



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Implementação e Análise do nRF24L01+ como beacon
Bluetooth Low Energy

Heitor Fonseca de Araujo

Recife, 2016

Heitor Fonseca de Araujo

Implementação e Análise do nRF24L01+ como beacon Bluetooth Low Energy

Monografia apresentada ao Centro de Informática
da Universidade Federal de Pernambuco para
obtenção do título de Bacharel em Engenharia da
Computação.

Orientador: Kiev Gama.

Recife, 2016

Agradecimentos

Agradeço a minha família, em especial aos meus pais, Jefferson Araújo e Maria dos Anjos, por minha criação, exemplo, apoio, educação, amor, cuidado, confiança, e suporte sem medida para minha contínua formação como pessoa e profissional. Aos meus irmãos Erick, Victor e Manuela por toda força e motivação. Agradeço ao meu avô Possidônio, que me acolheu em recife e da sua maneira sempre cuidou de mim e as minhas avós Osvaldina e Anadir a qual dedico esse trabalho.

Agradeço ao meu orientador Kiev Gama, por ter aceitado me orientar e por me ajudar na produção desse trabalho e no meu crescimento acadêmico durante a graduação. Agradeço também aos profissionais do CESAR, Sergio, Claudio, Dio e Rebeca, que me deram a oportunidade de estagiar na empresa e que me ajudaram na concepção e implementação desta pesquisa.

Sou grato ao Centro de Informática que através dos professores contribuíram de forma significativa com o meu crescimento profissional, e funcionários que colaboram facilitando processos; por tantas oportunidades oferecidas e pelo conhecimento construído aqui através desses anos.

Agradeço aos meus amigos, de dentro e fora do CIn, que sempre me ajudaram e me apoiaram. Agradeço aos companheiros dos diversos projetos ao longo das disciplinas, e aos meus colegas de turma, em especial, Arthur, David, Djeeffther, Guilherme, Maria, Jefferson, João Gabriel, Renato, Saulo, Guilherme, Yago, Italo e Marcílio com quem desde o primeiro período faço projeto.

Embora o CIn tenha me proporcionado muita tristeza e noites mal dormidas, conheci pessoas que me ensinaram muito mais do que um assunto de uma disciplina e cresci muito como pessoa nesses cinco anos. Gostaria de agradecer a Gabriel, Leonardo, em especial, a Ihago e Kerolayne por terem me ensinado tantas coisas boas que vou levar pro resto da minha vida. Agradeço a Luiza pelo grande companheirismo e pela revisão deste trabalho.

Agradeço a todos que participaram dessa etapa da minha vida com palavras de motivação, apoio e força para não desistir da caminhada. Sou muito grato por tudo!

Resumo

Nos últimos anos, a internet das coisas (IoT) tem chamado grande atenção de pesquisa. IoT é considerado como uma parte da Internet do futuro e vai incluir bilhões de "coisas" inteligentes se comunicando. O futuro da Internet consistirá em dispositivos conectados de forma heterogênea que ampliarão ainda mais as fronteiras do mundo com entidades físicas e componentes virtuais. Os beacons BLE surgem como um componente contextual chave de IoT [5] por possuir um protocolo simples e compatibilidade quase universal com dispositivos móveis. Por conta de seu baixo preço e economia de energia, o rádio nRF24L01+ está sendo usado cada vez mais nos dispositivos IoT como módulo de comunicação. Este trabalho tem como objetivo desenvolver uma biblioteca open-source e low energy do nRF24L01+ para a plataforma Arduino pro-mini e UNO, tornar o nRF24L01+ compatível com BLE como um beacon de aplicação broadcasting e disponibilizar uma API (Application programming interface) do nRF24L01+ beacon para que desenvolvedores que utilizam o rádio como sensor de comunicação possa tornar seu sistema compatível com BLE. Foi realizado a prova de conceito, caso de uso e testes de alcance e bateria. Os resultados mostraram que é possível tornar o nRF24L01+ compatível com BLE *broadcasting* com alcance e consumo de bateria compatíveis com os beacons BLE disponíveis no mercado.

Lista de Figuras

Figura 1 - Canais BLE e WIFI na banda de 2.4GHz.

Figura 2 - Formato do pacote de dados do BLE.

Figura 3 - Pacotes de dados do BLE com campos do cabeçalho.

Figura 4 - Pacote de dados BLE com dado broadcasting.

Figura 5 - Tipos de dados formatados no dado Broadcast.

Figura 6 - Flags que indicam as capacidades de operação do dispositivo.

Figura 7 - Formato do Manufacturer Specific Data

Figura 8 - Intervalo *Advertising*.

Figura 9 - Tipos de alimentação para beacons, fonte de energia (esquerda), bateria (direita).

Figura 10 - Módulo nRF24L01+ e nRF24L01+ com PA e LNA.

Figura 11 - Ilustração do nRF24L01+ no modo de recepção.

Figura 12 - Formato do pacote de dados do nRF24L01+.

Figura 13 - Pinos do módulo nRF24L01+.

Figura 14 - Arduino pro mini e seus respectivos pinos.

Figura 15 - Arduino IDE.

Figura 16 - Fluxo de empacotamento e desempacotamento do dado.

Figura 17 - Esquema de conexão utilizado no arduino pro mini e o nRF2401+.

Figura 18 - Pacote do nRF24L01+ compatível com BLE broadcasting.

Figura 19 - Visualização do beacon nrf24l01+ no nrf connect master.

Figura 20 - Flags (direita) e gráfico RSSI (esquerda) do nrf24l01+ beacon.

Figura 21 - Esquema de conexão dos sensores do sistema SECO.

Figura 22 - Visualização do pacote do sistema SECO compatível com BLE.

Figura 23 - Alcance do beacon nRF24L01+ e nRF24L01+ com PA e LNA variando a potência de transmissão.

Lista de Tabelas

Tabela 1 - Principais características do Bluetooth Low Energy.

Tabela 2 - Tipos de PDU e o respectivo nome do pacote.

Tabela 3 - Configurações de potência do nRF24L01+ e seu respectivo consumo de corrente.

Tabela 4 - Duração da bateria em horas do beacon nRF24L01+.

Sumário

1. Introdução
 - 1.1. Motivação e contextualização
 - 1.2. Objetivos
 - 1.3. Estrutura do Trabalho
2. Fundamentação Teórica
 - 2.1. Bluetooth Low Energy
 - 2.1.1. Beacons
 - 2.1.2. Formato do Pacote
 - 2.1.3. Intervalo *Advertising*
 - 2.1.4. Potência
 - 2.1.5. Alcance
 - 2.1.6. Protocolos
 - 2.2. nRF24L01+
 - 2.2.1. Modos Operacionais
 - 2.2.2. Taxa de transmissão de dados sem fio
 - 2.2.3. Frequência do canal de RF
 - 2.2.4. Controle de amplificador de potência
 - 2.2.5. Formato do Pacote
 - 2.3. Arduino
3. Compatibilidade nRF24L01+ e BLE
4. Metodologia
 - 4.1. Tecnologias e ferramentas utilizadas
 - 4.2. Desenvolvimento da biblioteca do nRF24L01+ para Arduino
 - 4.3. O módulo nRF24L01+ como beacon
5. Experimentos e Resultados
 - 5.1. Prova de conceito
 - 5.2. Transmitindo dados de sensores.
 - 5.3. Testes de alcance
 - 5.4. Teste de bateria
 - 5.5. Custo
6. Considerações finais e trabalhos futuros
7. Referências Bibliográficas

1. Introdução

1.1. Motivação e contextualização

O conceito de Internet das Coisas (Internet of Things - IoT) foi inicialmente introduzido por Kevin Ashton em 1999 [1]. A ideia inicial propunha a utilização de tags do tipo Radio-Frequency Identification (RFID) para que os objetos pudessem ser identificados em um ambiente, permitindo assim que os objetos se comuniquem entre eles para resolver um objetivo comum, considerando o contexto em que estão inseridos [7]. Com isso, os objetos passaram a carregar partes lógicas da aplicação com eles, o que lhes permitiram interagir com os seus usuários de forma mais intuitiva, levando ao conceito de objetos inteligentes [2].

Os objetos inteligentes são, portanto, objetos do cotidiano que passam a ser capazes de entender e reagir ao ambiente em que estão inseridos. Tais objetos podem ser vistos como blocos de construção para a Internet das Coisas [2]. Ao interagirem juntos, de forma combinada ou complementar, estes objetos produzem um ambiente inteligente, que pode ser definido como aquele que é capaz de adquirir e aplicar conhecimentos sobre si próprio e sobre as pessoas, a fim de melhorar a experiência destas no ambiente [3]. Esses ambientes inteligentes podem ser aplicados em diversos domínios para melhorar a qualidade de vida das pessoas. Já são realidade ambientes inteligentes em casas, escritórios, sistemas de transporte público e estacionamentos [4], projetados para uso do público geral. Os requisitos específicos de diferentes domínios de aplicações afetam a viabilidade de determinadas aplicações como, segurança, autonomia da bateria, alcance, custo e etc.

Beacons BLE (Bluetooth Low energy) são pequenos sensores sem fios que usam sinais de rádio para enviar um alerta de sua presença. Eles continuamente transmitem um simples sinal de rádio, que podem ser capturados por diversos dispositivos à exemplo dos smartphones via Bluetooth Low energy (também conhecido como Bluetooth *Smart* ou Bluetooth Versão 4.0+). Através de um aplicativo compatível com o BLE, o smartphone pode acionar uma determinada ação e permitir que pessoas, lugares e coisas se conectem à Internet indiretamente,

sendo assim um grande habilitador para internet das coisas, no varejo por exemplo, o beacon permitiria individualizar ações de abordagem (criar promoções, ofertas ou notificações específicas) do consumidor. Como o protocolo em si é simples, os beacons têm compatibilidade quase universal com dispositivos móveis e, como a conexão estabelecida pode ser usada para melhorar a experiência e a produtividade do usuário, elas são consideradas o componente contextual chave da Internet das Coisas (IoT) [5]. A tecnologia Beacon possibilita um novo e extenso mundo de possibilidades de conectividade e inúmeras oportunidades de interatividade entre dispositivos móveis e hardware existente.

Os principais requisitos a ser levado em conta ao desenvolver um sistema de beacons de IoT é o custo, alcance de sinal e autonomia da bateria. O preço de um único beacon BLE pode variar aproximadamente de U\$10\$ a U\$50 [6]. A considerar do tipo de alimentação, um beacon pode ser classificado como autônomo ou não (alimentados por uma fonte de energia). Alguns beacons têm uma vida útil de apenas alguns meses, enquanto outros podem executar até 5 anos. Outro fator é o alcance de sinal que o beacon é capaz de transmitir, chegando até 200 metros de distância. O preço também é um fator variável, pois depende das características do hardware, como resistência a água, ou com recursos extras como sensores de movimento e/ou temperatura.

Os módulos de comunicação são responsáveis pela comunicação entre dispositivos da rede IoT. Alguns dos módulos candidatos para a comunicação são os módulos WiFi, Bluetooth, ZigBee e módulos transmissores RF. O nRF24L01+ é um chip transceptor de baixo custo, projetado para aplicações sem fio operando em 2.4Ghz, de baixa potência onde a velocidade de comunicação pode chegar a 2Mbps. O módulo, da fabricante Nordic Semiconductors, fornece uma solução Ultra Low Power, permitindo de meses a anos de vida útil se alimentado com pilhas do tipo AA/AAA [7]. Devido ao uso da interface SPI, é possível interligar esse módulo à maioria dos microcontroladores disponíveis atualmente se tornando assim um grande facilitador para a internet das coisas.

1.2. Objetivos

Este trabalho tem como objetivo implementar uma biblioteca open-source e low energy do nRF24L01+ para a plataforma Arduino pro-mini e Arduino UNO, validar o nRF24L01+ como um beacon de aplicação broadcasting, disponibilizar uma API (Application programming interface) do nRF24L01+ beacon para que desenvolvedores que utilizam o rádio como sensor de comunicação possa tornar seu sistema compatível com Bluetooth Low Energy e comparar com outros beacons em termos de alcance, consumo de energia e preço.

1.3. Estrutura do trabalho

Para atingir os objetivos apresentados, este trabalho está organizado da seguinte forma: o capítulo 2 aborda a fundamentação teórica do trabalho, detalhando o bluetooth low energy e beacon, explicando um pouco da sua camada física, o formato de pacote, abordando seus requisitos de bateria, alcance, potência. O capítulo 2 apresenta também o módulo nRF24L01+ e o seu funcionamento, explicando sobre sua camada física e seu formato de pacote, seus modos de operação, requisitos de potência, consumo de energia e taxa de transmissão. O capítulo 2 finaliza explicando sobre o microcontrolador Arduino que foi utilizado detalhando seus pinos e interface SPI para configuração do nRF24L01+. O capítulo 3 apresenta qual as funcionalidades do BLE que podem ser reproduzidas pelo nRF24L01+ e quais procedimentos que o nRF24L01+ deve implementar para se tornar compatível com o BLE. No capítulo 4 é apresentado a metodologia do trabalho indicando no capítulo 5 os testes e resultados obtidos. Por fim o capítulo 6 apresenta as considerações finais e trabalhos futuro.

2. Fundamentação teórica

2.1. Bluetooth Low Energy

Bluetooth Low Energy é uma tecnologia sem fio de baixa potência desenvolvida pelo Bluetooth Special Interest Group (SIG) e característica distintiva da especificação Bluetooth 4.0 [8]. Em contraste com os padrões Bluetooth anteriores, o BLE foi introduzido para facilitar uma comunicação de curto alcance para dispositivos que não requerem grande quantidade de transferência de dados. A ideia principal do BLE era fornecer uma tecnologia eficiente para aplicações de monitoramento e controle onde as quantidades de dados são tipicamente muito baixas, como o envio de valores de sensor ou mensagens de controle curtas com mínimo overhead [9].

Muitos recursos do Bluetooth clássico são herdados no BLE. Vários novos recursos tornam os dispositivos BLE fáceis de configurar e tornam uma solução poderosa para a propagação de sinais de RF em ambientes fechados. Os dispositivos BLE operam na banda sem licença de 2.4GHz Industrial Scientific Medical (ISM). Esta comunicação sem fio define 40 canais de radiofrequência (RF) com espaçamento de canal de 2MHz. BLE só tem dois tipos de canais dedicados: anúncios (ou *advertising*) e dados. Os canais de anúncios são usados para a descoberta de dispositivos, transmissão de mensagem broadcast e configuração de conexão. Os dispositivos BLE que operam em modo *advertising* transmitem periodicamente informações broadcasting em três canais dedicados (37, 38 e 39) especificamente atribuídos para minimizar a colisão com os canais Wi-Fi mais utilizados (1, 6 e 11) [10] como mostra a Figura 1 abaixo.

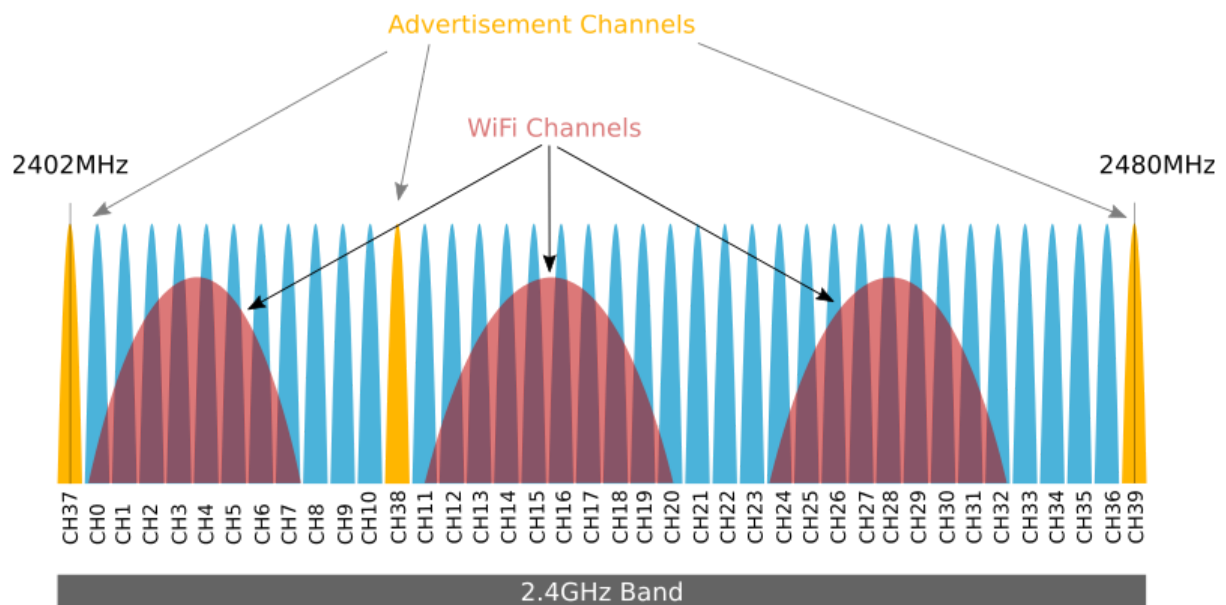


Figura 1 - Canais BLE e WIFI na banda de 2.4GHz.

Todos os canais físicos usam um esquema de modulação GFSK (Gaussian Frequency Shift Keying), que permite reduzir o consumo de energia de pico. A faixa de cobertura de BLE é tipicamente mais de várias dezenas de metros. A taxa de modulação para BLE é fixada em 1 Mbps, que é o limite de processamento físico superior para esta tecnologia [8]. O BLE é otimizado para enviar pequenos pedaços de informação com atraso mínimo (latência). O tempo total de envio de dados é geralmente inferior a 6 ms (em comparação com 100 ms com Bluetooth clássico). A Tabela 1 mostra algumas das principais características do BLE [11].

Faixa de frequência	2.400GHz - 2483.5GHz
Distância/Alcance	< 100 metros
Taxa de dados nominal	1 Mbps
Modulação (técnica / Esquema)	AFH/GFSK
Canais (Número / Largura de banda)	40 / 2 MHz
Latência	< 6ms
Corrente de pico/média	<15mA/~ μ A
Segurança	128-bit AES

Tabela 1 - Principais características do Bluetooth Low Energy.

2.1.1. Beacons

Um beacon em tecnologia sem fio é o conceito de transmissão de pequenos pedaços de informação. A informação pode ser qualquer coisa, desde dados ambientais (temperatura, pressão de ar, umidade, etc.) até dados de micro-localização (rastreamento de ativos, varejo e assim por diante) ou dados de orientação (aceleração, rotação e assim por diante). Os dados transmitidos são tipicamente estáticos, mas também podem ser dinâmicos e alterar ao longo do tempo. Com o uso de Bluetooth Low Energy, beacons podem ser projetados para executar por anos em uma bateria de moeda de única célula.

O Bluetooth low energy é ideal para beacons por ser de baixa potência e o ecossistema já está implantado na maioria dos smartphones ou outros dispositivos habilitados para BLE no mercado [3]. O baixo consumo de energia é conseguido mantendo o tempo de transmissão o mais curto possível e permitindo que o dispositivo entre no modo de espera entre as transmissões. Dispositivos BLE pode operar somente em um modo não-conectável de advertising (onde toda a

informação está contida no advertising), mas eles podem também permitir conexões [12].

O beacon não conectável é um dispositivo BLE no modo de *broadcasting*. Ele simplesmente transmite informações armazenadas internamente. Como a transmissão não conectável não ativa nenhuma capacidade de recebimento, ela alcança o menor consumo de energia possível, simplesmente despertando, transmitindo dados e voltando a dormir. O beacon conectável, em contrapartida, é um dispositivo BLE no modo periférico, o que significa que ele não só pode transmitir, mas também receber. Isso permite que um dispositivo central, um smartphone por exemplo, se conecte e interaja com serviços implementados no dispositivo de beacon. Os serviços fornecem uma ou mais características que podem ser modificadas por um dispositivo *peer*. Um exemplo dessas características pode ser uma sequência de dados que representa as informações *broadcast*. Desta forma, é possível ter um beacon configurável que seja facilmente atualizado.

2.1.2. Pacote de dados

Os dados transmitidos a partir de um dispositivo de BLE são formatados de acordo com a especificação do bluetooth [8]. O pacote consiste em 4 campos distintos: Preâmbulo (1 byte), endereço de acesso (4 bytes), PDU (Protocol Data Unit) (2 a 39 bytes) e CRC (3 bytes), tal como se pode ver na Figura 2. O tamanho mínimo de um pacote BLE é então 10 bytes (80 bits) e o maior pacote possível pode ter até 47 bytes (376 bits).



Figura 2 - Formato do pacote de dados do BLE.

O preâmbulo é um valor de 1 byte usado no receptor para executar a sincronização da frequência e estimativa de tempo no receptor. Será sempre 0xAA para pacotes de anúncio. O endereço de acesso é um número de 32 bits, único e gerado aleatoriamente pelo dispositivo no estado de iniciação e enviado no pedido

de conexão. Para pacotes de anúncio o valor é fixo definido como 0x8E89BED6. O CRC é um mecanismo de identificação de erros e é calculado sobre o PDU do pacote. É um CRC polinomial de 24 bits e todos os bits do PDU devem ser processados pela ordem de transmissão, ou seja, do menos para o mais significativo. O polinômio CRC tem a forma de $x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$. Relativos ao PDU, existem dois tipos, dependendo se o pacote é transmitido nos canais de anúncios ou nos canais de dados.

2.1.2.1. PDU dos Canais de Anúncio

O PDU, que diz respeito aos canais de anúncio, é constituído por um cabeçalho (header) de 2 bytes, visível na Figura 3, seguido do respectivo payload, que é específico para o tipo de PDU. Nesta categoria existem 7 subtipos de PDUs.

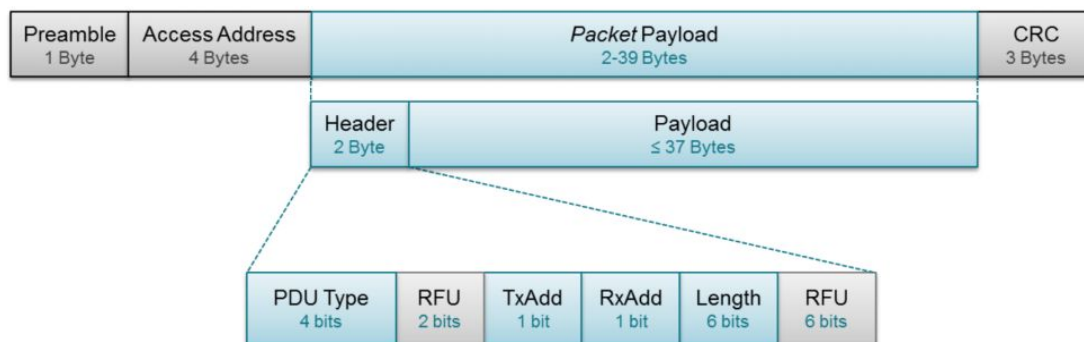


Figura 3 - Pacotes de dados do BLE com campos do cabeçalho.

O campo Header é composto por 6 subcampos. O PDU Type indica qual o tipo de PDU, pois dentro da categoria de PDUs de canais de anúncio existem 7 tipos possíveis. Estes tipos são descritos na Tabela 2. Os campos TxAdd e RxAdd tomam valores binários (zero ou um), dependendo do tipo de PDU. O bit TxAdd indica se o endereço do anunciante (contido no Payload) é público (TxAdd = 0) ou aleatório (TxAdd = 1). O campo Length é o tamanho do payload do PDU em bytes, que como está descrito na Figura 3 pode ir desde 6 a 37 bytes.

Tipo de PDU	Nome do Pacote
0000	ADV_IND

0001	ADV_DIRECT_IND
0010	ADV_NONCONN_IND
0100	SCAN_RSP
0101	CONNECT_REQ
0110	ADV_SCAN_IND

Tabela 2 - Tipos de PDU e o respectivo nome do pacote.

A PDU *CONNECT_REQ* é usada para requisições de conexões. Para aplicações de broadcasting, as PDUs utilizadas são *ADV_IND* e *ADV_NONCONN_IND* que são descritos como conectáveis e não conectáveis respectivamente, enquanto *ADV_SCAN_IND* é simplesmente um broadcaster não conectável que pode fornecer informações adicionais por meio de respostas de scan.

O payload do pacote inclui o endereço do anunciante juntamente com os dados de anúncio definidos pelo usuário, conforme mostrado na Figura 4. Estes campos representam o endereço e dados transmitidos pelo beacon.

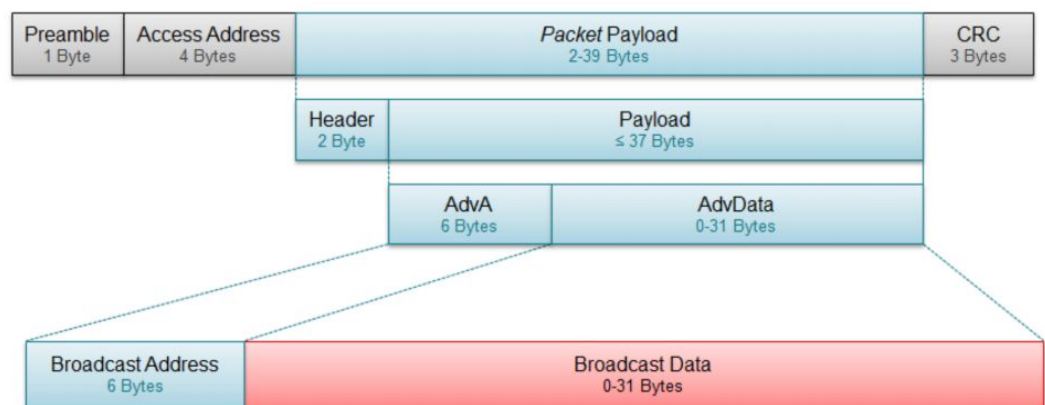


Figura 4 - Pacote de dados BLE com dado broadcasting.

Os dados broadcast podem ser formatados de acordo com os formatos de dados especificados na especificação do bluetooth [8], com alguns exemplos mostrados na Figura 5.

AD Data Type	Data Type Value	Description
Flags	0x01	Device discovery capabilities
Service UUID	0x02 - 0x07	Device GATT services
Local Name	0x08 - 0x09	Device name
TX Power Level	0x0A	Device output power
Manufacturer Specific Data	0xFF	User defined

Figura 5 - Tipos de dados formatados no dado Broadcast.

Flags (três primeiros bytes dos broadcast data) definem as capacidades do dispositivo e o formato pode ser visto na Figura 6. A diferença de *Limited Discoverable Mode* para *General Discoverable Mode* é somente restrição de tempo, onde se o BLE estiver configurado como Limited, depois de 180s não será capaz de detectá-lo [11]. A flag *BR/EDR Not supported* indica que o dispositivo não suporta bluetooth clássico (*Bluetooth Basic Rate/Enhanced Data Rate*). O *local Name* define o nome do dispositivo.

Byte	Bit	Flag/Value	Description
0		0x02	Length of this data
1		0x01	GAP AD Type Flags
2	0	LE Limited Discoverable Mode	180 s advertising
	1	LE General Discoverable Mode	Indefinite advertising time
	2	BR/EDR Not Supported	
	3	Simultaneous LE and BR/EDR (Controller)	
	4	Simultaneous LE and BR/EDR (Host)	
	5-7		Reserved

Figura 6 - Flags que indicam as capacidades de operação do dispositivo.

O Manufacturer Specific Data define O ID do fabricante e o dado a ser enviado. O seu formato pode ser visualizado na Figura 7.

Byte	Value	Description
0	0x03-0x1F	Length of this data
1	0xFF	Manufacturer-Specific Data Flag
2	0x0D	Company ID
3	0x00	(Example, 0x000D – Texas Instruments)
4 - 31	-	User Defined Data (optional)

Figura 7 - Formato do Manufacturer Specific Data

2.1.3. Intervalo de *advertising*

Um beacon consegue baixo consumo de energia residindo em estado de *sleep* a maior parte do tempo de operação, apenas acordando brevemente para transmitir dados [11]. O tempo entre esses eventos de transmissão é referido como intervalo de *advertising*, que é ilustrado na Figura 4. Para beacons não conectáveis, o intervalo não pode ser menor que 100 ms. Para beacons conectáveis, não pode ser menor que 20 ms. Para estes intervalos, um atraso pseudo-aleatório de 0-10ms é adicionado. Esta aleatoriedade ajuda a reduzir a possibilidade de colisões entre propagandas de diferentes dispositivos.



Figura 8 - Intervalo Advertising.

O intervalo de *advertising* é escolhido com base nos requisitos de consumo de energia versus latência. Intervalo mais alto permite mais tempo no modo de *sleep*, mas aumenta o tempo que um observador pode obter para obter um pacote broadcast.

2.1.4. Potência

Um beacon pode ser alimentado usando vários métodos. Existem três diferentes principais abordagens: Fonte de alimentação CC (USB, Rede, etc.), Baterias (CR2032, AAA, lítio e assim por diante) e absorção de energia (solar, cinética, e assim por diante). Alguns exemplos de alimentação podem ser visto na Figura 9. Normalmente, a escolha principal são as baterias que permitem projetos de produtos pequenos e de baixo custo, enquanto mantêm uma vida útil para a maioria das aplicações. A escolha do tipo de bateria é importante porque algumas baterias podem ser sensíveis a altas correntes de pico. A capacidade necessária é

baseada na frequência com que a mensagem *broadcast* é transmitida, e se qualquer processamento adicional é necessário (interação do sensor, algoritmos, e assim por diante). Se o projeto é alimentado por fonte de alimentação CC, a suposição seria que o consumo de energia não é um parâmetro crítico. Se ainda é crítica, a abordagem seria a mesma que para um projeto alimentado por bateria.

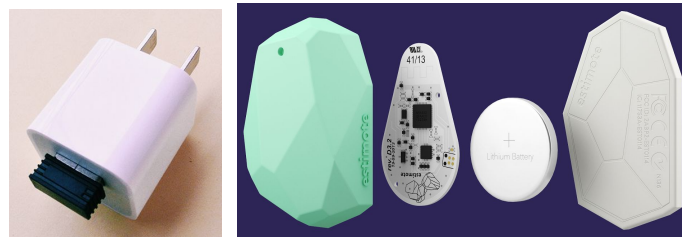


Figura 9 - Tipos de alimentação para beacons, fonte de energia (esquerda), bateria (direita).

2.1.5. Alcance

Na teoria de Radiofrequência, o alcance é dado por uma série de fatores como sensibilidade do receptor de rádio, potência de saída do transmissor de rádio, ambiente e desempenho da antena [9]. Uma aplicação de beacon pode exigir alcance de centímetros até várias centenas de metros. A potência de saída máxima permitida pela especificação do bluetooth [8] é de 10 dBm mas o valor normalmente varia entre -30 dBm e +4 dBm. O alcance do beacon é de tecnicamente até 70 metros. Em condições reais, no entanto, deve-se esperar até 40-50 metros [13].

2.1.6. Protocolos

Hoje existem dois principais protocolos/firmwares para beacons BLE, chamados, iBeacon [14] e Eddystone [17]. iBeacon criado pela Apple e é de código fechado, enquanto Eddystone é criado pelo Google e é de código aberto. A principal diferença é o formato do pacote transmitido pelos beacons. O iBeacon transmite apenas um pacote de advertising que tem um número de identificação exclusivo composto de três partes - UUID, *Major* e *Minor*, enquanto Eddystone especifica três tipos UID, URL e TLM). O Eddystone-UID é fundamentalmente idêntico ao do iBeacon. Todos os beacons que transmitem Eddystone-UID estão registradas no banco de dados do Google. O Eddystone-URL indica ao dispositivo para abrir um

URL, dessa forma, não exige que um aplicativo de desenvolvedor específico seja pré-instalado. O Eddystone-TLM envia dados de sensor. Não é de objetivo desse trabalho especificar os pacotes de cada protocolo.

2.2. nRF24L01(+)

O nRF24L01 + é um transceptor de único chip de 2,4 GHz fabricado pela Nordic Semiconductor, adequado para aplicações sem fio de ultra baixa potência [16]. Semelhante ao bluetooth e wifi, o nRF24L01+ foi projetado para operar na faixa de frequência 2.4GHz Industrial Scientific Medical (ISM). O rádio custa menos de 1\$ e existe também a versão do nRF24L01+ com PA (Power Amplifier) e LNA (Low Noise Amplifier), estes adicionam um circuito de amplificador de potência e circuito de Amplificador de baixo ruído, que permite transmitir distâncias mais longas e ter desempenho mais estável para os padrões da indústria. Este chega a custar pouco mais de 2\$. Os módulos nRF24L01+ disponíveis no mercado pode ser visto na Figura 10.

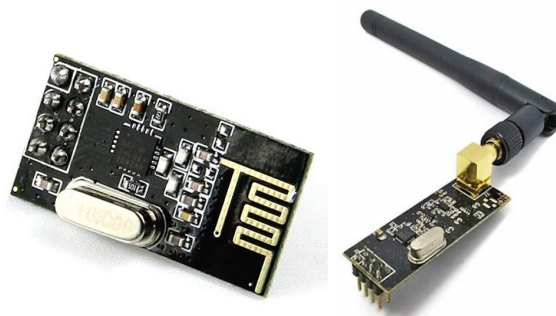


Figura 10 - Módulo nRF24L01+ e nRF24L01+ com PA e LNA.

O rádio usa a modulação GFSK e possui parâmetros configuráveis pelo usuário como canal de frequência, potência de saída e taxa de transmissão de dados sem fio. O nRF24L01+ suporta taxas de transmissão de dados de 250 kbps, 1 Mbps e 2Mbps. A alta taxa de transmissão de dados combinada com dois modos de economia de energia tornam o nRF24L01+ muito adequado para projetos de baixa energia [16].

O usuário pode operar e configurar o nRF24L01+ através de uma interface periférica serial (SPI). O mapa de registradores, acessível através do SPI, contém

todos os registradores de configuração no nRF24L01+ e está acessível em todos os modos de operação do chip.

O rádio possui a característica multi-receptor utilizada no modo de recepção de dados que contém um conjunto de seis canais de dados (*pipes*) paralelos com endereços únicos. Um *pipe* de dados é um canal lógico no canal físico de RF. Cada canal de dados tem a sua própria decodificação de endereço físico (endereço de *pipe* de dados) no nRF24L01+. O nRF24L01+ configurado como receptor pode receber dados endereçados a seis canais de dados diferentes em um canal de frequência como mostrado na Figura 11. Cada *pipe* de dados tem seu próprio endereço exclusivo e pode ser configurado para um comportamento individual.

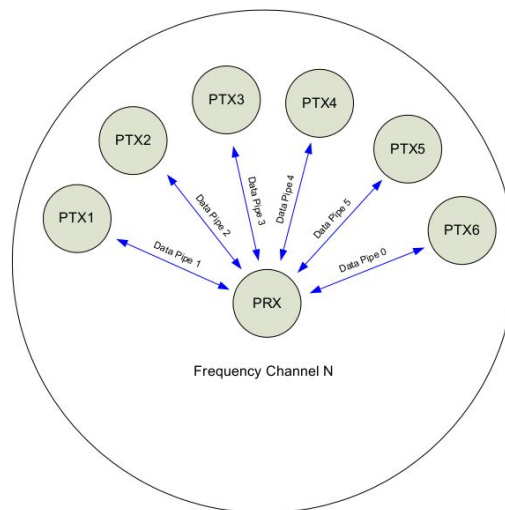


Figura 11 - Ilustração do nRF24L01+ no modo de recepção.

2.2.1. Modos operacionais do nRF24L01+

2.2.1.1. Modo *Power Down*:

No modo de Power Down, o nRF24L01+ é desativado com mínimo consumo de corrente. Todos os valores dos registradores disponíveis são mantidos e o SPI é mantido ativo, permitindo a mudança de configuração e o upload de registradores.

2.2.1.2. Modo Standby-I

O modo *Standby-I* é usado para minimizar o consumo médio de corrente enquanto mantém curtos tempos de inicialização. Neste modo, somente parte do oscilador de cristal está ativo.

2.2.1.3. Modo Standby-II

No modo *standby-II*, os buffers de clock extra estão ativos e mais corrente é usada em comparação com o modo *standby-I*. O rádio entra no modo *standby-II* quando está no modo de transmissão esperando dado, ou seja TX FIFO está vazia e não tem nada para transmitir.

2.2.1.4. Modo RX

O modo RX é um modo ativo onde o rádio nRF24L01+ é usado como um receptor. No modo RX, o receptor demodula os sinais do canal de RF. O mecanismo de protocolo de banda base procura constantemente um pacote válido. Se for encontrado um pacote válido (por um endereço correspondente e um CRC válido), o payload do pacote é colocado em um espaço vazio nas RX FIFO (O nRF24L01+ possui 3 RX FIFO). Se as RX FIFO estiverem cheias, o pacote recebido é descartado.

2.2.1.5. Modo TX

O modo TX é um modo ativo para a transmissão de pacotes. Para entrar neste modo, o nRF24L01+ tem que ter um payload na sua TX FIFO (o nRF24L01+ possui 3 TX FIFO). O nRF24L01+ permanece no modo TX até que termine de transmitir o pacote. Se o TX FIFO não estiver vazio, o nRF24L01+ permanece no modo TX e transmite o próximo pacote. Se o TX FIFO estiver vazio, o nRF24L01+ entra no modo *standby-II*.

2.2.2. Taxa de Transmissão de Dados sem Fio

A taxa de transmissão de dados sem fio é a taxa de sinalização modulada que o nRF24L01+ utiliza quando transmite e recebe dados. Pode ser 250kbps, 1Mbps ou 2Mbps. A utilização de uma taxa de dados sem fio inferior proporciona uma melhor sensibilidade do receptor do que uma taxa de dados de ar superior. Mas, a alta taxa de dados sem fio dá menor consumo de corrente média e menor probabilidade de colisões no ar.

2.2.3. Frequência do canal de RF

A frequência do canal de RF determina o centro do canal usado pelo nRF24L01+. O canal ocupa uma largura de banda inferior a 1MHz a 250kbps e 1Mbps e uma largura de banda inferior a 2MHz. NRF24L01+ pode operar em frequências de 2.400GHz até 2.525GHz. A resolução de programação da configuração da frequência do canal de RF é 1MHz. Em 2Mbps o canal ocupa uma largura de banda maior do que a resolução da configuração de frequência do canal de RF. Para garantir canais não sobrepostos no modo 2Mbps, o espaçamento entre canais deve ser de 2MHz ou mais. Em 1Mbps e 250kbps a largura de faixa do canal é a mesma ou mais baixa do que a definição da frequência do RF.

2.2.4. Controle de amplificador de potência

O controle amplificador de potência é usado para ajustar a potência de saída do amplificador de potência nRF24L01 +. No modo TX, o controle PA possui quatro opções de configuração. A tabela 3 mostra as possíveis configurações de potência e o respectivo consumo de corrente.

Potência RF de saída	Consumo de corrente DC
0dBm	11.3mA
-6dBm	9.0mA
-12dBm	7.5mA
-18dBm	7.0mA

Tabela 3 - Configurações de potência do nRF24L01+ e seu respectivo consumo de corrente.

2.2.5. Formato do pacote

A Figura 12 abaixo mostra o formato de pacote com o bit mais significativo a esquerda. O nRF24L01+ permite ao usuário desabilitar o campo CRC, tornando-o opcional.

Preamble 1 byte	Address 3-5 byte	Payload 1 - 32 byte	CRC 1-2 byte
-----------------	------------------	---------------------	--------------

Figura 12 - Formato do pacote de dados do nRF24L01+.

2.2.5.1. Preâmbulo

O preâmbulo é uma sequência de bits utilizada para sincronizar o demodulador de receptores com o fluxo de bits de entrada. O preâmbulo tem um byte de comprimento e é 01010101 ou 10101010. Se o primeiro bit no endereço é 1, o preâmbulo é automaticamente ajustado para 10101010 e se o primeiro bit é 0, o preâmbulo é automaticamente ajustado para 01010101. Isto é feito para garantir que haja suficientes transições no preâmbulo para estabilizar o receptor.

2.2.5.2. Endereço

Este é o endereço para o receptor. Um endereço garante que o pacote seja detectado e recebido pelo receptor correto, evitando conversações acidentais entre vários sistemas nRF24L01+. Pode ser de 3 a 5 bytes.

2.2.5.3. Payload

O payload é o conteúdo do pacote definido pelo usuário. Ele pode ter de 0 a 32 bytes de largura e é transmitido ao ar quando carregado na TX FIFO do nRF24L01+ e entra no modo de transmissão.

2.2.5.4. CRC

O CRC é o mecanismo obrigatório de detecção de erros no pacote. É 1 ou 2 bytes e é calculado sobre o endereço e Payload. O polinômio para CRC de 1 byte é $x^8 + x^2 + x + 1$ com o valor inicial de 0xFF. O polinômio para CRC de 2 bytes é $x^{16} + x^{12} + x^5 + 1$ com o valor inicial de 0xFFFF.

2.2.5.5. Interface de Dados e Controle

A interface de dados e controle dá acesso a todos os recursos do nRF24L01+ e pode ser vista na Figura 13.



Figura 13 - Pinos do módulo nRF24L01+.

A interface de dados e controle consiste nos seguintes seis sinais digitais 5V:

- IRQ (pino de interrupção)
- CE (pino chip enable, usado para ativar o chip no modo RX ou TX)
- CSN (pino chip select not, usado para o sinal SPI)
- SCK (pino serial clock pelo qual o mestre controla o pulso/clock da conexão)
- MOSI (pino pelo qual o mestre envia dados ao periférico)
- MISO (pino pelo qual o mestre recebe dados ao periférico).

2.3. Arduino

Arduino é uma plataforma open-source de prototipagem eletrônica com hardware e software flexíveis e fáceis de usar, destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos [16]. O arduino pode receber sinais de vários sensores eletrônicos e lidar com essas informações para controlar motores, leds e qualquer outro atuador. O modelo utilizado no projeto é um arduino pro mini 5V-16MHz (Figura 14) e possui 14 pinos de entrada/saída digitais (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas. Um cabeçalho de seis pinos pode ser conectado a um cabo FTDI para fornecer alimentação e comunicação via USB à placa.

A placa vem sem cabeçalhos pré-montados, permitindo o uso de vários tipos de conectores ou solda direta de fios. O layout do pino é compatível com o Arduino Mini. Existem duas versões do Pro Mini. Um executa em 3V3 e 8 MHz, o outro em 5V e 16 MHz.

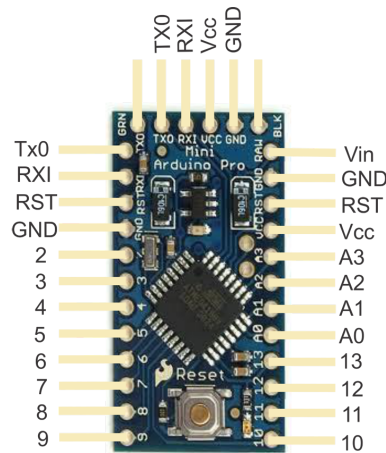


Figura 13 - Arduino pro mini e seus respectivos pinos.

O microcontrolador utilizado no arduino pro mini é o ATmega328 da Atmel, que utiliza a arquitetura de Harvard e é de 8 bits, RISC, com 32KB de memória flash, 1KB de EEPROM, 2KB de SRAM, 32 registradores de uso geral, portas para comunicação SPI entre outras características. O Arduino utiliza a interface SPI para comunicação com o módulo do nRF24L01+, logo no Arduino, tem que obrigatoriamente utilizar os pinos 11, 12, e 13 para os sinais MOSI, MISO e SCK e conectar com os respectivos pinos no nRF24L01+ como mostrado na Figura 13 (pinos 5, 6 e 7).

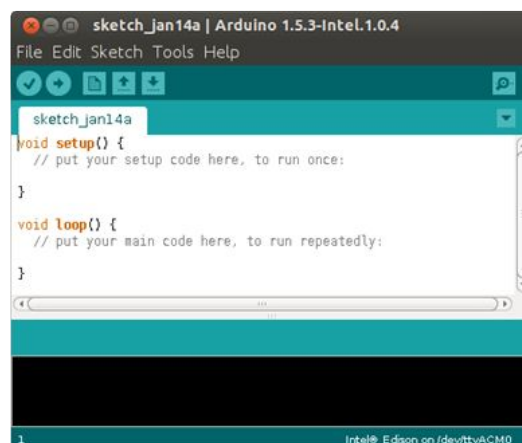


Figura 15 - Arduino IDE.

O software que controla os pinos e ações do Arduino pode ser desenvolvido no software chamado Arduino IDE (Figura 15), que conta com uma interface gráfica simples e intuitiva. A linguagem do arduino é uma DSL(domain specific language)

inspirado no C e C++. O código a ser embarcado no arduino deve estar no formato ".ino" e contar com as seguintes chamadas de procedimento:

- *setup*: Chamada quando o programa inicia. Ela é utilizada para iniciar variáveis, PinMode, e iniciar o uso de bibliotecas. Esta função é executada somente uma vez quando a placa é energizada ou cada vez que a placa é "resetada". Ajusta as configurações das portas de entrada e saída e a comunicação serial.
- *loop*: Faz exatamente o que seu nome sugere. É um Laço infinito onde o usuário cria a rotina do seu programa.

3. Compatibilidade nRF24L01+ e Bluetooth Low Energy

Ao analisarmos as Figuras 4 e 12 que representam os pacotes de dados do BLE e nRF24L01+ respectivamente, ambos possuem o formato de pacote similares, exceto que no nRF24L01+ possui um CRC de 16 bits e no BLE possui um CRC de 24 bits. O nRF24L01+ permite desativar o CRC, e então pode-se via software enviar o próprio CRC e verificar os CRCs recebidos. O BLE aplica o CRC em todo o payload, mas não ao endereço. O polinômio CRC tem a forma de $x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$. Para cada PDU de canal de advertising, o valor inicial deve ser predefinido com 0x555555. A posição 0 deve ser definida como o bit menos significativo e a posição 23 deve ser definida como o bit mais significativo do valor de inicialização. O CRC é transmitido primeiro o bit mais significativo, isto é, da posição 23 para a posição 0.

O BLE utiliza data whitening de dados para evitar sequências longas de zeros ou uns, ex. 0000000b ou 1111111b, no fluxo de bits de dados. O whitening deve ser aplicado no payload e é executado após o CRC (também é aplicado no CRC) no transmissor. O de-whitening é realizado antes do CRC no receptor como mostra a Figura 16. O whitener e o de-whitener são definidos da mesma maneira, usando polinômio da forma $x^7 + x^4 + 1$. Como o nRF24L01+ não aplica whitening, terá de ser implementado via software.

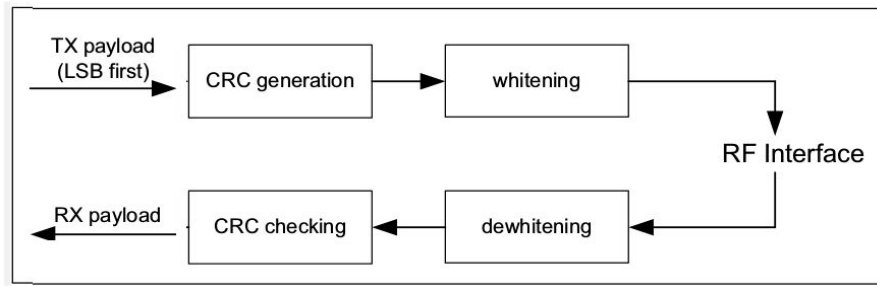


Figura 16 - Fluxo de empacotamento e desempacotamento do dado.

Os pacotes BLE podem ser de vários tamanhos, de até 39 bytes de payload. Para a PDU do tipo `CONNECT_REQ`, que possui 34 bytes de payload e 3 bytes de CRC, o que significa que para ser compatível com conexão, o nRF24L01+ teria que ser capaz de receber 37 bytes se quisermos verificar CRCs, e mesmo se escolhesse ignorar o CRC, seria preciso receber 34 bytes no total. O nRF24L01+ não é capaz de lidar com pacotes de mais de 32 bytes então o rádio não é capaz de aceitar conexões com outros dispositivos BLE. Para PDU do tipo `ADV_NONCONN_IND`, 1 byte seria reservado para o preâmbulo, 4 bytes para o endereço de acesso broadcast (sempre 0x8E89BED6), 3 bytes seria reservado para o CRC, 2 bytes para o header do pacote e 6 bytes para o endereço MAC, totalizando 16 bytes, e o restante dos bytes seria de dados puro a serem transmitido no payload, o que o torna compatível com o nRF24L01+ de 16 bytes de payload de dados puro, totalizando assim os 32 bytes de capacidade do nRF24L01+.

Além disso, para cada seção de pacote, o nRF24L01+ transmite primeiro o byte mais significativo e também transmite o bit mais significativo primeiro em cada byte. Isto é exatamente o oposto do BLE que transmite o byte menos significativo primeiro com o bit menos significativo primeiro em cada byte. Portanto é necessária a inversão de ordem de bit.

4. Metodologia.

4.1. Tecnologias e ferramentas utilizadas

Foram realizadas pesquisas para saber quais tecnologias seriam necessárias para o total desenvolvimento do trabalho. Essas tecnologias foram introduzidas na fundamentação teórica. As tecnologias utilizadas foram:

- Módulo nRF24L01+
- Arduino pro mini
- Beacon USB
- Linguagem C/C++
- Android

4.2. Desenvolvimento da biblioteca do nRF24L01+ para Arduino

A biblioteca open-source do nRF24L01+ para a plataforma Arduino utilizada neste trabalho foi desenvolvida no estágio curricular na empresa Centro de Estudos e Sistemas Avançados do Recife (CESAR) e está disponível em [17]. O motivo do desenvolvimento dessa biblioteca é que a maioria das bibliotecas do nRF24L01+ confiáveis já disponibilizadas estavam sobre a Licença Pública Geral (GPL - General Public Licence - em Inglês.). Estas restrições traduzem-se em certas responsabilidades para si, caso venha a distribuir cópias do software, ou modificá-lo. Por exemplo, se distribuir cópias de um programa sobre este tipo de licenciamento, gratuitamente ou por alguma quantia, tem que fornecer igualmente todos os direitos que possui sobre ele. Tem igualmente que garantir que os destinatários recebam ou possam obter o código-fonte. Exemplificando, se uma empresa utilizar uma biblioteca do nRF24L01+ que está sob a licença GPL para construção de um sistema, a empresa tem que obrigatoriamente disponibilizar os códigos utilizados. O código desenvolvido está sob a licença BSD (Berkeley Software Distribution). A licença BSD permite que o software distribuído sob a licença, seja incorporado a produtos proprietários. Trabalhos baseados no material podem até ser liberados

com licença proprietária. A biblioteca funciona pra linux e arduino (pro mini e UNO) e foi utilizado a linguagem C para o desenvolvimento.

Foi disponibilizada uma API para as funções do rádio. foram elas:

- `nrf24l01_init(void)`: Inicializar o rádio;
- `nrf24l01_deinit(void)`: Finalizar o rádio;
- `nrf24l01_set_channel(int channel)`: setar o rádio para o canal passado como parâmetro;
- `nrf24l01_open_pipe(pipe, addr, ack)`: Abre o *pipe* (canal de dados) 0 a 5, com o endereço *addr* habilitando ou não *acknowledgement*;
- `nrf24l01_close_pipe(pipe)`: Fecha o *pipe* passado como parâmetro;
- `nrf24l01_set_ptx(pipe)`: Coloca o *pipe* passado como parâmetro em modo de transmissão;
- `nrf24l01_ptx_data(buffer, len)`: Envia o *buffer* passado como parâmetro com o tamanho *len*;
- `nrf24l01_ptx_wait_datasent(void)`: Se o *ack* for ativado, essa função aguarda a chegada do *ack*;
- `nrf24l01_set_prx(void)`: Coloca o rádio em modo de recepção;
- `nrf24l01_prx_pipe_available(void)`: Verifica se tem dado em algum *pipe* e retorna qual o *pipe* está com dado a ser lido.
- `nrf24l01_prx_data(Buffer, len)`: Coloca em *Buffer* o dado disponível na RX FIFO com tamanho *len*.
- `nrf24l01_set_standby(void)`: Coloca o rádio em modo *standby-I*.
- `nrf24l01_set_power(power)`: Seta a potência de rádio de acordo com a tabela 3.

O rádio entra automaticamente no modo *standby-II* se está no modo de transmissão e não tem nada pra ser enviado.

O esquema de montagem do módulo nRF24L01+ com o Arduino pro mini pode ser visto na Figura 17.

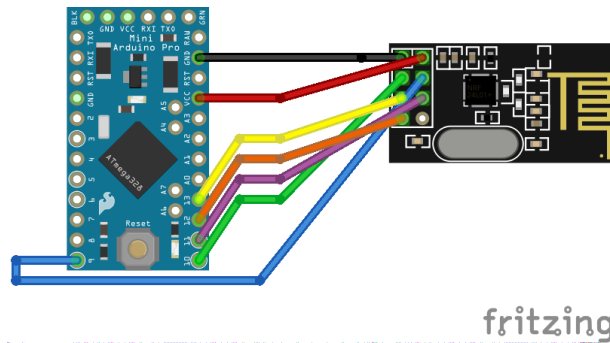


Figura 17 - Esquema de conexão utilizado no arduino pro mini e o nRF2401+.

Os pinos SCK, MOSI e MISO (pinos 5, 6 e 7 respectivamente) do nRF24L01+ foram conectados aos pinos SCK, MOSI e MISO (13, 12 e 11 respectivamente) do arduino. O GND e VCC (pinos 1 e 2) do nRF24L01+ foram conectados ao GND e VCC do arduino, embora o módulo opere a 3,3V, ele é tolerante ao nível lógico de 5V do Arduino pro mini e seus similares, sem precisar de conversão. Os pinos CE e CSN (pinos 3 e 4 respectivamente) do nRF24L01+ foram conectados aos pinos 10 e 9 respectivamente no arduino, deixando flexível ao usuário via código mudar os pinos CE e CSN no arduino. O pino de interrupção IRQ (pino 8) do nRF24L01+ não foi utilizado nesta implementação.

4.3. O módulo nRF24L01+ como beacon

No procedimento de setup no arduino, é chamado a função de inicializar o rádio. Nela é feito comandos para:

- SPI é inicializada para comunicação com o nRF24L01+.
- Desabilitar o CRC
- Desabilitar pacote de acknowledgement e desabilitar auto retransmissão (característica do rádio de retransmitir o pacote se o pacote ACK não foi recebido).
- Configurar o endereço para 4 bytes já que o endereço broadcasting é 0x8E89BED6
- Seta a taxa de transmissão do rádio para 1Mbps (mesmo que o BLE).
- Seta a potência de transmissão para 0dBm.

- Desabilita payloads dinâmicos e seta pra sempre receber pacotes de 32 bytes.
- Configura o endereço de transmissão para ser sempre 0x8E89BED6.

No procedimento de loop do arduino, onde é escrito a rotina do programa:

- Define o tipo de PDU para 00110b, ou seja, ADV_NONCONN_IND e *BR/EDR Not supported* para indicar que é beacon broadcasting não conectável e não suporta o bluetooth.
- Define o bit txAdd como 1 para indicar que o endereço do advertiser é aleatório.
- Define o length do payload.
- Define os próximos 6 bytes como o endereço do beacon.
- Define o beacon como *LE-General discoverable* mode e *BR/EDR Not supported*.
- Define o *local name* do dispositivo. O *local name* é um campo opcional.
- Define o id do fabricante e o dado a ser enviado.
- Seta o canal a ser enviado o dado (canais 36 37 e 38 do bluetooth, o que corresponde aos canais 2, 26 e 80 do nRF24L01+ e 2.402HGz, 2.426GHz, e 2.480GHz respectivamente na frequência).
- Gera o CRC do pacote baseado no payload invertendo os bytes.
- Aplica o data whitening ao pacote.
- Reverte a ordem dos bits.
- Transmite o pacote gerado.
- Espera em estado sleep durante o intervalo de broadcast definido pelo usuário.

O formato final do pacote pode ser visto na Figura 18, o local name utilizado foi “beaconRF” e pode ser mudado via código, bastando setar corretamente o seu tamanho em bytes. O dado a ser enviado pode ser desde dados estáticos definido pelo usuário até dados variáveis que estão sendo capturados via sensores. Para fins de prova de conceito, o byte “0x03” foi enviado como dado, foi utilizado 200ms

de intervalo *advertising*, endereço do beacon utilizado foi 0x11:22:33:44:55, o ID do fabricante como 0x0102.

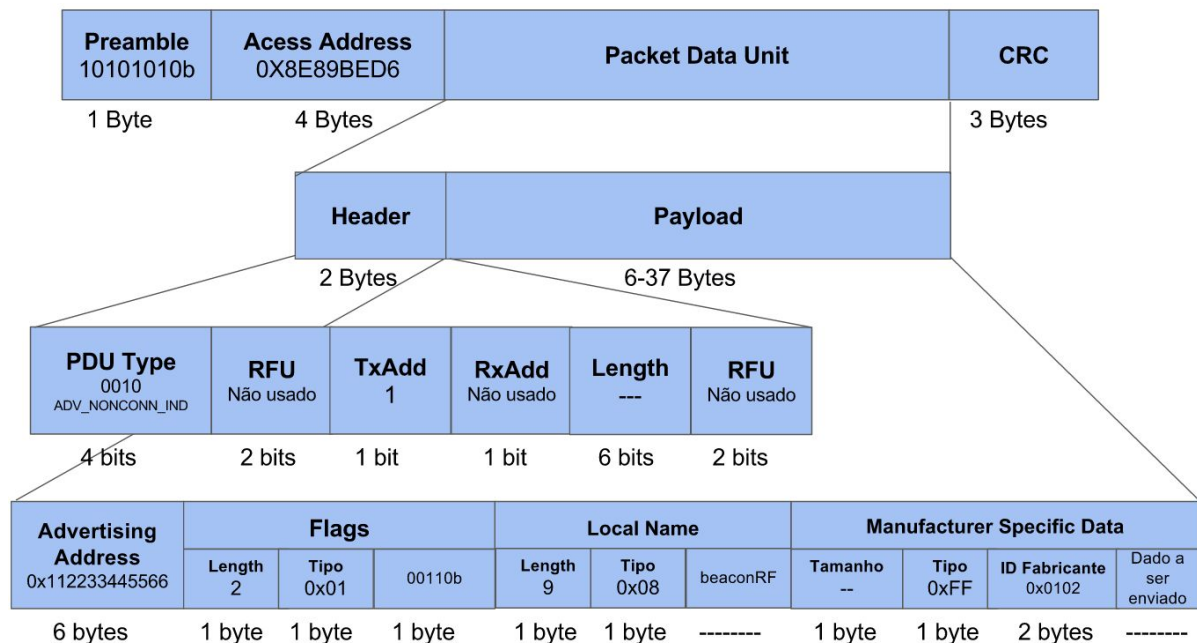


Figura 18 - Pacote do nRF24L01+ compatível com BLE broadcasting.

O campo Length, que indica o tamanho do payload do PDU em bytes, varia de acordo com o nome do dispositivo, se tiver, e com o tamanho do dado a ser enviado.

5. Experimentos e Resultados

5.1. Prova de conceito

Após a implementação do código, fez-se necessário a validação do funcionamento do nRF24L01+ se comportando como um dispositivo BLE. Para isto foi utilizado a ferramenta *nRF Connect for mobile* disponibilizada pela Nordic Semiconductors para Android e IOS. A ferramenta, anteriormente conhecido como *nRF Master Control Panel*, é uma ferramenta poderosa genérica que permite digitalizar, anunciar e explorar os seus dispositivos BLE e comunicar com eles [18]. Possui como principais características scans para dispositivos Bluetooth Low

Energy, analisar Dados *advertising*, mostrar o gráfico RSSI (indicador de intensidade de sinal recebido), conectar a um dispositivo Bluetooth Smart conectável e suporte BLE *advertising* (Android 5+ obrigatório).

O código descrito na seção 4.3 foi embarcado no arduino pro-mini e o mesmo foi conectado em uma fonte externa. A ferramenta foi instalada no smartphone Motorola G4 com versão de Android 6.0.1 e executada com a opção de bluetooth ativada. Ao selecionar a opção de scan, o aplicativo mostrará todos os dispositivos bluetooth que consegue detectar. De acordo com a Figura 19 pode-se verificar que o beaconRF foi detectado como um dispositivo BLE.

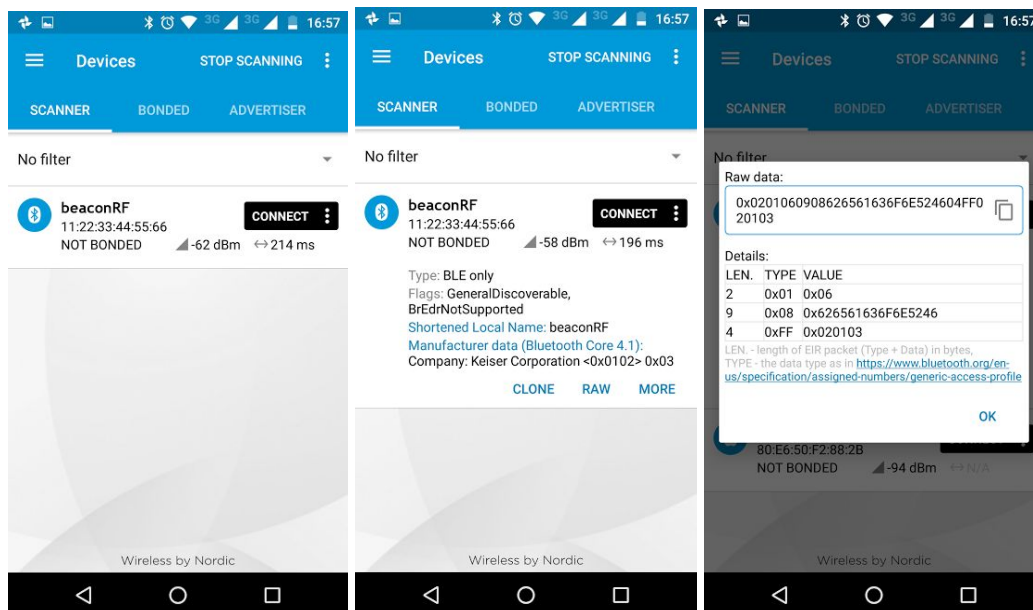


Figura 19 - Visualização do beacon nrf24l01+ no nrf connect master.

Pode-se verificar também que o “beaconRF” possui como tipo *BLE only*, flags *GeneralDiscoverable* e *BR/EDR not supported*, o ID do fabricante como 0x0102, o byte transmitido como 0x03 e o endereço como 11:22:33:44:55:6, assim como descrito no pacote da Figura 18 e mostra uma estimativa de intervalo *advertising* de aproximadamente 200ms. A ferramenta possui como opções, verificar mais detalhadamente as flags e serviços dos dispositivos BLE e mostrar o seu gráfico RSSI ao longo do tempo, como mostra a Figura 20.

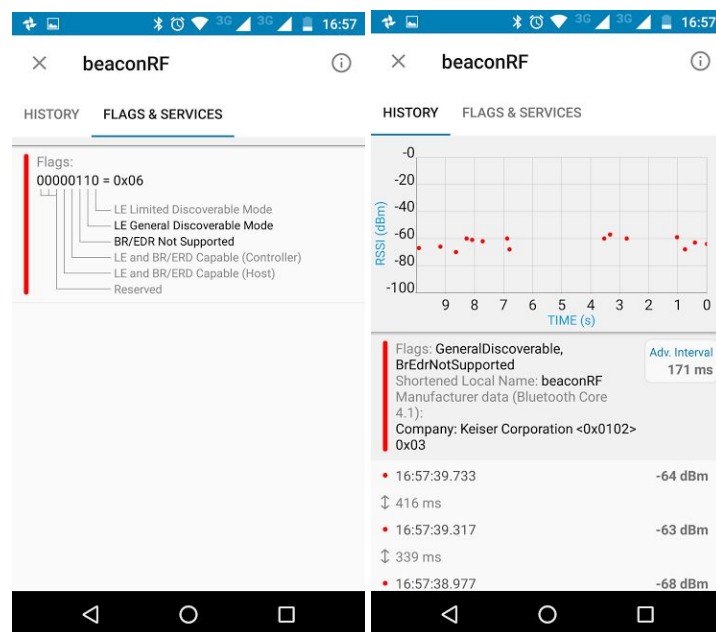


Figura 20 - Flags (direita) e gráfico RSSI (esquerda) do nrf24l01+ beacon.

5.2. Transmitindo dados de sensores

Após a prova de conceito do rádio se comportando como beacon, foi criado um caso de uso onde existe um sistema que utiliza o nRF24L01+ como módulo de comunicação e tornou seus pacotes compatíveis com BLE. O SECO¹ (Sistema de Escadourinhos Conectados) é um sistema que monitora o nível de fluidos e gases tóxicos nos bueiros da cidade através de sensores de distância e de gás. A ideia é que cada unidade seja posicionada debaixo da tampa do bueiro, os dados de cada unidade sejam propagados até o nó master que possui conexão com a internet e este envia os dados para o sistema web e os dados possam ser visualizados através desse sistema. Dois bytes de dados são enviados, o primeiro byte identifica o nível do bueiro (1-Baixo, 2-Médio e 3-Alto) e o segundo byte identifica o risco de explosão (1-Sem risco, 2-Médio risco, 3-Alto risco). Tornando o SECO compatível com BLE, iria permitir que os usuários que estivessem próximos aos bueiros e que tivessem a aplicação do sistema instalado pudessem verificar a atual situação do

1

<http://www2.recife.pe.gov.br/noticias/21/08/2016/pcr-anuncia-vencedores-do-desafio-hacker-cidadao-na-campus-party-recife>

mesmo. O esquema da conexão de sensores pode ser visto na Figura 21 e os pacotes transmitidos na Figura 22.

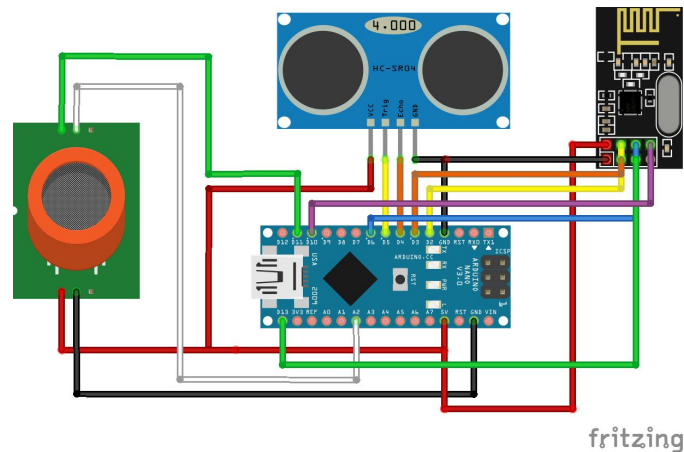


Figura 21 - Esquema de conexão dos sensores do sistema SECO.

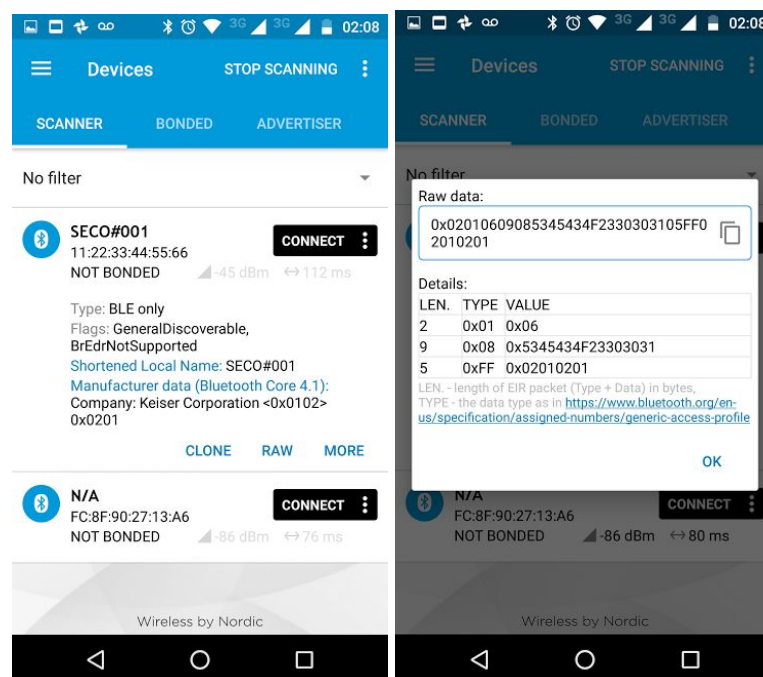


Figura 22 - Visualização do pacote do sistema SECO compatível com BLE.

Pela Figura pode-se observar o nome da unidade de sensor “SECO#001” e os bytes de dados transmitidos 0x02 e 0x01 indicando que o nível do bueiro está médio e não tem risco de explosão.

5.3. Teste de alcance

Foram realizados teste de alcance do beacon nRF24L01+ com ambos os rádios nRF24L01+ e nRF24L01+ com PA e LNA variando a potência do rádio configurada. Os testes foram realizados em um campo aberto, onde o beacon nRF24L01+ foi conectado em um carregador portátil USB e o receptor se afasta do rádio até começar a não receber mais pacotes. Os resultados para ambos os rádios podem ser visto na Figura 23.

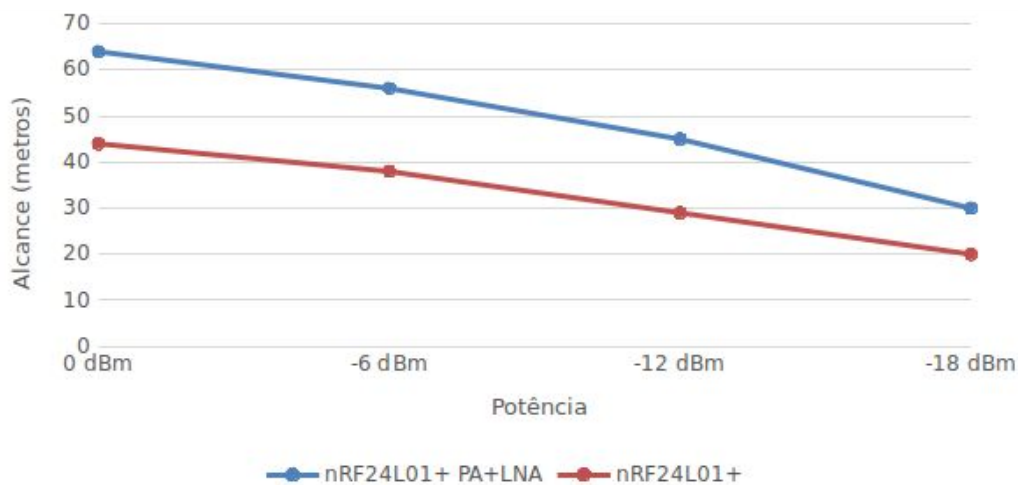


Figura 23 - Alcance do beacon nRF24L01+ e nRF24L01+ com PA e LNA variando a potência de transmissão.

Os resultados mostraram que o nRF24L01+ beacon apresentou bons resultados de alcance, chegando a até 43 metros para o nRF24L01+ e 64 metros para o nRF24L01+ com PA e LNA, valores estes compatíveis com os beacons padrões de mercado.

5.4. Teste de duração de bateria

Foi utilizado um carregador portátil usb de 2600mAh Tp Link fornecendo saída de até 5V e 2.4 amperes. Um cabo FTDI foi conectado ao arduino pro mini para fornecer alimentação via USB à placa. Para fins de teste, o intervalo

advertising foi configurado para 100ms para maior consumo. A ferramenta *nrf Connect* foi utilizada para saber o momento exato em que o último pacote foi capturado. O teste foi realizado seis vezes e os resultados podem ser visto na tabela 4 abaixo.

#Teste	Duração em horas
Teste #1	108,43
Teste #2	110,49
Teste #3	111,09
Teste #4	109,41
Teste #5	108,21
Teste #6	109,05

Tabela 4 - Duração da bateria em horas do beacon nRF24L01+.

Os resultados mostram que a abordagem do arduino pro mini e nRF24L01+ se comportando como beacon BLE não possui uma boa autonomia de bateria quando comparado a beacons BLE já existentes no mercado.

5.5. Custo

Todo o equipamento necessário para construir um nRF24L01+ como beacon BLE *broadcasting* é somente um arduino que custa em média U\$3,00 e um nRF24L01 que custa em média U\$1,00, tornando uma solução bem atrativa quando comparado com beacons disponíveis no mercado.

6. Considerações finais e trabalhos futuros

No trabalho, foi proposto uma implementação de uma biblioteca open-source para o rádio nRF24L01+, uma implementação para tornar o rádio compatível com o bluetooth low energy e disponibilizar uma API para que sistemas que já utilizem esse rádio possam tornar seus pacotes compatíveis com BLE. Esse estudo é de grande importância devido ao crescimento acelerado de dispositivos conectados a rede impulsionando cada vez mais a internet das coisas.

Inicialmente, foram mostrados conceitos fundamentais para o entendimento do trabalho, referentes ao bluetooth low energy e o nRF24L01+, detalhando toda sua estrutura de pacotes, modos de operação e seus requisitos. Logo depois foram apresentados as limitações do rádio e quais funcionalidades tem que suportar para que se torne compatível com BLE. Os resultados mostraram que é possível que o rádio se comporte como um beacon BLE broadcasting apresentando bons resultados de alcance, um consumo de energia moderado e com um custo bem abaixo quando comparado com beacons padrões do mercado. Foi apresentado um caso de uso onde tornou-se um sistema, que já utiliza o rádio como módulo de comunicação, compatível com BLE.

Para trabalhos futuros, pretende-se fazer uma análise comparativa com outros diversos tipos de beacons e realizar uma análise mais detalhada sobre o consumo de energia do beacon. Pretende-se também tornar a plataforma KNoT² (Knot Network of Things) desenvolvida no CESAR, que utiliza o nRF24L01+ para comunicação entre os devices, compatível com o Bluetooth Low Energy. O KNoT é uma Meta-Plataforma de código aberto para IoT que pretende desenvolver uma plataforma de Network of Things (KNoT) para promover a comunicação entre dispositivos e aplicações, bem como conectar padrões de comunicação, protocolos, bibliotecas e hardwares de referência.

² <http://knot.cesar.org.br/>

7. Referências Bibliográficas

- [1] Ashton, K. 2009. **That ‘internet of things’ thing**. RFID Journal, 22(7), 97-114.
- [2] Kortuem, G., Kawsar, F., Sundramoorthy, V., & Fitton, D. 2010. **Smart objects as building blocks for the internet of things**. IEEE Internet Computing, 14(1), 44-51.
- [3] Youngblood, G. M., Heierman, E. O., Holder, L. B., & Cook, D. J. 2005. **Automation intelligence for the smart environment**. In International Joint Conference On Artificial Intelligence (Vol. 19, p. 1513).
- [4] Mashal, I., Alsaryrah, O., Chung, T. Y., Yang, C. Z., Kuo, W. H., & Agrawal, D. P. 2015. **Choices for interaction with things on Internet and underlying issues**. Ad Hoc Networks, 28, 68-90.
- [5] **Beacon technology and the Internet of Things**. Disponível em: www.pegusapps.com/en/insights/beacon-technology-and-the-internet-of-things acessado em: 02 de dez de 2016.
- [6] Romet Kallas; **The Cost of a Beacon Project**. Disponível em: <https://unacast.com/post/the-cost-of-a-beacon-project> acessado em: 02 de dez de 2016
- [7] **Nordic Semiconductors nRF24L01+** Ultra low power 2.4GHz RF Transceiver. Disponível em: <http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01> Acessado em: 26 de nov de 2016.
- [8] **Specification of the Bluetooth System**, Covered Core Package, Version: 4.0; The Bluetooth Special Interest Group: Kirkland, WA, USA, 2010.
- [9] K. Townsend, C.Cufi, A&R. Davidson, “**Getting started with Bluetooth Low Energy**”, O'Reilly Media, Inc., 2014.
- [10] E. Mackensen, M. Lai, T. Wendt, “**Bluetooth low energy (BLE) based wireless sensors**”, in Sensors, 2012 IEEE, 2012, pp. 1–4.
- [11] C. Gomez, J. Oller, J. Paradells, “**Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology**”, in Sensors 2012, pp. 11734–11753.
- [12] <https://supportforums.cisco.com/pt/document/12885666> acessado em: 03 de dezembro 2016
- [13] Stian Sandve **Indoor Positioning System Using BLE Beacons**.

[14] **iBeacon**. Disponível em: <<http://developer.estimote.com/ibeacon/>> acessado em: 03 de dezembro 2016.

[15] **Eddystone**. Disponível em: <<http://developer.estimote.com/eddystone/>>

[16] **Arduino pro mini board Overview**. Disponível em: <<https://www.arduino.cc/en/Main/ArduinoBoardProMini>> acessado em: 27 de nov de 2016

[17] <https://github.com/CESARBR/knot-hal-source> acessado em: 02 de dezembro de 2016

[18] **nRF Master Control Panel-application**. Disponível em:

<<https://www.nordicsemi.com/eng/nordic/Products/nRF-Master-Control-Panel-application/nRF-Master-Control-Panel-APK-for-Android-4.3/36818/>> acessado em: 03 de dezembro 2016