Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

Robô Guardião: Projeto de um robô hexápode de baixo custo

Cassiano Rogério Da Silva

Prof. Dr. Dalton Pedroso De Queiroz(Orientador)

Dourados -MS
Novembro de 2015

Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

Robô Guardião: Projeto de um robô hexápode de baixo custo

Cassiano Rogério Da Silva

Novembro de 2015

Banca Examinadora:

Prof. Dr. Dalton Pedroso De Queiroz (Orientador)
Área de Computação - UEMS

Prof. Me. Diogo Fernando Trevisan
Área de Computação - UEMS

Prof. Mª. Jéssica Bassani de Oliveira
Área de Computação - UEMS

Robô Guardião: Projeto de um robô hexápode de baixo custo

Cassiano Rogério Da Silva

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso II devidamente corrigida e defendida por Cassiano Rogério Da Silva e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 27 de novembro de 2015.

Prof. Dr. Dalton Pedroso De Queiroz
(Orientador)

Dedico este trabalho à minha família, por todo apoio e incentivo. Aos meus amigos pelo amparo nos momentos de dificuldade, pela compreensão devido minha ausência em nossos poucos encontros e momentos de descontração. Aos meus professores pela benevolência ao ensinar e acreditar em mim quando nem eu mesmo acreditei.

"É possível encontrar a felicidade nas horas mais sombrias, basta lembrar-se de acender a luz."

Dumbledore, Harry Potter e o Prisioneiro de Azkaban

Agradecimentos

Eu gostaria de agradecer ao Professor Doutor Dalton Pedroso De Queiroz, por todo apoio, incentivo e companheirismo, por todo o conhecimento que transmitiu durante todo o decorrer deste trabalho.

Aos meus pais por todo o incentivo, não me deixarem desistir de um sonho, por acreditaram em meu potencial.

A minha irmã que esteve sempre ao meu lado, com palavras de apoio, desejos de boa sorte e companheirismo.

A minha namorada por estar ao meu lado e me incentivar nos momentos mais complicados.

Aos irmãos que a vida me deu, esses que eu faço questão de chamar de amigos, quero agradecer por acreditarem em mim e estarem sempre ao meu lado, por entenderem (as vezes não) que nem sempre eu poderia estar presente em nossos encontros, mas que ainda sim meu desejo era de estar rindo junto a eles.

Aos colegas de graduação que fizeram parte desta jornada, muitos sabem que hoje são mais que colegas, são amigos que eu vou levar comigo a vida toda.

Aos professores que por todos estes anos estiveram se dedicando a ensinar, mesmo quando o mundo conspirava com as mais improváveis adversidades, desde uma simples chuva, ou até mesmo a falta de internet, afinal ela é de suma importância.

Estendo ainda meus agradecimentos à pessoas que não encontro já faz algum tempo mas que fizeram parte desta conquista, aos amigos da Ontec, aos amigos LASA.

A todos aqueles que me incentivaram e me apoiaram durantes o tempo que fiz parte da equipe da Universidade Federal da Grande Dourados, meu agradecimento amigos.

A todos os meus amigos do CENAIC que acompanharam todo meu trabalho, e sorrindo e oferecendo sorrisos e palavras de apoio quando eu parecia ter vontade de desistir.

Agradeço à pessoas que nunca encontrei pessoalmente mas que com palavras, ações e com conteúdo me ajudou a manter a sanidade durante este período, obrigado a equipe Jovem Nerd.

Agradeço aos amigos do portal TCCendo que tanto ajudaram, com conteúdo e histórias de vitórias de tantos que já passaram por esta fase.

A esta instituição que faz parte da minha vida, obrigado Universidade Estadual de Mato Grosso do Sul, por me permitir desenvolver este projeto que um dia foi um sonho e

hoje se tornou realidade.

Obrigado a todos, pelo apoio e companheirismo.

Resumo

Neste trabalho foi desenvolvido um robô hexápode e seu *software* de controle. Sua estrutura foi confeccionada a partir de placas de PVC e o seu centro de comando foi baseado na plataforma Arduino. O objetivo desse projeto foi desenvolver um robô de baixo custo que poderá servir como um elemento estrutural que possa suportar e dar mobilidade a aparelhos de monitoramento e segurança, tais como, câmeras, sensores de presença, temperatura, etc. Apesar de existirem no mercado de sistemas de segurança soluções para esse tipo de transporte, elas são geralmente de alto custo e proprietários, então, o Robô Guardião, como foi chamado, pretende ser um projeto de qualidade, baseado em *Open Hardware* e com baixo custo.

Palavra-chave: Arduino, Robótica, Hexápode.

Sumário

Lista de Siglas

DSL Domain Specific Language.

PVC Polyvinyl Chloride que em português significa Policloreto de polivinila (ou policloreto de vinil).

Mhz Mega Hertz.

C Linguagem criada por Dennis Ritchie em 1972.

LED Light Emitting Diode.

IDE Integrated Development Environment ou Ambiente Integrado de Desenvolvimento.

USB Universal Serial Bus.

V Volt.

GPS Global Positioning System - Sistema de Posicionamento Global.

UFMS Fundação Universidade Federal de Mato Grosso do Sul.

CM Centímetros.

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Introdução

Atualmente estamos cada vez mais ligados e conectados. O tempo todo rodeados de equipamentos tecnológicos com as mais diversas finalidades. Nesse ambiente é que este projeto se insere: o Robô Guardião tem como proposta ser um elemento integrador que possa aliar a mobilidade aos modernos equipamentos de monitoramento e segurança, tais como, câmeras e sensores para as mais diversas finalidades.

Ele pretende ser uma plataforma base de uma estrutura ainda maior, como um sistema completo de monitoramento, transporte de equipamentos, ou exploração de ambientes.

Seu desenvolvimento se deu a partir de tecnologias e equipamentos modernos, de baixo valor, porém com confiabilidade e qualidade.

1.2 Objetivos

O objetivo desse projeto foi construir um robô hexápode de baixo custo a partir de placas de PVC para formar sua estrutura física e ter seu controle baseado na plataforma Arduino que possa ser usado como o elemento de mobilidade de um sistema de segurança e monitoramento.

1.3 Motivação

É comum pensar que algo que facilite o dia a dia das pessoas só pode ser criado fora de terras nacionais e que chegará em nossas mãos apenas após ser importado e em geral com um valor elevado, uma das ideias é que o projeto possa entregar uma ferramenta de baixo

custo e com um valor agregado baixo, tornando o projeto um referencial de tecnologia e valor reduzido.

A plataforma Arduino foi escolhida para controlar todo o sistema que integra o robô Guardião.

A placa Arduino desempenha a função de processador central de todo o sistema, trata-se de uma plataforma de fácil acesso com uma linguagem de programação relativamente simples (para alguém que já tenha tido contato com alguma linguagem de programação), sua linguagem descende da Linguagem C em padrões e termos, a programação é realizada de forma estruturada.

1.4 Metodologia

Para o desenvolvimento deste projeto foram realizadas diversas pesquisas acerca dos componentes que deveriam ser utilizados, como interligá-los e controla-los.

Foram construídos moldes através de desenhos e modelos no softwares *fritzing*, o desenvolvimento final foi obtido através da aplicação prática das pesquisas.

1.5 Organização do trabalho

O trabalho foi dividido em cinco capítulos e dois apêndices, onde cada um dos capítulos aborda de forma detalhada os aspectos que permeiam o projeto.

No primeiro capítulo pode-se encontrar a descrição do projetos e suas motivações.

O capítulo dois aborda as tecnologias que foram estudadas e utilizadas para o desenvolvimento do projeto.

O capítulo três demonstra como foi desenvolvido o *hardware* do projeto, desde os desenhos inicias de cada uma das peças, até o modelo eletrônico para realizar a conexão dos componentes do projeto.

No capítulo quatro é abordada a linguagem do Arduino, e quais funções foram utilizadas para o desenvolvimento do *software* de controle do projeto.

O capítulo cinco demonstra os resultados que foram obtidos com o desenvolvimento do robô guardião.

Capítulo 2

Conceitos Teóricos

Neste capítulo serão abordados os conceitos e recursos que foram utilizados para o desenvolvimento do projeto, tais como a robótica, monitoramento e Arduino.

2.1 Robótica

O termo robótica é derivado da palavra robô, esta mesma palavra tem origens antigas, ela foi utilizada pela primeira vez no ano de 1920 na Tchecoslováquia, conforme Carrara (2009).

> O precursor do termo robô (Groover, 1988) foi Karel Capek, novelista e escritor de uma peça teatral da Tchecoslováquia, que usou pela primeira vez, em 1920, a palavra "robota" (serviço compulsório, atividade forçada) originando a palavra "robot" em inglês e traduzido para o português como "robô"."

A robótica é uma parte das ciências exatas destinadas ao estudo e desenvolvimento de dispositivos autônomos ou semi-autônomos para auxiliar o ser humano no dia a dia, a proposta de desenvolvimento de dispositivos autônomos/semi-autônomos vem de encontro com diversos problemas já conhecidos pelo ser humano, como, falta de tempo, tarefas que podem causar risco de vida entre outros.

É complexo definir o que é um robô devido as suas diversas variações e aplicabilidade, entretanto nota-se aspectos/componentes comuns entre as diversas variações encontradas no cenário tecnológico atual.

de Souza (2005) define alguns desses aspectos/componentes encontrados nos robôs como.

- sistemas de locomoção;
- sensores;
- sistemas de processamento;

Ao observarmos projetos que envolvam robótica é fácil perceber os itens acima citados, um sistema de locomoção é algo primordial para a maioria dos robôs de modo a permitir que eles possam se locomover entre o ponto A e o ponto B ou mover algo entre o ponto A e o ponto B, para que tal locomoção possa ser executada também faz-se necessário a interação com o ambiente para que o robô possa saber quais obstáculos ele irá encontrar a sua frente, mas isto não é tudo, é necessário também ao robô entender e criar uma ação relacionada as leituras feitas, para tal é preciso um sistema de processamento, de modo que ele possa processar os dados coletados pelos sensores e tomar uma decisão.

2.2 Monitoramento

Monitoramento se refere ao ato de obter informações de um sistema, a partir dos dados gerados pelos mesmos.

A palavra pode ainda ter um sentido mais amplo, citando caso análogo onde ela é empregada quando é abordado segurança empresarial ou doméstica, quando tem-se câmeras, sensores térmicos, de presença entre outros, todo este conjunto é chamado de sistema de monitoramento.

A palavra monitoramento é derivada da palavra monitorar, segundo de Holanda Ferreira (2004) monitorar significa "Acompanhar e avaliar (dados fornecidos por aparelhagem técnica)." que é derivada da palavra monitor, de Sousa (2013) diz que: "A palavra monitor vem do latim: *monitum* e significa "aquele que dá conselho, que faz pensar, que adverte, que lembra".

Partindo das definições nota-se que processo de monitoramento de um ambiente ou equipamento pode ser longo e cansativo, principalmente para um ser humano, sem levar em consideração o custo de se manter mão de obra humana, para que se possa capturar as informações, sejam de um equipamento ou ainda observando um ambiente.

Mesmo demandando tempo e custos, faz-se necessário nos dias atuais algum sistema de monitoramento seja no âmbito domiciliar, empresarial ou público para que possa-se garantir a segurança dos que ali convivem, um sistema de monitoramento vai muito além de registar e acompanhar fatos e/ou acontecimentos, pode afastar invasores.

A partir deste conceito é inserido a robótica, de modo a oferecer recursos tecnológicos de modo a entregar equipamentos modernos que possam suprir as necessidades do ambiente,

neste exemplo, o monitoramento de um ambiente, principalmente quando o referido ambiente é dito insalubre para um ser humano.

2.3 Arduino

A plataforma Arduino é relativamente nova, sua criação é datada de 2005, quando um professor desejava ensinar eletrônica e programação para seus alunos, porém se deparou com a dificuldade de ensinar tais conceitos para alunos que não eram propriamente da área.

Conforme artigo do Grupo de Robótica (2012) da UFMS.

> "O Arduino surgiu em 2005, na Itália, com um professor chamado Massimo Banzi, que queria ensinar eletrônica e programação de computadores a seus alunos de design, para que eles usassem em seus projetos de arte, interatividade e robótica."

Não existiam plataformas que fossem simples e baratas para que pudessem ser utilizadas durante suas aulas, por este motivo ele criou uma ferramenta que fosse simples porém poderosa, nasceu assim a plataforma Arduino.

2.3.1 Definição

O Arduino é uma placa de controle de dados, entrada*(in)* e saída*(out)*, gerenciada por um microcontrolador *Atmel*[1] *AVR*, nesta plataforma podemos gerenciar, servo-motores, LEDs, sensores e outros componentes eletrônicos para as mais diversas finalidades.

Conforme artigo do Grupo de Robótica (2012) da UFMS.

> "Arduino é uma placa de controle de entrada de dados (IN), como sensores, e saída de dados (OUT), como motores e LEDs... A porta USB já fornece alimentação enquanto estiver conectado ao computador, e a tensão de alimentação quando desconectado pode variar de $7V$ a $12V$."

[1] *Atmel Corporation é uma manufaturadora de semicondutores, fundada em 1984.*

McRoberts (2011) diz que:

> "...um Arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele."
> "O Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software."

A plataforma Arduino possibilita a integração de diversos componentes eletrônicos, que a primeira vista podem parecer não se conectarem, mas graças as suas portas de comunicação torna está realidade possível.

McRoberts (2011) diz que:

> "O Arduino pode ser conectado a LEDs, *displays* (mostradores) de matriz de pontos, botões, interruptores, motores, sensores de temperatura, sensores de pressão, sensores de distância, receptores GPS, módulos *Ethernet* ou qualquer outro dispositivo que emita dados ou possa ser controlado."

A ideia por traz da tecnologia embarcada, remete a produtos já consagrados no nosso dia a dia como micro-ondas, televisores entre outros, são sistemas estes que tem placas controladoras com funções similares as do Arduino, em alguns casos limitadas ou superiores em alguns aspectos quando comparamos com uma placa Arduino.

Segundo Santos (2006):

> "Estamos cercados por sistemas embarcados, eles estão cada vez mais presentes em nosso dia-a-dia, máquinas de lavar, televisões, eletrodomésticos em geral possuem algum tipo de processamento, automóveis, caixas de banco eletrônicos, equipamentos de comunicação como modems, roteadores, etc."

Conforme a Figura 2.1 observamos que sua estrutura é composta por componentes eletrônicos, em sua grande maioria encontrados com facilidade em casas de produtos eletrônicos.

O mesmo é composto por um cristal oscilador de *16Mhz*, um regulador de tensão de *5V*, um botão *reset*[2], um plugue de alimentação, pinos de entrada e saída de dados, e alguns *LED*s que auxiliam na verificação do seu funcionamento e também uma porta de conexão *USB*. Este dispositivo pode ser alimentado com uma fonte de energia que pode variar de *7V* a *12V*, ou ainda quando conectado pela porta *USB* utiliza desta conexão para ser alimentado com *5V*.

Figura 2.1: Arduino Mega 2560 R3 (Imagem do autor).

[2]Reiniciar

2.3.2 Utilização

O Arduino tem se mostrado uma plataforma bastante versátil, prática e de fácil utilização exigindo poucos conhecimentos na área eletrônica por parte de seu utilizador, no entanto isso não o impede de ser utilizado em grandes projetos.

Conforme artigo do Grupo de Robótica (2012) da UFMS.

> "Os desenvolvedores do Arduino tentam manter sua linguagem fácil de usar para iniciantes, mas flexível o bastante para usuários avançados."

McRoberts (2011) diz que:

> "A maior vantagem do Arduino sobre outras plataformas de desenvolvimento de microcontroladores é a facilidade de sua utilização; pessoas que não são da área técnica podem, rapidamente, aprender o básico e criar seus próprios projetos em um intervalo de tempo relativamente curto."

O Arduino traz junto de si a proposta de popularizar e facilitar o desenvolvimento de diversos tipos de projetos, tanto de grande porte quanto os de pequeno e médio porte.

Conforme McRoberts (2011):

> "O hardware e o software do Arduino são ambos de fonte aberta, o que significa que o código, os esquemas, o projeto etc. podem ser utilizados livremente por qualquer pessoa e com qualquer propósito."

O aspecto mais importante ainda fica por conta do fato de ser um plataforma de fácil utilização, isso o torna um "marco"da tecnologia, no sentido de que ele viabiliza a possibilidade de pessoas sem conhecimento técnico profundo da área poderem realizar pequenos projetos sem grandes problemas e custos.

Conforme McRoberts (2011):

> "Desde que o Arduino Project teve início em 2005, mais de 150.000 placas Arduino foram vendidas em todo o mundo.
> Sua popularidade não para de crescer, e cada vez mais pessoas percebem o incrível potencial desse maravilhoso projeto de fonte aberta para criar projetos interessantes rápida e facilmente, com uma curva de aprendizagem relativamente pequena."

Segundo matéria publica pelo site Olhar Digital (2010) o Arduino tem sido aplicado em diversos projetos de automação residencial e comercial, projetos esses que vão desde um simples controle de acendimento de luzes, escadas que foram transformadas em teclas de piano ou um alimentador automático para um animal de estimação.

McRoberts (2011) diz que:

> "O Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou até mesmo à Internet para recuperar e enviar dados do Arduino e atuar sobre eles."

É notado que a plataforma tem sido utilizada para diversos fins, tal situação é dada visto o fato de não se tratar de uma plataforma criada com fins específicos, e ainda de bônus ter diversas versões no mercado, desde valores mais baixos, com recursos limitados, a versões com valores um pouco mais altos, porém com muito mais recursos que outras versões.

Dada as configurações "abertas" (plataforma que permite que seja adequada a natureza de diversos problemas) da plataforma, e visto que ela pode ser adequada a praticamente todo projeto que se possa imaginar, fazendo-se necessário apenas ajustes e aquisição de componentes específicos para realização do projeto que se quer desenvolver, estas características tornam a plataforma tão adequada e utilizada nos mais diversos tipos de problemas, projetos e necessidades.

A placa controladora Arduino pode ser facilmente integrada em um projeto em andamento ou utilizada para controlar toda a estrutura de um novo projeto que será desenvolvido, por este motivo é amplamente utilizada em projetos que envolvem robótica, com ela é possível controlar a intensidade de luz de um LED, o giro de um servo motor ou ainda fazer a leitura de um sensor.

Por essas características o Arduino é uma plataforma viável para a robótica permitindo a utilização de sensores para "visualização" do ambiente onde o projeto estará inserido e assim poder interagir com o mesmo, através de seus componentes como servo motores, LEDs, entre outros.

2.4 Lixo eletrônico

O lixo eletrônico é uma realidade da nossa sociedade nos dias atuais, devido aos constantes lançamentos de novas tecnologias os dispositivos se tornam obsoletos muito rapidamente, existem algumas possibilidades para que se possa dar destino para estes matérias, uma delas é a reutilização em projetos como o robô guardião.

Favera (2008) diz que:

> "Lixo eletrônico é definido como sendo todos os resíduos resultantes da rápida obsolescência de equipamentos eletro-eletrônicos.
> Nestes estão incluídos aparelhos compostos quase que totalmente por circuitos eletrônicos como televisores, celulares, computadores, ... mas também estão incluídos equipamentos eletrodomésticos que possuem alguma parte eletro-eletrônica."

Desse modo diversos componentes podem ser re-aproveitados como é o caso das carcaças de gabinetes que são utilizadas para construir o corpo do robô.

O uso do lixo eletrônico permite reduzir o custo final do projeto visto que, alguns materiais que seriam descartados podem ser re-aproveitados, como é o caso das carcaças, *LEDs* e parafusos.

Capítulo 3

Desenvolvimento do Hardware

Neste capítulo serão abordadas algumas das técnicas, instrumentos, componentes e métodos que foram adotadas para o desenvolvimento do projeto.

O projeto visou antes de mais nada a utilização de componentes de baixo custo e de fácil acesso, para que posteriormente, quando necessitar de manutenção, a mesma possa ser realizada com facilidade.

O projeto teve como base estrutural o projeto desenvolvido pelo francês Laurent Suchier (2012), fundador da página "http://www.syris.fr/", página criada com a finalidade de divulgar seus trabalhos.

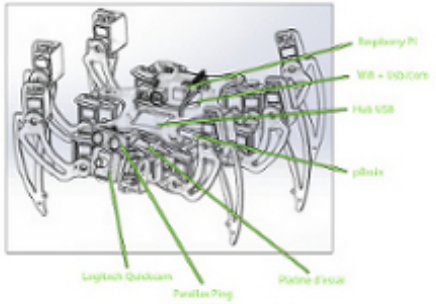

Figura 3.1: Hexapod Raspberry - Laurent Suchier (2012)

3.1 Proposta e materiais

Este projeto tem como proposta a economia, uma vez que projetos que partilham da ideia do Robô Guardião costumam apresentar alto custo.

Os componentes que foram utilizados para o projeto tiveram um valor aproximado de R$ 248,00 enquanto a média de projetos similares pesquisados foi de R$ 1.577,20, o custo do projeto é de aproximadamente 16% do valor médio de projetos similares, a tabela completa de valores pode ser consultada no capítulo cinco.

Toda a estrutura, isto é, o "esqueleto do robô", foi desenvolvido tomando como base canos de PVC (os mesmos utilizados na construção civil). Esse material foi escolhido por conta do seu baixo custo e por suas propriedades mecânicas, que se adéquam perfeitamente ao que se pretendia fazer em termos de estrutura para o robô. Também foram utilizadas carcaças de gabinetes de computadores obsoletos e outras peças e parafusos dos mesmos.

Dos materiais pesquisados, o PVC foi o que mais se adequou a proposta do projeto (muitos projetos fazem uso de placas de alumínio e/ou carbono como o projeto de Suchier (2012)), por tratar-se de um material maleável, com baixo custo e de fácil acesso.

O PVC, ao ser aquecido torna-se um material "mole" podendo ser comparado à um pedaço de couro, o que o torna perfeito para ser moldado de acordo com as necessidades do projeto. Ao esfriar ele mantêm a forma que apresentava enquanto estava quente.

Além do PVC fez-se necessário o uso de metal para revestir algumas das peças que compunham o esqueleto, para que estas peças apresentassem a estabilidade necessária, o metal também foi utilizado na construção peças para acomodar os servo motores[1] e interligar algumas estruturas.

Esse tipo de fabricação da estrutura do robô teve como premissa a utilização de um material barato e resistente, assim como a utilização de refugos de computadores. Essa proposta de barateamento de estruturas robóticas tem sido buscado por muitos, como por exemplo, pelo francês Suchier (2012).

3.2 Desenvolvimento de um modelo de molde

Dadas as propriedades do PVC, o mesmo foi cortado na vertical para que ele pudesse ser aberto e aquecido, tornando-se maleável. Essa maleabilidade conseguida com o aumento da temperatura possibilitou que o cano de PVC fosse remodelado na forma de uma chapa. Foi então deixada a mesma por alguns minutos para esfriar e passou-se a desenhar sobre ela o esqueleto do robô para ser posteriormente recortado.

Vários moldes foram desenvolvidos até que se pudesse chegar a um que contemplasse as necessidades estruturais do projeto. Na Figura 3.2 é possível observar alguns desses moldes

[1] São dispositivos que rotacionam seu eixo com precisão para mover algo.

e seu processo de desenvolvimento.

A partir dos vários testes feitos, observou-se que a melhor estrutura seria aquela onde os servo motores ficassem próximos uns dos outros e a placa central de controle do robô, que serviria de suporte para o Arduino, tivesse um tamanho reduzido de modo a não possibilitar qualquer flexão da estrutura.

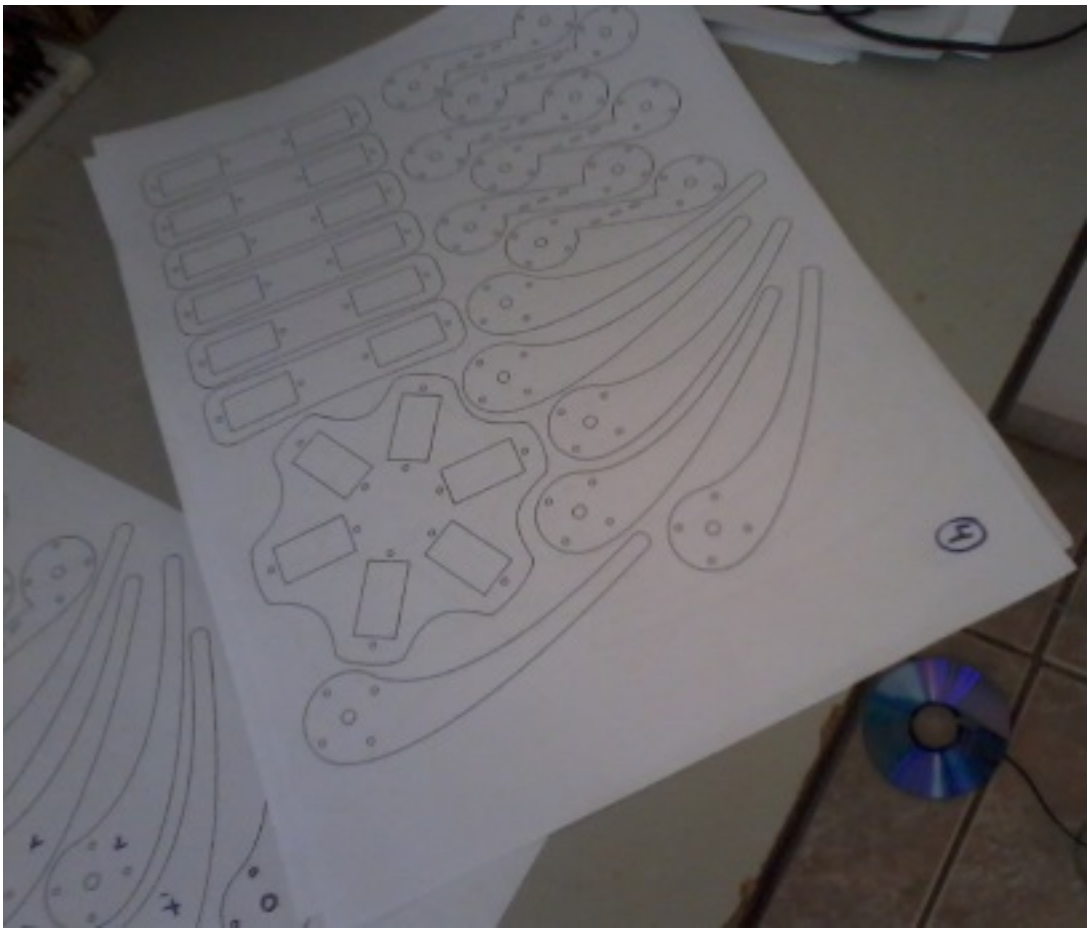

Figura 3.2: Molde inicial (Imagem do autor).

3.3 Criação do modelo do esqueleto

Inicialmente foi tomado como base a estrutura proposta no projeto francês, conforme Suchier (2012) - Figura 3.1. Porém, constatou-se que a mesma não atendia satisfatoriamente as necessidades do Robô Guardião, então, foram feitas todas as adequações necessárias para que essas necessidades fossem atendidas, conforme os objetivos delineados para o mesmo.

Pode se observar na Figura 3.3 a união das peças do primeiro protótipo de esqueleto do robô, porém o primeiro protótipo continha falhas estruturais que foram corrigidas, falhas essas que não permitiam a perfeita conexão das peças que compunham o projeto, e ainda o perfeito acomodamento dos componentes eletrônicos, sendo assim foi iniciado um novo desenho de molde em uma placa de PVC, afim de suprir necessidades que o modelo inicial não atendia.

Após o inicio das adequações fazia-se necessário a transcrição das adaptações novamente para a placa de PVC afim de se obter novas peças para dar inicio a montagem do segundo esqueleto, pode se observar o processo na Figura 3.4.

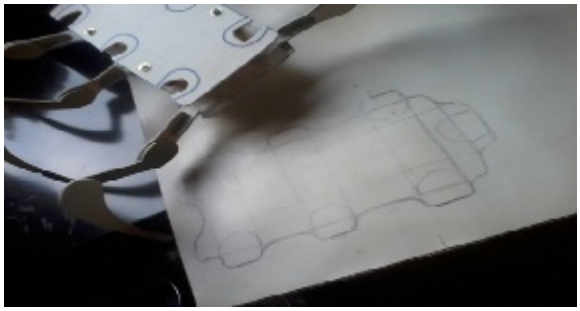

Figura 3.3: Adequação do molde inicial (Imagem do autor).

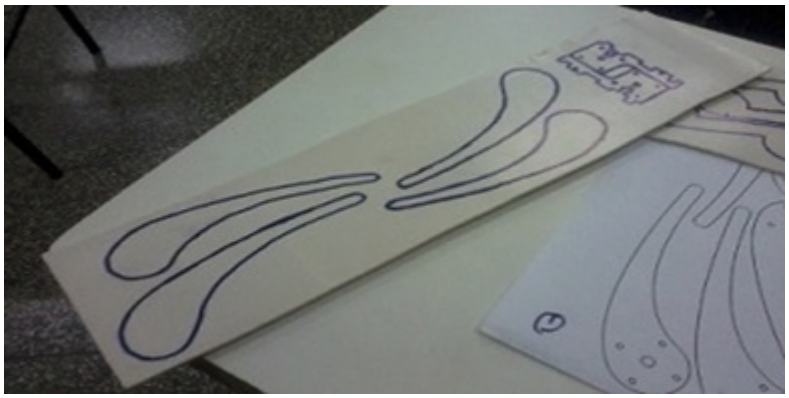

Figura 3.4: Transcrição do molde para a placa de PVC (Imagem do autor).

3.3.1 Montagem das partes que compõem o modelo do esqueleto

O molde criado para ser o corpo do projeto tem as seguintes dimensões 18CM x 9CM e 4CM de distância entre os servo motores, conforme a Figura 3.5.

Após definido o molde central e seu tamanho, foram criadas as articulações que são denominadas “coxas”, que interligam a ponta da pata ao corpo do robô, conforme Figuras 3.6 e 3.7.

Em seguida foram confeccionadas as pontas das patas que são denominadas “tíbias”, que são interligadas às “coxas”.

Estás são compostas por uma parte metálica e outra de PVC, o metal garante que a peça não fique flexível, caso contrario as “tíbias” não suportariam o peso do projeto, conforme pode ser observado na Figura 3.8.

Foi ainda adicionado na base da “tíbia” um base elíptica de modo que a área de contato com o piso se tornasse maior, a mesma ainda foi revestida de material emborrachado para que ele não derrape em alguns tipos de pisos.

Conforme pode ser visto na Figura 3.9 a articulação da “coxa” foi montada interligada a “tíbias” e ao corpo do projeto.

Foi criado um suporte para o encaixe do sensor ultrassônico, feito com parafusos de elevação utilizados em gabinetes de computador, o mesmo foi parafusado na estrutura do projeto, conforme pode ser observado na Figura 3.10.

Após a correta verificação do tamanho das peças, é feita a montagem do projeto.

Observa-se na Figura 3.11 uma prévia da estrutura final do projeto onde temos conectados os servo motores.

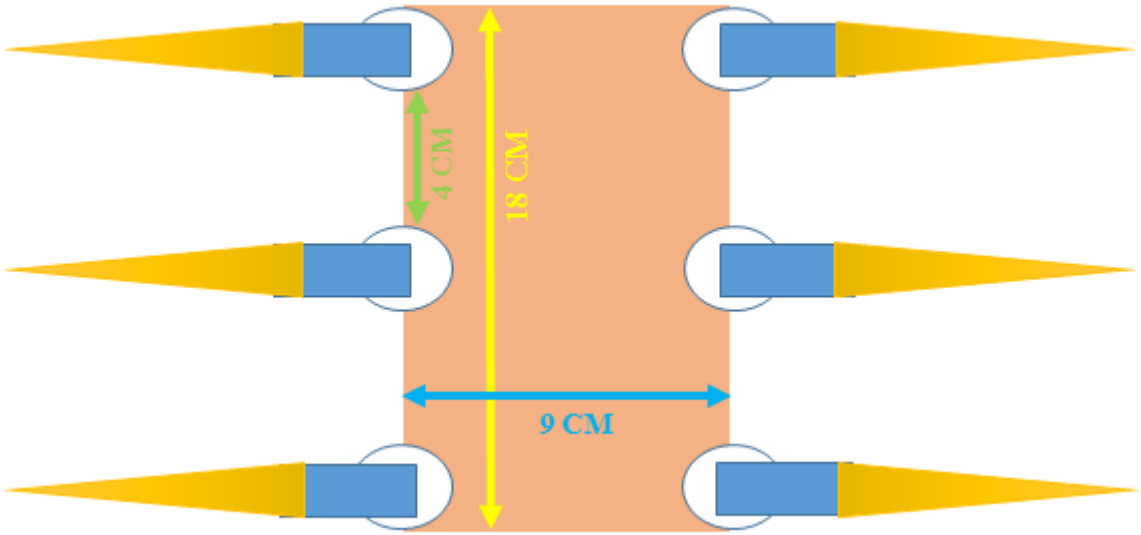

Figura 3.5: Molde central do projeto (Imagem do autor).

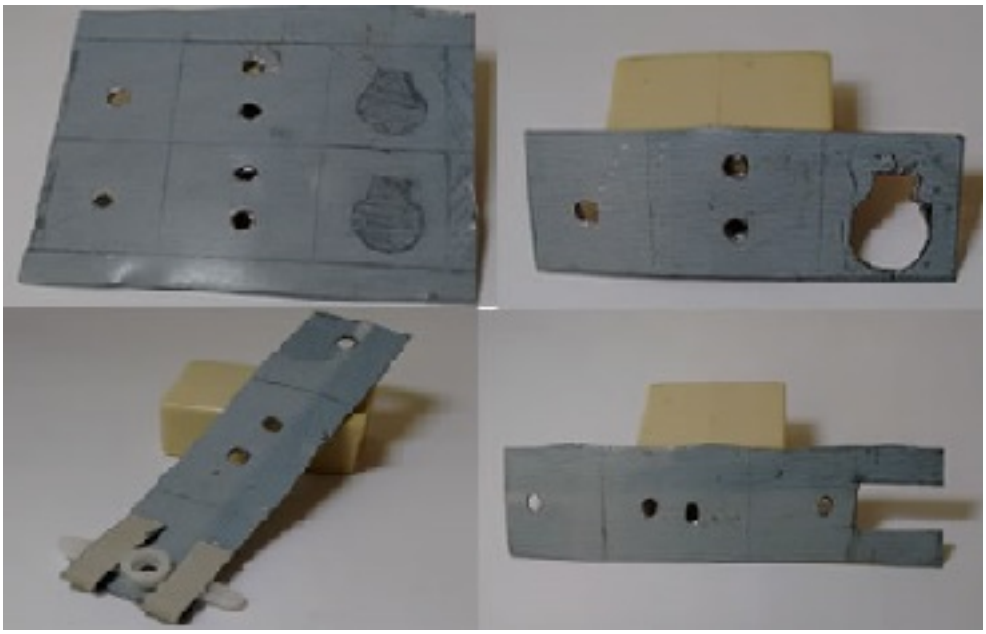

Figura 3.6: Articulação que compõe a coxa do projeto (Imagem do autor).

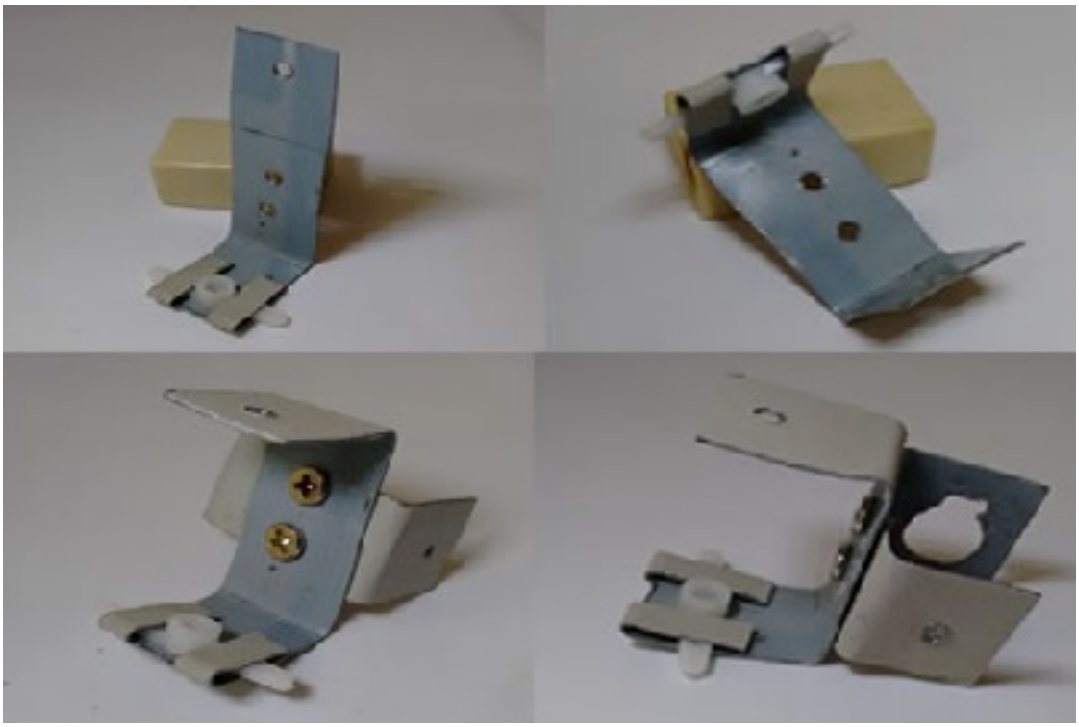

Figura 3.7: Articulação que compõe a coxa do projeto 2 (Imagem do autor).

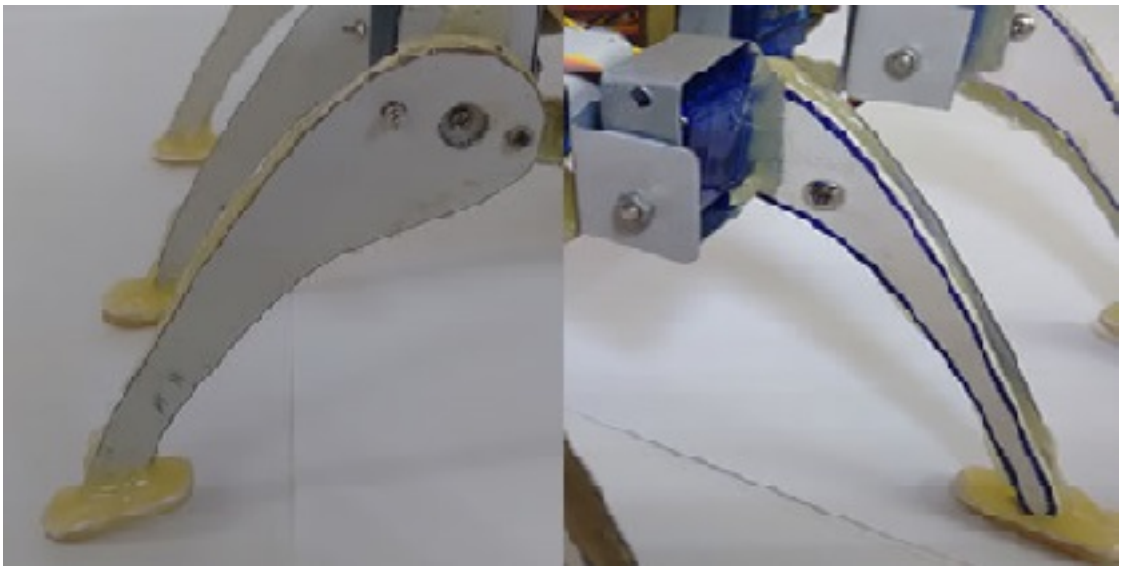

Figura 3.8: Tíbia composta por PVC e metal (Imagem do autor).

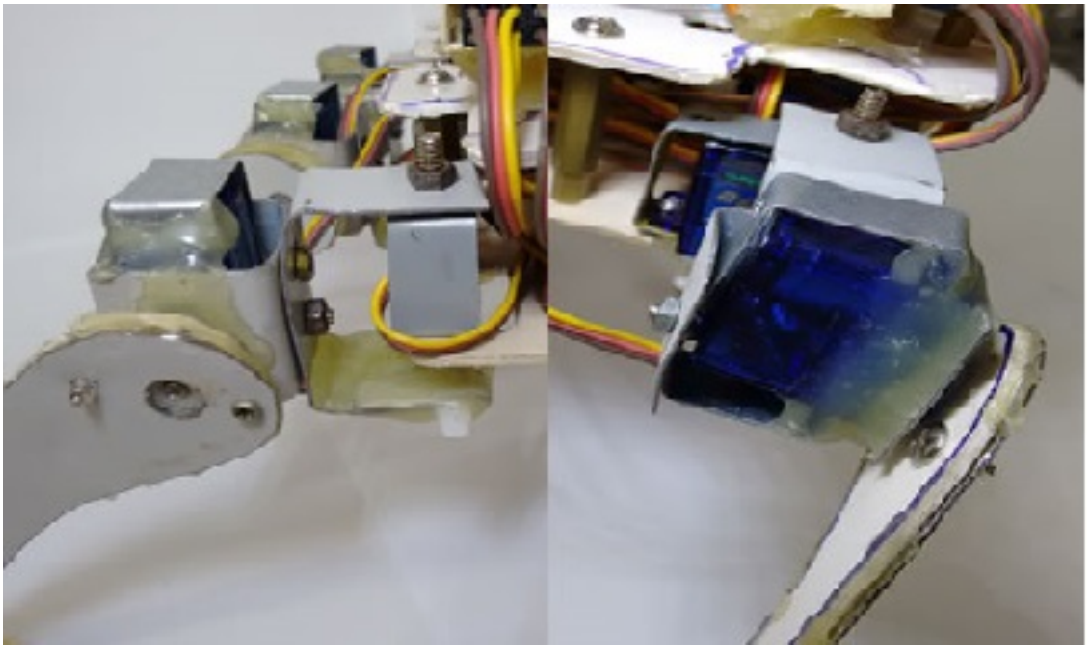

Figura 3.9: Articulação da coxa montada (Imagem do autor).

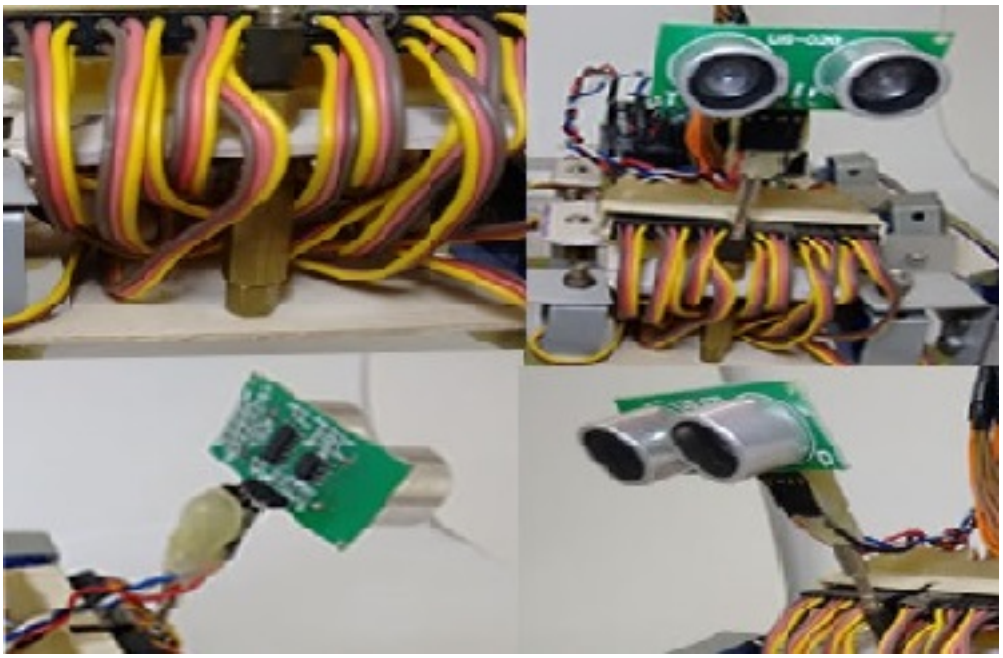

Figura 3.10: Suporte para o sensor ultrassônico (Imagem do autor).

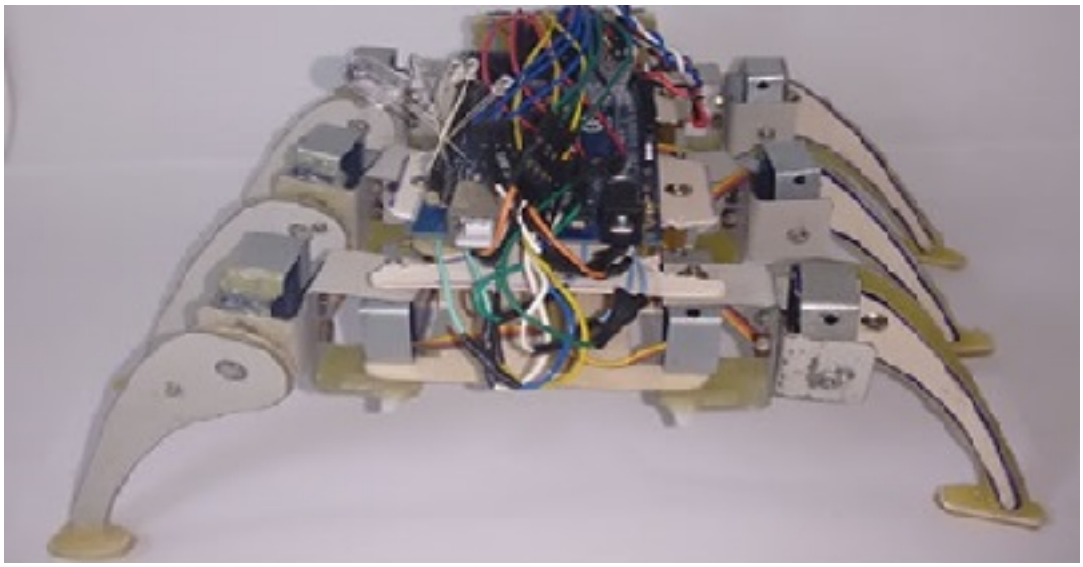

Figura 3.11: Projeto montado (Imagem do autor).

3.4 Esquema Eletroeletrônico

A seguir são apresentados os esquemas elétricos e eletrônicos que foram desenvolvidos para demonstrar a conexão dos componentes e a placa controladora Arduino.

A Figura 3.12 demonstra como foi realizada a montagem eletrônica dos componentes que compõe o robô desenvolvido.

A Figura 3.13 apresenta o esquema de conexão elétrica do robô guardião.

Os esquemas a seguir foram produzidos com o *software fritzing*, nos esquemas podemos encontrar os componentes a seguir:

- 1 Arduino Mega 2560 R3
- 12 Servo-Motores
- 1 Sensor ultrassônico
- 1 Protoboard[2] (Presente somente no esquema eletrônico)
- 1 Fonte de alimentação

[2]Também conhecida como matriz de contado, é uma placa que permite criar circuitos experimentais.

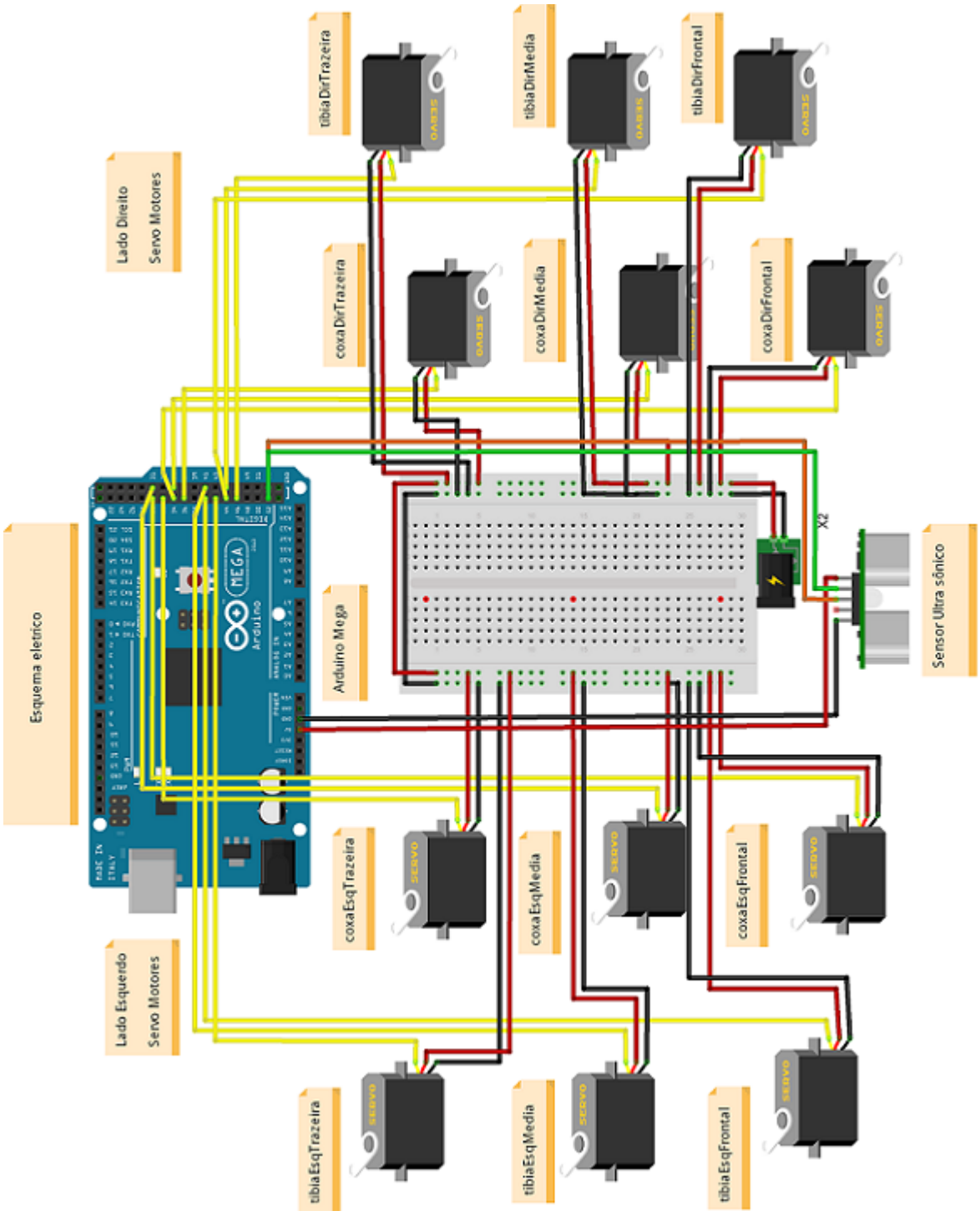

Figura 3.12: Esquema eletrônico (Imagem do autor).

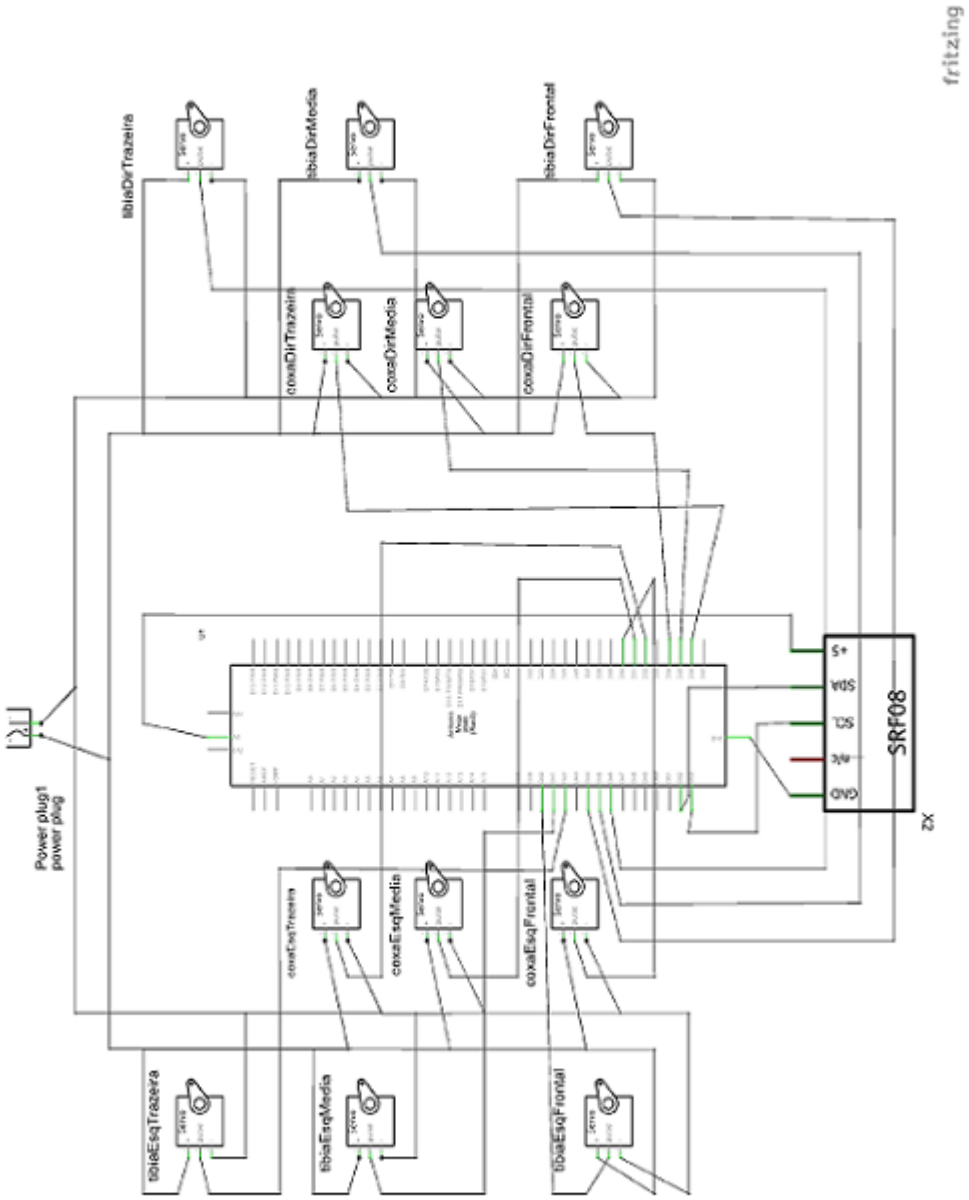

Figura 3.13: Esquema elétrico (Imagem do autor).

3.5 Componentes

Para o desenvolvimento do projeto foi necessário a utilização de alguns componentes eletrônicos tais como servo motores e sensor ultrassônico.

Servo Motor

Servo motor é um componente eletrônico, composto de um pequeno motor, um pequeno circuito para controlar o motor e algumas engrenagens, que permite a movimentação de projetos que envolvem principalmente robótica, eles permitem a movimentação de patas de robôs do tipo inseto por exemplo.

Conforme de Matos (2012).

> "Os servomotores actualmente incluem-se na categoria de motores especiais, sendo considerados motores de precisão podendo ser aplicados nas mais diversas actividades, das quais se destacam as seguintes:
>
> - Sistemas de Posicionamento;
> - Robótica Industrial;
> - Linhas de Transporte;
> - Máquina-Ferramenta a comando manual;
> - Sistemas flexíveis de manufactura;"

Sensor Ultrassônico

A proposta ainda visa utilização de sensores ultrassônicos que são os "olhos" do robô, o que permitirá evitar colisões.

Segundo artigo do site Newton C. Braga (2012) podemos comparar o sensor ultrassônico como o sonar de um morcego.

> "O princípio de operação desses sensores é exatamente o mesmo do sonar, usado pelo morcego para detectar objetos e presas em seu vôo cego."

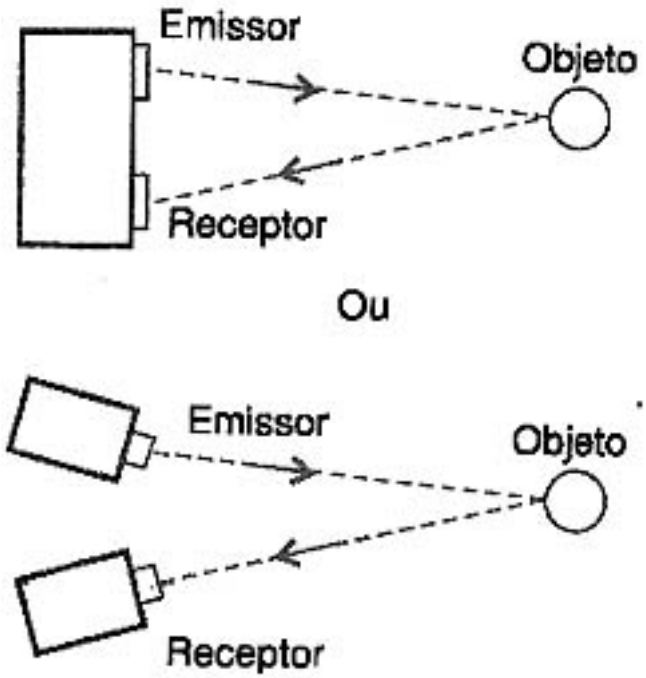

Figura 3.14: Funcionamento sensor ultrassônico segundo Newton C. Braga (2012)

Segundo Wendling (2010).

> " O princípio de funcionamento desse sensor é o seguinte: um oscilador emite ondas ultrassônicas, que resultam em um comprimento de onde na ordem de alguns centímetros, o que permite detectar objetos relativamente pequenos. As ondas refletidas pelo objeto são captadas pelo sensor, fornecendo assim um sinal que pode ser processado trazendo informações sobre o objeto no qual ocorreu a reflexão."

Newton C. Braga (2012) descreve ainda o funcionamento de um sensor ultrassônico, e de suas ondas a seguir.

> "...o pequeno comprimento de onda das vibrações ultrassônicas faz com que elas reflitam em pequenos objetos, podendo ser captadas por um sensor colocado em posição apropriada."

A Figura 3.14 demostra como é o funcionamento da onda enquanto emitida e refletida pelo objeto, até chegar no seu receptor.

O sensor ultrassônico permite ler o ambiente de modo a perceber a distância que se dá do sensor até o objeto que se encontra a frente, desta forma é possível interagir com o ambiente afim de evitar colisões.

Capítulo 4

Implementação do Software

4.1 Introdução

Todo o sistema de controle do robô foi desenvolvido na plataforma Arduino. Essa plataforma utiliza uma linguagem de programação própria, que se enquadra na categoria DSL e leva o mesmo nome da plataforma: "Linguagem de Programação do Arduíno".

Esta linguagem se inspira na linguagem C, por este motivo pode-se observar que, praticamente todos os recursos da linguagem C estão disponíveis para serem utilizados na linguagem do Arduino.

Segundo McRoberts (2011).

> "Para programar o Arduino (fazer com que ele faça o que você deseja) você utiliza o IDE do Arduino, um software livre no qual você escreve o código na linguagem que o Arduino compreende (baseada na linguagem C)."

Pode-se encontrar recursos como testes condicionais simples(*if else*), laços de repetições (*for, while, do while*), testes condicionais mais complexos (*switch case*), conforme a página de referência da plataforma Arduino, arduino.cc (2014).

A plataforma Arduino disponibiliza ainda através de sua página a IDE oficial da plataforma, Figura 4.1, dedicada para o desenvolvimento de códigos para o Arduino.

O *software* Arduino, tem versões para os sistemas Linux[1], *Mac OS*[2] e *Windows*[3]e a utilização da plataforma é indica pelo fato de ter total integração com a plataforma, trazendo consigo os *drivers*[4] necessários para a comunicação do computador com a placa Arduino.

[1]Sistema operacional de código livre.
[2]*Macintosh Operating System*: Sistema Operacional desenvolvido pela *Apple*.
[3]Sistema operacional desenvolvido pela *Microsoft*.
[4]*Software* que permite que um computador se comunique com algum hardware ou equipamento.

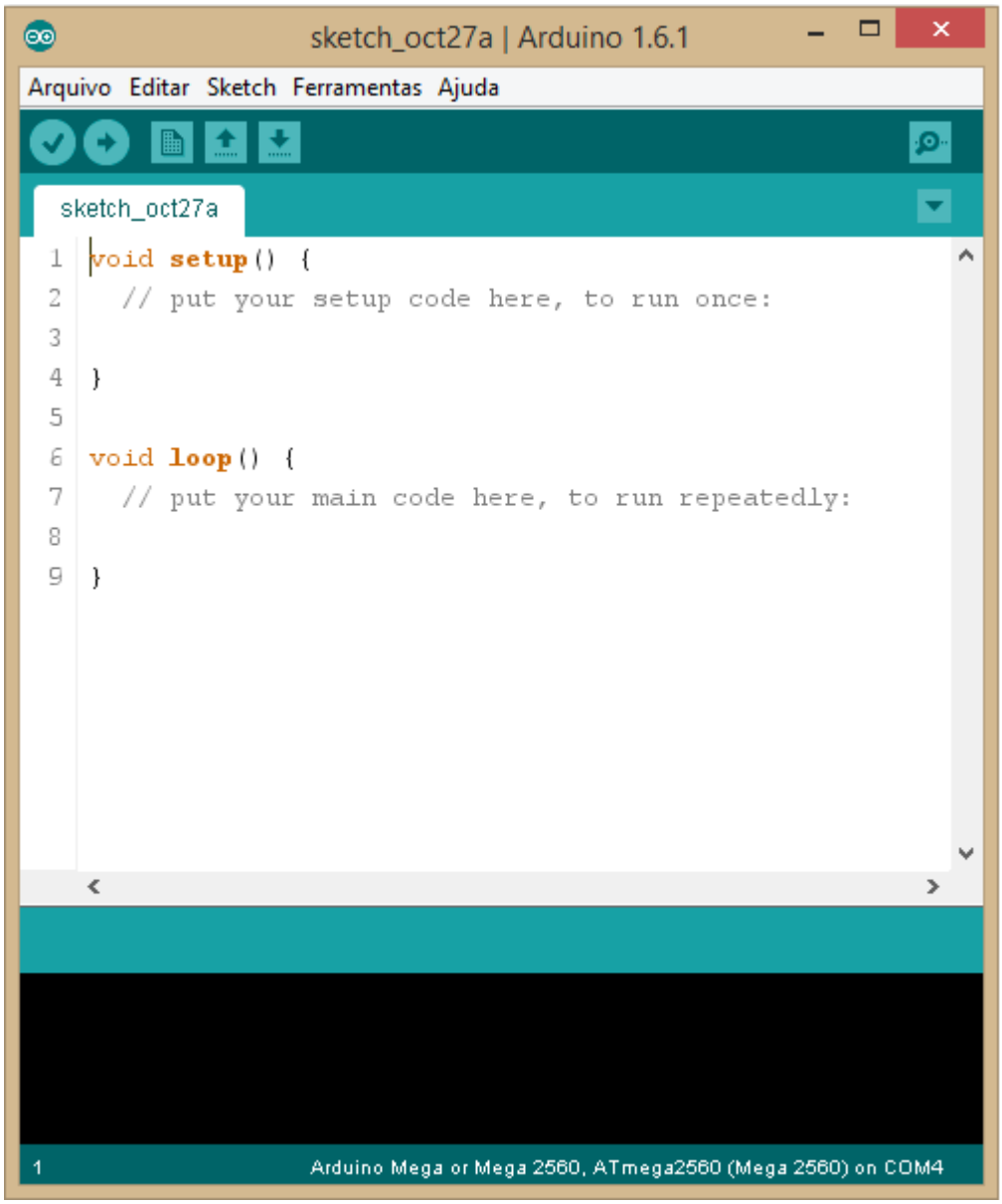

Figura 4.1: IDE Arduino (Imagem do autor)

4.2 Linguagem Arduino e suas funções

Para que a placa Arduino reconheça os códigos desenvolvidos e funcione de forma correta ela necessita de duas funções básicas que são elas, *"void setup()"* e *"void loop()"*, Gaier (2011) descreve suas funções da seguinte forma.

> "A função *void setup()* funciona como uma "função de inicialização", o que estará contido nela rodará apenas uma vez no programa. Nela, deve conter as informações principais para iniciar o seu programa, que estará contido dentro da função *void loop()*.
> Em *void loop()* deverá conter tudo o que você quiser que o programa faça."

4.2.1 Exemplo de código na linguagem do Arduino

A seguir um exemplo de código desenvolvido na linguagem do Arduino, que tem por função piscar o *LED* de indicação de escrita na placa Arduino.

```
1  /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  This example code is in the public domain.
6  */
7
8  // Pin 13 has an LED connected on most Arduino boards.
9  // give it a name:
10 int led = 13;
11
12 // the setup routine runs once when you press reset:
13 void setup()
14   {
15   // initialize the digital pin as an output.
16   pinMode(led, OUTPUT);
17   }
18
19 // the loop routine runs over and over again forever:
20 void loop()
21   {
22   digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
23   delay(1000);                // wait for a second
24   digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW
25   delay(1000);                // wait for a second
26   }
```

Neste projeto foi utilizado o sensor ultrassônico, a seguir temos um exemplo na linguagem do Arduino de utilização do sensor ultrassônico, o mesmo tem por função informar em centímetros qual a distancia do sensor até um objeto X.

```
1 #include <Ultrasonic.h>
2
```

```
3  Ultrasonic ultrasonic;
4
5  void setup()
6    {
7    //Inicia a porta serial
8    Serial.begin(9600);
9    // (Trig PIN,Echo PIN)
10   //Trig - Entrada do sensor (gatilho)
11   //Echo - Saída do sensor (Eco)
12   ultrasonic.attach(52,53);
13   }
14
15 void loop()
16   {
17   int distancia;
18   // função Ranging, faz a conversão do tempo de resposta do echo em
         centímetros, e armazena na variável distância.
19   distancia = (ultrasonic.Ranging(CM));
20   Serial.print("Distancia em CM: ");
21   Serial.println(distancia);
22   //Espera 1 segundo para fazer a leitura novamente.
23   delay(1000);
24   }
```

Também foi empregado no projeto servo motores, a seguir temos um exemplo de código que tem por função controlar um servo motor.

```
1  /* Sweep
2   by BARRAGAN <http://barraganstudio.com>
3   This example code is in the public domain.
4
5   modified 8 Nov 2013
6   by Scott Fitzgerald
7   http://arduino.cc/en/Tutorial/Sweep
8  */
9
10 #include <Servo.h>
11
12 Servo myservo;  // create servo object to control a servo
13                 // twelve servo objects can be created on most boards
14
15 int pos = 0;    // variable to store the servo position
16
17 void setup()
18 {
19   myservo.attach(9);  // attaches the servo on pin 9 to the servo object
20 }
21
```

```
22 void loop()
23 {
24   for(pos = 0; pos <= 180; pos += 1) // goes from 0 degrees to 180 degrees
25   {                                  // in steps of 1 degree
26     myservo.write(pos);              // tell servo to go to position in
           variable 'pos'
27     delay(15);                       // waits 15ms for the servo to reach the
           position
28   }
29   for(pos = 180; pos>=0; pos-=1)     // goes from 180 degrees to 0 degrees
30   {
31     myservo.write(pos);              // tell servo to go to position in
           variable 'pos'
32     delay(15);                       // waits 15ms for the servo to reach the
           position
33   }
34 }
```

4.3 Desenvolvimento do software de controle

Após definidos alguns recursos básicos para o projeto foi criado o esquema de ligação dos componentes eletrônicos no *software* Fritzing[5], conforme Figuras 3.12 e 3.13 do capítulo 3, criando assim um guia para a correta conexão dos componentes a qualquer momento.

O código fonte desenvolvido para controlar o robô desenvolvido encontra-se como apêndice e no texto, nesta seção será abordada a linguagem do Arduino, suas bibliotecas e funções.

A linguagem do arduino é derivada do C o que permite fazer usos de várias funções já conhecidas por programadores da linguagem C, mas além destas funções, a própria plataforma Arduino traz funções e bibliotecas para controlar componentes da sua arquitetura, serão abordadas as bibliotecas Ultrasonic.h e Servo.h.

Funções da linguagem do Arduino

A função "pinMode();" permite definir um pino como entrada ou saída de dados, por exemplo.

- "pinMode(13, OUTPUT);"

Define o pino 13 como sendo um pino de saída de dados, como por exemplo ligar e desligar um LED, além da opção OUTPUT que define que o pino é para saída, tem-se também a opção INPUT que define o pino como entrada de dados.

A função "delay();" faz com que o programa aguarde um tempo X, o valor de X deve ser informado em milissegundos, por exemplo "X = 1", equivale a um milissegundo, "X = 1000", equivale a um segundo.

- "delay(1000);"

[5] *Software open source*/livre que tem por função o desenvolvimento de desenhos de protótipos de circuitos elétricos.

Define que o programa deve aguardar um segundo, este recurso é muito utilizado por exemplo quando um servo motor executa uma ação que demanda um certo tempo X, desta forma, o servo motor terá tempo para executar sua tarefa e o programa irá aguardar até que o servo motor termine sua ação.

A função "digitalWrite();" permite escrever um valor alto(HIGH) 5V ou baixo(LOW) 0V(terra) no pino definido na função.

- "digitalWrite(9, HIGH);"

  Define que o pino irá fornecer 5V, útil por exemplo para ligar um LED.

Servo.h

A biblioteca Servo.h permite controlar um servo motor, esta biblioteca permite posicionar o servo em vários ângulos normalmente entre 0 e 180 graus.

A função "attach();" tem por função indicar qual o pino irá controlar o servo motor, por exemplo.

- "coxaEsqFrontal.attach(30);"

  Define o controle de um servo motor representado pelo nome "coxaEsqFrontal" no pino 30.

  A função "write();" permite escrever um valor no pino em que a variável está definida, por exemplo.

- "coxaEsqFrontal.write(90);"

  Define que a variável coxaEsqFrontal irá colocar o servo motor em noventa graus.

Ultrasonic.h

A biblioteca Ultrasonic.h não é nativa da IDE do Arduino. Para fazer uso dela é necessário que ela seja adicionada às bibliotecas da IDE Arduino.

Ela foi obtida através do blog RoboticLabVIEW, de autoria de de Lellis Barreto Júnior (2012).

A função "Ranging();" permite fazer a leitura do ambiente a partir de um sonar, seu retorno pode ser em centímetros(CM) ou em polegadas(INC), por exemplo.

- "flag = ultrasonic.Ranging(CM);"

  Adiciona o valor lido em centímetros na variável flag.

Funções desenvolvidas

O *software* desenvolvido é dividido em uma função de movimento andar(); e a função loop(), a função loop() é equivalente a função main() da linguagem C, é nela que é configurada a inicialização do servo motores para antes do movimento.

Função andar é responsável por realizar o movimento das patas que simulam o movimento. Sempre que o sensor ultrassônico retornar um valor de leitura acima de 50CM ela é invocada, e é realizado um novo movimento a partir do estado atual do servo motor.

A partir da leitura do ambiente a função andar() vai ser invocada e sua invocação seguida irá gerar a simulação de movimento.

Capítulo 5

Conclusão

5.1 Resultados

A partir deste projeto pode-se obter um robô hexápode construído a partir de lixo eletrônico e alguns componentes eletrônicos como servo motores e um sensor ultrassônico, conforme as Figuras B.1, B.2, B.3, B.4 e B.5. O mesmo recebeu um *software* desenvolvido na linguagem do Arduino.

O custo no desenvolvimento do projeto foi relativamente baixo quando comparado com projetos similares, os dados apresentados na Tabela 5.1 foram coletados em Novembro de 2015, coletados em sites brasileiros como Mercado Livre e também em sites internacionais como AliExpress ambos são sites de revenda de produtos na *internet*.

A Tabela 5.2 apresenta uma comparação de valores[1] de robôs similares ao robô guardião (os nomes apresentados na Tabela 5.2 foram os mesmos nomes disponibilizados pelo vendedor), a partir da Tabela 5.2 é possível notar uma grande diferença de valores nos projetos, o robô guardião custa 16% do valor médio dos modelos apresentados.

O robô hexápode desenvolvido é um tipo de robô que foi criado já pensando no ambiente ao qual ele pode ser inserido, de modo que ele pode superar ambientes que podem ter pequenos degraus ou desníveis.

Inicialmente foi pensado em se realizar um movimento amplo das patas, mas no decorrer do projeto foi observado que realizar este movimento faria com que o equilíbrio do robô fosse comprometido.

Foi também por esse motivo que o robô adotou a estrutura final atual, visto que foi a que menos interferiu de forma negativa no equilíbrio do conjunto.

Após longa observação também foi possível observar que a melhor posição das patas, coxa e tíbia é aquela em que as patas formam um arco quando observado de frente.

Este arco também pode ser observado na natureza em animais com a mesma estrutura corporal que o projeto. Esta estrutura permite maior estabilidade ao conjunto.

Apesar da preocupação com as pontas das patas, o projeto ainda assim apresenta instabilidade em alguns pisos, apresentando derrapamentos, o que atrapalha sua movimentação.

Também foi percebido que devido à proporção que o projeto adquiriu os servo motores que foram utilizados, não são os mais indicados para o projeto já que eles não tem potencia suficiente para por exemplo levantar o robô nas condições atuais caso ele caia, para resolver este problema faz-se necessário servo motores mais potentes e preferencialmente feito com engrenagens de metal, diferente dos utilizados que tem engrenagens plásticas.

Parte deste problema com os servo motores foi gerado principalmente pela adição do metal à estrutura do projeto, para que permitisse a criação de peças mais resistentes, diferente da ideia original que propunha

[1] Conversão de moedas realizada com base na cotação do dólar em Novembro de 2015.

apenas o uso do PVC.

Como melhoramento desse projeto, pode-se pensar em trabalhos futuros que levem em consideração a otimização dos componentes, principalmente a substituição dos servo motores atuais por servo motores mais potentes deverá resultar em uma estrutura mais forte e que sofrerá menos com a interação com o ambiente.

No mesmo contexto da otimização é pensado na ideia de agregação, para tal tem-se a ideia de adicionar mais articulações nas patas de modo a poder criar movimentos mais suaves e naturais e também uma melhor distribuição de peso entre as patas.

Acerca da interação com o ambiente é interessante a utilização de mais sensores, como o próprio sensor ultrassônico, para que se possa monitorar uma área ainda maior de uma só vez, desta forma também é possível criar novas funções de interação com o ambiente, deste modo novos movimentos para o robô podem ser criados.

Robô Guardião		
Produto	**Unidade em R$**	**Valor Final em R$**
1 Metro de Cano de PVC	R$ 12,00	R$ 12,00
Sensor Ultrassônico	R$ 10,00	R$ 10,00
12 Servo-motores	R$ 13,00	R$ 156,00
Arduino Mega	R$ 70,00	R$ 70,00
	Total:	**R$ 248,00**

Tabela 5.1: Tabela de preços de componentes do Robô Guardião.

Comparação de preços		
Produto	**Real (R$)**	**Dólar (US)**
Robô Guardião	R$ 248,00	US $66,17
6 patas robô, hexápode robô aranha total de 18 graus de liberdade educação / ensino / plataforma experimental	R$ 2.455,92	US $655,00
Conjunto completo Robot Hexapod aranha disco placa de seis 3DOF Kit quadro pernas e 18 Servo	R$ 1617,39	US $431.31
Kit completo 18 DOF 6 pés hexapod robô aranha	R$ 855,10	US $228,00
Arduino alumínio Hexapod aranha 20DOF quadro seis pernas Robot grupo de luxe w / Gripper MG996R Servo	R$ 1612,07	US $430,00
Robo - Soul CR-6 Hexapod Robtics seis pernas Kit Robot 18DOF aranha w / 32CH controlador Digital Servo	R$ 1345,54	US $358,88

Tabela 5.2: Tabela de comparação de preços de robôs hexápodes.

Referências Bibliográficas

arduino.cc (2014). Referência da linguagem. http://playground.arduino.cc/Portugues/Referencia, acesso em Setembro de 2014.

Carrara, V. (2009). Apostila de robótica. http://www.joinville.udesc.br/portal/professores/silas/materiais/Apostila_de_Robotica.pdf, acesso em Junho de 2015.

de Holanda Ferreira, A. B. (2004). Dicionário aurélio, versão 5. http://aureliopositivo.com.br/, acesso em Junho de 2015.

de Lellis Barreto Júnior, C. (2012). Biblioteca sensor ultra-sônico ide 22 e 1.0.1 - library ultrasonic sensor ide 22 e 1.0.1. http://roboticlabview.blogspot.com.br/2012/10/biblioteca-sensor-ultra-sonico-ide-22-e.html, acesso em Abril de 2014.

de Matos, N. M. R. (2012). Análise do funcionamento de um servomotor de corrente alternada com Ímãs permanentes. https://repositorio-aberto.up.pt/bitstream/10216/72684/1/000155343.pdf, acesso em Junho de 2015.

de Sousa, M. F. (2013). Conceitos básicos em monitoramento e avaliação. http://repositorio.enap.gov.br/bitstream/handle/1/992/SOUSA,%20Marconi%20Fernandes%20-%20Conceitos%20B%C3%A1sicos%20de%20Monitoramento%20e%20Avalia%C3%A7%C3%A3o.pdf?sequence=1, acesso em Junho de 2015.

de Souza, J. A. M. F. (2005). Robótica. http://webx.ubi.pt/~felippe/main_pgs/mat_didp.htm, acesso em Outubro de 2014.

Favera, E. C. D. (2008). Lixo eletrônico e a sociedade *. www-usr.inf.ufsm.br/~favera/elc1020/t1/artigo-elc1020.pdf, acesso em Novembro de 2015.

Gaier, M. B. (2011). Aprendendo a programar em arduino. http://pt.slideshare.net/Miojex360/apostila-para-programar-arduino, acesso em Setembro de 2014.

Grupo de Robótica (2012). Introdução ao arduino. http://destacom.ufms.br/mediawiki/images/9/9f/Arduino_Destacom.pdf, acesso em Março de 2014.

McRoberts, M. (2011). *Arduino básico*. São Paulo: Novatec Editora, 1 edition.

Newton C. Braga (2012). Como funcionam os sensores ultrassônicos (art691). http://www.newtoncbraga.com.br/index.php/robotica/3484-mec081, acesso em Outubro de 2014.

Olhar Digital (2010). Arduino: robótica para iniciantes. http://olhardigital.uol.com.br/video/arduino-robotica-para-iniciantes/10981, acesso em Março de 2014.

Santos, D. M. (2006). Projeto de sistemas embarcados: Um estudo de caso baseado em microcontrolador e seguindo aosd. http://www.lisha.ufsc.br/pub/Santos_BSC_2005.pdf, acesso em Setembro de 2015.

Suchier, L. (2012). Hexapod raspberry : Le chassis. http://www.syris.fr/index.php/hexapode/hexapode-raspberry-le-chassis/, acesso em Abril de 2014.

Wendling, M. (2010). Sensores. http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf, acesso em Junho de 2015.

Apêndice A

Código Fonte

Código fonte que controla o robô guardião.

```
1  ////////////////////////////////////////////////////////////
2  //                  Declaração de bibliotecas             //
3  ////////////////////////////////////////////////////////////
4  #include <Servo.h> //Biblioteca de controle dos servo motores
5  #include <Ultrasonic.h> //biblioteca de controle do sensor ultrassônico
6  ////////////////////////////////////////////////////////////
7  // Fim declaração de bilbliotecas                          //
8  ////////////////////////////////////////////////////////////
9
10 /*************************************************************
11 *************************************************************/
12 #define tempo 100000000
13 #define rodou 700
14 ////////////////////////////////////////////////////////////
15 // Declarção de servo motores  e sensor ultrason           //
16 ////////////////////////////////////////////////////////////
17 Servo coxaEsqFrontal;
18 Servo coxaEsqMedia;
19 Servo coxaEsqTrazeira;
20 Servo coxaDirFrontal;
21 Servo coxaDirMedia;
22 Servo coxaDirTrazeira;
23
24 Servo tibiaEsqFrontal;
25 Servo tibiaEsqMedia;
26 Servo tibiaEsqTrazeira;
27 Servo tibiaDirFrontal;
28 Servo tibiaDirMedia;
29 Servo tibiaDirTrazeira;
30
31 Ultrasonic ultrasonic; //Declaração do sensor ultrassônico.
32
33 ////////////////////////////////////////////////////////////
```

```
34 // Fim declaração de servo motores  e sensor ultrason       //
35 ////////////////////////////////////////////////////////////
36
37 /*************************************************************
38 *************************************************************/
39
40 ////////////////////////////////////////////////////////////
41 //            Configuração de componentes e portas         //
42 ////////////////////////////////////////////////////////////
43 void setup() //função de configuração do funcionamento da placa arduino e suas
       portas.
44 {
45   coxaEsqFrontal.attach(30);     //Define pino 30 como pino de controle do
         servo coxaEsqFrontal
46   coxaEsqMedia.attach(31);       //Define pino 31 como pino de controle do
         servo coxaEsqMedia
47   coxaEsqTrazeira.attach(32);    //Define pino 32 como pino de controle do
         servo coxaEsqTrazeira
48
49   coxaDirFrontal.attach(34);     //Define pino 34 como pino de controle do
         servo coxaDirFrontal
50   coxaDirMedia.attach(35);       //Define pino 35 como pino de controle do
         servo coxaDirMedia
51   coxaDirTrazeira.attach(36);    //Define pino 36 como pino de controle do
         servo coxaDirTrazeira
52
53   tibiaEsqFrontal.attach(40);    //Define pino 40 como pino de controle do
         servo tibiaEsqFrontal
54   tibiaEsqMedia.attach(41);      //Define pino 41 como pino de controle do
         servo tibiaEsqMedia
55   tibiaEsqTrazeira.attach(42);   //Define pino 41 como pino de controle do
         servo tibiaEsqTrazeira
56
57   tibiaDirFrontal.attach(44);    //Define pino 44 como pino de controle do
         servo tibiaDirFrontal
58   tibiaDirMedia.attach(45);      //Define pino 45 como pino de controle do
         servo tibiaDirMedia
59   tibiaDirTrazeira.attach(46);   //Define pino 46 como pino de controle do
         servo tibiaDirTrazeira
60
61   pinMode(13, OUTPUT);           //Define pino como saída de dados para um LED
62   pinMode(12, OUTPUT);           //Define pino como saída de dados para um LED
63   pinMode(11, OUTPUT);           //Define pino como saída de dados para um LED
64   pinMode(10, OUTPUT);           //Define pino como saída de dados para um LED
65   pinMode(9, OUTPUT);            //Define pino como saída de dados para um LED
66
67   ultrasonic.attach(52, 53);     // (Trig PIN,Echo PIN)
68
69   Serial.begin(9600);            //Permite a comunicação da placa com o
```

```
      computador para acompanhar a leitura da placa Arduino durante a execução
       do programa.
}
////////////////////////////////////////////////////////////
//          Fim configuração de componentes e placa       //
////////////////////////////////////////////////////////////

/*********************************************************
*********************************************************/

////////////////////////////////////////////////////////////
//                          Função main                   //
////////////////////////////////////////////////////////////
void loop()
{
  int flag = 0;
  // Colocar toda  a estrutura em modo inicial  de 90ž e 100ž.
  coxaEsqFrontal.write(90);
  coxaEsqMedia.write(90);
  coxaEsqTrazeira.write(90);

  coxaDirFrontal.write(90);
  coxaDirMedia.write(90);
  coxaDirTrazeira.write(90);

  tibiaEsqFrontal.write(100);
  tibiaEsqMedia.write(100);
  tibiaEsqTrazeira.write(100);

  tibiaDirFrontal.write(100);
  tibiaDirMedia.write(100);
  tibiaDirTrazeira.write(100);

  digitalWrite(13, LOW); //Desliga o LED na porta 13
  delay(rodou); //Aguarda o tempo definido.
  /*********************************************************
  *********************************************************/
  do //Inicia o laço que roda o sistema que controla o robô.
  {
    flag = ultrasonic.Ranging(CM); //Faz a leitura do ambiente e armazena na
        variável flag a distância em CM do objeto a frente.
    //Serial.print(flag);      //Imprime o valor lido na tela do monitor de
        leitura da IDE Arduino.
    //Serial.println(" CM"); //Imprime  CM na frente do valor lido.
    digitalWrite(9, HIGH); //Liga o LED na porta 9 da placa arduino
    andarFrente();  //Chama a função que realiza o movimento que simula o
        andar.
    digitalWrite(9, LOW); //Desliga o LED na porta 9 da placa arduino
  }while(flag>50); //Executa enquanto o valor lido pelo sensor for maior que
```

```
    50 CM.
}
////////////////////////////////////////////////////////////
//                    Fim função main                     //
////////////////////////////////////////////////////////////

void andarFrente() //Escopo da função que define o andar.
{
  //>>>>>>> start loop <<<<<<<
  digitalWrite(10, HIGH);       // liga o LED na porta 10.
  tibiaEsqFrontal.write(45);    //sobe
  delay(rodou/2);               //Aguarda o tempo definido.
  coxaEsqFrontal.write(70);     //frente
  delay(rodou/2);               //Aguarda o tempo definido.
  tibiaEsqFrontal.write(100);   //desce
  delay(rodou/2);               //Aguarda o tempo definido.

  tibiaDirFrontal.write(45);    //sobe
  delay(rodou/2);               //Aguarda o tempo definido.
  coxaDirFrontal.write(110);    //frente
  delay(rodou/2);               //Aguarda o tempo definido.
  tibiaDirFrontal.write(100);   //desce
  delay(rodou/2);               //Aguarda o tempo definido.
  digitalWrite(10, LOW);        // desliga o LED na porta 10.

  digitalWrite(12, HIGH);       // liga o LED na porta 12.
  tibiaEsqTrazeira.write(45);   //sobe
  delay(rodou/2);               //Aguarda o tempo definido.
  coxaEsqTrazeira.write(70);    // p/ frente
  delay(rodou/2);               //Aguarda o tempo definido.
  tibiaEsqTrazeira.write(100);  //desce
  delay(rodou/2);               //Aguarda o tempo definido.

  tibiaDirMedia.write(45);      //sobe
  delay(rodou/2);               //Aguarda o tempo definido.
  coxaDirMedia.write(110);      // p/ frente
  delay(rodou/2);               //Aguarda o tempo definido.
  tibiaDirMedia.write(100);     //desce
  delay(rodou/2);               //Aguarda o tempo definido.
  digitalWrite(12, LOW);        // desliga o LED na porta 12.

  digitalWrite(11, HIGH);       // liga o LED na porta 11.
  tibiaEsqMedia.write(45);      //sobe
  delay(rodou/2);               //Aguarda o tempo definido.
  coxaEsqMedia.write(70);       // p/ frente
  delay(rodou/2);               //Aguarda o tempo definido.
  tibiaEsqMedia.write(100);     //desce
  delay(rodou/2);               //Aguarda o tempo definido.

```

```
162     tibiaDirTrazeira.write(45);     //sobe
163     delay(rodou/2);                 //Aguarda o tempo definido.
164     coxaDirTrazeira.write(110);     // p/ frente
165     delay(rodou/2);                 //Aguarda o tempo definido.
166     tibiaDirTrazeira.write(100);    //sobe
167     delay(rodou/2);                 //Aguarda o tempo definido.
168     digitalWrite(11, LOW);          // desliga o LED na porta 11.
169
170     digitalWrite(12, HIGH);         // liga o LED na porta 12.
171     coxaEsqFrontal.write(90);       // p/ frente corpo
172     coxaEsqMedia.write(90);         // p/ frente corpo
173     coxaEsqTrazeira.write(90);      // p/ frente corpo
174
175     coxaDirFrontal.write(90);       // p/ frente corpo
176     coxaDirMedia.write(90);         // p/ frente corpo
177     coxaDirTrazeira.write(90);      // p/ frente corpo
178     delay(rodou/2);                 //Aguarda o tempo definido.
179     digitalWrite(12, LOW);          // desliga o LED na porta 12.
180     //>>>>>>>>> fim loop <<<<<<<<<
181 }
```

Apêndice B

Robô Guardião

Robô Guardião.

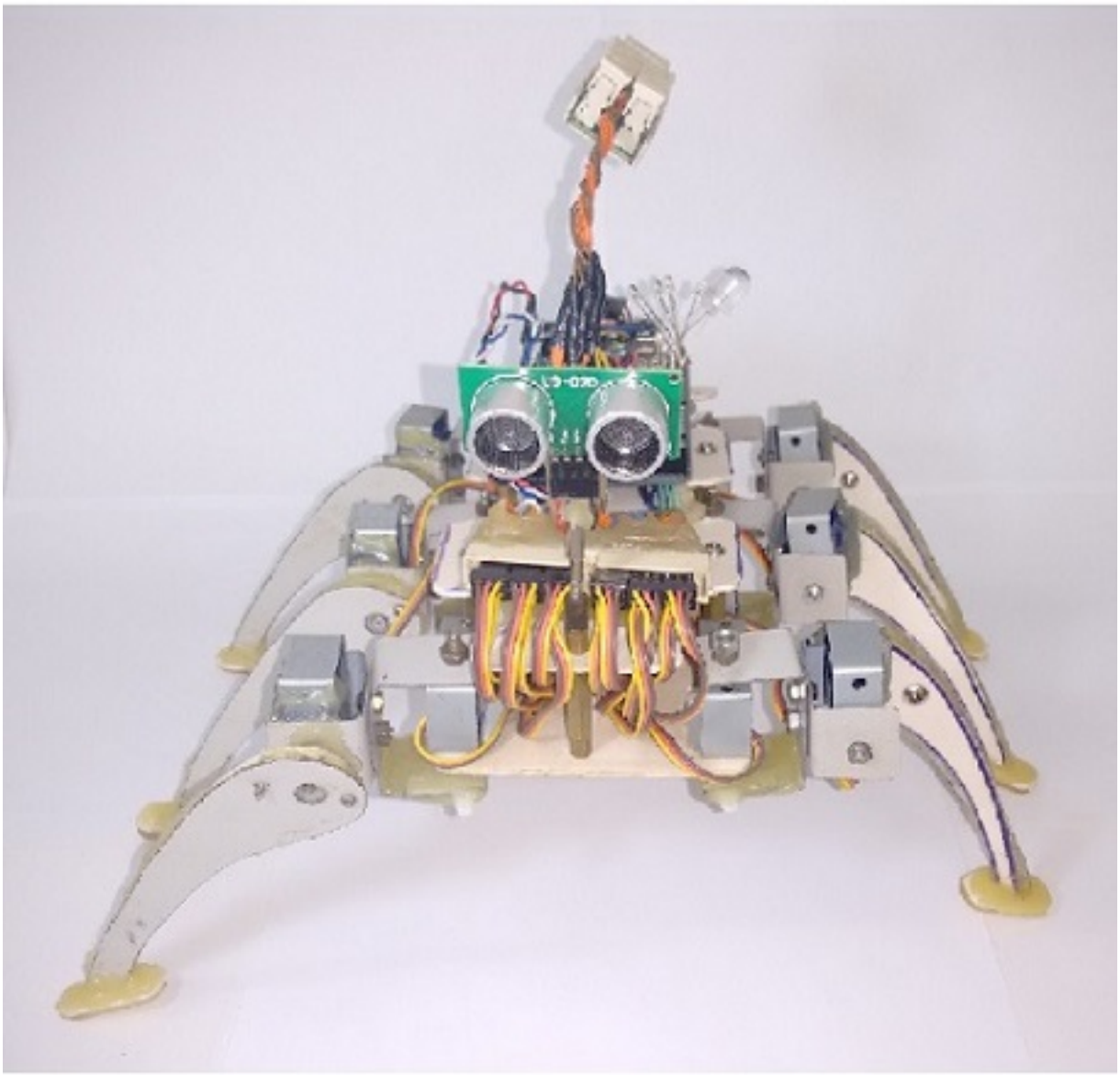

Figura B.1: Projeto Robô Guardião Frente(Imagem do autor).

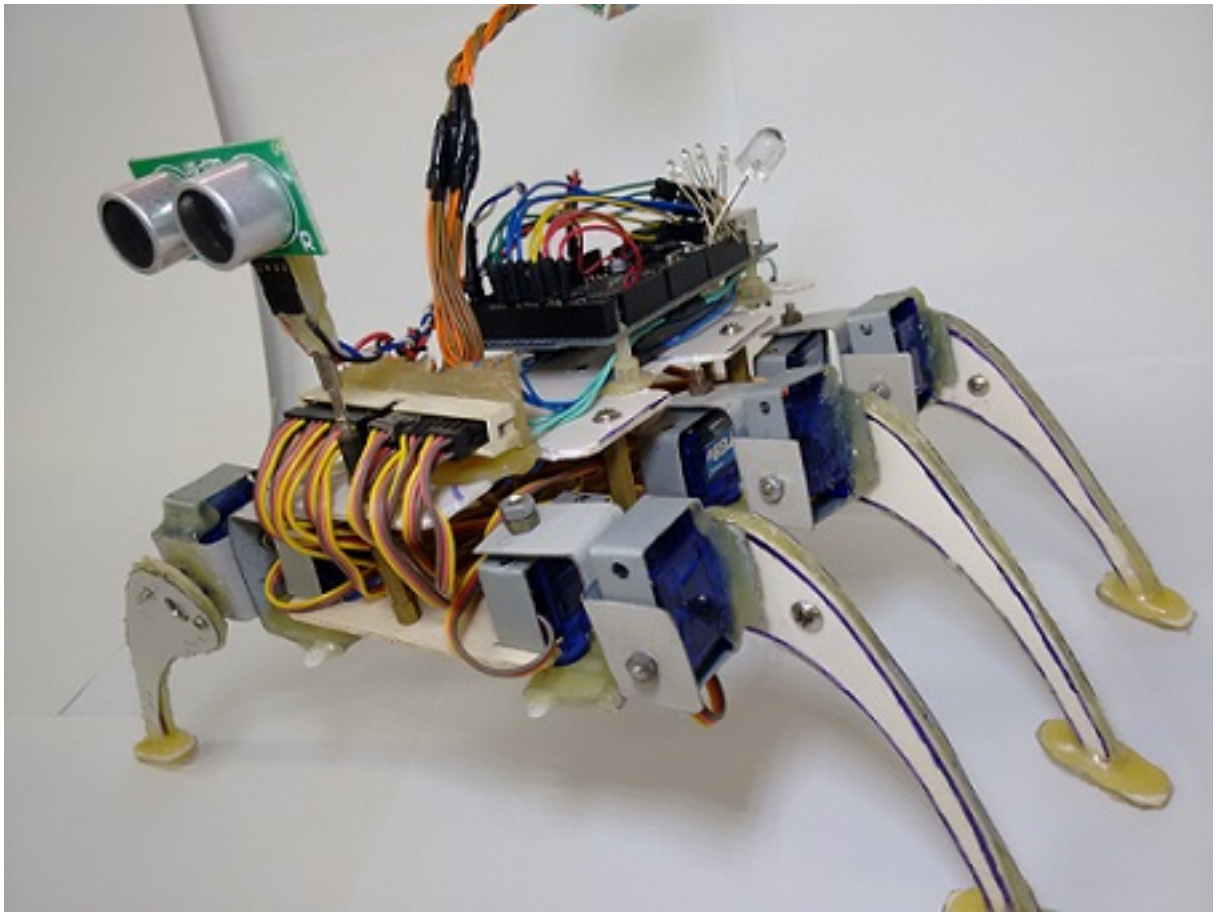

Figura B.2: Projeto Robô Guardião Lateral 1(Imagem do autor).

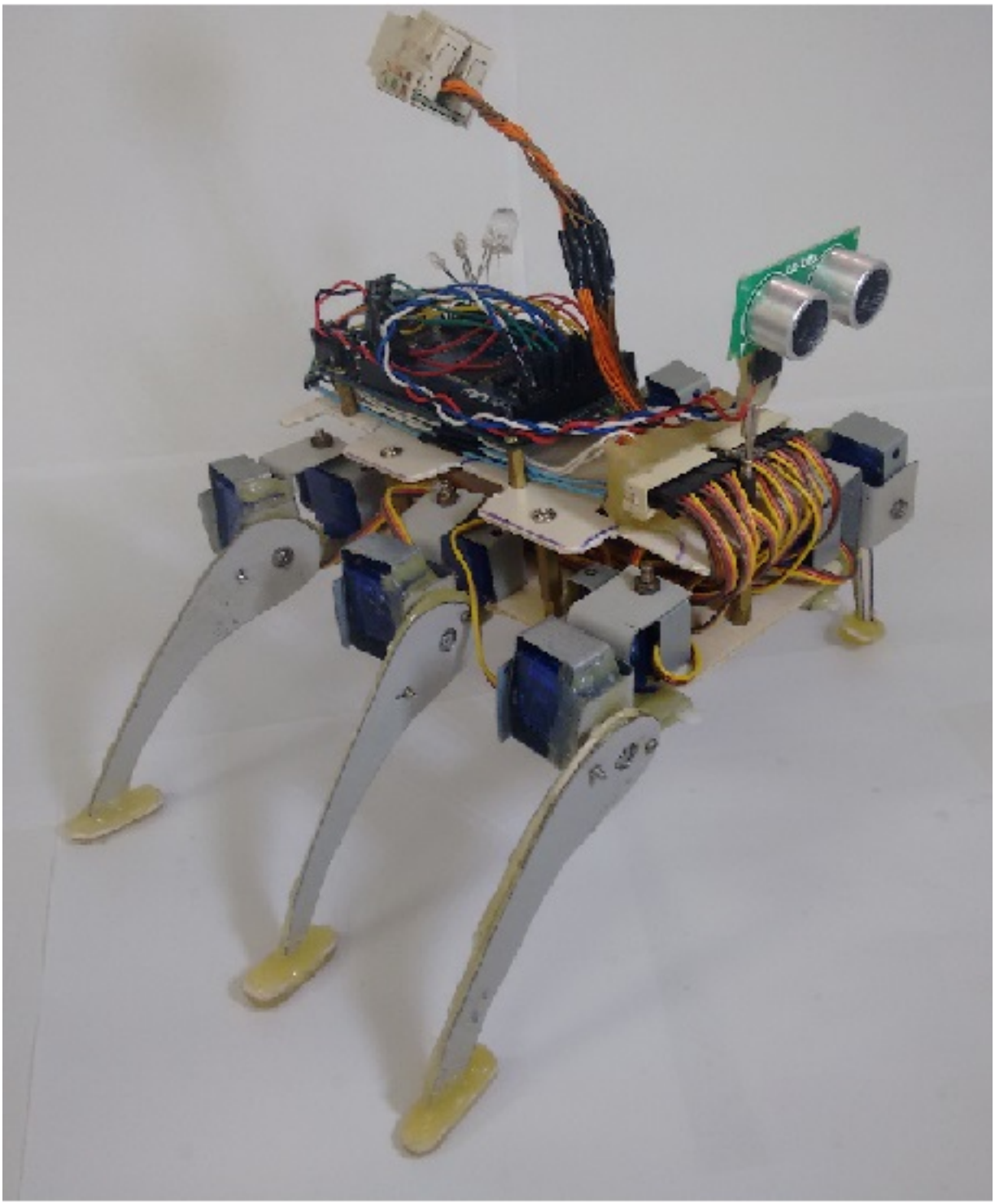

Figura B.3: Projeto Robô Guardião Lateral 2(Imagem do autor).

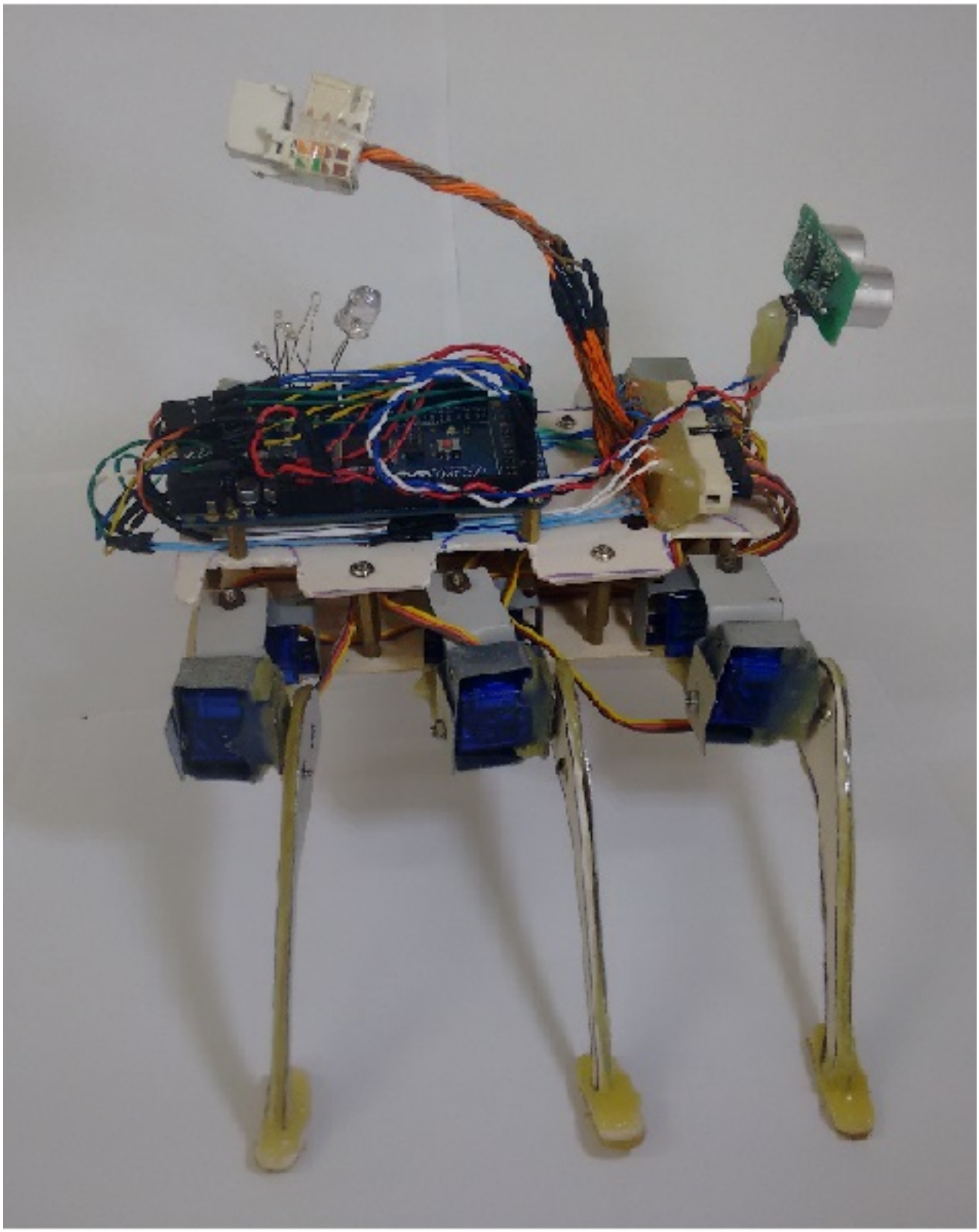

Figura B.4: Projeto Robô Guardião Lateral Traseira(Imagem do autor).

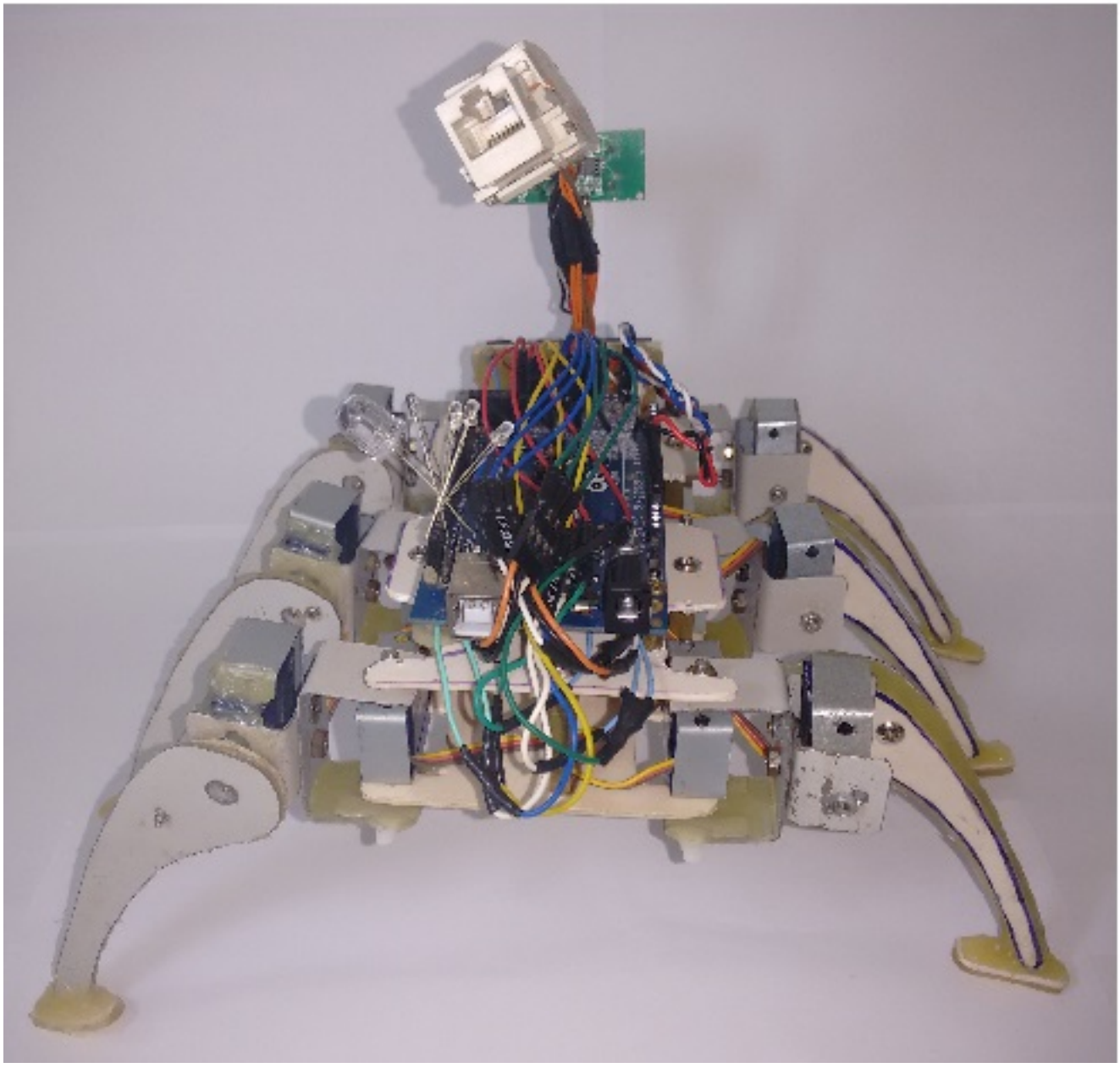

Figura B.5: Projeto Robô Guardião Traseira(Imagem do autor).