

# Acionamento de comandos do carro via controle de VOZ

## *1º ponto de controle*

Matheus Jericó Palhares  
Universidade de Brasília - UnB  
Faculdade Gama - FGA  
Brasília, Brasil.  
matheusjerico1994@hotmail.com

Lucas Raposo Souza Carvalho  
Universidade de Brasília - UnB  
Faculdade Gama - FGA  
Brasília, Brasil.  
lucas.raposo1995@gmail.com

**Resumo:** Dispositivo controlado por voz capaz de acionar algum comando, tendo como objetivo atuar no carro.

**Palavras chaves:** *RaspBerry Pi, controle de voz, controle e automação de carro.*

### I. JUSTIFICATIVA

Com o aumento populacional, aumentou-se a quantidade de carros principalmente em grandes centros urbanos, causando grandes congestionamentos, estresse durante o caminho devido à quantidade de carros na pista, entre outros. Nessa linha de raciocínio, nota-se que é necessário cada vez mais prestar no trânsito enquanto dirige e muitas vezes não é possível realizar tal tarefa com excelência, visto que o motorista pode ser distraído devido à necessidade de ligar o farol do carro, aumentar o volume do som ou até escolher uma música infantil para acalmar a criança no banco de trás. Nesses pequenos momentos de distrações podem ocorrer acidentes como batidas e atropelamentos.

Um dispositivo que é controlado por voz (Jasper) e atua em vários sistemas do carro, resolveria grande parte dos problemas e ainda traz um conforto associado ao luxo extra ao motorista e aos passageiros do automóvel.

### II. OBJETIVOS

Criar um dispositivo capaz de receber comando de voz de qualquer pessoa dentro do automóvel para realizar tarefas simples do veículo.

### III. REQUISITOS

- Projeto deve ser feito em plataforma RaspBerry.
- Projeto deve ser capaz de ser implementado em qualquer veículo, mesmo após a montagem feita pela montadora.

- Projeto deve ser capaz de receber comando de qualquer tom de voz.
- O produto deve acionar após ser falada uma frase ou palavra específica evitando confusões de frases não relacionadas a instrução.
- O produto deve atender pelo menos os seguintes comandos: Ligar farol, ligar som, aumentar volume, ligar ar condicionado.

### IV. BENEFÍCIOS

A implementação do projeto traz a facilidade de acionar ou ajustar comandos básicos do carro, sem que seja necessário retirar a atenção do trânsito, tarefa realizada apenas com o comando de voz. Tais tarefas que poderiam ocasionar acidentes graves caso não fossem feitas com a devida segurança.

### V. IMPLEMENTAÇÃO

Foram utilizados os seguintes componentes:

- Raspberry pi.
- Modulo relé.
- Microfone USB (possui conversos A/D integrado).

Na implementação foi utilizado a biblioteca Jasper, que foi instalada na Raspberry Pi seguindo o tutorial do próprio site da Jasper. Pode-se configurar duas vertentes, Speech-To-Text (STT), que converte o comando de voz para texto, que no caso utilizamos o Google API (STT), e podemos configurar o Text-To-Speech (TTS), que é exatamente o oposto do STT.

Após o microfone captar o “comando de voz”, ele faz a conversão do sinal analógico para o sinal digital. Dessa forma, a Jasper envia o “comando de voz” para a Google API via internet que é responsável por fazer a conversão do áudio para texto, quando concluído, é mandado de volta o texto convertido.

Após ter o áudio convertido para texto, o mesmo é comparado com as strings que foram adicionadas no arquivo “commands.conf”, a Jasper faz a comparação do texto convertido com a string, conforme na figura abaixo:

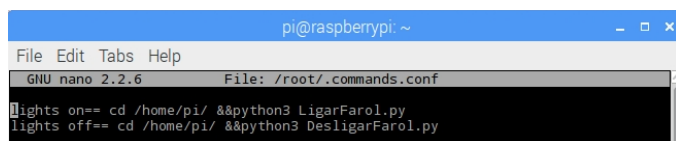


Figura 01: Arquivo commands.conf

Se o texto que foi convertido for “lights on”, a Jasper vai abrir o arquivo em python chamado AcenderFarol.py, que fará a habilitação dos pinos de saída.

Já quando o texto convertido for “lights off”, a Jasper vai abrir o arquivo em python chamado DesligarFarol.py, que será responsável por desabilitar os pinos.

Por enquanto, a Jasper só está atendendo falas em inglês porque o Google API está configurado para receber falas em inglês, mas será modificado para aceitar falas em português no próximo ponto de controle.



Figura 02: Arquivo AcenderFarol.py

Na figura acima observamos o código utilizado para habilitar o pino 7 de saída. Como o pino de saída da raspberry é 3.3 V, foi feita uma implementação em hardware para que a saída direcionada para o farol do carro seja de 12 V, pois é com 12 V que o farol do carro é acionado.

Para fazer a implementação em hardware, foi utilizado um transistor BC457 e um relé. O pino de saída em vez de estar conectado diretamente ao LED, está conectado ao transistor, que tem como função fazer chaveamento do relé.

Quando o pino 7 está em alto, seu valor é de 3.3V, como está conectado ao transistor no gate, o  $V_{GS} > V_{TH}$ , portanto o transistor vai conduzir corrente do dreno para source, fazendo com que o relé seja habilitado, fornecendo uma saída de 12 V.

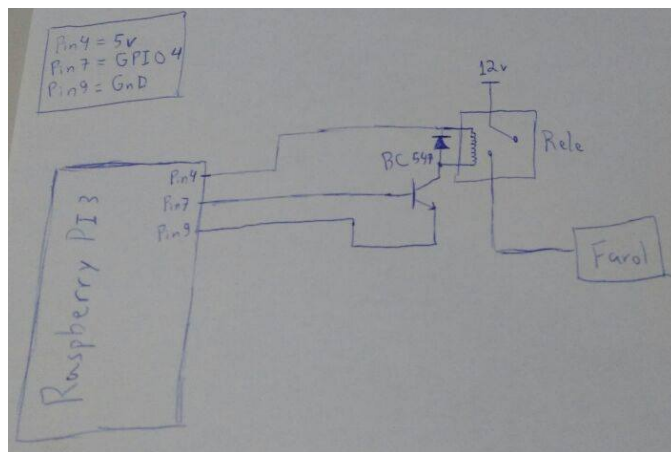


Figura 03: Implementação em hardware

## VI. PRÓXIMOS PASSOS

Considerar a troca do Controle de voz Jasper pelo Controle de voz Judy, pois é mais dedicado ao objetivo do projeto.

Como foi dito, existem duas ferramentas chamadas STT e TTS, por enquanto, estamos utilizando apenas o STT da Google API, mas existe possibilidade de mudança para o STT da Pocketsphinx, pois possui a vantagem de não utilizar internet para fazer a conversão do áudio pra texto. Será adicionada a ferramenta TTS da SVOX PICO, para que a raspberry se comunique por comando de voz conosco.

A vantagem de adicionar a ferramenta TTS é que vamos poder realizar uma pergunta para a Raspberry, e a mesma terá que responder por “comando de voz”.

Adicionar as implementações em hardware para fazer o acionamento do ar-condicionado e do sistema sonoro.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Jasper Project, configuração Jasper, disponível em: <[www.jasperproject.github.io/documentation](http://www.jasperproject.github.io/documentation)> Acesso em 30 de março de 2016
- [2] RaspBerry PI, controle de voz utilizando Siri, disponível em: <[www.raspberrypi.org](http://www.raspberrypi.org)> Acesso em 30 de março de 2016
- [3] Mitchell, M., Oldham, J. & Samuel, A., Advanced Linux Programming, Editora: Newriders, 2001.
- [4] Source Forge, siri proxy, disponível em: <<https://sourceforge.net/p/siriproxyrpi/wiki/Home/>> Acesso em 30 de março de 2016

