

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)
CURSO DE ENGENHARIA DE COMPUTAÇÃO

BAZZO, ÂNGELI
PIO, LUCAS

ROBÔ AUTÔNOMO SEGUIDOR DE PAREDES INTERNAS (RASPI)

OFICINA DE INTEGRAÇÃO 1 – RELATÓRIO FINAL

CURITIBA

2019

BAZZO, ÂNGELI
PIO, LUCAS

ROBÔ AUTÔNOMO SEGUIDOR DE PAREDES INTERNAS (RASPI)

Relatório Final da disciplina Oficina de Integração 1, do curso de Engenharia de Computação, apresentado aos professores que ministram a mesma na Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção da aprovação na disciplina.

Orientador: Prof. Gustavo Benvenutti Borba
Prof. Ronnier Frates Rohrich

CURITIBA

2019

RESUMO

. ROBÔ AUTÔNOMO SEGUIDOR DE PAREDES INTERNAS (RASPI). 20 f. Oficina de Integração 1 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2019.

Este trabalho apresenta a construção de um robô móvel que seja hábil para locomover-se em um ambiente controlado, seguindo paredes internas. Para o projeto de hardware foi utilizado o Chassi Redondo 2WD e ainda, o robô contou com um microcontrolador, sensores e atuadores que foram escolhidos no intuito de se construir um robô móvel autônomo. O protótipo apresentou um bom desempenho uma vez que ele foi capaz de perceber e realizar todas as manobras predefinidas, porém a falta de precisão dos componentes utilizados na construção do carrinho foi um agente limitante na distância total percorrida.

Palavras-chave: Robô móvel autônomo, ambiente controlado, seguindo paredes internas, microcontrolador, sensores, atuadores

ABSTRACT

. TÍTULO EM INGLÊS. 20 f. Oficina de Integração 1 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2019.

This work presents the construction of a mobile robot that can be used to move around in a controlled environment, following the inner walls. For the hardware design it was used the Round Chassis 2WD besides that, the robot had a microcontroller, sensors and actuators that were chosen in order to create an autonomous mobile robot. The prototype performed well as it was able to detect and perform all predefined maneuvers, but the lack of precision of the components used in the construction of the cart was a limiting agent of the total distance traveled.

Keywords: Autonomous mobile robot, controlled environment, following the inner walls, microcontroller, sensors, actuators

LISTA DE FIGURAS

FIGURA 1	– Circuito simplificado de uma Ponte H. Fonte (REIS, 2017).	7
FIGURA 2	– Ilustração do funcionamento de um sensor ultrassônico. Fonte (THOMSEN, 2011).	8
FIGURA 3	– Parede de teste confeccionada.	9
FIGURA 4	– Componentes adquiridos para a execução do projeto. São eles Arduino Uno R3, Driver Motor Ponte H L298ns, Chassis Redondo 2WD, Sensor de Distância Ultrassônico.	10
FIGURA 5	– Esquema de conexão dos sensores, Arduino e bateria.	11
FIGURA 6	– Esquema de conexão dos motores, Driver Ponte H, Arduino e baterias.	11
FIGURA 7	– Diagrama lógico do funcionamento do robô. Adaptado de (MCCOMB, 2013)	14
FIGURA 8	– Carrinho alterado com o uso da roda boba e de pesos para balanceamento.	16
FIGURA 9	– Diagrama de Gantt proposto no pré-projeto para planejamento de desenvolvimento.	17

SUMÁRIO

1	INTRODUÇÃO	6
2	FUNDAMENTAÇÃO TEÓRICA	7
2.1	PONTE H	7
2.2	SENSORES ULTRASSÔNICOS	8
3	METODOLOGIA	9
3.1	DESCRIÇÃO DO ROBÔ E DA PISTA DE TESTES	9
3.2	PROJETO DE HARDWARE	10
3.3	PROJETO DE SOFTWARE	12
4	RESULTADOS	15
4.1	CHASSIS	15
4.2	HARDWARE	15
4.3	SOFTWARE	16
5	CRONOGRAMA E CUSTOS DO PROJETO	17
5.1	CRONOGRAMA	17
5.2	CUSTOS	18
6	CONCLUSÕES	19
	REFERÊNCIAS	20

1 INTRODUÇÃO

Este documento apresenta o desenvolvimento de um protótipo utilizando microcontroladores. Trabalho requerido pela disciplina Oficinas de Integração I do curso de Engenharia de Computação da UTFPR. Deste modo, esse trabalho tem como intuito a introdução do aluno a praticas de documentação e apresentação de um projeto, bem como a organização do mesmo utilizando cronogramas e diagramas para gerenciar o tempo e o desenvolvimento do projeto em si. Além disso, também é possível, ao aluno, adquirir conhecimento acerca dos diferentes materiais aplicados no desenvolvimento do projeto e dos conceitos de outras disciplinas como física e matemática que podem vir a ser necessários para tal.

O objeto de estudo, portanto, é um robô autônomo que, a partir da utilização de sensores ultrassônicos mais o processamento dos dados e controle realizados pelo microcontrolador Arduíno Uno R3, é capaz de seguir paredes e assim realizar percursos em um ambiente controlado.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 PONTE H

Ponte H é um circuito que tem como objetivo mais comum o controle da direção de motores. Composto basicamente por quatro chaves, ela é capaz de controlar o sentido de rotação de um motor por meio de combinações das mesmas. O circuito básico de uma ponte H está representado na Figura 1. Quando as chaves S1 e S4 estiverem abertas e as chaves S3 e S2 estiverem fechadas a corrente no motor fluirá da direita para esquerda. No entanto, no caso de as chaves S1 e S4 sejam as que estão fechadas e das outras duas serem as que estão abertas a corrente fluirá da esquerda para a direita (SIDNEY, 2015).

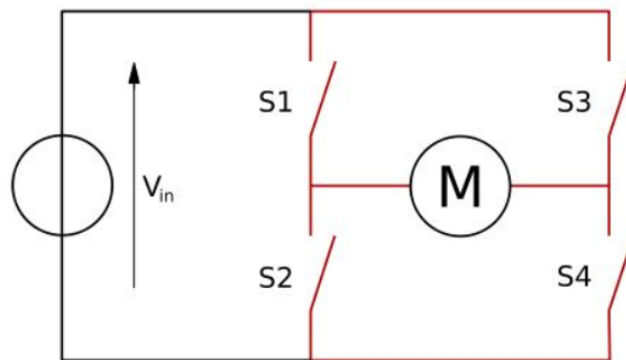


Figura 1: Circuito simplificado de uma Ponte H. Fonte (REIS, 2017).

A principal função da Ponte H, portanto, é conectar o microcontrolador a outros dispositivos elétricos, uma vez que este não é capaz de suportar a corrente necessária para o acionamento destes aparelhos. Neste caso a Ponte H funciona amplificando o sinal do microcontrolador, controlando assim qualquer que seja o dispositivo necessário.

2.2 SENSORES ULTRASSÔNICOS

Os sensores são os responsáveis pela percepção do robô acerca do ambiente. Por meio destes é possível que um sistema adquira informações necessárias para o seu processo de tomada de decisões. No caso do sensor ultrassônico a informação concebida é a distância. Como um SONAR (Sound Navigation and Ranging, ou navegação e determinação de distância pelo som), o sensor ultrassônico utiliza do ultra-som para perceber objetos a determinadas distancias como na Figura 2. O ultra-som emitido pelo sensor se move ate atingir um objeto sendo refletido. Deste modo, calculo da distância se da a partir do eco produzido pelo som propagado ate o obstaculo.

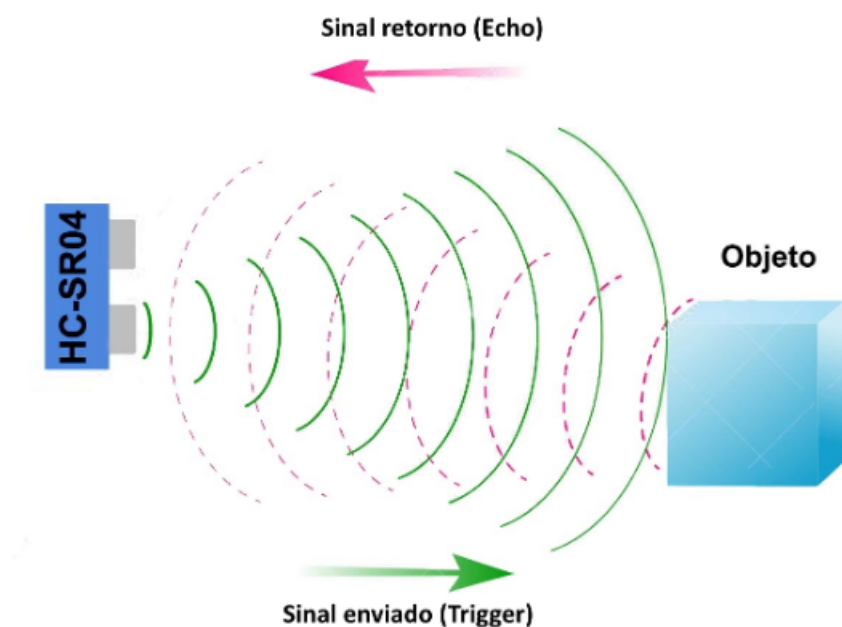


Figura 2: Ilustração do funcionamento de um sensor ultrassônico. Fonte (THOMSEN, 2011).

3 METODOLOGIA

3.1 DESCRIÇÃO DO ROBÔ E DA PISTA DE TESTES

O projeto físico do robô iniciou-se a partir da estrutura mecânica. O emprego do Chassis Redondo 2WD se deve a facilidade que sua simetria redonda proporcionaria as demais etapas do projeto. Uma vez redondo o robô poderia girar em torno de si mesmo sem se preocupar com possíveis obstáculos. O robô desenvolvido no trabalho contou com apenas um microcontrolador, dois sensores de distancia, um driver de potência, dois motores DC e duas fontes de energia. A estrutura física do robô é composta de duas placas de acrílico onde os componentes foram fixados com o auxílio de fitas e abraçadeiras plásticas.

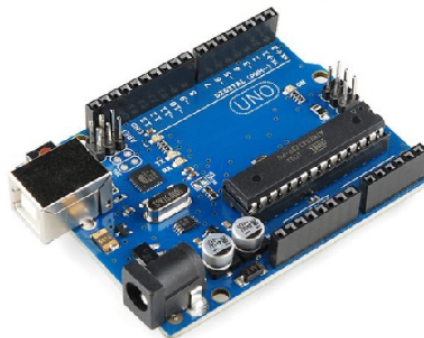
Em paralelo a construção do robô em si foram construídas umas paredes utilizando papelão, palitos de artesanato e cola quente como exemplificado na Figura 3. Deste modo, os testes poderiam ser mais dinâmicos, já que as paredes poderiam ser reorganizadas da forma que fosse necessário, e controlados, uma vez que os testes seriam mais consistentes, como a descrição do projeto sugere.



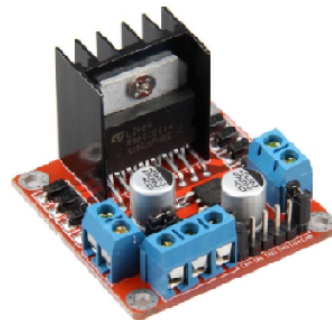
Figura 3: Parede de teste confeccionada.

3.2 PROJETO DE HARDWARE

O hardware foi projetado a partir das necessidades básicas que o protótipo necessitaria, seguir as paredes internas. A Figura 4 ilustra alguns componentes que se mostraram necessários para a execução dessas atividades e portanto foram adquiridos, são eles: Arduino Uno R3, Driver Motor Ponte H L298ns, Chassis Redondo 2WD, Sensor de Distância Ultrassônico. A partir disso o esforço foi direcionado à integração desses componentes.



Arduino Uno R3



Driver Motor Ponte H L298ns



Chassis Redondo 2WD



Sensor de Distância Ultrassônico

Figura 4: Componentes adquiridos para a execução do projeto. São eles Arduino Uno R3, Driver Motor Ponte H L298ns, Chassis Redondo 2WD, Sensor de Distância Ultrassônico.

A conexão feitas com os sensores foram feitas como demonstrado na Figura 5. A alimentação dos sensores é feita através da porta de 5V do Arduino Uno R3 e portanto o Gnd dos sensores também deve estar conectado ao Gnd do microcontrolador que por sua vez é alimentado por uma bateria de 9V. Já as portas lógicas, por onde os sensores mandam as informações necessárias para o calculo da distancia são conectados da seguinte forma: porta digital 8 e 9 conectadas respectivamente ao trig e ao echo do sensor lateral enquanto as portas 10 e 11 são conectadas ao trig e ao echo do sensor frontal.

Para o controle dos motores foram dedicadas as portas digitais de números 4, 5, 6

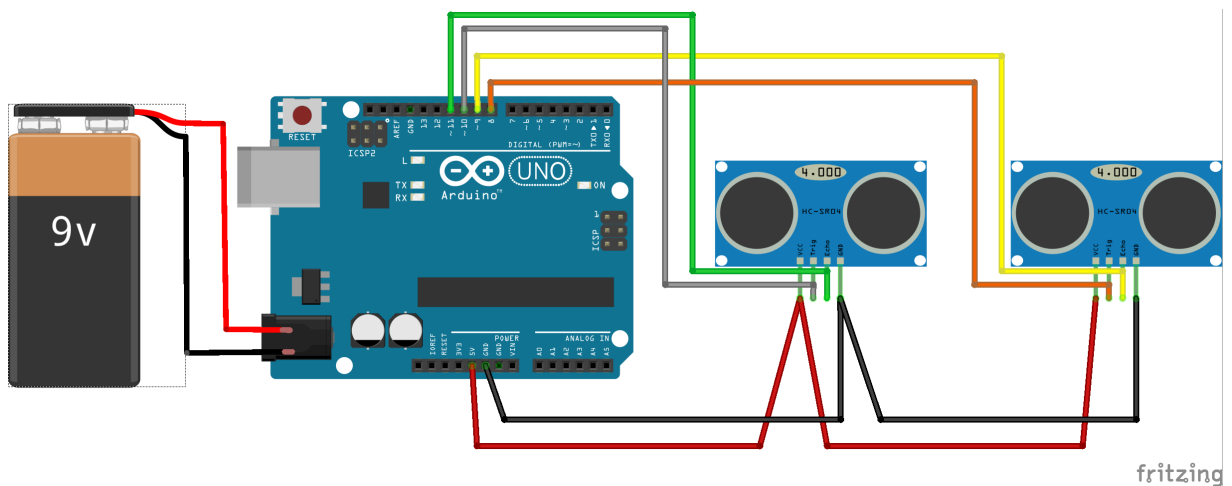


Figura 5: Esquema de conexão dos sensores, Arduino e bateria.

e 7. São as correntes que saem delas que vão ser responsáveis por controlar o Driver Motor Ponte H L298ns ao qual o Arduino está conectado, como mostra a Figura 6, realizando o processo de inversão de corrente e então possibilitando o controle da direção do giro dos motores. A alimentação do Driver de Motor por sua vez acabou por exigir uma alimentação externa ao Arduino uma vez que a corrente fornecida por este não seria suficiente para alimentar tudo. Neste caso foi utilizado uma caixa de quatro pilhas AA. Os motores foram conectados diretamente ao Driver de Motor. Também vale ressaltar que a o Gnd do Arduino e da Ponte H precisam estar conectados para que a passagem das correntes resultantes das portas digitais não sejam interrompidas.

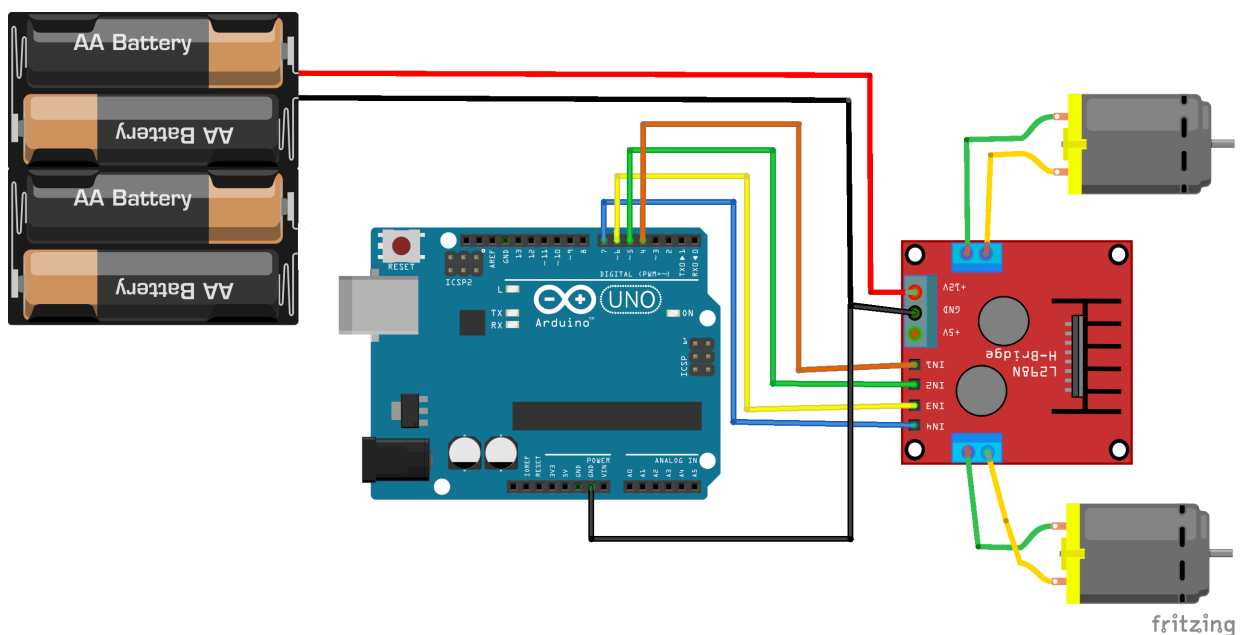


Figura 6: Esquema de conexão dos motores, Driver Ponte H, Arduino e baterias.

3.3 PROJETO DE SOFTWARE

A elaboração do software do projeto começou em um alto nível através de um diagrama (Figura 7) que foi pensado com o intuito de definir possíveis respostas do robô a determinadas situações. Em seguida os movimentos que seriam necessários para solucionar esses problemas foram implementados e testados como na Listagem 1 para que a execução fosse a mais precisa possível. As funções "arrumaDireita" e "arrumaEsquerda" só foram consideradas após os testes de desempenho mostrarem uma frequente desnivelção com relação a parede devido a falta de precisão do movimento dos motores.

Listagem 1: Pseudocódigo das movimentações pré definidas do robô

```

1 //Movimentos pre definidos
void para(){
3     para o movimento das rodas;
}
5 void vaiParaFrente(){
    gira as rodas para frente por milissegundos;
7     para();
}
9 void giraEsquerda(){
    enquanto a roda esquerda gira para trás a da direita gira para frente
        por tempo suficiente para o robô girar 90 graus;
11    para();
}
13 void viraDireita(){
    vai para frente por um tempo;
15    vira em arco para direita (roda da direita parada e roda da esquerda
        gira para frente);
    vai para frente por um tempo;
17    para();
}
19 void arrumaEsquerda(){
    gira a roda da direita para frente enquanto a roda da esquerda esta
        parada por milissegundos;
21    vai para frente por um tempo;
    para();
23 }
void arrumaDireita(){
25    gira a roda da esquerda para frente enquanto a roda da direita esta
        parada por milissegundos;

```

```

    vai para frente por um tempo;
27   para () ;
}

```

:

Em seguida foi necessário pegar a informação fornecida pelo sensor e converter em algo que pudesse ser utilizado como parâmetro logico para o resto do programa. Para isso foi preciso incluir a biblioteca "Ultrasonic.h". Com o uso de algumas funções dessa biblioteca a informação recebida estava pronta para ser interpretada logicamente como mostra a Listagem 2. Assim que o microcontrolador passa a ter os dados então começa a análise que vai resultar na tomada de decisões.

Listagem 2: Código do loop principal

```

// Carrega a biblioteca do sensor ultrassonico
2   #include <Ultrasonic.h>
// Define os pinos para o trigger e echo
4   #define pino_trigger_lateral 8
    #define pino_echo_lateral 9
6   #define pino_trigger_frente 10
    #define pino_echo_frente 11
8
// Inicializa o sensor nos pinos definidos acima
10   Ultrasonic ultrasonic1(pino_trigger_lateral , pino_echo_lateral);
    Ultrasonic ultrasonic2(pino_trigger_frente , pino_echo_frente);
12
//... aqui estao as funcoes de movimentacao predefinidas ...//
14 void loop() {
    //Le as informacoes do sensor em cm
16   float cmMsec_lateral , cmMsec_frente;
    long microsec = ultrasonic1.timing();
18   cmMsec_lateral = ultrasonic1.convert(microsec , Ultrasonic::CM);
    microsec = ultrasonic2.timing();
20   cmMsec_frente = ultrasonic2.convert(microsec , Ultrasonic::CM);

22   if(cmMsec_lateral <= 4 || cmMsec_lateral >=400){
        arrumaEsquerda();
24   }
    if (cmMsec_frente <10){
26   giraEsquerda();

```

```

}
28 else if (4<=cmMsec_lateral && cmMsec_lateral <=12){
    vaiParaFrente();
30 }
    else if ( cmMsec_lateral >=25 ){
32     viraDireita();
    }
34 else if (cmMsec_lateral >=12 && cmMsec_lateral <=25 ){
    arrumaDireita();
36 }
    para();
38 }

```

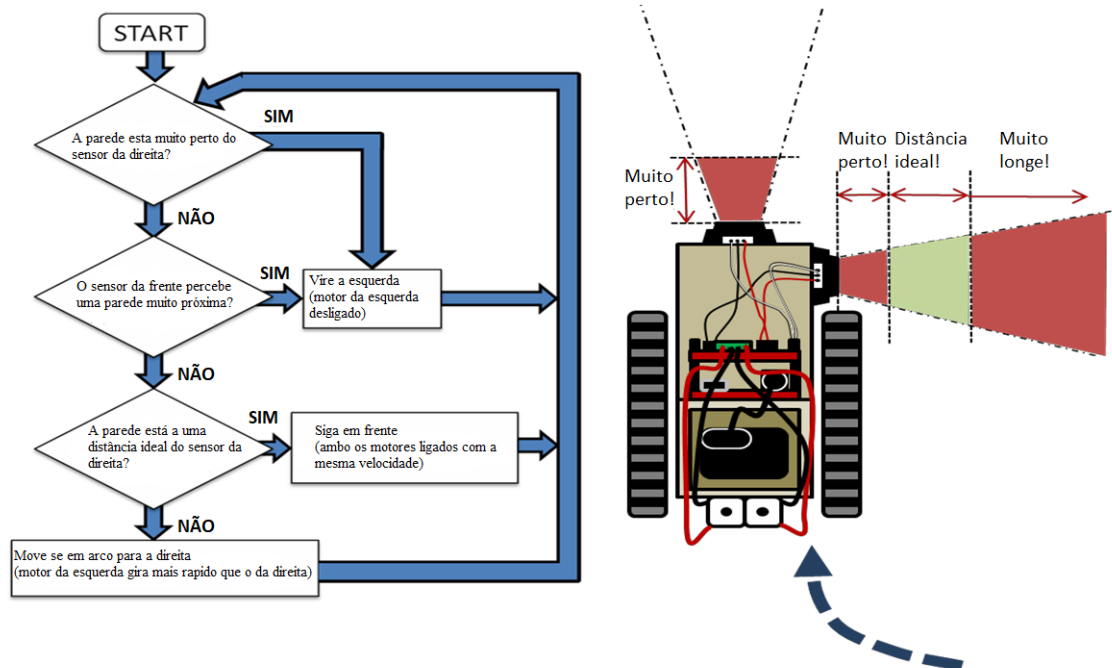


Figura 7: Diagrama lógico do funcionamento do robô. Adaptado de (MCCOMB, 2013)

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos a partir do processo de construção do robô apresentado no Capítulo 3. Os resultados se referem a montagem e configuração do hardware, a implementação do software e aos procedimentos realizados para corrigir os problemas encontrados durante a execução do método citado. Um vídeo de exposição acerca dos materiais utilizados e do resultado final foi produzido e esta disponível para acesso no link: <https://www.youtube.com/watch?v=ATVdEzgGpKE> .

4.1 CHASSIS

Durante o desenvolvimento do projeto o chasis que inicialmente facilitaria todo o desenvolvimento do protótipo acabou por se tornar um problema. Como sua estrutura possuía quatro rodas onde apenas duas eram traçadas o desbalanceamento consequente da integração de novos componentes ao projeto passou a fazer o carrinho patinar. Isso fez com que os movimentos se tornassem imprecisos.

Com o intuito de corrigir esse problema as rodinhas não traçadas foram retiradas e uma roda boba foi colocada no lugar. Com o carrinho agora instável tendo em vista a perda de uma das rodas que servia de suporte, alguns pesos foram fixados ao carrinho fazendo o equilíbrio do mesmo permanecer majoritariamente pendendo para o lado que ele possuía a roda de apoio. A versão do chasis alterada é mostrada na Figura 8.

4.2 HARDWARE

O desenvolvimento da parte do hardware referente a movimentação e da parte referente aos sensores ocorreu bem enquanto que individualmente. Nessas etapas, em ambos os casos, o Arduíno era quem alimentava todo o sistema sendo que ele era alimentado por um carregador portátil para celular. No momento da integração dessas partes percebeu-se que a corrente que o Arduíno poderia fornecer não era suficiente para todo o sistema uma vez que o recomendado

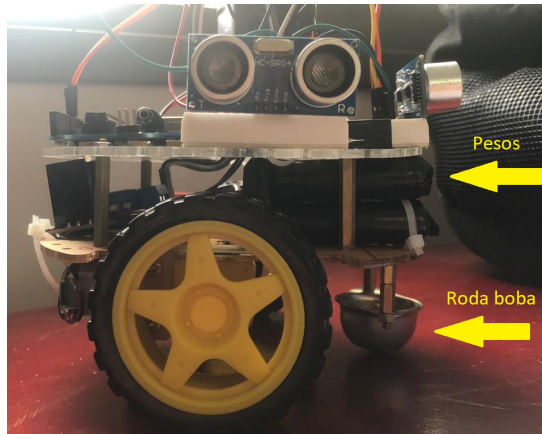


Figura 8: Carrinho alterado com o uso da roda boba e de pesos para balanceamento.

era 500mA e o necessário para o funcionamento dos anexos já era de 500mA desconsiderando o tanto que o peso do carrinho poderia influenciar.

Para tal problema foi proposto como solução alterar a alimentação do Driver de Motor usando, então, uma fonte externa. Esta fonte seria uma caixa de quatro pilhas AA. Os sensores continuaram utilizando o Arduíno como fonte de energia. Para evitar a poluição do espaço com o excesso de elementos grandes o carregador portátil que alimentava o Arduíno também foi substituído, dessa vez por uma bateria de 9v.

4.3 SOFTWARE

O software foi a peça mais sensível a mudanças do projeto. Os resultados de movimentação obtidos no início, para calibrar as ações do robô, tiveram que ser mudados ao longo do projeto conforme mais peso ou novas configurações de chassis fossem anexadas ao protótipo. Além disso, quando houve a integração total do hardware onde o robô sincronizava sua movimentação com os dados dos sensores a falta de precisão que a estrutura mecânica e as reações dos componentes resultavam se tornou evidente. O alinhamento do robô com a parede e uma característica extremamente necessária e para mantê-la pelo maior tempo possível as funções "arrumaDireita" e "arrumaEsquerda" foram implementadas como mostrado na Listagem 1.

5 CRONOGRAMA E CUSTOS DO PROJETO

5.1 CRONOGRAMA

Para uma melhor organização e consequentemente um melhor resultado do protótipo, durante a prestação do pre projeto, foi proposto um cronograma ilustrado na Figura 9. Cronograma este que foi seguido veementemente com exceção da data de entrega das paredes de teste que estava agendada para ser realizada antes do segundo marco mas só foi possível uma semana após o mesmo.

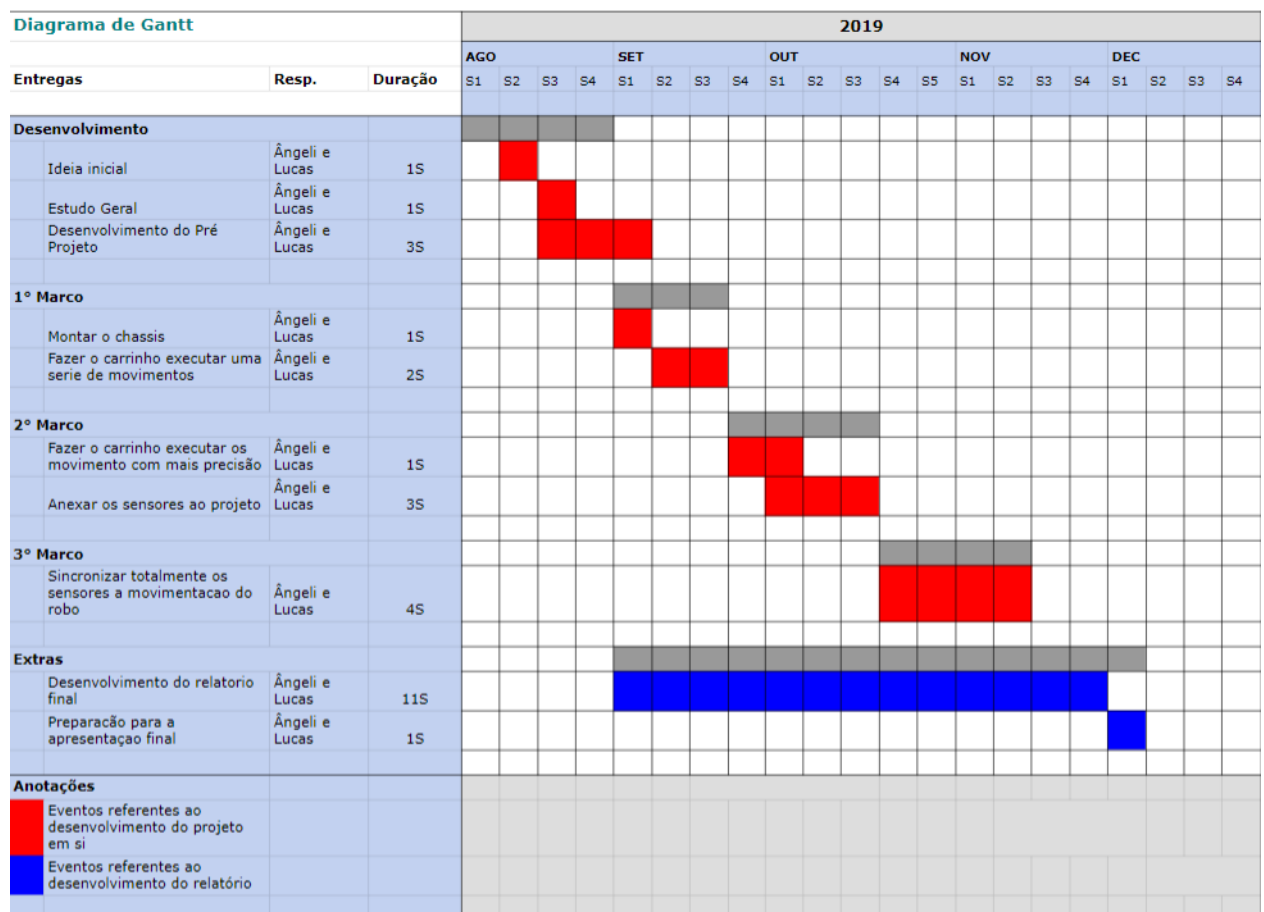


Figura 9: Diagrama de Gantt proposto no pré-projeto para planejamento de desenvolvimento.

5.2 CUSTOS

A aquisição de componentes que estavam fora do que foi proposto durante o pré-projeto não foi muito recorrente. Com exceção das baterias e da Roda boba, ambos componentes que foram necessários para a solução de alguma intempere durante o desenvolvimento do projeto. Outros componentes e ferramentas foram utilizados para a confecção do robô como jumpers ou papelão, mas estes não foram citados por possuírem preço insignificante.

Tabela 1: Tabela final de componentes adquiridos e preços

Quantidade	Nome	Preço(R\$)
1	Arduíno Uno R3	54,90
1	Chassis Redondo 2WD + 2 Motor DC 3-6V	90,90
1	Driver Motor Ponte H L298ns	19,90
2	Sensor de Distância Ultrassônico HC-SR04	21,80
	Baterias e Pilhas	41,00
1	Frete	18,90
1	Roda boba	21,00
Total		268,40

6 CONCLUSÕES

Neste projeto o objetivo inicial foi desenvolver um robô autônomo, o qual teria como função realizar um percurso seguindo paredes internas. O robô foi desenvolvido a partir da famosa estratégia "dividir para conquistar". Uma das partes era a responsável pela movimentação e reação do robô. Como os testes aconteciam em um ambiente controlado foi possível perceber quais movimentos seriam necessários e então implementá-los. A outra parte faz referência aos sensores e do software. Eles juntos deveriam tornar possível ao robô reconhecer sua situação atual em ordem de agir corretamente. Quando integradas, estas partes apresentaram problemas referentes apenas ao calibramento das ações e a detalhes de alimentação.

Portanto, conclui-se que a construção de um robô móvel autônomo envolve diversos problemas que requerem estudo aprofundado de tópicos relacionados a sensores. E quando estes são limitados por qualquer razão desde quantidade a acurácia a precisão com que a parte atuadora do protótipo precisa atuar é bem mais significativa. Deste modo a qualidade da percepção se torna um fator crucial na construção de sistemas de controle que implementam tarefas de navegação mais elaboradas.

REFERÊNCIAS

MCCOMB, M. **Octoblu IoT with chipKit Uno32 + Motor Shield**. 2013. Disponível em: <<https://chipkit.net/tag/robot/>>. Acesso em: 12/09/2019.

REIS, F. **Como funciona uma Ponte H – Controle direcional de motores DC**. 2017. Disponível em: <<http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/como-funciona-uma-ponte-h-controle-direcional-de-motores-dc/>>. Acesso em: 01/12/2019.

SIDNEY, G. Protótipo de um robô móvel autônomo seguidor de paredes internas. http://www.xbot.com.br/wp-content/uploads/2012/10/PB_OENC2014202.pdf, 2015. Acesso em: 1 dez. 2019.

THOMSEN, A. **Filipeflop**. 2011. Disponível em: <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>>. Acesso em: 01/12/2019.