ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
SCUOLA DI SCIENZE
CORSO DI LAUREA IN INFORMATICA

RDNet Network

Protocollo di comunicazione tra Controller e PC tramite cavo LAN

Relatore:

Dott. Ivan Lanese

Presentata da:

Simone Bertolini

Correlatore:

Ing. Giordano Porta

II Sessione

Anno accademico 2014/2015

Sommario

RINGRAZIAMENTI

Desidero ringraziare tutti coloro che mi hanno aiutato nella stesura della tesi con suggerimenti, critiche ed osservazioni: a loro va la mia completa gratitudine.
Ringrazio innanzitutto il Dott. Ivan Lanese, Relatore, e l'Ing. Giordano Porta, Correlatore.
Ringrazio inoltre i miei colleghi, che hanno saputo ascoltare ed interpretare le mie esigenze facilitando le mie ricerche.
Vorrei infine ringraziare le persone a me più care: la mia famiglia, Giulia, gli amici più stretti, a cui dedico questo mio lavoro.

1. INTRODUZIONE

Questo documento presenta la relazione di una tesi in azienda svolta nel periodo tra settembre e novembre dell'anno 2015 presso la ditta RCF spa a Reggio Emilia.
La tesi consiste nell'apportare migliorie ad un software preesistente, che opera nel campo di dispositivi audio di diverse tipologie (es casse e mixer), i quali normalmente vengono utilizzati nel sistema d'impianto audio in un concerto. Si sta quindi parlando di dispositivi molto complicati, che non hanno nulla a che vedere con le semplici casse usualmente utilizzate negli ambienti domestici per ascoltare musica.
Il progetto di base è denominato RDNet e rappresenta un ambiente che consente di collegare tre principali componenti: un PC, i dispositivi audio ed una macchina, controllore, che svolge una funzionalità di "ponte" tra questi due apparati. Il collegamento di queste parti risulta essere fondamentale nel momento in cui si ha un grande numero di dispositivi da testare e settare secondo le proprie esigenze. Purtroppo, queste operazioni si rivelano molto lente se effettuate per mezzo dei tasti presenti sui dispositivi stessi, i quali permettono unicamente di agire sul singolo, risultando inoltre meno intuitivi di un'interfaccia utente lato PC.
Al fine di apportare modifiche al software, è risultato fondamentale appropriarsi delle conoscenze globali del sistema in considerazione, dal livello più basso, quello componentistico, a quello più alto, tutto ciò che riguarda direttamente il software stesso.
L'obiettivo principale sviluppato all'interno di questa tesi, consiste nella creazione di una nuova ed importate funzionalità, che ha fortemente agevolato l'utilizzo di questo software.
La scelta di sviluppare questo argomento è nata dall'esigenza dell'azienda di rendere maggiormente flessibile il software e, soprattutto, di incrementare la distanza all'interno della quale il sistema può operare.

La nuova funzionalità sopracitata rappresenta l'implementazione del protocollo di comunicazione tra controllore e PC mediante l'utilizzo di una connessione Ethernet.
La spiegazione di come è stata affrontata questa modifica lato software, l'architettura sulla quale è stata sviluppata (ovvero l'architettura di RDNet) e quali vantaggi ha portato, vengono dettagliatamente spiegati nei prossimi capitoli.
Questo documento spiegherà inizialmente chi è l'azienda RCF, mostrando un breve sommario sulla sua storia. Passerà poi ad introdurre il sistema RDNet, elencandone dettagliatamente le sue funzionalità. Seguirà la spiegazione articolata di tutti i componenti, partendo dai dispositivi audio, attraversando i controllori per poi terminare con il software PC.
Quest'ultimo argomento non verrà descritto nella sua totalità (non essendo questa una guida software), ma saranno prese in considerazione e dettagliatamente spiegate tutte le parti che hanno avuto un ruolo fondamentale nello sviluppo di questa nuova funzionalità.

1.1 RCF cos'è?

RCF è un azienda appartenente al settore metalmeccanico, in particolare all'industria meccanica ed elettronica della provincia di Reggio Emilia.
RCF è divenuta nel corso degli anni uno dei brands leader nel mondo nella progettazione, produzione e commercializzazione di prodotti e sistemi ad alta tecnologia per l'Audio Professionale e Sonorizzazione Pubblica.
L'offerta di RCF nel tempo si è ampliata e diversificata per soddisfare tutte le necessità di amplificazione sonora e musicale: da singoli prodotti a soluzioni complete, da piccoli impianti a progetti complessi e di grandi dimensioni.

RCF commercializza i propri prodotti sui mercati esteri attraverso le sedi commerciali in Inghilterra, Germania, Francia, Spagna e Stati Uniti. Nei restanti paesi del mondo, RCF opera tramite un consolidato e capillare network di distributori.
Al giorno d'oggi RCF possiede oltre 250 dipendenti e commercializza i propri prodotti in circa 100 paesi nel mondo.

La storia di RCF inizia nel 1949 con la produzione di microfoni, apparecchi per P.A. (Power Amplifier, ovvero amplificatori che dato un basso segnale audio, lo amplificano generando un segnale riproducibile dagli altoparlanti) e trasduttori.

Negli anni ’50 e ’60 RCF si afferma velocemente come produttore OEM (Original Equipment Manufacturer, ovvero i cosiddetti fornitori di grandi aziende) di alcuni componenti, quali trasduttori, diventando così la prima azienda italiana a costruire un amplificatore da 300 W e la prima in Europa a realizzare un Centro di Ricerca di alto livello per lo studio dei trasduttori.Presto i prodotti firmati RCF vengono molto apprezzati in tutto il mondo e per oltre
trent’ anni i suoi woofers, midrange e driver a compressione vengono utilizzati da produttori di diffusori, molti dei quali devono la loro attuale fama all’alta tecnologia RCF.

Con il passare degli anni la produzione di apparecchi elettronici, quali amplificatori e diffusori, viene ampliata; queste componenti vengono vendute singolarmente e anche come parte di grandi sistemi per aeroporti, hotel, scuole, stazioni ferroviarie, stabilimenti industriali e luoghi pubblici.

Partendo dalla tecnologia dei trasduttori, negli anni '80 RCF inizia a realizzare casse acustiche professionali.

Nel 1996 l’azienda lancia la Serie ART: diffusori acustici dotati di

amplificatore incorporato. Il successo di questa tecnologia, utilizzata all'epoca da pochi costruttori al mondo, colloca immediatamente RCF fra i maggiori produttori di diffusori a livello internazionale.

Nel 2004 inizia una scalata verso il successo per RCF, durante la quale ogni anno vengono progettati, sviluppati, realizzati e messi sul mercato un considerevole numero di nuovi prodotti, frutto di accurati studi ed innumerevoli ore di test. Tra questi vi è anche l'innovativa serie TT+ High definition Touring and Theatre che ottiene subito un grande successo grazie agli amplificatori equipaggiati con DSP (Processore di segnale digitale) e alla tecnologia dei trasduttori di altissimo livello.

Tre anni dopo, nel luglio del 2007, la capogruppo RCF Group entra in Borsa e, attualmente, è quotata sul segmento standard del mercato MTA.

Nel 2008 viene lanciata la nuova ART 7 Series, l'evoluzione della famosa serie ART, questo rappresenta un passo in avanti fondamentale nella tecnologia dei diffusori attivi. Tutt'oggi questa tecnologia continua ad essere studiata e testata, producendo di conseguenza un continuo miglioramento dei prodotti RCF.

1.2 RDNet

RDNet è un sistema di monitoraggio e controllo in tempo reale di dispositivi audio mediante l'utilizzo di un computer con sistema operativo Microsoft Windows.
La rete è costituita da tre parti principali: dispositivi audio, controllore, PC.

I dispositivi audio sono connessi ad un controllore che svolge una funzione routing (instradare le comunicazioni) tra dispositivi ed il PC.
Questo sistema dà la possibilità all'utilizzatore di sorvegliare lo stato (temperatura, valori dei segnali audio, ventole di raffreddamento) dei dispositivi collegati, di alterarne il funzionamento, consentendo la modifica di parametri quali ad esempio il volume di uscita, muting, curva di filtraggio, parametri di compressione, etc.
Il funzionamento di tutto questo apparato avviene attraverso un flusso continuo di dati, raccolti dal controllore che analizza i dispositivi, verso il PC. Su quest'ultimo è presente un software che elabora lo streaming dei dati in input ed aggiorna l'interfaccia grafica, visualizzando in modo comprensibile le informazioni ricevute. Dal software possono anche essere generati comandi (ad esempio il mute); questi vengono spediti al controllore e successivamente applicati mediante una registrazione firmware sui dispositivi.
Lo streaming di dati è da considerarsi in tempo reale perché le informazioni relative allo stato di funzionamento delle macchine vengono acquisite dal sistema in un periodo temporale che risulta essere inferiore alla variazione delle grandezze fisiche osservate.
La connessione tra PC e controllore può essere stabilita mediante USB (2.0), RS232 (seriale) oppure Ethernet.
La scelta di quale dei tre metodi utilizzare deve essere effettuata sulla base della distanza presente tra le due componenti e dell'eventuale necessità di modificare dei settaggi all'interno del firmware del controllore.
Da una parte l'Ethernet consente di raggiungere maggiori distanze, ma dall'altra possiede delle limitazioni ai comandi da inviare al controllore.
Infatti, tutti i comandi che hanno come obiettivo la modifica di impostazioni riservate del controllore, come per esempio Mac-Address ed indirizzo IP, posso solamente essere inviati dal software se connesso in USB oppure RS232. La limitazione nei

confronti di Ethernet è voluta al fine di aumentare il livello di protezione.

1.2.1 Struttura della rete

La struttura della rete è stata studiata per consentire:

- Il riconoscimento della posizione di un'unità remota rispetto alle altre;
- Il funzionamento di dispositivi remoti anche in assenza di alimentazione;
- Il funzionamento di un dispositivo remoto in caso di guasto;
- Il supporto fino a 32 unità remote per linea di collegamento;
- Il monitoraggio periodico delle unità remote.

Da questi prerequisiti la struttura della rete ideale risulta pertanto una rete ibrida, frutto dell'unione di una configurazione a stella e di una configurazione a bus [1], dove il nodo centrale della stella, ovvero il master, ha un comportamento similare a quello di un router ed i nodi sono invece suddivisi in N sottoreti a bus di tipo multi-drop.
Pertanto ogni ramo della stella non è composto da un singolo elemento, bensì da un insieme di nodi che compongono una sottorete. Un esempio di rete RDNet viene illustrato in figura 1.2.
Il master svolge una funzione di smistamento dati, permettendo la comunicazione tra una rete esterna e le sottoreti presenti. Il master può ricevere comandi da una rete esterna o può generarli autonomamente e comunica con i singoli slave delle sottoreti con un ciclo di comunicazione periodica e sequenziale.
La tipologia della sottorete permette la connessione di dispositivi remoti in Daisy Chain e quindi facilita il cablaggio da parte degli installatori.
Daisy Chain è uno schema di cablaggio in cui più dispositivi sono collegati insieme in sequenza o in un anello. La struttura risulta essere simile a quella mostrata in figura 1.1.

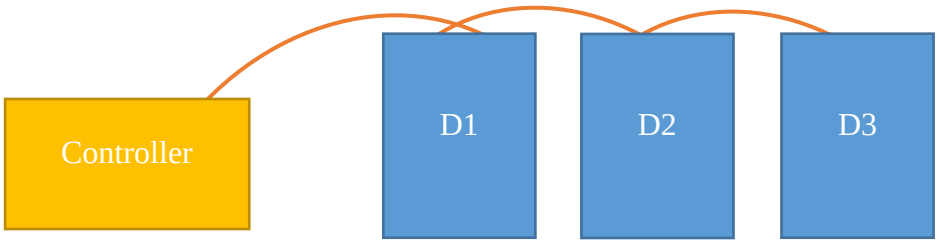

Fig. 1.1

Esempio di collegamento in Daisy Chain

Ogni sottorete è realizzata mediante il layer fisico EIA-485 [2] con connessione 2-wires, il quale specifica soltanto le caratteristiche elettriche del trasmettitore e del ricevitore, ma non indica, né raccomanda, alcun protocollo per la trasmissione dei dati. Il collegamento di ogni unità remota avviene mediante uno stub, ovvero una rete locale in cui esiste un solo router. In questo tipo di rete tutto il traffico diretto all'esterno deve necessariamente passare per l'unico router presente.
Il protocollo utilizzato adotta quindi una filosofia Master - Slave.
Nella comunicazione tra PC e controllore il primo dispositivo è master ed il secondo slave.
Invece nella comunicazione tra controllore e unità remote, il primo si comporta da master ed il secondo da slave.
Il controllore può quindi comportarsi da master o da slave a seconda dell'unità con cui comunica e deve pertanto svolgere le seguenti attività:

- Analizzare lo stato delle sottoreti, ovvero individuare in modo univoco le unità remote;
- Acquisire in maniera autonoma lo stato di funzionamento delle unità remote comportandosi da concentratore;
- Svolgere una attività di bridge nella comunicazione tra Host PC ed unità remote delle diverse sottoreti;
- Inviare lo stato di funzionamento globale della rete all'Host PC.

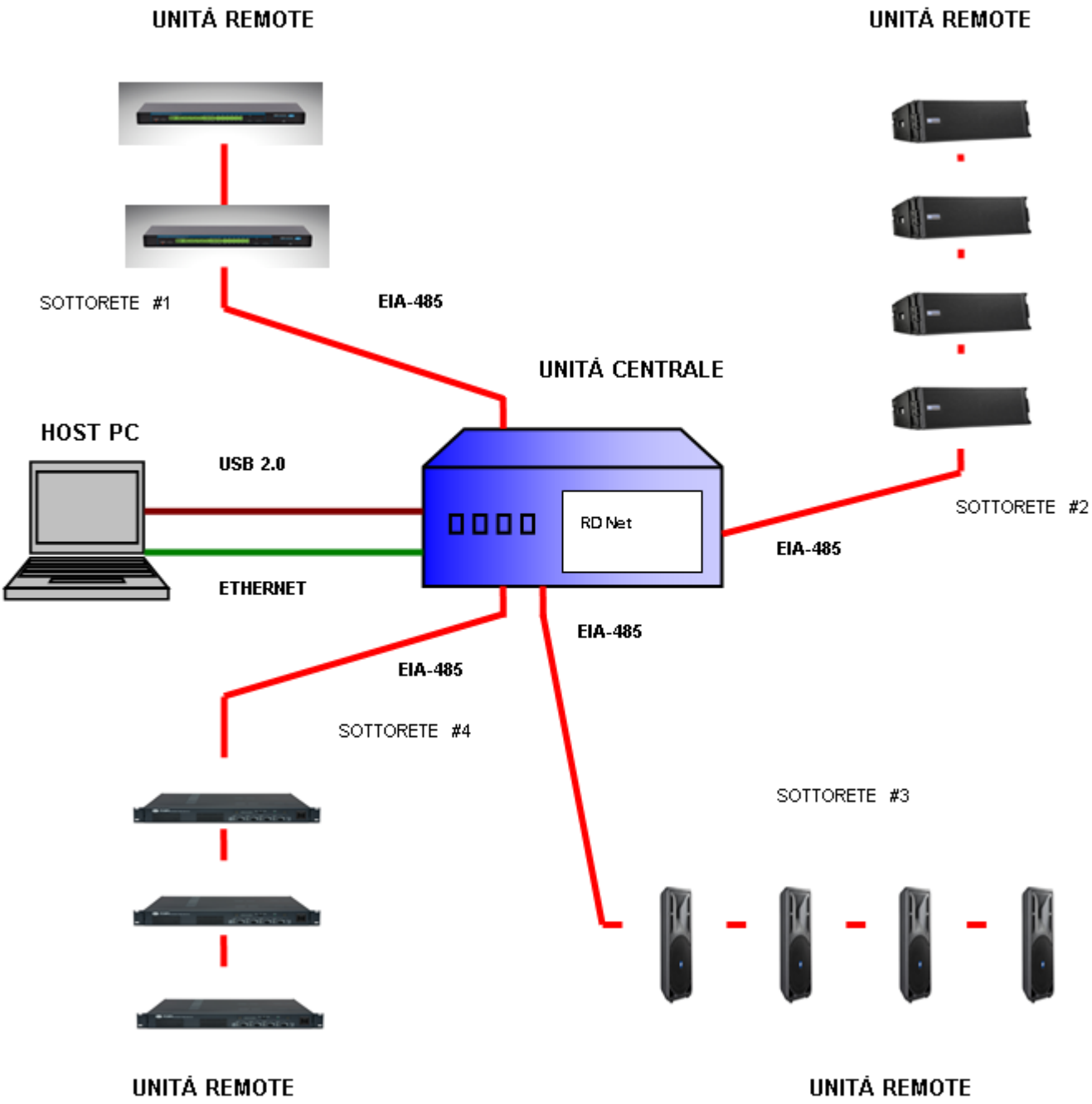

Fig. 1.2

Esempio della struttura della rete di RDNet

2. BACKGROUND

2.1 Struttura fisica dei dispositivi audio

In figura 2.1 viene illustrato un esempio di Cassa Audio (Modello TTL33A). Viene preso in considerazione questo modello in quanto la struttura di ogni cassa, a livello componentistico, è sempre simile.

Audio Cabinet

- Altoparlante
- Tubi Reflex

Amplifier housing

- Amplificatori
- Scheda input analogico
- Scheda DSP

Fig. 2.1

Cassa Modello TTL33A

Come mostrato in figura 2.1 il dispositivo è suddiviso in 2 macro componenti:

- Audio Cabinet
- Vano Amplificatore

L'Audio Cabinet contiene la struttura fisica della cassa che riproduce il suono, quindi cassa di risonanza e altoparlante.
La parte più interessante è però il vano amplificatore, dove sono posti gli amplificatori, la scheda di input del segnale analogico e la scheda DSP.
Lo schema elettrico viene di seguito mostrato evidenziando le componenti più rilevanti.

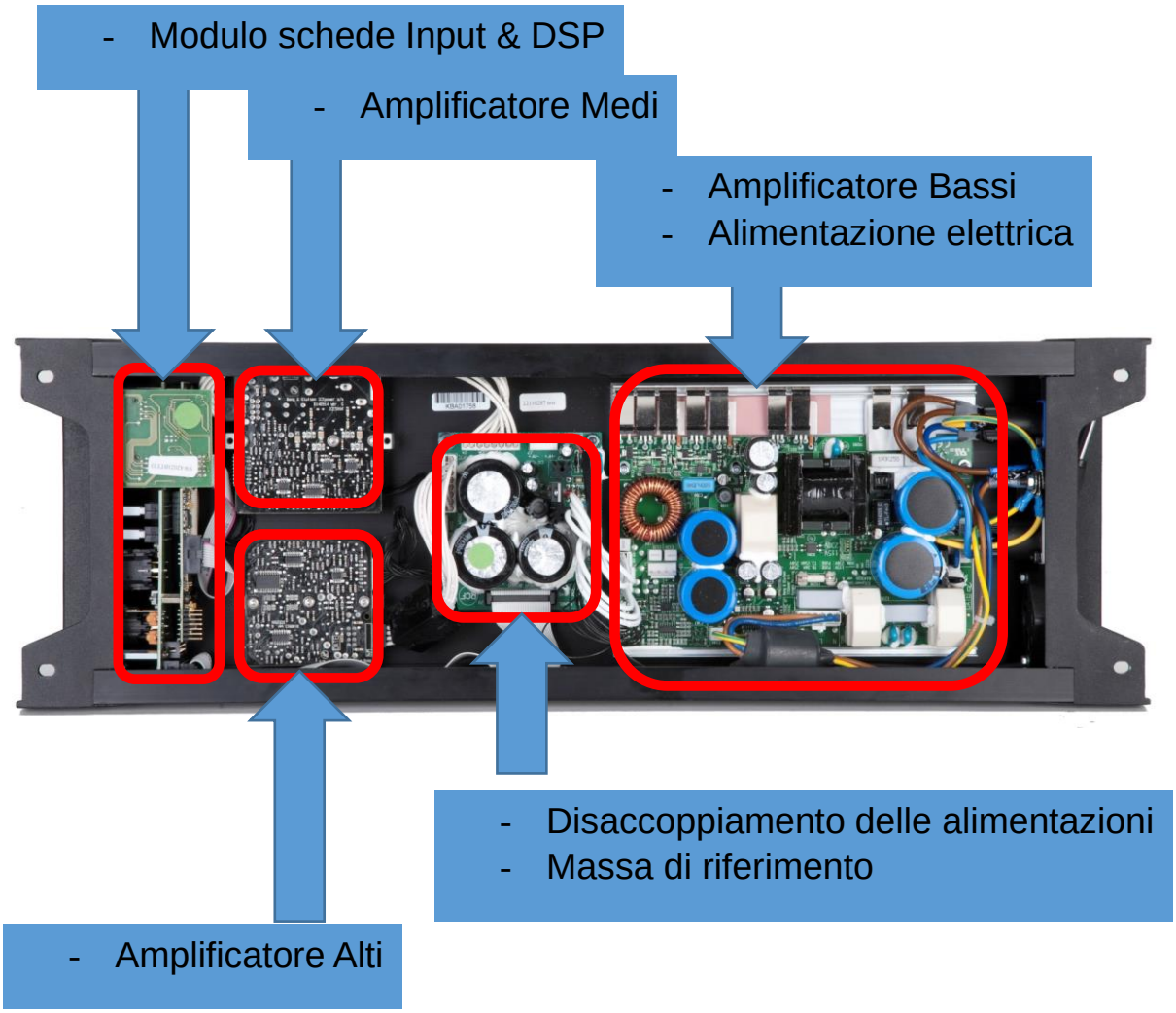

Fig. 2.2

Vano Amplificatore cassa Modello TTL33A

Scheda di Input: è la scheda che acquisisce il segnale analogico in ingresso e lo invia al DSP.

Scheda DSP: Acquisisce il segnale di ingresso analogico, lo filtra, in base ai livelli di frequenze che gli vengono imposti dal microcontrollore in alti, medi e bassi e lo converte in digitale.

Il microprocessore presente a fianco del DSP viene programmato mediante i tasti presenti sul dispositivo o in remoto da un controllore per mezzo di una porta EIA485.

I segnali digitali di uscita vengono poi spediti agli amplificatori (uno per ogni canale).

La parte di Disaccoppiamento delle alimentazioni serve per non avere cali di tensione sulla linea a fronte di una grande richiesta di energia.
Le masse del dispositivo assumono valori tra loro minimamente discostanti. Per questo motivo vengono collegate alla massa di riferimento che consente loro di azzerare queste differenze, assegnando ad ogni massa un valore standard.

2.2 Struttura fisica del controllore

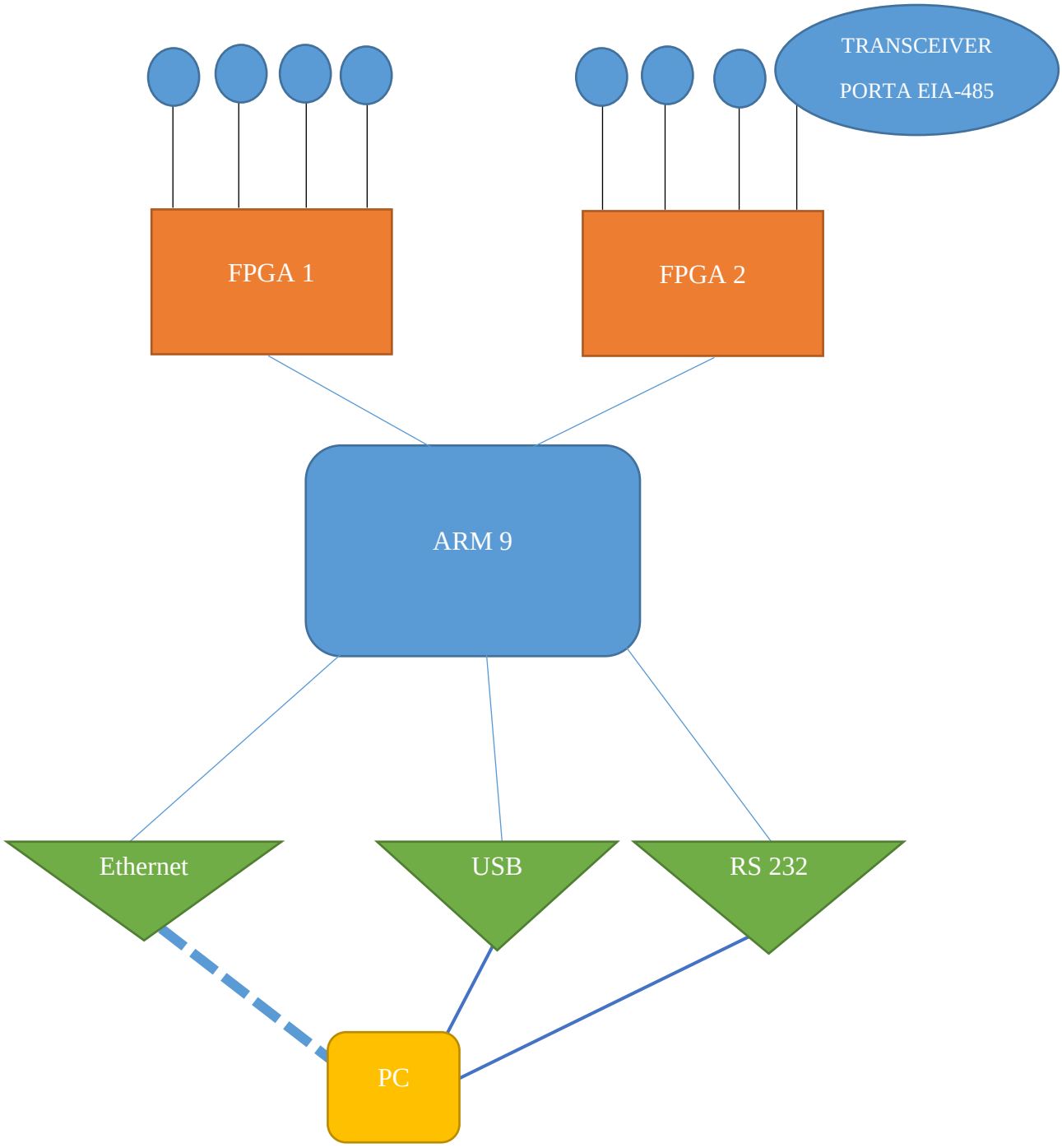

Fig. 2.3

Componenti principali presenti all'interno del controllore

In figura 2.3 è mostrata la struttura fisica semplificata del controllore, in modo da rendere più facile la comprensione del suo funzionamento.
Al centro di questa struttura troviamo un microprocessore ARM 9 [3], che mette in comunicazione il PC con i dispositivi audio, svolgendo così una funzione di routing.
Considerata la struttura dell'ARM 9, questo può usufruire di due linee di uscita (il controllore ne possiede invece otto) quindi per ovviare a questo problema vengono utilizzate due FPGA (field-programmable gate array), circuiti integrati le cui funzionalità sono programmabili via software, che consentono di gestire più ingressi/uscite mediante un unico canale condiviso.
Su ogni uscita del FPGA è posizionato un Transceiver collegato poi alla porta EIA-485, la cui funzionalità consiste nel convertire un segnale da TTL (Transistor-transistor logic) a Differenziale.
Un segnale TTL è un segnale strutturato a livelli. Questi sono dei livelli elettrici che permettono di rappresentare una variabile logica.
Un segnale differenziale, invece, è un segnale che consente di sopprimere i disturbi in quanto il dato trasmesso viene analizzato in basse alla differenza di tensione e non al livello della tensione stessa.
Il microprocessore è poi collegato alle porte Ethernet, USB e RS 232, le quali consentono di stabilire la connessione con il PC.

2.3 Protocollo di comunicazione tra dispositivi audio e controllore

Di seguito vengono illustrati i livelli dello standard ISO-OSI [4] (Standard che stabilisce l'architettura logica di una rete) per mezzo dei quali questo protocollo è stato implementato.

2.3.1 Livello Fisico

L'EIA RS-485, equivalente allo standard Europeo CCITT V11, è una specifica Modello OSI a livello fisico di una connessione seriale a due fili, half-duplex e multipoint. Lo standard specifica un sistema di gestione del segnale in forma differenziale.

La comunicazione seriale su EIA-485 tra unità master ed unità slave con i dispositivi, avviene alla velocità di 115200 bit/s.
Ciascun carattere è composto da una sequenza di bit secondo il seguente formato:

- 1 bit di start;
- 8 bit di dati;
- 0 bit di parità;
- 1 bit di stop.

Ogni carattere trasmesso è dunque composto da 10 bit, dei quali 8 di informazione e 2 di controllo. Per ridurre l'over-head della comunicazione non si usa il bit di controllo di parità.
In figura 2.4 viene mostrata la composizione dei bit di un singolo carattere.

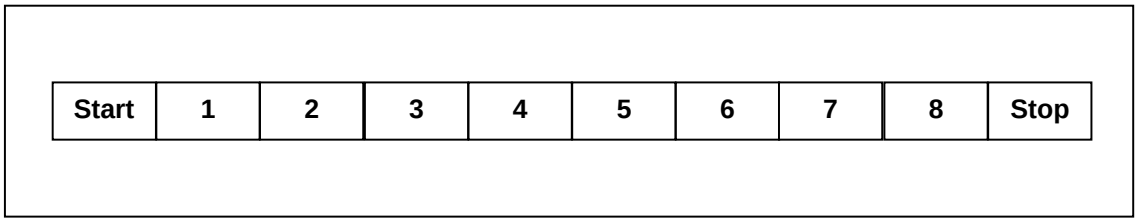

Fig. 2.4

Struttura di ogni carattere trasmesso sulla rete

2.3.2 Livello di Collegamento

Il protocollo di comunicazione è di tipo Master - Slave. Solo un

master (per volta) ed una o più unità slave sono connesse allo stesso bus seriale. Una comunicazione è sempre iniziata dal master. I nodi slave non trasmetteranno mai alcuna informazione se non su richiesta del master e non comunicheranno mai direttamente tra loro. Quindi il principio di comunicazione obbliga il master ad inviare ciclicamente una richiesta ad ogni slave ed a attendere una risposta.
Viene così definito un protocollo di comunicazione che possa permettere l'invio da master di comandi/dati sia ad un singolo slave (modalità unicast), sia a tutti gli slave (modalità broadcast), sia ad un gruppo di slave (modalità multicast) presenti su una sottorete:

- In modalità unicast, il master indirizza individualmente uno slave. Dopo aver ricevuto e processato la richiesta, lo slave ritorna un messaggio di risposta al master.
  In questa modalità ogni transizione sul bus coinvolge 2 messaggi: una richiesta dal master e una risposta dallo slave. Ogni slave deve avere un indirizzo univoco in modo tale da essere indirizzato indipendentemente dagli altri nodi.

- In modalità broadcast, il master può spedire una richiesta a tutti gli slave della sottorete. Nessuna risposta ritorna dagli slave ad una richiesta in broadcast. Il comando ricevuto viene successivamente processato da tutti gli slave.

- In modalità multi cast, il master genera un comando che ordina agli slave di mettersi in ascolto su un determinato indirizzo uguale per tutti. Di conseguenza gli slave avranno due indirizzi: uno di unicast e uno di multicast. L'indirizzo di multicast rappresenta un sottogruppo di slave. Successivamente il master, ovvero il controllore, inoltra il comando reale all'indirizzo di multicast del gruppo. I singoli slave indicano l'avvenuta elaborazione del comando attraverso un flag del proprio status. Questo flag è monitorato dal master per definire la chiusura dell'operazione di multi cast, ovvero ogni slave

spedisce una risposta di effettuata operazione al controllore. Quest'ultimo rimane in attesa fino alla ricezione di tutte le risposte oppure fino a quando il relativo timer di risposta non termina.

2.3.3 Macchina a stati del Master

Il comportamento del livello di collegamento è definito, all'interno del firmware del master e degli slave, da macchine a stati che controllano e gestiscono il traffico sulla rete.
La macchina a stati illustrata in figura 2.5 descrive il comportamento del master durante una comunicazione con l'unità remota slave:

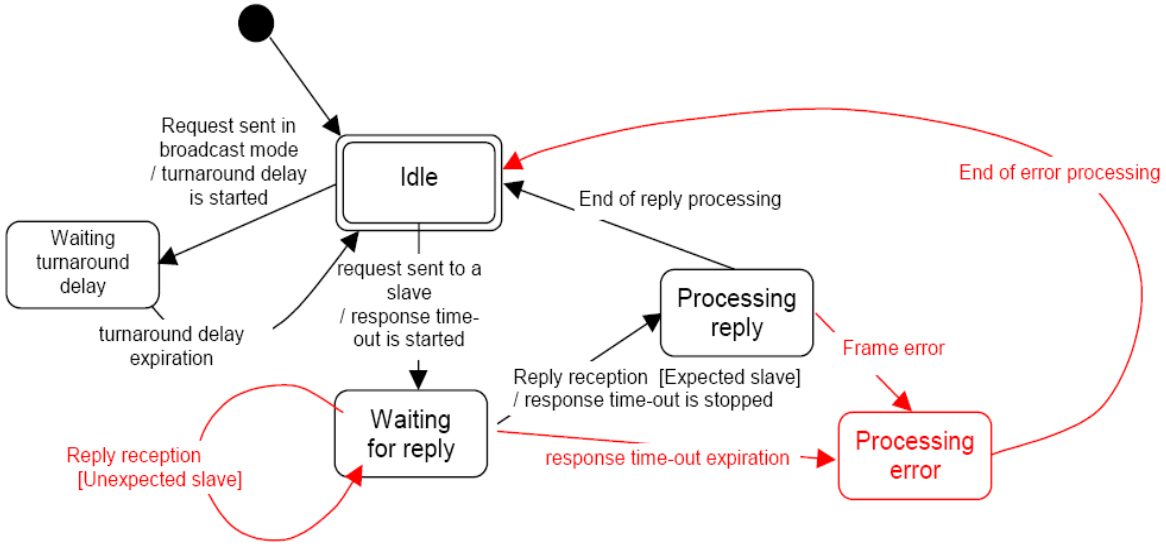

Fig. 2.5

Macchina a stati del Master

Lo stato di "idle" equivale a nessuna richiesta pendente. Questo è lo stato iniziale dopo l'accensione dell'unità master. Una richiesta può essere spedita solo a partire dallo stato di "idle". Dopo l'invio, il master lascia lo stato di "idle" e non può inviarne altre finché non torna nello stato iniziale.

Quando la richiesta inviata è di tipo unicast, il master passa nello stato “waiting for reply” ed un “Response Time-out” è avviato. Questo timer serve a evitare che il master attenda per un tempo indefinito una risposta dallo slave. Il tempo di Response Time-out è dipendente dall’applicazione in corso.
Quando il master riceve una risposta, come prima cosa decodifica e verifica il frame ricevuto. Vi sono diverse casistiche di errore:

1) Nell’eventualità di una risposta proveniente da uno slave non previsto, il timer di “Response Time-out” viene mantenuto attivo.
2) In caso si rilevi un errore nel frame, può essere eventualmente effettuata una nuova comunicazione con lo slave.
3) Se non vi è alcuna risposta, il periodo previsto di attesa si esaurisce ed è generato un errore.

Una volta gestito l’errore, il master torna nello stato di “idle” abilitando una nuova richiesta.
Se invece l’operazione è andata a buon fine, viene processata la risposta ottenuta ed il sistema viene riportato nello stato di “idle”.
Quando una richiesta da parte del master è inviata in broadcast, nessuna risposta è attesa. Ciò nonostante è previsto un tempo di ritardo prima di intraprendere una nuova comunicazione per permettere alle unità slave di terminare l’azione. Questo ritardo è chiamato “Turnaround delay”. Quindi il master, prima di tornare nello stato di “idle” ed inviare una nuova richiesta, attende nello stato “Waiting Turnaround delay”.
Se la richiesta inviata è di tipo Multicast, in seguito all’invio del pacchetto, il master si porta nello stato “Wait turnaround delay”, come nel caso di una comunicazione di tipo broadcast.
In modalità unicast il tempo di Response Timeout deve essere fissato sufficientemente lungo perché uno slave possa processare il comando ed inviare la risposta. In modalità broadcast il tempo di Turnaround delay deve essere lungo abbastanza affinchè ogni slave processi il comando e si prepari alla ricezione di quello successivo.

Quindi, il Turnaround delay dovrebbe essere inferiore al Response time-out.
Casi particolari: l'interrogazione di stato richiede una risposta estremamente rapida al fine di non gravare la rete di tempi di attesa. Ogni risposta deve partire, dopo 100 microsecondi dalla richiesta.

La richiesta di stato è suscettibile del timeout come ogni altro comando, ma lo slave che, con continuità, non risponde per un tempo pari a 10s verrà escluso dal ciclo di polling.

2.3.4 Macchina a stati dello Slave

La macchina a stati mostrata in figura 2.6 descrive il comportamento dello slave durante una comunicazione con l'unità centrale master:

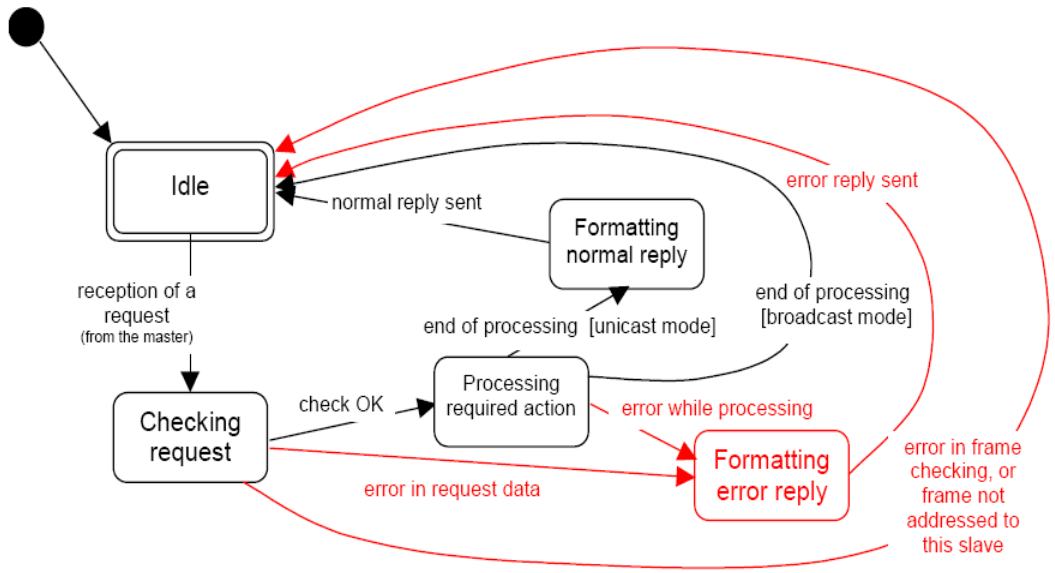

Fig. 2.6

Macchina a stati dello Slave

Lo stato di “idle” equivale a nessuna richiesta pendente. Questo è lo stato iniziale dopo l’accensione dell’unità slave.
Quando una richiesta è ricevuta, lo slave controlla il frame inviato prima di intraprendere l’azione richiesta. Una volta che il frame viene decodificato e validato, viene verificato se il comando è indirizzato alla stesso slave o se è rivolto ad altre unità remote della sottorete.
In caso affermativo, viene intrapresa, da parte dello slave, l’azione specificata all’interno del comando. Segue quindi un messaggio di risposta che può essere di tipo “normal”, se l’azione è eseguita correttamente, o di tipo “error” se l’azione ha avuto esito negativo. Se invece il messaggio non è destinato allo slave in questione, nessuna risposta è inviata al master.
Differenti eventi di errore si possono verificare durante una comunicazione tra master e slave: una formattazione del pacchetto errata, un’ azione specificata non valida per lo slave, etc. In seguito alla ricezione di un pacchetto Multicast lo slave seguirà lo stesso percorso di elaborazione previsto per le richieste ricevute dal master in modalità Broadcast.

2.3.5 Livello Applicazione

Per poter distinguere logicamente gli slave tra loro, è necessario assegnare un indirizzo univoco a ciascuno nodo collegato sulla linea. Tale assegnazione viene fatta in modo dinamico in fase di start-up della sottorete.
In modo sequenziale, a partire dall’ unità remota fisicamente più vicina al master, viene assegnato un indirizzo per ogni slave. Così facendo il master possiede la disposizione delle unità remote, conoscendone la posizione relativa.
Ogni unità slave ha integrato un relé, ovvero un deviatore che viene azionato da un elettromagnete. All’accensione del sistema, le unità slave hanno il relé in posizione aperta (essendo tutti i dispositivi

collegati in daisy chain, il relè è da considerarsi un pulsante che quando è aperto scollega i dispositivi successivi presenti sulla linea di collegamento). In questa configurazione le unità slave non propagano il segnale dal connettore di comunicazione di ingresso a quello di comunicazione di uscita e solo la prima unità slave è connessa alla linea e correttamente collegata.

3. PROTOCOLLO DI COMUNICAZIONE TRA CONTROLLORE E PC

3.1 Panoramica Generale

La comunicazione tra il controllore ed il PC è implementata a lato controllore mediante il firmware presente sul microprocessore (ARM9) e lato PC con un Software dedicato.
In quanto mi sono dedicato allo sviluppo della sola parte PC, non tratterò in questo documento del firmware del controllore. Si consideri in ogni caso il controllore, per quanto riguarda la comunicazione con il PC, come un mero instradatore di comandi.
Descriverò successivamente il livello applicativo, dello standard ISO-OSI, a cui il programma stesso appartiene ed il livello di trasporto, in quanto parte fondamentale utilizzata dal software. Gli altri livelli dello standard non sono argomento di questa tesi.
La connessione tra PC e controllore può essere di tre tipi: USB, Ethernet ed RS232.
La RS232 è un'interfaccia seriale a bassa velocità di trasmissione per lo scambio di dati tra dispositivi digitali.
La USB è anch’essa un collegamento seriale, ma è stata progettata per consentire a più periferiche di essere connesse mediante l'utilizzo di una sola interfaccia standardizzata e di un solo tipo di connettore.
Ethernet è una famiglia di tecnologie necessarie alla realizzazione e al funzionamento di reti locali le cui specifiche tecniche sono state stabilite con lo standard IEEE 802.3[5].

Mi soffermerò a spiegare unicamente la modalità di connessione via Ethernet che personalmente ho sviluppato e seguito in quanto le altre erano già esistenti.
Il software di RDNet è stato sviluppato unicamente per sistema operativo Windows, utilizzando l'ambiente di sviluppo Microsoft Visual Studio e presenta una interfaccia grafica intuitiva che agevola l'utente nell'utilizzo.

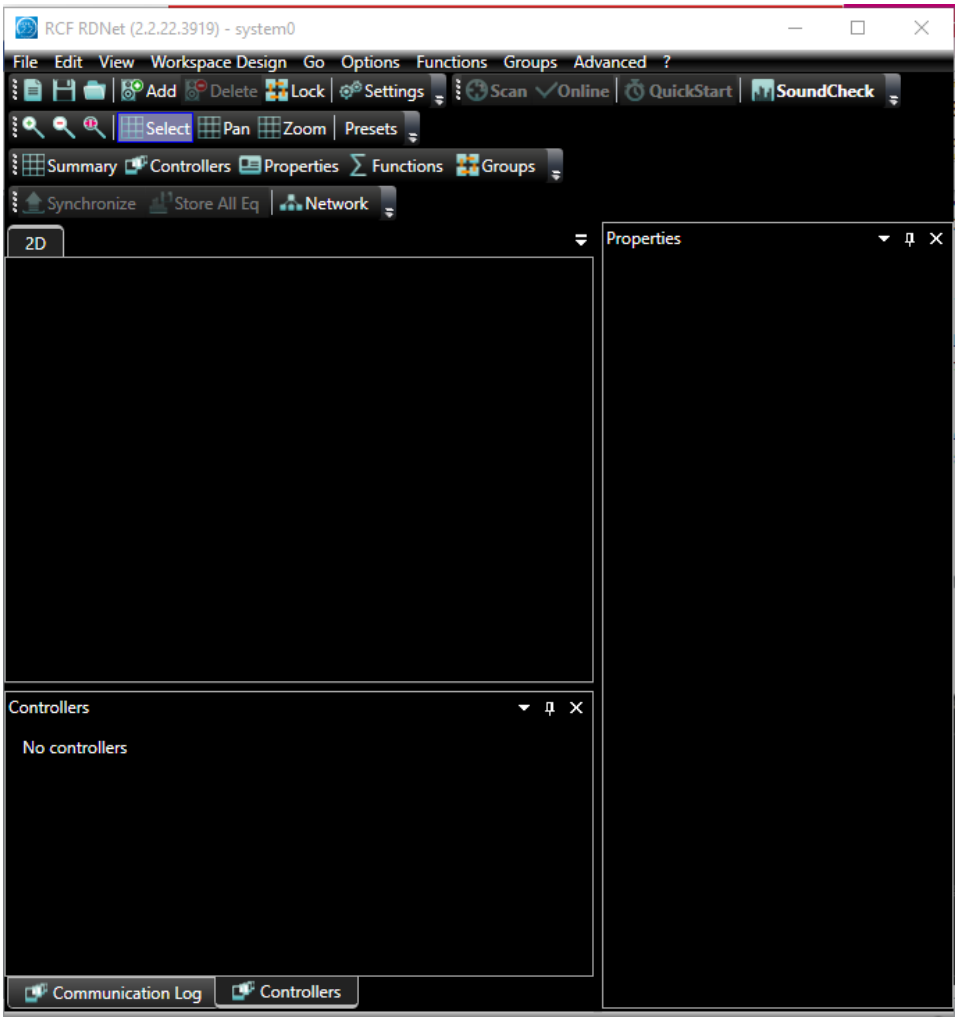

Fig. 3.1

Interfaccia grafica PC RDNet

L'applicativo è scritto in linguaggio C# seguendo la struttura e le regole del pattern di programmazione Model-View-View Model [6], nel quale:

Il Model rappresenta il punto di accesso ai dati;
La View rappresenta la vista dell'applicazione, l'interfaccia grafica che mostrerà i dati;
Il ViewModel è il punto di incontro tra la View e il Model: i dati ricevuti dal Model sono elaborati per essere presentati e passati alla View.

3.2 Analisi del codice preesistente

Una parte consistente del mio lavoro è stata l'apprendimento del programma sia lato utente, ovvero quali funzionalità esso consente di svolgere e in quale ambiente, sia lato sviluppatore, quindi la struttura del programma a livello di programmazione.
Mi è stato inizialmente spiegato quali erano le funzionalità di RDNet, eseguendo test con dispositivi e controllori reali.
Superata questa breve introduzione ho iniziato ad osservare ed analizzare il codice. Questo si è rivelato un duro impatto, considerando che il codice è composto da circa 80.000 righe. Ho quindi provato ad analizzare i diagrammi delle classi prodotti utilizzando una funzionalità di Visual Studio che automatizza un analisi del codice. Questo passaggio mi ha permesso di capire com'era strutturato il software a livello generale, ovvero da quali sezioni era composto senza dover eseguire un'analisi file per file.
Per comprendere meglio quello che stavo analizzando ho successivamente deciso di studiare le regole base di WPF [7] (Windows Presentation Foundation), essendo questa tecnologia a me sconosciuta. Fino ad ora avevo sempre utilizzato Windows Form [8]. Queste sono due librerie di classi che Microsoft mette a disposizione del programmatore e che identificano due diversi metodi di sviluppo. WPF è una modalità di programmazione innovativa, che, rispetto a Windows Form, consente di separare i ruoli di designer da quello di sviluppatore.

Il mio percorso di apprendimento è avanzato notevolmente nel momento in cui uno degli sviluppatori di RDNet si è preso l'incarico di spiegarmi a fondo la logica di base del software, consentendomi così di iniziare ad orientarmi all'interno del codice. Con lui ho successivamente iniziato un percorso istruttivo di analisi step by step del sorgente del software. Ci sono volute circa due settimane prima che iniziassi a comporre qualche riga di codice e riuscissi quindi a sviluppare qualcosa di concreto.
La quantità di codice prodotta per lo sviluppo della comunicazione mediante cavo ethernet si aggira intorno alle 800 righe (tra modifiche apportate e parti completamente nuove).
Le difficoltà ma11ggiori le ho riscontrate nell'imparare tutta la struttura del software ed il Background di tutto il sistema RDNet e nell'analizzare le possibili problematiche che potevano generarsi nel tempo, affinché il software raggiungesse un discreto livello di stabilità.

3.3 Livello Applicazione

Prima di iniziare ad utilizzare l'applicativo, è consigliabile costruire la rete connettendo i dispositivi al controllore e il controllore al PC.
I dispositivi devono essere connessi tra di loro in Daisy Chain e successivamente al controllore deve essere collegato il punto iniziale della catena. Per stabilire la connessione è necessario utilizzare un cavo EIA-485.

Una volta che la rete è attivata, si può avviare l'applicativo (in realtà si può costruire la rete anche in un secondo momento, cioè ad applicativo già avviato, tuttavia è consigliato strutturarla prima).
Per stabilire la connessione diretta con i controllori è necessario selezionare nel menù la voce "Go">>"Network Connection", in cui viene mostrata la finestra nella quale vengono elencati tutti i controllori presenti all'interno della stessa rete del software in

esecuzione. Nel sviluppare questa parte è stato considerato il fatto che ogni PC possa usufruire di diverse schede di rete ed infatti è presente all'interno del software un'interfaccia grafica nella quale selezionare la scheda di rete da utilizzare, ovvero quella in cui cercare i controllori da connettere. Questa finestra la si trova selezionando nel meno la voce "Option">>"Network Interfaces" ed è illustrata nella figura 3.2.

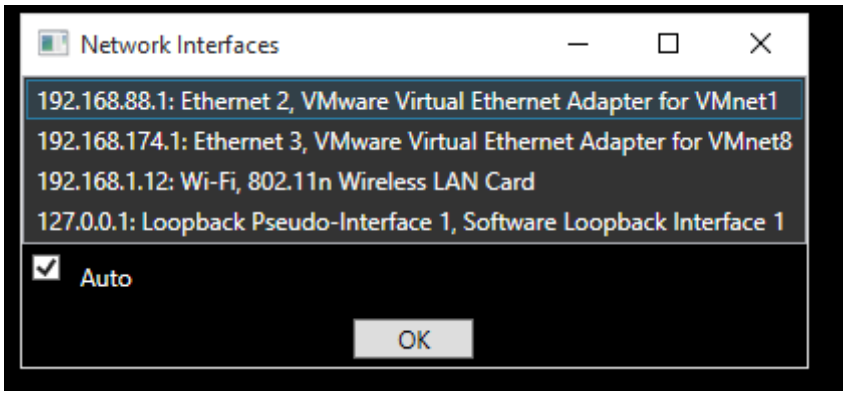

Fig. 3.2

Finestra Network Interfaces del software PC

In questa finestra, selezionando la scheda di rete desiderata, questa viene impostata come predefinita per tutto il software, viceversa spuntando "Auto" viene selezionata una tra quelle elencate in base ad una coda di priorità interna.
La scheda "Network Connection", rappresentata in figura 3.3, presenta i seguenti componenti:

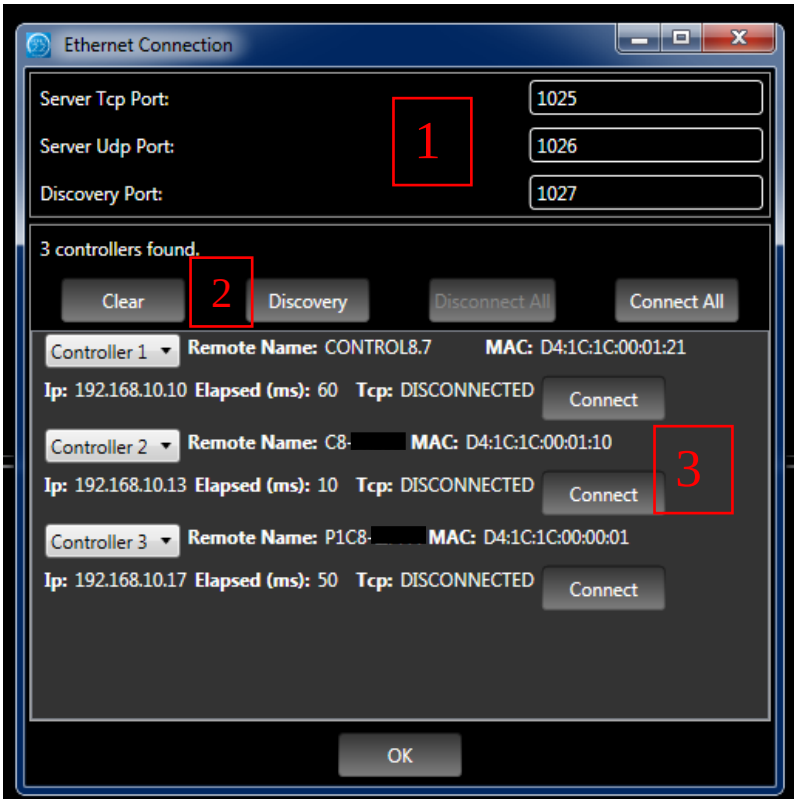

Fig. 3.3

Finestra Network Connection del software

1) Identificazione delle porte relative ai diversi servizi. Quelle indicate sono quelle di default ma attraverso un'apposita finestra possono essere cambiate.
2) Il pulsante Discovery genera una richiesta UDP [9] in broadcast chiedendo a tutti i controllori di identificarsi. Questi non appena ricevono la richiesta rispondono inviando il proprio nome e il proprio MAC-Address.
3) Cliccando il pulsante "Connect", viene eseguito il match tra il controllore a video, identificato dal menu a tendina, e quello reale al quale ci si sta collegando. Viene quindi lanciata una richiesta TCP [10] indirizzata ad un socket del controllore su una porta prestabilita. Questo, nel caso non sia ancora connesso con nessun PC, accetta la richiesta, altrimenti la rifiuta ed il software restituisce un messaggio di errore. Una volta che la

connessione è stata stabilita, i dati del controllore della lista reale vengono copiati in quello della lista virtuale.

Quando anche la connessione è stata stabilita, si può procedere a scansionare i dispositivi di ogni controllore, dove per ogni linea di collegamento è richiesto il numero di slave presenti.

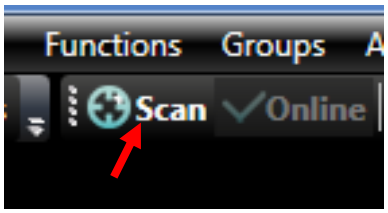

Fig. 3.4

Zoom sui pulsanti Scan e Online del menù di RDNet

Il pulsante “Scan” avvia la macchina a stati in figura 3.4 (In giallo è evidenziato il tragitto percorso utilizzando il cavo ethernet):

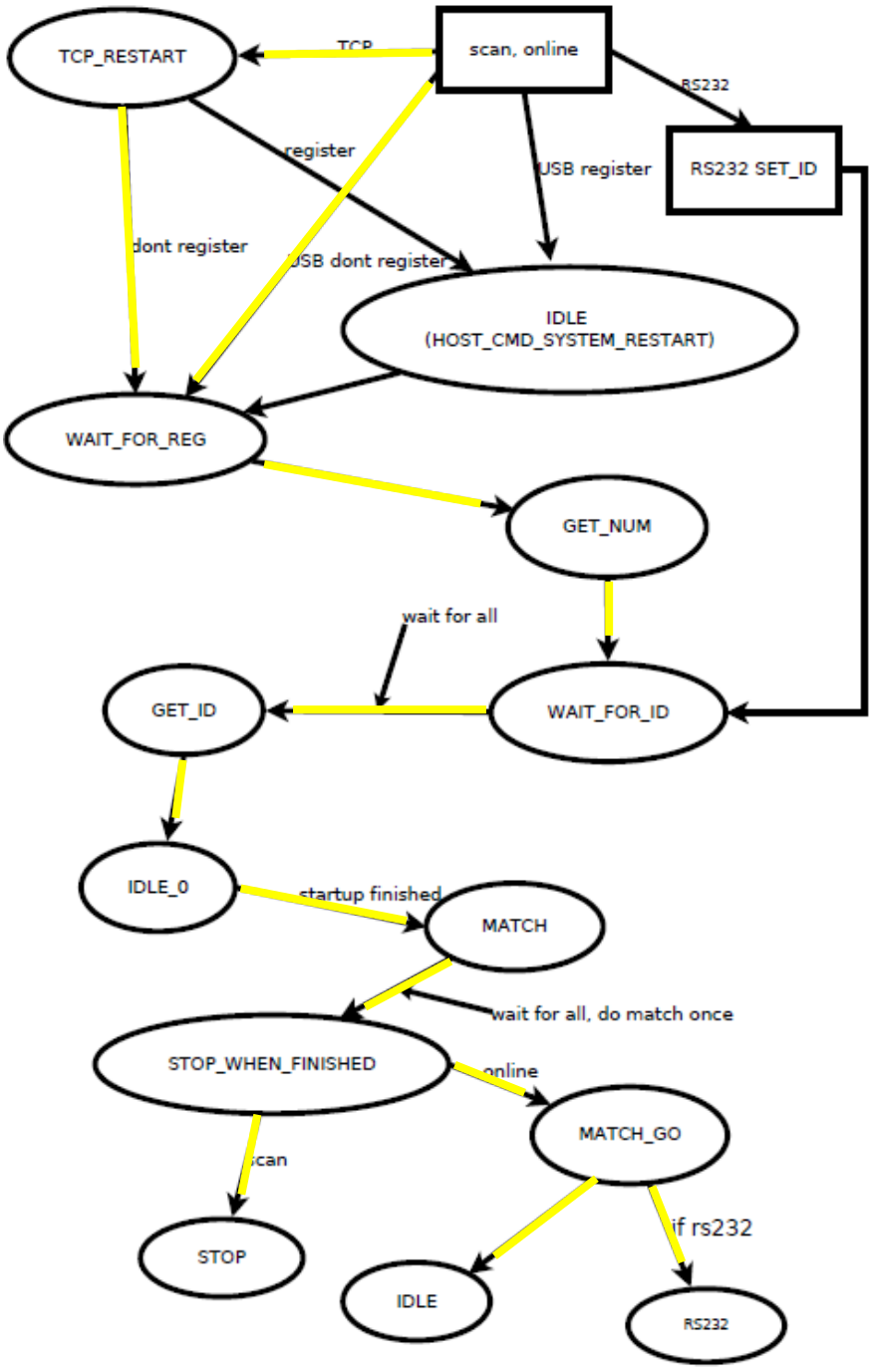

Fig. 3.5

Macchina a stati del software PC

Questo schema viene eseguito contemporaneamente per tutti i controllori ma nella spiegazione ne considererò uno solo.

1) Tramite TCP viene spedita al controllore la richiesta di registrazione di tutte le linee di slave (la registrazione è locale al controllore, il PC non si aspetta nessun dato di ritorno ma solo la terminazione del comando). Questa richiesta è bloccante per il software, per cui l'applicativo rimane in attesa dell' ack di terminazione prima di proseguire con l'esecuzione di altri comandi.
2) Una volta che il controllore ha effettuato la registrazione ed ha informato il PC dell'avvenuta esecuzione del suo compito, il PC domanda ai controllori quanti slave possiedono su ogni linea. Il controllore, che con il passo 1) ha generato una lista di dispositivi, la invia al PC. Anche questa è una richiesta bloccante.
3) Quando il pc ha ottenuto il numero dei dispositivi per ogni linea del controllore, richiede l'identità di ognuno di questi. L'identità consiste in:
    a. ID del dispositivo. Questo è composto da 3 numeri: il numero del controllore, la linea su cui il dispositivo è collegato (da 1 a 8) e la posizione dello slave sulla linea (es. 1.2.1).
    b. CRC [11] (controllo a ridondanza ciclica), per verificare l'allineamento dell'equalizzatore.
    c. Filtri (funzioni di taglio applicate al segnale che eliminano alcune frequenze), ID Logici (se configurati, rappresentano un metodo di identificare i dispositivi che non considera la posizione all'interno della rete, ma la posizione logica memorizzata nel software).
4) Terminata anche questa operazione il software PC possiede tutti i dati necessari per disegnare la scena a video.
5) Prima che questo venga fatto, però, il programma esegue il match dei dispositivi reali con quelli virtuali (nei paragrafi successivi verrà spiegato quali sono le differenze tra dispositvi reali e virtuali). Il match consiste in una verifica della struttura di oggetti simili (relativamente al modello e alla configurazione

del dispositivo) scorrendo le due liste, dispositivi reali e dispositivi virtuali, elemento per elemento. Quando vengono trovate differenze, queste vengono identificate e mostrate all'utente. (Es. dispositivo non trovato all'interno della lista virtuale, in questo caso appare un elenco con i dispositivi reali in più rispetto a quelli virtuali e viene chiesto all'utente se li vuole connettere al software).

6) Fino ad ora è stato descritto il funzionamento del solo tasto "scan"; se invece viene premuto "online", il processo viene completato con un ulteriore passo denominato match & go. Oltre a eseguire il match dello step 5, vengono copiate tutte le differenze, rilevate nella lista virtuale, all'interno dispositivi reali rendendole effettive.
7) Terminata questa fase viene finalmente disegnato il sinottico a video.

Quando il sinottico è completo ed il programma è operativo, nella coda dei comandi vengono inserite le richieste di status via UDP per ogni linea di dispositivi.
La richiesta dello stato degli slave viene inviata ciclicamente, in modo da poter avere un sinottico sempre aggiornato in tempo reale. Per stato si intende il valore di tutti i parametri che si possono variare all'interno di un dispositivo, quindi per esempio volume meters (i volumi dei diversi canali), muting (cassa muta oppure no), eventuali tasti premuti sull'interfaccia presente sul dispositivo, etc…
A questo punto si possono utilizzare tutte le funzionalità di RDNet per il monitoring ed editing delle configurazioni in tempo reale del dispositivo, ovvero:
-Eseguire la scansione di tutti i dispositivi connessi alla rete per mezzo dei controllori.
-Settare il livello di raffreddamento delle ventole, creare e gestire cluster di dispositivo.

-Settare il livello di ogni canale ed il delay su un dispositivo.
-Settare le funzioni di equalizzazione per ogni singolo dispositivo per cluster.
-Aggiungere/modificare filtri relativi a dispositivi presi singolarmente o in cluster.
-Controllare la frequenza e il guadagno dei dispositivi.
-Salvare parametri e settaggio in precedenza assegnati ai dispositivi.

Alcune di queste funzionalità possono ovviamente essere utilizzate anche su un sinottico virtuale, ma per renderle effettive è sempre necessario avere la struttura connessa al software del PC.

3.4 Livello di Trasporto

Come già scritto in precedenza, la comunicazione tra controllore e PC può essere stabilita attraverso USB, Ethernet oppure RS232.
In tutti i casi ogni funzione o comando ad alto livello tra l'unità Host PC e il controllore avviene tramite l'invio e la ricezione di una precisa coppia di PDU (Protocol Data Unit, è l'unità d'informazione o pacchetto scambiata tra due entità).
Questo scambio di dati non viene però effettuato nel momento in cui il pulsante a video, relativo alla funzione desiderata, viene premuto, ma solo quando il thread dei comandi va a pescare dalla coda dei comandi l'istruzione da eseguire. Infatti qualsiasi nuova impostazione selezionata comporta l'inserimento del comando in fondo alla coda.
Per ogni funzione generata dall'unità Host PC, una PDU viene trasmessa all'unità centrale (in una connessione con cavo ethernet il PDU viene inviato con protocollo TCP). Riconosciuto il suo indirizzo ID, questa attiva il task che genera l'esecuzione della routine di gestione di tale comando.

La funzione può interessare semplicemente l'unità centrale o può essere rivolta ad un' unità remota di una delle sottoreti attive.
Nel primo caso, il controllore risponde in modo appropriato con una PDU che rappresenta un comando di acknowledge, il quale può contenere dei dati da inviare all'Host PC, o un comando di no-acknowledge a cui è associato un codice di errore. Tali comandi ad alto livello sono utilizzati per controllare o analizzare unicamente lo stato di funzionamento dell'unità centrale.
Nel secondo caso, il comando ad alto livello non è rivolto all'unità centrale ma rientra nella classe di funzioni di controllo delle unità remote. In tale situazione l'unità centrale agisce da bridge e comunica la richiesta dell'Host PC all'unità remota.
La PDU rivolta all'unità remota è incapsulata all'interno del campo <DATA> della PDU generata dall'Host PC. In questo modo l'unità centrale, ogni qual volta riconosce il comando di bridge <CMD>=CMD_BRIDGE, deve semplicemente estrarre dal campo <DATA>=DATA_BRIDGE, la PDU da inviare all'unità remota.
In particolare, l'unità centrale deve estrarre dal campo <DATA> della PDU inviata dall'Host PC, il numero della sottorete SUBNET_ID, l'indirizzo dell'unità remota SLAVE_ID, la lunghezza LENGTH_DATA_SLAVE dei dati successivi ed il comando da eseguire nell'unità remota CMD_SLAVE. La parte rimanente del campo <DATA> rappresenta l'insieme di dati DATA_SLAVE associato alla PDU da inviare all'unità remota.
Quest'ultima risponde in modo appropriato con una PDU che rappresenta un comando di acknowledge, il quale può contenere dei dati da inviare all'unità Host PC, o un comando di no-acknowledge a cui è associato un codice di errore. La PDU di risposta dell'unità remota viene a sua volta incapsulata nella PDU di risposta che viene inviata all'Host PC da parte del controllore.
In questa modalità di funzionamento dell'unità centrale da bridge, la risposta all'Host PC è inviata solamente quando è tornata una PDU correttamente formattata dall'unità remota. Non è previsto un

controllo di timeout della risposta da parte dell'unità centrale in quanto tale controllo è gestito dall'Host PC.

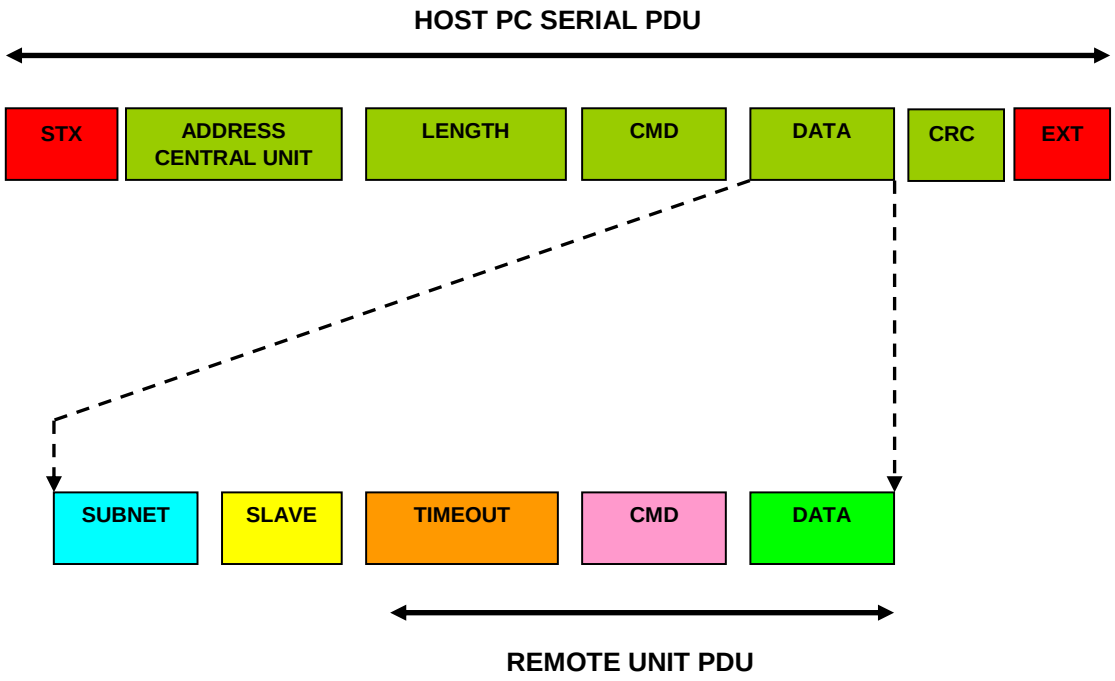

Fig. 3.6

Struttura di un PDU in RDNet

La lista delle funzioni esprimibili all'interno di un PDU è riassunta nella seguente tabella. Se il PDU contiene un istruzione non riconosciuta, RDNet lo scarta.

Comando	ID	Descrizione
Status_Request	1	Richiesta di stato della sottorete
Registration_Status	5	Verifica stato della registrazione
Firmware_Check	7	Richiesta versione Firmware del Router

System_Restart	8	Riavvia il router
UDP_stream_Set	9	Attiva lo stream UDP
TCP_Alive	10	Keep ALive del TCP
Set_SYS_Params	11	Aggiorna parametri sistema Router
Get_SYS_Params	12	Richiede parametri sistema al Router
Send_Multicast	13	Invia il pacchetto in multicast
Read_Multicast	14	Legge maschera di ACK del multicast
Router_Identify	15	Identifica Router con blinking dei LED
Set_router_name	16	Imposta il nome del router
Get_router_name	17	Ottiene il nome dal router
Safe_Bridge	256	Invia il pacchetto in unicast (Modalità SAFE)

L'invio di un PDU per mezzo di una connessione Ethernet è da considerarsi come un invio seriale di informazioni, ovvero una tubatura all'interno della quale vengo inseriti i dati da una delle due estremità (mittente) ed estratti dall'altra (destinatario). La dimensione di un PDU non è quantificabile, questo perché il valore dei dati al suo interno può variare e di conseguenza il ricevente non conosce mai la quantità esatta di bytes da estrarre per comporre un pacchetto.
Questo problema viene ovviato mediante l'utilizzo di un algoritmo apposito di Stuffing che consente di delimitare i confini del pacchetto prima che questo venga spedito sulla rete.
Come illustrato nella figura 3.6, il PDU è racchiuso in una sezione composta da STX in testa e ETX in coda. Questa dicitura indica una sequenza di byte che appunto delimita il pacchetto.

Esistono diverse tipologie di algoritmi di Stuffing, quello utilizzato in RDNet appartiene alla famiglia dei Byte Stuffing [12], ovvero quelli che analizzano l'informazione un byte alla volta.
Questo algoritmo, oltre ad inserire una sequenza di bytes all'inizio ed alla fine del PDU, sostituisce le occorrenze, all'interno dell'informazione, di questa sequenza semplicemente eseguendo una duplicazione dei bytes. In questo modo nel momento in cui si andrà ad analizzare un pacchetto, eseguendo un'operazione di sostituzione delle coppie di sequenze di byte con una unica sequenza, si otterrà il PDU originale.

3.5 Dispositivi Reali e Dispositivi Virtuali

Un argomento importante, legato al livello di applicazione, per la comprensione della logica del software è la differenza tra dispositivi reali e dispositivi virtuali.
Consideriamo di avere già installato l'impianto audio, quindi di avere tutti i controllori e tutti i dispositivi connessi alla rete, ma di non averli ancora connessi al software PC.
Pensiamo ora ad un utente in possesso di un solo PC sul quale è stato lanciato il Software RDNet e con il quale il giorno successivo egli dovrà effettuare dei test sull'impianto, ma per il momento il PC è scollegato dalla rete del controllore e dei dispositivi.
A questo punto un utente può costruire a video l'ipotetico schema di come dovrebbe essere l'impianto completo. Attraverso la voce di menù "Add", infatti, può aggiungere un qualunque tipo di dispositivo ed associarlo ad un controllore. Questa operazione può essere ripetuta un numero arbitrario di volte, generando così a video l'impianto completo come se si fosse premuto il pulsante di scansione dopo aver connesso i controllori.
Un utente non può aggiungere di sua volontà dei controllori, può solo aggiungere dispositivi associati ad un controllore identificato con un ID. Se l'ID del controllore non è già esistente questo viene generato e

viene visualizzato nell'apposita lista, ma privo di ogni informazione ad eccezione dell'ID.
Quanto menzionato in precedenza (controllori e dispositivi) sono componenti virtuali del programma. Esiste dunque una lista che immagazzina tutto ciò ed è disponibile a livello globale all'interno del software.

Per quanto riguarda invece i dispositivi reali, questi vengono registrati solo nel momento in cui il controller che li gestisce viene identificato e connesso al software. Successivamente viene eseguito la scansione di riconoscimento che li confronta con quelli virtuali (se esistenti). Quelli che compaiono in entrambe le liste sono inseriti nell'apposita lista di dispositivi reali. L'utente viene poi interrogato per decidere che cosa fare di quelli discordanti.

Per ogni dispositivo virtuale si possono eseguire tutte le funzionalità desiderate, essendo il suo funzionamento molto simile a quello di uno reale. Una volta collegato l'impianto e fatto il match dei dispositivi la configurazione impostata sui dispositivi virtuali può essere copiata su quelli reali premendo il pulsante "online".

3.6 Fase di Testing

Con la conclusione di questa nuova funzionalità, il software ha acquisito una maggiore complessità, soprattutto nella parte di comunicazione con i diversi controllori.
Un aumento della complessità porta inevitabilmente con se un aumento delle situazioni critiche ed errori all'interno del programma. Per correggere questa situazione sono stati necessari diversi giorni di testing, in cui ogni nuova funzione del programma, la cui esecuzione

andava a modificare apparati già esistenti, veniva sottoposta a controlli sul suo output.
Questo fase è stata svolta mediante l'utilizzo di una funzionalità di Visual Studio: creazione di Unit Test.
Le Unit Test rappresentano una verifica del funzionamento isolato delle parti software che sono testabili separatamente.
L'utente può quindi generare una porzione di codice, specificando input e output atteso, che va a verificare una seconda porzione di codice, da lui precedentemente scritta, la quale, una volta eseguita, mostra le discordanze tra il risultato atteso e quello ottenuto.
Nel momento in cui è stato verificato il corretto andamento delle nuove funzioni, è stata eseguita una pulizia del codice in eccesso, ossia eliminati quei frammenti di codice scritti e poi mai utilizzati oppure abbandonati perché rimpiazzati con altri.
Questo, che apparentemente può sembrare uno step molto semplice, può invece causare problemi a lungo e a breve termine. Risulta, infatti, non poi così difficile cancellare una parte corretta del codice, per errore, rendendo così l'applicativo non funzionante.
Fortunatamente nello sviluppo sono stati frequentemente inseriti commenti all'interno del codice ed i frammenti da eliminare sono stati segnalati.
L'utilizzo dei commenti all'interno del codice ne ha agevolata la lettura e la comprensione, velocizzando il processo di cancellazione.
Corretti tutti gli errori principali, il software è stato consegnato a dei tester che lo proveranno, funzionalità per funzionalità, nei prossimi mesi. Questo è un importante passaggio, perché spesso il progettista non riesce a considerare tutte le diverse sfaccettature di un programma e nemmeno la corretta ergonomia che si manifesta solo nel momento in cui un software lo si usa abitualmente.
La figura del tester possiede inoltre una importante responsabilità, ovvero costui è l'ultima persona all'interno dell'azienda che analizza il software prima della pubblicazione sul sito web aziendale.
Questo infatti si tratta di un software gratuito al quale vi si può accedere solo previo registrazione sul sito web dell'azienda.

3.7 Prossimi Sviluppi

Con l’implementazione della comunicazione tra controllore e PC si sono aperte molte strade di ampliamento e potenziamento di RDNET.
Una di queste è la modalità di ascolto.
Con questa nuova funzionalità del software non esisterà più solo un PC collegato alla rete dell’impianto acustico, ma si potranno connettere un numero illimitato di postazioni. Solo uno di questi, però, sarà considerato il master, quello che attualmente gestisce e controlla i dispositivi; i restanti saranno solamente “ascoltatori”, ovvero visualizzeranno il sinottico completo ed avranno aggiornamenti in tempo reale, ma potranno eseguire comandi sui dispositivi solo se autorizzati dal master.
Questa feature è attualmente in fase di sviluppo ed il suo piano di implementazione è già stato strutturato.
Il master eseguirà quindi una funzione di bridge tra gli ascoltatori ed i controllori.
La rete di comunicazione è mostrata in figura 3.7.

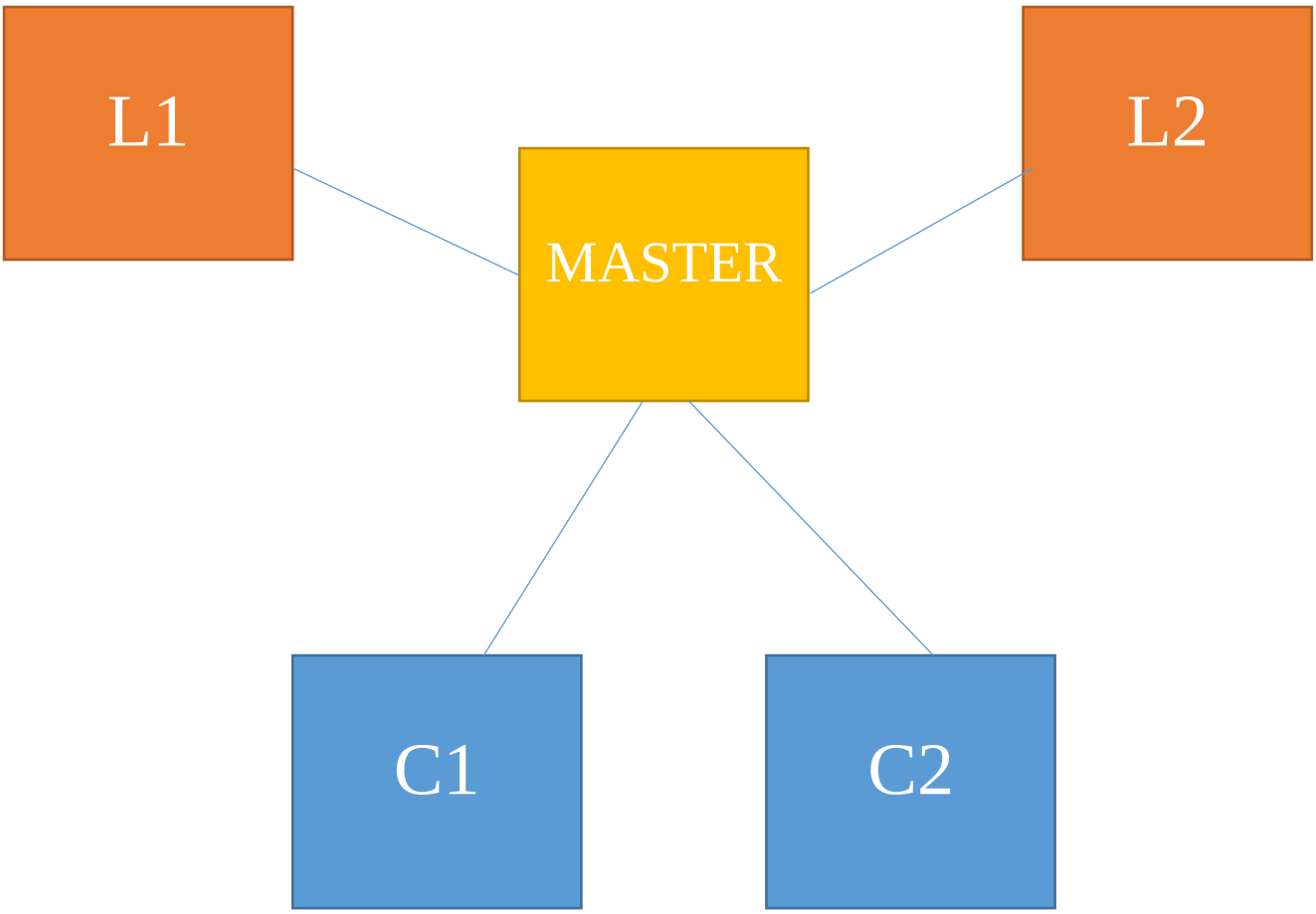

Fig. 3.7

Rete in presenza di software in modalità di ascolto.

Gli ascoltatori (L1, L2) comunicheranno unicamente con il master, questo perché i controllori possono essere connessi solo ad un PC alla volta.
Nell'interfaccia di RDNet si potrà avviare la connessione da ascoltatori a master mediante una richiesta TCP con allegata una password criptata. In questo caso è necessario inserire un controllo legato alla sicurezza per evitare che un qualunque utente in possesso del software possa, in un qualunque momento, connettersi in modalità di ascolto ad un PC che esegue attualmente una funzionalità di master. Questo per evitare qualunque tipo di inconveniente (rallentamenti sulla rete etc..). Il master, in base al numero massimo di ascoltatori che può gestire, deciderà se accettare o rifiutare.

A connessione avvenuta tutti gli ascoltatori potranno vedere il sinottico completo ed aggiornato in tempo reale. Anch'esso verrà passato sulla stessa connessione TCP. Su attivazione del master un ascoltatore potrà anche inviare comandi ad un controllore, ovviamente non direttamente, ma attraversando il master stesso, che si comporterà da server ed inoltrerà al controller corretto il comando. Da questo sistema il master verrà visto dagli ascoltatori come un controllore ed essi interagiranno con quest'ultimo come se comunicassero con un controllore reale.

Una seconda funzionalità che potrebbe essere sviluppata partendo dalla modalità di ascoltatore è la sostituzione di un Master nel caso, per qualunque motivo, si interrompa la connessione con i controllori. In questo caso verrebbe eletto un nuovo master tra tutti gli ascoltatori e ad esso passerebbe tutto il controllo dell'impianto. Nello sviluppo di questa feature si andrebbe incontro, tuttavia, a grossi problemi e a conflitti progettuali rispetto a quanto già esistente. Sarà quindi necessario eseguire un'accurata progettazione per implementarla utilizzando un metodo che dovrà essere quello meno invasivo.

4. CONCLUSIONI

Dopo la USB e l'RS232, per la comunicazione tra controllore e PC è stata finalmente implementata anche la possibilità di utilizzare un cavo ethernet.
Questa nuova funzionalità è la base di netti miglioramenti del programma per i seguenti motivi:

1) Ethernet non necessita di alcun tipo di driver installato sul Pc. Questo significa avere una maggiore comodità e flessibilità.
2) Il discostamento dall'utilizzo di driver evita problemi, quali ad esempio aggiornamenti mancati.
3) Con l'utilizzo di un cavo Ethernet si può connettere il PC ad N controllori alla stessa rete. In questo modo non vi è più la necessità più di un cavo per ogni connessione controllore – PC.
4) Il segnale in Ethernet riesce a percorrere centinaia di metri a differenza di un USB o RS232 che lo mantengono stabile per una decina di metri.

E' stato uno sviluppo lungo e molto difficoltoso, ma, una volta completato, ha dato grandi soddisfazioni sia a me come progettista che a chi, per primo, ha voluto testare questa nuova funzionalità.

5. REFERENCES

[1]. Tipologie di rete, http://it.ccm.net/contents/490-topologia-di-reti

[2]. The EXTENSION, Understanding EIA-485 Network, 1999, http://www.ccontrols.com/pdf/ExtV1N1.pdf

[3]. ARM 9, http://infocenter.arm.com/help/index.jsp

[4]. The Basic Reference Model for Open Systems Interconnection, http://www.ecma-international.org/activities/Communications/TG11/s020269e.pdf, 1984

[5]. Maris Graube, IEEE 802.3 approved as networking standard, June 1983

[6]. Gary McLean Hall, Pro WPF and Silverlight MVVM: Effective Application Development with Model-View-ViewModel, January 2011

[7]. Matthew MacDonald, Pro WPF in C# 2010, March 2010

[8]. Chris Selss, Windows Forms Programming in C#, September 2003

[9]. Postel, J., UDP, User Datagram Protocol, RFC 768, August 1980, https://www.ietf.org/rfc/rfc768.txt.

[10]. Postel, J., TCP, Transmission Control Protocol Specifications, RFC 793, September 1981, https://tools.ietf.org/html/rfc793

[11]. Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; and Vetterling, W. T. "Cyclic Redundancy and Other Checksums." Ch. 20.3 in Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed. Cambridge, England: Cambridge University Press, pp. 888-895, 1992.

[12]. Byte Stuffing - http://web.cs.wpi.edu/~rek/Undergrad_Nets/C02/BitByteStuffing.pdf