TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**VIỆN ĐIỆN TỬ - TRUYỀN THÔNG**



ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**ỨNG DỤNG ĐIỀU KHIỂN TIVI BẰNG GIỌNG NÓI THÔNG QUA RASPBERRY PI**

Sinh viên thực hiện:Trần Ngọc Tiến

Lớp ĐTTT04 - K58

Giảng viên hướng dẫn: TS. Nguyễn Tiến Hòa

Hà Nội, 3 - 2018

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**VIỆN ĐIỆN TỬ - TRUYỀN THÔNG**



ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**ỨNG DỤNG ĐIỀU KHIỂN TIVI BẰNG GIỌNG NÓI THÔNG QUA RASPBERRY PI**

Sinh viên thực hiện:Trần Ngọc Tiến

Lớp ĐTTT04 - K58

Giảng viên hướng dẫn: TS. Nguyễn Tiến Hòa

Cán bộ phản biện: Nguyễn Văn A

Hà Nội, 3 - 2018

**Đánh giá quyển đồ án tốt nghiệp**

**(Dùng cho giảng viên hướng dẫn)**

Giảng viên đánh giá:......................................................

Họ và tên Sinh viên:................................................ MSSV:…………………

Tên đồ án: ................................... ................................... ...................................

…………………………………………………………………………………..

***Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:***

***Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết (10)** | | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | | |
| 10a | | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | | |  | | | | |

***3. Nhận xét thêm của Thầy/Cô (****giảng viên hướng dẫn nhận xét về thái độ và tinh thần làm việc của sinh viên****)***

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

Ngày: / /201

Người nhận xét

(Ký và ghi rõ họ tên)

**Đánh giá quyển đồ án tốt nghiệp**

**(Dùng cho cán bộ phản biện)**

Giảng viên đánh giá:......................................................

Họ và tên Sinh viên:................................................ MSSV:…………………

Tên đồ án: ................................... ................................... ...................................

…………………………………………………………………………………..

***Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:***

***Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết (10)** | | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | | |
| 10a | | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | | |  | | | | |

***3. Nhận xét thêm của Thầy/Cô***

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

Ngày: / /201

Người nhận xét

(Ký và ghi rõ họ tên)

**LỜI NÓI ĐẦU**

**TÓM TẮT ĐỒ ÁN**

MỤC LỤC

[DANH MỤC HÌNH VẼ 3](#_Toc509610353)

[DANH MỤC BẢNG BIỂU 4](#_Toc509610354)

[MỞ ĐẦU 5](#_Toc509610355)

[CHƯƠNG 1: LÝ THUYẾT 6](#_Toc509610356)

[1.1 Socket.IO 6](#_Toc509610357)

[1.2 Nodejs 7](#_Toc509610358)

[1.3 Thư viện LIRC (Linux Infrared remote control) 8](#_Toc509610359)

[CHƯƠNG 2: THIẾT KẾ SƠ ĐỒ KHỐI 9](#_Toc509610360)

[CHƯƠNG 3: THIẾT KẾ CÁC KHỐI 11](#_Toc509610361)

[3.1 Ứng dụng trên điện thoại 11](#_Toc509610362)

[3.1.1 Danh sách kênh 12](#_Toc509610363)

[3.1.2 Thêm kênh, sửa kênh 18](#_Toc509610364)

[3.1.3 Sắp xếp kênh 22](#_Toc509610365)

[3.1.4 Ra lệnh điều khiển 23](#_Toc509610366)

[3.2 Module thu phát hồng ngoại 28](#_Toc509610367)

[3.3 Server trên Raspberry Pi 29](#_Toc509610368)

[3.3.1 Thư viện LIRC 29](#_Toc509610369)

[3.3.2 Server Nodejs trên Raspberry Pi 32](#_Toc509610370)

[KẾT LUẬN CHUNG 35](#_Toc509610371)

[TÀI LIỆU THAM KHẢO 36](#_Toc509610372)

[PHỤ LỤC 37](#_Toc509610373)

# DANH MỤC HÌNH VẼ

[Hình 1: Sơ đồ hoạt động của hệ thống 7](#_Toc508749086)

[Hình 2: Giao diện chính của ứng dụng 9](#_Toc508749087)

[Hình 3: Thành phần chức năng của ứng dụng 10](#_Toc508749088)

[Hình 4: Sơ đồ xây dựng một ListView 11](#_Toc508749089)

[Hình 5: Chức năng thêm kênh 16](#_Toc508749090)

[Hình 6: Chức năng sửa kênh 19](#_Toc508749091)

[Hình 7: Chức năng sắp xếp kênh 20](#_Toc508749092)

[Hình 8: Chức năng ra lệnh bằng giọng nói 21](#_Toc508749093)

[Hình 9: Sơ đồ hoạt động của Button Voice 22](#_Toc508749094)

[Hình 10: Mắt thu hồng ngoại TSOP1838 26](#_Toc508749095)

[Hình 11: Sơ đồ mạch phát tín hiệu hồng ngoại 26](#_Toc508749096)

[Hình 12: Cách hoạt động của Server Nodejs 30](#_Toc508749097)

# DANH MỤC BẢNG BIỂU

# MỞ ĐẦU

Nhà thông minh (Smart Home) là một ngôi nhà/căn hộ được trang bị hệ thống tự động tiên tiến dành cho điều khiển đèn chiếu sáng, nhiệt độ, truyền thông đa phương tiện, an ninh, rèm cửa, cửa và nhiều tính năng khác nhằm mục đích làm cho cuộc sống ngày càng tiện nghi, an toàn và góp phần sử dụng hợp lý các nguồn tài nguyên.

Hiện nay các hệ thống nhà thông minh đều cho phép người dùng điều khiển các thiết bị điện trong nhà thông qua ứng dụng trên điện thoại, tuy nhiên số lượng hệ thống cho phép người dùng điều khiển bằng giọng nói tiếng việt là chưa nhiều. Mặt khác, các hệ thống nhà thông minh hiện nay chỉ mới hỗ trợ điều khiển các thiết bị điện dưới dạng bật/tắt, chưa cho phép người dùng điều khiển sâu hơn chức năng của các thiết bị đó. Chẳng hạn, với thiết bị tivi không phải là smart tivi thì các hệ thống mới chỉ hỗ trợ bật/tắt tivi chứ chưa hỗ trợ chuyển kênh hay tăng âm lượng, giảm âm lượng…Hay với điều hòa cũng chưa hỗ trợ các chức năng như tăng/giảm nhiệt độ, hướng gió, tốc độ thổi,…

Vấn đề mà đồ án này giải quyết đó chính là thiết kế một mô hình hệ thống nhà thông minh đơn giản hỗ trợ người dùng điều khiển các thiết bị điện trong gia đình bằng giọng nói tiếng việt thông qua ứng dụng trên điện thoại. Mặt khác, hệ thống nhà thông minh có thể điều khiển các thiết bị điện ở mức sâu nhất, có nghĩa là người dùng có thể điều khiển các chức năng của các thiết bị như chuyển kênh tivi, tăng/giảm âm lượng, tăng/giảm nhiệt độ điều hòa, điều chỉnh tốc độ gió điều hòa…

Phương pháp mà đồ án sử dụng để giải quyết các vấn đề đó là sử dụng hệ thống nhận dạng ngôn ngữ tiếng việt của Google để lấy kết quả xử lý từ đó chuyển thành lệnh điều khiển. Đối với việc điều khiển sâu các chức năng của thiết bị điện, phương pháp ở đây là thiết kệ bộ phát hồng ngoại tương tự như remote của thiết bị đó và bộ phát này phải kết nối được với hệ thống nhà thông minh để nhận lệnh từ người dùng.

# CHƯƠNG 1: LÝ THUYẾT

## Socket.IO

Socket.IO là một thư viện javascript có mục đích tạo ra các ứng dụng realtime trên trình duyệt cũng như thiết bị di động. Việc sử dụng thư viện này cũng rất đơn giản và giống nhau ở cả server lẫn client.

Server: tạo một đối tượng socket bằng phương thức ***listen(port).*** Phương thức này chờ đợi một yêu cầu kết nối từ client.

- Client: Kết nối đến server bằng phương thức ***connect(url,{port: server\_port}).***

- Socket.IO cung cấp 3 event chính là ***connect, message và disconnect***. Chúng được kích hoạt khi client/server:

* ***connect:*** tạo kết nối
* ***message:*** nhận được thông điệp
* ***disconnect:*** ngắt kết nối

**Ví dụ:** Khai báo cho socket nhận một sự kiện “message”

socket.on("message", function(msg){

// console.log("Received: "+ msg);

});

Socket.IO có thể gửi và nhận các event tự tạo với phương thức ***emit().*** Hai phía gửi và nhận phải biết được tên của event đó để thực hiện giao tiếp:

**Ví dụ:**

// client gửi một dòng message "welcome" lên event "hello"

socket.emit("hello", {msg: "welcome"});

// Server nhận sự kiện event đưa lên

socket.on("hello", function (data) {

console.log(data);});

## Nodejs

Node.js là một hệ thống được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, sử dụng kỹ thật điều khiển theo sự kiện, nhập/xuất không đồng bộ để tối thiểu tổng chi phí và tối đại khả năng mở rộng. Node.js bao gồm có V8 JavaScript engine của Google, libUV, và vài thư viện khác.

Node.js được tạo bởi Ryan Dahl từ năm 2009, và phát triển dưới sự bảo trợ của Joyent.

Mục tiêu ban đầu của Dahl là làm cho trang web có khả năng push như trong một số ứng dụng web như Gmail. Sau khi thử với vài ngôn ngữ Dahl chọn Javascript vì một API Nhập/Xuất không đầy đủ. Điều này cho phép anh có thể định nghĩa một quy ước Nhập/Xuất điểu khiển theo sự kiện, non-blocking.

Vài môi trường tương tự được viết trong các ngôn ngữ khác bao gồm Twisted cho Python, Perl Object Environment cho Perl, libevent cho C và EventMachine cho Ruby. Khác với hầu hết các chương trình Javascript, Nodejs không chạy trên một trình duyệt mà chạy trên Server. Node.js sử dụng nhiều chi tiết kỹ thuật của CommonJs. Nó cung cấp một môi trường REPL cho kiểm thử tương tác.

Node.js là một ngôn ngữ mới, xây dựng thuần túy bằng javascript. Đây là một điểm lợi thế của Node.js để lập trình web-socket:

Thứ nhất: javascript là ngôn ngữ lập trình hướng sự kiện, mà trong lập trình thời gian thực, cách tiếp cận bằng lập trình sự kiện là cách tiếp cận khôn ngoan nhất.

Thứ hai: Node.js chạy non-blocking việc hệ thống không phải tạm ngừng để xử lý xong một request sẽ giúp cho server trả lời client gần như ngay tức thì.

Thứ ba: lập trình socket yêu cầu bạn phải xây dựng được mô hình lắng nghe – trả lời từ cả 2 bên. Nói khác đi, vai trò của client và server phải tương đương nhau, mà client thì chạy bằng javascript, nên nếu server cũng chạy bằng javascript nữa, thì việc lập trình sẽ dễ dàng và thân thiện hơn.

## Thư viện LIRC (Linux Infrared remote control)

Lirc là viết tắt của "Linux Infrared Remote Control". Đây là một gói phần mềm sẽ cho phép giải mã và lưu tín hiệu hồng ngoại để sử dụng sau này. Thư viện hỗ trợ cho các thiết bị chạy hệ điều hành Linux. Lirc cho phép thu các tín hiệu hồng ngoại và lưu chúng lại vào một file, đồng thời cho phép chúng ta phát lại các tín hiệu ấy qua module phát. Trong Lirc có 3 chức năng chính để phân tích các tín hiệu hồng ngoại:

**mode2 :** đầu ra pulse/space của tín hiệu hồng ngoại.

**irrecord :** dùng để ghi tín hiệu hồng ngoại vào file .conf dùng cho việc phát lại sau này.

**irsend**: gửi tín hiệu hồng ngoại từ dòng lệnh. Tín hiệu có thể được gửi một lần hoặc nhiều lần.

# CHƯƠNG 2: THIẾT KẾ SƠ ĐỒ KHỐI



Hình 1: Sơ đồ hoạt động của hệ thống

Trong sơ đồ hệ thống, điện thoại có chức năng là một Client thu tín hiệu giọng nói của người dùng để xác định kênh tivi người dùng muốn bật từ đó gửi lệnh lên cho Server Raspberry Pi thông qua giao thức Socket.

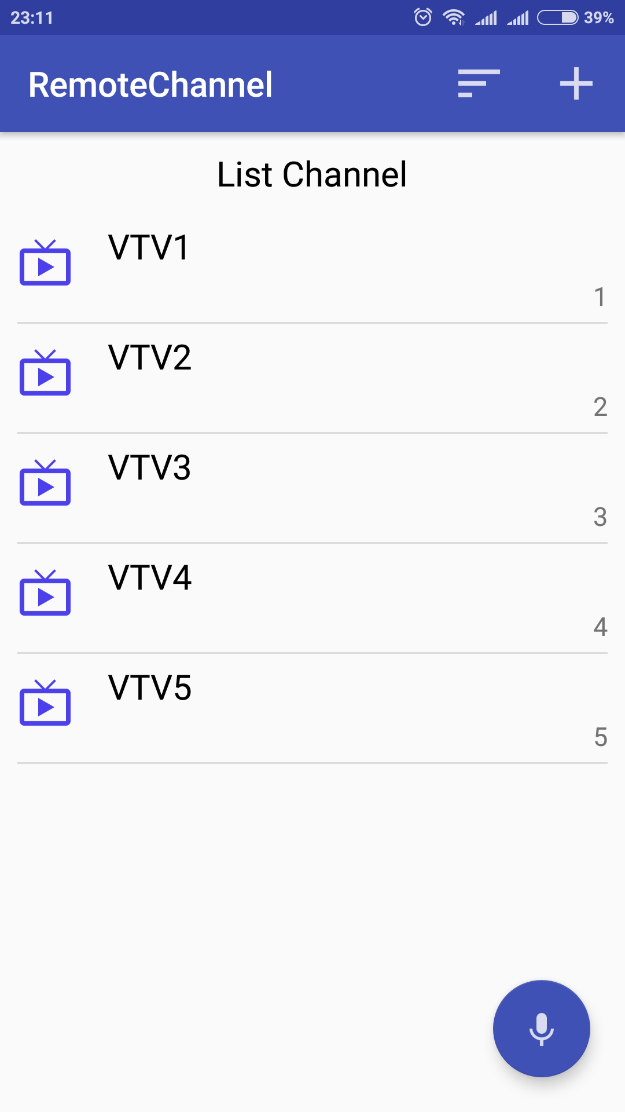
Raspberry đóng vai trò vừa là một Server vừa là nơi để điều khiển module thu phát hồng ngoại. Module hồng ngoại kết nối với Raspberry qua các chân GPIO. Raspberry sử dụng thư viện LIRC và điều khiển module hồng ngoại qua các lệnh trong thư viện quy định. Server trên Raspberry được viết bằng ngôn ngữ Nodejs, có chức năng lắng nghe kết nối từ ứng dụng, bắt gói tin từ ứng dụng gửi đến và thực hiện việc điều khiển thư viện LIRC.

Module thu hồng ngoại có chức năng thu tín hiệu từ remote tivi để thư viện LIRC lưu lại các tín hiệu đó phục vụ cho việc phát sau này. Module phát được điều khiển bởi thư viện LIRC sẽ phát ra đúng tín hiệu như ở remote tivi đã lưu trước đó.

# CHƯƠNG 3: THIẾT KẾ CÁC KHỐI

## Ứng dụng trên điện thoại

Ứng dụng được viết trên nền tảng hệ điều hành Android, có chức năng xử lý giọng nói đầu vào của người dùng và gửi lệnh điều khiển đến Server trên Raspberry Pi. Ngoài ra ứng dụng còn có chức năng cho phép người dùng thêm kênh mới hoặc chỉnh sửa kênh hiện tại, sắp xếp lại danh sách kênh theo tên kênh hoặc theo số kênh.



Hình 2: Giao diện chính của ứng dụng



Hình 3: Thành phần chức năng của ứng dụng

### Danh sách kênh

Danh sách kênh hiển thị các kênh do người dùng thêm vào. Một kênh có 2 thông số chính đó là:

* **Tên kênh:** là tên kênh truyền hình mà người dùng muốn gọi khi ra lệnh bằng giọng nói chẳng hạn: vtv1, hà nội 1, hà nội 2,…
* **Số kênh:** là số kênh tương ứng với tên kênh trên điều khiển tivi.

Để hiển thị được danh sách kênh, trong lập trình Android, ta sử dụng một **ListView** để hiển thị một danh sách do người dùng tự định nghĩa.

**ListView** là 1 **GroupView** sẽ giúp chúng ta hiển thị 1 dữ liệu lên màn hình dưới dạng 1 danh sách (có thể theo chiều đứng, hoặc nằm ngang).



Hình 4: Sơ đồ xây dựng một ListView

Để xây dựng 1 ListView ta cần có:

Thứ nhất: Về dữ liệu ta cần phải tổ chức dưới dạng 1 mảng dữ liệu: ta có thể sử dụng ArrayList<Object>, hoặc LinkedList, Enum, Cursor. Trong dự án này thì dữ liệu được sử dụng là ArrayList<Object>. Đối tượng Object ở đây chính là Kênh. Một kênh là một class riêng biệt với các thuộc tính là tên kênh, số kênh, và trạng thái đã chọn hay chưa chọn.

Sau đây là Object Kenh trong project:

public class Kenh {

public String nameChannel;

public String numberChannel;

private boolean Checked = false;

public Kenh(String nameChannel, String numberChannel) {

this.nameChannel = nameChannel;

this.numberChannel = numberChannel;

}

public boolean isChecked() {

return Checked;

}

public void setCheck(boolean check){

Checked = check;

}

}

**Thứ hai:** Về ListView ta sẽ khai báo 1 thẻ ListView trong MainActivity cần hiển thị và mapping nó tới Class xử lý.

listView = (ListView) findViewById(R.id.listView);

**Thứ ba:** itemViews là 1 file layout.xml trong resource layout là giao diện các item mà ta muốn hiển thị. Dữ liệu của từng đối tượng trong mảng dữ liệu sẽ được hiển thị trên item này tùy vào yêu cầu người xây dựng. Đây là dạng layout do người dùng tự thiết kế, gọi là custom listview.

**Thứ tư:** Adapter là 1 lớp khai báo các itemView và mapping dữ liệu từ các đối tượng trong mảng dữ liệu và hiển thị lên ListView trên activity. Đây cũng là thành phần quan trọng nhất khi xây dựng 1 Listview để đảm bảo được hiệu năng và trải nghiệm người dùng tốt nhất thì nó hoàn toàn phụ thuộc vào cách xây dựng Adapater.

public class KenhAdapter extends BaseAdapter {

Context context;

int layout;

List<Kenh> arrayChannel;

public KenhAdapter(Context context, int layout, List<Kenh> arrayChannel) {

this.context = context;

this.layout = layout;

this.arrayChannel = arrayChannel;

}

@Override

public int getCount() {

return arrayChannel.size();

}

@Override

public Object getItem(int position) {

return null;

}

@Override

public long getItemId(int position) {

return 0;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

convertView = inflater.inflate(layout, null);

//anh xa va gan gia tri

TextView nameChannel = (TextView)convertView.findViewById(R.id.tvTenKenh);

TextView numberChannel = (TextView)convertView.findViewById(R.id.tvSoKenh);

ImageView imageView = (ImageView)convertView.findViewById(R.id.imageView);

nameChannel.setText(arrayChannel.get(position).nameChannel);

numberChannel.setText(arrayChannel.get(position).numberChannel);

//Set icon cho kenh

if(arrayChannel.get(position).isChecked())

imageView.setImageResource(R.drawable.checked);

else

imageView.setImageResource(R.drawable.tivi);

return convertView;

}

}

Trong KenhAdapter các phương thức bắt buộc phải có:

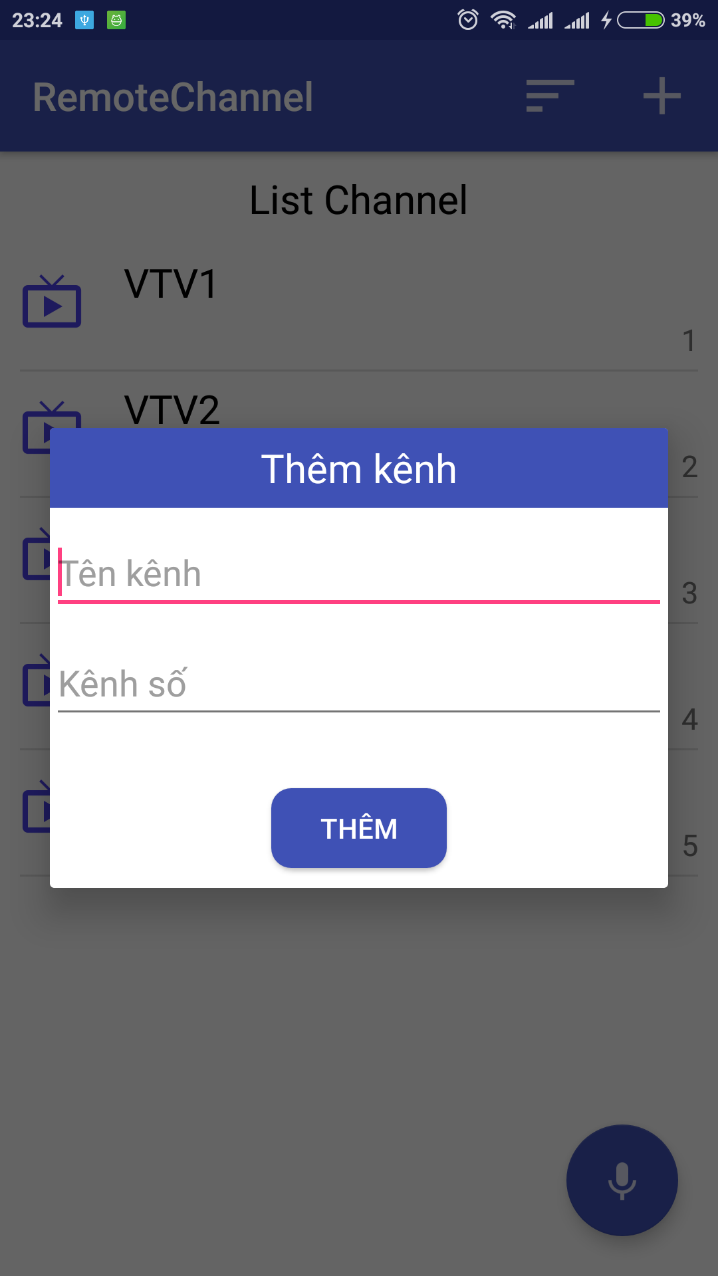
– **getView():**hỗ trợ Tạo ra các viewItem và mapping trực tiếp data lên các viewItem.

– **getCount():** trả về số lượng các viewItem dựa vào số size() của mảng data.

– **getItem(int position):** trả về Object dựa vào vị trí của đối tượng đó trong mảng dữ liệu.

– **getItemId(int position):** trả về id trên View của từng item mà adapter tạo ra.

### Thêm kênh, sửa kênh



Hình 5: Chức năng thêm kênh

Chức năng thêm kênh và sửa kênh cho phép người dùng thêm một kênh mới hoặc sửa kênh đã có trong danh sách. Chức năng sửa kênh sẽ hiện lên khi người dùng ấn vào một kênh bất kỳ. Trong project thì hai chức năng thêm kênh và sửa kênh là một hộp thoại Custom Dialog được thiết kế để người dùng có thể dễ dàng nhập dữ liệu.

**Dialog** có thể coi là một thông báo mà người dùng có thể tương tác trực tiếp được. Ví dụ khi muốn xóa một tập tin quan trọng hay muốn thoát một chương trình nào đấy thì việc hiển thì một thông báo để người dùng chắc chắn về hành vi của mình là rất quan trọng. Android hỗ trợ sẵn dialog cho lập trình viên nhưng đôi khi nó không phù hợp hoặc không đẹp, chúng ta có thể tự thiết kế dialog riêng theo ý mình mà không dùng đến layout có sẵn của Android.

Trong project, layout của hộp thoại dialog bao gồm 1 TextView là tên hộp thoại, 2 EditText cho phép người dùng nhập tên kênh và số kênh, 1 Button để cập nhật hoặc thêm kênh mới. Dưới đây là mã nguồn layout.xml cho hộp thoại Dialog:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical" android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="40dp"

android:text="Thêm kênh"

android:textSize="20dp"

android:textColor="@android:color/white"

android:background="@color/colorPrimary"

android:gravity="center"/>

<EditText

android:id="@+id/edtTenKenh"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Tên kênh"

android:layout\_marginTop="10dp"

android:inputType="textCapCharacters"/>

<EditText

android:id="@+id/edtSoKenh"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Kênh số"

android:layout\_marginTop="10dp"

android:inputType="number"/>

<Button

android:id="@+id/btnAddDialog"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:background="@drawable/botron"

android:text="Thêm"

android:textColor="@android:color/white"

android:layout\_gravity="center"

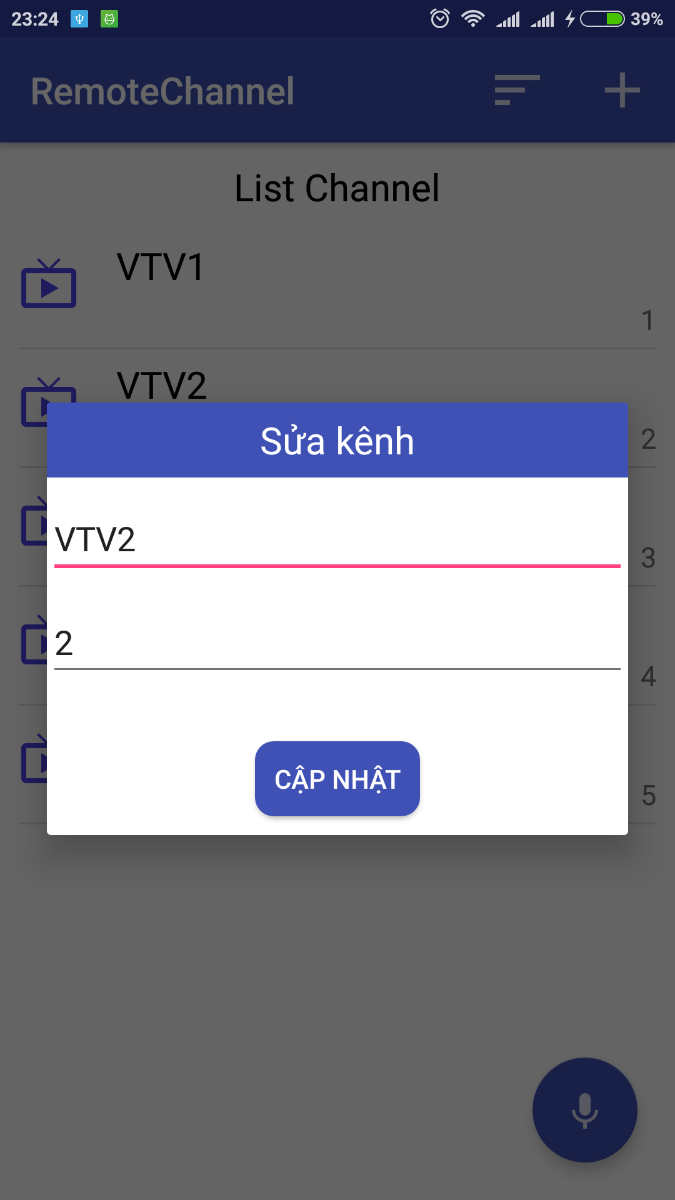
android:layout\_marginTop="30dp"/>

<android.support.v4.widget.Space

android:layout\_width="match\_parent"

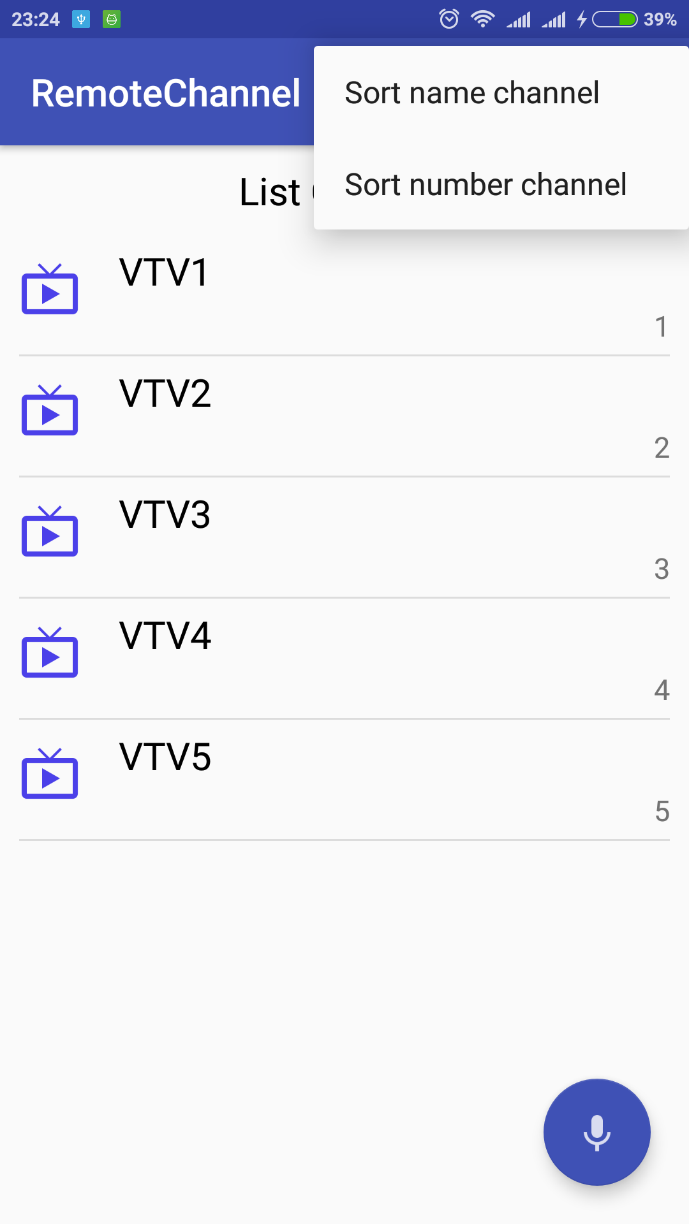
android:layout\_height="10dp" />

</LinearLayout>



Hình 6: Chức năng sửa kênh

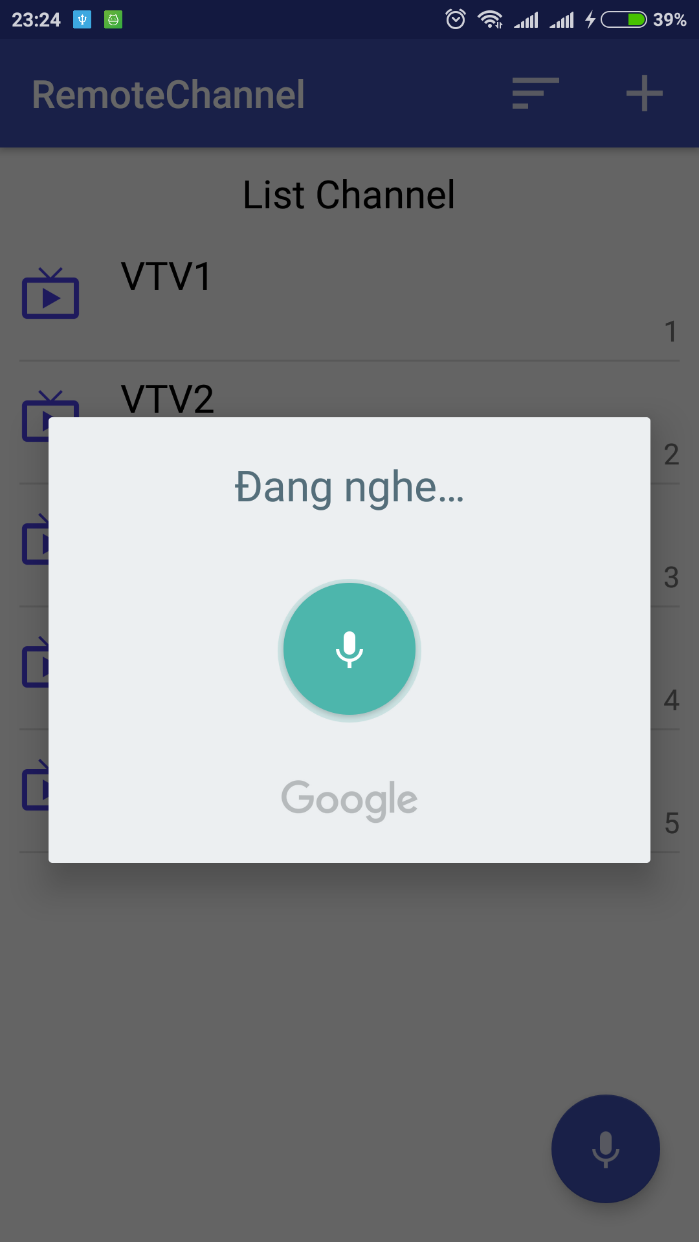
### Sắp xếp kênh



Hình 7: Chức năng sắp xếp kênh

Chức năng sắp xếp kênh được đặt cạnh chức năng thêm kênh trên thanh Toolbar. Chức năng sắp xếp kênh có nhiệm vụ sắp xếp lại danh sách kênh theo 2 tiêu chí: sắp xếp theo tên kênh và sắp xếp theo số kênh. Sắp xếp theo tên kênh là cách sắp xếp các tên kênh theo bảng chữ cái, sắp xếp theo số kênh sẽ sắp xếp kênh theo chiều tăng dần của số kênh.

### Ra lệnh điều khiển



Hình 8: Chức năng ra lệnh bằng giọng nói

Chức năng ra lệnh điều khiển bằng giọng nói được bật khi người dùng ấn vào Button voice góc dưới màn hình. Để thiết kế một button như vậy ta dùng một FloatingActionButton, đây là một thiết kế mới được Google đưa vào từ Android 5. Dưới đây là layout xml của FloatingActionButton:

<android.support.design.widget.FloatingActionButton

android:id="@+id/btnVoice"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_alignParentEnd="true"

android:layout\_marginBottom="20dp"

android:layout\_marginEnd="20dp"

android:onClick="VoiceClick"

android:src="@drawable/voice"

app:backgroundTint="@color/colorPrimary" />



Hình 9: Sơ đồ hoạt động của Button Voice

Cơ chế hoạt động của Button Voice là sẽ thu giọng nói đầu vào của người dùng (Voice Input), dữ liệu giọng nói sẽ được gửi lên Server của Google xử lý. Kết quả trả về là một chuỗi text dạng String của giọng nói đầu vào (Received Text). Chuỗi text nà được ứng dụng xử lý và gửi lệnh tới Server trên Raspberry Pi.

Để thu được giọng nói và gửi lên Server của Google thì ta dùng API RecognizerIntent. API này sẽ thu giọng nói của người dùng đồng thời gửi lên Server để xử lý và trả về một chuỗi String kết quả xử lý. Dưới đây là hàm xử lý giọng nói đầu vào:

private void promptSpeechInput() {

Intent intent = new Intent(RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL,

RecognizerIntent.LANGUAGE\_MODEL\_FREE\_FORM);

intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE, Locale.getDefault());

intent.putExtra(RecognizerIntent.EXTRA\_PROMPT, getString(R.string.speech\_prompt));

try {

startActivityForResult(intent, REQ\_CODE\_SPEECH\_INPUT);

} catch (ActivityNotFoundException a) {

Toast.makeText(getApplicationContext(),

getString(R.string.speech\_not\_supported),

Toast.LENGTH\_SHORT).show();

}

}

Để nhận được kết quả xử lý ta dùng phương thức bắt buộc là **onActivityResult:**

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

switch (requestCode) {

case REQ\_CODE\_SPEECH\_INPUT: {

if (resultCode == RESULT\_OK && null != data) {

processResult(data);

}

break;

}

}

}

Để nhận biết được người dùng yêu cầu kênh nào thì ta phải xử lý được chuỗi String nhận được. Ứng dụng sẽ yêu cầu người dùng nói một số mẫu câu đã được định sẵn bao gồm:

***Bật kênh + tên kênh*** hoặc ***kênh + tên kênh***

***Bật kênh + số kênh*** hoặc ***kênh + số kênh***

Đây là 4 mẫu câu mà người dùng để ra lệnh để bật kênh mong muốn. Phương pháp nhận dạng được kênh mà người dùng mong muốn bật ở đây là ta sẽ xét chuỗi string nhận được, tìm trong chuỗi từ khóa ***“kênh”***, khi tìm được vị trí của từ khóa ***“kênh”*** ta sẽ lấy phần chuỗi còn lại phía sau, phần chuỗi đó chính là kênh người dùng muốn bật. Dưới đây là hàm dùng để xử lý phần String:

private void processResult(Intent data) {

ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA\_RESULTS);

//Ket qua text tra ve

String textResult = result.get(0).trim().toUpperCase(); //chu hoa

//Tim index cuoi cua tu KENH trong chuoi ket qua

int lastIndex = textResult.lastIndexOf("KENH");

//Lay ten kenh hoac so kenh vi tri lastindex + 2

String resultChannel = textResult.substring(lastIndex + 2);

int resultSearch = SearchStringInArrayList(resultChannel, arrayChannel);

if(resultSearch != -1){

mSocket.emit("APPEMIT", arrayChannel.get(resultSearch).numberChannel);

Toast.makeText(this, "OK, turn on " + arrayChannel.get(resultSearch).nameChannel, Toast.LENGTH\_SHORT).show();

}

}

}

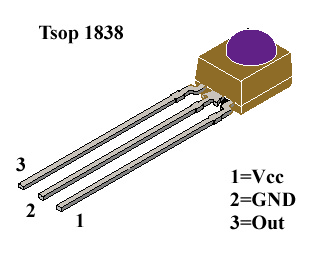
Khi nhận biết được kênh người dùng muốn bật, ta sẽ gửi số kênh tương ứng đến cho Server trên Raspberry xử lý. Để gửi được số kênh ta thông qua giao tiếp Socket bằng thư viện Socket.io. Câu lệnh để gửi số kênh đến Server:

mSocket.emit("APPEMIT", arrayChannel.get(resultSearch).numberChannel);

Ở đây **“APPEMIT”** là tên sự kiện mà ứng dụng gửi đến Server. Ở phía Server chỉ cần lắng nghe sự kiện **“APPEMIT”** là có thể bắt được dữ liệu ứng dụng gửi đến.

## Module thu phát hồng ngoại

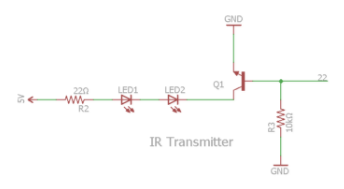
Để thu tín hiệu hồng ngoại từ remote, ta sử dụng mắt thu hồng ngoại TSOP1838 với tần số 38KHz, phù hợp với các loại điều khiển thông dụng.



Hình 10: Mắt thu hồng ngoại TSOP1838

Mắt thu hồng ngoại hoạt động ở mức điện áp 5V vì vậy chân Vcc của TSOP138 được nối với Vcc 5v của Raspberry Pi, chân ra tín hiệu số 3 sẽ được nối với **GPIO 23** của Raspberry Pi.

Để phát tín hiệu hồng ngoại từ Raspberry, ta sử dụng led phát hồng ngoại, để phát tín hiệu được xa hơn ta sử dụng thêm một transistor để khuếch đại tín hiệu. Dưới đây là sơ đồ mạch phát tín hiệu hồng ngoại:



Hình 11: Sơ đồ mạch phát tín hiệu hồng ngoại

**GPIO 22** của Raspberry sẽ là đầu ra của tín hiệu hồng ngoại, được nối vào cực B của transistor. Hai led hồng ngoại sử dụng là loại 5mm.

## Server trên Raspberry Pi

### Thư viện LIRC

Để cài đặt thư viện LIRC trên Raspberry ta chạy lệnh sau trên terminal:

sudo apt-get install lirc

Sau đó cần cài đặt chân GPIO nhận tín hiệu hồng ngoại và chân phát tín hiệu hồng ngoại. Ở đây chân thu tín hiệu là chân GPIO 23, chân phát tín hiệu hòng ngoại là chân GPIO 22. Những chỉnh sửa này được thực hiện trong file **/etc/modules**:

lirc\_rpi gpio\_in\_pin=23 gpio\_out\_pin=22

Tiếp theo cần phải chỉnh sửa thông số cài đặt hệ thống của Raspberry, mở file **/boot/config.txt** để chỉnh sửa, thêm vào file này dòng cuối cùng:

dtoverlay=lirc-rpi,gpio\_in\_pin=23,gpio\_out\_pin=22

Bước cuối cùng là chỉnh sửa lại file /etc/lirc/hardware.conf của thư viện LIRC như mẫu dưới:

# /etc/lirc/hardware.conf

#

# Arguments which will be used when launching lircd

LIRCD\_ARGS="--uinput"

# Don't start lircmd even if there seems to be a good config file

# START\_LIRCMD=false

# Don't start irexec, even if a good config file seems to exist.

# START\_IREXEC=false

# Try to load appropriate kernel modules

LOAD\_MODULES=true

# Run "lircd --driver=help" for a list of supported drivers.

DRIVER="default"

# usually /dev/lirc0 is the correct setting for systems using udev

DEVICE="/dev/lirc0"

MODULES="lirc\_rpi"

# Default configuration files for your hardware if any

LIRCD\_CONF=""

LIRCMD\_CONF=""

########################################################

Sau khi hoàn tất quá trình cài đặt thì khởi động lại Raspberry Pi.

Tiếp theo cần phải thu lại tín hiệu hồng ngoại từ Remote và lưu lại vào thư viện với tên phím tương ứng. Để Remote lại gần mắt thu và chạy lệnh sau:

sudo /etc/init.d/lirc stop

mode2 -d /dev/lirc0

Khi ấn một phím bất kì thì màn hình terminal xuất hiện các thông số của tín hiệu hồng ngoại dạng như sau:

space 14529891

pulse 8451

space 4734

pulse 274

space 884

pulse 403

space 532

pulse 635

space 649

pulse 440

space 680

pulse 462

Sau khi thư viện đã thu được tín hiệu hồng ngoại từ Remote thì ta tiến hành ghi lại các tín hiệu của từng nút trên remote và đặt tên cho chúng. Tên thường sẽ đặt theo chức năng của nó, chẳng hạn nút tăng âm lượng sẽ đặt là VOL\_UP,…Chạy lệnh sau để thực hiên:

sudo /etc/init.d/lirc/stop

irrecord -d /dev/lirc0 ~/lircd.conf

Quá trình trên kết thúc là ta đã ghi lại được các tín hiệu trên remote và lưu lại trong thư viện LIRC, cụ thể là trong file **~/lirc.conf.** Để phát một tín hiệu ra ngoài module phát, ta sử dụng lênh **irsend:**

irsend SEND\_ONCE REMOTE\_NAME CONTROL\_1

Ở đây REMOTE\_NAME là tên remote đặt ban đầu, CONTROL\_1 là phím cần phát đi mà đã đặt tên lúc thu tín hiệu. Khi lệnh trên thực hiện, module sẽ phát tín hiệu hồng ngoại giống với tín hiệu trên remote với phím tương ứng.

### Server Nodejs trên Raspberry Pi

Nhiệm vụ của Server Nodejs trên Raspberry Pi là bắt các lệnh mang tên **“APPEMIT”** từ ứng dụng trên điện thoại gửi tới và thực hiện lênh **irsend** với kênh mà ứng dụng gửi đến.



Hình 12: Cách hoạt động của Server Nodejs

Trong Nodejs hỗ trợ thư viện lirc\_node. Đây là thư viện hỗ trợ thực hiện các lệnh Irsend trong LIRC. Để thực hiện lệnh irsend bằng lirc\_node ta thực hiện:

lirc\_node = require('lirc\_node');

lirc\_node.init();

lirc\_node.irsend.send\_once(REMOTE\_NAME, KEY\_CONTROL, callback);

Trong đó, REMOTE\_NAME là tên remote đặt ban đầu, KEY\_CONTROL là phím người dùng muốn bật. KEY\_CONTROL được lấy từ bản tin của sự kiện APPEMIT do ứng dụng gửi đến.

Để bắt được sự kiện APPEMIT từ ứng dụng gửi đến ta dùng thư viện Socket.io. Lắng nghe sự kiện kết nối **connection** sau đó là sự kiện **APPEMIT**:

var socketio = require('socket.io')

var ip = require('ip');

var app = http.createServer();

var io = socketio(app);

io.on('connection', function(socket) {

console.log("App Connected");

socket.on('APPEMIT', function(data) {

});

});

Dưới đây là mã nguồn server hoàn chỉnh:

const PORT = 3484;

var socketio = require('socket.io')

var ip = require('ip');

var app = http.createServer();

var io = socketio(app);

lirc\_node = require('lirc\_node');

app.listen(process.env.PORT || PORT);

console.log("Server nodejs chay tai dia chi: " + ip.address() + ":" + PORT);

lirc\_node.init();

//Khi có mệt kết nối được tạo giữa Socket Client và Socket Server

io.on('connection', function(socket) {

console.log("App Connected");

socket.on('APPEMIT', function(data) {

console.log(data);

//socket.emit("SERVER", data);

lirc\_node.irsend.send\_once("tivi", data, function() {

console.log("Sent tivi" + data + "command!");

});

});

});

# KẾT LUẬN CHUNG

# TÀI LIỆU THAM KHẢO

# PHỤ LỤC