Título

Subtítulo

Relatório Final de
pesquisa de Iniciação Científica

Aluno: Leonardo José Held
Orientador: Raimes Moraes

Departamento de Engenharia Elétrica e Eletrônica
Curso de Graduação em Engenharia Eletrônica
Universidade Federal de Santa Catarina
Brasil

Conteúdo

Capítulo 1

Resumo

Palavras-chave: Instrumentação médica. Sistema respiratório. Sons respiratórios. Estetoscópio Eletrônico. Processamento Digital de Sinais.

Capítulo 2

Introdução

O diagnóstico prematuro correto de enfermidades respiratórias pode representar uma redução significativa de mortes. Nesse sentido, são utilizados métodos de asculta pulmonar, realizada rotineiramente por médicos treinados. Um ponto forte é o baixo custo, não possuindo necessidade de equipamentos proibitivamente caros e não invasivo, pois não exige movimentação excessiva de pacientes acamados. No entanto, o examinador pode ser impedido de executar o exame com acurácia devido à inexperiência e capacidade auditiva. Assim, surgiram estetoscópios eletrônicos de baixo custo, que podem captar e preparar sinais que podem ser processados via técnicas de análise, retirando a subjetividade e democratizando o acesso ao exame. Esse trabalho propõe o aprimoramento e expansão de um sistema de aquisição de sons respiratórios produzidos no LCS - UFSC. O módulo(MATTOS, 2018), - que antes contava com receptor e transmissor *Bluetooth 5.0*, e um periférico de aquisição de sinais analógicos - foi aprimorado com uma nova interface para sensor de posicionamento espacial.

2.1 Objetivo

O aprimoramento do sistema - referenciado na introdução deste capítulo -, tem como objetivo auxiliar o leque de informações disponíveis para processamento e próximas pesquisas nessa área. Em específico, informações sobre o posicionamento toráxico do paciente aliadas as informações extraídas dos sons pulmonares podem ajudar a mitigar fatores de erro no processamento desses mesmos dados.

(HUANG et al., 2020) sugere que a ascultação pulmonar pode ser usada para detecção antecipada de doenças como COVID-19, tornando estetoscópios eletrônicos ainda mais pertinentes pelas vantagens já citadas.

Capítulo 3

MATERIAL E MÉTODOS

3.1 Revisão bibliográfica

A tecnologia *Bluetooth 5.0* é uma tecnologia definida por um grupo de trabalho denominado *Bluetooth Special Interest Group*. Nesse, se define um padrão de comunicação sem fio entre dois - ou mais - dispositivos numa banda de frequência normalmente utilizada para fins industriais, científicos e médicos. Originalmente, o Bluetooth foi criado para estabelecer um tipo de comunicação simples, serial, que já existia na forma de conexão via cabo entre dispositivos.

De acordo com (AH, 2010), os principais pontos que fazem com que *Bluetooth* sejam uma boa escolha meios médicos são: interoperabilidade, baixo consumo, customização de formato de transmissão (pacote), compatibilidade eletromagnética (equipamentos médicos coexistem, não podendo inteferir negativamente uns com os outros), segurança na transmissão para proteger confidencialidade da informação e hub de sensores (diversos equipamentos médicos precisam ser atualizados em tempo real via internet para fins de telemedicina).

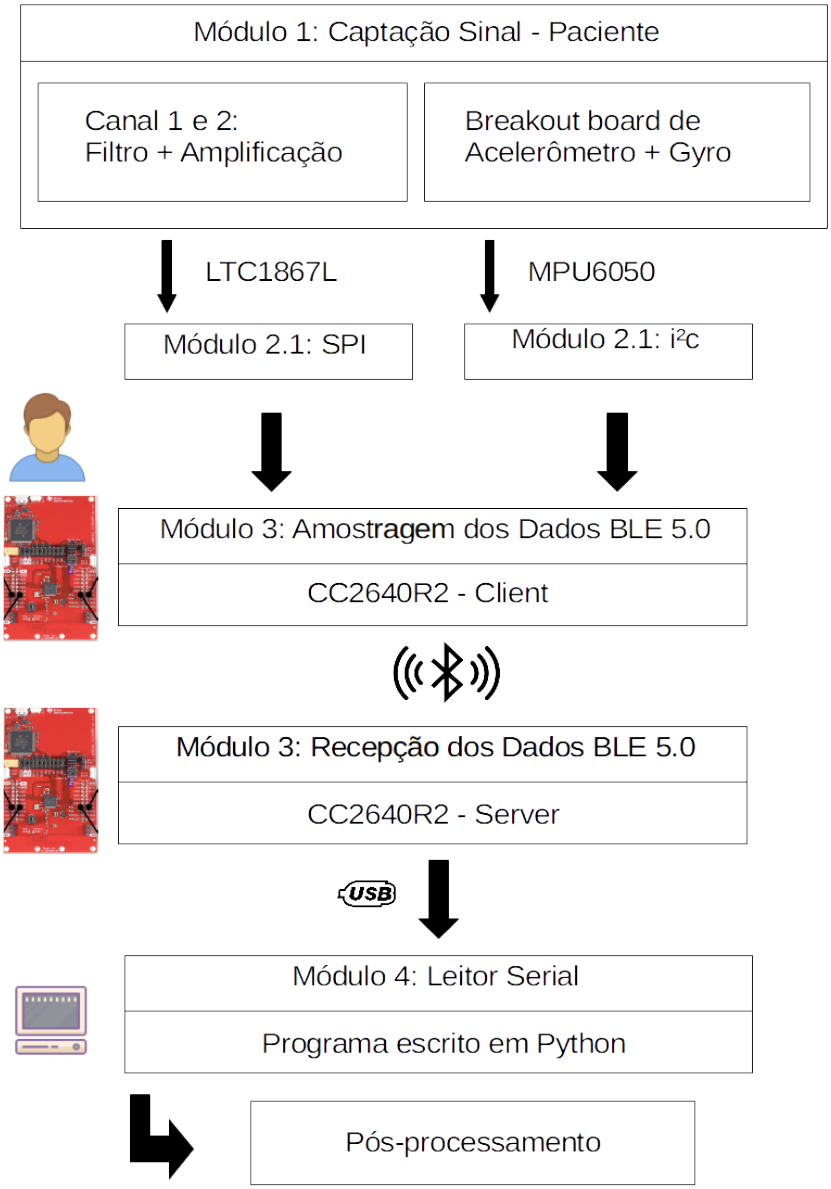
A necessidade de se utilizar uma tecnologia de ponta nesse projeto surge da grande quantidade de dados envolvi-

dos na transferência de sons respiratórios. Na especificação do *BLE* ou *Bluetooth Low Energy*, um padrão de Bluetooth utilizado pelo baixo consumo, o *throughput* é limitada em 2mpbs, porém, dado um *overhead* utilizado por headers e dependente do hardware onde o protocolo está sendo executado, a velocidade efetiva de transmissão é geralmente menor do que a especificada.

Também é um protocolo aberto para implementação, realizado pelas principais empresas de tecnologia mundiais, fomentando os pontos de interoperabilidade de equipamentos e segurança.

O sistema construído conta com uma interface para acelerômetro, giroscópio (**Módulo 2.2**) e um *ADC - Analog to Digital Converter*, Conversor Analógico-Digital - (**Módulo 2.1**). O acelerômetro + giroscópio é utilizado para obter informação espacial do tórax, e o Conversor Analógico-Digital é utilizado para obter sons respiratórios via uma interface de transdutores (**Módulo 1**).

Também conta com um programa escrito em Python que obtém os dados recebidos pelo USB - via serial - do módulo de recepção dos dados, denominado **Módulo 4**, que são enviados pelo módulo de amostragem, ou **Módulo 3**.



Fonte: O autor.

Dado o problema de aquisição de dados em períodos de tempo rígido, existe a necessidade de uso de um Sistema Operacional em Tempo Real rodando em ambos módulos de amostragem e recepção dos dados.

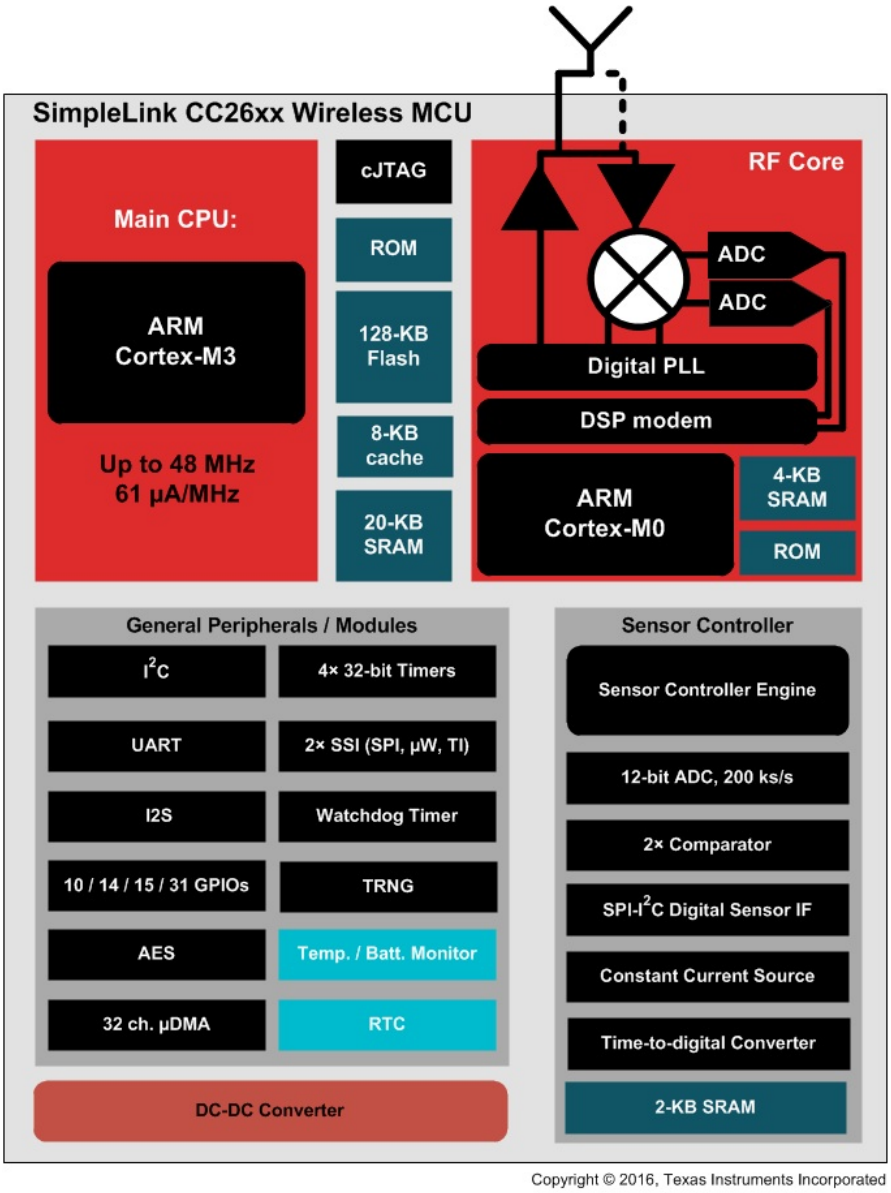
De acordo com (BARRY, s.d.) um Sistema Operacional em Tempo Real como um sistema que precisa saber o

tempo de término quão mais exato de uma tarefa. Ou seja, um sistema em tempo real se carateriza pelo caráter determínistico. É ideal para aplicações onde ações externas precisam ser induzidas em intervalos de tempo regulares e extremamente corretos, como aquisição de som, no nosso caso, ou o disparo de um air-bag.

Ambos módulos de Amostragem e Recepção dos dados são placas de desenvolvimento da *Texas Instruments*®, codenome `LAUCHPAD-XL` contendo o chip *CC2640R2* e que suportam a tecnologia *BLE 5.0*, *Bluetooth Low Energy 5.0*.

Ademais, a empresa disponibiliza juntamente com as placas um Sistema Operacional em Tempo Real denominado `TIRTOS` juntamente com um kit de desenvolvimento (SDK). Por estes, fazemos uso das facilidades do hardware de Bluetooth que existem nessa plataforma.

3.2 Características de módulos microcontrolados com capacidades *BLE 5.0*



Fonte: Texas Instruments®

Observando o esquemático de blocos do *SoC*, vemos que o mesmo conta com um processador ARM Cortex-M3 e ARM Cortex-M0. O primeiro serve de processador de propósito geral, e é extendido para os pinos de entrada e saída (*I/O*) de cada placa. O segundo é um processador de propósito específico que contém código que controla o circuito de radiofrequência (*RF*) responsável pela implementação do protocolo *Bluetooth* da placa.

Existe um protocolo de comunicação não-acessível pelo usuário da placa entre os dois processadores que serve para aquisição e envio de mensagens pelo Bluetooth. Fazemos uso, então, do *Cortex-M3* para acessar e obter dados dos sensores ou periféricos. Estes, por sua vez, são enviados via para a segunda *CC2640R2* usando a API da *TI*.

Algumas limitações de hardware ainda são realidade para essa tecnologia, fato que, por extensão, limita a escolha de placas de desenvolvimento no mercado. A tecnologia *BLE 5.0*, por mais que o protocolo da presente versão tenha sido homologado em 2016, ainda não é totalmente adotada e nem amplamente utilizada por vários dispositivos. Visto que o ecossistema do projeto já estava baseado na `LAUCHPAD-XL`, o projeto ali se deu prosseguimento.

3.3 Estudo de ambientes de programação, Sistema Operacional e exemplos de fabricante

A *TI* disponibiliza, em termos, o TIRTOS, um Sistema Operacional em Tempo Real (a partir daqui denominado como RTOS, ou *Real Time Operating System*, *Sistema Operacional em Tempo Real*, em tradução livre). Tal RTOS é uma peça de software capaz de gerenciar de forma inteligente, levando em consideração questões de tempo, rotinas e procedimentos escritos em linguagem de programação(INSTRUMENTS,

s.d.). A empresa, no entanto, não disponibiliza o código aberto de tal RTOS, mas sim uma *API - Application Programming Interface*, que permite seu uso. Essa *API* contém *drivers*, ou camadas de abstração entre o código e o uso do hardware. Por exemplo, um driver de timer para controlar ciclos de tempo, ou um driver de periférico como *SPI*.

O princípio de funcionamento básico de um RTOS é fazer com que várias rotinas (também chamadas de funções) possam ser executadas em *quasi*-paralelismo. Ou seja, eficientemente alocando tempo de execução de código em um único processador para concluir várias tarefas diferentes, criando uma ilusão de paralelismo entre as mesmas. Além, a parte de "Tempo Real"se refere aos padrões rígidos que o sistema impõe para execução de tarefas.

Tais sistemas são de suma importância para diversas aplicações na indústria e na ciência, sendo exaustivamente utilizado em diversos setores, como robótica, equipamentos biomédicos e quaisquer sistemas sensíveis a requerimentos de tempo.

A empresa também disponibiliza exemplos de aplicação que podem ser extendidos e modificados livremente, sob licença aberta. Um desses é o *Simple Port Profile*, de nome homônimo ao padrão apresentado na seção 3.1, que é um simples firmware que conecta duas plataformas *CC2640* e transfere texto numa configuração half-duplex (bidirecional, não simultâneo) entre as duas.

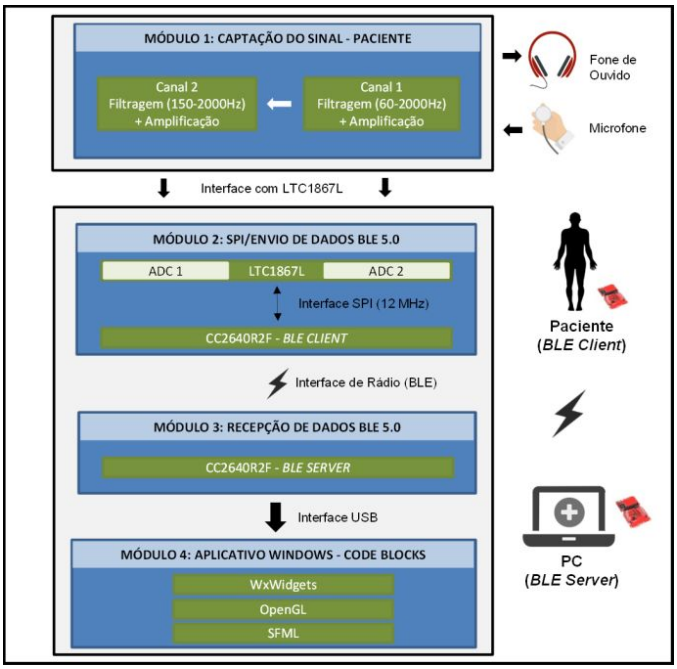
Esse exemplo foi compilado, estudado e analisado, visando compreendimento base do Sistema Operacional. O projeto também usa esse exemplo de aplicação como projeto-base.

3.4 Desenvolvimento de firmware para testes de sinais simulados de aúdio

A partir disso, foram iniciados estudos sobre como o sistema de aquisição e transmissão criado por (MATTOS, 2018). Durante esse passo, notamos as limitações promovidas pela construção e configuração padrão do hardware da plataforma *CC2640R2*.

O principal esforço se deu a trabalhar dentro de limitações extremas de memória volátil (memória não persistente). Com apenas 8`kb` de memória do tipo volátil, sendo que o Sistema Operacional em uso pode ocupar cerca de 25% desse valor, fazendo com que inserção de rotinas novas necessitem de cuidado para evitar possíveis *overflows*, ou estouro de memória por sua insuficiência.

O sistema conta com um *ADC* de 16 *bits* de dois canais modelo *LTC1867LCGN* da *Linear Devices®*. Este se conecta a uma front-end desenvolvida por [trabalho que desenvolveu a frontend]



Parâmetro	**Valor**
Modo de comunicação CC2640R2	Mestre
Modo de comunicação LTC1867L	Escravo
Taxa do clock	12MHz
Tamanho da Amostra	16 bits
CPOL (Clock Polarity)	0
CPHA (Clock Phase)	0
Modo	Callback

Fonte: (MATTOS, 2018)

O projeto pode ser utilizado para se obter sinais de com pouca interferência e distúrbio para pacientes. Os sinais podem, *a posteriori* ser processados e analisados por técnicas específicas de processamento de sinal. A interface também pode ser vista como um estetoscópio eletrônico de baixo custo e performance comparável aos dispositivos existentes no mercado, que são de código fechado e valor proibitivo para uma grande parcela de possíveis aplicações.

A comunicação $ADC \rightarrow CC2640$ é realizada pelo padrão *SPI*, tal como citado na seção 2.1 com as seguintes configurações:

Foram corrigidos avisos gerados por conversões digitais implícitas que existiam nos valores amostrados do ADC quando levadas ao buffer de transmissão utilizado pelo projeto.

Para disparar a aquisição desse periférico é utilizado um relógio em modo periódico que faz a requisição dos sinais a cada $100\mu s$.

O sistema foi então testado com sinais alternados tanto gerados pelo próprio microcontrolador quanto externos e obteve resposta satisfatória.

3.5 Desenvolvimento de Aplicativo para Notebook

O projeto original também conta com um aplicativo para Notebooks e Computadores Windows, que captura os pacotes transmitidos via Bluetooth pelo *Client* do sistema por emulação de porta serial via USB. O primeiro problema foi a falta de documentação sobre o projeto, que não provinha de instruções de compilação. As bibliotecas utilizadas no projeto também eram de versões antigas, com diversos problemas de interoperabilidade.

Ultrapassando o problema de compilação do projeto, percebemos que o mesmo utilizada a *API* do próprio sistema Windows NT. O código utilizado para acesso ao periférico da porta serial emulada sobre USB era datado, não se comportando bem em novas versões do sistema Windows NT. Decidimos então criar um novo programa que capture as informações de quaisquer porta serial, com a necessidade de fácil manutenção e pouca modificação entre sistemas operacionais host. Dessa forma, foi escrito um programa na linguagem de programação Python que obtém os dados da porta serial e os salva para pós-processamento, que mostrou bom desempenho. Se necessário, métodos de pós-processamento podem ser programados de maneira não obtusa no programa, gerando os dados de maneira rápida.

3.5.1 Código para atendimento de interrupção externa

Considerando que acessar o equipamento notebook que pode estar a alguns metros de distância do conjunto paciente, examinador e equipamento Bluetooth, surgiu a necessidade de um botão que dispare o programa de computador supracitado.

Uma modificação simples foi um algoritmo de *poooling* - ou espera - de inicialização remota da gravação, em con-

junto com uma modificação que não custou a já escassa memória de dados das placas. Aumentando o conforto do uso em pacientes com dificuldade de locomoção.

Quando um botão é pressionado no Client, o mesmo envia uma curta mensagem de 8 bytes para o Server. O Server é conectado via cabo USB ao Notebook. Quando o Notebook detecta essa mensagem, ele envia uma mensagem ao Client pelo Server para que se comece a inicialização da aquisição e ao mesmo tempo começa a gravação dos dados recebidos pelo Server.

3.6 Testes e aprimoramento do Sistema Desenvolvido

Uma das propostas concebidas pelo orientador foi a integração de um acelerômetro e giroscópio, dispositivo que mede aceleração e posição absoluta, de forma a obter dados espaciais do tórax do paciente. A partir dessa proposta foi elaborado uma interface parar aquisição de dados de pequenos sensores que contém tanto acelerômetro quando giroscópio.

O periférico de giroscópio e acelerômetro escolhido foi o *MPU6050*. Seu baixo custo, alta precisão e plena disponibilidade oferecem um bom dispositivo. O *MPU6050* é conta com 6 *ADCs*, cada um de 16 *bits*. Cada um desses *ADCs* é responsável pela digitalização de um eixo do acelerômetro ou do giroscópio embarcados na *breakout board*. Também oferece um filtro passa-baixas digitalmente programável, descartando a necessidade de pós-processamento e baixo consumo (em torno de *3.6mA* de acordo com a fabricante).

Nossa expansão incluiu o módulo de acelerômetro e giroscópio *MPU6050*, o que possibilita a captação de sinais

Parâmetro	**Valor**
Modo de comunicação CC2640R2	Mestre
Modo de comunicação MPU6050	Escravo
Taxa do clock	400kHz
Tamanho da Amostra	16 bits

de posição em três dimensões com n-graus de liberdade, onde n é o número de sensores acelerômetro e giroscópio disponíveis no sistema. Claramente existe um limite no número de sensores dado as limitações de memória e hardware disponíveis no sistema. Em específico, o MPU6050 possui um sistema simples de endereçamento, que permite endereçar até dois chips usando apenas um periférico i^2c. Como a plataforma atual possui dois desses periférico, é possível o endereçamento de até 4 combos de sensores de posição espacial. Qualquer pino pode ser reprogramado para funcionar como esse protocolo.

Abaixo são listados os parâmetros de configuração i^2c.

A metodologia de programação desse periférico se deu no desenvolvimento de um *driver* do mesmo, que abstraí rotinas de baixo nível do periférico via uma *API*. Como no programa escrito em Python para recebimento dos dados pela serial, optamos por uma forte filosofia de portabilidade, escrevendo a biblioteca de uma maneira portável e agnóstica à plataforma, podendo ser utilizada em outros microcontroladores e até outros projetos.

A taxa de aquisição desse sensor foi programada em função da taxa de aquisição do *ADC*. A cada disparo do relógio, existe uma simples lógica de contador que divide esse clock afim de realizar a aquisição do acelerômetro + giroscópio. Dessa forma, a taxa de aquisição do acelerômetro + giroscópio nunca pode ser maior do que a do *ADC*. De qualquer forma a taxa de 12MHz do *ADC* é maior que a taxa de 1KHz do acelerômetro e 8KHz do giroscópio.

Capítulo 4

RESULTADOS E DISCUSSÃO

As condições para realização dos testes não foi propícia devido aos eventos externos ocorridos no ano de 2020, impossibilitando tanto contato quanto acesso aos equipamentos e profissionais necessários para continuar os trabalhos normalmente, dadas as questões sanitárias. De qualquer forma, houveram testes regulares em todas as partes do projeto, o que garante o funcionamento em ambiente de desenvolvimento.

Ademais, a natureza de dispositivos embarcados é de baixo consumo e baixo processamento. Logo, a maior dificuldade em todas as etapas do projeto foi fazer uma alta manutenção de RAM (memória volátil). Ainda sim, a plataforma escolhida pode permitir comutações de código entre processadores e famílias de dispositivos, permitindo fácil portabilidade para microcontroladores com maior capacidade que venham a ser lançados suportando BLE 5.0.

Referências

AH, Omre. Bluetooth low energy: wireless connectivity for medical monitoring. **J Diabetes Sci Technol**, v. 4,2, n. 457-63, 2010.

BARRY, Richard. **Mastering the FreeRTOS™ Real Time Kernel**. [S.l.].

HUANG, Ying hui et al. The respiratory sound features of COVID-19 patients fill gaps between clinical data and screening methods. **medRxiv**, Cold Spring Harbor Laboratory Press, 2020. DOI: 10.1101/2020.04.07.20051060. eprint: https://www.medrxiv.org/content/early/2020/04/10/2020.04.07.20051060.full.pdf. Disponível em: ¡https://www.medrxiv.org/content/early/2020/04/10/2020.04.07.20051060¿.

INSTRUMENTS, Texas. **TI-RTOS Kernel (SYS/BIOS) User's Guide**. [S.l.].

MATTOS, Willian Daniel de. **Dissertação (mestrado)**. [S.l.: s.n.], 2018.

STALLMAN, Richard M.; DEVELOPERCOMMUNITY, GCC. **Using The Gnu Compiler Collection: A Gnu Manual For Gcc Version 4.3.3**. Scotts Valley, CA: CreateSpace, 2009. ISBN 144141276X.

Interface do Driver MPU6050

```
bool        mpu6050_test         ();
bool        mpu6050_writeByte    (char reg, char value);
int8_t      mpu6050_readByte     (char reg);
bool        mpu6050_reset        ();
bool        mpu6050_configAccel  (mpu6050_range_t range);
bool        mpu6050_configGyro   (mpu6050_dps_t range);
uint16_t    getAccel             (mpu6050_axis axis);
uint16_t    getGyro              (mpu6050_axis axis);
```