

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ENGENHARIA ELETRÔNICA

VITOR LUCAS RUBIO HUNGRIA

DESENVOLVIMENTO DE UM DISPOSITIVO DETECTOR DE ARRITMIAS CARDÍACAS

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO

2020

VITOR LUCAS RUBIO HUNGRIA

**DESENVOLVIMENTO DE UM DISPOSITIVO DETECTOR DE ARRITMIAS
CARDÍACAS**

Development of a cardiac arrhythmia detecting device

Trabalho de conclusão de curso de graduação
apresentado como requisito para obtenção do título de
Bacharel em Engenharia eletrônica da Universidade
Tecnológica Federal do Paraná (UTFPR).
Orientador: Dr. Eduardo Giometti Bertogna.

**CAMPO MOURÃO
2020**



TERMO DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO INTITULADO

DESENVOLVIMENTO DE UM DISPOSITIVO DETECTOR DE ARRITMIAS CARDÍACAS

DO(A) DISCENTE

VITOR LUCAS RUBIO HUNGRIA

Trabalho de Conclusão de Curso (TCC) apresentado em 07 de dezembro de 2020 às 10:00 como requisito parcial para obtenção do título de Bacharel em Engenharia Eletrônica. O(A) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com alterações.

Prof. Dr. Eduardo Giometti Bertogna
Orientador

Prof. Dr. Marcio Rodrigues da Cunha
UTFPR

Prof. Me. Lucas Ricken Garcia
UTFPR

AGRADECIMENTOS

Agradeço ao meu professor orientador Dr. Eduardo Giometti Bertogna quem me ajudou muito na utilização da plataforma de programação online MBED, me forneceu materiais nos quais pude me basear e sanou eventuais dúvidas e dificuldades que surgiram no decorrer do desenvolvimento do projeto.

Gostaria de agradecer o professor Dr. André Luiz Regis Monteiro pelas aulas de biomédica que possibilitaram um entendimento maior do sinal cardíaco e do funcionamento do coração.

Também quero agradecer aos demais professores do curso que contribuíram para minha formação em todas as áreas da Engenharia Eletrônica, tanto na parte prática como na teórica.

Por fim, agradeço a meus familiares e colegas de curso que me ajudaram a superar as dificuldades que apareceram por todos esses anos em que fiquei longe de casa estudando no Paraná.

RESUMO

As arritmias cardíacas são indicativos importantes de cardiopatias em pacientes com sintomas de problemas cardiológicos, e se adequadamente detectadas possibilita a redução da morbidade e da mortalidade ao se conduzir os procedimentos terapêuticos adequados antes da ocorrência de um evento de maior risco ao paciente. Este trabalho propõe a criação de um dispositivo capaz de monitorar o sinal eletrocardiográfico (ECG) de um paciente em busca de eventuais arritmias. Para isto, o dispositivo registra, em memória não volátil por um período de 24 h, os trechos de ECG que apresentarem anormalidades, possibilitando que um cardiologista possa analisar, posteriormente, os eventos registrados em busca de uma eventual cardiopatia. O dispositivo projetado consta de uma placa de desenvolvimento de baixo custo, conhecida como Blue Pill, com microcontrolador ARM Cortex-M3 dotado de conversor analógico para digital (ADC) de 12 bits para a aquisição do sinal de ECG, e um adaptador para cartão de memória SD a ser usado como meio de armazenamento dos dados de ECG anômalos. Para o teste do protótipo utilizou-se um simulador de ECG conectado à entrada do ADC, evitando assim a necessidade de testes em humanos. A detecção das arritmias é realizada monitorando-se as principais componentes morfológicas do sinal de ECG, de forma individualizada para cada batimento cardíaco, e da comparação destas com valores limites considerados normais, sendo os eventos anômalos gravados no cartão SD. Os resultados obtidos se mostraram promissores no monitoramento de arritmias cardíacas, sendo que os componentes do sinal elétrico foram detectados de forma coerente e o sinal gravado possui alterações visíveis. Esses resultados possibilitam que o projeto tenha condições de auxiliar profissionais médicos da área de cardiologia no estudo e detecção de cardiopatias.

Palavras chaves: Cardiopatias. Eletrocardiograma. Sistema microcontrolado.

ABSTRACT

Cardiac arrhythmias are important indications of heart disease in patients with symptoms of cardiac problems, and if properly detected, it allows the reduction of morbidity and mortality by conducting appropriate therapeutic procedures before the occurrence of an event of greater risk to the patient. This work proposes the creation of a low-cost device capable of monitoring the electrocardiographic signal (ECG) of a patient in search of possible arrhythmias. For this, the device registers, in non-volatile memory for a period of 24 h, the ECG traces that present abnormalities, allowing a cardiologist to later analyze the events recorded in search of a possible heart disease. The designed device consists of a low-cost development board, known as the Blue Pill, with an ARM Cortex-M3 microcontroller equipped with a 12-bit analog to digital converter (ADC) for the acquisition of the ECG signal, and an adapter for SD memory card to be used as a storage medium for anomalous ECG data. To test the prototype, an ECG simulator connected to the ADC input was used, thus avoiding the need for tests on humans. Arrhythmias are detected by monitoring the main morphological components of the ECG signal, individually for each heartbeat, and comparing them with limit values considered normal, with anomalous events recorded on the SD card. The results obtained showed promise in the monitoring of cardiac arrhythmias, the components of the signal were detected as expected and the written signal has some visible alterations. Those results allow the project to help medical professionals in the field of cardiology to study and detect cardiac diseases.

Keywords: Cardiopathy. Electrocardiogram. Microcontrolled system.

LISTA DE FIGURAS

Figura 1 – Representação do Coração Humano.....	13.
Figura 2 – Exemplo de um sinal ideal de ECG.....	14.
Figura 3 – Comparação da musculatura cardíaca com o sinal do ECG.....	15.
Figura 4 – Triângulo de Einthoven	16.
Figura 5 – Fluxograma de funcionamento.....	18.
Figura 6 – STM32F103	19.
Figura 7 – Modulo de leitura e escrita de cartão SD	20.
Figura 8 – ST-Link.....	20.
Figura 9 – Simulador de ECG	22.
Figura 10 – Módulo de comunicação serial.....	23.
Figura 11 – Projeto montado em protoborad.....	24.
Figura 12 – Diagrama de Blocos da montagem do projeto	25.
Figura 13 – Diagrama de blocos do algoritmo de Pan e Tompkins.....	26.
Figura 14 – Vetor teste não filtrado	27.
Figura 15 – Vetor teste após filtro passa baixas.....	27.
Figura 16 – Vetor teste após filtro passa banda.....	27.
Figura 17 – Vetor teste após filtro derivativo	27.
Figura 18 – Vetor teste após função quadrática.....	27.
Figura 19 – Vetor teste após janela de média movel	27.
Figura 20 – Detecções dos componente do ECG no vetor de testes.....	32.
Figura 21 – Sinal 1 Obtido com ausência do pico R.....	33.
Figura 22 – Sinal 2 Obtido com ausência do pico R.....	34.
Figura 23 – Sinal Obtido com 1200 amostras	34.

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

AD	Átrio Direito
ADC	<i>Analog to Digital Converter</i> (Conversor analógico para digital)
AE	Átrio Esquerdo
DAC	<i>Digital to Analog Converter</i> (Conversor digital para analógico)
DCI	Doença cardíaca isquêmica
DDA	Dispositivo Detector de Arritmia
ECG	Eletrocardiograma
ESV	Extrassístoles ventriculares
GPIO	<i>General Purpose Input/Output</i> (Pino de entrada e saída de uso geral)
IA	<i>Instrumentation amplifier</i> (Amplificador instrumental)
IAM	Infarto agudo do miocárdio
IC	Iniciação científica
PCR	Parada cardiorrespiratória
RFI	<i>Radio-Frequency Interference</i> (Interferência de Radiofrequência)
SPI	<i>Serial Peripheral Interface</i> (Interface Periférica Serial)
TV	Taquicardia ventricular
VD	Ventrículo Direito
VE	Ventrículo Esquerdo

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	11
1.2	Justificativa	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Sinal elétrico	13
3	METODOLOGIA	17
3.1	Microcontrolador	19
3.2	Plataforma Online MBED	20
3.3	Simulador de ECG	21
3.4	Métodos de detecção	25
3.4.2	Algoritmo de Pam e Tompkins	26
3.4.3	Detecção de componentes	28
3.4.4	Comparação de valores	30
4	RESULTADOS	32
5	CONCLUSÃO	36
	REFERÊNCIAS	38
	APÊNDICE A – ROTINA MAIN	40
	APÊNDICE B – ROTINA INIT HW	41
	APÊNDICE C – ROTINA DE INTERRUPÇÃO POR TIMEOUT	42
	APÊNDICE D – ROTINA DE PROCESSAMENTO DE BUFFER	43
	APÊNDICE E – ROTINAS PAN E TOMPKINS E DE DETECÇÃO DE COMPONENTES	44
	APÊNDICE F – ROTINAS DE DETECÇÃO DE COMPONENTES R E S	45
	APÊNDICE G – ROTINA DE DETECÇÃO DE COMPONENTE T	46
	APÊNDICE H – ROTINA DE DETECÇÃO DE COMPONENTE P	47
	APÊNDICE I – ROTINA DE DETECÇÃO DE COMPONENTE Q	48
	APÊNDICE J – ROTINAS DE CÁLCULO DAS MÉDIAS R E T	49
	APÊNDICE K – ROTINA DE DETECÇÃO DE ARRITMIAS	50
	APÊNDICE L – ROTINA DE ARMAZENAMENTO DE DADOS EM STRING	51
	APÊNDICE M – ROTINA DE CONTAGEM DE 24 HORAS	52
	APÊNDICE N – ROTINA DE GRAVAÇÃO EM MEMÓRIA EXTERNA	53

APÊNDICE O – CODIGO DO PROJETO.....	54
--	-----------

1 INTRODUÇÃO

De acordo com a Organização Pan-Americana de Saúde, as doenças cardiovasculares são a maior causa de mortes no mundo, e estima-se que 17,7 milhões de pessoas morreram vítimas de doenças cardiovasculares em 2015, sendo este número foi equivalente a 31% das mortes no mundo naquele ano. Mais de três quartos destas mortes ocorrem em países de baixa e média renda per capita, indicando a falta de preparo para detectar e lidar com essas doenças, já que nestes países as pessoas muitas vezes não têm o benefício de programas de avaliação cardiológica para indivíduos com fatores de risco ou com histórico familiar apresentando estes fatores de risco. Essas pessoas também possuem menos acesso a serviços de saúde adequados, sendo eles diagnosticados tardiamente, levando muitas vezes a morte prematura do indivíduo (OPAS, 2017).

As pessoas mais carentes em países de baixa e média renda são as mais afetadas. Em nível familiar, estão emergindo evidências suficientes para provar que as doenças cardiovasculares e outras doenças não transmissíveis contribuem para a pobreza devido às despesas catastróficas com saúde e gastos elevados além do planejado (OPAS, 2017).

A OPAS ressalta ainda que os pacientes, muitas vezes, não apresentam sintomas da doença cardiovascular existente. Um ataque cardíaco ou acidente vascular cerebral pode ser o primeiro aviso da doença subjacente. Os sintomas do ataque cardíaco incluem:

- Dor ou desconforto no centro do peito;
- Dor ou desconforto nos braços, ombro esquerdo, cotovelos, mandíbula ou costas.

Além disso, a pessoa pode ter dificuldade em respirar ou falta de ar; sensação de enjoo ou vômito; sensação de desmaio ou tontura; suor frio; e palidez. Mulheres são mais propensas a apresentar falta de ar, náuseas, vômitos e dores nas costas ou mandíbula.

A Sociedade Brasileira de Arritmias Cardíacas (SOBRAC) apontou em 2016 que mais de 20 milhões de brasileiros possuem algum tipo de arritmia cardíaca, este

número representa aproximadamente 10% da população, e cerca de 320 mil pessoas morrem anualmente no país vítimas de complicações cardíacas (SOBRAC, 2016).

As Arritmias são alterações no comportamento do sinal elétrico cardíaco, e elas podem ser de dois tipos:

- *Arritmias Benignas*: manifestam-se sem apresentar grandes complicações para o indivíduo;

- *Arritmias Malignas*: frequentemente associadas a complicações de saúde que podem levar a óbito.

O diagnóstico de arritmias se faz através do estudo do sinal elétrico do paciente que é obtido através de um ECG, “O registro de um ECG de repouso dura em média poucos minutos, tendo valor apenas nos casos de arritmias cardíacas sustentadas, e que não apresentem variabilidade espontânea durante o dia.” (BERTONHA, 1994, p2).

Mas na realidade, as arritmias podem se manifestar em variados períodos, e durante atividades físicas ou mentais, como por exemplo as extrassístoles ventriculares que podem não ser detectadas em exames de rotina por não apresentarem sintomas na hora do exame (BERTONHA, 1994).

Para solucionar este problema, em 1961 N. J. Holter e B. Glasscock desenvolveram um aparelho que era capaz de gravar o sinal cardíaco e que pudesse armazenar dados dos batimentos do paciente por 10 horas. Os dados eram gravados em fitas magnéticas, a fim de serem examinados posteriormente por cardiologistas, que poderiam determinar se o paciente em questão estaria ou não sofrendo algum tipo de alteração em seu batimento cardíaco (BERTONHA, 1994).

Inicialmente, por conta das limitações tecnológicas da época o aparelho não era muito portátil, possuindo o tamanho de 19,5 x 9,8 X 4,6 cm e pesando em torno de 1 Kg (BC MEDICAL JOURNAL, 2014). Entretanto, N. J. Holter considerou o aparelho como um grande avanço já que não era mais necessário que o paciente ficasse parado durante toda a gravação, e ele poderia se locomover levando o dispositivo nas costas, como o próprio N. J. Holter testou utilizando o equipamento enquanto pedalava em uma bicicleta estacionária. Apesar de inicialmente possuir grandes dimensões e peso elevado, o gravador Holter foi uma evolução enorme na cardiologia, e com o avanço da tecnologia, ficou cada vez mais portátil, e eficaz (BC MEDICAL JOURNAL, 2014).

Os gravadores Holter atuais se tornaram mais compactos e microcontrolados com capacidade de armazenamento maior em memória, podendo ser utilizados por longos períodos causando muito menos desconforto para o paciente, possibilitando seu uso durante o dia sem alterações drásticas na rotina diária do paciente.

Com a evolução das tecnologias de aquisição de ECG e dos microprocessadores, já é possível a implementação de algoritmos de detecção de arritmias em tempo real diretamente no dispositivo Holter. Portanto, na atualidade, já se dispõem de diversas formas de se identificar as arritmias cardíacas através de monitores de arritmia.

O diferencial de um monitor de arritmia para um Holter é prover uma análise em tempo real dos parâmetros do sinal eletrocardiográfico (ECG), e gerar um alerta na eventualidade de ocorrer um distúrbio no traçado do ECG, e então, identificar os trechos do ECG onde esses eventos ocorreram, salvando na memória não volátil o período de 24 horas após a detecção inicial destas alterações. Desta forma gravando um sinal mais relevante, já que o Holter tem que armazenar todos os dados do paciente em 24 horas independente do comportamento do sinal enquanto o monitor somente iniciará a gravação a partir de detecção de alguma anomalia.

1.1 Objetivos

O intuito deste trabalho foi o de desenvolver um dispositivo com tecnologia similar ao Holter, porém com algumas particularidades. O dispositivo é um detector de arritmias que terá como principal função monitorar em tempo real os valores dos componentes do sinal elétrico do coração do paciente comparando-os com valores considerados normais e em caso de discrepância, gravar as informações em um cartão SD.

Desta forma, os objetivos específicos deste trabalho foram:

1. Análise em tempo real de sinais de ECG normais e apresentando arritmias cardíacas;
2. Implementação de um algoritmo que possa identificar as diferentes componentes do ECG, e comparar os valores destas componentes com valores adotados como limite de um batimento saudável;

3. Implementação do hardware do detector de arritmias na forma de um sistema microcontrolado dotado de um adaptador para cartão de memória SD para a gravação dos sinais de ECG anômalos detectados.

1.2 Justificativa

Por conta de grandes quantidades de pessoas falecerem anualmente vítimas de complicações cardíacas, e que 10% dos brasileiros possuem algum tipo de arritmia (SOBRAC, 2016) um estudo nesta área é de grande importância para possibilitar uma redução na taxa de mortalidade anual decorrente de doenças cardíacas. Como este trabalho utiliza componentes de baixo custo para se detectar e registrar as arritmias, ele pode ser reproduzido e aprimorado em trabalhos futuros por outros pesquisadores da área.

2 FUNDAMENTAÇÃO TEÓRICA

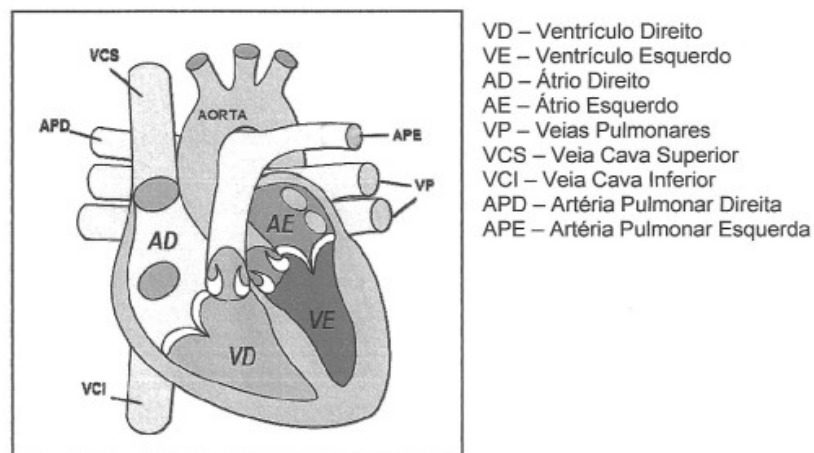
A fim de obter uma detecção de cardiopatias de forma coerente, foi necessário o estudo da anatomia cardíaca e da sua relação com o sinal elétrico do coração, chamado sinal eletrocardiográfico - ECG. Também houve a necessidade da pesquisa de maneiras de se implementar a detecção e análise de cada componente do sinal de ECG de maneira que torne possível seu estudo e classificação como saudável ou possivelmente patológico sendo estas apresentadas neste capítulo.

2.1 Sinal elétrico

Para entender melhor o tema deste trabalho, deve se ter algum conhecimento da fisiologia do coração e como ele funciona. Uma representação do coração pode ser observada na Figura 1.

O coração é composto de quatro cavidades: átrio direito (AD), átrio esquerdo (AE), ventrículo direito (VE) e ventrículo esquerdo (VE) como também das artérias aorta e pulmonar e das veias pulmonar e cava (GUYTON e HALL, 2011).

Figura 1 – Representação do coração humano.



Fonte: (SANTOS, 2015)

O coração é percorrido por sinais elétricos a cada batimento cardíaco, cada componente desses sinais representa a contração ou relaxamento de algum músculo

cardíaco. Através da análise das componentes desses sinais pode-se determinar se o coração apresenta alguma arritmia (GUYTON e HALL, 2011).

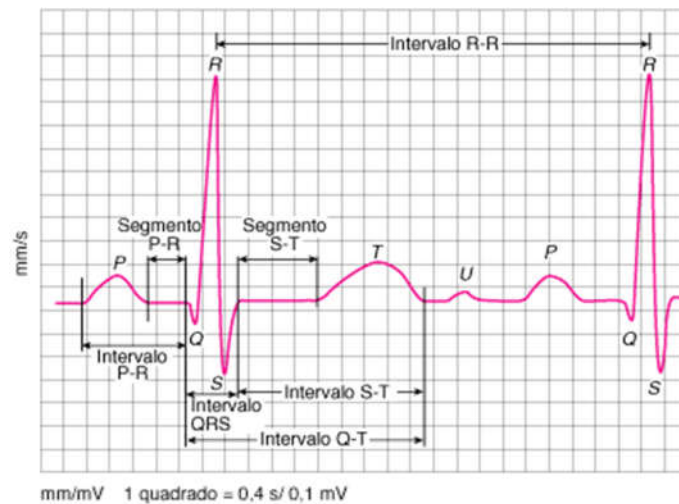
Um ECG é um exame que através de eletrodos fixados no corpo do paciente é capaz de reproduzir esses sinais elétricos e com isso possibilitar a análise da condição cardíaca do paciente naquele determinado momento (GUYTON e HALL, 2011).

Pode-se entender o que cada componente do sinal do ECG representa ao se analisar o processo do batimento cardíaco.

A Figura 2 mostra a imagem de um sinal de ECG com todos seus componentes.

O sinal de um eletrocardiograma, como o apresentado na Figura 2, possui onda P, complexo QRS, onda T e onda U, e é caracterizado pelos intervalos entre esses pontos, como o intervalo PR, segmento ST e intervalo QT.

Figura 2 – Exemplo de um sinal ideal de ECG e seus componentes.



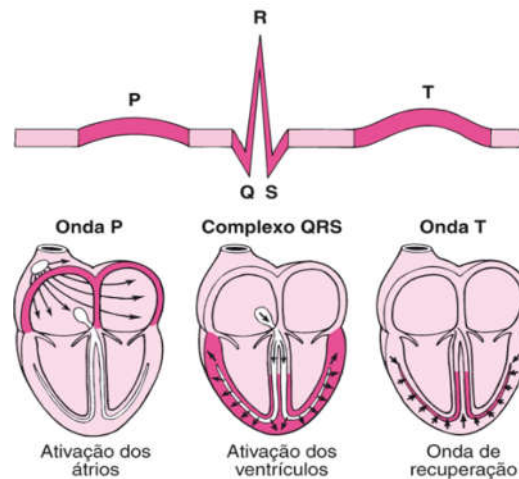
Fonte: (MANUAL MSD, 2017)

Em cada batimento, o sinal elétrico que percorre o coração se inicia com uma despolarização de uma contração atrial, num ECG essa etapa é representada pela onda P, que é quando o sangue entra no coração pelas veias Cava e Pulmonares e então os átrios se contraem para que caiba o máximo possível de sangue no coração. Depois, ocorre a despolarização e contração dos ventrículos, que é representado pela componente QRS do sinal, é onde o coração manda o sangue para fora pelas Artérias

Aorta e Pulmonar, então o coração relaxa e o processo se reinicia, o que é caracterizado pela onda T (GUYTON e HALL, 2011). Este processo pode ser observado na Figura 3.

Sendo assim, o dispositivo proposto neste trabalho tem como função, comparar a cada momento o sinal do paciente com um sinal ideal e registrar os casos em que exista alguma discrepância entre eles.

Figura 3 – Comparação da musculatura cardíaca com o sinal do ECG.



Fonte: (MANUAL MSD, 2017)

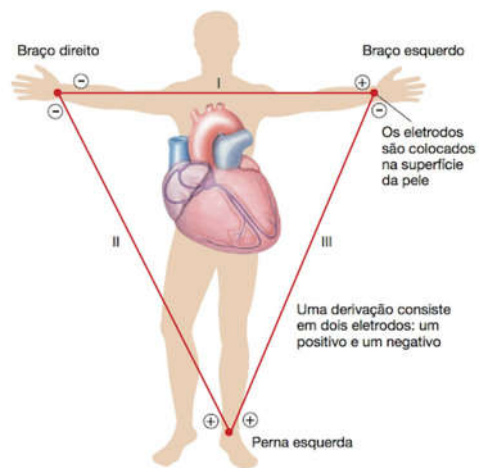
Um eletrocardiógrafo é um aparelho eletrônico cuja função é extrair o ECG do paciente através de diferentes derivações obtidas por eletrodos conectados a diferentes partes do corpo do paciente.

Existem diferentes formas de se posicionar os eletrodos para a extração do ECG através do eletrocardiógrafo, uma das mais comuns delas é a partir do triângulo de Einthoven, que os eletrodos são fixados nos braços do paciente e na perna Esquerda criando três derivações, como mostra a Figura 4.

As três derivações medem a diferença de potencial entre os pontos onde os eletrodos foram fixados.

Assim, a derivação I mede a diferença de potencial entre o braço esquerdo e o braço direito, já a derivação II mede a diferença de potencial entre a perna esquerda e o braço direito, por fim, a derivação III mede a diferença de potencial entre a perna esquerda e o braço esquerdo.

Figura 4 – Triângulo de Einthoven.



Fonte: (ARQUIVO BIOQUI, 2015)

3 METODOLOGIA

O sinal cardíaco do paciente será monitorado e seus dados armazenados em um buffer, e então, assim que os valores do sinal estiverem fora da normalidade, o dispositivo detectará uma anomalia e iniciará o processo de gravação, recuperando as informações armazenadas no buffer e então gravando as novas informações obtidas em tempo real por 24 horas, desta forma, o arquivo possuirá informações do comportamento do sinal elétrico desde antes da anomalia detectada até o fim de sua duração, resultando em mais informação para ser analisada pelo cardiologista. A Figura 5 a seguir exibe um diagrama de blocos com o funcionamento do dispositivo, diagramas de blocos explicando cada etapa do processo podem ser observados nos apêndices deste trabalho.

O processo se inicia pelo bloco Init HW, que representa a inicialização do hardware, determinando valores para *baud rate* e a frequência do timer. Os valores dos filtros passa altas e passa baixas são zerados para evitar problemas decorrentes de valores desconhecidos nos filtros. Uma comparação é feita então para verificar se o dispositivo está pronto para operar, no caso, a variável Status tem que ser diferente de zero. Se essa comparação for verdadeira, significa que as entradas dos filtros foram zeradas e o hardware foi iniciado corretamente.

A partir daí, o dispositivo irá estudar e tratar o sinal na função processa buffer. Nesta função o sinal é filtrado, seus componentes são detectados e comparações são feitas para verificar a normalidade do sinal.

Esse processo possui algumas saídas, a saída *ready* representa a estabilidade do sinal e será acionada assim que todas as posições dos filtros estiverem preenchidas. A saída *OUT* representa o sinal filtrado, e seu valor não pode ser negativo, já que o dispositivo trabalha apenas com valores de tensão positivos. A última saída é a *detect*, que representa a detecção de anomalias no sinal monitorado.

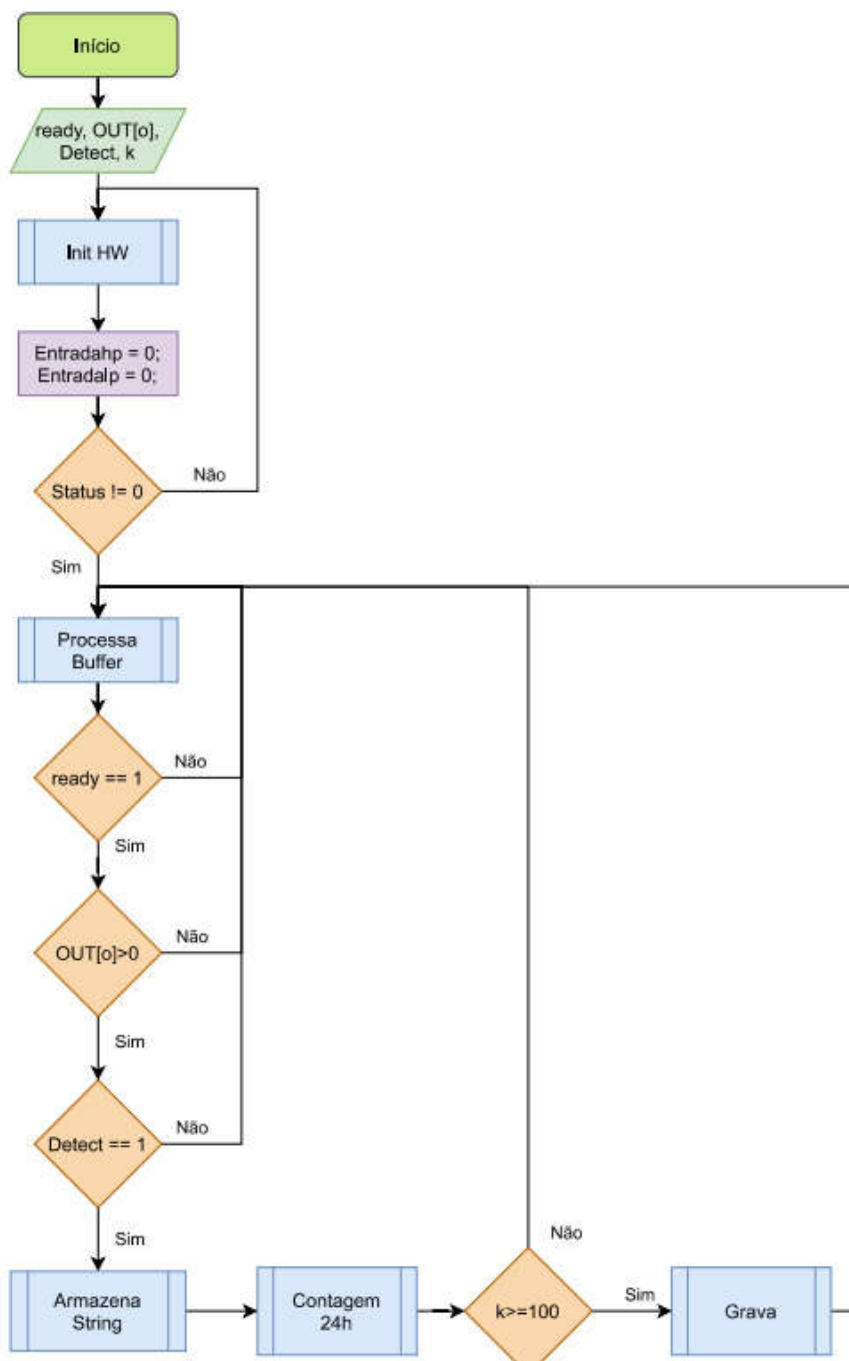
O algoritmo ficará num *loop* processando o buffer enquanto as condições não forem atendidas, que são *ready* = 1, *OUT* > 0 e *detect* = 1.

Se todas as condições forem cumpridas, significa que o sinal já se estabilizou e que algum comportamento anormal foi apresentado pelo sinal.

A partir daí, é iniciado o processo de armazenamento em *string*. O dispositivo irá armazenar os valores do sinal em uma *string* e incrementará o contador das 24

horas e caso haja mais de 100 (k) amostras na *string* gravará essa *string* no cartão SD, zerando-a e reiniciando o processo de armazenamento em string e gravação até que o contador atinja as 24 horas. Neste momento o valor de *detect* é alterado para 0, dando fim à detecção e gravação até que haja outra detecção.

Figura 5 – Fluxograma de funcionamento.



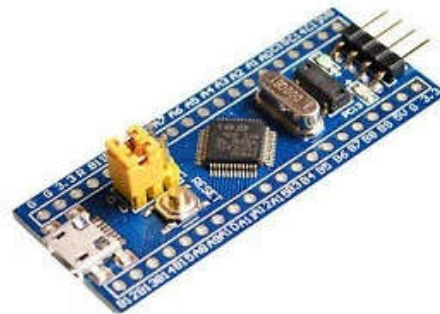
Fonte: Autoria própria

3.1 Microcontrolador

Para a implementação do algoritmo e processamento dos dados foi escolhida uma placa de desenvolvimento STM32F103, também conhecida como Blue Pill. Esta é exibida na Figura 6.

A placa Blue Pill é uma alternativa de baixo custo para a implementação de sistemas embarcados, sendo ideal para esta aplicação já que possui um microcontrolador ARM Cortex-M3 e é compatível com a plataforma de desenvolvimento online MBED que será utilizada para desenvolver o projeto. O STM32F103 possui um conversor analógico digital, ou *Analog to Digital Converter* (ADC) de 12 bits, ou seja, que vai de 0 à 4096 representando as tensões de 0 V à 3,3 V, e com esse conversor, será possível ler e interpretar os valores de tensão obtidos pelo simulador de ECG e com isso avaliar a normalidade do sinal cardíaco por este gerado.

Figura 6 – STM32F103.



Fonte: (STM32DUINO, 2019)

O microcontrolador receberá o sinal do simulador de ECG, através de seu ADC, o interpretará e então caso detecte um sinal anormal ele gravará em um cartão SD este sinal, através do módulo de leitura e escrita de cartão SD que pode ser utilizado com diversos microcontroladores, incluído o STM32F103, este módulo é exibido na Figura 7.

Figura 7 – Modulo de leitura e escrita de cartão SD.



Fonte: (FILIFELOP, 2019)

3.2 Plataforma Online MBED

A forma de programação do firmware do dispositivo será pela plataforma de desenvolvimento MBED, que é um compilador online em C++ da própria ARM para desenvolvimento de sistemas embarcados. Nele a programação se torna mais fácil pois pode-se baixar bibliotecas e exemplos com facilidade.

O código depois de compilado é baixado para o computador e então é gravado no microcontrolador através de um gravador do tipo ST-Link. O ST-Link é um depurador e gravador para os microcontroladores das famílias STM8 e STM32 que possui isolamento digital entre o computador e a placa e sendo capaz de suportar tensões de até 1000 Vrms (STMICROELECTRONICS). O ST-Link pode ser observado na Figura 8.

Para gravar o código compilado, o ST-Link é utilizado em conjunto com o software STLINK-Utility como forma de interface entre o gravador e o computador.

Figura 8 – ST-Link.



Fonte: (ARDUINO-PL, 2020)

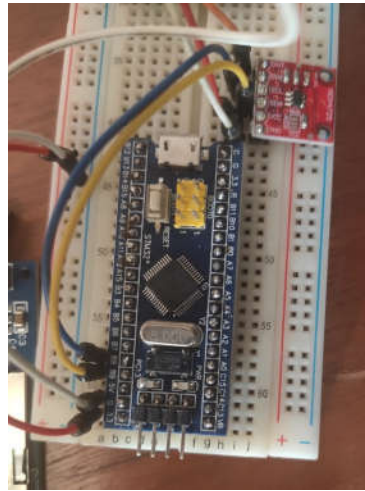
O STLINK-Utility é um software de interface para programação de microcontroladores STM32, ele providencia um ambiente fácil de ler, gravar e verificar a memória. Através dele é possível gravar na placa os códigos compilados no MBED, o software também exibe as informações alocadas em cada endereço da memória do microcontrolador e a possibilidade de apagá-la totalmente.

Uma das vantagens de se utilizar o IDE online MBED é a facilidade de importar bibliotecas, basta pesquisar na plataforma e importar. Sendo assim, a obtenção e utilização da biblioteca de gravação no cartão SD foi muito simples.

3.3 Simulador de ECG

Para analisar os batimentos cardíacos e detectar arritmias é preciso extrair o sinal de ECG do paciente, este processo é normalmente realizado através de um eletrocardiógrafo, porém neste trabalho será utilizado um simulador do sinal de ECG no lugar do eletrocardiógrafo, pois existe a impossibilidade de se obter sinais de ECG anômalos de indivíduos e ao se utilizar um eletrocardiógrafo podem ocorrer ruídos e artefatos no sinal de ECG, como interferências causadas pelo posicionamento incorreto dos eletrodos, movimentação muscular, ou até mesmo falta de gel condutor. Por estes motivos este trabalho utilizou um simulador de ECG contendo trechos de sinais de ECG indicadores de cardiopatia. Com o simulador de ECG todos estes problemas serão contornados e o sinal resultante em teoria deverá ser o mesmo que o sinal proveniente de pacientes com cardiopatias.

Para isso, foi utilizado um simulador de ECG desenvolvido pela turma de iniciação científica (IC) do orientador deste TCC Dr. Eduardo Bertogna utilizando um segundo STM32F103 e um módulo conversor digital-analógico (*Digital to Analog Converter* - DAC), o simulador pode ser observado na Figura 10.

Figura 9 – Simulador de ECG.

Fonte: Autoria própria

No microcontrolador do simulador foi gravado um código que reproduz o sinal de um ECG, que por sua vez é enviado ao DAC, que então terá na saída um sinal analógico simulando um ECG. Este código será lido pelo *Analog to Digital Converter* (ADC) do STM32F103 do detector de arritmias que então interpretará o sinal como um sinal de ECG normal e a partir dele iniciará as etapas de detecção e gravação quando detectar alguma alteração no sinal.

No desenvolvimento do projeto, antes de se utilizar o simulador de ECG, foi utilizado um vetor de testes gravado diretamente do código do detector de arritmia, desta forma, seria possível realizar diversos testes no algoritmo com um sinal que seria sempre constante.

Para verificar os resultados dos testes foi necessário o uso de outro periférico, um módulo de comunicação serial. Este módulo que permitiu que os dados obtidos nos testes fossem exibidos diretamente no computador. O módulo pode ser observado na Figura 11.

Este módulo foi conectado aos pinos GPIO (do inglês: *General Purpose Input/Output*) A10 e A09 do microcontrolador, referentes aos pinos da porta de comunicação serial Serial1, ou seja: RX e TX que representam os pinos associados ao receptor e transmissor da interface USART, respectivamente.

Figura 10 – Módulo de comunicação serial.



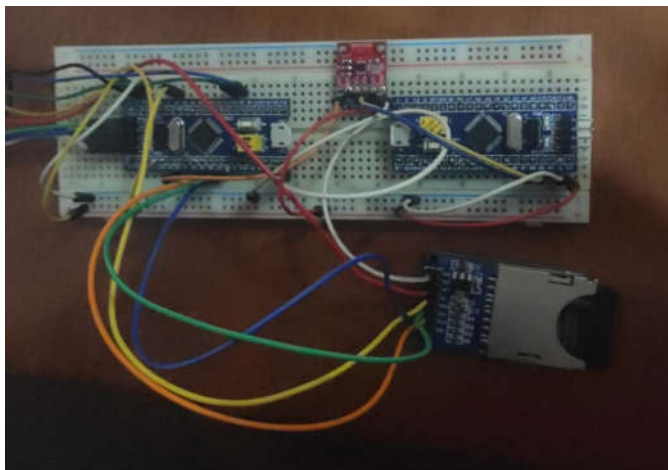
Fonte: (MERCADO LIVRE 2020)

Para observar os valores do sinal de ECG adquiridos pelo ADC do STM32F103 na tela no computador, foi necessária a utilização de um terminal que mostrasse o resultado da comunicação serial. Para isso foi utilizado o programa Tera Term. O Tera Term é um software *open-source* gratuito que emula diferentes tipos de terminais (TERATERM, 2019) e ele possibilitou que os dados de ECG pudessem ser visualizados e salvos para posterior interpretação dos resultados obtidos nos testes deste do algoritmo detector de arritmias desenvolvido para este projeto. Isso foi feito através da função `printf`, assim o terminal pôde exibir os valores adquiridos após o processamento do vetor de ECG, o que permitiu interpretar os resultados do detector e tomar ações diante de tais resultados.

As amostras do sinal simulado foram obtidas na frequência de 240 Hz através do método de estouro de timer, ou seja, o timer é “setado” de forma que após 4,17 ms ele estoure, então o microcontrolador irá parar a operação que ele estiver realizando para coletar uma nova amostra do ADC, essa amostra então é armazenada num vetor de buffer e o STM retoma a operação que ele estava executando anteriormente, a lógica da aquisição do sinal proveniente do simulador pode ser observada no apêndice C o projeto completo pode ser observado na Figura 11.

O microcontrolador fica tratando e analisando o buffer para detectar possíveis arritmias enquanto houver novas amostras. Caso não haja novas amostras, ele irá “dormir” por um determinado tempo, enquanto novas amostras são adquiridas pelo ADC.

Figura 11 – Projeto realizado em protoboard.



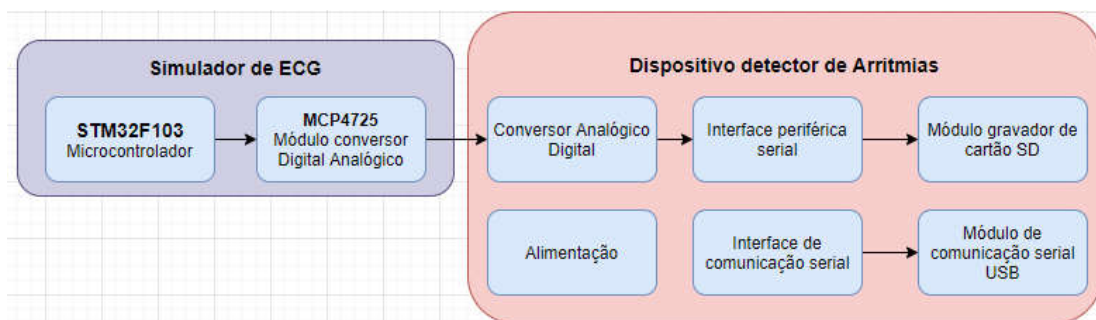
Fonte: Autoria própria

Caso uma suspeita de arritmia seja detectada, o STM32F103 irá iniciar o processo de gravação do sinal no cartão SD, criando um arquivo texto seguido por um índice para que assim possam ser armazenado múltiplos sinais, sempre que uma gravação for concluída, o índice será incrementado.

O procedimento de gravação foi realizado da seguinte forma, assim que o flag que indica a suspeita de arritmia é acionado, os valores do ECG passam a ser armazenados em uma *string*. Após a *string* estar preenchida com 100 amostras, o microcontrolador acessa o cartão SD e grava estes valores em um novo arquivo, e então as gravações passam a ser realizada neste arquivo até que 24 horas tenham decorrido desde a detecção inicial, e com isso a gravação é finalizada e o índice da gravação é incrementado, reiniciando o processo de gravação quando houver uma nova detecção. Tanto o processo de armazenamento em *string*, quanto a gravação e a forma que as detecções das suspeitas de arritmias são feitas podem ser observadas passo a passo nos fluxogramas presentes no apêndice deste documento.

O fluxograma na Figura 12 a seguir mostra de forma simplificada como o projeto foi montado e as interações entre os diferentes blocos que o compõem.

Figura 12 – Diagrama de Blocos da montagem do projeto.



Fonte: Autoria própria.

O simulador de ECG tem seu sinal enviado para o DAC que envia o sinal analógico para a entrada do ADC do STM32F103, esse sinal então é processado pelo algoritmo e enviado ao módulo de leitura e escrita de cartão SD através das portas SPI (*Seria Peripheral Interface*) então é realizada a gravação no cartão SD.

O Bloco de alimentação irá fornecer 3.3 V para o microcontrolador, isso deverá ser feito posteriormente através de bateria, para assim o dispositivo ser portátil., e pô fim a comunicação com o terminal do Tera Term é feita através da interface serial conectada ao módulo de comunicação serial USB.

3.4 Métodos de detecção

O sinal cardíaco possui diversas componentes que necessitam ser estudadas individualmente para podermos analisar a natureza do sinal, se este é um sinal normal ou se possui alguma característica patológica.

Para isso, é necessária a aplicação de alguns métodos que possibilitem a evidenciação dos diversos componentes do sinal e o estudo deles, assim permitindo a classificação do sinal em normal ou possivelmente patológico.

As próximas seções apresentam os métodos que foram utilizados neste projeto para tratamento do sinal e detecção de anomalias.

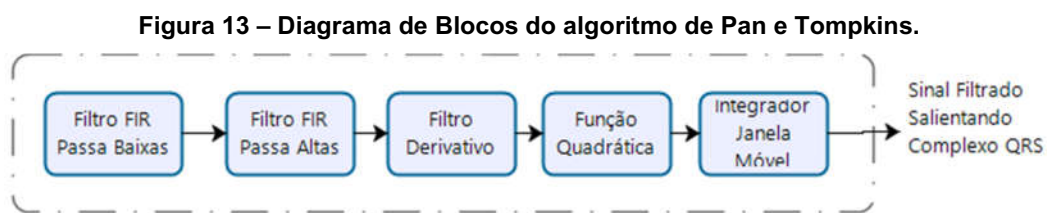
3.4.2 Algoritmo de Pam e Tompkins

A detecção de complexos QRS pode ser difícil, principalmente por causa de ruídos presentes num sinal de ECG causados por movimentação muscular, movimento de eletrodos, interferência da rede elétrica entre outros, e com isso, se torna mais complicada a detecção das arritmias.

O método utilizado foi a implementação do algoritmo de Pam Tompkins que leva em conta esses problemas e utilizando filtragem linear digital, transformação não linear e algoritmos de regra de decisão, é capaz de detectar 99.3% dos complexos QRS. (PAM, J.; TOMPKINS, W. J, 1985)

A primeira etapa do algoritmo é passar o sinal por um filtro passa-faixa com frequências de corte inferior e superior de 5 Hz e 15 Hz, respectivamente, valores que foram recomendados por Pan e Tompkins em seu trabalho para maximizar a energia do complexo QRS e reduzir ao máximo os ruídos (PAM, J.; TOMPKINS, W. J, 1985). Os filtros passa baixas e passa altas foram desenvolvidos no *Software Octave* como filtros digitais FIR através dos comandos “*fir1(20,0.0625)*” e “*fir1(20,0.021,'high')*” respectivamente. Então, para acentuar os máximos e mínimos, o sinal é derivado pela função utilizada no trabalho de Pan e Tompkins “ $H(z) = 0,1(-2z^{-2} - z^{-1} + z^1 + 2z^2)$ ” e elevado ao quadrado ponto a ponto, o que torna todos os pontos positivos, e então integrado por uma janela de média móvel. (PAM, J.; TOMPKINS, W. J, 1985)

A Figura 13 mostra um esquema deste processo, cuja lógica pode ser acompanhada no fluxograma no apêndice E.



Fonte: Autoria própria.

O Algoritmo de detecção desenvolvido baseado no modelo de Pan e Thompkins foi testado utilizando o vetor de testes representando um ECG, e os resultados foram extraídos através dos dados apresentados no terminal do Tera Term.

As figuras 14 a 19 mostram os resultados obtidos entre os passos deste processo.

Como este método suprime todo o sinal, apenas salientando o complexo QRS, a detecção é consistente e resistente a ruído.

Figura 14 - Sinal de ECG não filtrado.

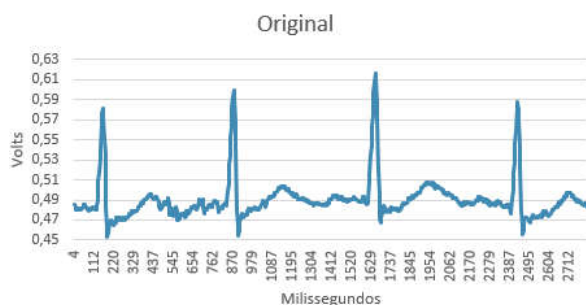


Figura 15 - Sinal após filtro passa baixas.



Figura 16 - Sinal após filtro passa altas.

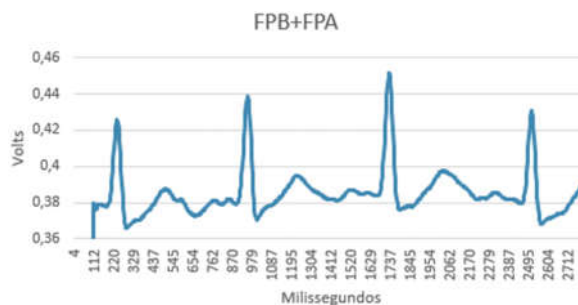


Figura 17 - Sinal após filtro derivativo.



Figura 18 - Sinal após função quadrática.

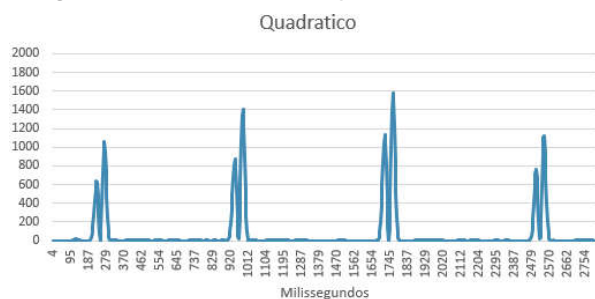
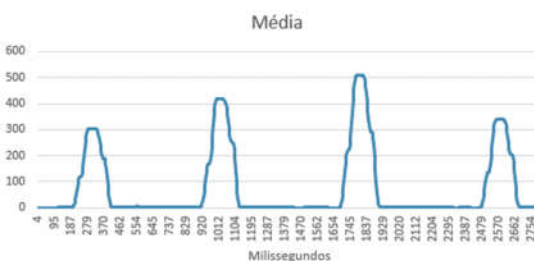


Figura 19 - Sinal após média móvel.



Fonte: Autoria própria

Após esse pré-processamento, são então realizadas as decisões que vão identificar os picos resultantes como componentes R ou não.

3.4.3 Detecção de componentes

A detecção dos picos R se baseou na saída do algoritmo de Pan e Tompkins, de forma que quando o valor do sinal filtrado pelo algoritmo for maior que 290 em binário e sua saída atual for igual à saída anterior com tolerância de 3 pontos para mais ou para menos R será detectado. Isso ocorre apenas nos picos do sinal.

A lógica aplicada para a detecção dos demais componentes do sinal foi a de analisar os pontos do sinal por etapas, sendo que ao passar por um ponto, levanta-se um *flag* indicando o final de um componente do sinal e início de outro, como por exemplo, ao se detectar **R**, o *flag* de detecção do ponto **S** é acionado, então, posteriormente quando o ponto **S** for detectado, esse *flag* será baixado e a *flag* do ponto **T** será levantado e assim sucessivamente, mapeando assim o sinal por partes.

Para a detecção do ponto **Q**, bastou utilizar o sinal pós processado pelo algoritmo de Pan e Tompkins, e assim que o *flag Q* estiver acionado, juntamente com o valor da saída for superior à 100 em binário, ou 80,57 mV em volts, é detectado o ponto **Q**. A mesma lógica foi aplicada ao ponto **R**, sendo que o valor mínimo da saída da janela de média móvel para detecção de R foi de 290 em binário ou 233,64 mV em volts. O pico **R** porém, possui uma peculiaridade, como não é possível que os intervalos entre os batimentos do coração humano sejam inferiores à 200 ms, a *flag* do pico **R** é levantada a cada 48 amostras, ou seja 200,16 ms e só é baixada novamente quando o próximo **R** é detectado. Isso é feito pois ele é o ponto mais consistente de se identificar e sua detecção não se confunde com as dos outros componentes do sinal, portanto, foi determinado que o pico **R** marca o início da detecção do sinal, e caso haja alguma interferência ou erro, e o código fique perdido no mapeamento do sinal, é o pico **R** que vai indicar que um novo ciclo começou, e assim reiniciará o mapeamento. Portanto, caso o detector perca uma detecção de T por exemplo, o algoritmo fará detecções incorretas esperando uma onda T, que já passou. Mas essa incoerência, ocorrerá apenas até o próximo pico R, pois é onde o sinal consegue identificar o pico e “setar” os *flags* da forma que deveriam estar.

Para o Ponto **S** a lógica foi bem simples, o ponto **S** será detectado quando seu *flag* estiver levantado e quando o sinal da saída da janela de média móvel tiver uma tensão inferior a 250 em binário ou 201,42 mV em volts, e que a tensão da saída atual

seja inferior à da amostra anterior, sua lógica pode ser observada no fluxograma presente no apêndice F.

Para o ponto **T** a detecção foi um pouco mais complexa, pois é um componente que tem uma duração envolvida, diferente do complexo **QRS** que são apenas três pontos de uma amostra cada.

A detecção foi realizada da seguinte forma, a tensão 0 V do sinal foi estipulada a partir da tensão do ponto S somada com 25 binário ou 20 mV, então, o algoritmo observa a onda enquanto a tensão fosse crescente, ou seja, a amostra atual maior que a anterior e acima de 0 V, e a partir do momento que a amostra atual se tornar inferior à amostra anterior detecta-se o pico da onda **T**, então a amplitude é calculada pela diferença do ponto 0 e do pico de **T**.

Após **T** detectado, a tensão atual é extraída novamente e o *flag P* é acionado, indicando o final da onda **T** e início da **P**.

Com a tensão final de **T**, é possível detectar o início da onda **P**, já que a tensão entre o início de **P** e o final de **T** se mantém relativamente constante, a detecção é descartada caso não haja variação significativa na tensão.

Como a maior parte das detecções das arritmias deste projeto se baseia na duração e na amplitude da onda **P**, sua detecção foi dada da seguinte forma: um contador de amostras é iniciado assim que a flag P é acionada, e ele é incrementado a cada ciclo. A cada ciclo o algoritmo compara a amostra atual com a anterior, enquanto ela for maior, ele entende que está em uma crescente. A partir do momento que o valor de tensão da amostra atual passa a ser menor que o da amostra anterior, o algoritmo interpreta como se a onda chegou no seu pico, e o valor atual de tensão é extraído como amplitude de P. O código então continua mapeando a duração, desta vez invertendo a lógica, já que a onda está agora numa decrescente. Assim enquanto a amostra atual for inferior à amostra anterior o contador é incrementado, e a partir do momento que a amostra atual se tornar maior ou igual a anterior, interpreta-se o fim da onda e finaliza-se a medição da duração da onda.

Os Fluxogramas das detecções de cada componente estão presentes individualmente no apêndice deste trabalho.

3.4.4 Comparação de valores

Um coração saudável possui valores relativamente constantes de amplitude e duração das componentes do seu sinal de ECG. Estes valores podem variar de pessoa para pessoa, mas há um limite entre variações de valores entre pacientes e valores que caracterizam um batimento anômalo. Assim, considerando um valor limite para tensão e duração dos componentes do ECG, pode se determinar se o coração apresenta comportamento patológico.

O método de detecção utilizado foi baseado nos valores de duração e amplitude da onda P tão como a amplitude de R e T em relação as suas médias dos últimos 10 batimentos. Os valores considerados normais para amplitude e duração da onda P são 0,25 mV e 0,1 s, respectivamente (MORSCH, J. A., 2018). Caso algum dos componentes do sinal de ECG monitorado apresentar valores 20% acima dos valores considerados, o sinal será considerado como uma possível arritmia, e então iniciará o processo de gravação.

Ou seja, a cada ciclo cardíaco, os valores de pico de P, T, e R serão obtidos e comparados com valores limites, por exemplo, se a amplitude de pico de P estiver acima de 0,28 mV ou sua duração superior a 0,13 s será detectada uma suspeita de arritmia e iniciará o processo de gravação, estes valores foram determinados à partir de observação e testes realizados ao decorrer do trabalho. Da mesma forma isso se aplica a T e R, sendo que quando a amplitude de T for 20% maior ou menor que às médias das ondas T anteriores, ou R for 20% maior ou menor que a média dos últimos 10 picos R, será indicada uma suspeita de arritmia e isso dará início ao processo de gravação.

Este método, apesar de ser simples, permite detecções de várias anomalias diferentes, porém devido a essa simplicidade ele se torna suscetível à detecção de falsos positivos, diminuindo sua eficácia. A tabela a seguir mostra todos os critérios utilizados para iniciar a gravação do sinal.

Tabela 1 – Condições de detecção

Critério	Condição
Onda P negativa	P tem valor de tensão decrescente ao invés de Crescente
Amplitude de P acima do limite	P tem valor de tensão de pico acima de 0,28 mV
Duração de P acima do limite	P tem valor de duração acima de 0,13 s
Amplitude de R acima ou abaixo da média	R tem valor de tensão 20% acima ou abaixo da média dos últimos 10 valores de R
Amplitude de T acima ou abaixo da média	T tem valor de tensão de pico 20% acima ou abaixo da média dos últimos 10 valores de T

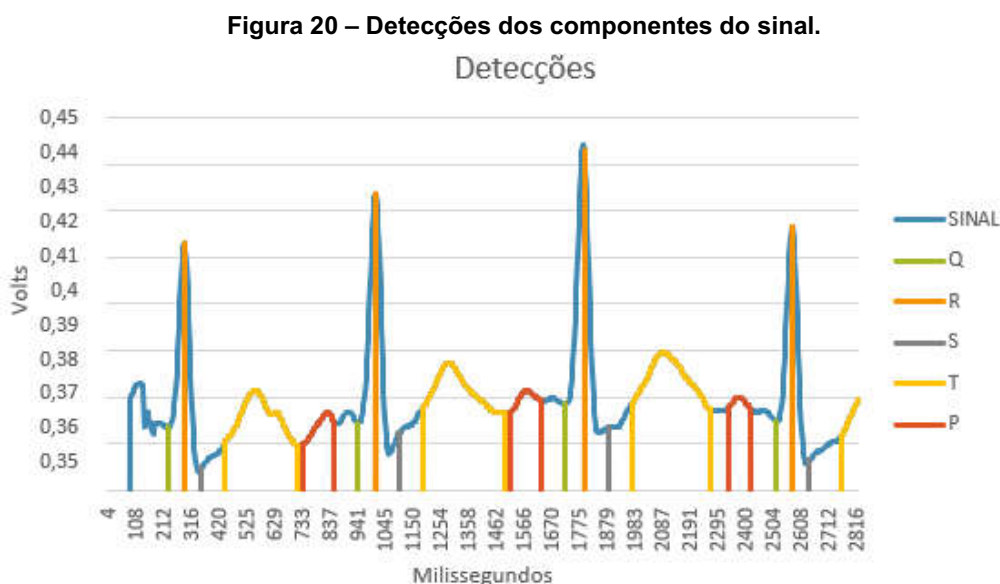
Fonte: Autoria própria

4 RESULTADOS

Os testes foram realizados utilizando o algoritmo desenvolvido, porém os valores detectados foram imprimidos diretamente no terminal do Tera Term, já que o computador utilizado para o desenvolvimento do projeto não conta com um leitor de cartão SD, alternativamente, pode-se utilizar o comando “*fread*” para obter o vetor do sinal, fazendo o microcontrolador ler o arquivo e exibi-lo no terminal, porém, cada vez que um arquivo novo é criado o nome muda, tanto no índice como no tipo de detecção, o que dificulta a leitura, por este motivo, a gravação no cartão SD foi testada uma única vez para verificar seu funcionamento e o resto dos testes foram realizados apenas utilizando o terminal.

O resultado das detecções dos componentes do sinal pode ser observado na Figura 20 com cada componente do ECG ressaltada no vetor de testes.

Os resultados apresentados foram considerados satisfatórios após a observação dos valores detectados e a comparação deles com os componentes de um ECG normal. Isso permitiu seguir com a detecção dos componentes e de arritmias no sinal do simulador de ECG.

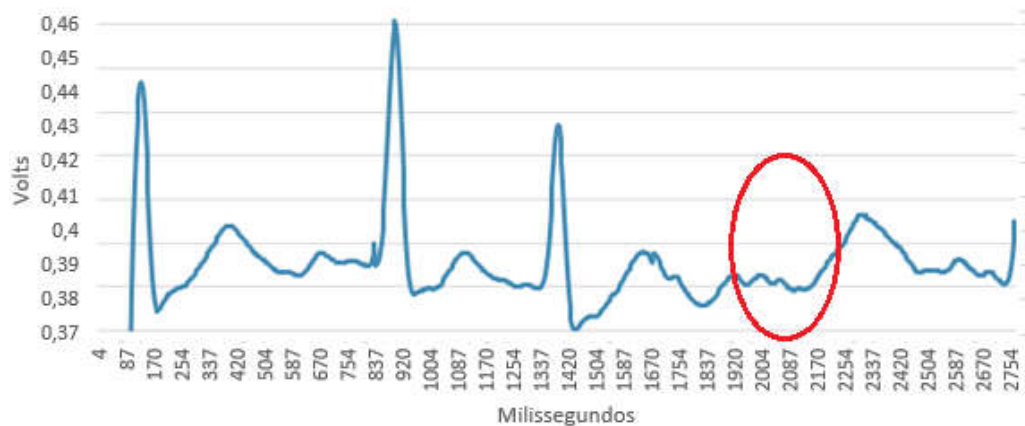


Fonte: Autoria própria

Após realizados alguns testes com o algoritmo desenvolvido, o sinal do ECG proveniente do simulador foi caracterizado como arritmia e gravado no cartão de

memória, alguns dos sinais gravados mostram casos claros de detecção segundo o algoritmo, como por exemplo a falta do Pico R em determinado batimento como exibido nas figuras 21 e 22.

Figura 21 – Sinal 1 obtido com ausência de pico R.



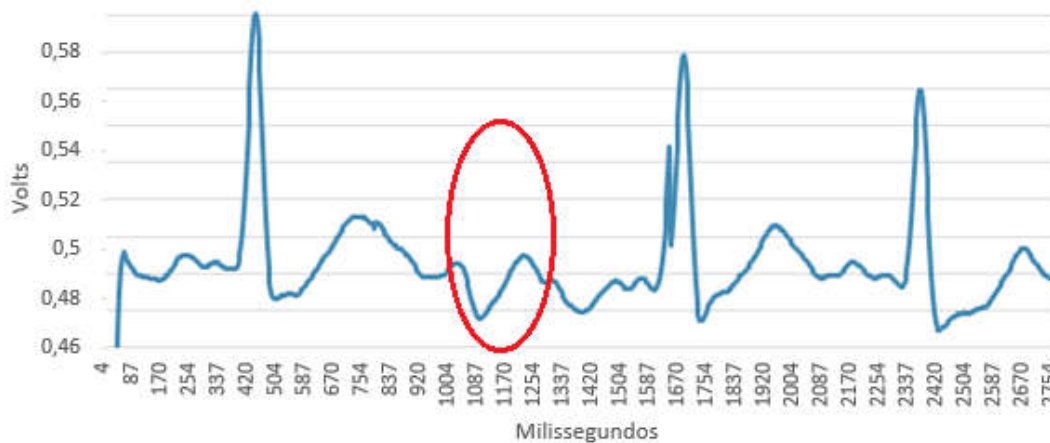
Fonte: Autoria própria

Na Figura 21, o sinal tem sua amplitude reduzida por conta do filtro passa faixas, essa atenuação foi corrigida aplicando um fator de multiplicação 1,275 vezes no sinal de saída, o que tornou sua amplitude bem próxima da do sinal não filtrado. O sinal exibido na Figura 22 já tem esse fator aplicado, o que pode evitar detecções incorretas por amplitude da onda p, já que sua amplitude é o critério em que a detecção se baseia mais fortemente.

Pode-se observar em ambas as figuras a ausência do pico R, nas regiões destacadas em cada figura.

Essa ausência do pico R, caracteriza uma suspeita de arritmia, já que o código tenta encontrar o pico R para realizar a medição, mas o intervalo limite é atingido e não há registro de valores que caracterizem um pico R, assim, o código fica "perdido" e por não conseguir interpretar o código, ele o registra como uma suspeita de arritmia.

Figura 22 – Sinal 2 Obtido com ausência do pico R.



Fonte: Autoria própria

Aumentando o número de amostras exibidas, pode-se observar mais anomalias no sinal, a Figura 23 mostra um vetor com 1200 posições.

Figura 23 – Sinal obtido com 1200 amostras.



Fonte: Autoria própria

Este sinal foi considerado como arritmia pelo dispositivo, e nele pode-se observar diversas anomalias, cada um dos comportamentos anormais será discutido a seguir.

1. Onda P Negativa

Como mencionado anteriormente, a onda P negativa é um dos critérios que o algoritmo considera uma arritmia, e provavelmente é o caso deste sinal em particular.

Após a detecção do final da onda T, o código considera a tensão atual como nula, ou seja, o final de cada onda T marca uma nova tensão zero. O ponto 1 mostra o instante no sinal onde a onda P está abaixo da tensão considerada como zero, o que caracteriza uma onda P negativa e, portanto, uma arritmia.

2. Onda T Ausente.

Outro comportamento anormal que o algoritmo considera uma suspeita de arritmia é a baixa amplitude da onda T. Neste código, existe a ausência total da onda T em determinado momento, exibido no ponto 2. O programa então considera o sinal após o ponto S como onda T, mas ele encontra outro pico R antes de detectar uma onda P, o que faz o programa “se perder” no sinal, apontando que isso não pode ser um batimento normal e o classifica como suspeita de arritmia.

3. Baixa amplitude do pico R.

Em determinado momento neste sinal, o valor do pico R é muito menor que a média das amplitudes dos demais picos, o que faz o algoritmo detectar o sinal como uma suspeita de arritmia.

A amplitude observada no ponto 3 é certamente menor que 80% da média dos picos anteriores, e, portanto, entra na classificação de suspeita de arritmia.

Pode-se perceber que os sinais obtidos realmente possuem anomalias que o código considera como possíveis problemas cardíacos, a questão é se esses sinais deveriam ou não ter sido recebidos pelo dispositivo, já que o sinal é apenas uma simulação gerada por outro micro controlador.

Além do sinal patológico na programação do simulador, também há outras possíveis causas da obtenção desses sinais irregulares como por exemplo as impedâncias parasitas resultantes da utilização de jumpers e protoboard ou o filtro que foi desenvolvido.

5 CONCLUSÃO

Durante o desenvolvimento do projeto, pode-se perceber que o sinal elétrico cardíaco é bem complexo e as detecções de cada uma de suas componentes pode ser demasiadamente difícil se levarmos em conta as variações na forma do sinal do coração de cada paciente e a possibilidade de ruídos externos. Apesar desta dificuldade, o algoritmo de Pan e Tompkins se mostrou muito eficaz na remoção de ruídos e detecção do pico R.

A detecção dos demais componentes se mostrou mais complexa, e foi implementada de forma simples, e como a componente em que a detecção mais se baseia é a onda P, que é a menos expressiva do sinal podendo ser deformada com mais facilidade por ruídos, foi adotada uma forma mais genérica de detecção das arritmias, o que infelizmente, contribuiu para resultados pouco consistentes.

O algoritmo em si, possui certa capacidade de detecção de anomalias, como observado nos resultados, porém devido à forma que o projeto foi desenvolvido, utilizando protoboard e jumpers, os resultados se mostraram inconsistentes.

A detecção foi feita de forma a evitar falsos negativos, já que numa situação de arritmia, um falso negativo pode ser muito pior para o paciente do que um falso positivo, sendo assim, foi adotada a classificação de suspeita nos casos em que os componentes do sinal não são claros o bastante para indicar uma arritmia certa, e caberá ao profissional de saúde determinar se estas suspeitas seriam de fato patologias ou apenas um batimento regular mais rápido ou mais lento que o normal. Porém por conta da lógica de detecção demasiadamente simples utilizando flags para mapear o sinal e da forma que o projeto foi montado, a confiabilidade desta detecção é questionável, e no estado atual o projeto não possui condições de substituir um Holter, diversas melhorias devem ser feitas no projeto para que ele tenha condições de detectar arritmias de forma confiável, como por exemplo a aplicação de um método de detecção mais consistente, como por exemplo a utilização de logica Fuzzy, matrizes Markovianas, ou outros métodos de detecção de padrões. Para isso seria necessária o estudo mais aprofundado da aplicação destes metodos em detecções de Arritmia. E por fim, uma placa de circuito impresso para evitar ruídos, interferências e impedâncias que estão presentes na montagem em protoboard com jumpers. Infelizmente, o projeto pecou nesta área devido à realidade pela qual passou, sendo

realizado durante uma pandemia nacional e tendo uma orientação quase que exclusivamente à distância.

Os objetivos do projeto em si foram alcançados, esses eram: a implementação de um algoritmo que possa identificar as diferentes componentes do ECG, comparar os valores destas componentes com valores adotados como limite de um batimento saudável e por fim, em caso de discrepância gravar o sinal em uma memória externa, porém isso de forma alguma significa que esta detecção seja confiável.

Por fim, o projeto ainda tem possibilidades de diversas melhorias, como por exemplo o registro do momento em que ocorreu a detecção, um botão que o paciente possa apertar em caso de mal estar, um alarme sonoro em caso de detecção, bateria para permitir a portabilidade do equipamento, entre outros.

REFERÊNCIAS

- ARDUINO-PL. **Programator ST-LINK/V2 (Compatible) Dla STM32 I STM8.** Disponível em: <https://arduino-pl.com/akcesoria-do-arduino/315-programator-st-linkv2-compatible-dla-stm32-i-stm8.html>. Acesso em 09 de dezembro de 2020.
- BERTONHA, E.G. **Gravador Holter Digital.** 1994. Tese (Mestrado em Engenharia Elétrica) - Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, 1994.
- BHARATI, Deokar M. Detecting Cardiac Arrhythmia by Extracting ECG Features. **International Journal of Scientific & Engineering Research**, Volume 5, Issue 5, May-2014.
- FILIFELOP. **Módulo Cartão SD Card.** FilipeFlop, Disponível em: https://www.filipeflop.com/produto/modulo-cartao-sd-card/?gclid=CjwKCAjwkPX0BRBKEiwA7THxiN391D88X9dzi0an6bNap0HLBrmL6cu60mbd2JG1nYiPdhyGfy8pGRoCeUUQAvD_BwE Acesso em 10 de novembro de 2019.
- GUYTON, Arthur C.; HALL, John E. **Tratado de fisiologia médica.** Rio de Janeiro, RJ: Elsevier, 2011. xxi, 1151 p. ISBN 9788535237351
- HENRIQUE, R. **Triângulo de Einthoven.** Arquivo BioQui. Disponível em: <http://arquivobioqui.blogspot.com/2015/11/triangulo-de-einthoven.html> Acesso em 17 de outubro de 2019.
- IOANNOU K., IGNASZEWSKI M., MCDONALD I. Ambulatory electrocardiography: The contribution of Norman Jefferis Holter, **BC MEDICAL JOURNAL**, março de 2014.
- LO4D.com **TERA TERM**, dezembro 2019, Disponível em: <https://tera-term.en.lo4d.com/windows>. acesso em 18 de maio de 2020.
- OGANIZAÇÃO PAN AMERICANA DA SAÚDE. **Doenças Cardiovasculares.** OPAS, 2017, Disponível em: https://www.paho.org/bra/index.php?option=com_content&view=article&id=5253:dododoe-cardiovasculares&Itemid=1096, Acesso em 13 de maio de 2020.
- PAM, J.; TOMPKINS, W. J. A Real-Time QRS Detection Algorithm, **IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING**, VOL. BME-32, NO. 3, MARCH 1985.
- SANTOS, D. **Testes Sobre Sistema Circulatório II**, Blog do Professor Djalma Snatos. Disponível em: <https://djalmasantos.wordpress.com/2016/12/09/testes-sobre-sistema-circulatorio-ii/> Acesso em 12 de outubro de 2019

SHEA, M.J. **Eletrocardiografia**. Manual MSD. Disponível em: <https://www.msmanuals.com/pt-pt/profissional/doen%C3%A7as-cardiovasculares/exames-e-procedimentos-cardiovasculares/eletrocardiografia> acesso em 12 de outubro de 2019.

SHEA, M.J. **Eletrocardiograma (ECG)**. Manual MSD. Disponível em: <https://www.msmanuals.com/pt-pt/casa/dist%C3%BArbios-docora%C3%A7%C3%A3o-e-dos-vasos-sangu%C3%ADneos/diagn%C3%B3stico-de-dist%C3%BArbios-docora%C3%A7%C3%A3o-e-dos-vasos-sangu%C3%ADneos/eletrocardiograma> Acesso em 12 de outubro de 2019.

SOCIEDADE BRASILEIRA DE ARRITMIAS CARDIACAS. **SOBRAC reúne cardiologistas nacionais e internacionais em porto alegre**. SOBRAC. Disponível em: https://sobrac.org/cbac2016/wp-content/uploads/2016/02/1479831487_Press_Release_CBAC2016_Aprovado.pdf Acesso em 13 de maio de 2020.

STM32DUINO. **Blue Pill**. STM32duino. Disponível em: https://stm32duinoforum.com/forum/wiki_subdomain/index_title_Blue_Pill.html Acesso em 22 de novembro de 2019.

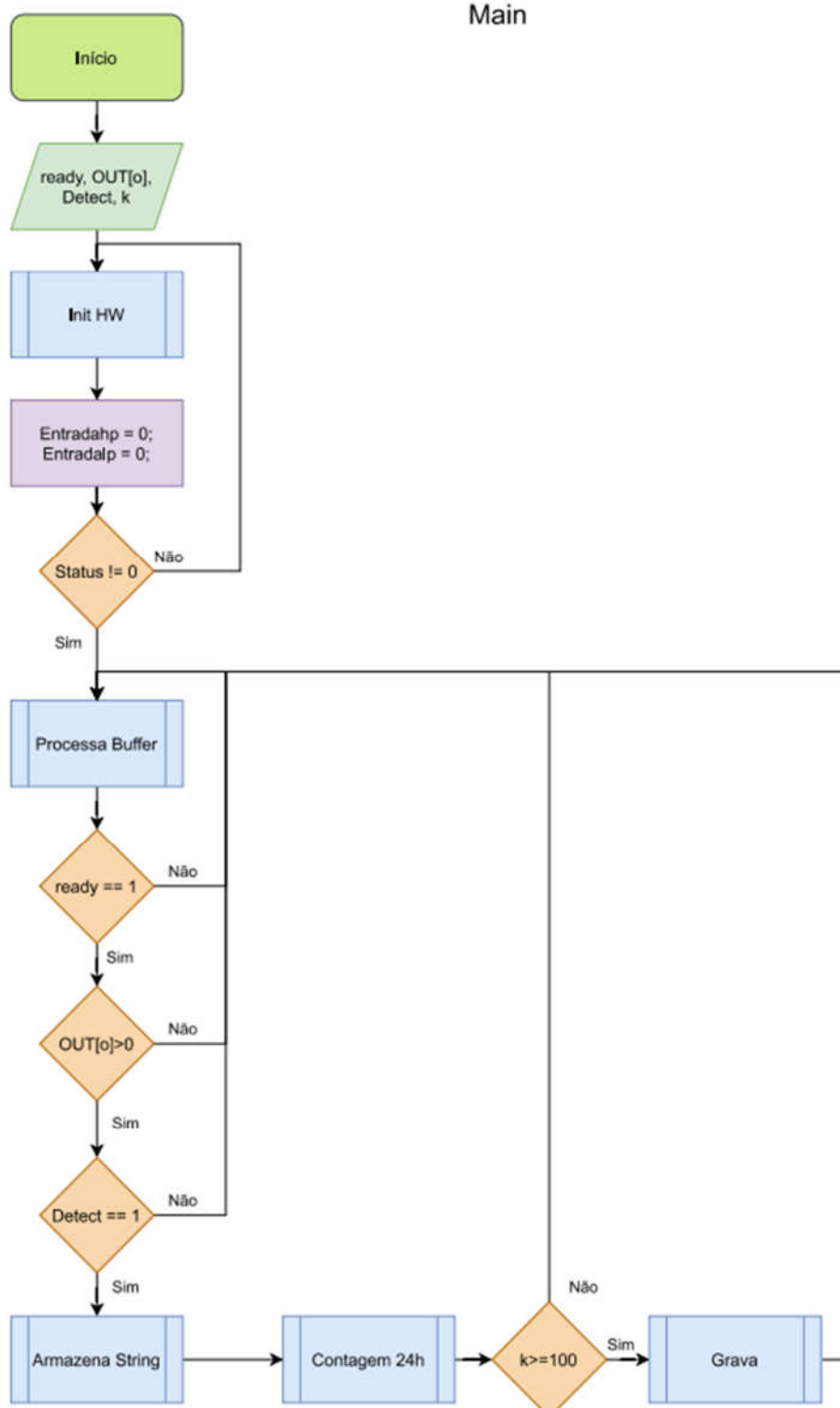
STMicroelectronics **ST-LINK/V2 Data brief**, abril 2020. Disponível em: https://www.st.com/resource/en/data_brief/st-link-slash-v2.pdf Acesso em 18 de maio de 2020

STMicroelectronics **STSW-LINK004**. Disponível em: <https://www.st.com/en/development-tools/stsw-link004.html> Acesso em 18 de maio de 2020

TELEMEDICINA MORSCH. **Ondas do Eletrocardiograma: Como funciona o ECG e como interpretar**. Disponível em: <https://telemedicinamorsch.com.br/blog/ondas-do-eletrocardiograma#:~:text=Ondas%20do%20eletrocardiograma%20normais&text=Nele%2C%20cada%20batimento%20possui%20uma,ritmo%20padr%C3%A3o%2C%20ou%20ritmo%20sinusal>. Acesso em 27 de outubro de 2020.

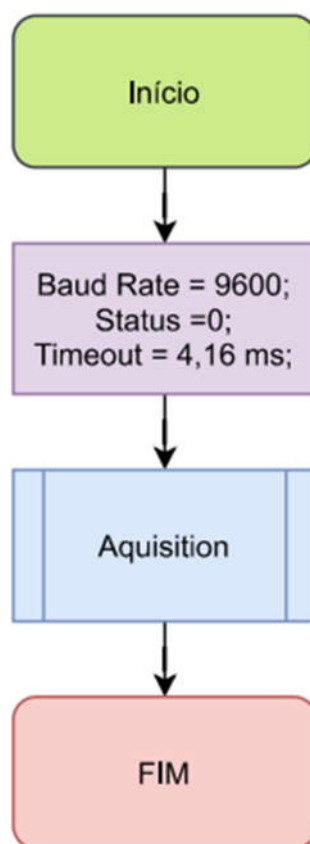
TOULSON, R.; WILMSHURST, T. **Fast and Effective Embedded Systems Design**. Oxford OX5 1GB, United Kingdom, 2012.

APÊNDICE A – ROTINA MAIN



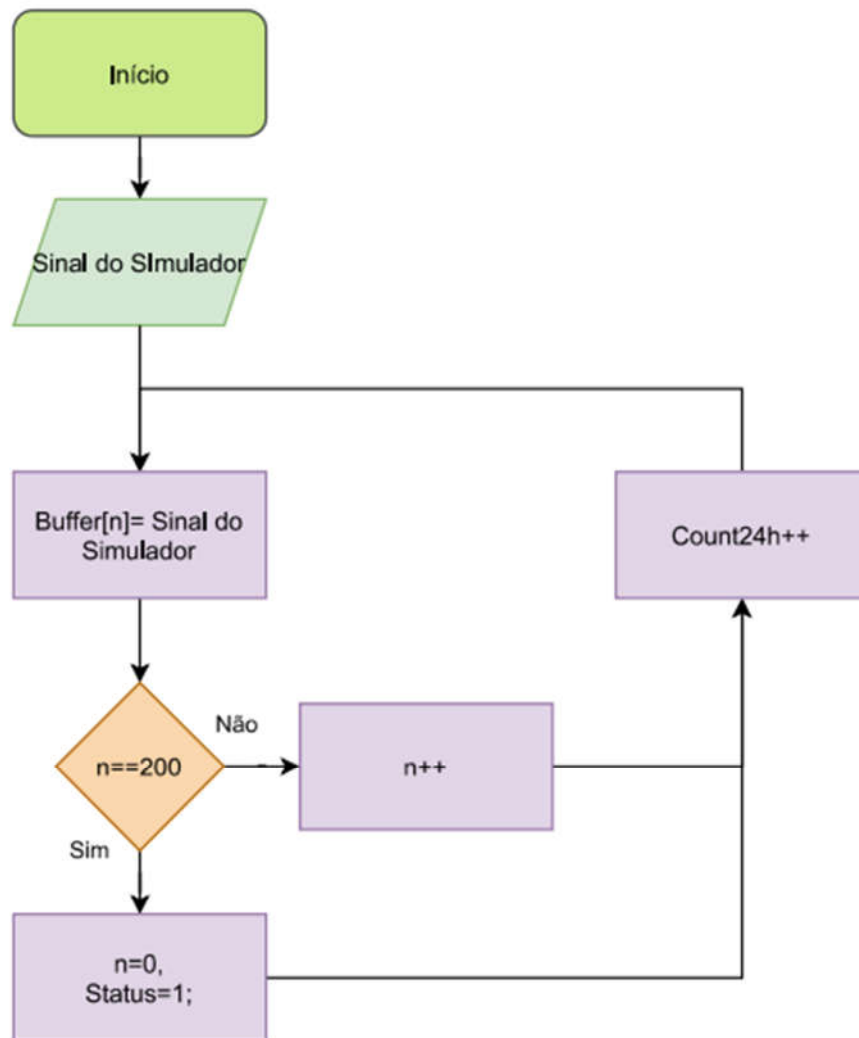
APÊNDICE B – ROTINA INIT HW

Init HW



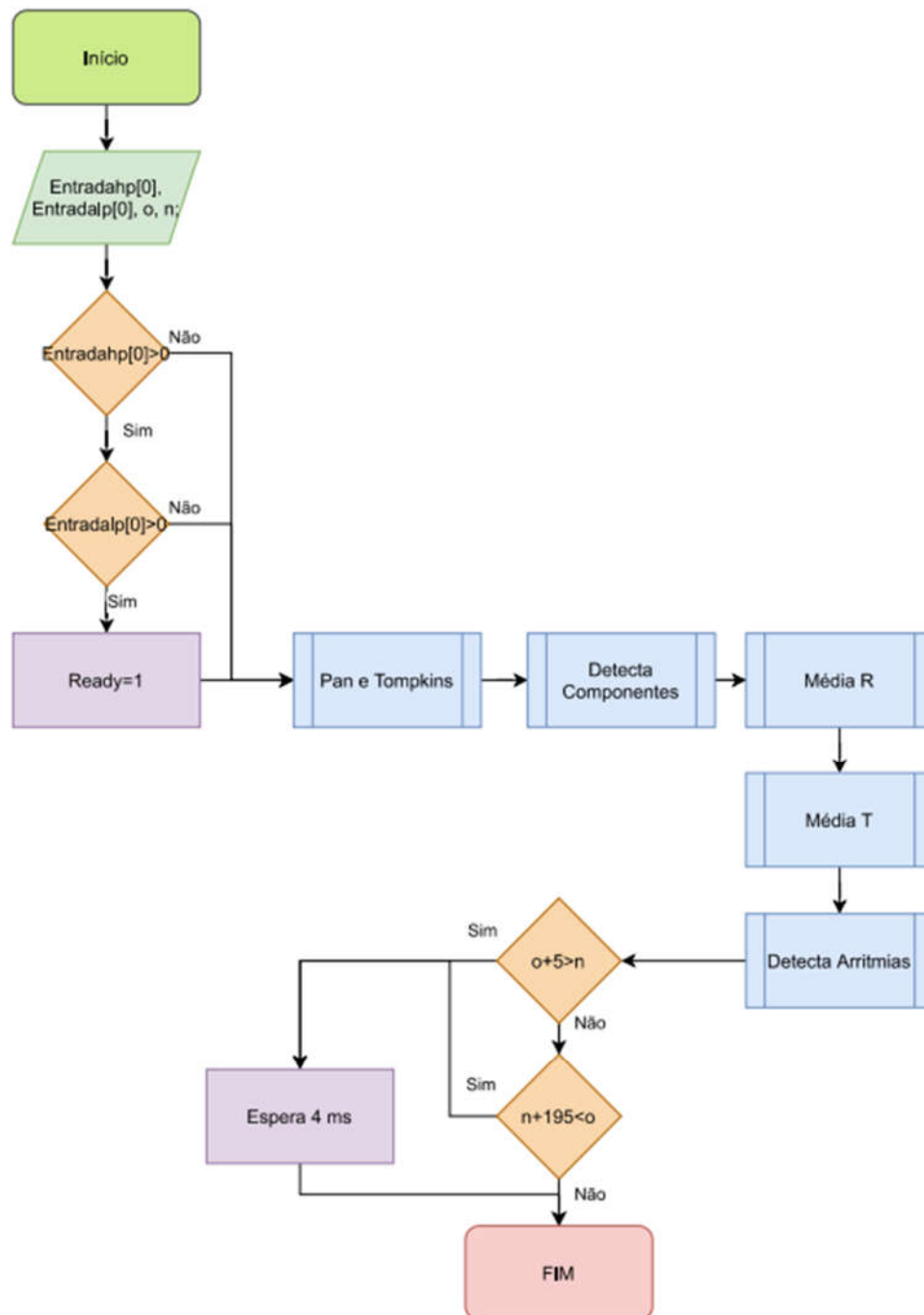
APÊNDICE C – ROTINA DE INTERRUPÇÃO POR TIMEOUT

Interrupção por Timeout Aquisition



APÊNDICE D – ROTINA DE PROCESSAMENTO DE BUFFER

Processa Buffer

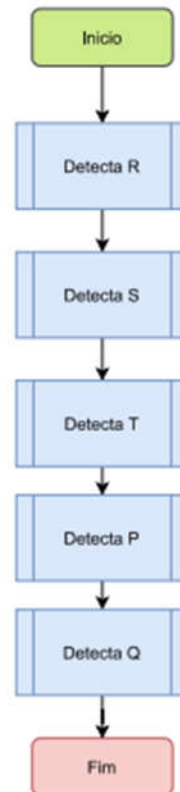


APÊNDICE E – ROTINAS PAN E TOMPKINS E DE DETECÇÃO DE COMPONENTES

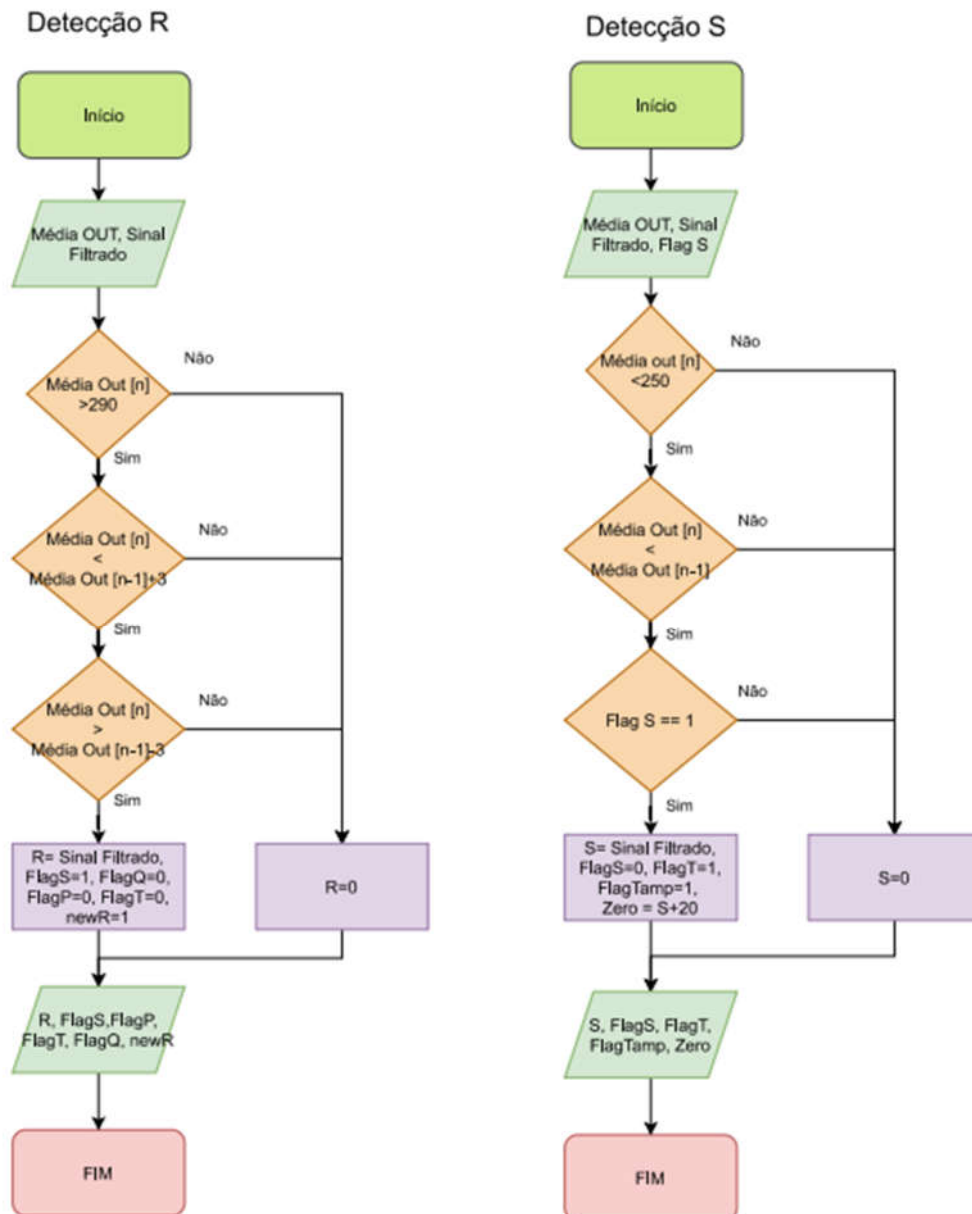
Pan e Tompkins



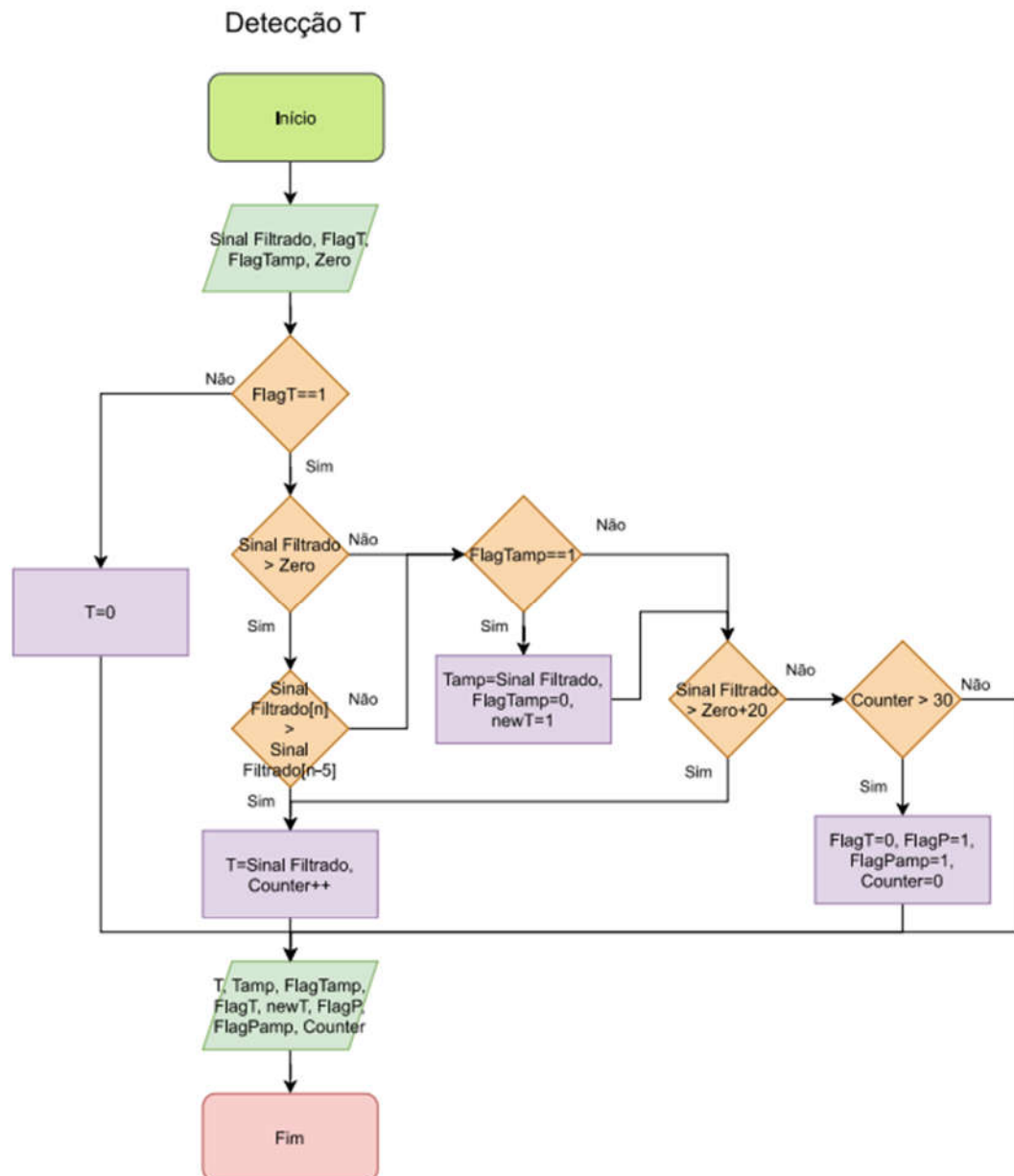
Detecta Componentes



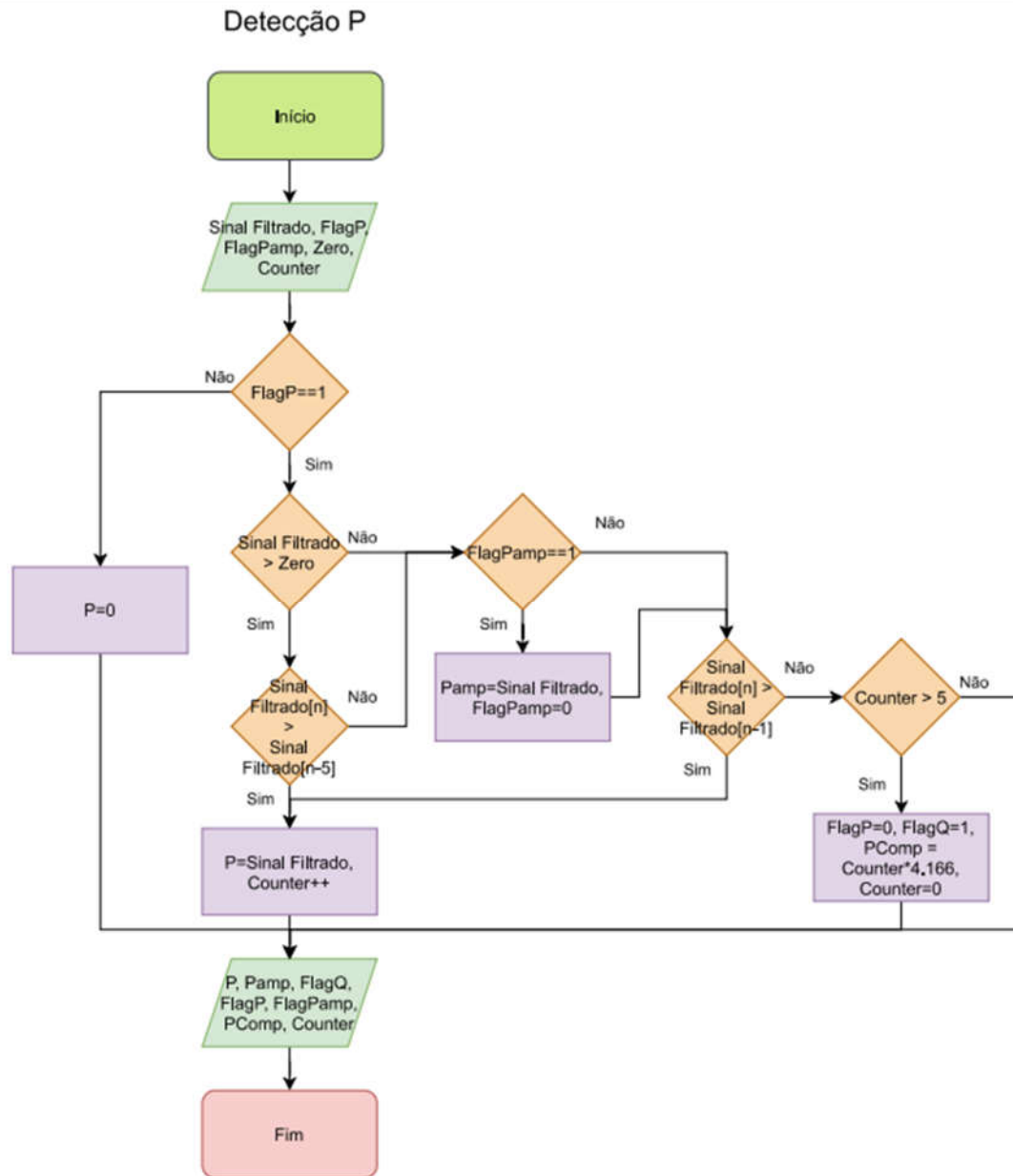
APÊNDICE F – ROTINAS DE DETECÇÃO DE COMPONENTES R E S



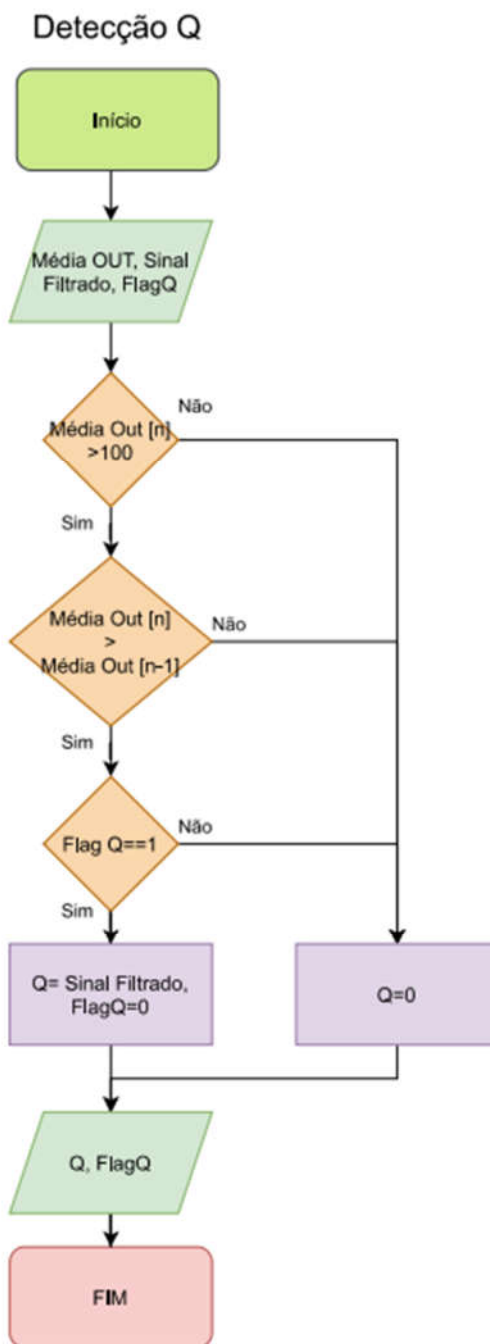
APÊNDICE G – ROTINA DE DETECÇÃO DE COMPONENTE T



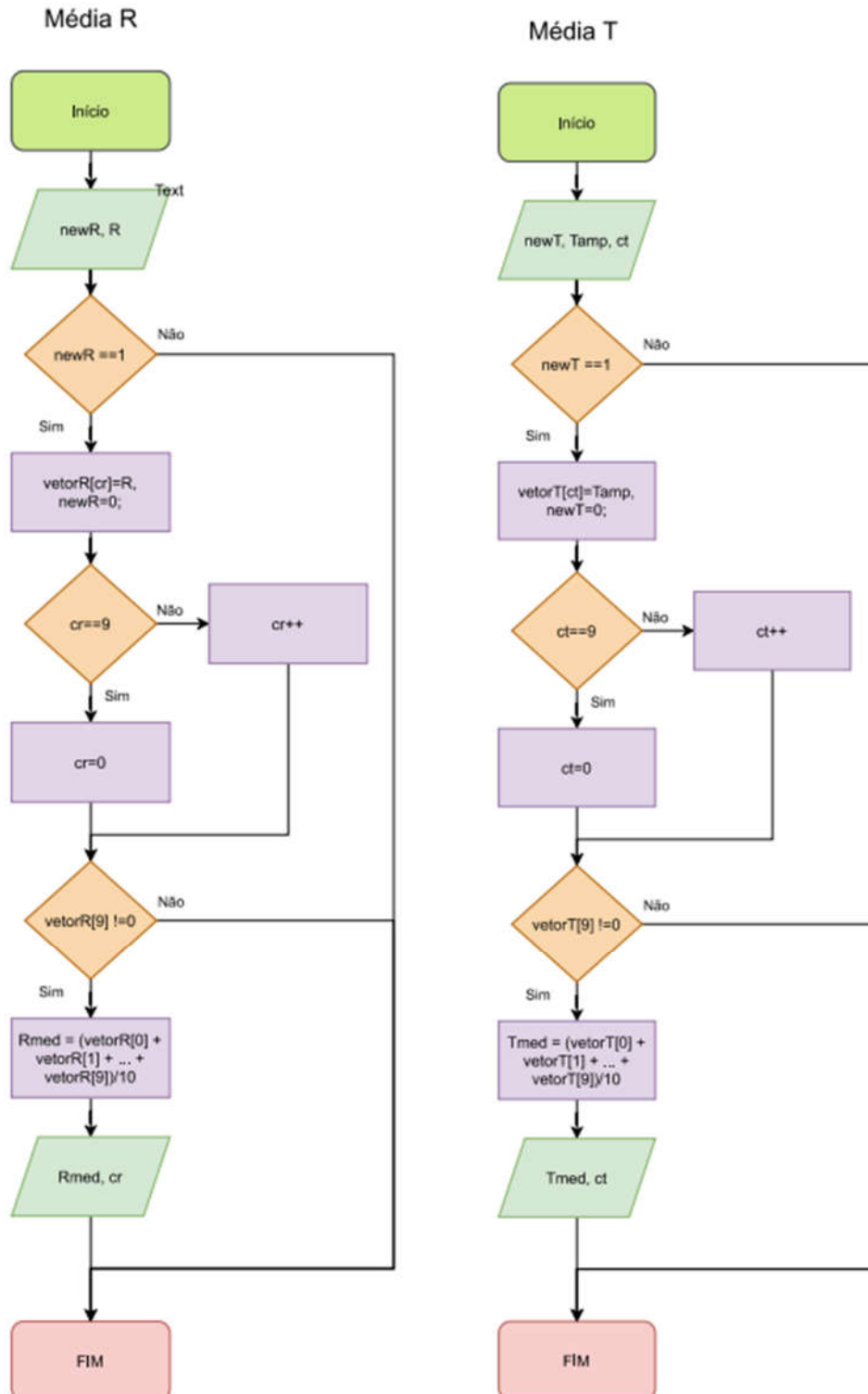
APÊNDICE H – ROTINA DE DETECÇÃO DE COMPONENTE P



APÊNDICE I – ROTINA DE DETECÇÃO DE COMPONENTE Q

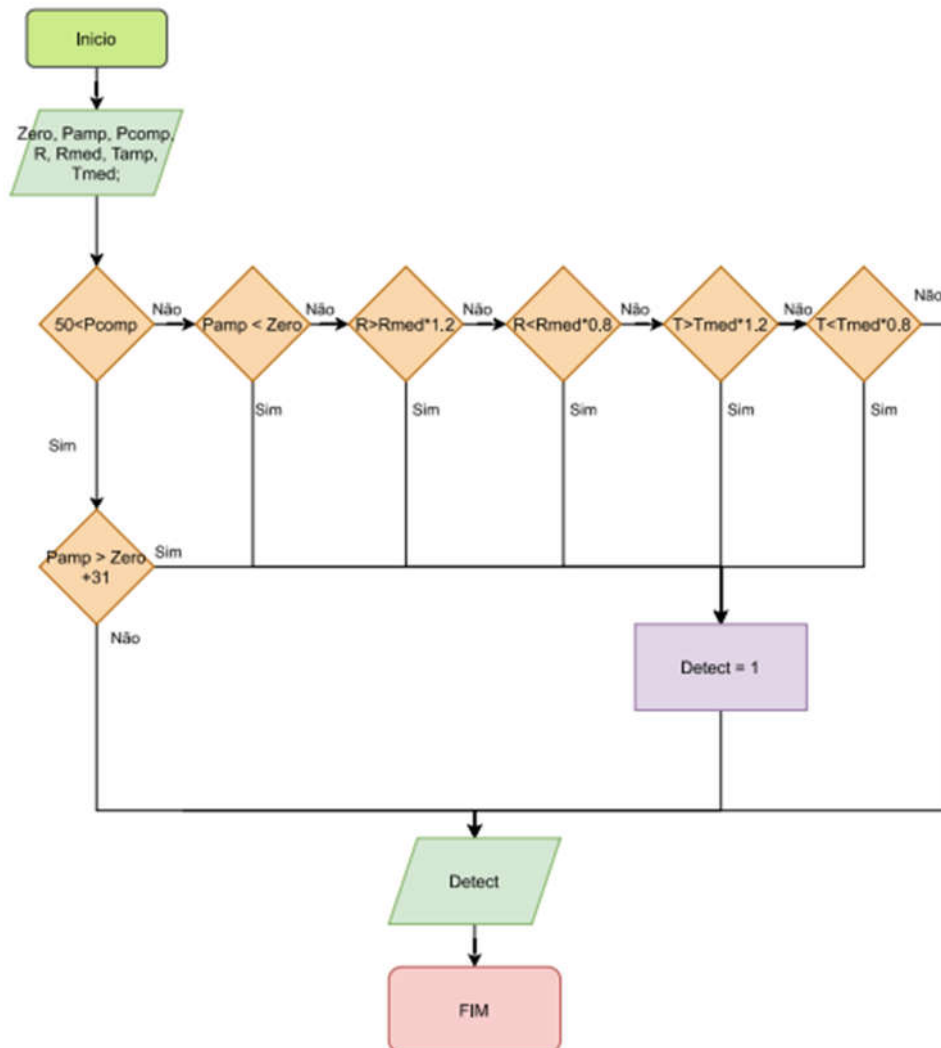


APÊNDICE J – ROTINAS DE CÁLCULO DAS MÉDIAS R E T



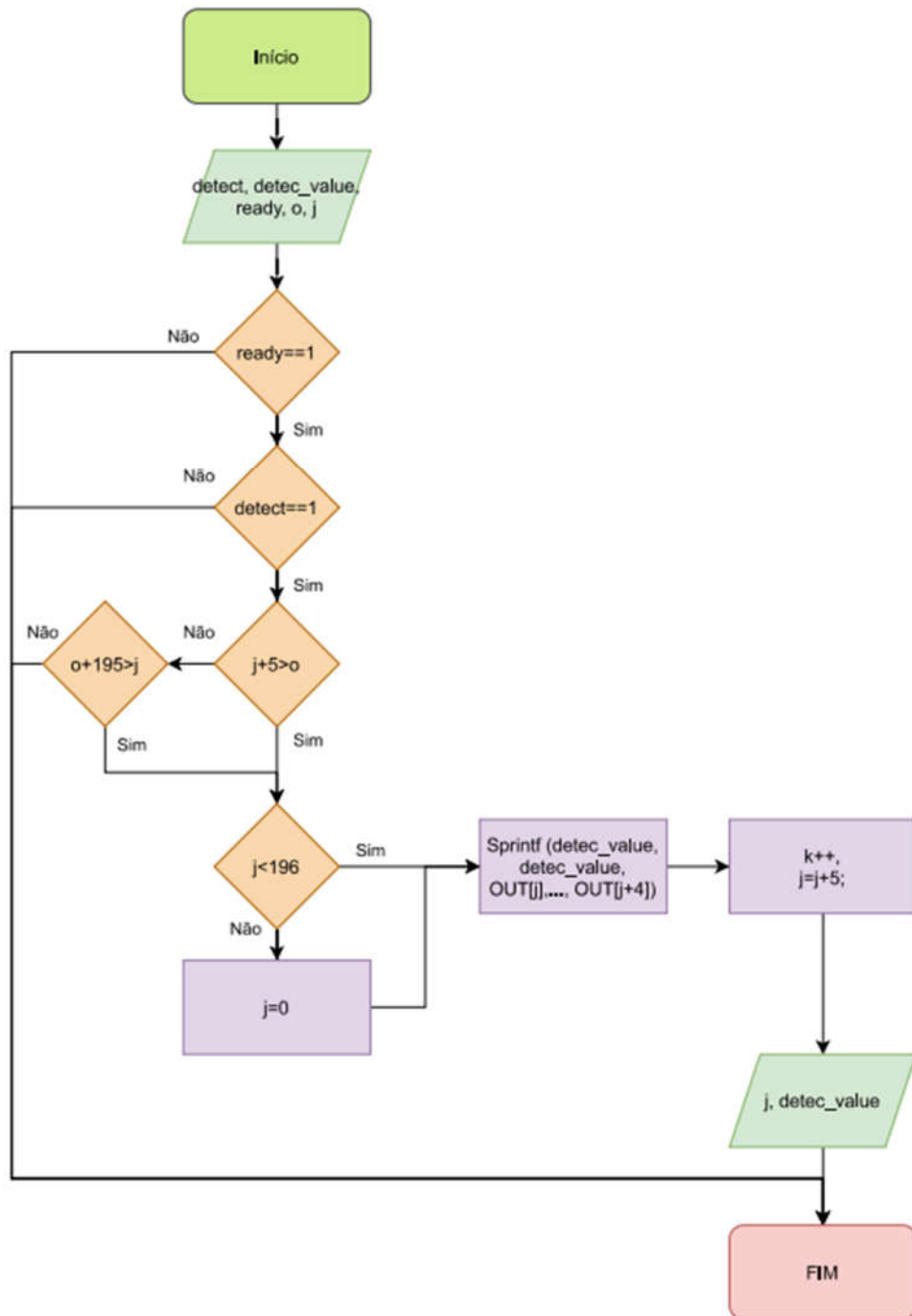
APÊNDICE K – ROTINA DE DETECÇÃO DE ARRITMIAS

Detecta Arritmias



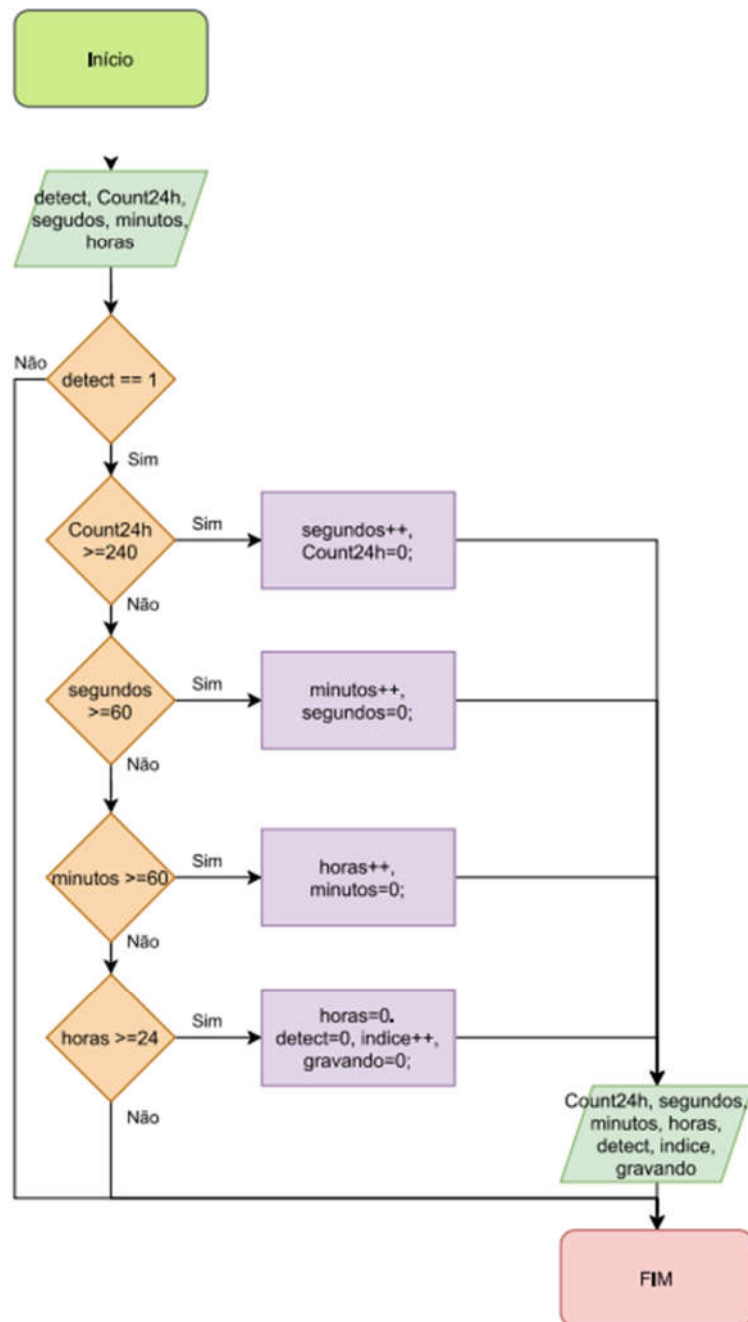
APÊNDICE L – ROTINA DE ARMAZENAMENTO DE DADOS EM STRING

Armazena String

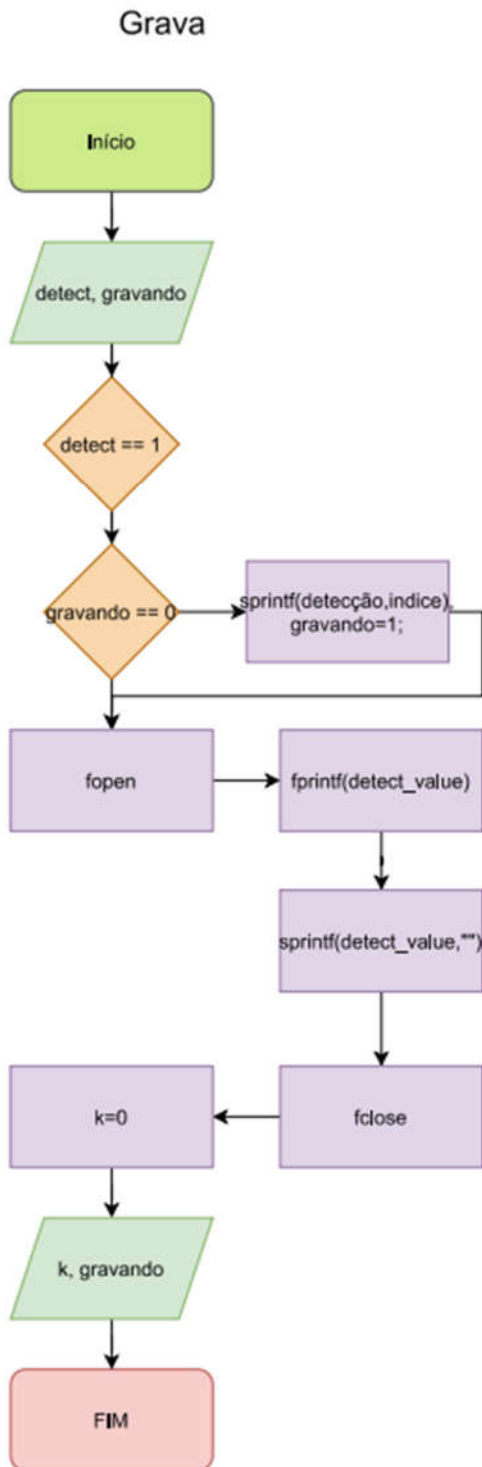


APÊNDICE M – ROTINA DE CONTAGEM DE 24 HORAS

Contagem 24h



APÊNDICE N – ROTINA DE GRAVAÇÃO EM MEMÓRIA EXTERNA



APÊNDICE O – CÓDIGO DO PROJETO.

```

1 #include "mbed.h"
2 #include "SDFileSystem.h"
3 #define N 200 // 200 é o nr de amostras do buffer circular
4 // colocar o valor que achar conveniente para N
5
6 AnalogIn analog_value(A1);
7
8 // Interrupções por Overflow do timer após 4,1666ms (240Hz)
9 Timeout Sampling; // Timer modo overflow, attach to Sampling.
10 //Pinos de gravação no SD
11 SDFileSystem sd(PA_7, PA_6, PA_5, PB_6, "sd");// mosi, miso, sck, cs
12 Serial async_port(PA_9, PA_10); // define transmitter and receiver
13
14 // Variáveis auxiliares
15 uint16_t buffer[N]; // Buffer para amostras lidas do ECG
16 uint8_t ns=0, status=0, acquire=0;
17 double entradalp[21], saidalp = 0;//entradas e saída do filtro passa baixas
18 double entradahp[21], saidahp = 0;//entradas e saída do filtro passa altas
19 double entradadf[5], saidadf;// entrada e saída do filtro derivativo
20 float sqr, Pcomp, Pamp=0, Tamp=0;// saída da função quadrática, comprimento da onda P, amplitude de pico da onda P, amplitude de pico da onda T
21 float mediamovel[31],mediamovelOUT, ant;//entradas da janela de média móvel, saída da janela de média móvel, saída anterior da janela de média móvel
22 int OUT[N], holder, counter=0, n=0, i=0, j=0,result, o=0, want=0, ready=0, detect=0, indice=0,k=0, sla=0, w=-15;// sinal filtrado, indica 0v
23 //n, i, j, o e k sevem de ponteiros para diferentes vetores e loops oant é o valor anterior de o ready indica que os filtros estão preenchidos
24 //detect indica detecção de arritmia, indice serve para diferencia múltiplos arquivos de texto de diferentes detecções
25 int count24h=0, S24h=0, M24h=0, H24h=0, gravando=0, teste;// amostras, segundos, minutos, horas da contagem de 24 horas
26 // gravando indica de que o sinal está sendo gravado impedindo a criação de um novo txt, teste serve para exibir o valor de qualquer variável no teraterm
27 int flgQ=0, flgS=0, flgT=0, flgTamp=0, flgP=0, flgPamp=0, newR=0, newT=0;// Flags dos componentes do sinal, indicando início de sua detecção
28 // newR e newT indicam que um novo valor de pico R e de amplitude de pico T foram detectados, e com isso serão gravados no vetor de média R e T
29 int p, q, r, rvet[10], rmed, s, t, tvet[10], tmed, cst, csr;// p, q, r, s, t são componetes do sinal rvet é o vetor de picos R
30 // rmed é o valor médio dos 10 ultimos picos R, tvet e tmed são a mesma coisa para a amplitude máxima da onda t, cst e csr contador do strng t e r
31 char detec_type[30], detec_value[2000];// nome do arquivo txt, strig com as amostras que será gravado no txt
32
33 //ISR associada ao timeout do TMR0 a taxa de 240Hz
34 void Aquisition(){
35     Sampling.detach();// Para contagem do Tmr0
36     buffer[ns++]=analog_value.read_u16(>>4); // Lê canal do ADC e guarda em buffer[ns].
37     if(ns==N){ // Se adquiriu N amostras indica como status=1.
38         ns=0;
39         status=1;
40     } // Buffer pronto para processamento
41     Sampling.attach(&Aquisition,0.004166); // Reinicia TMR0 para 240Hz.
42     count24h++;// incrementa uma amostra no contador 24h
43 }
44 // Inicializações do HW
45 void Init_HW(){
46     async_port.baud(9600);// define baud rate
47     ns=0; // Contador de amostras lidas do A/D index de buffer[N].
48     status=0; // Indicador do estado de ns compressão em curso ou não.
49     // Overflow TMR0 p/ 240Hz (Amostragem=240Hz/canal). Dispara contagem do TIMER0
50     Sampling.attach(&Aquisition,0.004166); // Timeout a taxa de 240Hz (T=4,16ms)
51     acquire=1;
52 }
53
54 void passa_baixas(){//função do filtro passa baixas
55     saidalp= ( 0.0039366*entradalp[20] +0.0059081*entradalp[19] +0.0110360*entradalp[18]
56             +0.0197861*entradalp[17] +0.0319797*entradalp[16] +0.0467544*entradalp[15]
57             +0.0626586*entradalp[14] +0.0778632*entradalp[13] +0.0904585*entradalp[12]
58             +0.0987770*entradalp[11] +0.1016838*entradalp[10] +0.0987770*entradalp[9]
59             +0.0904585*entradalp[8] +0.0778632*entradalp[7] +0.0626586*entradalp[6]
60             +0.0467544*entradalp[5] +0.0319797*entradalp[4] +0.0197861*entradalp[3]
61             +0.0110360*entradalp[2] +0.0059081*entradalp[1] +0.0039366*entradalp[0]);
62 }
63
64 void passa_altas(){//função do filtro passa altas
65     saidahp= (-0.0015003*entradahp[20] -0.0019474*entradahp[19] -0.0032253*entradahp[18]
66             -0.0052332*entradahp[17] -0.0077896*entradahp[16] -0.0106506*entradahp[15]
67             -0.0135350*entradahp[14] -0.0161543*entradahp[13] -0.0182432*entradahp[12]
68             -0.0195889*entradahp[11] +0.9814382*entradahp[10] -0.0195889*entradahp[9]
69             -0.0182432*entradahp[8] -0.0161543*entradahp[7] -0.0135350*entradahp[6]
70             -0.0106506*entradahp[5] -0.0077896*entradahp[4] -0.0052332*entradahp[3]
71             -0.0032253*entradahp[2] -0.0019474*entradahp[1] -0.0015003*entradahp[0]);
72 }

```



```

73
74 void derivativo(){//função do filtro derivativo
75     for (sla=0;sla<4;sla++){
76         entradadf[sla]=entradadf[sla+1];
77     }
78     entradadf[4]=saidahp;
79     saidadf= (0.2*entradadf[4] +0.1*entradadf[3] +0*entradadf[2] -0.1*entradadf[1] -0.2*entradadf[0]);
80 }
81
82 void quadratico(){//função quadrática
83     sqr= (saidadf*saidadf);
84     if (sqr>10000)
85         sqr=0;
86 }
87
88 void media_movel(){//função do janelamento de média móvel.
89     mediamovel[30]=sqr;
90     for (i=0;i<30;i++){
91         mediamovel[i]=mediamovel[i+1];//shift das entradas do filtro
92     }
93     ant = mediamovelOUT;// saída anterior do media movel recebe a atual
94     mediamovelOUT=(mediamovel[30]+mediamovel[29]+mediamovel[28]+mediamovel[27]+mediamovel[26]+mediamovel[25]+
95     mediamovel[24]+mediamovel[23]+mediamovel[22]+mediamovel[21]+mediamovel[20]+mediamovel[19]+mediamovel[18]+
96     mediamovel[17]+mediamovel[16]+mediamovel[15]+mediamovel[14]+mediamovel[13]+mediamovel[12]+mediamovel[11]+
97     mediamovel[10]+mediamovel[9]+mediamovel[8]+mediamovel[7]+mediamovel[6]+mediamovel[5]+mediamovel[4]+
98     mediamovel[3]+mediamovel[2]+mediamovel[1]+mediamovel[0])/31;
99 }
100
101 void pan_tompkins(){
102     for (i=0;i<20;i++){
103         entradalp[i]=entradalp[i+1]; //as ultimas 20 entradas do passa baixas são armazenadas
104         entradahp[i]=entradahp[i+1];//as ultimas 20 entradas do passa altas são armazenadas
105     }
106     entradalp[20]=buffer[o];
107     passa_baixas();
108     entradahp[20]=saidalp;
109     passa_altas();
110     OUT[o]=saidahp;// armazena o sinal filtrado no output
111     derivativo();
112     quadratico();
113     media_movel();
114 }
115
116 void detecta_Q(){// DETECÇÃO Q
117     if(mediamovelOUT >100 and mediamovelOUT> ant and flgQ==1){
118         q= OUT[w];
119         flgQ=0;
120         flgP=0;
121     }else{
122         q=0;
123     }
124 }
125
126 void detecta_R(){// DETECÇÃO R
127     if(mediamovelOUT >290 and ant+3>mediamovelOUT and ant-3<mediamovelOUT and flgS==0 and OUT[w-1]>OUT[w]){
128         r= OUT[w];
129         newR=1;
130         if(flgs==1 or flgQ==1)//caso algumas das flags esteja levantada no momento da detecção de R
131             detect=1;// indica arritmia
132         flgS=1;// seta flags como deveriam estar
133         flgP=0;
134         flgQ=0;
135     }else
136         r=0;
137 }

```

```

138
139 void detecta_S(){// DETECÇÃO S
140     if(mediamovelOUT < 120 and mediamovelOUT< ant and flgS==1){
141         flgT=1;
142         flgTamp=1;
143         s= OUT[w];
144         holder = OUT[w]+25;//holder marca o valor 0 de tensão no batimento atual
145         flgS=0;
146         counter=0;
147     }else{
148         s=0;
149     }
150 }
151
152 void detecta_T(){//DETECÇÃO T
153     if(flgt==1){
154         if(OUT[w] > holder and OUT[w]>OUT[w-5]){//primeira metade da onda T
155             t= OUT[w];
156             counter++;
157         }else{
158             if (flgTamp==1){// Detecta o pico da onda T
159                 Tamp = OUT[w];
160                 newT=1;
161                 flgTamp=0;
162             }
163             if(OUT[w] > holder+20){//Segunda metade da onda T
164                 t= OUT[w];
165                 counter++;
166             }else{
167                 if( counter >30){
168                     flgT=0;
169                     flgP=1;
170                     flgPamp=1;
171                     counter = 0;//reseta o contador de duração da onda T
172                 }else{
173                     t= OUT[w];
174                     counter++;
175                 }
176             }
177         }
178     }else
179         t=0;
180 }
181
182 void detecta_P(){//DETECÇÃO P
183     if(flgp==1){
184         if (OUT[w] > holder){
185             if(OUT[w]>OUT[o-1]){//Deteccão da primeira metade da onda P
186                 p= OUT[w];
187                 counter++;
188             }else{
189                 if(flgpamp==1){// Detecta o pico da onda P
190                     flgpamp=0;
191                     Pamp=OUT[w];
192                 }
193                 if(OUT[w] < OUT[w-1]){//Deteccão da segunda metade da onda P
194                     p= OUT[w];
195                     counter++;
196                 }else{
197                     if (counter >5){
198                         flgp=0;
199                         flgQ=1;
200                         Pcomp = counter *4.1666;// duração da onda P
201                         counter = 0;
202                     }
203                 }
204             }
205         }else{
206             if (counter >5 or mediamovelOUT >30){

```

```

207         flgP=0;
208         flgQ=1;
209         counter = 0;
210     }
211 }
212 }else
213     p=0;
214 }
215
216 void detecta_componentes(){// DETECÇÃO DOS COMPONENTES DO SINAL
217     detecta_R();
218     detecta_S();
219     detecta_T();
220     detecta_P();
221     detecta_Q();
222 }
223
224 void media_R(){// Calcula a média da amplitude dos últimos 10 picos R
225     if(newR==1){// caso tenha um R novo
226         rvet[csr]=r;// armazena no vetor de Rs
227         newR=0;// O R passa a ser considerado antigo
228         if (csr==9)
229             csr=0;
230         else
231             csr++;
232         if(rvet[9]!=0)// caso o vetor esteja totalmente preenchido
233             rmed = (rvet[0]+ rvet[1]+ rvet[2]+ rvet[3]+ rvet[4]+ rvet[5]+ rvet[6]+ rvet[7]+ rvet[8]+ rvet[9])/10;// calcula a média do vetor de picos R
234     }
235 }
236
237
238 void media_T(){// Calcula a média da amplitude das últimas 10 ondas T
239     if(newT==1){// caso tenha um T novo
240         tvet[cst]=Tamp;// armazena no vetor de Ts o valor do pico da onda T
241         newT=0;// O T passa a ser considerado antigo
242         if (cst==9)
243             cst=0;
244         else
245             cst++;
246         if(tvet[9]!=0)// caso o vetor esteja totalmente preenchido
247             tmed = (tvet[0]+ tvet[1]+ tvet[2]+ tvet[3]+ tvet[4]+ tvet[5]+ tvet[6]+ tvet[7]+ tvet[8]+ tvet[9])/10;// calcula a média do vetor de picos de T
248     }
249 }
250
251 void detecta_arritmias(){
252     if (Pamp < holder)// P negativa
253         detect =1;// detecta Patologia.
254     if (50<Pcomp<110 and Pamp> holder+31)// comprimento normal e amplitude acima do limite
255         detect =1;// detecta Patologia.
256     if (Pcomp>110 and Pamp> holder+31)// comprimento e amplitude acima dos limites
257         detect =1;// detecta Patologia.
258     if (r>rmed*1.2 or r<rmed*0.8)
259         detect =1;// detecta suspeita de Patologia.
260     if (Tamp>tmed*1.2 or t<tmed*0.8)
261         detect =1;// detecta suspeita de Patologia.
262 }
263
264 void processa_buffer(){
265     //entradalp[20]=buffer[o];
266     if(entradalp[0]>0 and entradahp[0]>0)// se as entradas mais antigas dos filtros tiverem algum valor armazenado
267         ready=1;// pronto para iniciar as detecções
268     pan_tompkins();
269     detecta_componentes();
270     media_R();
271     media_T();
272     detecta_arritmias();
273     if(o+5>ns or ns+195<o)// casonão hajam pelo menos 5 amostras entre o buffer e o sinal filtrado
274         wait_ms(4); // Espera
275 }
276
277
278 void grava(){//função de gravação no SD
279     if (detect==1 and gravando==0){// arritmia detectada, e gravação NÃO iniciada
280         sprintf(detec_type, "/sd/Suspeita%d.txt",indice);// Cria um novo arquivo txt para armazenar o sinal
281         gravando=1;// aponta o início da gravação
282     }

```

```

282 }
283 if (sd.mount()==-1)// caso haja problemas no mount do cartão SD
284     async_port.printf ("Erro no Mount\n");// printa mensagem de erro
285 FILE *fp = fopen(detec_type, "a");// abre o arquivo criado no modo gravação, acrescentando novos valores, e não substituindo
286 if(fp == NULL)// caso não encontre o arquivo
287     async_port.printf("Could not open file for write\n");// exibe mensagem de erro
288 fprintf(fp, "%s", detec_value);// grava os valores armazenados no string detec_value no arquivo de texto
289 sprintf(detec_value, "");// apaga os valores armazenados no string detec_value
290 fclose(fp);// fecha o arquivo
291 sd.unmount();// desmonta o cartão SD
292 //async_port.printf("gravado");// exibe a mensagem gravado
293 k=0;//seta o numero de amostras na gravação para 0
294 }
295
296 void contagem_24h(){//CONTAGEM DAS 24 HORAS DE GRAVAÇÃO
297     if(detec==1){
298         if(count24h>=240){// CONTAGEM DOS SEGUNDOS
299             S24h++;// aumenta contador de segundos
300             count24h=0;// zera contador de amostras
301         }
302         if (S24h>=60){// CONTAGEM DOS MINUTOS
303             M24h++;// aumenta contador de minutos
304             S24h=0;// zera contador de segundos
305         }
306         if(M24h>=60){// CONTAGEM DAS HORAS
307             H24h++;// aumenta contador de horas
308             M24h=0;// zera xontador de minutos
309         }
310         if(H24h>=24){// 24 H COMPLETAS
311             H24h=0;// zera contador de horas
312             detec=0;// FIM DA GRAVAÇÃO
313             indice++;//e aumentando em um o índice, para assim a próxima detecção não sobrescrever a primeira.
314             gravando=0;
315         }
316     }
317 }
318
319 void armazena_string(){
320     if (ready==1 and j<5 or w<want+5 or (w<want and w>5)){// se houverem 5 saídas para serem gravadas
321         want=w;// posição anterior do vetor recebe posição atual
322         if (j<195){// caso a saída esteja entre OUT[0] e OUT[195]
323             if (detec==1){// se houver detecção de arritmia
324                 sprintf(detec_value, " %c %d, %d, %d, %d, %d,", detec_value, OUT[j], OUT[j+1], OUT[j+2], OUT[j+3], OUT[j+4]);// armazena 5 saídas no string c
325                 k=k+1;// +1 numero de gravações no string
326                 j=j+5;//avança para as próximas 5 saídas
327                 n=n+5;//contagem do limitador de amostras
328             }
329         }else{// zera j quando ele atinge 200, assim começando a gravar a partir de OUT[0]
330             j=0;
331             if (detec==1){
332                 sprintf(detec_value, " %c %d, %d, %d, %d, %d,", detec_value, OUT[j], OUT[j+1], OUT[j+2], OUT[j+3], OUT[j+4]);
333                 k=k+1;
334                 n=n+5;
335                 j=j+5;
336             }
337         }
338     }
339 }
340
341 int main(){
342     Init_HW();
343     memset(entradahp, 0, sizeof(entradahp));//zera o vetor de entradas do filtro passa altas
344     memset(entradalp, 0, sizeof(entradalp));//zera o vetor de entradas do filtro passa baixas
345     memset(mediamovel, 0, sizeof(mediamovel));//zera o vetor de entradas da janela de média movel
346     memset(entradadf, 0, sizeof(entradadf));//zera o vetor de entradas do filtro derivativo
347     for(;acquire;){ // Inicia a aquisição de dados de ECG.
348         //if (n< 800){//limitador de amostras 676
349             if(status){
350                 processa_buffer();
351                 if(ready==1 and OUT[w]>0){// caso as entradas dos filtros estejam preenchidas
352                     armazena_string();
353                     //teste=OUT[0];// teste é a variavel que será printada no teraterm para tentar descobrir problemas no código
354                     //async_port.printf("%d, ",teste);
355                     if (o==199)//rotação
356                         o=0;
357                     else
358                         o++;
359                     if (w==199)//rotação
360                         w=0;

```

```
361         else
362             w++;
363         n++;
364         if(detect==1){// se houver detecção de arritmia
365             contagem_24h();// inicia a contagem de 24 horas
366             if(k>=100){// se houverem 100 gravações, ou seja 500 amostras no string
367                 async_port.printf("%c", detec_value);// printa no terra term o string com as saídas armazenadas
368                 grava();// grava o string no SD
369             }
370         }
371     }
372 }
373 //}
374 } // Após N amostras/canal processa os dados de ECG
375 Sampling.detach(); // Final da aquisição desabilita amostragem do ECG.
376 for(;;); // Espera
377 }
```