



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica e Informática

Projeto e desenvolvimento de um protocolo de comunicação para dispositivo vestível utilizando Bluetooth Low-Energy

Melissa Maria Barbosa da Silva

Campina Grande, PB
Dezembro de 2018

Melissa Maria Barbosa da Silva

**Projeto e desenvolvimento de um protocolo de comunicação
para dispositivo vestível utilizando Bluetooth Low-Energy**

*Trabalho de Conclusão de Curso
apresentado à Coordenação do
Curso de Graduação em Engenharia
Elétrica da Universidade Federal de
Campina Grande, Campus Campina
Grande, como parte dos requisitos
necessários para a obtenção do grau
de Bacharel em Engenharia Elétrica.*

Área de Concentração: Controle e Automação

Orientador: Danilo Freire de Souza Santos

Campina Grande, PB
Dezembro de 2018

Melissa Maria Barbosa da Silva

Projeto e desenvolvimento de um protocolo de comunicação para dispositivo vestível utilizando Bluetooth Low-Energy

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovado em ___/___/___

Professor Avaliador

Universidade Federal de Campina Grande
Avaliador

Danilo Freire de Souza Santoss

Universidade Federal de Campina Grande
Orientador

Agradecimentos

Primeiramente agradeço a Deus por sempre ouvir meus pedidos e orações, me acolhendo e me fortalecendo em momentos difíceis e me fazendo sentir sempre tão amada. Aos meus pais, que sempre lutaram para que eu concluísse meus estudos; Eu nunca esquecerei todo o esforço e amor que vocês tiveram para que nada atrapalhasse minha trajetória, sou demasiadamente grata.

Ao meu namorado Matheus Leor que sempre me deu forças e motivos para sorrir diante das mais diversas situações. Agradeço também a sua família, que virou minha família ao me acolherem de uma maneira tão bondosa.

Ao meu professor orientador dos Estados Unidos, Max Mintz, que sempre me aconselhou mesmo quando retornei ao Brasil e sempre acreditou muito em mim, como pessoa e como engenheira. Agradeço também ao meu professor orientador de TCC, Danilo Santos, que é um grande exemplo de um bom professor e educador, sempre apto a escutar seus alunos e ajudá-los da melhor maneira possível.

Aos meus amigos de Elétrica que me ajudaram muito ao decorrer do curso. Em especial a Mylena Karla, Ulisses Gomes, Iara Martins, Margareth Mee, Victor Germano, Ítalo Bezerra, Paulo Roberto, Samuel Barros, Yago Monteiro, Nathalia Neves, Karen Aragão, Bruna Karen, Viviane Martins, Raíssa Dantas, Saul Medeiros, e tantos outros que me ajudaram nem que seja com uma simples dúvida. Se eu cheguei aqui hoje, foi com a ajuda de vocês.

Aos meus amigos que são meus melhores amigos há anos, Patrícia Noy, Bruno Brasil, Cícero Freire e Matheus Andrade. Minha amizade com vocês fortaleceu meu coração em momentos difíceis.

Aos amigos que eu fiz no intercâmbio e que nunca saíram da minha vida. Roberta Resende, Dominic Eaton e Beatriz Fagundes. Vocês foram minha família no intercâmbio e eu nunca esquecerei de toda a ajuda, seja sentimental ou com estudos.

”Há sempre um lugar no palco da vitória para os guerreiros que nunca desistem após uma queda.”

Resumo

Buscando uma solução para a diminuição dos efeitos do mal de Parkinson que esse projeto tomou forma. A inserção da nossa sociedade atual em um mundo completamente moderno e em constante desenvolvimento tecnológico, abre possibilidade e esperança de um futuro melhor, principalmente na área da saúde. O Bluetooth de baixo consumo de energia rouba a cena nesse novo cenário de Internet das Coisas, sendo um dos principais responsáveis pela conectividade e envio de informações entre dispositivos inteligentes. O principal objetivo desse trabalho se torna então a criação de um protocolo de comunicação Bluetooth para garantir a conexão entre diferentes sistemas.

Palavras chave: Doença de Parkinson, Bluetooth de baixo consumo de energia, Bluetooth Low Energy, Internet das Coisas.

Abstract

This project seeks to find a solution that mitigates the detrimental effects of Parkinson's disease. The advancement of our society into a modern world filled with many technological advancements opens up countless opportunities and hope, especially in the area of health. Bluetooth energy "steals the show" when it comes to this new scenario created by the internet of things, and this technology is key in providing a sufficient infrastructure for connectivity and information transfer among intelligent devices. The main goal of this project then becomes the creation of a Bluetooth communication protocol to guarantee the connection between different systems.

Keywords: Parkinson's disease, Internet of Things, Bluetooth Low Energy.

Lista de Figuras

2.1.1 Blocos básicos e fundamentais IoT	17
2.2.1 Configurações entre versões do bluetooth e tipos de dispositivos	18
2.3.1 Pilhas de protocolos BLE	19
2.3.2 Canais de frequência no BLE	20
2.3.3 Canais de frequência	20
2.3.4 Estados da Camada de Enlace	21
2.3.5 L2CAP blocos arquitetônicos	23
2.3.6 Aplicação de STK e LTK	24
2.3.7 Arquitetura Cliente-Servidor	25
2.3.8 Representação Lógica de um Atributo	26
2.3.9 Descrição do serviço de frequência cardíaca	27
2.3.10 Comunicação entre Servidor GATT e Cliente GATT	28
2.3.11 Hierarquia de dados do GATT	29
3.0.1 Doença de Parkinson	33
3.1.1 Emma Watch	35
3.1.2 Diagrama geral do projeto	35
3.2.1 Hierarquia de dados do GATT para o projeto	36
3.3.1 Pedido de leitura de dados do cliente	38
3.3.2 Pedido de escrita de dados do cliente	40
4.3.1 Inclusão do dispositivo PWatch e criação de um servidor	44
4.3.2 Exemplos de serviços GATT já efetivados	44
4.3.3 Criação de um serviço que já existe	45
4.3.4 Criação de um serviço novo	45
4.3.5 Definição do serviço de acordo com o UUID gerado	45
4.3.6 Criação das respectivas característica do serviço	46
4.3.7 Definição das características de acordo com os UUIDs gerados	46
4.4.1 Aplicativo LightBlue informando as características do dispositivo conectado	47
4.4.2 Leitura das informações fornecidas pelo servidor	48
4.4.3 Escrita da informação desejada pelo próprio cliente	49

Lista de Tabelas

3.1	Requisitos GATT	37
3.2	Dependências de transporte	37
4.1	Cenário Hardware dos principais componentes do projeto	42
4.2	Requisitos Funcionais	43
4.3	Requisitos Não Funcionais	43

Lista de Abreviaturas e Siglas

AFH	Adaptive Frequency Hopping
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
GAP	General Access Profile
GATT	Generic Attribute Profile
HCI	Host Controller Interface
IoT	Internet of Things
ISM	Industrial, Scientific, and Medical
L2CAP	Logical Link Control and Adaptation Protocol
LL	Link Layer
PHY	Physical Layer
RFID	Radio Frequency Identification
SMP	Security Manager Protocol
UUID	Universally Unique Identifier

Sumário

1	Introdução	12
1.1	Objetivos	14
2	Fundamentação Teórica	15
2.1	Internet das Coisas	16
2.2	Bluetooth Low Energy	17
2.3	Arquitetura do Bluetooth Low Energy	18
2.3.1	Camada Física - <i>Physical Layer (PHY)</i>	19
2.3.2	Camada de Enlace - <i>Link Layer (LL)</i>	21
2.3.3	<i>Host Controller Interface (HCI)</i>	22
2.3.4	<i>Logical Link Control and Adaptation Protocol (L2CAP)</i>	22
2.3.5	<i>Security Manager Protocol (SMP)</i>	23
2.3.6	<i>Attribute Protocol (ATT)</i>	24
2.3.7	<i>Generic Attribute Profile (GATT)</i>	27
2.3.8	<i>General Access Profile (GAP)</i>	29
3	Projeto do Protocolo de Comunicação	33
3.1	Mal de Parkinson	34
3.2	O GATT na especificação do projeto	36
3.3	Princípio de Funcionamento	38
4	Desenvolvimento do Serviço Bluetooth	41
4.1	Visão Geral do Sistema	41
4.2	Requisitos do sistema	42
4.3	Implementação	44
4.4	Validação	47
5	Conclusão	50

Capítulo 1

Introdução

Quando a Revolução Tecnológica progressivamente começou a fazer parte da realidade da população mundial, a noção de Internet das Coisas, ou *Internet of Things* (IoT) destacava-se por ser uma das maiores protagonistas para tamanha mudança em aspectos inovadores e modernistas. O modo como interagimos com o mundo foi claramente modificado, assim como o mundo passou a interagir conosco de uma forma incrivelmente diferente, resultando em um planeta mais inteligente e responsivo.

De uma forma mais clara, a Internet das Coisas é a maneira como os objetos físicos estão conectados e se comunicando entre si e com o usuário, através de sensores inteligentes e softwares que transmitem dados para uma rede. Pode-se até comparar a um grande sistema nervoso que proporciona a troca de informações entre dois ou mais pontos. Desde um relógio ou um frigobar, até carros, máquinas, computadores e smartphones. Qualquer instrumento que possa ser imaginado pode, teoricamente, entrar para o mundo da Internet das Coisas.

Segundo o analista e pesquisador Andy Mulholland[3], a melhor forma de identificar algo como um dispositivo de IoT é analisando a presença de quatro capacidades intrínsecas: conexão, inteligência, interatividade e autonomia. A Nike tem um exemplo bastante relevante sobre a Internet das Coisas. A pulseira inteligente Nike+FuelBand SE registra movimentos dos usuários para ajudá-los a entender melhor sua saúde. Um aplicativo de alta conectividade no smartphone armazena todos os dados registrados pela pulseira. Após isso, é possível receber inúmeras avaliações sobre as atividades físicas praticadas pelo usuário e seu respectivo desempenho.

Pelo exemplo citado anteriormente é notável que a Internet das Coisas não se limita somente a oferecer um maior conforto e praticidade para a sociedade. Essa tecnologia em constante desenvolvimento mostra como a área da saúde foi amplamente beneficiada com a inserção dessa nova tendência de tecnologia da informação. Médicos em todo o planeta já utilizam a Internet das Coisas como uma forma de transformar a maneira como pensamos em cuidados com a saúde. Aplicativos móveis podem se conectar a dispositivos diversos para obter e relacionar dados sobre o nosso corpo em busca de insights médicos.

O Dr. Aviv S. Gladman[4], chefe do Departamento de Informação Médica na Mackenzie Health, em Toronto (Canadá), um dos grandes líderes norte-americanos no que se refere à IoT, analisa até que “Talvez nenhuma indústria detenha tantas promessas para o uso da IoT quanto a saúde. Na verdade, alguns vêem como a força rompedora que irá revolucionar os tratamentos e melhorar a saúde do paciente no século XXI”.

O projeto Seeing AI¹, desenvolvido pela Microsoft, utiliza algoritmos de identificação visual e linguagem natural para descrever o ambiente em que uma pessoa está, ler textos e responder a perguntas. Os avanços são tão grandes que as últimas versões já são capazes de identificar emoções nos rostos de pessoas. Como pode-se observar, são inúmeros os projetos que objetivam a criação de novos tratamentos, sendo estes agora mais precisos e com uma maior riqueza de informações, baseando-se em uma realidade com informações obtidas em tempo real.

A doença de Parkinson, amplamente falada e sendo até mesmo retratada em filmes, é uma doença neurológica, crônica e progressiva. Pacientes que sofrem com essa condição perdem gradativamente suas funções motoras devido aos danos no sistema nervoso central, os quais causam rigidez e tremores, dentre outras consequências. O tratamento é feito com medicamentos, que destinam-se ao controle dos sintomas e possibilitam que os pacientes tenham uma maior qualidade de vida. Infelizmente, apesar da existência de um tratamento específico, não há cura. Diante disso, é de extrema importância acompanhar a resposta de pacientes ao tratamento.

Os portadores da referida enfermidade sabem o quanto que a doença, em estágios avançados, interfere na realização de atividades simples, como amarrar o cadarço do sapato ou abotoar a camisa. Além dessas interferências físicas, ocorrem também um elevado sofrimento emocional não apenas pela queda de sua funcionalidade, mas também em função do preconceito de pessoas desinformadas sobre a doença. Diante do exposto, não é inesperado que existam investigações e grandes projetos envolvendo a Internet das Coisas nesse campo em particular.

Um dos melhores exemplos de um grande projeto que foi desenvolvido para o auxílio a essa doença atribui-se ao relógio conhecido como Emma Watch ², uma tecnologia usável que “confunde” o cérebro e permite a escrita com redução de tremores. Desta forma, mesmo que a mão trema, a pessoa não o nota e pode continuar a realizar qualquer tarefa de forma cotidiana, como por exemplo escrever ou pintar.

A assimilação de como a conectividade entre equipamentos e usuários seria estabelecida é obtida pela compreensão de como as redes de comunicação atuam, além da escolha correta da categoria dessas redes diante de cada situação em particular. As redes de comunicação não fogem daquilo que já é comumente utilizado, como tecnologias Wi-Fi ou Bluetooth. Sendo a tecnologia Bluetooth Low Energy uma componente chave quando se pensa em IoT, projetado para baixo

¹<https://www.microsoft.com/en-us/seeing-ai>

²<https://www.microsoft.com/en-us/research/project/project-emma/>

consumo de energia, maior alcance e vários modos de operação possibilitando o desenvolvimento de aplicações nas mais diversas áreas.

1.1 Objetivos

O principal objetivo desse trabalho é a avaliação, especificação, implementação e validação de um protocolo de comunicação sem fio a fim de se estabelecer comunicação entre uma pulseira auxiliadora, a ser utilizada por portadores da doença de Parkinson para a execução de tarefas básicas, e um computador. A conexão será estabelecida através da rede sem fio Bluetooth de baixo consumo de energia, de modo que a troca de informações entre os dispositivos será realizada por intermédio dessa conectividade.

Capítulo 2

Fundamentação Teórica

Em comparação a outros padrões sem fio, o rápido crescimento do Bluetooth Low Energy (ou Bluetooth 4.0) é relativamente fácil de explicar já que seu uso está tão intimamente ligado ao crescimento fenomenal de smartphones, tablets e computação móvel. A adoção precoce e ativa desse novo tipo de Bluetooth por grandes indústrias de telefonia móvel, como a Apple e a Samsung, abriu as portas para uma implementação mais ampla dessa nova geração Bluetooth. Portanto, esse capítulo apresenta como objetivo o fornecimento de uma compreensão sólida, prática e de alto nível de cada uma das camadas do Bluetooth Low Energy, para qualquer pessoa que demonstra interesse pelo assunto e deseja adquirir mais conhecimentos técnicos sobre o mesmo.

Inicialmente, o assunto de Internet das Coisas será abordado novamente, mas dessa vez para explicar como o termo surgiu e o porquê que a sua evolução foi tão abrupta. É importante entender essa base teórica para a percepção de onde o projeto se enquadra tecnologicamente. É compreensível também as dúvidas persistentes sobre o modo de conectividade. A maioria das pessoas hoje em dia se sentem mais confortáveis ao ouvir sobre Wi-Fi do que sobre o Bluetooth em si. Então a explicação do porquê dessa escolha específica de usabilidade é importante para um prévio discernimento sobre o assunto.

Cada camada que faz parte da pilha de Protocolos do Bluetooth Low Energy será devidamente tratada, com detalhes na sua composição, função e importância na arquitetura do Bluetooth. Como cada camada interage entre si e com outras camadas é de extrema relevância para entender como o sistema todo funciona em conjunto. Entender como os dados são organizados e transferidos por dispositivos Bluetooth de baixo consumo de energia e saber como os dispositivos com essa conectividade inteligente descobrem uns aos outros e estabelecem conexões seguras são as finalidades principais e esperadas ao decorrer da leitura dessa seção.

2.1 Internet das Coisas

O entendimento dos conceitos que regem esse projeto é importante, assim, pesquisas foram intensamente realizadas desde as mais básicas concepções de estudo. Toda essa busca para uma melhor assimilação do conteúdo, mostrou-se um grande instrumento na construção do conhecimento e na formação de um melhor senso crítico.

Em um mundo cada vez mais conectado, o desenvolvimento de ambientes virtuais tem crescido consideravelmente. Há cada vez mais uma demanda significativa de usuários procurando serviços modernos que interajam com o mundo cibernético. O cenário em que estamos inseridos atualmente mostra a relevância impactante da “Internet das coisas”, termo que surgiu em 1999 por Kevin Ashton, a fim de descrever um sistema em que a internet está conectada ao mundo físico por meio de sensores onipresentes.

Apesar do termo ter surgido nos anos 90, a Internet das Coisas ainda era uma ideia futurista. Foi entre 2008 e 2009 que essa tecnologia começou a crescer de forma contínua e crescente. Foi nessa época, através de um artigo publicado no RFID Journal, que Kevin Ashton[5] cita o que é tido por muitos como a definição da IoT: “Se tivéssemos computadores que soubessem tudo o que há para saber sobre coisas usando dados que foram colhidos, sem qualquer interação humana, seríamos capazes de monitorar e mensurar tudo, reduzindo o desperdício, as perdas e o custo. Gostaríamos de saber quando as coisas precisarão de substituição, reparação ou atualização, e se eles estão na vanguarda ou se tornaram obsoletas”,

É inquestionável o tremendo impacto dessa tecnologia na sociedade atual. Progressivamente, modificamos nossa maneira de se comunicar, realizar atividades e até mesmo consumir produtos ou adquirir serviços. De certo modo é como se não existisse mais limites sociais ou geográficos. Em residências, já é notável o desenvolvimento da IoT através do controle da iluminação, de câmeras e de alarmes de segurança, além disso até um simples termostato aprende sobre preferências e ajusta automaticamente a temperatura, tornando a vida bem fácil. No ramo da saúde já pode-se encontrar dispositivos que monitoram a frequência cardíaca, ingestão calórica, sono, caminhada, e muitas outras métricas, tornando a vida dos usuários mais saudável.

A Internet das coisas pode ser vista como a associação de inúmeras tecnologias, as quais são complementares no sentido de propiciar a integração dos objetos no ambiente físico ao mundo virtual, o diagrama da Figura 2.1.1 oferece essa noção de integração de sistemas. Uma das concepções mais importantes dessa associação refere-se a comunicação, que, por sua vez, corresponde as diversas técnicas utilizadas para conectar objetos inteligentes além de desempenhar um papel importante no consumo de energia. Algumas das tecnologias empregadas são Wi-Fi, Bluetooth, IEEE 802.15.4 e RFID.

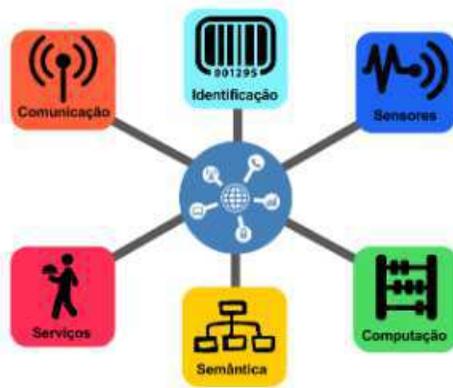


Figura 2.1.1: Blocos básicos e fundamentais IoT

2.2 Bluetooth Low Energy

A tecnologia de curto alcance Bluetooth foi desenvolvida em 1994 por dois engenheiros, Sven Matisson e Jaap Haartsen[6]. Na época ambos trabalhavam na empresa de telecomunicação sueca Ericsson e a idéia original era criar uma tecnologia que substituísse os diversos cabos de conexão por um único padrão wireless. Na prática, o Bluetooth é uma forma de conectar e trocar informações entre dispositivos como celulares, computadores, tablets e câmeras digitais através de uma frequência de rádio de curto alcance globalmente licenciada e segura. As especificações do Bluetooth foram desenvolvidas e licenciadas pelo “Bluetooth Special Interest Group”¹, que é uma organização formada por milhares de empresas de tecnologia ao redor do mundo.

Atualmente, pode ser visto que Bluetooth se divide em dois grupos:

- Bluetooth clássico: Que por sua vez se divide em Basic Rate/Enhanced Data Rate (BR/EDR), que são as versões 2.0 ou anteriores e o Bluetooth High Speed (HS), versão 3.0;
- Bluetooth Low Energy: Versão 4.0 ou superior.

O Bluetooth de baixo consumo de energia (BLE), também conhecido como Bluetooth inteligente, utiliza uma quantidade menor de energia ao ser comparado com as comunicações sem fio Bluetooth clássico e Wi-Fi, por exemplo. Essa eficiência energética tornou o BLE uma das melhores e mais compatíveis opções para IoT, significando a conectividade de dispositivos IoT por períodos mais prolongados, especialmente quando os dispositivos em questão são movidos a bateria. Além disso, a baixa taxa de dados do BLE o torna bastante adequado para utilização em situações em que somente informações de estado precisam ser alteradas.

¹<https://www.bluetooth.com/>

Dado sua característica principal ser a economia de energia, um dispositivo BLE permanece em modo *sleep* durante a maior parte do tempo, saindo desse estado somente para realizar conexões que duram milissegundos. Isso ocorre pois o mesmo foi projetado para aplicações que precisam enviar meramente poucas e esporádicas informações.

Não obstante, o BLE efetivamente impõe 2 tipos de dispositivos: O *single mode* ou *smart* que suporta apenas o protocolo BLE, ou seja, o mesmo realiza conexão apenas com um dispositivo semelhante; E o *dual-mode* ou *smart-ready*, que, através da execução de um protocolo que faz a mediação entre o Bluetooth clássico BR/EDR e o BLE, é possível implementar ambas as arquiteturas. Porém, essa implementação não costuma ter o mesmo ganho energético que um dispositivo single mode. Na figura 2.2.1 pode-se observar e comparar a composição estrutural dos tipos de Bluetooth citados.

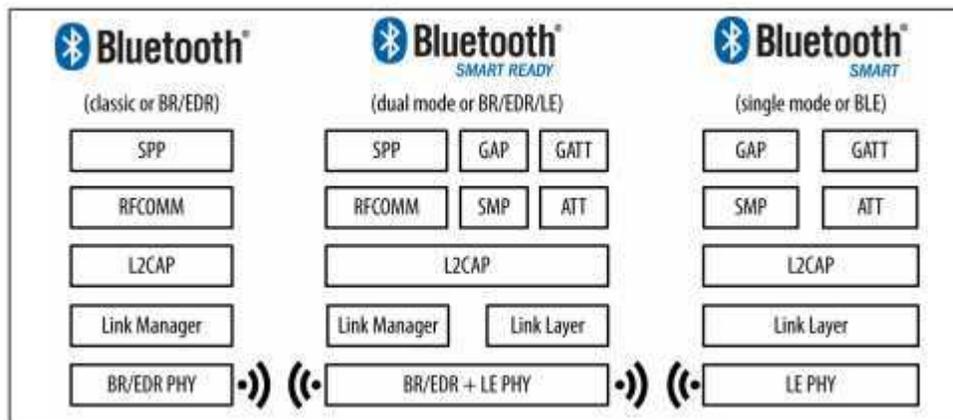


Figura 2.2.1: Configurações entre versões do bluetooth e tipos de dispositivos

2.3 Arquitetura do Bluetooth Low Energy

A arquitetura do BLE é consideravelmente similar a do Bluetooth Clássico. Em linhas gerais, é estruturada em três partes: Controlador, Host e Aplicações. O Controlador é tipicamente o dispositivo físico que transmite e recebe os sinais rádio e entende como é que esses sinais podem ser interpretados como pacotes com informação. O Host é uma camada de software que administra a forma como dois ou mais dispositivos se comunicam com outro e quantos serviços diferentes podem ser disponibilizados pelo rádio. E por fim, Aplicações utilizam o Host e, conseqüentemente, o Controlador para implementar um estudo de caso. A Figura 2.3.1 demonstra detalhadamente essa divisão de camadas.

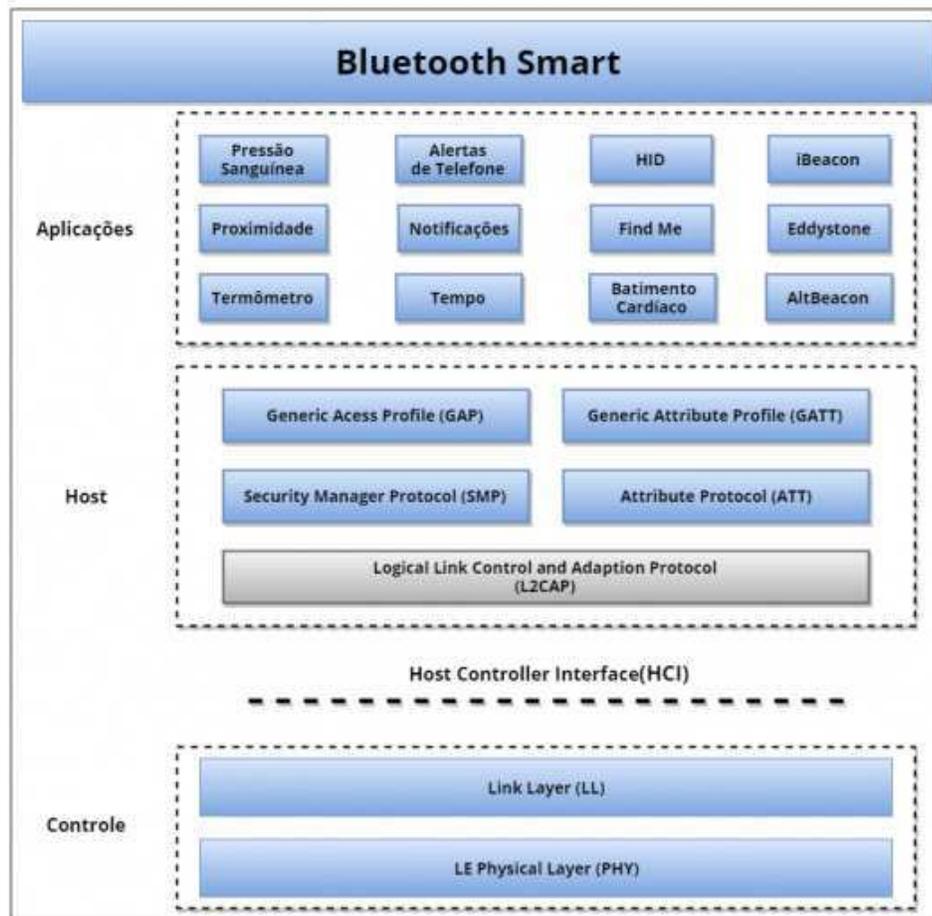


Figura 2.3.1: Pilhas de protocolos BLE

Embora os usuários geralmente façam interface apenas diretamente com as camadas superiores da Pilha de protocolos BLE, uma visão geral básica da pilha completa fornece uma base sólida para entender como e por que as coisas funcionam do jeito que funcionam.

2.3.1 Camada Física - *Physical Layer (PHY)*

Contém o circuito de comunicações analógicas, capaz de modular e desmodular sinais analógicos e transformá-los em símbolos digitais. Como pode ser visto na Figura 2.3.2 refere-se a camada mais baixa da pilha de protocolos, fornecendo seus serviços para a camada *Link Layer*. O rádio usa a banda ISM para se comunicar e divide essa banda em 40 canais, de 2,4000 GHz a 2,4835 GHz, cada um com espaçamento de 2MHz. Esses canais são divididos em duas categorias: *advertising channels* e *data channels*.

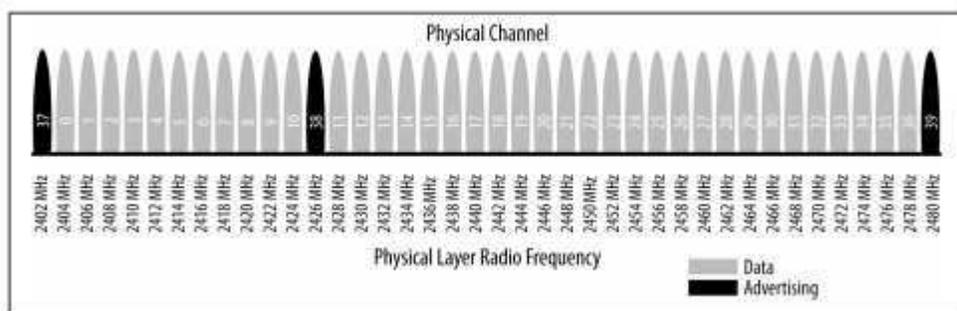


Figura 2.3.2: Canais de frequência no BLE

Os *advertising channels* ou canais de anúncio ocupam 3 dos 40 canais citados anteriormente e são utilizados para descoberta de dispositivos, estabelecimento de conexão e transmissões broadcast. Quando um dispositivo periférico BLE está no modo *advertising*, os pacotes de anúncios são enviados periodicamente em cada *advertising channel*. Já os *data channels* ou canais de dados, ocupam os canais restantes e são utilizados para a comunicação bi-direcional entre os dispositivos conectados além de fazer uso de uma técnica chamada *Adaptive Frequency Hopping*. Essa técnica faz com que o mapa de frequências disponíveis seja readaptado, de modo que os canais já ocupados sejam excluídos da lista de disponibilidade. Além disso, o AFH faz com que mestre e escravo utilizem o mesmo canal, a fim de impedir que um use um canal bom e o outro um canal ruim, o que poderia provocar uma série de erros e retransmissões.

Esse processo é importante pois, por um usar o espectro gratuito de 2.4 GHz, o BLE corre o risco de colidir com um sinal de Wi-Fi/802.11 que esteja operante na região, como podemos observar na Figura 2.3.3.

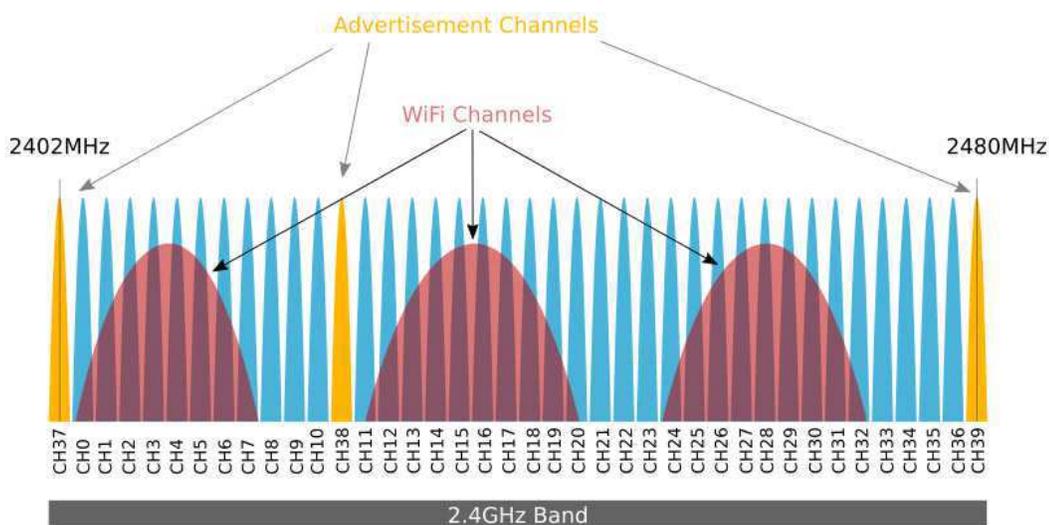


Figura 2.3.3: Canais de frequência

2.3.2 Camada de Enlace - *Link Layer (LL)*

Seção que interage diretamente com a camada física e geralmente é implementada como uma combinação de hardware e software personalizados. O software implementado nesta camada gerencia o estado do enlace do rádio, que é a forma como o dispositivo conecta-se a outros dispositivos. Um dispositivo BLE pode ser um mestre, um escravo, ou ambos, dependendo do caso e dos requisitos de utilização. Dispositivos que iniciam conexões são denominados mestres (*initiator*) e dispositivos que anunciam a sua disponibilidade e aceitam ligações são denominados escravos (*advertiser*).

Para a conexão entre dois dispositivos ser concretizada, estes podem assumir cinco estados dentro da camada de enlace, que podem ser visualizados na Figura 2.3.4 inserida logo abaixo.

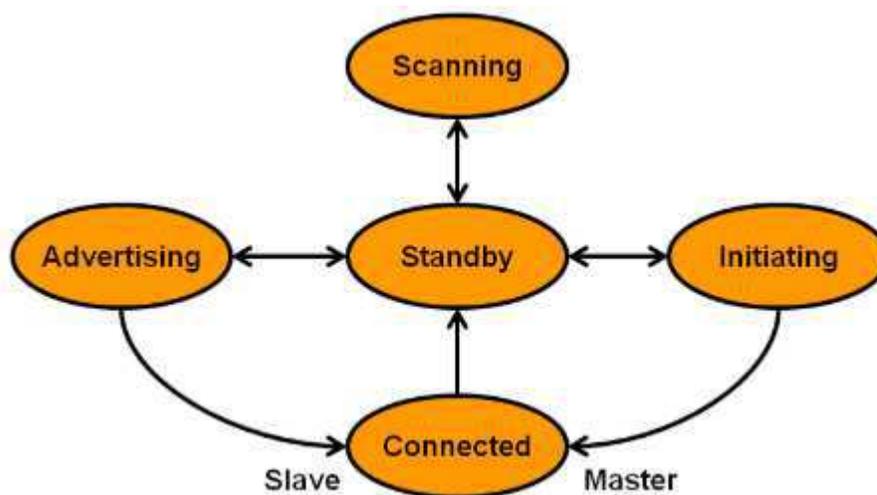


Figura 2.3.4: Estados da Camada de Enlace

Sendo,

- *Scanning*: O dispositivo nesse estado, chamado de scanner, escuta transmissões de pacotes de anúncios e pode solicitar informações adicionais a outros dispositivos;
- *Standby*: O dispositivo não recebe nem envia nenhum pacote; o estado pode ser acessado a partir de qualquer outro;
- *Advertising*: O dispositivo nesse estado, chamado de advertiser (anunciador), envia pacotes por cada advertising channel e pode responder a requisições de outros dispositivos;
- *Initiating*: O dispositivo nesse estado, chamado de initiator, diferentemente do scanner, tem a intenção de se conectar a um anunciador. Ao localizar um, ele responde requisitando o início de uma conexão;

- *Connection*: quando a conexão é formada entre o initiator e o advertiser, ambos entram nesse estado e se comunicam através dos data channels.

Além disso, a camada de enlace define os seguintes papéis:

- *Advertiser*: Dispositivo que envia pacotes de anúncio;
- *Scanner*: Dispositivo que explora os pacotes de anúncio;
- *Master*: Dispositivo que inicia uma conexão e a gerencia mais tarde;
- *Slave*: Dispositivo que aceita uma solicitação de conexão e segue o tempo do master.

Essas funções podem ser logicamente divididas em dois grupos: *Advertiser* e *Scanner* quando o dispositivo não se encontra em uma conexão; *Master* e *Slave* quando o dispositivo se encontra em uma conexão.

2.3.3 Host Controller Interface (HCI)

Refere-se a um protocolo padrão que realiza a comunicação entre o host e o controlador. Conforme as especificações do Bluetooth, é definido como um conjunto de comandos e eventos que fazem com que o host e o controlador interajam entre si, juntamente com o formato de pacotes de dados e um conjunto de regras para o controle de fluxo e outros procedimentos. Existem várias camadas de transporte padrão no HCI, cada uma usa uma interface de hardware diferente para transferir o mesmo comando, evento e pacote de dados. Os mais comuns são USB/UART para PCs e SPI para microcontroladores e smartphones.

2.3.4 Logical Link Control and Adaptation Protocol (L2CAP)

O protocolo de adaptação e controle de enlace lógico (Logical Link Control and Adaptation Protocol) transfere dados entre as camadas superiores do host e a pilha de protocolos da camada inferior. Essa camada é responsável pela: Multiplexação entre diferentes protocolos da camada superior, permitindo que eles compartilhem links da camada inferior; Segmentação e remontagem para permitir a transferência de pacotes maiores do que o que as camadas inferiores suportam; Gerenciamento de grupo, fornecendo transmissão unidirecional para um grupo de outros dispositivos Bluetooth e Gerenciamento de qualidade de serviço para protocolos da camada superior.

Para Bluetooth Low Energy, a camada L2CAP está roteando dois protocolos principais: o Protocolo de Atributos (ATT) e o Security Manager Protocol (SMP). O ATT forma a base da troca de dados em aplicativos BLE, enquanto o SMP fornece uma estrutura para gerar e distribuir chaves de segurança entre pares de dispositivos. Uma melhor visualização sobre sua estrutura é mostrada na figura 2.3.5.

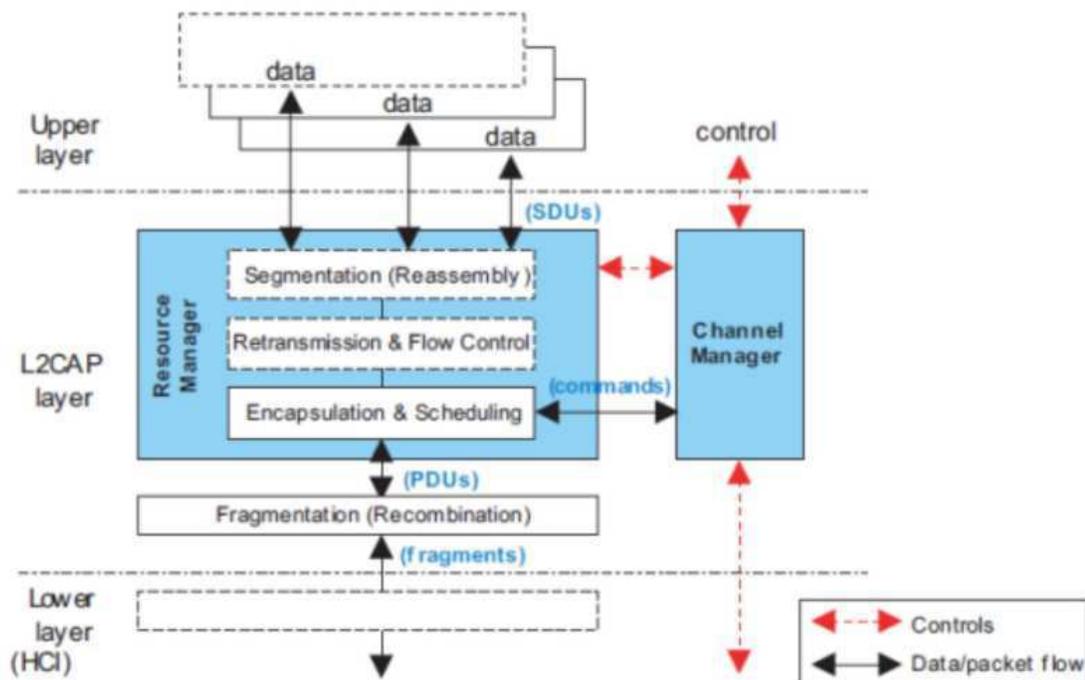


Figura 2.3.5: L2CAP blocos arquitetônicos

2.3.5 Security Manager Protocol (SMP)

É um protocolo e uma série de algoritmos de segurança projetados para fornecer à pilha de protocolos Bluetooth a capacidade de gerar e trocar chaves de segurança, permitindo que os pares de dispositivos se comuniquem com segurança por um enlace criptografado, a Figura 2.3.6 aborda essa definição de uma maneira lógica e de fácil visualização.

O SMP realiza o emparelhamento em três fases.

1. Os dois dispositivos BLE conectados anunciam suas capacidades de entrada e saída e, a partir dessa informação, determinam um método adequado para a fase 2.

2. O objetivo dessa fase é gerar a chave de curto prazo (STK) usada na terceira fase para garantir a distribuição das chaves. Os dispositivos concordam com uma Chave Temporária (TK) que junto com alguns números aleatórios cria o STK.

3. Nesta fase, cada dispositivo pode distribuir para o outro dispositivo até três chaves:

- Chave de longo prazo (LTK) usada para criptografia e autenticação da camada de enlace;
- Chave de resolução de assinatura de conexão (CSRK) usada para assinatura de dados na camada ATT;
- Chave de resolução de identidade (IRK, Identity Resolving Key) usada para gerar um endereço particular.

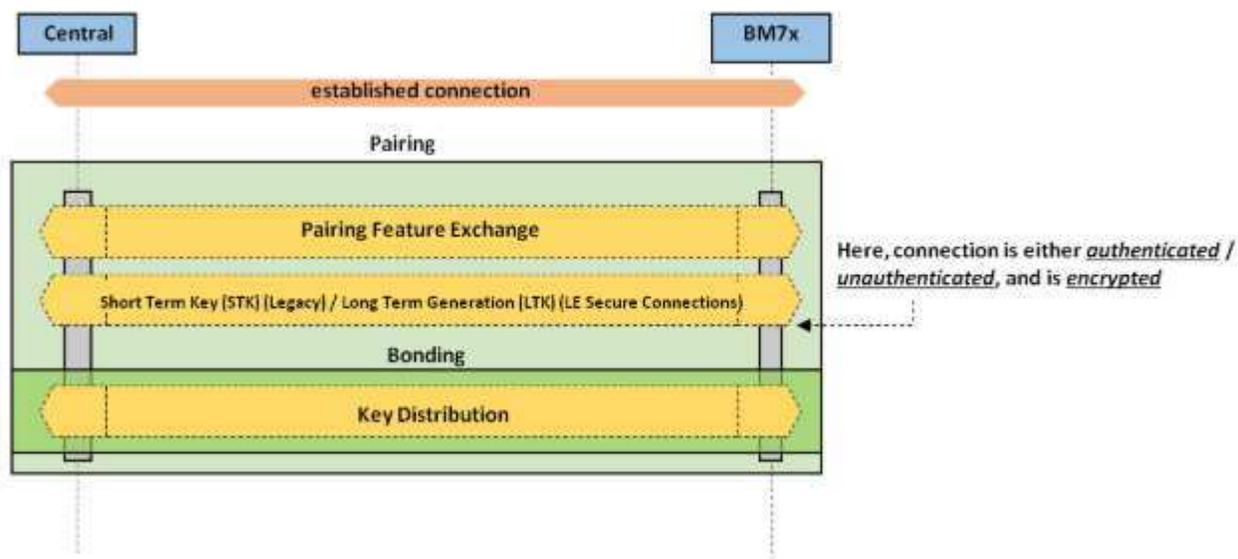


Figura 2.3.6: Aplicação de STK e LTK

2.3.6 Attribute Protocol (ATT)

O Protocolo de Atributos é um protocolo cliente/servidor sem estado, baseado nos atributos apresentados por um dispositivo. Os atributos são estruturas de dados que armazenam informações fornecidas pela camada GATT, que será abordada logo mais.

O Dispositivo que expõe os seus atributos é o servidor e o dispositivo vizinho é o cliente. Os papéis de mestre/escravo concedidos na camada de enlace são independentes dos papéis de cliente/servidor definidos neste protocolo. Por exemplo, um dispositivo mestre tanto pode ser um cliente como pode ser um servidor, e o mesmo acontece com o dispositivo que atribui-se como escravo. Torna-se possível também um dispositivo ser cliente e servidor simultaneamente. O cliente pode aceder aos atributos do servidor através do envio de pedidos, e o servidor, de modo a garantir eficiência, pode enviar ao cliente dois tipos de mensagens não solicitadas contendo os atributos, que são: notificações, onde o servidor envia dados ao cliente e este não confirma a sua receção; ou indicações, que requisitam uma confirmação por parte do cliente.

Em simples palavras, os servidores contêm dados organizados sob a forma de atributos e os clientes desejam utilizar essas informações, como pode ser facilmente entendido através da Figura 2.3.7.

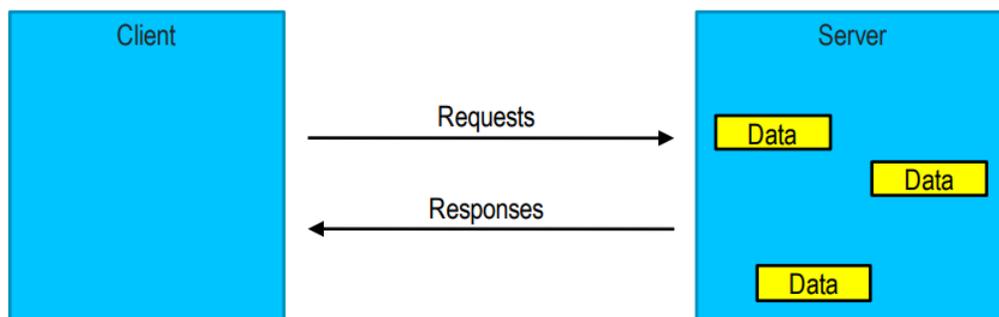


Figura 2.3.7: Arquitetura Cliente-Servidor

Um atributo é composto por 4 elementos:

- *Attribute Handles*

O handle do atributo refere-se a um número de 16 bits que muda conforme o número de atributos presentes e realiza a identificação dos mesmos no servidor, permitindo que o cliente faça referência à ele em requisições de leitura e escrita. Este valor nem sempre é sequencial e é muito utilizado pelo *softdevice* para fazer referência à vários atributos. Na concepção do programador é um modo eficaz de passar valores e informações entre funções além facilitar o controle da aplicação sobre o atributo e obter qualquer informação necessária.

- *Attribute Types (UUIDs)*

O Identificador Único Universal é um valor de 16, 32 ou 128 bits utilizado para identificar o tipo de cada atributo. Um UUID completo possuirá 128 bits (16 bytes), porém na procura de uma maior eficiência e um baixo consumo dos valiosos 27 bytes de payload da camada de enlace, as especificações do BLE introduziram dois UUIDs adicionais, de 16 e 32 bits. Estes formatos menores podem ser utilizados somente com UUIDs que são determinados pelas especificações do Bluetooth, ou seja, só é compatível com tecnologia Bluetooth.

É necessário compreender que o próprio ATT não define nenhum UUID. Isso é deixado para o GATT e perfis de nível superior.

- *Attribute Permissions*

As permissões que os atributos apresentam definem as regras de interação com cada um deles, ou seja, quando um atributo pode ser lido e/ou escrito e qual tipo de autorização é necessária para fazer estas operações. É importante observar que essas permissões se aplicam apenas ao valor do atributo, não necessariamente ao identificador, ao tipo e ao próprio campo de permissão. Permitindo que um cliente examine a tabela de atributos de um servidor e

encontre o que o servidor pode fornecer. Apesar de não necessariamente ler e escrever os valores.

- *Attribute Values*

O Valor do Atributo é a informação em si e pode ser qualquer tipo de dado, como o valor de frequência cardíaca dado em batimentos por minuto, o estado de um interruptor de luz ou uma string como “Hello World”. Ocasionalmente, também pode referir-se a uma informação sobre onde encontrar outros atributos e as propriedades dos mesmos.

A Figura 2.3.8 demonstra a representação de um atributo a partir dos 4 elementos previamente citados.

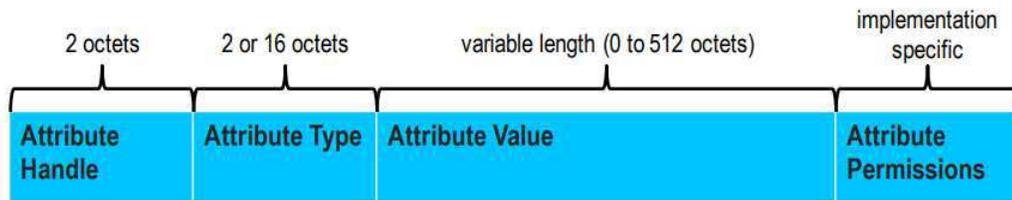


Figura 2.3.8: Representação Lógica de um Atributo

A partir de uma aplicação é que pode-se observar e assim entender melhor como cada um dos elementos citados anteriormente colaboram na formação de um atributo. Para isso, o perfil da frequência cardíaca foi devidamente analisado para um cuidadoso estudo de caso, como é mostrado na Figura 2.3.9.

Heart Rate Service	Handle	Type of attribute (UUID)	Attribute permission	Attribute value
Service Declaration	0x000C	Service declaration Standard UUID _{Service} 0x2800	Read Only, No Authentication, No Authorization	Heart Rate Service 0x180D
Characteristic Declaration	0x000D	Characteristic declaration Standard UUID _{Characteristic} 0x2803	Read Only, No Authentication, No Authorization	Properties (Notify), Value Handle (0x000E), UUID for Heart Rate Measurement characteristic (0x2A37)
Characteristic Value Declaration	0x000E	Heart Rate Measurement Characteristic UUID found in the Characteristic declaration value 0x2A37	Higher layer profile or implementation specific	Beats Per Minute E.g. "167"
Descriptor Declaration	0x000F	Client Characteristic Configuration Descriptor (CCCD) Standard UUID _{CCCD} 0x2902	Readable with no authentication or authorization. Writable with authentication and authorization defined by a higher layer specification or is implementation specific.	Notification enabled 0x000X
Characteristic Declaration	0x0010	Characteristic declaration Standard UUID _{Characteristic} 0x2803	Read Only, No Authentication, No Authorization	Properties (READ), Value Handle (0x0011), UUID for Body Sensor Location (0x2A38)
Characteristic Value Declaration	0x0011	Body Sensor Location UUID found in the Characteristic declaration value 0x2A38	Higher layer profile or implementation specific	Sensor Location (8-bit integer) E.g. 3 equals "Finger"

Figura 2.3.9: Descrição do serviço de frequência cardíaca

2.3.7 Generic Attribute Profile (GATT)

Remete-se a um perfil de base para todos os perfis LE de nível superior, definindo como vários atributos ATT são agrupados em uma ordem lógica e específica. O perfil da frequência cardíaca visto anteriormente é um exemplo de um grupo. Construído sobre o Attribute Protocol (ATT), é o modo que cliente e servidor compartilham suas informações de modo estruturado, trocando características, ou seja, o GATT basicamente define o formato dos dados expostos por um dispositivo BLE, determinando também os procedimentos necessários para acessar os dados expostos por um dispositivo.

O GATT define papéis de interatividade que os dispositivos podem adotar: O Servidor GATT, que corresponde ao servidor ATT, que é o dispositivo que expõe os dados que ele controla ou contém e, possivelmente, alguns outros aspectos de seu comportamento que outros dispositivos podem controlar; e o Cliente GATT, que corresponde ao cliente ATT, que por outro lado, é o dispositivo que faz interface com o servidor com o objetivo de ler os dados expostos do servidor e/ou controlar o comportamento do servidor. Essa interação entre Cliente e Servidor se torna mais clara a partir da visualização da Figura 2.3.10.

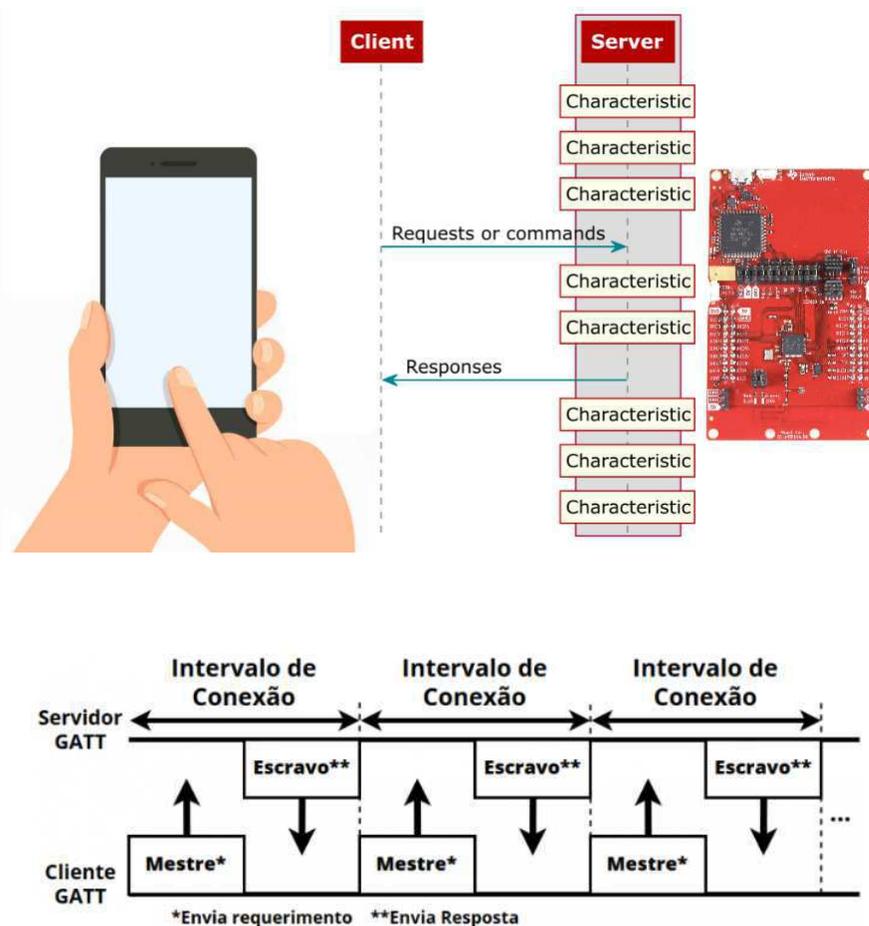


Figura 2.3.10: Comunicação entre Servidor GATT e Cliente GATT

O GATT utiliza o ATT como seu protocolo de transporte para trocar dados entre dispositivos. Esses dados são organizados hierarquicamente em seções chamadas serviços, que agrupam peças de dados do usuário conceitualmente relacionadas, denominadas características. Essa hierarquia é evidenciada na Figura 2.3.11.

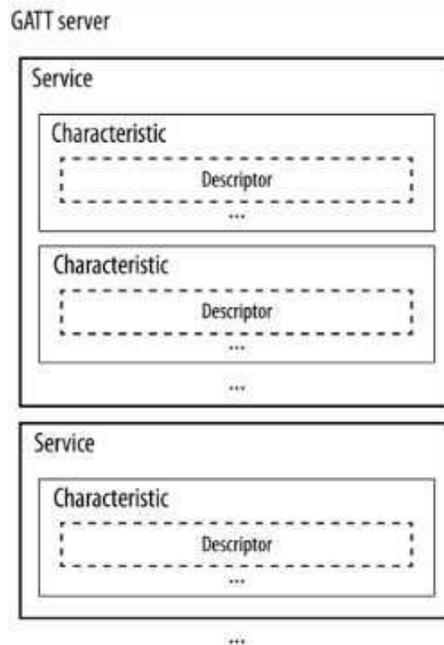


Figura 2.3.11: Hierarquia de dados do GATT

2.3.8 *General Access Profile (GAP)*

Trata-se de uma camada do protocolo BLE responsável pela função de conexão. É responsável pelo controle do modo de acesso e procedimentos do dispositivo, dentre eles, a descoberta de dispositivos, início e término da conexão, inicialização de características de segurança e configuração do dispositivo. Em outros termos, é o pilar que permite que dispositivos BLE interoperem entre si e que torna o dispositivo visível ao mundo exterior.

O GAP aplica ao BLE diferentes aspectos da interação dos dispositivos, que são conhecidos como: Papéis, Modos, Procedimentos e Segurança.

Um dispositivo pode ingressar em uma rede BLE adotando os seguintes papéis especificados nessa camada:

- *Broadcasting*: Não precisam se conectar explicitamente entre si para realizarem transferência de dados.

Dentro dessa categoria obtém-se os seguintes papéis:

Broadcaster : Dispositivo que realiza transmissão de dados através dos advertising channels, sem poder se conectar com outros dispositivos. Um bom exemplo para melhor entendimento seria um termômetro público que transmite leituras de temperatura para quaisquer dispositivos interessados. A função Broadcaster utiliza o papel Advertiser da camada de enlace.

Observador : Dispositivo que busca por anúncios, sem poder iniciar uma conexão. Um dispositivo com uma tela é uma aplicação típica de um Observador, como por exemplo um tablet que mostra para o usuário os dados de temperatura transmitidos por um Broadcaster. A função Observador utiliza o papel Scanner da camada de enlace.

- *Connecting*: Devem se conectar explicitamente entre si para realizarem transferência de dados.

Dentro dessa categoria obtém-se os seguintes papéis:

Central : Dispositivo que inicia uma conexão com um dispositivo periférico buscando primeiramente os pacotes de anúncio. Um dispositivo central pode se conectar a muitos outros dispositivos periféricos. Smartphones ou tablets na rede geralmente atuam como Central, já que ambos tem acesso a poderosos CPUs e recursos de memória, permitindo assim a perduração das conexões com vários dispositivos. A função Central corresponde ao papel Master da camada de enlace.

Periférico : Dispositivo que anuncia sua presença através de pacotes de anúncio para que os dispositivos centrais encontrem-os, e, assim, possam estabelecer uma conexão. Após a conexão, os periféricos não transmitem mais dados para outros dispositivos centrais e permanecem conectados ao dispositivo que aceitou a solicitação de conexão. A função Central corresponde ao papel Slave da camada de enlace.

Cada dispositivo em particular pode operar em um ou mais papéis de cada vez.

Os Modos são estados no qual o dispositivo pode alternar por um determinado período de tempo para atingir um objetivo particular ou permitir um par realizar um processo específico. Como exemplos temos:

- Modos de detecção

Não detectável : Significa que outros dispositivos não podem saber sobre a presença do periférico ou realizar qualquer investigação sobre sua natureza. Normalmente utilizado quando um dispositivo não quer ser encontrado por pares centrais, seja para estabelecer uma conexão ou mesmo para ser detectado.

Restritivamente detectável : Este modo permite que um dispositivo seja detectável por um período limitado e com uma prioridade reduzida.

Usualmente detectável : Este modo torna um dispositivo detectável pelo tempo que for necessário ou considerado necessário. Um dispositivo que muda para este modo expressa seu desejo de ser descoberto pares centrais, geralmente com a intenção de estabelecer uma conexão.

- Modos de conexão

Não conectável : O dispositivo é, como o nome do modo sugere, não conectável, o que significa que nenhuma central consegue estabelecer uma conexão com o aparelho.

Modo conectável direcionado : Ao realizar anúncio direcionado, um dispositivo envia pacotes de anúncio a uma alta frequência e por um curto período de tempo, sem carga de dados do usuário e com um endereço Bluetooth central de destino. Isso é fornecido como um modo de “reconexão rápida”, normalmente usado quando o periférico tem uma forte suspeita de que a central de destino já está tentando iniciar uma conexão e deseja estabelecê-la o mais rápido possível. Apenas a central cujo endereço Bluetooth coincide com o dos pacotes de anúncio enviados pelo periférico irá recebê-los

Modo conectável não direcionado :

Este é o modo conectável padrão, através do qual um periférico se torna conectável por um longo período de tempo e pode estar tentando se conectar a uma nova central ou a uma que já seja previamente conhecida por ela.

Ambos os modos conectáveis implicitamente exigem que o dispositivo envie os pacotes de anúncio com a intenção de se conectar a uma central.

Os procedimentos são uma sequência de ações realizadas por um dispositivo central para atingir uma meta, seja ela sobre conexão ou descoberta. Dessa forma, tem-se:

- Procedimentos de descoberta

Procedimento de descoberta limitada : Uma central executando este procedimento inicia a varredura ativa sem filtragem de lista branca para anunciantes e analisa cada pacote de anúncio que recebe. Se o sinalizador Detectável Limitado estiver definido, o dispositivo de mesmo nível será relatado ao aplicativo para ação futura.

Procedimento geral de descoberta : Uma central executando este procedimento inicia a varredura ativa sem filtragem de lista branca para anunciantes e analisa cada pacote de anúncio que recebe. Se os sinalizadores Detectável por Descoberta ou Detectável Geral forem definidos, o dispositivo de mesmo nível será relatado ao aplicativo para ação adicional.

- Procedimentos de conexão

Não dependem de tipos de pacotes de anúncio. Em vez disso, o tipo de procedimento de conexão usado depende do tipo de filtragem que a central impõe aos pacotes de entrada.

Procedimento de conexão automática : Com este procedimento de etapa única, o host preenche uma lista branca com uma matriz de dispositivos periféricos conhecidos e instrui o

controlador a se conectar ao primeiro detectado. De um modo geral, esse procedimento é útil quando a central já conhece um conjunto limitado de dispositivos e não tem preferência por qual deles se conectar.

Procedimento de conexão geral : Este procedimento de duas etapas é comumente usado para conectar-se a um novo periférico desconhecido. A central começa digitalizando sem uma lista branca, aceitando todos os pacotes de anúncio recebidos. Para cada periférico detectado, o aplicativo precisa decidir se deve se conectar a ele ou continuar com o próximo. Para fazer isso, o aplicativo pode solicitar ao usuário ou analisar os dados de publicidade incluídos no pacote de publicidade recebido. Quando um periférico é escolhido, a central se conecta a ele usando o procedimento de estabelecimento de conexão direta.

Procedimento de conexão seletiva : Esse procedimento é idêntico ao procedimento geral de estabelecimento de conexão, com a exceção de que o host usa uma lista branca de dispositivos conhecidos anteriormente para filtrar pacotes de anúncio de entrada. Isso pode ser útil em certos casos em que o usuário deve escolher qual dos vários periféricos conhecidos para se conectar.

Procedimento de conexão direta : Este procedimento de estabelecimento de conexão de etapa única conecta a central a um periférico específico. O host usa a camada de link para iniciar uma conexão com um único dispositivo, identificado por seu endereço Bluetooth, sem conhecimento prévio de sua presença. O procedimento pode falhar se o periférico de destino não estiver disponível ou não estiver em um modo conectável.

Por fim, mas nem por isso menos importante, a segurança é incorporada ao núcleo da tecnologia sem fio Bluetooth Low Energy. Desde a introdução do BLE na especificação do núcleo do Bluetooth 4.0, a transmissão segura de dados do usuário tem sido um objetivo principal do projeto, e seus sucessores baseiam-se em fundamentos anteriores para apertar ainda mais esses princípios de segurança.

A aplicação da segurança e da privacidade no BLE é fundamentada em dois pilares:

O primeiro deles, o Gerenciador de segurança (e seu protocolo) implementa os algoritmos criptográficos reais e trocas de protocolo, permitindo que dois dispositivos troquem dados com segurança e detectem uns aos outros de forma privada.

O segundo pilar refere-se aos Aspectos de segurança do GAP, O GAP define um conjunto de modos e procedimentos relacionados à segurança que permitem que a implementação confie na conexão para transportar dados confidenciais e aceitar com segurança a identidade de um dispositivo de mesmo nível. Os procedimentos de segurança são principalmente assimétricos, com a central e o periférico assumindo papéis diferentes durante a geração e troca de chaves e o subsequente uso deles para estabelecer uma conexão segura.

Capítulo 3

Projeto do Protocolo de Comunicação

A essa altura já foi consolidado o avanço da Internet das Coisas na área da medicina e os principais pontos e aspectos que deverão ser estudados para o entendimento e a otimização de projetos nessa área. Diante de toda a parte técnica já previamente estudada, esse capítulo abordará como tudo que já foi visto se enquadrará no respectivo projeto.

Inicialmente, entender a motivação do projeto é crucial para saber a importância da atuação da engenharia na área da saúde. O que instiga a construção dessa ideia é a sensibilidade para com todos os enfermos do Parkinson que a maioria das vezes encontram-se em uma situação de completa falta de esperança e as vezes até mesmo uma completa falta de alegria. Já existem vários projetos utilizando IoT para ajudar os portadores da doença, mas é incrível e completamente motivacional para nós, engenheiros, pensar que um simples invento pode mudar, de forma imensurável, a qualidade de vida de milhões de pessoas.

A Figura 3.0.1 é apresentada com o objetivo de informar os principais impactos dessa enfermidade ao redor do mundo.

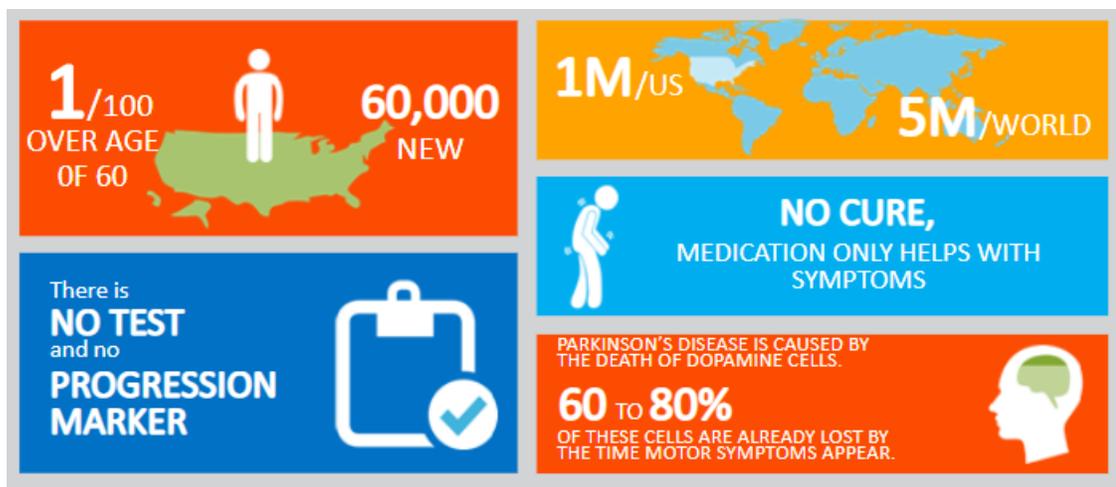


Figura 3.0.1: Doença de Parkinson

Através de todo o conhecimento adquirido sobre as camadas ATT e GATT, um serviço estruturado objetivando a utilização das suas respectivas características na troca de informações com outro dispositivo, é criado de modo que o cliente e o servidor possam interagir de modo planejado e organizado tendo em vista o auxílio para enfermos da doença citada.

Um protocolo de comunicação nada mais é que um software ou programa de computador que receberá ou enviará os dados a serem transmitidos, gerando, no início e no fim das mensagens transmitidas, os caracteres de controle, confirmação de recebimento, controle de sequência das mensagens e outros controles necessários para uma boa transmissão. Nesse capítulo será entendido como essa comunicação funcionará de acordo com o que for especificado para o projeto.

3.1 Mal de Parkinson

A Doença de Parkinson afeta a vida de um enfermo não somente nas habilidades motoras, o paciente que apresenta essa patologia exprime dificuldade na comunicação e pode até desenvolver demência de acordo com a fase da enfermidade. De acordo com a Organização Mundial da Saúde (OMS), esse distúrbio neurológico afeta cerca de 5 milhões de pessoas ao redor do mundo, das quais 200 mil casos se referem ao Brasil[12].

Os movimentos musculares do corpo utilizam uma substância química do cérebro, conhecida como dopamina, para atuar no controle do movimento e na capacidade de memória. É através dos neurotransmissores desse hormônio que ocorre a condução de impulsos nervosos para o restante do corpo. A doença de Parkinson se dá início quando as células nervosas do cérebro que produzem a dopamina são destruídas lentamente e gradativamente, levando à perda da atividade muscular.

A pretensão da investigação pela cura dessa enfermidade é de longa data, e infelizmente, nenhuma solução foi encontrada até então. Apesar da falta de conhecimento sobre o porquê que a doença ocorre e qual seria seu possível tratamento, há inúmeros recursos disponíveis para um possível controle de sintomas e retardamento do encadeamento desse mal.

No ano de 2017, a Microsoft expôs durante a Build 2017 o relógio conhecido como “Emma Watch”. O mesmo foi criado como uma solução para auxílio na redução dos tremores de uma pessoa específica, Emma, portadora da doença de Parkinson. O protótipo emite vibrações, semelhantes aos gerados pelos celulares, que “enganam”o cérebro, fazendo com que tremores na mão diminuam sensivelmente, mesmo que não cessem completamente.



Figura 3.1.1: Emma Watch

Deste modo, a partir de estudos com base no Emma Watch, mostrado na Figura 3.1.1, e objetivando uma maior solidariedade para com os pacientes que apresentam a doença de Parkinson que esse projeto é moldado. A engenharia conecta-se com a medicina para uma significativa assistência física com os portadores desse distúrbio. A partir do desenvolvimento do protótipo de uma pulseira auxiliadora e com fundamento na investigação, especificação e implementação de um protocolo de comunicação sem fio, Bluetooth, o usuário terá acesso aos serviços prestados pela pulseira através dessa conectividade.

A Figura 3.1.2 expressa a situação referida em forma de um diagrama simples. O sistema geral é formado por dois dispositivos: uma pulseira e um computador. O foco do projeto é justamente conseguir conectar esses dois aparelhos via BLE para uma bem sucedida e segura troca de informações. Sejam essas informações relativas a troca de dados ou comandos de controle (como o controle de um vibra inserido na pulseira, por exemplo).

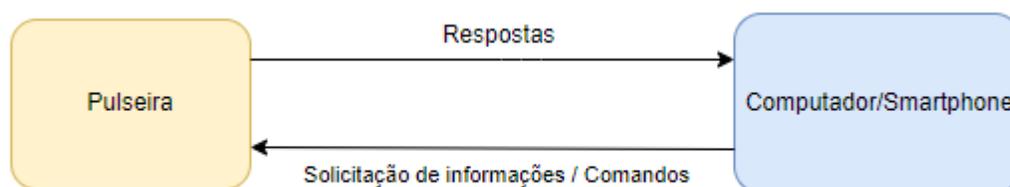


Figura 3.1.2: Diagrama geral do projeto

3.2 O GATT na especificação do projeto

Como a camada GAP manipula a maioria das funcionalidades relacionadas à conexão, a camada GATT da pilha de protocolos do BLE é utilizada para aplicação de comunicação de dados entre dois dispositivos conectados. Os dados são passados e armazenados na forma de características que são guardadas na memória do dispositivo Bluetooth de baixa energia.

O primeiro propósito do projeto foi então a pesquisa de uma definição geral do serviço que precisaríamos estabelecer e suas respectivas características, gerando assim uma ideia ampla do que demandava ser realizado. O diagrama da Figura 3.2.1 apresenta a distribuição de serviços e características GATT a serem utilizadas pela protótipo alvo desse projeto.

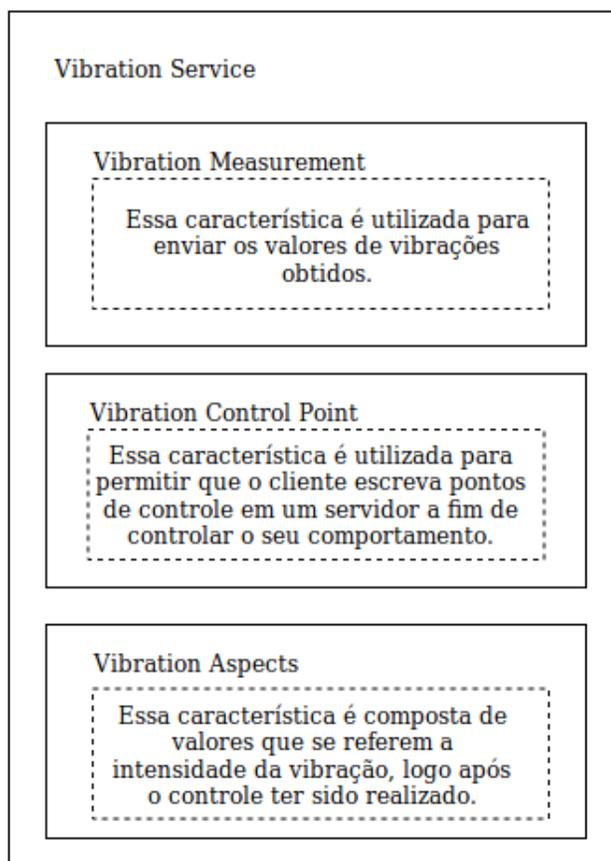


Figura 3.2.1: Hierarquia de dados do GATT para o projeto

A seguir serão descritas as principais características do projeto.

Serviço: Vibration

Descrição: O serviço em questão expõe a intensidade do tremor e outros dados referentes ao bracelete destinado a aplicações de saúde para portadores de Parkinson.

Tabela 3.1: Requisitos GATT

Subprocedimento	Requisito do servidor
Escrever valores característicos	Mandatório
Notificações	Mandatório
Ler descritores característicos	Mandatório
Escrever descritores característicos	Mandatório

No momento em que o portador de Parkinson utilizar o nosso protótipo de pulseira, a intenção é que, a partir da atuação dos acelerômetros, seja fornecido a localização exata do braço dele, a partir das coordenadas x , y e z . Se o usuário sofrer de um alto grau de Parkinson é constável que esses valores mudarão significavelmente em um curto período de tempo. Por tudo que já foi previamente estudado, nota-se que o sistema da pulseira, formado por um microcontrolador com, essencialmente, acelerômetros e vibrações, refere-se ao servidor, e um computador para testes, refere-se ao cliente.

Assim como as notificações são bastante oportunas no projeto, tanto para notificar falhas, quanto para notificar sucesso na operação desejada, é de extrema importância que o cliente escreva o comando de controle para os motores atuarem na frequência desejada. Assim, o usuário poderá administrar a performance dos motores de acordo com a sua necessidade. Por fim, logo após a utilização dos motores para o auxílio à diminuição das vibrações no braço, novos valores são enviados e lidos novamente pelo cliente, para um maior discernimento sobre uma possível melhoria no tremor.

Tabela 3.2: Dependências de transporte

Transporte	Suportado
Clássico	Falso
Baixa Energia	Verdadeiro
Alta Velocidade	Falso

A tecnologia BLE oferece uma variedade considerável de aplicações possíveis, entre elas é notável que parte de healthcare vem sendo cada vez mais utilizada. As chaves para essa tecnologia são a simplicidade, a compatibilidade com vários dispositivos e o baixo custo no desenvolvimento e no suporte. Por esses motivos e por ser o foco da pesquisa em si, nosso projeto utiliza o Bluetooth Low Energy 4.2 para estabelecer as devidas conexões assim como trocas de dados e informações.

3.3 Princípio de Funcionamento

Os dispositivos BLE expõem sua funcionalidade por meio da coleção de serviços, características e descritores. Os Serviços definem o contrato funcional do dispositivo LE e contêm uma coleção de características que definem o próprio serviço. Essas características, por sua vez, contêm descritores que descrevem as características. Esses 3 termos são genericamente conhecidos como os atributos de um dispositivo. Essa série de atributos são conhecidos como estruturas de dados que guardam informações a serem utilizadas pelo GATT ao cliente.

Quando uma solicitação de leitura de um Cliente GATT é recebida para um determinado atributo, a pilha de protocolos verifica as permissões do atributo e, se o atributo for legível, chama o perfil de retorno de chamada de leitura. O perfil copia o valor, realiza qualquer processamento específico do perfil e notifica o aplicativo, se desejado.

No caso do projeto em questão, o cliente solicita ao servidor a leitura referente aos dados do acelerômetro, tanto antes quanto depois do comando vibra ser acionado. Caso ocorra algum erro na leitura, o cliente é notificado. Caso não ocorra falhas, o atributo requerido é procurado dentro do servidor e enviado de volta para o cliente efetuar a leitura solicitada; A Figura 3.3.1 demonstra como essas solicitações de leitura funcionam na prática. Como já mencionado anteriormente, essas informações são de extrema importância para a evolução e acompanhamento do projeto e do suposto e esperado bem estar do paciente.

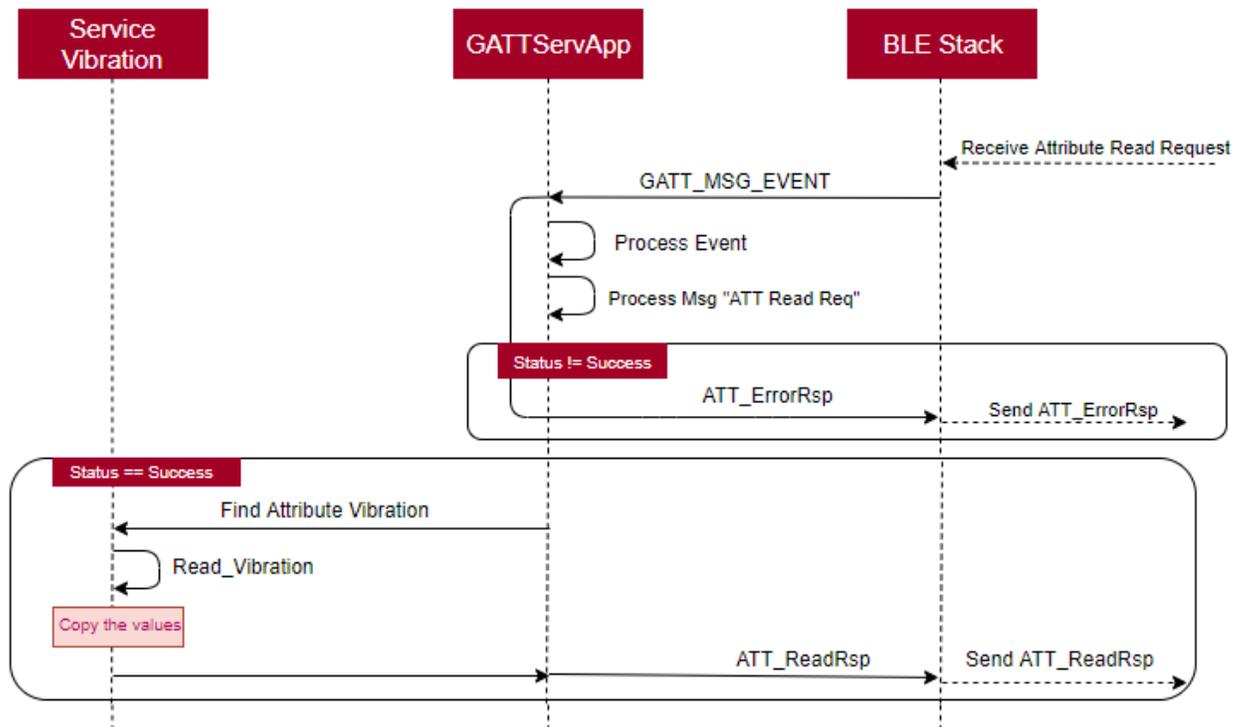


Figura 3.3.1: Pedido de leitura de dados do cliente

O GATT Server Application (GATTServApp) armazena e gerencia a tabela de atributos de todo o aplicativo. Vários perfis usam esse módulo para adicionar suas características à tabela de atributos. A pilha do BLE usa esse módulo para responder às solicitações de descoberta de um cliente GATT. Por exemplo, um cliente GATT pode enviar uma mensagem com a finalidade de descobrir todas as características principais. A pilha BLE no lado do servidor GATT recebe essa mensagem e usa o GATTServApp para localizar e enviar por via aérea todas as características primárias armazenadas na tabela de atributos.

Já quando uma solicitação de escrita de um cliente GATT é recebida para um determinado atributo, a pilha de protocolos verifica as permissões do atributo e, se o atributo for permitido por escrita, chama o retorno de chamada de escrita do perfil. O perfil armazena o valor a ser escrito, realiza qualquer processamento específico do perfil e notifica o aplicativo, se desejado.

No caso do projeto em questão, o cliente novamente faz uma solicitação ao servidor, mas dessa vez demanda uma escrita. Seguindo a mesma lógica da leitura, caso ocorra falhas, o cliente é notificado imediatamente. Caso não ocorra problemas, o atributo requerido é novamente procurado, só que dessa vez o cliente é apto a modificar valores, podendo assim enviar comandos de controle para o que se deseja ser modificado; A Figura 3.3.2 demonstra como essas solicitações de escrita funcionam na prática. No nosso caso, essa solicitação tem como objetivo controlar a atuação dos vibras, definindo seu respectivo comportamento.

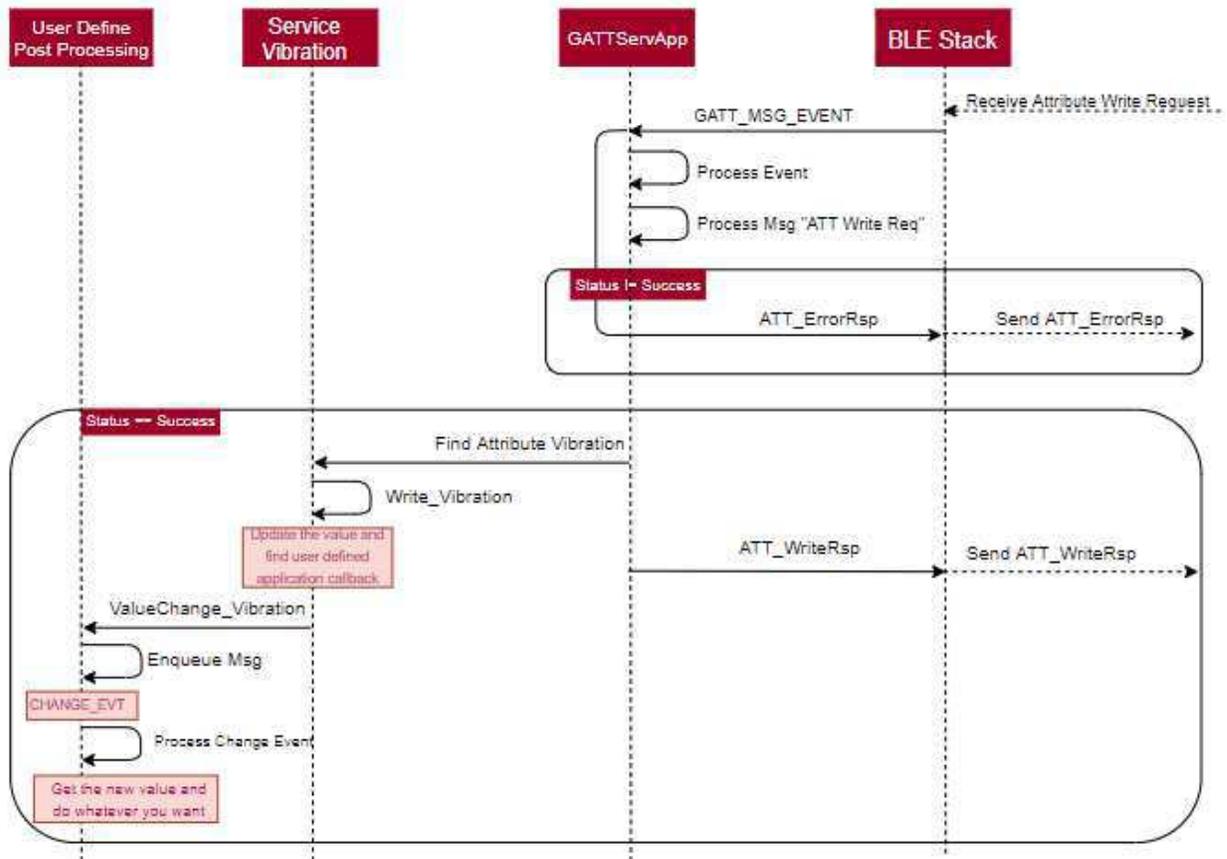


Figura 3.3.2: Pedido de escrita de dados do cliente

Capítulo 4

Desenvolvimento do Serviço Bluetooth

O foco desse capítulo é a implementação e a validação do protocolo de comunicação Bluetooth. A validação do software é um processo que comprova que o sistema cumpre com as funções as quais foi designado e mostra-se como uma importante ferramenta ao final do projeto.

Inicialmente, é montada uma visão geral do sistema físico; Embora o foco seja na parte do software, é importante lembrar que tanto o hardware quanto o software agem em conjunto para fornecer as ferramentas necessárias desenvolvidas ao decorrer do projeto. Os requisitos funcionais e não funcionais do sistema também são apresentados de uma forma geral, com o intuito de orientar os principais aspectos previstos para o projeto.

4.1 Visão Geral do Sistema

A parte física que compõe o projeto não será profundamente abordada, mas é importante uma perspectiva geral dos componentes necessários para uma troca de informações bem sucedida de acordo com o que se pretende executar, para isso um diagrama geral do sistema foi montado e apresentado na Tabela 4.1. O alvo era o ESP32 como o microcontrolador do sistema, já que o mesmo apresentava uma grande versatilidade, além de apresentar uma programação fácil de ser entendida e adaptada.

Instrumento	Função Geral no projeto
ESP32	Necessário para estabelecer a conexão dos componentes do circuito, com um computador.
Motor Vibracall 1027	Utilizado para gerar uma vibração na pulseira, o esperado é que o cérebro se concentre no pulso reduzindo as mensagens do cérebro para aquele ponto.
Acelerômetro MPU6050	Responsável por medir a inclinação e o movimento dos tremores no braço provenientes do Parkinson.
Pulseira em Velcro	Dispositivo vestível conectado aos componentes do circuito.

Tabela 4.1: Cenário Hardware dos principais componentes do projeto

4.2 Requisitos do sistema

Ao começar um projeto, os requisitos devem ser levantados, entendidos e documentados. Assim como realizar atividades de controle de qualidade para verificar, validar e garantir a qualidade dos mesmos. Gerenciar a evolução dos requisitos é importante, estando ciente de que os negócios com sua dinâmica não garantem estabilidade e podem vir a sofrer alterações. Desse modo é necessário manter a rastreabilidade entre os requisitos e as outras peças do projeto.

De um modo geral, o conjunto de requisitos de um sistema é definido durante as fases iniciais do processo de desenvolvimento. Tal conjunto é visto como uma especificação do que deveria ser implementado. Os requisitos são descrições de como o sistema deveria se comportar, e contêm informações do domínio da aplicação e restrições sobre a operação do sistema. As Tabelas 4.2 e 4.3 respectivamente apresentam os requisitos funcionais e não funcionais do nosso projeto.

Nome	Dispositivo Vestível
Descrição	Deve funcionar de maneira correta, enviando os dados do acelerômetro para o cliente e recebendo os comandos de vibração do mesmo.
Objetivo	Disponibilização desse dispositivo para usuários que o necessitam.

Nome	Conectabilidade
Descrição	A conexão entre o dispositivo vestível e o computador deve ser realizada por intermédio da conexão Bluetooth Low Energy.
Objetivo	Transferência de dados e informações entre cliente e servidor assegurando um baixo consumo de energia.

Tabela 4.2: Requisitos Funcionais

Nome	Mudanças
Descrição	O sistema deverá ser desenvolvido de modo a garantir a facilidade de possíveis mudanças.
Objetivo	Assegurar que uma possível melhoria no projeto será fácil de ser implementada.

Nome	Avaliações
Descrição	O sistema deverá ser desenvolvido de modo a possibilitar a fácil implementação de testes para possíveis otimizações no sistema.
Objetivo	Assegurar que o projeto não demonstre erros.

Nome	Reusabilidade
Descrição	O sistema deve ser desenvolvido de modo existir a possibilidade de reusabilidade do mesmo para outras aplicações.
Objetivo	Assegurar que o projeto possa servir de base para outros projetos com a mesma temática.

Tabela 4.3: Requisitos Não Funcionais

4.3 Implementação

Para implementar o que já foi previamente definido na especificação do protocolo de comunicação, gerou-se um código tendo como base o exemplo UART encontrado na biblioteca ESP32. O Microcontrolador ESP32 é um dispositivo para uso no campo IoT, mostrando-se como uma melhoria do seu antecessor, ESP8266, por apresentar um maior poder de processamento (Triple core), memória e novos recursos, incluindo Bluetooth 4.2 e sensores de touch capacitivo. Assim, acaba não sendo novidade uma significativa e recente procura por esse microcontrolador. Pelo que já foi anteriormente estudado, é compreendido que o ESP32 se tornará para o nosso projeto, um servidor BLE.

Primeiramente, faz-se necessário a inclusão de um dispositivo, em termos de hardware e software, e um servidor que detenha serviços, por isso ambos são devidamente criados em duas linhas de código mostradas e comentadas na Figura 4.3.1. É importante observar que o projeto em si é formado tanto pelo microcontrolador ESP32, quanto por acelerômetros, vibras, resistores, entre outros. Então, nosso dispositivo acaba sendo toda essa integração de equipamentos que forma o protótipo da pulseira, inicialmente denominada como “PWatch”.

```
BLEDevice::init("PWatch"); // Init the BLE Device
BLEServer *pServer = BLEDevice::createServer(); // Create the BLE Server
```

Figura 4.3.1: Inclusão do dispositivo PWatch e criação de um servidor

Logo após isso, o servidor deve implementar os seus devidos serviços, que pode ser realizado de duas formas:

A primeira delas é utilizando um serviço já existente. Pode parecer estranho mas acaba sendo comum a utilização de um serviço previamente efetivado. Um exemplo é o serviço de bateria, mostrado na Figura 4.3.2, que expõe o estado da bateria e o nível da bateria em um dispositivo. Suas devidas informações são encontradas no próprio site do Bluetooth.

Name	Uniform Type Identifier	Assigned Number	Specification
Generic Access	org.bluetooth.service.generic_access	0x1800	GSS
Alert Notification Service	org.bluetooth.service.alert_notification	0x1811	GSS
Automation IO	org.bluetooth.service.automation_io	0x1815	GSS
Battery Service	org.bluetooth.service.battery_service	0x180F	GSS

Figura 4.3.2: Exemplos de serviços GATT já efetivados

Dessa forma, o trecho do código que corresponderia a esse serviço já existente, seria escrito da seguinte forma especificada na Figura 4.3.3

```
//Create the BLE Service
BLEService *pService = pServer->createService(BLEUUID((uint16_t)0x180F));
```

Figura 4.3.3: Criação de um serviço que já existe

A segunda delas é utilizada quando não há um serviço semelhante existente, como pode ser visto na Figura 4.3.4. Com isso, dentro dessa alternativa, encontram-se também duas soluções bastante simples. Sendo a primeira delas utilizar o UUID dos próprios exemplos de código do ESP32 BLE, mas lembrando que outras pessoas tem acesso ao mesmo UUID; E a segunda opção, que será a utilizada na projeto, é a criação de UUID próprios, através de um site online chamado de “UUID Generator”.¹

```
//Create the BLE Service
BLEService *pService = pServer->createService(SERVICEVIBRATION_UUID);
```

Figura 4.3.4: Criação de um serviço novo

Sendo o “SERVICE_UUID” definido previamente no próprio programa, como mostra a Figura 4.3.5:

```
// UUID Gerado no UUIDGenerator
#define SERVICEVIBRATION_UUID "54210a05-191d-4c7d-a9eb-bff9alb6alc8"
```

Figura 4.3.5: Definição do serviço de acordo com o UUID gerado

¹<https://www.uuidgenerator.net/>

Como foi definido anteriormente, o serviço de vibração contém 3 características, então cada uma delas foi apropriadamente criada, respeitando suas respectivas propriedades. Os trechos de código mostrados na Figura 4.3.6 apresentam fragmentos da criação dessas características no ESP32.

```
// Create a BLE Characteristic : sending data to the client
BLECharacteristic *pCharacteristicVibrationMeasurement = pService->createCharacteristic(
    CHARACTERISTIC_UUID_VibrationMeasurement,
    BLECharacteristic::PROPERTY_NOTIFY
);

// Create a BLE Characteristic : receiving data from the client (command)
BLECharacteristic *pCharacteristicVibrationControlPoint = pService->createCharacteristic(
    CHARACTERISTIC_UUID_VibrationControlPoint,
    BLECharacteristic::PROPERTY_WRITE
);

// Create a BLE Characteristic: sending data that the client can read
BLECharacteristic *pCharacteristicVibrationAspects = pService->createCharacteristic(
    CHARACTERISTIC_UUID_VibrationAspects,
    BLECharacteristic::PROPERTY_READ
);
```

Figura 4.3.6: Criação das respectivas característica do serviço

Assim como o serviço de vibração teve seu próprio UUID, foi realizado o mesmo procedimento para autenticidade das suas características, o resultado encontra-se na Figura 4.3.7.

```
//Todos os UUID Gerados no UUIDGenerator
#define SERVICEVIBRATION_UUID "54210a05-191d-4c7d-a9eb-bff9alb6alc8"
#define CHARACTERISTIC_UUID_VibrationMeasurement "a3397cf2-7fd9-4201-9847-1735f781f8b5"
#define CHARACTERISTIC_UUID_VibrationControlPoint "ffc12d00-282c-4863-980a-f49573943fbe"
#define CHARACTERISTIC_UUID_VibrationAspects "614a887e-0ael-41d1-85ac-552022ea2cfc"
```

Figura 4.3.7: Definição das características de acordo com os UUIDs gerados

4.4 Validação

Através do aplicativo conhecido como “LightBlue”², além de ser possível realizar a conexão com todos os dispositivos que utilizam o recurso Bluetooth Low Energy, obtem-se o suporte completo de leitura, escrita e notificação ao vincular-se com um dispositivo.

O aplicativo foi de extrema importância para testes de funcionamento do código e otimização de processos. Para o código básico descrito na seção anterior, com suas respectivas definições de características, foi testado no LightBlue se essas informações realmente seriam oferecidas. A Figura 4.4.1 apresenta uma captura de tela com detalhes do resultado da conexão com o PWatch. Facilmente identificamos que todas as características foram corretamente criadas e agem no serviço vibração.

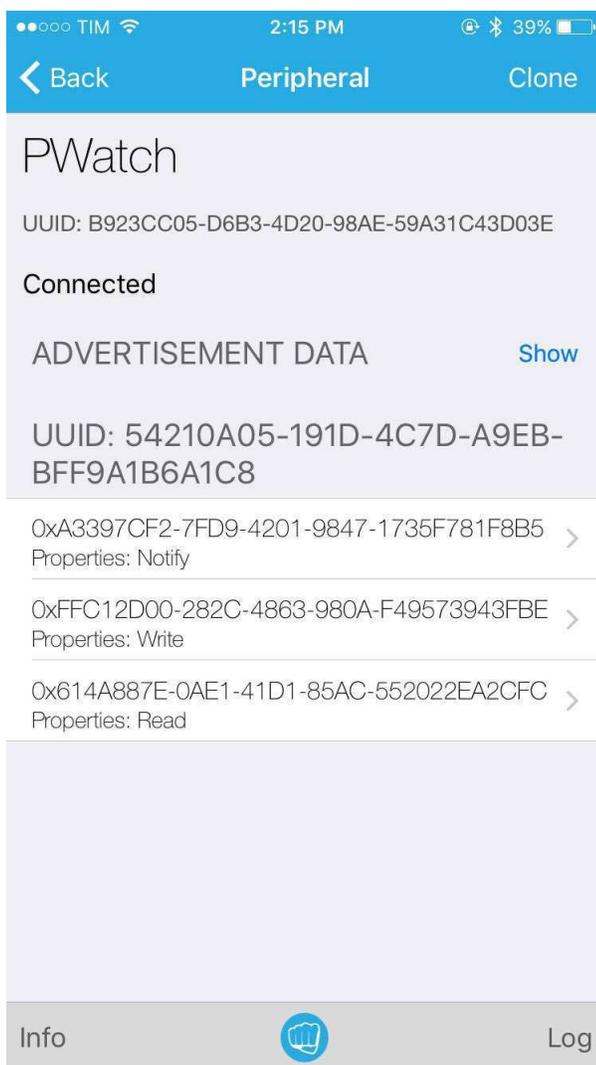


Figura 4.4.1: Aplicativo LightBlue informando as características do dispositivo conectado

²<https://itunes.apple.com/br/app/lightblue-explorer/id557428110?mt=8>

Realizou-se um teste para entender a atuação do servidor utilizando novamente a IDE do próprio arduino. Um array de dados foi enviado para que o cliente pudesse efetuar uma possível leitura. É bom salientar que esse array, expresso em valores de coordenadas x, y e z equivale a uma medição já feita da posição do braço de uma determinada pessoa em determinado momento.

Estabeleceu-se o seguinte valor previamente medido: (-468, -7748, 13744”) e pela captura de tela mostrada na Figura 4.4.2, pode-se observar que o teste foi bem sucedido, oferecendo ao cliente as informações desejadas.

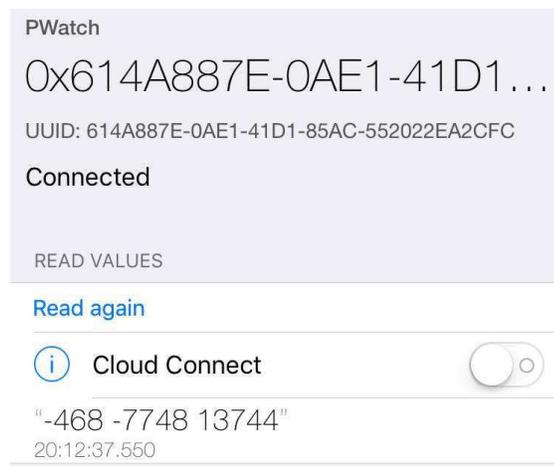


Figura 4.4.2: Leitura das informações fornecidas pelo servidor

Pode-se observar que o cliente consegue receber e ler os dados do array sem problemas via Bluetooth. Como foi visto anteriormente, esses dados são importantes para um possível diagnóstico de melhoria do paciente. Quando o enfermo não apresenta uma vibração significativa nas mãos é facilmente visualizado que a discrepância de valores de x, y e z de um array para o outro não é grande.

A partir da característica da escrita, o cliente é apto a escrever valores desejados. No projeto, esses valores correspondem a frequência de vibração em que os motores deveriam atuar. Mas para testes, foi escrito um valor qualquer para observar se essa funcionalidade estava dentro do esperado. Foi decidido então, escrever o número 111111 no próprio LightBlue. O entendimento de que só o fato de que o cliente é capaz de acessar um comando de escrita, já mostra que a característica está executando seus devidos propósitos, como mostrado na Figura 4.4.3.



Figura 4.4.3: Escrita da informação desejada pelo próprio cliente

Capítulo 5

Conclusão

No decorrer desse trabalho a importância da Internet das Coisas ficou cada vez mais notável. Embora exista uma tendência de utilização da tecnologia IoT na área de saúde apenas no monitoramento de pacientes, foi visto que essa tecnologia pode ser bem mais ampla do que isso. Uma visão mais global de IoT na área de saúde pode contemplar a área de dispositivos e ferramentas de saúde inteligentes, transferência de dados, armazenamento e colaboração.

O Bluetooth Low Energy mostrou-se como um grande aliado ao IoT, sendo ideal para aplicações que requerem transferência periódica de pequenas quantidades de dados. Assim sendo, O BLE é especialmente adequado para sensores, atuadores e outros dispositivos pequenos que requerem consumo de energia extremamente baixo.

O BLE utiliza um modelo de cliente / servidor. Um cliente, que requer dados, se conecta e acessa um ou vários servidores, que oferecem esses dados. O cliente normalmente opera na função de central e o servidor opera na função de periférico. Normalmente, um sensor ou um acessório é o servidor / periférico e um computador, telefone ou tablet é o dispositivo cliente / central. O entendimento de como essa conexão entre cliente e servidor pode ser estabelecida, foi realizada em virtude do estudo detalhado da pilha de protocolos BLE, mostrando também sua importância técnica no desenvolvimento de um projeto.

Nesse projeto foram desenvolvidos trabalhos somente com o ESP32, utilizando uma camada de abstração de hardware para a execução de testes com um possível servidor. Uma sugestão é a incrementação de mais complexos e detalhados requisitos para o sistema, lembrando que são primordiais para a boa manutenção e atualização do projeto desenvolvido.

Referências Bibliográficas

- [1] TOWNSEND, K.; Cufí, C.; Akiba; Davidson, R. *Getting Started with Bluetooth Low Energy.*, 1ª edição. OREILLY. 2014.
- [2] *5 Exemplos de IoT na área da saúde.* ICLINIC Disponível em <<https://blog.iclinic.com.br/exemplos-de-iot-na-area-da-saude/>>. Acesso em: 07 de outubro de 2018.
- [3] *O que é Internet das Coisas: a tecnologia está integrando o mundo.* Disponível em <<https://blog.mastertech.com.br/tecnologia/o-que-e-internet-das-coisas-tecnologia-esta-integrando-o-mundo/>>. Acesso em: 07 de outubro de 2018.
- [4] *A Internet das Coisas na saúde.* SETORSAUDE. Disponível em <<https://setorsaude.com.br/a-internet-das-coisas-na-saude/>>. Acesso em: 07 de outubro de 2018.
- [5] ASHTON, K. *That 'Internet of Things' Thing.* RFID JOURNAL. Disponível em <<https://www.rfidjournal.com/articles/view?4986>>. Acesso em: 08 de outubro de 2018.
- [6] *Who Invented Bluetooth?.* THOUGHT CO. Disponível em <<https://www.thoughtco.com/who-invented-bluetooth-4038864>>. Acesso em: 08 de outubro de 2018.
- [7] *Internet das coisas: entenda os seus impactos no mundo da medicina.* TELEMEDICINA. Disponível em <<http://portaltelemedicina.com.br/internet-das-coisas-entenda-os-seus-impactos-no-mundo-da-medicina/>>. Acesso em: 08 de outubro de 2018.
- [8] *Brief introduction to Bluetooth Low Energy.* APAGIARO. Disponível em <<https://apagiario.it/bluetooth-low-energy-presentation/>>. Acesso em: 14 de outubro de 2018.
- [9] *Bluetooth low energy Characteristics, a beginner's tutorial.* NORDIC SEMICONDUCTOR. Disponível em <<https://devzone.nordicsemi.com/tutorials/b/bluetooth-low-energy/posts/ble-characteristics-a-beginners-tutorial/>>. Acesso em: 14 de outubro de 2018.

- [10] *ATT and GATT*. EPXX. Disponível em: <https://epxx.co/artigos/bluetooth_gatt.html>. Acesso em: 16 de outubro de 2018.
- [11] PESSOA, L. *Visão Técnica do Bluetooth Smart*. Disponível em: <<https://www.embarcados.com.br/bluetooth-smart-visao-tecnica/>>. Acesso em: 16 de outubro de 2018.
- [12] *Mais de 200 mil pessoas apresentam mal de Parkinson no Brasil*. R7. Disponível em: <<https://noticias.r7.com/saude/mais-de-200-mil-pessoas-apresentam-mal-de-parkinson-no-brasil-11042018>>. Acesso em: 16 de outubro de 2018.
- [13] IWAMI, B.S. *Advertising em Bluetooth Low Energy*. Monografia - Universidade de São Paulo, São Carlos, 2018.
- [14] Guerra, D. *Bluetooth Low Energy - Redes BLE e aplicações em Redes de Sensores*. Dissertação de Mestrado - Técnico Lisboa, Lisboa, 2015.
- [15] MAIO, A.J.F *Bluetooth Low Energy para Monitorização da Postura no Ciclismo*. Dissertação de Mestrado - Universidade do Minho, Braga, 2014.