Ministério da
Ciência e Tecnologia

INPE-16634-PUD/216

SISTEMA DE RASTREAMENTO DE EMBARCAÇÕES DE PESCA POR SATÉLITES BRASILEIROS COM CRIPTOGRAFIA DE DADOS

André Barros Cardoso da Silva

Registro do documento original:
<http://urlib.net/sid.inpe.br/mtc-m19@80/2009/12.10.18.50>

INPE
São José dos Campos
2009

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3945-6911/6923
Fax: (012) 3945-6919
E-mail: pubtc@sid.inpe.br

“O verdadeiro discípulo é aquele que supera o mestre.
Do contrário, ainda estaríamos na Idade da Pedra”.

Aristóteles.

Dedico àqueles aos quais cada resposta é uma outra pergunta.

AGRADECIMENTOS

Ao Instituto Nacional de Pesquisas Espaciais (INPE) pela oportunidade oferecida através do Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA). À Fundação de Ciência, Aplicações e Tecnologia Espaciais (FUNCATE) pelo custeio das ferramentas e módulos utilizados. À Universidade de Taubaté (UNITAU) pelo incentivo oferecido através do Programa de Iniciação Científica (PIC/UNITAU) nos anos de 2008 e 2009, que motivaram o estudo e desenvolvimento deste projeto. Aos orientadores WILTON NEY DO AMARAL PEREIRA e WILSON YAMAGUTI pelo conhecimento e dedicação; e a todos aqueles que se colocaram disponíveis, direta e indiretamente, para auxiliar na conclusão deste trabalho.

RESUMO

O rastreamento de embarcações de pesca ao longo da costa marítima brasileira é um antigo sonho do Brasil. Com uma magnitude estimada em cinqüenta mil unidades, o Brasil possuía dados estatísticos ineficientes devido ao uso de rastreadores ser considerado, até então, um recurso opcional. Em virtude destas necessidades, em 2006 foi criado o Programa Nacional Rastreamento de Embarcações Pesqueiras por Satélite (PREPS), sob responsabilidade da Secretaria Especial de Aquicultura e Pesca da Presidência da República (SEAP/PR). O PREPS tem por objetivo monitorar embarcações de pesca com mais de 15m e/ou 50t de arqueação, estabelecendo a estas o uso obrigatório de rastreadores a bordo – lei vigorada desde Outubro/2008. Visando contribuir com os objetivos do PREPS, este trabalho propõe um sistema de rastreamento utilizando os satélites do Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA) – sistema composto por satélites desenvolvidos e operados pelo Instituto Nacional de Pesquisas Espaciais (INPE). A utilização de satélites nacionais neste segmento oferece considerável redução dos custos dos serviços de rastreamento, favorecendo assim as pequenas cooperativas de pesca. As balizas do SBCDA possuem um localizador GPS composto basicamente por um receptor GPS, um microcontrolador e um transmissor em UHF. O receptor GPS Trimble *Lassen iQ* realiza a aquisição dos dados de posição geográfica, encaminhando-os para o microcontrolador Microchip PIC18F4550, via protocolo NMEA-0183. Então, este microcontrolador gera um campo de 160 bits de acordo com o formato de mensagem *Header 0*, aplicando sobre este o algoritmo criptográfico AES (*Advanced Encryption Standard*). Após a cifragem, os dados de posição geográfica são transmitidos aos satélites do SBCDA através do transmissor ELTA HAL-2 (*High Accuracy Locator*), operante em 401,620MHz. O AES é um algoritmo simétrico de bloco baseado em permutações de *bytes* completos que permite flexibilidade na escolha dos tamanhos da chave simétrica e dos blocos de mensagem: 128, 192, 256 bits. A criptografia é um recurso inovador de aplicação potencial da baliza do SBCDA, capaz de oferecer ao usuário segurança nos dados transmitidos por suas embarcações. Aprimoramentos futuros como

cifragem em blocos menores de mensagem e algoritmos de codificação/decodificação (p.ex. *Turbo Codes*) são previstos.

ABSTRACT

The fishing vessel monitoring across the Brazilian maritime coast is an old dream of Brazil. By having fifty thousand units estimated magnitude, Brazil used to have an inefficient statistical data as far as the using of beacons was considered an optional resource. According to these needs, the National Program of Vessel Monitoring by Satellite (PREPS) was created in 2006 under responsibility of the Special Secretary of Aquaculture and Fishing of the Brazilian Government (SEAP/PR). The PREPS mainly aims the monitoring of fishing vessels above 15m and/or 50t gross weight, establishing the obligatory using of locators onboard – law into operation in October/2008. In order to contribute with PREPS goals, this work presents a GPS locator with data ciphering as an application of the Brazilian Environmental Data Collection System (SBCDA) – composed by satellites developed and managed by the National Institute for Space Research (INPE). The use of national satellites for this purpose helps small fishing co-operatives by offering reduced tracking service costs. The SBCDA beacons have an internal GPS locator basically composed by a GPS receiver, a microcontroller and an UHF transmitter. A Trimble Lassen iQ GPS receiver collects the geographic position data, forwarding them to a Microchip PIC18F4550 microcontroller based on NMEA-0183 protocol. Hence, this microcontroller generates a 160-bit field according to the Header 0 data format, applying the Advanced Encryption Standard (AES) cipher algorithm on it. After ciphered, the geographic position data are transmitted to the SBCDA satellites through an ELTA HAL-2 (High Accuracy Locator) transmitter, operating in 401.620MHz. The AES is a block cipher algorithm based on full byte permutations considering the flexibility of choosing the input data and symmetric cipher key block lengths: 128, 192, 256 bits. Ciphering is an innovative resource of potential application of the SBCDA beacon, being able to provide security to the vessels transmitted data. Future improvements such as ciphering on smaller data block lengths and coding/decoding algorithms (i.e. Turbo Codes) are foreseen.

LISTA DE FIGURAS

Pág.

1. Exemplo de PCD com diversos sensores acoplados....09
2. Aplicações das PCDs no Brasil (esq.) e sua evolução (dir.) 09
3. Segmento espacial do SBCDA e suas órbitas....11
4. Satélites de Coleta de Dados – SCD-1 (esq.) e SCD-2 (dir.)....11
5. Satélite CBERS-2B 12
6. Passagem diária dos satélites do SBCDA sobre a estação de Cuiabá/MT 13
7.1. PCDs enviando dados aos satélites do SBCDA 14
7.2. Satélites retransmitindo os dados para as estações terrenas do INPE 15
7.3. Estações terrenas encaminhando os dados para o CMCD 15
8. Banda de recepção dos satélites do SBCDA....16
9. Desvio Doppler causado na passagem do satélite pela PCD....18
10. A constelação GPS....20
11. Os três segmentos do GPS....21
12. Triangulação realizada por um conjunto de satélites do GPS....22
13. Plataformas de coleta de dados cadastradas no sistema Argos 23
14. Testes de recepção de embarcações de pesca utilizando o SBCDA 25
15. Sistema de rastreamento de embarcações de pesca utilizando o SBCDA....33
16. Receptor GPS Trimble *Lassen iQ*....35
17. *Starter kit* Trimble com o receptor GPS *Lassen iQ* acoplado....36
18. Antena do receptor GPS Trimble *Lassen iQ*....37
19. Diagrama de pinos do microcontrolador PIC18F4550 38
20. *Kit* completo do transmissor UHF ELTA HAL-2 40
21. Antena Synergetics QFH 14A-N 42
22. Bateria UNIPOWER UP1250....43
23. Protocolos de interface utilizados no localizador GPS....44
24. Processo de cifragem AES (Rijndael)....55
25. Tabela-S....57
26. Processo de decifragem AES (Rijndael)....64
27. Tabela-Si....71
28. Dados recebidos após o processo de cifragem AES (Rijndael)....80
29. Dados recebidos após o processo de decifragem AES (Rijndael)....80
30. Baliza do sistema Argos para rastreamento de embarcações....81
31. Balizas disponíveis para o rastreamento de embarcações de pesca 82
32. Protótipo da baliza do SBCDA....83
33. Diagrama de pinos do microcontrolador PIC24FJ64GA002 85
34. Transmissores estrangeiros para rastreamento de animais 88
35. Cristal externo de 32,768kHz acoplado ao PIC18F4550 90
36. Sistema de coordenadas do WGS84....100
37. Latitude (esq.) e longitude (dir.) da Terra....101
38. Arquitetura interna do PIC18F4550....104
39. Registrador TXSTA: controle e *status* da transmissão....105

40. Registrador RCSTA: controle e *status* da recepção.....105
41. Diagrama em blocos do teste da USART e UART.....108
42. Dados de posição geográfica enviados diretamente pelo receptor GPS.....109
43. Dados de posição geográfica processados pelo PIC18F4550.....109
44. ICD2br conectado à aplicação.....112

LISTA DE TABELAS

Pág.

1. Barramento de programação do transmissor UHF ELTA HAL-2 39
2. Interface de comunicação do protocolo NMEA-0183 45
3. Formatos de mensagem do protocolo NMEA-0183 46
4. Primeira posição fixa (absoluta) 48
5. Segunda, terceira e quarta posições fixas 48
6. Períodos de aquisição da posição geográfica 49
7. Número de iterações do algoritmo AES 53
8. Matriz dos blocos de mensagem 54
9. Matriz dos blocos de chave simétrica 54
10. Número de rotações do processo de cifragem AES 58
11. Número de rotações do processo de decifragem AES 69
12. Comprimento real da latitude em função da latitude 102
13. Comprimento real da longitude em função da latitude 102
14. Funções de *delay* da UART 106
15. Pinagem do *ICD2br* 112

LISTA DE QUADROS

Pág.

1. Primeira mensagem enviada pelo transmissor HAL-2 50
2. Programação da Adição da Chave (cifragem) 56
3. Programação da Substituição na Tabela-S 57
4. Programação da Permutação de Linhas (cifragem) 59
5. Programação da Permutação de Colunas (cifragem) 61
6. Programação do Cálculo de Chave (cifragem) 63
7. Programação da Adição da Chave (decifragem) 65
8. Programação da Permutação de Colunas (decifragem) 67
9. Programação da Permutação de Linhas (decifragem) 70
10. Programação da Substituição na Tabela-Si 71
11. Programação do Cálculo de Chave (decifragem) 73
12. Exemplo de cálculo para as funções de *delay* da UART 106
13. Algoritmo para controle da USART e UART 107
14. Exemplo de configuração da interrupção externa para o terminal RB0 111

LISTA DE ABREVIATURAS E SIGLAS

AES - *Advanced Encryption Standard*.

ANA - Agência Nacional de Águas.

ASCII - *American Standard Code for Information Interchange*.

CAST - *Chinese Academy of Space Technology*.

CBERS - *China-Brazil Earth Resources Satellite*.
CMCD - Centro de Missão de Coleta de Dados.
CNES - *Centre National d'Études Spatiales*.
CPTEC - Centro de Previsão de Tempo e Estudos Climáticos.
CRC - *Cyclic Redundancy Check*.
DES - *Data Encryption Standard*.
DGPS - *Differential Global Positioning System*.
EUSART - *Enhanced Universal Synchronous Asynchronous Receiver Transmitter*.
FTP - *File Transfer Protocol*.
GPS - *Global Positioning System*.
HDOP - *Horizontal Dilution of Precision*.
IBAMA - Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis.
ID - *Identification*.
INPE - Instituto Nacional de Pesquisas Espaciais.
MECB - Missão Espacial Completa Brasileira.
NMEA - *National Marine Electronics Association*.
NOAA - *National Oceanic and Atmospheric Administration*.
NSA - *National Security Agency*.
PCD - Plataforma de Coleta de Dados.
PCI - Placa de Circuito Impresso.
PDOP - *Position Dilution of Precision*.
PLL - Phase-locked loop
PREPS - Programa Nacional de Rastreamento de Embarcações Pesqueiras por Satélite.
RTCM - *Real Time Correction Messages*.
SBCDA - Sistema Brasileiro de Coleta de Dados Ambientais.
SCD - Satélite de Coleta de Dados.
SEAP/PR - Secretaria Especial de Aqüicultura e Pesca da Presidência da República.
SIVAM - Sistema de Vigilância da Amazônia.
TAIP - *Trimble ASCII Interface Protocol*.
TSIP - *Trimble Standard Interface Protocol*.
UART - *Universal Asynchronous Receiver Transmitter*.
UHF - *Ultra High Frequency*.
USART - *Universal Synchronous Asynchronous Receiver Transmitter*.
UTC - *Universal Time Code*.

VDOP - *Vertical Dilution of Precision*.
WDT - *Watchdog Timer*.
WGS - *World Geodetic System*.

SUMÁRIO

Pág.

1 INTRODUÇÃO 1
1.1 O problema 2
1.2 Objetivo 3
1.3 Delimitação do estudo 3
1.4 Relevância do estudo 4
1.5 Organização do trabalho 4

2 REVISÃO DA LITERATURA 5
2.1 Histórico da criptografia 5
2.2 O Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA) 6
2.2.1 As plataformas de coleta de dados 7
2.2.2 Os satélites do SBCDA 8
2.2.3 Visão geral do SBCDA 11
2.2.4 Localização das plataformas de coleta de dados 13
2.3 O Sistema de Posicionamento Global (GPS) 15
2.4 O sistema de rastreamento de embarcações de pesca 19

3 MÉTODO 22

4 RESULTADOS 27
4.1 O sistema proposto 27
4.1.1 Definição dos componentes e módulos 28
4.1.1.1 Receptor GPS Trimble *Lassen iQ* 29
4.1.1.2 Antena do receptor GPS Trimble *Lassen iQ* 30
4.1.1.3 Microcontrolador Microchip PIC18F4550 31
4.1.1.4 Transmissor UHJ ELTA HAL-2 (*High Accuracy Locator*) 32
4.1.1.5 Antena Synergetics QFH 14A-N 34
4.1.1.6 Bateria Unipower UP1250 36

4.1.2 Os protocolos de mensagem 37
4.1.2.1 O protocolo NMEA-0183 37
4.1.2.2 O formato de mensagem *Header 0* 40
4.1.2.3 O protocolo *Intel Hex Format* 42
4.1.3 O algoritmo de criptografia AES (Rijndael) 44
4.1.3.1 Processo de cifragem AES (Rijndael) 46
4.1.3.1.1 Adição da chave 46
4.1.3.1.2 Substituição na Tabela-S 47
4.1.3.1.3 Permutação de linhas 49
4.1.3.1.4 Permutação de colunas 50
4.1.3.1.5 Cálculo da chave 52
4.1.3.2 Processo de decifragem AES (Rijndael) 54
4.1.3.2.1 Adição da chave 54
4.1.3.2.2 Permutação de colunas 55
4.1.3.2.3 Permutação de linhas 59
4.1.3.2.4 Substituição na Tabela-Si 60
4.1.3.2.5 Cálculo da chave 61
4.1.3.3 Validação do algoritmo criptográfico AES (Rijndael) 63
4.1.3.3.1 Teste do algoritmo de cifragem AES (Rijndael) 64
4.1.3.3.2 Teste do algoritmo de decifragem AES (Rijndael) 66
4.1.3.3.3 Recepção dos dados após os testes realizados 68
4.1.4 A baliza do SBCDA 69
4.1.4.1 Características da baliza do SBCDA 71
4.1.5 Adaptação do *software* multiplataforma 72

5 DISCUSSÕES E PERSPECTIVAS 74
5.1 A baliza do SBCDA 74
5.2 Demais aplicações do Localizador GPS 74
5.3 O *software* multiplataforma 75
5.4 A fonte de sincronismo do localizador GPS 76

5.5 Criptografia em blocos menores de mensagem.....77
5.6 O algoritmo de codificação/decodificação.....78

6 CONCLUSÃO 79

7 REFERÊNCIAS.....80

APÊNDICE A – O SISTEMA GEODÉSICO MUNDIAL *WGS84* 84

APÊNDICE B – FUNCIONAMENTO DO MICROCONTROLADOR PIC18F4550 87

APÊNDICE C – O ICD2BR (*IN-CIRCUIT DEBUGGER*) 96

APÊNDICE D – TABELA DE CONVERSÃO ASCII – HEXADECIMAL.....97

APÊNDICE E – FLUXOGRAMA DO LOCALIZADOR GPS 98

APÊNDICE F – CIRCUITO ELÉTRICO DO LOCALIZADOR GPS.....100

APÊNDICE G – PCI DO LOCALIZADOR GPS E *LAYOUT* 101

1 INTRODUÇÃO

Os sistemas de telecomunicações vêm sofrendo profundas alterações com o advento da comunicação digital e de outras tecnologias, como comunicações via satélite, localização GPS, entre outros [1]. O Instituto Nacional de Pesquisas Espaciais (INPE), através do Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA), oferece a oportunidade de realizar experimentos de comunicação envolvendo diversas tecnologias de comunicação digital com ênfase em aplicações de coleta de dados ambientais. Este sistema utiliza em seu segmento espacial um conjunto de satélites, desenvolvidos e operados pelo INPE. Os serviços prestados por este sistema são relacionados à coleta de dados ambientais por Plataformas de Coleta de Dados (PCDs), que utilizam os satélites como meio de comunicação para transmissão dos dados até as estações terrenas de recepção. Atualmente, mais de setecentas PCDs foram instaladas no sistema atendendo a cerca de cem organizações usuárias, como a Agência Nacional de Águas (ANA), SIVAM, e diversos núcleos estaduais de meteorologia [2]. Porém, novas demandas de coleta de dados necessitam adquirir as posições geográficas de uma dada plataforma e ao mesmo tempo garantir a proteção dos dados contra acesso não permitido – como por exemplo, sistemas de rastreamento.

O rastreamento de embarcações de pesca ao longo da costa marítima brasileira, sempre foi um antigo sonho do Brasil em ingressar no setor pesqueiro modernizado. Com uma magnitude estimada em cinquenta mil embarcações, o país possuía dados estatísticos ineficientes devido ao rastreamento de embarcações ser considerado, até então, um recurso opcional. Deste modo, em 15 de Setembro de 2006 foi publicada no Diário Oficial da União a criação do Programa Nacional de Rastreamento de Embarcações Pesqueiras por Satélite (PREPS), com o objetivo de modernizar o monitoramento de embarcações de pesca no Brasil através da implantação de rastreadores nas embarcações, oferecendo cobertura nacional através do

uso de satélites [3]. A proposta deste programa é de financiar desde grandes empresas até pequenos empreendimentos, com o objetivo de monitorar as embarcações de pesca e controlar as operações da frota. De responsabilidade da Secretaria Especial de Aqüicultura e Pesca da Presidência da República (SEAP/PR), do Instituto Brasileiro de Meio Ambiente e dos Recursos Naturais Renováveis (IBAMA) e da Marinha do Brasil, o PREPS permite melhor fiscalizar: área de atuação das embarcações, rotas, profundidade, entre outros. Contudo, o PREPS estabelece uso obrigatório de rastreadores a toda embarcação de pesca com mais de 15m e/ou 50t de arqueação, lei em vigor desde Outubro de 2008 [4]. Estima-se que o país possua cerca de quatro mil embarcações pesqueiras que se enquadram nesta lei. O uso de rastreadores a estas embarcações, primeiramente, oferece maior segurança as suas respectivas tripulações. Além disso, os donos de embarcações podem acompanhar o trajeto de seus barcos munidos de uma senha de acesso, permitindo um controle mais eficiente sobre sua frota.

Em virtude desta necessidade, o convênio estabelecido entre a Universidade de Taubaté (UNITAU) e o INPE permitiu o desenvolvimento de uma baliza do SBCDA, onde um rastreador oferece o serviço de localização geográfica através de um receptor do Sistema de Posicionamento Global (GPS), aplicando sobre os dados de posição e tempo o algoritmo de criptografia AES (*Advanced Encryption Standard*). Recursos similares de localização e segurança de acesso são disponíveis em sistemas estrangeiros, contudo em termos nacionais, o uso de criptografia nos dados transmitidos por uma PCD tem característica inovadora.

Criado pelos pesquisadores belgas Vincent Rijmen e Joan Daemen, o AES (Rijndael) é um algoritmo simétrico de bloco baseado em permutações de bytes completos que permite flexibilidade na escolha dos tamanhos da chave simétrica e dos blocos de mensagem: 128, 192, 256 bits [5]. Visto como um dos métodos criptográficos mais fortes do mundo, o AES não possui chaves fracas em sua concepção. Os ataques impraticáveis a este algoritmo fizeram

com que o próprio governo norte-americano o utilizasse para a proteção de seus dados ultra-secretos.

1.1 O problema

O monitoramento de embarcações de pesca é um recurso já disponível em sistemas estrangeiros, por exemplo o sistema francês Argos. Este sistema é composto atualmente por uma constelação de cinco satélites, atendendo a diversas aplicações em coleta de dados, como: meteorologia, hidrologia, estudos da atmosfera e clima, entre outros. Porém, no Brasil a desvantagem deste sistema se refere principalmente ao elevado custo oferecido às comunidades usuárias. O mercado de embarcações de pesca brasileiro é dominado por empresas estrangeiras que utilizam sistemas estrangeiros (p.ex. sistema Argos) para comunicação, e com isso tem-se este serviço de monitoramento como um privilégio restrito somente às grandes cooperativas de pesca. O preço médio deste serviço para cada embarcação gira em torno de R$500,00/mês.

1.2 Objetivo

Em virtude do elevado custo oferecido pelos sistemas estrangeiros atuantes no campo de rastreamento de embarcações de pesca, o Instituto Nacional de Pesquisas Espaciais (INPE) objetiva contribuir com o Programa Nacional de Rastreamento de Embarcações Pesqueiras por Satélite (PREPS) através do oferecimento de uma solução nacional mais econômica por meio do Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA) – sistema composto por satélites desenvolvidos e operados pelo INPE. A utilização de satélites nacionais neste segmento oferece considerável redução dos custos dos serviços de rastreamento, favorecendo assim as pequenas cooperativas de pesca. Deste modo, este trabalho compreende o projeto completo de uma baliza para o rastreamento de embarcações de pesca, desde seu estudo de

viabilidade, até o desenvolvimento, montagem, testes em campo, e avaliação dos resultados.

1.3 Delimitação do estudo

Este trabalho se limita a apresentar uma solução nacional comercialmente compatível com os sistemas estrangeiros atuantes no mercado de rastreamento de embarcações de pesca. Embora a cifragem dos dados seja um recurso potencial e inovador da baliza do SBCDA, o algoritmo criptográfico AES (Rijndael) é abordado excluso enfoque matemático. Ainda observa-se a necessidade de implementação de um algoritmo de codificação, visando reduzir os danos causados na recepção de bits errados em mensagens criptografadas. Aprimoramentos da baliza referentes ao consumo de energia e minimização de *hardware* ainda oferecem futuras oportunidades para trabalhos de graduação e/ou iniciação científica. A implementação de um algoritmo de codificação (p.ex. *Turbo Codes*) ainda propicia o desenvolvimento de futuras teses de mestrado.

1.4 Relevância do estudo

Primeiramente, este trabalho se mostra de grande interesse do Departamento de Engenharia Elétrica, posto que permite ao aluno o desenvolvimento de novos conceitos, contato com tecnologias espaciais, e a realização de experimentos práticos – complementando assim sua formação como Engenheiro de Telecomunicações.

As tecnologias envolvidas neste sistema de rastreamento tem ajudado a elevar o nível de conhecimento sobre a frota pesqueira que atua no litoral brasileiro. Por se tratar de uma necessidade atual, esta aplicação ainda oferece a possibilidade de futuros estudos e/ou aprimoramentos por parte de alunos e professores, incentivando a pesquisa e o engajamento científico.

1.5 Organização do trabalho

O cenário do rastreamento de embarcações de pesca no Brasil e no mundo é introduzido pela Sessão 1, apresentando também os problemas e as soluções propostas por este trabalho. Conceitos preliminares relevantes são abordados na Sessão 2 com o objetivo de introduzir o leitor à criptografia e aos sistemas de comunicação por satélites (p.ex. GPS, SBCDA). O método que estabelece um conjunto de fases de desenvolvimento e testes é descrito na Sessão 3. A Sessão 4, por sua vez descreve toda a parte técnica do projeto, desde a escolha das ferramentas do localizador GPS até o desenvolvimento, montagem e testes da baliza do SBCDA. As dificuldades encontradas nestas etapas promovem uma discussão apresentada em tópicos na Sessão 5, apontando as possíveis soluções que dão continuidade ao estudo deste tema. Por fim, na Sessão 6 são analisados os resultados obtidos nos testes da baliza do SBCDA, permitindo avaliar seu desempenho em condições reais. Apêndices e arquivos anexos são disponibilizados como material de suporte ao conteúdo desta literatura.

2 REVISÃO DA LITERATURA

Um sistema de rastreamento via satélite envolve um conjunto de conceitos que devem ser previamente destacados, visando uma melhor compreensão sobre as partes envolvidas na concepção deste projeto. Deste modo, esta sessão se dedica a explorar alguns conceitos preliminares, tais como: histórico da criptografia, o Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA), o Sistema de Posicionamento Global (GPS), e por fim, o sistema de rastreamento de embarcações de pesca. Cabe a esta sessão uma abordagem superficial destes temas, considerando, em partes, os conceitos técnicos e matemáticos envolvidos.

2.1 Histórico da criptografia

Não se sabe ao certo em que ponto da história a criptografia se originou. Estima-se que tenha sido inventada junto à própria escrita. Contudo, a história tem nos mostrado que generais, reis e rainhas buscavam formas eficientes de comunicação para comandar seus exércitos e governar seus países. A importância de não revelar segredos e estratégias às forças inimigas já foram o motivo de êxito de muitas batalhas. Com isso, muitos códigos foram elaborados, e logo, muitos decifradores também.

A *esteganografia* (do grego *steganos*: "coberto"; e *graphein*: "escrever") é uma técnica bastante primitiva baseada na ocultação da mensagem, embora ainda possa ser encontrada em algumas aplicações atuais, por exemplo mensagens escondidas em imagens que trafegam pela Internet. Sua evolução nos trouxe ao atual conceito da *criptografia* (do grego *kryptos*: "escondido", "oculto"; e *graphein*: "escrever"), que agora objetiva ocultar o conteúdo da mensagem, e não mais a mensagem em si. Teoricamente, a criptografia pode ser entendida como o estudo das técnicas e princípios pelos quais uma informação pode ser transformada da sua forma original para outra ilegível, ou

aparentemente sem valor. Na prática, é um ramo especializado da teoria da informação com muitas contribuições de outros campos da matemática [6].

A criptografia moderna é basicamente formada pelo estudo dos algoritmos criptográficos que podem ser implementados em computadores, ou em sistemas embarcados através de processamento microcontrolado [7].

A criptografia eletrônica passou a ser utilizada a partir de 1974, com o primeiro algoritmo criptográfico comercial desenvolvido pela IBM, denominado *Lúcifer*. Sujeito a algumas alterações realizadas pela NSA (*National Security Agency*), este método criptográfico foi determinado como padrão em 1977, sob o nome de DES (*Data Encryption Standard*). Foi considerado muito eficaz e de difícil quebra, mas também bastante polêmico. Seus critérios de projeto não foram divulgados pelo governo norte americano. Especulava-se que algum tipo de fraqueza planejada pudesse ter sido inserida no DES, para que o governo pudesse facilmente ler mensagens cifradas por terceiros. Com as primeiras quebras sofridas em 1998, se fez necessária a escolha de um novo algoritmo padrão – o AES (*Advanced Encryption Standard*). Criado pelos pesquisadores belgas Vincent Rijmen e Joan Daemen, o método criptográfico AES (Rijndael) é um algoritmo baseado em permutações de bytes completos, permitindo flexibilidade ao usuário na escolha dos tamanhos de chave e blocos de mensagem: 128, 192, 256 bits [5]. Utiliza uma chave de criptografia simétrica (privada) sem restrições de escolha, pois até hoje não foram identificadas chaves fracas ou semi-fracas para este algoritmo [8].

2.2 O Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA)

O projeto deste sistema tem origem em 1980, com a aprovação do governo brasileiro sobre a Missão Espacial Completa Brasileira (MECB), que previa o desenvolvimento de quatro satélites, do segmento solo, e da infra-estrutura de integração e testes sob a responsabilidade do Instituto Nacional de Pesquisas

Espaciais (INPE). Tem como objetivo permitir a coleta de dados ambientais em localidades distribuídas por todo o território nacional através de Plataformas de Coleta de Dados (PCDs), e transmissão via satélite.

O sistema foi concebido de modo a atender a um número máximo de até quinhentas PCDs, embora hoje em dia possua mais de oitocentas unidades que atendem a mais de cem organizações usuárias. Dentre as principais aplicações deste sistema, destacam-se as previsões de tempo – realizadas pelo Centro de Previsão do Tempo e Estudos Climáticos (CPTEC), acompanhamento e controle de queimadas e enchentes, e estudos científicos diversos. A operacionalidade do sistema e a qualidade dos serviços prestados são fatores que justificam o investimento de mais de quinze milhões de dólares, realizado por parte dos usuários na aquisição e instalação de suas PCDs.

2.2.1 As plataformas de coleta de dados

Uma Plataforma de Coleta de Dados (PCD) é uma estação autônoma, instalada em locais remotos e/ou de difícil acesso, que tem por função coletar dados ambientais. Para isso, as PCDs são providas de sensores dos mais variados tipos: temperatura, velocidade e direção do vento, radiação solar, umidade, pressão, entre outros. Para poder operar de forma autônoma, as PCDs podem ser alimentadas por células solares e/ou baterias de longa duração, de modo a garantir uma vida útil desejável [9]. A Fig. 1 mostra um exemplo de PCD acoplada a diversos sensores:

Fig. 1. Exemplo de PCD com diversos sensores acoplados [9].

Atualmente, existem diversos tipos de PCDs destinadas as mais diversas aplicações, como: meteorologia, oceanografia, estudos químicos da atmosfera, monitoramento de bacias hidrográficas, rastreamento de animais, rastreamento de embarcações, monitoramento ambiental, entre outras. A Fig. 2 mostra a diversidade de aplicações das PCDs operantes no Brasil, junto a sua evolução desde 1993:

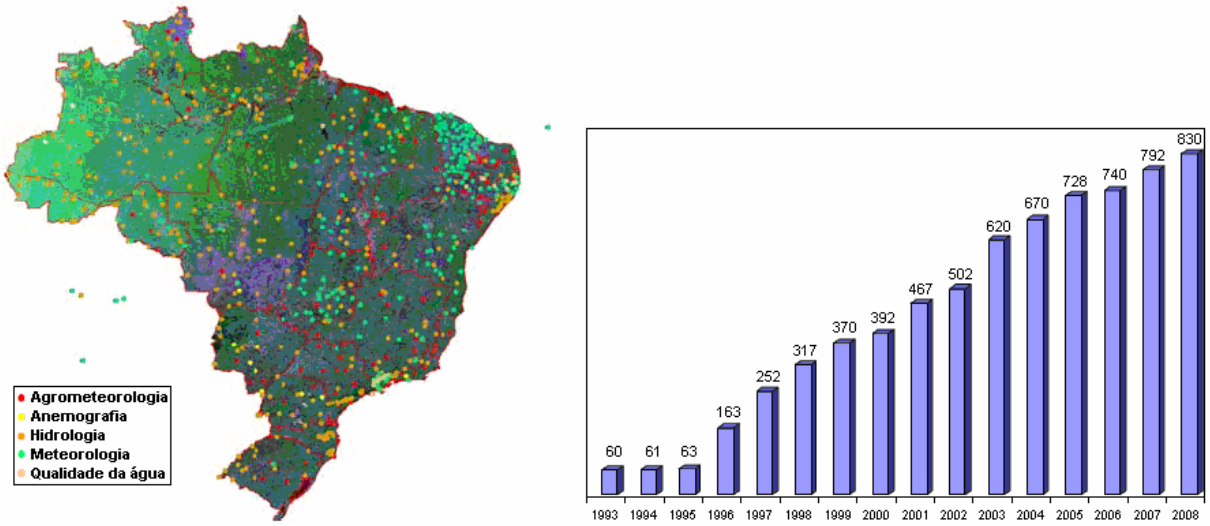


Fig. 2. Aplicações das PCDs no Brasil (esq.) e sua evolução (dir.) [10].

A Fig. 2 reflete o crescente interesse da comunidade usuária através do aumento do número de PCDs ao longo dos anos. Também nota-se que as aplicações em meteorologia e hidrologia são predominantes no SBCDA, compondo aproximadamente 86% da demanda do sistema.

Cada PCD recebe um número de identificação único no SBCDA denominado *ID*, que pode conter até 28 bits. Os dados transmitidos por estas PCDs podem assumir diferentes ciclos de transmissão (comumente a cada 90s ou 200s), podendo ser transmitidos aos satélites do SBCDA por dois canais: 401,620MHz ±3kHz e 401,650MHz ±3kHz. Esta imprecisão deve-se ao próprio oscilador local do transmissor da PCD [10].

2.2.2 Os satélites do SBCDA

A operação do SBCDA foi iniciada a partir do lançamento com sucesso do satélite SCD-1 em 1993, projetado para uma vida útil estimada em um ano, e que, surpreendentemente, completou mais de quinze anos em operação. Em 1998, com o lançamento do satélite SCD-2, o SBCDA recebeu importante reforço no seu segmento espacial garantindo maior confiabilidade e sinalizando à comunidade usuária a continuidade do sistema. Seguiu-se o lançamento do CBERS-1 em 1999, do CBERS-2 em 2003 e do CBERS-2B em 2007. O CBERS-1 operou até agosto de 2003 e o CBERS-2 operou até abril de 2005 em termos de coleta de dados [10]. Logo, o segmento espacial do SBCDA é atualmente composto por três satélites: SCD-1, SCD-2, e CBERS-2B, como mostrado na Fig. 3:

Fig. 3. Segmento espacial do SBCDA e suas órbitas [10].

Os Satélites de Coleta de Dados – SCD-1 e SCD-2, são satélites de órbita circular baixa (aproximadamente 750km de altitude), com uma inclinação de 25° em relação ao plano do Equador, permitindo cobe rtura adequada de todo o território nacional – quatorze órbitas por dia com oito passagens sobre o Brasil. São satélites de pequeno porte com dimensões da ordem de 1m³ (1m x 1m x 1m), e aproximadamente 115kg. A Fig. 4 mostra os satélites SCD-1 e SCD-2:

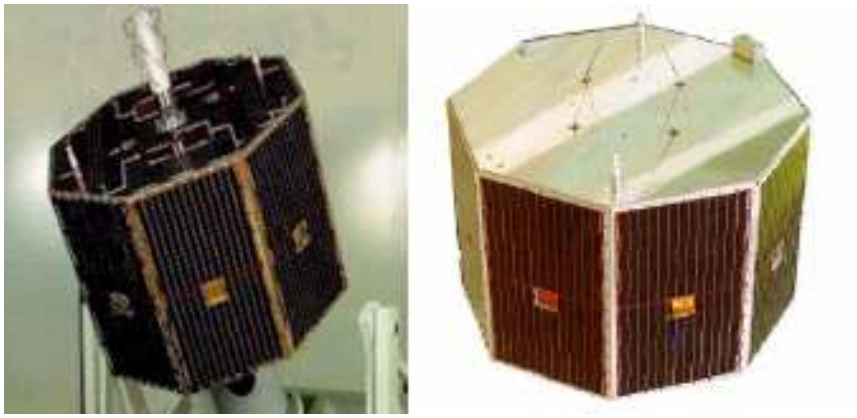


Fig. 4. Satélites de Coleta de Dados – SCD-1 (esq.) e SCD-2 (dir.).

Os sinais recebidos e processados por estes satélites atendem a uma probabilidade de sucesso superior a 95% durante suas passagens favoráveis, ou seja, aquelas as quais a PCD consegue fazer em média três transmissões durante o período de visibilidade mútua entre PCD, satélite, e estação terrena [11].

O programa CBERS (*China-Brazil Earth Resources Satellite*) foi desenvolvido através de uma parceria entre o INPE e a CAST (*Chinese Academy of Space Technology*), tendo como objetivo a criação de uma família de satélites para sensoriamento remoto. Por possuírem um *transponder* de coleta de dados a bordo, também podem atender ao SBCDA, auxiliando aos satélites SCDs.

O CBERS-2B é um satélite de órbita circular baixa (aproximadamente 778km) Sol-Síncrono, polar, com inclinação de 98,504°. Possui um corpo de dimensões (1,8 x 2,0 x 2,2)m, com um painel solar de (6,3 x 2,6)m, e aproximadamente 1450kg. A Fig. 5 mostra o satélite CBERS-2B:

Fig. 5. Satélite CBERS-2B.

O plano orbital do SCD-2 foi defasado em ascensão reta em relação ao do SCD-1 por um ângulo de 180 graus, de modo a garantir que as passagens do SCD-2 irão preencher cada período diário em que ocorrem passagens não visíveis do SCD-1 e vice-versa. Já o satélite CBERS-2B, de órbita polar, apresenta três ou quatro passagens por dia sobre a estação principal. Um sistema de coleta de dados baseado em satélites com órbitas de baixa inclinação (menor ou igual a 30°) se mostra muito adequado para o Brasil, pois, além de permitir um número grande de passagens por dia, apresenta uma cobertura satisfatória do sul do país [10]. A Fig. 6 mostra as passagens diárias dos satélites do SBCDA sobre a estação terrena do INPE em Cuiabá/MT:

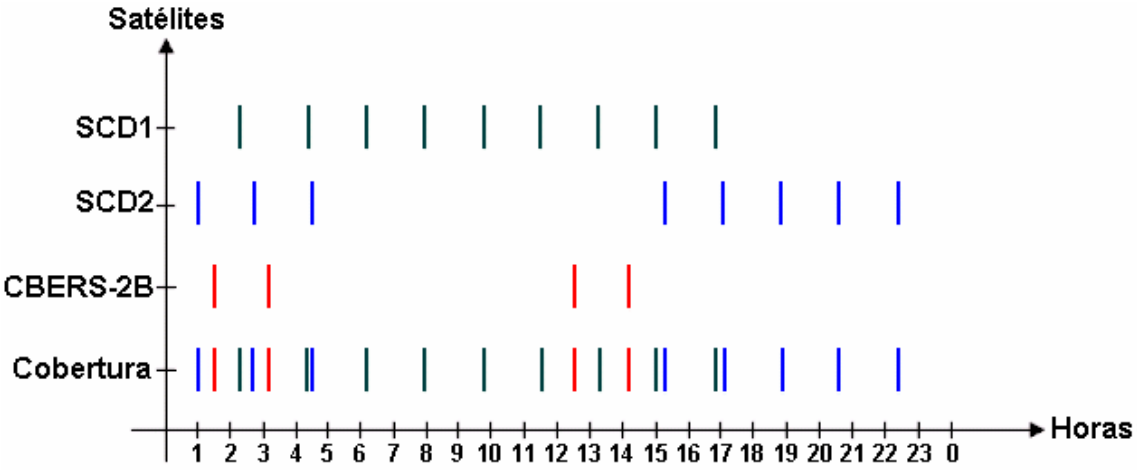


Fig. 6. Passagem diária dos satélites do SBCDA sobre a estação de Cuiabá/MT.

As vinte e uma passagens diárias dos satélites do SBCDA sobre a estação de Cuiabá/MT são mais do que suficiente para o rastreamento de embarcações de pesca, mesmo considerando as possíveis colisões de mensagens.

2.2.3 Visão geral do SBCDA

As estações terrenas do Instituto Nacional de Pesquisas Espaciais (INPE) se localizam em pontos estratégicos do país. A estação de Cuiabá/MT está

localizada no centro do país, enquanto a estação de Alcântara/MA situa-se muito próxima da linha do Equador. Projetos de novas estações estão previstos para os próximos anos, em Natal/RN e São José dos Campos/SP.

No SBCDA, os satélites funcionam como retransmissores de mensagens [12]. Cada satélite projeta uma área de visibilidade de 5000km de diâmetro sobre a superfície terrestre circular (*footprint*). Analogamente, as estações terrenas também possuem uma área de visibilidade para estes satélites. Deste modo, a retransmissão dos dados enviados pelas PCDs somente ocorrerá quando as plataformas e a estação de recepção compartilharem a mesma área de visibilidade. Satisfeita esta condição, os dados retransmitidos pelos satélites do SBCDA podem ser coletados pelas estações terrenas de Cuiabá/MT e/ou Alcântara/MA, processados e armazenados pelo Centro de Missão de Coleta de Dados (CMCD) em Cachoeira Paulista/SP, e então difundidos aos usuários através da Internet, via servidor FTP (*File Transfer Protocol*) [12]. A Fig. 7 mostra em detalhes como ocorre este processo:

Fig. 7.1. PCDs enviando dados aos satélites do SBCDA.

Fig. 7.2. Satélites retransmitindo os dados para as estações terrenas do INPE.

Fig. 7.3. Estações terrenas encaminhando os dados para o CMCD.

A Fig. 7.1 mostra as diversas PCDs transmitindo dados aos satélites do SBCDA, basicamente em dois canais: 401,620MHz ±3kHz e 401,650MHz ±3kHz. A Fig. 7.2 mostra a retransmissão destes dados às estações terrenas do INPE, em Cuiabá/MT e Alcântara/MA. Este enlace de descida se estabelece em 2267,520MHz. A Fig. 7.3 mostra os dados recebidos pelas estações terrenas sendo encaminhados ao Centro de Missão de Coleta de Dados (CMCD), em Cachoeira Paulista/SP. O envio desses dados ao usuário é feito através da Internet, em no máximo trinta minutos após a recepção [10].

A duração da janela de transmissão depende do ângulo de passagem dos satélites do SBCDA em relação às estações terrenas, podendo esta chegar a até doze minutos. Neste caso, as PCDs tem um maior aproveitamento para enviar seus dados. Porém, colisões entre sinais de PCDs diferentes podem ocorrer na recepção pelos satélites, conforme descrito na sessão a seguir.

2.2.4 Localização das plataformas de coleta de dados

Esta sessão objetiva responder a algumas perguntas chave, tais como:

a) Como compartilhar a faixa de recepção de 60 kHz em UHF entre as diversas plataformas?
b) Como considerar o efeito Doppler nos sinais recebidos?
c) Fazer o processamento a bordo ou em terra?

As PCDs dividem uma banda de 60kHz no espectro de UHF, destinada ao SBCDA. Embora haja uma diferença de 30kHz entre os dois canais de comunicação dos satélites do SBCDA, cada canal possui uma guarda de 15kHz. Deste modo, pode-se representar a banda disponível conforme mostra a Fig. 8:

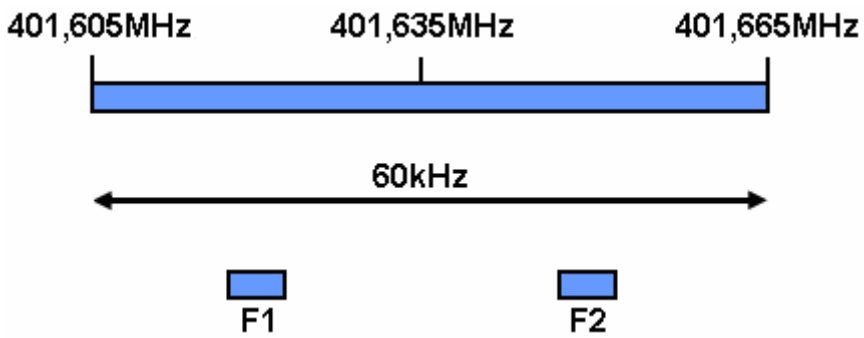


Fig. 8. Banda de recepção dos satélites do SBCDA.

Na Fig. 8, *F1* e *F2* representam os dois canais de comunicação dos satélites do SBCDA: 401,620MHz e 401,650MHz, respectivamente. Deste

modo, para uma faixa de guarda de 15kHz em *F1* e *F2*, tem-se uma banda total disponível de 60kHz (401,605MHz a 401,665MHz).

Com mais de oitocentas PCDs compartilhando apenas dois canais de comunicação, existe uma grande probabilidade de colisão entre as mensagens transmitidas por duas ou mais plataformas, posto que esta comunicação é aleatória em tempo e em frequência, e as PCDs não são interrogadas pelos satélites antes de iniciar uma transmissão. A aleatoriedade em tempo deve-se a programação dos ciclos de repetição das mensagens enviadas por cada PCD, comumente a cada 90s ou 200s. A aleatoriedade em frequência ocorre principalmente devido ao efeito Doppler, presente em sinais emitidos ou refletidos por objetos em movimento – satélites de órbita baixa se movem a uma velocidade de aproximadamente 8km/s. O cálculo do desvio Doppler pode ser realizado através da expressão 2.1 [13], permitindo localizar uma PCD mesmo que haja comunicações simultâneas:

$$\dot{\rho} = \frac{(f_r - f_t)}{f_t} \times c \qquad (2.1)$$

Onde,

$\dot{\rho} = v\cos(\theta) =$ velocidade do satélite relativa ao transmissor;

$f_r =$ frequência recebida pelo satélite;

$f_t =$ frequência de referência enviada pelo transmissor;

$(f_r - f_t) =$ desvio Doppler devido à velocidade relativa satélite-transmissor;

$c =$ velocidade de propagação da luz no vácuo (≅300km/s);

$\theta =$ ângulo entre o vetor velocidade do satélite e a posição do transmissor relativa ao satélite.

O efeito Doppler se manifesta causando um desvio positivo aparente na frequência de transmissão da PCD quando o satélite entra em seu campo de visibilidade. Por outro lado, quando o satélite se afasta da PCD o efeito

Doppler se manifesta causando um desvio negativo aparente em sua frequência de transmissão, como mostra a Fig. 9:

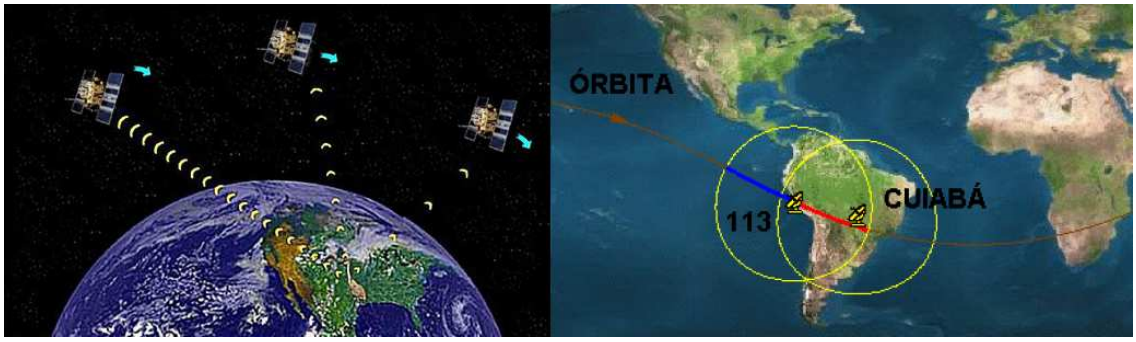


Fig. 9. Desvio Doppler causado na passagem do satélite pela PCD [13].

À direita, o trajeto azul da órbita mostra o desvio positivo na freqüência aparente do sinal, enquanto o trajeto vermelho mostra o desvio negativo. Os círculos amarelos delimitam o campo de visibilidade das estações terrenas de Cuiabá/MT e Lima (Peru).

Os satélites do SBCDA não realizam qualquer processamento a bordo, pois são dotados apenas de um *transponder* para coleta de dados. Logo, este processamento deve ser realizado em terra pelas estações terrenas, através de seus Processadores de Coleta de Dados (Procod).

2.3 O Sistema de Posicionamento Global (GPS)

Um desafio que o homem sempre teve desde os primórdios da humanidade foi determinar sua trajetória para chegar a um determinado local. Quando o homem se aventurou nos mares, a importância e a necessidade de técnicas de navegação mais exatas se tornaram evidente. Logo, foram criados os primeiros instrumentos de navegação, como: a bússola, o astrolábio, o sextante, entre outros. Com o passar do tempo, novas técnicas ainda mais precisas de posicionamento geográfico foram utilizadas (fundamentadas principalmente nos estudos de Maxwell, e nos experimentos de Hertz), dando

instrumentos de navegação, como: a bússola, o astrolábio, o sextante, entre outros. Com o passar do tempo, novas técnicas ainda mais precisas de posicionamento geográfico foram utilizadas (fundamentadas principalmente nos estudos de Maxwell, e nos experimentos de Hertz), dando origem aos primeiros sistemas de navegação: LORAN, OMEGA, ALPHA, entre outros.

Em 1960, a Marinha Americana tinha em operação o primeiro sistema de na navegação global, denominado TRANSIT. Em 1996, este foi tirado de operação devido à existência de um sistema mais moderno operado pelo Departamento de Defesa Americano (DoD), denominado GPS (*Global Positioning System*). Este sistema de navegação é capaz de oferecer a posição instantânea, bem como a velocidade e o horário de um ponto qualquer sobre a superfície terrestre ou bem próxima a ela, num referencial tridimensional.

A constelação é composta por vinte e quatro satélites operantes, posicionados de forma que em qualquer lugar do mundo e a qualquer momento existam no mínimo quatro satélites no plano horizontal do observador. Possui seis órbitas distintas ao redor da Terra a uma altitude de 20200km, distribuídos em seis planos orbitais com uma inclinação de 55° em relação ao equador, e com um período de revolução de doze horas siderais [14]. A Fig. 10 mostra uma representação da constelação GPS:

Fig. 10. A constelação GPS.

Cada satélite desta constelação possui a bordo um relógio atômico, que oferece uma base de tempo bastante precisa (da ordem de 10^{-11} s), e é freqüentemente sincronizado entre os elementos do sistema. Como os relógios atômicos são mais precisos que a própria rotação terrestre, o sistema GPS passou também a prover serviços de medição precisa de tempo.

O sistema possui três segmentos: controle, espacial e usuário. O Segmento de Controle é responsável pelo monitoramento e saúde operacional dos satélites. Trata-se de estações terrenas dispostas pelo mundo ao longo da Zona Equatorial que atuam no monitoramento de órbitas, manobras, realocação, sincronização dos relógios atômicos de bordo, e mensagens de navegação dos satélites (atualização dos dados de almanaque). O Segmento Espacial é constituído pela constelação de satélites em si. Por fim, o Segmento de Usuário é constituído pelos receptores de uso civil e militar, que recebem os sinais da constelação e calculam o posicionamento em três dimensões, velocidade e tempo [14]. A Fig. 11 mostra os três segmentos que compõem o GPS:

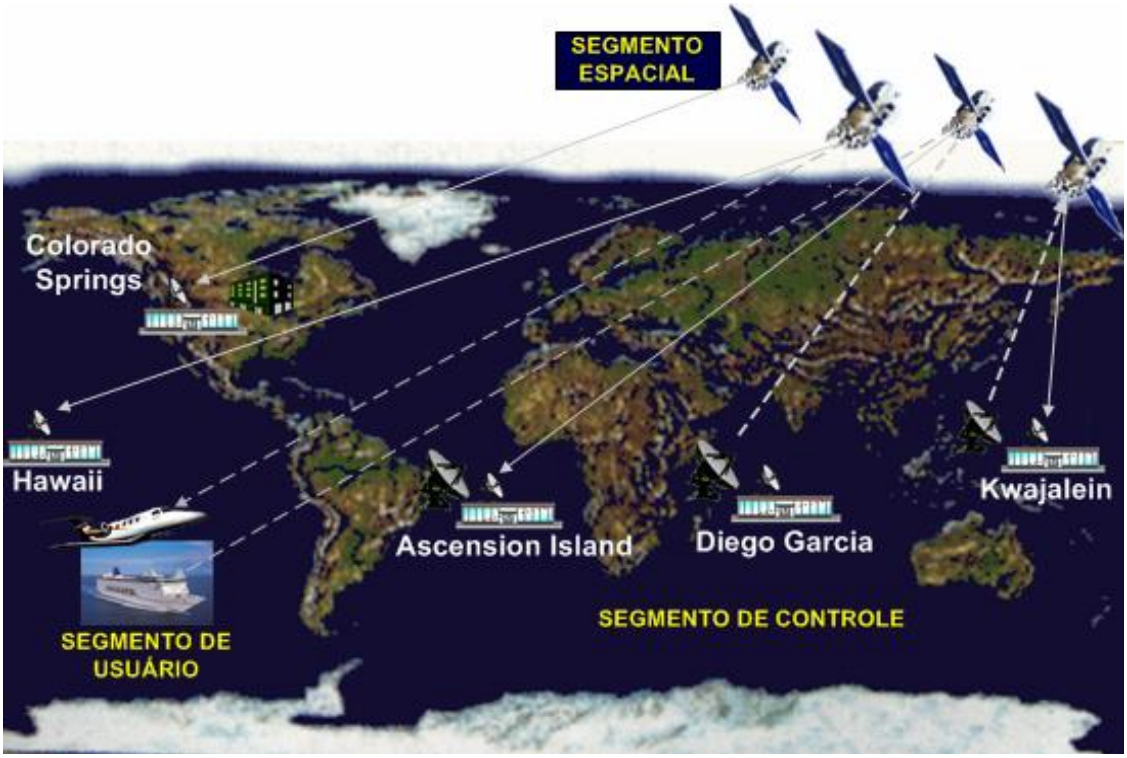


Fig. 11. Os três segmentos do GPS.

Embora controlado pelo Departamento de Defesa Americano, o GPS é livre a qualquer pessoa, organização ou governo. Desde o lançamento dos primeiros receptores no mercado, tem havido um crescente número de aplicações nos levantamentos geodésicos, geológicos, cartográficos e ambientais, induzindo a necessidade de adoção de sistemas de referências geocêntricos. Para a determinação de uma posição qualquer no Espaço ou na Terra, o GPS utiliza como referência o WGS84 (*World Geodetic System*), um sistema geodésico cuja função é representar matematicamente o planeta para cálculos de órbita e posição [15]. Maiores detalhes sobre o WGS84 podem ser encontrados no Apêndice A.

Cada satélite transmite um conjunto de dados sobre suas respectivas órbitas (efemérides), bem como a correção de horário e parâmetros de retardos atmosféricos. Este conjunto completo é denominado almanaque, e pode ser utilizado pelos receptores GPS por alguns meses sem requerer

atualização. Deste modo, os receptores em terra selecionam os dados de quais satélites utilizar, triangulando sua posição. A Fig. 12 mostra uma simulação de triangulação realizada pelo *software Google Earth*:

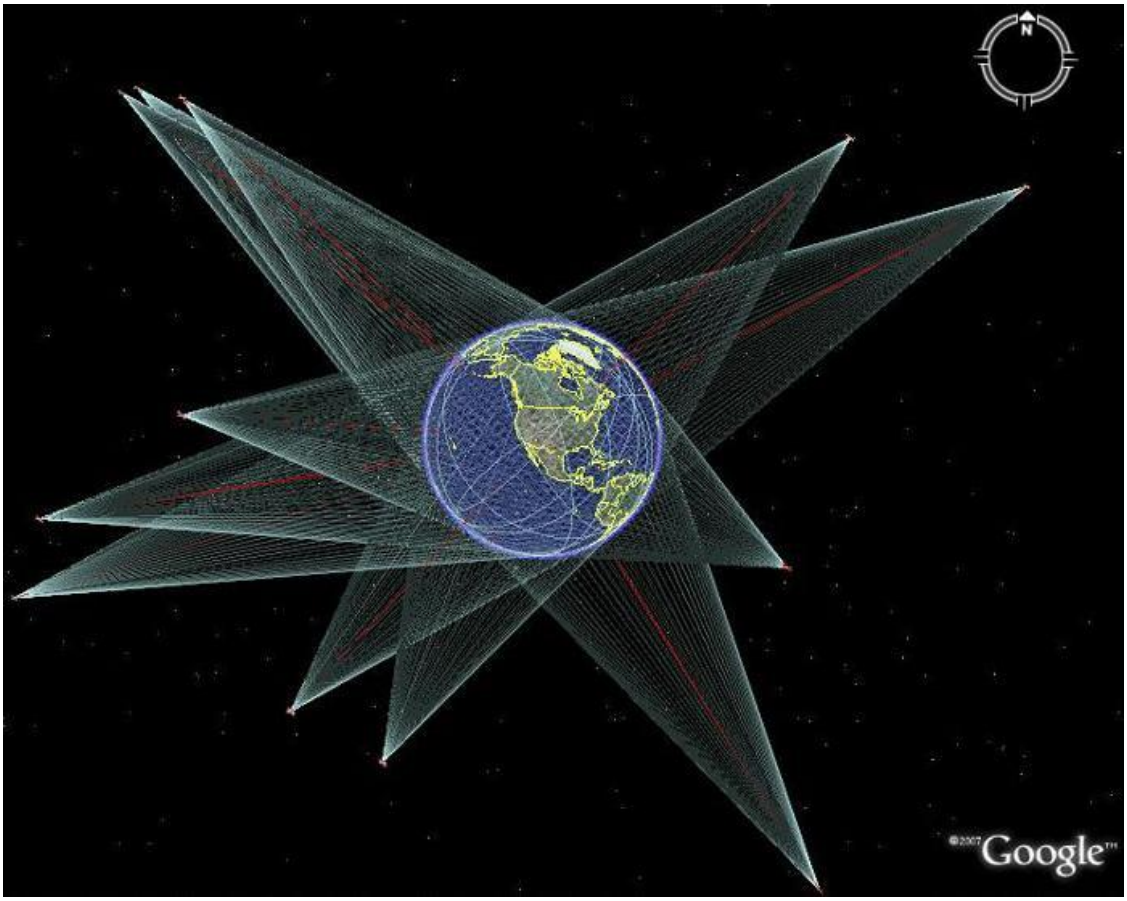


Fig. 12. Triangulação realizada por um conjunto de satélites do GPS.

O processo de triangulação é possível devido às áreas de visibilidade em comum (intersecção de *footprints*) de um conjunto de satélites do GPS. Logo, quanto mais satélites em visibilidade, melhor a precisão dos dados de posição geográfica. Por fim, estes dados são disponibilizados ao usuário final obedecendo a um protocolo de comunicação implementado pelos receptores GPS (p.ex. protocolo NMEA-0183).

2.4 O sistema de rastreamento de embarcações de pesca

O monitoramento de embarcações de pesca é um recurso já disponível em sistemas estrangeiros, que vem aderindo novas embarcações adeptas em diversos países do mundo. O sistema Argos, por exemplo, é composto atualmente por uma constelação de cinco satélites NOAAs, das famílias Argos 2 e Argos 3. A administração deste sistema é de responsabilidade da agência americana NOAA (*National Oceanic & Atmospheric Administration*), e da agência espacial francesa CNES (*Centre National d'Etudes Spatiales*), sob um acordo de cooperação. O sistema Argos é um dos grandes líderes do mercado de coleta de dados ambientais, atendendo a diversas aplicações em todo o mundo, como: meteorologia, hidrologia, estudos da atmosfera e clima, entre outros. A Fig. 13 mostra as aplicações deste sistema para o monitoramento de embarcações de pesca (triângulos), e bóias oceânicas (esferas):

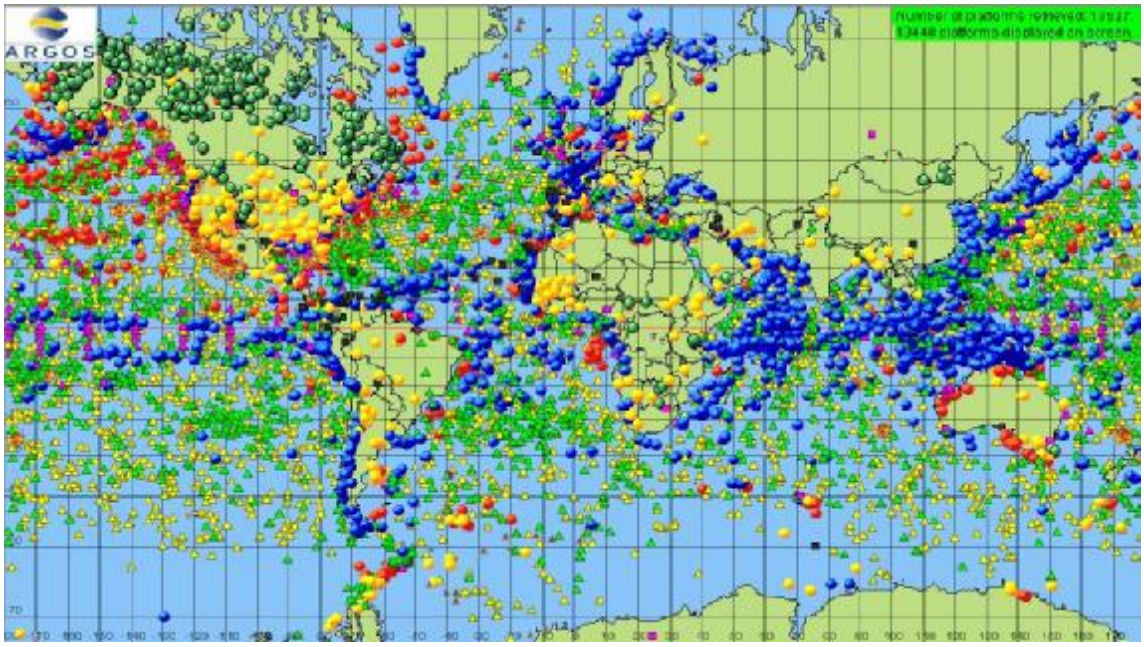


Fig. 13. Plataformas de coleta de dados cadastradas no sistema Argos [16].

Nota-se que muitos países utilizam o sistema Argos para o rastreamento de suas embarcações de pesca, como: Rússia, Peru, Japão, Estados Unidos,

Panamá, México, Venezuela, Honduras, Chile, Guatemala, entre outros. Porém, a desvantagem deste sistema se refere principalmente ao elevado custo oferecido às comunidades usuárias. No Brasil, o mercado de embarcações de pesca é dominado por empresas estrangeiras que comercializam transmissores baseados em sistemas estrangeiros (p.ex. sistema Argos), fazendo deste serviço de monitoramento um privilégio restrito somente às grandes cooperativas de pesca [12]. A OnixSat, por exemplo, é uma empresa que possui mais de vinte representantes no país, e vende seus rastreadores por aproximadamente R$2300,00. Ainda, soma-se a este valor uma taxa de utilização dos satélites estrangeiros que gira em torno de R$500,00/mês, por embarcação.

Com base nestas necessidades, o Instituto Nacional de Pesquisas Espaciais (INPE) visa contribuir com o PREPS através de um sistema de monitoramento de embarcações de pesca baseado em tecnologia nacional, considerando as restrições do Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA). Este sistema foi concebido de forma compatível ao sistema Argos, onde cada plataforma de coleta de dados possui um número de identificação único, e os sinais enviados são totalmente compatíveis em termos de freqüência de portadora, potência de transmissão, e formato de mensagem (*Header 0*). Em caso de emergência ou indisponibilidade dos satélites nacionais, os satélites NOAAs podem ser utilizados como backup dos satélites SCDs e CBERS [9]. A Fig. 14 mostra uma simulação de recepção feita pelo *software* STK (*Satellite Tool Kit*):

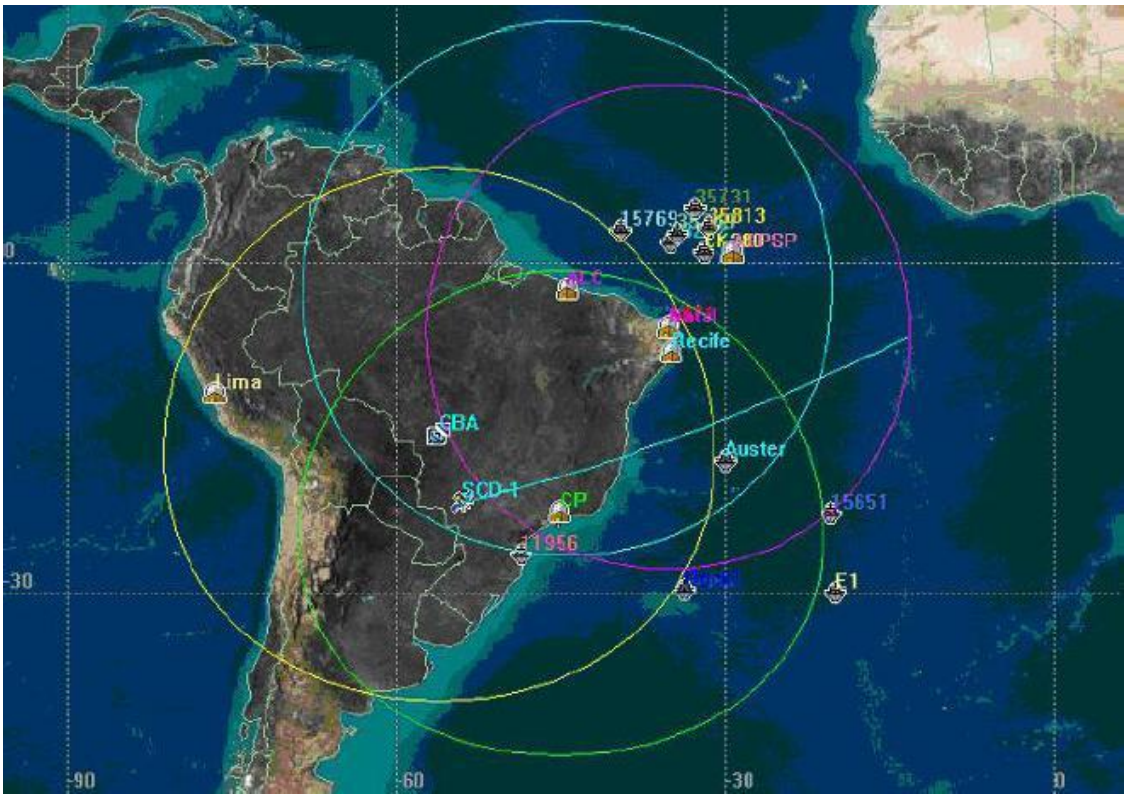


Fig. 14. Testes de recepção de embarcações de pesca utilizando o SBCDA.

Neste cenário de testes são mostradas algumas embarcações de pesca monitoradas pela SEAP, através dos satélites do SBCDA. Os dados transmitidos por estas embarcações são retransmitidos pelos satélites do SBCDA às estações terrenas de Cuiabá/MT (círculo amarelo), Alcântara/MA (círculo azul), Natal/RN (círculo roxo) e Cachoeira Paulista/SP (círculo verde).

3 MÉTODO

Por se tratar de um trabalho essencialmente de pesquisa tecnológica, a metodologia utilizada visa atender necessidades reais que nortearam a definição de objetivos e o estabelecimento de um conjunto de fases de desenvolvimento e testes. Estas fases incluem o desenvolvimento de um sistema protótipo avaliado em campo, em condições reais. Este trabalho deve responder a algumas perguntas chaves, tais como:

a) Qual o tipo de algoritmo criptográfico mais adequado, considerando as restrições dos serviços de coleta de dados oferecidos pelo Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA)?
b) Os tempos envolvidos no processamento dos dados nos processos de cifragem e decifragem são compatíveis com a aplicação desejada?
c) Considerando as diversas situações de transmissão de dados, quais os impactos na decifragem dos dados em função de possíveis erros de comunicação?
d) O protótipo de baliza construído atende às expectativas da comunidade usuária, considerando o ambiente marítimo bastante agressivo?
e) Qual o desempenho real de um sistema de rastreamento baseado neste protótipo, considerando o uso no SBCDA e o tipo de antena adotada?
f) O custo da baliza é compatível com a classe de usuários de recursos financeiros escassos (p.ex. pequenas cooperativas de pesca)?

Toda a infra-estrutura necessária para o desenvolvimento do rastreador de embarcações, como: ferramentas de desenvolvimento, módulo receptor GPS, transmissor em UHF, baterias, antenas e materiais para construção da baliza, foi disponibilizada pelo INPE, assim como a infra-estrutura necessária para a

execução dos testes em campo com o SBCDA. Deste modo, as seguintes etapas e fases foram planejadas para o alcance dos objetivos propostos:

1ª Etapa: Estudos preliminares.

Esta etapa visa o conhecimento preliminar das ferramentas utilizadas na concepção do projeto, possibilitando levantar os primeiros arranjos práticos do sistema.

- **ü** Familiarização com o Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA);
- **ü** Familiarização com o Sistema de Posicionamento Global (GPS);
- **ü** Estudos sobre os módulos receptores GPS disponíveis no mercado, junto à viabilidade de utilização oferecida por cada fornecedor;
- **ü** Estudos sobre os protocolos de dados utilizados por receptores GPS;
- **ü** Estudos sobre transmissores programáveis em UHF;
- **ü** Estudos sobre os transmissores em UHF disponíveis no mercado, junto à viabilidade de utilização oferecida por cada fornecedor;
- **ü** Estudos sobre os protocolos de dados utilizados por transmissores em UHF;
- **ü** Estudos sobre microcontroladores PIC;
- **ü** Estudos sobre as famílias de microcontroladores Microchip PIC16F e PIC18F, junto ao funcionamento de seus periféricos internos de comunicação serial (p.ex. USART e UART);
- **ü** Estudos sobre estruturas de linguagens de programação (p.ex. linguagem C).

2ª Etapa: Métodos criptográficos.

Esta etapa abrange os estudos realizados sobre diferentes algoritmos de criptografia e suas viabilidades de implementação.

- Estudos básicos sobre comunicação digital, codificação de fonte e criptografia, visando compreender a importância da segurança dos dados em uma comunicação digital;
- Conceitos básicos sobre criptografia: chave simétrica, chave assimétrica (pública), algoritmos de fluxo, algoritmo de bloco, entre outros;
- Pesquisas sobre métodos criptográficos e familiarização com seus respectivos algoritmos;
- Avaliação de performance e facilidade de implementação.

3ª Etapa: Elaboração do mapa de projeto.

Esta etapa reúne a elaboração do diagrama em blocos do localizador GPS e as ferramentas escolhidas para seu desenvolvimento. Todos os conceitos adquiridos nas etapas anteriores são utilizados nesta etapa.

- Definição das ferramentas que compõem o localizador GPS: módulo receptor GPS, microcontrolador, transmissor em UHF programável, antenas e baterias;
- Elaboração do diagrama em blocos do localizador GPS.

4ª Etapa: Desenvolvimento de software e hardware.

O conhecimento obtido nas etapas anteriores é utilizado nesta etapa para a programação do microcontrolador. Paralelamente ao desenvolvimento do *software* são adotados os modelos que definem o *hardware* do sistema, de modo a evitar desconformidades entre *software* e *hardware* ao longo do desenvolvimento do projeto.

- Implementação do protocolo de comunicação utilizado pelo receptor GPS, via USART (microcontrolador PIC);
- Implementação do protocolo de comunicação utilizado pelo transmissor em UHF, via UART (microcontrolador PIC);

- ü Implementação do método criptográfico escolhido na 2ª Etapa;
- ü Projeto da caixa de acondicionamento para os testes do localizador GPS;
- ü Desenvolvimento da placa de circuito impresso do localizador GPS.

5ª Etapa: Integração e testes do localizador GPS.

Esta etapa é responsável pela montagem do protótipo final, e os testes em campo do localizador GPS. São realizados ainda os últimos ajustes em laboratório.

- ü Montagem da caixa de acondicionamento do localizador GPS;
- ü Realização dos últimos testes de integração em laboratório, permitindo avaliar o comportamento do sistema com todos os módulos integrados;
- ü Integração dos módulos à caixa de acondicionamento;
- ü Determinação do local de testes do localizador GPS e levantamento das coordenadas geográficas do mesmo;
- ü Cumprimento do período de testes em campo aberto;
- ü Coleta dos dados e avaliação dos resultados de acordo com a precisão e coerência dos mesmos.

6ª Etapa: Estudos preliminares e elaboração das especificações da baliza do SBCDA.

Nesta etapa são realizados estudos sobre o campo de rastreamento de embarcações no Brasil e no exterior, junto ao PREPS e as instruções normativas da SEAP. Também são abordadas as balizas atualmente utilizadas por sistemas estrangeiros, permitindo estabelecer um conjunto de especificações para a elaboração da baliza do SBCDA.

7ª Etapa: Projeto preliminar da baliza.

Esta etapa é responsável pelo projeto da baliza do SBCDA, com base nos modelos previamente estudados. São determinadas as dimensões, peso e materiais, considerando seu ambiente hostil de operação.

8ª Etapa: Aprimoramento do software do localizador GPS.

O conceito de *multiplataforma* é aplicado ao *software* do localizador GPS, possibilitando seu funcionamento em plataformas de 8 e 16 bits. A implementação deste conceito traz consigo diversos benefícios, como: maior versatilidade de operação, maior capacidade de processamento, maior capacidade de armazenamento de dados, entre outros. Com isso, o localizador GPS passa a operar plenamente com as famílias de microcontroladores Microchip PIC18F, PIC24F e PIC24H.

9ª Etapa: Adaptação do software.

Nesta etapa são realizadas as modificações necessárias no *software* do localizador GPS, adequando-o às instruções normativas da SEAP. Dentre estas modificações, pode-se citar: implementação do botão de emergência, possibilidade de diferentes tipos de mensagem, período de aquisição de posição geográfica programável, entre outras.

10ª Etapa: Construção da baliza.

A baliza é o *hardware* responsável pelo acondicionamento do localizador GPS, transmissor em UHF, antenas e baterias, tendo sua construção baseada nos desenhos previamente realizados. Considera-se para esta aplicação a hipótese de uma baliza hermética, devido a sua operação em ambiente bastante agressivo e hostil.

11ª Etapa: Integração e testes da baliza do SBCDA.

Esta etapa é responsável pelos testes do localizador GPS com o *software* modificado de acordo com as instruções normativas da SEAP. Realizados os últimos testes em laboratório, o localizador GPS e os demais módulos são integrados à baliza do SBCDA, permitindo o início dos testes em campo, em condições reais. Este procedimento segue a mesma metodologia da 5ª Etapa.

12ª Etapa: Avaliação operacional e elaboração do relatório final do projeto.

Esta etapa é responsável pela validação dos dados recebidos. As mensagens enviadas pela baliza devem apresentar posições geográficas coerentes àquelas coletadas anteriormente no local de testes. Deste modo, o relatório final deste projeto se apóia nos resultados obtidos.

4 RESULTADOS

O desenvolvimento de um sistema de rastreamento via satélite aborda as etapas de um projeto completo, desde sua concepção: análise de viabilidade e estudos preliminares; até sua implementação prática: definição dos componentes e módulos, desenvolvimento, montagem, integração e testes, e por fim a análise dos resultados obtidos. Uma vez explorados todos os conceitos envolvidos na concepção deste projeto, esta sessão tem por objetivo abordar as etapas de sua implementação prática.

A infra-estrutura necessária para o desenvolvimento deste projeto (ferramentas de desenvolvimento, módulo receptor GPS, transmissor em UHF, antenas, entre outros) foi disponibilizada pelo Instituto Nacional de Pesquisas Espaciais (INPE), assim como os testes realizados em campo, com o auxílio do SBCDA.

4.1 O sistema proposto

Este trabalho propõe um sistema de rastreamento de embarcações de pesca completo, considerando as restrições do Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA). Esta aplicação considera a utilização de um localizador GPS acoplado às embarcações, como mostra a Fig. 15.

O localizador GPS pode ser dividido em três blocos principais: recepção, processamento e transmissão. O primeiro é responsável pela aquisição dos dados de posição geográfica da embarcação. Trata-se basicamente de um receptor do Sistema de Posicionamento Global (GPS) que possui uma antena ativa com amplificador de sinal interno. Os dados recebidos por este receptor são enviados a Unidade de Criptografia e Codificação de Dados, onde um microcontrolador faz o processamento dos mesmos. Esta interface é implementada com base no protocolo NMEA-0183, comumente utilizado em

aplicações de monitoramento/rastreio. Das mensagens de posição geográfica enviadas com base neste protocolo, são extraídos somente os dados imprescindíveis para a criação do campo de mensagem *Header 0*, limitado em 160 bits. Então, o microcontrolador aplica sobre este campo o algoritmo de criptografia AES (*Advanced Encryption Standard*), de modo a oferecer segurança aos dados antes da transmissão aos satélites do SBCDA. Esta transmissão, por sua vez, é realizada através de um transmissor operante em 401,620MHz – frequência de recepção dos satélites do SBCDA [17]. Os dados coletados pelas Estações Terrenas de Cuiabá (MT) e Alcântara (MA), são processados e armazenados pelo Centro de Missão de Coleta de Dados (CMCD), e então difundidos aos usuários através da Internet, via servidor FTP [12]. Por fim, o algoritmo de decifragem AES converte as mensagens processadas pelo CMCD à sua forma original, para então serem disponibilizadas ao usuário [17].

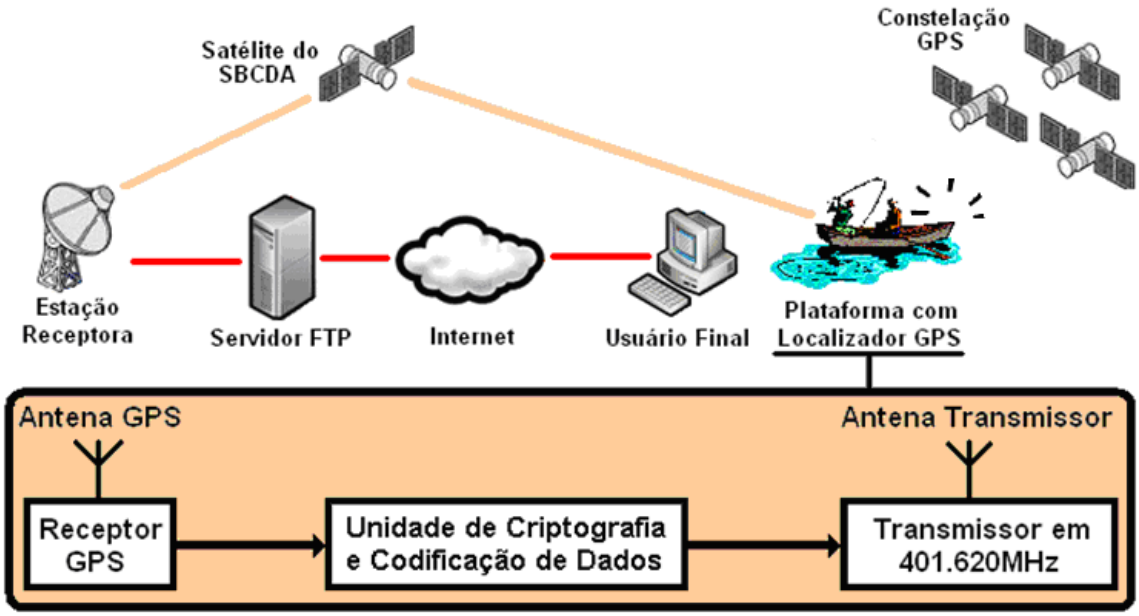


Fig. 15. Sistema de rastreamento de embarcações de pesca utilizando o SBCDA [17].

4.1.1 Definição dos componentes e módulos

As ferramentas utilizadas para a montagem da baliza do SBCDA foram escolhidas de modo a satisfazer a alguns fatores primordiais. Inicialmente optou-se por ferramentas de fácil disponibilidade no mercado, posto que o produto possa ficar comprometido em caso de indisponibilidade de material. O baixo custo e consumo também são fatores altamente desejáveis, pois torna uma ampla gama de aplicações economicamente viável, além de oferecer maior vida útil ao produto. Por fim, devem oferecer facilidade de acesso a literaturas (*datasheets*) e ferramentas de desenvolvimento, como: *starter kits*, programadores, *debuggers*, entre outros.

4.1.1.1 Receptor GPS Trimble *Lassen IQ*

Este modelo foi exclusivamente projetado pela Trimble para atender a aplicações em monitoramento/rastreio, combinando dimensões reduzidas e baixo consumo – fato o qual pode ser utilizado em celulares, *pagers*, câmeras digitais, entre outros. Dentre suas principais características podemos citar:

- Dimensões: 26mm x 26mm x 6mm;
- Consumo: <90mW (27mA) @ 3.3V;
- Protocolos suportados: NMEA-0183, TSIP, TAIP e DGPS (RTCM);
- MTBF: 60 anos;
- Resolução:
    - Horizontal: <8m (90%);
    - Altitude: <16m (90%);
    - PPS: 50ns.

A Fig. 16 mostra o receptor GPS Trimble *Lassen iQ*:

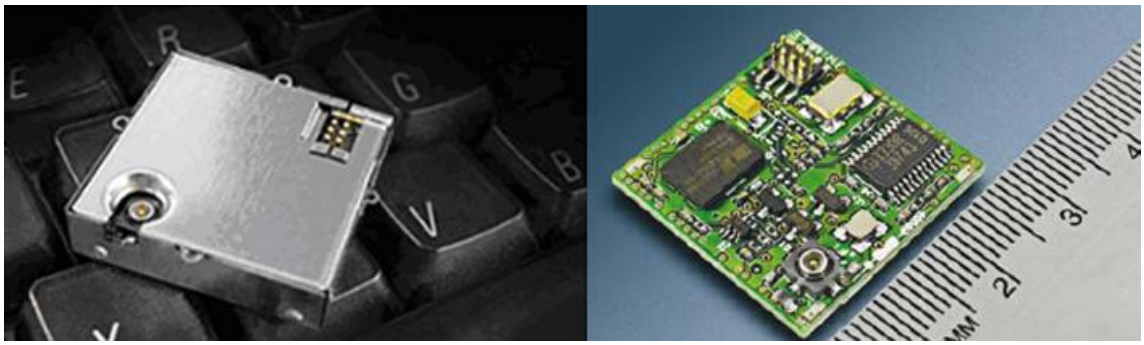


Fig. 16. Receptor GPS Trimble *Lassen iQ* [18].

Com um tempo médio entre falhas (MTBF) de sessenta anos, este receptor GPS é um dos modelos mais confiáveis do mercado [18]. Dentre os diversos protocolos que este modelo oferece, o NMEA-0183 foi escolhido para este projeto. A programação completa deste modelo pode realizada através do *software Trimble GPS Monitor*, disponibilizado pelo próprio fabricante.

O desenvolvimento em laboratório ainda contou com a ajuda de um *starter kit* da Trimble, baseado neste modelo de receptor. Este *kit* possui duas interfaces do tipo *DB9*, possibilitando a comunicação nos protocolos citados anteriormente. Os dados são disponibilizados em padrão RS-232, possibilitando a comunicação direta com um computador ou outro terminal de dados. A Fig. 17 mostra o *starter kit* da Trimble junto ao receptor GPS *Lassen iQ*, acoplados ao microcontrolador.

Informações mais detalhadas sobre este receptor GPS e seu *software* de programação podem ser encontradas nas referências bibliográficas utilizadas [18] e arquivos anexos.

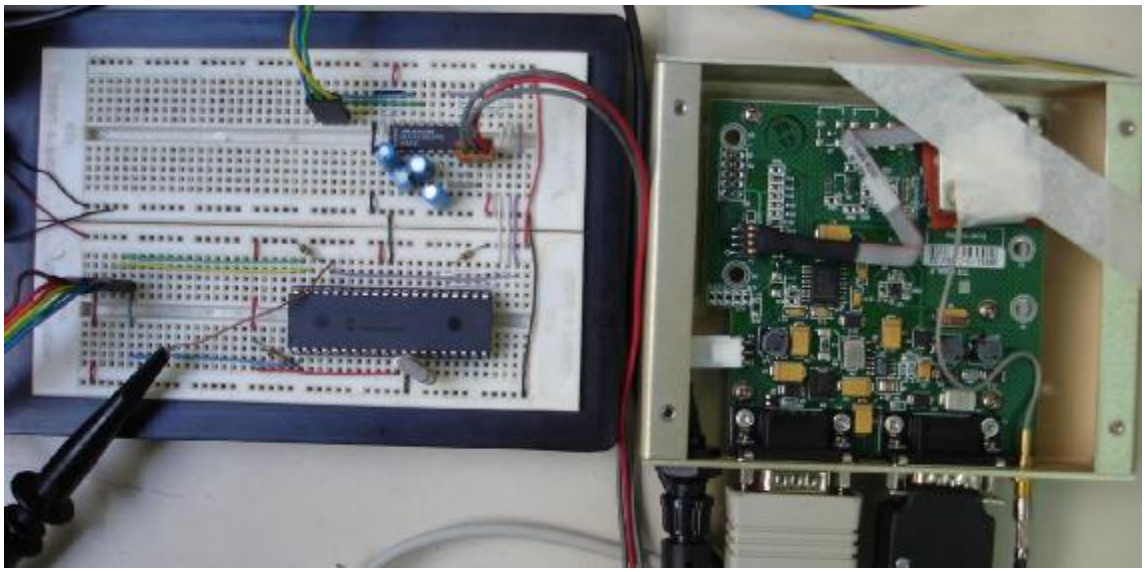


Fig. 17. *Starter kit* Trimble com o receptor GPS *Lassen iQ* acoplado.

4.1.1.2 Antena do receptor GPS Trimble *Lassen IQ*

O receptor GPS *Lassen iQ* possui um conjunto de antenas compatíveis, disponibilizadas pelo próprio fabricante. Para o caso particular deste projeto, foi escolhida uma antena ativa de montagem externa, cujas especificações seguem:

- Tensão de alimentação: 3V;
- Dimensões: 42mm x 50,5mm x 13,8mm;
- Fixação magnética;
- Comprimento do cabo: 5m.

A Fig. 18 mostra a antena deste receptor GPS:

Fig. 18. Antena do receptor GPS Trimble *Lassen iQ*.

4.1.1.3 Microcontrolador Microchip PIC18F4550

A Microchip é atualmente a grande líder no mercado de microcontroladores, com uma vasta linha de dispositivos que se adéquam as mais variadas aplicações em eletrônica/telecomunicações. A família PIC18F se destaca por oferecer maior velocidade de processamento, memória de programa linear, ampla gama de periféricos internos, e um *set* de instruções otimizado para programação em linguagem C. O modelo PIC18F4550 ainda foi projetado com tecnologia *nanoWatt*, capaz de combinar uma altíssima performance com um mínimo consumo. A Fig. 19 mostra o diagrama de pinos deste microcontrolador:

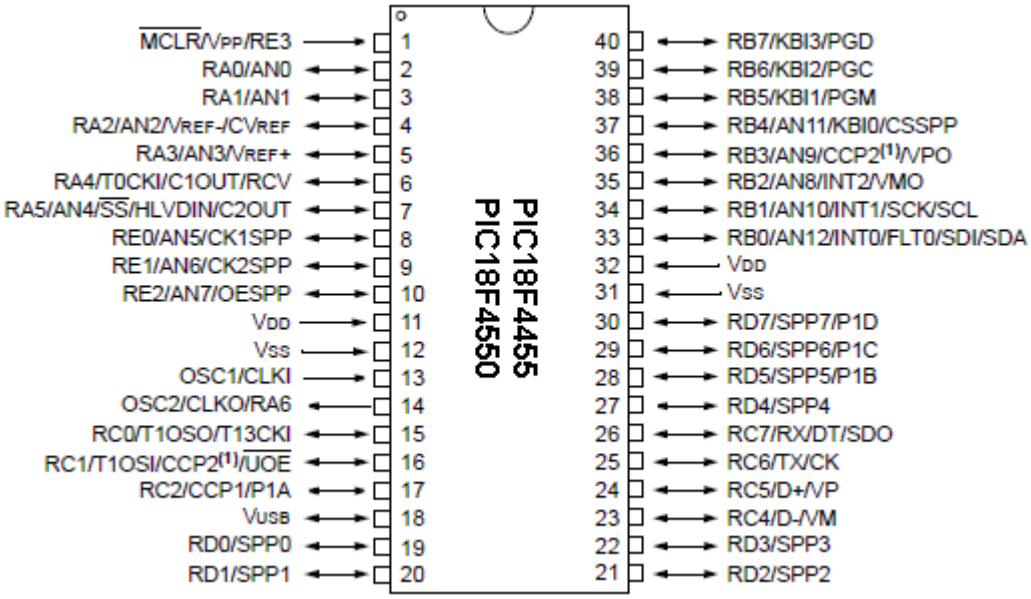


Fig. 19. Diagrama de pinos do microcontrolador PIC18F4550.

Dentre suas principais características, podemos citar:

- Tensão de operação: 2,0V a 5,5V;
- Tecnologia *nanoWatt*: 0,1µA (modo *sleep*);
- Processamento de 12MIPS (*full speed*);
- Memória de programa: 32kB (*flash*);
- Memória RAM: 2kB;
- WDT estendido: até 131s;
- Interrupções com níveis de prioridade;
- *Timers* de 8/16 bits (1/3);
- *Hardware* de multiplicação 8x8 bits;
- EUSART;
- PLL interno;
- Entre outros.

Informações mais detalhadas sobre a família PIC18F, e, especificamente o modelo PIC18F4550, podem ser encontradas nas referências bibliográficas utilizadas [19] e no Apêndice B.

4.1.1.4 Transmissor UHF Elta HAL-2 (*HIGH ACCURACY LOCATOR*)

Este transmissor foi especialmente desenvolvido pela empresa francesa ELTA para aplicações em coleta de dados, suportando tanto o sistema Argos quanto o SBCDA. Possui um barramento de programação que permite a configuração de todos os seus parâmetros de transmissão, como: endereço (*ID*), frequência de transmissão, potência, tamanho do campo de mensagem, entre outros. A Tabela 1 relaciona os pinos do barramento de programação com suas respectivas funções:

Tabela 1: Barramento de programação do transmissor UHF ELTA HAL-2.

Pino	Nome	Função
1	TXD	Transmissão serial de dados (RS-232)
2	RXD	Recepção serial de dados (RS-232)
3	GND	Terminal de terra (0V)
4	FLAG_TX	Indicador de transmissão
5	-	Não utilizado
6	-	Não utilizado
7	Contagem	Contagem de eventos externos
8	GND	Terminal de terra (0V)
9	AD1	Entrada do conversor Analógico/Digital 1
10	AD2	Entrada do conversor Analógico/Digital 2
11	AD3	Entrada do conversor Analógico/Digital 3
12	AD4	Entrada do conversor Analógico/Digital 4

Os terminais de transmissão/recepção serial (*TXD* e *RXD*) operam no padrão RS-232, possibilitando a conexão direta do transmissor a um computador ou outro terminal de dados. O indicador de transmissão (*FLAG_TX*) consiste em um pulso positivo de nível 3,3V enviado a cada transmissão de dados. Pode ser utilizado para sincronismo externo, de acordo com as transmissões realizadas. O terminal de contagem, por sua vez, pode

ser utilizado por eventos externos com um limite de até 32 bits. As entradas dos conversores analógico/digital (*AD1* a *AD4*) podem ser configuradas com uma resolução de até 10 bits, suportando uma tensão máxima de entrada de 3,3V [20]. A Fig. 20 mostra o transmissor UHF ELTA HAL-2:

Fig. 20. *Kit* completo do transmissor UHF ELTA HAL-2.

A programação deste transmissor pode ser realizada através do *software HAL2STORE*, fornecido pelo próprio fabricante, e permite alterar os seguintes parâmetros de transmissão:

- Endereço (*ID*): 20 ou 28 bits;
- Frequência de transmissão: 401,610MHz a 401,650MHz;
- Potência de transmissão: 0,5W a 2W;
- Tipo de mensagem:
  - Externa: 0 a 32 *bytes* (múltiplos de 4);
  - Interna: parâmetros internos do transmissor.
    - Tensão da bateria;
    - Contagem de eventos externos;
    - Conversores analógico/digital;
    - Entre outros.

Para o caso particular deste projeto, o transmissor HAL-2 foi programado para operar com um endereço de 28 bits (720DD00h) na frequência de 401,620MHz, com uma potência de 2W e uma mensagem externa de 32 *bytes*.

Dentre as principais características deste transmissor, considera-se [21]:

- Dimensões: 55mm x 45mm x 15mm;
- Ciclo de transmissão: 30s a 255 dias;
- Tensão de alimentação: 7V a 14V;
- Consumo @ 14V:
  - Mínimo: <50μA (modo *sleep*);
  - Máximo: <650mA (operando com 2W).

Maiores informações sobre o funcionamento deste transmissor, bem como o procedimento de instalação e utilização do *software HAL2STORE*, podem ser encontrados nas referências bibliográficas utilizadas [20]-[21] e arquivos anexos.

4.1.1.5 Antena Synergetics QFH 14A-N

Esta antena foi projetada especialmente para o uso em Plataformas de Coleta de Dados (PCDs), sendo utilizada neste projeto pelo transmissor UHF ELTA HAL-2 para o envio dos dados aos satélites do SBCDA. O modelo QFH 14A-N é completamente selado por um radome de fibra de vidro G-10, com um acabamento em duas camadas brancas de poliuretano, atingindo estendida vida útil sob as condições atmosféricas mais hostis (p.ex. ambiente marítimo). Dentre suas principais características, considera-se [22]:

- Meia-onda helicoidal quadrifilar;
- Modo axial (longitudinal);
- Polarização circular à direita;

- Hemisférica;
- Características elétricas:
  - Frequência: 401MHz;
  - Banda: 4MHz;
  - Potência de entrada: @ 50W;
  - SWR: @ 1,5;
  - Razão axial: @ 5dB;
  - Ganho: 3dBic (no zênite).
- Características mecânicas:
  - Dimensões: 7,6cm x 38,1cm;
  - Peso: 590g.
- Características ambientais:
  - Vento: 185km/h;
  - Chuva: 125mm/h;
  - Temperatura: -65°C a + 65°C;
  - Umidade relativa: 0 a 200%.

A Fig. 21 mostra a antena Synergetics QFH 14A-N:

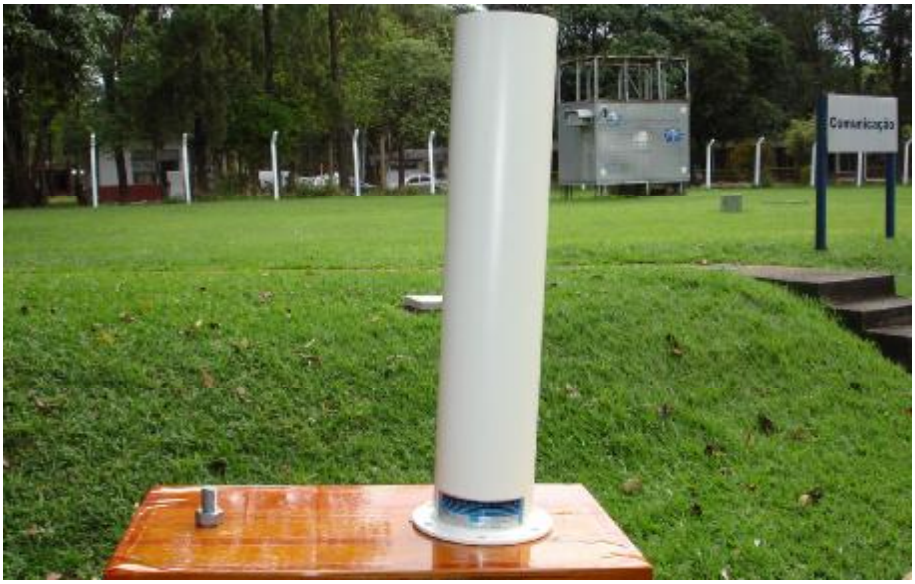


Fig. 21. Antena Synergetics QFH 14A-N.

4.1.1.6 Bateria Unipower UP1250

Esta é uma bateria recarregável de chumbo-ácida, completamente selada e regulada por válvula. Estas baterias foram projetadas de modo a minimizar a geração dos gases, proporcionando uma eficiência de 99% na recombinação dos mesmos durante o seu uso normal. Sua operação é livre de manutenção, pois não existe a necessidade de verificar a densidade do eletrólito e/ou adicionar água ao longo de sua vida útil. Sua construção garante o não vazamento de eletrólito e a operação em qualquer posição sem perda da sua capacidade nominal ou vida útil. As válvulas liberam os gases em excesso caso a pressão interna ultrapasse os níveis normais de pressão, evitando o acúmulo de gases no interior da bateria. A abertura e o fechamento das válvulas de segurança são automáticos [23].

O modelo UP1250 atinge uma média de mil ciclos de recarga ao longo de sua vida útil, produzindo uma tensão de alimentação de 12V com uma corrente nominal de 5,0Ah. Esta capacidade de corrente é suficiente para oferecer ao rastreador da baliza uma autonomia de até cinco dias – tempo suficiente para que a embarcação de pesca seja resgatada em uma situação adversa. A Fig. 22 mostra a bateria UNIPOWER UP1250 que será utilizada na baliza do SBCDA:

Fig. 22. Bateria UNIPOWER UP1250.

4.1.2 Os protocolos de mensagem

O localizador GPS possui um microcontrolador PIC18F4550 responsável não só pela criptografia e codificação dos dados de posição geográfica, mas também pela interface com o receptor GPS e o transmissor UHF, como mostra a Fig. 23:

Fig. 23. Protocolos de interface utilizados no localizador GPS.

O receptor GPS Trimble *Lassen iQ* é capaz de transmitir os dados de posição geográfica baseados em diversos protocolos, sejam estes próprios do fabricante (TSIP e TAIP) ou não (NMEA-0183 e RTCM). Dentre estas opções foi adotada a implementação do protocolo NMEA-0183, representado no *trajeto 1* da Fig. 23. Os dados enviados com base neste protocolo são recebidos e processados pelo PIC18F4550, que forma um campo de mensagem de 160 bits de acordo com o formato de mensagem *Header 0*, representado no *trajeto 2* da Fig. 23. Após a aplicação do algoritmo criptográfico, estes dados estão prontos para serem gravados na memória interna do transmissor UHF ELTA HAL-2, que os transmitirá aos satélites do SBCDA. A gravação destes dados é realizada com base no protocolo *Intel Hex Format*, representado no *trajeto 3* da Fig. 23.

Deste modo, esta sessão objetiva a descrição detalhada dos protocolos e formatos de mensagem envolvidos no processamento dos dados de posição geográfica pelo localizador GPS.

4.1.2.1 O protocolo NMEA-0183

O protocolo de interface NMEA-0183 foi criado pela *National Marine Electronics Association*, inicialmente para permitir a comunicação entre equipamentos de navegação náutica. Com sua difusão entre os padrões industriais de comunicação, este protocolo passou a ser utilizado em outras aplicações, tornando-se um protocolo padrão. Baseado em ASCII, o NMEA-0183 define tanto a interface de comunicação quanto o formato dos dados. A Tabela 2 mostra a interface de comunicação definida por este protocolo:

Tabela 2. Interface de comunicação do protocolo NMEA-0183 [18].

Características do Sinal	Padrão NMEA-0183
Baud rate	4800
Bits de dados	8
Paridade	Nenhum
Bits de Parada (*stop bit*)	1

Este protocolo oferece diversos formatos de mensagens de acordo com a aplicação desejada. Cada formato pode agregar à mensagem dados de localização, velocidade, tempo, ou ainda dados específicos da constelação GPS. A estrutura padrão destas mensagens é da forma [18]:

```
$IDMSG,D1,D2,D3,D4,D5,D6,D7,.....,Dn*CS[CR][LF]
```

Onde,

- **$**: Caractere padrão, indica o início da mensagem;
- **ID**: Mnemônico de indicação da fonte da informação composto por dois caracteres. Neste caso, a identificação *GP* corresponde a um receptor GPS;
- **MSG**: Mnemônico de identificação da mensagem composto por três caracteres. Descreve o conteúdo da mensagem, o número e a ordem do

campo de dados. Para aplicações envolvendo receptores GPS poder ser utilizados até sete formatos de mensagens, como mostra a Tabela 3:

Tabela 3. Formatos de mensagem do protocolo NMEA-0183.

Mensagem	$+ID+MSG	Descrição
GGA	$GPGGA	Mensagem completa da posição fixa
GLL	$GPGLL	Dados de posição geográfica e tempo (UTC)
GSA	$GPGSA	Dados sobre a constelação GPS, PDOP, HDOP, VDOP
GSV	$GPGSV	Dados sobre os satélites visíveis
RMC	$GPRMC	Mensagem reduzida com dados de posição geográfica
VTG	$GPVTG	Dados de velocidade
ZDA	$GPZTG	Dados de tempo: horário (UTC), dia, mês, ano

- **,** : Delimitador dos campos de dados;
- **D1...Dn**: Campos de dados;
- ***** : Delimitador de *checksum*,
- **CS**: *Checksum* de 8 bits, gerado por um XOR *byte* a *byte* entre todos os campos da mensagem, de **$** a *****;
- **[CR]**: Caractere de controle *Carriage Return*, em ASCII;
- **[LF]**: Caractere de controle *Line Feed*, em ASCII.

O receptor GPS *Lassen iQ* pode fornecer até todos os formatos de mensagens simultaneamente, a cada segundo. A programação destes formatos pode ser realizada através do *software Trimble GPS Monitor*, disponibilizado pelo próprio fabricante. Para o caso particular deste projeto, o receptor GPS *Lassen iQ* foi programado para fornecer somente o formato *GGA*, cuja descrição detalhada é mostrada a seguir [18]:

$GPGGA,hhmmss.ss,llll.llll,a,yyyyy.yyyy,b,t,uu,v.vv,wwwww,M,xxx,M,y.y,zzzz*CS<CR><LF>

Onde,

- **hhmmss.ss**: Tempo da posição fixa (UTC):
    - hh: horas (dois dígitos);
    - mm: minutos (dois dígitos);
    - ss.ss: segundos e frações de segundos (dois dígitos cada).
- **llll.llll**: Latitude da posição fixa: grau, minuto, segundo;
- **a**: Indicador da latitude: N (norte), ou S (sul);
- **yyyyy.yyyy** Longitude da posição fixa: grau, minuto, segundo;
- **b**: Indicador da longitude: E (leste) ou W (oeste);
- **t**: Indicador de qualidade do GPS:
    - “0” = sem GPS;
    - “1” = com GPS;
    - “2” = com DGPS (GPS diferencial).
- **uu**: Número de satélites GPS visíveis;
- **v.vv**: Diluição horizontal de precisão (HDOP);
- **wwwww**: Altitude da posição fixa;
- **M**: Unidade da altitude (metros);
- **xxx**: Separação no geóide, entre o WGS84 e o nível do mar;
- **M**: Unidade da separação no geóide (metros);
- **y.y**: Campo reservado para utilização do DGPS;
- **zzzz**: Campo reservado para utilização do DGPS.
- **[CR]**: Caractere de controle *Carriage Return*, em ASCII;
- **[LF]**: Caractere de controle *Line Feed*, em ASCII.

4.1.2.2 O formato de mensagem *Header-0*

Cada mensagem transmitida aos satélites do SBCDA deve ser estruturada de acordo com um formato de mensagem padrão. O formato *Header* 0 é utilizado para dados de localização onde se tem uma precisão da ordem de

0,001°, aproximadamente 100m. Este formato provê a cada mensagem uma posição absoluta (64 bits) e três posições relativas (32 bits cada), estabelecendo um campo final de 160 bits. A posição absoluta de cada mensagem refere-se à posição mais atual, ou seja, a última posição na qual foi realizada a aquisição de dados pelo receptor GPS. As posições relativas são juntamente transmitidas com a absoluta com o intuito de oferecer um histórico da posição ao usuário. Este recurso é imprescindível, pois nem sempre os satélites do SBCDA estarão visíveis para a embarcação de pesca, podendo ocasionar períodos sem recepção de dados. Deste modo, as posições relativas devem ser sempre referidas à posição absoluta, contendo somente as variações entre as coordenadas latitude e longitude (*deltas*), e o tempo de atraso entre cada transmissão (*delays*) [24].

A Tabelas 4 e 5 mostram o conteúdo das posições absoluta e relativas, respectivamente:

Tabela 4. Primeira posição fixa (absoluta) [24].

Header	**CRC**	**Longitude**	**Latitude**	**Horas**	**Minutos**	**Período**
4 bits	8 bits	19 bits	18 bits	5 bits	6 bits	4 bits

Tabela 5. Segunda, terceira e quarta posições fixas (relativas) [24].

D **Longitude**	D **Latitude**	***Delay***	***Time Index***
13 bits	13 bits	4 bits	2 bits

Para os itens das Tabelas 4 e 5, as seguintes convenções foram adotadas:

- Longitude absoluta: 0 (0°) a 360000 (360°);
- Latitude absoluta: 0 (90°S) a 180000 (90°N);
- Horas: 0 a 23 horas;
- Minutos: 0 a 59 minutos;
- Longitude relativa: 0 (-4°) a 8000 (+4°);

- Latitude relativa: 0 (-4°) a 8000 (+4°);
- *Delay*: 0 a 15 minutos;
- *Time index*: Adota-se como “00” (binário), indicando que o tempo real da aquisição é o contido na mensagem.

O campo *Header* identifica o formato da mensagem. Neste caso deve ser igual à zero (formato *Header 0*). O campo *CRC* (*Cyclic Redundancy Check*) é calculado para cada mensagem transmitida de acordo com o polinômio: X^7+X+1. É utilizado como um mecanismo de verificação de erros na mensagem recebida. O campo *Período* se refere à aquisição da posição geográfica, sendo este um parâmetro configurável pelo usuário [24]. Logo, cada período de aquisição corresponde a um conteúdo diferente no campo *Período* da mensagem, conforme mostra a Tabela 6:

Tabela 6. Períodos de aquisição da posição geográfica [24].

Conteúdo do campo *Período*	Período de aquisição correspondente
0000	20 minutos
1001	30 minutos
1010	45 minutos
0011	1 hora
1100	2 horas
0101	3 horas
0110	4 horas
1111	1 minuto (para demonstração)

O bit mais significativo do *nibble* correspondente ao conteúdo do campo *Período* é gerado através de uma operação lógica XOR bit a bit entre os três bits menos significativos do mesmo.

4.1.2.3 O protocolo *Intel hex format*

Este protocolo é utilizado para a gravação dos dados de posição geográfica na memória interna do transmissor HAL-2. A primeira mensagem enviada por este transmissor é mostrada no Quadro 1:

Quadro 1. Primeira mensagem enviada pelo transmissor HAL-2.

```
00 00 00  00   1B    5B 32 4A 1B   5B 31 3B 31 48  0D   0A
   (nulos)    (esc)  [  2  J (esc) [  1  ;  1  H  (cr) (lf)

30 33 45 36 35 34 35 33
0  3  E  6  5  4  5  3

 0D 0A  53 6F 66 74 77 61 72 65 3A 34 2E 30 32 20 5B 32 30 32 34
(cr)(lf) S o  f  t  w  a  r  e  :  4  .  0  2 esp [  2  0  2  4

30 33 31 34  1B   5B 4B 5D  1B    5B 4B
0  3  1  4  (esc) [  K  ]  (esc)  [  K



MENSAGEM NO HYPERTERMINAL:

03E65453
Software:4.02 [20240314]
```

O Quadro 1 mostra cada caractere enviado pelo transmissor HAL-2 nos formatos hexadecimal e ASCII. Na parte inferior consta a mensagem original recebida pela porta serial do computador, via *software HyperTerminal* do *Windows*.

O envio desta mensagem inicia o processo de gravação dos dados, como mostrado a seguir:

1. O PIC18F4550 envia os seguintes *bytes*, em hexadecimal: 1B5B484Ch;
2. O transmissor UHF ELTA HAL-2 envia os seguintes *bytes*, em hexadecimal: 1B5B48444F574E4C4F4144h;

3. O PIC18F4550 envia a seguinte palavra de configuração, em hexadecimal: 1B5B484C1310h;
4. O PIC18F4550 envia logo em seguida a mensagem a ser gravada na memória interna do transmissor HAL-2, em ASCII, como mostra o exemplo abaixo:

:1000390001020304050607080910111213141516 05 (CR) (LF)
:1000490017181920212223242526272829303132 5F (CR) (LF)
:00000001FF (CR) (LF)

Neste exemplo foi simulada a gravação de uma sequência de 32 bytes, em ASCII: “01020304050607080910111213141516171819202122232425262728 29303132”.

As duas primeiras linhas carregam consigo 16 *bytes* cada, enquanto a última encerra a sessão de gravação.

Como se pode ver, cada linhas deve ser iniciada com o caractere *dois pontos* (:), em ASCII, indicando início de gravação da mensagem:

:1000390001020304050607080910111213141516 05 (CR) (LF)
:1000490017181920212223242526272829303132 5F (CR) (LF)
:00000001FF (CR) (LF)

Os dois caracteres seguintes indicam a quantidade de pares de *bytes* que serão enviados ao transmissor HAL-2. Nota-se que a última linha não carrega dados:

:10 0039000102030405060708091011121314151605 (CR) (LF)
:10 0049001718192021222324252627282930313 25F (CR) (LF)
:00 000001FF (CR) (LF)

Os quatro caracteres seguintes representam a posição de memória a partir da qual o transmissor armazenará os *bytes* da mensagem:

:10003900010203040506070809101112131415160 5 (CR) (LF)
:10004900171819202122232425262728293031325F (CR) (LF)
:00000001FF (CR) (LF)

Os dois caracteres seguintes informam se o formato da mensagem é do tipo *00* ou *01*. Mensagens do tipo *00* informam a gravação de dados, enquanto mensagens do tipo *01* informam o fim de uma sessão de gravação – não carregando quaisquer *bytes* de mensagem. Nota-se novamente que as duas primeiras linhas são do tipo *00* (carregam dados), enquanto a última é do tipo *01* (não carrega dados), e encerra a sessão de gravação:

:10003900010203040506070809101112131415160 5 (CR) (LF)
:10004900171819202122232425262728293031325F (CR) (LF)
:00000001FF (CR) (LF)

Os trinta e dois caracteres seguintes indicam os dados que serão gravados na memória interna do transmissor HAL-2:

:10003900010203040506070809101112131415160 5 (CR) (LF)
:10004900171819202122232425262728293031325F (CR) (LF)
:00000001FF (CR) (LF)

Os dois caracteres seguintes representam o *checksum* da mensagem, uma soma que consiste em uma operação lógica XOR entre todos os termos da linha, exceto o caractere *dois pontos* (:). Ao final, aplica-se um “complemento de 2” ao resultado obtido por esta soma, e então retira-se o *byte* menos significativo:

:1000390001020304050607080910111213141516 05 (CR) (LF)
:10004900171819202122232425262728293031325F (CR) (LF)
:00000001FF (CR) (LF)

Por fim, temos os caracteres de controle CR (*Carriage Return*) e LF (*Line Feed*), necessários ao fim de cada linha:

:1000390001020304050607080910111213141516 05 (CR) (LF)
:1000490017181920212223242526272829303132 5F (CR) (LF)
:00000001FF (CR) (LF)

Maiores informações sobre este protocolo de comunicação podem ser encontradas nos arquivos anexos. O Apêndice D ainda oferece uma tabela de conversão ASCII – Hexadecimal.

4.1.3 O algoritmo de criptografia AES (Rijndael)

Para se iniciar um processo de criptografia, devemos primeiramente escolher uma *chave forte* para o sistema. Entende-se como *chave forte* aquela que é de difícil dedução [25]. Tanto a chave quanto os blocos de mensagem podem assumir três tamanhos: 16 bytes, 24 bytes e 32 bytes. O número de iterações de transformação da mensagem é variável em função dos tamanhos da chave e mensagem, como mostra a Tabela 7:

Tabela 7. Número de iterações do algoritmo AES [25].

Chave	Mensagem		
	16 bytes	**24 bytes**	**32 bytes**
16 bytes	10*	12	14
24 bytes	12	12	14
32 bytes	14	14	14

*Para o caso particular deste projeto, foi utilizada uma chave simétrica de 16 bytes junta a blocos de mensagens também de 16 bytes. Logo, são necessárias dez iterações para a transformação da mensagem.

A chave e cada bloco da mensagem devem ser constituídos matricialmente, como mostram as Tabelas 8 e 9:

Tabela 8. Matriz dos blocos de mensagem [26].

Bloco [0]	Bloco [4]	Bloco [8]	Bloco [12]
Bloco [1]	Bloco [5]	Bloco [9]	Bloco [13]
Bloco [2]	Bloco [6]	Bloco [10]	Bloco [14]
Bloco [3]	Bloco [7]	Bloco [11]	Bloco [15]

Tabela 9. Matriz dos blocos de chave simétrica [26].

Chave [0]	Chave [4]	Chave [8]	Chave [12]
Chave [1]	Chave [5]	Chave [9]	Chave [13]
Chave [2]	Chave [6]	Chave [10]	Chave [14]
Chave [3]	Chave [7]	Chave [11]	Chave [15]

Cada campo da matriz mensagem (*bloco[n]*) representa um *byte* da mensagem a ser criptografada, enquanto cada campo da matriz chave (*chave[n]*) representa um *byte* da chave secreta de criptografia. Com a chave devidamente escolhida, dá-se início ao processo de criptografia.

4.1.3.1 Processo de cifragem AES (Rijndael)

A representação completa do processo de cifragem AES (Rijndael) é demonstrada pelo diagrama em blocos da Fig. 24. Este diagrama mostra o *Contador de Rodadas*, responsável pelo número de iterações do algoritmo AES (Rijndael). Conforme já apresentado, este processo de cifragem deve executar dez iterações. O *Rcon* é um registrador de controle, e deve ser iniciado com o valor 1 (um). Respeitadas estas considerações, as sessões seguintes descrevem detalhadamente cada bloco do diagrama da Fig. 24.

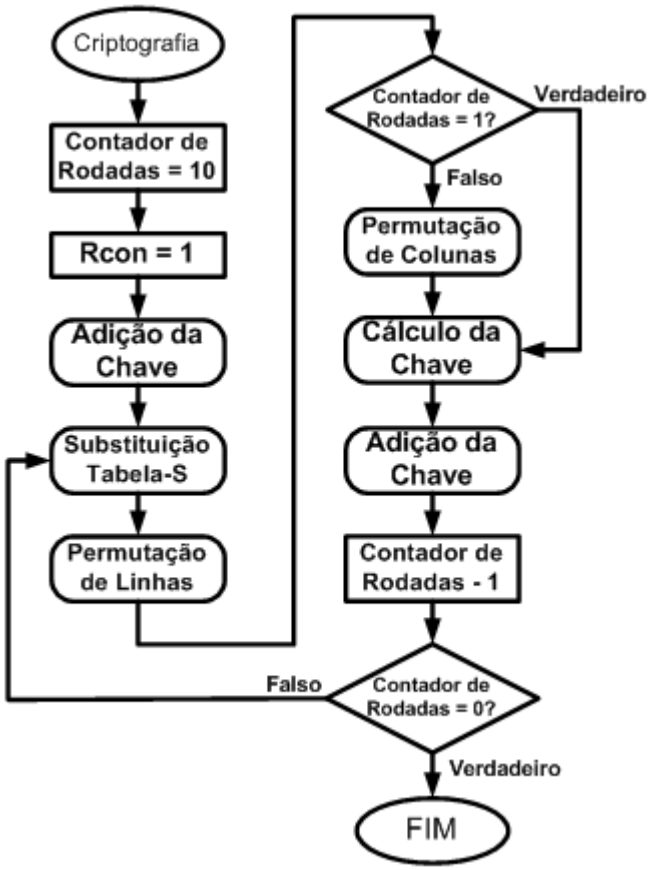


Fig. 24. Processo de cifragem AES (Rijndael) [24].

4.1.3.1.1 Adição da chave

Consiste em uma operação lógica XOR, realizada *byte* a *byte* entre a matriz chave e a matriz mensagem [5], como mostra o exemplo a seguir:

Mensagem: 0x3243F6A8885A308D313198A2E0370734h;
Chave: 0x2B7E151628AED2A6ABF7158809CF4F3Ch.

32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

$\oplus$

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

=

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

O trecho da programação que realiza esta função é mostrado no Quadro 2:

Quadro 2: Programação da Adição da Chave (cifragem).

```
#define BLOCKSIZE 16          //Constante do numero de bytes a serem criptografados.

Contador_rodadas = 10;        //Dez iterações para o processo de cifragem AES.
Rcon = 1;                     //Inicilização do registrador Rcon.

for(i=0;i<BLOCKSIZE;i++)
{
    msg_GPS[i]^=AESkey[i];    //Rotina de "Adição da Chave".
}
```

Como se pode ver, o *Contador de Rodadas* e o registrador *Rcon* devem ser previamente inicializados, e então se executa a rotina de Adição da Chave. A constante *BLOCKSIZE* define o tamanho dos blocos de mensagem e chave simétrica. Conforme apresentado, para este caso devem ser igual a 16 (dezesseis). O vetor *msg_GPS[n]* corresponde aos blocos de mensagem, indexados pela variável de controle *i*. Analogamente, o vetor *AESKey[n]* corresponde aos blocos de chave simétrica, também indexados pela variável de controle *i*. Deste modo, ocorre a operação lógica XOR *byte* a *byte* entre ambos os vetores.

4.1.3.1.2 Substituição na Tabela-S

Neste passo, cada bloco da matriz mensagem deve ser substituído por seu respectivo valor na Tabela-S [5]. Como exemplo, se um *Bloco[n]* da matriz mensagem possui armazenado o valor 73h, olha-se a posição 73 na Tabela-S (x=07; y=30) e armazena-se o conteúdo desta posição no *Bloco[n]* (no caso, 8Fh). A Fig. 25 ilustra este exemplo com a Tabela-S, que por sua vez é conhecida e disponibilizada pelo próprio algoritmo de cifragem:

x \ y	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
01	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
02	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
03	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
04	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
05	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
06	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
07	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
08	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
09	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
0A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
0B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
0C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
0D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
0E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
0F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Fig. 25. Tabela-S [27].

O trecho da programação que realiza esta função é mostrado no Quadro 3:

Quadro 3. Programação da Substituição na Tabela-S.

```
#define BLOCKSIZE 16              //Constante do numero de bytes a serem criptografados.

rom const unsigned char STable[ ] =
{
0x63,0x7C,0x77,0x7B,0xF2,0x6B,0x6F,0xC5,0x30,0x01,0x67,0x2B,0xFE,0xD7,0xAB,0x76,
0xCA,0x82,0xC9,0x7D,0xFA,0x59,0x47,0xF0,0xAD,0xD4,0xA2,0xAF,0x9C,0xA4,0x72,0xC0,
0xB7,0xFD,0x93,0x26,0x36,0x3F,0xF7,0xCC,0x34,0xA5,0xE5,0xF1,0x71,0xD8,0x31,0x15,
0x04,0xC7,0x23,0xC3,0x18,0x96,0x05,0x9A,0x07,0x12,0x80,0xE2,0xEB,0x27,0xB2,0x75,
0x09,0x83,0x2C,0x1A,0x1B,0x6E,0x5A,0xA0,0x52,0x3B,0xD6,0xB3,0x29,0xE3,0x2F,0x84,
0x53,0xD1,0x00,0xED,0x20,0xFC,0xB1,0x5B,0x6A,0xCB,0xBE,0x39,0x4A,0x4C,0x58,0xCF,
0xD0,0xEF,0xAA,0xFB,0x43,0x4D,0x33,0x85,0x45,0xF9,0x02,0x7F,0x50,0x3C,0x9F,0xA8,
0x51,0xA3,0x40,0x8F,0x92,0x9D,0x38,0xF5,0xBC,0xB6,0xDA,0x21,0x10,0xFF,0xF3,0xD2,
0xCD,0x0C,0x13,0xEC,0x5F,0x97,0x44,0x17,0xC4,0xA7,0x7E,0x3D,0x64,0x5D,0x19,0x73,
0x60,0x81,0x4F,0xDC,0x22,0x2A,0x90,0x88,0x46,0xEE,0xB8,0x14,0xDE,0x5E,0x0B,0xDB,
0xE0,0x32,0x3A,0x0A,0x49,0x06,0x24,0x5C,0xC2,0xD3,0xAC,0x62,0x91,0x95,0xE4,0x79,
0xE7,0xC8,0x37,0x6D,0x8D,0xD5,0x4E,0xA9,0x6C,0x56,0xF4,0xEA,0x65,0x7A,0xAE,0x08,
0xBA,0x78,0x25,0x2E,0x1C,0xA6,0xB4,0xC6,0xE8,0xDD,0x74,0x1F,0x4B,0xBD,0x8B,0x8A,
0x70,0x3E,0xB5,0x66,0x48,0x03,0xF6,0x0E,0x61,0x35,0x57,0xB9,0x86,0xC1,0x1D,0x9E,
0xE1,0xF8,0x98,0x11,0x69,0xD9,0x8E,0x94,0x9B,0x1E,0x87,0xE9,0xCE,0x55,0x28,0xDF,
0x8C,0xA1,0x89,0x0D,0xBF,0xE6,0x42,0x68,0x41,0x99,0x2D,0x0F,0xB0,0x54,0xBB,0x16
};                                        //Componentes da Tabela-S.

for(i=0;i<BLOCKSIZE;i++)
{
    msg_GPS[i]=STable[msg_GPS[i]];        //Substituição na Tabela-S.
}
```

Por medida de segurança, os elementos da matriz da Tabela-S foram armazenados no *software* como *constantes*, evitando o risco de serem alterados acidentalmente durante a execução do algoritmo. Como se pode ver, o vetor *msg_GPS[n]* é substituído pelo seu próprio valor na Tabela-S.

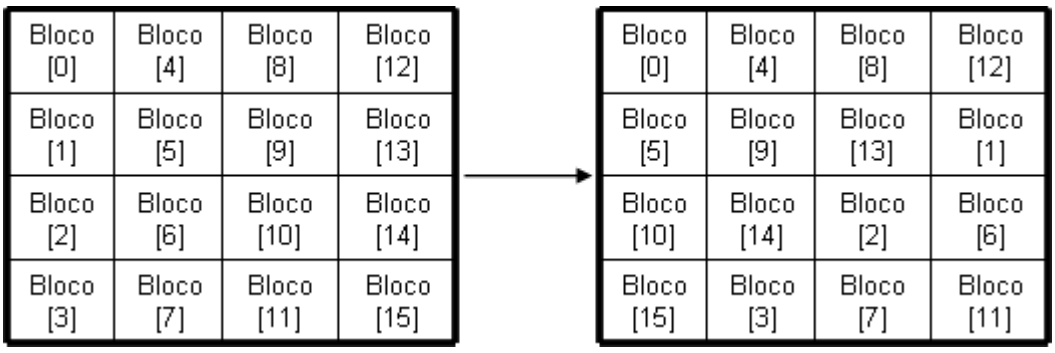
4.1.3.1.3 Permutação de linhas

Neste passo é realizado um processo de rotação cíclica à esquerda, aplicado às linhas da matriz mensagem [5]. Os números de rotações de cada linha são mostrados na Tabela 10:

Tabela 10. Número de rotações do processo de cifragem AES [28].

	Nº Rotações Linha 0	**Nº Rotações Linha 0**	**Nº Rotações Linha 0**	**Nº Rotações Linha 0**
Mensagem 16 bytes	0	1	2	3

Deste modo, tem-se:



O trecho da programação que realiza esta função é mostrado no Quadro 4:

Quadro 4. Programação da Permutação de Linhas (cifragem).

```
EncodeShiftRow(msg_GPS);                          //Chamada da função de Permutação de Linhas.

void EncodeShiftRow(unsigned char* stateTable)    //Função de Permutação de Linhas.
{
      unsigned char temp;                         //Criação da variável temporária temp.

      temp=stateTable[1];                         //Permutação na segunda linha.
      stateTable[1]=stateTable[5];
      stateTable[5]=stateTable[9];

      stateTable[9]=stateTable[13];
      stateTable[13]=temp;

      temp=stateTable[2];                         //Permutação na terceira linha.
      stateTable[2]=stateTable[10];
      stateTable[10]=temp;
      temp=stateTable[14];
      stateTable[14]=stateTable[6];
      stateTable[6]=temp;

      temp=stateTable[3];                         //Permutação na quarta linha.
  stateTable[3]=stateTable[15];
      stateTable[15]=stateTable[11];
      stateTable[11]=stateTable[7];
      stateTable[7]=temp;
}
```

Neste caso, a substituição dos termos da matriz mensagem ocorre individualmente, *byte* a *byte*.

4.1.3.1.4 Permutação de colunas

Este processo requer a criação de uma matriz fixa *c(x)*, tendo como entrada os vetores *a(n)*, e saída os vetores *b(n)* [5]. Então, faz-se a multiplicação matricial:

$$c(x) = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \longrightarrow \begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a0 \\ a1 \\ a2 \\ a3 \end{bmatrix}$$

Deste modo, temos como resultado [5]:

$$
\begin{aligned}
b_0 &= a_0 \times 02 \oplus a_1 \times 03 \oplus a_2 \times 01 \oplus a_3 \times 01 \\
b_1 &= a_0 \times 01 \oplus a_1 \times 02 \oplus a_2 \times 03 \oplus a_3 \times 01 \\
b_2 &= a_0 \times 01 \oplus a_1 \times 01 \oplus a_2 \times 02 \oplus a_3 \times 03 \\
b_3 &= a_0 \times 03 \oplus a_1 \times 01 \oplus a_2 \times 01 \oplus a_3 \times 02
\end{aligned}
$$

Porém, este processo ainda estabelece um conjunto de regras especiais, como mostrado a seguir [5]:

$$
\begin{aligned}
a_n \times 01 &= a_n \\
a_n \times 02 &= xtime(a_n) \\
a_n \times 03 &= a_n \oplus xtime(a_n)
\end{aligned}
$$

Aplicando estas regras especiais ao resultado obtido na multiplicação matricial, temos [5]:

$$
\begin{aligned}
b_0 &= [xtime(a_0)] \oplus [a_1 \oplus xtime(a_1)] \oplus [a_2] \oplus [a_3] \\
b_1 &= [a_0] \oplus [xtime(a_1)] \oplus [a_2 \oplus xtime(a_2)] \oplus [a_3] \\
b_2 &= [a_0] \oplus [a_1] \oplus [xtime(a_2)] \oplus [a_3 \oplus xtime(a_3)] \\
b_3 &= [a_{03} \oplus xtime(a_0)] \oplus [a_1] \oplus [a_2] \oplus [xtime(a_3)]
\end{aligned}
$$

O trecho da programação que realiza esta função é mostrado no Quadro 5:

Quadro 5. Programação da Permutação de Colunas (cifragem).

```
#define xtime(a) (((a)<0x80)?(a)<<1:(((a)<<1)^0x1b) )      //Macro de X-TIME.
#define BLOCKSIZE 16                          //Constante do numero de bytes a serem criptografados.

if(Contador_rodadas != 1)                     // Pula esta função na última rodada.
{
      unsigned char aux4,aux1,aux2,aux3;    //Criação de variáveis auxiliares.
      for(i=0;i<BLOCKSIZE;i+=4)
      {
            aux1=msg_GPS[i+0]^msg_GPS[i+1];
            aux3=msg_GPS[i+2]^msg_GPS[i+3];
            aux4=aux1^aux3;
            aux2=msg_GPS[i+2]^msg_GPS[i+1];

            aux1=xtime(aux1);
            aux2=xtime(aux2);
            aux3=xtime(aux3);

            msg_GPS[i+0]=aux4^aux1^msg_GPS[i+0];
            msg_GPS[i+1]=aux4^aux2^msg_GPS[i+1];
            msg_GPS[i+2]=aux4^aux3^msg_GPS[i+2];
            msg_GPS[i+3]=msg_GPS[i+0]^msg_GPS[i+1]^msg_GPS[i+2]^aux4;
      }
}
```

Por definição, *xtime* é um registrador de deslocamento linear de *feedback* que se repete a cada 51 ciclos, sendo utilizado para implementação das regras especiais deste processo [28]. No trecho de programa mostrado no Quadro 5, este registrador foi implementado através de uma *macro*, a qual atualiza seu valor a cada iteração do processo de cifragem.

4.1.3.1.5 Cálculo da chave

Esta rotina tem por função calcular uma nova chave, que será utilizada na iteração seguinte. Este é um dos grandes recursos que o algoritmo AES (Rijndael) possui para incrementar o nível de segurança dos dados. Cada uma das dez iterações utiliza uma chave diferente, que é baseada na chave anterior [5]. Procede da seguinte maneira:

- Passo 1: Na matriz das chaves, faz-se uma operação lógica XOR entre a Coluna 0 e o respectivo valor da Coluna 3, na Tabela-S [5]:

$$Chave[0]\hat{} = Tabela - S[Chave(12)]$$
$$Chave[1]\hat{} = Tabela - S[Chave(13)]$$
$$Chave[2]\hat{} = Tabela - S[Chave(14)]$$
$$Chave[3]\hat{} = Tabela - S[Chave(15)]$$

- Passo 2: Faz-se uma operação lógica XOR entre o conteúdo de *Chave [0]* e o registrador *Rcon* [5]:

$$Chave[0]\hat{} = Rcon$$

- Passo 3: Faz-se uma atualização do valor da variável *Rcon* de acordo com a variável *xtime* [5]:

$$Rcon = xtime(Rcon)$$

- Passo 4: Faz-se uma operação lógica XOR entre as colunas da matriz das chaves, na sequência mostrada abaixo [5]:

$$\begin{matrix} Chave[4]\hat{} = Chave[0] \\ Chave[5]\hat{} = Chave[1] \\ Chave[6]\hat{} = Chave[2] \\ Chave[7]\hat{} = Chave[3] \end{matrix} \longrightarrow \begin{matrix} Chave[8]\hat{} = Chave[4] \\ Chave[9]\hat{} = Chave[5] \\ Chave[10]\hat{} = Chave[6] \\ Chave[11]\hat{} = Chave[7] \end{matrix} \longrightarrow \begin{matrix} Chave[12]\hat{} = Chave[8] \\ Chave[13]\hat{} = Chave[9] \\ Chave[14]\hat{} = Chave[10] \\ Chave[15]\hat{} = Chave[11] \end{matrix}$$

O trecho da programação que realiza esta função é mostrado no Quadro 6:

Quadro 6. Programação do Cálculo de Chave (cifragem).

```
AESkey[0]=0x6c;                          //Inicialização da chave simétrica de 128 bits:
AESkey[1]=0xb3;                          //Chave: 6CB3DF613E590FA095CFA61BEBBAF960h.
AESkey[2]=0xdf;
AESkey[3]=0x61;
AESkey[4]=0x3e;
AESkey[5]=0x59;
AESkey[6]=0x0f;
AESkey[7]=0xa0;
AESkey[8]=0x95;
AESkey[9]=0xcf;
AESkey[10]=0xa6;
AESkey[11]=0x1b;
AESkey[12]=0xeb;
AESkey[13]=0xba;
AESkey[14]=0xf9;
AESkey[15]=0x60;

EncKeySchedule(AESkey);                  //Chamada da função de cálculo de chave.

void EncKeySchedule(unsigned char* key)  //Função de cálculo de chave.
{
      key[0]^=STable[key[13]];           //Passo 1.
      key[1]^=STable[key[14]];
      key[2]^=STable[key[15]];
      key[3]^=STable[key[12]];

      key[0]^=Rcon;                            //Passo 2.

      Rcon=xtime(Rcon);                        //Passo 3.

      key[4]^=key[0];                    //Passo 4.
      key[5]^=key[1];
      key[6]^=key[2];
      key[7]^=key[3];

      key[8]^=key[4];
      key[9]^=key[5];
      key[10]^=key[6];
      key[11]^=key[7];

      key[12]^=key[8];
      key[13]^=key[9];
      key[14]^=key[10];
      key[15]^=key[11];
}
```

4.1.3.2 Processo de decifragem AES (Rijndael)

A representação completa do processo de decifragem AES (Rijndael) é demonstrada pelo diagrama em blocos da Fig. 26:

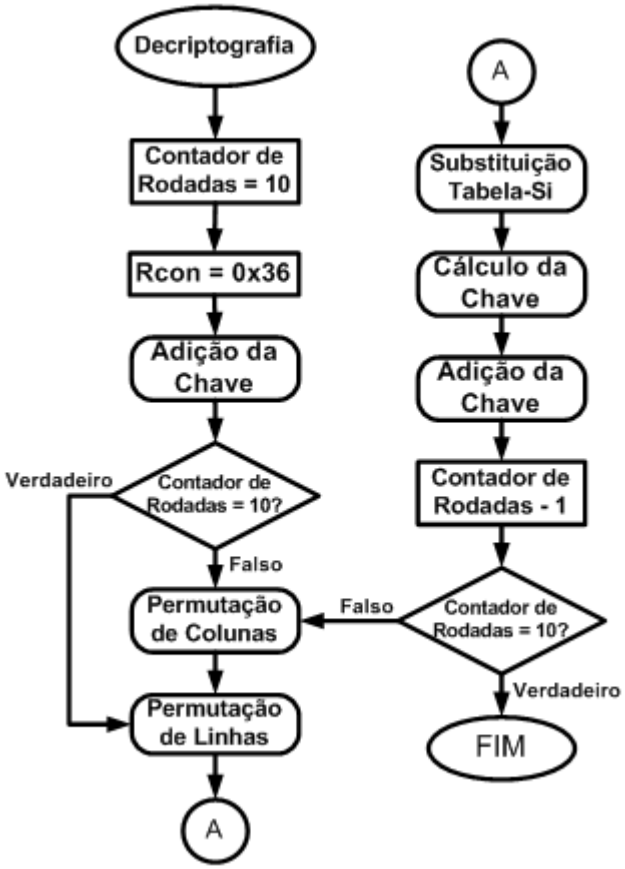


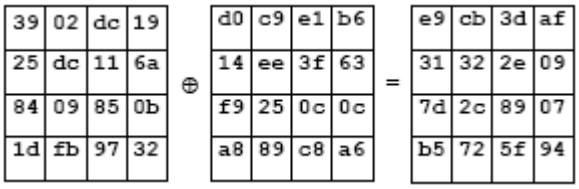
Fig. 26. Processo de decifragem AES (Rijndael) [24].

O diagrama em blocos da Fig. 26 mostra novamente o *Contador de Rodadas*, responsável pelo número de iterações do algoritmo AES (Rijndael). Conforme visto, este processo de decifragem também requer a execução de dez iterações. O registrador *Rcon* deve ser iniciado com o valor 36h (em hexadecimal). Respeitadas estas considerações, as sessões seguintes descrevem detalhadamente cada bloco do diagrama da Fig. 26.

4.1.3.2.1 Adição da chave

Consiste em uma operação lógica XOR, realizada *byte* a *byte* entre a matriz chave e a matriz mensagem cifrada [5], como mostra o exemplo a seguir:

Mensagem cifrada: 0x 3925841D02DC09FBDC118597196A0B32h;
Chave: 0x D014F9A8C9EE2589E13F0CC8B6630CA6h.



O trecho da programação que realiza esta função é mostrado no Quadro 7:

Quadro 7: Programação da Adição da Chave (decifragem).

```
#define BLOCKSIZE 16          //Constante do numero de bytes a serem criptografados.

Contador_rodadas = 10;        //Dez iterações para o processo de cifragem AES.
Rcon = 0x36;                  //Inicilização do registrador Rcon.

for(i=0;i<BLOCKSIZE;i++)
{
    msg_GPS[i]^=AESkey[i];    //Rotina de "Adição da Chave".
}
```

O *Contador de Rodadas* e o registrador *Rcon* devem ser previamente inicializados, e então se executa a rotina de Adição da Chave. A constante *BLOCKSIZE* define o tamanho dos blocos de mensagem e chave simétrica. Conforme visto anteriormente, para este caso devem ser igual a 16 (dezesseis). O vetor *msg_GPS[n]* corresponde aos blocos de mensagem, indexados pela variável de controle *i*. Analogamente, o vetor *AESKey[n]* corresponde aos blocos de chave simétrica, também indexados pela variável

de controle *i*. Deste modo, ocorre a operação lógica XOR *byte* a *byte* entre ambos os vetores.

4.1.3.2.2 Permutação de colunas

Este processo requer a criação de uma matriz fixa *c(x)*, tendo como entrada os vetores *a(n)*, e saída os vetores *b(n)* [5]. Então, faz-se a multiplicação matricial:

$$c(x) = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \longrightarrow \begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \begin{bmatrix} a0 \\ a1 \\ a2 \\ a3 \end{bmatrix}$$

Deste modo, temos como resultado [5]:

$$b_0 = a_0 \times 0E \oplus a_1 \times 0B \oplus a_2 \times 0D \oplus a_3 \times 09$$
$$b_1 = a_0 \times 09 \oplus a_1 \times 0E \oplus a_2 \times 0B \oplus a_3 \times 0D$$
$$b_2 = a_0 \times 0D \oplus a_1 \times 09 \oplus a_2 \times 0E \oplus a_3 \times 0B$$
$$b_3 = a_0 \times 0B \oplus a_1 \times 0D \oplus a_2 \times 09 \oplus a_3 \times 0E$$

Porém, este processo ainda estabelece um conjunto de regras especiais, como mostrado a seguir:

$$a_n \times 09 = a_n \oplus xtime(xtime(xtime(xtime(a_n))))$$
$$a_n \times 0B = a_n \oplus xtime(xtime(xtime(xtime(xtime(a_n)))))$$
$$a_n \times 0D = a_n \oplus xtime(xtime(xtime(xtime(xtime(xtime(a_n))))))$$
$$a_n \times 0E = a_n \oplus xtime(xtime(xtime(xtime(xtime(xtime(xtime(a_n)))))))$$

Nota-se que as regras especiais para o processo de decifragem AES requerem um cálculo matemática intenso, principalmente para um microcontrolador de 8 bits, como o PIC18F4550. Neste caso, para simplificar o

cálculo matemático, são utilizadas tabelas pré-calculadas para: *xtime(a)*, *xtime(xtime(a))*; *xtime(xtime(xtime(a)))*.

O trecho da programação que realiza esta função é mostrado no Quadro 8:

Quadro 8. Programação da Permutação de Colunas (decifragem).

```
rom const unsigned char x1[ ] =
{
0x00,0x02,0x04,0x06,0x08,0x0A,0x0C,0x0E,0x10,0x12,0x14,0x16,0x18,0x1A,0x1C,0x1E,
0x20,0x22,0x24,0x26,0x28,0x2A,0x2C,0x2E,0x30,0x32,0x34,0x36,0x38,0x3A,0x3C,0x3E,
0x40,0x42,0x44,0x46,0x48,0x4A,0x4C,0x4E,0x50,0x52,0x54,0x56,0x58,0x5A,0x5C,0x5E,
0x60,0x62,0x64,0x66,0x68,0x6A,0x6C,0x6E,0x70,0x72,0x74,0x76,0x78,0x7A,0x7C,0x7E,
0x80,0x82,0x84,0x86,0x88,0x8A,0x8C,0x8E,0x90,0x92,0x94,0x96,0x98,0x9A,0x9C,0x9E,
0xA0,0xA2,0xA4,0xA6,0xA8,0xAA,0xAC,0xAE,0xB0,0xB2,0xB4,0xB6,0xB8,0xBA,0xBC,0xBE,
0xC0,0xC2,0xC4,0xC6,0xC8,0xCA,0xCC,0xCE,0xD0,0xD2,0xD4,0xD6,0xD8,0xDA,0xDC,0xDE,
0xE0,0xE2,0xE4,0xE6,0xE8,0xEA,0xEC,0xEE,0xF0,0xF2,0xF4,0xF6,0xF8,0xFA,0xFC,0xFE,
0x1B,0x19,0x1F,0x1D,0x13,0x11,0x17,0x15,0x0B,0x09,0x0F,0x0D,0x03,0x01,0x07,0x05,
0x3B,0x39,0x3F,0x3D,0x33,0x31,0x37,0x35,0x2B,0x29,0x2F,0x2D,0x23,0x21,0x27,0x25,
0x5B,0x59,0x5F,0x5D,0x53,0x51,0x57,0x55,0x4B,0x49,0x4F,0x4D,0x43,0x41,0x47,0x45,
0x7B,0x79,0x7F,0x7D,0x73,0x71,0x77,0x75,0x6B,0x69,0x6F,0x6D,0x63,0x61,0x67,0x65,
0x9B,0x99,0x9F,0x9D,0x93,0x91,0x97,0x95,0x8B,0x89,0x8F,0x8D,0x83,0x81,0x87,0x85,
0xBB,0xB9,0xBF,0xBD,0xB3,0xB1,0xB7,0xB5,0xAB,0xA9,0xAF,0xAD,0xA3,0xA1,0xA7,0xA5,
0xDB,0xD9,0xDF,0xDD,0xD3,0xD1,0xD7,0xD5,0xCB,0xC9,0xCF,0xCD,0xC3,0xC1,0xC7,0xC5,
0xFB,0xF9,0xFF,0xFD,0xF3,0xF1,0xF7,0xF5,0xEB,0xE9,0xEF,0xED,0xE3,0xE1,0xE7,0xE5
};

rom const unsigned char x2[ ] =
{
0x00,0x04,0x08,0x0C,0x10,0x14,0x18,0x1C,0x20,0x24,0x28,0x2C,0x30,0x34,0x38,0x3C,
0x40,0x44,0x48,0x4C,0x50,0x54,0x58,0x5C,0x60,0x64,0x68,0x6C,0x70,0x74,0x78,0x7C,
0x80,0x84,0x88,0x8C,0x90,0x94,0x98,0x9C,0xA0,0xA4,0xA8,0xAC,0xB0,0xB4,0xB8,0xBC,
0xC0,0xC4,0xC8,0xCC,0xD0,0xD4,0xD8,0xDC,0xE0,0xE4,0xE8,0xEC,0xF0,0xF4,0xF8,0xFC,
0x1B,0x1F,0x13,0x17,0x0B,0x0F,0x03,0x07,0x3B,0x3F,0x33,0x37,0x2B,0x2F,0x23,0x27,
0x5B,0x5F,0x53,0x57,0x4B,0x4F,0x43,0x47,0x7B,0x7F,0x73,0x77,0x6B,0x6F,0x63,0x67,
0x9B,0x9F,0x93,0x97,0x8B,0x8F,0x83,0x87,0xBB,0xBF,0xB3,0xB7,0xAB,0xAF,0xA3,0xA7,
0xDB,0xDF,0xD3,0xD7,0xCB,0xCF,0xC3,0xC7,0xFB,0xFF,0xF3,0xF7,0xEB,0xEF,0xE3,0xE7,
0x36,0x32,0x3E,0x3A,0x26,0x22,0x2E,0x2A,0x16,0x12,0x1E,0x1A,0x06,0x02,0x0E,0x0A,
0x76,0x72,0x7E,0x7A,0x66,0x62,0x6E,0x6A,0x56,0x52,0x5E,0x5A,0x46,0x42,0x4E,0x4A,
0xB6,0xB2,0xBE,0xBA,0xA6,0xA2,0xAE,0xAA,0x96,0x92,0x9E,0x9A,0x86,0x82,0x8E,0x8A,
0xF6,0xF2,0xFE,0xFA,0xE6,0xE2,0xEE,0xEA,0xD6,0xD2,0xDE,0xDA,0xC6,0xC2,0xCE,0xCA,
0x2D,0x29,0x25,0x21,0x3D,0x39,0x35,0x31,0x0D,0x09,0x05,0x01,0x1D,0x19,0x15,0x11,
0x6D,0x69,0x65,0x61,0x7D,0x79,0x75,0x71,0x4D,0x49,0x45,0x41,0x5D,0x59,0x55,0x51,
0xAD,0xA9,0xA5,0xA1,0xBD,0xB9,0xB5,0xB1,0x8D,0x89,0x85,0x81,0x9D,0x99,0x95,0x91,
0xED,0xE9,0xE5,0xE1,0xFD,0xF9,0xF5,0xF1,0xCD,0xC9,0xC5,0xC1,0xDD,0xD9,0xD5,0xD1
};
```

```
rom const unsigned char x3[ ] =
{
0x00,0x08,0x10,0x18,0x20,0x28,0x30,0x38,0x40,0x48,0x50,0x58,0x60,0x68,0x70,0x78,
0x80,0x88,0x90,0x98,0xA0,0xA8,0xB0,0xB8,0xC0,0xC8,0xD0,0xD8,0xE0,0xE8,0xF0,0xF8,
0x1B,0x13,0x0B,0x03,0x3B,0x33,0x2B,0x23,0x5B,0x53,0x4B,0x43,0x7B,0x73,0x6B,0x63,
0x9B,0x93,0x8B,0x83,0xBB,0xB3,0xAB,0xA3,0xDB,0xD3,0xCB,0xC3,0xFB,0xF3,0xEB,0xE3,
0x36,0x3E,0x26,0x2E,0x16,0x1E,0x06,0x0E,0x76,0x7E,0x66,0x6E,0x56,0x5E,0x46,0x4E,
0xB6,0xBE,0xA6,0xAE,0x96,0x9E,0x86,0x8E,0xF6,0xFE,0xE6,0xEE,0xD6,0xDE,0xC6,0xCE,
0x2D,0x25,0x3D,0x35,0x0D,0x05,0x1D,0x15,0x6D,0x65,0x7D,0x75,0x4D,0x45,0x5D,0x55,
0xAD,0xA5,0xBD,0xB5,0x8D,0x85,0x9D,0x95,0xED,0xE5,0xFD,0xF5,0xCD,0xC5,0xDD,0xD5,
0x6C,0x64,0x7C,0x74,0x4C,0x44,0x5C,0x54,0x2C,0x24,0x3C,0x34,0x0C,0x04,0x1C,0x14,
0xEC,0xE4,0xFC,0xF4,0xCC,0xC4,0xDC,0xD4,0xAC,0xA4,0xBC,0xB4,0x8C,0x84,0x9C,0x94,
0x77,0x7F,0x67,0x6F,0x57,0x5F,0x47,0x4F,0x37,0x3F,0x27,0x2F,0x17,0x1F,0x07,0x0F,

0xF7,0xFF,0xE7,0xEF,0xD7,0xDF,0xC7,0xCF,0xB7,0xBF,0xA7,0xAF,0x97,0x9F,0x87,0x8F,
0x5A,0x52,0x4A,0x42,0x7A,0x72,0x6A,0x62,0x1A,0x12,0x0A,0x02,0x3A,0x32,0x2A,0x22,
0xDA,0xD2,0xCA,0xC2,0xFA,0xF2,0xEA,0xE2,0x9A,0x92,0x8A,0x82,0xBA,0xB2,0xAA,0xA2,
0x41,0x49,0x51,0x59,0x61,0x69,0x71,0x79,0x01,0x09,0x11,0x19,0x21,0x29,0x31,0x39,
0xC1,0xC9,0xD1,0xD9,0xE1,0xE9,0xF1,0xF9,0x81,0x89,0x91,0x99,0xA1,0xA9,0xB1,0xB9
};

if(round_counter != 10) //SE FOR DIFERENTE DE 10 FAZ O "DECODE MIX COLUMN"!!!
{

    {
        unsigned char temp0,temp1,temp2,temp3;
        for(i=0;i<16;i+=4)
        {
            temp3=x3[dados[i+0x00]]^x3[dados[i+0x01]]^x3[dados[i+0x02]]^x3[dados[i+0x03]]
            ^dados[i+0x00]^dados[i+0x01]^dados[i+0x02]^dados[i+0x03];

            temp0=x2[dados[i+0x00]]^x1[dados[i+0x00]]^x1[dados[i+0x01]]^x2[dados[i+0x02]]
            ^temp3^dados[i+0x00];

            temp1=x2[dados[i+0x01]]^x1[dados[i+0x01]]^x1[dados[i+0x02]]^x2[dados[i+0x03]]
            ^temp3^dados[i+0x01];

            temp2=x2[dados[i+0x02]]^x1[dados[i+0x02]]^x1[dados[i+0x03]]^x2[dados[i+0x00]]
            ^temp3^dados[i+0x02];


    temp3^=x2[dados[i+0x03]]^x1[dados[i+0x03]]^x1[dados[i+0x00]]^x2[dados[i+0x01]]
            ^dados[i+0x03];

            dados[i+0]=temp0;
            dados[i+1]=temp1;
            dados[i+2]=temp2;
            dados[i+3]=temp3;
        }
    }

`
```

As tabelas *X1*, *X2* e *X3* foram criadas com o intuito de reduzir o cálculo matemático para as regras especiais previamente descritas para este processo. Neste caso, há um compromisso entre o processamento e a capacidade de memória do microcontrolador. Modelos mais velozes são capazes de fazer o processamento desta rotina sem necessitar da utilização de tabelas. Porém, no caso do PIC18F4550 (8 bits), não há outra maneira senão a utilização destas tabelas. Neste caso, é altamente desejável que o modelo possua disponibilidade de memória para o armazenamento das mesmas.

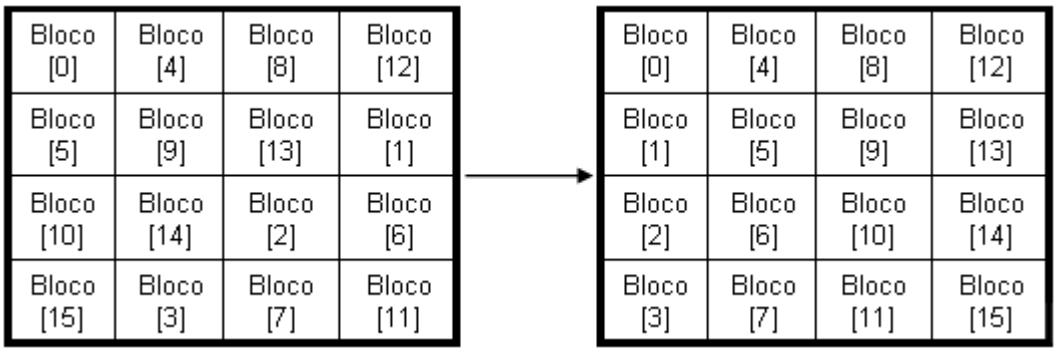
4.1.3.2.3 Permutação de linhas

Neste passo é realizado um processo de rotação cíclica à esquerda, aplicado às linhas da matriz mensagem cifrada [5]. Os números de rotações de cada linha são mostrados na Tabela 11:

Tabela 11. Número de rotações do processo de decifragem AES [28].

	Nº Rotações Linha 0	Nº Rotações Linha 0	Nº Rotações Linha 0	Nº Rotações Linha 0
Mensagem 16 bytes	0	3	2	1

Deste modo, tem-se:



O trecho da programação que realiza esta função é mostrado no Quadro 9:

Quadro 9.: Programação da Permutação de Linhas (decifragem).

```
DecodeShiftRow(dados);                          //Chamada da função de Permutação de Linhas.

void DecodeShiftRow(unsigned char* stateTable)   //Função de Permutação de Linhas.
{
    unsigned char temp;                         //Criação da variável temporária temp.

    temp=stateTable[1];                         //Permutação na segunda linha.
    stateTable[1]=stateTable[13];
    stateTable[13]=stateTable[9];
    stateTable[9]=stateTable[5];
    stateTable[5]=temp;

    temp=stateTable[2];                         //Permutação na terceira linha.
    stateTable[2]=stateTable[10];
    stateTable[10]=temp;
    temp=stateTable[14];
    stateTable[14]=stateTable[6];
    stateTable[6]=temp;
    temp=stateTable[7];                         //Permutação na quarta linha.
    stateTable[7]=stateTable[11];
    stateTable[11]=stateTable[15];
    stateTable[15]=stateTable[3];
    stateTable[3]=temp;
}
```

Neste caso, a substituição dos termos da matriz mensagem também ocorre individualmente, *byte* a *byte*.

4.1.3.2.4 Substituição na Tabela-Si

Neste passo, cada bloco da matriz mensagem cifrada deve ser substituído por seu respectivo valor na Tabela-Si [5]. Como exemplo, se um *Bloco[n]* da matriz mensagem possui armazenado o valor 8Fh (hexadecimal), olhamos a posição 8F na Tabela-Si (x=08; y=F0) e armazenamos o conteúdo desta posição no *Bloco[n]* (no caso, recuperando o valor 73h).

A Fig. 27 ilustra este exemplo com a Tabela-Si, que por sua vez é conhecida e disponibilizada pelo próprio algoritmo de decifragem:

x \ y	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
10	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
20	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
30	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
40	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
50	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
60	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
70	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
80	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
90	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A0	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B0	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C0	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D0	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E0	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F0	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Fig. 27. Tabela-Si [27].

O trecho da programação que realiza esta função é mostrado no Quadro 10:

Quadro 10. Programação da Substituição na Tabela-Si.

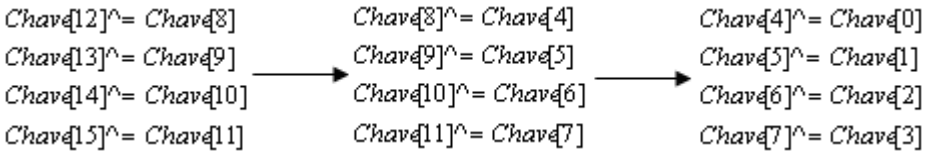
```
#define BLOCKSIZE 16            //Constante do numero de bytes a serem criptografados.

rom const unsigned char SiTable[ ] =
{
0x52,0x09,0x6A,0xD5,0x30,0x36,0xA5,0x38,0xBF,0x40,0xA3,0x9E,0x81,0xF3,0xD7,0xFB,
0x7C,0xE3,0x39,0x82,0x9B,0x2F,0xFF,0x87,0x34,0x8E,0x43,0x44,0xC4,0xDE,0xE9,0xCB,
0x54,0x7B,0x94,0x32,0xA6,0xC2,0x23,0x3D,0xEE,0x4C,0x95,0x0B,0x42,0xFA,0xC3,0x4E,
0x08,0x2E,0xA1,0x66,0x28,0xD9,0x24,0xB2,0x76,0x5B,0xA2,0x49,0x6D,0x8B,0xD1,0x25,
0x72,0xF8,0xF6,0x64,0x86,0x68,0x98,0x16,0xD4,0xA4,0x5C,0xCC,0x5D,0x65,0xB6,0x92,
0x6C,0x70,0x48,0x50,0xFD,0xED,0xB9,0xDA,0x5E,0x15,0x46,0x57,0xA7,0x8D,0x9D,0x84,
0x90,0xD8,0xAB,0x00,0x8C,0xBC,0xD3,0x0A,0xF7,0xE4,0x58,0x05,0xB8,0xB3,0x45,0x06,
0xD0,0x2C,0x1E,0x8F,0xCA,0x3F,0x0F,0x02,0xC1,0xAF,0xBD,0x03,0x01,0x13,0x8A,0x6B,
0x3A,0x91,0x11,0x41,0x4F,0x67,0xDC,0xEA,0x97,0xF2,0xCF,0xCE,0xF0,0xB4,0xE6,0x73,
0x96,0xAC,0x74,0x22,0xE7,0xAD,0x35,0x85,0xE2,0xF9,0x37,0xE8,0x1C,0x75,0xDF,0x6E,
0x47,0xF1,0x1A,0x71,0x1D,0x29,0xC5,0x89,0x6F,0xB7,0x62,0x0E,0xAA,0x18,0xBE,0x1B,
0xFC,0x56,0x3E,0x4B,0xC6,0xD2,0x79,0x20,0x9A,0xDB,0xC0,0xFE,0x78,0xCD,0x5A,0xF4,
0x1F,0xDD,0xA8,0x33,0x88,0x07,0xC7,0x31,0xB1,0x12,0x10,0x59,0x27,0x80,0xEC,0x5F,
0x60,0x51,0x7F,0xA9,0x19,0xB5,0x4A,0x0D,0x2D,0xE5,0x7A,0x9F,0x93,0xC9,0x9C,0xEF,
0xA0,0xE0,0x3B,0x4D,0xAE,0x2A,0xF5,0xB0,0xC8,0xEB,0xBB,0x3C,0x83,0x53,0x99,0x61,
0x17,0x2B,0x04,0x7E,0xBA,0x77,0xD6,0x26,0xE1,0x69,0x14,0x63,0x55,0x21,0x0C,0x7D
};                                          //Componentes da Tabela-S.

for(i=0;i<BLOCKSIZE;i++)
{
    dados[i]=SiTable[dados[i]];             //Substituição na Tabela-S.
}
```

Por medida de segurança, os elementos da matriz da Tabela-Si também foram armazenados no *software* como *constantes*. Como se pode ver, o vetor *dados[n]* é substituído pelo seu próprio valor na Tabela-Si.

4.1.3.2.5 Cálculo da chave

Esta rotina tem por função calcular uma nova chave no processo de decifragem, que será utilizada na iteração seguinte [5]. Procede da seguinte maneira:

- Passo 1: Faz-se uma operação lógica XOR entre as colunas da matriz das chaves, na sequência mostrada abaixo [5]:



- Passo 2: Na matriz das chaves, faz-se uma operação lógica XOR entre a Coluna 0 e o respectivo valor da Coluna 3, na Tabela-S [5]:

$$Chave[0]\wedge = Tabela - S[Chave(12)]$$
$$Chave[1]\wedge = Tabela - S[Chave(13)]$$
$$Chave[2]\wedge = Tabela - S[Chave(14)]$$
$$Chave[3]\wedge = Tabela - S[Chave(15)]$$

- Passo 3: Faz-se uma operação lógica XOR entre o conteúdo de *Chave [0]* e o registrador *Rcon* [5]:

$$Chave[0]\wedge = Rcon$$

- Passo 4: Faz-se uma atualização do valor da variável *Rcon* de acordo com a variável *xtime* [5]:

$$Rcon = xtime(Rcon)$$

O trecho da programação que realiza esta função é mostrado no Quadro 11:

Quadro 11. Programação do Cálculo de Chave (decifragem).

```
AESCalcDecodeKey(AESkey);                    //Chamada da função para calculo da última chave.
DecKeySchedule(AESkey);                      //Chamada da função de cálculo de chave.

void AESCalcDecodeKey(unsigned char* key)    //Função para cálculo da última chave.
{
    Contador_rodadas = 10;

    rcon=0x01;
    do                                       //Faz o cálculo da chave dez vezes, para pegar a última.
    {
        EncKeySchedule(key);
        Contador_rodadas--;
    }
    while(Contador_rodadas >0);
}

void DecKeySchedule(unsigned char* key)      //Função de cálculo de chave.
{
    key[12]^=key[8];                         //Passo 1.
    key[13]^=key[9];
    key[14]^=key[10];
    key[15]^=key[11];

    key[8]^=key[4];
    key[9]^=key[5];
    key[10]^=key[6];
    key[11]^=key[7];

    key[4]^=key[0];
    key[5]^=key[1];
    key[6]^=key[2];
    key[7]^=key[3];

    key[0]^=STable[key[13]];                 //Passo 2.
    key[1]^=STable[key[14]];
    key[2]^=STable[key[15]];
```

```
    key[3]^=STable[key[12]];

    key[0]^=rcon;                          //Passo 3.

    if(rcon &0x01)                         //Passo 4.
    {
        rcon = 0x80;
    }
    else
    {
        rcon >>= 1;
    }
}
```

A rotina para o cálculo de chave no processo de decifragem AES exige intrinsecamente a implementação de outra função, a qual chamamos de *AESCalcDecodeKey* no Quadro 10. Esta função é necessária devido ao fato do processo de decifragem AES utilizar as chaves calculadas de maneira inversa, ou seja, da última para a primeira. Logo, esta função calcula a última chave gerada no processo de cifragem, e então a cada iteração é calculada a chave anterior. Este processo será melhor ilustrado a seguir, na validação deste algoritmo de cifragem.

4.1.3.3 Validação do algoritmo criptográfico AES (Rijndael)

Os valores matriciais contidos nos testes a seguir foram retirados do próprio *software*, e confrontados com duas fontes confiáveis [8] e [29]. A mensagem e a chave de criptografia foram baseadas no Apêndice B (*Cipher Example*) da primeira fonte, onde são demonstradas as dez iterações necessárias para a cifragem de um texto puro. Deste modo, foi considerado para os testes:

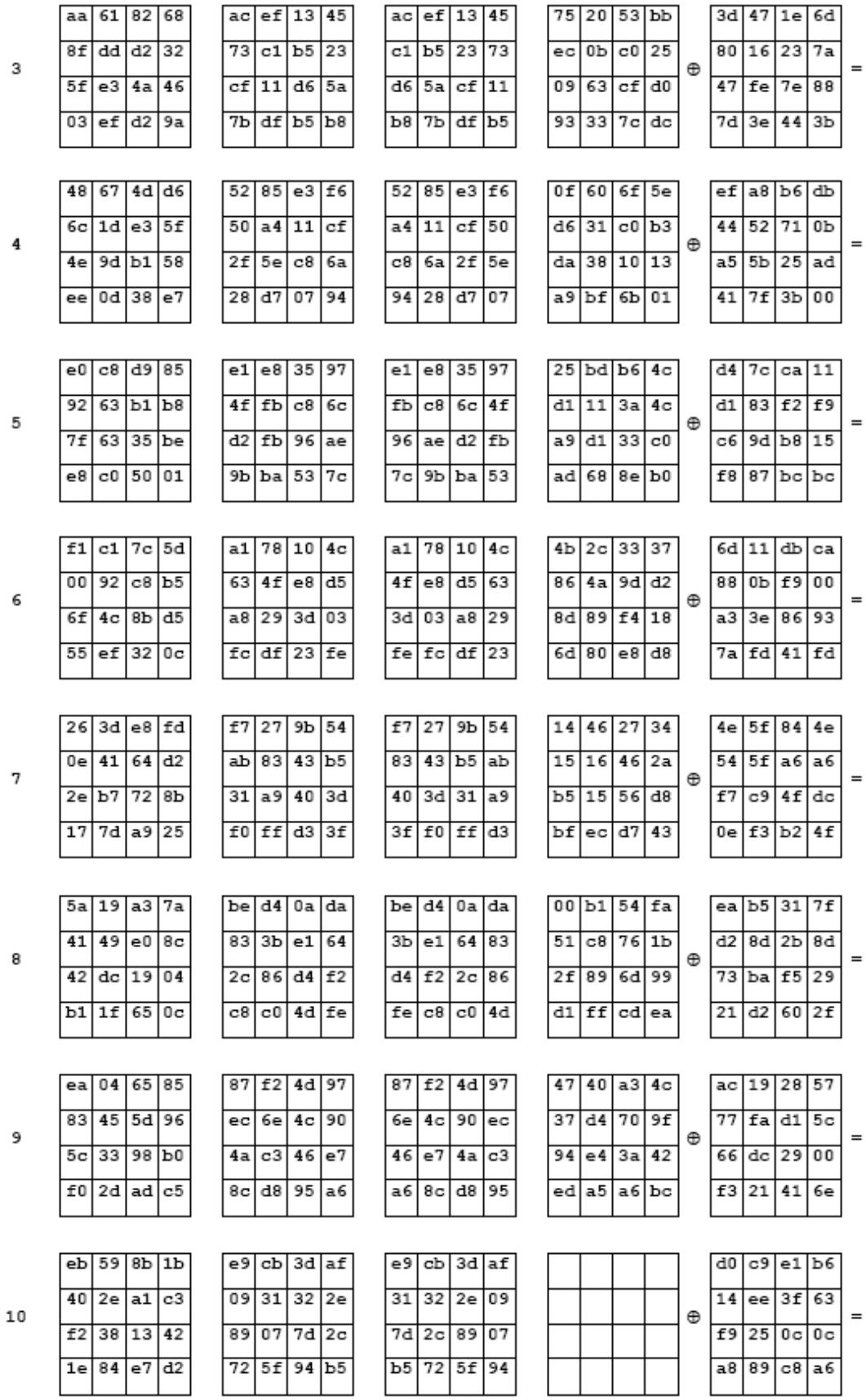
- Texto Puro: 0x3243F6A8885A308D313198A2E0370734h (128 bits).
- Chave Secreta: 0x2B7E151628AED2A6ABF7158809CF4F3Ch (128 bits).

4.1.3.3.1 Teste do algoritmo de cifragem AES (Rijndael)

O teste a seguir pode ser mais bem assimilado se acompanhado ao diagrama em blocos da Fig. 23. Primeiramente, o algoritmo recebe os dados referentes à mensagem e chave de criptografia (ambos de 128 bits). A operação lógica XOR realizada entre ambas as matrizes é executada pela rotina *Adição da Chave*. Então, dá-se início à primeira iteração do processo de criptografia AES (Rijndael). Cada iteração passa respectivamente pela seqüência de rotinas: Substituição na Tabela-S, Permutação de Linhas, e Permutação de Colunas. Ao final de cada iteração é calculada uma nova chave, através da rotina *Cálculo da Chave*, que será utilizada na iteração seguinte. Ao final, tem-se um total de dez chaves as quais foram utilizadas no processo completo da cifragem AES (Rijndael). Cada chave calculada reinicia o laço com uma nova operação lógica XOR realizada entre a matriz chave e a matriz mensagem, estabelecendo novamente a rotina *Adição da Chave*.

Seguem abaixo os dados retirados do teste prático do algoritmo de cifragem AES (Rijndael) [24]:

	Bytes da Mensagem	Substituição Tabela-S	Permutação de Linhas	Permutação de Colunas		Valor das Chaves	
Entrada	32 88 31 e0 43 5a 31 37 f6 30 98 07 a8 8d a2 34				⊕	2b 28 ab 09 7e ae f7 cf 15 d2 15 4f 16 a6 88 3c	=
Iterações							
1	19 a0 9a e9 3d f4 c6 f8 e3 e2 8d 48 be 2b 2a 08	d4 e0 b8 1e 27 bf b4 41 11 98 5d 52 ae f1 e5 30	d4 e0 b8 1e bf b4 41 27 5d 52 11 98 30 ae f1 e5	04 e0 48 28 66 cb f8 06 81 19 d3 26 e5 9a 7a 4c	⊕	a0 88 23 2a fa 54 a3 6c fe 2c 39 76 17 b1 39 05	=
2	a4 68 6b 02 9c 9f 5b 6a 7f 35 ea 50 f2 2b 43 49	49 45 7f 77 de db 39 02 d2 96 87 53 89 f1 1a 3b	49 45 7f 77 db 39 02 de 87 53 d2 96 3b 89 f1 1a	58 1b db 1b 4d 4b e7 6b ca 5a ca b0 f1 ac a8 e5	⊕	f2 7a 59 73 c2 96 35 59 95 b9 80 f6 f2 43 7a 7f	=


3
aa 61 82 68
8f dd d2 32
5f e3 4a 46
03 ef d2 9a
ac ef 13 45
73 c1 b5 23
cf 11 d6 5a
7b df b5 b8
ac ef 13 45
c1 b5 23 73
d6 5a cf 11
b8 7b df b5
75 20 53 bb
ec 0b c0 25
09 63 cf d0
93 33 7c dc
⊕
3d 47 1e 6d
80 16 23 7a
47 fe 7e 88
7d 3e 44 3b
=
4
48 67 4d d6
6c 1d e3 5f
4e 9d b1 58
ee 0d 38 e7
52 85 e3 f6
50 a4 11 cf
2f 5e c8 6a
28 d7 07 94
52 85 e3 f6
a4 11 cf 50
c8 6a 2f 5e
94 28 d7 07
0f 60 6f 5e
d6 31 c0 b3
da 38 10 13
a9 bf 6b 01
⊕
ef a8 b6 db
44 52 71 0b
a5 5b 25 ad
41 7f 3b 00
=
5
e0 c8 d9 85
92 63 b1 b8
7f 63 35 be
e8 c0 50 01
e1 e8 35 97
4f fb c8 6c
d2 fb 96 ae
9b ba 53 7c
e1 e8 35 97
fb c8 6c 4f
96 ae d2 fb
7c 9b ba 53
25 bd b6 4c
d1 11 3a 4c
a9 d1 33 c0
ad 68 8e b0
⊕
d4 7c ca 11
d1 83 f2 f9
c6 9d b8 15
f8 87 bc bc
=
6
f1 c1 7c 5d
00 92 c8 b5
6f 4c 8b d5
55 ef 32 0c
a1 78 10 4c
63 4f e8 d5
a8 29 3d 03
fc df 23 fe
a1 78 10 4c
4f e8 d5 63
3d 03 a8 29
fe fc df 23
4b 2c 33 37
86 4a 9d d2
8d 89 f4 18
6d 80 e8 d8
⊕
6d 11 db ca
88 0b f9 00
a3 3e 86 93
7a fd 41 fd
=
7
26 3d e8 fd
0e 41 64 d2
2e b7 72 8b
17 7d a9 25
f7 27 9b 54
ab 83 43 b5
31 a9 40 3d
f0 ff d3 3f
f7 27 9b 54
83 43 b5 ab
40 3d 31 a9
3f f0 ff d3
14 46 27 34
15 16 46 2a
b5 15 56 d8
bf ec d7 43
⊕
4e 5f 84 4e
54 5f a6 a6
f7 c9 4f dc
0e f3 b2 4f
=
8
5a 19 a3 7a
41 49 e0 8c
42 dc 19 04
b1 1f 65 0c
be d4 0a da
83 3b e1 64
2c 86 d4 f2
c8 c0 4d fe
be d4 0a da
3b e1 64 83
d4 f2 2c 86
fe c8 c0 4d
00 b1 54 fa
51 c8 76 1b
2f 89 6d 99
d1 ff cd ea
⊕
ea b5 31 7f
d2 8d 2b 8d
73 ba f5 29
21 d2 60 2f
=
9
ea 04 65 85
83 45 5d 96
5c 33 98 b0
f0 2d ad c5
87 f2 4d 97
ec 6e 4c 90
4a c3 46 e7
8c d8 95 a6
87 f2 4d 97
6e 4c 90 ec
46 e7 4a c3
a6 8c d8 95
47 40 a3 4c
37 d4 70 9f
94 e4 3a 42
ed a5 a6 bc
⊕
ac 19 28 57
77 fa d1 5c
66 dc 29 00
f3 21 41 6e
=
10
eb 59 8b 1b
40 2e a1 c3
f2 38 13 42
1e 84 e7 d2
e9 cb 3d af
09 31 32 2e
89 07 7d 2c
72 5f 94 b5
e9 cb 3d af
31 32 2e 09
7d 2c 89 07
b5 72 5f 94
⊕
d0 c9 e1 b6
14 ee 3f 63
f9 25 0c 0c
a8 89 c8 a6
=

Saída

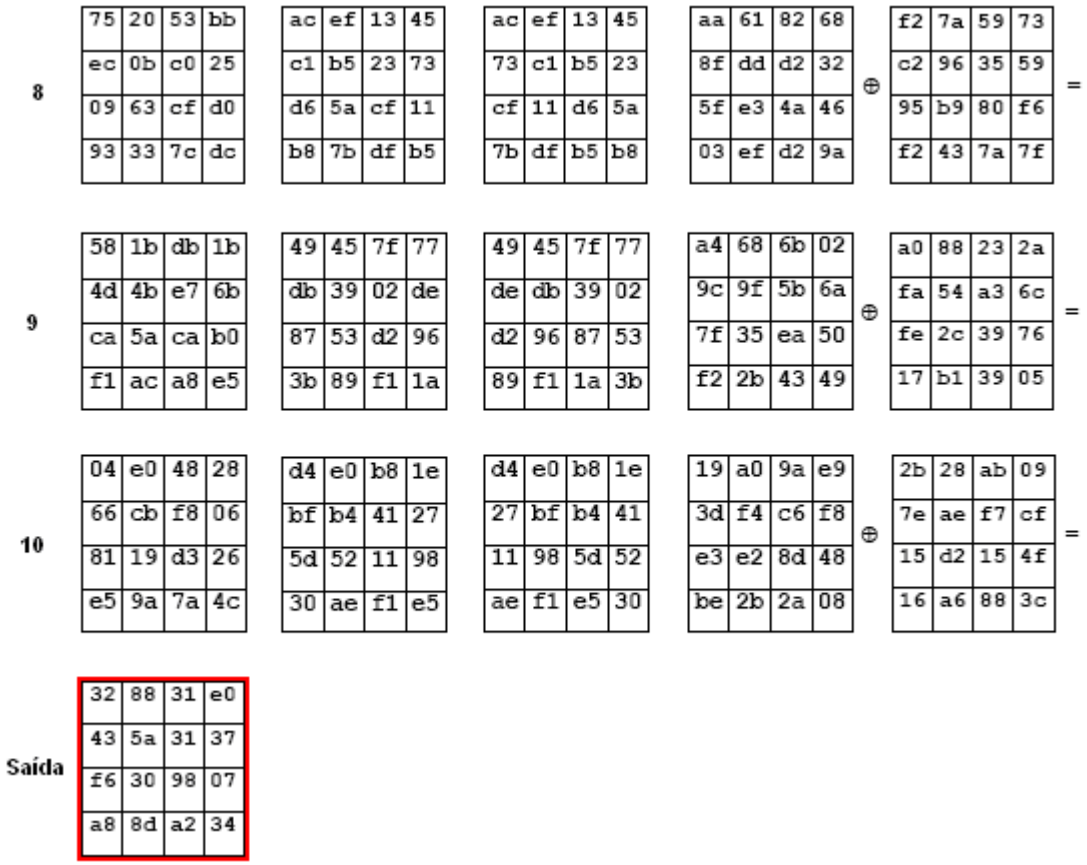
39	02	dc	19
25	dc	11	6a
84	09	85	0b
1d	fb	97	32

Nota-se que na última iteração não é executada a rotina *Permutação de Colunas*, como pode ser verificado no diagramas em blocos da Fig. 23. Ao final deste processo, tem-se a matriz de *Saída* com a mensagem criptografada [24].

4.1.3.3.2 Teste do algoritmo de decifragem AES (Rijndael)

A seguir são mostradas as etapas do processo de decifragem, responsável pela transformação da mensagem criptografada à sua forma original. Primeiramente, o algoritmo recebe os dados criptografados, mantendo a mesma chave secreta que atuou na cifragem dos mesmos (chave simétrica). A operação lógica XOR realizada entre as matrizes de mensagem e chave secreta é executada pela rotina *Adição da Chave*. Uma vez executada esta rotina, dá-se início à primeira iteração do processo de decifragem. Cada iteração passa respectivamente pela seqüência de rotinas: Permutação de Colunas, Permutação de Linhas, e Substituição na Tabela-Si. Analogamente ao processo de cifragem, ao final de cada iteração também é calculada uma nova chave através da rotina *Cálculo da Chave*, que será utilizada na iteração seguinte. Cada chave calculada reinicia o laço com uma nova operação lógica XOR realizada entre a matriz chave e a matriz mensagem, estabelecendo novamente a rotina *Adição da Chave*. Deste modo, seguem abaixo os dados retirados do teste prático do algoritmo de decifragem AES (Rijndael) [24]:

	Bytes da Mensagem				Permutação de Colunas				Permutação de Linhas				Substituição Tabela-Si					Valor das Chaves				
Entrada	39	02	dc	19													⊕	d0	c9	e1	b6	=
	25	dc	11	6a														14	ee	3f	63	
	84	09	85	0b														f9	25	0c	0c	
	1d	fb	97	32														a8	89	c8	a6	
Iterações																						
1	e9	cb	3d	af					e9	cb	3d	af	eb	59	8b	1b	⊕	ac	19	28	57	=
	31	32	2e	09					09	31	32	2e	40	2e	a1	c3		77	fa	d1	5c	
	7d	2c	89	07					89	07	7d	2c	f2	38	13	42		66	dc	29	00	
	b5	72	5f	94					72	5f	94	b5	1e	84	e7	d2		f3	21	41	6e	
2	47	40	a3	4c	87	f2	4d	97	87	f2	4d	97	ea	04	65	85	⊕	ea	b5	31	7f	=
	37	d4	70	9f	6e	4c	90	ec	ec	6e	4c	90	83	45	5d	96		d2	8d	2b	8d	
	94	e4	3a	42	46	e7	4a	c3	4a	c3	46	e7	5c	33	98	b0		73	ba	f5	29	
	ed	a5	a6	bc	a6	8c	d8	95	8c	d8	95	a6	f0	2d	ad	c5		21	d2	60	2f	
3	00	b1	54	fa	be	d4	0a	da	be	d4	0a	da	5a	19	a3	7a	⊕	4e	5f	84	4e	=
	51	c8	76	1b	3b	e1	64	83	83	3b	e1	64	41	49	e0	8c		54	5f	a6	a6	
	2f	89	6d	99	d4	f2	2c	86	2c	86	d4	f2	42	dc	19	04		f7	c9	4f	dc	
	d1	ff	cd	ea	fe	c8	c0	4d	c8	c0	4d	fe	b1	1f	65	0c		0e	f3	b2	4f	
4	14	46	27	34	f7	27	9b	54	f7	27	9b	54	26	3d	e8	fd	⊕	6d	11	db	ca	=
	15	16	46	2a	83	43	b5	ab	ab	83	43	b5	0e	41	64	d2		88	0b	f9	00	
	b5	15	56	d8	40	3d	31	a9	31	a9	40	3d	2e	b7	72	8b		a3	3e	86	93	
	bf	ec	d7	43	3f	f0	ff	d3	f0	ff	d3	3f	17	7d	a9	25		7a	fd	41	fd	
5	4b	2c	33	37	a1	78	10	4c	a1	78	10	4c	f1	c1	7c	5d	⊕	d4	7c	ca	11	=
	86	4a	9d	d2	4f	e8	d5	63	63	4f	e8	d5	00	92	c8	b5		d1	83	f2	f9	
	8d	89	f4	18	3d	03	a8	29	a8	29	3d	03	6f	4c	8b	d5		c6	9d	b8	15	
	6d	80	e8	d8	fe	fc	df	23	fc	df	23	fe	55	ef	32	0c		f8	87	bc	bc	
6	25	bd	b6	4c	e1	e8	35	97	e1	e8	35	97	e0	c8	d9	85	⊕	ef	a8	b6	db	=
	d1	11	3a	4c	fb	c8	6c	4f	4f	fb	c8	6c	92	63	b1	b8		44	52	71	0b	
	a9	d1	33	c0	96	ae	d2	fb	d2	fb	96	ae	7f	63	35	be		a5	5b	25	ad	
	ad	68	8e	b0	7c	9b	ba	53	9b	ba	53	7c	e8	c0	50	01		41	7f	3b	00	
7	0f	60	6f	5e	52	85	e3	f6	52	85	e3	f6	48	67	4d	d6	⊕	3d	47	1e	6d	=
	d6	31	c0	b3	a4	11	cf	50	50	a4	11	cf	6c	1d	e3	5f		80	16	23	7a	
	da	38	10	13	c8	6a	2f	5e	2f	5e	c8	6a	4e	9d	b1	58		47	fe	7e	88	
	a9	bf	6b	01	94	28	d7	07	28	d7	07	94	ee	0d	38	e7		7d	3e	44	3b	



Nota-se que no processo de decifragem AES (Rijndael), a chave inicial é a última chave calculada no processo de cifragem. Deste modo, tem-se na última iteração do algoritmo de decifragem a recuperação do texto puro e da chave secreta inicial, conforme queríamos demonstrar. A rotina *Permutação de Colunas* não é executada na primeira iteração do processo de decifragem, como pode ser verificado no diagrama em blocos da Fig. 25. Ao final deste processo, tem-se a matriz de *Saída* com a mensagem recuperada a sua forma original [24].

4.1.3.3.3 Recepção dos dados após os testes realizados

Neste teste de recepção de dados o microcontrolador PIC18F4550 foi conectado à porta serial de um computador, executando o algoritmo criptográfico AES (Rijndael). O texto puro e a chave secreta foram os mesmos

utilizados para os exemplos anteriores. O computador, neste caso, atuou como se fosse o transmissor UHF ELTA HAL-2, que também recebe os dados em padrão RS-232, armazenando-os em sua memória interna. O *software* utilizado para este teste foi o *FiveChannel V1.0*, capaz de exibir os dados convertidos em formato hexadecimal. A Fig. 28 e a Fig. 29 mostram os resultados dos testes de cifragem e decifragem:

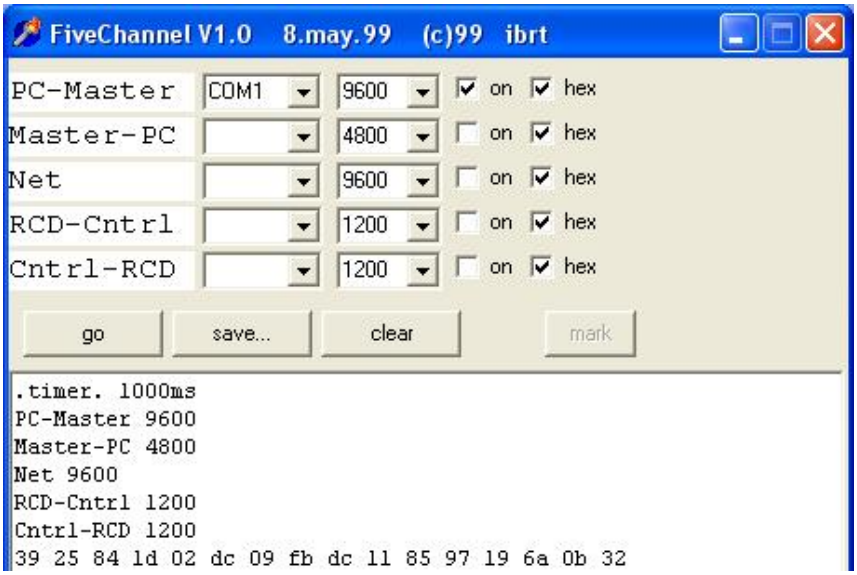


Fig. 28. Dados recebidos após o processo de cifragem AES (Rijndael).

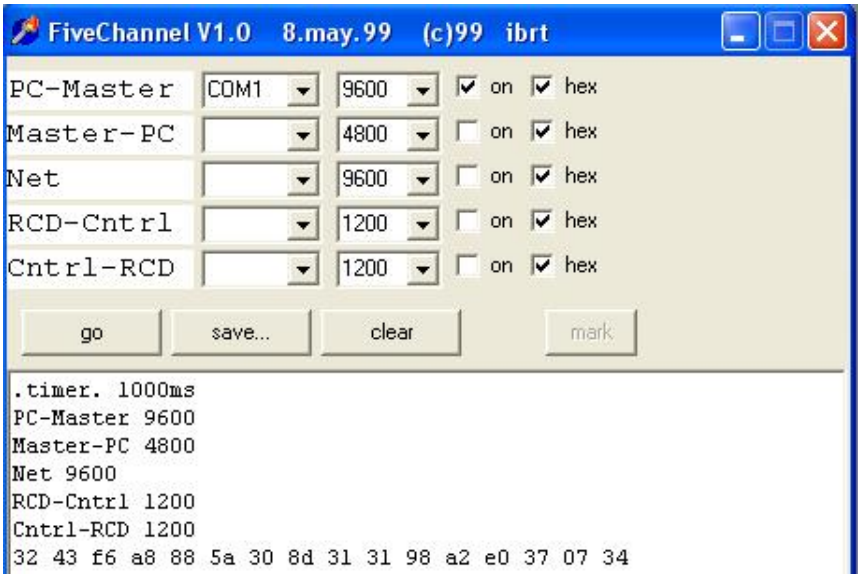


Fig. 29. Dados recebidos após o processo de decifragem AES (Rijndael).

4.1.4 A baliza do SBCDA

A baliza é basicamente o *hardware* que protege o localizador GPS das mais variadas condições climáticas existentes em alto mar, sendo também responsável pela integridade física e funcional das baterias, antenas, transmissor em UHF, entre outros [17]. A Fig. 30 mostra o arranjo prático de uma baliza do sistema Argos, desenvolvida pelo grupo francês *Martec serpe-iesm*:

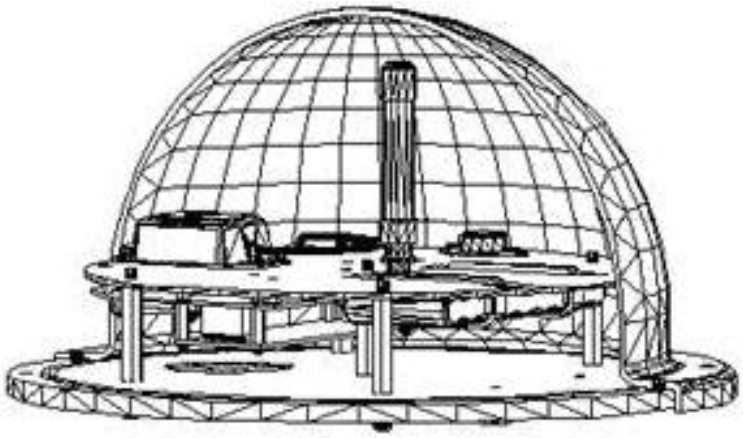


Fig. 30. Baliza do sistema Argos para rastreamento de embarcações [30].

As balizas utilizadas para o monitoramento de embarcações de pesca seguem normas de funcionamento, que no caso do Brasil são determinadas pela Secretaria Especial de Aquicultura e Pesca (SEAP). Dentre as principais características de funcionamento impostas ao localizador GPS destas balizas, estipula-se primeiramente o funcionamento em *hora cheia*. Isto é, se a baliza for inicialmente ligada em fração de hora (p.ex. as 15:37h), esta deve aguardar até a próxima *hora cheia* para realizar a primeira transmissão dos dados (no caso, as 16:00h). Para o caso particular deste projeto, o ciclo de repetição das mensagens é de 90s.

A baliza do SBCDA oferece flexibilidade ao usuário na escolha do período de aquisição da posição geográfica, através de um conjunto de chaves

externas. Contudo, a SEAP recomenda um período de aquisição a cada hora. Também possui externamente um botão de emergência que deve funcionar a qualquer momento, mesmo que a embarcação esteja à deriva, com problemas no motor, sem bateria, e com o sistema de rádio inoperante. O localizador GPS localiza imediatamente a embarcação com uma precisão melhor que 150m, repassando os dados de posição e tempo à Marinha como uma mensagem de socorro. A bateria da baliza oferece uma autonomia de até cinco dias em pleno funcionamento - tempo suficiente para que a embarcação possa ser resgatada em uma situação adversa [17]. A Fig. 31 mostra exemplos de balizas atualmente disponíveis no mercado:

Fig. 31. Balizas disponíveis para o rastreamento de embarcações de pesca.

Como se pode ver, as balizas são completamente seladas por um radome, que no caso particular deste projeto está incluso na própria construção da antena *Synergetics QFH 14A-N* – radome selado de fibra de vidro G10. Porém, ao considerar a possibilidade de uma baliza hermética se faz desejável um projeto térmico de seus módulos internos, de modo a evitar o superaquecimento dos mesmos quando expostos ao ambiente hostil.

4.1.4.1 Características da baliza do SBCDA

O projeto da baliza do SBCDA se baseia nos modelos atualmente disponíveis no mercado, que por sua vez operam com sistemas estrangeiros (p.ex. sistema Argos). A Fig. 32 mostra o protótipo da baliza do SBCDA:

Fig. 32. Protótipo da baliza do SBCDA.

Para esta baliza, são válidas as seguintes características:

- Dimensões: 16cm x 18cm x 46 cm;
- Peso: <5kg;
- Consumo @12V:
    - Modo inativo: <30mA;
    - Modo ativo, em 2W: <680mA.

Os materiais utilizados na construção da baliza do SBCDA foram escolhidos levando em consideração seu peso e durabilidade face ao ambiente hostil de operação da mesma. Deste modo, optou-se por uma caixa de alumínio devido às diversas características favoráveis que este material oferece: leve, resistente, baixo custo, fácil usinagem, bom coeficiente de reflexão de luz, entre outros.

Conforme visto anteriormente, a baliza do SBCDA é programada para um ciclo de transmissão de 90s, onde deve permanecer 89s em modo inativo (<30mA) e 1s em modo ativo (<680mA) – onde ocorre a transmissão dos dados para os satélites do SBCDA. O consumo em modo inativo deve-se ao fato do receptor GPS *Lassen iQ* permanecer constantemente ligado. Esta medida é necessária devido ao sincronismo do *clock* da baliza com o terminal PPS (*Pulse Per Second*) do receptor GPS, o que não é possível se o mesmo não estiver operando plenamente.

4.1.5 Adaptação do *software* multiplataforma

O conceito de *software* multiplataforma apresenta uma aplicação bastante favorável ao localizador GPS, trazendo consigo diversos benefícios principalmente quanto à flexibilidade de funcionamento baseada em diferentes famílias de microcontroladores. Este conceito se torna ainda mais atrativo quando consideramos a possibilidade de se trabalhar com microcontroladores de diferentes plataformas (por exemplo, 8 ou 16 bits). As novas famílias da Microchip PIC24F e PIC24H motivaram a implementação deste conceito, buscando agregar ao *software* um conjunto de vantagens baseado em uma plataforma de 16 bits. Dentre os diversos dispositivos que compõem esta família optou-se pelo possível uso do *PIC24FJ64GA002*, cujo diagrama de pinos é mostrado pela Fig. 33:

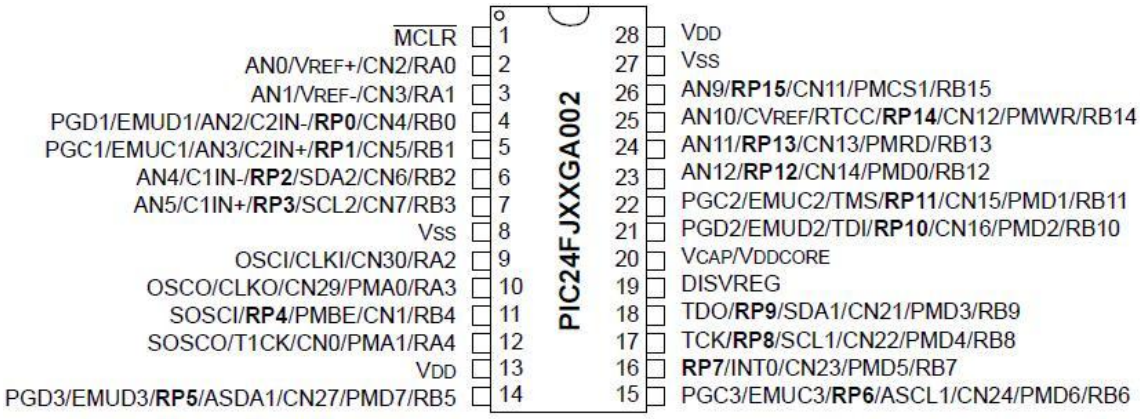


Fig. 33. Diagrama de pinos do microcontrolador PIC24FJ64GA002 [31].

Dentre suas principais características, podemos citar [31]:

- Tensão de operação: 2,0V a 3,6V;
- Tecnologia *nanoWatt*: 0,1μA (modo *sleep*);
- Processamento de 16MIPS (*full speed*);
- Memória de programa: 64kB (*flash*);
- Memória RAM: 8kB;
- WDT estendido: até 131s;
- Sistema de interrupção vetorial;
- Cinco *timers* de 16 bits;
- *Hardware* de multiplicação 17x17 bits;
- Duas UARTS;
- PLL interno;
- Entre outros.

Deste modo, este aprimoramento de *software* buscou ampliar os modos de funcionamento do localizador GPS através da flexibilidade de operação em plataformas de 8 e 16 bits, com isso suportando novas famílias de microcontroladores (p.ex. PIC24F e PIC24H). Contudo, os resultados dos testes que constam neste trabalho foram obtidos com base no

microcontrolador PIC18F4550, de 8 bits, fato o qual foi merecido a este modelo uma abordagem privilegiada.

5 DISCUSSÃO E PERSPECTIVAS

Desde as etapas iniciais de planejamento deste projeto, estudo dos conceitos envolvidos, até por fim o desenvolvimento e montagem de um primeiro protótipo, muitas dificuldades foram superadas por meio de eficazes soluções que permitiram um bom andamento de projeto, coerente com o cronograma de atividades estabelecido.

Deste modo, esta sessão objetiva descrever estas dificuldades em tópicos, justificando oportunamente as soluções encontradas para superá-las. Ainda cabe o estabelecimento das perspectivas que abrem campos para futuros estudos, podendo resultar em novos trabalhos de graduação, iniciação científica, e até teses de mestrado.

5.1 A baliza do SBCDA

Embora verificado o funcionamento da baliza do SBCDA, alguns aspectos ainda são relevantes para futuros estudos e aprimoramentos. Primeiramente, observa-se a possibilidade de acoplamento de sensores de temperatura e profundidade à baliza, posto que estas informações permitem aos pescadores saber os tipos de peixes que podem encontrar em um determinado local. Também se estuda a possibilidade de processamento de diferentes tipos de mensagens pelas balizas, de modo que o operador possa incluir dados particulares além das coordenadas geográficas de sua embarcação.

Por fim, ao considerar a possibilidade de uma baliza completamente hermética, ainda se faz desejável um projeto térmico de seus módulos internos, de modo a evitar o superaquecimento dos mesmos quando expostos ao ambiente marítimo hostil e agressivo.

5.2 Demais aplicações do localizador GPS

Naturalmente, um único protótipo não é capaz de atender a todos os tipos de demanda do mercado, posto que cada aplicação requer maior eficiência em determinados parâmetros. Com base nestas idéias, otimizações de *hardware* e consumo de energia devem ser uma meta para estudos futuros, buscando adaptar o localizador GPS às mais diversas e críticas aplicações.

Com efeito, podemos comparar duas aplicações que utilizariam o mesmo localizador GPS com criptografia de dados: rastreamento de embarcações de pesca, e rastreamento de animais. No primeiro caso, a baliza possui baixa restrição de peso e dimensões, uma vez que ficará disposta em local seguro nas embarcações. Da mesma forma, o consumo de energia também não é um fator crítico, posto que as baterias podem ser facilmente recarregadas pela própria embarcação. Porém, se considerarmos aplicações com alto índice de restrições (p.ex. rastreamento de animais), é altamente desejável que a PCD seja pequena e leve para não interferir no comportamento do mesmo. Neste caso, devem ser estudadas técnicas de minimização de *hardware*, tipos especiais de baterias (possivelmente agregadas a células solares), e componentes de baixíssimo consumo. A vida útil destes transmissores deve ser da ordem de *anos*, posto que o estudo se torna inviável quando requerida a recaptura precoce do animal em estudo. A Fig. 34 mostra exemplos de transmissores estrangeiros dedicados à biotelemetria:

Fig. 34. Transmissores estrangeiros para rastreamento de animais.

Com base nestas idéias e por simplicidade de desenvolvimento de um primeiro protótipo, o foco deste projeto consiste em atender a grandes aplicações, aparentemente sem restrições de tamanho e/ou consumo de energia.

5.3 O *software* multiplataforma

É evidentemente desejável futuros aprimoramentos em busca de ferramentas que ofereçam melhor eficiência com menor consumo de energia. Em termos operacionais, o microcontrolador PIC18F4550 ofereceu certa resistência de funcionamento ao ser submetido a uma freqüência de operação (*clock*) de 4MHz, tornando-se ineficiente e lento no processamento do algoritmo de criptografia AES (Rijndael). Posto que a elevação da freqüência de operação influi diretamente no consumo de energia do microcontrolador, foi estabelecido assim um *clock* de 10MHz, elevando suavemente seu consumo de energia e oferecendo melhor eficiência no processamento dos dados. A memória do microcontrolador se demonstrou suficiente, embora utilizada próxima ao seu limite nominal.

Com a adaptação do *software* para multiplataforma foi possível incrementar a velocidade de processamento através de um microcontrolador de 16 bits (p.ex. *PIC24FJ64GA004*), oferecendo também uma maior capacidade de memória e menor consumo de energia. Um microcontrolador mais veloz amplia a gama de aplicações do localizador GPS, possibilitando suprir novas demandas de mercado que requerem telemetrias mais sofisticadas, como: aviões, sondas espaciais, entre outros.

5.4 A fonte de sincronismo do localizador GPS

Previamente, foram adotadas duas possíveis fontes de sincronismo de *clock* para o localizador GPS: proveniente do receptor GPS *Lassen iQ*, ou por

meio de um oscilador externo (cristal) acoplado ao microcontrolador PIC18F4550.

A primeira opção utiliza como base de tempo o relógio atômico da constelação GPS, garantindo 1s com excelente precisão (da ordem de 10^{-11} s) por meio do terminal PPS (*Pulse-Per-Second*) do receptor GPS *Lassen iQ*. Além de manter o almanaque do receptor GPS sempre atualizado, oferece insignificante desvio de tempo (*delay*) entre as mensagens transmitidas. Por outro lado, o receptor deve permanecer constantemente ligado implicando em um consumo de energia desnecessário, posto que este só seja requerido no momento da aquisição da posição geográfica. Por meio da segunda opção, o receptor GPS permaneceria desligado durante a maior parte do tempo, e a base de tempo do sistema seria fornecida por um cristal externo de baixa freqüência (32,768KHz), acoplado ao microcontrolador PIC18F4550 como mostra a Fig. 35:

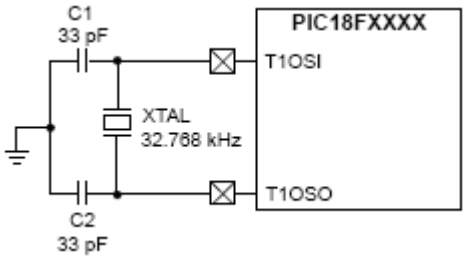


Fig. 35. Cristal externo de 32,768kHz acoplado ao PIC18F4550 [19].

Este cristal é largamente utilizado em projetos eletrônicos onde não se requer uma base de tempo bastante precisa, dada sua sensibilidade às variações de temperatura e/ou impedância intrínseca, apresentando ciclos instáveis.

Neste sentido, é desejável que uma aplicação de monitoramento ofereça uma boa precisão de tempo sujeita a quaisquer condições climáticas para não gerar atrasos acumulativos nas mensagens transmitidas. Por esta razão, o

localizador GPS utiliza como fonte de sincronismo o próprio receptor GPS *Lassen iQ*, pois para esta aplicação se torna mais viável uma melhor precisão de *clock* a uma pequena economia de consumo.

5.5 Criptografia em blocos menores de mensagem

Conforme visto anteriormente, sobre os 160 bits que compõem o campo de mensagem *Header 0* aplica-se o algoritmo criptográfico AES (Rijndael), que por sua vez trabalha com blocos de 128 bits. Logo, se faz necessário quebrar o campo de mensagem em dois blocos menores: 128 bits e 32 bits. O primeiro bloco é composto por uma posição absoluta e duas posições relativas, podendo ser cifrado corretamente pelo algoritmo AES (Rijndael). Porém, o segundo bloco contém somente a terceira posição relativa e é muito reduzido para a aplicação do algoritmo de cifragem. Como não se pode desprezar a terceira posição relativa, novos estudos criptográficos se fazem necessários a fim de se obter criptografia em blocos menores de mensagem, de modo a não invalidar o método criptográfico AES (Rijndael) a esta aplicação. Uma alternativa bastante válida consiste em aproveitar os 96 bits restantes do segundo bloco como mecanismos de redundância, de modo a oferecer melhor qualidade na recepção das mensagens transmitidas pela baliza do SBCDA. Neste sentido, também se faz necessário estudos sobre algoritmos de codificação, como abordado a seguir.

5.6 O algoritmo de codificação/decodificação

Esta sessão destina-se à superficial abordagem da necessidade de implementação de um algoritmo de codificação/decodificação para os dados transmitidos pela baliza do SBCDA.

A codificação é um processo tão desejável quanto a criptografia quando se envolvem tecnologias espaciais. A criptografia é responsável por oferecer segurança e sigilo aos dados transmitidos por uma determinada fonte. Já a

codificação é uma área do desenvolvimento de *software* que busca melhorar a qualidade na recepção dos dados. Existem diversos algoritmos de codificação que cumprem com êxito tal finalidade, se baseando na inserção de bits de redundância na mensagem. Logo, se estabelece um compromisso entre a qualidade que se deseja obter na recepção (medida em BER - *Bit Error Rate*) e o comprimento da mensagem, pois quanto menor a taxa de BER que se deseja obter maior deve ser a quantidade de bits de redundância.

Algoritmos criptográficos operando sem codificação podem ser até prejudiciais ao sistema de comunicação, posto que um único bit recebido erroneamente irá interagir com toda a mensagem no processo de decifragem, ocasionando uma perda substancial de conteúdo. Com efeito, nota-se que a codificação e a criptografia são mecanismos complementares.

Considera-se como opção viável a esta aplicação uma nova classe de códigos concatenados, denominada *Códigos Turbo* (*Turbo Codes*). Este poderoso código de correção de erros é recomendado pelo CCSDS (*Consultative Committee for Space Data System*) para a telemetria espacial, pois combina alta eficiência com facilidade de decodificação – fato que o torna concorrente direto dos códigos LDPC (*Low Dense Parity Check*). Os *Códigos Turbo* podem proporcionar melhorias de até 1dB quando comparados aos demais códigos atualmente utilizados, e sua performance é próxima do *ótimo* (limite de Shannon) [32].

Contudo, outros algoritmos de codificação devem ser estudados para esta aplicação, buscando o de melhor adaptação face ao comprimento da mensagem e o tempo de processamento exigido.

6 CONCLUSÃO

Os estudos realizados e os conceitos adquiridos face às dificuldades encontradas em cada etapa deste projeto contribuíram primeiramente para a ampliação do conhecimento na área de sistemas de telecomunicações, em especial a comunicação digital, satélites e sistema GPS. O conhecimento teórico destas ferramentas pode ser verificado na prática por meio das comunicações realizadas com os satélites do Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA).

O localizador GPS pode ser verificado por meio de simulações, onde foram obtidos dados de posição geográfica com precisão melhor que 150m. A criptografia é um recurso de aplicação potencial da baliza do SBCDA, capaz de oferecer ao usuário segurança nos dados transmitidos por suas embarcações. Excluso o enfoque matemático, o domínio praticável de um método de criptografia é de grande valia, pois possibilita futuramente o desenvolvimento de novas aplicações no segmento de transmissão segura de dados, oferecendo confiança e integridade às mensagens transmitidas.

A implementação do algoritmo criptográfico AES (Rijndael) pode ser verificada com base em uma bibliografia diversificada e confiável, permitindo afirmar que as mensagens recebidas pelos satélites do SBCDA foram corretamente decifradas. A adaptação do *software* para multiplataforma oferece diversos benefícios à baliza do SBCDA, principalmente pela flexibilidade de funcionamento baseada em diferentes famílias de microcontroladores – PIC18F, PIC24F e PIC24H.

O *hardware* da baliza foi projetado para operar em ambientes agressivos e hostis (p.ex. ambiente marítimo), constituindo uma baliza completamente hermética. Foi utilizada uma caixa de alumínio devido às diversas

características favoráveis que este material oferece: leve, resistente, baixo custo, fácil usinagem, bom coeficiente de reflexão de luz, entre outros.

No futuro, ainda deseja-se aprofundar em novos estudos criptográficos a fim de se obter criptografia em blocos menores de mensagem. Da mesma forma, observa-se a viabilidade de implementação de um algoritmo de codificação/decodificação (p. ex. *Turbo Codes*). Otimizações no consumo de energia, dimensões e peso do localizador GPS também são altamente desejáveis, buscando adaptá-lo às mais variadas e críticas aplicações (p.ex. rastreamento de animais).

7 REFERÊNCIAS

[1]SKLAR, B. **Digital communications:** fundamentals and applications. 2. ed. New Jersey: Prentice Hall PTR, 2001. 1078p.

[2]YAMAGUTI, W. et al. **O Sistema Brasileiro de Coleta de Dados Ambientais:** estado atual, demandas e estudos de propostas de continuidade da missão de coleta de dados. INPE, 2006. (SCD-ETD-002).

[3]Secretaria Especial de Aquicultura e Pesca (SEAP). **Instrução normativa interministerial nº2.** Brasília, 2006. 37p.

[4]Secretaria Especial de Aquicultura e Pesca (SEAP). **Instrução normativa interministerial nº22.** Brasília, 2007. 3p.

[5]DAEMEN, J.; RIJMEN, V. **The design of Rijndael:** AES – The Advanced Encryption Standard. 1. ed. New York: Springer-Verlag, 2002. 255p.

[6]KATZAN JR, H. **The standard data encryption algorithm.** 1. ed. New York: PBI, 1977. 134p.

[7]LUCCHESI, C. L. **Introdução à criptografia computacional.** 1. ed. Campinas: Editora da UNICAMP, 1986. 132p.

[8]FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION 197,1., 2001. **Announcing the Advanced Encryption Standard (AES).** FIPS, 2001. 52 p.

[9]KOMATSU, E. M.; KUCINSKIS, F. N.; OTA, P. H.; PADOVANI NETO, V. **Loc-B:** sistema de localização de bóias oceânicas. Relatório de Projeto de Graduação, Universidade de Mogi das Cruzes, 2003. 29p.

[10]YAMAGUTI, W.; ORLANDO, V.; PEREIRA, S. P. Sistema Brasileiro de Coleta de Dados Ambientais: status e planos futuros**.** In: SIMPÓSIO

BRASILEIRO DE SENSORIAMENTO REMOTO, 14. (SBSR), 2009, Natal. **Anais...** São José dos Campos: INPE, 2009. p.1633-1640. DVD, On-line. ISBN 978-85-17-00044-7. Disponível em: <http://marte.dpi.inpe.br/col/dpi.inpe.br/sbsr%4080/2008/11.17.21.20.46/doc/1633-1640.pdf>. Acesso em: 21 Ago. 2009.

[11]TUDE, E. A. P. et al. **Análise do sistema de coleta de dados da MECB/SS.** INPE, São José dos Campos, 1986. INPE-3820-NTE/253.

[12]SILVA, A. B. C.; PEREIRA, W. N. A.; YAMAGUTI, W. Sistema de rastreamento de embarcações de pesca por satélites brasileiros. In: SIMPÓSIO INTERNACIONAL DE INICIAÇÃO CIENTÍFICA DA UNIVERSIDADE DE SÃO PAULO (SIICUSP), 17, 2009, São Carlos. **Anais...** USP, 2009.

[13]SOUZA, C. T.; KUGA, H. K. **Software de localização de plataformas de coleta de dados.** INPE, São José dos Campos, 2005.

[14]BERNARDI, J. V. E.; LANDIM, P. M. B. **Aplicação do Sistema de Posicionamento Global (GPS) na coleta de dados.** DGA, IGCE, UNESP/Rio Claro, Lab. Geomatemática, Texto Didático 10, 31 pp. 2002.

[15]TRIMBLE. **GPS tutorial.** Disponível em: <http://www.trimble.com/gps/index.shtml>. Acesso em: 23 Out. 2009.

[16]OPSCOM40. **Conjunto de apresentações da 40ª reunião do comitê de operações do sistema Argos**. 2006.

[17]SILVA, A. B. C.; YAMAGUTI, W; PEREIRA, W. N. A. Sistema de rastreamento de embarcações de pesca por satélites brasileiros com criptografia de dados. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES (SBrT), 27, 2009, Blumenau. **Anais...** Sociedade Brasileira de Telecomunicações (SBT), 2009.

[18]TRIMBLE. **Lassen iQ GPS receiver:** system designer reference manual. 2005. 268p. Datasheet. Disponível em: <http://trl.trimble.com/docushare/dsweb/Get/Document-338501/Lassen iQ_Reference Manual_Rev B_April 2005.pdf>. Acesso em: 28 Out. 2009.

[19]MICROCHIP TECHNOLOGY INC. **PIC18F2455/2550/4455/4550.** 2009. 438p. Datasheet. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>. Acesso em: 28 Out. 2009.

[20]ELTA. **HAL 2 Argos / SCD:** installation and software manual. 2004. 17p. Datasheet.

[21]ELTA. **HAL 2:** localization & data collection Argos II and SCD UHF transmitter. 2009. 2p. Datasheet. Disponível em: <http://www.elta.fr/uk_doc/HAL2_eng.pdf>. Acesso em: 28 Out. 2009.

[22]SYNERGETICS INTERNATIONAL. **Model 14A-N and 14A-TNC half-wave quadrafilar helix antennas right-hand circular polarization.** Datasheet. Disponível em: <http://www.harshenviro.com/index_files/Page2315.html>. Acesso em: 28 Ago. 2009.

[23]UNICOBA. **Unipower série UP:** bateria chumbo-ácida selada regulada por válvula. Manual Técnico. Disponível em: <http://www.dee.ufc.br/~demercil/Pesquisa/GERAR1000/Docs/Baterias/Manual%20T%82cnico%20Bateria%20Unipower%20%20UP12xxx%20UP6xx%20e%20UP2xx%20cili.pdf>. Acesso em: 28 Out. 2009.

[24]SILVA, A. B. C.; YAMAGUTI, W; PEREIRA, W. N. A. Localizador GPS com criptografia de dados. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO (SBSR), 14, 2009, Natal. **Anais...** São José dos Campos: INPE, 2009. p. 1617-1624. DVD, On-line. ISBN 978-85-17-00044-7. Disponível em:

<http://marte.dpi.inpe.br/col/dpi.inpe.br/sbsr%4080/2008/11.17.14.36/doc/1617-1624.pdf>. Acesso em: 21 Ago. 2009.

[25]CARVALHO, D. B. **Segurança de dados com criptografia:** métodos e algoritmos. 2. ed. Rio de Janeiro: Book Express, 2001. 215p.

[26]FLOWERS, D.; SCHLUNDER, H. **Data encryption routines for PIC24 and dsPIC devices.** 2006. 18p. Application Note. Disponível em: <http://ww1.microchip.com/downloads/en/AppNotes/01044a.pdf>. Acesso em: 06 Ago. 2009.

[27]GUBEL, C. **Advanced Encryption Standard using the PIC16xxx.** 2002. 22p. Application Note. Disponível em: <http://ww1.microchip.com/downloads/en/AppNotes/00821a.pdf>. Acesso em: 06 Ago. 2009.

[28]FLOWERS, D. **Data encryption routines for the PIC18.** 2005. 34p. Application Note. Disponível em: <http://ww1.microchip.com/downloads/en/AppNotes/00953a.pdf>. Acesso em: 14 Jan. 2005.

[29]PERMADI, E. **Cryptography made easy:** Rijndael simulator. Disponível em: <http://jsnerd.googlepages.com/index01a.htm> Acesso em: 5 Maio 2008.

[30]GROUPE MARTEC ANGLAIS. **Baliza do sistema Argos para rastreamento de embarcações.** Disponível em: <http://en.martec.fr>. Acesso em: 31 Maio 2005.

[31]MICROCHIP TECHNOLOGY INC. **PIC24FJ64GA004 family.** 2008. 262p. Datasheet. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/39881c.pdf>. Acesso em: 04 Nov. 2009.

[32]POLLARA, F.; DOLINAR, S.; DIVSALAR, D. **Turbo codes and space communications.** Califórnia: JPL, 2004. 8p.

[33]MICROCHIP TECHNOLOGY INC. **MPLAB® C18 C compiler libraries.** 2005. 184p. Datasheet. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB_C18_Libraries_51297f.pdf>. Acesso em: 27 Out. 2009.

[34]LABTOOLS. **Guia do usuário ICD2br – in-circuit debugger.** 2009. 62p. Datasheet. Disponível em: <http://www.labtools.com.br/arquivos/manual ICD2-BR-2007.out_rev._09.pdf>. Acesso em: 27 Out. 2009.

APÊNDICE A – O SISTEMA GEODÉSICO MUNDIAL *WGS84*

As aplicações com satélites e pesquisas científicas são baseadas na geometria da Terra, que por sua vez não é uma esfera perfeita. A necessidade de se criar um sistema de coordenadas para se ter como base nas aplicações espaciais motivou o desenvolvimento de modelos para tentar descrever a geometria terrestre, da maneira mais próxima do real possível.

Deste modo, em 1960 surgiu o primeiro Sistema Geodésico Mundial – o WGS60 (*World Geodetic System*, de 1960). Foi o primeiro modelo de representação matemática da Terra, utilizado para determinar mais precisamente as posições na superfície do globo. Novas modificações foram realizadas nos anos seguintes, originando sistemas mais atualizados, como: WGS66, WGS72 e WGS84. O último tem sofrido algumas melhorias, embora sua base tenha permanecido inalterada. É atualmente utilizado pelo GPS em operações de satélites, determinação de efemérides radio-difundidas, e cálculo convencional de coordenadas (*datum*).

O elipsóide WGS84 é um sistema ortogonal tridimensional de coordenadas geocêntricas simples, como mostra a Fig. 36:

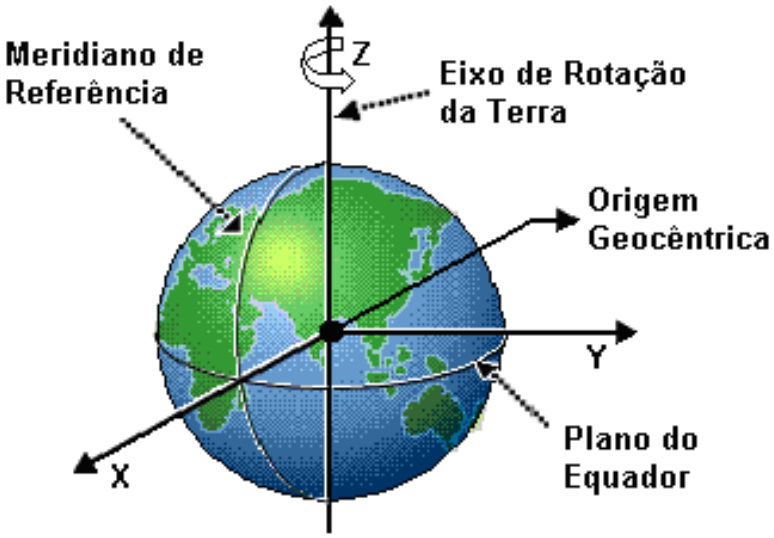


Fig. 36. Sistema de coordenadas do WGS84.

As coordenadas *X* e *Y* no plano do Equador, e *Z* no eixo de rotação da Terra, podem referenciar qualquer ponto sobre a superfície do planeta em termos de longitude, latitude, e altitude, respectivamente.

A latitude e longitude são representadas por uma subdivisão do geóide em graus, sendo estes subdivididos em minutos e segundos. Os graus de latitude vão de -90º (sul) a +90º (norte), tendo como referência a linha do Equador. Os graus de longitude vão de -180° (oeste) a +180° (leste), como mostra a Fig. 37:

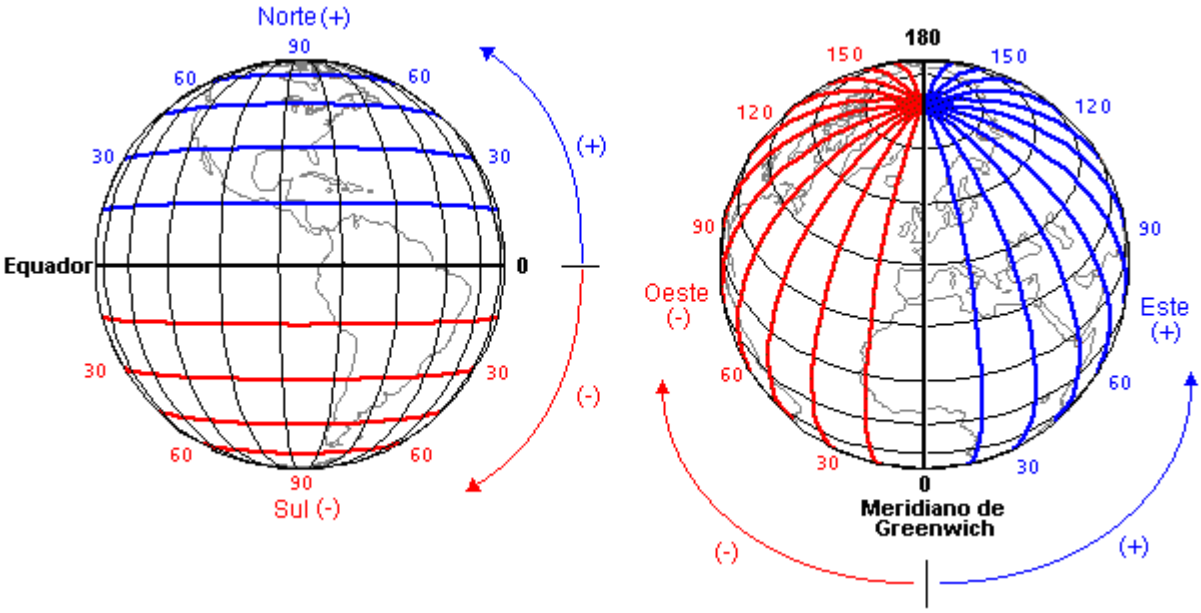


Fig. 37. Latitude (esq.) e longitude (dir.) da Terra.

Neste sistema de coordenadas, considera-se como comprimento médio da latitude aproximadamente 111km/grau. Porém, se considerarmos o achatamento da superfície terrestre nos pólos, o comprimento real da latitude pode sofrer algumas variações em função de algumas faixas, como mostra a Tabela 12:

Tabela 12. Comprimento real da latitude em função da latitude [9].

Faixa de Latitude (graus)	1° de Latitude (km)
0 - 1	110,567
39 - 40	111,023
89 - 90	111,699

No caso da longitude, como os meridianos convergem para os pólos, as variações de cada grau podem sofrer variações de 0km (nos pólos) a 111,321km (na linha do Equador), como mostra a Tabela 13:

Tabela 13. Comprimento real da longitude em função da latitude[9].

Latitude (graus)	1° de Longitude (km)
0	111,321
45	78,849
90	0

Por fim, a altitude não toma como referência o nível do mar, posto que este varia ao longo da superfície terrestre. Neste caso, entende-se como *altitude zero* a superfície do geóide WGS84, e por isso, alguns pontos da superfície real da Terra podem possuir altitudes “negativas” neste sistema de coordenadas.

APÊNDICE B – FUNCIONAMENTO DO PIC18F4550

Este apêndice tem por objetivo substituir o extenso *datasheet* do microcontrolador PIC18F4550, considerando somente os conceitos essenciais envolvidos na concepção deste projeto. Serão abordados conceitos básicos da arquitetura interna da Família PIC18F, módulos de comunicação serial EUSART e UART, e interrupção externa.

Maiores detalhes sobre a manipulação dos registradores e funções da EUSART e UART, bem como dos demais periféricos internos do PIC, podem ser encontrados nas referências bibliográficas utilizadas [19] e [33].

B.1 A família PIC18F

Considerada a grande sucessora da família PIC16F, esta nova família *High End* da *Microchip* possibilita desenvolvimentos mais complexos que exigem uma capacidade de memória maior. Dentre suas principais características se destaca a total compatibilidade com os microcontroladores das famílias anteriores (p.ex. PIC16F), possibilitando a migração sem grandes dificuldades.

Uma das evoluções desta família se encontra no acesso a memória de programa que agora é feito de forma linear, não existindo a necessidade de seleção de página. Isto é possível devido à capacidade estendida dos bancos de memória RAM desta nova família (256 bytes). O PIC16F877A, por exemplo, possui 367 bytes divididos em quatro bancos.

A arquitetura interna da família PIC18F é do tipo *Harvard,* ou seja, possui dois barramentos distintos para memória de programa e dados, com um tamanho de 14 bits e 8 bits, respectivamente. O fluxo de instruções é do tipo *Pipeline*, ou seja, enquanto uma instrução está sendo executada, a seguinte já está sendo lida. A Fig. 38 mostra a arquitetura interna do PIC18F4550:

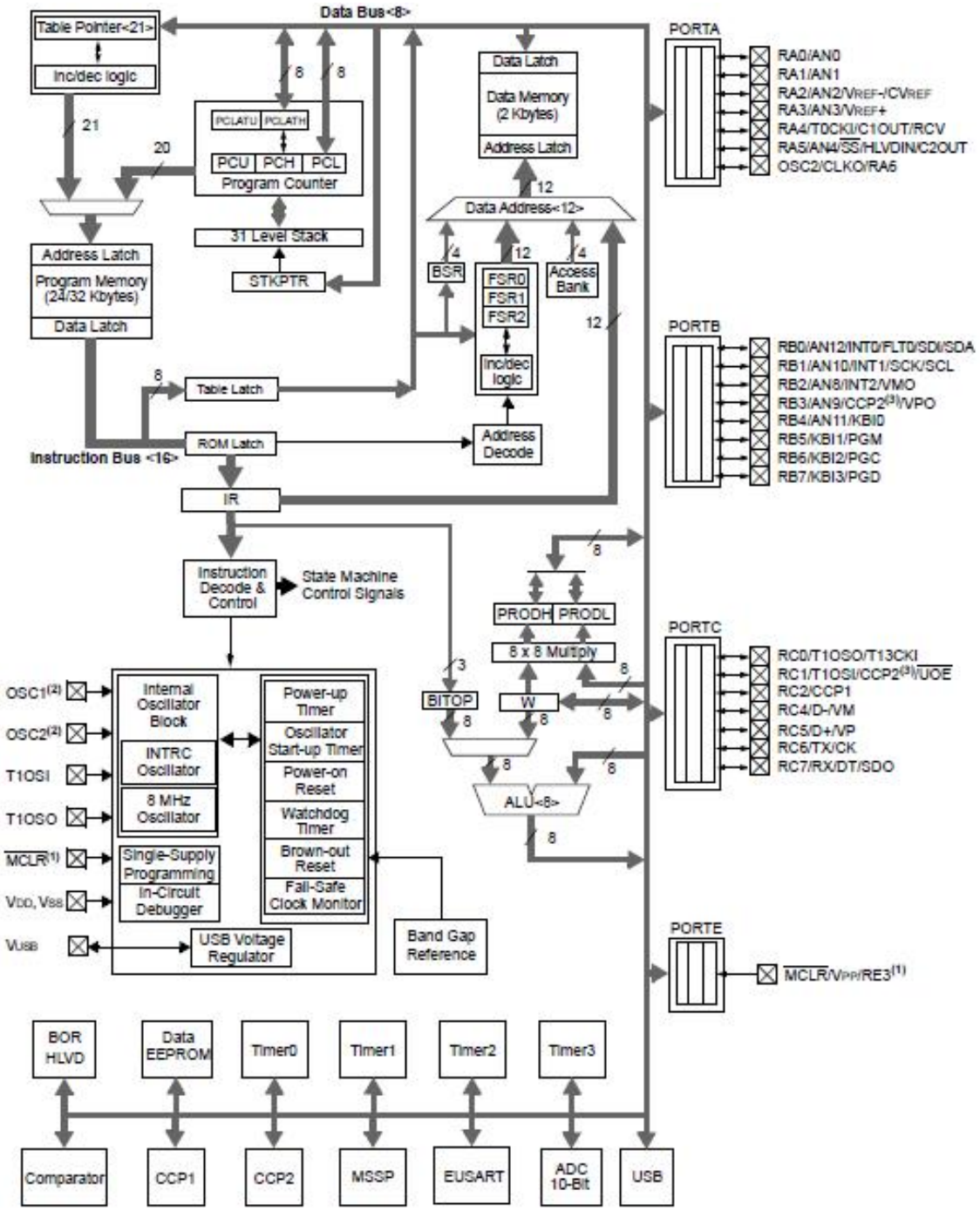


Fig. 38. Arquitetura interna do PIC18F4550 [19].

Como se pode ver, a teoria que envolve os diversos recursos da família PIC18F é bastante extensa, pois estes microcontroladores já podem vir incorporados a uma ampla gama de módulos internos, tais como: interface USB 2.0, pilhas para protocolos ECAN e Ethernet, controles de LCD e motor,

entre outros. Por outro lado, são microcontroladores do tipo RISC (*Reduced Instruction Set Computer*), isto é, possuem um *set* de instruções reduzido – trinta e cinco apenas. Por fim, esta família possui um compilador dedicado (MPLAB C18), cuja programação é realizada em linguagem C.

B.2 Módulo de comunicação serial EUSART

A EUSART (*Enhanced Universal Synchronous Asynchronous Receiver Transmitter*) é uma interface de comunicação serial por *hardware* bastante comum entre os microcontroladores PIC. Como o próprio nome diz, é capaz de trabalhar em dois modos de funcionamento: síncrono e assíncrono. Para o caso especial deste projeto foi utilizada a comunicação no modo assíncrono, *Full Duplex*. Logo, são utilizadas duas vias de dados: transmissão (*Tx*) e recepção (*Rx*), possibilitando tráfego simultâneo. Os registradores responsáveis por estas vias são mostrados na Fig. 39 e Fig. 40, respectivamente:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

Fig. 39. Registrador TXSTA: controle e *status* da transmissão [19].

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Fig. 40. Registrador RCSTA: controle e *status* da recepção [19].

B.3 Módulo de comunicação serial UART

A UART (*Universal Asynchronous Receiver Transmitter*) é uma interface de comunicação serial por *software* que segue o mesmo princípio de funcionamento da USART, porém opera somente em modo assíncrono e não possui registradores de controle. Sua implementação pode ser feita por um

terminal qualquer de Entrada/Saída do PIC – para o caso especial deste projeto foram utilizados os terminais RB4 (*Tx*), e RB5 (*Rx*). A UART requer a definição de algumas funções de *delay*, responsáveis pela taxa de transmissão (*baud rate*) e tempo de amostragem do bit, conforme mostra a Tabela 14:

Tabela 14. Funções de *delay* da UART [33].

Função	Cálculo (ciclos de máquina)
DelayTXBitUART	((((2*Fosc) / (4*Baud)) + 1) / 2) - 12
DelayRXHalfBitUART	((((2*Fosc) / (8*Baud)) + 1) / 2) - 9
DelayRXBitUART	((((2*Fosc) / (4*Baud)) + 1) / 2) - 14

Onde,

Fosc = Frequência do *clock* principal;
Baud = Taxa de transmissão desejada.

O Quadro 11 mostra um exemplo de cálculo para as funções de *delay* da UART, onde se considera uma frequência de *clock* principal igual a 10MHz, e um *baud rate* desejado de 9600bps:

Quadro 12. Exemplo de cálculo para as funções de *delay* da UART.

DelayTXBitUART:
((((2*Fosc)/(4*Baud))+1)/2)-12
((((2*10000000)/(4*9600))+1)/2)-12 = 249 ciclos.

DelayRXHalfBitUART:
((((2*Fosc)/(8*Baud))+1)/2)-9
((((2*10000000)/(8*9600))+1)/2)-9 = 122 ciclos.

DelayRXBitUART:
((((2*Fosc)/(4*Baud))+1)/2)-12
((((2*10000000)/(4*9600))+1)/2)-14 = 247 ciclos.

Os *delays* necessários para gerar o *baud rate* de transmissão e recepção da UART são aproximadamente iguais. A ligeira diferença entre ambos é

compensada pelo tempo de execução de suas próprias funções. Já o *delay* de amostragem de bit é aproximadamente a metade dos demais, posto que o melhor ponto de detecção é exatamente no meio do bit, onde a probabilidade de erro é menor.

O Quadro 13 mostra um exemplo de controle da UART e USART, onde os dados coletados pela USART são simplesmente passados para a UART:

Quadro 13. Algoritmo para controle da USART e UART.

```
#include <p18F4550.h>              //Definição do microcontrolador.
#include <usart.h>                 //Biblioteca da USART.
#include <sw_uart.h>               //Biblioteca da UART.

unsigned char usart_byte;                  //Dados recebidos pela USART.

void trata_usart(void);            //Função de tratamento da USART.
void DelayTXBitUART(void);                 //Função de baud rate da transmissão pela UART.
void DelayRXBitUART(void);                 //Função de baud rate da recepção pela UART.
void DelayRXHalfBitUART(void);             //Função de amostragem de bit da UART.

void DelayTXBitUART(void)                  //Delay de 249 ciclos.
{
Delay10TCYx(24);
Delay1TCYx(9);
}

void DelayRXHalfBitUART(void)              //Delay de 122 ciclos.
{
Delay10TCYx(12);
Nop();
Nop();
}

void DelayRXBitUART(void)                  //Delay de 247 ciclos.
{
Delay10TCYx(24);
Delay1TCYx(7);
}
```

```
void main()
{
      OpenUART();               //Função de criação do canal da UART.
      OpenUSART(USART_TX_INT_ON & USART_RX_INT_ON & USART_ASYNCH_MODE
            & USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH,129);
                        //Função de criação do canal da USART.

  while(1)
  {
      if(DataRdyUSART())          //Verifica se tem dado no buffer da USART.
      {
        usart_byte = ReadUSART();      //Caso POSITIVO, guarda o dado em "usart_byte".
        WriteUART(usart_byte);     //Transmite o dado através da UART.
      }
  }
}
```

Este exemplo foi utilizado nas primeiras etapas deste projeto com o intuito de verificar o funcionamento da USART e UART do PIC18F4550. A Fig. 41 mostra o diagrama em blocos referente a este teste:

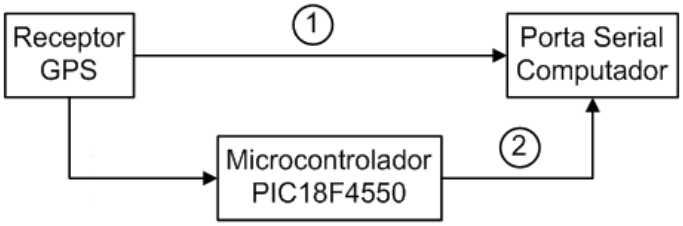


Fig. 41. Diagrama em blocos do teste da USART e UART.

Primeiramente, o receptor GPS foi conectado diretamente a porta serial do computador, e os dados recebidos foram salvos através do *software HyperTerminal* do *Windows* (*trajeto 1*). Posteriormente, o receptor GPS foi conectado ao PIC18F4550, que por sua vez foi conectado a porta serial do computador (*trajeto 2*). Os dados enviados pelo receptor GPS foram recebidos pela USART, e simplesmente encaminhados à porta serial do computador pela UART. Estes dados foram novamente salvos através do *software HyperTerminal* do *Windows*, e então comparados aos recebidos anteriormente, como mostra a Fig. 42 e a Fig. 43:

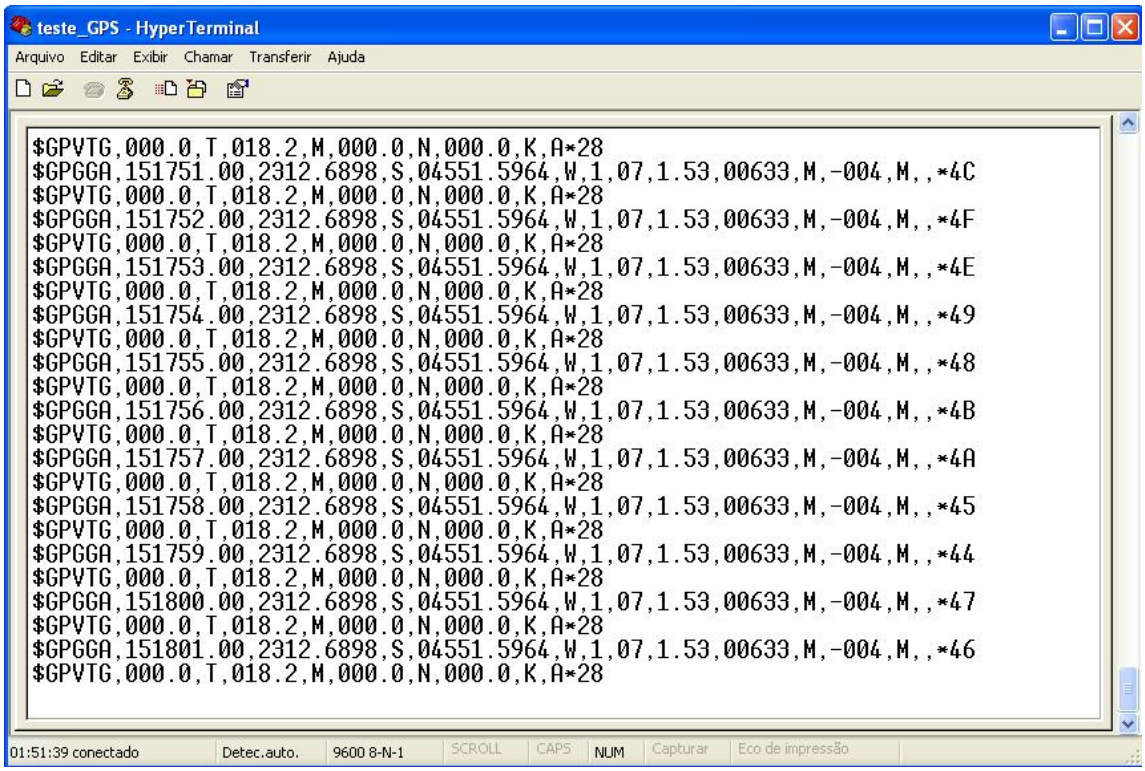


Fig. 42. Dados de posição geográfica enviados diretamente pelo receptor GPS.

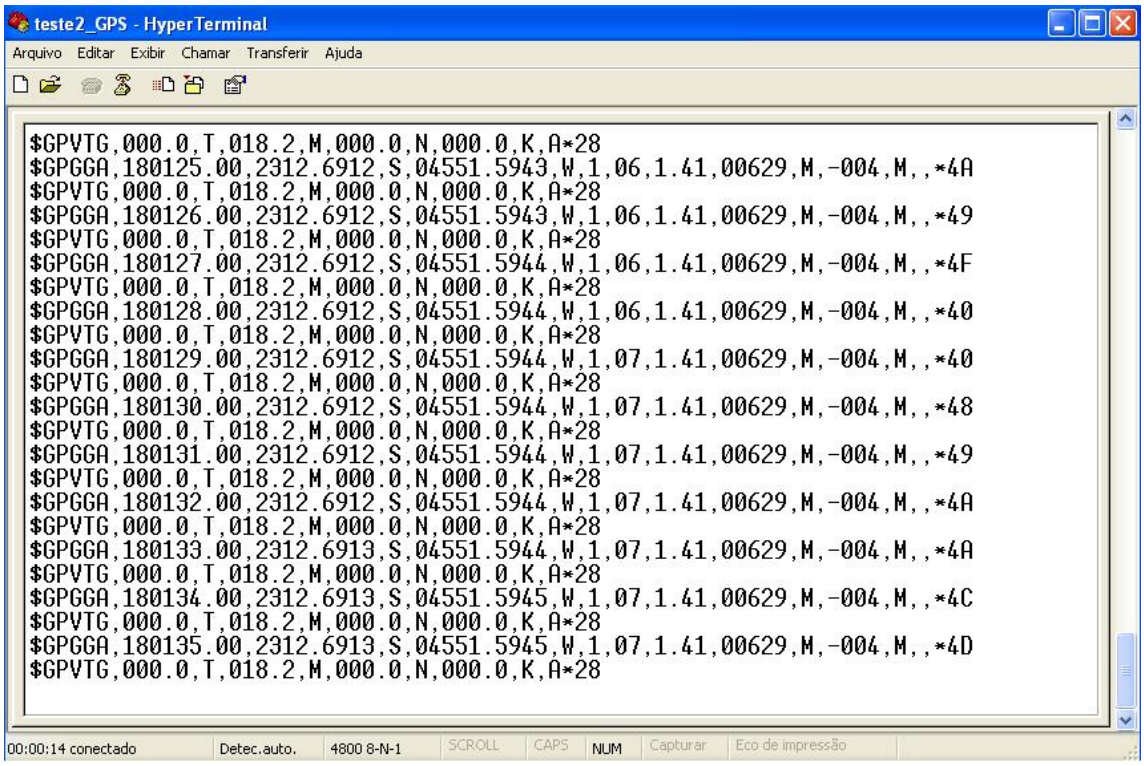


Fig. 43. Dados de posição geográfica processados pelo PIC18F4550.

Como se pode ver, o microcontrolador PIC18F4550 recebeu e enviou os dados corretamente, ainda sem nenhum processamento. Através deste primeiro teste, foi possível validar o funcionamento da USART e UART do microcontrolador.

B.4 Interrupção externa

Por definição, a interrupção é o ato que pára a execução de uma tarefa principal para executar outra secundária, e ao final desta retorna ao ponto de parada. Os microcontroladores da família PIC18F possuem dois vetores de interrupção, que são dois endereços da memória de programa. Estes vetores dividem as interrupções do PIC em níveis de prioridade não hierárquicos. O vetor de alta prioridade corresponde ao endereço 0x0080h (em hexadecimal) da memória de programa, enquanto o vetor de baixa prioridade corresponde ao endereço 0x0018h (em hexadecimal) da mesma. Ou seja, além de termos que habilitar as interrupções individualmente, também temos que definir sua prioridade. Para isso, o PIC18F4550 possui um conjunto de registradores dedicados (INTCON, INTCON2, INTCON3, PIR1, PIR2, PIE1, PIE2, IPR1, IPR2, etc.), cujo funcionamento detalhado pode ser encontrado nas referências bibliográficas utilizadas [19] e [33].

Para o caso especial deste projeto, foi utilizada uma interrupção externa de alta prioridade para recepção dos pulsos PPS (*Pulse per Second*) do receptor GPS.

O PPS é um terminal de saída do receptor GPS que envia um pulso positivo a cada segundo, com uma precisão de 50ns. Possui uma amplitude de 3,6V e uma corrente máxima de 5mA. Os pulsos de 4μs de largura geram uma mudança de estado no terminal RB0 do PIC18F4550, acionando a interrupção externa a cada borda de subida. Esta interrupção sendo gerada precisamente a cada segundo oferece ao microcontrolador um sincronismo externo

excelente para calcular os períodos de aquisição da posição geográfica. O Quadro 14 mostra um trecho de *software* onde é configurada a interrupção externa pelo terminal RB0 do PIC18F4550:

Quadro 14. Exemplo de configuração da interrupção externa para o terminal RB0.

```
#pragma code VETOR_HIGH_PRIORITY = 0x0008  //Indica a interrupção de alta prioridade.
void HIGH_int (void)
{
  _asm goto TRATA_PPS _endasm                    //Desvia o programa para o tratamento da
interrupção.
}
#pragma code

#pragma interrupt TRATA_PPS
void TRATA_PPS (void)                  //Rotina para tratamento da interrupção.
{
  CloseRB0INT();              //Reseta as configurações da interrupção externa do terminal RB0.
  INTCONbits.GIEH=0;                  //Desliga o bit de interrupcao global.

     if(INTCONbits.INT0IF==1)       //Testa o flag de interrupção externa do terminal RB0.
     {
          INTCONbits.INT0IF=0;          //Apaga o flag da interrupção externa do terminal RB0.
       PPS=PPS-1;                //Decrementa o valor da variável PPS.

          if(PPS==0)                  //Testa o valor da variável PPS.
          {
            desativa=0;           //Se igual a zero, limpa o flag "desativa".
          }
       OpenRB0INT(PORTB_CHANGE_INT_ON & RISING_EDGE_INT
           & PORTB_PULLUPS_OFF);     //Configura a interrupção externa no terminal RB0.
     }
}
```

Este trecho da programação mostra como é feito o controle do período de aquisição da posição geográfica. Neste caso, se for desejada uma aquisição a cada hora, o valor da variável PPS deve ser inicializado em 3600. Cada pulso gerado pelo receptor GPS ocasionará em uma interrupção externa, que ao ser tratada decrementará o valor da variável PPS. Logo, em exatamente uma hora será apagado o *flag desativa*, indicando uma nova aquisição da posição geográfica.

APÊNDICE C – O ICD2BR (*IN-CIRCUIT DEBUGGER*)

Desenvolvida pela LabTools e licenciada pela própria Microchip, esta ferramenta é capaz de realizar a gravação e depuração do *software* em todas as famílias PIC e dsPIC. Através da inserção de múltiplos pontos de parada (*breakpoints*) em determinados pontos do código-fonte, esta ferramenta permite uma simulação passo-a-passo direto na placa de aplicação, em tempo real. Com isso, a detecção e correção de erros podem ser feitas de maneira mais fácil e rápida. Porém, utiliza recursos do microcontrolador (memória e terminais) para desempenhar estas funções em tempo real, como mostra a Tabela 15:

Tabela 15. Pinagem do ICD2br [34].

Pino	Função
1	Não utilizado
2	GND
3	VCC
4	PGC (RB6)
5	PGD (RB7)
6	MCLR (RE3)

Como se pode ver, o ICD2br requer o controle exclusivo de três terminais do microcontrolador: RE3 (*Master Clear*), RB6 (PGC – *clock*) e RB7 (PGD – dados). A Fig. 44 mostra o ICD2br depurando o *software* na própria aplicação:

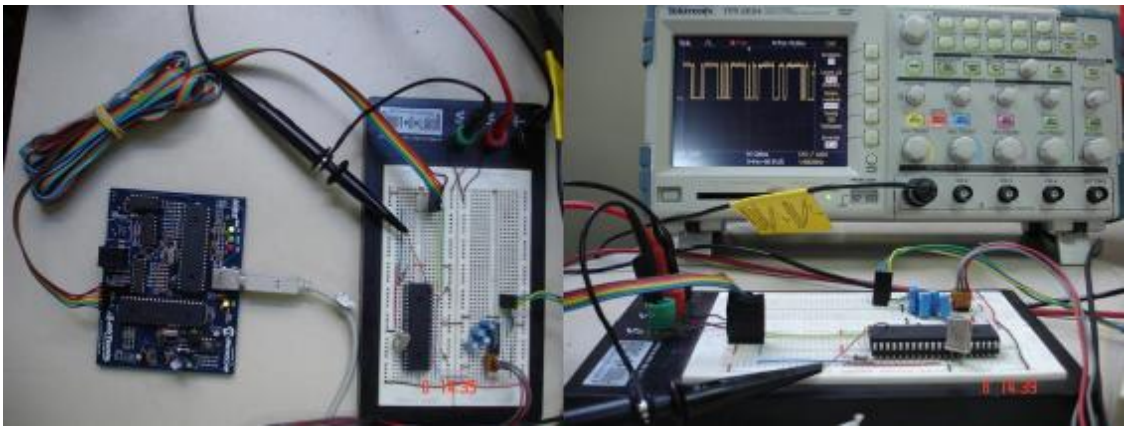


Fig. 44. ICD2br conectado à aplicação.

APÊNDICE D – TABELA DE CONVERSÃO ASCII – HEXADECIMAL

ASCII	Hex	Symbol
0	0	NUL
1	1	SOH
2	2	STX
3	3	ETX
4	4	EOT
5	5	ENQ
6	6	ACK
7	7	BEL
8	8	BS
9	9	TAB
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI

ASCII	Hex	Symbol
16	10	DLE
17	11	DC1
18	12	DC2
19	13	DC3
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US

ASCII	Hex	Symbol
32	20	(space)
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/

ASCII	Hex	Symbol
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

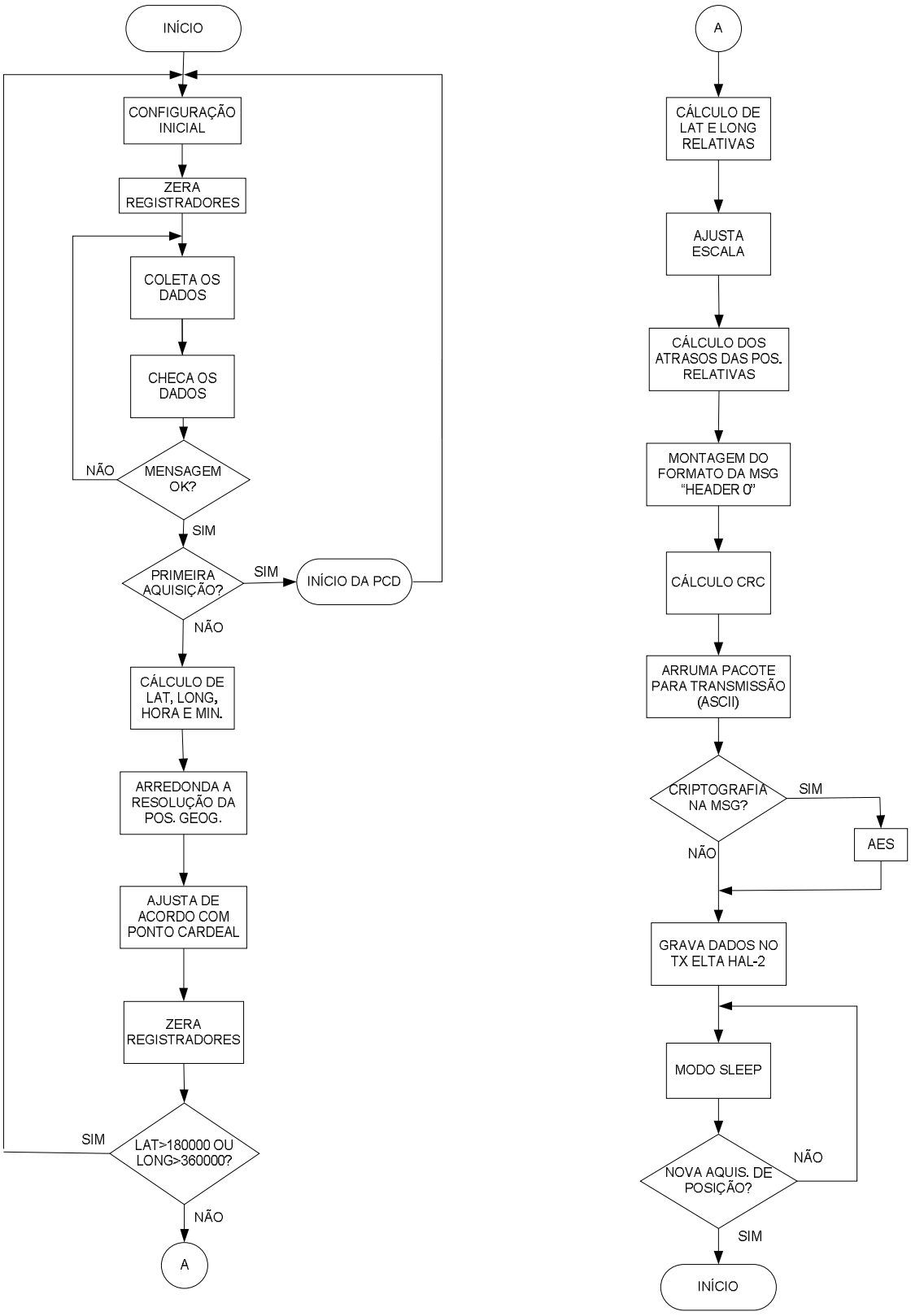
ASCII	Hex	Symbol
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O

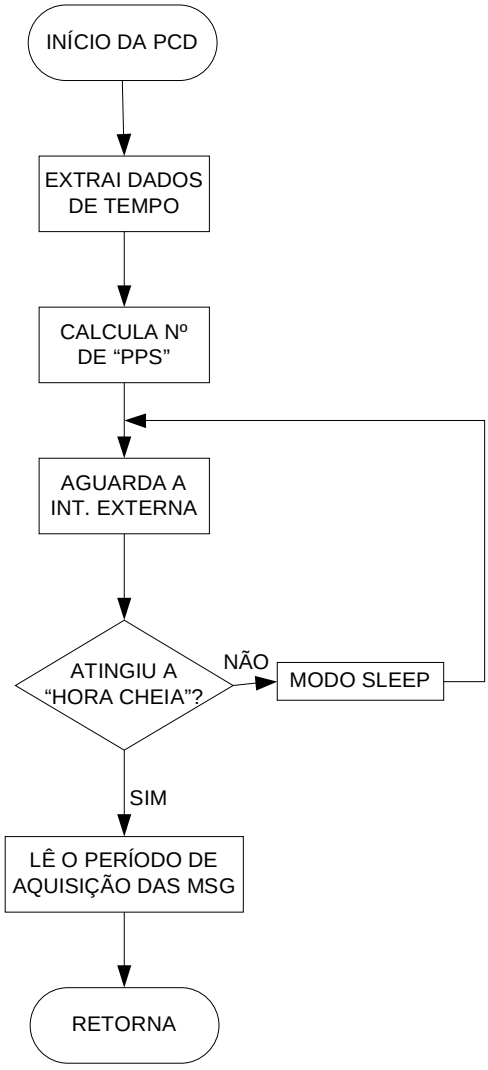
ASCII	Hex	Symbol
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

ASCII	Hex	Symbol
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o

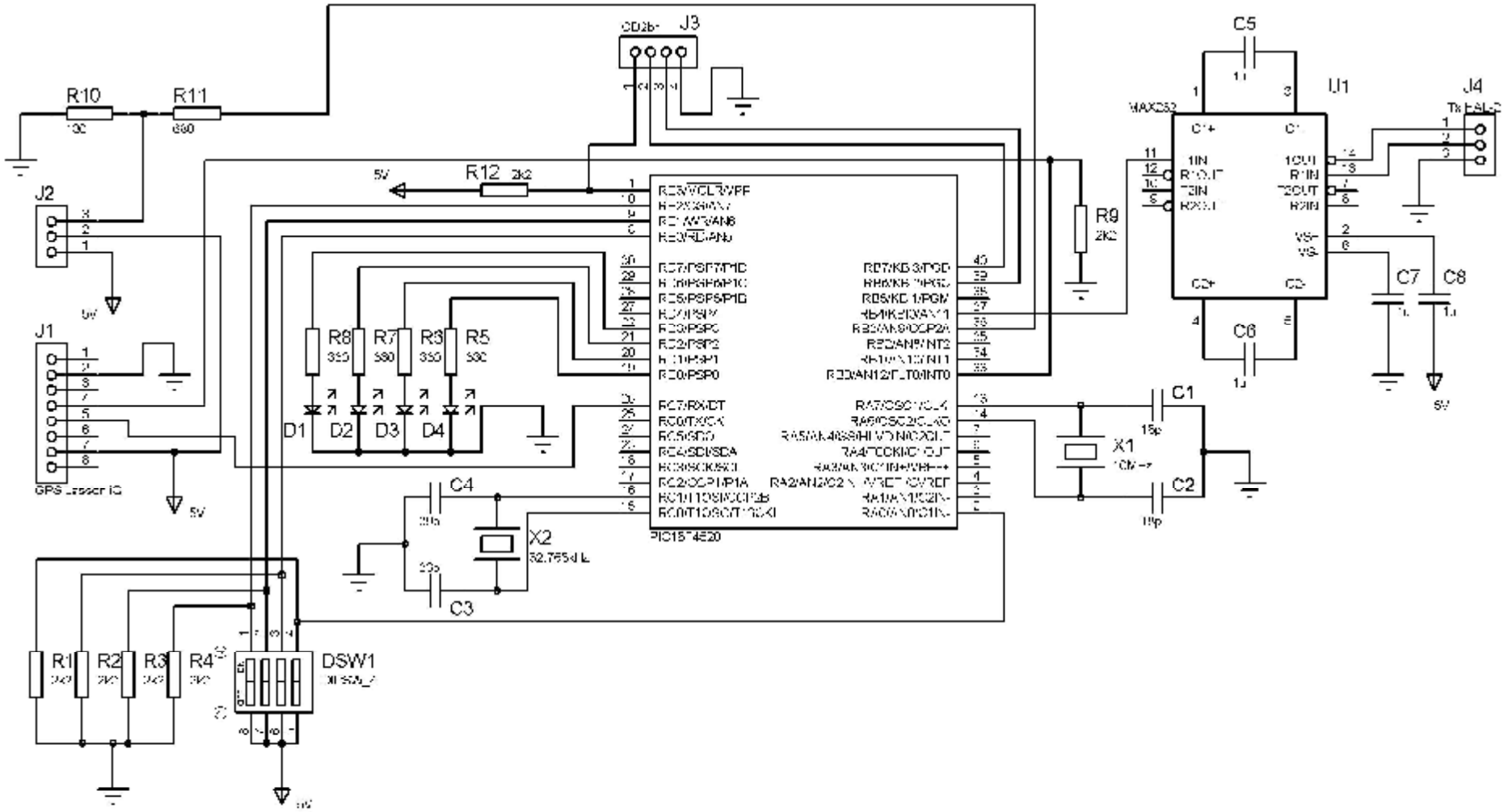
ASCII	Hex	Symbol
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	

APÊNDICE E – FLUXOGRAMA DO LOCALIZADOR GPS




INÍCIO DA PCD
EXTRAI DADOS DE TEMPO
CALCULA Nº DE “PPS”
AGUARDA A INT. EXTERNA
ATINGIU A “HORA CHEIA”?
NÃO
MODO SLEEP
SIM
LÊ O PERÍODO DE AQUISIÇÃO DAS MSG
RETORNA

APÊNDICE F – CIRCUITO ELÉTRICO DO LOCALIZADOR GPS



APÊNDICE G –PCI DO LOCALIZADOR GPS E *LAYOUT*