

Daniel Lucas Caraça
Ricardo Andrade Dalla Bernadina
William Shindi Shukuzawa

Localizador de Objetos: LOBSTER

São Paulo
2011

Daniel Lucas Caraça

Ricardo Andrade Dalla Bernadina

William Shindi Shukuzawa

Localizador de Objetos: LOBSTER

Monografia apresentada à Escola
Politécnica da Universidade
de São Paulo para obtenção do título
de Engenheiro.

São Paulo
2011

Daniel Lucas Caraça
Ricardo Andrade Dalla Bernadina
William Shindi Shukuzawa

Localizador de Objetos: LOBSTER

Monografia apresentada à Escola
Politécnica da Universidade
de São Paulo para obtenção do título
de Engenheiro.

COMISSÃO EXAMINADORA

Prof. Marcelo Zuffo
Escola Politécnica da USP

Prof. Sérgio TakeoKofuji
Escola Politécnica da USP

Resumo

Será feito o projeto de um dispositivo localizador de objetos que poderá localizar objetos de quaisquer dimensões em ambientes abertos ou fechados, com obstáculos ou não.

A localização do objeto será feita através de um chip, que ficará no objeto, que fará uma comunicação wireless com o dispositivo localizador através do protocolo ZigBee. O localizador, através de uma antena unidirecional e do RSSI fornecido pelo módulo, irá obter a potencia do sinal recebido de várias direções e com estes dados irá mostrar qual a direção de maior probabilidade de estar o objeto. Essa informação é atualizada constantemente, permitindo que usuário siga a direção apresentada para localizar o objeto.

Palavras-chave: *Zigbee*. Dispositivo Localizador.RSSI.

Abstract

An object tracking device that can locate objects of any sizes in open or closed rooms with or without obstacles will be projected.

The location of the object will be done through a chip, which will be on the object, which will make a wireless communication with the tracking device via the ZigBee protocol. The locator, through a unidirectional antenna and RSSI supplied by the module, get the power of the signal received from various directions and with those data will show what direction will most likely be the object. This information is updated constantly allowing the user to follow the given direction to locate the object

Keywords: *Zigbee*, Tracking device, RSSI.

SUMÁRIO

Conteúdo

FICHA CATALOGRÁFICA	3
Folha de aprovação.....	4
Resumo	5
Abstract.....	6
SUMÁRIO	7
1. Introdução	9
1.1 Objetivo	9
1.2 Motivação	9
1.3 Escopo.....	9
1.4 Método	10
1.5 Organização do Trabalho	10
2. Referencial Teórico	11
2.1 Tecnologia para a comunicação	11
2.1.1 Bluetooth:	11
2.1.2 ZigBee:	12
2.1.3 Wi-Fi:	13
2.1.4 Comparação entre as tecnologias	13
2.2 Tecnologia de antenas	15
2.2.1 Antena Omnidirecional	15
2.2.2 Antena Patch	15
2.2.3 Antena Yagi	16
2.2.4 Antena Setorial	16
2.2.5 Antena Espinha de peixe	17
2.2.6 Comparação entre antenas	18
2.3 Funcionamento do ZigBee	18
2.3.1 Histórico e características da tecnologia Zigbee	18
2.3.2 Arquitetura.....	21
2.3.3 Modelos de comunicação.....	23
2.3.4 Dispositivos: comunicação e descoberta	24

2.3.5 Tipos de dispositivos e de topologias.....	25
2.3.6 Transmissão de dados via ZigBee.....	27
2.4 Antena Patch	28
2.5 Conceitos adicionais	30
2.5.1 Motor de passo.....	30
3 Desenvolvimento	31
3.1 Equipamentos utilizados.....	31
3.1.1 Módulo	31
3.1.2 Arduino Duemilanove	34
Sumário(fonte: http://arduino.cc/en/Main/ArduinoBoardDuemilanove)	34
3.1.3 Antena Patch	35
3.1.4 Motor de Passo	35
3.2 Método de busca.....	36
3.2.1 Antena omnidirecional e localização por bips.....	36
3.2.2 Duas antenas direcionais e Lei dos senos	37
3.2.3 Uma antena unidirecional.....	38
3.3 Implementação.....	40
3.3.1 Partes do Sistema	40
3.3.2 Configuração dos módulos XBee	41
3.3.3 Arduino	43
3.3.4 Loop de aquisição de valor de potência.....	44
3.3.5 Análise dos dados.....	45
3.3.6 Informar resultados.....	46
3.3.6 Código.....	47
4. Resultados Obtidos.....	62
4.1 Teste 1: Sem obstáculos	62
4.2 Teste 1: Com obstáculos.....	63
5 Conclusão	63
5.1 Resultados e Contribuições / Inovações Esperadas	64
5.2 Considerações sobre o Curso de Graduação.....	64
6 Referências	66
7 Bibliografia Geral.....	67

1.Introdução

1.1 Objetivo

Este projeto tem como objetivo implementar um dispositivo que tenha a capacidade de nos mostrar a posição relativa ao usuário de um objeto, que será mostrada ao usuário através de um anguloque indica a direção que ele deve seguir para encontrar o objeto desejado. O dispositivo localizará o objeto por ondas eletromagnéticas.

1.2 Motivação

Freqüentemente perde-se objetos importantes nos ambientes nos quais habitualmente convivemos. Sejam carteiras, molhos de chaves ou documentos, sempre há aquela agonia de não saber onde estão e que deveriam ter sido guardados propriamente (escritórios, casas, restaurantes ou qualquer tipo de ambiente fechado são ótimos exemplos). Dessa forma, através da implementação de um localizador, seria possível encontrar qualquer artefato no qual tivesse sido colocado um chip especialmente projetado. Outras aplicações podem ser mencionadas como localização de carros em estacionamentos, malas em aeroportos, produtos em estoques, containers em portos, etc.

1.3 Escopo

O localizador será projetado para encontrar objetos de qualquer tamanhoem ambientes abertos ou fechados com obstáculos ou não em um raio de no máximo 30 metros. A precisão garantida de localização será de 1 metro.

1.4 Método

Para obtermos a localização do objeto serão utilizadas ondas eletromagnéticas a fim de obter alguma informação do objeto que nos indique a sua posição. Várias informações podem nos dar sua localização como potência de chegada, tempo de chegada, ângulo de chegada, diferença de tempo de chegada, etc. Alguns desses métodos serão estudados a seguir para obtermos o melhor para a nossa aplicação.

Para o objeto poder enviar uma informação ao localizador, será instalado um chip em cada objeto que se deseja localizar que será capaz de se comunicar com o localizador.

1.5 Organização do Trabalho

Este primeiro capítulo apresenta os objetivos, a motivação, o escopo e a organização deste ensaio.

O segundo capítulo, Referencial Teórico, traz um histórico sobre a tecnologia *ZigBee* e da *ZigBee Alliance*, apresenta as principais aplicações desta tecnologia e faz uma comparação entre o *ZigBee* e outras tecnologias sem fios, principalmente o *Wi-Fi* e o *Bluetooth*. Neste capítulo também é mostrada algumas antenas e a comparação entre elas, além de uma pequena explicação sobre motor de passos.

O terceiro capítulo traz o desenvolvimento detalhado do projeto. As especificações dos chips *Zigbee* e informações e fotos dos *kits*.

O quarto capítulo refere-se a explicação das próximas etapas a serem seguidas no projeto.

O quinto capítulo refere-se ao cronograma a ser seguido para a implementação prática.

O sexto capítulo mostra o orçamento levantado pelos autores.

O sétimo capítulo versa sobre a análise de riscos relacionados ao projeto.

O oitavo capítulo traz as considerações finais dos autores sobre a tecnologia *ZigBee*.

O nono capítulo corresponde às referências consultadas e utilizadas na preparação desta monografia.

O décimo capítulo traz a lista de material bibliográfico consultado para a realização desta projeto.

2. Referencial Teórico

2.1 Tecnologia para a comunicação

Serão apresentadas a seguir as tecnologias Wi-Fi, Bluetooth e ZigBee, que são protocolos de comunicação wireless utilizadas atualmente. Cada uma delas apresenta características específicas e assim são indicadas para diferentes aplicações.

2.1.1 Bluetooth:

Bluetooth[1] é um protocolo para comunicação wireless criado para substituir fios em comunicação entre dispositivos. Ele é projetado para ter uma taxa de transferência alta o suficiente para permitir transferência de arquivos, áudio e vídeo e também para ter um consumo de energia baixo para poder ser adicionado aos mais diversos dispositivos sem alterar significativamente a potência do mesmo. Hoje em dia o Bluetooth já pode ser encontrado em celulares, computadores, mouses, teclados, fones de ouvido e muitos outros.

Para interligar os dispositivos, o Bluetooth cria uma pequena rede ad-hoc[2] entre os dispositivos, chamada de piconet que é capaz de impedir que outros dispositivos, mesmo que sejam também Bluetooth, escutem a transmissão de dados. Nessa rede é possível conectar até 8 dispositivos ao mesmo tempo, sendo que um deles deve ser o mestre, que controla o fluxo e ordem da transmissão, e os outros 7 os escravos, mas isso não impede de criarmos uma rede com mais dispositivos, pois cada dispositivo pode participar de mais de uma piconet ao mesmo tempo, criando uma rede muito maior.

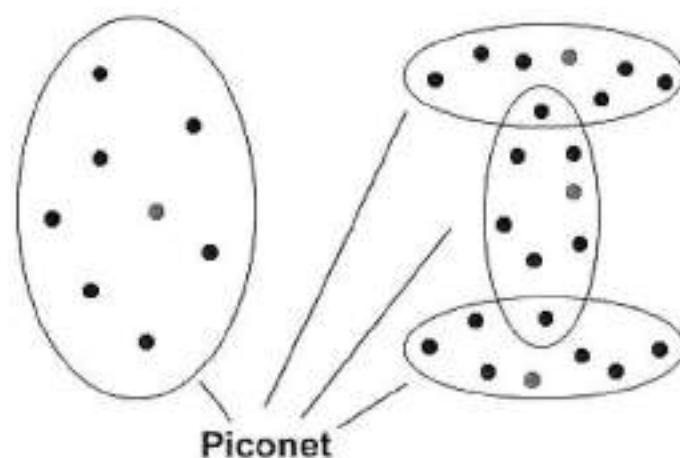


Figura 1 -Varia *piconets* em um ambiente

A frequência utilizada na transmissão dos dados está na zona ISM[3], que é uma zona de frequências abertas. Isso permite que o Bluetooth possa ser utilizado em dispositivos em qualquer parte do mundo, pois não estará infringindo nenhuma lei.

O alcance do sinal do *Bluetooth* pode variar com a potência utilizada no envio do sinal. Três configurações são possíveis ser escolhidas possibilitando atender a diversas aplicações:

Classe1: alcance de 100 metros

Classe 2: alcance de 10 metros

Classe 3: alcance de 1 metro

2.1.2ZigBee:

O ZigBee[4] é um padrão que foi desenhado para atender a necessidade de baixo custo e baixo consumo de energia de sensores wireless e de aplicações de controle. Dispositivos como sensores muitas vezes ficam em lugares isolados e por muito tempo, ou seja, o ideal é que sejam dispositivos que tenham uma imensa independência energética, não necessitando trocar as baterias por um longo período. Além disso, assim como os dispositivos de controle, eles não precisam

trocar muitos dados e necessitam uma baixa taxa transmissão de dados, permitindo maior economia da bateria.

O *Zigbee* tem como base o padrão IEEE 802.15.4[5] que foi criado com a idéia de suprir a necessidade de dispositivos controladores e sensores, que não necessitam de grande taxa de transmissão de dados, mas necessitam baixo consumo de energia, alto alcance e rede que suporte vários dispositivos. Além disso *ZigBee* adiciona ao padrão do IEEE uma camada de rede e uma camada de segurança para viabilizar a utilização comercial do mesmo.

Outra funcionalidade do *ZigBee* é que ele tem um modo de operação onde o dispositivo entra em modo de sleep. Como os dispositivos podem ficar horas sem ser utilizados, entrar no modo sleep permite adicional economia de energia. Os dispositivos *ZigBee* também possuem um tempo de wakeup muito baixo, permitindo obtermos dutycycles bem baixos.

2.1.3 Wi-Fi:

O Wi-Fi[6] tem como objetivo substituir os tantos fios das redes LAN[7] por conexão wireless. Assim o Wi-Fi é projetado para suportar a mesma taxa de transmissão de dados que os fios suportam e não apresenta nenhum tipo de mobilidade e preocupação com economia de energia, por funcionar sempre ligado a rede de energia.

Este protocolo utiliza as frequências de 2.4 e 5 GHz que se encontra na zona do ISM, permitindo assim ser utilizado por todos. Com sua alta transmissão de dados (chega a 450 Mbps), pode conectar dispositivos de vídeo, áudio, de tempo real, computadores, impressoras, dispositivos que acessem a internet e muito mais.

2.1.4 Comparação entre as tecnologias.

As três tecnologia apresentadas, apesar de todas serem wireless, possuem aplicação bem definidas e diferentes. O Wi-fi foi criado para substituir as redes LAN por redes WLAN[8], dando mobilidade aos equipamentos e limpeza no ambiente. Mas por tentar ter a mesma taxa de transmissão de um cabo LAN e não funcionar

ligado à rede de energia, ele não se preocupa com independência energética que é fundamental para dispositivos moveis, assim como o da nossa aplicação.

O Bluetooth já é feito para dispositivos moveis com capacidade de multimídia. Ele pode ser usado para transmitir desde pequenos textos até grandes arquivos. Por ter essa capacidade, o Bluetooth, apesar de se preocupar com a economia de energia, não consegue atingir um consumo de energia tão baixo. Por isso, comparado com o ZigBee, ele é menos indicado para a nossa aplicação.

O *ZigBee*, que foi projetado para monitoramento e controle, apresenta a menor taxa de transmissão de dados e o menor consumo das três tecnologias. Como em nossa aplicação não é necessário transmitir arquivos, vídeo ou som entre os dispositivos e sim apenas informações básicas e pequenas, o ZigBee é o protocolo que se encaixa melhor com a necessidade da nossa aplicação. Assim o ZigBee será o protocolo utilizado em nosso projeto.

Uma comparação entre algumas características importante dos protocolos analisados é mostrada na tabela 1.

Característica	Wi-Fi 802.11b/g	Bluetooth	Zigbee
Frequência	2,4 GHz	2,4GHz	868 MHz, 915 MHz e 2.4 GHz
Quantidade de Dispositivos	32	7	65000
Taxa de Transferência Rádio	11/54 Mbps	1/3 Mbps	250 kbps
Alcance	~100 m	~100 m	~240 - 3200 m (100mW)
Latência	~3 s	~10 s	~30 ms
Segurança	Authentication Service Set ID (SSID)(WEP/ WPA)	64-bit, 128-bit	128-bit AES (Advanced Encryption Standard)
Topologia de Rede	Point-to-Multipoint	Ad hoc piconets	Point-to-Point, Point-to-Multipoint, MESH

Tabela 1 - Comparação entre Wi-fi, Bluetooth e Zigbee (Fonte: Revista Saber Eletrônica. Edição de outubro/2010)

2.2 Tecnologia de antenas

2.2.1 Antena Omnidirecional

A antena que mais se aproxima da antena isotrópica[9] (ideal, impossível de se implementar, emite a mesma potência para todos os lados), pois emite a mesma potência para os 360°, mas apenas no plano horizontal. No plano vertical ela possui um ângulo de abertura como as outras antenas.

Devido a essas qualidades ela é a antena mais fácil de ser instalada, pois não precisamos escolher para qual direção ela vai olhar. Ela trabalha excepcionalmente bem em aplicações ponto multiponto e em broadcast, como por exemplo, em uma estação radio base.



Figura 2 – Antena Omnidirecional. (Fonte: <http://www.vivasemfio.com/blog/antenas-omni-semi-altamente-direcional/>)

2.2.2 Antena Patch

São antenas direcionais que são confeccionadas basicamente por uma placa condutora de circuito impresso montada em cima de uma placa quadrada, separados por um material isolante, sendo muitas vezes o ar.

As vantagens das antenas patch são o baixo custo de confecção, pequeno tamanho, possibilitando ser adicionado facilmente a outros dispositivos, leve e pode operar em duas ou três frequências diferentes. Suas desvantagens são pequena banda, pouca eficiência e baixo ganho.

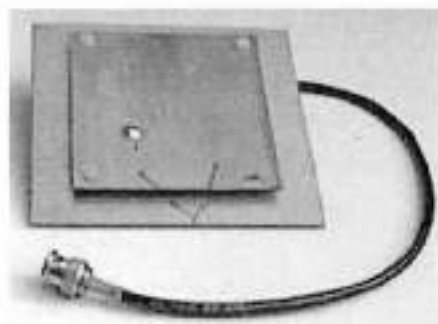


Figura 3 – Antena Patch. [Fonte: <http://home.iae.nl/users/plundahl/antenne/patchant.htm>]

2.2.3 Antena Yagi

É uma antena de alta potência, que pode ser utilizada tanto para transmitir sinais por grandes distâncias, quanto captar sinais fracos, que antenas menores não seriam capazes de captar. Nas redes wireless, são as que tem maior alcance, porém, só conseguem cobrir uma pequena área (para onde são apontadas). Sendo, assim, mais úteis para cobrir uma área específica, longe do ponto de acesso, ou para interligar duas redes distantes. Está exemplificada na figura 4.

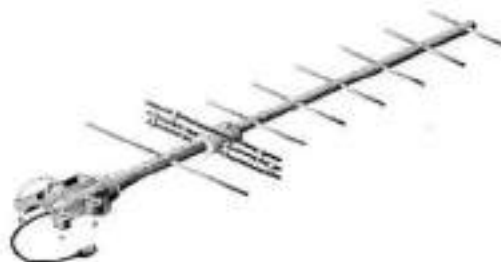


Figura 4: Antena Yagi.

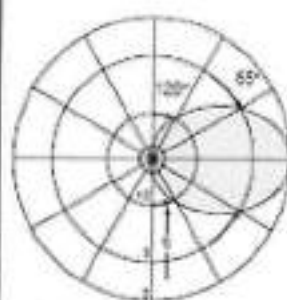
(Fonte: http://montevideolibre.org/manuales/libros:wndw:capitulo_4:antenas_y_diagramas)

2.2.4 Antena Setorial

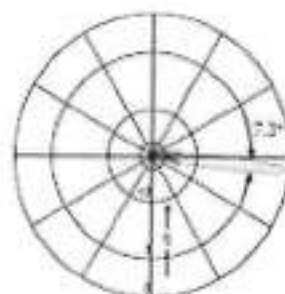
A palavra “setor” é utilizada no sentido geométrico, ou seja, designa uma porção da circunferência (30°, 60°, 90° e 120° são casos típicos) para o seu lóbulo principal. É bastante usada como estação base para telefones celulares ou em aplicações de redes Wi-fi e tem alcance máximo de 4 a 5 km. A figura 5 mostra um exemplo e a figura 6 os padrões horizontal e vertical



Figura 5: Antena Setorial



Horizontal Pattern



Vertical Pattern

Figura 6: Padrões horizontal e vertical de radiação.

(Fonte: <http://www.br.all.biz/g37372/>)

(Fonte: http://www.enotes.com/topic/Sector_antenna)

2.2.5 Antena Espinha de peixe

Antena direcional consistente de uma série de elementos coplanares arranjados em pares. Esses pares estão posicionados transversalmente ao longo de ambos os lados de uma linha de transmissão. A figura 7 ilustra esse modelo.



Figura 7. Antena espinha de peixe. (Fonte:

http://liderlins.com.br/produtos_prod_show.php?id=1243)

Devemos ter em conta que a frequência captada ou emitida pela antena depende tanto do comprimento das "espinhas" quanto da distância de separação. O gráfico a seguir mostra essa relação (com espaçamento de 1,0 cm)

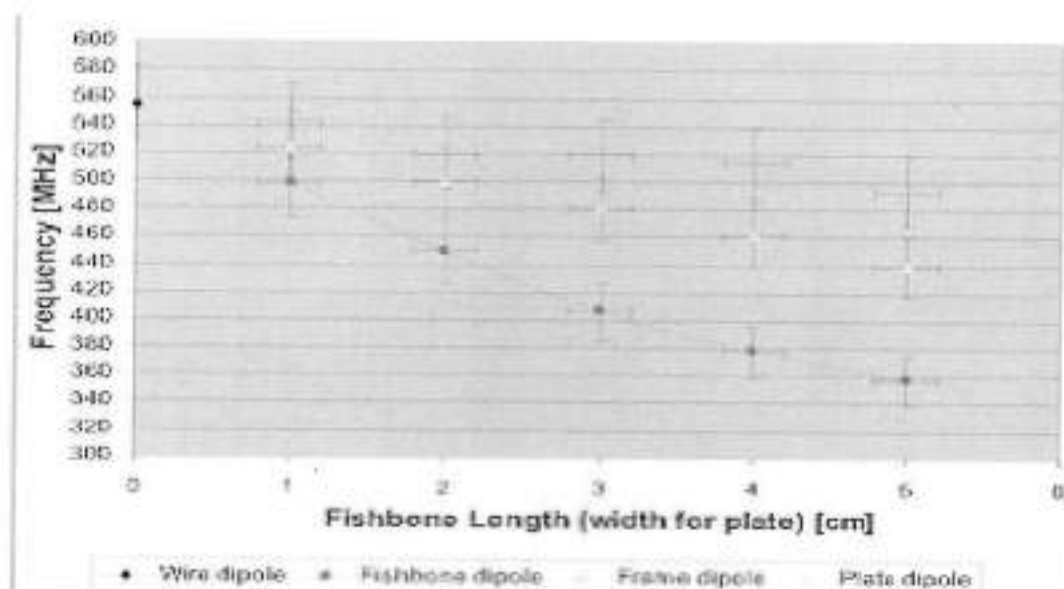


Gráfico 1: Medidas de Frequência de ressonância X Comprimento da "espinha".

2.6 Comparação entre antenas

Em nosso projeto será necessário a utilização dos dois tipos de antenas, uma direcional e uma omnidirecional. Para ambas as antenas, precisamos de alcance de no máximo 20 metros. Assim as antenas Yagi, Setorial e Espinha de peixe não são indicadas, pois elas possuem um alcance muito maior do que o desejado. A antena patch possui um alcance bem menor que as outras, mas o suficiente para o nosso projeto. Além disso ela possui custo bem menor, facilidade de confecção e tamanho pequeno, que é muito importante para o nosso projeto. Por esses motivos serão utilizados a antena omnidirecional e a antena patch em nosso projeto.

2.3 Funcionamento do ZigBee

2.3.1 Histórico e características da tecnologia Zigbee

A tecnologia *ZigBee* foi inicialmente concebida em 1998 por um grupo de desenvolvedores que perceberam que as tecnologias *Wi-Fi* e *Bluetooth* seriam inadequadas para diversas aplicações. O nome *ZigBee* foi criado a partir da analogia entre o funcionamento de uma Rede em Malha, e o modo como as abelhas trabalham e se locomovem. As abelhas que vivem em colméia voam em 'Zig...Zag', e dessa forma, durante um voo a trabalho em busca de néctar, trocam informações

com outros membros da colméia sobre, distância, direção e localização de onde encontrar alimentos. Uma Malha ZigBee dispõe de vários caminhos possíveis entre cada nó da Rede para a passagem da informação, assim, é possível eliminar falhas se um nó estiver inoperante, simplesmente mudando o percurso da informação. [10]

O padrão IEEE 802.15.4 que regulariza a utilização do ZigBee foi completado em maio de 2003. Poucos meses depois, a divisão de Semicondutores da *Philips* parou com seu grande investimento na tecnologia, mas a *Philips*, através de sua divisão de Iluminação manteve sua participação no desenvolvimento do ZigBee, fazendo parte inclusive do conselho da *ZigBee Alliance*[11]. São mais de 300 dos principais fabricantes de semicondutores, empresas de tecnologia, OEMs e companhias de serviço que compõem a associação *ZigBee Alliance*. Ele foi projetado para proporcionar uma solução *wireless* fácil de usar e caracteriza-se por ter arquiteturas seguras e confiáveis.

O ZigBee permite a implantação generalizada de redes sem fio com baixo custo e soluções de baixa potência. Ele tem a capacidade de operar durante anos com baterias comuns por uma série de aplicações de monitoramento e controle. Controles de iluminação, construção de sistemas de automação, monitoramento de tanques, controle de dispositivos médicos e aplicações de frota são apenas alguns dos muitos espaços onde a tecnologia ZigBee está fazendo avanços significativos .

As figuras 8 e 9 apresentam alguns exemplos de utilização em automação residencial e monitoramento médico:

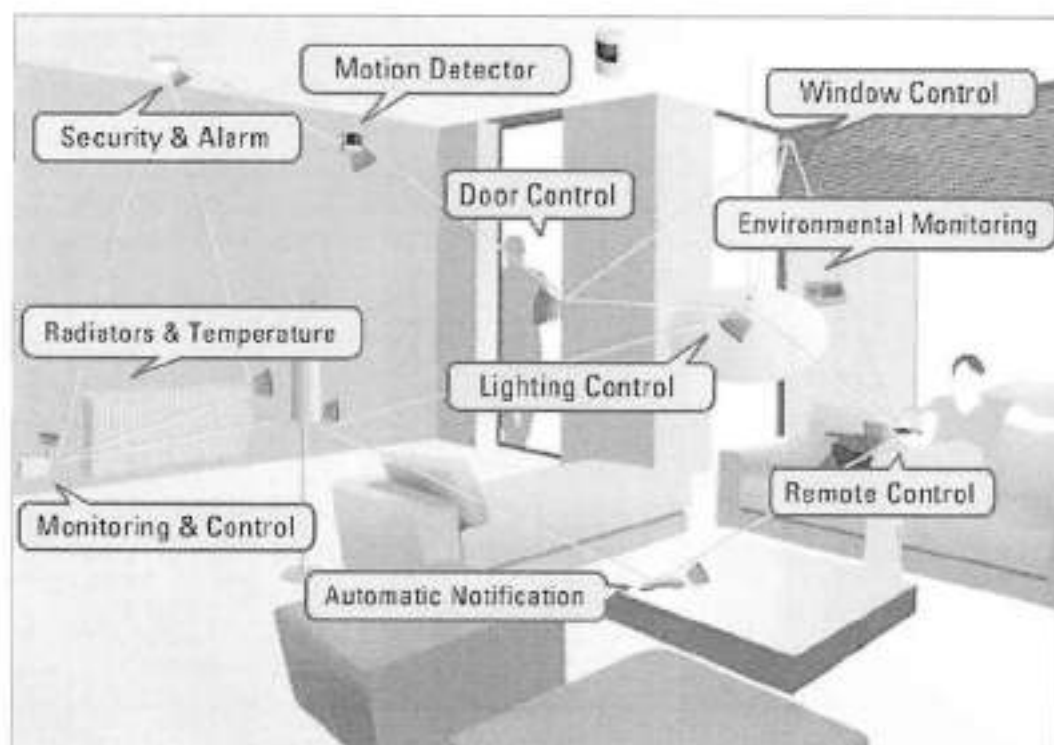


Figura 8- Exemplos de Aplicações para Automação Residencial (fonte: <http://www.tugatronica.com/zigbee-um-padroao-wireless-de-futuro/>)



Figura 9 – Exemplos de Aplicações para Monitoramento Médico (fonte: <http://www.vivasemfio.com/blog/tag/zigbee/>)

Um componente fundamental do protocolo *ZigBee* é a capacidade de suporte a redes *mesh*. Em uma rede *mesh*, os nós estão interligados com outros nós de forma que múltiplos caminhos estão disponíveis para conectar cada nó. As ligações entre nós são atualizadas e otimizados através de uma sofisticada e já embutida tabela de roteamento.

As redes *mesh* são descentralizadas por natureza, cada nó é capaz de realizar a auto-descoberta na rede. Além disso, à medida que os nós saem da rede, a topologia *mesh* permite que eles reconfigurem caminhos de roteamento baseados na nova estrutura de rede [18]. As características da topologia de rede e roteamento *ad-hoc* proporcionam uma maior estabilidade na mudança das condições ou falhas em nós individuais.

2.3.2 Arquitetura

O *ZigBee* segue as especificações muito parecidas com o do modelo IEEE 802.15.4, já que as camadas físicas e de enlace possuem as mesmas funções do que as já definidas por tal modelo.

A arquitetura *ZigBee* é composta pelas camadas física, de enlace, de rede e de aplicação como apresentadas na figura 4 e descritas nas sub-seções a seguir.

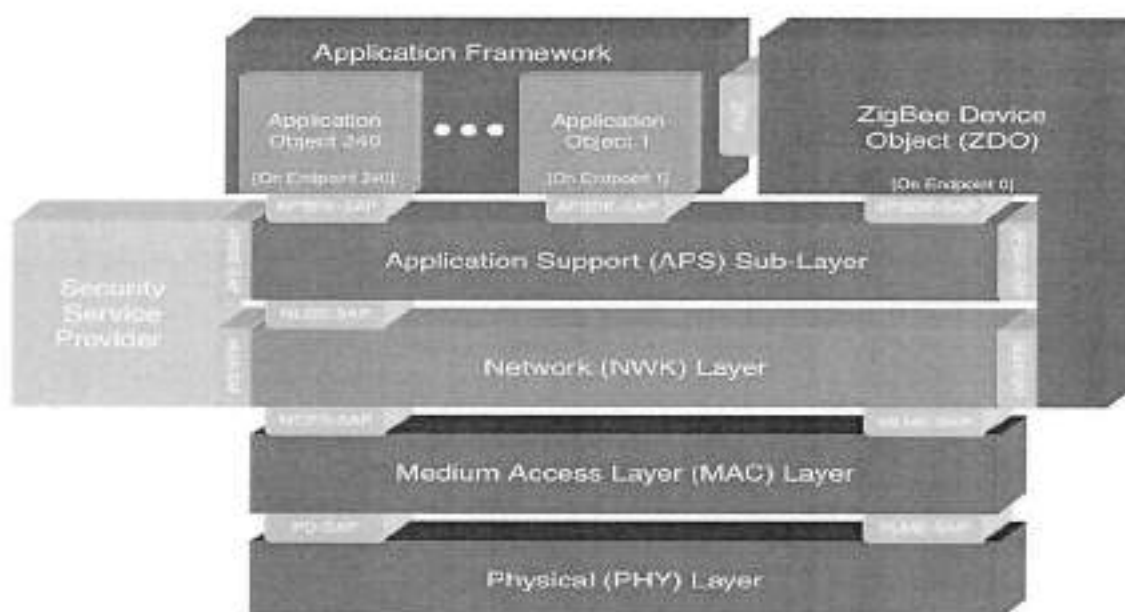


Figura 10 - Arquitetura ZigBee (fonte: <http://betonextreme.com/venray-printable-devry-university-application-in-indiana/>)

2.3.2.1 Camada Física (PHY)

A camada física consiste apenas no transporte dos pacotes de dados pelo ar e reconstituí-los no dispositivo receptor.

2.3.2.2 Camada de Enlace (MAC)

A camada de enlace (ou camada do endereço MAC: controle de acesso ao meio) cria o conceito de uma rede, incluindo um identificador PAN (*portable area network*) e possibilidade de reconhecimento de redes através de sinalização de respostas e requerimentos em *beacon*.

Além disto, esta camada possibilita um reconhecimento por conexão (mas não de multi-conexão ou de malha) e alguns comandos para formação de ou entrada em redes

2.3.2.3 Camada de Rede (NWK)

As principais funções da camada de rede são de possibilitar o uso correto da camada de enlace e fornecer uma interface adequada para a camada de aplicação.

A entidade de dados cria e controla os pacotes de dados na camada de rede a partir dos dados provenientes da camada de aplicação e utiliza a topologia da rede para executar o roteamento destes dados.

Além disto existe o controle desta camada, utilizado para controlar a configuração de novos dispositivos e criar novas redes. Este controle pode determinar se seus dispositivos vizinhos fazem parte de sua rede, além de descobrir novos vizinhos, roteadores e receptores, sendo que estes possibilitam uma comunicação direta e sincronização através da camada de enlace.

O protocolo de roteamento utilizado por esta camada é o AODV [*Add-hoc On-demand Distance Vector routing*]. Para se encontrar o dispositivo de destino de suas informações, um dispositivo emite um sinal para seus vizinhos, tentando estabelecer uma rota para este dispositivo. Isto é repetido pelos vizinhos até que o dispositivo

desejado é encontrado. Neste momento, este dispositivo responde através desta rota mais rápida de volta para o dispositivo que iniciou o processo, atualizando assim a tabela de roteamento deste dispositivo que utilizará esta rota para transmitir os dados desejados.

2.3.2.4 Camada de Aplicação

A mais alta camada definida pela especificação do *ZigBee* é a camada de aplicação e representa a interface do sistema *ZigBee* com seus usuários finais. Esta camada compreende a maioria dos componentes da especificação do *ZigBee* como o objeto do dispositivo *ZigBee* propriamente dito (*ZigBeeDeviceObject* – ZDO), seus procedimentos de gerenciamento, e os objetos de aplicação definidos pelo fabricante.

O ZDO é responsável pela definição do papel de cada dispositivo como um coordenador, um roteador intermediário ou um dispositivo final. Além disto é responsável pela descoberta de novos dispositivos na rede e pela identificação de seus papéis na rede. Depois da definição de seu papel e da identificação dos papéis dos outros, o ZDO estabelece conexões seguras com dispositivos externos e responde a requerimentos de conexão.

A subcamada de suporte à aplicação (APS) é o outro componente principal padrão da camada de aplicação, oferecendo uma interface bem definida e serviços de controle. Esta subcamada serve de ponte entre a camada de rede e os outros componentes da camada de aplicação, mantendo tabelas de conexão atualizadas para possibilitar uma rápida procura pelos dispositivos ideais, considerando os serviços necessários e a disponibilidade destes serviços entre os dispositivos da rede.

2.3.3 Modelos de comunicação

Objetos em comunicação cooperando para realizar tarefas específicas são o que constitui uma aplicação. O *ZigBee* tem como foco a distribuição do trabalho

entre diversos dispositivos, cada qual com seu nó individual *ZigBee* de modo a formar uma rede.

Estes objetos que formam a rede se comunicam através das conexões possibilitadas pela APS, sendo supervisionadas pelas interfaces do ZDO. Os serviços de dados da camada de aplicação seguem uma estrutura típica de requerimentos e confirmações. Para cada dispositivo podem existir outros 240 objetos de aplicação (seus vizinhos) que recebem a numeração de 1 a 240. O número 0 é reservado para o ZDO enquanto o 255 é reservado para broadcast na rede. Já a faixa de 241 a 254 não está em uso na especificação presente do *ZigBee*, mas pode vir a ser utilizada em especificações futuras.

O endereçamento é outra responsabilidade da camada de aplicação. Um nó da rede consiste principalmente de um transceptor de rádio de acordo com as especificações da norma IEEE 802.15.4 e também pela descrição de um ou mais dispositivos, ou seja, conjunto de atributos que podem ser consultados, definidos ou ainda monitorados por eventos. Este transceptor é a base para o endereçamento, e os dispositivos em um nó são identificados por um identificador de ponto final entre 1 e 240.

2.3.4 Dispositivos: comunicação e descoberta

Para que as aplicações possam se comunicar, seus dispositivos precisam utilizar um protocolo comum. Estas convenções são agrupadas em perfis. Além disto, a conexão é decidida através da identificação de entrada e saída de conjuntos, que são únicos dentro do contexto de um certo perfil, sendo associados ao fluxo de entrada em ou saída de um dispositivo. Tabelas de conexão contêm os pares de origens e destinos.

Dependendo da informação disponível, a descoberta de dispositivos pode ser dada por diferentes métodos.

Quando o endereço de rede é conhecido, o endereço IEEE pode ser requerido através de comunicação *unicast*. Se este endereço não é conhecido é utilizado uma busca em broadcast onde o endereço IEEE será parte da resposta. Os dispositivos finais vão simplesmente responder com o endereço requerido, enquanto

coordenadores e roteadores vão enviar o endereço de todos os seus dispositivos associados.

Este protocolo estendido de descobrimento permite que dispositivos externos conheçam sobre os dispositivos de uma e os serviços que oferecem, além de quais os pontos finais que responderão a uma busca iniciada pelo dispositivo descobridor.

A utilização de identificadores para grupos reforça a conexão de entidades complementares através das tabelas de conexão, mantidas pelos coordenadores. Estas tabelas devem estar sempre disponíveis em uma rede, de modo que os coordenadores muitas vezes apresentam fontes de energia permanentes. Já os backups gerenciados pelas camadas superiores podem ser necessários no caso de algumas aplicações.

A comunicação pode ocorrer logo após a associação. O modo de endereçamento direto utiliza tanto o endereço do rádio quanto o identificador de ponto final. Já o modo de endereçamento indireto utiliza todos os campos relevantes (endereço do rádio e identificador de ponto final como no endereçamento direto, e ainda o identificador de grupo e os atributos) e requer que estes dados sejam enviados a um coordenador, que mantém associações e traduz os requerimentos de comunicação. Este segundo modo é muito útil para simplificar alguns dispositivos, minimizando sua memória. Outros modos incluem o *broadcast* para todos os pontos finais de um dispositivo, ou ainda, o endereçamento de grupo, para comunicação com pontos finais pertencentes a um grupo de dispositivos.

2.3.5 Tipos de dispositivos e de topologias

Existem três classes de dispositivos lógicos:

2.3.5.1 Coordenador ZigBee (*ZigBeeCoordinator*, ZC)

É o tipo de dispositivo mais completo. Pode atuar como coordenador de uma rede em árvore assim como também servir de ligação a outras redes. Existe exatamente um coordenador em cada rede, em que este é o nó que a começa. Pode armazenar informação sobre a rede e atuar como sendo o seu "centro de confiança" na distribuição de chaves cifradas.

2.3.5.2 Roteador ZigBee (ZigBee Router, ZR)

Para além de oferecer um nível de aplicação para a execução do código do utilizador, pode atuar como roteador interligando dispositivos separados na topologia da rede.

2.3.5.3 Dispositivo final (ZigBee End Device, ZED)

Possui a funcionalidade necessária para comunicar com o nó pai (coordenador ou um roteador), mas não pode transmitir informação destinada a outros dispositivos. Desta forma, este tipo de nó pode estar "adormecido" na maioria do tempo, aumentando assim o tempo de vida das baterias. Um ZED tem requerimentos mínimos de memória e é, portanto, significativamente barato.

2.3.5.4 Full Function Device (FFD)

Pode funcionar em toda a topologia do padrão, desempenhando a função de coordenador de rede e consequentemente ter acesso a todos os outros dispositivos. Tratam-se de dispositivos de construção mais complexo.

2.2.5.5 Reduced Function Device (RFD)

É limitado a uma configuração com topologia em estrela, não podendo atuar como um coordenador da rede. Pode se comunicar apenas com um coordenador de rede. São dispositivos de construção mais simples.

Existem então três classes de dispositivos lógicos: coordenador, roteador e dispositivo final. Sendo que o coordenador e o roteador são implementáveis com base nos dispositivos físicos da classe FFD e o dispositivo final em dispositivos físicos de classe FFD ou RFD.

Em topologias com configuração em estrela, uma rede *ZigBee* requer pelo menos um dispositivo FFD atuando como coordenador da rede e os restantes dispositivos podem ser do tipo RFD para reduzir o custo do sistema. Para outras topologias, todos os dispositivos deve ser FFD.

2.3.5.6 Topologias

Estes dispositivos podem ser distribuídos em três tipos de topologias: estrela, árvore e malha. Estas topologias estão ilustradas na figura 11.

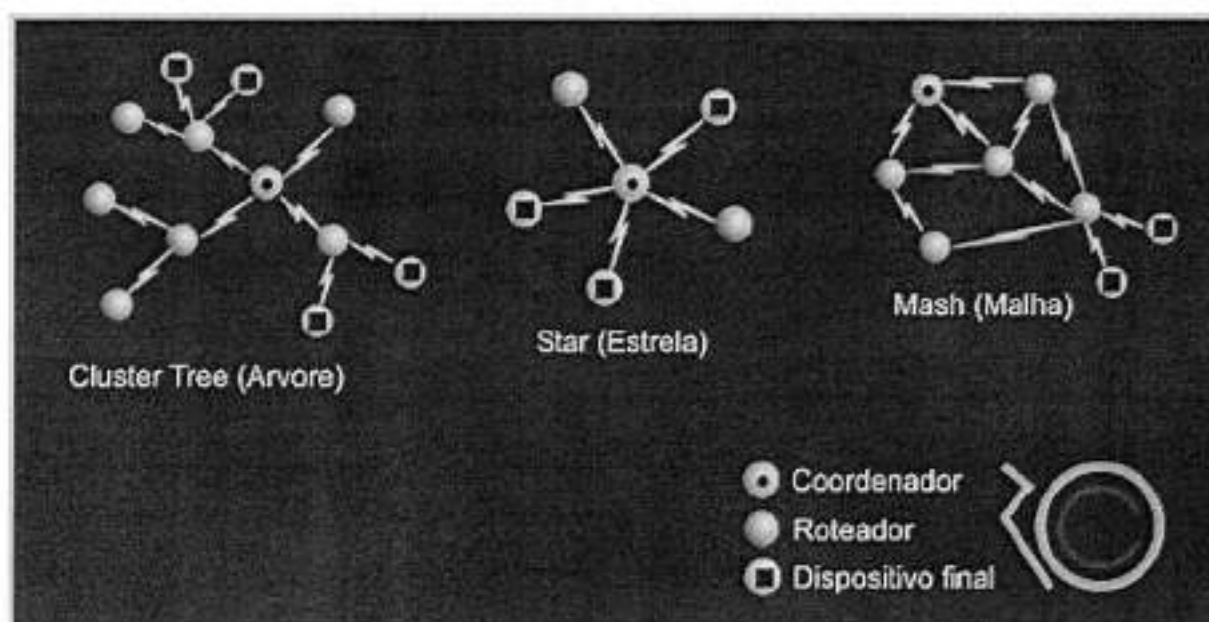


Figura 11 – Topologias: Árvore, Estrela e Malha (Fonte: <http://www.rogercom.com/ZigBee/ZigBee.htm>)

2.3.6 Transmissão de dados via *ZigBee*

Existem dois modos de operação do *ZigBee*: o *Beacon-enabled* e o *Non-Beacon-enabled*.

O modo *Non-Beacon-enabled* possui uma rede heterogênea, assim, enquanto alguns nós ficam ativos em tempo integral, outros podem ficar em modo de espera até que ocorra uma interrupção. Uma desvantagem deste modo é o aumento do consumo de energia dos nós que estão ativos todo o tempo, diminuindo a vida útil de sua bateria.

Diferente do modo anterior, o *Beacon-enabled* possui um maior poder no quesito economia de energia, pois nesse modo, o *ZigBee* não precisa ficar ativo durante longos intervalos de tempo, exceto quando este está transmitindo algum dado.

No *Beacon-enabled* o *Zigbee* transmite de tempos em tempos para confirmar sua presença na rede, como a base de tempos é variável, é possível diminuir o *duty-cycle* para obter uma maior economia de energia, contudo, a diminuição excessiva pode levar a uma falta de sincronia da temporização causando uma grande perda [17].

O modo básico de acesso no *ZigBee* é o CSMA/CA (*Carrier sense, multipleaccess/ collisionavoidance*), pois ele sempre verifica se há alguém "falando" antes de começar qualquer transmissão.

2.4 Antena Patch

Antenas são componentes muito importantes em sistemas de comunicação. Uma antena é um objeto capaz de transformar um sinal elétrico em um sinal eletromagnético e transformar um sinal eletromagnético em um sinal elétrico. Elas são capazes de receber e transmitir sinais em uma certa direção dependendo do seu tipo e também são dispositivos ressonantes, ou seja, funcionam em uma faixa de frequência.

Em termos simples, a polarização de uma antena é a direção do campo elétrico emitido por ela. As ondas emitidas por uma antena são polarizadas de acordo com o modo que a antena é construída. Caso o dipolo da antena seja vertical, a polarização será vertical, caso seja horizontal, a polarização será horizontal e caso tenhamos os dois casos, a polarização será circular. Isso mostra a dependência da antena transmissora com a antena receptora, pois caso uma não tenha a mesma polaridade da outra, elas não poderão se comunicar.

Como elas trabalham em frequências de RF, a impedância da antena e do circuito que esta conectado nela é muito importante. Caso eles não estejam casados, o circuito terá baixa eficiência de transmissão devido as reflexões que ocorrem antes do sinal chegar a antena.

Existem vários tipos de se implementar uma antena e cada tipo de antena possui as suas características específicas. A antena patch é uma antena feita basicamente de um componente irradiador, como ouro e cobre, um isolante e uma placa que será o terra. A emissão da onda eletromagnética ocorre devido aos campos entre o componente irradiador e o terra. Para melhor performance, um isolante grosso com constante dielétrica baixa deve ser usada. Para menor tamanho, deve ser usado um isolante com constante dielétrica alta resultando em menor eficiência e banda pequena.

O formato do componente irradiador é normalmente feito em figuras geométricas padrões como retângulo, círculo, triângulo, etc. Como exemplo, o formato retangular tem normalmente largura de 0.5 vezes o comprimento de onda e a profundidade dele deve ser centenas de vezes menor que o comprimento de onda.

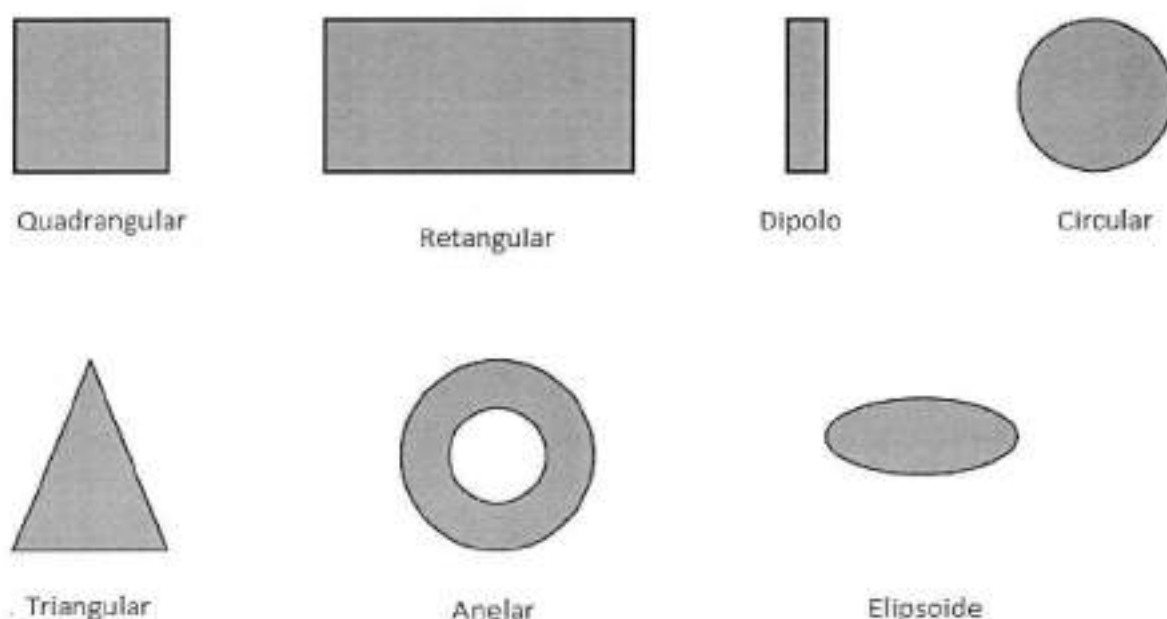


Figura 12: Formatos padrões [Fonte: <http://etd.lib.fsu.edu/theses/available/etd-04102004-143656/unrestricted/Chapter3.pdf>]

As antenas patch estão sendo cada vez mais usadas devido ao seu pequeno tamanho e volume. Elas já são utilizadas em celulares, pagers, mísseis e satélites.

Há dois modos de alimentar a antena, o modo com contato e sem contato. No modo com contato, a alimentação é ligada diretamente ao componente irradiador.

No modo sem contato, um casamento eletromagnético é feito para transferir o sinal da linha para o elemento irradiador.

2.5 Conceitos adicionais

2.5.1 Motor de passo

Motor de passo é um motor controlado por eletricidade que possui como diferencial a sua precisão de movimento. Apesar de sua velocidade e torque não serem muito grandes, a precisão que o motor de passo tem em seu movimento possibilita um grande controle, que já é utilizado em impressoras, câmeras, brinquedos, scanners, robôs, etc.

O princípio de funcionamento de um motor de passo é a utilização de solenóides alinhados dois a dois que atraem o rotor. Assim o rotor anda um ângulo pré-definido (o ângulo entre os solenóides) e fica fixo na posição mais próxima do solenóide magnetizado. Esse método é muito eficaz, trazendo um erro máximo de 5% e não cumulativo.

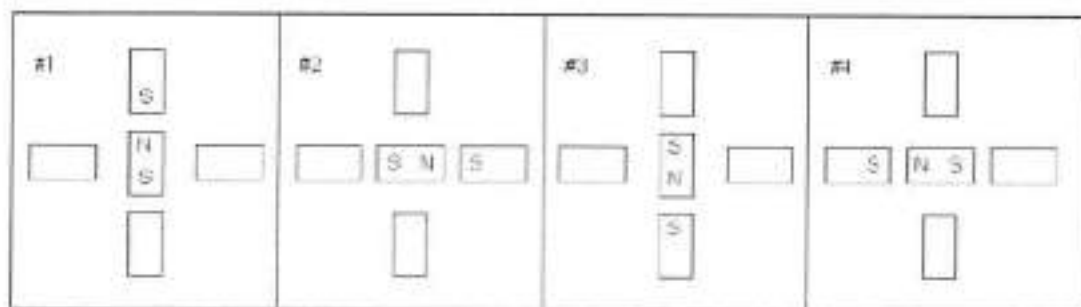


Figura 13 – Fases de um motor de passo unipolar de passo inteiro. [Fonte:

<http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>]

Um motor de passo é mais complexo para ser utilizado do que os outros motores elétricos. Enquanto nos outros motores você apenas coloca uma tensão nos terminais para fazer o motor funcionar, no motor de passo você precisa controlar qual solenóide está sendo polarizado e em qual sequência. Podemos assim imaginar o controle como ligando e desligando chaves que controlam os solenóides,

lembrando a lógica digital. Utilizando um microcontrolador e alterando a sua saída entre os números 1000, 0100, 0010 e 0001 podemos fazer um motor de passo dar uma volta completa com a velocidade que estamos alterando a saída.

Nº do passo	B3	B2	B1	B0	Decimal
1	1	0	0	0	8
2	0	1	0	0	4
3	0	0	1	0	2
4	0	0	0	1	1

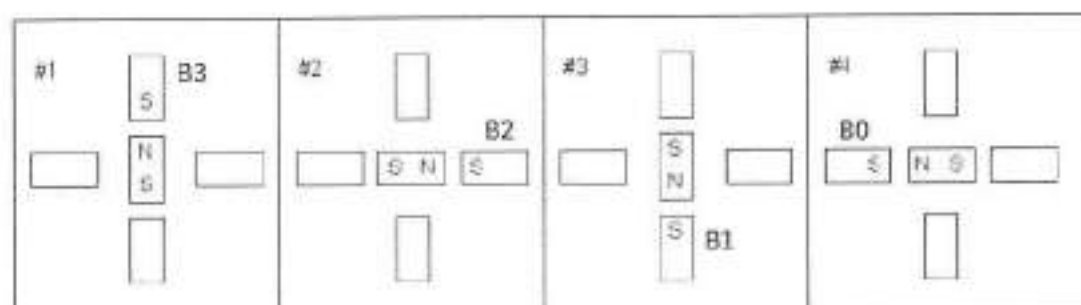


Figura 14 – Executando uma rotação completa em um motor de passo(Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>)

Assim podemos dizer que o motor de passo tem vantagens por poder ser controlado digitalmente, pela sua alta precisão no movimento, rápida resposta, mas tem como desvantagens o péssimo desempenho em altas velocidades, e a necessidade de uma estrutura mais complexa para poder não só controlá-lo como também para acioná-lo.

3. Desenvolvimento

3.1 Equipamentos utilizados

3.1.1 Módulo

Para fazer a localização do objeto vamos utilizar um aparelho localizador e um módulo que estará no objeto para fornecer ao localizador alguma informação que

o permita descobrir a localização do objeto. Esses módulos precisam estar ligados todo o tempo, pois eles precisam responder quando o usuário quiser localizá-los, e por isso um fator que precisa ser levado em consideração é o consumo da bateria. Além disso, como estamos adicionando um módulo ao objeto a ser localizado, se o tamanho do chip for muito grande, ele irá alterar a aparência do objeto e isso é indesejável. Como a bateria também tem dimensões consideráveis, precisamos utilizar fontes de energia menores, ocasionando em menor tempo de vida. Outro fator é que não queremos ficar trocando a bateria dos módulos sempre, pois isso seria um grande incômodo, principalmente se considerarmos vários objetos. A solução encontrada foi o módulo XBee® 802.15.4 serie 2 (figura 15) que possui todas as características desejáveis.



Figura 15 - XBee® (fonte: <http://www.skpang.co.uk>)

3.1.1.1 Dados sobre o XBee® 802.15.4 serie 2

Performance:

- Rendimento da Potência de saída: 1 mW (+0dBm);
- Alcance em ambientes internos/zonas urbanas: 30m;
- Alcance de RF em linha de visada para ambientes externos: 100m;
- Sensibilidade do receptor: -92dBm (1% PER);
- Frequência de operação: ISM 2.4 GHz;
- Taxa de dados de RF: 250kbps;

Alimentação:

- Tensão de alimentação: 2.8 à 3.4 VDC;
- Corrente de transmissão (típico): 45mA @ 3.3 VDC;
- Corrente de Recepção (típico): 50mA @ 3.3 VDC;

- Corrente de Power-downSleep: $<10 \mu\text{A}$ @ 25°C ;

Propriedades físicas:

- Dimensões: (2.438cm x 2.761cm);
- Peso: 0.10 oz (3g);
- Temperatura de operação: -40 to 85°C (industrial);
- Opções de antena: Conector U.FL RF, Chip ou Chicote (whip);

Rede :

- Imunidade a Interferência: DSSS (*DirectSequence Spread Spectrum*);
- Confiabilidade de entrega de pacotes: Retransmite novamente (*Retries*) & reconhecimento (*acknowledgements*);
- Topologia de Rede: *Peer-to-peer*(Par-a-par), ponto-a-ponto e ponto-a-multiponto;
- Endereçamento: 65.000 endereços de rede disponíveis para cada canal;
- Opções de filtros: PAN ID, canais e endereços;
- Criptografia: 128-bit AES;
- Número de canais selecionáveis via software: 12 canais de sequência direta;

Geral:

- Faixa de frequência: 2.4000 - 2.4835 GHz;

3.1.1.2 Kit de Desenvolvimento

Será utilizado um kit de desenvolvimentos fornecido pela loja virtual ZigBeeStore[12]. Esse é um kit que contém módulos, placas, cabos e manuais que facilitam o aprendizado, o teste e verificação da utilização do módulo XBee. Os itens do kit são os seguintes:

- Livro Xbee Desenvolvimento de aplicações
- INT700 Xbee placa serial - 03 unidades
- INT700 Xbee placa USB - 02 unidades
- Módulos Xbee" Extended range " modelo XBP24BZ7RPSMA - 03 unidades

- Módulos Xbee " Low range " modelo XB24Z7WHIP - 02 unidades
- Fonte de Alimentação - 03 unidades
- Antenas - 03 unidades
- Cabos de comunicação serial - 03 unidades
- Cabos de comunicação USB - 02 unidades
- Loopback - 03 unidades
- Vale treinamento AlcaboreStartUpZigBee

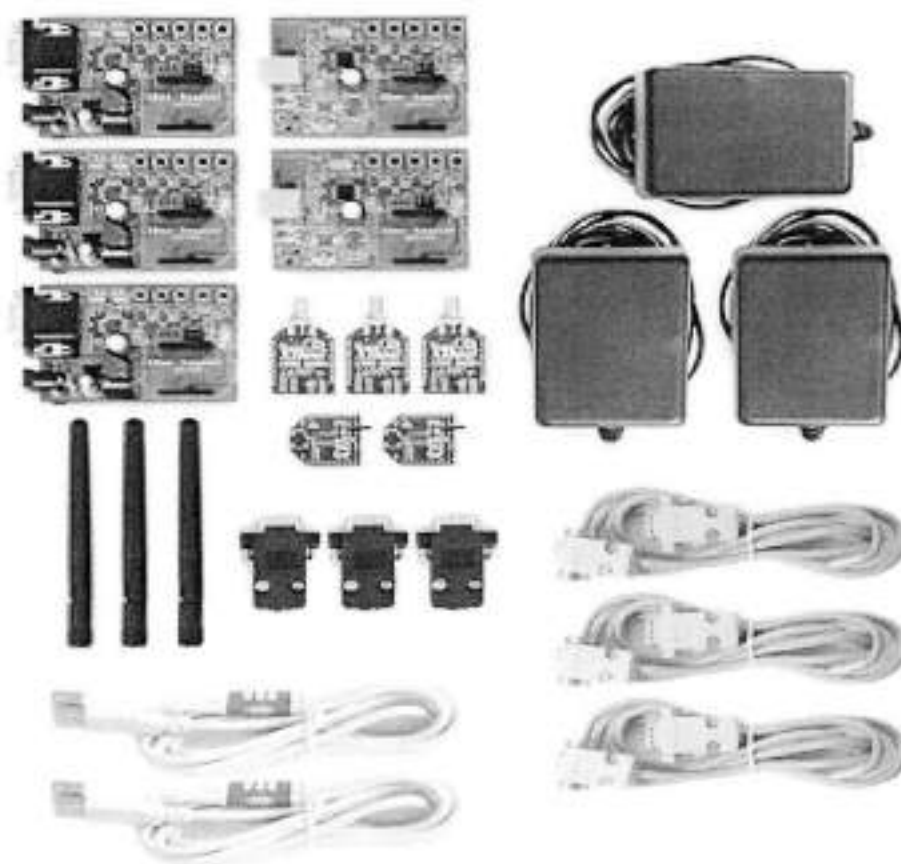


Figura 16 – kit Zigbee. (Fonte: www.fastcommerce.com.br)

3.1.2 Arduino Duemilanove

O Arduino Duemilanove ("2009") é uma placa baseada no microcontrolador ATmega168 ou ATmega328. Tem 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um de 16 MHz cristal oscilador, uma conexão USB, uma tomada de

força, um cabeçalho ICSP, e um botão de reset. Ele contém tudo o necessário para dar suporte ao microcontrolador, basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC para DC ou bateria. "Duemilanove" significa 2009 em italiano e é nomeado devido ao ano de seu lançamento. O Duemilanove é o último de uma série de placas Arduino USB.

Sumário(fonte: <http://arduino.cc/en/Main/ArduinoBoardDuemilanove>)

Microcontrolador	ATmega328
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limites)	6-20V
E / S digitais Pinos	14 (dos quais 6 oferecem saída PWM)
Analógica dos Pinos de Entrada	6
DC Current per I / O Pin	40 mA
Corrente DC 3.3V para Pin	50 mA
Memória Flash	16 KB (ATmega168) ou 32 KB (ATmega328), dos quais 2 KB usados por bootloader
SRAM	1 KB (ATmega168) ou 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) ou 1 KB (ATmega328)
Velocidade do relógio	16 MHz

3.1.3 Antena Patch

Utilizaremos uma antena patch para fazer a recepção do sinal do nosso localizador. Essa antena foi projetada pelo engenheiro Murilo HiroakiSeko e a nós foi emprestada para utilização neste projeto. Ela foi projetada para funcionar na frequência de 2.4 GHz, com lóbulo de AAA grause AA dbi.

3.1.4 Motor de Passo

Utilizaremos o motor de passo modelo M.P39 que possui as seguintes principais características:

Ângulo do passo:	$1.8^{\circ} \pm 5$
N° de Passos:	200
Tensão Nominal	12.0 Vdc
Corrente:	0.1 A/fase
Rotação:	150 – 900 RPM
Torque:	1.1 kg

3.2 Método de busca

3.2.1 Antena omnidirecional e localização por bips

Inicialmente optamos por utilizar uma antena omnidirecional para receber o sinal do enviado pelo objeto e utilizar um bip que mudaria com que frequência ele apitaria para mostrar ao usuário a distancia relativa do objeto. O localizador iria enviar um sinal para o objeto que responderia assim que recebesse. O localizador iria calcular o tempo que levou para a mensagem ir e voltar e com um cálculo simples, forneceria a distância que o objeto está do usuário, ou seja, mostrando uma circunferência onde o objeto pode estar. Quando o usuário andasse a distância iria aumentar ou diminuir e com isso ele poderia perceber se está se aproximando do objeto ou não. A idéia pode ser vista na figura 17.

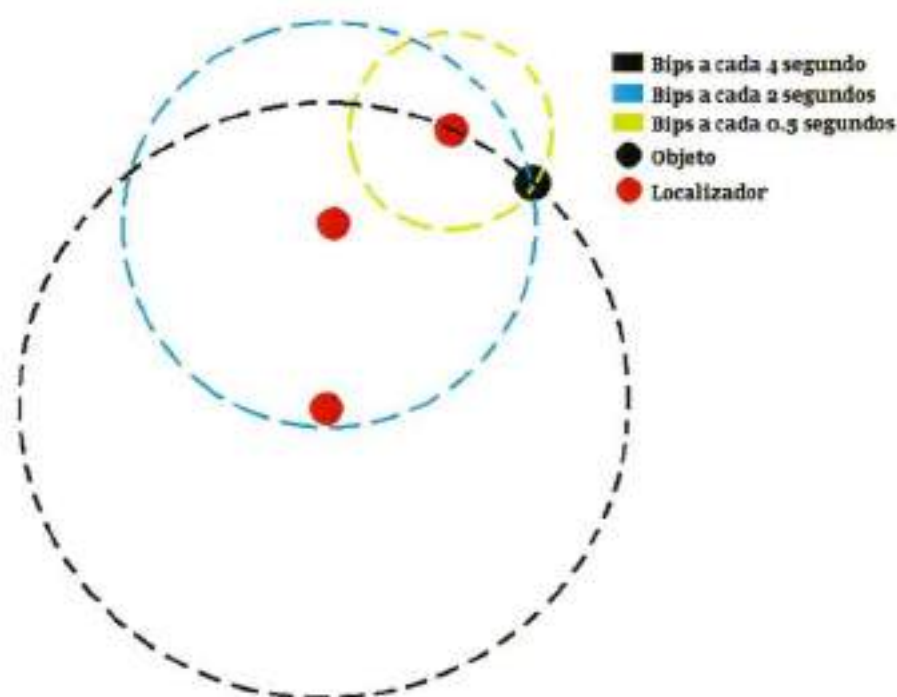


Fig 17 – Antena unidirecional e bips

Este método de localização possui algumas falhas. Primeiro o indicador por bips pode parecer bom, mas muitas vezes diferenciar o tempo entre os bips pode não ser tão fácil como parece, atrapalhando o usuário. Segundo, descobrir a direção que se deve seguir, tendo como a base a distância, pode ser bem difícil, pois temos que verificar varias possibilidades, lembrar sempre das posições anteriores e pensar logicamente, o que pode não ser muito fácil e trabalhoso, levando em conta que o equipamento deveria fazer isso para você. Outro problema muito importante é que quando não houver um caminho direto entre o objeto e o localizador, a distância fornecida é totalmente diferente da distância verdadeira, tornando a localização ainda mais difícil. Por estes motivos abandonamos esse método de localização.

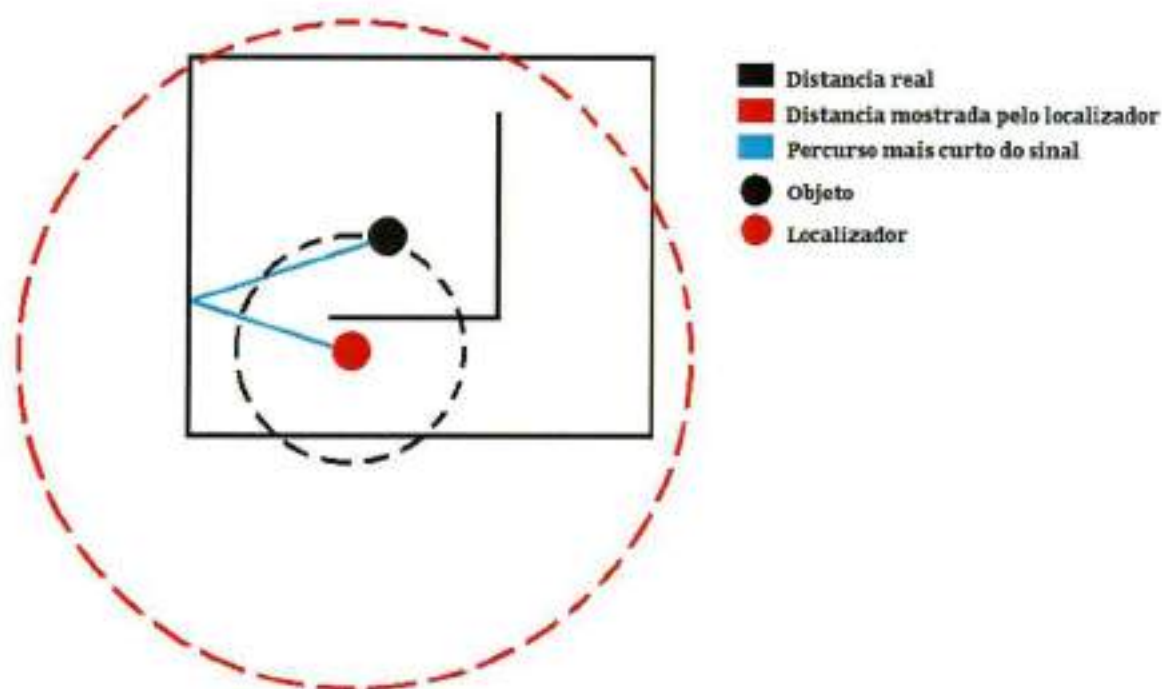


Fig 18 – Erro utilizando antena omnidirecional

3.2.2 Duas antenas direcionais e Lei dos senos

Outro método que estudamos para localizar o objeto é o de se utilizar duas antenas direcionais que girariam 360° e mediriam a potência do sinal recebido para descobrir de que direção está vindo a mensagem. Com isso teriam o ângulo em relação à antena que o objeto está e utilizando a lei dos senos poderiam localizar a distância do objeto.

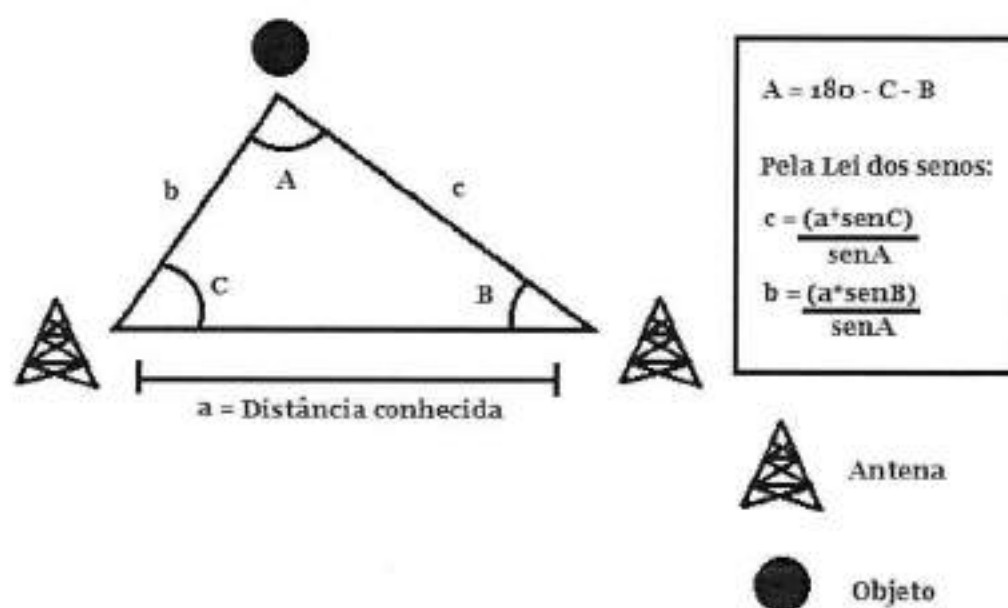


Fig. 19 – Duas antenas unidirecionais e Lei dos senos

Este método é bem interessante, pois pode fornecer a distância e a direção que o objeto se encontra. O problema deste método é que como desejamos fazer um aparelho localizador portátil, a distância entre as antenas é muito pequena em relação à distância até o objeto e isso ocasiona um erro muito grande na medida da distância. Como apenas a direção é uma medida confiável, optamos por usar um método mais simples, mais barato com o mesmo resultado.

3.2.3 Uma antena unidirecional

O método que vamos utilizar é parecido com o anterior, mas utilizamos apenas uma antena que irá girar 360° e nos informar em qual direção o objeto está. Assim o usuário vai seguindo esta direção até localizar o objeto desejado.

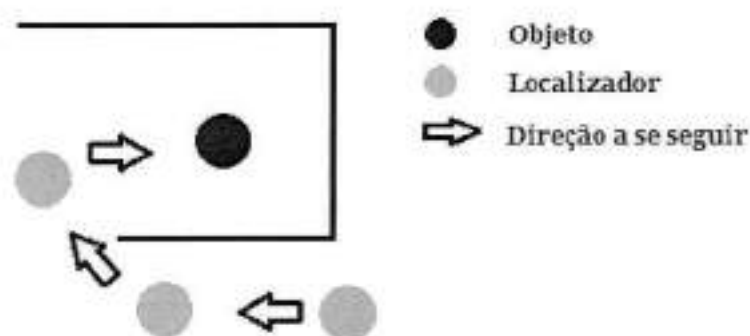


Fig. 20 – Uma antena unidirecional

Como pode ser visto no desenho, o aparelho irá indicar a direção que o usuário deve seguir para encontrar o objeto. Quando há obstáculos, a potência do sinal será mais alta no caminho feito pelo sinal onde a distância até o objeto é menor, pois a potência de um sinal diminui com o quadrado da distância[18]. Assim, apesar de não mostrar a verdadeira direção que está o objeto, ele mostra a direção do menor caminho que devemos seguir para encontrarmos o mesmo. Caso existam dois caminhos com a mesma medição de potência, qualquer um dos caminhos levará até o objeto, com distâncias aproximadamente iguais.

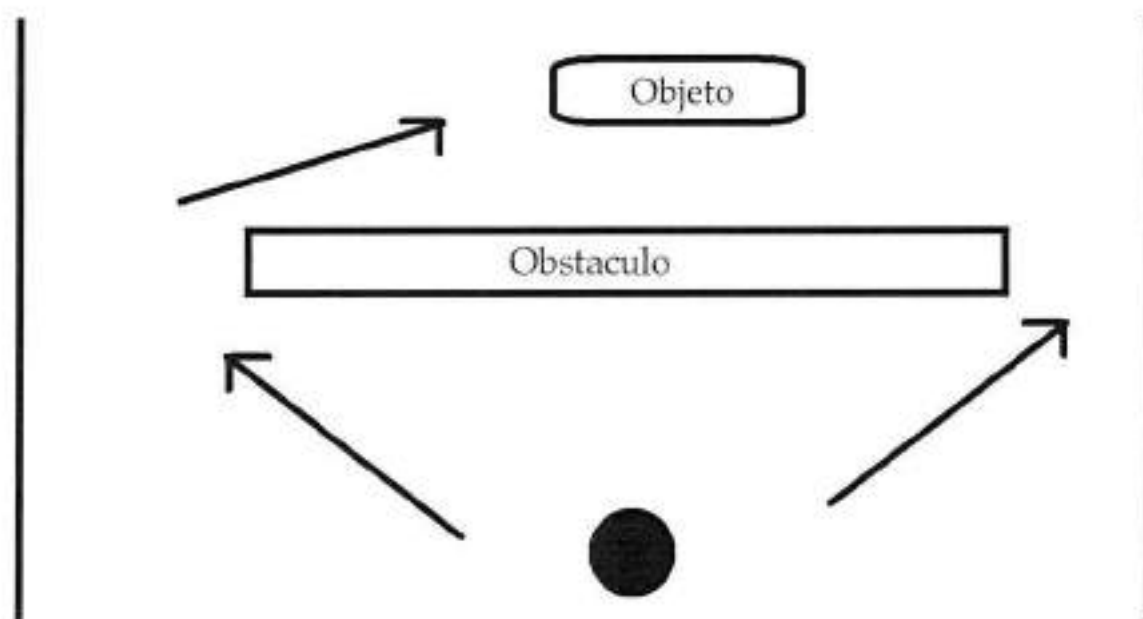


Fig. 21 – Dois caminhos com potências iguais

3.3 Implementação

Para implementar este método de localização, será utilizado 3 módulos ZigBee. Um será o responsável por enviar o sinal através de uma antena omnidirecional para o objeto que queremos localizar para garantir que o objeto receba. O segundo estará no objeto, que receberá o sinal do primeiro e responderá para o terceiro com uma antena omnidirecional, para garantir que o localizador receba. O terceiro receberá o sinal do segundo com uma antena unidirecional para poder medir a potencia com relação ao ângulo e assim descobrir a direção do objeto.

Para a antena poder girar os 360° e termos controle sobre o ângulo que estamos medindo, será utilizado um motor de passo.

Para termos todo o controle do sistema, será utilizado um microcontrolador, que será o coração do sistema, coordenando o motor e os módulos para trocarem mensagens e fazendo todo o cálculo para poder localizar o objeto.

3.3.1 Partes do Sistema

O sistema que será implementado pode ser basicamente separado em duas partes, o localizador e o chip que está no objeto.

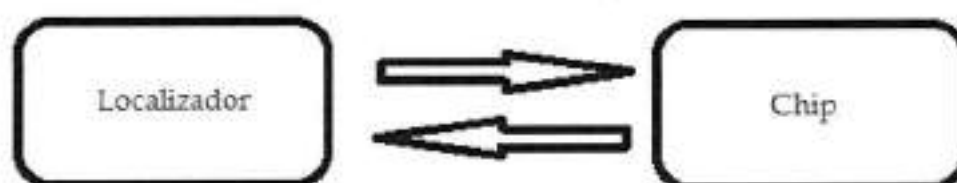


Fig. 22 – Partes do sistema simplificado

O chip que está no objeto apenas enviará de volta para o localizador a mensagem que recebeu, sendo assim que toda a inteligência do sistema fica no localizador. Este pode ser separado nos 3 principais elementos: chips XBees, Arduino e par antena/motor, que está interligado da seguinte forma:

simples: Primeiramente conecte seu dispositivo através de uma porta serial do seu computador (para fazer isso utilizamos uma das placas que vieram no kit de desenvolvimento). Na tela principal do software, você poderá escolher a porta que deseja usar para fazer a comunicação. Escolha a porta correta e aperte "*Test/Query*" para testar a comunicação. Se as informações do módulo aparecerem a comunicação está correta, caso contrário, tente verificar o motivo. Comunicando corretamente, vá na aba "*modem configuration*" e aperte "*Read*" para carregar todas as informações do seu módulo. Com as informações carregadas, você pode modificar qualquer uma delas e salvar através do botão "*Write*".

Para termos uma comunicação básica entre 2 módulos, por estarmos usando módulos da série 2, precisamos ter pelo menos um coordenador e o outro módulo podendo ser um roteador ou *enddevice*. Para alterar a função do módulo, altere o campo "*Function Set*" da aba "*modem configuration*" para a função desejada.

Outra aba interessante é a do "Terminal", pois permite enviar comandos diretamente para o módulo. Com esses comandos podemos mudar todas as configurações do módulo. Os comandos podem ser encontrados no manual do módulo XBee: ftp://ftp1.digi.com/support/documentation/90000976_H.pdf

3.3.2.2 Configurações

Como precisamos ter um coordenador na rede, este precisa estar no localizador, pois caso estivesse no objeto, o localizador teria que trocar de rede para localizar cada objeto e com isso limitaria o número de objetos localizáveis. Os outros dois módulos precisam ser roteadores, pois *enddevices* só podem conversar com os seus pais na rede e caso o coordenador configure de forma que não desejamos, o objeto não conseguirá falar diretamente com o módulo com antena unidirecional, impedindo a identificação da potência.

Assim o módulo que ficará no objeto será configurado como "roteador AT" e terá seu pino *Tx* curto-circuitado com o seu pino *Rx*, pois toda mensagem que ele receber ele deve devolver, para que o localizador possa determinar a sua posição. O

módulo que ficará no localizador e que terá uma antena omnidirecional também será configurado como "roteador AT". Ele será o módulo que enviará a mensagem para o objeto responder. O outro módulo que está no localizador, o que está com a antena unidirecional, será configurado como "coordenador AT", assim ele será o responsável por configurar a rede. A configuração básica necessária que foi utilizada em cada módulo é apresentada a seguir:

Localizador		Objeto
Coordenador AT	Roteador AT	Roteador AT
PAN ID: 777	PAN ID: 777	PAN ID: 777
Scan Channel: 7FFF	Scan Channel: 7FFF	Scan Channel: 7FFF
Node identifier: C	Node identifier: LB	Node identifier: R
No Sleep	No Sleep	Cyclic Sleep
	Destination node: C	Destination node LB

Tabela 2 – Configuração dos módulos

Com essas informações o módulo coordenador irá configurar a rede e o comando "destination node" irá definir o caminho que a mensagem irá percorrer (lembrando que o módulo que está no objeto precisa ter o seu pino de *Tx* e *Rx* curto-circuitado).

3.3.3 Arduino

O Arduino será o cérebro do dispositivo, pois coordenará todo o funcionamento do mesmo. Seu funcionamento seqüencial pode ser dividido da seguinte forma:

- Configuração dos módulos
- Loop de aquisição do valor de potencia
- Analise dos dados
- Informar resultado

3.3.3.1 Configuração dos módulos

Os módulos *XBee* podem ser configurados através da entrada serial do mesmo, utilizando uma sequência de comandos, sendo que o comando "+++" inicia o modo de comando. Como o coordenador já está previamente configurado, assim configurando a rede, o Arduino só precisará configurar o modulo roteador AT do localizador, que será feito através dos seguintes comandos:

Comando:	Resposta:
+++	OK<CR>
atndLB<CR>	OK<CR>

Tabela 3 – Comando para os módulos

Esses comandos definem para qual modulo será enviada a mensagem, ou seja, define qual objeto será localizado (nesse caso localizará o objeto que tenha como nome LB).

3.3.4 Loop de aquisição do valor de potência

A sequência de eventos que ocorre para a aquisição do valor de potencia é:

- ordenar que o modulo R envie uma mensagem
- Assim que o modulo C receber a mensagem, obter o valor de potencia e salvar em uma variavel
- Girar o motor e reiniciar o processo.

Para termos um valor mais confiável, obtivemos 5 valores de potência por rotação do motor para podermos tirar uma média.

3.3.5 Análise dos dados

Nos testes feitos percebemos que ao verificar o valor de potência, tivemos alguns valores inesperados, como picos. Isso ocorre devido a reflexões que podem fazer que em um determinado ponto a potência fique mais alta, mas isso não ocorre nos pontos perto do mesmo. Além disso, percebemos que há regiões onde temos potências igualmente altas, mas o objeto não se encontrava no meio dessa região, e sim tendendo para um lado. Nesses casos, podemos observar que ao lado dessa região de potências igualmente altas, tivemos umas regiões com potências altas também, que “puxavam” o objeto para esse lado. Essas observações podem ser vistas na figura abaixo.

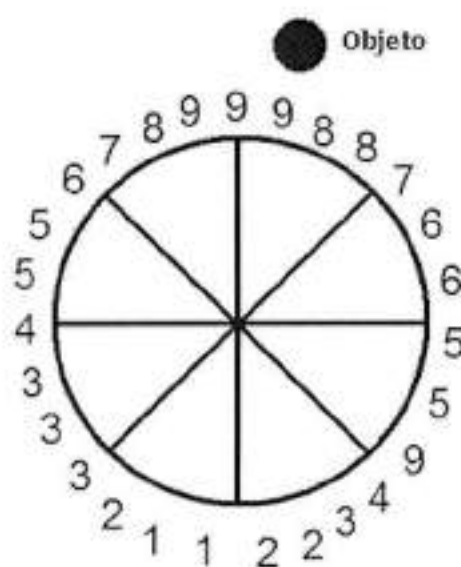


Fig. 24 – Potências recebidas

Para solucionar estes problemas, foi feito o seguinte algoritmo: Primeiramente é separado o círculo dos 360° em duas partes de 180° . Dessas partes é tirada a média de potência da região e ambas comparadas. A região da potência que tiver média maior receberá um ponto. Viramos esta divisão em 9° , ou seja, agora temos uma região que vai de 19° até 189° e outra que vai de 189° até 369° . Fazemos novamente a comparação e damos a pontuação. Quando rotacionarmos os 360° , o ângulo que tiver a maior pontuação será aquele que dirá a direção do objeto.

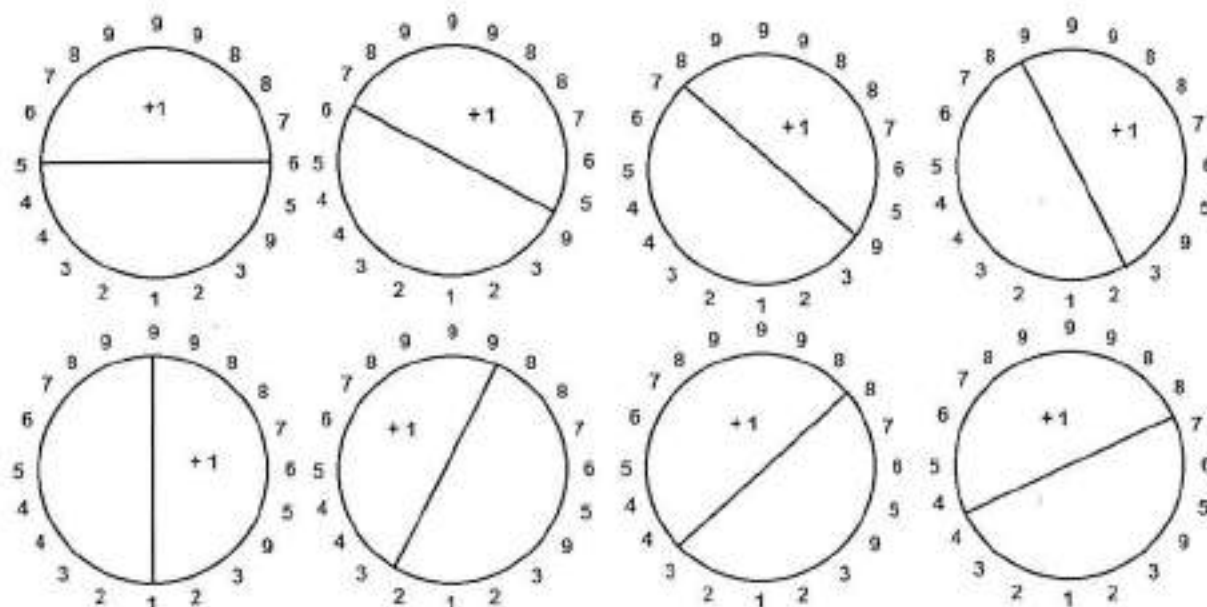


Fig. 25 - Algoritmo

3.3.6 Informar resultado

Com o valor do ângulo que diz a direção mais provável do objeto, mostramos através de um display de led a direção que usuário deve seguir.

3.3.7 Código

Será apresentado a seguir o código utilizado no Arduino, seguindo o que foi explicado nos itens anteriores:

```
#include <LiquidCrystal.h>
#define maxd 200
#define maxi 5

intmsg = 0;
int error = 0;
float por = 0;

intincoming = 0;           // Variavel para receber informacao serial
int v = 9;                 // Variavel do motor fio vermelho
int a = 8;                 // Variavel do motor fio amarelo
int d = 7;                 // Variavel do motor fio verde
int z = 6;                 // Variavel do motor fio azul
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Inicializacao do display LCD
intrssi = 0;               // Recebe o valor do RSSI
```

```

int dados [maxd];           // Vetor para armazenar os RSSIs
intcont = 0;                // Variavel para contar o angulo ou passos do motor
int cont2 = 0;              // Variavel com a mesma funcao do cont
intind = 0;                 // Variavel para contar quantos aquisicoes de RSSI fez por angulo
int ind2 = 0;               // Variavel com a mesma funcao do ind
intaux = 0;                // Variavel auxiliar
int aux2 = 0;              // Variavel com a mesma funcao do aux
int mais[40];               // Vetor para encontrar o angulo
int rep = 0;                // Vetor para angulos com a mesma chance de ser o angulo certo

void setup(){
  pinMode(v, OUTPUT);
  pinMode(a, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(z, OUTPUT);

  Serial.begin(9600);        // Inicializacao da porta serial
  lcd.begin(16, 2);          // Inicializacao do display 16 colunas por 2 linhas
  lcd.print("Projeto LOBSTER"); // Mensagem inicial
}

void loop() {
  while(Serial.available() > 0){
    incoming = Serial.read(); // Eliminar informacaonaodesejavel da entrada serial
  }
  Serial.println("Inicio!!");
  delay(1000);
  Serial.print("+++");
  delay(1000);
  if (Serial.available() > 0){
    if (Serial.read() == 79 || Serial.read() == 75 || Serial.read() == 13){ // Conferir se foi recebido o OK
      while(Serial.available() > 0){
        incoming = Serial.read(); // Eliminar informacaonaodesejavel da entrada serial
      }
      Serial.println("atdnLB");
      delay(1000);
      if (Serial.available() > 0){
        if (Serial.read() == 79 || Serial.read() == 75 || Serial.read() == 13){ // Conferir se foi recebido o OK
          while(Serial.available() > 0){
            incoming = Serial.read(); // Eliminar informacaonaodesejavel da entrada serial
          }
          // INICIO: INICIALIZAR O MOTOR
          digitalWrite(v, HIGH);
          digitalWrite(a, HIGH);
          digitalWrite(d, LOW);
          digitalWrite(z, LOW);
          // FINAL: INICIALIZAR O MOTOR
          while(true){
            for (cont2 = 0; cont2 < maxd; cont2++){ // Zerar o vetor de dados
              dados[cont2] = 0;
            }
            if (cont == 0){
              for (cont = 0; cont < maxd; cont++){
                for (ind = 0; ind < maxi; ind++){
                  Serial.print('a');
                  rssi = pulseIn(10, HIGH, 200);
                  aux = 0;

```

```

        while (rssi == 0){
rssi = pulseIn(10, HIGH, 200);
        aux++;
        if (aux > 20){
            if (digitalRead(10) == HIGH){
rssi = 200;
            }
            break;
        }
    }
    dados[cont] = dados[cont] + rssi;
}
    dados[cont] = dados[cont]/maxi;
    if(digitalRead(v) == HIGH &&digitalRead(a) == HIGH &&digitalRead(d) == LOW &&digitalRead(z) == LOW){
        digitalWrite(v, LOW);
        digitalWrite(d, HIGH);
    }
    else{
        if(digitalRead(v) == LOW &&digitalRead(a) == HIGH &&digitalRead(d) == HIGH &&digitalRead(z) ==
LOW){
            digitalWrite(a, LOW);
            digitalWrite(z, HIGH);
        }
        else{
            if(digitalRead(v) == LOW &&digitalRead(a) == LOW &&digitalRead(d) == HIGH &&digitalRead(z) ==
HIGH){
                digitalWrite(d, LOW);
                digitalWrite(v, HIGH);
            }
            else{
                if(digitalRead(v) == HIGH &&digitalRead(a) == LOW &&digitalRead(d) == LOW &&digitalRead(z) ==
HIGH){
                    digitalWrite(z, LOW);
                    digitalWrite(a, HIGH);
                }
            }
        }
    }
    delay(10);
}
cont = 0;
}
else{
    for (cont = maxd-1; cont>= 0; cont--){
        for (ind = 0; ind< maxi; ind++){
            Serial.print("a");
            rssi = pulseIn(10, HIGH, 200);
            aux = 0;
            while (rssi == 0){
rssi = pulseIn(10, HIGH, 200);
            aux++;
            if (aux > 20){
                if (digitalRead(10) == HIGH){
rssi = 200;
                }
                break;
            }
        }
    }
}

```



```

    }
    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}

// 9 e 189
aux = 0;
ind = 0;
for (cont2 = maxd/40; cont2 < maxd/2 + maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 1; cont2 < 21; cont2++){
        mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for(cont2 = 21; cont2 < 40; cont2++){
            mais[cont2]++;
        }
        for(cont2 = 0; cont2 < 1; cont2++){
            mais[cont2]++;
        }
    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}

// 18 e 198
aux = 0;
ind = 0;
for (cont2 = 2*maxd/40; cont2 < maxd/2 + 2*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 2*maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 2*maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 2; cont2 < 22; cont2++){
        mais[cont2]++;
    }
}
}

```

```

    else{
        if (aux < ind){
            for(cont2 = 22; cont2 < 40; cont2++){
mais[cont2]++;
            }
            for(cont2 = 0; cont2 < 2; cont2++){
mais[cont2]++;
            }
        }
        else{
            for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
            }
        }
    }

    // 27 e 207
    aux = 0;
    ind = 0;
    for (cont2 = 3*maxd/40; cont2 < maxd/2 + 3*maxd/40; cont2++){
        aux = aux + dados[cont2];
    }
    for (cont2 = maxd/2 + 3*maxd/40; cont2 < maxd; cont2++){
ind = ind + dados[cont2];
    }
    for (cont2 = 0; cont2 < 3*maxd/40; cont2++){
ind = ind + dados[cont2];
    }
    if (aux > ind){
        for(cont2 = 3; cont2 < 23; cont2++){
mais[cont2]++;
        }
    }
    else{
        if (aux < ind){
            for(cont2 = 23; cont2 < 40; cont2++){
mais[cont2]++;
            }
            for(cont2 = 0; cont2 < 3; cont2++){
mais[cont2]++;
            }
        }
        else{
            for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
            }
        }
    }

    // 36 e 216
    aux = 0;
    ind = 0;
    for (cont2 = 4*maxd/40; cont2 < maxd/2 + 4*maxd/40; cont2++){
        aux = aux + dados[cont2];
    }
    for (cont2 = maxd/2 + 4*maxd/40; cont2 < maxd; cont2++){
ind = ind + dados[cont2];
    }
    for (cont2 = 0; cont2 < 4*maxd/40; cont2++){

```

```

ind = ind + dados[cont2];
    }
    if (aux > ind){
        for(cont2 = 4; cont2 < 24; cont2++){
mais[cont2]++;
        }
    }
    else{
        if (aux < ind){
            for(cont2 = 24; cont2 < 40; cont2++){
mais[cont2]++;
            }
            for(cont2 = 0; cont2 < 4; cont2++){
mais[cont2]++;
            }
        }
        else{
            for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
            }
        }
    }
}
// 45 e 225
aux = 0;
ind = 0;
for (cont2 = 5*maxd/40; cont2 < maxd/2 + 5*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 5*maxd/40; cont2 < maxd; cont2++){
ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 5*maxd/40; cont2++){
ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 5; cont2 < 25; cont2++){
mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for(cont2 = 25; cont2 < 40; cont2++){
mais[cont2]++;
        }
        for(cont2 = 0; cont2 < 5; cont2++){
mais[cont2]++;
        }
    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
        }
    }
}
// 54 e 234
aux = 0;
ind = 0;

```

```

for (cont2 = 6*maxd/40; cont2 < maxd/2 + 6*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 6*maxd/40; cont2 < maxd; cont2++){
ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 6*maxd/40; cont2++){
ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 6; cont2 < 26; cont2++){
mais[cont2]++;
}
}
else{
    if (aux < ind){
        for(cont2 = 26; cont2 < 40; cont2++){
mais[cont2]++;
}
        for(cont2 = 0; cont2 < 6; cont2++){
mais[cont2]++;
}
}
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
}
}
}
// 63 e 243
aux = 0;
ind = 0;
for (cont2 = 7*maxd/40; cont2 < maxd/2 + 7*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 7*maxd/40; cont2 < maxd; cont2++){
ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 7*maxd/40; cont2++){
ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 7; cont2 < 27; cont2++){
mais[cont2]++;
}
}
else{
    if (aux < ind){
        for(cont2 = 27; cont2 < 40; cont2++){
mais[cont2]++;
}
        for(cont2 = 0; cont2 < 7; cont2++){
mais[cont2]++;
}
}
    else{
        for(cont2 = 0; cont2 < 40; cont2++){

```

```

mais[cont2]++;
}
}
}

// 72 e 252
aux = 0;
ind = 0;
for (cont2 = 8*maxd/40; cont2 < maxd/2 + 8*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 8*maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 8*maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for (cont2 = 8; cont2 < 28; cont2++){
        mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for (cont2 = 28; cont2 < 40; cont2++){
            mais[cont2]++;
        }
        for (cont2 = 0; cont2 < 8; cont2++){
            mais[cont2]++;
        }
    }
    else{
        for (cont2 = 0; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}

// 81 e 261
aux = 0;
ind = 0;
for (cont2 = 9*maxd/40; cont2 < maxd/2 + 9*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 9*maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 9*maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for (cont2 = 9; cont2 < 29; cont2++){
        mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for (cont2 = 29; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}

```

```

    }
    for(cont2 = 0; cont2 < 9; cont2++){
mais[cont2]++;
    }
}
else{
    for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
    }
}
}

// 90 e 270
aux = 0;
ind = 0;
for (cont2 = 10*maxd/40; cont2 < maxd/2 + 10*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 10*maxd/40; cont2 < maxd; cont2++){
ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 10*maxd/40; cont2++){
ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 10; cont2 < 30; cont2++){
mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for(cont2 = 30; cont2 < 40; cont2++){
mais[cont2]++;
        }
        for(cont2 = 0; cont2 < 10; cont2++){
mais[cont2]++;
        }
    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
        }
    }
}

// 99 e 279
aux = 0;
ind = 0;
for (cont2 = 11*maxd/40; cont2 < maxd/2 + 11*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 11*maxd/40; cont2 < maxd; cont2++){
ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 11*maxd/40; cont2++){
ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 11; cont2 < 31; cont2++){

```



```

mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for(cont2 = 31; cont2 < 40; cont2++){
            mais[cont2]++;
        }
        for(cont2 = 0; cont2 < 11; cont2++){
            mais[cont2]++;
        }
    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}

// 108 e 288
aux = 0;
ind = 0;
for (cont2 = 12*maxd/40; cont2 < maxd/2 + 12*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 12*maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 12*maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 12; cont2 < 32; cont2++){
        mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for(cont2 = 32; cont2 < 40; cont2++){
            mais[cont2]++;
        }
        for(cont2 = 0; cont2 < 12; cont2++){
            mais[cont2]++;
        }
    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}

// 117 e 297
aux = 0;
ind = 0;
for (cont2 = 13*maxd/40; cont2 < maxd/2 + 13*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 13*maxd/40; cont2 < maxd; cont2++){

```

```

ind = ind + dados[cont2];
    }
    for (cont2 = 0; cont2 < 13*maxd/40; cont2++){
ind = ind + dados[cont2];
    }
    if (aux > ind){
        for(cont2 = 13; cont2 < 33; cont2++){
mais[cont2]++;
        }
    }
    else{
        if (aux < ind){
            for(cont2 = 33; cont2 < 40; cont2++){
mais[cont2]++;
            }
            for(cont2 = 0; cont2 < 13; cont2++){
mais[cont2]++;
            }
        }
        else{
            for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
            }
        }
    }
    }

    // 126 e 306
aux = 0;
ind = 0;
for (cont2 = 14*maxd/40; cont2 < maxd/2 + 14*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 14*maxd/40; cont2 < maxd; cont2++){
ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 14*maxd/40; cont2++){
ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 14; cont2 < 34; cont2++){
mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for(cont2 = 34; cont2 < 40; cont2++){
mais[cont2]++;
        }
        for(cont2 = 0; cont2 < 14; cont2++){
mais[cont2]++;
        }
    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
mais[cont2]++;
        }
    }
}
}
}
}

```

```

// 135 e 315
aux = 0;
ind = 0;
for (cont2 = 15*maxd/40; cont2 < maxd/2 + 15*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 15*maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 15*maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for (cont2 = 15; cont2 < 35; cont2++){
        mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for (cont2 = 35; cont2 < 40; cont2++){
            mais[cont2]++;
        }
        for (cont2 = 0; cont2 < 15; cont2++){
            mais[cont2]++;
        }
    }
    else{
        for (cont2 = 0; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}
}

// 144 e 324
aux = 0;
ind = 0;
for (cont2 = 16*maxd/40; cont2 < maxd/2 + 16*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 16*maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 16*maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for (cont2 = 16; cont2 < 36; cont2++){
        mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for (cont2 = 36; cont2 < 40; cont2++){
            mais[cont2]++;
        }
        for (cont2 = 0; cont2 < 16; cont2++){
            mais[cont2]++;
        }
    }
}
}

```

```

    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}

// 153 e 333
aux = 0;
ind = 0;
for (cont2 = 17*maxd/40; cont2 < maxd/2 + 17*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 17*maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 17*maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 17; cont2 < 37; cont2++){
        mais[cont2]++;
    }
}
else{
    if (aux < ind){
        for(cont2 = 37; cont2 < 40; cont2++){
            mais[cont2]++;
        }
        for(cont2 = 0; cont2 < 17; cont2++){
            mais[cont2]++;
        }
    }
    else{
        for(cont2 = 0; cont2 < 40; cont2++){
            mais[cont2]++;
        }
    }
}

// 162 e 342
aux = 0;
ind = 0;
for (cont2 = 18*maxd/40; cont2 < maxd/2 + 18*maxd/40; cont2++){
    aux = aux + dados[cont2];
}
for (cont2 = maxd/2 + 18*maxd/40; cont2 < maxd; cont2++){
    ind = ind + dados[cont2];
}
for (cont2 = 0; cont2 < 18*maxd/40; cont2++){
    ind = ind + dados[cont2];
}
if (aux > ind){
    for(cont2 = 18; cont2 < 38; cont2++){
        mais[cont2]++;
    }
}
else{

```

```

        if (aux < ind){
            for(cont2 = 38; cont2 < 40; cont2++){
                mais[cont2]++;
            }
            for(cont2 = 0; cont2 < 18; cont2++){
                mais[cont2]++;
            }
        }
        else{
            for(cont2 = 0; cont2 < 40; cont2++){
                mais[cont2]++;
            }
        }
    }

    // 171 e 351
    aux = 0;
    ind = 0;
    for (cont2 = 19*maxd/40; cont2 < maxd/2 + 19*maxd/40; cont2++){
        aux = aux + dados[cont2];
    }
    for (cont2 = maxd/2 + 19*maxd/40; cont2 < maxd; cont2++){
        ind = ind + dados[cont2];
    }
    for (cont2 = 0; cont2 < 19*maxd/40; cont2++){
        ind = ind + dados[cont2];
    }
    if (aux > ind){
        for(cont2 = 19; cont2 < 39; cont2++){
            mais[cont2]++;
        }
    }
    else{
        if (aux < ind){
            for(cont2 = 39; cont2 < 40; cont2++){
                mais[cont2]++;
            }
            for(cont2 = 0; cont2 < 19; cont2++){
                mais[cont2]++;
            }
        }
        else{
            for(cont2 = 0; cont2 < 40; cont2++){
                mais[cont2]++;
            }
        }
    }

    // Calculando o maior
    aux = 0;
    ind = 0;
    rep = 0;
    for (cont2 = 0; cont2 < 36; cont2++){
        if (mais[cont2] > aux){
            aux = mais[cont2];
        }
        ind = cont2;
    }
    else{
        if (mais[cont2] == aux){

```

```

        rep = 1;
    }
}
}
aux = ind;
if (rep == 1){
    for (cont2 = 0; cont2 < 36; cont2++){
        if (mais[cont2] == mais[ind]){
            if (ind != cont2){
                if (cont2 == aux + 1){
                    aux = cont2;
                }
            }
            else{
                if (cont2 < 35){
                    aux2 = cont2;
                    for (ind2 = cont2 + 1; ind2 < 36; ind2++){
                        if (mais[cont2] == mais[ind2]){
                            if (ind2 == aux2 + 1){
                                aux2 = ind2;
                            }
                        }
                    }
                    if ((aux - ind) < (aux2 - cont2)){
                        aux = aux2;
                    }
                }
            }
        }
    }
    ind = (ind + aux)/2;
}
ind = (ind * 10) + 5;
// FINAL: CALCULO DE MEDIA

// INICIO: MOSTRAR AO USUARIO
Serial.println("Mostrando ao usuario");
lcd.setCursor(0,1);
lcd.print(" ");
lcd.setCursor(0,1);
lcd.print("Ang: ");
lcd.print(ind);
delay(10000); // VERIFICAR ESSE DELAY e o que vem abaixo
lcd.setCursor(0,1);
lcd.print(" ");
// FINAL: MOSTRAR AO USUARIO
}
}
else{
    Serial.println("Nao recebeu o OK do atdnLB");
}
}
else {
    Serial.println("Nao tem dados na porta Serial apos o atdnLB");
}
}
}

```

```

else {
  Serial.println("Nao recebeu o OK do +++");
}
}
else {
  Serial.println("Nao tem dados na porta Serial apos o +++");
}
}

```

4. Resultados Obtidos

Com a montagem apresentada, fizemos alguns testes e obtivemos os seguintes resultados:

4.2.1 Teste 1: Sem obstáculos

O primeiro teste que realizamos foi uma medida sem obstáculos. O resultado da medida é apresentado no grafico a seguir:

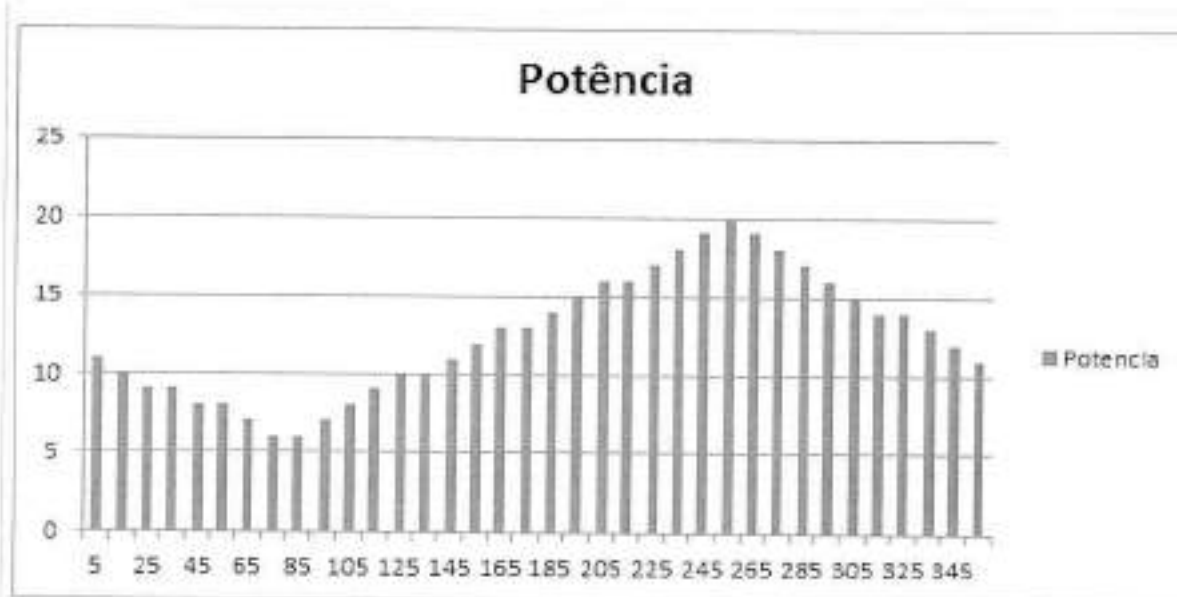
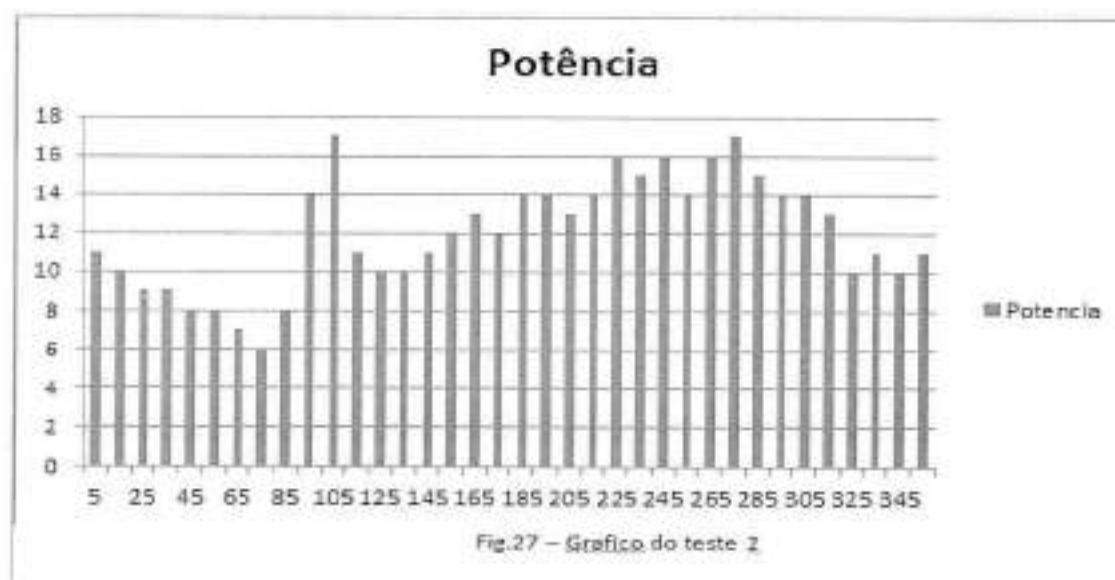


Fig. 26 – Gráfico do teste 1

O valor do eixo y do gráfico é o da pontuação dada pelo algoritmo. Podemos ver que uma maior pontuação foi dada para o ângulo 255, dizendo assim que o objeto está nesta direção.

4.2.2 Teste 1: Com obstáculos

O segundo teste realizado foi com obstáculos na direção direta até o objeto. O resultado foi o seguinte:



Como podemos ver na figura 27, com obstáculos a variação de potência fica mais "aleatória", com alguns picos imprevistos, devido a reflexões. Mesmo assim, podemos ver que a região perto do lugar do objeto tem ainda potência média mais alta.

5. Conclusão

Foi possível observar que é possível a confecção de um dispositivo localizador de objetos que utilize a potência como dado principal. As reflexões que existem das ondas eletromagnéticas causam muita diferença no valor esperado do valor real, mas através de um tratamento de dados, é possível ainda sim localizar o objeto ou a direção que devemos seguir para encontrá-lo.

Algumas limitações nos impediram de obter resultados melhores, como a limitação do RSSI, que não mostra valores de potências muito altos, assim quando

chegamos perto do objeto, a precisão se torna muito ruim, pois não temos como diferenciar os níveis de potência.

5.1 Contribuições/Inovações Esperadas

Foi feito um método de localizar objetos que funciona em ambientes fechados e abertos com obstáculos ou não. O método utilizado é simples e de fácil implementação, permitindo aparelhos pequenos que consumam baixa potência.

Uma ótima aplicação seria, por exemplo, em aeroportos, nos quais, em meio ao caos aéreo que os brasileiros têm enfrentado nos últimos anos, perder a mala, bolsa ou mochila é um problema vigente. E após o desembarque também seria uma maneira eficiente de identificar rapidamente a bagagem na esteira rolante. Outra seria para encontrar carros em estacionamentos, ou até mesmo localizar o seu carro quando o deixamos com o manobrista.

Os retornos são potencialmente altos, à medida que aprimora-se o projeto inicial até uma significativa redução de suas dimensões e desenvolve-se um possível dispositivo "adesivo" (a ser colado no objeto), que poderia ter grande aceitação no mercado, devido à facilidade que proporcionaria.

5.2 Considerações sobre o Curso de Graduação

A seguir estão listadas as disciplinas utilizadas na elaboração do trabalho:

PTC2359 - Engenharia de Comunicações: esta disciplina fornece os conceitos de sinais no domínio do tempo e frequência, propagação de sinais, além de modulação e demodulação de sinais. Tais conceitos são importantes no projeto, pois será utilizada a transmissão de dados entre o localizador e o chip, necessária para a obtenção da localização do chip.

PSI2431 - Propagação, Antenas e Microondas: esta disciplina fornece os conceitos básicos de propagação em espaço livre, além de conceitos básicos de antenas, como ganho da antena e área efetiva da antena. Estes conceitos são os

que determinam a distância máxima que haverá troca de sinal entre o localizador e o chip, parâmetros que dependem do funcionamento e ganho da antena.

MAC0122 - Princípios de Desenvolvimento de Algoritmos: nesta disciplina aprendemos desenvolver algoritmos e suas estruturas de dados básicas, conceitos necessários para o desenvolvimento do algoritmo de localização exata do objeto relativa à posição do usuário.

PSI2432 - Projeto e Implementação de Filtros Digitais: nesta disciplina aprendemos técnicas de projeto de filtros digitais, tais técnicas são necessárias para filtrar o sinal transmitido e recebido pelo localizador e pelo chip, pois estes sinais sofrerão interferências.

6.Referências

- [1] Bluetooth: <http://www.bluetooth.com>
- [2] Ad hoc: <http://www.cs.cornell.edu/home/ldzhou/adhoc.pdf>
- [3] ISM: <http://wireless.per.nl/reference/chaptr01/dtmmsyst/ism.htm>
- [4] ZigBee: <http://www.zigbee.org/>
- [5] Padrao IEEE 802.15.4 A Comprehensive Performance Study of IEEE 802.15.4:
http://www-ee.ccny.cuny.edu/zheng/papers/paper1_wpan_performance.pdf
- [6] Wifi: <http://www.wi-fi.org/>
- [7] LAN: <http://www.pulsewan.com/data101/pdfs/introlan.pdf>
- [8] WLAN: <http://www.pulsewan.com/data101/pdfs/introlan.pdf>
- [9] Antena isotrópica:
http://professores.unisanta.br/santana/downloads%5CTelematica%5CMicroondas_2%5CAntenas%20e%20Propagatpo%5CAntenas_MUITO_BOM.pdf
- [10] JC Digital: <http://jcdigital.blogspot.com/2008/10/o-que-uma-rede-zigbee.html>
- [11] Zigbee Alliance: <http://www.zigbee.org/>
- [12] ZigBee Store: www.zigbeestore.com.br
- [13] Eady, F. Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers. Newnes. Oxford 2007.
- [14] Farahani S. ZigBee Wireless Networks and Tranceivers. Newnes. Oxford 2008.
- [15] Filho, José Carlito de Oliveira. Revista Saber Eletrônica, São Paulo, edição de 10/2010.
- [16] Gislason, D. ZigBee Wireless Networking. Newnes. Oxford 2008.
- [17] Gutierrez, Jose A. Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15.4
- [18] Potencia de um sinal: <https://www.mar.mil.br/dhn/bhmn/download/cap-34.pdf>

[19] Ross, John. The Book of Wireless: A Painless Guide to Wi-Fi and Broadband Wireless

[20] A Comprehensive Performance Study of IEEE 802.15.4

http://www-ee.ccny.cuny.edu/zheng/papers/paper1_wpan_performance.pdf

7. Bibliografia Geral

Arduino Playground <http://www.arduino.cc/playground/Shields/Xbee01>

Association for Automatic Identification and Mobility

<http://www.aimglobal.org/technologies/> - acessado em 18/06/11

Callaway, Edgar H, Wireless Sensor Networks: Architectures and Protocols, Auerbach Publications, 2003

Jupiterweb <http://sistemas2.usp.br/jupiterweb/> - acessado em 23/06/11

ZigBee Alliance. www.zigbee.org - acessado em 20/06/11

ZigBee Alliance. ZigBee Wireless Sensor Applications for Health, Wellness and Fitness. ZigBee Alliance. March 2009.

ZigBee Wireless Technology for Low-Power Sensor Networks - www.commsdesign.com/showArticle.jhtml?articleID=192200323

<http://www2.atmel.com/> - acessado em 15/06/11

<http://www.daintree.net/resources/index.php> - acessado em 16/06/11

<http://www.ieee802.org/15/pub/TG4.html> - acessado em 17/06/11

<http://www.zigbee.org/LearnMore/WhitePapers.aspx> - acessado em 19/06/11

http://www.clarinox.com/docs/whitepapers/RealTime_main.pdf - acessado em 20/06/11

http://www.ember.com/products_zigbee_chips_e260.html - acessado em 20/06/11

<http://www.digi.com/> - acessado em 25/06/11

http://usuarios.upf.br/~fpassold/PIC/C_PIC.PDF - acessado em 25/06/11

<http://sensor-networks.org/index.php?page=1010510536> - acessado em 22/06/11

http://www.freescale.com/webapp/sps/site/overview.jsp?code=ZIGBEE_SMARTENERGY - acessado em 24/06/11

<http://www.fastcommerce.com.br> – acessado em 26/06/11

<http://www.skpang.co.uk> - acessado em 20/06/2011

http://ftp1.digi.com/support/documentation/90000982_B.pdf - acessado 25/06/11

<http://www.bluetooth.com> – acessado em 16/08/11

<http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>– acessado em 29/09/11

<http://www.ppgel.ufsj.edu.br/uaisoccer/downloads/1272062510.pdf>– acessado em 29/09/11