UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO

NETALIANNE MITCHELLE FAGUNDES HERINGER

SISTEMA JOANINHA: UMA PLATAFORMA ROBÓTICA PARA FINS EDUCACIONAIS

VITÓRIA – ES
DEZEMBRO/2014

NETALIANNE MITCHELLE FAGUNDES HERINGER

SISTEMA JOANINHA: UMA PLATAFORMA ROBÓTICA PARA FINS EDUCACIONAIS

Parte manuscrita do Projeto de Graduação da aluna **Netalianne Mitchelle Fagundes Heringer**, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Orientador: Prof. Dr. Anselmo Frizera Neto
Coorientador: Eng. Leonardo de Assis Silva

VITÓRIA – ES
DEZEMBRO/2014

NETALIANNE MITCHELLE FAGUNDES HERINGER

SISTEMA JOANINHA: UMA PLATAFORMA ROBÓTICA PARA FINS EDUCACIONAIS

Parte manuscrita do Projeto de Graduação do aluno **Netalianne Mitchelle Fagundes Heringer**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Aprovada em 19, de dezembro de 2014.

COMISSÃO EXAMINADORA:

_____

Prof. Dr. Anselmo Frizera Neto
Universidade Federal do Espírito Santo - UFES
Orientador

_____

Eng. Leonardo de Assis Silva
Coorientador

_____

Prof. Dr. Raquel Frizera Vassallo
Universidade Federal do Espírito Santo - UFES
Examinador

_____

Prof. Dr. André Ferreira
Universidade Federal do Espírito Santo - UFES
Examinador

RESUMO

É crescente o interesse pela utilização de robótica educacional como ferramenta pedagógica nas salas de aulas em algumas escolas. A real extensão do emprego da robótica aplicada à educação no Brasil é dada ao observar que existem diversos estudos científicos já realizados nessa área e o surgimento de empresas nacionais e internacionais na fabricação de produtos exclusivos para crianças com utilização da robótica para fins educacionais.

O intuito deste trabalho é o de apresentar a crianças uma nova forma de estar em contato com a tecnologia: oferecendo a possibilidade de utilizar competências cognitivas, criatividade, manejo do pensamento e exploração da lógica.

O sistema Joaninha foi desenvolvido com o intuito de atingir o público infantil, com faixa etária de 7 a 10 anos. O projeto é composto de um robô (*hardware*) e uma interface para controlá-lo (*software*), que apresenta duas versões: uma para a plataforma *Android* e outra para a plataforma *Windows*. O *software* permite à criança controlar o robô de forma manual, ou de forma programada.

LISTA DE FIGURAS

Figura 1 – Esquemático do Sistema: Visão Geral 9
Figura 2 – Plataforma Robótica da LEGO 12
Figura 3 – Interface do Software RoboEduc 13
Figura 4 – Componentes de um servomotor. 16
Figura 5 – Posicionamento dos componentes na caixa do servomotor. 16
Figura 6 – Malha de controle típica de um servomotor 17
Figura 7 – Esquemático de um CI para controle de servomotores 17
Figura 8 – Circuito para ilustração da variação de valor de tensão médio por chaveamento. 18
Figura 9 – Forma de onda resultante do chaveamento do circuito 19
Figura 10 – Formas de onda da tensão de saída com frequência de chaveamento fixa 20
Figura 11 – Sinais de controle de um servomotor 20
Figura 12 – Esquemático de uma máquina DC. 21
Figura 13 – Imã permanente, servomotor 22
Figura 14 – Arquitetura do *Arduino* 23
Figura 15 – Arduino UNO 23
Figura 16 – Estrutura física do robô 25
Figura 17 – Esquemático de ligação do módulo *Bluetooth* ao *Arduino* 26
Figura 18 – Esquemático de ligação de todos os componentes do protótipo 27
Figura 19 – Foto do robô e seus componentes 28
Figura 20 – Detalhes das placas de circuito montadas para atender ao projeto 28
Figura 21 – Janela principal da interface com o usuário para computador 29
Figura 22 – Interface com o usuário: adição de blocos à linha de programação. 31
Figura 23 – Interface com o usuário: detalhe da janela para configuração de velocidade e tempo 31
Figura 24 – Interface com o usuário: detalhe de outros elementos necessários ao funcionamento do programa 32
Figura 25 – Utilização da interface gráfica para celular 33
Figura 26 – Interface do AppInventor utilizado para programação do software para celular 33
Figura 27 – Interface do programa desenvolvido para o celular 34
Figura 28 – Exemplo da divisão em blocos da palavra recebida 35
Figura 29 – Exemplo da interpretação no protocolo dos diferentes modos de operação. 35
Figura 30 – Leitura do protocolo para os três blocos dados como exemplo 36

Figura 31 – Fluxograma ilustrando o funcionamento da subrotina Leitura de comandos 38
Figura 32 – Fluxograma ilustrando o funcionamento da subrotina Modo Programado 39
Figura 33 – Fluxograma ilustrando o funcionamento da subrotina Modo Manual 40
Figura 34 – Sequência de comandos gerada para experimentação do sistema. 43
Figura 35 – Experimento realizado: movimento do robô retílínio, sentido progressivo. 44
Figura 36 – Experimento realizado: movimento do robô retilínio, sentido retrógrado. 45
Figura 37 – Quadros demonstrando as direções executadas na execução da tarefa. 46
Figura 38 – Trajetória percorrida para a tarefa executada 47
Figura 39 – Experimentos realizados com usuários do sistema com o aplicativo do celular e com o computador. 48

LISTA DE TABELAS

Tabela 1 – Informações técnicas do servo motor SM-S4303R 25
Tabela 2 – Informações técnicas do módulo *Bluetooth* utilizado EGBT- 046S..... 26
Tabela 3 – Ilustração das tarefas possíveis a serem programadas 30
Tabela 4 – Informações referentes aos caracteres de tarefa do protocolo 36
Tabela 5 – Informações da tarefa referentes a cada caractere do protocolo 37
Tabela 6 – Resultados do experimento para movimento retilíneo, sentido progressivo 44
Tabela 7 – Resultados do experimento para movimento retilíneo, sentido retrógrado 45
Tabela 8 – Resultados do experimento para tarefa programada. 47
Tabela 9 – Características de cada usuário do experimento realizado 48
Tabela 10 – Resultados experimento com o aplicativo para celular. 49
Tabela 11 – Tempo de execução da tarefa por cada usuário do sistema utilizando o aplicativo para celular. 49
Tabela 12 – Resultados experimento com o computador em modo programado..... 50
Tabela 13 – Tempo de execução da tarefa por cada usuário do sistema utilizando o computador, modo programado. 51

SUMÁRIO

1 INTRODUÇÃO 8
1.1 Apresentação 8
1.2 Motivação 9
1.3 Revisão Bibliográfica 11
1.4 Objetivo geral e objetivos específicos 14
2 ESTRUTURA E PRINCÍPIOS DE FUNCIONAMENTO DO SISTEMA 15
2.1 Servomotor 15
2.2 Controle do servomotor 17
2.3 Motor de corrente contínua 21
2.4 A plataforma de prototipagem *Arduino* 22
3 O SISTEMA JOANINHA 24
3.1 A estrutura física do robô 25
3.2 A interface com o usuário 29
3.2.1 Interface para plataforma *Windows* 29
3.2.2 Interface para plataforma *Android* 32
3.3 O Protocolo de comunicação entre o *Software* e o *Hardware* 34
3.4 O Controle realizado no *Arduino* 37
3.4.1 Leitura de Comandos 38
3.4.2 Acionamento dos motores 41
4 VALIDAÇÃO EXPERIMENTAL, RESULTADOS E DISCUSSÕES 42
4.1 Protocolo experimental 42
4.2 Resultados e discussão do experimento 1 44
4.3 Resultados e discussão do experimento 2 46
4.4 Resultados e discussão do experimento 3 48
4.4.1 Experimento utilizando o Aplicativo para celular: Modo Manual 49
4.4.2 Experimento utilizando a interface pelo computador: Modo Programado. 49
4.4.3 Avaliação do sistema pelos participantes do experimento 51
5 CONCLUSÃO E TRABALHOS FUTUROS 53
REFERÊNCIAS 55

1 INTRODUÇÃO

1.1 Apresentação

A tecnologia e sua emergência nos meios de comunicação, nos processos industriais, na medicina e até mesmo nos relacionamentos pessoais é fator responsável pela modificação de comportamentos e formas de trabalho, tornando-se cada vez mais influente na vida humana.

Desde a popularização do computador e do advento da *Web* já se discute os efeitos da tecnologia no comportamento humano, nas interações sociais e principalmente na educação infantil. Segundo Delors (1998), é a busca por tecnologia que acelera o processo de desmaterialização do trabalho, acentuando na educação a necessidade de aptidões intelectuais e cognitivas. Assim, novas abordagens de uso da informática na educação, em sintonia com as novas demandas da sociedade como a robótica, foram e estão sendo pesquisadas.

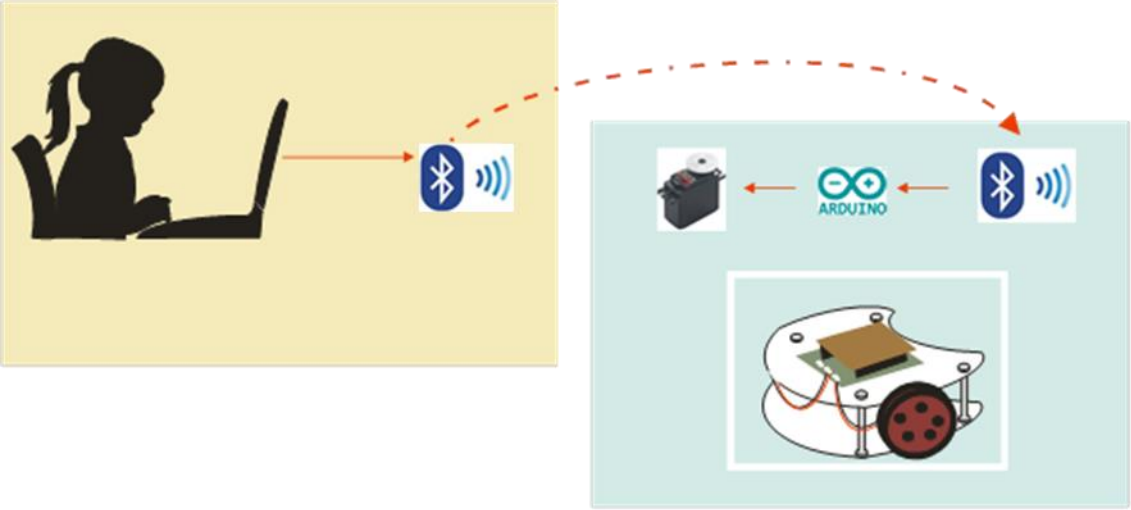
É de fato possível verificar um crescente interesse pela utilização de robótica educacional como ferramenta pedagógica nas salas de aulas em algumas escolas. A real extensão do emprego da robótica aplicada à educação no Brasil é dada ao observar que existem diversos estudos científicos já realizados nessa área (SILVA, 2006; ALVES et al. 2011) e o surgimento de empresas nacionais e internacionais na fabricação de produtos exclusivos para crianças com utilização da robótica para fins educacionais (LEGO, 2014).

Entretanto, os altos custos e a complexidade na interface com o usuário, a ausência de professores capacitados e principalmente a falta de uma política bem definida para a inclusão digital nas escolas, tornaram os *kits* de robótica educacional disponíveis em mercado de difícil acesso para toda sociedade.

É nesse contexto que o trabalho em questão se propõe a apresentar uma ferramenta para fins educacionais que aborda a programação de um robô por crianças, de tal maneira que o *kit* desenvolvido apresente baixo custo e interação mais fácil e intuitiva com o usuário, tornando-se acessível a idades pós-alfabetização, incluindo idades inferiores a 10 anos.

O esquemático da Figura 1 apresenta uma visão geral do trabalho desenvolvido e a interação do usuário com o protótipo.

Figura 1 – Esquemático do Sistema: Visão Geral



Fonte: Produção do próprio autor.

A criança interage com a plataforma gráfica desenvolvida no computador, programando o movimento que ela quer dar ao robô por meio de blocos. A comunicação serial, utilizando *Bluetooth*, é responsável por transmitir ao robô as informações geradas pelo usuário, por meio de um protocolo desenvolvido. O protocolo é lido e executado no Arduino que embarca o sistema do robô, gerando mobilidade e executando a aplicação programada inicialmente.

Cada etapa de construção e todos os componentes utilizados serão vistos com maior detalhe ao longo deste trabalho.

1.2 Motivação

Jacques Delors (1998) apresenta em seu trabalho uma visão da educação prevendo o pleno desenvolvimento humano, não somente considerando o crescimento econômico da sociedade. A plenitude desta ideia contempla aspectos biológicos, pessoais, cognitivos, subjetivos e também sociais: cultura, economia, política. Aprender então está não somente aliado ao fato de ter uma profissão, de conhecer sobre o mundo e sobre a sociedade, mas a educação faz parte da definição do ser humano que será formado, de sua capacidade de interação com o mundo.

Diante dos recursos dispostos atualmente, uma criança ao ser introduzida à educação já se depara com a completa interação entre tecnologia e ser humano. A tecnologia digital envolve

hoje um paradoxo entre a coexistência de alta complexidade tecnológica e amplo acesso da população aos meios eletrônicos digitais. Computadores, tablets, notebooks, smartphones, fazem parte da vida e da rotina na sociedade mostrando que o manejo de tecnologia deixou de ser privilégio de técnicos e profissionais da área, se tornando meio necessário à vivência no mundo atual.

Todos estes fatores repercutem na cultura educacional e, de fato, conclui-se que o contato precoce com a tecnologia já causa impacto nas novas gerações, colocando seus membros, mesmo na infância como consumidores de tecnologia. A possibilidade de ampliar a visão da nova geração não somente como consumidor, mas criador e desenvolvedor de tecnologia é uma ideia que tem aberto as portas dos estudos para educação infantil.

De acordo com Piaget (1976), o conhecimento é construído na medida em que o educando age sobre o objeto de conhecimento e sofre uma ação deste objeto. Papert (1986) utilizou os resultados de Piaget para repensar a educação, assumindo que o conhecimento é ativamente construído pelas pessoas, propondo que educar consiste em criar situações para que os aprendizes se engajem em atividades que alimentem este processo construtivo.

Considerando então o cenário apresentado, o objeto de estudo do trabalho se justifica por ser uma ferramenta com o intuito de apresentar a crianças uma nova forma de estar em contato com a tecnologia: como criador, revertendo seu papel de consumidor, estando engajados na tarefa de desenvolver robótica. Ainda apresenta a possibilidade de utilização de competências cognitivas, criatividade, manejo do pensamento e exploração da lógica, aspectos abordados por Delors em seu trabalho como essenciais para a formação do ser humano e necessários à educação.

De tal maneira, a motivação do trabalho está em apresentar de forma direta ao usuário o interesse por tecnologia, investigação, raciocínio lógico e materialização do conhecimento. De forma indireta, desenvolver características cognitivas e contribuir para uma nova visão de educação que faz parte da formação do indivíduo e de sua capacidade de interação.

O trabalho se justifica em tornar o protótipo desenvolvido acessível à comunidade apresentando baixo custo de aquisição e uma melhor interação do software programado com o

usuário, atendendo faixas etárias menores do que as que os produtos existentes no mercado se propõem a atender.

1.3 Revisão Bibliográfica

São poucos os registros da história da robótica aplicada à educação. Segundo Castilho (2006), entretanto, é notável o crescimento da utilização de softwares educacionais e equipamentos envolvendo robôs nas escolas e nos setores educacionais.

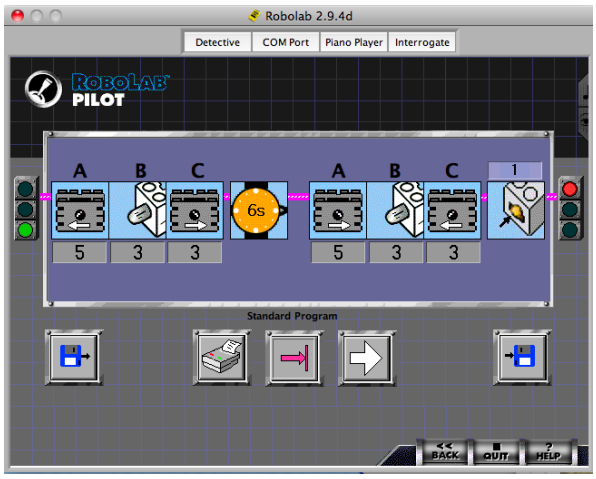
O primeiro conceito de tecnologia e computação (ainda antes do surgimento da *Web*) inserido na educação foi introduzido por Papert no final da década de 70, concebendo a LOGO (PAPERT, 1985), uma linguagem textual de programação aplicada a um ambiente de ensino de matemática e lógica, a qual serve para a comunicação do homem com o computador. Em seu trabalho, Silva (2006) aponta que a linguagem LOGO abriu um espaço de criação com capacidade de simular comandos, imagens e formas de maneira acessível ao público infantil.

Tempos depois, o Instituto de Tecnologia de Massachusetts (MIT) realiza a junção da linguagem LOGO com os brinquedos da LEGO trazendo a ideia de brinquedos montados pela própria criança (engrenagens, sensores, peças de montagem) e que ao conectar ao computador é possível programar em LOGO e proporcionar movimento e comportamento aos robôs montados.

A tecnologia foi aperfeiçoada e em parceria com a National Instruments foi desenvolvido o software RoboLab (ROBOLAB, 2014) que faz parte do *kit* de robótica da Lego (LEGO, 2014) lançado em 1998 denominado Mindstorm. O último modelo lançado pela LEGO foi o Mindstorm EV3 mostrado na Figura 2.

Figura 2 – Plataforma Robótica da LEGO

(a)



(b)

Fonte: MINDSTORMS, 2014.
Legenda: (a) Modelo Mindstorm EV3 da LEGO.
(b) Software Robolab de programação da LEGO.

O *kit* citado acima apresenta a característica de abordar tanto a montagem física do robô, com a possibilidade de escolha dos equipamentos utilizados (motores, sensores, dispositivos de comunicação), quanto a programação do próprio robô montado, tornando o modelo versátil e podendo ser desligado do computador após a programação por parte do usuário. Entretanto, o valor médio de um Lego Mindstorm EV3 é significativamente alto, tornando-se inviável o acesso por muitas escolas e redes de educação no país. Além disso, a complexidade da programação e montagem do robô restringe a faixa etária para usuários acima de 10 anos. Segundo Tavares, Antunes e Gonçalves (2004, p. 80),

> [...] devido a sua complexidade e seu potencial muito alto [Lego MindStorms] para a faixa etária a qual foi destinado e também ao aparecimento ou desenvolvimento de várias ferramentas de software para controle em nível um pouco mais baixo (usando linguagem 'C' ao invés da linguagem visual provida pelo fabricante), os *kits* passaram a ser mais usados nos ambientes de pesquisa e no ensino superior.

Foi diante da complexidade da interface do RoboLab que um grupo de pesquisa no departamento de Engenharia da Computação e Automação da Universidade Federal do Rio Grande do Norte (SILVA, 2006) desenvolveu um *software* denominado RoboEduc a fim de

obter uma plataforma educacional mais acessível e de fácil configuração e utilização pelo usuário. A Figura 3 demonstra a interface do *software* RoboEduc.

Figura 3 – Interface do Software RoboEduc

Fonte: SILVA, 2006.

O *software* apresenta 5 níveis de dificuldade diferentes que podem ser escolhidos pelo usuário, destinando-se a alunos desde a educação infantil até estudantes universitários. Entretanto, o modelo faz uso de hardwares de robótica já existentes permanecendo alto o custo de um conjunto completo para uso educacional.

Em contrapartida, um grupo de pesquisa da UNESP (Universidade Estadual Paulista), em Bauru foi responsável por uma iniciativa que permitiu a criação de robôs com baixo custo, utilizados para auxiliar a educação de computação para alunos universitários (ALVES et al, 2011), e que também ganhou aplicações na robótica de reabilitação, cujos testes preliminares dos robôs, realizados em parceria com a UFES (Universidade Federal do Espírito Santo), envolveram crianças paraplégicas, tetraplégicas e com alguns tipos de distrofia e deficiência cognitiva.

Os trabalhos citados demonstram que o campo de pesquisa com robótica na área de educação apresenta muitas iniciativas, comprovando sua importância. No entanto, ainda existe muito a se avançar e, por isso, a principal relevância desta pesquisa está em conseguir associar em um mesmo trabalho baixo custo no desenvolvimento do *hardware* com um *software* de fácil

compreensão, características que não foram abordadas em conjunto nos estudos aqui mencionados.

1.4 Objetivo geral e objetivos específicos

O objetivo geral deste projeto é desenvolver uma plataforma robótica (software e hardware), cuja interface e acesso sejam de fácil entendimento e baixo custo, destinado a crianças para fins educacionais.

Os objetivos específicos deste trabalho são:

- Realização de análise da bibliografia sobre a utilização de tecnologia na educação e levantar os mais recentes trabalhos na área, seus avanços e suas necessidades de melhoria;
- Avaliação dos melhores equipamentos a serem utilizados na confecção do protótipo visando baixo custo e acessibilidade;
- Montagem física do protótipo e acionamento do robô: atuação dos motores, desenvolvimento do controle de velocidade e do sistema de comunicação;
- Desenvolvimento de um protocolo de comunicação robô-PC para entendimento e execução de tarefas no robô, oriundas da interface com o usuário no computador;
- Programação da interface gráfica e desenvolvimento da forma de programação a ser utilizada pelo usuário, sendo esta de fácil entendimento e intuitiva.

2 ESTRUTURA E PRINCÍPIOS DE FUNCIONAMENTO DO SISTEMA

A área de estudos, pesquisa e desenvolvimento da robótica móvel está em crescente evolução se apresentando em várias aplicações e ganhando cada vez mais espaço em diversos ambientes. Alguns exemplos mais recentes da utilização da robótica estão em aplicações industriais (transporte automatizado e veículos de carga autônomos), urbanas (transporte público, cadeiras de rodas robotizadas), militares (sistemas de monitoramento aéreo remoto - VANTs, sistemas táticos e de combate) e de segurança e defesa civil e militar (controle e patrulhamento de ambientes, resgate e exploração em ambientes hostis) (WOLF, 2009).

Robôs móveis são máquinas construídas sobre estruturas não fixas, que devem apresentar um sistema de locomoção que lhe garanta mobilidade podendo conter rodas, esteiras, pernas, propulsores, asas ou pás rotativa. Quando apresentam um conjunto de sensores e uma programação lhe conferindo percepção do ambiente para navegação, os robôs móveis são classificados como autônomos. Quando a navegação é realizada por um operador remotamente, o robô é classificado como teleoperados. (SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011).

Para a realização completa de sua locomoção e navegação, um robô móvel necessita de dispositivos de hardware embarcados, configurados conforme a função e tarefa que objetivam seu funcionamento. Os principais dispositivos de *hardware* de um robô são seus sensores e atuadores podendo apresentar diversos tipos e configurações (DUDEK, 2000). Neste capítulo será realizada a abordagem teórica de alguns dispositivos de hardware que foram necessários para a realização deste trabalho.

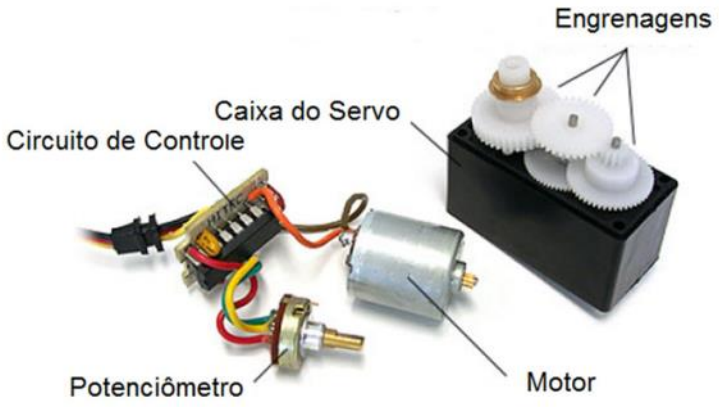
2.1 Servomotor

Servomotores são motores de corrente contínua (DC) que incorporam alguns dispositivos a fim de tornar seu uso mais fácil. O seu estator é do tipo imã permanente, utilizando de imãs de terras raras, que possuem a adição de elementos da série dos lantanídeos para potencializar seu desempenho. Sua principal característica como atuador na robótica consiste em realizar movimento de forma precisa e controlada. A partir de um sinal elétrico em sua entrada, o

servo motor posiciona seu eixo em uma determinada posição angular, controlado por seu próprio sistema.

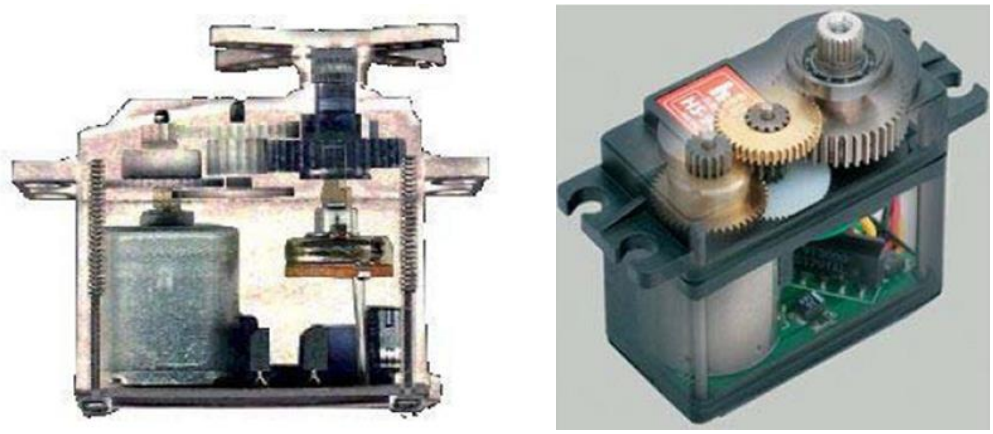
A Figura 4 apresenta os componentes internos de um servo motor. Cada um de seus componentes é posicionado na Caixa do Servo, como mostra a Figura 5.

Figura 4 – Componentes de um servomotor.



Fonte: EPUSP,2014

Figura 5 – Posicionamento dos componentes na caixa do servomotor.



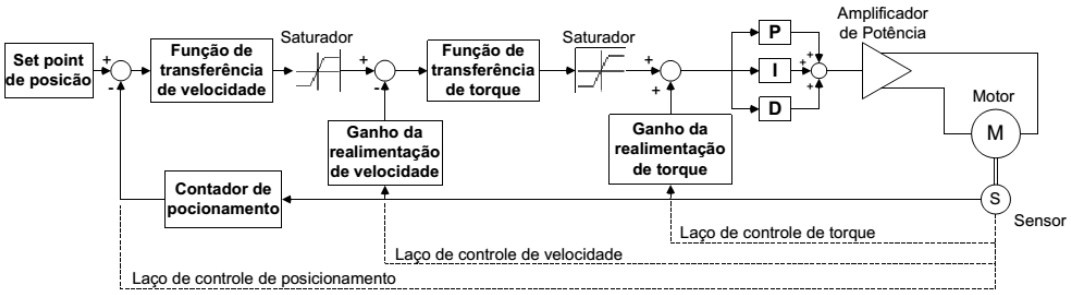
Fonte: SANTOS, 2007.

O potenciômetro é conectado ao eixo de saída do servo motor, monitorando a posição do mesmo, por meio do circuito de controle. As engrenagens reduzem a rotação do motor transferindo um torque maior ao eixo principal de saída, sendo estas acionadas pelo motor de corrente contínua.

2.2 Controle do servomotor

Os servomotores podem ser controlados tendo como referência torque, velocidade ou posição, conforme a Figura 6:

Figura 6 – Malha de controle típica de um servomotor

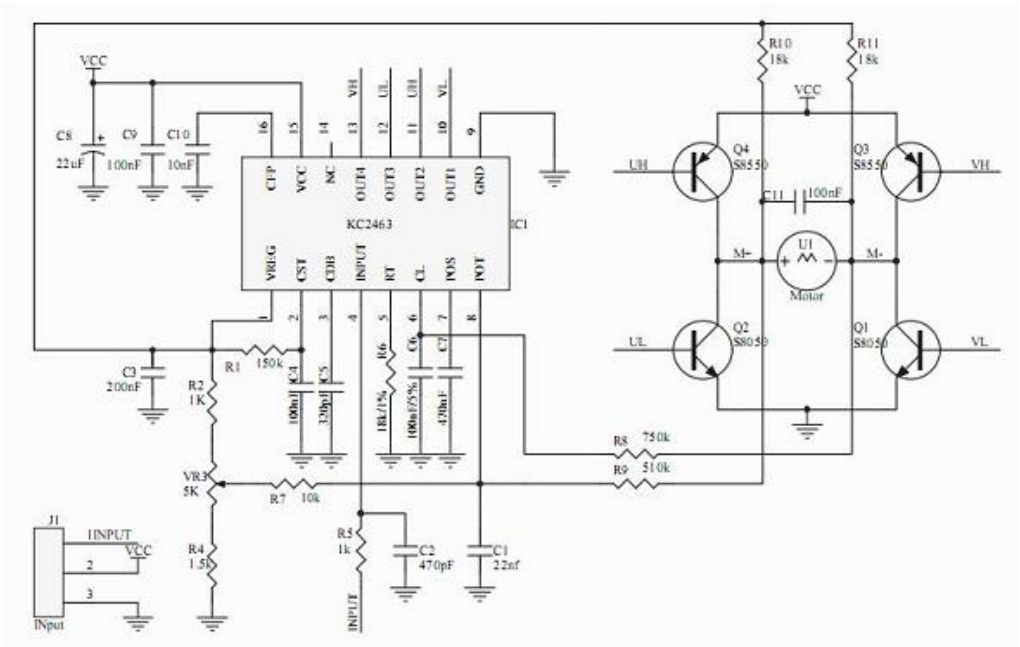


Fonte: ZUCATELLI; OLIVEIRA, 2007.

O controle aplicado apresenta um regime cascata: o parâmetro de posição gera uma referência para a velocidade que por sua vez gera a referência do torque. O fato de cada bloco ser realimentado torna o controle fino e refinado.

A Figura 7 apresenta o esquemático de um CI utilizado para controle de servomotores, e o acionamento do motor utilizando a ponte H formada por IGBT's (*Insulated Gate Bipolar Transistor*).

Figura 7 – Esquemático de um CI para controle de servomotores

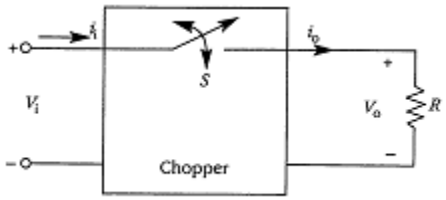


Fonte: CHENG TECHNOLOGY CO LTD, 20--.

É por meio do acionamento de transistores IGBT's (*Insulated Gate Bipolar Transistor*), que o sinal é enviado ao servomotor. O chaveamento dos IGBT's é comandado por modulação de largura de pulso (PWM).

A Modulação de Largura de Pulso (PWM) é uma tecnologia de controle de potência pela variação do valor médio de potência que é fornecida à carga. Para melhor entendimento da modulação, pode-se partir de um circuito imaginário como na Figura 8, considerando S uma chave ideal, de ação muito rápida e uma carga conectada ao circuito representada pela resistência R.

Figura 8 – Circuito para ilustração da variação de valor de tensão médio por chaveamento

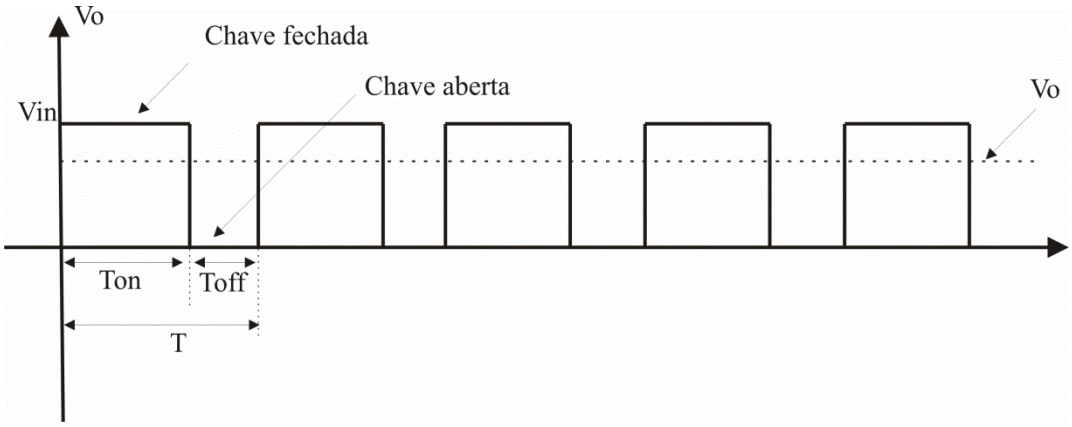


Fonte: AHMED, 2000.

Quando a chave S está aberta, não há circulação de corrente na carga e a potência recebida por ela é nula. Ao fechar a chave, a potência sobre a carga é a máxima fornecida pelo circuito, uma vez que a carga recebe a tensão total aplicada pela fonte.

Supondo agora que a tensão de saída seja ajustável em uma determinada faixa, de zero ao nível de entrada da seguinte maneira: a chave "s" opera de tal modo que ela permaneça fechada (circuito ligado) por um tempo Ton e desligada por um tempo Toff (circuito desligado) em cada ciclo de um período T predeterminado. A forma de onda resultante é mostrada na Figura 9:

Figura 9 – Forma de onda resultante do chaveamento do circuito



Fonte: AHMED, 2000 (modificado pelo autor).

Onde:

Vo= tensão média de saída

Vi= tensão de entrada

Ton= tempo de permanência da chave fechada

Toff= tempo de permanência da chave desligada

O valor da tensão média sobre a carga poderá ser determinado pelas equações (1) e (2):

$$Vo = \frac{Ton}{Ton + Toff} Vin \tag{1}$$

$$Vo = \frac{Ton}{T} Vin \tag{2}$$

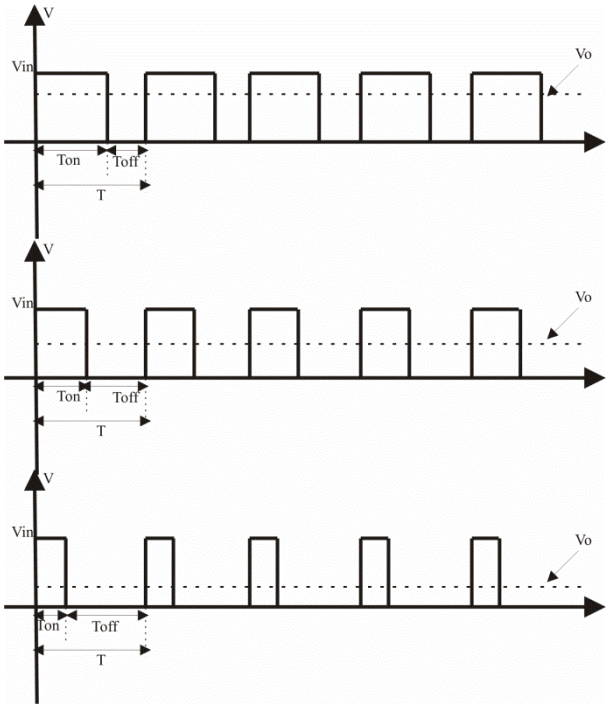
Desta maneira, utiliza-se do conceito de “ciclo de trabalho” que determina o valor médio de tensão sobre a carga e desta forma a potência fornecida pelo circuito à carga. O “ciclo de trabalho” pode ser obtido por meio da equação (3).

$$d = \frac{Ton}{T} \tag{3}$$

Onde d é o ciclo de trabalho.

No método PWM, a largura do pulso Ton varia enquanto o período de chaveamento total T é constante. A Figura 10 demonstra a variação da forma de onda de saída à medida que o ciclo de trabalho aumenta.

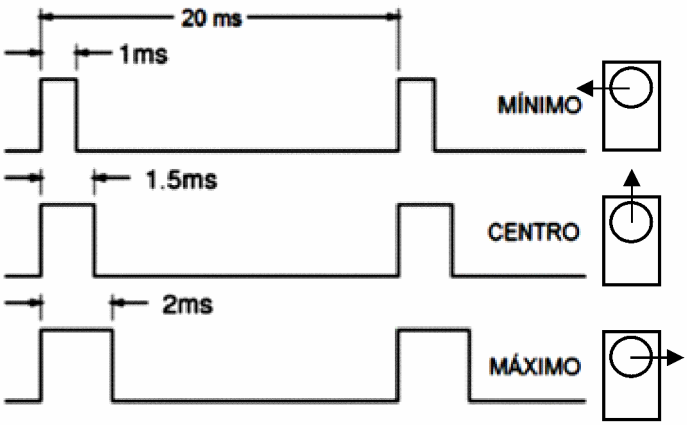
Figura 10 – Formas de onda da tensão de saída com frequência de chaveamento fixa



Fonte: AHMED, 2000 (modificado pelo autor).

Em servomotores o acionamento por PWM se dá em um sinal com período de 50 Hz e pulsos com largura de 1 até 2 ms. Para o caso de servos com rotação 180°, variando-se a largura do pulso a posição pode ser variada de 0 até 180° como mostrado na Figura 11, para o caso de servos com rotação contínua, o controle é realizado sobre a velocidade de rotação.

Figura 11 – Sinais de controle de um servomotor



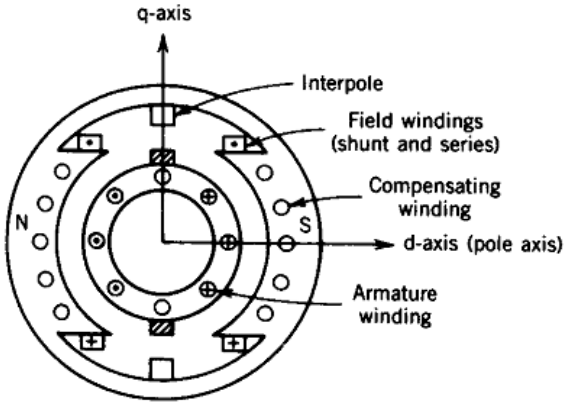
Fonte: EPUSP, 2014 (modificado pelo autor).

2.3 Motor de corrente contínua

As máquinas de corrente contínua são versáteis e extensivamente utilizadas na indústria. Uma grande variedade das características tensão-corrente ou torque-velocidade podem ser obtidas por meio de várias conexões do enrolamento de campo (SEN, 1997).

Em termos construtivos, as principais características das máquinas DC podem ser vistas na Figura 12. O enrolamento de armadura é localizado no rotor e o enrolamento de campo no estator. O estator de pólos salientes é excitado por mais de um enrolamento podendo ser utilizado o enrolamento shunt ou enrolamento série. A distribuição do fluxo criado pelos enrolamentos de campo no entreferro é simétrica em relação à linha central de polos salientes. Esse eixo é denominado de eixo de campo ou eixo direto.

Figura 12 – Esquemático de uma máquina DC.



Fonte: SEN, 1997

A tensão induzida nas bobinas do enrolamento de campo é alternada. Esta tensão é então convertida em DC nos terminais externos da armadura onde está localizado um comutator rotativo e de escovas estacionárias. A combinação escovas e comutador formam um retificador mecânico, resultando na tensão DC de armadura e em uma onda de força magneto motriz de armadura que está fixa no espaço.

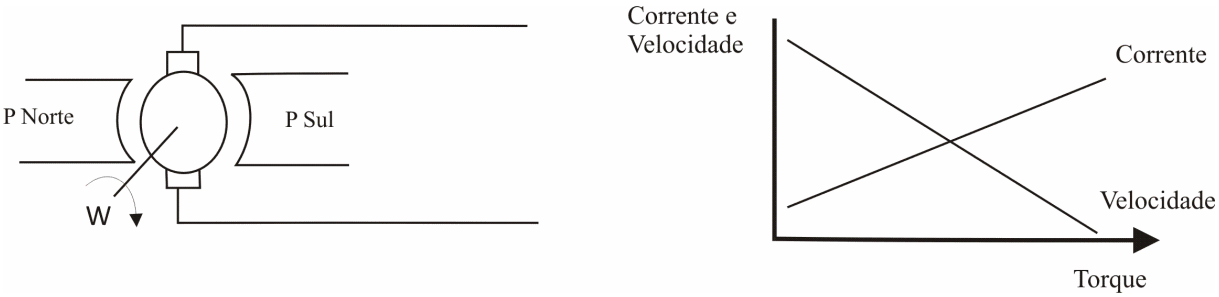
Existem diversos tipos de máquinas DC, cada uma delas apresentando características únicas. Um dos tipos é a máquina DC de imã permanente, mais utilizada para aplicações de baixa potência, como no emprego em servomotores. Nesta máquina, o enrolamento de campo é substituído por um imã permanente, simplificando sua construção. Esta simplificação resulta

na diminuição do espaço físico requerido pelo equipamento e um custo inferior às outras máquinas DC.

Os imãs permanentes não necessitam de excitação externa, nem dissipam potência suficiente para criar campos magnéticos. Por outro lado, existem certas limitações no uso dessa máquina, uma delas é a baixa intensidade de fluxo magnético que os imãs permanentes conseguem produzir.

Desta forma, a diferença de um servomotor para outro motor de corrente contínua é o fato deste não possuir uma bobina para gerar o campo, e sim um ímã permanente. O campo magnético é constante para qualquer módulo da corrente de armadura, apresentando uma resposta linear, como mostrado na Figura 13.

Figura 13 – Imã permanente, servomotor



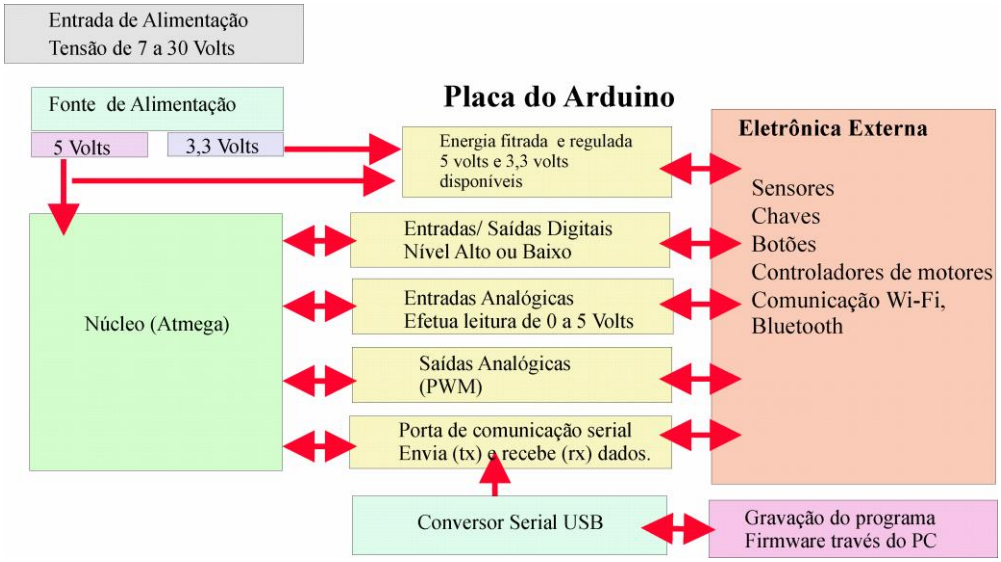
Fonte: ZUCATELLI; OLIVEIRA, 2007 (modificado pelo autor).

2.4 A plataforma de prototipagem *Arduino*

O *Arduino* é uma plataforma de prototipagem eletrônica. Um projeto totalmente aberto que apresenta hardware e software flexíveis e de fácil utilização, criado com o objetivo de permitir o desenvolvimento de controle de sistemas interativos, de baixo custo e mais acessível.

Na Figura 14, é possível observar a arquitetura do arduíno.

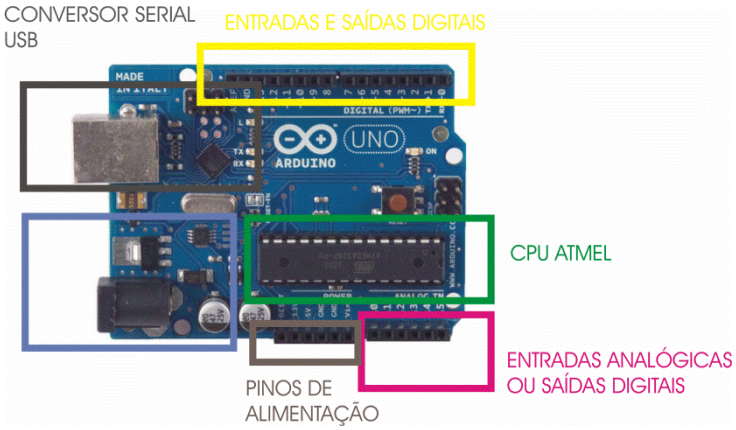
Figura 14 – Arquitetura do *Arduino*



Fonte: Produção do próprio autor.

São vários modelos possíveis de utilização *Arduino*, que se diferenciam na quantidade de memória ROM e nas configurações e quantidades de saídas e entradas disponíveis. O *Arduino* UNO foi utilizado neste trabalho, por ter um número de portas suficiente para a aplicação, mesmo sendo um modelo mais simples. A Figura 15 apresenta seu hardware, bem como as indicações dos componentes nele presentes.

Figura 15 – Arduino UNO



Fonte: Produção do próprio autor.

O processamento da placa Arduíno é realizado em um microcontrolador da ATMEL, o ATMega. O processador apresenta memória RAM, memória programada (ROM), unidade de processamento aritmética e portas de entrada e saída.

3 O SISTEMA JOANINHA

O sistema Joaninha foi desenvolvido com o intuito de atingir um público infantil, com faixa etária de 7 a 10 anos. Para garantir a atenção do usuário é que se pensou em conferir ao projeto aspectos figurativos e valorativos da percepção infantil, com objetos, ideias e imagens que remetem ao contexto vivido por uma criança. O próprio nome "Joaninha" é referência ao aspecto lúdico abordado no trabalho.

O projeto é composto de um robô (*hardware*) e uma interface para controlá-lo (*software*), que apresenta duas versões: uma para plataforma Android e outra para plataforma Windows. O *software* permite à criança controlar o robô de forma manual, ou de forma programada. Para que o robô entenda qual o modo utilizado e a ação a ser realizada pelo sistema, foi elaborado um protocolo de comunicação entre *software* e *hardware*.

O sistema pode operar de dois modos diferentes: "Modo Programado" e o "Modo Manual". O usuário deve escolher um modo de funcionamento do sistema ao utilizá-lo.

Modo Manual: Consiste no controle do robô utilizando o *software* programado para a plataforma *Android*. A velocidade de movimento do robô é fixada em 50 % da velocidade máxima e o usuário pode controlar o movimento do robô em movimentos retilíneo (no sentido progressivo e retrógrado) e de giro (sentido horário e antihorário) de forma instantânea.

Modo Programado: Consiste na programação de tarefas para o robô por meio da junção de blocos em uma interface gráfica desenvolvida para plataforma *Windows*. Cada bloco representa um movimento a ser conferido ao protótipo. O usuário também pode escolher e programar a velocidade e o tempo de execução de cada tarefa. As informações programadas são enviadas ao robô apenas quando o usuário permite que a aplicação seja executada.

Neste capítulo, será abordada a descrição de todos os componentes do projeto, como estrutura física do robô, a atuação dos motores, a interface gráfica com o usuário (por meio do computador e do celular) e o protocolo de comunicação desenvolvido.

3.1 A estrutura física do robô

O robô é composto principalmente por dois servomotores, um microcontrolador e um dispositivo de comunicação, como pode ser visto no esquema abaixo.

Peças de acrílico moldado deram forma à estrutura física do robô que pode ser visualizada na Figura 16. As rodas são acopladas aos servomotores, proporcionando o deslocamento do protótipo. O sinal de controle para a entrada do servo origina-se na saída do Arduíno, que também fornece a tensão necessária para alimentação do módulo *Bluetooth*. Todo o circuito é alimentado por uma bateria de 5 Volts, 1 Ampère, 3.5 Watts-hora.

Figura 16 – Estrutura física do robô

Fonte: Produção do próprio autor.

O servomotor utilizado é o modelo da SpringRC SM-S4303R, com rotação de 360º, cujas informações técnicas se encontram na Tabela 1:

Tabela 1 – Informações técnicas do servo motor SM-S4303R

	4.8 V		6V	
Peso (g)	Velocidade (r/min)	Torque (Kg.cm)	Velocidade (r/min)	Torque (Kg.cm)
44	60	3,3	70	4,8

Fonte: Produção do próprio autor.

O módulo utilizado no projeto para comunicação serial à distância é o modelo EGBT-046S, que utiliza o chip *Bluetooth* BC-417. Os dados técnicos do EGBT-046S podem ser apresentados na Tabela 2.

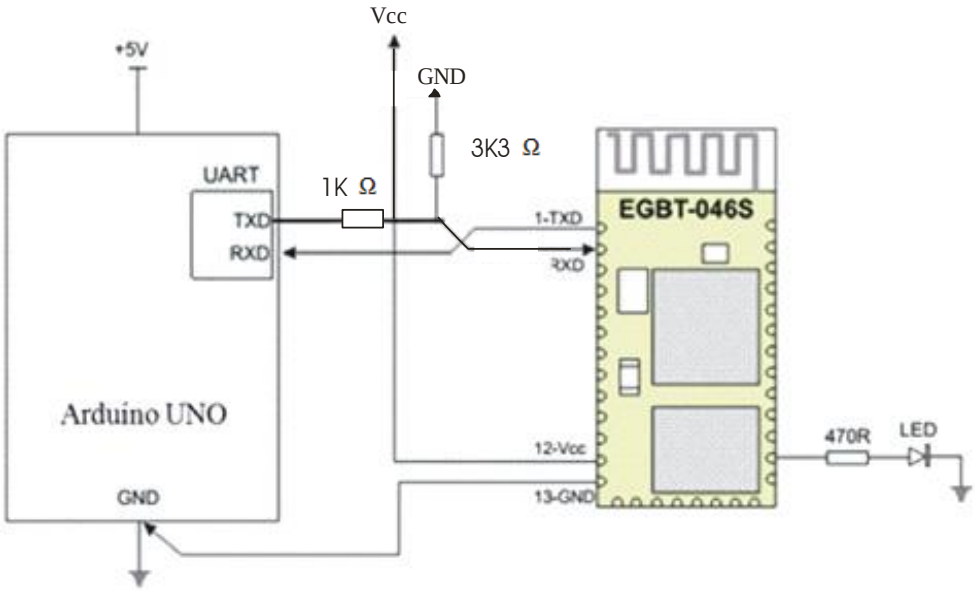
Tabela 2 – Informações técnicas do módulo *Bluetooth* utilizado EGBT- 046S

Dados Técnicos EGBT-046S	
Radio Chip:	CSR BC 417
Memória	Flash Externa 8 Mbit
Potência de saída	-4 a 6dbm Classe 2
Sensibilidade	-4 a 6dbm Classe 3
Interface	UART
Antena	Possui
Dimensão	27W x 13 H mm
Tensão	3,1 a 4,2 Vcc
Corrente	40mA max

Fonte: Produção do próprio autor.

O módulo *Blutooth* é alimentado por uma tensão contínua que pode variar de 3,1 V a 4,2 V. No caso da aplicação no projeto, a alimentação advém do próprio *Arduino*. No entanto, as portas de saída do *Arduino* fornecem sinal em 5 V e a porta RX do dispositivo de comunicação não é tolerante a tensões maiores que 3,3 V. Por isso é que se utilizou o esquema da Figura 17. Um divisor de tensão na saída do pino TX do microcontrolador garante que a tensão no pino RX do módulo *Bluetooth* apresente valores aceitáveis.

Figura 17 – Esquemático de ligação do módulo *Bluetooth* ao *Arduino*

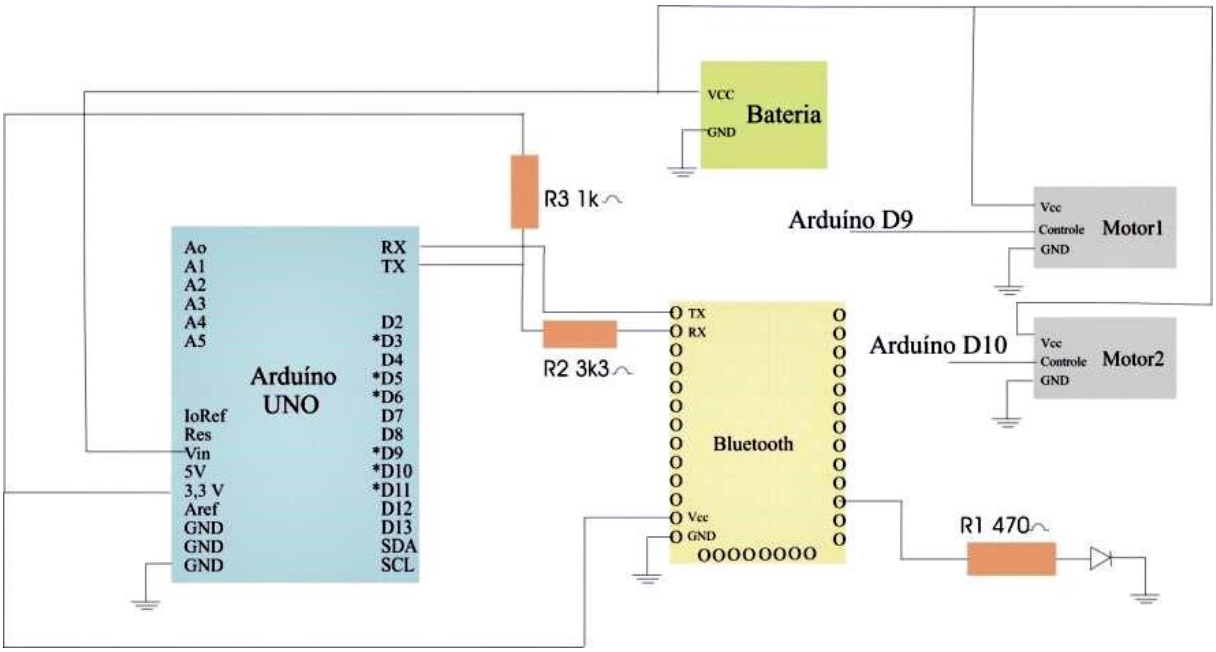


Fonte: E-GIZMO MECHATRONIX CENTRAL,20--.

O LED na Figura17 indica o status do dispositivo *Bluetooth*: enquanto o LED estiver aceso o seu status é de conectado. Se ele se apresentar piscando, isto significa que o dispositivo está esperando por conexão.

O esquemático visto na Figura 18 representa as ligações realizadas para o funcionamento do *hardware* do protótipo.

Figura 18 – Esquemático de ligação de todos os componentes do protótipo



Fonte: Produção do próprio autor.

Foi realizada a confecção das placas de circuito para integração de todos os componentes. A Figura 19 (a) é uma visão geral das placas de circuito confeccionadas e de todos os componentes já conectados e a bordo do robô. A Figura 19 (b) mostra o robô finalizado com a adequação da estrutura ao formato de joaninha.

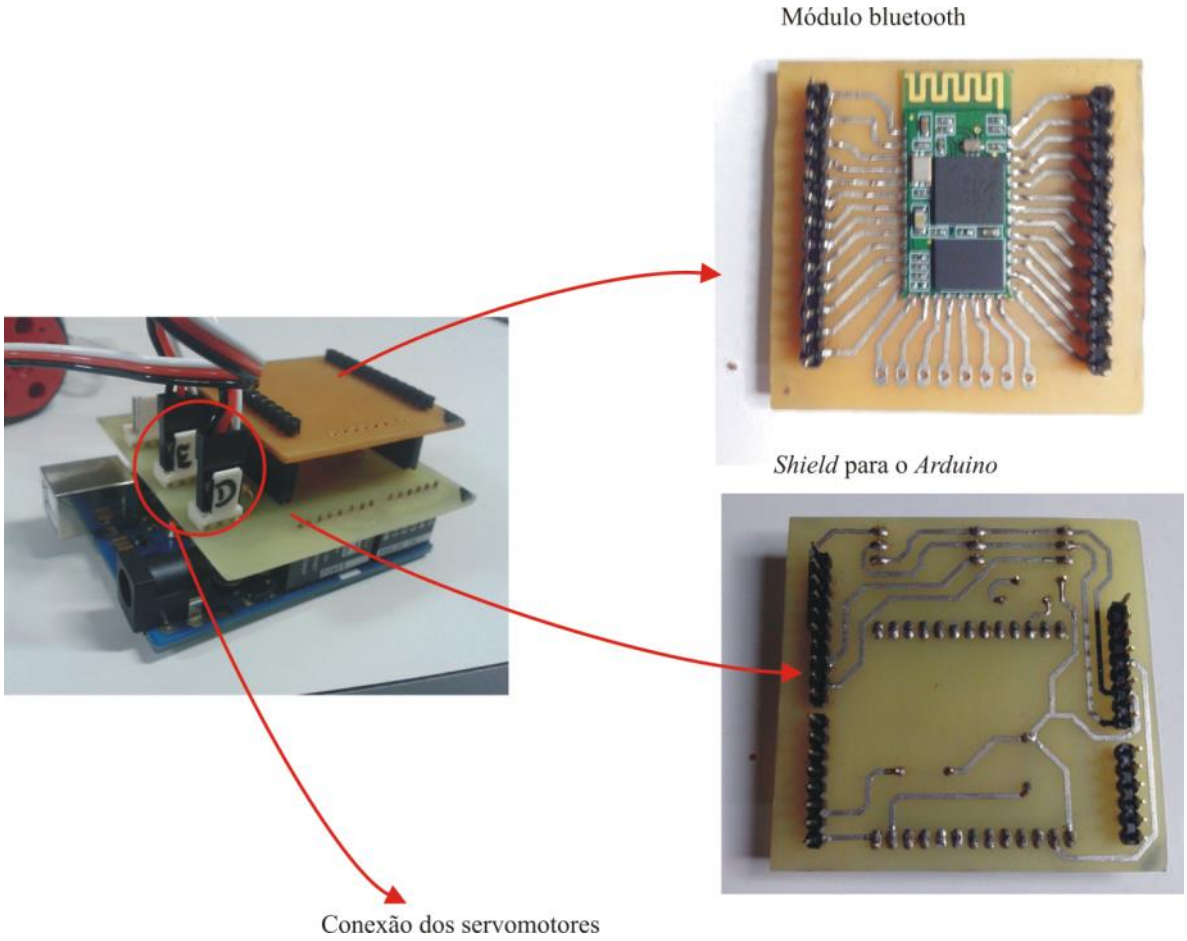
Figura 19 – Foto do robô e seus componentes

(a) (b)

Fonte: Produção do próprio autor.
Legenda: (a) Modelo estrutura do robô
(b) Carcaça do robô com a estrutura adaptada para apresentação do robô

Na Figura 20 podem-se visualizar, em detalhes, os componentes e suas ligações físicas na estrutura.

Figura 20 – Detalhes das placas de circuito montadas para atender ao projeto



Fonte: Produção do próprio autor.

3.2 A interface com o usuário

Para realizar a interação entre usuário e o sistema, o projeto oferece duas formas de utilizar o dispositivo: por meio do celular, ou por meio do computador. As duas aplicações apresentam funções diferentes e cada uma delas será detalhada nas seções 3.4.1 e 3.4.2 deste trabalho.

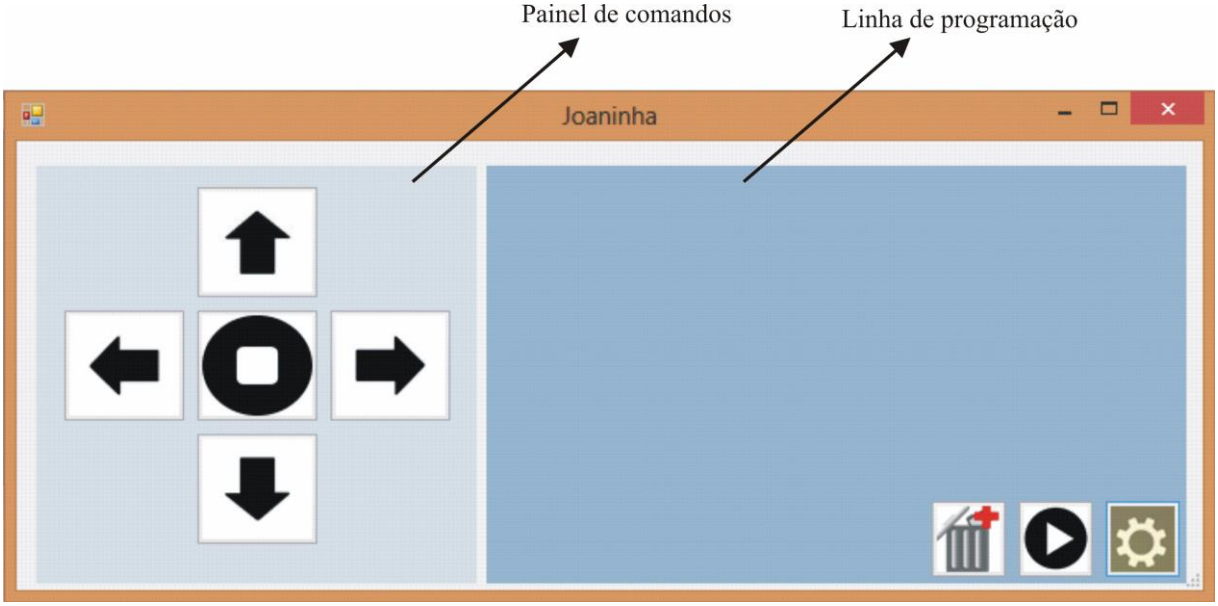
3.2.1 Interface para plataforma *Windows*

A interface do usuário com o sistema por meio do computador é a principal forma de acesso do projeto. É nesta interface onde é executado o Modo Programado do projeto Joaninha.

O *software* foi desenvolvido utilizando a linguagem de programação C#, por meio do *Visual Studio*. O principal objetivo do *software* é proporcionar à criança um meio de programar o movimento do robô, de maneira fácil e interativa.

Para isto, a interface é dividida em duas partes principais: o Painel de comandos e a Linha de programação como pode ser visto na Figura 21.

Figura 21 – Janela principal da interface com o usuário para computador



Fonte: Produção do próprio autor.

O Painel de comandos apresenta cinco blocos disponíveis ao usuário. Cada bloco representa uma direção diferente que pode ser passada ao robô, demonstrada pelas setas figuradas no próprio elemento, como detalhado na Tabela 3.

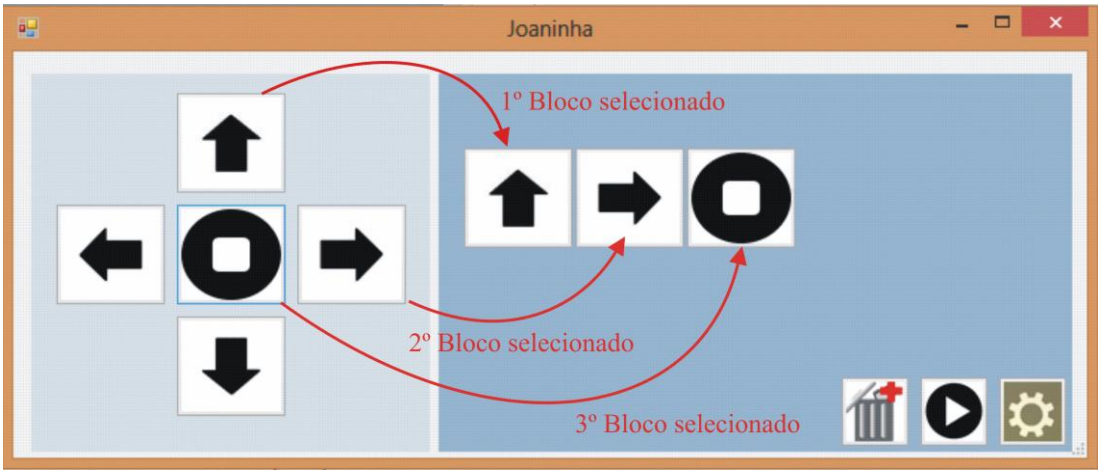
Tabela 3 – Ilustração das tarefas possíveis a serem programadas

Bloco	**Tarefa referente**
	Andar para frente
	Andar para direita
	Andar para esquerda
	Andar para trás
	Parar

Fonte: Produção do próprio autor.

Quando o usuário clica em um determinado bloco, escolhe um sentido e uma direção que o robô irá executar. O bloco escolhido pelo usuário é adicionado na linha de comandos do programa, sequencialmente.

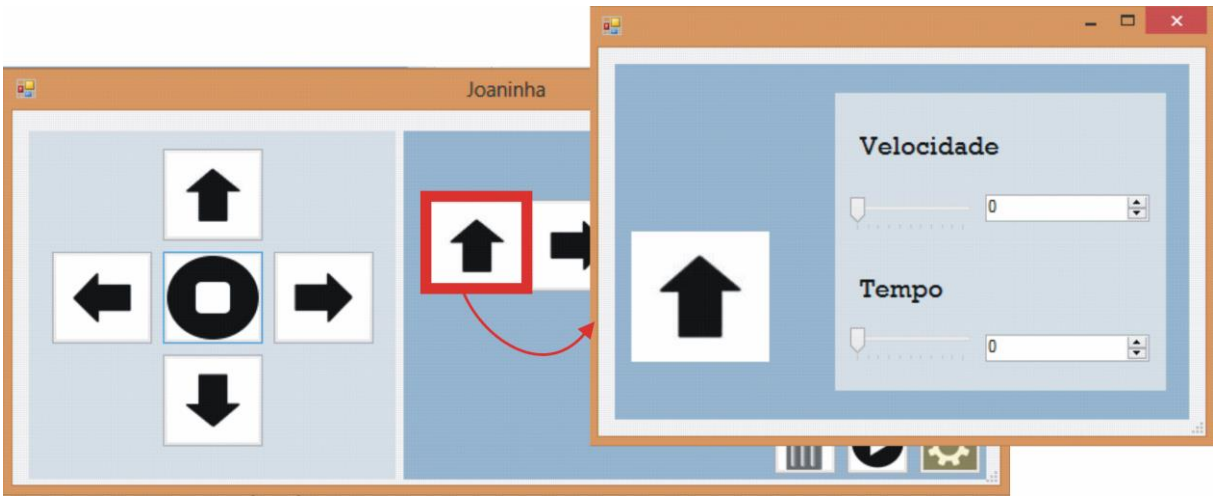
Figura 22 – Interface com o usuário: adição de blocos à linha de programação.



Fonte: Produção do próprio autor.

Ao realizar um clique sobre o novo bloco adicionado à linha de comandos, o usuário pode configurar aquela tarefa, que consiste na escolha de velocidade e tempo de execução do movimento selecionado. Desta forma, uma nova janela com opções de tempo e velocidade é aberta como mostrado na Figura 23.

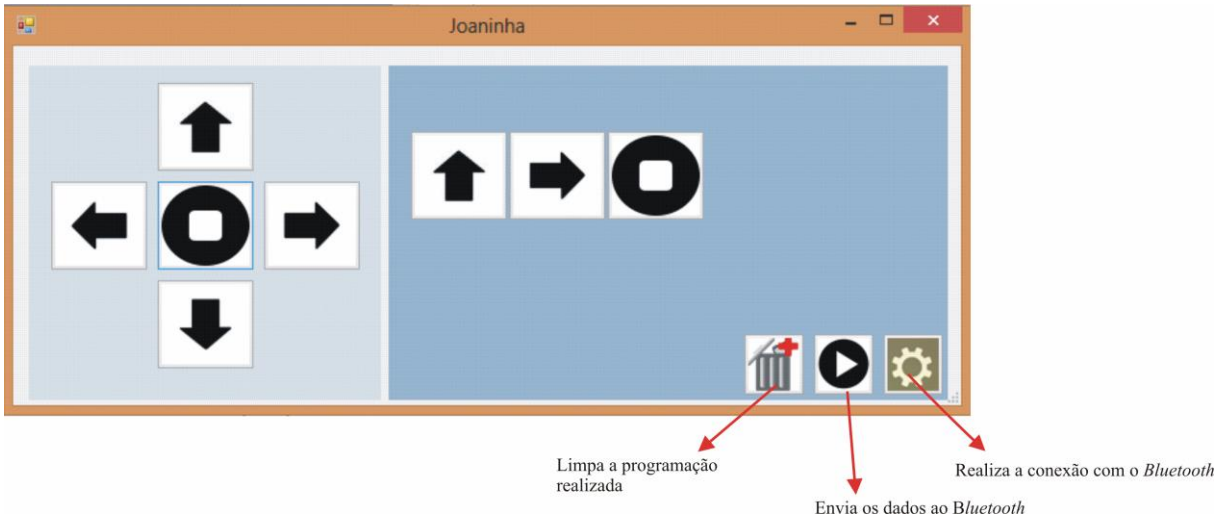
Figura 23 – Interface com o usuário: detalhe da janela para configuração de velocidade e tempo



Fonte: Produção do próprio autor.

Outros elementos necessários ao funcionamento do *software* estão dispostos na Figura 24, onde é explicada a função de cada elemento.

Figura 24 – Interface com o usuário: detalhe de outros elementos necessários ao funcionamento do programa



Fonte: Produção do próprio autor.

Após efetuar a programação do movimento, por meio da interface gráfica, a informação deve ser enviada ao robô. O usuário deve se conectar ao *Bluetooth* e enviar os dados clicando nos botões conforme demonstrado na Figura 24. O *software* escreve em um vetor de caracteres as informações programadas pelo usuário. Estas informações serão repassadas ao *hardware* pelo sistema de comunicação *Bluetooth*.

Para que a programação realizada pelo usuário seja enviada ao *hardware* de forma que este consiga traduzir a informação de maneira correta para o acionamento dos motores, foi criado um protocolo de comunicação entre *hardware* e *software*. O protocolo e a execução do movimento a partir dos caracteres recebidos serão explicados na seção 3.3 e 3.4 deste capítulo.

3.2.2 Interface para plataforma *Android*

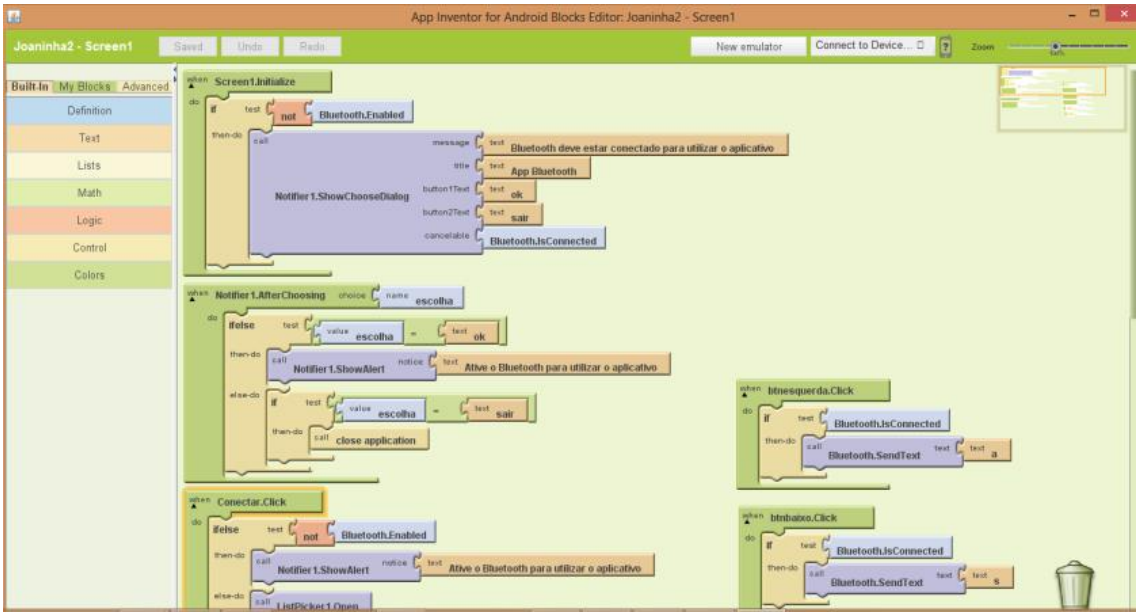
A interface do usuário com o sistema por meio do celular como ilustrada na Figura 25 é uma forma secundária de acesso do projeto. É nesta interface onde é executado o Modo Manual do sistema Joaninha. A principal diferença desta aplicação para a realizada no computador, é que pelo celular a criança apenas dirige o robô, sem se preocupar com a velocidade e o tempo de execução de cada tarefa: ao apertar um botão diferente o robô responde a este comando instantaneamente.

Figura 25 – Utilização da interface gráfica para celular

Fonte: Produção do próprio autor.

O *software* foi desenvolvido utilizando a plataforma de desenvolvimento de aplicativos desenvolvida pelo MIT (Instituto de Tecnologia de Massachusetts) denominada AppInventor. Esta plataforma consiste na edição de blocos lógicos com rotinas prontas. A interface do AppInventor é mostrada na Figura 26 .

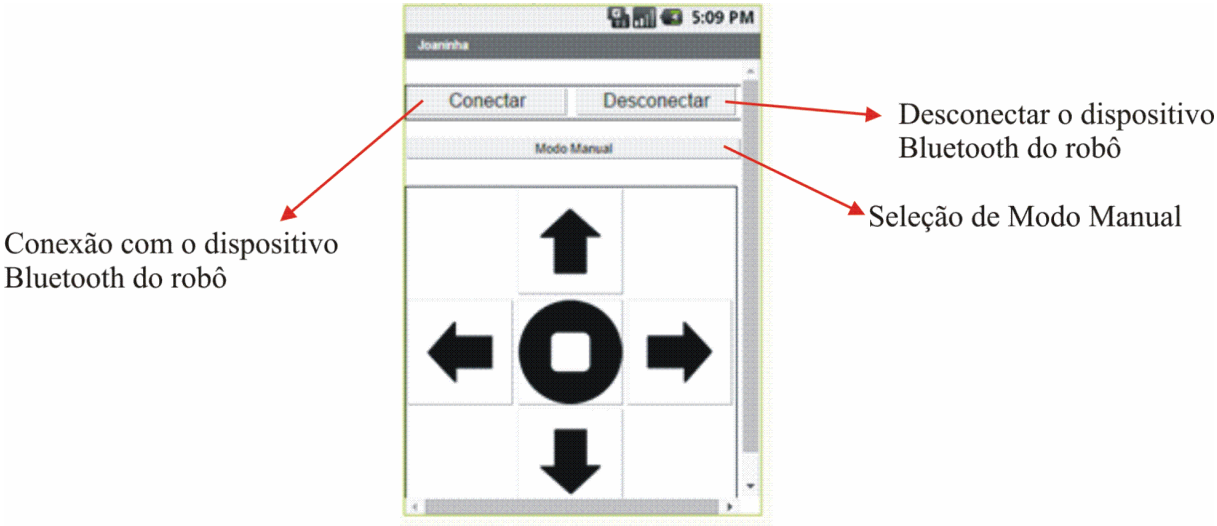
Figura 26 – Interface do AppInventor utilizado para programação do software para celular



Fonte: Produção do próprio autor.

O *display* mostrado na tela do celular do usuário é similar ao mostrado na Figura 27. No aplicativo é utilizado também imagens de setas para ilustrar o sentido e a direção do robô. Cada botão apresenta a mesma função dos blocos expostos na Tabela 3.

Figura 27 – Interface do programa desenvolvido para o celular



Fonte: Produção do próprio autor.

De forma análoga à interface para plataforma *Windows*, o aplicativo em *Android* utiliza o protocolo de comunicação criado para enviar os comandos ao robô, informando, no entanto, que o modo de operação é o Modo Manual. Isso pode ser ativado pelo botão "Modo Manual" apresentado na Figura 27. Sem a ativação deste botão, o robô continua em operação do "Modo Programado".

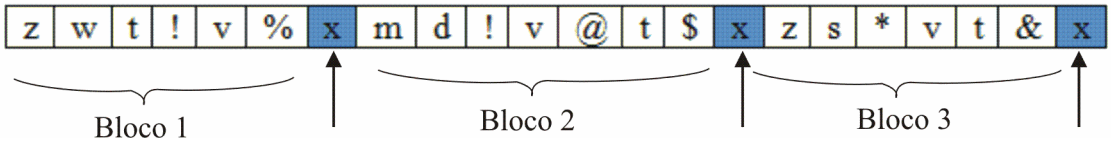
3.3 O Protocolo de comunicação entre o *Software* e o *Hardware*

Como apresentado nas seções anteriores, o módulo *Bluetooth* entrega ao *Arduino* um vetor de caracteres que representa a informação gerada pelo usuário. O sistema de controle programado no *Arduino* deve entender a informação recebida e acionar o movimento do robô de maneira correta, conforme desejado pela criança. Para que o *hardware* e o *software* consigam se comunicar de tal forma que a informação não se perca, foi necessário desenvolver um protocolo de comunicação.

O protocolo divide o vetor de caracteres recebido em blocos de informações. Para isto, utiliza-se o caractere “x” para localizar o término de um determinado bloco. Cada bloco representa uma tarefa a ser executada pelo robô. Cada tarefa deve ser acompanhada pelo seu tempo de execução e pela velocidade referente ao movimento escolhido.

Para melhor entendimento, toma-se como exemplo, o vetor de caracteres *“zwt!v%xmd!v@t$xzs*vt&x”*. A Figura 28 demonstra a divisão em blocos que o sistema deve identificar ao encontrar o caractere “x” no vetor escolhido como exemplo:

Figura 28 – Exemplo da divisão em blocos da palavra recebida

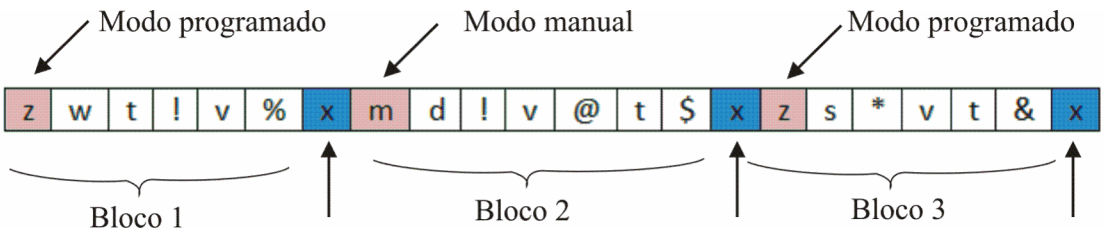


Fonte: Produção do próprio autor.

Para o exemplo dado, o vetor foi dividido em três blocos diferentes, o que representa 3 tarefas a serem executadas pelo robô.

O primeiro caractere de cada bloco é destinado ao reconhecimento do modo de operação. No protocolo, o caractere “z” representa o modo de operação programado, enquanto o caractere “m”, o modo manual. Pode-se tomar o mesmo exemplo anterior, onde se identifica para cada bloco do vetor, modos de operações diferentes como visto na Figura 29.

Figura 29 – Exemplo da interpretação no protocolo dos diferentes modos de operação.



Fonte: Produção do próprio autor.

Os próximos caracteres da sequência de cada bloco devem indicar a tarefa a ser executada, a velocidade e o tempo de execução. As tarefas possíveis e seus respectivos caracteres no protocolo podem ser vistos na Tabela 4.

Tabela 4 – Informações referentes aos caracteres de tarefa do protocolo

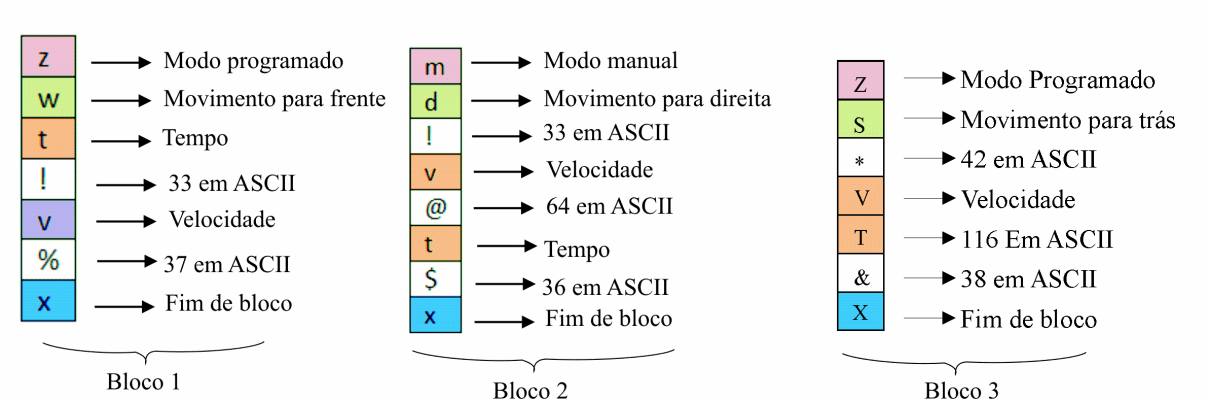
Caractere	Tarefa
w	Movimento para frente
s	Movimento para trás
a	Movimento para esquerda
d	Movimento para direita
q	Parar

Fonte: Produção do próprio autor.

Além dos caracteres de tarefa, o protocolo possui os caracteres “v” e “t” que representam respectivamente a informação de velocidade e de tempo da tarefa a ser executada no bloco. Todo caractere que sucede os caracteres “v” e “t” representam o valor de velocidade e tempo respectivamente, em ASCII.

Tomando ainda o vetor dado como exemplo, pode-se fazer a leitura completa de cada bloco identificado no vetor de caracteres, como mostrado na Figura 30.

Figura 30 – Leitura do protocolo para os três blocos dados como exemplo



Fonte: Produção do próprio autor.

No Bloco 1, a informação recebida e interpretada é que o usuário está utilizando o “Modo Programado”, deseja que o robô realize um movimento para frente, em um tempo de 33 segundos com uma velocidade de 37 por cento do valor máximo.

No Bloco 2 é identificado o modo de operação “Modo Manual”. Quando o *Arduino* identifica este modo, os outros caracteres que contém informações de velocidade e tempo não são necessários, uma vez que estes parâmetros no modo manual são fixados em: velocidade definida em 50% do valor máximo de PWM possível, e tempo de 0 segundo (um melhor entendimento sobre o modo pode ser visto na seção 3.4 deste capítulo).

O Bloco 3 apresenta um erro de informação: ao caractere de velocidade “v” não sucede nenhum valor para ser identificado como unidade da velocidade, pelo contrário, é passada a informação de tempo representada pelo caractere “t”. Neste caso, o *Arduino* entenderia que “t” é um parâmetro da velocidade (em ASCII t representa o inteiro 84) e o parâmetro de tempo da tarefa se perderia. Erros como este não podem acontecer, visto que prejudicam a tradução correta da informação transmitida. Estes critérios de segurança são tratados no envio da informação, cabendo ao *software* garantir que a informação passada ao *Arduino* esteja correta.

A Tabela 5 apresenta todos os caracteres utilizados no protocolo e os respectivos significados. Os demais caracteres são interpretados em ASCII como parâmetros da velocidade ou do tempo.

Tabela 5 – Informações da tarefa referentes a cada caractere do protocolo

Caractere	**Significado**	**Caractere**	**Significado**	**Caractere**	**Significado**
x	Fim de bloco	q	Parar	t	Tempo
z	Modo de operação programado	a	Movimento para esquerda	v	Velocidade
m	Modo de operação manual	s	Movimento para trás		
w	Movimento para frente	d	Movimento para direita		

Fonte: Produção do próprio autor.

3.4 O Controle realizado no *Arduino*

O microcontrolador do *Arduino* realiza duas tarefas principais para o funcionamento correto do sistema: a leitura de comandos recebidos do módulo *Bluetooth* e o acionamento dos motores baseado na informação recebida. Estas duas sub-rotinas e as sub-rotinas presentes

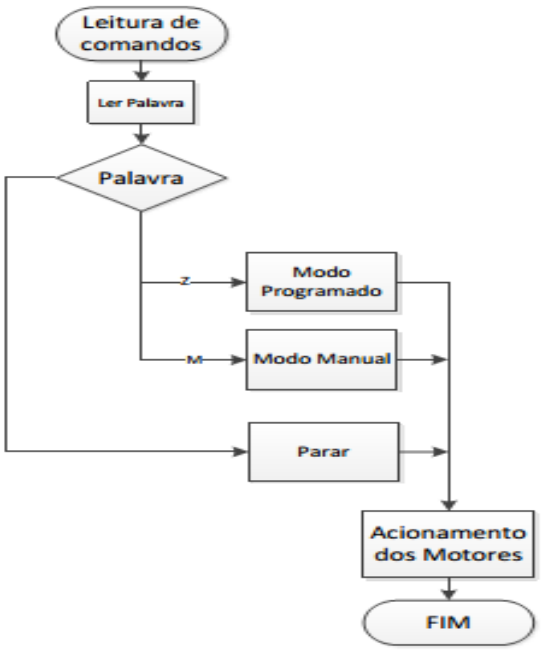
dentro destas, serão apresentadas neste capítulo de maneira a entender o funcionamento do controle realizado no *Arduino*.

3.4.1 Leitura de Comandos

O *Arduino* recebe como entrada um vetor de caracteres do sistema de comunicação. O usuário terá escolhido um modo de funcionamento do sistema. Se o modo escolhido for "Modo Programado", o microcontrolador irá utilizar uma rotina de interpretação do vetor recebido baseado no protocolo desenvolvido e explicado na subseção 3.2 deste capítulo.

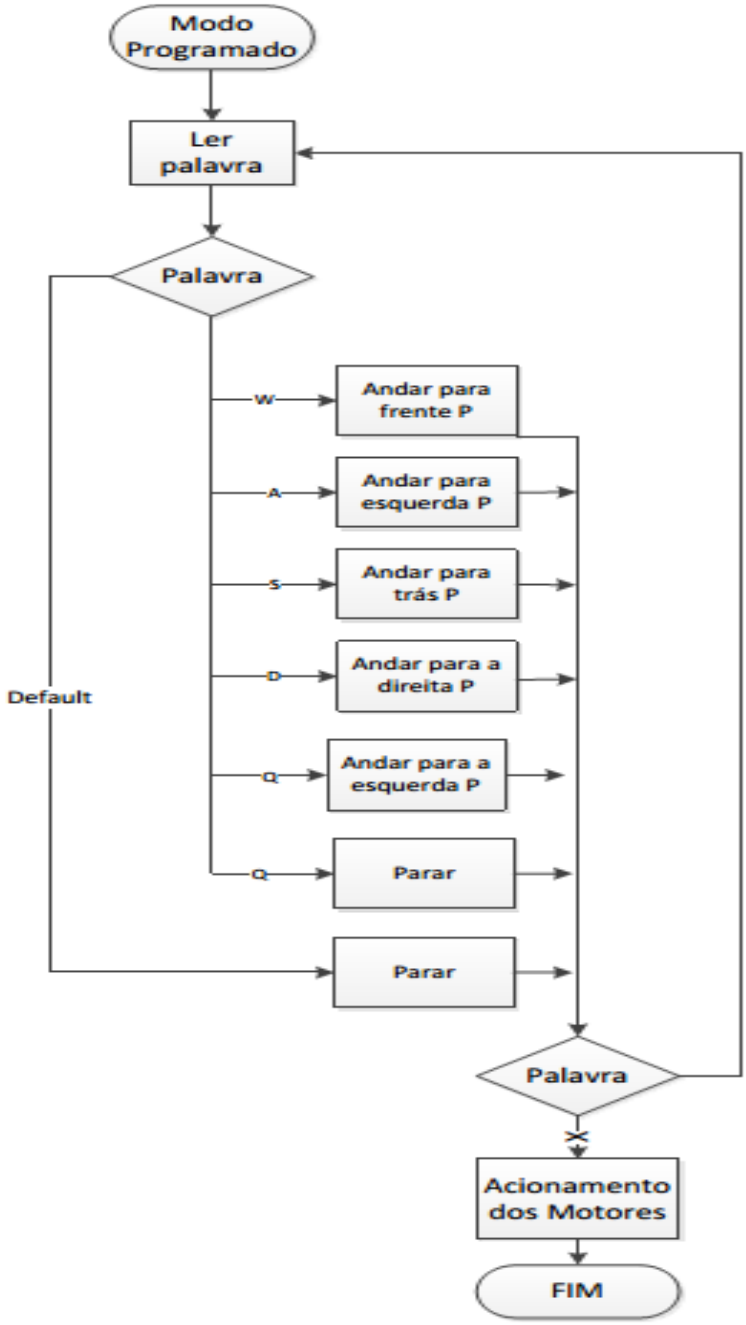
O fluxograma representado na Figura 31 demonstra o funcionamento do código programado para a leitura dos caracteres. A variável "Palavra" recebe os caracteres do sistema de comunicação e verifica qual é o primeiro byte. Se o modo escolhido for o programado, o programa chama a subrotina Modo Programado apresentada na figura 32. Se o modo escolhido for o manual, a subrotina Modo Manual é chamada e pode ser vista na Figura 33.

Figura 31 – Fluxograma ilustrando o funcionamento da subrotina Leitura de comandos



Fonte: Produção do próprio autor.

Figura 32 – Fluxograma ilustrando o funcionamento da subrotina Modo Programado

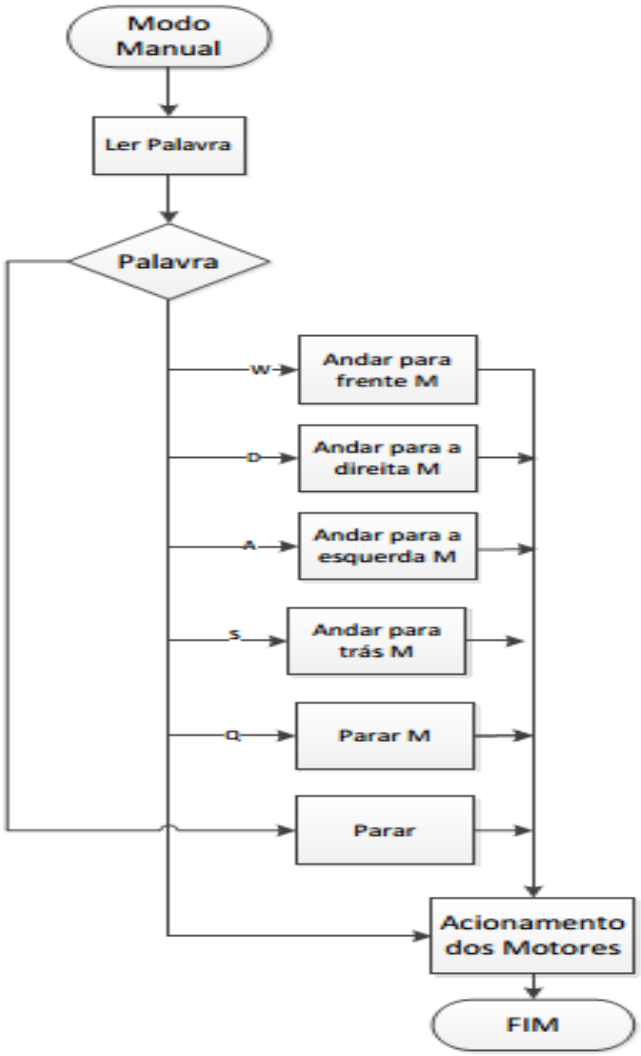


Fonte: Produção do próprio autor.

Ao identificar os caracteres de acordo com o protocolo de comunicação,nas funções “Andar para frente”, “Andar para a esquerda”, “Andar para a direita”, “Andar para trás” e “Parar” o programa inicializa uma variável “direção” e uma “velocidade” em um valor estimado pelo usuário, executando em seguida a subrotina “Acionamento dos motores”.

De maneira similar, o Modo Manual como mostrado na figura 33, identfica as funções a serem chamadas de acordo com o protocolo. A diferença é que nas funções “Andar para frente M”, “Andar para a esquerda M”, “Andar para a direita M”, “Andar para trás M” e “Parar M” é inicializada a variável “velocidade” em um valor fixo (50% do valor máximo do PWM), executando em seguida a subrotina “Acionamento dos motores”. O tempo aqui não é inicializado. O programa irá executar continuamente a tarefa lida até que uma nova tarefa diferente seja enviada ao sistema pelo usuário.

Figura 33 – Fluxograma ilustrando o funcionamento da subrotina Modo Manual



Fonte: Produção do próprio autor.

3.4.2 Acionamento dos motores

Na subrotina “Acionamento dos motores”, é de fato realizado a atuação dos servomotores a partir das variáveis inicializadas na rotina anterior. Ou seja, do conjunto de informações de direção, sentido e velocidade, é selecionada a largura de pulso que terá o sinal PWM a ser transmitido ao motor. O sinal é enviado por certo período de tempo determinado pela variável “tempo”.

No caso do Modo Manual, o acionamento dos motores é realizado continuamente até que o usuário envie uma informação diferente da atual.

4 VALIDAÇÃO EXPERIMENTAL, RESULTADOS E DISCUSSÕES

4.1 Protocolo experimental

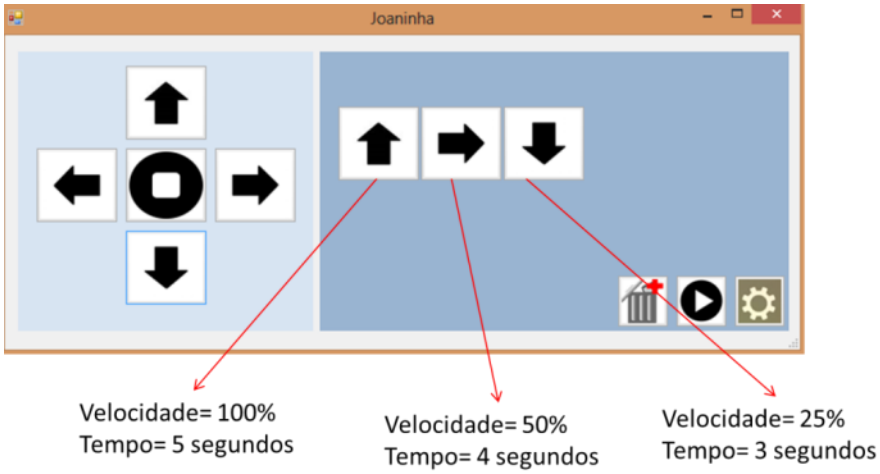
Foram realizados alguns testes para verificação do funcionamento do *hardware* do sistema, da comunicação entre *hardware* e *software* e da aplicabilidade do sistema com a utilização por usuários.

O primeiro teste realizado consistiu na definição da variável de entrada do sistema para um acionamento de valor máximo de tensão no PWM, no sentido progressivo e também no sentido retrógrado. O experimento foi repetido por 5 vezes, tendo sido capturados os pontos iniciais e finais de cada uma das trajetórias realizadas, bem como o tempo de execução.

O sistema não apresenta sensores para medição de posição ou velocidade, sendo que o controle realizado no sistema não apresenta realimentaçãode. Desta maneira não é possível obter por dados do controle qual é a velocidade acionada real. O experimento então 1 se propõe a obter a máxima velocidade que pode ser adquirida (com acionamento total do PWM). É então objetivo do experimento avaliar o protótipo robótico e sua funcionalidade, levantar informações de distância percorrida para estimar parâmetros como a velocidade média máxima do robô e a precisão de sua trajetória.

O segundo experimento realizado, diz respeito à capacidade do sistema de executar a trajetória solicitada pelo usuário. Para realizar tal estimativa, foi gerada na interface com o computador uma sequência de comandos como apresentado na Figura 34.

Figura 34 – Sequência de comandos gerada para experimentação do sistema.



Fonte: Produção do próprio autor.

Foram capturadas imagens de vários instantes do experimento a fim de estimar os pontos percorridos durante a execução dos movimentos programados e demonstrar a trajetória realizada. O teste foi repetido por 5 vezes.

O objetivo deste experimento foi o de avaliar se a comunicação entre *hardware* e *software* está acontecendo de maneira correta, se o sistema envia todos os parâmetros e tarefas, consegue entendê-los e executá-los. O experimento 2 analisa também a repetibilidade do robô, ou seja, sua capacidade de gerar a mesma resposta do sistema, para uma mesma entrada.

O terceiro experimento consistiu na utilização do sistema por 5 usuários na faixa-etária de 7 a 10 anos. O participante deveria executar uma tarefa que consistia em fazer com que o protótipo realizasse um deslocamento correto: conseguir conduzir o robô até uma determinada caixa, descrita às crianças como “Casa da Joaninha”.

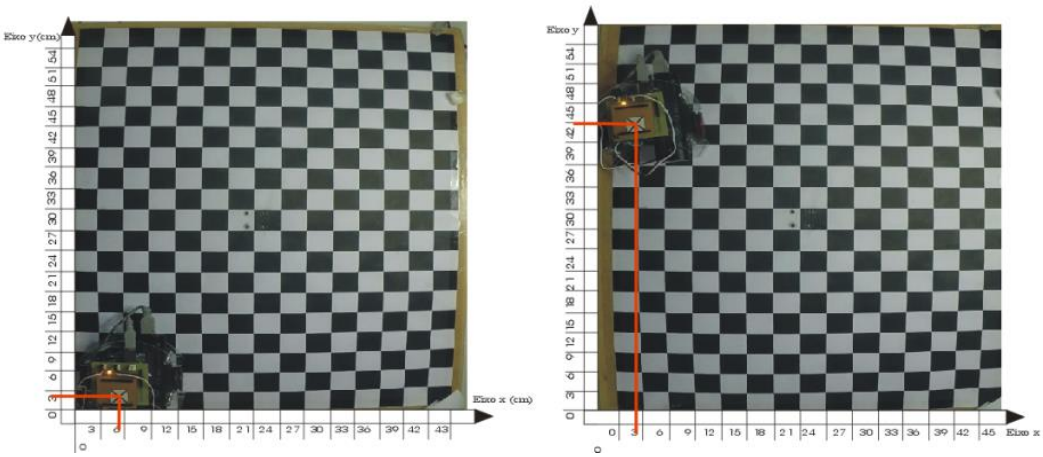
O experimento foi realizado inicialmente com a plataforma *Android e* posteriormente com a plataforma *Windows*. Inicialmente, nenhuma explicação era dada aos usuários, que deveriam ter um tempo de contato com o sistema, a fim de reconhecer a funcionalidade dos programas, na tentativa de executar a tarefa apresentada a eles. Após esta etapa, era dada uma explicação sobre os sistemas e a tarefa era executada por cada usuário novamente, agora tendo conhecimento de como o programa funciona por inteiro.

Os resultados foram levantados para todos os três experimentos e podem ser vistos nos itens 4.2, 4.3 e 4.4 deste capítulo.

4.2 Resultados e discussão do experimento 1

Inicialmente, foi enviado aos motores um comando para atuação com máxima tensão possível no PWM no sentido progressivo, como referenciado na Figura 35.

Figura 35 – Experimento realizado: movimento do robô retílínio, sentido progressivo.



Ponto Inicial Ponto Final

Fonte: Produção do próprio autor.

Na Figura 34 é possível observar que o robô parte do ponto estimado em (5,8;2,8) para o ponto (5,8;45). Os dados obtidos para todas as 5 repetições do movimento se encontram na Tabela 6.

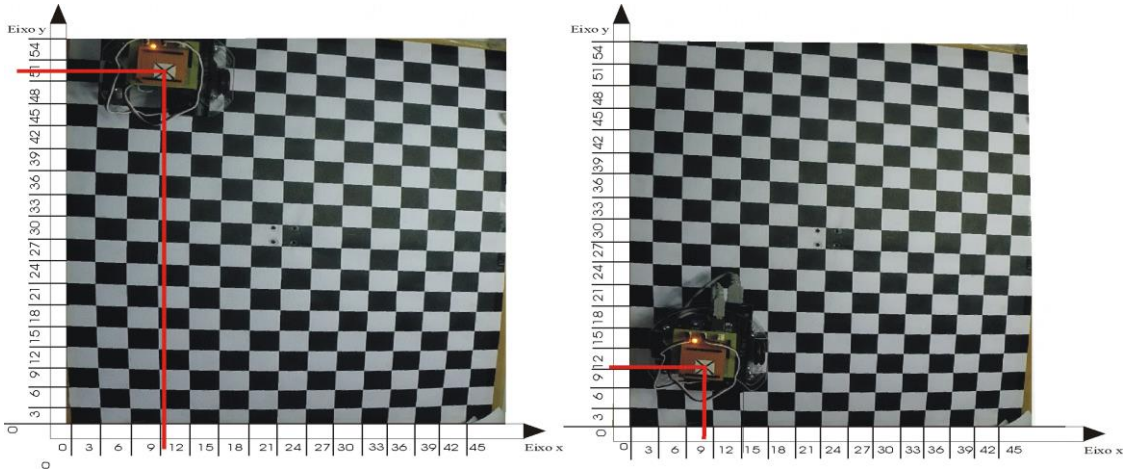
Tabela 6 – Resultados do experimento para movimento retilíneo, sentido progressivo

	Tempo (s)	Distância Percorrida (cm)	Velocidade (cm/s)	Média das Velocidades (cm/s)	Desvio Padrão Distância	Desvio Padrão Velocidade
Execução 1	5	42,20	8,44			
Execução 2	5	39,10	7,82			
Execução 3	5	40,91	8,18	8,10	1,08	0,22
Execução 4	5	40,61	8,12			
Execução 5	5	39,62	7,92			

Fonte: Produção do próprio autor.

O resultado para o sentido retrógrado do referencial, com velocidade máxima, pode ser visto na Figura 35.

Figura 36 – Experimento realizado: movimento do robô retilínio, sentido retrógrado.



Ponto Inicial Ponto Final

Fonte: Produção do próprio autor.

É possível perceber na Figura 36 que o robô inicia seu movimento no ponto estimado (9,2;49,5), e termina em (8,6;9,1). O experimento foi repetido por 5 vezes e os resultados são apresentados na Tabela 7.

Tabela 7 – Resultados do experimento para movimento retilíneo, sentido retrógrado

	Tempo (s)	**Distância Percorrida (cm)**	**Velocidade (cm/s)**	**Média das Velocidades (cm/s)**	**Desvio Padrão Distância**	**Desvio Padrão Velocidade**
Execução 1	5	40,40	8,08			
Execução 2	5	41,30	8,26			
Execução 3	5	41,70	8,34	8,20	0,49	0,097
Execução 4	5	41,00	8,20			
Execução 5	5	40,50	8,10			

Fonte: Produção do próprio autor.

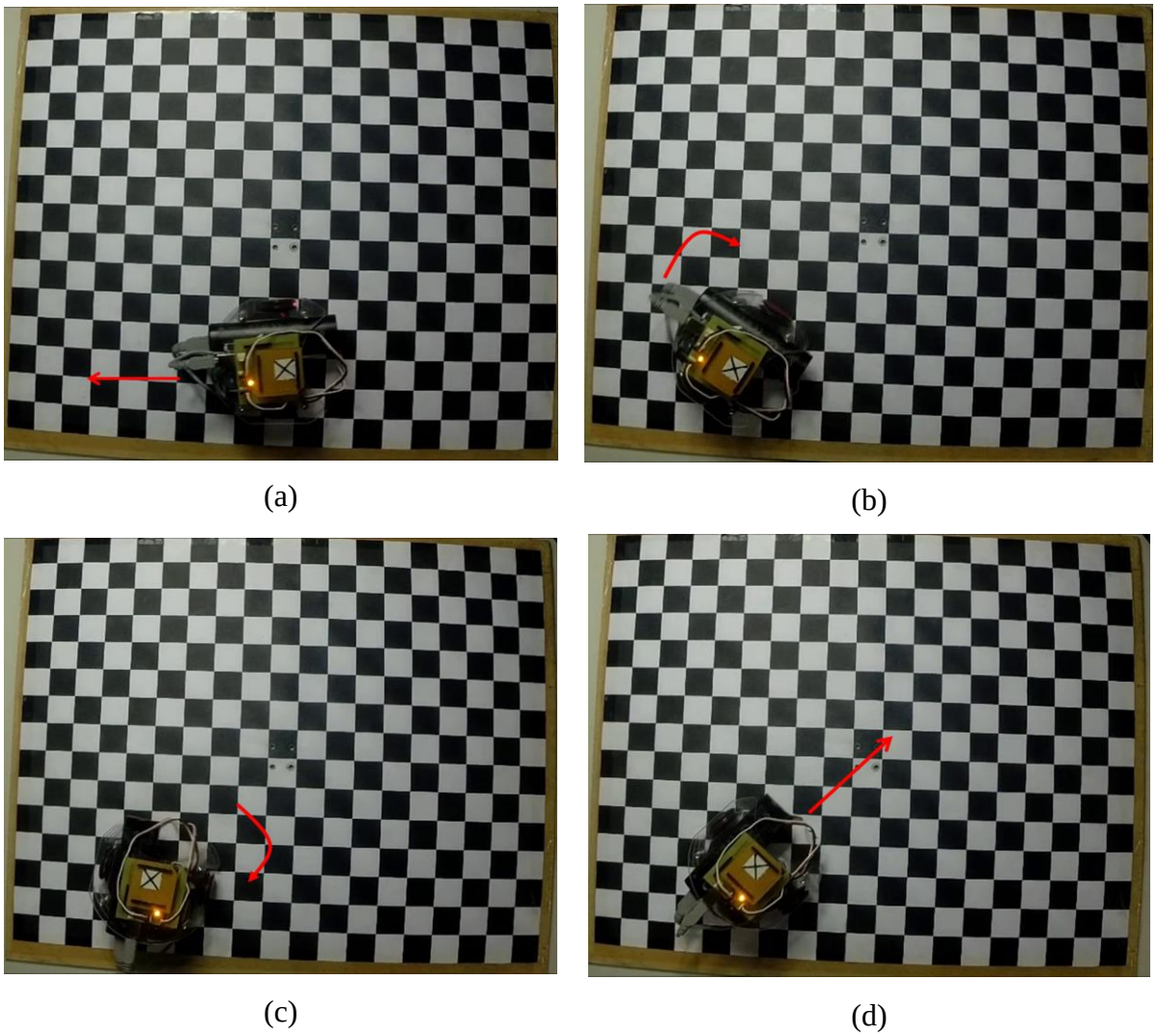
Com os experimentos apresentados, foi possível estimar uma velocidade média máxima para o protótipo de 8,10 cm/s para o movimento progressivo e 8,20 cm/s para o movimento retrógrado. É importante observar que esta velocidade pode variar também de acordo com o peso do protótipo (variação do torque), que seria o caso de uma possível substituição de bateria por um modelo diferente, ou até mesmo adição de sensores ou atuadores ao sistema.

Os valores de deslocamento aferidos em todas as experiências são muito próximos, com um desvio padrão de 1,08 e 0,49 para movimento progressivo e retrógrado, respectivamente. Isto demonstra que o sistema executa uma trajetória muito próxima da desejada para todos os experimentos aferidos.

4.3 Resultados e discussão do experimento 2

A partir da captura de quadros de imagens do experimento 2, foi possível estimar os pontos percorridos. A trajetória é ilustrada na Figura 37.

Figura 37 – Quadros demonstrando as direções executadas na execução da tarefa.

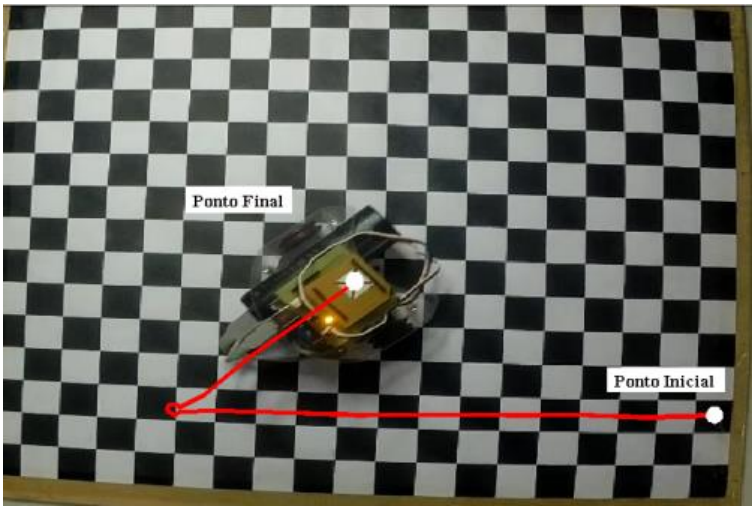


Fonte: Produção do próprio autor.

Legenda: (a) Movimento retilíneo do robô
(b) Início do giro sentido horário do robô
(c) Final do giro sentido horário do robô
(d) Movimento retrógrado do robô

A Figura 38 apresenta todo o deslocamento da trajetória realizada pelo robô.

Figura 38 – Trajetória percorrida para a tarefa executada



Fonte: Produção do próprio autor.

Para os movimentos retilíneos que compõe a tarefa programada no experimento, foram obtidos valores de distância percorrida e estimativas da velocidade como podem ser vistos na Tabela 8.

Tabela 8 – Resultados do experimento para tarefa programada.

		Tempo (s)	**Distância Percorrida**	**Velocidade**	**Média das Velocidades (cm/s)**	**Desvio Padrão Distância**	**Desvio Padrão Velocidade**
	Execução 1	5	40,53 (cm)	8,11(cm/s)			
	Execução 2	5	39,01 (cm)	7,80 (cm/s)			
	Execução 3	5	39,53 (cm)	7,91 (cm/s)	7.95	0,51	0,102
	Execução 4	5	40,03 (cm)	8,00 (cm/s)			
	Execução 5	5	39,58 (cm)	7,92 (cm/s)			
	Execução 1	3	20,18	6,73			
	Execução 2	3	20,62	6,87			
	Execução 3	3	20,93	6,98	6,73	0,18	0,53
	Execução 4	3	19,81	6,60			
	Execução 5	3	19,45	6,48			

Fonte: Produção do próprio autor.

A trajetória na Figura 37 foi estimada a partir dos quadros do vídeo gravado durante o experimento. A tarefa apresenta dois movimentos retilíneos, porém com velocidades diferentes e um movimento circular.

É possível concluir que a trajetória foi executada conforme desejada, realizando todos os movimentos programados na interface demonstrados na Figura 33. Isso comprova que a comunicação do sistema está sendo realizada de maneira correta, e que o robô é capaz de desenvolver variação de velocidades e de tempo em seu movimento quando isto é requerido.

4.4 Resultados e discussão do experimento 3

O sistema foi testado por usuários com sexos diferentes (meninos e meninas) com idade dentro da faixa etária estimada: 7 a 10 anos.

As características de cada usuário que participou dos experimentos, são apresentadas na Tabela 9.

Tabela 9 – Características de cada usuário do experimento realizado

Usuário	Idade	Sexo
01	10	Masculino
02	8	Masculino
03	8	Feminino
04	9	Feminino
05	10	Feminino

Fonte: Produção do próprio autor.

Cada criança realizou o experimento com ambas as plataformas: para celular e para computador como pode ser mostrada na Figura 39.

Figura 39 – Experimentos realizados com usuários do sistema com o aplicativo do celular e com o computador.

Fonte: Produção do próprio autor.

4.4.1 Experimento utilizando o Aplicativo para celular: Modo Manual

A Tabela 10 apresenta os resultados obtidos do experimento realizado pelas crianças utilizando o “Modo Manual” por meio do celular.

Tabela 10 – Resultados experimento com o aplicativo para celular.

Número de usuários que conseguiram reconhecer a funcionalidade do programa sem explicação prévia.	5
Tempo de reconhecimento da funcionalidade do programa	Instantâneo para todos os usuários
Número de usuários que conseguiram realizar a tarefa completamente	5

Fonte: Produção do próprio autor.

A Tabela 11 apresenta o tempo de execução da tarefa desenvolvida por cada usuário, que consiste em realizar o movimento do robô até que ele entre em uma caixa denominada “Casa da Joaninha”.

Tabela 11 – Tempo de execução da tarefa por cada usuário do sistema utilizando o aplicativo para celular.

Usuários	**Tempo de execução da tarefa**	**Média**	**Desvio Padrão**
Usuário 01	19 s		
Usuário 02	26 s		
Usuário 03	20 s	34 s	17,21
Usuário 04	65s		
Usuário 05	40s		

Fonte: Produção do próprio autor.

As opiniões dos diferentes usuários do sistema se basearam em reconhecer a facilidade de utilização do sistema para plataforma *Android*. Todos os usuários se demonstraram envolvidos com a tarefa e apresentaram interesse pela aplicação, tendo todos eles conseguido realizar o objetivo do experimento com um tempo médio do grupo de 34 segundos.

4.4.2 Experimento utilizando a interface pelo computador: Modo Programado.

A Tabela 12 apresenta os resultados obtidos do experimento realizado pelas crianças utilizando o “Modo Programado” por meio do computador.

Tabela 12 – Resultados experimento com o computador em modo programado.

Número de usuários que conseguiram reconhecer a funcionalidade totalmente do programa sem explicação prévia.	0
Número de usuários que conseguiram reconhecer a funcionalidade parcialmente do programa sem explicação prévia.	4
Número de usuários que não conseguiram creconhecer a funcionalidade do programa sem explicação prévia.	1

Fonte: Produção do próprio autor.

Os usuários não conseguiram identificar a total funcionalidade do programa sem explicação prévia. Dos 5 usuários, 4 conseguiram realizar o movimento com o robô, mas não identificaram que para cada tarefa era possível selecionar a velocidade e o tempo. Nenhuma criança conseguiu executar completamente a tarefa sem intervenções ou explicações do funcionamento do sistema.

Depois de explicadas as tarefas de cada botão e toda a funcionalidade do programa, todos os usuários conseguiram executar a tarefa. O tempo médio do desempenho de cada um é mostrado na Tabela 13.

Tabela 13 – Tempo de execução da tarefa por cada usuário do sistema utilizando o computador, modo programado.

Usuários	Tempo de execução da tarefa	Média	Desvio Padrão
Usuário 01	1m e 22 s		
Usuário 02	2 m e 32 s		
Usuário 03	4 m e 4 s	3 m e 8 s	1m e 3 s
Usuário 04	4 m e 3 s		
Usuário 05	3 m e 42 s		

Fonte: Produção do próprio autor.

Os usuários descreveram o Modo Programado como mais difícil, mas demonstraram interesse em programar o movimento do robô, tendo nenhum deles desistido da tarefa de levar o robô até a "Casa da Joaninha". O tempo médio de excução da atividade foi de 3 minutos e 8 segundos.

4.4.3 Avaliação do sistema pelos participantes do experimento

Em geral, os participantes identificaram o Modo Manual, utilizando o aplicativo para celular como mais fácil, tendo reconhecido sua funcionalidade sem auxílio de terceiros. O modo programado foi identificado com maior dificuldade, mas mesmo assim, após realizadas as explicações de seu funcionamento, todas as crianças conseguiram realizar as tarefas sem nenhuma desistência, dentro de um tempo médio aceitável.

Alguns comentários dos usuários foram selecionados a serem apresentados para pontuar aspectos do sistema observado pelos participantes:

> "Foi muito legal e fácil brincar com o celular, mas eu preferia que os botões funcionassem apenas quando eu clicasse e não ficassem no movimento até eu clicar outro botão" C01.

O comentário C01 demonstra uma melhoria a ser realizada no sistema entendida por parte do usuário. No sistema atual, quando um botão é selecionado, o robô repete o movimento até que outro botão seja escolhido.

> "O aplicativo é mais fácil de mexer, mas depois que eu soube que podia mudar velocidade e tempo no computador, achei os dois fáceis. Todos são divertidos" C02.

O comentário C02 demonstra o interesse do usuário por ambas as plataformas, apesar de ter a opinião de maior dificuldade na plataforma do computador. O usuário identifica a necessidade de explicação da funcionalidade do *software* para realizar a tarefa.

> "Acho que a joaninha deveria ter bolinhas pretas" C03.

O comentário C03 do usuário faz referência à forma física e o aspecto lúdico do sistema. O usuário identificou a importância da representação do robô em características do contexto infantil para atendimento do público alvo.

Todos os participantes entenderam a dificuldade maior do modo programado como um desafio, e foi possível notar a dedicação de todos em executar a tarefa. O observado valida a atenção que a interface ganhou dos usuários, se mostrando como uma ferramenta interessante. Todas as crianças que participaram do experimento alegaram que gostariam de ter o Sistema Joaninha para utilizar.

5 CONCLUSÃO E TRABALHOS FUTUROS

O Sistema Joaninha apresentou um bom desempenho, tanto em características construtivas e funcionalidade do sistema robótico, como na utilização por parte do público alvo do sistema.

Os objetivos deste trabalho foram cumpridos: o desenvolvimento de uma interface com o usuário, de um sistema de comunicação entre *hardware* e *software*, de um protocolo que viabilizou tal comunicação e de um controlador para acionamento dos motores.

É importante destacar que o robô apresentado neste projeto não possui uma realimentação no controle das variáveis de entrada do sistema. Não é utilizado nenhum tipo de sensor que permita este procedimento, como por exemplo, a utilização de *encoders*.

Desta forma, não é possível obter informações práticas de variáveis de entrada so sistema como a velocidade média. Os experimentos realizados no item 4.1 deste trabalho para a obtenção da velocidade média foi uma estimativa calculada sobre a informação estimada de medições do deslocamento do protótipo. Os resultados permitem chegar à conclusão de que o sistema é preciso, uma vez que a repetição das tarefas aconteceu de maneira efetiva, para todos os experimentos realizados. Para mesmo valor de entrada no PWM, obteve-se sempre o mesmo comportamento do robô.

Quanto à comunicação do sistema, foi possível verificar um funcionamento eficaz e transmissão dos comandos de maneira confiável. A tarefa programada apresentada na seção 4.1, foi executada de maneira correta pelo protótipo demonstrando uma boa interação do sistema.

A estimativa de preço para a realização do projeto é de R$350, sem incluir as horas de desenvolvimento. Vale ressaltar que um Mindstorm da LEGO custa, no Brasil, cerca de R$2800.

O projeto atendeu ao público alvo proposto. Os experimentos realizados com os usuários demonstraram que a interface do Modo Programado necessita de explicação para ser totalmente compreendida e nem todas as funcionalidades estão explícitas. Melhorias podem

ser realizadas no sistema a fim de tornar a aplicação mais intuitiva. No entanto, após explicação do sistema, as crianças puderam realizar as tarefas e alçancaram o objetivo. Todas conseguiram assimilar bem os comandos de tarefa, o comando de envio da informação e o de limpar a tela.

Como trabalhos futuros, alguns pontos podem ser levantados para a melhoria da ferramenta:

- Adequação de um sistema de proteção ao *hardware*, como sensores para impedir colisão com obstáculos.
- Ajuste da interface de determinadas tarefas, de forma a deixar explícito alguns comandos, como o de ajuste dos parâmetros do robô (velocidade e tempo). Uma forma de implementar esta alteração seria a chamada da janela no programa para inicializar tais variáveis logo quando o usuário escolher a tarefa a ser desenvolvida.
- A fim de que a interface possa ser utilizada em faixa etária inferior a de 7 anos, modificações podem ser previstas no sistema, construindo níveis de tarefas a serem programadas. O usuário pode ir apresentando evolução na ideia de programação enquanto brinca com os diferentes níveis de dificuldade do projeto.
- O controle do robô por meio do celular capturou a atenção dos usuários por apresentar uma maneira prática e portátil de utilizar o sistema. Um trabalho a ser desenvolvido pode ser o desenvolvimento do Modo Programado para aplicativos em plataforma *Android* além da plataforma *Windows*.

O projeto se apresentou uma proposta de baixo custo, interativa e atrativa ao público de crianças de 7 a 10 anos de idade. Em todos os experimentos com os usuários, ficou evidente a motivação de utilizar o sistema por parte das crianças, que entenderam os conceitos de programação apresentados pela interface.

REFERÊNCIAS

DELORS, J. **Educação:** um Tesouro a Descobrir. Relatório para a UNESCO da Comissão Internacional sobre Educação para o século XXI, São Paulo: Cortez, 1998.

SILVA, A. et al. RoboEduc**:** A Software for Teaching Robotics to Technological Excluded Children Using LEGO Prototypes. In: **IEEE LATIN AMERICAN ROBOTICS SYMPOSIUM – LARS**, 3.2006,Santiago, Chile. Abstracts.Santiago, Chile: Universidad de Chile, 10/2006.

ALVES, S.F.R. et al. Proposal of educational environments with mobile robots. **Robotics, Automation and Mechatronics (RAM)**, 2011 IEEE Conference on, p.264,269, 17-19, 2011.

LEGO. **Lego education**. LEGO Group. 2014. Disponível em: <http://education.lego.com/enus/>. Acesso em: 12 dez. 2014.

PAPERT, S. **Constructionism:** a new opportunity for elementary science education**.** Massachusetts Institute of Technology, The Epistemology and Learning Group. Proposta para a National Science Foundation, 1986.

PIAGET, J. **Piaget's Theory.** 1th ed. New York: Spring-Verlag, Inc. 1976.

CASTILHO, M. I. **Robótica na Educação:** Com que objetivos? 2002. Monografia de Especialização em Informática na Educação — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

PAPERT, S. **Logo:** Computadores e Educação. 1ª ed. São Paulo: Brasiliense, 1985.

ROBOLAB, **Robolab**. LEGO Group. 2014. Disponível em: <https://education.lego.com/en-us/lego-education-product-database/mindstorms/2000069-robolab-2-9-software>. Acesso em: 12 dez. 2014.

MINDSTORMS. **LEGO Mindstorms EV3**. LEGO Group. 2014. Disponível em: <http://www.lego.com/en-us/mindstorms/build-a-robot/ev3rstorm>. Acesso em: 12 dez. 2014.

TAVARES, D.M.; ANTUNES, V.A.; GONÇALVES, L.M.G. Em Evidência o Potencioal e Limitações dos Compiladores NQC e BrickOS e seus Respectivos Sistemas Operacionais. **Revista de Informática Teórica e Aplicada (RITA)**, v. 10, n. 4, p. 79-98, jan. 2004.

WOLF, D. F. et al. **Robótica Móvel Inteligente:** Da Simulação às Aplicações no Mundo Real**.** 1ª ed. Rio de Janeiro: SBC – Editora da PUC RIO. 2009.

SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. **Introduction to autonomous mobile robots**. 2nd. ed. USA: The MIT Press, 2011.

DUDEK, G.; JENKIN, M. **Computational Principles of Mobile Robotics**.

Cambridge, London, UK: The MIT Press, 2000.

ZUCATELLI, F; OLIVEIRA, M. **Controle de servomotores cc.** São Bernardo do Campo: 2007. Disponível em: < https://pt.scribd.com/doc/16300774/Controle-de-Servomotores-CC#download>. Acesso em: 12 dez. 2014.

CHENG TECHNOLOGY CO LTD. **KC2463: Semicondutor de circuito integrado para controle do servo**. Datasheet, 20--.

AHMED, A. **Eletrônica de Potência**.São Paulo: Prentice Hall, 2000.

EPUSP. **Controle de um servomotor.** São Paulo: 2014. Disponível em: <http://www2.pcs.usp.br/~labdig/pdffiles_2014/controle-servo-semestral.pdf>. Acesso em: 12 dez. 2014.

SEN, P.C.; **Principles of electric machines and power eletronics.** 2ºnd ed.
NOURBAKHSH, I. R.; SCARAMUZZA, D. **Introduction to autonomous mobile robots**. 2nd. ed. USA: The MIT Press, 2011.

SEN, P.C.; **Principles of electric machines and Power eletronics.** 2ºnd ed. Ontario: John Wiley & Sons, Inc. 1997.

E-GIZMO MECHATRONIX CENTRAL. **EGBT-046S: Bluetooth Modules.**Datasheet, 20--.