

# Assistente Pessoal Controlado por Voz

Ponto de Controle 4

Disciplina de Sistemas Embarcados

23 de novembro de 2017

Ingrid Miranda de Sousa

14/0143980

Engenharia Eletrônica

Universidade de Brasília

Brasília - DF

Email: ingridmsousa@hotmail.com

João V. A. Guimarães

12/0122045

Engenharia Eletrônica

Universidade de Brasília

Brasília - DF

Email: joaoguimaraes31@gmail.com

**Resumo**—Esse documento expõe uma proposta detalhada de projeto de um assistente pessoal capaz de cumprir diversas instruções dadas por comandos de voz afim de auxiliar o usuário em tarefas simples do dia-a-dia como checar emails, definir alarmes, pequenas pesquisas na web e também controles básicos de automação residencial como ligar e desligar luzes, abrir portas entre outras funcionalidades.

## A. Projeto de Hardware

O projeto geral de hardware está descrito na Figura 1. O mesmo foi feito levando em consideração os requisitos funcionais citados no relatório do Ponto de Controle 1 [1]. Os principais componentes do diagrama abaixo serão tratados individualmente nas seções a seguir.

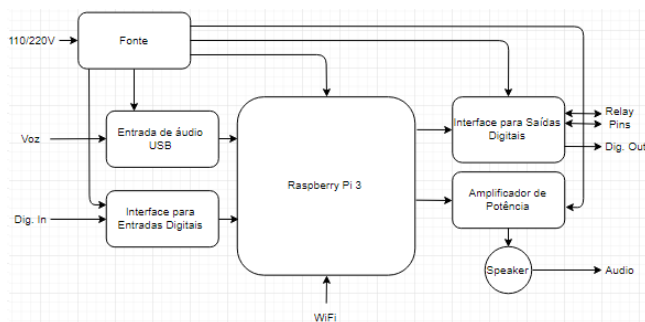


Figura 1. Projeto Geral de Hardware

## B. Interface para Saídas e Entradas Digitais

1) *Saídas Digitais:* A *Raspberry Pi3 Model B* possui 26 GPIOs (*General Purpose Input/Output*)

e as mesmas trabalham com nível lógico alto de 3V3 e nível lógico baixo de 0V. Isso de certa forma limita bastante as aplicações em que essas portas podem ser usadas, a maioria dos microcontroladores da família ATmega trabalham com nível lógico alto de 5V por exemplo, a indústria automotiva trabalha com nível lógico alto de 12V na maioria dos circuitos (exclui-se redes CAN). Em contrapartida é relativamente fácil modificar esse nível de tensão em questão visto que o mesmo se trata de um sinal digital. Uma solução relativamente simples consiste em utilizar as saídas digitais de 3V3 da *RaspberryPi* apenas para controlar circuitos que iram na prática acionar dispositivos externos em vez de acioná-los diretamente com a *RaspberryPi*. A Interface para saídas digitais pode ser dividida em dois circuitos replicáveis sendo o primeiro mostrado pela Figura 2 a seguir.

Nesse circuito a GPIO polariza um *MOSFET* - (*Metal-Oxide-Semiconductor Field-Effect Transistor*) que pode ser conectado a diferentes fontes de tensão. Quando a GPIO envia um nível lógico alto o *MOSFET* passa a conduzir a tensão escolhida (através do uso de um jumper) para a saída digital do dispositivo. Os resistores R1 e R2 são resistores de *pull-down* (ambos de 10kΩ e servem para garantir que quando a GPIO não estiver enviando nível lógico alto o *MOSFET* não estará polarizado e a saída do circuito será de nível lógico baixo (0V). Essa opção de poder escolher a tensão de

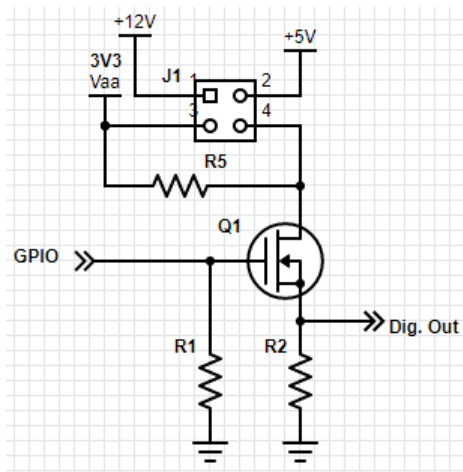


Figura 2. Circuito para Interface de Saídas Digitais

saída é muito útil pois assim a *RaspberryPi* pode interfacear com diversos dispositivos, o resistor R5 é um resistor de *pull-up* e é usado para garantir que quando nenhum jumper está conectado a tensão em nível lógico alto da saída digital será de 3V3.

2) *Controle Relê*: O circuito da interface de saídas digitais para controle do relê está apresentado na Figura 3 a seguir.

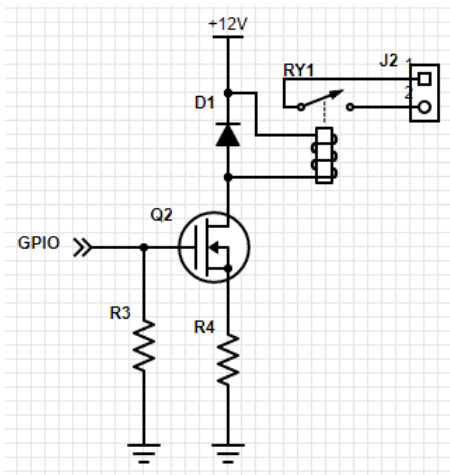


Figura 3. Circuito para Interface de Controle de relê

Sobre o que envolve o MOSFET esse circuito funciona de forma exatamente igual ao anterior, a diferença é que dessa vez em vez enviar um sinal digital esse circuito irá chavear um relê que pode controlar uma variedade maior de dispositivos, incluindo o uso de redes de corrente alternada, esmagadora maioria em redes residenciais. O diodo

serve para garantir que nenhuma corrente flua em sentido oposto ao ideal. Foi escolhido um MOSFET pois o mesmo não possui conexão física entre o terminal que está conectado na *RaspberryPi* e os demais e assim garante que apenas o resistor de *pull-down* irá drenar corrente (uma corrente mínima pois o mesmo deverá ter valor altíssimo).

3) *Interface para Entradas Digitais*: Para que o dispositivo possa trabalhar com diferentes tensões na entrada digital será usado o circuito na Figura 4.

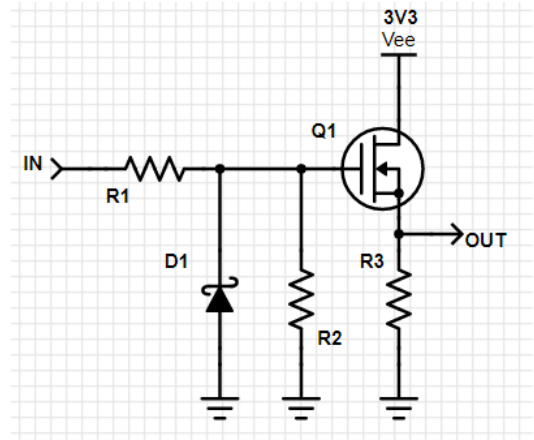


Figura 4. Circuito para Interface de Entradas Digitais

O circuito converte uma tensão de qualquer valor maior que 3V3 em 3V3 em dois estágios. No primeiro estágio essa tensão é regulada para 3V3 usando o diodo zener (modelo 1N4728A) e depois polariza o mosfet que coloca 3V3 na saída. O resistor R1 serve para regular a corrente que passa pelo zener, a corrente máxima desse zener é de 276mA logo com o resistor R1 em 1kΩ o circuito pode trabalhar praticamente com qualquer tensão menor que 200V e maior que 3V3. O resistor R3 é um resistor de *pull-down* e tem valor de 100kΩ, é usado para garantir que a tensão de entrada no raspberry pi seja de 0V quando a tensão de entrada do circuito seja menor que 3V3. O Resistor R2 acabou não sendo montado pois foi julgado como desnecessário.

4) *Circuitos montados*: Os três circuitos para interfaces digitais foram montados na mesma placa perfurada como mostra a Figura 5. Em todos os circuitos o mosfet escolhido foi o BS170.

Os circuitos são perfeitamente escaláveis e por isso foi escolhido montar apenas um de cada para demonstrar o funcionamento.

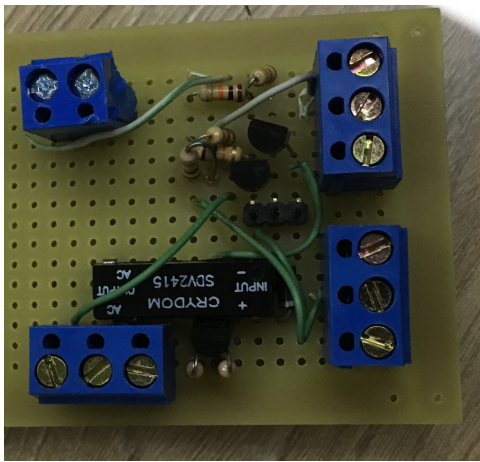


Figura 5. Circuito para Interfaces Digitais

### C. Amplificador de Potência

A saída de áudio do *RaspberryPi* é apenas uma saída de sinal e para atender os requisitos do dispositivo proposto será necessário um amplificador de potência. O circuito da Figura 6 foi tirado da internet [2] e é um amplificador com tensão de entrada de 12V e saída de 6W para um falante de 8Ohms (comum no mercado), esse circuito poderá ser modificado pois a potência do amplificador pode ser suficiente ou não para a aplicação.

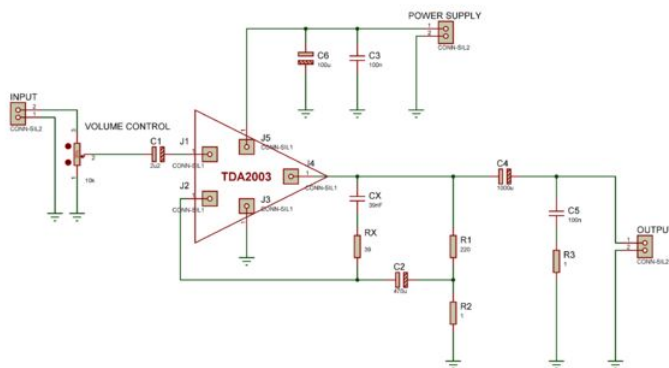


Figura 6. Circuito para Amplificador de Potência

O circuito foi montado fielmente ao esquemático como mostra a Figura 7.

### D. Fonte de Alimentação

Para o projeto será necessário ter fontes de tensão de três valores: 12V, 5V e 3V3. Como o foco desse projeto não é eletrônica de potência, será comprada uma fonte de 12V/DC e a tensão da mesma será



Figura 7. Circuito Montado para o Amplificador de Potência

convertida para os dois outros valores propostos: 5V e 3V3. Como a fonte de 5V é fundamental para o projeto será usado um regulador de tensão não-linear chaveado da *Texas Instruments*, o componente escolhido é o LM2596 [3]. O mesmo pode fornecer até 3A na saída e possui eficiência de 80%, o que elimina a necessidade de um dissipador grande de calor. Para a fonte de 3V3 será usado um regulador de tensão mais simples, o LM317 [4], apesar de ser um regulador mais simples a fonte de 3V3 será a menos demandada e por isso um regulador linear poderá ser usado. A Figura 8 mostra o projeto das fontes de alimentação.

A escolha dos componentes adjacentes aos reguladores foi feita de acordo com as recomendações dos datasheets.

O circuito foi montado fielmente ao esquemático como mostra a Figura 9.

### E. Projeto de Software

Em relação ao componente de software do projeto é necessário essencialmente reconhecer os comandos de voz feitos pelo usuário automaticamente, sem auxílio de palavras-chave ou de botões para acionar o sistema, recursos que costumam ser utilizados em outros projetos. Então o início do programa acontece ao identificar sinais no microfone em um

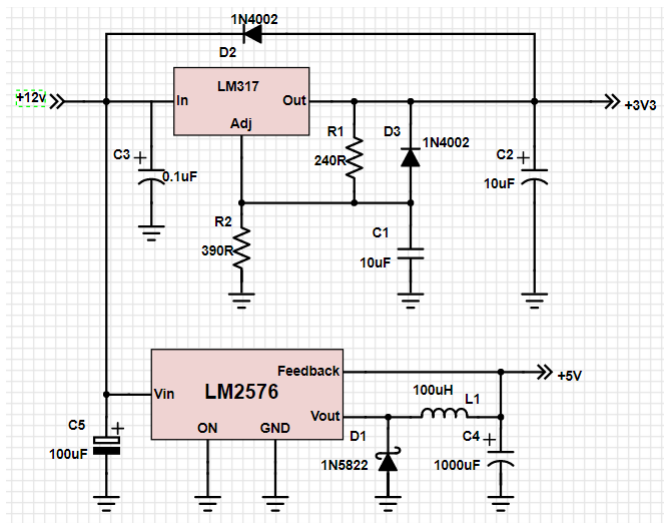


Figura 8. Fonte de Alimentação

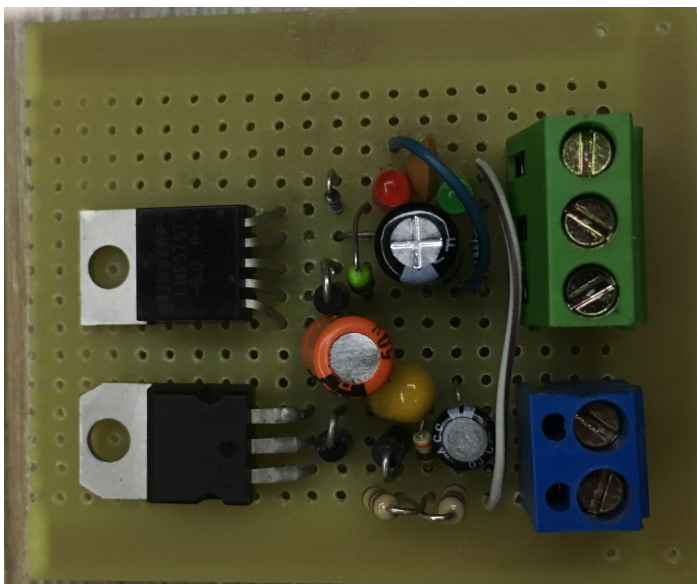


Figura 9. Circuito Montado para Fonte de Alimentação

volume mínimo pré-determinado. A partir desse instante o som começa a ser gravado, sendo que permanece desse modo durante alguns segundos. E após isso, o sinal de voz recebido será convertido para texto. Onde por fim, será analisado se alguma das palavras recebidas têm correspondência com a lista de comandos pré-determinados. Caso tenham, será realizada a função equivalente ao comando. E caso contrário, indicará que não houve palavra correspondente e voltará para o início do programa. O algoritmo correspondente à essa parte está apresentado na Figura 10 a seguir. E utiliza-se do pro-

grama Julius para a decodificação e reconhecimento de voz.

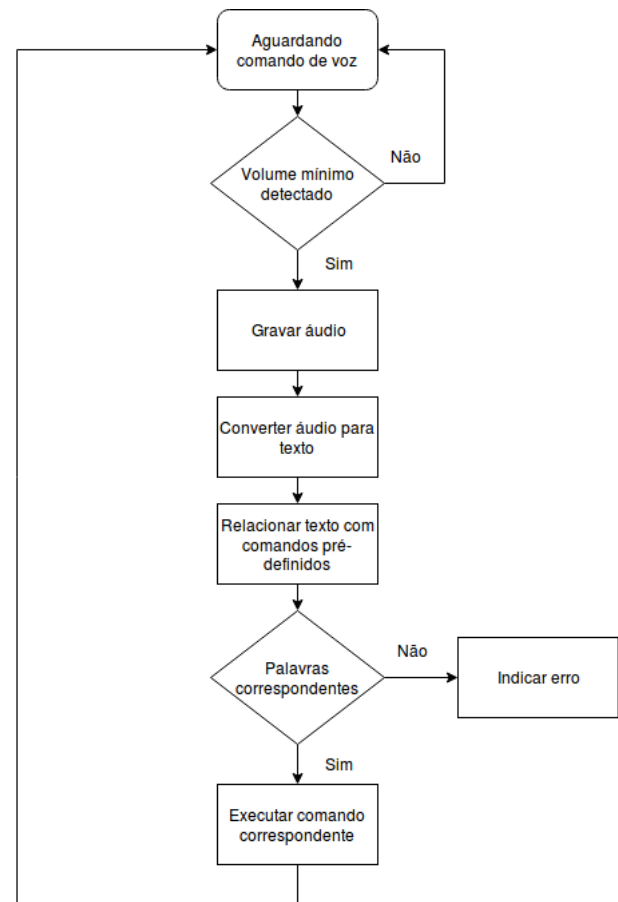


Figura 10. Fluxograma do algoritmo

## F. Configurações necessárias

Para o código funcionar da maneira correta foi necessário adaptar algumas configurações da raspberry pi e dos dispositivos tais como a webcam utilizada como microfone. Então, para o reconhecimento de voz instalou-se a biblioteca libasound2-dev, o programa Julius e o pacote Quickstart-Linux. E em relação ao microfone embutido na webcam Logitech c270 foi necessário checar por meio de diversos comandos se estava sendo reconhecido pela raspberry e funcionando da maneira correta. O primeiro consistiu em "lsusb", para checar se o dispositivo aparecia na lista. Logo após, utilizar o comando "alsamixer" a fim de aumentar o volume do microfone. Depois o "arecord -l" para testar se o microfone estava realmente sendo reconhecido. E por fim, o "arecord -D plughw:1,0 test.wav" para



gravar o som no arquivo test.wav, caso ao usar o comando "aplay test.wav"ouvir-se a gravação, o microfone está funcionando corretamente.

### G. Programa Julius

O Julius consiste em um programa com código aberto que é utilizado como decodificador para reconhecimento de fala. Foi desenvolvido sob os sistemas operacionais Windows e Linux. Possui amplo vocabulário, alto desempenho e versatilidade. Sendo possível executar o reconhecimento de voz tanto em tempo real com dispositivos de entrada, quanto por meio de arquivos de áudio previamente armazenados nos formatos WAV. Além disso, ele pode ser usado em conjunto com o software Coruja para adicionar a opção de reconhecimento de voz em português, tornando o programa mais acessível para os usuários.

### H. Controle de reprodutor de música por comando de voz

O programa ao ser executado procura pela entrada de áudio utilizada e o arquivo de configuração com extensão .jconf, local no qual são apontados quais devem ser os modelos acústicos e de linguagem a serem utilizados e o dicionário fonético que também é carregado. Se o carregamento de tais arquivos ocorrer sem erros, o programa Julius entra um laço de repetição, onde cada ciclo termina com o fim da entrada de áudio do usuário. Por fim, para cada comando dado pelo indivíduo, o programa reconhece se alguma das palavras ditas está entre as opções de palavras relacionadas aos comandos e realiza as ações associadas. Seu funcionamento pode ser observado na Figura 11, a qual representa as palavras que o programa foi capaz de reconhecer.

### I. Campanha que tira e envia foto ao ser acionada

Este programa consiste em uma campanha programada para tirar uma foto ao ser pressionada e em seguida enviar para um email predeterminado um aviso com tal foto anexada. Tal dispositivo pode auxiliar na segurança residencial e simplificar no cotidiano do usuário. O código correspondente ao seu funcionamento foi feito seguindo a lógica presente na Figura 12. De modo que para testar o seu funcionamento utilizou-se um botão push button como acionador do sistema e para que a foto fosse

```

pi@raspberrypi: ~/reproductor
File Edit Tabs Help
pi@raspberrypi:~$ cd reproductor/
pi@raspberrypi:~/reproductor$ export ALSADEV="plughw:1,0"
pi@raspberrypi:~/reproductor$ julius -C julian.jconf | ./command.py
Taking control of Rhythmbox media player.
Recognized input: Computer play
Recognized input: Computer show
<<< please speak >>>
(rhythmbox-client:1804): Rhythmbox-WARNING **: Unknown option --notify
Recognized input: Computer show
<<< please speak >>>
(rhythmbox-client:1806): Rhythmbox-WARNING **: Unknown option --notify
Recognized input: Computer next
<<< please speak >>>
(rhythmbox-client:1808): Rhythmbox-WARNING **: GDBus.Error:org.gtk.GDBus.Unmappe
dError.Quark._rb_5fshell_5fplayer_5ferror.Code1: No next song
Recognized input: Computer next
Recognized input: Computer silence
Recognized input: Computer silence
Recognized input: Computer silence
Recognized input: Computer prev
<<< please speak >>>
(rhythmbox-client:1809): Rhythmbox-WARNING **: GDBus.Error:org.gtk.GDBus.Unmappe
dError.Quark._rb_5fshell_5fplayer_5ferror.Code2: Not currently playing
Recognized input: Computer silence
Recognized input: Computer next
Recognized input: Computer show
<<< please speak >>>
(rhythmbox-client:1940): Rhythmbox-WARNING **: Unknown option --notify
Recognized input: Computer prev
(rhythmbox-client:1942): Rhythmbox-WARNING **: GDBus.Error:org.gtk.GDBus.Unmappe
dError.Quark._rb_5fshell_5fplayer_5ferror.Code1: No previous song
Recognized input: Computer pause
Recognized input: Computer silence
Recognized input: Computer pause
<<< please speak >>>C

```

Figura 11. Reconhecimento de voz feito pelo programa

tirada usou-se a mesma webcam Logitech c270. Para o envio do email escolheu-se como servidor o gmail e definiu-se a porta 465. O email enviado pelo programa pode ser visto na Figura 13, onde é possível observar que é enviada uma mensagem de alerta em conjunto com uma foto.

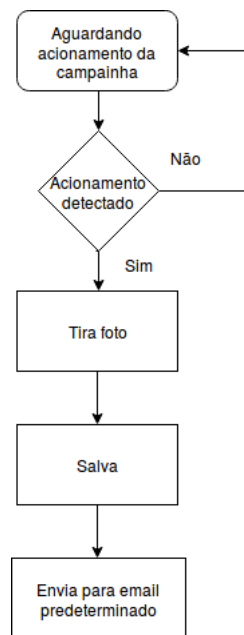
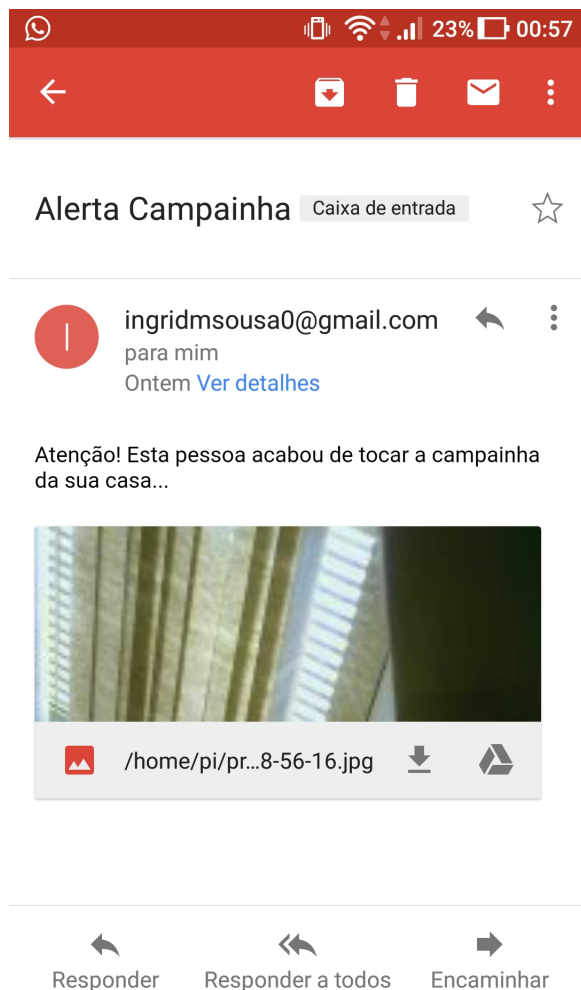


Figura 12. Fluxograma do algoritmo

### J. Resultados obtidos

Com o código realizado até o momento, foi possível executar o programa "Rhythmbox"que é



- [3] *LM2596 Simple Switcher Power Converter*, Texas Instruments, May 2016.
- [4] *LM317 3-Terminal Adjustable Regulator*, Texas Instruments, Sep. 2016.

Figura 13. Email enviado pelo programa

um player de música totalmente por meio de comandos de voz. Dentre as ações estabelecidas estão a execução do programa, tocar uma música, pausar, passar para a próxima ou retornar para a anterior. De modo que para serem realizados estão entre as palavras de comando: play, next, previous, stop e show. Além disso, alcançou-se o funcionamento correto do sistema de campanha. De modo que ao ser pressionada está acionando a webcam para tirar foto, logo após salva e envia como anexo para o email do usuário, tal como planejado.

## REFERÊNCIAS

- [1] J. Guimarães and I. Sousa, "Assistente pessoal controlado por voz - ponto de controle 1," 2017.
- [2] Unknown, "Gpio: Raspberry pi a and b," 2016. [Online]. Available: <http://www.instructables.com/id/10W-RMS-Audio-amplifier/>