colloquium exactarum

DESENVOLVIMENTO DE UM DISPOSITIVO PARA APOIO AO ENSINO DE COMPUTAÇÃO E ROBÓTICA

DEVELOPMENT OF DEVICE TO SUPPORT THE TEACHING OF COMPUTER SCIENCE AND ROBOTICS

Rafael de Oliveira Maia; Francisco Assis da Silva; Mário Augusto Pazoti; Leandro Luiz de Almeida; Danillo Roberto Pereira

UNOESTE - Universidade do Oeste Paulista, Faculdade de Informática – FIPP, Presidente Prudente – SP, Brasil. E-mail: chico, mario, llalmeida, danilopereira@unoeste.br

RESUMO – Este trabalho propõe o desenvolvimento de um dispositivo alternativo como elemento motivador ao aprendizado de computação e robótica utilizando as placas Raspberry PI e Arduino. São apresentadas as ligações de todo o hardware utilizado para a construção do dispositivo, denominado de Betabot, e também são relatadas as tecnologias utilizadas para a sua programação. Foi desenvolvido um ambiente para a escrita de programas a serem executados no Betabot. Com esse ambiente é possível escrever programas na linguagem de programação Python, utilizando bibliotecas com funções específicas para o dispositivo. Com o Betabot pode-se, por meio do uso de uma webcam e utilizando processamento de imagens, procurar por padrões como rostos, círculos, quadrados e cores. O dispositivo também possui funções para movimentar servos e motores, e capturar valores retornados por alguns tipos de sensores conectados às portas de comunicação. A partir da idealização deste trabalho, foi possível desenvolver um dispositivo fácil de ser manipulado e programado, que pode ser utilizado como apoio ao ensino de computação e robótica.

Palavras-chave: Robótica; Arduino; Raspberry PI; Python.

ABSTRACT – In this work we proposed the development of an alternative device as a motivating element to learn computer science and robotics using the Raspberry PI and Arduino boards. The connections of all hardware used to build the device called Betabot are presented and are also reported the technologies used for programming the Betabot. An environment for writing programs to run at Betabot was developed. With this environment it is possible to write programs in the Python programming language, using libraries with functions specific to the device. With the Betabot using a webcam and through image processing search for patterns like faces, circles, squares and colors. The device also has functions to move servos and motors, and capture values returned by some kinds of sensors connected to communication ports. From this work, it was possible to develop a device that is easy to be manipulated and programmed, which can be used to support the teaching of computer science and robotics.

Keywords: Robotics; Arduino; Raspberry PI; Python.

Recebido em: 03/06/2014
Revisado em: 30/08/2014
Aprovado em: 20/09/2014

1 INTRODUÇÃO

Nos dias atuais a área de tecnologia cresce de forma exponencial demandando mais profissionais qualificados. No entanto, o aprendizado de programação pode ser bastante complexo, chegando a ser desmotivador (DIJKSTRA, 1989). Diante desse fato, pesquisadores buscam metodologias para ensinar computação de uma forma simples e de fácil entendimento. Solomon e Papert (1976) foram alguns dos precursores nesta área. Eles criaram o "Logo", um software livre, no qual pessoas poderiam, a partir das linhas de código que escreviam, movimentar um objeto. Atualmente, existem alguns kits de robótica para serem aplicados na educação para o ensino de computação como, por exemplo, o "LEGO Mindstorms" (LEGO, 2013), usado em alguns projetos de ensino de computação (SILVA, 2009) e também com outras finalidades relacionados à robótica. Entretanto, o LEGO Mindstorms tem um custo muito alto, o que acaba inviabilizando a sua utilização em alguns casos. No Brasil, a empresa PNCA (PNCA, 2013) oferece o "Kit Alfa" que foi criado para ser usado por estudantes, educadores e entusiastas em robótica. O Kit Alfa é fácil de ser utilizado, sendo constituído de peças metálicas, sensores, motores e um módulo de controle programável, mas, entretanto, com capacidade de processamento limitado, o que não possibilita a utilização de dispositivos como, por exemplo, uma webcam, que possibilitaria a construção de robôs que reconhecem objetos, pessoas etc. Para reconhecer, por exemplo pessoas, faz-se necessária a utilização de algoritmos de Visão Computacional, que demanda muito processamento e memória, que não são encontrados nos kits de robótica atuais.

Esse trabalho busca contribuir com as iniciativas já iniciadas por alguns pesquisadores (SOLOMON; PAPERT, 1976; SILVA, 2009) buscando uma metodologia para contribuir no aprendizado de computação e robótica. Para isso, foi idealizado o desenvolvimento de um dispositivo com alto processamento constituído de um módulo programável e de componentes removíveis. O módulo programável foi desenvolvido utilizando-se de placas *open hardware* Raspberry PI (RASPBERRY PI, 2013) e Arduino (ARDUINO, 2013). A placa Raspberry PI, trata-se de um microcomputador com processador baseado em ARM (*Advanced* RISC *Machine*) do tamanho de um cartão de crédito, com entradas e saídas para periféricos e com suporte a um sistema operacional. Rodando uma derivação do Sistema Operacional Linux, denominada de Raspbian, a placa foi fundamental para utilizar de algoritmos de Visão Computacional, o que possibilitou a ampliação dos recursos do dispositivo. Foi utilizado um dos recursos disponíveis na

distribuição do Sistema Operacional que é a linguagem Python (uma linguagem de alto nível utilizada com o intuito de facilitar a escrita de códigos de programas pelo utilizador do dispositivo). A placa Arduino é um micro controlador e foi utilizado para ser o responsável pelos controles de hardware como motores, sensores e servos. Para que as placas Raspberry PI e Arduino possam trabalhar em sincronia, foi utilizada a comunicação serial, que quando o código em Python for executado no Raspberry PI, comandos sejam enviados e recebidos para manusear os sensores conectados ao Arduino.

O fluxo de funcionamento do dispositivo mostrado na Figura 1 representa a ordem dos eventos que acontecem após sua inicialização, sendo o início do processo quando as placas recebem alimentação e começam a funcionar. O sistema operacional do Raspberry PI é carregado, um aplicativo é executado e fornece duas opções de utilização do dispositivo: 1 – Modo PC (para desenvolver programas); 2 – Modo ferramenta (para executar os programas desenvolvidos). Caso seja escolhida a opção 1, o programa bloqueia a conexão entre as placas e carrega totalmente o sistema operacional. Na opção 2, é estabelecida a comunicação entre as placas, quando são buscados os arquivos de programas desenvolvidos (.py) e exibidos na tela para o usuário escolher qual será executado. Após a execução do programa, o ciclo é encerrado e inicia novamente o processo.

As demais seções deste trabalho estão organizadas da seguinte maneira: na Seção 2 são relatados os trabalhos relacionados de Silva (2009), Curcio (2008), Abe, Alberton e Merkle (2010) e Pacheco (2011); na Seção 3 é detalhada a estrutura física do dispositivo; na Seção 4 é apresentada a metodologia de desenvolvimento dos softwares para o dispositivo; a Seção 5 apresenta os experimentos realizados e os resultados obtidos; por fim, a Seção 6 apresenta a conclusão obtida e os trabalhos futuros.

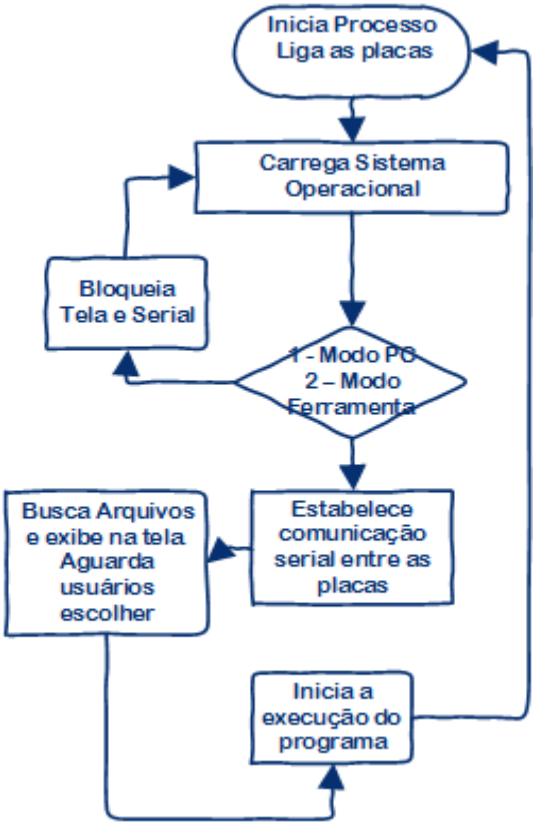

Figura 1. Fluxo de funcionamento do dispositivo.

2 TRABALHOS RELACIONADOS

Nesta Seção são apresentados os trabalhos relacionados a este trabalho proposto, que referem à utilização e desenvolvimento de componentes de robótica no aprendizado de computação e robótica.

No trabalho de Silva (2009) foi proposta uma metodologia para o aprendizado de robótica no ensino fundamental, baseada na teoria sócio histórica de Lev Vygotsky (2004). Essa é uma teoria em que o autor tenta explicar o desenvolvimento da mente humana inspirando-se nas ideias do materialismo dialético, sendo uma das bases da filosófica marxista. Esta metodologia em conjunto com o Kit Lego Mindstorms e um software Educacional desenvolvido pela equipe de robótica educacional do laboratório NatalNet, compõe o sistema de lógica pedagógica proposto, denominado RoboEduc. Foram realizadas atividades de robótica com crianças de 8 a 10 anos. As Atividades visaram produzir conhecimento sobre a construção de protótipos robóticos, sua programação e controle. Com a finalidade de avaliar o desempenho da metodologia, foram projetadas oficinas de robótica pedagógicas. O autor analisou a utilização do conjunto pedagógico (Lego Mindstorms e Software educacional) como elemento mediador do processo de ensino-aprendizagem e as contribuições que o uso da robótica pode trazer para o ensino desde o nível fundamental.

No trabalho de Curcio (2008), foi apresentado um método para a utilização da robótica educacional de baixo custo como ferramenta facilitadora no processo de ensino e aprendizagem, ressaltando a importância da utilização da tecnologia no ensino. O autor afirma que a criação de modelos tridimensionais controláveis por computador com o uso de materiais recicláveis, juntamente com software livre e interface de baixo custo, representa uma atividade que propicia resultados significativos. A prática desse trabalho fundamenta-se na formação contínua de docentes a fim de se efetivar a aplicação da robótica na educação. No decorrer do desenvolvimento do projeto, a autora menciona a importância de se propor a utilização de uma interface que envolva um custo inferior aos praticados no mercado para o ensino da robótica educacional. Os softwares “Imagine” e “SuperLogo” foram utilizados como ambiente de programação. A aplicação do método pelo autor permitiu a identificação de oportunidades de desenvolvimento de projetos multidisciplinares, capacitação continuada dos professores, profissionais envolvidos no projeto e, principalmente, realizar a avaliação da utilização da interface de baixo custo associado ao software livre. Em uma análise

prévia realizada pelo autor, ele demonstra o aumento da participação dos professores e alunos durante o ano letivo.

A contribuição de Abe, Alberton e Merkle (2010) para a área de robótica educacional foi realizar oficinas com apostilas elaboradas pelo grupo PET-CoCE, que foram composta de apresentação dos componentes eletrônicos, conceitos físicos básicos necessários para montar os projetos, bem como ilustração dos circuitos utilizados para cada projeto. As oficinas foram realizadas com alunos do grupo de altas habilidades do instituto de educação do Paraná (IEP-PEP). Os autores propuseram uma oficina de programação baseada em Arduino com o propósito de unir experiência com circuitos eletrônicos, sintaxe e lógica de programação, como variáveis, blocos condicionais, estrutura de repetições e funções. Cada projeto possui um esquema de como montar o circuito e o código para fazê-lo funcionar, sendo que em cada um foram introduzidos novos conceitos de programação, eletrônica e funções das bibliotecas do Arduino (leds, sensores, servos, etc.). No fim da oficina, apenas duas duplas produziram o trabalho final. O restante, 50% da turma inicial, acabou desistindo do curso. Tendo em vista que estes alunos possuíam pouco conhecimento prévio de eletrônica e programação, os autores consideraram que os eles apresentaram bom desempenho e aproveitamento. Além disso, foram percebidas dificuldades na transcrição dos códigos, em especial nos detalhes de sintaxe, comuns para quaisquer iniciantes em programação. De maneira geral, aqueles que concluíram o curso demonstraram compreender os projetos executados.

Por fim, no trabalho de Pacheco (2011) foi enfatizado o uso da robótica educacional no curso de licenciatura em ciências da computação da UFPB, para contribuir com a iniciação do ingressante na computação. O trabalho do autor foi desenvolvido levando em consideração os conceitos de aprendizagem significativa. Para a aplicação da metodologia, foi utilizado o “KitAlfa” da PNCA (2013), composto por uma unidade de controle, sensores analógicos e digitais, e diversas peças metálicas. A linguagem de programação do kit é denominada LEGAL, sendo desenvolvida pelo fabricante para ser utilizada especificamente por esse kit. A linguagem LEGAL é totalmente em português, criada para facilitar o aprendizado de crianças. As tutorias aconteciam a partir da elaboração semanal de exercícios relacionados aos conteúdos vistos em sala. Primeiramente, os alunos tentavam responder os exercícios utilizando a mesma metodologia que o professor utilizava nas aulas. Depois que escreviam seus algoritmos usando lápis e papel, eles transcreviam os códigos para o ambiente de programação da

linguagem LEGAL. Assim os alunos poderiam perceber a diferença em aprender programação utilizando os métodos tradicionais e utilizando a robótica educacional. Foram aplicados questionários de avaliação antes e depois de cada tutorial, nos quais tinham o mesmo grau de dificuldade em questões objetivas e análise de algoritmos. O autor mencionou que inicialmente os resultados não atingiram o que se esperava. Porém, ficou claro que a maior eficiência dessa metodologia proposta pelo autor foi o fator motivacional, pois a robótica conseguiu prender a atenção dos alunos tornando-os mais participativos nas aulas de programação.

3 ESTRUTURA FÍSICA E CONFIGURAÇÃO DO DISPOSITIVO ROBÓTICO

O dispositivo robótico "Betabot", assim nomeado, possui um módulo de controle construído de uma placa Raspberry PI Modelo B, uma placa de Arduino ATMega 2560, um *shield*[1] controlador de motor Ponte H L298n, quatro portas para sensores (ultrassônico, seguidor de linha, temperatura, toque, reconhecimento de cores etc.), duas portas para motores de corrente contínua, duas portas para servos, quatro portas USB, um teclado numérico, um display e uma saída HDMI. A alimentação é realizada por meio de duas baterias de 9V. Na Figura 2 é mostrado o módulo de controle do Betabot com seus componentes conectados (vista frontal). Na Figura 3 é mostrada uma vista de cima do módulo de controle, sem os componentes conectados.

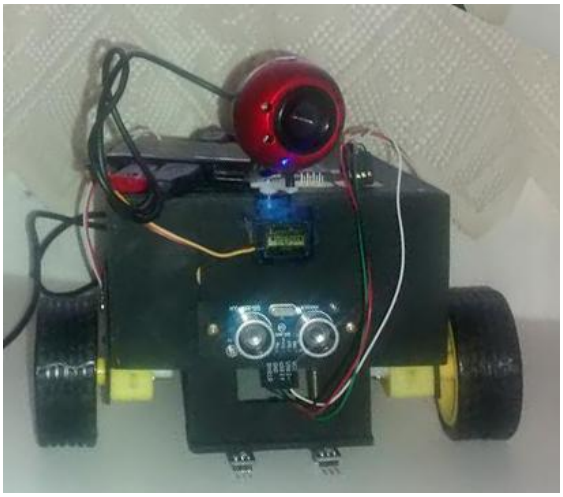

Figura 2. Imagem do módulo de controle do Betabot.



Figura 3. Vista superior do módulo de controle.

[1]*Shields*: são placas que podem ser conectados ao Arduino estendendo as suas capacidades (MCROBERTS, 2011).

Na Figura 3 é apresentada a vista superior do módulo de controle do dispositivo e seus componentes de interação com o usuário, sendo deles: A – portas para utilização de sensores (4 pinos); B – portas para motores de 2 pinos; C – portas para servos (3 pinos); D – Display; E - entradas USB; F – teclado numérico;

A Figura 4 mostra o interior do módulo de controle, em que podem ser vistos a placa RaspBerryPI (A), a placa Arduino (B), o shield controlador de motores Pont H (C), e suas conexões.

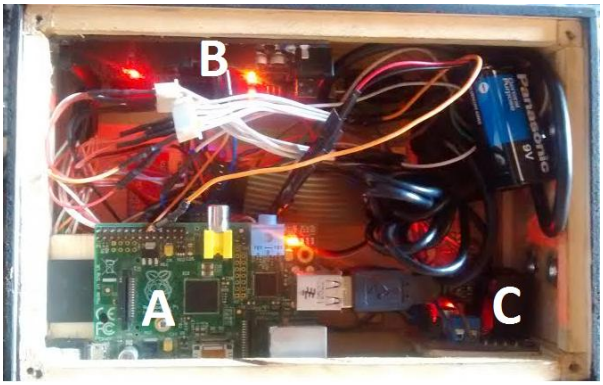

Figura 4. Vista interior do módulo de controle.

3.1 Componentes do dispositivo Betabot

A plataforma Arduino é conhecida pela compatibilidade de vários componentes presentes no mercado. Para a construção do dispositivo foram utilizados os seguintes componentes: motor de corrente contínua com caixa de redução, rodas, servo motor, sensor ultrassônico e sensor de linha.

Para utilização desses componentes foram construídas bases de madeira para afixá-los no módulo controlador do dispositivo, de forma prática e simples. A Figura 5 mostra alguns dos componentes utilizados em conjunto com o módulo controlador: dois motores de corrente contínua com rodas (Figura 5a); sensor de distância (Figura 5b); sensores de linha (Figura 5c).

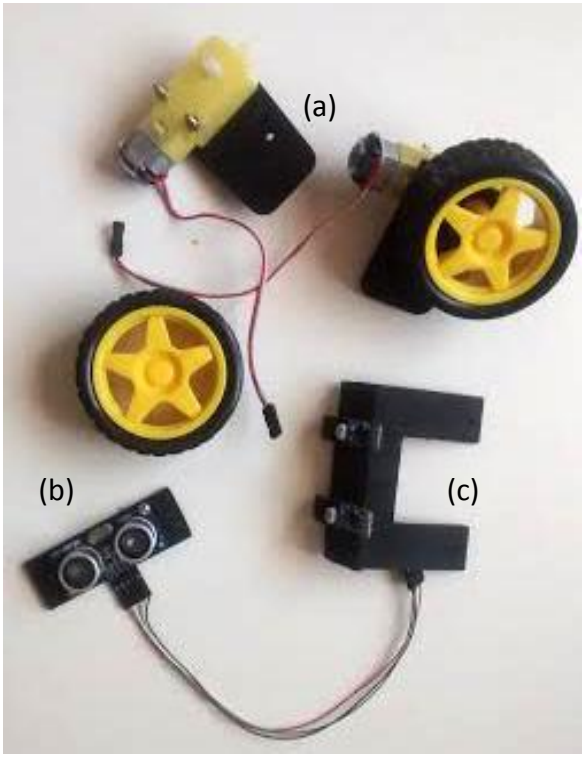

Figura 5. Componentes do Betabot. a) Motores de corrente continua, b) sensor de distância, c) sensor de linha.

A Figura 6 mostra a webcam utilizada para captura de imagens, juntamente com o servo motor na sua base para movimentação horizontal da webcam, e com isso capturar imagens em diferentes posições.

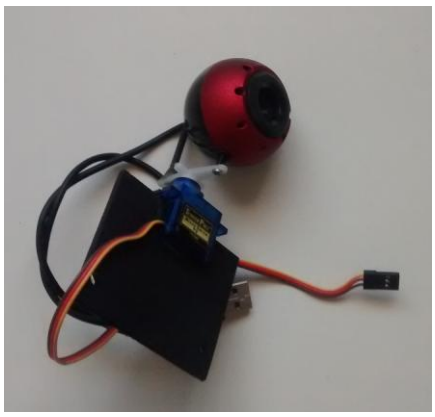

Figura 6. Webcam com servo em sua base.

3.2 Ligação interna das placas

Para o funcionamento do dispositivo Betabot, a placa RaspBerryPI, a placa Arduino e o *shield* controlador de motor precisaram ser conectados uns aos outros. Essa Seção mostra separadamente todas as ligações que foram realizadas no módulo de controle do dispositivo.

3.2.1 Ligação das placas RaspberryPI com a placa Arduino.

O RaspberryPI foi conectado ao Arduino por meio de um cabo USB, em que a comunicação é realizada de forma serial. Nessa comunicação, a informação é enviada um bit de cada vez, sequencialmente, no canal TTY S0 (endereço físico: 0x3f8, equivalente a COM1, no Sistema Operacional Windows) na velocidade de 9600 bits por segundos. A Figura 7 ilustra essa ligação utilizando um cabo USB.

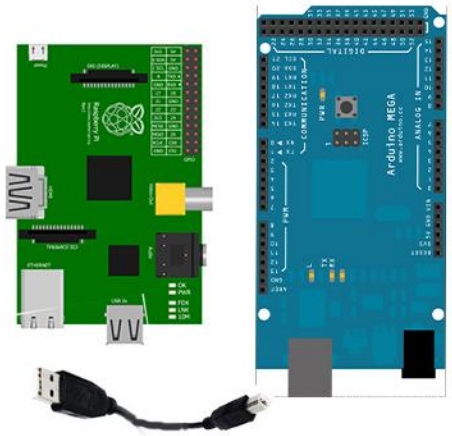

Figura 7. Ilustração da comunicação da placa Arduino com a placa Raspberry PI por meio de um cabo USB.

3.1.2 Ligação da placa Arduino com o controlador de motor Ponte H L298n.

O *shield* Ponte H L298n é um controlador de motor de corrente contínua e foi utilizado para dar mobilidade ao Betabot. O *shield* recebe sinais digitais da placa Arduino para determinar seu funcionamento. Foram utilizadas quatro portas digitais do Arduino (portas 32, 33, 34 e 35) além de duas portas com PWM (*Pulse Width Modulation*) (portas 12 e 13). A Figura 6 mostra o *shield* controlador de motor conectado à uma bateria de 9v e às portas do Arduino.

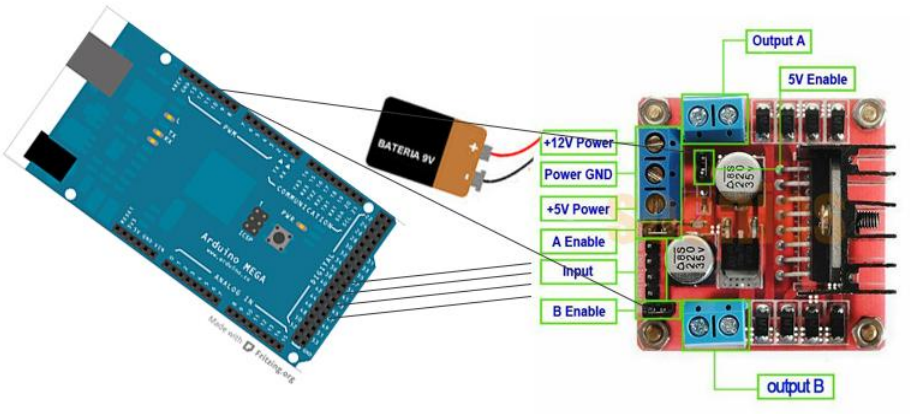

Figura 6. Arduino conectado ao controlador de Motor L298n.

3.2.3 Ligação da placa Arduino com o LCD Nokia 5110

Foi utilizado um LCD Nokia 5110 com uma tela de 84 x 48 pixels iluminada por um *backlight* azul ligado ao Arduino, utilizado para mostrar informações ao usuário como, por exemplo, as opções de escolhas iniciais e os programas a serem executados (Figura 8).

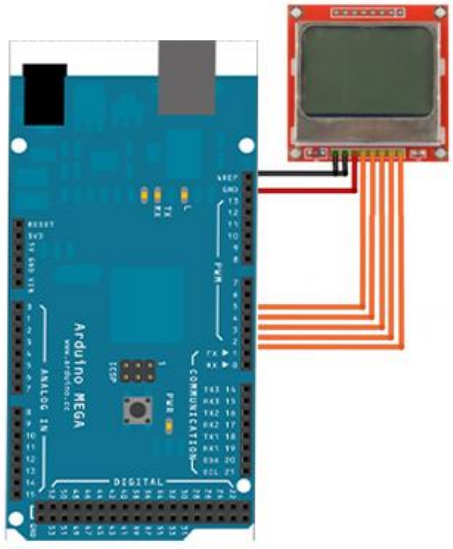

Figura 8. Ligação do Arduino com o LCD Nokia 5110.

Para a ligação do display LCD Nokia ao Arduino, foram utilizadas as portas 2, 3, 4, 5 e 6 do Arduino, além da alimentação regulada com resistores de 470 ohms para evitar danos ao display.

3.2.4 Ligação do Arduino com as Portas Externas

Para facilitar a usabilidade do dispositivo, foram criadas portas externas para utilização dos componentes, sendo elas de três tipos diferentes: de 2 pinos utilizada para a ligação de motores de corrente contínua; de 3 pinos utilizada para a ligação de servos; e de 4 pinos utilizada por sensores que necessitem de alimentação (5v e GND) servindo de entrada e saída, a exemplo de como funciona o sensor ultrassônico. Para a construção das portas externas foram utilizadas placas de circuito perfurado e conectores machos. O esquema de ligação das portas externas com as portas do Arduino pode ser observado na Figura 9.

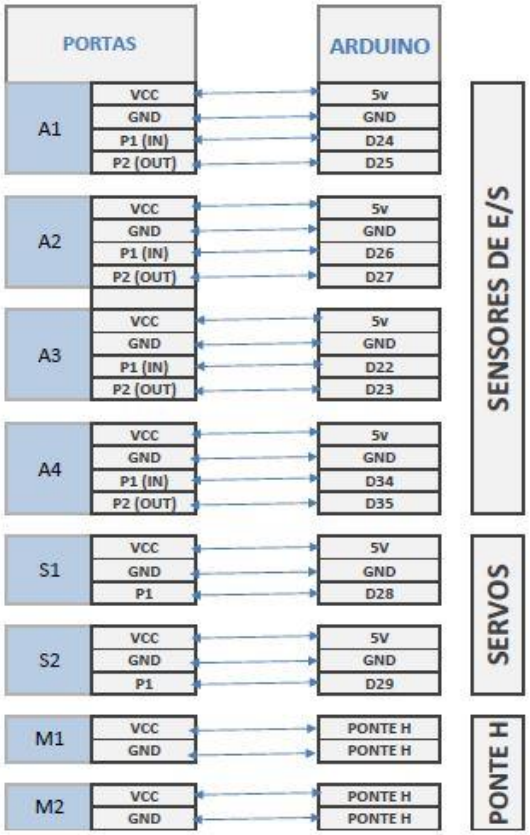

Figura 9. Esquema de ligação das portas externas com as portas do Arduino.

Como pode ser observado na Figura 9, cada porta recebeu uma sigla, em que a sigla: "A" indica portas de 4 pinos, usado pra entradas e saídas; "S" indica portas de 3 pinos usadas para servo motores; "M" indica portas para motores de corrente contínua.

3.2 Configuração Raspberry PI

O Raspberry PI necessita de um sistema operacional instalado para funcionar, este é executado a partir de um cartão SD (*Secure Digital card*). Como a arquitetura do processador do Raspberry PI é baseada em ARM (*Acorn RISC Machine*), foi utilizado o Sistema Operacional Raspbian, que é uma distribuição do Sistema Operacional Linux Debian para o Raspberry PI.

O Sistema Operacional Raspbian por padrão traz a linguagem de programação Python instalada. Para o desenvolvimento do trabalho, foi necessário instalar algumas bibliotecas do Python, dentre elas, a "pyserial", que é uma biblioteca responsável pela comunicação serial com o Arduino. Outra biblioteca que foi utilizada foi a "SimpleCV", que é um *framework*[2] de código aberto para construção de aplicações de Visão Computacional, com acesso às funções do OpenCV de forma simplificada (sem o conhecimento de profundidade de bits, formatos de arquivos, espaço de cores, gerenciamento de *buffer*, etc.). O *framework* possui funções que possibilitou a interatividade do Betabot, quando utilizada a webcam.

A IDE do Arduino e suas bibliotecas (servo, sensor de distância, sensor de linha, driver do display, entre outras) foi instalada no Raspberry PI e configurada para testes e desenvolvimento do Betabot.

[2]*Framework*: representam uma estrutura formada por blocos pré-fabricados de software que os programadores podem usar estender ou adaptar para uma solução específica e linguagens de padrões (FAYAD; SCHMIDT, 1997).

4 DESENVOLVIMENTO DE SOFTWARE PARA O DISPOSITIVO ROBÓTICO

O software do Betabot possibilita a programação por meio de um ambiente desenvolvido para este trabalho, utilizando as funcionalidades do Arduino. Foi escolhida a utilização da linguagem de alto nível Python, por sua natividade no Raspberry PI e também pelo seu alto desempenho, se comparada com outras linguagens de programação quando executadas dentro do ambiente do Raspberry PI. No lado do Arduino, a linguagem utilizada foi o Wiring (Baseada em C), que faz parte do ambiente de desenvolvimento do Arduino.

4.1 Comunicação do Raspberry PI com a placa Arduino.

Para comunicação entre as duas placas, que foram ligadas internamente por meio da porta USB, foi utilizada a tecnologia serial que consiste em enviar uma informação bit a bit de forma sequencial. Para que isso aconteça, as placas precisam estar sincronizadas, ou seja, possuir um mesmo valor na velocidade da transmissão. No lado do Raspberry PI foi utilizada a biblioteca "*pyserial*", que possui várias funções disponíveis para utilização na linguagem Python.

4.2 Programação Arduino

O programa para o Arduino é dividido em duas partes principais: "setup" e "loop". A função "setup" é utilizada para inicialização da placa e do programa. Essa função é executada uma vez quando a placa é inicializada. É informado para o hardware da placa o que será utilizado da placa. A função "loop" gerencia toda execução do programa. O programa escrito dentro da função loop é executado infinitamente, ou seja, ao terminar a execução da última linha desta função, o programa inicia novamente a partir da primeira linha da função "loop" e contínua executar até que a placa seja desligada ou a placa do Arduino seja reiniciada.

O programa escrito no Arduino configura todas as bibliotecas de sensores usados no dispositivo, identificando quais as portas e seus tipos que foram utilizados. Entre elas estão configuradas: "ultrasonic.h", "servo.h", "hokia_5110.h", "keypad.h", entre outras. Em sua função principal é lida a porta serial e capturados os dados até encontrar um "\n", que indica o término de um comando. Por exemplo, o comando "SER21" movimentará o servo, que foi indicado pelas três primeiras posições do comando, a quarta posição indica qual o servo será a movimentado (no caso o "2"), a quinta e última posição indica qual a direção do serviço (no exemplo o estado "1" representa movimentar para a esquerda). Esse comando

pode ser diferente pra cada tipo de sensor, outro exemplo seria para acionar o sensor de distância, em que é enviado o comando "ULT1", sendo o parâmetro "1" a porta na qual ele está conectado.

4.3 Desenvolvimento de software em Python para o Raspberry PI

Na camada de programação no Raspberry PI, foi utilizada a linguagem Python para a comunicação via serial com o Arduino, bem com também para utilização e desenvolvimento de bibliotecas de funções.

Foi escrito um programa que estabelece conexão com o Arduino e envia as informações, configura as portas do dispositivo por meio de objetos com o número e o tipo da porta (usadas na parte experta para conectar sensores, motores, etc.). O programa é dividido em sub-rotinas com nomes de fácil compreensão para serem utilizadas na programação, e por fim, ainda no programa principal do Python, são inicializados variáveis, objetos e funções para serem utilizadas na programação do Betabot.

4.4 Ambiente de programação do Betabot

Foi construído um aplicativo na linguagem Java para servir de ambiente de programação do Betabot. A necessidade de um ambiente assim foi ocasionada pela falta de usabilidade do Python, que possui ambientes pouco interativos e muitos comandos tem de ser escritos diretamente no terminal.

A função do ambiente é abrir um arquivo do Python, propiciar funções para reconhecer padrões (rosto, quadrado, círculo, etc.) e funções para movimentações (motores, servos e sensores). Com isso, ocultar detalhes de programação que somente seriam apenas compreendidos por programadores com um certo nível de experiência. O ambiente possui três botões: o primeiro abre um arquivo de projeto já existente, o segundo tem a função de salvar o arquivo, e o terceiro executa o arquivo com o *shell* do Python. O ambiente não tem como função checar sintaxe, servindo apenas como uma forma de minimizar a quantidade de comandos que seriam necessários serem digitados, direcionando o foco apenas para a programação pertinente aos comandos e funções do dispositivo.

4.5 Funcionalidades do Betabot

Com a utilização do ambiente descrito na Seção 4.4 é possível escrever programas para o Betabot e explorar os recursos disponíveis no dispositivo e também da linguagem de programação Python. Esses recursos funcionam com cada componente conectado ao Betabot, como por exemplo, a câmera (Figura 6) que tem um servo em sua base e é controlada a partir da função "ProcuraRosto()". A partir dessa função é

ativada a captura de imagens e com os recursos disponíveis na SimpleCV para detecção de rostos é informado se o rosto foi encontrado ou não. Com o mesmo intuito de funcionamento, as funções "ProcuraQuadrado()" e "ProcuraCirculo()" também informam se o padrão foi encontrado, no caso dessas funções quadrado e círculo respectivamente. Para controle do servo motor, que movimenta a câmera, pode-se utilizar a função "Olhar (porta, direção)". Para essa função são enviadas por parâmetro, a porta que o servo está conectado e direção (esquerda, direita, centro) que o Betabot deve "olhar", fazendo a câmera apontar para o lado escolhido.

Para a utilização dos motores de corrente contínua foi escrita a função "Andar (direção, tempo)", com o parâmetro direção e o tempo, sendo que a direção indica a trajetória do Betabot e o tempo indica o quanto será executada essa trajetória. Para escrever uma mensagem na tela do Betabot é utilizada a função "EscreverTela (linha, mensagem)", os parâmetros indicam o número da linha que será escrito no display (de 1 a 7) e a mensagem que será escrita nessa linha (até 12 Caracteres).

O sensor de distância que é conectado a uma porta de sensor utiliza a função "VerDistancia(porta)" que retorna um número inteiro que representa a distância do Betabot em centímetros de algum objeto que esteja posicionado à frente do dispositivo.

Para cada novo sensor que seja inserido no Betabot será preciso criar uma função específica para sua utilização.

5 EXPERIMENTOS

Esta Seção apresenta os experimentos usando o Betabot e é dividida em blocos compostos de descrição do funcionamento e o código de programa usando as funções criadas neste trabalho.

No Código 1 é criado um programa que faz o Betabot andar em linha reta enquanto a distância seja maior que 10 centímetros de qualquer objeto. A variável "Dist" recebe o valor capturado no sensor de distância na porta A1 e na repetição (comando "While"), cada comparação faz o dispositivo andar um segundo pra frente.

Código 1: Andar Até Objeto.

```
Inicio()
    Dist = VerDistancia(A1)
    While Dist > 10:
       Andar("frente", 1)
       Dist = verDistancia(A1)
Fim()
```

O Código 2 faz uma repetição infinita, em cada ciclo é capturada a distância e o Betabot anda um (1) segundo para frente. Após isso é feita uma comparação da distância, caso satisfeita, o dispositivo anda dois (2) segundo para a esquerda, mudando sua trajetória.

Código 2: Desviar de Objeto.

```
Inicio()
    Dist = VerDistancia(A1)
    While True:
       Andar("frente", 1)
       Dist = verDistancia(A1)
       If(Dist == 5)
           Andar("esquerda", 2)
Fim()
```

O Código 3 é um programa que movimenta o Betabot para esquerda até encontrar um rosto, para isso, uma variável "Resp" recebe o valor da função de procurar rosto. É verificado para cada lado que o dispositivo "olha" (movimenta a câmera) se existe rosto, caso não seja encontrado, o dispositivo anda 3 segundos para esquerda e continua a repetição.

Código 3: Procurar Rosto.

```
Inicio()
    Resp = ProcuraRosto()
    While !Resp:
       Olhar(S1, "esquerda")
       Resp = ProcuraRosto()
       IF !resp:
            Olhar(S1, "esquerda")
            Resp = ProcuraRosto()
       IF !resp:
            Olhar(S1, "direita")
            Resp = ProcuraRosto()
       IF !resp:
            Olhar(S1, "centro")
            Resp = ProcuraRosto()

       Andar("esquerda", 3)
Fim()
```

6 CONCLUSÃO

Apesar de a robótica ser uma tecnologia pouco utilizada em ambientes educacionais, existem alguns kits motivacionais interessantes que podem ser trabalhados no aprendizado de programação e robótica. Porém, todos eles possuem algum tipo de inviabilidade que acaba dificultando sua utilização em alguns tipos de ambientes, entre elas, estão: o alto custo, baixo processamento, limitação de extensão de componentes, difícil manutenção etc. Com relação às características dos kits motivacionais do mercado, o presente trabalho apresenta uma proposta do desenvolvimento de um dispositivo alternativo que poderá ser utilizado nesse meio.

Os resultados do dispositivo aplicado à motivação da aprendizagem de computação e robótica não poderão ser calculados detalhadamente, a princípio, por se tratar de um processo que exige uma maior capacitação e avaliação da metodologia aplicada.

Com uma análise voltada ao dispositivo, foi possível constatar que sua utilização é um pouco complexa por se tratar de um protótipo feito de forma artesanal, sem uma previa preocupação com os padrões utilizados nos kits educacionais comerciais. Ainda sim, foi possível atingir os objetivos do dispositivo que é iniciar o desenvolvimento de um dispositivo com características favoráveis como mecanismo de apoio ao aprendizado de computação e robótica utilizando de placas open hardware.

Na utilização do dispositivo, um aluno teria o contato direto com a linguagem de

programação Python, podendo trabalhar estruturas computacionais dentro de um dispositivo que vai interagir com a execução de seus códigos. Acredita-se que tudo isso possa favorecer a uma melhor motivação e um melhor entendimento de programação para os alunos.

Como trabalhos futuros a partir do Betabot tem-se o desenvolvimento de uma linguagem de programação específica para o dispositivo, montagem de um módulo capaz de converter todos os tipos portas de sensores, servos e motores em um padrão único usando portas USB (*Universal Serial Bus*), aplicação e integração de ambientes já usados no ensino de programação, como por exemplo, o Scratch, que é uma linguagem de programação interativa, em que nela os programas são desenvolvidos com blocos de montar que indicam comandos.

REFERÊNCIAS

ABE, K. S.; ALBERTON, B. A. V.; MERKLE, L. E. **Utilização da plataforma open-source Arduino como ferramenta de ensino de programação e eletrônica no ensino médio**. In: SCITE - SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA DA UTFPR, 17. 2010.

ARDUINO. Disponível em: <http://arduino.cc>. Acesso em: set. 2013.

CURCIO, C. P. C. **Proposta de método de robótica educacional de baixo custo.** 2008. Dissertação (Mestrado) – Instituto de Tecnologia para o Desenvolvimento – LACTEC, Curitiba, PR.

DIJKSTRA, E. W. On the cruelty of really teaching computing science. **Communication of ACM**, p. 1398-1404, 1989.

FAYAD, M. E.; SCHMIDT, D. C. Object-oriented application frameworks. **Communications of the ACM**, v. 40, p.10, 1997.

LEGO. **Lego mindstorms**. Disponível em: <http://mindstorms.lego.com/en-us/Default.aspx>. Acesso em: out. de 2013.

MCROBERTS, M. **Arduino básico**. São Paulo: Novatec, 2011.

PACHECO, T. R. **Uma experimentação do uso de robótica no ensino da programação**. 2011. Monografia (Graduação) – Universidade Federal da Paraíba.

PNCA. Disponível em: <http://www.pnca.com.br>. Acesso em: set. de 2013.

RASPBERRY PI. Disponível em <http://raspberrypi.org>. Acesso em: set. de 2013.

SILVA, A. F. **RoboEduc:** uma metodologia de aprendizado com robótica educacional. 2009. 127f. Tese (Doutorado) – Universidade Federal do Rio Grande do Norte, Natal, RN.

SOLOMON, C. J.; PAPERT, S. A case study of a young child doing turtle graphics in LOGO. In: THE NATIONAL COMPUTER CONFERENCE AND EXPOSITION. **Proceedings…** New York: ACM, June 7-10, p. 1049-1056, 1976.

VYGOTSKY, L. S. **Psicologia pedagógica.** São Paulo: Martins Fontes, 2004.