



Universidade do Porto

Faculdade de Engenharia

FEUP

Módulos de Comunicação Wireless para Sensores

André Teixeira da Silva

andre.teixeira@fe.up.pt

O presente trabalho foi desenvolvido no âmbito da disciplina de Projecto, Seminário ou Trabalho Final de Curso da Licenciatura em Engenharia Electrotécnica e de Computadores, ramo TEC, efectuado sob orientação dos Professores Doutor Jorge Gomes Barbosa e Miguel Velhote Correia, Professores Auxiliares do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto.

Departamento de Engenharia Electrotécnica e de Computadores

Julho de 2007

Projecto desenvolvido através de parceria



RealSabor Indústria Alimentar

e



Faculdade de Engenharia da Universidade do Porto

André Teixeira da Silva

aluno 020503055 da Faculdade de Engenharia da Universidade do Porto

andre.teixeira@fe.up.pt

Página *web* e conteúdos *online* do projecto em: www.fe.up.pt/~ee02055

Agradecimentos

Gostaria de expressar o meu agradecimento a todas as pessoas que, de alguma forma, contribuíram para a realização deste Projecto de Final de Curso e me ajudaram a ultrapassar as dificuldades por mim sentidas, nomeadamente:

- aos meus orientadores, Prof. Jorge Gomes Barbosa e Prof. Miguel Velhote Correia, por toda a ajuda prestada ao longo deste trabalho;
- ao Eng.º Miguel Fernandes, em representação da empresa RealSabor Indústria Alimentar, pela cooperação;
- aos Professores José Carlos Alves e João Paulo Sousa, pela disponibilidade na cedência de instalações e equipamento.

Por fim, ainda um último agradecimento aos meus pais e aos meus amigos, pelo incondicional apoio em cada momento desta jornada.

André Teixeira Silva

Resumo

No decorrer dos últimos anos tem-se assistido a uma enorme expansão de dispositivos de comunicação via rádio, que surgem nas mais distintas áreas, penetrando cada vez mais este tipo de tecnologias no quotidiano de pessoas e indústria.

As redes dedicadas à comunicação industrial, com transporte de informação relativa a sensores (de peso, de temperatura, de humidade, etc.) possuem características bastante distintas das redes pessoais ou das redes empresariais para transmissão de voz e dados.

Foi no contexto da linha de produção de uma indústria alimentar que se desenvolveu o presente trabalho, procurando identificar-se quais as necessidades dessas mesmas redes industriais, quais as suas especificidades e, através da investigação e desenvolvimento, que tecnologia e hardware poderiam proporcionar uma solução adequada.

O protocolo ZigBee™, lançado em 27 de Junho de 2005, é um protocolo orientado para uma comunicação wireless sob o binómio baixo consumo / baixo débito, suportado sobre a normativa IEEE 802.15.4 e surgindo como responsabilidade da ZigBee Alliance. Esta aliança, ao agrupar distintas empresas da área tecnológica em questão, revela também um objectivo deste protocolo: a standardização, em detrimento de protocolos proprietários anteriormente existentes.

Após um detalhado estudo sobre este protocolo, onde se inclui também a comparação com outras tecnologias, seguiu-se uma análise dos dispositivos disponíveis no mercado para o desenvolvimento de uma solução de comunicação.

Baseando-se o desenvolvimento do trabalho sobre o kit PICDEM Z da Microchip, embora inicialmente tivesse sido seleccionada uma solução recorrendo ao transceiver Maxstream XBee, este relatório tem como objectivo apresentar todo o esforço aplicado na realização deste projecto, retratando também as dificuldades encontradas.

Dado terem existido problemas com a pilha protocolar disponibilizada pela Microchip, o que provocou instabilidade na fase de desenvolvimento do projecto, são ainda sugeridas algumas opções para evolução futura deste trabalho assim como, a título conclusivo, uma breve reflexão sobre o estado de maturação desta tecnologia.

Abstract

In the course of the latest years one has seen an enormous increase of devices of communication by radiofrequency, which appear in the most different areas and this kind of technology is entering more and more in the daily life of people and industry.

The networks devoted to industrial communication, with conveyance of information related to sensors (of weight, temperature, humidity, etc.) have features that are very different from personal or enterprise networks to the transmission of voice and data.

It was in a concept of a production line that this work had its development, trying to identify the needs of those same industrial networks, their special features and, through investigation and development, to know which technology and hardware would come to a suitable solution.

The protocol ZigBee™, launched in 27th June 2005, it's a protocol aimed at a wireless communication under the binomial: low cost / low data rate, supported by the standard IEEE 802.15.4 and appearing under the responsibility of ZigBeeAlliance. This alliance, on grouping different companies of this technological area, also shows an aim of this protocol: the standardization instead of owner protocols that existed before.

After a detailed search about this protocol, in which other different technologies area compared, it was made an analysis of the available devices for the development of a solution of communication.

The development of this project is based on a PICDEM Z™ of Microchip, although at first it was selected a solution based on Maxstream's XBee transceiver. This report aims at presenting the whole effort used during the making of this project, showing at the same time all the face difficulties.

As there were problems caused by Microchip ZigBee stack, which gave origin to some instability in the phasis of development of the project, some options are suggested for the evolution of this project, as well as, just to conclude, a short reflection about the maturing of this technology.

Índice

1	INTRODUÇÃO	10
1.1	Contexto e Motivação.....	10
1.2	Guia de Leitura	12
2	ESTUDO DA TECNOLOGIA.....	13
2.1	Identificação de Requisitos Iniciais.....	13
2.2	Estudo das Redes Wireless	14
2.3	Comparação de diferentes tecnologias sem fios.....	16
2.4	O protocolo ZigBee – um estudo detalhado	19
2.4.1	Pequena perspectiva histórica sobre o ZigBee	21
2.4.2	Detalhes técnicos	21
2.4.3	Topologia da rede.....	21
2.4.4	Modos de operação da rede.....	23
2.4.5	Características dos diferentes tipos de dispositivos lógicos.....	24
2.4.6	A arquitectura protocolar ZigBee / IEEE 802.15.4.....	25
2.4.7	Tipos definidos de tramas:.....	27
2.4.8	Terminologia ZigBee.....	27
3	SELECÇÃO DE HARDWARE.....	34
3.1	Helicomm EZ-Net DevKit 1000.....	34
3.2	Silicon Labs 2.4 Ghz ZigBee Development Kit.....	35
3.3	Data Hunter Tick RS232 Parasitically Powered Data Radio	36
3.4	Maxstream XBee ZigBee RF Modules	38
4	DESENVOLVIMENTO DO TRABALHO.....	41
4.1	Sistema PIC / XBee.....	41
4.1.1	O microcontrolador PIC	42
4.1.2	O conversor MAXIM MAX 3221	42
4.2	Microchip PICDEM™ Z	45
4.2.1	O microcontrolador PIC18LF4620	47
4.2.2	Interface Intraplaca e Pilha protocolar	49
4.2.3	Esquema Eléctrico da Placa-Mãe do kit PICDEM Z.....	51
4.2.4	Esquema Eléctrico da Placa-Filha do kit PICDEM Z.....	52
4.3	Ferramentas de desenvolvimento	53
4.4	Utilização de Programa para leitura da porta-série. Balança Mettler...	53

4.5	Interacção ligação série.....	54
4.5.1	Escrita via Porta-Série	54
4.5.2	Programa de Leitura via Porta-série.....	57
4.5.3	Programa de Leitura via Porta-Série com Interrupção	57
4.6	Interacção entre nós ZigBee	59
4.7	Calendário do trabalho realizado.....	64
5	CONCLUSÃO	65
6	DESENVOLVIMENTOS FUTUROS.....	69
7	BIBLIOGRAFIA.....	71
A.	ANEXOS	73
A.1	Balança Mettler-Toledo Spider 1	73
A.1.1	Configuração Comunicação-Série.....	73
A.1.2	Parametrização do Equipamento	74
A.1.3	Configuração do Equipamento – Interface / Comunicação	76
A.2	Código realizado	79
A.2.1	Escrita Porta-Série.....	79
A.2.2	Leitura Porta-Série.....	79
A.2.3	Leitura Porta-Série com interrupção.....	80
A.2.4	Comunicação entre nós ZigBee	83

Índice de Figuras

Fig. 2.1 Logótipos das diferentes tecnologias Wi-Fi, ZigBee™, Bluetooth	16
Fig. 2.2 Representação gráfica da Taxa de Transferência de Dados vs Alcance ...	17
Fig. 2.3 Topologia de rede: Estrela	22
Fig. 2.4 Topologia de rede: Malha	23
Fig. 2.5 Topologia de rede: Árvore.....	23
Fig. 2.6 Camadas da Arquitectura Protocolar ZigBee.....	25
Fig. 3.1 Helicomm EZ-Net DevKit 1000	34
Fig. 3.2 Silicon Labs 2.4 Ghz ZigBee Development Kit	35
Fig. 3.3 Data Hunter Tick RS232 Parasitically Powered Data Radio.....	36
Fig. 3.4 Software de configuração Data Hunter	37
Fig. 3.5 Maxstream XBee ZigBee Module	38
Fig. 3.6 Vista Frontal XBee / XBee Pro.....	39
Fig. 3.7 Vista Lateral XBee / XBee Pro	39
Fig. 4.1 Esquema de nó ZigBee constituído por PIC, Transceiver e Antena.....	41
Fig. 4.2 Diagrama de pinos do MAXIM MAX3221	42
Fig. 4.3 Nó Microchip PICDEM™ Z	45
Fig. 4.4 Placa-filha, contendo transceiver Chipcon e antena PCB em F invertido...	46
Fig. 4.5 Diagrama de pinos do PIC18LF4620.....	48
Fig. 4.6 Pilha protocolar Microchip: esquema de arquitectura e ficheiros.....	50
Fig. 4.7 Esquema Eléctrico da Placa-Mãe do kit PICDEM Z	51
Fig. 4.8 Esquema Eléctrico da Placa-Mãe do kit PICDEM Z	52
Fig. 4.9 Módulo MPLAB ICD2	53
Fig. 4.10 Peripheral Interrupt Enable Register (PIE1).....	57
Fig. 4.11 Reset Control Register (RCON).....	58
Fig. 4.12 Interrupt Control Register (INTCON).....	58
Fig. 4.13 Peripheral Interrupt Priority Register 1 (IPR1)	58
Fig. 4.14 Nó Coordenador: criação de rede ZigBee bem sucedida	61
Fig. 4.15 Nó Endpoint: anomalia na procura de rede ZigBee	61
Fig. 4.16 PICDEM Z: Processo de Envio de Dados entre dois nós	63
Fig. A.1 Balança Mettler-Toledo Spider 1	74
Fig. A.2 Configuração HyperTerminal ligação-série	74

Índice de Tabelas

Tabela 2.1 Características Wi-Fi, Bluetooth e ZigBee	16
Tabela 2.2 Redes ZigBee: dispositivos lógicos e suas funções	24
Tabela 2.3 Descritores ZigBee.....	29
Tabela 2.4 Protocolo ZigBee: Arquitectura dos <i>Profiles</i>	30
Tabela 4.1 PICDEM Z: Interface microcontrolador/transceiver.....	49

1 Introdução

1.1 Contexto e Motivação

No mundo actual, as comunicações assumem um papel cada vez mais relevante.

A tecnologia progride, as necessidades simultaneamente criam-se a partir da oferta e orientam também a pesquisa. A quase permanente contactibilidade para com qualquer pessoa, a possibilidade de previamente à chegada à casa se haver activado o aquecimento central ou uma qualquer fotografia atravessar os cinco continentes em apenas escassos segundos, que poderão outrora ter sido cenário de ficção científica, estão hoje num plano totalmente real.

No decorrer dos últimos anos tem-se assistido a uma enorme expansão de dispositivos de comunicação via rádio, que surgem nas mais diversas áreas, penetrando cada vez mais este tipo de tecnologias no quotidiano de pessoas e empresas. A ausência de uma ligação por cabo e uma consequente versatilidade em termos de instalação ou posição dos dispositivos intervenientes na rede é a vantagem mais evidente das redes que se apoiam sobre este tipo de comunicação, a somar à possibilidade dos próprios emissores ou receptores poderem estar em movimento durante a comunicação.

A RealSabor é uma empresa do sector alimentar, especializada na produção de charcutaria e salsicharia fina. Na sua linha de produção existem múltiplos parâmetros que vão sendo sucessivamente sensorizados, como forma de assegurar a adequada qualidade na confecção dos produtos.

Desta forma, revelar-se-ia de interesse a possibilidade de comunicação dos diversos dados para uma unidade central, tendo em vista o seu posterior armazenamento e tratamento. Para que o impacto de introdução de um novo sistema seja diminuto, nomeadamente em termos de instalação, e não sendo desprezável a questão da mobilidade, torna-se relevante que a comunicação seja efectuada preferencialmente via rádio.

É neste contexto que surge o presente projecto, tendo sido objectivo principal conseguir uma sólida abordagem ao problema, desde a identificação de características desejáveis, o levantamento da tecnologia e hardware disponíveis, na busca de soluções adequadas e implementáveis.

Procurou-se que este relatório registasse todos os resultados e dificuldades encontrados, não só como súmula do trabalho realizado, mas também para futuros projectos a desenvolver neste campo.

De facto, mais além de um primeiro contacto com um projecto de Engenharia, nas suas diferentes fases, foram várias as dificuldades a contornar e a busca de alternativas, assim como a aceitação de que “nem tudo corre bem”, tornou-se inevitável. Nomeadamente em termos da planificação inicialmente prevista, acabou por existir o prolongamento de algumas etapas em detrimento de outras, de forma a existir uma evolução coerente e regida por prioridades.

Constituiu-se assim um desafio, quer em termos pessoais quer em termos académicos, como estudante de Engenharia e futuro Engenheiro.

1.2 Guia de Leitura

Este relatório está estruturado em 6 capítulos.

No capítulo inicial, “Introdução”, pretende-se introduzir o leitor ao contexto deste projecto e aos motivos que o levam a ser realizado.

No capítulo seguinte, o capítulo 2, que se intitula “Estudo da Tecnologia”, é feita uma identificação dos requisitos iniciais, um estudo e comparação dos protocolos disponíveis em termos de redes wireless (e que mais se afiguravam como possíveis de serem aplicados) assim como um estudo aprofundado sobre o protocolo ZigBee™. Dentro desta última secção encontramos várias subsecções, que procuram esclarecer o leitor sobre inúmeros aspectos deste protocolo (tido como o mais adequado para este projecto) quer em termos de uma breve introdução histórica quer em termos de detalhes técnicos de arquitectura e operação.

características análise da tecnologia de redes disponível.

No capítulo 3, “Seleção de Hardware”, são apresentados os resultados da pesquisa efectuada em termos de hardware disponível para desenvolvimento de uma solução adequada.

No capítulo 4, “Desenvolvimento do Trabalho”, apresenta-se a solução inicialmente projectada, o porquê da sua não-utilização e expõe-se a alternativa utilizada, o kit PICDEM-Z da Microchip. Em seguida, efectua-se uma descrição de toda esta fase que constitui o projecto, em termos de testes, desenvolvimento e dificuldades ocorridas. Apresenta-se ainda o calendário das tarefas realizadas.

A “Conclusão” constitui o capítulo 5, onde se faz um comentário global ao projecto e uma breve reflexão sobre a maturidade da tecnologia ZigBee.

Finalmente, o capítulo 6, “Desenvolvimentos Futuros” apresentam-se sugestões para evoluções futuras deste mesmo projecto, utilizando a experiência adquirida até à data.

Existem ainda informações adicionais, relacionadas com o trabalho realizado, na secção Anexos.

2 Estudo da Tecnologia

2.1 Identificação de Requisitos Iniciais

Numa linha de produção de géneros alimentícios, para um correcto controlo e adequação das características do produto final, existem diversos sensores – nomeadamente de peso, temperatura e humidade.

Pretende-se desenvolver um módulo de comunicação wireless (emissor / receptor) tendo em vista o acoplamento do mesmo aos vários tipos de sensores. Desta forma, possibilitar-se-á a comunicação, com ausência de fios, dos dados relativos aos sensores para uma unidade central (um computador ao qual estará acoplado também um destes módulos).

Embora alguns dos sensores permaneçam fixos ao longo de todo o processo, outros são sujeitos a deslocações devido a características do próprio método produtivo – o que faz com que não sejam necessariamente alimentados pela rede eléctrica, se não que por bateria própria.

É também relevante que as características físicas deste módulo permitam a sua integração num ambiente de linha de produção industrial.

Assim, para que se adequem ao fim a que se destinam, os módulos de comunicação deverão caracterizar-se por:

- um reduzido consumo de energia (ao qual estará também associada uma baixa complexidade), especialmente tendo em vista uma elevada autonomia nos casos de em que a alimentação se efectue por bateria;
- pequenas dimensões, para permitir um flexível acoplamento aos distintos sensores e elementos de medição, assim como o seu transporte;
- baixa dependência das condições exteriores e possibilidade de serem integrados em invólucro estanque, não criando assim restrições à sua utilização;
- adequação de interface, no sentido de que é desejável que seja ampla a capacidade de interligação a diferentes dispositivos, isto é que o interface disponibilizado seja adequado à sua área de aplicação (neste caso, o suporte a comunicação RS-232 será o mais versátil dado permitir acoplamento a balanças, sensores de temperatura ou humidade, ...);
- preço comportável, em conformidade com as funções a executar.

No caso do “módulo central” de recepção da informação, poderá admitir-se uma maior complexidade, estando sujeito a menores restrições de consumo e

dimensões, uma vez que previsivelmente estará sujeito a uma instalação fixa, ligado a um PC, e alimentado à rede.

Há ainda a destacar que se deverá efectuar uma previsão das possibilidades de evolução da solução adoptada. Quer isto dizer que será preferível optar por um sistema que se apoie num protocolo estandardizado, em detrimento de protocolo de comunicações próprio do fabricante, para que, no futuro, seja admissível a utilização de outro tipo de equipamentos na mesma rede (eventualmente sensores já com tecnologia de transmissão via-rádio incorporada, dispositivos de controlo, entre outros).

2.2 Estudo das Redes Wireless

O desenvolvimento da tecnologia permitiu o aparecimento de várias alternativas e protocolos, embora inicialmente tenha sido colocada ênfase principalmente na transmissão de dados e voz, com elevadas taxas de transferência. Esta perspectiva tornava os equipamentos com preços de aquisição elevados, resultando assim desinteressantes para aplicações mais simples.

De facto, só após o desenvolvimento dos vários protocolos para suporte de comunicações sem fios, de médio ou alto débito, como o Bluetooth ou o Wi-Fi, se começou a pensar num protocolo que respondesse às necessidades específicas de sensores e dispositivos de controlo. Até então os diversos fabricantes deste tipo de equipamentos adoptavam soluções próprias criando sérios problemas de interoperabilidade entre sistemas.

As redes dedicadas a comunicação industrial, com informação circulante relativa a sensores (de temperatura, humidade, peso, etc.) e dispositivos de controlo (botoneiras, relés, entre outros), possuem, na sua generalidade, características bastante distintas das redes pessoais ou empresariais para transmissão de voz e dados. Numa rede de âmbito industrial não se privilegiará tanto uma elevada taxa de transferência de dados mas sim uma latência baixa e um consumo de energia igualmente baixo, para preservar a vida útil das baterias.

Tal como foi identificado no nosso caso de estudo, é extremamente importante um reduzido consumo (de modo a que a facilidade de alimentação e autonomia, em caso de alimentação por bateria, sejam maximizadas), assim como se torna desejável um tamanho compacto dos módulos de comunicação, uma vez que se pretende que sejam elementos “acessórios” ao processo produtivo (e como tal, reduzida ao máximo a sua “presença”).

O grupo de trabalho 11 do IEEE (*Institute of Electrical and Electronics Engineers*), responsável pelos standards IEEE 802.11, considera a existência de quatro grandes grupos, em termos de redes sem-fios:

WPAN – *Wireless Personal Area Network*: onde se enquadram tecnologias wireless de pequeno alcance, entre 10 e 100 metros. Esta é a área de estudo e desenvolvimento normativo do grupo de trabalho 15 do IEEE, especializado nos standards WPAN. Esta área compreende as redes sem fios que utilizam dispositivos como os PDA's, PC's ou periféricos, sendo o campo de tecnologias como o Bluetooth (responsabilidade do subgrupo IEEE 802.15.1) ou IrDA. O Low Rate WPAN, que tenta reunir uma baixa *data rate*, baixa complexidade e oferecer um reduzido consumo energético, faz parte integrante deste campo e é sobre ele que foi desenvolvido o ZigBee, conjunto de especificações da responsabilidade do subgrupo de trabalho IEEE 802.15.4.

WLAN – *Wireless Local Area Network*: tecnologias que se destinam a redes sem fio com alcance situado entre os 100 e 300 metros, frequentemente como extensão ou alternativa a redes de cabeamento convencional (par de cobre, cabo coaxial ou fibra óptica).

WMAN – *Wireless Metropolitan Area Network*: neste campo o objectivo é relativo a acessos de banda larga em redes de âmbito metropolitano, cujo alcance será em torno de 6 km.

WWAN – *Wireless Wide Area Network*: o grupo mais amplo em termos de alcance, é orientado para serviços de telecomunicações (voz e dados), em longa distância de transmissão.

2.3 Comparação de diferentes tecnologias sem fios

Wi-Fi (IEEE 802.11b/g), Bluetooth (IEEE 802.15.1), ZigBee (IEEE 802.15.4)



Fig. 2.1 Logótipos das diferentes tecnologias Wi-Fi, ZigBee™, Bluetooth

ESPECIFICAÇÃO	DÉBITO	CONSUMO	PILHA PROTOCOLAR	VANTAGENS	PRINCIPAIS APLICAÇÕES
Wi-Fi (IEEE 802.11b/g)	54Mbps	>400mA TX, <i>standby</i> 20mA	1MB +	Elevada Taxa de Transferência	Internet, Transferências de Ficheiros, Vídeo/Aúdio
Bluetooth (IEEE 802.15.1)	1Mbps	>400mA TX, <i>standby</i> 0.20mA	≈ 250KB	Interoperabilidade, substituição de cabos	Periféricos de PC e Telemóveis, PDA's
ZigBee (IEEE 802.15.4)	100kbps	30mA TX, <i>standby</i> 0.20μA	≈ 32KB	Consumo, Latência, N.º de Nós, Fiabilidade, €	Controlos Remoto, Sensores, Dispositivos alimentados por bateria

Tabela 2.1 Características Wi-Fi, Bluetooth e ZigBee

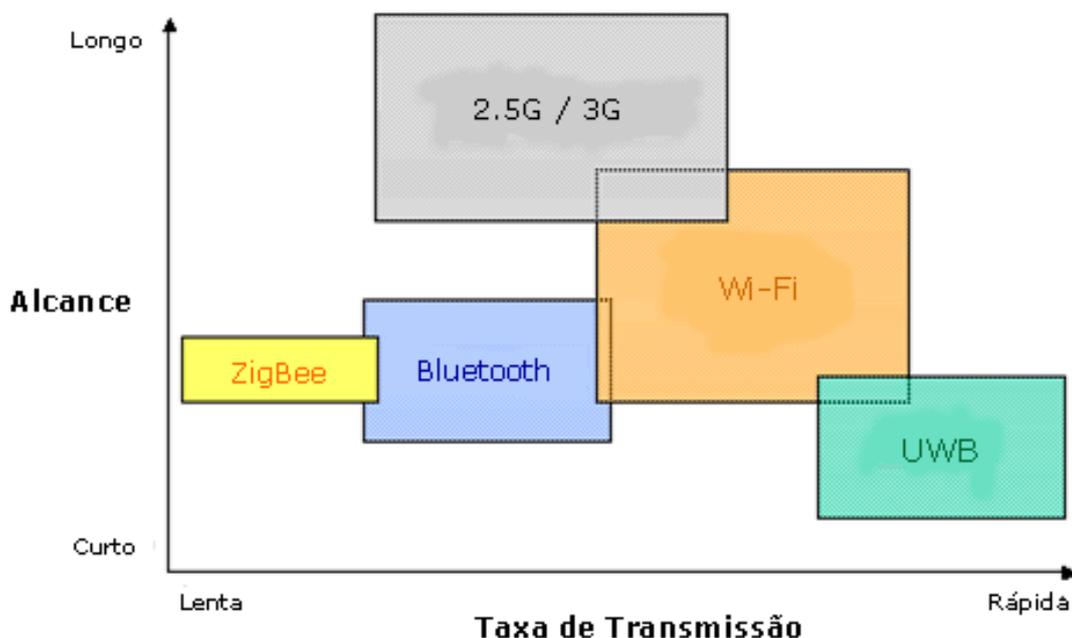


Fig. 2.2 Representação gráfica da Taxa de Transferência de Dados vs Alcance¹

Na Tabela 2.1 *Características Wi-Fi, Bluetooth e ZigBee* podemos verificar, de uma forma sumária, algumas das principais características dos standards wireless Wi-Fi, Bluetooth e ZigBee.

O Wi-Fi, baseado sobre as normas da família IEEE 802.11, é um protocolo orientado para aplicações que requeiram um elevado débito, tais como a transferência de ficheiros ou o acesso à Internet e a conteúdos multimédia. Por outro lado, dado os objectivos deste protocolo, o suporte à sua pilha protocolar, em termos de hardware, é bastante mais exigente (podendo assumir a mesma um tamanho consideravelmente elevado). O seu consumo energético, especialmente quando em standby, é também claramente superior ao das outras tecnologias que aqui se comparam.

Por ambos apresentarem débitos mais reduzidos em relação ao Wi-Fi, uma dúvida frequentemente surgida é se o Bluetooth e ZigBee são efectivamente protocolos “concorrentes”.

Segundo a própria ZigBee Alliance, o ZigBee foi desenvolvido para servir aplicações bastante distintas das do Bluetooth e, em especial, atentando a uma optimização máxima do consumo energético. Conforme se constata, efectivamente o protocolo ZigBee apresenta um consumo muito reduzido quando comparado com

¹ As tecnologias Ultra Wide Band (UWB) e GSM / UMTS (2.5G / 3G), embora representadas, pelas suas características não foram objecto de estudo.

o Wi-Fi ou Bluetooth, especialmente nos períodos de inactividade – o seu consumo será, nesta fase, cerca de 100 vezes inferior ao do Bluetooth!

Assim, e ainda que o ZigBee seja posteriormente objecto de análise mais detalhada, importa destacar os pontos de clara diferenciação entre este protocolo e a especificação Bluetooth:

ZigBee:

- duty cycle muito baixo e suporte para dispositivos de funções reduzidas (minimizando o consumo), possibilitando elevada autonomia quando alimentação por baterias;
- suporte a topologias de rede estáticas e dinâmicas, quer em estrela quer em malha;
- capacidade para permanecer longos períodos sem comunicação;
- permite a utilização de redes com mais de 65.000 nós, procurando garantir sempre baixa latência;
- uso de Direct Sequence Spread Spectrum permite que os dispositivos permaneçam em “sleep-mode” sem necessidade exigente de sincronização;

Bluetooth:

- duty cycle moderado (não tão elevado como no Wi-Fi), mas com consumo idêntico para qualquer tipo de dispositivo interveniente na rede;
- elevado QoS;
- redes quase estáticas de topologia em estrela e com um máximo previsto de 7 clientes;
- o recurso ao Frequency Hopping Spread Spectrum implica que seja extremamente difícil a criação de redes extensas sem custos elevados em termos de sincronização;

Conforme se pode constatar pelos dados anteriores, o ZigBee possibilita então um menor consumo de energia associado a débito e alcance aceitáveis, o que o torna bastante atractivo para dispositivos remotos que sejam alimentados por baterias, como o caso dos módulos para sensores no presente estudo. Por outro lado, a pilha protocolar é também menor em termos de complexidade, o que exige menores recursos nos dispositivos que a utilizem.

2.4 O protocolo ZigBee – um estudo detalhado

No seguimento do levantamento de tecnologias disponíveis efectuado anteriormente, concluiu-se ser o protocolo ZigBee aquele que mais se adequava aos objectivos deste projecto.

No sentido de satisfazer as necessidades crescente da “sociedade de informação”, o mercado das comunicações tem sofrido nos últimos anos uma evolução muito elevada. De facto, no contexto das redes sem fios (wireless) existem múltiplas alternativas, cada uma dela orientada a aplicações distintas.

O Zigbee, sendo uma tecnologia recente, teve a sua primeira versão apresentada ao público em 27 de Julho de 2005 (embora tenha efectivamente surgido em Dezembro de 2004) e a entidade que se apresenta responsável por este protocolo intitula-se ZigBee™ Alliance. A ZigBee™ Alliance é uma aliança constituída por mais de 200 empresas, oriundas de mais de 20 países distintos, na qual se integram também especialistas da área de telecomunicações e semicondutores, incluindo membros do IEEE.

Até à data não se encontrava no mercado nenhuma norma de redes sem fios globalmente aceite no âmbito de sensores e dispositivos de controlo. Contraindo-se aos elevados débitos oferecidos por outras tecnologias (como Bluetooth ou o WiFi), o ZigBee pretende associar a transmissão de dados sem fios a um reduzido consumo energético e com elevada fiabilidade. Assim, um objectivo desta norma é também uma tentativa de uniformizar o desenvolvimento de aplicações e dispositivos nesta área, abandonando-se sistemas proprietários distintos para dar lugar a um *standard*, que, ao ser compatível, proporciona evidentes vantagens de interoperabilidade.

Com uma vasta área de aplicação, desde o controlo industrial à automação de residências (*domótica*), o protocolo ZigBee possui então determinadas características que o tornam absolutamente distinto dos restantes, sustentando os motivos que levaram à sua criação:

- reduzido consumo de energia – *low power*;
- pilha protocolar de implementação simplificada (código de tamanho menor de um quarto da do Bluetooth, por exemplo), conduzindo a interfaces de baixo custo – *low cost*;

- possibilidade de suportar uma elevada densidade de nós por rede (num máximo de 65535 dispositivos por cada ZigBee Coordinator, valor manifestamente superior aos 8 do Bluetooth ou 30 do Wi-Fi);
- admite diferentes topologias da rede: estrela (*star*), malha (*mesh*) ou árvore (*cluster tree*), permitindo o estabelecimento de redes de nós “*ad-hoc*”.
- possuindo um tempo de ligação à rede menor que os outros protocolos e apresenta maior rapidez na passagem do modo standby a activo, o ZigBee apresenta também uma latência baixa – *low latency*;
- dois estados de operação: *active*, aquando do envio ou recepção de dados, e *sleep* (contrastando, por exemplo, com o Bluetooth, que apresenta inúmeros modos e estados de latência e potência) – a aplicação não terá de se preocupar em seleccionar o modo mais adequado;
- dois modos de operação da rede: *beaconing* e *non-beaconing*.
- elevada segurança, com recurso a *128-bit encryption*;
- elevada fiabilidade;
- suporte para duas classes de dispositivos físicos (definidos na norma IEEE 802.15.4), podendo ambos coexistir numa mesma rede:
 - * *Full Function Device* (FFD) – pode funcionar em qualquer que seja a topologia da rede, desempenhando a função de coordenador da rede e conseqüentemente ter acesso a todos os outros dispositivos. Assim, tratam-se de dispositivos de construção mais complexa;
 - * *Reduced Function Device* (RFD) – é limitado a uma configuração com topologia em estrela, não podendo actuar como coordenador da rede. Pode apenas comunicar-se com um coordenador de rede. São dispositivos de construção mais simples;
- às duas classes anteriores de dispositivos físicos correspondem três tipos de dispositivos lógicos: *Coordinator*, *Router* e *Endpoint*.

“*To enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard*” é a missão da ZigBee Alliance, segundo a própria.

2.4.1 Pequena perspectiva histórica sobre o ZigBee

Anteriormente à adopção da nomenclatura ZigBee e da constituição da ZigBee™ Alliance, a responsável por esta tecnologia era a empresa Philips, sendo nessa época designado por protocolo Home RF Lite.

Apesar de actualmente admitir distintas topologias, foi através da topologia em malha que se lhe associou o nome. Uma malha ZigBee apresenta múltiplos percursos possíveis entre cada dispositivo, o que permite eliminar um possível ponto de falha, através do “zig” e “zag” da informação pela rede.

As abelhas de mel, para além de viverem “em colmeia”, voam em ziguezague e dessa forma lhes é possível informar os restantes elementos da sua comunidade da distância, direcção e localização dos alimentos descobertos. Assim, o nome deste protocolo surge da analogia da estrutura e modo de funcionamento da rede de comunicações para com modo de vida das abelhas.

2.4.2 Detalhes técnicos

O ZigBee opera em três bandas de rádio conhecidas como ISM (Industrial, Scientific and Medical), as quais estão isentas de licenciamento. Globalmente, corresponde à banda de 2.4Ghz e ainda às bandas de 915Mhz (Estados Unidos) e 868 Mhz (Europa). Consoante a banda, varia a taxa de transmissão possível: em 2.4Ghz podem ser obtidas taxas de transmissão de 250Kbps, com 16 canais disponíveis; a 915Mhz, está disponível uma taxa de transmissão de 40Kbps e 10 canais de comunicação; no caso de 868Mhz, possibilita 1 canal e uma taxa de transmissão de 20Kbps.

Em termos de modulação, é utilizado O-QPSK (*Offset quadrature phase-shift keying*) para a banda dos 2.4Ghz e BPSK (*binary phase shift keying*) para os 915 ou 868Mhz.

2.4.3 Topologia da rede

Embora já anteriormente referidas de forma breve, importa detalhar cada uma das topologias de rede admitidas pelo ZigBee.

Uma vez que este standard perspectiva a rede de uma forma *ad hoc*, não existe uma topologia predeterminada nem um controlo obrigatoriamente “centralizado”. Note-se no entanto, e apesar desta característica dinâmica de configuração da rede, que em qualquer uma das topologias o Zigbee Coordinator é o dispositivo responsável por iniciar a rede.

Em seguida apresenta-se uma representação gráfica e sucinta abordagem do conceito inerente a cada topologia, podendo cada uma delas coexistir dentro de uma mesma rede:

ESTRELA (*Star*)

É ao ZigBee Coordinator que cabe todo o controlo da rede, assumindo este um papel central e de comunicação directa com todos os dispositivos Endpoint – é portanto o Coordinator que inicia e mantém os dispositivos na rede. Toda a informação em circulação na rede passa pelo nó Coordinator.

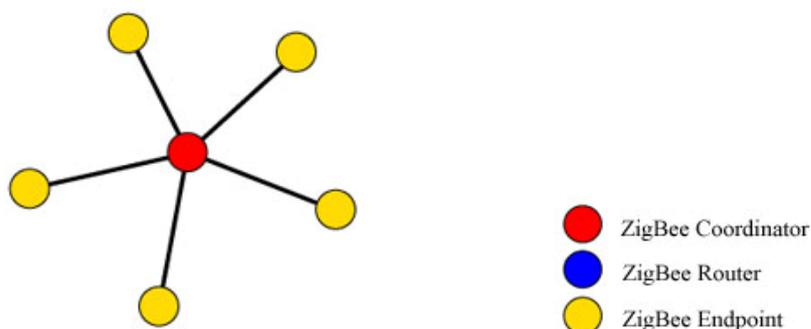


Fig. 2.3 Topologia de rede: Estrela

MALHA (*Mesh*)

Numa topologia em malha os dispositivos do tipo FFD (Coordinator/Routers) são livres de comunicar com outro dispositivo FFD. Isto permite, quando necessária, a expansão física da rede (maior alcance). O Coordinator regista toda a entrada e saída de dispositivos, mas não assume um papel tão preponderante em termos de fluxo de informação como na configuração anterior.

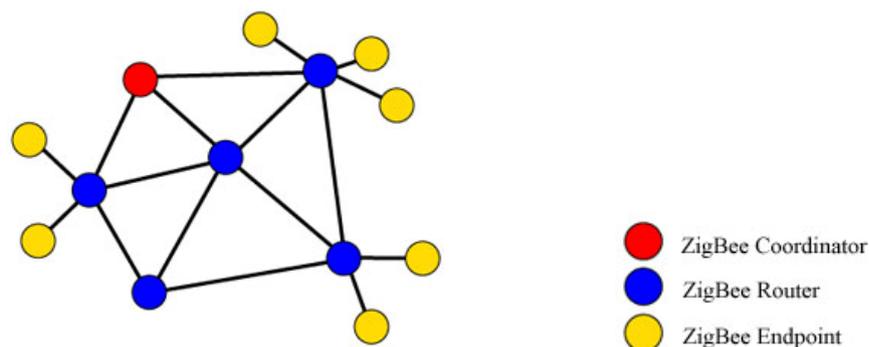


Fig. 2.4 Topologia de rede: Malha

ÁRVORE (Cluster Tree)

Apresentando semelhanças à rede em malha, também são usados dispositivos Router. No entanto, nesta topologia efectua-se a distribuição de dados e mensagens de controlo numa estrutura hierárquica, onde o Coordinator assume o papel de nó “nuclear” da rede.

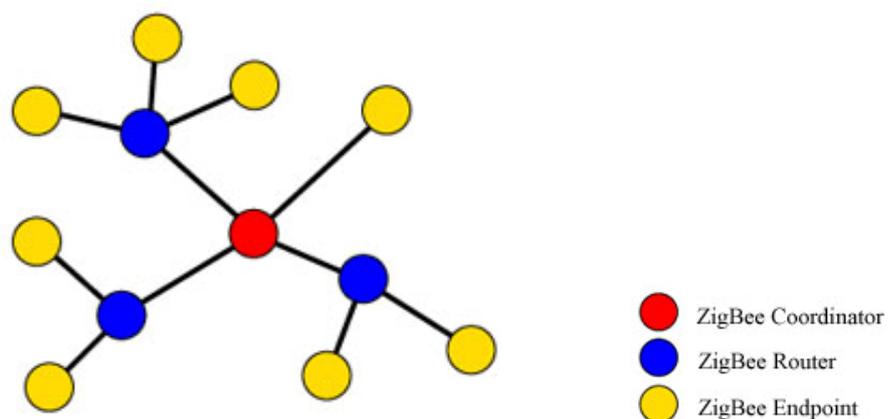


Fig. 2.5 Topologia de rede: Árvore

2.4.4 Modos de operação da rede

No modo *beaconing*, os nós ZigBee Routers transmitem periodicamente sinalização (*beacons*) a confirmar a sua presença aos outros nós da mesma rede, sendo que os restantes nós só necessitam de estar activos no momento da sinalização. Tal permite mantê-los no modo *sleep* entre sinalizações, com evidente

vantagem em termos de consumo energético (diminuem o seu duty cycle e, conseqüentemente, prolongam a autonomia da bateria a que possam estar ligados). O intervalo entre *beacons* pode variar entre os 15.36 ms e os 251.65s, para uma taxa de transmissão de 250kbit/s. No entanto, há que ter em conta que a operação com duty cycle reduzido (associada a intervalos prolongados entre *beacons*) requer uma temporização de elevada precisão, o que pode colidir com o interesse em produzir um dispositivo de baixo custo.

No modo *non-beaconing* sucede que a maioria dos dispositivos mantém os seus receptores permanentemente activos, sendo o consumo energético mais significativo (podendo tornar necessárias fontes de alimentação mais robustas).

2.4.5 Características dos diferentes tipos de dispositivos lógicos

Em termos de dispositivos físicos, os quais estão associados à norma IEEE 802.15.4, encontramos os FFD e os RFD conforme já referido anteriormente nesta mesma secção.

Sobre estes, o protocolo ZigBee efectua ainda a distinção entre três tipos de dispositivos lógicos que concretizam a rede de comunicação:

Dispositivo	Tipo de dispositivo físico associado (IEEE 802.15.4)	Função
Coordinator	FFD	Forma a rede, atribui endereços, suporta binding table. Existe apenas um por rede.
Router	FFD	Permite que mais nós se juntem à rede, ao aumentar o seu alcance físico. Pode também efectuar funções de controlo ou monitorização. A sua existência é opcional.
Endpoint	RFD ou FFD	Efectua acção de controlo ou monitorização através de dispositivo que lhe esteja associado (sensor, controlador, actuador...).

Tabela 2.2 Redes ZigBee: dispositivos lógicos e suas funções

2.4.6 A arquitectura protocolar ZigBee / IEEE 802.15.4

A arquitectura do protocolo ZigBee é composta por camadas, sendo que a camada executa serviços específicos ao dispor da camada superior: a entidade de dados fornece dados para o serviço de transmissão e a entidade de gestão fornece informação para todos os outros serviços. Cada entidade de serviço expõe uma interface para a camada superior através do ponto de acesso ao serviço (SAP) e cada SAP suporta um número de primitivas de serviço para activar a funcionalidade que se pretende solicitar.

Embora se baseie no modelo OSI (Open Systems Interconnection) de sete camadas, a arquitectura protocolar ZigBee apenas define, no entanto, as camadas de interesse para atingir as funcionalidades desejadas.

De uma forma simplificada, as diferentes camadas podem ser esquematizadas conforme se visualiza na Fig. 2.6 **Camadas da Arquitectura Protocolar ZigBee**

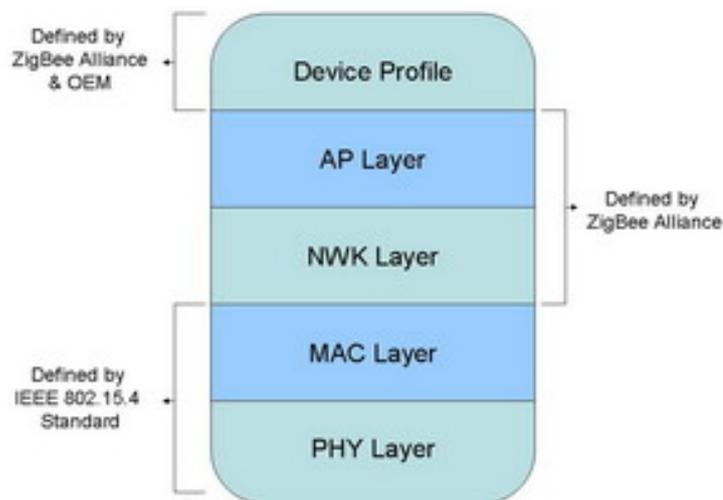


Fig. 2.6 Camadas da Arquitectura Protocolar ZigBee

As duas camadas mais baixas, a camada física (PHY) e camada de controlo de acesso ao meio (MAC), estão definidas pela norma IEEE 802.15.4. Construído sobre esta, é no protocolo ZigBee que se definem as camadas de rede imediatamente superiores. Ou seja, as camadas de rede (NWK) e o framework para a camada de aplicação (APL). Aqui se incluem a subcamada de suporte

aplicacional (APS), o Objecto de Dispositivo ZigBee (ZigBee Device Object – ZDO) e os Objectos de Aplicação (Application Objects).

A camada PHY é a responsável por permitir a transmissão e recepção de mensagens através de um canal físico RF (com as características já anteriormente explicitadas). Das suas funções fazem parte a activação e desactivação do transceiver, detecção da energia (ED – Receiver Energy Detection), indicação da qualidade da ligação (LQI – Link Quality Indication), selecção do canal e transmissão / recepção de pacotes através do meio físico.

À camada MAC cabe fundamentalmente o papel de controlar o acesso aos canais RF, utilizando mecanismos de prevenção de colisão CSMA-CA (Carrier Sense Multiple Access – Collision Avoidance), para o qual efectua comunicações com a camada inferior – camada PHY. Além disso, especifica o tipo de dispositivos permitidos na rede, a estrutura de tramas admissível e também está a seu cargo a sincronização e transmissão de tramas beacon, permitindo a fiabilidade de operação.

A camada NWK é, hierarquicamente, a primeira camada que é definida pela norma ZigBee, e apresentam-se da sua responsabilidade o início ou fim da ligação de um dispositivo à rede, a descoberta de novos dispositivos na vizinhança (e o armazenamento de informação relativa aos mesmos), a atribuição de endereços (apenas em dispositivos Coordinator). É ainda nesta camada que estão presentes os mecanismos de descoberta de rotas e encaminhamento de informação assim como de configuração de novos dispositivos.

Quanto à camada de aplicação (APL), contém a sub-camada Application Support Sublayer (APS), o ZigBee Device Object (ZDO) e a Application Framework (AF). Esta camada pretende assegurar uma correcta gestão e suporte para as diversas aplicações.

Para uma análise mais profunda e em detalhe, observe-se a seguinte representação gráfica da Fig. 2.7:

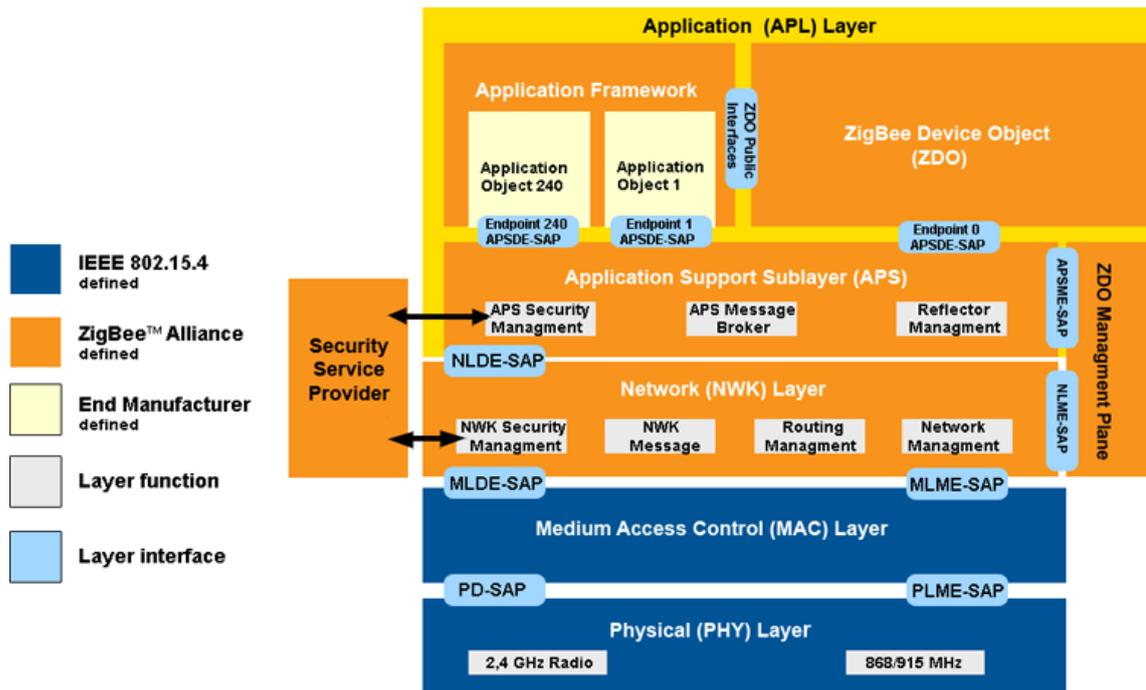


Fig. 2.7 Arquitectura Protocolar ZigBee

2.4.7 Tipos definidos de tramas

Tramas de comando MAC (MAC command) – para controlo dos nós clientes.

Tramas de dados – são usadas para todo o tipo de transferência de dados, Suportam 104 bytes e estão numeradas. A existência de uma sequência de frame-check permite assegurar uma transmissão fiável e sem erros.

Tramas de acknowledgment (ACK) – utilizadas para confirmar a recepção bem sucedida de um pacote. Os dispositivos utilizam o quiet time entre tramas para efectuar o acknowledgment.

Tramas de beacon – utilizadas pelos dispositivos Coordinator e Router para efectuar a transmissão de beacons.

2.4.8 Terminologia ZigBee [11] e [12]

Profile

É um acordo de mensagens, formatos de mensagens e procedimentos descrevendo a função dos dispositivos e suas interfaces de maneira a criar uma

aplicação distribuída. Por exemplo, existe o *profile* de Controlo da Iluminação Doméstica que prevê seis tipos diferentes de dispositivos (atribuindo-lhes identificadores e versões) a trocar mensagens KVP pré-definidas de modo a controlar o ligar e o desligar de lâmpadas, medir a luminosidade ou usar sensores de movimento. Cada *profile* terá de ter um identificador de *profile* único sendo necessário pedi-lo à ZigBee caso se pretenda criar um *profile*.

Atributo

Cada bloco de dados passado entre dispositivos através de comandos. O formato de cada atributo (tipo de dados) é definido no *profile*. Cada atributo terá de ter um identificador próprio definido no *profile*.

Cluster

Possui um grupo de atributos e pode ser Mandatory (tendo que estar presente em todos os dispositivos) ou Opcional conforme o estabelecido no *profile*. As interfaces são definidas entre *clusters*. Cada *cluster* terá de ter um identificador próprio definido no *profile*. Cada *cluster* só poderá ser de entrada ou saída, definindo-se isto no descritor simples. Pode haver até 256 *clusters* cada um com 65.536 Atributos.

Endpoint

Bloco funcional de código que suporta um ou mais *clusters*. A cada *endpoint* é atribuído um endereço em adição aos endereços únicos do dispositivo IEEE de 64 bits e de Rede (NWK) de 16 bits.

Descrição do dispositivo

Descreve um dispositivo dentro de um domínio e/ou segmento de aplicação. Um conjunto destes descritores forma um *profile*. Possui um identificador único (trocado no processo de descoberta).

Nó

Uma colecção de Descrições de Dispositivos e Aplicações independentes residentes numa única unidade rádio.

Descritores

Estruturas de dados usados pelos dispositivos para se descreverem, constituídas por 5 descritores individuais como mostra a Tabela 2.3 **Descritores ZigBee**. Alguns têm que existir sempre obrigatoriamente (*mandatory*) e outros são

opcionais. No momento da sua transmissão, seguem a ordem apresentada na tabela.

Nome do Descritor	Estatuto	Descrição
Node	Mandatory	Tipos e capacidades do nó
Node power	Mandatory	Características da alimentação
Simple	Mandatory	Descrições do dispositivo contidas no nó
Complex	Optional	Informação extra
User	Optional	Descritores definidos pelo utilizador

Tabela 2.3 Descritores ZigBee

Descritor de Nó

Nos seus campos indica o tipo de dispositivo (coordenador, *router* ou end device), as capacidades da APS, a frequência usada, as características da MAC (IEEE 802.15.4), o código do fabricante, o tamanho máximo do *buffer* e o tamanho máximo de transferência.

Descritor de potência do Nó

Especifica o modo actual de poupança de energia, as fontes de alimentação disponíveis, a fonte actual e o nível de carga para o caso de a fonte ser uma bateria.

Descritor Simples

Especifica informação para cada *endpoint* do nó em questão. Para cada *endpoint* é definido o seu número, o identificador de *profile*, especificada a descrição de dispositivo suportada neste *endpoint* em questão, a versão da descrição de dispositivo, é indicado se são usados o descritor Complexo (Complex) ou de Utilizador (User), o número de *clusters* de entrada, a lista dos *clusters* de entrada suportados por este *endpoint*, o número de *clusters* de saída e lista dos *clusters* de saída.

Descritor Complexo

É de uso opcional e possui os campos de língua e código de caracteres usados por ele próprio, o nome do fabricante, o nome do modelo do dispositivo, o número de série, o URL para mais informações, o ícone do dispositivo e o URL do ícone.

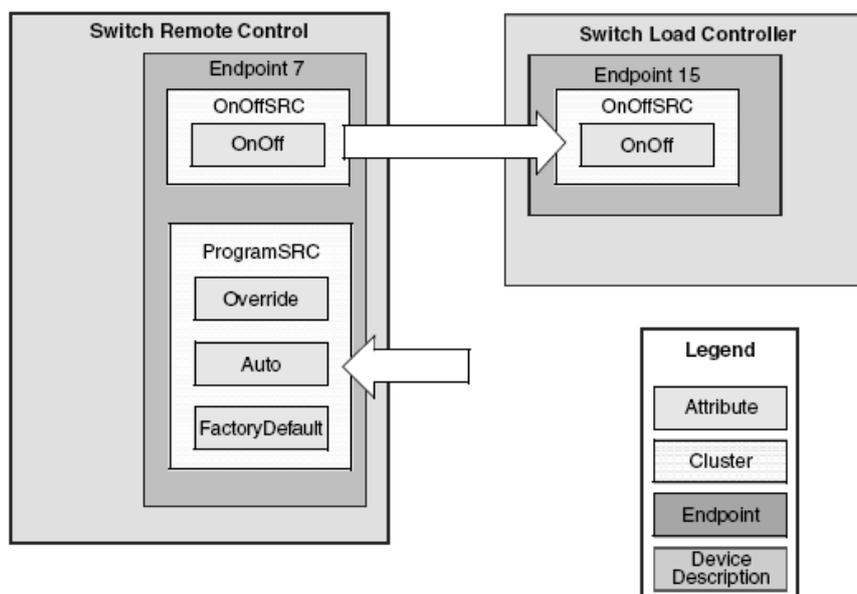


Tabela 2.4 Protocolo ZigBee: Arquitectura dos Profiles

Descritor de Utilizador

Contém apenas um campo (16 octetos) que permite ao utilizador final identificar o dispositivo dando-lhe o nome que pretender.

Localização de Dispositivo

Processo segundo o qual um dispositivo ZigBee pode descobrir outros dispositivos ZigBee enviando mensagens de pedido de identificação por *broadcast* ou *unicast*. Pode haver pedidos do endereço IEEE (já conhecido o NWK e por *unicast*) ou NWK (por *broadcast* e transporta o endereço IEEE). No caso do dispositivo ser um coordenador ou *router*, para além do seu endereço, enviará também os endereços dos dispositivos a ele associados.

Service Discovery

É um processo onde os serviços disponíveis nos *endpoints* de um dispositivo são descobertos por dispositivos externos. São rastreados todos os *endpoints* (por *unicast*) ou é usado o serviço de *match* (é enviada uma mensagem de difusão e só respondem os dispositivos em que os critérios coincidam), sendo usados todos os descritores do *endpoint*. No caso de haver dispositivos em “sleep-mode”, o seu coordenador ou *router* ZigBee poderá responder por eles. Existem os seguintes tipos de pedidos:

- **Endpoint Activo**

Este comando permite determinar os endpoints activos (a aplicação suporta um único *profile*).

- **Match Simple Descriptor**

Pede a identificação de *profile* e, opcionalmente, a de *clusters* e a do *endpoint* do dispositivo com os critérios fornecidos coincidentes.

- **Detector Simples/ Nó/ Potência/ Complexo/ Utilizador**

Estes comandos pedem os descritores respectivos do *endpoint*/dispositivo específico. Os 2 primeiros podem ser *broadcast* ou *unicast* e os outros somente *unicast*.

Tabela dos Vizinhos

Usada para manter o contacto com os outros dispositivos do POS (*Personal Operating Space*).

Binding

Criação de uma ligação lógica unidireccional entre um *endpoint/cluster*-fonte e um *endpoint* de destino. É executado após ser estabelecido uma ligação de comunicação. Estas informações de ligação de nós é armazenada na *Binding Table* existente no coordenador.

O Gestor de Binding (*Binding manager*) estabelece o tamanho inicial da tabela de *binding* e processa os pedidos de adição e remoção de entradas na tabela. Um *binding* pode ser pedido por *end devices* ou criado pelo coordenador.

Tabela de Binding

Cada entrada desta tabela é constituída pelo endereço, identificador do *endpoint* e identificador do *cluster* do dispositivo fonte e pelos endereço e *endpoint* do(s) dispositivo(s) de destino desta ligação de binding. Tem tamanho máximo pré estabelecido.

Serviços de Dados

A *application framework* fornece aos seus objectos de aplicação dois serviços de dados:

KVP

O serviço de Key Value Pair permite a manipulação de atributos através de uma aproximação de estados variáveis: *get*, *get response*, *set*, *set response*, *event transaction* e *event transaction response*. Adicionalmente, usa estruturas de dados etiquetadas usando XML comprimido. A sua estrutura é rígida.

Cada trama contém a seguinte informação:

- **Número de Sequência de Transacção** Número na sequência de transacção.
- **Tipo de Comando**
Indica o que a aplicação deverá fazer com a informação da trama: *Set*; *Event*; *Get + ACK*; *Set + ACK*; *Event + ACK*; *Get Response*; *Event Response*.
- **Tipos de Dados do Atributo**
Indica se o formato é *Unsigned 8-bit integer*, *Signed 8-bit integer*, *Unsigned 16-bit*, *Signed 16-bit integer*, *Semi precision*, *Absolut time*, *Relative time*, *Character string* ou *Octet string*.
- **Identificação do Atributo**
Indica o atributo alvo.
- **Código de Erro** (opcional)
É incluído somente nos comandos de resposta e pode indicar um dos seguintes valores: *Success*; *Invalid Endpoint*; *Unsupported attribute*; *Invalid command type*; *Invalid attribute data length*; *Invalid attribute data*; ou *Application defined error*.
- **Dados do atributo** (tamanho variável)
A informação aqui contida varia com o tipo de comando, o tipo de dados do atributo e da descrição do dispositivo usada.

MSG

O serviço de Message é usado para compatibilizar protocolos proprietários que não mapeiem bem KVPs. É transportada pelo mecanismo do KVP, tendo a diferença de não assumir conteúdo na trama de dados APS – estrutura de forma livre.

Cada trama contém a seguinte informação:

- **Número de sequência da transacção** – Número na sequência d transacção

- **Tamanho da transacção** – especifica o número de octetos usado.
- **Dados da transacção** – Possui tamanho variável (e a não ser que ambos os dispositivos suportem fragmentação, terá de ser inferior a *maxCommandSize*) e contém informação específica definida no *profile*.

Ambos os formatos são associados a um *clusterID* e só o *profile* da aplicação irá decidir quais os formatos das tramas usados para transferir qual informação, decidindo ainda o formato das tramas MSG.

[11] e [12]

3 Selecção de Hardware

3.1 Helicomm EZ-Net DevKit 1000

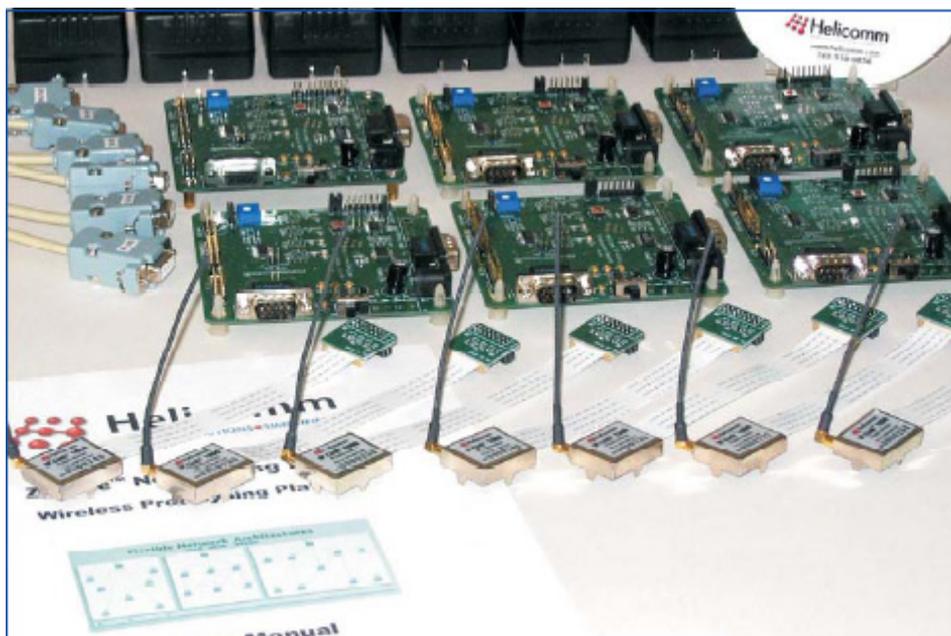


Fig. 3.1 Helicomm EZ-Net DevKit 1000

A Helicomm é uma empresa fundada em 2002 e dedicada ao wireless networking, através de kits de desenvolvimento para soluções adaptadas às necessidades do cliente. A estes kits podem ser adicionados módulos, para uma expansão de rede.

O kit EZ-Net DevKit 1000 contém 7 módulos compactos de comunicação wireless IP-Link 1000, 6 placas de interface com PC / sensor, o software específico Helicomm de configuração individual dos módulos e da rede (WIN-View Network Management Software), assim como toda a documentação, fontes de alimentação e cabos. As placas de interface possibilitam uma comunicação RS-232, suportando uma transmissão síncrona ou assíncrona de fluxos de dados – o que seria manifestamente interessante no caso deste projecto. O sistema é baseado em microcontrolador da família 8051, o que também poderia ser vantajoso dado esta família de microcontrolador já ter sido objecto de estudo no âmbito de algumas cadeiras. Para além disso, inclui um software específico que permite a configuração e controlo da rede, quer em termos individuais (cada módulo) quer globais

(topologia, tabelas de encaminhamento, pacotes trocadas, ...), assim como efectuar testes à mesma – através de simulação de rupturas entre nós, por exemplo.

No entanto, esta solução de desenvolvimento não se torna viável, dado apresentar-se com um custo de aquisição bastante elevado (\$2500 USD), possivelmente resultado de ser um kit que fornece à partida um considerável número de módulos de comunicação.

3.2 Silicon Labs 2.4 Ghz ZigBee Development Kit



Fig. 3.2 Silicon Labs 2.4 Ghz ZigBee Development Kit

Do fabricante Silicon Labs, este kit de desenvolvimento inclui seis nós wireless ZigBee completos, cada um deles com leds e botões para monitorização e controlo, sendo os mesmos baseados em microcontrolador da família 8051 e com possibilidade de alimentação dual: através de fonte de alimentação DC ou bateria de 9V. Cada um dos módulos possui dimensões compactas, o q é relevante para as possíveis aplicações deste projecto. Inclui ainda uma porta JTAG para programação do microcontrolador e interface USB.

Para além disso, inclui todo o software necessário ao desenvolvimento do projecto, incluindo o Silicon Laboratories Integrated Development Environment (IDE), o Keil Software 8051 Development Tools (assembler, linker e compilador de C) assim como as bibliotecas ZigBee Application Programming Interface (API). Faz parte do seu conteúdo também um driver para a criação de portas de comunicação (COM) virtuais. O programa de configuração da rede possui um interface gráfico, suportando as diferentes topologias previstas e incluindo até 13 modelos de estrutura de rede já pré-definidos.

Estão incluídos também todos os acessórios que poderão ser necessários, nomeadamente cabos, baterias e ainda 2 fontes de alimentação com conversão AC-DC – para alimentação através da rede eléctrica.

No entanto, esta solução apresenta um custo que, embora inferior ao do kit Aerocomm, se revela considerável – cerca de \$950 USD. Está disponível um kit de preço mais acessível, no qual se disponibilizam apenas dois nós ZigBee. Para além de que este último se revela insuficiente para que se efectue posterior teste de uma rede de três elementos (pelo menos), ainda assim o seu preço não se torna muito apelativo.

3.3 Data Hunter Tick RS232 Parasitically Powered Data Radio



Fig. 3.3 Data Hunter Tick RS232 Parasitically Powered Data Radio

Este dispositivo, comercializado pela empresa Data Hunter, revela-se bastante interessante, seja como elemento-acessório a uma rede já existente ou constituindo-se a própria rede com estes nós.

É alimentado de forma “parasita” através da porta-série do dispositivo ao qual se acopla e nele estão incorporadas todas as funções e dispositivos necessários à transmissão de informação, incluindo antena, transceiver, amplificador de potência RF e microcontrolador. De tamanho extremamente compacto e peso muito reduzido (35 gramas), a sua instalação exige apenas acoplamento à porta-série da entidade com a qual pretendemos estabelecer comunicação, suportando o protocolo RS-232.

Possui também um software específico que permite a configuração de todos os parâmetros dos distintos módulos e da rede, tais como taxa de transmissão (baud rate), paridade, tipo de controlo de fluxo, atribuição de endereços, número de tentativas de reenvio de um pacote, entre outros.

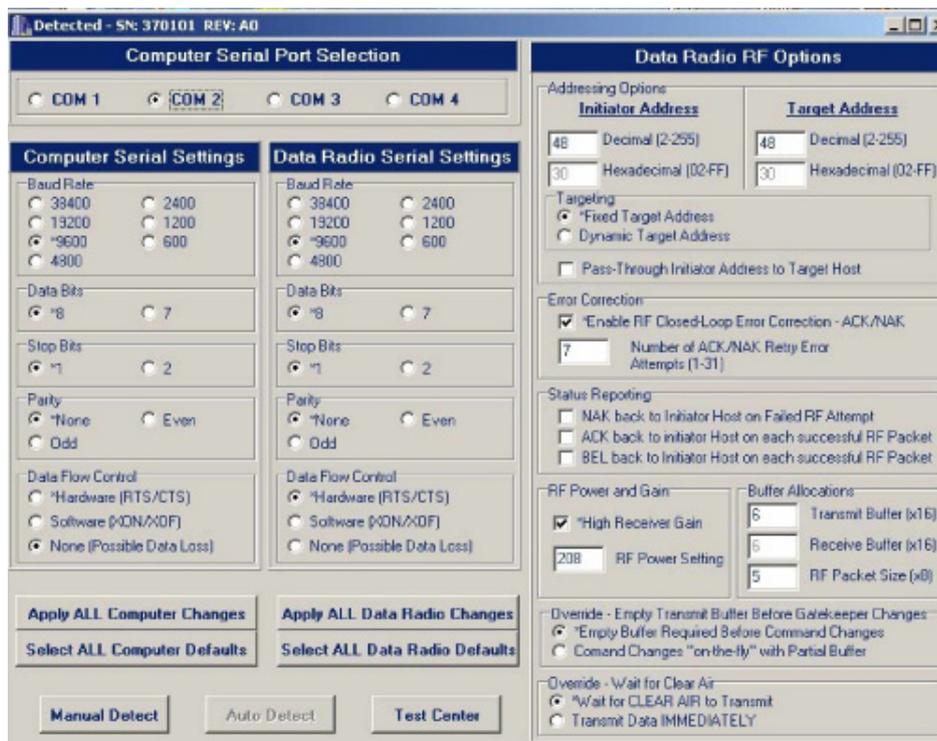


Fig. 3.4 Software de configuração Data Hunter

Numa análise mais aprofundada do produto, é, no entanto, possível verificar que a informação técnica apresenta lacunas. Entre outros, as topologias de rede admissíveis para a rede não são referidas e o consumo energético efectuado pelo aparelho levanta algumas dúvidas. Este último aspecto é de importância vital, dado que é necessário que esteja afastada a possibilidade de se causar danos no

dispositivo ao qual se acopla, ao tratar-se de uma alimentação-parasita via porta-série.

Por outro lado, e à data de realização deste estudo, a referência que suporta a operação através de protocolo ZigBee está anunciada como “brevemente em comercialização”, não sendo especificada uma data concreta. O preço apontado é de \$179 USD, o que, dadas as características do dispositivo, poderá ser considerado aceitável.

Sendo um conceito bastante interessante, fica a ideia de que estará ainda numa fase de desenvolvimento ou aperfeiçoamento, daí a sua disponibilidade como 100% ZigBee não ser ainda imediata e os detalhes técnicos não estarem ainda totalmente publicados e esclarecidos.

3.4 Maxstream XBee ZigBee RF Modules



Fig. 3.5 Maxstream XBee ZigBee Module

A empresa Maxstream disponibiliza os módulos transceiver de nome XBee, que operam na banda de 2.4Ghz, para integração em sistemas que pretendam transmitir informação por RF.

Suportando o protocolo ZigBee e diferentes topologias, os módulos caracterizam-se por ausência de configuração externa (“programação” por software próprio), sendo possível realizar as operações pretendidas através de comandos AT e de tramas específicas de controlo. Assim, destinam-se a interactuar com microcontroladores, através de uma interface SPI, encarregando-se da transmissão/recepção dos dados pretendidos (e estando o protocolo implementado no microcontrolador do próprio módulo).

Apresentam dimensões muito reduzidas (inferiores a 3cm x 3 cm) e são em 2 versões distintas (XBee / XBee Pro), as quais diferem na potência de transmissão. Ao XBee está associados uma potência de 1mW e um alcance de 100m ou 30m (outdoor / indoor), enquanto o XBee Pro opera com uma potência de 60mW, o que lhe permite alcançar um raio de transmissão até 1,6km ou 100m (outdoor / indoor). Em termos de consumo, são dispositivos que apresenta um consumo muito reduzido (inferior a 10 μ A) quando em estado *sleep*, subindo no entanto a necessidade de alimentação aquando da recepção/transmissão. No âmbito deste parâmetro, e como se torna evidente, é o XBee quem apresenta menor consumo energético – corrente de aproximadamente 50mA quando em estado transmissão/recepção (para uma tensão de alimentação de 3,3V).

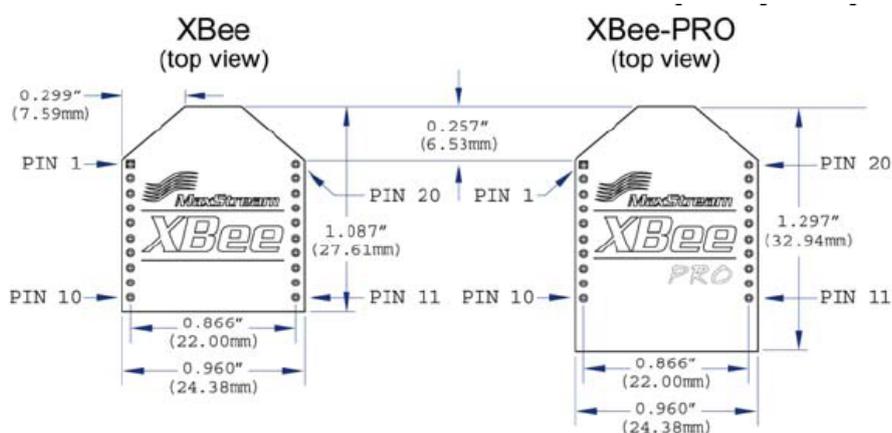


Fig. 3.6 Vista Frontal XBee / XBee Pro

XBee & XBee-PRO (side views)

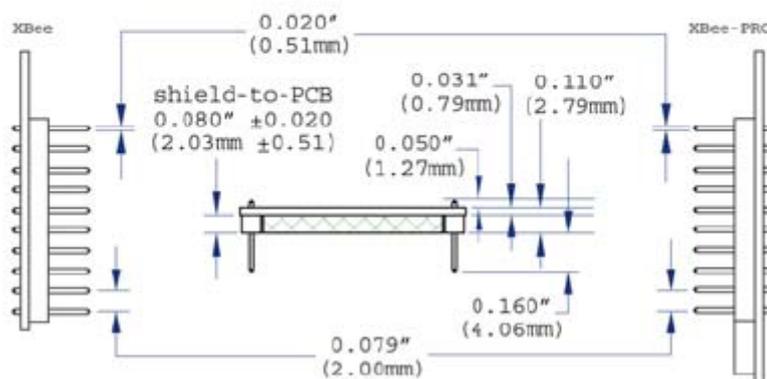


Fig. 3.7 Vista Lateral XBee / XBee Pro

Em termos de custo de aquisição, o módulo XBee é comercializado por \$19,00 USD, enquanto o XBee Pro se cifra nos \$32,00 USD. Cabe ao utilizador a eleição do tipo de antena que pretende que o módulo inclua: whip (pequeno dipolo), impressa no próprio chip ou ligação através de conector U.FL.

4 Desenvolvimento do trabalho

4.1 Sistema PIC / XBee

Por cumprir os requisitos necessários, o transceiver XBee da Maxstream foi a solução inicialmente elegida para o trabalho. Uma vez que o XBee apresenta um consumo mais reduzido assim como um preço inferior, seria perfeitamente aceitável, nesta fase, trabalhar com o XBee. No caso de posteriormente se verificar que o alcance era insuficiente, poderia ser efectuada uma troca da versão do transceiver, uma vez que o XBee e o XBee PRO são, em termos de ligação eléctrica e operação, perfeitamente intercambiáveis.

Desta forma, pretendia-se realizar um sistema que integrava o transceiver Xbee e um microcontrolador. A comunicação com os dispositivos a monitorizar / controlar seria realizada por porta-série, recorrendo ao protocolo RS-232, numa transmissão assíncrona de dados (UART), cabendo ao microcontrolador efectuar o coordenação dos fluxo de dados (sensor ↔ transceiver). O esquema representativo de tal sistema encontra-se representado na figura Fig. 4.1:

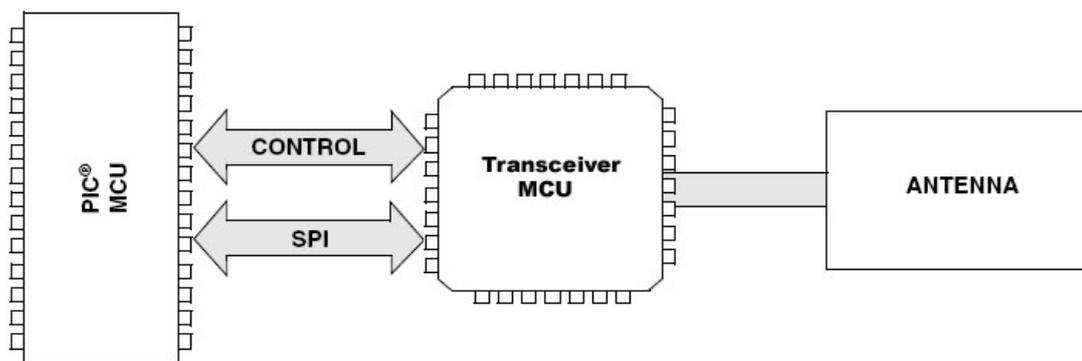


Fig. 4.1 Esquema de nó ZigBee constituído por PIC, Transceiver e Antena

Na interface entre o microcontrolador (elegido entre a família PIC) e o dispositivo a sensorizar, seria ainda necessário utilizar um módulo de conversão dos níveis eléctricos da transmissão-série, respeitando o standard RS-232.

4.1.1 O microcontrolador PIC

Após algum estudo, foram identificadas as características essenciais que o microcontrolador teria de apresentar (fundamentalmente uma forma de memória não-volátil, uma interface UART e um consumo energético reduzido, sendo também desejável uma simplicidade ao nível do número de pinos de interface).

Efectuando uma busca generalizada, concluiu-se que um microcontrolador que satisfazia amplamente os requisitos necessários era o microcontrolador de 8bits Microchip PIC18F330, o qual apresenta uma Data EEPROM de 128KB, interface EUSART, oscilador interno de 8Mhz, tensão de alimentação compatível com a do módulo XBee e consumo designado por “*Nanowatt*”. Além disso, o seu custo acessível (abaixo dos 2,50€) e a sua disponibilidade comercial tornaram-no uma opção viável.

4.1.2 O conversor MAXIM MAX 3221

Em termos de componentes, restava apenas seleccionar um módulo para conversão dos níveis eléctricos da transmissão via porta-série. Para este efeito existem múltiplos circuitos integrados de diversos fabricantes, sendo um deles o módulo MAXIM MAX3221.

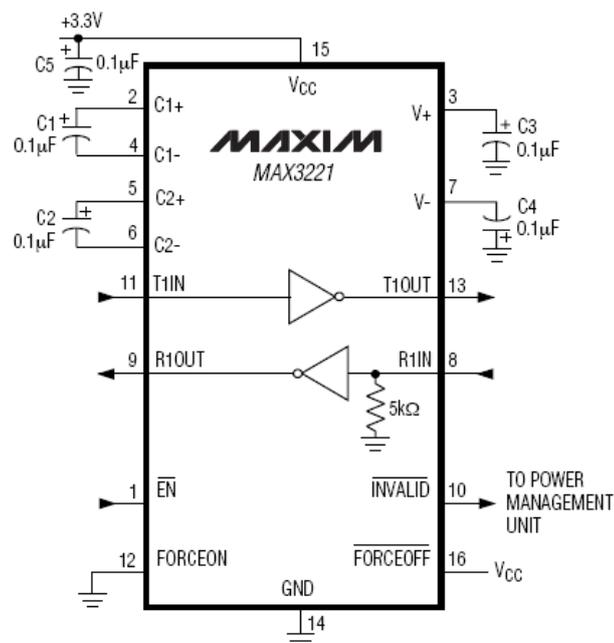


Fig. 4.2 Diagrama de pinos do MAXIM MAX3221

Para um melhor esclarecimento sobre o modo de funcionamento deste dispositivo, salienta-se as funções de alguns dos seus pinos:

T1IN – Entrada do transmissor TTL / CMOS

R1IN – Entrada do receptor RS-232

T1OUT – Saída do transmissor RS-232

R1OUT – Saída do transmissor TTL/CMOS

FORCEON – força o circuito a funcionar

/FORCEOFF – força o circuito a entrar no modo shutdown

/INVALID – pino que informa se existe algum sinal RS-232

Tendo como função a conversão entre sinais RS-232 e níveis lógicos CMOS, este circuito apresenta as seguintes características:

- é capaz de suportar uma taxa de transmissão de até 120kbps;
- consumo reduzido *Micropower* e com modo *Auto-Shutdown*, no qual apenas é necessária uma corrente de alimentação na ordem de 1 μ A;
- suporta uma taxa de transmissão de até 120kbps (garantindo uma slew rate de 6 V / μ A).

Dado que o consumo é uma das preocupações ao nível do projecto, o consumo reduzido deste integrado e a suas funções de shutdown como forma de controlar ainda mais a potência consumida, são características preponderantes para a sua escolha.

De destacar ainda que por este integrado poder ser alimentado a 3.3V (mesma tensão que o microcontrolador PIC) e o nível eléctrico RS-232 ser de 5.5V, recorre a uma técnica de *charge pumps* para poder reproduzir estes níveis eléctricos. As “bombas de corrente”, que não são mais que conjuntos de condensadores numa disposição específica, que operam quando a tensão de alimentação do circuito é inferior a 5.5V. O próprio fabricante disponibiliza informação acerca das capacidades a colocar, em função o intervalo de tensão de alimentação (no caso de 3.3V será 0,1 μ F em ambos os condensadores, identificados por C1 e C2).

O custo deste dispositivo situa-se abaixo dos 1,50€, pelo que não se revela um componente dispendioso.

Assim, o conjunto de hardware necessário à constituição deste sistema revela-se com um preço bastante aceitável e com a vantagem de ser inteiramente controlado pelo projectista, sem ter uma dependência tão grande de software proprietário do fabricante.

No entanto, foi após a encomenda do módulo XBee que esta alternativa se viu inviabilizada.

Ao contactar o representante autorizada pela comercialização em Portugal dos produtos Maxstream (empresa Lusomatrix), foi recebida a informação de que os módulos XBee ZigBee efectivamente ainda não estavam comercialmente disponíveis – ao contrário da informação disponível no site da Maxstream.

Ainda que numa versão Beta, no site da Maxstream os módulos XBee ZigBee surgiam já como disponíveis, à data do estudo de hardware realizado. Tendo sido constatado que as funções indisponíveis na versão Beta não eram críticas para a realização do projecto, e podendo posteriormente os transceivers serem trocados pela versão completamente concordante com o protocolo ZigBee. Isto afectaria fundamentalmente o consumo energético, dado que os estados de sleep (através dos dispositivos RFD) ainda não estariam disponíveis, mas, para uma primeira versão experimental, não se revelava prioritário – após a realização desta, e uma vez que não influenciaria o processo de transmissão, seriam facilmente trocados os transceivers pela versão definitiva.

Os módulos XBee ZigBee (Beta) só estariam efectivamente disponíveis em Abril de 2007 (Maxstream, USA) e, segundo informação da Lusomatrix, o prazo de entrega seria entre 4 a 6 semanas.

Tal invalidava, por requisitos temporais óbvios, a utilização deste hardware no projecto.

Foi ainda contactado directamente o responsável da Maxstream para as vendas na Europa, o account manager Mr. Rob Moore, assim como o representante em Espanha (delegação da Digi International), através do Sr. Carlos Saenz, tendo sido confirmadas as informações anteriormente dadas pela Lusomatrix e ainda indicada esta empresa como a forma efectivamente mais rápida de poder obter o produto no nosso país.

Foi nesta fase que se começou a verificar que, apesar de o protocolo ZigBee ter sido já oficialmente lançado há cerca de 2 anos, ainda não se encontra consolidado no mercado – quer porque o hardware disponível frequentemente possui firmware ainda versão beta ou ainda se encontra em desenvolvimento, quer

O kit PICDEM Z da Microchip tem como objectivo o desenvolvimento e demonstração de aplicações do ZigBee, fornecendo para tal dois nós ZigBee para a criação de uma rede simples.

O kit utilizado, com a denominação DM163027-2, inclui duas placas de demonstração, as quais são (cada uma delas) constituídas por:

- Uma placa-mãe, (*motherboard*) AC163027-1;
- Uma placa-filha (*daughter-card*), de referência AC-163027-2, que contém o transceiver e uma antena PCB (do tipo F invertido); sendo portanto onde se efectiva a comunicação comunicação RF;

O motivo da “separação física” existente é o possibilitar o uso de tipos diversos de transceiver (ou a substituição deste). A comunicação é efectuada com o microcontrolador através de uma interface SPI. Neste kit em concreto, cada nó utiliza uma placa-acessório PICDEM Z equipada com o transceiver Chipcon CC2420 RF 2.4Ghz.

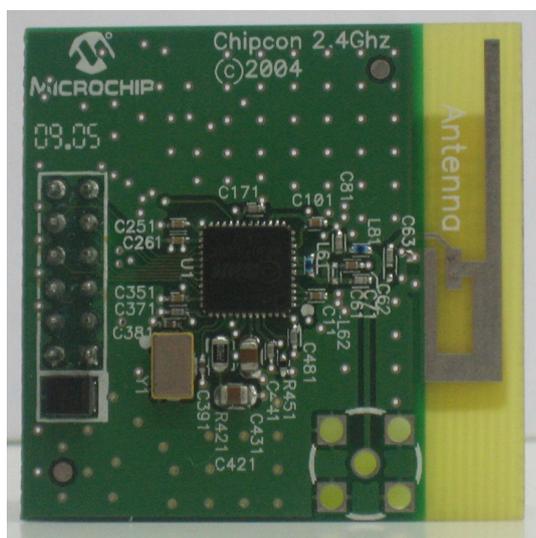


Fig. 4.4 Placa-filha, contendo transceiver Chipcon e antena PCB em F invertido

Cada nó inclui, através da sua placa-mãe, uma interface RJ-11, com o objectivo de utilizar o dispositivo MPLAB ICD2, para efectuar debugging e programação, e suporta ICSP (*in-circuit serial programming*).

O kit inclui ainda algumas outras funcionalidades, como é o caso:

- sensor de temperatura (sensor Microchip TC77), integrado na própria placa-mãe;
- interface RS-232;
- 2 LEDs e 2 botões, para interacção e demonstração.

Cada módulo (placa mãe + transceiver) inclui a possibilidade de alimentação por bateria de 9V ou através de fonte de alimentação DC, por bornes próprios. A tensão de alimentação disponibilizada é de 3.3V tanto para o microcontrolador como para o transceiver. O alcance de transmissão situa-se entre os 30 metros (*indoor*) e os 150 metros, em campo aberto.

O custo deste kit de demonstração situa-se nos 200€.

4.2.1 O microcontrolador PIC18LF4620

O microcontrolador utilizado, que se encontra na placa-mãe de cada nó, é um PIC da família 18, concretamente, PIC18LF4620.

Não se pretendendo efectuar um estudo exaustivo sobre o mesmo (e estando, para tal, disponível a sua *datasheet*) é, no entanto, importante destacar algumas das suas principais características:

- Nanowatt technology – num projecto que pretende um dispositivo de reduzido consumo energético, é importante um baixo consumo do microcontrolador. Este microcontrolador possui várias funcionalidades implementadas para que o seu consumo seja bastante diminuto, nomeadamente: *alternate run modes*, *on-the fly mode switching* e *multiple idle modes*. Esta última, p.ex., permite uma suspensão do núcleo do CPU do microcontrolador, enquanto os periféricos em utilização permanecem no seu estado funcional (levando a reduções de consumo de até 95%!).
- Interface USART endereçável com função *auto wake-up* através de *start-bit* e auto-deteção da taxa de transmissão;
- Suporta SPI e I2C;
- múltiplas opções em termos de tipo de oscilador, incluindo um interno;
- conversor A/D de 10 bits integrado;

- watchdog timer;
- 4 timers;
- 20 fontes de interrupção;
- 5 portas bidireccionais;
- memória flash programável de 64KBytes;
- 3986 bytes de SRAM de memória de dados;
- 1024 bytes de EEPROM para memória de dados;

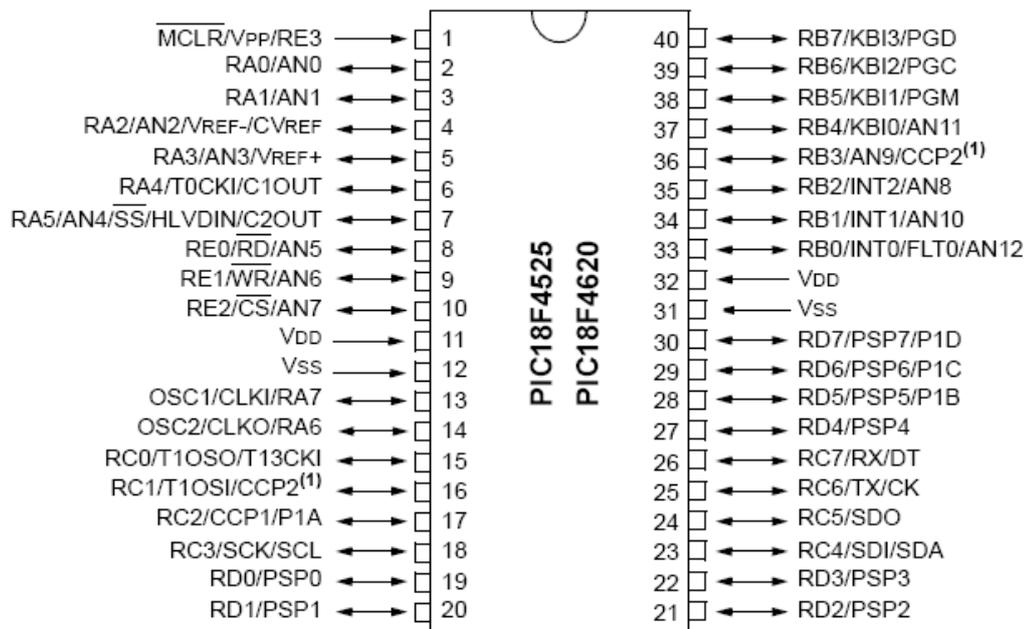


Fig. 4.5 Diagrama de pinos do PIC18LF4620

4.2.2 Interface Intraplaca e Pilha protocolar

Atentando na figura Fig. 4.1, podemos compreender conceptualmente o funcionamento de cada nó ZigBee.

O microcontrolador actua como SPI-master, implementando a camada MAC do protocolo ZigBee. A pilha protocolar recorre à sua memória flash interna para armazenar o endereço MAC, tabela de endereços e *binding table*. O transceiver encarrega-se da implementação da camada física assim como algumas funções MAC.

É possível ao utilizador alterar a configuração dos pinos do microcontrolador (para exercerem outras funções), o que não poderá é modificar os pinos alocados para funções de controlo SPI.

O interface entre o microcontrolador PIC 18LF4620 e o transceiver Chipcon encontra-se descrito na Tabela 4.1.

Pino PIC18LF4620	Pino Transceiver CC2420
RB0 [input]	FIFO
RB1 [input]	CCA [não utilizado]
RB2 [input]	SFD
RB3 [input]	FIFOP
RC0 [output]	CSn
RC1 [output]	VREG_EN
RC2 [output]	RESET
RC3 [output]	SCK
RC4 [input]	SO
RC5 [output]	SI

Tabela 4.1 PICDEM Z: Interface microcontrolador/transceiver

A pilha protocolar da Microchip, escrita em C, apresenta-se estruturada em camadas, tal como o especificado no protocolo ZigBee. A camada corresponde um ficheiro diferente, sendo as API's definidas num include file. Muitas das API's mais não são do que macros que chamam funções da camada imediatamente inferior. Na sua essência, o que a pilha protocolar faz é com que a aplicação principal possa ser desenvolvida não se ocupando de funções ligadas à comunicação (e coordenação) RF.

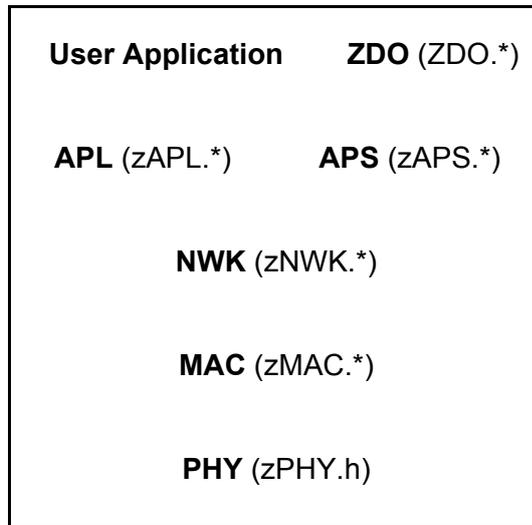


Fig. 4.6 Pilha protocolar Microchip: esquema de arquitectura e ficheiros

4.2.3 Esquema Eléctrico da Placa-Mãe do kit PICDEM Z

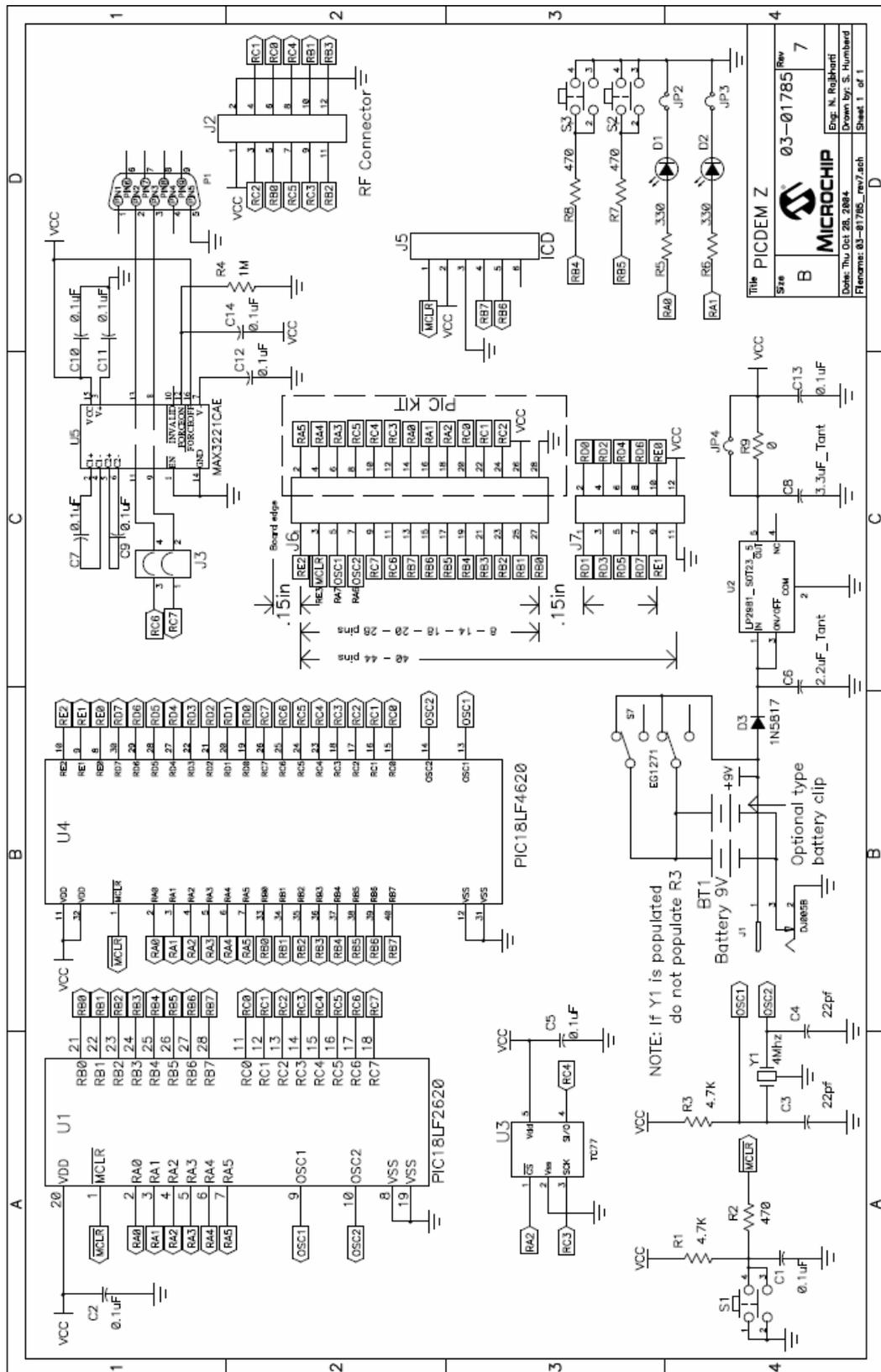
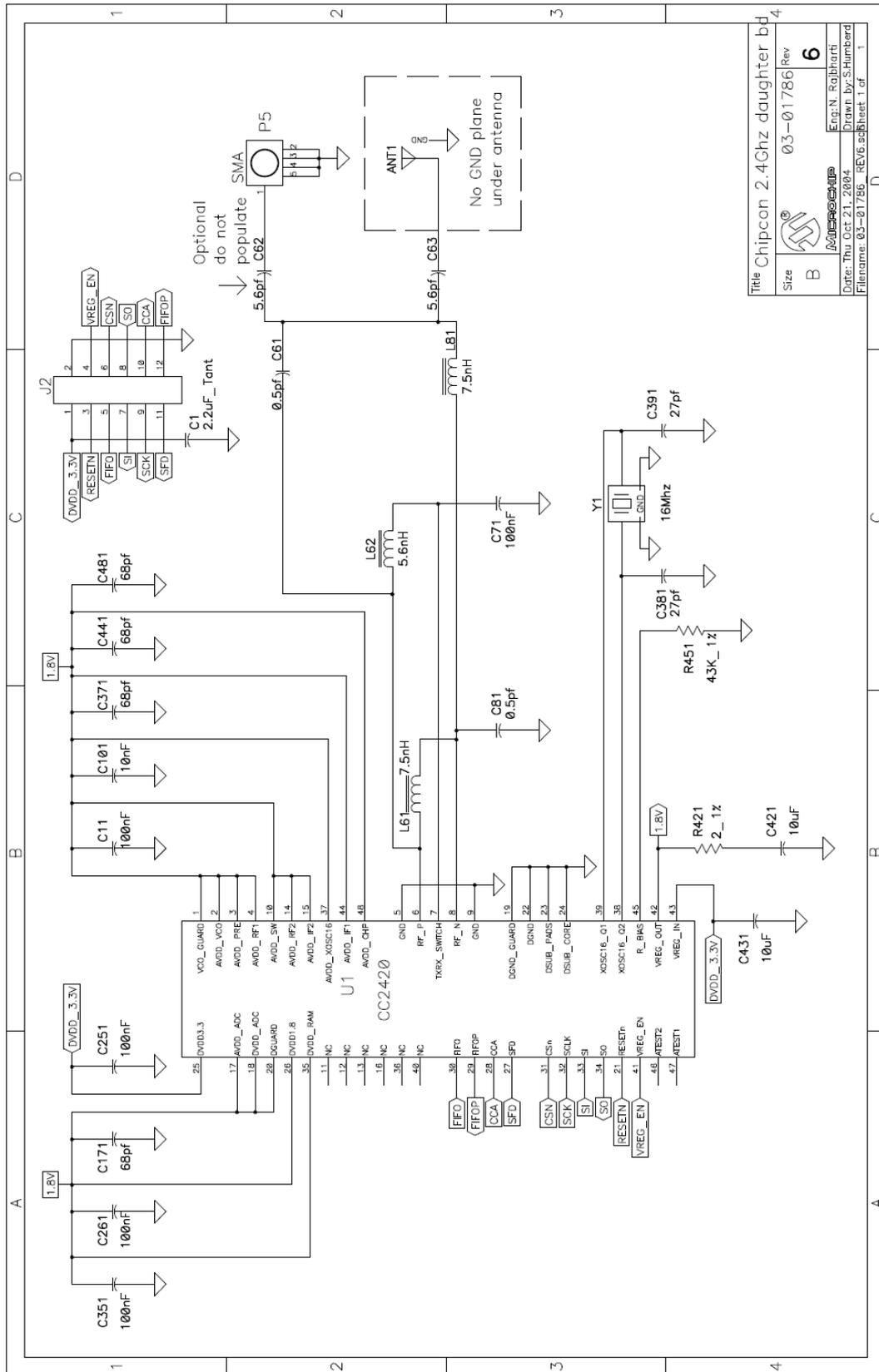


Fig. 4.7 Esquema Eléctrico da Placa-Mãe do kit PICDEM Z

4.2.4 Esquema Eléctrico da Placa-Filha do kit PICDEM Z



Title		Chipcon 2.4Ghz daughter bd	
Size	B	Rev	6
Date: Thu Oct 21, 2004		Eng: N. Rajbharti	
Filename: 03-01786_REV6.sch		Drawn by: S.Humberd	
		Sheet 1 of 1	

Fig. 4.8 Esquema Eléctrico da Placa-Filha do kit PICDEM Z

4.3 Ferramentas de desenvolvimento

Em termos de ferramentas de desenvolvimento trabalhou-se com o mesmo fabricante do kit PICDEM Z, a Microchip.

Ao nível de software foi utilizado o ambiente de desenvolvimento MPLAB IDE v7.50 em conjunto com o compilador C18 Student Edition, específico para a programação da família 18 dos microcontroladores PIC de 8bits. No caso deste último, existia a limitação de 60 dias de utilização, após a qual eram reduzidas as funções e conjunto de instruções suportados.

No que toca ao hardware, foi utilizado o MPLAB ICD2, o qual se conecta ao PC através de porta USB, suportando ICSP™ (In Circuit Serial Programming™), tecnologia proprietária da Microchip que permite em tempo real a programação, execução e *debugging* de programas – baseando-se na utilização de pinos dedicados para a comunicação com o microcontrolador e firmware específico *on-chip*.



Fig. 4.9 Módulo MPLAB ICD2

4.4 Utilização de Programa para leitura da porta-série. Balança Mettler-Toledo.

Entre os vários programas disponíveis para leitura de porta-série, utilizou-se neste trabalho o programa HyperTerminal, vulgarmente disponível com o próprio sistema operativo Windows (dependendo, no entanto, da versão Windows).

De referir que foi utilizada nesta fase do trabalho uma Balança Mettler-Toledo Spider 1, com capacidade de envio de dados através de RS-232. A configuração deste equipamento não foi muito imediata dado o seu menu ser pouco inteligível e o manual de instruções não estar disponível.

Depois de devidamente efectuada a selecção do modo de funcionamento apropriado e uma correcta parametrização, este equipamento emitia uma string de 20 caracteres com o valor da pesagem, a cada estabilização da pesagem sobre o seu prato.

Para uma melhor informação sobre este equipamento, assim como o estudo e configuração sobre ele realizados, sugere a leitura da secção 1 do anexo A.

4.5 Interacção ligação série

Dado que se pretende uma interacção com os sensores aos quais se acople os módulos, “lendo” os dados provenientes dos mesmos e comunicando-os a um “módulo central”, tornou-se imperativo que a familiarização com a programação do PIC e com as ferramentas de desenvolvimento MPLAB fosse iniciada com a interacção da porta-série.

4.5.1 Escrita via Porta-Série

Inicialmente pretendeu-se apenas que fosse possível ler, através do programa HyperTerminal, strings enviadas via porta-série pelo PIC.

Embora aparentemente simples, foi necessário um tempo considerável para a realização desta tarefa dado que foi também a fase de estudo da *data sheet* do PIC. Por outro lado, foi também um primeiro contacto com a programação de um microprocessador totalmente desconhecido até então assim como de um ambiente de desenvolvimento novo (e algo complexo).

Nas primeiras utilizações surgiram problemas com a utilização do dispositivo de programação/debug, o ICD2, nomeadamente porque, apesar de alimentado por USB, é necessário fornecer alimentação externa à placa (e ainda que esta não apresente, quando simultaneamente conectada ao ICD2 e à alimentação externa, qualquer sinal luminoso de estar em operação). Efectivamente, e embora o ICD2 apresente alimentação via USB, não é suficiente para suprir os requisitos do circuito. Esta situação aliou-se a uma outra, que estava relacionada com a

incorrecta configuração do dispositivo ICD2, devido a um ficheiro incorrecto de definição do dispositivo a programar, pelo que de ambas resultava continuamente um aviso de falha ao tentar efectuar programação (ou realizar debug).

Ultrapassada esta adversidade, conseguiu-se realizar a correcta programação do microcontrolador, embora o código descrito não produzisse a finalidade pretendida (e, conseqüentemente, não existia a correcta visualização de string no HyperTerminal).

O que estava a suceder era uma incorrecta configuração do tipo de oscilador a ser utilizado. Conforme se pode verificar no esquema eléctrico da placa-mãe, este módulo possui integrado um oscilador externo com cristal de 4Mhz. Para uma correcta utilização do mesmo é necessário que a linha *#pragma code OSC* inclua a referencia XT (relativa a “externo”) e não qualquer outro tipo de informação como malha RC ou oscilador interno.

Na realização desta programa recorreu-se já a funções específicas do compilador C18. A sua função *OpenUSART()* evita uma configuração “linha-a-linha” de cada um dos bits referentes à comunicação-série, sendo cada uma das opções pretendidas configurada na própria invocação da mesma. Para a utilização desta função (e outras relacionadas com a porta USART, *Universal Synchronous Asynchronous Receiver Transmitter*) é necessário incluir a biblioteca “usart.h”.

Para uma mais fácil percepção, os parâmetros a definir foram:

USART_TX_INT_OFF – interrupção devida a transmissão de dados via porta-série desactivada

USART_RX_INT_OFF – interrupção devida a recepção de dados via porta-série desactivada

USART_ASYNC_MODE – modo de operação assíncrono

USART_EIGHT_BIT – transmissão/recepção de 8 *data bits*

USART_CONT_RX – recepção contínua

USART_BRGH_HIGH – utilização de taxa de transmissão elevada

Foi ainda necessário configurar o parâmetro denominado “sbprg”, com o valor relacionado com a taxa de transmissão (*baud rate*) pretendida. Estabeleceu-se que, nesta fase, a baud rate a utilizar seria 19200bps, pelo que

$$baud\ rate = \frac{F_{osc}}{16 \times (sbprg + 1)} \Leftrightarrow sbprg = \frac{4 \times 10^6}{9600 \times 16 - 1} \approx 12$$

Conseguiu-se então, após o ultrapassar destes passos, a escrita com sucesso na porta-série.

Este primeiro programa proporcionou um maior à-vontade na utilização de todo o sistema envolto na programação do PIC.

4.5.2 Programa de Leitura via Porta-série

No caso deste programa, pretendeu-se a recolha de dois caracteres, via ligação-série, por parte do microcontrolador e solicitados ao utilizador no programa HyperTerminal (sendo digitados e comunicados PC → microcontrolador). O pedido de dois caracteres é realizado sucessivamente até que se introduza a combinação de caracteres 'qq'.

Foi concluído com sucesso e permitiu ultrapassar uma dificuldade que estava relacionada com o armazenamento de dados no microcontrolador. Foi ainda compreendido que o recurso à função *CloseUSART()*, que efectua o encerramento da ligação lógica, deve ser realizado com alguma precaução. Caso a esta mesma função seja invocada, a ligação é de imediato terminada, pelo que eventuais dados que se encontrem ainda na FIFO de saída já não poderão ser correctamente emitidos, gerando-se situações aparentemente “anómalas”.

Para o evitar, o mais adequado é o recurso à função *BusyUSART()*, que devolve diferentes valores consoante a ligação série ainda esteja ou não em utilização. Interpondo o teste deste valor antes da invocação da função *CloseUSART()*, é possível forçar o programa a que a aguardar o fim da comunicação dos dados, após o qual poderá então ser encerrada a ligação.

4.5.3 Programa de Leitura via Porta-Série com Interrupção

No caso anterior, o programa ficava “encravado” a aguardar a recepção de cada um dos caracteres. Uma vez que, no contexto deste projecto, deverá ser a aplicação a efectuar o “tratamento” dos dados aquando da sua recepção, realizaram-se as alterações necessárias para que tal acontecesse.

Desta forma, e ao contrário dos programas anteriores, foi necessário recorrer ao uso de interrupções, em concreto a interrupção de recepção de dados via porta-série. Assim, foi necessário atentar nos bits de configuração de interrupções em geral e das interrupções geradas pela porta-série.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

Fig. 4.10 Peripheral Interrupt Enable Register (PIE1)

No registo de activação das interrupções dos periféricos (PIE1), é necessário colocar no nível lógico alto o bit RCIE (*EUSART Receive Interrupt Enable* bit) para que seja activada a interrupção por recepção de dados na porta série. Tal é efectuado pela própria função OpenUSART(), já atrás descrita, quando na sua invocação está presente a indicação USART_RX_INT_ON.

É também preciso atribuir à interrupção gerada um nível de prioridade alto, por forma a que o seu atendimento seja não seja preterido face a outras interrupções. No registo RCON, é necessário activar o bit IPEN (*Interrupt Priority Enable*) o qual permitirá o estabelecimento de prioridades no atendimento das interrupções.

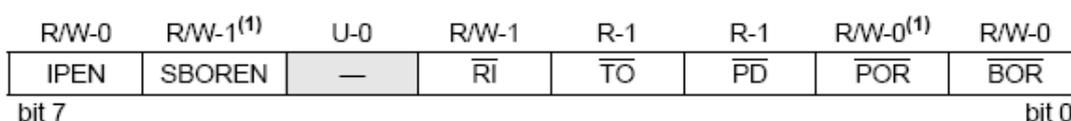


Fig. 4.11 Reset Control Register (RCON)

Adicionalmente, é necessário que sejam activadas todas as interrupções de prioridade elevada, o que é feito através do bit GIEH (*Global Interrupt Enable High*) do registo de controlo de interrupções (INTCON), que se encontra representado na Fig. 4.12 Interrupt Control Register (INTCON).

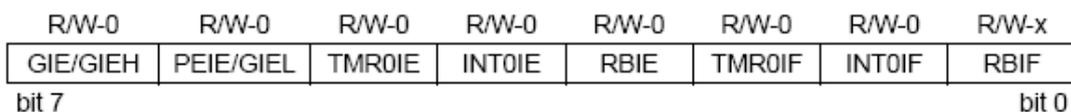


Fig. 4.12 Interrupt Control Register (INTCON)

É ainda fundamental atribuir o nível alto em termos de prioridade à interrupção pretendida. Corresponde isto a activarmos, no registo relativo à prioridade de interrupções originadas por periféricos (IPR1), o bit referente à recepção de dados via porta USART – bit identificado por RCIP (*EUSART Receive Interrupt Priority*).

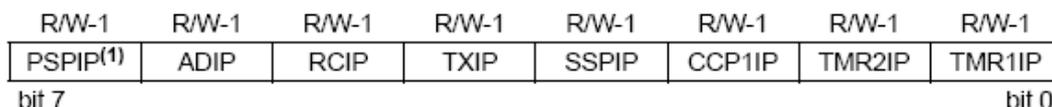


Fig. 4.13 Peripheral Interrupt Priority Register 1 (IPR1)

No atendimento a uma interrupção de alta prioridade, o microcontrolador vai, por predefinição, para o endereço 0008h. Foi então necessário definir, neste

endereço, instrução para que a interrupção de recepção de dados via porta-série seja atendida pela respectiva rotina de atendimento dessa mesma interrupção (*rx_handler*).

Foi feita a adaptação do programa anteriormente realizado e testou-se. Em seguida, procedeu-se à criação de um outro programa especificamente para interagir com os dados provenientes da balança Mettler-Toledo Spider 1. Este dispositivo, disponibilizado para que com ele pudessem ser realizados testes, envia dados via porta-série sobre a pesagem.

Desta forma, o que se efectuou foi um processo armazenamento do conjunto de caracteres que seria enviado pela balança, sob a forma de string, aos quais se adiciona uma “etiqueta” (para identificação da proveniência, em termos de sensor, desses dados) e o seu envio via porta-série. Foi ainda necessário colocar a taxa de transmissão num valor suportado pela balança, tendo sido seleccionado 9600bps (e, portanto, parâmetro *sbprg* tomou o valor 25).

Conforme se constata, este é já um aproximar à aplicação final, na qual não os dados não serão enviados pelo módulo para a porta série, mas sim enviados via RF para um outro módulo (que assume papel de “módulo central”) – o qual procederá depois à emissão, na sua porta-série, dos dados recebidos.

4.6 Interacção entre nós ZigBee

Posteriormente à fase de interacção da porta-série, passou-se então à fase de interacção (comunicação) entre os dois nós ZigBee.

A versão mais actual da pilha protocolar da Microchip (v1.0-3.8) é denominada pela própria Microchip como “Zigbee Compliant”, ou seja, é uma versão final, estável e de acordo com o protocolo da ZigBee Alliance. No entanto, embora tal não seja explicitamente referido na documentação desta versão da pilha protocolar, a mesma só é compatível com a versão mais recente do kit PICDEM Z. Os nós do kit que foi disponibilizado para este trabalho encontram-se equipados com o transceiver Chipcon CC2420, no caso da versão mais recente do kit o transceiver utilizado é da própria Microchip, o MRF24J40, sendo a placa-filha denominada pela referência AC163027-4. Esta incompatibilidade foi verificada após a instalação da referida stack, em que o nó coordenador não inicia uma rede ZigBee, ficando a aplicação a aguardar que seja reconhecido um transceiver Microchip.

Dado não existir qualquer outra versão disponível no site de suporte deste fabricante, foi necessário contactar directamente a Microchip, solicitando esclarecimento acerca desta questão. Paralelamente, foi-se desenvolvendo a interacção via porta-série, descrita nas secções anteriores, assim como se realizou o estudo e configuração da balança Mettler-Toledo (a qual é detalhada no correspondente anexo a este relatório).

Foi enviada, entretanto, por parte da Microchip a última versão compatível com o transístor existente: a versão 1.0-3.6, que surge apenas com algumas correcções face à 1.0-3.5, na qual se baseia. Aquando do seu recebimento, foi verificado, de forma sumária (dado se prosseguir com a tarefas que encontravam a decorrer), que em termos de coordenador estava funcional e que era realmente compatível com o transceiver Chipcon.

Foi quando efectivamente se iniciou o desenvolvimento da aplicação de comunicação entre nós que surgiram os problemas mais relevantes.

A pilha protocolar fornecida pela Microchip surge com alguns pequenos programas de demonstração de funcionalidades, para que o utilizador tenha a possibilidade não só de se inteirar das potencialidades da mesma, mas também para facilitar um primeiro contacto, em termos de desenvolvimento de código, invocando-se funções suportadas pela pilha protocolar.

O que se verificou foi que uma vez instalada uma aplicação de demonstração e, evidentemente, a pilha protocolar, o comportamento não foi o esperado. O nó coordenador inicia a rede ZigBee indicando, através da sua porta-série, que a mesma foi criada assim como o respectivo identificador atribuído à rede (*PAN ID*). No entanto, é no nó Endpoint que a situação se verifica ser anómala. O nó não se consegue juntar-se à rede criada, apresentando sucessivamente falha na tentativa de busca de uma rede disponível.

Através de alguns parâmetros de controlo, realizaram-se alterações a nível da comunicação, na tentativa de conseguir solucionar o problema, nomeadamente:

- #define ALLOWED_CHANNELS 0x07FFF800, de modo a permitir utilização de qualquer canal disponível;
- diminuição do número de dispositivos que podem ser aceites numa rede, através do parâmetro MINIMUM_JOIN_LQI, no ficheiro *zigbee.def*;
- alterações aos níveis energéticos da transmissão, nomeadamente no parâmetro MAX_ENERGY_THRESHOLD assim como colocação de PA_LEVEL no valor 0xFF (o que assegura a máxima potência de transmissão, embora não fosse provável este factor exercer influência negativa, dado as placas estarem a escassos 30 cm uma da outra);

Nenhuma das tentativas levadas a cabo surtiu efeito e, embora compreendido o processo de envio de dados num nó para outro, revelou-se até à data impossível quer testar quer ver o funcionamento de alguma das demonstrações fornecidas.

Na figura Fig. 4.16 *PICDEM Z: Processo de Envio de Dados entre dois nós* representa-se num diagrama o processo de envio (em termos de passos a realizar e funções protocolares a invocar) de um conjunto de dados entre um nó ZigBee e um outro nó de destino. Foi inclusivamente realizado algum código neste sentido, o qual se apresenta também em anexo, embora não tenha sido, pelo atrás descrito, possível testar o mesmo.

Conforme se refere na próxima secção, até à data de apresentação será feito o possível para que o projecto evolua, tentando que sejam ultrapassados os problemas relacionados com a pilha protocolar da Microchip.

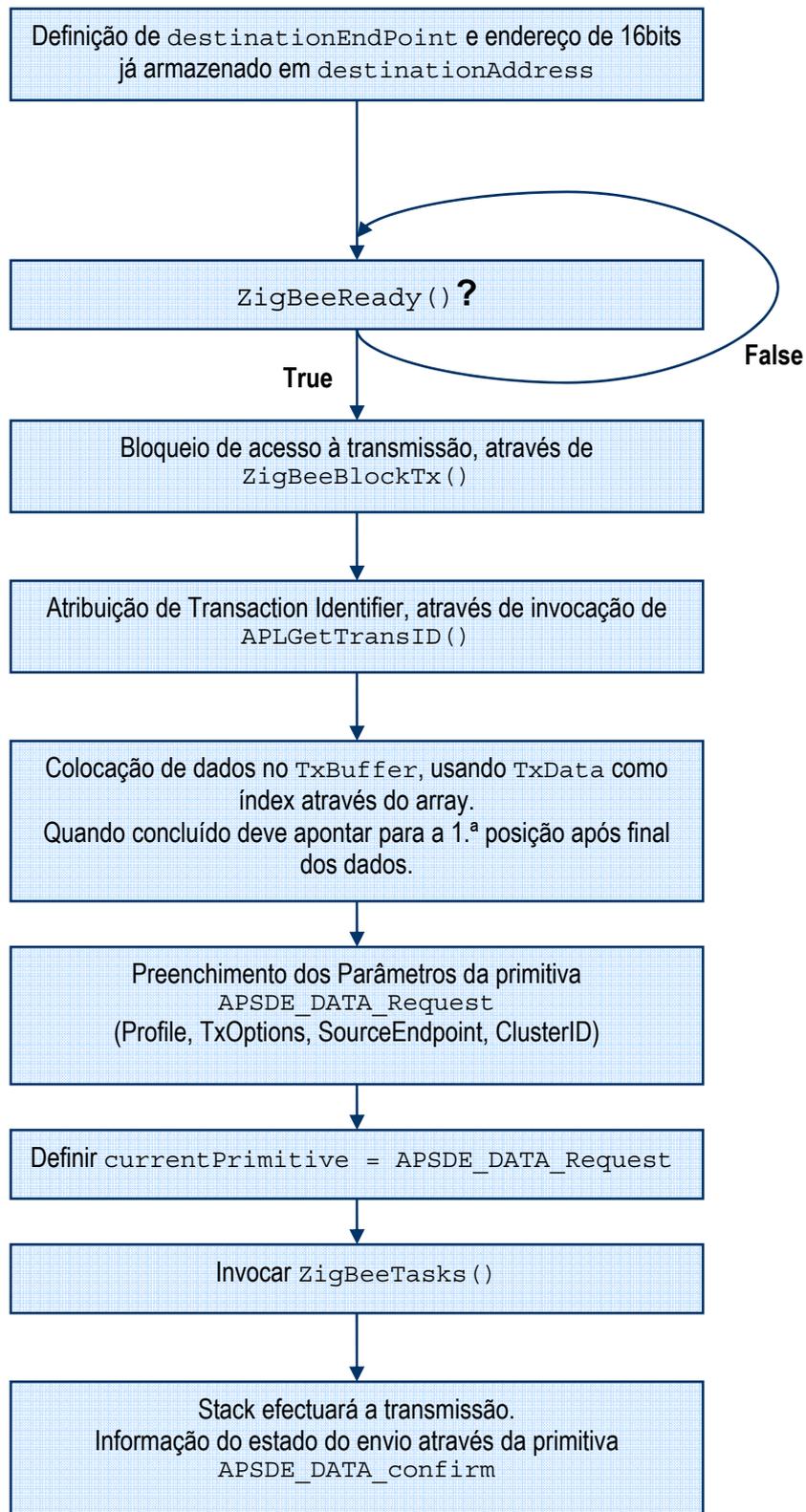


Fig. 4.16 PICDEM Z: Processo de Envio de Dados entre dois nós

4.7 Calendário do trabalho realizado

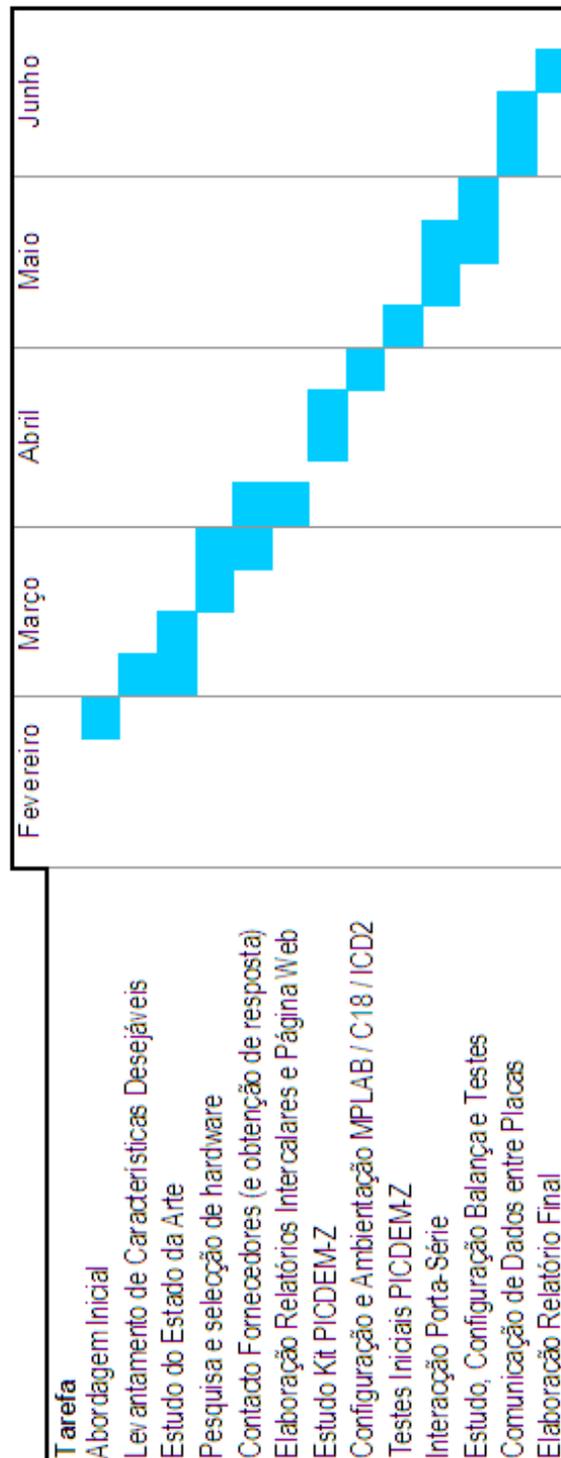


Fig. 4.17 Calendário do Trabalho Realizado

5 Conclusão

Este projecto abordou um tema bastante interessante, dado que a área das comunicações tem sido um dos sectores com mais desenvolvimento nos últimos anos / décadas. Dentro dela, as comunicações wireless assumem cada vez mais uma relevância maior.

Existe uma enorme diversidade no campo de aplicação das comunicações por rádio-frequência, quer em termos de utilização pessoal ou doméstica, quer a nível industrial. Desde os telefones GSM / UMTS ou os avanços na área da domótica à generalização dos hotspot's de acesso à Internet de banda larga e à sensorização remota da pressão dos pneus do automóvel, cada uma delas faz parte do desafio constante da Engenharia Electrotécnica. Assim, o campo das telecomunicações e a área dos microprocessadores / sistemas digitais são claramente áreas aliciantes de trabalho, com objectivos de satisfação de necessidades e liderança na inovação.

Como estudante de Engenharia, este projecto, integrado na parte curricular da minha Licenciatura, permitiu-me um primeiro contacto mais aprofundado com o que é o projecto de Engenharia.

Proposto ao abrigo de uma parceria entre a empresa RealSabor e a F.E.U.P., efectuou-se um estudo sobre uma necessidade na linha de produção desta empresa – e que, certamente, será comum a muitas outras.

Após uma breve abordagem ao problema, efectuou-se o planeamento das tarefas a realizar.

A primeira fase foi um levantamento das especificações que deveriam ser satisfeitas. Uma vez que se pretende acoplar o módulo a dispositivos de sensorização, constituindo-se, com os dados destes, uma rede de comunicação a nível industrial, existiam características a ter em conta – nomeadamente aspectos como o consumo, as dimensões do módulo, a largura de banda e latência requeridas, as interfaces de interesse, entre outros. Nesta parte começaram a surgir algumas hipóteses sobre soluções a pesquisar e aprofundar, estimulando assim o aparecimento de novas questões sobre o problema.

Tendo em conta os resultados da fase anterior, foi iniciado um estudo sobre o “estado da arte” no que toca às tecnologias de comunicação wireless, nomeadamente o confronto entre os vários protocolos disponíveis.

Após selecção do protocolo ZigBee, pelas suas características, como suporte mais adequado a este projecto, efectuou-se uma busca do hardware

disponível para o desenvolvimento. Sendo um protocolo inovador, esta busca teve de atentar cuidadosamente nas características dos produtos, nomeadamente o seu firmware de suporte ao protocolo – dado praticamente inúmeros estarem ainda nas suas versões beta (e, como tal, com algumas limitações face ao standard ZigBee definido pela ZigBee Alliance).

A opção por um sistema constituídos por componentes discretos, que incluía o microprocessador PIC 18F330, o transceiver ZigBee Maxstream Xbee e o módulo MAXIM MAX3223 (para conversão de níveis eléctricos RS-232) prendeu-se fundamentalmente com dois motivos: por um lado uma clara redução de custos face aos kits de desenvolvimento em tecnologia ZigBee disponibilizados pela maioria dos fabricantes; por outro, devido a uma maior “controlabilidade” que estaria associada ao conjunto – ou seja, menor dependência de softwares proprietários para o desenvolvimento de possíveis soluções.

Tomada a decisão final acerca deste conjunto foi o contacto com o fabricante / fornecedor que começou por revelar o carácter ainda pouco maduro desta tecnologia: o transceiver XBee, embora excelente compromisso em termos de preço, não estava, à data, disponível para comercialização – algo que não está explícito no site do fabricante -, encontrando-se ainda em desenvolvimento a sua versão ZigBee.

Dadas as restrições temporais para a realização do projecto, foi tomada a opção de implementar uma solução utilizando o kit PICDEM Z, disponível no laboratório de Redes e Serviços de Comunicação de Dados do DEEC. Efectuado novamente um estudo, mais aprofundado, sobre este kit quer em termos de características do mesmo quer em termos de pilha protocolar, foi concluído que era uma alternativa viável. Efectivamente, num projecto anteriormente realizado que se baseou em elementos deste kit, existiram vários problemas (relacionados com a pilha protocolar). No entanto, o facto do fabricante actualmente já disponibilizar uma versão “ZigBee Compliant” (ou seja, não beta) da pilha protocolar, para o kit PICDEM Z, levou a que se afastassem receios.

Efectuaram-se procedimentos de familiarização com o ambiente de desenvolvimento MPLAB e o programador, com possibilidade de in-circuit debug, ICD2. Nos primeiros testes detectou-se que essa mesma versão da pilha protocolar não era compatível com esta versão do kit, embora tal só se revelasse após instalação (e fosse omitido no seu lançamento). Efectivamente não suportava o transceiver Chipcon que equipava as placas do kit que estava a ser utilizado. Contactada a Microchip, aguardou-se o envio da nova stack ZigBee, a última versão compatível com o uso deste transceiver. Foi-se paralelamente desenvolvendo a capacidade de comunicação via porta-série (leitura, escrita e

armazenamento de dados), tendo-se previamente passado a efectuar a alimentação da placa por meio de cabo ligado a fonte de alimentação (para evitar troca de baterias).

Uma vez que foi disponibilizada uma balança Mettler-Toledo com capacidade de comunicação porta-série, foi estudada a forma de configuração e realizada a programação necessária a uma correcta interacção entre o módulo ZigBee e este dispositivo.

No entanto, foi na última fase do projecto e após recepção da referida informação da Microchip, que se constatou que ainda assim continuavam a surgir problemas: nesta versão, embora o módulo coordenador efectuasse o início da rede, não era possível o dispositivo Endpoint juntar-se a ela. Foram tentadas inúmeras soluções para este problema, que em nenhum caso produziram efeito, agravado pela relativa complexidade de todo o código associado a um protocolo deste tipo. Os programas de demonstração de interacção entre as duas placas (Coordinator e Endpoint), disponibilizados pelo fabricante, inclusivamente não sortiram qualquer efeito. Por outro lado, a Microchip prossegue neste momento apenas com desenvolvimento da pilha protocolar para o seu transceiver Microchip MRF24J40, tendo posto de parte o Chipcon CC2420.

A tecnologia ZigBee apresenta-se, sem dúvida, como bastante interessante e com uma vasta gama de aplicações. A baixa latência, a optimização do duty-cycle para um baixo consumo e o tamanho reduzido da pilha protocolar (quando comparado com o Bluetooth ou Wi-Fi), tornam-na a opção adequada para aplicações que necessitam de transmitir um reduzido fluxo de dados mas com necessidade “tempo-real”. Além do mais, suporta um elevado número de nós por rede, o que a faz ideal para aplicações de sensorização, controlo e monitorização de dispositivos.

No entanto, a documentação disponibilizada pelos fabricantes é muitas vezes ainda algo incompleta, as soluções comerciais também ainda não estão numa fase madura. Tal constata-se quer pela questão das inúmeras versões “beta” de (muitos) equipamentos, assim como a própria falta de fiabilidade do software disponibilizado pela Microchip (que é, alias, uma das empresas pioneiras neste tipo de produtos). Assim, embora não se possa considerar o ZigBee como um protocolo recém-lançado, está ainda numa fase de desenvolvimento e implantação comercial (pelo menos por parte da grande maioria dos fabricantes).

Face aos problemas, já explicitados, da pilha protocolar fornecida, à data de realização deste relatório não foi possível concluir a fase de desenvolvimento na qual se pretende ver concretizada a comunicação de dados entre as placas Endpoint e Coordinator. Até à data de apresentação deste projecto, continuarei a dedicar-me ao mesmo, promovendo a sua evolução – nomeadamente tendo expectativa que seja possível o suporte técnico da Microchip disponibilizar atempadamente uma versão protocolar funcional.

Este projecto permitiu-me adquirir um maior à vontade com a manipulação de componentes e dispositivos, nomeadamente microcontroladores e a sua interface, assim como o estudo e ampliação de conhecimentos em termos de protocolo de comunicação wireless. Mais especificamente, a introdução de uma tecnologia que, após maturação, certamente atingirá uma massificação no mercado.

O facto do projecto ter sido iniciado totalmente de raiz obrigou a um inicial “aproximar ao problema”, aliado a uma inevitável necessidade de estudo e investigação, assim como contacto com fornecedores e fabricantes, competências tantas vezes necessárias no mundo do trabalho. Dado o trabalho ter sido realizado de forma totalmente individual e não existindo um outro grupo de trabalho na mesma área, tal forçou a um intenso ultrapassar de dificuldades e indecisões, tentando sempre que a evolução do projecto fosse feita da forma mais coesa e coerente.

6 Desenvolvimentos Futuros

A realização do projecto atravessou vários percalços e, estando ainda a aguardar alguma eventual evolução até à data do término do mesmo, não é totalmente previsível o que ocorrerá.

No entanto, é certo que várias melhorias poderão ser realizadas, quer no âmbito deste projecto em concreto quer numa perspectiva de utilização dos módulos de comunicação para outras aplicações, dentro da mesma área.

Uma possível evolução a ponderar, mas que se reveste de particular interesse, é a aquisição de novas placas-filhas, equipadas com o transceiver Microchip MRF24J40. Tal permitiria o uso da pilha protocolar “ZigBee Compliant” (v.1.0-3.8), a qual estará numa versão mais madura que a disponível para o transceiver Chipcon. Para além disso, estaria disponível o apoio do fabricante, que neste momento já não presta suporte técnico a kits em cuja placa-filha está equipada com transceiver Chipcon. Desta forma seria possível não só observar correctamente a demonstração fornecida acerca das potencialidades do ZigBee como também realizar os testes adequados a uma aplicação de comunicação de dados entre dois nós. A aquisição de novas placas-filhas não seria de custo bastante elevado, cifrando-se o custo de cada uma em torno dos 40 €.

Algo também bastante útil, ainda que eventualmente morosa, será uma análise cuidadosa do código da última versão da pilha protocolar e questionar até que ponto seria possível a sua adaptação ao transceiver Chipcon.

Por outro lado, também será de interesse o desenvolvimento de software capaz de gerir, no computador ao qual se acople o módulo “central” (Coordinator), toda a informação que este reúne, provindo dos diversos dispositivos Endpoint. Ou seja, a utilização de uma base de dados, que permite armazenar a informação “sensorizada” e, eventualmente, a sua utilização no âmbito de processos de gestão e controlo. De referir que, numa versão eventualmente simplista, para efeitos de registo, estão disponíveis no mercado pequenos programas denominados por loggers, os quais se encarregam da escrita num ficheiro de toda a informação recebida via porta-série – a pode, por exemplo, ser posteriormente sujeita a tratamento. Disso é exemplo o programa *Advanced Serial Data Logger* da AGGSoft, que permite inclusivamente a monitorização simultânea de várias comunicações, assim como uma perfeita interacção em ambiente Windows e

portanto uma escrita directa em ficheiros Excel e Access, entre outras funcionalidades; por outro lado, o programa *RS232 Data Logger* da Eltima Software, que, embora numa lógica mais simples, se caracteriza por ser *open-source* e disponibiliza versões possíveis de serem integradas no próprio código de aplicações.

Uma outra particularidade do kit PICDEM-Z que pode ser explorada é o facto das placas incorporarem sensor de temperatura, o que naturalmente poderá constituir-se de interesse em medições da temperatura ambiente, por exemplo. Esta informação poderá obter-se de uma forma nada complexa, encontrando-se já o hardware no próprio módulo.

Dada a evolução deste tipo de tecnologias, importa também realçar que poderá ser de interesse o estudo e concretização da solução baseada em componentes discretos, através da utilização do módulo XBee da Maxstream (ou semelhante).

Conforme já abordado, esta solução apresentava-se bastante atractiva, não só pelo seu custo reduzido mas também pelas dimensões extremamente compactas aliadas a elevada versatilidade (fundamentalmente baseia-se na programação do microprocessador, não dependendo de software proprietário nem necessitando a aplicação desenvolvida de interacção directa com a pilha protocolar).

Para isso, penso que o mais importante será estar atento à efectiva “maturação” desta tecnologia e solidez das propostas comercializadas, especialmente tendo em atenção lançamento de versões integralmente compatíveis com a norma ZigBee (“ZigBee Compliant”).

7 Bibliografia

- [1] ZigBee Alliance
What You Should Know About ZigBee

- [2] Wexler, Joanie
Bluetooth and ZigBee – compare and constrast
Network World, 16 de Março de 2005

- [3] Karayannis, George
Emerging Wireless Standards

- [4] Oehen, Patrice
ZigBee: An Overview of the Upcoming Standard

- [5] Kinney, Patrick
ZigBee Technology: Wireless Control that Simply Works

- [6] Herzog, Chris
An overview of ZigBee – the power of the mesh

- [7] Santos Pinheiro, José Maurício
ZigBee em Home Area Network
Projecto de Redes, 12 de Fevereiro de 2006.

- [8] Santos Pinheiro, José Maurício
As redes com ZigBee
Projeto de Redes, 27 de Julho de 2004.

- [9] Souza Oliveira, Marília Eutímia
ZigBee

- [10] Malafaya, Hugo; Tomás, Luís; Sousa, João Paulo
Sensorização sem fios sobre ZigBee e IEEE 802.15.4

- [11] Microchip
Microchip Stack for ZigBee
- [12] Martins Monteiro, Paulo Jorge
Protótipos Industriais ZigBee™
- [13] Microchip
PIC18F2525/2620/4525/4620 Data Sheet
- [14] Maxstream
XBee™/XBee-PRO™ OEM RF Modules Data Sheet

A. Anexos

A.1 Balança Mettler-Toledo Spider 1

Fabricante: Mettler-Toledo

Modelo: [Prato+Terminal] Spider 1-15/A

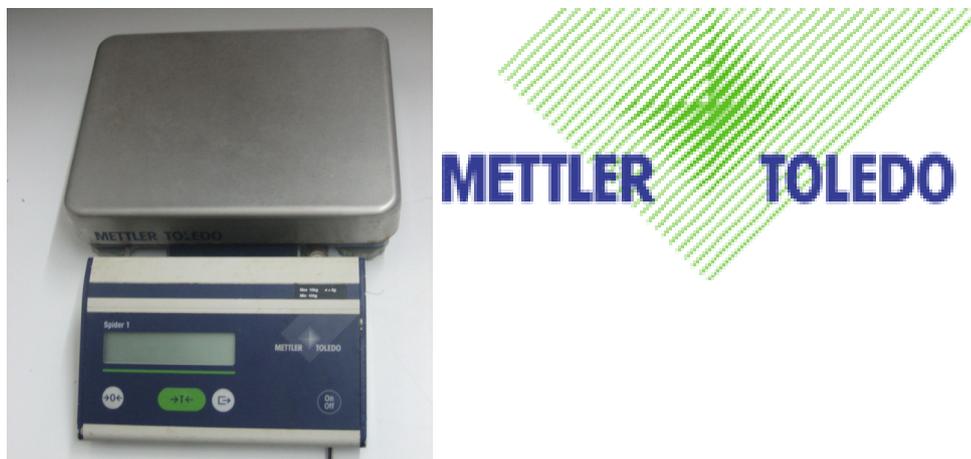


Fig. A.1 Balança Mettler-Toledo Spider 1

Este equipamento, cedido pela empresa RealSabor, é utilizado como instrumento de medida utilizado na linha de produção (na confecção de charcutaria, etc.). Apresenta a particularidade de possuir comunicação via porta-série, sobre o protocolo eléctrico RS-232.

A.1.1 Configuração Comunicação-Série

Em termos de configuração dos parâmetros da ligação de dados via porta-série, são distintas combinações, igualmente aceites quer pelo programa HyperTerminal quer pelo terminal de interface da balança.

Importa então definir cuidadosamente no programa HyperTerminal as características da ligação (as quais serão adoptadas também na balança), sendo que neste trabalho se colocaram conforme a figura **Fig. A.1 Balança Mettler-Toledo Spider 1**:

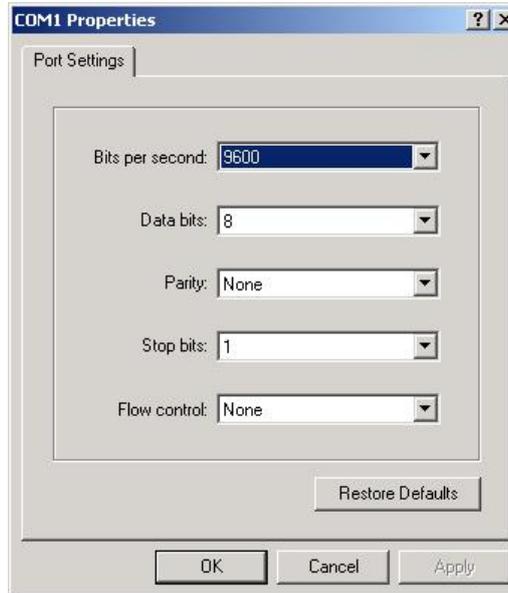


Fig. A.2 Configuração HyperTerminal ligação-série

A.1.2 Parametrização do Equipamento

A configuração deste equipamento não é intuitiva, com a agravante de o seu menu ser pouco perceptível e o manual de instruções não estar disponível. Assim, depois de algum contacto com o equipamento, torna-se necessário deixar clara a forma de o configurar.

Há que distinguir dois botões fundamentais neste processo:

Botão 1

Print 

Funciona como validação de opção.

Botão 2

Tara 

Funciona como botão de rejeição da opção oferecida (efectuando portanto a “rotação de menu”, ou seja, passando sucessivamente à opção seguinte).

Para aceder ao menu:

Pressionar durante alguns segundos o botão 1. Ao surgir a informação "MASTER", premir novamente o botão 1.

Estrutura do Menu:

```
UNITS [UNIDADE DE MEDIDA]
  KG
  T
  G
  LB

A-ZERO
  ON
  OFF

POWER-OFF [DESLIGAR AUTOMÁTICO APÓS PERÍODO DE INACTIVIDADE]
  ON
  OFF

SAVE
  ON
  OFF

VIBRATION
  HIGH
  MÉDIUM
  LOW

PROCESS
  UNIVERSAL
  ABSOLUTE
  DOSING

INTERFACE
  COMMUNICATION [PARÂMETROS DA COMUNICAÇÃO PORTA-SÉRIE]
    PROTOCOL
      XONOFF
      No

    PARITY
      ODD
      EVEN
      7 NO PARITY
      8 NO PARITY

    BAUD RATE
      300
```

```

        600
        1200
        2400
        4800
        9600

NODE [FORMA COMO A COMUNICAÇÃO DE DADOS É EFECTUADA]
    PRINT
    CYCLE
    DIALOG
    2.D DISP
    CONT
    DTB
    DTG
    T-CONT
    RS-KEY

DEFSTR [TIPO DE INFORMAÇÃO A ENVIAR]
    GROSS      [PESO BRUTO = PESO NETO + TARA]
    NET        [PESO NETO]
    TARE       [TARA]

LNFOR [LINE FORM - TIPO DE LINHA A ENVIAR]
    MULTL     [INFORMAÇÃO MÚLTIPLA EM CADA LINHA]
    SINGLE    [INFORMAÇÃO SINGULAR EM CADA LINHA]

CTRL

LIST
    PRINTI

RESET

END

```

Comando LIST:

Esta opção permite visualizar os parâmetros actuais de configuração do equipamento, os quais serão enviados via porta-série (sendo a comunicação efectuada com as características que se encontrem definidas no submenu INTERFACE, nesse instante).

A.1.3 Configuração do Equipamento – Interface / Comunicação

No contexto concreto do projecto que está a ser desenvolvido, pretende-se uma comunicação da informação aquando da estabilização do pesagem. Ou seja, supõe-se que o processo de medição é efectuada de forma sucessiva (a cada um

dos produtos da linha de produção) e, portanto, o pretendido é que a balança envie os dados correspondentes assim que estejam os mesmos estejam disponíveis.

Desta forma, em termos da forma como a comunicação é efectuada, o modo adequado é CYCLE, no qual a balança enviará o peso líquido (uma vez seleccionado NET em DEFSTR) aquando da estabilização da pesagem. Por outro lado, a forma da linha que é enviada deverá estar definida como SINGLE (embora, efectivamente, só esteja programado o envio de um tipo de peso).

Em termos de parâmetros da ligação de dados, para que esteja de acordo com o anteriormente descrito para o programa HyperTerminal, deverá ser seleccionado:

```
PROTOCOL: NO
PARITY: 8 NO PARITY
BAUD RATE: 9600
```

Assim, e recorrendo ao comando List, os parâmetros do terminal de pesagem deverão estar definidos da seguinte forma:

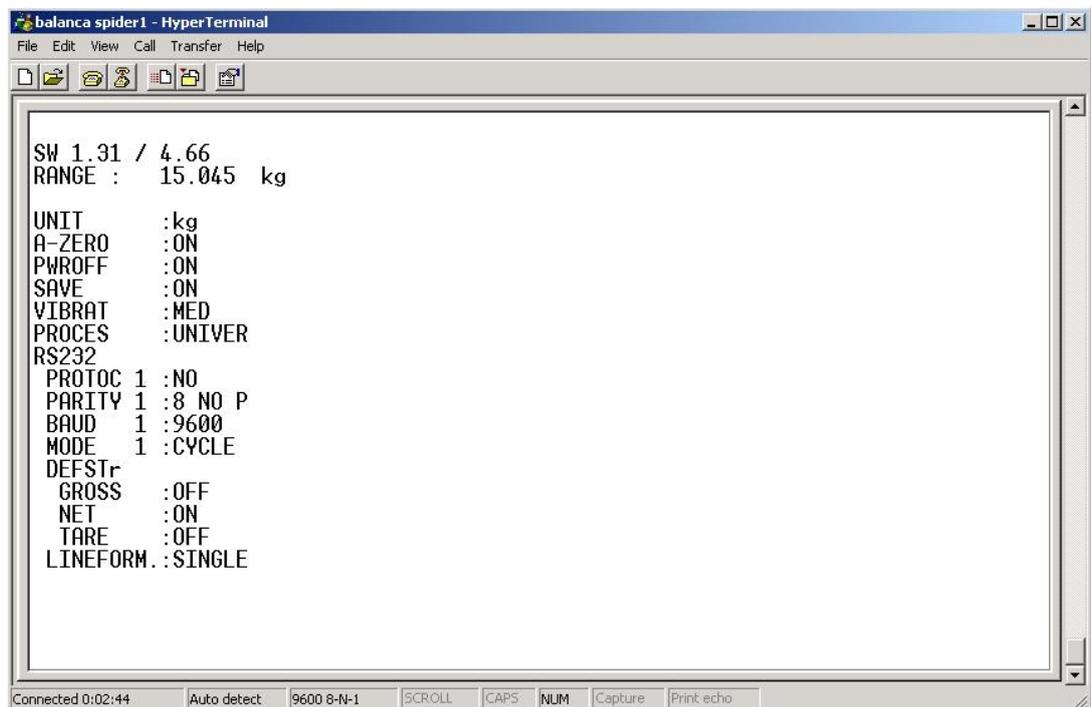


Fig. A.3 Balança Mettler-Toledo: uso do comando List

Ao realizarem-se as sucessivas pesagens, os dados serão enviados via porta-série, conforme se pode ver no exemplo abaixo:

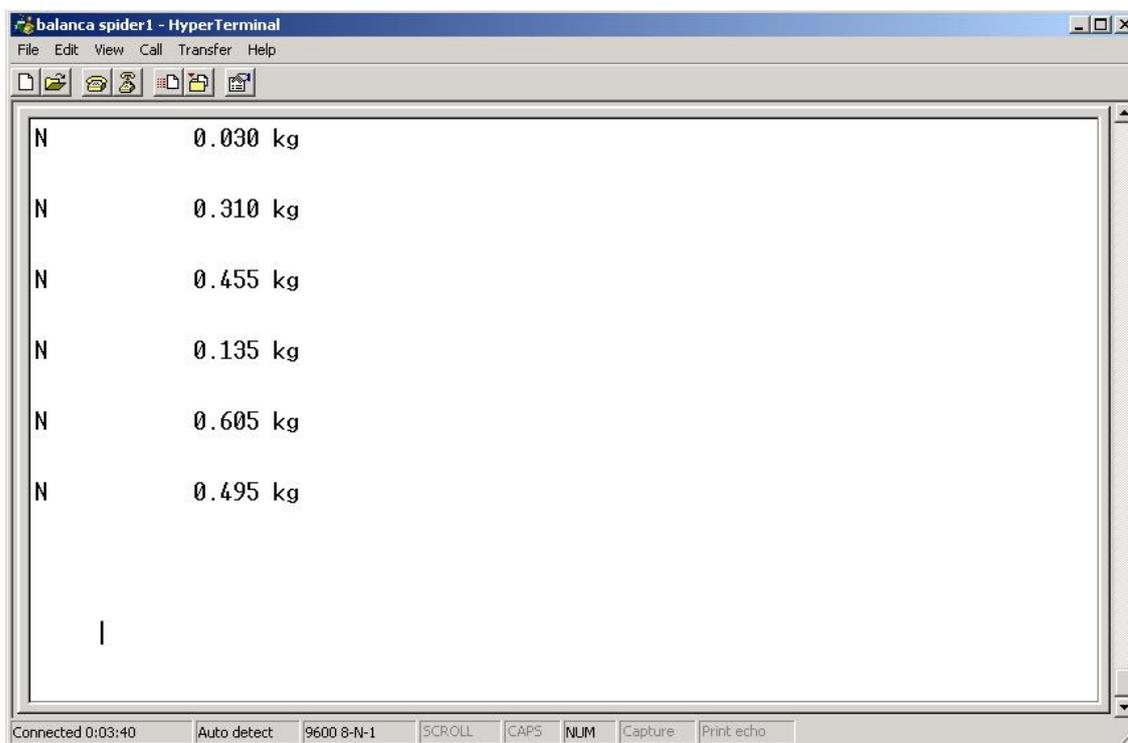


Fig. A.4 Balança Mettler-Toledo: dados pesagem em modo Cycle

A.2 Código realizado

A.2.1 Escrita Porta-Série

```
#include <pl8cxxx.h>
#include <string.h>
#include <usart.h>

#pragma config OSC=XT      //configuração do oscilador em modo externo

void main()
{
    OpenUSART(
        USART_TX_INT_OFF & // interrupcao transm off
        USART_RX_INT_OFF & // interrupcao recepcao off
        USART_ASYNC_MODE & //modo assincrono
        USART_EIGHT_BIT & // 8-bit
        USART_CONT_RX & // recepcao continua
        USART_BRGH_HIGH, // baudrate elevada
        12 // configuracao para 19200bps (oscilador de 4Mhz)
    );

    putsUSART("\n\r Hello! Estou no inicio do programa!\n");

    while(1)
    {
        while( ! PORTAbits.RA0 ); //wait for RA0 high
        WriteUSART( PORTD ); //write value of PORTD
        if(PORTD == 0x80) // check for termination
            break; // value
    }

    putsUSART("\n\r Estou no fim do programa\n");

    CloseUSART();
}
```

A.2.2 Leitura Porta-Série

```
#include <pl8cxxx.h>
#include <string.h>
#include <usart.h>

#pragma config OSC=XT      //configuração do oscilador em modo externo

void main(void)
{
    char inputstr[3];
    int control=0;

    OpenUSART(
        USART_TX_INT_OFF & // interrupcao transm off
        USART_RX_INT_OFF & // interrupcao recepcao on
```

```

    USART_ASYNC_MODE & // modo assincrono
    USART_EIGHT_BIT & // 8-bit
    USART_CONT_RX & // recepcao continua
    USART_BRGH_HIGH, // baudrate elevada
    12 // configuracao para 19200bps (oscilador de 4Mhz)
);

    putsUSART((const far rom char *) "\r\n***PROGRAMA DE TESTE
USART***:\r\n ");
    putsUSART((const far rom char *) "\r\n***LEITURA E ESCRITA DE 2
CARACTERES***:\r\n ");
    putsUSART((const far rom char *) "\r\n***O PROGRAMA TERMINARA AO
DIGITAR A COMBINACAO qq***:\r\n ");

    while (control==0){

        //escrita na usart
        putsUSART((const far rom char *) "\r\nDIGITE 2
CARACTERES:\r\n ");

        while(BusyUSART());
        while (!DataRdyUSART());

        // por os dois caracteres na string input
        getsUSART(inputstr, 2);

        putsUSART((const far rom char *) "\r\nDIGITOU CARACTERES: ");
        putsUSART(inputstr);
        putsUSART((const far rom char *) "\r\n");

        if(inputstr[1]=='q' && inputstr[0]=='q')
            control=1;

    };

    putsUSART((const far rom char *) "\r\nPROGR TERMINADO.DIGITOU 'qq'.
\r\n");
    // fecha a ligacao apenas quando terminar a sua utilizacao
    while(BusyUSART());
    CloseUSART();
    while(1);

}

```

A.2.3 Leitura Porta-Série com interrupção

Programa Genérico

```

#include <p18cxxx.h>
#include <stdio.h>
#include <string.h>
#include <usart.h>

#pragma config OSC=XT //configuração do oscilador em modo externo
#pragma config LVP = OFF
#pragma config WDT = OFF

```

```

#pragma config DEBUG = ON

void rx_handler(void);

#define BUF_SIZE 25

#pragma idata bigdata
char data[5][BUF_SIZE+1] = {
{ "String #0 - digitou 0\n\r" },
{ "String #1 - digitou 1\n\r" },
{ "String #2 - digitou 2\n\r" },
{ "Tecla Invalida (apenas 0-3) \n\r" }
};
#pragma idata

//definicao do ir para rx_handler, aquando da interrupcao
// 0x08 - vector de interrupcoes de alta prioridade

#pragma code rx_interrupt = 0x8
void rx_int (void)
{
    _asm goto rx_handler _endasm
}
#pragma code

// codigo a executar na interrupcao

#pragma interrupt rx_handler
void rx_handler (void)
{
    unsigned char c;
    putsUSART((const far rom char *) "\r\n Entrou na interrupcao!\r\n
");

    if (c >= '0' && c <= '3')
    {
        c -= '0';
        putsUSART (data[c]);
    }
    else
    {
        putsUSART (data[4]);
    }

    /* Limpar interrupt flag - EUSART Receive Interrupt Flag bit */
    PIR1bits.RCIF = 0;
}

void main(void)
{
    OpenUSART(
    USART_TX_INT_OFF & // interrupcao transm off
    USART_RX_INT_ON & // interrupcao recepcao on
    USART_ASYNCH_MODE & //modo assincrono
    USART_EIGHT_BIT & // 8-bit
    USART_CONT_RX & // recepcao continua
    USART_BRGH_HIGH, // baudrate elevada
    12 // configuracao para 19200bps (oscilador de 4Mhz)
    );
}

```

```

    RCONbits.IPEN=1; // activar nıveis de prioridade nas interrupcoes
    IPR1bits.RCIP=1; // activar interrupcao RX como alta prioridade
    INTCNbits.GIEH=1; // activar as interrupcoes de alta prioridade

    putsUSART((const far rom char *) "\r\n entrou no programa!\r\n ");
    if(PIE1bits.RCIE==1)
        putsUSART((const far rom char *) "\r\n esta activada a
Interrupcao de recepcao (RCIE)! \r\n");
    putsUSART((const far rom char *) "\r\n DIGITE UM ALGARISMO DE 0 a 3,
sera emitida a string correspondente a esse digito! \r\n ");

    while(1);
    while(BusyUSART());
    CloseUSART();

}

```

Programa de Interacção com a Balança

```

#include <p18cxxx.h>
#include <stdio.h>
#include <string.h>
#include <usart.h>

#pragma config OSC=XT //configuraco do oscilador em modo externo
#pragma config LVP = OFF
#pragma config WDT = OFF
#pragma config DEBUG = ON

/* PROGRAMA PARA LEITURA DOS DADOS ENVIADOS PELA BALANA - 9600BPS COM
INTERRUPCAO de RX */

void rx_handler(void);

#pragma idata stringdata
unsigned char inputstr[21];
#pragma idata

//definicao do ir para rx_handler, aquando da interrupcao
// 0x08 - vector de interrupcoes de alta prioridade

#pragma code rx_interrupt = 0x8
void rx_int (void)
{
    _asm goto rx_handler _endasm
}
#pragma code

// codigo a executar na interrupcao

#pragma interrupt rx_handler
void rx_handler (void)
{
    // por os 20 caracteres na string input
    while(BusyUSART());
    getsUSART(inputstr, 20);
    inputstr[20]='\0';
}

```

```

    putsUSART((const far rom char *) "\r\nSensor1 ");
    putsUSART(inputstr);
    putsUSART((const far rom char *) "\r\n");
    //while(BusyUSART()

    /* Limpar interrupt flag - EUSART Receive Interrupt Flag bit */
    PIR1bits.RCIF = 0;
}

void main(void)
{
    OpenUSART(
        USART_TX_INT_OFF & // interrupcao transm off
        USART_RX_INT_ON & // interrupcao recepcao on
        USART_ASYNC_MODE & //modo assincrono
        USART_EIGHT_BIT & // 8-bit
        USART_CONT_RX & // recepcao continua
        USART_BRGH_HIGH, // baudrate elevada
        25 // configuracao para 9600bps (oscilador de 4Mhz)
    );

    RCONbits.IPEN=1;
    IPR1bits.RCIP=1;
    INTCNbits.GIEH=1;

    putsUSART((const far rom char *) "\r\n entrou no programa!\r\n ");
    if(PIE1bits.RCIE==1)
        putsUSART((const far rom char *) "\r\n Esta activada a
Interrupcao de recepcao (RCIE)! \r\n");

while(1); // a aguardar interrupcao de RX
while(BusyUSART());
CloseUSART();

}

```

A.2.4 Comunicação entre nós ZigBee

```

/* Segmento de código */
/* Dispositivo Endpoint */

void rx_handler(void);

/* tamanho dos dados a receber - string enviada pela balança Mettler*/
#define TAM_INPUTSTR 20
#pragma idata stringdata
unsigned char inputstr[TAM_INPUTSTR+1];
#pragma idata

//definicao do ir para rx_handler, aquando da interrupcao
// 0x08 - vector de interrupcoes de alta prioridade

#pragma code rx_interrupt = 0x8
void rx_int (void)
{
    _asm goto rx_handler _endasm
}

```

```

#pragma code

// codigo a executar na interrupcao
/* Rotina de atendimento à interrupção de recepção de dados via porta-série,
contendo código de envio de dados via RF (não testado) */

#pragma interrupt rx_handler
void rx_handler (void)
{
    int i=0;
    // por os TAM_INPUTSTR caracteres na string input
    while(BusyUSART());
    getsUSART(inputstr, TAM_INPUTSTR);
    inputstr[TAM_INPUTSTR]='\0';

    if(ZigBeeReady())
    {
        ZigBeeBlockTx();

        TxBuffer[TxData++]=APL_FRAME_TYPE_KVP|1 //KVP, 1 transmissão
        TxBuffer[TxData++]=APLGetTransId(); // ID da transmissão
        TxBuffer[TxData++]=APL_FRAME_COMMAND_SET|(APL_FRAME_DATA_TYPE_UNIT8<<4);
        // Attribute ID LSB (conjunto CLUSTER + ATRIBUTO)
        TxBuffer[TxData++] = OnOffSRC_OnOff & 0xFF;
        // Attribute ID MSB
        TxBuffer[TxData++] = (OnOffSRC_OnOff >> 8) & 0xFF;

        for (i=0; i<=TAM_INPUTSTR; i++)
        {
            TxBuffer[TxData++]=inputstr[i];
        }

        params.APSDE_DATA_request.DstAddrMode=APS_ADDRESS_16_BIT;
        params.APSDE_DATA_request.DstEndpoint = destinationEndpoint;
        params.APSDE_DATA_request.DstAddress.ShortAddr = destinationAddress;
        params.APSDE_DATA_request.ProfileId.Val = MY_PROFILE_ID;
        //configuracoes
        params.APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS;
        params.APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_ENABLE;
        params.APSDE_DATA_request.TxOptions.Val = 0;
        // EP_SENSOR - nome do endpoint do sensor
        params.APSDE_DATA_request.SrcEndpoint = EP_SENSOR;
        params.APSDE_DATA_request.ClusterId = OnOffSRC_CLUSTER;
        // cluster com o mm nome do atributo e que seja correspondente
        currentPrimitive = APSDE_DATA_request;
    }
}

```