



UNIVERSITÁ DEGLI STUDI DI MESSINA

DIPARTIMENTO DI MATEMATICA E
INFORMATICA

CORSO DI LAUREA TRIENNALE IN INFORMATICA

Jarvis Domotica: L'Automazione Domestica Distribuita

Tesi di Laurea di:
Rocco Musolino

Relatore:
Ch.mo prof. Andrea Nucita

ANNO ACCADEMICO 2012-2013

INDICE

1 - INTRODUZIONE	5
1.1 Scopo del progetto	7
2 - COS'È LA DOMOTICA	8
2.1 Storia	8
2.2 Web of Things	9
2.3 Interessi Applicativi	10
2.4 Elementi essenziali	11
2.5 Costi e Diffusione	12
3 - METODOLOGIA DI SVILUPPO	14
3.1 Analisi dei requisiti	14
3.2 Metodologie di sviluppo	15
3.2.1 Jarvis PC	16
3.2.2 Jarvis Server	17
3.2.3 Jarvis Smartphone	18
3.2.4 Jarvis Microcontroller	20
4 - ARCHITETTURA	21
4.1 Progettazione Database	21
4.2 Progettazione Software	22
4.2.1 Modulo PC	23
4.2.2 Modulo Server	27
4.2.3 Modulo Smartphone	30
4.2.4 Modulo Microcontroller	32
4.3 Progettazione Hardware	32

4.3.1 Cos'è Arduino?	32
4.3.2 Prototipo: Prolunga Relè	36
4.3.3 Lunghezza di un'antenna ideale	40
4.3.4 Prototipo ambientazione suggestiva	41
4.3.5 Codifica e decodifica di segnali IR	43
4.3.6 Uno sguardo ai sensori	47
5 - SISTEMI ESISTENTI IN COMMERCIO	57
5.1 Il NinjaBlocks	57
5.2 Il Philips HUE	59
5.3 Le differenze con Jarvis	61
6 - CONCLUSIONI E SVILUPPI FUTURI	62
BIBLIOGRAFIA	66

1. INTRODUZIONE

Ne è passato di tempo, da quando l'informatica, settore di nicchia, immaturo e precoce, inizia a muovere i primi passi per diventare la disciplina più in voga degli ultimi 20 anni, con i primi prototipi di computer, i primi supporti di memorizzazione, le prime reti informatiche. Si è passati da timidi e goffi tentativi, alla più grande e inarrestabile rivoluzione tecnologica globale della storia, con protagonisti: i computer come li conosciamo oggi, la diffusione della banda larga, internet, l'informazione semantica, gli smartphone e la mobilità. La scienza di oggi è la tecnologia di domani, e quando parliamo di scienza e “domani”, pensiamo alla scienza interdisciplinare che più è destinata a ricoprire un ruolo fondamentale nella nostra vita quotidiana da qui agli anni a venire: la **Domotica**, dal latino *Domus*, *casa* ed robotica. Il complesso di tecnologie che più di ogni altra ha margini di sviluppo promettenti.

Il settore Domotico è in crescita perché parte della rivoluzione tecnologica cui siamo protagonisti, la scintilla alla quotidianità futura. L'automazione domestica comprende branche disciplinari che spaziano dall'elettrotecnica all'ingegneria edile, fino all'informatica. Tutte usate a beneficio di una migliore sicurezza, qualità della vita od consumo energetico con interfacce sempre più avanzate (vocale, lettura della retina, lettura impronte digitali) e attuatori sempre più specializzati (serrature elettroniche, termostati, servi lineari, relè).

Con **Jarvis Domotica** (o semplicemente, Jarvis), si racchiudono tutti gli elementi base di un sistema domotico distribuito (con accesso da più piattaforme indipendenti), estendendone al limite alcune caratteristiche, per una migliore scalabilità e flessibilità, utilizzando al meglio tecnologie software e hardware open source, già largamente consolidate in altri settori.

Questa tesi consta di una progettazione hardware e software. Questi due aspetti interagiscono strettamente per garantire un'esperienza utente “tangibile” e quanto più vicina alla realtà. Per garantire ciò, Jarvis prevede l'utilizzo di interfacce centralizzate/decentralizzate vocali e interfacce grafiche standard, distribuite su più

applicativi e su più sistemi non omogenei. L'utente interagisce con l'ambiente attraverso le interfacce messe a disposizione dal sistema.

Gli applicativi software da cui è composto Jarvis, prevedono un largo uso di Python, PHP, C, XML, Javascript, Ajax, HTML e CSS.

I componenti elettronici invece, utilizzano la piattaforma Open Source, Arduino, per comunicare con il sistema. Tra questi, si individuano ricevitori e trasmettitori FM a 433mhz, per la realizzazione di un protocollo di comunicazione radio, dei cavi seriali per il protocollo di comunicazione cablato. Degli emettitori e ricevitori infrarossi, per l'interazione del sistema con dispositivi esterni quali elettrodomestici: Condizionatori, Televisioni, Decoder. Relè allo stato solido per la commutazione di prese elettriche a 220V, quindi l'apertura e lo spegnimento di apparecchi collegati alla rete elettrica domestica.

Il principio seguito nella realizzazione del sistema Jarvis è quello del "Divide et Impera", ogni componente identifica un modulo, con regole ben specifiche per dialogare con il sistema e le interfacce. E' la suddivisione del progetto in moduli il vero punto di forza di Jarvis, il sistema si dota così di grande scalabilità e robustezza. Le scelte implementative si estendono su tutte le architetture supportate, che sia un calcolatore, un server o uno smartphone.

Una parte fondamentale del sistema domotico Jarvis, risiede nelle interfacce vocali, al cuore dell'implementazione dell'intero sistema. Esse permettono l'interazione diretta con l'utente, attraverso comandi predefiniti, riconosciuti e in seguito interpretati. Le interfacce vocali sono presenti anche nella versione smartphone di Jarvis, per accedere al pieno delle sue funzioni in tutta mobilità.

Le utilità di un sistema domotico abbracciano diversi aspetti del quotidiano, esso rappresenta un'unità portante o d'ausilio per l'intera struttura abitativa. La praticità di un sistema domotico sta nel garantire una certa facilità nell'esecuzione di operazioni predefinite. Jarvis estende questo concetto, presentando caratteristiche tipiche di un assistente vocale, con elaborazioni interne e risposte vocali contestuali.

1.1 Scopo del progetto

In questo elaborato, si passa in rassegna il progetto sviluppato, presentando una breve panoramica di quel che è la domotica, nel capitolo “Cos’è la Domotica”, per poi affrontare la metodologia di sviluppo, l’architettura alla base dell’hardware con la board open source Arduino e i componenti elettronici. Seguono il software, il lato client e i comandi riconosciuti da jarvis. Il lato server con l’interfaccia server e le richieste Cross-Domain in Ajax. Il lato smartphone e la simulazione di Python attraverso il progetto open source SL4A. Infine i casi d’uso reali del progetto Jarvis e le conclusioni con degli accenni ai miglioramenti che è possibile apportare. Lo scopo del progetto è realizzare un sistema domotico open source, per facilitare le operazioni quotidiane e offrire un’esperienza d’uso piacevole e all’avanguardia, compatibile con le tecnologie odierne, che presenti alcune delle caratteristiche base tipiche a sistemi di questo genere, pur introducendo caratteri di novità e semplicità da renderlo facilmente installabile e riproducibile dagli utenti. Con questo progetto si cerca di perseguire un approccio mirato all’abbattimento dei costi, è pensato per avere tempi d’installazione praticamente immediati, un basso costo di realizzazione e un basso costo di manutenzione. La personalizzazione e la facilità nell’upgrade sono altre caratteristiche che permettono al sistema di evolvere ed essere migliorato dalla comunità di internet.

2. Cos'è la Domotica?

“Domotics is the discipline that investigates how to realize an intelligent home environment”. - An Informatics Research Contribution to the Domotic Take-Off

La Domotica si riferisce a quella scienza interdisciplinare che sfrutta i computer e l'elettronica per il controllo intelligente di dispositivi, in un ambiente domestico. I sistemi domotici attualmente in uso sono molti, e spaziano dal semplice controllo remoto di luci, ad un più intelligente network di micro-controllori e sensori sparsi per casa. I benefici possono essere tanti, e maggiori nei casi di persone con particolari handicap motori.

Automazione è in un certo qual senso sinonimo di semplicità e risparmio. Molti compiti e processi quotidiani sono automatizzati da macchine sempre più specializzate, si pensi alle lavatrici, esse sono state sviluppate per semplificare e ridurre la quantità di lavoro manuale nel lavaggio della biancheria. Mentre i sistemi di allarme sostituiscono le sentinelle nelle mansioni di guardia e sicurezza. Sono tutti esempi d'automazione, piccoli tasselli di domotica a cui nessuno riesce più a rinunciare.

2.1 Storia

L'automazione domestica è da sempre uno scenario futuristico oggetto delle più stravaganti fiction TV. E' divenuta realtà solo dal 20th secolo grazie all'introduzione massiva dell'elettricità nelle case e la rapida evoluzione dell'informatica. I primi accenni di Domotica risalgono al 1960, quando Jim Sutherland, ingegnere della Westinghouse Electric, sviluppò un sistema chiamato ECHO IV, mai commercializzato. Grazie all'invenzione dei micro-controllori, i costi dell'elettronica di consumo si abbassarono rapidamente e tecnologie di controllo remoto ed automazione furono fin da subito adottate nelle infrastrutture industriali, per diminuire il bisogno di manodopera, avere un controllo totale sui macchinari e favorire la produzione di massa. Nonostante l'interesse, la fine degli anni 90 non diede il via alla diffusione delle tecnologie domotiche, ancora poco mature, costose e con la mancanza di un vero e proprio standard. Ritenute dunque, roba per “pochi”.

2.2 Web of Things

Quando si parla di web of things, letteralmente “internet delle cose”, intendiamo quegli scenari in cui gli apparecchi di ogni giorno, sono completamente integrati col web. Questo è possibile laddove ogni singolo device è dotato di un micro-controllore adibito alla connessione internet. L’internet delle cose espande l’eco-sistema dei dispositivi connessi a internet, non si parla più soltanto di computer o smartphone, ma di TV ed elettrodomestici di varia natura. Quanto può entusiasmare l’idea di un frigorifero connesso a internet? Che sappia comunicare e avvisare prima della scadenza di un prodotto? Nell’internet delle cose ogni dispositivo è connesso in rete, pronto e accessibile ovunque noi siamo. Questo scenario non si applica solo alla domotica, l’internet delle cose è già in sperimentazione in ambito Urbanistico, con molti progetti all’attivo:

- Smart Parking:

Molte città hanno spazi riservati ai parcheggi, connessi 24h su 24h alla rete, in grado di comunicare quando i parcheggi si liberano.

- Traffic Congestion:

Non è una novità, molti sensori in città sono predisposti al controllo del traffico, per evitare congestioni o indicare tragitti migliori alle utenze, essi comunicano costantemente questi dati agli automobilisti e agli addetti al traffico cittadino.

- Noise urban maps:

Mappe urbane che indicano in tempo reale, quanto rumore è presente in determinate zone della città.

E così via per molti altri settori quali: Agricoltura, Controllo industriale, Logistica.

2.3 Interessi Applicativi

La Domotica è adottata per ragioni di semplificazione, sicurezza ed efficienza energetica. Si pensi al controllo di ambienti climatizzati, ai sistemi d'allarme, all'irrigazione automatica o alle serrature elettroniche. Nelle semplici installazioni, "domotica" può essere considerata l'illuminazione automatica di una stanza al rilevamento di una presenza. Nelle installazioni più avanzate invece, la stanza può rilevare non solo la presenza ma sapere esattamente chi è entrato, settando una luce appropriata, una temperatura adeguata, il volume della musica, elencare persino le notizie del giorno. Altre operazioni automatizzate potrebbero riguardare il risparmio energetico. E' possibile immaginare lo scenario in cui la casa rimane vuota, e le luci vengono spente, i dispositivi d'allarme attivati e i sistemi di climatizzazione settati nelle modalità di basso consumo. L'impiego di sensori di presenza possono infatti far risparmiare ore di energia, sprecata da luci, quando non siamo in casa o nella stanza cui sono preposti i sensori. Non è difficile trovare funzionalità del genere, infatti, nei sistemi domotici, affiancate da molte altre caratteristiche tipiche ai sistemi d'illuminazione, quali:

- Apertura e chiusura di luci con telecomandi wireless.
- Apertura e chiusura di luci a timer.
- Controllo della luminosità delle luci in base alla luce ambientale rilevata.
- Possibilità di cambiare colore alle luci (LED RGB).
- Regolazione della luce ambientale proveniente dalle finestre.
- Controllo delle luci via smartphone e tramite internet.

Con un monitoraggio costante in remoto, diviene facile avere accesso in mobilità a tutte le funzioni di casa, controllando in tempo reale la temperatura degli ambienti, ma anche accertarsi che non vi siano rilevazioni di fuoriuscite di gas o fumo da parte dei sensori. Questi sono solo alcuni aspetti della Sicurezza utilizzati nei sistemi domotici:

- Rilevamento di incendi, fuoriuscite di gas.
- Teleassistenza.
- Simulazione di presenza

- Rilevamento di presenza
 - o Sensori di movimento
 - o Sensori magnetici per porte e finestre
 - o Sensori di pressione
 - o Sensori di rottura vetri

2.4 Elementi Essenziali

Gli elementi essenziali di un sistema Domotico includono:

Sensori:

- Sensori di Temperatura
- Sensori Fotocellule (per la luce ambientale)
- Sensori di Movimento (PIR)
- Sensori di Fumo
- Sensori di Gas
- Sensori di Pressione

Controllori:

- Centraline
- Micro-controllori per elaborazione
- Unità centralizzata di calcolo (un PC di servizio preposto ad alcuni compiti)

Attuatori:

- Valvole motorizzate
- Interruttori
- Motori
- Relè (Meccanici o Solidi)
- Attuatori lineari

Interfacce:

- Radiocomandi / Telecomandi.
- Interfaccia Vocale.
- Interfaccia Smartphone / Web.
- Monitor grafico preinstallato.

Network:

- Bus cablato.
- Comunicazione radio.

2.5 Costi e diffusione

Una casa automatizzata può essere un semplice raggruppamento di “controlli”, oppure può essere pesantemente automatizzata, al punto da poter controllare ogni apparecchio collegato in casa (Figura 2.1). I costi variano largamente in base alle soluzioni adottate, alla fornitura, ai componenti e all’installazione.

I costi nel tempo invece, possono riguardare il consumo di elettricità da parte del sistema, i costi di manutenzione dei componenti, o di aggiornamento e upgrade di quest’ultimi. Imparare a usare un sistema domotico complesso può effettivamente richiedere tempo, e gli addetti all’installazione devono essere adeguatamente preparati, con training specifici per il sistema che si installa (I sistemi domotici possono variare molto l’uno dall’altro, e così la loro installazione). Questo influisce molto sulla diffusione dei sistemi domotici nel mondo. Un buon sistema richiede che l’abitazione su cui venga installato sia predisposta per l’installazione, questo comporta il tenere in conto a priori della messa in funzione di un sistema di questo tipo nella propria casa. E’ durante la progettazione edile che il sistema trova già una collocazione. Per la stragrande maggioranza dei casi invece, è necessario adeguarsi ad abitazioni già costruite che non prevedevano in origine la messa in funzione di un sistema del genere. Sistemi meno invasivi, che si adattino senza predisposizione, esistono, pur rinunciando all’eleganza nell’installazione e talvolta a qualche funzionalità. “Grazie inoltre al Decreto Sviluppo, varato dal Governo, sarà economicamente vantaggioso riparare e mettere in regola gli impianti elettrici

domestici, usufruendo del comfort e della sicurezza di un impianto domotico allo stesso prezzo di un impianto tradizionale. “. Secondo una ricerca di “ABI Research”, nel 2012 negli Stati Uniti sono stati installati circa 1.5 milioni impianti domotici, con delle stime che prevedono di raggiungere gli 8 milioni entro in 2017.

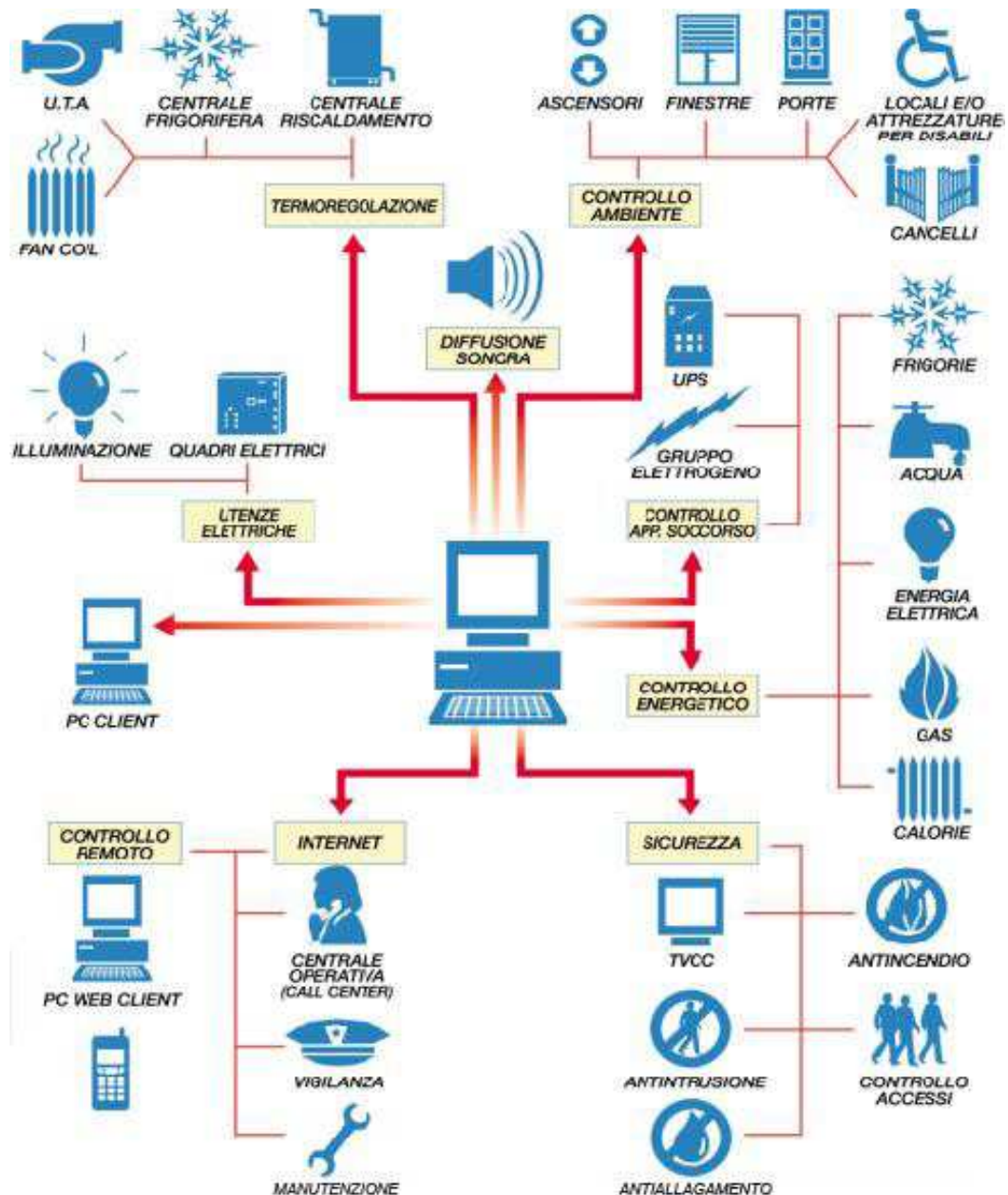


Figura 2.1 – Un sistema domotico completo

3. METODOLOGIA DI SVILUPPO

3.1 Analisi dei Requisiti

I requisiti da avere sono:

- Interfacce multiple
 - # Vocale
 - # Grafica
 - # Smartphone & Web
 - # Telecomando IR
- Facile espandibilità
- Installazione e configurazione rapida
- Protocollo di comunicazione cablato e via radio
- Assistente vocale
- Riconoscimento di comandi standard e interattivi
- Possibilità di controllare l'impianto audio
- Possibilità di pilotare elettrodomestici (TV, condizionatori, etc.)
- Controllo dell'illuminazione
- Sistema per la creazione di luce d'ambientazione suggestiva
- Controllo degli accessi
- Rilevamento di presenza
- Allarmi funzionali
- Autenticazione tramite tecnologie RFID

- Monitoraggio di temperature
- Sensori di allarme gas, metano, butano.
- Accessibilità del sistema dall'esterno
- Gestione delle credenziali abbinata ai tag RFID.

3.2 Metodologie di sviluppo

Le metodologie di sviluppo utilizzate nella realizzazione di questo progetto sono due. Per il lato hardware una progettazione bottom-up ha permesso di realizzare velocemente i prototipi di attuatori e interfacce periferiche, agevolando quindi il test d'unità, ancor prima d'aver assemblato l'intero sistema. Il lato software segue invece una più classica metodologia di sviluppo top-down: si resta focalizzati su un obiettivo, il codice viene scritto in maniera metodica e con uno scopo preciso. Tipico della progettazione Top-Down è il concetto di "Divide et Impera", Dividi e Domina, locuzione latina a indicare la suddivisione e frammentazione di un "progetto" al fine di aver un miglior controllo su di esso. E' un concetto molto diffuso in informatica, suddividendo il problema in sotto-problemi semplici si procede più velocemente e facilmente. Jarvis segue questo principio, è completamente modularizzato. In particolare, distinguiamo 4 macro suddivisioni del progetto:

- Jarvis PC
- Jarvis Server
- Jarvis Smartphone
- Jarvis Microcontroller

Ogni modulo ha delle proprie specifiche, delle proprie funzionalità e un'apposita architettura su cui essere eseguito. Inoltre, per la realizzazione sono stati utilizzati differenti linguaggi di programmazione che più soddisfavano i requisiti di semplicità ed efficienza.

3.2.1 Jarvis PC

Jarvis PC, è il cuore del sistema. Ha le caratteristiche di un server, ed è pensato per girare su un pc casalingo che funga da centrale per il coordinamento degli altri componenti. Il core del sistema gira in PHP, è quindi necessario dotare il PC di un software per web server e relativo interprete PHP (Vedi pacchetto applicativo XAMPP – Figura 3.1).

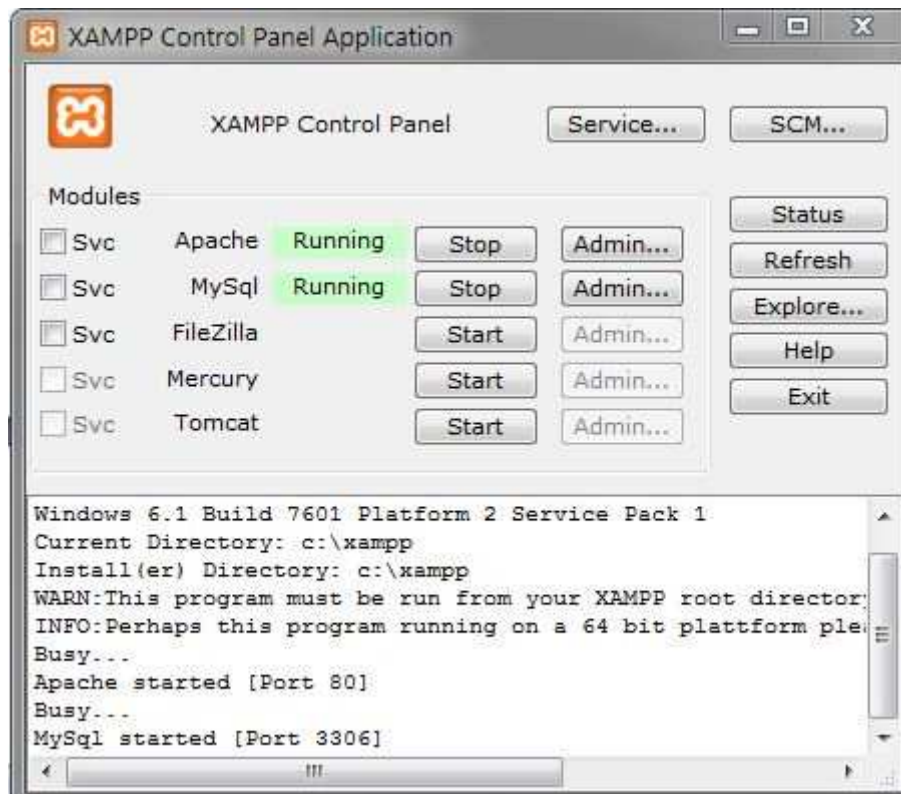


Figura 3.1 – Il pacchetto XAMPP avviato

Per l'avvio si richiede che venga dunque posizionata una scheda del browser sull'indirizzo locale. La index, pagina principale di Jarvis PC, nella sua veste grafica pulita si compone di una casella per il riconoscimento vocale, la possibilità di selezionare la lingua parlata e di uno spazio dedicato all'esito delle risposte dei comandi (Figura 3.2). L'interfaccia vocale presente in Jarvis PC permette la piena interazione con il sistema. Il modulo PC di Jarvis funge da "dispatcher", colui che inoltra a destinazione i comandi provenienti dalle varie interfacce disponibili. Ricevuto un comando, viene analizzato (nei capitoli successivi si vedrà come avviene l'analisi effettiva e il controllo del comando), e se correttamente

riconosciuto, eseguito. Da buon dispatcher, il modulo invia dei segnali di riconoscimento che verranno raccolti dalle riceventi sparse in casa, ed effettivamente eseguite dall'attuatore. L'interazione software-hardware avviene sfruttando linguaggi di programmazione che offrono supporto per la scrittura di codice di basso livello, come il C, o alcune librerie seriali di Python che verranno trattate più avanti.

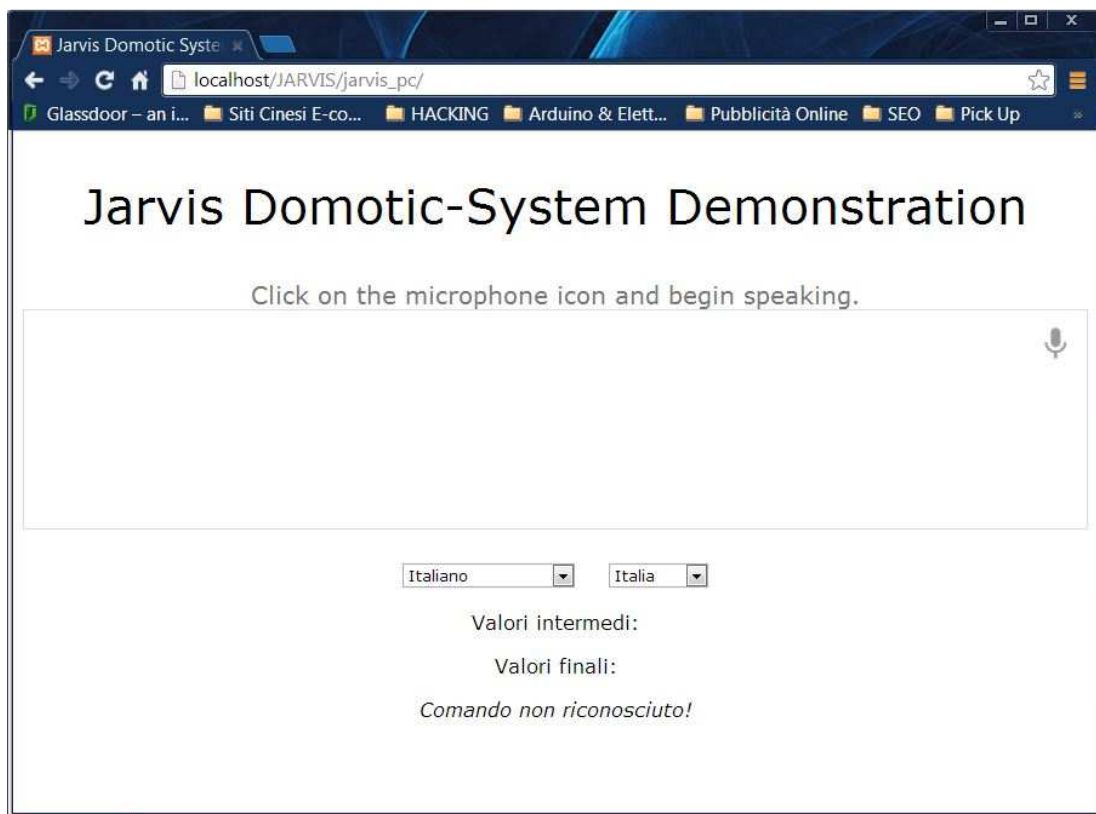


Figura 3.2 – Interfaccia vocale del modulo Jarvis PC

Oltre al riconoscimento vocale, è possibile dialogare con il modulo PC utilizzando altri terminali quali Jarvis Server o Jarvis Smartphone.

3.2.2 Jarvis Server

Il modulo Server è dotato di interfaccia grafica (Interfaccia grafica base - Figura 3.3), raggiungibile dall'esterno, esso risiede infatti su un server remoto, raggiungibile attraverso un nome di dominio 24h su 24h, il modulo server e il modulo pc godono

entrambi della tecnologia ajax per poter effettuare chiamate asincrone e controllare in real time l'uno lo stato dell'altro, dialogando in maniera trasparente. Con l'interfaccia grafica perennemente in linea è possibile avere ogni funzionalità domotica della propria abitazione a portata di click.

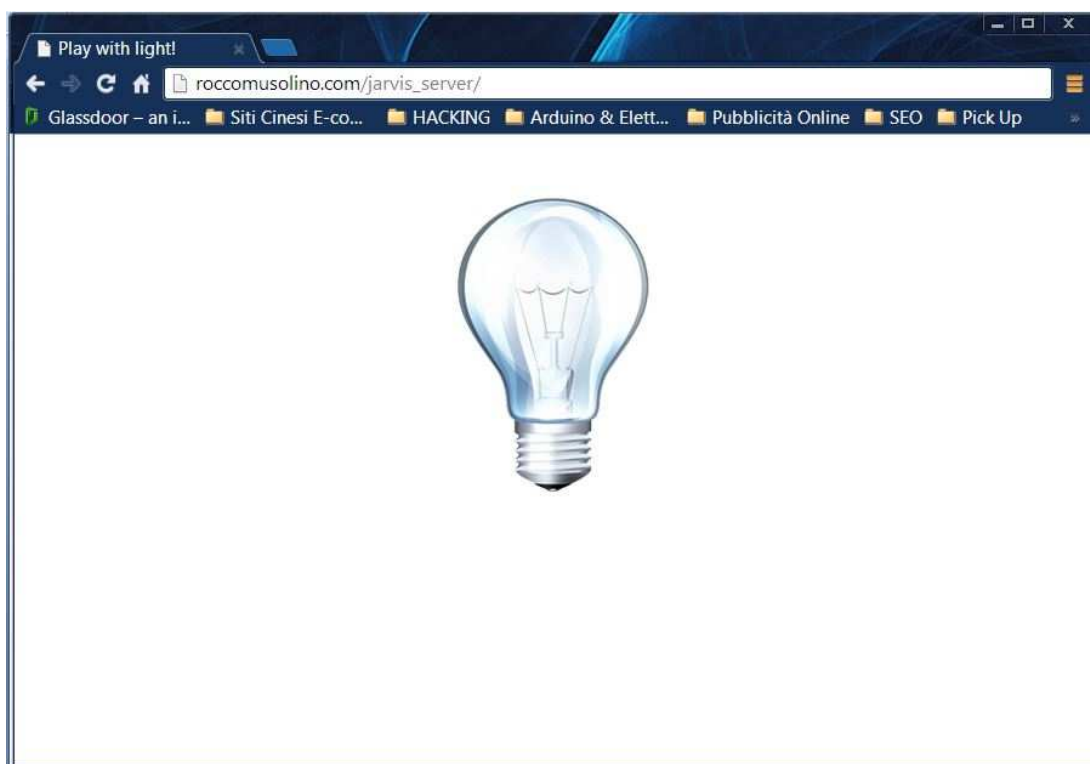


Figura 3.3 – Interfaccia grafica base del modulo Server

La particolarità di essere disponibile 24h/24h lo rende inoltre adatto per poter ricevere comandi da dispositivi mobile che necessitano di interagire col sistema in piena mobilità, è il jarvis server che funge da tramite nell'interazione con i dispositivi smartphone, pur non avendo la stessa interfaccia in comune.

3.2.3 Jarvis Smartphone

Lo Smartphone è ciò che più si avvicina al concetto di “telecomando intelligente”, lo abbiamo sempre con noi e assume un ruolo importante nella nostra routine quotidiana. Abbiamo assistito al boom della telefonia, e l'interfaccia che più di ogni

altra non può mancare in un sistema domotico è forse quella smartphone. Jarvis Smartphone è quel modulo del sistema Jarvis installabile su cellulari con sistema operativo Android. Si tratta di un'app sviluppata in un'ambiente simulato con SL4A, le peculiarità risiedono nell'avere un controllo vocale e grafico su smartphone, in completa mobilità, senza dover necessariamente ricorrere a device fissi e decisamente meno portatili (Figura 3.4). L'esecuzione di un comando avviene su due livelli, il primo è quello smartphone, in cui si cerca una corrispondenza con la lista di definizioni dei comandi. Se viene trovata, il comando è eseguito localmente sullo smartphone, vuol dire dunque che il comando riconosciuto era effettivamente destinato allo smartphone, se nessuna corrispondenza viene trovata, il comando ricevuto viene inoltrato al modulo Server che terrà il comando in memoria, in attesa che il modulo PC lo prelevi e lo esegua all'interno dell'abitazione.

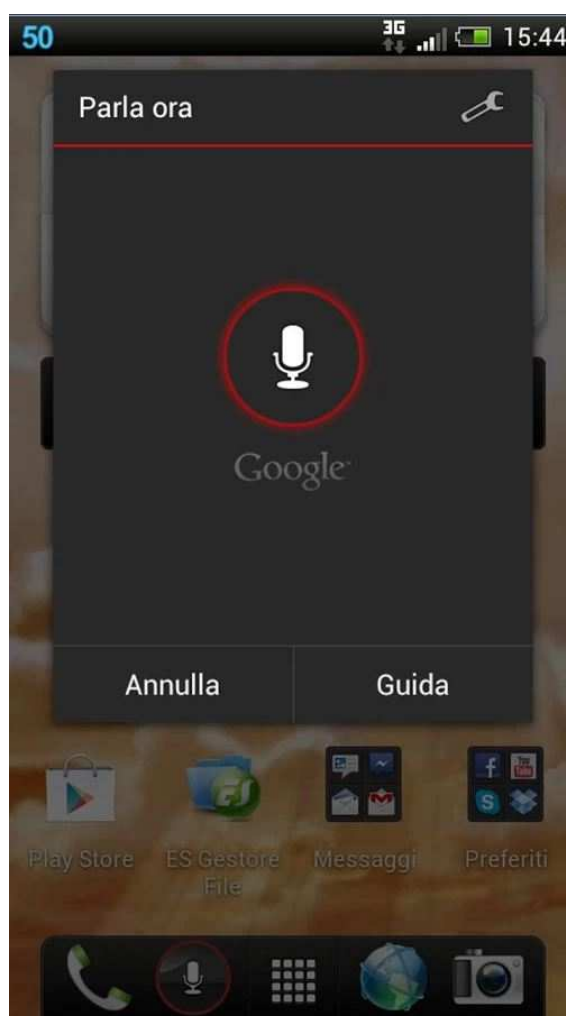


Figura 3.4 – Esecuzione di uno script su smartphone tramite SL4A

3.2.4 Jarvis Microcontroller

La comunicazione con i dispositivi di casa, quindi gli attuatori alla base della tecnologia domotica, è il Jarvis Microcontroller, il componente hardware che si fa da portavoce per il segnale riconosciuto dal modulo Jarvis PC. Il microcontrollore utilizzato fa parte della board open source Arduino, esso è collegato in seriale con il modulo PC e permette una comunicazione rapida sullo standard USB. A sua volta il microcontrollore è collegato ad opportuni trasmettitori radio per l'invio del segnale alle stazioni riceventi sparse per la casa, che pilotano gli opportuni attuatori. Questo modulo è fondamentale nella realizzazione di un sistema domotico. Rappresenta il confine tra ciò che è virtuale, all'interno di calcolatori elettronici, a ciò che è tangibile.

Definiti i moduli principali del progetto, la topologia alla base della rete Jarvis è facilmente deducibile e visibile nella figura 3.4.1 .

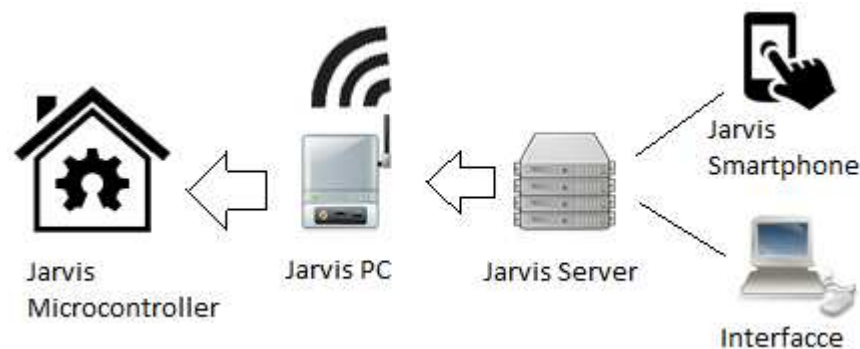


Figura 3.4.1 – Topologia di rete di Jarvis

4. ARCHITETTURA

Definita la topologia del progetto, possiamo ad analizzare nel dettaglio come ogni singolo modulo e componente del sistema è stato strutturato e realizzato, per dialogare in piena armonia con gli altri.

4.1 Progettazione Database

Durante la progettazione di Jarvis, l'utilizzo di una base di dati è stata fondamentale per meglio soddisfare i requisiti del sistema. In particolare la conservazione dei dati assume un aspetto fondamentale quando si ha a che fare con credenziali d'accesso che necessitano di essere opportunamente conservate in tutta sicurezza e di dati che invece, necessitano di essere accessibili dall'esterno per monitoraggi e applicazioni specifiche. Fatta questa premessa, in Jarvis sono due i tipi di base di dati che entrano in gioco, da un lato un DBMS ampiamente conosciuto sul web quale MySQL, dall'altro un tipologia d'archiviazione pubblica che meglio di ogni altra si presta per avere interazioni e letture pubbliche dall'esterno, i file XML.

MySQL è un DBMS basato sul modello relazionale, esso fa uso del linguaggio dichiarativo SQL (Structured Query Language) per le interrogazioni. Nel web è diventato quasi uno standard, per via della sua facilità d'uso e la spinta data da CMS di blogging gratuiti quali Wordpress, Joomla, che sfruttano questo DBMS per l'archiviazione dei contenuti.

Il linguaggio XML (Extensibil Markup Language), più che un database è un linguaggio di markup, quindi di marcatura, utilizzato per marcare documenti o porzioni di testo strutturate. Il vantaggio nell'usare un linguaggio di questo tipo per marcare i contenuti è di tipo funzionale. I documenti strutturati infatti, contribuiscono alla definizione di quel che è il web semantico, in cui il contenuto di ogni pagina web è facilmente individuabile da utenze umane e da utenze automatiche, i cosiddetti "bot" ad esempio, che girano per la rete.

In Jarvis MySQL è stato utilizzato per contenere i dati d'autenticazione al sistema. E' infatti possibile creare più utenti all'interno del database e assegnare ad ognuno di essi un ruolo che sarà poi determinante all'interno del sistema per selezionare le funzioni ad esso riconducibili, sempre in base ai privilegi di quel ruolo. L'amministratore ha il completo accesso agli utenti e al sistema in tutte le sue funzionalità. L'uso di XML è stato invece necessario, per poter offrire "una finestra al mondo esterno", tutti i dati dei sensori vengono infatti memorizzati direttamente dal modulo PC di Jarvis, su un file XML, esso è accessibile esternamente, quindi con i contenuti strutturati in modo opportuno, è possibile ad esempio conoscere la temperatura di una determinata stanza da un qualsiasi applicativo che si interfaccia al

file XML. L'utilizzo di un file XML è pratico per la formattazione dei dati, dati che dovranno essere poi presi in rassegna da applicazioni esterne, qui viene dunque evidenziata l'importanza del "parsing" delle informazioni. In un mondo strutturato, l'informazione può essere veicolata in modo automatico. La scelta di XML è fondamentale nell'interazione fra applicativi che nativamente non avrebbero modo di comunicare fra loro.

Nel progetto si evidenzia inoltre, l'uso di file XML per la configurazione dello stesso. Per le stesse ragioni esposte sopra, se più applicativi desiderano conoscere la configurazione attuale del sistema, gli basterà interrogare il file xml.

Qui di seguito nella figura 4.1 un esempio di file di configurazione dove si specificano i percorsi di default per l'archivio musicale e dei film:

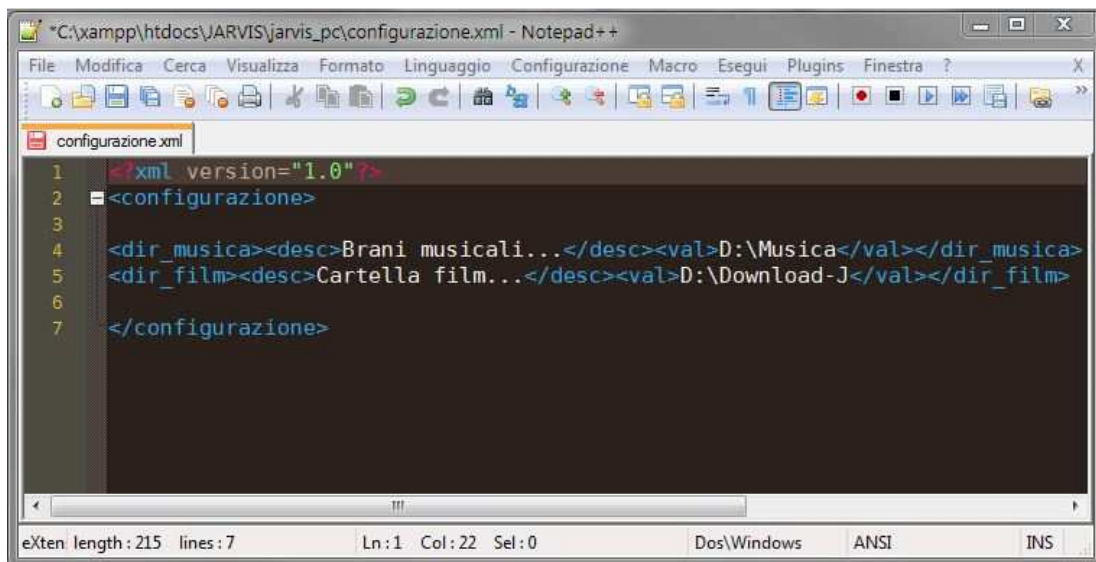


Figura 4.1 - configurazione.xml

4.2 Progettazione Software

Precedentemente si è visto come una metodologia di sviluppo di tipo Divide et Impera, sia adatta a grandi progetti strutturati. La frammentazione dei componenti avviene in primis qui, in quello che è il software di Jarvis. Un insieme di script e applicativi, preposti al riconoscimento e all'esecuzione dei comandi. La suddivisione vista precedentemente riguardava gli attuali 4 moduli che formano in toto il sistema. Si analizzano adesso, le suddivisioni attuate all'interno dei moduli, partendo dal primo e fondamentale Jarvis PC fino ad arrivare al Modulo Microcontroller.

4.2.1 Modulo PC

Il modulo PC è la parte essenziale dell'intero sistema, permette di ricevere comandi dalle interfacce, filtrarli, interpretarli ed eseguirli comunicando direttamente con il modulo Microcontroller. Il modulo è composto da un insieme di file qui elencati:

- File di configurazione (configurazione.xml).
- Motore d'esecuzione (execution-engine.php).
- Interfaccia vocale (index.html).
- Lista dei comandi riconosciuti dall'interfaccia vocale (lista-comandi.xml).
- Le funzioni implementate (moduli-funzione.php).
- Assistente vocale, con funzioni di TTS (tts.php)
- Comunicazione seriale (serial-communication.py)
- Esecuzione di contenuti multimediali locali (play.py)

La pagina d'avvio è index.html, essa detiene l'interfaccia vocale e l'interazione asincrona con il server e i dispositivi attuatori. Quando si parla d'interazione asincrona, si intende una richiesta effettuata in maniera asincrona con tecnologia AJAX (sfruttando la libreria javascript jQuery). Questa tecnologia è asincrona perché non in sincro con il normale ciclo di "botta e risposta" tipico della comunicazione client-server attraverso il protocollo HTTP.

L'interfaccia vocale è possibile a partire dalla versione di Google Chrome 25. Una delle più interessanti caratteristiche introdotte è appunto il supporto al riconoscimento vocale attraverso le Web Speech API. Le Web Speech API sono librerie javascript per consentire il riconoscimento "Speech to Text". Supporta svariate lingue, dall'inglese all'italiano, al francese e così via. Attualmente gli sviluppatori hanno due modi per implementare il riconoscimento vocale con queste API sulle proprie pagine. La prima è attraverso il webkit all'interno di Chrome, la seconda è definendo l'oggetto nel linguaggio javascript. Realizzare un motore di riconoscimento vocale da 0 è un lavoro che può richiedere anni, appoggiarsi alle librerie gratuite offerte da google, permette di risparmiare tempo, con una qualità accettabile. Google dà anche la possibilità di vedere i risultati intermedi durante il riconoscimento, molte parole infatti possono essere ambigue per il motore vocale; il risultato intermedio altro non è che la parola più probabile fra quelle riconosciute, ma non ancora definitivamente confermata. Usare le API Google velocizza lo sviluppo ma presenta anche degli svantaggi, che risiedono per lo più nel dover sottostare ad alcune regole imposte da Google, come:

- L'utente deve dare il consenso al browser di accedere al microfono prima di iniziare una sessione d'ascolto (con i siti in https, il permesso dopo che viene dato una volta, non viene nuovamente richiesto).
- La sessione d'ascolto è limitata a 60 secondi.

- Il riconoscimento vocale è disponibile solo su Chrome, gli altri browser quali Safari, Firefox, Internet Explorer, ci si aspetta lo supportino a breve.

Lo svantaggio comunque più notevole, per entrambi i metodi e per le API fornite da google, è che il riconoscimento avviene sui server in remoto e non localmente, quindi è necessario disporre di una connessione internet sempre attiva. (Motivo per cui il riconoscimento continuo ha comunque un limite di 60 secondi).

Ad un intervallo fisso avviene l'interazione asincrona fra il modulo pc e il modulo server, entrambi devono rimanere sincronizzati fra loro. Il server per aggiornare l'interfaccia, il modulo pc per poter controllare l'arrivo di nuovi comandi dal server. Un qualsiasi comando vocale o non, viene captato dal file index.html e mandato al file execution-engine.php, il motore d'esecuzione, il core di questo modulo. Il motore d'esecuzione riceve il comando grezzo. Il comando grezzo (altro non è che una serie di parole) viene manipolato per ottenere una formattazione universalmente riconosciuta nel file dei comandi, viene quindi convertito in minuscolo, ripulito da lettere accentate ed i numeri convertiti in caratteri arabi (esempio la parola "due" diventa effettivamente 2). Le parole che compongono il comando successivamente vengono suddivise, per formare un array, così da effettuare i controlli su singole parole e non necessariamente su tutta la frase in ordine (esempio: "Musica stop" equivale a "Stop Musica"). L'array è confrontato con il dizionario di frasi presente nel sistema jarvis, il dizionario è un elenco di comandi testuali a cui è associata una determinata funzione, una singola funzione può essere associata a più comandi testuali, così da garantire l'inserimento di più sinonimi per una singola parola. Il dizionario è contenuto nel file lista-comandi.xml (Figura 4.2) e si suddivide in due tipologie di comandi, da un lato i comandi standard, dall'altro i comandi interattivi. Differenziare i comandi è stato necessario per poter garantire in alcune frasi riconosciute vocalmente, una certa "personalizzazione" o "interattività". Se volessimo riprodurre un brano musicale di cui conosciamo già il titolo o l'autore ci basterebbe dire "avvia cremonini". Appare subito chiaro come "cremonini" non possa essere una parola contenuta all'interno del dizionario, poiché in questo modo dovremmo definire una voce nel dizionario per ogni canzone o film che abbiamo nel nostro computer, il che renderebbe il dizionario estremamente grande e ingestibile. L'idea del comando interattivo è riconducibile all'idea di una funzione con parametri in ingresso, una funzione senza parametri in ingresso ci si aspetta faccia qualcosa di "statico" nel tempo, in modo fisso e senza variazioni. Una funzione con un parametro in ingresso invece, è influenzata dal parametro che gli viene passato, di conseguenza l'output potrà variare largamente. Alla stessa maniera il comando interattivo, si aspetta che una parte del comando sia variabile, come nel caso "avvia cremonini". "avvia" è la parte fissa che identifica una funzione che si aspetta valori in ingresso, questi valori in ingresso altro non sono che la restante parte del comando, nel nostro esempio "cremonini", che poteva essere un valore arbitrario.

L'esempio di un comando standard classico potrebbe essere quello “apri la luce” oppure “spegni luce”. Il sistema prenderà il comando così com'è, effettuerà la scomposizione delle parole, se riconosciuto nella parte del dizionario che contiene comandi classici non si aspetterà nient'altro ed eseguirà la funzione associata a questo comando. Nell'execution-engine viene anche definita la priorità fra i comandi. I comandi interattivi hanno maggiore priorità dei comandi classici, e fra i comandi stessi la priorità maggiore è detenuta da chi si trova più in cima nel dizionario.

```
<!-- LUCE illuminazione ON -->
<cmd><text>apriti luce</text><funzione>luceON</funzione></cmd>
<cmd><text>apri luce</text><funzione>luceON</funzione></cmd>
<cmd><text>apri la luce</text><funzione>luceON</funzione></cmd>
<cmd><text>accendi luce</text><funzione>luceON</funzione></cmd>
<cmd><text>illumina stanza</text><funzione>luceON</funzione></cmd>
<cmd><text>attiva luce</text><funzione>luceON</funzione></cmd>
<cmd><text>piu luce</text><funzione>luceON</funzione></cmd>
<cmd><text>troppo scuro</text><funzione>luceON</funzione></cmd>

<!-- LUCE illuminazione OFF -->
<cmd><text>meno luce</text><funzione>luceOFF</funzione></cmd>
<cmd><text>spegni luce</text><funzione>luceOFF</funzione></cmd>
<cmd><text>spegni la luce</text><funzione>luceOFF</funzione></cmd>
<cmd><text>chiudi luce</text><funzione>luceOFF</funzione></cmd>
<cmd><text>chiudi la luce</text><funzione>luceOFF</funzione></cmd>
<cmd><text>stop luce</text><funzione>luceOFF</funzione></cmd>
<cmd><text>disattiva illuminazione</text><funzione>luceOFF</funzione></cmd>
<cmd><text>troppa luce</text><funzione>luceOFF</funzione></cmd>

<!-- Orario e Data -->
<cmd><text>orario</text><funzione>time_ora</funzione></cmd>
<cmd><text>che ore sono</text><funzione>time_ora</funzione></cmd>
<cmd><text>dimmi che ore sono</text><funzione>time_ora</funzione></cmd>
<cmd><text>data</text><funzione>time_data</funzione></cmd>
<cmd><text>data odierna</text><funzione>time_data</funzione></cmd>
<cmd><text>quanto abbiamo oggi</text><funzione>time_data</funzione></cmd>
```

Figura 4.2 – Il dizionario, con i comandi, sinonimi e funzione associata

I controlli discussi sopra, abbiamo visto, sono effettuati dal motore d'esecuzione. Se esiste un match fra il comando ricevuto e una voce del dizionario, la palla viene passata al file: moduli-funzione.php. Esso contiene tutte le definizioni di funzioni eseguibili in Jarvis. Le funzioni, per lo stesso discorso affrontato poco fa, possono avere o meno parametri in ingresso. Questa parte di programma si occupa di eseguire la funzione che il comando ha evocato. Per implementare facilmente nuove funzioni aggiuntive è sufficiente aggiungerla nel file moduli-funzione.php, e scriverne il nome abbinato nel file lista-comandi.xml assieme alle frasi che verranno associate all'esecuzione di questa funzione. In questo modo il sistema è facilmente scalabile, si possono aggiungere nuove funzionalità in tutta semplicità, senza dover entrare nei meandri del codice.

Ogni funzione definita in questo file di moduli esecutivi può includere script esterni, o eseguire comandi attraverso script python per funzioni legate maggiormente all'hardware e alla macchina locale (per questo si necessita di un apposito file di configurazione che contenga valori di default usati da Jarvis – vedi `configurazione.xml`), ed è questo che avviene quando un comando come “accendi luce” viene riconosciuto e mandato in esecuzione. Si richiamerà lo script python predisposto alla funzionalità di comunicazione seriale. La comunicazione seriale è il collegamento USB che avviene fra il computer e il microcontrollore. Il microcontrollore fa da confine al lato software. Questa comunicazione è a carico del file `serial-communication.py` che viene chiamato tramite linea di comando dal file `moduli-funzione.php`. Lo script python adibito alla comunicazione seriale, ha il compito di scansionare tutte le porte seriali del computer alla ricerca di un dispositivo attivo connesso e in ascolto, una volta rilevato, avvia la comunicazione sfruttando le librerie python `pyserial`, gli invia il comando e chiude la comunicazione seriale restituendo l'esito di risposta al file chiamante. Un altro esempio di comando che vede un'interazione con la macchina locale è “riproduci film il gladiatore”, questo presuppone una ricerca all'interno del computer, e più nello specifico nella cartella destinata a contenere tutto l'archivio film, per poter avviare sul computer locale il film con titolo “il gladiatore”.

Il feedback dei comandi è essenziale per capire l'esito di un comando. In Jarvis il feedback può essere in due modi, testuale: la conferma è visibile graficamente sullo schermo. Oppure pronunciato vocalmente: una voce avvisa che l'operazione è stata completata. Nel secondo caso, si introduce la classe per la pronuncia vocale, TTS (Text to Speech). La classe TTS compone l'assistente vocale di Jarvis. Un sistema di pronuncia vocale, al pari di un sistema di riconoscimento vocale, richiede molto lavoro per essere sviluppato e quasi nessuno offre gratuitamente questi motori vocali. Il motore per la pronuncia usato in questo progetto, è ancora una volta, appartenente a Google, se pur sprovvisto di API per il supporto agli sviluppatori, in rete si trovano classi che si appoggiano a questo motore, per trasformare del testo in un file audio. La classe utilizzata in Jarvis viene riportata interamente nella figura 4.3.

```

<?php
class TextToSpeech {
    public $mp3data; // attributo della classe

    function setText($text,$file_name,$path_dir) { // metodo di classe
        $text = trim($text);
        if(!empty($text)) {
            $testo_originale = $text;
            $text = urlencode($text);

            $this->mp3data = file_get_contents("http://translate.google.com/translate_tts?tl=it&q={$text}");

            $put_file = $path_dir.$file_name.".mp3";
            file_put_contents($put_file, $this->mp3data);
        } else { return false; }
    }
}
?>

```

Figura 4.3 – Classe TTS per la pronuncia vocale

Quel che appare dal codice è che questa classe non implementa nativamente un motore vocale, ma si appoggia a Google, e al motore vocale utilizzato dal servizio Google Translate. Tutto ruota attorno al fatto che è possibile generare un output vocale dando un parametro *q* a questa URI, ed eventualmente è possibile anche cambiare lingua con il parametro *tl*:

http://translate.google.com/translate_tts?tl=it&q=TestoDaPronunciare

Questo meccanismo è la voce dell'assistente vocale Jarvis. L'assistente vocale nell'attuale implementazione è in grado di pronunciare l'ora e la data attuale con il comando "che ore sono", ripetere frasi ascoltate in precedenza oppure dare conferma dei comandi.

4.2.2 Modulo Server

Con il modulo server il progetto assume una connotazione distribuita. Il server si interpone fra la rete locale in cui gira il modulo PC, e il cloud, punto d'accesso per l'utenza da remoto. Il modulo Smartphone accede in mobilità al sistema casalingo sfruttando questo modulo Server. La realizzazione consta di pochi file essenziali:

- File che contiene il comando attuale (comando.txt)
- File di log, per tenere traccia dei comandi ricevuti (log.txt)
- Interfacce grafiche (ad esempio luce.html)
- File per la ricezione dei comandi da altri dispositivi (switch.php)

Il file di comandi, è nel nostro caso in formato .txt ma è possibile immaginare l'uso di altri tipi di formati, che consentano un accesso al file immediato o meno, a seconda delle necessità.

Questo file come si è detto, contiene un comando dopo che è stato ricevuto dal server. Switch.php è il file che si occupa di registrare i comandi che arrivano al server. Gli applicativi esterni utilizzano questo file per comunicare un comando. Lo script provvede ad associare un identificativo univoco al comando e incollare il comando e il suo identificativo nel file comando.txt, questo file rimane sul server, sarà il modulo PC a controllare periodicamente il file (in modo asincrono), in cerca di un id identificativo il cui comando associato non è stato ancora processato. L'utilità di associare un id ad ogni nuovo comando torna utile quando si invia ripetutamente lo stesso comando, con l'id generato casualmente si potrà avere una gestione migliore dei comandi uguali ripetuti, nel caso si vogliano implementare funzionalità di Toggle. Un miglioramento sotto questo punto di vista è possibile: l'utilizzo di shell SSH o socket evita la macchinosa riscrittura di un file ad ogni nuovo comando. Con i socket è possibile inviare direttamente un segnale al modulo PC ogniqualvolta si riceve un nuovo comando. Questo eviterebbe il controllo periodico in AJAX da parte del modulo domestico di nuovi comandi, però comporterebbe l'aggiunta di un sistema di static DNS al modulo PC e la creazione di una regola di routing nella sezione port forwarding del router, che a quel punto non effettuerebbe più connessioni in uscita, ma si limiterebbe a restare in ascolto per connessioni in ingresso.

Ogni comando che passa dal server viene tracciato assieme all'IP dell'utente chiamante e i dati identificativi circa l'ora e il giorno in cui è stato inviato al server. Questi dati vengono elaborati dal file switch.php che in append provvede a salvarli nel file di log.txt (analogamente al discorso precedente: a seconda delle esigenze è possibile scegliere supporti di memorizzazione differenti, che offrano magari un più alto livello di sicurezza).

Precedentemente si è parlato di interfacce grafiche sul server, e in generale accessibili da browser e potenzialmente, da una moltitudine di device (smartphone compresi). Jarvis server possiede un'interfaccia grafica predefinita, accessibile dal web. La caratteristica di quest'interfaccia è l'impiego di tecnologia ajax per restare aggiornata e in linea con lo stato degli attuatori in casa. Se ad esempio una luce è stata accesa, e rimane accesa in casa, quando si vorrà accedere in seguito all'interfaccia grafica server, essa dovrà rispecchiare lo stato delle cose, e l'interruttore sulla pagina web dovrà essere su ON. Questo si ottiene controllando periodicamente file xml condivisi fra il server e il modulo PC, ma quando si cerca di effettuare richieste in Ajax dette Cross-Domain si incappa nelle *cross-domain policy*, che altro non sono che restrizioni adottate dai browser per evitare a una pagina su un

server A, di richiedere contenuti a una pagina su un server B, al fine di evitare attacchi hacker di tipo XSS (Cross Site Scripting).

4.2.2.1 Cross-Origin Resource Sharing

Come fare allora per effettuare richieste in AJAX verso domini remoti? Quindi per richiedere contenuti a server che non sia quello su cui la pagina d'origine risiede?

Il W3C con una bozza definisce brevemente come il browser e il server devono comunicare per raggiungere destinazioni che non coincidano con il server d'origine (provenienza). Nella pratica vengono dunque utilizzati dei particolari header per consentire o negare una richiesta da un altro dominio. Ad esempio, se uno script, su un sito A:

http://www.sito-a.com/script.js

effettua una richiesta in GET/POST via AJAX ad un sito B:

http://www.sito-b.com/pagina.php

Si ottiene un errore da console javascript. La richiesta fallirebbe, per ragioni di origine dei dati da un dominio differente "sconosciuto" (la cross-origin policy). La soluzione risiede nel settare un apposito header Access-Allow-Origin nello script di destinazione, che nel caso di uno script in php possiamo semplicemente fare usando la funzione header() del linguaggio. Il W3C fornisce in questo senso, diversi header per ogni esigenza:

- *Access-Control-Allow-Origin: http://www.sito-a.com*
- *Access-Control-Allow-Origin: **
- *Access-Control-Allow-Methods: POST, GET*
- *Access-Control-Allow-Headers: NCZ*
- *Access-Control-Max-Age: 1728000*

Quindi modificando la nostra pagina.php presente sul sito-b in questo modo:

```
<?php header("Access-Control-Allow-Methods: POST, GET"); ?>
```

La richiesta ajax da parte di un qualsiasi sito, sia in GET che POST avverrebbe con successo.

4.2.2.2 XSS – Cross Site Scripting

La vulnerabilità di tipo *Cross Site Scripting*, è quella maggiormente diffusa nei siti web. Spesso è trascurata, perché all'ombra di attacchi più pericolosi, come la *SQL*

Injection. Gli attacchi *XSS* e *SQL Injection* sono simili nelle modalità con cui solitamente si inietta il codice malevolo. La differenza sta che la prima inietta il codice per venir seguito sul browser bersaglio, la seconda inietta il codice per essere eseguito sul database. Questo genere di attacchi sono possibili perché gli sviluppatori non filtrano adeguatamente lato server, i caratteri inseribili nei campi di un form html. Se un sito web è vulnerabile ad un attacco XSS, gli utenti che ignari navigheranno su quel sito, eseguiranno sul proprio browser del codice malevolo. Questo codice può essere scritto per compiere una miriade di funzionalità a seconda dei casi. Nel peggiore dei casi, se si tratta di un sito web con una sessione attiva, si potrà far arrivare all'esterno i cookie dell'utente, permettendo all'attaccante di impersonare l'utente loggato sul sito, con tutti i suoi privilegi.

4.2.3 Modulo Smartphone

Nel modulo smartphone, sono state realizzate 2 modalità d'accesso differenti al sistema domotico. La prima è attraverso l'interfaccia web offerta dal server, quindi navigando dal browser del dispositivo mobile all'indirizzo del server. La seconda è attraverso un'app per Android, realizzata e installata sul dispositivo, che fornisce un'interfaccia vocale comunicante col modulo server, al quale invia il comando destinato all'esecuzione remota. L'applicazione è attualmente disponibile solo per i dispositivi con sistema operativo mobile Android. La caratteristica di questo applicativo sta nella sua realizzazione, non è scritta nel linguaggio nativo comune a tutte le applicazioni android ma è scritta utilizzando il linguaggio di scripting Python che a sua volta gira su un livello intermedio. Questo livello è il *SL4A (Scripting Layer for Android)*. Un'applicazione che interpreta diversi linguaggi di scripting sul proprio smartphone Android, velocizzando i tempi di realizzazione di un'app e consentendo di sviluppare prototipi di applicazioni senza dover necessariamente imparare linguaggi ben più complessi usati regolarmente nella stesura di applicazioni Android.

4.2.3.1 SL4A – Scripting Layer for Android

L'SL4A è un progetto open source che mira a portare i linguaggi di scripting nel sistema Android. Esso si installa sul dispositivo come un'applicazione, ed ha accesso a molte delle API che una normale applicazione nativa avrebbe, per poterle successivamente offrire agli script che ci facciamo girare sopra, ma con un'interfaccia semplificata.

Per iniziare a lavorare sull'ambiente di SL4A è necessario abilitare sul proprio smartphone android la possibilità di installare applicazioni anche da sorgenti sconosciute, a quel punto possiamo scaricare sul telefono il file .apk di sl4a (nel momento in cui scrivo giunto alla versione 6: sl4a_r6.apk) e poi il file .apk del

relativo linguaggio di scripting che vogliamo utilizzare per scrivere la nostra prima app (io ho optato per Python, ma ci sono molti altri linguaggi di scripting disponibili: perl, javascript, jruby, lua), tutto è reperibile nella sezione download della repository ufficiale di SL4A (<https://code.google.com/p/android-scripting/downloads/list>). Dopo aver installato SL4A e il linguaggio di scripting sul dispositivo, nella cartella di sl4a troveremo degli utili sorgenti in python di prova, come un classico HelloWorld. Possiamo dunque sbizzarrirci e modificarli a nostro piacimento, provando tutti i metodi che più ci aggradano, usando come referencia le pratiche API reference messe a nostra disposizione all'indirizzo:

<https://code.google.com/p/android-scripting/wiki/TableOfContents?tm=6>

Con centinaia di funzioni per interagire con le finestre e l'interfaccia grafica (UI), la rubrica contatti, il bluetooth, il wireless, i sensori, la fotocamera, il media player, il riconoscimento vocale.

4.2.3.2 Jarvis.py

Installato l'ambiente d'esecuzione, possiamo eseguire gli script. Quello che compone il modulo smartphone di Jarvis è il file "jarvis.py". Questo file utilizza alcuni metodi che interagiscono con le API basilari del sistema Android, tra cui: *ttsSpeak*, *recognizeSpeech*, *makeToast*, *toggleWifiState*, *vibrate*. Rispettivamente per la pronuncia vocale, il riconoscimento vocale, i messaggi di alert, l'attivazione/disattivazione del Wifi e la vibrazione. (Qualora si vogliano utilizzare le API per il riconoscimento vocale, sarà necessario aver installato sul dispositivo un'applicazione che faccia uso di tale tecnologia, come ad esempio "Google Now"), *ttsSpeak* e *recognizeSpeech* permettono di creare un assistente vocale con poche righe di codice. Questi metodi abbinati alle librerie native di python, garantiscono ampi spazi di manovra.

L'ambiente SL4A implementa in tutto e per tutto il linguaggio python, e in quanto tale siamo liberi di importare anche librerie comuni al linguaggio, come le librerie "pyserial" o le "urllib2", per l'interazione con pagine web in remoto.

Jarvis.py ha la possibilità di funzionare come servizio in background oppure come Attività normale, le app android hanno infatti queste due distinte modalità di funzionamento, se l'app funziona come servizio, essa rimane costantemente in background in attesa di un tasto di scelta rapida che "risvegli" le sue funzionalità. Come attività normale invece, sarà in funzione solo se in primo piano sullo schermo (foreground). In sintesi, Jarvis.py rappresenta sia un'interfaccia per l'interazione remota con gli altri moduli, sia un assistente vocale a se stante, in grado di ricevere comandi dall'utente ed eseguirli localmente sul telefono. Per farlo, i comandi ricevuti dall'interfaccia vocale dell'applicativo, devono essere analizzati su due livelli: il primo è quello smartphone, il secondo è quello server. L'analisi è basilare: il

comando ricevuto dallo smartphone viene confrontato con un dizionario interno all'app che segue la stessa struttura logica di quello presente nel Modulo PC, ad ogni riscontro, viene eseguita la funzione locale associata alla voce nel dizionario. Se al contrario, non sono stati ottenuti riscontri, il comando viene inviato al modulo server, che lo terrà in memoria, aspettando il parsing dal modulo PC.

Un esempio di comando locale potrebbe essere “chiama casa”. Lo smartphone associerebbe la parola “chiama” ad una voce del dizionario, preposta appositamente per far partire una chiamata verso il contatto “casa” presente in rubrica.

4.2.4 Modulo Microcontroller

Il modulo microcontroller, è quello a cui tocca il lavoro sporco. Esso è il modulo che più di tutti lavora a basso livello. Ha il compito di interagire direttamente con l'hardware, con i prototipi elettronici realizzati appositamente per questo progetto. Riceve e trasmette comandi direttamente col modulo PC. Avvalendosi di una comunicazione seriale e radio a 433mhz.

La progettazione software di questo modulo è strettamente legata all'hardware che pilota. Il codice è caricato nel microcontrollore della board di prototipazione usata, Arduino. Sia il codice che l'hardware di questo modulo si vedranno nel dettaglio nella sezione Progettazione Hardware, poiché l'uno dipende dall'altro e non possono scindersi.

4.3 Progettazione Hardware

La progettazione Hardware di jarvis comprende la quasi totalità di quello che è il modulo Microcontroller, e si riassume in esso. Il modulo è stato realizzato usando la board open source Arduino. Il cui bootloader, è programmabile in un linguaggio simil C. Gli altri componenti sono prototipi elettronici realizzati appositamente per essere usati con questo progetto.

4.3.1 Cos'è Arduino?

“Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments.” – Arduino.cc

Arduino, come recita il sito ufficiale, è una piattaforma di prototipazione open source, facile da usare. Realizzata per designer, artisti, hobbisti e chiunque interessato a creare ambienti e oggetti interattivi (Figura 4.4). Può essere paragonabile a un piccolo computer, operante a 16Mhz, e con una piccola memoria interna. Moltissimi elettrodomestici utilizzano microcontrollori per svolgere i loro compiti base.



Figura 4.4 – Board Arduino Uno

Arduino interagisce con l'ambiente ricevendo input dai sensori. E' capace di controllare luci, motori e attuatori di ogni genere. E' facilmente programmabile, con un linguaggio simile al C, da un'ambiente di sviluppo (IDE) progettato appositamente per arduino. Dall'IDE si scrive codice per il microcontrollore di arduino, lo si testa e lo si carica sulla board (upload). Arduino è un marchio registrato, la board tuttavia è open source, sono disponibili online gli schemi elettrici (Figura 4.5), è possibile comprarla assemblata o costruirselo da soli, con pochi pezzi di elettronica.

I modelli di Arduino disponibili sono ad oggi più di 10, ognuno con le proprie caratteristiche e particolari specifiche che li rendono più o meno adatti a varie tipologie di progetto. In Jarvis la prototipazione avviene con una board Arduino classica, chiamata "Arduino Uno", il cui schema del microcontrollore

ATMEGA328P/ATMEGA168P è visibile nella figura 4.6 e con un'altra board chiamata “Arduino Nano”, più piccola e compatta della versione “Uno”.

Arduino S3v3 Revision 2

Released under the Creative Commons Attribution Share-Alike 2.5 License

<http://creativecommons.org/licenses/by-sa/2.5/>

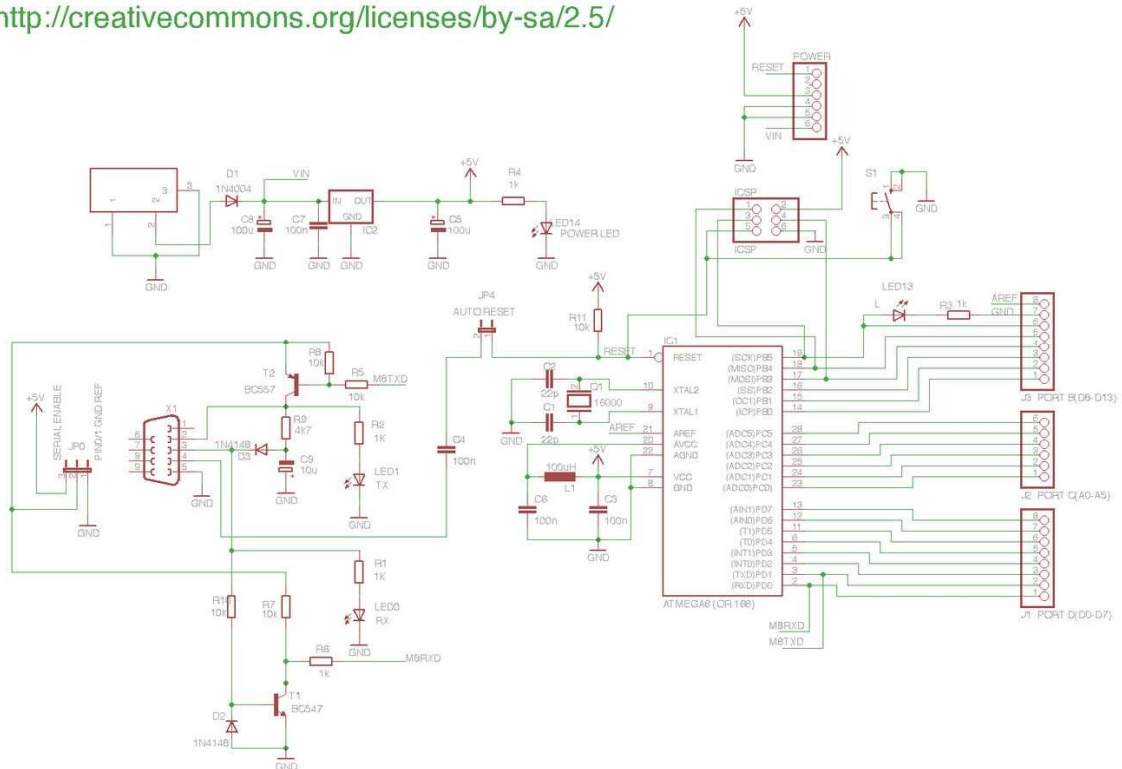


Figura 4.5 – Schema elettrico di Arduino.

Atmega168 Pin Mapping

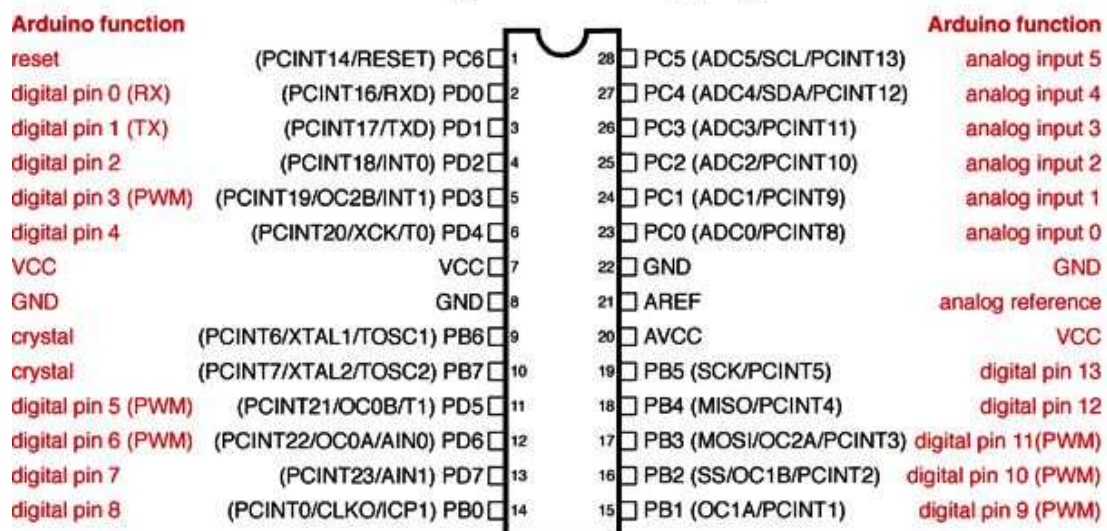


Figura 4.6 – Schema Microcontrollore Atmega168p/328p Arduino.

In figura 4.6 si notano sulla destra i 6 pin analogici disponibili sulla board, mentre a sinistra i pin digitali. I pin digitali, possono assumere lo stato digitale 1 o 0, che corrispondono ad una corrente passante di 5 Volt (HIGH) o 0 Volt (LOW), con un amperaggio di massimo 40mA. I pin analogici supportano correnti modulate, percepiscono le variazioni di tensione fra i 0 e 5 V.

Arduino può essere alimentato tramite il connettore USB, il pin VIN presente sulla board, o attraverso la spina per alimentazione esterna. Le prime due soluzioni necessitano di una corrente né più alta né più bassa di 5V per permettere alla board di funzionare. La spina di alimentazione esterna invece, supporta dai 6 ai 20V, ed è direttamente collegata al regolatore di tensione interno. La versione “Nano” di arduino invece, non possiede una spina di regolazione esterna, è il pin VIN ad essere regolato dal regolatore di tensione.

La versione Nano di Arduino (Figura 4.7) ha le stesse funzioni di una board classica arduino, ma è pensata per un uso definitivo nei progetti, dovuto soprattutto alle sue dimensioni e il costo ridotto. E’ utilizzata in Jarvis per i “satelliti”, cioè le parti mobili, gli attuatori, disposti in giro per casa, comunicanti con il modulo controllore connesso al PC, via radio (Figura 4.8).

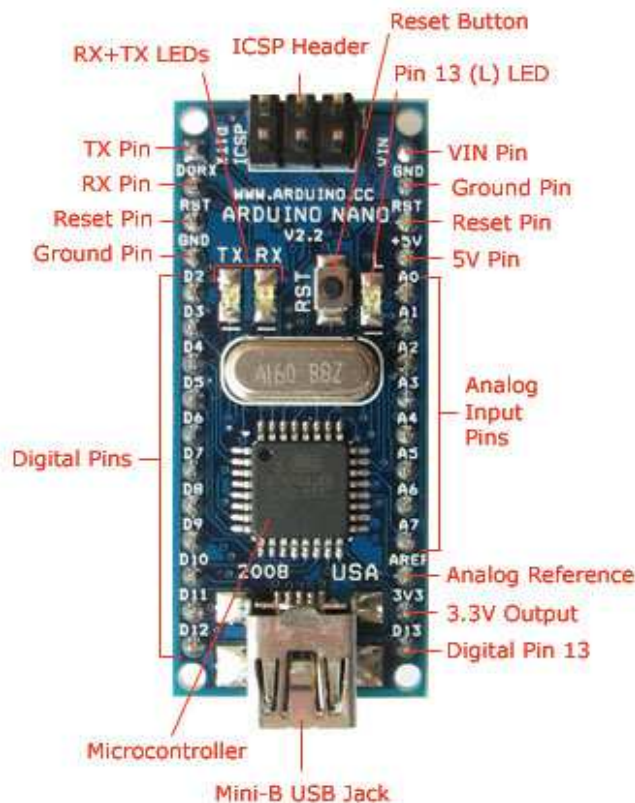


Figura 4.7 – Visuale superiore di Arduino Nano



Figura 4.8 – Moduli radio Trasmettitore e Ricevitore 433mhz.

Passiamo ad analizzare il reparto hardware dei “satelliti”, gli attuatori che insieme permettono di controllare gli apparecchi domestici. Partendo da un piccolo prototipo elettronico realizzato per pilotare elettrodomestici e luci che ho chiamato “prolunga relè”.

4.3.2 Prolunga relè – prototipo v 2.0

Questo primo prototipo consta di 5 componenti hardware e di un software di gestione per il microcontrollore.

I componenti usati sono:

- Una prolunga elettrica, con un’uscita femmina.
- Un microcontrollore per pilotare l’attuatore (Arduino).
- Un attuatore.
- Alimentatore DC 12V 1A.
- Moduli radio ricevente e trasmittente.

L’attuatore utilizzato è un relè di tipo SSR (Solid State Relay), visibile nell’immagine 4.9, ha la caratteristica di non avere parti meccaniche. Il relè è un dispositivo interruttore in grado di commutare il proprio stato e permettere o negare il fluire della corrente da due estremità. Commutare il relè è possibile con basse correnti, come quelle emanate da un microcontrollore.



Figura 4.9 – Relè SSR 48-480V/AC.

I componenti per la comunicazione, sono i due moduli radio mostrati in precedenza (Figura 4.8). La parte della trasmittente viene collegata direttamente al microcontrollore che a sua volta è collegato in seriale al modulo PC di Jarvis (Figura 5.0).

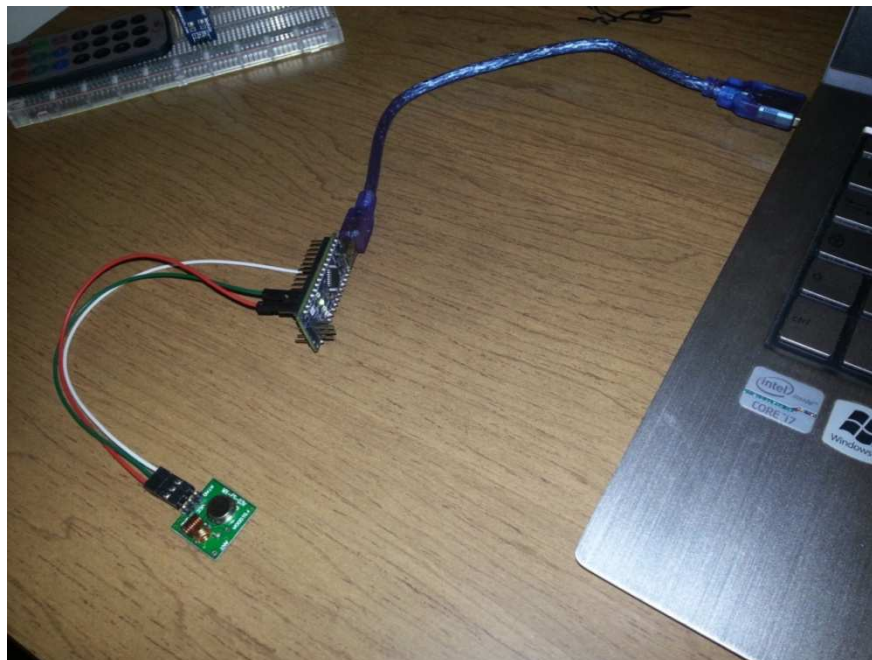


Figura 5.0 – Trasmittente Radio connessa tramite arduino al PC

Uno schema dei moduli trasmissivi è visibile in figura 5.1

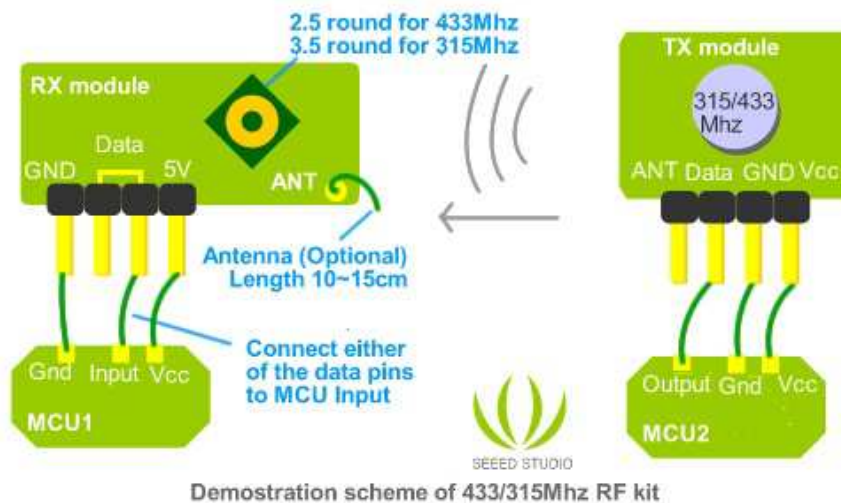


Figura 5.1 – Schema moduli trasmissivi

La ricevente (RX) fa invece parte del circuito che pilota il relè allo stato solido. Per ottenere un lavoro pulito, l'arduino è stato inscatolato e alimentato attraverso un alimentatore da parete. Nella scatola è presente anche la ricevente, il relè e l'alimentatore privato del case protettivo. Il collegamento è visibile nella figura 5.2.

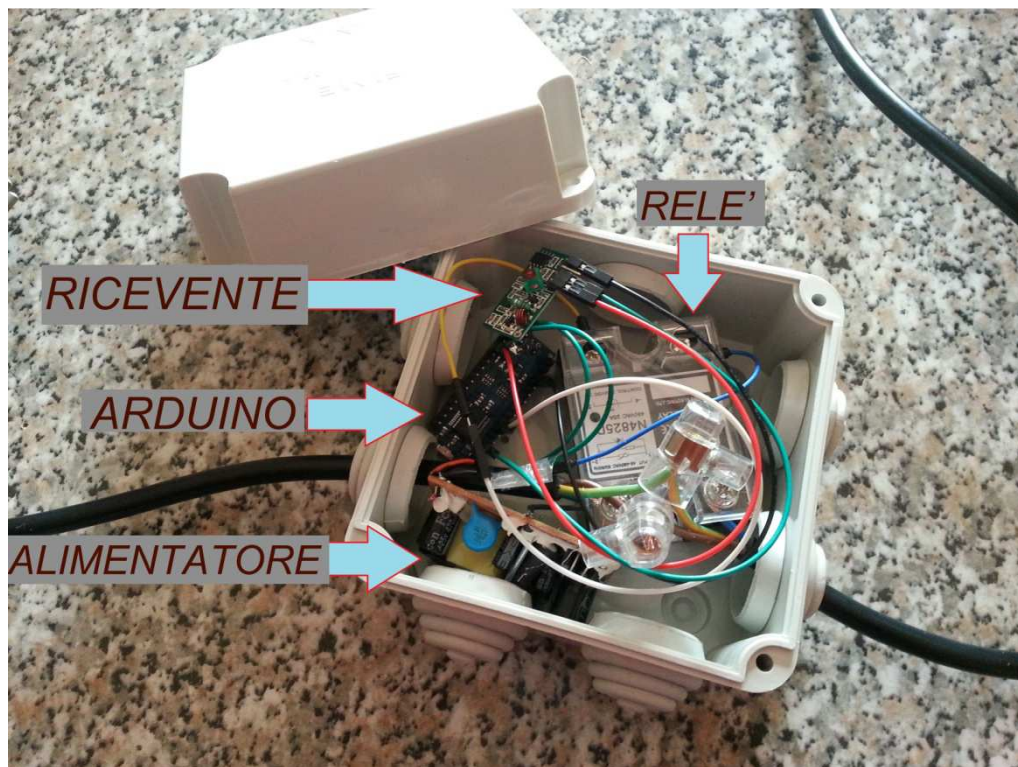


Figura 5.2 – Prototipo “prolunga relè” inscatolato

Qualsiasi elettrodomestico collegato a questa speciale prolunga, è interfacciato al modulo PC e quindi all'intero sistema jarvis. La logica di funzionamento è molto semplice e riassunta qui di seguito:

Le librerie *VirtualWire* che permettono al microcontrollore di interfacciarsi con il ricevitore radio. La ricevente radio è collegata al microcontrollore, collegato al relè che agisce sulla presa in cui si collega un elettrodomestico. Quando riceve un comando dalla trasmittente (che rappresenta il modulo microcontroller di jarvis, collegato al modulo PC), attiva il pin digitale a cui è connesso il relè, innescando la commutazione e permettendo alla corrente di fluire nella prolunga. Il codice del microcontrollore arduino che permette tutto questo è visibile in figura 5.3.

```
// Receiver - Al pin 8 è connesso il relè.

#include <VirtualWire.h>
void setup()
{
    //Serial.begin(9600); // Debugging only
    pinMode(8, OUTPUT);

    // Initialise the IO and ISR
    vw_set_ptt_inverted(true); // Required for DR3100
    vw_setup(2000);           // Bits per sec
    vw_rx_start();            // Start the receiver PLL running
}

void loop()
{
    uint8_t buf[VW_MAX_MESSAGE_LEN];
    uint8_t buflen = VW_MAX_MESSAGE_LEN;
    if (vw_get_message(buf, &buflen)) // Non-blocking
    {
        int i;
        String x = ""; //conterrà la stringa ricevuta

        for (i = 0; i < buflen; i++)
        {
            //Serial.print(buf[i], HEX);
            //Serial.print(" ");
            //Serial.print((char) buf[i]); //casting a carattere del codice ascii
            x += (char)buf[i];
        }

        //Serial.println("");
        if (x == "on"){ // Se il msg ricevuto è "on"
            //Serial.println(x);
            digitalWrite(8, HIGH);
        }else if(x == "off"){
            digitalWrite(8, LOW);
        }
    }
}
```

Figura 5.3 – Codice microcontrollore arduino lato RX.

Il codice di figura 5.3 mostra nella funzione *setup()* alcune istruzioni per l'inizializzazione delle librerie per la comunicazione radio, la parte di *loop()* invece, evidenzia l'attesa continua di un comando, che se uguale a "on" o "off" permette al relè di commutare.

Il ricevitore radio può essere migliorato e nel capitolo conclusivo si mostrano i possibili miglioramenti all'aspetto "rete e comunicazione" fra i dispositivi, adottando altri tipi di trasmissione radio. Per ora, un miglioramento sostanziale lo si ottiene installando un'antenna esterna. L'antenna interna ai moduli radio visti in precedenza, garantiscono una trasmissione non superiore ai 5 m. senza ostacoli. Il che per una casa di media grandezza non è affatto l'ideale. L'intallazione di un'antenna esterna per questi moduli a basso costo, è quindi d'obbligo.

4.3.3 Qual è la lunghezza di un'antenna ideale?

Le antenne, sono obbligatorie per garantire la diffusione del segnale in maniera efficace. Costruire un'antenna per circuiti elettronici è possibile anche con materiali di scarto, come antenne si possono utilizzare fili di rame, riciclati da cavi telefonici o di rete (smaltati o meno, poco cambia, se smaltati avranno soltanto una maggiore resistenza all'ossidazione, che non compromette né incide sul segnale). La domanda che persiste è, quanto devono essere lunghe le antenne?

Esiste un modo per calcolare la lunghezza ottimale: Quando si realizza un'antenna per un dispositivo RF il massimo segnale lo si ottiene con un'antenna lunga esattamente quanto la lunghezza d'onda (λ) da trasmettere/ricevere. La formula per calcolare tale lunghezza d'onda è la seguente:

$$\lambda = v / f$$

Dove v è la velocità di propagazione nel mezzo di trasmissione, che nell'etere è pari alla velocità della luce, 300.000 Km/s, espressa in metri, ed f è la lunghezza d'onda, espressa nell'unità di misura Hertz.

Quindi per 433 MHz si ha:

$$300.000.000 / 433.000.000 = 0,6928 \text{ m} \rightarrow 69,28 \text{ cm}$$

Spesso, non è praticabile fare l'antenna di pari lunghezza di λ e si ricorre alle frazioni. Si usano λ mezzi, λ quarti e così via. Un quarto di λ nel nostro caso è:

$$69,28 / 4 = 17,32 \text{ cm.}$$

Il risultato ottenuto dimostra come per una frequenza di 433Mhz sia necessario un'antenna di almeno 17.32 cm, che è un quarto d'onda della lunghezza ottimale.

Il prototipo finale con tanto di antenne saldate, è visibile in figura 5.4 (a sinistra il modulo Microcontroller di Jarvis, connesso al PC tramite USB, a destra uno degli attuatori “satellite” adibito al controllo di elettrodomestici).



Figura 5.4 – Prototipo finale con antenne saldate

4.3.4 Prototipo Ambientazione suggestiva con striscia di led RGB

Un aspetto curioso nell'illuminazione degli ambienti, è legato al comfort che si crea, con illuminazioni adatte e consone alla stanza in cui ci troviamo. Luci soffuse sono adatte per la camera da letto, o quando guardiamo un film in soggiorno, luci chiare e forti sono ideali quando leggiamo un libro. Con le normali lampadine ottenere effetti del genere è praticamente impossibile. Al contrario, con le nuove lampade a led RGB, è possibile ottenere varie tonalità di colori, e aumentare e diminuire l'insensità della luce a piacimento. In Jarvis un primo prototipo d'illuminazione RGB avviene con una striscia di LED (Figura 5.4.1). Ogni LED è del tipo SMD5050, saldato su una striscia conduttrice, con un driver LPD8806 ogni 2 led. Il Driver è in grado di

consentire il controllo di ogni singolo led indipendentemente dagli altri, utilizzando un microcontrollore. Questo garantisce giochi di luci che non si avrebbero con normali strisce di led. In un metro di striscia sono presenti circa 32 led.

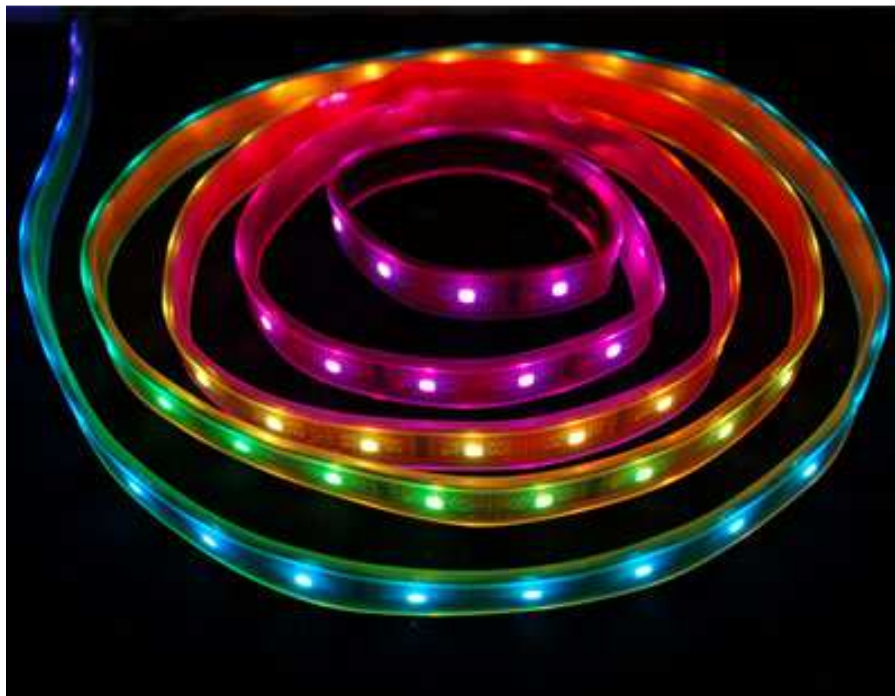


Figura 5.4.1 – Striscia di Led RGB LPD8806 pilotabile

In figura 5.4.2 un esempio di ambiente illuminato da strisce di led RGB.



Figura 5.4.2 – Ambiente illuminato da RGB Led Strip.

Il microcontrollore usato, abbiamo visto, è basato su Arduino. Il collegamento con la striscia di led avviene in maniera abbastanza rapida. L'unica accortezza da prestare durante la realizzazione del prototipo è stata l'alimentazione. La striscia viene alimentata attraverso una corrente di 5V. Ma l'Amperaggio necessario per alimentare un metro di striscia dev'essere almeno di 2 Ampere. Il collegamento con Arduino deve quindi essere effettuato a "ponte" con un alimentatore domestico in grado di fornire un voltaggio non superiore ai 5V e con un Amperaggio di almeno 2 A, (Figura 5.4.3).

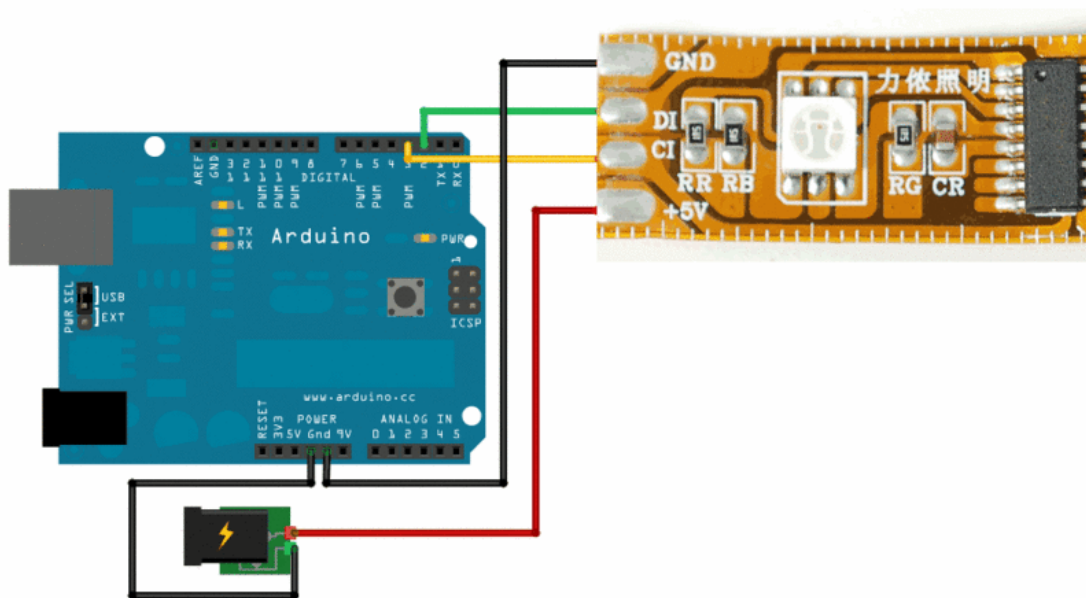


Figura 5.4.3 – Collegamento di Arduino con striscia RGB LPD8806

4.3.5 Prototipo Codifica e Decodifica dei segnali IR (Infrarossi).

Il segnale infrarosso, è forse il metodo di comunicazione remoto più convenzionale nelle nostre abitazioni. Ogni telecomando che conosciamo è quasi per definizione a tecnologia infrarossa. Questo vuol dire che se il telecomando ha un emettitore infrarosso, quasi sicuramente dall'altra parte ci sarà un dispositivo con ricevitore infrarosso. Lo spettro dell'infrarosso ha una lunghezza d'onda fra i $700nm - 1\text{ mm}$, mentre l'occhio umano percepisce lunghezze d'onda solo fra i $380nm - 700nm$, perciò non è visibile all'occhio umano (Figura 5.5).

Light comparison ^[4]			
Name	Wavelength	Frequency (Hz)	Photon Energy (eV)
Gamma ray	less than 0.01 nm	more than 30 EHz	124 keV – 300+ GeV
X-Ray	0.01 nm – 10 nm	30 EHz – 30 PHz	124 eV – 124 keV
Ultraviolet	10 nm – 380 nm	30 PHz – 790 THz	3.3 eV – 124 eV
Visible	380 nm – 700 nm	790 THz – 430 THz	1.7 eV – 3.3 eV
Infrared	700 nm – 1 mm	430 THz – 300 GHz	1.24 meV – 1.7 eV
Microwave	1 mm – 1 meter	300 GHz – 300 MHz	1.24 μ eV – 1.24 meV
Radio	1 mm – 100,000 km	300 GHz – 3 Hz	12.4 feV – 1.24 meV

Figura 5.5 – Lunghezze d’onda e frequenze delle onde elettromagnetiche

Il prototipo di jarvis, che fa uso dell’infrarosso, è chiamato “Infrared remote py-control”, la dicitura “py”, anticipa che per funzionare è necessario eseguire il codice su un interprete python. Questo satellite o se preferiamo attuatore esterno, ha due modalità di funzionamento:

La prima permette di emulare un circuito chiuso. Il componente chiave di questa modalità è il ricevitore infrarosso collegato al modulo Microcontroller di Jarvis. Può essere considerato alla stregua di un’interfaccia per interagire col sistema jarvis, attraverso un semplice telecomando infrarosso, vedi quello della TV o del decoder.

L’altra modalità vede come unico componente chiave utilizzato per la realizzazione, l’emettitore infrarosso. In questa modalità d’uso, al contrario dell’altra, la comunicazione non avviene col sistema Jarvis, bensì con gli elettrodomestici dotati di ricevitore infrarosso (TV, Decoder, Condizionatori e così via). L’emettitore infrarosso serve dunque per inviare comandi in uscita dal sistema jarvis, il ricevitore infrarosso, per ricevere comandi in entrata.

I componenti hardware utilizzati (Figura 5.6.1 – Figura 5.6.2) sono:

- Ricevitore IR (sulla sinistra nella Figura 5.6.1 e 5.6.2)
- Emettitore IR (sulla destra nella Figura 5.6.1 e 5.6.2)
- Telecomando IR

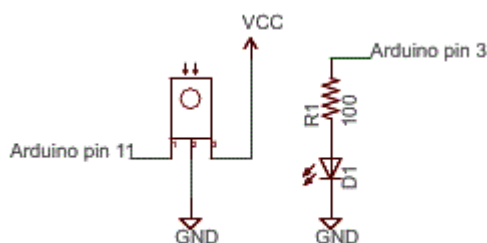


Figura 5.6.1:

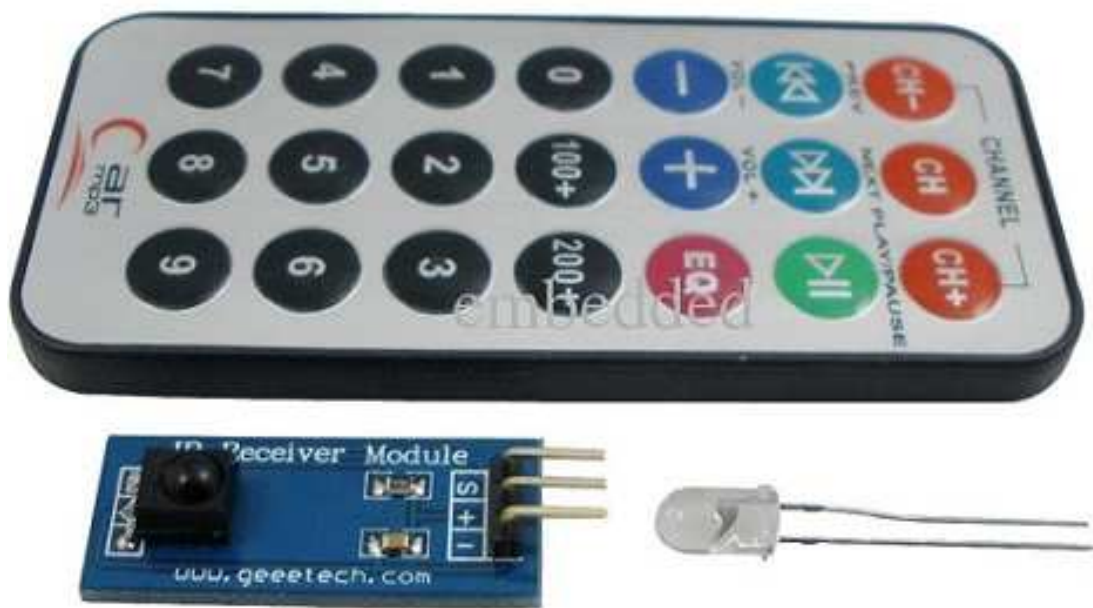


Figura 5.6.2 – Componenti per comunicazione infrarossa

La modalità di ricezione permette di comunicare con Jarvis attraverso un telecomando infrarossi. Il codice usato da Arduino, per ricevere i comandi infrarossi e inviarli in seriale al modulo PC è riportato in figura 5.7.

```
#include <IRremote.h> // per elaborare i segnali infrarossi

int RECV_PIN = 7; //pin a cui è connesso il ricevitore infrarosso
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup()
{
  Serial.begin(9600); //avviamo la comunicazione seriale col Modulo PC di Jarvis
  irrecv.enableIRIn(); // Start del ricevitore IR
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX); //inviame il segnale letto in seriale
    irrecv.resume(); // Aspettiamo un altro valore
  }
}
```

Figura 5.7 – Codice microcontrollore per ricezione infrarossa

Dal codice si nota subito l'utilizzo delle librerie *IRremote.h*, necessarie per l'elaborazione dei segnali infrarossi ricevuti. Il pin 7 è quello dove è collegato il ricevitore infrarosso, l'istruzione *Serial.begin(9600)*, permette all'arduino di avviare una comunicazione seriale con il PC ad una frequenza di 9600 baud, per poter inviare messaggi in uscita. L'istruzione d'invio è proprio: *Serial.println(results.value, HEX)*;

Il messaggio contenuto in *results.value* sarà ricevuto dal modulo PC ed elaborato lato Jarvis, per associarvi una funzione specifica.

Di rimpetto, Jarvis implementa l'altra modalità di funzionamento, quella dell'emissione di un segnale.

La modalità d'emissione di un segnale IR richiede un'elaborazione un po' particolare. Una caratteristica delle onde elettromagnetiche infrarosse, è che sono emesse da qualsiasi corpo caldo, persino il corpo umano emette onde nello spettro infrarosso. Di conseguenza, un ricevitore IR, deve poter stabilire se ciò che riceve è dettato da una sorgente artificiale o naturale, ignorando tutti i disturbi non desiderati. Per elaborare un segnale utilizzando la luce infrarossa, è necessario modularlo. La modulazione è effettuata in Jarvis, dalla libreria *IRremote.h* installata sull'Arduino e dal diodo emettitore, attraverso specifici "balzi" di voltaggio. I ricevitori sono programmati per riconoscere queste modulazioni, associando ad ogni modulazione un evento o un valore che viene elaborato giunto a destinazione finale.

Un esempio di codice per l'invio di un segnale è visibile in Figura 5.8.

```
#include <IRremote.h>

IRsend irsend;

void setup() {
}

void loop() {
  irsend.sendSony(0x68B92, 20);
  delay(100);
  irsend.sendSony(0x68B92, 20);
  delay(100);
  irsend.sendSony(0x68B92, 20);
  delay(300000);
}
```

Figura 5.8 – IRremote transmitting.

I protocolli per l'invio di un segnale IR sono molteplici. Nella figura 5.8, si cerca di inviare il segnale "0x68B92" codificato in 20 bit, attraverso il protocollo di SONY, che richiede, inoltre, debba essere ripetuto 3 volte l'invio per essere interpretato bene lato ricezione. Altri protocolli, sono quelli definiti dai maggiori produttori mondiali d'apparecchi a comunicazione infrarossa, tra cui spuntano SONY, NEC e i formati RC5 e RC6. La libreria *IRremote* garantisce dei metodi, anche per questi formati (Figura 5.9).

```

IRsend irsend;
    Create the transmit object. A fixed pin number is always used, depending on which timer the library is utilizing.

irsend.sendNEC(IRcode, numBits);
    Send a code in NEC format.

irsend.sendSony(IRcode, numBits);
    Send a code in Sony format.

irsend.sendRC5(IRcode, numBits);
    Send a code in RC5 format.

irsend.sendRC6(IRcode, numBits);
    Send a code in RC6

```

Figura 5.9 – Metodi di IRremote

Utilizzando un ricevitore e un emettitore per la cattura e decodifica di un segnale e poi la codifica ed emissione dello stesso segnale, è possibile realizzare telecomandi universali, in grado di “imparare” e pilotare qualsiasi tipo di apparecchio che faccia uso della tecnologia a infrarossi per la comunicazione a distanza. Questo è quello che Jarvis impiega per interfacciarsi ad apparecchi che comunicano via infrarosso.

Per maggiori approfondimenti è possibile consultare il sito ufficiale delle librerie IRremote e il progetto di telecomando universale, di Ken Shirriff:

http://www.pjrc.com/teensy/td_libs_IRremote.html

<http://www.righto.com/2009/09/arduino-universal-remote-record-and.html>

4.3.6 Uno sguardo ai sensori

Un altro sensore che funziona nello spettro dell’infrarosso è il sensore piroelettrico PIR (Figura 6.0). Esso si basa su quanto detto in precedenza: ogni corpo caldo emette onde infrarosse, compreso il corpo umano.



Figura 6.0 – Sensore piroelettrico di movimento

Questo sensore, entra in scena nell'ambito della sicurezza, rilevando infatti le radiazioni infrarosse prodotte dal corpo umano, è in grado di stabilire entro un certo raggio d'azione, la presenza di "attività umana". E' un sensore compatto, a basso costo e facile da usare grazie all'output digitale. E' adottato in Jarvis, nei molteplici satelliti che sono adibiti al rilevamento di presenza. Insieme a questo, in Jarvis, e più in particolare nell'ambito domotico, si usano altre tipologie di sensori, come quelli elencati a seguire:

- Fotoresistenze (Figura 6.1)
- Sensori di gas, metano, butano (Figura 6.2)
- Sensori acustici (Figura 6.3)
- Sensori termici di temperatura (Figura 6.4)
- Sensori per il calcolo di distanze (Figura 6.5)
- Sensori di prossimità immediata (Figura 6.6)
- Sensori magnetici (hall sensor – Figura 6.7)
- Sensori a Tecnologia RFID (Figura 6.8)

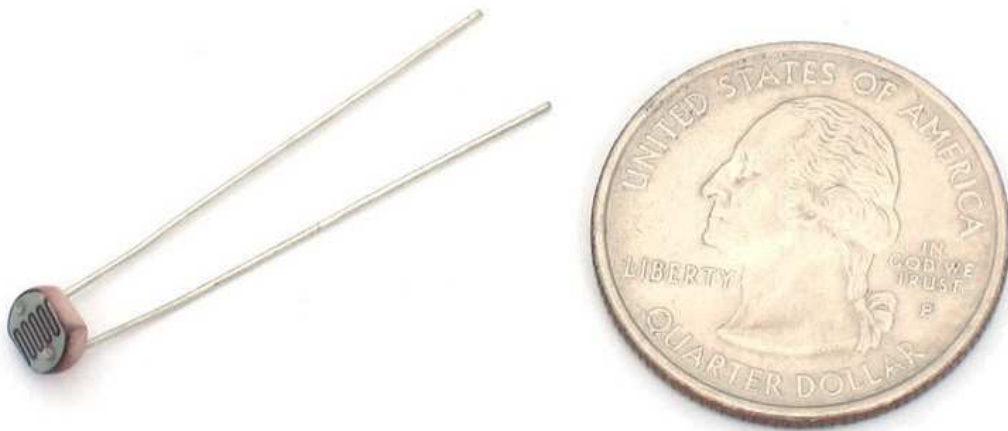


Figura 6.1 – Fotoresistenza

La fotoresistenza o fotocellula, è fondamentalmente una resistenza, che modifica la sua resistività (Ohm) a seconda di quanta luce giunge sulla sommità. Sono sensori estremamente economici, di varie dimensioni ma non molto accurati. Questo tipo di sensore, può in modo grossolano rilevare se è presente della luce, e in che quantità. Ma se il progetto richiede calcoli e valori precisi, l'uso di questo sensore è sconsigliato. In Jarvis è sufficiente per garantire il rilevamento di luce nelle stanze.



Figura 6.2 – Sensore di Gas, metano, butano.

La tipologia di sensori di gas usati in Jarvis è visibile nella figura 6.2. Si tratta di un sensore digitale. Il potenziometro presente sulla board permette di calibrare la sensibilità del sensore. Quando viene rilevata una fuoriuscita di Gas, il pin S emette un segnale.



Figura 6.3 – Sensore acustico: microfono.

I sensori acustici permettono di rilevare i livelli di rumore entro un certo raggio d'azione. Ogni microfono presenta sulla board dei circuiti per l'amplificazione del suono. L'ascolto avviene registrando il livello di rumore di fondo, e adottando la soglia massima su tale rilevazione iniziale. Questo permette una migliore isolazione dei rumori di fondo per concentrarsi su altri rumori generati. L'utilità di questi sensori in ambito domestico, spaziano dal settore Sicurezza, al settore Usabilità. E' possibile infatti creare dei circuiti detti "clapper", per azionare dispositivi elettronici con il solo battito delle mani.

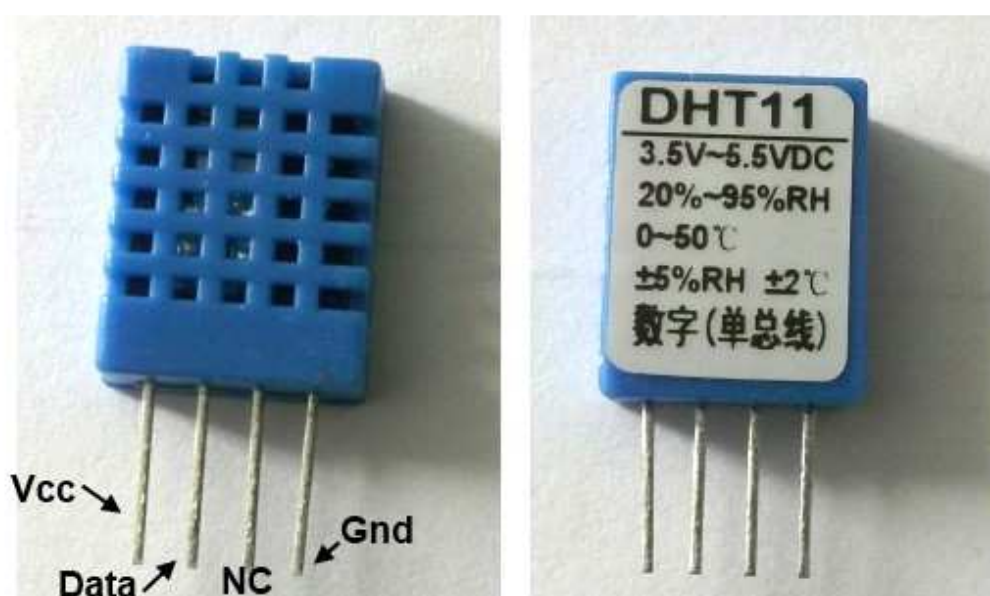


Figura 6.4 – Sensori di temperatura

I sensori di temperatura permettono di effettuare rilevazioni circa la temperatura ambientale. Si comportano come normali termometri. Per un progetto domotico, monitorare la temperatura è necessario se abbiamo a che fare con attuatori adibiti al controllo del riscaldamento centralizzato. Le informazioni sulla temperatura degli ambienti in casa, sono necessarie e imprescindibili da ogni altra cosa.

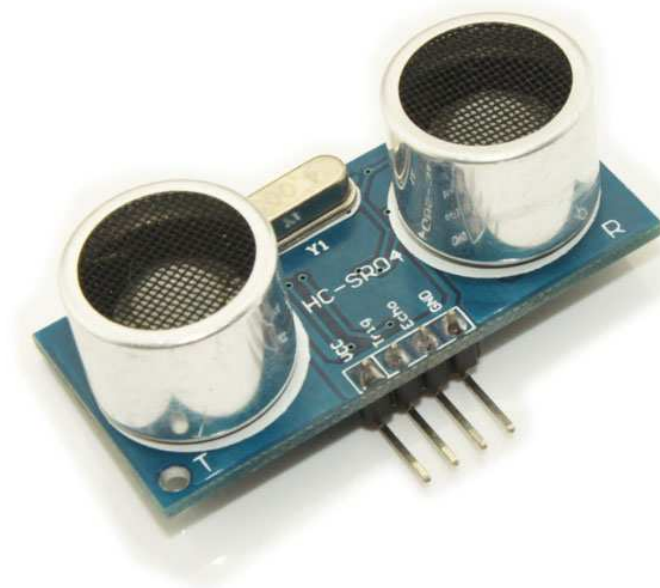


Figura 6.5 – Sensore ultrasuoni per il calcolo di distanze

Calcolare la distanza, può tornare utili per particolari applicazioni o funzionalità di casa. Il sensore che permette il calcolo della distanza di un oggetto è mostrato in figura 6.5, permette il calcolo di distanza fino a 5 m. Funziona sfruttando gli ultrasuoni: un suono viene emesso, rimbalza sull'oggetto e viene ri-captato dal sensore (Figura 6.5.1). Calcolando il tempo trascorso dall'emissione, fino al ricevimento, e conoscendo la velocità di propagazione del suono nell'etere. Si può calcolare la distanza:

$$V = s / t \rightarrow s = V * t$$



Figura 6.5.1



Figura 6.6 – Sensore di prossimità immediata

Il sensore di prossimità immediata, ha un funzionamento analogo al sensore per il calcolo di distanze, è dotato di un emettitore e un ricevitore. Funziona con tecnologia infrarossa e non ad ultrasuoni. Con questo sensore compatto è possibile rilevare la presenza di oggetti molto adiacenti ai due led infrarossi. E' usato in scenari in cui si necessita sapere se un determinato oggetto è in posizione o meno (quindi adiacente al sensore o meno). Si può immaginare un frigorifero che all'interno faccia uso di questi sensori, per captare se un determinato prodotto è esaurito o meno, certo, se si ha il vizio di lasciare le scatole dei prodotti esauriti, all'interno del frigo, la cosa non tornerebbe utile.



Figura 6.7

I sensori magnetici, tornano di grandi utilità nel settore della Sicurezza. Questi sensori rilevano la presenza di campi magnetici in prossimità e permettono di variare il proprio stato in base alla presenza o assenza di questi campi. L'uso più comune che ne viene fatto, è nei sistemi di allarme per porte e finestre (Figura 6.7.1). In figura da un lato c'è un piccolo magnete, dall'altro un sensore magnetico. Quando i componenti si allontanano, il sensore rileva l'assenza del campo magnetico, facendo scattare l'allarme e segnalando una possibile effrazione.

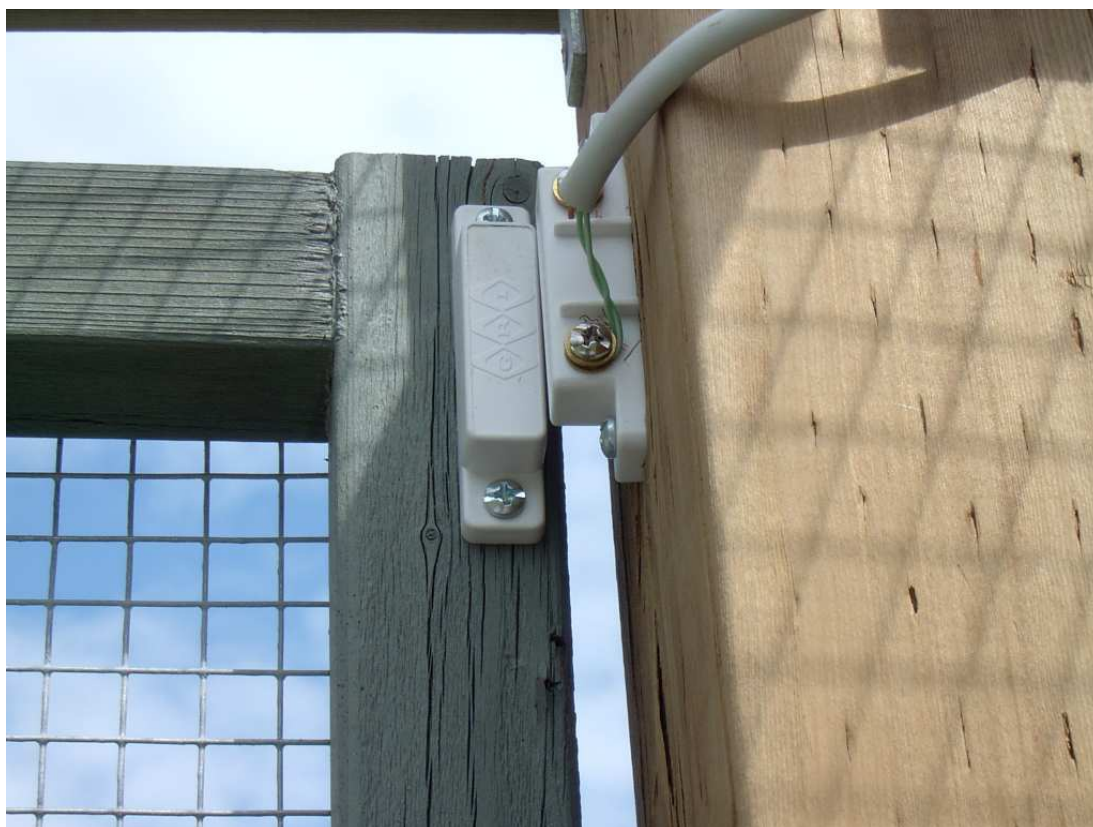


Figura 6.7.1 – Un sensore magnetico all'aperto.



Figura 6.8 – Antenna RFID

La tecnologia RFID (Radio Frequency Identification), è una tecnologia senza fili per l'identificazione di oggetti, persone o animali, attraverso dei *Tags* detti anche *Trasponder*, che passando davanti un'antenna vengono riconosciuti. Può essere considerata come l'evoluzione del codice a barre. La tecnologia RFID è usata in ambito sportivo, domestico, commerciale, veterinario. Si pensi nel campo automobilistico, per registrare l'esatto tempo in cui una macchina taglia il traguardo, oppure nell'ambito commerciale, con questi dispositivi anti taccheggio attaccati ai capi d'abbigliamento, che altro non sono che Trasponder RFID.

In ambito domestico invece è usata per il controllo degli accessi in una struttura. Parliamo di chiavi elettroniche, ogni tag possiede un ID, che può essere autorizzato o meno ad aprire una determinata porta elettronica. Infine nel settore veterinario, è prassi comune impiantare dei piccoli tag RFID sottopelle negli animali, per poterli riconoscere e schedare (Figura 6.8.1). Data la necessità di rimpicciolire i tags per renderli facilmente impiantabili sottopelle, ogni trasponder emette il segnale di riconoscimento se ricaricato per induzione dall'antenna del lettore, questo evita l'utilizzo di batterie nei trasponder, ed è alla base della tecnologia RFID.

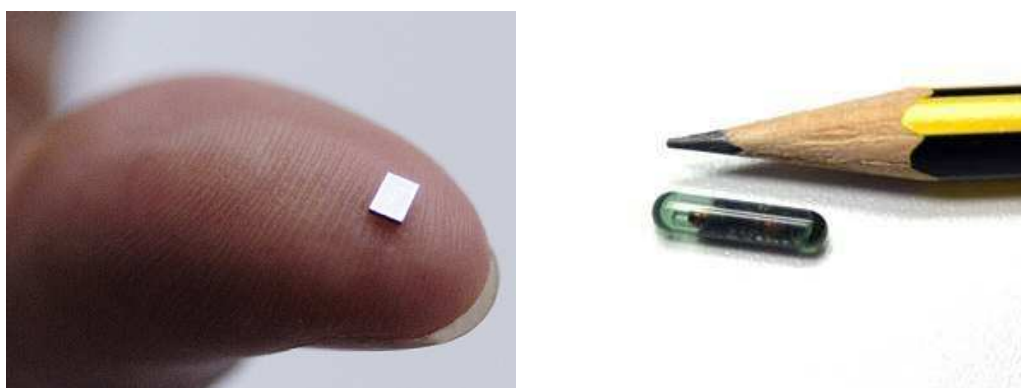


Figura 6.8.1 – Trasponder RFID impiantabili sotto pelle

Le varietà di trasponder sul mercato è disarmante. Esistono tags sottili quanto un foglio di carta, che si applicano su una vastità di oggetti come fossero adesivi (Figura 6.8.2).



Figura 6.8.2 – Trasponder RFID adesivi

Altri tag possono essere sottoforma di portachiave, e danno più l'idea di una chiave elettronica in se e per se (Figura 6.8.3).



Figura 6.8.3 – Trasponder RFID a portachiave

Trasponder di queste dimensioni come detto in precedenza, non richiedono l'utilizzo di batterie per inviare il segnale di riconoscimento, in quanto caricate per induzione dall'antenna del lettore stesso, sono detti *Tag Passivi*. Nei tag passivi la distanza fra il tag e l'antenna, non può essere molto grande. Per riconoscimenti a più lunghe distanze, si usano altri tipi di tag, chiamati *Tag Attivi*. I tag attivi, pur rinunciando a delle dimensioni ultra compatte, sono dotati di una batteria interna e possono essere riconosciuti da antenne poste ad una distanza molto più elevata.

La frequenza di comunicazione fra antenna e trasponder può variare, la più popolare è sui *125 KHz*. Ma con l'evoluzione della tecnologia, si è pensato di aumentare la frequenza, a scapito della distanza, e a beneficio della velocità di trasferimento. Si pensi alla più recente evoluzione di questa tecnologia, chiamata *NFC (Near Field Communication)*, presente su molti smartphone di fascia alta, funzionante ad una frequenza di *13,56 Mhz*. L'NFC trova impiego nelle transazioni via smartphone o nel trasferimento file a distanza ravvicinata. I molti circuiti per lo scambio monetario, come VISA o MasterCard, stanno introducendo nei loro terminali di pagamento, nuovi apparecchi che supportino il pagamento via smartphone con tecnologia NFC.

L'utilizzo che viene fatto in Jarvis di tale tecnologia, è relativo al controllo d'accesso sulla porta principale di casa. Montando una serratura elettronica (Figura 6.8.4), pilotata da un microcontrollore direttamente connesso all'antenna RFID, è possibile realizzare un sistema di controllo degli accessi basato su riconoscimento tramite trasponder RFID.



Figura 6.8.4 – Serratura elettronica con lettore RFID

5. SISTEMI ESISTENTI IN COMMERCIO

Molti sistemi domotici in commercio, non garantiscono standard di novità tali da essere presi realmente in considerazione dal mercato. Ciò che vale la pena guardare, nasce da piccoli sviluppatori in giro per il mondo, che meglio esprimono un'idea di domotica innovativa e al passo coi tempi. Tra i progetti esistenti, uno in particolare si avvicina maggiormente all'idea che sta dietro a Jarvis. Il progetto *NinjaBlocks*.

5.1 NinjaBlocks - Il *NinjaBlocks* (Figura 7.1) è un progetto open source e open hardware, creato da Mark Wotton e Marcus Schappi, divenuto popolare grazie ai finanziamenti in crowdfunding racimolati su [KickStarter.com](https://www.kickstarter.com).

Consiste di un piccolo computer che monitora l'ambiente ricevendo input da una grande varietà di sensori. Può inoltre controllare luci, prese e altri attuatori. E' facilmente personalizzabile, con il Ninja Rules Engine si possono creare regole per interagire in maniera dinamica con qualsiasi tipo di attuatore al verificarsi di un evento:

- Accendere una luce quando si entra in stanza.
- Avvisare via SMS quando la lavatrice finisce il lavaggio.
- Scattare e caricare su dropbox una foto quando qualcuno entra in stanza.
- Accendere in remoto i dispositivi al verificarsi di eventi generici.



Figura 7.1 – NinjaBlocks board.

Con questa piattaforma ci si concentra sul dialogo con l'hardware attraverso i linguaggi web che meglio si conoscono, si programma l'automazione domestica, senza doversi preoccupare del lato hardware, del protocollo radio che viene usato o altri dilemmi sui circuiti integrati. Il kit con cui viene fornito infatti, fornisce degli appositi attuatori e sensori che di base interagiscono alla perfezione col ninjablocks (Figura 7.2 – 7.3 – 7.4). La piattaforma è davvero completa, offre una miriade di funzionalità e la possibilità di interagirvi con le API realizzate appositamente per gli smanettoni.



Figura 7.2 – Wireless Button per NinjaBlock



Figura 7.3 – Sensore di movimento per NinjaBlock



Figura 7.4 – Sensore di temperatura e umidità per NinjaBlock

Oltre agli attuatori prodotti per la board, è possibile dialogare con altre periferiche esterne, che supportino lo stesso protocollo radio a 433mhz o che abbiano un accesso a internet. La dashboard del NinjaBlock oltre a monitorare i valori dei sensori, permette di creare regole d'interazione esterne, è come se man mano istruissimo il nostro Ninja. Sul sito stesso viene presentato un esempio di comunicazione esterna con un altro device domotico, recentemente introdotto sul mercato da Philips: *HUE personal wireless lighting* (Figura 7.5).

5.2 Il Philips HUE

Si tratta di un sistema per l'illuminazione domestica e la creazione di ambienti suggestivi attraverso lampadine con bulbi wireless e un "bridge" ponte, che è il cervello del sistema, in grado di comunicare sulla frequenza di queste lampade wireless e far cambiare loro luce, per meglio ricreare un ambiente confortevole. Un singolo bridge supporta fino a 50 lampadine, esso è connesso alla rete domestica di casa per garantire un accesso anche da remoto. Gli sviluppatori Philips hanno pensato inoltre, di fornire delle API per garantire l'accesso al bridge da applicazioni esterne. E' possibile realizzare il proprio sistema domotico che interagisca con i bulbi HUE. In alternativa si possono utilizzare anche le app fornite dalla stessa Philips per interagire con il sistema HUE, attraverso gli smartphone (Figura 7.6).



Figura 7.5 – Sistema Philips HUE



Figura 7.6 – Interfacciarsi al Bridge da Smartphone

5.3 Le differenze con Jarvis

Le differenze risiedono nell'utilizzo di hardware industriale per una maggiore compattezza dei componenti. E in una progettazione software che non vede l'utilizzo di un pc domestico dedicato all'esecuzione del software di sistema, ma a una soluzione indipendente. Questa feature verrà introdotta nelle prossime versioni di Jarvis, impiegando in sostituzione del pc domestico su cui far girare il modulo jarvis pc, un più elegante Raspberry Pi. Questo garantirebbe quei canoni di compattezza e indipendenza, tipico dei sistemi industriali. Un'altra differenza sta nella molteplicità di interfacce disponibili su Jarvis, attualmente sistemi come NinjaBlocks, sono sprovvisti di interfaccia vocale e di un assistente vocale interattivo.

Un sistema come Philips HUE, per la creazione di ambientazioni suggestive all'interno dell'abitazione è paragonabile al sistema con striscia di LED visto in precedenza nel capitolo sull'Architettura, Paragrafo Progettazione Hardware 4.3, sotto-paragrafo 4.3.4. L'effetto è simile, se pur le differenze risiedono ancora nella progettazione del comparto hardware, totalmente differente, dal punto di vista dei componenti e dell'installazione. Ciò non toglie che sistemi come il Philips HUE, essendo dotati di API per gli sviluppatori, possano essere affiancati a quelli attualmente presente in Jarvis, per garantire maggiore dinamicità sotto tutti gli aspetti.

6. Conclusioni e Sviluppi futuri

I miglioramenti e sviluppi futuri a breve termine del progetto possono essere riassunti qui di seguito:

- Miglioramento del sistema di comunicazione radio (utilizzo di Transreceiver). (1)
- Creazione di un livello d'esecuzione a livello server. (2)
- Porting del modulo Smartphone su sistemi differenti da Android. (3)
- Applicazioni smartphone in esecuzione come servizio e non attività. (4)
- Nuova interfaccia radio con telecomando RF 433Mhz. (5)
- Autenticazione migliorata tramite tecnologia RFID. (6)
- Impiego di una componente hardware ottimizzata per il modulo PC (vedi Raspberry Pi). (7)
- Riconoscimento vocale migliorato, con auricolare bluetooth. (8)
- Riduzione dei tempi di verifica dei comandi vocali, confrontando direttamente sui valori intermedi. (9)
- Aggiunta di nuove funzionalità al sistema domotico. (10)
- Miglioramento al sistema di priorità per i comandi vocali. (11)

(1) Uno dei primi miglioramenti, consiste nel migliorare il sistema di comunicazione radio. Infatti gli attuali microcontrollori, comunicano fra loro utilizzando due circuiti differenti, uno per la ricezione e uno per la trasmissione, questo comporta l'utilizzo di più risorse e dinamiche di trasmissione più complesse. Utilizzando appositi moduli Transreceiver (Figura 7.7), che integrino sia una ricevente che una trasmittente, è possibile garantire una comunicazione radio full-duplex (bidirezionale non simultaneo).

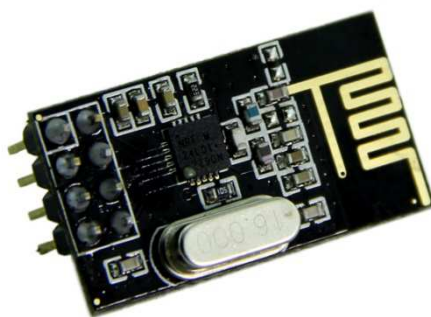


Figura 7.7 – NRF24L01 Transreceiver

(2) Il secondo miglioramento prevede la creazione di un livello d'esecuzione lato server. Attualmente sono presenti due livelli d'esecuzione, il primo lato smartphone,

il secondo lato PC. Con un livello d'esecuzione lato server è pensabile la realizzazione di ulteriori funzionalità che necessitino di una presenza attiva 24 ore su 24, come può essere un server. Il lato smartphone passerebbe il comando al lato server se non riconosciuto, e il server farebbe lo stesso con il modulo PC. Si viene così a creare una vera e propria rete di elaborazione distribuita.

(3) La terza miglioria consiste nell'effettuare il Porting dell'applicativo per smartphone, su sistemi operativi differenti da Android. Si pensi ad iOS o Windows Mobile. (4) L'applicazione, in aggiunta, dovrà girare come servizio in background e non come semplice attività in foreground. In questo modo è possibile richiamarla rapidamente e immettere in circolo i comandi istantaneamente.

(5) La quinta miglioria, garantisce un'altra interfaccia per l'utente. Quella fisica con un telecomando RF. Un telecomando radio al contrario di un normale telecomando infrarosso, permette l'invio del segnale anche senza una diretta visibilità del ricevitore (Figura 7.8).



Figura 7.8 – Telecomando Radio con ricevitore, 433Mhz.

(6) Ancora, per migliorare l'autenticazione con il sistema, è possibile pensare ad una componente aggiuntiva che faccia uso di tecnologia a trasponder RFID, che affianchi il normale utilizzo di credenziali d'accesso.

(7) Una settima miglioria riguarda la sostituzione del PC domestico normalmente usato per far girare il modulo PC, con una soluzione ad hoc, pensata per un utilizzo definitivo e permanente. Ciò che più si avvicina all'idea di un computer essenziale, economico, flessibile, a basso consumo e allo stesso tempo in grado di supportare l'esecuzione degli script, è il Raspberry Pi (Figura 7.9). Questo mini PC è realizzato dalla Raspberry Pi Foundation, organizzazione che si occupa di stimolare l'insegnamento dell'informatica e della programmazione nelle scuole.

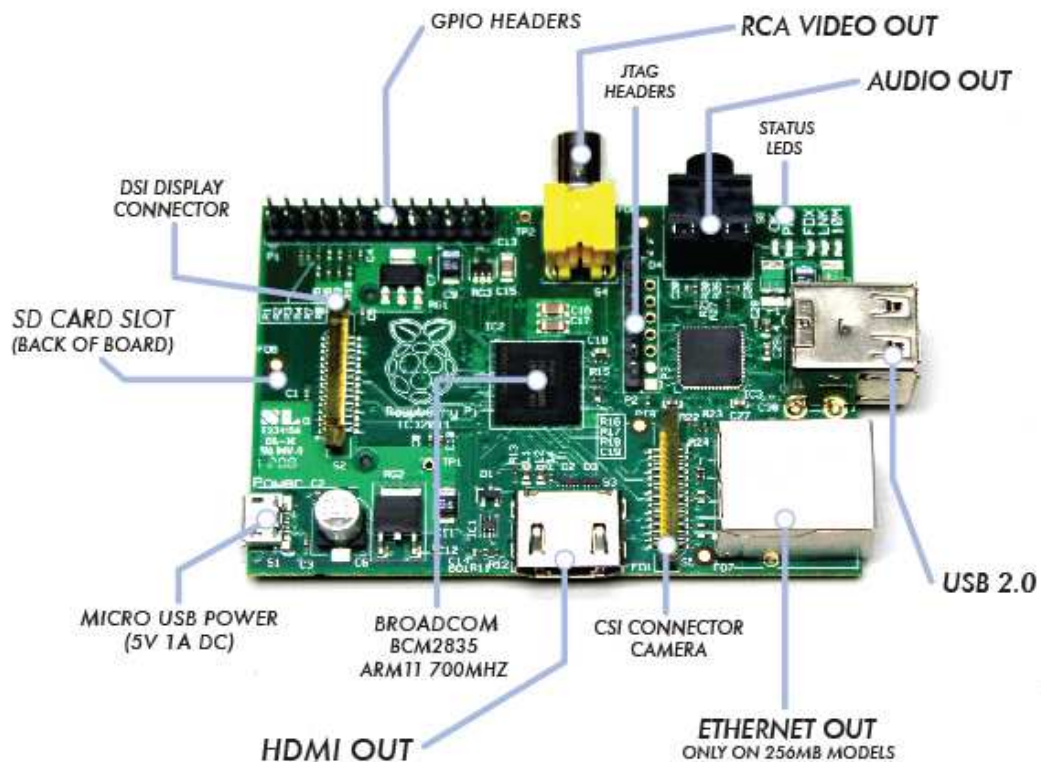


Figura 7.9 – Raspberry Pi

(8) Un'ottava miglioria, prevede l'utilizzo di auricolari bluetooth per l'interazione con l'assistente vocale. Gli auricolari bluetooth, permettono una maggiore usabilità, una maggiore affidabilità, e una migliore qualità audio, indispensabile per un buon riconoscimento. (9) Migliorare ulteriormente il comparto vocale del modulo PC lo si può fare, agendo sui valori intermedi. Piuttosto che aspettare il valore finale di una parola riconosciuta, è possibile operare direttamente con i valori intermedi delle API google. In tal modo si ha un riconoscimento dei comandi fino a 4 volte più rapido nei casi peggiori.

(10) La decima miglioria è l'aggiunta di una nuova gamma di funzionalità per estendere l'ecosistema funzionale di Jarvis. Traduzioni simultane, annotazioni vocali e molto altro.

(11) Infine, un ultimo miglioramento applicabile, risiede nel sistema di priorità che assicuri una priorità dei comandi vocali in base al numero maggiore di match tra il comando e la voce nel dizionario, piuttosto che in base all'ordine di comparsa nel dizionario.

In conclusione, scopo di questo progetto è stato realizzare e presentare un sistema domotico, a basso impatto sull'abitazione, che faccia uso di tecnologie moderne pur restando altamente sotto i prezzi abituali. Trattandosi di un prototipo in via del tutto sperimentale, molte migliorie sono praticabili. Jarvis ha la capacità di crescere in funzione delle necessità e delle disponibilità perché dotato di grande scalabilità e modularità. Il progetto rimarrà open source e liberamente disponibile in rete, per far sì che venga usato e migliorato a piacimento dagli internauti.

Da qui a pochi anni è prevedibile un incremento d'interesse da parte di tutti i Big del settore informatico verso la domotica. Quest'ambito rappresenta il futuro, perché parte integrante della tecnologia, sempre più insinuata nelle nostre vite, irrinunciabile, indispensabile, che rende tutti noi, maniaci del controllo.

BIBLIOGRAFIA

[Deltaenergie.it – figura 2.1]	13
“An Informatics Research Contribution to the Domotic Take-Off”. Ercim- news.ercim.eu.	8
“Home Automation costs”. Totalavcontrol.co.uk	12
http://magazine.greenplanner.it/2012/08/03/decreto-sviluppo-incentivi-per-rifare-limpianto-elettrico-di-casa/	11
1.5 Million Home Automation Systems installed in the US – ABI Research . . .	12
http://www.python.org	23
TTS Class – http://masnun.com	26
http://www.hackerstribе.com/2013/fare-richieste-cross-domain-in-get-e-post-con-jquery-o-xmlhttprequest/	29
http://www.hackerstribе.com/vocabolario/xss-cross-site-scripting/	29
SL4A - https://code.google.com/p/android-scripting/	30
http://www.arduino.cc	32
Imagine RF module - http://www.seedstudio.com	38
http://www.hackerstribе.com/2013/prolunga-rele-v2-0-python-atmega328p/ . . .	36
Haynes, William M., ed. (2011). CRC Handbook of Chemistry and Physics (92nd ed. ed.). CRC Press. p. 10.233.	44
Hackerstribе.com/2013/vlc-hacking-infrared-remote-py-control-2-0/	43
http://www.pjrc.com/teensy/td_libs_IRremote.html	43
http://www.ladyada.net/learn/sensors/pir.html	47
http://learn.adafruit.com/digital-led-strip/	41

http://learn.adafruit.com/photocells	48
http://www.ninjablocks.com	57
https://www.meethue.com	59
http://www.raspberrypi.org	64