



Redes Móveis e Sem Fios



INSTITUTO SUPERIOR TÉCNICO

Projeto Intermédio - Sistema de controlo de música numa casa

Professor:
António Grilo

Elementos do Grupo:
Diana Fonseca Nº 86967
Gonçalo Mão-Cheia Nº 87007

Conteúdo

1	Introdução	2
2	Apresentação do projeto	3
3	Arquitetura do projeto	5
3.1	Raspberry Pi	5
3.2	Web Server	6
3.3	NodeMCU EXP8266	6
3.3.1	Implementação do EXP8266	6
3.3.2	Configuração do EXP8266	6
4	Mobile App	8
4.1	Front End	8
4.2	Back End	9
5	Estado atual do projeto	10

1 Introdução

Com o desenvolvimento de novas tecnologias nos últimos anos, o mundo encontra-se cada vez mais recetivo a novas ideias e projetos tecnológicos que permitem melhorar a qualidade de vida. Um dos temas mais falados dentro das áreas tecnológicas é a domótica. Domótica baseia-se na ideia da automatização das tarefas comuns numa casa, deste modo surge assim a ideia de Casas Inteligentes.

Com base nesta ideia de Casas Inteligentes e porque nos foi dada a possibilidade de criar um projeto **dentro das** IoT (Internet of Things) decidimos assim criar um sistema de controlo de música entre divisões de uma casa. Este sistema é desenvolvido com base num protótipo que permite o controlo de música entre duas divisões através de sensores de movimento, em que quem usufrui do espaço, tem controlo sobre o mesmo através de uma aplicação.

Neste relatório apresentamos a descrição do trabalho desenvolvido numa fase intermédia, que será posteriormente acabado e devidamente apresentado. Começamos assim por descrever o projeto, a arquitetura planeada neste ponto do trabalho e o estado atual em que o projeto se encontra.



2 Apresentação do projeto

O projeto apresentado consiste num sistema inteligente de controlo de música numa casa. Através de sensores de movimento, podemos perceber em que divisão da casa se encontra uma pessoa e, consequentemente ligar automaticamente música nessa divisão. O utilizador tem acesso ao controlo do sistema de música através de uma aplicação descrita de seguida neste relatório que lhe permite continuar/**pausar** a música, mudar de musica e/ou aumentar e diminuir o som. Tem acesso também a botões de controlo de volume da música nas divisões onde se encontra.

Para o desenvolvimento do sistema de música planeámos usar o seguinte material:

- 1 Raspberry Pi;
- 2 NodeMCU EXP8266;
- 2 sensores de movimento;
- 4 botões;
- 2 colunas de som;

Com base no movimento detetado pelos sensores de movimento, o sinal é recebido pelo NodeMCU, que comunica com o web server, e este, por sua vez, com um Raspberry Pi que controla a música na coluna de som presente nos espaço onde o movimento é detetado. Existem 2 **botões** ligados a cada NodeMCU de modo a que quem está na divisão a ouvir a música possa controlar o volume premindo os mesmos. Ao ser **primido** um botão, há uma instrução que é enviada para o web server através do NodeMCU e este faz atuar no Raspberry Pi o seu efeito. Temos assim 2 divisões, cada uma com um EXP8266 onde se encontram ligados um botão de aumento de volume, um botão de diminuição de volume e um sensor de movimento.

O utilizador tem acesso à aplicação que foi desenvolvida em Android Studio e mediante as suas ações na mesma, há a comunicação das instruções ao web server, instruções essas passadas ao Raspberry Pi como descrito no caso do uso dos botões físicos.

Os atuadores do projeto, neste caso as colunas de som, encontram-se assim ligadas a um Raspberry Pi.

A transmissão de informação entre o Raspberry Pi, os NodeMCU e a aplicação móvel será realizada pelo web server Firebase.

Desta forma, o Raspberry Pi recebe instruções da Firebase e atua de acordo com as mesma, enquanto que a **mobile app** e os NodeMCU enviam instuções para a Firebase.

Na **figura 1** podemos ver um esquema representativo da descrição da arquitetura do projeto em desenvolvimento. De notar que na imagem apresentada, as ligações físicas entre os elementos encontram-se representadas por linhas pretas. As linhas laranja tracejadas correspondem a uma comunicação wireless entre os elementos.



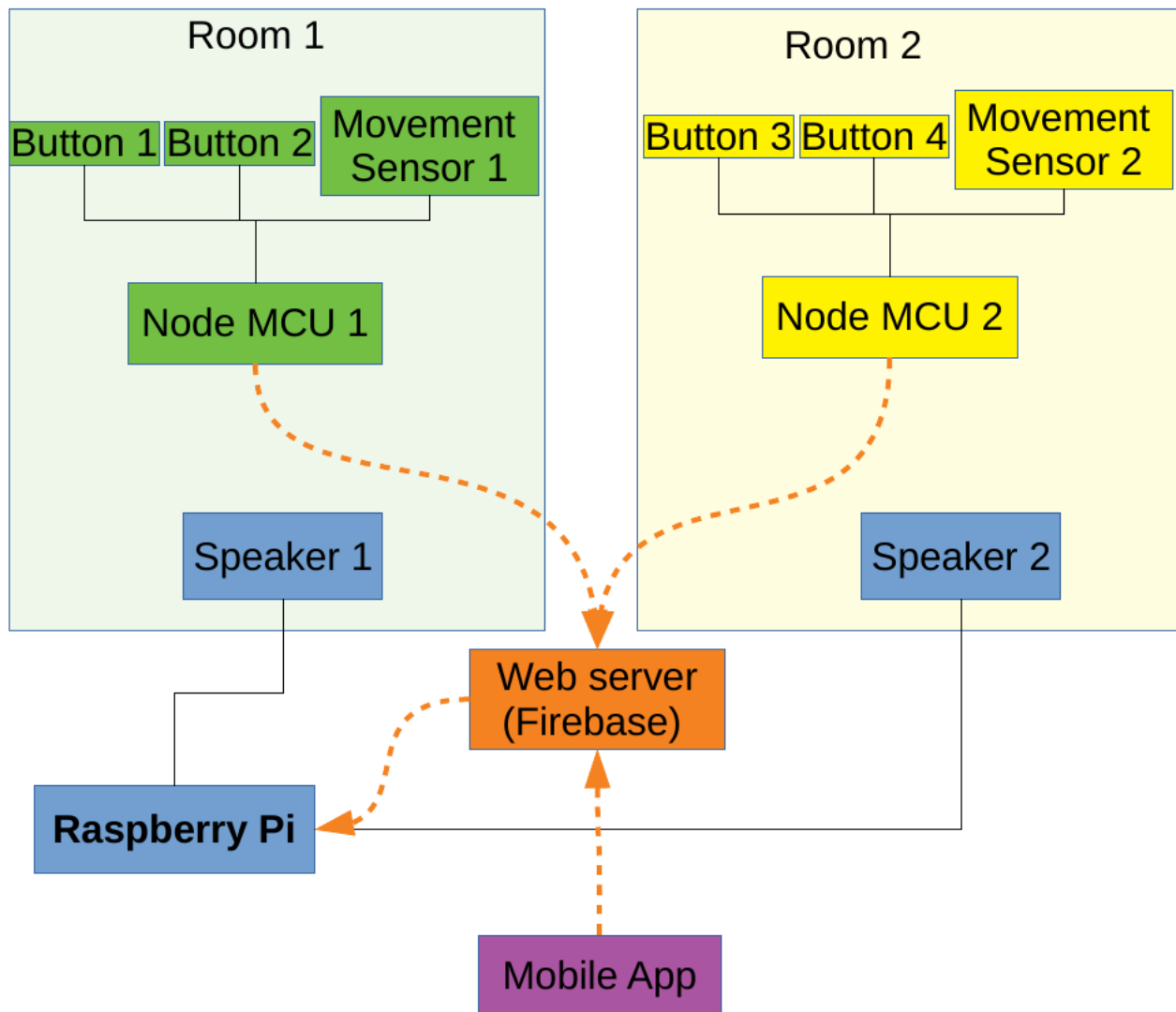


Figura 1: Diagrama com a arquitetura geral do projeto

3 Arquitetura do projeto

3.1 Raspberry Pi

Em termos de conexões físicas, o Raspberry Pi está apenas conectado às colunas de som através de um cabo de som de 3.5 mm, como representado na figura 2.

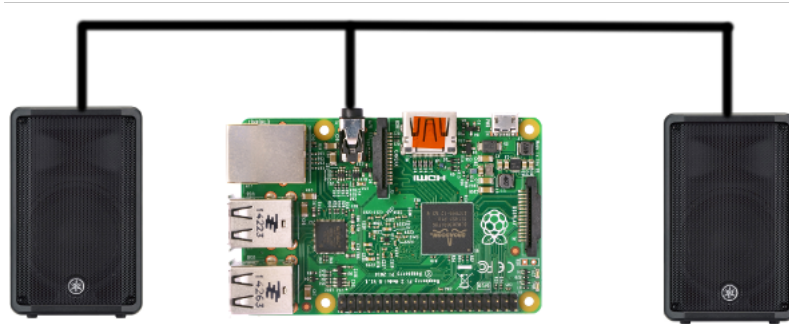


Figura 2: Ligação do Raspberry Pi com as colunas de som

Relativamente à implementação de código no Raspberry Pi, corre um script em Python. Neste script é utilizada a biblioteca `firebase`, de forma a permitir fazer a leitura e de seguida apagar mensagens do web server.

Primeiramente é feita a conexão ao `web server`. Depois é necessário abrir uma sessão do programa Tizonia, que nos permite ter acesso ao Spotify através do terminal. No entanto, esta sessão do programa Tizonia necessita de permanecer aberta, para mais tarde ser possível enviar comandos de forma a alterar a música e o volume. Devido a isto esta sessão do programa Tizonia é aberta com o comando `tmux`.

O comando `tmux` é um multiplexer de terminais, que permite fazer `detach` de um processo e abrir o mesmo mais tarde. É através deste comando e da sua funcionalidade `send-keys`, que conseguimos comunicar com o programa Tizonia enquanto corremos a script de Python que lê e processa continuamente as mensagens contidas no web server.

As intruções que serão interpretadas pelo Raspberry Pi são as seguintes:

- **PLAY**: `pausa` ou continua a música
- **UP**: aumenta o volume
- **DOWN**: diminui o volume
- **NEXT**: muda para a música seguinte
- **PREV**: muda para a música anterior
- **ONOF**: liga ou desliga o programa Tizonia
- **DIV1**: redireciona o som para a divisão 1
- **DIV2**: redireciona o som para a divisão 2

3.2 Web Server

O Web Server foi implementado através da plataforma Firebase da Google. É assim utilizada uma Firebase Realtime Database, que permite estabelecer a conexão entre a mobile app, os NodeMCUs e o Raspberry Pi. Através deste Web Server conseguimos assim transmitir mensagens em tempo real de forma a oferecer ao utilizador uma experiência natural e **fluída**.

3.3 NodeMCU EXP8266

3.3.1 Implementação do EXP8266

Na figura 3 podemos ver o esquema de ligação dos botões de controlo de volume e o sensor de movimento. Os dois botões usados são digitais e ao serem pressionados, o NodeMCU deteta e envia para a base de dados a instrução UP ou DOWN, consoante o botão pressionado. O sensor ao detetar movimento, envia informação de movimento para o NodeMCU que envia para a Firebase a instrução DIV1 ou DIV2 (mediante a divisão).

A comunicação do NodeMCU com a firebase é possível após a configuração do mesmo para se conectar à rede wifi da casa em que está e que se ligue a esta após lhe ser atribuído um IP. Fica assim apto a comunicar com o web server que também ele é configurado no código do mesmo.

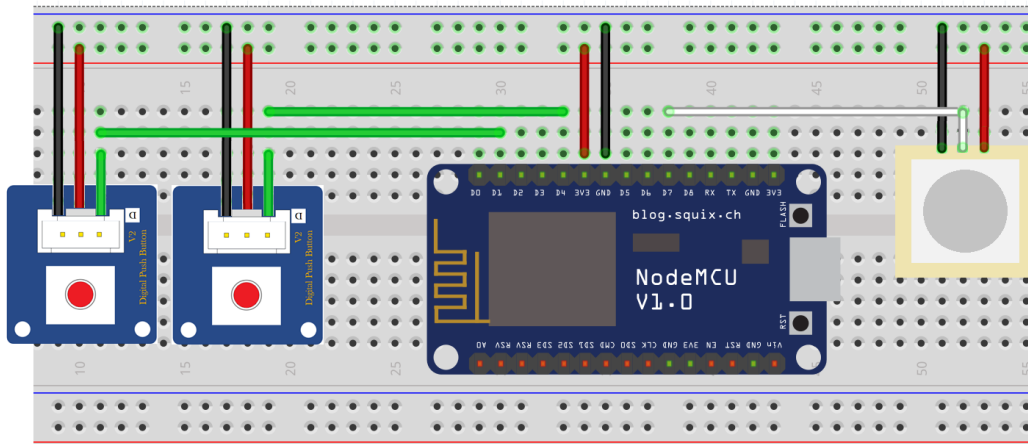


Figura 3: Ligação do Raspberry Pi com as colunas de som

3.3.2 Configuração do EXP8266

Para a configuração do NodeMCU definimos a firebase a que o nosso equipamento deve comunicar as instruções e as informações da rede wifi a que é para se ligar. De seguida definimos os locais das conexões dos botões e do sensor através dos números associados a cada pin do Raspberry Pi e definimos os dois botões e o sensor de movimento como INPUT e o D0 como OUTPUT de modo a fazer se a conexão com a Internet e de seguida a sessão com a Firebase.

Seguidamente foram configurados cada botão para, quando pressionados, enviarem a instrução UP ou DOWN, consoante o botão pressionado, de modo a aumentar ou diminuir o

volume a música. Quando o sensor de movimento detecta algo, envia uma instrução para a Firebase de modo a começar a música, sendo que temos um sensor na divisão 1 que envia DIV1 para poder começar a tocar música na divisão 1 e caso seja o sensor da divisão 2 a ser acionado, o seu respectivo NodeMCU envia a instrução DIV2 para o server poder comunicar com o Raspberry Pi e este ligar a música na coluna da divisão 2.

4 Mobile App

4.1 Front End

A aplicação de controlo do sistema pelo utilizador foi desenvolvida em Andoid Studio. A sua interface pode ser observada na figura 4.



Figura 4: Interface da Mobile App

A interface é simples, com 6 botões que permitem realizar algumas funções básicas relativamente à reprodução de música. As mensagens enviadas por cada botão são:

- PAUSE/PLAY: PLAY
- VOL +: UP
- VOL -: DOWN
- NEXT: NEXT
- PREVIOUS: PREV
- ON/OFF: ONOF

4.2 Back End

Primeiramente, quando é ligada, é estabelecida a conexão da App com o web server Firebase. Seguidamente, esta aguarda que um botão seja premido e envia uma instrução para o Firebase, de acordo com o botão premido.

5 Estado atual do projeto

Depois de apresentada e analisada a arquitetura do projeto, podemos fazer um balanço do que está já implementado e do que falta implementar/melhorar. Neste sentido, analisamos um pouco o trabalho já desenvolvido e o que ainda se encontra em desenvolvimento .

De momento, o web server Firebase encontra-se totalmente operacional.

A mobile app consegue fazer upload de mensagens para o mesmo e o Raspberry Pi consegue ler e processar estas mensagens.

A mobile app também se encontra totalmente operacional, apesar de não descartarmos a possibilidade de alterações futuras de forma a melhorar o nosso projeto.

Existe, no entanto, um obstáculo relativamente à alteração do balanço de som no Raspberry Pi, dado que este está configurado a mono e não encontramos forma de alterar esta configuração para stereo. A solução atual para este problema será adicionar um sound card de maneira a poder adicionar esta feature, visto que, é através da modificação de balanço do som que planeamos trocar a divisão na qual está a ser reproduzida música.

Em relação à implementação do circuito com os NodeMCU já é feita a comunicação com a Firebase das intruções de aumento e redução do volume. O NodeMCU envia instruções de UP e DOWN para a Firebase. As instruções de DIV1 e DIV2 ainda não se encontram implementadas e a lógica do acionamento do sensor de movimento também ainda não se encontra implementada. Foi desenvolvido este sistema com um dos 2 NodeMCU que temos de modo a duplicar tudo para o segundo NodeMCU.

Depois de concluídos os passos de faltam, iremos testar todas as funcionalidades do projeto em simultâneo. Iremos também estudar a implementação de novas funcionalidades e melhoramento das funcionalidades atuais até ao final do projeto, pelo que este relatório descreve o que está desenvolvido e pensado no momento e um pouco do que ainda vai ser feito e que pode ser alterado.

