

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Scienze di Internet

Realizzazione di
un sistema di automazione domotica
con tecnologia open-hardware

Relatore:
Chiar.mo Prof.
Luciano Bononi

Presentata da:
Luca Giuliano

Sessione II
Anno Accademico 2013-2014

Indice

1	Introduzione	5
2	La domotica	7
2.1	Aree di automazione	10
2.2	Predisposizione dell'impianto	12
2.3	Distribuito o Centralizzato?	13
2.4	Le soluzioni attualmente in commercio	14
2.5	Filosofia Open-Hardware	18
3	Realizzazione del sistema	21
3.1	Panoramica	23
3.1.1	Centralina d'intelligenza	26
3.1.2	Gestore delle luci	26
3.1.3	Termostato e controllore Caldaia	27
3.1.4	Riconoscitore vocale	28
3.1.5	Videosorveglianza e streaming remoto	28
3.2	Strutturazione ed Implmentazione dei componenti	29
3.2.1	Le Interfacce	31
3.2.2	Web Server	41
3.2.3	Il Gestore delle Luci	44
3.2.4	Il Termostato	48
3.2.5	La Videosorveglianza	54
3.2.6	Media Center	57
3.2.7	Applicazione Mobile	60

3.2.8	Altri Servizi	62
4	Utilizzo del sistema: casi d'uso	65
4.1	Automatizzazione	68
4.2	Risparmio energetico	71
5	Conclusioni	73
5.1	Considerazioni tra sistema implementato e sistemi proprietari . . .	77
5.2	Sviluppi Futuri	79

Capitolo 1

Introduzione

Negli ultimi anni l'evoluzione tecnologica ha sviluppato diverse soluzioni per agevolare la vita quotidiana: dagli smartphone ai dispositivi indossabili, la tecnologia tende sempre di più a creare strumenti che l'uomo può utilizzare nella vita di tutti i giorni per semplificare le attività più comuni. Molti interagiscono tra di loro attraverso degli ambienti software e con protocolli di comunicazione standard ben definiti. Smartphone con sistemi operativi come *Android* e *iOS* creano una semi automazione, attraverso un parco applicazioni molto vasto. Un esempio è dato dall'utilizzo dei servizi per promemoria intelligenti oppure per la fruizione di contenuti attraverso comandi vocali. La *Domotica* agevola gli aspetti della quotidianità all'interno dell'ambiente casalingo o, più in generale, di ambienti antropizzati.

Il concetto di *automazione della casa* ha preso piede negli ultimi anni, creando una nuova materia di studio che comprende varie discipline. Attraverso la modifica di un impianto elettrico, la domotica integra elementi "intelligenti" creando una comunicazione attiva tra l'utente e l'ambiente casalingo, attraverso dispositivi che fungono da interfaccia.

Lo studio della domotica si focalizza sugli ambienti in cui l'uomo passa parte del suo tempo giornaliero, tra cui i grandi edifici. In questo caso si parla di *building automation* o "*automazione degli edifici*". L'edificio intelligente consente la gestione coordinata, integrata e computerizzata degli impianti tecnologici (climatizzazione, distribuzione acqua, gas ed energia, impianti di sicurezza), delle reti informatiche e delle comunicazioni con l'evidente scopo di migliorare la gestione, il comfort,

la sicurezza; tuttavia a dispetto di ciò la realizzazione di un impianto domotico comporta il più delle volte un costo elevato, senza avere la possibilità di poter modificare l'impianto stesso. Il motivo principale è dovuto sia dalle restrizioni di legge, sia dalla chiusura del sistema da parte del fornitore.

A questo problema si contrappone un nuovo tipo di tecnologia che sta progredendo negli ultimi anni: la tecnologia **open-hardware**.

La tecnologia open-hardware consente di creare un sistema automatizzato con un costo di gran lunga più basso e con livelli di gestione funzionali simili a quelli dei sistemi proprietari. Questa tecnologia permette alla persona di creare il proprio impianto domotizzato, dando la possibilità di poter progettare e creare un sistema utilizzabile agevolmente nella quotidianità. Grazie all'ausilio di piattaforme sia software che hardware, si ha la possibilità di creare il proprio ambiente domotizzato che comunichi con i dispositivi della casa, quali: luci, tapparelle, elettrodomestici, impianti di riscaldamento, e altro ancora. Le piattaforme hardware per questa tecnologia sono principalmente *Arduino* e *Raspberry Pi*, due dei dispositivi più utilizzati in tecnologia open-hardware. Questi consentono, tramite appositi segnali di *input* ed *output*, di interfacciarsi con una vasta gamma di componenti hardware, come *relays* e *sensori* di vario tipo. Un esempio è la comunicazione tra due dispositivi per rilevare la temperatura del salone e per poter accendere il riscaldamento casalingo.

Questa tesi ha l'intento di spiegare come può essere realizzato un sistema domotizzato casalingo utilizzando dispositivi open-hardware. Inizialmente verranno messi in chiaro i concetti chiave generici di un sistema domotico e verranno discussi i prodotti attualmente in commercio e verrà fatta una piccola introduzione sul concetto di open-hardware. Successivamente verrà discusso il sistema realizzato dandone una panoramica, si esaminerà la strutturazione sia software che hardware e le tecnologie ed i dispositivi utilizzati, per poi enucleare casi d'uso. A seguire le conclusioni.

Capitolo 2

La domotica

Il termine *Domotica* deriva dall'unione dei termini *domus* e *robotics* e spazia oltre alle competenze in ambito informatico ed elettronico, avvalendosi dell'architettura e di determinati campi dell'ingegneria come: energetica, edile, dell'automazione, elettrotecnica, delle telecomunicazioni.

Sviluppandosi nell'ultimo ventennio cerca di migliorare la qualità della vita, la sicurezza, semplificare la progettazione, l'installazione e la manutenzione dell'impianto casalingo, da cui "*casa intelligente*", cioè un ambiente automatizzato da apparecchiature tecnologiche che vanno oltre le apparecchiature tradizionali e che rendono l'ambiente parzialmente autonomo e programmabile dall'utente.

Un sistema domotico consiste dunque principalmente nella realizzazione o modifica di un impianto elettrico in modo da introdurre elementi di automazione. Attraverso la presenza di questa automazione, l'abitazione o edificio diventa finalmente in grado di comunicare in modo attivo e bidirezionale con l'utente mediante l'utilizzo di dispositivi denominati **interfacce**.

Le interfacce, ovvero quei dispositivi che consentono l'utilizzo delle funzionalità implementate nell'ambiente domotizzato, possono essere di due tipi: *attive*, qualora l'utente possa gestire l'impianto in tutti i suoi aspetti, oppure *passive*, qualora il sistema esegua determinate azioni in modo automatico e preimpostate dall'utente stesso. Un esempio di interfaccia può essere l'interruttore casalingo, che consente l'accensione di una luce all'interno dell'impianto elettrico, oppure l'accensione della medesima luce attraverso un'**applicazione mobile** (tablet o smartphone).

L'interazione con le parti della casa (luci, termostati, rilevatori di intrusioni, ecc.) attraverso le interfacce consente di creare una nuova entità: lo **scenario**.

Uno scenario è un insieme di azioni eseguibili contemporaneamente ed attivabili con un unico comando, impartito a seconda dell'interfaccia che si usa. Un esempio di scenario è la possibilità attraverso un unico pulsante di accendere tutte le luci del salone, oppure, in tema di automazione, appena l'utente rientra in casa si accendono le luci se all'esterno è buio, il contrario se vi è ancora luce. Il tema degli scenari è molto importante poichè contribuisce ad agevolare molte delle azioni compiute nella quotidianità, rendendole automatizzate.

Altri due elementi importanti sono: il fattore *tempo* e la *gestione remota*. Un sistema di temporizzazione consente all'utente di programmare temporalmente azioni automatiche, che non dovranno più essere gestite manualmente. La *gestione remota* consente l'utilizzo delle funzioni domotiche attraverso delle interfacce che operano a distanza, come la gestione di azioni attraverso SMS o una apposita applicazione che utilizza la connessione internet.

Di seguito vengono presentati esempi delle entità appena descritte:

- ***Esempi di scenari:***

- *ESCO DI CASA: “Spegni tutte le luci, chiudi le tapparelle e attiva l'antifurto”*
- *RISVEGLIO: “Apri al 50% le tapparelle della camera e accendi la macchina del caffè”*

- ***Esempi di temporizzazioni:***

- *Tutti i giorni dal lunedì al venerdì alle ore 7.00 attiva lo scenario “risveglio”*
- *Tutte le sere alle ore 23.00 chiudi la valvola del gas cucina e attiva l'antifurto*
- *Lunedì, mercoledì e sabato alle ore 5.00 attiva lo scenario “irrigazione”*

- ***Esempi di controllo via SMS (gestione remota):***

- SMS: “Luce esterna: accendi” (accende la luce esterna)
- SMS: “Temperatura soggiorno: ? ” (richiede la temperatura attuale dell’area soggiorno)”

Descritte le entità cardine di un sistema domotizzato, trattiamo ora degli elementi funzionali.

Un sistema domotizzato è composto principalmente da tre elementi funzionali: un dispositivo detto **intelligenza**, le **interfacce** e le **comunicazioni**.

Il **dispositivo d’intelligenza** è la componente che crea e gestisce le relazioni tra i dispositivi e le interfacce: monitora costantemente i vari dispositivi, rendendoli disponibili per la fruizione delle azioni all’utente; solitamente è identificato da un server e può essere associato ad una rete di elementi che comunicano tra di loro.

Le **interfacce** sono state discusse precedentemente e fanno parte di quell’insieme di funzionalità che l’utente può utilizzare nell’impianto domotizzato. In pratica forniscono gli elementi di interazione necessari a poter compiere delle azioni all’interno dell’ambiente domotico.

Le **comunicazioni** permettono agli elementi di intelligenza di dialogare tra loro o con i vari dispositivi dell’impianto. La maggior parte delle volte sono implementati in modo da avere una bassa complessità che ne consente le operazioni primarie e basilari, come ad esempio l’accensione o lo spegnimento di una luce (si può pensare come uno stato 0/1). Attraverso degli standard si possono implementare delle comunicazioni di più alto livello, come la termoregolazione dell’ambiente o la gestione degli scenari.

Oltre agli standard di comunicazione, vengono utilizzati dei dispositivi chiamati *gateway* o *residential gateway*, simili a dei router avanzati, che permettono la comunicazione tra il sistema intelligente e l’esterno della casa. Questi sistemi svolgono il compito che hanno dei router avanzati, ovvero **connettere tutto il sistema casalingo al mondo esterno**.

Descritti gli elementi funzionali, possiamo ora esporre come questi debbano soddisfare delle caratteristiche di funzionamento e quali vantaggi o svantaggi portano rispetto ad un impianto tradizionale.

Le caratteristiche di funzionamento di un impianto domotizzato fanno parte delle funzionalità intrinseche di un tradizionale impianto casalingo:

- **Semplicità:** il fatto che un sistema domotico sia rivolto ad un pubblico che ha competenze sia professionali che non professionali, necessita di una componente fondamentale quale la semplicità di utilizzo del medesimo, in modo che le azioni più comuni siano *user friendly*, naturali ed univoche.
- **Continuità di funzionamento:** il sistema deve essere funzionante in modo continuativo, fornendo anche una assistenza adeguata in caso di guasti momentanei o critici.
- **Affidabilità:** il sistema deve funzionare anche in caso di guasto. Deve sempre fornire il servizio per cui è stato progettato o almeno con funzionalità ridotte; deve essere in grado di segnalare un mancato funzionamento in modo semplice e immediato, sia all'utente che al servizio di assistenza.
- **Basso costo:** un costo ridotto del sistema consente non solo una diffusione ma anche la possibilità di fidelizzazione. Il basso costo è rivolto maggiormente alle periferiche finali (sensori, relays) e nella rete di interconnessione tra queste periferiche.

Da queste caratteristiche di funzionamento scaturiscono dei vantaggi che si tramutano in: **risparmio energetico**, in cui il sistema automatizzato deve eliminare gli sprechi energetici dei sistemi tradizionali, e **automatizzazione delle azioni quotidiane**, che semplificano le azioni quotidiane.

Le caratteristiche sopracitate mettono in risalto un concetto fondamentale: la casa non diventa “intelligente” perchè dotata di elementi intelligenti ma l'automazione e la strutturazione del sistema rendono l'utilizzo dell'impianto già presente più **facile** ed **intuitivo**, introducendo l'elemento di l'**automazione**.

2.1 Aree di automazione

All'interno dell'ambiente casalingo esistono varie aree di automazione ed uno degli scopi principali della domotica è quello di rendere automatizzati questi ambienti.

Le aree di automazione si possono distinguere in:

- **Gestione dell'ambiente**
- **Gestione degli apparecchi domestici**
- **Comunicazione ed informazione**
- **Sicurezza**

La **gestione dell'ambiente** si riferisce principalmente all'automazione degli elementi che governano l'ambiente casalingo, come climatizzazione, illuminazione, irrigazione, ecc. Questa gestione viene effettuata attraverso la comunicazione e l'interfacciamento con *Sensori* o *Attuatori*, che rilevano dati o modificano l'ambiente in modo da soddisfare le esigenze dell'utente, come: la termoregolazione della casa (ad esempio mostrando la temperatura attuale e, secondo le condizioni climatiche, accendere o spegnere il riscaldamento o il climatizzatore), l'illuminazione dell'ambiente, e così via. Inoltre possono essere utilizzati anche per una costante verifica dei consumi energetici, volti al risparmio energetico comunque garantendo e preservando lo stesso confort. Altri esempi di gestione ambientale sono: la gestione dell'acqua per uso sanitario, pannelli fotovoltaici o solari, caldaia. Tutti i dispositivi che gestiscono l'ambiente devono rispettare sia gli standard minimi di funzionamento e qualità per cui sono stati progettati, sia i *carichi energetici* degli elementi utilizzati all'interno dell'ambiente. Un esempio è la gestione di sovraccarichi o distacchi derivati dall'eccessivo utilizzo o da blackout con gruppi di continuità energetica.

La **gestione degli apparecchi domestici** si occupa degli elettrodomestici che possono essere comandati dal sistema domotizzato. Lavatrici, lavastoviglie, frigoriferi, forni e apparecchi sanitari, se precedentemente predisposti per un uso domotico, possono essere utilizzati dall'utente in modo semplice e automatizzato. Il sistema domotico deve consentirne non solo la gestione a distanza, ma anche la manutenzione repentina e sempre aggiornata di ogni singolo apparecchio. In questa categoria rientrano anche citofoni (generici o VOIP), telefonia in generale, accesso ad internet, ecc. Queste ultime categorie forniscono una gestione automatizzata

delle comunicazioni in entrata ed uscita dalla casa, consentendone il controllo totale all'utente.

La **gestione della sicurezza** infine riguarda tutti gli aspetti di sicurezza dell'ambiente casalingo tramite: sistemi di video sorveglianza, antifurti e antintrusioni, antincendi, antiallagamento, fumi o gas, telesoccorso e teleassistenza. La categoria rientra anche nell'ambito delle intrusioni esterne e la conseguente automazione. La sicurezza dell'ambiente casalingo si può dividere in due categorie: **safety**, ovvero la sicurezza della casa da parte di fughe di gas, incendi o eventi dannosi, oppure **security**, la sicurezza contro intrusioni esterne non autorizzate, rapine, atti vandalici. In caso di eventi di questo genere, oltre ad intervenire con specifiche azioni di pronto intervento, il sistema deve essere strutturato in modo da segnalare prontamente e anche a distanza l'accaduto, in modo da effettuare un eventuale intervento. Se sono presenti delle telecamere, allora sarà possibile visualizzare eventuali registrazioni o streaming in tempo reale. La trasmissione e la qualità dello streaming è importante per poter riconoscere appieno il tipo di intrusione e fornire un pronto intervento.

2.2 Predisposizione dell'impianto

Da quanto detto fino ad ora si evince che una buona progettazione dell'impianto ed una preventiva predisposizione sia importante per creare una struttura fisica per la domotica.

Per **predisposizione** dell'impianto elettrico si intende la realizzazione di canalette, vani tecnici, scatole di derivazioni ed altro volti all'inserimento degli elementi domotici, quindi la creazione di una base per il sistema domotizzato.

L'importanza della predisposizione sta nella creazione di una base per la domotica all'interno di un impianto elettrico persistente, poichè costituisce un investimento nel futuro piuttosto che un costo, dando la possibilità di evoluzione funzionale dell'ambiente casalingo tradizionale in linea con le esigenze sempre più influenzate dall'evoluzione tecnologica, perciò molteplici e mutevoli nel corso del

tempo.

Una adeguata predisposizione risulta pertanto una scelta vantaggiosa poichè consente un risparmio notevole rispetto alla riprogettazione o creazione di un impianto *ex novo* per l'utilizzo degli elementi domotici. Quindi una integrazione futura senza predisposizione avrà sicuramente un costo maggiore per la progettazione ed installazione del sistema, diversamente da un impianto già predisposto.

2.3 Distribuito o Centralizzato?

Il problema principale che affligge il campo della domotica riguarda la scelta del tipo di sistema: **distribuito** o **centralizzato**.

Un sistema **distribuito** è un insieme di apparecchiature che comunicano tra loro ed in modo indipendente, mentre un sistema **centralizzato** è un sistema in cui ad una centralina vengono affidati tutti i compiti di gestione, interfacciamento e comunicazione con i dispositivi.

Un sistema centralizzato presenta un costo inferiore rispetto a quello distribuito poichè presenta un unico componente per l'elettronica di controllo al contrario del sistema distribuito, in cui tutti gli elementi del sistema devono avere una componente di controllo indipendente, e quindi aumentando gli eventuali costi di manutenzione. Viceversa, un sistema distribuito costituisce una maggiore affidabilità rispetto ad uno centralizzato poichè ogni dispositivo può lavorare autonomamente; in un sistema centralizzato se viene meno il nodo centrale, tutti gli altri elementi diventano inutilizzabili. Di fatto un sistema distribuito ha la possibilità di essere sempre funzionante anche in caso di guasto e di estensione degli elementi del sistema. Inoltre da maggiore flessibilità rispetto ad uno centralizzato.

L'utilizzo di un tipo di sistema a discapito dell'altro, dipende sempre dalle esigenze e dalla soluzione che può sembrare la migliore. Capita che alcuni utilizzino un approccio **ibrido**, costituito da componenti indipendenti che fanno riferimento ad un nodo centrale. Tuttavia in impianti particolarmente complessi, dove le funzioni richieste sono svariate, il server centrale può diventare un nodo fondamentale di funzionamento del sistema. Ovviamente questo comporta le problematiche sopra-

ciate, ma in alcuni casi, ad esempio per i sistemi più complessi come i grandi edifici, l'uso di una tipologia piuttosto che l'altra diventa una scelta obbligata.

2.4 Le soluzioni attualmente in commercio

Esistono diverse soluzioni domotiche in commercio, tutte implementate e progettate negli ultimi anni. Queste soddisfano le esigenze dell'utente, attraverso la scelta di sistemi distribuiti, centralizzati oppure ibridi.

I sistemi domotici in commercio sono principalmente di due tipi: **proprietary** e **open**.

I sistemi *proprietary* fanno parte di quelle soluzioni che utilizzano hardware e software creato dalla ditta fornitrice del servizio, tra cui troviamo *bTicino* con il sistema *MyHome*. *Myhome* di bTicino è un sistema ibrido, costituito da un dispositivo centrale che comunica, attraverso un bus doppino a due fili oppure tramite tecnologia ZigBee, con dispositivi indipendenti che consentono di interagire con l'ambiente casalingo.

«MyHOME utilizza due tecnologie installative: BUS con doppino 2 fili e radio ZigBee e può anche essere integrato a sistemi diversi quali per esempio KNX e DALI. Mediante apposite interfacce e l'impiego del protocollo TCP/IP, MyHOME è in grado di gestire e supervisionare la casa anche da remoto utilizzando un telefono fisso, mobile o una connessione Internet.»[1]



Figura 2.1: MyHome bTicino

Il sistema consente la gestione di tutte le aree di automazione precedentemente descritte, quindi una gestione altamente automatizzata delle luci, della termorego-

lazione, diffusione sonora, sicurezza, e così via. Inoltre utilizza due tipi di tecnologie per la comunicazione: via cavo e wireless. L'utilizzo di due tecnologie per la comunicazione rende il sistema altamente espandibile attraverso componenti che comunicano tra loro. *«La tecnologia BUS si basa sull'impiego di dispositivi collegati tra loro mediante un cavo (BUS) a due conduttori, utilizzato per il trasporto delle informazioni e l'alimentazione elettrica a bassa tensione (27 Vdc). Tramite apposite interfacce gli impianti a BUS MyHOME possono essere facilmente integrati ad impianti a BUS con protocolli di comunicazione diversi, quali standard Konnex, Dali ecc. ed estesi con dispositivi di comando radio ZigBee.[...] La tecnologia Radio ZigBee si basa sull'impiego di dispositivi di comando alimentati a batteria e dispositivi attuatori alimentati dalla rete elettrica e connessi al carico da controllare, che comunicano via radio con segnale avente una frequenza di 2,4 GHz. Grazie a queste caratteristiche è possibile realizzare con estrema flessibilità nuovi impianti elettrici o modifiche di esistenti con ridotti cablaggi e minimi interventi alla struttura muraria. Mediante una interfaccia specifica, i dispositivi di comando radio ZigBee possono essere integrati ad un impianto con tecnologia BUS (più precisamente l'impianto Automazione).»*[1]

MyHome utilizza come protocollo di comunicazione il protocollo *Open WebNet*. *«Open Web Net è un linguaggio disponibile in forma “dominio pubblico” sviluppato per permettere a chiunque abbia conoscenze informatiche di linguaggi ad alto livello, di realizzare funzioni innovative scambiando dati e comandi tra un'unità remota e il sistema MyHOME. Il linguaggio fornisce un livello astratto di comunicazione che permette al professionista del settore (System integrator ecc.) di effettuare il controllo remoto di MyHOME utilizzando dispositivi come PC, Smartphones, Tablet o di realizzare applicazioni evolute derivanti dall'integrazione di MyHOME con apparecchiature ed elettrodomestici di altri marchi, senza curarsi dei dettagli dell'installazione e senza conoscere la tecnologia a BUS SCS.»*[1]

I sistemi **open** fanno parte di quel tipo di sistemi che utilizzano solitamente tecnologia **open-hardware**, la quale consente una alta personalizzazione sia da parte del produttore che dell'acquirente. I sistemi open utilizzano tecnologie persistenti e protocolli di comunicazione non proprietari, come *MQTT*. Le aziende che for-

niscono questi sistemi guadagnano principalmente sulla vendita dei *KIT* e dalla relativa installazione, dei pacchetti contenenti la soluzione domotica pronta all'uso e personalizzabile. Un esempio di soluzioni domotizzate open è *Giardino Domotizzato*. Questa ditta fornisce un dispositivo centralina che comanda gli elementi di un impianto di irrigazione, utilizzando il progetto open chiamato *Open Sprinkler*. «Le centraline di irrigazione che proponiamo hanno la caratteristica principale di poter essere controllate da remoto e da qualsiasi apparato: smartphone, computer, tablet, tv, senza subire l'impatto di costi eccessivi. Il privato appassionato o l'Azienda che si occupa di irrigazione sa quanto sia importante un tempestivo intervento per salvaguardare le aree verdi domestiche o le zone di produzione, così come ad esempio un campo da calcio, da golf, un frutteto o, ancora, una vigna. Le centraline di irrigazione o programmatori sono un valido supporto alle attività, un componente essenziale per la gestione dell'irrigazione; grazie alle nostre potrete risparmiare tempo e denaro, e farlo risparmiare ai Vostri clienti: niente più chilometri da percorrere per sistemare l'orario o modificare i tempi.»[2]

Questa centralina può essere costruita attraverso l'ausilio di uno dei dispositivi open-hardware più utilizzati: il *Raspberry Pi*, che verrà descritto nel paragrafo successivo.

Open Sprinckler crea un sistema centralizzato, dove la centralina costituisce il nodo centrale per la comunicazione tra gli elementi del giardino come elettrovalvole, sensori di pioggia e di umidità. La comunicazione avviene attraverso cavo e tramite le porte di comunicazione fornite dalla centralina. Questa soluzione offre la possibilità di integrazione nell'impianto di irrigazione già esistente in modo che i costi per la creazione risultino notevolmente abbassati; inoltre il fattore *predisposizione* gioca un ruolo importante in questo caso.

Ma esistono vantaggi o svantaggi tra i due tipi di sistemi? La risposta a questa domanda è da cercare nelle esigenze che l'utente ha e i costi che può affrontare.

Come si può vedere, entrambi i sistemi forniscono una assistenza adeguata in caso di guasti. Inoltre danno la possibilità di espansione ma in modi differenti: i sistemi proprietari forniscono dei protocolli di comunicazione per l'installazione di nuovi dispositivi, che devono poter comunicare attraverso un specifico protocollo, men-



Figura 2.2: Centralina di irrigazione Open Sprinkler

tre i sistemi open non solo forniscono dei protocolli di comunicazione (solitamente open) ma anche la possibilità poter integrare con altri tipi di protocolli, lasciando la scelta all'utente più esperto. Ad esempio *bTicino* utilizza *Open WebNet* come protocollo per questo scopo, mentre *OpenSprinkler*, attraverso l'espansione per *Raspberry PI*, consente all'utente di scegliere il protocollo di comunicazione più adatto. Inoltre la possibilità che il sistema *open* offre è di poter comunicare con il sistema *proprietario*, in modo da integrare maggiori dispositivi ed ambienti, abbassando ulteriormente i costi.

2.5 Filosofia Open-Hardware

Prima di introdurre il capitolo successivo, è necessario fare una breve introduzione sull'open-hardware, in modo da poter meglio spiegare le componenti che costituiscono gli elementi del sistema implementato.

La tecnologia *open-hardware* è una tecnologia che segue la filosofia *open-source* applicandola a dispositivi hardware. L'*open-hardware* crea dispositivi fisici costruiti e progettati dai *maker* con specifiche messe online e disponibili a tutti.

La diffusione di questa tecnologia è dovuta principalmente dall'introduzione nel mercato di dispositivi che hanno reso possibile la fruizione. *Arduino* e *Raspberry PI*, due delle principali piattaforme più utilizzate nell'*open-hardware*, hanno permesso grazie al basso costo e la facilità d'uso di espandere questo fenomeno introducendo all'informatica e all'elettronica anche gli utenti più inesperti.

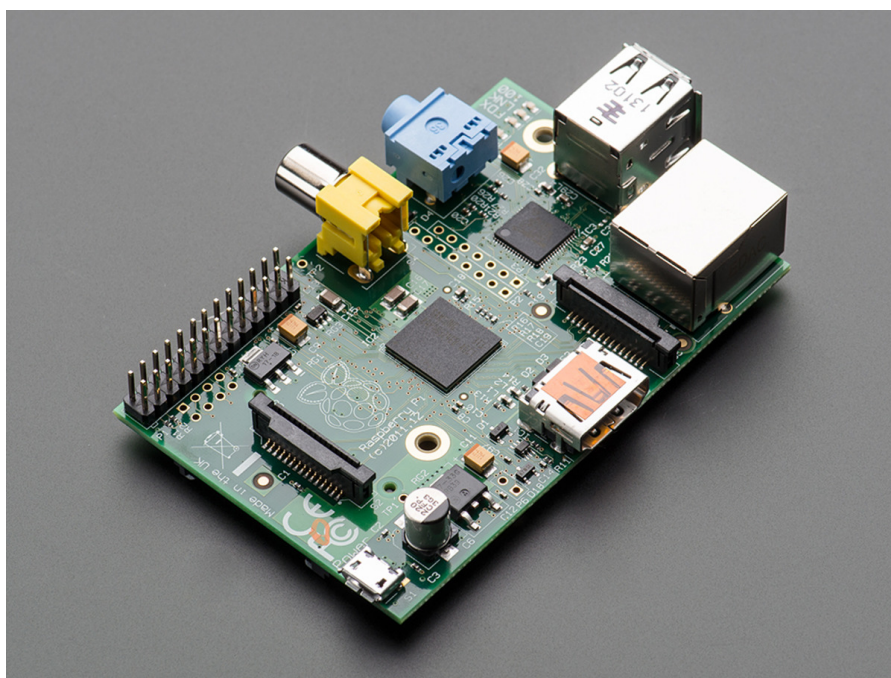


Figura 2.3: Raspberry PI Modello B

Raspberry Pi è un vero e proprio computer con un processore **ARM V11** da **700MHz**, **RAM da 256 o 512 MB** e connessioni **USB** e **Ethernet**. La componente principale del Raspberry Pi è la possibilità di emettere segnali digitali attraverso dei PIN detti GPIO, con i quali è possibile comunicare con diversi di-

spositivi elettronici compatibili.

Corredato inoltre da porta **HDMI**, **Analog VIDEO/AUDIO** e **Slot per SD Card**. Esistono tre tipi di *Raspberry Pi*: **Modello A**, simile al **Modello B** (mostrato nella figura La figura) ma sprovvista di una porta USB e di connessione Ethernet, **Modello B** ed il recente **Modello B+**, che aggiunge più **GPIO** e due porte aggiuntive USB. Nato principalmente per scopi didattici e per affacciare al mondo dell'elettronica e dell'informatica i più giovani, inoltre viene utilizzato, grazie alla possibilità di emettere e riconoscere input ed output digitali, anche per l'applicazione a progetti più importanti.

Arduino è un progetto che parte dall'idea di creare un **microcontrollore** facilmente programmabile e che possa interagire con una vasta gamma di dispositivi hardware in circolazione. La parte hardware è costituita da una scheda a singolo **microcontrollore** (che cambia a seconda dei modelli) capace di gestire input ed output fisici, interfacciandosi con dispositivi hardware come: led, bottoni, motori passo passo, relays, ecc. La parte software e di programmazione che fa parte di *Arduino* consiste in una **IDE cross-platform**, il cui linguaggio di programmazione è una derivazione di *Wiring* e *Processing*, progetti fondamentali nell'ambito del physical computing e dell'interaction design.

Questi dispositivi **consentono una connessione sia wired che wireless**, con moduli aggiuntivi o attraverso specifici moduli hardware implementati nei dispositivi, ad esempio: *Arduino* consente una connessione wired tramite il modulo *Ethernet Shield* e wireless tramite vari dispositivi come *XBee*, mentre *Raspberry PI* consente una connessione wired tramite la porta *Ethernet* fornita nella scheda stessa e wireless tramite appositi hardware come chiavette *USB WiFi* di produttori compatibili.

Nel capitolo successivo verrà trattata l'implementazione di un sistema utilizzando componenti open-hardware e con l'ausilio di *Raspberry PI*.

Capitolo 3

Realizzazione del sistema

Il primo capitolo introduce allo studio della domotica nelle sue componenti generiche, mostrando i campi di applicazione e le regole da seguire per la progettazione, ed in questo capitolo tratteremo come è possibile realizzare un sistema domotico utilizzando la tecnologia open-hardware.

Il primo interrogativo che ci si pone è: perchè sviluppare un sistema domotico utilizzando tecnologia open-hardware? A questo interrogativo esistono due risposte principali che hanno portato alla realizzazione del sistema che verrà descritto: per **abbattere i costi di produzione**, poichè le piattaforme open-hardware come Raspberry Pi ed Arduino sono disponibili a prezzi accessibili ai privati eliminando costi aggiuntivi da parte di produttori di sistemi proprietari, per **esigenza**, in modo da ottenere una automazione della casa come descritta nel capitolo precedente, e per **cultura personale**, poichè la tecnologia open-hardware consente di acquisire conoscenze in campo elettronico/informatico importanti anche ad utenti meno esperti.

Per i motivi sopracitati si è optato per l'utilizzato di **Raspberry PI** ed **elettronica Arduino e Raspberry Pi** compatibile, come *relays* e sensori di temperatura, offrendo una soluzione accessibile a tutti sia in termini di costi che in termini di potenzialità. I costi, rispetto ad un impianto domotico proprietario, vengono ridotti di oltre **dieci volte**, come i costi di produzione, di installazione del sistema, dell'elettronica di consumo e di produzione del software per la gestione del sistema. Inoltre l'implementazione di tipo *open-hardware* consente all'utente esperto

di selezionare o costruire tutti gli elementi del sistema, dando la possibilità di personalizzare l'impianto. Infine i costi di manutenzione risultano dimezzati sia per il tipo di elettronica utilizzata (a basso costo), sia rispetto alla manutenzione che è tutta a carico del realizzatore, il quale può scegliere che tipo di manutenzione effettuare e su quale componente del sistema, abbattendo ulteriormente i costi di intervento rispetto ad un produttore proprietario. D'altronde, la progettazione e l'implementazione *ex-novo* del sistema necessita di conoscenze minime in elettronica ed informatica, per poter implementare le strutture del sistema.

L'elettronica di consumo, come *relays* e sensori di temperatura è reperibile nei negozi di elettronica ed elettrotecnica a costi molto bassi. Basti pensare che un sensore di temperatura, utilizzato per la termoregolazione dell'ambiente, costa meno di 80 centesimi a pezzo. Inoltre, è necessario che l'elettronica sia compatibile con i dispositivi open-hardware e che possa comunicare attraverso appositi driver e di conseguenza non tutta l'elettronica può essere compatibile con i dispositivi open-hardware in circolazione; ad esempio non tutti i dispositivi elettronici compatibili con Arduino, che emette segnali sia digitali che analogici, è utilizzabile su Raspberry Pi, che emette solo segnali digitali. Trattandosi di elettronica semplice in termini di strutturazione il costo è molto ridotto, al contrario se si utilizzassero dei dispositivi di elettronica più complessi, come schede progettate ad hoc, il costo aumenterebbe.

Date le precedenti premesse, il sistema implementato è stato concepito per poter essere **usato** anche da un **utente inesperto**, creando una esperienza d'uso e una installazione del sistema nell'impianto casalingo alla portata di qualsiasi target d'utenza. Questo punto è stato curato cercando di massimizzare la semplicità e mostrando come un sistema open-hardware *ex-novo* possa essere progettato ed implementato in modo da creare un ecosistema fruibile e personalizzabile da qualsiasi tipo utenza. A tal fine sono stati implementati meccanismi sia software che hardware, come interfacce utente semplici ed intuitive oppure con l'inserimento di componenti software per la configurazione, in modo da usufruire delle funzionalità del sistema ed installarlo nell'ambiente casalingo in modo facile, veloce e soprattutto alla portata di tutti. Nelle sezioni successive verrà descritto il sistema implementa-

to, dando una panoramica delle funzionalità messe a disposizione e successivamente discutendo tutti gli aspetti di progettazione ed implementazione.

3.1 Panoramica

Il sistema domotico implementato è di tipo **ibrido**, composto da dispositivi che svolgono funzioni sia da nodo centrale che di nodo indipendente. I vantaggi e svantaggi di un sistema ibrido sono stati discussi nel capitolo precedente e la scelta di questa tipologia è dovuta principalmente alla bassa complessità del sistema stesso, in modo da consentire l'eventuale integrazione con dispositivi più complessi. Per la realizzazione sono stati utilizzati dei *Raspberry Pi model B e model A*, che hanno la necessaria (se non eccessiva) potenza di calcolo per poter effettuare tutte le operazioni richieste dall'utente e degli altri dispositivi. I sistemi operativi utilizzati nei dispositivi sono due: **Raspbian**, derivata di *Debian 7.0 Squeeze* per *Raspberry Pi*, e **RaspBMC**, derivata di *Raspbian* con funzionalità di media center con l'utilizzo di *XBMC*.

Sono stati utilizzati questi due dispositivi sia per quanto riguarda le loro specifiche, come la potenza di calcolo e l'implementazione di tipo *open-hardware*, sia per quanto riguarda la tipologia di sistema domotizzato, consentendo di avere in un unico dispositivo il controllore dell'elettronica, l'intelligenza e il protocollo di comunicazione. Inoltre il *Raspberry Pi* è stato scelto per via della sua forte scalabilità, data dall'interfacciamento con la maggior parte dei dispositivi hardware digitali in commercio, come sensori di temperatura e *relays Arduino*. Il *Raspberry Pi*, un computer a tutti gli effetti, consente di essere utilizzato per diversi scopi all'interno del sistema, ad esempio la creazione di termostato, centraline di irrigazione, centraline d'intelligenza, Web Server, e così via. Quest'ultimo fattore è molto importante, poichè la **versatilità** è la chiave di questo dispositivo e consente di creare dispositivi più complessi per una integrazione futura all'interno dell'impianto domotizzato.

La tipologia ibrida del sistema comporta la definizione di una rete di nodi ibrida, a metà tra una rete a **stella** ed una **mesh**. Una rete a **Stella**, o *Star Network*,

è un tipo di rete in cui un nodo centrale, visibile a tutti i nodi, comunica ed intermedia tra i nodi della rete. Una **rete Mesh**, o *Mesh Network*, è una rete decentralizzata, in cui non esiste un nodo centrale e dove tutti i nodi connessi alla rete possono comunicare tra loro attraverso un protocollo di comunicazione. Nel caso del sistema implementato, un dispositivo detto **Centralina d'intelligenza** viene eletto nodo centrale, fornendo le funzionalità di comunicazione tra i nodi e a cui tutti i nodi si connettono per avere informazioni e comunicare tra loro, ma allo stesso tempo ogni dispositivo connesso può comunicare con un altro per richiedere informazioni e eseguire azioni instaurando una comunicazione diretta. Questa tipologia di rete comporta un **vantaggio** l'interoperabilità dei dispositivi, consentendo di eleggere uno dei dispositivi connessi a centralina d'intelligenza in caso di guasto del nodo centrale; ma comporta anche uno **svantaggio** derivato dalla necessità di inserire una gerarchizzazione e dei vincoli tra i dispositivi, in modo che le richieste fatte non collidano tra di loro.

Per quanto riguarda le comunicazioni, esse si dividono in due tipi: **comunicazione tra i dispositivi** e **comunicazione con l'elettronica**. Le **comunicazioni tra dispositivi** possono essere sia *wired* (*Ethernet*) che *wireless* (attraverso il protocollo *802.11*), per cui è stata implementata una comunicazione semplice e leggera tra i dispositivi, che rende univoche le richieste effettuate. Per fare questo è stato utilizzato un **Server Web** che gestisce le richieste da parte sia delle interfacce fornite dal sistema sia dei dispositivi. Il server web contiene le interfacce per la gestione e l'utilizzo del sistema per l'utente e le primitive di comunicazione, effettuando le richieste attraverso il protocollo TCP/IP . Sono state implementate delle **chiamate asincrone** per ogni funzionalità del sistema ed un *payload* ad hoc per ogni chiamata, come l'accensione delle luci, termoregolazione ed altro ancora. Le **comunicazioni con l'elettronica** sono di tipo wired attraverso librerie scritte in linguaggio *Python* e *script in Bash*, che consentono l'interfacciamento con l'elettronica come *relays*, attivandone il funzionamento. Tutti questi aspetti verranno discussi dettagliatamente nelle sezioni successive.

Grazie a questo tipo di implementazione è stato possibile realizzare il sistema creando le seguenti soluzioni:

- **Centralina d'intelligenza**
- **Gestore delle Luci**
- **Termostato e controllore Caldaia**
- **Riconoscitore Vocale**
- **Media Center**
- **Video Sorveglianza e Streaming remoto**

Queste sono alcune delle soluzioni che si possono sviluppare attraverso Raspberry Pi, consentendo al progettista di poter spaziare nella creazione di nuove soluzioni.

Tutti gli elementi implementati utilizzano una logica di sistema semplice, poichè, comunicando tra loro attraverso TCP/IP, compiono azioni di **automazione**. I dispositivi sono collegati alla rete casalinga attraverso il router domestico, che smista gestisce le connessioni di tutto l'ambiente. Quindi l'entità detta *Home Gateway* è data dal router casalingo che implementa le funzionalità di routing dei dispositivi e di esposizione del nodo centrale all'esterno della rete domestica. Per esporre il sistema al di fuori della rete domestica è necessario conoscere l'IP pubblico del sistema e reindirizzare le richieste direttamente al nodo centrale o ad uno dei nodi esposti. Per fare ciò è stato utilizzato il servizio *Dynamic DNS*, servizio di host implementato sulla maggior parte dei router in commercio.

Le funzionalità implementate dai dispositivi vengono presentate attraverso delle apposite interfacce, con cui l'utente interagisce. Le **interfacce** sono di tre tipi: una a lato **applicativo**, tramite una applicazione per dispositivi mobili e un portale web, una **ambientale**, attraverso comandi vocali attivabili tramite riconoscitore ambientale o dall'applicazione mobile, e di **automazione**, attraverso l'utilizzo dei *TAG NFC* che compiono determinate azioni. Queste interfacce sono state progettate con lo scopo di semplificare l'utilizzo del sistema nel quotidiano ed accessibile a qualsiasi tipo di utenza, e verranno discusse successivamente nel dettaglio.

Spieghiamo brevemente le soluzioni precedentemente elencate, illustrandone lo scopo e poi l'implementazione.

3.1.1 Centralina d'intelligenza

La centralina d'intelligenza funge da nodo centrale del sistema, la quale smista e normalizza le richieste provenienti dalle interfacce e dai dispositivi. Nel sistema, l'intelligenza, le interfacce e le comunicazioni sono state strutturate in modo che questi servizi siano implementati all'interno di un unico dispositivo, che svolge tutte queste funzionalità. Considerata la natura dei dispositivi utilizzati questa risulta la soluzione migliore, anche se sarebbe auspicabile separare la logica implementativa di queste componenti in caso di sistemi domotizzati più complessi. *Qualunque dispositivo connesso alla rete può essere eletto a centralina d'intelligenza*, poiché l'implementazione software creata per ogni dispositivo consente di gestire tutti gli aspetti di controllo delle funzionalità implementate all'interno del sistema. Inoltre, la centralina di intelligenza può anche svolgere il compito di controllore dell'elettronica, gestendo l'elettronica al suo interno come *relays* e sensori di vario tipo.

Il Raspberry Pi fornisce una soluzione *All-in-one*, dando la possibilità di poter utilizzare un unico dispositivo che svolge il compito sia da centralina d'intelligenza che da nodo finale; questo è uno dei motivi principali dell'utilizzo di questa piattaforma.

La centralina d'intelligenza gestisce e normalizza tutte le richieste effettuate dalle interfacce o dai dispositivi attraverso delle funzioni primitive di comunicazione, che consentono l'esecuzione di azioni come controllo delle luci, termoregolazione, attività multimediali del media center, e così via, impartendone l'esecuzione. Infine, la centralina d'intelligenza può essere utilizzata come nodo che *consente di accedere dall'esterno della rete* al sistema domotizzato, funzionalità utile in chiave di automazione.

3.1.2 Gestore delle luci

Una delle funzionalità basilari di un qualsiasi impianto domotizzato è la gestione dei punti luce; il sistema implementato fornisce una gestione delle luci attraverso uno o più dispositivi che, fatta richiesta attraverso l'interfaccia di riferimento, provvede ad accendere e spegnere i punti luce richiesti. Nell'implementazione è stato scelto di gestire massimo sei punti luce per dispositivo, corrispondenti a sei

delle *GPIO* disponibili nel *Raspberry Pi* consentendo, attraverso la connessione di più dispositivi installati nella rete, la gestione di un numero elevato di punti luce. L'accesso ad ogni punto luce può essere effettuato in due modi: o attraverso l'**interfaccia** messa a disposizione dal **nodo centrale** della rete, che contiene al suo interno la configurazione di tutti i punti luce disponibili, oppure collegandosi **direttamente al dispositivo che controlla i specifici punti luce**, previa mappatura dei dispositivi connessi alla rete. Questa funzionalità intrinseca del sistema consente di comprendere il vantaggio citato precedentemente: in caso di guasto del nodo centrale, è possibile comunque usufruire delle funzionalità implementate collegandosi direttamente al nodo desiderato oppure eleggendo uno dei nodi finali della rete a nodo centrale.

3.1.3 Termostato e controllore Caldaia

Il Termostato o Controllore della Caldaia è quel componente domotico che gestisce la termoregolazione della casa rispetto all'impianto di riscaldamento. Un comune termostato consente di impostare la temperatura desiderata nell'ambiente in cui è installato e di poter accendere o spegnere la caldaia fino a raggiungimento della temperatura desiderata; inoltre consente di impostare i tempi di accensione o spegnimento attraverso un *timer*. Il componente domotizzato del termostato si avvale di tutte le funzionalità di un termostato comune, ma è in grado di fornire una automazione attraverso la comunicazione con sensori di temperatura sparsi nelle stanze, consentendo il monitoraggio degli ambienti e garantendo lo stesso comfort e riducendo gli sprechi energetici.

Attraverso una interfaccia apposita, il termostato implementato consente di controllare e gestire l'ambiente sia collegandosi direttamente al dispositivo, sia collegandosi al nodo centrale. Inoltre l'interfaccia è stata progettata per la gestione della temperatura ambientale, con l'impostazione dei valori di temperatura e programmazione del timer; può essere fruita attraverso l'interfaccia web e mobile, inoltre, tramite l'ausilio di un nodo centrale, è possibile utilizzare l'interfaccia vocale e di automazione tramite tag NFC.

3.1.4 Riconoscitore vocale

Il Riconoscitore Vocale fa parte di un particolare tipo di interfaccia implementata dal sistema che consente di eseguire azioni tramite comandi vocali, configurabili attraverso l'interfaccia web, consentendone l'inserimento, la modifica e l'eliminazione. I comandi vocali possono essere di due tipi: **preimpostati**, che svolgono determinate azioni già predisposte dal sistema, e **configurabili manualmente**, dove l'utente può inserire manualmente il tipo di comando e l'azione svolgere. Questa funzionalità risulta utile sotto diversi aspetti tra cui: **un utilizzo immediato del sistema**, che può essere fruito in qualsiasi parte della casa, **semplicità d'uso**, una totale **personalizzazione** (attraverso l'interfaccia web) e la **fruizione di contenuti multimediali**. Nella progettazione del sistema è stata data maggiore importanza alle funzioni di gestione multimediale, creando l'interfacciamento con un media center comandato attraverso il riconoscitore ambientale o il riconoscitore nell'applicazione mobile, inoltre si può comandare il media center attraverso funzionalità manuali all'interno dell'interfaccia Web o nell'applicazione Mobile.

3.1.5 Videosorveglianza e streaming remoto

Lo streaming remoto e la videosorveglianza implementata nel sistema forniscono gli elementi di gestione della sicurezza discussi nel capitolo precedente. Il sistema implementa, attraverso delle videocamere, la rilevazione di **intrusioni non autorizzate**, inviando un avviso in tempo reale direttamente all'utente, e lo **streaming in tempo reale**. L'aspetto della sicurezza all'interno dell'ambiente casalingo è molto importante e l'implementazione tramite dispositivi come Raspberry Pi consentono di poter gestire tutti gli aspetti, ad esempio attraverso la connessione con webcam ip oppure webcam wired.

È stata progettata una interfaccia apposita per poter fruire i contenuti, basata sull'utilizzo di *webcam* che possono essere utilizzate o dal nodo centrale oppure direttamente da un dispositivo connesso alla rete. Inoltre vengono impostati dei meccanismi di sicurezza per la gestione delle intrusioni, con appositi avvisi inviati all'utente. I servizi sono stati implementati separatamente nel sistema: il sistema di gestione delle intrusioni non autorizzate utilizza il programma *Motion* che at-

traverso degli script ed un programma client email invia un messaggio istantaneo ad un indirizzo di posta preconfigurato appena viene rilevato un movimento, mentre il sistema di streaming in tempo reale utilizza il programma *MJPG-Streamer*, che apre un canale streaming di tipo HTTPS trasmesso in tempo reale.

3.2 Strutturazione ed Implmentazione dei componenti

In questa sezione descriveremo come è stato strutturato il sistema nelle sue componenti, spiegando non solo il loro funzionamento nel dettaglio ma anche le scelte progettuali implementate successivamente.

Il sistema domotizzato implementato, come detto in precedenza, è di tipo ibrido. Le componenti che consentono che questo sistema sia ibrido sono principalmente due: **le primitive di comunicazione** tra i dispositivi e la possibilità di **configurazione**. La comunicazione tra i dispositivi è una comunicazione **TCP/IP** con chiamate **asincrone** ad un **webserver** installato su ogni dispositivo. Le primitive di comunicazione sono delle richieste che vengono effettuate dalle interfacce utente oppure dai dispositivi stessi di tipo *HTML POST e GET* che, a seconda della funzionalità richiesta, consentono l'esecuzione di comandi impartiti utilizzando degli *script* lato server. Queste primitive sono univoche e consentono di eseguire le azioni a più basso livello, comunicando con l'elettronica; sono **implementate su ogni dispositivo**, che ha la possibilità sia di comandare l'elettronica di altri dispositivi connessi alla rete, sia di poter comandare l'elettronica applicata a se stesso.

Per fare in modo che vi sia una strutturazione ibrida, è stato utilizzato un **file di configurazione**, che consente di configurare il sistema affinché possa eseguire le operazioni richieste. Ogni dispositivo contiene il proprio file di configurazione, che a sua volta contiene i dati per usufruire sia delle funzionalità messe a disposizione dal dispositivo, sia delle funzionalità di altri dispositivi connessi alla rete nel caso si trattasse del nodo centrale. All'interno sono contenuti tutti i parametri di riferimento ai dispositivi connessi alla rete ed alle loro funzionalità attraverso una

associazione $\langle \text{chiave} \rangle, \langle \text{valore} \rangle$. Ogni parte del file di configurazione contiene le seguenti sezioni:

- **Impostazioni Generali del sistema**
- **Comandi Utente (inerenti ai comandi vocali)**
- **Comandi di gestione vocale**
- **Parametri di connessione per Controllore Luci e Caldaia**

Ognuna di queste consente di usufruire delle componenti domotizzate del sistema. Le impostazioni generali del sistema contengono quei parametri che sono comuni a ogni dispositivo. Le funzionalità minime garantite da ogni dispositivo contenute in questa sezione sono quelle di *riconoscimento dei comandi vocali* e di servizi riguardanti il *controllore del media center*. Inoltre si trova l'*indirizzo IP* del dispositivo gestore del *termostato*, che consente alle primitive di reperire l'ubicazione esatta all'interno della rete.

I comandi utente sono quei comandi che possono essere attivati attraverso le funzionalità di riconoscimento vocale, che verranno descritte successivamente. Sono strutturati sempre come $\langle \text{chiave} \rangle \langle \text{valore} \rangle$, dove la chiave è il comando da pronunciare mentre il valore è l'esecuzione corrispondente.

I parametri corrispondenti al gestore delle luci sono utilizzati per la connessione e l'interfacciamento del sistema con i punti luce messi a disposizione dal dispositivo. Sempre tramite l'associazione chiave valore, ci sono due parametri: il *label*, cioè il nome indicativo della luce che si vuole azionare, e il *numero*, indicativo di quale luce si vuole azionare e contenente un eventuale indirizzo IP per l'azionamento da parte di un dispositivo esterno. Successivamente verrà spiegato il motivo della strutturazione di quest'ultimo parametro, mentre il primo esiste soltanto per dare un feedback all'utente su quale luce si vuole comandare.

Infine sono contenute altre sezioni nel file di configurazione che possono essere aggiunte e modificate, come il gestore della radio web, che consente di attivare lo streaming della radio scelta. L'aggiunta di sezioni consente di integrare nuove funzionalità nel sistema.

Come si può notare l'utilizzo del file di configurazione innesca la natura ibrida del sistema, poichè se ogni dispositivo ha il proprio file di configurazione, allora è possibile utilizzare i dispositivi in modo indipendente (tipologia distribuita) ed allo stesso tempo è possibile eleggere uno di questi a nodo centrale e fruire di tutte le funzionalità implementate dai singoli dispositivi direttamente da quest'ultimo (tipologia centralizzata). È importante una buona strutturazione del file di configurazione, che dia la possibilità non solo di configurare i dispositivi, ma di trasmettere alle primitive di comunicazione i dati minimi per la gestione delle richieste.

3.2.1 Le Interfacce

Un sistema domotico implementa delle interfacce, in cui l'utente usufruisce le funzionalità del sistema. Le interfacce implementate nel sistema realizzato sono principalmente di tre tipologie: **Riconoscitore vocale**, che a sua volta può essere di tipo ambientale o in applicazione mobile, **Interfaccia Web**, dove si trovano tutte le funzionalità del sistema e la loro configurazione, e **Applicazione Mobile**, che consente di usufruire delle funzionalità implementate sia dall'Interfaccia Web sia dal Riconoscitore Vocale. Una quarta interfaccia è data dai **Tag NFC**, ma di questa ci occuperemo successivamente poichè parte integrante dell'automazione.

Il **riconoscitore vocale**, derivato da un progetto accademico in collaborazione con Lorenzo Casanova chiamato *Sistema A.V.I.D. (Automated Voice Interactions Domotics)*, è un **riconoscitore online** che consente, tramite un set di comandi definiti dall'utente o preimpostati nel sistema, di eseguire delle azioni specifiche sia all'interno che all'esterno dell'ambiente casalingo. Questo riconoscitore ha due componenti: una detta "*ambientale*", dove il riconoscimento del comando avviene in una zona ben definita all'interno dell'ambiente casalingo, ed uno *mobile*, utilizzando il riconoscitore vocale dell'applicazione mobile che verrà trattata successivamente.

Il riconoscitore vocale ambientale è la componente sia **software** che **hardware** con il compito di **riconoscere i comandi dell'utente** impartiti dal sistema **nell'ambiente casalingo**. Questo riconoscitore è composto principalmente da una *scheda audio USB* per l'input vocale, poichè il Raspberry Pi non ha nella scheda audio

incorporata una porta di ingresso audio ma solo una in uscita, e da un *microfono*, possibilmente ambientale poichè consente una migliore qualità di registrazione. Questo riconoscitore è stato implementato attraverso script in linguaggio *Python*,

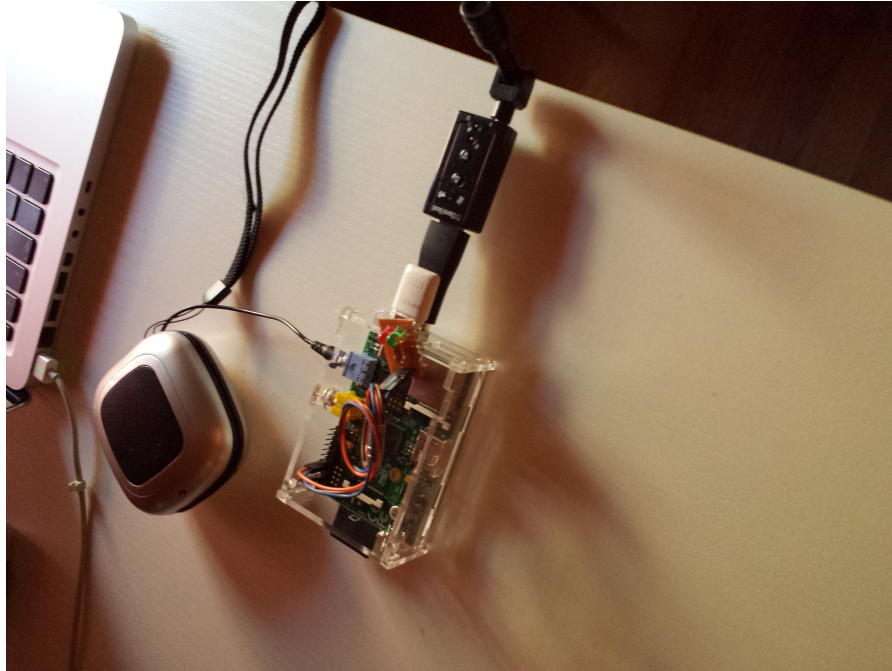


Figura 3.1: Riconoscitore ambientale con Raspberry Pi Model B

utilizzando programmi che ne eseguono il riconoscimento attraverso le *Speech Recognition API V2 di Google*. La figura 3.1 mostra l'implementazione del riconoscitore ambientale su Raspberry Pi Model B.

Il riconoscitore vocale mobile consente di usufruire delle funzionalità di riconoscimento in mobilità, direttamente dal proprio smartphone. Questo svolge le stesse operazioni ed ha le stesse funzionalità di quello ambientale, ma consente all'utente di utilizzarlo al di fuori dell'ambiente casalingo utilizzando l'apposita applicazione mobile. Anch'esso utilizza *Google API* per il riconoscimento, precisamente utilizza un plugin per il framework **Phoneygap** che si interfaccia con il servizio **Google Now**; l'applicazione mobile è implementata utilizzando il framework **Phoneygap**, con l'ausilio di *HTML5*, *CSS3*, *Javascript*, *jQuery Mobile*.

Entrambi i riconoscitori si avvalgono dell'ausilio del **file di configurazione** che contiene al suo interno i comandi da eseguire attraverso una strutturazione $\langle \text{chiave} \rangle, \langle \text{valore} \rangle$, in cui si ha: *command*, che indentifica un comando vocale, ed *exec*,

che identifica l'azione da eseguire rispetto a quel comando. Per effettuare il riconoscimento le due soluzioni, attraverso le richieste di speech recognition utilizzando le *Google Speech API*, restituiscono in formato testuale il riconoscimento vocale ed inviano una richiesta direttamente al web server che si occupa, con il programma chiamato *interpreter*, di trovare ed eseguire il comando inserito all'interno del file di configurazione. Il programma che gestisce la speech recognition esegue il controllo di esistenza del comando richiesto ed in caso affermativo lo esegue, altrimenti restituisce un messaggio di errore di mancata esecuzione. Un esempio può essere “*accendi luci salone*” oppure “*avvia musica*”, comandi di automazione per la gestione delle luci e della multimedialità. L'Implementazione di tipo online consente di avere un vantaggio derivato dal riconoscimento (quasi sempre corretto) del parlato grazie al servizio Google; tuttavia si presenta anche lo svantaggio di non poterlo usare in mancanza di connessione internet, poichè per effettuare il riconoscimento viene fatta richiesta direttamente ai server di Google.

L'**Interfaccia Web** consente l'utilizzo delle funzioni domotizzate attraverso un comune **portale web full responsive**, quindi ottimizzato per tutti i tipi di dispositivi utilizzando i temi messi a disposizione dal sito *HTML5UP*. Questa interfaccia consente di utilizzare le funzionalità implementate nel sistema domotico in modo semplice ed intuitivo, dando la possibilità, anche ad utenti meno esperti, di usufruire e configurare il sistema. A questo aspetto è stato dedicato molto tempo nella progettazione, con l'obbiettivo di rendere configurabile il sistema in modo semplice e veloce anche per gli installatori, cercando di porre l'utente comune allo stesso livello.

L'interfaccia web implementa tre funzionalità base: **la gestione delle luci**, con la rappresentazione astratta degli interruttori casalinghi che si riferiscono a specifici punti luce, la **termoregolazione** dell'ambiente, dando una interfaccia specializzata nella programmazione ed attivazione del termostato (precedentemente descritto), e la **videosorveglianza**, consentendo di poter attivare/disattivare la videosorveglianza e di visionare lo streaming in tempo reale. Inoltre consente, attraverso una pagina web, la configurazione del sistema.

La **gestione dei punti luce** avviene attraverso la rappresentazione degli interrut-

tori sotto forma di *bottoni slide*, interfaccia base per l'accensione e lo spegnimento. I singoli punti luce vengono identificati da un *label* che indica quale punto luce si vuole comandare; è stata scelta questo tipo di rappresentazione poichè più semplice e familiare nell'utilizzo quotidiano, oltre ad essere l'interfaccia minimale per questa funzionalità (mostrata nella figura 3.2).

La **gestione del termostato** o della termoregolazione consente di poter accedere

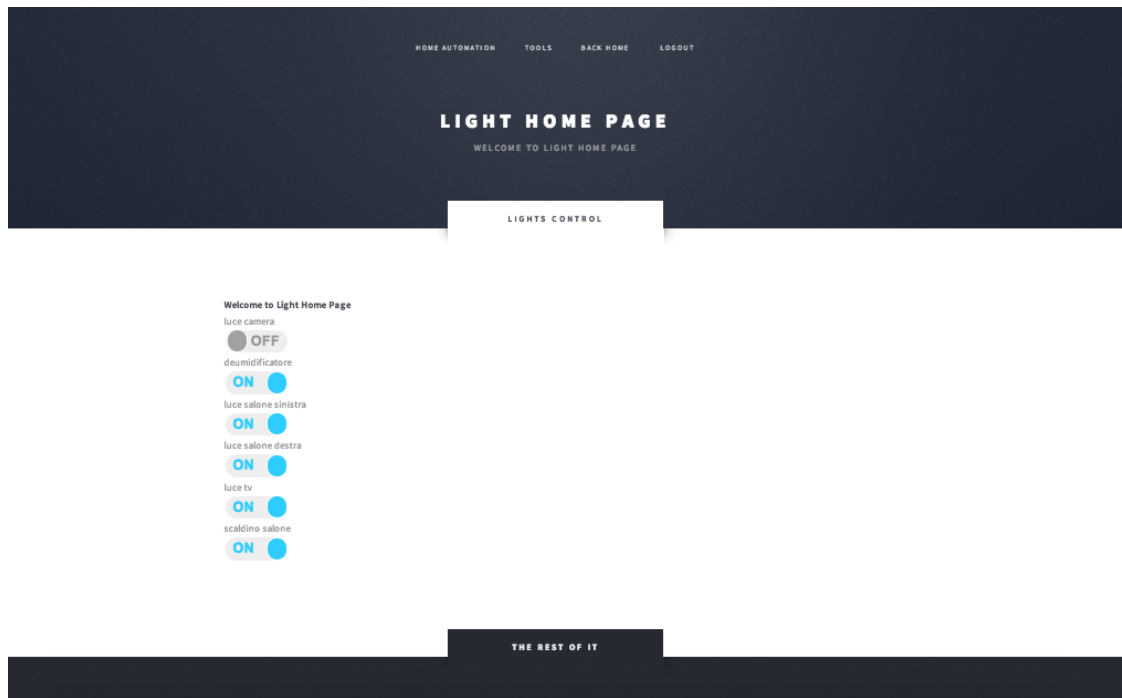


Figura 3.2: Gestore Luci nell'interfaccia web

alle funzionalità basilari, come accensione/spegnimento della caldaia ed impostazione del timer. L'interfaccia presenta una visualizzazione a *due colonne*, come si nota dalla figura 3.3: la prima colonna a sinistra consente di configurare il timer del termostato impostando, per ogni giorno della settimana, l'accensione e lo spegnimento e di visualizzare o eliminare il programma impostato; la seconda colonna a destra consente di vedere lo stato del termostato e di effettuare l'accensione o lo spegnimento, con la possibilità di impostare la temperatura desiderata. Tutti i dati che vengono visualizzati in questa interfaccia sono inerenti al termostato installato nella rete casalinga, la cui ubicazione può essere reperita attraverso il file di configurazione (IP del termostato nella sezione generale).

Riguardo alla **videosorveglianza**, il sistema consente di attivare o disattivare il

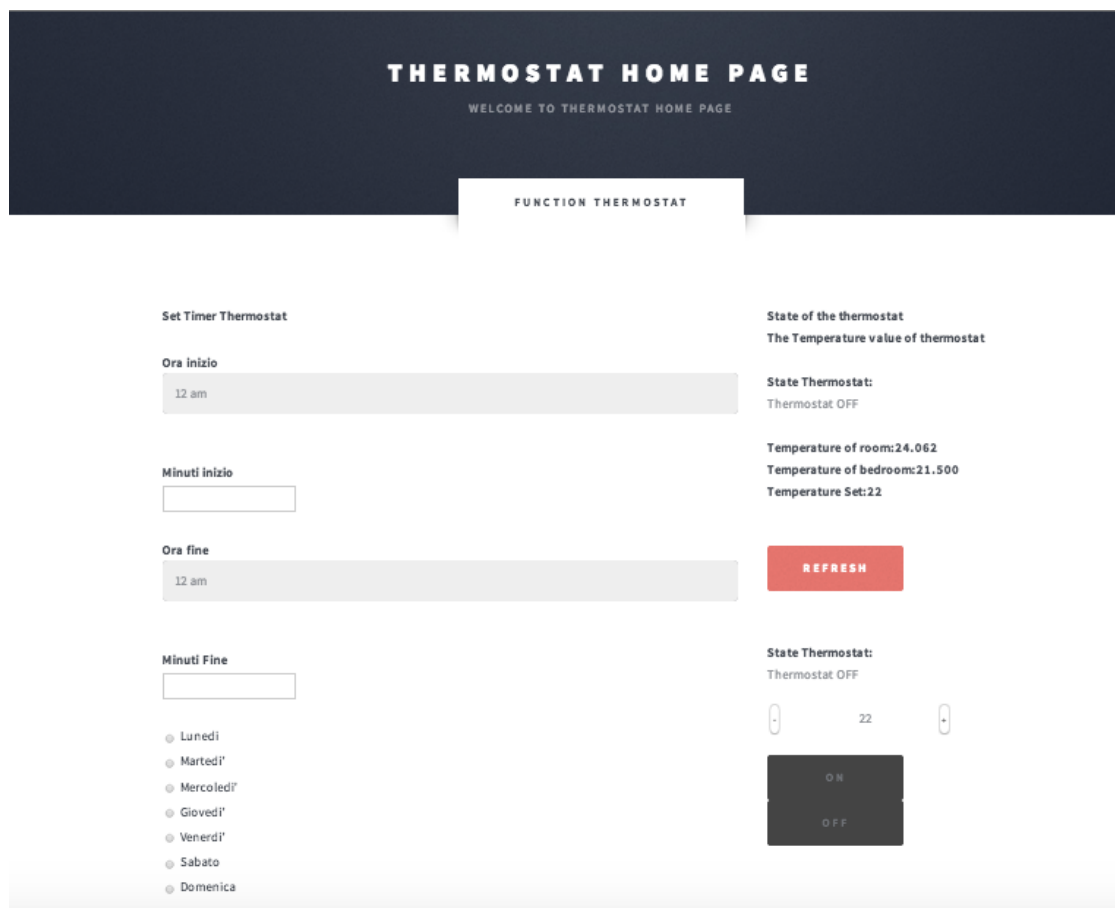


Figura 3.3: Gestione del termostato interfaccia web

servizio sia di *avviso di una intrusione* che di *streaming in tempo reale*. Per ragioni implementative i due servizi non possono essere fruiti simultaneamente, poichè si tratta di servizi differenti (trattati in seguito), perciò sono state utilizzate due pagine web di cui una contiene il servizio di streaming in tempo reale (un esempio è mostrato nella figura 3.4), mentre l'altra contiene la schermata dell'ultima variazione all'interno della stanza; entrambe le pagine contengono il controllore delle luci, utile in caso di scarsa luminosità ambientale, e la possibilità di ruotare la webcam attraverso il controllo di un servo motore. Infine l'ultima e più importante interfaccia per le funzionalità presentate dal sistema è l'**interfaccia di configurazione**, che consente di modificare il **file di configurazione** dove sono contenute le informazioni per le funzionalità del sistema. Questa interfaccia è stata implementata rendendola di facile utilizzo, cercando di evidenziare un workflow controllato e ben definito per ogni funzionalità. Come si può vedere dalla figura 3.7, l'interfaccia

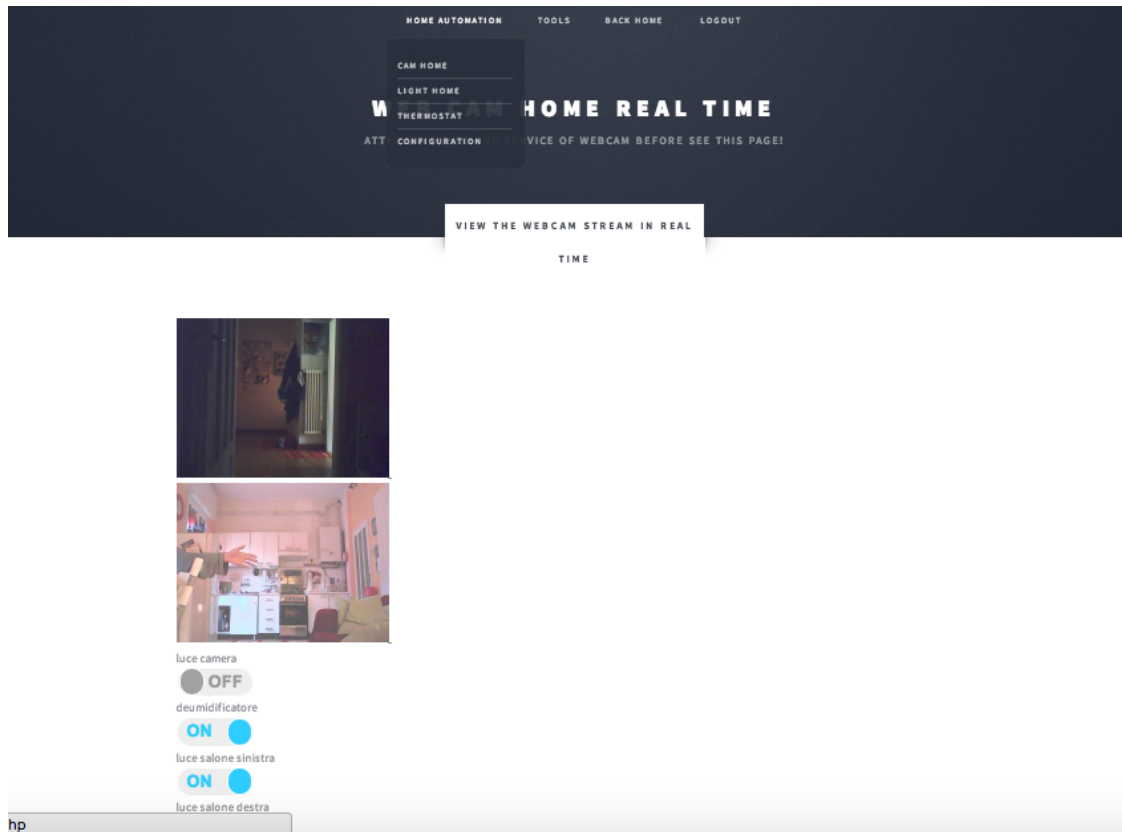


Figura 3.4: Gestione del termostato interfaccia web

presenta delle *sezioni*, corrispondenti a quelle contenute nel file di configurazione. Ogni sezione è specializzata rispetto ad ogni funzionalità implementata nel sistema, consentendo di modificare o inserire parametri a seconda del tipo di funzionalità, creando vincoli nell'utilizzo dettati dal tipo di impostazione; un esempio è la sezione contenente le impostazioni generali che possono solo essere modificate ma non eliminate o aggiunte poichè i parametri sono indispensabili per il funzionamento del sistema, mentre altre sezioni consentono l'aggiunta, modifica ed eliminazione dei parametri, come la sezione inerente le luci. Le modalità di inserimento e modifica dei parametri sono presentate attraverso dei *popup modali*, che chiedono all'utente di inserire il comando desiderato, come nella figura 3.6. La tipologia di parametri da inserire cambia a seconda della funzionalità presa in considerazione: per il gestore delle luci viene richiesto il nome della luce (*label*) e il *numero* della luce con l'eventuale indirizzo IP (facoltativo); per l'inserimento o modifica di un comando utente viene richiesto il comando e l'esecuzione corrispondente; per la modifica dei parametri principali e di quelli del media center è possibile solo inserire la modifica



Figura 3.5: Interfaccia web di configurazione per le luci



Figura 3.6: Popup di inserimento/modifica comandi nella configurazione

del valore, ma non di aggiungere nuovi campi. Ogni singola funzionalità dell'Interfaccia Web passa attraverso dei **messaggi asincroni** di tipo *POST* al web server utilizzando il framework *jQuery*, che attiva le primitive di normalizzazione della richiesta e richiama le utility per la modifica del file di configurazione situate nella

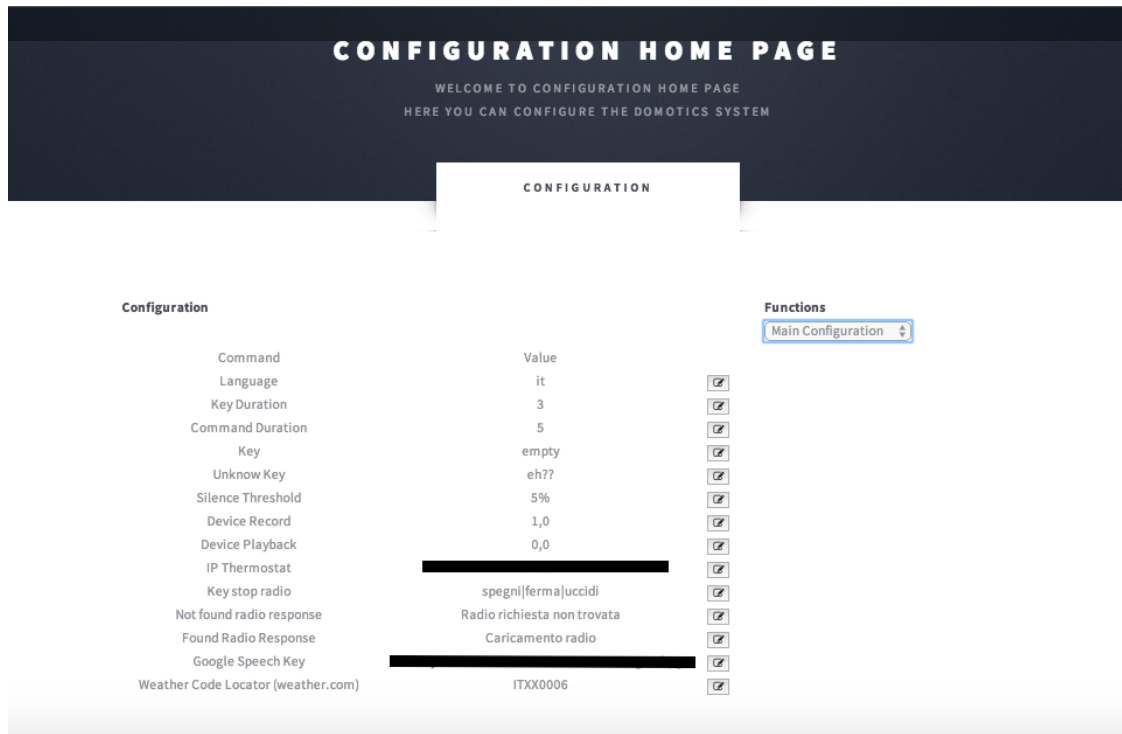


Figura 3.7: Interfaccia web di configurazione principale

cartella principale del sistema. I parametri di modifica ed inserimento vengono passati al web server attraverso queste chiamate asincrone, identificate da un *ID* di richiesta che specifica il tipo di modifica da effettuare nella sezione apposita del file di configurazione. Infine, l'interfaccia web presenta tutti i *meccanismi di sicurezza* implementati nel sistema, come gestore degli accessi tramite nome utente, password e visualizzazione dei log di accesso, mostrati in una pagina apposita. Il dettaglio di questi meccanismi verranno discussi successivamente.

Un'altra interfaccia molto importante è l'**Interfaccia Mobile**, dotata di una applicazione multi piattaforma che consente di utilizzare le funzioni implementate nell'impianto. L'applicazione è stata progettata per agevolare l'utilizzo delle componenti domotizzate, consentendo di usufruire delle funzionalità domotizzate, ma a differenza dell'interfaccia web *non è possibile modificarne le impostazioni*, essendo un vincolo progettuale dell'applicazione. L'interfaccia mobile si differenzia da quella web non solo per aspetto, come si uò vedere nella figura 3.13, ma anche per contenuti: aggiunge il controllore manuale del mediacenter (figura 3.11), con le funzioni di navigazione e di playback, aggiunge la possibilità di invocare il

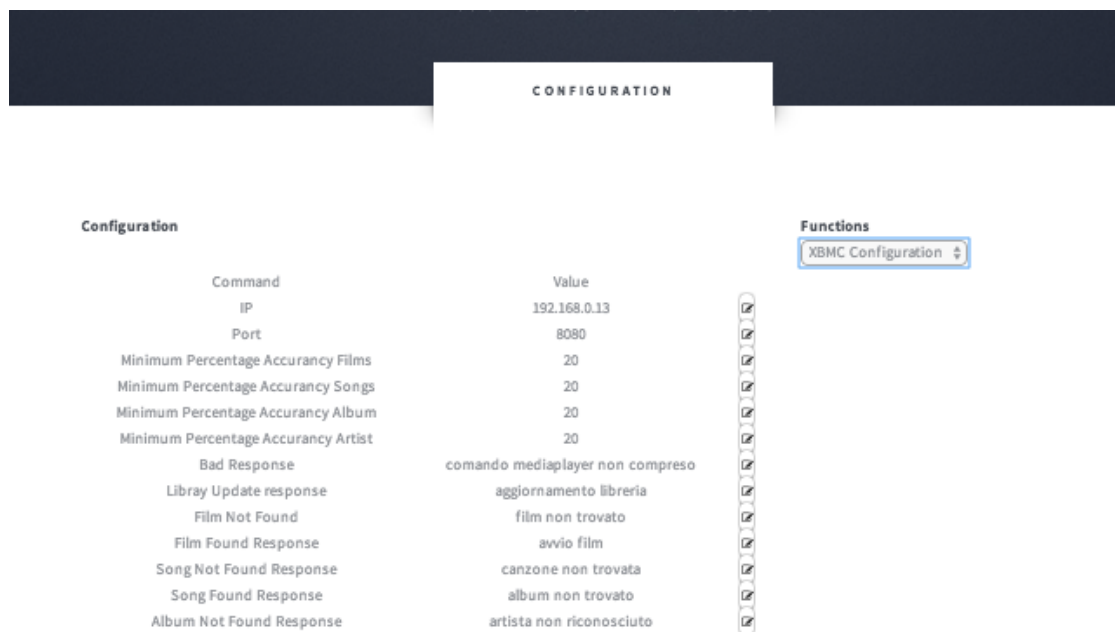


Figura 3.8: Interfaccia web di configurazione per il mediacenter

riconoscitore vocale descritto precedentemente, attivando le funzionalità vocali di Google Now; al contrario non è possibile modificare le impostazioni del sistema, funzionalità implementata solo all'interfaccia web, non implementa le funzioni di videosorveglianza e non implementa la gestione del timer del termostato. L'interfaccia mobile consente, ugualmente all'interfaccia web, di comandare i punti luce (figura 3.10) e il termostato (figura 3.9). Le differenze di funzionalità che offre l'interfaccia mobile rispetto a quella web è dovuta al fatto che l'applicazione mobile si interpone tra il riconoscitore vocale e l'interfaccia web, creando il connubio tra le due soluzioni. Per questo motivo la progettazione dell'interfaccia rispecchia i vincoli imposti per diversificarla dalle altre interfacce.

L'ultima interfaccia che discuteremo è data dai **TAG NFC**, che fanno parte sia delle interfacce che degli elementi di automazione. I Tag possono essere considerati delle interfacce poichè, attraverso la strutturazione del protocollo di comunicazione implementato in ogni dispositivo, è possibile configurare un TAG NFC per eseguire determinate azioni di automatizzazione. L'utente ha la possibilità di usufruire dei *TAG NFC* attraverso un lettore, che deve essere connesso alla rete domotizzata.

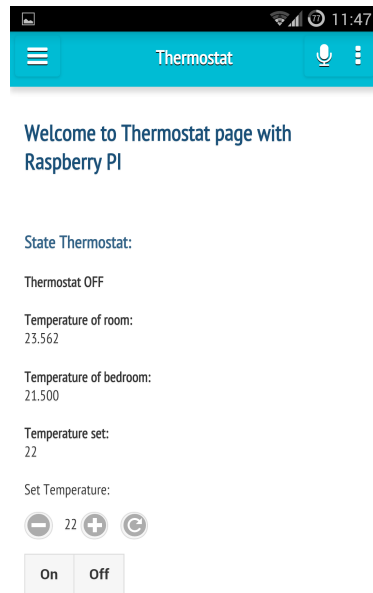


Figura 3.9: Controllare Termostato interfaccia mobile

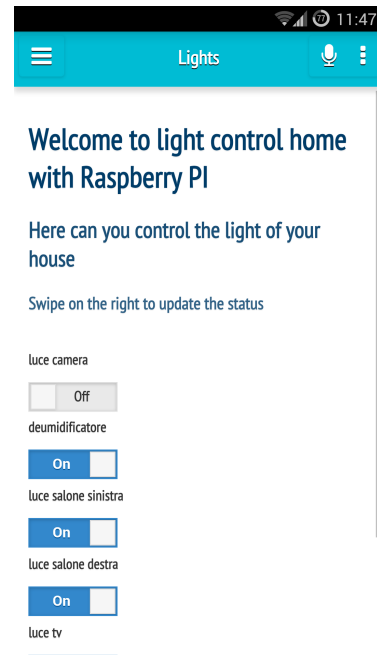


Figura 3.10: Controllare punti luce interfaccia mobile

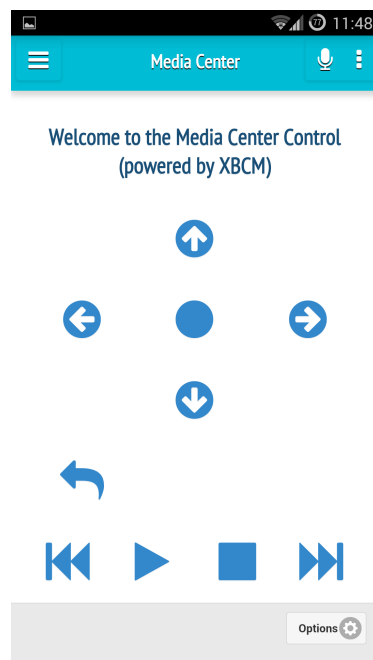


Figura 3.11: Controllare mediacenter interfaccia mobile



Figura 3.12: Configurazione applicazione mobile

Figura 3.13: Esempi di Interfacce Applicazione Mobile

Nell'implementazione del sistema il lettore è identificato da uno smartphone, che grazie ad un sensore NFC consente di eseguire le azioni impostate nel Tag, come ac-

censione dei punti luce oppure l'attivazione o disattivazione della videosorveglianza. L'implementazione con Tag sarà discussa nel capitolo successivo.

3.2.2 Web Server

Il web server è una componente importante del sistema, poicè ha il compito di gestire tutte le richieste da parte sia dei dispositivi che dalle interfacce. Questo è stato implementato utilizzando un'**unica logica di comunicazione**, sia per le interfacce, che richiedono l'esecuzione di azioni specifiche, sia per i dispositivi connessi alla rete che fanno richiesta delle funzionalità implementate. Le richieste fanno riferimento ad un servizio lato server che si occupa di gestirle rispetto alla componente del sistema, come il gestore delle luci o il termostato; questo servizio è implementato in ogni dispositivo e consente di gestire le richieste da parte dei dispositivi rispetto a tutte le funzionalità del sistema. Inoltre, il web server contiene le interfacce utente del portale web, discusse nella sezione precedente ed in cui sono riassunte tutte le funzionalità.

Sono stati utilizzati due tipi di server: **Apache 2.4** ed **nginx**. *Apache 2.4* è stato utilizzato per il nodo centrale mentre *nginx* per i dispositivi connessi nella rete. In una prima implementazione è stato utilizzato il server *Apache* in tutti i dispositivi per via della familiarità con le funzionalità del server, successivamente si è optato per l'utilizzo di *ngix* nei dispositivi connessi alla rete domotizzata, poichè molto più leggero rispetto ad *Apache* rispetto al carico di risorse. Infatti si consiglia di utilizzare *ngix* poichè il *Raspberry Pi* non ha una grande potenza di calcolo, alleggerendo l'esecuzione delle richieste.

L'implementazione dei meccanismi di gestione delle richieste per i servizi, la sicurezza riguardante l'accesso al sistema tramite credenziali e la tipologia di connessione, fanno parte delle funzioni che il webserver deve garantire. Questi meccanismi consentono l'accesso alle funzionalità dell'interfaccia web autenticandosi con *username* e *password*, tramite una connessione *HTTPS*. Il gestore delle credenziali è all'interno di un *Database MYSQL*, che contiene una tabella con le seguenti colonne:

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)        | NO   | PRI |          | auto_increment |
| username  | varchar(30)    | YES  | UNI | NULL    |                 |
| password  | varchar(30)    | YES  |     | NULL    |                 |
| name      | varchar(30)    | YES  |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+

```

in cui *id* identifica univocamente l'utente all'interno del DB, *username* e *password* sono le credenziali di accesso e *name* identifica il nome completo dell'utente. Ogni utente registrato viene loggato all'interno dell'interfaccia web ad accesso avvenuto, inserendo nella tabella *log* i relativi dati di login, che sono:

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| ip         | varchar(15)    | YES  |     | NULL    |                 |
| date      | datetime       | YES  |     | NULL    |                 |
| user_agent | varchar(100)   | YES  |     | NULL    |                 |
| user      | varchar(20)    | YES  |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+

```

dove *ip* indica l'indirizzo IP di riferimento con cui l'utente si è loggato, *date* indica la data ed ora di login, *useragent* indica lo useragent del browser con cui l'utente si è connesso e *user* indica il nome completo dell'utente loggato. Ad ogni login vengono create delle sessioni in cui sono contenuti i dati dell'utente e viene registrato l'accesso al sistema, per cui ogni pagina dell'interfaccia utente viene visualizzata soltanto se i dati sono congrui con l'avvenuto login, altrimenti viene ripresentata la pagina di login. Il web server crea sessioni di utilizzo di dieci minuti (figura 3.14), che risultano un tempo ragionevole per usufruire delle funzionalità del sistema. I meccanismi di sicurezza appena descritti fanno parte della gestione minima della sicurezza, i quali possono essere integrati con altri meccanismi come *OSAuth 2.0* per l'autenticazione del client, in modo da rendere l'accesso al sistema più sicuro. La comunicazione tra i dispositivi avviene attraverso richieste delle funzionalità desiderate direttamente al webserver di riferimento del dispositivo, che esegue il comando richiesto. I messaggi passati tra dispositivi sono richieste di tipo *GET* o *POST* per le quali viene specificata una *chiave d'accesso* alla funzionalità, imple-

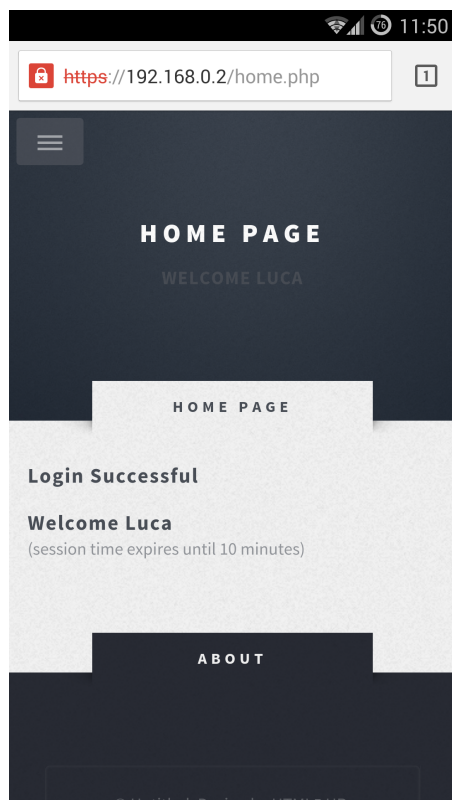


Figura 3.14: Homepage Interfaccia Web

mentato come meccanismo di sicurezza. Successivamente il web server si occupa di richiedere il servizio che controlla l'elettronica oppure di reindirizzare la richiesta al dispositivo connesso alla rete in modo che venga gestita. Le richieste *GET* sono principalmente per l'utilizzo dei *Tag NFC*, poichè all'interno di questi non è possibile scrivere la richiesta di tipo *POST*, risultando un limite importante; per questo motivo, le richieste *GET* possono essere fatte solo all'interno della rete casalinga, ma non all'esterno di essa.

Infine il Web Server implementa il meccanismo di gestione delle richieste dall'esterno dell'ambiente casalingo, presentando unicamente l'interfaccia web attraverso il dispositivo centrale o un dispositivo designato e bloccando l'accesso a tutti gli altri dispositivi ed i relativi servizi, che possono essere fruiti solamente dall'interfaccia web esposta. I vincoli che il web server presenta rispetto a questa funzionalità sono:

- impossibilità di accesso al codice sorgente delle richieste
- impossibilità di accesso alle risorse condivise tra i dispositivi

- impossibilità di accesso diretto ai dispositivi della rete domotica
- impossibilità di fare richieste di tipo GET
- fruizione dei contenuti e delle funzionalità solamente tramite interfaccia web

3.2.3 Il Gestore delle Luci

Cominciamo ora a descrivere nel dettaglio l'implementazione dei componenti domotizzati. In questa sezione parliamo del gestore delle luci, primo componente implementato e progettato per il sistema domotico.

Il **gestore delle luci** fornisce quelle funzionalità per l'accensione e lo spegnimento dei punti luce all'interno dell'ambiente casalingo e presenta due componenti: una di *comunicazione con l'elettronica* ed una di *comunicazione con le interfacce*. Entrambe le componenti sono implementate attraverso dei *programmi scritti in linguaggio Python* che ne consentono la gestione.

La componente di comunicazione che permette di interfacciarsi con l'elettronica applicata al dispositivo, identificata da dei relay Arduino compatibili, è sia *software* che *hardware*: software poichè implementata attraverso degli script in Python utilizzando la libreria *RPI.GPIO* che consente di modificare lo stato delle GPIO del Raspberry Pi, sia in input che in output; *hardware* dovuta alla connessione wired tra il dispositivo e l'elettronica, utilizzando un unico bus di comunicazione direttamente collegato alla GPIO desiderata, attraverso tre cavi che si connettono al relay Arduino compatibile (*3v*, *GND* e *Signal*). La GPIO viene aperta in output in modo che venga azionato il relay a seconda dello stato desiderato (accesso o spento). All'interno dello script Python è stata inserita una mappatura delle *GPIO* disponibili, consentendo l'utilizzo di sei GPIO per questa funzionalità. Ad ogni numero, crescente da uno a sei, corrisponde una determinata *GPIO* e viene uno stato 1 o 0 per indicare l'accensione o lo spegnimento della luce. È stata progettata ed implementata questa soluzione al fine di: *semplificare l'installazione* da parte dell'utente, il quale si ritrova a collegare solamente un filo alla *GPIO* desiderata e poi scegliere attraverso un semplice numero quale adoperare; *semplificare la gestione da parte del web server*, il quale richiede l'azione per il numero scelto sen-

za indicare direttamente la *GPIO*, ed infine *unificare le comunicazioni con gli altri dispositivi*, i quali contengono tutti la stessa mappatura per azionare i punti luce di un determinato dispositivo. La chiamata agli script Python può essere eseguita nel seguente modo:

```
./managerLights.py 1 1
```

corrispondente all'accensione della luce in posizione uno. Per poter meglio comprendere il funzionamento di questi script, di seguito possiamo trovare l'esempio di codice per l'azionamento di una luce senza l'ausilio delle librerie *Python*:

```
#!/bin/bash

n_gpio=$1
value=$2
dir='/home/pi/.domotics/scripts/lights';

#usage
if [[ $# -eq 0 ]] ; then
    echo "Insert_number_GPIO";
    read n_gpio;
    echo "Insert_value_GPIO"
    read value;
fi

#control if does exist the directory corresponding the gpio
if [ ! -d "/sys/class/gpio/gpio$n_gpio" ];
then
    #echo "cartella non esiste"
    cd /sys/class/gpio/
    echo $n_gpio > export
    cd gpio$n_gpio;
else
    #echo "cartella esiste"
```

```

    cd /sys/class/gpio/gpio$n_gpio;
fi

#on off value
echo out > direction;
echo $value > value;

#control light's output
if [ $value -eq 1 ];
then

    echo 1 > $dir/value_$n_gpio
    echo "luce␣accesa";
fi

if [ $value -eq 0 ];
then
    #echo "chiudo cartella"
    #cd /sys/class/gpio/
        #echo $n_gpio > unexport
    echo 0 > $dir/value_$n_gpio
    echo "luce␣spenta"
fi

```

Come si può vedere, il codice ha come valore in entrata la *GPIO* desiderata e lo stato in output da emettere; questo script può essere richiamato direttamente da uno script in *Python* che legge il file di configurazione per reperire le informazioni necessarie, contenente la mappatura delle *GPIO* ed azionare l'interruttore dallo script sopracitato.

La comunicazione attraverso le interfacce è costituita da funzioni di comunicazione lato client che, tramite i parametri definiti nel file di configurazione, richiedono di accendere o spegnere la luce tramite i servizi lato server. Quando viene fatta richiesta dall'interfaccia utente, viene richiamata una funzione che tramite un mes-

saggio *HTML* di tipo *POST*, passa come parametro il numero di luce, l'eventuale indirizzo IP e lo stato 0 o 1. Di conseguenza il web server che gestisce la richiesta richiama la componente software per effettuare l'accensione o lo spegnimento del punto luce desiderato. Queste richieste sono *asincrone*, poichè permettono di non bloccare le funzionalità dell'interfaccia e consentono di dare uno stato del servizio, avviene se le luci sono accese o spente.

Come si può notare l'implementazione del sistema risulta di molto semplificata anche nella visualizzazione dell'interfaccia, grazie all'ausilio delle librerie Python.

Per punto luce implementato dal sistema è stato costruita una interfaccia di ge-

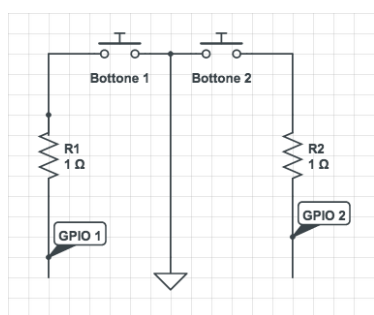


Figura 3.15: Circuito dei bottoni

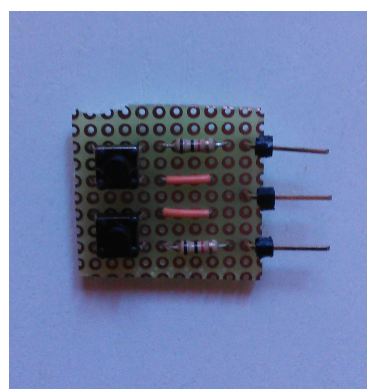


Figura 3.16: Prototipo dei pulsanti

stione attraverso dei pulsanti. Questi pulsanti (vedi figura 3.16) sono stati contruiti attraverso dei *bottoni push*, creando una interfaccia per la gestione “classica” dei punti luce, in caso di mal funzionamento o di situazioni comuni in cui non si necessita di utilizzare le funzioni evolute del sistema. Questi bottoni si interfacciano con le *GPIO* del *Raspberry Pi* attraverso dei programmi *Python* che leggono lo stato del bottone e, al conseguente cambiamento, ne eseguono l'azione di accensione e spegnimento. Lo schema elettronico dei bottoni è molto semplice (vedi figura 3.15), e sono state utilizzate delle resistenze e una piastra forata per la costruzione.

Infine per ogni punto luce è stato utilizzato un *relay*, a cui fanno capo *una presa maschio* ed *una presa femmina*, collegate direttamente alla presa elettrica. Questa scelta progettuale è necessaria per una ragione importante: se si dovesse inserire il *relay* all'interno dell'impianto elettrico preesistente bisognerebbe rifare la certificazione energetica dell'impianto, mentre con il suddetto metodo consente di collegare

i punti luce desiderati ed anche apparecchi elettronici, consentendo una portabilità del sistema.

3.2.4 Il Termostato

Il termostato è la componente domotizzata che consente di controllare e gestire la termoregolazione dell'ambiente casalingo. Il termostato implementato è un dispositivo *completamente autonomo*, composto da un **Raspberry Pi Model A** e da quattro componenti di elettronica: un **relay**, un **display LCD 16x2**, dei **bottoni** ed un **sensore di temperatura DS18B20**, come mostrato nella figura 3.17 come primo prototipo.



Figura 3.17: Prototipo di termostato implementato nel sistema

La scelta di usare un *Raspberry Pi Model A* risiede nella **ridotta complessità** di questo dispositivo, dal **basso costo** rispetto al *Model B*, dall'utilizzo di questo dispositivo soltanto in **modalità' wireless** e dall'inutilità di avere due porte USB; di fatto questa soluzione consente di abbassare ulteriormente i costi di realizzazione ed evitare sprechi di risorse.

La strutturazione del dispositivo è davvero minimale, come anche il suo utilizzo. Esistono due interfacce per poter usare il dispositivo: una *interfaccia software*, fruibile dall'interfaccia web o dall'applicazione mobile precedentemente descritte, e una *interfaccia fisica*, data dal display e dai bottoni che eseguono determinate azioni. L'**interfaccia fisica** è una interfaccia semplice e minimale, dove vengono visualizzate le informazioni come lo stato del termotato, la temperatura attuale della stanza e la temperatura impostata tramite il display *LCD 16x2* (vedi figura 3.18). I bottoni danno la possibilità di poter eseguire quattro specifiche azioni: **accensione del display**, **accensione/spegnimento della caldaia**, **rimozione o reimpostazione timer** e **spegnimento del termostato**. Questa interfaccia è stata progettata allo scopo di poter dare all'utente una interfaccia comune a questo tipo di dispositivi, oltre alla possibilità di utilizzare il dispositivo senza dover passare dall'interfaccia web per le funzionalità minime.

L'interfaccia software è esattamente quella descritta nella sezione riguardante le interfacce, dove sono presenti le impostazioni del dispositivo e la possibilità di comandarlo a distanza. Questa è la più importante, poichè consente di usufruire di tutte le funzionalità che il termostato offre.

Il funzionamento del termostato è molto semplice: l'utente imposta una determinata temperatura limite, alla quale il termostato fa riferimento; una volta acceso il servizio, il termostato attiva il *relay* collegato direttamente alla caldaia tramite cavo che attiva il circuito della caldaia, attendendo il raggiungimento della temperatura desiderata che viene rilevata attraverso il sensore di temperatura posto nel termostato o interfacciandosi con i sensori sparsi nella casa; al raggiungimento della temperatura impostata, il termostato chiude il circuito spegnendo il *relay* e la caldaia si spegne. Per effettuare tutte queste operazioni sono stati implementati dei *programmi scritti in linguaggio Python* che rilevano la temperatura comunican-



Figura 3.18: Dettaglio della visualizzazione delle informazioni nel display del termostato

do con il sensore *DS18B20* attraverso apposite librerie kernel per *sensori 1-wire* ed attivare o disattivare il relay. Il sensore *DS18B20* è un sensore digitale che si interfaccia con la libreria *1-wire* del *Raspberry Pi*. Per poter utilizzare questo sensore occorre abilitare il modulo kernel nel seguente modo:

```
sudo modprobe w1-gpio  
sudo modprobe w1-therm
```

Successivamente, la rilevazione della temperatura consiste nella lettura di un

file all'interno del filesystem del Raspberry Pi. Un esempio di codice per la lettura del valore della temperatura è il seguente, preso dal sito Adafruit [3]:

```
import os
import glob
import time

os.system('modprobe_w1-gpio')
os.system('modprobe_w1-therm')

base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c, temp_f

while True:
    print(read_temp())
```

```
time.sleep(1)
```

Come si può notare il programma legge costantemente il valore della temperatura direttamente dal file *w1_slave*, contenente la temperatura rilevata dal sensore. Il sensore è posto nella parte bassa del dispositivo, poichè il calore generato sia dall'ambiente che dal dispositivo tende ad andare verso l'alto, falsando la rilevazione della temperatura. Il collegamento del sensore con il Raspberry Pi è molto sempli-

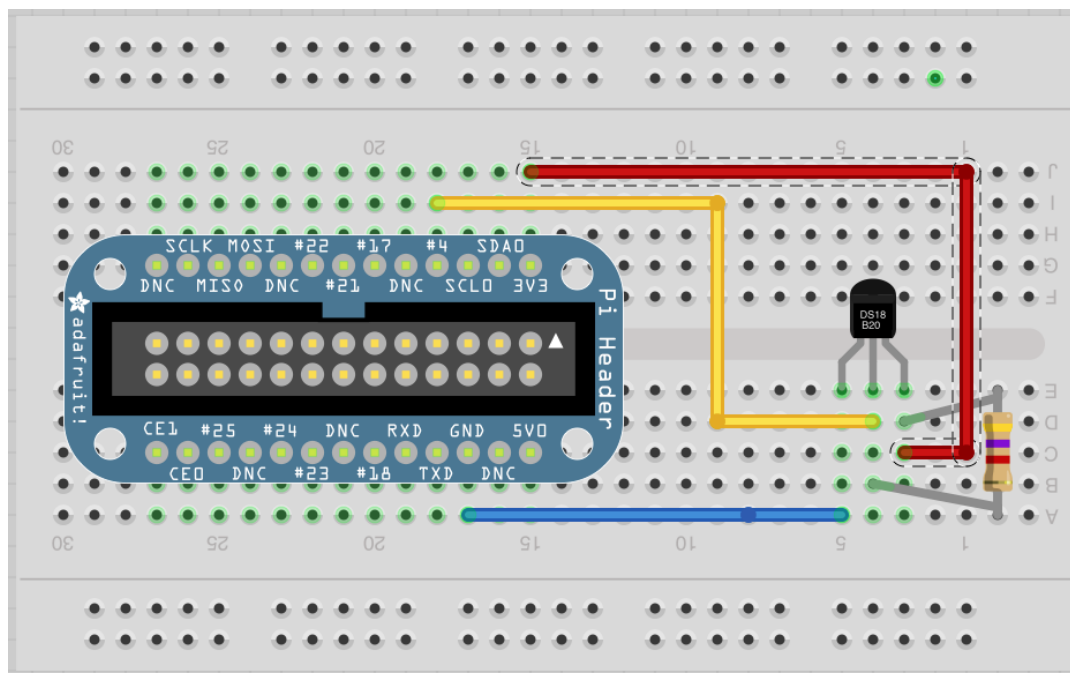


Figura 3.19: Dettaglio della connessione del sensore DS18B20 (powered by Adafruit)

ce, ed utilizza la GPIO 4 per la connessione con dispositivi *1-wire* e una resistenza da $4.7\ \omega$ (vedi figura 3.19)

Il programma che si occupa dell'accensione della caldaia, chiamato *thermostat.py*, è un processo avviato dall'utente oppure dal timer impostato, rimanendo in un loop fino a che non viene richiesto lo spegnimento. Questo loop rileva costantemente la temperatura e ne confronta il valore impostato dall'utente, determinando l'accensione o lo spegnimento della caldaia. Ai fini di raccolta dati vengono salvate l'ora e la data di spegnimento o accensione in un apposito database, in modo da tener traccia delle operazioni. Il programma di spegnimento, chiamato *stop_thermostat.py*, invia un segnale di arresto a *thermostat.py*, che provvede a spegnere il *relay* e a chiudere il canale comandato dalla *GPIO*. Entrambi i programmi vengono richia-

mati dalle due interfacce, che utilizzano questi script.

Per implementare la funzione di timer è stato utilizzato il programma *crontab*, che permette di avviare programmi o script a determinati giorni del mese che della settimana. «I file *crontab* contengono la lista dei job e altre istruzioni per il demone di cron. Gli utenti possono avere dei file *crontab* individuali e spesso ci sono dei file *crontab* a livello di sistema (solitamente in */etc* o in una sottodirectory di */etc*) che possono essere utilizzati, ma modificati solo dagli amministratori di sistema. Ogni linea di un file *crontab* segue un formato particolare, composta da una serie di campi separati da spazi o tabulazioni. I campi possono avere un solo valore o una serie di valori.»[4] Alla programmazione del timer, che avviene attraverso l'interfaccia web, vengono impostate le righe del *crontab* in base all'ora di accensione e all'ora di spegnimento nel seguente modo:

```
30 07 * * 6-7 sudo /home/pi/thermostat/thermostat.py
30 08 * * 6-7 sudo /home/pi/thermostat/stop_thermostat.py
00 06 * * 1-5 sudo /home/pi/thermostat/thermostat.py
30 07 * * 1-5 sudo /home/pi/thermostat/stop_thermostat.py
```

viene indicata in questo caso l'accensione della caldaia tramite lo script *thermostat.py* alle ore 07:30 di sabato e domenica e alle ore 06:00 dal lunedì al venerdì, e lo spegnimento alle 08:30 il sabato e la domenica e alle 07:30 dal lunedì al venerdì. La sintassi del file *crontab* è la seguente[4]:

```
----- [m]inute: minuto (0 - 59)
| .----- [h]our: ora (0 - 23)
| | .----- [d]ay [o]f [m]onth: giorno del mese (1 - 31)
| | | .----- [mon]th: mese (1 - 12) OPPURE jan,feb,mar,apr...
| | | | .---- [d]ay [o]f [w]eek: giorno della settimana (0 - 6) (domenica=0 o
| | | | | 7) OPPURE sun,mon,tue,wed,thu,fri,sat
| | | | |
* * * * * comando da eseguire
```

Questo programma è stato utilizzato per la funzione di timer poichè risulta lo strumento più adatto senza dover creare un programma apposito per gestire questa funzionalità. Tuttavia vi è un limite: *crontab* può essere eseguito solo su *Raspberry Pi* e su sistemi *Unix Like*, quindi in caso di implementazione del dispositivo su

altre piattaforme come Arduino necessita della creazione di una utility apposita. Un punto di forza del termostato domotizzato è la possibilità di poter gestire la termoregolazione a distanza, anche al di fuori dell'ambiente casalingo. Questo aspetto è molto importante, sia per un utilizzo più gradevole, consentendo di accendere e spegnere la caldaia direttamente dal cellulare oppure impostando un timer di accensione, sia per risparmi energetici, conferendo all'utente maggiore libertà e totale controllo dell'accensione e dei relativi consumi. Inoltre, il dispositivo, grazie alla connessione wireless, può essere connesso direttamente alla rete e quindi posizionato in qualsiasi punto della casa.

3.2.5 La Videosorveglianza

La videosorveglianza è il componente che gestisce gli aspetti di sicurezza dell'ambiente casalingo descritto nei paragrafi precedenti, per la quale sono stati utilizzati due servizi principali: uno di **avviso delle intrusioni** non autorizzate in tempo reale, attraverso il programma **motion**, e uno di **streaming in tempo reale**, attraverso il programma **mjpg-streamer**. Entrambi i servizi si avvalgono dell'utilizzo di una webcam collegata alla porta USB del *Raspberry Pi*.

Motion rileva i movimenti all'interno dell'ambiente utilizzando la webcam come sorgente di input, consentendo non solo di catturare l'immagine del relativo movimento, ma anche di eseguire una determinata azione corrispondente al rilevamento avvenuto, tramite un programma o uno script. Il programma è altamente configurabile ed è stato impostato in modo che ad ogni variazione rilevata nell'ambiente venga catturata l'immagine ed inviata tramite con un messaggio email generato da uno script indicante l'ubicazione della webcam e la relativa immagine, come mostrato nella figura 3.20 .

Il servizio rimane attivo fino a che l'utente non decide di fermarne l'esecuzione. Dall'utilizzo di questo programma si ha il vantaggio principale di non dover implementare dei meccanismi di rilevamento dei movimenti attraverso appositi sensori di movimento, ed inoltre viene inviato quasi istantaneamente un messaggio di avviso all'utente consentendo di visualizzare lo stato in tempo reale e di effettuare un pronto intervento. Tuttavia si presenta anche uno svantaggio, dovuto allo spam-

ATTENTION!!! Cam Detection at Tue Nov 11 08:24:44 CET 2014 on BEDROOM!

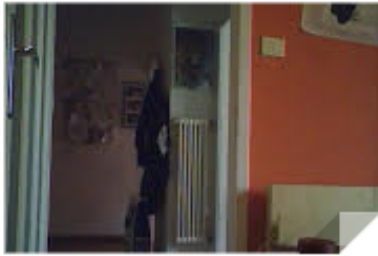


Figura 3.20: Esempio di messaggio di allerta da parte del sistema di videosorveglianza (Motion)

ming di email generato dal servizio, rimanendo la possibilità di di impostare il programma per inviare una email al cambiamento dello stato ambientale, può essere impostato solo l'intervallo di attesa tra un rilevamento e l'altro. Quest'ultimo aspetto è un limite intrinseco del programma dovuto all'utilizzo del programma soltanto in dispositivi *Unix Like*. Un esempio di configurazione del servizio è il seguente:

```
daemon=on
norm=0
input=8
auto_brightness=on
framerate=1
gap=2
height=480width=640
#jpeg_filename=motion/%Y-%m-%d-%H:%M:%S
jpeg_filename=viewcam
#lightswitch 20
locate=on
minimum_frame_time=1
minimum_gap seconds=2
noise_level=32
noise_tune=off
output_all=off
```

```

output_motion=off
output_normal=on
#on_event_start=echo ciao
on_motion_detected=sendmail
quality=75
snapshot_interval=300
target_dir=/var/www
text_left= Cam Home Bedroom
text_right=%d %b %Y\n%k:%M:%S
threshold=2000
threshold_tune=off
v4l2_palette=6
videodevice=/dev/video0
webcam_localhost=off
webcam_maxrate=2
#webcam_port=8080
wecam_quality=85

```

Una ampia documentazione può essere reperita dal sito di Motion. Molto importante è la qualità dell'immagine che il sistema offre: per via dell'utilizzo delle webcam, la qualità dell'immagine è stata impostata ad un fotogramma di risoluzione *480x640 pixels*, buon connubio tra qualità e leggerezza dell'immagine.

Il servizio di **streaming in tempo reale** utilizza invece un programma diverso denominato **MJPEG-Streamer**. Il programma consente di inserire lo streaming in tempo reale direttamente in una pagina web, attraverso le librerie *Linux-UVC*, come mostrato nella figura 3.4. «*MJPEG-streamer takes JPGs from Linux-UVC compatible webcams, filesystem or other input plugins and streams them as MJPEG via HTTP to webbrowsers, VLC and other software. It is the successor of uvc-streamer, a Linux-UVC streaming application with Pan/Tilt*»[5] Entrambi i servizi descritti vengono visualizzati nell'apposita sezione dell'interfaccia web.

Il motivo della scelta dell'utilizzo di questi due servizi risiede nelle loro funzionalità: entrambi forniscono uno streaming in tempo reale, ma solo *MJPEG-Streamer* consente di incorporare lo streaming all'interno di una pagina web, senza l'ausi-

lio di plugin aggiuntivi. Unica limitazione è l'impossibilità di utilizzare entrambi i servizi simultaneamente, poichè riservano l'utilizzo del canale di comunicazione con la webcam in modo esclusivo; l'unico modo di aggirare questa problematica è l'utilizzo di una webcam secondaria, ma i consumi energetici non permetterebbero il corretto funzionamento del sistema dato l'eccessivo assorbimento di entrambe le webcam collegate al *Raspberry Pi*. Per questo motivo si è optato per separare le esecuzioni dei due servizi, inserendo un elemento di automazione che verrà discusso successivamente. Riguardo al tema dei consumi energetici, per evitare il sovraccarico la webcam è collegata al Raspberry Pi per mezzo di un *HUB USB alimentato* e non auto alimentato.

3.2.6 Media Center

Il media center è la componente multimediale del sistema che utilizza il famoso media center **XBMC** (figura 3.21), pluripremiato programma open-source con licenza GPL che consente di usufruire delle funzioni di entertainment e di media player multi piattaforma. Consente di riprodurre la maggior parte di video, musica ed altri contenuti multimediali attraverso una vasta gamma di codec e di interfacciarsi con supporti multimediali locali oppure connessi alla rete, attraverso i protocolli di condivisione più utilizzati. Consente inoltre l'implementazione e l'inserimento di plugin per poter incrementare le funzionalità del sistema.

Il programma è compatibile anche con la piattaforma ARM, esattamente quella utilizzata nel *Raspberry Pi*. Esistono diversi sistemi che offrono le funzionalità di mediacenter con XBMC, implementate appositamente per Raspberry Pi tra cui: *OpenElec*, progetto open-source sotto forma di distribuzione Linux, *RaspBMC* e *Xbian*, entrambe basate su Raspbian 7.0. Tra le tre distribuzioni sopracitate risulta più idonea *RaspBMC* per l'utilizzo all'interno del sistema domotizzato. Di seguito le ragioni:

- È basata su *Raspbian*, che consente di installare tutti i programmi necessari per le funzionalità implementate.



Figura 3.21: Interfaccia principale di XBMC

- Possibilità di interfacciamento con le *GPIO*, grazie alle librerie implementate su *Raspbian*.
- Integrazione con i programmi utilizzati e possibilità di aggiornamento del sistema.
- maggior leggerezza e stabilità rispetto alle altre distribuzioni; basti pensare che *Xbian* è ancora in versione Beta.

Nel sistema implementato è stato designato un dispositivo apposito, precisamente un *Raspberry Pi Model B*, che esegue *RaspBMC* per le funzionalità di mediacenter ed implementa il controllore delle luci e la videosorveglianza della stanza in cui è installato. Grazie alle funzionalità descritte ed alla scalabilità del sistema, il dispositivo può essere utilizzato indipendentemente, connesso ad un nodo centrale oppure essere eletto a sua volta come nodo centrale.

Le funzionalità di media center implementate nel sistema, derivate dalla implementazione del sistema *A.V.I.D.* in collaborazione con Lorenzo Casanova, consentono di comandare le funzioni base del media center, come la navigazione all'interno dei menu, funzioni di playback e aggiornamento delle librerie di musica e video (per la descrizione dell'utilizzo di queste funzionalità si rimanda ai paragrafi precedenti). L'implementazione è stata effettuata attraverso degli script *Python* che

si interfacciando, dato l'indirizzo IP del dispositivo, con le API fornite da *XBMC*.

Un esempio di utilizzo di queste API è dato dal seguente codice:

```
# *****
# UNDERSTAND REQUEST TYPE, ITEMNAME and SETUP DATA
# *****
reqToDo = 1
printResp = ""
if userInput == "pauseToggle":
    userReq = {"jsonrpc": "2.0", "method": "Input.ExecuteAction", "params": {"
        action": "pause"}, "id": 1}
    printResp = "Player:␣toggle␣pause"
elif userInput == "stop":
    userReq = {"jsonrpc": "2.0", "method": "Input.ExecuteAction", "params": {"
        action": "stop"}, "id": 1}
    printResp = "Player:␣stop"
elif userInput == "skipNext":
    userReq = {"jsonrpc": "2.0", "method": "Input.ExecuteAction", "params": {"
        action": "skipnext" }, "id": 1}
    printResp = "Player:␣next"
elif userInput == "skipPrevious":
    userReq = {"jsonrpc": "2.0", "method": "Input.ExecuteAction", "params": {"
        action": "skipprevious"}, "id": 1}
    reqToDo = 2 # need to do double back to go to previous item
    printResp = "Player:␣previous"
elif userInput == "restart":
    userReq = {"jsonrpc": "2.0", "method": "Input.ExecuteAction", "params": {"
        action": "skipprevious"}, "id": 1}
    printResp = "Player:␣restart"
else:
    print "ERROR:␣Unknown␣user␣command"
    sys.exit(0)

#print "DEBUG: userReq:", userReq

# *****
# HTTP REQUEST: send command
# *****
# set raspiMediaPlayer URL
url = 'http://'+xbmcIP+':'+xbmcPORT+'/jsonrpc'
# set request HEADER
headers = {'content-type': 'application/json'}
# execute request
payload = userReq
```

```
print printResp
while reqToDo > 0:
    response = requests.post(url, data=json.dumps(payload), headers=headers)
    reqToDo = reqToDo -1
    if reqToDo > 0:
        time.sleep(1)
```

Il seguente codice mostra come, attraverso la costruzione di un oggetto json che si interfaccia con le API di XBMC, si effettuino le richieste di play, forward, pause, stop e backward. Di seguito alcune delle tante funzioni implementate:

- Playback brano musicale specifico.
- Playback brani musicale in modalità shuffle.
- Playback di un specifico film.
- Aggiornamento ed eliminazione libreria Musica e Film.

Tutte queste funzioni possono essere fruite sia attraverso il riconoscitore vocale, sia ambientale o mobile, sia attraverso l'applicazione mobile, vedi figura 3.11. Entrambe le interfacce inviano una richiesta al web server della funzionalità, attraverso il protocollo di comunicazione che indentifica il tipo di richiesta ed esegue il comando attraverso gli script progettati. La diversità risiede unicamente nell'interfaccia utilizzata, descritte nel paragrafo precedente.

3.2.7 Applicazione Mobile

L'applicazione mobile è stata descritta nella sua interfaccia nei paragrafi precedenti, ora descriveremo la sua implementazione. L'applicazione fornisce l'interfaccia per le funzionalità implementate nel sistema domotico attraverso una applicazione per dispositivi mobili Android, utilizzando il framework *Phonegap 2.9.1* con l'ausilio dei linguaggi *HTML5*, *CSS3* e *Javascript* con il framework *jQuery Mobile 1.4.5*. La scelta di utilizzare questo *framework* risiede in diversi fattori:

- Il framework è multiplatforma: consente di implementare l'applicazione per la maggior parte dei dispositivi mobile come *Android* ed *iOS*.

- Bassa complessità: non si necessitano di tutte le risorse del dispositivo e quindi sono sufficienti le funzionalità implementate dal framework.
- Semplicità di implementazione: il *framework* utilizza linguaggi lato client come *HTML5* e *CSS3*, consentendo una facile e veloce implementazione senza dover apprendere le *SDK* di ogni singola piattaforma.
- Free: il framework si può utilizzare gratuitamente, senza dover pagare licenze come per le *SDK* native.

L'applicazione implementa dei messaggi asincroni utilizzando una comunicazione *HTTPS* con chiamate *POST* al web server, a cui vengono inviati i dati relativi all'esecuzione della richiesta. Questi messaggi sono implementati attraverso le primitive del *framework jQuery*, che consente di strutturare le chiamate asincrone in modo semplice. L'esempio di sintassi è dato dal seguente codice.

```
function onOffLight(num,change){
  var external_ip="";
  if (num.indexOf('192.168') != -1){
    var res = num.split("_");
    num=res[0];
    external_ip=res[1];
  }
  if (change){
    $.ajax({
      type:"POST";
      url:"http://" + ip_server + "/home_light.php",
      data: {user: "Home", password: "", light: num, status_light: "1", external:
        external_ip},
      dataType: "text"
    })
  }
  else{
    $.ajax({
      type:"POST",
      url:"http://" + ip_server + "/home_light.php",
      data: {user: "Home", password: "", light: num, status_light: "0", external:
        external_ip},
      dataType: "text"
    })
  }
}
```

}

In questo codice è racchiusa la funzione di accensione e spegnimento dei punti luce, facendo la richiesta direttamente al web server dell'accensione o lo spegnimento della luce. Nell'applicazione sono salvati gli indirizzi IP del server di riferimento del dispositivo centrale oppure del dispositivo specifico, attraverso un database condiviso. Successivamente vengono inviati nome utente e password che identificano le credenziali di accesso al sistema per la sicurezza, il numero di luce e lo stato desiderato.

L'applicazione consente di impostare i parametri di funzionamento ed interfacciamento con il sistema, inserendo i parametri per la comunicazione. Al primo avvio vengono richiesti all'utente l'indirizzo IP del nodo centrale o di riferimento e l'indirizzo IP del termostato. Questi vengono memorizzati nel database e possono essere modificati successivamente attraverso la pagina di impostazioni.

3.2.8 Altri Servizi

Grazie alla piattaforma *Raspberry Pi* e all'utilizzo di software implementati per il sistema operativo *Linux Debian* è possibile aggiungere su un dispositivo che fa da nodo centrale della rete ulteriori servizi. Nel nodo centrale sono stati aggiunti servizi come:

- **Codivisione dei file attraverso server SAMBA:** consente, attraverso il suddetto protocollo, di condividere cartelle e stampanti nella rete.
- **Accesso remoto tramite protocollo SSH:** consente di poter accedere in maniera sicura al nodo centrale, consentendo anche la modifica del sistema e la possibilità di navigare il file system del dispositivo.
- **Server Torrent per la gestione del P2P:** serve per la gestione dei file torrent e consente di scaricare file .torrent da internet.
- **Rete VPN:** consente di connettersi alla rete locale dall'esterno tramite il protocollo VPN

- **Server Repository di tipo SVN:** strumento di collaborazione e gestione dei progetti.

Questi servizi sono aggiuntivi rispetto alle funzionalità domotizzate e permettendo di integrare diversi strumenti che agevolano diverse operazioni, come la gestione o condivisione di file e stampanti. I servizi sono aggiuntivi e mostrano la versatilità della piattaforma che offre il *Raspberry Pi*.

In conclusione di questo capitolo precisiamo che il sistema, grazie alla sua versatilità, è stato implementato in modo che possa essere fruito anche su altri sistemi, che siano di tipo *Unix Like* o *Windows*, grazie all'implementazione con linguaggi come Python ed all'utilizzo di librerie condivise da tutti i sistemi. Il sistema è stato ottimizzato in particolar modo per *Raspberry Pi*, ma alcune delle componenti possono essere fruite da altri dispositivi, come *PC* o *Mac*; un esempio è dato dal riconoscitore vocale in aggiunta alle funzionalità di mediacenter, che può essere fruito anche su sistemi *Linux* installando il webserver del sistema domotizzato ed indicando l'indirizzo di riferimento di *XBMC* (che può essere anche lo stesso computer). Questo mostra la grande scalabilità della piattaforma e la possibilità di integrazione con diversi dispositivi.

Nel capitolo successivo mostreremo alcuni casi d'uso del sistema e verranno analizzati vantaggi e svantaggi del sistema.

Capitolo 4

Utilizzo del sistema: casi d'uso

Ora cominciamo a descrivere i casi d'uso del sistema, mostrando i meccanismi di automazione degli elementi descritti precedentemente ed i relativi consumi energetici.

L'**automazione** è uno dei principali motivi di implementazione del sistema realizzato, importante sia per gestire le azioni effettuate di frequente nell'ambiente casalingo, ma anche per conferire maggiore semplicità e velocità nell'esecuzione delle azioni all'interno dell'ambiente domotizzato, in tal modo evitando sprechi e dando maggiore libertà all'utente. Per come è stato implementato il sistema, alcune delle azioni da eseguire risultano macchinose, come l'accensione delle luci di tutti gli ambienti o l'attivazione o disattivazione della videosorveglianza, per questo motivo sono stati progettati dei scenari d'uso. Questi scenari derivano dalle azioni effettuate più comunemente e sono di tipo *statico*, poichè vengono creati in base alle esigenze dell'utente e quindi studiati e progettati *ad hoc* per le situazioni che verranno descritte nel paragrafo successivo. Tramite la strutturazione degli scenari vengono attuati dei **meccanismi di risparmio energetico** che a loro volta possono essere **diretti** o **indiretti**: diretti se viene impostata l'azione per un determinato componente, indiretti se la conseguenza di una determinata azione su un componente si ripercuote sia sull'ambiente che sugli altri componenti del sistema. Si evince da quanto detto che gli scenari sono dei meccanismi di automazione che riguardano sia i componenti del sistema domotico, sia l'ambiente a cui si fa riferimento.

I casi d'uso sono strettamente correlati all'utilizzo dei componenti del sistema domotico. Essi sono diversi e si possono riassumere in due categorie: **multimediali** e **funzionali**. I casi d'uso inerenti alla *multimedialità* derivano dall'utilizzo dell'ambiente domotizzato con il sistema multimediale tramite il mediacenter **XBMC**. L'utilizzo del media center può avvenire in due modi: attraverso i *comandi vocali*, tramite riconoscitore ambientale o in applicazione mobile, ed attraverso i *comandi manuali*, tramite l'apposita applicazione implementata oppure con applicazioni implementate per il programma *XBMC*. Un set di comandi possibili sono:

- “Avvia film Armageddon”: il sistema cerca all'interno della libreria video il film armageddon e lo avvia, in caso contrario avverte l'utente che non ha trovato il film.
- “Avvia Musica”: l'utente utilizza il sistema per mettere in playback la libreria musicale in modalità casuale.
- “Pausa”: il sistema mette in pausa l'esecuzione corrente.

I casi d'uso multimediali sono i più semplici e derivano dall'implementazione del sistema *A.V.I.D.*

I casi d'uso inerenti alle *funzionalità* del sistema sono parte integrante dell'utilizzo degli elementi domotizzati nell'ambiente casalingo. L'utilizzo di questi elementi deriva da: l'**interfaccia applicativa**, web o tramite applicazioni mobile, e dal **riconoscitore vocale**, tramite **funzionalità fisiche** (come bottoni e display) e tramite **TAG NFC**. Con l'interfaccia applicativa è possibile comandare manualmente le funzioni implementate nel sistema. Un esempio di uso è dato dall'utilizzo dell'applicazione mobile per accendere o spegnere i punti luce forniti dal sistema. Vediamo alcuni degli esempi più complessi:

- *Controllo remoto della termoregolazione*: all'esterno dell'ambiente casalingo è possibile settare la temperatura desiderata ed accendere il termostato prima di rientrare all'interno dell'ambiente.
- *Controllo remoto dei punti luce*: dall'esterno è possibile azionare i punti luce in condizioni di scarsa luminosità per migliorare l'immagine dello streaming remoto

- *Spegnimento delle luci*: in caso di disattenzione è possibile spegnere, attraverso sia comandi vocali sia tramite le interfacce web e mobile, i punti luce da una stanza ad un'altra.
- *Accensione del termostato*: l'accensione del termostato può essere effettuata attraverso l'applicazione di riferimento, direttamente da qualsiasi punto della casa, purchè connessi alla rete casalinga.

Molti dei comandi descritti possono essere fruiti anche attraverso i riconoscitori vocali, con l'unica limitazione di utilizzo all'interno dell'ambiente casalingo, quindi connessi alla rete domestica. Alcuni dei casi d'uso per questa interfaccia sono le seguenti:

- *“Accendi Luci del salone”*: comando che accende tutti i punti luce del salone.
- *“Avvia riscaldamento”*: accende il termostato alla temperatura precedentemente impostata.
- *“Spegni tutte le luci”*: spegne tutte le luci della casa.

Altra importante funzionalità è dovuta all'implementazione di elementi fisici manuali, come bottoni e display. Questi elementi sono stati progettati per poter sopperire alle problematiche inerenti all'impossibilità di utilizzare le funzionalità evolute introdotte dal sistema, come l'applicazione mobile, interfaccia web e così via. Sono stati aggiunti ad ogni dispositivo dei pulsanti che consentono di interfacciarsi con le funzionalità domotizzate in modo *“classico”*. Un esempio di utilizzo di questi elementi può essere dato dal guasto del router casalingo e quindi la conseguente accensione del punto luce attraverso il bottone designato per quella luce, oppure l'accensione del riscaldamento attraverso i pulsanti posto nella parte frontale del termostato. Questi casi d'uso sono molto importanti per garantire un utilizzo continuativo ed affidabile del sistema, componente fondamentale in un sistema domotizzato.

Infine i casi d'uso dovuti all'interfaccia dei *TAG NFC* verrà trattata nel paragrafo successivo riguardante l'automazione.

4.1 Automatizzazione

La componente di automazione del sistema è stata studiata ragionando su una logica di utilizzo rispetto a degli scenari preimpostati. Questi scenari sono parte integrante dell'utilizzo quotidiano delle funzionalità implementate dal sistema, consentendo di automatizzarle. Esistono diversi scenari che si configurano nel sistema che possono essere fruiti dagli elementi di automazione, come i *TAG NFC* (vedi figura 4.3). Due degli scenari più importanti ed utilizzati all'interno del si-



Figura 4.1: TAG NFC

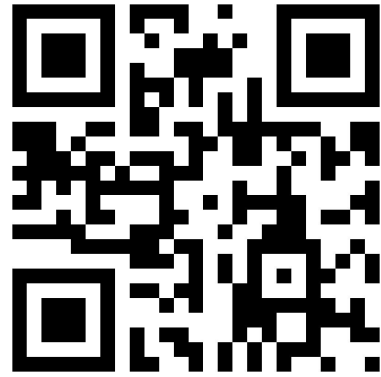


Figura 4.2: QR CODE

Figura 4.3: Esempi di TAG NFC e QR CODE

stema, e la loro conseguente automazione, sono: la “*chiusura del sistema*”, azione imputabile all'uscita da parte dell'utente dall'ambiente casalingo, e la conseguente “*entrata nell'ambiente casalingo*”.

La prima azione consente di disattivare tutti gli elementi della casa, spegnendo tutte le luci collegate all'ambiente casalingo e disattivando il termostato, qualunque sia lo stato di queste componenti; successivamente viene attivata la videosorveglianza in tutti i punti in cui è implementato il servizio, attivando il rilevamento di intrusioni. È bene evidenziare quindi come tale automazione agevoli l'utente nelle azioni più comuni: chiudere la casa spegnendo tutti i punti luce e attivando “*l'allarme*” ove installato.

La seconda in particolare, azione consente all'utente di attivare degli elementi scelti all'entrata nell'ambiente casalingo, per cui si attiveranno i punti luce scelti e verrà disattivata la videosorveglianza. Dunque: verranno attivati i punti luce se il rientro è compreso tra le 17:00 PM e le 05:00 AM, ore in cui la luce naturale scarseggia, e

verrà disattivata la videosorveglianza.

Queste operazioni sono state strutturate ed implementate attraverso dei **TAG NFC (Near Field Communication)**, che contengono la richiesta ben specifica che viene inviata al webserver attraverso lo smartphone dotato di lettore NFC. L'automazione tramite *NFC* utilizza lo smartphone come lettore mentre il compito di eseguire le azioni è lasciato al nodo centrale del sistema. Questa componente del sistema è importante poichè consente di automatizzare le azioni che l'utente compie, agevolando gli aspetti della quotidianità. L'implementazione è molto versatile, poichè consente di utilizzare qualsiasi supporto che contenga l'informazione dell'azione automatizzata, come ad esempio i suddetti *TAG NFC* o *QRCODE*. Infatti, all'interno di questi supporti è contenuta una semplice richiesta di tipo *GET* per il webserver che esegue l'azione. Sia per la natura della richiesta che per il pericolo di attacchi *Man-in-the-middle*, l'automazione può essere *eseguita solo all'interno dell'ambiente casalingo*. Questa è una limitazione dovuta alla natura della richiesta *GET*, poichè unica richiesta che può essere elaborata ed inviata dal programma che legge il *TAG NFC* o il *QRCODE*. Esistono due programmi utilizzati come lettori, entrambi sviluppati per piattaforma *Android*, e sono: **Trigger**, che consente di leggere, scrivere ed eseguire le azioni scritte su un *TAG NFC*, e **Barcode Scanner**, che consente di leggere *QRCODE*. Questi due programmi inviano direttamente la richiesta al web server aprendo il browser dello smartphone, eseguendo l'azione. All'interno del web server è implementato un servizio, identificato da una pagina web *PHP*, che gestisce le richieste di automazione, ricevendo i parametri delle richieste *GET* che sono:

- *Username e Password*: Credenziali per accedere alla funzionalità di servizio
- *Tipo di richiesta*: indica la tipologia di richiesta identificata dall'automazione da eseguire
- *Parametri*: parametri aggiuntivi per l'esecuzione della richiesta

Questi due scenari sono stati strutturati anche all'interno dell'interfaccia mobile (vedi 4.4), conferendo uno strumento più familiare per l'automazione di queste funzionalità.

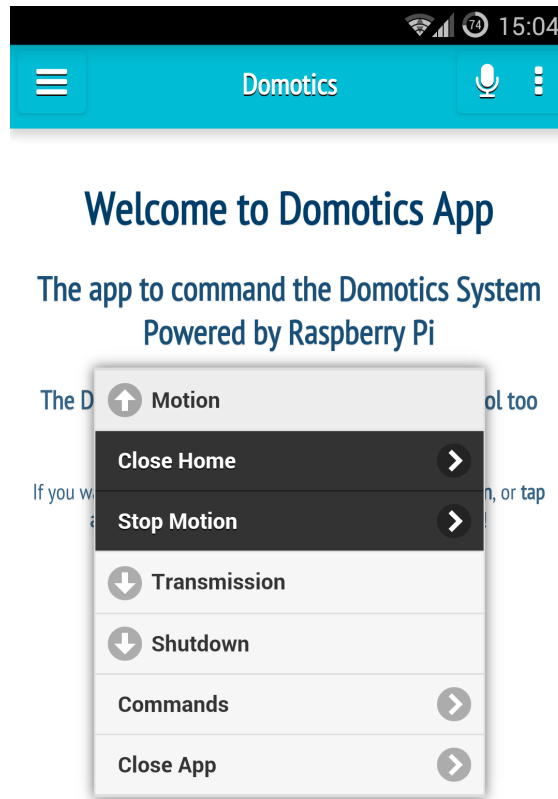


Figura 4.4: Interfaccia automazione applicazione mobile

Il motivo dell'implementazione di queste due automazioni attraverso *TAG NFC* risiede nella facilità di utilizzo tipica di questa interfaccia che, a differenza delle precedenti descritte, consente di eseguire le azioni in modo più veloce. Infatti, consente di eseguire il minor numero di passaggi per compiere le azioni desiderate, limitando l'utente al solo passaggio dello smartphone verso il *TAG NFC*, ed inoltre consente una maggiore dimestichezza da parte dell'utente che si trova ad utilizzare il sistema con la componente “*intelligente*”.

Un altro esempio di scenario per l'automatizzazione è l'utilizzo del riconoscitore vocale e delle sue funzionalità nell'ambiente casalingo. Grazie alla personalizzazione tramite interfaccia web dei comandi, l'utente può configurare a suo piacimento l'insieme di azioni che il sistema deve compiere rispetto ad un comando vocale. Un esempio è dato dalla strutturazione del comando di “*spegnimento luci salone*”

descritto precedentemente. Quindi anche questo strumento risulta molto potente e versatile in chiave di automazione.

Si può notare come manchi la gestione tramite interfaccia web o applicazione mobile delle azioni di automazione, che vengono eseguite a livello di codice dal servizio web. In uno sviluppo futuro questa sarà una componente molto importante del sistema, che conferirà all'utente la possibilità di strutturare degli scenari di automazione tramite delle azioni, scegliendo sia i parametri da utilizzare che il tipo di scenario desiderato. Inoltre, verranno inseriti e progettati gli elementi di automazione direttamente nell'interfaccia web e dell'applicazione mobile, consentendo all'utente di fruire delle stesse funzionalità.

4.2 Risparmio energetico

L'utilizzo quotidiano del sistema domotizzato comporta non solo comfort, ma anche la gestione di un insieme di azioni che portano ad un vero e proprio risparmio energetico. Questo sistema non prevede meccanismi di gestione diretta dei consumi energetici, poichè mancano le implementazioni dei sistemi di rilevamento dei consumi e di gestione degli apparecchi più complessi, come gli elettrodomestici. L'uso coscienzioso delle funzionalità del sistema, grazie agli elementi di automazione ed al consumo energetico che hanno di dispositivi, comporta consumi contenuti e perfino ridotti, fattore principale che consente il risparmio energetico, poichè è possibile configurare il sistema in modo tale da evitare sprechi dovuti a disattenzione o da un utilizzo intensivo.

Un esempio è lo scenario di "*chiusura della casa*", il quale consente di eliminare sprechi dovuti da un uso improprio di quel componente. Tramite questa automazione si è riscontrato un risparmio notevole, che varia tra il 15% e il 20% di consumi in meno, così come la componente di termoregolazione, che consente di gestire al meglio la regolazione della temperatura ambientale e di evitare inutili sprechi. È comprovato che l'utilizzo di un termostato consente di risparmiare notevolmente rispetto ad un ambiente privo di questo elemento. Il termostato domotizzato consente di utilizzare un termostato classico con l'aggiunta di automazioni e di

controlli remoti, conferendo a questo strumento maggiore flessibilità e quindi comportando minori costi.

Un altro fattore a cui si è data grande importanza è il consumo derivato dai dispositivi implementati. Poichè i dispositivi sono stati realizzati attraverso Raspberry Pi, le specifiche di funzionamento e di consumo sono per il 90% causate da questo dispositivo, il quale consuma circa 700mA. Si può notare come il consumo di questi apparecchi sia di gran lunga più basso (a parità di funzionalità) rispetto all'implementazione con altri dispositivi, come la strutturazione del un nodo centrale attraverso una macchina server o con un PC. Il consumo dovuto all'utilizzo di questi dispositivi è molto basso. Per massimizzare ulteriormente il risparmio si consiglia l'utilizzo combinato di dispositivi Raspberry Pi ed Arduino.

Capitolo 5

Conclusioni

La tesi mostra l'implementazione di un sistema domotizzato mediante tecnologia open-hardware, che rispetta le specifiche tipiche di un sistema domotico (illustrate nel capitolo 2) ed affronta le problematiche inerenti agli aspetti comuni alla progettazione ed implementazione. Viene mostrato come sia possibile realizzare un sistema *ex-novo* e come sia alla portata di tutti gli utenti che hanno delle conoscenze basilari di elettronica ed informatica. La realizzazione del sistema completo ha comportato circa quattro mesi di lavoro, quindi un lasso di tempo medio molto breve tempo. I programmi di gestione e di comunicazione dell'elettronica sono basilari rispetto ad implementazioni simili e più complesse, e consentono, con poche righe di codice e grazie all'ausilio di librerie già implementate per i linguaggi più comuni, di gestire la comunicazione tra il dispositivo e l'elettronica; quest'ultimo aspetto è stato reso possibile grazie dall'implementazione tramite *Raspberry Pi*, risultata la piattaforma più flessibile ed adatta alla realizzazione di tutte le tipologie di dispositivi presenti in un impianto domotizzato. Inoltre la piattaforma consente l'interfacciamento con la maggior parte dell'elettronica in commercio utilizzabile per dispositivi domotizzati, conferendo una forte scalabilità al sistema.

Sono stati messi in risalto due aspetti fondamentali in questo sistema: la **semplicità di utilizzo** del sistema e l'**automazione** degli elementi connessi. La **semplicità e versatilità** d'utilizzo è una componente importante ai fini dell'uso nella quotidianità, dando un valore aggiunto al sistema che si differenzia da una implementazione puramente ingegneristica; per questo sono state affrontate tutte

le tematiche di usabilità e *user experience*, ponendosi nei panni di un utente privo di conoscenze su qualsiasi tipo di sistema domotizzato e che ha una minima familiarità con l'utilizzo di smartphone e personal computer. A tal scopo sono state create **tre interfacce** che si differenziano per utilizzo e presentazione, conferendo diversi modi di interazione all'utente e la scelta di cosa eseguire e come eseguirlo. Le tre interfacce sono progettate come strumenti di presentazione delle funzionalità domotizzate semplici ed intuitive. La visualizzazione dei punti luce attraverso dei bottoni, che vengono distinti attraverso un label indicante quale punto luce azionare, la visualizzazione dello stato del termostato e le relative impostazioni, mettendo in risalto l'accensione e lo spegnimento nell'interfaccia, sono tutti esempi che mostrano una tipologia di interfaccia minimale e familiare per tutte queste funzionalità. Inoltre è stata data la possibilità all'utente di poter configurare a proprio piacimento gli elementi dell'impianto domotizzato attraverso un'interfaccia apposita. Quest'ultima è uno degli elementi cardine del sistema, che consente di:

- Strutturare la **tipologia di rete** ibrida descritta
- **Interconnettere le funzionalità** dei dispositivi implementati
- **Configurare** in modo semplice le funzionalità del sistema
- **Gestire** il sistema in modo che sia sempre funzionante ed affidabile.

Gli aspetti elencati sono importanti e sono stati strutturati in modo tale che qualsiasi tipo di utenza possa **utilizzare** ed **installare** il sistema, mettendo allo stesso livello sia l'utente esperto che l'utente inesperto, affinché entrambi abbiano la stessa capacità di installare il sistema. Questo aspetto è stato realizzato grazie sì alla configurazione del sistema, ma anche alla progettazione di dispositivi che necessitano soltanto una semplice connessione, come può essere il collegamento di fili, o il posizionamento del dispositivo, come accade per il controllore delle luci o il termostato, attribuendo mobilità e versatilità al sistema.

L'**automazione** è un'altra componente di spicco del sistema, che consente di effettuare in modo veloce e semplificato le azioni all'interno dell'ambiente, dando

all'utente la possibilità di interagire con l'ambiente domotizzato attraverso azioni univoche e strutturate per degli scenari (discussi nel capitolo 4). Principalmente è stata usata l'interfaccia implementata attraverso **TAG NFC** che si sono dimostrati gli elementi più adatti ad una automazione vera e propria all'interno dell'ambiente casalingo. Questo tipo di automazione si differenzia rispetto ad altri sistemi, che utilizzano chiavi o smartcard, per la semplicità d'uso e di implementazione, utilizzando lo smartphone come mezzo di lettura e scrittura delle azioni; si ha così un vantaggio di non dover utilizzare un lettore di *TAG NFC* per implementare la funzionalità, ma anche uno svantaggio, poichè è necessario che lo smartphone sia dotato di lettore *NFC*. Il sistema consente non solo di utilizzare i *TAG NFC* per l'automazione ma anche uno strumento più versatile come QRCODE, utilizzato nella prima versione del sistema. L'automazione è resa **semplice** ed **intuitiva** per **qualsiasi utente**, poichè attraverso un'unica azione (lettura del *TAG* tramite smartphone) è possibile eseguire diverse azioni all'interno di uno o più ambienti della casa.

Il sistema realizzato presenta una complessità bassa nell'implementazione dei programmi, sia di interfacciamento con elettronica, sia di gestione delle funzionalità del sistema, dovuta principalmente alla grande documentazione in rete, attraverso vari progetti di interfacciamento con l'elettronica open-hardware e diverse community che affrontano problematiche per questa tecnologia e sviluppano soluzioni per dispositivi open-hardware; si ha inoltre la possibilità di interfacciarsi con l'elettronica attraverso dei linguaggi di alto livello come *Python* e *Bash*, utilizzando librerie open-source. L'utilizzo di questi linguaggi ha consentito uno sviluppo crescente del sistema, consentendo di integrare nuovi elementi e di ampliare le funzionalità che esso può offrire. Quindi l'utilizzo di tecnologia open-hardware consente in modo semplice, attraverso community ben sviluppate e strutturate, di progettare una soluzione domotizzata e di ampliarne le funzionalità. Inoltre la bassa complessità è uno dei punti base ed importanti del sistema realizzato, poichè evidenzia un aspetto importante di questa tesi: chiunque abbia minime conoscenze informatiche ed elettroniche può creare un sistema domotizzato.

Una altro aspetto sviluppato all'interno della tesi è l'importanza della versatilità

che un sistema domotico deve avere. Il sistema realizzato, grazie all'utilizzo di dispositivi open-hardware, consente la **portabilità** e l'**installazione** in un impianto che non ha necessariamente una predisposizione per la domotica, sia riguardo agli elementi non strettamente correlati alle funzionalità casalinghe (come multimedia e videosorveglianza), sia rispetto a quelli strettamente correlati (come luci e termostato). Tutti gli elementi possono essere trasportati da un ambiente ad un altro, dal controllore delle luci al termostato: il dispositivo che gestisce punti luce consente la sua mobilità poichè si interfaccia con una comune presa elettrica, mentre il termostato, che utilizza tecnologia *WiFi 802.11*, può essere posizionato in qualsiasi punto della casa.

Un aspetto che consente la mobilità è la **strutturazione del tipo di rete** che il sistema domotizzato implementa poichè ogni dispositivo è indipendente dagli altri, ma allo stesso tempo può comunicare con tutti i dispositivi connessi alla rete attraverso le primitive di comunicazione implementate per i dispositivi e l'ausilio del file di configurazione; quest'ultimo aspetto mostra la grande versatilità del sistema, evidenziando le potenzialità e gli scenari che il sistema realizzato offre.

Infine, la gestione della **multimedialità** è un aspetto che consente di rendere interattivo l'ambiente casalingo. L'interattività è un aspetto che è stato curato in particolar modo, grazie all'ausilio del progetto *A.V.I.D.* realizzato in ambito accademico con la collaborazione di Lorenzo Casanova, mostrando come sia possibile interagire con l'ambiente casalingo non soltanto attraverso delle interfacce applicative, ma anche utilizzando un'interfaccia alternativa: i comandi vocali. La possibilità di utilizzare comandi vocali conferisce una **interazione attiva** con la casa che risponde ai comandi vocali eseguendo il comando richiesto che può variare dall'interazione con il media center fino all'azionamento degli elementi domotizzati come punti luce e termostato.

5.1 Considerazioni tra sistema implementato e sistemi proprietari

Nella parte iniziale della tesi vengono distinti in due tipologie i sistemi domotizzati: proprietari e open. Dallo studio effettuato si nota subito che la differenza tra sistemi open e proprietari risiede nella **scalabilità** dei sistemi e dalle differenti documentazioni poiché un sistema open è rivolto maggiormente ad una diffusione della conoscenza piuttosto che alla sola commercializzazione; quest'ultimo aspetto è stata una problematica per il reperimento di informazioni riguardo ai sistemi proprietari. Il sistema realizzato con tecnologia open-hardware mostra dei punti di forza e di debolezza rispetto ai sistemi proprietari. Cominciamo ora a descrivere i punti di forza.

Nel sistema viene mostrata una logica di interazione e scalabilità che non è presente all'interno dei sistemi proprietari: un sistema proprietario fornisce solitamente un interfacciamento con altri dispositivi, a volte attraverso dei protocolli open (vedi capitolo 2 con il sistema *MyHome bTicino*), che però non dà la libertà di scelta della tipologia di servizi o funzionalità rispetto ad un sistema proprietario, come la scelta (o implementazione) di un protocollo di comunicazione differente da quello utilizzato o la comunicazione con elettronica differente rispetto a quella fornita dal sistema proprietario. Un esempio lampante è l'accesso dall'esterno del sistema alle funzionalità implementate: un sistema proprietario struttura un servizio (solitamente a pagamento) per l'accesso del sistema dall'esterno, mentre il sistema realizzato consente di creare l'accesso al sistema dando la possibilità all'utente (più esperto) di scegliere che tipo di accesso effettuare, senza costi aggiuntivi. Il sistema implementato, a differenza di uno proprietario, consente proprio la **scalabilità** e **versatilità** necessaria per poter integrare funzionalità rispetto a delle nuove esigenze dell'utente.

Un punto di forza importante del sistema realizzato rispetto ad uno proprietario è la capacità di mettere allo stesso livello un utente inesperto ed un utente esperto (come un installatore), dando la possibilità di configurare ed installare con estrema facilità il sistema, a differenza di uno proprietario che, rispetto sia all'in-

stallazione che ad una eventuale manutenzione, richiede un intervento specializzato da parte di un tecnico. Inoltre, sono stati progettati i meccanismi che consentono una **fruizione dei contenuti** in modo **semplice** ed **intuitivo** attraverso la vasta gamma di interfacce proposte, diversamente da un sistema proprietario che consente l'interazione soprattutto attraverso l'utilizzo di una interfaccia applicativa. La possibilità di scissione di ogni componente domotico rispetto alla rete di elementi domotizzati è un altro punto di forza importante, che manca rispetto ai sistemi proprietari poiché viene venduto solitamente un sistema *All-In-One*, completo di tutte le funzionalità e quindi non incorporabile nelle sue unità minime né trasferibile da un impianto ad un altro.

Al contrario, un sistema proprietario mostra una differenza importante rispetto al sistema implementato: la scissione di tutti i componenti logici. Infatti, nel sistema implementato si ha un unico dispositivo sia per il controllore dell'elettronica sia per la componente intelligente e per la componente di comunicazione, mentre nei sistemi proprietari si tende a scindere questi componenti, per via dei costi di gestione e di implementazione. Questa differenza importante è dettata dalla tipologia di prodotto poiché il sistema implementato utilizza il *Raspberry Pi* per svolgere i compiti di queste tre componenti. Infatti, se un tale sistema venisse commercializzato si avrebbe una problematica inerente ai costi dei dispositivi e dell'elettronica che rispetto ad un approccio proprietario e di scissione delle componenti risulterebbe notevolmente abbassato. Inoltre, un punto di forza dei sistemi proprietari è dovuta ad una manutenzione fatta dalla ditta fornitrice, mentre il sistema realizzato necessita di una manutenzione da parte dell'utilizzatore. Tuttavia il sistema realizzato consente la commercializzazione, fornendo un dispositivo unico e specializzato per svolgere determinate azioni nell'ambiente casalingo e abbassando i costi notevolmente rispetto ad un sistema proprietario che fornisce una soluzione *All-In-One*. Inoltre, consente attraverso la connessione con altri dispositivi che comunicano tra loro e la possibilità di eleggere un dispositivo a nodo centrale, di creare un sistema indipendente ed autonomo, abbassando i costi rispetto ad un sistema proprietario in qualsiasi caso.

5.2 Sviluppi Futuri

Questa tesi nasce principalmente da una **esigenza**: automatizzare l'ambiente casalingo in modo tale che le azioni effettuate più frequentemente vengano eseguite in modo facile e veloce. Il sistema allo stato attuale fornisce diversi servizi che possono essere ampliati integrando nuove funzionalità. Negli sviluppi successivi verranno implementati dei meccanismi per la gestione dell'automazione, attraverso delle interfacce di configurazione che consentiranno di scegliere quale tipo di automazione eseguire. Conseguentemente a questa gestione si avrà la possibilità di configurare gli scenari che saranno quindi prestabiliti (come quelli discussi nel capitolo 4). Questa componente, sempre seguendo la linea guida di semplicità e velocità d'utilizzo, sarà una parte importante del sistema, creando uno strumento che conferisce all'utente la totale gestione dei possibili scenari all'interno di uno o più ambienti della casa.

Successivamente verranno introdotti nuovi dispositivi strutturati in modo "ibrido", comprendenti *Arduino*, *Raspberry Pi* ed elettronica compatibile. L'idea che sta alla base è creare un connubio tra le due tecnologie, le quali hanno punti in comune ma anche punti di vantaggio e svantaggio l'una rispetto all'altra. Un esempio è la strutturazione di un sistema di punti luce in cui in ogni punto luce si ha un *Arduino*, che tramite una comunicazione *RF (Radio Frequency)* o *ZigBee* (attraverso modulo *XBee*) riceve il comando da un server centrale (in questo caso un *Raspberry Pi*) di azionare i punti luce. Il connubio tra le due tecnologie porterebbe alla scissione delle componenti del sistema discusso precedentemente, utilizzando *Arduino* come controllore dell'elettronica, *Raspberry Pi* come elemento d'intelligenza e strutturando un protocollo di comunicazione *end-to-end*. Inoltre il connubio delle due tecnologie comporterebbe un ulteriore abbassamento dei costi, poiché l'*Arduino* ha nettamente dei costi inferiori rispetto al *Raspberry Pi*, ed anche lo spreco di risorse necessarie, poiché la potenza di calcolo per effettuare le operazioni sarebbe proporzionata ai dispositivi utilizzati. Infine porterebbe ad un avanzamento della portabilità del sistema poiché aggiungerebbe la componente wireless indipendente dal gateway casalingo. Infatti, allo stato dell'arte, il sistema consente di inserire dei punti wireless all'interno dell'ambiente (come il termostato), ma questa connes-

sione è vincolata al *home gateway* casalingo. Inserendo una componente wireless attraverso comunicazione *RF* o *ZigBee* con dispositivi *Arduino* si potrebbe fare a meno della componente *Home Gateway*, rendendo sempre di più indipendente il sistema. Come si può notare, l'utilizzo di questi due dispositivi crea nuove soluzioni sia per gli elementi già presenti che per elementi nuovi all'interno dell'ambiente casalingo.

Grazie alla strutturazione del sistema, nuovi componenti saranno integrati con facilità, stando al passo con l'evoluzione tecnologica e delle esigenze all'interno dell'ambiente casalingo, seguendo sempre la linea di semplicità e velocità d'utilizzo tipiche del sistema realizzato.

Elenco delle figure

2.1	MyHome bTicino	14
2.2	Centralina di irrigazione Open Sprinkler	17
2.3	Raspberry PI Modello B	18
3.1	Riconoscitore ambientale con Raspberry Pi Model B	32
3.2	Gestore Luci nell'interfaccia web	34
3.3	Gestione del termostato interfaccia web	35
3.4	Gestione del termostato interfaccia web	36
3.5	Interfaccia web di configurazione per le luci	37
3.6	Popup di inserimento/modifica comandi nella configurazione	37
3.7	Interfaccia web di configurazione principale	38
3.8	Interfaccia web di configurazione per il mediacenter	39
3.9	Controllore Termostato interfaccia mobile	40
3.10	Controllore punti luce interfaccia mobile	40
3.11	Controllore mediacenter interfaccia mobile	40
3.12	Configurazione applicazione mobile	40
3.13	Esempi di Interfacce Applicazione Mobile	40
3.14	Homepage Interfaccia Web	43
3.15	Circuito dei bottoni	47
3.16	Prototipo dei pulsanti	47
3.17	Prototipo di termostato implementato nel sistema	48
3.18	Dettaglio della visualizzazione delle informazioni nel display del termostato	50
3.19	Dettaglio della connessione del sensore DS18B20 (powered by Adafruit)	52

3.20	Esempio di messaggio di allerta da parte del sistema di videosorveglianza (Motion)	55
3.21	Interfaccia principale di XBMC	58
4.1	TAG NFC	68
4.2	QRCODE	68
4.3	Esempi di TAG NFC e QRCODE	68
4.4	Interfaccia automazione applicazione mobile	70

Bibliografia

- [1] bTicino. *MyHome Domotica*. 2014. URL: http://www.bticino.it/media/downloads/MyHOME_domotica.pdf (cit. alle pp. 14, 15).
- [2] Giardino Domotico. *Centraline domotiche*. 2014. URL: <http://www.giardinodomotico.it/centraline-irrigazione> (cit. a p. 16).
- [3] Adafruit. *Adafruit's Raspberry Pi Lesson 11. DS18B20 Temperature Sensing*. 2014. URL: <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/software> (cit. a p. 51).
- [4] Wikipedia. *Crontab*. 2014. URL: <http://it.wikipedia.org/wiki/Crontab> (cit. a p. 53).
- [5] MJPG-Streamer Tom. *MJPEG-streamer*. 2011. URL: <http://sourceforge.net/p/mjpg-streamer/wiki/Home/> (cit. a p. 56).
- [6] Wikipedia. *Home Automation*. 2014. URL: http://en.wikipedia.org/wiki/Home_automation.
- [7] Wikipedia. *Domotica*. 2014. URL: <http://it.wikipedia.org/wiki/Domotica>.
- [8] Centro Domotica. *Cose la domotica*. 2014. URL: <http://www.centrodomotica.it/domotica.aspx>.
- [9] Raspberry Pi Foundation. *What is a Raspberry Pi*. 2014. URL: <http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>.
- [10] Wikipedia. *Raspberry Pi*. 2014. URL: http://it.wikipedia.org/wiki/Raspberry_Pi.

Quest'opera e tutti i suoi contenuti sono stati rilasciati con licenza Creative Commons Attribuzione - **Non commerciale** - Condividi allo stesso modo 4.0 Internazionale. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-sa/4.0/>.



Ringraziamenti

Anche il traguardo della Laurea Magistrale è concluso, chiudendo il quinquennio del Corso di Informatica.

Questo traguardo non sarebbe mai stato raggiunto senza l'apporto delle tante persone hanno caratterizzato il mio percorso.

Per primi ringrazio i miei genitori, Patrizia e Gianluigi, per avermi sempre aiutato e sostenuto in questo percorso, con le parole e il sostegno che solo dei genitori possono dare ad un figlio. Grazie per tutto quello che avete fatto e fate tutt'ora per me, dandomi sempre un esempio da imitare di amorevolezza, comprensione e fiducia che solo dei genitori come voi sanno dare. Grazie dal profondo del cuore.

Ringrazio tutta la mia famiglia, mia sorella Michela e tutti i miei zii, che in questi anni di università hanno sempre creduto e sostenuto le mie capacità, dandomi sempre coraggio e fiducia in tutto quello che ho fatto.

Ringrazio tutti i miei amici, che in questi anni mi hanno supportato in questa avventura, in particolare: Alessandro, Umberto, Lorenzo, Luca, e tutti gli altri.

Ringrazio anche i miei compagni di università, per aver percorso la magistrale insieme affrontando ogni ostacolo, tra un aperitivo e l'altro, e trovando sempre un aiuto valido e professionale.

Ringrazio i Professori del Corso di Laurea Di Scienze di Internet, per aver mostrato non solo una grande professionalità ed umanità, ma anche una grande voglia e passione per ogni insegnamento tenuto.

Ultima, ma non meno importante, ringrazio dal profondo del cuore la persona che in questi ultimi mesi mi ha sopportato e supportato in tutto, dandomi fiducia e credendo in me, sempre disponibile ed aiutandomi in tutto e per tutto. Grazie Ilaria.

Ora incomincia un nuovo capitolo della mia vita, con speranza e determinazione.