UNIVERSIDADE DE BRASÍLIA – UNB
FACULDADE GAMA

ELETRÔNICA EMBARCADA 120871- TURMA A 2018.2

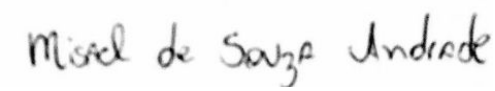
PONTO DE CONTROLE II

CAPINATOR - CORTADOR DE GRAMA MICROCONTROLADO

Jhonathan Nicolas M. Silva,
Matrícula: 16/0031621

Programa de Engenharia Eletrônica
Faculdade Gama - Universidade de Brasília
Área Especial de Indústria Projeção A Brasília,
CEP: 72.444-240
email: jnicolas@aluno.unb.br

Misael de Souza Andrade
Matrícula: 16/0015669



Programa de Engenharia Eletrônica
Faculdade Gama - Universidade de Brasília
Área Especial de Indústria Projeção A Brasília,
CEP: 72.444-240
email: misas.andrade@aluno.unb.br

RESUMO

Este relatório tem como objetivo apresentar a proposta do nosso projeto da disciplina de Eletrônica Embarcada, um Cortador de Grama Autônomo. Para tanto usaremos como elemento principal do produto o **MSP430** da Texas Instruments, um microcontrolador RISC de 16 bits voltados para aplicações de baixo consumo de energia. O projeto visa melhorar a qualidade de vida do usuário, promovendo-o comodidade e segurança nos cuidados de jardins, quintais, campos e etc.

1. JUSTIFICATIVA

Partindo da ideia principal do produto, que é garantir comodidade e consequentemente segurança para o consumidor, é importante apresentar alguns pontos presentes nos produtos similares atuais controlados por um usuário, abordando fatores que inspiram o projeto.

Um dos fatores mais importantes que justificam a escolha do projeto é a comodidade que o produto traz ao consumidor, o cortador de grama autônomo diminui quase por completo o tempo gasto pelo usuário em seu manuseio, onde sem esforço nenhum basta programá-lo que o mesmo faz todo o serviço sem a presença do usuário.

Ao comparar com os modelos não-autônomos presentes no mercado, temos que através da autonomia do cortador o usuário poupará esforço físico e tempo, sendo este relevante quanto a questões pessoais e de exposição do corpo ao sol, e aquele relevante levando em consideração o trabalho de erguer ou empurrar o peso do cortador, onde o peso médio dos produtos presentes no mercado são de 2,6 kg para aparadores[3], 8,7 kg para roçadeiras[4] e 25 kg para cortadores[5].

Outro fator importante que destaca a necessidade de automação do cortador de grama é a prevenção de acidentes. Atualmente, a necessidade de interação e manuseio do equipamento vem trazendo muitos riscos para o usuário, onde o número de acidentes com estes produtos é alto e preocupante, onde por exemplo nos Estados Unidos, as máquinas de cortar grama levam a cerca de 55 mil lesões por ano em que cerca de 75 destas pessoas morrem por decorrência dessas lesões[7]. Com o cortador de grama autônomo, o número de acidentes idealmente seria reduzido a zero, uma vez que o usuário não precisaria manter contato ou manter-se próximo do cortador durante sua operação.

Sendo assim o nosso projeto teria total capacidade de inserção neste mercado, tendo como vantagem principal a sua autonomia em operar.

2. OBJETIVOS

Projetar um cortador de gramas que seja capaz de:

- dado uma área limitada fechada, desviar de obstáculos cortando a grama do local;
- cortar gramas de áreas planas e levemente inclinado;
- desviar dos obstáculos de maneira autônoma;
- assegurar a saúde do operador ao manuseá-lo;
- ser energeticamente viável;
- não evadir do local limitado.

3. REQUISITOS DO PROJETO

O projeto proposto neste documento será descrito neste tópico, onde serão apresentados os requisitos que ele deverá atender. Se tratando de um cortador de grama é necessário estabelecer uma altura mínima que ele deverá atingir no corte, a altura da grama a ser cortada depende muito da sua variedade[1] sendo que a menor delas é de 3cm esse referencial será um dos nossos requisitos onde o equipamento deverá possibilitar a grama a ter no máximo essa altura, caso contrário ele não cumprirá esse requisito. Outro fator importante é a autonomia do equipamento o cortador deverá ser capaz de desviar de obstáculos para que possa ser capaz de cortar a grama com a mínima interferência humana possível, alguns exemplos de obstáculos que o equipamento deverá desviar são: Árvores, pedras que impeçam a navegação, paredes, entre outros. Outro requisito importante do projeto é a eficiência nesse caso existe tanto o corte da grama que deverá ser preciso quanto a velocidade que o cortador vai trabalhar, nesse caso ele deverá estar apto a percorrer no mínimo 0,35 km/h, analisando algumas propostas semelhantes de projeto[2] concluímos que essa velocidade é aceitável.

Analisando que um dos nossos objetivos é aumentar de forma eficiente a segurança na parte da jardinagem quando diz respeito a operar cortadores de grama o projeto deverá ter alguns requisitos de segurança. Um destes muito importante é o desligamento automático do motor responsável pelo giro das lâminas quando o equipamento for levantado ou eventualmente sofrer algum tipo de capotamento, também deverá ser levado em consideração a situação quando uma pessoa tenta levantar o equipamento em funcionamento, ambas situações tem como objetivo evitar que o operador sofra algum tipo de corte devido a exposição da lâmina. Um outro caso importante é quando o cortador estiver tentando desviar de um obstáculo ele deve reduzir a velocidade do giro das lâminas isso por duas razões principais, uma delas é que uma eventual perda de precisão poderia acertar esse objeto com a lâmina quando o equipamento estiver tentando contorná-lo o que ocasionaria prejuízo tanto para o equipamento quanto para o objeto em questão com uma redução da velocidade, caso houvesse uma colisão a chance de prejuízos muito grandes seriam reduzidas, a outra razão é pelo fato de que caso o objeto se trate de um animal ou uma pessoas com uma velocidade muito alta poderia haver um estilhaço de gramas nessa direção.

Figura 1. Situações onde a lâmina do cortador deve ser desligada.

Outra medida de segurança que deverá ser feita como um dos requisitos do projeto é o acionamento de um aviso sonoro por meio de, por exemplo, um buzzer apitando em uma determinada frequência para indicar às pessoas próximas que o

equipamento está em funcionamento, isso pode ser essencial visto que possivelmente os motores que serão utilizados são bem mais silenciosos do que os que costumam ser utilizados nesse tipo de equipamento.

Em relação à eficiência energética o cortador deverá conseguir operar por 20 minutos sem carregar, o tempo de carregamento não é um requisito visto que depende da bateria a ser utilizada, mas não poderá ser maior que 60 minutos, existem muitos modelos de cortadores de grama manuais no mercado se assemelham com essas especificações em relação ao tempo de funcionamento e de carregamento[3], portanto, entendemos que não haverá grandes prejuízos em relação a isso se esses dois requisitos forem cumpridos. A tabela 1 resume os requisitos que o projeto deverá atender em seu pleno funcionamento.

Tópico	Requisito
Segurança	Desligamento do motor em caso de eventual capotamento ou tentativa de expor a lâmina.
	Redução da velocidade da lâmina ao contornar obstáculos.
	Aviso sonoro durante o funcionamento.
Eficiência energética	Autonomia de no mínimo 20 minutos.
	Tempo de carregamento de no máximo 60 minutos.
Eficiência operacional	Percorrer 0,35 km/h.
	Desviar de obstáculo sem interverência humana.
	Não permitir que a grama fique maior que 3cm.

Tabela 1. Requisitos do projeto.

4. BENEFÍCIOS

O principal benefício que este equipamento trará é o aumento da segurança para essa atividade. Aproximadamente 75 pessoas morrem por ano em decorrência destas lesões nos EUA. Especialmente trágicos são os casos de espectadores inocentes,frequentemente crianças, lesados ou mortos por alguém "dirigindo" uma máquina de cortar grama[7]. Um cortador de grama autônomo traria um nível de segurança muito maior que os manuais, pois não dependeria do estado psicológico ou físico do operador além de não representar riscos ergonômicos devido ao longo período de o indivíduo fica operando essas máquinas. Além desses fatores existe o risco de estilhaços de grama, madeira e entre outros objetos dessa natureza que pode trazer riscos para as pessoas que estão operando essas máquinas, pode ocorrer por exemplo ataques de abelhas devido ao barulho ou impacto de algum destroço em colméias[7], esse risco seria quase totalmente reduzido com o equipamento proposto neste projeto.

Se compararmos esse projeto com roçadeiras muito utilizadas a base de gasolina, teríamos um grande ganho energético, pois o equipamento funciona com bateria, podendo ser carregada após a utilização.

Um outro benefício muito claro é que devido ao fato do equipamento dispensar um operador. O custo de serviços desse tipo de jardinagem é potencialmente reduzido dado que esse serviço pode ser relativamente caro, principalmente quando se trata de gramados de estádios como, por exemplo o gramado do estádio Nacional de Brasília onde o custo de manutenção é de aproximadamente 100 mil reais por mês[8], boa parte desse custo é devido ao corte da grama, se houvesse um cortador autônomo provavelmente o custo do corte da grama seria reduzido devido a ausência de um operador barateando a operação.

O tempo em média que uma pessoa gasta para cortar a grama da sua casa é de 20 minutos, o fato é que muitas pessoas não tem esse tempo disponível 2 a 3 vezes na semana que é um das frequências mais utilizadas para cortar a grama, por isso recorrem a

outros meios como, por exemplo, calçadas de concreto o que prejudica o urbanismo local e o meio ambiente. Com o cortador autônomo proposto nesse projeto poderíamos dar a essas pessoas uma oportunidade de cuidar da grama de suas casas com um investimento de tempo extremamente menor uma vez que ele não precisaria operar a máquina.

O nosso projeto propõe implementar um cortador de grama que aumenta a segurança da atividade de jardinagem, tenha uma eficiência energética aceitável e uma eficiência operacional também considerável. Todos esses fatores trariam benefícios para aqueles que precisam desse serviço e que não se dispõem de tempo, ou até mesmo sejam leigas no ofício. Temos convicção que se desenvolvido esse projeto poderia contribuir para a saúde e o meio ambiente de forma muito eficiente trazendo segurança e qualidade para aqueles que precisam deste serviço.

5.**DESENVOLVIMENTO**

A Fim de cumprir com os requisitos do projeto descritos no item 4 foi feito um levantamento para encontrar os melhores sensores, capazes de adquirir os dados da forma mais eficiente para as tomadas de decisão, que serão responsáveis por controlar os motores de tração para a movimentação do cortador e o motor de corte, que será responsável por cortar a grama conforme anteriormente mencionado, além disso.

Nesse sentido foram desenvolvidos os primeiros códigos utilizando a linguagem C no ambiente do Code Composer Studio(CCS) software da empresa Texas Instruments e para ver os resultados utilizamos a comunicação serial RS232 via UART possibilitando fazer os melhores ajustes.

Após o levantamento dos sensores foi necessário projetar a estrutura do cortador que irá comportar todos os componentes e irá se tornar o produto propriamente dito, para isso procuramos a maioria das soluções existentes no mercado e pessoas que se propuseram a fazer algo parecido com o projeto aqui documentado.

5.1 **SENSORES**

Conforme as pesquisas bibliográficas feitas tivemos tivemos os seguintes resultados para a escolha dos sensores com o intuito de cumprir os requisitos conforme a tabela 2.

Tópico	Requisito	
Segurança	Desligamento do motor em caso de eventual capotamento ou tentativa de expor a lâmina.	Sensor IR
	Redução da velocidade da lâmina ao contornar obstáculos.	
	Aviso sonoro durante o funcionamento.	Buzzer
Eficiência energética	Autonomia de no mínimo 20 minutos.	Bateria
	Tempo de carregamento de no máximo 60 minutos.	Sensor Ultrassônico
Eficiência operacional	Percorrer 0,35 km/h.	Ponte H
	Desviar de obstáculo sem interverência humana.	
	Não permitir que a grama fique maior que 5cm.	Relé

Tabela 2. Sensores levantados em função dos requisitos.

.

5.1.1 **SENSOR IR**

O sensor óptico reflexivo, mais conhecido como sensor IR, é muito usado em projetos com microcontroladores, como robô segue-faixa, identificação de objetos reflexivos, entre outros. No projeto, foi utilizado o Módulo Sensor Óptico TCRT5000, Figura 2.

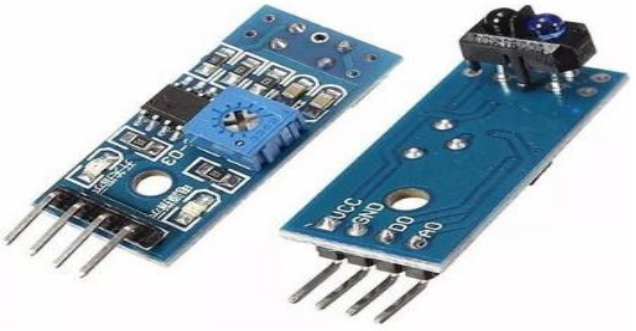


Figura 2. Módulo Sensor Óptico TCRT5000.

Este Sensor Óptico Reflexivo Fototransistor TCRT5000[11] de reflexão possui acoplado no mesmo dispositivo um sensor infravermelho (emissor) e um fototransistor (receptor). Foi especialmente projetado para bloquear outras faixas de luz que não seja a do próprio emissor, evitando que luminosidades do ambiente venham causar alguma interferência.

Integrando-o ao projeto, podemos descrever seu funcionamento da seguinte forma: alimentamos com +5V a entrada VCC e o GND com o terra do MSP430, a saída D0 é a resposta digital do sensor, onde D0=1 para objeto detectado e D0=0 para objeto não detectado. A saída A0 do módulo é a resposta analógica do sensor, não havendo necessidade de uso desta no projeto. Há também no módulo sensor, um potenciômetro para ajuda da distância de detecção de objeto, podendo variar de 8mm a 25mm.

5.1.2 **SENSOR ULTRASSÔNICO**

Um sensor muito utilizado na área da robótica para encontrar obstáculo e mensurar de forma rápida e com um nível razoável de precisão é o sensor ultrassônico para esse projeto será utilizado o módulo HC-SR04 que possui o seguinte funcionamento. Esse módulo possui em torno de 15º de ângulo de detecção razoável para a nossa aplicação.

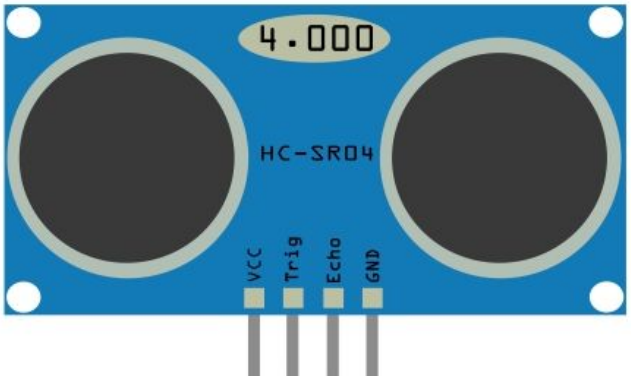


Figura 3. Simbologia do módulo HC-SR04.

Alimentamos o módulo com +5V no pino VCC e o GND ao terra do MSP, e então enviamos um pulso de duração de 10us para o pino trigger e então recebemos como resposta no pino echo um sinal de pulso onde o tempo em nível lógico alto é proporcional a distância de ida e volta que o som levou para percorrer encontrando um obstáculo.

Diagrama de tempo HC-SR04

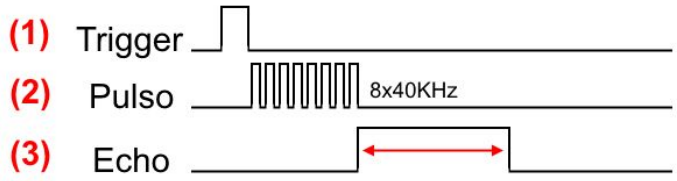


Figura 4. Diagrama de tempo HC-SR04

Conforme o datasheet do fabricante[10] a equação que define a distância em metros é:

$$d = \frac{Tempo\ de\ resposta\ do\ sensor \times Velocidade\ do\ som\,(340\ m/s)}{2}$$

Equação 1. distância do sensor ultrassônico em função do eco.

Aplicando esse sensor e suas especificações na launchpad do MSP430G25553 obtivemos sucesso conforme será visto no item 5.2

5.1.3 **MÓDULO PONTE H**

Para que pudéssemos controlar a velocidade e a direção dos motores de tração responsáveis pela movimentação do cortador, precisamos essencialmente de controlar o sentido dos motores qual motor deveria estar ligado e desligado para que pudesse ser feito o círculo responsável pela mudança da direção do cortador, além desses fatores precisamos alimentar os motores com uma fonte externa, pois a launchpad não conseguiria fornecer nos pinos a potência necessária para esse controle.

Nesse sentido optamos por utilizar um módulo de ponte H ou motor shield, que é em suma um circuito que permite o controle de motores CC, passo e servos, mudando sua direção, velocidade e no caso dos motores de passo a precisão também, isso tudo mudando os valores lógicos da entrada da ponte H, porém a alimentação dos motores será por meio de uma fonte externa algo que o módulo que escolhemos L298N permite fazer.

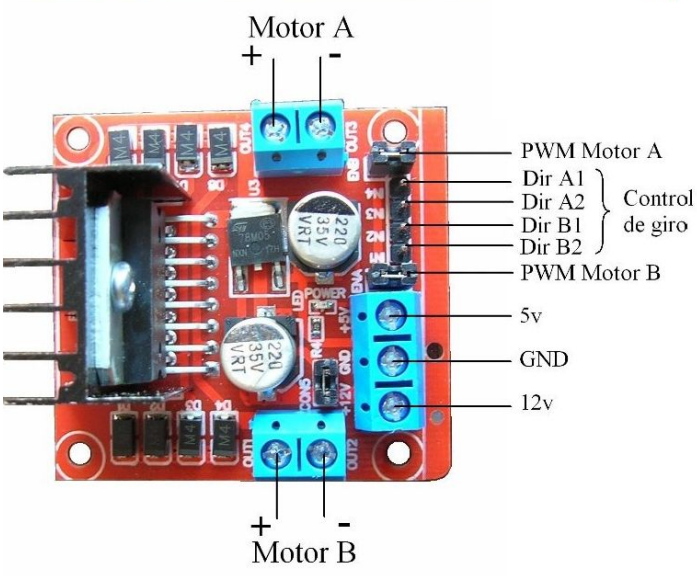


Figura 5. módulo de ponte H L298N.

Os motores são controlados conforme a seguinte tabela verdade

Giro	Dir A1	Dir A2
Horario	1	0
Antihorario	0	1
Motor detenido	0	0

Tabela 3. Tabela verdade do controle dos motores do módulo de ponte H L298N.

Ao implementar esse módulo com o MSP430G2553 obtivemos sucesso conforme será visto no item 5.2.

5.1.4 **MÓDULO RELÉ E BUZZER**

O relé se trata de uma chave eletromecânica que algo aplicarmos uma determinada tensão ele fecha o contato de forma eletromagnética, ou seja de isoladamente com o pino que provocou essa indução, possibilitando assim que com uma tensão pequena sejamos capazes de chavear circuitos que tem uma grande potência. Optamos por esse módulo invés de uma segunda ponte H, pois precisaríamos para controlar o motor de corte apenas da regulação Ligado/Desligado, não havendo a necessidade de controlar a velocidade ou até mesmo o sentido do motor de corte, esse módulo para funcionar corretamente basta alimentá-lo e então controlar de forma binária o chaveamento pelo pino IN. Possivelmente teremos que fazer um conversor de nível lógico pois os pinos do MSP são de no máximo 3.3V e o módulo funciona com uma tensão no pino IN de 5V.

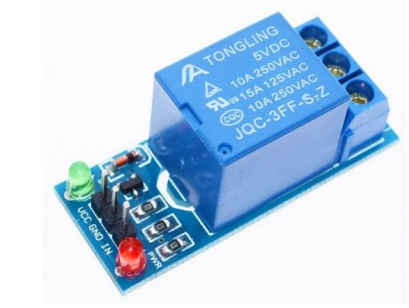


Figura 6. módulo relé.

O módulo buzzer nos permite gerar um alerta sonoro para um alarme conforme a aplicação desse projeto de forma muito simples, sem a necessidade de um sinal de áudio complexo o que dificultaria a implementação deste requisito. Para funcionar adequadamente, basta alimentar adequadamente com o VCC e o GND e controlar o áudio pela entrada I/O do módulo.

Figura 7. módulo buzzer passivo.

5.2 FUNCIONAMENTO DOS CÓDIGOS

Para a validação dos dados dos sensores utilizamos a comunicação serial, dessa forma foi possível verificar empiricamente se os resultados esperados no sensores estavam de acordo. Para a comunicação serial RS232 foram utilizados os pinos P1.1 e P1.2 respectivamente como RX e TX.

5.2.1 SENSOR IR

O código para implementação do sensor óptico reflexivo, foi feito apenas na Linguagem C, utilizado o software CCS para programação direta no MSP430. No mesmo, foram utilizadas apenas a biblioteca padrão "msp430g2553.h" do MSP430, e a biblioteca "legacymsp430.h", necessária para comunicação UART. Podemos ver na Figura 8, a o diagrama da implementação do sensor reflexivo.

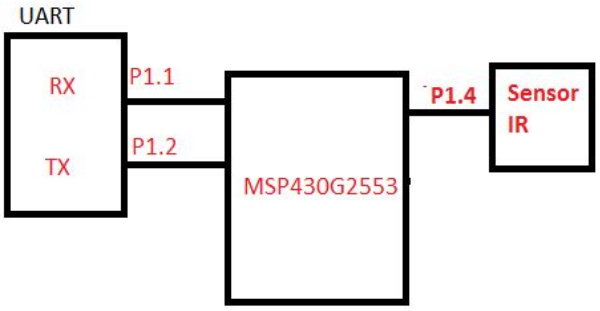


Figura 8. Implementação do Sensor Reflexivo.

O código presente pode se visto neste projeto no anexo Apêndice(A). Podemos verificar que o funcionamento consiste em utilizar a Launchpad para enviar a todo momento uma string para o computador via UART indicando o estado do SENSOR-IR em P1.4. Envia a string "DETECTADO" se o sensor detectar o objeto e "NAO DETECTADO" se o sensor não estiver detectando o objeto. Podemos ver as respostas do sensor sendo enviadas via UART na Figura 9.

Figura 9. Saída serial do MSP com o código do sensor IR.

5.2.2 SENSOR ULTRASSÔNICO

O código feito para implementar o sensor ultrassônico foi feito apenas na linguagem C (vide Apêndice B) , utilizando o apenas a biblioteca msp430g2553.h padrão da Texas Instruments para o microcontrolador em questão. Foi feita a seguinte conexão para a implementação conforme a figura 10.

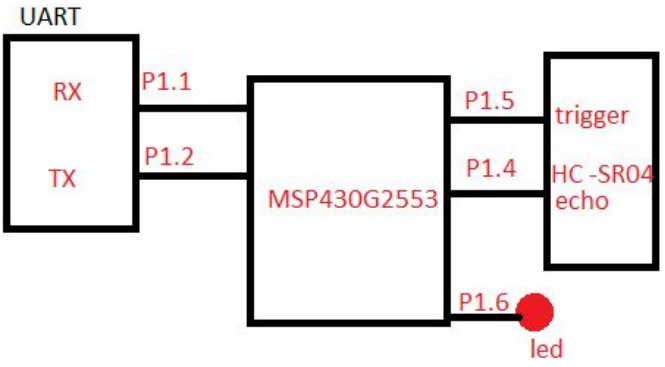


Figura 10. Implementação do SR04.

O intuito da implementação era basicamente através do módulo HC-SR04 obter a distância do sensor em relação a algum obstáculo e mostrar essa distância em cm na saída serial.

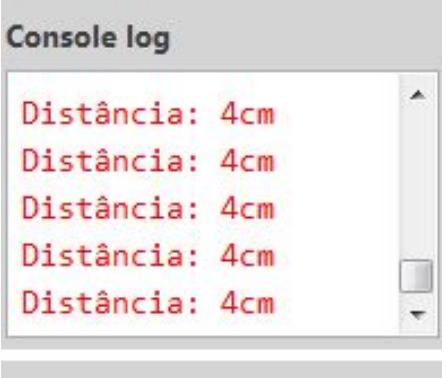


Figura 11. Saída serial do MSP com o código do sensor ultrassônico.

5.2.3 PONTE H

A montagem da ponte H para a implementação foi feito seguindo a figura 12.

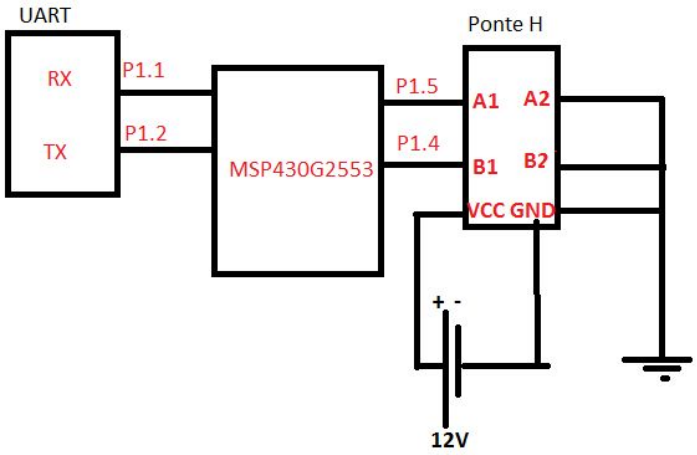


Figura 12. Implementação da ponte H L298N no MSP.

O código que foi implementado no MSP (vide Apêndice C) tinha o seguinte intuito, conforme o número que digitarmos na entrada serial RS232 teríamos o seguinte resultado:

1 = Virar a direita, liga apenas o motor da esquerda.

2 = Virar para esquerda, liga apenas o motor da direita

3 = Ir para frente, liga os dois motores.

qualquer outro número = Para os dois motores .

Figura 13. Resultado da implementação do código da ponte H na saída serial

Como pode ser visto a implementação foi um sucesso. O funcionamento dos motores pode ser visto no pelo seguinte link: Vídeo dos motores em fucionamento(https://drive.google.com/file/d/1067QDIo7K3LKBb8EQ9VrdizH9EYuQZuu/view?usp=sharing)

5.3 ESTRUTURA

Tomando os requisitos iniciais do projeto, montamos a base do chassi do cortador. A estrutura do chassi possui uma plataforma metálica para suporte da estrutura, uma roda boba para direção, dois motores de tração e duas rodas acopladas aos motores de tração.

A plataforma metálica possui uma área de aproximadamente 30cm x 15cm, e a roda boba de dimensões 6cm de diâmetro. Ambas podem ser vistas acopladas na Figura 14.

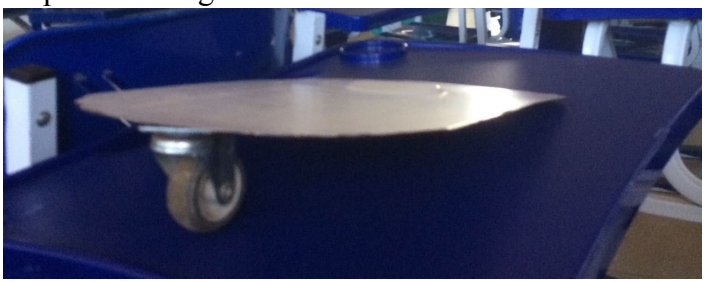


Figura 14. Plataforma Base + Roda Boba.

Os motores utilizados para tração foram motores de vidro elétrico de carro, modelo BOSCH 12V, vide Figura 15.

Figura 15. Motor Bosch Vidro Elétrico 12v.

Um dos embates na montagem da estrutura do cortador foi a escolha das rodas traseiras, pois estas deveriam ser resistentes, de diâmetros consideravelmente grandes. Todavia, as encontradas

para esta aplicação eram quase que financeiramente inviáveis. Porém com muita pesquisa e criatividade, decidimos construir manualmente rodas de madeiras para atender as necessidades do projeto. As rodas deveriam seguir as especificações da Figura 16, sendo madeira o material escolhido.

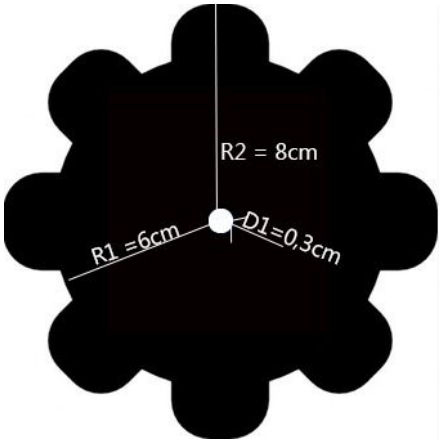


Figura 16. Modelo e Especificações das rodas traseiras.

Após a fabricação das rodas ocorrer com sucesso, pudemos acoplá-las aos motores e posteriormente, juntando-os na plataforma base do chassi. Nesse ponto podemos chegamos na seguinte estrutura.

Figura 17. Estrutura atual 1.

Figura 18. Estrutura atual 2.

Figura 19. Estrutura atual 3.

Onde ela foi testada e apresenta pleno funcionamento no que diz respeito à locomoção, e a tração que os motores conseguem suportar, além disso existe um espaço para o disco de corte que será integrado ao chassi, e também possui um espaço suficiente para a instalação dos sensores e da bateria que serão integrados.

6. REFERÊNCIAS

1. SILVANA, Teixeira. **Em que altura devo cortar a grama?** 2018. Disponível em: <https://www.cpt.com.br/cursos-jardinagem/artigos/em-que-altura-devo-cortar-a-grama>. Acesso em: 05 set. 2018.

2.PEDRÃO, Rodrigo. **GOAT (Cortador de Grama Autônomo) - Instituto Mauá de Tecnologia - Projeto TG.** 2014. Disponível em:

<https://www.youtube.com/watch?v=R-s7Zx80uL8>. Acesso em: 05 set. 2018.

3.FERRAMENTAS, Palácio das. **Cortador De Grama Costal Bateria 20v Wg169e Bivolt Worx:** Mercado Livre. 2018. Disponível em: <https://produto.mercadolivre.com.br/MLB-979366175-cortador-de-grama-costal-bateria-20v-wg169e-bivolt-worx-_JM>. Acesso em: 05 set. 2018.

4.AGROTAMA. **Roçadeira Lateral a Gasolina 72cc - RN72L - Nagano.** 2018. Disponível em: <https://www.agrotama.com.br/produtos/rocadeira-lateral-a-gasolina-72cc-rn72l/nagano-101021597,35,126/>. Acesso em: 05 set. 2018.

5.MERLIN, Leroy. **Cortador de Grama Gasolina MC-80G 3,75Hp Trapp.** 2018. Disponível em: <https://www.leroymerlin.com.br/cortador-de-grama-gasolina-mc-80g-3,75hp-trapp_85069404>. Acesso em: 05 set. 2018.

6.SUCURSAL, Edson Gil Junior da. **Ataque de abelhas fere sete pessoas e mata cachorro.** 2008. Disponível em: <https://www.gazetadopovo.com.br/vida-e-cidadania/ataque-de-abelhas-fere-sete-pessoas-e-mata-cachorro-b8aemkuqzd07f380164k9f2ha/>. Acesso em: 05 set. 2018.

7. SANHUDO, Jose Antonio Veiga. **O QUE VOCÊ DEVE SABER ANTES DE CORTAR GRAMA.** Disponível em: <http://www.clinicadope.net/index.php/dicas/21-se-voce-esta-pensando-em-cortar-grama-deveria-ler-isto>. Acesso em: 05 set. 2018.

8. LIMA, Marcos Paulo. **Gramado do Mané Garrincha é um dos mais caros do país, mas sofre com reclamações.** Disponível em: <https://www.df.superesportes.com.br/app/19,89/2015/06/08/noticia_futebol_nacional,60342/gramado-do-mane-garrincha-e-um-dos-mais-caros-do-pais-mas-sofre-com-reclamacoes.shtml>. Acesso em: 05 set. 2018.

9.JORNAL CRUZEIRO DO SUL. **Saiba qual o intervalo ideal para o corte e manutenção do gramado.** Disponível em: <http://www2.jornalcruzeiro.com.br/materia/776467/saiba-qual-o-intervalo-ideal-para-o-corte-e-manutencao-do-gramado>. Acesso em: 05 set. 2018.

10.ELEFREAKS. **Ltrasonic Ranging Module HC - SR04.** Disponível em: <https://www.mouser.com/ds/2/813/HCSR04-1022824.pdf>. Acesso em: 05 out. 2018.

11.SILICONIX, Vishay. **TCRT5000 Datasheet(PDF):** Vishay Siliconix. 2000. Disponível em: <http://html.alldatasheet.com/html-pdf/26406/VISHAY/TCRT5000/182/1/TCRT5000.html>. Acesso em: 05 out. 2018.

7. APÊNDICE

A.CÓDIGO PARA A DETECÇÃO DE OBSTÁCULO USANDO O SENSOR COM A COMUNICAÇÃO SERIAL

```
// Utiliza a Launchpad para enviar a todo momento
// uma string para o computador via UART indicando o estado
// do SENSOR-IR em P1.4. Envia a string "DETECTADO" se o sensor
// detectar o objeto e "NAO DETECTADO" se o sensor não estiver
// detectando o objeto.
//

#include <msp430g2553.h>
#include <legacymsp430.h>

// Definição de entrada do SENSOR e
// demais definições para a comunicação UART
#define SENSOR BIT4
#define RX BIT1
#define TX BIT2

#define BAUD_9600   0
#define BAUD_19200  1
#define BAUD_38400  2
#define BAUD_56000  3
#define BAUD_115200 4
#define BAUD_128000 5
#define BAUD_256000 6
#define NUM_BAUDS   7
```

```c
void Send_Data(unsigned char c);
void Send_String(char str[]);
void Init_UART(unsigned int baud_rate_choice);

int main(void)
{
    volatile int i = 0;
    WDTCTL = WDTPW + WDTHOLD; // Para WatchDogTimer

    // Escolha da frequência para 1MHz
    BCSCTL1 = CALBC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;

    P1OUT = SENSOR;     //Define resistor do BIT5 como PullUp
    P1REN |= SENSOR;    //Define que usará resistor no BIT4
    P1DIR &= ~SENSOR;   //Define SENSOR como entrada


    Init_UART(BAUD_9600);
    _BIS_SR(GIE);
    //Detecção de obstáculo e resposta pelo sensor
```

```c
    while(1)
    {

        if ((P1IN & SENSOR)==0)
            //Se SENSOR detectar obstáculo (SENSOR = 0)
            //envia string "DETECTADO"
            Send_String("\nDETECTADO\n");
        else
            //Se não detectar (SENSOR = 1) envia string "NAO DETECTADO"
            Send_String("\nNAO DETECTADO\n");

    }
    return 0;
}

//Função de envio de dados (utilizada em Send_String)
void Send_Data(unsigned char c)
{
    while((IFG2&UCA0TXIFG)==0);
    UCA0TXBUF = c;
}
```

```c
// Função para enviar string
void Send_String(char str[])
{
    int i;
    for(i=0; str[i]!= '\0'; i++)
        Send_Data(str[i]);
}

//COMUNICAÇÃO UART
void Init_UART(unsigned int baud_rate_choice)
{
    unsigned char BRs[NUM_BAUDS] = {104, 52, 26, 17, 8, 7, 3};
    unsigned char MCTLs[NUM_BAUDS] = {UCBRF_0+UCBRS_1,
                                      UCBRF_0+UCBRS_0,
                                      UCBRF_0+UCBRS_0,
                                      UCBRF_0+UCBRS_7,
                                      UCBRF_0+UCBRS_6,
                                      UCBRF_0+UCBRS_7,
                                      UCBRF_0+UCBRS_7};
    if(baud_rate_choice<NUM_BAUDS)
    {
        // Habilita os pinos para transmissao serial UART
        P1SEL2 = P1SEL = RX+TX;
        // Configura a transmissao serial UART com 8 bits de dados,
        // sem paridade, comecando pelo bit menos significativo,
        // e com um bit de STOP
        UCA0CTL0 = 0;
        // Escolhe o SMCLK como clock para a UART
        UCA0CTL1 = UCSSEL_2;
        // Define a baud rate
        UCA0BR0 = BRs[baud_rate_choice];
        UCA0BR1 = 0;
        UCA0MCTL = MCTLs[baud_rate_choice];
        // Habilita a interrupcao por chegada de dados via UART
        IE2 |= UCA0RXIE;
    }
}
```

B.CÓDIGO PARA A DETECÇÃO DA DISTÂNCIA UTILIZANDO O SENSOR ULTRASSÔNICO E COMUNICAÇÃO SERIAL

Welcome Guide | teste.py | blink.c | Telemetry C

```c
/*Código para a detecção da distância ultilizando o sensor ultrassônico
Ultilizando a comunicação serial para informar a distância o trigger
do sensor está ligado no pino p1.5 e o ecgo no p1.4. Ultilizando os
pinos p1.1 e p1.2 para respectivamente  o RX e TX da comunicação Init_UART
ultilzamos também um led no pino P1.6 para indicar o funcionamento do código

*/
#include <msp430g2553.h>

#define RX BIT1
#define TX BIT2
#define LED BIT6
#define trigger BIT5
#define echo BIT4
int miliseconds;
int distance;
long sensor;
#define BAUD_9600   0
#define BAUD_19200  1
#define BAUD_38400  2
#define BAUD_56000  3
#define BAUD_115200 4
#define BAUD_128000 5
#define BAUD_256000 6
#define NUM_BAUDS   7

void Send_Data(unsigned char c);
void Send_Int(int n);
void Send_String(char str[]);
void Init_UART(unsigned int baud_rate_choice);
```

```c
int main(void)
{
    volatile int i = 0;
    WDTCTL = WDTPW + WDTHOLD;

    BCSCTL1 = CALBC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;
    P1DIR |= LED;                           // P1.0 as output for LED
    P1OUT &= ~LED;
    P1OUT |= BTN;
    P1REN |= BTN;
    P1DIR &= ~BTN;
    CCTL0 = CCIE;                           // CCR0 interrupt enabled
    CCR0 = 1000;                 // 1ms at 1mhz
    TACTL = TASSEL_2 + MC_1;                // SMCLK, upmode
    CCTL0 = CCIE;                           // CCR0 interrupt enabled
    _BIS_SR(GIE);
    Init_UART(BAUD_9600);

    while(1)
    {
        P1IE &= ~0x01;      // disable interupt
        P1DIR |= trigger;         // trigger pin as output
        P1OUT |= trigger;         // generate pulse
        __delay_cycles(10);             // for 10us
        P1OUT &= ~trigger;                // stop pulse
        P1DIR &= ~echo;         // make pin P1.2 input (ECHO)
            P1IFG = 0x00;                 // clear flag just in case anything happened before
        P1IE |= echo;           // enable interupt on ECHO pin
        P1IES &= ~echo;         // rising edge on ECHO pin
```

```c
void Send_Data(unsigned char c)
{
    while((IFG2&UCA0TXIFG)==0);
    UCA0TXBUF = c;
}

void Send_Int(int n)
{
    int casa, dig;
    if(n==0)
    {
        Send_Data('0');
        return;
    }
    if(n<0)
    {
        Send_Data('-');
        n = -n;
    }
    for(casa = 1; casa<=n; casa *= 10);
    casa /= 10;
    while(casa>0)
    {
        dig = (n/casa);
        Send_Data(dig+'0');
        n -= dig*casa;
        casa /= 10;
    }
}
```

```c
void Send_String(char str[])
{
    int i;
    for(i=0; str[i]!= '\0'; i++)
        Send_Data(str[i]);
}

void Init_UART(unsigned int baud_rate_choice)
{
    unsigned char BRs[NUM_BAUDS] = {104, 52, 26, 17, 8, 7, 3};
    unsigned char MCTLs[NUM_BAUDS] = {UCBRF_0+UCBRS_1,
                                      UCBRF_0+UCBRS_0,
                                      UCBRF_0+UCBRS_0,
                                      UCBRF_0+UCBRS_7,
                                      UCBRF_0+UCBRS_6,
                                      UCBRF_0+UCBRS_7,
                                      UCBRF_0+UCBRS_7};
    if(baud_rate_choice<NUM_BAUDS)
    {
        // Habilita os pinos para transmissao serial UART
        P1SEL2 = P1SEL = RX+TX;
        // Configura a transmissao serial UART com 8 bits de dados,
        // sem paridade, comecando pelo bit menos significativo,
        // e com um bit de STOP
        UCA0CTL0 = 0;
        // Escolhe o SMCLK como clock para a UART
        UCA0CTL1 = UCSSEL_2;
        // Define a baud rate
        UCA0BR0 = BRs[baud_rate_choice];
        UCA0BR1 = 0;
        UCA0MCTL = MCTLs[baud_rate_choice];
```

C.Código para o controle da ponte H e comunicação serial

```c
/*
Código que realiza o controle da ponte H de dois motores CC para o controle
da direção. Para o controle par air para a direita ou esquerda para uma roda
e liga a outra e para ir para frente liga as duas rodas, além disso usa-se
a comunicação serial para a direção atual do cortador.


*/
#include <msp430g2553.h>
#include <legacymsp430.h>

#define RX BIT1 //  Bit para comunicação serial RX
#define TX BIT2 //  Bit para comunicação serial TX
#define P1MOTORA BIT5 //Pino para o motor A
#define P1MOTORB BIT4 //Pino para o motor B

//Definições para comunicação uart
#define BAUD_9600   0
#define BAUD_19200  1
#define BAUD_38400  2
#define BAUD_56000  3
#define BAUD_115200 4
#define BAUD_128000 5
#define BAUD_256000 6
#define NUM_BAUDS   7
```

```c
//Cabeçario para as funções---------------
void Send_Data(unsigned char c);
void Send_Int(int n);
void Send_String(char str[]);
void Init_UART(unsigned int baud_rate_choice);
//----------------------------------------

volatile char direcao = 'P'; //Variável global

int main(void)
{
    volatile int i = 0; // Count

    WDTCTL = WDTPW + WDTHOLD; // Desliga o WDT

    //Definições para o timer A
    BCSCTL1 = CALBC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;
    //-------------------------

    //Inicializando os pinos dos motores1
    P1OUT &= ~P1MOTORA;
    P1OUT &= ~P1MOTORB;
    P1DIR |= P1MOTORA;
    P1DIR |= P1MOTORB;
    //----------------------
```

```c
    //Habilita interrupções---------------
    Init_UART(BAUD_9600);
    _BIS_SR(GIE);
    //----------------------------------

    while(1)
    {
        // Trecho para imprimir a string na serial conforme o valor de direcao---
        switch(direcao)
        {
        case 'P':
            Send_String("Direção Atual: Parado\n");
        break;
        case 'D':
             Send_String("Direção Atual: Direita\n");
        break;
        case'E':
             Send_String("Direção Atual: Esquerda\n");
        break;
        case 'F':
             Send_String("Direção Atual: Frente\n");
        break;
        default:
            Send_String("Direção Atual: Parado\n");
        //-----------------------------------------------------------------
        __delay_cycles(10000);; //Aplica um delay para o envio na serial
        }
     }
    return 0;
}
```

```c
//Função para a comunicação serial
void Send_Data(unsigned char c)
{
    while((IFG2&UCA0TXIFG)==0);
    UCA0TXBUF = c;
}
//--------------------------------


//Função para enviar a string----------
void Send_String(char str[])
{
    int i;
    for(i=0; str[i]!= '\0'; i++)
        Send_Data(str[i]);
}
//---------------------------------------
```

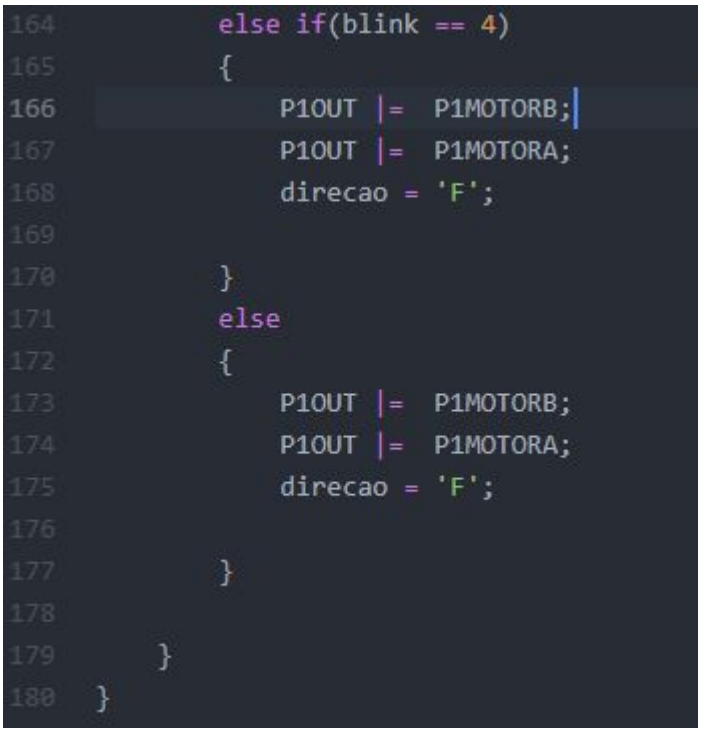
```c
//Função para ultilizar a UART------------------
void Init_UART(unsigned int baud_rate_choice)
{
    unsigned char BRs[NUM_BAUDS] = {104, 52, 26, 17, 8, 7, 3};
    unsigned char MCTLs[NUM_BAUDS] = {UCBRF_0+UCBRS_1,
                                      UCBRF_0+UCBRS_0,
                                      UCBRF_0+UCBRS_0,
                                      UCBRF_0+UCBRS_7,
                                      UCBRF_0+UCBRS_6,
                                      UCBRF_0+UCBRS_7,
                                      UCBRF_0+UCBRS_7};
    if(baud_rate_choice<NUM_BAUDS)
    {
        // Habilita os pinos para transmissao serial UART
        P1SEL2 = P1SEL = RX+TX;
        // Configura a transmissao serial UART com 8 bits de dados
        // sem paridade, comecando pelo bit menos significativo,
        // e com um bit de STOP
        UCA0CTL0 = 0;
        // Escolhe o SMCLK como clock para a UART
        UCA0CTL1 = UCSSEL_2;
        // Define a baud rate
        UCA0BR0 = BRs[baud_rate_choice];
        UCA0BR1 = 0;
        UCA0MCTL = MCTLs[baud_rate_choice];
        // Habilita a interrupcao por chegada de dados via UART
        IE2 |= UCA0RXIE;
    }
}
```

```c
interrupt(USCIAB0RX_VECTOR) Receive_Data(void)
{
    unsigned char blink = UCA0RXBUF - '0';
    //Lógica para ligar e desligar os motores
    if(blink<10)
    {
        if(blink == 0)
        {
            P1OUT &= ~P1MOTORA;
            P1OUT &= ~P1MOTORB;
            direcao = 'P';

        }
        else if(blink == 1)
        {
            P1OUT |=  P1MOTORA;
            P1OUT &= ~P1MOTORB;
            direcao = 'E';

        }
        else if(blink == 2)
        {
            P1OUT |=  P1MOTORB;
            P1OUT &= ~P1MOTORA;
            direcao = 'D';


        }
```



```
164             else if(blink == 4)
165             {
166                 P1OUT |=  P1MOTORB;
167                 P1OUT |=  P1MOTORA;
168                 direcao = 'F';
169 
170             }
171             else
172             {
173                 P1OUT |=  P1MOTORB;
174                 P1OUT |=  P1MOTORA;
175                 direcao = 'F';
176 
177             }
178 
179         }
180     }
```