

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

ALEX AYRES STAVARENGO

**Telemetria de Dados e Imagens para Plataforma Autônoma para Coleta de
Dados Hidrológicos**

São Carlos
2006

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

ALEX AYRES STAVARENGO

**Telemetria de Dados e Imagens para Plataforma Autônoma para Coleta de
Dados Hidrológicos**

Dissertação apresentada à Escola de Engenharia
de São Carlos da Universidade de São Paulo para
obtenção do título de Mestre em Engenharia
Elétrica.

Área de Concentração: Processamento de Sinais
de Instrumentação
Orientador: Prof. Dr. Valentin Obac Roda

São Carlos
2006

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

S798t Stavarengo, Alex Ayres
Telemetria de dados e imagens para plataforma
autônoma para coleta de dados hidrológicos / Alex Ayres
Stavarengo ; orientador Valentin Obac Roda. -- São
Carlos, 2006.

Dissertação (Mestrado - Programa de Pós-Graduação e
Área de Concentração em Processamento de Sinais e
Instrumentação) -- Escola de Engenharia de São Carlos da
Universidade de São Paulo.

1. Telemetria. 2. GPS. 3. Microcontroladores.
4. Veículos autônomos. 5. Navegação. I. Título.

FOLHA DE JULGAMENTO

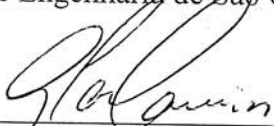
Candidato: Bacharel **ALEX AYRES STAVARENGO**

Dissertação defendida e julgada em 22-08-2006 perante a Comissão Julgadora:



Prof. Associado **VALENTIN OBAC RODA (Orientador)**
(Escola de Engenharia de São Carlos/USP)

aprovado



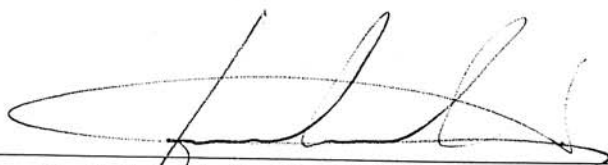
Prof. Associado **GLAUCO AUGUSTO DE PAULA CAURIN**
(Escola de Engenharia de São Carlos/USP)

APROVADO

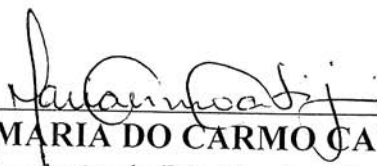


Prof. Dr. **ORIDES MORANDIN JUNIOR**
(Universidade Federal de São Carlos/UFSCar)

APROVADO



Prof. Associado **HOMERO SCHIABEL**
Coordenador do Programa de Pós-Graduação em
em Engenharia Elétrica



Profa. Titular **MARIA DO CARMO CALIJURI**
Presidente da Comissão de Pós-Graduação da EESC

DEDICATÓRIA

Aos meus pais, Antonio Carlos e Nêusa, que desistiram de muitos de seus sonhos
para tornarem possíveis os meus.

AGRADECIMENTOS

À DEUS, inteligência suprema, por todas as infinitas oportunidades de crescimento espiritual e profissional a mim concedidas.

As palavras falham, na tentativa de explicitar todo o meu agradecimento e carinho aos meus pais, Antônio Carlos e Nêusa, que em todos os momentos de minha vida tem me apoiado a prosseguir com serenidade e coragem. Este trabalho é também de vocês!

Gostaria de agradecer em especial ao orientador Valentin Obac Roda, pelo apoio, amizade, orientação, críticas e por todo o conhecimento que me foi transmitido e sem os quais este trabalho não seria concluído. Você é um exemplo de profissional, o qual não hesitarei em me espelhar.

À minha esposa Erika, por suportar as minhas longas ausências durante todos esses anos de trabalho.

Ao amigo de república Guilherme, com quem partilhei com muita alegria a maior parte de meu tempo durante essa fase pela busca do conhecimento.

Ao técnico de laboratório Ruiberto, pela efetiva participação e suporte na fabricação de todos os tipos de materiais para acabamento do projeto.

Aos colegas de laboratório de meu orientador, pelos momentos de companheirismo e alegria compartilhados.

Aos funcionários do Dep. de Engenharia Elétrica, que direta ou indiretamente, contribuíram com este trabalho.

À FAPESP, pelo suporte financeiro sem o qual este trabalho não seria realizado.

A todas as pessoas que de alguma forma me incentivaram ou contribuíram para a realização deste trabalho

“Pedras no caminho? Guardo todas. Um dia, vou construir um castelo.”

Fernando Pessoa

RESUMO

STAVARENGO, A. A. **Telemetria de dados e imagens para plataforma autônoma para coleta de dados hidrológicos**. 2006. 199f. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

A aquisição de dados das variáveis químicas e físicas da água de lagos e reservatórios é de extrema importância. Monitorando a qualidade da água distribuída e consumida em nossas cidades estamos protegendo o nosso recurso natural mais valioso e conseqüentemente nossas próprias vidas. Esse monitoramento na atualidade ainda se realiza de forma bastante convencional e trabalhosa. Esta dissertação apresenta o desenvolvimento de um sistema de telemetria de dados e vídeo completo entre uma plataforma autônoma para coleta de dados hidrológicos e uma estação de trabalho em terra. Este sistema fará uma composição com outros projetos já desenvolvidos nesta área de pesquisa, como a embarcação convencional utilizada como plataforma autônoma com seus motores, sensores e atuadores; a sonda de coleta de dados, responsável pela aquisição das variáveis químicas e físicas da água; sistema de posicionamento, que utiliza um *GPS* e bússola eletrônica para orientação, atuando e coletando dados dos sensores da embarcação; e sistema de gerenciamento, responsável por definir e orientar quais os pontos de coletas a serem pesquisados. Foram desenvolvidos algoritmos de comunicação entre os dispositivos de *hardware* existentes, efetivou-se a comunicação da plataforma autônoma e do microcomputador na estação terrestre através de um enlace de rádio frequência, desenvolveu-se um programa visual que permite o acompanhamento do estado dos sensores/atuadores do sistema em terra em tempo mínimo e habilita para o usuário um controle remoto de navegação da embarcação autônoma de forma manual e completa se ocorrer a falha de algum equipamento de posicionamento. O programa armazena todas as variáveis referentes ao processo de coleta em um banco de dados, permitindo a análise e comparação das diferentes medições realizadas em diferentes períodos.

Palavras Chave – telemetria, gps, microcontroladores, veículos autônomos, sistemas embarcados.

ABSTRACT

STAVARENGO, A. A. **Image and data telemetry for an autonomous water quality monitoring platform.** 2006. 199f. Master's Dissertation – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

The data acquisition of the water chemical and physical variables in lakes and reservoirs is of extreme importance. Monitoring the quality of water supplied and consumed in our cities we are protecting the most valuable of our natural resources, and consequently our own lives. Monitoring data quality, still in our days, is done in a conventional and hard manner. This work presents the development of a data and video telemetry system between an autonomous platform for water monitoring and a ground base. This system will complement the following already developed projects in this area, the autonomous platform constructed from a conventional boat with its engines, sensors and actuators; the data collecting probe, responsible for the acquisition of the water chemical and physical variables; the positioning system based on GPS and an electronic compass, responsible for navigation to the data collecting points in the reservoir. Communication algorithms between all the hardware were developed, radio frequency links were used to transmit data between the ground and the autonomous platform. A visual software application, that allows to visualize the state of the system variables, was developed. A remote control mode of the platform allows full manual navigation control of the platform, being of extreme importance in case of system crash. The visual software application developed allows to store all the important research variables in a data bank, allowing further analysis of the collected data.

Keywords – telemetry, gps, microcontrollers, autonomous vehicles, embedded systems.

LISTA DE FIGURAS

Figura 1. Embarcação utilizada como plataforma autônoma	15
Figura 2 . Sistema de botoeiras.....	16
Figura 3. Sistema completo do <i>hardware</i> embarcado	18
Figura 4. Espectro eletromagnético.....	40
Figura 5. Comparação de perdas entre uma linha coaxial e a atenuação no espaço livre utilizando a freqüência de 100MHz.....	41
Figura 6. Características dos sistemas de irradiação para projeto de antenas	46
Figura 7. Protótipo do barco	52
Figura 8. Diagrama em blocos do sistema projetado	53
Figura 9. Dirigível usado no projeto AURORA	54
Figura 10. <i>Hardware</i> do projeto ARARA	58
Figura 11. Módulo receptor GPS.....	62
Figura 12. Esquemático da interface do módulo receptor GPS.....	65
Figura 13. Vista completa da caixa aberta.	66
Figura 14. Módulo da bússola eletrônica.....	66
Figura 15. Esquemático da interface com a bússola eletrônica	68
Figura 16. Encapsulamento da bússola eletrônica.....	69
Figura 17. Kit de desenvolvimento da <i>Maxstream</i>	70
Figura 18. Exemplos de uso do aplicativo de diagnóstico dos <i>tranceivers</i>	72
Figura 19. Encapsulamento do <i>modem</i> R.F.	73
Figura 20. Kit completo para transmissão de áudio/vídeo.....	74
Figura 21. Estação terrestre	75
Figura 22. Componentes do enlace multimídia embarcado	76
Figura 23. Diagrama em blocos do sistema de telemetria de dados e imagens	78
Figura 24. Menu Arquivo e tela principal do aplicativo	81
Figura 25. Arquivo digital da telemetria de dados	81
Figura 26. Entrada de dados e otimização de rota.....	83
Figura 27. Fluxograma do algoritmo de otimização de rota	84
Figura 28. Pacote completo da telemetria de dados	86
Figura 29. Fluxograma do recebimento de pacotes da telemetria de dados.....	87
Figura 30. Tela principal do aplicativo para monitoração e controle da embarcação. 89	
Figura 31. Fluxograma do procedimento da bússola eletrônica.....	90

Figura 32. Fluxograma do procedimento do sensor GPS	91
Figura 33. Fluxograma dos sensores de navegação.....	92
Figura 34. Piloto virtual.....	94
Figura 35. Relatório da telemetria de dados	96
Figura 36. Canais seriais de todo o <i>hardware</i> embarcado	98
Figura 37. Fluxograma de sincronização entre sistemas microcontrolados	100
Figura 38. Aplicativo de testes dos valores digitais de leme e manche.....	104
Figura 39. Fluxograma dos aplicativos para captura dos dados de leme e manche	105
Figura 40. Sensores resistivos do leme e manche.....	106
Figura 41. Calibração e atuação na embarcação.....	108
Figura 42. Pinagem da bússola eletrônica	110
Figura 43. Rotina de calibração da bússola eletrônica.....	111
Figura 44. Imagem circular usada para validação dos dados da bússola	113
Figura 45. Fluxograma do cálculo matemático para a bússola eletrônica.....	114
Figura 46. Fluxograma da rotina que extrai sentença do sensor GPS.....	116
Figura 47. Memória e endereçamento de dados do microcontrolador	118
Figura 48. Uso de memória na forma padrão e pós-otimização.....	119
Figura 49. Baterias usadas para partida do motor de popa e eletrônica embarcada.	122
Figura 50. Interface da eletrônica com a embarcação	122
Figura 51. Acabamento e infraestrutura para eletrônica embarcada.....	123
Figura 52. Ponto origem, possíveis destinos e rumos da embarcação	125
Figura 53. Problema da convergência da latitude e longitude.....	129
Figura 54. Simulação da trajetória do algoritmo de navegação.....	132
Figura 55. Campo de futebol da USP de São Carlos	133
Figura 56. Equipamentos para simulação do fluxo da telemetria	134
Figura 57. Relatório da telemetria de dados	135
Figura 58. Represa do Lobo (Broa).....	137
Figura 59. Resultados do teste dos enlaces de rádio freqüência.....	138

LISTA DE TABELAS

Tabela 1- Parâmetros da sonda	16
Tabela 2 - Classificação de robôs sob diferentes critérios	25
Tabela 3 - Características gerais do módulo receptor GPS	62
Tabela 4 - Medidas de desempenho	63
Tabela 5 - Exemplo numérico da sentença NMEA escolhida.....	64
Tabela 6 - Principais características da bússola eletrônica.....	67
Tabela 7 - Aspectos gerais dos <i>tranceivers</i> R.F.....	71
Tabela 8 - Mensagens enviadas pelo sistema de gerenciamento.....	101
Tabela 9 - Pinagem do conversor A/D	103
Tabela 10 - Valores digitais das posições de manche e leme.....	107
Tabela 11 - Endereço e identificação de cada caractere do sensor <i>GPS</i>	120
Tabela 12 - 1ª alternativa de decisão de giro para a embarcação	127
Tabela 13 - 2ª alternativa de decisão de giro para a embarcação	128
Tabela 14 - Incremento e decremento das coordenadas sobre eixos de referência	128
Tabela 15 - Primeira aproximação do algoritmo de navegação	129
Tabela 16 - Segunda aproximação ao ponto destino	130
Tabela 17 - Exemplo numérico da navegação autônoma	131

SUMÁRIO

1. INTRODUÇÃO	13
1.1 Objetivos do trabalho de mestrado	19
1.2 Organização do texto.....	21
2. REVISÃO DE LITERATURA	23
2.1 Visão geral sobre robôs móveis autônomos	23
2.1.1 Classificações e percepções.....	25
2.1.2 Arquiteturas de controle	27
2.1.2.1 Arquiteturas hierárquicas.....	31
2.1.2.2 Arquiteturas heterárquicas.....	31
2.1.2.3 Arquiteturas <i>subsumption</i>	32
2.1.2.4 Arquiteturas híbridas.....	33
2.2 Sistema de georeferenciamento por <i>GPS</i>	33
2.2.1 A rádio navegação	33
2.2.2 Tecnologia <i>GPS</i>	34
2.2.3 Fatores que afetam a precisão do sistema	36
2.2.4 Aplicações do <i>GPS</i> em veículos autônomos.....	38
2.3 Transmissão por rádio frequência	38
2.3.1 Princípios fundamentais	38
2.3.2 O espectro eletromagnético	39
2.3.3 Tecnologia <i>spread spectrum</i>	42
2.4 Antenas – irradiações e ganhos	45
2.5 Regulamentação.....	47
2.6 Sistemas de navegação.....	48
2.7 Conclusões	50
3. PROJETOS CORRELATOS	51
3.1 Caracterização do Sensor <i>GPS</i> e sua aplicação em um barco autônomo	51
3.2 Projeto AURORA	54
3.3 Projeto ARARA	56
3.4 Conclusões	59
4. SENSORES E MÓDULOS R.F. DA TELEMETRIA.....	61
4.1 Sensores embarcados para navegação	61
4.1.1 Módulo receptor <i>GPS</i>	62
4.1.1.2 Interface.....	64
4.1.1.3 Encapsulamento	65
4.1.2 Bússola eletrônica.....	66
4.1.2.1 Interface.....	67
4.1.2.2 Acabamento e proteção.....	68
4.2 Equipamentos para comunicação por R.F.....	69
4.2.1 Enlace da telemetria de dados.....	70
4.2.1.1 Proteção e acabamento.....	73
4.2.2 Enlace de R.F. multimídia	74
4.2.2.1 Proteção e acabamento.....	75

5. PROJETO E IMPLEMENTAÇÃO	77
5.1 Diagrama em blocos da arquitetura implementada	77
5.2 Aplicativo da estação terrestre	79
5.2.1 Iniciando o aplicativo	80
5.2.2 Entrada de dados e otimização de rota	82
5.2.3 Telemetria de dados	86
5.2.4 Tela principal do aplicativo para monitoração e controle da embarcação ..	88
5.2.5 Piloto virtual	94
5.2.6 Geração de relatórios	95
5.3 Eletrônica embarcada	97
5.3.1 Padrão de comunicação e política de troca de mensagens	98
5.3.2 Comunicação com sensores da embarcação	102
5.3.2.1 Sensores do leme e manche	102
5.3.2.1.1 Acoplamento dos sensores e valores digitais reais	106
5.3.2.1.2 Testes e calibração	107
5.3.2.2 Comunicação com a bússola eletrônica	110
5.3.2.2.1 Teste de validação	112
5.3.2.2.2 Cálculo matemático do giro da bússola	113
5.3.2.3 Comunicação com sensor GPS	115
6. TESTES E RESULTADOS	121
6.1 Infraestrutura da embarcação e eletrônica	121
6.2 Algoritmo de navegação autônoma	124
6.2.1 Simulação numérica da trajetória	131
6.3 Testes da telemetria de dados no campo de futebol	133
6.4 Testes de transmissão na represa do Lobo	136
7. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS	140
REFERÊNCIAS BIBLIOGRÁFICAS	143
ANEXO A – Aplicativo da estação terrestre	148
ANEXO B – Sistema de Gerenciamento	164
ANEXO C – Sistema de Posicionamento	184

1. Introdução

A qualidade da água de rios e reservatórios e particularmente daqueles destinados ao abastecimento público, é de grande importância atual. Considerando-se os múltiplos impactos a que estão sujeitos e à crescente degradação dos mesmos, o monitoramento da qualidade da água em locais de captação como lagos e represas, utilizadas para o uso e consumo humano, vem recebendo atenção especial de grupo de pesquisadores interessados na avaliação e manutenção das propriedades naturais do líquido da vida.

Nos tempos atuais o trabalho de coleta das variáveis físicas e químicas da água, por grupos de pesquisadores, ainda é feita de maneira bastante árdua e até certo ponto com pouco respaldo tecnológico. Os pesquisadores dirigem-se até determinados pontos pré-estabelecidos de coleta e com a ajuda de equipamentos de geoposicionamento estacionam a embarcação e com a instrumentação disponível medem e registram as variáveis hidrológicas a diferentes profundidades. Esse processo é muitas vezes realizado em situações adversas de clima e período (como sob chuva forte e em período noturno). Embora esta metodologia seja satisfatória apresenta como um dos principais inconvenientes a limitação dos pontos de coleta por corpo aquático em função do tempo necessário, a dificuldade de se coletar dados em situações adversas de clima (tempestades) e tempo (à noite), o custo desta mão-de-obra de coleta que é geralmente de alto valor e também o risco de vida dos pesquisadores nestas e outras situações.

O Laboratório de Instrumentação Virtual e Microprocessada (LIV μ P) do Departamento de Engenharia Elétrica da EESC/USP vem desenvolvendo uma plataforma autônoma completa para medição automática das variáveis físicas e

químicas em lagos e reservatórios, com o objetivo de substituir os meios convencionais utilizados para este tipo de missão por um mais eficiente e racional. A automação das medidas deverá permitir o barateamento de todo o custo da operação e principalmente retirar os pesquisadores da tarefa mecânica de capturar as variáveis hidrológicas, permitindo a eles somente a quantificação e qualificação de todos os dados referentes ao processo de coleta. Esse esforço se insere no interesse em desenvolver instrumentação dedicada para monitoração do meio ambiente, particularmente do meio ambiente aquático.

Outros trabalhos desenvolvidos pelo LIV μ P na área de instrumentação eletrônica incluem: um medidor de pH e concentração de Íons específicos utilizando o microprocessador Z80, cujo desenvolvimento foi apoiado pelo programa de instrumentação do PADCT (1), um monitor para transientes da rede elétrica utilizando o microprocessador 8086 (2), um microscópio foto-acústico (3), instrumentação para o estudo do efeito dos transientes de rede em microprocessadores (4), diversas interfaces de aquisição de dados e sensores de uso específico (5,6,7,8), um medidor de pH virtual (9), sonda para monitoração da qualidade da água com o apoio da FAPESP e projetos de iniciação científica para controle (10) e acionamento de cargas (11) da embarcação, através de sistemas microcontrolados, os quais serão discutidos adiante.

A plataforma autônoma já desenvolvida consiste de uma embarcação não tripulada, contendo instrumentação e equipamento de navegação de geoposicionamento comerciais. A embarcação usada é de padrão comercial (figura 1), com casco em alumínio e acionada por motor de popa com potência de 30 HP, movido à gasolina e provido de partida elétrica, além da partida manual convencional. A embarcação ainda possui acessório de direção acionada por um

sistema de redução com caixa de engrenagens e comandos tipo alavanca (manche) para aceleração/desaceleração e mudança de rumo (bombordo/estibordo).



Figura 1. Embarcação utilizada como plataforma autônoma

A embarcação foi adaptada para receber os mecanismos de controle de navegação, através de um sistema de acionamento de deslocamento da direção e de um sistema de acionamento da alavanca de aceleração. Estes sistemas utilizam motores de corrente contínua que são controlados por um conjunto de relês. Os seguintes sensores e atuadores foram incorporados à embarcação:

- Motor de corrente contínua para acionamento da direção/leme
- Motor de corrente contínua para acionamento da aceleração/câmbio
- Acionamento da ignição do motor de partida do motor de popa
- Acionamento de desligamento emergencial do motor de popa
- Sensor resistivo para indicação do rumo da embarcação
- Sensor resistivo para indicação da posição da alavanca de aceleração
- Bússola eletrônica para indicação da orientação da navegação
- Equipamento de geoposicionamento por satélite (*GPS*) comercial

Para a navegação manual um conjunto de botoeiras (figura 2) é utilizado como interface para os motores da embarcação. As botoeiras verdes¹ são destinadas para o controle do motor que movimenta o manche, controlando a aceleração ou

reversão, e as botoeiras amarelas² controlam a movimentação do leme do motor de popa, que está diretamente associado ao volante de direção.



Figura 2 . Sistema de botoeiras

As variáveis físicas e químicas serão medidas utilizando uma sonda vedada à água, interligada a um sistema de registro de dados através de cabo com sistema de lançamento e recolhimento automático, ainda a ser desenvolvido. A sonda e equipamentos de registro medem as seguintes variáveis: pH, temperatura, condutividade, oxigênio dissolvido e turbidez (tabela 1). A profundidade da sonda será monitorada através do comprimento do cabo lançado (até 30m de profundidade).

Tabela 1- Parâmetros da sonda

Parâmetros	Escalas	Resolução
P.H.	0-14	0.01
Temperatura (Celsius)	0-60 ⁰	0.1 ⁰
Condutividade	0 - 100 mS/cm (várias escalas)	0.1 % da escala
Oxigênio (ppm)	0-20	0.1
Turbidez (NTU)	0-800	1

O conjunto das variáveis medidas é hoje utilizado por pesquisadores do Departamento de Ecologia e Biologia Evolutiva (DEBE) da Universidade Federal de São Carlos e do Centro de Recursos Hídricos e Ecologia Aplicada da Escola de

Engenharia de São Carlos, associando estes dados em estudos de ecologia de reservatórios.

O acionamento mecânico e o controle de navegação da embarcação são feitos por dois sistemas microcontrolados desenvolvidos em trabalhos anteriores, sendo denominados por sistema de posicionamento e sistema de gerenciamento.

O sistema de posicionamento automatiza o posicionamento da embarcação, e para tanto, foram integrados vários sensores e dispositivos de acionamento de direção, sentido, aceleração, posicionamento e caracterização da atual condição da embarcação. O geoposicionamento da embarcação é determinado por um equipamento comercial *GPS*, que disponibiliza as coordenadas da localização, ou seja, o valor da latitude e da longitude, além das variáveis de velocidade média, altura e distância percorrida. Associado a este se utiliza como auxílio à navegação uma bússola eletrônica. O sentido de deslocamento, assim como as acelerações da embarcação, são determinadas pelo mesmo manche, ou seja, do ponto morto pode-se movimentá-lo para frente ou para trás, engatando para avançar ou marcha ré, e a partir de qualquer um destes pontos de engate a continuidade do movimento determina o valor da aceleração. O sistema de gerenciamento comunica-se serialmente com outros três sistemas, sendo eles: microcomputador, sistema de posicionamento e sonda de coleta de dados. Todas essas comunicações são feitas no padrão assíncrono bidirecional. O sistema de gerenciamento recebe de um microcomputador portátil, através de sua porta serial, os dados referentes aos pontos de coleta em coordenadas de latitude e longitude. Os pontos de coleta pesquisados são armazenados em uma memória não volátil para posterior descarregamento destes dados no microcomputador portátil ao final da pesquisa.

O sistema de gerenciamento pode ser considerado como o circuito mestre e o sistema de posicionamento como o circuito escravo, pois o sistema de gerenciamento é responsável por definir e, principalmente, acompanhar o processo de andamento da navegação propiciando a chegada ao ponto estipulado no menor tempo possível. Este modelo de arquitetura hierárquico apresenta como vantagem uma estrutura modular e bem definida, sendo possível análises de desempenho e melhor controlabilidade. A grande desvantagem é a falta de flexibilidade, onde qualquer alteração no sistema implica em modificações significativas.

O sistema embarcado desenvolvido até o momento pelo Grupo de Instrumentação Virtual e Microprocessada está mostrado na figura 3, onde seqüencialmente temos como principais componentes: *GPS* comercial, bússola eletrônica, placa de interface com a embarcação, sistema de gerenciamento, placa de aquisição de dados analógicos e sistema de posicionamento.

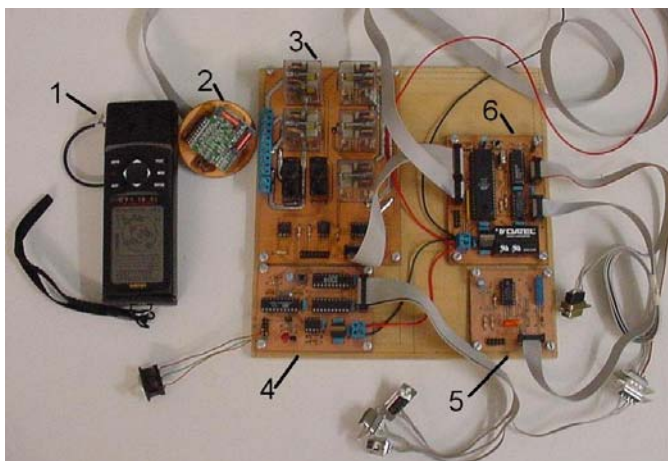


Figura 3. Sistema completo do *hardware* embarcado

1.1 Objetivos do trabalho de mestrado

Este trabalho de mestrado tem como principal objetivo desenvolver um sistema completo de telemetria de dados e imagens entre a plataforma de coleta de dados hidrológicos (embarcação autônoma) e uma estação terrestre, para monitoramento remoto on-line de todos os sensores e atuadores embarcados. Para concretizar esta meta há diversas tarefas de desenvolvimento e um conjunto de restrições de projeto a serem consideradas.

As tarefas essenciais para o sucesso desta implementação incluem a criação de um aplicativo visual para a base terrestre que permite as seguintes funcionalidades: receber todos os pontos de coleta, em coordenadas de latitude e longitude, sendo estes o conjunto de pontos a serem pesquisados pela plataforma autônoma; implementar um algoritmo de otimização de rota que promova uma navegação entre todos os pontos no menor tempo possível; geração de relatórios pós-pesquisa em forma de gráficos e histogramas, para efeitos de comparação entre diferentes pesquisas e validação da telemetria; permitir a tomada de novas decisões de rota sobre a embarcação em qualquer momento da pesquisa em campo, efetuando ajustes ou retornando-a ao ponto inicial de partida por motivo de pane ou desistência da operação (piloto virtual); todas as informações relevantes durante o processo de coleta devem ser visualizadas no aplicativo desenvolvido para a estação terrestre no menor tempo possível.

Do ponto de vista da eletrônica presente na embarcação as principais tarefas de desenvolvimento são: implementar uma arquitetura de controle destinada a robôs móveis autônomos consistente no sistema embarcado; redefinição da programação das rotinas dos microcontroladores já existentes, adequando-as para os novos

sensores, provendo funcionalidade e eficiência (otimização); propiciar o fluxo de informações entre os subsistemas e a estação terrestre, implementando checagem de erro e controle de tempo máximo de resposta para cada sensor; adaptação e calibração mais robusta dos sensores embarcados; implementar um algoritmo de navegação autônoma para testes e simulações do tráfego da telemetria de dados e imagens; proteger todos os equipamentos destinados à plataforma autônoma contra água e poeira, permitindo pesquisas em situações adversas de clima; construção de toda infraestrutura necessária para garantir a segurança de operacionalidade da plataforma.

Para efetuar a comunicação entre a eletrônica embarcada e a estação terrestre tem-se como tarefa o estudo de diversos equipamentos de rádio transmissão, com o objetivo de compatibilizar sua capacidade de transferência de dados digitais com a necessidade de uso para a navegação remota e autônoma (longo alcance de transmissão, sem atrasos e interferências), além de características adicionais como controle de fluxo e erros. Para o enlace multimídia, a ser usado na implementação do piloto virtual, o par emissor/receptor deve ser capaz de gerar uma imagem que permita a operação remota da embarcação.

A implementação do sistema de telemetria de dados e imagens prevê também a máxima reusabilidade dos projetos do Grupo de Instrumentação Microprocessada, além de respeitar todas as restrições inerentes ao projeto: reaproveitamento de todo o *hardware* já produzido, mas com mudanças necessárias e essenciais que visem uma maior funcionalidade e praticidade, como a aplicação de sensores mais adequados ao sistema embarcado sem afetar a estrutura de projeto já criada; utilizar como equipamentos de geoposicionamento a bússola eletrônica e *GPS*,

ambos dedicados; usar equipamentos para a comunicação por ondas de rádio frequência que atendam a todas as normas definidas pela ANATEL .

1.2 Organização do texto

O capítulo 2 tem como objetivo a revisão de literatura dos elementos envolvidos na construção da implementação proposta, apresentando uma visão geral sobre robôs móveis autônomos com suas características, sensores e arquiteturas de controle; sistema de georeferenciamento por *GPS*, mostrando sua tecnologia, fatores de precisão e aplicações em robôs móveis autônomos; a transmissão por rádio frequência, com enfoque voltado principalmente para a explanação da tecnologia de transmissão usada pelo projeto para a transferência do pacote da telemetria de dados até a estação terrestre; fundamentos de irradiação e ganhos em antenas, importante na definição de máxima distância de recepção e finalmente uma breve abordagem sobre diversos sistemas de navegação.

O capítulo 3 apresenta projetos desenvolvidos em outros centros de pesquisa que são correlatos com a implementação proposta nesta dissertação: um projeto de barco autônomo com sistema de navegação baseado nos sensores bússola digital e *GPS*; uma aeronave autônoma com foco em processamento de imagem, com o objetivo de reconhecer problemas em linhas de transmissão, controle de pragas na agricultura, contagem de animais em fazendas, etc. e um dirigível robótico não tripulado para uso em sensoriamento remoto.

O capítulo 4 apresenta as principais especificações dos sensores de georeferenciamento embarcados e dos enlaces de rádio frequência para a telemetria

de dados e imagens. Sobre os sensores *GPS* e bússola eletrônica tem-se as informações do modelo, medidas de desempenho, interface de dados e encapsulamento. Para os equipamentos R.F. há uma explicação detalhada do kit de desenvolvimento usado para a telemetria de dados e todas as características de desempenho do enlace R.F. multimídia.

O capítulo 5 tem como principal objetivo descrever todo o sistema proposto para esta dissertação, apresentando diagrama em blocos do sistema, ilustrações sobre a formatação do pacote de dados, funcionalidades do aplicativo da estação terrestre e fluxogramas detalhando a programação de rotinas de interface com todos os dispositivos embarcados, incluindo testes de validação e desempenho de todos os sensores envolvidos.

O capítulo 6 apresenta um modelo de sistema de navegação autônomo para testes de eficiência da telemetria de dados e imagens como instrumento de diagnóstico; aborda sobre como foi realizado a infraestrutura da embarcação para receber a eletrônica embarcada com os dispositivos de segurança; teste de alcance dos enlaces de rádio frequência; desempenho do aplicativo da estação terrestre em receber um fluxo de pacote de dados e a apresentação dos resultados para o operador do sistema com consistência.

O capítulo 7 finaliza o trabalho apresentando conclusões e sugestões para trabalhos futuros sobre a plataforma autônoma.

2. Revisão de literatura

Neste capítulo são revistos os conceitos mais importantes referentes à fundamentação teórica relevante para o embasamento desta dissertação, contribuindo para a escolha do modelo de arquitetura para o sistema embarcado, sensores, enlaces de rádio-freqüência e sistema de navegação.

Neste contexto, na seção 2.1 é apresentada uma visão geral sobre robôs móveis autônomos, evidenciando suas possíveis classificações, percepções, arquiteturas de controle e algumas referências a robôs autônomos de outros autores; na seção 2.2 há a descrição do modelo de georeferenciamento usando um sensor GPS; a seção 2.3 trata dos princípios fundamentais da transmissão por rádio-freqüência e apresenta uma tecnologia de transmissão digital de sinais que será usada como o canal de dados entre a plataforma autônoma e a estação base; a seção 2.4 exemplifica os modelos existentes de antena, de acordo com sua irradiação e ganho para diferentes propostas de uso; a regulamentação do uso do espectro eletromagnético e as máximas potências admitidas são discutidas na seção 2.5; na seção 2.6 aborda-se os diferentes métodos utilizados para aplicação de um sistema de navegação embarcado e finalmente na seção 2.7 uma breve conclusão sobre os assuntos abordados.

2.1 Visão geral sobre robôs móveis autônomos

A robótica é uma área do conhecimento que tem evoluído de forma muito rápida nos últimos anos, contudo o projeto e a implementação de robôs e autômatos

desenvolvem-se há várias décadas. Na verdade, é difícil estabelecer a data precisa do surgimento dos primeiros autômatos (ancestrais dos atuais robôs), sendo que o termo “robot”, segundo DUDEK, G. e JENKIN, M. (12), foi empregado pela primeira vez por Karel Capek. Os primeiros robôs eram, na verdade, autômatos complexos, verdadeiras obras de arte mecânicas que executavam tarefas de modo repetitivo. Estes robôs deram origem aos atuais braços manipuladores de base fixa, que atualmente são largamente adotados na indústria automotiva. Somente mais recentemente surgiram os robôs móveis, que se caracterizam pela sua capacidade de se deslocar de modo guiado, semi-autônomo ou totalmente autônomo. Apesar de existirem referências mais antigas a dispositivos móveis comandados à distância, podemos considerar que sua origem foi fortemente influenciada pelo desenvolvimento de artefatos militares, como por exemplo, veículos autônomos teleguiados ou mesmo autoguiados como as bombas V1e V2 desenvolvidas pela Alemanha durante a 2ª. Guerra Mundial.

O desenvolvimento da robótica evoluiu portanto dos autômatos, passando pelos robôs manipuladores de base fixa, pelos dispositivos móveis guiados a distância e chegando mais recentemente aos robôs móveis semi-autônomos e mesmo os totalmente autônomos. Um exemplo de sucesso do desenvolvimento de veículos móveis de alta tecnologia é o rover Sojourner que explorou o planeta Marte em 1997. Entretanto, cabe destacar que mesmo os robôs de exploração espacial como o Sojourner, e seus sucessores Spirit e Opportunity, são dispositivos de autonomia limitada, necessitando ainda de comandos enviados a distância por seres humanos que controlam a missão.

2.1.1 Classificações e percepções

Em meio a tantos tipos distintos de robôs e veículos com diferentes recursos e níveis de autonomia, é importante que se busque identificar e organizá-los em categorias. A tabela 2 apresenta uma classificação dos robôs de acordo com suas funções, mobilidade e autonomia:

Tabela 2 - Classificação de robôs sob diferentes critérios

Funcionalidade: Tipos de Aplicações	<ul style="list-style-type: none"> • Mecanismos eletro-mecânicos com movimento repetitivo pré-definido e fixo (autômatos) • Manipuladores: Braços de base fixa (robôs industriais) • Manipuladores: Braços de base móvel • Robôs Móveis (ver abaixo os tipos de mobilidade): <ul style="list-style-type: none"> ✓ Terrestres: <i>indoor</i>, <i>outdoor off-road</i>, veículos convencionais <i>outdoor</i> (automóveis) ✓ Aquáticos, Submarinos e Aéreos ✓ Exploração Espacial
Mobilidade: Dispositivos de Locomoção	<ul style="list-style-type: none"> • Robôs móveis com deslocamento com rodas • Robôs móveis com deslocamento com esteiras • Robôs com deslocamento por propulsão • Robôs com deslocamento por pernas
Autonomia: Grau de Inteligência do Robô	<ul style="list-style-type: none"> • Totalmente comandados a distância (e.g. carro de “controle remoto” = tele-comandado, tele-operado) • Veículos guiados: dependem de informações externas, sem as quais não podem operar (AGV – <i>Automated Guided Vehicle</i>) • Veículos semi-autônomos: operam controlados remotamente, mas possuindo um certo grau de autonomia (e.g. podem parar de modo a evitar o choque com obstáculos) • Veículos autônomos: operam sem intervenção humana, de forma totalmente automatizada. Uma vez iniciada sua operação, executam tarefas de modo autônomo

Todos os robôs comunicam-se com o ambiente aonde atuam através de suas percepções. A percepção é um dos componentes principais dos robôs móveis, pois é através do conjunto de sensores que pode ser garantido ao sistema um maior nível de autonomia e robustez, interagindo diretamente com o ambiente real. Existem

sensores para as mais diversas aplicações, sendo alguns com respostas simples e outros que envolvem calibrações e interpretações de resultados:

- **Encoder:** Mede o número de rotações das rodas, obtendo assim informações de odometria (medida de deslocamento).
- **Sensor de Infravermelho:** Mede a distância entre o sensor e um obstáculo posicionado em frente a ele através da estimativa da distância calculada em função do retorno da reflexão da luz na superfície.
- **Sensor à Laser:** Mede a distância entre o sensor e um obstáculo posicionado em frente – sensor direcional de alta precisão.
- **Sonar (ultra-som):** Estima a distância entre o sensor e os obstáculos ao seu redor através da reflexão sonora. É um sensor mais sensível a perturbações, obtendo medidas aproximadas.
- **Sensor de Contato:** Identifica quando ocorre uma colisão entre o veículo (sensor) e um ponto de contato (medida de contato/pressão).
- **Bússola Eletrônica:** Identifica a orientação do veículo em relação ao campo magnético da terra (medida de orientação – posição relativa).
- **GPS:** Identifica a posição absoluta do veículo no globo terrestre, baseando-se na rede de satélites GPS (medida de posicionamento absoluto).
- **Visão Artificial (imagens):** O uso de imagens permite que sejam adquiridas a partir de uma ou mais câmeras (visão unidimensional, estéreo ou omnidirecional) descrições do ambiente (imagem monocromática ou colorida). Isto permite que se implemente técnicas de determinação de posicionamento relativo, posicionamento absoluto, detecção e estimativa de deslocamento, detecção de obstáculos, eliminação de vibrações, etc.

- **Potenciômetros:** Usados para se medição de posição linear ou angular de sistemas mecânicos como volante, manche e leme de um veículo aquático (sinal analógico).
- **Acelerômetros:** usados para medir aceleração de objetos. Através dele pode-se obter velocidade, cálculo de distância, e com a integração de um conjunto de valores obter até as coordenadas do percurso realizado pelo objeto.
- **Giroscópios:** dispositivo usado para orientação em navios, aviões e aeronaves espaciais.

O adequado uso dos sensores, bem como a correta interpretação dos dados fornecidos por estes (modelo sensorial, incluindo precisão, distância alcançada, suscetibilidade ao ruído, etc), é que permitirá a correta implementação e/ou simulação de sistemas robôs móveis autônomos (RMA's) mais robustos. Um outro tema de grande relevância na robótica é a integração de sensores (fusão sensorial), explorando a complementaridade entre eles, já que possuem diferentes escalas de sensibilidade e confiabilidade (e.g. fusão de dados de um radar, sonar e sensor laser).

2.1.2 Arquiteturas de controle

De forma simplificada, o que proporciona autonomia aos chamados robôs móveis autônomos é o sistema de *hardware* e aplicativo responsável pela execução, controle e coordenação em tempo real das diversas funções que colaboram para o alcance dos objetivos da missão. Este sistema é o que será aqui denominado por arquitetura de controle. Tomando-se como base estudos recentes a respeito do

assunto, particularmente os publicados em (13), (14) e (15) podem-se citar como características desejáveis em uma boa arquitetura de controle destinada a robôs móveis autônomos:

➤ **Extensibilidade:** a inclusão de novos sensores, atuadores e módulos funcionais deve ser facilitada, sem que sejam necessárias grandes modificações da estrutura já existente. E ainda os novos sistemas devem ter a capacidade de interagir com os componentes pré-existentes.

➤ **Generalidade:** os componentes de arquitetura devem ser suficientemente genéricos para que possam ser intercambiadas entre diferentes veículos, ou substituídos por componentes equivalentes sem modificações significativas. Esta característica é desejável tanto para o *hardware* quanto para o aplicativo da arquitetura.

➤ **Previsibilidade:** o comportamento como um todo do sistema deve ser facilmente previsível com base nos objetivos da missão e nos dados fornecidos pelo sistema sensorial. A conclusão é que o sistema não deve se comportar de forma aleatória.

➤ **Modelo do ambiente:** em aplicações de veículos autônomos esses modelos codificam as estruturas do ambiente, permitindo determinar a posição absoluta do veículo e por conseguinte planejar trajetórias coerentes.

➤ **Reatividade:** ambientes operacionais reais são dinâmicos. Um veículo autônomo deve ser reativo tanto para situações planejadas como para as não previstas no modelo inicial.

➤ **Deteção e recuperação de erros:** erros podem aparecer durante a execução de uma missão, especialmente em uma missão de longo alcance. A arquitetura deve prover um mecanismo de deteção de erros e processos de

recuperação definidos pelo usuário que evitem o cancelamento da missão. Quando a recuperação é impossível, os módulos defeituosos devem ser indicados como danificados de forma a evitar que sejam utilizados em missões subsequentes.

➤ **Tolerância à falhas:** sempre que o nível de redundância dos componentes do veículo é suficiente, qualquer mecanismo de reposição dos componentes falhos por componentes equivalentes é desejável. Para missões de longo alcance a tolerância à falhas é de grande importância.

➤ **Programabilidade:** o usuário deve necessitar de ferramentas para especificação e verificação de missões. Enquanto linguagens de especificação de missões são adequados para desenvolvedores, interfaces gráficas são mais adequadas para usuários finais.

Ao avanço da criação de robôs móveis autônomos um grande número de arquiteturas de controle foram implementadas, tanto em veículos terrestres, aéreos e aquáticos. Em todos os casos há características comuns, o que possibilita a classificação dos mesmos segundo diferentes critérios como a distribuição da abstração ao longo das unidades de controle, a presença ou não de modelos e representações de ambiente, etc. Tomando-se por base a presença ou não de conhecimento sobre o modelo do ambiente operacional do veículo, as arquiteturas são subdivididas em:

➤ **Arquiteturas deliberativas:** são baseadas principalmente em um processo de planejamento e em um modelo de execução da missão, sendo especificada como um conjunto de objetivos a serem alcançados. Através da modelagem do ambiente a missão é planejada e um sistema de controle executa o plano de ação obtido. Arquiteturas desenvolvidas segundo esta abordagem geralmente apresentam

comportamentos previsíveis, porém não se adaptam com facilidade a ambientes muito dinâmicos, nos quais as mudanças ocorrem rapidamente.

➤ **Arquiteturas comportamentais:** também conhecidas como arquiteturas reativas, são divididas nos diversos comportamentos que se deseja para o veículo. Normalmente, uma missão é descrita como uma seqüência de fases com um conjunto de comportamentos ativos. Os comportamentos reagem continuamente aos estímulos detectados pelo sistema de percepção. O comportamento global do veículo surge da combinação dos comportamentos elementares ativos. Esta arquitetura é típica de pesquisas onde há somente escassas informações sobre o ambiente a ser investigado.

➤ **Arquiteturas híbridas:** estas arquiteturas são as que possuem o maior número de implementações. Elas unem características das arquiteturas deliberativas e comportamentais, com elementos determinísticos sendo utilizados para proporcionar ao sistema comportamentos previsíveis e com elementos reativos sendo utilizados para se prover ações de resposta mais rápidas em situações que o sistema não é capaz de prever.

A classificação acima, apesar de correta, não leva em conta o detalhamento das implementações de suas arquiteturas de controle (estruturação de projeto). Analisando a arquitetura em relação aos módulos construídos surgem as categorias de arquitetura hierárquica, heterárquica, *subsumption* e híbridas. Os detalhes de cada abordagem serão apresentadas nas subseções posteriores.

2.1.2.1 Arquiteturas hierárquicas

Utilizam uma abordagem hierárquica para dividir o sistema em níveis. Os níveis superiores são responsáveis pelos objetivos principais da missão, como a navegação e os cálculos mais complexos, enquanto os níveis inferiores solucionam problemas específicos da missão, como capturar os dados dos sensores. Esta arquitetura é sempre serial, onde a comunicação entre níveis só pode ser feita por módulos adjacentes. Níveis superiores delegam comandos para os inferiores, que como resposta recebem informações de retorno. O fluxo de informação decresce da base para o topo da hierarquia. Como vantagem apresentam uma estrutura modular e bem definida, sendo possível análises de desempenho e melhor controlabilidade. A grande desvantagem é a falta de flexibilidade, onde qualquer alteração no sistema implica em modificações significativas.

Para exemplificar este tipo de abordagem temos em (16) um veículo aquático submarino com a implementação de 3 camadas hierárquicas. A camada inferior (*Servo Level*) atua sobre os motores através de malha de controle clássica. A camada intermediária (*Task Level*) é responsável pela codificação e sequenciamento das tarefas, e a camada superior (*Organization Level*) proporciona a interface com o usuário por telas gráficas.

2.1.2.2 Arquiteturas heterárquicas

Ao contrário da arquitetura anterior esta utiliza uma estrutura paralela onde os módulos do sistema comunicam-se diretamente, sem níveis intermediários ou

superiores. A grande vantagem é sua flexibilidade, pois os sensores podem ser acessados por quaisquer componentes do sistema, permitindo aplicações de processamento paralelo. A grande desvantagem é a falta de supervisão, dificultando a controlabilidade e aumentando o tráfego de informações nos barramentos. Em (17) há um exemplo dessa arquitetura que foi modelada como uma comunidade de agentes, que são entidades autônomas que coordenam suas ações através de protocolos de comunicação.

2.1.2.3 Arquiteturas *subsumption*

A arquitetura *subsumption* (controle por níveis) idealizada por BROOKS (18) é realizada por procedimentos paralelizáveis sem um nível superior no comando. Procedimentos são níveis da arquitetura de controle que são ativados por sensores executando ações. Neste esquema os níveis possuem o conceito de prioridade, sendo que um nível de alta prioridade pode suprimir por um longo tempo um nível de baixa prioridade. Para evitar este tipo de problema quando um nível de alta prioridade é ativado por um período curto de tempo, o de menor prioridade reassume o controle. Neste tipo de arquitetura não há uma estrutura de dados global, e isto representa um comportamento dinâmico real, onde cada subsistema tem seus próprios dados e controles. A desvantagem dessa arquitetura é a dificuldade de sincronismo entre níveis e a dificuldade de verificação do sistema, pela falta de um sistema supervisor.

2.1.2.4 Arquiteturas híbridas

Como o próprio nome já diz é uma combinação de todas as arquiteturas anteriores relatadas. O sistema é dividido em níveis (superior e inferior), os quais utilizam diferentes níveis de abstração. O nível superior adota uma arquitetura hierárquica para implementar estratégias. O nível inferior mescla as arquiteturas heterárquicas e *subsumption* para controle dos subsistemas de *hardware*. Na arquitetura *subsumption*, comandos do nível superior são traduzidos em procedimentos correspondentes que são ativados. Já na arquitetura hierárquica o nível inferior consiste de vários módulos operando concomitantemente. Uma situação crítica pode ativar a supervisão superior para assumir o controle. Como vantagem o a flexibilidade dos níveis inferiores é alcançada, em detrimento do aumento da dificuldade da verificação formal do sistema como um todo.

2.2 Sistema de georeferenciamento por *GPS*

2.2.1 A rádio navegação

O uso de sinais de rádio para determinar o geoposicionamento de objetos foi um avanço significativo na navegação. O equipamento para rádio-navegação apareceu em 1912, não era muito preciso, mas funcionou até que a II Grande Guerra permitisse o desenvolvimento do *RADAR* (19) e a capacidade de medir intervalos de tempo entre emissão e recepção de ondas de rádio. Para determinar a

posição, medem-se os intervalos de tempo dos sinais provenientes de locais conhecidos. Os sinais de rádio são emitidos de transmissores exatamente ao mesmo tempo e têm a mesma velocidade de propagação. Um receptor localizado entre os transmissores detecta qual sinal chega primeiro e o tempo até a chegada do segundo sinal. Se o operador conhece as exatas localizações dos transmissores, a velocidade das ondas de rádio e o intervalo de tempo entre os dois sinais, pode-se calcular sua localização em uma dimensão. Se forem usados três transmissores, obtém-se uma posição bi-dimensional, em latitude e longitude. O sistema de posicionamento por satélites (*GPS*) funciona baseado em princípios semelhantes. Os transmissores de rádio são substituídos por satélites que orbitam a Terra a grande altitude (da ordem de 20.200km de altura) e permitem conhecer a posição em três dimensões: latitude, longitude e altitude.

2.2.2 Tecnologia GPS

A tecnologia atual permite que qualquer pessoa possa se localizar no planeta com uma precisão consideravelmente grande. O sofisticado sistema que tornou isso realidade é o *GPS*, concebido pelo departamento de defesa americano no início da década de 1960. O sistema foi declarado totalmente operacional apenas em 1995, consistindo de 24 satélites que orbitam a Terra a 20.200 km, duas vezes por dia, e emitem simultaneamente sinais de rádio codificados. Cada satélite emite um sinal que contém: código de precisão, código geral e informação de status.

Como outros sistemas de rádio-navegação, todos os satélites enviam seus sinais de rádio exatamente ao mesmo tempo, permitindo ao receptor avaliar o

intervalo entre emissão/recepção. A potência de transmissão é de apenas 50W. A hora-padrão *GPS* é passada para o receptor do usuário. Receptores *GPS* em qualquer parte do mundo mostrarão a mesma hora, inclusive os milésimos de segundos. A hora-padrão é altamente precisa, porque cada satélite contém um relógio atômico, com precisão de nano segundos, sendo mais preciso que a própria rotação da Terra.

No sistema *GPS* o receptor tem que reconhecer as localizações dos satélites. Uma lista de posições, conhecida como almanaque, é transmitida de cada satélite para os receptores. Controles em terra rastreiam os satélites e mantêm seus almanaques atualizados. Cada satélite tem códigos de precisão e códigos gerais únicos, e o receptor pode distinguí-los.

Os receptores civis medem os intervalos de tempo entre a recepção dos sinais dos códigos gerais. O conceito da rádio-navegação depende inteiramente da transmissão simultânea de rádio-sinais. O controle de terra pode interferir, fazendo com que alguns satélites enviem seus códigos ligeiramente antes ou depois dos outros. A interferência deliberada introduzida pelo departamento de defesa americano é a fonte da disponibilidade seletiva, que inserem ruídos que afetam diretamente a precisão do sensor.

Os sinais gerados pelos satélites contêm um "código de identidade" (ou pseudo-aleatório), dados efêmeros (de status) e dados do almanaque. O código de identidade *PRN* (*Pseudo-Random Code*) identifica qual satélite está transmitindo. Usa-se como referência dos satélites seus *PRNs* (1 a 32). O código pseudo-aleatório permite que todos os satélites do sistema compartilhem a mesma frequência sem interferências. É um sistema engenhoso que torna o *GPS* prático e relativamente barato de se usar. Ao contrário dos satélites de TV, que estão em órbitas geo-

síncronas (estacionários no céu) e transmitem poderosos sinais para refletores parabólicos em terra, o satélite GPS envia sinais com poucas informações e de baixa potência para antenas de tamanho muito reduzido. De fato, os sinais *GPS* são tão fracos que não são maiores que o ruído de fundo (de rádio) inerente ao planeta Terra. A extração da informação do código pseudo-aleatório se baseia em uma comparação realizada em ciclos de sinais, podendo ser alterado (apenas o código de precisão) e permitindo que o governo americano controle o acesso ao sistema do satélite.

Os dados efêmeros (de status) são constantemente transmitidos e contém informações de status do satélite (operacional ou não), hora, dia, mês e ano. Os dados de almanaque dizem ao receptor onde procurar cada satélite a qualquer momento do dia. Com um mínimo de três satélites, o receptor pode determinar uma posição latitude/longitude – que é chamada posição fixa 2D (bi-dimensional). Com a recepção de quatro ou mais satélites, um receptor pode determinar uma posição 3D, isto é, latitude, longitude e altitude. Pelo processamento contínuo de sua posição, um receptor pode também determinar velocidade instantânea e direção do deslocamento (bússola digital).

2.2.3 Fatores que afetam a precisão do sistema

Um fator que afeta a precisão é a geometria dos satélites - localização dos satélites em relação uns aos outros - sob a perspectiva do receptor *GPS*. Se um receptor *GPS* estiver localizado sobre 4 satélites e todos estiverem na mesma região do céu, sua geometria é pobre. Na verdade, o receptor pode não ser capaz de se

localizar, pois toda a medida de distância provém da mesma direção geral. Isto significa que a triangulação é pobre e a área comum da intersecção das medidas é muito grande (isto é, a área onde o receptor busca sua posição cobre um grande espaço). Dessa forma, mesmo que o receptor mostre uma posição, a precisão não é boa. Com os mesmos 4 satélites, se espalhados em todas as direções, a precisão melhora drasticamente. Tomemos como exemplo 4 satélites separados em intervalos de 90° a norte, sul, leste e oeste. A geometria é ótima, pois a medida provém de várias direções. A área comum de intersecção é muito menor e a precisão muito maior. A geometria dos satélites torna-se importante quando se usa o receptor *GPS* próximo a edifícios ou em áreas montanhosas ou vales. Quando algum satélite é bloqueado, a posição relativa dos demais determinará a precisão, ou mesmo se a posição pode ser obtida. Um receptor de qualidade indica não apenas os satélites disponíveis, mas também onde estão no céu (azimute e elevação), permitindo ao operador saber se o sinal de um determinado satélite está sendo obstruído.

Outra fonte de erro é a interferência resultante da reflexão do sinal em algum objeto, a mesma que causa a imagem “fantasma” na televisão. Como o sinal leva mais tempo, o receptor acredita que o satélite está mais longe do que a realidade. Outras fontes de erro que podem ser associadas são o atraso na propagação dos sinais devido aos efeitos atmosféricos e alterações do relógio interno. Em ambos os casos, o receptor *GPS* é projetado para compensar os efeitos.

2.2.4 Aplicações do *GPS* em veículos autônomos

O sensor *GPS* tornou possível a implementação da navegação georeferenciada para veículos móveis autônomos. Antes de sua descoberta a referência ao ambiente era pré-determinada pelo sistema ou adquirida ao longo da pesquisa, mapeada por diversos sensores. Com a evolução da tecnologia e o advento do *GPS* a agricultura de precisão agora tem à disposição máquinas agrícolas de grande porte que trabalham sem nenhuma supervisão humana; aeronaves são usadas para tirar fotos digitais de plantações para posterior análise de produtividade, controle de pragas e até aplicação de pesticidas; dirigíveis autônomos fazem inspeção de redes elétricas e desmatamentos na Amazônia; veículos autônomos aquáticos navegam por rios e represas analisando a qualidade da água ingerida por populações. No capítulo Projetos correlatos abordam-se sistemas que utilizam o sensor *GPS* para efetuar a navegação autônoma.

2.3 Transmissão por rádio frequência

2.3.1 Princípios fundamentais

A transmissão sem fio para comunicação digital entre uma rede de computadores teve início no Havaí (20), onde usuários ficavam separados por ilhas e o sistema de telefonia era inadequado. O primeiro sistema de computador a utilizar

rádios ao invés de ligações físicas ponto-a-ponto foi o sistema Aloha, da Universidade do Havaí, em 1971.

As ondas de rádio são relativamente fáceis de gerar, podem percorrer longas distâncias e são largamente utilizadas na comunicação dentro de interiores e em espaço aberto. As ondas de rádio são omnidirecionais, ou seja, trafegam em todas as direções a partir da fonte, sendo que o transmissor e o receptor não precisam estar devidamente alinhados fisicamente para ocorrer uma comunicação.

As propriedades das ondas de rádio dependem de sua frequência. Em frequências baixas as ondas de rádio atravessam bem os obstáculos, mas sua potência decai rapidamente com o aumento da distância entre o transmissor e o receptor. Em frequências altas as ondas de rádio tendem a percorrer linhas retas, são refletidas por obstáculos e são absorvidas pela chuva. Em todas as faixas de frequências as ondas de rádio estão sujeitas a interferências de motores e equipamentos elétricos, além do problema do uso da mesma faixa de frequências por diferentes usuários. Por isso há a necessidade de pedido de licença do governo no uso de transmissores de rádio em determinadas faixas de frequência.

2.3.2 O espectro eletromagnético

Ondas eletromagnéticas que podem propagar-se pelo espaço (mesmo no vácuo). Essas ondas foram descobertas pelo físico britânico James Clerk Maxwell em 1865 e produzidas pelo físico alemão Heinrich Hertz em 1887. O número de oscilações por segundo de uma onda eletromagnética é chamado de frequência (f) e é medido em Hz (em homenagem a Hertz). A distância entre dois pontos máximos

ou mínimos consecutivos da onda é chamado comprimento de onda, expresso pela letra grega λ (lambda).

No vácuo, todas as ondas eletromagnéticas se propagam na velocidade da luz, independentemente da sua frequência, e seu valor aproximado é 3×10^8 m/s. No cobre ou na fibra óptica, esta velocidade decresce aproximadamente para 2/3 desse valor e se torna ligeiramente dependente da frequência, como descrito em (21).

A relação fundamental entre f , λ e c (velocidade da luz) no vácuo é:

$$\lambda \cdot f = c \tag{1}$$

Considerando a velocidade da luz constante, e se uma variável é conhecida, então podemos desenvolver a fórmula e chegar a um resultado real. Por exemplo, ondas de 1MHz têm aproximadamente 300m de comprimento e ondas de 1cm têm uma frequência de 30GHz.

As ondas de rádio, microondas, infravermelho e porções de luz visíveis do espectro podem ser utilizadas para a transmissão de informações através da modulação da amplitude, frequência ou fase das ondas (figura 4). A radiação ultravioleta, os raios-X e os raios gama poderiam ser bons propagadores de ondas devido às suas altas frequências, mas são difíceis de produzir, modular, não se propagam bem através de construções e são nocivos para os seres vivos (22).

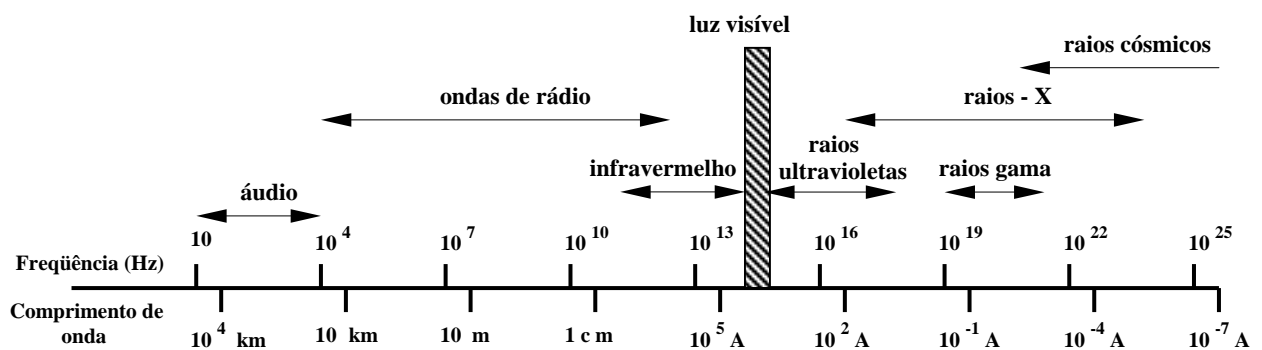


Figura 4. Espectro eletromagnético

O volume de informação que uma onda eletromagnética pode transportar está diretamente relacionado com a sua largura de banda, ou seja, quanto maior a frequência maior a capacidade de transmissão de *bits* por *Hertz*. A descoberta das ondas eletromagnéticas no uso em sistemas de comunicação revolucionou a área de transmissão de dados, permitindo a comunicação entre áreas distantes com um grande alcance e com menores perdas (figura 5) em relação às linhas físicas.

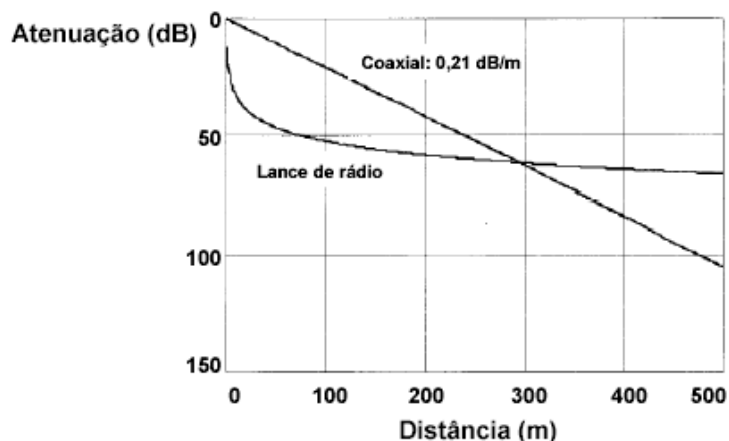


Figura 5. Comparação de perdas entre uma linha coaxial e a atenuação no espaço livre utilizando a frequência de 100MHz

O uso do espectro eletromagnético é regulamentado por organizações nacionais e internacionais. Nos Estados Unidos, o *FCC (Federal Communications Commission)* aloca espectro para rádios AM e FM, televisão, telefonia celular, operadoras de telefonia, polícia, militares, navegação, governo e diversos outros usuários. No Brasil, essa responsabilidade é delegada a ANATEL (23). Em âmbito mundial, uma agência do *ITU-R (Radiocommunication Sector)* é responsável por essa regulamentação: a *WRC (World Radiocommunications Conference)*. Muitas vezes as agências nacionais não seguem as recomendações da *WRC*. Conseqüentemente, dispositivos que utilizam rádio frequência podem funcionar apenas em um certo país ou região.

2.3.3 Tecnologia *spread spectrum*

Spread Spectrum é uma técnica de codificação para a transmissão digital de sinais, originalmente desenvolvida pelos militares durante a segunda guerra mundial, com o objetivo de transformar as informações a serem transmitidas num sinal parecido com um ruído radioelétrico evitando assim a monitoração pelas forças inimigas. A técnica consiste em codificar e modificar o sinal de informação executando o seu espalhamento no espectro de frequências. O sinal espalhado ocupa uma banda maior que a informação original, porém possui baixa densidade de potência, apresentando uma baixa relação sinal/ruído. Para os receptores convencionais esta comunicação pode até ser imperceptível.

Os rádios *spread spectrum* utilizam as faixas de frequências livres adotadas por vários países, inclusive o Brasil, denominadas internacionalmente como bandas *ISM* (*Instrumentation, Scientific & Medical*) definidas nas faixas de 900MHz, 2,4GHz e 5,8GHz. A técnica de *spread spectrum* é implementada através dos seguintes processos: Salto de Frequência (*Frequency Hopping*), Seqüência Direta (*Direct Sequence*) ou então uma combinação dos dois processos (Sistema Híbrido).

Na técnica de *spread spectrum* empregando a tecnologia por saltos de frequência, a informação transmitida troca de um canal para outro numa seqüência pseudo-aleatória. Esta seqüência é determinada por um circuito gerador de códigos pseudo-aleatórios que trabalham num padrão pré-estabelecido.

O receptor por sua vez deve estar sincronizado com o transmissor, ou seja, deve saber previamente a seqüência de canais onde o transmissor vai saltar para poder sintonizar estes canais e receber os pacotes transmitidos.

As vantagens da utilização desta técnica de transmissão são:

- ✓ Os canais que o sistema utiliza para operação não precisam ser seqüenciais.
- ✓ A probabilidade de diferentes usuários utilizarem a mesma seqüência de canais é muito pequena.
- ✓ A realização de sincronismo entre diferentes estações é facilitada em razão das diferentes seqüências de salto.
- ✓ Maior imunidade às interferências.
- ✓ Equipamentos de menor custo.

As desvantagens da utilização desta técnica de transmissão são:

- ✓ Ocupação maior do espectro em razão da utilização de diversos canais ao longo da banda.
- ✓ O circuito gerador de freqüências (sintetizador) possui grande complexidade.
- ✓ O sincronismo entre a transmissão e a recepção é mais crítico.
- ✓ Baixa capacidade de transmissão, da ordem de 2 Mbit/s.

Empregando a tecnologia de Seqüência Direta, o sinal de informação é multiplicado por um sinal codificador com característica pseudo-aleatória, conhecido como *chip sequence* ou pseudo-ruído. O sinal codificador é um sinal binário gerado numa freqüência muito maior do que a taxa do sinal de informação. Ele é usado para modular a sinal da portadora expandindo a largura da banda do sinal de rádio freqüência transmitida.

No receptor o sinal de informação é recuperado através de um processo complementar usando um gerador de código local similar e sincronizado com o código gerado na transmissão.

Em razão da utilização de uma grande largura de banda para transmissão, os sistemas em seqüência direta dispõem de poucos canais dentro da banda. Estes canais são totalmente separados de forma a não gerar interferência entre eles,

sendo este princípio utilizado pelo *CDMA* (*Code Division Multiple Access*) na telefonia celular atual.

As vantagens desta técnica de transmissão são:

- ✓ O circuito gerador de frequência (sintetizador) é mais simples, pois não tem necessidade de trocar de frequência constantemente.
- ✓ O processo de espalhamento é simples, pois é realizado através da multiplicação do sinal de informação por um código.
- ✓ Maior capacidade de transmissão, da ordem de 11 Mbit/s.

As desvantagens desta técnica de transmissão são:

- ✓ Maior dificuldade para manter o sincronismo entre o sinal *PN-code* gerado e o sinal recebido.
- ✓ Maior dificuldade para solução dos problemas de interferências.
- ✓ Equipamentos de maior custo.

Finalizando os processos de transmissão dessa tecnologia temos os sistemas híbridos, que combinam as duas técnicas de modulação: saltos em frequência e seqüência direta. Esses sistemas utilizam, alternadamente, uma técnica de cada vez, mantendo a outra inoperante.

A norma ANATEL define que quando um sistema está operando com salto em frequência, cada salto deve ter um tempo médio de ocupação em qualquer frequência que não exceda 0,4 segundos em um período de tempo igual ao número de frequências de salto utilizada multiplicada por 0,4. Quando o sistema estiver operando em seqüência direta o sistema deve ter densidade média de potência de no máximo 8 dBm em qualquer faixa de 3 KHz, medidas em um intervalo de 1 segundo.

2.4 Antenas – irradiações e ganhos

A antena pode ser considerada o componente mais crítico em um sistema de transmissão por ondas de rádio. Ela é responsável por transferir o sinal gerado pelo transmissor para o espaço e, portanto, deve estar perfeitamente calibrada para a frequência de trabalho (24). Existem inúmeros modelos de antenas com suas características peculiares e que devem ser levadas em conta quando se pretende montar um sistema irradiante, sendo a polarização uma delas. Os sinais de rádio podem ser polarizados verticalmente ou horizontalmente. Grande parte dos serviços de comunicações usam a polarização vertical. A polarização horizontal é usada em alguns serviços de comunicação e também pelas emissoras de televisão. Existe ainda a polarização circular que é adotada pelas emissoras de rádio FM. A vantagem deste tipo de polarização é que a antena do rádio pode estar tanto na vertical quanto na horizontal que não ocorrerão perdas na recepção. Se uma estação transmite um sinal polarizado verticalmente, a antena receptora também deve estar posicionada desta forma, caso contrário haverá uma perda em torno de 20dB.

O ganho de uma antena é umas de suas principais características. As antenas plano-terra com $1/4$ de comprimento de onda tem ganho de 0dB. Já antenas com comprimento de onda $5/8$ tem rendimento bem maior (3dB), sendo mais indicadas para maiores coberturas. Rádio amadores usam esse tipo de antena em suas estações, o que proporciona uma grande melhoria nas comunicações. Por fim uma outra característica das antenas de fundamental importância na área de cobertura proporcionada por um sistema irradiante é o tipo de irradiação (25). Existem três

tipos: omnidirecional (não direcional), a direcional e a isotrópico. A figura 6 mostra, para cada eixo cartesiano, a eficiência de cada tipo de antena:

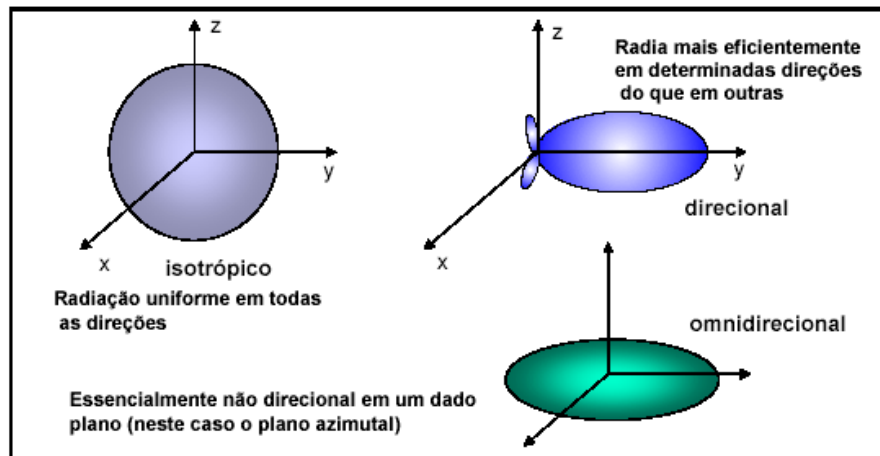


Figura 6. Características dos sistemas de irradiação para projeto de antenas

As antenas omnidirecionais irradiam seus sinais em até dois planos cartesianos, o que explica a dificuldade de se captar sinais desta forma em diferentes níveis de altura. As antenas direcionais irradiam seus sinais em uma direção bem definida, conseguindo desta forma atingir grandes distâncias nas comunicações, sendo ideal para transmissões onde se conhece previamente a localização do par emissor/receptor. Já as antenas isotrópicas irradiam seus sinais em todas as direções possíveis, mas com a consequência de não conseguirem atingir grandes distâncias de comunicação. Se usarmos um transmissor conectado a uma antena omnidirecional e que irradie seu sinal em um raio de 1km, este mesmo transmissor, com uma antena direcional, irradiará o sinal a uma distância de mais de 50km. Por isso a determinação da característica da antena utilizada sempre vai ser dependente do modelo de projeto de comunicação que será feito.

2.5 Regulamentação

No Brasil a legislação para este tipo de sistema foi inicialmente definida pela ANATEL, através da Norma 02/93, posteriormente pela Norma 012/96 (resolução 209 de Jan/2000) e atualmente pela resolução 305 de Jul/2002 – Regulamento sobre Equipamentos de Radiocomunicação de Radiação Restrita.

As faixas de frequências estabelecidas para uso por equipamentos de radiocomunicação empregando a técnica de *spread spectrum*, para aplicações ponto a ponto e ponto multiponto, estão assim definidas: 902MHz a 928MHz, 2400MHz a 2483,5MHz e de 5725MHz a 5850MHz. Desta forma os sistemas que utilizam a tecnologia de *spread spectrum* não necessitam da licença ANATEL para a sua instalação e operação, desde que sejam atendidos os requisitos das Resoluções 209 e 305.

A regulamentação vigente estabelece as condições de operação para os sistemas que operam por Saltos de Frequência, para os sistemas que operam em Seqüência Direta e para os Sistemas Híbridos. Nos sistemas de salto em frequência, na faixa de 900MHz a potência de pico máxima de saída do transmissor não deve ser superior a 1W para sistemas que empreguem no mínimo 50 canais de salto e 0,25W para sistemas empregando menos de 50 canais de salto. Sistemas operando nas faixas de 2,4GHz e 5,8GHz devem trabalhar com potência de pico máxima de saída do transmissor não superior a 1W. Nos sistemas de seqüência direta a potência de pico máxima de saída do transmissor não pode ser superior a 1W para todas as faixas de frequências, e no sistema híbrido devem alcançar um ganho de processamento de, no mínimo, 17 dB na combinação dessas técnicas. Cabe

ressaltar que a Resolução 305 estabeleceu a potência de saída do transmissor para cada sistema baseado numa antena com ganho de 6 dB.

A resolução estabelece também que os sistemas ponto a ponto operando na faixa de 2,4GHz, podem fazer uso de antenas de transmissão com ganho direcional superior a 6dB, desde que a potência de pico máxima na saída do transmissor seja reduzida de 1dB para cada 3dB que o ganho direcional da antena exceder a 6dB.

Os sistemas que operam na faixa de 5,8GHz utilizados exclusivamente em aplicações ponto a ponto, podem fazer uso de antena de transmissão com ganho direcional superior a 6dB sem necessidade de uma correspondente redução na potência de pico máxima na saída do transmissor.

2.6 Sistemas de navegação

Formalmente navegação é o processo de conduzir um veículo de um lugar para outro utilizando algum método de orientação (26) e envolvem questões sobre qual é a posição atual, qual a posição destino, como chegar ao destino a partir da posição atual, qual caminho seguir e que mudanças de percursos são necessárias. Estas questões podem ser respondidas através de vários modelos de navegação:

- **Correção de posição:** consiste na correção da posição de um veículo em relação a um ponto de referência. É o método mais simples e o mais antigo. Existem três técnicas para corrigir posições:

- ✓ **Leitura de mapas:** consiste em associar o que pode ser visto a olho nu com um mapa.

- ✓ Navegação celestial: consiste em calcular a altitude de um corpo celestial (o ângulo entre a linha de visão do usuário e o horizonte) e então obter a posição do observador.
- ✓ Distância e direção de pontos identificáveis: utiliza as direções e as distâncias do veículo até pontos com posições conhecidas, obtendo a posição atual por cálculos trigonométricos.

- **Dead reckoning:** baseia-se na última posição conhecida do veículo, na hora em que essa posição foi obtida e na média da velocidade e da direção para a determinação da nova posição. A velocidade é decomposta para obter as componentes de direção, sendo convertidas em medidas angulares (latitude e longitude) e adicionadas à posição inicial para obter a posição atual.

- **Navegação inercial:** é um sistema que determina a posição e a velocidade a partir da aceleração. A aceleração do veículo é obtida por sensores internos e a velocidade é calculada através de um processo matemático de integração. Em um segundo estágio de integração é calculado a posição do veículo com base em sua posição inicial e sua velocidade.

- **Rádio navegação:** determina a posição de um veículo medindo o tempo de viagem de uma onda eletromagnética de um transmissor em solo até um receptor contido no veículo. Os sistemas de rádio-navegação podem ser classificados em:

- ✓ Rádio-navegação ativa: o veículo envia pulsos à base que os retransmite em uma frequência diferente.
- ✓ Rádio-navegação passiva: o veículo espera a estação enviar os pulsos utilizados no cálculo de sua posição.

- **GPS:** é um sistema orientação de rádio-navegação baseado em satélites artificiais. O usuário conhece a posição de um conjunto de satélites em relação a um

referencial inercial e sua posição em relação a este conjunto, podendo obter sua posição. Este sistema consiste de 24 satélites em 6 órbitas circulares a 20.200km acima da Terra, com um período de 12 horas.

2.7 Conclusões

Este capítulo teve como principal objetivo o estudo de diferentes abordagens de modelos de classificação e arquiteturas para um sistema embarcado com a identificação das principais características que os equipamentos envolvidos necessitam possuir para validar a implementação proposta para esta dissertação. Esse estudo teórico é essencial para a formação estrutural de todo o trabalho, garantindo que todas as aplicações e funcionalidades decorridas da implementação da telemetria de dados e imagens para robôs móveis tenham validade e embasamento científico, corroborando diretamente em aumentar o conhecimento dentro desta área de pesquisa.

3. Projetos Correlatos

O desenvolvimento de sistemas de navegação envolvendo o sensor GPS permite a determinação precisa da posição que qualquer objeto no globo terrestre, e isto trouxe novas oportunidades para a criação de veículos autônomos (terrestres, aéreos e aquáticos). Esta nova tecnologia torna tecnicamente viável a idéia de automatizar processos antes totalmente dependentes da ação direta humana. A seguir são apresentados alguns projetos oriundos de universidades e instituições de pesquisa que se assemelham à implementação definida para esta dissertação.

3.1 Caracterização do Sensor GPS e sua aplicação em um barco autônomo

Este projeto foi implementado pelos departamentos de Sistemas e Controle - IEES e Telecomunicações-IEET do ITA, tendo por objetivo a construção de um barco autônomo, cujo protótipo é mostrado na figura 7, com o intuito de servir como plataforma de testes para estratégias de controle, instrumentação e processamento de sinais (27). Neste trabalho implementou-se um sistema de navegação para aplicação no barco, baseado nos sensores bússola digital e GPS. Um programa desenvolvido em linguagem C coleta os dados do receptor *Trimble*, informando latitude e longitude.

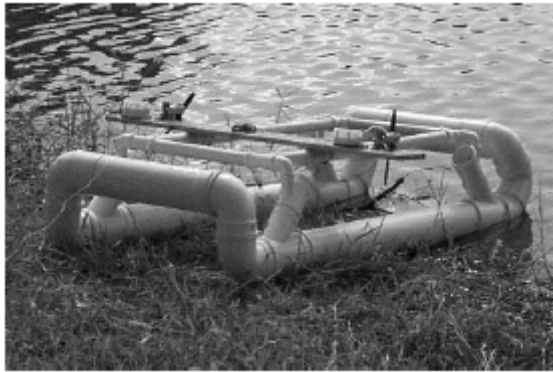


Figura 7. Protótipo do barco

O sistema de propulsão do barco é constituído por duas hélices acionadas por motores modelo O.S. MAX 46 FX (motor a explosão de dois tempos). A cada um deles foi conectado um servoposicionador FUTABA S3003 (que controla a abertura da válvula de entrada de ar), que recebe comandos do computador de bordo, modulados em *PWM* (*Pulse Width Modulation*) por um circuito específico. O aumento da abertura provoca um aumento da potência desenvolvida, aumentando também o empuxo fornecido por cada hélice.

Foi adotado como computador de bordo, por questões de custo e simplicidade, um microcomputador IBM-PC 486DX4/100MHz, que possui as funções de coleta de dados, controle e gerenciamento das atividades de comunicação. A alimentação do micro é provida por um *no-break*, acoplado na parte de baixo da tábua de fixação do computador. O *no-break* confere ao sistema uma autonomia de aproximadamente 20 minutos e fornece energia elétrica durante os ensaios de campo.

A estrutura do barco contém dois equipamentos que fazem o barco realizar curvas para a direita ou esquerda, bem como seu circuito de controle e interfaceamento com o computador de bordo. As curvas também podem ser realizadas por diferença nos empuxos do motor direito e do esquerdo. O leme é

acionado por um motor DC controlado por um circuito que recebe comandos do computador de bordo.

A integração de sensores, atuadores e aplicativo de controle resultam em uma malha de controle multivariável. Os comandos podem ser pré-programados, de modo a fazer com que o barco siga uma trajetória especificada, ou transmitidos por um operador em terra através do enlace de rádio frequência. A figura 8 mostra as comunicações do computador de bordo com o sistema e o conjunto de sensores e atuadores embarcados no barco projetado.

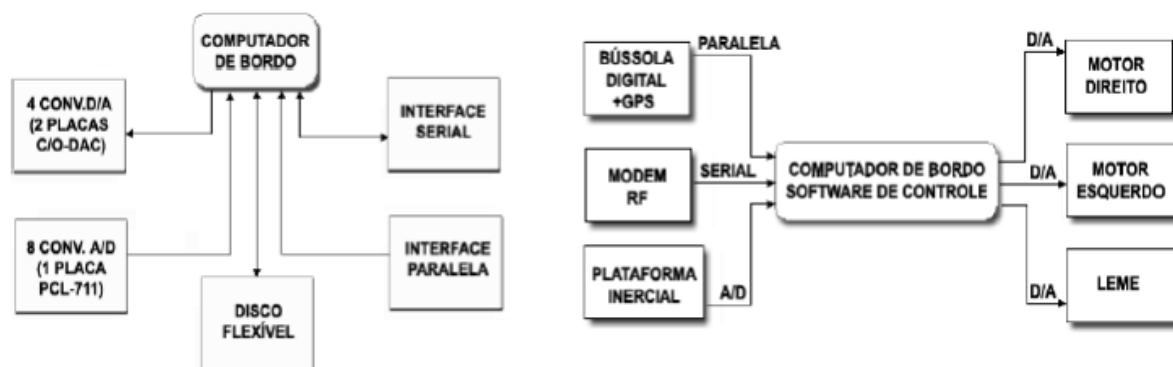


Figura 8. Diagrama em blocos do sistema projetado

Para realizar a coleta de dados, utilizou-se um receptor *Trimble*, um computador e um compilador de linguagem C/C++. O receptor *Trimble* utilizado fornece vários tipos de saídas de dados. Foi escolhida uma saída de dados que fornece a latitude e a longitude do veículo a cada segundo. O formato de saída de dados foi enviado à porta do receptor, ligada à porta COM2 do microcomputador utilizado no laboratório. O programa reconhece estes dados para coletar a informação desejada. Para isso, é necessário que o programa reconheça o cabeçalho "\$PTNL,PJ1" e, conhecendo o padrão de envio de dados, coleta a latitude e longitude procuradas.

3.2 Projeto AURORA

Desenvolver tecnologia de operação em dirigíveis robóticos não tripulados para uso em sensoriamento remoto, monitoração ambiental e inspeção aérea. Esse é o foco dos estudos dos Pesquisadores do Laboratório de Robótica e Visão Computacional (LRVC) do Centro de Pesquisas Renato Archer (Cenpra), do Ministério da Ciência e Tecnologia (MCT), localizado em Campinas. O projeto AURORA – *Autonomous Unmanned Remote Monitoring Robotic Airship*, é considerado um dos mais avançados programas de desenvolvimento para aeronaves que se sustentam no ar graças a depósito de gases (28). O dirigível (figura 9) decola com o auxílio de operadores e segue de forma automática uma trajetória de vôo previamente designada.



Figura 9. Dirigível usado no projeto AURORA

O Projeto AURORA está dividido em três fases, com valores diferentes para o número máximo de horas de vôo, para a distância máxima percorrida e para a capacidade de carga útil transportada. Os componentes da primeira fase do Projeto AURORA compreendem o dirigível, subsistema de navegação e o controle a bordo da aeronave, incluindo sensores internos, *hardware*, aplicativo, os sensores de

missão e uma estação base móvel. Os subsistemas de navegação e controle de missão são responsáveis pela aquisição de dados e pelo controle dos atuadores. O *hardware* do dirigível é composto por um computador, microprocessadores, sensores e atuadores. O sistema de comunicação é composto por enlaces de rádio frequência que transmitem dados e comandos entre o dirigível e a estação base. Inclui-se também transmissão de imagens de vídeo, capturadas pelas câmeras fixadas no dirigível.

O sistema de controle foi projetado como uma estrutura hierárquica de três camadas. A camada mais baixa contém os atuadores que manobram o dirigível conduzindo ao longo de seu curso. Na camada intermediária algoritmos de controle comandam os atuadores baseando-se nas decisões feitas na camada superior, que decide quais algoritmos de controle serão ativados e os atuadores necessários. Acima do sistema de controle existe uma outra camada para lidar com a navegação autônoma. Exemplos dessa arquitetura são encontrados em (29) e (30).

Dirigíveis não tripulados já funcionam como aeromodelos, por meio de um rádio controle da terra. O AURORA propõe, como contribuição científica e tecnológica, a concepção do aplicativo necessário à operação autônoma do veículo, num nível ainda não encontrado comercialmente. Isso inclui aspectos inovadores, desde os algoritmos de controle para a estabilização da aeronave em vôo e seguimento de trajetória, até um nível hierárquico superior compreendendo a inteligência para percepção, diagnóstico e tomada de decisão, fatores necessários à operação autônoma do dirigível robótico.

Um robô móvel autônomo deste nível possui um vasto campo de aplicação. Pode ser empregado no monitoramento e estudo de florestas e regiões de interesse ecológico, como a Amazônia. Faz levantamento em áreas rurais de aspectos

agropecuários, tais como cobertura ou uso do solo, avaliação de colheitas e de número de animais. Também pode auxiliar na medição da composição do ar e de níveis de poluição e sua dispersão em centros urbanos e industriais. Além disso, o dirigível robótico serve para a inspeção de grandes estruturas - como oleodutos, gasodutos e linhas de transmissão, levantamento de ocupação urbana e prospecção topográfica, mineral e arqueológica. Aplicações em segurança pública ou vigilância também estão na lista dos usos do veículo aéreo. Para as aplicações de monitoração ambiental climatológica e de biodiversidade estão o sensoriamento e monitoração de florestas e de regiões de interesse ecológico, levantamentos agrícolas e agropecuários, medição da composição do ar e de níveis de poluição em cidades e centros industriais, e estudos limnológicos em rios, lagos e regiões costeiras.

3.3 Projeto ARARA

O projeto ARARA (Aeronaves de Reconhecimento Assistidas por Rádio e Autônomas), desenvolvido no Instituto de Ciências Matemáticas e Computação da USP de São Carlos, com cooperação da EMBRAPA-CNPDIA de São Carlos, está centrado no uso de aeronaves em escala reduzida para monitoramento. Este projeto possui como principal objetivo a construção de UAV's (Unmanned Aerial Vehicles) na obtenção de fotografias aéreas para o monitoramento de plantações e áreas sob controle ecológico. O projeto está dividido em quatro fases, sendo cada um com níveis de complexidade e custo crescentes para diferentes aplicações:

- **Fase I:** possui o mínimo de funcionalidade e apresenta o menor custo com a máxima facilidade de implementação (31). Nessa fase a aeronave é baseada em componentes convencionais utilizados em aeromodelos e o controle de vôo é feito a partir do solo em um raio de até 1Km. Uma câmera fotográfica (digital ou convencional) é adaptada a aeronave e é disparada via comando por rádio. É possível a instalação de câmera e um transmissor de vídeo para o monitoramento de regiões em tempo real ou para o melhor posicionamento da aeronave sobre a área a ser fotografada. Esse tipo de sistema é adequado para aplicações onde o vôo da aeronave pode ser controlado de uma estação em solo, da mesma forma praticada com os aeromodelos. Sistemas dessa fase apresentam o menor custo entre todos os sistemas definidos no Projeto ARARA.

- **Fase II:** Essa fase envolve o projeto e a construção de um aeromodelo com uma grande variedade de sensores para o controle do vôo tais como *GPS*, giroscópio, bússola, velocímetro, altímetro, horizonte artificial e instrumentos do motor (32). Estes permitem o vôo sem a necessidade de acompanhamento visual da aeronave, embora possa ser remotamente pilotada a partir de uma estação em solo. As imagens são vistas pelo piloto na tela de um microcomputador em solo, sendo transmitidas através de duas câmeras de vídeo instaladas na aeronave. O painel de controle assemelha-se a o de um simulador, mostrando todas os sensores/atuadores envolvidos no sistema.

- **Fase III:** Nesta fase há um sistema de navegação e controle (33) permitindo o vôo autônomo e missões completamente autônomas (34). O usuário pode programar a rota de vôo da aeronave e as tarefas a serem cumpridas como fotografar ou filmar uma área pré-definida.

- **Fase IV:** A fase IV acrescenta ao projeto processamento de imagens a bordo

da aeronave. A execução de missões é controlada por decisões baseadas no processamento das imagens obtidas em tempo real. Com essa tecnologia há a possibilidade de uso na identificação de objetos como linhas de transmissão de energia, estradas, cursos de rio, entre outros.

Na fase IV o STT (Sistema de Telemetria e Telecomando) deste projeto foi desenvolvido para possibilitar o envio de sinais de telemetria e telecomando entre uma aeronave e uma estação solo. Para realizar a transmissão de dados entre o UAV e a estação solo foi especificado um protocolo de comunicação específico (35). Este protocolo tem a função de coordenar a transmissão dos dados através de um canal de comunicação digital sem fio, possibilitando, assim, a comunicação bidirecional entre o *UAV* e a estação solo. Os dados transmitidos do *UAV* para a estação solo (telemetria), são úteis para o acompanhamento das missões. Os dados transmitidos da estação solo para o *UAV* (telecomando) são utilizados quando o usuário deseja pilotar remotamente ou modificar algum parâmetro de configuração do *UAV*. A figura 10 mostra os componentes de *hardware* e seus respectivos módulos e interligações existentes:

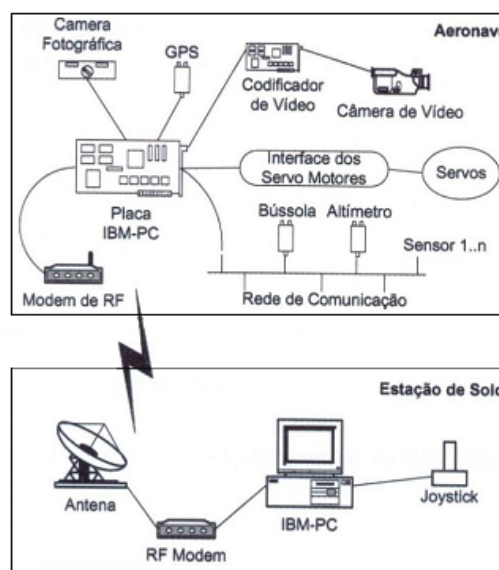


Figura 10. *Hardware* do projeto ARARA

O sistema de telemetria e telecomando do projeto ARARA permite a interação entre um usuário e um *UAV*, remotamente pilotado ou autônomo. Utilizando este sistema o usuário pode acompanhar um voo observando as mudanças de estado da aeronave (altitude, direção, velocidade, etc), transmitir comandos para a mesma e assistir ao vídeo transmitido pela aeronave. O objetivo do STT é fazer o usuário interagir com a aeronave como num jogo de simulador de voo.

3.4 Conclusões

Os projetos correlatos descritos neste capítulo propiciam para este trabalho de mestrado a confirmação de que a comunicação entre dispositivos de veículos autônomos por ondas de rádio e o uso de equipamentos de geoposicionamento na navegação é atualmente utilizado em diversos projetos de automação nas mais diferentes plataformas (protótipo de embarcação, aeronave e dirigível).

O protótipo de embarcação geoposicionado é interessante pelo fato de também utilizar uma plataforma para navegação fluvial e de uso da bússola eletrônica e equipamento *GPS* em sua navegação, possuindo também para comunicação em tempo online um enlace de rádio frequência. Seus princípios de funcionamento fundamentam e mostram a possibilidade de se realizar as implementações propostas neste trabalho.

O projeto AURORA similarmente utiliza um aplicativo na estação terrestre e *hardware* embarcado, com inúmeros exemplos de aplicações que uma plataforma autônoma pode ser usada. O sistema como um todo utiliza uma arquitetura muito próxima da implementação a ser apresentada posteriormente, o que indica a

viabilidade de sistemas deste tipo para automação de veículos para monitoração de variáveis ambientais.

A principal contribuição do projeto ARARA como referência bibliográfica é a idéia de utilização de *modems* de rádio frequência de uso comercial que possam suprir todas as funcionalidades que o projeto exige, como velocidade de transmissão e tecnologia de transmissão de dados sem fio adequado. O projeto ARARA também inclui em sua arquitetura a transmissão de imagens em tempo online, provando a necessidade de seu uso para controle, instrumentos de navegação para geoposicionamento, imprescindíveis para navegação autônoma e um aplicativo de visualização de todos valores dos sensores/atuadores envolvidos no sistema, de forma altamente correlata às implementações que serão discutidas adiante.

4. Sensores e módulos R.F. da telemetria

4.1 Sensores embarcados para navegação

A navegação autônoma da plataforma autônoma desenvolvida é toda embasada pelos sensores *GPS* e bússola eletrônica. Atualmente o sensor embarcado *GPS* é de um modelo comercial de uso geral, o que implica sempre em configurações iniciais manuais, fugindo do escopo original do projeto, que é o uso de dispositivos eletrônicos dedicados e adaptáveis a algum encapsulamento. Desta forma adquiriu-se um módulo sensor *GPS* do fabricante *Dynatech* (36) que se encaixa perfeitamente ao conceito proposto para o projeto. Todas as informações referentes ao mesmo serão discutidas em tópico a seguir.

A bússola eletrônica, do fabricante *Precision Navigation, Inc* (37), é essencial para a navegação, já que o *GPS* não é capaz de fornecer a orientação magnética sem a movimentação contínua da embarcação, pois seus cálculos são trigonométricos. Na coleta de dados hidrológicos a embarcação estacionará, e nesse momento, pela ação da própria correnteza e clima, a mesma sofrerá fatalmente mudança de orientação. Nesse caso a bússola eletrônica ajudará na correção da orientação e também proferir precisão na manobrabilidade. Detalhes de seu funcionamento serão vistos em tópico subsequente.

4.1.1 Módulo receptor GPS

O módulo receptor GPS, com apenas 55mm de comprimento, incorpora um circuito completo de recepção e decodificação de sinais dos satélites, não requerendo ajuste, programação ou calibração para operação (figura 11). Ele proporciona interface serial direta, facilitando sua conexão com outros dispositivos.



Figura 11. Módulo receptor GPS

A comunicação com o mundo exterior dá-se por um canal serial na velocidade de 4800bps, sendo o pacote de saída de dados na formatação padrão ASCII, com um baixo consumo de energia. A tabela 3 fornece suas principais características de operação:

Tabela 3 - Características gerais do módulo receptor GPS

Chipset	ST Microelectronics: STB5610 (front-end de RF), ST20GP7 (CPU RISC 32 bits)
Número de canais	12 simultâneos, com correlação por <i>hardware</i> /aplicativo
Frequência de operação	Sistema GPS banda L1 (1.575,42MHz)
Comunicação serial	4800bps, 8N1, TTL nível 3,3Vcc
Dados de saída	Sentenças NMEA 0183
Tensão de alimentação	Módulo: 3,3Vcc +5% Antena Ativa: 3,3Vcc a 5,0Vcc
Consumo	130mA e 5mA em standby
Temperatura de operação	-40 ⁰ C a +85 ⁰ C
Conector para antena ativa	SMA

As medidas de seu desempenho (tabela 4) são favoráveis ao uso na implementação proposta, justamente pelo ambiente não oferecer obstáculos naturais

e nem pontos de obscurecimento. A medida mais crítica é em relação à primeira aquisição, que tem duração máxima de 90 segundos. Este atraso não refletirá no baixo desempenho do tempo de resposta da telemetria, pois antes da embarcação começar a navegar o usuário do *aplicativo* gastará um tempo maior do que a restrição para configurar e inserir todo o conjunto de pontos de coleta no sistema.

Tabela 4 - Medidas de desempenho

Precisão ⁽¹⁾ ⁽²⁾	Posicionamento horizontal	melhor que 3 metros
	Posicionamento vertical	melhor que 5 metros
Tempo de aquisição para o primeiro fixo (Time To First Fix, TTFF)	Cold <i>start</i> autônomo	90 segundos (típico) ⁽³⁾
	Warm <i>start</i>	30 segundos
	Hot <i>start</i>	3 segundos
	Requisição interna	0,5 segundos
	Após obscurecimento ⁽⁴⁾	1 segundo
Datum	WGS84	

(1) Em condições favoráveis: >85% céu visível pela antena, ausência de precipitação e atividade solar normal
 (2) Sinal SA desligado pelo Departamento de Defesa dos Estados Unidos (desde maio/2000)
 (3) Tempo máximo observado em condições favoráveis descritas em (1)
 (4) Perda de sinal momentânea provocada por obstáculos físicos com túneis e grandes estruturas

O módulo receptor *GPS* comunica-se serialmente através de um protocolo bem definido, desenvolvido pela *National Marine Electronics Association* (NMEA). A versão 0183 implementada é a mais completa do ponto de vista da informação, e seu conjunto de dados mais completo. Essa abordagem permite uma maior flexibilidade na interface, sendo esta diretamente incorporada ao *hardware* pelo padrão RS-232 (todas as comunicações dos dispositivos eletrônicos embarcados e os canais seriais entre os microcontroladores utilizam como veículo de comunicação o padrão RS-232).

A saída serial do módulo receptor *GPS* disponibiliza vários tipos de sentenças com finalidades diversas: informar a posição em latitude e longitude, estado dos satélites, etc. A sentença escolhida para uso (tabela 5) fornece todas as informações necessárias para implementação da navegação autônoma proposta:

Tabela 5 - Exemplo numérico da sentença NMEA escolhida

\$GPRMC,093244,A,5312.022,N,02917.232,W,034.1,042.7,210504,019.8,E*HH	
093244	Fixo determinado às 9:32:44s
A	Alerta do receptor de navegação A = normal; V = alerta
5312.022,S	Latitude 53 ⁰ 12,022' Sul
02917.232,W	Longitude 29 ⁰ 17,232' Oeste
034.1	Velocidade em nós
042.7	Curso verdadeiro em graus
210506	Data
019.8,E	Varição magnética local
HH	Palavra soma ("Checksum")

A sentença sempre é formada por caracteres ASCII imprimíveis, começando pelo identificador "\$", seguido de um campo que identifica o equipamento transmissor, pelos campos de dados separados por vírgulas e finalizando por uma palavra soma, calculado pela lógica booleana "ou". A sentença pode ter até 82 caracteres, incluindo-se o "\$" e o CR/LF, e se um campo não estiver disponível ele é omitido na sentença, mas as vírgulas que o delimitam são transmitidas sem espaços entre elas. Para descobrir a correta posição de cada dado dentro do pacote a única forma de avaliação é através da contagem das vírgulas separadoras.

4.1.1.2 Interface

A alimentação do módulo receptor *GPS* é diferente de todos os componentes eletrônicos já embarcados, necessitando neste caso a construção de uma nova interface. O esquemático da figura 12 mostra todas as conexões entre os componentes usados:

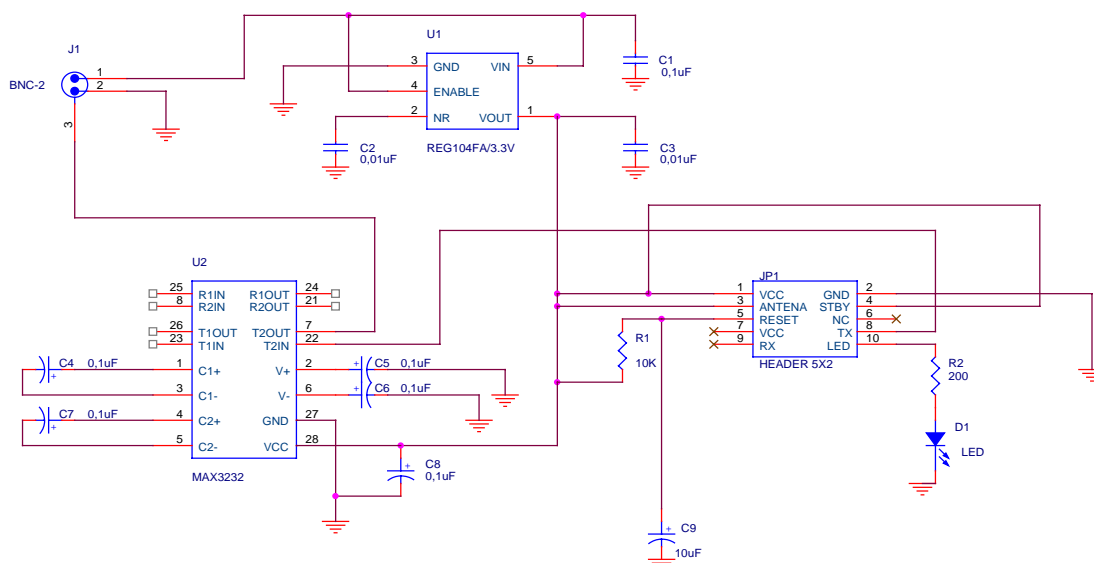


Figura 12. Esquemático da interface do módulo receptor GPS

O regulador de tensão na entrada do circuito converte a alimentação que vem de uma bateria automotiva de 12Vcc para os 3,3Vcc usados no módulo. O componente MAX3232 é um conversor TTL/CMOS que disponibiliza em sua saída o padrão RS-232, o qual é o adotado para todas as comunicações embarcadas. Todos os sinais de entrada e saída estão disponibilizados em um único conector, onde um *led* verificador de estado foi incorporado para análise direta de funcionamento do módulo durante o processo de pesquisa em campo.

4.1.1.3 Encapsulamento

O módulo receptor *GPS* e a interface foram conectadas e agrupadas em uma caixa plástica, para proteção e facilitação de seu manuseio (figura 13). Apenas dois furos são necessários para seu funcionamento, sendo um para a antena ativa e o outro para o cabo que conecta o *hardware* ao sistema microcontrolado central.

Estes receberam um tratamento para não permitir que entre poeira ou gotas de água que possam prejudicar ou mesmo inviabilizar o funcionamento do sistema.



Figura 13. Vista completa da caixa aberta

4.1.2 Bússola eletrônica

A *Vector Electronic Module* é uma bússola eletrônica fabricada pela empresa *Precision Navigation, Inc*, de baixo custo e com 2 eixos magnéticos para uso em aplicações embarcadas (figura 14). Ela utiliza tecnologia de magnetômetro magneto-indutivo que garante alta precisão e baixo consumo.



Figura 14. Módulo da bússola eletrônica

Entre suas principais características de trabalho estão a precisão de leitura de 2 graus, resolução de 1 grau e configurabilidade para frequência de trabalho. A tabela 6 apresenta um resumo das principais características deste sensor:

Tabela 6 - Principais características da bússola eletrônica

Precisão	2 graus
Resolução	1 grau
Alimentação	5Vcc
Frequência de trabalho	5Hz em baixa resolução 2,5Hz em alta resolução
Formato de saída	decimal ou binária
Configuração	mestre ou escravo
Temperatura de operação	-20°C até 70°C

Todas essas características fundamentam seu uso na proposta de implementação de telemetria de dados e navegação autônoma, contribuindo diretamente para um melhor desempenho na otimização de rotas e maior precisão de mudança de direção pelo uso do leme.

4.1.2.1 Interface

Por tratar-se de um equipamento no padrão OEM há a necessidade da construção de uma interface de comunicação entre a eletrônica embarcada e a bússola eletrônica (figura 15).

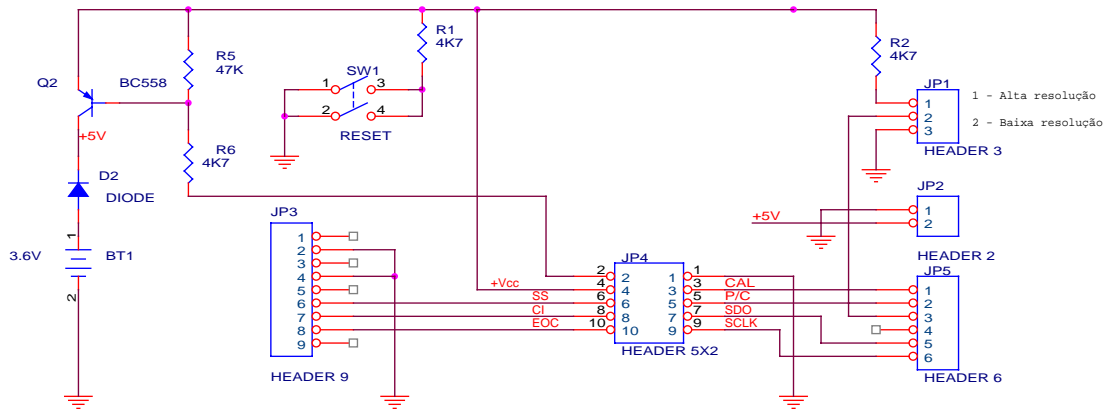


Figura 15. Esquemático da interface com a bússola eletrônica

Através desta interface pode-se escolher a freqüência de trabalho (alta/baixa), resetar a bússola em caso de descalibração e esta disponibiliza ao microcontrolador do hardware embarcado todos os pinos relevantes para a configuração, incluindo o canal de dados que fornecerá o valor da orientação no padrão desejado.

4.1.2.2 Acabamento e proteção

A bússola eletrônica e a interface foram encapsuladas dentro de um cano PVC com tampas inferiores e superiores (figura 16). Para fixação foram utilizados dois parafusos de rosca em abas coladas à circunferência do cano e construídas do mesmo material do cano. Abaixo da bússola eletrônica furou-se o cano para a passagem do cabo flexível para que este fosse posteriormente ligado à caixa dos sistemas microcontrolados. Mesmo estando a passagem abaixo dos componentes eletrônicos, o que já evitaria o contato com água ou poeira, a passagem recebeu um tratamento de vedação adicional.



Figura 16. Encapsulamento da bússola eletrônica

4.2 Equipamentos para comunicação por R.F.

Para toda arquitetura de controle destinada a aplicações de robôs móveis autônomos é essencial existir um canal de monitoramento à distância que permita avaliar seu comportamento ao longo da pesquisa e também para uma pós-avaliação mais criteriosa, a fim de investigar possíveis falhas de implementação. Esse canal de comunicação é extremamente necessário mesmo em veículos totalmente autônomos.

Para satisfazer este requisito usam-se em geral enlaces de rádio frequência. Para a implementação proposta decidiu-se pelo uso de dois canais de comunicação: um dedicado exclusivamente à telemetria de dados, que monitorará todo o sistema embarcado, e outro para a transferência do sinal multimídia (áudio + vídeo), essencial para o acompanhamento visual da embarcação mesmo a longas distâncias, permitindo a aplicação do piloto virtual via aplicativo. A especificação dos dispositivos de cada canal de dados é vista com detalhes nas seções subseqüentes.

4.2.1 Enlace da telemetria de dados

Para a escolha dos equipamentos de comunicação da telemetria de dados entre a embarcação autônoma e a estação terrestre os seguintes requisitos foram levados em consideração: grande alcance na comunicação (maior que 10km), facilidade de uso (configuração), interface compatível com o projeto já proposto (padrão RS-232), alta velocidade no fluxo de dados, política de retransmissão de pacote com erros e criptografia.

A partir do modelamento das restrições e depois de uma pesquisa refinada com alguns fabricantes de modems de rádio frequência optou-se pelo conjunto de desenvolvimento do fabricante *MaxStream* (38), composto de dois módulos transmissores-receptores (*tranceivers*) e uma gama de acessórios com extensa documentação, fonte de alimentação, cabos de conexão, aplicativo de testes para simulação de comunicação e testes de *hardware*, conjunto de antenas, etc. Este conjunto está ilustrado na figura 17.



Figura 17. Kit de desenvolvimento da *Maxstream*

Com transmissores de potência máxima de 1W o fabricante garante transmissões numa faixa de até 64km de distância de raio. A tabela 7 mostra as

características de desempenho, alimentação, tamanho, faixa de frequência de uso e tecnologia envolvida, entre outros aspectos.

Tabela 7 - Aspectos gerais dos *tranceivers* R.F.

Características Gerais	
Banda de operação	902-928MHz
Tecnologia de transmissão	spread spectrum FHSS
Interface de dados serial	3V – 5V CMOS UART
Encriptação	256 bits
Desempenho	
Potência de saída	1mW – 1W (0 – 30dBm)
Alcance em ambiente fechado	900m
Alcance em ambiente sem obstáculos	64Km
Velocidade de transmissão	9.6 ou 115.2kbps
Sensitividade	-110dBm (9600bps)
Alimentação	
Tensão de trabalho	2,8 a 5,5Vcc regulado
Corrente no transmissor	730mA (@5V e 1W no transmissor)
Corrente no receptor	80mA (@5V e 1W no transmissor)
Operação em standby	1uA
Propriedades físicas	
Tamanho	3,65cm x 6,05cm x 0,51cm
Peso	18gramas
Antena	ativa (conectores RPSMA ou MMCX)
Temperatura de trabalho	-40°C até 85°C (industrial)

A principal ferramenta de diagnóstico que vem junto ao kit de desenvolvimento é um aplicativo que verifica se os módulos operam corretamente e faz um teste de comunicação entre eles. Esta ferramenta é essencial na etapa de projeto, pois permite delimitar explicitamente nos testes reais se o problema está no equipamento de transmissão ou se está no *hardware* embarcado. Na figura 18 temos duas telas gráficas com exemplos de uso dessa importante ferramenta de testes:

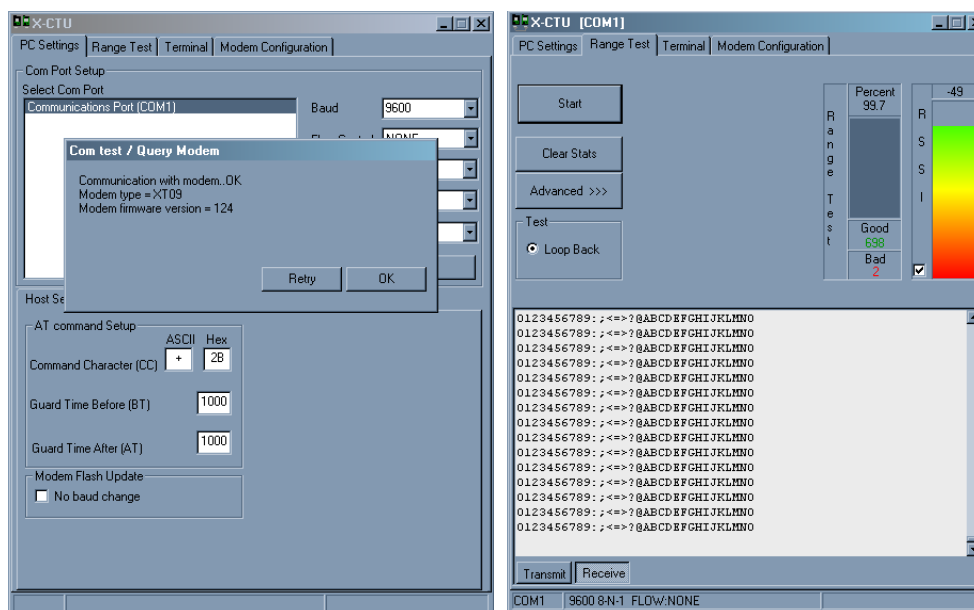


Figura 18. Exemplos de uso do aplicativo de diagnóstico dos *tranceivers*

Escolhendo no aplicativo a pasta *PC Settings*, opção *Test/Query* e conectando o *tranceiver* R.F. na interface serial do *notebook* o usuário saberá se o *modem* está em funcionamento normal ou não (imagem à esquerda). Isso é importante saber pois pode ocorrer a falta do sincronismo imediato entre os *modems* justamente pelo uso da tecnologia de saltos de frequência, sendo às vezes necessário reiniciar o *hardware* novamente, acelerando assim o processo de uso dos mesmos.

O teste de transmissão entre *modems* é visto no aplicativo pela pasta *Range Test* (imagem à direita). O usuário alimenta um *modem* e conecta-o ao canal serial do *notebook*, enquanto o outro *modem* é também alimentado e em seu conector é adicionada uma peça de diagnóstico que vem junto ao kit, denominada *loopback*. Todos os dados recebidos por este *modem* serão prontamente reenviados ao *modem* origem, fechando assim a malha. Desta forma é possível avaliar quais expressões foram transmitidas com sucesso ou não. No exemplo da imagem enviou-se um conjunto de caracteres da tabela ASCII e observou-se o envio de 700 pacotes, sendo apenas 2 deles com erros (99,7% de acerto). Este teste foi realizado em campo aberto, na Represa do Broa, a uma distância entre *modems* de até 6km.

Outros testes de transmissão e maiores detalhes serão vistos na seção Testes e resultados.

4.2.1.1 Proteção e acabamento

O par de *modems* R.F. recebeu tratamento de proteção e acabamento especial, justificado pelo seu valor de custo e por conter uma eletrônica que não permite exposição à poeira e umidade. Para encapsulamento (figura 22) tem-se caixa plástica com parafusos na tampa e para fixação dos *modems* às mesmas apoios de borracha e pequenos parafusos com rosca.



Figura 19. Encapsulamento do *modem* R.F.

Para garantir a completa vedação do sistema a conexão serial do equipamento ficou interna à caixa, com pode ser visto pela imagem à direita. Apesar do kit de desenvolvimento disponibilizar cabos seriais um novo foi produzido reduzindo o tamanho da caixa e garantindo melhor fixação. As saídas para a antena, fonte e cabo serial foram vedadas com cola de silicone.

4.2.2 Enlace de R.F. multimídia

A transmissão dos dados multimídia para a estação terrestre é realizada por um canal separado da telemetria de dados, através de um outro conjunto de transmissão (figura 20). O sistema completo de comunicação inclui uma câmera de vídeo com tecnologia CCD de 450 linhas de resolução, transmissor (frequência de 2,434 GHz) com potência de 1W (39), antena receptora direcional e o monitor de vídeo para visualização da imagem obtida pela câmera e recebimento do áudio, através de um microfone na embarcação.



Figura 20. Kit completo para transmissão de áudio/vídeo

O canal de dados multimídia é de extrema relevância para o projeto, já que através dele o usuário em terra, além de receber a telemetria de dados completa da embarcação, consegue acompanhar e escutar tudo o que acontece a bordo, pelo monitor disponível em conjunto com o aplicativo. No início da pesquisa de coleta de dados o usuário pilotará remotamente a embarcação até um local de partida seguro (longe das margens), e no caso da telemetria acusar algum erro na leitura de algum sensor da navegação, que impeça o *hardware* embarcado de navegar com sucesso automaticamente, o usuário poderá através do aplicativo em terra abortar a missão

pretendida e ativar o piloto virtual, que disponibiliza o controle de atuação sobre os motores da aceleração, direção e propulsão. Nesse caso, independentemente se a embarcação está ou não no plano de visão do usuário, ele poderá navegar com segurança e trazer novamente a embarcação ao seu ponto de partida.

Através desse canal de dados disponibilizado por este projeto posteriormente novos pesquisadores terão a oportunidade de incorporar à embarcação algoritmos de visão computacional que previnam colisão com obstáculos e que auxiliem a navegação, já que a presente implementação não aborda o assunto. Idéias para projetos futuros que complementem o projeto descrito nesta dissertação serão melhores discutidos no tópico sugestões para trabalhos futuros.

4.2.2.1 Proteção e acabamento

Para a estação terrestre (figura 21) a antena receptora recebeu um suporte de fixação para melhora da recepção do sinal.



Figura 21. Estação terrestre

Todos os outros componentes do enlace multimídia embarcados foram alinhados num cano de alumínio e fixados na popa do barco (figura 22).



Figura 22. Componentes do enlace multimídia embarcado

Pela imagem de vídeo o usuário do sistema tem uma ampla visão de toda a embarcação e ainda acompanha a ação do sistema embarcado, quando a navegação autônoma atua sobre os motores do leme, acionando a direção, e sobre a aceleração, mudando a posição do manche. É a partir desta imagem que o usuário pilotará a embarcação em todo início de pesquisa e nas situações de emergência, através da operação de piloto virtual, disponibilizada pelo aplicativo em tempo mínimo.

5. Projeto e implementação

Neste capítulo é apresentada a implementação do sistema de telemetria completo e controle da plataforma autônoma. A seção 5.1 descreve todo o diagrama em blocos do modelo de arquitetura implementada, propiciando uma visão macroscópica de todo o projeto. A seção 5.2 apresenta as características e fundamentos da criação do aplicativo terrestre para monitoramento da telemetria de dados em tempo on-line, incluindo os menus de interface com o usuário, formatação do arquivo de dados do relatório, entrada dos dados, otimização de rota, descrição do pacote de dados da telemetria e todos os métodos aplicados para a apresentação gráfica dos sensores embarcados. O capítulo se finaliza com a seção 5.3 que relata toda a metodologia aplicada na comunicação com os sensores embarcados, incluindo fluxogramas para descrição das rotinas implementadas em linguagem de baixo nível, testes de validação e calibração do sistema.

5.1 Diagrama em blocos da arquitetura implementada

O sistema desenvolvido permite a navegação autônoma da embarcação por geoposicionamento e orientação da bússola eletrônica. O tráfego de dados da telemetria e do sinal multimídia é realizado por enlaces de rádio frequência, objetivando a visualização em tempo on-line de todos os parâmetros de relevância no transcorrer da pesquisa pelo aplicativo construído para uso na estação terrestre, conforme visto na figura 23.

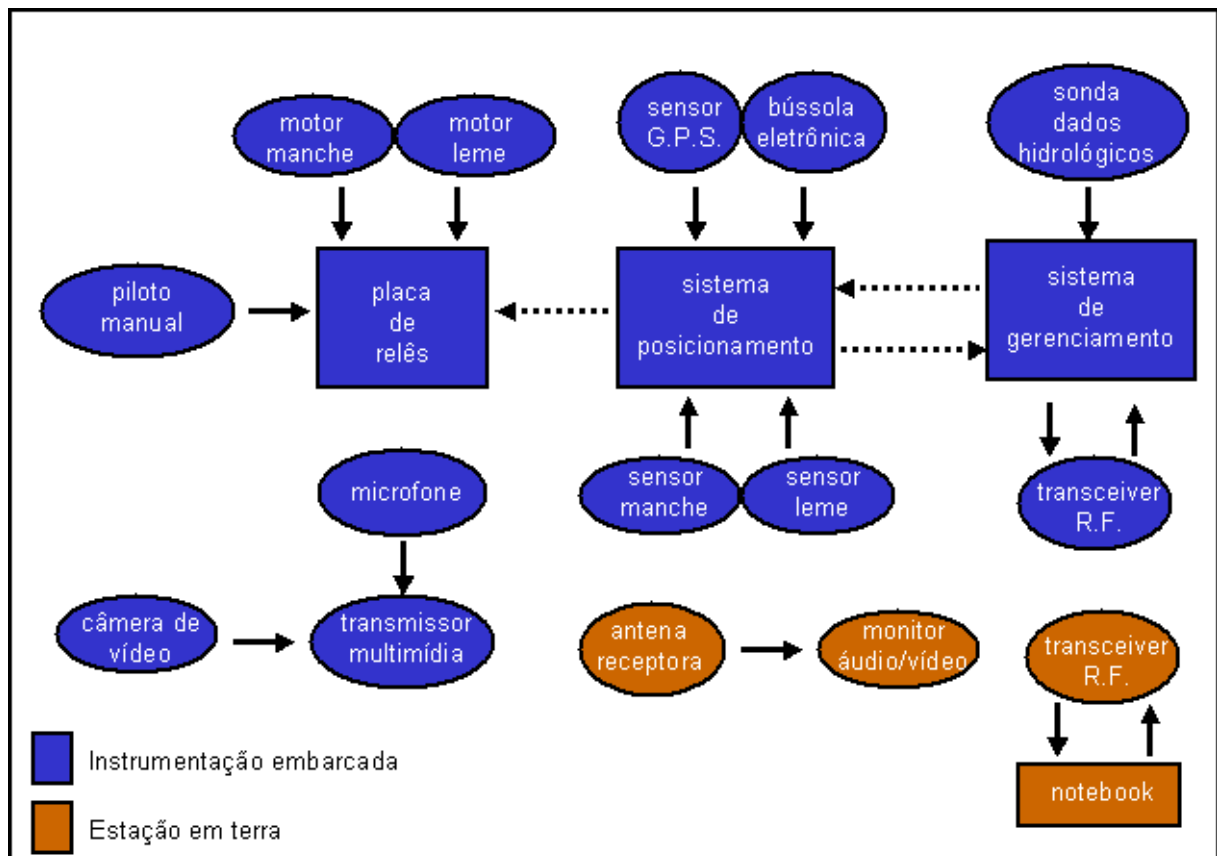


Figura 23. Diagrama em blocos do sistema de telemetria de dados e imagens da plataforma autônoma para coleta de dados hidrológicos

O usuário do sistema, através do aplicativo da estação terrestre, indica quais são os pontos de pesquisa a serem coletados, sendo esses referenciados pela sua latitude e longitude global. Os pontos inseridos serão ordenados através de um algoritmo de otimização, objetivando minimizar o processo de navegação. Após este primeiro passo a instrumentação embarcada recebe um pacote de dados através de um enlace de rádio frequência (*transceiver*) com a ordem de pesquisar o primeiro ponto de coleta. O *hardware* presente na embarcação tem como responsabilidade interfacear todos os sensores disponíveis, efetuar a navegação autônoma e enviar os pacotes de dados para a estação terrestre. O sistema de gerenciamento tem a responsabilidade de aplicar um método de navegação e gerenciar toda a comunicação da eletrônica embarcada, e para isso comunica-se serialmente com o sistema de posicionamento, através de pacotes de troca de mensagens com

métodos de controle de erro e fluxo, requisitando informações quanto ao status dos sensores e atuadores e enviando comandos para mudança de rumo da embarcação. O sistema de posicionamento tem a responsabilidade de empacotar todas as informações a respeito dos sensores e atuadores conectados ao seu módulo e de atuar diretamente na placa de relês, que faz a interface direta com os motores conectados à embarcação.

Todo o pacote de dados que o sistema de gerenciamento recebe durante o processo de pesquisa é imediatamente enviado, através do módulo *transceiver*, para a estação terrestre, que mostra em tela gráfica o andamento completo da missão. A responsabilidade do transmissor multimídia é enviar os sinais de áudio e vídeo para um monitor na estação terrestre, que é usado para o acompanhamento visual da embarcação e principalmente na necessidade da ativação do modo de emergência, permitindo ao usuário do sistema a navegação remota da embarcação (piloto virtual).

Os detalhes de implementação de cada bloco do diagrama, como a interface com os sensores e atuadores, troca de mensagens entre sistemas, políticas de prevenção de travamentos e as especificações sobre o uso do aplicativo são apresentados a seguir.

5.2 Aplicativo da estação terrestre

O programa implementado na estação terrestre é o principal veículo de comunicação do usuário do sistema com a plataforma autônoma. Desenvolvido em linguagem de alto nível, na plataforma Delphi 7.0, tem como principais funções

enviar as coordenadas dos pontos de coleta para a eletrônica embarcada, através do *modem* da telemetria de dados, receber o fluxo de pacote de dados, para visualmente mostrar o status de todos os sensores/atuadores embarcados na tela principal, permitir o uso do piloto virtual para navegação remota e armazenar todo o fluxo de dados da telemetria em arquivo digital, sendo este disponibilizado para consulta via geração de relatório na tela.

5.2.1 Iniciando o aplicativo

O aplicativo disponibiliza ao usuário um menu de entrada com as seguintes opções:

- **Novo:** nova pesquisa de coleta da telemetria de dados.
- **Abrir Relatório:** geração de relatório de arquivo via tela principal.
- **Sair:** finaliza o programa.

A tela principal do aplicativo (figura 24), que mostra todos os objetos que formam a interface visual dos dados para o usuário, sempre é o pano de fundo para todas as opções que o aplicativo possui, evidenciando que esta é a sua principal função.

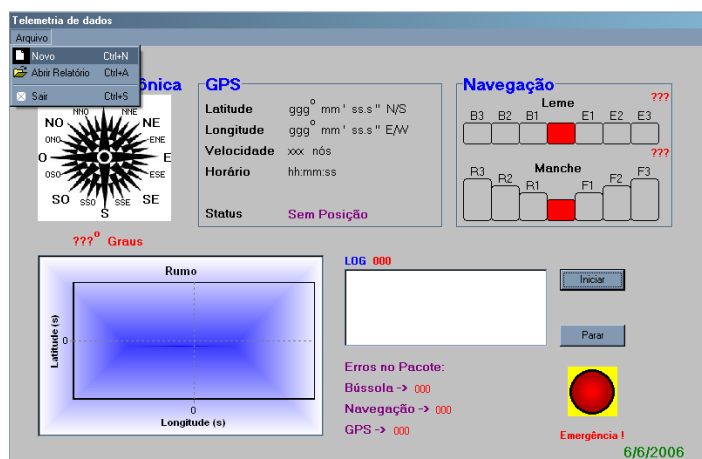


Figura 24. Menu Arquivo e tela principal do aplicativo

A geração de uma nova pesquisa da telemetria de dados para a coleta de dados hidrológicos implica sempre na criação de um novo arquivo digital com um cabeçalho específico, através de um padrão de formatação que será interpretado posteriormente pelo gerador de relatório. A figura 25 apresenta a estrutura de arquivo implementada e descreve cada elemento do arquivo texto digital:

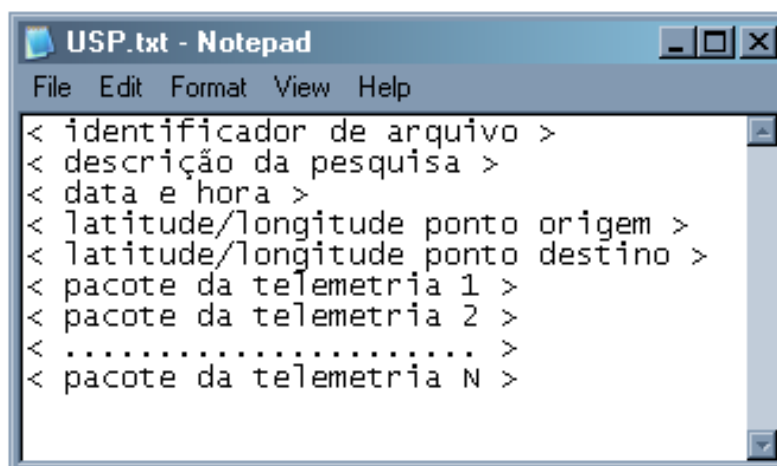


Figura 25. Arquivo digital da telemetria de dados

Com o identificador de arquivo exclui-se uma probabilidade de erro, em tempo de execução, pelo aplicativo por abrir algum arquivo que não tenha sido criado com a finalidade de armazenamento dos pacotes da telemetria de dados; a descrição da pesquisa fornece uma rápida introdução dos objetivos da coleta de dados; o ponto

origem e destino são importantes para futuras comparações em forma gráfica para indicar a precisão da navegação e os pacotes da telemetria de dados são armazenados seqüencialmente em arquivo texto para serem novamente remontados graficamente via relatório para o usuário do aplicativo.

5.2.2 Entrada de dados e otimização de rota

O início à pesquisa dos dados dos sensores embarcados depende essencialmente de um conjunto de pontos de coleta, de um processo de otimização de rota e o estabelecimento da comunicação, via rádio freqüência, entre a estação terrestre e a embarcação microcontrolada para a transferência do ponto destino da coleta.

Através de um conjunto de dados de possíveis pontos de coleta dos dados hidrológicos o aplicativo executa um algoritmo de otimização de rotas, reordenando os pontos de coleta, com o objetivo de minimizar a distância necessária para percorrer todos os pontos e para retornar ao ponto de partida. O algoritmo também indica o caminho ótimo que será percorrido, de acordo com a ordem de inserção dos pontos, e também o custo total da operação (figura 26).

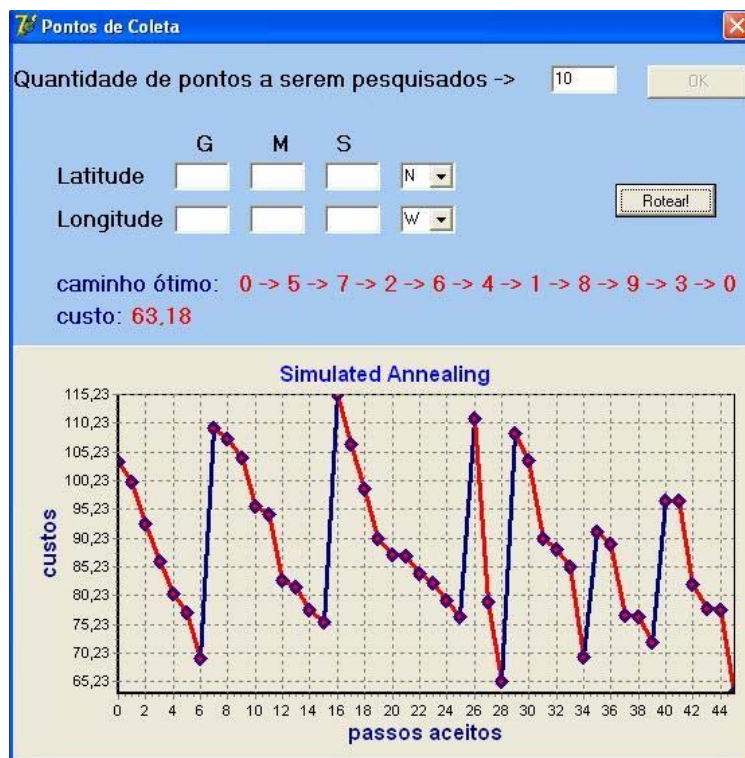


Figura 26. Entrada de dados e otimização de rota

O modelo de otimização de rotas implementado utiliza uma abordagem probabilística, chamado de método de recozimento (40). A cada passo do algoritmo, a solução prévia é sujeita a uma perturbação aleatória, de maneira diferente das implementações de algoritmos de gradiente descendente, que possuem uma grave característica: sua solução é extremamente dependente do ponto de partida. Se o ponto de partida for um vale, ele ficará preso neste vale, não havendo forma de saída de um mínimo local para alcance um mínimo global. Já a implementação usada permite mudanças de uma forma aleatória que podem aumentar a função critério, mas com o objetivo de prevenir que a convergência da solução fique em mínimos locais.

A probabilidade de a solução ocupar movimentos colina acima é controlado por um parâmetro de temperatura. O processo inicia derretendo o sistema em uma temperatura suficientemente alta para que todos os movimentos aleatórios sejam aceitos. A seguir a temperatura é abaixada vagarosamente de acordo com um

regime de resfriamento. A seqüência das temperaturas e o número de perturbações constituem a programação do recozimento, e a convergência do procedimento é fortemente relacionada com a programação de recozimento utilizada. O fluxograma da figura 27 apresenta a abordagem escolhida:

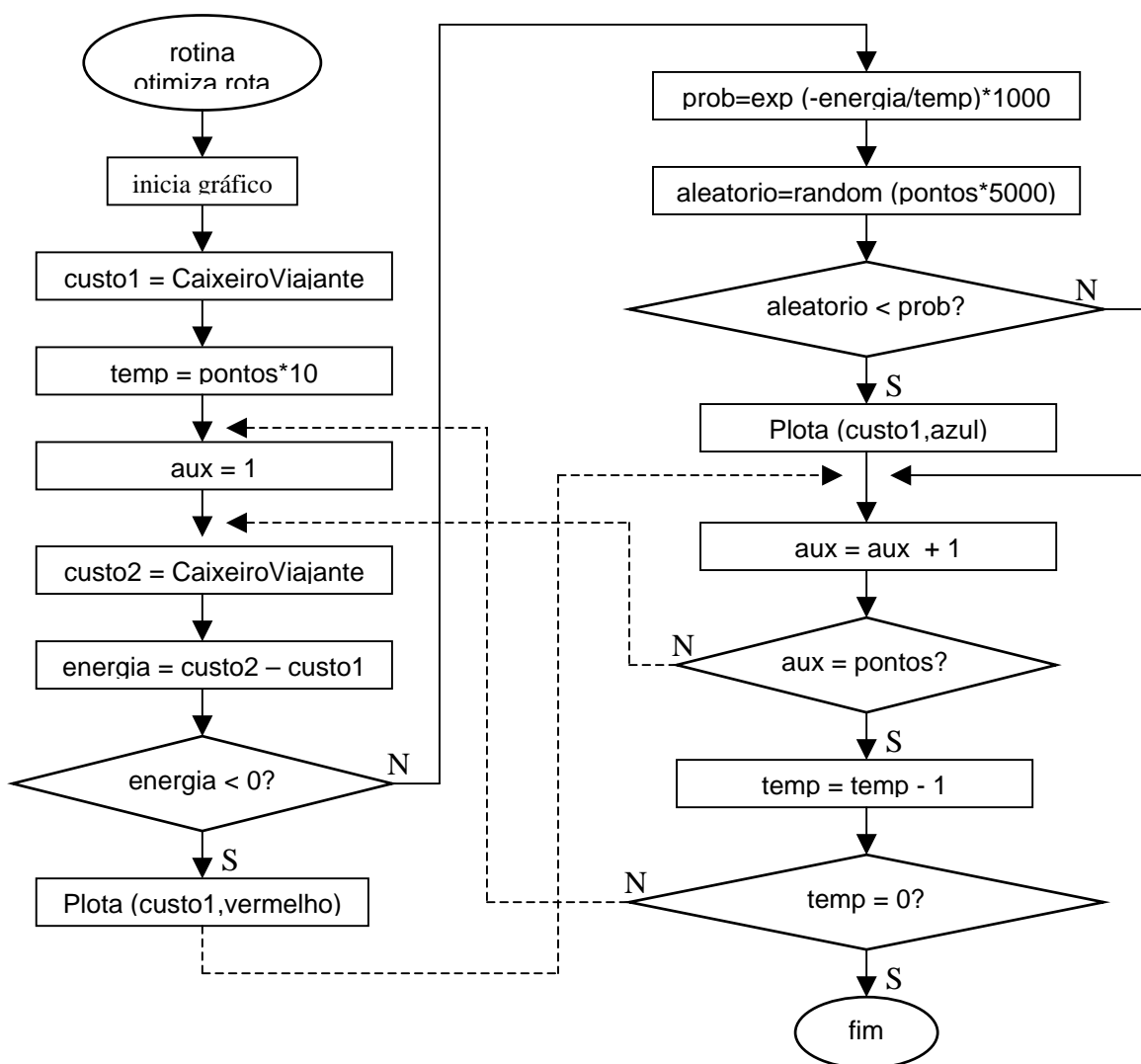


Figura 27. Fluxograma do algoritmo de otimização de rota

A rotina CaixeiroViajante sempre retorna um caminho, com partida e fim no ponto inicial da pesquisa, como solução para o problema de otimização, usando uma função aleatória que sorteia a ordem e seqüência dos pontos de coleta. Sempre que a diferença entre duas soluções (energia) for negativa ela é aceita como a melhor solução da iteração processada (regime de resfriamento), caso contrário ela também

pode ser aceita caso o cálculo da probabilidade permitir. O gráfico da figura 26 indica a quantidade de passos necessários para a convergência do método e seu custo final a cada passo. Sua principal característica é não permitir que uma solução de mínimo local seja sempre aceita por não se encontrar melhores soluções próximas, ficando o problema restrito a um vale (passos 6,15,28, etc). As linhas azuis do gráfico indicam a aceitação do uso do fator probabilístico, que permite a saída das soluções do vale. Sua principal vantagem é a velocidade de resposta (dependente da solução inicial e quantidade de passos) para uma solução quase ótima, sendo sempre aceita, já que navegar uma distância um pouco maior do que o mínimo possível não prejudica e nem invalida todos os procedimentos adotados para a navegação autônoma da embarcação.

Definido o primeiro ponto destino para a pesquisa o aplicativo envia pelo *modem* da estação terrestre um pacote de dados que contém as coordenadas (em graus, minutos e segundos) de latitude e longitude, além de um cabeçalho identificador de pacote e uma palavra soma para detecção de erros, que será analisado pela eletrônica embarcada microcontrolada. Se porventura algum dado do pacote estiver corrompido a eletrônica embarcada envia um caractere de retorno à base, pedindo a retransmissão do mesmo. A política de troca de mensagens entre os dispositivos será vista com maiores detalhes no tópico 5.3 – Eletrônica embarcada.

5.2.3 Telemetria de dados

Estabelecida a conexão com a embarcação instrumentada microcontrolada, o aplicativo recebe continuamente um fluxo de pacotes de dados com as informações referentes aos diversos sensores embarcados. O formato do pacote de dados é mostrado na figura 28:

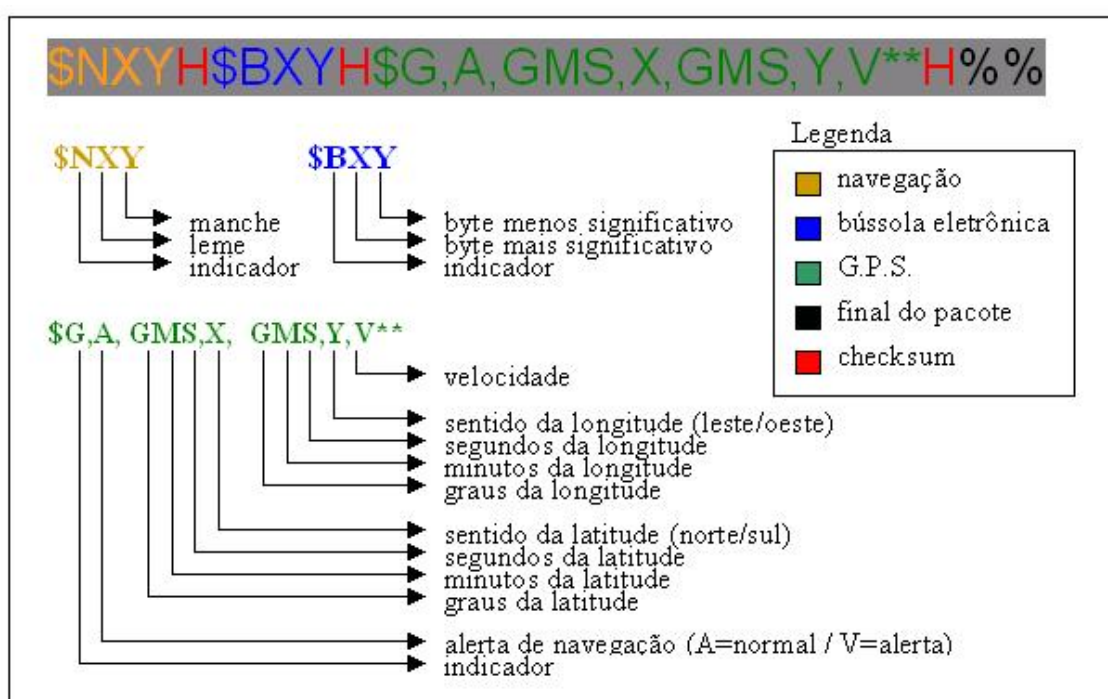


Figura 28. Pacote completo da telemetria de dados

O pacote de dados pode ser subdividido em três partes: navegação, que indica a posição mecânica do manche e leme da embarcação; bússola eletrônica, indicador de orientação e a cadeia GPS, responsável por todos os dados referentes ao georeferenciamento da embarcação como latitude, longitude, status do sensor e velocidade momentânea. Importante ressaltar que cada cadeia possui um caractere identificador de início e um caractere final de segurança (palavra soma), que indica se a cadeia está íntegra ou se algum caractere chegou corrompido. Estas

informações são essenciais para que o aplicativo saiba a posição e validade de cada cadeia dentro do pacote da telemetria, com o objetivo de mostrar na tela para o usuário todas as informações de forma clara e correta. Os dados da sonda de coleta de dados hidrológicos de cada ponto são armazenados em uma memória não volátil do sistema eletrônico embarcado, não sendo transmitidas juntamente ao pacote da telemetria em tempo on-line.

Os pacotes são transmitidos serialmente a uma taxa de 9600 bits por segundo, e os caracteres do pacote são armazenados em um *buffer* para depois serem lidos pelo aplicativo. O fluxograma da figura 29 apresenta a metodologia de recebimento do pacote de telemetria de dados via porta serial:

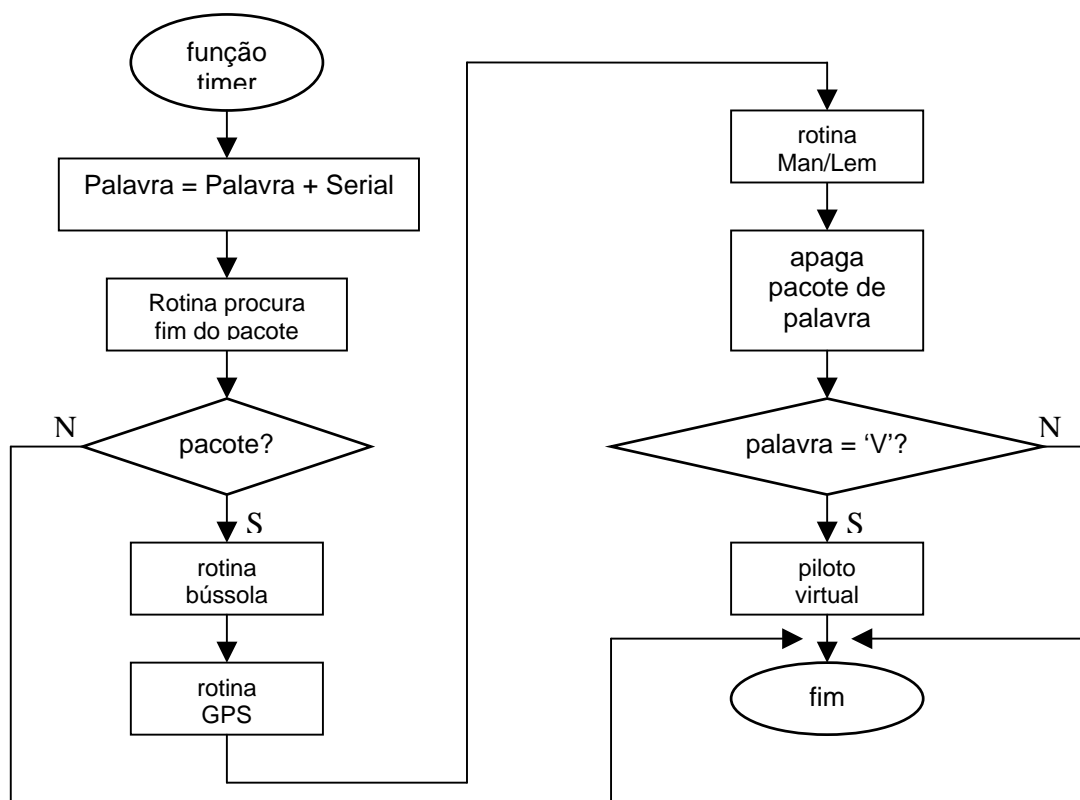


Figura 29. Fluxograma do recebimento de pacotes da telemetria de dados

A função do objeto *timer* é ativada em tempo de execução a cada 1 segundo e associa todos os caracteres do *buffer* da porta serial do microcomputador com a variável *palavra*. O sucesso da transmissão completa do pacote verifica-se quando

a função encontra o conjunto de dados ‘%%’ na formatação ASCII, dentro de toda cadeia de caracteres recebida. Se a função não encontra os identificadores de fim do pacote de dados da telemetria ela encerra seu procedimento para retornar novamente à execução após 1 segundo com o objetivo de receber o pacote de dados por completo.

Finalizado o processo de recebimento do pacote da telemetria de dados as rotinas dos sensores embarcados são seqüencialmente invocadas, o pacote é retirado da variável palavra e a função analisa se em anexo ao pacote há o caractere indicador de ativação do piloto virtual, que desvia o fluxo de programação para tratamento desta informação. O tópico seguinte trata com detalhes da implementação do piloto virtual para a navegação remota da embarcação a longas distâncias.

5.2.4 Tela principal do aplicativo para monitoração e controle da embarcação

Os pacotes de dados da telemetria são continuamente recebidos pelo aplicativo a partir do início do estabelecimento da comunicação com a embarcação instrumentada. Os resultados de todos os sensores serão somente gerados na tela a partir da confirmação da validade do sinal de recepção proveniente da cadeia *GPS*, permitindo assim o início da pesquisa. O aplicativo dispõe para o usuário a análise em tempo minimizado de uma ampla gama de informações, como pode ser visto através da figura 30.

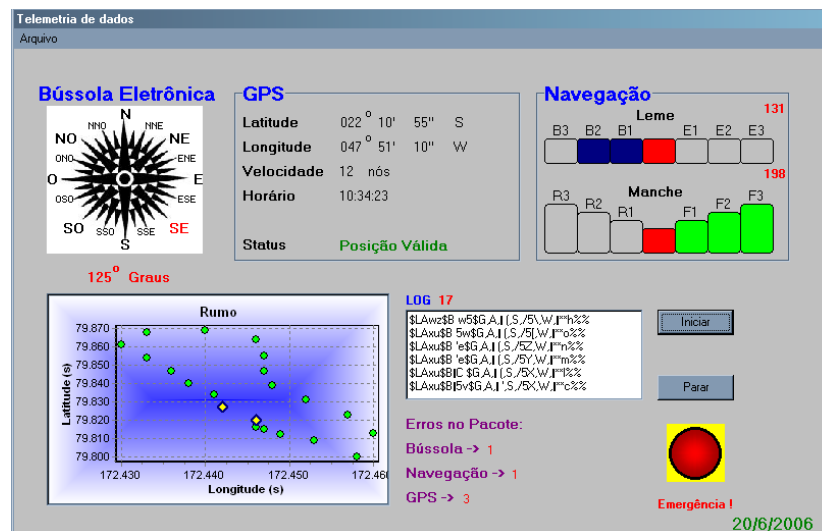


Figura 30. Tela principal do aplicativo para monitoração e controle da embarcação

As principais características e especificações de cada bloco de dados da tela principal do aplicativo são:

- **Bússola:** informa o valor em graus (0..359), destacando graficamente em vermelho a orientação na rosa dos ventos.
- **GPS:** informa as coordenadas de posição da embarcação (latitude e longitude), velocidade momentânea e o *status* do sensor (sem posição ou válido).
- **Navegação:** informa em vermelho o valor digital (0..255) do conjunto manche/leme e graficamente a posição mecânica instantânea de cada um na embarcação:
 - ✓ **Leme:**
 - bombordo1, bombordo2, bombordo3 na cor azul.
 - estibordo1, estibordo2, estibordo3 na cor verde.
 - posição central na cor vermelha.
 - ✓ **Manche:**
 - reverso1, reverso2, reverso3 na cor azul
 - frente1, frente2, frente3 na cor verde

- ponto morto na cor vermelha
- **Percurso:** informa passo a passo o posicionamento da embarcação em relação aos pontos de partida, destino e anterior.
- **Arquivo de log:** Identifica os pacotes da telemetria e possui um índice de erro para cada sensor embarcado, para análise de confiabilidade.

A ilustração da bússola eletrônica contém 16 objetos relacionados à indicação na rosa dos ventos (N,S,L,E,NE,SE, etc) e um objeto que identifica o valor em graus da orientação da embarcação. O pacote da telemetria de dados é interpretado pelo procedimento da bússola (figura 31), e os objetos são modificados conforme a interpretação dos dados.

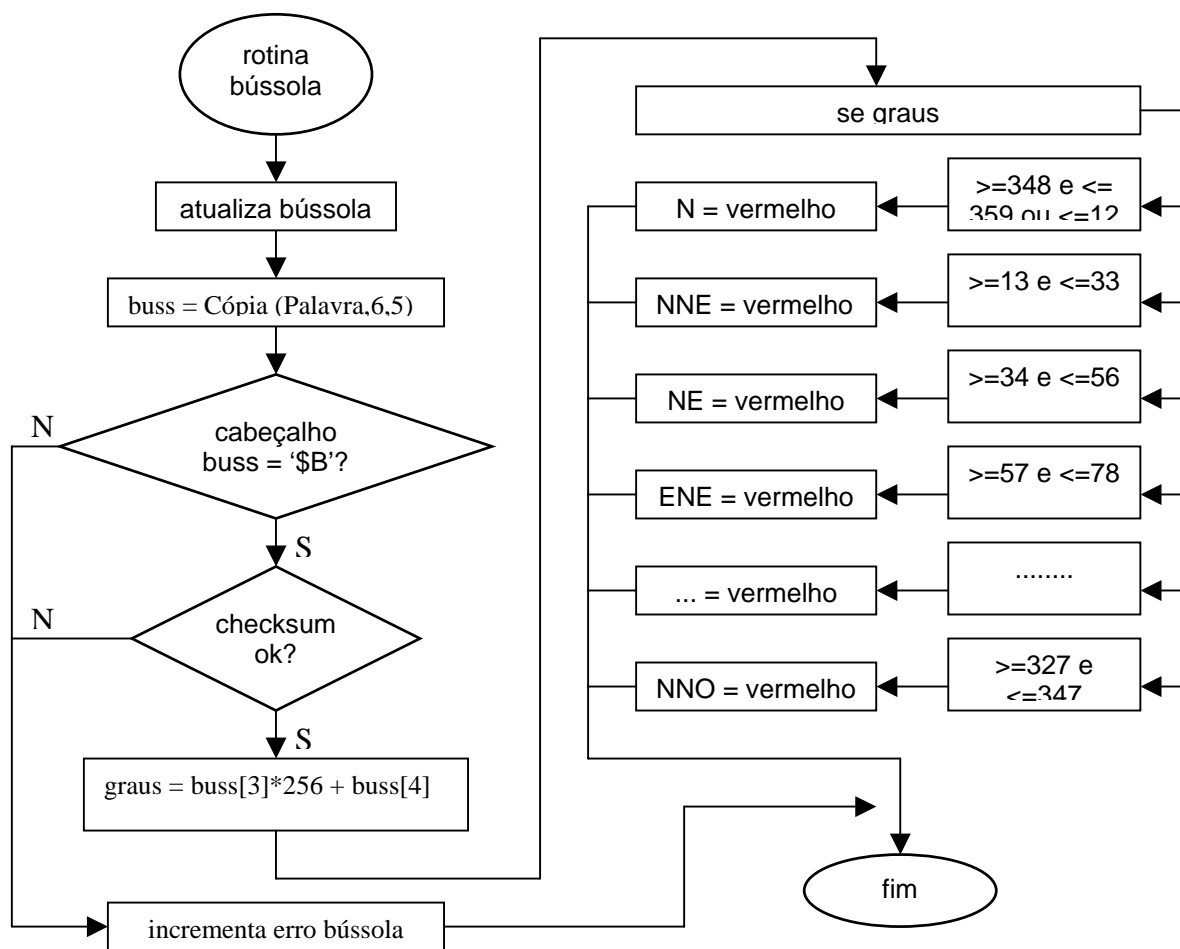


Figura 31. Fluxograma do procedimento da bússola eletrônica

O procedimento bússola caracteriza o valor do sensor embarcado através da indicação em escala decimal da variável graus e pela mudança do atributo cor do objeto correspondente à orientação na rosa dos ventos. Para definir a faixa de valores de cada orientação dividiu-se o valor máximo de leitura (360 graus) pela quantidade de objetos de orientação. No caso de erro na leitura do pacote a sub-rotina atualiza bússola indica na tela o valor em graus com os caracteres '???' na cor vermelha, e a seguir o procedimento incrementa em uma unidade no *log* de arquivo o erro do pacote, tornando portanto transparente a acusação em tempo on-line do erro de leitura do sensor da bússola.

A ilustração para os dados do sensor *GPS* segue a mesma metodologia de verificação de pacote e controle de erro, mas possui objetos com outras funcionalidades e apresentação diferente dos dados da bússola eletrônica. O fluxograma da figura 32 mostra a rotina de análise da cadeia *GPS*:

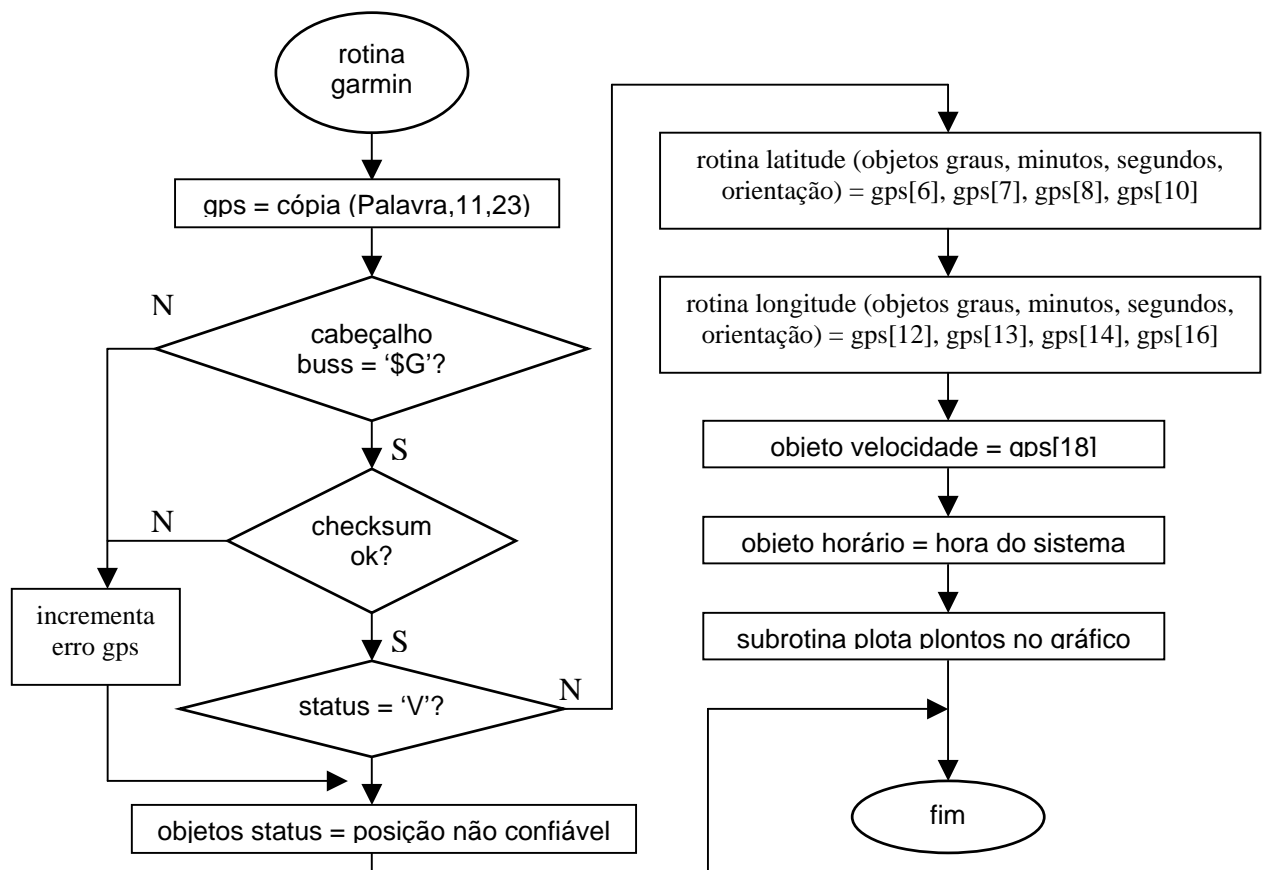


Figura 32. Fluxograma do procedimento do sensor GPS

A rotina garmin faz uma cópia das posições dos caracteres do pacote de telemetria de dados referente à cadeia do sensor *GPS*, executa filtros para o cabeçalho da cadeia e erro de pacote, mostra o status de recepção do sensor *GPS* e aplica os valores às variáveis somente se o caractere responsável em indicar a validade dos dados de georeferenciamento estiver ativado. A hora do relógio do sistema é apresentada juntamente aos dados do sensor e o ponto georeferenciado é plotado no gráfico da figura 30, indicando a posição relativa da embarcação.

A principal função da ilustração dos sensores de navegação (leme e manche) é indicar suas posições reais na embarcação. Da mesma forma que os sensores anteriores o procedimento da navegação efetua filtros para cabeçalho da cadeia e erros dos pacotes, além de indicar a posição do leme e manche pelos atributos de cor dos objetos. O fluxograma da figura 33 mostra os principais passos do procedimento de captura dos sensores de navegação:

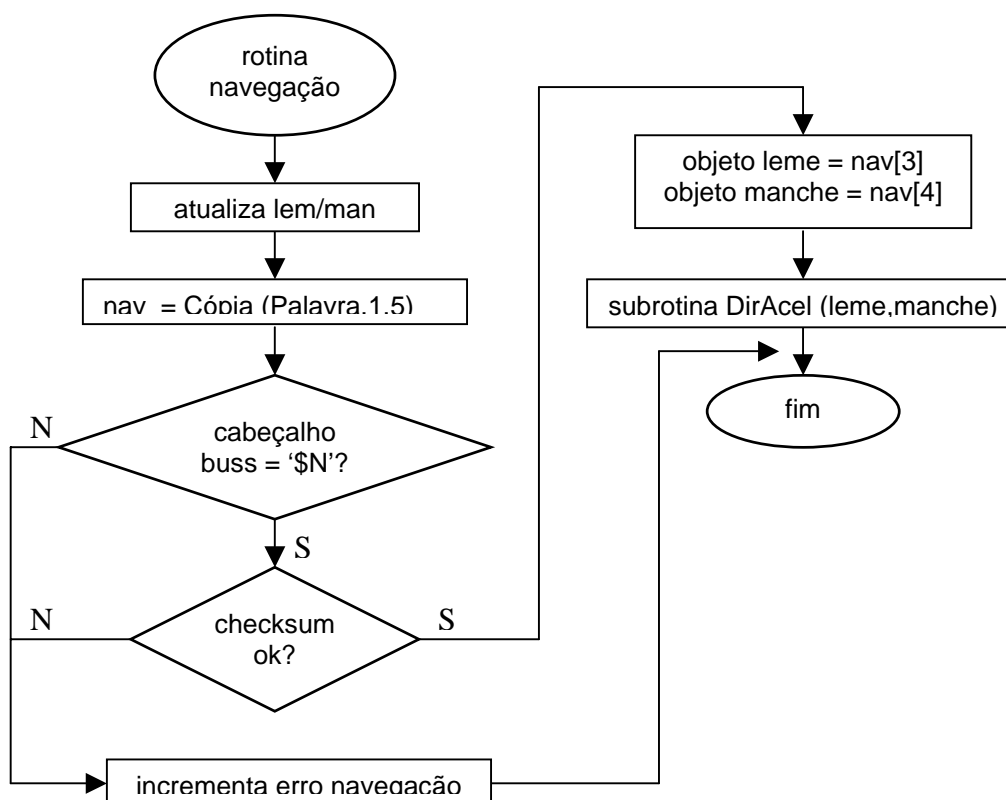


Figura 33. Fluxograma dos sensores de navegação

A subrotina *DirAcel* recebe como entrada de parâmetros os valores digitais dos sensores resistivos do manche e leme da embarcação. Ela é responsável em definir a vizinhança de valores para cada posição dos dados e atribuir cores de identificação para os objetos visuais correspondentes. Para melhora da visualização dos dados adotou-se que quando a resposta dos sensores indicar posições secundárias, como aceleração na fase 2, ou leme a estibordo 3, os atributos de cores dos objetos antecessores também serão ativados, mostrando a idéia da seqüencialidade do movimento dos motores da embarcação.

O gráfico de navegação, que faz conjunto a todos os sensores da interface principal do aplicativo informa, em tempo real, a posição da embarcação, em coordenadas de latitude e longitude, ao longo do processo de navegação. Os pontos em amarelo indicam o ponto de partida e destino da embarcação, e os pontos verdes indicam o caminho percorrido. As séries gráficas utilizadas dividem o mesmo espaço de amostragem e a escala do gráfico é sempre redimensionada a cada nova inserção, permitindo que todos os pontos de navegação da embarcação estejam presentes ao mesmo tempo na tela. A partir deste gráfico é possível descobrir, em tempo de execução, se a embarcação está convergindo para o ponto destino ou se navega em círculos.

Todos os pacotes de dados que trafegam pelo sistema embarcado (íntegros ou com erros) são transmitidos para a estação terrestre, onde são contados e mostrados no arquivo de *Log*. Para cada bloco de informação sobre os sensores (*GPS*, bússola e navegação) há um índice quantitativo dos pacotes recebidos com algum erro de informação. É de extrema importância reconhecer a falha em tempo mínimo para que se definam estratégias de recuperação do controle da embarcação, pois se por algum problema ocorrer na recepção do sinal *GPS* a navegação

autônoma não será capaz de retornar ao seu ponto de partida, além de ser uma excelente ferramenta de diagnóstico para uma análise pós-pesquisa, para validação ou não dos sensores embarcados.

5.2.5 Piloto virtual

Como método de segurança e flexibilidade de controle o aplicativo disponibiliza ao usuário a função piloto virtual, permitindo um meio de intervenção direta na navegação autônoma da embarcação (figura 34).

Uma vez invocado o processo do piloto virtual, um sinal sonoro é emitido e um pacote de dados é enviado à eletrônica embarcada informando o novo controle. A embarcação interrompe o processo de navegação e sensoriamento, prontamente colocando seu manche em ponto morto e leme ao centro, finalizando o procedimento de busca ao ponto destino e aguardando pelos comandos remotos.



Figura 34. Piloto virtual

Através desta interface tem-se o controle de todos os motores da embarcação (manche, leme, popa), com o mesmo modelo adotado para o sistema de navegação autônomo, prevendo mobilidade total da embarcação na operação remota.

Para inserir confiabilidade ao método proposto de comunicação remota propõe-se que toda ação enviada para a embarcação instrumentada o aplicativo congela a tela de comandos e aguarda pela resposta do sistema eletrônico embarcado, garantindo que a ação foi executada corretamente e que apenas uma instrução é executada por vez. O caractere responsável em indicar o início do piloto virtual vem anexado ao pacote da telemetria de dados, e é executado como uma sub-rotina do procedimento Timer da seção 5.2.3. O conjunto de comandos, associado ao sinal multimídia presente na estação terrestre, torna-se uma poderosa ferramenta capaz de retornar a embarcação ao seu ponto de origem com segurança, para análise e inspeção dos eventuais problemas ocorridos durante a fase de navegação.

A interface do Piloto Virtual é utilizada também ao início da pesquisa para posicionamento do barco a uma distância razoável da margem, garantindo espaço suficiente para as manobras de mudança de rumo quando ativada a implementação da navegação autônoma da embarcação.

5.2.6 Geração de relatórios

O aplicativo gerador de relatório tem como principais funções a descrição completa da telemetria de dados, informando ao usuário data e hora, comentário da pesquisa, mostrar toda a movimentação da embarcação ao longo do tempo na forma gráfica e histogramas para orientação da bússola, leme e manche. Todas as

informações são seqüencialmente retiradas do arquivo texto digital gerado pela pesquisa de coleta de dados em tempo de execução (seção 5.2.1). A figura 35 mostra um exemplo de relatório:

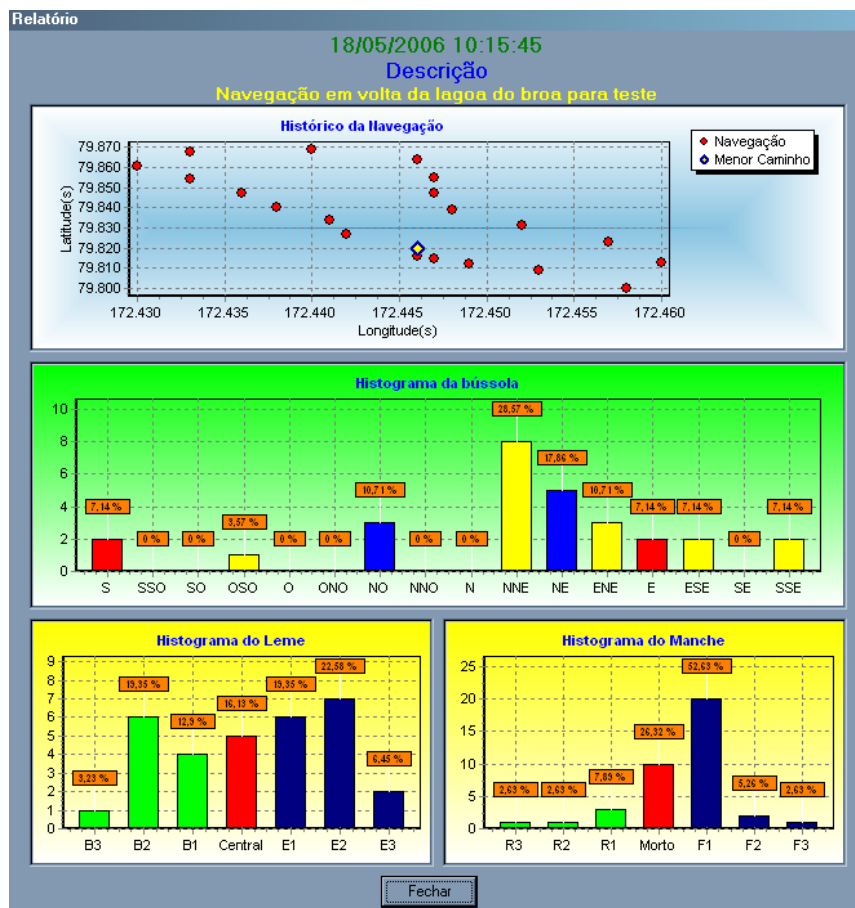


Figura 35. Relatório da telemetria de dados

Todos os gráficos são derivados do mesmo objeto *TChart*, mas o modelo de apresentação e as funções de organização dos dados diferem-se entre os sensores para uma melhor visualização dos dados.

O objeto gráfico, da série tipo ponto bidimensional, caracteriza a distribuição dos pontos através de seu método *AddXY*, de parâmetros de entrada abscissa e ordenada, toda a trajetória percorrida pela embarcação em coordenadas geográficas (pontos vermelhos). Por tratar-se de um evento de testes, no exemplo, os pontos de partida e destino são coincidentes e representados pelo ponto amarelo. O modelo de

apresentação segue as mesmas características já apresentadas para o gráfico de percurso da navegação em tempo mínimo.

Os histogramas da bússola, leme e manche, são formados por objetos de séries do tipo barras verticais que indicam a percentagem de cada um, através do método *AddBar* com parâmetros de entrada quantidade, nome e cor, no conjunto dos pacotes da telemetria de dados. No exemplo da figura 35 o histograma da bússola apresenta intensa freqüência entre as orientações Norte-Nordeste e Leste, média freqüência nas orientações Noroeste, Sul e Sul-Sudeste e baixa freqüência para as outras orientações. Pela análise dos histogramas da navegação pode-se concluir que a embarcação mudou de direção com grande freqüência, com exceção às posições extremas, e que o manche permaneceu a maior parte do tempo na posição de primeira aceleração e em ponto morto.

O relatório tem como principal objetivo medir a atividade dos sensores da bússola eletrônica e sensores de navegação, além de permitir ao usuário avaliar a navegação da embarcação durante todo o processo de pesquisa, medindo a eficácia ou não da metodologia aplicada no algoritmo de navegação autônoma para a embarcação.

5.3 Eletrônica embarcada

Vimos na seção anterior todas as informações que a eletrônica embarcada fornece, via pacote da telemetria de dados, para a estação terrestre. Para alcançar o objetivo de construção desse pacote foram necessárias programações de rotinas em linguagem de baixo nível para interfacear cada sensor separadamente, gerar

políticas de troca de mensagens entre os sistemas microcontrolados e a estação terrestre, além de implementar técnicas de otimização contra efeitos de travamento e tempo de resposta muito lento. Os tópicos subseqüentes especificarão todas as metodologias aplicadas na eletrônica embarcada com o objetivo final de permitir o fluxo de informações em tempo mínimo para a estação terrestre, para efeito de diagnóstico e acompanhamento das missões pelo operador do sistema.

5.3.1 Padrão de comunicação e política de troca de mensagens

A grande maioria das comunicações dos sensores com os sistemas microcontrolados, troca de mensagens entre esses dispositivos e a estação base realizam-se através de comunicações seriais. Na seqüência temos a figura 36 que mostra todos os canais seriais envolvidos na instrumentação embarcada.

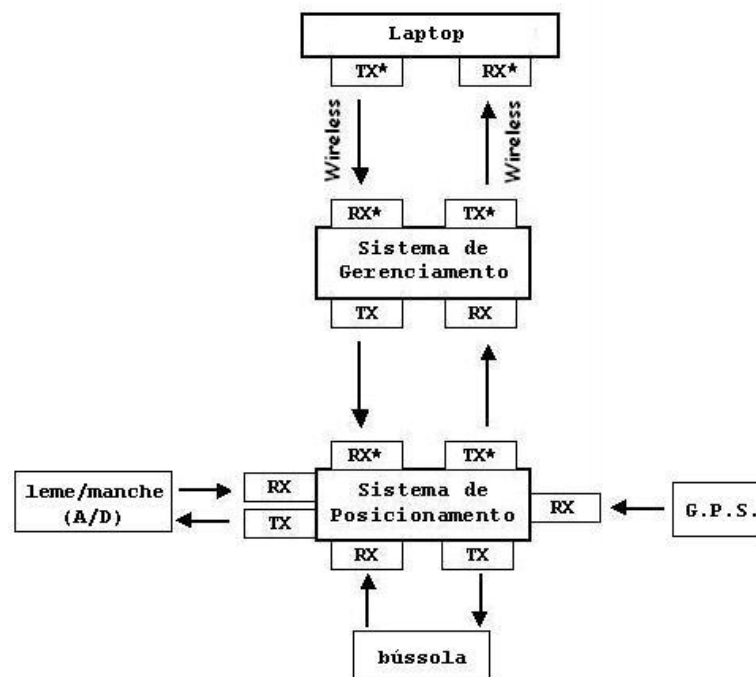


Figura 36. Canais seriais de todo o *hardware* embarcado

Os microcontroladores da família Atmel, sendo o modelo AT89C2051 para o sistema de posicionamento e o modelo AT89C52 para o sistema de gerenciamento, possuem apenas um canal serial half-duplex próprio, identificados na figura com o caractere asterisco. Estes foram alocados estrategicamente nos pontos mais críticos do sistema, propiciando uma rápida intervenção da estação terrestre sobre a eletrônica embarcada, em eventual necessidade. Seu uso é bastante simples, precisando somente uma configuração de alguns parâmetros para que toda a sincronização e o recebimento dos caracteres fiquem automatizados. Pela escassez de portas seriais houve a necessidade de implementação de rotinas específicas para complementar a comunicação com os sensores e entre os sistemas microcontrolados.

Os canais seriais entre os sistemas microcontrolados e o microcomputador da estação terrestre atuam no modo assíncrono com taxa de velocidade de 9600bps. A rotina do canal receptor *GPS*, que recebe vários pacotes de dados, foi desenvolvida também no modo assíncrono e com taxa de velocidade de 4800bps, sendo esta restrição de velocidade imposta pelo *hardware* do equipamento. Já os canais seriais do conversor analógico/digital do manche e leme e os canais seriais da bússola são todos síncronos, pois cada equipamento recebe como ponto de partida um byte de sincronismo e configuração, retornando o dado na formatação desejada. Em tópicos adiante veremos maiores detalhes de como as rotinas foram implementadas e quais são as palavras de controle dos canais síncronos de cada sensor.

O fluxo de informações entre os sistemas de gerenciamento e posicionamento é intenso, e como esses canais de comunicação são controlados por rotinas e não pelo *hardware*, não pode ser descartada a probabilidade de eventuais problemas de sincronização, geradores de erros em pacotes. Para resolver a questão

implementou-se uma política de troca de mensagens. A figura 37 aborda a implementação desta solução:

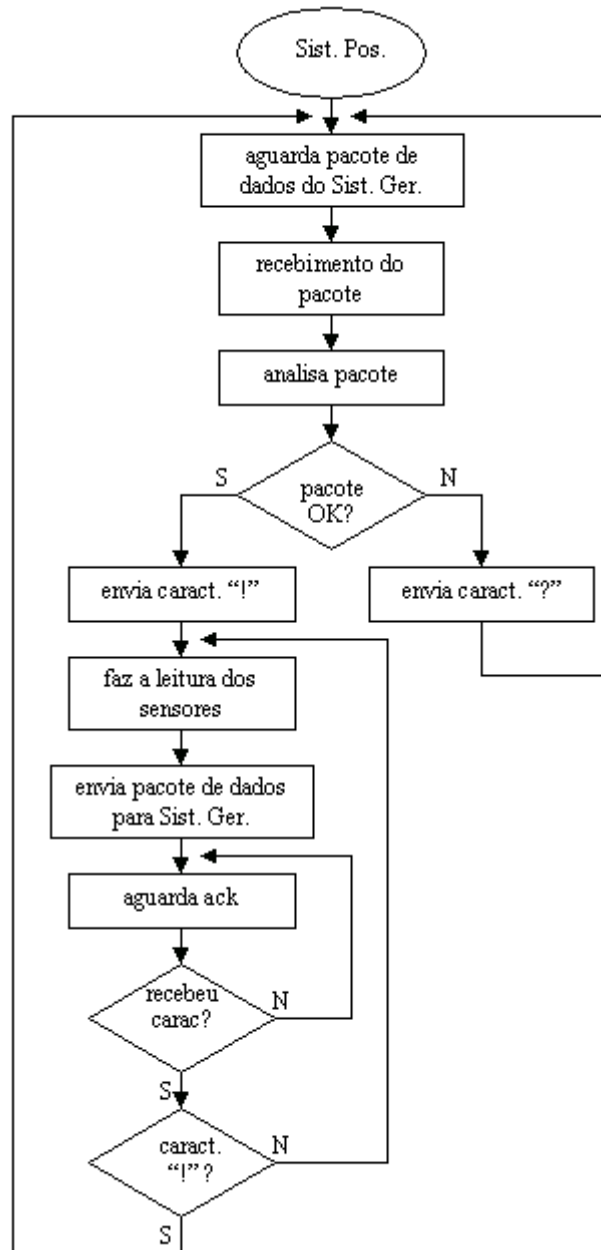


Figura 37. Fluxograma de sincronização entre sistemas microcontrolados

Para toda a transmissão de pacotes entre sistemas microcontrolados o emissor faz a transferência dos dados, sendo o último caractere responsável pelo controle de erro (palavra soma), e aguarda um caractere de retorno, com a indicação no pacote se ele está corrompido ou se ocorreu falta de sincronização na comunicação. Nesse caso ocorre a retransmissão, e a operação de tratamento contra erros e falta de

sincronização se reinicia. Essa abordagem é substancial para a implementação proposta, que tem como principal objetivo o livre fluxo de informações até a estação terrestre, responsável em receber todo o pacote da telemetria de dados no menor tempo possível e apresentar as repostas dos sensores visualmente ao operador do sistema.

A arquitetura hierárquica implementada, dividindo o sistema em níveis, assume que o sistema de gerenciamento é o responsável em delegar tarefas ao nível inferior (sistema de posicionamento), sempre recebendo um retorno das informações requeridas. Para implementar a comunicação de dados entre os níveis foi estabelecida uma formatação de dados para troca de mensagens, garantindo a padronização de transferência de informações. Nesta padronização são sempre transferidos 5 caracteres, onde o caractere '\$' identifica o início do conjunto de dados. O próximo caractere simboliza o comando que está sendo dado, seguido de dois caracteres de controle (dependendo da ação), e por fim o verificador de erros. A tabela 8 exemplifica cada ação dos pacotes enviados pelo sistema de gerenciamento ao nível inferior:

Seqüência	Função
\$LzzH	requisição de leitura de todos os sensores
\$ALMH	mudança de rumo da embarcação L = índice pra leme M = índice para manche
\$RzzH	resetar embarcação
\$IzzH	desligar motor de popa
\$SzzH	ligar motor de popa
H – palavra soma dos caracteres, com exceção ao caractere '\$'	
z – valor numérico sem significado	

A primeira seqüência requisita o pacote da telemetria de dados completo para o sistema de posicionamento, ou seja, o sistema de gerenciamento receberá os 33 bytes ilustrados na figura 28. A segunda seqüência delega a ordem de mudança de

posição do leme e manche da embarcação. Os caracteres de controle ‘L’ e ‘M’ recebem índices como valor, que são transformados depois pelo sistema de posicionamento em valores digitais reais, que correspondem a uma posição física de cada um dos atuadores, alterando a posição da direção do leme e o controle de aceleração do manche. Posteriormente um novo pacote de dados é enviado ao sistema de gerenciamento informando os novos valores digitais destes sensores.

A terceira, quarta e quinta seqüência são usadas principalmente para início de navegação e piloto virtual. Antes de iniciar a navegação, por medida de segurança, o sistema de gerenciamento envia a seqüência para reiniciar a embarcação, ou seja, centralizar a direção do leme e colocar o acelerador do manche em ponto morto, enviando em seguida a seqüência para ligar o motor da embarcação, dando início ao processo de navegação da plataforma autônoma. Já o piloto virtual utiliza-se dessas três seqüências, além das que comandam ações na mudança de rumo da navegação, permitindo o controle total de todos os motores da embarcação.

5.3.2 Comunicação com sensores da embarcação

5.3.2.1 Sensores do leme e manche

Os dados do posicionamento da embarcação são obtidos através da conversão dos sinais analógicos obtidos dos sensores resistivos presentes no manche e leme da embarcação para valores digitais. Para realizar esta conversão utilizou-se o circuito integrado conversor analógico digital ADS7841 da Texas Instruments, com 4

canais independentes e até 12 *bits* de resolução da amostra. Seu funcionamento depende do envio de uma palavra de controle de configuração para cada amostra de conversão. A tabela 9 mostra o byte de configuração:

Tabela 9 - Pinagem do conversor A/D

Bit	Nome	Descrição
7	S	Start Bit
6-4	A2-A0	Seletor de canais
3	MODE	Resolução da conversão
2	SGL/DIF	Multiplexador de canais
1-0	PD1-PD0	Power-Down

Definiram-se as palavras de controle 97H para configurar a conversão do sensor do leme e 0DFH para a conversão do sensor do manche. O significado completo da palavra de controle é mostrado a seguir:

Leme

1	0	0	1	0	1	1	1
MSB							LSB

- Start *bit* no décimo primeiro clique de máquina.
- Primeiro canal de conversão (CH0).
- Resolução de 8 *bits* (em *hardware*).
- Sem multiplexar canais.
- Sem economia de energia - dispositivo sempre ligado.

Manche

1	1	0	1	1	1	1	1
MSB							LSB

- Start *bit* no décimo primeiro clique de máquina.
- Segundo canal de conversão (CH1).
- Resolução de 8 *bits* (em *hardware*).
- Sem multiplexar canais.
- Sem economia de energia - dispositivo sempre ligado.

Tanto para enviar a palavra de configuração para retirada da amostra quanto o recebimento do byte digital a comunicação é sempre síncrona, ou seja, gera-se um pulso de *clock* para o conversor para cada *bit* de configuração, até a transferência do byte por completo. Para receber o dado convertido faz-se o mesmo procedimento, recebendo, a cada pulso de *clock*, os *bits* da amostra.

Para visualizar os dados digitais dos sensores resistivos do leme e manche construiu-se um aplicativo visual (figura 35), que envia um comando para o sistema de gerenciamento, que repassa a informação de leitura dos sensores para o sistema de posicionamento, retornando um pacote de dados, até o aplicativo visual.



Figura 38. Aplicativo de testes dos valores digitais de leme e manche

O fluxograma da figura 39 mostra a evolução das rotinas, a comunicação entre os aplicativos embarcados e o aplicativo visual para a captura dos dados digitais do leme e manche para controle da embarcação.

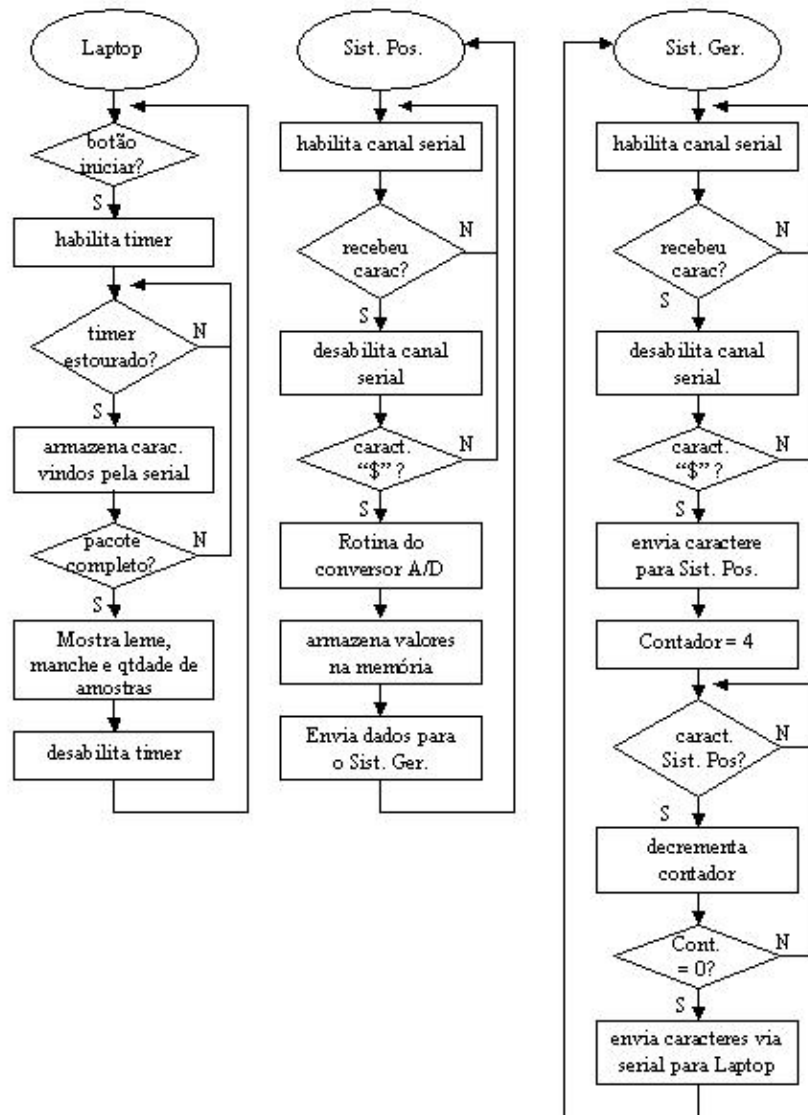


Figura 39. Fluxograma dos aplicativos para captura dos dados de leme e manche

Utilizando sensores resistivos para testes de bancada observou-se através do aplicativo de testes a linearidade dos valores para toda faixa digital correspondente a 1 byte, e a diferença de saída de medição, para uma mesma posição resistiva, não ultrapassou 1 unidade para ambas vizinhanças, validando o uso deste tipo de sensor para controle e medição de posicionamento da embarcação.

5.3.2.1.1 Acoplamento dos sensores e valores digitais reais

Para o sensor do leme adaptou-se um sensor resistivo blindado multivoltas no centro da roldana responsável em transferir a carga do motor para a direção, e para o manche adaptou-se um sensor resistivo com a mesma metodologia (figura 40). Os primeiros resultados das amostras colhidas em diversas posições para ambos identificou a não linearidade dos valores digitais no processo da mudança de direção e aceleração, não possibilitando a simples divisão igualitária de um byte para as faixas de leitura e controle das posições.



Figura 40. Sensores resistivos do leme e manche

Nos testes reais de aceleração e inversão de rotação do motor de popa viu-se também uma não linearidade da velocidade adquirida pela embarcação, ocasionada pela mecânica adaptada para controle de velocidade indireta, diferente de motores de popa convencionais, onde a aceleração é realizada por uma manopla acoplada diretamente ao motor. Nesse caso as posições reais e possíveis para a manobra eletrônica da embarcação tem como modelo a experiência de navegação adquirida

ao longo dos testes na lagoa. A tabela 10 apresenta os valores digitais para cada posição do manche e leme:

Tabela 10 - Valores digitais das posições de manche e leme

Manche		Leme	
valor digital	aceleração/potência	valor digital	direção
175	Popa/máximo	150	bombordo 3
164	Popa/médio	125	bombordo 2
150	popa/mínimo	100	bombordo 1
130	ponto morto	71	central
110	proa/mínimo	58	estibordo 1
93	proa/médio	45	estibordo 2
80	proa/máximo	34	estibordo 3

Os valores digitais são os utilizados como parâmetros pela rotina do sistema de posicionamento para movimentar o leme e o manche da aceleração. Os estados atuais dos sensores são comparados com a referência da tabela, definindo o sentido de giro e acionando os motores, até que a estado dos sensores seja maior ou menor do que a referência, dependendo da lógica aplicada para cada controle do motor.

5.3.2.1.2 Testes e calibração

Para simulação da movimentação dos motores do manche e leme construiu-se um aplicativo em alto nível (figura 41), com o objetivo de enviar um pacote de dados à eletrônica embarcada com as seguintes opções de decisão:

- Resetar: posiciona o leme na posição central e manche em ponto morto.
- Atuar: o usuário define manualmente a posição de leme e manche.
- Leitura: somente a leitura do estado dos sensores de navegação.

- Emergência: correspondem à opção resetar e desligar o motor de popa.

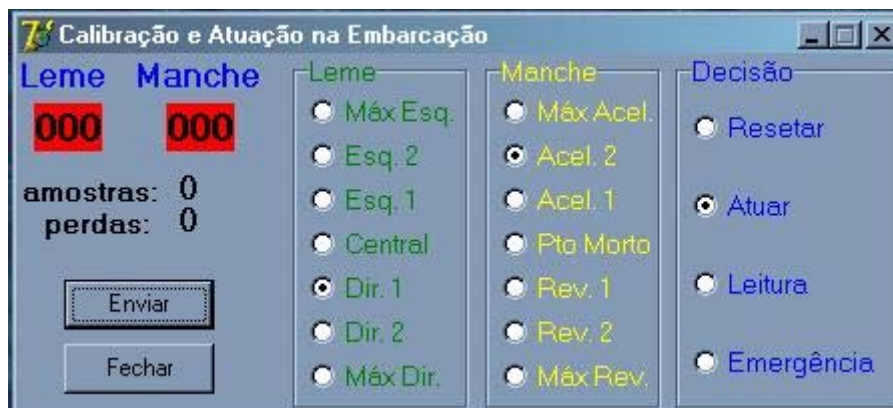


Figura 41. Calibração e atuação na embarcação

O primeiro procedimento é a escolha da decisão a ser tomada e na seqüência o pacote de dados é enviado pelo *modem* serial para o sistema de gerenciamento, com a responsabilidade de ser somente um intermediário para a ação de controle, transferindo o pacote ao sistema de posicionamento, que é onde estão configuradas as entradas dos sensores do manche e leme e também a interface para a placa de relês que controlam a posição dos motores elétricos.

Se a opção for somente leitura então o aplicativo desabilita as opções de escolha de ação, e como resposta recebe um pacote que identifica o valor digital de leme e manche, além do caractere de integridade (palavra soma). Se o pacote está íntegro então o programa indica nos objetos seletores de leme e manche suas posições, além de mostrar que o pacote foi recebido com sucesso, incrementado o valor na tela, e no caso de fracasso a opção perdas é incrementada.

A principal função deste aplicativo para teste e simulação é o controle direto dos motores do leme e manche. Através dessa opção o usuário define, através dos botões seletores de cada objeto, a nova posição da embarcação. Se, por exemplo, o usuário escolher “Esq. 2” para leme e “Acel. 1” o aplicativo envia um pacote de dados ao *hardware* embarcado, identificando neste a natureza da ação e o código

para cada nova posição. O sistema de posicionamento, ao receber esta ordem, deve acionar os motores e, independente da posição anterior mecânica, alcançar o novo objetivo. A decisão Resetar é a correspondente das opções “Central” para leme e “Pto. Morto” para o manche e a operação emergência tem o mesmo objetivo, adicionando-se a tarefa de interromper o funcionamento do motor de popa.

Para todas as opções o aplicativo de teste recebe de volta um pacote que indica os valores digitais das novas posições e mostra-os nas caixas de opção para cada sensor. Os sucessivos testes na embarcação mostraram que, para cada posição, deve haver uma faixa de valores digitais que deve ser respeitada como vizinhança, para que não ocorra principalmente no aplicativo a falsa anotação de uma posição nas caixas seletoras dos sensores, confundindo o usuário na análise e no uso do piloto virtual como ferramenta de navegação e emergência. O problema ocorre por causa da grande inércia dos motores, que mesmo desligados no momento exato, modificam o curso final dos atuadores, alterando o resultado da leitura final.

A metodologia aplicada para eliminar a falsa interpretação de resultados foi efetuar testes sucessivos, para cada posição, dos valores máximos e mínimos de vizinhança da leitura dos sensores. Para garantir que a maior inércia do motor atuava no sistema a posição inicial no teste dos motores era o mais distante possível da posição objetivo, garantindo assim a maior velocidade aos motores, e conseqüentemente a maior inércia. Como exemplo, para a posição “Dir. 1” de leme, a posição inicial para cálculo da vizinhança máxima é “Máx. Dir”, e para cálculo da vizinhança mínima, a posição inicial é “Max. Esq”. Através deste filtro de análise o aplicativo de teste e calibração passou a mostrar acertadamente todas as posições de leme e manche nas suas respectivas caixas seletoras.

5.3.2.2 Comunicação com a bússola eletrônica

A bússola eletrônica é um equipamento essencial para o posicionamento da embarcação, pois através de sua leitura a mesma corrigirá seu rumo com precisão, otimizando o processo de pesquisa. Sua grande vantagem é de ser de natureza magnética, diferentemente da bússola incorporada ao *GPS*, que calcula a orientação através da diferença entre dois pontos distintos de coordenadas globais. Portanto o uso da bússola eletrônica torna-se altamente justificável.

A comunicação com o dispositivo realiza-se sempre na forma serial síncrona, tanto para sua calibração como para a saída de dados. Os pinos de interface da bússola com o sistema de posicionamento são vistos na figura 42:

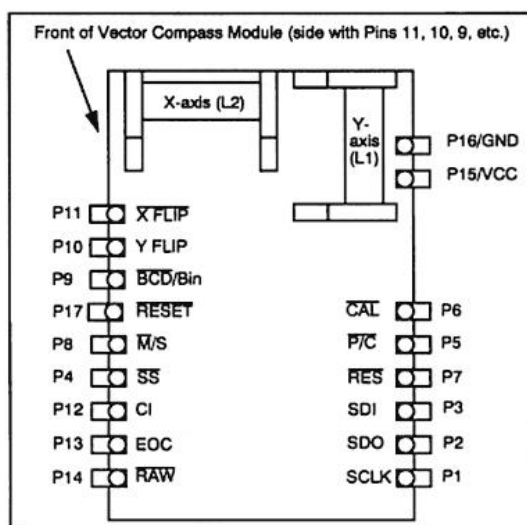


Figura 42. Pinagem da bússola eletrônica

As funções dos pinos utilizados para a calibração e saída de dados são:

- SS: seleção de operação para modo escravo.
- SCLK: *clock*.
- P/C: efetua pedido de leitura.

- CAL: calibração da bússola.
- CI: indicador de calibração.
- EOC: indicador de fim de conversão.
- SDO: saída de dados seriais.

Ao iniciar o *hardware* da embarcação, o sistema de posicionamento executa a rotina de calibração inicial da bússola eletrônica (figura 43), essencial para que os dados sejam sempre consistentes durante a navegação da embarcação.

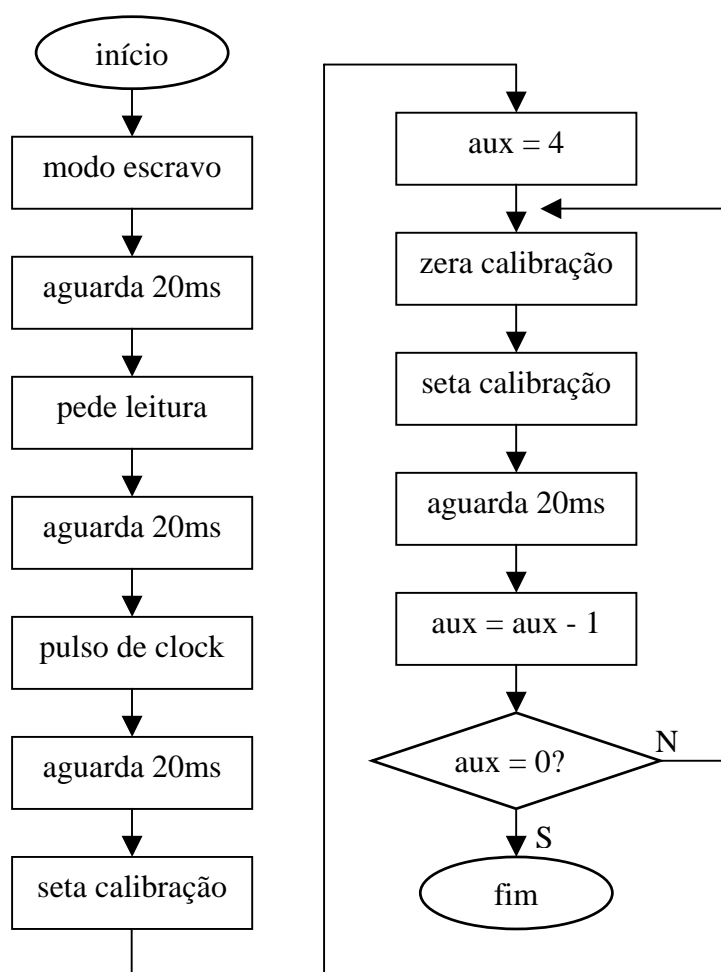
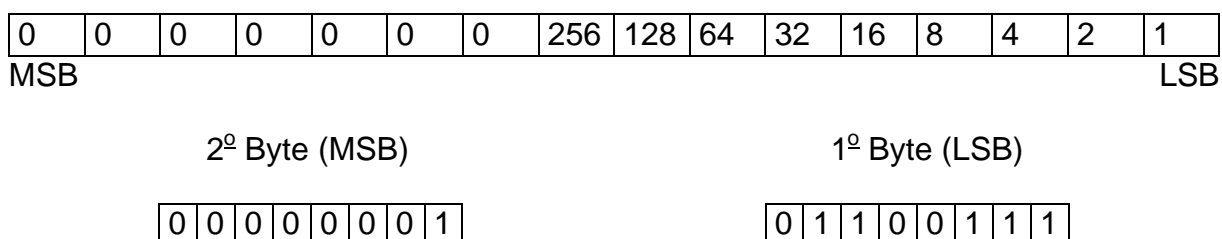


Figura 43. Rotina de calibração da bússola eletrônica

As temporizações entre os blocos são necessárias para obedecer aos critérios do diagrama de calibração discutido no manual do usuário do equipamento. A configuração da operação de saída de dados é sempre escrava, ou seja, a saída de dados da bússola é sempre ativada externamente.

Para o microcontrolador receber o valor de orientação da bússola eletrônica envia-se um sinal de *clock* para cada *bit* da amostra, com saída deste sempre pelo pino SDO. Efetuando-se a operação por 12 vezes consecutivas temos o ciclo completo de leitura. A bússola disponibiliza dois tipos de formatações de saída (BCD e binária), representadas por uma faixa de valores que começa em 0 decimal e vai até 359 da escala decimal, indicando a orientação na rosa dos ventos. A representação escolhida foi pela saída binária, para uso do microcontrolador sem conversões adicionais, mas sendo necessárias 2 posições de memória para armazenar todo o dado. Para a apresentação do dado na escala decimal soma-se então o valor dos índices representativo de cada um dos 12 *bits* da amostra conforme o exemplo abaixo:



$$\text{Graus} = 1 + 2 + 4 + 32 + 64 + 256 = 359 \text{ (valor máximo)}$$

(2)

5.3.2.2.1 Teste de validação

Para o teste de validação das amostras dos dados da bússola eletrônica utilizou-se, como artifício de comparação, uma imagem circular impressa graduada até 360^o (figura 44). Num ponto qualquer do encapsulamento da bússola marcou-se

um referencial e a mesma sofreu seguidas rotações em torno de seu eixo e acima da imagem para efeito de comparação.

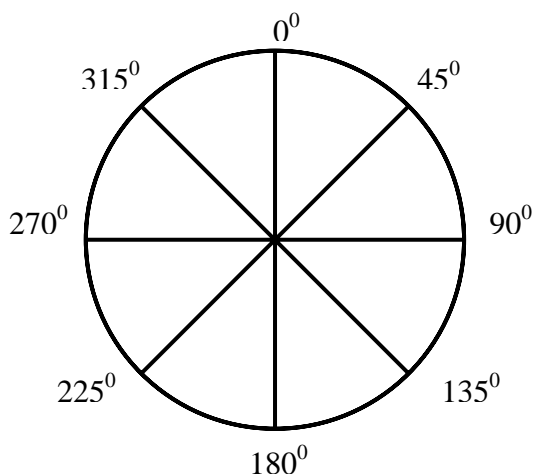


Figura 44. Imagem circular usada para validação dos dados da bússola

Para as leituras seqüenciais sobre um mesmo ponto observa-se que as medidas ficaram dentro das especificações descritas na seção 4.2.1, e para as leituras comparadas entre diversos referenciais do encapsulamento da bússola e a imagem circular impressa graduada os resultados também ficaram dentro das especificações do fabricante, validando o uso desta como ferramenta de precisão para ajudar no processo de navegação da embarcação.

5.3.2.2.2 Cálculo matemático do giro da bússola

O sistema de gerenciamento, responsável pelos critérios de navegação da embarcação, utiliza a orientação da bússola eletrônica para manter e alterar a trajetória da embarcação com precisão. Dependendo da situação pode-se mudar o rumo por diferentes valores. Se a embarcação não converge ao ponto destino então uma

grande mudança de rumo deve ser adotada, diferente da situação onde a embarcação converge lentamente e em apenas uma coordenada, logo para otimizar o processo da pesquisa adota-se correções mais suaves na orientação da mesma. Todas estas situações exigem como controle uma rotina dedicada de cálculos matemáticos, com o valor lido da bússola, que ocupa duas posições distintas da memória de dados do microcontrolador 8 *bits*, pois seu valor tem como fundo de escala o número 359 decimal e apresenta contagem cíclica. O fluxograma da figura 45 mostra a rotina em baixo nível de programação, que efetua cálculos matemáticos para o valor da bússola eletrônica:

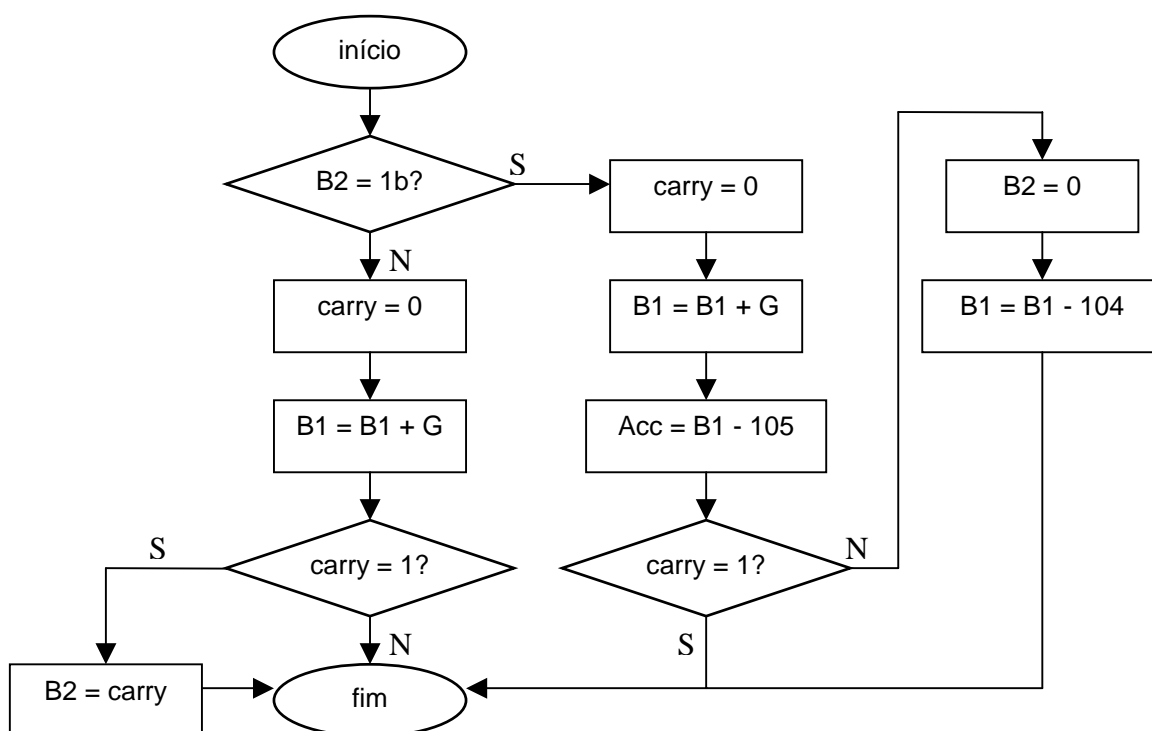


Figura 45. Fluxograma do cálculo matemático para a bússola eletrônica

Considerações:

- B2 = byte mais significativo da bússola eletrônica
- B1 = byte menos significativo da bússola eletrônica
- Acc = variável auxiliar (acumulador)

- *carry* = indica todo estouro de cálculo de um byte (*overflow e underflow*)
- *G* = variável numérica indicativa da quantidade de mudança de rumo
- *b* = escala binária

Esta rotina tem como função somar a variação de rumo, em dois bytes, ao valor atual da bússola, para determinar mudanças de trajetória da embarcação. O uso do *carry* é a ferramenta de teste para achar a solução dentro do círculo de valores pretendidos, já que sua sinalização acusa sempre o estouro do cálculo matemático. Como exemplo, a variável B1 tem como valor máximo a centena 104, caso a variável B2 assuma valor diferente de zero ($104B1 + 255B2 = 359$), ou B1 assume o byte completo se B2 for igual a zero (orientação menor que 255 graus). Com o uso desta rotina o sistema de gerenciamento aplica, como critério de comparação para mudança de orientação da embarcação, a rotina de cálculos numéricos apresentada com eficácia.

5.3.2.3 Comunicação com sensor GPS

Com as informações das especificações técnicas da interface do sensor *GPS*, e com o conhecimento detalhado da sentença *NMEA*, decidiu-se que a principal função da rotina que captura os dados do sensor é a de reconhecer, dentro de todo o pacote recebido, a sentença com cabeçalho *\$GPRMC*, detentora de todas as informações necessárias para navegação e também indicadora do nível de recepção dos satélites pelo *GPS* (tabela 5). Apesar de alguns campos desta sentença não serem de utilidade para a implementação proposta todo o pacote de dados foi recebido pela rotina, já que a sentença carrega em suas últimas posições o

caractere verificador palavra soma, que é o único método de garantia e confiabilidade que o sistema tem pra validar o recebimento do pacote do sensor *GPS* com sucesso.

Conhecendo o padrão de comunicação, formatação e quantidade de caracteres construiu-se a rotina que recebe a sentença de cabeçalho \$GPRMC do sensor *GPS* e a armazena na memória de dados do microcontrolador. O fluxograma da figura 46 apresenta os principais passos efetivados para o cumprimento do objetivo proposto:

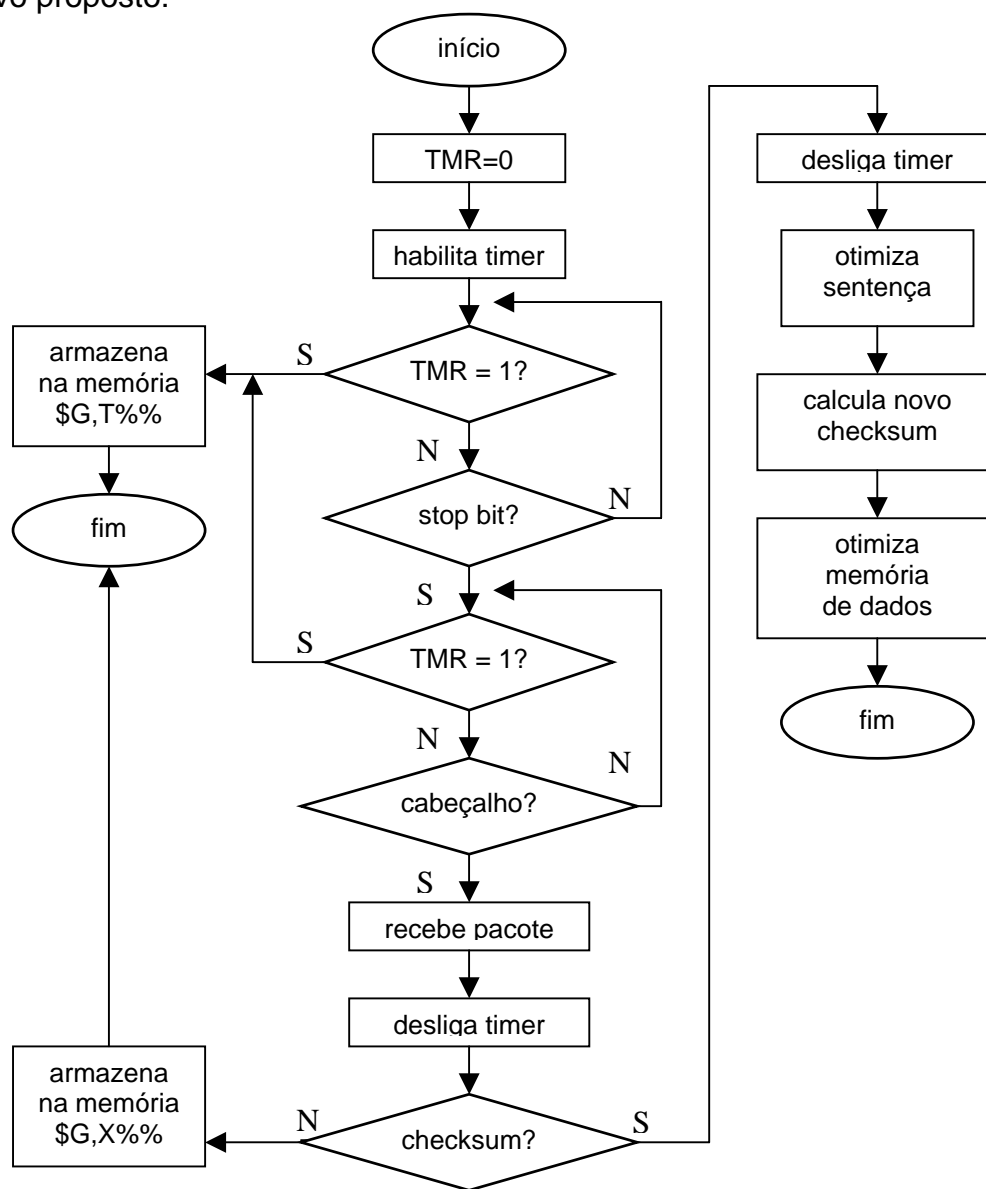


Figura 46. Fluxograma da rotina que extrai sentença do sensor *GPS*

A canal serial de comunicação com o *GPS* é do tipo assíncrono, ou seja, com transmissão individual dos caracteres, delimitados pelo *start bit* e *stop bit*. A rotina procura dentro de todas sentenças *NMEA* disponíveis o cabeçalho do pacote que se deseja adquirir. Se esses caracteres, por falha do módulo *GPS*, não forem encontrados, a rotina entra em *loop* infinito conhecido como *deadlock*, travando o sistema. Além deste problema há a probabilidade de ocorrer falta de alimentação para o sensor, produzindo também uma falha crítica no sistema. Para evitar imprevistos utilizou-se a função de interrupção por tempo do microcontrolador. Com essa função ativada o pacote de dados com cabeçalho \$GPRMC tem um tempo determinado para ser capturado do módulo *GPS*. Estourado o tempo há a garantia da ocorrência de uma falha grave, mas o sistema estando preparado ele tem a capacidade de produzir um pacote de dados e armazená-lo na memória indicando o erro. O sistema de gerenciamento reconhece a falha no pacote, não o utilizando para o cálculo de navegação, mas adiciona-o ao pacote da telemetria de dados que vai até a estação terrestre, onde o usuário tem como alternativa de solução pilotar a embarcação virtualmente até um ponto seguro para avaliação dos danos ao *hardware*.

Os 68 caracteres do pacote de dados do *GPS* são posicionados seqüencialmente na memória de dados do microcontrolador, a partir da posição 10H até a posição 63H, excetuando-se as 16 posições endereçadas *bit a bit*, que são reservadas para uso interno (figura 47). A quantidade de memória gasta no armazenamento corresponde a 58% do total das posições válidas, um valor extremamente oneroso, que evita a criação de variáveis de ambiente para controle e atuação de dispositivos através das entradas/saídas e canais seriais.

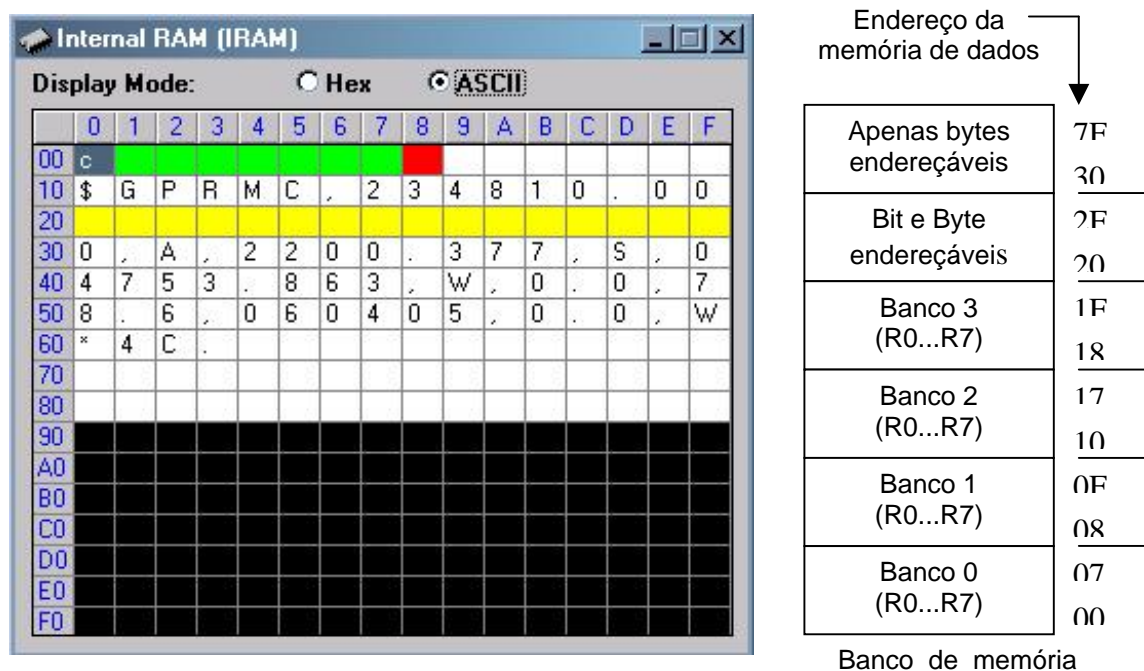


Figura 47. Memória e endereçamento de dados do microcontrolador

O sistema de posicionamento não sofrerá falta de memória para os dados de geração do sistema de controle, pois sua maior função na implementação da arquitetura hierárquica é especificamente criar o pacote da telemetria de dados. O grande gargalo de memória encontra-se no sistema de gerenciamento, que tem como responsabilidade utilizar esses dados da memória para efetuar cálculos para a navegação e requisitar, através de seus canais seriais, ações de controle sobre o sistema de posicionamento, não sendo possível portanto receber um pacote de dados tão extenso quanto o do sensor *GPS*, que ocuparia posições de armazenamento dos dados de controle de trajetória e controle de mensagens.

Para otimizar o espaço de memória disponível optou-se em armazenar somente as várias essenciais para uso na navegação, disponibilizando somente uma posição de memória para cada um dos dados recebidos, ao contrário da sentença NMEA escolhida, que reserva uma posição para cada caractere dos dados. O cabeçalho também foi reduzido a apenas dois identificadores (“\$G”), e a palavra soma foi novamente recalculada, aplicando como metodologia a operação

lógica booleana xor entre todos os caracteres do pacote, à exceção do primeiro ('\$'). A comparação entre o uso da memória da sentença NMEA padrão e a memória otimizada é mostrada na figura 45:

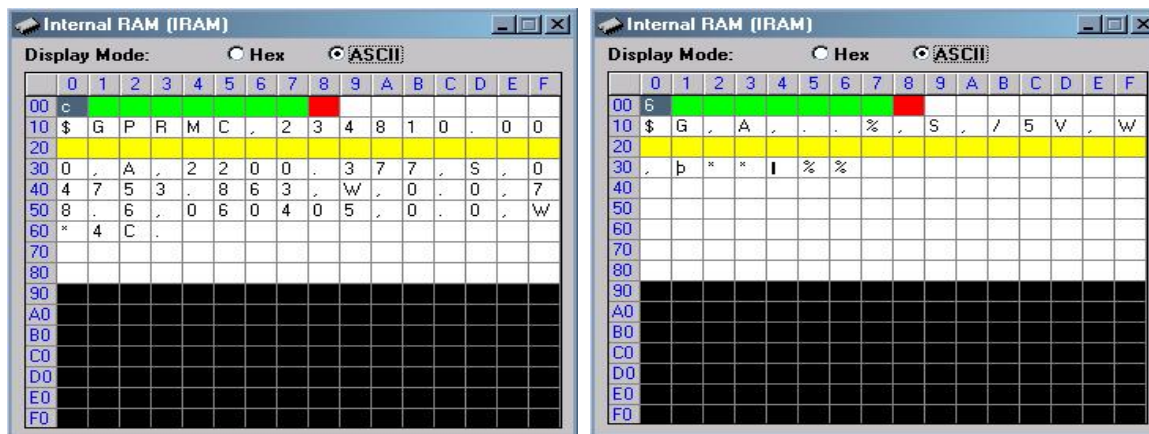


Figura 48. Uso de memória na forma padrão e pós-otimização

Os caracteres representativos das coordenadas de latitude e longitude ocupavam ao todo 22 posições, incluindo os caracteres separadores e as posições em relação aos hemisférios. Com a otimização o espaço ocupado reduz-se a somente 11 posições, representando os mesmos valores, à exceção da primeira unidade da casa dos segundos da latitude e longitude, com o objetivo do valor não ficar maior do que 1 byte e dificultar futuros cálculos da navegação. O caractere da palavra soma original foi recalculado para a nova sentença e otimizado, já que seu valor ocupava duas posições de memória e sua representação em ASCII não podia ser diretamente aplicada para validar a sentença, por culpa de sua forma de apresentação. Uma outra rotina de conversão transforma a palavra soma em um valor real, a ser prontamente utilizado pelo sistema de gerenciamento para validar o recebimento deste pacote.

A tabela 11 identifica a representação de cada caractere e sua respectiva posição na memória de dados do microcontrolador:

Tabela 11 - Endereço e identificação de cada caractere do sensor *GPS*

endereço	identificação	endereço	Identificação
10H	início da string	1CH	minutos da longitude
11H	demarcador do GPS	1DH	segundos da longitude
12H	separador	1EH	separador
13H	estado da recepção	1FH	posição no hemisfério
14H	separador	30H	separador
15H	graus da latitude	31H	velocidade
16H	minutos da latitude	32H	marcador
17H	segundos da latitude	33H	marcador
18H	separador	34H	palavra soma
19H	posição no hemisfério	35H	fim pacote
1AH	separador	36H	fim pacote
1BH	graus da longitude		

A padronização da divisão dos identificadores através de separadores e o uso de um cabeçalho tem como principal objetivo o de manter a legibilidade e clareza da implementação.

6. Testes e resultados

6.1 Infraestrutura da embarcação e eletrônica

Para efetuar os testes em campo a embarcação foi toda reestruturada, incluindo duas baterias para alimentação de toda a eletrônica, novos chicotes elétricos mais robustos, chaves de proteção de controle dos motores, alinhamento das polias com correias dentadas, fixação da eletrônica, e toda a estrutura necessária para que a embarcação tivesse condições de ser transportada até ao campo de testes.

Duas baterias automotivas de 12V/60A independentes (figura 49) foram usadas, sendo uma responsável pela partida do motor de popa e a outra para a alimentação dos motores de acionamento e eletrônica de controle, com o objetivo de evitar que os ruídos provenientes da ignição do motor à combustão interfiram na estabilidade do sistema eletrônico. Considerando o consumo de corrente da eletrônica embarcada (1A/h), associado ao consumo do motor da direção (8A/h) e leme (6A/h), com tempos de funcionamento próximo de 20% do período total para a mudança de rumo, temos para os testes de navegação e simulação aproximadamente 23 horas de autonomia, tempo suficiente para diversas pesquisas na represa. O consumo de corrente do motor do manche não foi considerado no cálculo da autonomia máxima, pois nunca funciona concorrentemente com o motor do leme, possui consumo de corrente inferior e tem menor probabilidade de uso.



Figura 49. Baterias usadas para partida do motor de popa e eletrônica embarcada.

A interface do sistema eletrônico de controle com a embarcação realiza-se por um par de chicotes de conectores robustos, que alimentam todo o circuito e permitem o controle direto da ação sobre os motores, através da placa de relês que controlam a etapa de potência do sistema. A figura 50 mostra a disposição dos pinos destes conectores:

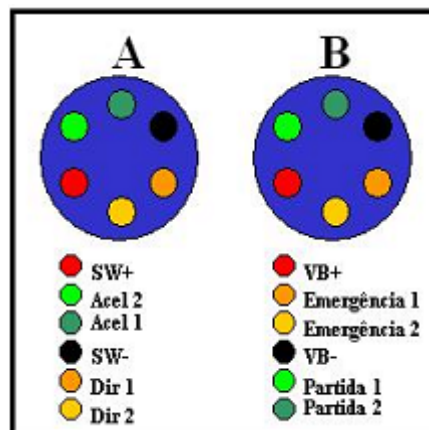


Figura 50. Interface da eletrônica com a embarcação

Através da interface física da eletrônica com a embarcação os motores do manche e leme são acionados em ambas as direções e o motor de popa é acionado e desligado eletronicamente. A alimentação do sistema eletrônico microcontrolado (VB+ e VB-) é separada da placa de relês que controlam os motores (SW+ e SW-) para a segurança dos testes em campo. A alimentação da placa de relês passa primeiramente por uma chave mecânica, que habilita para o operador da embarcação o controle manual, através de botoeiras, e impede momentaneamente o

controle eletrônico em caso de travamento ou uso inseguro. Uma chave geral também foi instalada no painel central para rapidez de acesso pelo operador para situações de emergência, como a eminência de choque com outras embarcações ou pela rápida proximidade com a margem da represa.

O chicote elétrico é conectado ao console principal da eletrônica embarcada (figura 51), que inclui o sistema de gerenciamento, posicionamento e a placa de relês.

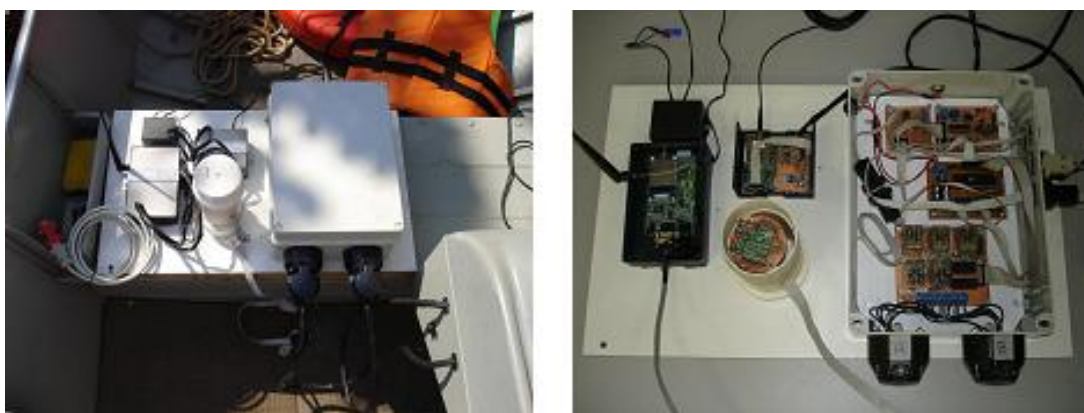


Figura 51. Acabamento e infraestrutura para eletrônica embarcada

Os sensores *GPS*, bússola eletrônica, leme, manche e o *modem* da telemetria de dados comunicam-se com o console principal através de interfaces seriais separadas, e todos estão sobre uma mesma plataforma de apoio e presa no casco da embarcação, amenizando vibrações provenientes do próprio ambiente de trabalho. Cada sensor possui incorporado ao seu encapsulamento um *led* de diagnóstico que indica o estado atual de leitura (sensor *GPS*) e se há alimentação contínua, permitindo a monitoração visual de falhas nos equipamentos eletrônicos na fase de testes de pesquisa em campo.

6.2 Algoritmo de navegação autônoma

A implementação de um algoritmo de navegação autônoma para o sistema de gerenciamento é mais uma ferramenta de testes para validar a telemetria de dados, reunindo assim todas as rotinas de controle dos sensores no sistema de gerenciamento em um só fluxo de programa, que altera dinamicamente a posição da embarcação ao longo do tempo e conseqüentemente muda o estado de todos os sensores presentes na embarcação, gerando uma coleta rica de dados.

Para toda a pesquisa realizada em campo a embarcação deve alcançar os pontos de coleta de dados hidrológicos através do menor caminho possível e de forma autônoma, otimizando todo o processo. O menor caminho possível neste processo de navegação entre o ponto origem e destino é uma linha reta que liga os dois pontos, mas pelo fato embarcação ser motorizada e possuir um sentido de aceleração, tomadas de decisão de rota devem ser calculadas com o auxílio dos equipamentos de geoposicionamento. O gerenciamento da rota e atuação na embarcação realiza-se por sistemas microcontrolados embarcados, programados em linguagem de baixo nível, e esse ambiente de trabalho sugere algumas restrições a respeito das técnicas viáveis de programação das rotinas que implementam a navegação autônoma:

- Operadores matemáticos: as operações possíveis dentro desse ambiente são básicas, onde uma simples operação de multiplicação e divisão podem ocasionar erros imprevisíveis;
- Memória de dados limitada: para armazenar apenas o ponto atual de localização (latitude/longitude) e seu antecessor da embarcação precisamos de 12 posições de memória, além do pacote de dados, que mesmo reduzido e otimizado

pelo sistema de posicionamento, ocupa 33 posições da memória. Dessa forma não há espaço disponível para o armazenamento de um grande conjunto de variáveis de controle;

- Memória de programa limitada: por limitações no *hardware* da embarcação já desenvolvida em trabalhos anteriores o algoritmo não pode alcançar grande nível de complexidade matemática, o que acarretaria no estouro da memória (2KBytes) e conseqüentemente na falha de conclusão da implementação do sistema.

Há diversos algoritmos conhecidos que tratam sobre o assunto, mas todos eles não se encaixam na arquitetura aplicada devido à sua complexidade de implementação em um sistema restritivo como o já existente. A metodologia aplicada na construção do algoritmo de navegação autônomo leva em conta primeiramente os elementos restritivos do projeto, na tentativa de alcançar um planejamento de rota o mais próximo possível do ideal. A partir deste momento são criadas hipóteses de navegação que possam indicar uma trajetória que alcance um nível aceitável de aproximação ao ponto destino. A figura 52 mostra o ponto origem x , seus possíveis destinos A,B,C e D e as possíveis hipóteses de rumo da embarcação 1,2,3 e 4:

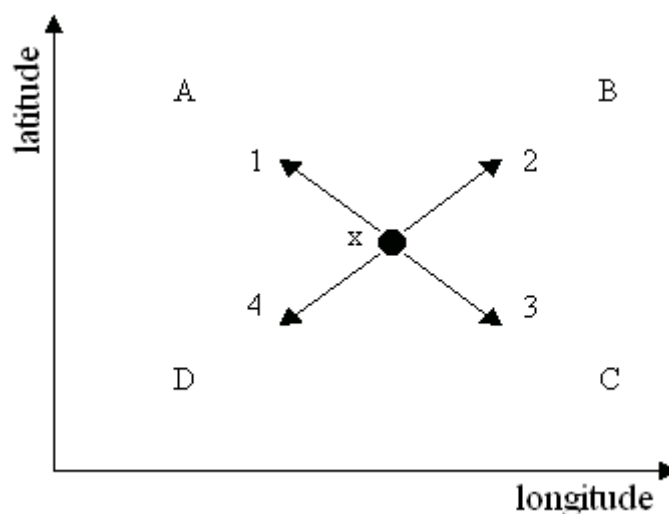


Figura 52. Ponto origem, possíveis destinos e rumos da embarcação

Dentre todas as possibilidades de mudança de rumo de navegação surgem as seguintes alternativas:

✓ **Manter o rumo:** quando a diferença, de latitude e longitude, entre o ponto origem e destino diminui entre duas leituras consecutivas dos dados de geoposicionamento, indicando o rumo correto da embarcação. Exemplo: origem no ponto x, destino no ponto C, direção e sentido da embarcação autônoma para rumo 3.

✓ **Girar 180⁰:** quando a diferença, de latitude e longitude, entre o ponto origem e destino aumenta entre duas leituras consecutivas dos dados de geoposicionamento, indicando um distanciamento em relação ao ponto destino, para ambos os eixos. Exemplo: origem no ponto x, destino no ponto D, direção e sentido da embarcação autônoma para rumo 2.

✓ **Girar a embarcação em sentido horário/anti-horário:** quando a diferença, de somente um dos eixos (latitude ou longitude), entre o ponto origem e destino, entre duas leituras consecutivas de geoposicionamento, aumenta. A conclusão é que por um dos eixos das orientações a embarcação aproxima-se do ponto destino, mas em contrapartida se distancia em relação ao outro eixo. Exemplo: origem no ponto x, destino no ponto C, direção e sentido para rumo 2 (aproximação em longitude e distanciamento em latitude) ou origem no ponto x, destino no ponto C, direção e sentido da embarcação autônoma para rumo 4 (aproximação em latitude e distanciamento em longitude).

As decisões de manter o rumo ou efetuar um giro de 180⁰ à embarcação são claras, o que não se verifica para a tomada de decisão de giro, necessária sempre que não houver a convergência de valor em relação a um dos eixos de orientação, ao longo do trajeto da embarcação. Definir exatamente para qual direção girar, ou

seja, distinguir qual o sentido de giro mais ideal, é crucial para a otimização do processo de navegação, pois a embarcação é um veículo de grande tamanho (6m) e de alta potência, onde qualquer manobra exige muito tempo e espaço para deslocamento.

Como primeira hipótese para decisão de giro definiu-se que se a diferença de leitura, entre dois pontos distintos, implica na convergência somente da longitude, então a embarcação toma como decisão girar para o sentido horário, caso contrário o sentido de giro é anti-horário. A tabela 12 apresenta os resultados possíveis de decisões, para todos os rumos e destinos.

Tabela 12 – 1ª alternativa de decisão de giro para a embarcação

rumos	destinos			
	A	B	C	D
1	mantém	anti-horário	180 ⁰	horário
2	anti-horário	mantém	horário	180 ⁰
3	180 ⁰	horário	mantém	anti-horário
4	horário	180 ⁰	anti-horário	mantém

A análise da tabela em conjunto com a figura 52 permite concluir um erro de 25% do total de decisões possíveis para a navegação, mas especificamente para tomada de decisão de giro o erro representa exatos 50% do total das possibilidades (células 1B, 1D, 3B e 3D), valor extremamente alto, inviabilizando esta estratégia de navegação. Invertendo a abordagem de decisão de giro, ou seja, se a latitude converge, para leituras de geoposicionamento em tempos distintos, a embarcação toma como decisão o giro para o sentido horário, temos na prática a apresentação da mesma porcentagem de erros, fator exemplificado pela tabela 13:

Tabela 13 – 2ª alternativa de decisão de giro para a embarcação

rumos	destinos			
	A	B	C	D
1	mantém	horário	180 ⁰	anti-horário
2	horário	mantém	anti-horário	180 ⁰
3	180 ⁰	anti-horário	mantém	horário
4	anti-horário	180 ⁰	horário	mantém

A tabela mostra a propagação do erro para as células 2A, 2C, 4A e 4C. Neste sentido conclui-se que somente com os dados das diferenças entre dois pontos consecutivos de leitura do *GPS* torna-se impossível o cálculo da navegação autônoma para a embarcação instrumentada. A necessidade de adição de mais variáveis ao sistema é necessária para que a estratégia possa alcançar resultados satisfatórios.

Ao longo dos testes e simulações para a estratégia de navegação verificou-se que, pelos dados do sensor *GPS*, a latitude e a longitude sofrem um processo de incremento ou decremento de seus valores com relação aos seus eixos de referência, caso do meridiano que corta o país do Equador (latitude), e do Meridiano de Greenwich (longitude). A tabela 14 é então adicionada à estrutura de hipóteses:

Tabela 14 - Incremento e decremento das coordenadas sobre eixos de referência

	rumos			
	1	2	3	4
Latitude	incremento	incremento	decremento	decremento
Longitude	decremento	incremento	incremento	decremento

O conjunto dos fatores avaliados elimina o fator de ambigüidade na decisão da direção do giro da embarcação e aplica um padrão correto de abordagem de navegação com 100% de acerto. A tabela 15 indica a primeira aproximação do algoritmo de navegação, com objetivo de se alcançar o ponto destino:

Tabela 15 - Primeira aproximação do algoritmo de navegação

		diferença para o ponto destino			
		latitude positivo longitude negativo	latitude negativo longitude positivo	latitude positivo longitude positivo	latitude negativo longitude negativo
incremento e decremento de posição	Latitude e Longitude com mesma direção	sentido anti-horário	sentido horário	mantém o curso	girar 180 ^o
	Latitude e Longitude com direções opostas	sentido horário	sentido anti-horário	mantém o curso	girar 180 ^o

Com a aplicação deste método a embarcação aproxima-se do objetivo, ao longo da pesquisa, por ambas coordenadas. Contudo, não há a garantia de convergência mútua. A figura 53 exemplifica o problema:

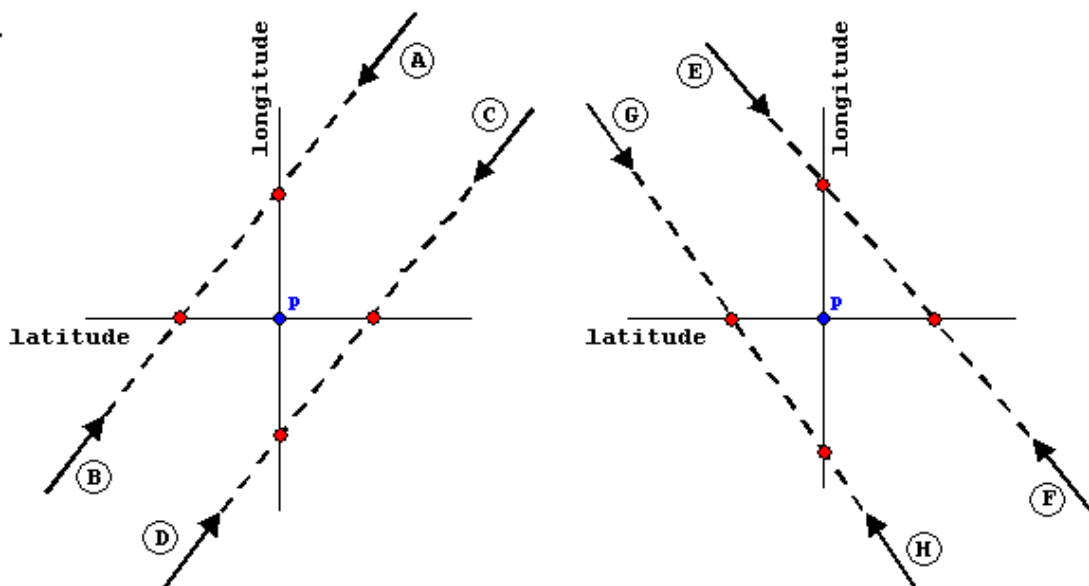


Figura 53. Problema da convergência da latitude e longitude

As direções A, B, C, D, E, F, G e H comprovam, na maioria das vezes, que sempre se alcança o ponto destino P através de somente uma coordenada, e não das duas simultaneamente.

Sabendo-se qual coordenada convergirá primeiramente, através do cálculo da diferença de valores entre o ponto atual e destino, e adicionando os parâmetros de incremento e decremento de posição em relação ao eixo dos hemisférios acha-se uma metodologia para a segunda aproximação (tabela 16), que avalia precisamente para onde a embarcação autônoma deve girar para que ela alcance o ponto destino pelas duas coordenadas conjuntas.

Tabela 16 - Segunda aproximação ao ponto destino

		diferença de valor para ponto destino	
		latitude > longitude	latitude < longitude
variação em relação aos hemisférios	lat. incr. ou lat. decr.	sentido anti-horário	sentido horário
	long. incr. ou long. decr.	sentido horário	sentido anti-horário

A segunda aproximação ao ponto destino é útil nos casos em que há a piora da diferença da convergência entre as coordenadas do ponto atual, ao longo das medidas de geoposicionamento. Se a diferença entre coordenadas decresce naturalmente então a embarcação mantém o rumo até que se encontre finalmente o ponto destino, ou recomeça-se o algoritmo novamente (1ª aproximação), no caso de sucesso em apenas umas das coordenadas.

6.2.1 Simulação numérica da trajetória

Para validação da estratégia de navegação autônoma, através de um exemplo numérico, simulou-se o percurso da embarcação em papel graduado de duas dimensões (latitude e longitude) e resolução de 0,5cm. O tempo para cada leitura do sensor corresponde à marcação da distância de 1cm no papel, e para mudança de direção tem-se como ferramenta o transferidor para o exato cálculo em graus do giro da embarcação. A tabela 17 apresenta os resultados da abordagem escolhida, sendo:

- Ponto origem: latitude 10, longitude 35 e 30 graus de orientação
- Ponto destino: latitude 40 e longitude 10
- Giro da bússola eletrônica: 20 graus

Tabela 17 – Exemplo numérico da navegação autônoma

Pontos	Lat.	Lon.	Búss.	Dif. destino	Lat. destino	Dif. destino	Lon. destino	Saldo Lat.	Saldo Lon.	Direção do giro	Lat – Lon
A	10	35	30	30	25			x	x	x	x
B	12	37	30	28	27			2	-2	anti-horário	x
C	15	37	10	25	27			3	0	anti-horário	x
D	18	36	350	22	26			3	1	mantém	4
E	21	36	350	19	26			3	0	anti-horário	7
F	23	34	330	17	24			2	2	anti-horário	7
G	25	32	310	15	22			2	2	anti-horário	7
H	26	29	290	14	19			1	3	mantém	5
I	27	26	290	13	16			1	3	mantém	3
J	28	23	290	12	13			1	3	mantém	1
K	29	20	290	11	10			1	3	mantém	1
L	30	17	290	10	7			1	3	horário	3
M	32	14	310	8	4			2	3	horário	4
N	35	13	330	5	3			3	1	mantém	2
O	38	11	330	2	1			3	2	mantém	1
P	40	10	330	0	0			x	x	x	x

O cálculo de percurso para a primeira aproximação foi utilizado até o ponto “C”, objetivando achar o rumo para a embarcação. Deste ponto adiante a tabela de segunda aproximação é utilizada quando a diferença entre latitude e longitude não decresce (última coluna - pontos E, F e G) impedindo a convergência gradual das coordenadas, que começa somente partir do ponto “H”. O cálculo finaliza se a diferença entre o ponto atual e destino é próxima ou igual a zero (ponto P).

A trajetória marcada no papel graduado foi inserida num gráfico bidimensional (figura 54), através do programa Origin, para a análise visual da simulação de navegação da embarcação autônoma:

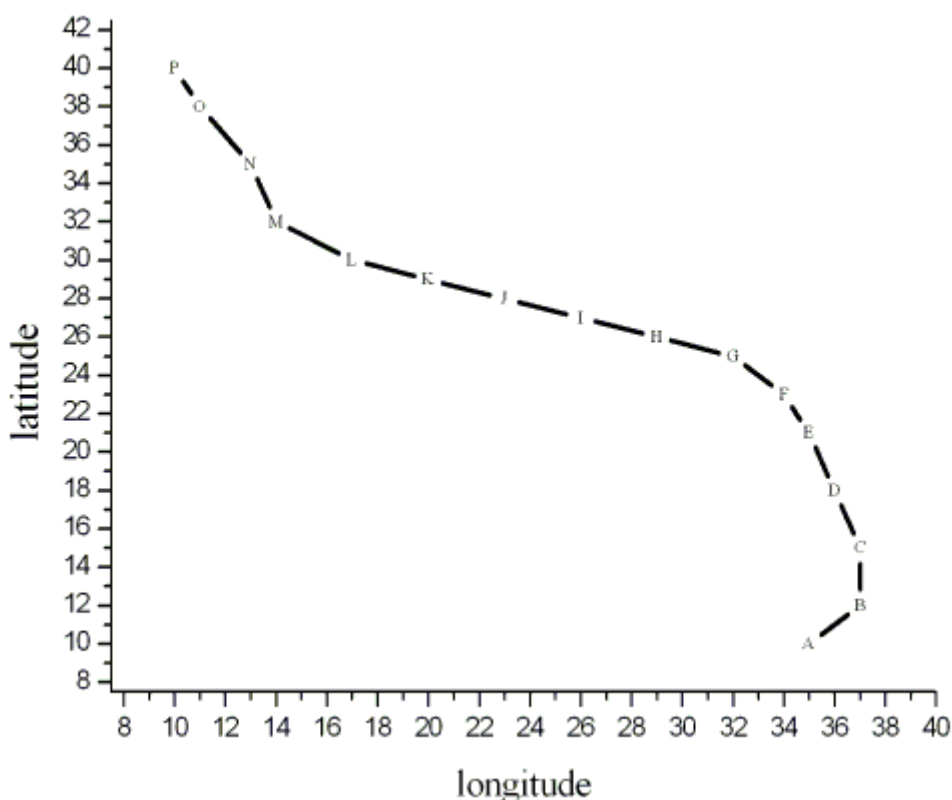


Figura 54. Simulação da trajetória do algoritmo de navegação

Apesar do algoritmo de navegação utilizar um modelo matemático simples sua resposta é satisfatória, considerando-se não haver erros na escolha de direção de giro. Observa-se também que a convergência do algoritmo depende da resolução escolhida para o giro da bússola, que pode ser dimensionado dinamicamente tanto

para a primeira aproximação até encontrar o rumo (giro maior) ou para a segunda aproximação se a embarcação está próxima do objetivo (giro menor). Esse algoritmo de navegação é uma ferramenta que agrega valor ao projeto e permite uma aproximação da realidade para os testes da aquisição da telemetria de dados e imagens em tempo mínimo.

6.3 Testes da telemetria de dados no campo de futebol

O teste de avaliação de operação do sistema de telemetria de dados implementado foi realizado no campo de futebol da USP de São Carlos (figura 55), com o objetivo de se observar o comportamento do fluxo dos dados ao longo do tempo de simulação do percurso.



Figura 55. Campo de futebol da USP de São Carlos

A imagem da foi obtida através do programa Google Earth, que contém uma ferramenta de medição de distância entre dois pontos, utilizada para o cálculo total da área do campo (aproximadamente 9000m²).

Para simulação da navegação da embarcação usou-se como veículo de transporte da eletrônica embarcada uma carriola de uso geral (figura 56). As entradas analógicas do leme e manche são simuladas através de sensores resistivos conectados serialmente ao console principal. A estação terrestre é composta pelo *notebook* com o aplicativo da telemetria e o *transceiver*.



Figura 56. Equipamentos para simulação do fluxo da telemetria

Para o primeiro teste de operação, a eletrônica embarcada percorreu toda a extensão periférica do campo de futebol, e o aplicativo na estação terrestre recebeu em tempo on-line todas as informações. O algoritmo de navegação autônoma está implementado e ativo no sistema de gerenciamento embarcado, com apenas a modificação de sua estrutura geral nas ações de mudança de rumo para esta simulação, sendo trocadas pela ação de leitura dos sensores. A figura 57 apresenta o relatório gerado ao final do teste, que mostra os valores de todo o conjunto de sensores embarcados.

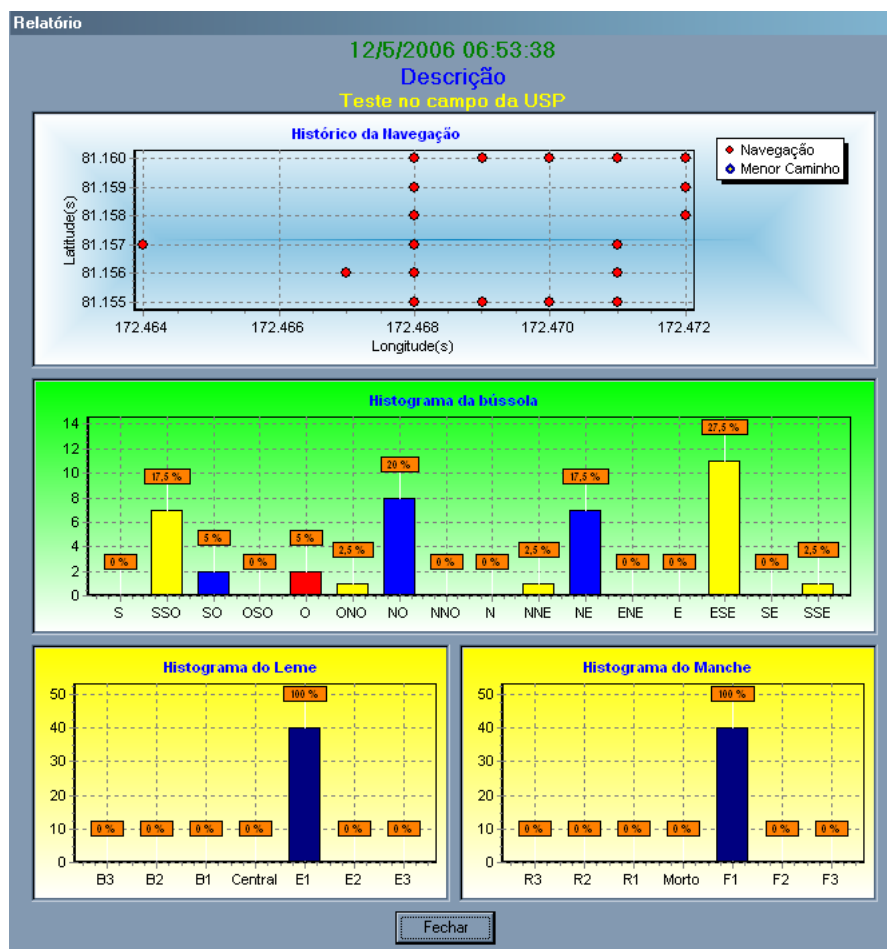


Figura 57. Relatório da telemetria de dados

A eletrônica embarcada transmitiu para o aplicativo da estação terrestre exatamente 40 pacotes da telemetria de dados. O histórico de navegação mostra a formação geométrica do campo de futebol, com algumas coordenadas fora da linha desejada, ocorrida pela precisão do sensor e pelos arredondamentos dos segundos das coordenadas pelas rotinas dos microcontroladores embarcados. O gráfico de percurso somente apresenta 19 pontos da telemetria porque os restantes são coincidentes. O histograma da bússola mostra a freqüência de cada orientação, tendo 4 das orientações ficando ressaltadas, representando cada lado do campo de futebol. Os sensores do leme e manche não sofreram variações e permaneceram na mesma posição, pois seriam necessárias várias paradas do veículo de transporte para a mudança da resistência de cada um e os mesmos já foram intensamente

simulados conforme visto na seção 5.3.2.1. O teste para o leme e manche vem a comprovar a confiabilidade dos resultados, sem ocorrência de erros e com amostragem completa em apenas uma posição do histograma para cada sensor.

Para confirmação da consistência de um grande fluxo de transmissão da telemetria de dados o veículo de transporte percorreu aleatoriamente todo o espaço de teste, produzindo a quantidade de 118 pacotes de dados, sendo todos eles recebidos com 100% de precisão, validando desta forma o uso da arquitetura e implementação projetada como uma ferramenta coerente para o monitoramento e coleta de dados à longa distância.

6.4 Testes de transmissão na represa do Lobo

Os testes de transmissão dos enlaces de rádio frequência foram realizados na Represa do Lobo (Broa), por sua localização e onde a USP possui um centro de recursos hídricos.

A Represa do Lobo (Broa), mostrada em duas visões na figura 58, foi construída em 1936 pela antiga Central Elétrica de Rio Claro S.A., com a finalidade de produzir energia elétrica e hoje é utilizada apenas com fins de lazer e pesquisa. Tem aproximadamente 8km de comprimento, profundidade máxima de 11,5m e 22.000.000m³ de água. A represa pertence ao município de Itirapina, e dista aproximadamente 18 Km de São Carlos. A construção de uma represa representa uma mudança drástica no ambiente, alterando totalmente suas características, transformando um ambiente terrestre em ambiente aquático, destruindo assim a

vegetação e alterando a fauna aí existente, além de modificar as características físico-químicas e biológicas da água.



Figura 58. Represa do Lobo (Broa)

O teste de transmissão do enlace da telemetria de dados e do enlace de rádio frequência multimídia tem como responsabilidade conhecer o máximo alcance da recepção dos dados e testar sua confiabilidade.

Na estação terrestre o aplicativo de teste do *transceiver* da telemetria de dados é habilitado para transmitir continuamente um pacote de dados para a instrumentação eletrônica embarcada, e esta retorna automaticamente o mesmo pacote até a base. A qualidade de transmissão do enlace de rádio frequência multimídia é monitorado através da imagem e áudio recebido pela antena direcional, conectada ao monitor da estação terrestre. O percurso do teste de validação é realizado pelo usuário, que pilota a embarcação manualmente por todo o perímetro da represa e depois de forma aleatória por toda sua superfície. Os resultados da pesquisa do *transceiver* de dados são mostrados na figura 59.

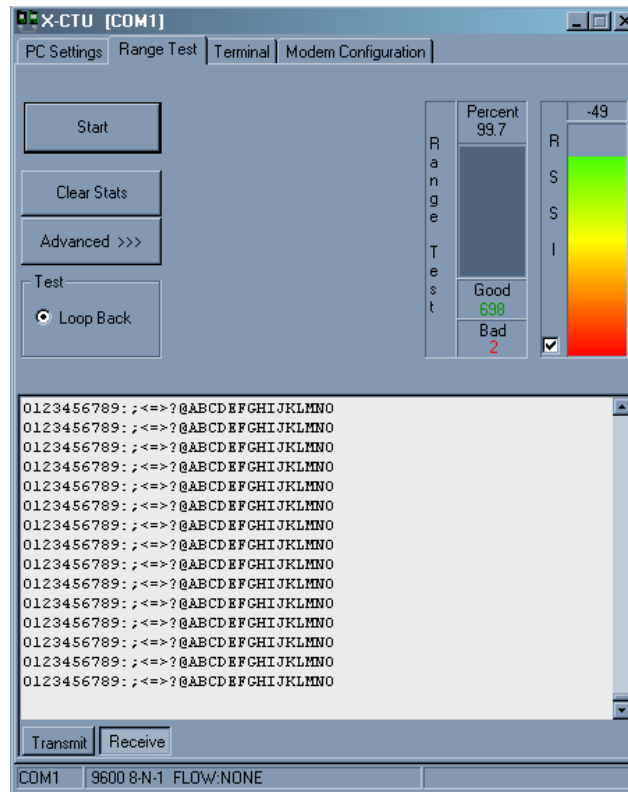


Figura 59. Resultados do teste dos enlaces de rádio frequência

O aplicativo de teste do *tranceiver* recebeu em tempo mínimo um total de 700 pacotes de caracteres na formatação ASCII, com perda de informação de dados em apenas 2 deles, representando um sucesso de 99,7% na comunicação em tempo mínimo. Os comunicadores são detentores da tecnologia de controle de fluxo e erro, portanto os dois pacotes recebidos com problemas foram prontamente reenviados e recebidos novamente, diferenciando dos demais apenas pelo pequeno atraso na política de comunicação.

O enlace de rádio frequência multimídia apresentou, durante toda a navegação pelo perímetro da represa, interferências na imagem e ruídos no áudio, principalmente quando as antenas de transmissão e recepção não estavam alinhadas, por culpa da tecnologia de transmissão das antenas ser de modo direcional. Apesar de todos os problemas apresentados a imagem e o áudio são suficientes para se aplicar o método do piloto virtual, para uso da navegação

autônoma para um melhor posicionamento do ponto de partida da embarcação para a pesquisa e no caso de problemas decorrentes na falha dos sensores embarcados, que impossibilitem a embarcação de realizar a navegação automaticamente de forma segura.

7. Conclusões e sugestões para trabalhos futuros

Este trabalho teve por objetivo o desenvolvimento de um sistema de telemetria de dados e imagens para uma plataforma autônoma para coleta de dados hidrológicos, dando continuidade aos desenvolvimentos do Grupo de Instrumentação Virtual e Microprocessada da EESC/USP de São Carlos, para a implementação de uma embarcação autônoma completa.

Para o desenvolvimento de todo o sistema foram revistos conceitos básicos sobre robôs móveis autônomos (classificações, sensores e arquiteturas de controle), equipamentos de transmissão por rádio frequência a longas distâncias, irradiações e ganhos de antenas, sistemas de georeferenciamento por *GPS* e modelos de sistema de navegação. Os projetos correlatos descritos neste documento confirmam o uso de enlaces de rádio frequência para comunicação e equipamentos de geoposicionamento para uso em sensoriamento remoto, inserindo a implementação apresentada no mesmo grau tecnológico que outras referências da comunidade científica.

Os testes e calibrações dos sensores de navegação (leme e manche), bússola eletrônica e *GPS*, assim como as simulações e os experimentos em campo do alcance de transmissão dos enlaces de rádio frequência e o fluxo da telemetria de dados e imagens foram condizentes com as expectativas iniciais, validando a construção do *hardware* da eletrônica embarcada, sistema de aquisição, controle e o aplicativo da telemetria de dados da estação terrestre dentro dos objetivos e requisitos de sistema propostos, satisfazendo plenamente o uso da telemetria como meio de monitoramento de dados e controle da navegação, e conseqüentemente

culminando numa plataforma totalmente operacional, servindo como base para futuros desenvolvimentos de veículos autônomos aquáticos ou não.

Sentiu-se nos testes realizados que muitos aperfeiçoamentos poderão ser realizados para melhorar o desempenho da plataforma, tanto do ponto de vista de atuação como do ponto de vista operacional. Os aperfeiçoamentos necessários devem permitir uma série de trabalhos ao nível de graduação (iniciação científica), ao nível pós-graduação e poderão ter também desdobramentos em diversas áreas ligadas a veículos autônomos e robótica móvel. Os aperfeiçoamentos propostos na plataforma autônoma são os seguintes:

- Aperfeiçoamento dos sistemas de acionamento dos comandos de aceleração (manche) e de direção (leme), através do controle de potência por modulação de largura de pulso (“PWM”) permitindo assim um melhor controle da velocidade de ambos motores; sensores elétricos e óticos redundantes em cada acionamento (fusão sensorial); interface de comunicação digital entre os sistemas de acionamento e o processador embarcado, do tipo CAN ou outra a escolher . Deverão também ser estudados sistemas de segurança que permitam o retorno imediato dos sistemas de acionamento a uma posição neutra em caso de alguma pane no controle da embarcação.

- Colocação de 2 motores de popa de corrente contínua, de pequena potência, para aproximação com maior precisão dos pontos de coleta. O sistema de propulsão atual da embarcação é um motor de popa a gasolina de 30Hp, bom para navegação entre os pontos porém de difícil controle para aproximação dos pontos de coleta. Deverão ser também desenvolvidos os sistemas de controle de potência dos motores, utilizando modulação por largura de pulso e interface digital tipo CAN, ou outra a escolher.

- Substituição dos microcontroladores atualmente utilizados por uma unidade de processamento embarcada tipo PC100. A unidade de processamento embarcada deverá ser interfaceada aos sistemas de acionamento e sensores da embarcação e aos sistemas de telemetria de dados e de imagens.
- Desenvolvimento de um sistema de lançamento da sonda para análise da qualidade da água, com controle de lançamento, recolhimento e monitoração da profundidade.
- Substituição dos sistemas de posicionamento da embarcação *GPS* e bússola eletrônica por sistemas de melhor desempenho. Os sensores utilizados atualmente são lentos (frequência de amostragem da ordem de 2,5Hz) o que dificulta o procedimento de navegação.
- Desenvolvimento dos programas de controle da embarcação para o novo sistema de processamento e hardware utilizado.
- Aperfeiçoamento dos programas da estação base (inserção dos pontos de pesquisa por mapas georeferenciados) e melhorias no controle remoto manual da embarcação.
- Construção de uma base giratória microcontrolada para o transmissor embarcado do sinal multimídia, direcionando a antena para a estação terrestre, objetivando a melhora da recepção do sinal.
- Desenvolvimento e avaliação de rotinas de navegação autônoma, utilizando os novos aperfeiçoamentos incorporados.

Referências bibliográficas

- (1) CNPQ. Medidor de pH e concentração de íons microprocessado. IN: INSTRUMENTAÇÃO: Catálogo de protótipos, processos e serviços, PADCT/SINST, 1995.
- (2) SILVA, A. M.; RODA, V. O. Monitor para registro de distúrbios em rede C.A. Anais do XV encontro Nacional de Física da Matéria Condensada, Caxambú, SP, 1987.
- (3) ROSSI, J. C.; RODA, V. O. Técnica de obtenção de imagem por microscopia foto-acústica. Anais da X reunião de Física do Estado Sólido, Caxambú, SP, 1987.
- (4) TRINDADE Júnior, O.; RODA, V. O. Sistema de testes para estudo de distúrbios da rede elétrica sobre microcomputadores. Anais do XXII Congresso Nacional de Informática - SUCESU'89, pp. 203-211, 1989.
- (5) RODA, V. O. Dispositivos sensores com fibras óticas. IN: 1º ENCONTRO SOBRE DISPOSITIVOS À FIBRA E GUIAS PLANARES PARA COMUNICAÇÕES ÓTICA, Campinas, 1987.
- (6) SILVA, R. V.; RODA, V. O.; ARAKAKI, H. Conversor A.D. de 12/14 bits com entradas multiplexadas para microcomputador IBM-PC. Anais do VIII Congresso de Iniciação Científica e Tecnológica em Engenharia, São Carlos, 1989.
- (7) SANTOS, M. O.; RODA, V. O. Sensor de índice de refração. Anais do VIII Congresso de Iniciação científica e Tecnológica em Engenharia, São Carlos, 1989.
- (8) TOZZO, P. A.; RODA V. O. Amplificadores para a medida de campos magnéticos produzidos por axônios e nervos isolados. Anais do IX Congresso de Iniciação Científica e Tecnológica em Engenharia, São Carlos, 1990.

(9) BRUNO, R.; RODA, V. O. Virtual pH Meter for PCs. *Electronics Today International*, vol. 26, no. 6, 1996.

(10) ARMELIN, F. B. **Acionamento e Controle Microprocessado para Navegação de Embarcação com Auxílio de GPS.** (2002). 30f. Projeto de Iniciação Científica - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2002.

(11) ARMELIN, F. B. **Sistema de Navegação Autônoma para Embarcação de Coleta de Dados Hidrológicos com Auxílio de GPS.** (2003). 28f. Projeto de Iniciação Científica - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2003.

(12) DUDEK, G.; JENKIN, M. **Computation Principles of Mobile Robotics.** Cambridge University Press, Cambridge, UK, 2000.

(13) RIDAO, P.; BATTLE, J.; AMAT, J.; ROBERTS, G. N. Recent trends in control architectures for autonomous underwater vehicles. *International Journal of Systems Science* 30(9): 1033-1056, 1999.

(14) VALAVANIS, K. P.; GRACANIAN, D.; MATIJASEVIC, M.; KOLLURU, R.; DEMETRION, G. A. Control architectures for autonomous underwater vehicles. *IEEE Control Systems Magazine* 17(6): 48-64, 1997.

(15) HALL, W. D; ADAMS, M. B. *Autonomous vehicles applicativo taxonomy.* Proceedings of the IEEE Symposian on Autonomous Underwater vehicles tecnology, The OES of IEEE, IEEE, p. 49-64, 1992.

(16) WANG, H. H.; ROCK, S. M. *OTTER: The design and development of an intelligent underwater robot.* *Journal of Autonomous Robots* 3 (2-3): 297-320, 1996.

(17) BARROS, E. A. **A Cooperative Control System and Its Application to the Collision Avoidance Guidance of Autonomous Underwater Vehicles.** (2004). PhD thesis, University of Tokyo, Tokyo, Japan, 2004.

- (18) BROOKS, R. A. (1986). *A robust Layered Control System for a Mobile Robot*. IEEE Journal of Robotic and Automation VRA-2, n.1, p. 14-23.
- (19) BLAHUT, R. E. et al. (1992). *Radar and sonar*. Ed. Springer-Verlag, New York, USA.
- (20) TANENBAUM, A. S. (1996). *Computer networks*. Ed Prentice Hall, New Jersey, USA.
- (21) YACOUB, M. D. (1993). *Foundation of mobile radio engineering*, CRC Press, Florida, USA.
- (22) TANENBAUM, A. S. (1996). *Computer networks*, Prentice Hall, 3ed, New Jersey, USA.
- (23) ANATEL – Quadro de atribuição de faixas de frequência no Brasil. Disponível em: <http://www.anatel.gov.br/Radiofrequencia/default.asp?CodArea=4&CodPrinc=1>. Acesso em: 20 ago. 2004.
- (24) ESTEVES, L. C. (1981). *Antenas Teoria Básica e Aplicações*, Editora McGraw-Hill do Brasil, São Paulo, São Paulo.
- (25) SARTORI, J. C. (1999). *Linhas de Transmissão e Carta de Smith: Projeto Assistido por Computador*, Escola de Engenharia de São Carlos (EESC) – Universidade de São Paulo, Editora USP, São Carlos, São Paulo.
- (26) SMITH, S. G. (1993). *Automatic navigation in the air and at sea*. Aeronautical Journal, v.97, n.966, p.183-194, June, 1993.
- (27) FÉ, I. M. M.; HEMERLY, E. M. (2002). *Caracterização do Sensor GPS e Aplicação em Barco Autônomo*, Departamentos de Sistemas e Controle (IEES) e

Telecomunicações (IEET) do Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, São Paulo.

(28) AURORA – Dirigível Robótico não Tripulado (2004). Centro de Pesquisas Renado Archer (Cenpra) do Ministério de Ciência e Tecnologia, Campinas, São Paulo.

(29) RAMOS, J. J. G. et al. A aplicativo environment for an autonomous unmanned airship. In: PROCEEDINGS OF THE 1999 IEEE/ASME INTERNATIONAL CONFERENCE ON ADVANCED INTELLIGENTE MECHATRONICS. Atlanta, USA, 1999.

(30) ELFES, A. et al. (1998). *Projeto AURORA*. In: XII BRAZILIAN CONFERENCE ON AUTOMATICS. Proceedings... Uberlândia, MG, p.79-84.

(31) SOUZA, N. P. **Sistemas Alternativos para a Obtenção de Imagem Aérea e sua Aplicação na Agricultura**. (1999). 124f. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 1999.

(32) RIBEIRO, R. E. **STT - Um Sistema para Telemetria e Telecomando de Aeronaves do Projeto ARARA**. (2001). 137f. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2001.

(33) BARBOSA, L. C. P. **SiNaCom – Sistema de Navegação e Controle de Missão do Projeto ARARA**. (2001). 137f. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2001.

(34) NERIS, L. O. **Um Piloto Automático para as Aeronaves do Projeto ARARA**. (2001). 113f. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2001.

(35) TREVIZANI, K. M. **Uma Extensão do Sistema de Telemetria e Telecomandos do Projeto ARARA para Transmissão Digital de Vídeo e Dados**. (2002). 126f. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo, São Carlos, 2002.

(36) DYNATECH. Disponível em: <http://www.dynatech.com.br/>. Acesso em: 10 set. 2004.

(37) PRECISION NAVIGATION, INC. Disponível em: <http://www.pnicorp.com/>. Acesso em: 01 jun. 2006.

(38) MAXSTREAM – Data radio modems and radio modules. Disponível em: <http://www.maxstream.net/>. Acesso: 03 jun. 2006.

(39) WIRELESS VIDEO CAMERAS. Disponível em: <http://wirelessvideocameras.com>. Acesso: 15 jun. 2006.

(40) REEVS, C. R. (1995). *Modern Heuristic Techniques for Combinatorial Problems*, Ed. McGraw-Hill, California, USA.

ANEXO A – Aplicativo da estação terrestre

program MestradoTelemetria;**uses**

```
Forms,  
UTelemetria in 'UTelemetria.pas' {Form1},  
UCAixeiroViajante in 'UCAixeiroViajante.pas' {Form2},  
UPilotoVirtual in 'UPilotoVirtual.pas' {Form3},  
URelatorio in 'URelatorio.pas' {Form4};
```

```
{$R *.res}
```

begin

```
Application.Initialize;  
Application.CreateForm(TForm1, Form1);  
Application.CreateForm(TForm2, Form2);  
Application.CreateForm(TForm3, Form3);  
Application.CreateForm(TForm4, Form4);  
Application.Run;
```

end.

unit UTelemetria;**interface****uses**

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, QCCom32, StdCtrls, ExtCtrls, jpeg, Buttons, Menus, ColorGrd,
TeEngine, TeeFunci, Series, TeeProcs, Chart, MmSystem;

type

TForm1 = class(TForm)
Serial: TQCCom32;
Image1: TImage;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Centena: TLabel;
Label20: TLabel;
Label1: TLabel;
Enviar: TButton;
GroupBox1: TGroupBox;
Vel: TLabel;
Lat: TLabel;
Lon: TLabel;
hor: TLabel;
Sta: TLabel;
Latitude: TLabel;
Latg: TLabel;
LatM: TLabel;
LatS: TLabel;
Label25: TLabel;
Longitude: TLabel;
LonG: TLabel;
LonM: TLabel;
LonS: TLabel;
Label26: TLabel;
no: TLabel;
Velocidade: TLabel;
horario: TLabel;
Posicao: TLabel;
GroupBox2: TGroupBox;
LEM: TLabel;
leme: TLabel;
MAN: TLabel;
manche: TLabel;
Shape1: TShape;
BF1: TShape;
BF2: TShape;
BF3: TShape;
BR1: TShape;
BR2: TShape;
BR3: TShape;
Shape8: TShape;
BE1: TShape;
BE2: TShape;
BE3: TShape;
BB1: TShape;
BB2: TShape;
BB3: TShape;
R3: TLabel;
R2: TLabel;
R1: TLabel;
F1: TLabel;
F2: TLabel;
F3: TLabel;
E1: TLabel;
E2: TLabel;
E3: TLabel;
B1: TLabel;
B2: TLabel;
B3: TLabel;
MainMenu1: TMainMenu;
Arquivo1: TMenuItem;
Novo: TMenuItem;
Abrir: TMenuItem;
ArqLog: TMemo;
Timer: TTimer;
Label19: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Label24: TLabel;
LogCont: TLabel;
LChkBus: TLabel;
LChkNav: TLabel;
LChkGps: TLabel;
LData: TLabel;

```

Separador: TMenuItem;
NavegVisual: TChart;
Series2: TPointSeries;
sirene: TImage;
BtParar: TButton;
LEmergencia: TLabel;
Shape2: TShape;
TimerAck: TTimer;
Sair: TMenuItem;
AbreArquivo: TOpenDialog;
NovoArquivo: TOpenDialog;
Series1: TPointSeries;
procedure FormActivate(Sender: TObject);
procedure Bussola();
procedure Garmin();
procedure ManLem();
procedure AtualizaBussola();
procedure AtualizaLem();
procedure AtualizaMan();
procedure EnviarClick(Sender: TObject);
procedure NovaPesquisaClick(Sender: TObject);
procedure NovoClick(Sender: TObject);
procedure TimerPacDados(Sender: TObject);
procedure Inicializa(Sender: TObject);
procedure Plota(grla,mila,sela,grlo,milo,selo :integer);
procedure SireneClick(Sender: TObject);
procedure BtPararClick(Sender: TObject);
procedure TimerAckTimer(Sender: TObject);
procedure SairClick(Sender: TObject);
procedure AbrirClick(Sender: TObject);
procedure NavegVisualClickSeries(Sender: TCustomChart; Series: TChartSeries; ValueIndex: Integer; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

private
{ Private declarations }
public
end;

var
Form1: TForm1;
Palavra,NMEA: String;
contador: integer;
contbus,contnav,contgps: integer;
Arquivo: TextFile;
FPlotaPontos: boolean; //indica qdo o G.P.S entra em atividade
GPSOK: integer; // =1 plota no gráfico
aux: integer;

implementation
uses UCaixeiroViajante, UPilotoVirtual, URelatorio;

{$R *.dfm}

procedure TForm1.FormActivate(Sender: TObject);
var
AuxDataHora: string;

begin
GPSOK:=0;
AuxDataHora:=DateTimeToStr(now);
Delete(AuxDataHora,10,10);
LData.Caption:=AuxDataHora;
contbus:=0;
contnav:=0;
contgps:=0;
FPlotaPontos:=false;
Serial.Open;
AtualizaMan();
AtualizaLem();
end;

procedure TForm1.Plota(grla,mila,sela,grlo,milo,selo :integer);
begin
Series1.AddXY( (((grlo*60)+ milo)*60)+ selo,(((grla*60)+ mila)*60)+ sela ); // x = longitude e y = latitude
end;

procedure TForm1.AtualizaBussola();
begin
Centena.Caption:= '???';
Label2.Font.Color:=clBlack;
Label3.Font.Color:=clBlack;
Label4.Font.Color:=clBlack;
Label5.Font.Color:=clBlack;
Label6.Font.Color:=clBlack;
Label7.Font.Color:=clBlack;
Label8.Font.Color:=clBlack;
Label9.Font.Color:=clBlack;
Label10.Font.Color:=clBlack;
Label11.Font.Color:=clBlack;
Label12.Font.Color:=clBlack;
Label13.Font.Color:=clBlack;
Label14.Font.Color:=clBlack;
Label15.Font.Color:=clBlack;
Label16.Font.Color:=clBlack;
Label17.Font.Color:=clBlack;
end;

```

```

procedure TForm1.Bussola();
var
  graus,MSB,LSB,i,chk: integer;
  buss: string;
  aux: char;

begin
  chk:=0;
  AtualizaBussola();
  buss:=Copy(Palavra,6,5);
  If (Pos('$B',buss) <> 0) then //bússola presente no pacote
  begin
    for i:=2 to 4 do chk:=(chk xor integer(buss[i]));
    If ( integer(buss[5]) <> chk ) then //dado corrompido
    begin
      inc(contbus);
      LChkBus.Caption:=IntToStr(contbus);
    end
    else
    begin
      aux:=buss[3];
      MSB:=integer(aux);
      aux:=buss[4];
      LSB:=integer(aux);
      graus:=MSB*256 + LSB;
      If (graus > 359) then Centena.Caption:='xxx';

      If (((graus>=348) and (graus<=359)) or ((graus<=12))) then Label2.Font.Color:=clRed (N)
      else
      If ((graus>=79) and (graus<=101)) then Label5.Font.Color:=clRed (E)
      else
      If ((graus>=169) and (graus<=191)) then Label4.Font.Color:=clRed (S)
      else
      If ((graus>=259) and (graus<=281)) then Label3.Font.Color:=clRed (O)
      else
      If ((graus>=34) and (graus<=56)) then Label6.Font.Color:=clRed (NE)
      else
      If ((graus>=124) and (graus<=146)) then Label7.Font.Color:=clRed (SE)
      else
      If ((graus>=214) and (graus<=236)) then Label9.Font.Color:=clRed (SO)
      else
      If ((graus>=304) and (graus<=326)) then Label8.Font.Color:=clRed (NO)
      else
      If ((graus>=13) and (graus<=33)) then Label10.Font.Color:=clRed (NNE)
      else
      If ((graus>=57) and (graus<=78)) then Label11.Font.Color:=clRed (ENE)
      else
      If ((graus>=102) and (graus<=123)) then Label12.Font.Color:=clRed (ESE)
      else
      If ((graus>=147) and (graus<=168)) then Label13.Font.Color:=clRed (SSE)
      else
      If ((graus>=192) and (graus<=213)) then Label14.Font.Color:=clRed (SSO)
      else
      If ((graus>=237) and (graus<=258)) then Label15.Font.Color:=clRed (OSO)
      else
      If ((graus>=282) and (graus<=303)) then Label16.Font.Color:=clRed (ONO)
      else
      If ((graus>=327) and (graus<=347)) then Label17.Font.Color:=clRed; (NNO)
    end
  end
  If (graus < 10) then Centena.Caption:='00' + IntToStr(graus)
  else
  If ((graus > 9) and (graus < 100)) then Centena.Caption:='0' + IntToStr(graus)
  else
  Centena.Caption:=IntToStr(graus);
end;
end;

procedure TForm1.AtualizaLem();
begin
  BE1.Brush.Color:=ClSilver;
  BE2.Brush.Color:=ClSilver;
  BE3.Brush.Color:=ClSilver;
  BB1.Brush.Color:=ClSilver;
  BB2.Brush.Color:=ClSilver;
  BB3.Brush.Color:=ClSilver;
end;

procedure TForm1.AtualizaMan();
begin
  BF1.Brush.Color:=ClSilver;
  BF2.Brush.Color:=ClSilver;
  BF3.Brush.Color:=ClSilver;
  BR1.Brush.Color:=ClSilver;
  BR2.Brush.Color:=ClSilver;
  BR3.Brush.Color:=ClSilver;
end;

procedure TForm1.ManLem();
var
  l,m :char;
  aux1,aux2,i,chk: integer;
  ManLem: string;

begin
  chk:=0;
  ManLem:=Copy(Palavra,1,5);
  If ( (Pos('$N',ManLem) <> 0) or (Pos('$L',ManLem) <> 0) )then // navegação presente no pacote
  begin
    for i:=2 to 4 do chk:=(chk xor integer(ManLem[i]));
    If ( integer(ManLem[5]) <> chk ) then //dado corrompido
    begin

```

```
inc(contnav);
LChkNav.Caption:=IntToStr(contnav);
end
else
begin
l:=ManLem[3];
aux1:=integer(l);
Leme.Caption:=IntToStr(aux1);
m:=ManLem[4];
aux2:=integer(m);
Manche.Caption:=IntToStr(aux2);
Case (aux1) of
145..255: begin
    AtualizaLem();
    BB3.Brush.Color:=CINavy;
    BB2.Brush.Color:=CINavy;
    BB1.Brush.Color:=CINavy;
    end;

118..135: begin
    AtualizaLem();
    BB2.Brush.Color:=CINavy;
    BB1.Brush.Color:=CINavy;
    end;

93..107: begin
    AtualizaLem();
    BB1.Brush.Color:=CINavy;
    end;

64..77: begin
    AtualizaLem();
    end;

52..61: begin
    AtualizaLem();
    BE1.Brush.Color:=CILime;
    end;

40..48: begin
    AtualizaLem();
    BE1.Brush.Color:=CILime;
    BE2.Brush.Color:=CILime;
    end;

0..34: begin
    AtualizaLem();
    BE1.Brush.Color:=CILime;
    BE2.Brush.Color:=CILime;
    BE3.Brush.Color:=CILime;
    end;
end;

Case (aux2) of
0..82: begin
    AtualizaMan();
    BR3.Brush.Color:=CINavy;
    BR2.Brush.Color:=CINavy;
    BR1.Brush.Color:=CINavy;
    end;

83..99: begin
    AtualizaMan();
    BR2.Brush.Color:=CINavy;
    BR1.Brush.Color:=CINavy;
    end;

100..119: begin
    AtualizaMan();
    BR1.Brush.Color:=CINavy;
    end;

121..137: begin
    AtualizaMan();
    end;

142..157: begin
    AtualizaMan();
    BF1.Brush.Color:=CILime;
    end;

158..170: begin
    AtualizaMan();
    BF1.Brush.Color:=CILime;
    BF2.Brush.Color:=CILime;
    end;

174..255: begin
    AtualizaMan();
    BF1.Brush.Color:=CILime;
    BF2.Brush.Color:=CILime;
    BF3.Brush.Color:=CILime;
    end;
end;
end;
end;
end;
```

```

procedure TForm1.Garmin();
var
i,aux,chk: integer;
auxdatahora: string;

begin
chk:=0;
NMEA:=Copy(Palavra,11,23);
If ( Pos('$G',NMEA) <> 0 ) then // GPS presente no pacote
begin
for i:=2 to 18 do chk:=(chk xor integer(NMEA[i]));
If ( integer(NMEA[21]) <> chk ) then // dado corrompido ou inválido
begin
If (NMEA[4] = 'V') then
begin
Posicao.Caption:='Posição Não Confiável';
Posicao.Font.Color:=clRed;
end
else
begin
inc(contgps); //dado corrompido mostra na tela
LChkGps.Caption:=IntToStr(contgps);
end;
end
else
begin //Posição confiável
Latitude.Caption:=NMEA[10]; // orientação da latitude
Longitude.Caption:=NMEA[16]; // orientação da longitude

aux:=integer(NMEA[6]);
If (aux < 10 ) then LatG.Caption:='00' + IntToStr(aux)
else
If (aux < 100 ) then LatG.Caption:='0' + IntToStr(aux)
else
LatG.Caption:=IntToStr(aux);

aux:=integer(NMEA[7]);
If (aux < 10) then LatM.Caption:='0' + IntToStr(aux) + ""
else
LatM.Caption:=IntToStr(aux) + "";

aux:=integer(NMEA[8]);
If (aux < 10) then LatS.Caption:='0' + IntToStr(aux) + ""
else
LatS.Caption:=IntToStr(aux) + "";

aux:=integer(NMEA[12]);
If (aux < 10 ) then LonG.Caption:='00' + IntToStr(aux)
else
If (aux < 100 ) then LonG.Caption:='0' + IntToStr(aux)
else
LonG.Caption:=IntToStr(aux);

aux:=integer(NMEA[13]);
If (aux < 10) then LonM.Caption:='0' +IntToStr(aux) + ""
else
LonM.Caption:=IntToStr(aux) + "";

aux:=integer(NMEA[14]);
If (aux < 10) then LonS.Caption:='0' + IntToStr(aux) + ""
else
LonS.Caption:=IntToStr(aux) + "";

aux:=integer(NMEA[18]);
Velocidade.Caption:=IntToStr(aux);

AuxDataHora:=DateTimeToStr(now);
Delete(AuxDataHora,1,10); // horário
Horario.Caption:=AuxDataHora;

Posicao.Caption:='Posição Válida';
Posicao.Font.Color:=clGreen;

If (FPlotaPontos = true) then Plota(integer(NMEA[6]),integer(NMEA[7]),integer(NMEA[8]),integer(NMEA[12]),integer(NMEA[13]),integer(NMEA[14]));
inc(GPSOK); // =1 flag pra começar a plotar no gráfico origem e destino
end;
end;
end;

procedure TForm1.EnviaClick(Sender: TObject);
var
PegaCarac: string;

begin
Serial.Write('@');
Repeat
PegaCarac:=Serial.Read;
Until (PegaCarac = '!');
ArqLog.Lines[contador]:=PegaCarac;
Timer.Enabled:=true;
end;

procedure TForm1.TimerPacDados(Sender: TObject);
var
PosCarac,i: integer;
lag1,lag2,lag3,log1,log2,log3: integer;
chkdest: integer;
aux: real;

```

```

begin
Palavra:=Palavra + Serial.Read;
PosCarac:=Pos('%%',Palavra);
If (PosCarac <> 0) then
begin
inc(contador);
For i:=1 to (PosCarac + 1) do ArqLog.Lines[contador]:=ArqLog.Lines[contador] + Palavra[i];
LogCont.Caption:=IntToStr(contador);

If (FPlotaPontos = true) then Writeln(Arquivo,Palavra);           //grava pacotes após achar validade do GPS

Bussola();
ManLem();
Garmin();
Delete(Palavra,1,33); // tamanho do pacote da telemetria
If (Palavra = 'V') then // piloto virtual
begin
Timer.Enabled:=false;
Form3.ShowModal;
end;
If (GPSOK = 1) then
begin
Timer.Enabled:=false;
Form2.ShowModal; //pega os pontos de coleta de dados hidrológicos
Series2.AddXY( ( ( integer(NMEA[12])*60)+integer(NMEA[13]))*60)+ integer(NMEA[14]),( ( integer(NMEA[6])*60)+integer(NMEA[7])) *60)+ integer(NMEA[8] );

// x = longitude e y = latitude <<< PONTO ORIGEM >>>

Series2.AddXY(Matriz[StrToInt(caminho[2])).Lon,Matriz[StrToInt(caminho[2])).Lat);

// <<< PONTO DESTINO >>> (FORM2)

Writeln(Arquivo,NMEA[6] + NMEA[7] + NMEA[8] + NMEA[12] + NMEA[13] + NMEA[14] );

//grava ponto de origem

FPlotaPontos:=true; //começa a plotar todos os pacotes

// transformação dos segundos totais do G.P.S. para graus, minutos e segundos

lag1:= trunc( (Matriz[StrToInt(caminho[2])).Lat)/3600 );
lag2:= trunc(((Matriz[StrToInt(caminho[2])).Lat)/3600) - (trunc(((Matriz[StrToInt(caminho[2])).Lat)/3600) ) ))*60;
aux:= (((Matriz[StrToInt(caminho[2])).Lat)/3600) - (trunc(((Matriz[StrToInt(caminho[2])).Lat)/3600) ) ))*60- lag2;
lag3:=trunc(aux*60);

log1:= trunc( (Matriz[StrToInt(caminho[2])).Lon)/3600 );
log2:= trunc(((Matriz[StrToInt(caminho[2])).Lon)/3600) - (trunc(((Matriz[StrToInt(caminho[2])).Lon)/3600) ) ))*60;
aux:= (((Matriz[StrToInt(caminho[2])).Lon)/3600) - (trunc(((Matriz[StrToInt(caminho[2])).Lon)/3600) ) ))*60- log2;
log3:=trunc(aux*60);

chkdest:= ( ( ( ( integer('G') xor lag1) xor lag2) xor lag3) xor log1) xor log2) xor log3);

Serial.Write('$G' + char(lag1) + char(lag2) + char(lag3) + char(log1) + char(log2) + char(log3) + char(chkdest));

//Pto destino GMSGMSH (lat/long)

Writeln(Arquivo, char(lag1) + char(lag2) + char(lag3) + char(log1) + char(log2) + char(log3) );

//grava ponto destino

TimerAck.Enabled:=true;
end;
end;
end;

procedure TForm1.NovaPesquisaClick(Sender: TObject);
begin
Form2.ShowModal;
end;

procedure TForm1.NovoClick(Sender: TObject);
var
Texto: string;
begin
If NovoArquivo.Execute then
begin
AssignFile(Arquivo,NovoArquivo.FileName + '.txt');
Rewrite(Arquivo);
Texto:=InputBox('Descrição', 'Digite uma breve descrição sobre a nova pesquisa,');
Writeln(Arquivo, 'Telemetria de Dados e Imagens');
Writeln(Arquivo,Texto);
Writeln(Arquivo,DateTimeToStr(now));
Form3.ShowModal();
end;
end;

procedure TForm1.Inicializa(Sender: TObject);
begin
contador:=0;
end;

procedure TForm1.sireneClick(Sender: TObject);
begin
Shape2.Visible:=true;
PlaySound('c:\Windows XP Exclamation.wav',0,SND_FILENAME); //som de alerta
Serial.Write(""); // início de comando manual
end;

```

```
procedure TForm1.BtPararClick(Sender: TObject);
begin
  Serial.Close;
  CloseFile(Arquivo);
end;

procedure TForm1.TimerAckTimer(Sender: TObject);
var
  ack: string;

begin
  ack:=Form1.Serial.Read;
  If (ack = '!') then
  begin
    TimerAck.Enabled:=false;
    Timer.Enabled:=true;
  end;
end;

procedure TForm1.SairClick(Sender: TObject);
begin
  Form1.Close;
end;

procedure TForm1.AbrirClick(Sender: TObject);
var
  x: string;

begin
  If AbreArquivo.Execute then
  begin
    AssignFile(Arquivo,AbreArquivo.FileName);
    Reset(Arquivo);
    Readln(Arquivo,x); //cabecalho
    If (x <> 'Telemetria de Dados e Imagens') then
    begin
      If MessageDlg('Arquivo inválido!!!',mtError,[mbOk],0) = mrOk then CloseFile(Arquivo);
    end
    else Form4.ShowModal;
  end;
end;

procedure TForm1.NavegVisualClickSeries(Sender:TCustomChart;Series:TChartSeries;ValueIndex:Integer;
                                         Button:TMouseButton;Shift:TShiftState;X,Y:integer);
begin
  ShowMessage(' Clicked Series: '+Series.Name+' at point: '+ Floattostr(Series.XValue[valueindex]) + ',' + Floattostr(Series.YValue[valueindex]));
end;

end.
```

unit UCaixeiroViajante;**interface****uses**

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, Math, TeEngine, Series, ExtCtrls, TeeProcs,
Chart, TeeFunci;

type

```
TForm2 = class(TForm)
  Inserir: TButton;
  LatG: TEdit;
  Label1: TLabel;
  Label3: TLabel;
  LatM: TEdit;
  LatS: TEdit;
  LonG: TEdit;
  LonM: TEdit;
  LonS: TEdit;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  ComboBox1: TComboBox;
  ComboBox2: TComboBox;
  Label2: TLabel;
  Edit1: TEdit;
  Rotear: TBitBtn;
  Maxpts: TButton;
  Chart1: TChart;
  Series1: TLineSeries;
  TeeFunction1: TAddTeeFunction;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label11: TLabel;
  F2BitFechar: TButton;
  Label7: TLabel;
  procedure InserirClick(Sender: TObject);
  procedure RotearClick(Sender: TObject);
  procedure MaxptsClick(Sender: TObject);
  function TruncaReal(x: real): real;
  procedure Caixeiro_Viajante();
  procedure Annealing();
  procedure FormActivate(Sender: TObject);
  procedure F2BitFecharClick(Sender: TObject);
```

private

```
{ Private declarations }
```

public

```
end;
```

var

```
Form2: TForm2;
Matriz: array of
  record
    Lat: real;
    Lon: real;
  end;
caminho,rota: string; //caminho é indice para calcular custo final
Matrizao: array of array of real;
pontos,aux_cont: integer;
custo1,custo2,energia: real; //calculam custo local e custo real do caminho mínimo
```

implementation

```
uses UTelemetria;
```

```
{SR *.dfm}
```

procedure TForm2.FormActivate(Sender: TObject);

```
begin
  F2BitFechar.Enabled:=false;
  Form2.ClientHeight:=152;
  Form2.ClientWidth:=551;
  LatG.Clear;
  LatG.Clear; // limpa os Edit's para novos dados
  LatM.Clear;
  LatS.Clear;
  LonG.Clear;
  LonM.Clear;
  LonS.Clear;
  Edit1.Clear;
  Rotear.Visible:=false;
  Inserir.Visible:=false;
  Maxpts.Enabled:=true;
  Label9.Caption:= '?????';
  Label11.Caption:= '?????';
end;
```

function TForm2.TruncaReal(x: real): real;

```
var
  aux: integer;
  num: string;

begin
  num:=FloatToStr(x);
  aux:=Pos(',',num); //retorna posição do ponto flutuante
  TruncaReal:=StrToFloat(Copy(num,1,aux+2));
end;
```



```

procedure TForm2.Caixaiero_Viajante();                                //define caminho probabilístico
var
  aux,i,j,x: integer;
  carac: string;

begin
  custo2:=0; //inicializa custo
  aux:=0; //conta os pontos já inseridos no percurso
  caminho:=0; //primeiro ponto é origem
  rota:=0; //mostra caminho ótimo na tela
  Randomize;
  For i:=1 to pontos*10 do
    begin
      x:=random(pontos);
      carac:=lnTToStr(x);
      If ( ((Pos(carac,caminho))=0) and (carac <> '0') ) then
        begin
          caminho:=caminho + carac;
          rota:=rota + ' -> ' + carac;
          inc(aux);
        end;
      if(aux = pontos -1) then break;
    end;
  caminho:=caminho + '0'; // último ponto é destino
  rota:=rota + ' -> 0';
  For i:=1 to pontos do custo2:=custo2 + Matrizao[StrToInt(caminho[i]),StrToInt(caminho[i+1])];
end;

procedure TForm2.Annealing();
var
  prob: real; //variação de energia e probabilidade
  aleatorio,temp,j: integer;

begin
  Series1.Clear;
  Caixaiero_Viajante; //primeira solução aleatória
  custo1:=custo2;
  For temp:=pontos*10 downto 1 do
    begin
      For j:=1 to (pontos + 1) do
        begin
          Caixaiero_Viajante;
          energia:=(custo2 - custo1);
          If (energia <= 0) then // minimizou o custo
            begin
              custo1:=custo2; //aceita o custo
              Series1.Add(custo1,",clRed); //plota no gráfico
            end
          else //análise probabilística
            begin
              prob:=exp(-energia/temp)*1000;
              aleatorio:=random(pontos*5000);
              If (aleatorio < prob) then //aceita a perturbação
                begin
                  custo1:=custo2; //plota no gráfico
                  Series1.Add(custo1,",clNavy);
                end;
            end;
        end;
      end;
    Label9.Caption:=rota;
    Label11.Caption:=FloatToStr(custo1);
  end;

procedure TForm2.MaxptsClick(Sender: TObject);
begin
  Inserir.Visible:=true; //habilita botão de Inserir
  pontos:=StrToInt(Edit1.Text) +1; //pega qtdade de pontos + origem
  SetLength(Matriz,pontos); //alocação dinâmica
  SetLength(Matrizao,pontos,pontos); //alocação dinâmica

  Matriz[0].Lat:=((integer(NMEA[6])*60)+ integer(NMEA[7]))*60 + integer(NMEA[8]) );
                                                                    //pega do G.P.S. ponto origem do sistema

  Matriz[0].Lon:=((integer(NMEA[12])*60)+ integer(NMEA[13]))*60 + integer(NMEA[14]) );

  aux_cont:=1; //contador
  Maxpts.Enabled:=false; // desabilita bt de pontos
end;

procedure TForm2.InserirClick(Sender: TObject);
begin
  Matriz[aux_cont].Lat:=(( ( StrToInt(LatG.Text)*60)+ StrToInt(LatM.Text))*60+ StrToInt(LatS.Text) );

  Matriz[aux_cont].Lon:=(( ( StrToInt(LonG.Text)*60)+ StrToInt(LonM.Text))*60+ StrToInt(LonS.Text) );
  inc(aux_cont);
  If (aux_cont=pontos) then
    begin
      Rotear.Visible:=true;
      Inserir.Enabled:=false;
    end;
  LatG.Clear;
  LatM.Clear; // limpa os Edit's para novos dados
  LatS.Clear;

```

```
LonG.Clear;
LonM.Clear;
LonS.Clear;
end;

procedure TForm2.RotearClick(Sender: TObject);
var
  i,j: integer;
  valor: real;
begin
  For i:=0 to pontos-1 do
    For j:=0 to pontos-1 do
      //coloca os custos das
      //distâncias na tabela de custos
      If (i=j) then Matrizao[i,j]:=1000000 // penalização
      else
        begin
          valor:=Sqrt(((Matriz[j].Lat - Matriz[i].Lat)*
            (Matriz[j].Lat - Matriz[i].Lat))+
            ((Matriz[j].Lon - Matriz[i].Lon)*
            (Matriz[j].Lon - Matriz[i].Lon))); //mínima distância entre dois pontos
          Matrizao[i,j]:=TruncaReal(valor);
        end;
    Annealing;
    Form2.ClientHeight:=534;
    Inserir.Visible:=false;
    F2BtFechar.Enabled:=true;
  end;

procedure TForm2.F2BtFecharClick(Sender: TObject);
begin
  Form2.Close;
end;

end.
```

unit UPilotoVirtual;**interface****uses**

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, QCom32;

type

TForm3 = class(TForm)
Leme: TRadioGroup;
Manche: TRadioGroup;
BtAtuar: TButton;
BtFechar: TButton;
BtDesligar: TButton;
Aguarda: TLabel;
TimerAck: TTimer;
TimerRecGer: TTimer;
T100ms: TTimer;
BtLigar: TButton;

procedure BtFecharClick(Sender: TObject);
procedure BtAtuarClick(Sender: TObject);
procedure IniciaEmergencia(Sender: TObject);
procedure BtDesligarClick(Sender: TObject);
procedure AcaoEmergencia(acao: char; CLeme,CManche: integer);
procedure TimerAckTimer(Sender: TObject);
procedure TimerRecGerTimer(Sender: TObject);
procedure BtLigarClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;
ack: string;
StrGer: string;
chk: integer;
Tms: boolean;

//comando para resetar a embarcação - leme ao centro e manche no pto morto

implementation

uses UTelemetria;

{\$R *.dfm}

procedure TForm3.IniciaEmergencia(Sender: TObject);

begin

Aguarda.Visible:=true;
BtAtuar.Visible:=false;
BtLigar.Visible:=false;
BtDesligar.Visible:=false;
Leme.Visible:=false;
Manche.Visible:=false;
AcaoEmergencia('A',3,3);
TimerAck.Enabled:=true;

//ATUA LEME AO CENTRO E MANCHE NO PTO. MORTO

end;

procedure TForm3.TimerAckTimer(Sender: TObject);

begin

ack:=ack + Form1.Serial.Read;
If ((Pos('E!',ack)) <> 0) then
begin
TimerAck.Enabled:=false;
TimerRecGer.Enabled:=true;
StrGer:='';
ack:='';
end;

end;

procedure TForm3.BtAtuarClick(Sender: TObject);

begin

AcaoEmergencia('A',Leme.ItemIndex,Manche.ItemIndex);

// atuar na embarcação

end;

procedure TForm3.BtLigarClick(Sender: TObject);

begin

AcaoEmergencia('I',Leme.ItemIndex,Manche.ItemIndex);

//ligar motor de popa

end;

procedure TForm3.BtDesligarClick(Sender: TObject);

begin

AcaoEmergencia('S',Leme.ItemIndex,Manche.ItemIndex);

//desligar motor de popa

end;

procedure TForm3.AcaoEmergencia(acao: char; CLeme,CManche: integer);

begin

Form1.Serial.Write('W');
BtAtuar.Enabled:=false;
BtLigar.Enabled:=false;
BtDesligar.Enabled:=false;
Leme.Enabled:=false;
Manche.Enabled:=false;
chk:=integer(acao) xor CLeme xor CManche;
Form1.Serial.Write('\$' + acao + IntToStr(CLeme) + IntToStr(CManche) + char(chk));
TimerAck.Enabled:=true;

// indica piloto virtual

end;

```
procedure TForm3.TimerRecGerTimer(Sender: TObject);
begin
  StrGer:=StrGer + Form1.Serial.Read;
  If (length(StrGer) >= 5) then
  begin
    TimerRecGer.Enabled:=false;
    chk:=0;
    chk:=chk xor integer(StrGer[2]) xor integer(StrGer[3]) xor integer(StrGer[4]);
    If (integer(StrGer[5]) = chk) then
    begin
      Aguarda.Visible:=false;

      BtAtuar.Visible:=true;
      BtLigar.Visible:=true;
      BtDesligar.Visible:=true;
      Leme.Visible:=true;
      Manche.Visible:=true;

      BtAtuar.Enabled:=true;
      BtLigar.Enabled:=true;
      BtDesligar.Enabled:=true;
      Leme.Enabled:=true;
      Manche.Enabled:=true;

      Case integer(StrGer[3]) of //Valor para leme
        195..255: Leme.ItemIndex:=0;
        144..164: Leme.ItemIndex:=1;
        103..123: Leme.ItemIndex:=2;
        66..81: Leme.ItemIndex:=3;
        51..65: Leme.ItemIndex:=4;
        35..50: Leme.ItemIndex:=5;
        0..30: Leme.ItemIndex:=6;
      end;

      Case integer(StrGer[4]) of //Valor para manche
        180..255: Manche.ItemIndex:=6;
        158..175: Manche.ItemIndex:=5;
        141..157: Manche.ItemIndex:=4;
        122..140: Manche.ItemIndex:=3;
        105..121: Manche.ItemIndex:=2;
        83..103: Manche.ItemIndex:=1;
        0..75: Manche.ItemIndex:=0;
      end;
    end;
  end;
end;

procedure TForm3.BtFecharClick(Sender: TObject);
begin
  Form1.Serial.Write('X'); //representa fim do comando manual
  Form3.Close;
end;

end.
```

unit URelatorio;**interface****uses**

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, TeEngine, Series, ExtCtrls, TeeProcs, Chart;

type

```
TForm4 = class(TForm)
  LDescricao: TLabel;
  LData: TLabel;
  Label1: TLabel;
  BtFechar: TButton;
  Chart1: TChart;
  Chart2: TChart;
  Chart3: TChart;
  Series1: TBarSeries;
  Series2: TBarSeries;
  Series3: TBarSeries;
  Chart4: TChart;
  Series5: TPointSeries;
  Series4: TPointSeries;
  Label2: TLabel;
  procedure IniciaForm4(Sender: TObject);
  procedure BtFecharClick(Sender: TObject);
  procedure Plota(grla,mila,sela,grlo,milo,selo :integer);
```

private

```
{ Private declarations }
```

public

```
{ Public declarations }
```

```
end;
```

var

```
Form4: TForm4;
bussola,leme,manche: integer;
x: string;
```

implementation

```
uses UTelemetria;
```

```
{$R *.dfm}
```

procedure TForm4.Plota(grla,mila,sela,grlo,milo,selo :integer);

```
begin
  Series4.AddXY( (((grlo*60)+ milo)*60)+ selo,(((grla*60)+ mila)*60)+ sela ); // x = longitude e y = latitude
end;
```

procedure TForm4.IniciaForm4(Sender: TObject);

```
var
  S,SSO,SO,OSO,O,ONO,NO,NNO,N,NNE,NE,E,ENE,E,ESE,SE,SSE: integer; //bússola
  B3,B2,B1,Central,E1,E2,E3: integer; //leme
  R3,R2,R1,Morto,F1,F2,F3: integer; //manche
tam: integer;
begin
  tam:=0;

  S:=0;SSO:=0;SO:=0;OSO:=0;O:=0;ONO:=0;NO:=0;NNO:=0;N:=0;NNE:=0;NE:=0;ENE:=0;E:=0;ESE:=0;SE:=0;SSE:=0;
  B3:=0;B2:=0;B1:=0;Central:=0;E1:=0;E2:=0;E3:=0;
  R3:=0;R2:=0;R1:=0;Morto:=0;F1:=0;F2:=0;F3:=0;
```

```
  ReadLn(Arquivo,x);
  LDescricao.Caption:=x; //descrição da coleta
```

```
  ReadLn(Arquivo,x); // data e hora da pesquisa
  LData.Caption:=x;
```

```
  ReadLn(Arquivo,x); //origem da pesquisa
```

```
  Series5.AddXY( (((integer(x[4])*60)+ integer(x[5]))*60)+ integer(x[6]),(((integer(x[1])*60)+ integer(x[2]))*60)+ integer(x[3])) );
// x = longitude e y = latitude
```

```
  ReadLn(Arquivo,x); //destino da pesquisa
```

```
  Series5.AddXY( (((integer(x[4])*60)+ integer(x[5]))*60)+ integer(x[6]),(((integer(x[1])*60)+ integer(x[2]))*60)+ integer(x[3])) );
// x = longitude e y = latitude
```

```
While not EOF(Arquivo) do
```

```
begin
  ReadLn(Arquivo,x);
```

```
  If ( Pos('%%',x) <> 0 ) then //pacote válido
```

```
begin
  Series4.AddXY( (((integer(x[22])*60)+ integer(x[23]))*60)+ integer(x[24]),(((integer(x[16])*60)+ integer(x[17]))*60)+ integer(x[18])) );
// x = longitude e y = latitude
```

```
inc(tam);
Label2.Caption:=IntToStr(tam);
```

```
bussola:=(integer(x[8])*256)+ integer(x[9]); //MSB e LSB
Case (bussola) of
  348..359: inc(N);
  0..12: inc(N);
  13..33: inc(NNE);
  34..56: inc(NE);
```

```

57..78: inc(ENE);
79..101: inc(E);
102..123: inc(ESE);
124..146: inc(SE);
147..168: inc(SSE);
169..191: inc(S);
192..213: inc(SSO);
214..236: inc(SO);
237..258: inc(OSO);
259..281: inc(O);
282..303: inc(ONO);
304..326: inc(NO);
327..347: inc(NNO);
end;

leme:=integer(x[3]);
manche:=integer(x[4]);

Case (leme) of
  195..255: inc(B3);
  144..164: inc(B2);
  103..123: inc(B1);
  66..81: inc(Central);
  51..65: inc(E1);
  35..50: inc(E2);
  0..30: inc(E3);
end;

Case (manche) of
  180..255: inc(R3);
  158..175: inc(R2);
  141..157: inc(R1);
  122..140: inc(Morto);
  105..121: inc(F1);
  83..103: inc(F2);
  0..75: inc(F3);
end;
end; //pacote válido

end; //fim de arquivo

Series1.AddBar(B3,'B3',clLime); //HISTOGRAMA DE LEME
Series1.AddBar(B2,'B2',clLime);
Series1.AddBar(B1,'B1',clLime);
Series1.AddBar(Central,'Central',clred);
Series1.AddBar(E1,'E1',clNavy);
Series1.AddBar(E2,'E2',clNavy);
Series1.AddBar(E3,'E3',clNavy);

Series2.AddBar(R3,'R3',clLime); //HISTOGRAMA DO MANCHE
Series2.AddBar(R2,'R2',clLime);
Series2.AddBar(R1,'R1',clLime);
Series2.AddBar(Morto,'Morto',clred);
Series2.AddBar(F1,'F1',clNavy);
Series2.AddBar(F2,'F2',clNavy);
Series2.AddBar(F3,'F3',clNavy);

Series3.AddBar(S,'S',clRed); //HISTOGRAMA DA BÚSSOLA
Series3.AddBar(SSO,'SSO',clYellow);
Series3.AddBar(SO,'SO',clBlue);
Series3.AddBar(OSO,'OSO',clYellow);
Series3.AddBar(O,'O',clRed);
Series3.AddBar(ONO,'ONO',clYellow);
Series3.AddBar(NO,'NO',clBlue);
Series3.AddBar(NNO,'NNO',clYellow);
Series3.AddBar(N,'N',clRed);
Series3.AddBar(NNE,'NNE',clYellow);
Series3.AddBar(NE,'NE',clBlue);
Series3.AddBar(ENE,'ENE',clYellow);
Series3.AddBar(E,'E',clRed);
Series3.AddBar(ESE,'ESE',clYellow);
Series3.AddBar(SE,'SE',clBlue);
Series3.AddBar(SSE,'SSE',clYellow);

end;

procedure TForm4.BitFecharClick(Sender: TObject);
begin
  Form4.Close;
end;

end.

```

ANEXO B – Sistema de Gerenciamento

```
#####
PROJETO.....:  TELEMETRIA DE DADOS E IMAGENS PARA
                PLATAFORMA AUTÔNOMA PARA COLETA DE
                DADOS HIDROLÓGICOS

DESENVOLVIDO POR.....:  ALEX AYRES STAVARENGO
VERSÃO.....:  GER22
DATA.....:  01/06/2006

OBJETIVOS DO PROGRAMA:

- ALGORITMO DE NAVEGAÇÃO (ATUA SOBRE A EMBARCAÇÃO INDIRETAMENTE)
- REQUISITA DADOS DOS SENSORES PARA O SISTEMA DE POSICIONAMENTO
- ENVIA O PACOTE DE TELEMETRIA DE DADOS PARA ESTAÇÃO TERRESTRE
#####
```

<<< DESCRIÇÃO DAS ROTINAS UTILIZADAS PELO SISTEMA >>>

TEMP1 -> Rotina de atraso de tempo de aprox. 1 segundo

TEMP002-> Rotina de atraso de tempo de aprox. 2 milissegundos

ENVPOS -> Canal serial transmissor implementado que envia 1 byte para o sistema de gerenciamento
- Entrada: byte a ser transmitido colocado em Acc

RECD -> Canal serial receptor implementado que recebe 1 byte do sistema de posicionamento
- Saída: byte recebido na variável RECEB

PACCOM -> Envia um pacote de dados para o transceiver através de canal serial próprio
- Saída: \$NLMX\$B\$SDX\$G,A,GMS,X,GMS,Y,V**H%%
- Posição na memória: 10H A 1FH - 30H A 40H (33 bytes)

```
'$' - demarcador
'N' - indicador de navegação
'L' - valor digital do leme
'M' - valor digital do manche
'X' - checksum ( menos caracter '$')
'$' - demarcador
'B' - indicador de bússola
'S' - byte mais significativo da bússola
'D' - byte menos significativo da bússola
'X' - checksum ( menos caracter '$')
'$' - demarcador
'G' - indicador do GPS
',' - separador
'A' - caracter de estado de funcionamento:
    A: posição válida
    V: posição inválida
    T: GPS não encontrado (sem alimentação)
    X: erro de checksum
',' - separador
'GMS' - graus, minutos e segundos da latitude em valores reais
',' - separador
'X' - orientação da latitude (Norte ou Sul)
',' - separador
'GMS' - graus, minutos e segundos da latitude em valores reais
',' - separador
'Y' - orientação da longitude (oeste e leste)
',' - separador
'V' - velocidade da embarcação em nós
'***' - acusa que próximo caracter é o checksum
'H' - checksum ( menos caracter '$')
'%%' - indica fim do pacote de dados
```

PACPOS -> Recebe um pacote de dados do sistema de posicionamento através do uso da rotina RECD
- Saída: \$NLMX\$B\$SDX\$G,A,GMS,X,GMS,Y,V**H%%
- Posição na memória: 10H A 1FH - 30H A 40H (33 bytes)

CHKPOS -> Recebe ack do sistema de posicionamento
- Saída: F_ACKPOS = 1 mostra pacote enviado intacto
F_ACKPOS = 0 mostra pacote enviado corrompido

LEITPOS -> Envia pacote de dados para sistema de posicionamento pedindo leitura do estado do manche e leme
- Saída: \$LXYH enviado byte a byte

```
'$' - demarcador
'L' - indicador de leitura ou atuação
'X' - valor do leme
'Y' - valor do manche
'H' - checksum ( menos caracter '$')
```

CHECKSUM -> Confere sobre o pacote de dados recebido do sistema de posicionamento a validade de leitura de cada sensor
- Saída: FLAG_AD = 0 dado válido do leme/manche
FLAG_AD = 1 dado corrompido

```
FLAG_BUS = 0 dado válido da bússola
FLAG_BUS = 1 dado corrompido
```

```
FLAG_GPS = 0 dado válido do GPS
FLAG_GPS = 1 dado corrompido
```

LATAUX -> Faz um swap das variáveis de graus, minutos e segundos da latitude para que a operação não seja negativa (ponto destino > ponto atual)

- SUBGLAT -> Faz a subtração dos graus da latitude entre ponto atual e ponto destino
- Saída: variável DIGLAT
- SUBMLAT -> Faz a subtração dos minutos da latitude entre ponto atual e ponto destino
- Saída: variável DIMLAT
- SUBSLAT -> Faz a subtração dos segundos da latitude entre ponto atual e ponto destino
- Saída: variável DISLAT
- LONAUX -> Faz um swap das variáveis de graus, minutos e segundos da longitude para que a operação não seja negativa (ponto destino > ponto atual)
- SUBGLON -> Faz a subtração dos graus da longitude entre ponto atual e ponto destino
- Saída: variável DIGLON
- SUBMLON -> Faz a subtração dos minutos da longitude entre ponto atual e ponto destino
- Saída: variável DIMLAT
- SUBSLON -> Faz a subtração dos segundos da longitude entre ponto atual e ponto destino
- Saída: variável DISLON
- DIFLATANDES -> Calcula a distância da latitude entre o ponto atual e ponto destino (em módulo)
- Saída: graus em DIGLAT
minutos em DIMLAT
segundos em DISLAT
ZEROLAT = 0 há distância entre os pontos
ZEROLAT = 1 pontos com mesma coordenada
- DIFLONANDES -> Calcula a distância da longitude entre o ponto atual e ponto destino (em módulo)
- Saída: graus em DIGLON
minutos em DIMLON
segundos em DISLON
ZEROLON = 0 há distância entre os pontos
ZEROLON = 1 pontos com mesma coordenada
- INCDECLAT -> Analisa se ocorreu incremento ou decremento da latitude entre o ponto atual e ponto anterior
- Saída: FIDLAT = 0 então diminuiu a latitude
FIDLAT = 1 então aumentou a latitude
- INCDECLON -> Analisa se ocorreu incremento ou decremento da longitude entre o ponto atual e ponto anterior
- Saída: FIDLON = 0 então diminuiu a longitude
FIDLON = 1 então aumentou a longitude
- CASALAT -> Calcula o saldo da latitude entre o ponto atual e o ponto anterior
- Saída: FSALAT = 0 então saldo é negativo
FSALAT = 1 então saldo é positivo
- CASALON -> Calcula o saldo da longitude entre o ponto atual e o ponto anterior
- Saída: FSALON = 0 então saldo é negativo
FSALON = 1 então saldo é positivo
- DIFLATLONAT -> Calcula a diferença entre a latitude e a longitude (em módulo)
- Saída: DLALOG é a diferença em graus
DLALOM é a diferença em minutos
DLALOS é a diferença em segundos
- GIRABUSS -> Soma ao valor do giro da bússola na memória para o sentido horário (0..359 graus)
- Entrada: variável CONT (qtas vezes repete a rotina)
- Saída: GBUS1 (MLB) e GBUS2 (MSB - 1bit) na memória
- SBPG -> Subtrai valor real da bússola do valor a ser alcançado com o giro da embarcação
- Saída: F_SUBSPG = 0 valor alcançado
F_SUBSPG = 1 continuar girando a embarcação
- PTATAN -> Transforma ponto atual do G.P.S. em ponto anterior (troca de variáveis)
- MELALOBU -> Busca na memória GMS de latitude e longitude além do valor da bússola
- TELEM -> Pede leitura dos sensores para sist. Pos. e analisa checksum que vem junto ao pacote
- GERATUA -> Efetua a mundaça de rota da embarcação até encontrar o valor da bússola (nova orientação)
- PRENAV -> Realiza todos os ajustes necessários iniciais para iniciar a rotina principal de navegação do sistema
- NAVRUMO -> Encontra o rumo correto para a embarcação
- Saída: ZEROLAT = 0 não chegou ponto destino
ZEROLAT = 1 ponto atual é ponto destino (fim)
RUMCER = 0 não encontrou o rumo ainda
RUMCER = 1 encontrou rumo então fim de rotina
- NAVGRAD -> Rotina de aproximação ao ponto destino, convergindo simultaneamente as coordenadas de latitude e longitude.
Maiores detalhes é só acompanhar os comentários ao longo do seu desenvolvimento
- RPTDEST -> Recebe posição do ponto destino do P.C. a ser alcançado
- Entrada: \$GABCDEFH
- Posição na memória: 10H a 18H
- '\$' - demarcador
'G' - indicador de GPS
'A' - graus da latitude destino
'B' - minutos da latitude destino
'C' - segundos da latitude destino
'D' - graus da longitude destino
'E' - minutos da longitude destino
'F' - segundos da longitude destino
'H' - checksum (menos caracter '\$')
- Saída: valores colocados nas variáveis do pto destino; se erro de checksum envio de ack pro P.C. e pedido de retransmissão

PCGER -> Recebe caracteres do ponto destino da estação terrestre

EMERG -> Emergência da embarcação - Piloto Virtual

```
#####
# CONSTANTES:                                     #
#####

TEMPO1      EQU    240      ; CONSTANTES PARA CONTAGEM DE TEMPO
TEMPO2      EQU    240
TEMPO3      EQU    009
TEMPO4      EQU    002
TEMPO5      EQU    002
STEMP       EQU    045      ; CONSTANTE PARA USO DE 9600 BPS
STEMP2      EQU    022      ; CONSTANTE PARA USO DE 9600 BPS
PRIMEIRO    EQU    10H      ; PRIMEIRA PARTE DOS DADOS DA TELEMETRIA NA MEMÓRIA (BANCOS 2/3)
SEGUNDO     EQU    30H      ; SEGUNDA PARTE DOS DADOS DA TELEMETRIA NA MEMÓRIA

#####
# VARIÁVEIS:                                       #
#####

TXD         EQU    0B3H      ; PORTA P3.3, TX IMPLEMENTADO
RXD         EQU    0B2H      ; PORTA P3.2, RX IMPLEMENTADO

ESTADO      EQU    41H      ; INDICA SE É LEITURA, ATUAÇÃO, LIGA, DESLIGA OU EMERGENCIA

RECEB       EQU    42H      ; CARACTER RECEBIDO DO SIST. POS.

AUX         EQU    43H
AUX1        EQU    44H      ; VARIÁVEIS AUXILIARES PARA NAVEGAÇÃO, LEME E MANCHE
AUX2        EQU    45H

CONT        EQU    46H      ; CONTADOR

CHSUM       EQU    47H      ; VALOR CALCULADO DO CHECKSUM DO GPS

;*** PONTO DESTINO ***
DEGLAT      EQU    48H      ; GRAUS DA LATITUDE
DEMLAT      EQU    49H      ; MINUTOS DA LATITUDE DO GPS
DESLAT      EQU    4AH      ; MINUTOS LONGITUDE DO GPS

DEGLON      EQU    4BH      ; GRAUS DA LONGITUDE
DEMLON      EQU    4CH      ; SEGUNDOS DA LATITUDE DO GPS
DESLON      EQU    4DH      ; SEGUNDOS LONGITUDE DO GPS

;*** PONTO CORRENTE ***
GRALAT      EQU    4EH      ; GRAUS DA LATITUDE
MINLAT      EQU    4FH      ; MINUTOS DA LATITUDE DO GPS
SEGLAT      EQU    50H      ; SEGUNDOS DA LATITUDE DO GPS

GRALON      EQU    51H      ; GRAUS DA LONGITUDE
MINLON      EQU    52H      ; MINUTOS LONGITUDE DO GPS
SEGLON      EQU    53H      ; SEGUNDOS LONGITUDE DO GPS

;*** PONTO ANTERIOR ***
ANGLAT      EQU    54H      ; GRAUS DA LATITUDE
ANMLAT      EQU    55H      ; MINUTOS DA LATITUDE DO GPS
ANSLAT      EQU    56H      ; MINUTOS LONGITUDE DO GPS

ANGLON      EQU    57H      ; GRAUS DA LONGITUDE
ANMLON      EQU    58H      ; SEGUNDOS DA LATITUDE DO GPS
ANSLON      EQU    59H      ; SEGUNDOS LONGITUDE DO GPS

;*** DIFERENÇA ATUAL***
DIGLAT      EQU    5AH      ; DIFERENÇA DA LATITUDE EM GRAUS
DIMLAT      EQU    5BH      ; DIFERENÇA DA LATITUDE EM MINUTOS
DISLAT      EQU    5CH      ; DIFERENÇA DA LATITUDE EM SEGUNDOS

DIGLON      EQU    5DH      ; DIFERENÇA DA LONGITUDE EM GRAUS
DIMLON      EQU    5EH      ; DIFERENÇA DA LONGITUDE MINUTOS
DISLON      EQU    5FH      ; DIFERENÇA DA LONGITUDE EM SEGUNDOS

;*** DIFERENÇA ANTERIOR***
DAGLAT      EQU    60H      ; DIFERENÇA DA LATITUDE EM GRAUS
DAMLAT      EQU    61H      ; DIFERENÇA DA LATITUDE EM MINUTOS
DASLAT      EQU    62H      ; DIFERENÇA DA LATITUDE EM SEGUNDOS

DAGLON      EQU    63H      ; DIFERENÇA DA LONGITUDE EM GRAUS
DAMLON      EQU    64H      ; DIFERENÇA DA LONGITUDE MINUTOS
DASLON      EQU    65H      ; DIFERENÇA DA LONGITUDE EM SEGUNDOS

GBUS1       EQU    66H      ; GIRO DA BÚSSOLA
GBUS2       EQU    67H      ; PARTE ALTA DA BUSSOLA
CINT        EQU    68H      ; CONTADOR DOS BYTES RECEBIDOS PELO CANAL SERIAL PRÓPRIO

AUXNAV      EQU    69H      ; AUXILIAR QUE AJUDA NOS CÁLCULOS DE NAVEGAÇÃO

AUXGRA      EQU    6AH      ; AUXILIAR PARA GRAUS
AUXMIN      EQU    6BH      ; AUXILIAR PARA MINUTOS
AUXSEG      EQU    6CH      ; AUXILIAR PARA SEGUNDOS

DLALOG      EQU    6DH      ; DIFERENÇA ENTRE LATITUDE E LONGITUDE DO MESMO PONTO
DLALOM      EQU    6EH
DLALOS      EQU    6FH
```

```

#####
;# FLAGS: #
#####

FIDLAT EQU 00H ; FLAG DA LATITUDE - FLAGS QUE ACUSAM INCREMENTO OU DECREMENTO DE POSIÇÃO ***
FIDLON EQU 01H ; FLAG DA LONGITUDE - BITS - DECREMENTO = 0 / INCREMENTO = 1

FSALAT EQU 02H ; FLAG DA LATITUDE - FLAGS DO SALDO DA NAVEGAÇÃO ENTRE DOIS PONTOS ***
FSALON EQU 03H ; FLAG DA LONGITUDE - BITS - POSITIVO = 1 / NEGATIVO = 0

ZEROLAT EQU 04H ; = 1 LATITUDE -> PONTO ATUAL É PONTO DESTINO
ZEROLON EQU 05H ; = 1 LONGITUDE -> PONTO ATUAL É PONTO DESTINO

RUMCER EQU 06H ; 0 = RUMO NAO ENCONTRADO / 1 = RUMO CORRETO
HORANT EQU 07H ; 0 = ULTIMO GIRO ANTI-HORÁRIO / 1 = ULTIMO GIRO HORÁRIO

FLAG_AD EQU 08H ; FLAG DO A/D (ERRO = 1)
FLAG_BUS EQU 09H ; FLAG DA BÚSSOLA (ERRO = 1)
FLAG_GPS EQU 0AH ; FLAG DO GPS (ERRO = 1)

FLAG_NR EQU 0BH ; = 1 ENTÃO RETORNA AO COMEÇO DO ALGORITMO DE NAVEGAÇÃO
F_ACKPOS EQU 0CH ; ACK DO PACOTE QUE VEM DO SIST. POS. (= 1 ERRO NO PACOTE)
F_SBUSPG EQU 0DH ; SUBTRAÇÃO DO VALOR REAL DA BÚSSOLA COM O VALOR Q QUEREMOS CHEGAR
; = 0 OBJETIVO ALCANÇADO = 1 SUBTRAIR NOVAMENTE

RXSER EQU 0EH ; RECEPÇÃO DE DADO ATIVADO/DESATIVADO DO P.C.
ZERO EQU 0FH ; INDICA SE ALGUMA OPERAÇÃO DEU ZERO (INCDECLAT E INCDECLON)

#####
;# ENTRADAS E SAÍDAS: #
#####

LED EQU 0B7H ; PORTA P3.7, PARA ACIONAMENTO DO LED

#####
;# VETOR DE RESET: #
#####

ORG 00H ; ENDEREÇO DA MONTAGEM DO PROGRAMA
MOV SP,#70H ; POSICIONA APONTADOR DE PILHA
LJMP INICIO ; DESVIA PARA A ROTINA DE INICIALIZAÇÃO

#####
;# ROTINAS DE INTERRUPÇÃO: #
#####

ORG 23H ; INTERRUPÇÃO DO CANAL SERIAL
JB TI,XXX

CLR RI ; LIMPA RI, QUE INDICA QUE A INT OCORREU
MOV A,SBUF ; A = CARÁCTER RECEBIDO
SETB RXSER ; RECEBEU DADO DO P.C.
SJMP YYY

XXX:
CLR TI ; LIMPA TI QUE INDICA QUE A INT OCORREU

YYY:
RETI ; RETORNA DA INTERRUPÇÃO

#####
;# ROTINAS: #
#####

; TEMPO DE 1s
;=====
TEMP1:

T21: MOV R6,#TEMPO3 ; CARREGA R6 COM VALOR DE TEMPO3
T22: MOV R5,#TEMPO2 ; CARREGA R5 COM VALOR DE TEMPO2
T23: MOV R4,#TEMPO1 ; CARREGA R4 COM VALOR DE TEMPO1
T24: DJNZ R4,T24 ; DECREMENTA R4, VOLTA À T24 SE DIF. DE 0
DJNZ R5,T23 ; DECREMENTA R5, VOLTA À T23 SE DIF. DE 0
DJNZ R6,T22 ; DECREMENTA R6, VOLTA À T22 SE DIF. DE 0
RET ; FIM DA ROTINA

;=====

; TEMPO DE 20ms
;=====
TEMP002: ; ROTINA PARA CONTAGEM DE 20ms
MOV R7,#TEMPO5 ; CARREGA R7 COM VALOR DE TEMPO5
T021: MOV R6,#TEMPO1 ; CARREGA R6 COM VALOR DE TEMPO1
T022: DJNZ R6,T022 ; DECREMENTA R6, VOLTA À T022 SE DIF DE 0
DJNZ R7,T021 ; DECREMENTA R7, VOLTA À T021 SE DIF DE 0
RET ; FIM DA ROTINA

;=====

; SERIAL IMPLM. TX GER->POS
;=====
ENVPOS: ; PARA O SISTEMA DE POSICIONAMENTO
CLR TXD ; START BIT, INÍCIO DA TRANSMISSÃO
MOV AUX1,#STEMP ; AGUARDA O TEMPO ENTRE OS BITS

TSER1:
DJNZ AUX1,TSER1

```

```

ENVD1:  MOV     AUX2,#8           ; 8 BITS A SEREM ENVIADOS
        RRC     A               ; BIT A SER ENVIADO NO CARRY
        MOV     TXD,C           ; TXD = CARRY
        MOV     AUX1,#STEMP    ; AGUARDA O TEMPO ENTRE OS BITS

TSER2:  DJNZ    AUX1,TSER2
        DJNZ    AUX2,ENVD1     ; AGUARDA O ENVIO DOS 8 BITS
        SETB   TXD             ; STOP BIT, FIM DA TRANSMISSÃO
        RRC     A               ; RETORNÁ A E C AS CONDIÇÕES INICIAIS
        MOV     AUX1,#STEMP    ; AGUARDA O TEMPO ENTRE OS BITS

TSER3:  DJNZ    AUX1,TSER3
        RET                      ; FIM DA ROTINA
=====
; SERIAL IMPLM. RX POS->GER
=====
RECD:   JB      RXD,RECD        ; AGUARDA START BIT
        MOV     AUX1,#STEMP2   ; AGUARDA METADE DO TEMPO ENTRE BITS

RSER1:  DJNZ    AUX1,RSER1
        JB      RXD,RECD        ; VERIFICA SE START BIT É VÁLIDO
        MOV     AUX2,#8        ; RECEPÇÃO DE 8 BITS

RECD1:  MOV     AUX1,#STEMP    ; AGUARDA O TEMPO ENTRE BITS

RSER2:  DJNZ    AUX1,RSER2
        MOV     C,RXD           ; CARRY = RXD
        RRC     A               ; GUARDA BIT RECEBIDO EM A
        DJNZ    AUX2,RECD1     ; VERIFICA SE OS 8 BITS FORAM RECEBIDOS
        MOV     RECEB,A        ; RECEB = DADO RECEBIDO
        RET                      ; FIM DA ROTINA
=====
; ENVIO DE PACOTE DE DADOS PARA COMPUTADOR
=====
PACCOM: MOV     CONT,#16        ; 1A. PARTE DO PACOTE
        MOV     R0,#PRIMEIRO   ; POSIÇÃO DE MEMÓRIA DO 1o. CARAC. DE NMEA

PACCOM1: MOV     A,@R0          ; VALOR DO CARACTER DE NMEA EM ACC
        MOV     SBUF,A         ; MANDA PARA SISTEMA DE GERENCIAMENTO
        ACALL  TEMP002         ; TEMPO DE 20ms
        INC     R0
        DJNZ   CONT,PACCOM1    ; DESVIA SE FIM DA 1a. PARTE DO PACOTE

        MOV     CONT,#17        ; RESTO DO PACOTE
        MOV     R0,#SEGUNDO    ; 2a. POSIÇÃO DE MEMÓRIA

PACCOM2: MOV     A,@R0          ; VALOR DO CARACTER DE NMEA EM ACC
        MOV     SBUF,A         ; MANDA PARA SISTEMA DE GERENCIAMENTO
        ACALL  TEMP002         ; TEMPO DE 20ms
        INC     R0
        DJNZ   CONT,PACCOM2    ; DESVIA SE FIM DE NMEA
        RET
=====
; RECEBE PACOTE DE DADOS DO SIST. POS.
=====
PACPOS: MOV     R0,#PRIMEIRO   ; TAMANHO DA STRING DE DADOS
        MOV     CONT,#16

PACPOS1: ACALL  RECD           ; CANAL SERIAL IMPLEMENTADO
        MOV     @R0,RECEB
        INC     R0
        DJNZ   CONT,PACPOS1    ; DESVIA SE FIM DA PRIMEIRA PARTE DO PACOTE

        MOV     R0,#SEGUNDO    ; 2a. POSIÇÃO DE MEMÓRIA
        MOV     CONT,#17

PACPOS2: ACALL  RECD           ; CANAL SERIAL IMPLEMENTADO
        MOV     @R0,RECEB
        INC     R0
        DJNZ   CONT,PACPOS2    ; DESVIA SE FIM DE NMEA
        RET
=====
; ANALISA O ACK QUE VEIO DO SIST. POSICIONAMENTO
=====
CHKPOS: ACALL  RECD           ; RECEBE ACK DO SIST. POS.
        MOV     A,RECEB
        CJNE   A,#!,ACKPN1    ; DESVIA SE ACK PACOTE É DE ERRO
        SJMP   ACKPACO        ; OK PACOTE ENVIADO COM SUCESSO

ACKPN1: CLR     F_ACKPOS
        SJMP   FIMCHKPOS

ACKPACO: SETB   F_ACKPOS       ; AGUARDA ACK PARA COMEÇAR A RECEBER PACOTE DO POSICIONAMENTO
        ; ACK RECEBIDO COM SUCESSO!

FIMCHKPOS: CLR     C
        RET
=====

```

; ENVIA PACOTE DE DADOS DE LEITURA DA NAVEGAÇÃO PARA SIST. POS.

=====

```
LEITPOS:
    ACALL    TEMP1

    MOV     A,#'$'
    ACALL   ENVPOS           ; ENVIA CHARACTER PARA POS.
    ACALL   TEMP002

    MOV     R0,#11H
    MOV     A,@R0           ; ATUA/NAVEGA
    ACALL   ENVPOS           ; ESTADO
    ACALL   TEMP002

    INC     R0

    MOV     A,@R0
    ACALL   ENVPOS           ; LEME
    ACALL   TEMP002

    INC     R0

    MOV     A,@R0
    ACALL   ENVPOS           ; MANCHE
    ACALL   TEMP002

    MOV     A,#0
    XRL    A,11H
    XRL    A,12H
    XRL    A,13H
    ACALL   ENVPOS           ; ENVIA CHECKSUM PARA POS.
    ACALL   CHKPOS           ; ANALISA CHECKSUM QUE VEM DO POSICIONAMENTO

    JB     F_ACKPOS,F_LPOS   ; PULA SE ACK COM SUCESSO
    SJMP    LEITPOS         ; VOLTA E ENVIA NOVAMENTE PACOTE DE DADOS PARA POSICIONAMENTO

F_LPOS:
    RET
```

=====

; CHECKSUM DOS DADOS

=====

```
CHECKSUM:
    CLR     FLAG_AD
    CLR     FLAG_BUS
    CLR     FLAG_GPS

    MOV     A,#00H
    MOV     R0,#PRIMEIRO
    INC     R0               ; PULA $

    XRL    A,@R0           ; 'N'
    INC     R0

    XRL    A,@R0           ; LEME
    INC     R0

    XRL    A,@R0           ; MANCHE
    INC     R0

    MOV     AUX,@R0        ; CHECKSUM DO A/D

    CJNE   A,AUX,FAIL_AD   ; PULA SE DADO CORROMPIDO
    SJMP    CHKBUS

FAIL_AD:
    SETB   FLAG_AD

CHKBUS:
    MOV     A,#00H
    INC     R0               ; '$'

    INC     R0               ; 'B'

    XRL    A,@R0           ; 'B'
    INC     R0

    XRL    A,@R0           ; BYTE MAIS SIGNIFICATIVO DA BÚSSOLA
    INC     R0

    XRL    A,@R0           ; BYTE MENOS SIGNIFICATIVO DA BÚSSOLA
    INC     R0

    MOV     AUX,@R0        ; CHECKSUM DA BÚSSOLA

    CJNE   A,AUX,FAIL_BUS  ; PULA SE DADO CORROMPIDO
    SJMP    CHKGPS

FAIL_BUS:
    SETB   FLAG_BUS

CHKGPS:
    MOV     R0,#1DH        ; POSIÇÃO DO ESTADO DA STRING GPS
    MOV     A,@R0

    CJNE   A,#'V',CH1      ; PULA SE NAO FOR 'V'
    SETB   FLAG_GPS        ; DADO INVÁLIDO
    SJMP    F_CHECK        ; VAI PARA FIM DE ROTINA

CH1:
    CJNE   A,#'T',CH2      ; PULA SE NAO FOR 'T'
    SETB   FLAG_GPS        ; DADO INVÁLIDO
```

```

CH2:    SJMP    F_CHECK      ; VAI PARA FIM DE ROTINA
        CJNE   A,#'X',CH3    ; PULA SE NAO FOR 'X'
        SETB  FLAG_GPS      ; DADO INVÁLIDO
        SJMP  F_CHECK      ; VAI PARA FIM DE ROTINA

CH3:    CJNE   A,#'A',CH4    ; DADO PODE SER 'A'
        SJMP  AOKGPS        ; PULA SE NAO FOR 'A'
                                ; DADO VÁLIDO

CH4:    SETB   FLAG_GPS      ; VALOR DESCONHECIDO
        SJMP  F_CHECK      ; VAI PARA FIM DE ROTINA

AOKGPS: MOV    A,#00H        ; ZERA ACUMULADOR
        MOV   R0,#PRIMEIRO+11 ; PULA O '$' DO GPS
        MOV   CONT,#5        ; CARACTERES MENOS O '$' NO BANCO 2/3
CHK1:   XRL   A,@R0          ; XOR ENTRE END. MEM. ANTER. E POST.
        INC   R0
        DJNZ  CONT,CHK1

        MOV   R0,#SEGUNDO    ; CONTINUAÇÃO
        MOV   CONT,#12       ; STRING GPS ATÉ O VALOR DE VELOCIDADE

CHK2:   XRL   A,@R0          ; CALCULA XOR
        INC   R0
        DJNZ  CONT,CHK2

        INC   R0             ; PULA '*'
        INC   R0             ; PULA '*'

        MOV   AUX,@R0
        CJNE  A,AUX,CHK4     ; DESVIA SE ERRO!!!
        SJMP  CHK5           ; SUCESSO NA COMUN. COM GPS OEM
CHK4:   SETB  FLAG_GPS      ; DADOS DO GPS CORROMPIDOS

CHK5:

F_CHECK: CLR   C
        RET

;=====
; EFETUA AS TROCAS DE VARIÁVEIS NECESSÁRIAS PARA SUBTRAÇÃO DA LATITUDE
;=====
LATAUX: MOV   AUXNAV,DESLAT   ; TROCA TEMPORARIAMENTE VALORES ATUAL E DESTINO
        MOV   DESLAT,SEGLAT ; PARA OTIMIZAÇÃO DE CÓDIGO
        MOV   SEGLAT,AUXNAV ; (DESTINO > ATUAL)

        MOV   AUXNAV,DEMLAT
        MOV   DEMLAT,MINLAT
        MOV   MINLAT,AUXNAV

        MOV   AUXNAV,DEGLAT
        MOV   DEGLAT,GRALAT
        MOV   GRALAT,AUXNAV

        MOV   AUXGRA,GRALAT ; VARIÁVEIS AUXILIARES
        MOV   AUXMIN,MINLAT
        MOV   AUXSEG,SEGLAT

        RET

;=====
; EFETUA AS TROCAS DE VARIÁVEIS NECESSÁRIAS PARA SUBTRAÇÃO DA LONGITUDE
;=====
LONAUX: MOV   AUXNAV,DESLON  ; TROCA TEMPORARIAMENTE VALORES ATUAL E DESTINO
        MOV   DESLON,SEGLON ; PARA OTIMIZAÇÃO DE CÓDIGO
        MOV   SEGLON,AUXNAV ; (DESTINO > ATUAL)

        MOV   AUXNAV,DEMLON
        MOV   DEMLON,MINLON
        MOV   MINLON,AUXNAV

        MOV   AUXNAV,DEGLON
        MOV   DEGLON,GRALON
        MOV   GRALON,AUXNAV

        MOV   AUXGRA,GRALON ; VARIÁVEIS AUXILIARES
        MOV   AUXMIN,MINLON
        MOV   AUXSEG,SEGLON

        RET

;=====
; SUBTRAÇÃO DOS GRAUS DA LAT.
;=====
SUBGLAT: MOV   A,AUXGRA      ; GRAU LAT. A SER SUBTRAIDO
        MOV   AUXNAV,DEGLAT
        SUBB  A,AUXNAV
        MOV   DIGLAT,A      ; ARMAZENA DIFERENCA DA LATITUDE EM GRAUS
        RET
;=====

```

; SUBTRAÇÃO DOS MINUTOS DA LAT.

;=====

```

SUBMLAT:
  CLR      C
  MOV      A,AUXMIN      ; MINUTO LAT. A SER SUBTRAIDO
  MOV      AUXNAV,DEMLAT
  SUBBB    A,AUXNAV
  JC       EMPGLAT      ; PULA SE PRECISA EMPRESTAR UM GRAU
  MOV      DIMLAT,A      ; ARMAZENA DIFERENÇA DE MINUTOS DA LATITUDE
  SJMP     FIMSUBM

EMPGLAT:
  CLR      C
  DEC      AUXGRA      ; DECREMENTA UM GRAU
  MOV      A,AUXMIN
  ADD      A,#60      ; ADICIONA 1 GRAU
  SUBBB    A,AUXNAV
  MOV      DIMLAT,A      ; ARMAZENA DIFERENÇA DA LATITUDE EM MINUTOS

FIMSUBM:
  RET

```

; SUBTRAÇÃO DOS SEGUNDOS DA LAT.

;=====

```

SUBSLAT:
  CLR      C
  MOV      A,SEGLAT      ; SEGUNDOS LAT. ATUAL
  MOV      AUXNAV,DESLAT
  SUBBB    A,AUXNAV
  JC       EMPMLAT      ; PULA SE PRECISA EMPRESTAR UM MINUTO
  MOV      DISLAT,A      ; ARMAZENA DIFERENÇA DE SEGUNDOS DA LATITUDE
  SJMP     FIMSUBS

EMPMLAT:
  CLR      C
  DEC      AUXMIN      ; DECREMENTA UM MINUTO
  MOV      A,AUXSEG
  ADD      A,#60      ; ADICIONA 1 MINUTO
  SUBBB    A,AUXNAV
  MOV      DISLAT,A      ; ARMAZENA DIFERENÇA DE SEGUNDOS DA LATITUDE

FIMSUBS:
  RET

```

; SUBTRAÇÃO DOS GRAUS DA LON.

;=====

```

SUBGLON:
  MOV      A,AUXGRA      ; GRAU LAT. A SER SUBTRAIDO
  MOV      AUXNAV,DEGLON
  SUBBB    A,AUXNAV
  MOV      DIGLON,A      ; ARMAZENA DIFERENÇA EM GRAUS
  RET

```

; SUBTRAÇÃO DOS MINUTOS DA LON.

;=====

```

SUBMLON:
  CLR      C
  MOV      A,AUXMIN      ; MINUTO A SER SUBTRAIDO
  MOV      AUXNAV,DEMLON
  SUBBB    A,AUXNAV
  JC       EMPGLON      ; PULA SE PRECISA EMPRESTAR UM GRAU
  MOV      DIMLON,A      ; ARMAZENA DIFERENÇA DE MINUTOS
  SJMP     FIMSUBMLON

EMPGLON:
  CLR      C
  DEC      AUXGRA      ; DECREMENTA UM GRAU
  MOV      A,AUXMIN
  ADD      A,#60      ; ADICIONA 1 GRAU
  SUBBB    A,AUXNAV
  MOV      DIMLON,A      ; ARMAZENA DIFERENÇA EM MINUTOS

FIMSUBMLON:
  RET

```

; SUBTRAÇÃO DOS SEGUNDOS DA LON.

;=====

```

SUBSLON:
  CLR      C
  MOV      A,SEGLON      ; SEGUNDOS ATUAL
  MOV      AUXNAV,DESLON
  SUBBB    A,AUXNAV
  JC       EMPMLON      ; PULA SE PRECISA EMPRESTAR UM MINUTO
  MOV      DISLON,A      ; ARMAZENA DIFERENÇA DE SEGUNDOS
  SJMP     FIMSUBSLON

EMPMLON:
  CLR      C
  DEC      AUXMIN      ; DECREMENTA UM MINUTO
  MOV      A,AUXSEG
  ADD      A,#60      ; ADICIONA 1 MINUTO
  SUBBB    A,AUXNAV
  MOV      DISLON,A      ; ARMAZENA DIFERENÇA DE SEGUNDOS DA LATITUDE

FIMSUBSLON:
  RET

```

; CALCULA DIFERENÇA DE LATITUDE ENTRE PONTO ATUAL E DESTINO

;=====

DIFLATANDES: ; *** DIFERENÇA CALCULADA EM MÓDULO ***

```

  CLR      C
  CLR      ZEROLAT      ; HÁ DISTÂNCIAS ENTRE OS PONTOS (HIPÓTESE)

```

```

MOV      AUXGRA,GRALAT      ; COLOCA VALORES DA LATITUDE ATUAL EM VARIÁVEIS AUXILIARES
MOV      AUXMIN,MINLAT
MOV      AUXSEG,SEGLAT

; *** DEFINE ORDEM DE SUBTRAÇÃO ***

MOV      A,DEGLAT          ; GRAUS DESTINO
CJNE     A,GRALAT,LAT1     ; DESVIA SE GRAUS DIFERENTES
MOV      DIGLAT,#0         ; ZERO GRAUS DE DIFERENÇA

MOV      A,DEMLAT          ; MINUTOS DESTINO
CJNE     A,MINLAT,LAT2     ; DESVIA SE MINUTOS DIFERENTES
MOV      DIMLAT,#0        ; ZERO MINUTOS DE DIFERENÇA

MOV      A,DESLAT          ; SEGUNDOS DESTINO
CJNE     A,SEGLAT,LAT3     ; DESVIA SE SEGUNDOS DIFERENTES
MOV      DISLAT,#0        ; ZERO SEGUNDOS DE DIFERENÇA

LAT1:    SJMP              FZLAT      ; *** OBJETIVO ALCANÇADO ATUAL = DESTINO ***
; GRAUS, MINUTOS E SEGUNDOS DIFERENTES DE ZERO

MOV      A,DEGLAT          ; GRAUS DE DESTINO
MOV      AUXNAV,GRALAT     ; AUX

SUBB     A,AUXNAV          ; PULA SE ATUAL > DESTINO
JC       ATLAT1            ; DESTINO > ATUAL
ACALL    LATAUX
SJMP     ATLAT1

LAT2:    ; MINUTOS E SEGUNDOS DIFERENTES DE ZERO

MOV      A,DEMLAT          ; MINUTOS DE DESTINO
MOV      AUXNAV,MINLAT     ; AUX
SUBB     A,AUXNAV
JC       ATLAT2            ; PULA SE ATUAL > DESTINO
ACALL    LATAUX
SJMP     ATLAT2

LAT3:    ; APENAS SEGUNDOS DIFERENTES DE ZERO
; SEGUNDOS DE DESTINO
; AUX
MOV      A,DESLAT
MOV      AUXNAV,SEGLAT
SUBB     A,AUXNAV
JC       ATLAT3            ; PULA SE ATUAL > DESTINO
ACALL    LATAUX
SJMP     ATLAT3

ATLAT1:  ACALL    SUBSLAT
ACALL    SUBMLAT
ACALL    SUBGLAT
SJMP     FIMDIFLAT

ATLAT2:  ACALL    SUBSLAT
ACALL    SUBMLAT
SJMP     FIMDIFLAT

ATLAT3:  ACALL    SUBSLAT
SJMP     FIMDIFLAT

FZLAT:   SETB     ZEROLAT      ; DIFERENÇA ZERO ENTRE PONTO ATUAL E DESTINO

FIMDIFLAT:
CLR      C
RET

```

```

; CÁLCULO DA DIFERENÇA DE LONGITUDE ENTRE O PONTO ATUAL E O PONTO DESTINO

```

```

=====
DIFLONANDES:      ; *** DIFERENÇA CALCULADA EM MÓDULO ***

CLR      C
CLR      ZEROLON          ; HÁ DISTÂNCIAS ENTRE OS PONTOS (HIPÓTESE)

MOV      AUXGRA,GRALON    ; COLOCA VALORES DA LONGITUDE ATUAL EM VARIÁVEIS AUXILIARES
MOV      AUXMIN,MINLON
MOV      AUXSEG,SEGLON

; *** DEFINE ORDEM DE SUBTRAÇÃO

MOV      A,DEGLON         ; GRAUS DESTINO
CJNE     A,GRALON,LON1    ; DESVIA SE GRAUS DIFERENTES
MOV      DIGLON,#0        ; ZERO GRAUS DE DIFERENÇA

MOV      A,DEMLON         ; MINUTOS DESTINO
CJNE     A,MINLON,LON2    ; DESVIA SE MINUTOS DIFERENTES
MOV      DIMLON,#0        ; ZERO MINUTOS DE DIFERENÇA

MOV      A,DESLON         ; SEGUNDOS DESTINO
CJNE     A,SEGLON,LON3    ; DESVIA SE SEGUNDOS DIFERENTES
MOV      DISLON,#0        ; ZERO SEGUNDOS DE DIFERENÇA

SJMP     FZLON            ; OBJETIVO ALCANÇADO ATUAL = DESTINO

LON1:    ; GRAUS, MINUTOS E SEGUNDOS DIFERENTES DE ZERO
; GRAUS DESTINO
; AUX
MOV      A,DEGLON
MOV      AUXNAV,GRALON
SUBB     A,AUXNAV
JC       ATLON1            ; PULA SE ATUAL > DESTINO
ACALL    LONAU
SJMP     ATLON1

LON2:    ; MINUTOS E SEGUNDOS DIFERENTES DE ZERO
; MINUTOS DESTINO
; AUX
MOV      A,DEMLON
MOV      AUXNAV,MINLON

```



```

SUBB      A,AUXNAV
JC        ATLON2
ACALL    LONAUX
S JMP     ATLON2
LON3:
MOV      A,DESLO
MOV      AUXNAV,SEGLON
SUBB     A,AUXNAV
JC        ATLON3
ACALL    LONAUX
S JMP     ATLON3
ATLON1:
ACALL    SUBSLON
ACALL    SUBMLON
ACALL    SUBGLON
S JMP     FIMDIFLON
ATLON2:
ACALL    SUBSLON
ACALL    SUBMLON
S JMP     FIMDIFLON
ATLON3:
ACALL    SUBSLON
S JMP     FIMDIFLON
FZLON:
SETB     ZEROLON
FIMDIFLON:
CLR      C
RET

```

```

=====
; ARMAZENA FLAG DE INCREMENTO/DECREMENTO DE LATITUDE
=====

```

```

INCDECLAT:

```

```

CLR      ZERO
MOV      A,GRALAT
CJNE     A,ANGLAT,CGLAT
MOV      A,MINLAT
CJNE     A,ANMLAT,CMLAT
S JMP     CSLAT
CGLAT:
CLR      C
MOV      A,ANGLAT
MOV      AUXNAV,GRALAT
SUBB     A,AUXNAV
JC        LFINCLAT
CLR      FIDLAT
S JMP     FIM_LAT
CMLAT:
CLR      C
MOV      A,ANMLAT
MOV      AUXNAV,MINLAT
SUBB     A,AUXNAV
JC        LFINCLAT
CLR      FIDLAT
S JMP     FIM_LAT
CSLAT:
CLR      C
MOV      A,ANSLAT
MOV      AUXNAV,SEGLAT
SUBB     A,AUXNAV
JZ        NOMUD1
JC        LFINCLAT
CLR      FIDLAT
S JMP     FIM_LAT
NOMUD1:
SETB     ZERO
LFINCLAT:
SETB     FIDLAT
FIM_LAT:
CLR      C
RET

```

```

=====
; ARMAZENA FLAG DE INCREMENTO/DECREMENTO DE LONGITUDE
=====

```

```

INCDECLON:

```

```

CLR      ZERO
MOV      A,GRALON
CJNE     A,ANGLON,CGLON
MOV      A,MINLON
CJNE     A,ANMLON,CMLON
S JMP     CSLON
CGLON:
CLR      C
MOV      A,ANGLON
MOV      AUXNAV,GRALON
SUBB     A,AUXNAV
JC        LFINCLON
CLR      FIDLON
S JMP     FIM_LON
CMLON:
CLR      C
MOV      A,ANMLON
MOV      AUXNAV,MINLON
SUBB     A,AUXNAV
JC        LFINCLON
CLR      FIDLON

```

```

CSLON:  Sjmp    FIM_LON      ; FIM DE ANALISE DA LON.
        CLR     C              ; COMPARA SEGUNDO DA LATITUDE
        MOV     A,ANSLON
        MOV     AUXNAV,SEGLON
        SUBB    A,AUXNAV
        JZ      NOMUD2
        JC      LFINCLON      ; PULA PQ INCREMENTOU SEGUNDO DA LON.
        CLR     FIDLON        ; ARMAZENA DECREMENTO DO SEGUNDO DA LON.
        Sjmp    FIM_LON      ; FIM DE ANALISE DA LON.

NOMUD2: SETB    ZERO          ; INDICA QUE NAO HOUVE MUDANCA DE LONGITUDE

LFINCLON: SETB   FIDLON       ; INCREMENTO DA LATITUDE

FIM_LON:  CLR     C
        RET

```

```

=====
; CÁLCULO DO SALDO DA LATITUDE (POSITIVO OU NEGATIVO -> BIT FSALAT)
=====

```

```

CASALAT: CLR     C
        MOV     A,DAGLAT      ; DIF. GRAUS ANTERIOR
        MOV     AUXNAV,DIPLAT ; DIF. GRAUS ATUAL
        SUBB    A,AUXNAV
        JZ      NODGLA       ; DESVIA SE NÃO EXISTIR DIFERENÇA

        MOV     A,DAGLAT      ; DIF. GRAUS ANTERIOR
        MOV     AUXNAV,DIPLAT ; DIF. GRAUS ATUAL
        SUBB    A,AUXNAV
        JC      LFSALLAT     ; PULA SE SALDO NEGATIVO
        SETB    FSALAT       ; SETA BIT COM SALDO POSITIVO
        Sjmp    FIMSALA      ; PULA PARA FIM DE CÁLCULO DE SALDO

NODGLA:  MOV     A,DAMLAT     ; DIF. MINUTOS ANTERIOR
        MOV     AUXNAV,DIMLAT ; DIF. MINUTOS ATUAL
        SUBB    A,AUXNAV
        JZ      NODMLA       ; DESVIA SE NÃO EXISTIR DIFERENÇA
        MOV     A,DAMLAT     ; DIF. MINUTOS ANTERIOR
        MOV     AUXNAV,DIMLAT ; DIF. MINUTOS ATUAL
        SUBB    A,AUXNAV
        JC      LFSALLAT     ; SETA BIT COM SALDO POSITIVO
        SETB    FSALAT
        Sjmp    FIMSALA

NODMLA:  MOV     A,DASLAT     ; DIF. SEGUNDOS ANTERIOR
        MOV     AUXNAV,DISLAT ; DIF. SEGUNDOS ATUAL
        SUBB    A,AUXNAV
        JC      LFSALLAT     ; SETA BIT COM SALDO POSITIVO
        SETB    FSALAT
        Sjmp    FIMSALA

LFSALLAT: CLR    C
        CLR    FSALAT        ; LIMPA BIT COM SALDO NEGATIVO

FIMSALA:  RET

```

```

=====
; CÁLCULO DO SALDO DA LONGITUDE (POSITIVO OU NEGATIVO -> BIT FSALON)
=====

```

```

CASALON: CLR     C
        MOV     A,DAGLON     ; DIF. GRAUS ANTERIOR
        MOV     AUXNAV,DIGLON ; DIF. GRAUS ATUAL
        SUBB    A,AUXNAV
        JZ      NODGLO       ; DESVIA SE NÃO EXISTIR DIFERENÇA

        MOV     A,DAGLON     ; DIF. GRAUS ANTERIOR
        MOV     AUXNAV,DIGLON ; DIF. GRAUS ATUAL
        SUBB    A,AUXNAV
        JC      LFSALLON     ; PULA SE SALDO NEGATIVO
        SETB    FSALON       ; SETA BIT COM SALDO POSITIVO
        Sjmp    FIMSALO      ; PULA PARA FIM DE CÁLCULO DE SALDO

NODGLO:  MOV     A,DAMLON     ; DIF. MINUTOS ANTERIOR
        MOV     AUXNAV,DIMLON ; DIF. MINUTOS ATUAL
        SUBB    A,AUXNAV
        JZ      NODMLO       ; DESVIA SE NÃO EXISTIR DIFERENÇA

        MOV     A,DAMLON     ; DIF. MINUTOS ANTERIOR
        MOV     AUXNAV,DIMLON ; DIF. MINUTOS ATUAL
        SUBB    A,AUXNAV
        JC      LFSALLON     ; SETA BIT COM SALDO POSITIVO
        SETB    FSALON
        Sjmp    FIMSALO

NODMLO:  MOV     A,DASLON     ; DIF. SEGUNDOS ANTERIOR
        MOV     AUXNAV,DISLON ; DIF. SEGUNDOS ATUAL
        SUBB    A,AUXNAV
        JC      LFSALLON     ; SETA BIT COM SALDO POSITIVO
        SETB    FSALON
        Sjmp    FIMSALO

LFSALLON: CLR    C
        CLR    FSALON       ; LIMPA BIT COM SALDO NEGATIVO

FIMSALO:  RET

```

; CALCULA DIFERENÇA ENTRE LATITUDE E LONGITUDE DO PONTO ATUAL

```

=====
DIFLATLONAT:                                     ; *** DIFERENÇA CALCULADA EM MÓDULO ***

      CLR      C
      MOV      A,DIGLAT                          ; GRAUS LATITUDE
      SUBB     A,DIGLON
      JC       NOLATLON1                        ; PULA SE NAO DER CARRY, SUBTRAÇÃO COM SUCESSO
      MOV      A,DLALOG                          ; ARMAZENA DIF. EM GRAUS DE LAT/LONG
      SJMP     MLATLON                          ; VAI PRA CALCULAR DIF. EM MINUTOS DE LAT/LON

OVLATLON1:
      MOV      A,DIGLON                          ; GRAUS LONGITUDE
      SUBB     A,DIGLAT
      MOV      A,DLALOG                          ; ARMAZENA DIF. EM GRAUS DE LAT/LONG

MLATLON:
      CLR      C
      MOV      A,DIMLAT                          ; MINUTOS LATITUDE
      SUBB     A,DIMLON
      JC       NOLATLON2                        ; PULA SE NAO DER CARRY, SUBTRAÇÃO COM SUCESSO
      MOV      A,DLALOM                          ; ARMAZENA DIF. EM MINUTOS DE LAT/LONG
      SJMP     SLATLON                          ; VAI PRA CALCULAR DIF. EM SEGUNDOS DE LAT/LON

NOLATLON2:
      CLR      C
      MOV      A,DIMLON                          ; GRAUS LONGITUDE
      SUBB     A,DIMLAT
      MOV      A,DLALOM                          ; ARMAZENA DIF. EM MINUTOS DE LAT/LONG

SLATLON:
      MOV      A,DISLAT                          ; SEGUNDOS LATITUDE
      SUBB     A,DISLON
      JC       NOLATLON3                        ; PULA SE NAO DER CARRY, SUBTRAÇÃO COM SUCESSO
      MOV      A,DLALOS                          ; ARMAZENA DIF. EM SEGUNDOS DE LAT/LONG
      SJMP     FLATLONX                         ; VAI PRA FIM DE CÁLCULO

NOLATLON3:
      MOV      A,DISLON                          ; SEGUNDOS LONGITUDE
      SUBB     A,DISLAT
      MOV      A,DLALOS                          ; ARMAZENA DIF. EM SEGUNDOS DE LAT/LONG

FLATLONX:
      CLR      C
      RET
=====

```

; EFETUA O GIRO NA BÚSSOLA

```

=====
GIRABUSS:

IGIRA:
      MOV      R0,#17H
      MOV      GBUS2,@R0                        ; PEGA VALOR DA BÚSSOLA NA MEMÓRIA
      INC      R0
      MOV      GBUS1,@R0

      MOV      A,GBUS2

      CJNE     A,#0,MAI255                      ; PULA SE BÚSSOLA TEM GRAUS > 255 (2 BYTES)
      CLR      C
      MOV      A,GBUS1                          ; < 255
      ADD      A,#15
      MOV      GBUS1,A
      JNC      FIMGIHO                          ; SOMA FEITA CORRETAMENTE SEM ULTRAPASSAR 255 GRAUS
      MOV      GBUS1,A                          ; SOMA > 255
      MOV      GBUS2,#1
      AJMP     FIMGIHO

MAI255:
      CLR      C
      MOV      A,GBUS1
      ADD      A,#15                            ; ADICIONA 30 GRAUS DE GIRO
      MOV      GBUS1,A
      SUBB     A,#105                            ; VALOR MÁX ADMITIDO (359 GRAUS)
      JNC      MAI359                            ; > 359 VOLTA A CONTAR A PARTIR DO ZERO
      SJMP     FIMGIHO

MAI359:
      MOV      GBUS2,#0                          ; BYTE ALTO DA BÚSSOLA FICA COM ZERO (< 255)
      MOV      A,GBUS1
      SUBB     A,#104                            ; RETIRA TUDO Q PASSOU DO LIMITE
      MOV      GBUS1,A                          ; ARMAZENA VALOR CORRETO

FIMGIHO:
      CLR      C
      MOV      R0,#17H
      MOV      @R0,GBUS2                        ; ARMAZENA VALOR CORRIGIDO DA BÚSSOLA NA MEMÓRIA
      INC      R0
      MOV      @R0,GBUS1

      DJNZ     CONT,IGIRA                       ; CALCULA QUANTAS VEZES GIRA A BÚSSOLA

      RET
=====

```

; SUBTRAI VALOR REAL DA BÚSSOLA DO VALOR A SER ALCANÇADO COM O GIRO

```

=====
SBPG:
      CLR      C
      MOV      A,GBUS2

```

```

SUBB    A,17H
JC      BB3          ; ENTAO A DIFERENÇA ENTRE ELES É > 255 (0-1 NO MSB)
JNZ     BB3          ; ENTAO A DIFERENÇA ENTRE ELES É > 255 (1-0 NO MSB)

CLR     C
MOV     A,GBUS1
SUBB   A,18H          ; ENDEREÇO DE GBUS1 DA BÚSSOLA DE POS.
JC      BB1          ; TROCA POSIÇÃO DE SUBTRAÇÃO
SJMP   BB2

BB1:    MOV     A,18H          ; ENDEREÇO DE GBUS1 DA BÚSSOLA DE POS.
SUBB   A,GBUS1

BB2:    CLR     C
SUBB   A,#6
JNC     BB3          ; DIFERENÇA ENTRE ELES > 5 NÃO ACEITÁVEL
CLR     F_SBUSPG    ; SUCESSO NA SUBTRAÇÃO OBJETIVO ALCANÇADO!!!
SJMP   FIMSBPB

BB3:    SETB   F_SBUSPG    ; OBJETIVO NÃO ALCANÇADO

FIMSBPB:
RET

;=====
; TRANSFORMA PONTO ATUAL DO G.P.S. EM PONTO ANTERIOR
;=====
PTATAN:
MOV     A,GRALAT
MOV     ANGLAT,A
MOV     A,GRALON
MOV     ANGLON,A
MOV     A,MINLAT
MOV     ANMLAT,A
MOV     A,MINLON
MOV     ANMLON,A
MOV     A,SEGLAT
MOV     ANSLAT,A
MOV     A,SEGLON
MOV     ANSLON,A
RET

;=====
; BUSCA NA MEMÓRIA LATITUDE, LONGITUDE E BÚSSOLA
;=====
MELALOBU:
MOV     GRALAT,1FH    ; BUSCA NA MEMORIA GRAUS DA LATITUDE
MOV     MINLAT,30H   ; BUSCA NA MEMORIA MINUTOS DA LATITUDE
MOV     SEGLAT,31H   ; BUSCA NA MEMORIA SEGUNDOS DA LATITUDE
MOV     GRALON,35H   ; BUSCA NA MEMORIA GRAUS DA LONGITUDE
MOV     MINLON,36H   ; BUSCA NA MEMORIA MINUTOS DA LONGITUDE
MOV     SEGLON,37H   ; BUSCA NA MEMORIA SEGUNDOS DA LONGITUDE

MOV     GBUS1,18H    ; BUSCA NA MEMÓRIA LSB DA BÚSSOLA
MOV     GBUS2,17H    ; BUSCA NA MEMÓRIA MSB DA BÚSSOLA
RET

;=====
; PEDE LEITURA PARA POS. E ANALISA ACK E CHECKSUM DO PACOTE
;=====
TELEM:
MOV     11H,#'L'
ACALL  LEITPOS        ; REQUISIÇÃO DE LEITURA DE NAVEGAÇÃO PARA POS.

ACALL  PACPOS        ; RECEBE PACOTE DE POSICIONAMENTO
ACALL  PACCOM        ; ENVIA PACOTE DE DADOS PARA O COMPUTADOR MESMO COM ERROS (ANÁLISE POSTERIOR)
ACALL  CHECKSUM      ; VERIFICA OS CHECKSUM DO A/D, BÚSSOLA E G.P.S

MOV     A,1DH        ; POSIÇÃO NA MEMÓRIA DO ESTADO ATUAL DO GPS
CJNE   A,#'A',TELEM ; GPS COM DADOS DE POSICIONAMENTO AINDA INCERTOS

JB     FLAG_GPS,TELEM ; STRING GPS COM ERRO
JB     FLAG_AD,TELEM  ; STRING NAVEGAÇÃO COM ERRO
JB     FLAG_BUS,TELEM ; STRING BUSSOLA COM ERRO
CLR   C
RET

;=====
; EFETUA A MUDANÇA DE ROTA DA EMBARCAÇÃO ATÉ ENCONTRAR O VALOR DA BÚSSOLA
;=====
GERATUA:
DENV:
MOV     CONT,#2      ; CONTADOR

ACALL  LEITPOS        ; REQUISIÇÃO DE LEITURA DE NAVEGAÇÃO PARA POS.
ACALL  PACPOS        ; RECEBE PACOTE DE POSICIONAMENTO
ACALL  PACCOM        ; ENVIA PACOTE DE DADOS PARA O COMPUTADOR

ACALL  CHECKSUM      ; VERIFICA OS CHECKSUM DO A/D, BÚSSOLA E G.P.S

JB     FLAG_AD,GERATUA ; STRING NAVEGAÇÃO COM ERRO
JB     FLAG_BUS,GERATUA ; STRING BUSSOLA COM ERRO

MOV     11H,#'A'
MOV     12H,#'3'    ; LEME AO CENTRO
MOV     13H,#'2'    ; MANCHE COM 1A ACELERAÇÃO

```

```

      DJNZ      CONT,DNV
      RET
=====
; EFETUA TODOS OS AJUSTES INICIAIS NECESSÁRIOS PARA NAVEGAÇÃO
=====
PRENAV:
      ; *** ARMAZENA VALORES DO PT. ATUAL ***
      ACALL    MELALOBU      ; BUSCA NA MEMÓRIA LATITUDE, LONGITUDE E BÚSSOLA
      ACALL    RPTDEST      ; RECEBE LATITUDE E LONGITUDE DO P.C. DO PONTO DESTINO A SER ALCANÇADO

      ACALL    DIFLATANDES   ; DIF. LATITUDE ENTRE PT. ATUAL E DESTINO
      ACALL    DIFLONANDES   ; DIF. LONGITUDE ENTRE PT. ATUAL E DESTINO
      ; *** ARMAZENA DIFERENÇAS E VALORES DOS PONTOS EM ANTERIOR ***

      MOV      ANGLAT,GRALAT ; PONTO ANTERIOR
      MOV      ANMLAT,MINLAT
      MOV      ANSLAT,SEGLAT
      MOV      ANGLON,GRALON
      MOV      ANMLON,MINLON
      MOV      ANSLON,SEGLON

      MOV      DAGLAT,DIGLAT ; DIFERENÇA ANTERIOR
      MOV      DAMLAT,DIMLAT
      MOV      DASLAT,DISLAT
      MOV      DAGLON,DIGLON
      MOV      DAMLON,DIMLON
      MOV      DASLON,DISLON

      RET
=====
; ENCONTRA O RUMO DA NAVEGAÇÃO
=====
NAVRUMO:
      ACALL    MELALOBU      ; BUSCA NA MEMÓRIA LATITUDE, LONGITUDE E BÚSSOLA
      CLR      RUMCER      ; BIT DE RUMO CORRETO DA EMBARCAÇÃO / 0 = SEM RUMO CORRETO

TABRUMO:
      ACALL    DIFLATANDES   ; DIF. LAT. ENTRE PONTO ATUAL E DESTINO
      ACALL    DIFLONANDES   ; DIF. LON. ENTRE PONTO ATUAL E DESTINO

      ACALL    INCDECLAT     ; CALCULA INCREMENTO/DECREMENTO DE POSIÇÃO DA LATITUDE (FLAG)
      ACALL    INCDECLON     ; CALCULA INCREMENTO/DECREMENTO DE POSIÇÃO DA LONGITUDE (FLAG)

      ACALL    CASALAT       ; CALCULA VALOR DE SALDO DE LAT (FLAG)
      ACALL    CASALON       ; CALCULA VALOR DE SALDO DE LONGITUDE (FLAG)

      JB      ZEROLAT,OKLAT  ; DIFERENÇA ZERO ENTRE PONTO ATUAL E DESTINO NA LATITUDE
      SJMP    CONTNAV       ; SENAO CONTINUA NAVEGAÇÃO

      JB      ZEROLON,OKLON  ; ZERO TAMBEM PARA LONGITUDE = FIM DE PESQUISA PONTO DESTINO ALCANÇADO!
      SJMP    CONTNAV

      OKLON:
      LJMPL   OKLALO

      CONTNAV:
      JB      ZERO,FANAX     ; LATITUDE OU LONGITUDE NÃO TIVERAM MUDANÇA DE VALOR LOGO MANTER
      AJMPL   MUDRT         ; A POSIÇÃO DO BARCO PARA EVITAR ALGUM ERRO DE CÁLCULO

      FANAX:
      LJMPL   FANALISE      ; LABEL POIS "FANALISE" ESTÁ MUITO LONGE

      MUDRT:
      ; *** ANÁLISE DE MUDANÇA DE ROTA PELA BÚSSOLA SE NECESSÁRIO ***
      ; *** USO DA TABELA DO SENTIDO DE ROTAÇÃO PARA ANÁLISE ***

      JB      FIDLAT,INCLAT  ; PULA SE INCREMENTOU LATITUDE
      JNB     FIDLON,ISALA   ; PULA SE DECREMENTOU LONGITUDE
      AJMPL   ISALB

      INCLAT:
      JNB     FIDLON,ISBAUX  ; PULA SE DECREMENTOU LATITUDE
      SJMPL   ISALA         ; CONTINUA

      ISBAUX:
      LJMPL   ISALB         ; PULA PARA POSIÇÃO BEM ABAIXO

      ISALA:
      ; *** LAT/LON INCREMENTADO -- OU -- LAT/LON DECREMENTADO ***

      JB      FSALAT,SAL1    ; PULA SE SALDO DE LATITUDE POSITIVO
      JNB     FSALON,AUXSAL3 ; PULA SE SALDO DE LONGITUDE NEGATIVO
      SJMPL   SAL0          ; CONTINUA NORMALMENTE

      AUXSAL3:
      LJMPL   SAL3         ; PULA PARA CÓDIGO BEM ABAIXO

      SAL0:
      ; *** GIRAR SENTIDO HORÁRIO ***

      MOV      CONT,#2      ; 2x PARA DIREITA (30 GRAUS)
      ACALL    GIRABUSS

```

```

ATDN1:
MOV      11H,#A'
MOV      12H,#4'      ; PRIMEIRA DIREITA PARA LEME
MOV      13H,#2'      ; PRIMEIRA ACELERAÇÃO PARA MANCHE

ACALL   GERATUA      ; GIRA A EMBARCAÇÃO ATÉ ENCONTRAR VALOR DA BÚSSOLA PEDIDO
ACALL   SBPG         ; ANALISA SE A BÚSSOLA ALCANÇOU A DIREÇÃO DESEJADA (+ 30 GRAUS)
JB      F_SBUSPG,ATDN1 ; DIFERENÇA ENTRE OS VALORES > 5 GRAUS... ATUADENOVO1
AJMP    FANALISE

;*** MANTÉM O CURSO ***

SAL1:
JNB     FSALON,SAL2   ; PULA SE SALDO DE LONGITUDE NEGATIVO

SETB    RUMCER       ; RUMO CERTO PASSAR A USAR AGORA TABELA DE APROXIMAÇÃO
ACALL   DIFLATLONAT  ; DIFERENÇA CALCULADA EM MÓDULO DE LAT-LON
LJMP    FANALISE

;*** GIRAR SENTIDO ANTI HORÁRIO ***

SAL2:
MOV     CONT,#22      ; "GIRA" ATÉ CHEGAR AO VALOR (22 * 15 = 330)
ACALL   GIRABUSS

ATDN2:
MOV     11H,#A'
MOV     12H,#2'      ; PRIMEIRA ESQUERDA PARA LEME
MOV     13H,#2'      ; PRIMEIRA ACELERAÇÃO PARA MANCHE

ACALL   GERATUA      ; GIRA A EMBARCAÇÃO ATÉ ENCONTRAR VALOR DA BÚSSOLA PEDIDO
ACALL   SBPG         ; ANALISA SE A BÚSSOLA ALCANÇOU A DIREÇÃO DESEJADA (- 30 GRAUS)
JB      F_SBUSPG,ATDN2 ; DIFERENÇA ENTRE OS VALORES > 5 GRAUS... ATUADENOVO2
AJMP    FANALISE

;*** GIRAR 180 GRAUS SENTIDO HORÁRIO ***

SAL3:
MOV     CONT,#12      ; GIRA À DIREITA ATÉ CHEGAR AO VALOR (12 * 15GRAUS = 180)
ACALL   GIRABUSS

ATDN3:
MOV     11H,#A'
MOV     12H,#5'      ; SEGUNDA DIREITA PARA LEME
MOV     13H,#2'      ; PRIMEIRA ACELERAÇÃO PARA MANCHE

ACALL   GERATUA      ; GIRA A EMBARCAÇÃO ATÉ ENCONTRAR VALOR DA BÚSSOLA PEDIDO
ACALL   SBPG         ; ANALISA SE A BÚSSOLA ALCANÇOU A DIREÇÃO DESEJADA (+ 180 GRAUS)
JB      F_SBUSPG,ATDN3 ; DIFERENÇA ENTRE OS VALORES > 5 GRAUS... ATUADENOVO3
AJMP    FANALISE

;***** LAT. DECR / LON. INCR -- OU -- LAT.INCR / LON DECR *****

ISALB:
JB      FSALAT,SAL4   ; PULA SE SALDO DE LATITUDE POSITIVO
JNB     FSALON,SAL6   ; PULA SE SALDO DE LONGITUDE NEGATIVO

LJMP    SAL2          ;*** GIRAR SENTIDO ANTI HORÁRIO ***

SAL4:
JNB     FSALON,SAL5   ; PULA SE SALDO DE LONGITUDE NEGATIVO

;*** MANTÉM O CURSO ***
SETB    RUMCER       ; RUMO CERTO PASSAR A USAR AGORA TABELA DE APROXIMAÇÃO
ACALL   DIFLATLONAT  ; DIFERENÇA CALCULADA EM MÓDULO DE LAT-LON
LJMP    FANALISE

;*** GIRAR SENTIDO HORÁRIO ***

SAL5:
LJMP    SAL0

;*** GIRAR 180 GRAUS SENTIDO HORÁRIO ***

SAL6:
LJMP    SAL3

FANALISE:
;*** FIM DA ANÁLISE DE NECESSIDADE DE ALTERAÇÃO DE ROTA ***

ACALL   PTATAN      ; *** PONTO ATUAL VIRA O PONTO ANTERIOR ***
ACALL   DIFLATLONAT ; DIFERENÇA CALCULADA EM MÓDULO DE LAT-LON

OKLALO:
RET

;=====
; FAZ A LAT. E LON. DIMINUIREM PROPORC. EM DIREÇÃO AO PONTO DESTINO
;=====
NAVGRAD:
ACALL   MELALOBU     ; BUSCA NA MEMÓRIA LATITUDE, LONGITUDE E BÚSSOLA

ACALL   DIFLATANDES  ; DIF. LAT. ENTRE PONTO ATUAL E DESTINO
ACALL   DIFLONANDES  ; DIF. LON. ENTRE PONTO ATUAL E DESTINO
ACALL   INCDECLAT    ; CALCULA FLAGS DE INCREMENTO/DECREMENTO DE POSIÇÃO DA LATITUDE
ACALL   INCDECLON    ; CALCULA FLAGS DE INCREMENTO/DECREMENTO DE POSIÇÃO DA LONGITUDE

JB      ZEROLAT,OKLAT1 ; DIFERENÇA ZERO ENTRE PONTO ATUAL E DESTINO
SJMP    CONTNAVX      ; SENAO CONTINUA NAVEGAÇÃO

OKLAT1:
JB      ZEROLON,OKLON1 ; DIFERENÇA ZERO ENTRE PONTO ATUAL E DESTINO = SUCESSO
SJMP    CONTNAVX

OKLON1:
LJMP    OKLALO1      ; FIM DA ROTINA

```

CONTRAVX:

	MOV MOV MOV	AUXGRA,DLALOG AUXMIN,DLALOM AUXSEG,DLALOS	; AUXS PRESERVAM VALORES ANTERIORES DAS DIFERENÇAS
	ACALL	DIFLATLONAT	; DIFERENÇA CALCULADA EM MÓDULO DE LAT-LON
	CLR MOV SUBB JZ JC SJMP	C A,AUXGRA A,DLALOG ZERM TABGRAD ZZZ	; DIFERENÇA = 0 ANALISAR MINUTOS ; DIFERENÇA (LAT - LON) AUMENTOU ENTAO USAR TABELA DE GRADIENTE ; ANALISAR DIFERENÇA ENTRE MINUTOS E SEGUNDOS
ZERM:	CLR MOV SUBB JZ SJMP	C A,AUXMIN A,DLALOM ZERS ZZZ	; DIFERENÇA = 0 ANALISAR SEGUNDOS
ZERS:	CLR MOV SUBB JZ	C A,AUXSEG A,DLALOS TABGRAD	; DIFERENÇA = 0 É RUIM ENTAO USAR TABELA DE GRADIENTE PARA DIMINUIR A DIFERENÇA (LAT-LON)
ZZZ:	CLR MOV SUBB JC	C A,AUXMIN A,DLALOM TABGRAD	; ESTOUROU DIFERENÇA AUMENTOU ENTAO USAR TABELA DE GRADIENTE
	CLR MOV SUBB JC	C A,AUXSEG A,DLALOS TABGRAD	; ESTOUROU DIFERENÇA AUMENTOU ENTAO USAR TABELA DE GRADIENTE
			; *** OK DIFERENÇA DIMINUI ENTAO MANTER O RUMO ***
MANTRUMO:	LJMP	FFTAB	
TABGRAD:			; *** TABELA DE ACERTO DO GRADIENTE DE NAVEGAÇÃO ***
	JB	ZERO,FFTABX	; LATITUDE OU LONGITUDE NÃO TIVERAM MUDANÇA DE VALOR LOGO MANTER
	AJMP	COTAB	; A POSIÇÃO DO BARCO PARA EVITAR ALGUM ERRO DE CÁLCULO
FFTABX:	LJMP	FFTAB	
COTAB:	CLR JB JNB AJMP	C FIDLAT,INCLATX FIDLON,ISALAX ISALBX	; PULA SE INCREMENTOU LATITUDE ; PULA SE DECREMENTOU LONGITUDE
INCLATX:	JNB	FIDLON,ISALBX	; PULA SE DECREMENTOU LATITUDE
ISALAX:			; *** LAT/LON INCREMENTADO OU DECREMENTADO ***
	MOV SUBB JZ JC SJMP	A,DIGLAT A,DIGLON NODLALOG GIRH1 LFGIRNEG	; DIF. GRAUS LAT ; DIF. GRAUS LON ; DESVIA SE NÃO EXISTIR DIFERENÇA ; DIFERENÇA DA LATITUDE MENOR DO QUE DA LONGITUDE ; SENAO DIFERENÇA DA LATITUDE MAIOR DO QUE DA LONGITUDE
GIRH1:	SJMP	LFGIRPOS	
NODLALOG:	MOV SUBB JZ JC SJMP	A,DIMLAT A,DIMLON NODLALOM GIRH2 LFGIRNEG	; DIF. MINUTOS LAT ; DIF. MINUTOS LON ; DESVIA SE NÃO EXISTIR DIFERENÇA ; DIFERENÇA DA LATITUDE MENOR DO QUE DA LONGITUDE ; SENAO DIFERENÇA DA LATITUDE MAIOR DO QUE DA LONGITUDE
GIRH2:	SJMP	LFGIRPOS	
NODLALOM:	MOV SUBB JC SJMP	A,DISLAT A,DISLON GIRH3 LFGIRNEG	; DIF. SEGUNDOS LAT ; DIF. SEGUNDOS LON ; DIFERENÇA DA LATITUDE MENOR DO QUE DA LONGITUDE ; SENAO DIFERENÇA DA LATITUDE MAIOR DO QUE DA LONGITUDE
GIRH3:	SJMP	LFGIRPOS	
ISALBX:			;***** LAT. DECR / LON. INCR -- OU -- LAT.INCR / LON DECR *****
	CLR MOV SUBB	C A,DIGLAT A,DIGLON	; DIF. GRAUS LAT ; DIF. GRAUS LON

```

JZ      NODLALOGY      ; DESVIA SE NÃO EXISTIR DIFERENÇA
JC      GIRH4          ; DIFERENÇA DA LATITUDE MENOR DO QUE DA LONGITUDE
SJMP    LFGIRPOS      ; SENAO DIFERENÇA DA LATITUDE MAIOR DO QUE DA LONGITUDE

GIRH4:
  SJMP    LFGIRNEG

NODLALOGY:
MOV     A,DIMLAT      ; DIF. MINUTOS LAT
SUBBB   A,DIMLON      ; DIF. MINUTOS LON
JZ      NODLALOMY    ; DESVIA SE NÃO EXISTIR DIFERENÇA
JC      GIRH5        ; DIFERENÇA DA LATITUDE MENOR DO QUE DA LONGITUDE
SJMP    LFGIRPOS      ; SENAO DIFERENÇA DA LATITUDE MAIOR DO QUE DA LONGITUDE

GIRH5:
  SJMP    LFGIRNEG

NODLALOMY:
MOV     A,DISLAT      ; DIF. SEGUNDOS LAT
SUBBB   A,DISLON      ; DIF. SEGUNDOS LON
JC      GIRH6        ; DIFERENÇA DA LATITUDE MENOR DO QUE DA LONGITUDE
SJMP    LFGIRPOS      ; SENAO DIFERENÇA DA LATITUDE MAIOR DO QUE DA LONGITUDE

GIRH6:
  SJMP    LFGIRNEG

LFGIRPOS:
CLR     C              ;
MOV     CONT,#1       ; 1x PARA DIREITA (15 GRAUS)
ACALL   GIRABUSS      ; GIRA SENTIDO HORÁRIO

ATDN4:
MOV     11H,#A'       ;
MOV     12H,#4'       ; PRIMEIRA DIREITA PARA LEME
MOV     13H,#2'       ; PRIMEIRA ACELERAÇÃO PARA MANCHE

ACALL   GERATUA       ; GIRA A EMBARCAÇÃO ATÉ ENCONTRAR VALOR DA BÚSSOLA PEDIDO
ACALL   SBPG          ; ANALISA SE A BÚSSOLA ALCANÇOU A DIREÇÃO DESEJADA (+ 30 GRAUS)
JB      F_SBUSPG,ATDN4 ; DIFERENÇA ENTRE OS VALORES > 5 GRAUS... ATUADENOVO4
SETB   HORANT
SJMP    FFTAB

LFGIRNEG:
CLR     C              ;
MOV     CONT,#23      ; GIRA SENTIDO ANTI-HORÁRIO (23 * 15GRAUS = 345 GRAUS)
ACALL   GIRABUSS
CLR     HORANT

ATDN5:
MOV     11H,#A'       ;
MOV     12H,#2'       ; PRIMEIRA ESQUERDA PARA LEME
MOV     13H,#2'       ; PRIMEIRA ACELERAÇÃO PARA MANCHE

ACALL   GERATUA       ; GIRA A EMBARCAÇÃO ATÉ ENCONTRAR VALOR DA BÚSSOLA PEDIDO
ACALL   SBPG          ; ANALISA SE A BÚSSOLA ALCANÇOU A DIREÇÃO DESEJADA (- 30 GRAUS)
JB      F_SBUSPG,ATDN5 ; DIFERENÇA ENTRE OS VALORES > 5 GRAUS... ATUADENOVOS

FFTAB:
ACALL   PTATAN        ; *** PONTO ATUAL VIRA O PONTO ANTERIOR ***
ACALL   DIFLATLONAT   ; DIFERENÇA CALCULADA EM MÓDULO DE LAT-LON

OKLALO1:
; *** FIM DA NAVEGAÇÃO ***

RET

=====
; RECEBE CARACTERES VINDO DO P.C.
=====
PCGER:
MOV     R0,#PRIMEIRO

STR1:
CLR     RXSER          ; NAO RECEBEU DADO
SETB   REN            ; HABILITA RECEPÇÃO DE DADOS

JNB    RXSER,STR1     ; ESPERA A RECEPÇÃO DE DADOS DO P.C.
MOV    @R0,SBUF       ; CARACTER RECEBIDO
INC    R0
CLR    RXSER
DJNZ  CINT,STR1
CLR    REN            ; DESABILITA RECEPÇÃO DE DADOS

; AGORA ANALISA CHECKSUM

MOV    A,#00H         ; ZERA ACUMULADOR
MOV    R0,#PRIMEIRO+1 ; PULA '$'

XK1:
XRL    A,@R0          ; XOR ENTRE END. MEM. ANTER. E POST.
INC    R0
DJNZ  CONT,XK1

MOV    AUX,@R0        ; CHECKSUM REAL
RET

=====
; RECEBE POSIÇÃO DO PTO DESTINO DO P.C. A SER ALCANÇADA
=====
RPTDEST:
MOV    CINT,#9        ; STRING '$GGMSGMSH'
MOV    CONT,#7
ACALL  PCGER

CJNE  A,AUX,XK4       ; DESVIA SE ERRO!!!
SJMP  XK5             ; SUCESSO NA COMUN. COM GPS OEM

```



```

XK4:   CLR      C           ; ENVIA UM ACK DE ERRO PRO P.C.
       ACALL   TEMP002
       MOV     SBUF,#'?
       ACALL   TEMP002
       AJMP    RPTDEST    ; VOLTA PRO INICIO

XK5:   ACALL   TEMP002
       MOV     SBUF,#'!'   ; ACK DE SUCESSO - ENVIA CARACTER PARA P.C.
       ACALL   TEMP002

       MOV     R0,#PRIMEIRO+2 ; *** ARMAZENA VALORES DO PT. DESTINO ***

       MOV     DEGLAT,@R0
       INC     R0

       MOV     DEMLAT,@R0
       INC     R0

       MOV     DESLAT,@R0
       INC     R0

       MOV     DEGLON,@R0
       INC     R0

       MOV     DEMLON,@R0
       INC     R0

       MOV     DESLON,@R0
       CLR     C

       RET

=====
; USO DO PILOTO VIRTUAL PARA TRAZER EMBARCAÇÃO PARA PONTO DE ORIGEM
=====
EMERGENCIA:
       MOV     SBUF,#'E'   ; INDICA AO P.C. QUE PODE COMEÇAR A RECEBER AS ORDENS DE NAVEGAÇÃO

ERORD:
       MOV     CINT,#5
       MOV     CONT,#3
       ACALL   PCGER      ; RECEBE DADOS DO P.C.

       CJNE   A,AUX,XK4E  ; DESVIA SE ERRO!!!
       SJMP   XK5E        ; SUCESSO NA COMUN.

XK4E:  CLR     C           ; ENVIA UM ACK DE ERRO PRO P.C.
       MOV     SBUF,#'?
       AJMP    ERORD      ; ACK DE ERRO - ENVIA CARACTER PARA P.C.
                               ; INICIA VARIÁVEIS

XK5E:  MOV     SBUF,#'!'   ; ACK DE SUCESSO - ENVIA CARACTER PARA P.C.

RNOOK:                               ; ENVIA STRING DE ATUAÇÃO EM POS

       ACALL   LEITPOS    ; REQUISIÇÃO DE LEITURA DE NAVEGAÇÃO PARA POS.

       JB     F_ACKPOS,RESOK ; PULA SE ACK COM SUCESSO
       SJMP   RNOOK      ; VOLTA E ENVIA NOVAMENTE PACOTE DE DADOS PARA POSICIONAMENTO

RESOK:
       ACALL   PACPOS
       ACALL   PACCOM
       RET

#####
;#  ROTINA PRINCIPAL:                                     #
#####

INICIO:
       MOV     IE,#90H    ; INTERRUPÇÃO SERIAL HABILITADA
       MOV     TMOD,#20H ; TIMERS EM MODO 2
       MOV     TCON,#40H ; TIMER 1 LIGADO
       MOV     TH1,#0FDH ; TH1 E TL1 COM VALOR DE 253 PARA GERAR
       MOV     TL1,#0FDH ; TAXA DE TRANSMISSÃO DE 9600 BPS
       MOV     SCON,#40H ; CANAL SERIAL NO MODO 1
       SETB   TXD        ; STOP BIT, SEM TRANSMISSÃO

       SETB   LED        ; ACENDE O LED
       ACALL  TEMP1      ; ESPERA 2s
       CLR   LED         ; APAGA O LED
       CLR   RXSER       ; NAO RECEBEU DADO
       SETB  REN         ; HABILITA RECEPÇÃO DE DADOS

WAIT:
       JNB   RXSER,WAIT ; ESPERA A RECEPÇÃO DE DADOS DO PC
       CLR  RXSER       ; LIMPA FLAG DE EMERGENCIA
       CJNE A,#'W',WAIT1 ; PILOTO VIRTUAL
       ACALL EMERGENCIA ; CONTROLE MANUAL DA EMBARCAÇÃO
       SETB REN
       SJMP WAIT

WAIT1:
                               ; INICIO DO RECEBIMENTO DOS DADOS

       JNB   RXSER,WAIT1
       CLR  RXSER
       CJNE A,#'@',WAIT1
       MOV  SBUF,#'!'

       ACALL  TELEM      ; TELEMETRIA DE DADOS COMPLETA PARA NAVEGAÇÃO

```

```

; *** OK BARCO PRONTO PRA NAVEGAR ***
; *** FAZER LEITURA DO GPS,A/D E BUSS ENTRE AS NOVAS MEDIDAS ***
ACALL  PRENAV      ; PRÉ-CONFIGURAÇÃO PARA NAVEGAÇÃO (ARMAZENA PTO COMO ANTERIOR
;E PEGA PTO DESTINO DO P.C.)

MOV    11H,#'A'
MOV    12H,#'3'
MOV    13H,#'2'      ; CENTRO DO LEME
; PRIMEIRA ACELERAÇÃO PARA MANCHE

; *** 1A. NAVEGAÇÃO ***
ACALL  LEITPOS
ACALL  PACPOS      ; REQUISIÇÃO DE LEITURA DE NAVEGAÇÃO PARA POS.
; RECEBE PACOTE DE POSICIONAMENTO

SETB  REN
CLR   RXSER      ; HABILITA CANAL DE EMERGÊNCIA
; LIMPA FLAG DE INTERRUPÇÃO

NORUMO:      ; *** PROCURA O RUMO ***

JB    RXSER,EMERG
SJMP  OPNORM1    ; EMERGÊNCIA - USO DO PILOTO VIRTUAL POR PROBLEMAS NA EMBARCAÇÃO

EMERG:
CLR   RXSER
SETB  REN
MOV   SBUF,#'V'
SJMP  WAIT      ; HABILITA PILOTO VIRTUAL POR INT. HARDWARE
; INDICA AO P.C. -> CARAC PILOTO VIRTUAL
; PROGRAMA PARA DE SEGUIR FLUXO NORMAL E PULA LOOP DE CHAMADA PARA EMERGÊNCIA

OPNORM1:
ACALL  TELEM      ; TELEMETRIA DE DADOS COMPLETA PARA NAVEGAÇÃO

ACALL  NAVRUMO

JB    ZEROLAT,VERLON1
SJMP  VERUMO

VERLON1:
JB    ZEROLON,FIMNAVEG  ; DIFERENÇA ZERO ENTRE PONTO ATUAL E DESTINO (LONGITUDE) = FIM DA NAVEGAÇÃO!!!

VERUMO:
JNB   RUMCER,NORUMO  ; VOLTA SE NAO ENCONTROU O RUMO.
; RUMO ENCONTRADO

NOGRAD:      ; *** GRADIENTE DE NAVEGAÇÃO ***

JB    RXSER,EMERG  ; EMERGÊNCIA - USO DO PILOTO VIRTUAL POR PROBLEMAS NA EMBARCAÇÃO

ACALL  TELEM      ; TELEMETRIA DE DADOS COMPLETA PARA NAVEGAÇÃO

ACALL  NAVGRAD    ; GRADIENTE DE NAVEGAÇÃO

JB    ZEROLAT,VERLON2
JB    ZEROLON,NORUMO
SJMP  NOGRAD      ; DIFERENÇA ZERO DE LATITUDE
; LONG=0 MAS LAT<->0 RECOMEÇAR ALGORITMO
; LAT<->0 E LONG<->0 GRADIENTE DE NAVEGAÇÃO

VERLON2:
JB    ZEROLON,FIMNAVEG
SJMP  NORUMO      ; DIFERENÇA ZERO DE LONGITUDE = FIM DA NAVEGAÇÃO!!!
; LAT=0 MAS LONG<->0 RECOMEÇAR ALGORITMO

FIMNAVEG:    ; ÚLTIMO PACOTE

MOV    ESTADO,#'S'
ACALL  LEITPOS      ; REQUISIÇÃO DE LEITURA DE NAVEGAÇÃO PARA POS.

ACALL  PACPOS
ACALL  PACCOM      ; RECEBE PACOTE DE POSICIONAMENTO
; ENVIA PACOTE DE DADOS PARA O COMPUTADOR

LOOPX:
AJMP  LOOPX      ; VOLTA AO INÍCIO DO LOOP

#####
;# FIM DO PROGRAMA: #
#####

END

```

ANEXO C – Sistema de Posicionamento

```
#####
PROJETO.....: TELEMETRIA DE DADOS E IMAGENS PARA
                PLATAFORMA AUTÔNOMA PARA COLETA DE DADOS
                HIDROLÓGICOS.
DESENVOLVIDO POR.....: ALEX AYRES STAVARENGO
VERSÃO.....: POS22   DATA.....: 01/06/2006
=====
OBJETIVO DO PROGRAMA:
- CAPTURAR DADOS DE TELEMETRIA (GPS,BÚSSOLA,MANCHE E LEME) E
  ENVIÁ-LOS PARA O SIST. GERENCIAMENTO.
=====
DESCRIÇÃO DO PROGRAMA:

A PLACA DE POSICIONAMENTO RECEBE DO SISTEMA DE GERENCIAMENTO UMA
REQUISIÇÃO DE LEITURA DE DADOS DE TELEMETRIA. A PLACA CAPTURA ESSES
DADOS E OS ENVIÁ PARA O SISTEMA DE GERENCIAMENTO.

*** PREVENÇÃO DE DEADLOCK NO GPS OEM   E CHECKSUM DOS DADOS
#####
```

<<< DESCRIÇÃO DAS ROTINAS UTILIZADAS PELO SISTEMA >>>

```
-----
TEMP1 -> Rotina de atraso de tempo de aprox. 1 segundo

TEMP2 -> Rotina de atraso de tempo de aprox. 2 segundos

TEMP002-> Rotina de atraso de tempo de aprox. 2 milisegundos

DEZENA -> Transforma um caractere em um valor decimal (0..9)
- Entrada: caractere no endereço apontado por R0
- Saída: número colocado em Acc

SERINI -> Inicia canal serial próprio para transmissão de dados (9600bps) para o Sistema de Gerenciamento

BUSINI -> Inicialização e calibração da bússola eletrônica

BUSCONV -> Efetua a leitura da bússola (0..359 graus)
- Saída: BUSH (MSB) e BUSL (MLB)

ADCONV -> Converte dados analógicos em digitais do leme e manche
- Entrada: palavra de controle em Acc (97H para leme e DFH para manche)
- Saída: valor digital armazenado em Acc

POSBUS -> Faz a leitura da bússola e armazena na memória
- Saída: $BSDX
- Posição na memória: 15H a 19H

        '$' - demarcador
        'B' - indicador de bússola
        'S' - byte mais significativo da bússola
        'D' - byte menos significativo da bússola
        'X' - checksum ( menos caractere '$')

POSGPS -> Rotina de recepção de dados do GPS a 4800bps
- Saída: $GPRMC,234810.000,A,2200.377,S,04753.863,W,0.0,78.6,060405,0.0,W*4C. (exemplo de valores)
- Posição na memória: 10H a 1FH - 30H a 6D

CHECKSUM -> Rotina utilizada por POSGPS que indica em F_CHECK se a com. serial com o GPS foi realizada com sucesso
- Saída: F_CHECK = 0 temos comunicação perfeita
        F_CHECK = 1 temos ruído na comunicação

MEMGPS -> Rotina utilizada por POSGPS que reconstrói a string GPS na memória, fazendo a conversão de valores de caractere
para valor real quando necessário e também armazenando somente o estritamente necessário
- Saída: $G,A,GMS,X,GMS,Y,V**H%%
- Posição na memória: 1AH À 1FH - 30H A 40H

        '$' - demarcador
        'G' - indicador do GPS
        '!' - separador
        'A' - caractere de estado de funcionamento:
            A: posição válida
            V: posição inválida
            T: GPS não encontrado (sem alimentação)
            X: erro de checksum
        '!' - separador
        'GMS' - graus, minutos e segundos da latitude em valores reais
        '!' - separador
        'X' - orientação da latitude (Norte ou Sul)
        '!' - separador
        'GMS' - graus,minutos e segundos da latitude em valores reais
        '!' - separador
        'Y' - orientação da longitude (oeste e leste)
        '!' - separador
        'V' - velocidade da embarcação em nós
        '***' - acusa que próximo caractere é o checksum
        'H' - checksum ( menos caractere '$')
        '%%' - indica fim do pacote de dados

S2RECD -> Canal serial implementado para uso na recepção da string do G.P.S. com velocidade de 4800 bps
- Saída: variável SER2DR = byte recebido

ENVPACOTE -> Envia pacote de dados para o Sistema de Gerenciamento
- Saída: $NLMX$BSDX$G,A,GMS,X,GMS,Y,V**H%%
- Posição na memória:10H a 1FH - 30H a 40H (33 bytes no total)
```

RECGER -> Recebe do Sist. Gerenciamento um pacote de dados de 5 bytes pelo canal serial próprio na posição de memória de 10H a 14H
- Saída: \$ALMH

'\$' - demarcador
'A' - indicador de navegação (A=atuar; L=leitura; S=parar; R=resetar)
'L' - valor digital do leme
'M' - valor digital do manche
'X' - checksum (menos caractere '\$')

CHKGER -> Analisa o checksum que veio no pacote recebido do sist. de gerenciamento e retorna um ack de estado
- Saída: F_ACKGER = 1 pacote recebido íntegro
F_ACKGER = 0 pacote corrompido

DEFATUA -> Define atuação sobre a embarcação (leitura,parar,emergencia,atuar)
- Saída: \$NLMX

- Posição na memória: 10H a 14H

'\$' - demarcador
'N' - indicador de navegação
'L' - valor digital do leme
'M' - valor digital do manche
'X' - checksum (menos caractere '\$')

ACVALEM -> Transforma caractere de controle vindo do sist. Ger. para um valor digital correto para leme
- Saída: valor digital colocado em DESLLEM

ACVALMAN -> Transforma caractere de controle vindo do sist. Ger. para um valor digital correto para manche
- Saída: valor digital colocado em DESLMAN

XLEME -> Realiza a mudança de rumo do leme de acordo com o valor da variável DESLLEM

XMANCHE -> Realiza a mudança de rumo do manche de acordo com o valor da variável DESLMAN

RES_BOAT -> Coloca a embarcação em posição de saída pra trabalho
- Saída: leme centralizado e manche sem aceleração

```
#####
# VARIÁVEIS:                                     #
#####

RXSER          EQU    7FH    ; INDICADOR DE DADO RECEBIDO
RECEB          EQU    7EH    ; CARACTER RECEBIDO
SER2DR         EQU    7DH    ; DADO RECEBIDO PELO CANAL 2

AUX1           EQU    7CH
AUX2           EQU    7BH
CONT           EQU    7AH

BUSH           EQU    79H    ; BYTE MAIS SIGN. DO DADO DA BÚSSOLA
BUSL           EQU    78H    ; BYTE MENOS SIGN. DO DADO DA BÚSSOLA

MANCHE         EQU    77H    ; VALOR DIGITAL DO MANCHE
LEME           EQU    76H    ; VALOR DIGITAL DO LEME

CONTT          EQU    75H
AUX            EQU    74H
CHSUM          EQU    73H
DESLMAN        EQU    72H    ; VALOR DE DESLOCAMENTO PARA NAVEGAÇÃO MANCHE
DESLLEM        EQU    71H    ; VALORES DE DESLOCAMENTO PARA NAVEGAÇÃO LEME
ATUALMAN       EQU    70H    ; VALOR ATUAL DE MANCHE
ATUALLEM       EQU    6FH    ; VALOR ATUAL DE LEME

LATLON         EQU    20H    ; GUARDA VALOR NUMÉRICO DE LATITUDE E LONGITUDE
;DA STRING DE DADOS ORIGINAL

#####
;# FLAGS:                                         #
#####

F_TMR0         EQU    08H    ; = 1 ENTÃO ESTOURO DO TIMER0 PARA O GPS
F_CHECK        EQU    09H    ; = 1 ENTÃO VALIDADE DO CHECKSUM DO GPS OEM
F_ACKGER       EQU    0AH    ; ACK DO PACOTE QUE VEM DO SIST. GER. ( = 1 ERRO NO PACOTE )

#####
;# ENTRADAS E SAÍDAS:                             #
#####

RXD2           EQU    0B6H    ; P3.6, RECEPÇÃO DE DADOS SERIAL DO GPS
LED            EQU    0B2H    ; PORTA P3.2, PARA ACIONAMENTO DO LED

##### BÚSSOLA ELETRÔNICA #####

SS             EQU    80H    ; PORTA P0.0, PARA SELEÇÃO DE ESCRAVO
SCLK           EQU    81H    ; PORTA P0.1, PARA O CLOCK SERIAL
PC            EQU    82H    ; PORTA P0.2, PARA PEDIDO DE LEITURA
BRST          EQU    83H    ; PORTA P0.3, PARA RESETAR A BÚSSOLA
CI            EQU    84H    ; PORTA P0.4, PARA IND. DE CALIBRAÇÃO
EOC           EQU    85H    ; PORTA P0.5, PARA IND. FIM DE CONVERSÃO
SDO           EQU    86H    ; PORTA P0.6, PARA SAÍDA DE DADOS SERIAIS

##### CONVERSOR A/D #####

CS             EQU    90H    ; PORTA P1.0, HABILITAÇÃO DO CONV. A/D
DCLK          EQU    91H    ; PORTA P1.1, CLOCK DO CONV. A/D
```

```
DIN      EQU      92H      ; PORTA P1.2, ENTRADA DADOS DO CONV. A/D
DOUT    EQU      93H      ; PORTA P1.3, SAIDA DADOS DO CONV. A/D
```

```
##### PLACA DE RELÉS #####
```

```
*** HARDWARE DESLIGADO E CABOS CONECTADOS = COMANDO MANUAL ATIVADO
*** HARDWARE LIGADO E BIT SWE ATIVADO (NÍVEL 0) -> COMANDO MANUAL ATIVADO
*** HARDWARE LIGADO E BIT SWE NÃO ATIVADO (NÍVEL 1) -> COMANDO MANUAL DESATIVADO
```

```
SWE      EQU      87H      ; PORTA P0.7, HABILITAÇÃO DE COMANDO MANUAL DA EMBARCAÇÃO
DIRE     EQU      0A0H     ; PORTA P2.0, DIRE + SDIRE LIGA INVERTE MOTOR -> ESQUERDA
SDIRE    EQU      0A1H     ; PORTA P2.1, LIGAR MOTOR DO LEME -> DIREITA
ACELE    EQU      0A2H     ; PORTA P2.2, ACELE + SACELE INVERTE MOTOR -> DESACELERAÇÃO
SACELE   EQU      0A3H     ; PORTA P2.3, LIGA MOTOR DO MANCHE -> ACELERAÇÃO
STARTE   EQU      0A4H     ; PORTA P2.4, PARTIDA MOTOR DE COMBUSTÃO
STOPE    EQU      0A5H     ; PORTA P2.5, PARADA MOTOR DE COMBUSTÃO
```

```
#####
# VETOR DE RESET:                                     #
#####
```

```
ORG      00H      ; ENDEREÇO DE INICIALIZAÇÃO
MOV      SP,#22H  ; POSICIONA APONTADOR DE PILHA
AJMP     INICIO   ; DESVIA PARA A ROTINA DE INICIALIZAÇÃO
```

```
#####
# ROTINAS DE INTERRUPÇÃO:                             #
#####
```

```
ORG      0BH      ; INTERRUPÇÃO DO TIMER0
DJNZ     CONTT,TIM0 ; DESVIA SE CONTT<=0
SETB     F_TMR0    ; DEADLOCK DO SISTEMA
TIM0:    RETI
```

```
ORG      23H      ; INTERRUPÇÃO DO CANAL SERIAL
JB       TI,XXX   ; VÁ PARA XXX SE FOR TRANSMISSÃO
CLR      RI       ; LIMPA RI, QUE INDICA QUE A INT OCORREU
MOV      A,SBUF   ; A = CARACTER RECEBIDO
MOV      RECEB,A  ; RECEB = CARACTER RECEBIDO
SETB     RXSER    ; RECEBEU DADO
SJMP     YYY
```

```
XXX:
YYY:    CLR      TI      ; INDICAÇÃO DE TRANSMISSÃO DE DADO
        RETI         ; RETORNA DA INTERRUPÇÃO
```

```
#####
# ROTINAS:                                             #
#####
```

```
; TEMPO DE 1s
```

```
=====
```

```
TEMP1:  MOV      R6,#TEMPO3 ; ROTINA PARA CONTAGEM DE 2s
        ; CARREGA R6 COM VALOR DE TEMPO3
T122:   MOV      R5,#TEMPO2 ; CARREGA R5 COM VALOR DE TEMPO2
T123:   MOV      R4,#TEMPO1 ; CARREGA R4 COM VALOR DE TEMPO1
T124:   DJNZ     R4,T124    ; DECREMENTA R4, VOLTA À T24 SE DIF. DE 0
        DJNZ     R5,T123    ; DECREMENTA R5, VOLTA À T23 SE DIF. DE 0
        DJNZ     R6,T122    ; DECREMENTA R6, VOLTA À T22 SE DIF. DE 0
        RET
```

```
=====
```

```
; TEMPO DE 2s
```

```
=====
```

```
TEMP2:  MOV      R7,#TEMPO4 ; CARREGA R7 COM VALOR DE TEMPO4
T21:    MOV      R6,#TEMPO3 ; CARREGA R6 COM VALOR DE TEMPO3
T22:    MOV      R5,#TEMPO2 ; CARREGA R5 COM VALOR DE TEMPO2
T23:    MOV      R4,#TEMPO1 ; CARREGA R4 COM VALOR DE TEMPO1
T24:    DJNZ     R4,T24     ; DECREMENTA R4, VOLTA À T24 SE DIF. DE 0
        DJNZ     R5,T23     ; DECREMENTA R5, VOLTA À T23 SE DIF. DE 0
        DJNZ     R6,T22     ; DECREMENTA R6, VOLTA À T22 SE DIF. DE 0
        DJNZ     R7,T21     ; DECREMENTA R7, VOLTA À T21 SE DIF. DE 0
        RET
```

```
=====
```

```
; TEMPO DE 20ms
```

```
=====
```

```
TEMP002: MOV      R7,#TEMPO5 ; CARREGA R7 COM VALOR DE TEMPO5
T021:    MOV      R6,#TEMPO1 ; CARREGA R6 COM VALOR DE TEMPO1
T022:    DJNZ     R6,T022    ; DECREMENTA R6, VOLTA À T022 SE DIF DE 0
        DJNZ     R7,T021    ; DECREMENTA R7, VOLTA À T021 SE DIF DE 0
        RET
```

```
=====
```

```
; TRANSFORMA VALORES EM CARACTER PRA DEZENA NUMÉRICA
```

```
=====
```

```
DEZENA: MOV      A,@R0      ; CARACTERE NO ACUMULADOR
```

```

SUBB    A,#48          ; DESLOCAMENTO NA TABELA ASCII
CLR     C              ; LIMPA CARRY
MOV     LATLON,A
RL      A              ; x2
RL      A              ; x4
RL      A              ; x8
ADD     A,LATLON       ; SOMA O NUMERAL
ADD     A,LATLON       ; SOMA DE NOVO
MOV     LATLON,A
INC     R0              ; UNIDADE DA DEZENA NA STRING DE DADOS GPS
MOV     A,@R0
SUBB   A,#48
ADD     A,LATLON       ; ADICIONA DEZENA À UNIDADE
RET

=====
; INICIA CANAL SERIAL
=====
SERINI:
MOV     IE,#92H        ; INTERRUPÇÃO SERIAL/TIMER 0/TIMER 1 HABILITADA
MOV     TMOD,#21H      ; TIMER1 EM MODO 2 (REGARGA AUT.) E TIMER0 EM MODO 1 (16 BITS)
MOV     TH1,#0FDH      ; TH1 E TL1 COM VALOR DE 253 PARA GERAR
MOV     TL1,#0FDH      ; TAXA DE TRANSMISSÃO DE 9600 BPS
MOV     TH0,#0FFH
MOV     TL0,#0FDH
MOV     SCON,#40H      ; CANAL SERIAL NO MODO 1 (TAXA VARIÁVEL)
SETB   TR1             ; HABILITA TIMER 1 PARA CONTAGEM DE 9600 BPS
RET     ; FIM DA ROTINA

=====
; INICIA BÚSSOLA
=====
BUSINI:
SETB   SS              ; INICIALIZAÇÃO DA BÚSSOLA
ACALL  TEMP002
SETB   PC
ACALL  TEMP002
SETB   SCLK
ACALL  TEMP002
SETB   BRST
MOV     R0,#04

BI:
CLR    BRST
ACALL  TEMP002
SETB   BRST
ACALL  TEMP002
DJNZ   R0,BI
RET     ; FIM DA ROTINA

=====
; LEITURA DA BÚSSOLA
=====
BUSCONV:
CLR    PC              ; LEITURA DE DADO CONVERTIDO PELA BÚSSOLA
ACALL  TEMP002        ; NÍVEL 0 EM PC PARA REQUISITAR LEITURA
SETB   PC              ; ESPERA 20ms
BUS1:
JNB    EOC,BUS1       ; AGUARDA EOC = 1
ACALL  TEMP002        ; ESPERA 20ms
CLR    SS              ; NÍVEL 0 EM SS PARA RETIRAR OS DADOS
ACALL  TEMP002        ; ESPERA 20ms
MOV     R0,#8          ; 8 PRIMEIROS BITS A RECEBER (MSB)

BUS2:
CLR    SCLK            ; NÍVEL 0 NO CLOCK
ACALL  TEMP002
SETB   SCLK            ; NÍVEL 1 NO CLÓCK
ACALL  TEMP002
MOV     C,SDO          ; BIT LIDO COLOCADO NO CARRY
RLC    A               ; BIT LIDO COLOCADO NO ACUMULADOR
DJNZ   R0,BUS2        ; SE NÃO RECEBEU 8 BITS, VOLTA A BUS2
MOV     BUSH,A         ; 8 BITS LIDOS COLOCADOS EM BUSMSB
ACALL  TEMP002
MOV     R0,#8          ; 8 ÚLTIMOS BITS A RECEBER (LSB)

BUS3:
CLR    SCLK            ; NÍVEL 0 NO CLOCK
ACALL  TEMP002
SETB   SCLK            ; NÍVEL 1 NO CLOCK
ACALL  TEMP002
MOV     C,SDO          ; BIT LIDO COLOCADO NO CARRY
RLC    A               ; BIT LIDO COLOCADO NO ACUMULADOR
DJNZ   R0,BUS3        ; SE NÃO RECEBEU 8 BITS, VOLTA A BUS3
MOV     BUSL,A         ; 8 BITS LIDOS COLOCADOS EM BUSLSB
ACALL  TEMP002        ; ESPERA 20ms
SETB   SS              ; NÍVEL 1 EM SS, FIM DA RETIRADA DE DADOS
RET     ; FIM DA ROTINA

=====
; CONVERSÃO DE ANALÓGICO PARA DIGITAL
=====
ADCONV:
CLR    CS              ; CONVERSÃO ANALÓGICA PARA DIGITAL
MOV     R1,#8          ; CONVERSOR A/D HABILITADO
                        ; 8 BITS A SEREM ENVIADOS, 8 CLOCKS

ADC1:
CLR    DCLK            ; NÍVEL DO CLOCK = 0
RLC    A               ; BIT A SER ENVIADO NO CARRY
MOV     DIN,C          ; BIT NO SINAL DIN
SETB   DCLK            ; NÍVEL DO CLOCK = 1
DJNZ   R1,ADC1        ; SE NÃO ENVIOU 8, VOLTA A ADC1
CLR    DIN              ; DADOS DE CONTROLE = 0
CLR    DCLK            ; NÍVEL DO CLOCK = 0
RLC    A               ; VOLTA ACUMULADOR AO VALOR ORIGINAL

```

```

ADC2:  MOV      R1,#8          ; 8 BITS A SEREM RECEBIDOS, 8 CLOCKS
      SETB     DCLK          ; NÍVEL DO CLOCK = 1
      CLR      DCLK          ; NÍVEL DO CLOCK = 0
      MOV      C,DOUT        ; BIT LIDO COLOCADO NO CARRY
      RLC      A             ; BIT LIDO COLOCADO NO ACUMULADOR
      DJNZ    R1,ADC2        ; SE NÃO RECEBEU 8, VOLTA A ADC2
      SETB     CS            ; DESABILITA O CONVERSOR D/A
      CLR      C             ;
      RET                          ; FIM DA ROTINA
=====
; RECEPÇÃO DE DADOS DA BÚSSOLA
=====
POSBUS:
      ACALL    BUSSCONV      ; CAPTURA DADO DA BÚSSOLA
      MOV      A,#00H
      MOV      R0,#PRIMEIRO+5 ; APÓS DADOS DO CONVERSOR

      MOV      @R0,#'$'      ; CARACTER '$'
      INC      R0

      MO       @R0,#'B'      ; CARACTER 'B'
      XRL     A,#'B'         ; XOR COM 1o. CARACTER VÁLIDO
      INC      R0

      MOV      @R0,BUSH      ; PARTE ALTA DO DADO DA BÚSSOLA
      XRL     A,BUSH
      INC      R0

      MOV      @R0,BUSL      ; PARTE BAIXA DO DADO DA BÚSSOLA
      XRL     A,BUSL
      INC      R0

      MOV      @R0,A         ; ARMAZENA CHECKSUM DA BÚSSOLA
      RET
=====
; RECEPÇÃO DE DADOS DO GPS
=====
POSGPS:
      CLR      F_TMR0        ; TIMER0 NÃO ESTOURADO
      SETB     TR0           ; HABILITA TIMER0 DE 16BITS
      MOV      CONTT,#30H    ; 3s ANTES DO DEADLOCK

INISTR:
      MOV      AUX,#16       ; 8 BITS DE FIM DE STRING (STOP BIT)
STPBIT: JNB     RXD2,INISTR   ; SE NAO FOR FIM, RECOMEÇA CONTAGEM!
      MOV      AUX1,#STEMP2  ; SEMPRE OLHAR MEIO PERÍODO DO BIT
BITT:   DJNZ   AUX1,BITT
      DJNZ   AUX,STPBIT

GPRMC:  JB      F_TMR0,JDEADGPS ; DEADLOCK NO SISTEMA
      SJMP   INIGPRMC

JDEADGPS: LJMP   DEADGPS      ; PULO RELATIVO POIS PROGRAMA É MUITO GRANDE

INIGPRMC:
      MOV      R0,#PRIMEIRO+10 ; POSIÇÃO PÓS BÚSSOLA

      LCALL    S2RECD         ; RECEBE CARACTER PELO CANAL SERIAL 2
      CJNE    A,#'$',GPRMC   ; CARACTER = '$'?
      MOV      @R0,#'$'      ; CARACTER '$'
      INC      R0

      LCALL    S2RECD         ; RECEBE CARACTER PELO CANAL SERIAL 2
      CJNE    A,#'G',GPRMC   ; CARACTER = 'G'?
      MOV      @R0,#'G'      ; CARACTER 'G'
      INC      R0

      LCALL    S2RECD         ; RECEBE CARACTER PELO CANAL SERIAL 2
      CJNE    A,#'P',GPRMC   ; CARACTER = 'P'?
      MOV      @R0,#'P'      ; CARACTER 'P'
      INC      R0

      LCALL    S2RECD         ; RECEBE CARACTER PELO CANAL SERIAL 2
      CJNE    A,#'R',GPRMC   ; CARACTER = 'R'?
      MOV      @R0,#'R'      ; CARACTER 'R'
      INC      R0

      LCALL    S2RECD         ; RECEBE CARACTER PELO CANAL SERIAL 2
      CJNE    A,#'M',GPRMC   ; CARACTER = 'M'?
      MOV      @R0,#'M'      ; CARACTER 'M'
      INC      R0

      LCALL    S2RECD         ; RECEBE CARACTER PELO CANAL SERIAL 2
      CJNE    A,#'C',GPRMC   ; CARACTER = 'C'?
      MOV      @R0,#'C'      ; CARACTER 'C'

      CLR      C             ;
      MOV      R0,#SEGUNDO    ; 2A. POSIÇÃO DE MEMÓRIA DE DADOS
      MOV      CONT,#62      ; NMEA SEGUNDA PARTE

MMM:   LCALL    S2RECD         ; RECEBE NMEA DO GPS
      MOV      @R0,SER2DR
      INC      R0
      DJNZ    CONT,MMM       ; DESVIA SE FIM DE NMEA

```



```

        CLR          TR0                ; OPERAÇÃO BEM SUCEDIDA SEM ESTOURO DE TIMER0

PVIR1:  MOV          R0,#SEGUNDO
        CJNE        @R0,#',,PVIR2      ; DESVIA SE NAO É VÍRGULA
        SJMP        PVIR3              ; ACHOU
PVIR2:  CLR          C
        INC         R0
        SJMP        PVIR1

PVIR3:
PVIR5:  INC         R0
        CJNE        @R0,#',,PVIR4      ; DESVIA SE NAO É SEGUNDA VÍRGULA
        SJMP        PVIR6              ; ACHOU VÍRGULA
PVIR4:  CLR          C
        INC         R0
        SJMP        PVIR5

PVIR6:
        INC         R0                ; AGORA R0 TEM ENDEREÇO DO ESTADO DE VALIDADE DO SINAL GPS
        MOV         A,@R0
        CJNE        A,#'V',CHECKK      ; DESVIA SE VALOR VÁLIDO - CARACTER 'A'

        MOV         R0,#1AH            ; SENAO SINAL SEM SATÉLITE ENCONTRADO

        MOV         @R0,#'$'           ; CARACTER '$'
        INC         R0

        MOV         @R0,#'G'           ; CARACTER 'G'
        INC         R0

        MOV         @R0,#','           ; CARACTER ','
        INC         R0

        MOV         @R0,#'V'           ; CARACTER 'V' -> POSIÇÃO INVÁLIDA
        INC         R0

        MOV         @R0,#'%'           ; CARACTER '%' -> FIM DO PACOTE DE DADOS
        INC         R0

        MOV         @R0,#'%'           ; RECONFIRMA FIM DE TRANSMISSAO DE PACOTE DE DADOS

        SJMP        FIMGPS

CHECKK:
        ACALL       CHECKSUM           ; VALIDA OPERAÇÃO
        JB          F_CHECK,DEADGPS    ; CHECKSUM INVÁLIDO OU TIMER ESTOURADO
        ACALL       MEMGPS             ; RECONSTRÓI STRING GPS NA MEMÓRIA PARA ECONOMIA DE ESPAÇO
        SJMP        FIMGPS             ; SUCESSO NA COMUNICAÇÃO COM O GPS!!!

DEADGPS:
        CLR         TR0                ; ARRUMA STRING PARA DEADLOCK OU ERRO NO CHECKSUM
        MOV         R0,#PRIMEIRO+10   ; DESLIGA INTERRUPTÃO DO TIMER0
        MOV         @R0,#'$'           ; CARACTER '$'
        INC         R0
        MOV         @R0,#'G'           ; CARACTER 'G'
        INC         R0
        MOV         @R0,#','           ; CARACTER ','
        INC         R0
        JB          F_TMR0,STTMR0      ; PULA SE PROBLEMA FOI DO DEADLOCK
        MOV         @R0,#'X'           ; SENÃO CARACTER 'X' -> CHECKSUM INVÁLIDO

        SJMP        STCHECK

STTMR0:
        MOV         @R0,#'T'           ; ACUSA DEADLOCK NO SISTEMA

STCHECK:
        INC         R0
        MOV         @R0,#'%'           ; CARACTER '%' -> FIM DO PACOTE DE DADOS
        INC         R0
        MOV         @R0,#'%'           ; CARACTER '%' -> CONFIRMA FIM DO PACOTE DE DADOS

FIMGPS:
        RET

;=====
; CHECKSUM DOS DADOS DO GPS
;=====
CHECKSUM:
        CLR         C                    ; LIMPA CARRY

        CLR         F_CHECK              ; ZERA FLAG DE INDICAÇÃO DE ERRO NO CHECKSUM (0=VÁLIDO / 1=INVÁLIDO)

        MOV         A,#00H              ; ZERA ACUMULADOR
        MOV         R0,#PRIMEIRO+11     ; PULA O '$'
        MOV         CONT,#5             ; CARACTERES MENOS O '$' NO BANCO 2/3
CHK1:   XRL         A,@R0                ; XOR ENTRE END. MEM. ANTER. E POST.
        INC         R0
        DJNZ        CONT,CHK1

        MOV         R0,#SEGUNDO          ; CONTINUAÇÃO
        CJNE        @R0,#*',CHK3        ; DESVIA SE NAO CHEGOU FIM DE STRING '*'
        SJMP        CHK6
CHK3:   CLR         C                    ; FIM DE STRING
        XRL         A,@R0                ; CALCULA XOR
        INC         R0
        SJMP        CHK2
CHK6:   CLR         C                    ; VOLTA PRA INICIO DE CALCULO

```

```

INC      R0
MOV      AUX,A           ; VALOR CALCULADO
MOV      A,@R0
SUBB    A,#58           ; VALOR ACIMA DO DECIMAL 9 NA TAB. ASCII
JC      HJK1           ; CARRY ESTOUROU? ENTAO É NÚMERO
MOV      A,@R0           ; PEGA 1o. CARAC. CHECK
SUBB    A,55           ; É LETRA (A,B,C,D,E,F) -> TRANSFORMA EM DECIMAL
SWAP    A               ; TROCA 4MSB-4LSB
MOV      CHSUM,A        ; ARMAZENA 1o. VALOR
SJMP    ZXC1
HJK1:   MOV      A,@R0           ; PEGA 1o. CARAC. CHECK
SUBB    A,#47           ; É NÚMERO (TRANSFORMA EM DECIMAL)
SWAP    A               ; TROCA 4MSB-4LSB
MOV      CHSUM,A        ; ARMAZENA 1o. VALOR
ZXC1:   CLR      C               ; LIMPA CARRY PRA NOVO CALCULO
INC      R0
MOV      A,@R0           ; VALOR ACIMA DO DECIMAL 9 NA TAB. ASCII
SUBB    A,#58           ; CARRY ESTOUROU? ENTAO É NÚMERO
JC      HJK2           ; PEGA 2o. CARAC. CHECK
MOV      A,@R0           ; É LETRA (A,B,C,D,E,F) -> TRANSFORMA EM DECIMAL
SUBB    A,#55           ;
SWAP    A               ;
SJMP    CHKXY
HJK2:   MOV      A,@R0           ; PEGA 2o. CARAC. CHECK
SUBB    A,#47           ; É NÚMERO (TRANSFORMA EM DECIMAL)
CHKXY:  ORL      A,CHSUM       ; RESULTADO FINAL!!!
CJNE    A,AUX,CHK4      ; DESVIA SE ERRO!!!
SJMP    CHK5
CHK4:   CLR      C               ; SUCESSO NA COMUN. COM GPS OEM
CHK5:   SETB    F_CHECK       ; ERRO DE COMUN. COM GPS
CLR      C
RET

```

```

=====
; RECONSTROI STRING GPS NA MEMÓRIA
=====

```

```

MEMGPS:
MEM1:   MOV      R0,#SEGUNDO
CJNE    @R0,#',',MEM2   ; DESVIA SE NAO FOR ','
SJMP    MEM3
MEM2:   CLR      C
INC     R0
SJMP    MEM1           ; VOLTA PRA INICIO DE CALCULO
MEM3:   CLR      C
INC     R0             ; PULA 1a. VÍRGULA
MEM4:   CJNE    @R0,#',',MEM5 ; DESVIA SE NAO FOR ','
CLR     C
SJMP    MEM6
MEM5:   CLR      C
INC     R0
SJMP    MEM4           ; VOLTA PRA INICIO DE CALCULO
MEM6:   INC     R0             ; PULA 2a. VÍRGULA = CARACTERE DE ALERTA DE NAVEGAÇÃO

MOV     R1,#PRIMEIRO+12 ; POSIÇÃO $G(X)
MOV     @R1,#','        ; ARMAZENA ','
INC     R1
MOV     A,@R0           ; ARMAZENA ALERTA DO RECEPTOR DE NAVEGAÇÃO
MOV     @R1,A
INC     R1
MOV     @R1,#','        ; ARMAZENA ','
INC     R1             ; PRÓXIMA POSIÇÃO
INC     R0
INC     R0
ACALL   DEZENA         ; GRAUS DA LATITUDE
MOV     @R1,A          ; ARMAZENA GRAUS DA LATITUDE

MOV     R1,#SEGUNDO   ; SEGUNDA PARTE DA MEMORIA

INC     R0
ACALL   DEZENA
MOV     @R1,A          ; ARMAZENA MINUTOS DA LATITUDE
INC     R1
INC     R0             ; PULA O PONTO "2201.896"
INC     R0
ACALL   DEZENA
MOV     @R1,A          ; ARMAZENA SEGUNDOS DA LATITUDE
INC     R1
MOV     @R1,#','        ; ARMAZENA ','
INC     R1
INC     R0
INC     R0
INC     R0             ; PULA MILHAR DOS SEGUNDOS E VÍRGULA
MOV     A,@R0           ; ORIENTAÇÃO DA LATITUDE
MOV     @R1,A          ; ARMAZENA ORIENTAÇÃO DA LATITUDE
INC     R1
MOV     @R1,#','        ; ARMAZENA ','
INC     R1
INC     R0
INC     R0             ; PULA ',' E CENTENA DA LONGITUDE (NÃO INTERESSA)
ACALL   DEZENA
MOV     @R1,A          ; ARMAZENA GRAUS DA LONGITUDE
INC     R1
INC     R0
ACALL   DEZENA
MOV     @R1,A          ; ARMAZENA MINUTOS DA LONGITUDE
INC     R1
INC     R0
INC     R0             ; PULA O PONTO "04753.863"
ACALL   DEZENA
MOV     @R1,A          ; ARMAZENA SEGUNDOS DA LONGITUDE

```

```

INC      R1
MOV     @R1,#'      ; ARMAZENA ","
INC     R1
INC     R0
INC     R0
INC     R0          ; PULA MILHAR DOS SEGUNDOS E VÍRGULA
MOV     A,@R0      ; ORIENTAÇÃO DA LONGITUDE
MOV     @R1,A      ; ARMAZENA ORIENTAÇÃO DA LONGITUDE
INC     R1
MOV     @R1,#'      ; ARMAZENA ","
INC     R1
INC     R0
INC     R0          ; PULA ','
ACALL   DEZENA
MOV     @R1,A      ; ARMAZENA VELOCIDADE DA EMBARCAÇÃO
INC     R1
MOV     @R1,#**     ; DELIMITADOR DO CHECKSUM
INC     R1
MOV     @R1,#**     ; DELIMITADOR QUE MARCA O CHECKSUM
INC     R1

MOV     A,#00H     ; ZERA ACUMULADOR
MOV     R0,#PRIMEIRO+11 ; 1o. CARACTERE APÓS '$'
MOV     CONT,#5    ; CARACTERES MENOS O '$' NO BANCO 2/3
MEM7:   XRL     A,@R0 ; XOR ENTRE END. MEM. ANTER. E POST.
INC     R0
DJNZ   CONT,MEM7

MOV     R0,#SEGUNDO ; CONTINUAÇÃO
MEM8:   CJNE   @R0,#**,MEM9 ; DESVIA SE NAO CHEGOU FIM DE STRING '**'
CLR     C
INC     R0
CJNE   @R0,#**,MEM9A ; DESVIA SE NAO CHEGOU FIM DE STRING '**'
SJMP   MEM10       ; FIM DE STRING

MEM9A:  CLR     C
DEC     R0          ; SÓ ACERTA O PONTEIRO
MEM9:   CLR     C
XRL     A,@R0      ; CALCULA XOR
INC     R0
MEM10:  SJMP   MEM8      ; VOLTA PRA INICIO DE CALCULO
CLR     C
MOV     @R1,A      ; ARMAZENA CHECKSUM CALCULADO NA MEMÓRIA
INC     R1
MOV     @R1,#%'    ; INDICA FIM DO PACOTE
INC     R1
MOV     @R1,#%'    ; REAFIRMA FIM DO PACOTE, POIS CARACTERES E NÚMEROS MISTURAM-SE NA STRING
INC     R1

MOV     A,#6EH     ; CÁLCULO DE LIMPEZA DE LIXO DA MEMORIA
SUBB   A,R1
MOV     CONT,A
MEM11:  MOV     @R1,#00
INC     R1
DJNZ   CONT,MEM11
RET

```

```

;=====
; CANAL SERIAL IMPLEMENTADO:
;=====

```

```

S2RECD:
        JB      F_TMR0,FMSERI      ; ANALISA DEADLOCK DO SISTEMA POR TIMER0

        JB      RXD2,S2RECD        ; AGUARDA START BIT
MEM7:   MOV     AUX1,#STEMP2        ; AUX1 = STEMP2, PARA AGUARDAR MEIO PERÍODO
        DJNZ   AUX1,SR1            ; DA TAXA DE TRANSMISSÃO DE 4800 BPS
        JB      RXD2,S2RECD        ; VERIFICA SE START BIT É VÁLIDO
        MOV     AUX2,#8            ; 8 BITS A SEREM RECEBIDOS
S2RD:   MOV     AUX1,#STEMP        ; AUX1 = STEMP2, PARA AGUARDAR O PERÍODO
SR2:    DJNZ   AUX1,SR2            ; DA TAXA DE TRANSMISSÃO DE 4800 BPS
        MOV     C,RXD2             ; CARRY = RXD
        RRC     A                  ; GUARDA BIT RECEBIDO EM A
        DJNZ   AUX2,S2RD          ; VERIFICA SE OS 8 BITS FORAM RECEBIDOS
        MOV     SER2DR,A           ; SER2DR = DADO RECEBIDO
        MOV     AUX1,#STEMP        ;
FIMBYT: DJNZ   AUX1,FIMBYT         ; VAI PARA O MEIO DO STOPBIT
FMSERI: RET                       ; FIM DA ROTINA

```

```

;=====
; ENVIO DE PACOTE DE DADOS PARA SIST. GERENC.
;=====

```

```

ENVPACOTE:
MEM1:   MOV     CONT,#16           ; 1A. PARTE DO PACOTE
        MOV     R0,#PRIMEIRO      ; POSIÇÃO DE MEMÓRIA DO 1o. CARAC. DE NMEA

        MOV     A,@R0             ; VALOR DO CARACTER DE NMEA EM ACC
        MOV     SBUF,A            ; MANDA PARA SISTEMA DE GERENCIAMENTO
        ACALL   TEMP002           ; TEMPO DE 20ms
        INC     R0
        DJNZ   CONT,PACO1        ; DESVIA SE FIM DA 1a. PARTE DO PACOTE

        MOV     CONT,#17         ; RESTO DO PACOTE
        MOV     R0,#SEGUNDO      ; 2a. POSIÇÃO DE MEMÓRIA

        MOV     A,@R0             ; VALOR DO CARACTER DE NMEA EM ACC
        MOV     SBUF,A            ; MANDA PARA SISTEMA DE GERENCIAMENTO
        ACALL   TEMP002           ; TEMPO DE 20ms
        INC     R0
        DJNZ   CONT,PACO2        ; DESVIA SE FIM DE NMEA

```

```

RET
=====
; RECEBE PACOTE DE DADOS DO SIST. DE GERENCIAMENTO
=====
RECGER:
    SETB    REN           ; HABILITA RECEPÇÃO DE DADOS
    MOV     CINT,#5      ; CONTADOR DE BYTES DA INT. GER -> POS
    MOV     R0,#PRIMEIRO ; $ALMH ( (ATUAR,LEITURA,EMERG,PARAR),LEME,MANCHE,CHECKSUM)

    CLR     RXSER        ; NAO RECEBEU DADO
    SETB    REN         ; HABILITA RECEPÇÃO DE DADOS

ESPERA2:
    JNB     RXSER,ESPERA2 ; ESPERA A RECEPÇÃO DE DADOS DE GER
    MOV     @R0,SBUF     ; CARACTER RECEBIDO
    INC     R0
    CLR     RXSER
    DJNZ   CINT,ESPERA2
    CLR     REN         ; DESABILITA RECEPÇÃO DE DADOS
    RET

=====
; ANALISA O CHECKSUM QUE VEIO JUNTO COM O PACOTE RECEBIDO DE GER
=====
CHKGER:
    MOV     A,#00H      ; ZERA ACUMULADOR
    MOV     R0,#PRIMEIRO
    INC     R0
    XRL    A,@R0       ; PULA '$'
    INC     R0
    XRL    A,@R0       ; AÇÃO A SER TRATADA
    INC     R0
    XRL    A,@R0       ; LEME
    INC     R0
    XRL    A,@R0       ; MANCHE
    INC     R0
    MOV     AUX,@R0    ; CHECKSUM REAL

    CJNE   A,AUX,ECHKG ; CORROMPIDO !!!
    MOV     A,#!
    ACALL  TEMP002     ; ACK DE SUCESSO
    MOV     SBUF,A
    SETB   F_ACKGER   ; ENVIA PARA GERENCIAMENTO
    SJMP   FIMCHKGER  ; SETA FLAG DE SUCESSO

ECHKG:
    MOV     A,#?'
    ACALL  TEMP002     ; ACK DE ERRO
    MOV     SBUF,A
    CLR    F_ACKGER   ; ENVIA PARA GERENCIAMENTO
    CLR    F_ACKGER   ; LIMPA FLAG - ERRO NO PACOTE

FIMCHKGER:
    RET
=====
; DEFINIR ATUAÇÃO NA EMBARCAÇÃO
=====
DEFATUA:
    MOV     R0,#PRIMEIRO
    INC     R0
    ; PULA CARACTER '$'

LER:
    CLR     C
    CJNE   @R0,#'L',RES ; SE NAO FOR LEITURA TENTA RESETAR
    CLR     C

    INC     R0
    MOV     A,#97H
    ACALL  ADCONV      ; SE FOR SÓ LEITURA DA NAV. SEM ATUAR
    MOV     @R0,A      ; PALAVRA DE CONTROLE - CH0 (LEME)
    ; REALIZA A CONVERSÃO
    ; COLOCA VALOR REAL DE LEME NA MEMÓRIA

    INC     R0
    MOV     A,#0DFH
    ACALL  ADCONV      ; PALAVRA DE CONTROLE - CH1 (MANCHE)
    MOV     @R0,A      ; REALIZA A CONVERSÃO
    ; COLOCA VALOR REAL DE MANCHE NA MEMÓRIA

    SJMP   CHKNAV

RES:
    CJNE   @R0,#'R',PARAR ; SE NAO RESETAR TENTA PARAR
    CLR     C
    ACALL  RESBOAT     ; COLOCAR O HARDWARE EM POSIÇÃO INICIAL DE NAVEGAÇÃO
    SJMP   CHKNAV

PARAR:
    CLR     C
    CJNE   @R0,#'S',LIGAR ; SE NAO FOR PARAR ENTAO TENTA LIGAR MOTOR
    SETB    SWE        ; ATIVA COMANDO AUTOMÁTICO
    ACALL  TEMP002

    SETB   STOPE      ; DESLIGA MOTOR DE POPA
    ACALL  TEMP1      ; TEMPO DE 1s
    ACALL  TEMP1
    CLR    STOPE      ; PÁRA DE TENTAR DESLIGAR MOTOR DE POPA
    ACALL  TEMP002
    CLR    SWE        ; DESATIVA COMANDO AUTOMÁTICO

    INC     R0
    MOV     A,#97H
    ACALL  ADCONV      ; PALAVRA DE CONTROLE - CH0 (LEME)
    MOV     @R0,A      ; REALIZA A CONVERSÃO
    INC     R0
    MOV     A,#0DFH
    ACALL  ADCONV      ; PALAVRA DE CONTROLE - CH1 (MANCHE)
    MOV     @R0,A      ; REALIZA A CONVERSÃO

```

	MOV	@R0,A	; COLOCA VALOR REAL DE MANCHE NA MEMÓRIA
	SJMP	CHKNAV	
LIGAR:	CLR	C	
	CJNE	@R0,#1',ATUAR	; SE NAO FOR LIGAR ENTAO É ATUAR
	SETB	SWE	; ATIVA COMANDO AUTOMÁTICO
	ACALL	TEMP002	
	SETB	STARTE	; LIGA MOTOR DE POPA
	ACALL	TEMP1	; TEMPO DE 1s
	CLR	STARTE	; PÁRA DE TENTAR DESLIGAR MOTOR DE POPA
	ACALL	TEMP002	
	CLR	SWE	; DESATIVA COMANDO AUTOMÁTICO
	INC	R0	
	MOV	A,#97H	; PALAVRA DE CONTROLE - CH0 (LEME)
	ACALL	ADCONV	; REALIZA A CONVERSÃO
	MOV	@R0,A	; COLOCA VALOR REAL DE LEME NA MEMÓRIA
	INC	R0	
	MOV	A,#0DFH	; PALAVRA DE CONTROLE - CH1 (MANCHE)
	ACALL	ADCONV	; REALIZA A CONVERSÃO
	MOV	@R0,A	; COLOCA VALOR REAL DE MANCHE NA MEMÓRIA
	SJMP	CHKNAV	
ATUAR:	CLR	C	; ATUA SOBRE A EMBARCAÇÃO ('A' - ATUAR OU 'V' - PILOTO VIRTUAL)
	INC	R0	
	MOV	A,@R0	
	ACALL	ACVALLEM	; ACERTA VALOR DE LEME
	ACALL	XLEME	; ATUA SOBRE O LEME
	MOV	A,#97H	
	ACALL	ADCONV	
	MOV	@R0,A	; ARMAZENA VALOR REAL DE LEME
	INC	R0	
	MOV	A,@R0	
	ACALL	ACVALMAN	; ACERTA VALOR DE MANCHE
	ACALL	XMANCHE	; ATUA SOBRE O MANCHE
	MOV	A,#0DFH	
	ACALL	ADCONV	
	MOV	@R0,A	; ARMAZENA VALOR REAL DE MANCHE
CHKNAV:	MOV	A,#00H	
	MOV	R0,#PRIMEIRO	; 1a. POSIÇÃO DE MEMÓRIA
	INC	R0	; PULA '\$'
	XRL	A,@R0	; XOR COM 1a. CARACTER INDICADOR DA AÇÃO
	INC	R0	
	XRL	A,@R0	; XOR COM LEME
	INC	R0	
	XRL	A,@R0	; XOR COM MANCHE
	INC	R0	
	MOV	@R0,A	; COLOCA CHECKSUM NA MEMORIA
	RET		
=====			
; ACERTA O VALOR CORRETO DA POSIÇÃO DO LEME			
=====			
ACVALLEM:			
	CJNE	A,#0',TLE1	; COMPARA SE É POSIÇÃO 1 (TOTALM PRA ESQUERDA OU TOTAL. PRA FRENTE)
	MOV	A,#150	; MÁX. ESQUERDA É 185
	SJMP	TLE8	; FINALIZA
TLE1:	CJNE	A,#1',TLE2	
	MOV	A,#125	; ESQUERDA 2
	SJMP	TLE8	
TLE2:	CJNE	A,#2',TLE3	
	MOV	A,#100	; ESQUERDA 1
	SJMP	TLE8	
TLE3:	CJNE	A,#3',TLE4	
	MOV	A,#71	; POSIÇÃO CENTRAL
	SJMP	TLE8	
TLE4:	CJNE	A,#4',TLE5	
	MOV	A,#58	; DIREITA 1
	SJMP	TLE8	
TLE5:	CJNE	A,#5',TLE6	
	MOV	A,#45	; DIREITA 2
	SJMP	TLE8	
TLE6:	CJNE	A,#6',TLE8	
	MOV	A,#34	; MÁX DIREITA É 30
TLE8:	MOV	DESLLM,A	
	CLR	C	
	RET		
=====			
; ACERTA O VALOR CORRETO DA POSIÇÃO DO MANCHE			
=====			
ACVALMAN:			
	CJNE	A,#0',TM1	; COMPARA SE É POSIÇÃO 1 (TUDO PRA FRENTE OU TUDO PRA TRÁS)
	MOV	A,#175	; MÁX. ACELERAÇÃO É 180
	SJMP	TM8	; FINALIZA
TM1:	CJNE	A,#1',TM2	

	MOV	A,#164	; ACELERAÇÃO 2
	SJMP	TM8	
TM2:	CJNE	A,#2, TM3	
	MOV	A,#150	; ACELERAÇÃO 1
	SJMP	TM8	
TM3:	CJNE	A,#3, TM4	; POSIÇÃO CENTRAL
	MOV	A,#130	
	SJMP	TM8	
TM4:	CJNE	A,#4, TM5	
	MOV	A,#110	; REVERSO 1
	SJMP	TM8	
TM5:	CJNE	A,#5, TM6	
	MOV	A,#93	; REVERSO 2
	SJMP	TM8	
TM6:	CJNE	A,#6, TM8	
	MOV	A,#80	; MÁX. REVERSÃO É 75
TM8:	MOV	DESLMAN,A	
	CLR	C	
	RET		

=====

; MUDANÇA DE DIREÇÃO DO LEME

=====

XLEME:	SETB	SWE	; ATIVA COMANDO AUTOMÁTICO
	CLR	C	; LIMPA CARRY
	MOV	A,#97H	; PALAVRA DE CONTROLE - CH0 (LEME)
	ACALL	ADCONV	; REALIZA A CONVERSÃO
	SUBB	A,DESLLEM	; POSIÇÃO A SER IDA
	JC	YESQ	; MOTOR À ESQUERDA (POS. ATUAL > POS. DESEJADA)
	SETB	SDIRE	; LIGA MOTOR PARA DIREITA
	ACALL	TEMP002	
LOPDIR:	CLR	C	; LIMPA CARRY
	ACALL	TEMP002	; 20ms DE TEMPO
	MOV	A,#97H	; PALAVRA DE CONTROLE - CH0 (LEME)
	ACALL	ADCONV	; REALIZA A CONVERSÃO
	SUBB	A,DESLLEM	; TESTA SE CHEGOU NA POSIÇÃO DESEJADA
	JNC	LOPDIR	; AINDA NÃO CHEGOU RECALCULA POSIÇÃO
	CLR	SDIRE	; DESLIGA MOTOR PARA DIREITA
	ACALL	TEMP002	
	SJMP	FIMXLEME	; VÁ PRA FIM DE ROTINA
YESQ:	SETB	SDIRE	; LIGA MOTOR PARA DIREITA
	ACALL	TEMP002	
	SETB	DIRE	; INVERTE MOTOR PARA GIRAR À ESQUERDA
	ACALL	TEMP002	
LOPESQ:	CLR	C	; LIMPA CARRY
	ACALL	TEMP002	; 20ms DE TEMPO
	MOV	A,#97H	; PALAVRA DE CONTROLE - CH0 (LEME)
	ACALL	ADCONV	; REALIZA A CONVERSÃO
	SUBB	A,DESLLEM	; VER SE ESTOURA CARRY
	JC	LOPESQ	; AINDA NÃO CHEGOU AO MEIO - RECALCULA POSIÇÃO
	CLR	SDIRE	; DESLIGA MOTOR PARA DIREITA
	ACALL	TEMP002	
	CLR	DIRE	; DESLIGA MOTOR DE INVERSÃO
	ACALL	TEMP002	
FIMXLEME:	CLR	C	
	CLR	SWE	; DESATIVA COMANDO AUTOMÁTICO
	RET		

=====

; MUDANÇA DE DIREÇÃO DO MANCHE

=====

XMANCHE:	SETB	SWE	; ATIVA COMANDO AUTOMÁTICO
	CLR	C	; LIMPA CARRY
	MOV	A,#0DFH	; PALAVRA DE CONTROLE - CH1 (MANCHE)
	ACALL	ADCONV	; REALIZA A CONVERSÃO
	SUBB	A,DESLMAN	; POSIÇÃO A SER IDA
	JC	YACEL	; (POS. ATUAL < POS. DESEJADA)
	SETB	SACELE	; LIGA MOTOR DE ACELERAÇÃO
	ACALL	TEMP002	
LOPACEL:	CLR	C	; LIMPA CARRY
	ACALL	TEMP002	; 20ms DE TEMPO
	MOV	A,#0DFH	; PALAVRA DE CONTROLE - CH1 (MANCHE)
	ACALL	ADCONV	; REALIZA A CONVERSÃO
	SUBB	A,DESLMAN	; TESTA SE CHEGOU NA POSIÇÃO DESEJADA
	JNC	LOPACEL	; AINDA NÃO CHEGOU RECALCULA POSIÇÃO
	CLR	SACELE	; DESLIGA MOTOR ACELERAÇÃO
	ACALL	TEMP002	
	SJMP	FIMXMAN	; VÁ PRA FIM DE ROTINA
YACEL:	SETB	SACELE	; LIGA MOTOR DE ACELERAÇÃO
	ACALL	TEMP002	

```

      SETB   ACELE           ; INVERTE MOTOR PARA DESACELERAR
      ACALL  TEMP002
LOPDESAC:
      CLR    C               ; LIMPA CARRY
      ACALL  TEMP002         ; 20ms DE TEMPO
      MOV    A,#0DFH        ; PALAVRA DE CONTROLE - CH1 (MANCHE)
      ACALL  ADCONV         ; REALIZA A CONVERSÃO

      SUBB   A,DESLMAN      ; VER SE ESTOURA CARRY
      JC    LOPDESAC        ; AINDA NÃO CHEGOU - RECALCULA POSIÇÃO
      CLR    SACELE         ; DESLIGA MOTOR PARA ACELERAÇÃO
      ACALL  TEMP002
      CLR    ACELE          ; DESLIGA MOTOR DE INVERSÃO
      ACALL  TEMP002

FIMXMAN:
      CLR    C               ;
      CLR    SWE            ; DESATIVA COMANDO AUTOMÁTICO
      RET

;=====
; RESETA E CALIBRA TODOS O HARDWARE DA EMBARCAÇÃO
;=====
RESBOAT:
      MOV    R0,#12H
      MOV    A,@R0
      ACALL  ACVALLEM      ; ACERTA VALOR DE LEME
      ACALL  XLEME         ; ATUA SOBRE O LEME

      MOV    A,#97H
      ACALL  ADCONV
      MOV    R0,#12H
      MOV    @R0,A        ; ARMAZENA VALOR REAL DE LEME

      MOV    R0,#13H
      MOV    A,@R0
      ACALL  ACVALMAN     ; ACERTA VALOR DE MANCHE
      ACALL  XMANCHE      ; ATUA SOBRE O MANCHE

      MOV    A,#0DFH
      ACALL  ADCONV
      MOV    R0,#13H
      MOV    @R0,A        ; ARMAZENA VALOR REAL DE MANCHE

      RET

;=====
;#####
;# ROTINA DE INICIALIZAÇÃO: #
;#####

INICIO:
      CLR    SWE           ; PLACA DE RELÉS -> DESABILITADA
      CLR    DIRE          ; M. DIR. -> DESLIGADO
      CLR    SDIRE         ; M. ESQ. -> DESLIGADO
      CLR    ACELE        ; M. POPA FR -> DESLIGADO
      CLR    SACELE       ; M. POPA TR -> DESLIGADO
      CLR    STARTE       ; PARTIDA M. POPA -> DESATIVADA
      CLR    STOPE        ; PARADA M. POPA -> DESATIVADA
      SETB   CS           ; CONVERSOR A/D -> DESABILITADO
      CLR    DIN           ; BIT A ENVIAR AO ADC -> 0
      CLR    DCCLK        ; NÍVEL DO CLOCK DO ADC -> 0
      CLR    LED          ; LED DE INDICAÇÃO -> APAGADO
      CLR    F_TMR0       ; FLAG TIMER0
      CLR    F_CHECK      ; FLAG CHECKSUM GPS
      CLR    F_ACKGER     ; FLAG ACK PACOTE VINDO DE GER

      MOV    IE,#90H      ; INTERRUPÇÃO SERIAL HABILITADA
      MOV    TMOD,#20H    ; TIMERS EM MODO 2
      MOV    TCON,#40H    ; TIMER 1 LIGADO
      MOV    TH1,#0FDH    ; TH1 E TL1 COM VALOR DE 253 PARA GERAR
      MOV    TL1,#0FDH    ; TAXA DE TRANSMISSÃO DE 9600 BPS
      MOV    SCON,#40H    ; CANAL SERIAL NO MODO 1

      ACALL  SERINI       ; INICIALIZAÇÃO DO CANAL SERIAL
      ACALL  BUSINI       ; INICIALIZAÇÃO DA BÚSSOLA

;#####
;# ROTINA PRINCIPAL: #
;#####

PRINCIPAL:
      SETB   LED           ; ACENDE O LED
      ACALL  TEMP2         ; ESPERA 2s
      CLR    LED           ; APAGA O LED

      MOV    R0,#PRIMEIRO ; POSICIONAR EMBARCAÇÃO COM LEME E MANCHE AO MEIO
      INC    R0            ; PULA CHARACTER '$'
      INC    R0            ; PULA ESTADO
      MOV    @R0,#'3'     ; LEME AO MEIO
      INC    R0            ;
      MOV    @R0,#'3'     ; MANCHE AO MEIO
      ACALL  RESBOAT

      CLR    RXSER        ; NAO RECEBEU DADO
      SETB   REN          ; HABILITA RECEPÇÃO DE DADOS

PRIPRI:
      CLR    C             ;
      ACALL  RECGER       ; RECEBE PACOTE DE DADOS DO SIST. DE GERENCIAMENTO

```

```
ACALL TEMP002
ACALL CHKGER ; ANALISA SE PACOTE VÁLIDO E ENVIA ACK PRA GERENCIAMENTO

JB F_ACKGER,INIACAO ; PULA SE ACK COM SUCESSO
SJMP PRIPRI ; VOLTA E RECEBE NOVAMENTE PACOTE DE DADOS

INIACAO:
ACALL DEFATUA ; ATUA SOBRE A EMBARCAÇÃO
MOV A,11H ; POSIÇÃO DE MEMÓRIA QUE INDICA A ATUAÇÃO PEDIDA PELO GERENCIAMENTO
CJNE A,#'I',LESTOP ; DESVIA SE NAO FOR LIGAR MOTOR
SJMP ACKPG

LESTOP:
CJNE A,#'S',LEMERG ; DESVIA SE NAO DESLIGAR MOTOR
SJMP ACKPG

LEMERG:
CJNE A,#'V',LEIBUSS ; DESVIA SE NAO FOR PILOTO VIRTUAL
SJMP ACKPG

LEIBUSS:
ACALL POSBUS ; ROTINA DE LEITURA DA BÚSSOLA
MOV A,11H ; POSIÇÃO DE MEMÓRIA 11H
CJNE A,#'A',LEGPS ; DESVIA SE NAO FOR ATUAÇÃO -> LEITURA DA COMPLETA DE TELEMETRIA
SJMP ACKPG

LEGPS:
ACALL POSGPS ; LEITURA DE TODOS OS SENSORES
; ROTINA DE CAPTURA DO NMEA DO GPS

ACKPG:
ACALL TEMP002
ACALL ENVPACOTE ; ENVIA DADOS PARA SIST. GERENC.
SJMP PRIPRI ; REPETE PROCEDIMENTO

#####
;# FIM DO PROGRAMA: #
#####

END
```


Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)