

Centro Universitário Positivo - UnicenP
Núcleo de Ciências Exatas e Tecnológicas – NCET
Engenharia da Computação
Luiz Antonio Maldonado Marthos

Sistema de Mapeamento em Ambiente Fechado

Curitiba
2005

Centro Universitário Positivo - UnicenP
Núcleo de Ciências Exatas e Tecnológicas – NCET
Engenharia da Computação
Luiz Antonio Maldonado Marthos

Sistema de Mapeamento em Ambiente Fechado

Monografia apresentada à disciplina de Projeto Final, como requisito parcial à conclusão do Curso de Engenharia da Computação. Orientador: Prof. Roberto Selow.

Curitiba
2005

TERMO DE APROVAÇÃO

Luiz Antonio Maldonado Marthos

Sistema de Mapeamento em Ambiente Fechado

Monografia aprovada como requisito parcial à conclusão do curso de Engenharia da Computação do Centro Universitário Positivo, pela seguinte banca examinadora:

Prof. Alessandro Zimmer

Prof. Luiz Carlos Pessoa Albini

Prof. Roberto Selow

Curitiba, 01 de janeiro de 2006.

AGRADECIMENTOS

Agradeço a todos os que me apoiaram e que estiveram do meu lado todo o tempo.

SUMÁRIO

1	INTRODUÇÃO	1
2	OBJETIVOS	3
3	FUNDAMENTAÇÃO TEÓRICA.....	4
3.1	Mapeamento robótico	4
3.2	Monitoramento de tensões e correntes	5
3.2.1	Medição de tensão.....	5
3.2.2	Sensores de corrente	8
4	ESPECIFICAÇÃO	10
4.1	O robô utilizado	11
4.2	Módulos desenvolvidos.....	11
4.3	Especificação do Hardware.....	12
4.3.1	Funções do Hardware	12
4.3.2	Componentes utilizados	13
4.3.3	Diagrama em bloco com descrição detalhada	14
4.4	Especificação do Software	15
5	HARDWARE	19
5.1	Esquemáticos.....	19
5.2	Sinais de interface.....	20
5.2.1	Primeira Folha dos esquemáticos	20
5.2.2	Segunda Folha dos Esquemáticos	21
5.2.3	Terceira Folha dos Esquemáticos	22
5.2.4	Quarta Folha de Esquemáticos	22
5.2.5	Quinta Folha de Esquemáticos	23
5.3	Lista de materiais	23
6	SOFTWARE	25
6.1	Software	25
6.1.1	Diagrama de casos de uso	29
6.1.2	Diagrama de Classes.....	33
6.2	Dados	34
6.2.1	Modelo Relacional	34
6.2.2	Dicionário de Dados	35
6.3	<i>Firmware</i>	35
6.3.1	Diagrama de estados.....	35
6.3.2	Fluxograma.....	36
7	PROCEDIMENTOS DE TESTES DO PROJETO	38

8	CRONOGRAMA DE DESENVOLVIMENTO	39
9	ESTUDO DE VIABILIDADE TÉCNICO-ECONÔMICA	40
10	RESULTADOS.....	41
10.1	Hardware.....	41
10.2	Software	41
10.3	Resultados da validação	42
11	CONCLUSÕES	44
12	REFERÊNCIAS BIBLIOGRÁFICAS.....	45
13	BIBLIOGRAFIA.....	46
14	ANEXO	47
	ANEXO 01 – ESQUEMÁTICO 01	48
	ANEXO 02 – ESQUEMÁTICO 02	50
	ANEXO 03 – ESQUEMÁTICO 03	52
	ANEXO 04 – ESQUEMÁTICO 04	54
	ANEXO 05 – ESQUEMÁTICO 05	56
	ANEXO 06 – CRONOGRAMA.....	58

LISTA DE FIGURAS

Figura 1 - Conversor Analógico Digital	6
Figura 2 - Exemplo de Conversão.....	7
Figura 3 - Visão geral do sistema	10
Figura 4 - Diagrama em blocos do Hardware.....	14
Figura 5 - Tela principal do Software.....	25
Figura 6 - Tela do modo do robô.....	26
Figura 7 - Tela do modo Controle	27
Figura 8 - Tela do modo Mapeamento	28
Figura 9 - Diagrama de casos de uso	29
Figura 10 - Diagrama de Seqüência para o Caso de Uso Controlar Robô.....	30
Figura 11 - Diagrama de Seqüência para o Caso de Uso Começar a Mapear	31
Figura 12 - Diagrama de Seqüência para o Caso de Uso Fazer Mapeamento	31
Figura 13 - Diagrama de Seqüência para o Caso de Uso Setar Tempos do Histórico	32
Figura 14 - Diagrama de Classes	33
Figura 15 - Modelo Relacional.....	34
Figura 16 - Diagrama de estados	36
Figura 17 – Fluxograma.....	37
Figura 18 - Mapa resultante da validação.....	43

LISTA DE TABELAS

Tabela 1 - Exemplo de valores de uma conversão	7
Tabela 2 - Sinais da primeira folha de esquemáticos.....	21
Tabela 3 - Sinais da segunda folha de esquemáticos	21
Tabela 4 - Sinais da terceira folha de esquemáticos	22
Tabela 5 - Sinais da quarta folha de esquemáticos	22
Tabela 6 - Sinais da quinta folha de esquemáticos	23
Tabela 7 - Lista de Componentes	24
Tabela 8 - Dicionário de dados - Tabela Medidas	35
Tabela 10 - Estimativa de custo	40

LISTA DE SIGLAS

Placa de rede *Wireless* – Placa de rede com transmissão de dados sem fio.

Software – Programa de computador.

Hardware – Conjunto de componentes elétricos sozinhos ou em conjunto com componentes eletrônicos.

Firmware – Programa que é executado pelo hardware.

Bitmap – Um formato de imagem padrão, que pode ser visualizado em qualquer computador.

LISTA DE SÍMBOLOS

U - Diferença de Potencial.

R - Resistência.

I - Corrente.

A - Unidade de corrente (Ampère).

V - Unidade de tensão (Volts).

Ω - Unidade de resistência elétrica (Ohm).

m - Metros.

mm² - Milímetros quadrados.

°C - Graus centígrados.

Bps – bits per second.

RESUMO

Este trabalho vem ao encontro da necessidade de mapeamento de ambientes onde um ser humano não possa entrar, propondo um algoritmo, que utilizando um robô, possa mapear este ambiente. Ambiente este que deverá ser fechado, ou seja, com limites específicos. O algoritmo que faz o reconhecimento do ambiente, utiliza valores de 6 sensores de distância, que controlarão a distância que o robô está de objetos fixos e móveis no ambiente, e de 2 fotorefletores, que controlam a odometria.

Também foi implementado um módulo, chamado de módulo de utilidades, que é responsável por monitorar as tensões nas baterias, o consumo de corrente do computador embarcado no robô, além das tensões de alimentação do mesmo, e a corrente dos dois motores.

Há um terceiro módulo em que o robô poderá andar pelo ambiente sendo controlado à distância.

ABSTRACT

This Project comes to help the mapping necessity in environments that humans can not enter, it proposes an algorithm, that using a robot, make a mapping in an environment. This environment have to be closed, or either, with specific limits. The algorithm that make the recognition of the environment, uses values of 6 distance sensors, that control the distance that the robot is from mobile and fixed objects, and from 2 photo reflectors, that control the odometry.

Also a module, called module of utilities, have been implemented, that is responsible for monitoring the batteries voltage, the current consumption of the embarked computer on the robot, beyond the feeding voltage, and the current of the two motors.

There is a third module where the robot can ride by the environment being long-distance controlled.

1 INTRODUÇÃO

A relação entre o homem e o ambiente no qual ele vive é muito extensa e complexa. A criação de máquinas veio a resolver muitos problemas e tornar esta relação mais fácil. Porém muitas situações ainda oferecem riscos, e precisam de uma solução. Tem-se como exemplo os de ambientes impróprios ao homem devido ao risco de contaminação, altas temperaturas, alto nível de periculosidade, etc.

Este trabalho tem como meta a implementação de um sistema de mapeamento e navegação em um robô móvel, possibilitando a este autonomia para atuar em ambientes impróprios para o homem. Este robô navega por um ambiente fechado, semi-estruturado, reconhecendo os detalhes e desviando dos obstáculos. O reconhecimento é feito através da leitura de sensores de distância e da odometria, o que permite a criação de um mapa do ambiente. Um ambiente é chamado de semi-estruturado, segundo Freire^[1], quando o interior deste é plano, sem nenhuma concavidade, mais que admita a presença de objetos móveis e pessoas em seu interior, e cuja configuração das paredes e objetos é desconhecida.

Outro modo que o robô pode operar é sendo controlado à distância. Com esse controle, pode-se verificar algo que está em alguma parte específica do ambiente, ou até mesmo verificar se algum objeto fixo está no lugar que deveria estar. Além disso, se o robô já tiver passado por esse ambiente e feito o mapeamento dele, pode-se ter em uma imagem as informações desse ambiente para comparar com informações de um novo mapeamento.

O objetivo deste projeto é implementar um módulo de mapeamento e navegação no robô móvel que seja capaz de determinar os objetos fixos em um ambiente.

Este trabalho está organizado da seguinte forma:

No capítulo dois são apontados os objetivos deste projeto.

No capítulo três é mostrada a teoria e é tratado do mapeamento robótico. Em seguida como é feita a conversão dos valores de tensão das duas baterias, de

alimentação do computador a corrente que este e os motores consomem, e como foi implementado o sensor de corrente.

No capítulo quatro, é exposto como é o robô, e como ele é utilizado no projeto, os módulos desenvolvidos com o detalhamento de hardware e software.

No capítulo cinco, são mostrados os esquemáticos de *hardware*, e explicado o seu funcionamento.

No capítulo seis, são descritos o software, os seus casos de uso, diagrama de classe a implementação do banco de dados, e o detalhamento do firmware.

No capítulo sete, é explicado como foram realizados os testes para a validação da pesquisa apresentada neste trabalho.

No capítulo oito, é mostrado o cronograma de desenvolvimento.

No capítulo nove, é feita uma estimativa de custo do projeto.

No capítulo dez são descritos os resultados obtidos, tanto para o software como para o hardware.

No capítulo onze é apresentada uma conclusão sobre o projeto.

No capítulo doze, estão as referências citadas no corpo do texto.

E, finalmente, no décimo terceiro capítulo, são acrescentados os anexos referenciados no texto.

2 OBJETIVOS

Este trabalho tem dois objetivos específicos a serem atingidos, o primeiro é a construção de um módulo de mapeamento do ambiente, que também será responsável em evitar colisões com obstáculos que se apresentem na frente do robô móvel. O segundo módulo monitora as variáveis indispensáveis ao funcionamento do robô móvel, que são: tensões das duas baterias, correntes consumidas pelos dois motores, e pela fonte de alimentação da placa mãe do computador do robô e as diferentes tensões usadas pela placa mãe.

3 FUNDAMENTAÇÃO TEÓRICA

Nesse tópico, será apresentada a teoria imprescindível para a realização do projeto. Primeiramente é apresentada a fundamentação teórica necessária para a implementação do módulo de mapeamento e em seguida a fundamentação para a implementação do módulo monitor das variáveis do robô.

3.1 Mapeamento robótico

Com o grande crescimento do uso da robótica, vê-se o aumento do uso do mapeamento robótico. Segundo Sebastian^[2], o mapeamento robótico deriva do problema de adquirir modelos de ambientes físicos através de robôs móveis.

A maneira como fazer o mapeamento é um dos maiores problemas quando este modelo deve ser preciso e confiável, pois podem acontecer erros de diversos tipos, como por exemplo, erros nas medições dos sensores, derrapagens das rodas, ou leitura errônea da odometria, que implicarão na construção imprecisa do mapa de um determinado trecho.

Sebastian^[2] chama os ambientes que são mapeados de estáticos, estruturados, e de tamanho limitado. Estes já possuem métodos confiáveis para mapeamento. Há também ambientes não estruturados, dinâmicos e de grande escala, os quais necessitam de uma grande pesquisa para que se possa fazer o mapeamento.

Para se construir um mapa, um robô deve conhecer sua localização, o que está a sua volta, e para onde ir. Para saber o que há em volta, é comum usar sensores infravermelhos, câmeras de vídeo, sonares, lasers, radares, sensores de impacto, sensores de bússola e GPS. Mas, estes sensores podem fornecer medidas com erros, ou as medidas repletas de ruído. Outra informação importante para o mapeamento, é o controle de movimentos do robô durante o mapeamento, por exemplo, para onde ele está indo, e em que direção virou, ou se continuou em linha reta.

3.2 Monitoramento de tensões e correntes

A fundamentação teórica para o desenvolvimento do módulo de monitoramento de tensões e correntes foi estudada separadamente. Primeiramente foi visto a fundamentação necessária às medições de tensões e em seguida a fundamentação necessária às medições de corrente.

3.2.1 Medição de tensão

Uma forma simples de se medir tensão é utilizando conversores analógico digitais. Tensões nada mais são do que diferenças de potenciais elétricos, de diferentes origens, referenciados a um mesmo terra.

Para se utilizar valores analógicos, deve-se convertê-los para valores digitais, para isso pode-se utilizar um conversor analógico digital. Sua função é de converter grandezas analógicas em seqüências de dados binários, '0's e '1's.

A forma mais simples de se fazer essa conversão esta demonstrada na figura 1. Uma tensão de referência é colocada no circuito, e essa tensão passa por vários resistores, na configuração de divisores de tensão, ocorre uma queda de tensão em cada resistor, gerando outras tensões de referência em cada saída desses resistores. Essas tensões de referência servem de entrada de comparadores de tensão, a outra entrada é o próprio sinal que se deseja converter. Após passar pelos comparadores, os valores de saída deles são ligados à entrada de portas XOR. A saída desses blocos é ligada a um barramento de três bits, passando, quando necessário, por uma porta NOT. Esse barramento é ligado com resistores ao terra.

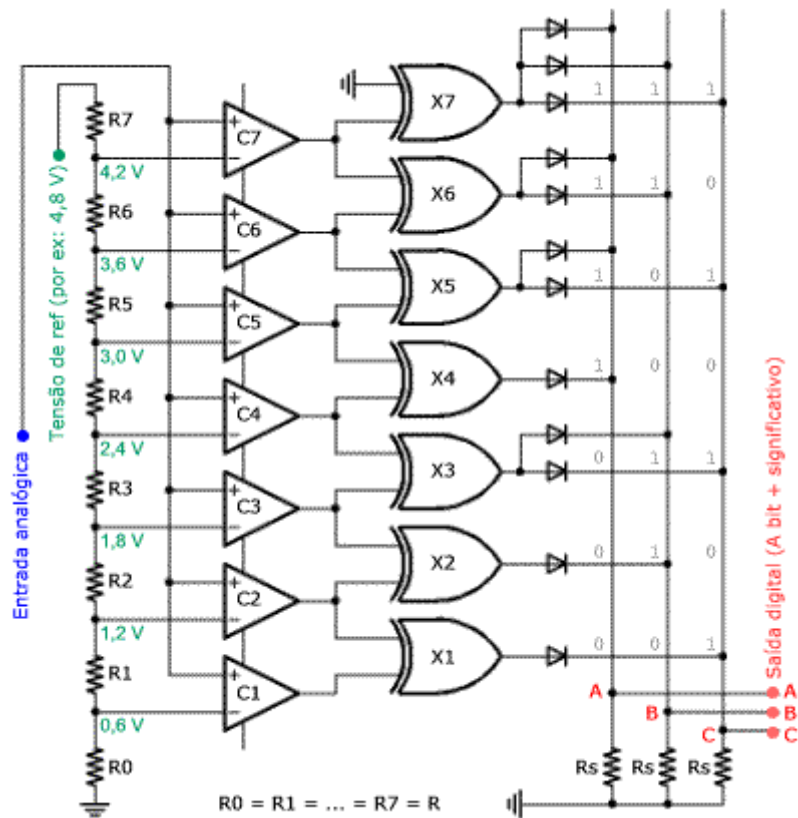


Figura 1 - Conversor Analógico Digital

Essa conversão também pode ser feita por um circuito integrado. Normalmente esses circuitos integrados têm um sinal de entrada, que varia de 0V a 5V, e a configuração de saída é de oito bits, ou seja, o sinal tem uma saída com oito valores binários, ou 0 a 255 em valores decimais. Existem conversores com maior número de bits na saída, por exemplo, 16 bits. Para se calcular exatamente a resolução do conversor, usamos a seguinte fórmula:

$$\text{Resolução} = \frac{\text{Valor máximo que se queira converter}}{2^{\text{Número de bits da saída}}}$$

No caso de um ADC0804, seria $5/2^8$, sendo o cinco a tensão máxima em volts e oito o número de bits do AD, o que resulta em 19,5mV. Isto significa que a cada variação de 19,5mV na entrada, ocorre uma variação de um bit na saída.

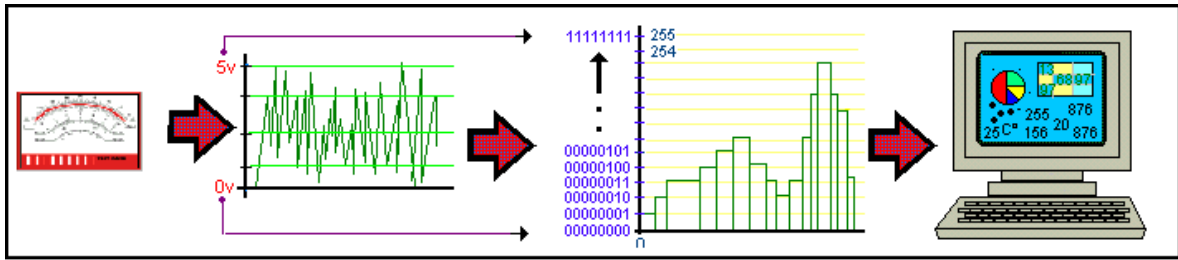


Figura 2 - Exemplo de Conversão

Existem conversores que podem tratar mais de uma entrada, mais não ao mesmo tempo, como por exemplo, o ADC0808. Nele existe um barramento de endereços, onde se pode selecionar uma das oito entradas que vai ser convertida. Podendo fazer a conversão de vários sinais, com a utilização de um único conversor.

Na tabela 1 se tem um exemplo, onde é demonstrado o valor de saída do circuito em relação ao valor de entrada.

Volts	Decimal	Binário
0,0000	0	00000000
0,0195	1	00000001
0,0390	2	00000010
0,0585	3	00000011
0,0780	4	00000100
.	.	.
.	.	.
.	.	.
4,9920	255	11111111

Tabela 1 - Exemplo de valores de uma conversão

Para que os valores de tensão fossem medidos pelo módulo de medição, deve-se deixá-los entre 0 volts e a referência do conversor analógico-digital, no caso 5 volts, para que ele possa convertê-los. Dessa maneira, valores que passam desse limite, tanto para valores positivos ou negativos, devem ser referenciados dentro desse limite. Essa referência é feita através de divisores de tensão, o que deixa com que as tensões positivas fique entre 0 e 5 volts, e as negativas entre -5 e 0 volts. Após esta divisão, essas tensões passam por amplificadores operacionais, na configuração de buffer, para as tensões positivas, e buffer inversor, para as tensões negativas. Além de inverter as tensões negativas, a configuração de buffer nos amplificadores operacionais também é usada para que não seja consumida corrente do circuito do qual estamos medindo essas tensões, para que não haja queda de tensão, pois os amplificadores

operacionais possuem uma impedância de entrada muito alta, o que faz com que eles não consumam corrente da fonte do sinal. A corrente consumida pelo conversor analógico-digital é fornecida pela alimentação do amplificador operacional.

Isto faz com que todas as tensões que são medidas fiquem no limite de 0 e 5 volts, podendo então servir de entrada para o conversor analógico-digital.

3.2.2 Sensores de corrente

Os sensores de corrente usados no projeto foram construídos, de acordo com a Lei de resistividade elétrica de um fio, que nos fala que a resistência de um fio depende do comprimento l , da área da seção transversal S , e da resistividade do material que o fio é produzido ρ (rô). Segundo a seguinte fórmula:

$$R = \rho * (l/S)$$

Portanto, sabendo-se a resistência de um fio, e sendo essa muito pequena, para não interferir na corrente que queremos medir, pode-se chegar a uma tensão equivalente a esta corrente.

Haverá uma queda de tensão sobre esse fio, que segue a equivalência da Lei de Ohm, cuja fórmula é esta: $U = R * I$, sendo U a tensão, R a resistência, e I a corrente.

Tendo então essa tensão, e sabendo o valor da resistência desse fio, podemos chegar no valor de corrente que está passando pelo fio no momento da medição.

No lugar do fio, foi utilizado um fusível de 20A, cuja resistência é muito baixa. Como a tensão que é medida no fusível também é baixa, e não há referência ao terra de nenhuma das duas pontas desse fusível, foi usado um amplificador de instrumentação.

Um amplificador de instrumentação pode ser usado para se referenciar um valor de tensão ao terra em que se está trabalhando, no caso, as entradas do amplificador de instrumentação foram ligadas cada uma a uma ponta do fusível. Como a tensão é muito baixa, um ganho foi dado para essas tensões. Após isto, foi somado um valor de tensão, para que esta não fosse negativa, e então convertida pelo conversor analógico-digital.

4 ESPECIFICAÇÃO

Aqui serão apresentadas uma visão geral do sistema e uma breve descrição dos módulos que compõe o projeto.

Este trabalho utiliza um robô já implementado, que possui um bloco de sensores de distância e um bloco de odometria. O trabalho acrescentou o bloco de mapeamento e o bloco de monitoramento de correntes e tensões. O primeiro bloco utilizará os sensores de medição de distância e a odometria para construir o mapa. O bloco de medição de tensões e correntes fará a medição direta da tensão e utilizará sensores de corrente (também chamados de resistor de shunt).

Em diversas etapas será necessária a utilização de um conversor AD embutido no microcontrolador usado no projeto, para que ele possa ler as grandezas.

Na Figura 03 temos uma representação simples do robô, já com os módulos englobados pelo projeto.

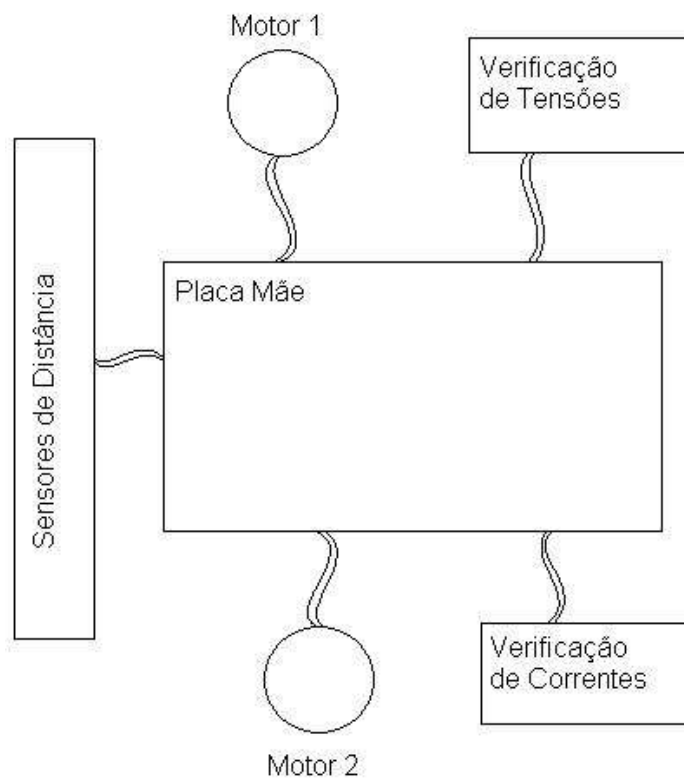


Figura 3 - Visão geral do sistema

4.1 O robô utilizado

O robô que foi utilizado neste projeto é composto por uma base de madeira, dois motores de corrente contínua, e uma terceira roda para apoio. Preso nas rodas dos motores de corrente contínua, há um disco rajado, que será utilizado por fotorelectores para determinar o tanto que a roda está girando. Em cima dessa base ficam duas baterias automotivas. Acima das baterias a outro patamar, que abriga a fonte para o computador embarcado, além de outros circuitos próprios do robô. Acima deste, há um terceiro patamar, que abriga o computador, e seis sensores de distância.

O controle deste robô é feito pelo computador embarcado. Os sinais da odometria, e dos sensores de distância passam por um conversor analógico-digital, e em seguida por um microcontrolador, e são enviados pela serial para esse computador. O computador possui também uma placa de rede *wireless*, e é por ela que se comunica com um computador remoto. A placa utilizada é fabricada pela LG, e seu modelo é LW2110P. A mesma utiliza o protocolo IEEE 802.11b.

A alimentação do robô é feita a partir das duas baterias automotivas (12V), onde uma é destinada a alimentar somente os motores, e outra à alimentação da placa mãe e demais circuitos.

4.2 Módulos desenvolvidos

O bloco de monitoramento de correntes e tensões serve para que o usuário possa saber, a qualquer momento, quanto o robô está consumindo de corrente, qual é o nível de tensão de cada bateria, e se as tensões da placa mãe estão de acordo com o esperado.

O bloco de mapeamento serve para que o usuário obtenha um mapa de determinado local, representado por uma imagem, sem ter que estar dentro desse local para isso.

Os dados para que seja feito o mapeamento são adquiridos de forma automática, conforme o robô for andando pelo ambiente, os sensores informam a que distância que os objetos estão do robô.

Os dados desses dois blocos são coletados pela placa mãe do robô, através de conversores analógico-digital e microcontroladores, e enviados para o computador remoto através de uma placa de rede *wireless*, pelo software do computador embarcado.

O software no computador remoto é responsável pela recepção das mensagens enviadas pelo computador embarcado no robô, por mostrar na tela as informações de tensões e correntes em tempo real, e pelo mapeamento do ambiente, que consiste em impedir a colisão do robô com qualquer objeto, e desenhar o mapa para o usuário.

Ele também armazena as informações desse mapeamento uma imagem, a qualquer momento que o usuário quiser, ou ao ser fechado, o que possibilita o acesso às informações desse mapeamento posteriormente.

Além disso, há a possibilidade de criar um histórico das tensões e correntes, em um tempo pré-definido pelo usuário, onde esses valores são gravados em um banco de dados que pode ser acessado posteriormente.

4.3 Especificação do Hardware

4.3.1 Funções do Hardware

O hardware tem como função fazer a aquisição dos sinais de distância, através do bloco de sensores de distância, que utiliza sensores infravermelhos da Sharp, modelo GP2Y0A02YK. Além de mensurar os sinais de tensão das duas baterias, e de alimentação do computador embarcado no robô, além das correntes consumidas pelos dois motores, pelo computador e demais circuitos. Fazer o tratamento desses sinais,

para poder convertê-los e mandá-los ao computador remoto através do software do computador embarcado.

Esses sinais serão adquiridos por um conversor analógico-digital, sendo tratados por um microcontrolador que transmite esses sinais para o computador embarcado no robô. Este faz a transmissão desses sinais para o computador remoto, que trata esses dados, grava os necessários, e realiza o mapeamento.

A aquisição da corrente é feita da seguinte maneira, os fios de alimentação do computador e dos motores são colocado em série, cada um, com um fusível. Esse fusível serve como um resistor, de resistência baixíssima, o que não influencia na corrente nem na tensão que passa pelo o circuito. Em cima desse fusível há uma queda de tensão, que será equivalente à corrente que o circuito está consumindo. Esse fusível será chamado de sensor de corrente, ou resistor de *shunt*.

Todos os sinais de tensão e corrente serão tratados por amplificadores operacionais, para que possam ser digitalizados por um conversor analógico-digital embutido no microcontrolador. Após a digitalização, os valores em binário de saída do conversor analógico digital são transmitidos via serial para o computador embarcado no robô, e o software que está rodando nele envia estes dados para o computador remoto, pela placa de rede *wireless*.

Pelo motivo de haver mais sinais a serem convertidos pelo conversor analógico-digital do que entradas do mesmo, foi utilizado um multiplexador analógico, modelo 74LS4051, para que alguns sinais fossem medidos na mesma entrada do conversor analógico-digital.

4.3.2 Componentes utilizados

- Sensor de distância Sharp 2Y0A02;
- Microcontrolador MSC1210Y5;
- Fusíveis de 20A (Sensores de corrente);

- Amplificador Operacional OPA4277;
- Amplificador de Instrumentação INA128;
- Multiplexador analógico 74LS4051.

4.3.3 Diagrama em bloco com descrição detalhada

Neste tópico é apresentado o diagrama em blocos do hardware e comentado o seu funcionamento.

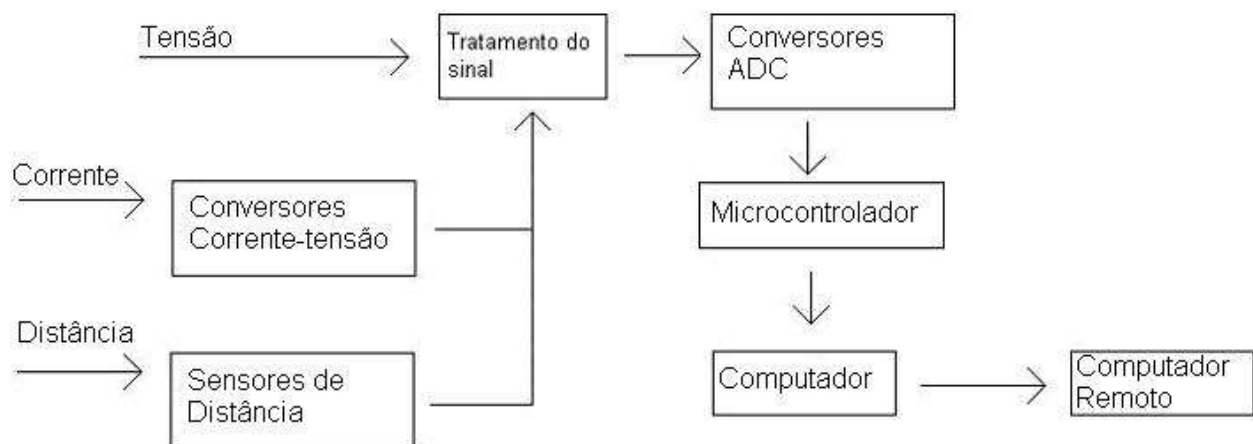


Figura 4 - Diagrama em blocos do Hardware

Os sinais de corrente passam pelos sensores de corrente, em seguida o sinal é tratado, passando por amplificadores de instrumentação onde é aplicado um ganho em cima desse sinal, a fim de habilitá-lo à conversão, após isto ele passa por um somador de tensão, a fim de impedir tensões negativas quando a corrente é invertida.

O bloco de sensores de distância nos dá na saída uma tensão relativa à distância que ele está do objeto.

Os sinais da tensão sofrem tratamento para se adequarem à faixa de tensão que o conversor analógico-digital interno ao microcontrolador pode converter.

Após todos os sinais já estarem tratados, passam por um conversor analógico-digital. Esse conversor tem capacidade de converter 8 sinais analógicos diferentes, como precisamos converter mais sinais, será utilizado um multiplexador analógico, que pode colocar numa mesma entrada do conversor analógico-digital mais de um sinal, em tempos diferentes.

Após a conversão, o microcontrolador irá enviar as informações para o computador embarcado no robô. O software que está rodando nesse computador envia as informações para o computador remoto, através da placa de rede *wireless*.

O software que está rodando no computador remoto toma decisões das direções que o robô irá tomar, que são se o robô continuará andando para frente, se irá fazer uma curva de 90° para direita ou para esquerda. Além de gravar num histórico, com intervalo de tempo pré-determinado pelo usuário, os valores de corrente e tensão lidos do robô e o dia e hora da leitura. Há também a possibilidade de gravar o resultado do mapeamento em uma imagem no momento em que o usuário necessitar, além de que ela é gravada no momento em que o programa é fechado, impedindo possíveis perdas por descuido do usuário. Esta gravação é feita para posterior avaliação ou comparação do mapa.

4.4 Especificação do Software

O *software* foi implementado na linguagem C++, utilizando a ferramenta de programação Borland Builder 6.

Este *software*, que foi concebido em ambiente gráfico, facilitará o entendimento do usuário quando o mesmo recebe o resultado do mapeamento. A utilização de um ambiente gráfico se dá por ser mais compreensível e de fácil utilização pelo usuário.

O layout da tela principal do *software* se dá da seguinte forma, na parte superior há um menu, onde o usuário pode escolher qual modo quer que o *software* funcione,

ou sair do programa. Na parte inferior há uma barra de status, onde mensagens são mostradas ao usuário, conforme pode-se verificar na Figura 5.

Na tela do modo remoto, na parte superior há um menu, a primeira opção do menu é a escolha de modo, onde o usuário poderá mudar o modo em que o *software* está rodando, ou desabilitar os modos, voltando para a tela principal do *software*. A segunda opção é de conexão, onde o usuário poderá fazer uma conexão com o *software* no computador embarcado no robô, fechar essa conexão ou desligar o robô. A terceira opção é de mapeamento, em que o usuário poderá escolher para iniciar o mapeamento, somente se tiver conectado com o *software* do computador embarcado no robô, parar esse mapeamento, ou iniciar um novo. A quarta opção é onde o usuário poderá setar o intervalo de tempo em que deseja que os valores de tensão das baterias e de alimentação do computador embarcado no robô, e das correntes consumidas por ele e pelos motores sejam gravados, além de iniciar e parar essa gravação. A quinta opção é para a visualização de um relatório dos valores de tensão das baterias e de alimentação do computador embarcado no robô, e das correntes consumidas por ele e pelos motores lidos e salvos anteriormente. A sexta opção serve caso o usuário deseje salvar o mapeamento naquele momento. A imagem é salva com o seguinte nome: AAAAMMDD_HHMMSS, onde AAAA é o ano em que a imagem foi feita, MM o mês, DD o dia, HH a hora, MM o minuto e SS o segundo.

Ao lado direito dessa tela, são mostrados os valores que estão sendo lidos das tensões das baterias e de alimentação do computador embarcado no robô, e das correntes consumidas por ele e pelos motores. Ao centro, e ocupando o lado esquerdo, é mostrada a imagem atual em que se encontra o mapeamento, sendo ela toda cinza quando é um novo mapeamento, ou quando um mapeamento já foi feito, a imagem é cinza nas partes em que o robô não passou, tendendo a branca nas partes em que ele passou e não encontrou nenhum objeto, e tendendo a preto nas partes em que ele passou e encontrou algum objeto. Esta tela está demonstrada na Figura 8.

Na tela do modo do local, na parte superior há um menu, a primeira opção do menu é a escolha de modo, onde o usuário poderá mudar o modo em que o *software* está rodando, ou desabilitar os modos, voltando para a tela principal do *software*. A segunda opção é para sair. Na parte esquerda da tela há oito caixas de texto, onde são mostrados os valores lidos dos seis sensores e o número de voltas que as duas rodas

andaram. Abaixo destas caixas de texto, há dois botões, que se pressionados, vai ler os valores dos sensores, ou os valores de tensão e corrente. Do lado direito há onze caixas de texto, que mostram os valores de tensão e corrente lidos. Esta tela está demonstrada na Figura 6.

Na tela do modo de controle, na parte superior há um menu, a primeira opção do menu é a escolha de modo, onde o usuário poderá mudar o modo em que o software está rodando, ou desabilitar os modos, voltando para a tela principal do software. A segunda opção é de conexão, onde o usuário poderá fazer uma conexão com o software no computador embarcado no robô, fechar essa conexão ou desligar o robô. A terceira opção é onde o usuário poderá setar o intervalo de tempo em que deseja que os valores de tensão das baterias e de alimentação do computador embarcado no robô, e das correntes consumidas por ele e pelos motores sejam gravados, além de iniciar e parar essa gravação. A quarta opção é para sair do programa.

Abaixo deste menu, há um texto, que informa como movimentar o robô, abaixo outro texto, informando para que direção o robô vai, conforme a tecla que foi apertada. Abaixo destes textos, tem onze caixas de texto, e mais abaixo mais oito indicações, sendo seis delas mostrando a que distância cada sensor esta de algum objeto, e duas delas mostrando quantas voltas cada roda andou. Esta tela está demonstrada na Figura 7.

A imagem onde vai ser mostrado o mapeamento, é um conjunto de *grids*, que representa o ambiente em que o robô está se movimentando, na forma de grades de ocupação, como fala Dutra^[3]. Estes *grids* terão cores diferentes, representando locais pelos quais o robô ainda não passou, locais pelos quais ele já passou e verificou que tem obstáculos, e lugares que estão livres. O nível de intensidade dessas cores será denotado pelo número de vezes que o robô passa por um mesmo local.

Cada *grid* tem um valor numérico, e este valor é definido pelo resultado do bloco de sensores de distância naquele *grid*, Quanto mais vezes o robô passar por um mesmo *grid*, e o resultado for o mesmo, maior ou menor será número naquele *grid*. Este número inicialmente é 128 (cento e vinte e oito), cor cinza, que indica uma região inexplorada. *Grids* com números menores que o inicial, tendendo à cor preta, indicam

regiões ocupadas, com obstáculos, com números maiores que o inicial, tendendo à cor branca, indicam regiões livres.

Pode ser que ocorra um ruído, por exemplo, um animal passando perto do robô, o robô identificará um objeto no local, mas da próxima vez que passar pelo mesmo local, identificará um local livre, anulando o ruído que aquele animal provocou.

Cada *grid* reconhecido pelo robô tem um tamanho de 20cm por 20 cm. O valor do conjunto de *grid* será armazenado em forma de imagem. O tamanho máximo que essa imagem pode ter é de 500 x 500. Podendo então, o robô mapear uma área de tamanho máximo igual a 100m².

5 HARDWARE

5.1 Esquemáticos

Neste tópico serão descritos os esquemáticos do *hardware*, que se encontram em anexo.

No anexo 01 se encontra a primeira folha dos esquemáticos, onde são mostrados os sinais de entrada dos sensores de corrente dos dois motores, e da corrente total que o computador embarcado no robô está consumindo. Estes sinais passam por amplificadores de instrumentação. Os amplificadores de instrumentação recebem um sinal de antes da passagem pelo sensor de corrente, e outro após essa passagem, um ganho é aplicado à diferença dos sinais de entrada, a diferença desses sinais, com o ganho aplicado é a saída.

No anexo 02 se encontra a segunda folha dos esquemáticos, onde são mostrados os sinais de entrada de tensão das duas baterias, e dos sinais de 12V da placa mãe e do HD. O sinal da bateria dos motores passa por um amplificador de instrumentação e após isso, junto com o sinal da outra bateria passam pelo amplificador operacional somente como um buffer, mas com um divisor de tensão na entrada, para que a tensão das baterias seja limitada entre 0 e 5 V. Os sinais de 12V, da placa mãe e do HD, também passam por um divisor de tensão e por um buffer, a passagem pelo divisor de tensão serve para que estes sinais se tornem adequados à conversão que será feita posteriormente. A passagem destes sinais por amplificadores operacionais serve para que não haja perda de corrente para o circuito de conversão.

No anexo 03 se encontra a terceira folha dos esquemáticos, onde são mostrados os sinais de 5V da placa mãe e do HD, além dos sinais de -5V e -12V da placa mãe. Os sinais de 5V passam pelos amplificadores operacionais como um buffer somente, já os sinais de -5V e -12V passam por buffers inversores, para que possam ser

convertidos posteriormente. O sinal de -12V passa, antes do buffer inversor, por um divisor de tensão.

No anexo 04 se encontra a quarta folha de esquemáticos, onde são mostradas as entradas de todos os sinais tratados e prontos para a conversão, o sinal de entrada da serial, o microcontrolador e o circuito necessário para que o mesmo funcione. Como temos mais que 8 sinais, e o conversor analógico-digital interno ao microcontrolador só tem 8 entradas, usamos um multiplexador analógico. Portanto os sinais das duas baterias, da corrente do motor 2 e da corrente da placa mãe passam primeiramente por este multiplexador, para então entrarem, um de cada vez, na entrada AIN7 do conversor analógico-digital.

No anexo 05 se encontra a quinta folha de esquemáticos, onde são mostrados os conversores de tensão, a entrada é uma tensão que pode variar de 9 a 12V. Esta tensão passa pelo 7805, que converte para 5V. Em seguida, o 5V entra no PT5062, que converte para tensões de 15V e -15V.

5.2 Sinais de interface

Aqui serão mostrados os sinais de entrada e saída das folhas de esquemáticos, e o significado deles.

5.2.1 Primeira Folha dos esquemáticos

Na tabela 2 são mostrados os sinais de entrada e saída da primeira folha de esquemáticos, e ao lado o significado de cada um deles.

15V	Entrada de 15V
-15V	Entrada de -15V
GND	Entrada de Terra
CorrenteMotor1a	Entrada da tensão antes do resistor de shunt
CorrenteMotor1b	Entrada da tensão após do resistor de shunt
CorrenteMotor2a	Entrada da tensão antes do resistor de shunt
CorrenteMotor2b	Entrada da tensão após do resistor de shunt
AlimentacaoPCa	Entrada da tensão antes do resistor de shunt
AlimentacaoPCb	Entrada da tensão após do resistor de shunt
CorrenteMotor1Saida	Saída da tensão equivalente à corrente do Motor1
CorrenteMotor2Saida	Saída da tensão equivalente à corrente do Motor2
CorrentePCSaida	Saída da tensão equivalente à corrente da placa do PC

Tabela 2 - Sinais da primeira folha de esquemáticos

5.2.2 Segunda Folha dos Esquemáticos

Na tabela 3 são mostrados os sinais de entrada e saída da segunda folha de esquemáticos, e ao lado o significado de cada um deles.

15V	Entrada de 15V
-15V	Entrada de -15V
GND	Entrada de Terra
Bateria1	Entrada da tensão da primeira bateria
Bateria1	Entrada da tensão da segunda bateria
12V PC	Entrada da tensão de 12V do PC
12V HD	Entrada da tensão de 12V do HD
Bateria1Saida	Saída da tensão equivalente à tensão da Primeira Bateria
Bateria2Saida	Saída da tensão equivalente à tensão da Segunda Bateria
12V PC Saída	Saída da tensão equivalente à tensão de 12V do PC
12V HD Saída	Saída da tensão equivalente à tensão de 12V do HD

Tabela 3 - Sinais da segunda folha de esquemáticos

5.2.3 Terceira Folha dos Esquemáticos

Na tabela 4 são mostrados os sinais de entrada e saída da terceira folha de esquemáticos, e ao lado o significado de cada um deles.

15V	Entrada de 15V
-15V	Entrada de -15V
GND	Entrada de Terra
5V PC	Entrada da tensão de 5V do PC
5V HD	Entrada da tensão de 5V do HD
-5V PC	Entrada da tensão de -5V do PC
-12V HD	Entrada da tensão de -12V do HD
5V PC Saída	Saída da tensão equivalente à tensão de 5V do PC
5V HD Saída	Saída da tensão equivalente à tensão de 5V do HD
-5V PC Saída	Saída da tensão equivalente à tensão de -5V do PC
-12V HD Saída	Saída da tensão equivalente à tensão de -12V do HD

Tabela 4 - Sinais da terceira folha de esquemáticos

5.2.4 Quarta Folha de Esquemáticos

Na tabela 5 são mostrados os sinais de entrada e saída da quarta folha de esquemáticos, e ao lado o significado de cada um deles.

5V	Entrada de 5V
GND	Entrada de Terra
12V PC Saída	Entrada da tensão referente aos 12V do PC
12V HD Saída	Entrada da tensão referente aos 12V do HD
5V PC Saída	Entrada da tensão referente aos 5V do PC
5V HD Saída	Entrada da tensão referente aos 5V do HD
-5V PC Saída	Entrada da tensão referente aos -5V do PC
-12V PC Saída	Entrada da tensão referente aos -12V do PC
CorrenteMotor1Saída	Entrada da tensão referente à corrente do motor 1
CorrenteMotor2Saída	Entrada da tensão referente à corrente do motor 2
Bateria1Saída	Entrada da tensão referente à tensão da bateria 1
Bateria2Saída	Entrada da tensão referente à tensão da bateria 2
CorrentePCSaída	Entrada da tensão referente à corrente total consumida pelo PC

Tabela 5 - Sinais da quarta folha de esquemáticos

5.2.5 Quinta Folha de Esquemáticos

Na tabela 6 são mostrados os sinais de entrada e saída da quinta folha de esquemáticos, e ao lado o significado de cada um deles.

5V	Saída de tensão de 5V
15V	Saída de tensão de 15V
-15V	Saída de tensão de -15V
GND	Saída de terra

Tabela 6 - Sinais da quinta folha de esquemáticos

5.3 Lista de materiais

A seguir será mostrada a lista de materiais necessária para a confecção do hardware.

Quantidade	Referência	Nome
2	c1,c2	18 pF
2	C3,C11	10 uF
2	C4,C5	300nF
4	C6,C7,C8,C9	1 uF
1	C10	10 nF
1	J5	Conector de Alimentação
1	P1	CONNECTOR DB9
3	R1,R3,R5	RESISTOR SHUNT
3	R2,R4,R6	5,6K
9	R7,R9,R11,R13,R15,R16,R17,R19,R20	1K
5	R8,R10,R12,R14,R18	555R
1	R21	10K
1	R22	2K2
1	SW1	<i>Push-button Reset</i>
1	SW2	<i>Dip-switch 2</i>
1	SW4	Chave H-H 90°
3	U1,U2,U3	INA128/BB
1	U3 e U4	MSC1210Y5
1	U5	LM7805
1	U6	MAX232
1	U7	74LS4051
1	Y1	Cristal de 11,059 MHz
2	1A,2A	OPA4277PA
1	3A	PT5062C

Tabela 7 - Lista de Componentes

6 SOFTWARE

6.1 Software

O software tem como principais objetivos controlar o robô, receber informações dos microcontroladores e realizar o mapeamento propriamente dito. Permitindo ao usuário que interaja com o robô, ou que tenha acesso instantâneo ao mapeamento que está sendo feito. O software é dividido em três subsoftwares distintos, que são chamados modo de controle, modo do robô e modo de mapeamento.

Esses três modos são acionado um de cada vez, dentro da janela principal do software. A janela principal do software pode ser vista na figura 5.



Figura 5 - Tela principal do Software

O modo do robô recebe informações de dois microcontroladores, pelas portas seriais, COM1 e COM2, trata essas informações, e manda para o modo de mapeamento ou de controle essas informações quando requisitado. Além de comandar a direção em que o robô irá andar, conforme informação recebida do modo de controle ou de mapeamento. A figura 6 mostra como é a tela deste modo.



Figura 6 - Tela do modo do robô

Quando o usuário aperta as flechas direcionais, ou o espaço, no modo de controle, este manda informações para o modo do robô, mandando que ele vire para algum lado, ande para frente ou para trás, ou que pare. O usuário pode setar um intervalo de tempo para que seja requisitadas as informações dos microcontroladores pelo modo do robô, e que sejam mostradas neste modo. A tela do modo de controle é mostrada na figura 7.

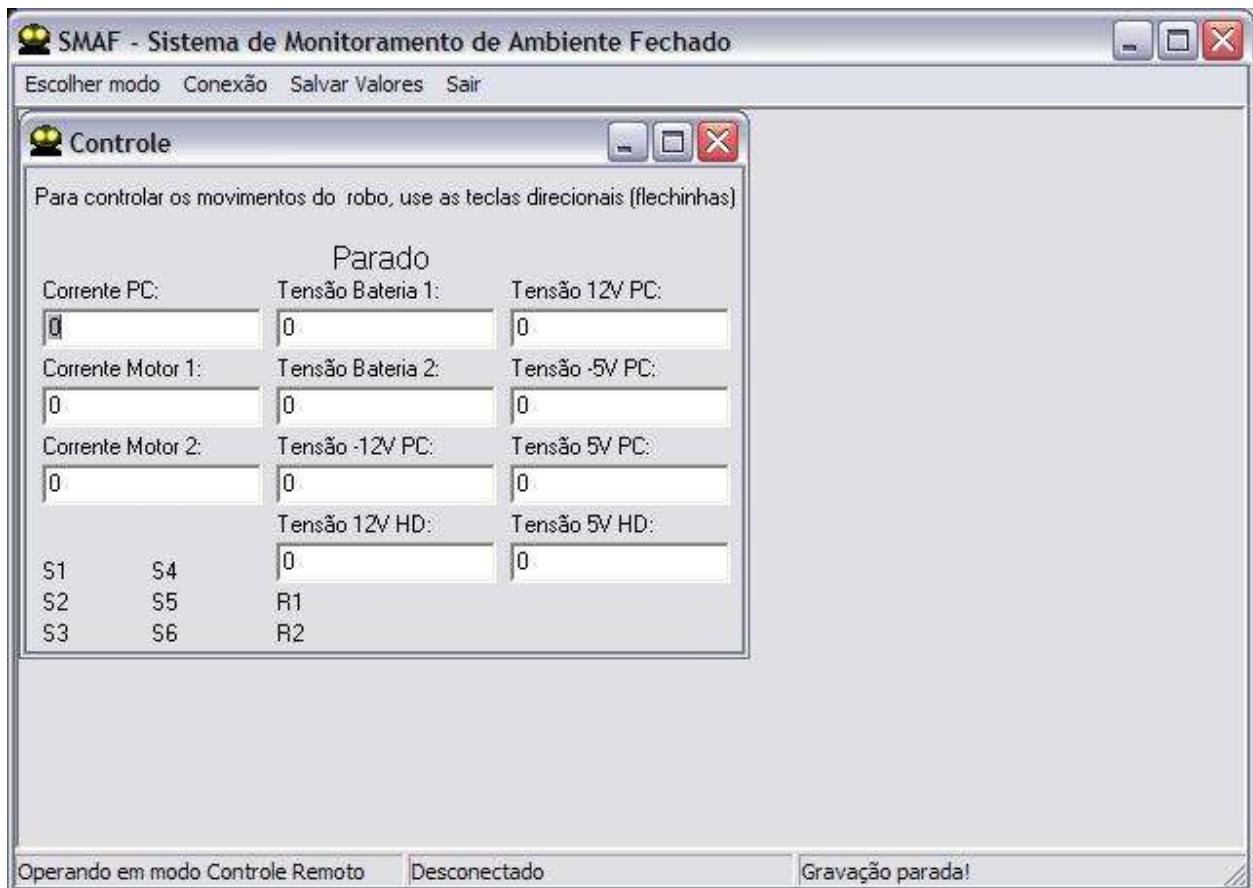


Figura 7 - Tela do modo Controle

No modo de mapeamento, ocorre a requisição, para o modo do robô, das informações do microcontrolador dos sensores, tratamento dessas informações e decisão de que direção tomar, além de desenhar o mapa na tela para que o usuário possa vê-lo. O usuário pode também setar um intervalo de tempo para que seja requisitada as informações do outro microcontrolador pelo modo do robô. A tela deste modo é mostrada na figura 8.

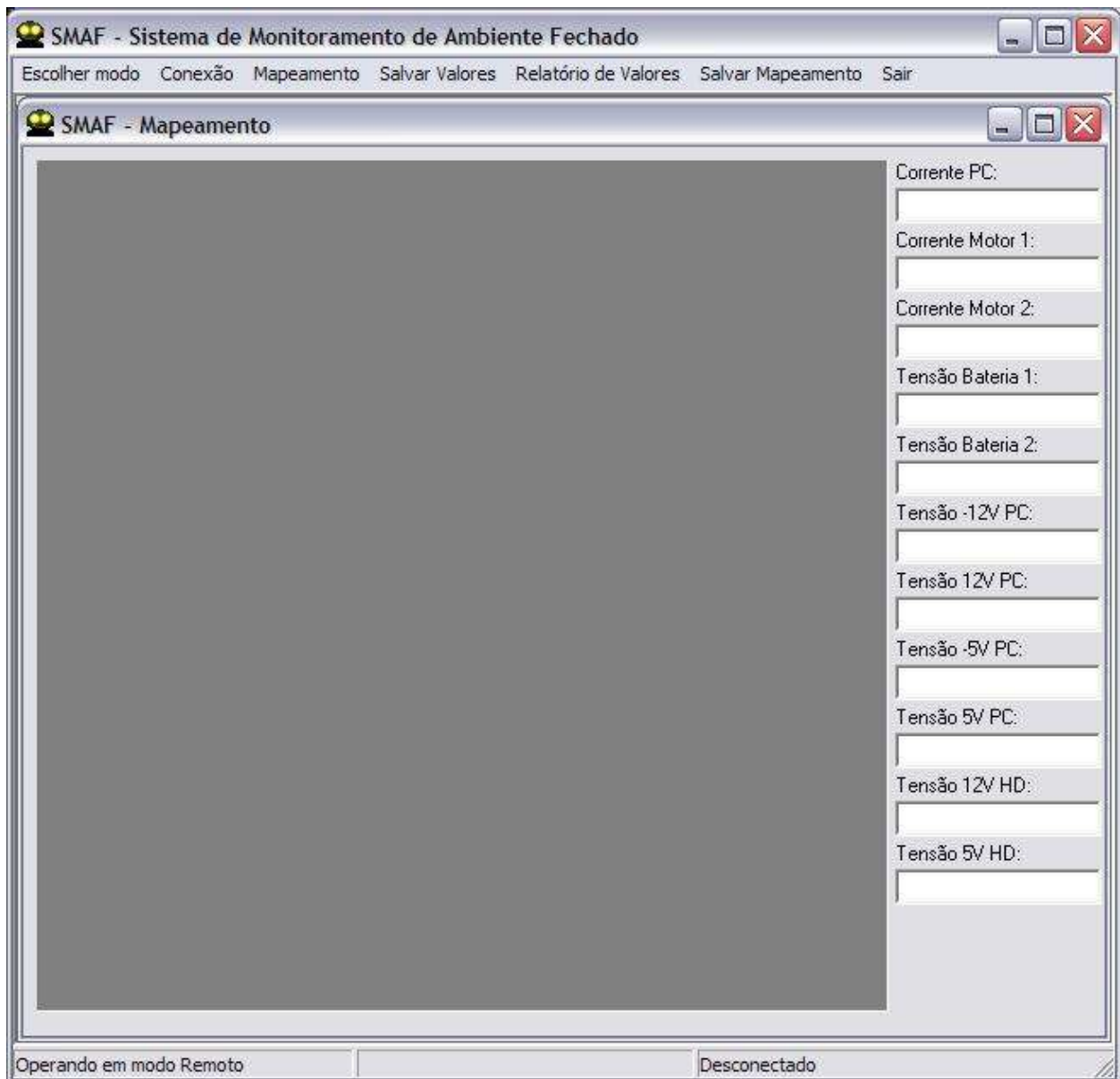


Figura 8 - Tela do modo Mapeamento

Todas as requisições e troca de mensagem são feitas por conexão TCP, que deve ser iniciada entre os modos antes de qualquer pedido ser feito.

6.1.1 Diagrama de casos de uso

O diagrama de casos de uso é mostrado na Figura 9. Ocorrem seis principais casos de uso.

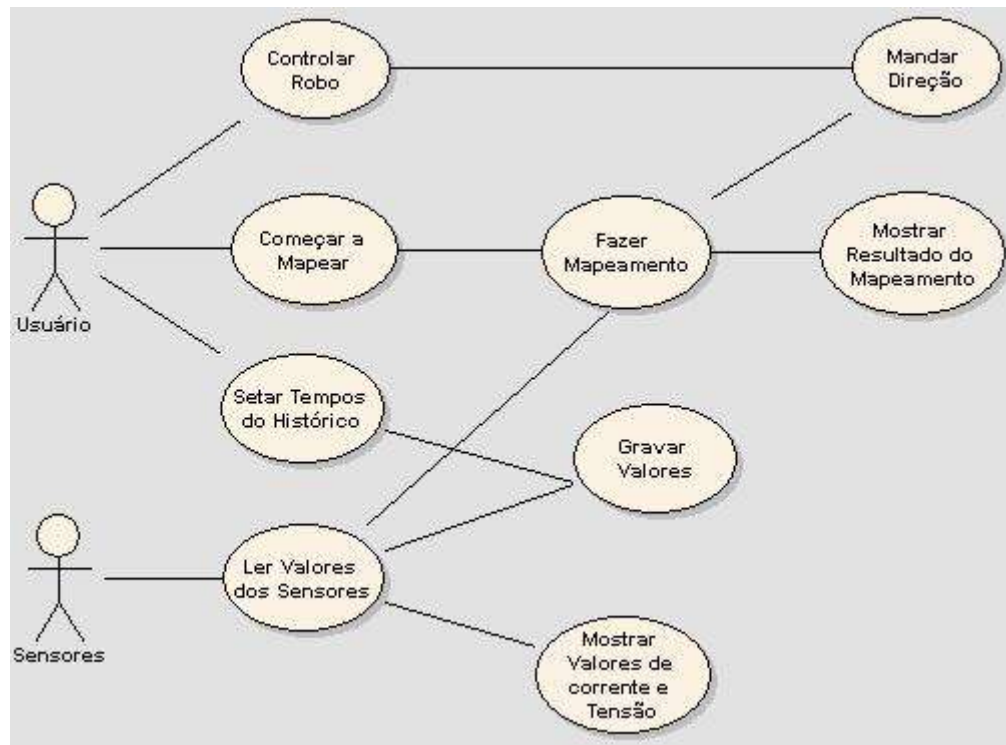


Figura 9 - Diagrama de casos de uso

6.1.1.1 Controlar Robô

A figura 10 mostra o diagrama de seqüência do caso de uso controlar robô. O propósito deste caso de uso é controlar a direção em que o robô irá se mover.

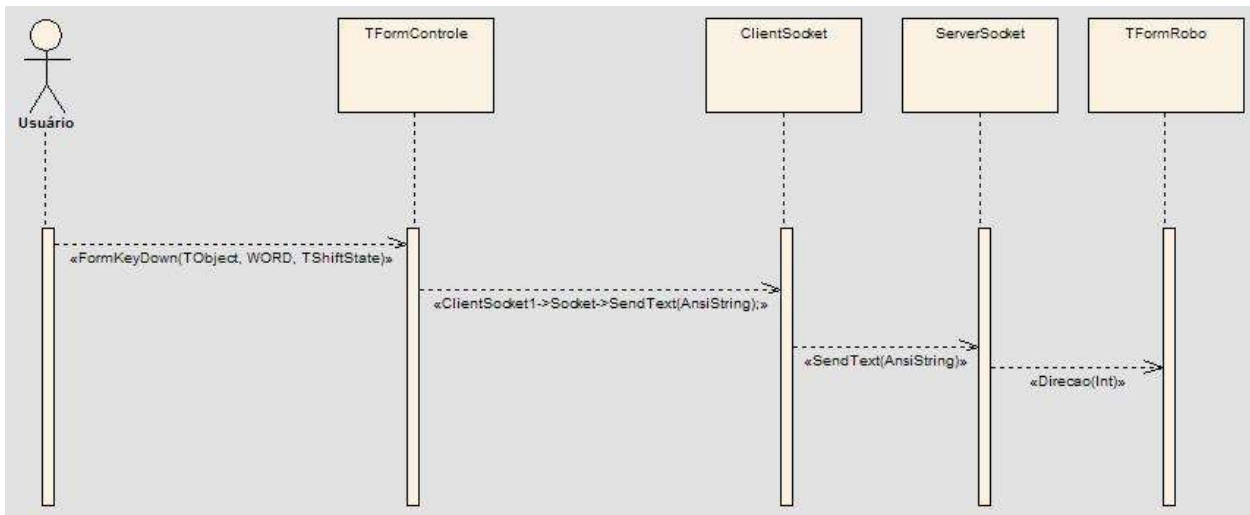


Figura 10 - Diagrama de Seqüência para o Caso de Uso Controlar Robô

6.1.1.2 Começar a Mapear

A figura 11 mostra o diagrama de seqüência do caso de uso Começar a Mapear. O propósito deste caso de uso é começar o mapeamento.

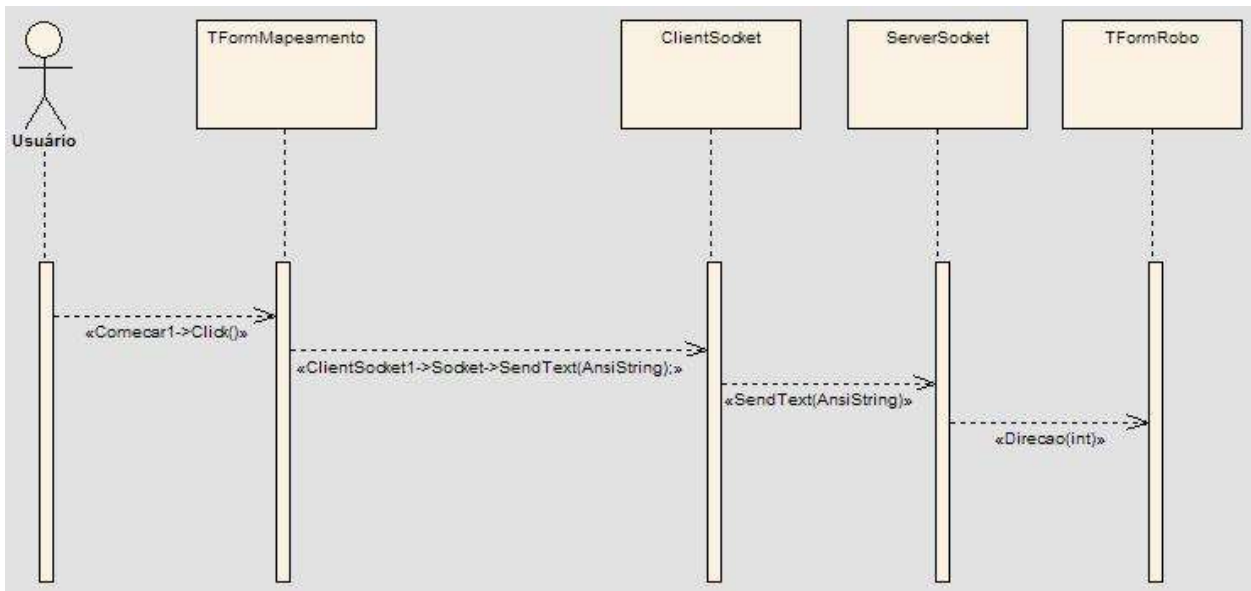


Figura 11 - Diagrama de Seqüência para o Caso de Uso Começar a Mapear

6.1.1.3 Fazer Mapeamento

A figura 12 mostra o diagrama de seqüência do caso de uso Fazer Mapeamento. O propósito deste caso de uso é fazer o mapeamento.

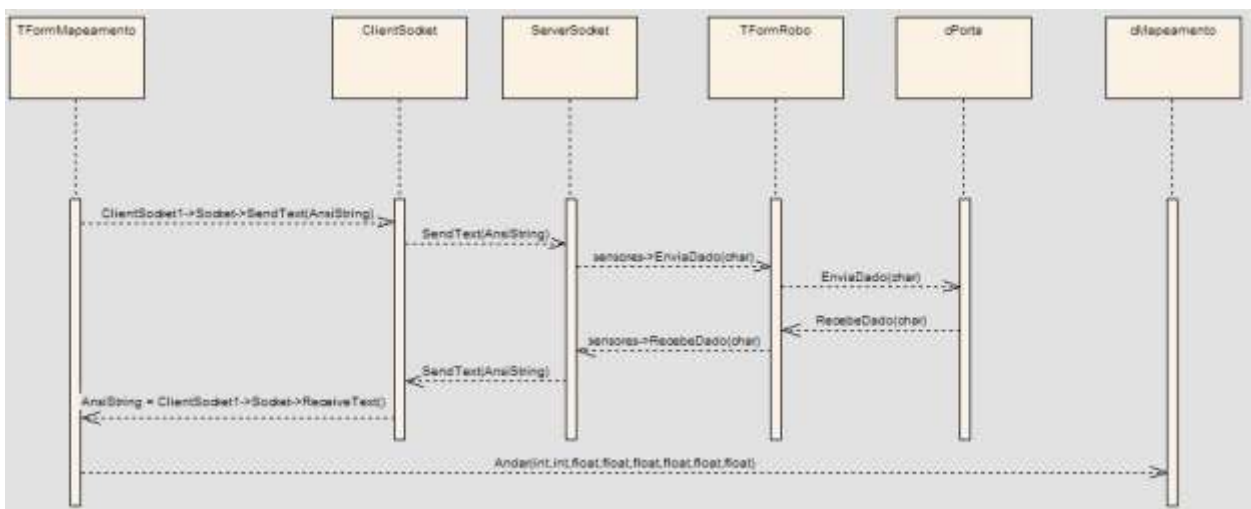


Figura 12 - Diagrama de Seqüência para o Caso de Uso Fazer Mapeamento

6.1.1.4 Setar Tempos do Histórico

A figura 13 mostra o diagrama de seqüência do caso de uso Setar Tempos do Histórico. O propósito deste caso de uso é setar o intervalo de tempo em que os valores serão salvos no banco de dados.

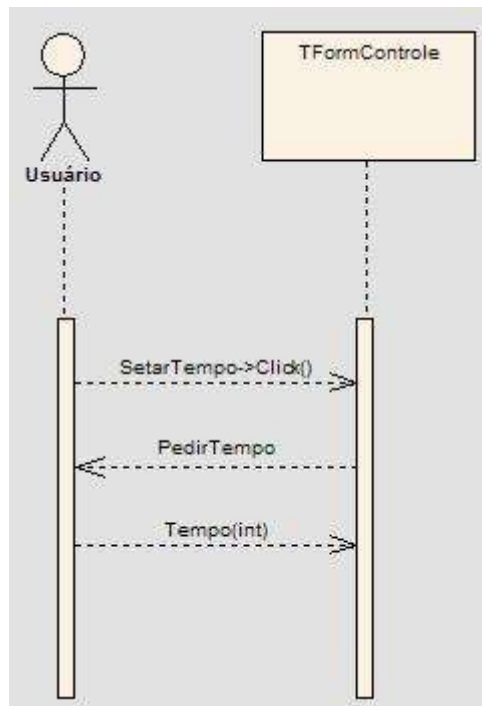


Figura 13 - Diagrama de Seqüência para o Caso de Uso Setar Tempos do Histórico

6.1.2 Diagrama de Classes

O diagrama de classes do software é mostrado na Figura 14.

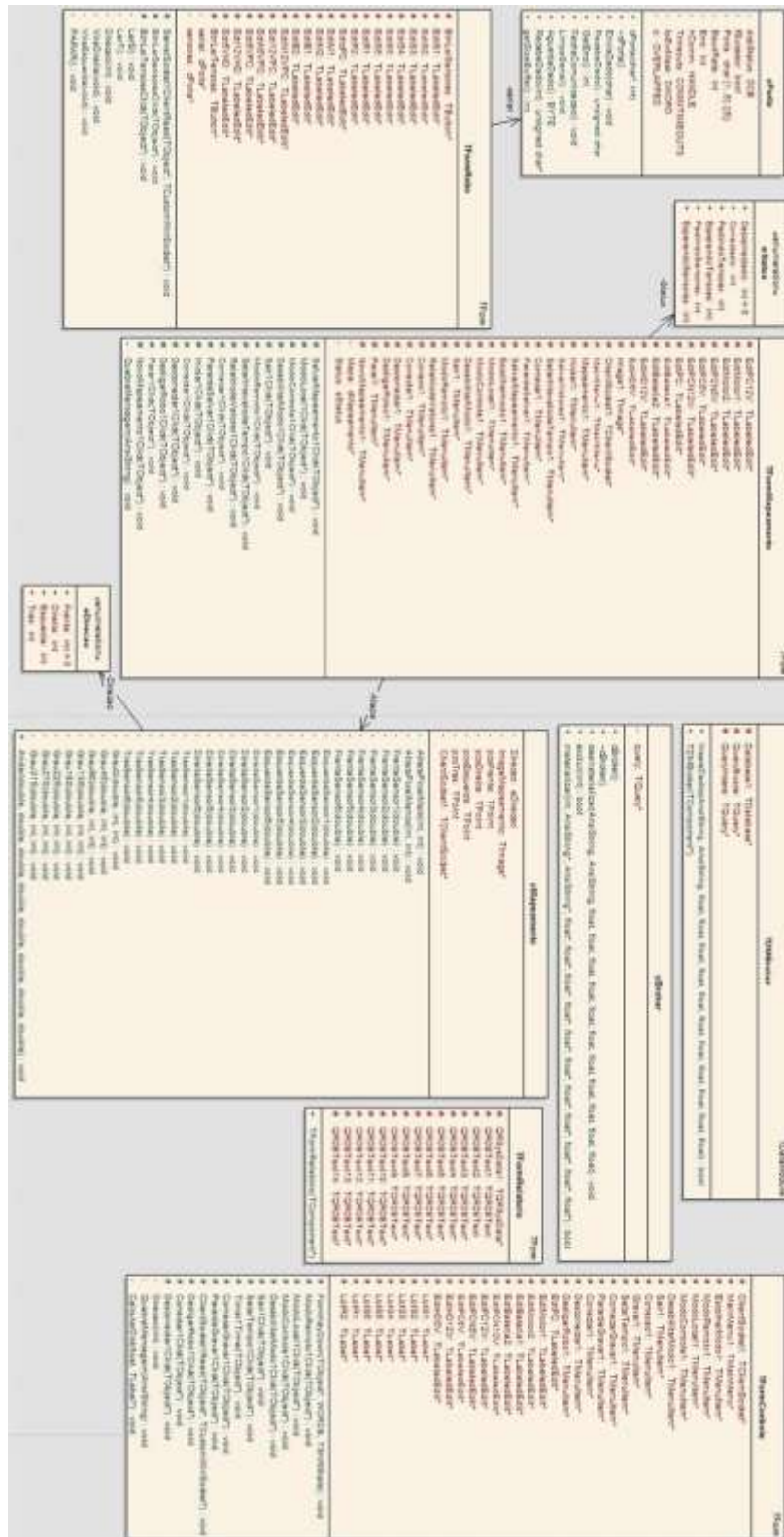


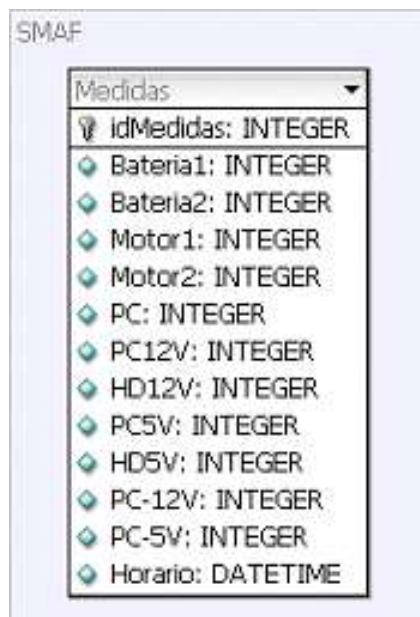
Figura 14 - Diagrama de Classes

6.2 Dados

Neste tópico será mostrado como está organizado o banco de dados que o *software* usa.

6.2.1 Modelo Relacional

A figura 15 mostra o modelo relacional do banco de dados, em que são salvos os valores de tensão das duas baterias, da alimentação do computador embarcado no robô, da corrente que este consome, e dos dois motores.



The image shows a screenshot of a database schema tool window titled 'SMAF'. It displays the structure of a table named 'Medidas'. The table has the following fields:

Field Name	Data Type
idMedidas	INTEGER
Bateria1	INTEGER
Bateria2	INTEGER
Motor1	INTEGER
Motor2	INTEGER
PC	INTEGER
PC12V	INTEGER
HD12V	INTEGER
PC5V	INTEGER
HD5V	INTEGER
PC-12V	INTEGER
PC-5V	INTEGER
Horario	DATETIME

Figura 15 - Modelo Relacional

6.2.2 Dicionário de Dados

A tabela 8 mostra o significado de cada campo do banco de dados.

Tabela Medidas	
Campo	Significado
idMedidas	Identificador
Bateria1	Valor de tensão na bateria 1
Bateria2	Valor de tensão na bateria 2
Motor1	Valor de corrente do motor1
Motor2	Valor de corrente do motor2
PC	Valor de corrente da placa mãe
PC12V	Valor de tensão no 12V do PC
HD12V	Valor de tensão no 12V do HD
PC5V	Valor de tensão no 5V do PC
HD5V	Valor de tensão no 5V do HD
PC-12V	Valor de tensão no -12V do PC
PC-5V	Valor de tensão no -5V do PC
Horário	Horário em que a medida foi feita

Tabela 8 - Dicionário de dados - Tabela Medidas

6.3 Firmware

Neste tópico será mostrado como está organizado o firmware que o microcontrolador roda.

6.3.1 Diagrama de estados

A figura 16 mostra o diagrama de estados do firmware, onde a variável “funcionando” determina se haverá conversão dos dados do conversor analógico-digital, ou se o microcontrolador deve ficar esperando. A variável é setada para 1 quando é enviado pela serial um char, no caso “i”.

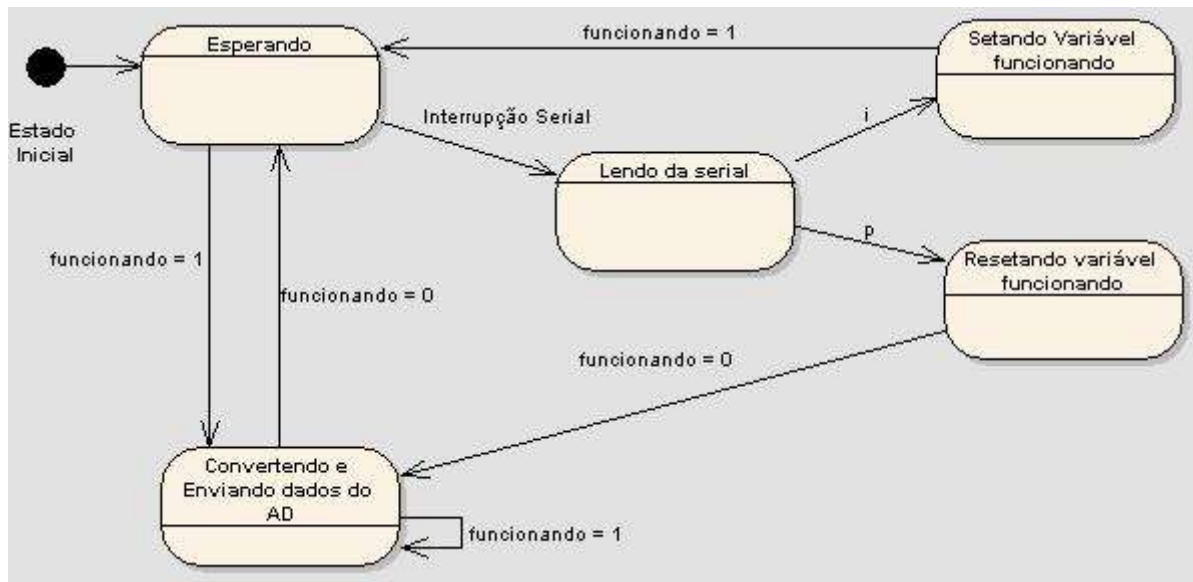


Figura 16 - Diagrama de estados

6.3.2 Fluxograma

A figura 17 mostra o fluxograma do *firmware* embarcado no microcontrolador. Pode-se ver que logo que o microcontrolador é iniciado, ele configura a serial, timers e o conversor analógico-digital. Após isto, fica preso por um laço infinito, onde somente sai quando acontece uma interrupção serial, e o que veio pela serial foi “i”. Quando “i” é recebido, o *firmware* começa a converter os valores, começando da entrada 1, mostrada quando a variável “nrAD” é igual a 0. O valor é lido, transmitido pela serial, e a variável “nrAD” é incrementada. Quando o valor dessa variável passar de 6, o microcontrolador endereça um multiplexador analógico, para que os sinais que estão nas entradas desse multiplexador sejam convertidos por uma única entrada, em tempos distintos.

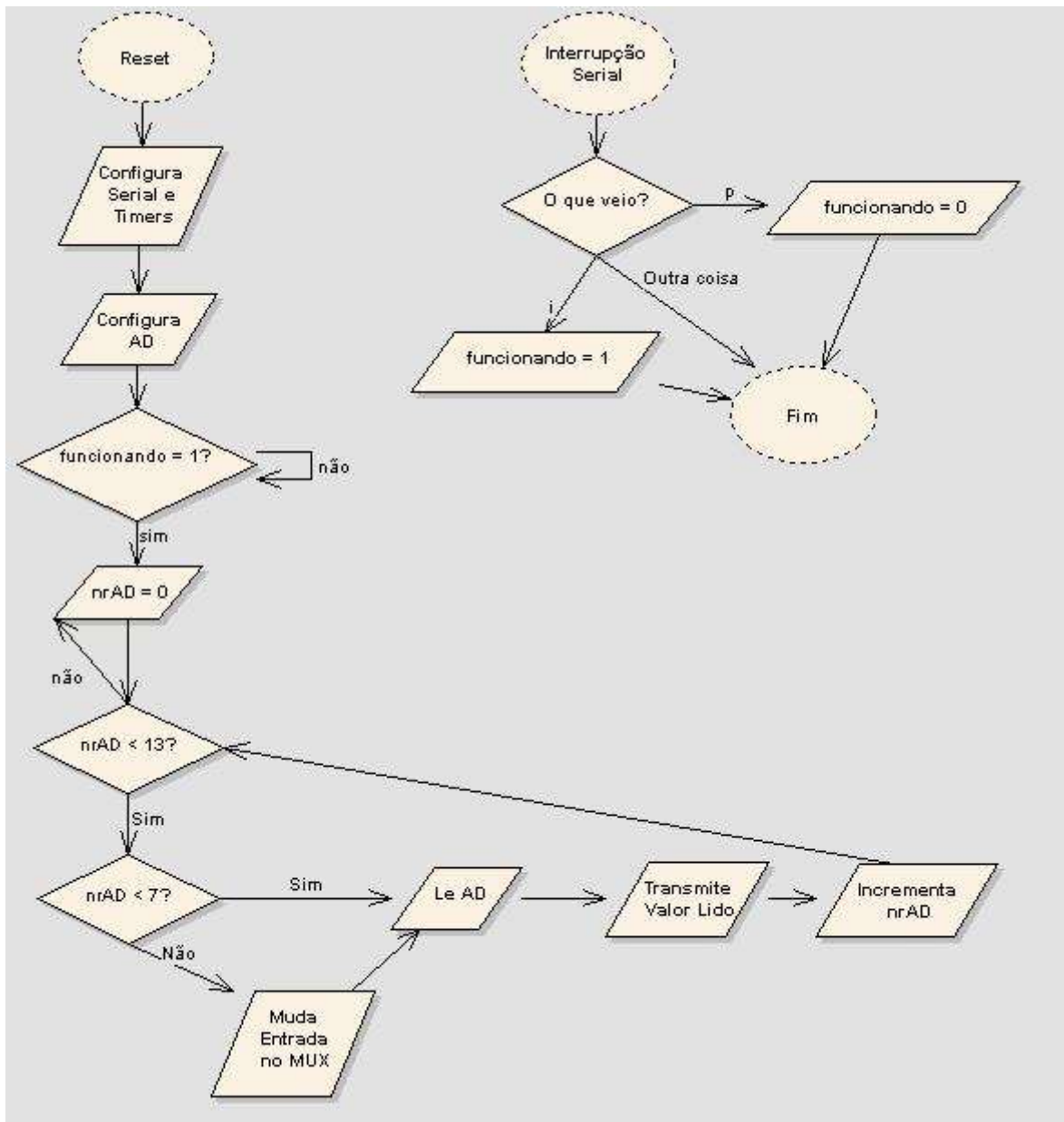


Figura 17 – Fluxograma

7 PROCEDIMENTOS DE TESTES DO PROJETO

Os testes foram feitos em uma sala com objetos fixos, na qual se sabe onde os objetos fixos estão dispostos dentro da mesma podendo então ser verificado, conforme o robô for andando, se o mapeamento está sendo feito de forma adequada.

8 CRONOGRAMA DE DESENVOLVIMENTO

O cronograma mostra as datas limites em que deverão ser entregues as partes do projeto final, e quais as atividades que serão realizadas em cada época.

O cronograma está no anexo 06, ao final do documento.

9 ESTUDO DE VIABILIDADE TÉCNICO-ECONÔMICA

Na tabela 10 está uma estimativa do custo do projeto. Esta estimativa não incorpora o custo de desenvolvimento do robô utilizado, nem de recursos como computador, sistema operacional, fontes analógicas, osciloscópios, multímetros, dentre outros, que foram fornecidos pelo UNICENP.

O custo dos materiais utilizados foi baixo, pois alguns destes foram mandados pelo fabricante sem custo algum, numa política denominada “envio de amostras”.

Recurso	Quantidade	Custo Unit. (R\$)	Custo do Recurso (R\$)
Microcontrolador MSC1210	1	U\$12,00*	33,60
Regulador de Tensão LM 7805	1	1,00	1,00
Regulador de tensão PT5062	1	U\$11,30*	31,64
Amplificador de Instrumentação INA128	4	U\$4,95*	55,44
Amplificador operacional OPA4277	3	U\$3,85*	32,34
MAX 232	1	U\$0,58*	1,63
Multiplexador Analógico 74LS4051	1	2,50	2,50
Crystal 11,0592 MHz	1	2,00	2,00
<i>Push-Buttons</i>	1	0,10	0,10
<i>Dip-switch</i>	1	0,30	0,30
Chave H-H 90°	1	2,00	2,00
Resistores diversos	20	0,10	2,00
Capacitores diversos	10	0,40	4,00
Conector JACK	5	0,50	2,50
Conector DB9 com capa	1	1,00	1,00
Barra de conectores fêmea torneada	10	2,00	20,00
Horas de Trabalho	900	10,00	9000,00
Borland C++ Builder Standard	1	2999,00	2999,00
		Total:	12191,05

* utilizado taxa de câmbio U\$1,00 = R\$ 2,80 em março de 2005.

Tabela 9 - Estimativa de custo

10 RESULTADOS

A seguir serão apresentados os resultados obtidos tanto para *hardware* quanto para *software*.

10.1 Hardware

Aconteceram alguns problemas no andamento do projeto que prejudicaram o hardware do mesmo. Dois microcontroladores MSC1210y5 queimaram, fazendo com que o hardware tivesse que ser deixado de lado, até um novo microcontrolador ser conseguido.

Outro problema que aconteceu, foi o conversor de tensão PT5062 também ter queimado, como não foi conseguido um novo regulador, impediu-se de ser colocado no robô o *hardware*, sendo este simulado em protoboard, com auxílio de fontes de energia.

O *firmware* projetado para o *hardware* funciona tanto para ele diretamente no robô, como em protoboard, portanto nenhuma mudança teve que ser feita.

A comunicação do *firmware* com o computador do robô, que é feita pela serial, tem uma velocidade de 19200 bps, fixa no *firmware* e no *software*. Já a comunicação entre o computador remoto e o computador do robô tem velocidade fixada pelas placas de rede *wireless*.

10.2 Software

A proposta inicial deste projeto era de se guardar as informações do mapeamento em um banco de dados, em forma de matriz, onde cada linha por coluna

seria um grid. A maneira como o mapeamento foi salvo mudou, utilizando-se uma imagem bitmap, pois o nível de dificuldade em se trabalhar com ela é muito menor do que quando trabalhado com uma matriz, além de ser muito mais fácil a visualização por parte do usuário no resultado do mapeamento.

Para a comunicação com o hardware, como já dito anteriormente feita pela serial, não foram utilizadas *threads*, pois grandes quantidades de informação só seriam recebidas após um comando emitido pelo próprio software.

Para receber informações dos sensores, o software manda o comando, e espera num looping até o buffer da serial atingir o tamanho desejado, que é o número de caracteres já previamente acertado, e então descarrega esse buffer, tratando após isto dos dados. Um código de segurança contra falhas de comunicação foi implementado dentro deste looping, este código verifica se uma variável, zerada logo antes de se enviar o comando pela serial, passa de um limite, se passar e o número de caracteres no buffer não for o correto, ele despreza esses dados, não tratando nenhum deles. Esse limite que o software pode ficar esperando a serial responder foi determinado através de testes.

Para a recepção dos valores de tensão das duas baterias, e de alimentação do computador embarcado no robô, da corrente que este consome e dos dois motores, o software manda um comando pela serial, e a recepção foi feita de maneira similar à comunicação com os sensores, diferindo somente ao limite de caracteres, que são três. Esses três caracteres indicam que a comunicação com o microcontrolador está funcionando. O mesmo código de proteção foi implementado nesta comunicação, diferindo somente no limite, que também foi determinado através de testes. Pelo motivo de os valores enviados para cada leitura do conversor analógico-digital ser variável, não se pode colocar um limite total no looping que lê o buffer da serial.

10.3 Resultados da validação

Na validação do projeto se obteve os resultados esperados. O robô andou pelo ambiente sem bater em objetos ou pessoas, desviando deles. Um caminho por onde

ele deveria andar foi feito, e o mapa que ele achou deste caminho é o mostrado na figura 18.

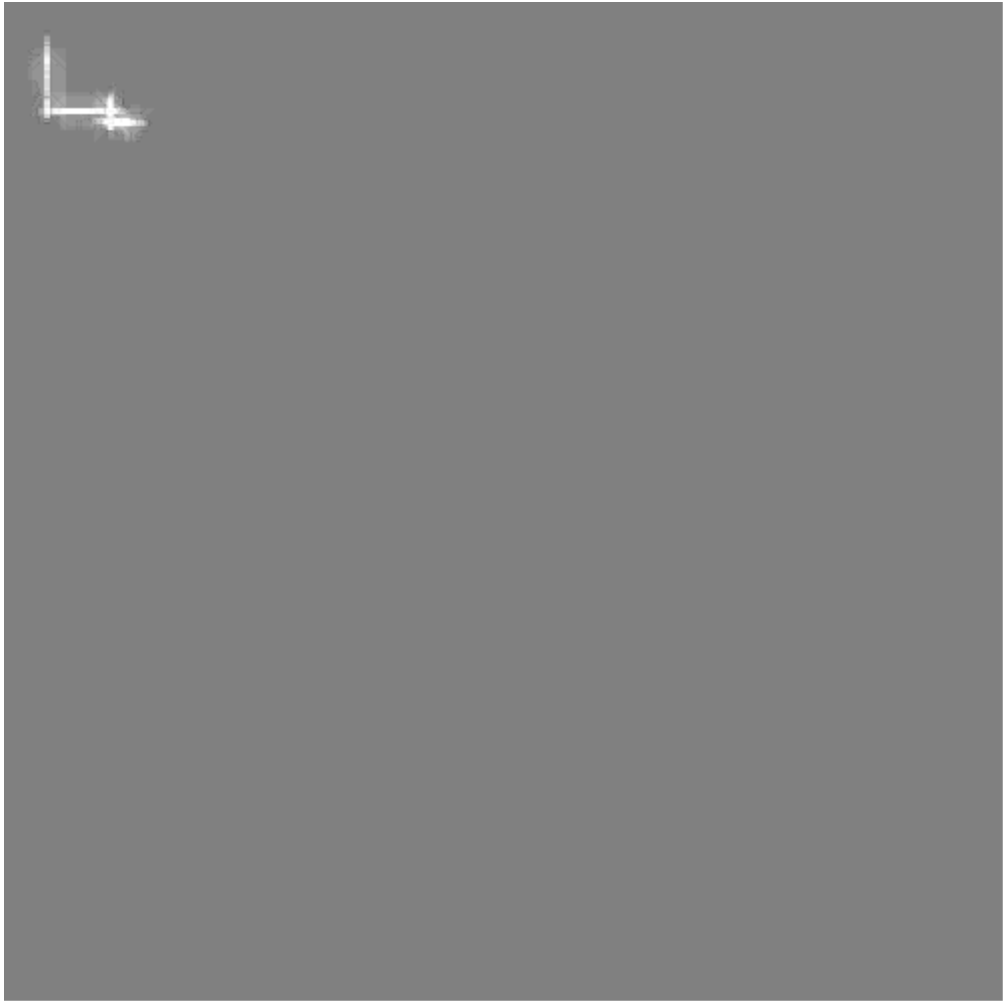


Figura 18 - Mapa resultante da validação

11 CONCLUSÕES

O projeto teve uma dificuldade razoavelmente grande com o microprocessador e com o conversor de tensão PT5062C, pois aconteceu a perda de 3 microcontroladores, 2 queimados e 1 com a serial sem funcionar, sem motivo aparente para esses problemas, e para a queima do conversor de tensão. Além desses microcontroladores que foram inutilizados, um quarto teve duas de suas entradas analógicas queimadas quando o PT5062C queimou, impedindo que essas entradas fossem utilizadas.

Apesar destes problemas, o *firmware* foi feito, e simulado num ambiente de programação para microcontroladores, testado num último microcontrolador e deixado funcional. Apesar deste ter 2 de suas 8 portas de entrada do conversor analógico digital queimadas quando o PT5062C queimou.

Com o conversor de tensão queimado, não se pode colocar o *hardware* funcionando em paralelo com o mapeamento.

O *software* funcionou corretamente, permitindo que um mapa fosse feito. Apesar de diferenças entre os dois motores, um andando mais que outro, fazendo com que o robô se movimentasse indo em direção a um lado, em vez de ir reto, o mapa se deu bem claro e coeso com o caminho que o robô fez.

Apesar dos problemas encontrados com o microcontrolador, gostei de trabalhar com ele, pois não há a necessidade de retirá-lo da placa para gravação, nem mesmo uso de memórias externas, a gravação se dá pela própria serial, tendo apenas que aterrar um pino e colocar outro em vcc.

O fato de fazer com que o robô não colidisse e ao mesmo tempo mapeasse o ambiente foi um desafio proveitoso de se vencer, pois é um objeto móvel, que reage ao ambiente que ele se encontra.

12 REFERÊNCIAS BIBLIOGRÁFICAS

[1] – Freire, E.O, *Controle de Robôs Móveis por Fusão de Sinais de Controle Usando Filtro de Informação Descentralizado*, Agosto de 2002, tese de doutorado submetida a Universidade Federal do Espírito Santo.

[2] – Thrun, Sebastian, *Robotic Mapping: A Survey*, Fevereiro de 2002, Universidade Carnegie Mellon, Escola de Ciência da Computação, Pittsburgho.

[3] – Dutra, P.; Sousa, M.; Andriolli, G.; Álvares, A.; Ferreira, J. *NAVMAP: Um Sistema Para Navegação Por Mapeamento Do Robô Móvel NOMAD XR4000*, Setembro de 2003, Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, apresentado no VI Simpósio Brasileiro de Automação Inteligente, Bauru.

13 BIBLIOGRAFIA

- Schildt, H.; Guntle, G.; Borland C++ Builder, a referência completa; tradução de Edson Furmankiewicz, Rio de Janeiro, Campus, 2001.

- Pertence Júnior, A; Eletrônica analógica: amplificadores operacionais e filtros ativos: teoria, projetos, aplicações e laboratório, Porto Alegre, Bookman, 2003

14 ANEXO

Os anexos seguem a seguir, sendo eles as folhas de esquemáticos do hardware, e o cronograma de desenvolvimento.

ANEXO 01 – ESQUEMÁTICO 01

ANEXO 02 – ESQUEMÁTICO 02

ANEXO 03 – ESQUEMÁTICO 03

ANEXO 04 – ESQUEMÁTICO 04

ANEXO 05 – ESQUEMÁTICO 05

ANEXO 06 – CRONOGRAMA