



TRABALHO DE GRADUAÇÃO

**DESENVOLVIMENTO DE INTERFACE USB E SPI PARA
ELETROCARDIOGRAFO DE ALTA RESOLUÇÃO**

**Carlos Augusto Nascimento dos Santos
Felipe Pradera Resende**

Brasília, setembro de 2009

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA**

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA

TRABALHO DE GRADUAÇÃO

**DESENVOLVIMENTO DE INTERFACE USB E SPI PARA
ELETROCARDIOGRAMA DE ALTA DEFINIÇÃO**

Carlos Augusto Nascimento dos Santos
Felipe Pradera Resende

Relatório submetido ao Departamento de Engenharia Elétrica como
requisito parcial para obtenção do grau de Engenheiro Eletricista

Banca Examinadora

Prof. Ricardo Zelenovski, ENE/UnB
Orientador

Eng. Marcelo Villegas
Examinador Externo

Guiler Alberto Cruz Silva, Mestre
ENE/UnB
Examinador interno

Dedicatória

Dedico este trabalho à minha família e meus amigos que me ajudaram a chegar até aqui me dando força para continuar.

Carlos Augusto Nascimento dos Santos

Ao grande amigo, fonte de muito animo nos momentos mais difíceis do curso e durante esse trabalho Carlos Augusto Nascimento dos Santos

Felipe Pradera Resende

Agradecimentos

Agradeço, em primeiro lugar, ao amigo e colega Giulier Alberto que foi o precursor e tornou possível a conclusão deste trabalho com bons resultados, ajudando com sua grande experiência em eletrocardiografia e programação.

Agradeço ao professor Ricardo Zelenovsky que mostrou a luz quando surgiram problemas aparentemente insolúveis.

Agradeço às minhas tias Vânia e Sirley pelo apoio, que literalmente entregaram as chaves de suas casas para que pudesse chegar a qualquer hora e me sentir em casa.

Agradeço à minha tia Sirlene pela oportunidade do meu primeiro emprego, que hoje vejo a importância que teve na minha permanência na Universidade e que me abriu as portas do serviço público.

Agradeço aos meus pais: Marcos, pelo exemplo, e Cinália, pelo carinho.

Agradeço a Deus por tudo.

Carlos Augusto Nascimento dos Santos

Aos meus pais, Astolpho e Consuelo, meus irmãos, Carolina, Denise, Gabriela e Victor e minha esposa Claudiane

Ao professor Ricardo Zelenovsky pela paciência, ao Dr. Giulier Alberto fundamental pelo termino bem sucedido desse trabalho.

A todos que torceram por mim nesses anos de graduação

Ao Bondoso Pai Celestial, pela saúde e capacidade de chegar até aqui.

Felipe Pradera Resende

RESUMO

Neste trabalho procurou-se elaborar a comunicação entre um sistema analógico de aquisição que recebe e trata o sinal cardíaco obtido na superfície da pele de uma pessoa e um computador pessoal de forma a obter um eletrocardiograma de alta definição. Como se desejava obter um eletrocardiograma de alta definição, foi utilizado um conversor análogo-digital com 24 bits de resolução para a quantização, modelo ADS1256 da *Texas Instruments*. Para a transmissão do sinal, agora digital, foi escolhida a interface *Universal Serial Bus* (USB) devido à disponibilidade desse tipo de interface na placa de desenvolvimento selecionada para controlar esta transmissão, modelo ATMEL SAM7-H256 da *Olimex*, e a compatibilidade com os diversos modelos de PCs disponíveis no mercado. Finalmente, para verificar a correta aquisição e transmissão da informação cardíaca, foi criado um software cuja função era plotar as 12 derivações de eletrocardiograma possíveis

ABSTRACT

DESIGN OF A HIGH RESOLUTION ECG WITH 12 LEADS AND USB INTERFACE USING ARM MICROCONTROLLER

Author: Carlos Augusto Nascimento dos Santos and Felipe Pradera Resende
Supervisor: Ricardo Zelenovsky
Graduate Program in Electrical Engineering
Brasília, September of 2009

In this work was made the communication between an analogical system that receives and process a cardiac signal obtained from the surface of the skin and a personal computer to obtain ECG. Were used an 24 bit A/D converter model ADS1256 Texas Instruments. The digital signal is transmitted by a USB. This work presents the design and development of data interface that allows connection with an ARM microcontroller to perform the tasks of control and communication to a computer via the USB interface. To verify fullness of the process of acquiring and transmission software was develop to plot the 12 ECG derivations.

SUMÁRIO

LISTA DE FIGURAS	IV
LISTA DE TABELAS.....	V
LISTA DE SÍMBOLOS	VI
CAPÍTULO 1.....	1
INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO.....	1
1.2 OBJETIVOS DO PROJETO	1
1.3 ORGANIZAÇÃO DO DOCUMENTO	2
CAPÍTULO 2.....	3
FUNDAMENTAÇÃO TEÓRICA.....	3
2.1 SINAL CARDÍACO	3
2.2 USB	7
2.3 SPI.....	9
CAPÍTULO 3.....	11
DESENVOLVIMENTO	11
1. MÓDULO DE AQUISIÇÃO DO SINAL DE ECG	11
2. MÓDULO DE CONVERSÃO ANALÓGICA-DIGITAL.....	14
3. MÓDULO USB DESENVOLVIDO NA PLACA ATMEL SAM7-H256.....	25
4. SOFTWARE DE PLOTAGEM DAS DERIVAÇÕES CARDÍACAS	27
.....	30
CAPÍTULO 4.....	31
CONSIDERAÇÕES FINAIS.....	31
3.1 CONCLUSÕES	31
3.2 SUGESTÕES DE MELHORIAS.....	32
REFERÊNCIAS BIBLIOGRÁFICAS.....	33

LISTA DE FIGURAS

Figura 2-1 – Triângulo de Einthoven, DI, DII e DIII.....	4
Figura 2-2 – Derivações Precordais.....	5
Figura 2-3 – Sinal típico de um ECG.....	6
Figura 2-4 – Vetocardiograma.....	6
Figura 2-1 – Diagrama da ligação entre dispositivos SPI	10
Figura 2-2 – Diagrama de transferência de bits via SPI.....	10
Figura 3-1 – Diagrama de blocos do circuito analógico completo.....	11
Figura 3-2 – Diagrama de blocos do módulo de retorno.....	12
Figura 3-3 – Diagrama de blocos do módulo canal.....	13
Figura 3-4 – Uma derivação de ECG após o tratamento analógico	13
Figura 3-5 – Derivação ruidosa e FFT do sinal	14
Figura 3-6 – Diagrama de blocos do ADC ADS1256.....	15
Figura 3-7 – Ligação dos pinos do ADC ADS1256.....	16
Figura 3-8 – Diagrama das entradas digitais e do MUX do ADC.....	18
Figura 3-9 – Comunicação entre ARM e ADC	19
Figura 3-10 – Freqüência do relógio serial (1,86 MHz).....	21
Figura 3-11 – Ciclo para troca das entradas do MUX do ADC	21
Figura 3-12 – Ciclo de comandos para conversão A/D.....	22
Figura 3-13 - Tempo entre comandos no barramento SPI (1,14 μ s)	23
Figura 3-14 – Tempo após comando SYNC (3,87 μ s).....	23
Figura 3-15 – Tempo após comando RDATA (6,95 μ s).....	24
Figura 3-16 – Freqüência de amostragem obtida no ADC.....	25
Figura 3-17 – Fluxograma da parte USB do <i>firmware</i>	26
Figura 3-18 – Tela do software com as 12 derivações eletrocardiográficas contendo interferências da rede elétrica	29
Figura 3-19 – Tela do software com as 12 derivações eletrocardiográficas livre de interferências da rede elétrica	30

LISTA DE TABELAS

Tabela 2-1 – Condutores que compõe um cabo USB	8
Tabela 3-1 – Descrição dos pinos do ADC ADS1256	15
Tabela 3-2 – Tabela de comandos utilizados pelo ADC	17
Tabela 3-3 – Características de temporização do ADC.....	20
Tabela 3-4 – Valores de tempo da comunicação SPI.....	24

LISTA DE SÍMBOLOS

Siglas

ADC	<i>Analog to Digital Converter</i> (conversor analógico-digital)
ARM	<i>Advanced RISC Machine</i>
CDC	<i>Communications Device Class</i> (classe de dispositivos de comunicação)
DC	<i>Direct Current</i> (corrente contínua)
ECG	Eletrocardiograma
ECGAR	Eletrocardiograma de Alta Resolução
MUX	<i>MUltipleXer</i> (multiplexador)
PC	<i>Personal Computer</i> (computador pessoal)
RAM	<i>Random Access Memory</i> (memória de acesso aleatório)
RISC	<i>Reduced Instruction Set Computer</i> (computador com instruções
reduzidas)	
SPI	<i>Serial Peripheral Interface Bus</i> (barramento serial para periféricos)
USB	<i>Universal Serial Bus</i>

Capítulo 1

Introdução

1.1 Motivação

Tendo em vista a escassez de eletrocardiógrafos de alta resolução, foi desenvolvida, como parte final de um programa de pós-graduação, uma placa para tratamento analógico de sinais de ECG preparada para receber um conversor análogo-digital com 24 bits, capaz de proporcionar a resolução desejada, e um microcontrolador de arquitetura ARM que possibilita a transmissão dos dados coletados a um computador pessoal.

Com o intuito de criar a parte digital, que completaria o projeto citado acima, fez-se necessário dominar os assuntos que cercam a conversão analógica-digital, a transmissão de dados entre o conversor e o microcontrolador e a transmissão USB realizada por este.

Motivado por este desafio, iniciou-se uma fase de estudos contemplando a comunicação entre os dispositivos envolvidos como: protocolo SPI, estrutura e modos de operação de dispositivos que possuem interface USB, além dos *datasheets* dos diversos componentes utilizados. Sendo seguida por projetos de *firmware* e *software*, sempre tendo em vista o melhor cumprimento dos requisitos impostos pelo projeto de pós-graduação que originou este trabalho.

1.2 Objetivos do Projeto

Esse trabalho de graduação tem o intuito de desenvolver o *firmware* para digitalização dos sinais analógicos das 8 derivações de um ECG em alta resolução e a transmissão desses dados para um microcomputador por meio de interface USB implementada sobre um microcontrolador de arquitetura ARM.

Ao final, com os dados disponíveis desse processo, possibilitar a plotagem e armazenamento dos sinais correspondentes aos 12 canais na tela do microcomputador para análise pela autoridade médica competente.

1.3 Organização do Documento

Este documento está dividido em 4 capítulos conforme descrito a seguir.

No capítulo 2 é descrita a fundamentação teórica necessária para a compreensão do trabalho e as tecnologias aplicadas, incluindo informações sobre o sinal cardíaco, ECG, interface USB e protocolo SPI.

No capítulo 3 é exibido o desenvolvimento seqüencial do projeto, passando pela parte de aquisição do sinal cardíaco, pelo tratamento desse sinal pela placa analógica, pela conversão análoga-digital, pelo envio dos dados ao microcontrolador e, finalmente, a entrega desses dados à porta USB do PC para plotagem das 12 derivações em *software* desenvolvido exclusivamente para esse fim.

No capítulo 4 são analisados os resultados obtidos e são feitas sugestões de melhorias para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

2.1 Sinal cardíaco

O Eletrocardiograma (ECG) é o registro das variações do potencial elétrico do músculo cardíaco em atividade.

As ondas de polarização e despolarização que se propagam ao longo das fibras cardíacas podem ser consideradas dipolos em movimento. Estes dipolos determinam campos elétricos variáveis que podem ser detectados pela medida da diferença de potencial através de eletrodos colocados na superfície cutânea.

Desta forma, os potenciais gerados pelo coração durante o ciclo sístole-diástole (contração/relaxamento) podem ser registrados aplicando-se eletrodos em diferentes posições do corpo. Na prática, existem locais padronizados onde os eletrodos de registro são colocados, de acordo com orientações pré-estabelecidas. Na realidade, o que se mede é a diferença de potencial elétrico entre dois pontos no campo elétrico gerado pelo dipolo elétrico cardíaco ao longo do ciclo cardíaco. Os pontos de medida são escolhidos e padronizados, originando as várias derivações.

Normalmente são colocados 5 eletrodos à superfície corporal: um em cada punho, um em cada tornozelo e um móvel que pode ser colocado na superfície torácica sucessivamente em seis posições diferentes. Por convenção, o eletrodo do punho direito recebe o nome de R (right), o punho esquerdo de L (left) e o do tornozelo esquerdo de F (foot). O eletrodo do tornozelo direito é ligado ao fio terra.

Estes eletrodos podem ser ligados entre si de 15 maneiras diferentes. Todavia, somente 12 são utilizadas na prática médica. Cada uma destas ligações é conhecida como uma derivação do eletrocardiograma.

Einthoven imaginou o coração no centro de um triângulo equilátero cujos vértices estariam representados pelo braço direito (R), braço esquerdo (L), e perna esquerda (F). A Figura 2.1 mostra esquematicamente os três eletrodos e as derivações bipolares no triângulo de Einthoven. Essa orientação foi baseada na Segunda Lei de Kirchoff que diz que num circuito fechado, a soma das diferenças de potencial é igual a zero. Neste triângulo, Einthoven inverteu a polaridade de DII a fim de obter registro positivo da onda R nas três derivações.

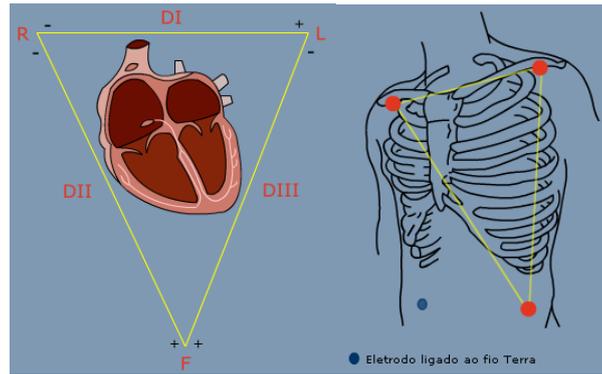


Figura 2.1 – Triângulo de Einthoven, DI, DII e DIII

As ligações feitas são:

- DI = VL-VR (braço esquerdo - braço direito)
- DII = VF-VR (perna esquerda - braço direito)
- DIII = VF-VL (perna esquerda - braço esquerdo)

$$DI + (-DII) + DIII = 0$$

Logo:

$$Vl - Vr - (Vf - Vr) + Vf - Vl = 0$$

$$Vl - Vr - Vf + Vr + Vf - Vl = 0$$

Os potenciais Vl , Vr e Vf gerados nos vértices do triângulo (de Einthoven) são obtidos pelas equações:

$$V_r = \frac{K \times M \times \cos(120^\circ - \alpha)}{d_3^2}$$

$$V_f = \frac{K \times M \times \cos(120^\circ - \alpha)}{d_2^2}$$

$$V_l = \frac{K \times M \times \cos(\alpha)}{d_1^2}$$

Sabemos que $d_1 = d_2 = d_3 = d$, portanto trigonometricamente verificamos que:

$$Vl + Vr + Vf = 0$$

Wilson introduziu o chamado terminal central (T) cujo potencial é considerado zero. O potencial de cada ponto V_I, V_r e V_f, é medido com referência a esse terminal. O terminal de Wilson é obtido unindo-se os vértices do triângulo de Einthoven a um terminal central através de resistências.

Este sistema com o terminal central fornece registros de potenciais com amplitudes muito baixas, razão pela qual foi substituído na prática eletrocardiográfica pelas derivações aumentadas de Goldberger.

Essas derivações de Goldberger, também chamadas de derivações precordiais, são obtidas unindo-se o terminal de Wilson (T) onde o eletrodo negativo é colocado. O eletrodo explorador, positivo, é colocado sucessivamente sobre as seis posições da superfície torácica como ilustrado na Figura 2.2, a saber:

- a) Quarto espaço intercostal, á direita do esterno (V₁)
- b) Quarto espaço intercostal, à esquerda do esterno (V₂)
- c) A meio caminho entre os pontos V₂ e V₄ (V₃)
- d) Quinto espaço intercostal esquerdo, na linha clavicular média (V₄)
- e) Quinto espaço intercostal esquerdo, na linha axilar anterior (V₅)
- f) Quinto espaço intercostal esquerdo, na linha axilar média (V₆)

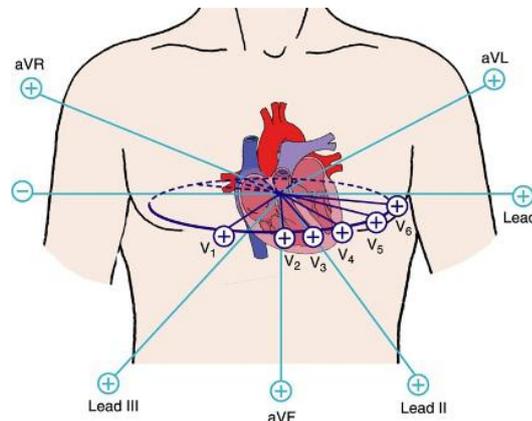


Figura 2.2 – Derivações Precordiais

O primeiro grupo de células a se despolarizarem são as células do nodo sinusal que são auto excitáveis. A onda de atividade se propaga e temos a despolarização dos dois átrios. A despolarização atrial produz um vetor dirigido predominantemente para frente e para a esquerda, determinando a onda P no registro eletrocardiográfico, Figura 2.3.

Em seguida, a onda é transmitida ao nódulo atrioventricular, ocorrendo logo após a despolarização do septo interventricular, da esquerda para a direita (onda Q). A onda de despolarização atinge, em seguida, as paredes do ventrículo (onda R) Em vista da predominância da massa ventricular esquerda, o vetor resultante volta-se para a esquerda, para baixo e para trás. Por último, tem-se a despolarização da região alta posterior do septo interventricular e das paredes ventriculares, parte que não recebe ramificação da rede de Purkinje. O vetor resultante gira para uma posição oblíqua dirigida para trás, para cima e para a direita (onda S).

A repolarização ventricular se processa numa ordem totalmente distinta, progredindo do epicárdio para o endocárdio. Isto se deve a serem mais curtos os potenciais de ação das células próximas ao epicárdio, resultando daí serem estas as primeiras a se respolarizarem (onda T). A direção média predominante durante a repolarização é do ápice para a base do coração, isto é, aproximadamente oposta à direção da despolarização ventricular (onda R).

A repolarização do átrio é encoberta pela despolarização do ventrículo, não sendo registrada no ECG. A Figura 2.4 ao lado ilustra a direção dos vetores de despolarização e repolarização no coração e o vetocardiograma de um homem normal. Esta ilustração nos permite ter uma noção temporal dos processos que ocorrem durante a despolarização e repolarização do coração.

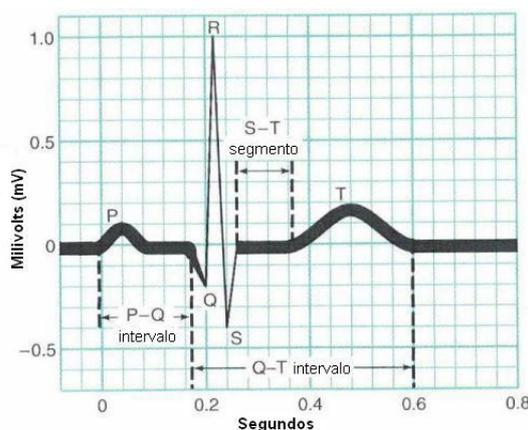


Figura 2.3 – Sinal típico de um ECG

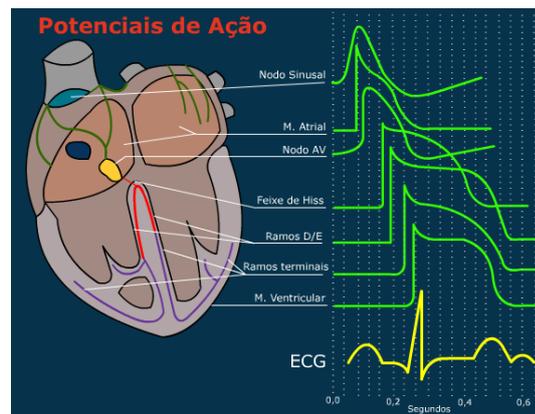


Figura 2.4 – Vetocardiograma

2.2 USB

O barramento Universal Serial Bus (USB) se tornou um padrão na microinformática. O seu desenvolvimento partiu da colaboração de diversas empresas líderes de mercado nos segmentos de software e hardware. Sua popularidade se deve à grande facilidade do uso não havendo normalmente a necessidade de configurações no hardware para que o mesmo funcione, basta conectá-lo a um host e o mesmo se encarrega de reconhecer a conexão elétrica do dispositivo.

Alem disso o dispositivo permite alta velocidade de comunicação sem comprometer a confiabilidade, os custos dos componentes são baixos e o sistema tem suporte em todos os sistemas operacionais.

O padrão se mostra bastante fácil para os usuários finais dos produtos que o utilizam para comunicação. Essa única interface permite, com diferentes tipos de conectores também padronizados, a utilização de até 127 dispositivos diferentes através da ligação de hubs. O processo automático de configuração permite que ao conectar o dispositivo ou periférico no barramento o hub detecta o dispositivo e faz uma ponte com o host que comunica ao sistema operacional e este já providencia a configuração, com isso não há a necessidade de reiniciar o sistema antes de usar o dispositivo. Como cada dispositivo compartilha o barramento, os recursos do sistema ficam livres para outros periféricos que não possuam essa facilidade.

O USB suporta barramentos com três velocidades de acordo com a versão da especificação. A versão 1.0 (low-speed) atinge a taxa de 1,5Mbps. A versão 1.1 (full-speed) atinge a taxa de 12Mbps. Por fim, a versão 2.0 (high-speed) atinge a taxa de 480Mbps. Todo microcomputador compatível com o USB opera por padrão nas versões 1.0 e 1.1. A versão 2.0 da especificação requer um hardware específico presente na placa-mãe do microcomputador ou em uma placa de expansão.

O USB foi projetado para operar com comunicação serial assíncrona, ou seja, sem o uso de um sinal de relógio de referência e com sinais de dados diferenciais. Para manter o sincronismo dos sinais entre transmissor e receptor é utilizada a codificação NRZI (Non-Return to Zero Inverted). O comprimento máximo do cabo de comunicação é de cinco metros. O cabo é blindado e composto por condutores conforme mostrado na tabela 2.1.

PINO	SINAL	COR
1	VBus(+5V)	Vermelho
2	D-	Branco
3	D+	Verde
4	GND	Preto

Tabela 2.1: Condutores que Compõe o USB

Cada dispositivo USB, pela especificação, deve implementar subdispositivos que respondam eletricamente às transações. Esses subdispositivos são chamados de Endpoints. Cada Endpoint recebe um endereço único que é composto por um número e pela direção dos dados, se envia ou recebe. Todo dispositivo deve ter um Endpoint de número zero que realiza transações de controle e, portanto, envia e recebe dados. Na verdade o Endpoint de número zero é composto por dois Endpoints, um para envio e outro para recepção, no mesmo endereço. São disponibilizados quatro bits para endereçamento dos Endpoints permitindo, assim, um total de dezesseis incluindo o de número zero.

A transferência de dados no padrão USB é realizada através de pacotes de dados. Os pacotes são do tipo token, data e handshake.

Os principais pacotes do tipo token são os pacotes setup que são utilizados para indicar que o próximo pacote data será para escrita no endpoint zero; pacotes in que são utilizados para indicar que o próximo pacote data deverá ser lido do endpoint de saída e; pacotes out que indicam que o próximo pacote data deverá ser escrito no endpoint de entrada.

Os pacotes do tipo data podem ser data1 e data0 e são efetivamente os dados transportados do host para o dispositivo ou do dispositivo para o host de acordo com o pacote token que os precederam.

Os pacotes do tipo handshake podem ser ack, nak e stall. O ack é utilizado para informar ao receptor, host ou dispositivo, que o dado foi recebido corretamente. O nak é utilizado para informar que o host ou dispositivo está ocupado ou inoperante no momento. Por fim o stall é utilizado para indicar um erro na comunicação.

A comunicação entre um dispositivo USB e um host é realizada mediante transações. A transação nada mais é que um conjunto de pacotes enviados e recebidos sem erros. Podem ser de quatro tipos: control, bulk, interrupt e isochronous. As transações do tipo control devem ser implementadas por todos os dispositivos USB e

têm a finalidade de permitir ao host o envio de solicitações de informações sobre o dispositivo e ler essas informações. As transações do tipo bulk podem ser implementadas apenas nos dispositivos que operam nas versões 1.1 (full-speed) e 2.0 (high-speed) e são utilizadas quando a taxa de transferência de dados não é crítica, isto é, o dispositivo ou o host podem aguardar a liberação do barramento quando este está ocupado. As transações do tipo interrupt podem ser implementadas por qualquer dispositivo USB e são utilizadas quando o dispositivo ou o host necessitam de atenção periodicamente como um mouse ou um teclado, por exemplo. As transações do tipo isochronous podem ser implementadas apenas em dispositivos que operam nas versões 1.1 (full-speed) e 2.0 (high-speed) e são utilizadas por dispositivos que necessitam da garantia do tempo de entrega dos dados mas que podem tolerar erros ocasionais uma vez que nesse tipo de transação não há a retransmissão de dados.

2.3 SPI

A *Serial Peripheral Interface* (interface periférica serial) é uma interface serial síncrona utilizada na comunicação entre dispositivos, mais comumente entre processador-dispositivo ou processador-processador respeitando a hierarquia mestre escravo.

A comunicação SPI consiste basicamente em dois registradores de deslocamento onde são transmitidos bit-a-bit os dados de um a outro dispositivo SPI. Durante a transferência de dados, um dos dispositivos atua como mestre, controlando o fluxo de dados, enquanto o outro atua como escravo recebendo ou transmitindo dados por meio de deslocamentos de bits. Existem basicamente dois protocolos distintos: no *Multiple Master Protocol* vários processadores podem revezar na condição de mestre, já no *Single Master Protocol* há um único mestre com vários escravos. Porém, enquanto um mestre pode enviar dados a vários periféricos simultaneamente, somente um escravo por vez pode remeter dados ao mestre em um determinado instante.

Como utilizaremos neste projeto apenas o modo mais simples de comunicação SPI, com um mestre e um escravo, esse foi o foco dado nessa parte introdutória.

Na comunicação SPI há duas linhas de dados e duas de controle:

- *Master Out Slave In* (MOSI): Fornece o dado da saída do mestre que será deslocado para dentro do escravo

- *Master In Slave Out (MISO)*: Fornece os dados que saem do escravo para o mestre. Só um escravo poderá transmitir por vez.
- *Serial Clock (SPCK)*: Linha de controle que parte do mestre e regula o fluxo de bits, definindo uma das varias possíveis taxas de transmissão. Para cada bit transmitido há um ciclo do SPCK
- *Slave Select (NSS)*: Habilita ou desabilita escravos.

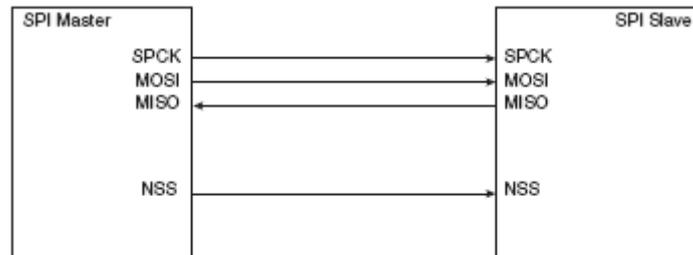


Figura 2-1 – Diagrama da ligação entre dispositivos SPI

Após apresentar o diagrama de ligação é apresentado abaixo uma figura que ilustra a transferência de bits:

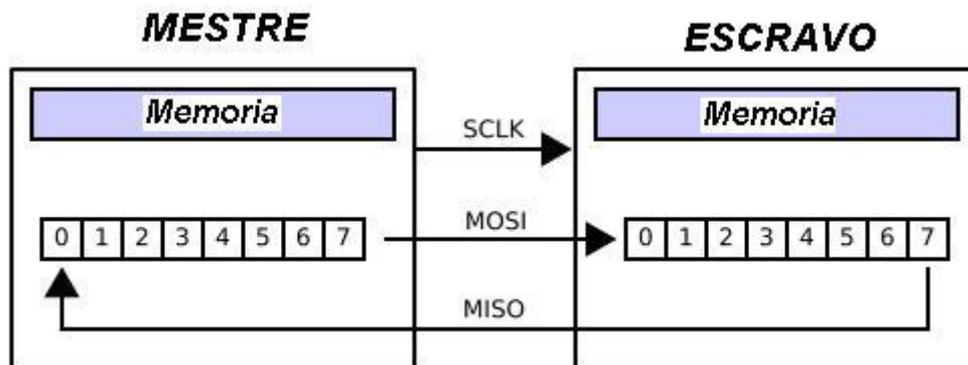


Figura 2-2 – Diagrama de transferência de bits via SPI

Duas características podem ser observadas no diagrama de transferência, uma delas é a ligação em anel feita entre os pinos MOSI e MISO e a segunda é que se faz necessário enviar bits para receber bits do outro dispositivo.

Capítulo 3

Desenvolvimento

1. Módulo de aquisição do sinal de ECG

Como dito anteriormente, o foco deste trabalho é a parte digital do eletrocardiógrafo desenvolvido, no entanto, não se pode deixar de mencionar o tratamento analógico, uma vez que decisões foram tomadas baseadas nas características delimitadas pela parte analógica do circuito. Dito isso, inicia-se uma descrição básica dos elementos que compõem tal seção.

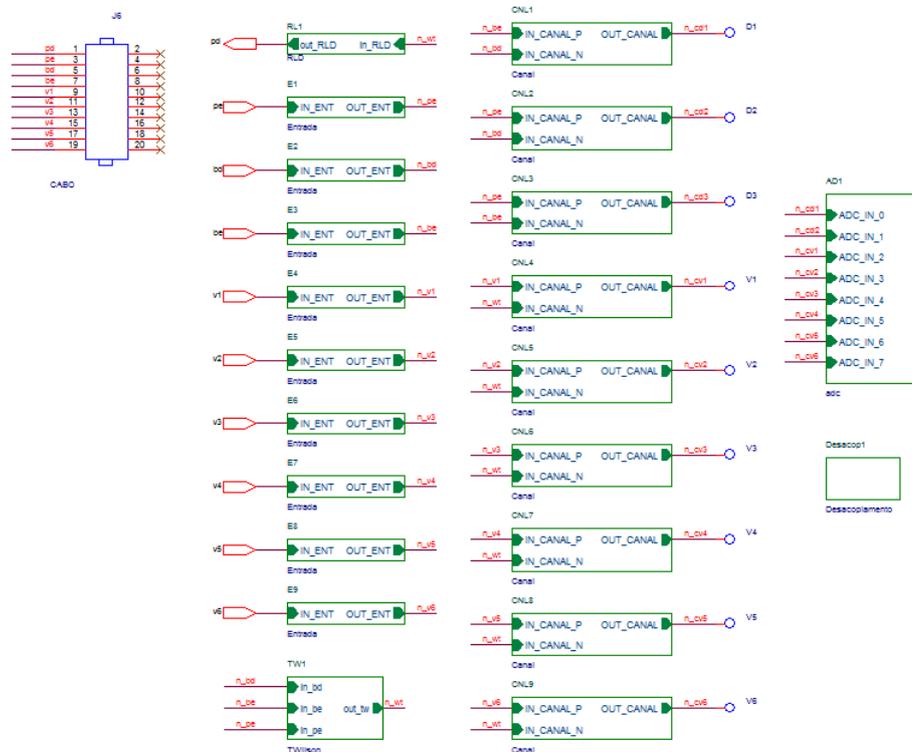


Figura 3-1 – Diagrama de blocos do circuito analógico completo

O circuito tem como entrada 10 eletrodos que estarão em contato com a pele do paciente. Destes, nove (oriundos dos pontos preconizados dos estudo de Einthoven, conforme descrição apresentada na figura XX) são conectados aos módulos de *buffer* que possuem ganho unitário e cuja função é manter o nível de potência dos sinais de entrada.

O eletrodo não mencionado acima funciona como retorno e é descrito nos próximos dois parágrafos.

Dentre as nove saídas dos módulos de *buffer* descritos acima, as três que são usadas para obter as derivações bipolares periféricas (DI, DII e DIII) são direcionadas ao chamado terminal de Wilson, que realiza a soma desses sinais e a transmite ao circuito driven right leg (RLD), Figura 3-2.

O circuito RLD, que recebeu a saída do terminal de Wilson, limita a corrente, dá um ganho de -39 V/V e aplica sua saída no paciente pelo décimo eletrodo. Esse conjunto de retorno tem a função de ser referência para as medições de sinais e proporcionar a estabilização dos mesmos, evitando que hajam flutuações que podem chegar a saturar os componentes amplificadores que serão apresentados. A figura abaixo é o diagrama de blocos do módulo de retorno descrito.

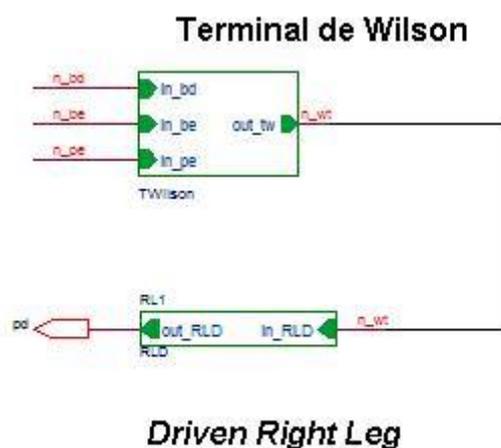


Figura 3-2 – Diagrama de blocos do módulo de retorno

Os 9 sinais de entrada, que já passaram pelo estágio de *buffer*, são então direcionados aos pares aos módulos denominados de canal. Os pares que entraram em cada canal são escolhidos de forma a definir cada uma das derivações eletrocardiográficas. Para a derivação DI, por exemplo, são devidamente colocados os sinais provenientes dos terminais do braço esquerdo e do braço direito.

As funções do módulo canal são quatro, conforme descrito abaixo:

- Instrumentação – Consiste em um amplificador diferencial de instrumentação que realiza a subtração dos sinais de entrada e aplica um ganho de $9,93$ V/V,
- Filtro Passa-Altas – Elimina as frequências inferiores a $0,05$ Hz (Basicamente o sinal DC),
- Ganho – Amplificador operacional não inversor que insere ganho 101 V/V, e

- Filtro Passa-Baixas – Elimina as frequências superiores a 200 Hz.

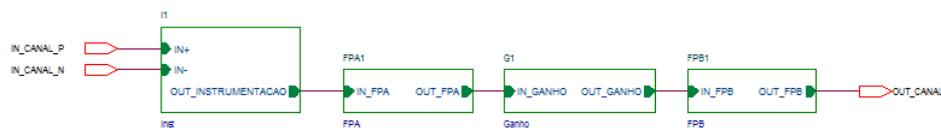


Figura 3-3 – Diagrama de blocos do módulo canal

Após estes processos, os sinais são disponibilizados nas entradas do ADC, onde se tem início o principal escopo deste trabalho, conversão analógica digital e transmissão ao PC por meio de porta USB.

A figura abaixo mostra uma das derivações bipolares periféricas obtida na tela de um osciloscópio digital ao final do tratamento analógico:

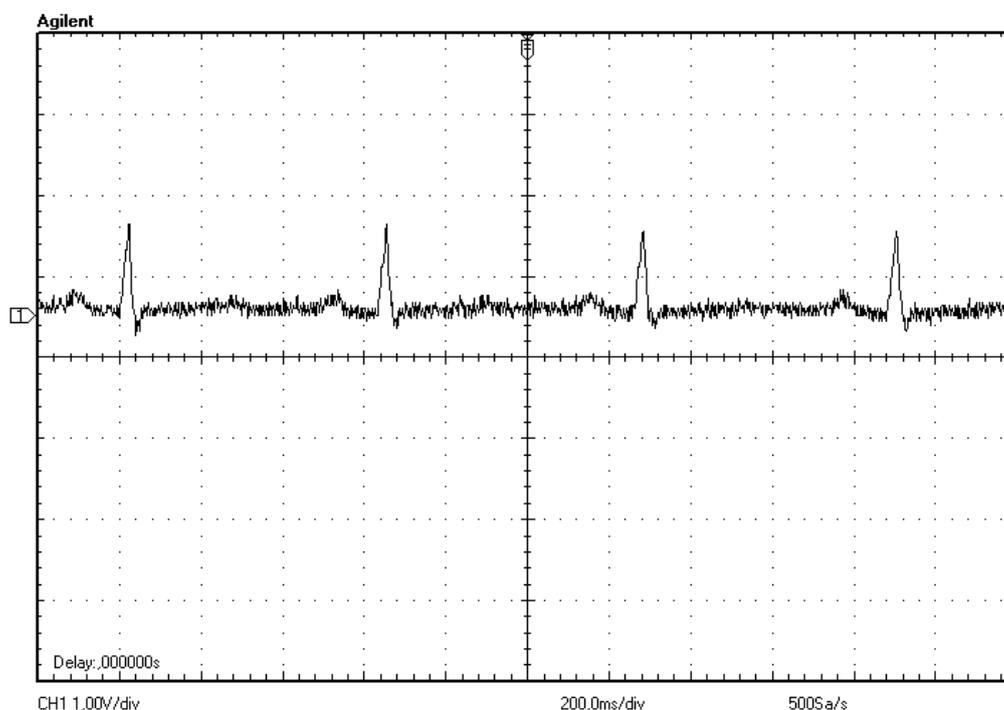


Figura 3-4 – Uma derivação de ECG após o tratamento analógico

Observamos um sinal bem definido onde podemos verificar as ondas de ECG, no entanto, a linha de base do sinal apresenta uma interferência que supomos ser proveniente da rede elétrica, frequência de 60 Hz e suas harmônicas.

Para confirmar essa suspeita, o osciloscópio foi programado de modo a apresentar a FFT do sinal, para que possamos verificar em quais faixas de frequência está distribuída a potência do sinal, e foi observado o sinal apresentado abaixo:

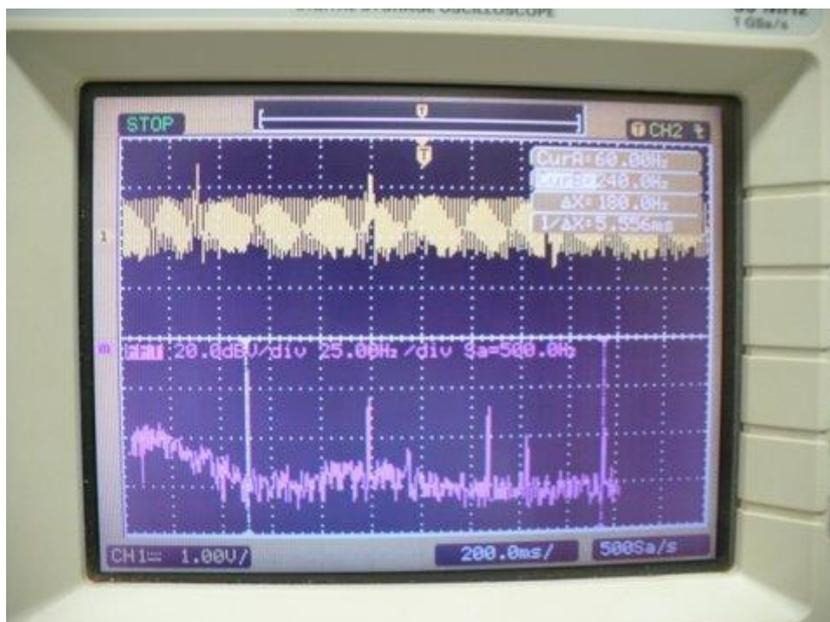


Figura 3-5 – Derivação ruidosa e FFT do sinal

Confirmando o que esperávamos, observamos grandes distribuições de potência nas frequências de 60 Hz, 120 Hz e 180 Hz. Problema que era previsto e cuja solução seria implementada por meio de filtros digitais no software responsável por plotar as derivações na tela do PC.

2. Módulo de conversão analógica-digital

Para a conversão analógica-digital, foi escolhido o ADC modelo ADS1256 do fabricante *Texas Instruments*. Tal escolha se deu pela quantidade de entradas disponibilizadas por esse dispositivo ser compatível com o número de sinais captados, por oferecer resolução de 24 bits, destacar a taxa de amostragem igual a 30 mil amostras por segundo com troca rápida dos sinais de entrada e ter comunicação compatível com o processador ARM utilizado.

Além das características citadas acima, o ADS1256 é um conversor do tipo sigma-delta de quarta ordem de muito baixo ruído, conta com um *buffer* de entrada, com um amplificador programável capaz de ganhos de 1 à 64 em passos de potência de 2 (1, 2, 4, 8, 16, 32 e 64) e um filtro digital programável.

A figura abaixo apresenta o diagrama de blocos do componente:

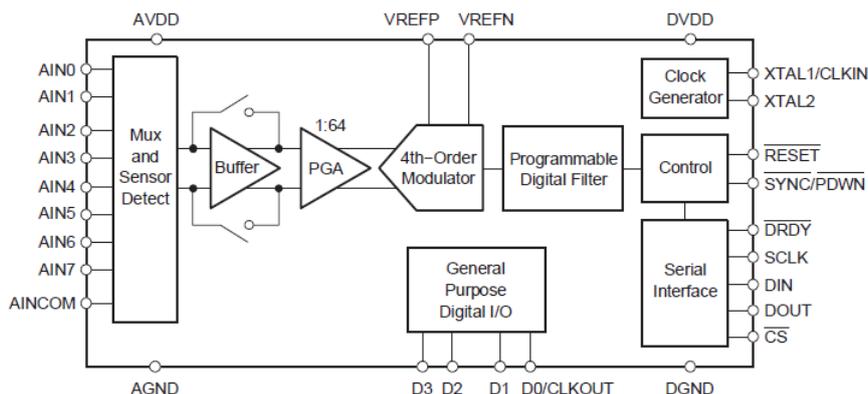


Figura 3-6 – Diagrama de blocos do ADC ADS1256

A tabela apresenta abaixo identifica os pinos exibidos no diagrama de blocos do ADC:

Tabela 3-1 – Descrição dos pinos do ADC ADS1256

Sinal	Descrição
AVDD	Fonte de alimentação analógica
AGND	Terra analógico
VREFN	Entrada de referência negativa
VREFP	Entrada de referência positiva
AINCOM	Entrada analógica comum
AIN0	Entrada analógica 0
AIN1	Entrada analógica 1
AIN2	Entrada analógica 2
AIN3	Entrada analógica 3
AIN4	Entrada analógica 4
AIN5	Entrada analógica 5
AIN6	Entrada analógica 6
AIN7	Entrada analógica 7
SYNC/PDWN	Entrada de sincronização/Desligar energia
RESET	Entrada de reinicialização
DVDD	Fonte de alimentação digital
DGND	Terra digital
XTAL2	Conexão ao cristal oscilador
XTAL1/CLKIN	Conexão ao cristal oscilador / entrada de relógio externo
CS	Sinal de componente selecionado

DRDY	Saída “dados prontos”
DOUT	Saída de dados seriais
DIN	Entrada de dados seriais
SCLK	Entrada do relógio serial
D0/CLKOUT	Entrada/Saída digital 0 / Saída de relógio serial
D1	Entrada/Saída digital 1
D2	Entrada/Saída digital 2
D3	Entrada/Saída digital 3

A ligação dos pinos do ADC é mostrada na figura a seguir, no entanto, o detalhamento dessa ligação ficará restrito à parte ligada à conversão dos sinais de entrada:

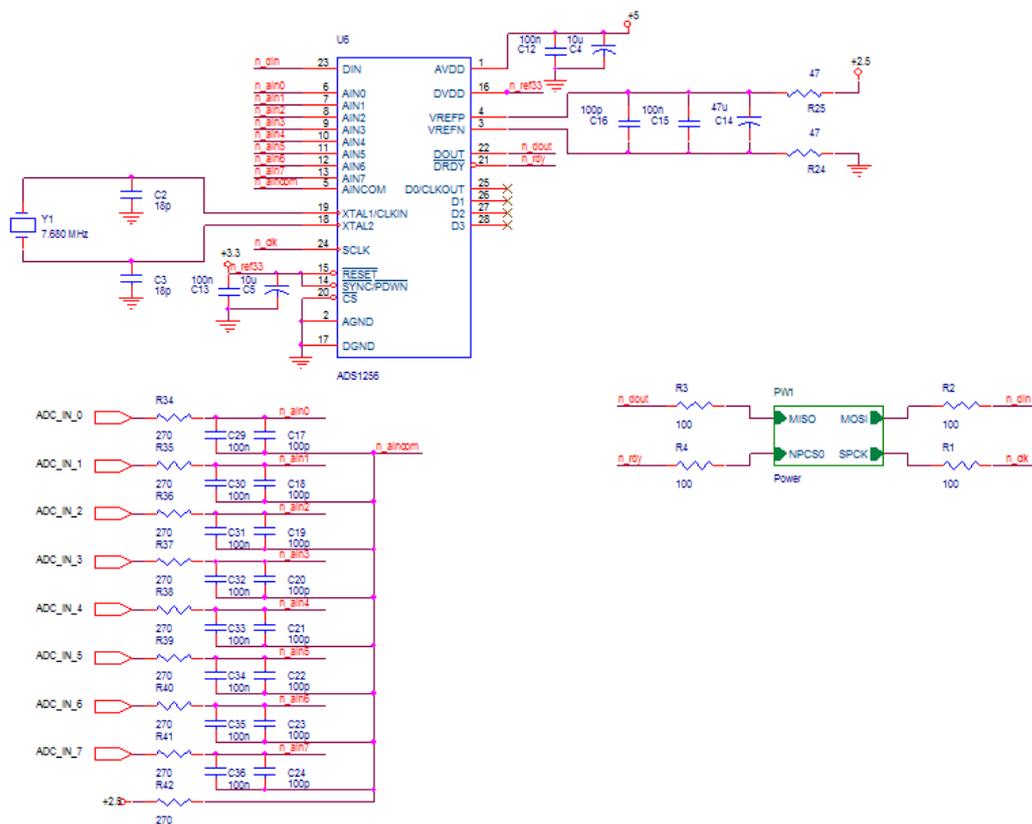


Figura 3-7 – Ligação dos pinos do ADC ADS1256

O dispositivo possui 11 registradores de 8 bits para configurações, leitura dos estados de conversão e calibração. Disponibiliza também 15 rotinas que possibilitam escrita e leitura dos registradores, autocalibração, sincronização, entre outros. Dos comandos, os que foram utilizados neste projeto seguem na tabela abaixo:

Tabela 3-2 – Tabela de comandos utilizados pelo ADC

Comando	Descrição
WAKEUP	Completa o comando SYNC ou sai do modo STANDBY
RDATA	Leitura de dados
RREG	Lê de um registrador
WREG	Escreva em um registrador
SELFCAL	Autocalibração de <i>offset</i> e ganho
SYNC	Sincroniza a conversão analógica para digital

O ADS1256 possui 9 entradas analógicas diferenciais ligadas à um MUX que, de acordo com a programação feita nos registradores, faz a seleção dos sinais que serão subtraídos e darão origem a um único sinal digital em sua saída. Como uma dessas entradas deve ser o canal de referência (AINCOM) serão utilizados nas demais entradas os sinais correspondentes às derivações DI, D3, V1, V2, V3, V4, V5 e V6, tendo em vista que DII pode ser obtido algebricamente de acordo com a Lei de Einthoven ($DII = DI + DIII$), não será limitação ao projeto:

A figura abaixo representa as entradas analógicas e o MUX de entrada do ADC:

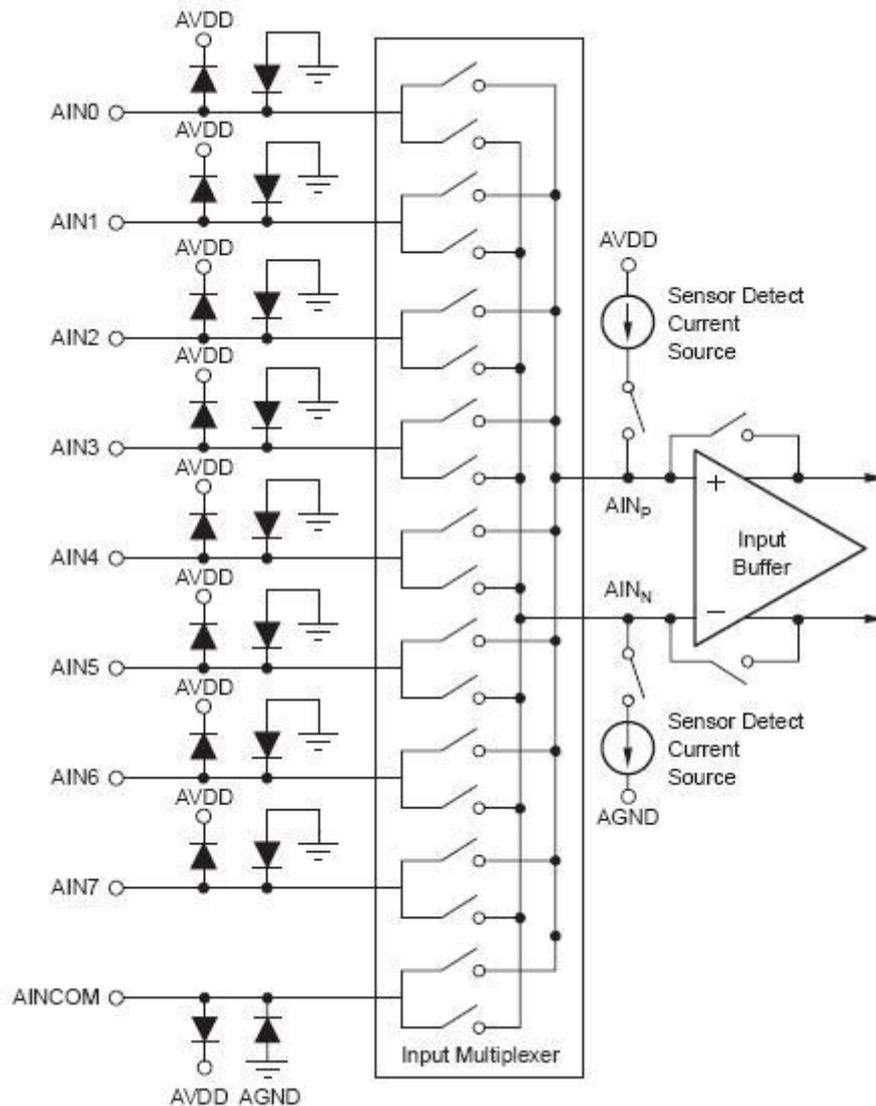


Figura 3-8 – Diagrama das entradas digitais e do MUX do ADC
(modificado) (Texas Instruments, 2008a)

A presença desse MUX foi positiva pois possibilitou que 9 sinais fossem aplicados a um só dispositivo, mas, em contrapartida, causou uma grande e inesperada queda na taxa de amostragem do sinal. Era esperada uma queda na taxa de amostragem devido à comutação dos sinais e o *datasheet* do dispositivo preconizava uma “troca rápida” das entradas. No entanto, após a montagem e configuração para mudança pelo MUX, foi constatada uma queda de 30k amostras/s (no caso de entradas fixas) para apenas 4.374 amostras/s (utilizando o multiplexador).

A taxa de amostragem desejada era de 1.000 amostras/s por derivação, mas, devido à limitação descrita acima, apenas 500 amostras/s por derivação foram obtidas, ficando demasiadamente próximo ao limite de amostragem que é de 400 amostras/s para o sinal de ECG limitado a 200 Hz, como descrito na parte analógica.

Após essas considerações, analisaremos os resultados obtidos na comunicação entre o ADC e o microcontrolador.

A comunicação entre o ARM e o ADC é feita a quatro fios e utiliza a interface SPI onde o microcontrolador é o mestre e o conversor é o escravo. O mestre é responsável por gerar o pulso de relógio serial (SPCK) e assim controlar o fluxo de dados que circulam no barramento. Pelo terminal MOSI, o ARM envia os comandos para sincronismo, configuração e leitura de dados ao pino de entrada digital DIN; pelo MISO, recebe os dados convertidos transmitidos pela saída digital DOUT; e pelo NPCS 0 faz varredura do pino DRDY, verificando se existem dados prontos para serem enviados.

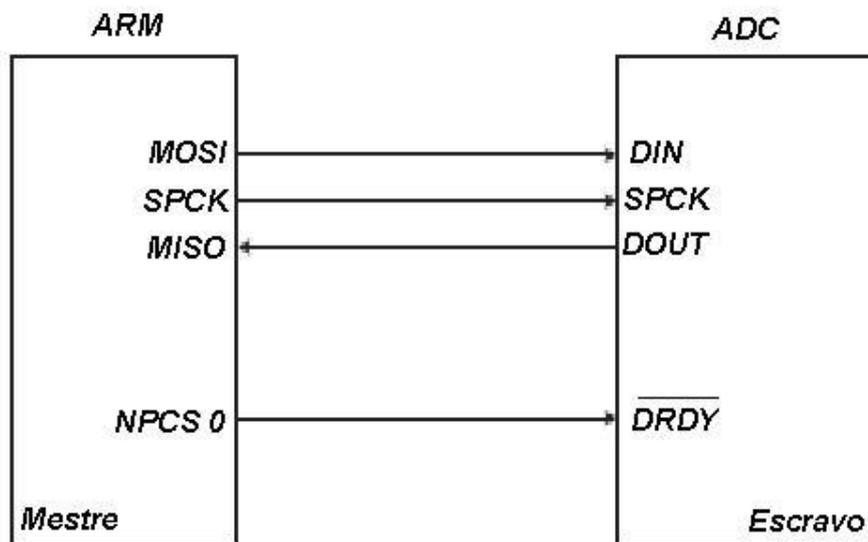


Figura 3-9 – Comunicação entre ARM e ADC

Basicamente o ARM irá controlar o ADC segundo os seguintes passos:

1. Envia os comandos de sincronismo e coloca o ADC em estado ativo;
2. Envia os comandos de escrita nos registradores de configuração, deixando o ADC em conformidade com os requisitos de projeto como por exemplo: determina a ordem da seqüência de bits (do mais significativo para o menos significativo), habilita a autocalibragem, habilita o *buffer* analógico de entrada, dá um ganho de 2 no amplificador programável e define a taxa de envio de dados igual a 30k amostras/s;
3. Envia a seqüência de comandos que determina quais sinais na entrada do MUX são selecionados e, que por sua vez, serão subtraídos um do outro e convertidos em dados digitais de 24 bits;

4. Detecta, pelo pino NPC0, se existem dados (da seleção feita no passo anterior) prontos para serem transmitidos; e
5. Envia o comando para leitura dos dados prontos.

Por se tratar de comunicação síncrona, ser necessária a interpretação de comandos e escrita de configurações nos registradores, a simplicidade conceitual do protocolo esbarra na dificuldade de implementação pelas rígidas determinações impostas pela temporização.

A tabela abaixo apresenta as principais limitações de tempo apresentadas por este protocolo:

Tabela 3-3 – Características de temporização do ADC

Característica	Valor
Relógio Principal – proveniente do oscilador de cristal (<i>master clock</i>)	- Frequência: 7,68MHz - Período: 130,21ns
Relógio Serial – usado para comunicação	- No mínimo 4 períodos do relógio principal - Frequência máxima: 1,91MHz - Período mínimo: 521ns
Tempo necessário entre duas transmissões no barramento SPI	- No mínimo 4 períodos do relógio principal - Período mínimo: 521ns
Tempo necessário para recuperação de dados após os comandos RDATA, RDATAc e RREG	- No mínimo 50 períodos do relógio principal - Duração mínima: 6,51µs
Tempo necessário entre comandos após os comandos RDATAc e SYNC	- No mínimo 24 períodos do relógio principal - Duração mínima: 3,125µs
Tempo necessário para estabilização após o comando WREG para a troca das entradas do MUX	- Em 30k Amostras/s: 210µs - Em 15k Amostras/s: 250µs - Em 7,5k Amostras/s: 310µs - Em 3,75k Amostras/s: 440µs

Como descrito na segunda linha da tabela acima, verificamos que a frequência máxima no barramento deve ser igual a $\frac{1}{4}$ da frequência do relógio principal (proveniente de um oscilador de cristal de 7,68 MHz) obtendo 1,92 MHz. Com o uso de

um osciloscópio digital foi verificada a frequência do relógio serial no barramento igual a 1,86 MHz, conforme apresentada na figura abaixo:

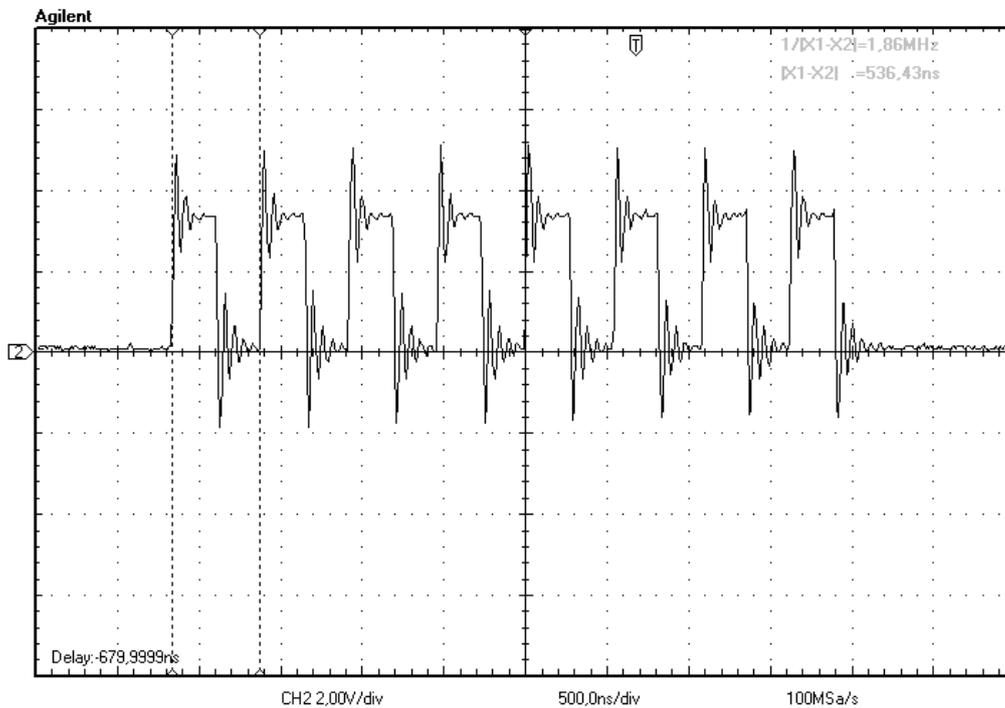


Figura 3-10 – Frequência do relógio serial (1,86 MHz)

O fabricante sugere o modo de operação mais eficiente para o ADC realizar a troca de entradas do MUX e a leitura dos dados digitalizados. A figura mostra que, quando o sinal que indica que o dado está digitalizado (DRDY) vai para nível baixo, deverá ser enviado o comando para trocar a entrada do MUX, os comandos SYNC e WAKEUP para iniciar a conversão com a nova configuração de entrada e, só então, é retirado o dado da configuração de entrada anterior, ou seja, enquanto um dado da porta seguinte é convertido, a leitura da porta inicial é realizada.

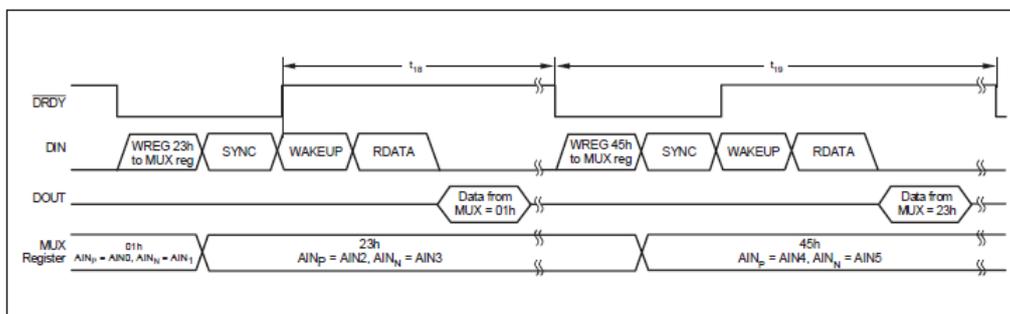


Figura 3-11 – Ciclo para troca das entradas do MUX do ADC
(Texas Instruments, 2008a)

Seguindo as características de temporização do *datasheet* do componente, foi calculado que: o tempo mínimo entre transmissões no barramento SPI é de 521ns, tempo mínimo após o comando SYNC é de 3,125 μ s e tempo mínimo após o comando RDATA e a realização das leituras é de 6,51 μ s.

Apresentamos nas próximas figuras os resultados obtidos no teste do ciclo completo da troca das entradas do MUX. Pode-se perceber que, após a leitura do dado convertido da configuração de canal anterior do MUX, conforme sugerido e mostrado na figura acima, com a temporização entre comandos SPI, a temporização após o comando SYNC e a temporização após o comando RDATA respectivamente.

Nas três figuras o sinal superior indica o sinal de dado pronto (DRDY) e o sinal inferior o relógio do barramento SPI (SPCK).

Os três primeiros conjuntos de ciclos de relógio são referentes ao comando WREG que seleciona as entradas do MUX do ADC. O quarto conjunto de ciclos de relógio se refere ao comando SYNC, o quinto ao comando WAKEUP, o sexto ao comando RDATA e os três seguintes à leitura dos três bytes resultantes da conversão, conforme apresentado na figura abaixo:

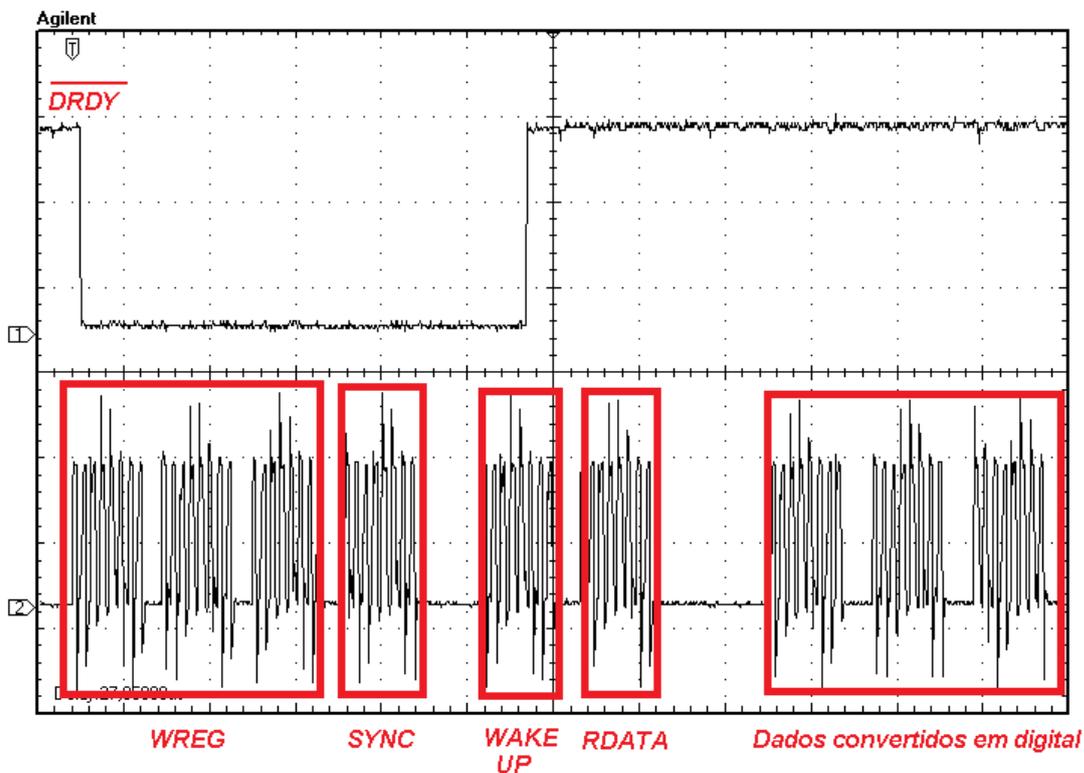


Figura 3-12 – Ciclo de comandos para conversão A/D

Os tempos implementados foram de $1,14\mu\text{s}$ entre comandos no barramento, $3,87\mu\text{s}$ após o comando SYNC e $6,95\mu\text{s}$ após o comando RDATA.

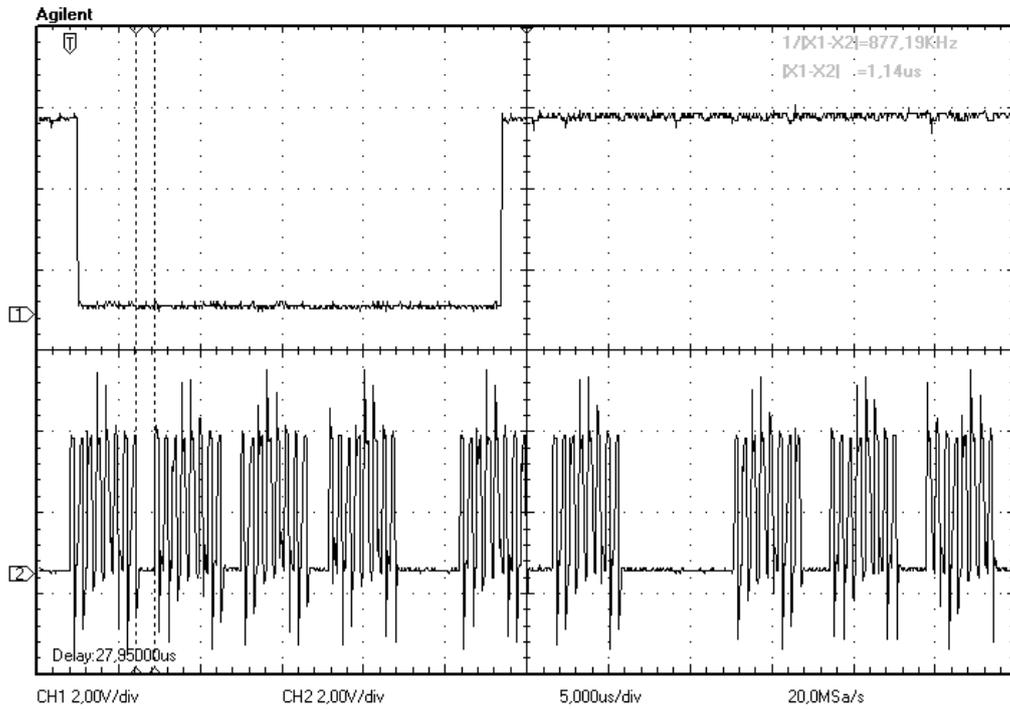


Figura 3-13 - Tempo entre comandos no barramento SPI ($1,14\mu\text{s}$)

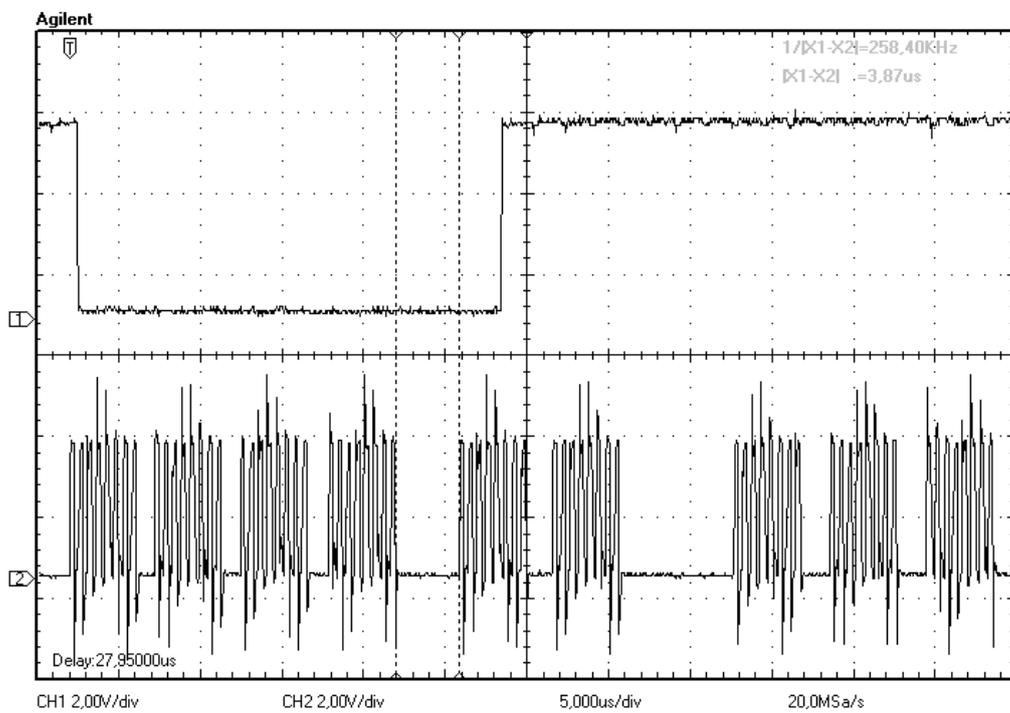


Figura 3-14 – Tempo após comando SYNC ($3,87\mu\text{s}$)

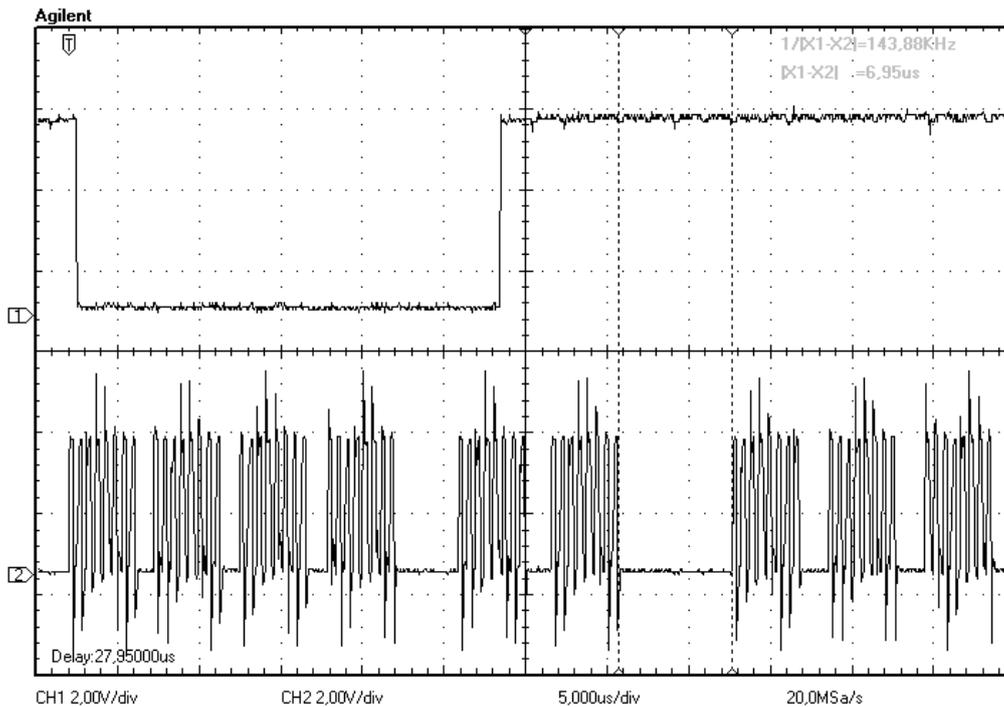


Figura 3-15 – Tempo após comando RDATA (6,95 μ s)

Comparando os valores mínimos com os obtidos na prática, foi gerada a seguinte tabela, que mostra o funcionamento dentro da temporização esperada:

Tabela 3-4 – Valores de tempo da comunicação SPI

TEMPO	MÍNIMO	OBTIDO
Entre comandos	524 ns	1,14 μ s
Após comando SYNC	3.125 μ s	3,87 μ s
Após comando RDATA	6,51 μ s	6,95 μ s

Como citado anteriormente, a frequência de amostragem obtida foi de 500 amostras por derivação, totalizando 4k amostras/s nos 8 canais, a próxima figura apresenta a tela do osciloscópio que exibe a taxa de 4 k amostras por segundo.

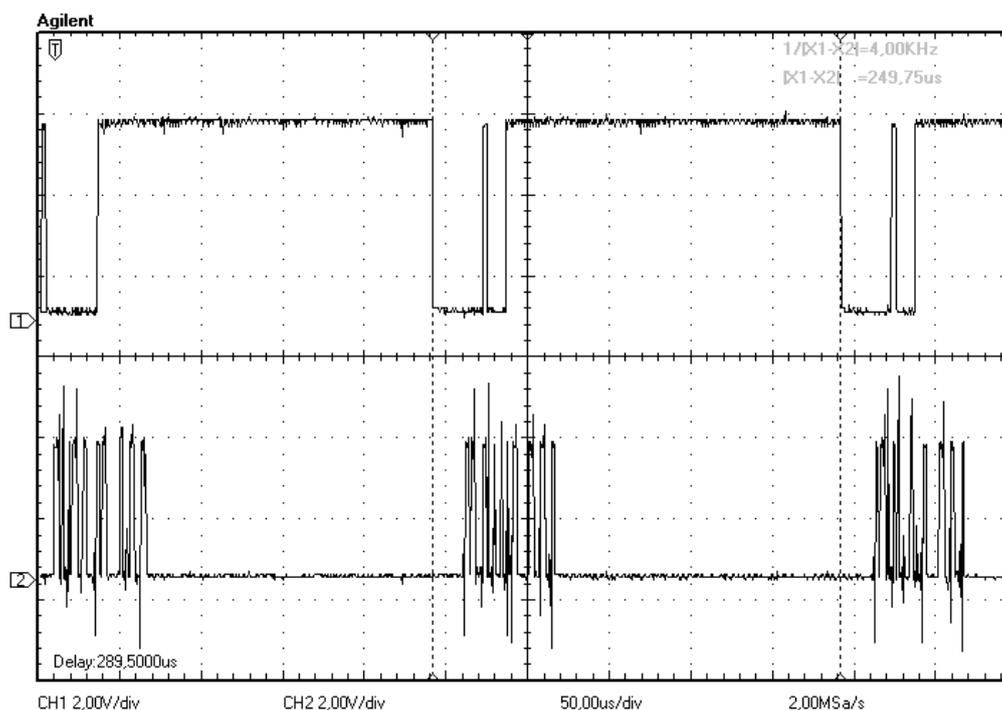


Figura 3-16 – Frequência de amostragem obtida no ADC

3. Módulo USB desenvolvido na placa ATMEL SAM7-H256

Utilizando o ambiente de desenvolvimento IAR Embedded Workbench foi projetado e implementado o *firmware*, em linguagem C, para a placa de desenvolvimento modelo ATMEL SAM7-H256 do fabricante *Olimex*. A função do mesmo é, além de controlar a comunicação com o ADC, receber os dados transmitidos pelo ADC e retransmití-los por meio da interface USB disponível a um PC.

O *firmware* em questão foi escrito de modo que a placa se comporte como um dispositivo USB da classe CDC e repasse os dados recebidos do ADC em pacotes do tipo *bulk*.

Essa configuração foi escolhida porque a classe CDC foi criada para transmissão contínua de dados, diferente de classes como HID, por exemplo, que transmite apenas quando existe uma interrupção.

Já os pacotes, são do tipo *bulk* porque têm a característica de garantir a confiabilidade dos dados. A desvantagem dos pacotes *bulk* é que não reservam banda para transmissão, mas isso não foi problema porque não se pretende ligar mais de um dispositivo por barramento.

Ao se conectar um dispositivo CDC na porta USB de um PC, o mesmo é reconhecido de forma transparente como uma porta de comunicação nos moldes da RS-

232, sendo assim compatível com virtualmente todos os sistemas operacionais. Além disso, tal característica possibilita a aquisição de dados de maneira mais fácil para futuros tratamentos em *software*, cujos ambientes de desenvolvimento têm pacotes para tratamento de dados recebidos por esse tipo de porta de comunicação.

O fluxograma abaixo mostra como funciona a parte USB do firmware:

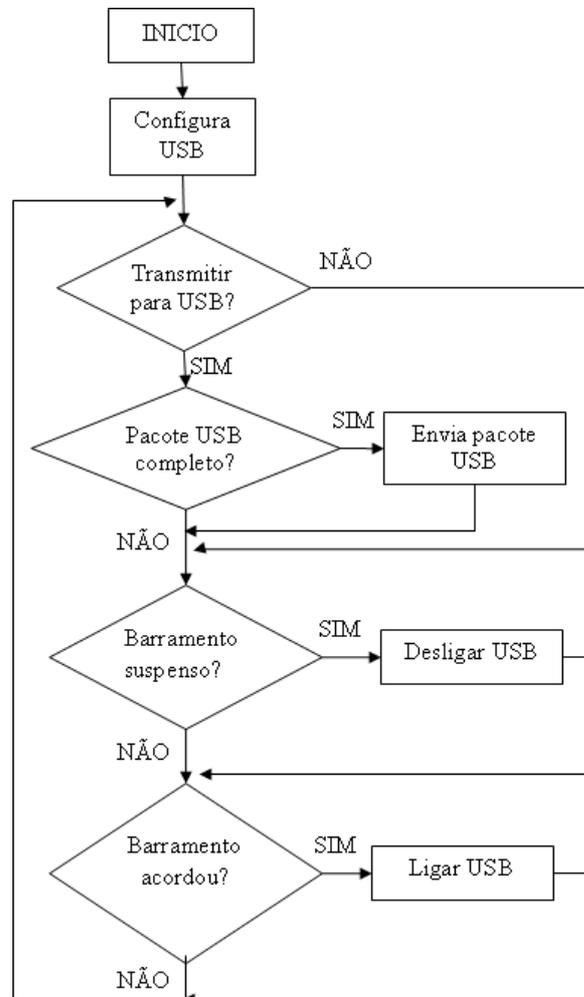


Figura 3-17 – Fluxograma da parte USB do *firmware*

O processo “Configura USB” é responsável por configurar a todas as opções do *framework* USB referentes à classe implementada, ligar as interrupções e permitir o processo de enumeração.

Depois da configuração e enumeração, o programa entra em um laço infinito que possui quatro processos decisórios. Os dois últimos são para colocar ou retirar o dispositivo USB do estado de economia de energia, desligando ou ligando o barramento. O primeiro verifica se foi ligada a função para transmitir pelo barramento USB, em caso positivo, os dados que serão transmitidos vão sendo inseridos em um vetor até que este chegue ao seu limite de tamanho, pré-definido no início do programa,

o que leva ao próximo processo decisório. Quando o vetor tem todos os seus bits preenchidos, inicia a transmissão dos dados armazenados neste pelo barramento.

No *firmware* foram definidos 4 comandos que são enviados pelo PC pelo barramento USB e interpretados pelo microcontrolador. Dois desses comandos são usados para configurar e calibrar o ADC, os outros dois são para iniciar e parar a digitalização do sinal e envio pelo barramento USB.

A tabela abaixo sumariza os comandos implementados no *firmware* da placa de desenvolvimento da *Olimex*:

Tabela 3-5 – Lista de comandos enviados ao ARM pelo barramento USB

Comando	Ação
D (ou d)	Digitalizar: liga a função que permite a conversão AD e a transmissão via USB quando o vetor atinge a quantidade de dados pré-configurada
P (ou p)	Parar: desliga a função de conversão, pára a conversão AD e o envio de dados pelo barramento USB
S (ou s)	<i>Setup</i> : configura o ADC
C (ou c)	Calibração: realiza a autocalibração do ADC

4. Software de plotagem das derivações cardíacas

Como parte final deste projeto foi desenvolvido um *software* na plataforma Visual Basic com o objetivo de plotar as 12 derivações de ECG provenientes dos sinais captados e tratados pela placa analógica, digitalizados pelo ADC e enviados via USB pelo processador ARM.

Em linhas gerais, o programa realiza as seguintes ações:

1. Abre a porta de comunicação virtual criada pela classe CDC implementada no *firmware*;
2. Envia o comando “D” quando se clica no botão “Digitalizar”;
3. Entra em um laço que recebe os 24 bits correspondentes ao primeiro ponto do gráfico de cada uma das 8 derivações oriundas do circuito e calcula as 4 demais, em seguida, recebe o segundo conjunto de bits do ponto seguinte e armazena em um vetor de dados. O software é dividido em duas seções: uma que recebe os dados e outra que desenha o gráfico, fazendo uma linha entre os pontos adjacentes;
4. O laço anterior continua até que seja acionado o botão “Parar”, então é enviado o comando “P” ao barramento USB, o ARM interpreta e para a

conversão e envio de dados ao PC, e então é fechada a porta de comunicação.

Neste ponto do projeto que foram projetados e implementados os filtros digitais com intuito de atenuar as interferências provenientes da rede elétrica, frequência de 60 Hz e suas harmônicas de 120 Hz e 180 Hz, tendo em vista que o sinal de ECG foi limitado à 200 Hz, não foram necessários filtros para atenuar harmônicas de ordens mais altas.

Os filtros implementados foram do tipo *notch* IIR de 2ª ordem. Esse tipo de filtro age ponto a ponto sobre o sinal realizando operações algébricas baseadas na taxa de amostragem do sinal de entrada e os coeficientes do filtro, calculados utilizando a ferramenta *fitfilt* do *software Matlab*.

Por ter uma grande dependência da taxa de amostragem do sinal de entrada e terem sido verificadas pequenas variações nessa taxa, quando implementados os filtros, ao contrário que se imaginava, se mostraram agentes de degradação do sinal. Por esse motivo e devido ao fato que o sinal obtido na plotagem foi capaz de exibir todas as ondas esperadas do sinal eletrocardiográfico, os filtros digitais não foram inseridos na versão final do *software* responsável pelo desenho dos gráficos de ECG.

A figura abaixo mostra o sinal eletrocardiográfico obtido onde podemos observar a interferência da rede elétrica, mas podemos verificar a presença de todas as ondas de ECG esperadas:



Figura 3-18 – Tela do software com as 12 derivações eletrocardiográficas contendo interferências da rede elétrica

Já na próxima ilustração, observamos um sinal livre de interferências devido ao fato do sistema estar ligado em baterias, isolando a interferência da rede elétrica, comprovando que o não funcionamento dos filtros digitais foram preponderantes ao desempenho do ECGAR desenvolvido neste trabalho:

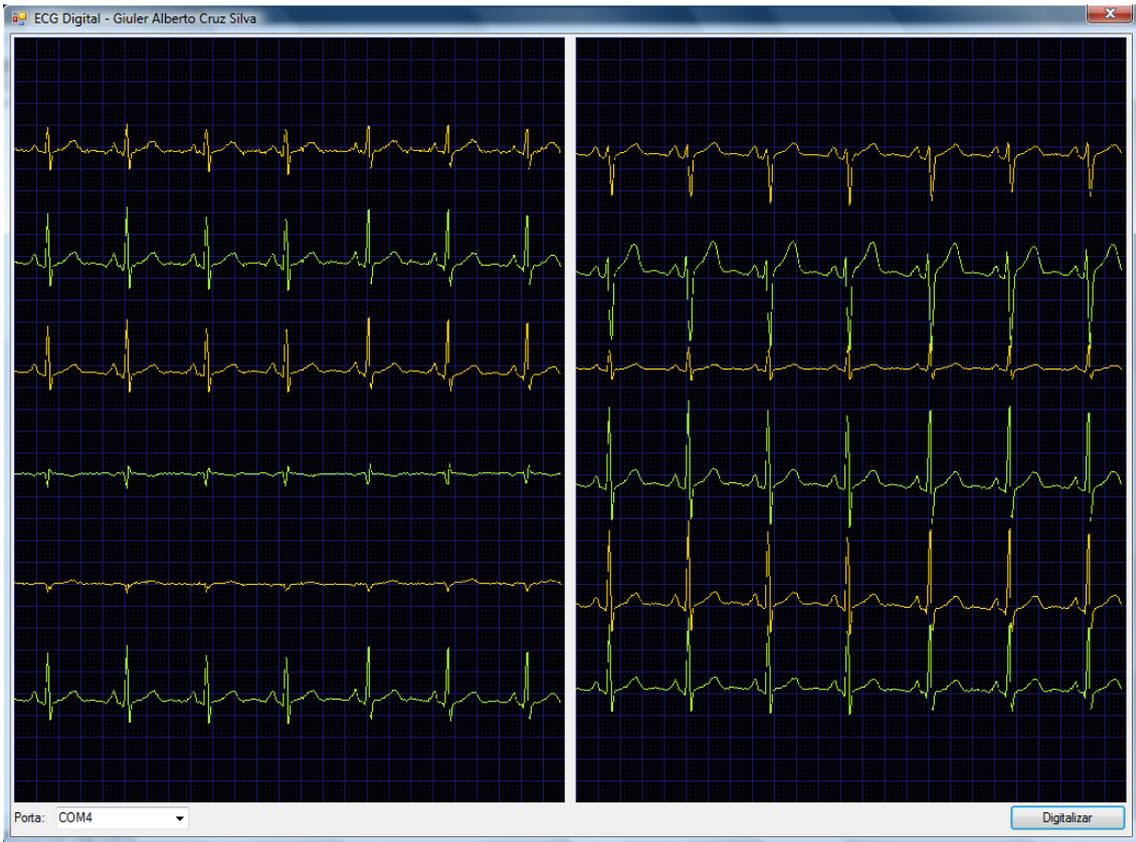


Figura 3-19 – Tela do software com as 12 derivações eletrocardiográficas livre de interferências da rede elétrica

Capítulo 4

Considerações Finais

3.1 Conclusões

Neste trabalho foi desenvolvido *firmware* que possibilitou transformar 8 sinais de ECG analogicamente tratados e disponibilizados nas portas de um ADC de 24 bits em um gráfico exibido na tela de um microcomputador, dando origem a um ECGAR que era o objetivo principal.

Os resultados foram satisfatórios apesar da restrição imposta pela taxa de amostragem do ADC, que em primeiro momento parecia mais que suficiente para as necessidades de projeto, mostrou-se bastante limitada por ser necessária a troca constante dos elementos a serem digitalizados.

O microcontrolador funcionou como o esperado e a placa de desenvolvimento, na qual estava inserido, se mostrou bastante versátil. O desenvolvimento do *firmware* foi facilitado por exemplos e materiais impressos fornecidos pelo fabricante do ARM, apesar do ambiente de programação não ser otimizado, demandando muito tempo e testes variados na tentativa de entender seu comportamento.

A única fragilidade do sistema criado aparece nas pequenas variações da taxa de amostragem dos sinais, provavelmente causada pela grande quantidade de interrupções de variadas prioridades que se intercalam com grande frequência. Tal fragilidade se torna visível quando são aplicados os filtros digitais projetados para eliminar a interferência de 60 Hz e suas harmônicas proveniente da rede elétrica, uma vez que para o bom funcionamento desses filtros, é necessária uma taxa de amostragem conhecida e constante.

3.2 Sugestões de melhorias

A primeira recomendação a ser feita é a alteração do ADC para outro modelo que possibilite uma maior taxa de amostragem e, conseqüentemente, uma maior robustez do sistema como um todo.

Com essa mudança, acredita-se que a instabilidade na taxa de amostragem vista no modelo de ADC atual desapareça e a implementação dos filtros digitais seja eficaz.

Em seguida, deverá ser aprimorado o programa para aquisição dos dados recebidos na porta USB para que seja capaz de fazer medidas diretamente sobre o sinal de ECG, uma vez que o software apresentado neste trabalho apenas possibilita a visualização e armazenamento da informação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] B. F. Hoffman & P. F. Cranefield McGraw-Hill. "*Eletrophysiology of the Heart*".
- [2] Masuda, Masako O.. *Coração, uma Bomba Movida a Eletricidade*. Instituto de Biofísica Carlos Chagas Filho, Universidade Federal do rio de Janeiro. Ciênci
- [3] The Electrocardiogram – looking at the heart of electricity,
- [4] http://nobelprize.org/educational_games/medicine/ecg/ecg-readmore.html
- [5] AXELSON, Jan. USB Complete. 2. ed., Estados Unidos da América: Lakeview Research, 2001.
- [6] ZELENOVSKY, Ricardo; MENDONÇA, Alexandre. Microcontroladores: Programação e Projeto com a Família 8051. 1. ed., Rio de Janeiro: MZ editora, 2005.
- [7] SALOMÃO, Andrei & FRIEDMAN, Alberto de Carvalho. Desenvolvimento de uma Interface USB para Comunicação com o Microcontrolador 8051. Projeto Final de Graduação. Brasília: Universidade de Brasília, 2005.