

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**



BRUNO SANTOS CAMPOS

**DISPOSITIVO ELETRÔNICO VESTÍVEL DE
MONITORAMENTO DE IDOSOS**

VITÓRIA – ES
ABRIL/2017

BRUNO SANTOS CAMPOS

**DISPOSITIVO ELETRÔNICO VESTÍVEL DE MONITORAMENTO
DE IDOSOS**

Parte manuscrita da Proposta de Projeto de Graduação do aluno **Bruno Santos Campos**, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para aprovação na disciplina “Projeto de Graduação 2”.

Prof. Dr.rer.nat. Hans-Jorg A. Schneebeli
Orientador

Bruno Santos Campos
Aluno

VITÓRIA – ES
ABRIL/2017

BRUNO SANTOS CAMPOS

DISPOSITIVO ELETRÔNICO VESTÍVEL DE MONITORAMENTO DE IDOSOS

Parte manuscrita do projeto de graduação do aluno **Bruno Santos Campos**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovado em 20 de abril de 2017.

COMISSÃO EXAMINADORA:

Prof. Dr.rer.nat. Hans-Jorg A. Schneebeli
Orientador

Profa. Dra. Eliete Maria de O. Caldeira
Examinador

Profa. Dra. Raquel Frizera Vassallo
Examinador

RESUMO

Este trabalho apresenta um projeto de dispositivo eletrônico vestível, bem como o embasamento teórico, seus objetivos, estrutura, testes e resultados. O dispositivo desenvolvido neste projeto será uma plataforma portátil de monitoramento de atividades humanas, instalada na forma de uma cinta vestível e possuirá um sistema embarcado microcontrolado para implementar funções como o botão de pânico e sensoriamento de movimentação com detecção de quedas, recursos úteis em casos de emergência. O público-alvo do projeto são os idosos e pessoas com necessidade de atenção especial. Foi realizada uma análise técnica das tecnologias disponíveis para cumprimento de cada etapa do projeto e as escolhas foram devidamente justificadas. O objetivo era obter um equipamento eletrônico eficiente, de baixo custo, capacidade técnica adequada, que atendesse às demandas do usuário, constituísse uma plataforma com funcionalidades expansíveis, configurando assim a melhor solução disponível para o problema. Ao final deste trabalho obteve-se um sistema embarcado capaz de detectar quedas e emitir sinais de socorro do usuário para seus cuidadores via *Twitter*, permitindo a redução do tempo de espera por atendimento em casos urgentes. A comunicação entre o módulo e a central é realizada por um par de transceptores bidirecionais que cobrem toda a área da residência do usuário, atendendo aos requisitos do projeto.

Palavras chave: dispositivos vestíveis, computadores vestíveis, sistemas embarcados, sistema microcontrolado, botão de pânico, monitoramento de idosos, suporte ao idoso.

LISTA DE FIGURAS

Figura 1 - Etapas de desenvolvimento de um sistema embarcado.	15
Figura 2 – Arquitetura do dispositivo móvel ligado ao usuário.	20
Figura 3 – Arquitetura da central do sistema instalada em ponto fixo.	21
Figura 4 – Estrutura do sistema de monitoramento remoto.	21
Figura 5 – Plataforma de desenvolvimento <i>open-source</i> Arduino UNO R3.	23
Figura 6 – ATmega328P: Corrente de alimentação ativa vs. Frequência de operação (1-20MHz).	25
Figura 7 – ATmega328P: Corrente de alimentação ociosa vs. Frequência de operação (1-20MHz).	25
Figura 8 – Posicionamento dos eixos do acelerômetro, representado pelo disco azul.	26
Figura 9 – Gráfico comparativo das principais tecnologias de comunicação sem fio.	28
Figura 10 – Arquitetura de <i>hardware</i> do dispositivo vestível.	31
Figura 11 – Arquitetura de hardware da central do sistema.	32
Figura 12 – Fluxograma da rotina principal do dispositivo vestível.	35
Figura 13 – Fluxograma da estrutura das interrupções do dispositivo vestível.	38
Figura 14 – Fluxograma da rotina de envio de dados.	38
Figura 15 – Fluxograma da rotina principal da central fixa do sistema.	40
Figura 16 – Protótipo da central do sistema de monitoramento remoto.	42
Figura 17 – Protótipo do dispositivo vestível.	43
Figura 18 – Sinal de saída do eixo ‘y’ do acelerômetro obtido com o usuário de pé e parado.	45
Figura 19 – Sinal de saída do eixo ‘y’ do acelerômetro obtido durante caminhada lenta do usuário.	45
Figura 20 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto o usuário sentava-se e levantava-se de um banco.	46
Figura 21 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto o usuário caía de joelhos.	46
Figura 22 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto o usuário caía de lado.	47
Figura 23 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto o usuário caía para trás.	47

Figura 24 – Sinal de saída do eixo ‘y’ do acelerômetro obtido simulando uma queda frontal.....	48
Figura 25 – Sinal de saída do eixo ‘y’ do acelerômetro obtido simulando uma queda enquanto o usuário estava deitado.....	48
Figura 26 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto usuário caminhava rapidamente.....	49
Figura 27 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto usuário dava um pulo.	49
Figura 28 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto usuário realizava uma sequência de pulos curtos.	50
Figura 29 – Central do sistema com o módulo receptor de sinal RF de 433MHz.....	51
Figura 30 – Planta baixa da residência com representação da localização da central e áreas de interesse.....	52
Figura 31 – Módulo NRF24L01 com antena SMA de alto alcance..	53
Figura 31 – Página da conta criada no <i>Twitter</i> com as mensagens enviadas pelo sistema.....	54
Figura 32 – <i>Printscreen</i> da tela do usuário cuidador com detalhe para a notificação recebida.	55

LISTA DE TABELAS

Tabela 1 – Especificações técnicas da plataforma Arduino onde o microcontrolador estaria instalado.....	24
Tabela 2 – Especificações técnicas do módulo de comunicação NRF24L01.	29
Tabela 3 – Especificações técnicas do regulador de tensão ideal para o projeto	30
Tabela 4 – Cálculo de autonomia do sistema	30

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
ARM	<i>Advanced RISC Machine</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
EBC	Empresa Brasil Comunicação
GSM	<i>Global System for Mobile Communications</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
LED	<i>Light Emitting Diode</i>
IP	<i>Internet Protocol</i>
OMS	Organização Mundial da Saúde
RF	Rádio Frequência
RISC	<i>Reduced instruction set computing</i>
SMS	<i>Short Message Service</i>
UFES	Universidade Federal do Espírito Santo
USB	<i>Universal Serial Bus</i>

SUMÁRIO

1. INTRODUÇÃO	9
1.1 Motivação	9
1.2 Computadores vestíveis.....	10
1.3 Definição do problema	13
1.4 Metodologia.....	14
1.5 Estrutura do trabalho	16
2. MONITORAMENTO DE ATIVIDADES	17
2.1 Aquisição de informações	17
2.2 Arquitetura do sistema.....	18
3. DESENVOLVIMENTO DO HARDWARE	22
3.1 Arquitetura de <i>hardware</i> do dispositivo vestível	22
3.2 Arquitetura de <i>hardware</i> da central do sistema de monitoramento	32
4. DESENVOLVIMENTO DO SOFTWARE.....	34
4.1 Arquitetura de <i>software</i> do dispositivo vestível	34
4.2 Arquitetura de software da central do sistema de monitoramento	39
5. RESULTADOS EXPERIMENTAIS	42
5.1 Resultados dos testes de detecção de quedas	44
5.2 Resultados dos testes de comunicação entre o dispositivo móvel e a central.	50
5.3 Resultados dos testes de comunicação entre o sistema e os cuidadores.....	53
5.4 Análise dos resultados	55
6. CONCLUSÃO	57
REFERÊNCIAS BIBLIOGRÁFICAS	60
ANEXO	63
APENDICE	64

1. INTRODUÇÃO

1.1 Motivação

Conforme o Relatório Mundial de Saúde e Envelhecimento de 2015 da Organização Mundial de Saúde, o número de pessoas com mais de 60 anos no Brasil deverá crescer muito mais rápido que a média internacional (OMS, 2015). Em 2015 o Brasil possuía 23 milhões de pessoas acima de 60 anos, o que correspondia a 12,5% da população. Ainda segundo a OMS, enquanto a quantidade de idosos vai duplicar no mundo até o ano de 2050, ela quase triplicará no Brasil.

Em paralelo a esta estatística, segundo o Instituto Brasileiro de Geografia e Estatística de 2005 a 2015, o número de pessoas que moram sozinhas aumentou no país de 10,4% para 14,6%, especialmente a partir dos 50 anos (IBGE, 2015). Nesta faixa etária, a proporção de arranjos unipessoais subiu de 57,3% para 63,7% no período. Este crescimento está relacionado com o envelhecimento da população. No total, 15,7% das pessoas que têm mais de 60 anos não têm companhia em casa.

Embora muitos idosos façam questão de manter a sua independência, as pessoas com idade superior a 60 anos precisam de alguns cuidados especiais pois, devido à perda de reserva funcional, massa muscular e agilidade, esta classe está sujeita a maiores riscos de quedas e outros acidentes (GLOBO ESPORTE, 2013). Segundo estudo realizado pelo Laboratório de Pesquisa em Envelhecimento Humano da UERJ (Universidade Estadual do Rio de Janeiro), em conjunto com o CNPq (Conselho Nacional de Pesquisas) e a FAPERJ (Fundação de Apoio à Pesquisa do Estado do Rio de Janeiro) com um grupo de pessoas acima de 65 anos, foi constatado que mais de 30% dos entrevistados haviam sofrido pelo menos uma queda nos últimos 12 meses (FAPERJ, 2015). A pesquisa confirma que as quedas em idosos têm como causa uma série de fatores externos e intrínsecos, associados, que vai do uso de medicamentos a pisos inadequados.

Os riscos envolvidos neste tipo de acidente tornam-se ainda maiores quando o idoso não possui uma companhia para lhe fornecer suporte. A demora no atendimento a uma pessoa acidentada pode representar a diferença entre a vida e a morte.

Segundo Maria Angélica Sanchez, presidente do Departamento de Gerontologia da Sociedade Brasileira de Geriatria e Gerontologia, apesar destes riscos, “o Brasil não se preparou para o envelhecimento de sua população e não tem estruturas adequadas para garantir dignidade e autonomia aos idosos” (EBC, 2015). Tal fato representa um problema da sociedade brasileira e precisa ser tratado.

Considerando a responsabilidade dos profissionais da área de tecnologia de utilizar seus conhecimentos no desenvolvimento de métodos e ferramentas que auxiliem a sociedade no tratamento de suas demandas, a situação apontada representa uma verdadeira oportunidade.

Kosir (2015) constatou que a tecnologia inteligente é, certamente, algo que será a chave para o funcionamento ideal da nossa sociedade futura, especialmente quando se trata de saúde e segurança. Com o desenvolvimento tecnológico dos últimos anos, os dispositivos eletrônicos estão se tornando cada vez mais práticos e populares.

Prova disto, é que a CES (*Consumer Electronics Shows*), maior feira de eletrônicos dos EUA, já em 2014 vislumbrava um futuro com expansão da computação vestível e da internet das coisas (FOLHA, 2014). As constatações permitem concluir que o mercado de desenvolvimento de tecnologias na área de suporte a pessoas é crescente e está aberto.

A demanda por novas ferramentas para atender à crescente população de idosos é cada vez maior. Considerando também os estudos que apontam previsões de crescimento significativo deste grupo em nossa sociedade nos próximos anos, o momento não poderia trazer melhor oportunidade para desenvolver um dispositivo móvel que, por meio de sensores, monitore as atividades de locomoção do idoso e proporcione maior segurança e bem-estar para o usuário.

Outro fator que contribui com a oportunidade de desenvolvimento do projeto é a alta disponibilidade dos componentes eletrônicos envolvidos na composição do sistema por um custo relativamente baixo.

1.2 Computadores vestíveis

Desde o início do século é crescente o aumento dos investimentos em estudos que possam desenvolver novas tecnologias para atender às necessidades humanas. Existe a previsão de que

o mercado de computação vestível movimentou US\$19 bilhões no mundo em 2018 (O GLOBO, 2013). As novas tecnologias estão cada vez mais presentes em nossa rotina e de maneira cada vez mais rápida, seja em casa, no trabalho ou nos momentos de lazer. O avanço tecnológico permite que novas tecnologias alcancem áreas nunca antes atingidas e velhos problemas sejam solucionados.

As últimas décadas trouxeram computadores mais potentes e cada vez menores, e atualmente carrega-se dentro do bolso máquinas mais ágeis do que aquelas de alguns anos atrás (COUTO, 2007). Essa evolução tecnológica permite hoje que se utilizem os “dispositivos vestíveis” abrindo uma série de oportunidades aos desenvolvedores de soluções tecnológicas nas mais diversas áreas.

Apesar de amplamente divulgado nos tempos atuais, o conceito de computação vestível não é novo. Em 1998, Steve Mann cunhou o termo num artigo intitulado *Definition of Wearable Computer* (MANN's, 1988, tradução nossa):

Um computador vestível é um computador que está localizado no espaço pessoal do usuário, controlado pelo usuário, e possui constância de operação e interação, ou seja, está sempre ligado e sempre acessível. Mais notavelmente, ele é um dispositivo que está sempre com o usuário, e permite que o usuário digite comandos ou os execute, enquanto anda ou faz outras atividades.

Ou seja, o dispositivo deve ter funcionamento ininterrupto e possuir alto grau de autonomia em relação às ações do usuário, de modo que não prejudique a realização de outras atividades cotidianas.

Uma visão geral dos principais desafios enfrentados pela prática da medicina é apresentada juntamente com a necessidade de soluções tecnológicas que podem “prevenir” os problemas (PARK; JAYARAMAN. 2003, tradução nossa). Neste contexto, diversas plataformas para sensores e dispositivos de monitoramento vêm sendo desenvolvidas para monitorar discretamente a saúde e o bem-estar dos indivíduos, seja de maneira direta ou remota, conforme exemplo a seguir.

Park e Jayaraman (2003) apresentaram uma camisa “inteligente” capaz de prevenir a Síndrome de Morte Súbita Infantil por meio de sensores que monitoram a atividade vital da criança. A

aplicação da tecnologia neste caso específico, além de garantir a segurança da criança monitorada, traz conforto aos seus cuidadores.

Esta e outras aplicações já estão disponíveis no mercado, como por exemplo, aparelhos corretores de postura, detectores de queda, monitoradores de respiração e frequência cardíaca, aparelhos que auxiliam pessoas a pararem de fumar e até mesmo monitores do sono (KOSIR, 2015).

Algumas destas tecnologias são direcionadas para fornecimento de suporte aos idosos por meio de serviços de monitoramento de suas condições e atividades (PORTO SEGURO, 2016). José Luis Silva, diretor da Porto Seguro Proteção e Monitoramento explica que o serviço de monitoramento dos idosos atende a uma demanda do mercado, detectada por pesquisas realizadas com a base de clientes da empresa e apoiado por dados do setor de saúde (PORTO SEGURO, 2016). Porém, apesar do desenvolvimento de novas ferramentas ser constante, ainda existem enormes lacunas carentes de soluções práticas e funcionais.

Atualmente já existem no mercado internacional algumas ferramentas de suporte e monitoramento do idoso, bem como serviços de acompanhamento continuado 24 horas por dia, sete dias por semana (UOL, 2017).

Dentre as diversas vertentes do uso da tecnologia este projeto estará focado no desenvolvimento e implementação de uma solução tecnológica que visa aumentar o grau de segurança e qualidade de vida dos seres humanos. Conforme já mencionado, a classe alvo escolhida é o grupo representados pelos idosos, pessoas com mais de 60 (sessenta) anos de vida que, devido à perda natural das capacidades de locomoção e reação, estão sujeitos a riscos de acidentes, necessitando por consequência de atenção especial.

O projeto está incluído no campo de dispositivos vestíveis que operam com sensores, como acelerômetros, que se comunicam com outros equipamentos e tem o objetivo de reforçar a segurança e aumentar o bem-estar do usuário.

1.3 Definição do problema

Este trabalho estará focado no desenvolvimento de um dispositivo eletrônico que traga maior segurança e auxilie o idoso em suas atividades rotineiras, com foco em momentos de emergência, permitindo que os seus cuidadores recebam mensagem de socorro ou que indiquem que o usuário está fora de condições de risco. O envio deste tipo de alarme pode reduzir o tempo de chegada de socorro em situações de emergência, reduzindo o risco de complicações físicas e até a morte do usuário.

O objetivo do projeto é o desenvolvimento do *hardware* e *software* de um sistema eletrônico microcontrolado que permita o monitoramento remoto de uma pessoa em ambiente doméstico, detectando situações de risco, como quedas, e realizando comunicação com uma segunda pessoa, que no caso seriam os cuidadores ou responsáveis.

As condições de contorno que delimitam o problema são as seguintes:

- ✓ O sistema deve possuir o maior grau de autonomia possível.
- ✓ O sistema deve possuir dispositivos de entrada para permitir o acionamento do usuário em casos de emergência e cancelamento de alarmes.
- ✓ O sistema deve ser capaz de detectar quedas.
- ✓ O sistema deve funcionar em todos os pontos da residência do usuário, portanto o alcance dos componentes de comunicação deve ser observado.
- ✓ A comunicação entre elementos do sistema deve ser sem fios.
- ✓ A confiabilidade das informações que trafegam entre os módulos do sistema deve ser garantida.
- ✓ O sistema deve possuir manutenção simples e requerer o menor nível de conhecimento técnico possível do usuário.

Para projetar um sistema eletrônico que atenda aos requisitos do usuário, na próxima seção deste trabalho, serão estudados os diferentes métodos de projeto de sistemas microcontrolados e será definida a estrutura mais adequada para o projeto entre as soluções disponíveis no mercado.

1.4 Metodologia

Uma metodologia de projeto de sistemas embarcados estabelece os passos a serem seguidos durante a realização do projeto e representa um fator crítico de sucesso no desenvolvimento de um projeto de dispositivo eletrônico envolvendo partes de *hardware* e *software* (Santos, 2006).

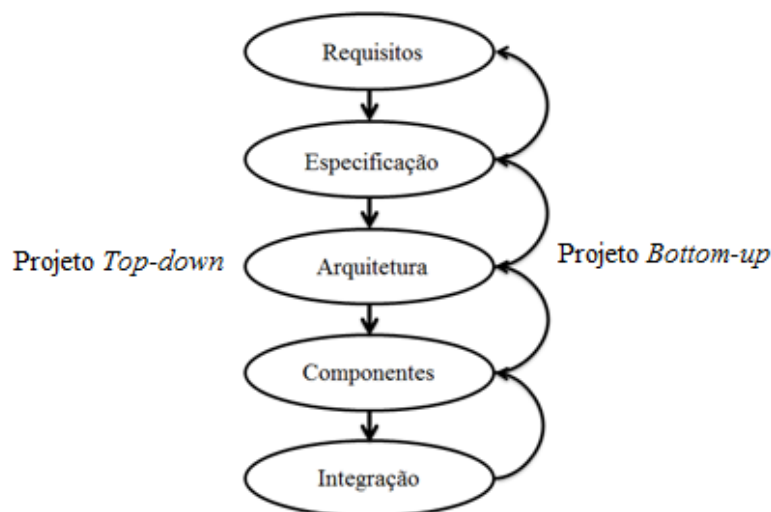
O estabelecimento de uma metodologia contribui com o cumprimento dos requisitos iniciais do projeto, bem como facilita o fluxo de seu andamento (Wolf, 2008). O projeto de sistemas embarcados em geral é dividido em uma série de etapas.

A Figura 1 ilustra as duas principais metodologias de desenvolvimento de um sistema embarcado segundo Wolf (2008). De acordo com Wolf (2008, p.12, tradução nossa), na abordagem de projeto *top-down* cada etapa pode ser resumida da seguinte maneira:

- a) Requisitos: É a primeira etapa. Nesta fase é criada uma descrição detalhada dos resultados que se planeja obter;
- b) Especificação: É descrito como o sistema irá se comportar, sem observar ainda como o mesmo será construído;
- c) Arquitetura: O detalhamento interno do sistema começa a ser construído, ainda sob o aspecto dos principais componentes;
- d) Componentes: Os componentes específicos de *hardware* e os módulos de *software* ainda não implementados são desenvolvidos como parte do projeto.
- e) Integração: No fim do projeto está a fase de integração, que envolve, além da construção completa dos sistemas, os testes dos componentes de *hardware* e de *software*. Nesta fase grande parte dos *bugs* é descoberta.

A outra alternativa de abordagem é a visão *bottom-up*, representada pelas setas laterais da Figura 1. Geralmente esta abordagem é utilizada para conhecer o funcionamento e as propriedades de alguns componentes antes do fim do projeto. A visão adotada no desenvolvimento do projeto em análise será a *top-down*.

Figura 1 - Etapas de desenvolvimento de um sistema embarcado.



Fonte: Wolf, 2008.

Santos (2006, p.20) ressalta que durante estas etapas, o projeto de sistemas embarcados deve considerar também os requisitos não funcionais como: (i) Custo de produção. (ii) Desempenho. (iii) Consumo de energia. (iv) Interface com o usuário. A cada etapa de refinamento do projeto, estes requisitos devem ser verificados para certificar que o sistema atende à especificação inicial.

Portanto, como o dispositivo proposto tem a finalidade específica de auxiliar as atividades de monitoramento e suporte ao idoso, o primeiro passo é estudar os requisitos do usuário. Nesta fase são realizadas entrevistas com idosos e seus cuidadores para melhor compreensão das demandas envolvidas.

Conhecendo-se as demandas do usuário, o próximo passo é levantar as opções tecnicamente viáveis de sistemas embarcados microcontrolados capazes de atender aos requisitos do usuário. A seguir, este grupo de soluções será filtrado utilizando-se os requisitos técnicos do projeto. Esta fase corresponde à etapa de especificação, detalhada anteriormente.

Após definição da estrutura da solução, o microcontrolador será definido e, posteriormente, serão listados quais módulos e componentes adicionais são necessários para compor a solução. Estas atividades se enquadram nas etapas de arquitetura e componentes. Por fim, após aquisição de todos os componentes, desenvolvimento do *software* e montagem do *hardware*, será

realizada uma série de testes com o primeiro protótipo e após, os requisitos da solução serão avaliados. Nesta fase será possível identificar alguns ajustes necessários para melhor desempenho da solução.

1.5 Estrutura do trabalho

Este trabalho está dividido em 6 capítulos, onde o primeiro deles apresenta o contexto de inserção do problema, as justificativas para o desenvolvimento do trabalho, os objetivos do mesmo, bem como sua metodologia e estrutura. No Capítulo 2, serão discutidos os aspectos da arquitetura do dispositivo eletrônico projetado. O Capítulo 3 mostrará o processo de desenvolvimento do *hardware*, enquanto o processo de desenvolvimento do *software* será apresentado no Capítulo 4.

Os resultados e as conclusões obtidos serão mostrados nos Capítulos 5 e 6, respectivamente. Em seguida serão apresentadas as referências do trabalho. O trabalho conta também com um anexo e um apêndice, onde será encontrado o código de programação desenvolvido.

2. MONITORAMENTO DE ATIVIDADES

Este capítulo definirá todos os requisitos técnicos necessários para atendimento dos requisitos do projeto e objetivos específicos, garantindo assim por consequência o correto funcionamento do sistema, e apresentará também a arquitetura do mesmo.

Assim, após conceituar os principais elementos teóricos do projeto será realizada uma intercontextualização entre os objetos, consolidando a arquitetura do sistema proposto.

2.1 Aquisição de informações

O sucesso do projeto está relacionado com a captação de informações significantes, seu devido tratamento e envio entre os dispositivos do sistema. Antes de tratar do processo de aquisição de dados em si serão apresentadas as grandezas que são os potenciais alvos de monitoramento.

Por tratar-se de um sistema com um dispositivo vestível que estará em contato direto com o usuário, e considerando o público-alvo definido no Capítulo 1, os seguintes eventos e variáveis são pontos de interesse deste projeto:

1. Situações de queda: Por meio da aplicação de um acelerômetro de três eixos é possível detectar variações abruptas de posição do usuário e consequentemente detectar quedas ou escorregões. A aquisição e interpretação destes eventos permite que, mesmo o usuário estando inconsciente, o cuidador seja comunicado sobre o ocorrido, permitindo adoção de medidas de socorro.
2. Situações de emergência: A disponibilização de um botão de pânico que envie um sinal de emergência em situações críticas pode permitir ao usuário a obtenção de uma resposta médica em menor tempo.

Considerando as condições de contorno deste projeto, expostas na Seção 1.3 deste documento, o sistema desenvolvido abrangerá os dois pontos de interesse abordados acima, sejam eles a capacidade de detectar quedas e o botão de pânico. Considerando a estrutura modular do sistema, no futuro, será possível realizar expansões de suas funcionalidades por meio da adição de novos módulos.

2.2 Arquitetura do sistema

Para alcançar o objetivo do projeto, nesta seção serão discutidas as alternativas de arquiteturas do sistema com maior capacidade de atendimento às necessidades deste projeto. As possibilidades de arquitetura levantadas para o sistema são as seguintes:

- Central e sensores instalados em pontos fixos pela residência do usuário: Esta solução seria composta de uma unidade central processadora de dados interligada a uma série de sensores espalhados pela casa do usuário. A detecção de presença poderia monitorar as atividades do usuário permitindo identificar sua localização e um possível caso de inatividade. Esta alternativa traz a vantagem de dispensar os dispositivos móveis e suas complexidades de projeto, porém é limitada sobre o ponto de vista de cobertura do sistema, uma vez que todos os ambientes monitorados deveriam possuir um sensor próprio. O usuário não teria que carregar consigo um dispositivo eletrônico, porém não teria à sua disposição todas as alternativas de monitoramento apresentadas na seção anterior.
- Central móvel integrada com os sensores com o usuário: Esta opção traz a possibilidade de todos os sensores estarem instalados com o usuário em uma plataforma vestível. Tal fato apresenta algumas vantagens em relação à última alternativa uma vez que a quantidade de sensores será significativamente menor, a limitação da cobertura fica restrita à capacidade do módulo de comunicação externa, o sistema integrado será composto de um único dispositivo, o que facilita o desenvolvimento e a manutenção, e dispensa a necessidade de módulos de comunicação entre os sensores e a central. A desvantagem é a necessidade de alimentação por baterias, elevado consumo de energia (devido ao número de módulos periféricos), complexidade de realizar a comunicação com os supervisores do funcionamento do dispositivo e o tamanho do conjunto composto pelo microcontrolador, os sensores e o módulo de comunicação externa.

Na tentativa de aproveitar-se das vantagens provenientes da segunda proposta de arquitetura sem sofrer limitações devido a suas desvantagens, foi estabelecido um terceiro modelo de arquitetura onde a central principal do sistema ficaria instalada em um ponto fixo da residência do usuário e interligada a uma central móvel mais simples, responsável por gerenciar os sensores e instalada em uma plataforma vestível que estará em contato direto com o usuário na

forma de uma cinta. Deste modo, o tamanho, o peso, o consumo de energia e a complexidade do microcontrolador do dispositivo móvel seriam reduzidos, trazendo viabilidade ao projeto. Este último modelo de arquitetura será o utilizado neste projeto. Seguindo a técnica de abordagem *top-down*, apresentada na Seção 1.4, serão estabelecidos os seguintes requisitos e especificações do projeto:

Requisitos gerais

- O sistema será composto de dois dispositivos, um equipamento de monitoramento do usuário, que transmitirá sinais de interesse para uma central responsável pela aquisição dos dados e comunicação com um usuário externo.
- O sistema deverá estabelecer uma sessão de comunicação entre o usuário e seu cuidador sempre que desejado ou caso detectada uma situação de emergência. O envio deste tipo de informação é popularmente chamada de botão de pânico, e tem a função de fornecer suporte aos idosos em situação de risco.

Unidade de sensoriamento remoto

- O dispositivo deverá possuir tecnologia vestível, ou seja, deve ser constituído de uma plataforma móvel que ficará em contato direto com o usuário monitorado.
- A unidade deve possuir baixo consumo de energia de modo a permitir sua alimentação por meio de uma bateria de pequeno porte.
- A unidade deve ser construída em uma plataforma de uso confortável para o usuário sem prejuízo ao funcionamento da mesma.
- O módulo móvel deve ser capaz de detectar variações abruptas de posição e quedas do usuário.
- O módulo não deve confundir a movimentação normal do usuário e gerar falsos alarmes.
- O módulo móvel também deve possuir um botão de emergência que permita ao usuário monitorado enviar sinais de pânico para a central.
- Além do botão de emergência, um botão para cancelamento de alarmes falsos deve estar presente.

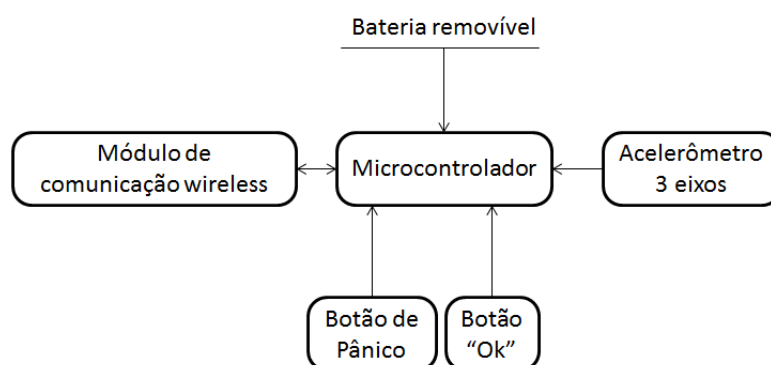
Central do sistema

- A central do sistema deverá possuir módulo de comunicação externa que permita o envio de informações para o usuário cuidador, externo.
- A central do sistema deve ser projetada para permitir a instalação em uma residência sem qualquer prejuízo à comunicação entre os módulos.

Portanto, para cumprir os requisitos do projeto e, considerando as possíveis atividades de monitoramento apontadas na Seção 2.1, a arquitetura do sistema será dividida em duas partes. A primeira parte corresponde ao dispositivo vestível ligado direto ao usuário e a segunda parte será uma central receptora dos dados, responsável pela comunicação com os usuários cuidadores. A arquitetura da primeira parte do sistema está representada na Figura 2:

Figura 2 – Arquitetura do dispositivo móvel ligado ao usuário.

Estrutura do dispositivo vestível



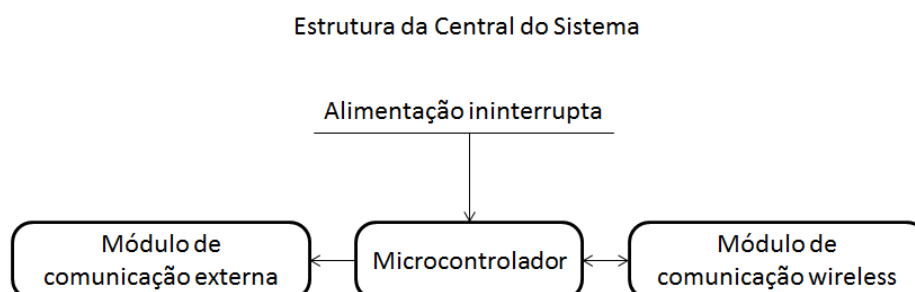
Fonte: Produção do próprio autor.

Observa-se que a estrutura do dispositivo vestível é composta basicamente por um microcontrolador alimentado por uma bateria removível, os sensores e um módulo de comunicação sem fio. O microcontrolador deve ser compatível com a aplicação vestível, ou seja, deve possuir baixo consumo de energia, tamanho reduzido e ser instalável em plataforma móvel. O sistema de alimentação por bateria removível deve proporcionar confiabilidade e uma autonomia adequada ao sistema. Os elementos sensores também devem possuir tamanhos e peso reduzidos mantendo compatibilidade com o microcontrolador.

A segunda parte do sistema, a central de recepção e tratamento das informações possui menores restrições de especificação técnica quando comparada ao dispositivo móvel, porém deve ter

capacidade de autenticar as informações e dar fluidez aos dados, garantindo a eficácia do sistema. A arquitetura da segunda parte do sistema é apresentada na Figura 3:

Figura 3 – Arquitetura da central do sistema instalada em ponto fixo.

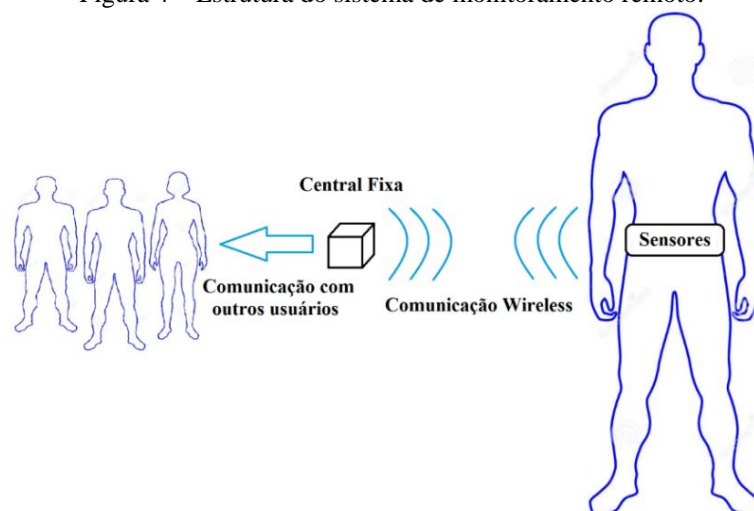


Fonte: Produção do próprio autor.

Como pode ser observado, um microcontrolador também é a base de sua estrutura, a alimentação vem de um ponto fixo de energia e a central possui a tarefa de receber as informações do dispositivo remoto, analisá-las e efetuar o repasse para os usuários cadastrados no sistema por meio de um módulo de comunicação externa.

Com a arquitetura do sistema definida, os dois próximos capítulos tratarão do desenvolvimento do *hardware*, abordando a especificação de componentes e suas características técnicas, e em seguida será discutido o processo de desenvolvimento do *software* e os requisitos de integração do projeto. A Figura 4 apresenta a estrutura do sistema proposto:

Figura 4 – Estrutura do sistema de monitoramento remoto.



Fonte: Produção do próprio autor.

3. DESENVOLVIMENTO DO HARDWARE

Este capítulo descreve o sistema integrado de monitoramento do usuário. Para um melhor entendimento do projeto, conforme explicado na Seção 1.4 – Metodologia, está sendo utilizada uma abordagem *top-down*, ou seja, de cima para baixo, sendo inicialmente explicado o *hardware* que compõe o sistema neste capítulo. No capítulo seguinte serão fornecidos maiores detalhes sobre o *software* do sistema.

3.1 Arquitetura de *hardware* do dispositivo vestível

Por tratar-se do principal elemento de controle e interface com o usuário, o desenvolvimento do *hardware* do dispositivo vestível é iniciado pela especificação do microcontrolador. Para realizar a escolha correta do microcontrolador que será usado no projeto é necessário analisar uma série de parâmetros e pré-requisitos como, por exemplo, a arquitetura, o consumo, os periféricos, as velocidades disponíveis e a capacidade de processamento, o tamanho e o custo.

Lima (2014) estabelece que o melhor microcontrolador é aquele que permite um desenvolvimento rápido de uma solução eletrônica segura conforme os requisitos de projeto e além disso, tenha disponibilidade no mercado, baixo custo, baixo tempo de entrega e que a equipe de desenvolvimento tenha experiência em sua operação. É importante também que esse item tenha longo prazo de fornecimento, para que o produto tenha uma longa vida no mercado e possa se pensar em um plano de manutenção e assistência técnica adequada.

Considerando os requisitos do projeto, apresentados na Seção 1.3, tem-se que o microcontrolador deste projeto deve:

- Ser compacto;
- Ter pouco peso em sua plataforma;
- Consumir pouca energia;
- Ser compatível com os periféricos que garantam as funcionalidades observadas na Seção 2.1;
- Programação simplificada;
- Custo acessível;
- Permitir sua instalação em estrutura vestível.

Por tratar-se de um conjunto de *hardware* que estará em contato direto com o usuário o último requisito de projeto tem importância fundamental. Foi escolhido então, um microcontrolador simples e de pequeno porte, amplamente difundido entre desenvolvedores do ramo acadêmico: o ATmega328P da Atmel. O ATmega328P é um microcontrolador de 8 bits, com arquitetura *Harvard* modificada, pertencente à família AVR da Atmel. Este microcontrolador opera na faixa de frequência entre 2 e 16MHz, pesa 28g (incluindo a plataforma) e tem baixo custo.

Neste projeto, para o módulo vestível do dispositivo, será utilizado o Arduino UNO R3 a versão integrada do microcontrolador com a plataforma de computação *open-source*. Esta plataforma é responsável por abrigar o microcontrolador ATmega328P e fornecer sinais de *clock* e alimentação, essenciais para seu funcionamento. O tamanho da plataforma, a disponibilidade, a facilidade de programação e a especificidade para prototipagem foram fatores decisivos para escolha desta plataforma. Na versão final do dispositivo os componentes estarão integrados em um *case* com função de proteção aos elementos de *hardware*. A Figura 5 mostra o módulo de desenvolvimento escolhido nesta etapa do projeto.

Figura 5 – Plataforma de desenvolvimento *open-source* Arduino UNO R3.



Fonte: Arduino AG (2017).

Tendo em vista as necessidades do projeto, a aplicação desta plataforma é adequada uma vez que, permite o desenvolvimento e os testes dos recursos exigidos pelo projeto, além de possibilitar a conversão do *hardware* em uma placa de circuito impresso mais compacta e a adição de novas funcionalidades no futuro. As especificações técnicas do microcontrolador são apresentadas na Tabela 1.

Tabela 1 – Especificações técnicas da plataforma Arduino onde o microcontrolador estaria instalado.

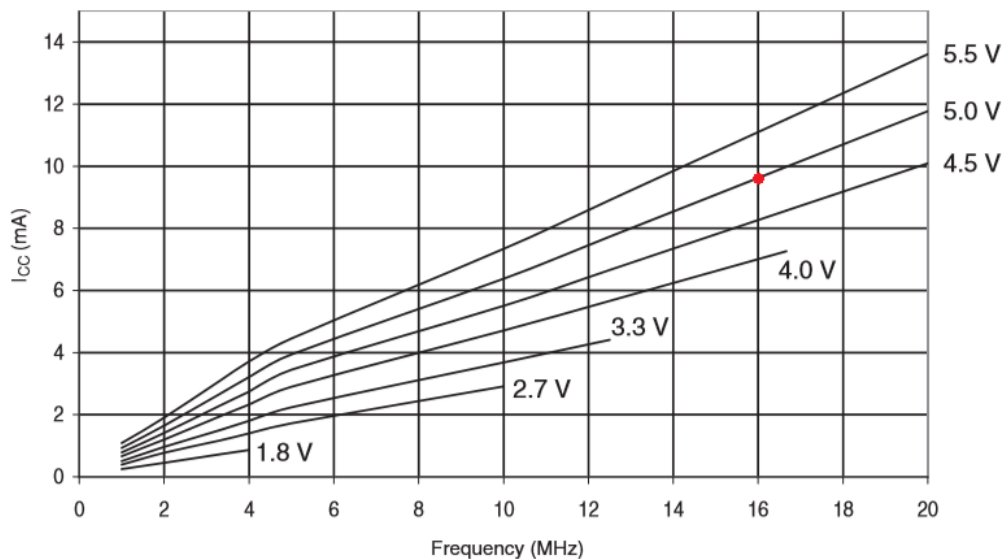
Parâmetro	Unidade
Microcontrolador	ATmega328P
Tensão de operação	5V
Tensão de entrada	7-12V
Pinos de Entrada/Saída digital	14
Pinos de Entrada/Saída analógica	6
Canais PWM	6
Corrente CC por pino E/S	20mA
Memória Flash	32kB
SRAM	2kB
EEPROM	1kB
Clock	16MHz

Fonte: Produção do próprio autor. Dados retirados do manual do fabricante ATMEGA(2015).

Além de atender os parâmetros técnicos do projeto o microcontrolador escolhido traz algumas versatilidades como a flexibilidade de projeto e a configuração simplificada. A plataforma utilizada possui um sistema de regulação de tensão integrado, que permite a alimentação por meio de uma bateria externa. Porém, considerando a possibilidade de aplicação do microcontrolador diretamente em uma placa de circuito impresso em uma etapa futura, deverá ser especificado um circuito próprio de alimentação. No protótipo deste projeto será utilizado o sistema de alimentação já integrado com a plataforma Arduino UNO R3.

Para o correto projeto do circuito de alimentação foram levantados os dados de consumo do microcontrolador no modo ativo e no modo ocioso. A frequência de operação padrão utilizada é 16MHz. Sabe-se, pela interpretação do gráfico da Figura 7, que a redução da frequência de operação pode reduzir o consumo do microcontrolador, porém neste momento o projeto será realizado considerando a frequência nominal. O nível de tensão de operação do microcontrolador é de 5,0 V. A Figura 6 apresenta as curvas de consumo de corrente no modo ativo para diversos níveis de frequência e tensão de alimentação, o ponto de operação do projeto está marcado no gráfico.

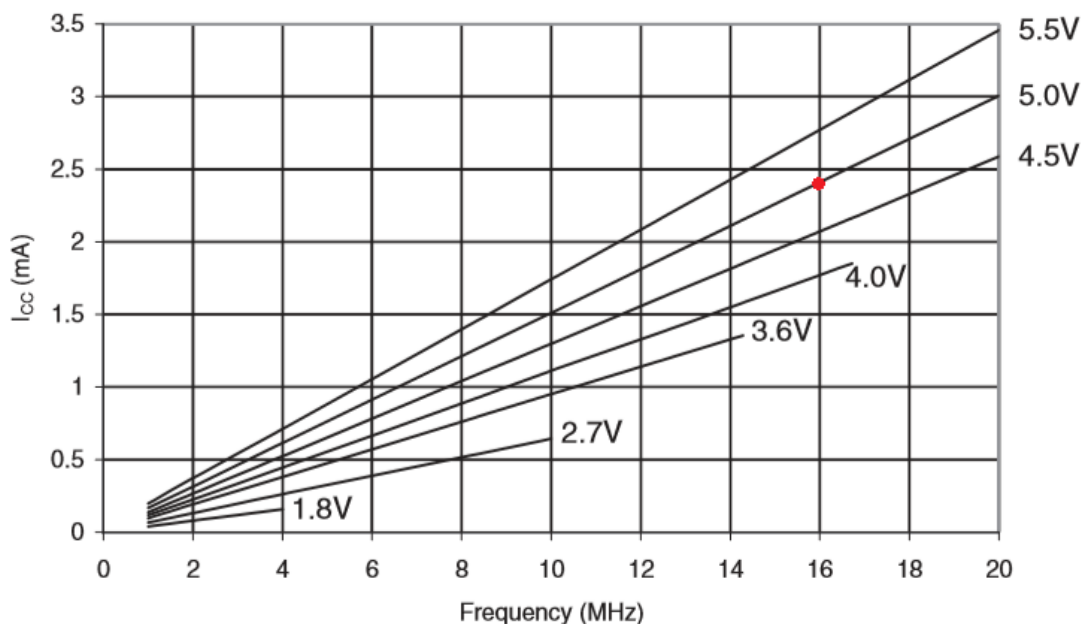
Figura 6 – ATmega328P: Corrente de alimentação ativa vs. Frequência de operação (1-20MHz).



Fonte: Manual do fabricante, ATMEGA (2015).

No modo ocioso pode-se observar pelo gráfico da Figura 7 que para os mesmos níveis de tensão e frequência o consumo de corrente é menor, como esperado.

Figura 7 – ATmega328P: Corrente de alimentação ociosa vs. Frequência de operação (1-20MHz).



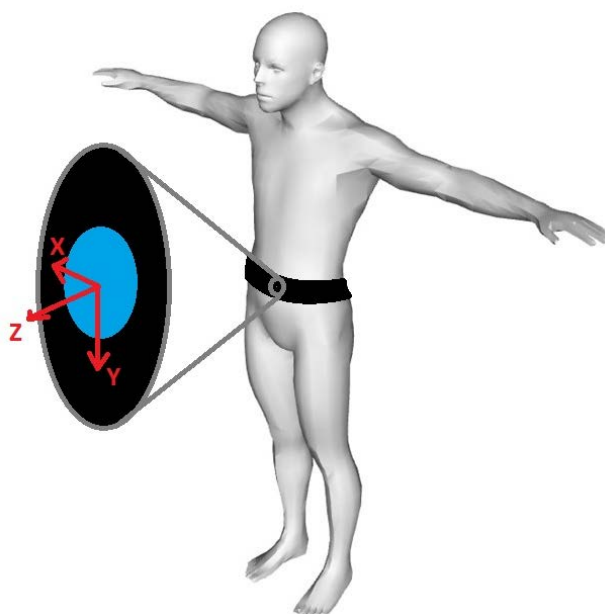
Fonte: Manual do fabricante, ATMEGA (2015).

Conforme apresentado nos gráficos, a corrente de alimentação ativa é aproximadamente 9,7mA e a corrente de alimentação ociosa 2,4mA. Com a base da plataforma definida, resta a tarefa de

especificar os módulos periféricos, como os sensores e os módulos de alimentação e comunicação. A seguir serão especificados os demais componentes do módulo vestível.

O acelerômetro tem a função de detectar as quedas do usuário. Neste projeto o algoritmo de detecção de quedas utiliza dois eixos do acelerômetro para realizar as detecções, – eixos ‘y’ e ‘z’ - permitindo a redução dos disparos falsos uma vez que fornecerão mais informações sobre a variação da posição do usuário, facilitando a identificação de situações reais de queda. O eixo ‘x’ não foi levado em consideração uma vez que, nos testes, os movimentos relacionados ao mesmo não estavam diretamente ligados a situações de quedas. Na Figura 8r tem-se a disposição física do acelerômetro no dispositivo.

Figura 8 – Posicionamento dos eixos do acelerômetro, representado pelo disco azul.



Fonte: Produção do próprio autor.

O próprio ponto de instalação, bem como a orientação do acelerômetro na cinta de tecido dispensa a calibração do sensor a cada uso, uma vez que sempre será utilizado na mesma posição.

A força de interesse é a gravidade, logo o medidor deve medir forças de até 1g. O acelerômetro de três eixos escolhido para este projeto foi o ADXL335. Este sensor tem pequenas dimensões, com cerca de 1,5cm de diâmetro, e baixo consumo de energia, que em situação de operação

chega a drenar $350\mu\text{A}$. A tensão de alimentação pode variar na faixa de 1.8V a 3.6V, sem qualquer prejuízo ao seu funcionamento. Existe uma versão com um regulador de tensão integrado que permite a alimentação em 5V. A escala está na faixa de $\pm 3,6\text{g}$, a não-linearidade do sensor é de $\pm 0,3\%$ da escala completa e a imunidade a ruído é boa, o que atende corretamente às demandas deste projeto (SPARKFUN, 2016). Este acelerômetro também possui compatibilidade com o Arduino, é costurável em tecido e de baixo custo.

Foram previstos dois botões integrados no dispositivo. O primeiro botão possui a função de pânico. Em situações de emergência, o usuário ao pressioná-lo envia um sinal de socorro para a central do sistema que por sua vez redireciona a mensagem a outro usuário cadastrado, um cuidador por exemplo. O segundo botão, denominado Botão “Ok”, tem a função de enviar uma mensagem informando que está tudo bem com o usuário. Pode ser pressionado para cancelar um alarme falso por exemplo.

Também foi prevista a utilização de um sinal sonoro para fornecer sinais sobre o funcionamento do dispositivo ao usuário, permitindo, por exemplo, que o usuário saiba se seu sinal de socorro realmente chegou à central responsável pelo envio da mensagem a terceiros.

Por fim, tem-se o módulo de comunicação *wireless*. Este componente tem o objetivo de enviar as informações colhidas dos sensores e botões para a central do sistema. Deve possuir tecnologia confiável e baixo consumo de energia. As seguintes tecnologias disponíveis no mercado foram estudadas com objetivo de identificar aquela que melhor se adequa às necessidades do projeto (SINGH, 2013):

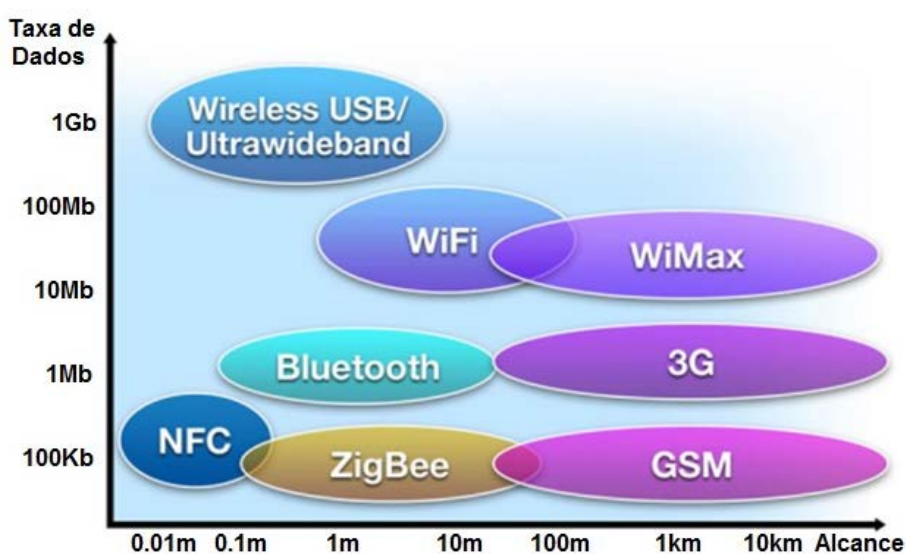
- *Bluetooth*: Possui a vantagem de ter segurança intrínseca. Sua restrição é que os módulos convencionais possuem o alcance limitado a 10m, o que prejudica sua aplicação em uma residência onde as distâncias entre os cômodos podem ser bem maiores. Também há necessidade do dispositivo estar pareado com a central receptora.
- *WiFi*: Apesar de possuir bom alcance (podendo chegar a 90m) e tamanho compacto, a transmissão de dados via WiFi depende da disponibilidade da rede e possui gerenciamento mais complexo que as outras tecnologias.
- *GSM*: O módulo GSM, apesar de não apresentar problemas de alcance, já que funcionaria nas áreas onde há sinal de celular, traz a necessidade de possuir um SIM

card habilitado para realização da comunicação, bem como possui preço elevado e alto consumo de energia em operação em comparação com as outras tecnologias.

- ZigBee: O módulo de comunicação Zigbee opera com alcance que pode atingir de 100 a 1000m dependendo da potência de transmissão. O preço e a aplicação em uma plataforma vestível são fatores limitadores. Estes módulos são ideais para composição de uma rede única com vários dispositivos.
- Rádio Frequência – 433MHz: A utilização de um módulo de comunicação RF de 433MHz, além de ser uma alternativa barata, de fácil instalação e ter alcance compatível com os requisitos do projeto (até 50m com presença de obstáculos), possui baixo consumo de energia e configuração simplificada. Sua desvantagem é a necessidade de implantação de algoritmos que melhorem a segurança da comunicação.
- Rádio Frequência – 2,4GHz: Este módulo RF com frequência de operação de 2,4GHz (antena embutida) possui a vantagem de permitir a comunicação bidirecional. Seu preço e consumo de energia são muito semelhantes aos módulos RF que operam na frequência de 433MHz porém seu alcance é ligeiramente menor devido a sua tensão de alimentação reduzida.

A Figura 9 apresenta um gráfico comparativo com as principais tecnologias de comunicação sem fio, mostrando uma relação entre a taxa de dados e o alcance de cada uma:

Figura 9 – Gráfico comparativo das principais tecnologias de comunicação sem fio.



Fonte: SINGH (2013).

O alcance necessário deste projeto está na faixa entre 10 e 100m logo, a tecnologia de comunicação NFC (*Near Field Communication*) é descartada de imediato uma vez que possui o alcance muito curto para o projeto. O Wireless USB é específico para taxas de dados muito elevadas, fora do escopo deste projeto. Considerando o alcance e a taxa de dados do WiMax, a aplicação da tecnologia geraria um sistema superestimado, aumentando o custo do projeto e reduzindo sua viabilidade.

Portanto, considerando as tecnologias estudadas e levando em consideração os outros requisitos do projeto, principalmente os que estão relacionados com a compatibilidade vestível, baixo custo e baixo consumo de energia, foram testadas inicialmente duas tecnologias para implantação no protótipo e realização de testes. São elas os módulos de comunicação RF de 433MHz e de 2,4GHz. O resultado dos testes do sistema em cada situação será analisado no Capítulo 5 – Resultados experimentais. O módulo que obteve o melhor desempenho foi o módulo de comunicação NRF24L01. Esse módulo é baseado no chip *Nordic Semiconductor* NRF24L01, que integra um transceptor RF de 2,4GHz *half-duplex*, ou seja, capaz de enviar e receber até 32 *bytes* de dados por vez com taxas de transmissão de até 2Mbps, porém não simultaneamente. Desenvolvido para aplicações sem fio (*wireless*), utilizando o protocolo *Enhanced ShockBurst* na sua camada de enlace de dados, é configurado através da interface SPI (*Serial Peripheral Interface* - Interface Periférica Serial) e contém até 128 canais que operam na faixa 2,4GHz (NORDIC SEMICONDUCTOR, 2006). As especificações do módulo são apresentadas na Tabela 2:

Tabela 2 – Especificações técnicas do módulo de comunicação NRF24L01.

Parâmetro	Unidade
Frequência de Operação	2.4-2.5GHz
Tensão de operação	1.9-3.6V
Modulação	GFSK
Máxima potência de saída	0 dBm
Máxima taxa de transmissão	2000 kbps
Consumo no modo TX @0 dBm	11,3mA
Consumo no modo RX @2000kbps	12,3mA
Consumo no modo <i>stand-by</i>	900nA

Fonte: Produção do próprio autor. Dados retirados do manual do fabricante NORDIC SEMICONDUCTOR(2006).

Este *transceiver* também pode ser utilizado em conjunto com um módulo regulador próprio que permite que a alimentação primária seja de 5V. O módulo também garante estabilidade nos níveis de alimentação, melhorando a eficiência da comunicação.

Conforme apresentado, foi definida a tensão de 5V para alimentação do microcontrolador e seus periféricos. Esta tensão é um parâmetro técnico do regulador de tensão utilizado. A fim de garantir estabilidade na tensão de alimentação do microcontrolador e maior eficiência na transmissão de potência entre a fonte e as cargas o regulador de tensão deve obedecer às especificações técnicas apresentadas na Tabela 3.

Tabela 3 – Especificações técnicas do regulador de tensão ideal para o projeto

Parâmetro	Unidade
Tensão de entrada	6V a 10V
Tensão de saída	5V
Corrente de saída	50mA
Corrente quiescente (50mA)	17 μ A
Corrente quiescente (<i>standby</i>)	1 μ A
Queda de tensão típica @25°C	48mV em 50mA
Temperatura de Operação na junção	-40°C a 125°C
Dissipação máxima de potência T<25°C	555mW

Fonte: Produção do próprio autor. Dados retirados do manual do fabricante *Texas Instruments* (2009).

O regulador de tensão deve ter baixa dissipação de potência de modo que não desperdice energia de aquecimento. O modelo recomendado para o projeto é o TPS770 da *Texas Instruments*. Este é um regulador de tensão linear com consumo ultrabaixo e baixa queda de tensão, adequado às necessidades deste projeto. Considerando os componentes especificados foi realizado o cálculo de consumo de energia do sistema. Os resultados são apresentados na Tabela 4.

Tabela 4 – Cálculo de autonomia do sistema

Ítem	Unidade
CONSUMO DOS COMPONENTES	mA
Microcontrolador modo ativo	9,7
Microcontrolador modo ocioso	2,4
Módulo de Comunicação modo RX	12,3
Módulo de Comunicação modo ocioso	0,9
Acelerômetro	0,35
Regulador de tensão (corrente quiescente)	0,017
CONSUMO DO SISTEMA	mA
Modo ativo	22,37
Modo ocioso	3,67

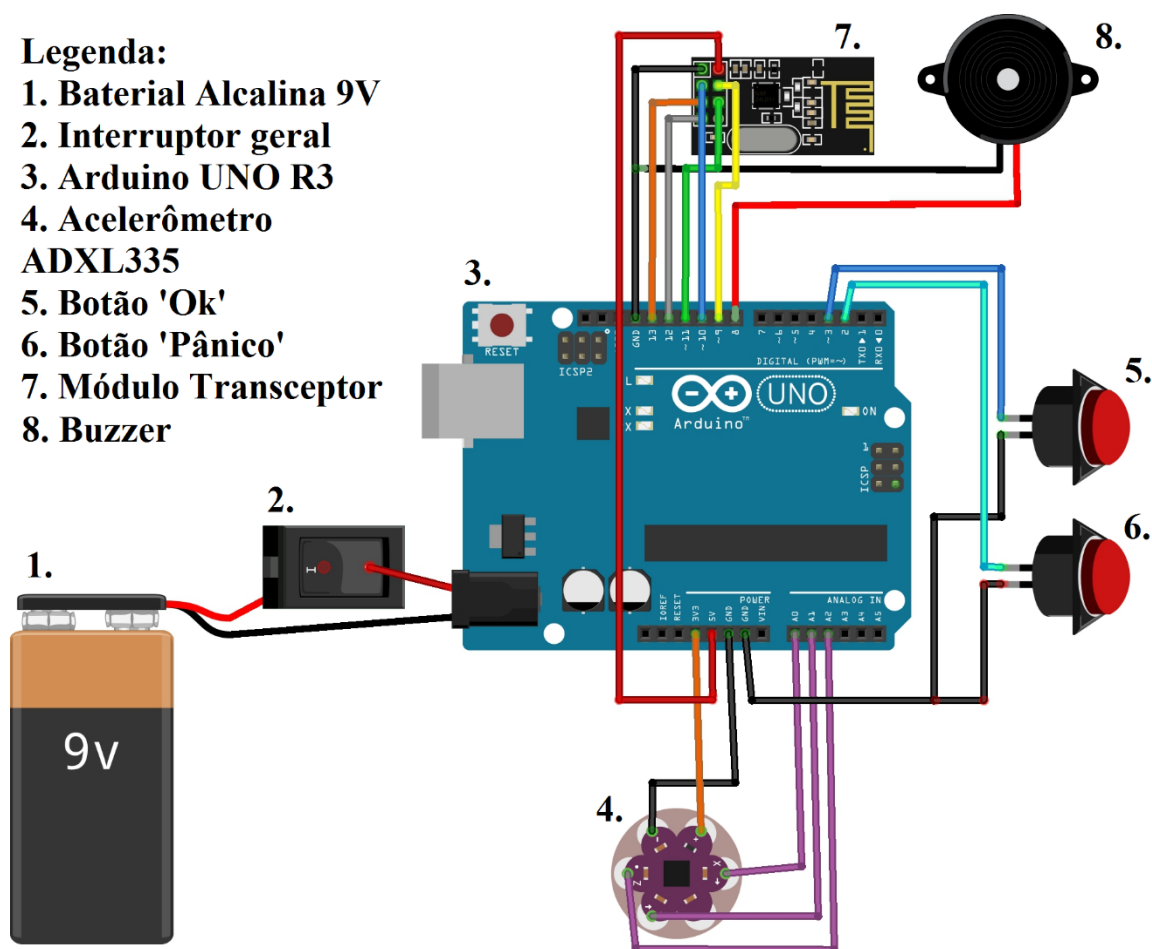
Fonte: Produção do próprio autor.

A capacidade da bateria deve permitir o uso ininterrupto do sistema por um período considerável, proporcionando maior autonomia ao dispositivo. A bateria escolhida tem tensão

nominal de 9V com capacidade de 500 mAh. Além de garantir boa autonomia ao dispositivo móvel, esta bateria é de fácil acesso no mercado e possui versão recarregável. Em modo ocioso a autonomia pode chegar a mais de cinco dias de funcionamento ininterrupto. Considerando a operação em modo ativo em 5% do tempo a autonomia ainda ultrapassa quatro dias.

Tendo em vista a especificação apresentada dos componentes do dispositivo móvel, seguindo a filosofia de projeto, a arquitetura de *hardware* do dispositivo vestível proposta é apresentada na Figura 10.

Figura 10 – Arquitetura de *hardware* do dispositivo vestível.



Fonte: Desenhado por meio do software *Fritzing* v0.9.3. Produção do próprio autor.

Ressalta-se que as imagens dos componentes estão fora de escala para permitir melhor visualização. O trajeto dos condutores também foi representado de maneira didática.

Conforme apresentado na Figura 9, resumidamente tem-se que o sistema é composto por uma bateria alcalina (1), um interruptor geral do sistema (2), a plataforma Arduino UNO R3 com microcontrolador Atmega328P (3), o acelerômetro de três eixos ADXL335 (4), dois botões com funções específicas (5 e 6), o módulo de comunicação NRF24L01 (7) e o *buzzer* (8). Todos estes elementos estarão instalados em uma *case* protegidos contra quedas e com uma disposição que facilita o uso pelo usuário.

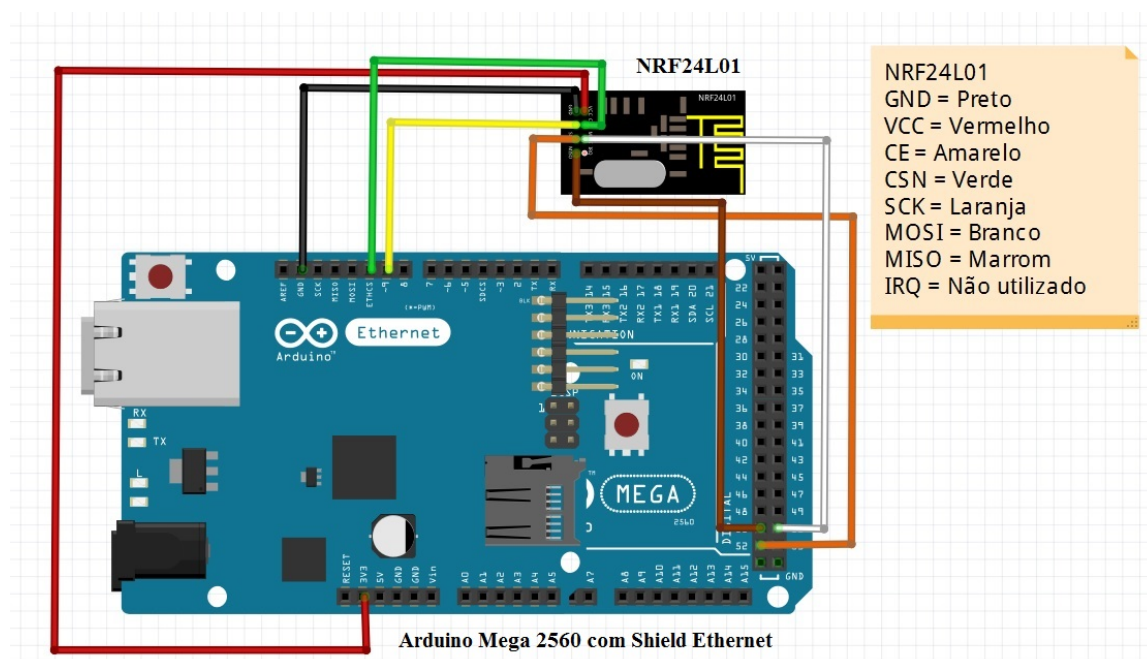
3.2 Arquitetura de *hardware* da central do sistema de monitoramento

Por se tratar de um conjunto de *hardware* que estará instalado em um ponto fixo da residência do usuário, este equipamento não possui as mesmas restrições que o dispositivo móvel.

O foco na especificação deste dispositivo é garantir que o mesmo esteja sempre apto para receber os sinais da plataforma vestível, tratar as informações e gerar os alarmes necessários para os usuários cuidadores cadastrados.

Logo, a arquitetura de *hardware* da central microcontrolada do sistema é mostrada na Figura 11.

Figura 11 – Arquitetura de hardware da central do sistema.



Fonte: Produção do próprio autor.

Assim como a Figura 10, os componentes e condutores da Figura 11 estão fora de escala porque a imagem possui caráter didático para permitir melhor visualização. O sistema é basicamente composto de uma plataforma Arduino Mega 2560, com microcontrolador ATmega2560, um *shield* Ethernet W5100 e um módulo *transceiver* NRF24L01 de 2,4GHz.

O microcontrolador Arduino Mega 2560 foi escolhido por atender todas as necessidades técnicas do projeto e por questões de disponibilidade, evitando assim a compra de outro microcontrolador. O que não significa que um microcontrolador mais simples não pudesse atender às demandas técnicas sem prejuízo à qualidade do sistema proposto.

O módulo receptor e transmissor RF de 2,4GHz é idêntico ao instalado na plataforma móvel junto ao usuário. A comunicação é bidirecional e configurada via *software*. Cabe ressaltar que ambos os microcontroladores utilizados no projeto são compatíveis com todos os itens monitoráveis apresentados na Seção 2.1 e com as tecnologias de transmissão de dados expostas na última seção.

Entre as soluções possíveis para estabelecer comunicação entre a central do sistema e os usuários cuidadores cadastrados estão as seguintes:

- E-mail;
- Chamada telefônica;
- SMS;
- Aplicativos de comunicação (*WhatsApp, Twitter, Facebook* e outros).

Neste projeto o *shield* Ethernet W5100 foi aplicado ao projeto com objetivo de permitir que os usuários cuidadores sejam comunicados via internet por meio da plataforma *Twitter*. O *Twitter* é uma rede social composta de um servidor para *microblogging*, uma forma de informação de *blog* que permite aos usuários atualizações breves de imagens e texto. Por meio do *Twitter* a central consegue mandar mensagens diretamente para um grupo de usuários cadastrados, atendendo aos objetivos do projeto. Maiores detalhes sobre o processo de trânsito da informação serão abordados no Capítulo 4 – Desenvolvimento do Software.

4. DESENVOLVIMENTO DO SOFTWARE

Este capítulo dá seguimento à apresentação dos detalhes do sistema integrado de monitoramento do usuário. Nesta etapa serão discutidos os requisitos de integração e detalhes do software seguindo a abordagem *top-down*.

O *software* embarcado do sistema é responsável pelo aproveitamento das funcionalidades do *hardware*, bem como da concretização dos objetivos propostos do projeto. A análise será dividida em duas partes. A primeira tratará do *software* do dispositivo vestível e a segunda cuidará do *software* da central do sistema. Será possível observar que, apesar de possuírem estruturas diferentes, ambos os *softwares* mantêm certo grau de similaridade.

A fim de facilitar a compreensão das estratégias selecionadas para o projeto foram elaborados alguns fluxogramas para explicação do código, que está disponibilizado como apêndice deste trabalho.

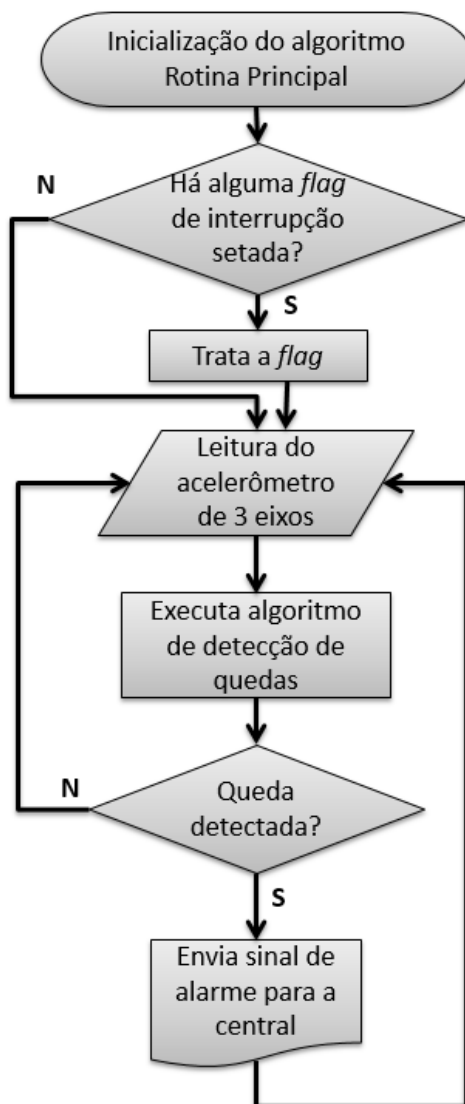
4.1 Arquitetura de *software* do dispositivo vestível

O dispositivo móvel tem a tarefa de detectar qualquer situação de risco e realizar a comunicação imediata com a central. Portanto, o algoritmo que originou seu código foi elaborado em torno deste requisito.

A unidade deve monitorar constantemente o sensor de aceleração, com objetivo de detectar quedas e, em segundo plano, deve atender qualquer chamada dos botões. É importante que o usuário cuidador receba uma comunicação em períodos de tempo a fim de ter a certeza de que o sistema está operante. Para tratar estes requisitos de *software* foi elaborada uma rotina com o algoritmo de detecção de quedas no *loop* principal e as demais tarefas serão tratadas como interrupções, permitindo assim que o sistema tenha capacidade de cumprir todas as suas funções.

A Figura 12 apresenta o fluxograma da rotina principal do código.

Figura 12 – Fluxograma da rotina principal do dispositivo vestível.



Fonte: Produção do próprio autor.

Conforme já mencionado, a rotina principal do código é responsável pela leitura do acelerômetro e execução do algoritmo detector de quedas. O microcontrolador realiza a leitura dos dados analógicos do acelerômetro instalado na cinta presa ao usuário e realiza a tradução da posição. Para detectar situações de queda foram testados dois algoritmos diferentes. O primeiro identificava as situações de queda analisando a velocidade da variação da inclinação do usuário através do acompanhamento dos valores lidos dos três eixos do acelerômetro. O método apresentou bons resultados, porém alguns movimentos abruptos geraram alarmes falsos, e por isso foi desenvolvido um segundo algoritmo.

O que o novo algoritmo faz é medir a desaceleração do corpo do usuário. Durante uma queda o corpo sofre a aceleração da gravidade por certo período e em sequência experimenta uma desaceleração em um período consideravelmente menor. O acelerômetro é o sensor ideal para medir esta desaceleração. Foram escolhidos dois eixos específicos do acelerômetro e foi realizada uma série de testes a fim de definir-se o *set point* do algoritmo para garantir um gatilho confiável para o sistema de detecção de quedas. Logo, foram definidos como interessantes os eixos ‘y’ e ‘z’, que receberam valores de *set point* próprios. Caso o valor lido de um desses eixos ultrapasse o limite é gerado um alarme de queda. O eixo ‘x’ não foi levado em consideração uma vez que, durante os testes foi percebido que os movimentos nestes eixos não estão diretamente ligados a uma queda.

O algoritmo aqui desenvolvido guarda certa simplicidade, uma vez que o foco do projeto é a elaboração de um sistema que atenda todos os requisitos pré-definidos. A preocupação é garantir que a plataforma seja compatível com um futuro algoritmo de detecção mais complexo e robusto.

Fora da rotina principal do código estão as rotinas de interrupção por *software* e por *hardware*. A primeira rotina de interrupção é uma rotina de interrupção por tempo que tem o objetivo de realizar um teste de comunicação entre os dispositivos em intervalos de tempo pré-definidos, indicando ao usuário cuidador uma possível falha de comunicação do sistema. Portanto, quando o contador de tempo da interrupção se esgota, o *software* vai para uma breve rotina de interrupção, onde indica a necessidade de realizar um teste de comunicação (realizando um contato com a central para isto), reseta o contador e retorna para o código principal. No código principal o teste é realizado sem prejudicar a leitura da posição. Um sinal então é enviado com o objetivo de informar à central que o sistema está operando sem qualquer alteração. A central recebe este sinal e envia outro de autenticação, confirmando a capacidade de comunicação. Caso o dispositivo móvel não receba este retorno em tempo hábil, é gerado um sinal de alarme. Esta rotina é uma importante ferramenta de auto-diagnose do sistema.

Além da interrupção por tempo foram implementadas também duas outras rotinas de interrupção. Diferentemente da rotina recém-apresentada, estas são interrupções por *hardware*.

Foram instalados dois botões na cinta. O primeiro botão tem a função de emitir um sinal de pânico caso o usuário esteja sob uma situação de risco e deseje pressioná-lo, já o segundo botão tem o objetivo oposto, indicar que está tudo sob controle, permitindo inclusive o cancelamento de um alarme falso. O repasse destas informações para o usuário cuidador é realizado pela central do sistema.

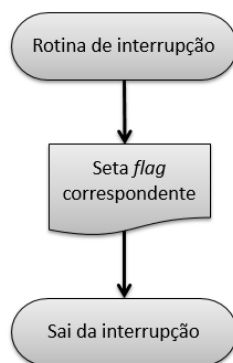
A utilização de botões e interrupções de *hardware* trazem consigo dois problemas relacionados, o efeito *bouncing* e o *jitter*. O *bouncing* está relacionado com a instabilidade do sinal na saída de uma chave devido a trepidações oriundas geralmente da alteração de sua posição (VARGAS, 2016). Já o *jitter* é um fator que reduz a qualidade da transmissão dos dados. Trata-se do erro do tempo entre os sinais elétricos digitais, o transiente elétrico da mudança de um estado para o outro (MATIAS, 2012).

Para definição da melhor estratégia de tratamento do efeito *bouncing* dos botões e da redução do *jitter*, foi realizado um estudo de caso do uso dos botões. O sistema deve funcionar da seguinte maneira:

- Caso o usuário pressione o botão com a função de Pânico, uma *flag* de pânico é imediatamente setada.
- Se durante os próximos cinco segundos o usuário pressionar o botão com a função “Ok” o alarme é cancelado e uma *flag* para envio de um sinal de “Ok” é setada.
- Caso o usuário pressione o botão com a função “Ok”, uma *flag* correspondente é imediatamente setada.
- Se durante os próximos cinco segundos o usuário pressionar o botão de Pânico o sinal de “Ok” é cancelado e uma *flag* de pânico é setada.
- Caso uma *flag* permaneça setada por cinco segundos seguidos o sinal correspondente é enviado para a central fixa do sistema.
- O sistema inicia a contagem de tempo e tomadas de decisão a partir do primeiro acionamento de cada botão. As interrupções são ativadas em nível baixo, especificamente na borda de descida. Acionamentos repetidos após o disparo da contagem não têm efeito, neutralizando as implicações do *bounce*.
- A rotina de interrupção é dedicada para o *set* das *flags*, o tratamento do sinal é realizado no código principal, reduzindo o *jitter*.

Na Figura 13 está representado o fluxograma dos algoritmos de interrupção do código.

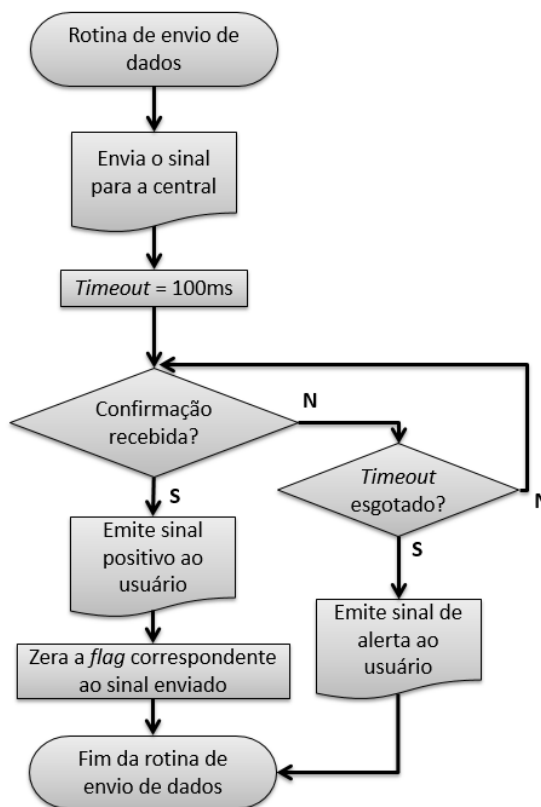
Figura 13 – Fluxograma da estrutura das interrupções do dispositivo vestível.



Fonte: Produção do próprio autor.

As duas rotinas são bem semelhantes entre si. A única diferença é a função ativada pelo botão. Apesar do envio e do recebimento dos sinais estarem representados dentro da rotina de interrupção, na fase de testes tais tarefas serão realizadas dentro da própria rotina principal do código de modo que o envio dos sinais não prejudique a rotina de leitura do acelerômetro e a detecção das quedas. A rotina de envio de dados é basicamente a apresentada na Figura 14.

Figura 14 – Fluxograma da rotina de envio de dados.



Fonte: Produção do próprio autor.

Nesta etapa de desenvolvimento do *software* a maior preocupação era garantir que este sinal seja gerado, transmitido e ter certeza de seu recebimento. O processo de transmissão de dados, representado por meio do fluxograma da Figura 14, tem o objetivo de traduzir as *flags* geradas nas interrupções para mensagens codificadas, que serão enviadas para a central fixa do sistema. Inicialmente a *flag* é traduzida em uma mensagem, que pode caracterizar uma situação de queda, emergência, normalidade ou teste de comunicação. A mensagem codificada é enviada pelo transceptor configurado como transmissor e logo em seguida o mesmo é reconfigurado para receptor, quando, a partir de então, aguarda por um determinado período pela confirmação de recebimento da central fixa, este período é chamado de *timeout*. Em paralelo à esta situação, o transceptor da central fixa, configurado como receptor, após receber uma mensagem, realiza a decodificação da mesma, setando uma *flag* e reconfigura-se com a função de transmissor. Em seguida, a central aguarda um tempo para que o transceptor do dispositivo móvel chaveie de transmissor para receptor e envia um sinal de confirmação de volta, acusando o recebimento da mensagem. O dispositivo vestível recebe esta resposta e acusa o recebimento para o usuário. Caso esta confirmação não seja recebida no tempo estipulado, no caso igual 100ms, é gerado um alarme sobre a falha de comunicação

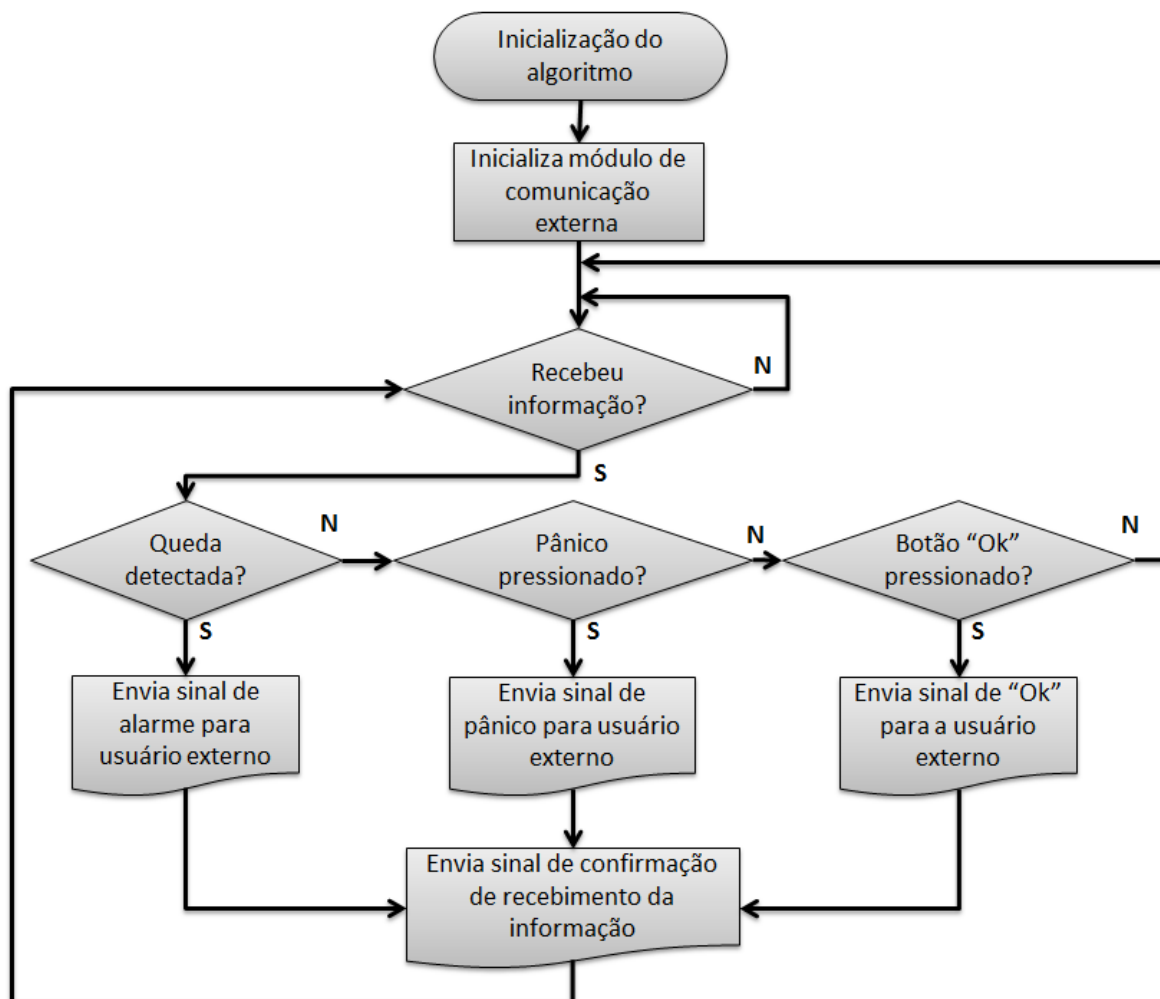
Conforme apresentado no Capítulo3, o módulo bidirecional NRF24L01 foi escolhido no lugar do módulo de comunicação RF de 433MHz, uma vez que este último não possui comunicação bidirecional, o que traz complexidade à tarefa de obter a acusação de recebimento por parte da central. A fim de comprovar as vantagens da escolha, foram realizados alguns testes com estes componentes. Os resultados serão apresentados e discutidos no Capítulo 5 – Resultados experimentais.

4.2 Arquitetura de software da central do sistema de monitoramento

A central do sistema possui a tarefa de monitorar constantemente o recebimento de um sinal proveniente do módulo vestível, realizar o repasse de informação aos usuários cuidadores cadastrados e emitir uma confirmação ao usuário.

O fluxograma do algoritmo da central está representado na Figura 15. É possível notar o direcionamento de sua estrutura para o repasse de informações.

Figura 15 – Fluxograma da rotina principal da central fixa do sistema.



Fonte: Produção do próprio autor.

Por meio do módulo de comunicação RF as informações são recebidas. Assim que são identificadas, é gerado o envio de um sinal de confirmação para o dispositivo móvel e, posteriormente, o módulo de comunicação externa recebe a ordem para transmitir a informação.

Conforme informado na Seção 3.2, o módulo escolhido para realizar esta tarefa foi o *shield* Ethernet W5100. Este componente baseia-se no chip WIZnet *Ethernet* W5100 que fornece acesso à rede IP (*Internet Protocol*) nos protocolos TCP ou UDP. Neste projeto o módulo é interligado a uma rede IP doméstica e configurado via protocolo DHCP (*Dynamic Host Configuration Protocol*), ou seja, recebe um endereço de IP válido automaticamente pela própria rede.

Após estabelecimento da conexão com a internet, por meio do protocolo OAuth e de um aplicativo dos servidores do *Twitter* o sistema consegue realizar postagens no *blog*. O OAuth é um protocolo de autorização para API's (*Application Programming Interface*) *web* voltado a permitir que aplicações *client* acessem um recurso protegido em nome de um usuário sem necessidade de disponibilização da senha padrão. Para isso é necessário que seja gerado um *token* de acesso que fará a autenticação do sistema que deseja realizar a postagem. Neste caso, o *token* é um passe de acesso temporário que permite a utilização da conta do *Twitter* pelo servidor de postagem para publicação de mensagens. Este *token* é gerado por meio de uma aplicação aberta disponível no próprio site do Arduino (Arduino, 2012).

Foi criada uma conta no *Twitter* de uso exclusivo do sistema e, por meio da configuração apresentada, a central consegue fazer *posts* na rede social com os detalhes das informações que geraram o alarme, indicando o problema ocorrido, a hora e podendo inclusive fazer menções às contas dos usuários cuidadores com objetivo de permitir que eles recebam uma notificação instantânea a cada *post*.

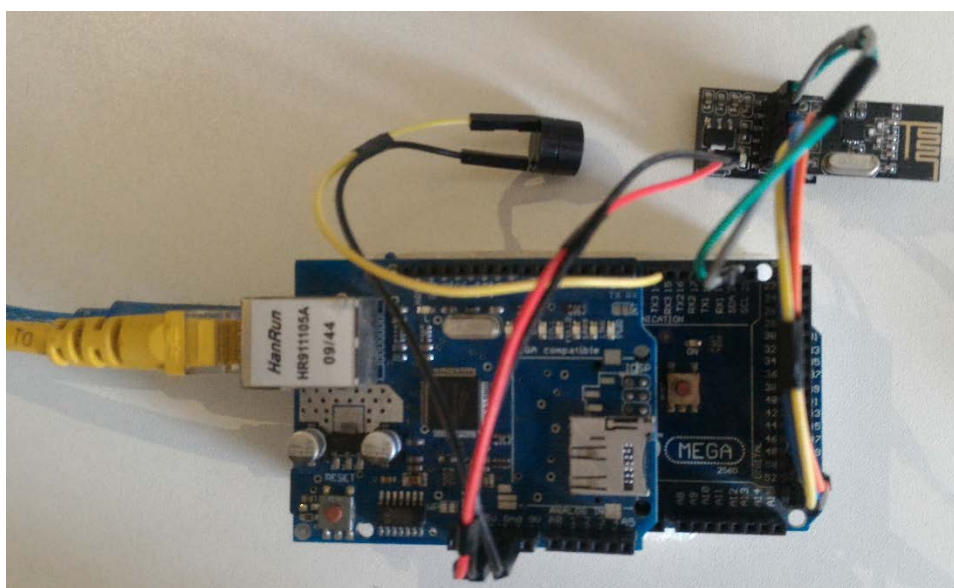
Neste capítulo foi apresentada a estrutura dos algoritmos dos dois dispositivos que compõem o sistema. Os microcontroladores foram programados por meio da linguagem de programação C. O *software* embarcado possui detalhes técnicos peculiares aos elementos de *hardware* do projeto e encontra-se disponível comentado no Apêndice I deste trabalho. No capítulo seguinte serão tratados os resultados experimentais obtidos com as arquiteturas de *hardware* e *software* utilizadas.

5. RESULTADOS EXPERIMENTAIS

Neste capítulo serão apresentados e discutidos os resultados experimentais obtidos na montagem do projeto, assim como serão apresentadas algumas sugestões para melhorias futuras. Inicialmente foi desenvolvido um protótipo com a estrutura de *hardware* e *software* apresentada nos dois últimos capítulos.

Na Figura 16 tem-se a imagem do protótipo da central do sistema:

Figura 16 – Protótipo da central do sistema de monitoramento remoto.



Fonte: Produção do próprio autor.

O dispositivo para testes possui exatamente a mesma topologia de *hardware* apresentada na Seção 3.2, sendo composta de um Arduino Mega 2560, um *shield* Ethernet W5100 e um módulo de Rádio Frequência NRF24L01 operando em 2,4GHz. Nesta versão foi instalado uma pequena sirene para emissão de sinais de confirmação e alguns pinos de ligação do módulo transmissor foram alterados. A central foi devidamente alimentada e interligada à rede IP da residência onde foram realizados os testes.

Na Figura 17 é apresentada a imagem do protótipo do dispositivo vestível.

Figura 17 – Protótipo do dispositivo vestível.



Fonte: Produção do próprio autor.

O protótipo do dispositivo vestível foi instalado em uma *case* de plástico adaptada com uma tira elástica de tamanho ajustável, permitindo o uso na posição da cintura do usuário. O sistema é alimentado com uma bateria alcalina removível, cuja tensão vale 9V. Também foi instalado um interruptor para ligar e desligar o circuito.

Para permitir o correto funcionamento do módulo de comunicação externa foi criada uma conta de usuário no *Twitter* com codinome @Panic_Report. Esta conta foi configurada e é utilizada para envio das mensagens originadas pelo sistema.

Com base em todas as especificações de projeto e desenvolvimento de elementos de *hardware* e *software*, nos testes espera-se que:

- o dispositivo móvel funcione em todos os pontos da residência onde será testado;
- o dispositivo móvel detecte variações abruptas de posição que configurem quedas do usuário;
- o dispositivo móvel seja de uso confortável ao usuário;
- o dispositivo móvel seja capaz de enviar o sinal de acionamento dos botões;
- a central do sistema esteja apta para receber todos os sinais do dispositivo móvel;
- a central do sistema seja capaz de gerar alarmes via *Twitter* em tempo adequado;
- o sistema não gere falsos alarmes durante a duração dos testes.

Também foram testadas, além da capacidade de detecção de quedas e leitura dos botões, o atraso da comunicação entre os elementos, o alcance do dispositivo móvel e a comunicação com o servidor externo.

5.1 Resultados dos testes de detecção de quedas

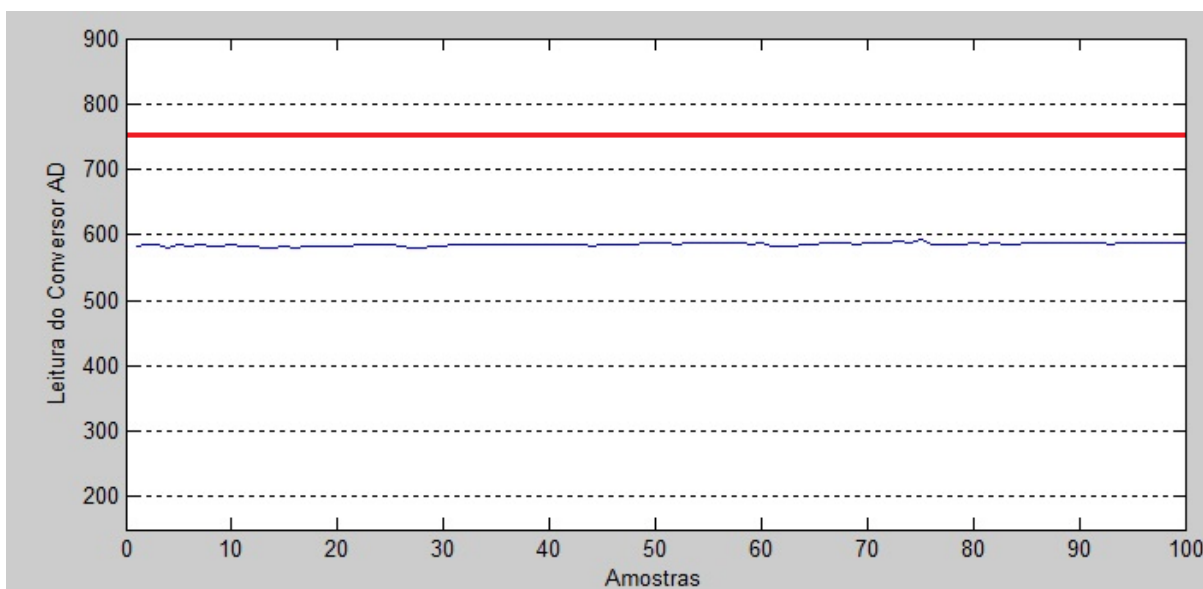
Para testar a capacidade do dispositivo móvel de detectar quedas e realizar a captação do sinal de interesse, o dispositivo foi instalado em um usuário e interligado ao computador via interface serial USB. No computador, através do uso do software MATLAB® os sinais foram lidos ao longo do tempo, permitindo a composição de diversos gráficos para cada situação.

O sinal transmitido pela porta serial é o valor instantâneo de um eixo específico do acelerômetro instalado no dispositivo. Neste teste foram lidos os valores de saída do eixo 'y' do acelerômetro, o principal eixo relacionado às quedas. A leitura foi realizada a cada 40ms por questões de estabilidade, uma vez que o sensor possui um pequeno atraso na atualização dos dados. Cabe registrar que, o sinal transmitido pelo microcontrolador é proveniente da saída de seu conversor Analógico-Digital (AD), responsável por enquadrar o sinal analógico, que pode variar entre 0V e Vcc, em uma faixa digital que varia de 0 a 1023. Esta faixa estará representada no eixo vertical dos gráficos de análise do sinal, apresentados a seguir.

Conforme explicado na Seção 4.1 deste documento, a queda é detectada através da comparação deste sinal originado pelo acelerômetro e traduzido pelo conversor AD com um *set point* pré-definido. Desacelerações abruptas caracterizam uma queda.

A linha horizontal no valor 750 representa o *set point* anteriormente mencionado. Valores acima desta linha representam uma situação de queda. Na Figura 18 até a Figura 28 são apresentados os sinais obtidos do eixo 'y', principal eixo de variação da aceleração durante situações de queda. O eixo 'z' será usado como eixo complementar, para detectar situações de queda em que o eixo principal não é tão sensível. A seguir será apresentada uma série de ensaios com a resposta do sistema para diversas situações do dia-a-dia do usuário. Por exemplo, o gráfico da Figura 18 foi obtido enquanto o usuário encontrava-se de pé e parado.

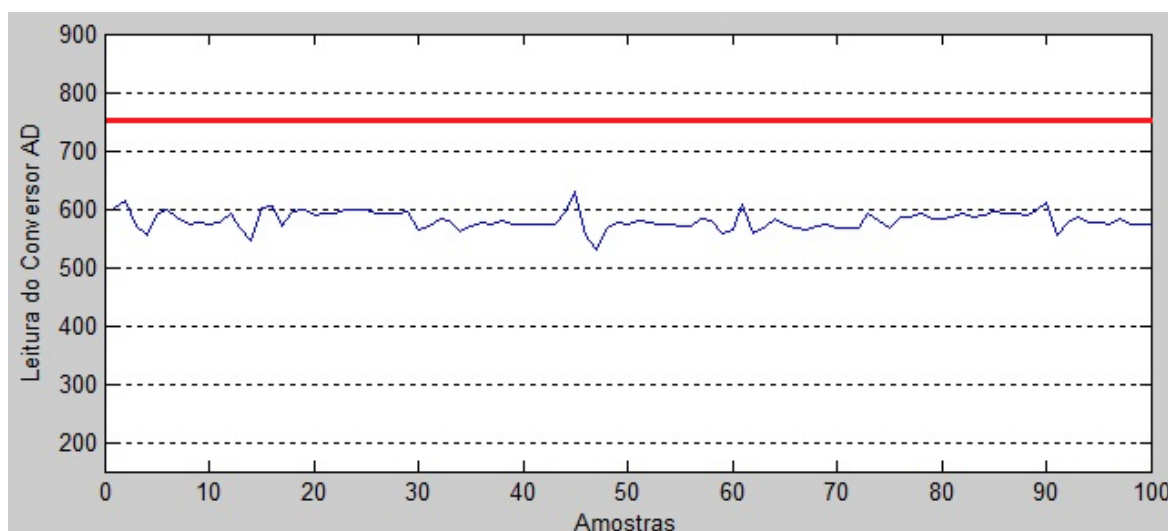
Figura 18 – Sinal de saída do eixo 'y' do acelerômetro obtido com o usuário de pé e parado.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

Pode-se observar que quando não há movimento o sinal de interesse permanece praticamente constante próximo a 600. O gráfico da Figura 19 foi obtido quando o usuário caminhava lentamente.

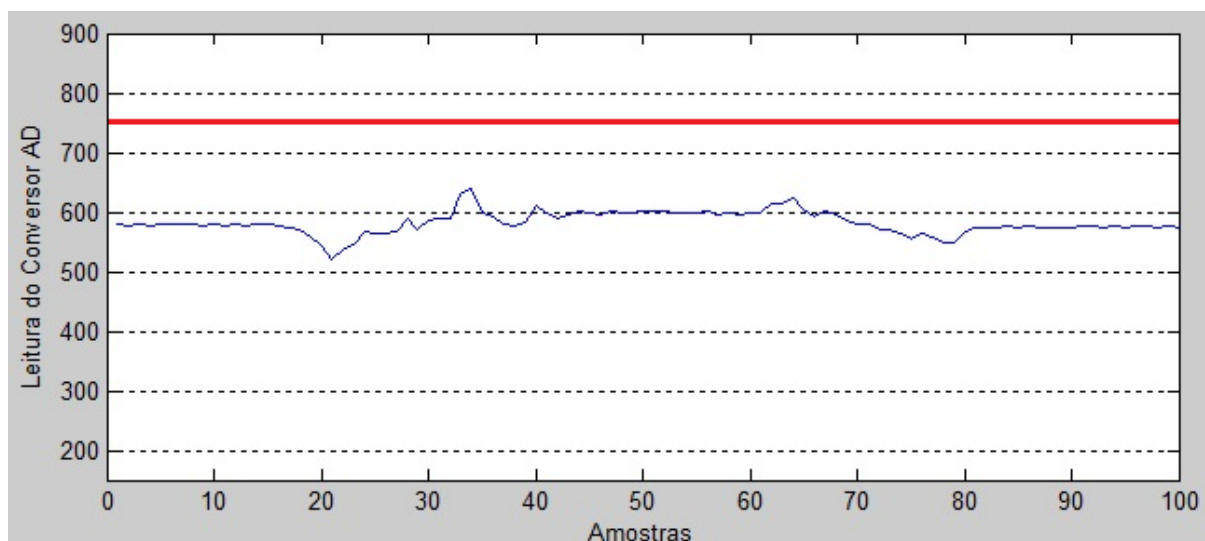
Figura 19 – Sinal de saída do eixo 'y' do acelerômetro obtido durante caminhada lenta do usuário.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

O resultado seguinte, mostrado na Figura 20, foi obtido quando o usuário se sentava e em seguida levantava-se de um banco.

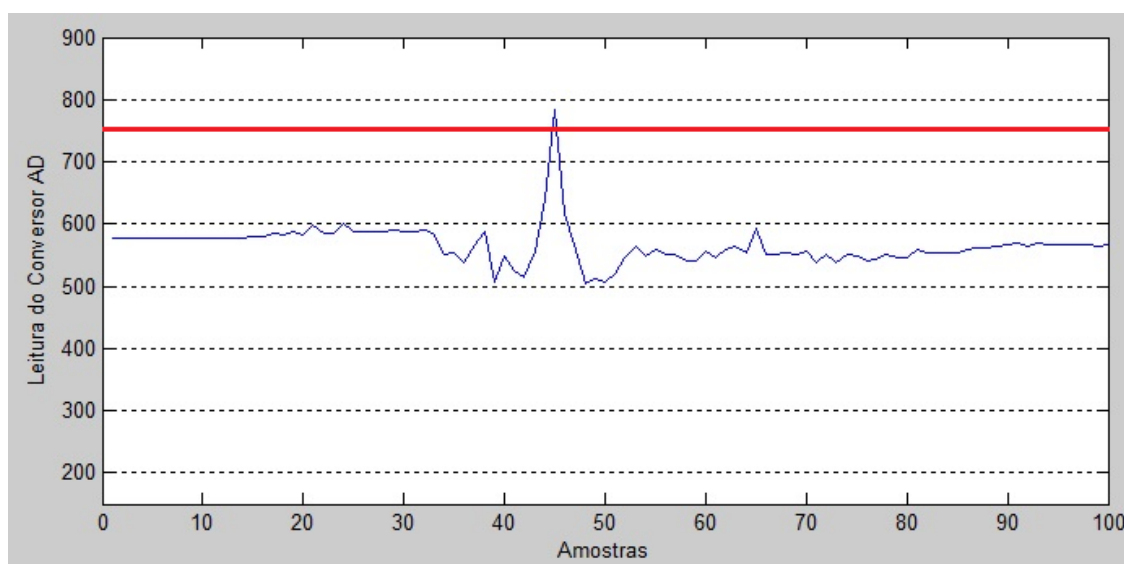
Figura 20 – Sinal de saída do eixo 'y' do acelerômetro obtido enquanto o usuário sentava-se e levantava-se de um banco.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

A partir de então, foram realizadas simulações de situações reais de queda. Inicialmente foi simulada uma queda de joelhos. O resultado está mostrado na Figura 21.

Figura 21 – Sinal de saída do eixo 'y' do acelerômetro obtido enquanto o usuário caía de joelhos.

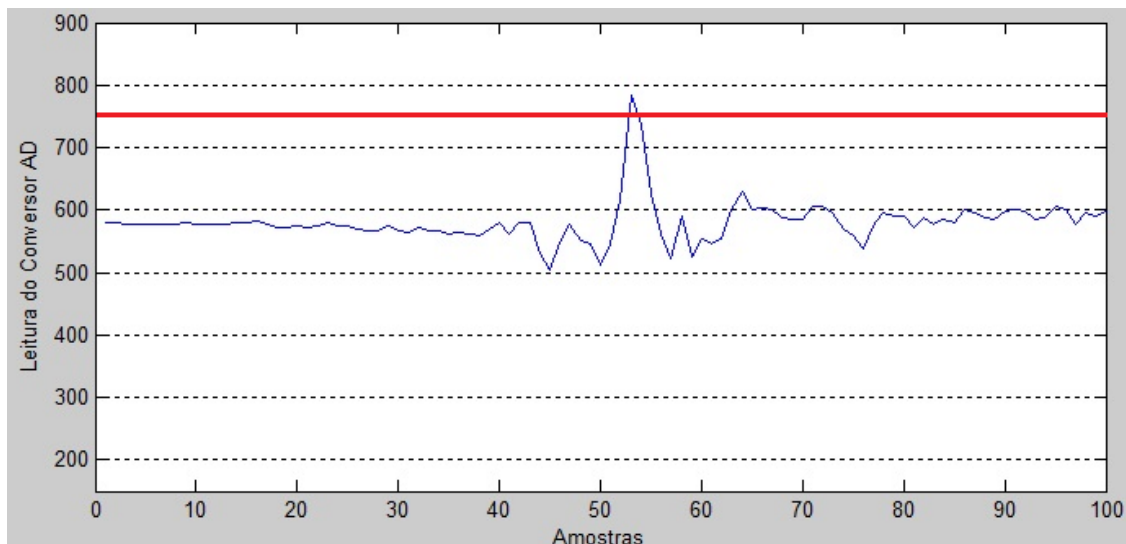


Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

Observa-se que o perfil de resposta deste ensaio é bem diferente dos anteriores. No momento do impacto é possível observar um pico no sinal, que ultrapassa o limite estabelecido de 750,

caracterizando assim uma queda. Para uma queda lateral o resultado obtido está apresentado na Figura 22.

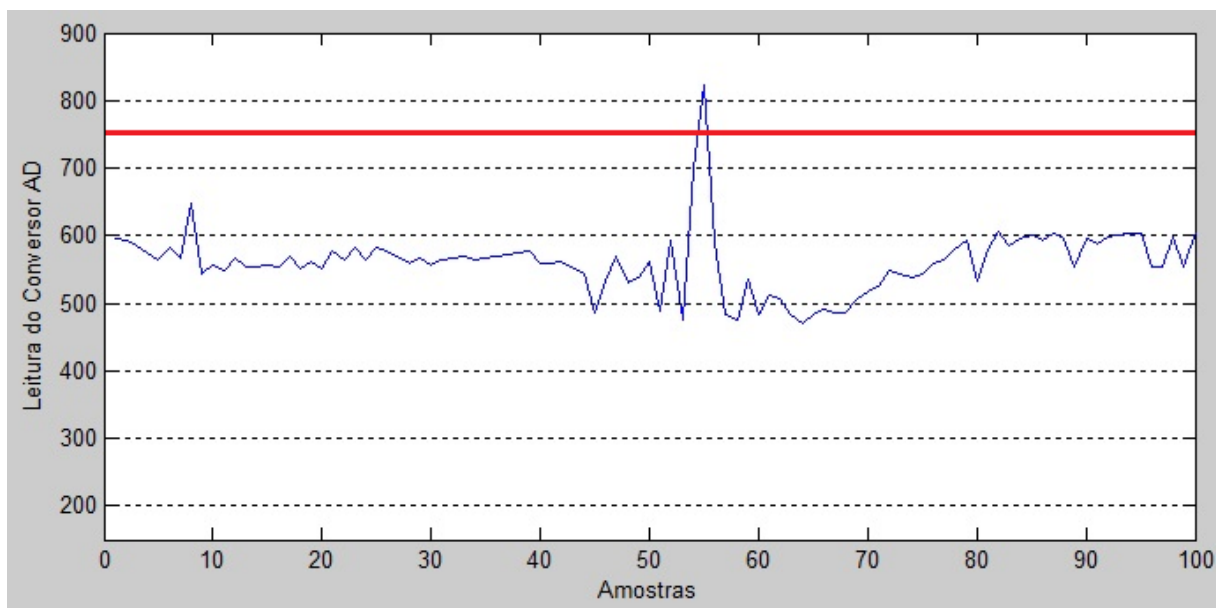
Figura 22 – Sinal de saída do eixo 'y' do acelerômetro obtido enquanto o usuário caía de lado.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

Em uma queda para trás, simulando um desmaio, a resposta do sistema é ainda mais abrupta, conforme visto na Figura 23.

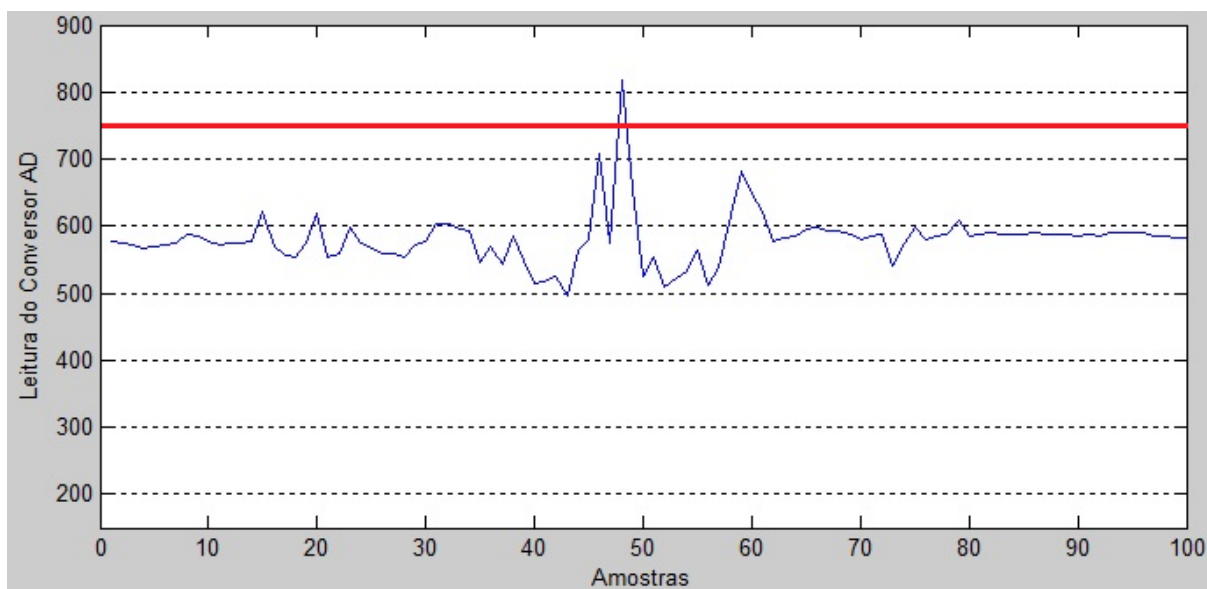
Figura 23 – Sinal de saída do eixo 'y' do acelerômetro obtido enquanto o usuário caía para trás.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

Na Figura 24 é possível observar que uma resposta similar foi obtida quando simulada uma queda frontal.

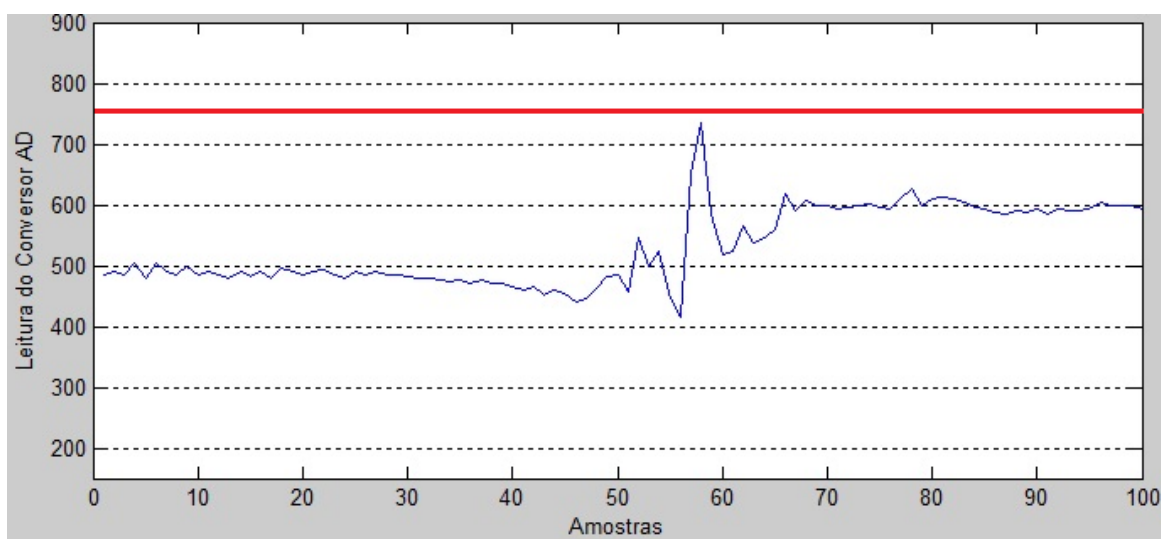
Figura 24 – Sinal de saída do eixo 'y' do acelerômetro obtido simulando uma queda frontal.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

Também foi simulada uma queda enquanto o usuário estava deitado em uma cama à 50cm do chão. O resultado obtido foi o mostrado na Figura 25.

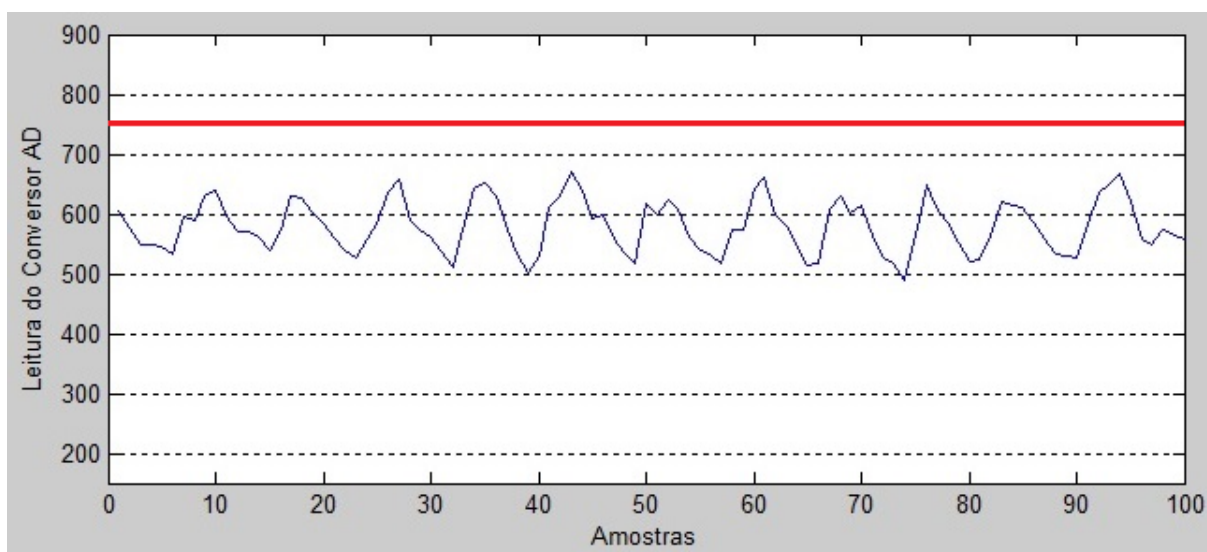
Figura 25 – Sinal de saída do eixo 'y' do acelerômetro obtido simulando uma queda enquanto o usuário estava deitado.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

Nesta situação é possível notar que apesar da variação abrupta, o pico do sinal ficou abaixo do *set point* pré-estabelecido. Para este caso de uso recomenda-se o ajuste do *set point* no nível 700. Para finalizar os testes, também foram selecionadas situações que poderiam gerar alarmes falsos. Situações em que o usuário realizaria movimentos mais rápidos e bruscos. O seguinte resultado mostrado na Figura 26 foi obtido enquanto o usuário praticava uma caminhada acelerada.

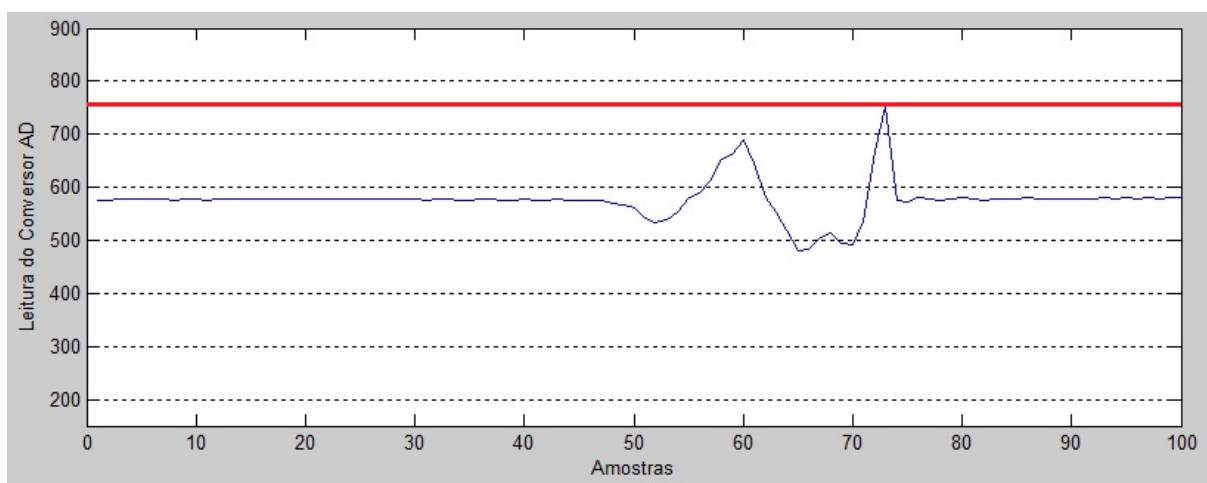
Figura 26 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto usuário caminhava rapidamente.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

A Figura 27 mostra o caso para qual o sinal foi captado enquanto o usuário dava um pulo:

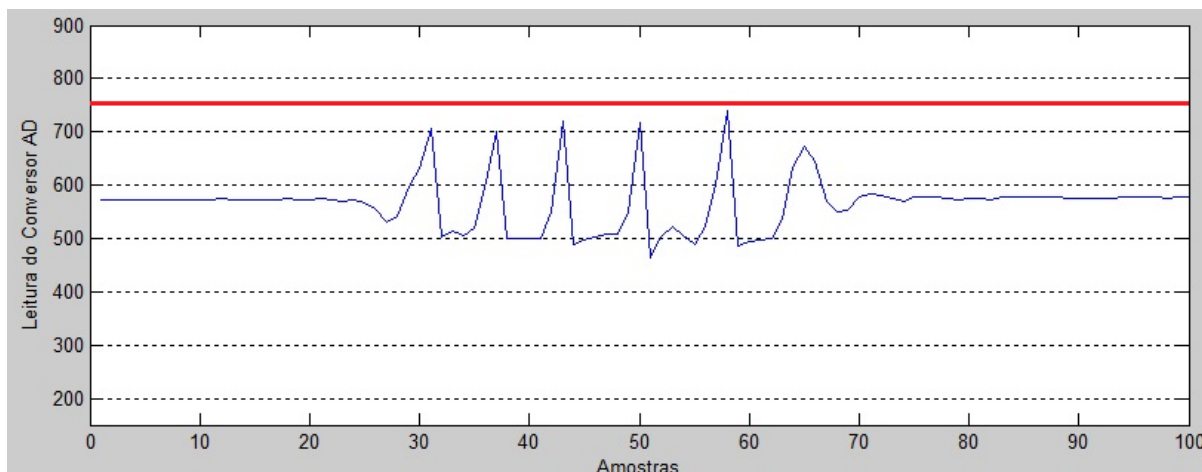
Figura 27 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto usuário dava um pulo.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

Por fim, na Figura 28 os sinais foram captados enquanto o usuário dava uma sequência de pulso curtos:

Figura 28 – Sinal de saída do eixo ‘y’ do acelerômetro obtido enquanto usuário realizava uma sequência de pulso curtos.



Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

É possível notar que nestes casos existe o risco de que os picos do sinal alcancem o valor especificado do *set point* gerando um alarme falso. Aumentar o *set point* gera o risco de que o sistema ignore uma queda real. Portanto, considerando a estrutura utilizada é necessário definir um valor ótimo para o ponto de disparo permitindo que o sistema detecte as principais quedas e ignore situações de alarme falso.

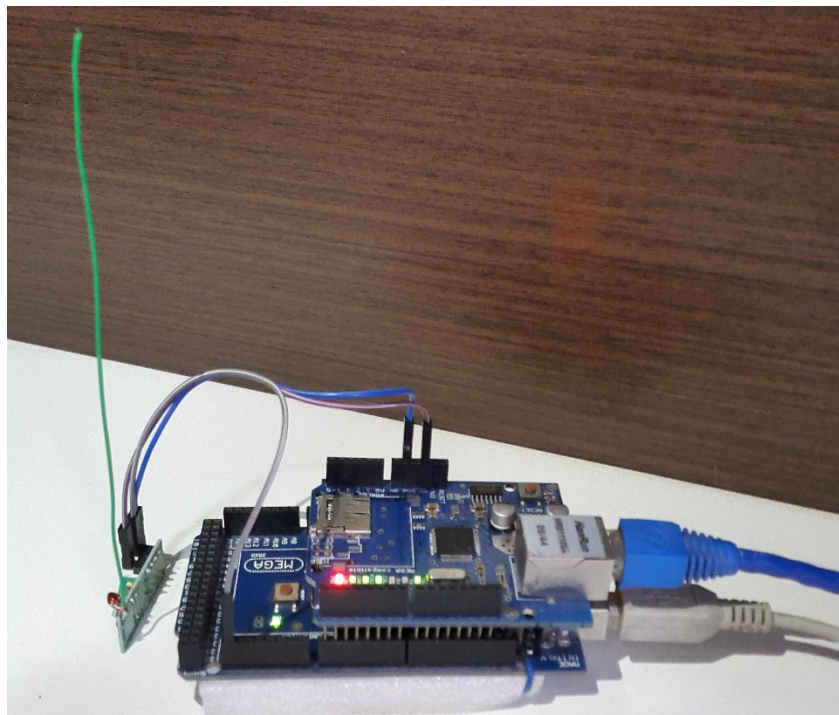
5.2 Resultados dos testes de comunicação entre o dispositivo móvel e a central.

Os testes de comunicação foram realizados com dois módulos de comunicação diferentes, são eles o conjunto transmissor e receptor RF 433MHz e os transceptores NRF24L01 que operam na faixa de 2,4GHz. Foram realizadas análises quanto ao alcance, simplicidade de instalação, confiabilidade da comunicação e consumo de energia.

Inicialmente, foi utilizado o módulo RF 433MHz. Este é um módulo simples de comunicação unidirecional, composto por um receptor e um transmissor. A modulação é do tipo FSK (*Frequency Shift Key*) e a tensão de alimentação no transmissor é de 9V, mesma da bateria. Também foi utilizada uma antena, tanto no módulo receptor quanto no módulo transmissor, com objetivo de aumentar o alcance do grupo. O transmissor foi instalado na cinta móvel e o

receptor na central fixa. Na sequência, tem-se uma foto da central do sistema operando com módulo receptor de 433MHz.

Figura 29 – Central do sistema com o módulo receptor de sinal RF de 433MHz.

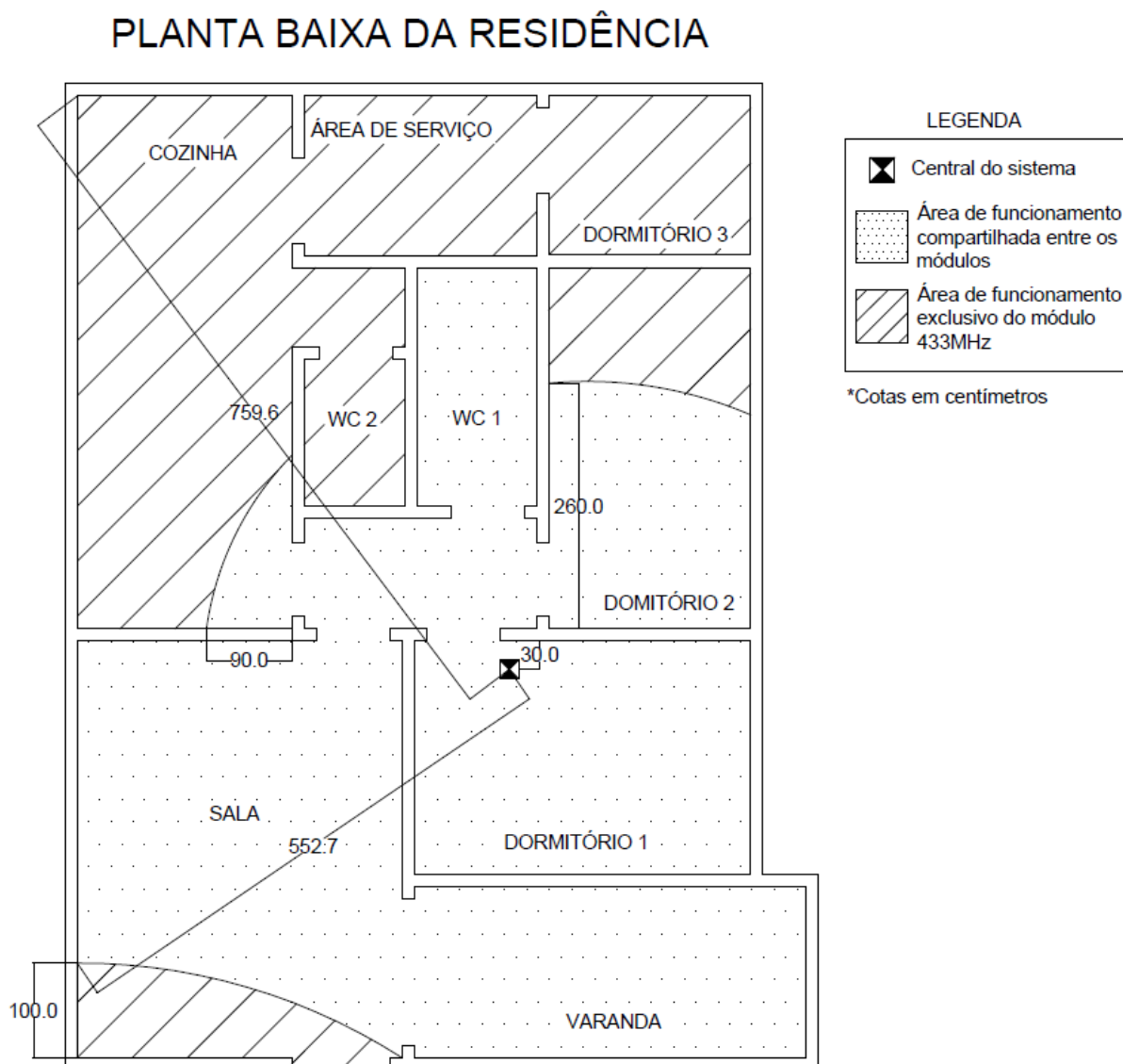


Fonte: Produção do próprio autor. Imagem capturada do *software* MATLAB®.

O segundo módulo testado foi o par de transceptores NRF24L01. A principal vantagem deste dispositivo é a capacidade de configurá-los dinamicamente como receptores ou transmissores, permitindo a comunicação bidirecional. Este módulo opera em frequência mais alta que os tradicionais, conforme especificações apresentadas no Capítulo 3. Não foi utilizada antena externa neste teste.

Os conjuntos foram testados em uma residência, com aproximadamente 70m². Na Figura 30 tem-se uma representação gráfica do apartamento onde foram realizados os primeiros testes. Estão em destaque: a localização da central do sistema, a área de funcionamento efetivo dos módulos, as distâncias de interesse e a presença de paredes (obstáculos), que naturalmente dificultam a navegação do sinal de rádio emitido pelo transmissor do dispositivo vestível.

Figura 30 – Planta baixa da residência com representação da localização da central e áreas de interesse.



Fonte: Produção do próprio autor.

Conforme pode ser observado na imagem, a central fixa do sistema foi instalada no Dormitório 1, e o dispositivo móvel foi deslocado pelo interior do apartamento enquanto eram realizados testes de comunicação com o objetivo de levantar a área de alcance aproximada do módulo.

O primeiro módulo de comunicação foi capaz de realizar contato com a central a partir de todos os pontos da residência, sendo a maior distância testada de aproximadamente 7,6 metros com obstáculos. Já o segundo módulo de comunicação, composto pelos transceptores NRF24L01, apresentou distância máxima de contato de aproximadamente 5,5 metros e, inicialmente, não foi capaz de cobrir efetivamente toda a área da residência. Cabe ressaltar que, os transceptores

utilizados não possuíam antena externa. Segundo os fornecedores, a instalação de uma antena SMA (*SubMiniature version A*) pode aumentar o alcance em área aberta do módulo de 50 metros em campo aberto para 1000 metros, operando na taxa de transferência de 250 Kb/s (ELECTRODRAGON, 2017). O alcance em áreas com obstáculos é muito dependente da configuração do ambiente, como foi possível observar nos testes. Porém, concluiu-se que, caso o alcance em áreas fechadas do módulo com a antena externa tenha o mesmo aumento proporcional do alcance em áreas externas quando comparado com o módulo sem antena, a utilização de um módulo *wireless* NRF24L01 com uma antena SMA, como o da Figura 30, na central fixa atenderia às necessidades deste projeto.

Figura 31 – Módulo NRF24L01 com antena SMA de alto alcance..



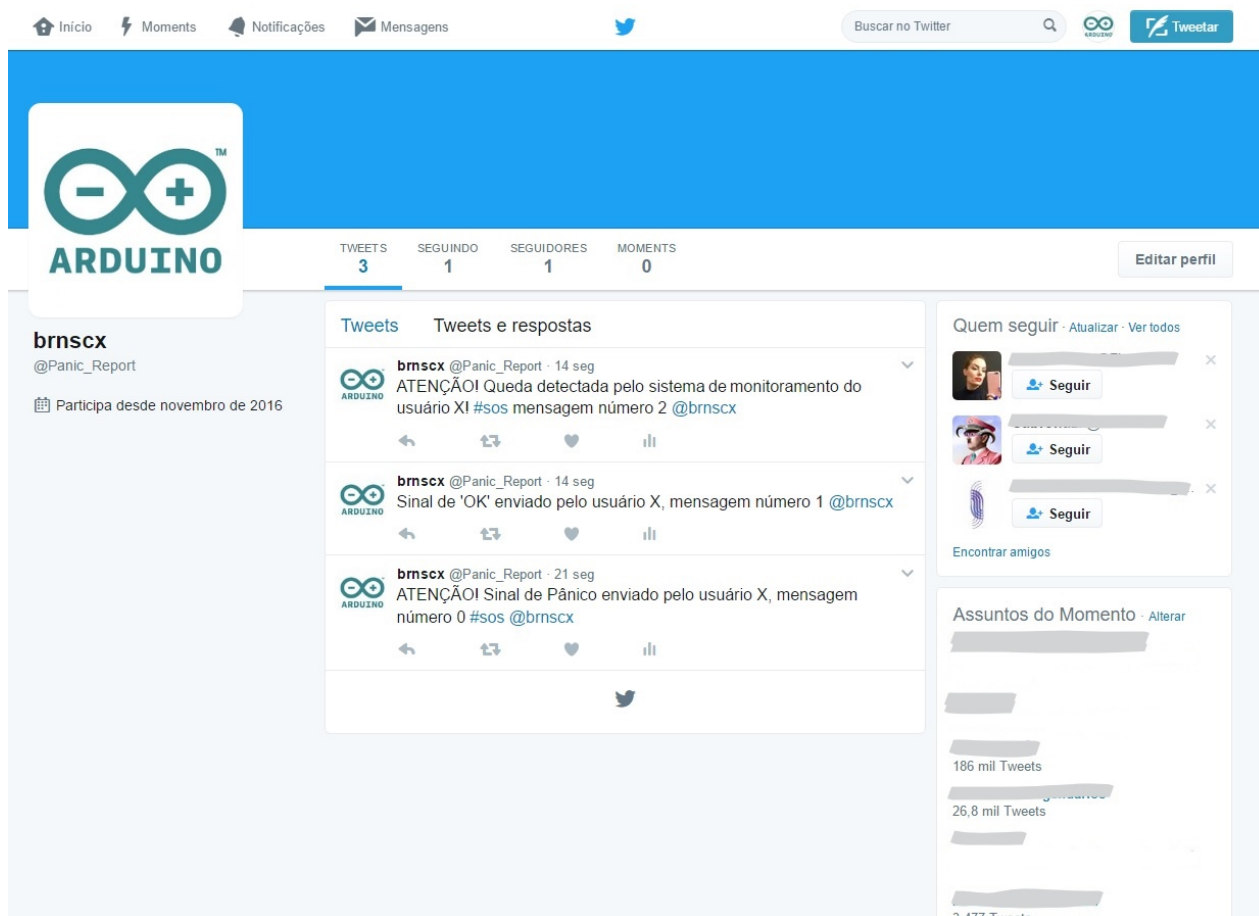
Fonte: FELIPEFLOP(2017).

O grande destaque deste segundo módulo em relação ao primeiro é a capacidade de realizar a comunicação nos dois sentidos, permitindo a confirmação de recebimento dos dados entre os microcontroladores, bem como a futura implantação de algoritmos de criptografia, com objetivo de aumentar a segurança das informações. Cabe informar que o tempo decorrido entre a geração de um sinal de alarme, acionamento por botões ou detecção de queda, seu envio à central fixa e o retorno da confirmação de recebimento é de aproximadamente 16ms.

5.3 Resultados dos testes de comunicação entre o sistema e os cuidadores.

Nesta seção serão apresentados os resultados dos testes de comunicação entre a central fixa do sistema e os cuidadores cadastrados. Na Figura 31, será apresentada a página *web* do *Twitter* do usuário criado especificamente para realizar a transmissão de mensagens do sistema.

Figura 32 – Página da conta criada no *Twitter* com as mensagens enviadas pelo sistema.



Fonte: Produção do próprio autor.

O tempo total observado entre a geração dos alarmes e a postagem na rede social foi em média de 2 segundos. A fim de não ter problemas na postagem por motivos de *spam*, geração de mensagens repetidas, foi gerada uma numeração das mensagens no fim de cada postagem. É possível observar que as mensagens começam com a menção a outro usuário cadastrado como cuidador, o @brnscx. Esta é uma funcionalidade do *Twitter* que permite notificar terceiros usuários imediatamente após publicação da postagem, aumentando a probabilidade do outro usuário ver a mensagem em um tempo curto. Na Figura 32 estão representadas duas imagens da tela de um *smartphone* com sistema operacional Android, por meio do qual o usuário cuidador recebe as notificações.

Figura 33 – *Printscreen* da tela do usuário cuidador com detalhe para a notificação recebida.

Fonte: Produção do próprio autor.

O gerenciamento da conta na rede social não é complexo e pode ser realizado pelo próprio usuário cuidador. Por meio do *Twitter* é possível realizar a comunicação de um evento do sistema para um número ilimitado de usuários em tempo real, bastando apenas que os mesmos sigam a página criada.

5.4 Análise dos resultados

Pode-se concluir que os testes iniciais foram exitosos, porém alguns pontos passíveis de ajustes e otimizações foram detectados. O algoritmo de detecção de quedas é simples, mas funcional. Sabe-se que nenhum dispositivo é completamente imune a alarmes falsos e falhas de detecção, porém o sistema desenvolvido foi capaz de cobrir a maioria dos casos de interesse. A estrutura da plataforma inclusive, foi desenvolvida para permitir a implantação de algoritmos mais complexos no futuro.

O módulo de comunicação RF 433MHz com antena externa apresentou-se como uma alternativa simples e com excelente alcance. Porém, para aumentar o nível de confiabilidade e segurança da comunicação seria necessária a instalação de outro par transmissor/receptor para possibilitar a bidirecionalidade do envio de dados, tornando sua implantação inviável.

A segunda alternativa testada, o transceptor NRF24L01 de 2,4GHz, apesar de possuir gerenciamento mais complexo, proporcionou comunicação bidirecional com robustez, permitindo a validação das informações e a confirmação do recebimento das mensagens pela central do sistema. Nos testes foi observado que para a versão final, o módulo deve estar equipado com uma antena externa, a fim de aumentar o alcance do conjunto.

Foi possível observar também o correto funcionamento dos botões do dispositivo móvel e suas respectivas rotinas de funcionamento, que ativam interrupções de *hardware* do microcontrolador. Pode-se concluir que os testes realizados validaram os requisitos que nortearam o desenvolvimento do dispositivo e indicaram funcionalidades que podem ser posteriormente implementadas por meio de um *upgrade* do *hardware* e do *software* do sistema.

6. CONCLUSÃO

Foi apresentado, neste projeto de graduação, as etapas de desenvolvimento de um dispositivo eletrônico sem fios capaz de detectar quedas e momentos de emergência, desde sua idealização até a construção de um protótipo. A aplicação da tecnologia vestível nos projetos de engenharia vem crescendo ao longo dos últimos anos. Foi possível observar que, apesar de não ser um conceito inédito, esta é uma área da engenharia em constante evolução e com grande perspectiva de crescimento para o futuro.

Para desenvolver o dispositivo que atendesse a todos os requisitos de projeto foi escolhida a topologia *top-down*. Portanto, inicialmente foram analisados os requisitos, definidas as especificações, em seguida foi escolhida a arquitetura adequada e os componentes do dispositivo. Por fim, foi realizada a integração dos elementos e os testes do sistema.

O resultado do projeto foi um protótipo composto por uma central fixa e um dispositivo de tecnologia vestível, este último utilizado como uma cinta pelo usuário. A central fixa comunica-se com o usuário por meio de um módulo de comunicação via rádio. Caso seja detectado um alarme, a central fixa aciona os cuidadores por meio da plataforma do *Twitter*, realizando a postagem de mensagens em uma página específica criada para uso exclusivo do sistema. O dispositivo móvel desenvolvido foi adaptado em uma cinta de uso prático. Ele é leve, de fácil uso e baixo consumo de energia.

A plataforma desenvolvida permite ainda a adição de novas ferramentas de monitoramento por meio da atualização do *software* do sistema, como por exemplo um algoritmo de monitoramento da qualidade do sono e supervisão de postura. Vale destacar que a implementação dos recursos via *software* tem a capacidade de agregar valor ao produto sem adição de custo significativo.

Após a fase de testes foram identificadas algumas áreas do projeto que poderão sofrer atualizações no futuro permitindo a adição de novas funcionalidades ao sistema. Poderão ser implementados os seguintes recursos por meio da adição de módulos adicionais:

1. Monitoramento da temperatura corporal do usuário por meio da instalação de um sensor de temperatura, permitindo a detecção de condições de risco ou doenças com antecedência.
2. Monitoramento da frequência cardíaca através de um sensor de batimentos, viabilizando a análise do comportamento cardíaco do usuário, permitindo a detecção de paradas e arritmias.
3. Monitoramento do sono: o mesmo acelerômetro dedicado à detecção de quedas pode ser utilizado para avaliar os movimentos realizados pelo usuário durante seus períodos de sono, permitindo uma análise quanto à qualidade do seu descanso.
4. Monitoramento da postura: Estrategicamente posicionado, um sensor de posição ou um acelerômetro de três eixos pode acompanhar a postura do usuário ao longo do dia, indicando possíveis desvios e possibilitando a correção antes do acontecimento de qualquer dano à saúde do usuário.
5. Confeção de placa de circuito impresso compacta com todos os componentes dos dispositivos.
6. Instalação dos componentes do dispositivo móvel em uma *case* com proteção contra respingos d'água, grau de proteção IP66, permitindo o uso do equipamento durante o banho.

Ainda poderão ser implantados os seguintes recursos por meio da atualização do *software* dos dispositivos:

7. Configuração de um filtro de detecção de quedas mais robusto.
8. Contagem de passos para saber se o usuário está realizando a quantidade necessária de exercícios diários.
9. Redução temporária da frequência de operação do microcontrolador para reduzir o consumo de energia.
10. Algoritmo de monitoramento do usuário durante o sono, permitindo a detecção de quedas, movimentação corporal e medição da qualidade do sono. Tal recurso pode ser implementado utilizando o próprio acelerômetro do dispositivo móvel ou por meio de um microfone para captação de áudio.
11. Elaboração de relatórios sobre os eventos ocorridos no sistema ao longo do uso pelo usuário.

12. Desenvolvimento de algoritmo de criptografia dos dados que trafegam pelo sistema.
13. Verificação do nível de carga da bateria e geração de alarmes quando houver necessidade de substituição.

Por fim, pode-se concluir que os requisitos do projeto definidos na etapa de detalhamento do problema foram plenamente atendidos e que o protótipo desenvolvido foi capaz comprovar as funcionalidades previstas para o dispositivo.

REFERÊNCIAS BIBLIOGRÁFICAS

ATMEGA. Datasheet: **ATmega328 microcontroller**, San José, 2015. Disponível em <http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf>. Acesso em: 12 dez. 2016.

ARDUINO PLAYGROUNG, Twitter Library. 2012. Disponível em <<http://playground.arduino.cc/Code/TwitterLibrary>>. Acesso em: 29 nov. 2016.

ARDUINO AG. Arduino/Genuino UNO. Londres, 2017. Disponível em <https://www.arduino.cc/en/uploads/Main/A000066_iso_both.jpg>. Acesso em: 20 mar. 2017.

BEECHAM RESEARCH. Laboratório de pesquisa sobre dispositivos vestíveis. Boston, 2013. Disponível em <<http://www.beechamresearch.com/article.aspx?id=20>>. Acesso em: 25 mar. 2017.

COUTO, E.; Goellner, S.V. “Uma estética para corpos mutantes. Corpos mutantes: ensaios sobre novas deficiências corporais”. Porto Alegre: Ed. UFRGS, 2007.

ELECTRODRAGON. Loja *online* de eletrônicos. 2017. Disponível em <<http://www.electrodragon.com/product/1000-meters-nrf24l01-long-distance-module/#prettyPhoto>>. Acesso em: 23 mar. 2017.

EMPRESA BRASIL DE COMUNICAÇÃO. **Brasil não se preparou para cuidar da população idosa, diz gerontóloga**. Brasília, 2015. Disponível em <<http://www.ebc.com.br/noticias/2015/06/brasil-nao-se-preparou-para-cuidar-da-populacao-idosa-diz-geriatra>> . Acesso em: 29 mai. 2016.

FAPERJ – Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro. **Mais qualidade de vida para terceira idade**. Rio de Janeiro, 2015. Disponível em <<http://www.faperj.br/?id=2856.2.9>>. Acesso em: 08 dez. 2016.

FELIPEFLOP. Loja *online* de eletrônicos. Florianópolis, 2017. Disponível em <<http://www.filipeflop.com/pd-3e8c55-modulo-wireless-nrf24l01-transceiver-com-antena-sma.html>> . Acesso em: 23 mar. 2017.

FOLHA DE SÃO PAULO. **CES aponta para futuro da computação vestível e de objetos conectados**. São Paulo, 2014. Disponível em <<http://www1.folha.uol.com.br/tec/2014/01/1396196-ces-aponta-para-futuro-de-computacao-vestivel-e-objetos-conectados.shtml>> . Acesso em: 26 nov. 2016.

GLOBO ESPORTE. **Principal causa de morte entre idosos, quedas podem diminuir com exercício**. Rio de Janeiro, 2013. Disponível em <<http://globoesporte.globo.com/eu-atleta/saude/noticia/2013/06/principal-cao-de-morte-entre-idosos-quedas-podem-diminuir-com-exercicio.html>> . Acesso em: 20 abr. 2017.

IBGE - Instituto Brasileiro de Geografia e Estatística. **Síntese de Indicadores Sociais, 2015**. Disponível em

<<http://www.ibge.gov.br/home/estatistica/populacao/condicaoodevida/indicadoresminimos/sinteseindicisociais2016/>> . Acesso em: 08 dez. 2016

KOSIR, S. Wearbles in Healthcare. **Warable Technologies**, Munich, abr. 2015. Disponível em: <<https://www.wearables-technologies.com/2015/04/wearables-in-healthcare/>>. Acesso em: 27 nov. 2016.

LIMA, T. **O melhor microcontrolador para seu projeto**. 2014. Disponível em <<http://www.embarcados.com.br/microcontrolador/>> . Acesso em: 29 mai. 2016.

MANN'S, S. **Definition of "Warable Computer"**. In: INTERNATIONAL CONFERENCE OF WEARABLE COMPUTING, 1998, Fairfax. Anais. Fairfax: Mai. 1998.

MARWEDEL, P. **Embedded systems design**. 1. ed. Dortmund: Kluwer Academic Publishers, 2003.

MATIAS. **O que é jitter?** Artigo do *blog Ambar Audio Dreams*, 2012. Disponível em <<http://www.ambaraudio.com.br/o-que-e-jitter/>> . Acesso em: 20 abr. 2017.

NORDIC SEMICONDUCTOR. Datasheet: **Transceiver NRF24L01**, 2006. Disponível em <https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf> . Acesso em: 30 jan. 2017.

O GLOBO. **Mercado de computação vestível deve alcançar US\$19 bilhões em 2018**. Rio de Janeiro, 2013. Disponível em <<http://oglobo.globo.com/tecnologia/mercado-de-computacao-vestivel-deve-alcancar-us-19-bilhoes-em-2018-10371494>> . Acesso em: 26 nov. 2016.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. **Relatório mundial de envelhecimento e saúde de 2015**. Disponível em: <<http://sbgg.org.br/wp-content/uploads/2015/10/OMS-ENVELHECIMENTO-2015-port.pdf>>. Acesso em: 30 mai. 2016.

PORTO SEGURO. **Porto Seguro Alerta lança botão de emergência para idosos que moram sozinhos**. 2016. Disponível em <<http://www.portoseguro.com.br/aportoseguro/fiquepordentro/noticias/portoseguroalertalanca-botaodeemergenciaparaidososquemoramsozinhos>> . Acesso em: 29 mai. 2016.

PARK, S.; JAYARAMAN, S. Enhancing the quality of life through wearable technology. **IEEE Engineering in Medicine and Biology Magazine**, Houghton, v. 22, p. 41-48, jul. 2003.

SPARKFUN. Datasheet: **Accelerometer ADX335**, 2016. Disponível em <<https://www.sparkfun.com/datasheets/Components/SMD/adx1335.pdf>> . Acesso em: 11 dez. 2016.

SINGH, H. **NFC Technology comparison**. 2013. Disponível em <<http://thezerolife.com/1328/nfc-technology-near-field-communication-comparison-difference-with-bluetooth-in-detail-2>>. Acesso em: 15 jan. 2017.

TEXAS INSTRUMENTS. Datasheet: **Regulador de Tensão**, Dallas, 2009. Disponível em <<http://www.ti.com/product/TPS770/technicaldocuments>>. Acesso em: 11 dez. 2016.

UOL. **Monitorar idosos a distância, com pulseira e sensor de queda vira negócio**, São Paulo, 2017. Disponível em <<https://economia.uol.com.br/empreendedorismo/noticias/redacao/2017/01/27/monitorar-idosos-a-distancia-com-pulseira-e-sensor-de-queda-vira-negocio.htm>>. Acesso em: 20 abr. 2017.

VARGAS. **Técnicas digitais**, São Paulo, 2016. Disponível em <http://www.vargasp.com/download/livros/Tecnicas_digitais.pdf>. Acesso em: 20 abr. 2017.

WOLF, W. **Computers as components: principles of embedded computing system design**. 2. ed. Burlington: Morgan Kaufmann, 2008.

APENDICE

Código fonte do dispositivo vestível

```

/*
  Bruno Santos Campos - Engenharia Elétrica UFES
  Software do Circuito Transmissor de Dados
  Dispositivo Móvel, alimentado à bateria com módulo de comunicação
  NRF24L01 2.4GHz
*/

//----- Bibliotecas utilizadas -----//
#include <SPI.h>           //Biblioteca para comunicacao com dispositivos
                          //seriais, no caso, o modulo RF
#include "nRF24L01.h"     //Funções de radio comunicacao
#include "RF24.h"
#include "printf.h"       //Funcoes de print serial
#include "TimerOne.h"     //Interrupção de tempo 1

//----- Configuracao do Hardware de Comunicacao -----//
RF24 radio(9, 10);       //Definição dos pinos CE e CSN do modulo NRF24L01

//----- Topologia -----//
const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL }; //Enderecos
                          //para rede com 2 nos de comunicacao.

//----- Variaveis -----//
// Constantes com a numeracao dos pinos:
const int xpin = A0;      //Eixo x do acelerometro
const int ypin = A1;      //Eixo y do acelerometro
const int zpin = A2;      //Eixo z do acelerometro
const int emergpin = 2;   //Botao de Emergencia
const int okpin = 3;      //Botao de Cancelamento de Panico
const int buzzpin = 8;    //Saida do Buzzer

const int cod_panico = 10; //Codigo de sinal de panico
const int cod_ok = 20;     //Codigo de sinal de Ok
const int cod_queda = 30;  //Codigo de sinal de queda
const int cod_teste = 40;  //Codigo de sinal de teste

const int tempo_canc = 2000; //Tempo para cancelamento de envio
const int setpoint_y = 500;  //Setpoint de deteccao de queda para o eixo
Y
const int setpoint_z = 600;  //Setpoint de deteccao de queda para o eixo
z

float x1;                   // Armazenamento do estado dos eixos
do potenciometro
float y1;
float z1;

int sampleDelay = 40;      //Numero de milisegundos entre
leituras

//Flags
int queda = 0;             //Identifica situacao de queda
int panic = 0;            //Identifica situacao de emergencia
int allok = 0;            //Identifica situacao OK
int testecom = 0;         //Identifica necessidade de realizacao
de teste de comunicacao

```

```

int time_panic = 0; //Variaveis auxiliares para tratamento
do bounce
int time_allok = 0;
int time_aux = 0;

//----- ROTINA DA INTERRUPTAO DE PANICO -----//
void panico()
{
  Serial.print(" Interrompido Panico \n");
  panic = 1;
  allok = 0;
  time_panic = millis();
}

//----- ROTINA DA INTERRUPTAO "TUDO OK" -----//
void allisok()
{
  Serial.print(" Interrompido OK \n");
  allok = 1;
  panic = 0;
  time_allok = millis();
}

//----- ROTINA DA INTERRUPTAO TESTE DE COMUNICACÃO -----//
void callback()
{
  Serial.print(" Interrompido Teste Comunicacao \n");
  testecom = 1;
}

//----- FUNÇÃO DE ENVIO DE DADOS -----//
void send(int CODE) {
  radio.stopListening(); //Primeiro, pausa a escuta e então
  pode-se começar a transmitir.

  unsigned long mail = CODE; //Prepara o envio da mensagem
  printf("Enviando agora o codigo: %lu ...", mail);
  bool ok = radio.write( &mail, sizeof(unsigned long) );

  if (ok)
    printf("Ok..."); //Apos confirmacao do envio emite
  alerta e passa para proxima etapa
  else {
    printf("Falha de envio.\n\r");
    tone(buzzpin, 500, 1000); //Toca o Buzzer por 1000ms
    radio.startListening(); //Caso haja alguma falha no envio e
  emitdo um bipe longo e volta-se a ouvir
    return;
  }

  radio.startListening(); //Sinal enviado, o modulo agora
  inicia a escuta para captar a resposta
  unsigned long started_waiting_at = millis(); //Espera ate receber a
  resposta ou tempo de aguardo expirar (250ms)
  bool timeout = false;
  while ( ! radio.available() && ! timeout )
    if (millis() - started_waiting_at > 200 ) timeout = true; //Aguarda o
  recebimento dae uma mensagem de retorno ou 200ms

```

```

if ( timeout ) //Descreve os resultados
{
    printf("Falha, tempo de resposta expirada.\n\r"); //Em caso de falha
de timeout também emite um bipe longo
    tone(buzzpin, 500, 1000); //Toca o Buzzer
por 1000ms
}
else
{
    unsigned long got_time; //Pega a resposta e compara com o
que foi enviado para realizar a autenticacao
    radio.read( &got_time, sizeof(unsigned long) );

    if (got_time == mail) { //Simplesmente confere se a
mensagem recebida e identica a enviada
        // Spew it
        printf("Resposta Recebida: %lu, Autenticado! \n"); //Em caso de falha
de timeout emite dois bipes curtos
        if (CODE != cod_queda) { //Caso nao seja uma mensagem de
queda emite um bipe curto para indicar confirmacao
            tone(buzzpin, 500, 100); //Toca o Buzzer por 100ms
        }
    }
    else {
        printf("Falha na Autenticacao");
        tone(buzzpin, 500, 1000); //Toca o Buzzer por 1000ms
    }
}
// Estabilização
delay(10);
}
//----- SETUP -----//
void setup(void)
{
    pinMode(xpin, INPUT); //Define a entrada do sinal dos eixos do
acelerometro
    pinMode(ypin, INPUT);
    pinMode(zpin, INPUT);

    pinMode(buzzpin, OUTPUT); //Define a saida do buzzer
    digitalWrite(buzzpin, LOW);

    pinMode(emergpin, INPUT_PULLUP); //Botao de Emergencia - Interrupcao
por borda de descida
    attachInterrupt(digitalPinToInterrupt(emergpin), panico, FALLING);
    digitalWrite(emergpin, HIGH); //Resistor de pullup interno
ativado

    pinMode(okpin, INPUT_PULLUP); //Botao Ok - Interrupcao por borda
de descida
    attachInterrupt(digitalPinToInterrupt(okpin), allisok, FALLING);
    digitalWrite(okpin, HIGH); //Resistor de pullup interno
ativado

    Timer1.initialize(10000000); // Inicializa o Timer1 e configura para
um período de 10 segundos. Este tempo pode ser maior
    Timer1.attachInterrupt(callback); // Configura a função callback() como a
função para ser chamada a cada interrupção do Timer1

```

```

    Serial.begin(57600);                // Inicia a comunicacao serial e
imprime preambulo
    printf_begin();
    printf("\n Bruno Campos - EE UFES - PG \n\r");
    printf(" Relatorio de Eventos \n\r");
    printf(" Dispositivo de Transmissao dos Dados \n");

    radio.begin();                      //Inicializa dispositivo de
transmissao RF
    radio.setRetries(15, 15);           // Opcionalmente, aumenta o delay
entre novas tentativas
    radio.setPayloadSize(8);           //Opcionalmente, reduz o tamanho do
payload. Aparentemente aumenta a confiabilidade
    radio.openWritingPipe(pipes[0]);    //Define os canais de comunicacao
para leitura e escrita
    radio.openReadingPipe(1, pipes[1]);
    radio.startListening();            //Inicia Escuta
//radio.printDetails();                //Apresenta as configuracoes do
dispositivo RF
}

//----- PROGRAMA PRINCIPAL -----//
void loop(void)
{
    //----- TESTE DE EVENTO DE PANICO -----
    time_aux = millis();

    if (panic && (time_aux - time_panic) > tempo_canc ) { //Verifica se a
flag esta setada e aguarda o tempo de cancelamento de envio
        send(cod_pânico); //Envia código do pânico
        panic = 0;
    }
    //----- TESTE DE EVENTO DE OK -----
    if (allok && (time_aux - time_allok) > tempo_canc ) { //Verifica se a
flag esta setada e aguarda o tempo de cancelamento de envio
        send(cod_ok); //Envia código do Ok
        allok = 0;
    }
    //----- TESTE DE EVENTO DE COMUNICACAO -----
    if (testecom) {
        send(cod_teste); //Envia código de teste de comunicacao
        testecom = 0;
    }
    //----- LEITURA DA POSICAO -----
    // int x = analogRead(xpin);
    // delay(1);
    int y = analogRead(ypin);
    delay(1); //Insere um pequeno delay entre as leituras
para melhorar a estabilidade.
    int z = analogRead(zpin);

    //----- VERIFICACAO DE QUEDA -----
    if (y > setpoint_y || z > setpoint_z) {
        Serial.println(" Queda detectada !! \n");
        send(cod_queda); //Envia código de queda
    }
    delay(sampleDelay); // delay ate a proxima leitura
}

```

Código fonte da central fixa do sistema

```

/*
  Bruno Santos Campos - Engenharia Elétrica UFES
  Software do Circuito Receptor de Dados
  Dispositivo fixo com módulo de comunicação NRF24L01 2.4GHz
*/

//----- Bibliotecas utilizadas -----//
#include <SPI.h>           //Biblioteca para comunicacao com dispositivos
                          //seriais, no caso, o modulo RF
#include "nRF24L01.h"     //Funcoes de radio comunicacao do transceptor
                          //NRF24L01
#include "RF24.h"
#include "printf.h"       //Funcoes de print serial
#include <Ethernet.h>     //Conectividade com a Internet
#include <Twitter.h>     //Funcoes de postagens pelo servidor do Twitter

//----- Configuracao Ethernet -----//
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // MAC do módulo
Ethernet
//byte ip[] = { 192, 168, 0, 30 };                // Define o IP
estatico para a placa

//----- Configuracao do Twitter -----//
// OAuth Token - Codigo para permitir a postagem de mensagens na conta
// criada para o sistema
// Obtido em: http://cd64.de/arduino-twitter-token
Twitter twitter("800449986613755904-IUB4OGTxSCenVTBzbLaK9fJMqHJjnBL");

//----- Configuracao do Hardware de Comunicacao -----//
RF24 radio(19, 20);      // Definição dos pinos CE e CSN do modulo
                          //NRF24L01

//----- Topologia -----//
const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL }; // Enderecos
                          //para rede com 2 nos de comunicacao.

//----- Variaveis -----//
const int buzzpin = 14; // Saida do Buzzer

const int cod_panico = 10; //Codigo de sinal de panico recebido
const int cod_ok = 20;    //Codigo de sinal de Ok recebido
const int cod_queda = 30; //Codigo de sinal de queda recebido

int i = 0;                //Contador de mensagens
char buf[100];           //Armazenamento provisório das mensagens que
                          //serao enviadas

//----- SETUP -----//
void setup(void)
{
  Serial.begin(57600);    //Imprime preambulo
  printf_begin();
  printf("\n Bruno Campos - EE UFES - PG \n\r");
  printf(" Relatorio de Eventos \n\r");
  printf(" Dispositivo de Recepcão dos Dados \n \n");

  pinMode(buzzpin, OUTPUT); //Define o pino 14 como
                              //saída do buzzer

```

```

digitalWrite(buzzpin, LOW);

//----- INICIALIZAÇÃO COMUNICACAO -----//
radio.begin(); //Inicializa dispositivo de
transmissao RF
radio.setRetries(15, 15); //Opcional: aumenta o delay entre
novas tentativas
radio.setPayloadSize(8); //Opcional: reduz o tamanho do
payload. Aparentemente aumenta a confiabilidade
radio.openWritingPipe(pipes[1]); //Configura os enderecos para envio
e recebimento de informacoes
radio.openReadingPipe(1, pipes[0]);
radio.startListening(); //Inicia Escuta
// radio.printDetails(); //Apresenta as configuracoes do
dispositivo RF NRF24L01

//----- INICIALIZAÇÃO ETHERNET -----//
if (!Ethernet.begin(mac)) { // Tenta realizar a
configuracao via DHCP
printf(" Falha na obtencao de IP. Dispositivo Offline \n \n"); //
Em caso de falha emite alerta.
}
else {
delay(100); // Tempo de
estabilizacao
Serial.print(F("Conectado! Endereco IP: ")); // Imprime o endereco de
IP do sistema
for (int i = 0; i < 4; i++)
{
Serial.print(Ethernet.localIP()[i], DEC);
Serial.print(".");
}
}
}
//----- FUNCAO TWEET -----//
void tweet(char msg[]) {
Serial.println("Conectando ...");
if (twitter.post(msg)) { //Inicia postagem de
mensagem no Twitter
int status = twitter.wait(&Serial);
if (status == 200) {
Serial.println("OK."); //Em caso de sucesso
emite mensagem de confirmacao
} else {
Serial.print("Erro! Codigo: "); //Em caso de falha exhibe
sinal de erro e o respectivo codigo
Serial.println(status);
}
} else {
Serial.println("Falha de conexao"); //Em caso de falha de
conexao emite alerta
}
}
//----- PROGRAMA PRINCIPAL -----//
void loop(void)
{
if ( radio.available() ) //Recebe os dados, verifica e envia
resposta.
{

```

```

    unsigned long got_pack;      // captura pacote até baixar toda a
informação
    bool done = false;
    while (!done)
    {
        done = radio.read( &got_pack, sizeof(unsigned long) ); // Captura o
ultimo pacote.
        printf("Sinal recebido: %lu...", got_pack);           // Apresente o
sinal recebido

        tone(buzzpin, 500, 100);                             //Toca o Buzzer por
1000ms

        if (got_pack == cod_panico) {                        //Verifica o tipo de
sinal recebido e realiza os posts
            printf("\n Sinal de Panico recebido \n");
            sprintf(buf, "ATENÇÃO! Sinal de Pânico enviado pelo usuário X,
mensagem número %d #sos @brnscx", i);
            tweet(buf);
            i++;
        }
        else if (got_pack == cod_ok) {
            printf("\n Sinal de OK recebido");
            sprintf(buf, "Sinal de 'OK' enviado pelo usuário X, mensagem número
%d @brnscx", i);
            tweet(buf);
            i++;
        }
        else if (got_pack == cod_queda) {
            printf("\n Sinal de Queda recebido");
            sprintf(buf, "ATENÇÃO! Queda detectada pelo sistema de
monitoramento do usuário X! #sos mensagem número %d @brnscx", i);
            tweet(buf);
            i++;
        }
        delay(20);     //Aguarda um pequeno intervalo de tempo para
que o outro modulo chaveie de transmissor para receptor
    }
    radio.stopListening();                                  // Para de receber
para poder enviar a resposta.
    radio.write( &got_pack, sizeof(unsigned long) ); // Envia de volta o
sinal recebido sem alteracoes.
    printf(" Enviando resposta.\n \n \r");
    radio.startListening();                                //Inicia a escuta para
poder capturar os proximos pacotes
    }
}

```