

FCTBoat - Barco controlado remotamente utilizando Raspberry Pi e Android

Luiz Paulo Rabachini¹

¹Departamento de Matemática e Computação
Faculdade de Ciência e Tecnologia
Universidade Estadual Paulista
Presidente Prudente, SP – Brasil

luiz.rabachini@gmail.com.br

***Resumo.** O objetivo do presente trabalho é criar um protótipo de hardware que simule um barco e que seja controlado através do microcomputador Raspberry Pi. As instruções de navegação são recebidas por este através de um serviço local acessível via internet, que as processa e controla os movimentos do barco. As instruções de navegação são fornecidas pelo usuário através de um aplicativo para dispositivos móveis com sistema operacional Android, responsável pela coleta e envio dos comandos ao serviço remoto.*

1. Introdução

A fim de aplicar conceitos ligados a disciplina de FPGA, foi desenvolvido um protótipo de hardware com dispositivos eletrônicos que simula um barco e é controlado remotamente por um aplicativo executado em dispositivos com o sistema operacional Android. A seguir, são descritos os componentes utilizados nesse processo e a forma como interação.

2. Materiais e Métodos

Nesta seção serão abordados os principais elementos utilizados na elaboração deste trabalho. Ao final, é esperado que seja possível compreender melhor como a arquitetura do sistema foi criada e a forma como os elementos que a compõem funcionam.

2.1. Raspberry Pi

Com o objetivo de difundir a computação em comunidades carentes e incentivar crianças ao redor do mundo a criarem programas para computadores, foi criado em 2008 o microcomputador Raspberry Pi, que possui o tamanho aproximado de um cartão de crédito, saídas de áudio e vídeo, comunicação para rede ethernet e diversos outros recursos. Seu baixo custo fez com que fosse adotado rapidamente em diversos projetos, como automação residencial e robótica [Foundation 2013a].

Esse sistema é baseado na arquitetura ARM, criada inicialmente para dispositivos móveis e que possui um baixo custo de produção. Possui ainda hardware livre e conta com distribuições do sistema operacional Linux criadas especificamente para ele, como a Raspbian [Foundation 2013b], que fica armazenado em um cartão de memória do tipo SD. Seu processador possui clock de 700Mhz e 512Mb de memória do tipo CPU RAM,

que é integrada ao processador e outros componentes em um único chip SoC (System-on-a-chip). O sistema é alimentado a partir de uma fonte de energia externa que fornece uma tensão de 5V e 1A de corrente.

Por apresentar portas lógicas para entrada e saída (GPIO) controladas de forma simples através de bibliotecas desenvolvidas em linguagem de programação Python, esse dispositivo foi escolhido para controlar as funções do protótipo através de instruções recebidas pela interface de rede *wireless*, adicionada ao sistema através de uma interface USB. Os principais componentes disponíveis no Raspberry Pi podem ser visualizados no diagrama da Figura 1 e o aspecto geral do dispositivo é apresentado na Figura 2.

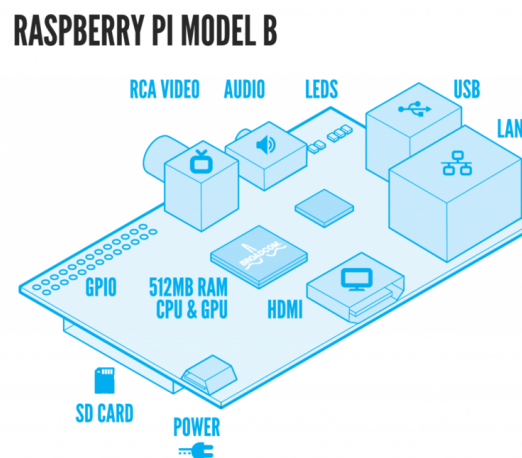


Figure 1. Principais componentes do Raspberry Pi. Fonte: [Foundation 2013a]



Figure 2. Aspecto geral do Raspberry Pi. Fonte: [Foundation 2013a]

2.2. Sistema Operacional Android

O sistema operacional Android foi desenvolvido com o objetivo de suprir as necessidades criadas por novos dispositivos de comunicação móveis, como *smartphones* e *tablets*, que possuem restrições de memória e processamento. É o sistema mais utilizado no mercado e possui um conjunto completo de ferramentas para desenvolvimento de aplicativos, que são criados em linguagem de programação Java.

Por contar com um SDK (Software Development Toolkit) completo e de fácil programação [Google 2013], este sistema foi adotado para construção da aplicação que coleta os dados para navegação e envia ao serviço remoto para controle do barco.

3. Desenvolvimento do Trabalho

O protótipo foi desenvolvido a partir de componentes eletrônicos encontrados em lojas do ramo e peças velhas de computador e pode ser visualizado na Figura 3.

Todo o sistema foi fixado sobre uma placa padrão para circuitos, utilizada por apresentar uma matriz de perfurações que facilitam a soldagem dos componentes. Na parte dianteira existem duas ventoinhas provenientes de fontes de alimentação para PC que proporcionam ao sistema movimentação lateral. Na parte de trás existe outra ventoinha responsável por impulsionar o conjunto para frente. A disposição desses elementos pode ser vista na Figura 3(a).

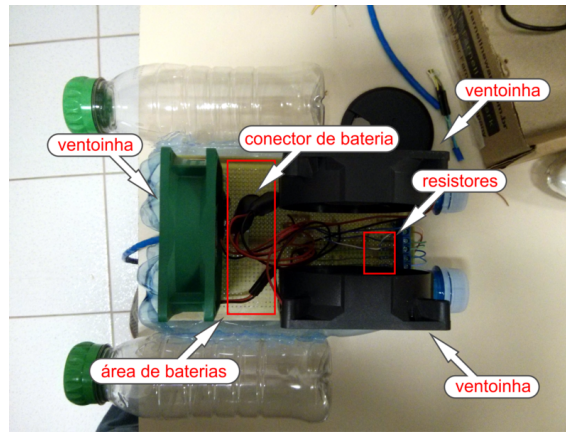
As ventoinhas utilizadas possuem tensão de 12V cada e são alimentadas por um conjunto de até três baterias de 9V, que são conectadas em série e proporcionam juntas 27V ao sistema. O excesso de tensão existente aqui não as danifica devido ao baixo período de autonomia que essas baterias proporcionam. Outra forma de alimentar o sistema é através de uma fonte externa, como as provenientes de computadores velhos, que foi utilizada nos testes efetuados.

Como o Raspberry Pi é capaz de fornecer no máximo 3.25V com uma corrente baixa nas portas GPIO, as ventoinhas são alimentadas pelo sistema através da ativação dos transistores fixados no centro da placa padrão. Esses transistores são do tipo NPN, que é ativado quando o pino base recebe uma tensão, no caso a proveniente do Raspberry Pi pelas portas GPIO, e permite a passagem de corrente entre os pólos coletor e emissor, no sentido do primeiro para o segundo. Cada ventoinha conta com um transistor desse tipo de forma independente. O pólo base de cada um é conectado a uma resistência a fim de que a corrente de 3.25V não chegue totalmente ao transistor, o que pode saturá-lo e comprometer o fluxo de energia. Para facilitar a conexão do conjunto de cabos entre o barco e o Raspberry Pi, foram utilizados seis bornes, em que é possível aparafusar os cabos. A função de cada ponto de fixação é descrita na Figura 4.

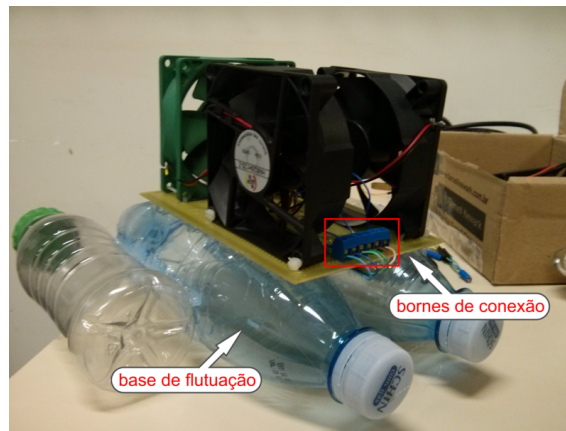
O serviço executado no Raspberry Pi, que recebe os comandos remotos e controla as ventoinhas do barco, foi implementado em linguagem de programação Python para facilitar a integração com a biblioteca de controle dos GPIOs, disponível em [btcraston@googlemail.com 2013]. Esse serviço opera na porta fixa 26000 e não permite usuários simultâneos para que não haja interferência nos controles. Os dados enviados pelo aplicativo e recebidos por ele são formatados da seguinte forma:

cmd&cmd&...&

O caractere '&' é responsável por separar os comandos de movimentação, ilustrados como 'cmd', que são enviados sempre de forma única, mas podem ser acumulados no buffer de dados do aplicativo e serem enviados em conjunto. Cada comando contém as intensidades do movimento geradas pelo usuário no sentido *x*, esquerda ou direita, e *y*, frente ou trás, e são separados pelo caractere ';' da seguinte forma:



(a)



(b)

Figure 3. Elementos que compõem o protótipo: (a) Vista superior. (b) Vista lateral.

$$v_x; v_y$$

Em movimentações frontais, v_y apresenta valor positivo e em movimentações para trás valor negativo. O mesmo vale para v_x , em que valores positivos indicam movimentação para direita e negativos para esquerda. Ambos, v_x e v_y , possuem valores que variam de -100 a 100 e correspondem ao percentual da barra de movimentação utilizada pelo usuário para descrever o movimento, como ilustra a Figura 5(c).

O aplicativo Android possui duas telas de interação. Na primeira, vista na Figura 5(a), são fornecidos o IP e porta do serviço de controle do barco executado no Raspberry Pi e é realizada uma tentativa de conexão. Caso seja bem sucedida, é exibida a tela com as opções de controle de movimentação. Caso contrário, é exibida uma mensagem de erro e uma nova tentativa deve ser realizada.

A tela de controle dos movimentos é composta por um referencial central, linhas guia indicando os eixos x e y e referenciais adicionais que indicam as posições correspondentes à 50% das intensidades máximas do movimento (Figura 5(b)). A medida que o usuário percorre a tela do aplicativo com o dedo pressionado, um referencial é exibido e

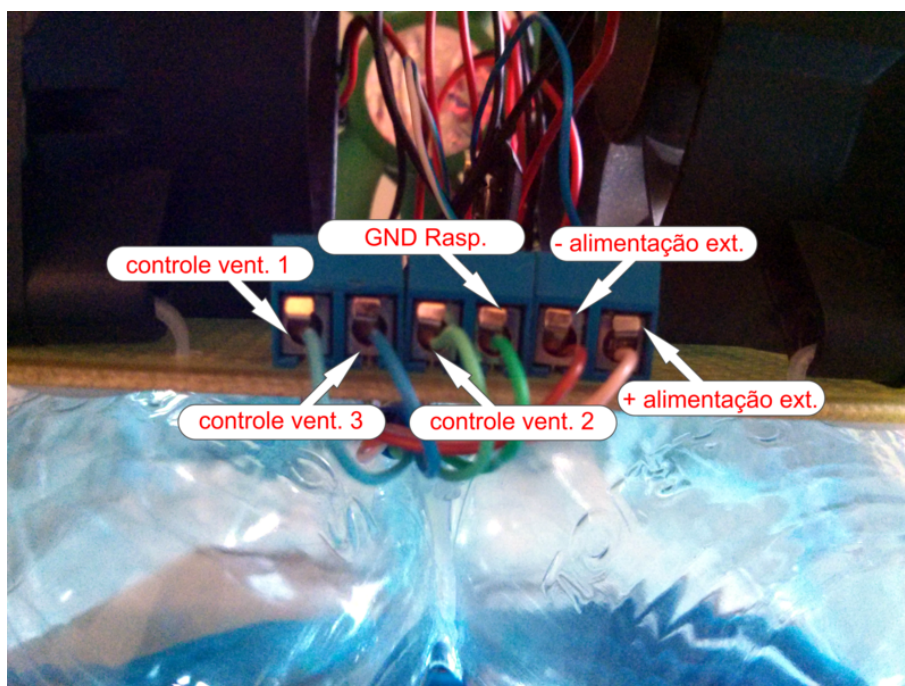


Figure 4. Conectores.

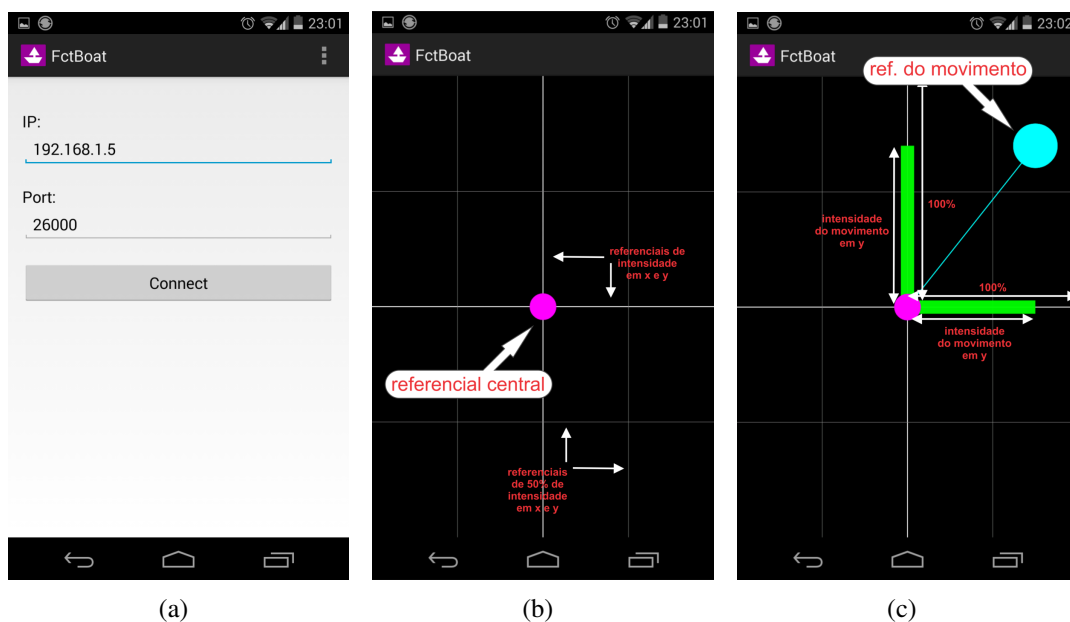


Figure 5. Interface da aplicação: (a) Conexão com o serviço remoto. (b) Tela de controle inicial. (c) Referenciais de controle.

as guias de intensidade dos movimentos, representadas pelas barras verdes, são alteradas (Figura 5(c)). Em cada alteração desse referencial é enviado um novo comando ao serviço remoto com as intensidades do movimento em x e y .

Como o protótipo apenas altera o estado das ventoinhas entre ligado e desligado, os valores das intensidades do movimento são analisados e caso ultrapassem o limiar de 50% do valor máximo (100 ou -100) este estado é alterado. Como exemplo, para v_x entre

$[-49, 0]$ ou $[0, 49]$ as ventoinhas laterais são mantidas desligadas. Caso seja menor do que -49 , a ventoinha direita é ligada e o conjunto se move para a esquerda. Caso seja maior do que 49 , a ventoinha esquerda é ligada e o conjunto se move para a direita. O mesmo vale para v_y , porém para movimentos frontais e traseiros.

4. Trabalhos Futuros

O aplicativo e o serviço estão preparados para movimentos traseiros, porém o protótipo do barco não conta com ventoinha para esse tipo de movimento. A adição desta possibilitará esse tipo de movimentação.

Embora o hardware do barco não esteja preparado para alterar a velocidade das ventoinhas, o serviço remoto e o aplicativo de controle manipulam dados precisos que descrevem a intensidade dos movimentos. No atual estágio o protótipo é capaz apenas alternar o estado das ventoinhas entre ligado e desligado. Adicionar a função de PWM, que é nativa na API do Raspberry Pi para Python, pode trazer um melhor controle nos movimentos. Esse recurso exige a alteração dos transistores utilizados atualmente por outros que suportem a utilização de frequências na base e por isso não foi realizada até o momento.

O cabo de rede que realiza a comunicação entre o Raspberry Pi e o protótipo pode ser removido com a utilização de outro dispositivo de controle, como o Arduino. Outra alternativa para eliminá-lo é utilizar uma bateria de 5V e 1A no próprio barco e fixá-la na placa padrão, porém isso aumenta o peso do conjunto e por isso não foi realizado. É necessária a análise de outros meios de alimentação.

5. Resultados e Conclusão

O protótipo funcionou de forma satisfatória nos testes realizados, sendo capaz de se movimentar por espaços curtos com certa estabilidade. O aplicativo desenvolvido cria uma interface de navegação muito sugestiva e facilita o manuseio do sistema.

De acordo com o apresentado, os objetivos do trabalho puderam ser alcançados de forma satisfatória.

References

- btcroston@googlemail.com (2013). Raspberry-gpio-python. <https://code.google.com/p/raspberry-gpio-python/>. Acessado em 28 de novembro de 2013.
- Foundation, T. R. P. (2013a). About us. <http://www.raspberrypi.org/about>. Acessado em 28 de novembro de 2013.
- Foundation, T. R. P. (2013b). Welcome to raspbian. <http://www.raspbian.org/>. Acessado em 28 de novembro de 2013.
- Google (2013). Adt plugin. <http://developer.android.com/tools/sdk/eclipse-adt.html>. Acessado em 28 de novembro de 2013.