

UNIVERSIDADE REGIONAL DE BLUMENAU – FURB
CENTRO DE CIÊNCIAS TECNOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**ESTUDO E IMPLEMENTAÇÃO DE TECNOLOGIA SEM FIO USANDO
PROTOCOLO ZIGBEE**

MICHAEL EBERLE SIEMEINTCOSKI

BLUMENAU

2009

Livros Grátis

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

Milhares de livros grátis para download.

MICHAEL EBERLE SIEMEINTCOSKI

**ESTUDO E IMPLEMENTAÇÃO DE TECNOLOGIA SEM FIO USANDO
PROTOCOLO ZIGBEE**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Tecnológicas da Universidade Regional de Blumenau – FURB como requisito parcial à obtenção do grau de Mestre em Engenharia Elétrica.

Prof. Dr. Orlando José Tobias - Orientador

BLUMENAU

2009

ESTUDO E IMPLEMENTAÇÃO DE TECNOLOGIA SEM FIO USANDO PROTOCOLO ZIGBEE

MICHAEL EBERLE SIEMEINTCOSKI

Esta Dissertação foi julgada adequada para obtenção do título de Mestre em Engenharia Elétrica, Área de Concentração Sistemas de Energia, Linha de Pesquisa Eletrônica Industrial, e aprovada em sua forma final pelo Programa de Pós Graduação em Engenharia Elétrica da Universidade Regional de Blumenau.

Prof. Orlando José Tobias, Doutor
Orientador, FURB

Prof. Adriano Péres, Doutor, FURB
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Sérgio Vidal Garcia Oliveira, Doutor, FURB
Presidente

Prof. Adriano Péres, Doutor, FURB

Prof^o Patrick Kuo-Peng, Doutor, UFSC

Blumenau, 19 de junho de 2009

Dedico este trabalho a todos os mestrandos que buscam conciliar horas de atividade acadêmica, trabalho e família. A jornada é exaustiva, mas recompensadora.

AGRADECIMENTOS

Meus agradecimentos a todos que contribuíram para a realização deste trabalho.

A minha esposa Eliane, que sempre acreditou na conclusão deste trabalho.

Ao Prof^o Orlando José Tobias, pela dedicação e paciência na orientação dos estudos.

Ao Prof^o Adriano Peres, pela confiança depositada em nosso trabalho.

Ao Prof^o Sérgio Vidal, pela competência e orientação na preparação da apresentação da defesa de mestrado.

Ao meu irmão Charles, por não medir esforços e usar de sua competência na correção ortográfica deste trabalho.

A grande amiga Luiza Kleinubing, que apesar de suas inúmeras atribuições, encontra tempo para utilizar sua sabedoria e conhecimento na organização deste trabalho.

Aos colegas de mestrado Alexandre José Araújo dos Santos e Juliane Maira Bento, verdadeiros guerreiros e pessoas sinceras.

Ao SENAI/SC que acreditou e financiou grande parte de meus estudos.

A FURB, por oportunizar esta formação no campus de Blumenau, sendo impraticável para minha pessoa, fora do mesmo.

E, finalmente a Deus, por sua onipresença e poder incalculável.

RESUMO

As tecnologias das redes sem fio apresentam diferenciais competitivos relevantes, tanto no aspecto construtivo quanto no aspecto de venda do produto. Estas tecnologias permitem que equipamentos industriais fixos, como conversores de frequência, utilizem uma rede sem fio para realizar diversas tarefas como transferência de dados, monitoramento, controle, dentre outras.

Neste trabalho é estudado o protocolo Zigbee que utiliza o padrão IEEE 802.15.4 e a norma Zigbee Alliance, especialmente projetado para aplicações em ambientes industriais e prediais. As diversas tecnologias de comunicação sem fio disponível, bem como a análise da norma IEEE 802.15.4 protocolo Zigbee e Zigbee Alliance, são analisados neste trabalho. Dentre os temas abordados, destacam-se as principais características das camadas físicas (PHY), enlace (MAC), rede (NWK) e aplicação (APL) do protocolo Zigbee; os tipos de topologias da rede, bem como os modos de acesso a rede, denominado acesso ao meio; os tipos de conexão e criação de rede; as ferramentas de hardware, de programação, controle e supervisão, bem como a apresentação de resultados obtidos experimentalmente. Durante o estudo, algumas plataformas de processamento de sinais digitais são estudadas. É necessário o estudo e experimento de uma plataforma de 8 bits e uma de 16 bits. O estudo do protocolo de comunicação SPI é necessário para melhor entendimento da comunicação entre microcontrolador e a antena de transmissão de dados.

Palavras – Chave: Transmissão de dados. Dispositivos eletrônicos sem fio. Rede sem fio. Ambientes industriais. Comunicação sem fio. Fluxo de dados. Comunicação de dados.

ABSTRACT

The wireless's networks technologies have relevant differential competitive, both in appearance and in the constructive aspect of selling the product. These technologies allow industrial equipment fixed, and voltage converters, using a wireless network to perform various tasks such as data transfer, monitoring, control, among others.

In this work is studied the Zigbee protocol that uses the IEEE 802.15.4 standard and the Zigbee Alliance norm, specially projected to applications in industrial and public environment. The diverse technologies of wireless communication available, as well as the analysis of the IEEE 802.15.4 standard, Zigbee protocol and Zigbee Alliance, are analyzed in this work. Among the broached subjects, the main characteristics of the physic layer are detached (PHY), enlace (MAC), web (NWK) and application (APL) of the Zigbee protocol; the types of topologies of the web, as well as the access ways to the web, called access to the environment; the types of connection and creation of web; the tools of hardware, programming, controlling and supervising, as well as the presentation of results gotten experimentally. During the study, some digital signal processing platforms are studied. It is needed the study and experiment of a 8 bits platform and a 16 bits platform. The study of the communication protocol SPI is needed to the best understanding of the communication between the microcontroller and the data transmit antenna (transiver).

Keywords: Ad hoc network. Low data rate. Low power. LR-WPAN. Mobility. Personal area network (PAN). Radio frequency (RF). Short range. Wireless. Wireless personal area network (WPAN).

LISTA DE SIGLAS

APL - Camada de aplicação

CPU - Unidade Central de Processamento

DSC - Controlador Digital de Sinais

DSP – Processador Digital de Sinais (Digital Signal Processing)

GTSs - Alocamento de Espaços (Slots) em Tempo Garantido

IEEE - Institute of Electrical and Electronics Engineers

LR-WPAN - Wireless Personal Area Network the Low Rate

MAC – Camada de enlace (Media Access Control)

MCU – Microcontrolador

NWK - Camada de rede

OSI – Sistema Internacional de protocolo aberto (Open Systems Interconnection)

PC - Contador de Programa

PHY – Camada física (Physical)

POS – Espaço de operação pessoal (Personal Operating Space)

SAP - Serviços de ponto de acesso (Access Point)

WPAN - Wireless Personal Área Network

ZDO - Zigbee Device Object

RFD – Dispositivo final Zigbee

FFD – Dispositivo completo Zigbee

ZENA – Ferramenta de programação e supervisão do protocolo Zigbee

MPLAB – Software de programação da Microchip para microcontroladores

APS – Sub-camada da camada de aplicação APL

CFP - Período de disputa-livre

ED - Detecção de energia

LQI - Indicação da qualidade do link

MiWi – Protocolo de comunicação sem fio do fabricante Microchip

PPDU - Pacote de beacon da camada física (PHY)

LISTA DE FIGURAS

Figura 1 (a) – Família IEEE802	Figura 1(b) Sub-grupo da 802.15- WPAN	16
Figura 2 – Comparação entre modelo OSI e padrão IEEE 802		17
Figura 3 - Aplicações Zigbee		20
Figura 4 -Arquitetura do protocolo Zigbee		21
Figura 5 – Configuração Estrela		24
Figura 6 – Configuração agrupamento em árvore		25
Figura 7 – Configuração em malha		25
Figura 8 – Estrutura do Superframe		26
Figura 9 – Estrutura do Superframe com GTS's		27
Figura 10 – Comunicação do dispositivo final para o coordenador (rede sinalizado)		28
Figura 11 – Comunicação do dispositivo final para o coordenador (rede não sinalizado)		28
Figura 12 – Comunicação do coordenador para um dispositivo (rede sinalizado)		29
Figura 13 – Comunicação do coordenador para um dispositivo (rede não sinalizado)		29
Figura 14 – Serviços Primitivos		30
Figura 15 – Modelo de referência PHY: Serviços disponíveis (PD-SAP e PLME-SAP)		32
Figura 16 – Modulação e função de espalhamento		33
Figura 17 – O-QPSK chip offsets		34
Figura 18 – Seqüências do chip baseado na amostra da forma do pulso		34
Figura 19 – Frame de Beacon		35
Figura 20 – Frame de Dados		36
Figura 21 – Frame de Reconhecimento		36
Figura 22 – Frame de Comando do MAC		37
Figura 23 – Modelo referencial da camada MAC		38
Figura 24 – Sequência de mensagens descritos no serviço de dados MAC		38
Figura 25 – Formato Geral do Frame da camada MAC		39
Figura 26 – Formato Frame de controle da camada MAC		40
Figura 27 – Sequência de configuração para: Formato campo do modo de endereçamento de destino do campo de controle do frame		42
Figura 28 – Sequência de configuração para: Formato campo do modo de endereçamento da fonte do campo de controle do frame		42
Figura 29 – Modelo de referência da camada de rede NWK		44
Figura 30 – Camada de aplicação (APL)		45
Figura 31 – Exemplo de aplicação de Zigbee com circuito composto de dois nós		47
Figura 32 – Relacionamento entre os dispositivos lógicos		48
Figura 33 – Formato do Frame Geral do APS		50
Figura 34 – Formato do campo Frame de controle		50
Figura 35 – Formato do campo Controle de Frame\Tipo de Frame		51
Figura 36 – Formato do campo Controle de Frame\Modo de entrega		51
Figura 37 – Formato do campo Controle de Frame\Modo de Endereçamento Indireto		51
Figura 38 – Formato do campo Controle de Frame\Modo de reconhecimento		52
Figura 39 – Formato do campo Destinatário do Endpoint		52
Figura 40 – Formato do campo Identificador de Agrupamento (Cluster)		53
Figura 41 – Formato do campo Identificador de Perfil (Profile)		53
Figura 42 – Formato do campo Fonte do Endpoint		54
Figura 43 – Formato do campo Frame payload		54
Figura 44 – Formato do campo Frame payload\Dados		55

Figura 45 – Formato do campo Frame payload\Comando	55
Figura 46 – Formato do campo Frame payload\Reconhecimento	55
Figura 47 – (a) Definição do Frame de reconhecimento\ Normal	56
(b) Definição do Frame de reconhecimento\ Indireto	56
Figura 48 – Relação DSP x MCU x DSC	58
Figura 49 – Pinagem DSPIC30F4011	59
Figura 50 – Arquitetura da memória RAM	60
Figura 51 – Diagrama de Bloco do dsPIC30F4011 (40/44-PIN).....	63
Figura 52 – Circuito eletrônico do protocolo CAN	64
Figura 53 – Fonte de alimentação do processador e do sistema AD	64
Figura 54 – Circuito eletrônico dsPIC 30F4011	65
Figura 55 – Protótipo utilizando a plataforma em 16 bits usando o dsPIC 30F4011.....	65
Figura 56 – Unidade Central de Processamento Simplificada	68
Figura 57 – Barramento interligando CPU e Memória	69
Figura 58 – Pinagem do PIC 18F4620	70
Figura 59 – Diagrama de Bloco do PIC 18F4620	71
Figura 60 – Cristal ressonador XTAL	72
Figura 61 – Diagrama de funcionamento do oscilador do PIC 18F4620.....	73
Figura 62 – Diagrama de bloco do sistema de controle de restauração RESET do PIC18F4620	74
Figura 63 – Mapa de memória de programa (Flash) do PIC 18Fxx.....	76
Figura 64 – Mapa de memória de dados (RAM) do PIC 18Fxx.....	76
Figura 65 – Mapa de registradores do PIC 18F4620	77
Figura 66 – Exemplo de Algoritmo de multiplicação 8x8bits.....	78
Figura 67 – Conexão mestre x escravo SPI	81
Figura 68 – Transmissão da variável número 1.....	82
(a) Circuito eletrônico para transmissão e recepção de dados com protocolo SPI.....	82
(b) Pacote transmitido do SPI (Frame) – transmissão da variável 1 (o pino de referência está ligado no oscilador)	82
Figura 69 – Transmissão da variável número 2.....	82
(a) Frame do SPI – transmissão da variável 2 (o pino de referência está ligado no oscilador)	82
Figura 70 – Transmissão da variável número 3.....	83
(a) Circuito eletrônico para transmissão e recepção de dados com protocolo SPI.....	83
(b) Frame do SPI – transmissão da variável 3 (o pino de referência está ligado no oscilador)	83
Figura 71 – Arquitetura Microchip Stack do protocolo Zigbee.....	86
Figura 72 – Diretórios e subdiretórios da ferramenta Microchip Stack	87
Figura 73 –Subdiretório Common	88
Figura 74 –Subdiretório DemoCoordinator.....	88
Figura 75 –Subdiretório DemoRFD.....	89
Figura 76 –Subdiretório DemoRouter	89
Figura 77 –Subdiretório Documents.....	89
Figura 78 – Arquivos (Zigbee.DEF, myZigbee.C e nome do projeto. HEX) criados para cada aplicação	90
Figura 79 – Arquivos-fonte das camadas físicas (PHY), de enlace (MAC), de rede (NWK) e aplicação (APL) do protocolo Zigbee	91
Figura 80 – Placa de desenvolvimento PICDEM Z	92
Figura 81 – Hardware do PICDEM Z (sinais de controle adicionados)	93
Figura 82 – Software ZENA	95

Figura 83 – Hardware para captura de radiofrequência	95
Figura 84 – Arquivos criados pelo software ZENA.....	95
Figura 85 – Botão RB5 do dispositivo RFD (Placa eletrônica de desenvolvimento PICDEM™ Z)	99
Figura 86 – Modo de transferência de dados do RFD para o coordenador, com sincronismo – Norma IEEE 802.15.4.....	99
Figura 87 – Abertura de projeto no MPLAB (Arquivo fonte: MpDemoCoordApp.mcp)	101
Figura 88 – Conexão física dos dispositivos ICD2, PICDEM Z e PC.....	101
Figura 89 – Mensagem de Teste referente à conexão do ICD2.....	103
Figura 90 – Comando e Mensagem para apagar a memória no PIC18Fxx.....	103
Figura 91 – Comando e Mensagem para programação e verificação do arquivo fonte ZigBee Coordenador	104
Figura 92 – Mensagem de abertura do ZigBee Coordenador (comunicação serial RS 232).....	104
Figura 93 – Mensagem de confirmação do ID do ZigBee Coordenador.....	105
Figura 94 – Mensagem de confirmação do estado de operação do ZigBee Coordenador (rede formada)	105
Figura 95 – Comando e Mensagem para programação e verificação do arquivo fonte ZigBee RFD.....	106
Figura 96 – Mensagem de abertura do ZigBee RFD (comunicação serial RS 232)	107
Figura 97 – Mensagem de confirmação do ID do ZigBee RFD.....	107
Figura 98 – Mensagem de confirmação da associação do RFD com um coordenador Zigbee.....	108
Figura 99 – Mensagem de confirmação de conexão e saída do RFD com um coordenador Zigbee.....	108
Figura 100 – Mensagem de envio e confirmação do dado do RFD ao Coordenador ZigBee	108
Figura 101 – Mensagem de envio e não confirmação do dado do RFD ao Coordenador ZigBee	109
Figura 102a – Modo de transferência de dados do Coordenador para o RFD – Norma IEEE 802.15.4.....	109
Figura 102b – Frame de Beacon monitorado em tempo real pelo Analisador ZENA do Coordenador ZigBee	110
Figura 107 – Frame de Dados MAC monitorado em tempo real no Analisador ZENA.....	112
Figura 108 – Frame de Dados APS descrito na Norma IEEE 802.15.4.....	112
Figura 109 – Frame de Dados APS monitorado em tempo real no Analisador ZENA ..	112
Figura 110 – Botões RB5 e RB4 do dispositivo Coordenador	114
Figura 111 – Led (RA1) do dispositivo RFD.....	114
Figura 112 - Comunicação do coordenador para um dispositivo RFD (rede sinalizada): Norma IEEE 802.15.4.....	115
Figura 113 - Comunicação do coordenador para um dispositivo RFD (rede sinalizada): Analisador ZENA	115
Figura 114 – Frame de Dados APL descrito na Norma Zigbee Alliance	116
Figura 115 – Frame de Dados APL monitorado em tempo real no Analisador ZENA ..	117
Figura 116 – Mensagem de formação da rede	119
Figura 117 – Mensagem de envio e confirmação de dado (atributo) do coordenador para o RFD.....	119

LISTA DE TABELAS

Tabela 1 – Relação de frequência de operação	22
Tabela 2 IEEE 802.15.4 - Tipos de dispositivos padronizados para uma LR-WPAN	23
Tabela 3 - Relação de dispositivos entre IEEE e o protocolo Zigbee.....	23
Tabela 4 – Bandas de frequência e seus baud rates	31
Tabela 5 – Mapa: Conversão de símbolo decimal para valor modulado chip.....	33
Tabela 6 – Serviço primitivo MCPS-SAP.....	38
Tabela 7 – Tipo de Frame da camada MAC	40
Tabela 8 – Campo Modo de endereçamento para valores de destino e fonte	41
Tabela 9 – Frame de comando da camada MAC	43
Tabela 10 - Tabela de comparação de DSPIC30F4011	62
Tabela 11 – Tabela de comparação da família PIC 18Fxx	70
Tabela 12 - Estrutura de diretórios de arquivos-fonte.....	87

SUMÁRIO

1 INTRODUÇÃO GERAL	14
1.1 INTRODUÇÃO.....	14
1.2 ORGANIZAÇÃO DO PADRÃO IEEE 802.15	16
1.3 ORGANIZAÇÃO DO TRABALHO	17
2 IEEE 802.15.4 – WPAN Low Rate (curto alcance) Zigbee	19
2.1 INTRODUÇÃO.....	20
2.1.1 Tipos de Dispositivos	22
2.1.2 Acesso ao meio.....	26
2.1.3 Modo de transferência de dados	27
2.1.4 Conceito de serviços primitivos	30
2.2 CAMADA FÍSICA (PHY)	31
2.2.1 Especificações da frequência de 2.4GHz	32
2.2.2 Formato do frame (Camada Física PHY).....	35
2.3 CAMADA MAC	37
2.3.1 Serviços e especificações da camada MAC	38
2.3.2 Formatos do frame da camada MAC.....	39
2.4 CAMADA DE REDE (NWK)	43
2.5 CAMADA APL.....	44
2.5.1 Dispositivos lógicos de controle.....	46
2.5.2 Formato do frame (Camada APL)	49
2.6 CONCLUSÃO.....	56
3 DSC – CONTROLADOR DIGITAL DE SINAIS (DSPIC30F4011).....	58
3.1 INTRODUÇÃO.....	58
3.2 DSPIC 30F4011.....	59
3.3 DESENVOLVIMENTO DE PROTÓTIPO USANDO O DSPIC 30F4011	62
3.4 CONCLUSÃO.....	66
4 MICROCONTROLADOR PIC 18FL4620	67
4.1 INTRODUÇÃO.....	67
4.2 PIC 18F4620.....	67
4.3 FONTES DE INTERRUPTÃO DO MICROCONTROLADOR PIC 18F4620	78
4.4 SISTEMA DE COMUNICAÇÃO SERIAL	79
4.5 CONCLUSÃO.....	83
5 FERRAMENTAS DE DESENVOLVIMENTO	85
5.1 INTRODUÇÃO.....	85
5.2 ARQUITETURA DO MICROCHIP STACK - INTRODUÇÃO.....	85
5.3 DETALHAMENTO DO MICROCHIP STACK.....	87
5.4 PICDEM™ Z - PLACA DE DESENVOLVIMENTO	91
5.5 ZENA	94
5.6 CONCLUSÃO.....	96
6 ANÁLISE DOS RESULTADOS	97
6.1 INTRODUÇÃO.....	97
6.2 EXPERIMENTO 1: PROTOCOLO ZIGBEE.....	98
6.2.1 Dispositivo Coordenador – Experimento 1	100

6.2.2 Dispositivo RFD – Experimento 1	105
6.2.3 Analisando os dados no Analisador ZENA - Experimento 1	109
6.3 EXPERIMENTO 2: PROTOCOLO ZIGBEE.....	114
6.3.1 Descrição do experimento – Relato da camada APS	114
6.3.2 Detalhamento do Frame APS	116
6.4 CONCLUSÃO.....	120
7 CONCLUSÃO GERAL	122
REFERÊNCIAS	126
ANEXO 1 – REGISTRADOR DE ESTADO – STATUS	128
ANEXO 2 – REGISTRADOR DE CONTROLE DA MEMÓRIA EEPROM.....	129
ANEXO 3 – CÓDIGO FONTE EM ASSEMBLY PARA ROTINA DE MULTIPLICAÇÃO 16 X16 BITS.....	130
ANEXO 4 – PORTA	131
ANEXO 5 – PORTB	132
ANEXO 6 – PORTC	133
ANEXO 7 – PORTD	134
ANEXO 8 – PORTE	135
ANEXO 9 – FONTES DE INTERRUÇÃO.....	136
ANEXO 10 – REGISTRADOR INTCON	137
ANEXO 11 – REGISTRADOR INTCON2	138
ANEXO 12 – REGISTRADOR INTCON3	139
ANEXO 13 - REGISTRADOR SSPSTAT	140
ANEXO 14- REGISTRADOR SSPCON1.....	141
ANEXO 15- REGISTRADOR OSCCON E OSCTUNE.....	142
ANEXO 16 - PASSOS PARA INSTALAR OS ARQUIVOS DO MICROCHIP STACK... 143	
ANEXO 17 - ARQUIVOS FONTE MICROCHIP STACK EM SUBDIRETÓRIOS ZIGBEESTACK.....	144
ANEXO 18 - ARQUIVOS FONTES COMUNS MICROCHIP EM SUBDIRETÓRIOS COMUNS	145
ANEXO 19 - DEMONSTRAÇÃO DE COORDENADOR DE PROTOCOLO ZIGBEE EM SUB-DIRETÓRIOS DEMOCOORDINATOR	146
ANEXO 20 - DEMONSTRAÇÃO DE DISPOSITIVO FINALS DE PROTOCOLO ZIGBEE EM SUBDIRETÓRIOS DEMORFD	147
ANEXO 21 – CÓDIGO C PARA GERENCIAMENTO SPI.....	148

1 INTRODUÇÃO GERAL

Atualmente existem diversos padrões de comunicação sem fio (WiMax, Wi-Fi, Bluetooth, WLAN.802.11.b/g) com capacidade de atingir médias e altas taxas de transmissão de dados em aplicações de voz, vídeo e redes de computadores. Entretanto, ainda não está definido um padrão para transmissão de dados entre dispositivos de controle em ambientes industriais, como:

- Controle de sistemas de iluminação
- Controle de temperatura de máquinas e ambientes
- Controle de velocidade de motores elétricos
- Sistema de posicionamento de máquinas (servoacionamento)
- Controle de sensores
- Controle de talhas e pontes rolantes
- Detectores de fumaça.

1.1 INTRODUÇÃO

Os ambientes industriais exigem que os sistemas colem, armazenem, processem e transfiram dados em distâncias de até 500m. Os ambientes industriais possuem características que dificultam a propagação de ondas eletromagnéticas, como as barreiras físicas (metais, concreto, cabos elétricos), ruídos, altas temperaturas e interferências eletromagnéticas.

Um padrão de comunicação de dados utilizando tecnologia sem fio para uso em ambientes industriais, deve apresentar características técnicas compatíveis com as necessidades e condições ambientais das indústrias, devendo apresentar:

- Alcance compatível às distâncias físicas existentes nos ambientes industriais
- Taxa de transmissão de dados compatível com a velocidade exigida pelos dispositivos de controle utilizados nos ambientes industriais
- Baixo consumo de energia
- Custo reduzido para a implementação da tecnologia

As redes de comunicação sem fio, compostas de equipamentos fixos ou móveis, são chamadas de redes pessoais sem fio (*Wireless Personal Area Network* - WPAN). O alcance

dessas redes, denominado de espaço de operação pessoal (*Personal Operating Space - POS*), pode variar de 10 à 100 metros em todas as direções, seja parado ou em movimento. Esse alcance pode ser estendido com o uso de dispositivos roteadores que são responsáveis pelo aumento do alcance do sinal eletromagnético. A maioria das tecnologias WPANs utiliza um dispositivo roteador para gerenciar o fluxo de dados da rede e aumentar o alcance dos dispositivos utilizados nos ambientes industriais.

Na IEEE (2003), são descritas as principais características das WPANs:

- Baixo consumo de Energia
- Alcance: tipicamente entre 0 até 10 metros, podendo atingir até 100 metros
- Tamanho reduzido da antena de transmissão
- Baixo custo de implantação
- Permitem a sobreposição de outras redes na mesma área
- Capacidade para atender, no mínimo, 16 dispositivos na mesma área de atuação.

As WPANs são utilizadas para integrar múltiplos equipamentos fixos ou móveis definidas pelo padrão IEEE 802.15, dividindo-se em quatro grupos:

Grupo 1: IEEE 802.15.1 – *WPAN/Bluetooth*

Grupo 2: IEEE 802.15.2 – *Coexistence*

Grupo 3: IEEE 802.15.3 - *WPAN High Rate*

Grupo 4: IEEE 802.15.4 - *WPAN Low Rate* (IEEE STD 802.15.1, 2002)

Neste trabalho é estudado o padrão IEEE 802.15.4 - *WPAN Low Rate* (curto alcance) denominado protocolo Zigbee. O padrão é caracterizado por ter uma baixa taxa de transmissão (até 250Kbps), curto alcance (até 100m), baixa potência (30mW), baixo custo e facilidade de implantação. Suas principais aplicações debruçam-se na automação predial e no controle industrial em malha aberta.

O padrão IEEE 802.15.4 foi homologado com o nome de Zigbee, por um consórcio de empresas denominadas de *ZigBee Alliance* (Philips, Mitsubishi, Motorola, Honeywell, Freescale, Samsung, Texas Instruments, Siemens).

Durante o estudo, algumas dúvidas foram levantadas sobre o protocolo Zigbee, entre elas:

- Qual a distância de alcance em ambientes industriais do protocolo Zigbee?
- Como funciona e quais componentes eletrônicos são utilizados no desenvolvimento do hardware?

- Qual o consumo de energia (em horas) de um dispositivo eletrônico utilizando o protocolo Zigbee?
- Como funciona o sistema de endereçamento dos dispositivos de uma rede de comunicação Zigbee?
- Como são as topologias de uma rede de comunicação Zigbee? Como é o acesso ao meio (a conversa entre dispositivos)?
- É possível monitorar e identificar o frame (pacote de dados) de uma rede de comunicação Zigbee?
- Existem diferentes tipos de dispositivos em uma mesma rede de comunicação Zigbee?
- Existem ferramentas para programar e controlar uma rede de comunicação Zigbee?

1.2 ORGANIZAÇÃO DO PADRÃO IEEE 802.15

O IEEE (*Institute of Electrical and Electronics Engineers*) é responsável pela criação de normas que definem o funcionamento e a construção de diversos dispositivos existentes no mercado.

Em maio de 1999, constituem o grupo de trabalho IEEE 802.15, dentro da família IEEE 802, objetivando a normalização para a conectividade das redes pessoais sem fio (WPANs). Na Figura 1(a) são listados os principais padrões da família IEEE 802 e seu subgrupo é ilustrado na Figura 1(b).

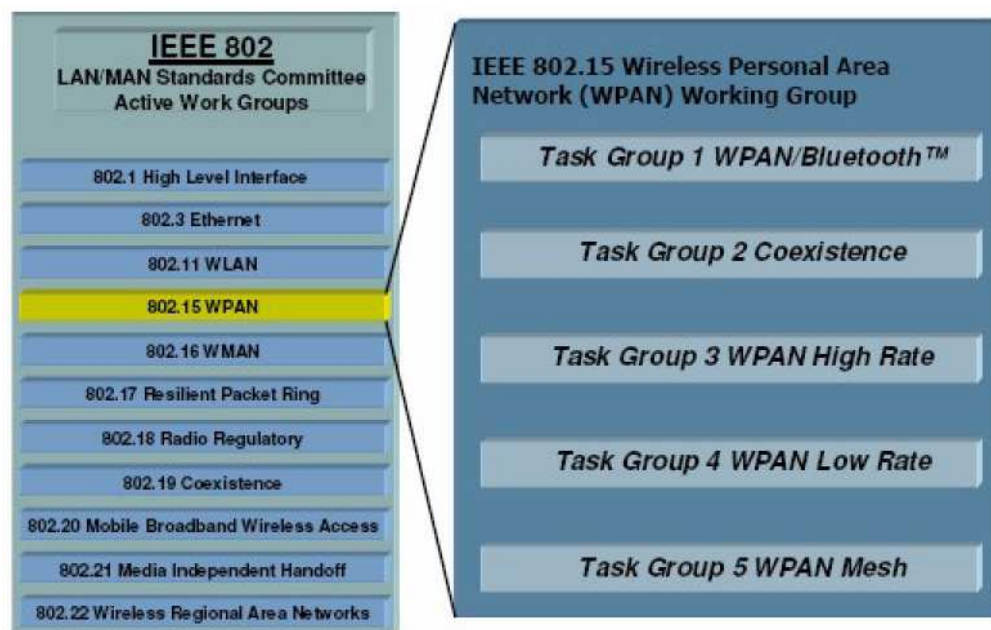


Figura 1 (a) – Família IEEE802

Figura 1(b) Sub-grupo da 802.15- WPAN

A finalidade do grupo IEEE 802.15 WPAN é definir um conjunto de especificações para as camadas físicas (*Physical* - PHY) e para a camada de controle de acesso ao meio (*Media Access Control* - MAC), objetivando a comunicação sem fio de curta distância, dentro do modelo de referência OSI (*Open Systems Interconnection*). A Figura 2 ilustra uma comparação entre o modelo OSI e o padrão IEEE 802.

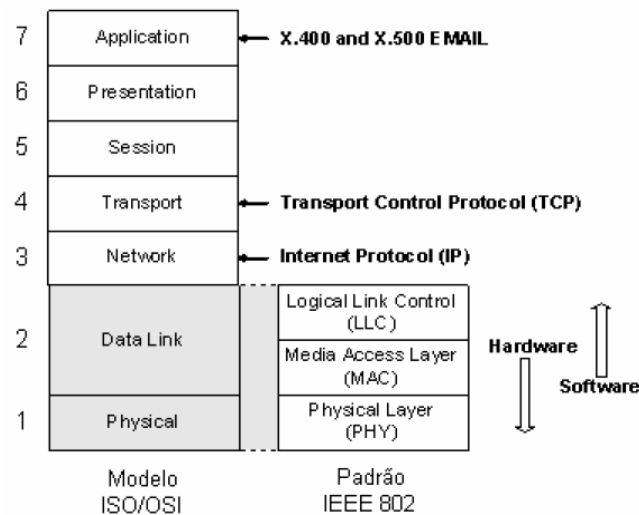


Figura 2 – Comparação entre modelo OSI e padrão IEEE 802

A camada física define como a informação é transferida para outra camada, no caso das WPANs a transferência realizada é via rádio-freqüência. A camada de enlace do modelo OSI corresponde à subcamada de controle de acesso ao meio (MAC) que define como e quando o rádio deve ser usado para comunicação (IEEE, 2003). O capítulo 2 detalha a funcionalidade das camadas.

1.3 ORGANIZAÇÃO DO TRABALHO

A presente dissertação organiza o trabalho da seguinte forma:

- O Capítulo 1 apresenta uma revisão bibliográfica sobre as tecnologias de comunicação sem fio.
- O Capítulo 2 apresenta a norma IEEE 802.15.4 protocolo Zigbee e a norma Zigbee Alliance, suas características e particularidades.

- O Capítulo 3 apresenta um estudo da plataforma de processamento de sinais digitais de 16 bits. Relaciona as características construtivas e vantagens dos DSC (Controladores de Sinais Digitais), destacando o hardware e ferramentas de desenvolvimento de software para a família dsPIC 30F40xx do fabricante Microchip.
- O Capítulo 4 apresenta um estudo da plataforma microcontrolada usando um processador de 8 bits. Relaciona as características construtivas do hardware e ferramentas para o desenvolvimento do código fonte. No mesmo capítulo, é realizado ensaios de laboratório sobre o funcionamento do protocolo de comunicação serial SPI e do periférico USART, ambos utilizados na comunicação entre a antena de rádio e a plataforma de processamento de sinais digitais.
- O Capítulo 5 estuda as ferramentas de comunicação, programação e monitoração de uma rede de comunicação Zigbee, fornecida pelo fabricante Microchip.
- O Capítulo 6 apresenta os resultados dos ensaios de laboratório da rede de comunicação Zigbee. O capítulo detalha o uso da ferramenta Microchip Stack, os resultados obtidos no estudo por meio de ensaios de bancada. Detalha o procedimento utilizado nos testes de laboratório, e busca responder as questões levantadas na introdução deste capítulo.

2 IEEE 802.15.4 – WPAN Low Rate (curto alcance) Zigbee

Neste capítulo é descrita a arquitetura normalizada pelo padrão OSI (Open System Interconnection), abrangendo as camadas físicas (PHY), camada de enlace (MAC), camada de rede (NWK) e camada de aplicação (APL). O protocolo Zigbee é definido pelo IEEE como sendo uma LR-WPAN (Wireless Personal Área Network de Low Rate) operando três diferentes faixas de frequência (2.4GHz, 915MHz e 868MHz).

Em seguida, com base nas normas do IEEE e do grupo Zigbee Alliance (*consórcio de empresas destinadas à normalização da camada de rede e da camada de aplicação do protocolo Zigbee – Motorola, Microchip, Texas, Chipcon, Freescale, Philips, Ember, Siemens, Samsung, Scheneider electric, Eaton*) são descritos três tipos de dispositivos:

- Dispositivo coordenador;
- Dispositivo roteador;
- Dispositivo final.

Estes dispositivos citados podem ser agrupados em alguma das seguintes topologias:

- Estrela;
- Árvore;
- Malha.

Este capítulo também descreve os tipos de acesso ao meio e os tipos de transferência de dados. Os tipos de acesso ao meio são definidos por dois mecanismos:

- Sinalizado;
- Não-sinalizado.

O modo sinalizado difere do não-sinalizado pela presença de sincronizadores de pacotes (frames), denominados de *beacons*. Os beacons são pacotes de dados objetivando a sincronização de uma rede Zigbee. São gerados pelo dispositivo coordenador da rede. A transferência de dados entre as camadas (PHY, MAC, NWK e APL) é realizada por meio dos Serviços de Access Point (SAP). Ela ocorre por meio de três tipos de mecanismos:

- Transferência de dados para o Coordenador;
- Transferência de dados do Coordenador, ou
- Transferência de dados ponto a ponto.

O capítulo 2 também descreve o formato dos pacotes (frame) do protocolo Zigbee, que pode apresentar até 127 bytes de comprimento. Também são descritos os formatos dos frames da camada de enlace – MAC, as características da camada de rede e suas atribuições.

Ao final do capítulo 2 a camada de aplicação (APL) é descrita observando as seguintes características:

- Formato do frame da camada de aplicação (APL);
- Definição de sub-endereçamento: Dispositivos lógicos de controle (Perfil, cluster, End Point e Atributo);
- Configurações da camada de aplicação (APL).

2.1 INTRODUÇÃO

A norma IEEE 802.15.4 descreve as características técnicas do protocolo Zigbee. O Zigbee é um protocolo de comunicação em rede sem-fio, especificamente designado para dispositivos que utilizam uma baixa taxa de transmissão de dados. Dentre as aplicações que podem utilizar o protocolo Zigbee estão: redes de automação, sistemas de segurança residencial, controle de redes industriais, medição remota, dentre outros (Figura 3).

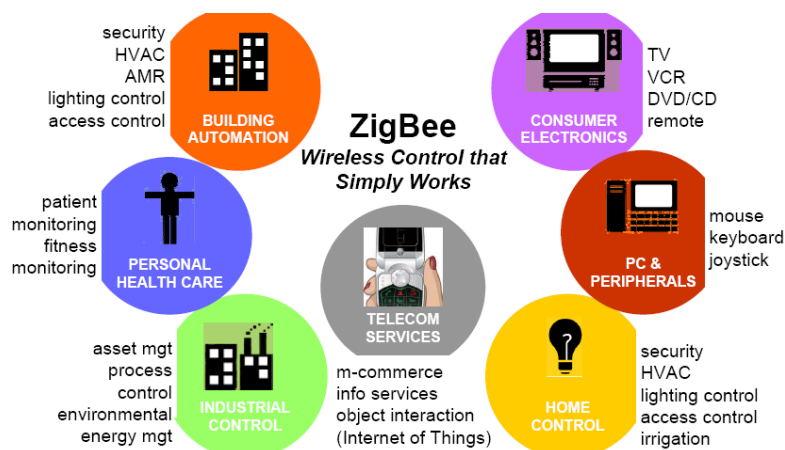


Figura 3 - Aplicações Zigbee

Comparando com outros protocolos de rede sem fio (Wi-Fi e Bluetooth), o protocolo Zigbee oferece baixa complexidade de hardware e baixo custo de implementação.

O Zigbee é um protocolo de comunicação sem fio normalizado pela IEEE 802.15.4 e pelo grupo Zigbee Alliance, fornecendo uma metodologia padrão para as funções de acordo

com o modelo OSI, incluindo formações de rede, transmissão e recepção de dados e identificação de dispositivos.

A arquitetura do Zigbee (definido como pilha) inclui um número de camadas normalizadas definidas pela IEEE 802.15.4, sendo elas, a camada média de controle de acesso (MAC) e a camada física (PHY). A camada da rede (NWK) e a camada de aplicação (APL) são normalizadas pela Zigbee Alliance. Cada uma destas camadas fornece serviços e aplicações necessárias para o funcionamento do protocolo. A Figura 4 apresenta a arquitetura do protocolo Zigbee de acordo com o IEEE e a Zigbee Alliance.

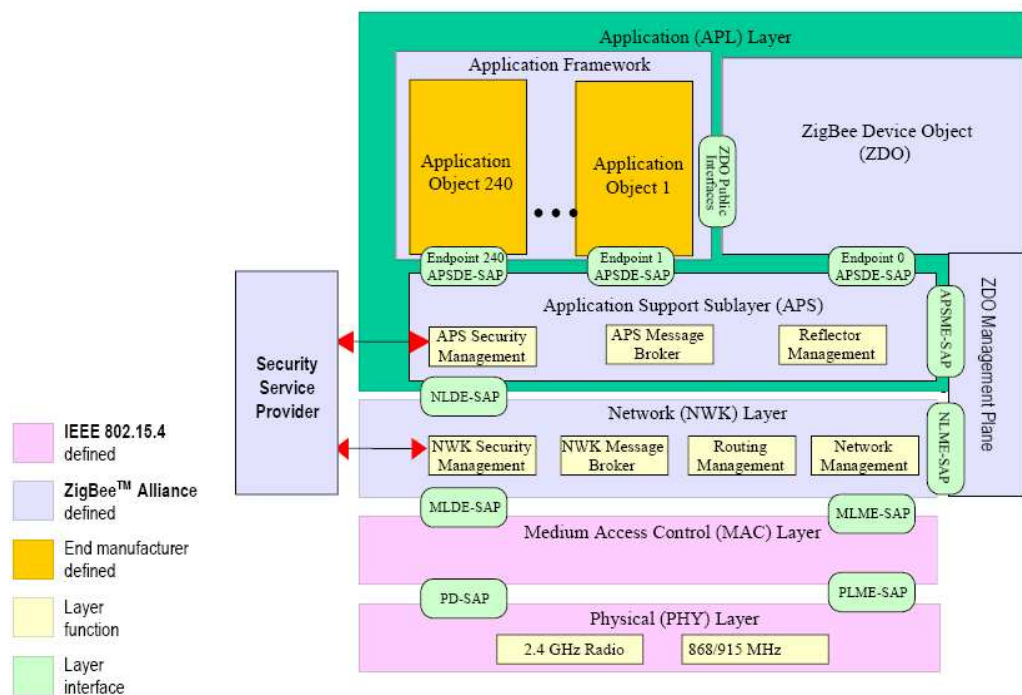


Figura 4 -Arquitetura do protocolo Zigbee

O protocolo Zigbee é classificado pelo IEEE como sendo uma rede LR-WPAN (Wireless Personal Área Network - Low Rate). Uma LR-WPAN é uma rede de comunicação de dados que permite a conectividade sem-fio nas aplicações com baixa taxa de transmissão (máximo de 250kbps).

As principais características de uma rede LR-WPAN são:

- Facilidade na instalação
- Transferência de dados com segurança
- Operação com custo baixo de implementação
- Baixo consumo de energia elétrica, possibilitando o uso de baterias

- Operação em topologia estrela, malha e ponto a ponto
- Endereçamento de 16 ou 64 bits.

A norma IEEE 802.15.4 estabelece três possíveis faixas de frequências de operação: 2.4GHz, 915MHz e 868MHz. Cada faixa de frequência oferece um número fixo de canais. Por exemplo, a faixa de frequência de 2.4GHz oferece 16 canais (canais 11-26); a faixa 915MHz oferece 10 canais (canais 1-10); enquanto que a faixa de 868MHz oferece 1 canal (canal 0). A Tabela 1 relaciona as faixas de frequência e suas respectivas características técnicas.

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868/915	868–868.6	300	BPSK	20	20	Binary
	902–928	600	BPSK	40	40	Binary
2450	2400–2483.5	2000	O-QPSK	250	62.5	16-ary Orthogonal

Tabela 1 da Norma IEEE 802.15.4 – Relação de frequência de operação

A velocidade da taxa de transmissão de dados (baud rate) do protocolo Zigbee depende da frequência de operação. Assim a faixa de 2.4GHz oferece 250kbps; a faixa 915MHz oferece 40kbps, enquanto que a faixa 868MHz oferece 20kbps.

2.1.1 Tipos de Dispositivos

A IEEE 802.15.4 padroniza dois tipos de dispositivos para uma rede LR-WPAN, que são mostrados na Tabela 2. A Tabela 3 apresenta os três tipos de dispositivos existentes no protocolo Zigbee e faz uma relação com a definição da IEEE.

Um protocolo de rede Zigbee pode assumir vários tipos de configurações. Em todas as configurações de rede há, pelo menos, dois componentes principais:

- Um Coordenador do Nó e
- Um Dispositivo final.

O coordenador de um protocolo Zigbee é um dispositivo de função completa (FFD), apresentando as seguintes funções: ligar e desligar outros tipos de dispositivos através da

rede, formar a rede, alocar endereços nos demais tipos de dispositivo da rede, controlar a transferência de dados, entre outras.

Tipos de dispositivos	Serviços Oferecidos	Fonte Típica	Configuração típica do receptor
Completo (FFD)	A maioria ou tudo	Rede elétrica	Ligado quando inativo
Reduzido (RFD)	Limitado	Bateria	Desligado quando inativo

Tabela 2 IEEE 802.15.4 - Tipos de dispositivos padronizados para uma LR-WPAN

Tipos de dispositivos do protocolo Zigbee	Tipo de dispositivo da IEEE	Função típica
Coordenador	FFD: dispositivo com funções completas	Liga através da rede, forma rede, aloca endereços de rede, armazena a ligação em tabela
Roteador	FFD: dispositivo com funções completas	Opcional, estende a capacidade física da rede, permite mais nós ligados na rede, pode executar também monitorando e/ou controle das funções
Dispositivo final	FFD: dispositivo com funções completas RFD: dispositivo com funções reduzidas	Executa monitorando e/ou controlando as funções de execução

Tabela 3 - Relação de dispositivos entre IEEE e o protocolo Zigbee

Um dispositivo final pode ser um dispositivo de função completa (FFD) ou um dispositivo com função reduzida (RFD). Um RFD é o menor e mais simples nó do protocolo Zigbee, executando apenas funções operacionais, ou seja, podendo ligar ou desligar uma saída digital ou monitorar o valor de uma entrada digital ou analógica.

Uma rede WPAN com protocolo Zigbee pode ainda optar pelo dispositivo roteador. Dentre suas funções podemos destacar a possibilidade de estender a capacidade física da rede, permitindo um maior número de nós ligados na mesma rede, aumentando o alcance da rede.

Uma rede com protocolo Zigbee pode ser configurada segundo as seguintes topologias: topologia em estrela, agrupamento em árvore ou malha. A configuração de uma rede em topologia estrela consiste de um coordenador e um ou mais dispositivos finais. Na topologia em estrela, todos os fins de dispositivos se comunicam somente com o coordenador. Se um dispositivo final precisar transferir dados para outro dispositivo, será feito um envio de

seus dados para o coordenador que, por sua vez, enviará os dados para o destinatário. A Figura 5 ilustra uma rede em protocolo Zigbee com topologia estrela.

Outra configuração de rede é a topologia de agrupamento em árvore. Nesta configuração, dispositivos finais podem interligar com o coordenador do protocolo Zigbee ou com o roteador do protocolo Zigbee.

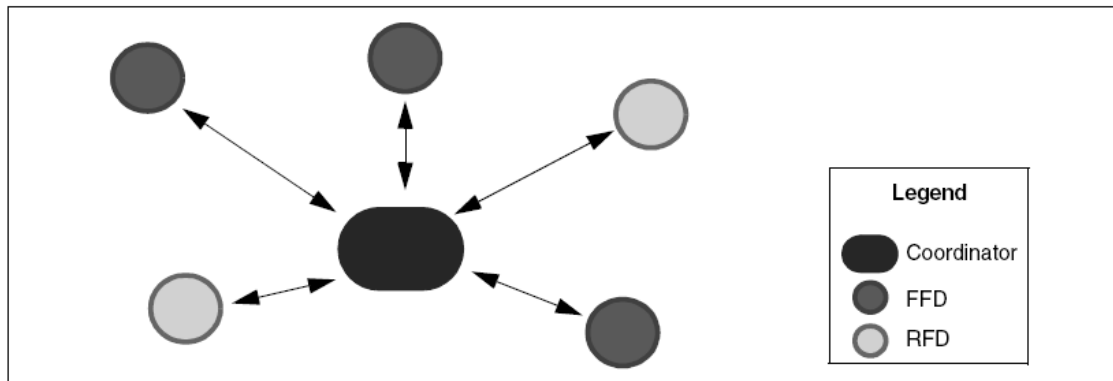


Figura 5 – Configuração Estrela

Os dispositivos roteadores têm duas funções. A primeira é aumentar o número de nós que podem ser ligados na rede; a segunda é estender o alcance físico da rede. Com a adição de um roteador, um dispositivo final não necessita estar no mesmo canal de frequência da portadora do coordenador. Este dispositivo final pode comunicar-se com um roteador que, por sua vez, comunica-se com o coordenador da rede. A Figura 6 ilustra uma rede em protocolo Zigbee com topologia agrupamento em árvore.

Uma rede em malha é similar às configurações do agrupamento em árvore, exceto que os FFDs podem rotear mensagens diretamente para outros FFDs, ao invés de seguir a estrutura de árvore, ou seja, não precisam passar os dados pelos roteadores. As mensagens para os RFDs ainda têm que passar pelo coordenador. A vantagem dessa topologia é que o tempo das mensagens latentes pode ser reduzido, aumentando a confiabilidade do sistema. A Figura 7 ilustra uma rede com uma topologia em malha.

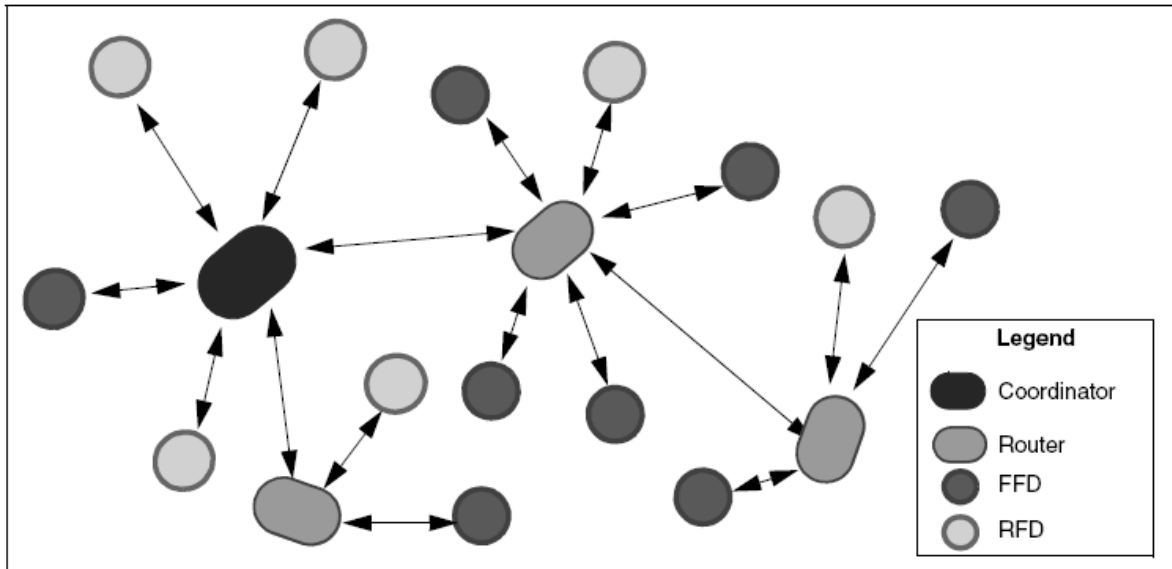


Figura 6 – Configuração agrupamento em árvore

O agrupamento em árvore e a topologia em malha são também conhecidos como rede de multiponto, devido às habilidades para realizar transferência por meio de dispositivos, enquanto que a topologia estrela é uma rede de um único ponto.

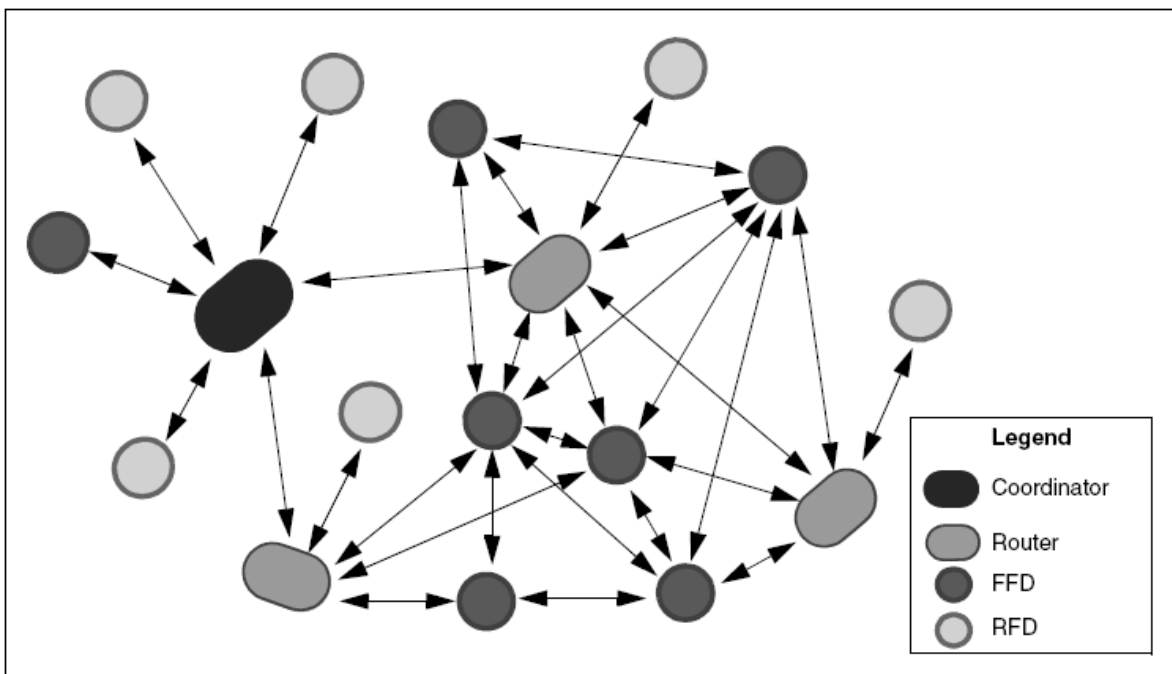


Figura 7 – Configuração em malha

2.1.2 Acesso ao meio

Uma rede sem-fio com protocolo Zigbee é uma rede de multiacesso, significando que todos os nós na rede têm igual acesso ao barramento de comunicação. Há dois tipos de mecanismos de multiacesso: sinalizado e não sinalizado.

Em uma rede não sinalizada, todos os nós da rede permitem transmitir a qualquer hora, contanto que o canal esteja ativo.

Em uma rede sinalizada, é permitida a transferência dados somente em intervalos de tempo (slots) pré-definidos. O coordenador inicia periodicamente com um sinal de sincronismo (denominado de frame de beacons), onde todos os nós da rede (dispositivos) passam a se sincronizar com este frame. Cada nó está autorizado a estabelecer comunicação em períodos de disputa de acesso (CAP), permitindo transmitir e receber dados.

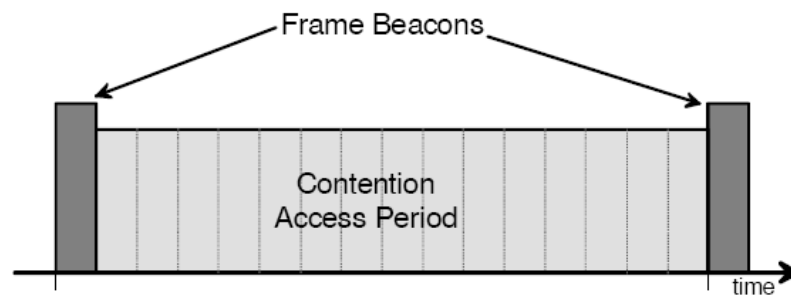


Figura 8 – Estrutura do Superframe

Este período de disputa de acesso, emitido pelo coordenador (ver Figura 8) é dividido em 16 intervalos de tempo iguais. O pacote de dados para sincronização (frame de beacons) é transmitido no primeiro intervalo de tempo de cada superframe. Se um coordenador não desejar usar uma estrutura de superframe, pode desligar as transmissões de beacon, usadas para sincronizar os dispositivos da rede. Todo dispositivo que deseje se comunicar durante o período de disputa de acesso (CAP) entre dois beacons, competirá com outros dispositivos. Todas as transmissões serão terminadas pelo tempo do beacon seguinte da rede.

Ainda em uma rede sinalizada, pode haver uma parcela ativa e outra inativa. Durante a parcela inativa, o coordenador pode entrar em uma modalidade de baixo consumo (low-power). Na parcela ativa, para aplicações que requerem a necessidade de comunicação entre alguns dispositivos específicos, o coordenador pode dedicar parcelas do frame ativo para comunicação exclusiva com um grupo de dispositivos. Estas parcelas são chamadas espaços (slots) de tempo garantido (GTSs - Alocamento de Slots em Tempo Garantidos). A forma do

GTSs possui um período de disputa-livre (CFP), aparecendo sempre na extremidade do frame ativo, que começa imediatamente depois do período de disputa de acesso (CAP), como ilustrado na Figura 9. O coordenador pode alocar até sete GTSs, entretanto, a maior parcela é destinada para o período de disputa ao acesso (CAP) com outros dispositivos da rede ou com novos dispositivos que desejam acessar a rede. Todas as transações do CAP estarão completas antes que o CFP comece. Da mesma forma, cada dispositivo que transmite em um GTS se assegurará de que sua transação esteja completa antes do período de GTS seguinte.

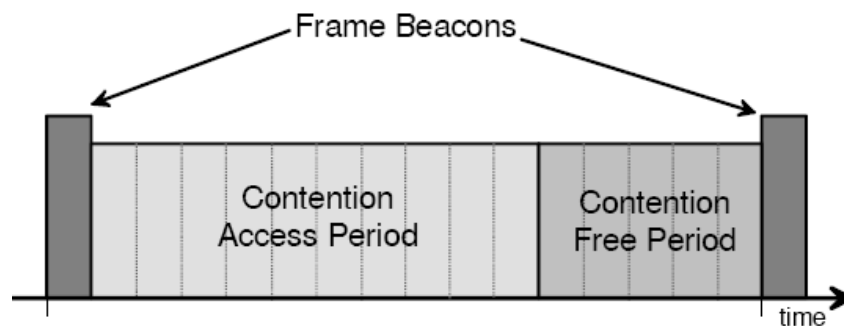


Figura 9 – Estrutura do Superframe com GTS's

A versão atual da ferramenta de desenvolvimento Microchip (utilizada nos experimentos de laboratório) suporta redes sinalizadas e não-sinalizadas.

2.1.3 Modo de transferência de dados

Existem três tipos de transferência de dados: o modo de transferência de dados de um dispositivo final para um coordenador; o modo de transferência de dados de um coordenador para um dispositivo final e, finalmente, o modo de transferência de dados entre dois fins de dispositivo da rede. Na topologia em estrela, somente os dois primeiros modos destas transações são usados, uma vez que os dados podem ser trocados somente com o gerenciamento de um coordenador. Os mecanismos para cada tipo de transferência dependem se a rede suporta a transmissão de pacotes de sincronização (beacons).

Modo 1: Transferência de dados para o Coordenador

Este modo de transferência de dados é o mecanismo para transferir dados de um dispositivo final para um coordenador. Quando um dispositivo final deseja transferir dados a

um coordenador em uma rede sinalizada, ele aguarda até receber o sinal de beacon da rede. Quando o sinal de sincronização (frame de beacons) é encontrado, o dispositivo sincroniza-se com a estrutura do superframe. No momento apropriado, o dispositivo final transmite seu pacote de dados ao coordenador, o coordenador reconhece a recepção bem sucedida dos dados, transmitindo um pacote de dados opcional do reconhecimento. Esta seqüência é ilustrada na Figura 10.

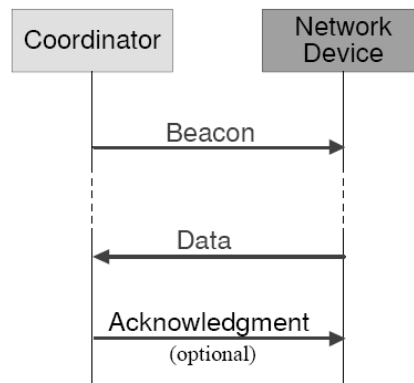


Figura 10 – Comunicação do dispositivo final para o coordenador (rede sinalizado)

Quando um dispositivo final deseja transferir dados em uma rede não sinalizada (sem frame de beacon), ele transmite simplesmente seu pacote de dados ao coordenador. O coordenador reconhece a recepção bem-sucedida dos dados transmitindo um pacote de dados opcional de reconhecimento. Esta troca de informações é ilustrada na Figura 11.

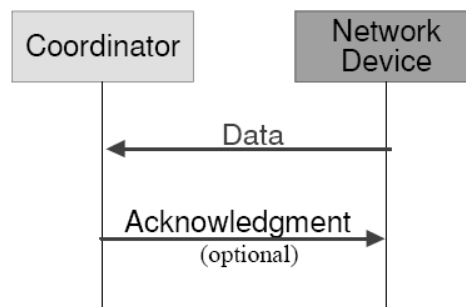


Figura 11 – Comunicação do dispositivo final para o coordenador (rede não sinalizado)

Modo 2: Transferência de dados do Coordenador

Este modo de transferência de dados é o mecanismo para realizar a transferência de dados de um coordenador para um dispositivo final. Quando o coordenador deseja transferir dados em uma rede sinalizada, envia um sinal de sincronização (frame de beacons) para a rede com uma mensagem de dado pendente. O dispositivo final periodicamente monitora os

beacons da rede e, se uma mensagem estiver pendente, transmite um comando solicitando os dados. O coordenador reconhece a recepção bem-sucedida do pedido dos dados, transmitindo um pacote de dados opcional de reconhecimento. O pacote de dados pendente é emitido e o dispositivo final reconhece a recepção bem sucedida dos dados, transmitindo um frame de reconhecimento. Após receber o pacote de reconhecimento, a mensagem é removida da lista de mensagens pendentes no coordenador. Esta troca de informações é ilustrada na Figura 12.

Quando um coordenador deseja transferir dados a um dispositivo final em uma rede não sinalizada, ele armazena os dados para que o dispositivo apropriado estabeleça o contato e solicite os dados. Um dispositivo final pode fazer o contato transmitindo um pedido de dados a seu coordenador. O coordenador reconhece o requerimento dos dados transmitindo um frame de reconhecimento. Se os dados forem pendentes, o coordenador transmite o pacote de dados ao dispositivo final. Se os dados não forem pendentes, o coordenador transmite um frame de dados com uma mensagem (de comprimento zero) para indicar que nenhum dado era pendente. O dispositivo reconhece a recepção bem-sucedida dos dados transmitindo um frame de reconhecimento. Esta troca de informações é ilustrada na Figura 13.

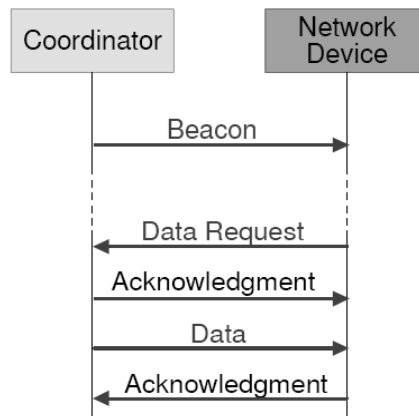


Figura 12 – Comunicação do coordenador para um dispositivo (rede sinalizado)

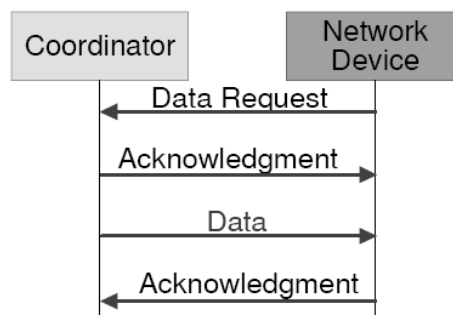


Figura 13 – Comunicação do coordenador para um dispositivo (rede não sinalizado)

Modo 3: Transferência de dados Ponto a ponto.

Em uma rede de ponto a ponto, cada dispositivo final pode se comunicar com outro dispositivo final de mesma frequência de operação. A fim de realizar eficazmente esta ação, os dispositivos que desejam se comunicar necessitará estabelecer constantemente o sincronismo com outros dispositivos.

2.1.4 Conceito de serviços primitivos

Os serviços de uma camada são as potencialidades de oferta ao usuário da camada mais elevada, ou, nos serviços da camada mais baixa. Este conceito é ilustrado na Figura 14, mostrando a hierarquia de serviços e o relacionamento entre dois usuários.

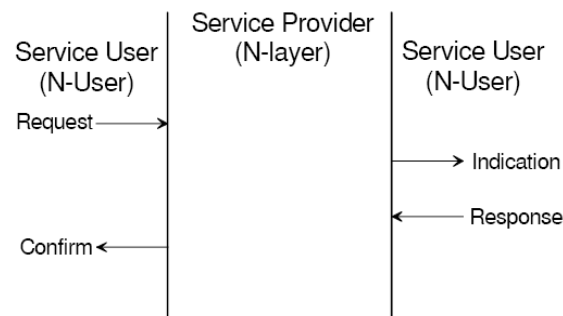


Figura 14 – Serviços Primitivos de acordo com a Norma IEEE 802.15.4

Os serviços são especificados descrevendo o fluxo de informação entre o usuário-N e a camada-N. Este fluxo de informação é modelado pelos eventos discretos instantâneos que caracterizam a provisão de um serviço. Cada evento discreto consiste em passar um serviço primitivo de uma camada para outra camada por meio de um Serviço de Acesso Point - SAP.

Um serviço primitivo pode ser algum dentre os quatro tipos genéricos detalhados a seguir:

- Pedido (Request);
- Indicação (Indication);
- Resposta (Response);
- Confirmação ou Reconhecimento (Confirm);

- a) **Pedido:** O serviço primitivo de pedido é passado do usuário-N à camada-N para solicitar que um serviço seja iniciado.
- b) **Indicação:** O serviço primitivo de indicação é passado da camada-N ao usuário-N para indicar um evento interno da camada-N que seja significativo ao usuário-N.
- c) **Resposta:** O serviço primitivo de resposta é passado do usuário-N à camada-N para terminar um procedimento solicitado previamente por um serviço primitivo de indicação.
- d) **Confirmar:** O serviço primitivo de confirmação é passado da camada-N ao usuário-N para saber os resultados de um ou mais serviços primitivos de pedido.

2.2 CAMADA FÍSICA (PHY)

A camada física (PHY) é responsável pelas seguintes tarefas:

- Ativação e desativação do transmissor/receptor de rádio
- Detecção de energia (ED)
- Indicação da qualidade do link (LQI)
- Seleção da frequência do canal
- Transmissão e recepção física dos dados.

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868/915	868–868.6	300	BPSK	20	20	Binary
	902–928	600	BPSK	40	40	Binary
2450	2400–2483.5	2000	O-QPSK	250	62.5	16-ary Orthogonal

Tabela 4 – Bandas de frequência e seus baud rates

Em um total de 27 canais, numerados de 0 a 26 ilustrado na Tabela 4, dezesseis canais estão disponíveis na faixa de 2450 MHz, 10 na faixa de 915 MHz e um na faixa de 868 MHz. A frequência central destes canais é definida como segue:

$$F_c = 868.3 \text{ in megahertz, for } k = 0$$

$$F_c = 906 + 2(k - 1) \text{ in megahertz, for } k = 1, 2, \dots, 10$$

$$\text{and } F_c = 2405 + 5(k - 11) \text{ in megahertz, for } k = 11, 12, \dots, 26$$

where

k is the channel number.

A camada física fornece uma relação entre a camada MAC e o canal de rádio físico, através de um programa (firmware) e do hardware de rádio frequência (RF).

A camada PHY inclui uma entidade de gerência chamada de PLME. O PLME é responsável por manter uma base de dados dos fins de dispositivos controlados que pertencem à camada PHY. Esta base de dados é consultada como a base da informação da rede WPAN.

A camada PHY fornece dois serviços alcançados através de dois SAPs: serviço de manutenção dos dados da camada PHY alcançado através do PD-SAP e o serviço de gerência da camada PHY, alcançado através do PLME-SAP. A Figura 15 ilustra os dois serviços da camada física.

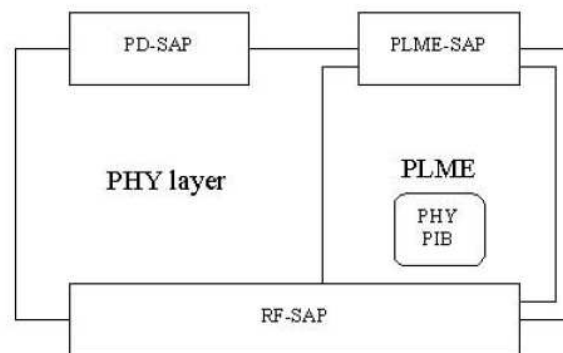


Figura 15 – Modelo de referência PHY: Serviços disponíveis (PD-SAP e PLME-SAP)

2.2.1 Especificações da frequência de 2.4GHz

A taxa de dados de acordo com a IEEE 802.15.4 (2450 MHz) será de 250 kb/s.

O canal de 2450 MHz da camada PHY emprega uma técnica de modulação denominada ary-16 quasi-orthogonal. Durante cada período de dados, quatro bits de informação são usados para selecionar uma seqüência pseudo-randomica de 16 bit (PN) para ser transmitida. As seqüências do PN, para a transmissão dos dados, são concatenadas e

agregadas em um chip (sinal binário formado por 32 bits) modulado na frequência portadora que usa uma chave offset de inversão de fase de quadratura (offset de quadrature phase-shift keying / O-QPSK). O diagrama de bloco ilustrado na Figura 16 fornece uma referência à modulação em 2450 MHz da camada PHY.

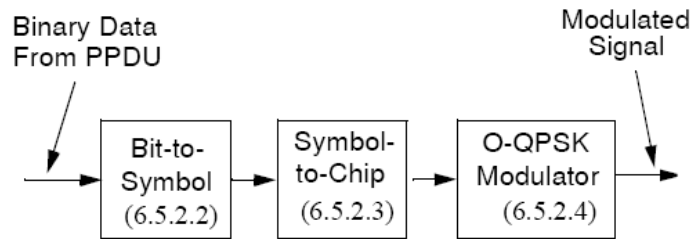


Figura 16 – Modulação e função de espalhamento

Data symbol (decimal)	Data symbol (binary) (b_0, b_1, b_2, b_3)	Chip values ($c_0 c_1 \dots c_{30} c_{31}$)
0	0000	11011001110000110101001000101110
1	1000	11101101100111000011010100100010
2	0100	00101110110110011100001101010010
3	1100	00100010111011011001110000110101
4	0010	01010010001011101101100111000011
5	1010	00110101001000101110110110011100
6	0110	11000011010100100010111011011001
7	1110	10011100001101010010001011101101
8	0001	10001100100101100000011101111011
9	1001	10111000110010010110000001110111
10	0101	01111011100011001001011000000111
11	1101	01110111101110001100100101100000
12	0011	00000111011110111000110010010110
13	1011	01100000011101111011100011001001
14	0111	10010110000001110111101110001100
15	1111	11001001011000000111011110111000

Tabela 5 – Mapa: Conversão de símbolo decimal para valor modulado chip

Todos os dados binários contidos no PPDU (pertence a camada PHY) serão codificados usando a modulação O-QPSK. Dentro de cada byte, o menor símbolo significativo (b_0, b_1, b_2, b_3) é processado primeiramente, enquanto que o símbolo mais significativo (b_4, b_5, b_6, b_7) é processado posteriormente. Cada símbolo de dado será convertido em uma seqüência de 32 bit como especificado na Tabela 5.

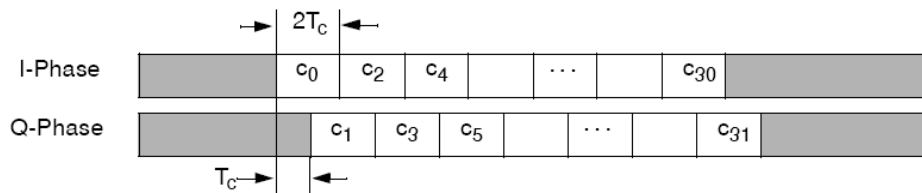


Figura 17 – O-QPSK chip offsets

Os chips pares ($c_0, c_2, c_4\dots$) são modulados no portador (I) in-phase e os chips ímpares ($c_1, c_3, c_5\dots$) são modulados no portador (Q) da quadratura-fase. Cada símbolo de dado é representado por uma seqüência de 32 bits. A taxa de chip é de 2.0 Mchip/s e representa 32 vezes a taxa do símbolo. Para dar forma ao offset entre I-phase e Q-fase, os chips da Q-fase serão atrasados pelo tempo T_c com relação a I-fase (ver figura 17), onde o T_c é o inverso da taxa do chip.

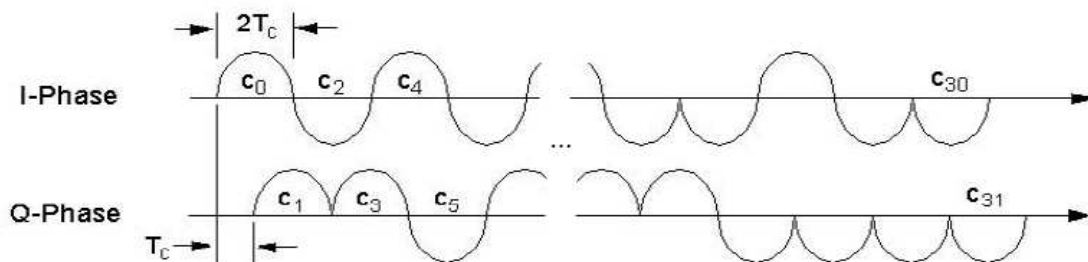


Figura 18 – Seqüências do chip baseado na amostra da forma do pulso

Durante cada período do símbolo, o chip menos significativo c_0 é transmitido primeiramente e o chip mais significativo, c_{31} , é transmitido por último. Este assunto, por tratar de características técnicas designadas pela área de telecomunicações, não é o foco dos estudos deste trabalho.

2.2.2 Formato do frame (Camada Física PHY)

A estrutura do frame foi projetada para manter uma baixa complexidade no sistema e, ao mesmo tempo, sendo robusto para trabalhar em ambientes suscetíveis a ruídos. Cada camada sucessiva do protocolo adiciona à estrutura com cabeçalhos. O LR-WPAN define quatro estruturas de frame na camada física:

- Um frame de beacon usado por um coordenador para sincronizar uma rede
- Um frame de dados usado para todas as transferências de dados
- Um frame de reconhecimento, usado para confirmação da recepção bem-sucedida
- Um frame de comando, usado para assegurar todo controle de transferências.

A Figura 19 apresenta a estrutura de um **frame de beacon**, que se origina na camada MAC. Um coordenador pode transmitir beacons em uma rede sinalizada.

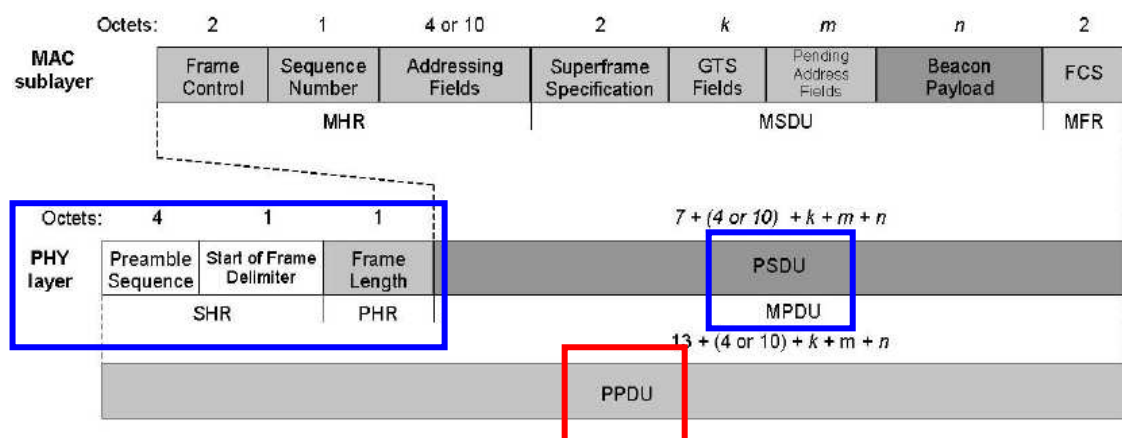


Figura 19 – Frame de Beacon

O MPDU é transferido à camada física para o PSDU como um pacote de beacon. O PSDU é prefixado com um cabeçalho de sincronização (SHR), contendo a seqüência do preâmbulo, o delimitador do início do frame (SFD) e um cabeçalho (PHR) que contém o comprimento do PSDU em bytes. O SHR, o PHR e o PSDU dão forma ao pacote de beacon do PHY (PPDU).

A Figura 20 apresenta a estrutura do **frame de dados**, o qual é originado nas camadas superiores.

Os dados são transferidos da camada do MAC (MPDU) para a camada PHY (PSDU). O PSDU é prefixado com um SHR, contendo a seqüência do preâmbulo e campos de SFD e um

PHR que contém o comprimento do PSDU em bytes. O SHR, o PHR, e o PSDU dão forma ao pacote de dados da camada PHY (PPDU).

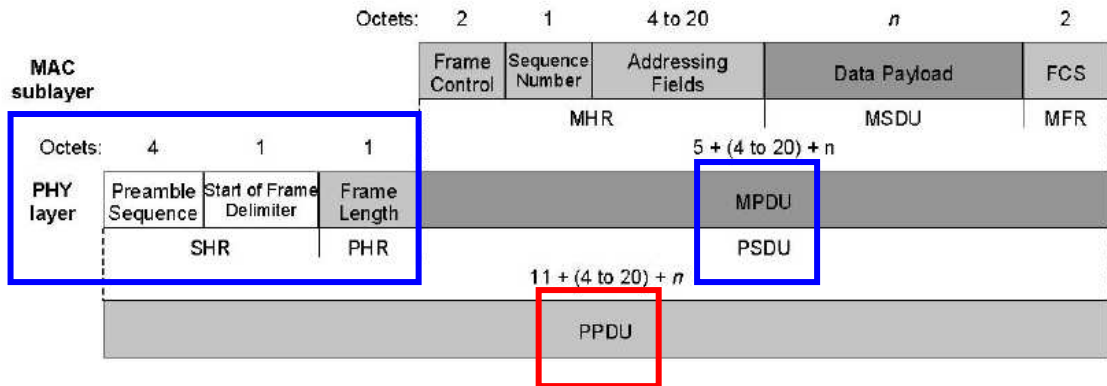


Figura 20 – Frame de Dados

A Figura 21 apresenta a estrutura do **frame de reconhecimento**. Da mesma forma que os frames anteriores, o SHR, o PHR e o PSDU dão forma ao pacote do reconhecimento PHY (PPDU).

A Figura 22 mostra a estrutura do **frame de comando**. O MPDU é passado para a camada PHY (PSDU). O SHR, o PHR e o PSDU dão forma ao pacote de comando do PHY (PPDU).

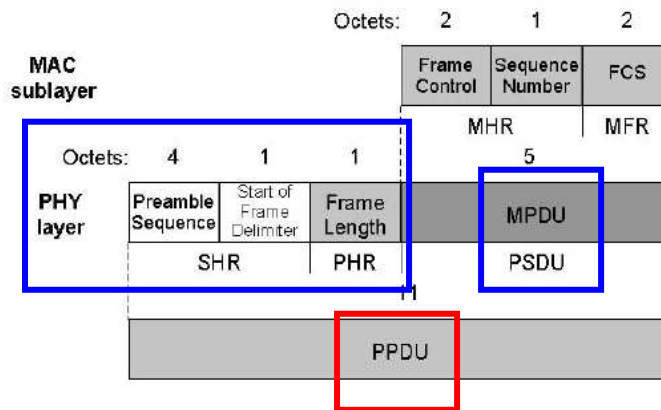


Figura 21 – Frame de Reconhecimento

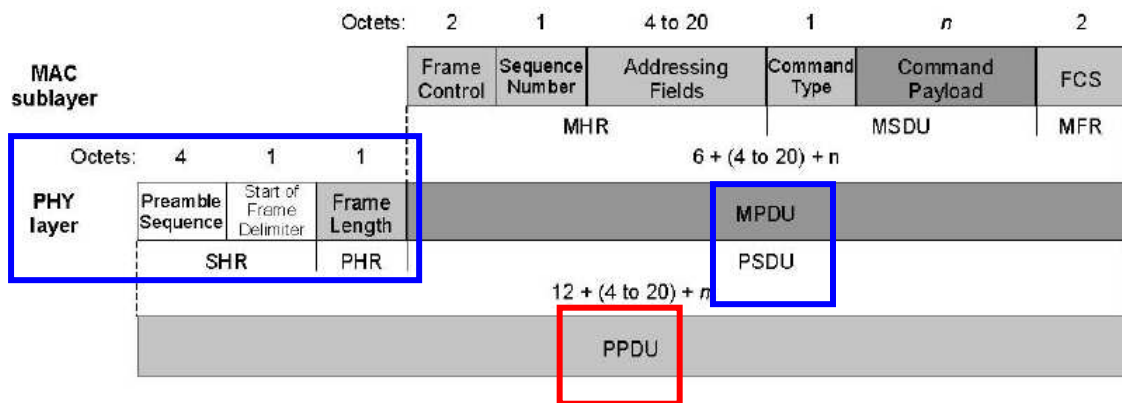


Figura 22 – Frame de Comando do MAC

O padrão IEEE 802.15.4 foi desenvolvido com uma disponibilidade limitada de fonte de alimentação, sendo desenvolvido para favorecer dispositivos que necessitem o uso de baterias (battery-powered). Em muitas aplicações os dispositivos utilizam baterias como fonte de alimentação. Esta prática é comum em dispositivos onde sua recarga ocorre em intervalos grandes. Os dispositivos baseados em baterias (battery-powered) requisitarão um ciclo para reduzir o consumo de potência. Estes dispositivos passarão a maior parte de seu tempo de funcionamento operacional em um estado do repouso (sleep). Entretanto, cada dispositivo se conectará periodicamente com o canal de radiofrequência a fim de determinar se existe uma mensagem pendente. Este mecanismo permite que o dispositivo se decida entre o consumo da bateria e a latência da mensagem.

2.3 CAMADA MAC

Neste sub-capítulo é especificada a camada de enlace (MAC) do protocolo IEEE 802.15.4. A camada do MAC assegura todo o acesso ao canal de rádio físico e é responsável pelas seguintes tarefas:

- Geração dos beacons da rede se o dispositivo for um coordenador
- Sincronização dos beacons
- Segurança dos dados do dispositivo
- Endereçamento da ID.

2.3.1 Serviços e especificações da camada MAC

A camada MAC fornece dois serviços através de dois SAPs:

- 1) Serviço de dados, usando o MCPS-SAP e
- 2) Serviço de gerência do MAC, usando o MLME-SAP.

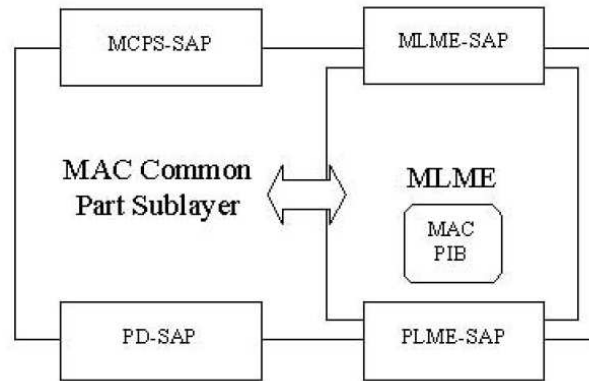


Figura 23 – Modelo referencial da camada MAC

A Tabela 6 apresenta os serviços primitivos suportados pelo MCPS-SAP. Os serviços primitivos marcados com (♦) são opcionais para um RFD.

MCPS-SAP primitive	Request	Confirm	Indication
MCPS-DATA	X	X	X
MCPS-PURGE	♦	♦	—

Tabela 6 – Serviço primitivo MCPS-SAP

A Figura 24 ilustra a seqüência de mensagens necessárias para realizar a transferência de dados de forma bem-sucedida entre dois dispositivos.

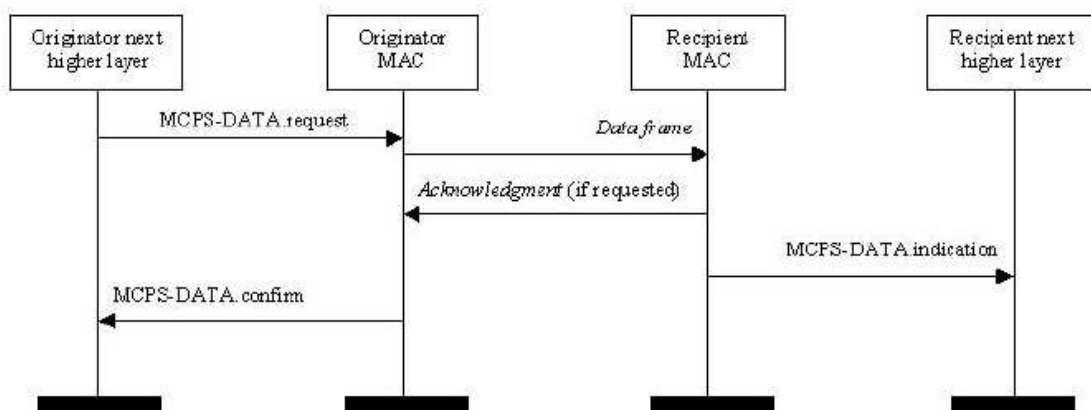


Figura 24 – Seqüência de mensagens descritos no serviço de dados MAC

2.3.2 Formatos do frame da camada MAC

Cada frame do MAC (MPDU) consiste dos seguintes componentes básicos:

a) Um MHR que compreende o controle do frame, o número de seqüência e a informação do endereço.

b) Um payload do MAC de comprimento variável contendo a especificação do tipo do frame. Os frames de reconhecimento não contêm um payload.

c) Um MFR que contém um frame para checagem de seqüência – FCS, ou seja, um frame para checagem de erros.

Os frames na camada MAC são descritos como uma seqüência de campos em uma ordem específica. Todos os formatos do frame são descritos na ordem em que são transmitidos pela camada PHY, da esquerda para a direita, onde o bit menos significativo é transmitido primeiramente. Os bits dentro de cada campo são numerados de 0 (leftmost é o menos significativo) a $k - 1$ (rightmost é o mais significativo), onde o comprimento do campo é de k bits. Os campos longos com mais de um byte são emitidos ao PHY, na ordem do byte que contém os bits menos significativos ao byte que contém os bits mais significativos.

O frame geral do MAC será formatado como ilustrado na Figura 25.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Figura 25 – Formato Geral do Frame da camada MAC

O campo de controle do frame (Frame control) possui 16 bit de comprimento, contendo informação que define o tipo do frame, os campos de endereçamento e outros flags (bandeiras) de controle. A Figura 26 apresenta a subdivisão do campo de controle do frame (Frame control).

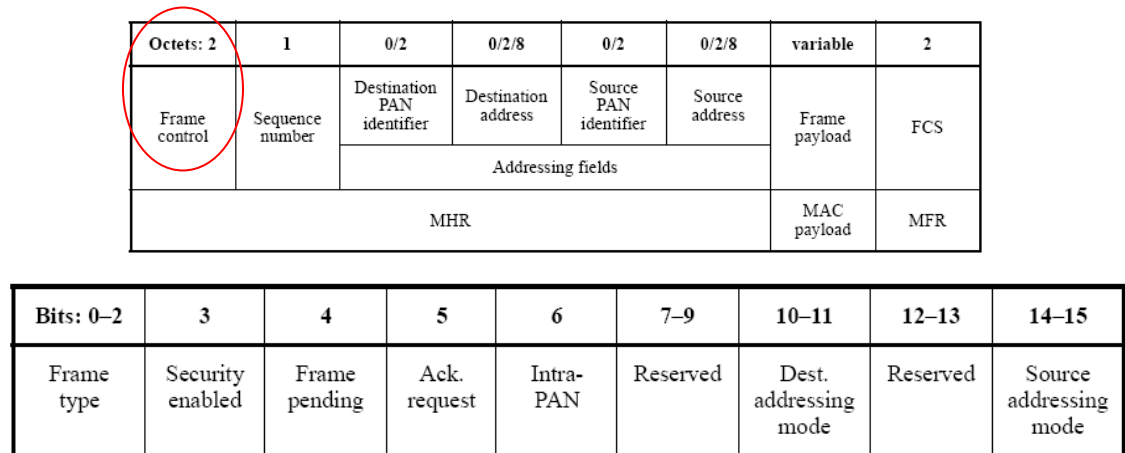


Figura 26 – Formato Frame de controle da camada MAC

O tipo de frame (Frame type) é definido por três bits e é ajustado para um dos valores listados na Tabela 7.

Frame type value $b_2 b_1 b_0$	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100–111	Reserved

Tabela 7 – Tipo de Frame da camada MAC

O campo de segurança (Security enabled) é habilitado com um bit de comprimento e está ajustado a 0 se o frame não for protegido por criptografia. Se o campo é ajustado a um, o frame estará protegido, usando as chaves de criptografia definidas pela camada MAC.

O campo pendente do frame (Frame pending) é configurado com um bit de comprimento. Estando com valor 1, ele informa que o dispositivo que emite o frame possui uma pendência de dado para transmitir.

O campo de reconhecimento de pedido (Ack. Request) é configurado com 1 bit de comprimento, especificando se um reconhecimento é requerido do dispositivo receptor de um frame de dados ou de comando da camada MAC. Se este campo tiver valor 1, o dispositivo-receptor emitirá um frame de reconhecimento após ter recebido e validado um frame de dados ou comando. Se este campo tem valor 0, o dispositivo-receptor não emitirá um frame do reconhecimento.

O campo da intra-PAN é configurado com um bit de comprimento, especificando se o frame da camada MAC deve ser emitido dentro da mesma PAN (intra-PAN) ou para outra PAN (inter-PAN). Se este campo for igual a 1 e os endereços do destino e da fonte estiverem atuais, o frame deve ser emitido para outra PAN. Se este campo for igual a 0 e os endereços do destino e da fonte estiverem atuais, o frame de controle emitirá o dado para a mesma PAN.

O campo da modalidade de endereçamento do destinatário e da fonte (*Dest. Addressing mode e Source Addressing mode*) é configurado com dois bits de comprimento, sendo ajustado a um dos valores alistados na Tabela 8.

Addressing mode value $b_1 b_0$	Description
00	PAN identifier and address field are not present.
01	Reserved.
10	Address field contains a 16 bit short address.
11	Address field contains a 64 bit extended address.

Tabela 8 – Campo Modo de endereçamento para valores de destino e fonte

O campo número de seqüência do frame geral (Sequence number) é configurado com 8 bits de comprimento, especificando um identificador original da seqüência para o frame.

O campo do identificador da PAN do destinatário (Destination PAN identifier) no frame geral é constituído por 16 bits de comprimento, especificando o ID original da PAN. Um valor de 0xFFFF neste campo representará o identificador da PAN de transmissão, que será aceito como um identificador válido da PAN por todos os dispositivos localizados neste canal. (Figura 27).

O campo de endereçamento de destino (Destination address) é constituído por 16 ou 64 bits de comprimento. Um valor de 16 bits (0xFFFF) representará como um endereço curto, válido por todos os dispositivos localizados nesse canal. (Figura 27).

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
Addressing fields							
MHR						MAC payload	MFR

Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame type	Security enabled	Frame pending	Ack. request	Intra-PAN	Reserved	Dest. addressing mode	Reserved	Source addressing mode

Addressing mode value $b_1 b_0$	Description
00	PAN identifier and address field are not present.
01	Reserved.
10	Address field contains a 16 bit short address.
11	Address field contains a 64 bit extended address.

Figura 27 – Seqüência de configuração para: Formato campo do modo de endereçamento de destino do campo de controle do frame

O campo do identificador da PAN da fonte (Source PAN identifier) é constituído por 16 bits de comprimento, especificando o identificador original da PAN do frame. (Figura 28).

O campo de endereço da fonte (Source address) é constituído por 16 ou 64 bits de comprimento, de acordo com o valor especificado no subcampo do modo de endereçamento da fonte do campo do controle do frame, especificando o endereço de origem do frame.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame type	Security enabled	Frame pending	Ack. request	Intra-PAN	Reserved	Dest. addressing mode	Reserved	Source addressing mode

Addressing mode value $b_1 b_0$	Description
00	PAN identifier and address field are not present.
01	Reserved.
10	Address field contains a 16 bit short address.
11	Address field contains a 64 bit extended address.

Figura 28 – Seqüência de configuração para: Formato campo do modo de endereçamento da fonte do campo de controle do frame

Os frames de comando definidos pela camada MAC são listados na Tabela 9. Os comandos do MAC serão transmitidos somente no CAP para a PAN com beacons (sinal

sincronizador), habilitados ou em qualquer momento para a PAN com beacons não habilitados.

Command frame identifier	Command name	RFD		Subclause
		Tx	Rx	
0 x 01	Association request	X		7.3.1.1
0 x 02	Association response		X	7.3.1.2
0 x 03	Disassociation notification	X	X	7.3.1.3
0 x 04	Data request	X		7.3.2.1
0 x 05	PAN ID conflict notification	X		7.3.2.2
0 x 06	Orphan notification	X		7.3.2.3
0 x 07	Beacon request			7.3.2.4
0 x 08	Coordinator realignment		X	7.3.2.5
0 x 09	GTS request			7.3.3.1
0 x 0a–0 x ff	Reserved			—

Tabela 9 – Frame de comando da camada MAC

2.4 CAMADA DE REDE (NWK)

A camada de rede é requerida para intermediar o funcionamento entre a camada de enlace MAC (normalizado pela IEEE 802.15.4) e a camada de aplicação (normalizado pela Zigbee Alliance). A interface da camada de rede com a camada de aplicação fornece dois serviços: serviço de transmissão de dados e serviço de gerenciamento de dados.

A entidade NLDE é responsável pelo serviço de transmissão de dados, associado a um Access Point, NLDE-SAP. A entidade NLME é responsável pelo serviço de gerenciamento de dados através da associação com um Access Point, NLME-SAP. A Figura 29 ilustra as entidades da camada de rede.

A entidade NLDE tem como atribuições:

- Geração de uma rede
- Roteamento de dispositivos específicos

A entidade NLME tem como atribuições:

- Configuração de um novo dispositivo
- Ligamento de rede de dispositivos para coordenadores ou roteadores

- Definir endereços de novos dispositivos
- Descoberta de roteadores
- Descoberta de dispositivos.

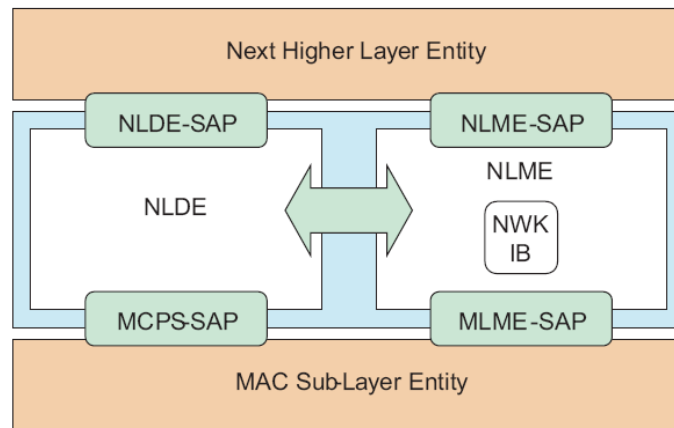


Figura 29 – Modelo de referência da camada de rede NWK

O foco deste trabalho concentrou-se nas camadas físicas (PHY), de enlace (MAC) e de aplicação (APL). A camada de rede (NWK) foi estudada, mas por tratar de uma camada intermediária, não foi analisado seu comportamento em relação as demais camadas.

2.5 CAMADA APL

A camada de aplicação (APL), ilustrada na Figura 30, possui como responsabilidades (funções exclusivas do coordenador):

- Manutenção de tabelas para ligamento
- Habilidade de combinação entre dois dispositivos baseados em seus serviços e em suas necessidades
- Comunicar mensagens entre diferentes dispositivos (sub-endereçamento).

Nesta camada, há a existência da subcamada (APS), que fornece uma interface entre a camada de rede (NWK) e a camada de aplicação (APL) através de um set geral de serviços que são usados pelos Zigbee Device Object (ZDO) e objetos de aplicação definidos pelo fabricante (Application Objects). Os serviços são fornecidos por duas entidades:

- 1º) A entidade dos dados do APS, através do Acess Point denominado APSDE-SAP
- 2º) E a entidade da gerência do APS, através do Acess Point denominado APSME-SAP.

O APSDE fornece o serviço de transmissão de dados entre dois ou mais dispositivos localizados na mesma rede. Os serviços de emissão e recepção de dados, fornecido pelo APSDE-SAP, incluem os serviços primitivos, sendo: *o pedido, a confirmação, a resposta e a indicação para transferência de dados.*

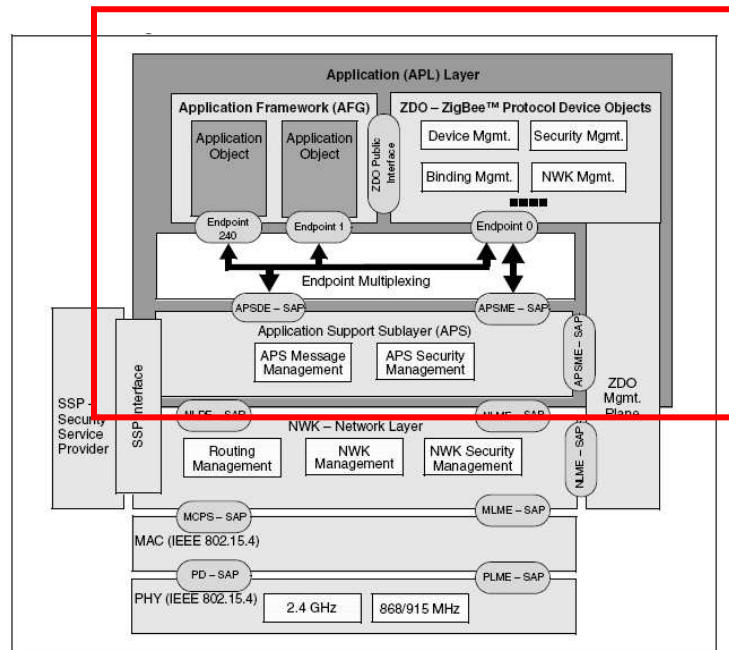


Figura 30 – Camada de aplicação (APL)

O APSME fornece serviços para a descoberta e o ligamento dos dispositivos (binding) e mantém uma base de dados de objetos controlados (tabela obrigatória para coordenador do ZigBee).

As responsabilidades do ZDO (Zigbee Device Object) incluem definir o papel do dispositivo dentro da rede (por exemplo, coordenador de ZigBee ou dispositivo de extremidade (RFD) e estabelecer um relacionamento seguro entre os dispositivos da rede.

A Aplicação Framework é o ambiente em que os objetos de aplicação (Application Objects) são alocados sobre dispositivos Zigbee. Dentro do Framework, os objetos de aplicação emitem e recebem dados através do APSDE-SAP. O controle e a gerência dos objetos de aplicação são executados através da interface ZDO public.

Até 240 objetos de aplicação distintos podem ser definidos. Cada um conectará a um endpoint posicionado de 1 a 240. Dois endpoints adicionais são definidos para o uso de APSDE-SAP:

- 1º) O endpoint 0 é reservado para a interface dos dados com o ZDO

2º) E o endpoint 255 é reservado para a interface de dados com a função de broadcast (transmissão) de dados a todos os objetos de aplicação.

Os endpoints 241-254 são reservados para o uso futuro.

2.5.1 Dispositivos lógicos de controle

A camada APL é também responsável pelo sub-endereçamento dos dispositivos lógicos de controle. Esta divisão é composta de:

- Perfil de protocolo;
- Cluster ou agrupamento;
- Atributo ou valor;
- End-point;

Um perfil no protocolo Zigbee é simplesmente a descrição de componentes lógicos (dispositivos) e suas interfaces. Os perfis são um acordo de mensagens que permitem aplicações específicas residindo em dispositivos separados para emitir comandos, solicitar dados e criar uma aplicação inter-relacionada. Os perfis são desenvolvidos por fabricantes do protocolo Zigbee para dirigir-se a soluções e necessidades específicas de tecnologia na indústria ou nas áreas prediais. Os perfis são simultaneamente soluções técnicas dentro do padrão de Zigbee, concentrando esforços de usabilidade dentro de uma área de aplicação específica. Por exemplo: Um termostato em um nó, comunica-se com um controlador da fornalha em outro nó. Juntos, dão forma cooperativa a um perfil de aplicação, que podemos denominar de *perfil de temperatura*. Dentro dos perfis, existe a necessidade de descrever uma subdivisão denominada de clusters ou agrupamento.

Os clusters estão dentro do espaço de um perfil em particular. As decisões obrigatórias são feitas combinando um identificador do conjunto de saída a um identificador do conjunto de entrada, supondo que ambas estão dentro do mesmo perfil.

Dentro dos clusters definimos os atributos, ou códigos binários que determinam uma ação específica. Cada informação que pode ser passada entre dispositivos, como uma chave de estado ou uma leitura de potenciômetro, é definida como um atributo. Estes atributos possuem uma nomeação individual, como uma identidade única. Estes atributos se agrupam em clusters (agrupamentos). Por exemplo: no mesmo termostato pode conter um atributo de saída “temperatura” que representasse a temperatura atual de uma fornalha. O controlador da

fornalha pode fazer medição deste atributo com uma entrada e controlar a temperatura da fornalha de acordo com o valor recebido do termostato.

Outro exemplo pode ser um controle de iluminação residencial, *perfil iluminação*; o cluster (agrupamento) é definido como *liga/desliga DRC* (controle da iluminação), que contém códigos binários com um atributo (liga/desliga), o qual devem ser representados com um valor de 8 bits, sendo que o valor *0xFF* significa “liga” e *0x00* significa “desliga” e o valor *0xF0* significa “saída pulsante (toggle output)” .

A Microchip Stack para o protocolo Zigbee, ferramenta de desenvolvimento para o protocolo em estudo, fornece suporte para o perfil via um arquivo cabeçalho (Bloco Funcional) com as seguintes informações:

- *Perfil ID*
- *IDs do dispositivo e versão*
- *IDs do cluster (agrupamento)*
- *IDs do atributo*
- *Tipos de dados do atributo.*

Cada bloco funcional do código, que suporta um ou mais agrupamentos, é chamado de endpoint. Diferentes dispositivos comunicam-se através de seus endpoints.

A Figura 31 ilustra dois nós, cada um possui uma única antena. Um nó possui dois interruptores e o outro possui quatro lâmpadas.

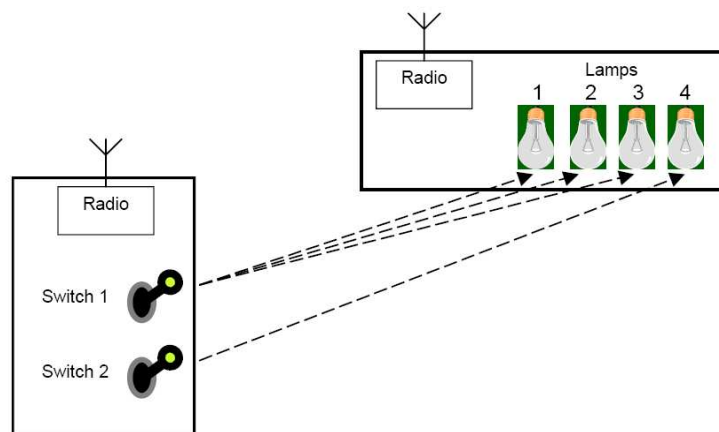


Figura 31 – Exemplo de aplicação de Zigbee com circuito composto de dois nós de acordo com fabricante Microchip

Os interruptores e as lâmpadas são subunidades que contêm uma descrição do dispositivo (perfil, cluster e atributo). Cada nó (interruptores e lâmpadas) possui um endereço na rede Zigbee.

No exemplo da Figura 31, o interruptor 1 controla as lâmpadas 1, 2 e 3, enquanto que o interruptor 2 controla a lâmpada 4.

O protocolo Zigbee fornece um nível de sub-endereçamento (sub-addressing), usado conjuntamente com os mecanismos da IEEE802.15.4. Um número de endpoint pode ser usado para identificar interruptores e lâmpadas individuais. No exemplo acima, o interruptor 1 poderia usar o endpoint 3, enquanto que o interruptor 2 poderia usar o endpoint 21. Similarmente, a lâmpada 4 poderia usar o endpoint 40, bem como as demais poderiam possuir seus próprios endpoints. Portanto, para ligar a lâmpada 4 (endpoint 40), o interruptor 2 (endpoint 21) deve enviar um atributo (comando para ligar = 0xFF) ao agrupamento do endpoint 40. Para desligar a lâmpada 4 (endpoint 40), o interruptor 2 (endpoint 21) deve enviar um atributo (comando para desligar = 0x00) ao agrupamento do endpoint 40.

Cada subunidade identificada em um nó (tal como os interruptores e as lâmpadas) é atribuída a seu próprio endpoint específico na escala de 1-240.

A figura 32 mostra graficamente como os vários dispositivos lógicos se relacionam uns com os outros. A figura mostra dois dispositivos de controle residencial, denominado perfil iluminação, cada um com um endpoint. A chave de controle de carga (endpoint 15, switch load controller) tem uma entrada no cluster (agrupamento) OnOffSRC com atributos ON e OFF. A chave do controle remoto (endpoint 7, switch remote control) tem uma saída do agrupamento OnOffSRC com atributos ON e OFF e uma entrada no agrupamento ProgramSRC com atributos Override, Auto e FactoryDefault em seu endpoint.

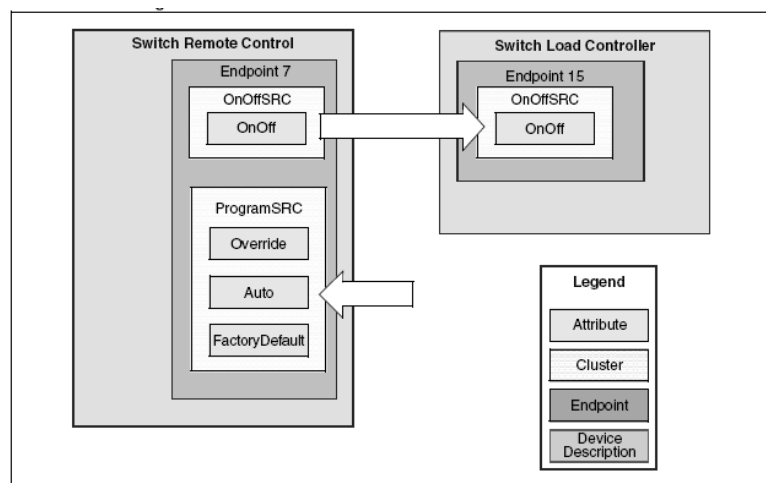


Figura 32 – Relacionamento entre os dispositivos lógicos

2.5.2 Formato do frame (Camada APL)

Cada nó em uma rede com protocolo Zigbee tem um Identificador Único Estendido - EUI-64 (IEEE) que define que cada dispositivo de comunicação que usa o protocolo Zigbee tem que ter um único endereço 64 bits MAC. Este endereço é definido com 24 bits de Indicação Única Organizada (OUI), adicionado de 40 bits nomeados pelo fabricante. Os OUIs podem ser comprados da IEEE para assegurar a singularidade global.

O tamanho máximo do pacote na IEEE 802.15.4 é de até 127 bytes, incluindo o valor do CRC de 16 bits . O valor do CRC verifica a integridade do frame. O mecanismo de confirmação CRC não faz indicação que o frame foi processado corretamente.

A IEEE 802.15.4 usa um mecanismo opcional para confirmar a transferência de dados (importante não confundir este mecanismo de confirmação com o CRC). Isso certifica que um pacote (frame) seja realmente entregue. Se o frame é transferido com uma solicitação de flag ACK e o reconhecedor não recebe dentro de certo período de tempo, o transmissor tentará novamente a transmissão antes de declarar um erro. Isto é um sinal muito importante para que o receptor indique que o frame foi recebido corretamente pela camada MAC.

Tanto o dispositivo que envia quanto o que recebe precisa estar atento ao formato do frame da transação. O formato do frame da subcamada APS é composta dos seguintes componentes básicos:

- Um cabeçalho do APS que compreende o controle do frame e informação de endereçamento
- Um payload do APS de comprimento variável, contendo informações específicas do tipo de frame.

Os frames da subcamada APS são descritos em uma seqüência de campos em ordem específica. Todos os formatos do frame nesta subclasse são descritos na ordem em que são transmitidos pela camada de NWK, ou seja, da esquerda para a direita, onde o bit menos significativo é transmitido primeiramente. Os bits dentro de cada campo são numerados de 0 (leftmost o bit menos significativo) a k-1 (rightmost o bit mais significativo), onde o comprimento do campo é k de bits. Os campos que são mais longos do que um byte são enviados para a camada de NWK na ordem do byte contendo os bits menos significativos ao byte contendo os bits mais significativos.

O formato do frame do APS é composto de um cabeçalho do APS e de um payload do APS. Os campos do cabeçalho do APS aparecem em uma ordem fixa. O frame geral do APS é formatado como ilustrado na Figura 33.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Figura 33 – Formato do Frame Geral do APS

O campo de controle do frame (Frame control) tem comprimento de 8 bits, contém a informação que define o tipo do frame, os endereços de campos e outros flags de controle. O campo de controle do frame é ilustrado na Figura 34.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Bits: 0-1	2-3	4	5	6	7
Frame type	Delivery mode	Indirect address mode ^a	Security	Ack. request	Reserved

Figura 34 – Formato do campo Frame de controle

O tipo de frame é definido por dois bits, possibilitando três tipos de frames, sendo o frame de dados, de comando e de reconhecimento, apresentado na Figura 35.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Bits: 0-1	2-3	4	5	6	7
Frame type	Delivery mode	Indirect address mode ^a	Security	Ack. request	Reserved

Frame type value b ₁ b ₀	Frame type name
00	Data
01	Command
10	Acknowledgement
11	Reserved

Figura 35 – Formato do campo Controle de Frame\Tipo de Frame

O modo de entrega (Delivery mode) é configurado com dois bits, como ilustrado na Figura 36.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Bits: 0-1	2-3	4	5	6	7
Frame type	Delivery mode	Indirect address mode ^a	Security	Ack. request	Reserved

Delivery mode value b ₃ b ₂	Delivery mode name
00	Normal unicast delivery
01	Indirect addressing
10	Broadcast
11	Reserved

Figura 36 – Formato do campo Controle de Frame\Modo de entrega

Se o valor no *Delivery Mode Value* configurado for 0b01, o endereçamento é indireto e o destino ou a fonte do endpoint serão omitidos, dependendo do valor do sub-campo modalidade de endereço indireto conforme Figura 37 (Indirect address mode). Se o valor no *Delivery Mode Value* for 0b10, a mensagem é uma transmissão para todos dispositivos da rede (Broadcast). Neste caso, a mensagem irá a todos os dispositivos e todos os endpoints (endpoint 255). Um valor 0b00 indica que o endereçamento é normal.

Bits: 0-1	2-3	4	5	6	7
Frame type	Delivery mode	Indirect address mode ^a	Security	Ack. request	Reserved

Figura 37 – Formato do campo Controle de Frame\Modo de Endereçamento Indireto

O modo de endereçamento indireto (Indirect address mode) tem um comprimento de um bit. Se este subcampo for fixado em 1, o destinatário do endpoint estará omitido no frame, indicando uma transmissão indireta ao coordenador de ZigBee. Se este subcampo for fixado em 0, a fonte do endpoint estará omitida no frame. Se o subcampo da modalidade de entrega (Delivery mode) do campo de controle do frame não indicar o endereçamento indireto, o subcampo de endereçamento indireto (Indirect address mode) estará ignorado.

O subcampo de pedido de reconhecimento (acknowledgement) possui um bit de comprimento e especifica, durante a transmissão, se o receptor necessita emitir um frame de reconhecimento (acknowledgement). A Figura 38 apresenta a configuração para tal fim.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Bits: 0-1	2-3	4	5	6	7
Frame type	Delivery mode	Indirect address mode ^a	Security	Ack. request	Reserved

Figura 38 – Formato do campo Controle de Frame\Modo de reconhecimento

Se este subcampo for fixado em 1, o receptor construirá e enviará um frame de reconhecimento para a origem após ter determinado que o frame recebido seja válido. Se este subcampo for fixado em 0, o receptor não enviará um frame de reconhecimento para a origem após ter determinado que o frame recebido seja válido.

O campo de destino do endpoint, apresentado na Figura 39, tem comprimento de 8 bits, especificando o endpoint do receptor. Até 240 objetos de aplicação (Application Objects) distintos podem ser definidos, cada um se conectará em um endpoint posicionado de 1 a 240.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Figura 39 – Formato do campo Destinatário do Endpoint

Dois endpoints adicionais são definidos para o uso do APSDE-SAP (entidade de dados):

- 1º) O endpoint 0 é reservado para a interface dos dados com o ZDO
- 2º) E o endpoint 255 é reservado para a interface de dados com a função de broadcast (transmissão) de dados a todos os objetos de aplicação.

Os endpoints 241-254 são reservados para o uso futuro.

O campo do identificador de agrupamento (Cluster Identifier) tem comprimento de 8 bits, especificando o identificador do cluster que deve ser usado na operação de ligamento (binding) no coordenador de ZigBee ou no dispositivo indicado pelo pedido (request). Tal campo é destinado para frames de dados, e não para um frame de comando. A Figura 40 ilustra o campo de identificação do cluster.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Figura 40 – Formato do campo Identificador de Agrupamento (Cluster)

O Profile identifier (PI), ilustrado na Figura 41, tem um comprimento de dois bytes, especificando o identificador de perfil de ZigBee para qual o frame é pretendido e será usado durante o processo de filtro de mensagens em cada dispositivo durante o processo de entrega do frame.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Figura 41 – Formato do campo Identificador de Perfil (Profile)

O campo da fonte do endpoint (Source Endpoint), ilustrado na Figura 42, tem um comprimento de 8 bits, especificando a origem do endpoint. A fonte do endpoint pode ter um valor 0x00, indicando que o frame originou-se no ZigBee (ZDO) de cada dispositivo. Se a

fonte do endpoint apresenta um valor entre 0x01 até 0xf0, indica que o frame originou-se em uma operação de aplicação. Todos endpoints restantes (0xf1-0xfe) são reservados.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Figura 42 – Formato do campo Fonte do Endpoint

O campo do frame payload tem um comprimento variável e contém informações específicas dos tipos individuais de frame. A divisão do frame (que não é o estudo deste trabalho) é ilustrada na Figura 43.

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
	Addressing fields				
APS header					APS payload

Figura 43 – Formato do campo Frame payload

São definidos três tipos de frame payload:

- 1º) Dados
- 2º) Comando
- 3º) Reconhecimento.

A ordem dos campos do frame de dados é idêntica à ordem do frame geral do APS, ou seja, da esquerda para a direita onde o bit menos significativo é transmitido primeiramente.

O campo do header do APS para um frame de dados (Figura 44) conterá o frame de controle, identificador de cluster, identificador de perfil e campos da fonte e destino do endpoint. O campo de destino do endpoint será incluído em um frame de dados, de acordo com valor do subcampo do modo de entrega (Delivery mode) do campo de controle do frame.

Octets: 1	0/1	1	2 ^a	0/ ^b 1	Variable
Frame control	Destination endpoint	Cluster identifier	Profile identifier	Source endpoint	Data payload
APS header					APS payload

Figura 44 – Formato do campo Frame payload\Dados

O frame do comando do APS será formatado como ilustrado em Figura 45.

Octets: 1	1	Variable
Frame control	APS command identifier	APS command payload
APS header	APS payload	

Figura 45 – Formato do campo Frame payload\Comando

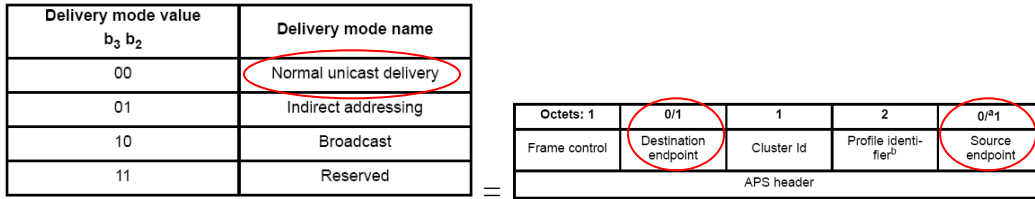
A ordem dos campos do frame de comando do APS é idêntica à ordem do frame geral do APS. O campo do identificador de comando do APS (APS command identifier) identifica o comando do APS que está sendo usado. O campo do payload do comando do APS (APS command payload) conterá o comando próprio do APS.

O frame de reconhecimento será formatado como ilustrado em Figura 46.

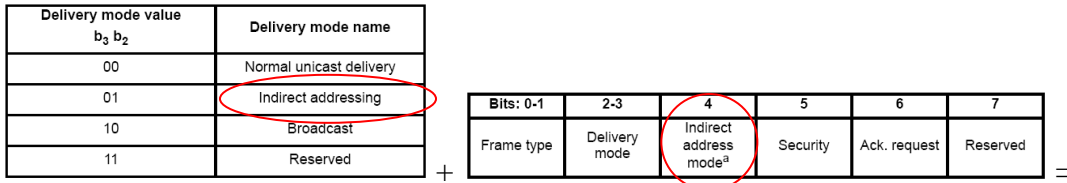
Octets: 1	0/1	1	2	0/ ^a 1
Frame control	Destination endpoint	Cluster Id	Profile identifier ^b	Source endpoint
APS header				

Figura 46 – Formato do campo Frame payload\Reconhecimento

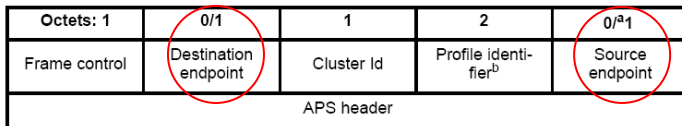
O campo do header do APS para um frame de reconhecimento conterá os campos do controle do frame, do identificador de cluster e do identificador de perfil. Se o modo de entrega (delivery mode) indicar endereçamento direto, os campos da fonte e do destino do endpoint estarão incluídos no frame de reconhecimento. Se a modalidade de entrega indicar endereçamento indireto os campos da fonte e do destino do endpoint estarão incluídos em um frame de reconhecimento de acordo com o valor do modo de endereçamento indireto do campo de controle do frame. A Figura 47(a) e (b) esclarecem as possíveis configurações do frame de reconhecimento.



(a)



Se...



(b)

Figura 47 – (a) Definição do Frame de reconhecimento\ Normal
(b) Definição do Frame de reconhecimento\ Indireto

2.6 CONCLUSÃO

O capítulo ilustra uma estrutura complexa e organizada do protocolo Zigbee. As camadas físicas PHY e de enlace MAC, são regulamentadas pela norma IEEE 802.15.4. As demais camadas (NWK e APL) são regulamentadas pela Zigbee Alliance.

No Brasil predomina a frequência portadora de 2,4 GHz, a qual origina uma taxa de transmissão de 250Kbits/s, apesar de existirem poucos casos de aplicações.

A topologia permite configurações simples, como a ponto a ponto, a configurações mais complexas como a topologia em rede. A topologia estrela é a mais difundida nas aplicações encontradas. Todas se utilizam de um número reduzido de dispositivos, composto em sua maioria por 1 coordenador da WPAN (FFDs), 1 roteador opcional e diversos fins de dispositivos (RFDs).

A formação da rede utilizando protocolo Zigbee pode ser sincronizada ou não sincronizada. A rede sincronizada utiliza um frame de sincronismo em intervalos de tempo

programados, para manter todos os dispositivos da rede sincronizados. Esta mensagem é denominada de Frame de beacon.

O protocolo demonstra possuir robustez para transmissão e recepção de dados. A modulação realizada pelo processo offset de quadrature phase-shift keying / O-QPSK, é projetado para resistir a interferências e ruídos. A mensagem de confirmação (ACK) de dado, permite garantir a confiabilidade da informação.

Os quatro serviços primitivos do protocolo foram desenvolvidos para simplificar a comunicação e reduzir o consumo de memória de programa. Diversos flags são configuráveis, permitindo que o usuário defina o tamanho do pacote (frame) a ser transmitido entre seus produtos, podendo chegar até 127 bytes.

Na camada de aplicação se encontra a estrutura de endereços para os diferentes dispositivos. Cada dispositivo é subendereçoado com um perfil, um cluster e um atributo. Todos são incluídos dentro de um sub-endereço de endpoint. Cada dispositivo possui um endereço composto de 64 bits, sendo 24 bits definidos como Indicação Única Organizada (OUI), adicionado de 40 bits nomeados pelo fabricante. Os OUIs podem ser comprados da IEEE para assegurar a singularidade global.

O protocolo Zigbee se apresenta como uma alternativa para a comunicação entre dispositivos eletrônicos que requerem robustez, baixo consumo e baixo custo de implantação. Precisa melhorar nos aspectos de complexidade (sua estrutura de camadas se apresenta de forma complexa), confiabilidade dos dados (as informações são perdidas) e redução do consumo de memória de programa (o coordenador pode chegar a 150k de uso da memória de programa).

3 DSC – CONTROLADOR DIGITAL DE SINAIS (DSPIC30F4011)

Neste capítulo é estudado o dispositivo dsPIC 30F4011 do fabricante Microchip. O capítulo objetiva analisar o mecanismo de processamento do dispositivo e analisar a compatibilidade com os processos de transmissão e recepção de dados através de uma rede sem fio utilizando o protocolo zigbee.

3.1 INTRODUÇÃO

O dsPIC 30F4011 possui um conjunto de instruções de um microcontrolador e suporte para funções de um DSP (*Digital Signal Processing*). O componente possui 40 pinos, sendo 30 deles configuráveis como entradas e/ou saídas (*Input e Output*). É classificado como um DSC.

O que é um DSC? Trata-se de um Controlador Digital de Sinais (DSC) consistindo em um dispositivo de 16 bits com arquitetura Harvard modificada, combinando as vantagens de um microcontrolador (MCU) de 16 bits com as de um Processador Digital de Sinais (DSP). Este controlador permite algumas aplicações, entre elas: controle de motores, conversão de potência, sensores de alta velocidade, processamento de sinais de áudio e voz, conectividade e modem, telecom, encriptação e aplicações automotivas, dentre outras.

De acordo com Oliveira e Andrade (2006):

Os controladores de sinais digitais (dsPICs) são divididos em famílias de aplicações. Existe uma família voltada para o uso geral (LCDs, controle de entradas e saídas digitais), outra relacionada ao controle de motores e uma terceira voltada à operação com sensores.

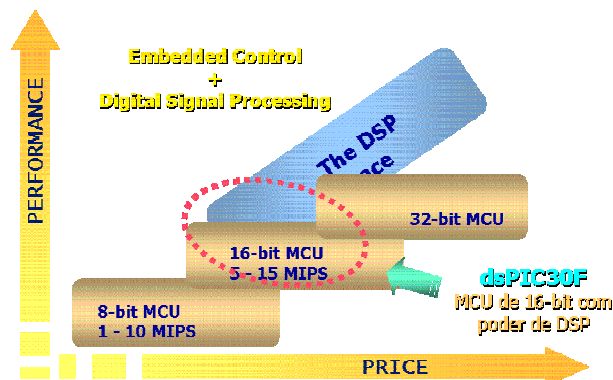


Figura 48 – Relação DSP x MCU x DSC

A Figura 48 ilustra a relação de desempenho e custo de microcontroladores de 8 bits, 16 bits, 32 bits, com DSCs e DSPs.

A Microchip é uma empresa situada nos Estados Unidos da América, considerada como uma das principais fabricantes de DSCs no mercado mundial. Entre seus produtos, a família 30Fxx apresenta uma série de derivações com diferentes características de construção, que permite diferentes capacidades de processamento (MIPS – Milhões de Instruções por Segundo), diversos tipos e quantidades de pinos bem como um número expressivo de periféricos, entre eles, os protocolos de comunicação CAN e SPI.

3.2 DSPIC 30F4011

A unidade central de processamento (CPU) do dsPIC 30F40xx é composta por um set de instruções, incluindo um amplo suporte para DSP. A CPU tem uma palavra de instrução de 24 bits. O contador de programa (PC) é de 24 bits e consegue endereçar até 4M x 24 bits no espaço de memória de programa do usuário.

Todas as instruções são executadas em um único ciclo de máquina, exceto aquelas que mudam o fluxo do programa, como as instruções de movimento de palavras duplas (MOV. D) e instruções de tabela.

O modelo dsPIC 30F4011 apresenta 40 pinos distribuídos entre portas (PORTs), alimentação e periféricos. A Figura 49 mostra a pinagem do dsPIC30F4011.

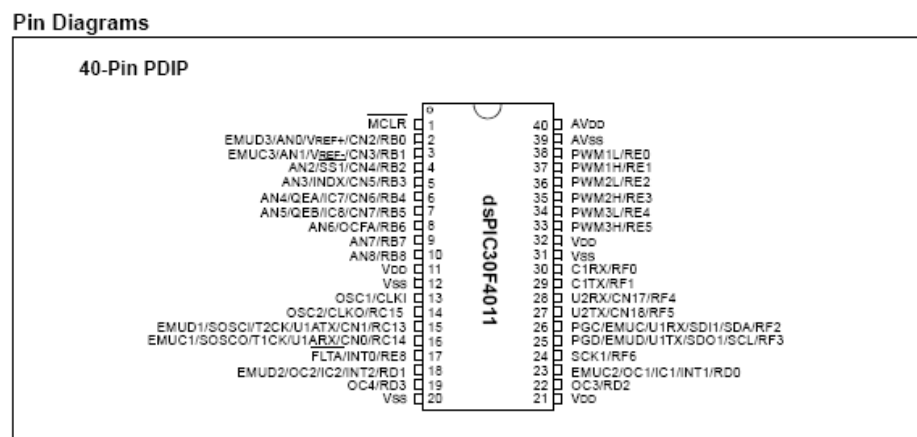


Figura 49 – Pinagem DSPIC30F4011

Os dsPICs possuem no modelo do programador 16 registradores Work (W) de 16 bits. Cada um desses registradores (W) pode ser utilizado como registrador de deslocamento de

endereço ou de dados. O décimo - sexto registrador (W15) funciona como um ponteiro de pilha de software para as interrupções e instruções de chamadas (CALL).

O set de instruções dos dispositivos do modelo dsPIC 30F40xx tem duas classes de instruções: uma para MCU (microcontrolador) e outra para DSP (Processador de Sinais Digitais). Estas duas classes de instruções são integradas de modo homogêneo na arquitetura .

O espaço dos dados pode ser endereçado com 32000 palavras ou 64000 bytes e é dividido em dois blocos: memória de dados (RAM) X e Y. Cada bloco de memória possui uma unidade de gerenciamento de endereços independente (AGU).

A classe de instruções do MCU opera somente pela AGU da memória X, enquanto que algumas instruções da classe de instruções do DSP operam pelas AGU's X e Y. A Figura 50 ilustra a divisão da memória RAM.

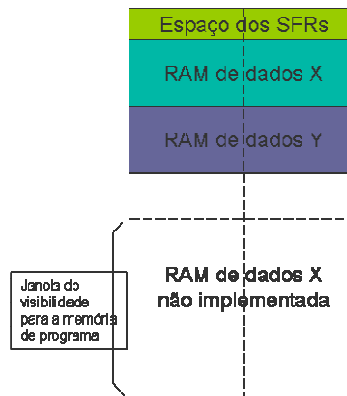


Figura 50 – Arquitetura da memória RAM

A CPU suporta os modos de endereçamento literal, direto à memória, direto ao registrador e indireto ao registrador. Cada instrução é associada a um grupo de endereçamento predefinido, dependendo das suas exigências fracionais. Para muitas instruções, o dsPIC 30F40xx é capaz de executar uma leitura da memória de dados, uma leitura do registrador Work, uma escrita na memória de dados e uma leitura da memória de programa (instrução) em um único ciclo. Como resultado, temos um suporte para instruções de três operadores, permitindo operações como $A+B=C$, executadas em um único ciclo.

A classe de instrução para o DSP caracteriza o dsPIC como um processador de alta velocidade utilizando um sistema de multiplicação de 17 x 17 bits, uma unidade lógica e aritmética (ALU) de 40 bits, dois acumuladores de 40 bits (ACC) e um registrador de deslocamento bidirecional para 40 bits. O registrador de deslocamento é capaz de deslocar um valor de 40 bits até 15 bits para a direita, ou até 16 bits para a esquerda em um único ciclo. As

instruções de DSP operam de modo homogêneo com todas as outras instruções e foram projetadas para aperfeiçoar o desempenho para operação em tempo real.

Podemos destacar no dsPIC 30F40xx como principais características:

- Arquitetura de Harvard modificada
- Compilador C com instruções otimizadas e arquitetura com modos de endereçamento flexíveis
- 83 instruções básicas
- 24 bits de instruções, 16 bits de dados
- 48 Kbyte de memória de programa Flash (16K instrução Word)
- 2 Kbyte de dados na memória RAM
- 1 Kbyte de dados na memória EEPROM
- Até 30 MIPS
 - DC para entrada de clock externa de 40 MHz
 - 4 MHz até 10 MHz para entrada de oscilador com PLL ativo (4x, 8x, 16x)
- 30 fontes de interrupção:
 - 3 fontes de interrupção externa
 - 8 níveis de prioridade usuário - selecionáveis para cada
 - Interrupção do arquivo fonte
- Registrador de 16 x 16 bits.

A classe de instrução que constitui o funcionamento do DSP no dsPIC 30F40xx, tem como principais características:

- Dois acumuladores de 40 bits com lógica opcional de saturação
- Multiplicação 17 x 17 bits
- Todas as instruções de DSP ocorrem em único ciclo
- ± 16 bits de turno de único ciclo.

De Oliveira (2006) afirma em sua obra que os DSCs apresentam “52 funções otimizadas, codificadas em C e prontas para realizar diversas operações relacionadas ao tratamento de sinais digitais, ganhando com isso maior agilidade e facilidade na construção de um firmware.”

A classe de instrução do microcontrolador apresenta uma quantificação de periféricos dentro de critérios estabelecidos, onde podemos destacar:

- Alta corrente nos pinos de I/O: 25 mA

- Módulo de tempo com Prescaler programável: cinco contadores de tempo de 16 bits
- 16 bits para a função de entrada de Captura
- 16 bits para a função de Comparadores PWM
- 3 linhas de módulos de SPI
- Suporta o protocolo I2C com 7/10 bits de endereçamento
- 2 módulos UART com FIFO Buffers
- 1 módulo CAN, versão 2.0B compliant.

O sistema de sincronismo dos dsPIC 30Fxx é determinado por meio de osciladores associados as modalidades de programação. No dsPIC 30F40xx as modalidades internas de programação do cristal são de até 40 MHz com 4x (PLL). As modalidades externas podem suportar até dois RC de até 4MHz. O sistema fornece uma escala completa de frequências de clock de disparo, sendo monitorado à prova de falhas do clock de disparo, permitindo a parada programada segura se o clock de disparo periférico parar.

Os modelos dsPIC apresentam uma família de componentes que podem variar de quantidade de pinos e periféricos a capacidade de memória de dados e de programa. A Figura 51 ilustrado em diagrama de blocos, a constituição dos pinos e periféricos do dsPIC 30F4011.

A Tabela 10 apresenta um comparativo das principais características técnicas entre alguns componentes da família de dsPICs do fabricante Microchip.

Device	Pins	Program Mem. Bytes/ Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	10-Bit A/D 1 Msp/s	Quad Enc	UART	SPI	I ² C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F4012	28	48K/18K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	-
dsPIC30F4011	40/44	48K/18K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	68K/22K	2048	1024	5	4	4	8 ch	18 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	18 ch	Yes	2	2	1	2

Tabela 10 - Tabela de comparação de DSPIC30F4011

3.3 DESENVOLVIMENTO DE PROTÓTIPO USANDO O DSPIC 30F4011

Para realizar o estudo e ensaio do protocolo Zigbee, foi desenvolvido um circuito eletrônico utilizando um processador de 16 bits de barramento. Os testes foram executados

utilizando o DSC modelo dsPIC 30F4011 da Microchip. O circuito eletrônico desenvolvido apresenta alguns periféricos acoplados como: LCD 4x20, botões, leds, entrada analógica e comunicação serial em protocolo CAN.

O desenvolvimento do algoritmo foi realizado em linguagem de programação C, necessitando de estudos e ensaios do compilador C30 (ferramenta de compilação para o código fonte do modelo dsPIC 30Fxx).

A proposta do circuito eletrônico é ilustrada nas Figuras 52, 53 e 54.

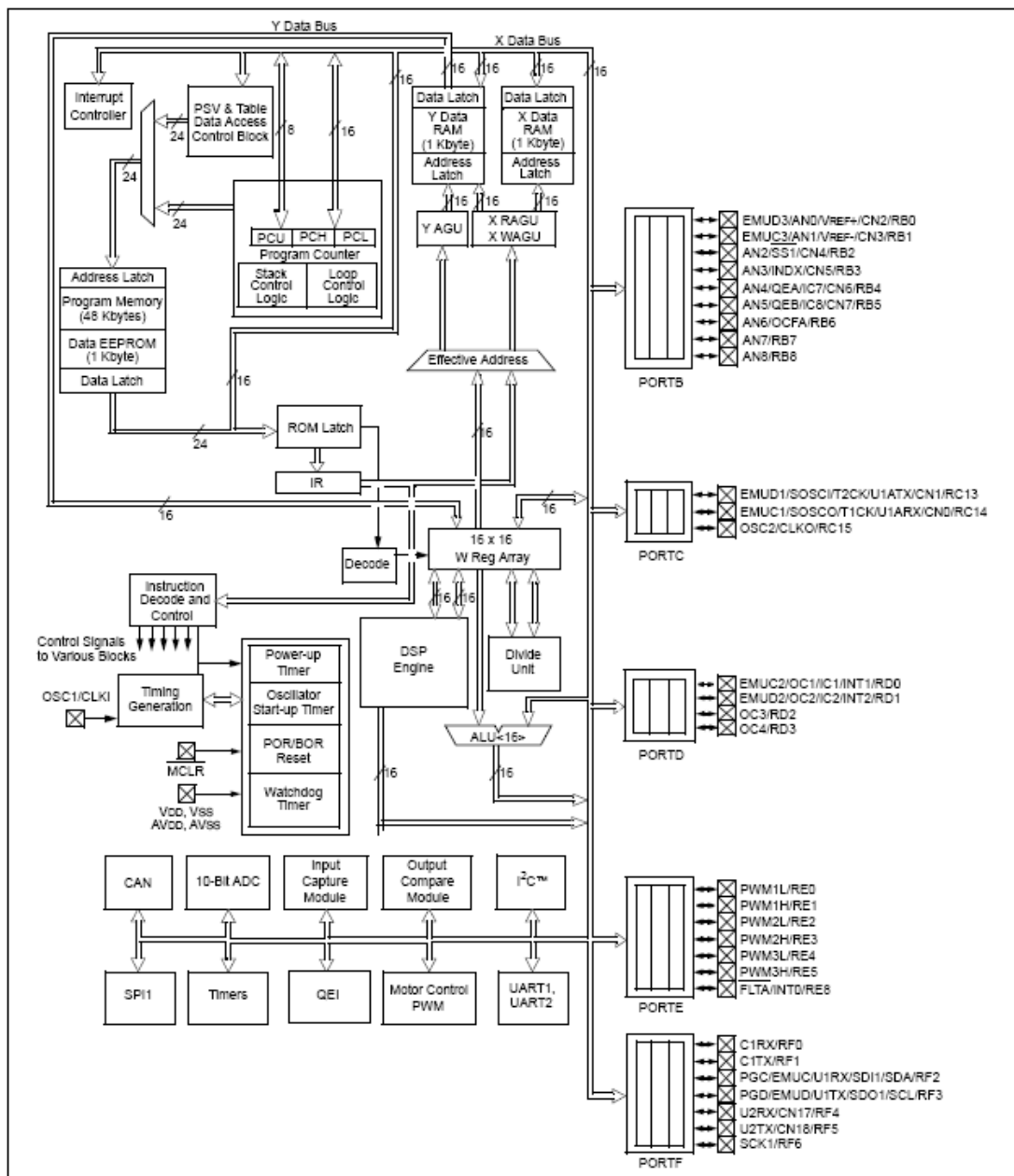


Figura 51 – Diagrama de Bloco do dsPIC30F4011 (40/44-PIN)

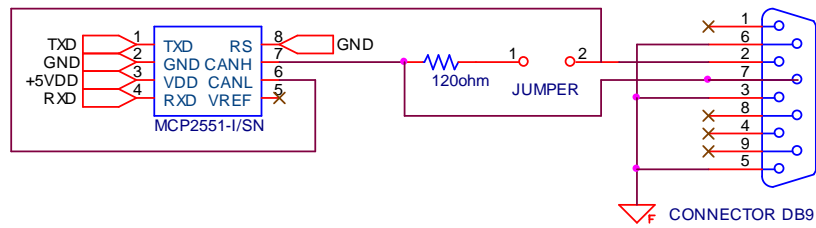


Figura 52 – Circuito eletrônico do protocolo CAN

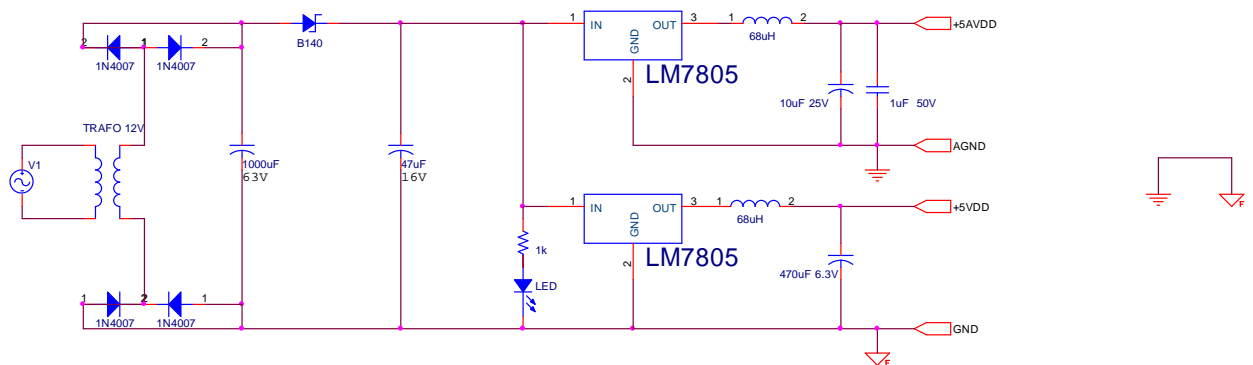


Figura 53 – Fonte de alimentação do processador e do sistema AD

O circuito desenvolvido apresenta um controlador de sinais digitais dsPIC30F4011 com gravação in-circuit com capacidade de 30MIPS e memória de programa de 48Kbytes, necessários para a implementação do protocolo Zigbee, que necessita no máximo de 30Kbytes para a função do coordenador. A forma de comunicação entre o transmissor/receptor de sinais (transiver) e o DSC, de acordo com a ferramenta de desenvolvimento Microchip Stack, é a comunicação serial SPI, presente no dsPIC 30F4011. A Figura 55 ilustra o protótipo desenvolvido.

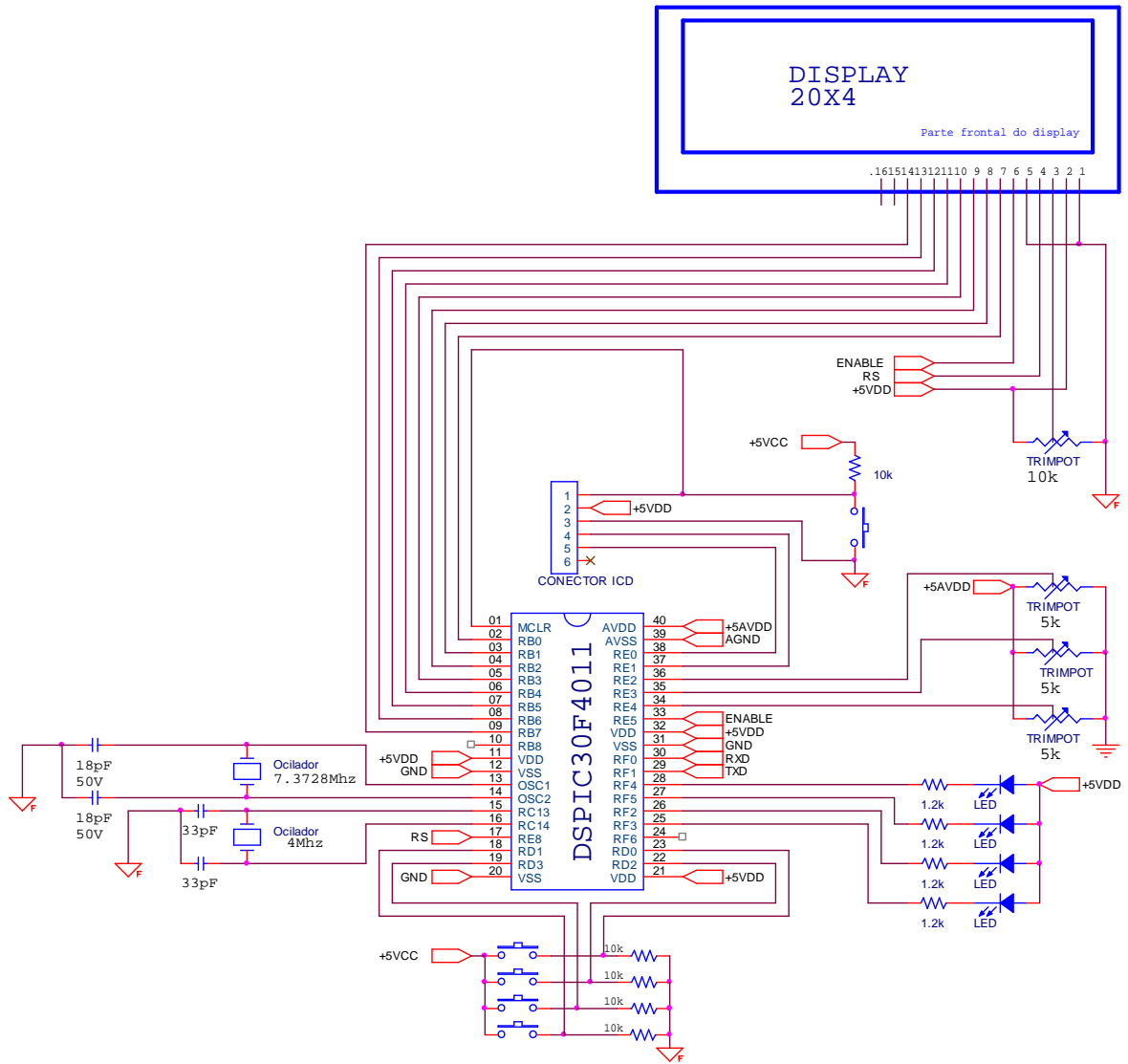


Figura 54 – Circuito eletrônico dsPIC 30F4011

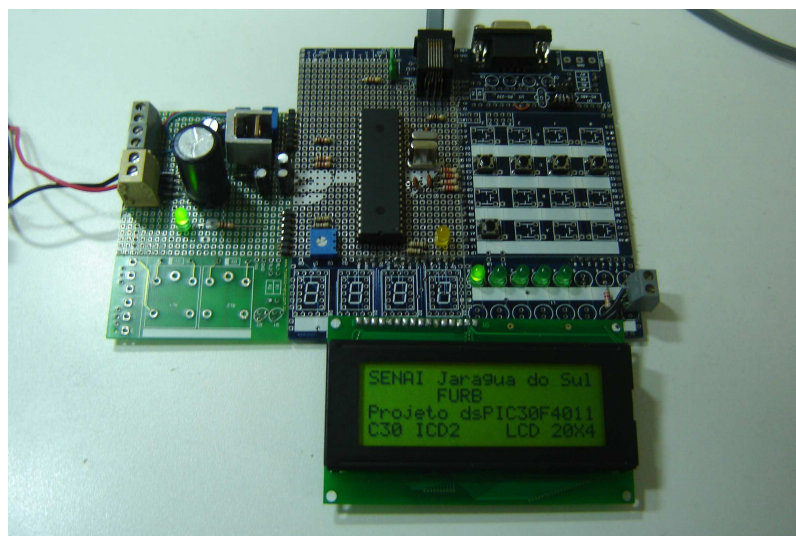


Figura 55 – Protótipo utilizando a plataforma em 16 bits usando o dsPIC 30F4011

3.4 CONCLUSÃO

O componente eletrônico dsPIC 30F4011 apresentou características compatíveis para o gerenciamento do protocolo de comunicação Zigbee, onde se destacam:

- Velocidade de processamento dos dados acima das necessidades mínimas para execução do protocolo Zigbee (30MIPS)
- Número e diversidade de periféricos acima do necessário para execução do protocolo Zigbee. Os únicos periféricos necessários são o UART e o protocolo SPI, ambos existentes no componente
- Possui o tipo de encapsulamento PDIP. Esta condição é necessária para os testes com o componente
- Memória de programa de 48kbytes, bem acima da capacidade mínima necessária de 30kbytes
- Multiplicador de hardware integrado, função não utilizada para o processamento da rede de comunicação Zigbee
- Maior custo para implementação do protocolo Zigbee devido aos custos do componente dsPIC 30F4011
- Complexidade para programação do código fonte. O fabricante Microchip não oferece suporte técnico para o protocolo Zigbee utilizando o componente dsPIC 30F4011.

Diante das características destacadas, conclui-se que uma rede de comunicação Zigbee pode ser gerenciada por processadores com capacidades de até 8 MIPS e memórias de programa de até 30kbytes. A plataforma de processamento necessária para o desenvolvimento do protocolo Zigbee, deve atender a três características essenciais:

- Característica 1: Deve possuir uma memória de programa igual ou superior a 30kbytes.
- Característica 2: Deve possuir um conjunto mínimo de periféricos composto do sistema UART e do protocolo SPI.
- Característica 3: Deve ter capacidade de processamento igual ou superior a 8MIPS.

Considerando o custo de implantação e as características técnicas do componente, o modelo dsPIC30F4011 é considerado inadequado para a implementação de uma rede de comunicação Zigbee. A plataforma a ser estudada deverá atender as características mínimas apresentadas neste capítulo. O componente a ser estudado é o PIC 18F4620 que é detalhado no capítulo 4 desta dissertação.

4 MICROCONTROLADOR PIC 18FL4620

Neste capítulo é estudado o microcontrolador PIC 18F4620 do fabricante Microchip. O estudo deste microcontrolador é um dos meios essenciais para o funcionamento de uma rede sem fio utilizando o protocolo Zigbee.

O capítulo descreve as principais características técnicas e de funcionamento do microcontrolador PIC 18F4620. Ilustra também o funcionamento da arquitetura do componente, bem como os principais registradores para programação de um microcontrolador.

4.1 INTRODUÇÃO

Os microcontroladores são dispositivos que evoluíram com a tecnologia dos circuitos integrados. Com o seu desenvolvimento, tornou possível armazenar centenas de milhares de transistores em uma única pastilha, permitindo a produção de microprocessadores e dos primeiros microcontroladores digitais, obtidos a partir de periféricos externos, tais como memória, entradas e saídas, temporizadores dentre outros.

Um microcontrolador difere de um microprocessador em um aspecto fundamental, sua funcionalidade. Um microcontrolador é a composição de um microprocessador (ULA – Unidade Lógica Aritmética) adicionado de memórias, barramentos e periféricos.

O PIC 18F4620 é um microcontrolador robusto e com características técnicas compatível para o processamento de uma rede de comunicação Zigbee.

4.2 PIC 18F4620

O PIC 18F4620 é um microcontrolador de 8 bits com 40 pinos, sendo 36 deles configuráveis como entradas e/ou saídas.

Os microcontroladores PIC 18F4620 apresentam três tipos de memória:

- Memória do programa
- Memória de dados RAM e
- Memória de Dados EEPROM.

A capacidade de multiplicar, dividir, subtrair e movimentar de uma posição da memória para outra é de responsabilidade da Unidade Central de Processamento (CPU) ilustrada na Figura 56. As posições de memória nela contidas chamam-se registros, descrito a seguir.

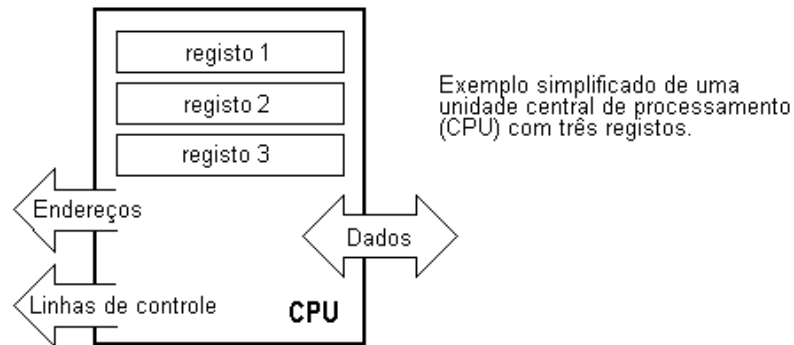


Figura 56 – Unidade Central de Processamento Simplificada

Os registros são posições de memória cujo papel é executar operações matemáticas (ou outras operações) envolvendo dados.

O barramento é responsável pela condução dos dados dentro do sistema. Existem dois tipos de barramento: barramento de dados e de endereço. As ligações entre a CPU e a memória é realizado pelos dois tipos de barramentos (bus), sendo caracterizado pela capacidade de condução de dados, podendo ser de 8, 16, 32 ou 64 bits. Para a família PIC 18FXX, a capacidade de condução de dados do barramento é de 8 bits. A Figura 57 ilustra o ligamento do bloco CPU com o bloco memória por meio de barramentos.

Para que os microcontroladores possam entrar em contato com o mundo exterior, eles necessitam de um conjunto de entradas e saídas digitais e de um conjunto de periféricos. As entradas e saídas digitais são sinais de tensão (5Vcc para a família PIC 18FXX) que podem ser lidas ou escritas, ou seja, pode-se receber um sinal digital de tensão em um pino físico, ou enviar um sinal de tensão através de um pino físico.

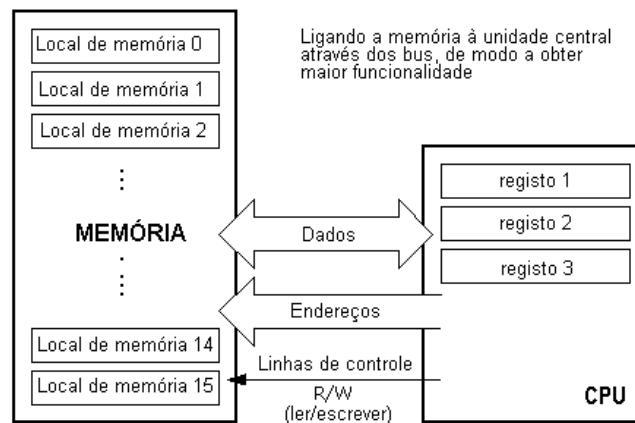


Figura 57 – Barramento interligando CPU e Memória

Os periféricos (no microcontrolador) são acessórios disponibilizados pelo fabricante e destinados a realizar funções que podem variar de conversão A/D até uma comunicação serial.

No microcontrolador PIC da família 18Fxx (Microchip), o fabricante disponibiliza os seguintes periféricos:

- Dissipador para altas correntes (High-current) que podem chegar a 25 mA por pino
- 4 fontes de interrupções de entrada
- 3 fontes de interrupções externos programáveis
- 2 módulos de captação (Compare/PWM (CCP))
- 4 saídas de PWM - Polaridade selecionável
- Módulo de sincronismo mestre do (MSSP), que suporta três linhas SPI™ e de mestre\escravo de I2C™
- Módulo endereçável de USART (Sustentações de interface RS485, RS232 e LIN 1.2.)
- Arquitetura otimizada para o compilador C
- Retenção dos dados EEPROM (100 anos de retenção)
- 8 x 8 e 16 x 16 bits multiplicador por hardware.

Existem diferenças técnicas entre microcontroladores da mesma família. As características técnicas dos diferentes microcontroladores da família PIC 18Fxx são apresentados na Tabela 11.

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		EUSART	Comp.	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI™	Master I ² C™			
PIC18F2525	48K	24576	3986	1024	25	10	2/0	Y	Y	1	2	1/3
PIC18F2620	64K	32768	3986	1024	25	10	2/0	Y	Y	1	2	1/3
PIC18F4525	48K	24576	3986	1024	36	13	1/1	Y	Y	1	2	1/3
PIC18F4620	64K	32768	3986	1024	36	13	1/1	Y	Y	1	2	1/3

Tabela 11 – Tabela de comparação da família PIC 18Fxx

Dependendo do modelo de microcontrolador, um pino pode ter uma função ou até quatro funções configuráveis. Por exemplo, o microcontrolador PIC 18F4620, ilustrado na Figura 58, possui 40 pinos distribuídos em três funções básicas:

- Pinos de entradas e saídas digitais que são denominadas de PORTs (PORTA, PORTB, PORTC, PORTD e PORTE)
- Pinos de alimentação e
- Pinos destinados a periféricos.

40-Pin PDIP

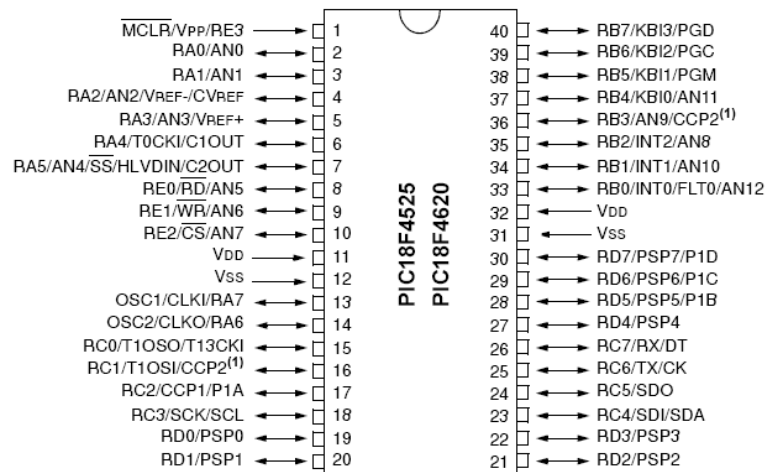


Figura 58 – Pinagem do PIC 18F4620

Dependendo do dispositivo selecionado e das características permitidas, há até cinco PORTs disponíveis. Alguns pinos dos PORTs de entradas ou saídas (I/O) são multiplexados por meio de um periférico do dispositivo. Em geral, quando um periférico for usado, o pino correspondente ao periférico não pode ser usado como um pino de entrada ou saída de uso geral.

Cada PORT tem três registradores especiais para sua operação, sendo eles:

- Registrador TRIS (define o sentido dos dados)
- Registrador PORT (lê os níveis nos pinos do dispositivo) e
- Registrador LAT (mantém o estado lógico alto no pino).

A Figura 59 ilustra o diagrama de bloco do microcontrolador PIC 18F4620, detalhando todos os pinos dos diferentes PORTs.

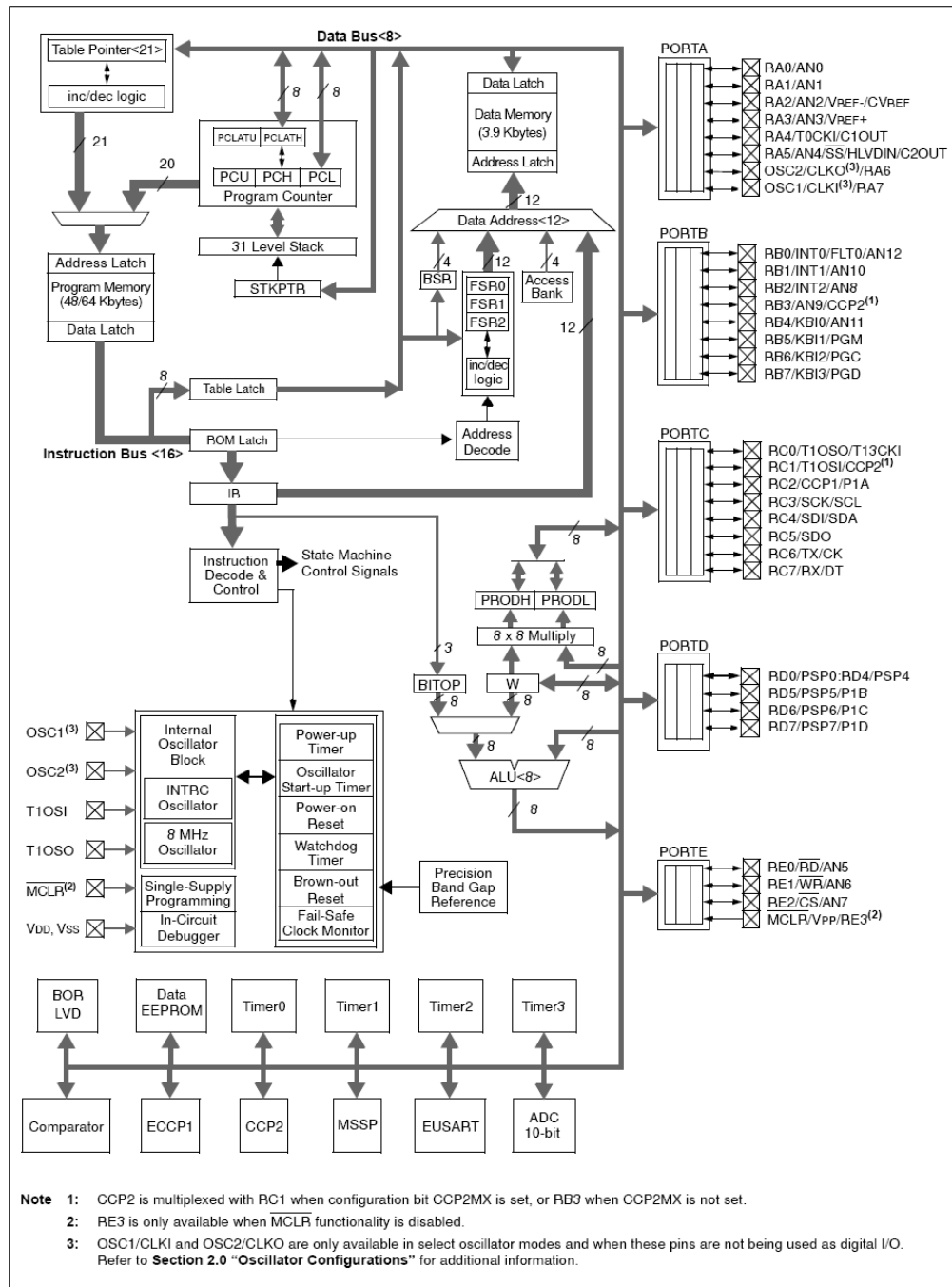


Figura 59 – Diagrama de Bloco do PIC 18F4620

Os microcontroladores da família PIC18FXX utilizam a tecnologia denominada baixa voltagem (low-voltage), possuindo memória do tipo Flash realçada (designada com a letra “F” no código do dispositivo tal como PIC18F2620) que podem operar em níveis de tensão entre 4.2V a 5.5V. Alguns dispositivos apresentam a característica de baixa voltagem designadas pelas letras “LF” estampadas no código do dispositivo (como PIC18LF4620), que podem operar em uma faixa de tensão entre 2.0V a 5.5V. Os dispositivos com a denominação “LF” são apropriados para equipamentos que usam baterias como fonte de alimentação. Para dispositivos eletrônicos designados a função de comunicação sem fio, como o protocolo Zigbee que utilizam baterias como fonte de alimentação, a característica de baixa voltagem é essencial.

Para um microcontrolador operar de forma simétrica e intermitente, é necessário a inclusão de um sistema de sincronismo. Os osciladores, assim como são designados, são componentes eletrônicos responsáveis pela sincronização e pelo movimento ordenado do microcontrolador. A Figura 60 ilustra o esquema de ligação de um oscilador de cristal (XTAL) nos pinos de um microcontrolador (OSC1 e OSC2).

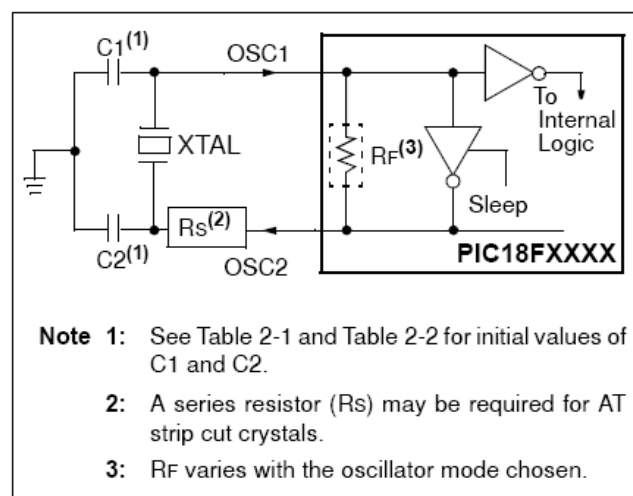


Figura 60 – Cristal ressonador XTAL

Estes osciladores operam em diferentes modalidades. O microcontrolador PIC18F4620 opera em 10 modalidades diferentes. O usuário pode programar os bits da configuração, através dos bits FOSC3: FOSC0, no registrador especial OSCCON, permitindo as seguintes configuração:

1. Cristal LP de baixo consumo (Low-Power)
2. Cristal/ressonador de XT

3. Cristal de alta velocidade/ressonador de HS
4. Cristal de alta velocidade/ressonador de HSPLL com o PLL permitido
5. Resistor externo/capacitor de RC com FOSC/4 saída em RA6
6. Resistor externo/capacitor de RCIO com entrada e saída em RA6
7. Oscilador INTIO1 interno com saída FOSC/4 em RA6 e entrada e saída em RA7
8. Oscilador INTIO2 interno com entrada e saída em RA6 e RA7
9. Pulso de disparo externo do EC com saída FOSC/4
10. Pulso de disparo externo de ECIO com I/O em RA6.

A Figura 61 ilustra as 10 modalidades de funcionamento dos osciladores (clock) do PIC18F4620 que são definidas pela configuração do Postscaler.

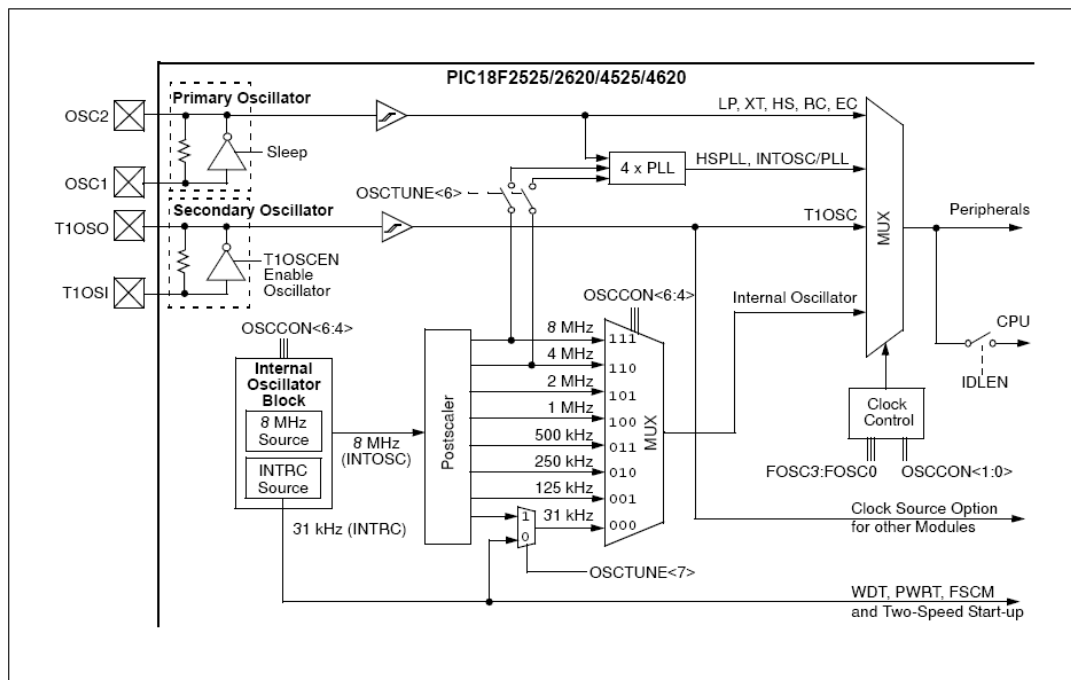


Figura 61 – Diagrama de funcionamento do oscilador do PIC 18F4620

O registrador especial OSCCON controla diversos aspectos da operação do oscilador, que é operado sob diferentes modalidades. Detalhes do registrador especial OSCCON para controle do oscilador são apresentados no Anexo 1.

Os microcontroladores, em situações específicas, permitem a execução de uma operação de restauração denominada de RESET. Os dispositivos PIC18F4620 possuem vários tipos de restauração, podendo ser:

- a) Restauração depois de energizado [Power on Reset (POR)]
- b) MCLR (Master clear) restaura durante a operação normal

A memória é a parte fundamental do microcontrolador. Nela é armazenada diversas informações.

Há três tipos de memória no PIC18FXX, descritos a seguir:

- 1) Memória do programa
- 2) Memória de dados RAM e
- 3) Memória de dados EEPROM.

Os microcontroladores da Microchip possuem uma arquitetura Harvard, ou seja, existe um barramento para a memória de dados e outro para a memória de programa. Isto permite um acesso simultâneo aos dois espaços de memória, aumentando a velocidade de processamento.

A memória EEPROM é utilizada para armazenar dados não voláteis e é considerado um dispositivo periférico gerenciado por um conjunto (SET) de registradores de controle.

A memória Flash do PIC18F4620 tem 64 Kbytes de espaço, podendo armazenar até 32.768 instruções (single-word). A memória Flash apresenta dois endereços (denominados vetores de interrupção), sendo o primeiro para alta prioridade (0008h) e outra para baixa prioridade (0018h). A restauração (RESET) do microcontrolador está no endereço 0000h e os vetores de interrupções estão nos endereços 0008h e em 0018h da memória de programa. Os mapas de memória do programa são ilustrados na Figura 63.

A memória de dados do PIC18F4620 é do tipo RAM, ou seja, memória volátil. Cada registrador tem um endereço de 12 bits, e é armazenado na memória de dados (reservando até 4096 bytes). O espaço da memória é dividido em 16 bancos de dados, sendo cada banco de dado composto por 256 bytes. A Figura 64 ilustra a organização da memória de dados para o PIC18F4620.

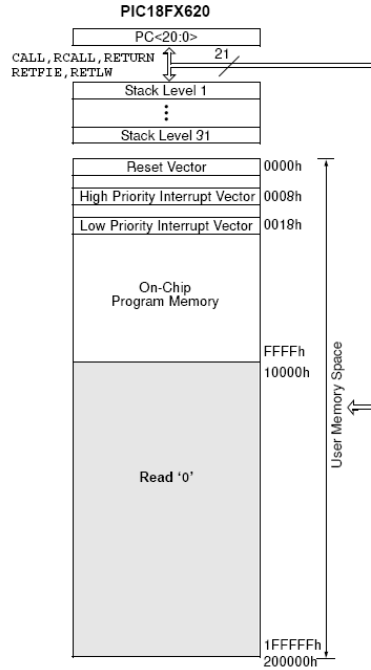


Figura 63 – Mapa de memória de programa (Flash) do PIC 18Fxx

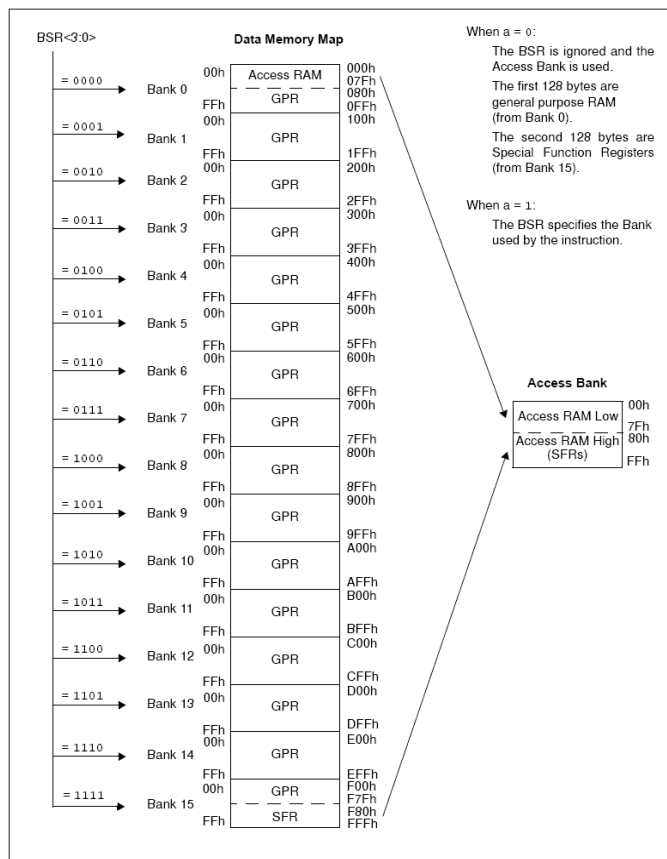


Figura 64 – Mapa de memória de dados (RAM) do PIC 18Fxx

Os registradores especiais (SFRs) são designados para configurar as características de hardware e de software dos microcontroladores, ou seja, são usados pelo processador central e pelos módulos periféricos controlando a operação desejada do dispositivo. Estes registradores estão armazenados na memória RAM, cujo endereço começa ao final da memória dos dados (FFFh) e estende-se até a metade superior do banco 15 (F80h a FFFh). Uma lista destes registradores é ilustrada na Figura 65.

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCCh	CCPR2H	F9Ch	__ ⁽²⁾
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Bh	OSCTUNE
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	__ ⁽²⁾
FF9h	PCL	FD9h	FSR2L	FB9h	__ ⁽²⁾	F99h	__ ⁽²⁾
FF8h	TBLPTRU	FD8h	STATUS	FB8h	BAUDCON	F98h	__ ⁽²⁾
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	PWM1CON ⁽³⁾	F97h	__ ⁽²⁾
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS ⁽³⁾	F96h	TRISE ⁽³⁾
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD ⁽³⁾
FF4h	PRODH	FD4h	__ ⁽²⁾	FB4h	CMCON	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	__ ⁽²⁾
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	__ ⁽²⁾
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG	F8Fh	__ ⁽²⁾
FEEh	POSTINC0 ⁽¹⁾	FCEh	TMR1L	FAEh	RCREG	F8Eh	__ ⁽²⁾
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXREG	F8Dh	LATE ⁽³⁾
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD ⁽³⁾
FEBh	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	EEADRH	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	__ ⁽²⁾
FE7h	INDF1 ⁽¹⁾	FC7h	SSPSTAT	FA7h	EECON2 ⁽¹⁾	F87h	__ ⁽²⁾
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSPCON1	FA6h	EECON1	F86h	__ ⁽²⁾
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSPCON2	FA5h	__ ⁽²⁾	F85h	__ ⁽²⁾
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	__ ⁽²⁾	F84h	PORTE ⁽³⁾
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	__ ⁽²⁾	F83h	PORTD ⁽³⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA

- Note** 1: This is not a physical register.
 2: Unimplemented registers are read as '0'.
 3: This register is not available on 28-pin devices.

Figura 65 – Mapa de registradores do PIC 18F4620

Todos os dispositivos PIC18Fxx possuem um multiplicador por hardware 8 x 8 bits como parte do ULA (Unidade Lógica Aritmética). O multiplicador executa uma operação matemática de multiplicação em maior velocidade armazenando o resultado em dois registradores especiais, PRODH e PORDL de 16 bits cada.

A operação do multiplicador não afeta nenhuma bandeira de aviso (flag) no registrador especial STATUS¹. Fazer a multiplicação por hardware, permite que o cálculo seja terminado em um único ciclo de instrução, trazendo vantagens computacionais como redução do tamanho do código. Esta capacidade permite que os dispositivos PIC18xx sejam utilizados em aplicações reservadas, antes exclusivamente, para processadores de sinais digitais (DSPs). A Figura 66 ilustra um exemplo de código em linguagem assembler para multiplicação.

```

MOVWF  ARG1, W      ;
MULWF  ARG2          ; ARG1 * ARG2 ->
                          ; PRODH:PRODL

```

Figura 66 – Exemplo de Algoritmo de multiplicação 8x8bits

O PIC 18FXX permite também a multiplicação de 16 x 16 bits, gerando um resultado de 32 bits. Exemplo no Anexo 3.

4.3 FONTES DE INTERRUPTÃO DO MICROCONTROLADOR PIC 18F4620

De acordo com Pereira (2002), “interrupção é um evento externo ao programa que provoca a parada da sua execução, a verificação e tratamento do referido evento e em seguida o retorno do programa ao ponto em que havia sido interrompido”.

Ainda segundo Pereira (2002), “quando ocorre um evento de interrupção, a instrução em execução é completada, o conteúdo do contador do programa (ou equivalente) é armazenado na pilha e o programa é desviado para um endereço conhecido como vetor de interrupção”.

¹ Status é um registrador especial responsável pela configuração dos bancos da memória RAM.

Os dispositivos PIC18F4620 têm múltiplas formas de gerar (denominada de fontes) interrupção. Também permitem priorizar o tipo de fonte de interrupção e nível de prioridade alto ou nível baixo (high ou low) de interrupção.

O vetor de alta prioridade (HIGH) da interrupção está no endereço 0008h e o vetor de baixa prioridade (LOW) está em 0018h.

Há dez registradores que são usados para controlar a operação de interrupção. Estes registradores são:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2.

Os bits de configuração (denominadas de chaves das fontes de interrupções) podem ser visualizadas no Anexo 4.

4.4 SISTEMA DE COMUNICAÇÃO SERIAL

Uma das características existentes nos microcontroladores é a presença de protocolo que permitem a comunicação de dados de forma serial.

O estudo de alguns protocolo de comunicação é necessário para o entendimento do protocolo Zigbee, que utiliza o protocolo SPI para comunicação serial do microcontrolador com o transmissor de RF.

O microcontrolador PIC 18Fxx possui um módulo de sincronismo mestre (MSSP), designado para comunicação serial com outros microcontroladores ou componentes eletrônicos. O módulo MSSP pode operar em duas modalidades, sendo:

- 1) Relação periférica de série (SPI)
- 2) Inter-Integrado, circuito (I2C).

A tecnologia de comunicação SPI (Serial Peripheral Interface) foi desenvolvida pela Motorola para a linha de processadores da família de microcontroladores MC68K. O SPI é um protocolo síncrono que opera no modo duplo canal (full duplex) e é composto por 4 sinais. O protocolo SPI não permite o endereçamento; a comunicação só pode ser feita entre

dois pontos, sendo um deles o ponto mestre (Master) e outro o escravo (Slave), apesar de existir técnicas para comunicação entre diversos dispositivos.

A modalidade de Relação periférica de série, permite que 8 bits de dados sincronizados, sejam transmitidos e recebidos simultaneamente. Para realizar uma comunicação, são utilizados três pinos do microcontrolador:

- 1) Pino de Dados de saída SDO (RC5/SDO)
- 2) Pino de Dados de entrada SDI (RC4/SDI/DAS)
- 3) Pino para pulso do oscilador de sincronismo SCK (RC3/SCK/SCL).

Adicionando um quarto pino, podemos operar com endereçamento:

- Pino para seleção do Escravo SS (RA5/AN4/SS/HLVDIN/C2OUT).

Com este quarto pino é possível montar uma rede de comunicação contendo um componente mestre e vários escravos, desde que o mestre controle individualmente os pinos/SS de cada um dos escravos. Esse controle terá de ser implementado manualmente no software e deve garantir que somente um escravo seja ativado no mesmo intervalo de tempo, evitando conflitos nas saídas SDO.

O microcontrolador PIC 18Fxx mestre é responsável pela geração do sinal sincronizador (clock). Seu pino SCK deve ser configurado pelo registrador PORTC como saída. Já o pino SCK do escravo deve ser configurado como entrada. O registrador especial SSBUF é responsável em armazenar o dado a ser transmitido e recebido. A troca de dados é síncrona, ou seja, enquanto um pino está mandando dados, o outro está recebendo dados. Toda vez que são transmitidas informações são gerados 8 pulsos de sincronismo na saída do SCK.

O registrador especial SSPCON é usado para configurar o oscilador nos bits SSPM3 ao SSPM0 e também para ajustar o modo de operação, ou seja, se o componente é configurado na função mestre ou escravo.

O registrador especial SSPSTAT ajusta a condição de recebimento. O sistema pode efetuar a leitura da porta SDI em dois pontos distintos, conforme o valor imposto em SMP.

Na frequência máxima de 20MHz do oscilador do microcontrolador, a taxa de transferência do protocolo SPI pode chegar a 5,0Mbps.

A Figura 67 ilustra as características de instalação física dos pinos do microcontrolador para o protocolo SPI.

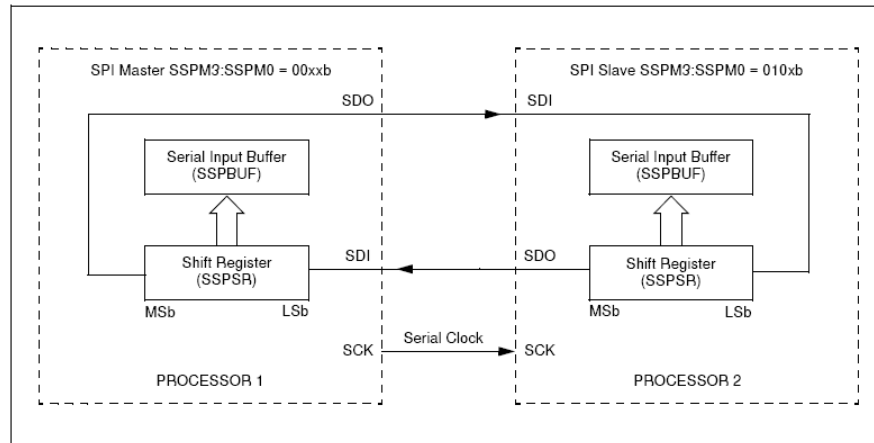


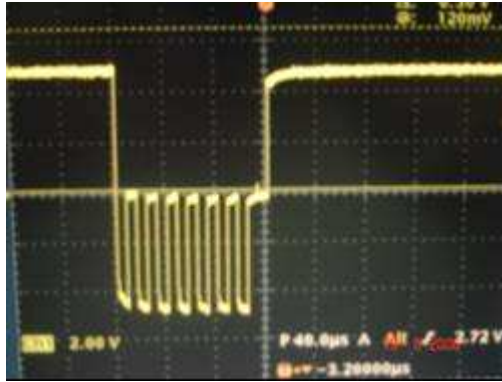
Figura 67 – Conexão mestre x escravo SPI

Em laboratório, foram realizados testes utilizando microcontroladores da família PIC 16F8xx da Microchip, com o protocolo SPI. Os testes apresentaram rapidez e confiabilidade na comunicação serial entre dois microcontroladores da família PIC 16F877 (o componente 16F877 foi utilizado neste experimento devido a simplicidade do componente em relação ao 18Fxx). As Figuras 68(a), 68(b), 69(a), 70(a) e 70(b) ilustram o funcionamento do protótipo desenvolvido para testes do protocolo SPI. Um componente 16F877 transmite uma variável e outro recebe a mesma variável; ambos apresentam a variável no LCD.

Transmissão da variável 1



(a)



(b)

Figura 68 – Transmissão da variável número 1

(a) Circuito eletrônico para transmissão e recepção de dados com protocolo SPI

(b) Pacote transmitido do SPI (Frame) – transmissão da variável 1 (o pino de referência está ligado no oscilador)

Transmissão da variável 2

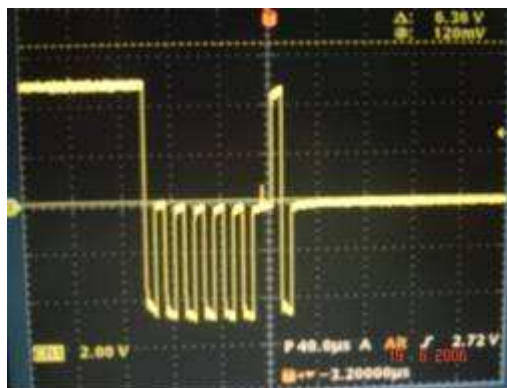
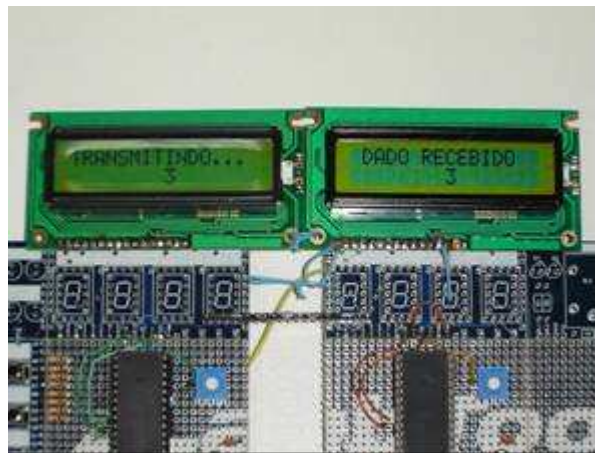


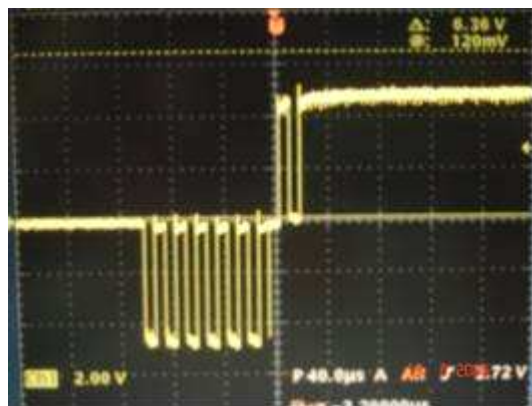
Figura 69 – Transmissão da variável número 2

(a) Frame do SPI – transmissão da variável 2 (o pino de referência está ligado no oscilador)

Transmissão da variável 3



(a)



(b)

Figura 70 – Transmissão da variável número 3

(a) Circuito eletrônico para transmissão e recepção de dados com protocolo SPI

(b) Frame do SPI – transmissão da variável 3 (o pino de referência está ligado no oscilador)

4.5 CONCLUSÃO

O componente eletrônico PIC 18F4620 apresentou desempenho satisfatório para execução do protocolo Zigbee. Sua capacidade de memória de programa é suficiente para armazenar os códigos dos dispositivos RFDs e FFDs. O componente inclui a tecnologia nanowatt, operando em nível de tensão de 3.3V, uma característica necessária do protocolo

Zigbee . O compilador C18, ferramenta de depuração do código, é consolidada no mercado e permite comunicar-se com a ferramenta Microchip Stack.

Seu processamento apresenta capacidades de até 8 MIPS e memórias de programa de até 40 kB, necessárias para armazenar o código de programa do protocolo Zigbee.

O componente está em fabricação, apresenta 40 pinos com diferentes periféricos, entre eles destacamos o periférico USART e o protocolo SPI, ambos necessários para o uso da ferramenta Microchip Stack.

O capítulo pode apresentar algumas características da plataforma PIC 18F4620 para suportar o protocolo de comunicação Zigbee.

5 FERRAMENTAS DE DESENVOLVIMENTO

Neste capítulo são estudadas as ferramentas de desenvolvimento para uma rede sem-fio utilizando o protocolo Zigbee.

5.1 INTRODUÇÃO

Entre as ferramentas, o Microchip Stack, escrita em linguagem de programação C, é projetado para operar os microcontroladores Microchip's da família PIC18Fxx, permitindo a configuração de parâmetros e a definição dos endereços da camada de enlace e de aplicação (MAC e APL). O Analisador ZENA é uma ferramenta que permite configurar e analisar redes sem-fio que utilizam o protocolo Zigbee, sendo composto de software e hardware com a finalidade de programar e monitorar o protocolo Zigbee e MiWi. Ainda neste capítulo, é apresentada a ferramenta PICDEM™ Z, que é uma placa de desenvolvimento que utiliza o protocolo Zigbee em microcontroladores da família Microchip PIC18Fxx.

Estas ferramentas permitem que experimentos práticos sejam executados para comprovação e validação dos conteúdos apresentados pelas normas IEEE 802.15.4, norma Zigbee Alliance e o procedimento de funcionamento do protocolo SPI.

5.2 ARQUITETURA DO MICROCHIP STACK - INTRODUÇÃO

O Microchip Stack é uma ferramenta escrita em linguagem C, projetada para operar com os microcontroladores Microchip's da família PIC18Fxx. Esta ferramenta fornece um conjunto de arquivos (código fonte modelo em linguagem C) para as diferentes camadas do protocolo de comunicação Zigbee. Ela utiliza a memória de programa flash para armazenar seus parâmetros, incluindo os endereços da camada MAC.

As principais características são:

- Utiliza a versão 1.0 das especificações do protocolo ZigBee
- Usa a faixa de frequência de portadora de 2.4 GHz
- Possui suporte para todos os tipos de dispositivos do protocolo ZigBee (coordenadores, roteadores e fins de curso)

- É portátil para os microcontroladores da família PIC18Fxx
- Possui suporte para o compilador Microchip MPLAB C18
- Os nomes dos códigos-fonte (em linguagem C) estão alinhados com o protocolo ZigBee e com as especificações da IEEE 802.15.4.

As versões 1.0 até 3.5 da ferramenta Microchip Stack possuem limitações, não garantindo a segurança dos dados durante a transmissão e recepção (criptografia de dados). A ferramenta não fornece apoio para algumas funções, dentre as quais podem ser mencionadas:

- O suporte técnico não assessora a formação de redes
- Não resolve conflito de endereços
- O conserto de rota automático não é executado.

A ferramenta Microchip Stack foi projetada para seguir as especificações técnicas da norma IEEE 802.15.4 (protocolo Zigbee). A Figura 71 ilustra a arquitetura dessa ferramenta com base no protocolo Zigbee, onde as camadas físicas, de enlace, de rede e de aplicação são destacadas. A ferramenta é projetada para ser executada em uma plataforma de hardware denominada *PICDEM™ Z Demonstration Board*, porém, pode ser adaptada facilmente para qualquer hardware equipado com um microcontrolador compatível.

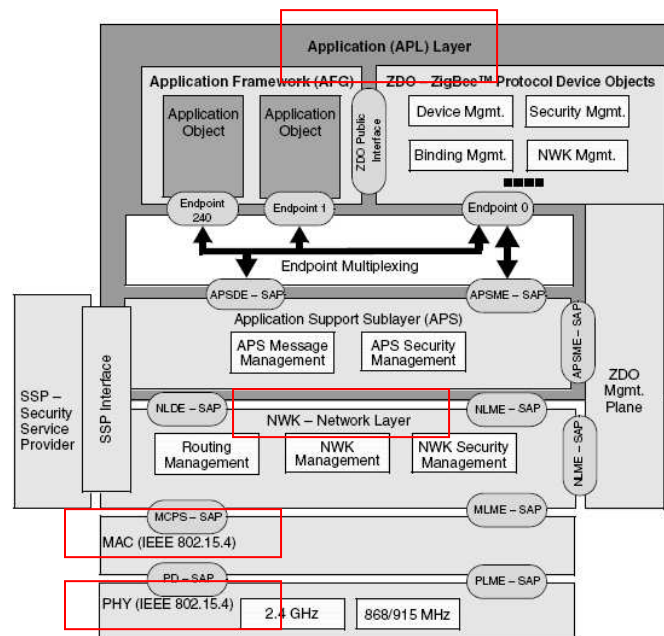


Figura 71 – Arquitetura Microchip Stack do protocolo Zigbee

A ferramenta Microchip Stack pode ser baixada sem custo de licença na página do fabricante da Microchip (www.microchip.com).

5.3 DETALHAMENTO DO MICROCHIP STACK

Para instalar os arquivos da ferramenta Microchip Stack, é necessário baixar os arquivos (download) desde o sítio do fabricante (Microchip). Os arquivos são distribuídos unicamente para sua instalação no sistema operacional Windows®. A instalação dos arquivos pode ser executada por meio dos passos descritos no Anexo XVI.

A ferramenta Microchip Stack, uma vez instalada, consiste de múltiplos diretórios e arquivos. Para garantir a compatibilidade com outras aplicações Zigbee da Microchip, existem sub-diretórios e arquivos comuns que são armazenadas em um único diretório. Cada novo projeto Zigbee é armazenado em seu próprio diretório. A Figura 72 ilustra os diretórios criados.

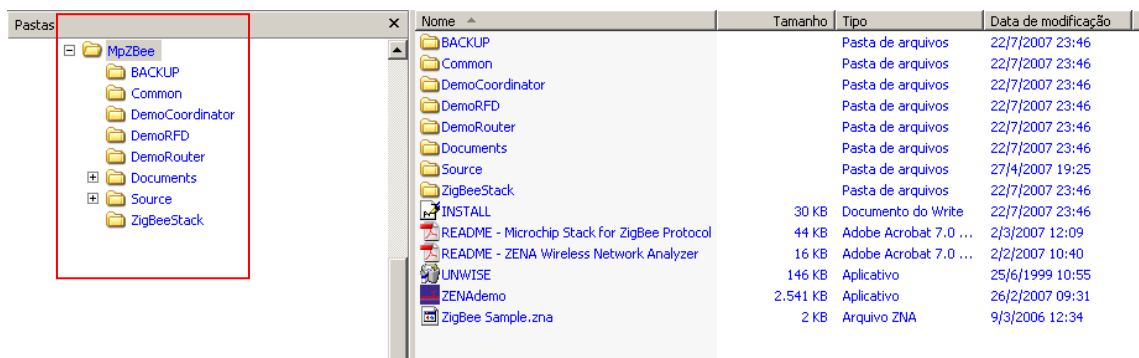


Figura 72 – Diretórios e subdiretórios da ferramenta Microchip Stack

A Tabela 12 descreve o conteúdo de cada subdiretório.

Nome do diretório	Conteúdo
Common	Arquivos-fontes comuns para o protocolo Zigbee e outras notas de aplicação Microchip.
Demo Coordinator	Código-fonte para demonstração de aplicação do coordenador de protocolo Zigbee.
Demo RFD	Código-fonte para demonstração de aplicação de dispositivo RFD de protocolo Zigbee.
Demo Router	Código fonte para demonstração de aplicação de roteador de protocolo Zigbee.
Documetation	Documentação da ferramenta Microchip Stack para protocolo Zigbee
Source	Código-fonte para aplicação de coordenador de protocolo Zigbee Código-fonte para aplicação de dispositivo RFD de protocolo Zigbee
ZigBee Stack	Arquivo-fonte comum da ferramenta Microchip Stack para protocolo Zigbee

Tabela 12 - Estrutura de diretórios de arquivos-fonte

O subdiretório *Common* possui os arquivos-fontes de uso comuns para qualquer aplicação Zigbee. A Figura 73 ilustra os arquivos que compõem o subdiretório, destacando-se os arquivos-fontes destinados às seguintes funções:

- comunicação serial entre o microcontrolador e o transmissor/receptor utilizando o protocolo de comunicação SPI (*MSPI.c*);
- compilador (*Compiler.h*) *C18*;
- comunicação serial entre o microcontrolador e o computador por meio do periférico USART (*Console.c*);

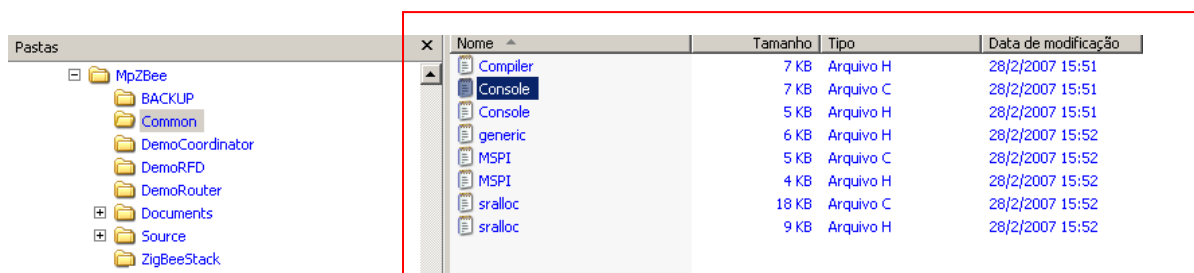


Figura 73 –Subdiretório Common

O subdiretório *DemoCoordinator* contém os códigos-fontes demonstrativos para o funcionamento de um dispositivo coordenador Zigbee. A Figura 74 ilustra os arquivos que compõem esse subdiretório. Destaca-se o arquivo *Coordinator.c*, que é um arquivo que configura as características de funcionamento do dispositivo coordenador.

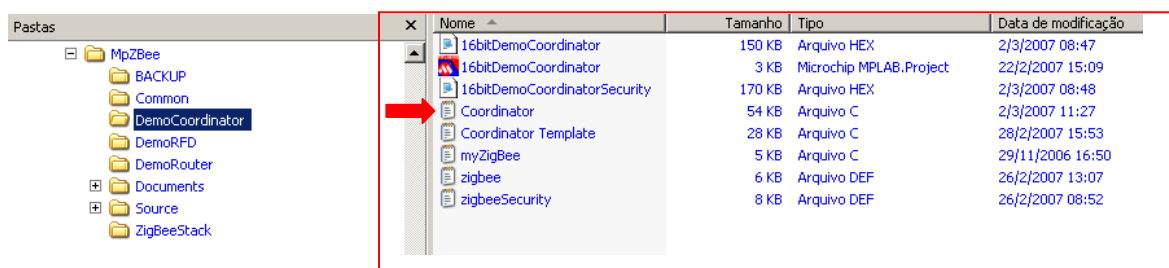


Figura 74 –Subdiretório DemoCoordinator

O subdiretório *DemoRFD* possui os códigos-fontes demonstrativos para funcionamento de um dispositivo final RFD Zigbee, ilustrado na Figura 75. Destaca-se o arquivo *RFD.c*, sendo um arquivo que configura as características de funcionamento do dispositivo final RFD.

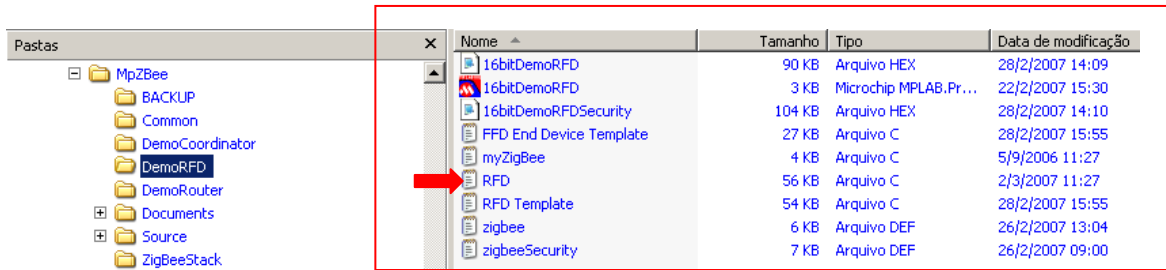


Figura 75 –Subdiretório DemoRFD

O subdiretório *DemoRouter* possui os códigos-fontes demonstrativos para funcionamento de um roteador Zigbee, ilustrado na Figura 76. Destaca-se aqui o arquivo *Router.c*, arquivo que configura as características de funcionamento do dispositivo roteador.

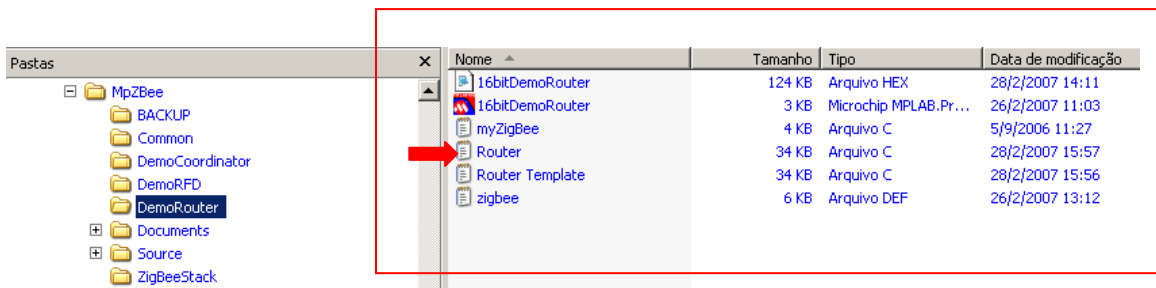


Figura 76 –Subdiretório DemoRouter

O subdiretório *Documents* possui todas as informações (documentos, catálogo técnico, catálogo de fabricante e guias de usuários) para realizar o esclarecimento do funcionamento de cada dispositivo Zigbee, como ilustrado na Figura 77. Destaca-se o arquivo *Microchip Stack for Zigbee*, que é um guia que fornece as condições de uso da ferramenta Microchip Stack.

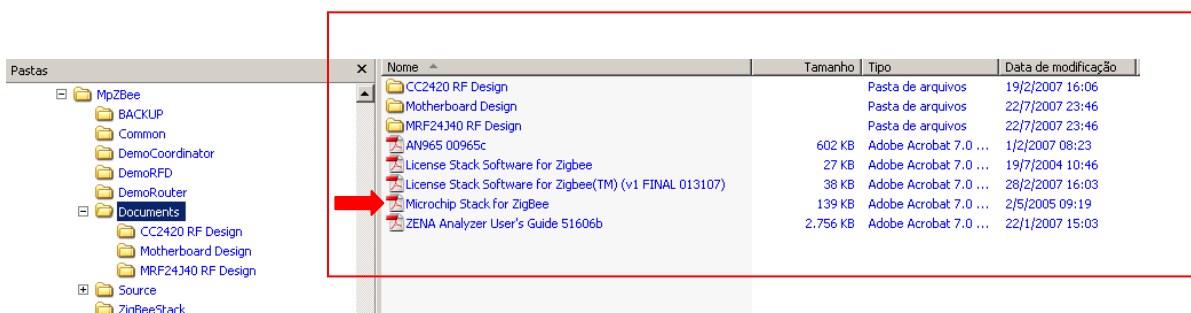


Figura 77 –Subdiretório Documents

O subdiretório *Source* contém o código fonte que dá suporte para todos os tipos de dispositivos do protocolo Zigbee. A Figura 78 ilustra o subdiretório citado. Por exemplo: cada

uma das aplicações Zigbee possui seu próprio arquivo *Zigbee.def* e *myZigBee.c* em seu respectivo diretório, além do arquivo HEX.

O arquivo *myZigBee.c*, entre suas funções, é responsável em definir o nível de sub-endereçamento (sub-addressing). Aqui são descritos os diferentes sub-endereçamentos para as seguintes funções:

- *Perfil ID*
- *IDs do dispositivo e versão*
- *IDs do cluster (agrupamento)*
- *IDs do atributo.*

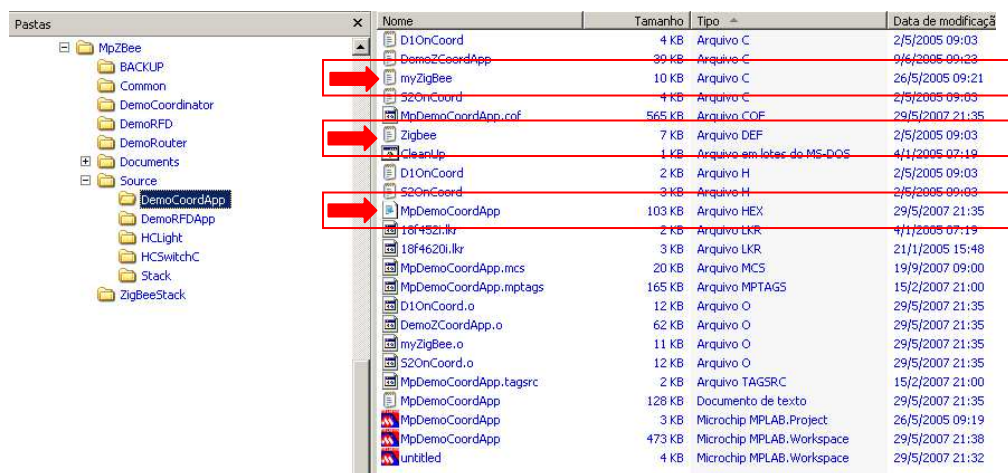


Figura 78 – Arquivos (Zigbee.DEF, myZigbee.C e nome do projeto. HEX) criados para cada aplicação

O arquivo *Zigbee.def* é responsável em definir:

- Características físicas do microcontrolador (frequência do oscilador, velocidade de transmissão do periférico USART)
- Tipos de transmissor /receptor de rádio frequência
- Definição do tipo de dispositivo
- Definição da frequência portadora de transmissão/recepção de dados
- Tempo de duração de varredura da rede
- Numero máximo de Clusters
- Numero de endpoints.

O arquivo *MPDemoCoordApp.hex* é o projeto de aplicação do Zigbee.

O subdiretório *ZigBeeStack* contém todos os códigos fontes para dar suporte às camadas do protocolo Zigbee (PHY, MAC, NWK e APL) em conformidade com a norma IEEE 802.15.4. A Figura 79 ilustra esse subdiretório.

Pastas	Nome	Tamanho	Tipo	Data de modificação
MpZBee	BACKUP			
	Common			
	DemoCoordinator			
	DemoRFD			
	DemoRouter			
	Documents			
	Source			
	ZigBeeStack			
	SymbolTime	6 KB	Arquivo C	28/2/2007 14:01
	SymbolTime	7 KB	Arquivo H	28/2/2007 14:04
	zAPL	7 KB	Arquivo H	28/2/2007 14:04
	zAP5	154 KB	Arquivo C	28/2/2007 14:01
	zAP5	8 KB	Arquivo H	28/2/2007 14:04
	zHCLighting	27 KB	Arquivo H	28/2/2007 14:04
	zigbee	4 KB	Arquivo H	28/2/2007 14:04
ZigbeeTasks	12 KB	Arquivo C	28/2/2007 14:01	
ZigbeeTasks	40 KB	Arquivo H	28/2/2007 14:04	
zMAC	8 KB	Arquivo H	28/2/2007 14:04	
zMAC_MRF24J40	135 KB	Arquivo C	28/2/2007 15:57	
zMAC_MRF24J40	5 KB	Arquivo H	28/2/2007 15:57	
zNWK	48 KB	Arquivo C	28/2/2007 15:57	
zNWK	19 KB	Arquivo H	28/2/2007 15:57	
zNWK	287 KB	Arquivo C	28/2/2007 15:57	
zNWK	11 KB	Arquivo H	28/2/2007 15:57	
zPHY	5 KB	Arquivo H	28/2/2007 15:57	
zPHY_MRF24J40	30 KB	Arquivo C	28/2/2007 15:58	
zPHY_MRF24J40	7 KB	Arquivo H	28/2/2007 15:58	
zPHY_UZ2400	19 KB	Arquivo C	26/2/2007 12:58	
zPHY_UZ2400	6 KB	Arquivo H	26/2/2007 12:58	
zSecurity	4 KB	Arquivo H	28/2/2007 15:58	
zSecurity_MRF24J40	24 KB	Arquivo C	28/2/2007 15:58	
zSecurity_MRF24J40	4 KB	Arquivo H	28/2/2007 15:58	
zZDO	105 KB	Arquivo C	28/2/2007 15:59	
zZDO	11 KB	Arquivo H	28/2/2007 15:58	

Figura 79 – Arquivos-fonte das camadas físicas (PHY), de enlace (MAC), de rede (NWK) e aplicação (APL) do protocolo Zigbee

5.4 PICDEM™ Z - PLACA DE DESENVOLVIMENTO

PICDEM™ Z é uma placa eletrônica de desenvolvimento que utiliza o protocolo Zigbee em microcontroladores da família Microchip PIC18Fxx. A placa é equipada com um conector modular de seis pinos para comunicação com o dispositivo de gravação MPLAB ICD2 que é um circuito interno de depuração (In-Circuit Debugger). O dispositivo de gravação MPLAB ICD 2 permite gravar, reprogramar ou modificar a memória do microcontrolador PIC18Fxx, permitindo ainda analisar o código-fonte (debugger code) para todas as plataformas de microcontroladores existentes da Microchip.

O PICDEM™ Z possui como principais características:

- Suporte para a ferramenta Microchip Stack para os tipos de dispositivos RFD, FFD e Coordenador

- Utiliza o microcontrolador PIC 18LF4620 com características de baixo consumo de energia elétrica (nanowatt), memória interna flash de 64K e periféricos integrados robustos
- Utiliza cartão de transmissão e recepção de dados (transceiver) da Microchip MRF 24J40 ou da Chipcon padrão IEEE 802.15.4
- Possui uma placa com interface USB; (opcional)
- Suporta a ferramenta de configuração Software Microchip ZENA
- Suporta um analisador de rede sem-fio
- Possui programação serial em circuito interno (in-circuit), com conexão da interface ICD2
- Possui uma interface RS 232
- Utiliza fonte de energia com entrada de 9VDC com regulador de tensão para 3.3VDC
- Possui um sensor de temperatura (Microchip TC77), Leds e chaves para ensaios.

A Figura 80 ilustra as diferentes partes que constituem a placa de desenvolvimento PICDEM™ Z.

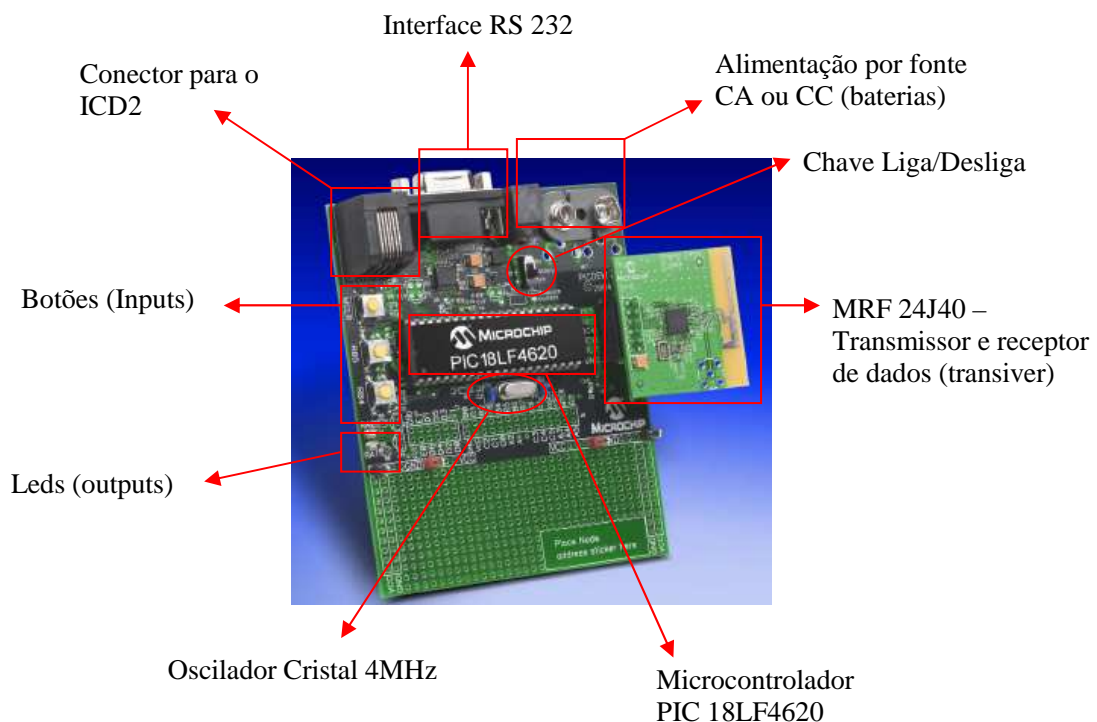


Figura 80 – Placa de desenvolvimento PICDEM Z

A placa de desenvolvimento PICDEM™ Z permite criar uma rede sem-fio com o protocolo Zigbee, utilizando os seguintes dispositivos:

- Um microcontrolador Microchip com uma interface SPI
- Um transmissor de RF
- Uma antena
- Uma fonte de alimentação
- Um conjunto de periféricos (botões, leds, conectores para comunicação).

Como ilustrado na Figura 81, o microcontrolador é conectado ao transmissor e receptor de dados por meio de um barramento, utilizando o protocolo de comunicação SPI. O microcontrolador é definido como mestre SPI e o transmissor e receptor de dados é definido como escravo SPI. O barramento SPI é usado para interagir com o transmissor e receptor de dados.

A placa de desenvolvimento PICDEM™ Z associado à ferramenta Microchip Stack fornecem o hardware e o código-fonte, respectivamente, para gerenciar as funções do protocolo SPI, otimizando os esforços para administrar as funções de transmissão e recepção de dados. No código de programação mostrado do Anexo XXI, é apresentado o código fonte para gerenciamento do protocolo SPI em linguagem C, cujo objetivo é controlar a comunicação do protocolo Zigbee entre o microcontrolador e o transmissor/ receptor de dados por meio de radiofrequência.

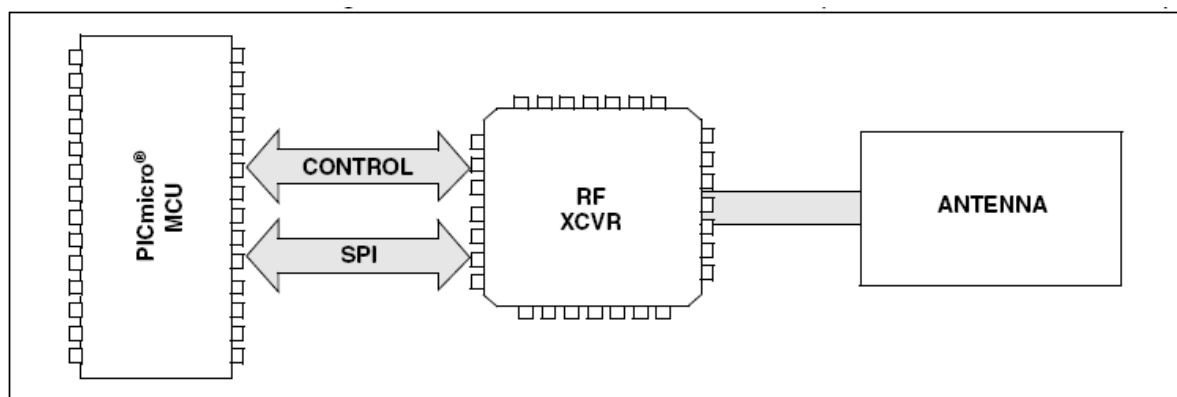


Figura 81 – Hardware do PICDEM Z (sinais de controle adicionados)

A placa de desenvolvimento PICDEM Z Microchip utiliza uma fonte de 3.3V para o microcontrolador e para o transmissor/receptor de radiofrequência. Também, é possível usar a rede elétrica ou uma bateria para sua alimentação de energia. Tipicamente, o coordenador do

protocolo Zigbee e os roteadores operam com uma fonte de energia oriunda da rede elétrica e os dispositivos finais (RFD) operam com baterias.

5.5 ZENA

O ZENA é uma ferramenta de configuração e análise para uma rede sem-fio que utiliza o protocolo Zigbee.

Para ajudar no desenvolvimento de aplicações do protocolo Zigbee, a Microchip fornece uma ferramenta composta de programas e placas (hardware) de baixo custo, chamado de ZENA, permitindo criar configurações para aplicações específicas do protocolo Zigbee. O software ZENA é fornecido gratuitamente como parte do Microchip Stack e pode ser obtido do site do fabricante. Entretanto, o hardware deve ser adquirido por um fornecedor credenciado pela Microchip.

O software ZENA oferece a capacidade de criar arquivos fonte da camada de aplicação (APL) do protocolo Zigbee que, associado aos códigos fonte da ferramenta Microchip Stack, permite criar diferentes aplicações com o protocolo Zigbee. A versão do software ZENA com características completas (full), associado à placa de hardware, permite capturar em tempo real as atividades da rede sem-fio para protocolos Zigbee e MiWi². A Figura 82 ilustra a tela do software ZENA.

O hardware permite analisar o tráfego da rede sem-fio baseada nos protocolos Zigbee e MiWi. Para a captura dos pacotes (frames) do protocolo, é necessária a aquisição de uma placa de hardware que inclui um detector de radiofrequência, que pode ser conectado a um PC por meio de uma porta USB. A Figura 83 mostra uma foto do hardware de captura de RF.

Durante o uso da ferramenta ZENA para configuração do protocolo Zigbee, são criados três arquivos para a aplicação, sendo eles: *zigbee.def*, *myZigBee.c* e *zLink.lkr*, descritos na Seção 5.3 deste trabalho.

É altamente recomendável que seja usado o software ZENA para gerar estes arquivos no lugar de editar os arquivos manualmente, desde que os arquivos sejam interdependentes. A Figura 84 ilustra a tela onde aparecem esses arquivos. Todos os fabricantes recomendam o uso das ferramentas de configuração e monitoração.

² MiWi é um protocolo proprietário da Microchip, com comunicação sem fio, destinado à transmissão e recepção de dados com capacidade e velocidade inferior ao protocolo Zigbee.

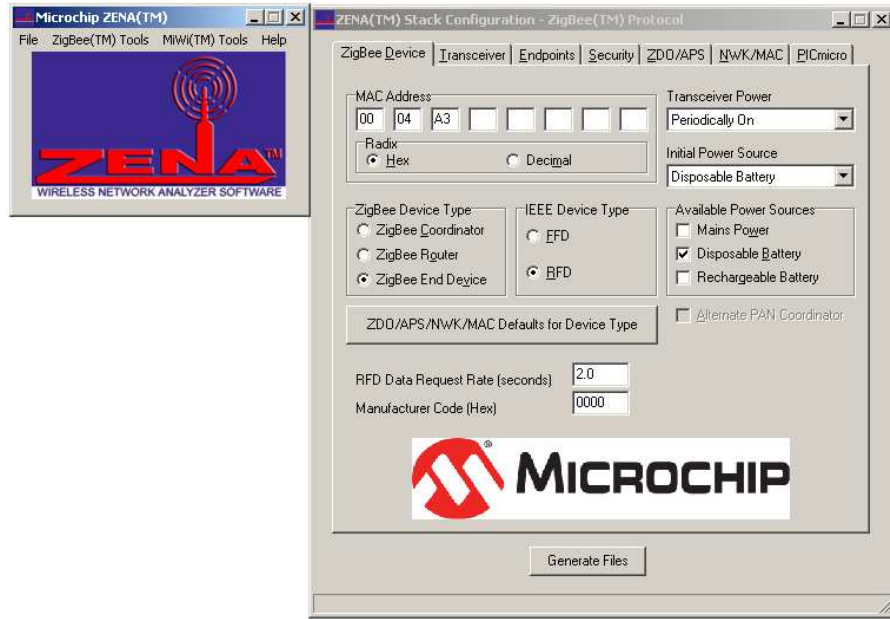


Figura 82 – Software ZENA

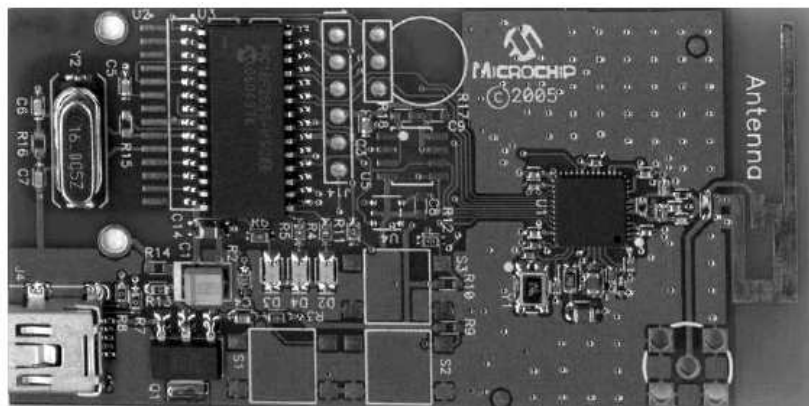


Figura 83 – Hardware para captura de radiofrequência

Nome	Tamanho	Tipo	Data de modificação
myZigBee	4 KB	Arquivo C	13/6/2007 17:30
zigbee	4 KB	Arquivo DEF	13/6/2007 17:30
zLink.lkr	4 KB	Arquivo LKR	13/6/2007 17:30

Figura 84 – Arquivos criados pelo software ZENA

5.6 CONCLUSÃO

A ferramenta Microchip Stack para o protocolo Zigbee fornece uma biblioteca modular, fácil de usar, aplicável e independente. Especificamente, é projetada para suportar mais de um transmissor de RF com mudanças mínimas para camada superior APL. As aplicações podem ser facilmente repassadas de um transmissor de RF para outro.

A ferramenta Microchip Stack é projetada para funcionar em associação com o compilador MPLAB C18, mas pode ser modificada para suportar outros compiladores. Os arquivos de código fonte fornecidos facilitam o projeto de novos aplicativos. A ferramenta Microchip Stack, é responsável pelos códigos fonte de uso comum do protocolo Zigbee. A placa de desenvolvimento PICDEM™ Z é uma placa eletrônica que utiliza o protocolo Zigbee em microcontroladores da família Microchip PIC18Fxx, equipada com um microcontrolador (40 pinos) de 8 bits. Possui um transmissor e receptor de dados que se comunica com o microcontrolador por meio do protocolo SPI, possuindo chaves e leds de sinalização e sendo alimentado por baterias. A placa de desenvolvimento PICDEM™ Z é o hardware do protocolo Zigbee.

O ZENA é uma ferramenta composta de um software e de um hardware. O software de programação e análise permite configurar os três principais arquivos para desenvolvimento de um novo projeto. Com a aquisição da placa de hardware, permite-se analisar com detalhes a montagem dos pacotes de mensagens das diferentes camadas do protocolo. Pode-se identificar se os dados são transferidos durante cada período de símbolo (um período de símbolo equivale a 16ms, utilizando o Microchip Stack, a duração de uma varredura está entre 0.031s e 4.2min). O ZENA é uma ferramenta para configurar novos aplicativos do protocolo Zigbee e associado ao hardware correspondente permite monitorar os pacotes de mensagens em uma rede Zigbee.

Todas as ferramentas mostram-se eficientes e com funções bem delimitadas. São complementares, mas de uso complexo.

6 ANÁLISE DOS RESULTADOS

Neste capítulo são ensaiados e analisados os resultados de uma rede sem fio utilizando o protocolo Zigbee. Pretende-se, ao final, responder as seguintes questões levantadas no capítulo 1:

- Qual a distância de alcance em ambientes industriais do protocolo Zigbee?
- Como funciona e quais componentes eletrônicos são utilizados no desenvolvimento do hardware?
- Qual o consumo de energia (em horas) de um dispositivo eletrônico utilizando o protocolo Zigbee?
- Como funciona o sistema de endereçamento dos dispositivos de uma rede de comunicação Zigbee?
- É possível monitorar e identificar o frame (pacote de dados) de uma rede de comunicação Zigbee?
- Existem ferramentas para programar e controlar uma rede de comunicação Zigbee?

6.1 INTRODUÇÃO

Para auxiliar nas respostas acima levantadas, dois experimentos práticos foram realizados. O experimento 1 objetiva a transmissão de dados de um dispositivo RFD ao dispositivo coordenador, permitindo analisar o funcionamento do modo de sincronismo e pedido de dados dos dispositivos RFDs. Os dados transmitidos pelos dispositivos coordenadores e RFDs são capturados pelo analisador ZENA e encaminhados ao periférico USART, sendo comparado com a norma IEEE 802.15.4 e com a norma Zigbee Alliance. O experimento permite uma análise por comparação entre os descritos das normas (estudados nos capítulos anteriores) e a prática demonstrada pelo analisador ZENA do fabricante Microchip.

O experimento 2 permite identificar os endereços e sub-endereços da camada APS. Por meio do analisador ZENA, é possível identificar o funcionamento dos sub-endereços de responsabilidade da camada de aplicação (APL). A identificação dos end

points, dos clusters e do tipo de perfil são alguns sub-endereços ilustrados no analisador ZENA em concordância com a norma Zigbee Alliance.

Para os experimentos, um conjunto de ferramentas deve ser utilizado.

Para o hardware, a placa eletrônica de desenvolvimento PICDEM™ Z é utilizada e associada às ferramentas ICD2³ e o analisador ZENA. Os softwares utilizados são classificados em 2 categorias:

Categoria 1: softwares utilizados para programação e compilação do microcontrolador, entre eles o MPLAB e o C18.

Categoria 2: softwares utilizados para programação e análise da rede sem fio Zigbee, entre eles o Microchip Stack e o analisador ZENA.

Os objetivos deste capítulo são:

- Comparar os formatos dos pacotes (frame) descritos na norma IEEE 802.15.4 e a norma Zigbee Alliance, com o formato dos pacotes (frame) coletados no analisador ZENA do fabricante Microchip em tempo real
- Verificar o funcionamento do sistema (modo) de sincronismo entre dispositivos coordenador e RFDs em uma rede Zigbee
- Identificar os dispositivos coordenadores e RFDs por meio dos endereços da camada MAC
- Analisar o funcionamento dos sub-endereços da camada APS, comparando os descritivos da norma Zigbee Alliance referente ao end point, clusters e perfil, com os resultados coletados no analisador ZENA em tempo real.

6.2 EXPERIMENTO 1: PROTOCOLO ZIGBEE

O experimento 1 objetiva a transmissão de dados de um dispositivo RFD ao dispositivo coordenador, permitindo analisar o funcionamento do modo de sincronismo e pedido de dados dos dispositivos RFDs.

Para o experimento são utilizadas as seguintes ferramentas de desenvolvimento:

- Microchip Stack
- PICDEM™ Z
- ZENA

³ ICD2 é uma ferramenta de gravação e depuração para microcontroladores e DSPs da família Microchip. Suporta a monitoração em tempo real.

- ICD2 (compilador e gravado do microcontrolador)
- MPLAB (software de programação do microcontrolador do fabricante Microchip).

Os dados transmitidos são acionados pelo botão RB5 (placa eletrônica de desenvolvimento PICDEM™ Z) do dispositivo RFD, ilustrado na Figura 85.

O experimento compara o formato do pacote (frame) descrito na norma IEEE 802.15.4 e Zigbee Alliance, com o formato do pacote (frame) coletado no Analisador ZENA.

É importante lembrar que a norma IEEE 802.15.4 determina os modos de transferência de dados.



Figura 85 – Botão RB5 do dispositivo RFD (Placa eletrônica de desenvolvimento PICDEM™ Z)

No experimento 1, o coordenador emite sinal de sincronismo por meio do pacote de dados para sincronismo. Em seguida aguarda recepção de dado, transmitida pelo dispositivo RFD, podendo ou não confirmar o recebimento (acknow), conforme ilustrado na Figura 86.

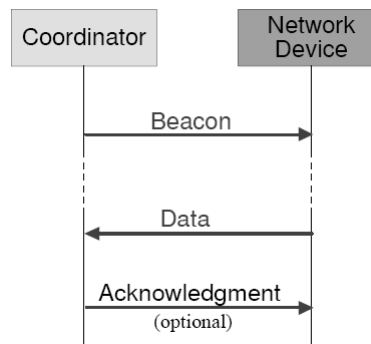


Figura 86 – Modo de transferência de dados do RFD para o coordenador, com sincronismo – Norma IEEE 802.15.4

Para execução do experimento 1, é necessário configurar, gravar e monitorar separadamente cada dispositivo. O dispositivo coordenador apresenta características que diferem dos dispositivos RFDs. Cada um dos dispositivos é detalhado separadamente no experimento 1.

6.2.1 Dispositivo Coordenador – Experimento 1

O dispositivo coordenador necessita de um procedimento para gravação, configuração e execução do protocolo Zigbee. Este procedimento considera o uso do software MPLAB® IDE para construir as aplicações. Apresentamos em seguida, as etapas do procedimento utilizado no experimento 1:

1. Instalação do arquivo fonte da ferramenta Microchip Stack para o protocolo de ZigBee (instalado na raiz do diretório C).
2. Importante deslocar os arquivos fonte da ferramenta Microchip Stack dentro do diretório *MpZBee* na raiz do disco rígido. Nesta condição o projeto de aplicação *MpDemoCoordApp.mcp* trabalha corretamente.
3. Uso do MPLAB IDE utilizando os arquivos de projeto apropriados do Microchip Stack: *MpZBee\Source\DemoCoordApp\MpDemoCoordApp.mcp* para a aplicação do dispositivo coordenador do protocolo Zigbee, e *MpZBee\Source\DemoRFDApp\MpDemoRFDApp.mcp* para a aplicação do dispositivo RFD. A Figura 87 ilustra a abertura do diretório para o coordenador no software MPLAB IDE. Observe o diretório *MpZBee* do Microchip Stack sendo aberto pelo diretório raiz C.

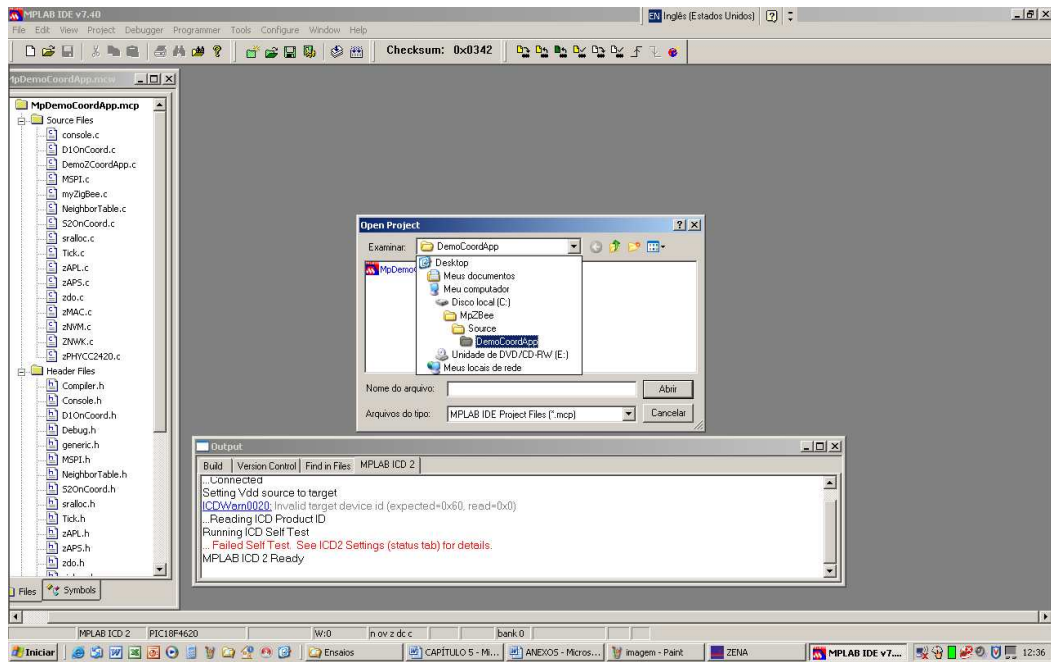


Figura 87 – Abertura de projeto no MPLAB (Arquivo fonte: MpDemoCoordApp.mcp)

4. Utilizou-se o compilador C18 para compilar o código fonte da ferramenta Microchip Stack. A Figura 88 ilustra a conexão entre o computador (PC), a ferramenta de gravação ICD2 e a placa eletrônica de desenvolvimento PICDEM™ Z⁴.

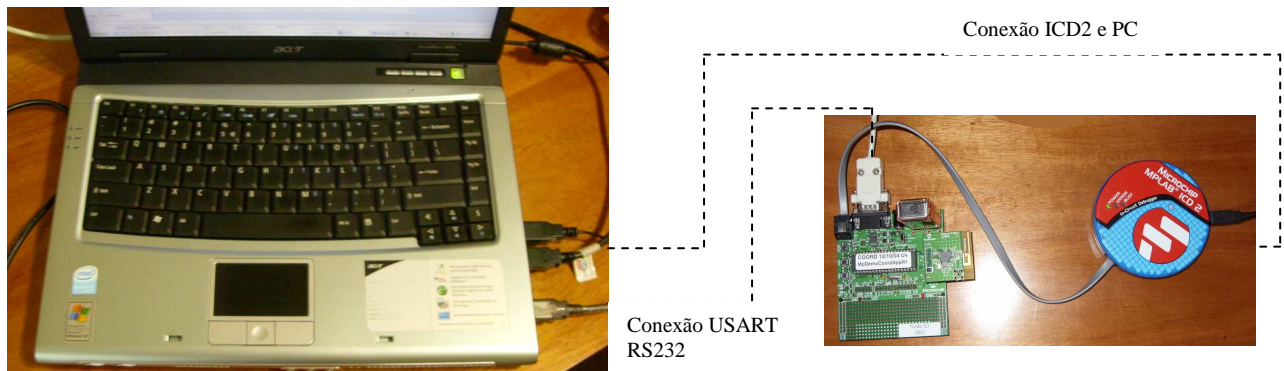


Figura 88 – Conexão física dos dispositivos ICD2, PICDEM Z e PC

5. Utilizou-se a ferramenta ICD2 como um programador. Conectamos o ICD2 com a placa de desenvolvimento PICDEM™ Z em seu conector designado, aplicando energia no conector designado.

6. Foi habilitado o ICD2 como um programador e selecionamos a opção do ICD2 *menu programador* e executamos um procedimento de teste (self-test). Após conexão do PC com o ICD2, uma mensagem de teste foi exibida. A Figura 89 ilustra a mensagem de teste (self-test) para a conexão do ICD2.
7. Apagou-se a memória do microcontrolador PIC18Fxx por meio do comando *Erase target device*. A Figura 90 ilustra o comando e a mensagem de sucesso.
8. Programou-se o arquivo fonte da ferramenta Microchip Stack. A Figura 91 ilustra a mensagem de programação e verificação correta do arquivo.
9. Após alguns segundos, apareceu a mensagem de confirmação de sucesso da programação, "*Programming succesful*". (Figura 91)
10. Desenergizou-se o circuito e desconectou-se o cabo entre ICD2 e a placa de desenvolvimento PICDEM™ Z demo board.
11. A verificação de funcionamento do programa pode ser evidenciada por meio da conexão serial (RS 232) entre o PC e a placa de desenvolvimento PICDEM™ Z. Para ver a operação da rede Zigbee, é recomendado a conexão da placa de desenvolvimento PICDEM™ Z com uma porta serial de um PC. Recomenda-se utilizar o HyperTerminal ou outro software de interface serial para comunicar com o PICDEM™ Z demo board. A porta deve ser configurada com as seguintes características: 19200bps, 8 data bytes, 1 stop bit, sem paridade e sem controle de fluxo. Estas características são definidas no arquivo *Console.c* da ferramenta Microchip Stack.
12. Configurou-se o endereço identificador da placa (*Node ID*), por meio da opção 1 do *menu choice*. A Figura 92 ilustra a mensagem pelo HyperTerminal.
13. Após informar o endereço de ID, uma mensagem ilustrada na Figura 93 informa a confirmação do ID.
14. Entrou em operação (*Run application*) pela opção 0. A Figura 94 ilustra a mensagem de confirmação.

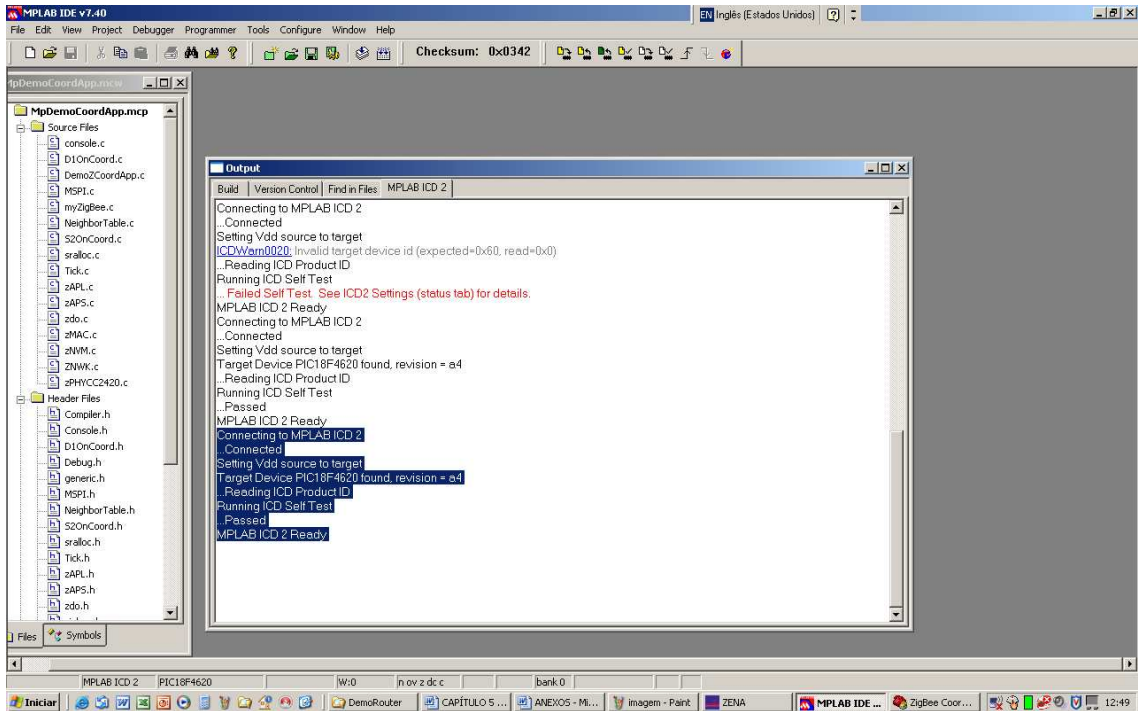


Figura 89 – Mensagem de Teste referente à conexão do ICD2

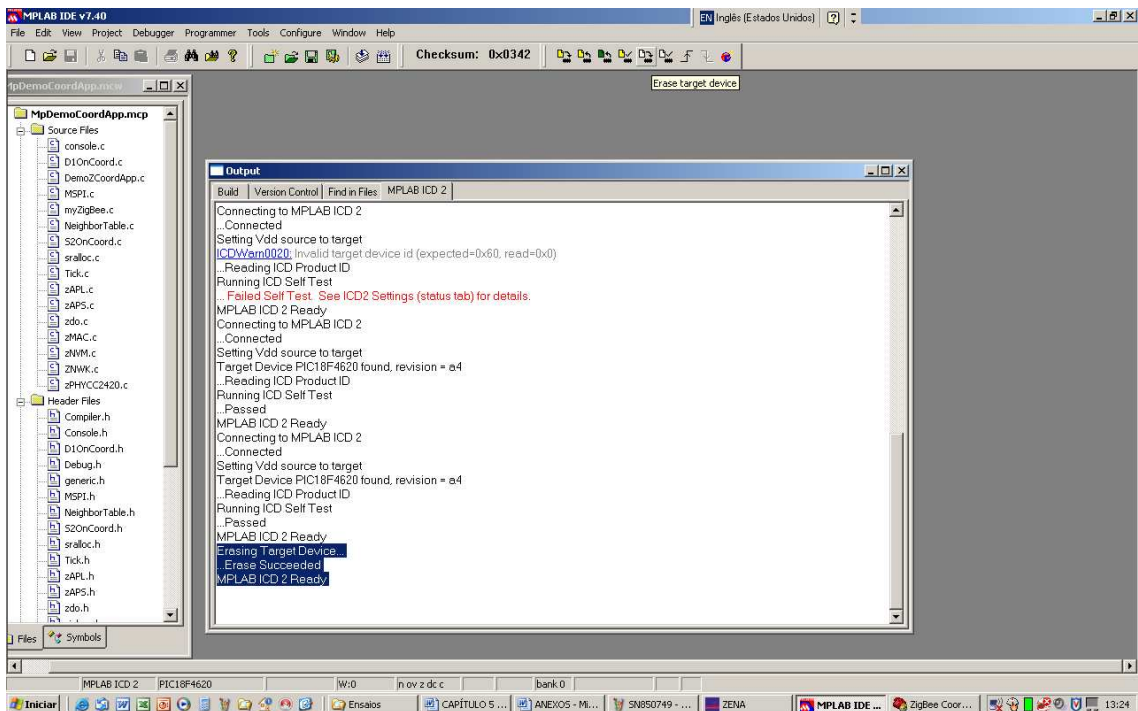


Figura 90 – Comando e Mensagem para apagar a memória no PIC18Fxx

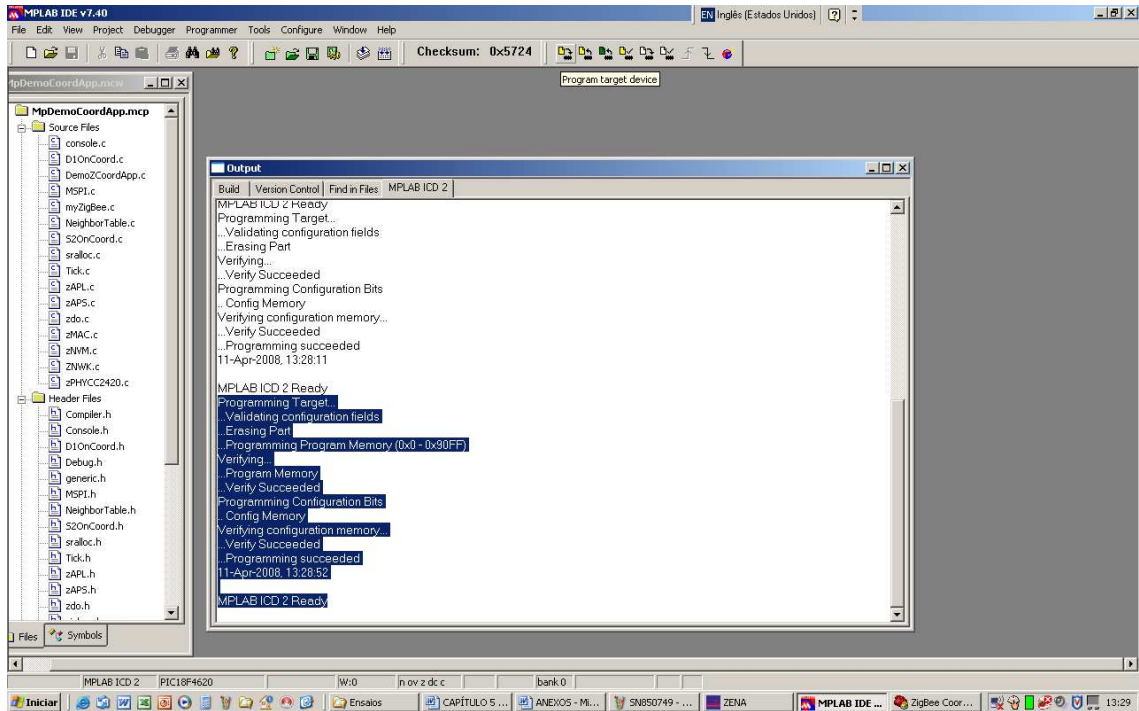


Figura 91 – Comando e Mensagem para programação e verificação do arquivo fonte ZigBee Coordenador

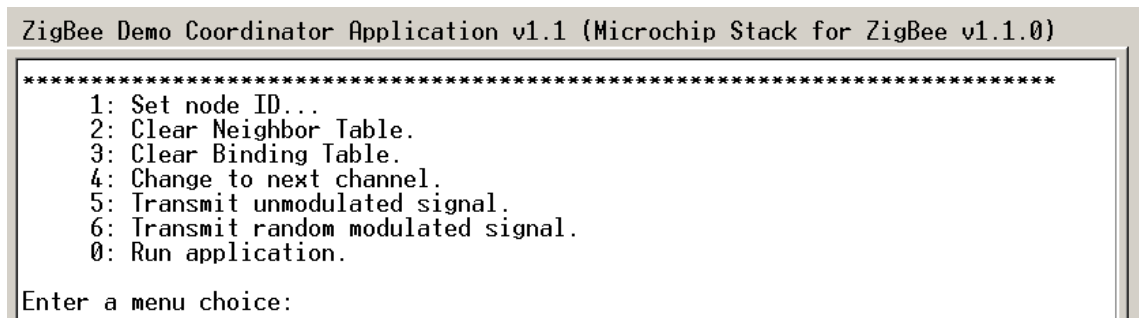


Figura 92 – Mensagem de abertura do ZigBee Coordenador (comunicação serial RS 232)

```

Built on Apr 11 2008 (Non-debug Build)
*****
1: Set node ID...
2: Clear Neighbor Table.
3: Clear Binding Table.
4: Change to next channel.
5: Transmit unmodulated signal.
6: Transmit random modulated signal.
0: Run application.

Enter a menu choice:
Enter up to 4 digit long decimal board id:

*****
ZigBee Demo Coordinator Application v1.1 (Microchip Stack for ZigBee v1.1.0)
  Built on Apr 11 2008 (Non-debug Build)
*****
1: Set node ID...
2: Clear Neighbor Table.
3: Clear Binding Table.
4: Change to next channel.
5: Transmit unmodulated signal.
6: Transmit random modulated signal.
0: Run application.

Enter a menu choice:

```

Figura 93 – Mensagem de confirmação do ID do ZigBee Coordenador

```

*****
ZigBee Demo Coordinator Application v1.1 (Microchip Stack for ZigBee v1.1.0)
  Built on Apr 11 2008 (Non-debug Build)
*****
1: Set node ID...
2: Clear Neighbor Table.
3: Clear Binding Table.
4: Change to next channel.
5: Transmit unmodulated signal.
6: Transmit random modulated signal.
0: Run application.

Enter a menu choice: Now running application...
Starting a new network...
Now operating in next channel...
New network successfully started.
-

```

Figura 94 – Mensagem de confirmação do estado de operação do ZigBee Coordenador (rede formada)

6.2.2 Dispositivo RFD – Experimento 1

O dispositivo RFD necessita de um procedimento similar ao dispositivo coordenador para gravação, configuração e execução do protocolo Zigbee. Este procedimento considera o uso do software MPLAB® IDE para construir as aplicações. Apresentam-se, em seguida, as etapas do procedimento utilizado no experimento 1 para o dispositivo RFD:

1. Este procedimento é similar ao apresentado para o ZigBee dispositivo Coordenador. Por questões práticas, serão demonstrados e ilustrados os comandos a partir da gravação do código fonte `MpZBee\Source\DemoRFDApp\MpDemoRFDApp.mcp` da ferramenta Microchip Stack. A Figura 95 ilustra a mensagem de programação e verificação corretas.
2. Depois de alguns segundos, a mensagem de sucesso na programação apareceu, “*Programming successful*”. (Figura 95)
3. Desenergizou-se o circuito e desconectou-se o cabo entre ICD2 e a placa de desenvolvimento PICDEM™ Z demo board.
4. Verificou-se o funcionamento do programa por meio da conexão serial RS 232 entre o PC e a placa de desenvolvimento PICDEM™ Z. A configuração serial recomendada deve seguir as características: 19200bps, 8 data bytes, 1 stop bit, sem paridade e sem controle de fluxo.
5. Configurou-se o endereço identificador da placa (*Node ID*), por meio da opção 1 do *menu choice*. A Figura 96 ilustra a mensagem.
6. Após informar o endereço de ID, uma mensagem ilustrada na Figura 97 confirmou o ID.
7. Conectou-se o dispositivo RFD a um dispositivo coordenador Zigbee pela opção 2 (*Join a network*). A Figura 98 ilustra a mensagem de confirmação.

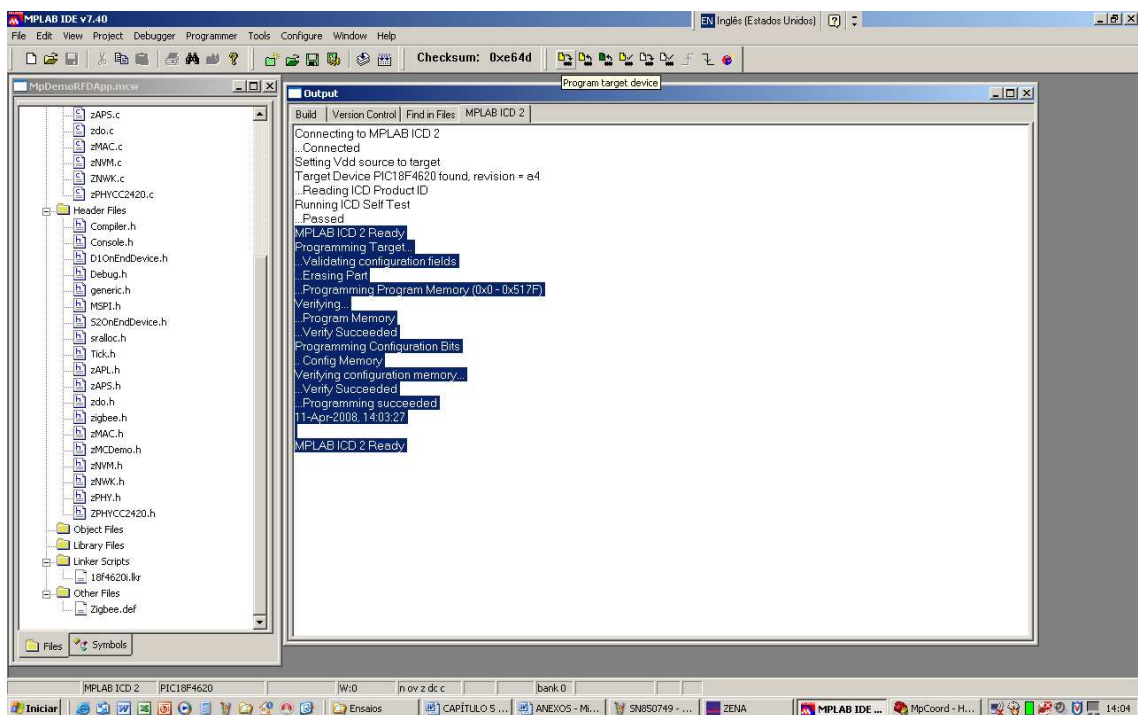


Figura 95 – Comando e Mensagem para programação e verificação do arquivo fonte ZigBee RFD

```

*****
ZigBee Demo RFD Application v1.1 (Microchip Stack for ZigBee v1.1.0)
  Built on Apr 11 2008 (Non-debug Build)
*****
1: Set node ID...
2: Join a network.
3: Perform quick demo binding (Must perform #2 first).
4: Bind statically known endpoints together (Must perform #2 first).
5: Leave a previously joined network (Must perform #2 first).
6: Change to next channel.
7: Transmit unmodulated signal.
8: Transmit random modulated signal.
0: Save changes and exit.
Enter a menu choice: _

```

Figura 96 – Mensagem de abertura do ZigBee RFD (comunicação serial RS 232)

```

ZigBee Demo RFD Application v1.1 (Microchip Stack for ZigBee v1.1.0)
  Built on Apr 11 2008 (Non-debug Build)
1: Set node ID...
5: Leave a previously joined network (Must perform #2 first).
6: Change to next channel.
7: Transmit unmodulated signal.
8: Transmit random modulated signal.
0: Save changes and exit.
Enter a menu choice:
Enter exact 4 digit long decimal board id:
*****
ZigBee Demo RFD Application v1.1 (Microchip Stack for ZigBee v1.1.0)
  Built on Apr 11 2008 (Non-debug Build)
*****
1: Set node ID...
2: Join a network.
3: Perform quick demo binding (Must perform #2 first).
4: Bind statically known endpoints together (Must perform #2 first).
5: Leave a previously joined network (Must perform #2 first).
6: Change to next channel.
7: Transmit unmodulated signal.
8: Transmit random modulated signal.
0: Save changes and exit.
Enter a menu choice:

```

Figura 97 – Mensagem de confirmação do ID do ZigBee RFD

8. Salvou-se e no *menu choice* pela opção 0 (*Save changes and exit*). A Figura 99 ilustra a mensagem de confirmação.

```

*****
ZigBee Demo RFD Application v1.1 (Microchip Stack for ZigBee v1.1.0)
  Built on Apr 11 2008 (Non-debug Build)
*****
  1: Set node ID...
  2: Join a network
  3: Perform quick demo binding (Must perform #2 first).
  4: Bind statically known endpoints together (Must perform #2 first).

  5: Leave a previously joined network (Must perform #2 first).

  7: Transmit unmodulated signal.
  8: Transmit random modulated signal.
  0: Save changes and exit.

Enter a menu choice: Attempting to join a coordinator...

Successfully associated.

```

Figura 98 – Mensagem de confirmação da associação do RFD com um coordenador Zigbee

```

*****
ZigBee Demo RFD Application v1.1 (Microchip Stack for ZigBee v1.1.0)
  Built on Apr 11 2008 (Non-debug Build)
*****
  1: Set node ID...
  2: Join a network.
  3: Perform quick demo binding (Must perform #2 first).
  4: Bind statically known endpoints together (Must perform #2 first).
  5: Leave a previously joined network (Must perform #2 first).
  6: Change to next channel.
  7: Transmit unmodulated signal.
  8: Transmit random modulated signal.
  0: Save changes and exit.

Enter a menu choice:
Now running application...
Rejoin successful.
-

```

Figura 99 – Mensagem de confirmação de conexão e saída do RFD com um coordenador Zigbee

Após as configurações executadas dos dispositivos coordenador e RFD, o botão RB5 no dispositivo RFD poderá ser acionado para enviar um dado com um pedido de confirmação para o coordenador do protocolo ZigBee. Caso o dispositivo coordenador esteja ativo na rede, uma mensagem (Figura 100) no RFD confirmará a transmissão do dado e a confirmação que o coordenador recebeu.

```

|S2: Sending switch update...          |
|S2: Confirmed switch update.         |

```

Figura 100 – Mensagem de envio e confirmação do dado do RFD ao Coordenador ZigBee

Caso o dispositivo coordenador não esteja ativo na rede, uma mensagem (Figura 101) no RFD confirmará a transmissão do dado e a não confirmação do coordenador da rede ZigBee.

```
S2: Sending switch update...
S2: Retrying switch update...
S2: Sending switch update...
S2: Retrying switch update...
S2: Sending switch update...
S2: Exhausted switch update retries; removing it...
```

Figura 101 – Mensagem de envio e não confirmação do dado do RFD ao Coordenador ZigBee

6.2.3 Analisando os dados no Analisador ZENA - Experimento 1

Antes da transmissão e recepção dos dados na rede Zigbee, o coordenador executa o procedimento de sincronismo, transmitindo sinais de Beacon em uma rede sinalizada. Analisando o experimento com o auxílio da ferramenta ZENA, é possível identificar a transmissão dos dados pelo dispositivo RFD e a confirmação pelo dispositivo coordenador. A Figura 102a ilustra o modo e sentido de transmissão dos dados de acordo com a norma IEEE 802.15.4. A Figura 102b comprova o requerimento de Beacon do coordenador em uma rede sinalizada, por meio do uso do Analisador ZENA. (destacada em retângulo verde tracejado)

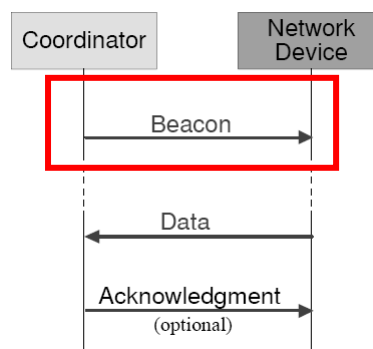


Figura 102a – Modo de transferência de dados do Coordenador para o RFD – Norma IEEE 802.15.4

Frame	Time(us)	Len	MAC Frame Control					Seq Num	Dest PAID	Dest Addr	Beacon Request	FCS		
00001	+3708240 -3708240	10	Type	Sec	Pend	ACK	IPAN	0x5D	0xFFFF	0xFFFF		RSSI	Corr	CRC
			CMD	N	N	N	N					-01	0x6C	OK

Figura 102b – Frame de Beacon monitorado em tempo real pelo Analisador ZENA do RFD ZigBee

Enquanto não houver resposta de um RFD, o coordenador solicita sincronismo da rede (*Beacon Request*). O Experimento prático 1 confirma o funcionamento do sistema de sincronismo do coordenador por meio dos pacotes de dados para sincronismo. Comparamos a seguir o pacote de dados da norma IEEE com os dados coletados no Analisador ZENA. (Figura 103a e 103b)

Frame de Beacon

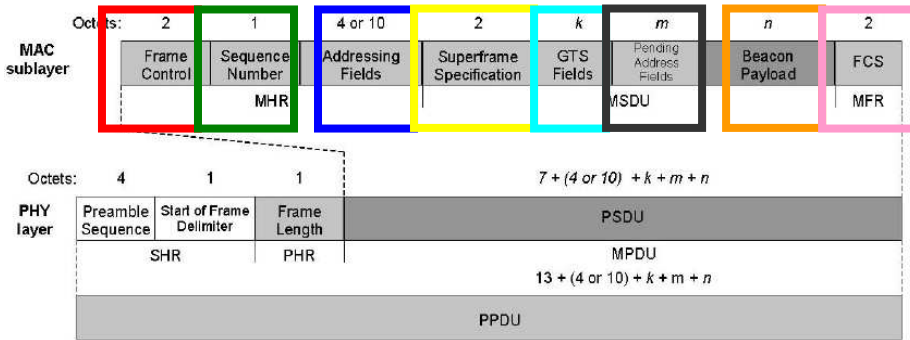


Figura 103a – Frame de Beacon descrito na Norma IEEE 802.15.4

Frame	Time(us)	Len	MAC Frame Control					Seq Num	Dest PAID	Dest Addr	Beacon Request	FCS		
00003	+163013456 -298319264	10	Type	Sec	Pend	ACK	IPAN	0x01	0xFFFF	0xFFFF		RSSI	Corr	CRC
			CMD	N	N	N	N					-13	0x6B	OK

Frame	Time(us)	Len	MAC Frame Control					Seq Num	Source PAID	Source Addr	SuperFrame Specification					GTS Specification		PendAddr Spec			
00004	+6880 -298326144	16	Type	Sec	Pend	ACK	IPAN	0x5F	0x3DA8	0x0000	BO	SO	CAP	Batt	Coord	Assc	Permit	Count	ExtAddr	ShortAddr	
			ECN	N	N	N	N				None	None	0x0	N	Y	Y	N	0x0	0x0	0x0	0x0

Beacon Payload							FCS		
DevCap	Depth	RtrCap	NWKVer	StkProf	ProtII		RSSI	Corr	CRC
Y	0x0	N	0x1	0x0	0x00		+03	0x6C	OK

Figura 103b – Frame de Beacon monitorado em tempo real no Analisador ZENA

Cada pacote descrito na norma IEEE 802.15.4, com seus respectivos bits de controle, aparece no analisador ZENA. O tipo de pacotes de dados é definido pelo comando *Type*: *BCN* observado no ZENA. São identificados o superframe com um período de disputa livre para acesso ao meio, denominado de CAP e os GTS, slots de tempo garantido utilizados para garantir a comunicação entre alguns dispositivos (esta função está desligada [*GTS Specification: Permit: N*]).

O pacote de dados de comando é descrito na norma IEEE de acordo com a ilustração da Figura 104, e é comprovado no ZENA pela mensagem *Type: CMD* ilustrado na cor azul escuro na Figura 105 (*Frame control*). Os endereços da fonte e do destinatário são ilustrados na cor azul clara. Exatamente a seqüência descrita na norma IEEE 802.15.4 (Figura 104), é coletado no analisador ZENA (*Dest PAN, Dest Addr, Source PAN e Source Addr* - Figura 105).

Frame de Comando

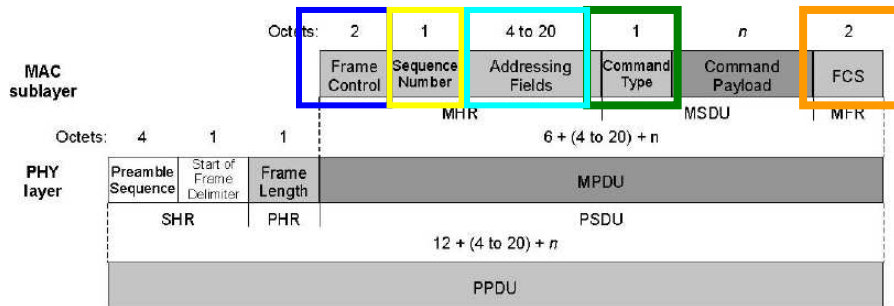


Figura 104 – Frame de Comando descrito na Norma IEEE 802.15.4

Frame	Time(us)	Len	MAC Frame Control							Seq Num	Dest PAN	Dest Addr	Source PAN	Source Addr	Data Request	FCS		
00026	+497728 =5946544	14	Type	Sec	Pend	ACK	IPAN		0xF	0x0396	0x0000	0x0396	0x0001		+21	0x6B	OK	

Figura 105 – Frame de Comando monitorado em tempo real no Analisador ZENA

No pacote de dados da camada MAC, ilustrado na Figura 106, o analisador ZENA comprova a seqüência descrita na norma IEEE. No dado transmitido o coordenador não solicita confirmação de envio (*Ack. Request*). A confirmação é vista no analisador ZENA em cor amarela tracejada (*ACK = N, ilustrado pela Figura 107*).

Frame de Dados

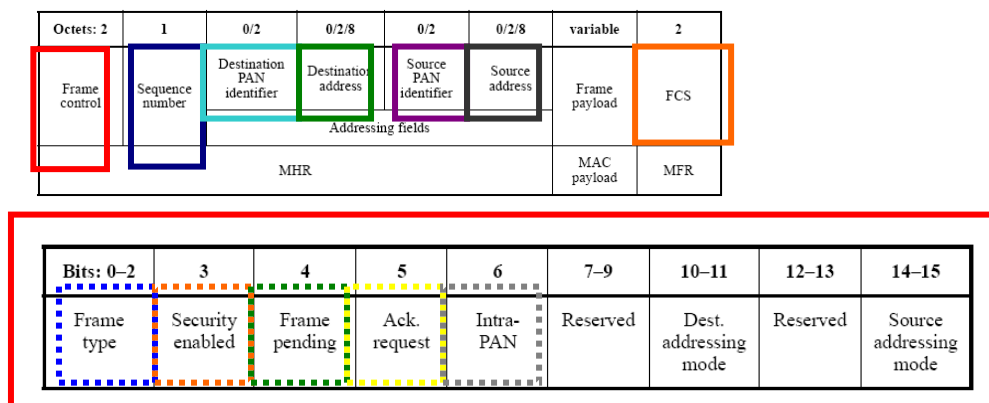


Figura 106 – Frame de Dados MAC descrito na Norma IEEE 802.15.4

Frame	Time(us)	Len	MAC Frame Control					Seq Num	Dest PAA	Dest Addr	Source PAA	Source Addr	FCS		
00130	+7632 =25928400	3	Type	Sec	Pend	ACK	IPAA						RSSI	Corr	CRC
			DATA	N	N	N	N	0x3D	0x3D15	0x0002	0x3D15	0x0000	-08	0x6B	OK

Figura 107 – Frame de Dados MAC monitorado em tempo real no Analisador ZENA

No Frame de dados da camada APS, o ZENA comprova a seqüência descrita na norma da Zigbee Alliance (Figura 108). No dado transmitido, o coordenador define o cluster, o profile e a fonte do end-point. A confirmação é vista no analisador ZENA (Figura 109) nas cores verde, azul-escuro e azul-claro contínuas, respectivamente (*Cluster ID =0x00, Profile Id=0x0000, Source EP=0x01*).

Octets: 1	0/1	0/1	0/2	0/1	Variable
Frame control	Destination end-point	Cluster Identifier	Profile Identifier	Source endpoint	Frame payload
Addressing fields					APS payload
APS header					

Bits: 0-1	2-3	4	5	6	7
Frame type	Delivery mode	Indirect address mode ^a	Security	Ack. request	Reserved

Figura 108 – Frame de Dados APS descrito na Norma Zigbee Alliance

APS Frame Control					Cluster ID	Profile ID	Source EP	AF Header	APS Payload	FCS			
Type	Deliv	Mode	Sec	ACK	ID	ID	EP	Cnt Type	0x10	0x00	RSSI	Corr	CRC
DAT	IND	To	N	N	0x00	0x0000	0x01	0x02 RES	0x00		+08	0x6C	OK

Figura 109 – Frame de Dados APS monitorado em tempo real no Analisador ZENA

O experimento 1 permite identificar:

- O protocolo Zigbee possui um sistema de sincronismo, denominado frame de beacon, gerenciado pelos dispositivos coordenadores para todos os dispositivos RFDs. Quando ativada esta função, todos os dispositivos RFDs buscam sincronizar com um coordenador de uma rede Zigbee, antes de começar a transmitir ou receber dados.

- A seqüência de transmissão da camada MAC para os frames de sincronismo (Beacon), frame de comando e frame de dados estão exatamente em conformidade com o descrito na norma IEEE 802.15.4.
- Uma única rede PAN é utilizada no experimento (*Intra PAN = N*). Todo controle da rede é realizado pelo dispositivo coordenador. O coordenador é responsável pela geração do sinal de sincronismo e ele determina a forma de acesso ao meio.
- O não uso de criptografia, identificado pelo Frame *Control*, *SEC = N*. O fabricante Microchip afirma não possuir recursos para criptografia, apesar do protocolo Zigbee permitir. Em pesquisa recente⁵, o fabricante Texas afirma ter recursos para uso de criptografia.
- O uso do CRC para checagem de erros. Apesar dos experimentos não demonstrarem como é feito o processo de checagem de erros, é visível o uso do sistema CRC (*FCS: CRC = ok*).
- O coordenador é o responsável em solicitar ou não a confirmação de dados dos RFDs (*Ack = N*). Também é de responsabilidade do coordenador, monitorar os dados a serem enviados aos RFDs, memorizando possíveis pendências (*Pend = N*). Os códigos do pacote demonstram que em todo momento não há pendência de dados a ser transmitido (*Pend = N*) e, também, não é solicitado pelo coordenador, confirmação de dados para o RFD (*Ack = N*).
- O coordenador utiliza os superframes pelo fabricante para sincronismo de uma rede. Na análise, permite um período para o CAP e outro para o GTS.
- A seqüência de transmissão da camada APL de um pacote de dados do coordenador para um dispositivo RFD está em acordo com a norma Zigbee Alliance.
- A rede possui endereços da camada MAC bem definida. Os endereços são adquiridos pelo fabricante associados a uma Indicação Única Organizada (OUI). Na figura 107, é possível ver um código único para a PAN, tendo a fonte (coordenador) e o destino (RFD) com códigos diferentes.
- O coordenador apresenta sub-endereços da camada APS. Os sub- endereços são criados pelo software ZENA (arquivo *myZigBee.c*). São identificados no experimento o perfil (0x0000) cluster (0x00), e o end point (0x01).

⁵ www.texas.com

6.3 EXPERIMENTO 2: PROTOCOLO ZIGBEE

6.3.1 Descrição do experimento – Relato da camada APS

O experimento 2 objetiva a transmissão de dados do dispositivo coordenador para o dispositivo RFD utilizando um perfil e cluster denominado iluminação. O dado transmitido é acionado pelos botões RB5 e RB4 do dispositivo coordenador, ilustrado na Figura 110.



Figura 110 – Botões RB5 e RB4 do dispositivo Coordenador

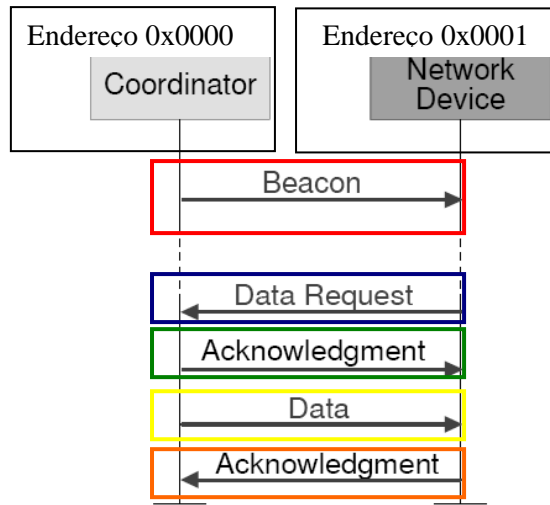
Após a execução das etapas de configuração descritas no experimento 1, o experimento 2 possui o seguinte princípio de funcionamento: é acionado um led (RA1) no dispositivo RFD por meio do botão RB4 do dispositivo coordenador. A Figura 111 ilustra o led (RA1) no dispositivo RFD.



Figura 111 – Led (RA1) do dispositivo RFD

Os formatos dos pacotes (frames) são monitorados pelo Analisador de espectro ZENA, comparando os dados coletados com a norma IEEE 802.15.4 e a norma Zigbee Alliance em tempo real.

A Figura 112 e 113 ilustram o modo de transferência de dados descrito na norma IEEE 802.15.4 e praticado no experimento pelo analisador de espectro ZENA.



**Figura 112 - Comunicação do coordenador para um dispositivo RFD (rede sinalizada):
Norma IEEE 802.15.4**

Frame	Time(us)	Len	MAC Frame Control	Seq Num	Dest PAN	Dest Addr	Beacon Request	FCS										
00002	+5354880	10	Type Sec Pend ACK IPAN CMD N N N N	0x01	0xFFFF	0xFFFF		RSSI Corr CRC +01 0x6C OK										
Frame	Time(us)	Len	MAC Frame Control	Seq Num	Source PAN	Source Addr	SuperFrame Specification			GTS Specification		PendAddr Spec						
00003	+1728	16	Type Sec Pend ACK IPAN BCN N N N N	0xF5	0x032B	0x0000	BO	SO	CAP	Batt	Coord	Assoc	Permit	Count	ExtAddr	ShortAddr		
							None	None	0x0	N	Y	Y	N	0x0	0x0	0x0		
Frame	Time(us)	Len	MAC Frame Control	Seq Num	Dest PAN	Dest Addr	Source PAN	Source Addr	Data Request	FCS								
00010	+542032	14	Type Sec Pend ACK IPAN CMD N N N Y N	0x04	0x032B	0x0000	0x032B	0x0001		RSSI	Corr	CRC						
										+04	0x6C	OK						
Frame	Time(us)	Len	MAC Frame Control	Seq Num	FCS													
00011	+944	5	Type Sec Pend ACK IPAN ACK N N N N	0x04	RSSI	Corr	CRC											
					-10	0x6B	OK											
Frame	Time(us)	Len	MAC Frame Control	Seq Num	Dest PAN	Dest Addr	Source Addr	HWK Header			APS Header		APS Payload		FCS			
00012	+13488	29	Type Sec Pend ACK IPAN DATA N N Y Y	0xF8	0x032B	0x0001	0x0000	0x04	0x00	0x01	0x00	0x04	0x12	0x13	0x32	0x11	0x00	
								0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xF0	-10	0x6C
Frame	Time(us)	Len	MAC Frame Control	Seq Num	FCS													
00013	+1744	5	Type Sec Pend ACK IPAN ACK N N N N	0xF8	RSSI	Corr	CRC											
					+04	0x6C	OK											

**Figura 113 - Comunicação do coordenador para um dispositivo RFD (rede sinalizada):
Analisador ZENA**

O frame de beacon é identificado no ZENA pelo conjunto de bytes contornados em vermelho. O coordenador é responsável em emitir um sinal de beacon.

O frame de comando, em cor azul-escuro, demonstra um pedido de dados de um dispositivo RFD.

A confirmação do entendimento pelo coordenador referente ao pedido de dados, denominado de frame de reconhecimento, é verificada na cor verde.

O frame de dado é transmitido, enviando um dado do coordenador para o RFD. A cor amarela capturada no ZENA comprova a ação.

A confirmação de recebimento de dado pelo RFD ocorre de acordo com os dados capturados no ZENA. É identificada pela cor laranja.

6.3.2 Detalhamento do Frame APS

O frame de dados da camada do APL descrito na norma Zigbee Alliance é comparado com dados coletados em tempo real pelo analisador ZENA ilustrado nas Figuras 114 e 115 respectivamente.

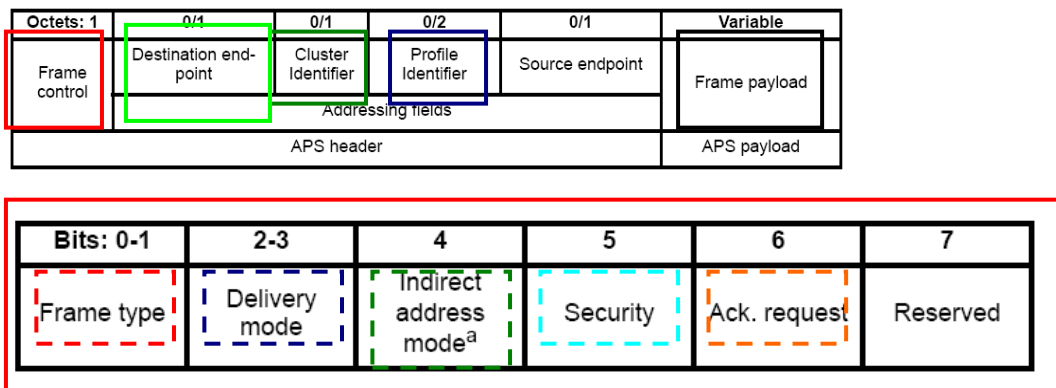


Figura 114 – Frame de Dados APL descrito na Norma Zigbee Alliance

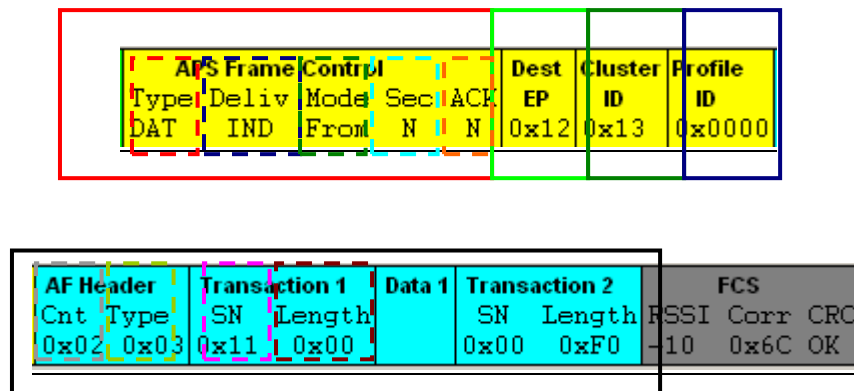


Figura 115 – Frame de Dados APL monitorado em tempo real no Analisador ZENA

O Frame Control faz abertura de um byte contendo 8 bits de configuração. Os bits 0-1 definem o tipo de frame. No ZENA é identificado o frame do tipo dados (*Type: DAT*).

Os bits 2-3 definem o modo de endereçamento (*mode Delivery*). Pelo ZENA é identificado o modo de endereçamento indireto, portanto o endereçamento da fonte do endpoint será omitido (*Deliv: IND*).

O bit 4 define o modo de endereçamento indireto (*Indirect address mode*). Se este subcampo for fixado em 1, o destinatário do endpoint estará omitido no frame, indicando uma transmissão indireta ao coordenador de ZigBee. Se este subcampo for fixado em 0, a fonte do endpoint estará omitida no frame. Pela identificação do ZENA identificamos o valor 0 (*Mode: From*).

O bit 5 define a solicitação de criptografia nos dados. A função está desabilitada pela informação identificada no ZENA (*Sec: N*).

O bit 6 define se o receptor necessita emitir um frame de reconhecimento (*acknowledgement*). No ZENA identificamos o frame de reconhecimento desligado (*ACK: N*).

O campo de destino do endpoint, apresentado na Figura 114, tem comprimento de 8 bits, especificando o endpoint do receptor. Até 240 objetos de aplicação (*Application Objects*) distintos podem ser definidos, cada um se conectará em um endpoint posicionado de 1 a 240. No ZENA é identificado o endereço 12 (*Dest EP : 0x12*).

O campo do identificador de agrupamento (*Cluster Identifier*) tem comprimento de 8 bits, especificando o identificador do cluster que deve ser usado na operação de ligamento (*binding*) no coordenador de ZigBee ou no dispositivo indicado pelo pedido (*request*). Tal campo é destinado para frames de dados, e não para um frame de comando. No ZENA é indicado o endereço 13 (*Cluster ID: 0x13*).

O Profile identifier (PI), ilustrado na Figura 114, tem um comprimento de dois bytes, especificando o identificador de perfil de ZigBee para qual o frame é pretendido. No ZENA é indicado o endereço 0 (*Profile ID: 0x0000*).

O protocolo Zigbee possui dois formatos de Frame: o frame formato chave de valor par KVP e o frame formato da mensagem (MSG). Ambos os formatos de armação são associados com um Agrupamento ID, mas frames KVP são designados para transferir um pedaço de informação associado com um atributo que usa uma estrutura rígida, enquanto frames MSG transferem informação que usa uma estrutura de forma flexível. O perfil da aplicação deixa especificado que formatos de frame devem ser usados para transferir informação.

Um frame KVP contém a seguinte informação:

- 1 - Contador de transação;
- 2 - Tipo do Frame; e
- 3 – Transação:
 - 3.1- Transação de número de sucessão
 - 3.2 - Tipo de comando e Tipo de Dados de Atributo
 - 3.3 - ID do atributo
 - 3.4 - Erro de código (opcional)
 - 3.5 - Dado do atributo (tamanho da variável)

Um frame MSG contém a seguinte informação:

- 1 - Contador de transação
- 2 - Tipo do Frame
- 3 - Transação
 - 3.1- Transação de número de sucessão
 - 3.2- Tamanho da transação
 - 3.3- Dado da transação

De acordo com informação monitorada pelo ZENA, o formato do frame é do tipo MSG, contendo a descrição:

- contador de transação (*Cnt: 0x02*),
- tipo de frame (*type: 0x03*),
- número de transação (*SN: 0x11*)
- tamanho da transação (*Length: 0x00*) e dado de transação.

Durante o experimento 2, em uma transmissão de um dado do dispositivo coordenador para o dispositivo RFD, é possível identificar a comunicação por meio da interface RS232 entre coordenador e computador. A mensagem comunica quando um nó é encontrado na rede (um dispositivo RFD). A Figura 116 ilustra a mensagem.

```
Trying to start network...
Now operating in next channel.
Network established.
A node has rejoined.
```

Figura 116 – Mensagem de formação da rede

Após a formação da rede, é possível enviar um comando para ligar o led. No momento da transmissão do dado (por meio do botão RB4 do coordenador), uma mensagem de transmissão de 'mensagem de luz' é indicada pela porta RS232. Em seguida é indicado o valor do atributo. Uma mensagem de confirmação de recebimento do RFD é apresentada por meio da porta RS232. A Figura 117 ilustra a mensagem.

```
Sending light message...
Sending switch value...
Switch MAC acknowledge received.
Light message done.
```

Figura 117 – Mensagem de envio e confirmação de dado (atributo) do coordenador para o RFD

O experimento 2 permite identificar:

- a camada APL do protocolo Zigbee possui um sistema de subendereço composto por: perfil, cluster, endpoint e atributo.
- O protocolo Zigbee possui dois formatos de Frame: o frame formato chave de valor par KVP e o frame formato da mensagem (MSG). Nos experimentos, foi comprovado o uso do formato MSG pelo fabricante Microchip.
- O tempo de resposta pode ser variável. Nos experimentos, o tempo de respostas varia de 0,5 a 1,5s entre o comando do coordenador e a resposta do dispositivo RFD.

- Mais uma vez, compra-se que o fabricante não possui recursos para uso de criptografia. Portanto o sinal não possui segurança nas informações.
- Nos experimentos, o sistema de comunicação sem fio da rede Zigbee funcionou com distâncias de até 80m sem barreiras. Acima desta distância, a rede demonstrou ser falha e com tempo de resposta acima dos 5s.

6.4 CONCLUSÃO

Os experimentos de laboratório atenderam as informações apresentadas na norma IEEE 802.15.4 e na norma Zigbee Alliance. Os pacotes de mensagens (frames) e suas características de endereçamentos e sincronismo (beacon) foram identificados pelos experimentos realizados com ferramentas adequadas de captura e análise.

O Frame de beacon é presente em uma rede sinalizada conforme descrita na norma IEEE. Tem a função de manter sincronizados os diferentes dispositivos constituintes da rede. O sinal é gerado pelo dispositivo coordenador. Nos experimentos o frame de beacon aparece apenas quando o dispositivo coordenador estiver ativado.

O Frame de comando pode ser emitido por dispositivos coordenador ou dispositivo final (RFD). Os tipos de comandos podem ser gerados por ambos os dispositivos, coordenador ou dispositivo final. O tipo de comando pode solicitar um pedido de dados, fazer uma indicação, solicitar um reconhecimento ou enviar uma resposta. Nos experimentos, o código do programa solicita um comando para pedido de dados.

Na análise dos experimentos, afirma-se o perfeito funcionamento do protocolo de comunicação sem fio Zigbee em conformidade com as normas IEEE 802.15.4 e a norma Zigbee Alliance.

O experimento desenvolvido com protocolo pode ser descrito como:

- Apresentou resposta em tempos satisfatórios em distâncias de até 80m sem barreiras utilizando como hardware o kit PICDEMZ. Acima desta distância, apresentou irregularidades.
- O tempo de resposta é alto, comparado a outros protocolos. Apesar do fabricante e da norma afirmar que o protocolo possui um tempo de resposta em milissegundos, é visível o tempo de resposta do protocolo na escala de segundos (1s em média).
- O protocolo não possui segurança dos dados. Apesar do protocolo permitir a inclusão de criptografia, o fabricante não possui o recurso.

- O sistema de endereçamento é complexo. Além do endereço da camada MAC, subendereços devem ser criados para a camada APL. Este conjunto de endereços, torna o sistema confiável, mas, ao mesmo tempo, lento. Uma informação de um dispositivo coordenador para um dispositivo RFD pode conter 127 bytes de informação.
- Os equipamentos para composição do hardware são simples. Um microcontrolador de 8 bits com alguns periféricos e uma antena de transmissão são necessários para a construção do hardware. Já o software é complexo, depende de arquivos pré-formatados e de um programador de subendereçamentos (nos experimentos denominados ZENA).
- O consumo de energia é uma característica positiva do protocolo. Para os experimentos foram utilizadas baterias de 9V que duraram 6 meses de experimentos, totalizando 4320h intermitente.
- As topologias do protocolo apesar de diversificado, são complexas. Nos experimentos apenas a topologia estrela foi testada.
- Os frames, ou pacotes de informação foram monitorados pelo analisador ZENA. Exatamente o descrito pelas normas IEEE e Zigbee Alliance foi identificado nos experimentos de laboratório. Importante destacar, que no caso deste fabricante, o frame de beacon é utilizado na programação como forma de sincronismo da rede.
- Os experimentos foram realizados com ferramentas e equipamentos licenciados pelos fabricantes.

7 CONCLUSÃO GERAL

O Zigbee é um protocolo de comunicação sem fio normalizado pelas normas IEEE 802.15.4 e Zigbee Alliance. É distribuído em 4 camadas das 7 possíveis do padrão OSI. As camadas físicas (PHY) e de enlace (MAC), são regulamentadas pela norma IEEE 802.15.4. As demais camadas de rede e aplicação (NWK e APL) são regulamentadas pela Zigbee Alliance.

No Brasil predomina a frequência portadora de 2,4 GHz com uma taxa de transmissão de 250Kbps.

O protocolo é construído em topologias como ponto a ponto, rede e estrela (a mais difundida). A formação da rede pode ser sincronizada ou não sincronizada. A rede sincronizada utiliza um frame de sincronismo em intervalos de tempo programado. Esta mensagem de sincronismo é denominada de frame de beacon.

Na camada de enlace do Zigbee encontra-se o identificador do dispositivo, conhecido como ID. Cada dispositivo possui um endereço composto de 64 bits, sendo 24 bits definidos como Indicação Única Organizada (OUI), adicionado de 40 bits nomeados pelo fabricante. Os OUIs podem ser comprados da IEEE para assegurar a singularidade global. Na camada de aplicação, cada dispositivo é subendereçoado com um perfil, um cluster e um atributo. Todos são incluídos dentro de um subendereço denominado endpoint.

Para o funcionamento do protocolo Zigbee, duas plataformas de processamento foram utilizadas. O dsPIC 30F4011 e o PIC 18LF4620.

O componente eletrônico dsPIC 30F4011 apresentou características compatíveis para o gerenciamento do protocolo de comunicação Zigbee, onde se destacam:

- Velocidade de processamento dos dados acima das necessidades mínimas para execução do protocolo Zigbee (30MIPS)
- Número e diversidade de periféricos acima do necessário para execução do protocolo Zigbee. Os únicos periféricos necessários são o UART e o protocolo SPI, ambos existentes no componente
- Possui o tipo de encapsulamento PDIP.
- Memória de programa de 48kbytes, bem acima da capacidade mínima necessária de 30kbytes
- Maior custo para implementação do protocolo Zigbee devido aos custos do componente dsPIC 30F4011.

- Complexidade para programação do código fonte. O fabricante Microchip não oferece suporte técnico para o protocolo Zigbee utilizando o componente dsPIC 30F4011.

O componente eletrônico PIC18F4620 apresentou desempenho satisfatório para execução do protocolo Zigbee, apresentando como características:

- Capacidade de memória de programa é suficiente para armazenar os códigos dos dispositivos RFDs e FFDs.
- Tecnologia nanowatt, operando em nível de tensão de 3.3V, uma característica necessária do protocolo Zigbee .
- Compilador C18, ferramenta de depuração do código, é consolidada no mercado e permite comunicar-se com a ferramenta Microchip Stack.
- Seu processamento apresenta capacidades de até 8 MIPS e memórias de programa de até 40 kB, necessárias para armazenar o código de programa do protocolo Zigbee.

Diante das duas propostas, o Zigbee pode ser processado por componente que apresente as seguintes características:

- processadores com capacidades de até 8 MIPS;
- memórias de programa de até 30kbytes.
- Deve possuir um conjunto mínimo de periféricos composto do sistema UART e do protocolo SPI.

Considerando o custo de implantação e as características técnicas do componente, o modelo dsPIC30F4011 é considerado inadequado para a implementação de uma rede de comunicação Zigbee. O componente utilizado nos estudos foi o PIC 18F4620.

Para a programação, gravação e monitoração do protocolo Zigbee, algumas ferramentas foram utilizadas, entre elas destacamos o Microchip Stack, o PICDEM™ Z e o ZENA.

A ferramenta Microchip Stack fornece um conjunto de códigos fonte de forma modular. É projetada para funcionar em parceria com o compilador MPLAB C18, mas pode ser modificada para suportar outros compiladores.

A ferramenta de desenvolvimento PICDEM™ Z é uma placa eletrônica que utiliza o protocolo Zigbee em microcontroladores da família Microchip PIC18Fxx para testes funcionais do protocolo. É composta de processador, transmissor/receptor de RF e periféricos.

O ZENA é uma ferramenta composta de um software e de um hardware. O software permite programar o protocolo Zigbee. O hardware permite analisar com detalhes a

montagem dos pacotes de mensagens das diferentes camadas do protocolo. Todas as ferramentas mostram-se eficientes e com funções bem delimitadas e de uso complexo.

Estas ferramentas foram utilizadas nos experimentos de laboratório, o qual atenderam satisfatoriamente as informações apresentadas na norma IEEE 802.15.4 e na norma Zigbee Alliance e permitiram conclusões referente ao protocolo Zigbee, entre elas:

Os pacotes de mensagens (frames) e suas características de endereçamentos e sincronismo (beacon) foram identificados pelos experimentos realizados com ferramentas de captura ZENA.

O Frame de beacon é presente em uma rede sinalizada conforme descrita na norma IEEE. O sinal é gerado apenas pelo dispositivo coordenador.

Os Frames de dados da camada APL apresentaram subendereçamentos conforme descrito na norma Zigbee Alliance. Os subendereçamentos apresentaram-se nos experimentos na ordem:

- Destino do End-point;
- Identificador do Cluster;
- Identificador do Profile;
- Fonte do End-point; e
- o valor do dado, denominado de atributo.

O protocolo Zigbee apresentou:

- Resposta em tempos satisfatórios em distâncias de até 80m sem barreiras utilizando como hardware o kit PICDEMZ.
- Não possui segurança dos dados. A criptografia é um código ainda não implementado por este fabricante.
- Endereço da camada MAC e subendereços da camada APL. Este conjunto de endereços pode conter 127 bytes de informação.
- Os equipamentos para composição do protocolo são: um microcontrolador de 8 bits com alguns periféricos, uma antena de transmissão/recepção e um código fonte;
- O consumo de energia é baixo. Para os experimentos foram utilizados baterias de 9V que duraram 6 meses de experimentos, totalizando 4320h intermitente.
- O frame de sincronização (beacon) é utilizado na programação como forma de sincronismo da rede.

Os estudos referentes ao protocolo Zigbee, podem ser complementados com estudo e experimento de algumas características, entre elas:

O estudo e ensaio de dispositivos em topologias de estrela e rede, utilizando mais de dois dispositivos;

A formação de redes com uso do dispositivo roteador, responsável também pela expansão física da rede;

A criptografia de sinal. Por tratar-se de uma rede sem fio, o estudo e utilização de uma rede criptografada (dependendo do objetivo) pode ser necessária;

Em síntese é possível afirmar que, o uso do protocolo ZigBee, não é recomendado para controle de processos em malha fechada que necessitam de resposta altas. Seu tempo de resposta, apesar de confiável, não é suficiente para controles como posição, velocidade, contagem entre outros. Entretanto, é possível utilizá-lo em sistemas de controle, independente de ser malha aberta ou fechada, que respondem a velocidades baixas como controle de iluminação, temperatura, umidade, nível de água, comandos discretos e outros.

REFERÊNCIAS

AMERICAS Corporate Office. **Data sheet dsPIC 30F6014**. USA: Microchip Technology Incorporated, 2005.

AMERICAS Corporate Office. **Data sheet dsPIC 30F4011**. USA: Microchip Technology Incorporated, 2005.

AMERICAS Corporate Office. **Data sheet PIC 18F4620**. USA: Microchip Technology Incorporated, 2001.

FALBRIORD, Claude. **Protocolo e aplicações de redes de computadores**. São Paulo: Érica, 2003.

HAAG, Marcelo Tiago. **Estudo de redes pessoais sem fio de curta distância padrão IEEE 802.15**: 2006. 112pg. (Redes Industriais) – Universidade Regional de Blumenau, FURB, Blumenau, 2006.

IDOETA, Ivan Valeije; CAPUANO, Francisco Gabriel. **Elementos de Eletrônica Digital**. 28. ed. São Paulo: Érica, 1998.

LOPES, Ricardo A. **Sistema de redes para controle e automação**. Rio de Janeiro: Book Express, 2000.

MICROCHIP Technology. **PICDEM™ Z Demonstration Kit User's Guide**. USA, 2006.

MICROCHIP Technology Incorporated. **ZENA™ Wireless Network Analyzer User's Guide**. USA, 2007.

MORAIS, Alexandre Fernandes de; CIRONE, Antonio Carlos. **Redes de computadores da ethernet a internet**. São Paulo: Érica, 2003.

OLIVEIRA, André Schneider de; ANDRADE, Fernando Souza de. **Sistemas embarcados: hardware e firmware na prática**. São Paulo: Érica, 2006. 320p.

PEREIRA, Fábio. **Microcontroladores PIC: programação em C**. São Paulo: Érica, 2002.

SEVERINO, Antonio Joaquim. **Metodologia do Trabalho Científico**. São Paulo: Cortez, 2002.

SILVA JUNIOR, Vidal Pereira da. **Aplicações e Práticas do Microcontrolador 8051**. São Paulo: Érica, 1994.

SOUZA, David José de. **Recursos Avançados 16F877**. São Paulo: Érica, 2000.
THE INSTITUTE of Electrical and Electronics Engineers. **IEEE 802.15.4**. New York: IEEE, 2003.

THE INSTITUTE of Electrical and Electronics Engineers. **ZigBee Specification**. New York: IEEE, 2003.

TOKHEIM, Roger L. **Introdução aos Microprocessadores**. Tradução de: Anatólio Laschuk e Fernando Fontes Barbosa. São Paulo: McGraw-Hill do Brasil, 1985.

ZIGBEE Alliance. **Zigbee**. USA, 2004.

Sítios:

- <http://www.microchip.com>
- <http://www.ieee.org>
- <http://www.zigbee.org>
- <http://www.texas.com>
- <http://www.freescale.com>

ANEXO 1 – REGISTRADOR DE ESTADO – STATUS

REGISTER 5-2: STATUS REGISTER

	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
	—	—	—	N	OV	Z	DC	C
bit 7								bit 0

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **N:** Negative bit

This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).

1 = Result was negative

0 = Result was positive

bit 3 **OV:** Overflow bit

This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7 of the result) to change state.

1 = Overflow occurred for signed arithmetic (in this arithmetic operation)

0 = No overflow occurred

bit 2 **Z:** Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/borrow bit

For ADDWF, ADDLW, SUBLW and SUBWF instructions:

1 = A carry-out from the 4th low-order bit of the result occurred

0 = No carry-out from the 4th low-order bit of the result

Note: For $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.

bit 0 **C:** Carry/borrow bit

For ADDWF, ADDLW, SUBLW and SUBWF instructions:

1 = A carry-out from the Most Significant bit of the result occurred

0 = No carry-out from the Most Significant bit of the result occurred

Note: For $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low-order bit of the source register.

ANEXO 2 – REGISTRADOR DE CONTROLE DA MEMÓRIA EEPROM

REGISTER 6-1: EECON1: DATA EEPROM CONTROL REGISTER 1

	R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
bit 7								bit 0

- bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
 1 = Access Flash program memory
 0 = Access data EEPROM memory
- bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
 1 = Access Configuration registers
 0 = Access Flash program or data EEPROM memory
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **FREE:** Flash Row Erase Enable bit
 1 = Erase the program memory row addressed by TBLPTR on the next WR command
 (cleared by completion of erase operation)
 0 = Perform write only
- bit 3 **WRERR:** Flash Program/Data EEPROM Error Flag bit
 1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation, or an improper write attempt)
 0 = The write operation completed
 Note: When a WRERR occurs, the EEGPD and CFGS bits are not cleared.
 This allows tracing of the error condition.
- bit 2 **WREN:** Flash Program/Data EEPROM Write Enable bit
 1 = Allows write cycles to Flash program/data EEPROM
 0 = Inhibits write cycles to Flash program/data EEPROM
- bit 1 **WR:** Write Control bit
 1 = Initiates a data EEPROM erase/write cycle or a program memory erase/write cycle.
 (The operation is self-timed and the bit is cleared by hardware once write is complete.
 The WR bit can only be set (not cleared) in software.)
 0 = Write cycle to the EEPROM is complete
- bit 0 **RD:** Read Control bit
 1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEGPD = 1 or CFGS = 1.)
 0 = Does not initiate an EEPROM read

Legend:			
R = Readable bit	W = Writable bit		
S = Bit can be set by software, but not cleared	U = Unimplemented bit, read as '0'		
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

ANEXO 3 – CÓDIGO FONTE EM ASSEMBLY PARA ROTINA DE MULTIPLICAÇÃO 16 X16 BITS

EXAMPLE 8-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```

MOVWF ARG1L, W
MULWF ARG2L      ; ARG1L * ARG2L->
                  ; PRODH:PRODL

MOVWF PRODH, RES1 ;
MOVWF PRODL, RES0 ;

;

MOVWF ARG1H, W
MULWF ARG2H      ; ARG1H * ARG2H->
                  ; PRODH:PRODL

MOVWF PRODH, RES3 ;
MOVWF PRODL, RES2 ;

;

MOVWF ARG1L, W
MULWF ARG2H      ; ARG1L * ARG2H->
                  ; PRODH:PRODL

MOVWF PRODL, W   ;
ADDWF RES1, F    ; Add cross
MOVWF PRODH, W   ; products
ADDWFC RES2, F   ;
CLRF WREG        ;
ADDWFC RES3, F   ;

;

MOVWF ARG1H, W   ;
MULWF ARG2L      ; ARG1H * ARG2L->
                  ; PRODH:PRODL

MOVWF PRODL, W   ;
ADDWF RES1, F    ; Add cross
MOVWF PRODH, W   ; products
ADDWFC RES2, F   ;
CLRF WREG        ;
ADDWFC RES3, F   ;

```

ANEXO 4 – PORTA

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RA0/AN0	RA0	0	O	DIG	LATA<0> data output; not affected by analog input.
		1	I	TTL	PORTA<0> data input; disabled when analog input enabled.
	AN0	1	I	ANA	A/D input channel 0 and Comparator C1- input. Default input configuration on POR; does not affect digital output.
RA1/AN1	RA1	0	O	DIG	LATA<1> data output; not affected by analog input.
		1	I	TTL	PORTA<1> data input; disabled when analog input enabled.
	AN1	1	I	ANA	A/D input channel 1 and Comparator C2- input. Default input configuration on POR; does not affect digital output.
RA2/AN2/ VREF-/CVREF	RA2	0	O	DIG	LATA<2> data output; not affected by analog input. Disabled when CVREF output enabled.
		1	I	TTL	PORTA<2> data input. Disabled when analog functions enabled; disabled when CVREF output enabled.
	AN2	1	I	ANA	A/D input channel 2 and Comparator C2+ input. Default input configuration on POR; not affected by analog output.
	VREF-	1	I	ANA	A/D and comparator voltage reference low input.
	CVREF	x	O	ANA	Comparator voltage reference output. Enabling this feature disables digital I/O.
RA3/AN3/VREF+	RA3	0	O	DIG	LATA<3> data output; not affected by analog input.
		1	I	TTL	PORTA<3> data input; disabled when analog input enabled.
	AN3	1	I	ANA	A/D input channel 3 and Comparator C1+ input. Default input configuration on POR.
	VREF+	1	I	ANA	A/D and comparator voltage reference high input.
RA4/T0CKI/C1OUT	RA4	0	O	DIG	LATA<4> data output.
		1	I	ST	PORTA<4> data input; default configuration on POR.
	T0CKI	1	I	ST	Timer0 clock input.
	C1OUT	0	O	DIG	Comparator 1 output; takes priority over port data.
RA5/AN4/SS/ HLVDIN/C2OUT	RA5	0	O	DIG	LATA<5> data output; not affected by analog input.
		1	I	TTL	PORTA<5> data input; disabled when analog input enabled.
	AN4	1	I	ANA	A/D input channel 4. Default configuration on POR.
	SS	1	I	TTL	Slave Select input for SSP (MSSP module).
	HLVDIN	1	I	ANA	High/Low-Voltage Detect external trip point input.
C2OUT	0	O	DIG	Comparator 2 output; takes priority over port data.	
OSC2/CLKO/RA6	RA6	0	O	DIG	LATA<6> data output. Enabled in RCIO, INTIO2 and ECIO modes only.
		1	I	TTL	PORTA<6> data input. Enabled in RCIO, INTIO2 and ECIO modes only.
	OSC2	x	O	ANA	Main oscillator feedback output connection (XT, HS and LP modes).
	CLKO	x	O	DIG	System cycle clock output (Fosc/4) in RC, INTIO1 and EC Oscillator modes.
OSC1/CLKI/RA7	RA7	0	O	DIG	LATA<7> data output. Disabled in external oscillator modes.
		1	I	TTL	PORTA<7> data input. Disabled in external oscillator modes.
	OSC1	x	I	ANA	Main oscillator input connection.
	CLKI	x	I	ANA	Main clock input connection.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

ANEXO 5 – PORTB

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RB0/INT0/FLT0/AN12	RB0	0	O	DIG	LATB<0> data output; not affected by analog input.
		1	I	TTL	PORTB<0> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled. ⁽¹⁾
	INT0	1	I	ST	External interrupt 0 input.
	FLT0	1	I	ST	Enhanced PWM Fault input (ECCP1 module); enabled in software.
	AN12	1	I	ANA	A/D input channel 12. ⁽¹⁾
RB1/INT1/AN10	RB1	0	O	DIG	LATB<1> data output; not affected by analog input.
		1	I	TTL	PORTB<1> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled. ⁽¹⁾
	INT1	1	I	ST	External interrupt 1 input.
	AN10	1	I	ANA	A/D input channel 10. ⁽¹⁾
RB2/INT2/AN8	RB2	0	O	DIG	LATB<2> data output; not affected by analog input.
		1	I	TTL	PORTB<2> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled. ⁽¹⁾
	INT2	1	I	ST	External interrupt 2 input.
	AN8	1	I	ANA	A/D input channel 8. ⁽¹⁾
RB3/AN9/CCP2	RB3	0	O	DIG	LATB<3> data output; not affected by analog input.
		1	I	TTL	PORTB<3> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled. ⁽¹⁾
	AN9	1	I	ANA	A/D input channel 9. ⁽¹⁾
	CCP2 ⁽²⁾	0	O	DIG	CCP2 compare and PWM output.
		1	I	ST	CCP2 capture input.
RB4/KBI0/AN11	RB4	0	O	DIG	LATB<4> data output; not affected by analog input.
		1	I	TTL	PORTB<4> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled. ⁽¹⁾
	KBI0	1	I	TTL	Interrupt on pin change.
	AN11	1	I	ANA	A/D input channel 11. ⁽¹⁾
RB5/KBI1/PGM	RB5	0	O	DIG	LATB<5> data output.
		1	I	TTL	PORTB<5> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI1	1	I	TTL	Interrupt on pin change.
	PGM	x	I	ST	Single-Supply Programming mode entry (ICSP™). Enabled by LVP configuration bit; all other pin functions disabled.
RB6/KBI2/PGC	RB6	0	O	DIG	LATB<6> data output.
		1	I	TTL	PORTB<6> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI2	1	I	TTL	Interrupt on pin change.
	PGC	x	I	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation. ⁽³⁾
RB7/KBI3/PGD	RB7	0	O	DIG	LATB<7> data output.
		1	I	TTL	PORTB<7> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI3	1	I	TTL	Interrupt on pin change.
	PGD	x	O	DIG	Serial execution data output for ICSP and ICD operation. ⁽³⁾
		x	I	ST	Serial execution data input for ICSP and ICD operation. ⁽³⁾

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Configuration on POR is determined by the PBDEN configuration bit. Pins are configured as analog inputs by default when PBDEN is set and digital inputs when PBDEN is cleared.

2: Alternate assignment for CCP2 when the CCP2MX configuration bit is '0'. Default assignment is RC1.

3: All other pin functions are disabled when ICSP or ICD are enabled.

ANEXO 6 – PORTC

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RC0/T1OSO/T13CKI	RC0	0	O	DIG	LATC<0> data output.
		1	I	ST	PORTC<0> data input.
	T1OSO	x	O	ANA	Timer1 oscillator output; enabled when Timer1 oscillator enabled. Disables digital I/O.
	T13CKI	1	I	ST	Timer1/Timer3 counter input.
RC1/T1OSI/CCP2	RC1	0	O	DIG	LATC<1> data output.
		1	I	ST	PORTC<1> data input.
	T1OSI	x	I	ANA	Timer1 oscillator input; enabled when Timer1 oscillator enabled. Disables digital I/O.
	CCP2 ⁽¹⁾	0	O	DIG	CCP2 compare and PWM output; takes priority over port data.
		1	I	ST	CCP2 capture input.
RC2/CCP1/P1A	RC2	0	O	DIG	LATC<2> data output.
		1	I	ST	PORTC<2> data input.
	CCP1	0	O	DIG	ECCP1 compare or PWM output; takes priority over port data.
		1	I	ST	ECCP1 capture input.
	P1A ⁽²⁾	0	O	DIG	ECCP1 Enhanced PWM output, channel A. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
	RC3/SCK/SCL	RC3	0	O	DIG
1			I	ST	PORTC<3> data input.
SCK		0	O	DIG	SPI™ clock output (MSSP module); takes priority over port data.
		1	I	ST	SPI clock input (MSSP module).
SCL		0	O	DIG	I ² C™ clock output (MSSP module); takes priority over port data.
		1	I	I ² C/SMB	I ² C clock input (MSSP module); input type depends on module setting.
RC4/SDI/SDA	RC4	0	O	DIG	LATC<4> data output.
		1	I	ST	PORTC<4> data input.
	SDI	1	I	ST	SPI data input (MSSP module).
	SDA	1	O	DIG	I ² C data output (MSSP module); takes priority over port data.
		1	I	I ² C/SMB	I ² C data input (MSSP module); input type depends on module setting.
RC5/SDO	RC5	0	O	DIG	LATC<5> data output.
		1	I	ST	PORTC<5> data input.
	SDO	0	O	DIG	SPI data output (MSSP module); takes priority over port data.
RC6/TX/CK	RC6	0	O	DIG	LATC<6> data output.
		1	I	ST	PORTC<6> data input.
	TX	1	O	DIG	Asynchronous serial transmit data output (USART module); takes priority over port data. User must configure as output.
		1	I	ST	Synchronous serial clock input (USART module).
RC7/RX/DT	RC7	0	O	DIG	LATC<7> data output.
		1	I	ST	PORTC<7> data input.
	RX	1	I	ST	Asynchronous serial receive data input (USART module).
	DT	1	O	DIG	Synchronous serial data output (USART module); takes priority over port data.
		1	I	ST	Synchronous serial data input (USART module). User must configure as an input.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I²C/SMB = I²C/SMBus input buffer; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Default assignment for CCP2 when the CCP2MX configuration bit is set. Alternate assignment is RB3.

2: Enhanced PWM output is available only on PIC18F4525/4620 devices.

ANEXO 7 – PORTD

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RD0/PSP0	RD0	0	O	DIG	LATD<0> data output.
		1	I	ST	PORTD<0> data input.
	PSP0	x	O	DIG	PSP read data output (LATD<0>); takes priority over port data.
		x	I	TTL	PSP write data input.
RD1/PSP1	RD1	0	O	DIG	LATD<1> data output.
		1	I	ST	PORTD<1> data input.
	PSP1	x	O	DIG	PSP read data output (LATD<1>); takes priority over port data.
		x	I	TTL	PSP write data input.
RD2/PSP2	RD2	0	O	DIG	LATD<2> data output.
		1	I	ST	PORTD<2> data input.
	PSP2	x	O	DIG	PSP read data output (LATD<2>); takes priority over port data.
		x	I	TTL	PSP write data input.
RD3/PSP3	RD3	0	O	DIG	LATD<3> data output.
		1	I	ST	PORTD<3> data input.
	PSP3	x	O	DIG	PSP read data output (LATD<3>); takes priority over port data.
		x	I	TTL	PSP write data input.
RD4/PSP4	RD4	0	O	DIG	LATD<4> data output.
		1	I	ST	PORTD<4> data input.
	PSP4	x	O	DIG	PSP read data output (LATD<4>); takes priority over port data.
		x	I	TTL	PSP write data input.
RD5/PSP5/P1B	RD5	0	O	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	PSP5	x	O	DIG	PSP read data output (LATD<5>); takes priority over port data.
		x	I	TTL	PSP write data input.
P1B	0	O	DIG	ECCP1 Enhanced PWM output, channel B; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.	
RD6/PSP6/P1C	RD6	0	O	DIG	LATD<6> data output.
		1	I	ST	PORTD<6> data input.
	PSP6	x	O	DIG	PSP read data output (LATD<6>); takes priority over port data.
		x	I	TTL	PSP write data input.
P1C	0	O	DIG	ECCP1 Enhanced PWM output, channel C; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.	
RD7/PSP7/P1D	RD7	0	O	DIG	LATD<7> data output.
		1	I	ST	PORTD<7> data input.
	PSP7	x	O	DIG	PSP read data output (LATD<7>); takes priority over port data.
		x	I	TTL	PSP write data input.
P1D	0	O	DIG	ECCP1 Enhanced PWM output, channel D; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.	

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

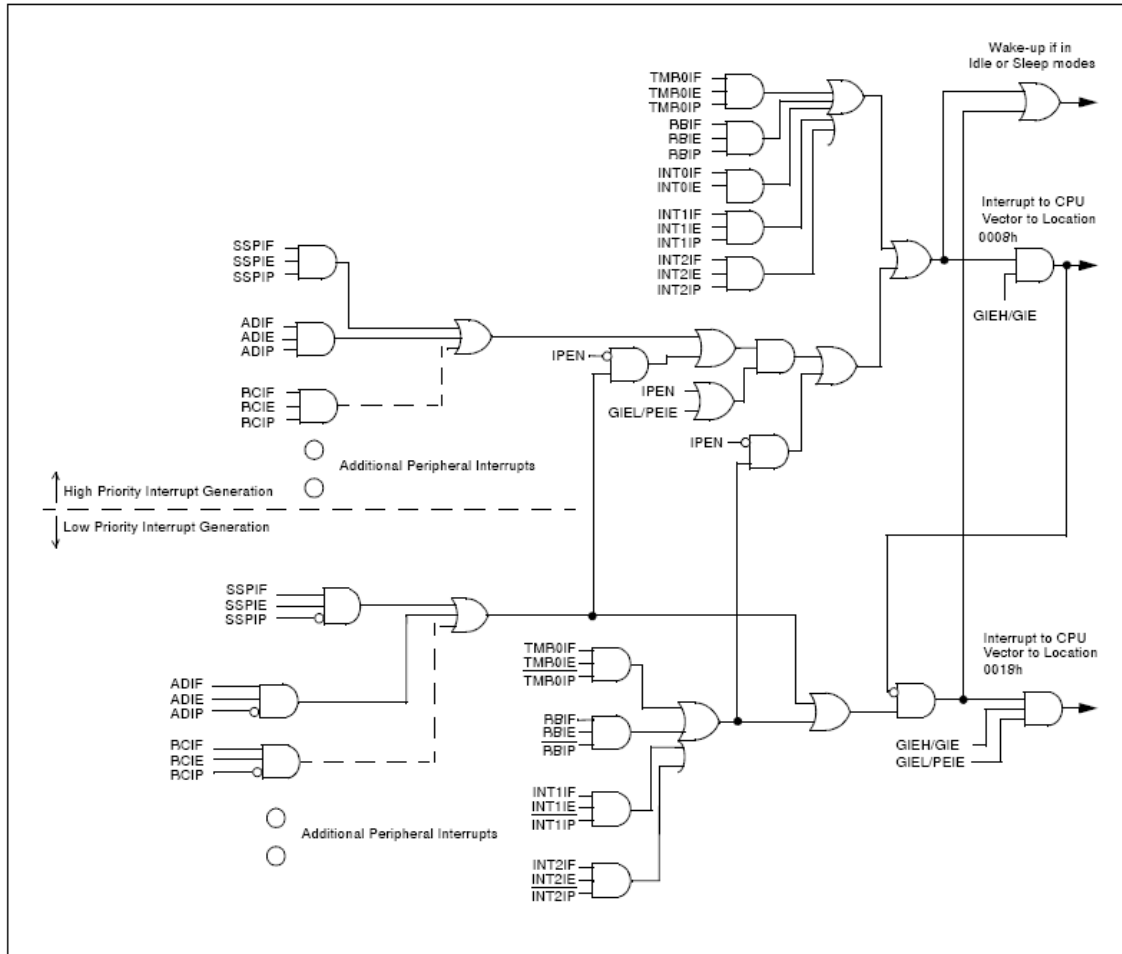
ANEXO 8 – PORTE

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RE0/ $\overline{\text{RD}}$ /AN5	RE0	0	O	DIG	LATE<0> data output; not affected by analog input.
		1	I	ST	PORTE<0> data input; disabled when analog input enabled.
	$\overline{\text{RD}}$	1	I	TTL	PSP read enable input (PSP enabled).
	AN5	1	I	ANA	A/D input channel 5; default input configuration on POR.
RE1/ $\overline{\text{WR}}$ /AN6	RE1	0	O	DIG	LATE<1> data output; not affected by analog input.
		1	I	ST	PORTE<1> data input; disabled when analog input enabled.
	$\overline{\text{WR}}$	1	I	TTL	PSP write enable input (PSP enabled).
	AN6	1	I	ANA	A/D input channel 6; default input configuration on POR.
RE2/ $\overline{\text{CS}}$ /AN7	RE2	0	O	DIG	LATE<2> data output; not affected by analog input.
		1	I	ST	PORTE<2> data input; disabled when analog input enabled.
	$\overline{\text{CS}}$	1	I	TTL	PSP write enable input (PSP enabled).
	AN7	1	I	ANA	A/D input channel 7; default input configuration on POR.
$\overline{\text{MCLR}}$ / $\overline{\text{VPP}}$ /RE3 ⁽¹⁾	$\overline{\text{MCLR}}$	—	I	ST	External Master Clear input; enabled when MCLRE configuration bit is set.
	$\overline{\text{VPP}}$	—	I	ANA	High-voltage detection; used for ICSP™ mode entry detection. Always available, regardless of pin mode.
	RE3	— ⁽²⁾	I	ST	PORTE<3> data input; enabled when MCLRE configuration bit is clear.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: RE3 is available on both 28-pin and 40/44-pin devices. All other PORTE pins are only implemented on 40/44-pin devices.
2: RE3 does not have a corresponding TRIS bit to control data direction.

ANEXO 9 – FONTES DE INTERRUPTÃO



ANEXO 10 – REGISTRADOR INTCON

REGISTER 10-1: INTCON: INTERRUPT CONTROL REGISTER

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7								bit 0

- bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
 1 = Enables all unmasked interrupts
 0 = Disables all interrupts
When IPEN = 1:
 1 = Enables all high priority interrupts
 0 = Disables all interrupts
- bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
 1 = Enables all unmasked peripheral interrupts
 0 = Disables all peripheral interrupts
When IPEN = 1:
 1 = Enables all low priority peripheral interrupts
 0 = Disables all low priority peripheral interrupts
- bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
 1 = Enables the TMR0 overflow interrupt
 0 = Disables the TMR0 overflow interrupt
- bit 4 **INT0IE:** INT0 External Interrupt Enable bit
 1 = Enables the INT0 external interrupt
 0 = Disables the INT0 external interrupt
- bit 3 **RBIE:** RB Port Change Interrupt Enable bit
 1 = Enables the RB port change interrupt
 0 = Disables the RB port change interrupt
- bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
 1 = TMR0 register has overflowed (must be cleared in software)
 0 = TMR0 register did not overflow
- bit 1 **INT0IF:** INT0 External Interrupt Flag bit
 1 = The INT0 external interrupt occurred (must be cleared in software)
 0 = The INT0 external interrupt did not occur
- bit 0 **RBIF:** RB Port Change Interrupt Flag bit
 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
 0 = None of the RB7:RB4 pins have changed state

Note: A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

ANEXO 11 – REGISTRADOR INTCON2

REGISTER 10-2: INTCON2: INTERRUPT CONTROL REGISTER 2

	R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
	RBP \overline{U}	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7								bit 0

- bit 7 **RBP \overline{U}** : PORTB Pull-up Enable bit
 1 = All PORTB pull-ups are disabled
 0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG0**: External Interrupt 0 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 5 **INTEDG1**: External Interrupt 1 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 4 **INTEDG2**: External Interrupt 2 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 3 **Unimplemented**: Read as '0'
- bit 2 **TMR0IP**: TMR0 Overflow Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 1 **Unimplemented**: Read as '0'
- bit 0 **RBIP**: RB Port Change Interrupt Priority bit
 1 = High priority
 0 = Low priority

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

ANEXO 12 – REGISTRADOR INTCON3

REGISTER 10-3: INTCON3: INTERRUPT CONTROL REGISTER 3

	R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF

bit 7

bit 0

- bit 7 **INT2IP:** INT2 External Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 6 **INT1IP:** INT1 External Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **INT2IE:** INT2 External Interrupt Enable bit
 1 = Enables the INT2 external interrupt
 0 = Disables the INT2 external interrupt
- bit 3 **INT1IE:** INT1 External Interrupt Enable bit
 1 = Enables the INT1 external interrupt
 0 = Disables the INT1 external interrupt
- bit 2 **Unimplemented:** Read as '0'
- bit 1 **INT2IF:** INT2 External Interrupt Flag bit
 1 = The INT2 external interrupt occurred (must be cleared in software)
 0 = The INT2 external interrupt did not occur
- bit 0 **INT1IF:** INT1 External Interrupt Flag bit
 1 = The INT1 external interrupt occurred (must be cleared in software)
 0 = The INT1 external interrupt did not occur

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

ANEXO 13 - REGISTRADOR SSPSTAT

REGISTER 17-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/ \bar{A}	P	S	R/ \bar{W}	UA	BF
bit 7							bit 0

- bit 7 **SMP:** Sample bit
SPI Master mode:
 1 = Input data sampled at end of data output time
 0 = Input data sampled at middle of data output time
SPI Slave mode:
 SMP must be cleared when SPI is used in Slave mode.
- bit 6 **CKE:** SPI Clock Select bit
 1 = Transmit occurs on transition from active to Idle clock state
 0 = Transmit occurs on transition from Idle to active clock state
Note: Polarity of clock state is set by the CKP bit (SSPCON1<4>).
- bit 5 **D/ \bar{A} :** Data/Address bit
 Used in I²C mode only.
- bit 4 **P:** Stop bit
 Used in I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.
- bit 3 **S:** Start bit
 Used in I²C mode only.
- bit 2 **R/ \bar{W} :** Read/Write Information bit
 Used in I²C mode only.
- bit 1 **UA:** Update Address bit
 Used in I²C mode only.
- bit 0 **BF:** Buffer Full Status bit (Receive mode only)
 1 = Receive complete, SSPBUF is full
 0 = Receive not complete, SSPBUF is empty

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

ANEXO 14- REGISTRADOR SSPCON1

REGISTER 17-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7								bit 0

- bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)
 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
 0 = No collision
- bit 6 **SSPOV:** Receive Overflow Indicator bit
SPI Slave mode:
 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
 0 = No overflow
Note: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit
 1 = Enables serial port and configures SCK, SDO, SDI and \overline{SS} as serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
Note: When enabled, these pins must be properly configured as input or output.
- bit 4 **CKP:** Clock Polarity Select bit
 1 = Idle state for clock is a high level
 0 = Idle state for clock is a low level
- bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits
 0101 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control disabled, \overline{SS} can be used as I/O pin
 0100 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control enabled
 0011 = SPI Master mode, clock = TMR2 output/2
 0010 = SPI Master mode, clock = Fosc/64
 0001 = SPI Master mode, clock = Fosc/16
 0000 = SPI Master mode, clock = Fosc/4
Note: Bit combinations not specifically listed here are either reserved or implemented in I²C mode only.

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

ANEXO 15- REGISTRADOR OSCCON E OSCTUNE

REGISTER 2-1: OSCTUNE: OSCILLATOR TUNING REGISTER

R/W-0	R/W-0 ⁽¹⁾	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
INTSRC	PLLEN ⁽¹⁾	—	TUN4	TUN3	TUN2	TUN1	TUN0	
bit 7								bit 0

- bit 7 **INTSRC:** Internal Oscillator Low-Frequency Source Select bit
 1 = 31.25 kHz device clock derived from 8 MHz INTOSC source (divide-by-256 enabled)
 0 = 31 kHz device clock derived directly from INTRC internal oscillator
- bit 6 **PLLEN:** Frequency Multiplier PLL for INTOSC Enable bit⁽¹⁾
 1 = PLL enabled for INTOSC (4 MHz and 8 MHz only)
 0 = PLL disabled
- Note 1:** Available only in certain oscillator configurations; otherwise, this bit is unavailable and reads as '0'. See Section 2.6.4 "PLL in INTOSC Modes" for details.
- bit 5 **Unimplemented:** Read as '0'
- bit 4-0 **TUN4:TUN0:** Frequency Tuning bits
 01111 = Maximum frequency
 • •
 • •
 00001
 00000 = Center frequency. Oscillator module is running at the calibrated frequency.
 11111
 • •
 • •
 10000 = Minimum frequency

REGISTER 2-2: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R/W-1	R/W-0	R/W-0	R ⁽¹⁾	R-0	R/W-0	R/W-0	
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0	
bit 7								bit 0

- bit 7 **IDLEN:** Idle Enable bit
 1 = Device enters Idle mode on SLEEP instruction
 0 = Device enters Sleep mode on SLEEP instruction
- bit 6-4 **IRCF2:IRCF0:** Internal Oscillator Frequency Select bits
 111 = 8 MHz (INTOSC drives clock directly)
 110 = 4 MHz
 101 = 2 MHz
 100 = 1 MHz⁽³⁾
 011 = 500 kHz
 010 = 250 kHz
 001 = 125 kHz
 000 = 31 kHz (from either INTOSC/256 or INTRC directly)⁽²⁾
- bit 3 **OSTS:** Oscillator Start-up Time-out Status bit⁽¹⁾
 1 = Oscillator start-up time-out timer has expired; primary oscillator is running
 0 = Oscillator start-up time-out timer is running; primary oscillator is not ready
- bit 2 **IOFS:** INTOSC Frequency Stable bit
 1 = INTOSC frequency is stable
 0 = INTOSC frequency is not stable
- bit 1-0 **SCS1:SCS0:** System Clock Select bits
 1x = Internal oscillator block
 01 = Secondary (Timer1) oscillator
 00 = Primary oscillator

- Note 1:** Reset state depends on state of the IESO configuration bit.
2: Source selected by the INTSRC bit (OSCTUNE<7>), see text.
3: Default output frequency of INTOSC on Reset.

ANEXO 16 - PASSOS PARA INSTALAR OS ARQUIVOS DO MICROCHIP STACK

1. Execute a instalação do arquivo. Um sistema operacional de instalação Windows wizard o guiará pelo processo de instalação.
2. Depois de instalado o software, você tem que aceitar o acordo de licença de software clicando “I Accept”.
3. Se você deseja instalar os arquivos de GERBER para o PICDEMZ demonstrativo, você tem que aceitar um segundo acordo de licença. Se você escolher não aceitar este acordo, você ainda pode instalar a ferramenta Microchip Stack.
4. Depois de completar o processo de instalação, você deve ver o “Microchip Stack for ZigBee” grupo de código fonte do protocolo. O código fonte completo será copiado no diretório MpZBee no HD de seu computador.
5. Recorra ao arquivo README distribuído com o código fonte para a lista de aprimoramentos e limitações da versão instalada.

**ANEXO 17 - ARQUIVOS FONTE MICROCHIP STACK EM SUBDIRETÓRIOS
ZIGBEESTACK**

Nome do arquivo	Descrição
<i>SymbolTime.c, .h</i>	Executa contagem de tempo de funcionamento para o Microchip Stack para o protocolo ZigBee™
<i>zAPL.h</i>	Aplicação de arquivo de cabeçalho de interface nivelado para o Stack. Este é o único arquivo que o código de aplicação precisa ser incluído
<i>zAPS.c, .h</i>	Camada APS do protocolo Zigbee
<i>zHCLighting.h</i>	Informação de perfil de iluminação, controle residencial para protocolo Zigbee.
<i>zigbee.h</i>	Constantes genéricas do protocolo Zigbee
<i>ZigBeeTasks.c, .h</i>	Dirige fluxo de programa pelas camadas do Stack
<i>zMAC.h</i>	Arquivo genérico de cabeçalho da camada MAC IEEE 802.15.4
<i>zMAC_CC2420.c, .h</i>	IEEE802.15.4 camada MAC para o transmissor Chipcon CC2420
<i>zNVM.c, .h</i>	Executa e armazena funções em memória não-volátil
<i>zNWK.c, .h</i>	Protocolo Zigbee para a camada de rede (NWK)
<i>zPHY.h</i>	Genérico IEEE 802.15.4 para a camada física (PHY) de arquivo de cabeçalho
<i>zPHY_CC2420.c, .h</i>	IEEE 802.15.4 camada PHY para o transmissor Chipcon CC2420
<i>zZDO.c, .h</i>	Camadas do protocolo Zigbee ZDO (ZDP).

**ANEXO 18 - ARQUIVOS FONTES COMUNS MICROCHIP EM SUBDIRETÓRIOS
COMUNS**

Nome do arquivo	Descrição
<i>Compiler.h</i>	Compilador de definições específicas
<i>Console.c, .h</i>	Interface com o código USART (opcional)
<i>Generic.h</i>	Constantes genéricas e definições de tipo
<i>MSPI.c, .h</i>	Interface com o código SPI
<i>Sralloc.c, .h</i>	Código de distribuição de memória dinâmica

**ANEXO 19 - DEMONSTRAÇÃO DE COORDENADOR DE PROTOCOLO ZIGBEE
EM SUB-DIRETÓRIOS DEMOCOORDINATOR**

Nome do arquivo	Descrição
<i>Coordinator Template.c</i>	Modelo para criar aplicações de coordenador de protocolo Zigbee
<i>Coordinator.c</i>	Principal aplicação do arquivo fonte
<i>DemoCoordinator.mcp</i>	Arquivo de projeto
<i>myZigBee.c</i>	Gerado pelo software ZENA™. Contém informação de aplicação específica
<i>Zigbee.def</i>	Gerado pelo software ZENA. Contém informação de aplicação específica.
<i>zLink.lkr</i>	Gerado pelo software ZENA. Projeto linker script

**ANEXO 20 - DEMONSTRAÇÃO DE DISPOSITIVO FINALS DE PROTOCOLO
ZIGBEE EM SUBDIRETÓRIOS DEMORFD**

Nome do arquivo	Descrição
<i>DemoRFD.mcp</i>	Arquivo de projeto
<i>FFD End Device Template.c</i>	Modelo para criar FFD e aplicações de dispositivo final
<i>RFD template with ACKs.c</i>	Gerado pelo software ZENA™. Contém informação de aplicação específica
<i>RFD Template.c</i>	Modelo para criar aplicações de RFD quando o RFD pelo reconhecimento de nível APS
<i>RFD.c</i>	Principal aplicação do arquivo fonte
<i>Zigbee.def</i>	Gerado pelo software ZENA. Contém informação de aplicação específica.
<i>zLink.lkr</i>	Gerado pelo software ZENA. Projeto linker script

ANEXO 21 – CÓDIGO C PARA GERENCIAMENTO SPI

```
#include "MSPI.h"
#include "Zigbee.def"

void SPIPut(BYTE v)
{
    SSPIF = 0;
    do
    {
        WCOL = 0;
        SSPBUF = v;
    } while( WCOL );
    while( SSPIF == 0 );
    SSPIF = 0;
}
```

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](#)