione delle scosse sismiche e di moduli radio per la comunicazione dei valori ottenuti. Ogni nodo quindi sarà equiparato ad un altro. Per tanto con quest'ultima affermazione siamo in grado di stabilire che la rete sarà composta da tanti nodi e che la comunicazione dovrà essere di tipo multipoint ma soprattutto che avremo la necessità di creare una rete di tipo mesh. Le reti mesh come già detto prima sono molto utilizzate per la realizzazione di architetture di sensori. Successivamente si avrà la necessità di avere un portale su cui poter consultare le letture dei sensori sia in tempo reale che lo storico. Ogni nodo quindi invierà all'applicativo oltre che il valore rilevato anche il proprio identificativo in modo tale che in fase di rielaborazione sia in grado di stabilire quale sensore ha rilevato cosa. L'applicativo sarà di tipo web e per tanto risponde agli standard delle reti TCP/IP. La rete mesh invece che andremo a realizzare risponde agli standard ZigBee. Due standard che definiscono delle regole all'interno di una data rete ma che tra loro due non possono comunicare. Una rete TCP/IP non potrà mai comunicare con una rete ZigBee se non per mezzo di un gateway (hardware o software). Un gateway è un componente che all'interno del nostro scenario rappresenta il punto di collegamento tra la rete ZigBee e la rete TCP/IP. Nel nostro

caso adottiamo una soluzione software ma ricordo che esistono anche soluzioni hardware. Recentemente Digi ha messo in commercio un gateway hardware che consente di far comunicare reti ZigBee con reti TCP/IP. Ogni nodo adotterà come MCU un Arduino. La scelta ricade su di esso poiché consente una facile integrazione con i moduli Xbee ma soprattutto consente l'interfacciamento con qualunque sensore in logica TTL e quindi anche con i sensori sismografici. Nella simulazione dell'applicazione andremo ad utilizzare dei potenziometri poiché il sensore sismografico ha un costo per unità elevato.

4.2 Presentazione applicativo PHP

Dopo la premessa iniziale vediamo brevemente come funziona l'applicativo web. Applicazione scritta in PHP e dotata di un DBMS MySQL. Il DB ha il compito di archiviare le letture e renderle permanenti nel sistema. L'applicativo ha due pagine. Una semplice home page di presentazione e un'altra pagina invece che mostra sotto forma di tabella l'archivio delle letture (Fig 1.4). Inoltre espone tre API. Le API sono dei set di istruzioni che concorrono al raggiungimento di un obbiettivo. Nel caso specifico mediante l'inserimento di una determinata URL sono in grado di registrare un nuovo nodo, inserire le letture dei sensori o testare il sistema. L'uso delle API è molto comodo in quanto in maniera molto sintattica sono in grado di far eseguire le operazioni di mio interesse. Ovviamente viene lasciato il compito al programmatore di sviluppare la funzione che dovrà essere eseguita alla chiamata dell'API.



id	node	date	value	alarm
11	Gruppo 1	2015-03-24 23:17:44	1111	None
10	Gruppo 1	2015-03-24 14:07:32	1111	None
9	Gruppo 1	2015-03-24 13:59:09	1111	None
8	Gruppo 1	2015-03-20 23:34:19	1111	None
7	Gruppo 1	2015-03-20 23:20:33	2365	Alarm
6	Gruppo 2	2015-03-20 23:05:22	1111	None
5	Gruppo 1	2015-03-20 22:47:03	1111	None
4	Gruppo 1	2015-03-20 18:45:01	1880	None
3	Gruppo 2	2015-03-20 18:45:00	2900	Alarm
2	Gruppo 1	2015-03-20 18:45:00	3900	Alarm
1	Gruppo 1	2015-03-20 18:40:00	1900	None

Fig 1.4 Archivio storico rilevazioni

Il DB oltre ad avere il compito di tener traccia in maniera persistente delle rilevazioni deve stabilire se un nodo è autorizzato o meno ad inserire la rilevazione. Per tanto ogni nodo al suo primo collegamento dovrà registrarsi in modo tale che verrà aggiunto al database. (Fig 1.5)

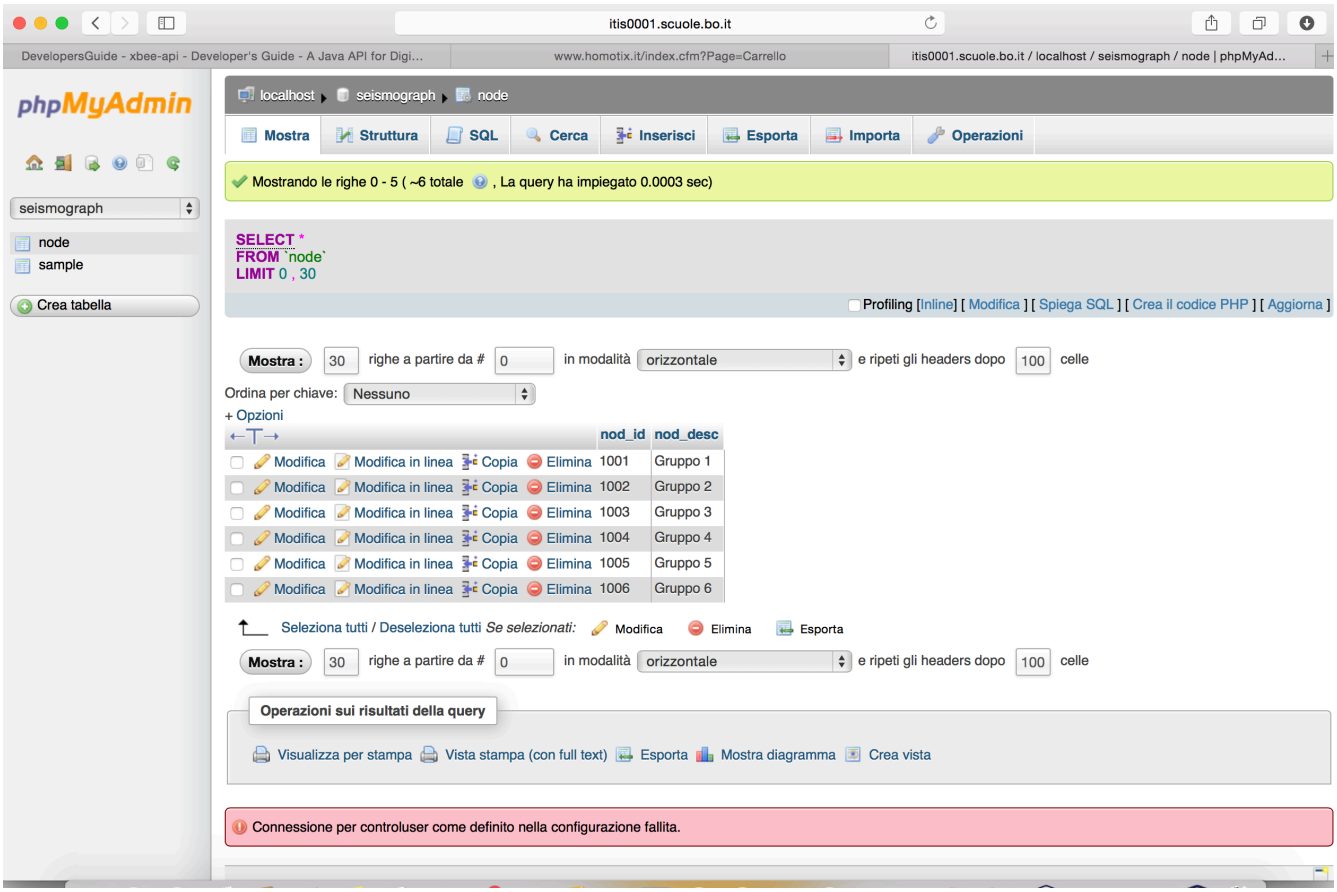


Fig 1.5 Tabella Nodi del DB

Infine vediamo due porzioni di codice che hanno il compito di inserire all'interno del DB i valori rilevati sul campo. La Fig 1.6 mostra il set di istruzione che vengono eseguite al momento in cui viene chiamata l'API, mentre la Fig 1.7 mostra i metodi che vanno a manipolare effettivamente il DB.

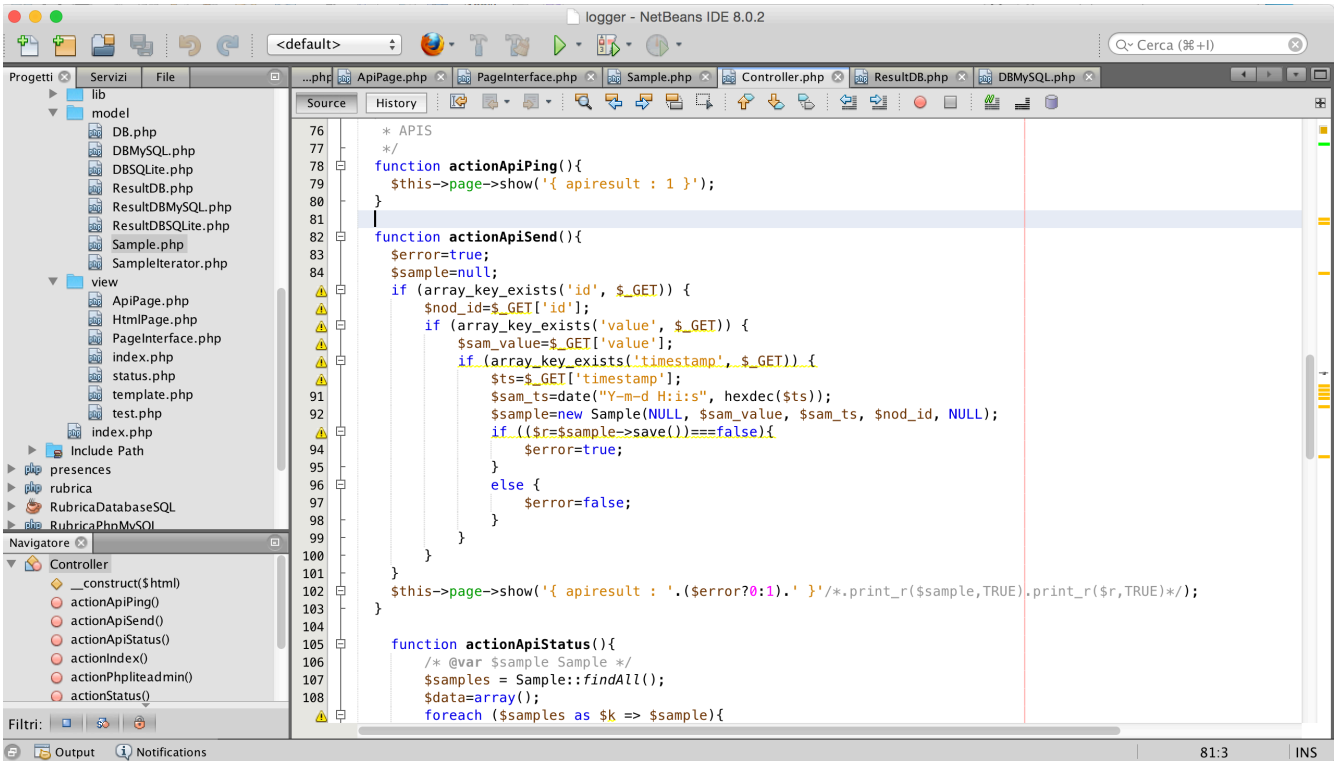


Fig 1.6 Istruzioni API send

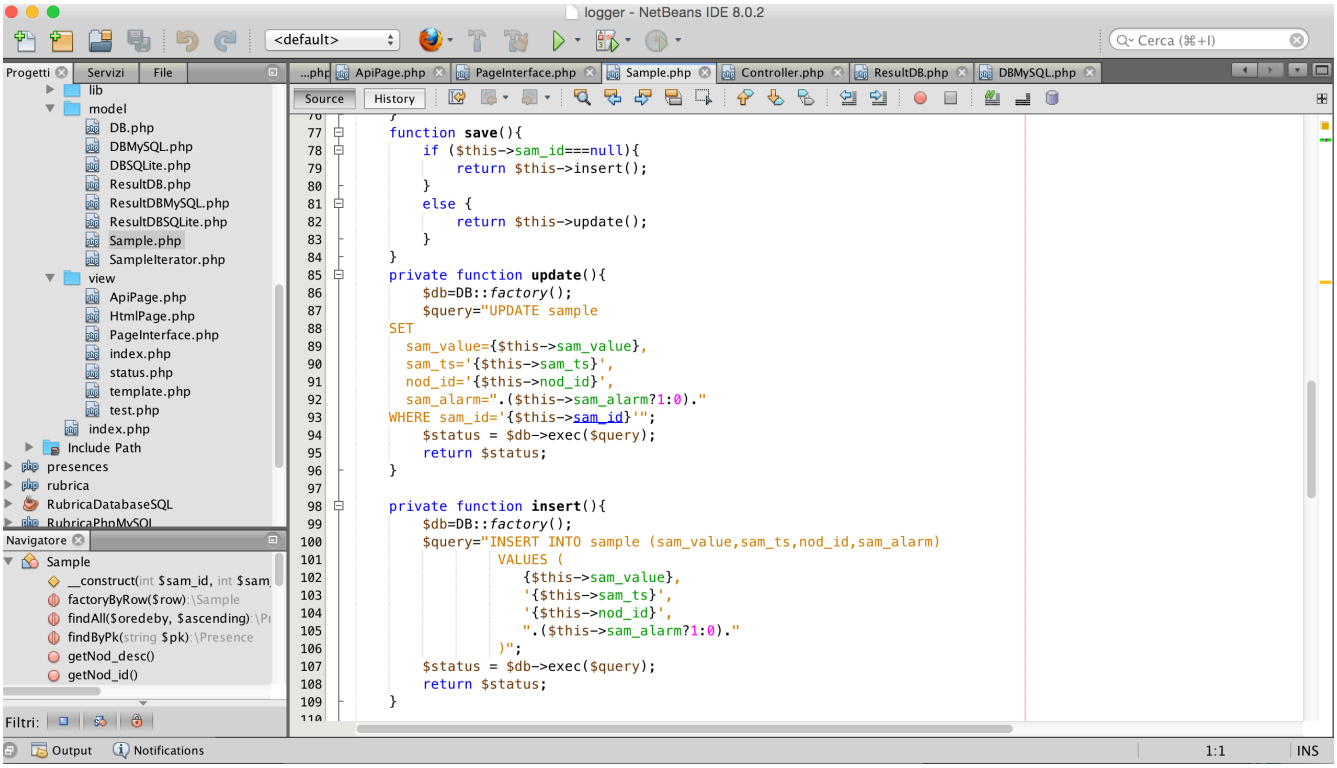


Fig 1.7 Istruzioni SQL

4.3 Presentazione applicativo Java Gateway

Nell'introduzione iniziale abbiamo detto che due reti che per loro natura vengono implementate su due protocolli differenti non possono comunicare tra loro se non mediante l'uso di un gateway che ha il compito di fare da interprete tra le due reti e rendere per tanto comprensibili le informazioni ad entrambi le reti in gioco. Nel nostro caso ci avvaliamo dell'uso di un gateway software. Il programma è realizzata in Java ed ha il compito di ricevere da seriale le informazioni dalla rete ZigBee e successivamente mediante l'uso delle API descritte precedentemente comunicare le informazione alla Web Application. SI noti come l'applicazione utilizzi il canale seriale per comunicare con la rete ZigBee. Questo perché facendo uso di un dongle USB con installato sopra il modulo Xbee , viene rilevato dal PC come una comune seriale. In realtà tutto ciò ci avvantaggia poiché in Java esistono diverse libreria per la comunicazione seriale. Recentemente è stata rilasciata la Library Jssc che facilita notevolmente la comunicazione seriale. Di fatto viene instaurata una comunicazione su di una porta seriale. Nella Fig 1.8 vediamo la connessione sulla porta seriale e l'esecuzione dell'API mentre nella Fig 1.9 vediamo il metodo che viene richiamato durante l'esecuzione dell'API che restituisce il timestamp in formato unix time esadecimale. Il programma segue la scomposizione model, view, control. Inoltre tutte le operazioni di connessione ed esecuzione API sono inserite all'interno di un loop infinito in modo tale che finchè non si chiude il programma resta in ascolto. Questa prima versione è priva di grafica ed è in grado di eseguire solo l'API di invio dati. Particolare attenzione va fatta al timestamp. L'API accetta un timestamp solo in formato esadecimale. E' necessario per tatno eseguire la conversione. In questo caso abbiamo la classe Date che mediante il metodo getTime() ci restituisce i millisecondi passati dall' 1/1/'70. Infine il metodo toHexString() converte il long int in una stringa esadecimale. In questo modo siamo in grado di passare un valore accettabile all'applicativo web.

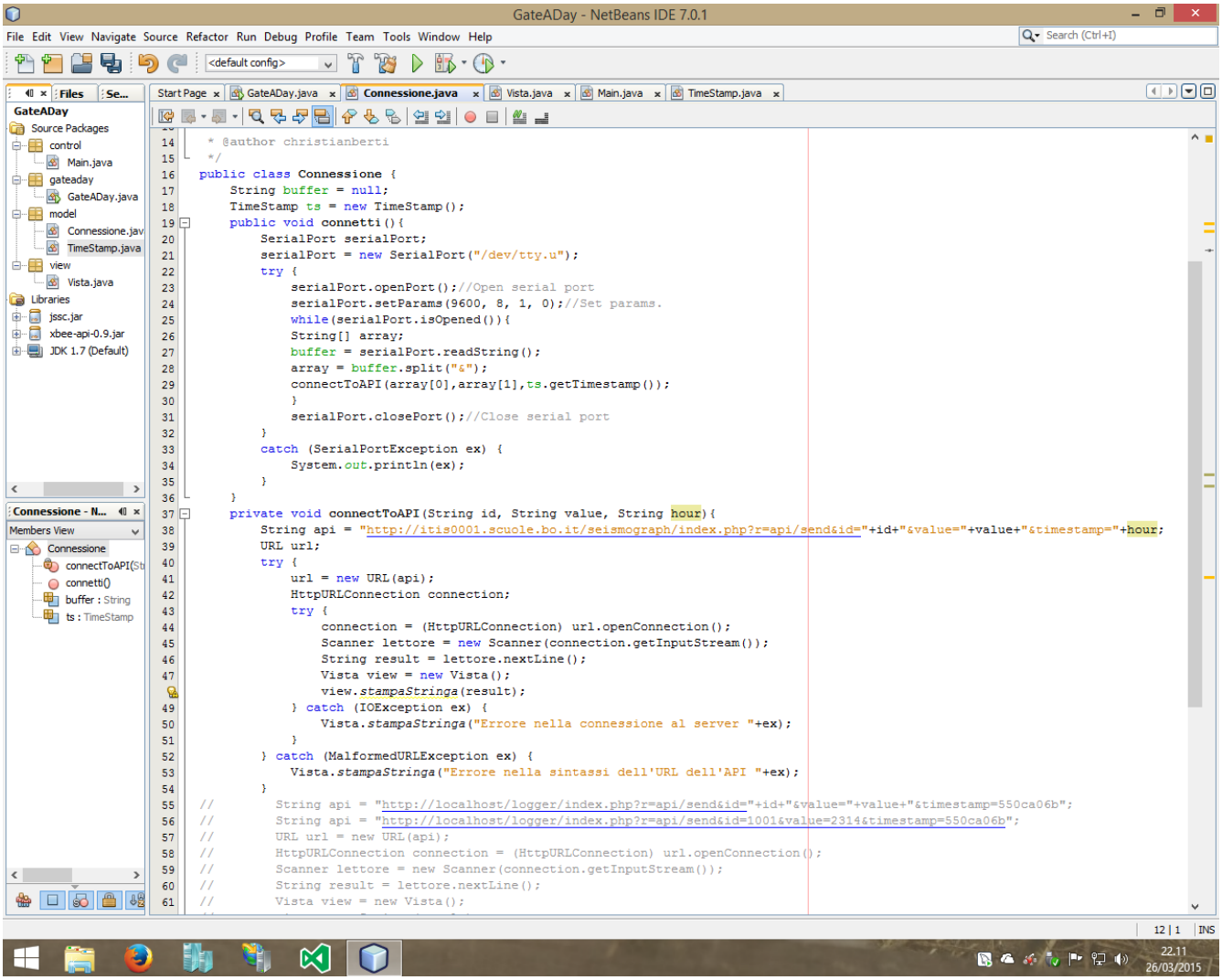


Fig 1.8 Connessione seriale ed esecuzione API

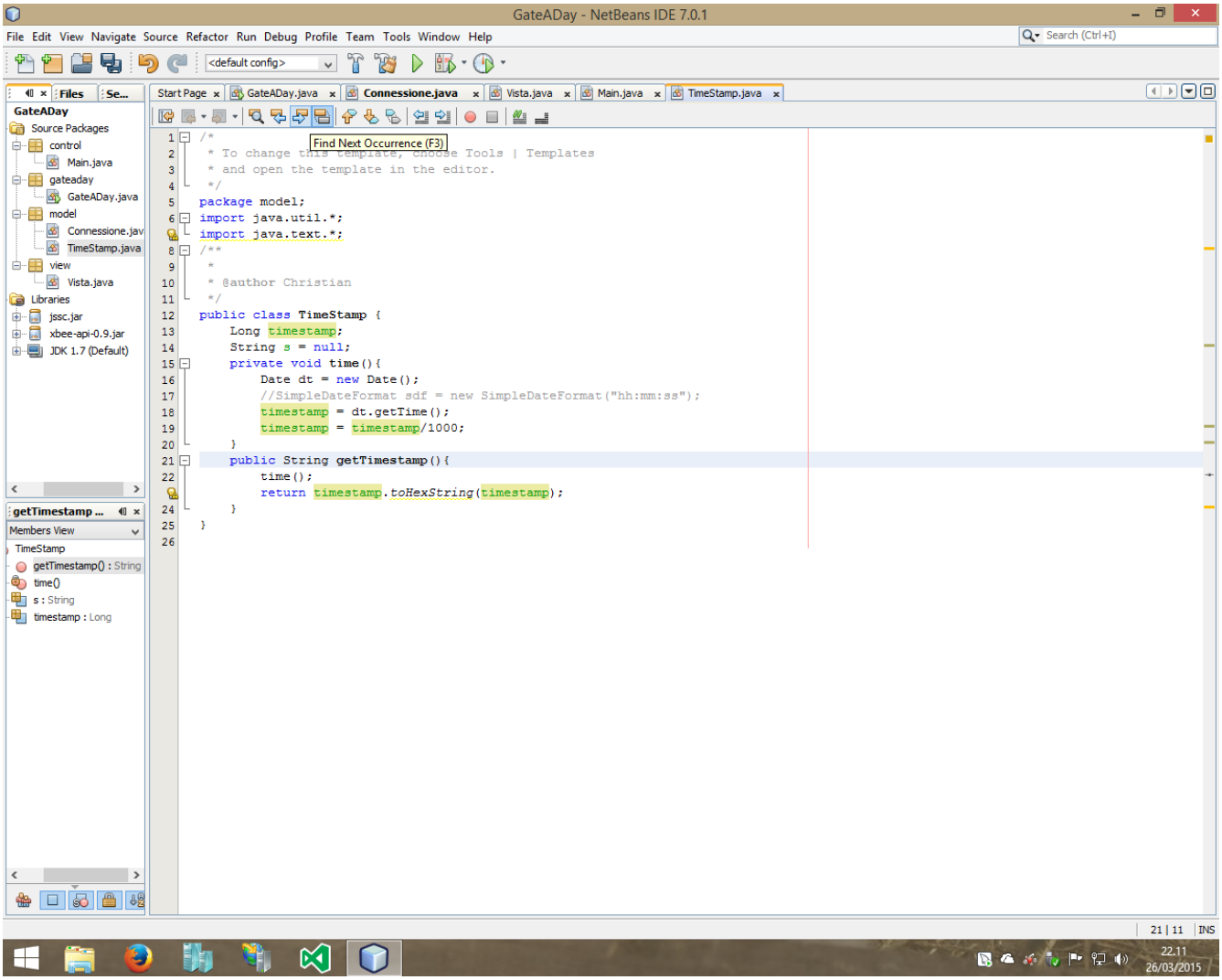


Fig 1.9 Timestamp in formato unix time esadecimale

4.4 Presentazione libreria xbee-arduino

Ora cominciamo a scendere più nel pratico. Abbiamo già sostenuto che nel nostro progetto utilizzeremo Arduino poiché è di facile comprensione, è compatibile con i moduli Xbee ed è compatibile con i sensori in logica TTL. Abbiamo anche definito che i moduli saranno configurati per operare in modalità API. A questo punto dobbiamo cominciare a pensare come realizzare il nostro codice Arduino. Possiamo metterci li e scrivere migliaia di righe di codice che ci consentono di spedire le informazioni, controllare che il pacchetto sia giunto a destinazione, ottenere l'indirizzo remoto, verificare la qualità del segnale oppure usare una comoda libreria dal nome Xbee-Arduino che mette a disposizione tutto un insieme di funzioni che ci saranno di grande aiuto. Essa può essere utilizzate sia nel caso si operi in modalità AT che in modalità API. Un grande vantaggio che offre tale

libreria è la capacità di definire la forma che dovrà avere il nostro pacchetto da spedire. Sarà quindi composto da una parte fissa e da una parte variabile, che conterrà effettivamente le informazioni e che prende il nome di payload. Nella parte fissa sono contenuti gli indirizzi più i soliti caratteri di CR e LF. A questo link si trova la documentazione dell'intera libreria. Nel nostro specifico caso sarà necessario analizzare solo alcuni metodi. Ovviamente prima di tutto occorre importare nello sketch la libreria scaricata poiché non è una libreria nativa di Arduino. Per convenzione d'ora in poi con xb indicherò un oggetto della classe Xbee. Il metodo xb.send(packet) consente di inviare il pacchetto specificato in rete, il corrispondente metodo xb.read() consente di leggere un pacchetto che arriva nella rete. Infine il metodo xb.getResponse() consente di ottenere una risposta sull'avvenuta consegna del pacchetto al destinatario. Attorno a questi tre metodi girerà l'intero nostro sistema. Si consiglia inoltre di dare un'occhiata agli sketch di esempio che troverete come al solito all'interno della cartella degli esempi di Arduino. Diamo un'occhiata alle sezioni più importanti del codice. La prima cosa che notiamo dallo sketch di esempio è la creazione di un nuovo oggetto di classe xb, la definizione del nodo che dovrà ricevere il pacchetto ed infine la definizione del formato del pacchetto. (Fig 2.1)

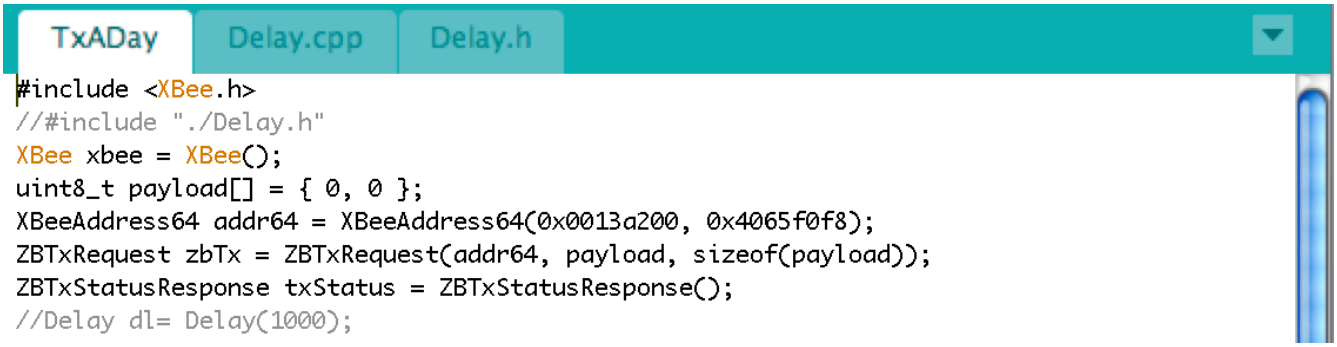


Fig 2.1 Creazione oggetti classe Xbee

L'oggetto zbTx è il cuore della trasmissione. E' la forma che assumerà il nostro pacchetto. E' composto da tre parti di cui due fisse. La prima che nell'esempio prende il nome di addr64 contiene l'indirizzo in base 64 del nodo ricevitore, la seconda parte che prende il nome di payload è la parte del pacchetto vera e propria che contiene le informazioni. Payload è una variabile che sta ad indicare il tipo di dato in byte. Nella fattispecie è un dato a due byte.

4.5 Test comunicazione coordinator – router

Dopo aver guardato il reference della libreria passiamo a testare l'effettiva comunicazione dei moduli. Per far ciò possiamo tranquillamente usare gli sketch di

esempio. Differenziamo però i led giacché l'esempio utilizza lo stesso led per indicare sia un errore che l'avvenuta comunicazione. Correggiamo inoltre la dichiarazione del pin riguardante la lettura analogica. Come si può notare manca la A di analog davanti al 5.

4.6 Attività pratica

Viene fornito ora uno schema a blocchi del sistema (Fig 2.2) che si vuole realizzare. Seguendo quanto appreso fino ad' ora e con l'aiuto dei tutor/mentor vediamo di realizzare insieme questo sistema.

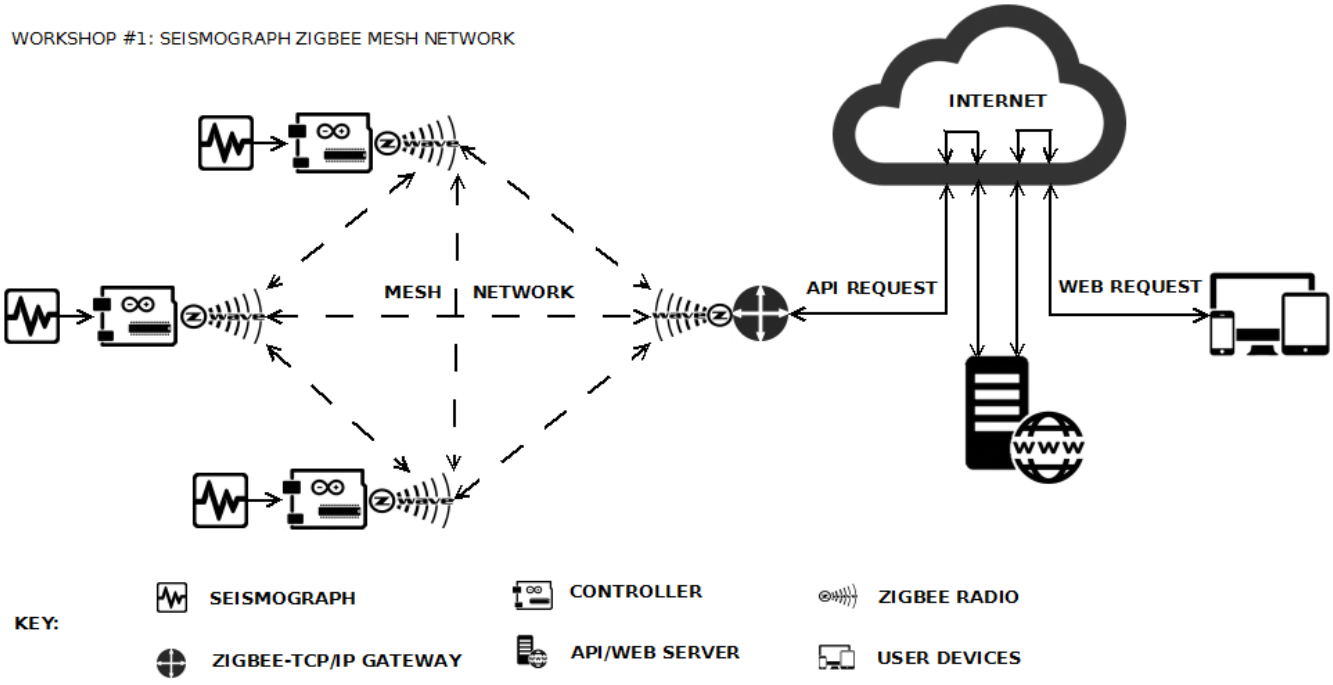


Fig 2.2 Schema a blocchi

...E ora mettiamoci al lavoro!