



**SAPIENZA**  
UNIVERSITÀ DI ROMA

**Relazione progetto**  
**Interazione multimodale**  
**Anno Accademico 2013/2014**

**Studenti**

Del Prete Alessandro

Di Florio Andrea

**Docente**

Prof.ssa Maria De Marsico

# Sommario

<b>Introduzione .....</b>	<b>2</b>
<b>Obiettivi .....</b>	<b>2</b>
<b>Struttura relazione .....</b>	<b>2</b>
<b>Capitolo 1: Architettura del sistema .....</b>	<b>3</b>
<b>1.1 Decodifica segnali .....</b>	<b>5</b>
<b>1.2 Invio dei segnali .....</b>	<b>7</b>
<b>1.3 Diagramma delle classi applicazione Android .....</b>	<b>11</b>
<b>Capitolo 2: Casi d'uso e diagrammi.....</b>	<b>13</b>
<b>2.1 Descrizione casi d'uso.....</b>	<b>13</b>
<b>2.2 Diagrammi di attività .....</b>	<b>17</b>
<b>2.3 Diagrammi di sequenza.....</b>	<b>23</b>
<b>Capitolo 3: Scenari .....</b>	<b>26</b>
<b>3.1 Scenario 1: Accesso all'applicazione e apprendimento delle prime funzionalità .....</b>	<b>26</b>
<b>3.2 Scenario 2: Avvio film da hard disk.....</b>	<b>30</b>
<b>3.3 Scenario 3: Interazione con un'applicazione presente su una Smart tv .....</b>	<b>33</b>
<b>Conclusioni.....</b>	<b>35</b>
<b>Possibili sviluppi futuri .....</b>	<b>35</b>
<b>Bibliografia.....</b>	<b>36</b>

# Introduzione

Negli ultimi anni è in costante diffusione l'utilizzo della tecnologia basata sulla speech recognition, sempre più impiegata per scrivere sms, gestire smartphone, scrivere testi ecc. L'idea del nostro progetto è di realizzare un sistema che, attraverso la speech recognition, sia in grado di comandare a distanza un televisore. In commercio esiste un dispositivo simile di lg (Magic Remote Control), compatibile esclusivamente solo con televisori della stessa fabbrica e con un costo che si aggira attorno ai 60€[12].

## Obiettivi

Il progetto che abbiamo deciso di sviluppare si prefigge di creare un telecomando vocale che abbia dei costi contenuti e in particolare sia in grado di:

- Riconoscere comandi vocali enunciati dall'utente;
- Inviare comandi semplici, normalmente inviati premendo un bottone sul telecomando;
- Inviare comandi complessi, normalmente inviati premendo una serie di bottoni sul telecomando.

## Struttura relazione

La relazione è strutturata come segue:

- Nel Capitolo 1 verrà descritta l'architettura del sistema, analizzando dettagliatamente tutti i componenti e l'applicazione sviluppata.
- Nel Capitolo 2 verranno descritti i casi d'uso, illustrati i diagrammi di sequenza e i diagrammi di attività.
- Nel Capitolo 3 saranno presentati alcuni scenari di utilizzo del sistema.
- Nel Capitolo 4 saranno discussi i possibili sviluppi futuri del sistema e le diverse modalità di utilizzo.

# Capitolo 1

## Architettura del sistema

Per non far lievitare i costi si è optato per l'utilizzo di uno smartphone Android posseduto da milioni di persone, e una scheda Arduino con un costo che si aggira attorno i 20 euro (molto meno rispetto al telecomando prodotto da LG). Lo smartphone sarà utilizzato per effettuare la speech recognition sfruttando le API[9] messe a disposizione da Google, mentre Arduino per inviare il codice infrarossi. Inizialmente si pensava di effettuare il riconoscimento vocale attraverso il pc, tramite la libreria java Sphinx, ma essa presentava diverse problematiche, nello specifico i comandi venivano riconosciuti in modo non corretto; inoltre il pc risultava poco comodo. La comunicazione tra entrambi i dispositivi avviene attraverso la porta bluetooth. La scelta di questa architettura è vantaggiosa perché la maggior parte degli smartphone Android in commercio non dispone della porta infrarossi.

E' stata sviluppata un'applicazione che effettua una speech recognition e invia i comandi attraverso il bluetooth. I comandi inviati dallo smartphone sono decodificati da Arduino e utilizzati per selezionare il comando da inviare al televisore o al dispositivo controllato.

Nello specifico gli elementi utilizzati sono i seguenti:

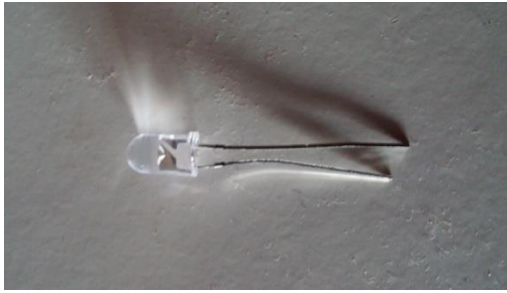
1. Smartphone Android
2. Scheda Arduino Uno[2]



3. Modulo bluetooth HC-05[8]



#### 4. IR sender led



#### 5. IR receiver TSOP4838



#### 6. Breadboard



#### 7. Resistenza



#### 8. Cavi di collegamento

## 1.1 Decodifica segnali

Per poter gestire un televisore Samsung sono stati decodificati i segnali IR(infrarossi) provenienti da un telecomando compatibile. A tale scopo è stata utilizzata la libreria IRremote[3] per Arduino.



*Figura 1.1.1: Arduino uno collegato ad un IR receiver TSOP4838*

Il codice utilizzato per decodificare i segnali è il seguente:

```
1.  #include <IRremote.h>
2.
3.  int PIN = 12;
4.
5.  IRrecv irrecv(PIN);
6.
7.  decode_results risultato;
8.
9.  void setup() {
10.   Serial.begin(9600);
11.   irrecv.enableIRIn(); //Imposta il ricevitore
12. }
13.
14. void loop() {
15.   if (irrecv.decode(&risultato)) { //Decodifica il valore
16.     Serial.println(risultato.value, HEX);
17.     irrecv.resume(); //Riceve il prossimo valore
18.   }
19. }
```

La riga di codice 5 mostra il parametro irrecv impiegato per ricevere il segnale infrarossi dal telecomando e prende come parametro in input il pin utilizzato per la ricezione. Nelle righe 10 e 11 vengono impostate rispettivamente la frequenza per la comunicazione con la porta USB di Arduino e l'abilitazione per la ricezione del segnale, mentre l'istruzione sulla riga 15 decodifica il valore del segnale inviato (in formato esadecimale) e lo stampa successivamente (riga 16) a video.

I valori dei comandi principali sono stati inseriti nella seguente tabella: la prima colonna contiene il bottone premuto sul telecomando, la seconda l'azione eseguita e la terza colonna il valore esadecimale decodificato.

COMANDO	AZIONE	VALORE ESADECIMALE
<b>Accendi/Spegni</b>	Televisione accesa/spenta	E0E040BF
<b>1</b>	Canale uno	E0E020DF
<b>2</b>	Canale due	E0E0A05F
<b>3</b>	Canale tre	E0E0609F
<b>4</b>	Canale quattro	E0E010EF
<b>5</b>	Canale cinque	E0E0906F
<b>6</b>	Canale sei	E0E050AF
<b>7</b>	Canale sette	E0E030CF
<b>8</b>	Canale otto	E0E0B04F
<b>9</b>	Canale nove	E0E0708F
<b>0</b>	Canale zero	E0E08877
<b>Volume più</b>	Alza il volume	E0E0E01F
<b>Volume meno</b>	Abbassa il volume	E0E0D02F
<b>Canale più</b>	Cambia il canale avanti	E0E048B7
<b>Canale meno</b>	Cambia il canale indietro	E0E008F7
<b>Muto</b>	Disattiva l'audio	E0E0F00F
<b>Televideo</b>	Accede al televideo	E0E034CB
<b>Lista canali</b>	Visualizza una lista dei canali	E0E0D629
<b>Strumenti</b>	Visualizza un menu per l'impostazione di audio, immagine, timer di spegnimento	E0E0D22D
<b>Info</b>	Visualizza informazioni sul programma in corso	E0E0F807
<b>Indietro</b>	Torna indietro	E0E01AE5
<b>Esci</b>	Esce da un qualsiasi menu (Menu, strumenti, info)	E0E0B44B
<b>Alto</b>	Freccia in alto (utilizzata ad esempio per cambiare ingresso della tv)	E0E006F9
<b>Basso</b>	Freccia in basso	E0E08679
<b>Sinistra</b>	Freccia a sinistra	E0E0A659
<b>Destra</b>	Freccia a destra	E0E046B9
<b>Ok</b>	Consente di accedere e confermare la modifica di un'opzione, di entrare in cartelle contenenti file multimediali situati su periferiche esterne o di aprire applicazioni della televisione	E0E016E9
<b>Smart</b>	Accede alle applicazioni della Smart tv	E0E09E61
<b>Play</b>	Fa partire il film	E0E0E21D
<b>Pausa</b>	Mette in pausa il film	E0E052AD
<b>Stop</b>	Stop del film	E0E0629D
<b>Vai avanti</b>	Manda avanti il film	E0E012ED
<b>Vai indietro</b>	Manda indietro il film	E0E0A25D

<b>Menu</b>	Visualizza il menu (sintonizzazione, gestione canali)	E0E058A7
<b>Sottotitoli</b>	Visualizza i sottotitoli	E0E0A45B
<b>Sorgente</b>	Visualizza gli ingressi della televisione	E0E0807F
<b>Guida</b>	Visualizza una guida dei canali	E0E0F20D
<b>Registra</b>	Registra un canale	E0E0926D



*Figura 1.1.2: Telecomando utilizzato per la decodifica dei segnali*

## 1.2 Invio dei segnali

Una volta ottenuti i segnali del telecomando, è stato utilizzato uno smartphone Android per effettuare il riconoscimento vocale e l'invio di caratteri mediante comunicazione Bluetooth alla scheda Arduino, a cui è collegato il modulo bluetooth. Ogni carattere è identificato da una codifica a 6 valori e ciascuna di queste codifiche viene associata ad un segnale da inviare alla televisione. Quando la scheda Arduino riceve una codifica di un carattere, verifica di quale codifica si tratta ed invia il segnale associato attraverso il led infrarossi (IRSender) alla televisione.

Per gestire la connessione bluetooth è stata utilizzata la libreria SoftwareSerial di arduino. Nel seguente codice vengono mostrate le istruzioni rilevanti per il funzionamento:

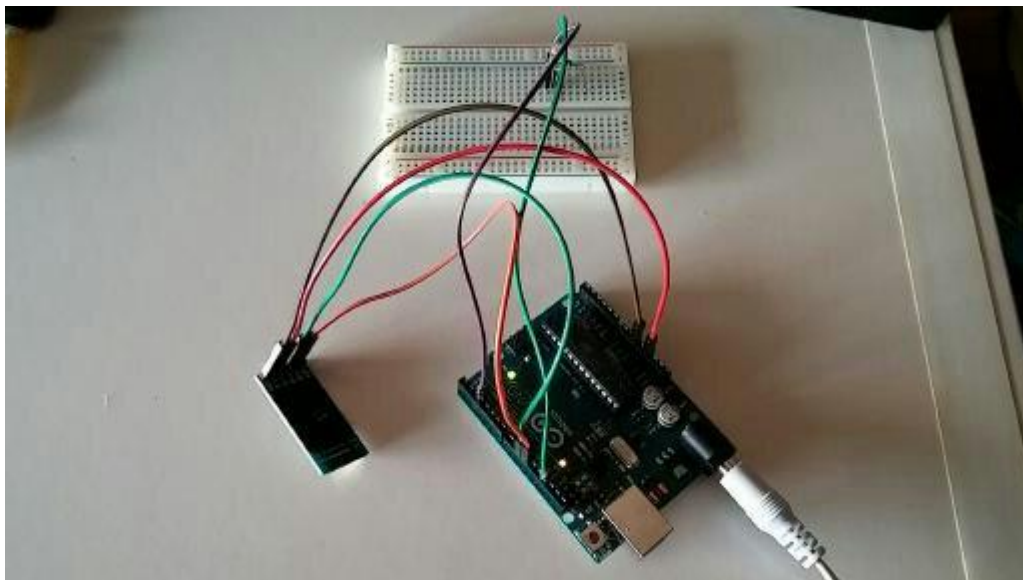


```

1.  #include <SoftwareSerial.h>
2.
3.  #define rxPin 10
4.  #define txPin 11
5.  [...]
6.  void setup() {
7.    //vengono impostati i pin per ricevere e inviare
8.    pinMode(rxPin, INPUT);
9.    pinMode(txPin, OUTPUT);
10.   mySerial.begin(38400);
12. }

```

Le righe 3 e 4 definiscono i pin di Arduino impiegati per la ricezione e l'invio dei segnali bluetooth e vengono passati come parametri alle funzioni `pinMode` contenute nelle righe 8 e 9. Infine nelle righe 10 viene impostata la frequenza per la comunicazione bluetooth.



*Figura 1.2.1: Arduino Uno collegato ad modulo bluetooth HC-05 e ad un led IR sender tramite breadboard*

Per ottimizzare la gestione della memoria in Arduino si è scelto di memorizzare ogni comando in unica variabile invece che in sei, in particolare il valore contenuto nella variabile si ottiene, per i caratteri che hanno come ultimi 3 valori 255 con questa formula (dove `val(i)` identifica l'*i*-esimo valore ricevuto):

$$\text{Id comando} = (\text{val1} * 1) + (\text{val2} * 2) + (\text{val3} * 3)$$

Per i restanti comandi invece la formula è:

$$\text{Id comando} = (\text{val1} * 1) + (\text{val2} * 2) + (\text{val3} * 3) + (\text{val4} * 4) + (\text{val5} * 5) + (\text{val6} * 6).$$

Nella tabella sottostante la prima colonna indica il comando enunciato dall'utente, la seconda il carattere inviato dallo smartphone ad Arduino, le colonne successive la decodifica del carattere in Arduino ed infine l'id associato al comando:

<b>Comando vocale associato</b>	<b>Carattere</b>	<b>Val 1</b>	<b>Val 2</b>	<b>Val 3</b>	<b>Val 4</b>	<b>Val 5</b>	<b>Val 6</b>	<b>Id comando</b>
<b>Accendi/Spegni</b>	<b>a</b>	120	0	248	255	255	255	864
<b>Uno</b>	<b>1</b>	120	0	128	255	255	255	504
<b>Due</b>	<b>2</b>	128	128	128	255	255	255	768
<b>Tre</b>	<b>3</b>	248	128	128	255	255	255	888
<b>Quattro</b>	<b>4</b>	0	248	128	255	255	255	880
<b>Cinque</b>	<b>\$</b>	0	128	128	255	255	255	640
<b>Sei</b>	<b>6</b>	128	248	128	255	255	255	1008
<b>Sette</b>	<b>7</b>	248	248	128	255	255	255	1128
<b>Otto</b>	<b>£</b>	128	0	248	0	248	255	3642
<b>Nove</b>	<b>9</b>	120	224	128	255	255	255	952
<b>Zero</b>	<b>0</b>	248	248	248	255	255	255	1488
<b>Volume più</b>	<b>+</b>	248	120	30	248	255	255	4375
<b>Volume meno</b>	<b>_</b>	248	120	254	248	120	254	1250
<b>Canale più</b>	<b>^</b>	128	120	254	255	255	255	1130
<b>Canale meno</b>	<b>}</b>	120	254	248	255	255	255	1372
<b>Muto</b>	<b>b</b>	128	0	248	255	255	255	872
<b>Televideo</b>	<b>n</b>	128	248	248	255	255	255	1368
<b>Lista canali</b>	<b>=</b>	120	254	128	255	255	255	1012
<b>Strumenti</b>	<b>/</b>	248	120	224	255	255	255	1160
<b>Informazioni</b>	<b>h</b>	0	248	248	255	255	255	1240
<b>Indietro</b>	<b>§</b>	128	0	248	128	248	255	4154
<b>Esci</b>	<b>à</b>	248	0	0	248	255	255	4045
<b>Alto</b>	<b>y</b>	120	224	248	120	224	248	1312
<b>Basso</b>	<b>d</b>	0	128	248	255	255	255	1000
<b>Sinistra</b>	<b>.</b>	128	120	224	255	255	255	1040
<b>Destra</b>	<b>,</b>	0	120	224	255	255	255	912
<b>Ok</b>	<b>k</b>	248	120	254	248	255	255	5047
<b>Smart</b>	<b>ù</b>	248	0	120	224	248	255	4274
<b>Play</b>	<b>ò</b>	248	0	128	128	248	255	3914
<b>Pausa</b>	<b>ì</b>	248	0	0	120	254	255	3528
<b>Stop</b>	<b>-</b>	120	254	120	224	255	255	4689
<b>Vai avanti</b>	<b>]</b>	248	248	120	254	255	255	4809
<b>Vai indietro</b>	<b>[</b>	120	254	120	254	255	255	4925
<b>Menu</b>	<b>m</b>	120	254	248	248	255	255	5169
<b>Sottotitoli</b>	<b>°</b>	128	0	0	248	255	255	3925
<b>Sorgente</b>	<b>#</b>	248	0	128	255	255	255	632
<b>Guida</b>	<b>g</b>	248	128	248	255	255	255	1248
<b>Registra</b>	<b>*</b>	128	120	30	248	255	255	4255

Inoltre sono stati creati dei comandi speciali che permettono all'utente di realizzare funzionalità più complesse, utilizzando una combinazione dei comandi precedentemente descritti. Nello specifico sono stati creati comandi vocali che permettono di cambiare alcuni canali pronunciandone solamente il nome, evitando di inviare singole cifre in sequenza allo speech recognizer. Inoltre sono stati realizzati altri tre comandi molto interessanti: "Hard Disk", "Giallo Zafferano" e "Sintonizzazione canali". Il primo permette di accedere direttamente alla periferica USB collegata alla televisione, il secondo esegue l'accesso all'applicazione Giallo Zafferano (funzionalità valida solo per smart tv) ed infine il terzo consente una sintonizzazione automatica dei canali della televisione. Chiaramente ciò permette notevoli vantaggi: immaginiamo ad esempio una persona anziana che deve sintonizzare i canali della propria televisione e si trova difficoltà ad individuare la sequenza di tasti per effettuare l'operazione. Il problema può essere risolto in maniera molto semplice grazie all'invio dell'unico comando vocale "sintonizzazione canali".

In seguito viene mostrata la tabella dei comandi speciali realizzati: la prima colonna viene indicato il comando enunciato dall'utente, nella seconda il carattere inviato dallo smartphone ad Arduino, nelle colonne successive la decodifica in Arduino del carattere ed infine l'id associato al comando:

Comando vocale associato	Carattere	Val 1	Val 2	Val 3	Val 4	Val 5	Val 6	Id comando
Cielo	c	248	0	248	255	255	255	992
Rtl 102.5	p	0	248	-1	255	255	255	493
Real Time	é	248	0	120	224	120	254	3628
DMAX	)	120	224	120	224	255	255	4629
Premium Cinema	j	128	120	254	248	255	255	4927
Premium Calcio	5	120	30	127	224	255	255	4262
Hard Disk	e	120	30	248	248	255	255	4721
Giallo Zafferano	i	120	224	248	248	255	255	5109
Sintonizzazione canali	u	120	30	255	248	255	255	4742

Come per la ricezione del segnale, anche per l'invio è stata utilizzata la libreria IRremote. Il riconoscimento dei comandi è stato gestito tramite uno switch. Il successivo frammento di codice riconosce il comando "Accendi/Spegni" e trasmette il corrispondente segnale alla televisione, il procedimento per gli altri comandi è analogo.

```

1. case 864:
2.     Serial.println("Accendi/Spegni");
3.     irsend.sendSamsung(0xE0E040BF, 32);
4.     delay(100);
5.     break;

```

La riga 3 contiene la funzione sendSamsung, utilizzata per inviare il segnale alla tv, che prende in input come parametri il valore esadecimale del comando (in questo caso accensione e spegnimento)

e il relativo numero di bit. Da notare infine la funzione delay(100) nella riga 4 che permette di gestire il ritardo fra il segnale appena inviato e il successivo segnale da inviare.

### **1.3 Diagramma delle classi applicazione Android**

L'applicazione si compone di 3 classi principali:

1. BluetoothChatService;
2. BluetoothViewer;
3. Device List Activity.

La classe BluetoothChatService è utilizzata per la creazione e la gestione delle connessioni bluetooth. Per realizzare nel modo migliore le due funzionalità vengono utilizzate le due sottoclassi ConnectThread e ConnectedThread; la prima serve per stabilire la connessione, mentre la seconda per gestirla. Di fondamentale importanza è anche il metodo write() utilizzato per inviare il comando ad Arduino.

La classe principale (Main Activity) dell'applicazione è BluetoothViewer, che permette:

1. La creazione della connessione bluetooth;
2. L'attivazione della speech recognition;
3. La gestione del comando inserito;
4. L'invio del comando ad Arduino.

La prima avviene creando un oggetto BluetoothChatService, la speech recognition viene avviata attivando il bottone "mSendButton", il comando inserito è gestito dal metodo whatsend(string), che seleziona il comando da inviare tramite il bluetooth. Infine l'invio del messaggio ad Arduino si verifica attivando il metodo sendmessage(string), che utilizza il metodo write() dell'oggetto BluetoothChatService.

L'ultima classe Device List Activity è un'activity che consente all'utente di visualizzare la lista dei dispositivi associati e da associare.

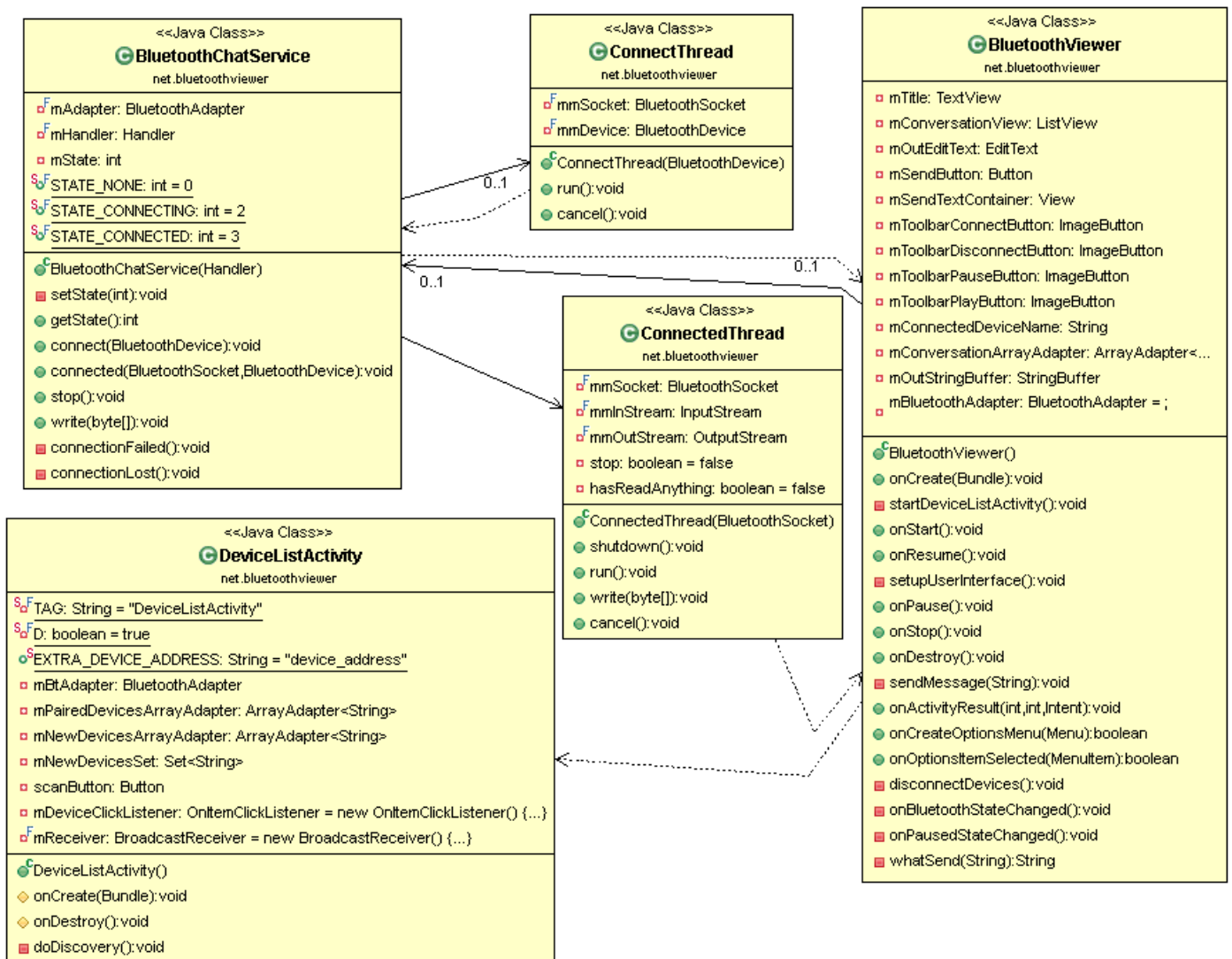


Figura 1.3.1 Diagramma delle classi applicazione

# Capitolo 2

## Casi d'uso e diagrammi

### 2.1 Descrizione casi d'uso

Successivamente vengono presentati i diversi casi d'uso, l'unico attore del sistema è l'utente, ovvero colui che vuole utilizzare il telecomando vocale.

<b>Caso d'uso</b>	Attivazione bluetooth
<b>Attori</b>	Utente
<b>Precondizione</b>	Attivazione Applicazione Bluetooth disattivato
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"><li>1. Il sistema chiede all'utente se vuole attivare il bluetooth</li><li>2. Se l'utente non autorizza l'attivazione del bluetooth:<ol style="list-style-type: none"><li>2.1 Il sistema chiude l'applicazione Android</li><li>2.2 Il caso d'uso termina</li></ol></li><li>3. L'utente autorizza l'attivazione del bluetooth</li><li>4. Il sistema attiva il bluetooth</li></ol>
<b>Post condizione</b>	Il dispositivo è stato associato

<b>Caso d'uso</b>	Prima connessione smartphone e Arduino bluetooth
<b>Attori</b>	Utente
<b>Precondizione</b>	Attivazione Bluetooth
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"><li>1. L'utente richiede la connessione a un dispositivo</li><li>2. Il sistema chiede all'utente se vuole ricercare nuovi dispositivi</li><li>3. L'utente seleziona la ricerca dei dispositivi</li><li>4. Il sistema mostra i dispositivi da associare</li><li>5. L'utente seleziona il dispositivo</li><li>6. Il sistema richiede l'inserimento del pin per l'associazione</li><li>7. L'utente inserisce il pin</li><li>8. Il sistema effettua l'connessione</li></ol>
<b>Post condizione</b>	Il dispositivo è stato connesso

<b>Caso d'uso</b>	Connessione smartphone e Arduino bluetooth
<b>Attori</b>	Utente
<b>Precondizione</b>	Attivazione Bluetooth  Prima connessione smartphone e Arduino bluetooth già effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente richiede la connessione a un dispositivo</li> <li>2. Il sistema propone all'utente la lista dei dispositivi già connessi in precedenza</li> <li>3. L'utente seleziona il dispositivo</li> <li>4. Il sistema connette il dispositivo</li> </ol>
<b>Post condizione</b>	Lo smartphone è connesso

<b>Caso d'uso</b>	Comando generico
<b>Attori</b>	Utente
<b>Precondizione</b>	Connessione bluetooth effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente attiva la speech recognition</li> <li>2. L'utente enuncia un comando</li> <li>3. Se il comando non è riconosciuto: <ol style="list-style-type: none"> <li>3.1 Il sistema mostra sullo schermo dello smartphone che il comando enunciato non è stato riconosciuto</li> <li>3.2 Il caso d'uso termina</li> </ol> </li> <li>4. Il sistema mostra sullo schermo dello smartphone il comando enunciato</li> <li>5. Il sistema invia attraverso Arduino il comando enunciato</li> </ol>
<b>Post condizione</b>	Il comando è stato eseguito

<b>Caso d'uso</b>	Cambio canale da 0 a 9
<b>Attori</b>	Utente
<b>Precondizione</b>	Connessione bluetooth effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente attiva la speech recognition</li> <li>2. L'utente enuncia un numero tra 0 e 9</li> <li>3. Se il comando non è riconosciuto: <ol style="list-style-type: none"> <li>3.1 Il sistema mostra sullo schermo dello smartphone che il comando enunciato non è stato riconosciuto</li> <li>3.2 Il caso d'uso termina</li> </ol> </li> <li>4. Il sistema mostra sullo schermo dello smartphone il comando enunciato</li> <li>5. Il sistema invia attraverso Arduino il comando enunciato</li> </ol>
<b>Post condizione</b>	Il canale è stato cambiato

<b>Caso d'uso</b>	Cambio canale da 10 a 999
<b>Attori</b>	Utente
<b>Precondizione</b>	Connessione bluetooth effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. &lt; Cambio canale da 0 a 9&gt;</li> <li>2. &lt;Cambio canale da 0 a 9&gt;</li> <li>3. Se l'utente desidera visualizzare un canale a 2 cifre: <ol style="list-style-type: none"> <li>3.1 Il caso d'uso termina</li> </ol> </li> <li>4. &lt; Cambio canale da 0 a 9&gt;</li> </ol>
<b>Post condizione</b>	Viene visualizzato un canale da 10 a 999

<b>Caso d'uso</b>	Avvio film da hard disk esterno
<b>Attori</b>	Utente
<b>Precondizione</b>	Connessione bluetooth effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente attiva la speech recognition</li> <li>2. L'utente enuncia il comando "Hard Disk"</li> <li>3. Il sistema mostra sullo schermo dello smartphone il comando enunciato</li> <li>4. Il sistema, attraverso Arduino, invia al televisore i comandi per effettuare l'operazione indicata dall'utente</li> </ol>
<b>Post condizione</b>	Il film è stato avviato dall'hard disk
<b>Sequenza Alternativa</b>	<ol style="list-style-type: none"> <li>1. &lt;Comando generico&gt; Sorgente</li> <li>2. &lt;Comando generico&gt; Basso</li> <li>3. &lt;Comando generico&gt; Basso</li> <li>4. &lt;Comando generico&gt; Basso</li> <li>5. &lt;Comando generico&gt; Ok</li> <li>6. &lt;Comando generico&gt; Ok</li> <li>7. L'utente sceglie il film desiderato utilizzando i comandi generici</li> </ol>

<b>Caso d'uso</b>	Alzare/abbassare volume
<b>Attori</b>	Utente
<b>Precondizione</b>	Connessione bluetooth effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente attiva la speech recognition</li> <li>2. L'utente enuncia il comando "alza volume"/ "abbassa volume"</li> <li>3. Il sistema mostra sullo schermo dello smartphone il comando enunciato</li> <li>4. Il sistema, attraverso Arduino, invia al televisore i comandi per effettuare l'operazione indicata dall'utente</li> </ol>
<b>Postcondizione</b>	Il volume è stato alzato/ abbassato



<b>Caso d'uso</b>	Cambio canale enunciando il nome
<b>Attori</b>	Utente
<b>Precondizione</b>	Connessione bluetooth effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente attiva la speech recognition</li> <li>2. L'utente enuncia il nome del canale</li> <li>3. Se il comando non è riconosciuto: <ol style="list-style-type: none"> <li>3.1 Il sistema mostra sullo schermo dello smartphone che il comando enunciato non è stato riconosciuto</li> <li>3.2 Il caso d'uso termina</li> </ol> </li> <li>4. Il sistema mostra sullo schermo dello smartphone il comando enunciato</li> <li>5. Il sistema, attraverso Arduino, invia al televisore i comandi/il comando per effettuare l'operazione indicata dall'utente</li> </ol>
<b>Postcondizione</b>	Il cambio del canale è stato effettuato

<b>Caso d'uso</b>	Attivazione "Giallo Zafferano" (Applicazione Smart TV)
<b>Attori</b>	Utente
<b>Precondizione</b>	Connessione bluetooth effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente attiva la speech recognition</li> <li>2. L'utente enuncia il comando "Giallo Zafferano"</li> <li>3. Se il comando non è riconosciuto: <ol style="list-style-type: none"> <li>3.1 Il sistema mostra sullo schermo dello smartphone che il comando enunciato non è stato riconosciuto</li> <li>3.2 Il caso d'uso termina</li> </ol> </li> <li>4. Il sistema mostra sullo schermo dello smartphone il comando enunciato</li> <li>5. Il sistema, attraverso Arduino, invia al televisore i comandi per effettuare l'operazione indicata dall'utente</li> </ol>
<b>Postcondizione</b>	La Smart tv mostra il browser
<b>Sequenza Alternativa</b>	<ol style="list-style-type: none"> <li>1. &lt;Comando generico&gt; Smart</li> <li>2. &lt;Comando generico&gt; Giù</li> <li>3. &lt;Comando generico&gt; Sinistra</li> <li>4. &lt;Comando generico&gt; Sinistra</li> <li>5. &lt;Comando generico&gt; Ok</li> </ol>

<b>Caso d'uso</b>	Sintonizzazione canali
<b>Attori</b>	Utente
<b>Precondizione</b>	Connessione bluetooth effettuata
<b>Sequenza degli eventi</b>	<ol style="list-style-type: none"> <li>1. L'utente attiva la speech recognition</li> <li>2. L'utente enuncia il comando "Sintonizzazione Canali"</li> <li>3. Se il comando non è riconosciuto: <ol style="list-style-type: none"> <li>3.1 Il sistema mostra sullo schermo dello smartphone che il comando enunciato non è stato riconosciuto</li> <li>3.2 Il caso d'uso termina</li> </ol> </li> <li>4. Il sistema mostra sullo schermo dello smartphone il comando enunciato</li> <li>5. Il sistema, attraverso Arduino, invia al televisore i comandi per effettuare l'operazione indicata dall'utente</li> </ol>
<b>Post condizione</b>	La sintonizzazione è stata effettuata
<b>Sequenza Alternativa</b>	<ol style="list-style-type: none"> <li>1. &lt;Comando generico&gt; Menu</li> <li>2. &lt;Comando generico&gt; Giù</li> <li>3. &lt;Comando generico&gt; Giù</li> <li>4. &lt;Comando generico&gt; Destra</li> <li>5. &lt;Comando generico&gt; Giù</li> <li>6. &lt;Comando generico&gt; Giù</li> <li>7. &lt;Comando generico&gt; Ok</li> <li>8. &lt;Comando generico&gt; Ok</li> <li>9. &lt;Comando generico&gt; Ok</li> <li>10. &lt;Comando generico&gt; Ok</li> <li>11. &lt;Comando generico&gt; Ok</li> </ol>

## 2.2 Diagrammi di attività

Successivamente vengono illustrati i diagrammi di attività relativi ad alcuni casi d'uso precedentemente descritti. I diagrammi di attività rappresentano i diversi stati del sistema:

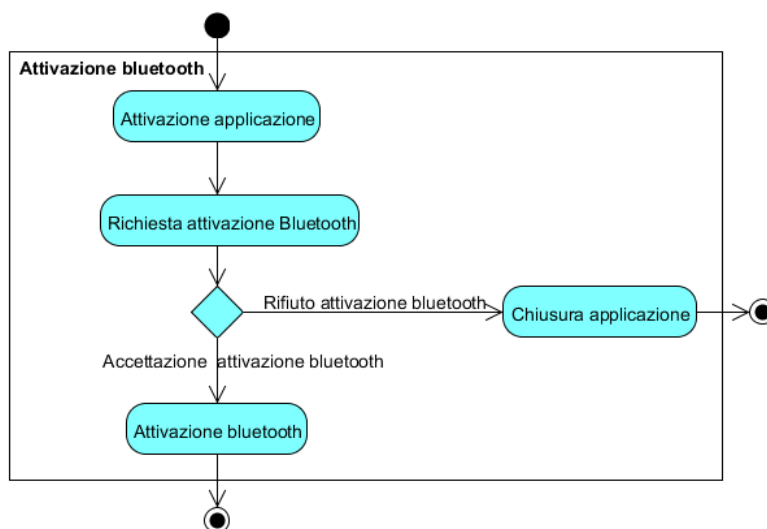


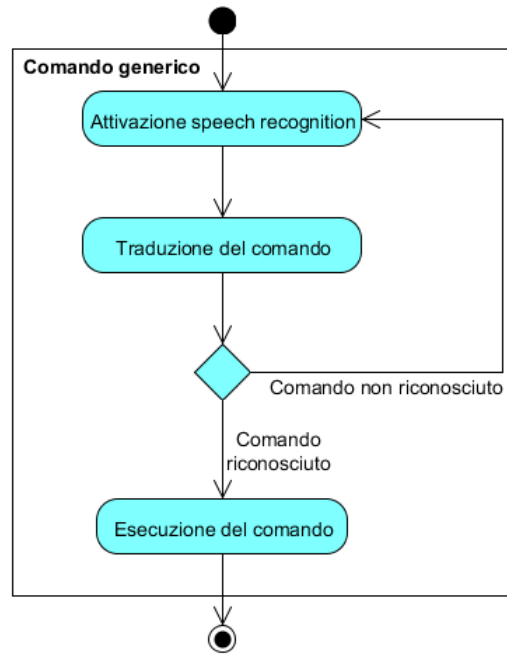
Figura 2.2.1 Diagramma attività attivazione bluetooth



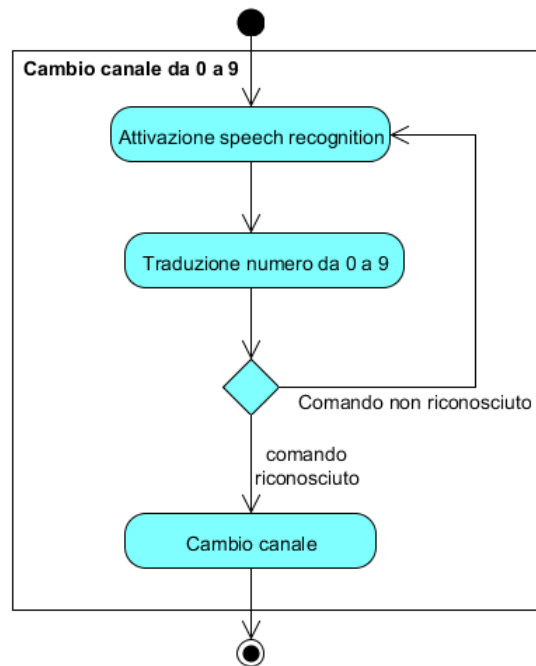
*Figura 2.2.2 Diagramma attività Prima connessione smartphone Arduino*



*Figura 2.2.3 Diagramma attività connessione smartphone Arduino*



*Figura 2.2.4 Diagramma attività Comando generico*



*Figura 2.2.5 Diagramma attività cambio canale da 0 a 9*

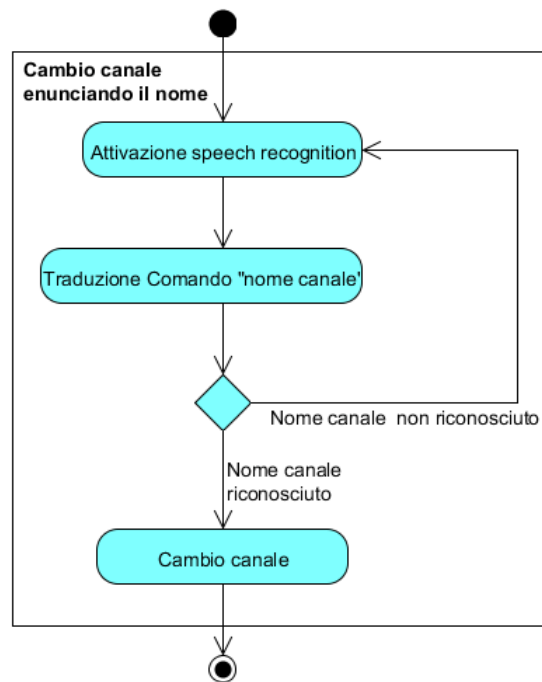


Figura 2.2.6 Diagramma attività, cambio canale enunciando il nome

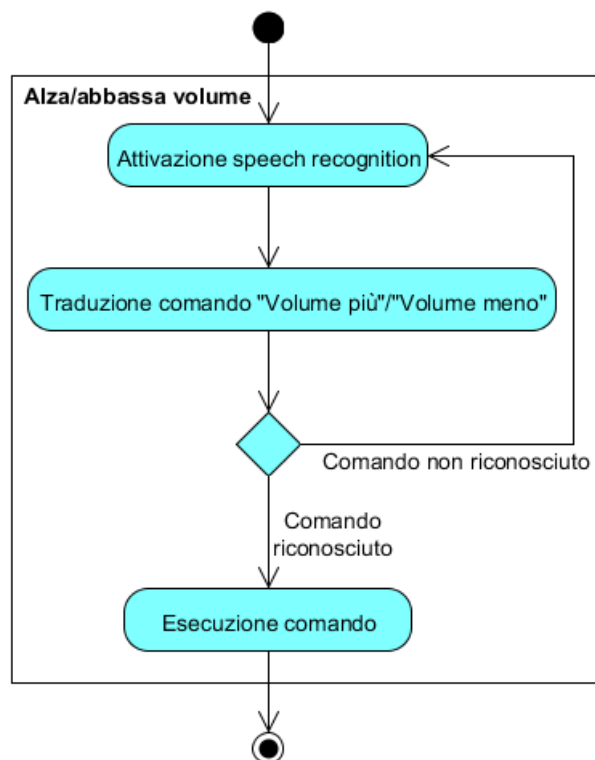


Figura 2.2.7 Diagramma attività Alza/Abbassa volume

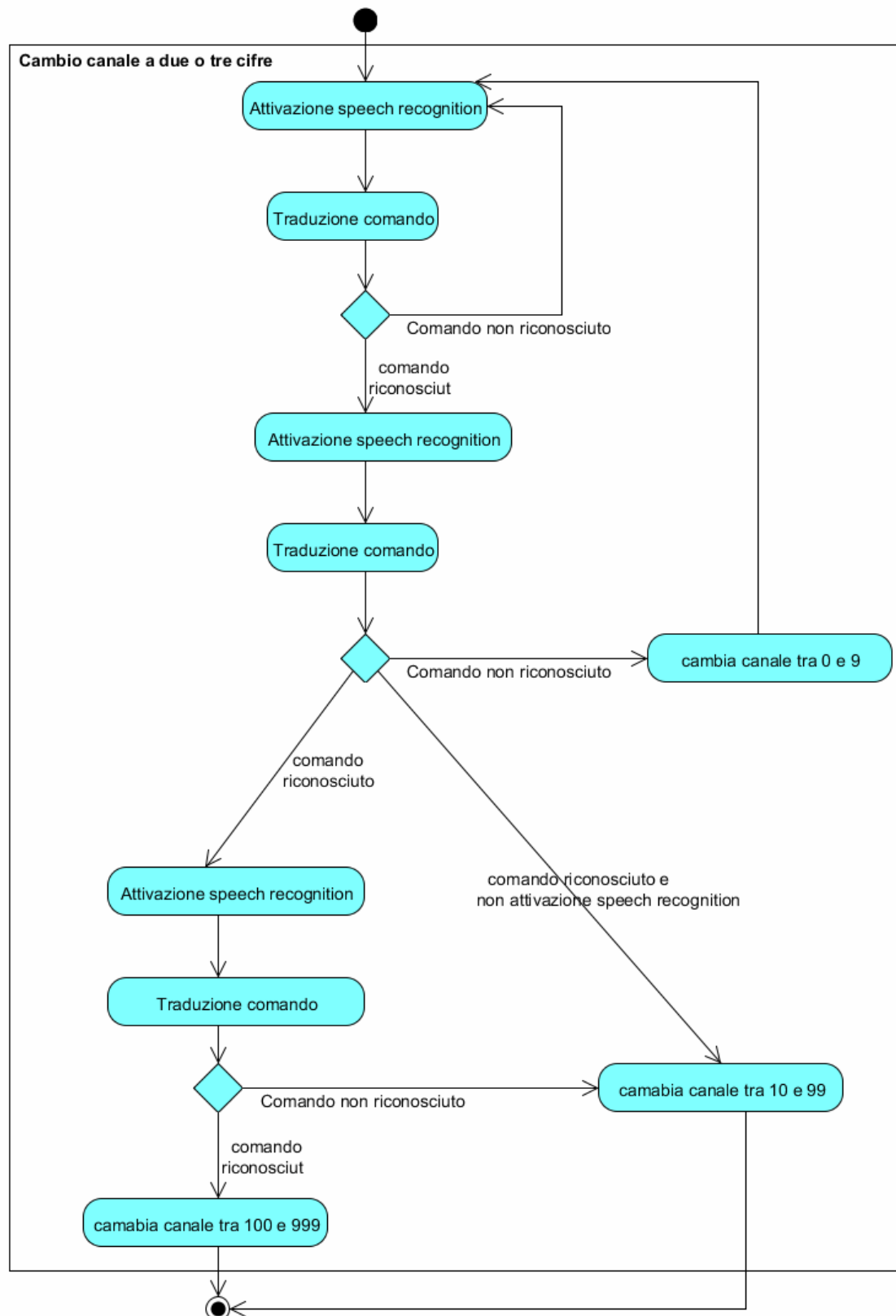


Figura 2.2.8 Diagramma attività, cambio canale a 2 o più cifre

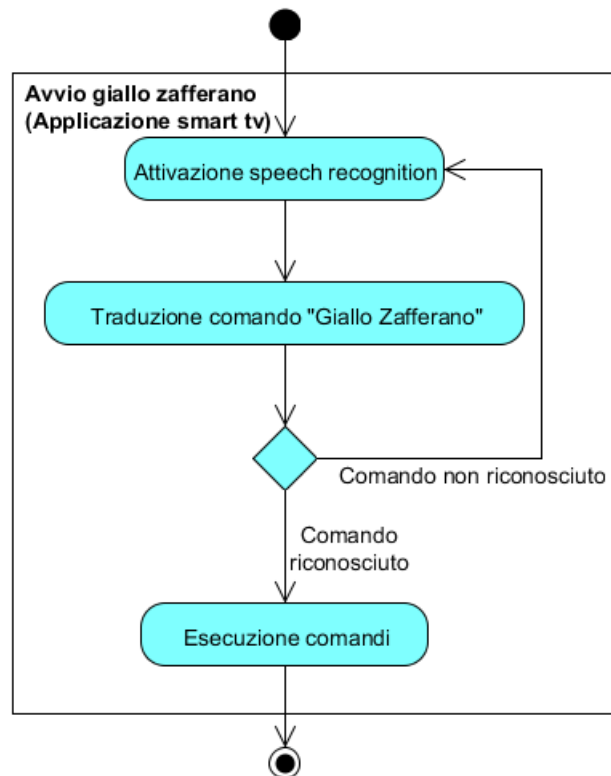


Figura 2.2.9 Diagramma attività, Avvio giallo zafferano (Applicazione Smart tv)

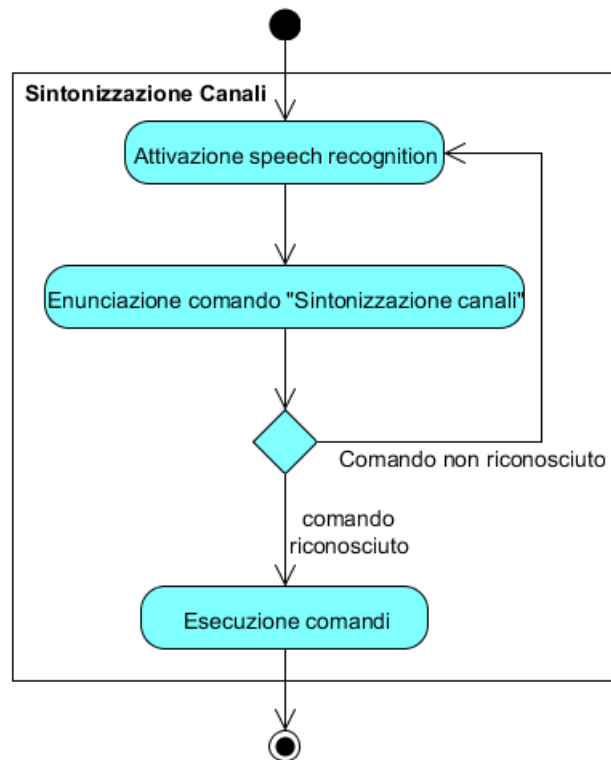
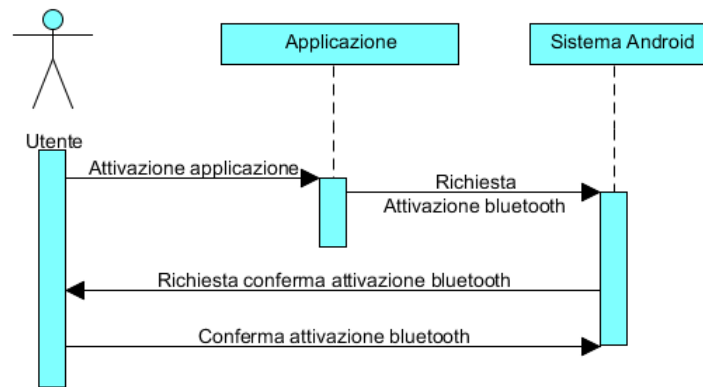


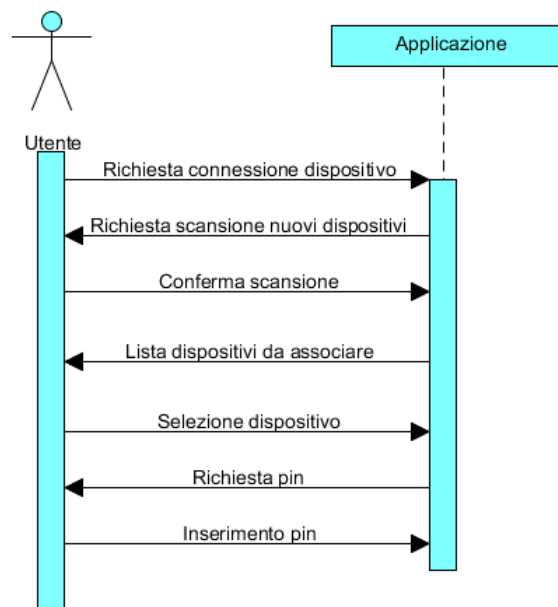
Figura 2.2.10 Diagramma attività Sintonizzazione canali

## 2.3 Diagrammi di sequenza

Successivamente vengono illustrati i diagrammi di sequenza relativi ad alcuni casi d'uso precedentemente descritti. I diagrammi di sequenza rappresentano le interazioni tra gli oggetti del sistema:



*Figura 2.3.1 Diagramma di sequenza Attivazione bluetooth*



*Figura 2.3.2 Diagramma di sequenza Prima connessione dispositivo*



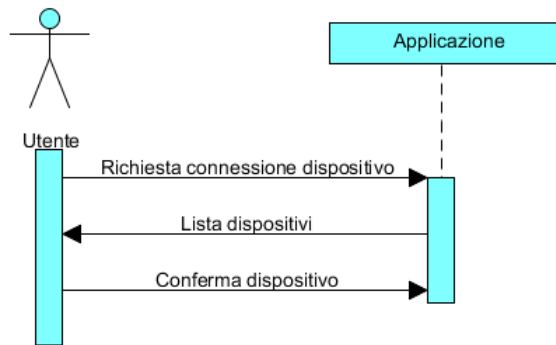


Figura 2.3.3 Diagramma di sequenza Connessione bluetooth

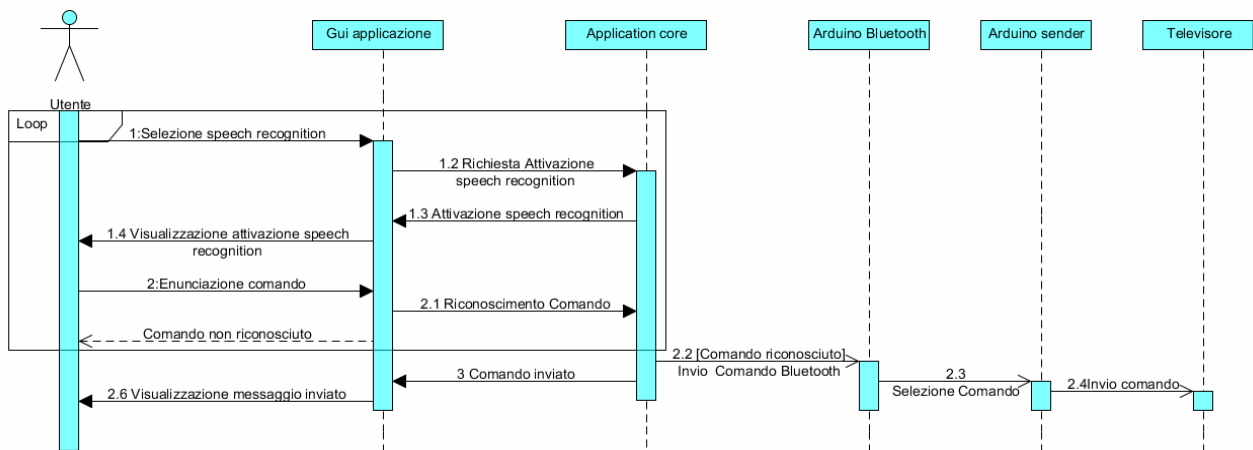


Figura 2.3.4 Diagramma di sequenza invio comando generico

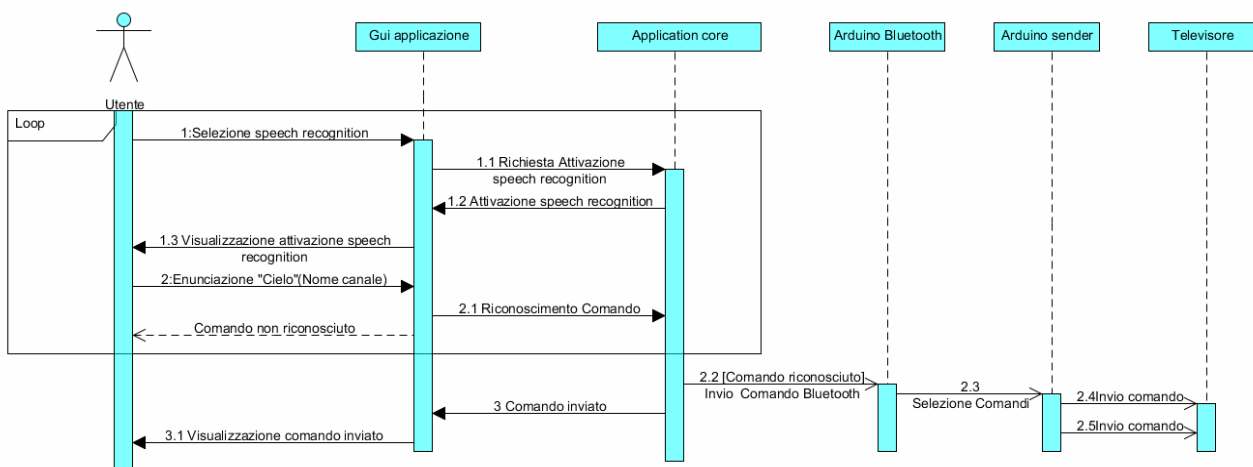


Figura 2.3.5 Diagramma di sequenza cambio canale enunciando il nome

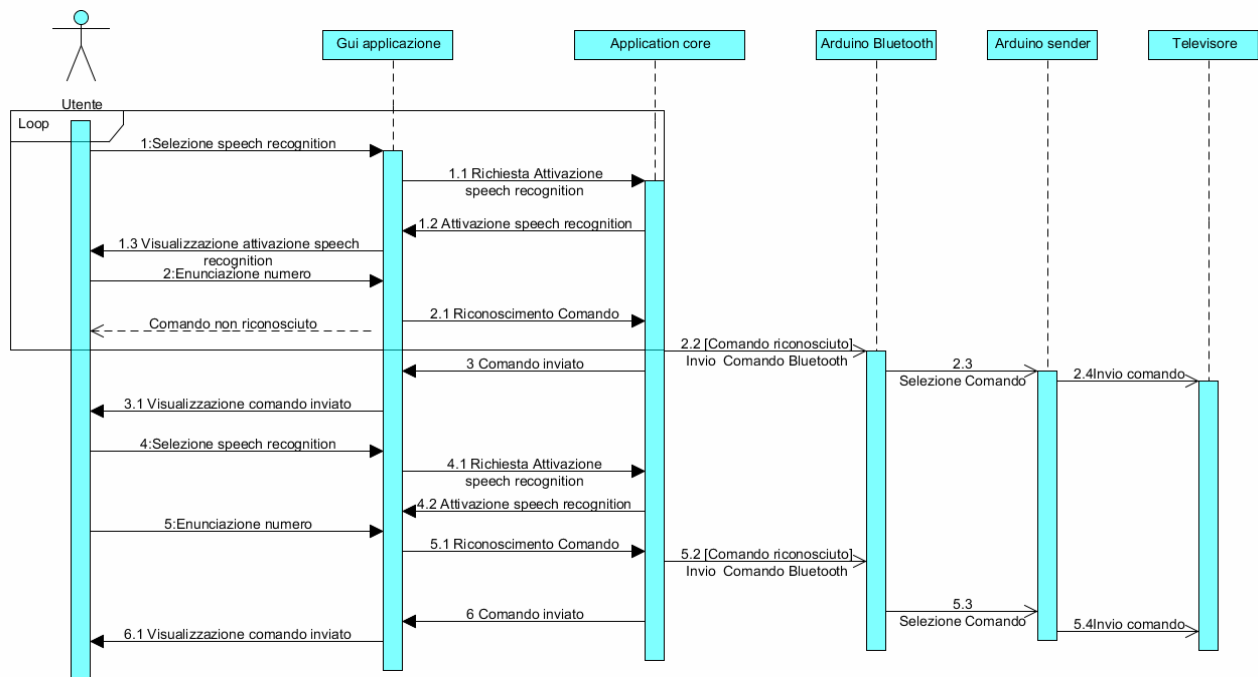


Figura 2.3.6 Diagramma di sequenza cambio canale a due cifre

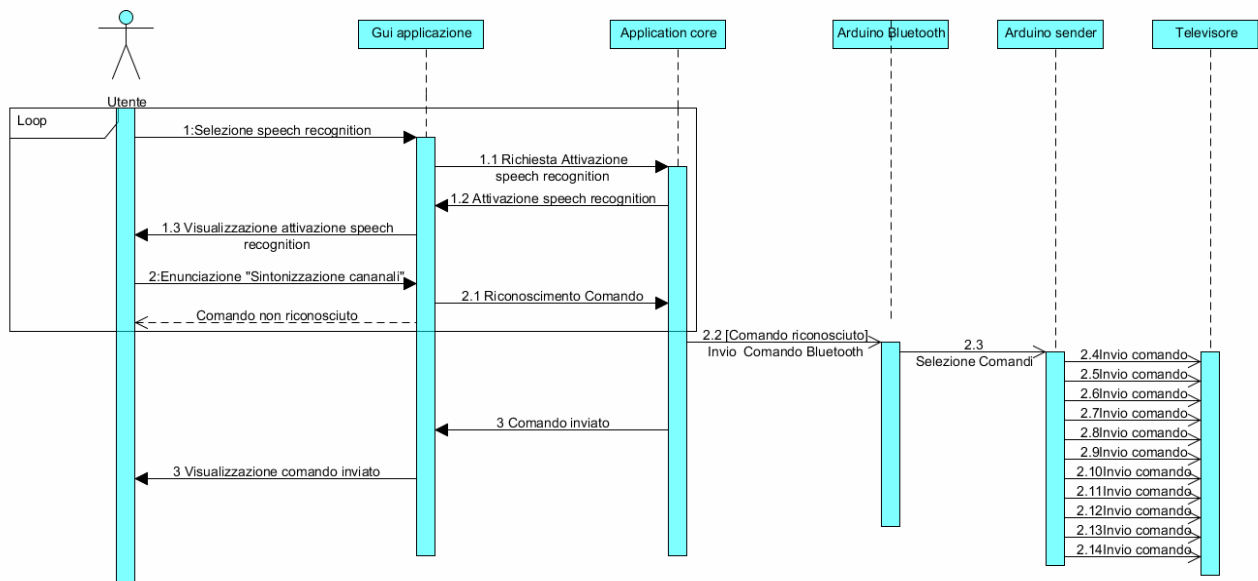


Figura 2.3.7 Diagramma di sequenza Sintonizzazione canale

# Capitolo 3

## Scenari

In questo capitolo vengono presentati alcuni possibili scenari di utilizzo dell'applicazione. Gli scenari descritti sono i seguenti:

- Scenario 1: Accesso all'applicazione e apprendimento delle prime funzionalità
- Scenario 2: Avvio film da hard disk
- Scenario 3: Interazione con un'applicazione presente su una smart tv

### 3.1 Scenario 1: Accesso all'applicazione e apprendimento delle prime funzionalità

Andrea ha appena installato l'applicazione BlueCommand sul proprio smartphone e vuole iniziare a prendere confidenza con le prime funzionalità.



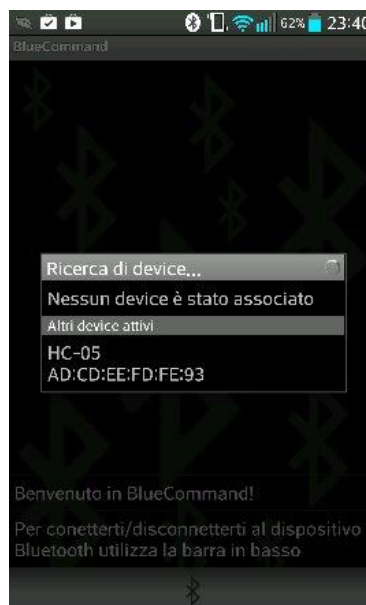
**Schermata 1 Applicazione** – Schermata all'apertura dell'applicazione, viene mostrato un messaggio in cui viene chiesto all'utente l'autorizzazione per attivare la connessione bluetooth. Se l'utente preme il pulsante "Sì" può passare alla prossima schermata, altrimenti se clicca "no" l'applicazione si chiude.

Andrea adesso può premere il bottone per associare il modulo bluetooth di arduino al proprio smartphone.



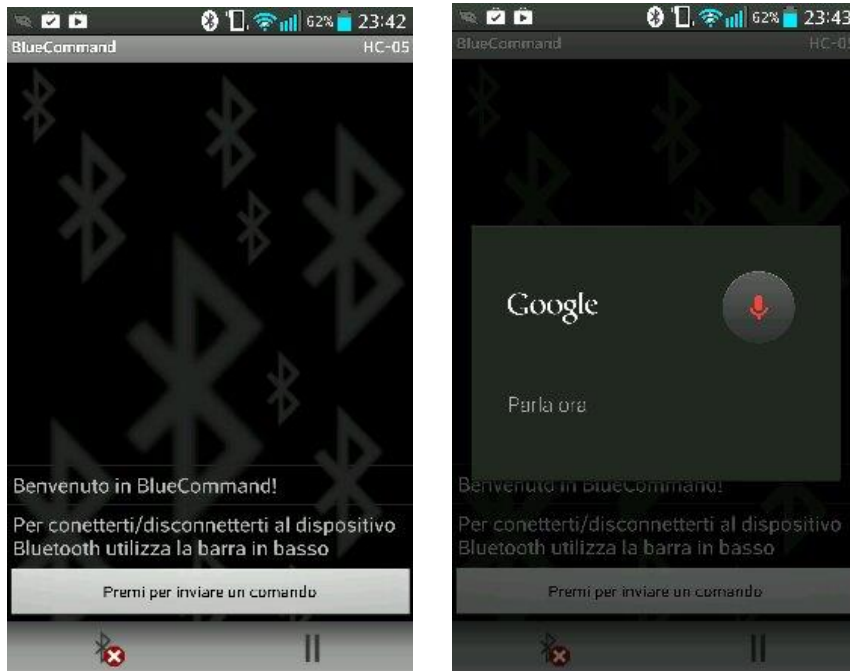
**Schermata 2 Applicazione** – Questa schermata mostra un messaggio di benvenuto all’utente e presenta un pulsante con raffigurato un simbolo bluetooth per connettersi a un dispositivo.

Inizialmente nessun device è stato ancora associato. Andrea decide dunque di cercare device nelle vicinanze e aspetta che vengano visualizzati dispositivi bluetooth sullo schermo. Successivamente associa il modulo bluetooth (HC-05) di arduino al proprio smartphone inserendo il PIN.



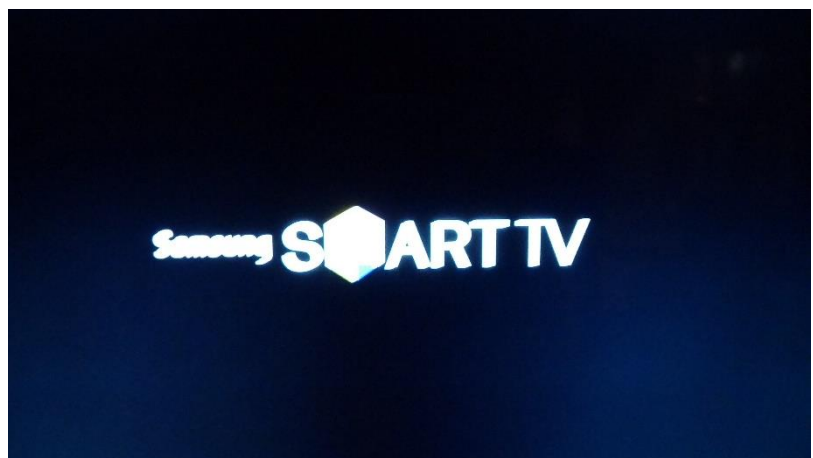
**Schermate 3, 4, 5 Applicazione** – La schermata 3 mostra un messaggio dei device attualmente associati. Se nessuno dispositivo è stato associato allora viene proposto all’utente di cercare dispositivi nelle vicinanze mediante il pulsante “Ricerca di device”. La schermata 4 mostra i device trovati. L’utente premendo sul nome del dispositivo sceglie quale device associare. La schermata 5 permette all’utente di inserire il PIN segreto per abbinare il device.

A questo punto Andrea clicca sul bottone per attivare lo speech recognizer e il programma resta in attesa di un comando vocale.



**Schermate 6, 7 Applicazione** – La schermata 6 presenta il pulsante “Premi per inviare un comando” con cui l’utente può attivare lo speech recognizer. La schermata 7 mostra una schermata con lo speech recognizer abilitato, in attesa di ricevere un comando vocale da parte dell’utente.

Andrea desidera accendere il suo televisore samsung e pronuncia il comando “accendi”. La televisione si accende.



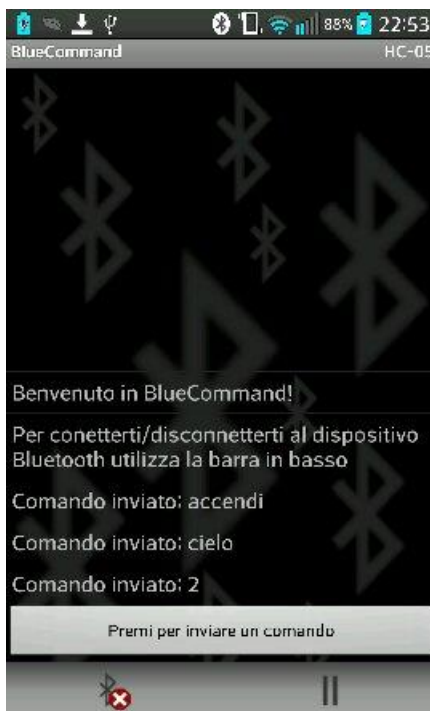
**Schermata 8 Applicazione, 9 Televisione** – Nella schermata 8 viene visualizzato il comando vocale “accendi” pronunciato dall’utente sullo schermo dell’applicazione. La schermata 9 mostra la conseguente accensione della televisione.

Andrea trova la televisione impostata sul canale 4, tuttavia non è interessato alla visione del film attualmente in corso e decide di pronunciare il comando “cielo” per verificare se c’è un programma di suo interesse su quel canale.



**Schermata 10 Applicazione, 11 Televisione** – Nella schermata 10 viene visualizzato il comando vocale “cielo”[11] pronunciato dall’utente sullo schermo dell’applicazione. Nella schermata 11 viene effettuato il cambio del canale.

Neanche questo programma lo entusiasma molto e decide dunque di pronunciare il comando “due”.



**Schermata 12 Applicazione, 13 Televisione** – Nella schermata 12 viene visualizzato il comando vocale “2” pronunciato dall’utente sullo schermo dell’applicazione. Nella schermata 13 viene effettuato il cambio del canale.

Stavolta il programma gli suscita molta curiosità, ma non riesce a sentire bene la televisione perché il volume è troppo basso. Decide di pronunciare il comando “volume più” per tre volte alzando così il volume da 6 a 9.



**Schermata 14 Applicazione, 15 Televisione** – Nella schermata 14 viene visualizzato il comando vocale “volume più” pronunciato dall’utente insieme ai precedenti comandi. La schermata 15 mostra il volume che si alza.

## 3.2 Scenario 2: Avvio film da hard disk

Alessandro ha già installato e configurato l’applicazione sul proprio telefono e vuole vedere un film dal proprio hard disk collegato alla televisione.

A questo punto ha due possibilità:

- Pronuncia la serie di comandi vocali per accedere alla memoria dell’hard disk;
- Pronuncia il comando diretto “Hard Disk”.

Alessandro decide di pronunciare il comando “Hard Disk” ed entra nella cartella contenente i film.





**Schermata 16 Applicazione, 17 Televisione** – Nella schermata 16 viene visualizzato il comando vocale “hard disk” pronunciato dall’utente sullo schermo dell’applicazione. Nella schermata 17 viene eseguito l’accesso all’hard disk.

Sull’hard disk sono presenti diversi film, alessandro attraverso la serie di comandi vocali “basso”, “basso”, “basso”, “destra”, “ok” decide di voler vedere il suo film preferito “The Avengers”. Il film inizia.



**Schermata 18 Applicazione, 19 Televisione** – Nella schermata 18 viene visualizzata la serie di comandi vocali pronunciati dall’utente per avviare il film desiderato. La schermata 19 mostra la cartella dell’hard disk contenente i film e il filmato selezionato prima di pronunciare l’ultimo comando “ok”.

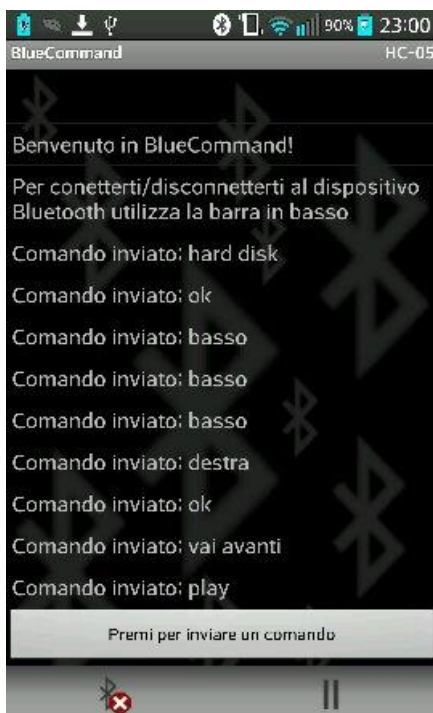


Alessandro non vuole vedere i minuti iniziali del film ritenendoli noiosi, invoca dunque il comando “Vai avanti”. Il film si avvolge in avanti.



**Schermata 20 Applicazione, 21 Televisione** – Nella schermata 20 viene visualizzato il comando vocale “vai avanti” pronunciato dall’utente sullo schermo dell’applicazione. Nella schermata 21 il film va avanti come confermato dal messaggio in cima allo schermo della televisione.

Alessandro pronuncia “play” e ricomincia a guardare il film.



**Schermata 22 Applicazione, 23 Televisione** – Nella schermata 22 viene visualizzato il comando vocale “play” pronunciato dall’utente sullo schermo dell’applicazione. Nella schermata 23 il film riparte come confermato dal messaggio in cima allo schermo della televisione.

### 3.3 Scenario 3: Interazione con un'applicazione presente su una Smart tv

Maria possiede una Smart tv[7] Samsung e vuole accedere all'applicazione Giallo Zafferano, in quanto è interessata a cucinare una ricetta speciale per degli ospiti che mangeranno a casa sua.

Le se presentano due possibilità:

- Pronuncia la serie di comandi vocali per accedere all'applicazione;
- Pronuncia il comando diretto "Giallo Zafferano".

Maria sceglie la seconda opzione e, dopo aver atteso il caricamento dell'applicazione, nota subito una ricetta che la entusiasma: il filetto alla piastra.



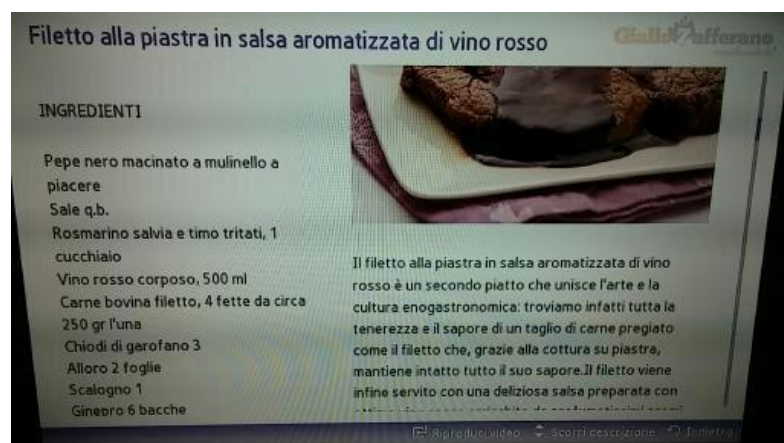
Schermata 24 Applicazione, 25 Televisione – Nella schermata 24 viene visualizzato il comando vocale “Giallo Zafferano” pronunciato dall’utente sullo schermo dell’applicazione. Nella schermata 25 l’applicazione “Giallo Zafferano” è stata aperta e l’utente può scegliere una ricetta pronunciato la serie di comandi desiderata.

Decide quindi di accedere alla ricetta pronunciando “Ok”.



**Schermata 26 Applicazione, 27 Televisione – Nella schermata 26 viene visualizzato il comando vocale “ok” pronunciato dall’utente sullo schermo dell’applicazione. La schermata 27 mostra la ricetta desiderata dall’utente in seguito al comando vocale pronunciato.**

Maria, molto interessata alla preparazione della ricetta, decide di pronunciare il comando “Basso” per poter proseguire la lettura. Una volta visualizzati gli ingredienti, Maria decide di appuntarli su un foglio di carta e andare a comprarli al supermercato.



**Schermata 28 Applicazione, 29 Televisione – Nella schermata 28 viene visualizzato il comando vocale “basso” pronunciato dall’utente sullo schermo dell’applicazione. Nella schermata 29 viene effettuato lo scrolling verso il basso della pagina.**

# Conclusioni

Lo sviluppo del sistema è stata un'ottima opportunità per approfondire lo sviluppo di applicazioni Android e per iniziare ad approfondire alcuni aspetti legati ad Arduino.

L'obiettivo del progetto è stato realizzare un telecomando vocale che permette la gestione di una smart tv Samsung. A tale scopo sono stati utilizzati una scheda Arduino e uno smartphone Android; la prima viene utilizzata per inviare il segnale infrarossi alla tv mentre la seconda per effettuare il riconoscimento vocale. Molto importante è stato facilitare l'utilizzo del televisore per l'utente, semplificando notevolmente alcune funzionalità complesse; infatti l'utente, enunciando un solo comando, effettua operazioni che con un normale telecomando risulterebbero poco immediate.

## Possibili sviluppi futuri

Il sistema sviluppato è un prototipo per gestire tv della Samsung, ma in futuro si potranno comandare altri tipi di tv cambiando solamente parte del codice Arduino.

Oltre alla tv si possono essere gestiti altri dispositivi come decoder, home theatre e lettori Dvd. In particolare per i decoder il sistema sarà utilizzato come estensore di telecomando; il bluetooth permette al segnale di passare attraverso le pareti, quindi basterà posizionare la scheda Arduino vicino al decoder e utilizzare lo smartphone in un'altra stanza dove arriva già il segnale audio e video per poter cambiare canale.

Il sistema sviluppato potrà gestire qualunque dispositivo sia in grado di ricevere comandi attraverso una porta infrarossi. Si potranno comandare più dispositivi contemporaneamente con l'utilizzo della voce, per esempio: tv, decoder, serrande elettriche, climatizzatore, pc, ecc.

Infine negli ultimi mesi alcuni smartphone di fascia alta dispongono di una porta infrarossi, in questi non sarà più necessario avere a disposizione la scheda Arduino. Lo smartphone Android integrerà nell'applicazione, sia la traduzione del comando, sia l'invio alla tv o ad altro elettrodomestico. Con questa modalità sarà più semplice vendere il sistema, gestire i codici per comandare i dispositivi e basterà scaricare un'applicazione dal playstore di Google per poter gestire molti elettrodomestici.

# Bibliografia

1. Android Api bluetooth <http://developer.android.com/guide/topics/connectivity/bluetooth.html>
2. Arduino uno <http://arduino.cc/en/Main/arduinoBoardUno>
3. Libreria iremote <https://github.com/shirriff/Arduino-IRremote>
4. Tutorial Arduino ir sender ir receiver <http://garagelab.com/profiles/blogs/tutorial-arduino-ir-sender-and-receiver>
5. Gestione infrarossi <https://learn.sparkfun.com/tutorials/ir-communication/all>
6. Comunicazione Android e Arduino bluetooth <http://bellcode.wordpress.com/2012/01/02/android-and-arduino-bluetooth-communication/>
7. Smart tv samsung <http://www.samsung.com/it/consumer/tv-audio-video/smart-tv/>
8. Modulo bluetooth HC-05 <http://www.exp-tech.de/service/datasheet/HC-Serial-Bluetooth-Products.pdf>
9. Api speech recognition <http://developer.android.com/reference/android/speech/package-summary.html>
10. Android speech recognition esempi <http://android-developers.blogspot.it/2009/09/introduction-to-text-to-speech-in.html>
11. LCN <http://www.sviluppoeconomico.gov.it/images/stories/digitale/nazionali.pdf>
12. Lg Magic Remote Control <http://www.lg.com/global/magicremote/>