

UTILIZAÇÃO DA PLATAFORMA ARDUINO E DO RASPBERRY PI NA CONSTRUÇÃO DE UM ROBÔ VEICULAR COM CÂMERA

Genival Alves dos Anjos Filho¹, Bruno Silvério Costa², Camilo Alves Carvalho²

¹Graduando do Curso de Engenharia Elétrica – Instituto Federal de Educação Ciência e Tecnologia (IFBA) – Av. Amazonas, nº 3.150, Bairro Zabelê, 45.075.265 – Vitória da Conquista, BA – Brasil.

^{2 e 3}Professor do Instituto Federal de Educação Ciência e Tecnologia (IFBA) – Av. Amazonas, nº 3.150, Bairro Zabelê, 45.075.265 – Vitória da Conquista, BA – Brasil.

genival.anjos@yahoo.com.br, brunosilverio@ifba.edu.br,
camilocarvalho@ifba.edu.br

Abstract. *The autonomous mobile robotic systems has increasing been part of the human entertainment and in the performance of tasks considered unhealthy and dangerous. Its development has become more popular in both industry and universities, as a result of the great improvement in the industrial automation systems. In this context, this article presents the development of an autonomous mobile robot that has a vision system responsible for the perception of the environment that it will perform. The robot build is a prototype that has the Arduino platform, two stepper motor and a camera.*

Resumo. *Os sistemas robóticos móveis autônomos têm a cada dia mais feito parte do entretenimento do ser humano e das realizações de tarefas insalubre e perigosas. O seu desenvolvimento tem conquistado espaço tanto na indústria quanto nas universidades, devido à intensa modernização dos sistemas de automação industriais. Nesse contexto, o texto apresenta o desenvolvimento de um robô móvel autônomo que possui um sistema de visão responsável pela percepção do ambiente em que atua. O robô construído é um protótipo que possui uma plataforma Arduino, um microcomputador Raspberry Pi, um módulo de potência, dois motores de passo e uma câmera.*

1. Introdução

Os sistemas robóticos móveis autônomos têm conquistado espaço tanto nas universidades quanto na indústria, devido à intensa modernização dos sistemas de automação industriais nos últimos anos. Entre os fatores que impulsionam essa modernização pode-se incluir a competitividade crescente, a rápida alteração dos produtos oferecidos ao mercado, o avanço tecnológico, entre outros, que visam aumentar a produtividade, a qualidade e a confiabilidade dos produtos. Além da área de manufatura e transporte de materiais, outras aplicações destes sistemas incluem o trabalho em ambientes perigosos ou insalubres e a exploração espacial (COSTA, 2003).

O grande avanço na área da robótica somente foi possível com o surgimento do microprocessador, pois o controle da trajetória dos robôs articulados envolve cálculos complexos que devem ser realizados em tempo real (ROSARIO, 2005). De modo geral,

devido aos inúmeros recursos que os sistemas microprocessados oferecem, tem feito à robótica atravessar uma época de contínuo crescimento.

Particularmente, em aplicações com manipuladores robóticos, (CETINKUNT, 2008) afirma a necessidade de a máquina fazer uma inspeção do ambiente (por exemplo, utilizando sistema de visão) e decida a estratégia de movimento para cada eixo individual. A interpretação dos dados de sensores e a geração de estratégias de movimentos inteligentes em tempo real constituem um dos atuais desafios no controle do movimento programado de dispositivos robóticos.

Ainda segundo (CETINKUNT, 2008), existem três componentes principais em um sistema de visão:

- a) Câmera: é a cabeça do sensor, feita com um dispositivo de arranjo fotossensível, e o circuito de conversão analógico-digital (A/D) para converter o sinal analógico para a forma digital.
- b) Computador e software para processamento de imagens.
- c) Sistema de iluminação.

Uma parte importante e frequentemente negligenciada é quanto ao sistema de iluminação. O sistema de visão forma uma imagem pela medição de luz refletida de objetos em seu campo de visão. Desse modo, a luz refletida depende muito da fonte de luz, que deve ser inserida seja com a luz ambiente ou estruturada para uma melhor inspeção de superfícies.

Dentro dessa abordagem (BIANCHI, 2000), afirma que a visão computacional não é mais vista como um problema contido em si mesmo, mas sim como um sistema que interage com o ambiente através da percepção e ação. O objetivo deste trabalho é a descrição do projeto de um robô móvel que possui um sistema de visão desenvolvido em um nível introdutório, para a inspeção de uma superfície local a fim de reconhecer a posição de uma linha (pista). O caminho desejado não é conhecido a priori, além disso, se por qualquer motivo o desenho da pista for obstruído o robô deve reconhecer e parar.

Para atingir esse objetivo, foi construído um robô constituído por uma plataforma Arduino, um microcomputador *Raspberry Pi*, um módulo de potência, dois motores de passo e uma câmera para aquisição de imagens. Dentre os dispositivos citados, a ênfase maior será dada ao *Raspberry Pi* que é um computador do tamanho de um cartão de crédito, e assume um papel importante para o sistema de visão no projeto, e também no envolvimento do estudante com o *Software Livre*.

2. Material e Métodos

A utilização do computador digital como um dispositivo compensador cresceu nas últimas três décadas conforme o preço e a confiabilidade dos computadores digitais têm crescido consideravelmente. Estima-se que atualmente existam aproximadamente 100 milhões de sistemas de controle usando computadores, embora o tamanho e a potência possam variar de forma significativa (DORF, 2011).

O *Raspberry Pi* é um microcomputador de dimensões reduzidas, desenvolvido pela *Raspberry Pi Foundation* e pela Universidade de *Cambridge*. A Figura 1 ilustra

esse microcomputador onde todo o hardware é integrado numa única placa de circuito impresso.

2.1. O *Raspberry Pi*

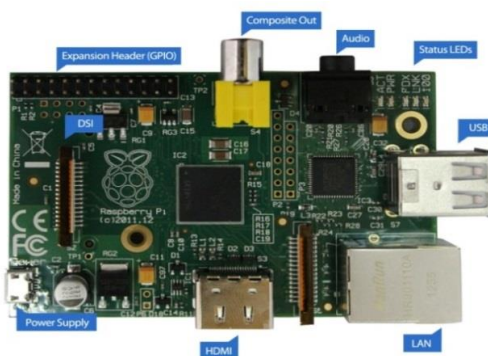


Figura 1. *Raspberry Pi* modelo B.

O microcomputador modelo B possui um controlador Ethernet e duas portas USB, sendo o diferencial desse modelo em relação ao A. Além disso, dentre os sistemas operativos compatíveis para o microcomputador, foi instalado o *Raspbian*, que é uma distribuição Linux baseada em Debian, e oferece bastante suporte sendo oficialmente recomendada pela fundação *Raspberry*.

O processador *Broadcom* tem algumas características proprietárias que exigem *drivers* de dispositivos especiais (“*binary blob*”) e programas que não estão incluídos em nenhuma distribuição Linux padrão. Além disso, enquanto a maioria dos computadores *desktop* tem gigabytes de memória RAM (*Random Access Memory*) e centenas de *gigabytes* de armazenamento, o *Pi* é mais limitado em ambos os aspectos (RICHARDSON, 2013). Esse tipo de limitação não será um empecilho para a sua aplicação no projeto, pois quando se trata de robótica uma minimização no uso dos recursos é essencial. E para isso é necessário o uso de dispositivos de baixo custo, mas que seja suficiente para as aplicações, mesmo tendo a capacidade de processamento reduzida.

2.2. A Plataforma Arduino

Para compor a unidade robótica também foi utilizada a plataforma Arduino, sendo esta uma ferramenta “*open source*” que possui uma placa microcontroladora baseada no ATmega 328 da família AVR.

Segundo os seus desenvolvedores, o Arduino é capaz de perceber o ambiente por receber a informações de uma variedade de sensores e pode estimular um ambiente controlando luzes, motores e outros atuadores.

Outro aspecto importante é que o padrão Arduino permite a conexão de módulos periféricos, denominados “*shields*” (“escudos”), disponíveis comercialmente, para expansão do sistema, atendendo a uma grande variedade de aplicações (NOGUEIRA, 2011). Desse modo, os *shields* são placas que se encaixam em cima do Arduino para aumentar a sua funcionalidade de forma específica, nesse trabalho foi construído um *shield* denominado módulo de potência.

3. Resultados e Discussões

3.1. O Sistema Visual Desenvolvido

O sistema visual desenvolvido é composto por uma câmera do tipo webcam devido ao seu baixo custo, sendo esta posicionada na parte superior do robô.

O primeiro processo para um sistema de visão é a aquisição de imagem. A Figura 2 apresenta a imagem capturada pela câmera de 220x340 pixels de resolução, caracterizando o teste inicial da Webcam.

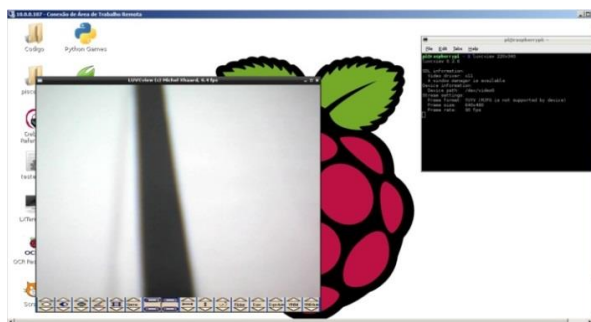


Figura 2. *Raspberry Pi* modelo B.

Após a aquisição de imagem, esta foi submetida ao *HasciiCam* que é um *software* portátil para o sistema operacional Linux produzido pelo italiano Jaromil Jan Hubicka. O *HasciiCam* é capaz de capturar o vídeo da Webcam e convertê-lo em letras *ASCII* (*American Standard Code for Information Interchange*) formatando a saída em uma janela *ASCII* de tempo real, ou a imagem em um arquivo de texto simples, sendo esse o propósito da sua aplicação nesse projeto. A Figura 3 é o resultado da conversão da imagem obtida pela webcam em letras *ASCII*.

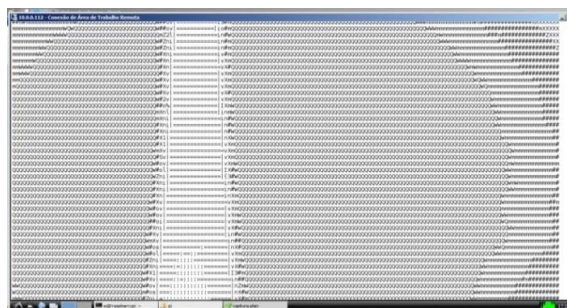


Figura 3. Imagem convertida para texto *ASCII*.

Em relação a Figura 3, vale salientar que durante os ensaios a parte da imagem em que se encontra a pista, geralmente eram composta pelos caracteres (‘.’ ‘;’ e ‘=’). Desse modo foi definido um mecanismo de controle para a direção do robô a partir da utilização da imagem convertida em texto *ASCII*. Para tanto, foram divididas horizontalmente cinco regiões em partes iguais: Muito a Esquerda, Esquerda, Centro, Direita e Muito a Direita, conforme ilustra a Figura 4.

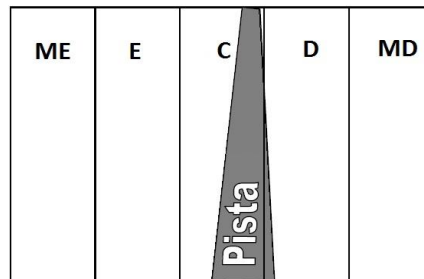


Figura 4. Áreas definidas para o controle da direção.

Após a imagem ser convertida, a saída em texto *ASCII* passou a ter um tamanho de 160x60, com isso as regiões da linha selecionada foram definidas como:

- Muito a Esquerda: $0 \leq ME < 30$.
- Esquerda: $31 \leq E < 63$.
- Centro: $64 \leq C < 96$.
- Direita: $97 \leq D < 129$.
- Muito a Direita: $130 \leq MD < 160$.

Este sistema foi desenvolvido por um mecanismo capaz de interpretar os dados obtidos por um sistema de visão para em seguida gerar as estratégias de movimento. Para que tal desenvolvimento fosse possível, após as etapas de conversão da imagem em texto *ASCII* e ter as regiões de atuação definidas, foi utilizada a linguagem *Python* como uma ferramenta para obter referência ao objeto do tipo arquivo de texto e assim ser feita uma varredura da primeira linha a fim de encontrar a posição da pista. Além dessas funcionalidades, o código contou com o auxílio da biblioteca *PySerial* que permite fazer os dados serem enviados ou lidos pela porta serial, isto foi utilizado para que o *Raspberry* envie informações correspondentes a orientação do robô para o Arduino controlar os motores. É mostrado na Figura 5 um trecho do código desenvolvido no (*SPE*) *Stani's Python Editor*.

```

59 # para Muito a direita
60 cont5 = 0
61 for x in linha[130:160]:
62     if (x == '.' or x == '=' or x == ':'):
63         cont5 = cont5+1
64     if cont5>5:
65         Muito_Direita = True
66     else:
67         Muito_Direita = False
68
69 while True:
70     if Centro == True:
71         print 'Ativa os dois motores'
72         ser.write('1')
73     elif Esquerda == True:
74         print 'Ativa o motor direito'
75         ser.write('2')
76     elif Direita == True:
77         print 'Ativa o motor esquerdo'
78         ser.write('3')
79     elif Muito_Esquerda == True:
80         print 'Ativa 2 passos o motor direito'
81         ser.write('4')
82     elif Muito_Direita == True:
83         print 'Ativa 2 passos o motor esquerdo'
84         ser.write('5')
85     else:
86         print "Linha nao encontrada"
87         break
88
89
90 arq.close()

```

Figura 5. Trecho final do código desenvolvido no Python.

3.2. Interligando o Arduino ao Raspberry Pi via GPIO UART

Como os computadores trabalham internamente com caracteres inteiros, mas precisam se comunicar por uma linha serial, um bit por vez, foram desenvolvidos chips para converter de caractere para serial e vice-versa, são os chamados *UART's (Universal Asynchronous Receiver/Transmitter)*, de forma que permite enviar e receber caracteres simultaneamente (TANEMBAUM, 2008).

Para desenvolver a comunicação entre os dois dispositivos via *GPIO (General Purpose Input/Output) UART (Universal Asynchronous Receiver/Transmitter)*, foi construído um circuito conversor de nível lógico. Em relação aos níveis de tensão, o *Raspberry Pi* trabalha com 3.3 V enquanto o Arduino com 5 V. Esse conversor é um dispositivo que baixa o nível de 5 V para 3,3 V ou aumenta o nível de 3,3 V para 5 V, o circuito possui dois transistores *MOSFET's (Metal Oxide Semiconductor Field Effect Transistor)* sua característica de chaveamento é rápida, sendo um transistor utilizado para transmissão e outro na recepção. O esquema elétrico do circuito conversor é mostrado na Figura 6.

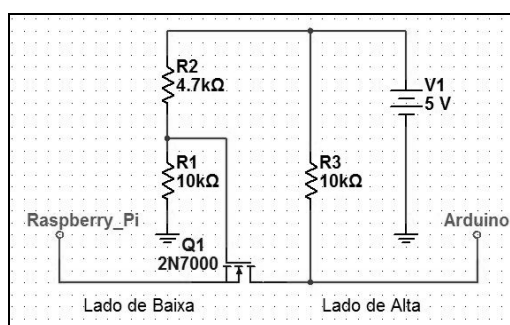


Figura 6. Esquema elétrico do circuito conversor de nível.

3.3. Módulo de Potência

Para acionar os dois motores de passo, foi utilizado o circuito integrado ULN 2803, internamente este *CI (Circuito Integrado)* possui oito *drivers*, cada *driver* é composto por dois transistores acoplados, chamado de transistor *Darlington*.

O módulo de potência foi projetado em uma placa de circuito impresso, sendo constituído por um circuito conversor nível lógico e mais dois *CI's ULN 2803 (driver)*, um *CI* para cada motor. Isso porque, segundo o *datasheet* do ULN 2803, é possível ligar as entradas e saídas em paralelo para aumentar o consumo máximo de corrente suportado pelo chip. Então, para aumentar a capacidade de corrente máxima, foram ligadas as entradas e saídas duas a duas, o que é suficiente para os motores extraídos de impressoras sobre o qual estamos usando. A Figura 7 mostra o resultado do desenvolvimento desse circuito no *software Eagle* versão estudante.

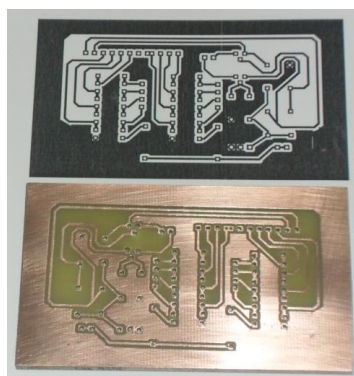


Figura 7. Confeccionando a placa de circuito impresso.

3.4. A Plataforma Robótica

O robô foi montado inicialmente em uma placa metálica, para sustentar e permitir a afixação das duas rodas traseiras aos motores de passo e uma roda dianteira menor do tipo livre. Em seguida, sobre uma placa de acrílico começamos a montar o *hardware* adicionando os dispositivos que constituem em: *Raspberry Pi*, Arduino, conversor de nível lógico, *driver* (módulo de potência) e Webcam. Em relação ao projeto da placa do módulo de potência, buscamos alinhar a maior parte dos seus conectores aos do Arduino, a fim de proporcionar um encaixe direto, o mesmo foi feito para encaixar os cabos dos motores, conectando-os na parte inferior da placa. A Figura 8 apresenta o protótipo construído.

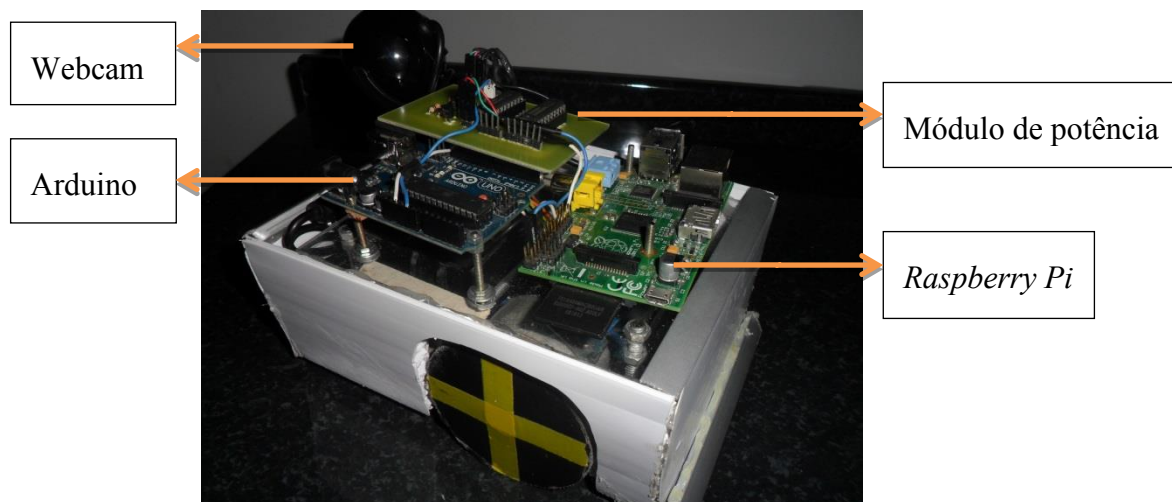


Figura 8. Protótipo construído utilizando motores de passo.

Segundo (CETINKUNT, 2008), diferente de uma câmera digital utilizada para tirar fotos, as imagens capturadas por um sistema computacional devem ser processadas em intervalos periódicos num ambiente de automação.

Os primeiros testes relacionados ao movimento do robô mostrou o que já era esperado, os motores só eram ativados a partir da primeira e única imagem, isso porque os comandos para acionar o *HasciiCam* e o código no *Python* eram executados apenas uma vez. Desse modo, foi desenvolvido um *Script (Shell)* para executar os comandos no terminal Linux por uma determinada quantidade de vezes de forma automática, sendo

possível atualizar as imagens a serem convertidas em arquivo de texto e executar o código *Python* para decidir qual a estratégia de movimento.

4. Conclusões

Os primeiros testes do código que realiza a identificação da posição da pista foram executados em um *notebook* com o sistema operacional Linux (Ubuntu 13.0) e possibilitou a demonstração da eficácia do código. No entanto, ao executar o *script* no *Raspberry* foi observado que a pista não era encontrada em alguns instantes. Isso pode estar ligado ao fato da baixa taxa de *frames*, algo em torno de seis *fps* (*frames* por segundo) conseguido nos ensaios feitos com a webcam pelo barramento USB. No entanto, existem alternativas recentes para melhorar isso, como uma mini câmera que permite a conexão ao *Raspberry* por um cabo fita, e tem uma interface *CSI* (*Camera Serial Interface*) desenvolvida especificamente para interface com a câmera, uma vez que oferece elevadas taxas de transferência de dados e permite enviar dados precisos e em tempo real para o robô.

Através desse trabalho pode-se notar como é possível desenvolver um sistema de visão a um nível introdutório, para a inspeção de uma superfície. Envolvendo de uma forma simplificada um conjunto de ferramentas computacionais relacionadas ao software livre, a fim de gerar um resultado conveniente a ser aplicado e desenvolvido por outros estudantes para que estes possam aprimorar e contribuir para a sociedade.

O desenvolvimento desse projeto também contribui pela utilização do *Raspberry Pi* e do Arduino em um único sistema. Isso foi motivado, devido à capacidade computacional do primeiro e a eficácia em nível de *hardware* do segundo. Essa conexão possibilita uma proteção maior ao minicomputador que possui seus conectores *GPIO's* ligados diretamente ao processador. Nesse contexto, a conexão de ambos dispositivos permitiu criar um sistema que atendesse as necessidades do projeto.

Tendo em vista os resultados obtidos, percebe-se que automação industrial tem sido um grande impulsionador da tecnologia de robótica, e esta por sua vez, tem permitido demonstrar a eficácia dos sistemas autônomos. A proposta para os trabalhos futuros é aperfeiçoar o processo de aquisição de imagens com a aplicação da mini câmera citada. Já se planeja também a realização de novos projetos como a construção de um veículo aéreo não tripulado baseado em *quadrotors* (helicóptero de quatro hélices) utilizando o *Raspberry Pi* e Arduino.

Agradecimentos

Ao CNPq, pela bolsa de iniciação científica que possibilitou a realização desse trabalho.

5. Referências

BIANCHI Reinaldo A. C. REALI-COSTA Ana H., **Sistema de Visão Computacional de um Time de Futebol de Robôs**. Congresso Brasileiro de Automática - CBA 2000, São Paulo, Brasil.

- COSTA, Eduardo R., GOMES Marcel L., BIANCHI Reinaldo A. C.: **Um Mini Robô Móvel Seguidor de Pistas Guiado por Visão Local**. VI Simpósio Brasileiro de Automação Inteligente, Setembro, 2003.
- CETINKUNT, Sabri. **Mecatrônica**, Rio de Janeiro: Editora LTC, 2008.
- DORF, Richard C., **Sistemas de Controle Modernos**. 11^a ed. Rio de Janeiro: Editora LTC, 2011.
- GOMES, R. F., **Introdução ao Shell Script no Linux**. Disponível em <<http://www.devmedia.com.br/introducao-ao-shell-script-no-linux/25778#ixzz2ZSIZoRhP>> Acesso em 03/06/2013.
- HASCIICAM, **HasciiCam: (h) Ascii for the Masses**. Disponível em <<http://www.dyne.org/software/hasciicam/>> Acesso em 25/11/2013.
- NOGUEIRA F. E., Matos R. R., Scolari R. D., Campos L. J. **Do Scratch ao Arduino: Uma proposta para o ensino introdutório de programação para cursos superiores de tecnologia**. Anais: XXIX Congresso Brasileiro de Educação em Engenharia, Blumenau - SC, 2011.
- RICHARDSON, Matt. **Primeiros Passos com o Raspberry Pi**. São Paulo: Editora Novatec, 2013.
- ROSÁRIO, J. M. **Princípios de Mecatrônica**. São Paulo: Pearson/Prentice Hall, 2005.
- TANENBAUM, Andrew S., **Sistemas Operacionais**. Porto Alegre: Editora Bookman, 2008.