

Automação Residencial por Voz para Pessoas com Mobilidade Reduzida Empregando Jasper e Raspberry Pi

N. de V. SOUSA*, S. E. N. CORREIA*. S. C. COSTA*.

*Programa de Pós-graduação em Engenharia Elétrica, Instituto Federal da Paraíba, João Pessoa, Brasil

e-mail: naironsousa@yahoo.com.br

Abstract – *The number of people with some type of physical disability already amounts to around 23.9%, according to IBGE data. Lately the growth of technologies directed to Residential Automation has been notable. Known as intelligent systems, they are able to identify people and an infinity of actions, such as the voice, through specific sensing allied to a microcontrolled system and, thus, enabling human machine interaction (HMI). In this context, this article presents the development of a low-cost automatic speech recognition system using the embedded Raspberry Pi platform, applied in a small-scale residential automation project for people with reduced mobility. Voice recognition was implemented through the Jasper software, which acts as a transmitter and receiver of the data, which in turn processes and implements a pre-established control action on the electrical components (Actuators) in the residential project. In this project, the results and discussions about the proposed system will be presented in terms of efficiency and effectiveness, after conducting practical tests.*

Palavras-chave: *Domótica, Inteligência Artificial, reconhecimento de voz, Raspberry Pi.*

Introdução

Sasaki [1] define uma pessoa com deficiência como sendo aquela que possui um comprometimento físico, sensorial ou mental, o qual lhe traz limitação, incapacidade ou desvantagem em relação à categoria das pessoas sem nenhum tipo de deficiência. Já para o conceito de pessoa com mobilidade reduzida segundo resolução da ONU, Decreto nº 5.296/2004, Art. 5º, é aquela que “não é portadora de deficiência, mas tem dificuldade de movimentar-se, de flexibilidade, coordenação motora e percepção” [2].

Projetos de automação com o intuito de auxiliar pessoas com algum tipo de deficiência utilizando sistemas embarcados e voz já foram testados, utilizando placas principalmente como Arduino, Raspberry Pi e BeagleBone, e *softwares open source* que apresentam boa taxa de aceitação e recursos

disponíveis. As duas últimas apresentando um melhor custo benefício, pois possuem processadores que comportam os *softwares* de reconhecimento e com o mesmo tamanho físico.

Perico *et al.* [3], teve como objetivo controlar uma plataforma elevatória para cadeiras de rodas, disponibilizada pelo PROTA (Programa de Tecnologia Assistiva), implementando um sistema de reconhecimento de voz utilizando a plataforma Raspberry Pi. O comando de voz era transmitido para o Raspberry através do programa de reconhecimento de voz, que utilizava o decodificador Julius e um código em Python. O modelo acústico utilizado foi o LapSAM e o dicionário fonético o UFPAdic 3.0, ambos do grupo de pesquisa Fala Brasil. Os estudos foram executados com a participação de cinco voluntários. Obteve-se um bom tempo de resposta, baixo custo de reconhecimento de voz. Além disso, o sistema era totalmente independente de conexão com a internet ou qualquer outra fonte de dados.

Pimentel *et al.* [4] propuseram uma plataforma para reconhecimento de voz utilizando um módulo de captura (*Voice Recognition Module V2*) e uma placa embarcada BeagleBone Black, utilizando o sistema operacional Ubuntu Linux, com um ambiente de desenvolvimento Python, onde o treinamento dos comandos para validação são realizados anteriormente no próprio módulo. A ativação/desativação de cargas é feita através do envio de comandos via interface serial entre as placas empregadas. Foram obtidas respostas adequadas aos comandos, mesmo pronunciadas por indivíduos diferentes, com uma taxa média global de acertos de 95,9%, mesmo em ambientes ruidosos.

Neste contexto, este trabalho visa apresentar um sistema para o reconhecimento de fala automático, onde será utilizado a plataforma embarcada Raspberry Pi B+ e o *software* Jasper, esse desenvolvido recentemente e especificamente para tal plataforma que, atualmente, tem vasto recurso de código aberto e custo reduzido, atendendo ao ideal do projeto. O sistema é voltado para pessoas com algum tipo de necessidade especial, que tenha a voz

em perfeito estado, com o intuito de auxiliá-las em atividades básicas do dia a dia, em sua residência.

O texto do trabalho encontra-se organizado da seguinte forma: na Seção de materiais e métodos será apresentada a placa embarcada, o *software* de reconhecimento utilizado e a placa do circuito implementado. Em seguida, são apresentados os resultados obtidos nos testes de validação do sistema. Logo após são apresentadas as discussões e conclusões, além de recomendações futuras para melhoramento do sistema.

Materiais e métodos

O projeto baseia-se no fluxograma apresentado na Figura 1. O Raspberry Pi B+ embarca o *software* de assistência Jasper, que após a captura do comando dito pelo usuário, processa e o valida, podendo ser utilizado para as mais diversas aplicações. Os comandos de voz são enviados pelo Jasper, através da internet, para serem reconhecidos pelo Speech API STT, onde se encontra o *PocketSphinx*, que decodifica a fala. Após a decodificação dos comandos de voz, o Speech API STT devolve os comandos em forma de texto para o Raspberry Pi, que são traduzidos para o código ASCII e a ação é realizada na placa de circuito, correspondente ao comando reconhecido para os dispositivos a serem acionados.

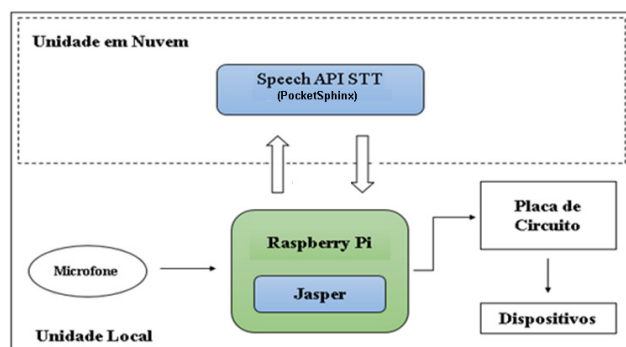


Figura 1. Modelo geral da metodologia empregada.

Jasper - O Jasper criado em 2014 por Charles Marsh e Shubhro Sahaé, é uma plataforma de código aberto instituído especificamente para permitir a interação de usuários com o Raspberry PI através de comandos de voz. Além de comandos de ativação de periféricos que podem ser implementados através de uma interface simples em Python, ele disponibiliza informações e notificação de redes sociais através de autofalantes integrados à placa [5].

Com o sistema operacional preparado, foi feita a instalação e as configurações no Jasper, como o STT a ser usado, modelo de linguagem, vocabulário,

nível de hierarquia dos módulos e a inserção de um Script de interface com GPIO's (General Purpose Input/Output).

Como o objetivo é que seja um dispositivo totalmente *open source*, optou-se por usar na aplicação o *PocketSphinx* STT engine na língua inglesa, que é uma das alternativas sugeridas para o Jasper, para a conversão de voz para texto.

PocketSphinx - É um decodificador de fala contínua independente de falantes, de código aberto baseado no projeto CMU *Sphinx*, que utiliza modelos acústicos escondidos de Markov (HMMs) e um modelo de linguagem estatística de *n*-grama. O *PocketSphinx* foi especialmente adaptado para dispositivos portáteis e móveis que tenham em sua estrutura um processador ARM.

Raspberry Pi B+ - O Raspberry Pi B+ (Figura 2) é um pequeno computador embarcado de desenvolvimento, barato e educacional. Os sistemas operacionais suportados utilizam o núcleo Linux, em sua maioria gratuitos e de código aberto.

A placa conta com um processador de um chip Broadcom BCM2835 de 700 MHz, 32 bits, construído sobre a arquitetura ARM.

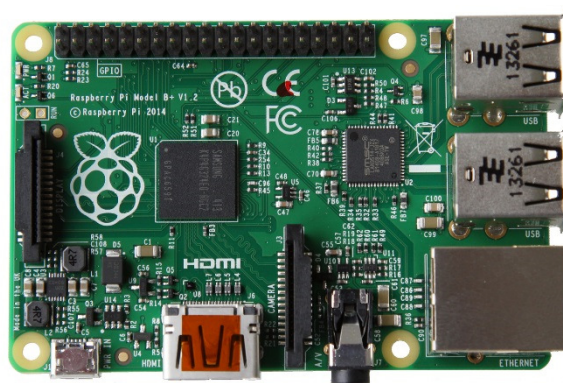


Figura 2. Plataforma embarcada Raspberry Pi B+.

O modelo B+, escolhido para este projeto, tem 512 MB de memória RAM, Slot para cartão de memória micro SD (*Secure Digital*), 4 portas USB 2.0 podendo fornecer até 500mA de corrente [6].

De início, se utilizou um adaptador Wi-Fi para instalação do sistema operacional *Raspbian*, que é uma variante do Debian baseada no ARM *hard-float*, sendo um porte da arquitetura Wheezy.

Com o ambiente preparado, foi feita a instalação e as configurações necessárias do Jasper com o *PocketSphinx* e do driver de áudio, ajustando a sensibilidade do microfone ao ambiente. Para a captura, será adaptado um driver externo USB de áudio (*virtual 7.1 Ch Sound*), pelo fato do Raspberry Pi não possuir entrada para microfone.

Para o propósito da automatização, foi necessário fazer algumas alterações no Jasper, baseadas na proposta de Brasizza [7]. Criou-se um módulo (*script* em Python) para uma interface com as GPIO's do Raspberry Pi, que por sua vez controla a placa de circuito que é apresentada na Figura 3, ativando e desativando cargas.

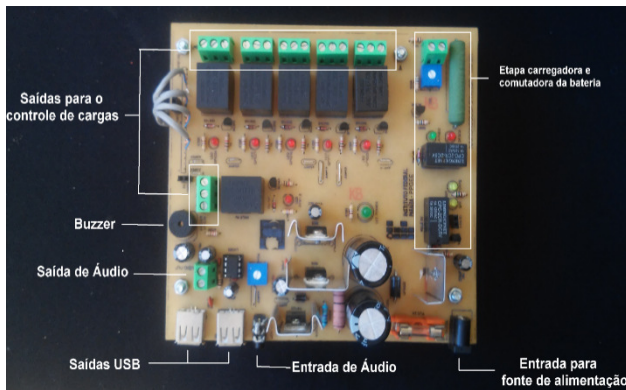


Figura 3. Placa de circuito para o controle de cargas.

No módulo, foram postas as palavras-chave, conforme a Tabela 1, que deveriam ser reconhecidas e validadas, realizando as ações pré-configuradas para as GPIO's.

Tabela 1. Comandos definidos.

Comandos		Ação
Door	Open	Ativa relé 1
	Close	Desativa relé 1
Garden	Light on	Ativa relé 2
	Light off	Desativa relé 2
Kitchen	Light on	Ativa relé 3
	Light off	Desativa relé 3
Bethroom	Light on	Ativa relé 4
	Light off	Desativa relé 4
Hall	Light on	Ativa relé 5
	Light off	Desativa relé 5

Placa de circuito - Para a prototipagem, foi feita uma placa de circuito impresso (Figura 3) que é acoplada ao Raspberry Pi B+ por meio de conectores, facilitando assim o acionamento das cargas que serão controladas. Consta com relés para o controle de até 6 cargas, com tensão nominal de até 220 V com uma potência máxima de 2200 W, como lâmpadas, ventiladores, entre outros. Possui, ainda, um *buzzer* para indicação sonora, fusível para proteção contra sobrecarga e entrada para bateria externa de 6 V.

Na Figura 4 podem ser vistos os passos seguidos para a realização de ativação das cargas.

O circuito pode ser alimentado com uma fonte externa, de preferência chaveada, com um suprimento de 9 a 12 Volts por 2 Ampères, onde, CI's reguladores de tensão fazem a conversão para os 5 Volts necessários para o funcionamento do Raspberry Pi B+ e dos outros periféricos.

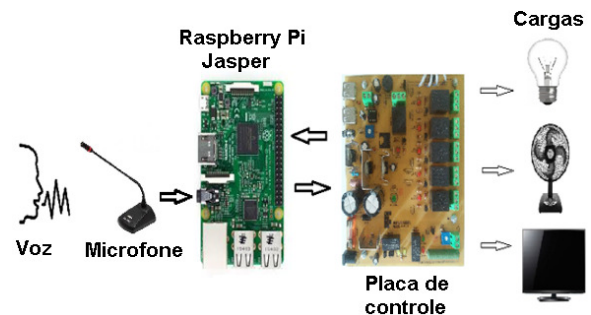


Figura 4. Diagrama de funcionamento do projeto.

Resultados

Os testes foram realizados com cinco voluntários com idade entre 18 a 27 anos, sendo três homens e duas mulheres, todos com nível de inglês básico, não fluentes na língua. Os testes foram realizados em um ambiente domiciliar comum, sem as cargas ligadas ao circuito. Os participantes fizeram a pronúncia de cada conjunto de palavras cinco vezes, obtendo um total de 50 comandos pronunciados.

Os resultados com as taxas de acertos para cada usuário em porcentagem, e a média geral podem ser vistos em seguida na Tabela 2.

Tabela 2. Resultados taxa de acertos

Usuários	Acertos
1º Voluntário	92 %
2º Voluntário	88 %
3º Voluntário	96 %
4º Voluntário	90 %
5º Voluntário	86 %
Média geral de acertos	90,4 %

Os resultados foram satisfatórios, só em poucos casos os comandos ditos não foram reconhecidos, que podem ter sido ocasionados por alguns fatores:

- Pela pronúncia incorreta do idioma;
- Qualidade do microfone;
- Ruídos externos.

Foi observado que os comandos com o conjunto de três palavras eram mais difícil de serem reconhecidos, como por exemplo "Hall Light off".

A eficiência por outro lado se mostrou favorável, sendo a ação correspondente realizada para cada comando aceito, ou seja, as GPIO's eram ativadas ou desativadas instantaneamente.

Discussão e conclusões

Com os testes preliminares com o Jasper utilizando o *PocketSphinx* e a plataforma embarcada Raspberry Pi, comprovaram a viabilidade de se implementar uma proposta de interface por interação por voz de baixo custo, o que torna o dispositivo mais acessível, visando uma automação residencial para pessoas com algum tipo de mobilidade reduzida.

A grande facilidade de manipulação da plataforma e do *software* torna fácil a adaptação e o controle dos mais diversos dispositivos externos, como eletrodoméstico e a parte elétrica do ambiente (interruptores e tomadas).

A placa de circuito atendeu a proposta esperada de facilitar a comunicação entre o Raspberry Pi e o circuito de controle das cargas, observando que os conectores (KRE) de ligação às cargas externas são de fácil manuseio, podendo assim o sistema ser adaptado a qualquer instalação elétrica ou aparelho já instalado em uma residência.

Para projetos futuros, pode ser vista a substituição do *PocketSphinx STT engine*, por um API (*Application Programming Interface*) com suporte para o idioma português Brasil, que faria o sistema ter um índice de reconhecimento mais elevado, por ser o idioma nativo, o que melhoraria a eficácia consideravelmente.

Foi gravado o resultado da validação do módulo, tendo sido o vídeo disponibilizado na internet, através do seguinte link, no YouTube: <https://youtu.be/qvYYyk1J3XA>.

Agradecimentos

Ao IFPB pela ajuda financeira.

Referências

- [1] Sasaki, RK. Como chamar as pessoas com deficiência. Vida independente: história, movimento, liderança, conceito, filosofia e fundamentos. São Paulo: RNR, 2003. 12-16 p.
- [2] BRASIL. Decreto no 5296, de 02 de dezembro de 2004. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm. Acessado em: 04/08/2017.

[3] Perico, A et al. Sistema de reconhecimento de voz para automatização de uma plataforma elevatória Trabalho de Conclusão de Curso (Graduação - Curso de Engenharia Industrial Elétrica Ênfase em Automação). Universidade Tecnológica Federal do Paraná, 2014, 97f.

[4] Pimentel, VCA et al. Uma plataforma de baixo custo comandada por voz para tecnologias assistivas com programação em python. In: Congresso Brasileiro de Engenharia Biomédica, XXIV, MG. Anais..., Uberlândia: Universidade Federal de Uberlândia, 2014, pag. 1414-1417.

[5] Jasper documentação. Disponível em: <http://jasperproject.github.io/documentation/>. Acessado em 15/08/2017.

[6] Richardson, M. e Wallace, S. Primeiros passos com o Raspberry Pi. São Paulo: Novatec Editora, 2013.

[7] Brasizza, M. Caixa de casamento no RASPBERRY. Canal raspberry. Brasil, jan. 2016. Disponível em: <https://embarcados.club/caixa-de-casamento-no-raspberry/>. Acessado em: 10/06/2017.