

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

IMPLEMENTAÇÃO DE UMA REDE DE
SENSORES DE PRESENÇA
UTILIZANDO O DISPOSITIVO TEXAS
EZ430-RF2500

Elaborado por:

Eurico Moura Bezerra Sobral

Recife, Fevereiro de 2010.

Livros Grátis

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

Milhares de livros grátis para download.

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**IMPLEMENTAÇÃO DE UMA REDE DE SENSORES
DE PRESENÇA UTILIZANDO O DISPOSITIVO
TEXAS EZ430-RF2500**

por

EURICO MOURA BEZERRA SOBRAL

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da
Universidade Federal de Pernambuco como parte dos requisitos para a obtenção do grau de
Mestre em Engenharia Elétrica.

ORIENTADOR: RAFAEL DUEIRE LINS, Ph.D.

Recife, Fevereiro de 2010.

© Eurico Moura Bezerra Sobral, 2010.

Resumo da Dissertação apresentada à UFPE como parte dos requisitos necessários
para a obtenção do grau de Mestre em Engenharia Elétrica.

IMPLEMENTAÇÃO DE UMA REDE DE SENSORES UTILIZANDO DISPOSITIVO TEXAS EZ430-RF2500

Eurico Moura Bezerra Sobral

Fevereiro / 2010

Orientador: Rafael Dueire Lins, Ph.D.

Área de Concentração: Comunicações.

Linha de Pesquisa: Rede de Sensores

Palavras-chave: Redes de Sensores Sem Fio (RSSF), Localização *Indoor*, Propagação de Sinais, eZ430-RF2500.

Número de Páginas: 108

RESUMO: O desenvolvimento tecnológico tem sido responsável pelo aparecimento de dispositivos cada vez menores com boa capacidade de processamento e que possuem baixo consumo de energia, além de permitirem a comunicação sem fio. É nesse cenário que está inserido o dispositivo da *Texas Instruments* eZ430-RF2500, uma plataforma para o desenvolvimento de sistemas microcontrolados com suporte à comunicação serial e rádio frequência (RF). As Redes de Sensores Sem Fio fazem uso extensivo desses dispositivos. O objetivo principal de uma RSSF é coletar dados e, se necessário, atuar conforme as informações coletadas, controlando um determinado evento. Aplicações dessas redes têm surgido nas mais diversas áreas, como por exemplo: monitoramento de ambiente, previsão de desastres, transporte, medicina, entretenimento, projetos militares, etc.

Esta dissertação descreve o uso de um dispositivo de baixo custo (eZ430-RF2500) para desenvolver uma rede de sensores de presença que juntos formam um sistema de localização. Para atingir esse objetivo, foi necessário o estudo das características e recursos disponíveis no dispositivo, bem como, efetuar sua configuração de maneira a atender a necessidade da proposta. Além disso, avaliou-se o desempenho do sistema a fim obter as suas condições de funcionamento.

Abstract of Dissertation presented to UFPE as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

IMPLEMENTING A NETWORK OF PRESENCE SENSORS USING THE TEXAS EZ430-RF2500

Eurico Moura Bezerra Sobral

February / 2009

Supervisor: Rafael Dueire Lins, Ph.D.

Area of Concentration: Communications.

Line of Research: Computer Networks

Keywords: Networks, Sensors, eZ430-RF2500

Number of Pages: 108

ABSTRACT: Technological development has been responsible for the appearance of smaller and smaller devices with reasonable processing power, low power consumption, and also allowing wireless communication. In this scenario we find the device EZ430-RF2500, from Texas Instruments, a platform for the development of microcontrolled systems, supporting serial communication and radio frequency (RF). These devices are extensively used in Wireless Sensors Networks. The main purpose of a WSN is to collect data and, if necessary, to act upon the information collected, controlling a particular event. Applications of these networks have emerged in several areas, such as environmental monitoring, disaster forecasting, transportation, medicine, entertainment, military projects, etc.

This thesis describes the use of a low-cost device (EZ430-RF2500) to develop a presence sensors network, acting as a location system. To achieve this goal it was necessary to study the characteristics and resources available on the device, as well as configuring the device. In addition, we evaluated the performance of the system to get their operating conditions.

Conteúdo

1.	Introdução.....	12
1.1.	Motivação	13
1.2.	Objetivo	13
1.3.	Organização da Dissertação	14
2.	O dispositivo Texas eZ430-RF2500	15
2.1.	eZ430-RF2500	15
2.2.	O Microcontrolador MSP430F2274	18
2.2.1.	A CPU do MSP430F2274	19
2.2.2.	Modos de Operação	24
2.2.3.	Modos de Comunicação Serial	25
2.2.4.	Conversor A/D de 10 Bits	27
2.3.	O Transceptor CC2500	30
2.3.1.	Interface de Controle	31
2.3.2.	Pilhas de Transmissão e Recepção	32
2.3.3.	Manipulador de Pacotes.....	32
2.3.4.	Formatos de Modulação	34
2.3.5.	RSSI e LQI	35
3.	Ensaio de propagação	38
3.1.	A propagação de Sinais.....	38
3.2.	Modelos para propagação de sinais em ambientes <i>outdoors</i>	39
3.2.1.	Modelo de Young	39
3.2.2.	Modelo do logaritmo da distância	40
3.2.3.	Modelo 2-raios.....	40
3.3.	Modelos para propagação de sinais em ambientes <i>indoors</i>	42
3.3.1.	Modelo ITU	42
3.3.2.	Logaritmo da distância (considerando obstáculos)	43
3.3.3.	Modelo de Souza e Lins	44

3.4.	Ensaios de Propagação (<i>Outdoor</i>)	44
3.4.1.	Metodologia do Ensaio	44
3.4.2.	Resultados das medições	45
3.4.3.	Análise dos resultados	47
3.5.	Ensaios de Propagação (<i>Indoor</i>)	50
3.5.1.	Metodologia do Ensaio	50
3.5.2.	Resultados e Análise.....	52
4.	Redes de Sensores Sem Fio.....	56
4.1.	As Redes de Sensores Sem Fio (RSSF).....	56
4.1.1.	Caracterização das RSSF.....	57
4.2.	Localização em RSSF.....	59
4.3.	Aplicações das RSSF	63
5.	Sensor de Presença e Sistema de Localização	65
5.1.	Sensor de Presença.....	65
5.1.1.	Arquitetura do Sensor de Presença.....	66
5.1.2.	Descrição do Protocolo de Comunicação	67
5.1.3.	Características do Sensor de Presença.....	71
5.2.	Sistema de Localização.....	71
5.2.1.	Arquitetura do Sistema de Localização	72
5.2.2.	<i>Softwares</i> de Gerenciamento da Rede	75
6.	Conclusões e Trabalhos Futuros	77
	Referências	80
	Apêndice A. Código Fonte dos Programas do dispositivo eZ430-RF2500	84
A.1 -	Software dos Elementos Móveis (EM's)	84
A.2 -	Software dos Pontos de Acesso (PA).....	88
A.3 -	Configuração dos Registradores do CC2500 para Modulação FSK 250kBaud (CC2500_FSK_250K.h)	94
A.4 -	Arquivo com Funções de Leitura e Escrita Para Interface SPI. (UART_SPI_FUNC.h)	96

Apêndice B. Técnicas de Modulação	99
B.1 - Modulação OOK.....	99
B.1.1. Modulação OOK	100
B.1.2. Demodulação OOK	101
B.2 - Modulação FSK.....	102
B.2.1. Modulação 2-FSK	103
B.2.2. Demodulação 2-FSK	104
B.3 - Modulação GFSK.....	105
B.4 - Modulação MSK	105
B.4.1. Modulação MSK	107
B.4.2. Demodulação MSK	107

Lista de Figuras

Figura 1.1 – Placa de Emulação eZ430-RF2500.....	13
Figura 2.1 – eZ430-RF2500.	16
Figura 2.2 – eZ430-RF2500 Placa Alvo.....	17
Figura 2.3 – Pinagem do MSP430x22x4.....	18
Figura 2.4 – Registrador de <i>Status</i>	21
Figura 2.6 – Divisão da memória <i>FLASH</i>	23
Figura 2.6 – Função de transferência do sensor de temperatura.	29
Figura 2.7 – Pinagem do CC2500.	30
Figura 2.8 – Diagrama de blocos simplificado – CC2500.	31
Figura 2.11 – Visão geral do processo de transmissão e recepção de pacotes	34
Figura 2.10 – Formato do pacote.....	34
Figura 2.11 – RSSI medido em função da Potência de entrada, para algumas taxas de transferência.....	37
Figura 3.1 – Modelo de Propagação de 2-Raios.....	41
Figura 3.2 – Janela do <i>software</i> de aquisição das potências.....	45
Figura 3.3 – Modelo de Young <i>versus</i> Valores Experimentais.....	48
Figura 3.4 – Modelo Logaritmo da Distância <i>versus</i> Valores Experimentais.....	48
Figura 3.5 – Modelo de 2-Raios <i>versus</i> Valores Experimentais.	49
Figura 3.6 – Planta Baixa do local de realização dos ensaios <i>indoor</i> (medidas em metros).	51
Figura 4.1 – Determinação da localização pelo método da <i>lateração</i>	61
Figura 4.2 – Determinação da localização pelo método da <i>angulação</i>	61
Figura 4.3 – Estruturas do algoritmo APIT.	63
Figura 5.1 – Placa Alvo com suporte a baterias.	66
Figura 5.2 – Arquitetura do Sensor de Presença.	67
Figura 5.3 – Confirmação de Presença de um EM.	68
Figura 5.4 – Diagrama temporal TX e RX: (a) Ponto de acesso, (b) Resposta do PA a uma confirmação, (c) Resposta do EM a um chamado do PA.....	68
Figura 5.5 – Diagrama de Fluxo Operacional Simplificado: (a) PA, (b) EM.	70
Figura 5.6 – Formato dos pacotes: (a) MSG, (b) ACK.	70

Figura 5.7 – Arquitetura do sistema de localização.....	73
Figura 5.8 – Campo de dados do pacote UDP	74
Figura 5.9 – Janela dos <i>softwares</i> : (a) PA HOST, (b) PA SERVIDOR	76
Figura B.1 – Modulação OOK (<i>On Off Keying</i>)	100
Figura B.2 – Modulador OOK.....	100
Figura B.3 – Demodulador não coerente OOK.	101
Figura B.4 – Detector de envelope para sinais AM.....	102
Figura B.5 – Sinal 2-FSK: (a) Sunde's FSK e (b) FSK com descontinuidade de fase	103
Figura B.6 – Modulador FSK Não Coerente	104
Figura B.7 – Demodulador 2-FSK	104
Figura B.8 – Modulação MSK	106
Figura B.9 – Modulador MSK.....	107
Figura B.10 – Demodulador MSK	108

Lista de Tabelas

Tabela 2.1 - Descrição dos pinos da placa alvo.....	17
Tabela 2.2 - Descrição dos pinos – Placa de Suporte a Bateria.....	17
Tabela 2.3 - Valores assumidos pelos registradores R2 e R3.....	21
Tabela 2.4 - Mapa de memória do MSP430F2274.....	23
Tabela 2.5 - Características dos modos LPMx.	24
Tabela 2.6 – Valores típicos para o $RSSI_{offset}$	36
Tabela 3.1 – Valor para o coeficiente de perda (N) para alguns ambientes.	43
Tabela 3.2 – Valores do Fator de Perda por Piso $L_f(m)$	43
Tabela 3.3 – Medições <i>outdoor</i> utilizando a modulação FSK e taxa de 250 kBaud.....	46
Tabela 3.4 – Parâmetros Utilizados no Modelo de 2-Raios.	49
Tabela 3.5 – Parametrização utilizada para Trajetória 1 do ensaio.	52
Tabela 3.6 – Parametrização utilizada para Trajetória 2 do ensaio.	52
Tabela 3.7 – RMSE em relação ao modelo ITU para as modulações utilizadas seguindo a Trajetória 1.	52
Tabela 3.8 - RMSE em relação ao modelo ITU para as modulações utilizadas seguindo a Trajetória 2.	53
Tabela 3.9 - RMSE em relação ao modelo Logaritmo da Distância com WAF para as modulações utilizadas seguindo a Trajetória 1.....	54
Tabela 3.10 - RMSE em relação ao modelo Logaritmo da Distância com WAF para as modulações utilizadas seguindo a Trajetória 2.....	54
Tabela 4.1 – Classificação das RSSF segundo a configuração.	58
Tabela 4.2 – Classificação das RSSF segundo o sensoramento.	58
Tabela 4.3 – Classificação das RSSF segundo a Comunicação.	59
Tabela 5.1 – Características do Sensor de Presença.	71

Lista de Acrônimos

CRC	<i>Cyclic Redundancy Check</i>
FIFO	<i>First In First Out</i>
FAF	<i>Floor Attenuation Factor</i>
FSK	<i>Frequency Shift Keying</i>
GFSK	<i>Gaussian shaped Frequency Shift Keying</i>
GPR	<i>General Purpose Register</i>
ISM	<i>Industrial, Scientific and Medical</i>
IrDA	<i>Infrared Data Association</i>
I ² C	<i>Inter-Integrated Circuit</i>
JTAG	<i>Joint Test Action Group</i>
LSB	<i>Least Significante Bit</i>
LQI	<i>Link Quality Indicator</i>
LAN	<i>Local Area Network</i>
LIN	<i>Local Interconnect Network</i>
LPM	<i>Low Power Mode</i>
MEMS	<i>Micro Eletro-Mechanical Systems</i>
MSK	<i>Minimum Shift Keying</i>
MSB	<i>Most Significante Bit</i>
OOK	<i>On Off Keying</i>
PQT	<i>Preable Quality Threshold</i>
PC	<i>Program Counter</i>
RSSI	<i>Received Signal Strength Indicator</i>
RSSF	<i>Rede de Sensores Sem Fio</i>
RISC	<i>Reduced Instruction Set Computer</i>
RGC	<i>Registradores Geradores de Constantes</i>
RMSE	<i>Root Mean Square Error</i>
SPI	<i>Serial Peripheral Interface</i>
SNR	<i>Signal Noise Ratio</i>
SIMO	<i>Slave In Master Out</i>
SOMI	<i>Slave Out Master In</i>
SPAF	<i>Soft Partition Attenuation Factor</i>
SP	<i>Stack Pointer</i>
SR	<i>Status Register</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
UDP	<i>User Datagram Protocol</i>
USB	<i>Universal Serial Bus</i>
USCI	<i>Universal Serial Communication Interface</i>
VoIP	<i>Voice over IP</i>
WAF	<i>Wall Attenuation Factor</i>

1. Introdução

O desenvolvimento de dispositivos e aplicações que utilizam a faixa de frequência não licenciada ISM (*Industrial, Scientific and Medical*) tem se tornado bastante comum nos últimos anos. A faixa ISM contempla às frequências de 2400 a 2483.5 MHz. Uma linha de aplicações nessa área são as redes de sensores sem fio (RSSF). O objetivo principal de uma RSSF é coletar dados e, se necessário, atuar conforme as informações coletadas, controlando um determinado evento. Os sensores são utilizados para coletar as informações que se deseja avaliar, por exemplo, temperatura, umidade, pressão, volume, etc. Existem vários contextos em que a aplicação desse tipo de rede se faz necessária, dentre eles o médico, ambiental, militar e de automação.

Ser capaz de se comunicar uns com os outros, trocando informações com boa velocidade de comunicação, baixo consumo de potência e baixo custo, são características dos dispositivos que compõe uma RSSF. É nesse cenário que está inserido o dispositivo da *Texas Instruments* eZ430-RF2500 [1], que pode ser visto também como uma plataforma para o desenvolvimento de sistemas microcontrolados com suporte à comunicação serial e rádio frequência (RF).

O dispositivo eZ430-RF2500 (Figura 1.1) é bastante compacto e com poucos centímetros de comprimento. Além disso, seu baixo consumo de corrente permite que ele possa ser alimentado por pequenas baterias concebendo assim portabilidade. Essa característica motivou a utilização desse dispositivo em um sistema que possui elementos móveis.

Neste trabalho é mostrado detalhes sobre a utilização do dispositivo eZ430-RF2500 como um sensor de presença, e também é abordada uma aplicação envolvendo esse sensor

como elemento de um sistema de localização que utiliza uma LAN para conectar as várias células que compõem o sistema.



Figura 1.1 – Placa de Emulação eZ430-RF2500.

1.1. Motivação

A criação por si só de uma rede de sensores móveis não define exatamente a aplicação. Várias aplicações que utilizam um sistema como esse podem ser sugeridas, entre elas a que é apresentada neste trabalho. A principal motivação deste trabalho é utilizar a informação proveniente de uma rede de sensores na aplicação que consiste em: realizar o roteamento de chamadas VoIP de um servidor executando o Asterisk [2] para um ponto próximo ao usuário. O Asterisk é um *software* gratuito que pode operar com diferentes tecnologias utilizadas em telefonia, sendo capaz de realizar basicamente as mesmas funções de um aparelho PBX (*Private Branch Exchange*).

A informação da localização de um indivíduo ou elemento dentro de uma área (casa, escritório, fábrica, etc...) também pode ser utilizada como meio para realização de um controle de acesso, que pode restringir tanto o acesso físico como o acesso a informações, garantindo mais segurança do ponto de vista pessoal, patrimonial e da propriedade intelectual.

1.2. Objetivo

O objetivo geral deste trabalho é utilizar um dispositivo de baixo custo (eZ430-RF2500) para desenvolver um sistema de localização, baseado em uma rede de sensores, que

forneça elementos necessário a um outro sistema responsável pelo roteamento de chamadas VoIP. Para atingir esse objetivo é necessário o conhecimento das características e recursos disponíveis no dispositivo, bem como, efetuar sua configuração de maneira a atender a necessidade da proposta. Além disso, deve-se avaliar o desempenho do sistema a fim de se obter as suas condições de funcionamento. Isso é feito com uma avaliação do dispositivo, do ponto de vista da propagação de sinais, obtendo informações sobre o alcance e seu comportamento em ambientes *indoor* e *outdoor*.

1.3. Organização da Dissertação

A organização da dissertação segue a ordem tomada durante o desenvolvimento do sistema apresentado neste trabalho. No segundo capítulo é apresentado em detalhes o dispositivo utilizado na elaboração do sistema, descrição dos principais elementos e recursos, processador e suas principais características, chip de rádio frequência, tipo de comunicação utilizada, etc. No terceiro capítulo é feita uma análise do comportamento do dispositivo, do ponto de vista da propagação de sinais, onde é avaliado o alcance e funcionalidade dos recursos de equalização e modulação disponíveis nesse dispositivo. Ainda no terceiro capítulo, é feita uma análise de alguns modelos de propagação tanto para ambientes internos como para ambientes externos, no sentido de testar a aderência desses modelos ao equipamento utilizado.

No quarto capítulo são apresentados alguns sistemas de redes de sensores sem fio (RSSF) e os mecanismos de localização de um elemento dentro desse tipo de rede.

O quinto capítulo descreve em detalhes tudo que foi desenvolvido com a finalidade de implementar o sistema de localização, ou seja, o protocolo de comunicação, as características dos sensores, a arquitetura da rede e os softwares para monitorar o sistema.

Por fim o sexto capítulo trata das conclusões obtidas no desenvolvimento do sistema, além de resumir os resultados obtidos e sugestão para trabalhos futuros. O Apêndice A contém o código fonte do programa desenvolvido para o dispositivo, escrito em linguagem C para o micro controlador MSP430F2274 da *Texas*. O Apêndice B descreve sucintamente as técnicas de modulação disponíveis no eZ430-RF2500.

2. O dispositivo Texas eZ430-RF2500

Neste capítulo são descritas as principais funcionalidades e recursos do dispositivo da *Texas Instruments* eZ430-RF2274 [1], uma plataforma de desenvolvimento para projetos de sistemas microcontrolados com suporte a comunicação serial e rádio frequência (RF). Para compreender melhor o funcionamento deste dispositivo também são abordados alguns aspectos dos demais *chips* que compõe essa placa. Por se tratar de um dispositivo bastante versátil, permitindo inúmeras aplicações, o foco principal deste capítulo serão os elementos que foram utilizados durante o desenvolvimento do projeto tratado nesta dissertação.

2.1. eZ430-RF2500

O dispositivo eZ430-RF2500 (Figura 2.1) é uma plataforma que pode ser utilizada para prototipação de sistemas microcontrolados com suporte a transmissão e recepção em rádio frequência.

Esta ferramenta faz parte de uma família de dispositivos criados com a mesma finalidade, e que nos últimos anos sofreram modificações, melhorando a compatibilidade com outros dispositivos e interfaces, além da redução na complexidade de uso. O dispositivo eZ430-RF2500 é composto por duas placas principais, uma que é responsável pela gravação e comunicação via USB, chamada de placa de emulação e outra placa, também chamada de placa alvo (*target board*), onde se localizam o microcontrolador MSP430F2274 [3][4] e o chip de RF CC2500 [5]. Estas placas se comunicam por meio de uma interface serial USCI (*Universal Serial Communication Interface*) trabalhando no modo UART (*Universal Asynchronous Receiver Transmitter*). Já a comunicação entre o

MSP430F2274 e o CC2500 é feita por meio da interface serial SPI (*Serial Peripheral Interface*).

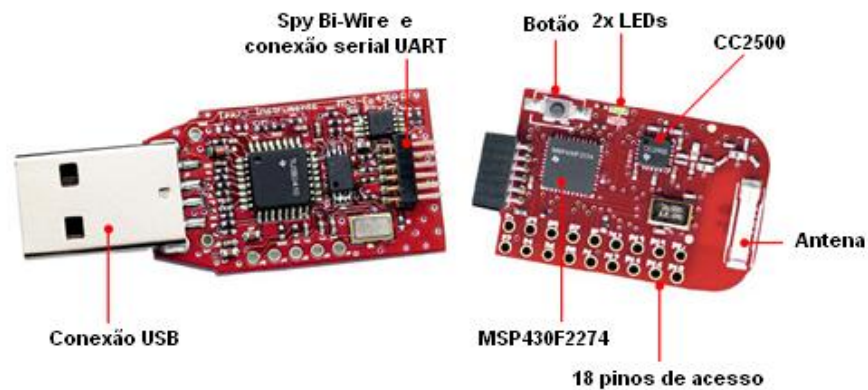


Figura 2.1 – eZ430-RF2500.

A interface USB (*Universal Serial Bus*) também proporciona aplicações nas quais, um PC recebe e envia informações ao dispositivo, sendo possível ter acesso aos periféricos do microcontrolador presente na placa alvo. Na verdade, a interface USB é emulada como uma porta serial comum (RS-232). Ou seja, ao serem instalados os *drivers* do dispositivo, a placa é reconhecida e conectada a uma porta serial virtual. Toda comunicação entre a placa eZ430-RF2500 e o PC é feita por meio desta interface.

Também é possível apagar, programar ou depurar programas, diretamente no dispositivo, pelos mesmos pinos utilizados na comunicação serial. Os pinos disponíveis para essa comunicação e suas funções podem ser vistos na Figura 2.2 e na Tabela 2.2. Este acesso à programação e depuração só é possível devido à interface de comunicação JTAG (*Joint Test Action Group*), presente em todos os dispositivos da família MSP430 [5][6]. A maioria dos dispositivos possui a interface JTAG a 4 fios. Embora outros sejam compatíveis com a nova geração, também chamada de *Spy Bi-Wire*, por utilizar 2 fios ao invés dos 4 fios da versão anterior.

Depois de programada, a placa alvo pode ser inserida em outros sistemas que possuam compatibilidades com interfaces seriais ou mesmo ser incorporada a outros circuitos onde as informações serão repassadas via rádio frequência para outro dispositivo compatível.

A placa alvo possui ainda vários pinos de entradas e saídas, bem como pinos de acesso aos periféricos do microcontrolador, como por exemplo: módulo PWM, capture/compare, conversor A/D, etc (ver Tabela 2.1). Também nota-se a presença de dois LEDs, um

vermelho e um verde, utilizados para sinalização, e um botão conectado a uma entrada de *pullup* (Figura 2.1). A utilização desses recursos dependerá da aplicação desenvolvida pelo usuário.

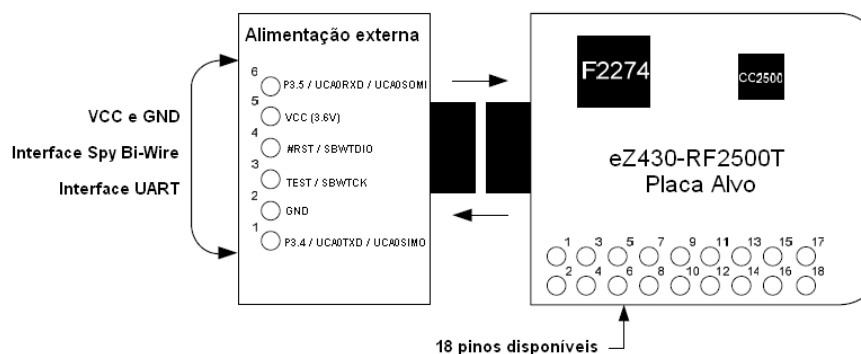


Figura 2.2 – eZ430-RF2500 Placa Alvo.

Tabela 2.1 - Descrição dos pinos da placa alvo.

Pino	Função	Descrição
1	GND	Referência de terra
2	VCC	Tensão de alimentação
3	P2.0/ACLK/A0	E/S de propósito geral/Saída de clock/ Entrada analógica A0
4	P2.1/SMCLK/A1	E/S de propósito geral/Saída de clock/ Entrada analógica A1
5	P2.2/TA0 /A2	E/S de propósito geral/Timer_A/ Entrada analógica A2
6	P2.3/TA1/A3	E/S de propósito geral/Timer_A/ Entrada analógica A3
7	P2.4/TA2/A4	E/S de propósito geral/Timer_A/ Entrada analógica A4
8	P4.3/TB0/A12	E/S de propósito geral/Timer_B/ Entrada analógica A12
9	P4.4/TB1/A13	E/S de propósito geral/Timer_B/ Entrada analógica A13
10	P4.5/TB2/A14	E/S de propósito geral/Timer_B/ Entrada analógica A14
11	P4.6/TBOUTH/A15	E/S de propósito geral/Saída Timer_B para alta impedância/ Entrada analógica A15
12	GND	Referência de terra
13	P2.6/XIN (GDO0)	E/S de propósito geral/Entrada para o cristal oscilador
14	P2.7/XOUT (GDO2)	E/S de propósito geral/Saída para o cristal oscilador
15	P3.2/UCB0SOMI	E/S de propósito geral/USCI_B0 <i>Slave Output Master Input</i>
16	P3.3/UCB0CLK	E/S de propósito geral/USCI_B0 clock
17	P3.0/UCB0STE	E/S de propósito geral/USCI_B0 <i>Slave Transmit Enable</i>
18	P3.1/UCB0SIMO	E/S de propósito geral/USCI_B0 <i>Slave Input MasterOutput</i>

* Os pinos 13 a 18 são utilizados para comunicação entre o MSP430F2274 e o CC2500

Tabela 2.2 - Descrição dos pinos – Placa de Suporte a Bateria.

Pino	Função	Descrição
1	P3.4/UCA0TXD/ UCA0SIMO	E/S geral /TX USCI/ <i>Slave Input MasterOutput</i>
2	GND	Referência de terra
3	#RST/SBWTIO	Reset/Entrada de teste Spy Bi-Wire
4	TEST/SBWTCK	Teste para modo JTAG/Spy Bi-Wire <i>clock test</i>
5	VCC (3.6V)	Tensão de alimentação
6	P3.5/UCA0RXD/UCA0SOMI	E/S geral /RX USCI/ <i>Slave Output Master Input</i>

2.2. O Microcontrolador MSP430F2274

O “cérebro” da placa alvo é o microcontrolador MSP430F2274, responsável por armazenar os programas do usuário, receber informações dos periféricos, além de realizar todo controle do chip de rádio frequência CC2500.

O MSP430F2274 é um microcontrolador que possui arquitetura RISC (*Reduced Instruction Set Computer*) de 16-Bit e trabalha com um relógio de 16MHz, possui também um baixo consumo de energia, podendo trabalhar com tensões de alimentação de 1,8V a 3,6V. Esta última característica torna possível aplicações onde a alimentação do circuito se dá por meio de baterias ou pilhas, o que permite sua portabilidade, permitindo embarcá-lo em sistemas móveis. A Figura 2.3 mostra a pinagem desse microcontrolador.

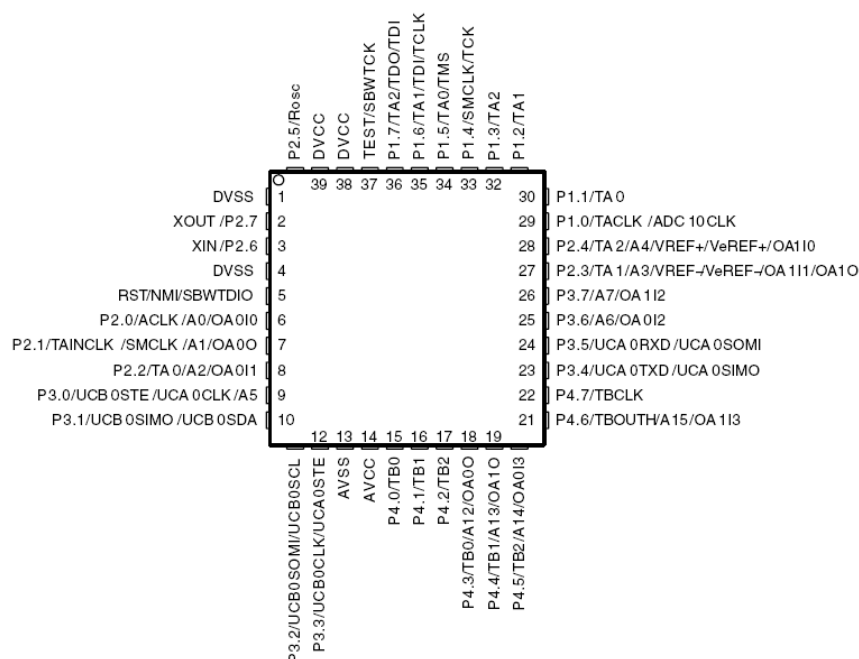


Figura 2.3 – Pinagem do MSP430x22x4.

Dentre as principais características deste microcontrolador pode-se citar:

- Baixo consumo de corrente;
 - Modo ativo 270μA;
 - Modo espera 0,7μA;
- Cinco modos de economia de energia;
- Saída rápida dos modos de economia de energia (~1μs);
- Conversor analógico digital integrado;

- Dois amplificadores operacionais configuráveis;
- 32 KB de memória flash;
- Timer de 16-Bits;
- Modos de comunicação serial
 - UART com detecção automática de velocidade;
 - SPI (*Serial Peripheral Interface*);
 - I²C (*Inter-Integrated Circuit*);
- Compatibilidade de programação utilizando padrão JTAG SPY Bi-Wire;
- Programação em assembly e C.

2.2.1. A CPU do MSP430F2274

O projeto deste microcontrolador foi voltado para proporcionar características desejadas por técnicas de programação, por exemplo, as linguagens de alto nível como o C.

Uma das características principais da arquitetura RISC é a de possuir instruções e tempos de execuções constantes, de fato isso não é completamente verificado nesta CPU, que possui algumas instruções de comprimento e tempos de execuções variáveis. Por isso, alguns especialistas não a consideram uma arquitetura RISC [6].

Este microcontrolador possui uma arquitetura de barramento clássica von Neumann, tendo a CPU um espaço único para endereçamento da memória. O barramento de endereços possui 16-bits permitindo o endereçamento de até 65.536 posições de memória [4]. Da mesma forma o barramento de dados também possui 16-bits, o que proporciona uma redução no número de buscas na memória, pois é possível processar mais informações de uma única vez.

Também é possível encontrar a presença de dezesseis registradores todos de 16-bits. Eles são nomeados de R0 a R15, sendo os registradores R0, R1, R2 e R3 com funções dedicadas e os registradores R4 a R15 para propósito geral (GPR – *General Purpose Register*). Dentre os registradores especiais têm-se: Contador de Programa (PC – *Program Counter*), Apontador da Pilha (SP – *Stack Pointer*), Registrador de Status (SR – *Status Register*) e os Registradores Geradores de Constantes.

2.2.1.1. Contador de Programa (R0)

Neste registrador encontra-se o endereço da posição de memória, que deverá ser lida e que contém a próxima instrução a ser executada pela CPU. As instruções possuem um número par de bytes (dois, quatro ou seis), tendo cada instrução um tamanho mínimo de 16-bits e por esta razão o bit 0 do contador de programa está sempre em nível lógico “0”.

Por se tratar de um registrador de 16-bits é possível endereçar até 64KB e após um *reset* no sistema, o contador de programa irá apontar para o endereço 0xFFFFE (em hexadecimal), que corresponde a posição na memória, da primeira instrução do programa do usuário.

2.2.1.2. Apontador de Pilha (R1)

A pilha é utilizada pela CPU para armazenar o endereço de retorno nas chamadas de sub-rotinas e de interrupções. O Apontador de Pilha guarda o endereço de memória onde se localiza o topo da pilha. Assim como no caso do contador de programa, o apontador de pilha também trabalha com um número par de bytes, tendo nas operações de empilhamento (armazenamento na pilha) e desempilhamento (esvaziamento de dados da pilha) um decremento e um incremento de duas posições respectivamente.

Em linguagens de alto nível como o C o preenchimento da pilha é feito de maneira automática, mas é possível também a utilização das instruções PUSH e POP, que permitem armazenar e ler um dado na pilha, possibilitando o salvamento de contexto em interrupções.

Um detalhe importante é que após o *reset* do sistema o conteúdo do apontador de pilha é indeterminado, sendo necessária sua inicialização.

2.2.1.3. Registrador de *Status* (R2)

Como o próprio nome sugere, este registrador armazena informações sobre o estado atual da CPU, ou seja, ele armazena as *flags* de controle da CPU, como por exemplo:

- V – *Flag* de estouro (*overflow*);
- GIE – Controle global das interrupções (habilita ou desabilita interrupções);
- N – *Flag* de estouro negativo (Indica uma operação com resultado negativo);
- Z – *Flag* de zero (Indica quando uma operação resulta num resultado nulo);

- C – *Flag* de transporte (Indica se houve um “vai um” na operação);
- SCG1 – Gerador de relógio 1;
- SCG0 – Gerador de relógio 0;
- OSCOFF – Oscilador desligado;
- CPUOFF – CPU desligada.

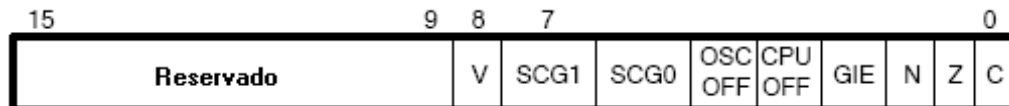


Figura 2.4 – Registrador de *Status*.

Este registrador também possui outra função, ele atua como gerador de constantes para instruções emuladas.

2.2.1.4. Gerador de Constantes (R2 e R3)

O MSP430F2274 possui um conjunto com 54 instruções sendo 27 instruções físicas e 24 instruções emuladas, estas instruções emuladas só são possíveis graças aos registradores geradores de constantes R2 e R3. Um conjunto de instruções reduzido significa que serão necessários menos bits para representar um *op-code*, reduzindo a quantidade de memória necessária para armazenar uma instrução. Para evitar um aumento no número de instruções físicas foram criados os RGC (Registradores Geradores de Constantes), que são utilizados como um dos operandos de uma instrução física para criar uma instrução emulada. As instruções emuladas não trazem ônus do ponto de vista do desempenho.

Os registradores geradores de constantes podem assumir os valores mostrados na tabela abaixo.

Tabela 2.3 - Valores assumidos pelos registradores R2 e R3.

Registrador	As	Constante	Significado
R2	00	Nenhuma	Acesso ao registrador R2
R2	01	(0)	Modo de endereçamento absoluto
R2	10	0x0004	Constante 4, útil para acessar o bit N no SR
R2	11	0x0008	Constante 8, útil para acessar o bit GIE no SR
R3	00	0x0000	Constante 0, útil para operações de apagamento
R3	01	0x0001	Constante 1, útil para acessar o bit C no SR
R3	10	0x0002	Constante 2, útil para acessar o bit Z no SR
R3	11	0xFFFF	Constante -1, útil para inversão de bits

O campo **As** especifica o modo de endereçamento da fonte da operação, encontrando-se presente nos bits das instruções [4].

2.2.1.5. Registrador de Propósitos Gerais (R4 a R15)

Os Registradores de Propósito Gerais (GPR) podem ser utilizados livremente, conforme a necessidade do usuário. No total existem doze registradores nomeados de R4 a R15, estes registradores podem ser utilizados tanto em operações de 8 *bits* como em operações com 16 *bits*. Geralmente são utilizados para armazenamento de variáveis e ponteiros.

2.2.1.6. Organização da Memória

A quantidade de memória disponível vai depender da versão do chip, para este modelo tem-se uma memória *FLASH* com 32kB de memória principal e 256 *bytes* de memória de informação, além de uma memória RAM com 1kB. A memória *FLASH* principal está dividida em 64 segmentos de 512 *bytes*, já a memória de informação está dividida em quatro segmentos de 64 *bytes*. A memória principal e a memória de informação diferem apenas no endereço físico e no tamanho dos segmentos [4]. A Figura 2.5 mostra essa divisão.

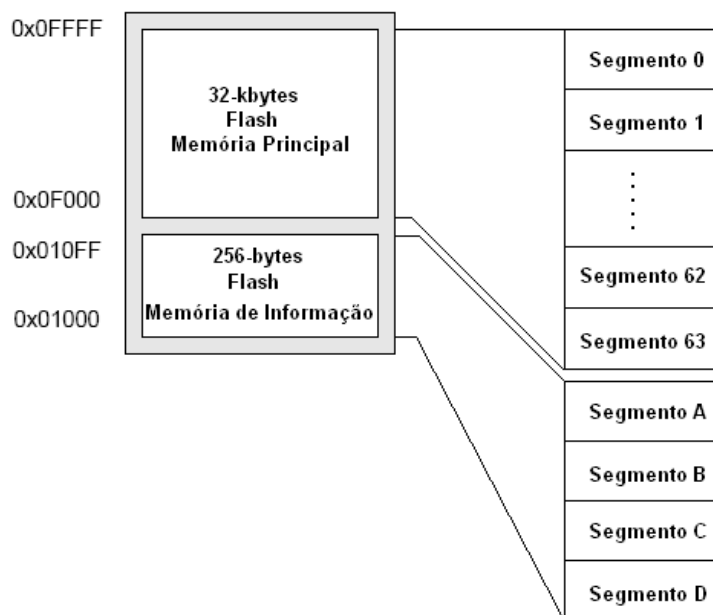


Figura 2.5 – Divisão da memória *FLASH*.

Embora seja possível a escrita no nível de um *bit*, um *byte* ou uma palavra, o apagamento é feito apenas por segmento, ou seja, a menor quantidade de informação que pode ser apagada é o que corresponde a um segmento. As posições de memórias são formadas por *bytes*. Na escrita ou leitura de uma palavra são utilizadas duas posições de memória, neste caso a posição par da memória recebe o *byte* menos significativo e a posição ímpar recebe o *byte* mais significativo.

A Tabela 2.4 mostra resumidamente o mapa da memória deste dispositivo:

Tabela 2.4 - Mapa de memória do MSP430F2274.


Endereço	Descrição
0x0000 a 0x000F	Registradores de Funções Especiais
0x0010 a 0x00FF	Registradores de controle de periféricos (acesso de 8 bits)
0x0100 a 0x01FF	Registradores de controle de periféricos (acesso de 16 bits)
0x0200 a 0x05FF	Memória RAM – 1KB
0x0600 a 0x0BFF	Área não implementada
0x0C00 a 0x0FFF	ROM de <i>BOOT</i>
0x1000 a 0x10FF	FLASH – Memória de Informação
0x8000 a 0xFFFF	FLASH – Memória Principal
0xFFC0 a 0xFFFF	Vetores de interrupção e <i>reset</i>

2.2.2. Modos de Operação

A família MSP430 possui vários modos de operação, esses modos de operação estão relacionados com o consumo de corrente do dispositivo. Ao todo existem seis modos, cada modo é definido de acordo com o estado dos *bits* do registrador de *status* (CPUOFF, OSCOFF, SCG0 e SCG1). A vantagem da seleção dos modos ser realizada por meio de mudanças nos *bits* do registrador de *status* é que na ocorrência de uma interrupção o registrador de *status* é salvo na pilha, sendo possível retornar ao modo de operação definido antes da interrupção. Também podemos manipular o conteúdo do registrador de *status*, salvo na pilha, e modificar o modo de operação durante uma rotina de tratamento de interrupção.

Além do modo de operação normal existem outros cinco modos de baixo consumo, chamados de *Low Power Modes* (LPM), que são numerados de zero a quatro (LPM0 a LPM4). Dentre os modos LPM, o de menor consumo é o LPM4 e o de maior consumo é o LPM0 conforme visto na Tabela 2.5.

Tabela 2.5 - Características dos modos LPMx.

Consumo	Modo	CPUOFF	OSCOFF	SCG0	SCG1	Descrição
	Normal	0	0	0	0	CPU e sinais de <i>clock</i> ativados
	LPM0	1	0	0	0	CPU e clock principal (MCLK) desativados. SMCLK e ACLK ativos
	LPM1	1	0	1	0	Idêntico ao LPM0, mas com o DCO desativado.
	LPM2	1	0	0	1	Idêntico ao LPM1, mas o SMCLK é desativado.
	LPM3	1	0	1	1	Idêntico ao LPM2 e o gerador DC do DCO é desativado.
	LPM4	1	1	1	1	CPU e sinais de clock desativados.

Ao entrar em um dos modos de baixo consumo, a CPU irá parar a execução do programa, mas o estado dos registradores e pinos de E/S, além do conteúdo da memória, são preservados.

Quando uma interrupção habilitada ocorre, o microcontrolador sai do modo de baixo consumo, salvando o conteúdo do contador de programa (PC) e do registrador de *status*

(SR). Em seguida, o registrador de *status* é modificado automaticamente para que a CPU entre no modo normal [6].

Os modos de baixo consumo foram criados para atender a três principais necessidades:

- Baixíssimo consumo de corrente;
- Melhorar a eficiência para uma aplicação específica;
- Redução no consumo individual por periféricos.

2.2.3. Modos de Comunicação Serial

A comunicação serial desenvolve um papel importante na conectividade entre dispositivos e periféricos, muitas vezes é necessário transferir mensagens com comprimento de alguns *bytes* entre dispositivos e não seria prático o envio simultâneo dessas informações. Por tanto, a comunicação serial atua, do lado do transmissor, fragmentando a mensagem e a enviando *bit a bit* sequencialmente. Do lado do receptor, esta mensagem que chegou fragmentada é reorganizada, retornando ao seu estado original.

O microcontrolador MSP430F2274 possui dois módulos de comunicação serial universal (USCI), chamados de USCI_A0 e USCI_B0. Cada módulo de comunicação suporta alguns modos de operação.

O módulo USCI_A0 suporta os seguintes modos:

- Modo serial assíncrono (UART);
- Modelagem de pulso para IrDA (*Infrared Data Association*);
- Detecção automática de velocidade para comunicações LIN (*Local Interconnect Network*);
- SPI.

O módulo USCI_B0 suporta os seguintes modos:

- I²C;
- SPI.

O modo de comunicação UART dos microcontroladores da família MSP430 suportam tamanhos de caracteres de 7 e 8 bits com paridade par, ímpar ou nenhuma, além de possuir registradores e buffers separados para transmissor e receptor. Também é possível escolher se o bit mais significativo (MSB – *Most Significant Bit*) ou o bit menos significativo (LSB – *Least Significant Bit*) é transmitido primeiro. Várias opções de velocidades, circuitos

para detecção de erros e suporte a endereçamento, vetores de interrupção para transmissão e recepção também estão presentes neste módulo.

Quando os modos de comunicação serial estão devidamente configurados, a transmissão pela UART é um processo bastante simples, sendo executada automaticamente pelo módulo serial. Para iniciar a transmissão basta escrever o *byte* que se deseja transmitir no registrador UCAxTXBUF e monitorar a *flag* de interrupção UCAxTXIFG do registrador IFIG2. Uma interrupção é sinalizada cada vez que o *buffer* (UCAxTXBUF) está pronto para receber um novo dado. O *flag* UCAxTXIFG é limpo a cada novo byte escrito no *buffer* [4]. Se a interrupção estiver habilitada, quando um byte for transmitido, a CPU irá parar e entrar na rotina de tratamento desta interrupção. Isto é útil quando se deseja enviar vários bytes, ou seja, durante a rotina de tratamento da interrupção o programa do usuário poderá carregar no *buffer* o próximo byte que será transmitido.

Outro importante modo de comunicação serial suportado pelos dispositivos da família MSP430 é o SPI. A comunicação neste modo é síncrona e operar com três ou quatro pinos, outras características desse modo são:

- Comprimento dos dados de 7 ou 8 *bits*;
- Opção de LSB ou MSB, como primeiro bit a ser enviado;
- Modos mestre ou escravo;
- *Buffer* do transmissor e do receptor separados, além dos registradores de deslocamento serem independentes.

Este tipo de interface é bastante utilizada para conectar dispositivos como, por exemplo, memórias e cartões de memórias FLASH, potenciômetros digitais, conversores A/D e D/A, o MSP430F2274 e o transceptor de RF CC2500, etc.

Uma das características citadas anteriormente é que este protocolo é do tipo mestre-escravo, ou seja, a comunicação é controlada e iniciada pelo mestre e o dispositivo escravo apenas responde as solicitações do mestre. Tanto na comunicação a 3-fios como na comunicação a 4-fios, existem dois pinos destinados a transmissão e recepção de informações, são eles: SIMO (*Slave In Master Out*) e SOMI (*Slave Out Master In*). O SIMO é utilizado para envio de informações do dispositivo mestre para o escravo, já o SOMI é utilizado para envio de informações do dispositivo escravo para o mestre. O

terceiro pino desta configuração é o pino de *clock*, que é provido pelo mestre. O quarto pino é opcional (comunicação a 4-fios) e serve para habilitar o dispositivo escravo, sendo importante em sistemas com mais de um escravo.

A transmissão de informações pelo mestre é feita por meio da escrita de um dado no registrador UCB0TXBUF. A cada transmissão um *byte* é recebido simultaneamente, enviado pelo escravo, e armazenado no registrador UCB0RXBUF. A leitura é feita da mesma forma que a escrita, sendo o conteúdo do registrador UCB0RXBUF o dado enviado pelo dispositivo escravo.

No dispositivo escravo o processo de transmissão e recepção é idêntico ao do dispositivo mestre, a única diferença é que a informação só é enviada quando o mestre executa uma transmissão com o escravo habilitado (quarto pino em nível lógico “0”). As *flags* UCB0TXIFG e UCB0RXIFG são utilizadas para indicar o estado da transmissão e recepção, respectivamente.

2.2.4. Conversor A/D de 10 Bits

Em várias situações necessitamos de converter um sinal analógico em um sinal digital. Medições realizadas por diversos tipos de transdutores resultam em sinais analógicos, o armazenamento e processamento desses sinais são em sua grande maioria realizados por circuitos digitais. O MSP430F2274 possui dois conversores A/D, um de 10 bits e outro de 12 bits, neste item é mostrado apenas o de 10 bits.

As principais características deste conversor são:

- Taxas acima de 200 ksps (*Kilo Samples per Second*);
- Configuração de amostragem periódica;
- Seleção da tensão de referência no próprio chip (1,5V ou 2,5V);
- Seleção de referência interna ou externa via *software*;
- Sensor de temperatura conectado ao conversor A/D via *software*;
- DTC (*Data Transfer Controller*) para armazenamento automático dos resultados convertidos.

Neste circuito a entrada analógica selecionada é conectada ao circuito de amostragem e retenção (*sample and hold*), que é comandado pelo temporizador de amostragem, este

circuito inicia e temporiza a fase de amostragem do sinal analógico. Dependendo da duração do ciclo de amostragem os bits ADC10SHTx do registrador ADC10CTL0 são configurados, podendo assumir 4, 8, 16 ou 64 ciclos de *clock*. A duração do ciclo de amostragem pode ser estimada considerando que a entrada analógica pode ser modelada por um filtro passa-baixa. Este cálculo também leva em consideração a resistência da fonte, além da resistência e capacitância interna do conversor [4][6].

A conversão do sinal analógico em digital tem duração de 13 ciclos de *clock*, cuja fonte pode ser escolhida pela configuração dos bits ADC10SSELx do registrador ADC10CTL1. Após a configuração de todos os registradores, o início da conversão pode ser feita pela fonte indicada nos bits SHSx do registrador ADC10CTL1 ou via *software* por meio do bit ADC10SC no registrador ADC10CTL0. Neste último caso, o bit ADC10SC é automaticamente apagado após o início da amostragem/conversão.

O resultado da conversão é armazenado no registrador de 16 bits ADC10MEM e pode estar representado em binário ou na notação complemento de dois, dependendo da escolha feita no bit ADC10DF do registrador ADC10CTL1. No caso do resultado estar em binário os dez bits considerados são os bits 9-0 e os demais bits (15-10) são mantidos em zero, já no caso da notação complemento de dois os bits considerados devem ser os bits 15-6 e os demais são mantidos em zero.

Existem ainda quatro modos de operação com relação aos canais disponíveis, estes modos dizem respeito ao modo como a leitura de cada canal ou de uma seqüência de canais é feita, são eles: um canal sem repetição, seqüência de canais sem repetição, um canal repetidamente e seqüência de canais repetidamente [4].

Os dispositivos da família MSP430 que possuem um conversor A/D de 10 bits são dotados de um sensor de temperatura integrado ao chip, este sensor de temperatura é conectado ao conversor A/D via *software* pela configuração dos bits INCHx no registrador ADC10CTL1. As demais configurações devem ser feitas igualmente como no caso de uma entrada externa. O valor de uma conversão depende da escolha dos valores das tensões de referências, e é dada por:

$$V_{ADC} = 1023 \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}. \quad (2.1)$$

O limite superior de uma conversão é definido pelo valor da tensão de referência V_{R+} , o valor mínimo é definido pela tensão V_{R-} e V_{in} representa tensão de entrada no conversor. No caso do sensor de temperatura $V_{R-} = 0$ V e $V_{R+} = 1,5$ V ou $V_{R+} = 2,5$ V dependendo do bit REF2_5V do registrador ADC10CTL0. Substituindo V_{R-} e isolando V_{in} em (2.1), tem-se:

$$V_{in} = \frac{V_{ADC} \cdot V_{R+}}{1023}. \quad (2.2)$$

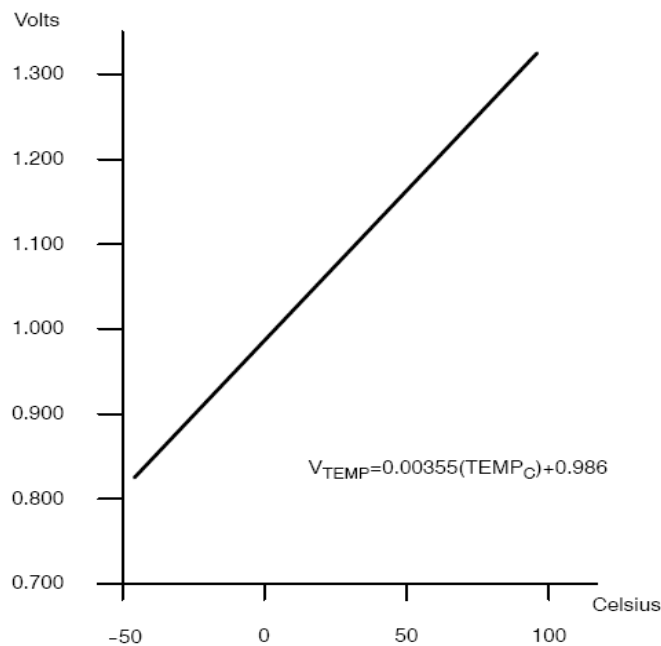


Figura 2.6 – Função de transferência do sensor de temperatura.

De posse da curva de transferência do sensor de temperatura, mostrada na Figura 2.6 e substituindo V_{in} por V_{TEMP} na equação da curva, é possível encontrar o valor da temperatura em graus Celsius por meio da seguinte equação:

$$TEMP_C = \left(\frac{V_{ADC} \cdot V_{R+}}{1023} - 986 \right) \cdot \frac{1}{3,55}. \quad (2.3)$$

Para garantir a correta conversão dos valores medidos pelo conversor é necessário que o tempo de amostragem seja superior a 30 μ s. O valor do *offset* (0,986 V) é um valor típico definindo pelo fabricante, sendo assim pode ser necessária a calibração do sensor.

2.3. O Transceptor CC2500

O circuito integrado CC2500 [5] é um transceptor de rádio frequência que opera na faixa não licenciada ISM (*Industrial, Scientific and Medical*) que corresponde às frequências de 2400 a 2483,5 MHz. Este dispositivo é bastante utilizado em aplicações de áudio sem fio, teclado e mouse sem fio, joysticks *wireless*, rede de sensores, transferência de arquivos, etc.

O chip CC2500 é responsável por toda comunicação *wireless* realizada pelo dispositivo eZ430-RF2500, mas para completo funcionamento deste chip é necessário um elemento controlador. O microcontrolador MSP430F2274 é o elemento controlador do CC2500, ele deverá armazenar os programas do usuário contendo as rotinas necessárias para o controle do chip de rádio frequência. A comunicação entre os microcontrolador e o CC2500 é feita por meio de uma interface serial operando no modo SPI.

Por meio da interface SPI tem-se acesso aos registradores de configuração e registradores de *status* do chip CC2500. Também é por meio desta interface que são enviados do microcontrolador para o transceptor os dados que deverão ser transmitidos via rádio frequência. A Figura 2.7 e Figura 2.8 mostram o diagrama de pinos do CI e seu diagrama de blocos simplificado, respectivamente.

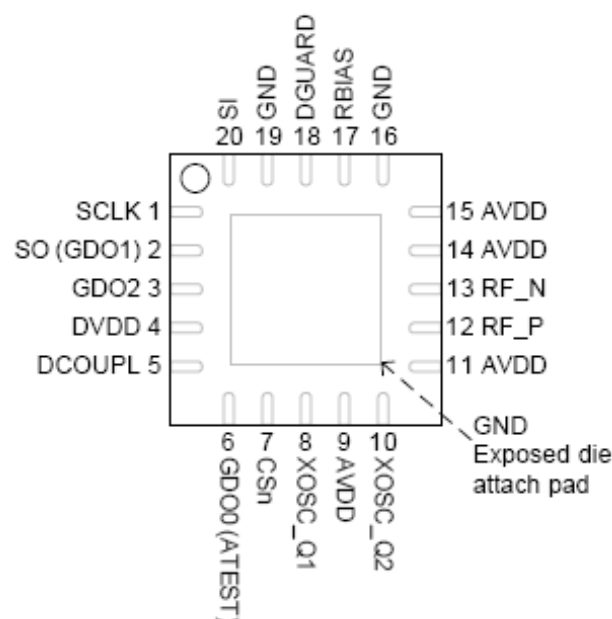


Figura 2.7 – Pinagem do CC2500.

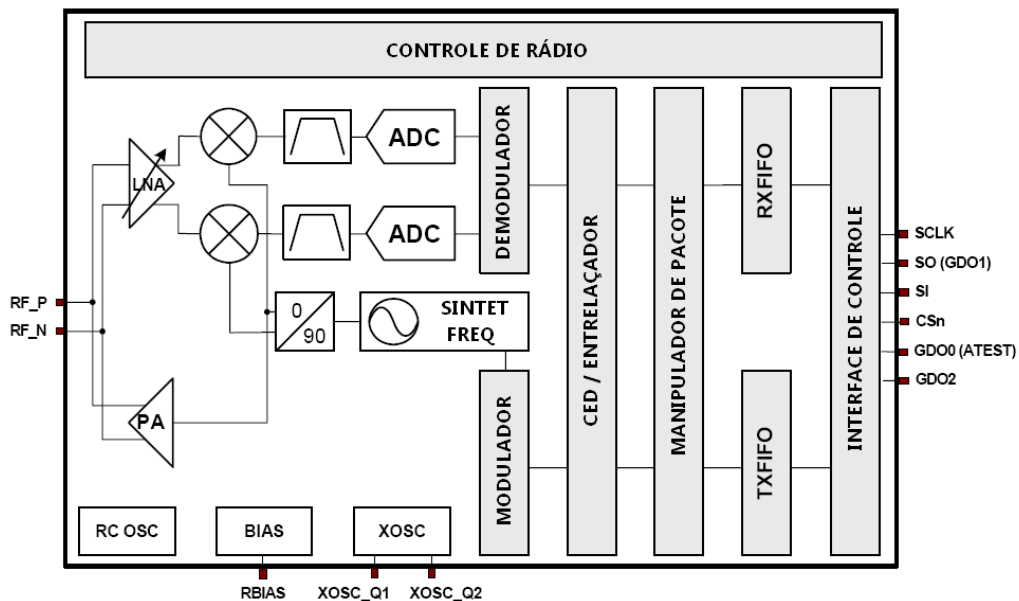


Figura 2.8 – Diagrama de blocos simplificado – CC2500.

2.3.1. Interface de Controle

Como dito anteriormente todo controle do CC2500 é feito através da interface SPI a 4-fios. Na Figura 2.8 observam-se no bloco da Interface de Controle seis pinos utilizados para controle, dos quais quatro pinos são reservados para comunicação serial SPI, são eles: SO (*slave output*), SI (*slave input*), SCLK (fonte de *clock*), CSn (*chip select*). Os demais pinos (GDO0 e GDO2) são utilizados pelo CC2500, dependendo de uma configuração prévia, para monitoramento do chip de RF podendo ser utilizados no microcontrolador como entrada de eventos que geram interrupções, ou seja, mudanças de estado nesses pinos poderão gerar interrupções no microcontrolador.

Toda troca de dados realizada por esta interface SPI é feita transmitindo-se o bit mais significativo primeiro. As transações sempre começam com o envio de um cabeçalho de um byte, o bit mais significativo deste byte indica se a operação é uma leitura ou escrita, o segundo bit mais significativo indica se a transferência será uma sequência de bytes ou se será apenas um byte, modos *burst* e *single byte*, respectivamente. Assim, pode-se concluir que a capacidade de endereçamento para os registradores é de apenas seis bits, ou seja, de 0x00 a 0x3F [4]. Durante a transferência de dados o pino CSn deve ser mantido em nível baixo (o chip é ativado em nível lógico “0”) até o final da transmissão.

Sempre que um byte é enviado pela interface SPI com destino ao chip de RF, um byte de *status* é enviado do CC2500 ao dispositivo simultaneamente. Esse byte de *status* contém informações sobre a máquina de estados do transceptor de RF.

2.3.2. Pilhas de Transmissão e Recepção

Neste CI existem duas pilhas, uma destinada à transmissão (TXFIFO) e outra a recepção (RXFIFO). As duas pilhas são do tipo FIFO (*First In First Out*), ou seja, o primeiro byte armazenado será o primeiro *byte* a ser lido. Cada uma delas possui 64 *bytes* de capacidade, portanto caso seja necessária a transmissão ou recepção de mensagens maiores que este valor é exigido um controle mais elaborado por parte do programa contido no microcontrolador.

O byte de *status*, enviado quando um comando é transmitido pela interface SPI, contém informações sobre o estado dessas pilhas, sendo possível saber se ocorreu um *overflow* na pilha de recepção ou um *underflow* na pilha de transmissão, ou ainda, uma estimativa de quantos *bytes* estão armazenados em cada uma delas. Para se ter uma estimativa da quantidade de *bytes* armazenados em cada pilha, basta modificar o *bit* de leitura/escrita do byte de comando enviado ao CC2500. Se a operação estiver selecionada como leitura, os três bits menos significativos do byte de *status* indicarão quantos *bytes* estão armazenados na RXFIFO. Por outro lado, se a operação for uma escrita, estes *bits* indicarão a quantidade de *bytes* disponível na TXFIFO.

Como a capacidade da TXFIFO e da RXFIFO é de 64 *bytes*, é impossível, com os três bits recebidos do byte de *status*, saber a ocupação exata das pilhas. Portanto, só é possível saber se há 15 bytes ou mais na RXFIFO ou 49 *bytes* ou menos na TXFIFO. Contudo, existem dois registradores de *status* (RXBYTES e TXBYTES) que indicam a quantidade exata de bytes disponíveis em cada pilha.

2.3.3. Manipulador de Pacotes

O Manipulador de Pacotes é o bloco responsável pelo tratamento dos pacotes que chegam e que saem do CC2500. Este manipulador insere ou extrai elementos no pacote que são destinados à maioria dos protocolos de rádio. Dependendo da configuração realizada pelo usuário, é possível inserir um determinado número de bytes de preâmbulo, uma palavra de

sincronismo com dois ou quatro *bytes*, além da aplicação do CRC (*Cyclic Redundancy Check*) [8] sobre o campo de dados do pacote.

Do lado do transmissor opcionalmente pode-se aplicar uma sequência pseudo-aleatória aos dados transmitidos, esse recurso é chamado de *data whitening* [5]. Esta sequência é aplicada por meio de uma operação OU-EXCLUSIVO entre os bits da sequência transmitida e os bits da sequência pseudo-aleatória. Os bytes de preâmbulo e da palavra de sincronismo ficam livres da aplicação da sequência pseudo-aleatória. Uma das vantagens da aplicação desta sequência aos dados transmitidos deve-se ao fato que, em transmissões *wireless*, quanto mais aleatória for a sequência de bits transmitidos e quanto mais livre de um nível DC o sinal estiver, mais suave será a distribuição de potência sobre a largura de banda ocupada, o que aumenta o desempenho do receptor [5]. Ainda do lado do transmissor, está disponível a aplicação de códigos corretores de erros, permitindo trabalhar sob condições onde a relação sinal ruído (SNR – *Signal Noise Ratio*) é baixa. Juntamente com a aplicação do código corretor de erros é aplicado um entrelaçador, cuja função é evitar que erros em surto afetem ainda mais os dados transmitidos. A Figura 2.9 mostra as etapas do processo de envio e recepção do pacote, quando as opções de correção de erro e entrelaçamento estão ativadas.

Do lado do receptor é feita a detecção dos *bytes* de preâmbulo, da palavra de sincronismo e é calculado sobre os dados recebidos o CRC, que será comparado com o enviado pelo transmissor. Opcionalmente pode ser feita a verificação de endereço e do comprimento do pacote, permitindo que os pacotes sejam descartados caso a verificação de endereço falhe ou se o tamanho do pacote for diferente do esperado. Além disso, se foi aplicada aos dados a sequência pseudo-aleatória no transmissor, a mesma sequência será aplicada aos dados no receptor, recuperando assim o estado original do pacote. O mesmo acontece com o entrelaçador e o código corretor de erros, ou seja, nos dados recebidos é aplicado o entrelaçador e é feita a decodificação da informação corrigindo os possíveis erros (Figura 2.9).

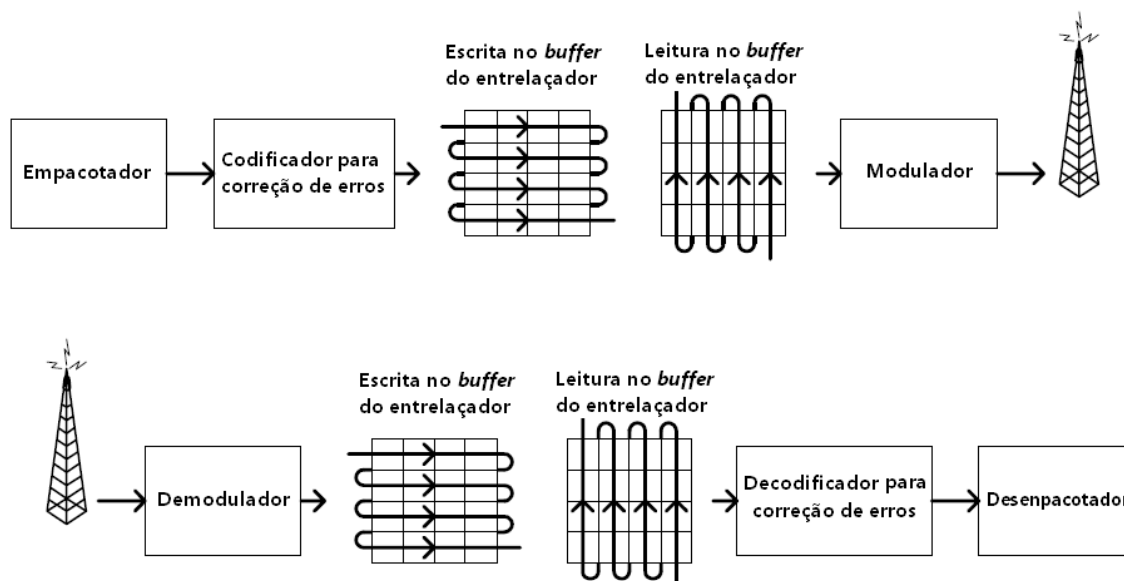


Figura 2.9 – Visão geral do processo de transmissão e recepção de pacotes

A Figura 2.10 ilustra o formato do pacote, bem como os bytes inseridos de acordo com a configuração desejada pelo usuário.

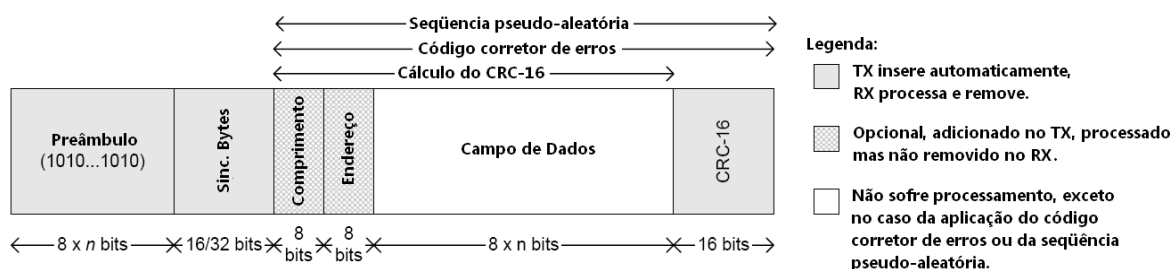


Figura 2.10 – Formato do pacote.

2.3.4. Formatos de Modulação

O CC2500 pode trabalhar em quatro modulações diferentes, FSK (*Frequency Shift Keying*), GFSK (*Gaussian shaped Frequency Shift Keying*), MSK (*Minimum Shift Keying*) e OOK (*On Off Keying*), além disso, pode atingir taxas de transmissão de até 500 kBaud. Opcionalmente, pode-se utilizar a codificação Manchester, embora não possa ser utilizada simultaneamente com a correção de erros e o entrelaçamento.

Devido à necessidade de configuração de dezenas de registradores é recomendado que se utilize um software de auxílio na configuração, chamado SmartRF® [9]. Este software

calcula os valores dos registradores de configuração do modem, dos filtros, do sintetizador de frequência, o valor dos desvios de frequência, velocidade de transmissão, dentre outros.

2.3.5. RSSI e LQI

A utilização do RSSI (*Recived Signal Strength Indicator*) e do LQI (*Link Quality Indicator*), dentre outros recursos, além de serem métricas importantes para medir a qualidade do link, podem ser utilizados para aumentar o alcance no qual a palavra de sincronismo é recebida com sucesso.

Durante a transmissão, os primeiros *bits* transmitidos são os bits do preâmbulo (Figura 2.10), após o envio dos bytes de preâmbulo a palavra de sincronismo (dois ou quatro bytes) é enviada e posteriormente os dados de informação serão transmitidos. O receptor por sua vez, fica aguardado a chegada dos *bytes* de preâmbulo. Ao identificar o preâmbulo o receptor se prepara para receber a palavra de sincronismo, se os bytes da palavra de sincronismo conferir com os *bytes* de sincronismo armazenados no receptor, o pacote é tratado pelos próximos estágios, caso contrário será descartado. Não necessariamente, todos os bits da palavra de sincronismos devem ser recebidos com sucesso, é possível configurar de maneira que 15 dos 16 bits confirmem ou 30 dos 32 bits sejam recebidos com sucesso. O aumento na tolerância a erros na palavra de sincronismo pode ser usado para aumentar o alcance do link [5].

Outro recurso que pode ser utilizado é o PQT (*Preable Quality Threshold*), este indicador opera sobre os bits do preâmbulo. O preâmbulo nada mais é que uma sequência alternada de 1's e 0's transmitida antes do envio do pacote. O PQT funciona incrementando um contador interno cada vez que o receptor recebe um bit diferente do bit recebido anteriormente, e decrementa o contador por 8 cada vez que o bit recebido é igual ao bit anterior. O valor do limiar é quatro vezes o valor configurado no registrador PKTCTRL1 [5]. A palavra de sincronismo que for precedida de um preâmbulo que ultrapassou o limiar será verificada e se estiver correta o pacote passará para os próximos estágios, caso contrário o preâmbulo é descartado e o receptor esperará o próximo.

Uma das vantagens deste dispositivo sobre seus concorrentes é a presença de um registrador RSSI e outro LQI. Estes registradores são utilizados para se ter uma estimativa do nível de potência e da qualidade do sinal recebido, sendo importantes no momento da

decisão de interromper o link de comunicação para evitar o recebimento de dados corrompidos. O RSSI é medido continuamente até que a palavra de sincronismo seja recebida, então seu valor é mantido até a próxima vez que o chip entrar no modo RX. O valor do RSSI quando calculado em dBm possui uma precisão de 0,5 dB. Quando lido diretamente do registrador de *status* destinado a ele, o valor estará na notação complementar a dois. Para converter o valor lido do registrador em dBm, basta:

- Converter o valor do registrador de hexadecimal para decimal ($RSSI_{dec}$)

- Se $RSSI_{dec} \geq 128$,

$$RSSI_{dBm} = \frac{RSSI_{dec} - 256}{2} - RSSI_{offset} ;$$

- Caso contrário, Se $RSSI_{dec} < 128$ então,

$$RSSI_{dBm} = \frac{RSSI_{dec}}{2} - RSSI_{offset} .$$

O valor do offset vai depender da taxa de transmissão, a Tabela 2.6 mostra os valores típicos do $RSSI_{offset}$.

Tabela 2.6 – Valores típicos para o $RSSI_{offset}$.

Taxa de transmissão (kBaud)	$RSSI_{offset}$ (dB)
2,4	71
10	69
250	72
500	72

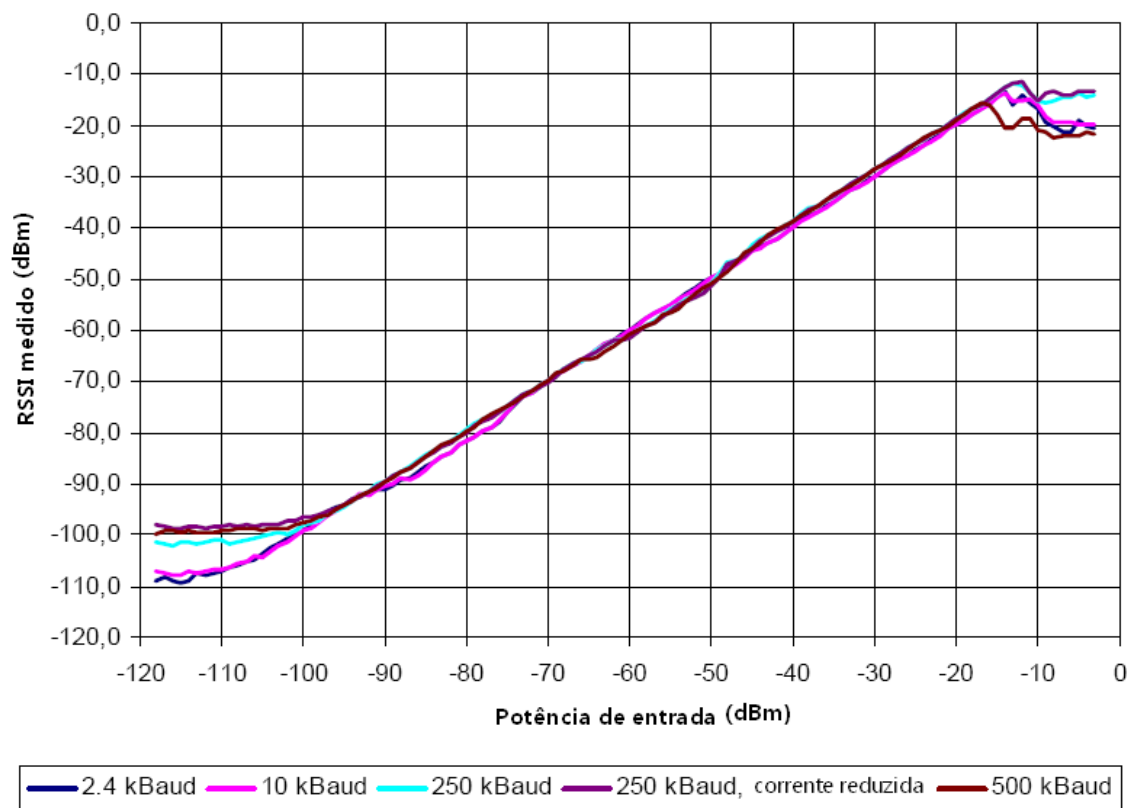


Figura 2.11 – RSSI medido em função da Potência de entrada, para algumas taxas de transferência.

A Figura 2.11 mostra o gráfico que relaciona a potência recebida com o valor medido pelo RSSI para diferentes taxas de transferências. Observa-se que existe uma região bem linear que compreende as potências entre -95 dBm a -25 dBm, nessa faixa o valor medido pelo RSSI é bem próximo a potência de entrada no dispositivo, o que torna a medida mais confiável.

O LQI também é uma métrica que está relacionada com a qualidade do sinal recebido, ele indica o quão facilmente um sinal recebido será demodulado.

Neste capítulo foi apresentado o dispositivo eZ430-RF2500, seus componentes principais e os recursos disponíveis. No próximo capítulo é mostrado um estudo do dispositivo do ponto de vista da propagação de sinais, onde é avaliado seu comportamento quando operando em ambientes *indoor* e *outdoor*.

3. Ensaios de propagação

Neste capítulo serão abordados aspectos da propagação de sinais na faixa de 2,4 GHz. Quando se observa a transmissão de ondas eletromagnéticas, dois comportamentos distintos podem ser percebidos. Um deles é o macro-comportamento, que está associado a incrementos significativos na distância entre o transmissor e o receptor, o outro é o micro-comportamento, que por sua vez está associado a pequenos incrementos nessa distância.

É sabido da literatura que as ondas eletromagnéticas sofrem atenuação significativa com o incremento da distância, por tanto, encontrar meios de prever esse comportamento tem sido um desafio durante anos. Alguns modelos que tentam prever o macro-comportamento da propagação de sinais serão abordados aqui.

Apesar de utilizarem fatores determinísticos, os modelos que serão apresentados também possuem caráter empírico. Além disso, esses modelos se dividem em duas categorias, dependendo do ambiente de propagação. Os modelos destinados a ambientes abertos (*outdoor*) e a ambientes fechados (*indoor*).

O objetivo deste capítulo é introduzir alguns desses modelos e observar o seu comportamento para a aplicação apresentada nessa dissertação, além de verificar a eficiência dos mecanismos de equalização e modulação disponíveis no dispositivo utilizado (eZ430-RF2500), avaliando o alcance do *link*.

3.1. A propagação de Sinais

Muitos efeitos que surgem durante a propagação de sinais estão associados com fenômenos bastante conhecidos, como por exemplo, a difração e a reflexão [10][11]. Durante a propagação de uma onda eletromagnética, esses fenômenos atuam em conjunto

interferindo na propagação direta da onda. Assim é notável a complexidade em modelar tais sinais.

A modelagem de sinais toma por base a medição da potência ao longo do percurso percorrido pelo sinal. No caso da propagação em espaço livre, a equação de Friis (Equação 4.1) [11] pode ser utilizada para estimar o nível de potência do sinal recebido a uma determinada distância do transmissor. Essa equação tem como parâmetros a potência transmitida (P_T), o ganho da antena transmissora (G_T) e receptora (G_R), a distância que separa o transmissor e o receptor (d) e o comprimento de onda (λ) do sinal.

$$P_R = P_T \frac{G_T G_R \lambda^2}{4\pi^2 d^2}. \quad (3.1)$$

Porém, essa equação não leva em conta o fenômeno da reflexão da onda eletromagnética, quando essa se propaga próxima ao solo, por exemplo. Assim, modelos mais completos que levam em consideração os fenômenos como reflexão, difração, espalhamento, etc., possuem elementos de caráter determinístico e também de caráter empírico.

3.2. Modelos para propagação de sinais em ambientes *outdoors*.

3.2.1. Modelo de Young

O Modelo de Young foi definido para uma faixa de frequência que vai de 150 MHz a 3.700 MHz [11][12]. Nesse modelo, é considerado que a potência do sinal transmitido decai com a quarta potência. Do ponto de vista da perda de percurso, ou do inglês “*path-loss*”, que representa a relação em decibéis entre potência transmitida e a potência recebida, observada na Equação (3.2), o modelo de Young é descrito pelos parâmetros vistos na Equação (3.3).

$$PL_{dB}(d) = 10 \log \frac{P_t}{P_r}, \quad (3.2)$$

$$PL_{dB}(d) = \frac{d^4}{G_t G_r (h_t h_r)^2 \beta}. \quad (3.3)$$

Na qual, d é a distância que separa o transmissor do receptor (em metros) G_t e G_r representam o ganho da antena do transmissor e do receptor respectivamente, e h_t e h_r são as alturas do transmissor e receptor respectivamente. O parâmetro β é chamado de fator de desordem (*clutter factor*).

3.2.2. Modelo do logaritmo da distância

No modelo do logaritmo da distância é baseado na escolha de um ponto de referência cujo trajeto entre o transmissor e esse ponto de referência não existem obstáculos. A partir deste ponto de referência, considera-se que a perda de sinal se dá por meio de um fator de atenuação (n) obtido empiricamente como mostrado na Equação 3.4. A distância de referência é representada por d_0 , e d representa a distância entre transmissor e receptor.

$$PL(d) \propto \left(\frac{d}{d_0} \right)^n. \quad (3.4)$$

A Equação de Friis pode ser utilizada para representar as perdas da potência do sinal que ocorrem do transmissor até o ponto de referência, basta considerar que as antenas possuem ganho unitário. Sendo assim, a perda total entre transmissor e receptor é dada por:

$$PL(d)_{dB} = PL(d_0)_{dB} + 10n \log \left(\frac{d}{d_0} \right). \quad (3.5)$$

É comum considerar a distância de referência como sendo 1 m, por motivos de simplificação.

3.2.3. Modelo 2-raios

O modelo de 2-Raios leva em consideração o sinal recebido por meio da transmissão direta e o sinal que é refletido no solo, provocando interferências no receptor (Figura 3.1). A intensidade dos sinais transmitidos e refletidos podem ser relacionadas pelo coeficiente de reflexão de Fresnel (Γ) [11][13]. Esse coeficiente depende de fatores como o ângulo de incidência (θ_i), permissividade elétrica do meio (ϵ_r) e da polarização do sinal, como mostrado nas Equações (3.6) e (3.7).

$$\Gamma_{Hor} = \frac{\sin(\theta_i) - \sqrt{\epsilon_r - \cos^2(\theta_i)}}{\sin(\theta_i) + \sqrt{\epsilon_r - \cos^2(\theta_i)}}, \quad (3.6)$$

$$\Gamma_{Ver} = \frac{\epsilon_r \cdot \sin(\theta_i) - \sqrt{\epsilon_r - \cos^2(\theta_i)}}{\epsilon_r \cdot \sin(\theta_i) + \sqrt{\epsilon_r - \cos^2(\theta_i)}}. \quad (3.7)$$

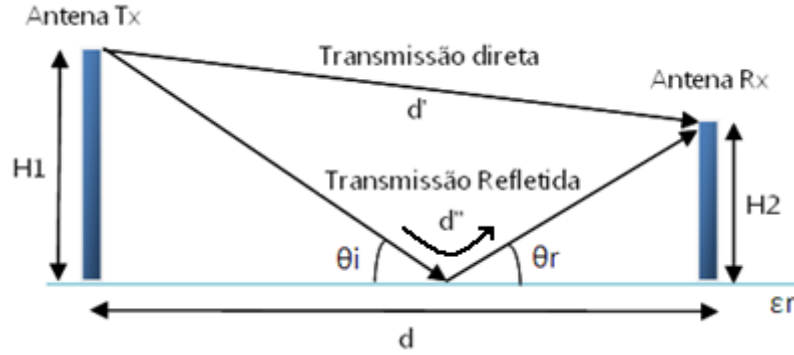


Figura 3.1 – Modelo de Propagação de 2-Raios.

Os cálculos das distâncias percorridas pelo sinal na transmissão direta e pelo sinal refletido são mostrados nas Equações (3.8) e (3.9) respectivamente.

$$d' = \sqrt{d^2 + (H_1 - H_2)^2}, \quad (3.8)$$

$$d'' = \sqrt{d^2 + (H_1 + H_2)^2}, \quad (3.9)$$

$$D_{dif} = d'' - d'. \quad (3.10)$$

A diferença de fase entre o sinal transmitido e o sinal refletido pode ser calculada pela diferença de percurso dos sinais [11], como indicado na Equação (3.11).

$$\theta_{dif} = \frac{D_{dif} 2\pi}{\lambda}, \quad (3.11)$$

A potência do sinal recebido é então dada pela soma entre a potência recebida devido à transmissão direta e a transmissão refletida. Considerando a polarização horizontal, tem-se:

$$P_{total} = P_{dir} + \cos\left(D_{dif} \cdot \frac{2\pi}{\lambda}\right) \cdot \frac{\Gamma_{Hor}}{|\Gamma_{Hor}|} \cdot P_{ref}, \quad (3.12)$$

em que,

$$P_{dir} = P_T \frac{G_T G_R \lambda^2}{(4\pi \cdot d')^2}, \quad (3.13)$$

$$P_{ref} = P_T \frac{G_T G_R \lambda^2}{(4\pi d'')^2 \cdot |\Gamma_{Hor}|}. \quad (3.14)$$

O valor da potência do sinal convertida em dBm é calculado por meio da Equação (3.15).

$$P_{dBm} = 10 \log \left(\frac{P_{total}}{1mW} \right). \quad (3.15)$$

3.3. Modelos para propagação de sinais em ambientes *indoors*.

A modelagem da propagação de sinais em ambientes *indoor* pode ser apresentada de duas formas: específica para um local e para locais genéricos. A específica para um local utiliza o mapeamento mais detalhado do ambiente (móvelia, material das estruturas, etc.) e a para locais genéricos utiliza aproximações mais gerais sobre o comportamento do sinal no ambiente [12]. As específicas para o local são mais complexas, portanto não são muito utilizadas. Aqui, são apresentados modelos mais genéricos, devido ao objetivo deste trabalho.

3.3.1. Modelo ITU

O modelo ITU [14] (*International Telecommunication Union*) é uma evolução do modelo do logaritmo da distância (Seção 3.2.2). Esse modelo leva em consideração a quantidade de pisos existentes entre transmissor e receptor, conforme indicado na Equação (3.16).

$$PL(d)_{dB} = 20 \log(f) + N \log(d) + Lf(m) - 28dB, \quad (3.16)$$

Em que, d é a distância entre o transmissor e o receptor em metros, f é a frequência do sinal (em MHz), N é o coeficiente de perda com a distância, $Lf(m)$ é o fator de perda por piso e m é o número de pisos entre o transmissor e o receptor. A Tabela 3.1 mostra alguns valores para o coeficiente de perda, N , com relação ao tipo de ambiente de propagação e a frequência do sinal.

Tabela 3.1 – Valor para o coeficiente de perda (N) para alguns ambientes.

Frequência	Residencial	Escritório	Comercial
900 MHz	-	33	20
1,2 a 1,3 GHz	-	32	22
1,8 a 2 GHz	28	30	22
4 GHz	-	28	22
5,2 GHz	-	31	-

Na Tabela 4.2 são mostradas as expressões para o cálculo do fator de perda por piso em função da frequência do sinal e do tipo de ambiente de propagação.

Tabela 3.2 – Valores do Fator de Perda por Piso $L_f(m)$.

Frequência	Residencial	Escritório	Comercial
900 MHz	-	9 (m=1) 19 (m=2) 24 (m=3)	-
1,8 a 2 GHz	4m	15 + 4(m-1)	6 + 3(m-1)
5,2 GHz	-	16 (apenas para m=1)	-

3.3.2. Logaritmo da distância (considerando obstáculos)

Este é outro modelo utilizado para descrever o macro-comportamento da propagação de sinais em ambientes *indoor*, e é resumido em [15]. Nesse modelo a influência de paredes, pisos e divisórias, que estejam no caminho da propagação, é levada em consideração. A expressão para perda de percurso é indicada pela Equação (3.17).

$$PL(d)_{dB} = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) + FAF_{dB} + pSPA_{dB} + qWAF_{dB}. \quad (3.17)$$

A influência de pisos, divisórias e paredes é representada pelos termos FAF (*Floor Attenuation Factor*), SPAF (*Soft Partition Attenuation Factor*) e WAF (*Wall Attenuation Factor*), respectivamente. A quantidade desses obstáculos é representada pelas variáveis p e q . Considerando um exemplo onde não há mudança de piso e nem divisórias, a expressão para perda de percurso se resume a:

$$PL(d)_{dB} = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) + q \cdot WAF_{dB}. \quad (3.18)$$

Em [16] encontra-se uma tabela com valores típicos para as atenuações imposta por obstáculos, para sinais na faixa de 2,4GHz.

3.3.3. Modelo de Souza e Lins

O modelo de Souza e Linz [15] leva em consideração a umidade relativa do ar em sua composição. Este modelo comprova que, além da distância, a umidade relativa fornece informação relevante na predição do comportamento da propagação de sinais. Esse modelo é válido tanto para ambientes *indoor* como para ambientes *outdoors*. Por necessitar de um equipamento adicional, capaz de medir a umidade relativa do ar, esse modelo é apresentado apenas para informação. A Equação (3.19) apresenta a formulação matemática desse modelo.

$$PL(d, UR)_{dB} = +\beta_1 \log(d) + \beta_2 d + \beta_3 \log(UR). \quad (3.19)$$

Em que β_i são os coeficientes de atenuação, d é a distância e UR representa a umidade relativa do ar. Os coeficiente β_i são positivos, $0 < UR < 1$, $d \geq 1$ metro (distância de referência).

3.4. Ensaios de Propagação (*Outdoor*)

O que motivou a execução dos ensaios de propagação foi a definição do alcance médio de cada célula do sistema de localização desenvolvido neste trabalho. Com a realização dos ensaios, é possível ter uma dimensão do comportamento do dispositivo, tanto em ambientes abertos (*outdoor*) como em ambientes fechados (*indoor*). Assim, é possível definir o nível de potência do transmissor (em média) que seja capaz de cobrir uma área desejada.

3.4.1. Metodologia do Ensaio

Os ensaios foram realizados utilizando o dispositivo Texas eZ430-RF2500 descrito na seção 2.1. Esse dispositivo foi programado (detalhes no Apêndice A) para enviar e receber um pacote de dados com um *byte* de comprimento e enviar, por meio de conexão USB, para um Notebook os valores das potências recebidas pelo receptor e pelo transmissor (registrador RSSI). O dispositivo também foi configurado para operar utilizando a

modulação FSK a uma taxa de 250 KBaud, além disso, também foram feitos testes com o recurso PN9 ativado. O recurso PN9 (*data whitening*) corresponde à aplicação de uma sequência pseudo-aleatória aos dados transmitidos. Essa sequência tende a melhorar o desempenho do receptor, como explicado na seção 2.3.3.

A janela do *software* desenvolvido e utilizado para aquisição dos dados é mostrado na Figura 3.2.

O experimento, realizado em ambiente *outdoor*, ocorreu no estacionamento do Centro de Tecnologia e Geociência da UFPE, no período da noite de maneira que não houvesse carros ou pessoas circulando. O transmissor e o receptor ficaram sempre em linha de visada e foram recolhidas cerca de dez amostras para cada distância. A medida final foi dada pela média aritmética dos valores obtidos.

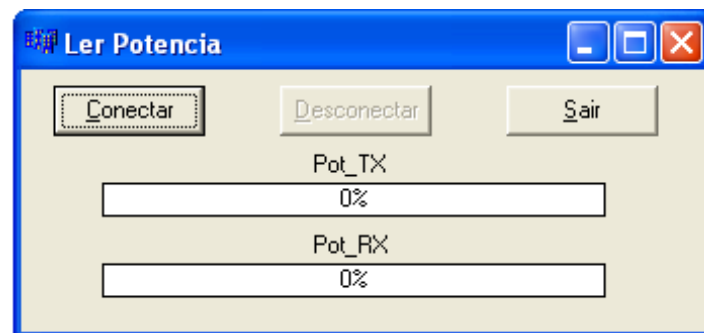


Figura 3.2 – Janela do *software* de aquisição das potências.

3.4.2. Resultados das medições

Na Tabela 3.3 podem ser visto os resultados das medições realizadas, onde estão apresentadas as potências recebidas pelo transmissor e pelo receptor. Como dito anteriormente, o dispositivo foi programado para enviar um *byte* ao receptor, que por sua vez devolveria a resposta com uma confirmação (*acknowledgement*). Assim os valores da tabela representam o valor da potência do sinal no recebimento do *byte* enviado e recebido.

A potência de transmissão configurada no dispositivo foi de 0 dBm. Na Tabela 3.3 observa-se que para distância zero a potência média é de aproximadamente -17 dBm. Essas perdas se devem provavelmente ao acoplamento entre as antenas e pelo fato delas serem onidirecionais.

Outro detalhe importante ocorre entre 24 m e 28 m de distância entre transmissor e receptor, nessa distância houve uma interrupção no recebimento do sinal. Isso se deve a interferências construtivas e destrutivas que impõe ao sinal uma oscilação em distâncias próximas ao transmissor. Essas oscilações fazem o sinal cair abaixo do nível de sensibilidade do receptor, dando a impressão que o sinal chegou ao seu limite. Porém aumentando a distância, o sinal volta a aparecer. Essas oscilações são previstas no modelo de 2-Raios.

Tabela 3.3 – Medições *outdoor* utilizando a modulação FSK e taxa de 250 kBaud.

Distância	Sem PN9		Com PN9	
	Potência do Transmissor[dBm]	Potência do Receptor[dBm]	Potência do Transmissor [dBm]	Potência do Receptor [dBm]
0	-16	-18	-17	-18
2	-66	-66	-66	-65
4	-73	-75	-71	-70
6	-77	-77	-80	-80
8	-80	-81	-73	-73
10	-74	-75	-72	-72
12	-82	-82	-81	-80
14	-80	-81	-77	-76
16	-79	-80	-75	-75
18	-78	-79	-78	-77
20	-79	-80	-78	-77
22	-83	-84	-83	-83
24	-82	-84	Sem Sinal	Sem Sinal
26	Sem sinal	Sem sinal	-84	-83
28	-84	-84	-85	-84
30	-85	-86	-82	-81
33	-81	-81	-83	-82
38	-84	-85	-84	-84
43	-85	-85	-82	-81
48	-86	-86	-83	-83
53	-86	-87	-84	-84
58	-84	-85	-86	-86
63	-83	-85	-87	-86
68	-86	-87		
78	-87	-88		
88	-87	-88		
98	-87	-89		

3.4.3. Análise dos resultados

Como pode ser observado na Figura 3.3 o modelo de Young não foi bem representativo e apresentou um erro bastante grande, $RMSE = 16,5 \text{ dB}$ e $RMSE_PN = 15,7 \text{ dB}$. A Raiz do Erro Médio Quadrático ($RMSE - Root \text{ Mean Square Error}$) é dada pela Equação (3.20), e é uma medida utilizada aqui, para avaliar o quão próximo dos pontos teóricos os pontos experimentais ficaram. Quanto menor o RMSE mais representativo é o modelo.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_{1,i} - x_{2,i})^2}{n}}. \quad (3.20)$$

No modelo de Young as oscilações provocadas pelas interferências não são representadas. No caso dessa modelagem, foi considerado que as antenas possuem ganho unitário e que o parâmetro é $\beta=0,03$. Esse valor do parâmetro foi escolhido de modo que a curva do modelo se aproximasse ao máximo dos pontos experimentais. A consideração do ganho unitário, para as antenas, pode ser um dos motivos da não adequação do modelo, mas pra efeito de simplificação isso foi considerado.

Nota-se também uma melhora no nível do sinal recebido com a utilização do recurso PN9 e em alguns pontos (entre 40 e 60 metros) ele acompanhou os dados do modelo.

No gráfico comparativo entre o modelo do Logaritmo da Distância e os dados experimentais (Figura 3.4), também observa-se uma discrepância entre os valores teóricos e experimentais, embora sejam mais representativos que o modelo de Young. Os valores do RMSE sem a utilização do *data whitening* para esse modelo foi de $RMSE = 9,4 \text{ dB}$ e com a utilização do *data whitening* foi de $RMSE_PN9 = 8,8 \text{ dB}$. O modelo do logaritmo da distância apresenta-se como uma boa opção para modelar o comportamento da propagação de sinais em ambientes abertos (*outdoor*). Como parâmetros para esse modelo foi utilizado o fator de atenuação $n=2,5$. Vale lembrar que o valor teórico indicado para esse parâmetro é 2. O cálculo do fator $PL(d_0)_{dB}$ pode ser feito por meio da equação de Friis para o espaço livre, de modo que o valor encontrado para uma frequência de $2,5 \text{ GHz}$, e considerando mais uma vez o ganho das antenas unitário, foi de $40,39 \text{ dB}$.

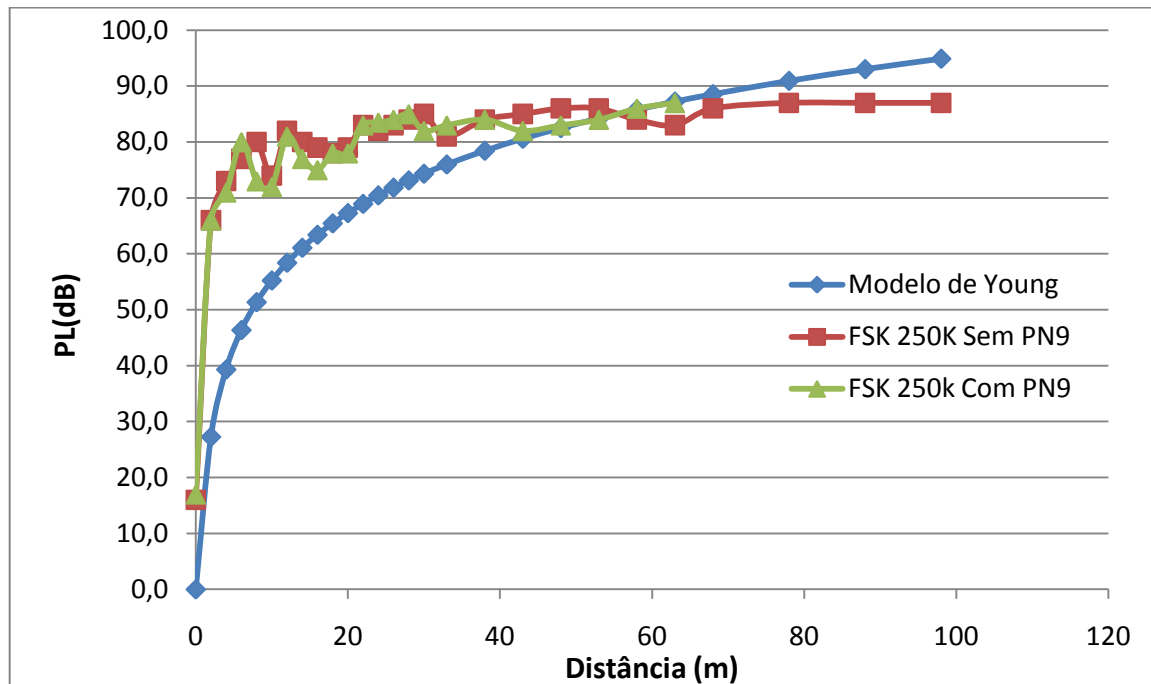


Figura 3.3 – Modelo de Young *versus* Valores Experimentais.

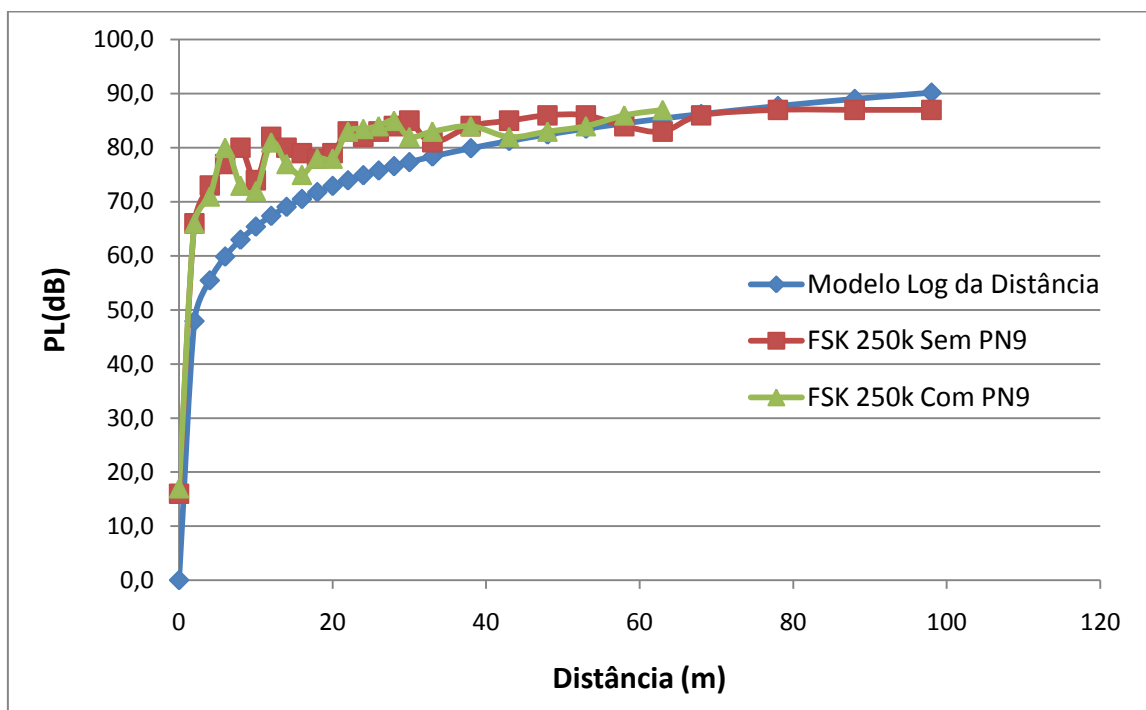


Figura 3.4 – Modelo Logaritmo da Distância *versus* Valores Experimentais.

O último modelo testado para ambientes abertos foi o modelo de 2-Raios. Esse modelo mostrou-se pouco representativo, embora em [17] observa-se que sob condições mais favoráveis ele apresente um desempenho melhor do que o que está sendo exposto aqui. Os valores para o RMSE foram de $14,2\text{ dB}$ na configuração normal e $12,9\text{ dB}$ quando utilizado o recurso PN9, mostrando assim um desempenho médio melhor que o modelo de Young. Questões como polarização da antena, ganho das antenas, altura entre transmissor e receptor, coeficiente de reflexão do solo, etc., são fatores que influenciam na modelagem. Os parâmetros considerados para essa modelagem podem ser vistos na Tabela 3.4.

Tabela 3.4 – Parâmetros Utilizados no Modelo de 2-Raios.

Parâmetro	Valor
P_t	0 dBm
G_t	1
G_r	1
H_t	1,5 m
H_r	1,5 m
ε_r	10

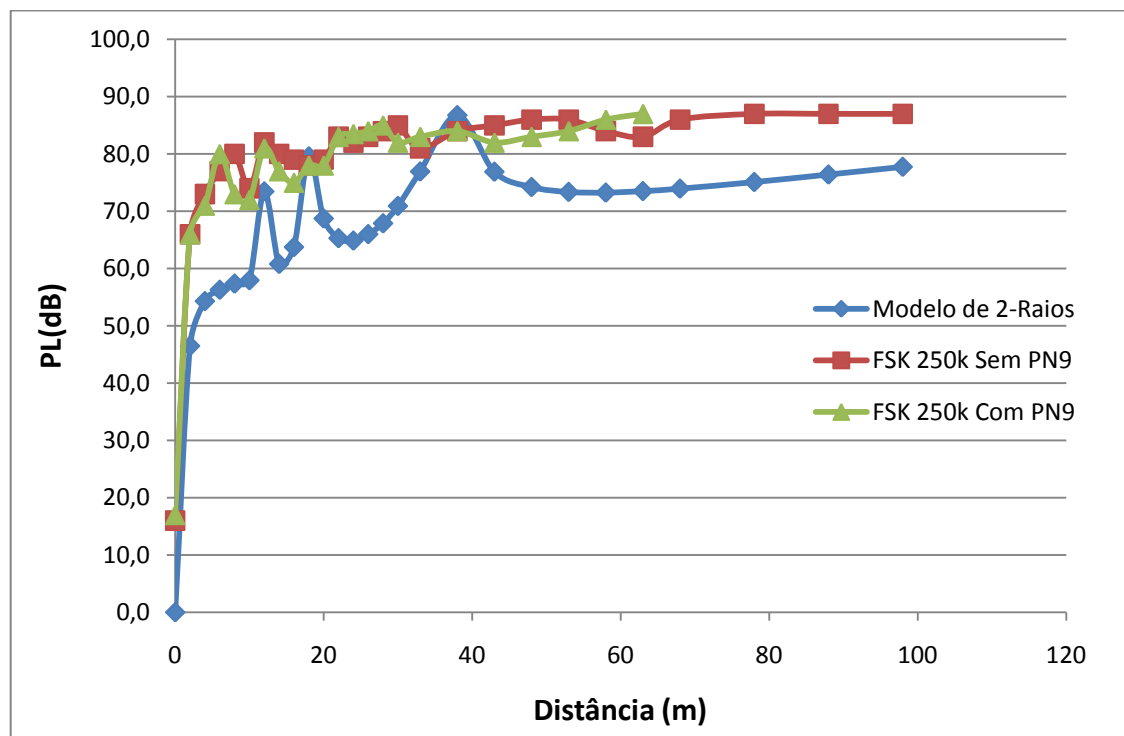


Figura 3.5 – Modelo de 2-Raios *versus* Valores Experimentais.

Além da escolha correta dos parâmetros as condições de realização dos ensaios também influenciam nos resultados, por exemplo, estar em áreas onde existem vários dispositivos e redes sem fio funcionando, aparelhos celulares e outras fontes de ondas eletromagnéticas podem interferir nos resultados. Os ensaios de propagação nesse tipo de ambiente devem levar todos esses critérios em consideração, embora nem sempre isso seja possível como nesse caso, onde sem dúvidas os ensaios foram realizados sob influência de várias fontes de ondas eletromagnéticas.

A modelagem mostrou que os modelos não aderiram muito bem aos resultados, mas esses modelos ainda podem ser usados como um meio de obter uma estimativa do comportamento dos sinais e dispensar a realização de ensaios para definição do alcance dos dispositivos em ambientes abertos.

O modelo de 2-Raios prevê oscilações no sinal e que de fato foram percebidas nos ensaios. Essas oscilações podem ser utilizadas para explicar o aparecimento de uma região de sombra próxima ao transmissor.

A utilização do recurso PN9, que é a aplicação de uma sequência pseudo-aleatória aos dados na saída do transmissor, mostrou-se pouco eficiente no sentido de melhorar a recepção, provocando ainda uma diminuição no alcance com relação a sua não utilização. Uma opção pode ser utilizar esse recurso, em ambientes externos, quando se deseja trabalhar em distâncias curtas.

3.5. Ensaios de Propagação (*Indoor*)

Os ensaios de propagação *indoor* visam definir o alcance, em ambientes internos, do dispositivo mencionado no Capítulo 2. Além disso, duas modulações foram testadas a fim de avaliar o desempenho de cada uma, a FSK e a MSK.

3.5.1. Metodologia do Ensaio

Igualmente ao caso *outdoor* o software (Figura 3.2) foi utilizado para aquisição do nível de potência do sinal recebido, e também a configuração do dispositivo é a mesma utilizada na Seção 3.4. As modulações utilizadas no ensaio foram configuradas para 250 kBaud e com e sem o *data whitening* (PN9).

O experimento foi realizado em uma residência, cuja planta baixa pode ser vista na Figura 3.6.

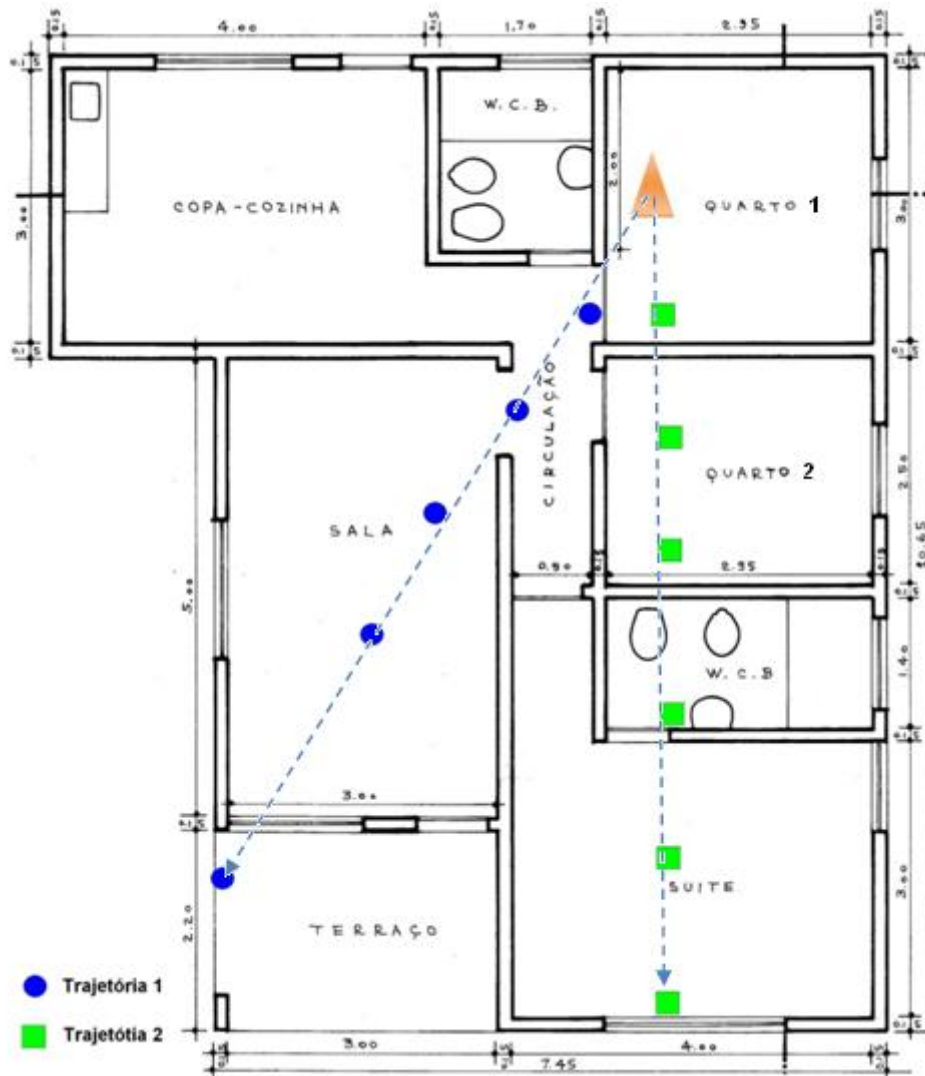


Figura 3.6 – Planta Baixa do local de realização dos ensaios *indoor* (medidas em metros).

Foram realizadas duas baterias de testes, uma seguindo a Trajetória 1, mais livre de obstáculos partindo do quarto 1 em direção a sala, e outra seguindo a Trajetória 2 que atravessa as paredes do quarto 2 e banheiro.

Os modelos de propagação utilizados foram o da ITU e o modelo do Logaritmo da Distância (considerando obstáculos). A métrica usada para comparar os resultados das medições com os resultados previstos pelo modelo foi o RMSE, definido na Equação (3.20).

3.5.2. Resultados e Análise

A parametrização recomendada para o modelo ITU é $N=28$ e o fator de perda por piso não foi desconsiderado, visto que os ensaios foram realizados no mesmo piso. A frequência considerada foi de 2433 MHz . Para o modelo do Logaritmo da Distância com WAF, o parâmetro n recomendado vale 2, a perda de percurso até a distância de referência $PL(d_0)_{dB}=40,39$, conforme mencionado para o caso *outdoor*. A perda por parede foi considerada 6 dB , em [16] encontra-se uma tabela com os valores das perdas para outros tipos de anteparos.

Os valores dos parâmetros mostrados na Tabela 3.5 e Tabela 3.6 foram ajustados de maneira a reduzir o RMSE e estão longe dos valores recomendados.

Tabela 3.5 – Parametrização utilizada para Trajetória 1 do ensaio.

	FSK	FSK_PN9	MSK	MSK_PN9
ITU (N)	58	65	66	64
Log. Dist com WAF (n)	4,9	5,5	5,6	5,5

Tabela 3.6 – Parametrização utilizada para Trajetória 2 do ensaio.

	FSK	FSK_PN9	MSK	MSK_PN9
ITU (N)	68	68	66	68
Log. Dist com WAF (n)	5,5	5,2	4,9	5,1

Tabela 3.7 – RMSE em relação ao modelo ITU para as modulações utilizadas seguindo a Trajetória 1.

Modelo Distância	PL(d)[dB]				
	ITU	FSK	FSK_PN9	MSK	MSK_PN9
1,5 m	49,9	69	78	68	71
3 m	67,4	82	82	83	82
4,5 m	77,6	83	86	89	87
6 m	84,9	84	86	91	91
7,5 m	90,5	84	86	89	90
9 m	95,1	87	-	93	92
	RMSE	10,9	14,08	10,2	10,7

As tabelas 3.7 e 3.8 mostram os valores do RMSE quando comparando o modelo ITU e as medições realizadas com o dispositivo eZ430-RF2500, configurado para as modulações indicadas, e também nessas modulações utilizando o recurso *data whitening* (PN9). Da Tabela 3.7 observa-se um RMSE médio acima de 10 dB, e no caso da modulação FSK Com PN9 o sinal ficou abaixo da sensibilidade do receptor, provocando a perda do *link*, por tanto, não atingindo os 9 m de distância entre transmissor e receptor. Em todo caso, a modulação MSK apresentou o melhor desempenho, conseguindo receber sinais com -93 dBm de intensidade, enquanto que o recurso PN9 não se mostrou decisivo na melhoria do sinal recebido.

Tabela 3.8 - RMSE em relação ao modelo ITU para as modulações utilizadas seguindo a Trajetória 2.

	PL(d)[dB]				
Modelo	ITU	FSK	FSK_PN9	MSK	MSK_PN9
Distância					
1,5 m	51,7	70	74	67	74
3 m	72,2	80	85	79	85
4,5 m	84,1	86	87	88	87
6 m	92,6	83	84	83	84
7,5 m	99,2	-	93	93	93
9 m	104,6	-	-	-	-
	RMSE	11,1	12,5	9,14	12,52

Na Trajetória 2, com um maior número de obstáculos é possível observar, por meio da Tabela 3.8, que o alcance máximo foi de 7,5 m e no caso da modulação FSK foi ainda menor, 6 m. Outros dispositivos que operam, por exemplo com o padrão 802.11, ultrapassam facilmente essa distância sob as mesmas condições. Isso se deve ao protocolo utilizado, tipo de modulação, antena, potência transmitida, dentre outros fatores.

Tabela 3.9 - RMSE em relação ao modelo Logaritmo da Distância com WAF para as modulações utilizadas seguindo a Trajetória 1.

Modelo Distância	PL(d)[dB]				
	Log. Distancia com WAF	FSK	FSK_PN9	MSK	MSK_PN9
1,5 m	55,0	69	78	68	71
3 m	69,8	82	82	83	82
4,5 m	78,4	83	86	89	87
6 m	84,5	84	86	91	91
7,5 m	89,3	84	86	89	90
9 m	93,1	87	-	93	92
	RMSE	8,5	11,5	7,8	8,2

Tabela 3.10 - RMSE em relação ao modelo Logaritmo da Distância com WAF para as modulações utilizadas seguindo a Trajetória 2.

Modelo Distância	PL(d)[dB]				
	Log. Distancia com WAF	FSK	FSK_PN9	MSK	MSK_PN9
1,5 m	55,0	70	74	67	74
3 m	69,8	80	85	79	85
4,5 m	78,4	86	87	88	87
6 m	84,5	83	84	83	84
7,5 m	89,3	-	93	93	93
9 m	93,1	-	-	-	-
	RMSE	12,4	14,3	11,2	14,3

O modelo do Logaritmo da Distância com WAF se mostrou mais representativo que o modelo da ITU, para o caso da Trajetória 1 (Tabela 3.9), pois o RMSE médio ficou abaixo dos 10 dB. Para o caso da Trajetória 2 (Tabela 3.10) o mesmo fato não foi verificado, o RMSE médio foi maior do que 12 dB. Uma justificativa pode ser o fator da atenuação por paredes WAF considerada (6 dB). As paredes do banheiro são mais grossas que as demais, devido à cerâmica, além disso, nesse recinto existem tubulações de água que absorvem o sinal de microondas.

Os resultados obtidos com os ensaios sugerem que a utilização da modulação MSK é mais adequada, visto que na maioria dos casos foi possível atingir a distância de 9 m sem a perda do *link*. A utilização do recurso *data whitening* não influenciou como esperado na melhoria do nível de sinal recebido, por tanto, sua utilização apenas consome mais recursos de processamento o que implica em um consumo maior de corrente por parte do

dispositivo. Os modelos da ITU e do Logaritmo da Distância com WAF não foram representativos para a aplicação de interesse. Esses modelos são mais utilizados em redes sem fio baseadas no padrão IEEE 802.11 na faixa de 2,4 GHz, assim seus parâmetros típicos se adequam melhor a esse padrão.

4. Redes de Sensores Sem Fio

O desenvolvimento de dispositivos e aplicações que utilizam a faixa de frequência não licenciada ISM (*Industrial, Scientific and Medical*) tem se tornado bastante comum nos últimos anos. Uma linha de aplicações nessa área são as redes de sensores sem fio (RSSF). Nesse capítulo são apresentadas algumas das características desse tipo de rede, que vêm se tornando cada vez mais importantes nas mais diversas áreas de atividade humana. O objetivo principal de uma RSSF é coletar dados e se necessário atuar conforme as informações coletadas, controlando um determinado evento ou monitorando um ambiente. Os sensores conectados a essa rede são utilizados para coletar as informações que se deseja avaliar, por exemplo, temperatura, umidade, pressão, volume, etc. Muitas aplicações que utilizam as redes de sensores sem fio precisam de informações sobre a posição dos terminais da rede. Essa necessidade de localizar um nó dentro da rede, também é tratada neste capítulo. Além disso, são mostradas algumas técnicas para localização de um elemento dentro de uma RSSF.

4.1. As Redes de Sensores Sem Fio (RSSF)

Na literatura é comum encontrar várias definições sobre as redes de sensores sem fio, dependendo do enfoque. Uma dessas definições classifica uma rede de sensores sem fio como sendo uma rede sem fio formada por um grande número de sensores pequenos e imóveis plantados numa base *ad hoc* para detectar e transmitir alguma característica física do ambiente [18]. Outra definição que toma uma perspectiva diferente diz que uma rede de sensores sem fio é um conjunto de nós individuais (sensores) que operam sozinhos, mas

que podem formar uma rede com o objetivo de juntar as informações individuais de cada sensor para monitorar algum fenômeno, e esses nós podem se mover juntamente com o fenômeno observado [18]. Em aplicações onde há a necessidade da mobilidade dos nós da rede, como por exemplo, o monitoramento do comportamento de animais, o primeiro enfoque citado se mostra incompleto, visto que esse não prevê a movimentação dos nós da rede. Portanto, para o foco desta dissertação a segunda definição é mais adequada.

As redes de sensores sem fio apresentam várias vantagens em relação aos sensores tradicionais. Facilidade na instalação, economia com cabeamento, aplicação em ambientes hostis, são algumas das vantagens da implantação desse tipo de rede. Essas redes têm sido viabilizadas pela rápida convergência de três tecnologias: microcontroladores, comunicação sem fio e micro sistemas eletro-mecânicos (MEMS – *Micro Eletro-Mechanical Systems*).

O projeto de uma RSSF leva em consideração muitos fatores, dentre eles a tolerância a falhas, escalabilidade, custo de implantação, ambiente operacional, restrições de *hardware*, meio de transmissão e consumo de energia. Todos esses fatores necessitam de requisitos específicos durante a concepção dos nós da rede. Além disso, para cada camada da pilha de protocolos também deve-se atender as exigências de requisitos [19]. A camada física fica encarregada das definições do tipo de modulação, estratégias para autonomia de energia, projetos para reduzir o custo do *hardware* e estratégias para superar os efeitos da propagação de sinais. A sub-camada de controle de acesso ao meio da camada de enlace de dados é responsável basicamente por evitar colisões na transmissão dos pacotes e pela definição dos esquemas para correção de erros e modos de economia de energia. A camada de rede cuida dos requisitos para mudança de topologia, endereçamento, escalabilidade e interoperabilidade com outras redes. Requisitos de segurança em cada camada da pilha de protocolos também devem ser levados em consideração. Deve ser lembrado que os requisitos citados são influenciados pela aplicação, portanto os parâmetros de configuração, operação e manutenção variam com o objetivo da aplicação.

4.1.1. Caracterização das RSSF

Em geral a classificação de uma RSSF depende do seu objetivo e da área de aplicação. A aplicação definirá as funções exercidas pelos nós da rede e também a arquitetura desses

nós (processador memória, sensores, fonte de energia, etc.), bem como, o tipo de protocolo utilizado na comunicação, os serviços oferecidos, a distribuição dos nós na rede e o tempo de vida da rede.

As redes de sensores sem fio podem ser classificadas conforme sua configuração, o sensoriamento e o tipo de comunicação. A Tabela 4.1, a Tabela 4.2 e a Tabela 4.3 mostram as características das RSSF conforme esses critérios de classificação.

Tabela 4.1 – Classificação das RSSF segundo a configuração.

Configuração		
Composição	Homogênea	Apresentam a mesma capacidade de <i>hardware</i> .
	Heterogênea	Apresenta nós com <i>hardware</i> diferentes.
Organização	Hierárquica	Os nós estão organizados em grupos, esses grupos possuem um líder e podem se organizar hierarquicamente.
	Plana	Os nós não estão organizados em grupos.
Mobilidade	Estacionária	Os nós permanecem na mesma posição onde foram colocados inicialmente, ou seja, os nós não se movem ao longo da rede.
	Móvel	Os nós podem se mover ao longo da rede.
Densidade	Balanceada	A distribuição e concentração dos nós por unidade de área da rede são consideradas ideais para o objetivo da rede.
	Densa	Apresenta uma alta concentração de nós.
	Esparsa	Apresenta uma baixa concentração de nós.
Distribuição	Irregular	Rede que apresenta uma distribuição não uniforme dos nós.
	Regular	Rede que apresenta uma distribuição uniforme de nós.

Tabela 4.2 – Classificação das RSSF segundo o sensoriamento.

Sensoriamento		
Coleta	Periódica	A coleta dos dados de interesse é feita em intervalos regulares. Ex: Monitoramento de fenômenos que ocorrem apenas durante o dia ou durante a noite.
	Contínua	Os dados são coletados continuamente. Ex: medição da temperatura em um processo industrial.
	Reativa	A coleta está vinculada a ocorrência de um determinado evento. Ex: aplicações que detectam a presença de objetos ou movimento de objetos na área monitorada.
	Tempo Real	A maior quantidade de dados possível é coletada no menor intervalo de tempo. Ex: aplicações onde há risco de mortes humanas.

Tabela 4.3 – Classificação das RSSF segundo a Comunicação.

Comunicação		
Disseminação	Programada	Os nós disseminam as informações em intervalos regulares.
	Contínua	Os nós disseminam as informações continuamente.
	Sob Demanda	Os nós disseminam as informações quando solicitados ou na ocorrência de eventos.
Tipo de conexão	Simétrica	Todas as conexões entre os nós possuem o mesmo alcance.
	Assimétrica	As conexões entre os nós possuem alcances diferentes.
Transmissão	Simplex	Os nós apenas transmitem as informações
	Half-duplex	Os nós podem transmitir ou receber informações.
	Full-duplex	Os nós podem transmitir e receber informações ao mesmo tempo.

4.2. Localização em RSSF

O recurso da localização possibilita a existência de inúmeras aplicações que hoje em dia são fundamentais para a vida das pessoas. Não é a toa que vários projetos já foram desenvolvidos e ainda existem muitos outros sendo pesquisados. Um exemplo básico de um sistema de localização que funciona muito bem nos dias atuais é o GPS (*Global Position System*), que começou a surgir em 1973 graças a investimentos realizados pelo Departamento de Defesa dos Estados Unidos e utiliza 24 satélites de órbita média (18.000 km de altitude). [20][21]. Outros sistemas utilizam a rede de celular GSM (*Global System for Mobile Communications*) para localizar um usuário em uma de suas células e oferecer serviços baseados em localização (LBS – *Location Based Services*) [22]. Existem ainda sistemas que utilizam as redes WLAN (*Wireless Location Area Network*) para localização de dispositivos ou pessoas, sendo aplicado principalmente em ambientes *indoor* [23][24]. Cada um desses exemplos possui focos diferentes, embora as técnicas utilizadas para localização de um elemento ou indivíduo são basicamente reaproveitadas.

Em uma rede de sensores, a informação da posição é de fundamental importância para dar mais consistência aos dados coletados, além de ser inerente a muitas aplicações. Uma das formas mais simples de se conhecer a localização de um terminal de uma rede de sensores é a colocação manual desses terminais em locais pré-estabelecidos [25]. Contudo, essa solução não é prática quando se tem um número muito grande de sensores. Equipar os nós da rede com receptores do sistema GPS pode ser uma solução, mas isso implicaria em um alto custo além do maior consumo de energia. Em face dos diversos problemas que

envolvem essa questão, nos últimos anos vários esforços têm sido feitos no sentido de encontrar soluções viáveis do ponto de vista dos custos de implantação, da complexidade e do consumo de energia.

Quanto maior o grau de precisão na localização, requerido pela aplicação, maior será a complexidade do método utilizado ou dos requisitos do sistema. A observação de grandezas físicas que possam ser relacionadas com a localização dos nós da rede são freqüentemente utilizadas como um mecanismo de estimativa da posição. Uma dessas grandezas é a potência da onda eletromagnética que chega ao terminal receptor. Em geral, os dispositivos que compõe uma rede de sensores sem fio são dotados de circuitos que estimam o nível do sinal recebido. O RSSI (*Recived Signal Strength Indicator*) é um indicador da potência com que o sinal chega ao receptor e esse valor pode ser relacionado com a distância percorrida pelo sinal. É importante lembrar que esses sinais sofrem influência do meio de propagação e podem gerar erros na localização, por isso, é necessário o conhecimento de modelos de propagação para melhorar o desempenho do método de localização.

Uma das técnicas de localização que utiliza a medida de potência como base para estimar a posição de um elemento da rede é a triangulação. A triangulação é baseada nas propriedades geométricas dos triângulos e pode ser dividida em *lateração* e *angulação* [20][25].

A *lateração* consiste em calcular a distância de um objeto em relação a múltiplos pontos de referência (pelo menos três). Se as posições dos terminais O_1 , O_2 e O_3 (ver Figura 4.1) são conhecidas basta estimar as distâncias d_1 , d_2 e d_3 , por meio da medida do RSSI e utilizando modelos para propagação de sinais, e em seguida aplicar propriedades geométricas para descobri a localização do terminal X . Esse tipo de *lateração* é baseada na atenuação do sinal transmitido. Existe ainda outro tipo que se baseia na medida do tempo que o sinal leva para ir de um terminal a outro com velocidade conhecida (velocidade da luz) [20], contudo é necessário *hardware* para medida de tempos muito pequenos, o que representa um custo maior.

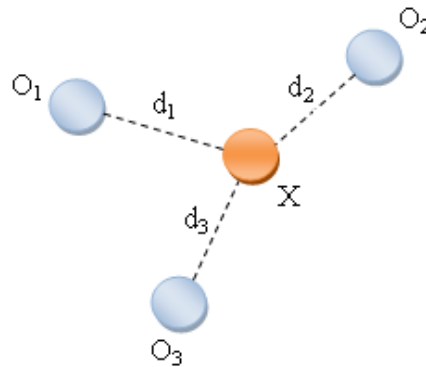


Figura 4.1 – Determinação da localização pelo método da *lateração*.

A *angulação* é uma técnica baseada na medida dos ângulos de chegada dos sinais transmitidos e nas distâncias dos pontos de referência (O_1 e O_2). A Figura 4.2 mostra o caso bidimensional, onde são necessários apenas dois pontos de referência. Para utilização dessa técnica é necessário o uso de antenas direcionais, que irão detectar o ângulo no qual o sinal recebido é mais forte, essas antenas representam um aumento no tamanho dos receptores, um maior consumo de energia e maior complexidade.

Devido à limitada capacidade de processamento e memória dos nós sensores, geralmente os cálculos para determinação da posição são realizados por uma unidade central, que é dotada de um banco de dados contendo informações sobre os nós da rede.

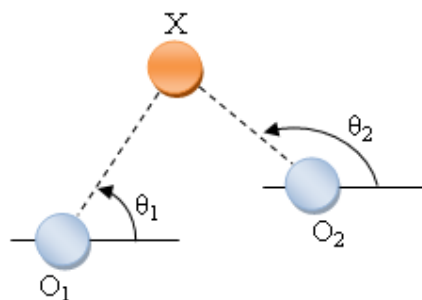


Figura 4.2 – Determinação da localização pelo método da *angulação*.

Na literatura, é possível encontrar diversas técnicas de localização que se baseiam na informação da medida de distância entre os terminais vizinhos, na construção de mapas locais com base na vizinhança para construção posterior de um mapa global, na utilização de algoritmos iterativos que buscam reduzir uma determinada função custo, considerando um fator ponderador para reduzir o erro na localização, dentre outras técnicas também citadas em [25].

Podem-se citar mais duas técnicas simples e livres de medições da potência do sinal, que possuem princípios semelhantes aos utilizados na elaboração do sistema de localização desenvolvido nesse trabalho e que é mostrado no capítulo 5.

A primeira é uma técnica simples, destinada à ambientes externos que consiste na localização de um elemento baseada apenas na conectividade, sem a necessidade de medições do nível de potência [26]. Um dos motivos para o desenvolvimento desse tipo de técnica é que as técnicas baseadas na medição da potência do sinal, não oferecem boas aproximações em ambientes externos e principalmente em curtas distâncias, pois o sinal sofre atenuação devido à interferência por múltiplos percursos. Além disso, existem dispositivos que não são dotados de circuitos medidores de RSSI. Essa técnica consiste na localização de um elemento móvel, baseada no centróide das posições dos terminais fixos, que estão na área de cobertura do terminal móvel e cujas medidas de conectividade ultrapassam um determinado limiar. Quanto mais terminais fixos, maior será a precisão do sistema, embora o custo também aumente. Uma das desvantagens dessa técnica é que ela requer que os terminais fixos formem uma estrutura de malha regular e isso nem sempre é possível na prática.

A segunda técnica conhecida como APIT (*Approximate Point-In-Triangulation*) [27] faz uso da posição previamente conhecida de terminais fixos (terminais-âncora) e da informação da conectividade do elemento que se deseja localizar. A técnica funciona utilizando várias combinações de terminais-âncoras, de maneira que cada combinação forma um triângulo, e dentro deste triângulo é realizada uma busca pelo terminal desconhecido (Figura 4.3). As combinações de terminais-âncoras que não localizam o terminal desconhecido são descartadas, reduzindo a área da provável localização. Essas combinações são testadas a exaustão ou até que a precisão desejada seja alcançada. O próximo passo do algoritmo é calcular o centro de gravidade da região dada pelas intersecções dos triângulos que identificaram a presença do terminal desconhecido. Esse centro de gravidade corresponde à provável localização do terminal desconhecido. Assim como no método anterior, quanto mais terminais-âncora, maior a precisão na localização.

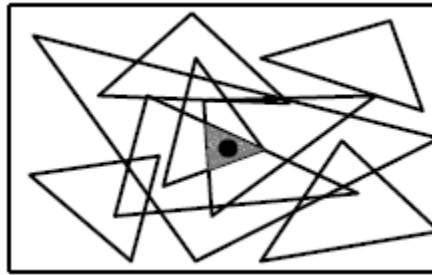


Figura 4.3 – Estruturas do algoritmo APIT.

No caso do sistema de localização desenvolvido nesta dissertação, a localização é dada apenas pela conectividade, e devido ao tipo de aplicação que motivou a elaboração desse sistema, nenhuma técnica especial foi utilizada. A localização é simplesmente dada pela posição do terminal fixo (posição conhecida) ao qual o elemento desconhecido está conectado. Embora o sistema envie informações sobre o nível de potência dos terminais, essas informações não são utilizadas para localização, ou seja, o sistema pode representar uma plataforma para futuras implementações que necessitem dessas informações.

4.3. Aplicações das RSSF

Com o aumento do interesse na área das RSSF cresceu também o número de aplicações decorrentes de pesquisas e desenvolvimento de novos *hardwares* e *softwares*. Hoje existe o envolvimento das RSSF nas áreas de saúde, comércio e indústria, militar, monitoramento ambiental, dentre outras.

Na saúde as redes de sensores sem fio são empregadas no auxílio ao diagnóstico de doenças, monitoramento de pacientes, administração de medicamentos, etc.

No monitoramento de ambientes pode-se citar: o rastreamento de animais, identificação de áreas perigosas afetadas por ameaças químicas ou biológicas, identificação de possíveis desastres ambientais, detecção e prevenção de incêndios, monitoramento de plantações e áreas de difícil acesso, em áreas urbanas o monitoramento da qualidade do ar, do ruído e do tráfego, etc.

Nas áreas de comércio e indústria encontram-se aplicações no monitoramento de infra-estruturas, em linhas de produção, monitoramento de máquinas e processos industriais, automação e controle de acesso, controle de estoque e inventários, estes últimos auxiliam a otimização no fornecimento, distribuição e estoque.

No campo militar são importantes sistemas para comando, controle, comunicação e inteligência. As redes de sensores podem se organizar rapidamente mesmo sem uma infraestrutura pré-estabelecida, além de poderem se integrar facilmente a outros sistemas, sendo esta uma característica desejada por esse tipo de aplicação. Um exemplo é sua aplicação para monitoramento do perímetro de uma base militar em um ambiente hostil. Também podem ser utilizadas para a vigilância em campos de batalha, detecção e reconhecimento de ataques biológicos ou químicos.

Vários projetos que utilizam as RSSF já foram desenvolvidos e outros estão em desenvolvimento ou sendo aperfeiçoados, exemplos desses projetos podem ser encontrados nas seguintes referências: DSRC (*Dedicated Short-Range Communication*) [28], monitoramento de abrigos na ilha de Great Duck [29], DARPA SensIT [30], Detecção de movimentos com a utilização de acelerômetros em RSSF [31], projeto Code Blue [32].

5. Sensor de Presença e Sistema de Localização

Neste capítulo é apresentado, em detalhes, o desenvolvimento de um sensor de presença que faz uso do dispositivo eZ430-RF2500. Esse sensor de presença será utilizado como elemento de um sistema de localização, onde vários desses sensores, são conectados por meio de uma rede local (LAN) de maneira a formar uma rede de sensores de presença. O objetivo principal do sistema de localização, desenvolvido neste trabalho, é o de fornecer elementos necessários à localização de um usuário, dentro da rede de sensores. A informação da localização do usuário, permitirá o redirecionamento de chamadas telefônicas (VoIP) realizadas em um servidor Asterisk [2], ou seja, as chamadas destinadas a um usuário, que faz parte da rede, poderão ser redirecionadas para o ponto da rede mais próximo ao que esse usuário se encontra. O sistema de localização desenvolvido será discutido em detalhes neste capítulo.

5.1. Sensor de Presença

O sensor de presença desenvolvido faz uso do dispositivo eZ430-RF2500, que como dito no Capítulo 2, é composto por duas placas. A primeira é chamada de placa de emulação e a segunda chamada de placa alvo. A placa de emulação é responsável pela gravação dos programas do usuário na placa alvo e pela comunicação da placa alvo com um computador (PC – *Personal Computer*), por meio da interface USB. Existe ainda, um

suporte para baterias no qual, pode-se acoplar uma placa alvo. Esse suporte é mostrado na Figura 5.1.

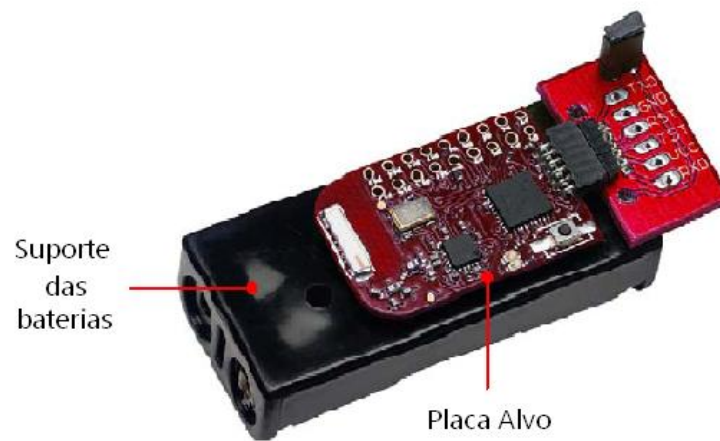


Figura 5.1 – Placa Alvo com suporte a baterias.

A utilização desse suporte, permite que a placa alvo possa se mover pelo ambiente ou ser depositada em um local onde não haja fontes de energia. Nas próximas subseções é apresentada a arquitetura do sensor de presença e detalhes do protocolo de comunicação entre os dispositivos.

5.1.1. Arquitetura do Sensor de Presença

São definidos dois elementos básicos utilizados para a detecção de presença, um deles é chamado de ponto de acesso (PA) e o outro é chamado de elemento móvel (EM). O PA é um dispositivo eZ430-RF2500 (Figura 2.1) e o EM é uma placa alvo alimentada por baterias ou pilhas (Figura 5.1). O PA é o elemento que permanece fixo e está conectado a um computador por meio da interface USB. Por outro lado, o EM pode se deslocar pelo ambiente, por isso a necessidade de ser alimentado por baterias. A comunicação do EM com o PA é feita via *wireless*.

Para identificar a presença de um EM dentro de um ambiente, é necessário que esse ambiente esteja sendo coberto por um PA. Quando um EM entra na área de cobertura do PA, este identifica a presença do EM e o associa a sua lista de dispositivos conectados.

Como mostrado na Figura 5.2 apenas o EM2 está na área de cobertura do PA. Caso o EM2 saia da área de cobertura, a comunicação entre eles será interrompida e o EM2 será desassociado da lista de sensores conectados ao PA.

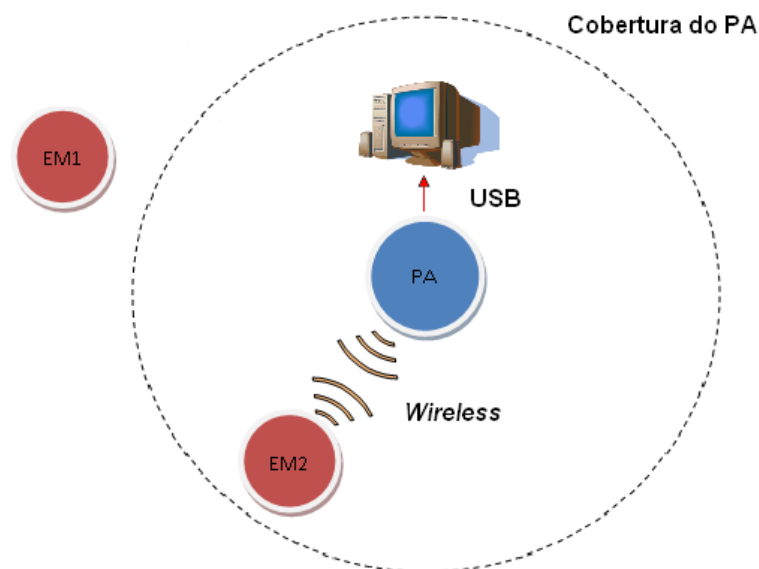


Figura 5.2 – Arquitetura do Sensor de Presença.

5.1.2. Descrição do Protocolo de Comunicação

Todos os PA possuem um endereço único que os identificam, o mesmo ocorrendo com os EM. Cada ponto de acesso suporta no máximo 255 elementos móveis conectados a ele, pois a capacidade de endereçamento do dispositivo é de apenas 8 *bits*. O endereço ZERO não é utilizado como um endereço válido. Uma mudança no protocolo de comunicação pode permitir um aumento da capacidade do PA em se conectar com mais EMs, mais isso não será discutido neste trabalho, visto que o número atual é suficiente.

Repetidamente, o PA executa uma varredura nos endereços previamente cadastrados, enviando uma mensagem (MSG) a cada EM. Caso o EM chamado esteja na área de cobertura desse PA, ele responderá com uma confirmação (ACK) ao chamado (Figura 5.3).

Todos os EMs que estiverem na área de cobertura do PA escutarão o envio da MSG, porém, uma verificação de endereço é realizada e apenas o EM de destino irá responder com o ACK. Nessa situação ocorre um problema encontrado também em outros protocolos, a mensagem recebida é descartada pelos EMs que não são o destinatário, o que

caracteriza um problema conhecido como *overhearing*. Esse problema provoca o desperdício de energia por parte dos EMs, porém em alguns casos não pode ser evitado.

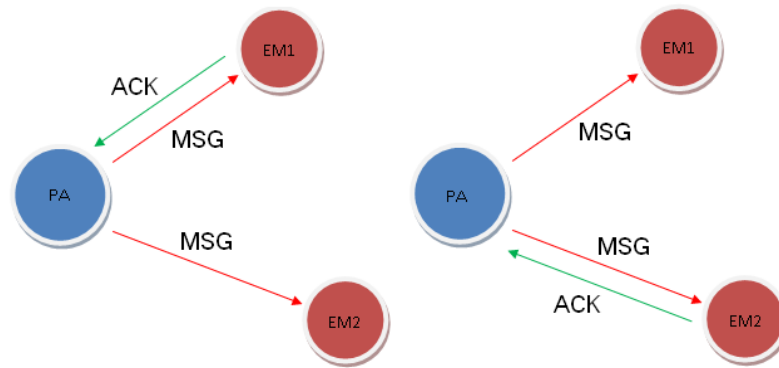


Figura 5.3 – Confirmação de Presença de um EM.

Vários EMs podem estar associados a um único PA, assim para evitar colisões nas respostas dos EM cada confirmação é realizada num determinado *time slot* (Figura 5.4). Devido à natureza assíncrona das confirmações enviadas pelos EM, é destinado um tempo máximo de espera (*timeout*) para o recebimento das respostas, caso nenhuma resposta seja enviada nesse tempo o PA passará para o próximo endereço. O tempo de transmissão vai depender da taxa de transmissão utilizada e do tamanho do pacote. Esse tempo é uma composição dos tempos de calibração do sintetizador de frequência do *chip* de RF (CC2500), adicionado do tempo de transmissão do pacote, não ultrapassando os 2 ms.

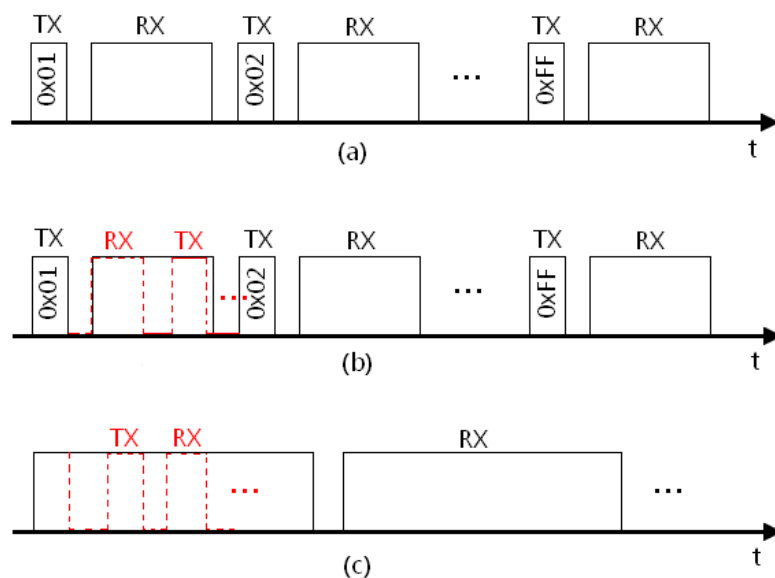


Figura 5.4 – Diagrama temporal TX e RX: (a) Ponto de acesso, (b) Resposta do PA a uma confirmação, (c) Resposta do EM a um chamado do PA.

Os tempos de recepção são variáveis e estão associados ao recebimento de um pacote válido (Figura 5.4b e Figura 5.4c). Após o EM receber um chamado do PA ele entrará em modo de transmissão e enviará uma confirmação dentro do *time slot* reservado para tal, voltando em seguida para o modo de recepção. Imediatamente após a recepção de uma confirmação (ACK) pelo PA, é feito o tratamento dos dados do pacote ACK, o incremento no endereço de busca, a configuração e a entrada no modo de transmissão. Periodicamente é feita uma reconfiguração do EM como receptor. Durante essa reconfiguração é feito também o esvaziamento da pilha de recepção (RX FIFO) e calibração do sintetizador de frequências, isso previne travamentos e melhora o desempenho da recepção. A Figura 5.5 mostra o diagrama de fluxo simplificado das operações de transmissão e recepção do PA e do EM.

Percebe-se que o EM está constantemente escutando o canal, aguardando o recebimento de uma MSG. Assim como no *overhearing*, existe um desperdício de energia com a escuta prolongada do canal, o mais correto seria que o EM escutasse o canal apenas quando houvesse uma mensagem a ser recebida. Embora essa situação não seja tratada pelo protocolo desenvolvido, outros recursos podem ser utilizados para evitar o desperdício de energia, que para esse tipo de dispositivo é essencial.

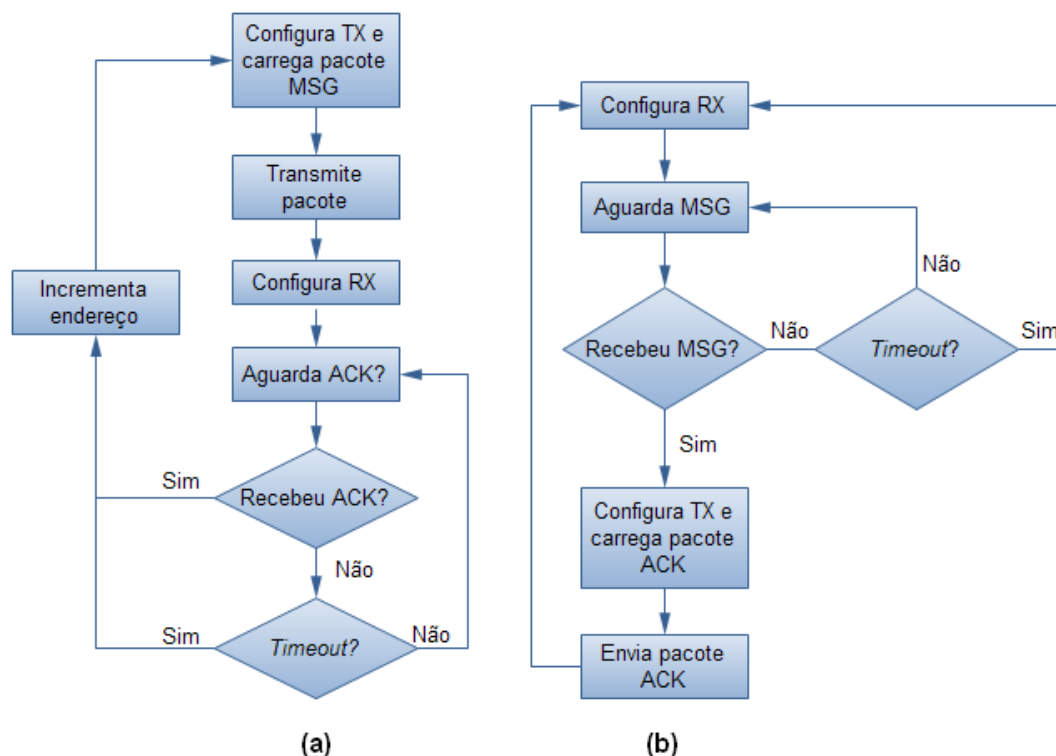
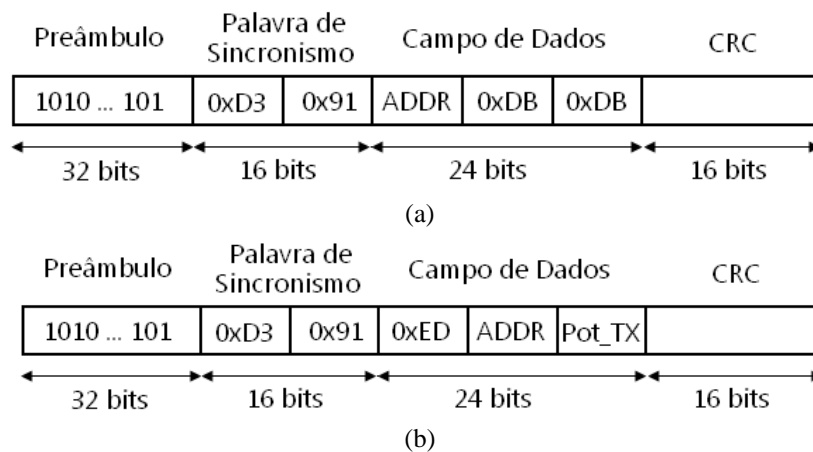


Figura 5.5 – Diagrama de Fluxo Operacional Simplificado: (a) PA, (b) EM.

Sempre que um EM é associado a um PA, um contador é inicializado e uma vez a cada 0,5 s o contador de todos os EMs associados ao PA é decrementado. Esse artifício é utilizado para definir um *timeout* na presença dos EMs. Se o contador atingir um limite mínimo esse EM será considerado fora da área de cobertura do PA. A cada confirmação recebida com sucesso, o contador do EM que enviou a confirmação, é inicializado. A desassociação de um EM ocorre após aproximadamente 10 s de inatividade. O formato dos pacotes enviados tanto pelo PA quanto pelo EM podem ser vistos na Figura 5.6.

**Figura 5.6** – Formato dos pacotes: (a) MSG, (b) ACK.

A transmissão do pacote começa com o envio de 4 *bytes* de preâmbulo seguidos de 2 *bytes* da palavra de sincronismo. Após o envio da palavra de sincronismo são enviados os dados propriamente ditos. O campo de dados tanto do pacote de chamado (MSG) quanto do pacote de confirmação (ACK) possuem 3 *bytes*, esse valor fixo é importante pois permite a utilização do *Data Whitening* e do FEC. O primeiro *byte* do campo de dados da MSG é reservado para o endereço (ADDR) do EM que está sendo chamado. Os outros dois *bytes* são *dummy bytes*, utilizados apenas para completar o campo de dados. No campo de dados do ACK o primeiro *byte* identifica que o pacote é proveniente de um EM, isso evita que, caso um PA esteja na área de cobertura de outro, este não confunda o outro PA como sendo um EM. O segundo *byte* do campo de dados do ACK contém o endereço do EM que está enviando a confirmação e o terceiro *byte* contém o valor do registrador RSSI que mede a potência do sinal recebido pelo EM. Essa potência pode ser utilizada para se ter uma estimativa da distância entre o PA e o EM.

Uma característica desse sensor deve-se ao fato de um EM poder ser associado a dois ou mais PA ao mesmo tempo, ou seja, se existir uma região que esteja sendo coberta por mais de um PA os EMs presentes nesta região serão associados aos vários PA. Essa característica pode não ser desejada por algumas aplicações, como no caso do sistema de localização que será mostrado ainda neste capítulo. Uma solução seria utilizar o valor da potência recebida para identificar qual é o PA mais próximo ao EM, mas devido a problemas de múltiplos percursos que geram interferências sobre o sinal transmitido, não é garantido que apenas observando a potência recebida pode-se identificar qual PA está mais próximo ao EM. Outro artifício que pode ser utilizado para contornar essa característica é apresentado na seção 5.2.

5.1.3. Características do Sensor de Presença

Algumas das características do sensor de presença podem ser vistas na Tabela 5.1 e foram definidas com base na análise realizada no capítulo 3.

Tabela 5.1 – Características do Sensor de Presença.

Características	Sensor de Presença
Modulação	MSK
Taxa de Transmissão	250 kBaud
Potência de Saída	0 dBm
Sensibilidade do Receptor	-93 dBm
Alcance <i>indoor</i> em ambientes com poucos anteparos	9 m
Alcance <i>indoor</i> em ambiente com muitos anteparos	7,5 m
Alcance <i>outdoor</i> sem obstáculos	90 m
Consumo de corrente	21 mA
Duração da bateria	~1 ano
Capacidade de conexão	255 Sensores
Recursos configurados	CRC, Verificação de endereço, Sensor de temperatura

5.2. Sistema de Localização

Uma das motivações para o desenvolvimento de um sistema de localização consiste na aplicação desse sistema em conjunto com o *software* Asterisk. Esse *software* é capaz de

realizar as mesmas funções de um aparelho PBX (*Private Branch Exchange*), podendo operar com diferentes tipos de tecnologias utilizadas em telefonia, ou seja, de aparelhos analógicos a aparelhos que operam com VoIP. As chamadas realizadas pelo Asterisk podem ser feitas tanto por aparelhos físicos quanto por *softphones*. Os aparelhos físicos que utilizam VoIP são dotados de conexão *Ethernet*, já os *softphones* são *softwares* executados em PCs e operam identicamente aos ramais físicos. Assim o Asterisk funciona como um *gateway* de voz interligando todas essas tecnologias de forma transparente ao usuário.

O sistema de localização tem o objetivo de localizar um usuário dentro de um ambiente e disponibilizar essa informação para o servidor Asterisk. Com essa informação, o servidor é capaz de redirecionar as chamadas telefônicas, destinadas ao usuário, para o ponto da rede mais próximo onde ele se encontra. Conforme a configuração do servidor é possível disponibilizar ou bloquear determinados serviços, dependendo da localização, fornecendo mais segurança e comodidade aos usuários.

Esse sistema utiliza um conjunto de sensores de presença, idênticos ao mostrado na seção anterior, de maneira a formar uma rede de sensores que são interligados por uma rede local. Nesta seção é descrito em detalhes o funcionamento e as principais características do sistema de localização desenvolvido.

5.2.1. Arquitetura do Sistema de Localização

Como visto no capítulo 4, existem na literatura várias arquiteturas e técnicas para localização de posição, muitas delas são baseadas no princípio da triangulação. Outras técnicas utilizam a medição do ângulo do sinal de chegada (AoA – *Angle of Arrival*) ou ainda a medição do tempo de propagação do sinal de RF (TDoA – *Time difference of Arrival*). Essas técnicas não serão utilizadas neste trabalho, pois a proposta desta seção é apresentar um sistema de localização que utilize a arquitetura do sensor de presença desenvolvido. Além disso, técnicas como o AoA e o TDoA necessitam de *hardware* específico como antenas direcionais e *hardware* para medição de tempo de alta granularidade (medidas muito pequenas de tempo), sendo também técnicas recomendadas apenas para ambientes externos [33].

O Sistema de Localização aqui apresentado é baseado em um sistema de células, e faz uso dos elementos definidos na seção 5.1. Como dito na seção 4.2, esse sistema é baseado na conectividade dos dispositivos, não fazendo uso de técnicas mais elaboradas, devido ao tipo de aplicação alvo. A localização é simplesmente dada pela posição do terminal fixo (de localização conhecida) ao qual o elemento desconhecido está conectado.

Cada célula é composta por um PA ao qual podem ser associados os diversos EM. Todos os PA são conectados a um *host* de uma rede local e enviam a um servidor, por meio de pacotes UDP, as informações de quais EMs estão conectados ao PA. No lugar da rede local pode-se utilizar também, uma rede local sem fio (WLAN) ou até mesmo uma rede mista, dependendo da necessidade. A Figura 5.7 mostra a arquitetura desse sistema de localização.

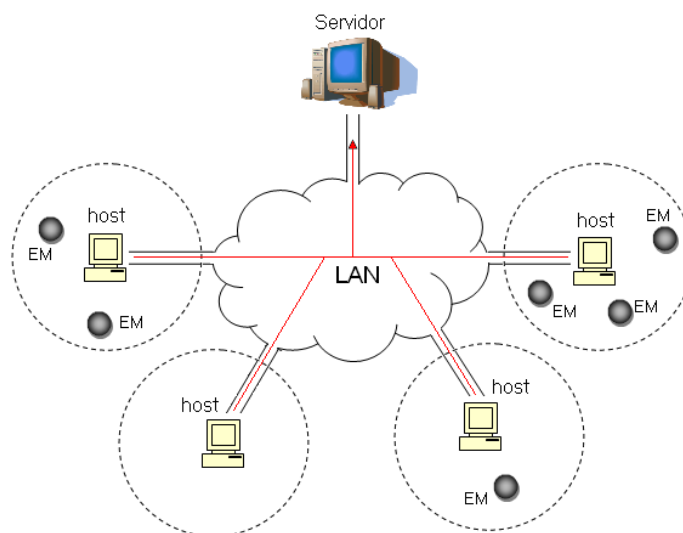


Figura 5.7 – Arquitetura do sistema de localização.

Esse sistema de localização foi concebido para fornecer elementos necessários a um sistema de redirecionamento de chamadas telefônicas realizadas em um servidor Asterisk, no qual cada usuário deve estar de posse de um EM e dependendo de qual PA o usuário estiver mais próximo suas chamadas telefônicas são redirecionadas para aquele *host*. Os usuários podem se deslocar livremente pelas células que compõem a rede de sensores de presença. Caso um usuário de posse de um EM se desloque, por exemplo, de um *host1* para um *host2*, 10 s após sair da área de cobertura do *host1* esse EM será desassociado deste *host* e se já estiver sendo coberto pelo *host2* será associado a ele.

Por não fazer uso de técnicas como a triangulação, a precisão na localização está associada ao alcance da área de cobertura dos PA. Tipicamente em ambientes *indoor* o alcance de um PA (transmitindo em máxima potência) possui um raio de aproximadamente 10 m, em ambiente externos essa distância é ainda maior podendo chegar a 100 m. Evidentemente, assim como em qualquer outro sistema *wireless* que opere na faixa de 2,4 GHz, o alcance está relacionado com fatores ambientais e com o nível de potência do sinal transmitido. Como mostrado no Capítulo 3, existem modelos que podem prever o comportamento do sinal com relação à distância entre transmissor e receptor. Assim, em cada caso onde o sistema de localização for implantado existe a necessidade de fazer um mapeamento do ambiente, para definir a área de cobertura das células. Quanto menor forem as células maior será a precisão na localização de um EM, porém obviamente essa diminuição acarreta em um aumento no número de dispositivos necessários para cobrir a mesma área. A diminuição da área de cada célula é conseguida modificando-se a potência de transmissão dos PAs e dos EMs.

Na Figura 5.8 é mostrado o campo de dados dos pacotes que são enviados ao servidor. Esses pacotes são montados no PA e enviados ao *host* por meio da interface USB, em seguida o *host* coloca os dados em pacotes UDP e os envia ao servidor. O protocolo UDP foi escolhido por não ser necessária uma transmissão orientada a conexão, além do que, a perda de pacotes não é um fator crítico, pois os PAs estão constantemente enviando pacotes ao servidor com informações sobre os EMs associados a ele.

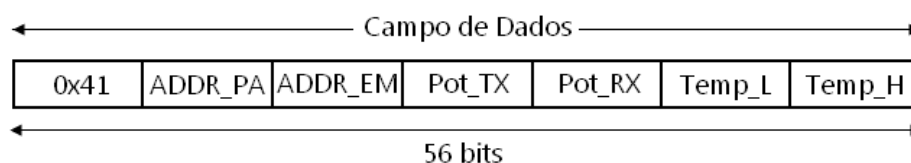


Figura 5.8 – Campo de dados do pacote UDP .

O primeiro *byte* do campo de dados (0x41) é apenas um *byte* utilizado para identificação. O segundo e o terceiro *byte* indicam os endereços do PA e do EM que está enviando o ACK respectivamente. O quarto e o quinto *byte* indicam a potência que a MSG chegou ao EM e a potência que o ACK chegou ao PA, respectivamente. *Temp_L* e *Temp_H* indicam a parte baixa e a parte alta do valor medido pelo conversor A/D do MSP430F2274, respectivamente. Esses valores remetem a tensão do sensor de temperatura

interno ao microcontrolador. É necessário ainda uma conversão desses valores antes que a temperatura seja dada em graus centígrados, como mostrado na seção 2.2.4.

5.2.2. *Softwares de Gerenciamento da Rede*

Tão importantes quanto os dispositivos em si, são os *softwares* que devem ser executados nas máquinas *host* e no servidor. Os *softwares* foram desenvolvidos com a finalidade de gerenciar as informações emitidas pelos sensores. Basicamente as informações enviadas pelos *hosts* ao servidor são os endereços dos APs e dos EMs, o nível de potência dos sinais e a temperatura próxima aos APs. Esses *softwares* foram escritos em linguagem C/C++ e o compilador utilizado foi o *Borland C++ Builder 6™*.

A função do servidor é concentrar as informações repassadas pelos PA espalhados na rede e gerar um arquivo de *log* contendo informações sobre os EMs ativos, ou seja, associados a algum PA. Além disso, a hora em que cada EM é associado a um PA também é registrada no arquivo *log*. Esse artifício permite contornar a questão das múltiplas associações, que surge quando vários PAs cobrem uma mesma área. Como o objetivo do sistema é prover elementos suficientes para o redirecionamento de chamadas, a característica de múltiplas associações deve ser contornada, evitando que a mesma ligação provoque chamadas em vários terminais. Assim é necessário um critério para escolha de qual PA manterá a conexão com o EM. O critério é definido pelo tempo de conexão, ou seja, o EM pertence ao PA ao qual ele está associado a mais tempo. Isso significa que em caso de regiões que são cobertas por mais de um ponto de acesso, um elemento móvel conectado a um desses pontos de acesso, só será desassociado se sair da área de cobertura do seu ponto de acesso original. Entretanto, o EM que estiver desassociado momentaneamente, será associado ao PA que primeiro receber seu pacote de confirmação (ACK).

Cada host da rede deve executar o software PA HOST ((a)
(b)

Figura 5.9a). Na janela desse *software* encontram-se listados os EMs conectados ao PA. Quando um EM é selecionado na lista, as informações da “Potência TX” e “Potência RX” são atualizadas. Os campos “IP destino” e “Porta” contêm o IP (*Internet Protocol*) e a Porta do servidor que concentra as informações de todos os PA conectados na rede. O

botão “Conectar” abre a porta de comunicação serial (USB emulada como RS-232) e inicializa a aquisição e envio das informações captadas pelo PA. O *software* também está configurado para informar a temperatura em graus centígrados próxima ao PA, e é indicada no campo “Temperatura”.

O servidor deve executar o *software* PA SERVIDOR ((a)
(b)

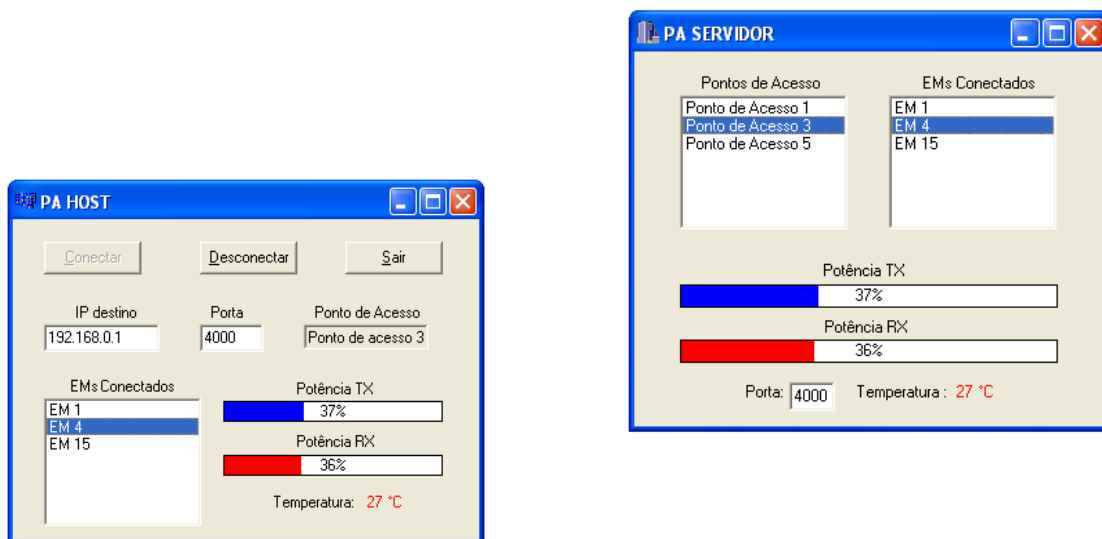


Figura 5.9b). Na janela desse *software* encontram-se listados todos os pontos de acesso ativos na rede, bem como os respectivos EM conectados a eles. Um arquivo *log* é gerado e atualizado sempre que um EM é adicionado, removido ou associado a outro PA.

(a)

(b)

Figura 5.9 – Janela dos *softwares*: (a) PA HOST, (b) PA SERVIDOR

6. Conclusões e Trabalhos Futuros

As redes de sensores sem fio (RSSF) têm sido aplicadas nas mais diversas áreas da atividade humana, com a expectativa de que cada vez mais faça parte das nossas vidas. Ajudar a obter informações, coletando dados de maneira eficiente, ajudando no tratamento de pacientes, na agricultura, automação, entretenimento, segurança, dentre outras, são características das redes de sensores sem fio.

Neste trabalho foi apresentada a elaboração de uma rede de sensores de presença que juntos agregam informações sobre a localização de cada um deles. Essa informação pode ser utilizada por outros sistemas de maneira a tornar possível aplicações como as descritas anteriormente. Sendo assim, uma das motivações para a criação desse sistema, foi a de utilizá-lo para fornecer a informação da localização dos sensores, a fim de transferir chamadas de voz, gerenciadas por um servidor executando o *software* Asterisk, para um ponto mais próximo ao qual o usuário (de posse do sensor) se encontra.

Durante a elaboração desse projeto, várias conclusões puderam ser tiradas. Uma delas diz respeito ao dispositivo utilizado, o eZ430-RF2500, que se mostrou bastante versátil e configurável. Tal característica permite que se possa ter um maior grau de liberdade na criação de sistemas que utilizem a comunicação sem fio.

Durante os ensaios de propagação realizados com o eZ430-RF2500, com o intuito de definir o alcance máximo tanto em ambientes *indoor* como em ambientes *outdoor*, além de testar a aderência de alguns modelos de propagação, notou-se que os modelos de propagação utilizados não foram bem representativos nos dois ambientes testados. Parte disso pode ser justificado pelas condições nas quais os testes foram realizados e em parte

pelos próprios modelos utilizados. Esses modelos possuem boa aderência às redes *WI-FI* 802.11. Já o alcance máximo foi satisfatório e atendeu as expectativas.

O recurso de equalização utilizado durante o teste, o *data whitening*, não mostrou a eficiência esperada, por tanto não foi habilitado na versão final do *software*, visto que ativá-lo representa um consumo maior de energia. Outra conclusão foi que a utilização da modulação MSK se mostrou mais adequada, dado que na maioria dos casos foi possível atingir a distância de 9 m (*indoor*) sem a perda do *link*.

Com relação ao sistema de localização, sua integração com redes locais concebe versatilidade e facilita o aumento da cobertura numa determinada área. Durante a elaboração do protocolo de comunicação, vários problemas surgiram e medidas foram tomadas no sentido de solucionar esses problemas, por exemplo, a comunicação realizada em *time slots*. Outro exemplo foi a utilização de um *byte* para identificar uma mensagem de um sensor, isso corrigiu o problema de associação de PAs como sensores. As regras de associação e desassociação definidas no *software* de gerenciamento da rede, também é exemplo de medidas tomadas durante a elaboração do sistema.

Como sugestão para trabalhos futuros pode citar:

- Utilização outros modelos de propagação, buscando descrever melhor o comportamento do dispositivo. Dentre os modelos que devem ser testados, pode-se utilizar o modelo de Souza e Lins o qual considera a umidade relativa do ar como parâmetro.
- Propor um modelo de propagação específico para o dispositivo caso os outros modelos não sejam representativos.
- Realizar ensaios com outros tipos de modulação, tanto para ambientes internos como para ambientes externos. Nesses ensaios, habilitar o recurso de correção de erro FEC (*Forward Error Correction*) e avaliar o desempenho da recepção do sinal. Também variar a velocidade de comunicação e observar o comportamento do *link*.
- Melhorar o protocolo de comunicação tentando evitar os problemas de *overhearing* e da escuta prolongada do canal.
- Integrar o sistema de localização desenvolvido com o sistema de roteamento de chamadas VoIP, com servido Asterisk.

- Buscar melhorar o protocolo de comunicação entre o ponto de acesso (PA) e o elemento móvel (EM), a fim de reduzir o consumo de energia aumentando assim a vida útil das baterias.

Referências

- [1] Texas Instruments. eZ430-RF2500 Development Tool, User's Guide – slau227e, Abr. 2009. Disponível em: <<http://focus.ti.com/lit/ug/slau227e/slau227e.pdf>>. Acesso em: 24 Fev 2010.
- [2] <http://www.asterisk.org/>
- [3] Texas Instruments. MSP430x22x4 Datasheet, jul. 2007. Disponível em: <<http://focus.ti.com/lit/ds/symlink/msp430f2274.pdf>> Acesso em: 24 Fev 2010.
- [4] Texas Instruments; MSP430x2xx Family, User's Guide – slau144e, 2008; Disponível em: <<http://focus.ti.com/lit/ug/slau144e/slau144e.pdf>>. Acesso em: 24 Fev 2010.
- [5] Texas Instruments; CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver; Datasheet, 2007. Disponível em: <<http://focus.ti.com/lit/ds/symlink/cc2500.pdf>> .Acesso em: 24 Fev 2010.
- [6] PEREIRA, Fábio; Microcontroladores MSP430 - Teoria e Prática, São Paulo, Érica, 2005.
- [7] Texas Instruments; Programming a Flash-Based MSP430 Using the JTAG Interface, Application Report – slaa149g, dez. 2005, Rev.2009. Disponível em: < <http://focus.ti.com/lit/an/slaa149g/slaa149g.pdf> >. Acesso em: 24 Fev 2010.
- [8] S. B. Wicken; *Error control Systems For Digital Communication and Storage*; New Jersey: Prentice Hall, 1995.
- [9] <http://www.ti.com>

- [10] BENSKEY, Alan; *Short Range Wireless Communication* 2nd Edition: Elsevier, 2004.
- [11] RAPPAPORT, T. S.; *Wireless Communications Principles and Practice*. 2nd Edition; New Jersey: Prentice Hall, 2002.
- [12] SEYBOLD, J. S.; *Introduction to RF Propagation*. 1st Edition; New Jersey: John Wiley & Sons, 2005.
- [13] BARTON D. K.; LEONOV S. A.; *Radar Technology Encyclopedia*; Boston London, Artech House, 1998.
- [14] ITU-R Recommendations. *Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900MHz to 100GHz*, ITU-R P.1238-5, Suíça, fev. 2007.
- [15] SOUZA, S. R; Um Novo Modelo para Propagação em Redes Wi-Fi; Recife, 2008; Dissertação de Mestrado, Universidade Federal de Pernambuco.
- [16] PHALAVAN K.; LEVESQUE H. A.; *Wireless Information Network*. 2nd Edition, New Jersey: Wiley, 2005.
- [17] Texas Instruments; *Range Measurements in an Open Field Environment*, Design Note – swraa169a, 2008. Disponível em: <<http://focus.ti.com/lit/an/swra169a/swra169a.pdf>>. Acesso em: 24 Fev 2010.
- [18] Cadernos do IME, 2003, disponível online: <http://www.ime.uerj.br/cadernos/cadinf/vol14/>; Acesso em: 24 Fevereiro 2010.
- [19] ZHAO, Feng; GUIBAS, Leonidas; *Wireless Sensors Networks: An Information Processing Approach*; San Francisco: Elsevier, 2004.
- [20] BARLEZE, Alessandro; Fusão de Dados em Esquemas Híbridos Envolvendo AGPS para Localização de Posicionamento; Curitiba, 2003; Dissertação de Mestrado – Centro de Ciências Exatas e Tecnologia, Universidade Católica do Paraná.
- [21] TANENBAUM, A.; “Redes de Computadores”, 8ª Tiragem da Tradução da Quarta Edição, Editora Campus, 2003.
- [22] SILVA, Paula Patrícia; Utilizando a Tecnologia SIG para Adicionar Valor aos

- Serviços Baseados em Localização para Usuários de Dispositivos Móveis Celulares; Rio de Janeiro, 2004; Dissertação de Mestrado – Universidade Federal do Rio de Janeiro.
- [23] HARDER, Adam; SONG, Lanlan; WANG, Yu; *Towards an Indoor Location System Using RF Signal Strength in IEEE 802.11 Networks*; International Conference on Information Technology: Coding and Computing (ITCC), 2005.
- [24] CHEN, Yi-Chao; CHIANG, Ji-Rung; CHU, Hao-hua; HUANG, Polly; TSUI, Arvin Wen; International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems; "*Sensor-Assisted Wi-Fi Indoor Location System for Adapting to Environmental Dynamics*", Canada, 2005.
- [25] MOREIRA, Rafael Barbosa; Análise de Técnicas de Localização em Redes de Sensores Sem Fio; Campinas – SP, 2007; Dissertação de Mestrado, Universidade Estadual de Campinas.
- [26] BULUSU, Nirupama; HEIDEMANN John; ESTRIN, Deborah; *GPS-less low-cost outdoor localization for very small devices*, IEEE Personal Communications, 7(5):28-34, Outubro 2000.
- [27] HE, Tian; HUANG, Chengdu; BLUM, Brian M.; STANKOVIC, John A.; ABDELZAHER, Tarek; *Range-Free Localization Schemes for Large Scale Sensor Networks*; Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, 81-95, 2003.
- [28] DSRC IEEE Working Group; Disponível em:
< <http://www.learmstrong.com/DSRC/DSRCHomeset.htm> >; Acesso em: 24 fevereiro 2010.
- [29] SZEWCZYK, Robert; OSTERWEIL, Eric; POLASTRE, Joseph; HAMILTON, Michael; MAINWARING, Alan; ESTRIN, Deborah. *Application driven systems research: Habitat monitoring with sensor networks*. Communications of the ACM, 2004.
- [30] DARPA Information Processing Technology Office; Disponível em: www.darpa.mil/ipto/programs/sensit/index.htm; Acesso em: 24 de fevereiro

2010.

- [31] ZORKOT, Alan C.; RUIZ, Linnyer B.; ASSUNÇÃO, Helen P.; Uma Ferramenta para Detecção de Movimento Utilizando Redes de Sensores Sem Fio; XXVI Congresso da SBC, julho 2006.
- [32] MALAN, D. David; JONES, Thaddeus Fulford et al.; (2004), *Codeblue: An ad hoc sensor network infrastructure for emergency medical care*. In Proceeding of the International Workshop on Wearable and Implantable Body Sensor Networks.
- [33] NUNES, Bruno A. Arouche; Um Sistema de Localização para Redes Wi-Fi Baseado em Níveis de Sinal e Modelo Referenciado de Propagação; Rio de Janeiro – RJ, 2006; Universidade Federal do Rio de Janeiro.
- [34] SCHWARTZ, Mischa.; *Information transmission, modulation, and noise*, 2. ed.; Tokyo, McGraw-Hill, 1970.
- [35] XIONG, Fuqin; *Digital Modulation Techniques*; Boston London, Artech House, 2000.

Apêndice A. Código Fonte dos Programas do dispositivo eZ430-RF2500

A.1 - Software dos Elementos Móveis (EM's)

```
//-----  
// DISPOSITIVO: EZ430-RF2500  
//  
// AUTOR: EURICO MOURA  
// DATA: 27/02/2009  
// MODIFICADO:  
//  
// COMENTÁRIO: Este programa espera a recepção de um chamado endereçado a este  
//             dispositivo e envia uma confirmação (ACK) contendo 3 bytes. O pri-  
//             -meiro byte corresponde ao identificador de sensor (0xED) o se-  
//             -gundo corresponde ao endereço do sensor e o terceiro a potência do  
//             sinal recebido.  
//-----  
// ARQUIVOS INCLUSOS  
  
#include "stdbool.h"  
#include "msp430x22x4.h"  
#include "includes/CC2500_DEF.h"  
#include "includes/CC2500_MSK_250k.h"  
//-----  
// LISTA DE VARIÁVEIS GLOBAIS  
  
bool TX_OK = false;  
unsigned long TEMP = 0;  
const unsigned char ED_ADDR=0x30;    // max 0x63  
//-----  
// CARREGA PILHA  
//-----  
// preamble / SyncWord / Lengh / ADDR / DataField / CRC-16 //  
//-----  
void CARREGA_TX_FIFO(char potencia)  
{  
    unsigned char dados[3];  
    unsigned char i;
```

```

dados[0]=0xED;           // primeiro byte do pacote (identificador de sensor)
dados[1]=ED_ADDR;        // segundo byte (endereço do sensor)
dados[2]=potencia;        // terceiro byte (potência do sinal recebido)

IE2 &= ~0x0F;            // desabilita interrupções do modo SPI
P3OUT |= 0x01;           // desliga e liga dispositivo escravo
P3OUT &= ~0x01;          // para limpar qualquer acesso ao chip
IFG2 &= ~UCB0RXIFG;      // zera flag de recepção

UCB0TXBUF = BURST | 0x3F; // transmite o comando
while(!(IFG2 & UCB0RXIFG)); // espera recepção do byte STATUS

for(i=0;i<3;i++)         // escrita na pilha em modo rajada
{
    IFG2 &= ~UCB0RXIFG;   // zera flag de recepção
    UCB0TXBUF = dados[i]; // transmite byte de teste
    while(!(IFG2 & UCB0RXIFG)); // espera recepção do byte STATUS
}
P3OUT |= 0x01;           // desliga chip escravo (finaliza transmissão)
}
//-----
// PISCA LED

void PISCA_LED1(void)
{
    P1OUT = 1;             // acende LED vermelho
    __delay_cycles(20000); // deley ~0.1s
    P1OUT = 0;             // acende LED verde
}
void PISCA_LED2(void)
{
    P1OUT = 2;             // acende LED verde
    __delay_cycles(20000); // deley ~0.1s
    P1OUT = 0;             // acende LED verde
}

//-----
// CC2500 CONFIG TX

void CC2500_TX_CONFIG(void)
{
    UART_SPI_RW(IOCFG0,0x06); // verificar pg.46 datasheet
    TX_OK = false;
    TEMP = 0;
}

//-----

```

```
// CC2500 CONFIG RX
```

```
void CC2500_RX_CONFIG(void)
```

```
{
    UART_SPI_RW(IOCFG0,0x41);           // verificar pg.46 datasheet
    TX_OK = false;
    TEMP = 0;
}
```

```
//-----
```

```
// FLUSH TX RX
```

```
void FLUSH(void)
```

```
{
    STROBE(SIDLE);
    __delay_cycles(900);
    STROBE(SFTX);
    __delay_cycles(300);
    STROBE(SFRX);
    __delay_cycles(300);
}
```

```
//-----
```

```
// CONFIGURA INTERRUPTÃO NO PINO GDO0 (P2.6)
```

```
void GDO0_INT_CONFIG(void)
```

```
{
    P2DIR &= ~0x40;           // GDO0 como entrada
    P2SEL &= ~0x40;           // se PxSEL=1 as interrupções do PORTx
    ficam desabilitadas
    P2IE |= 0x40;             // P2.6 habilita interrupção
    P2IES |= 0x40;            // P2.6 (borda de descida)
    P2IFG &= ~0x40;           // P2.6 IFG (limpa flag de interrupção)
    __enable_interrupt();      // habilita interrupção
}
```

```
//-----
```

```
// FUNÇÕES DE TRATAMENTO DE INTERRUPTÕES NO PINO GDO0
```

```
// TRATAMENTO DAS INTERRUPTÕES NO MODO TX e RX
```

```
#pragma vector = PORT2_VECTOR           // proxima função vai tratar a interrupção
```

```
__interrupt void TX_RX_INT(void)        // informa função de tratamento da
interrupção
```

```
{
    P2IFG &= ~0x40;           // P2.6 IFG (limpa flag de interrupção)
    TX_OK = 1;
    TEMP = 100001;
}
```

```
//-----
```

```

// ROTINA PRINCIPAL
void main(void)
{
    bool RX_OK = false;
    // configurações -----
    WDTCTL = WDTPW + WDTHOLD;    // desabilita watchdog para prevenir time out
    BCSCTL1 = CALBC1_1MHZ;        // frequência de operação do microcontrolador
    DCOCTL = CALDCO_1MHZ;
    __delay_cycles(10000);        // aguarda estabilização do sistema
    UART_SPI_CONFIG();            // função de inicialização da interface SPI
    STROBE(SRES);                 // reseta chip de RF
    CC2500_CONFIG();              // configura modo de transmissão e registradores
    GDO0_INT_CONFIG();            // configura interrupções
    UART_SPI_RW(0x3E,0xFF);       // configura potência de saída.
    UART_SPI_RW(ADDR,ED_ADDR);    // carrega endereço individual do sensor
    P1DIR = 3;                    // p1.0 e p1.1 como saídas
    //-----
    while(1)
    {
        // Recepção -----
        PISCA_LED1();             // pisca led vermelho
        CC2500_RX_CONFIG();       // configura chip para modo RX
        STROBE(SRX);              // entra no modo RX
        while(RX_OK==false)       // aguarda recepção, timeout = ~100ms
        {
            while(TEMP<100000)
                TEMP++;
            RX_OK=true;
        }
        RX_OK = false;            // limpa flag de recepção
        // verifica se o número de bytes recebidos é igual ao transmitido
        if (UART_SPI_RW(0xFB,DUMMY) == 0x03)
        {
            // verifica CRC (código de redundância ciclica)
            if ((UART_SPI_RW(0xF8,DUMMY)&0x80)==0x80)
            {
                // Transmissão -----

                FLUSH();
                // carrega pacote para transmissão
                CARREGA_TX_FIFO(UART_SPI_RW(0xF4,DUMMY));
                CC2500_TX_CONFIG();    // configura chip para modo TX
                STROBE(STX);           // comando para transmissão dos dados
                while(TX_OK==false);   // aguarda transmissão completa.
                PISCA_LED2();          // pisca led verde
            }
        }
    }
}

```



```

        // esvazia pilha de recepção, restaura pino GDO0 e previne travamentos.
        FLUSH();
    }
}
//-----

```

A.2 - Software dos Pontos de Acesso (PA)

```

//-----
// ARQUIVOS INCLUSOS

#include "stdbool.h"
#include "msp430x22x4.h"
#include "includes/CC2500_DEF.h"
#include "includes/CC2500_MSK_250k.h"

// LISTA DE VARIÁVEIS GLOBAIS -----

bool TX_OK = false;
unsigned int TEMP;
unsigned int ADCTEMP;
unsigned char RX_FIFO_BUFF[3];
unsigned char TX_UART_BUFF[7];

const unsigned char AP_ADDR=0x02;      // modificar para cada terminal de acesso

//-----
// PROTÓTIPOS DE FUNÇÕES

void CARREGA_TX_FIFO(char addr);
void PISCA_LED1(void);
void PISCA_LED2(void);
void CC2500_TX_CONFIG(void);
void CC2500_RX_CONFIG(void);
void GDO0_INT_CONFIG(void);
void UART_9600_CONFIG(void);
//-----
// CARREGA PILHA
//-----
// preamble / SyncWord / Lengh / ADDR / DataField / CRC-16 //
//-----
void CARREGA_TX_FIFO(char addr)
{
    unsigned char dados[3];
    unsigned char i;

```

```

dados[0]=addr;           // primeiro byte do pacote (ADDR)
dados[1]=0xDB;           // segundo byte (dummy)
dados[2]=0xDB;           // terceiro byte (dummy)

IE2 &= ~0x0F;            // desabilita interrupções do modo SPI
P3OUT |= 0x01;           // desliga e liga dispositivo escravo
P3OUT &= ~0x01;          // para limpar qualquer acesso ao chip
IFG2 &= ~UCB0RXIFG;      // zera flag de recepção

UCB0TXBUF = BURST | 0x3F; // transmite o comando
while(!(IFG2 & UCB0RXIFG)); // espera recepção do byte STATUS
for(i=0;i<3;i++)         // escrita na pilha em modo rajada
{
    IFG2 &= ~UCB0RXIFG;   // zera flag de recepção
    UCB0TXBUF = dados[i]; // transmite byte de teste
    while(!(IFG2 & UCB0RXIFG)); // espera recepção do byte STATUS
}
P3OUT |= 0x01;           // desliga chip escravo (finaliza transmissão)
}
//-----
// LER PILHA

void LER_RX_FIFO(void)
{
    unsigned char i;
    IE2 &= ~0x0F;         // desabilita interrupções do modo SPI
    P3OUT |= 0x01;        // desliga e liga dispositivo escravo
    P3OUT &= ~0x01;       // para limpar qualquer acesso ao chip
    IFG2 &= ~UCB0RXIFG;   // zera flag de recepção
    UCB0TXBUF = READ | BURST | 0x3F; // transmite o comando
    while(!(IFG2 & UCB0RXIFG)); // espera recepção do byte STATUS

    for(i=0;i<3;i++)      // escrita na pilha em modo rajada
    {
        IFG2 &= ~UCB0RXIFG; // zera flag de recepção
        UCB0TXBUF = DUMMY;  // transmite byte de teste
        while(!(IFG2 & UCB0RXIFG)); // espera recepção de um byte da pilha
        RX_FIFO_BUFF[i] = UCB0RXBUF; // copia dados para um buffer
    }
    P3OUT |= 0x01;        // desliga chip escravo (finaliza transmissão)
}
//-----
// PISCA LED
void PISCA_LED1(void)
{
    P1OUT = 1;            // acende LED vermelho

```



```

P3SEL = 0x30;           // P3.4,5 = USCI_A0 TXD/RXD
UCA0CTL1 |= UCSSEL_2;   // SMCLK
UCA0BR0 = 104;          // 1MHz 9600 - parte baixa do registrador
UCA0BR1 = 0;            // 1MHz 9600 - parte alta do registrador
UCA0MCTL = UCBRS0;      // Modulation UCBRSx = 1
UCA0CTL1 &= ~UCSWRST;   // **Initialize USCI state machine**
/*
  OBS: Uma interrupção é gerada cada vez que o buffer (UCA0TXBUF) está pronto
  para receber um novo dado. A flag UCA0IFG é limpa a cada novo byte escrito
  no buffer. No nosso caso a interrupção não é necessária, vamos apenas observar
  a flag UCA0IFG que é ativada e desativada independentemente da interrupção.
*/
}
//-----
// CONFIGURA INTERRUPTÃO NO PINO GDO0 (P2.6)

void GDO0_INT_CONFIG(void)
{
    P2DIR &= ~0x40;      // GDO0 como entrada
    P2SEL &= ~0x40;      // se PxSEL=1 as interrupções do PORTx
    // ficam desabilitadas
    P2IE |= 0x40;        // P2.6 interrupt enabled
    P2IES |= 0x40;       // P2.6 Hi/lo edge (borda de descida)
    P2IFG &= ~0x40;      // P2.6 IFG cleared (limpa flag de interrupção)
    __enable_interrupt(); // habilita interrupção
}
//-----
// CONFIGURA CONVERSOR ADC10 PARA SENSOR DE TEMPRATURA

void ADC10_TEMP_CONFIG(void)
{
    ADC10CTL1 = INCH_10 + ADC10DIV_4;
    ADC10CTL0 = SREF_1 + ADC10SHT_3 + REFON + ADC10ON +
    ADC10IE+ADC10SR;
    __delay_cycles(50);   // Tempo de estabilização do Vref.
}

//-----
// FUNÇÕES DE TRATAMENTO DE INTERRUPTÕES NO PINO GDO0
// TRATAMENTO DAS INTERRUPTÕES PINO GDO0 (p2.6) MODO TX e RX

#pragma vector = PORT2_VECTOR // proxima função vai tratar a interrupção
__interrupt void TX_RX_INT(void) // informa função de tratamento da interrupção
{
    P2IFG &= ~0x40;          // P2.6 IFG cleared (limpa flag de interrupção)
    TX_OK = true;

```

```

    TEMP=1001;
}
//-----
// TRATAMENTO DAS INTERRUPTÕES NO CONVERSOR ADC10
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    __bic_SR_register_on_exit(CPUOFF);    // Clear CPUOFF bit from 0(SR)
}
//-----
// ROTINA PRINCIPAL

void main(void)
{
    bool RX_OK = false;
    unsigned char addr;
    unsigned char i;

    // configurações -----
    WDTCTL = WDTPW + WDTHOLD;            // desabilita watchdog para prevenir time out

    BCCTL1 = CALBC1_1MHZ;                // frequência de operação do microcontrolador
    DCOCTL = CALDCO_1MHZ;
    __delay_cycles(10000);                // aguarda estabilização do sistema

    UART_SPI_CONFIG();                   // função de inicialização da interface SPI do
    MSP430F2274

    STROBE(SRES);                        // reset do chip cc2500
    CC2500_CONFIG();                     // configura registradores
    GDO0_INT_CONFIG();                   // configura interrupções no pino GDO0
    ADC10_TEMP_CONFIG();                 // configura conversor A/D (Temperatura)
    UART_SPI_RW(0x3E,0xFF);              // configura potência
    UART_SPI_RW(ADDR,0xED);              // configura endereço de verificação
    P1DIR = 3;                           // p1.0 e p1.1 como saídas
    addr = 1;                            // inicializa endereços de busca

    //-----

    while(1)
    {
        // Transmissão -----
        CARREGA_TX_FIFO(addr);           // carrega pacote para transmissão
        CC2500_TX_CONFIG();               // configura chip para transmissão
        STROBE(STX);                      // comando para transmissão dos dados
        while(TX_OK==false);              // aguarda transmissão completa.
        // esvazia pilhas, restaura pino GDO0 e previne travamentos.
        FLUSH();
    }
}

```

```

    addr++;
    addr = addr% 100;                // limita o número de endereços de busca
// Recepção -----
    CC2500_RX_CONFIG();              // configura chip para modo RX
    STROBE(SRX);
    while(RX_OK==false)              // aguarda recepção, timeout = ~1ms.
    {
        while(TEMP<1000)
            TEMP++;
        RX_OK = true;
    }
    RX_OK = false;                    // limpa flag de recepção

// verifica se o número de bytes recebidos é igual ao transmitido
if (UART_SPI_RW(0xFB,DUMMY) == 0x03)
{
    // verifica CRC (código de redundância ciclica)
    if ((UART_SPI_RW(0xF8,DUMMY)&0x80)==0x80)
    {
        LER_RX_FIFO();               // Ler pilha de recepção
        ADC10CTL0 |= ENC + ADC10SC;   // Inicia amostragem e conversão
        __bis_SR_register(CPUOFF + GIE); // LPM0 com interrupções habilitadas

        ADCTEMP = ADC10MEM;

// Transmissão Serial UART -----

        // Montagem do pacote enviado pela serial
        TX_UART_BUFF[0] = 0x41;
        TX_UART_BUFF[1] = AP_ADDR;
        TX_UART_BUFF[2] = RX_FIFO_BUFF[1];
        TX_UART_BUFF[3] = RX_FIFO_BUFF[2];
        TX_UART_BUFF[4] = UART_SPI_RW(0xF4,DUMMY);
        // armazena a parte baixa da variável de 16 bits
        TX_UART_BUFF[5] = (unsigned char) ADCTEMP;
        // deslocamento para direita, pega parte alta do registrador
        ADCTEMP >>= 8;

        // armazena parte alta do registrador
        TX_UART_BUFF[6] = (unsigned char) ADCTEMP;
        // configura comunicação serial a 9600 baud
        UART_9600_CONFIG();
        // transmite pacote pela serial
        for(i=0;i<7;i++)
        {
            while (!(IFG2&UCA0TXIFG));
            UCA0TXBUF = TX_UART_BUFF[i];

```

```

    }
    // restabelece modo de comunicação serial SPI
    UART_SPI_CONFIG();
    PISCA_LED2();
    // esvazia pilhas, restaura pino GDO0 e previne travamentos.
    FLUSH();
  }
}
// esvazia pilhas, restaura pino GDO0 e previne travamentos.
FLUSH();
}
}
//-----

```

A.3 - Configuração dos Registradores do CC2500 para Modulação FSK 250kBaud (CC2500_FSK_250K.h)

Os valores dos registradores do chip de RF CC2500 foram obtidos por meio do *software* SmartRF® Studio da *Texas Instruments*.

```

// CARACTERISTICA DO LINK
//
//   MODULAÇÃO  => 2-FSK
//   TAXA        => 250kBound
//   DESVIO      => 38kHz
//   FILTRO      => 540kHz
//
//-----
// INCLUDES

```

```

#include "msp430x22x4.h"
#include "CC2500_DEF.h"
#include "UART_SPI_FUNC.h"

```

```

//-----
// FUNÇÃO DE CONFIGURAÇÃO

```

```

void CC2500_CONFIG(void)
{
    __delay_cycles(1000);           // espera estabilização do sistema.
}

```

```

//-----
// REGISTRADORE DESTINADOS A FORMATAÇÃO DO PACOTE DE DADOS
//-----
UART_SPI_RW(IOCFIG2,0x0B);    //
UART_SPI_RW(IOCFIG0,0x06);    // configurado dentro do programa principal
UART_SPI_RW(FIFOTHR,0x07);    // 33 - 32 (tx - rx)
UART_SPI_RW(PKTLEN,0x03);     // pacote com 3 bytes
UART_SPI_RW(PKTCTRL0,0x04);    // CRC chek, CC2400_EN = 0, whitening (0x44)
UART_SPI_RW(PKTCTRL1,0x05);    // verificação de endereço, append_status
UART_SPI_RW(CHANNR,0x00);      //
//UART_SPI_RW(ADDR,0x02);      // deve ser modificado para cada dispositivo ED.

//-----
// REGISTRADORES DESTINADOS A CONFIGURAÇÃO DO LINK
//-----

UART_SPI_RW(FSCTRL1,0x0A);
UART_SPI_RW(FSCTRL0,0x00);
UART_SPI_RW(FREQ2,0x5D);
UART_SPI_RW(FREQ1,0x93);
UART_SPI_RW(FREQ0,0xB1);
UART_SPI_RW(MDMCFG4,0x2D);
UART_SPI_RW(MDMCFG3,0x3B);
UART_SPI_RW(MDMCFG2,0x03);
UART_SPI_RW(MDMCFG1,0x22);
UART_SPI_RW(MDMCFG0,0xF8);
UART_SPI_RW(DEVIATN,0x44);
UART_SPI_RW(MCSM2,0x07);
UART_SPI_RW(MCSM1,0x30);
UART_SPI_RW(MCSM0,0x18);
UART_SPI_RW(FOCCFG,0x1D);
UART_SPI_RW(BSCFG,0x1C);
UART_SPI_RW(AGCTRL2,0xC7);
UART_SPI_RW(AGCTRL1,0x00);
UART_SPI_RW(AGCTRL0,0xB0);
UART_SPI_RW(FREND1,0xB6);
UART_SPI_RW(FREND0,0x10);
UART_SPI_RW(FSCAL3,0xEA);
UART_SPI_RW(FSCAL2,0x0A);
UART_SPI_RW(FSCAL1,0x00);
UART_SPI_RW(FSCAL0,0x11);
UART_SPI_RW(FSTEST,0x59);
UART_SPI_RW(TEST2,0x88);
UART_SPI_RW(TEST1,0x31);
UART_SPI_RW(TEST0,0x0B);
}

```



```
//-----
```

A.4 - Arquivo com Funções de Leitura e Escrita Para Interface SPI. (UART_SPI_FUNC.h)

```
//-----
```

```
#include "msp430x22x4.h"
```

```
#include "CC2500_DEF.h"
```

```
//-----
```

```
// FUNÇÃO DE INICIALIZAÇÃO DA INTERFACE SPI.-----
```

```
void UART_SPI_CONFIG()
```

```
{
```

```
    P3DIR = 0x0B;           // configura direção dos pinos I/O
```

```
    P3OUT |= 0x01;         // desabilita dispositivo escravo
```

```
    UCB0CTL1 = UCSWRST;     // ativa RESET para configuração
```

```
                        // fonte de clock SMCLK
```

```
    UCB0CTL1 = UCSWRST | UCSSEL0 | UCSSEL1;
```

```
                        // borda de captura, MSB, master, sincrono
```

```
    UCB0CTL0 = UCCKPH | UCMSB | UCMST | UCSYNC;
```

```
    UCB0BR0 = 2;           // fator de divisão do clock
```

```
    UCB0BR1 = 0;           //
```

```
    P3SEL |= 0x0E;         // configuração de porta, seleção de função
```

```
    UCB0CTL1 &= ~UCSWRST;  // desabilita RESET, fim da configuração
```

```
}
```

```
//-----
```

```
// FUNÇÃO DE RESET DO CHIP DE RF
```

```
//USADO APENAS NA INICIALIZAÇÃO. APÓS A INICIALIZAÇÃO BASTA USAR
```

```
//O COMANDO SRES.
```

```
void RESET_CC2500()
```

```
{
```

```
    P3OUT |= 0x01;         // CSN HIGH
```

```
    __delay_cycles(30);    // espera ~30us (assumindo CLK de 1MHz - 1 ciclos por us)
```

```
    P3OUT &= ~0x01;        // CSN LOW
```

```
    __delay_cycles(30);    // espera ~30us
```

```
    P3OUT |= 0x01;        // CSN HIGH
```

```
    __delay_cycles(45);    // espera ~40us (pelo menos)
```

```
    P3OUT &= ~0x01;        // CSN LOW
```

```
    while(P3IN & 0x04);    // espera SO ir para baixo
```

```
    IFG2 &= ~UCB0RXIFG;    // escreve um comando e espera confirmação.
```

```
    UCB0TXBUF = SRES;      // comando strobe SRES - reseta chip e põe no estado
```

```
IDLE
```

```

while(!(IFG2 & UCB0RXIFG)); // espera recepção do byte STATUS

while(P3IN & 0x04);          // espera SO ir para baixo

P3OUT |= 0x01;               // CSN HIGH fim do RESET
}

//-----
// FUNÇÃO QUE ESCRIVE UM COMANDO SEGUIDO DE UM DADO DE 8 BITS
//NA SPI.

unsigned char UART_SPI_RW (unsigned char comando, unsigned char dado)
{
    unsigned char temp;
    IE2 &= ~0x0F;             // desabilita interrupções do modo SPI
    P3OUT |= 0x01;            // desliga e liga dispositivo escravo
    P3OUT &= ~0x01;           // para limpar qualquer acesso ao chip
    IFG2 &= ~UCB0RXIFG;       // zera flag de recepção
    UCB0TXBUF = comando;      // transmite o comando
    while(!(IFG2 & UCB0RXIFG)); // espera recepção do byte STATUS
    IFG2 &= ~UCB0RXIFG;       // zera flag de recepção
    UCB0TXBUF = dado;         // transmite o dado, se a operação for uma leitura
    "dado" pode assumir
    while(!(IFG2 & UCB0RXIFG)); // qualquer valor.
    temp = UCB0RXBUF;         // recebe um byte de dado ou status
    P3OUT |= 0x01;            // desliga chip escravo
    return (temp);            // retorna valor recebido
}

//-----
// FUNÇÃO QUE ESCRIVE UM COMANDO STROBE

unsigned char STROBE(unsigned char comando)
{
    unsigned char temp;
    IE2 &= ~0x0F;             // desabilita interrupções do modo SPI
    P3OUT |= 0x01;            // desliga e liga dispositivo escravo
    P3OUT &= ~0x01;           // para limpar qualquer acesso ao chip
    IFG2 &= ~UCB0RXIFG;       // zera flag de recepção
    UCB0TXBUF = comando;      // transmite o comando
    while(!(IFG2 & UCB0RXIFG)); // espera recepção do byte STATUS
    temp = UCB0RXBUF;         // recebe byte de dado ou status
    P3OUT |= 0x01;            // desliga chip escravo
    return (temp);            // retorna valor recebido
}

//-----

```

O arquivo “CC2500_DEF.h” simplesmente associa os nomes dos registradores aos seus endereços físicos correspondentes no CC2500. Com exceção das definições mostradas abaixo, todos os nomes dos registradores, bem como seus endereços, podem ser encontrados no *datasheet* do chip de RF [5].

```
#define BURST          0x40 // Comando para leitura ou escrita em surto
#define WRITE          0x00 // Comando para escrita
#define READ           0x80 // Comando para leitura
#define DUMMY          0xDB // Byte de um dado qualquer
```

Apêndice B. Técnicas de Modulação

Este apêndice trata, resumidamente, das principais técnicas de modulação disponíveis no chip de RF CC2500. As técnicas de modulação digital disponíveis nesse dispositivo são: OOK (*On Off Keying*), FSK (*Frequency Shift Keying*), GFSK (*Gaussian Frequency Shift Keying*) e MSK (*Minimum Shift Keying*)

B.1 - Modulação OOK

A modulação OOK (*On-Off Keying*) é um tipo de modulação em amplitude onde é feito o chaveamento entre zero (estado *OFF*) e algum nível de amplitude do sinal (estado *ON*). Essa modulação é um caso particular da modulação ASK (*Amplitude Shift Keying*). A modulação ASK permite a utilização de vários níveis discretos de amplitude do sinal modulado, quando isso ocorre, ela também é chamada de MASK. Quando vários níveis de amplitudes são utilizados os dados que se deseja transmitir são agrupados em blocos de *bits* de comprimento n , em que $n = \log_2^M$. Assim o parâmetro M representa a quantidade de níveis de amplitudes possíveis para o sinal modulado, e cada bloco de n *bits* está associado a um nível de amplitude A_i . O caso binário da modulação ASK, chamado de BASK (*Binary Amplitude Shift Keying*) é observado quando os símbolos zeros e uns são representados por pulsos de uma portadora sinusoidal, com frequência f_c , mas de amplitudes diferentes A_0 e A_1 . Quando uma dessas amplitudes é igual a zero o resultado é a modulação OOK. A Figura B.1 mostra um sinal binário e sua forma modulada com a modulação OOK.

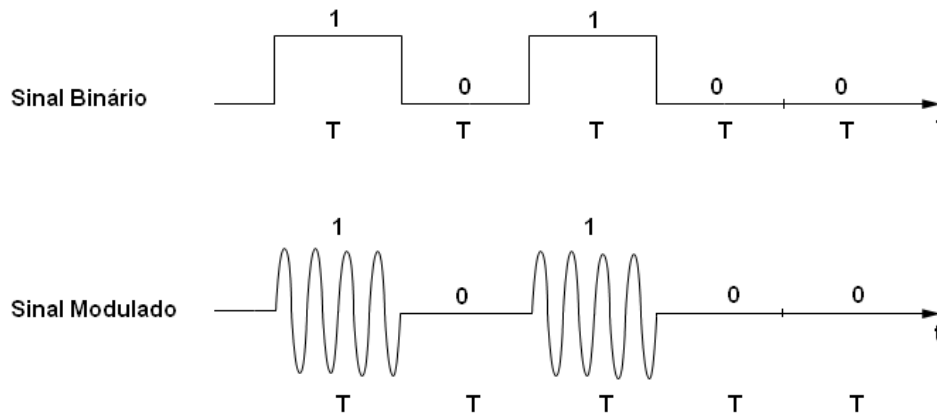


Figura B.1 – Modulação OOK (*On Off Keying 2-ASK*)

B.1.1 - Modulador OOK

A Figura B. mostra o diagrama do modulador OOK. O sinal binário é simplesmente multiplicado pela portadora produzindo o sinal de saída $s(t)$. A expressão matemática do sinal transmitido pode ser representada pela Equação B.1. Em que $p(t)$ é a função que representa os *bits* do sinal transmitido e A é a amplitude máxima da portadora.

$$s(t) = Ap(t)\cos(\omega_c t) \quad (\text{B.1})$$

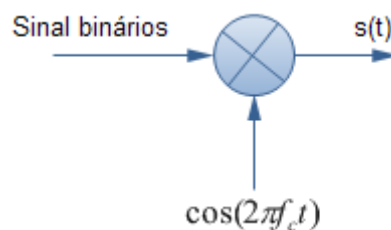


Figura B.2– Modulador OOK

Seja $P(\omega)$ a transformada de Fourier do sinal $p(t)$, então a transformada de Fourier do sinal transmitido (B.2) é dada por:

$$S(\omega) = \frac{A}{2} [P(\omega - \omega_c) + P(\omega + \omega_c)] \quad (\text{B.2})$$

Como previsto, a multiplicação pelo sinal $\cos(\omega_c t)$ desloca ao espectro de frequência do sinal original (anteriormente em banda base) e reduz a amplitude pela metade. Outro fato importante é que a largura de banda do sinal modulado é o dobro da largura de banda do

signal original, visto que agora o sinal modulado possui duas componentes, uma centrada em ω_c e outra em $-\omega_c$ [34].

B.1.2 - Demodulador OOK

O diagrama da Figura mostra o processo de demodulação de sinais OOK. A demodulação é basicamente composta por um filtro passa banda (*Bandpass Filter - BPF*) e um detector de envoltória ou envelope. O filtro passa banda é centrado na frequência de transmissão do sinal (f_c) e possui largura de banda igual à largura de banda do sinal original. Esse filtro tem a função de selecionar o sinal desejado, rejeitando o ruído [34]. O detector de envelope extrai da portadora as informações transmitidas. A Figura .3 também mostra detalhes de um detector de envelope em quadratura. Outro tipo de detector de envoltória muito utilizado em modulações em amplitude (AM) pode ser visto na Figura Por estarem na presença de ruído, as informações obtidas na saída do detector de envelope necessitam passar por um detector de nível, que possui um limiar de tolerância para cada nível de amplitude e após essa classificação a sequência binária transmitida é obtida na saída do demodulador.

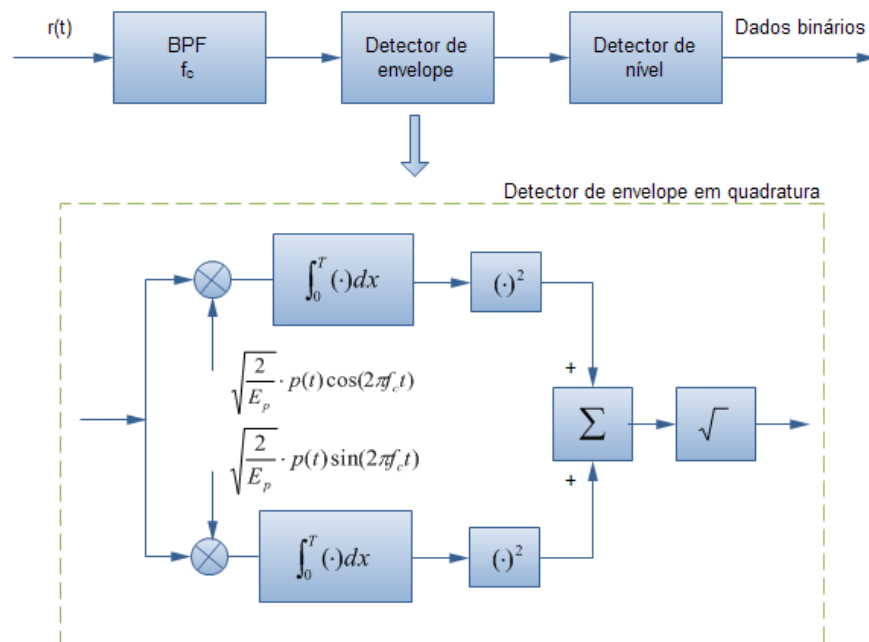


Figura B.3 – Demodulador não coerente OOK.

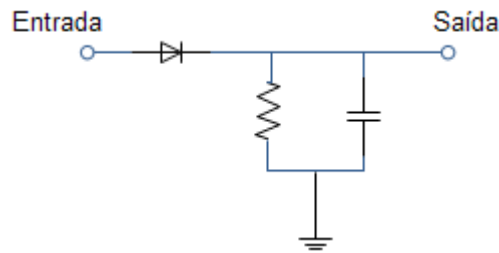


Figura B.4 – Detector de envelope para sinais AM.

B.2 - Modulação FSK

Em geral as técnicas de modulação operam modificando algumas das propriedades físicas dos sinais. Na seção anterior foi mostrada a modulação OOK, que se trata de uma modulação cuja característica física do sinal modificada é a amplitude. Nesta seção é visto outro tipo de modulação, que tem na variação da frequência do sinal seu princípio de funcionamento. A modulação FSK (*Frequency Shift Keying*) é um dos tipos de modulação mais utilizados na indústria das comunicações. Seu princípio de funcionamento é descrito a seguir.

A modulação FSK mais básica opera associando uma frequência para representar o *bit* “0” e outra para representar o “1”, essa modulação também é chamada de 2-FSK. Existem variações dessa modulação, em que são utilizadas várias frequências diferentes, uma para cada símbolo transmitido. Assim como no caso da modulação M-ASK, na qual os dados que se deseja transmitir são agrupados em blocos de n bits, com M representando o número de símbolos possíveis, na modulação M-FSK cada símbolo é associado a uma frequência diferente.

A forma 2-FSK, também chamada de FSK binária, pode ser representada pela Equação B.3.

$$\begin{aligned} s_1(t) &= A \cos(2\pi f_1 t + \Phi_1), kT \leq t \leq (k+1)T, \text{ para o bit "1"} \\ s_2(t) &= A \cos(2\pi f_2 t + \Phi_2), kT \leq t \leq (k+1)T, \text{ para o bit "0"} \end{aligned} \quad (\text{B.3})$$

O resultado disso pode ser visto na Figura B.5 que mostra o sinal modulado para uma sequência de bits que se deseja transmitir. As fases iniciais dos sinais, representadas por Φ_1 e Φ_2 , podem ser iguais ou não. Quando os sinais possuem a mesma fase, são ditos coerentes, caso contrário não-coerentes. O efeito da diferença de fase são as

descontinuidades que surgem no sinal quando ocorre uma mudança de frequência (Figura B.b), isso provoca um aumento da largura de banda do sinal.

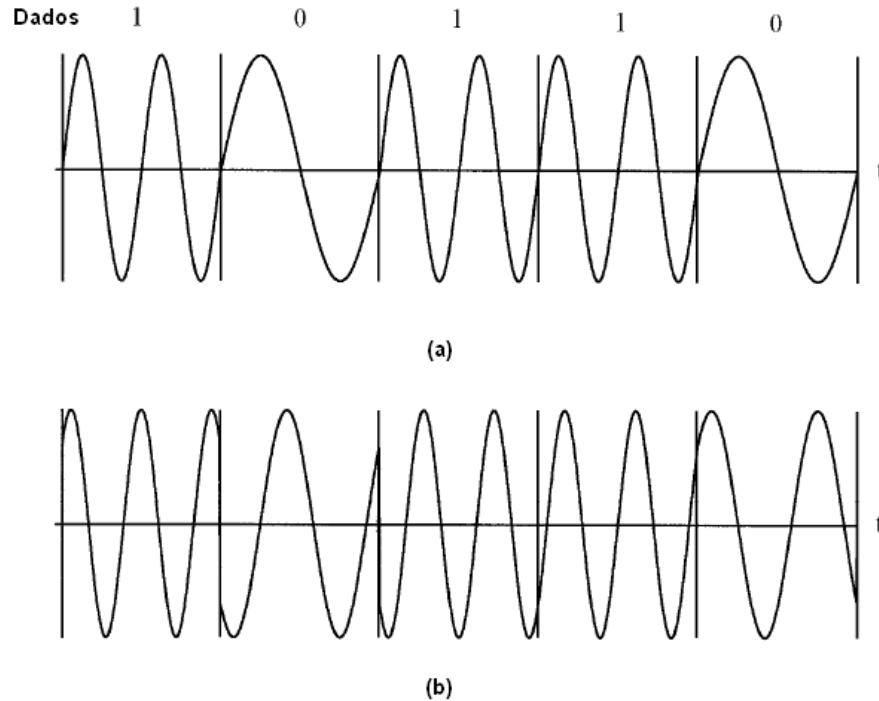


Figura B.5 – Sinal 2-FSK: (a) Sunde's FSK e (b) FSK com descontinuidade de fase

Para evitar que as descontinuidades apareçam, basta escolher as frequências f_1 e f_2 de modo que $s_1(t)$ e $s_2(t)$ sejam ortogonais, ou seja, a Equação 3.10 é satisfeita [35]. Além disso, esses sinais devem ser coerentes ($\Phi_1 = \Phi_2$). Quando a continuidade de fase do sinal modulado é mantida, a modulação é dita *Sunde's FSK*.

$$\int_0^T s_1(t)s_2(t)dt = 0 \quad (\text{B.4})$$

B.2.1 - Modulador 2-FSK

A Figura B. mostra o diagrama de blocos do modulador FSK. Os dados binários que se deseja transmitir são aplicados a um multiplexador que irá selecionar o sinal gerado pelo Oscilador 1 ou pelo Oscilador 2. No modulador coerente, um sintetizador de frequências é utilizado para gerar as duas frequências de maneira que elas possuam a mesma fase [35].

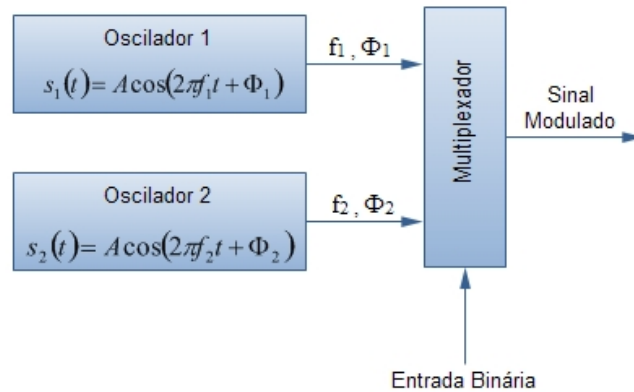


Figura B.6 – Modulador FSK Não Coerente

B.2.2 - Demodulador 2-FSK

O demodulador 2-FSK (Figura B.) utiliza as propriedades de sinais ortogonais para realizar a demodulação do sinal transmitido. No primeiro estágio do demodulador, o sinal $r(t)$, que leva em consideração o ruído do canal, é multiplicado por cossenos nas frequências f_1 e f_2 . Em seguida o sinal é integrado no período. A multiplicação por cosseno seguida da integração no período representa o produto interno, como indicado na Equação B.4. Assim, se os sinais forem ortogonais a saída é nula, caso contrário, é diferente de zero.

Supondo que o canal não introduza ruído no sinal, então se for transmitido $s_1(t)$ a saída do integrador superior será maior que zero, enquanto que a saída do integrador inferior será aproximadamente zero. A soma das saídas dos integradores passa por um *Threshold* que associa “1” para entradas positivas e “0” para negativas, obtendo-se assim os dados transmitidos. O mesmo acontece na presença do ruído $n(t)$, porém para sinais com intensidade próximas as do ruído, erros na demodulação podem ocorrer.

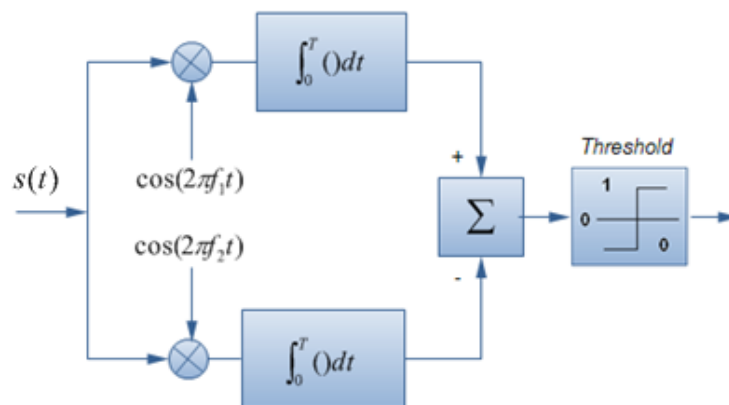


Figura B.7 – Demodulador 2-FSK

A demodulação não-coerente é um pouco mais complexa que a coerente. Um demodulador não coerente pode demodular um sinal coerente, mas o contrário não ocorre. Assim, sinais não coerentes só podem ser demodulados por demoduladores não coerentes. A dificuldade da demodulação não coerente é trabalhar com sinais com fase desconhecida, o que implica na utilização de funções de correlação, além de filtros combinados [35].

B.3 - Modulação GFSK

A modulação GFSK (*Gaussian Frequency Shift Keying*) é praticamente idêntica a modulação FSK, com a diferença que a modulação GFSK é uma modulação de fase contínua e nessa modulação o sinal em banda base é passado através de um filtro gaussiano antes de entrar no modulador FSK. A função do filtro gaussiano é suavizar as transições de amplitude que ocorrem no sinal em banda base, feito isso, reduz-se a quantidade de componentes de alta frequência existentes no sinal original, melhorando a eficiência espectral e o consumo de energia.

B.4 - Modulação MSK

A modulação MSK é um tipo de modulação de fase contínua e que pode ser derivada de outras modulações, como por exemplo, da modulação OQPSK (*Offset Quadrature Phase Shift Keying*) e da CPFSK (*Continuous Phase Frequency Shift Keying*). Por preservar a fase na transmissão de um *bit* não há transição abrupta no sinal, essa característica faz com que a modulação MSK possua uma melhor distribuição espectral.

O sinal MSK possui uma componente em fase e outra em quadratura, ou seja, os canais $I(t)$ e $Q(t)$ são modulados com portadoras ortogonais com período de $4T$. A Equação B.5 mostra a expressão para o sinal modulado. A Figura B.8 ilustra a modulação de uma sequência binária onde se pode observar a formação de um sinal MSK.

$$s(t) = AI(t)\cos\left(\frac{\pi t}{2T}\right)\cos\left(\pi f_c t\right) + AQ(t)\sin\left(\frac{\pi t}{2T}\right)\sin\left(\pi f_c t\right) \quad (\text{B.5})$$

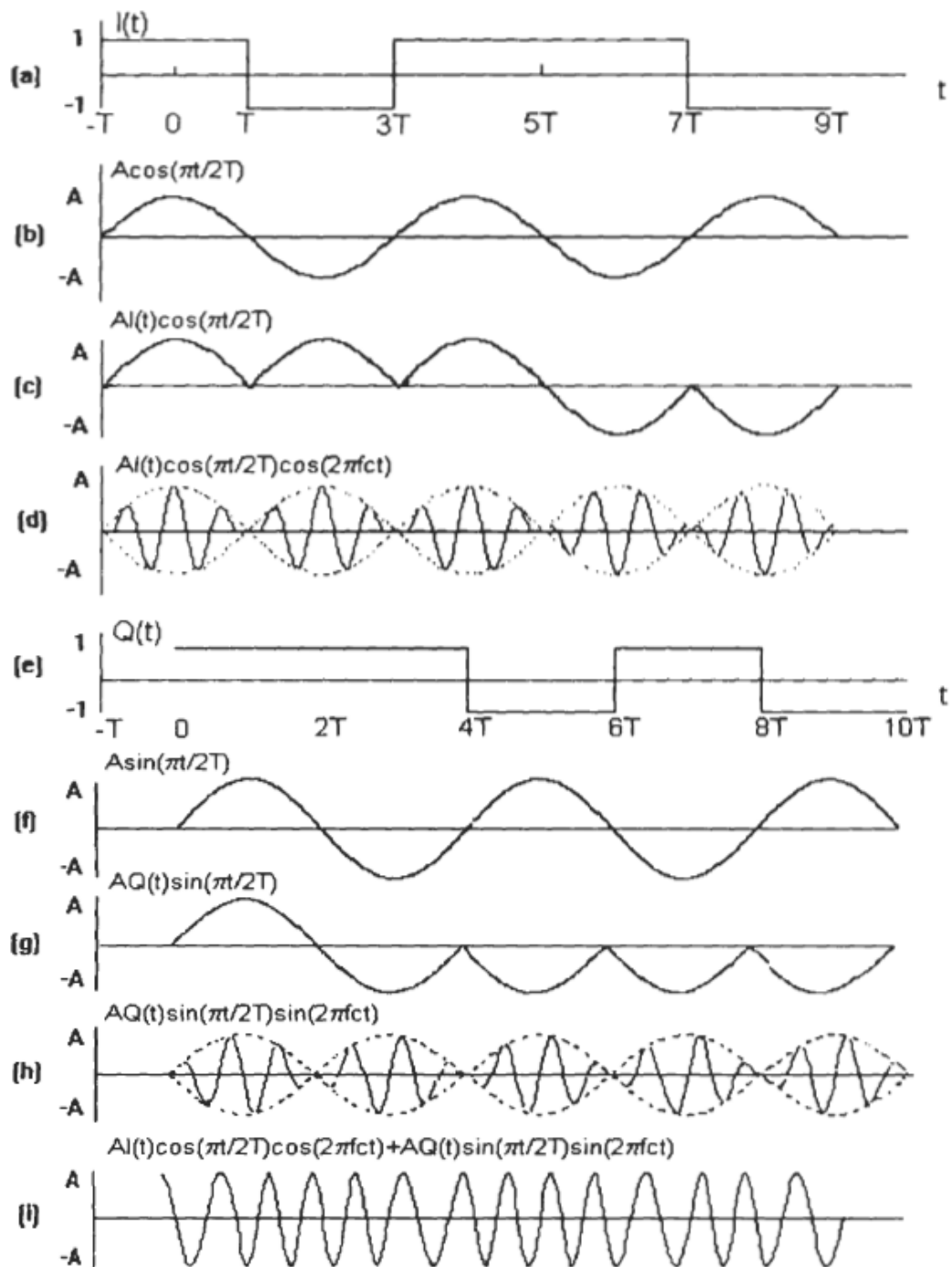


Figura B.8 – Modulação MSK

É importante notar que $I(t)$ e $Q(t)$ representam a parte par e ímpar (respectivamente) de um sinal binário, sendo obtida por meio de um conversor série/paralelo.

B.4.1 - Modulador MSK

O modulador MSK pode ser visto na Figura B.9, como dito anteriormente o sinal binário $r(t)$ passa por um conversor, de modo a gerar as componentes $I(t)$ e $Q(t)$. Após passar pelo conversor os sinais são modulados em fase e em quadratura e em seguida é aplicada a portadora, por fim os dois sinais são somados e transmitidos. Um detalhe é que o sinal em quadratura é deslocado de um período em relação ao sinal em fase e cada símbolo de I e Q possuem um intervalo de $2T$.

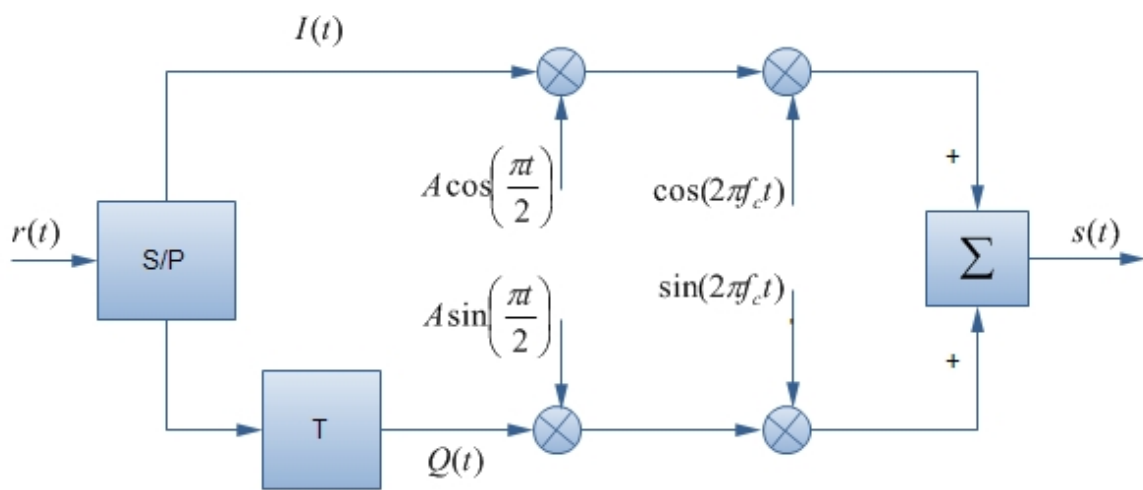


Figura B.9 – Modulador MSK

B.4.2 - Demodulador MSK

Os primeiros passos da demodulação do sinal MSK seguem basicamente a mesma linha do sinal FSK, ou seja, no receptor o sinal é multiplicado novamente pela portadora e filtrado com um filtro passa-baixa, a fim de obter as componentes $I(t)$ e $Q(t)$ em fase e quadratura, em seguida à essas componentes é aplicado o produto interno (definido na Equação B.4) como explicado na modulação FSK. O estágio final (*Threshold*) restaura os níveis discretos dos sinais que compõe a sequência binária transmitida.

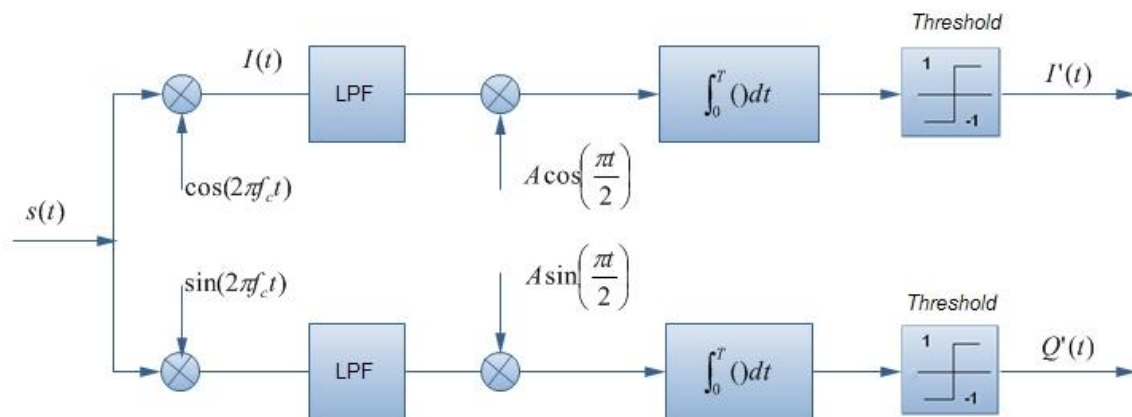


Figura B.10 – Demodulador MSK

Livros Grátis

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

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)