



**LUIZ FERNANDO MEDEIROS ARAÚJO**

**DESENVOLVIMENTO DE UM SISTEMA  
EMBARCADO PARA CONTROLE DE UM  
ROBÔ MÓVEL**

**LAVRAS – MG**

**2013**

**LUIZ FERNANDO MEDEIROS ARAÚJO**

**DESENVOLVIMENTO DE UM SISTEMA EMBARCADO PARA  
CONTROLE DE UM ROBÔ MÓVEL**

Monografia apresentada ao Colegiado do Curso de  
Ciência da Computação, para a obtenção do título  
de Bacharel em Ciência da Computação.

Orientador

Prof. DSc. Wilian Soares Lacerda

**LAVRAS – MG**

**2013**

**LUIZ FERNANDO MEDEIROS ARAÚJO**

**DESENVOLVIMENTO DE UM SISTEMA EMBARCADO PARA  
CONTROLE DE UM ROBÔ MÓVEL**

Monografia apresentada ao Colegiado do Curso de  
Ciência da Computação, para a obtenção do título  
de Bacharel em Ciência da Computação.

APROVADA em 19 de Abril de 2013.

Prof. MSc. Luciano Mendes dos Santos

UFLA

Prof. MSc. Thomaz Chaves de Andrade Oliveira

UFLA



Prof. DSc. Wilian Soares Lacerda

(Orientador)

**LAVRAS – MG**

**2013**

*Dedico esta monografia aos meus pais Fernando e Rosângela.*

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, por me conceder inteligência e força para conseguir terminar este importante trabalho em minha vida.

Aos meus amados pais Fernando e Rosângela pelo apoio e confiança.

Aos meus irmãos Thiago e Isabella por terem me apoiado e pelos sábios conselhos durante esta etapa.

A minha vó vicentina pelo exemplo de vida e os ensinamentos. Em especial aos meus avós que me deixaram ao longo dessa caminhada.

A minha namorada Nayara pelo amor incondicional e pela paciência que teve comigo durante diversos momentos.

Aos meus amigos de curso André, Igor, Francisco e Guilherme pelo apoio e companheirismo durante esta graduação.

Ao professor Wilian Lacerda pela paciência e conhecimentos passados para realização deste trabalho.

Aos companheiros da república Boléia Brunão, Pablo e Daniel.

Aos demais funcionários e professores do DCC a gratidão pelo apoio e pelos conhecimentos que me foram passados nesta universidade.

**MUITO OBRIGADO!**

## RESUMO

A necessidade de se criar formas autônomas para controle de robôs móveis, faz com que seja necessário o desenvolvimento de mecanismos capazes de realizar tal função, uma dessas formas é por meio da utilização de um sistema embarcado. Este trabalho apresenta o desenvolvimento de um sistema embarcado para controle de um robô de rodas que é capaz de seguir uma linha fixada no chão, desviar de obstáculos presentes no trajeto e retornar novamente a seguir a linha. Para realização desse trabalho utilizou-se sensores de infravermelho para a identificação da linha a ser seguida, um sensor de ultrassom HC-SR04 para identificação dos obstáculos, controlou-se os motores utilizando um driver L298 e o utilizou-se o microcontrolador PIC 18F4550 para executar o *software* que controla e processa as informações provenientes dos sensores. Além da utilização apresentada neste trabalho, este sistema pode ser utilizado em diversas aplicações, como exemplos, uma cadeira de rodas automática e um automóvel controlado sem intervenção humana.

**Palavras-Chave:** Sistema embarcado; Microcontrolador; Robô móvel.

## **ABSTRACT**

The need to create ways to control autonomous mobile robots, makes it necessary to develop mechanisms capable of performing this function, one of these ways is through the use of an embedded system. This paper presents the development of an embedded system to control a robot wheel that is able to follow a line fixed on the ground, and dodge obstacles in the path and back again to follow the line. To carry out this project we used infrared sensors to identify the line to be followed, the obstacles are identified by an ultrasonic sensor HC-SR04, engine control is done using a driver L298 and 18F4550 PIC microcontroller is used to run software that controls and processes information from the sensors. Besides the use presented here, this system can be used in various applications, ranging from a automatically wheelchair to a vehicle controlled without human intervention.

**Keywords:** Embedded System; Microcontroller; Mobile robot.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	Contextualização e Motivação . . . . .	12
1.2	Objetivos do trabalho . . . . .	13
1.2.1	Objetivo geral . . . . .	13
1.2.2	Objetivos específicos . . . . .	13
1.3	Organização do trabalho . . . . .	14
<b>2</b>	<b>SISTEMAS EMBARCADOS E OS COMPONENTES ENVOLVIDOS</b>	<b>15</b>
2.1	Sistemas Embarcados . . . . .	15
2.2	Microcontrolador . . . . .	16
2.2.1	Microcontrolador <i>PIC</i> . . . . .	17
2.2.2	<i>PIC18F4550</i> . . . . .	18
2.3	Linguagem C . . . . .	18
2.4	Compilador MikroC . . . . .	20
2.5	Sensores . . . . .	21
2.5.1	Sensores de ultrassom . . . . .	21
2.5.2	Sensor de ultrassom HC-SR04 . . . . .	22
2.5.3	Sensores de Luz e infravermelho . . . . .	23
2.5.4	Sensores de toque . . . . .	24
2.6	Atuadores . . . . .	24
2.6.1	Motor DC . . . . .	25
2.6.2	Servomotor . . . . .	26
2.6.3	<i>Display LCD</i> . . . . .	27
<b>3</b>	<b>ROBÓTICA MÓVEL E COMPONENTES UTILIZADOS EM RO- BÓTICA</b>	<b>28</b>
3.1	Robótica móvel . . . . .	28



3.1.1	Ponte H . . . . .	30
3.1.2	Driver L298 . . . . .	31
3.1.3	Circuito regulador de tensão LM7805 . . . . .	32
3.1.4	PWM . . . . .	32
3.1.5	Diodo de roda livre . . . . .	34
<b>4</b>	<b>METODOLOGIA</b>	<b>35</b>
4.1	Montagem da parte mecânica . . . . .	35
4.2	Montagem da parte eletrônica . . . . .	36
4.2.1	Montagem sensor infravermelho . . . . .	38
4.3	Programa e testes . . . . .	38
<b>5</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>40</b>
5.1	Hardware de controle do robô . . . . .	40
5.2	Hardware do sensor de infravermelho . . . . .	43
5.3	Programa de controle do robô . . . . .	43
<b>6</b>	<b>CONCLUSÃO</b>	<b>47</b>
6.1	Contribuições do trabalho . . . . .	47
6.2	Pontos negativos . . . . .	47
6.3	Trabalhos futuros . . . . .	48
<b>A</b>	<b>Anexos</b>	<b>51</b>
A.1	Anexo A - Controle_ robo.c . . . . .	51
A.2	Anexo B- Esquemático da placa de controle do robô . . . . .	58

## LISTA DE FIGURAS

1.1	Esquema de ligação dos componentes de <i>hardware</i> . . . . .	13
2.1	Componentes de <i>hardware</i> de um sistema embarcado - Fonte:(WILMSHURST, 2007) . . . . .	16
2.2	Distribuição dos pinos do PIC18F4550 - Fonte:(MICROCHIP, 2009) . . . . .	19
2.3	Estrutura de um programa linguagem C - Fonte:(LAUREANO, 2005) . . . . .	19
2.4	Ambiente do <i>MikroC</i> . . . . .	20
2.5	Exemplo de funcionamento ultrassom . . . . .	22
2.6	Diagrama de ondas geradas por um sensor de ultrassom - Fonte:(WILMSHURST, 2007) . . . . .	22
2.7	Sensor ultrassom HC-SR04 - Fonte:(CYTRON, 2012) . . . . .	23
2.8	Princípio de funcionamento do sensor infravermelho - Fonte: (WILMSHURST, 2007) . . . . .	24
2.9	Exemplo de motor DC com caixa de redução - Fonte: (LOVINE, 2004) . . . . .	25
2.10	Diagrama funcionamento servomotor - Fonte: (WILMSHURST, 2007) . . . . .	26
2.11	Exemplo de servomotor - Fonte: (BRÄUNL, 2006) . . . . .	27
3.1	Exemplos robô de rodas - Fonte:(BRÄUNL, 2006) . . . . .	29
3.2	Desenho do robô utilizado no projeto . . . . .	29
3.3	Representação de uma Ponte H . . . . .	30
3.4	Circuito integrado L298- Fonte:(STMICROELECTRONICS, 2000) . . . . .	31
3.5	Exemplos de ondas de <i>PWM</i> geradas com diferentes larguras de pulso - Fonte:(BRÄUNL, 2006) . . . . .	33
3.6	Esquema de funcionamento diodo de roda livre - Fonte:(WILMSHURST, 2007) . . . . .	34
4.1	Robô montado . . . . .	35
4.2	Desenho esquemático simplificado do robô . . . . .	37

4.3	Esquemático do sensor infravermelho . . . . .	38
4.4	Fluxograma do programa de controle do robô . . . . .	39
5.1	Vista superior da placa montada . . . . .	41
5.2	<i>Led e Display LCD</i> indicando o modo de segurança . . . . .	42
5.3	Mensagens mostradas no <i>display LCD</i> pelo robô . . . . .	42
5.4	Sensor infravermelho montado . . . . .	43
5.5	Visualização em osciloscópio da onda de <i>PWM</i> gerada . . . . .	44
5.6	Vista frontal do robô com sua placa de controle . . . . .	45
5.7	Vista superior do robô com sua placa de controle . . . . .	45

## LISTA DE TABELAS

3.1	Valores de operação do L298 - Fonte:(STMICROELECTRONICS, 2000) .	32
3.2	Características elétricas do LM7805 - Fonte:(FAIRCHILD, 2012) . . . .	32
4.1	Componentes eletrônicos utilizados no projeto . . . . .	36
5.1	Estimativa de preço dos componentes eletrônicos utilizados . . . . .	46

# 1 INTRODUÇÃO

## 1.1 Contextualização e Motivação

Atualmente, quase todo produto que necessita de algum tipo de controle possui um sistema embarcado. Sistemas embarcados são dispositivos computacionais de propósito específico, geralmente integrados a um sistema externo para desempenhar uma função específica (SIQUEIRA *et al.*, 2006).

Robôs móveis são pequenos robôs que possuem grande número de sensores e atuadores e são controlados por um pequeno sistema informático, onde os controles são feitos a bordo desse robô (BRÄUNL, 2006).

Estes robôs podem ser controlados de diversas formas utilizando diversos componentes eletrônicos. Uma dessas formas é por meio da utilização de sistema embarcado incorporado a um microcontrolador. Neste trabalho desenvolveu-se um sistema embarcado, utilizando microcontrolador *PIC* acoplado a um robô móvel no intuito de controlar as funções relacionadas ao robô como, por exemplo, velocidade dos motores de locomoção e a direção do robô.

Além dessa utilização, esse tipo de sistema pode ser facilmente adaptado a um automóvel para detectar e prevenir colisões, ou até mesmo em uma cadeira de rodas elétrica para que essa possa funcionar de forma automática desviando obstáculos e seguindo uma determinada direção.

Os dispositivos de *hardware* principais que utilizou-se nesse trabalho são o microcontrolador PIC 18F4550 e o circuito integrado L298. O PIC18F4550 destinou-se ao controle e gerenciamento do sistema embarcado, enquanto o L298 atuou no controle dos motores de locomoção do robô. Na figura 1.1 apresenta-se o esquema de ligação entre componentes do sistema embarcado.



**Figura 1.1:** Esquema de ligação dos componentes de *hardware*

## 1.2 Objetivos do trabalho

Nesta seção apresenta-se os objetivos do presente trabalho, bem como os esforços necessários para alcançá-lo.

### 1.2.1 Objetivo geral

O presente trabalho tem como foco o desenvolvimento de um sistema embarcado para controle de um robô móvel capaz de seguir uma linha e desviar de obstáculos encontrados durante o percurso, através da utilização de um microcontrolador *PIC*, sensores e atuadores. Para identificação dessa linha utilizou-se um sensor infravermelho, o obstáculo identificou-se com um sensor de ultrassom e para a locomoção do robô utilizou-se 4 motores DC.

### 1.2.2 Objetivos específicos

Para obtenção do objetivo geral apresentado na seção 1.2.1 são necessários:

- Estudar e entender os conceitos envolvidos com sistema embarcado e robótica móvel.
- Elaborar o projeto de hardware que servirá de suporte para o sistema embarcado.
- Realizar a montagem e teste desse *hardware* e interligá-lo ao robô móvel.
- Desenvolver o programa para ser executado neste sistema embarcado.

### **1.3 Organização do trabalho**

Este trabalho encontra-se dividido em 6 capítulos. No capítulo 1 são apresentados os conceitos, a motivação e as justificativas para realização desse trabalho e a contextualização do tema. No capítulo 2 são apresentadas algumas definições e conceitos envolvendo sistemas embarcados e os componentes presentes em sistemas embarcados. No capítulo 3 contêm as definições sobre robótica móvel e as características dos itens de *hardware* envolvidos. A metodologia e os métodos são apresentados no capítulo 4. Os resultados obtidos com o presente trabalho são apresentados no capítulo 5. Por fim, no capítulo 6 apresenta-se a conclusão e as discussões relacionadas ao trabalho.

## 2 SISTEMAS EMBARCADOS E OS COMPONENTES ENVOLVIDOS

Neste capítulo serão introduzidos conceitos e definições envolvidos com sistemas embarcados, bem como os componentes utilizados em sistemas embarcados.

### 2.1 Sistemas Embarcados

Segundo Kamal (2008), sistema embarcado é um sistema que tem incorporado *software* e *hardware* de computador, que faz parte de um sistema dedicado a um aplicativo ou parte específica de um sistema ou aplicação maior.

Outra definição para sistemas embarcados é apresentada por Wilmshurst (2007), como sendo um sistema cuja principal função não é computacional, mas é controlado por um computador incorporado em seu interior.

Segundo Reis (2004), o termo “embarcado” deve-se ao fato de que estes sistemas são projetados para que funcionem de forma independente de uma fonte de energia externa. Eles são formados pelos mesmos componentes que um computador normal, ou seja, processador, memória e dispositivos de interfaces. A principal diferença entre um sistema embarcado e um computador, é que ele executa bem apenas uma tarefa, enquanto os computadores executam diversas tarefas.

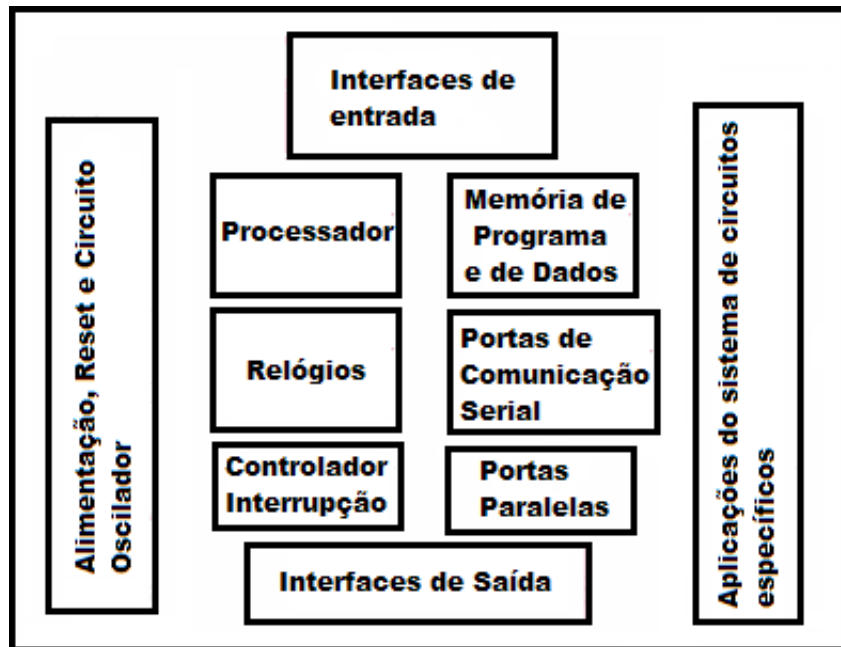
Os sistemas embarcados chegaram para formar a época pós-PC. O crescente número de sistemas embarcados deve-se ao fato do aumento no número de aplicações que se apoiam nesses sistemas. As ferramentas e tecnologias atuais possuem diversas limitações, assim se faz necessário o estudo e desenvolvimento desses sistemas.

Atualmente os mais diversos produtos possuem sistemas embarcados, por mais que esses tipos de produtos possam parecer diferentes, em todos eles o sistema embarcado desempenha funções parecidas. Até mesmo a geladeira domés-



tica é um exemplo de produto que possui um sistema embarcado incorporado, nesse caso sua função é controlar a temperatura e acionar o compressor caso necessário (WILMSHURST, 2007).

Na figura 2.1 é possível visualizar todos componentes de *hardware* de um sistema embarcado.



**Figura 2.1:** Componentes de *hardware* de um sistema embarcado - Fonte:(WILMSHURST, 2007)

## 2.2 Microcontrolador

Microcontrolador é um único chip, que possui um sistema inteiro encapsulado em seu interior. Ele possui características semelhantes às de um computador pessoal. O microcontrolador contém uma CPU, memória RAM, dispositivos de entrada e saída, memória ROM e o conversor analógico digital (LOVINE, 2004).

Para Souza (2006), microcontrolador pode ser definido como um componente eletrônico, programável que é capaz de realizar controle de processos ló-

gicos. Este controle de processos pode ser definido como controle de periféricos dentre os quais têm destaque os *LEDs*, *displays*, sensores e resistências.

O termo “programável” deve-se ao fato que toda lógica de operação é estruturada na forma de um programa e é gravada dentro do componente. Assim toda vez que ele é alimentado novamente, seu programa interno é então executado. Sua “inteligência”, está associada a Unidade Lógica e Aritmética(U<sub>LA</sub>), é nessa unidade que todas as operações matemáticas e lógicas são executadas. Desta forma, quanto mais poderosa for a sua U<sub>LA</sub>, maior será sua capacidade de processamento (SOUZA, 2006).

Segundo Kamal (2008), microcontroladores são computadores de baixo custo. Sua capacidade de executar e armazenar programas exclusivos os tornam extremamente versáteis. Além disso, eles podem executar funções matemáticas e lógicas sofisticadas, e circuitos eletrônicos complexos.

Analisando fisicamente um microcontrolador, é possível visualizar um circuito integrado ladeado por uma certa quantidade de pinos que varia de acordo com o tipo e modelo do microcontrolador. Esses pinos são utilizados para os mais diversos fins, dentre os quais destacam-se alimentação, oscilador, portas de entrada e saída, *reset* e controle (LOVINE, 2004).

### **2.2.1 Microcontrolador PIC**

O termo *PIC* define um conjunto de microcontroladores fabricados pelo *MICROCHIP*. Esses microcontroladores são característicos por apresentarem estrutura interna *Harvard*, enquanto os demais apresentam arquitetura *Von-Neuman*. A diferença entre estas duas arquiteturas está no funcionamento do barramento, na arquitetura de *Von-Neuman* existe apenas um único barramento para dados e instruções. Já a arquitetura *Harvard* apresenta dois barramentos separados para dados e instruções (SOUZA, 2006).

O tipo de instrução que é utilizada pelos microcontroladores *PIC*, é a tecnologia do tipo *RISC*, que consiste de um número de instruções reduzido. Os microcontroladores *PIC* possuem um conjunto com cerca de 35 instruções. Os demais microcontroladores podem chegar a mais de cem instruções.

O número reduzido de instruções exige um esforço maior do programador quando necessário a criação de funções mais complexas.

### **2.2.2 PIC18F4550**

Segundo a Microchip (2009), o PIC18F4550 é um microcontrolador que possui incorporado memória *flash* de 32Kbytes e USB 2.0, além do suporte de comunicação com alta e baixa velocidade para todos os tipos transferência de dados suportados.

Ainda, a Microchip (2009) ressalta que essa família de microcontroladores, possuem recursos capazes de diminuir o consumo energia durante sua operação. O PIC18F4550 possui ainda 12 opções de osciladores, possibilitando o desenvolvimento de uma ampla variedade de aplicações.

O PIC18F4550 e os demais microcontroladores pertencentes a sua família possuem um conjunto de instruções estendidas, o que aumenta o número de instruções em oito. Além disso, ele possui um módulo comunicação padrão RS232 e fornece suporte para o protocolo de barramento *LIN* (MICROCHIP, 2009).

Quanto a distribuição dos pinos, o PIC18F4550 possui 13 canais para conversão de sinal analógico digital, além das portas de entrada e saída R(A,B,C,D,E). O conversor A/D possui a resolução de 10 bits. Na figura 2.2 apresenta-se a distribuição dos pinos e as portas correspondentes a cada pino.

## **2.3 Linguagem C**

Segundo Laureano (2005) a linguagem C, é uma linguagem estruturada e padronizada que foi criada na década de 1970, que tem como ponto forte sua

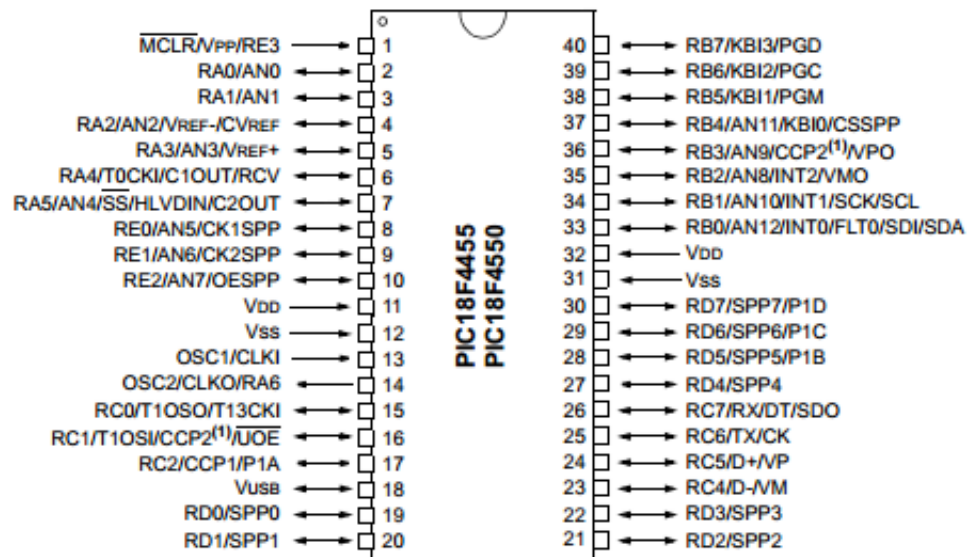


Figura 2.2: Distribuição dos pinos do PIC18F4550 - Fonte:(MICROCHIP, 2009)

eficiência e é a linguagem mais utilizada para desenvolvimento de sistemas operacionais.

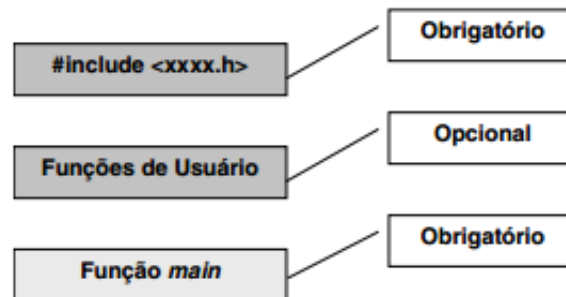


Figura 2.3: Estrutura de um programa linguagem C - Fonte:(LAUREANO, 2005)

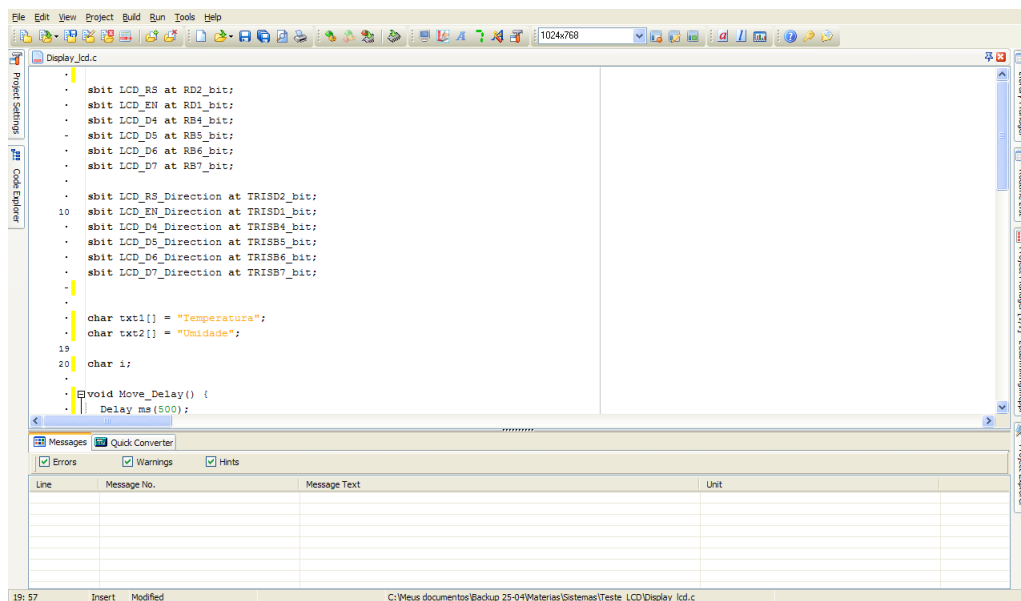
Na figura 2.3 é apresentada a estrutura de um programa básico na linguagem C, são três blocos principais utilizados. No primeiro bloco são feitos todos os *includes* necessários para o programa, o segundo bloco contém as funções de

usuários, que neste caso são opcionais, e por fim o último bloco é bloco principal, no qual é definida a função *main* (LAUREANO, 2005).

A linguagem C é utilizada por alguns compiladores para microcontroladores, o *MikroC* apresentado na próxima seção, é um desses compiladores que à utilizam.

## 2.4 Compilador MikroC

O *MikroC* é um compilador para a linguagem C, criado e mantido pela *Mikroelektronika* para programação de microcontroladores *PIC* das famílias PIC 12, PIC 16 e PIC 18.



**Figura 2.4:** Ambiente do *MikroC*

Segundo a MIKROELEKTRONIKA (2006), o *MikroC* é um ambiente de desenvolvimento integrado sofisticado, que possui diversos recursos para microcontroladores PIC, tornando assim mais fácil o desenvolvimento de aplicações para sistemas embarcados. O ambiente do *MikroC* é apresentado na figura 2.4.

O *PIC* é um microcontrolador conhecido mundialmente, usado em diversas aplicações, e a linguagem C é conhecida por sua eficiência. O *MikroC* é o resultado desta junção (MIKROELEKTRONIKA, 2006).

## 2.5 Sensores

Segundo Wilmshurst (2007), existe uma enorme infinidade de sensores disponíveis, passando por sensores antigos chegando até os mais modernos existentes. O funcionamento desses sensores é baseado em um ou mais fenômenos físicos, capazes de converter a variável física para elétrica. A eletromecânica, ótica e a ultrassônica são alguns desses fenômenos físicos.

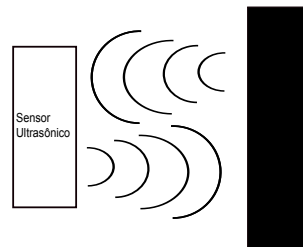
A escolha dos sensores varia de acordo com a necessidade da aplicação e com as diferentes características envolvidas com cada sensor. Essas características irão determinar a performance, economia e aplicabilidade de cada sensor (NIKU, 2001).

### 2.5.1 Sensores de ultrassom

O ultrassom é utilizado para detecção e medição, por exemplo, para determinar uma simples distância e na obtenção de imagens médicas de alta complexidade (WILMSHURST, 2007).

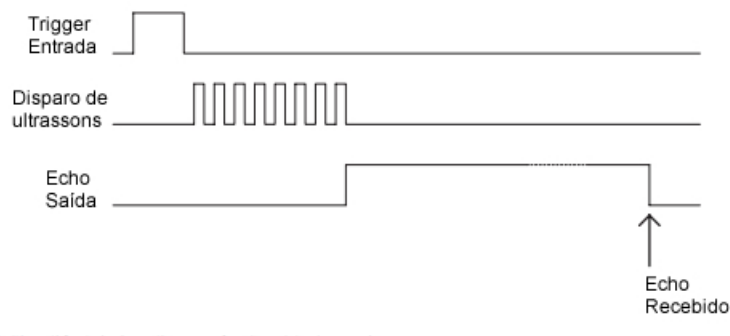
O sensor de ultrassom baseia-se no envio constante de sinais. De acordo com o tempo que este sinal demora para retornar ao sensor, é possível determinar a distância entre o sensor e o elemento onde o sinal foi refletido. A figura 2.5 ilustra o funcionamento de um sensor de ultrassom.

Na figura 2.6 estão representadas três formas de ondas. O primeiro formato de onda representa a entrada ou *Trigger* do sensor de ultrassom, no qual é feito pulso inicial de disparo para o sensor começar a disparar os sinais de ultrassom. Estes sinais são representados pela segunda forma de onda. A terceira forma



**Figura 2.5:** Exemplo de funcionamento ultrassom

de onda representa a saída ou *echo* do sensor, em que é enviado o pulso indicando a ocorrência de algum obstáculo.



**Figura 2.6:** Diagrama de ondas geradas por um sensor de ultrassom - Fonte:(WILMSHURST, 2007)

Segundo Niku (2001), os sensores de ultrassom emitem ondas de som de alta frequência, geralmente 200KHz. Estes sensores podem operar em dois modos diferentes: modo contrário ou modo difuso(eco). No modo contrário, o receptor e emissor são colocados frente a frente. Enquanto no modo difuso, o emissor e receptor são integrados.

### 2.5.2 Sensor de ultrassom HC-SR04

Segundo a Cytron (2012), o HC-SR04 é um módulo ultrassônico composto por um receptor e transmissor que funciona utilizando o mesmo princípio

de sonar que possuem os golfinhos e morcegos para determinar a distância de um objeto ou obstáculo.

Este sensor opera com uma tensão de 5V DC, seu consumo é de 15mA em operação e aproximadamente 2mA em repouso. A largura de pulso para disparo é de 10uS e seu alcance varia de 2cm a 400 cm (CYTRON, 2012).



**Figura 2.7:** Sensor ultrassom HC-SR04 - Fonte:(CYTRON, 2012)

Conforme a figura 2.7, é possível notar que o HC-SR04 possui 4 pinos: *VCC*, *Trig*, *Echo* e *GND*. O pino *VCC* deve ser alimentando com uma tensão de 5V e o *GND* deve ser ligado diretamente ao terra. O *Trig* deve receber um pulso com valor alto por 10uS, quando isso ocorre o mesmo gera no pino *Echo* uma sequência de ultrassons com frequência de 40KHz e ao encontrar um obstáculo, é gerado um pulso. Assim, para determinar a distância do sensor ao obstáculo é necessário calcular a largura desse pulso e ajustar esse valor utilizando uma regra de três simples, no intuito de obter a distância em sua respectiva unidade.

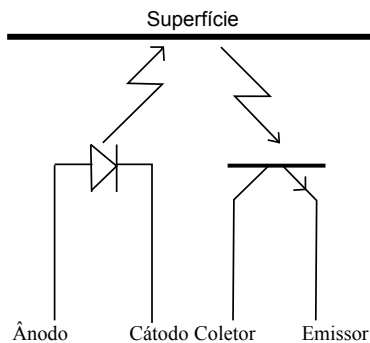
### 2.5.3 Sensores de Luz e infravermelho

Niku (2001) define sensor de luz como sendo um sensor que altera sua resistência de acordo com a intensidade de luz que ele recebe. A resistência funciona de forma inversamente proporcional à intensidade da luz recebida pelo sensor. Assim quanto maior for a intensidade, menor será a resistência, portanto, maior a corrente.



O sensor infravermelho é um tipo especial de sensor de luz, enquanto os sensores de luz em geral reconhecem a luz visível humana, o sensor infravermelho reconhece somente a gama dos infravermelhos.

Segundo Wilmshurst (2007), o sensor infravermelho funciona com um par *Led* infravermelho e fototransistor montados lado a lado. Quando a luz emitida pelo infravermelho é refletida pela superfície, o fototransistor conduz. Na figura 2.8 ilustra-se o princípio de funcionamento do sensor infravermelho.



**Figura 2.8:** Princípio de funcionamento do sensor infravermelho - Fonte: (WILMSHURST, 2007)

### 2.5.4 Sensores de toque

Segundo Niku (2001), sensores de toque são sensores que enviam algum sinal quando ocorre algum contato físico. Esse tipo de sensor pode ser utilizado em um robô móvel como um detector de obstáculos durante a navegação, caso encontre um obstáculo, um sinal é enviado para o sistema de controle do robô.

## 2.6 Atuadores

Segundo Wilmshurst (2007), atuadores são responsáveis por realizar os movimentos elétricos, lineares ou rotativos. Existe uma infinidade de atuadores, desde aqueles capazes de mover toneladas de peso até atuadores minúsculos. Estes

últimos têm destaque importante na medicina, já que eles podem ser incorporados ao corpo humano, por exemplo, atuando no controle de algum medicamento (MARWEDEL, 2003).

### 2.6.1 Motor DC

Motor DC é um tipo especial de motor em que, o estator é um conjunto de ímãs fixos criando um campo magnético fixo, e o rotor carrega a corrente (NIKU, 2001).

Wilmshurst (2007) ressalta que os motores DC são utilizados em uma gama de aplicações que vão desde a movimentação de trens elétricos até minúsculos sistemas mecatrônicos. Na figura 2.9 apresenta-se um exemplo de motor DC semelhante ao utilizado nesse projeto.



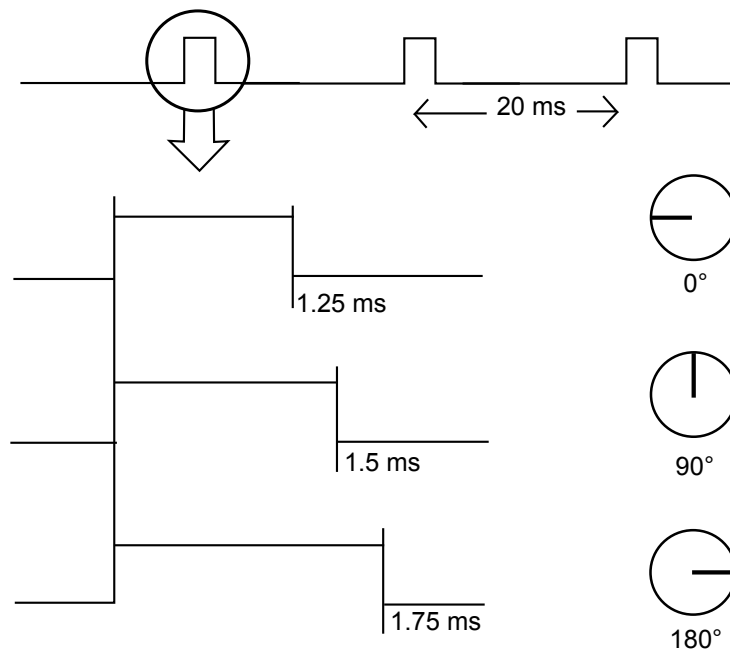
**Figura 2.9:** Exemplo de motor DC com caixa de redução - Fonte: (LOVINE, 2004)

Segundo Bräunl (2006), o método mais comum de locomoção de robôs móveis é por meio dos motores DC, além disso, esses motores são limpos, silenciosos e capazes de produzir energia para uma variedade de tarefas

## 2.6.2 Servomotor

Segundo Lovine (2004), servomotor são motores que possuem uma orientação de controle de realimentação de posicionamento, o que possibilita que o rotor possa ser posicionado de forma precisa. Ainda, Lovine (2004) ressalta que os servomotores são utilizados na grande maioria dos modelos aviões, carros e barcos rádio-controlados. Devido a este mercado consumidor existe uma infinidade de modelos disponíveis para utilização.

Para Bräunl (2006) servomotor é um motor de corrente contínua com alta qualidade para ser utilizado em malhas fechadas, capaz de lidar com mudanças rápidas de posição, velocidade e aceleração.



**Figura 2.10:** Diagrama funcionamento servomotor - Fonte: (WILMSHURST, 2007)

Na figura 2.10 está representado a posição do servomotor de acordo com o pulso recebido por ele. No caso dessa representação, para que o servomotor esteja num ângulo de 90 graus, por exemplo, basta enviar um pulso de 1.5ms. A largura

do pulso para cada ângulo varia conforme o modelo do servomotor. Outro fator importante para o funcionamento do servomotor é o tempo entre os pulsos, nesse caso utiliza-se 20ms. Na figura 2.11 apresenta-se um exemplo de servomotor.



**Figura 2.11:** Exemplo de servomotor - Fonte: (BRÄUNL, 2006)

### 2.6.3 *Display LCD*

O display de cristal líquido (LCD) é umas das tecnologias que contribuíram para a evolução eletrônica tornando-se um importante componente em celulares, *notebooks* e *tablets*.

O cristal líquido é um composto que é capaz de polarizar qualquer luz que passe através dele, esse composto é capaz de responder a aplicação de um campo elétrico alterando o alinhamento das moléculas ou mudando a direção da polarização da luz. Geralmente esse cristal é colocado entre duas placas de vidro paralelas (WILMSHURST, 2007).

Neste trabalho, utilizou-se um *Display LCD* comum em diversos projetos em eletrônica, composto de 16 colunas por 2 linhas. Esse *Display* foi ligado no modo paralelo utilizando 4 bits de dados.

### 3 ROBÓTICA MÓVEL E COMPONENTES UTILIZADOS EM ROBÓTICA

Neste capítulo são apresentadas algumas definições sobre robótica e os itens de *hardware* envolvidos com o presente trabalho.

#### 3.1 Robótica móvel

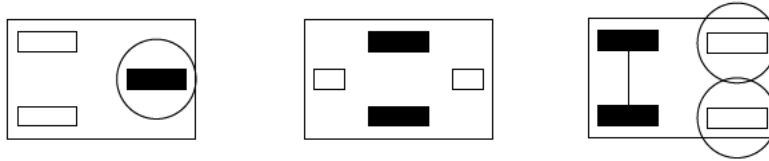
Segundo WOLF *et al.* (2009) a relevância da robótica móvel nas últimas décadas é devido ao grande número de aplicações apoiadas neste conceito, bem como os interesses econômicos envolvidos em relação ao seu desenvolvimento. Ainda, WOLF *et al.* (2009) ressalta a importância da computação na potencialização deste robôs, possibilitando sistemas mais robustos e inteligentes.

Segundo Bräunl (2006), com o advento da robótica os robôs móveis também sofreram diversas mudanças. Antigamente os robôs eram controlados por sistemas computacionais grandes e pesados, assim esses sistemas não podiam ser transportados junto ao robô fazendo necessário a utilização de cabos ou outros dispositivos. Atualmente os robôs podem ser controlados por um pequeno e leve sistema a bordo do próprio robô.

Segundo Jung *et al.* (2005) a robótica móvel tem evoluído baseado em 4 aspectos: locomoção, percepção, controle e inteligência e por fim a comunicação. A locomoção é baseada no uso dos atuadores, que são responsáveis pela execução das ações. Já a percepção baseia-se no uso de sensores para identificação do ambiente. Controle e inteligência definem o sistema de controle que será incorporado junto ao nível de inteligência. A comunicação é importante para a troca de informações entre os robôs e a sua interação com outros robôs.

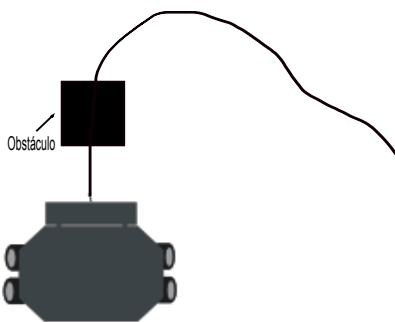
O tipo mais simples de robô é o robô de rodas, esse tipo possui duas ou mais rodas motrizes e exigem pelo menos dois motores, um para condução da roda

e outro para viragem (BRÄUNL, 2006). Mostra-se na figura 3.1 alguns exemplos de robôs de rodas.



**Figura 3.1:** Exemplos robô de rodas - Fonte:(BRÄUNL, 2006)

Na figura 3.1 são apresentados três tipos diferentes de robô. O robô a esquerda possui apenas uma única roda de viragem. O robô central possui um sistema chamado “drive diferencial”, ou seja, é feita a combinação de duas rodas motrizes, esse é um dos modelos mais utilizados em robôs móveis. O robô mais à direita possui um sistema de direção diferencial, onde ambas as rodas traseiras são combinadas e encarregadas de determinar a direção do robô.



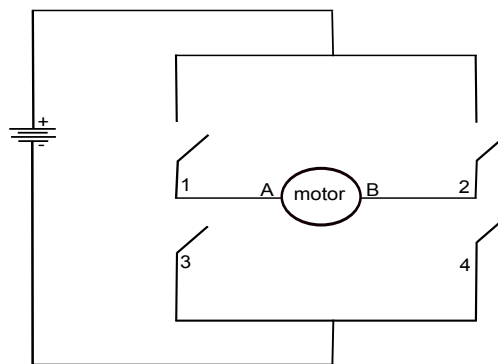
**Figura 3.2:** Desenho do robô utilizado no projeto

Neste projeto utilizou-se um robô de rodas com 4 motores DC. Cada motor é ligado a uma respectiva roda. Apresenta-se na figura 3.2 um modelo desse robô. Os motores serão interligados de forma lateral.

### 3.1.1 Ponte H

Um problema conhecido no controle utilizando motores é a alternância de polaridade para realização de uma mudança de direção. É desejável que essa mudança seja realizada com apenas dois bits de informação, utilizando a mudança no fluxo da corrente (NIKU, 2001).

A *ponte H* é um mecanismo capaz de inverter o sentido de rotação de um motor, permitindo que o motor possua dois sentidos de rotação. O funcionamento da ponte H é apresentado na figura 3.3 à seguir.

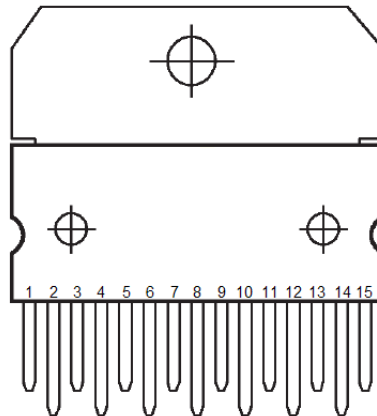


**Figura 3.3:** Representação de uma Ponte H

O circuito apresentado na figura 3.3 possui quatro chaves numerada de 1 a 4 e representa-se os terminais do motor pelas letras A e B. Essas chaves podem ser ligadas de duas maneiras diferentes: 1 e 4 ou 2 e 3. As chaves são ligadas, sempre alternadas, possibilitando que o motor possa girar em dois sentidos diferentes. Quando as chaves 1 e 4 estão ligadas, o terminal A é ligado ao polo positivo da bateria e o terminal B é ligado ao polo negativo. Já quando as chaves 2 e 3 são ligadas, o terminal A é ligado ao polo negativo da bateria e o terminal B ao polo positivo. Existem circuitos eletrônicos capazes de realizar a ponte H, um destes dispositivos é o driver L298 apresentado na seção seguinte.

### 3.1.2 Driver L298

Segundo STMicroelectronics (2000) o L298 é um circuito integrado monolítico que possui uma ponte dupla para acionamento de relés, solenóides, motores DC e de passo. O driver L298 possui quatro entradas e quatro saídas. Ainda, possui dois canais para acionamento dessas entradas. A tensão de operação deste circuito é de até 46V e corrente de no máximo 4A(STMICROELECTRONICS, 2000). Ao se utilizar o driver L298 é necessário o uso de diodos de roda livre, apresentado na seção 3.1.5.



**Figura 3.4:** Circuito integrado L298- Fonte:(STMICROELECTRONICS, 2000)

Conforme a figura 3.4 é possível notar a dimensão das distribuição de todos os pinos presentes no driver L298. Os pinos 1 e 15 são utilizados para controlar a corrente da carga ligada ao driver. Os pinos 5,7,10 e 12 correspondem às entradas, enquanto os pinos 2,3,13 e 14 são as respectivas saídas. A alimentação lógica do integrado é feita no pino 9, alimentação da carga no pino 4 e o pino 8 é o terra ou GND. Os pinos 6 e 11 são responsáveis pela ativação ou desativação das saídas do driver. Apresenta-se na tabela 3.1 os valores máximos para o correto funcionamento do driver.



**Tabela 3.1:** Valores de operação do L298 - Fonte:(STMICROELECTRONICS, 2000)

<b>Símbolo</b>	<b>Parâmetro</b>	<b>Valor</b>	<b>Unidade</b>
$V_S$	Tensão de alimentação	50	V
$V_{SS}$	Tensão de alimentação lógica	7	V
$V_I$	Tensão de alimentação da carga	-0.3 até 7	V
$V_{sens}$	Sensor de tensão	-1 até 2.3	V
$P_{tot}$	Potência total dissipada	25	W
$T_{op}$	Temperatura de operação	-25 até 130	°C

### 3.1.3 Circuito regulador de tensão LM7805

Segundo Fairchild (2012), esse regulador é capaz fornecer uma tensão fixa de saída positiva de 5V e fornecer na saída a corrente de 1A. Esse CI possui ainda um sistema de proteção interno para evitar que ocorra sua queima. O LM7805 é composto por três terminais. No pino 1 deve ser ligado a tensão de entrada, o pino 2 é ligado ao terra e o pino 3 fornece a tensão de saída.

Na tabela 3.2 são apresentadas algumas características elétricas que possui o circuito integrado LM7805.

**Tabela 3.2:** Características elétricas do LM7805 - Fonte:(FAIRCHILD, 2012)

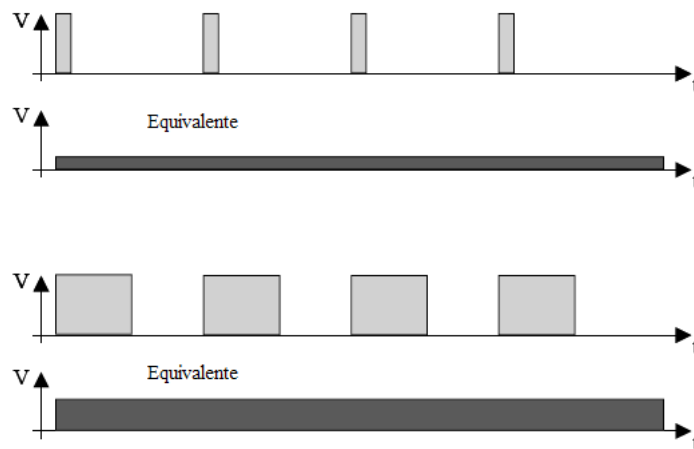
<b>Símbolo</b>	<b>Características</b>	<b>Mín.</b>	<b>Típ.</b>	<b>Máx.</b>	<b>Unidade</b>
$V_O$	Tensão de saída	4.8	5.0	5.2	V
$I_Q$	Corrente em repouso		5.0	8.0	mA
$V_N$	Tensão de ruído na saída		42.0		$\mu$ V
$r_O$	Resistência de saída		15.0		m $\Omega$

### 3.1.4 PWM

Segundo Wilmshurst (2007), PWM, ou modulação por largura de pulso, pode ser definida como uma técnica capaz de controlar componentes analógicos a

partir de uma saída digital. O sinal de PWM pode ser gerado por *software* ou por *hardware*.

Bräunl (2006) exemplifica a utilização do PWM para controle de um motor DC. Ao se utilizar um sinal analógico para controlar a velocidade do motor é preciso variar a tensão para que isso ocorra. Por meio do uso de PWM isso é feito gerando vários pulsos numa mesma frequência, desta forma a velocidade do motor varia de acordo com a largura deste pulso.



**Figura 3.5:** Exemplos de ondas de *PWM* geradas com diferentes larguras de pulso - Fonte:(BRÄUNL, 2006)

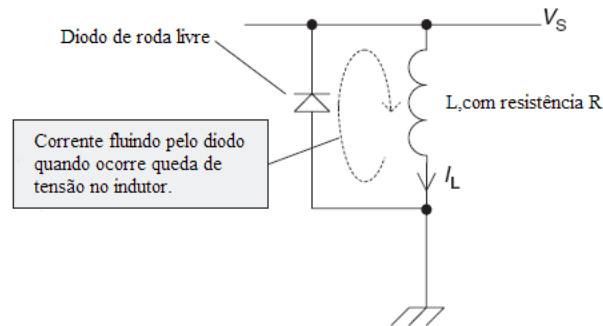
A figura 3.5 ilustra dois exemplos de onda PWM. A primeira possui uma largura de pulso menor, nesse caso o motor teria uma velocidade menor em relação à velocidade obtida com a segunda forma de onda. Alguns microcontroladores possuem a capacidade de geração do sinal de PWM por *hardware*. O microcontrolador PIC é um exemplo deles, nesse caso existem funções pré-definidas para realização de tal função.

### 3.1.5 Diodo de roda livre

Segundo Wilmshurst (2007) a relação de tensão pela corrente é dada pela seguinte fórmula:

$$V = -L \cdot \frac{di}{dt},$$

onde  $V$  é a tensão que passa pelo indutor,  $i$  a corrente que passa pelo indutor e  $L$  sua indutância. Quando ocorre uma queda de tensão no indutor, a corrente aumenta de forma exponencial num curto espaço de tempo, gerando uma tensão muito alta.



**Figura 3.6:** Esquema de funcionamento diodo de roda livre - Fonte:(WILMSHURST, 2007)

A função do diodo de roda de livre é descarregar a energia armazenada no indutor, evitando-se assim a queima dos demais componentes. O diodo de roda livre deve ser ligado em paralelo e polarizado inversamente. A figura 3.6 ilustra o esquema de ligação deste diodo ao circuito.

## 4 METODOLOGIA

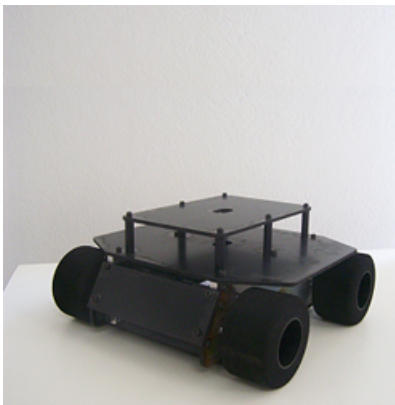
Nesta seção serão apresentadas como foram feitas as montagens das partes mecânica e eletrônica. Também, apresentam-se o desenvolvimento e testes que foram realizados no *software*.

A parte de *hardware* foi desenvolvida utilizando placas fenolite pré-fabricadas com trilhas de cobre, que contém a furação para soldagem dos componentes que se utilizou neste projeto. Os componentes foram interligados utilizando fios de cobre e solda composta por 60% estanho e 40% chumbo.

A montagem e os testes foram realizados no laboratório de eletrônica do departamento de Ciência da Computação da Universidade Federal de Lavras. Os componentes eletrônicos e o *kit* que contém a parte mecânica do robô foram obtidos junto ao departamento de Ciência da Computação.

### 4.1 Montagem da parte mecânica

Para montagem da parte mecânica utilizou-se um *kit* da *Lynxmotion* modelo *Predator SR2-KT* que contém toda parte estrutural e mecânica para montagem do robô que foi utilizado no presente trabalho. Na figura 4.1 apresenta-se o robô após a montagem realizada.



**Figura 4.1:** Robô montado

O robô *Predator SR2-KT* em sua montagem original possui 6 motores DC 12V, porém neste projeto foram removidos os 2 motores medianos de cada lado, os 4 motores restantes foram interligados forma lateral e em paralelo, desta forma o robô pode mover-se em varia direções.

## 4.2 Montagem da parte eletrônica

Para a montagem desta etapa foram utilizados alguns componentes eletrônicos, listados na tabela 4.1.

**Tabela 4.1:** Componentes eletrônicos utilizados no projeto

Quantidade	Descrição
1	microntrolador PIC18F4550
1	driver L298
1	crystal oscilador 20MHz
1	módulo LCD MGD1602B
1	soquete de 40 pinos para o PIC18F4550
3	capacitores eletrolíticos de 100 $\mu$ F
3	capacitores cerâmicos 100nF
2	capacitores cerâmicos 10pF
3	diodos do tipo LED
2	resistores de 1 $\Omega$
2	resistores de 220 $\Omega$
1	resistor de 1,2K $\Omega$
2	resistores 4,7K $\Omega$
5	resistores de 10K $\Omega$
2	fototransistores 3mm TIL 78
2	diodos <i>led</i> infravermelho 5mm TIL 32
1	trimpot com resistência 10k $\Omega$
3	botões do tipo <i>Push button</i>
1	sensor ultrassom HC-SR04
1	servomotor futaba modelo S3003

Esses componentes foram devidamente soldados na placa de fenolite e interligados de acordo com as informações contidas no esquemaço mostrado na figura 4.2.

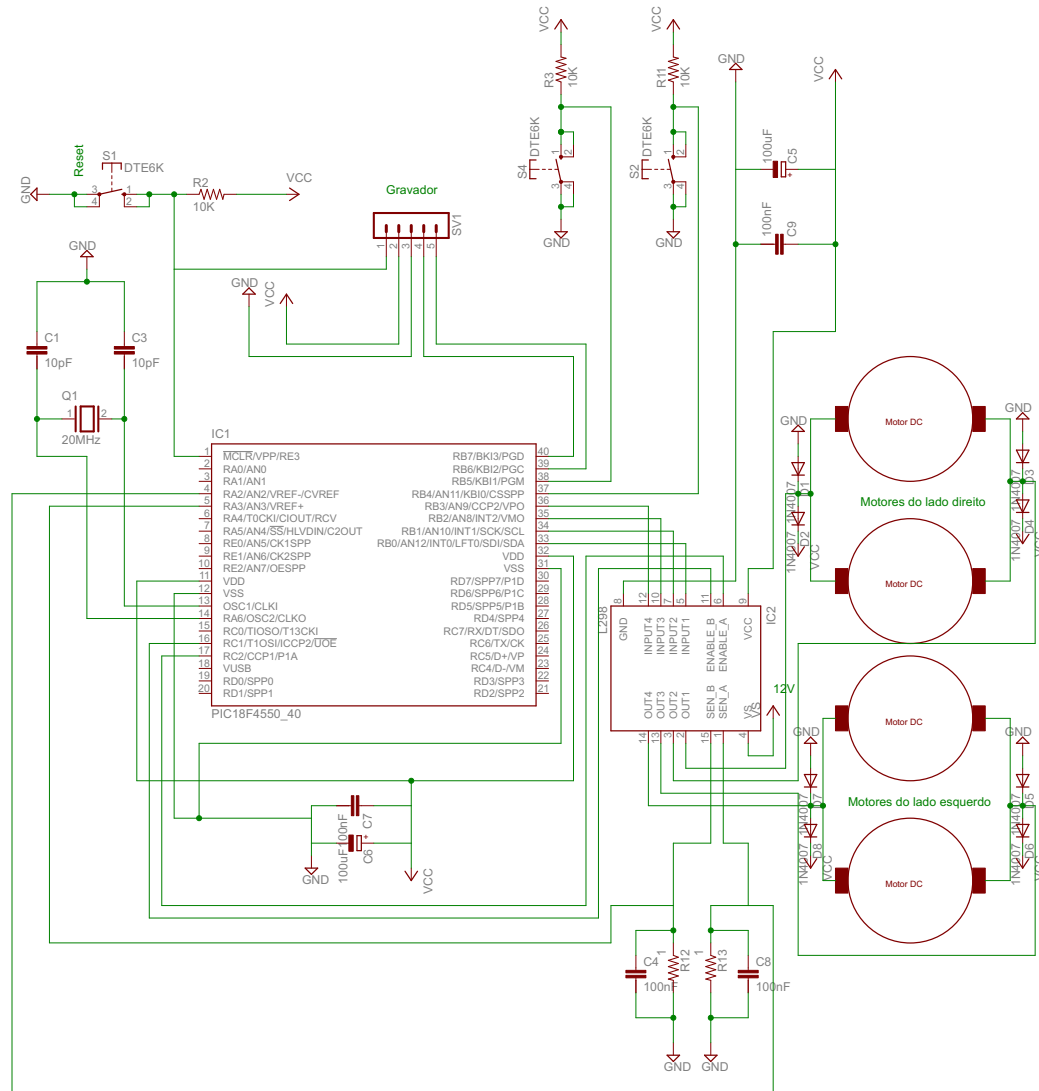
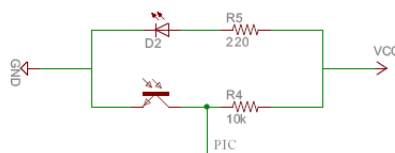


Figura 4.2: Desenho esquemático simplificado do robô

Na figura 4.2 é apresentado o esquemático simplificado da placa de controle do robô com o microcontrolador ligado ao driver L298. Mostram-se, também, os botões do tipo *push button* utilizados para alguns testes durante o projeto.

#### 4.2.1 Montagem sensor infravermelho

O sensor infravermelho foi montado utilizando um par de diodo e fototransistor modelo TIL32 e TIL 78, respectivamente. Juntos, eles formam um par emissor e receptor de infravermelho. O emissor TIL32 foi ligado ao resistor de  $220\Omega$  e o fotodiodo TIL78 foi ligado ao resistor de  $10K\Omega$ . Na figura 4.2 é apresentado o esquemático utilizado para a montagem do sensor infravermelho.

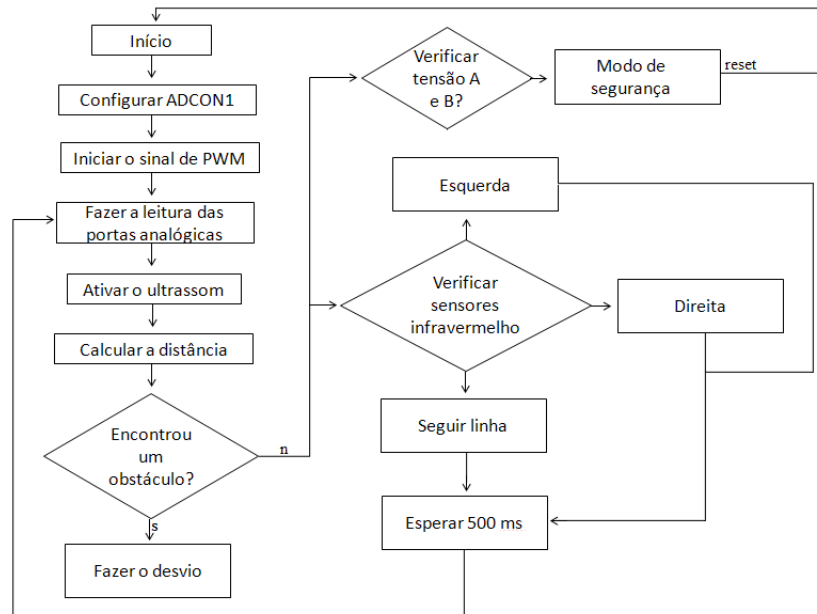


**Figura 4.3:** Esquemático do sensor infravermelho

### 4.3 Programa e testes

Após o desenvolvimento do programa utilizando a linguagem C com o compilador *MikroC PRO*, esse programa é gravado no microcontrolador *PIC*. Para essa gravação utilizou-se um gravador de *PIC* modelo *MicroICD* e o programa responsável por fazer essa gravação é o *PICKit2* na versão 2.61. Na figura 4.4 apresenta-se o fluxograma do programa responsável por controlar o robô.

O programa inicialmente gera os sinais *PWM* que são utilizados para o acionamento dos motores, em seguida é gerado um pulso de 10us para ativação do sensor de ultrassom HC-SR04, posteriormente é feita a leitura do conversor analógico digital que está ligado aos sensores de infravermelho. De acordo com essa leitura o microcontrolador define qual deve ser a direção tomada pelo robô.



**Figura 4.4:** Fluxograma do programa de controle do robô

Caso o sensor de ultrassom detecte um obstáculo à uma distância menor que 5cm do robô, realiza-se o desvio deste obstáculo. Em seguida, o robô procura a linha novamente e ao encontrá-la, o processo descrito anteriormente é reiniciado.

Para o melhor desempenho do robô, foram realizados diversos testes no intuito de calibrá-lo e realizar a correção de possíveis falhas ocorridas durante o desenvolvimento do *software*.



## 5 RESULTADOS E DISCUSSÃO

Neste capítulo serão apresentados os resultados obtidos com o desenvolvimento do sistema embarcado de controle do robô, bem como, o *hardware* e o *software* obtidos.

### 5.1 Hardware de controle do robô

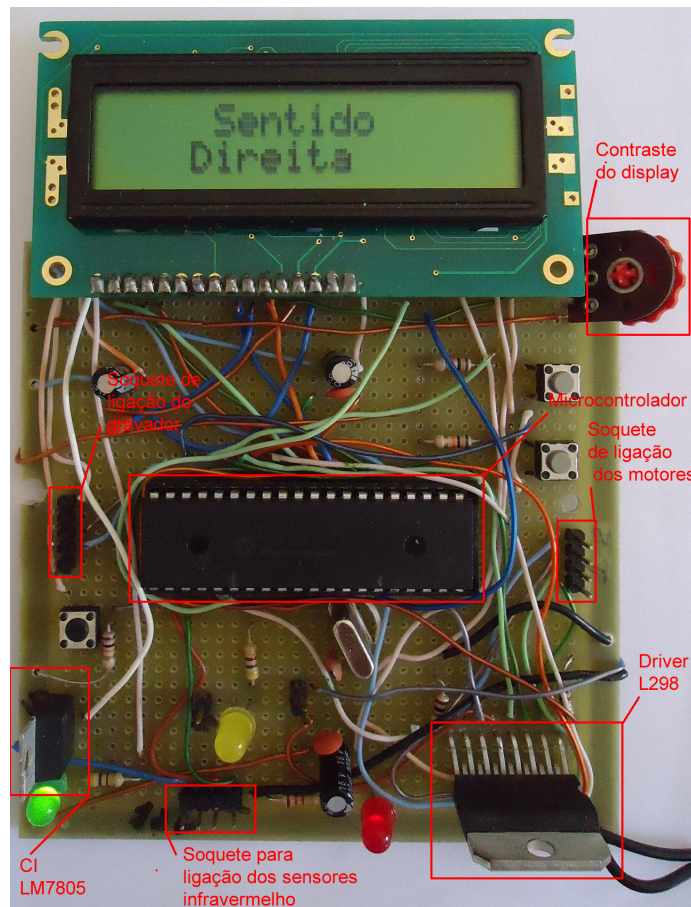
A placa ilustrada na figura 5.1, contém a indicação dos componentes utilizados no projeto. Esta placa foi elaborada seguindo o esquemático apresentado no anexo B e a sua função é controlar o robô e seus periféricos.

O microcontrolador PIC 18F4550 foi colocado na parte central da placa para facilitar a ligação com os demais componentes colocados ao redor da placa. Esse microcontrolador foi destinado a executar o programa desenvolvido, bem como realizar as demais tarefas atribuídas a ele pelo programa. No canto inferior da placa foi colocado o driver L298, responsável pela ligação e acionamento dos motores. Este driver é ligado ao microcontrolador, que por sua vez envia sinais para determinar qual o sentido de rotação dos motores. Nos pinos 1 e 15, do L298, foram ligados resistores de  $1\Omega$ .

O sinal de *clock* para o microcontrolador foi gerado utilizando um cristal oscilador externo de 20MHz, ligado em paralelo com dois capacitores de 10pF. Esses capacitores garantem estabilidade e potencializam a frequência gerada pelo oscilador.

Para o *reset* do microcontrolador foi utilizado um botão do tipo *push button* ligado no pino 1 do microcontrolador. Outros dois botões desse mesmo tipo foram utilizados para realização de testes durante o desenvolvimento do projeto.

Foram utilizadas diferentes tensões de alimentação. Para a alimentação do microcontrolador e a parte lógica dos demais componentes utilizou-se uma tensão de 5V. Os motores foram alimentados com uma tensão de 12V. Assim todo o sis-



**Figura 5.1:** Vista superior da placa montada

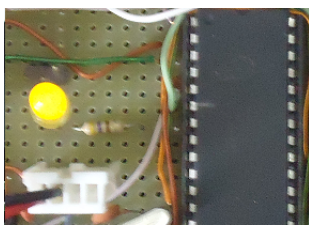
tema foi alimentado com uma tensão de 12V e a tensão de 5V foi obtida com um CI regulador de tensão LM7805.

No intuito de reduzir ruídos provenientes da fonte de alimentação foram utilizados dois capacitores de diferentes tipos. Um capacitor cerâmico de 100nF e outro eletrolítico de 100 $\mu$ F/16V. Ligou-se de forma paralela com a fonte de alimentação e aos componentes da placa.

A ligação entre os componentes foram feitas utilizando fio de cobre. Os componentes externos da placa, como os sensores, *display LCD* e motores, foram ligados a placa utilizando barra de pinos e conectores. Para que o microcontrolador

pudesse ser removido da placa com mais facilidade, caso houvesse necessidade, foi utilizado um soquete de 40 pinos.

Utilizou-se um *led* para indicação de um "modo de segurança", esse modo foi programado como forma de proteção para o driver L298. Para o funcionamento desse modo, os pinos 1 e 15 do driver L298 foram ligados na entrada analógica do microcontrolador. De acordo com a leitura do conversor A/D, os sinais de *PWM* e a entradas do driver são desligados ou não, caso necessário, evitando-se assim a queima desse driver. Nas figuras 5.2(a) e 5.2(b) são apresentadas, o *led* de indicação deste modo e a mensagem informada no *display LCD*, respectivamente.



(a) *Led*



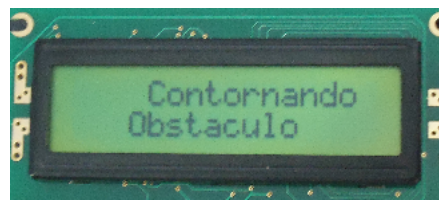
(b) *Display LCD*

**Figura 5.2:** *Led* e *Display LCD* indicando o modo de segurança

O *display LCD* foi utilizado para remeter mensagens de cada estado do robô durante o trajeto. Nas figuras 5.3(a) e 5.3(b) apresentam-se algumas dessas mensagens.



(a) Indicando o contorno



(b) Indicando o sentido

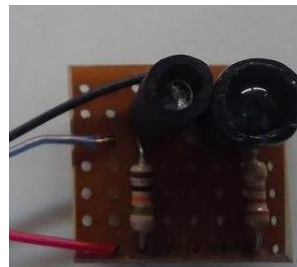
**Figura 5.3:** Mensagens mostradas no *display LCD* pelo robô

## 5.2 Hardware do sensor de infravermelho

O sensor de infravermelho usado nesse projeto, foi montado conforme a seção 4.2.1. Nas figuras 5.4(a) e 5.4(b) apresenta-se o resultado final da montagem do sensor infravermelho.



(a) Visão lateral do sensor



(b) Visão superior do sensor

**Figura 5.4:** Sensor infravermelho montado

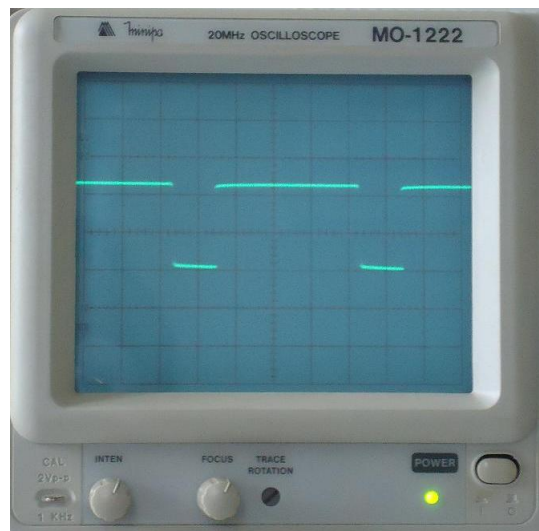
Foram utilizados dois pares desse conjunto: um deles na identificação de mudanças na linha para a esquerda e o outro para mudanças à direita. Os sensores foram ligados nas portas analógicas do microcontrolador e de acordo com a leitura do conversor A/D o programa determina qual deverá ser direção tomada pelo robô. Pela figura 5.4, nota-se que diodo e o fotodiodo foram cobertos no intuito melhorar o desempenho e diminuir a incidência de fontes externas de luz.

## 5.3 Programa de controle do robô

O programa (Anexo A) foi desenvolvido gradativamente de acordo com as necessidades da aplicação.

O controle dos motores foi feito utilizando sinais de *PWM* gerados pelo próprio microcontrolador. Dentro do programa existem quatro funções definidas para operação do *PWM*: `PWM_Init()`, `PWM_Start()`, `PWM_Stop()` e `PWM_Set_Duty()`. A função `PWM_Init()` é responsável por configurar a frequência do *PWM*. Nesse

caso utilizou-se a frequência de 10KHz e a função **PWM\_Set\_Duty()** define o *duty cycle*, que neste caso utilizou-se 75%. Já as funções **PWM\_Start()** e **PWM\_Stop()** iniciam e param respectivamente o *PWM* na sua respectiva porta. Após a implementação dessas funções utilizou-se o osciloscópio para verificar o funcionamento do *PWM*.



**Figura 5.5:** Visualização em osciloscópio da onda de *PWM* gerada

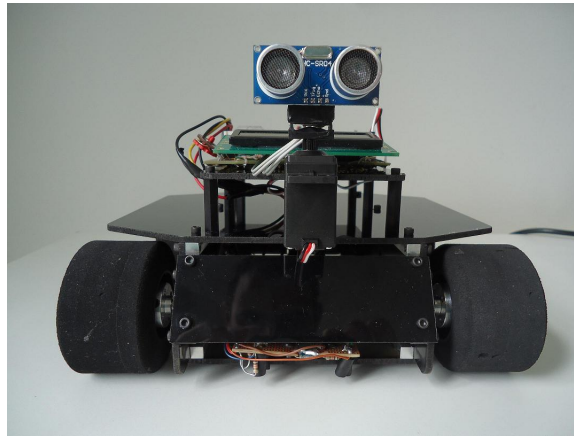
Na figura 5.5 apresenta-se o resultado do teste feito em laboratório utilizando uma carga (resistor  $10K\Omega$ ). O sensor de ultrassom foi acionado por *software* a partir de um pulso de 10 $\mu$ s gerado de acordo com o manual do fabricante e as informações contidas na seção 2.5.2. Para calcular distância de um determinado objeto foi criada uma função **Calcula\_distancia()**. Essa função possui um contador que é incrementado quando o sensor está em nível lógico "1" e zerado quando está nível lógico "0". De acordo com o valor desse contador é feito um cálculo para determinar a distância aproximada em centímetros.

O servomotor foi acionado por *software* utilizando a largura do pulso para determinar o ângulo do eixo do motor. Para tanto foram criadas três funções

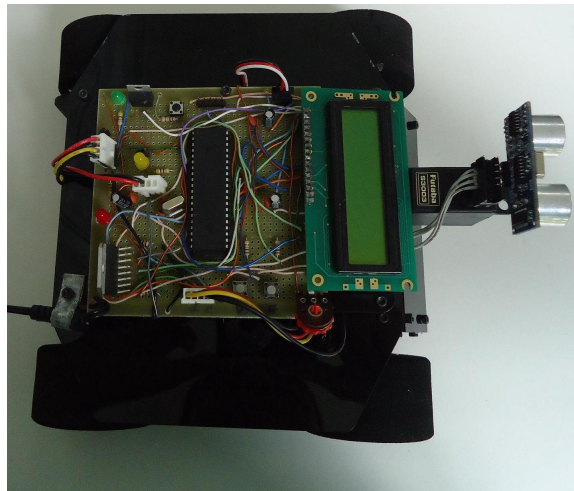
**Servo\_00()**, **Servo\_45()** e **Servo\_90()**, cada uma delas definem o respectivo ângulo do servomotor.

Nas figuras 5.6 e 5.7 apresenta-se o robô montado com sua placa de controle em dois ângulos diferentes, respectivamente frontal e superior.

Esse link: <http://www.youtube.com/watch?v=B6isBHs4i8A> demonstra o funcionamento do robô.



**Figura 5.6:** Vista frontal do robô com sua placa de controle



**Figura 5.7:** Vista superior do robô com sua placa de controle

Na tabela 5.1 é apresentado o custo estimado dos componentes eletrônicos utilizados no projeto, com os valores orçados no dia 20/04/2013.

**Tabela 5.1:** Estimativa de preço dos componentes eletrônicos utilizados

Quantidade	Descrição	Preço unitário	Subtotal
1	microntrolador PIC18F4550	30,35	30,35
1	driver L298	14,75	14,75
1	cristal oscilador 20MHz	0,84	0,84
1	módulo LCD MGD1602B	26,89	26,89
1	soquete de 40 pinos para o PIC18F4550	2,00	2,00
3	capacitores eletrolíticos de 100 $\mu$ F	0,15	0,45
3	capacitores cerâmicos 100nF	0,10	0,30
2	capacitores cerâmicos 10pF	0,10	0,20
3	diodos do tipo LED	0,18	0,54
2	resistores de 1 $\Omega$	0,15	0,30
2	resistores de 220 $\Omega$	0,15	0,30
1	resistor de 1,2K $\Omega$	0,15	0,15
2	resistores 4,7K $\Omega$	0,15	0,30
5	resistores de 10K $\Omega$	0,15	0,75
2	fototransistores 3mm TIL 78	0,43	0,86
2	diodos <i>led</i> infravermelho 5mm TIL 32	0,42	0,84
1	trimpot com resistência 10k $\Omega$	1,50	1,50
3	botões do tipo <i>Push button</i>	0,60	1,80
1	sensor ultrassom HC-SR04	10,99	10,99
1	servomotor futaba modelo S3003	23,00	23,00
-	-	<b>Valor total</b>	117,11



## **6 CONCLUSÃO**

### **6.1 Contribuições do trabalho**

O objetivo desse trabalho foi apresentar um sistema embarcado para controlar um robô móvel autônomo capaz de seguir uma linha no chão e ao encontrar um obstáculo durante o trajeto, desviar desse obstáculo e encontrar novamente a linha continuar o percurso.

O desenvolvimento desse sistema foi possível com a elaboração de um projeto de hardware e a interligação desse hardware a parte mecânica do robô móvel. Além disso, foi necessário à criação de um programa capaz de tornar este sistema autônomo e inteligente o suficiente para seu pleno funcionamento de acordo com as mais diversas condições.

Durante a elaboração do sistema, foram encontradas algumas dificuldades na obtenção de alguns sensores devido à indisponibilidade, bem como na integração de todos os componentes do sistema embarcado para seu correto funcionamento. Ao acrescentar o sensor HC-SR04, houve problemas para incorporá-lo a placa já que algumas portas do microcontrolador estavam sendo utilizadas por outros componentes, o que causou uma série de rearranjos na placa para que o sistema funcionasse.

Apesar das dificuldades encontradas, o sistema embarcado em questão atendeu as expectativas propostas. O robô foi capaz de seguir a linha e desviar dos obstáculos encontrados durante o trajeto. Com algumas adaptações simples esse sistema pode ser utilizado numa vasta gama de aplicações.

### **6.2 Pontos negativos**

Devido a utilização de sensores infravermelho, presença de fontes de luz externa interferem no funcionamento do sistema em questão. O microcontrolador



utilizado possui um número limitado de portas com função a *PWM*, o que tornou o *software* um pouco mais complicado. Ainda, as rodas do robô apresentam uma borracha lisa, o que fez com que o robô deslize algumas vezes durante o percurso.

O robô não apresenta bom desempenho no caso de curvas muito acentuadas, devido algumas limitações do sensor infravermelho e aos atrasos existentes na leitura desses sensores.

### **6.3 Trabalhos futuros**

Como trabalho futuro, podem ser adicionados mais sensores de ultrassom no intuito de melhorar o desempenho do robô no desvio dos obstáculos; a instalação de uma câmera sem fio no robô para transmitir as imagens em tempo real do percurso; a adição de um sensor infravermelho central; Melhorar o *software* desenvolvido; a identificação da linha por cor; a confecção da placa em circuito impresso.

Além das melhorias acima citadas, utilizando o mesmo o *hardware* fazendo apenas alterações a nível de programa podem ser desenvolvidas inúmeras aplicações para os mais diversos fins.

## REFERÊNCIAS BIBLIOGRÁFICAS

BRÄUNL, T. *Mobile Robot Design and Applications with Embedded Systems*. 2. ed. Berlin: Springer, 2006.

CYTRON. *HC-SR04 User's Manual V1.0*. Johor, 2012. Disponível em: <[https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL\\_pfa39RsB-x2qR4vP8saG73rE/edit?pli=1](https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit?pli=1)>.

FAIRCHILD. *LM78XX/LM78XXA-3-Terminal 1A Positive Voltage Regulator*. [S.l.], 2012. Disponível em: <<http://www.fairchildsemi.com/ds/LM/LM7805.pdf>>.

JUNG, C. R.; OSÓRIO, F. S.; KELBER, C.; HEINEN, F. Computação embarcada: projeto e implementação de veículos autônomos inteligentes. *Anais do CSBC'05-XXIV Jornada de Atualização em Informática (JAI)*, v. 1, n. 1, p. 1358–1406, 2005. Disponível em: <[http://www.unisinos.br/\\_diversos/congresso-sbc2005/\\_dados/anais/pdf/arc0288.pdf](http://www.unisinos.br/_diversos/congresso-sbc2005/_dados/anais/pdf/arc0288.pdf)>.

KAMAL, R. *Embedded Systems – Architecture, Programming and Designing*. 2. ed. New Delhi: Mcgraw Hill, 2008.

LAUREANO, M. *Programando em C para Linux, Unix e Windows*. 1. ed. Rio de janeiro: Brasport, 2005.

LOVINE, J. *Pic Robotics - A Beginners Guide To Robotics Projects Using The Pic Microcontroller*. 1. ed. New York: Mcgraw Hil, 2004.

MARWEDEL, P. *Embedded Systems Design*. 1. ed. Dordrecht: Springer, 2003.

MICROCHIP. *Datasheet PIC18F2455/2550/4455/4550*. Califórnia, 2009. Disponível em: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>>.

MIKROELEKTRONIKA. *MikroC: User's manual*. Califórnia, 2006. Disponível em: <[http://www.mikroe.com/pdf/mikroc/mikroc\\_manual.pdf](http://www.mikroe.com/pdf/mikroc/mikroc_manual.pdf)>.

NIKU, S. B. *Introduction to Robotics – Analysis, Systems, Applications*. 1. ed. New Jersey: Prentice Hall, 2001.

REIS, C. *Sistemas Operacionais para Sistemas Embarcados*". [S.l.]: EDUFBA, 2004.

SIQUEIRA, F. T.; MENEGOTTO, C. C.; WEBER, T. S.; NETTO, J. C.; WAGNER, F. R. Desenvolvimento de sistemas embarcados para aplicações críticas. *Instituto de informática - UFRGS*, 2006. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/errc/2006/005.pdf>>.

SOUZA, D. J. D. *Desbravando o PIC : ampliado e atualizado para PIC16F628A*. 10. ed. São Paulo: Érica, 2006.

STMICROELECTRONICS. *MikroC: User's manual*. [S.l.], 2000. Disponível em: <[http://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](http://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)>.

WILMSHURST, T. *Designing embedded systems with PIC microcontrollers: principles and applications*. 1. ed. Londres: Elsevier, 2007.

WOLF, D. F.; OSÓRIO, F. S.; SIMÕES, E.; JR., O. T. *Robótica Inteligente: Da Simulação às Aplicações no Mundo Real*. Rio de Janeiro: Editora da PUC Rio, 2009.

## A ANEXOS

### A.1 Anexo A - Controle\_robo.c

```

//Autor:Luiz Fernando          matricula:200910296
// Programa de controle do robo

//Configuracao dos bits do display LCD
sbit LCD_RS at RD6_bit;
sbit LCD_EN at RD5_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RC7_bit;
sbit LCD_D6 at RC6_bit;
sbit LCD_D7 at RD7_bit;

sbit LCD_RS_Direction at TRISD6_bit;
sbit LCD_EN_Direction at TRISD5_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISC7_bit;
sbit LCD_D6_Direction at TRISC6_bit;
sbit LCD_D7_Direction at TRISD7_bit;

void Mostra_Display(char texto1[16],char texto2[16]){
    Lcd_Init();                //Inicia o Display
    Lcd_Cmd(_LCD_CLEAR);      //limpa o Display
    Lcd_Cmd(_LCD_CURSOR_OFF); //Desabilita o cursor
    Lcd_Out(1,5,texto1);
    Lcd_Out(2,4,texto2);
}

//Funcao para Inverter o sentido de rotacao dos motores
void Reverso(){
    Mostra_Display("Sentido","Inverso");
    if(PORTB.RB0 == 0 && PORTB.RB1 == 1){
        PORTB.RB0 = 1;
    }
}

```

```

        PORTB.RB1 = 0;
    }else{
        PORTB.RB0 = 0;
        PORTB.RB1 = 1;
    }
    if(PORTB.RB2 == 0 && PORTB.RB3 == 1){
        PORTB.RB2 = 1;
        PORTB.RB3 = 0;
    }else{
        PORTB.RB2 = 0;
        PORTB.RB3 = 1;
    }
    delay_us(300);
}
//Funcao para girar o robo para a direita
void Direita(){
    Mostra_Display("Sentido","Direita");
    PORTB.RB0 = 1;
    PORTB.RB1 = 0;
    PORTB.RB2 = 0;
    PORTB.RB3 = 1;
    delay_us(510);
}
//Funcao para covergir para a direita
void Direita_desvio(){
    Mostra_Display("Desvio a","Direita");
    PORTB.RB0 = 1;
    PORTB.RB1 = 0;
    PORTB.RB2 = 0;
    PORTB.RB3 = 1;
    delay_ms(2000);
}
//Funcao para covergir para a esquerda
void Esquerda_desvio(){
    Mostra_Display("Desvio a","Esquerda");

```

```

        PORTB.RB0 = 0;
        PORTB.RB1 = 1;
        PORTB.RB2 = 1;
        PORTB.RB3 = 0;
        delay_ms(1000);
    }
    //Funcao para girar o robo para a esquerda
    void Esquerda(){
        Mostra_Display("Sentido", "Esquerda");
        PORTB.RB0 = 0;
        PORTB.RB1 = 1;
        PORTB.RB2 = 1;
        PORTB.RB3 = 0;
        delay_us(500);
    }
    //Funcao para desativar o PWM e o L298
    void Modo_seguranca(){
        Mostra_Display("Modo", "Seguranca");
        PORTA.RA5 = 1;
        PWM1_Stop();
        PWM2_Stop();
        PORTB.RB0 = 0;
        PORTB.RB1 = 0;
        PORTB.RB2 = 0;
        PORTB.RB3 = 0;
        delay_ms(250);
    }
    //Funcao para deslocar o robo para frente
    void Segue_linha(){
        Mostra_Display("Seguindo", "Linha");
        PORTB.RB0 = 0;
        PORTB.RB1 = 1;
        PORTB.RB2 = 0;
        PORTB.RB3 = 1;
        delay_us(350);
    }

```

```

}
//Funcao para seguir ao lado do objeto encontrado pelo sensor
void Segue_reto(){
    Mostra_Display("Contornando", "Obstaculo");
    PORTB.RB0 = 0;
    PORTB.RB1 = 1;
    PORTB.RB2 = 0;
    PORTB.RB3 = 1;
    delay_ms(200);
}
//Funcao que indica o fim do percurso
void Fim_linha(){
    Mostra_Display("Fim de", "Linha");
    PORTB.RB0 = 0;
    PORTB.RB1 = 0;
    PORTB.RB2 = 0;
    PORTB.RB3 = 0;
    delay_ms(200);
}
//Funcao que calcula a distancia do objeto
int Calcula_distancia(){
int cont = 0;
    PORTC.RC0 = 1;
    delay_us(10); //Pulso 10uS para acionamento do ultrassom
    PORTC.RC0 = 0;
    while(PORTA.RA4 == 0){
        delay_us(10);
        cont=0;
    }
    while(PORTA.RA4 == 1){
        delay_us(10);
        cont++;
    }
    return(cont / 5.2); //Calculo da distancia em centimetros
}

```

```

//Funcao que rotaciona o servomotor para 0 graus
void Servo_0(){
    int i=0;
    for(i=0;i<=20;i++){
        PORTE.RE0 = 1;
        delay_us(600);
        PORTE.RE0 = 0;
        delay_us(19400);
    }
}

//Funcao que rotaciona o servomotor para 90 graus
void Servo_90(){
    int i=0;
    for(i=0;i<=20;i++){
        PORTE.RE0 = 1;
        delay_us(1500);
        PORTE.RE0 = 0;
        delay_us(18500);
    }
}

//-----Variaveis-----
int tensao_a,tensao_b,infravermelho_1,infravermelho_2,limite;
//-----

void main(){
    ADCON1 = 0b00001011;//Configura os pinos de AN0 ate AN3 como analogicoes
    TRISA = 0;
    TRISA.RA0 = 1; //Define os pinos RA0,RA1,RA2 e RA3 como entrada
    TRISA.RA1 = 1;
    TRISA.RA2 = 1;
    TRISA.RA3 = 1;
    TRISA.RA4 = 1;
    PORTA = 0;
    TRISB = 1;
    TRISE.RE0 = 0;
    PORTE = 0;
}

```



```

TRISB.RB3 = 0; //Define os pinos RB0, RB1, RB2 e RB3 como Saida
TRISB.RB2 = 0;
TRISB.RB1 = 0;
TRISB.RB0 = 0;
TRISB.RB5 = 1;
TRISB.RB4 = 1;
PORTB = 0;
PORTC = 0;
TRISC = 0;
PWM1_Init(10000);
PWM2_Init(10000);
PWM1_Start();
PWM2_Start();
PWM1_Set_Duty(192); //Define o ciclo como 75%
PWM2_Set_Duty(192);
Servo_0(); //Posiciona o servomotor na posicao inicial
while(1){
    infravermelho_1 = adc_read(0); //Faz a leitura do conversor A/D no canal 0
    infravermelho_2 = adc_read(1);
    tensao_a = adc_read(2); //Verifica a tensao no L298
    tensao_b = adc_read(3);
    //Define os limites alcancados pelo sensor
    if(calcula_distancia() > 2 && calcula_distancia() < 400){
        limite = 1;}
    else{
        limite = 0;}
    delay_ms(50);
    if(calcula_distancia() <= 5 && limite == 1){
        Reverso();
        Servo_90();
        Direita_desvio();
        delay_ms(250);
        while(calcula_distancia() < 25){
            Segue_reto();
            delay_ms(30);

```

```
    }  
  
    Servo_90 ();  
    Segue_reto ();  
    Esquerda_desvio ();  
    Esquerda_desvio ();  
    Segue_reto ();  
    Segue_reto ();  
    Segue_reto ();  
    delay_ms (30);  
    while (calcula_distancia () < 25) {  
        Segue_reto ();  
        delay_ms (30);  
        if (calcula_distancia () > 20) {  
            PORTB.RB0 = 0;  
            PORTB.RB1 = 1;  
            PORTB.RB2 = 1;  
            PORTB.RB3 = 0;  
            delay_ms (100);  
        }  
        if (calcula_distancia () < 11) {  
            PORTB.RB0 = 1;  
            PORTB.RB1 = 0;  
            PORTB.RB2 = 0;  
            PORTB.RB3 = 1;  
            delay_ms (100);  
        }  
    }  
    Segue_reto ();  
    Segue_reto ();  
    Esquerda_desvio ();  
    infravermelho_1 = adc_read (0);  
    infravermelho_2 = adc_read (1);  
    delay_ms (10);  
    while (infravermelho_1 < 100 || infravermelho_2 < 100) {  
        Mostra_Display ("Procurando", "Linha");  
    }
```

```

        PORTB.RB0 = 0;
        PORTB.RB1 = 1;
        PORTB.RB2 = 0;
        PORTB.RB3 = 1;
        delay_us(55);
        infravermelho_1 = adc_read(0);
        infravermelho_2 = adc_read(1);
    }
    Mostra_Display("Linha", "Encontrada");
    PORTB.RB0 = 0;
    PORTB.RB1 = 1;
    PORTB.RB2 = 0;
    PORTB.RB3 = 1;
    delay_ms(100);
    PORTB.RB0 = 1;
    PORTB.RB1 = 0;
    PORTB.RB2 = 0;
    PORTB.RB3 = 1;
    delay_ms(1500);
    Segue_linha();
    Servo_0();
} else {
    if(tensao_a >= 150 || tensao_b >= 150){Modo_seguranca();}
    if(infravermelho_1 < 150 && infravermelho_2 < 150){ Segue_linha();}
    } else if(infravermelho_1 > 150 && infravermelho_2 < 150){ Direita();}
    } else if(infravermelho_1 < 150 && infravermelho_2 > 150){ Esquerda(); }
    }
}
delay_ms(3);
}

```

## A.2 Anexo B- Esquemático da placa de controle do robô

