



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Interface gráfica para digitalização de sinal miográfico

Proposta de Trabalho de Conclusão de Curso

Elton Moreira Carvalho



São Cristóvão – Sergipe

2017

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Elton Moreira Carvalho

Interface gráfica para digitalização de sinal miográfico

Proposta de Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador(a): Daniel Oliveira Dantas

São Cristóvão – Sergipe

2017

Sumário

1	Introdução	5
2	Objetivos do projeto	6
3	Metodologia	8
3.1	Fases do Projeto	8
3.2	Cronograma de Atividades	8
3.3	Recursos	9
4	Fontes de pesquisa	10
	Referências	11

	Anexos	12
--	---------------	-----------

Lista de ilustrações

Figura 1 – Tela Principal	13
Figura 2 – Configuração de Exibição	13
Figura 3 – Configuração de Captura	14

Lista de tabelas

Tabela 1 – Cronograma das atividades previstas	8
Tabela 2 – Legenda do cronograma das atividades previstas	9

1

Introdução

Desde os primórdios o ser humano procura maneiras de suprir os mais diversificados problemas encontrados no seu cotidiano, partindo de coisas simples como a dificuldade de trocar o canal da televisão, até outras um pouco mais complexas como a substituição de membros do corpo humano perdidos ou nunca desenvolvidos naturalmente.

Utilizando uma técnica de monitoramento da atividade elétrica das células musculares denominada eletromiografia é possível o desenvolvimento de próteses artificiais que simulam um membro do corpo humano, porém um conjunto de hardware e software capaz de captar e visualizar este tipo de sinal ainda é muito caro e de difícil acesso para a comunidade acadêmica, pensando nisso este plano de trabalho tem o objeto de desenvolver uma interface de software *opensource* capaz de exibir e armazenar sinais mioelétricos captados por um hardware externo desenvolvido pelo Departamento de Computação da Universidade Federal de Sergipe.

2

Objetivos do projeto

Este plano de trabalho tem como objetivo criar um software, especificamente uma interface gráfica e *firmware* que permita um computador se comunicar com a placa TIVA EK-TM4C123GXL que irá receber, exibir e armazenar os sinais eletromiográficos capturados por um circuito externo desenvolvido no Departamento de Computação (DComp) da Universidade Federal de Sergipe (UFS), tornando assim fácil a utilização e tratamento destes sinais por outros pesquisadores.

A placa TIVA possui 12 conversores A/D, portanto, pode capturar 12 sinais a cada leitura. Será conectada a placas feitas sob medida, para capturar os sinais miográficos. Cada placa poderá possuir até 12 canais e os sinais podem ser multiplexados caso haja necessidade. A interface gráfica permitirá a escolha de parâmetros de captura e de exibição. Os parâmetros de captura devem ser transmitidos à placa TIVA por meio de um protocolo. Os parâmetros de exibição afetam apenas a visualização do sinal na tela do computador.

O software deve ser chamado através de um script bash que redireciona a saída para o arquivo especificado pelo primeiro parâmetro. O programa salvará a série temporal em um vetor na memória, pois no futuro pode ser necessário implementar outras funcionalidades mais flexíveis de visualização, exibição e edição do sinal. Para acelerar a alocação e a execução, o vetor deve crescer em blocos ao invés de uma posição por vez. O programa deve ter controle do número de posições usadas do vetor.

A interface gráfica não conhece a configuração do hardware, logo deve ser genérica de modo que funcione com qualquer configuração. O usuário, por sua vez, deve ser capaz de indicar a configuração correta do hardware, por exemplo, número de canais a serem lidos e número de placas a serem multiplexadas.

A placa TIVA recebe a solicitação e retorna uma informação de erro ou sucesso conforme sua capacidade. Futuramente pode-se definir um conjunto de funções para consultar a placa quanto a suas capacidades e configuração.

Anexo a este documento possui algumas imagens de como deverá ser a interface básica do software.

3

Metodologia

3.1 Fases do Projeto

O projeto será dividido em três fases principais podendo existir sub-fases entre cada uma. A primeira fase é focada no estudo da plataforma TIVA e das funcionalidades do Python 2.7 para o desenvolvimento da interface gráfica e *firmware*, nesta fase será definida também as especificações do software. A segunda fase consiste no desenvolvimento do software em Python 2.7 referente à interface gráfica, seguindo as especificações da fase anterior. Por fim a terceira fase será destinada ao desenvolvimento da rede neural responsável por classificar os sinais obtidos pelo software.

3.2 Cronograma de Atividades

Atividade	Meses											
	Out/17	Nov/17	Dez/17	Jan/18	Fev/18	Mar/18	Abr/18	Mai/18	Jun/18	Jun/18	Jul/18	Ago/18
01	X	X										
02	X	X										
03	X	X										
04		X	X	X	X							
05				X	X	X	X					
06								X				
07									X	X		
08										X	X	X

Tabela 1 – Cronograma das atividades previstas

Legenda	
01	Estudar plataforma TIVA EK-TM4C123GXL
02	Estudar o Code Composer Studio (CCS)
03	Estudar documentação Python 2.7
04	Desenvolver interface gráfica em Python 2.7
05	Desenvolver código de integração com a TIVA
06	Testar e validar os dados obtidos durante o processo
07	Estudar Redes Neural
08	Implementar e treinar a Rede Neural

Tabela 2 – Legenda do cronograma das atividades previstas

3.3 Recursos

Será necessário um hardware que realiza a captura dos sinais eletromiográficos, uma placa TIVA C Series TM4c123G para a conversão destes sinais e um ambiente Python 2.7 devidamente configurado para o desenvolvimento do software.

4

Fontes de pesquisa

Para fins de estudo será utilizada a documentação fornecida pela *Texas Instruments* referente à placa EK-TM4C123GXL ([INSTRUMENTS, 2017a](#)) e ao ambiente de programação Code Composer Studio ([INSTRUMENTS, 2017b](#)). A documentação da linguagem de programação Python ([PYTHON, 2017](#)) e de suas bibliotecas QtPy4 ([PYQT4, 2017](#)), Serial ([PYSERIAL, 2017](#)) e PyQtGraph ([PYQTGRAPH, 2017](#)) servirá de base para o desenvolvimento do software de recepção e exibição dos sinais capturados.

A fundamentação teórica do trabalho será construída utilizando a base do *Institute of Electrical and Electronics Engineers* ([IEEE, 2017](#)) como principal fonte de pesquisa dos artigos e trabalhos relacionados ao tema aqui proposto.

Referências

IEEE. *IEEE Explore Digital Library*. 2017. Disponível em: <<http://ieeexplore.ieee.org.ez20.periodicos.capes.gov.br/Xplore/guesthome.jsp?reload=true>>. Acesso em: nov 2017. Citado na página 10.

INSTRUMENTS, T. *ARM® Cortex®-M4F Based MCU TM4C123G LaunchPad™ Evaluation Kit*. 2017. Disponível em: <<http://www.ti.com/tool/EK-TM4C123GXL>>. Acesso em: nov 2017. Citado na página 10.

INSTRUMENTS, T. *Code Composer Studio v7 Wiki*. 2017. Disponível em: <http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v7>. Acesso em: nov 2017. Citado na página 10.

PYQT4. *PyQt4 Reference Guide*. 2017. Disponível em: <<http://pyqt.sourceforge.net/Docs/PyQt4/>>. Acesso em: nov 2017. Citado na página 10.

PYQTGRAPH. *PyqtGraph Documentation*. 2017. Disponível em: <<http://www.pyqtgraph.org/documentation/>>. Acesso em: nov 2017. Citado na página 10.

PYSERIAL. *pySerial's documentation*. 2017. Disponível em: <<https://pythonhosted.org/pyserial/>>. Acesso em: nov 2017. Citado na página 10.

PYTHON. *Python 2.7.14 documentation*. 2017. Disponível em: <<https://docs.python.org/2/>>. Acesso em: nov 2017. Citado na página 10.

Anexos

Figura 1 – Tela Principal



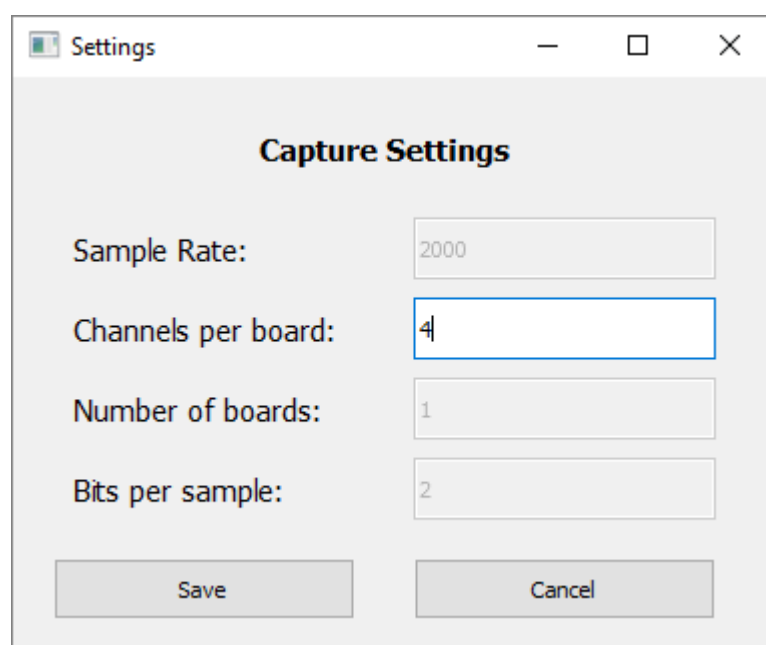
Fonte: o autor

Figura 2 – Configuração de Exibição

The screenshot shows a 'Settings' dialog box with a title bar containing a gear icon and the text 'Settings'. The dialog has standard window controls (minimize, maximize, close). The main content area is titled 'Display Settings'. It contains several configuration options, each with a label and a corresponding input field: 'Swipe:' with a text box containing '1000'; 'Zero:' with a text box containing '0'; 'Vertical Tick:' with a text box containing '1'; 'Horizontal Tick:' with a text box containing '100'; 'Channels' with a text box containing '4'; and 'Amplitude:' with two text boxes, the first containing '0' and the second containing '4'. At the bottom of the dialog, there are two buttons: 'Save' and 'Cancel'.

Fonte: o autor

Figura 3 – Configuração de Captura



The image shows a software window titled "Settings" with standard Windows window controls (minimize, maximize, close). Inside the window, the title "Capture Settings" is centered. Below the title, there are four configuration options, each with a text label and a corresponding input field:

- Sample Rate:** The input field contains the value "2000".
- Channels per board:** The input field contains the value "4".
- Number of boards:** The input field contains the value "1".
- Bits per sample:** The input field contains the value "2".

At the bottom of the dialog, there are two buttons: "Save" on the left and "Cancel" on the right.

Fonte: o autor