COBENGE 2014
ENGENHARIA: Múltiplos saberes e atuações
16 a 19 de setembro | Juiz de Fora - MG

SISTEMA DE MEDIÇÃO DE TEMPO DE RESPOSTA: UMA APLICAÇÃO PARA A MEDIÇÃO DE TEMPO DE REFLEXO E TREINAMENTO DE ATLETAS

Pedro Buarque Caminha Monteiro – pbcm2@ecomp.poli.br
Escola Politécnica de Pernambuco - Universidade de Pernambuco
Rua Benfica, 455. Madalena. CEP 50.720-001 – Recife - Pernambuco

Sérgio Campello Oliveira – scampello@ecomp.poli.br
Escola Politécnica de Pernambuco – Universidade de Pernambuco
Rua Benfica, 455. Madalena. CEP 50.720-001 – Recife - Pernambuco

Resumo: *Sistemas de medição de tempo de resposta podem ser utilizados em diversas áreas da engenharia. Por exemplo, a medição do tempo de resposta de pessoas idosas pode ajudar a identificar a capacidade cognitiva bem como servir como estímulo para a sua melhoria. Sistemas computacionais podem ser utilizados para a construção desses sistemas, porém as interfaces de entrada e saída presentes nos computadores pessoais tradicionais não são adequados para esse tipo de aplicação, uma vez que podem ser necessárias dimensões bem maiores. Para solucionar o problema das múltiplas interfaces digitais necessárias para ativação das saídas e medição dos tempos de resposta podem ser utilizados sistemas microcontrolados. Este artigo apresenta o desenvolvimento de um sistema microcontrolado capaz de ativar 24 diferentes saídas e medir o tempo de reação. As saídas são luminosas compostas por um conjunto de LEDs e as entradas são botões acionados manualmente. O sistema completo será utilizado para o treinamento de atletas. O conjunto de entradas e saídas pode ser programado para gerar sequências de treinamento que exijam o deslocamento lateral, agachamentos e saltos para o acionamento dos botões. Esse tipo de treinamento agregará a capacidade de redução do tempo de reação bem como do aumento da agilidade do atleta. Os primeiros testes serão feitos em atletas da modalidade Tênis que são bastante exigidos nas modalidades de reação citadas. Todo o trabalho foi desenvolvido no âmbito de um trabalho de conclusão de curso demonstrando como a interação entre as áreas, educação física e engenharia de computação neste caso, pode ser motivante para os alunos da graduação em engenharia.*

Palavras-chave: *Tempo de resposta, Sistemas microcontrolados, Treinamento de atletas*

1. INTRODUÇÃO

Com a necessidade de sistemas computacionais cada vez mais compactos e eficientes, a utilização de microcontroladores se tornou mais evidente em quase todas as áreas de aplicação. Com o aumento da complexidade dos projetos de sistemas embarcados, tem exigido novos níveis de abstração em soluções de software que possam interagir com o *hard-ware* da forma mais eficiente possível.

Dentro deste contexto, destacam-se sistemas que trabalham com medições de tempo de respostas a determinados estímulos ou eventos. Um exemplo são os sistemas de tempo real, onde suas respostas ao ambiente devem ser dadas dentro de um tempo hábil o suficiente para que o sistema não entre em um estado inconsciente. Análises de tempo de reação são úteis também, por exemplo, para identificar melhorias no estado cognitivo de idosos que praticam atividades físicas, ou aumentar o desempenho de atletas nos esportes.

A proposta deste trabalho é a implementação de um sistema computacional embarcado para realizar a medição de tempo entre eventos utilizando nós sensores. Como teste inicial será desenvolvido um sistema capaz de auxiliar no treinamento do tempo de reação de atletas. O treinamento será realizado a partir de um software controlado pelo usuário, que irá enviar comandos para uma placa microcontrolada. Esta ficará responsável por transmitir esses comandos para cada um dos nós sensores que funcionarão como estímulos para o atleta. Deverá ser mensurado o intervalo de tempo entre o recebimento do estímulo por parte do atleta e a ação de resposta realizada pelo mesmo ao apertar um botão, indicando o fim da contagem.

2. DESENVOLVIMENTO DO HARDWARE

O *hardware* desenvolvido neste projeto é dividido em três partes: uma placa microcontrolada que ficará responsável por receber comandos de um computador e transmiti-los para cada uma das placas de interação com o atleta, obedecendo todas as configurações definidas no software; uma placa de conexões que fará as ligações entre a placa controladora e todas as placas de interação; vinte e quatro placas utilizadas para a interação com o atleta. A "Figura 1" ilustra o diagrama de blocos de todo o *hardware* utilizado no projeto.

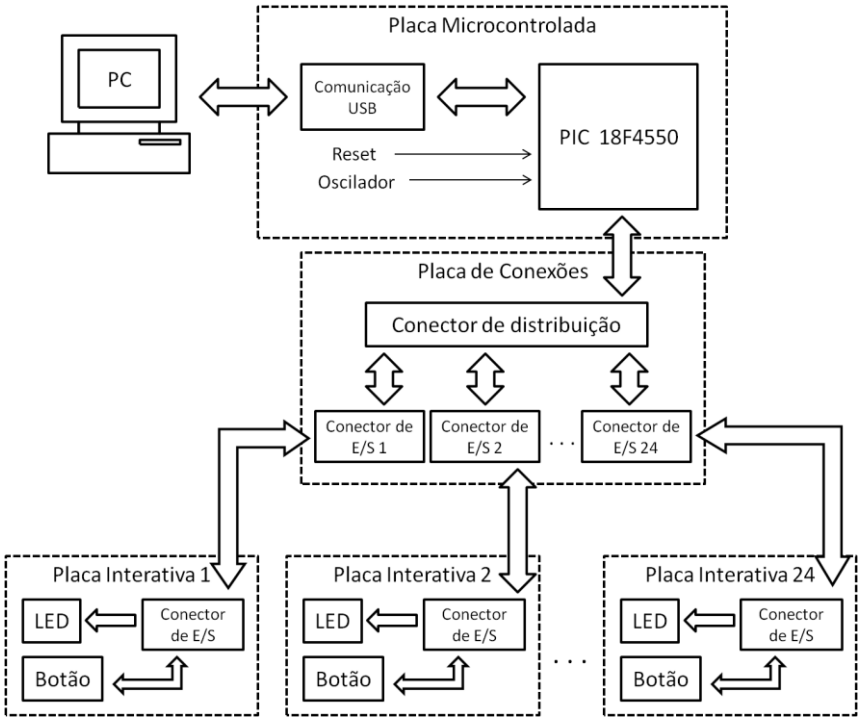

Figura 1 – Diagrama de blocos do *hardware*.

2.1. Placa microcontrolada

O funcionamento de todas as partes da aplicação dependerá de uma placa principal, responsável por enviar e receber comandos para que o treinamento seja realizado. Esses comandos serão enviados via USB por um sistema de interface com o usuário, indicando qual placa de interação irá estimular o atleta. O circuito da placa controladora é dividido em seis partes, são elas: o microcontrolador PIC18F4550,o circuito oscilador, a gravação com o *bootloader*, a comunicação USB, o circuito de re-inicialização e o conector de entrada e saída como ilustra a "Figura 2".

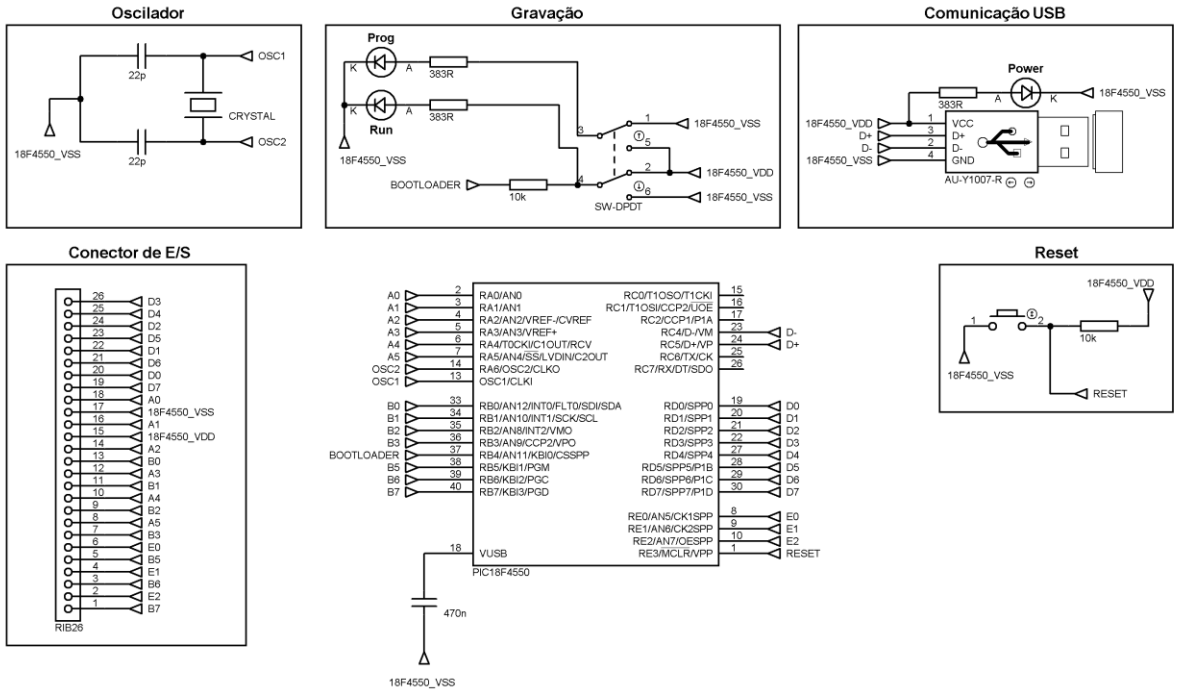

Figura 2 – Circuito da placa microcontrolada.

Microcontrolador PIC18F4550

O microcontrolador explorado neste projeto é o PIC18F4550 (Microchip, 2004) que possui todas as vantagens da família PIC18 (alto desempenho computacional a um preço acessível) com adição de um material de alta resistência e programação avançada em memória flash. O PIC18F4550 apresenta melhorias no projeto que tornam esses microcontroladores uma ótima escolha para muitas aplicações de alto desempenho. O PIC utilizado apresenta funcionalidades como: durabilidade de memória, auto-programação, Universal Serial Bus (USB), portas bidirecionais de entrada e saída, quatro *timers* disponibilizados.

Oscilador

O oscilador ou *clock* do microcontrolador pode ser configurado de quatro maneiras distintas, dependendo do uso. Esta configuração é feita via software e é aceita pelo microcontrolador durante sua gravação. O *clock* determinará a velocidade de operação do microcontrolador. Dentre os modos de oscilação existentes, o XT é a configuração mais

utilizada, sendo necessário apenas um cristal e dois capacitores ligados ao PIC. Neste circuito é utilizado um cristal de 20 MHz e dois capacitores de 22 pF.

Gravação

O *bootloader* é um *firmware* que quando instalado no microcontrolador, permite a gravação de programas diretamente através de uma porta USB. Para a gravação do *bootloader* será necessário a utilização de um gravador convencional uma única vez. O circuito de gravação com o *bootloader* possui uma chave DPDT para realizar a seleção entre o modo de gravação e execução. Quando ativado o modo de gravação (o LED "Prog" acenderá), será enviado um sinal alto para o pino RB4 indicando que o PIC entrou em modo *boot*. Da mesma forma, quando ativado o modo execução (o LED "Run" acenderá), será enviado um sinal baixo para o pino RB4 desabilitando o modo boot do PIC.

Comunicação USB

O circuito de comunicação via USB é de fundamental importância para o projeto pois com ele é possível a realização do treinamento, já que é através da USB que ocorre a comunicação entre o sistema de interface com o usuário e a placa controladora. Como mostrado anteriormente a comunicação USB será essencial também para a atualização do *firmware* na própria placa, sem a necessidade de remoção do microcontrolador.

Reset

O PIC possui internamente circuitos que controlam o *Power-on reset*. Basicamente isso significa que o microcontrolador necessita de poucos componentes externos para realizar o *start-up*. O circuito de *reset* utiliza um resistor, ligado entre o pino de MCLR e o VCC, no valor de 10k *ohms* e uma chave para realizar o reset manual, ligada entre o MCLR e o "terra".

Conector de E/S

Para realizar a transmissão e recepção dos dados, foi utilizado um conector *header* de 26 vias para cabos *flat*. Serão utilizadas 24 vias para a transmissão de sinais dos pinos do microcontrolador, uma via para o VCC e uma para o "terra".

2.2. Placa de conexões de E/S

Assim como mostra a "Figura 1", os componentes que fazem essa placa são apenas conectores responsáveis por toda a distribuição do sistema. A placa de conexões de E/S foi adicionada ao projeto para tirar essa responsabilidade da placa microcontrolada e deixar o todo o sistema mais simples e fácil de montar. Esta placa possui um conector central de distribuição e 24 conectores para transmitir os sinais para cada uma das placas de interação. Para o conector de distribuição foi usado um conector *header* de 26 vias para cabos *flat*, e para cada conector de E/S foi utilizado conectores KK de 3 vias com os sinais do pino, VCC e "terra".

2.3. Placa interativa

A principal função da placa interativa é acender um LED para que o atleta seja estimulado visualmente e receber o sinal lógico de um botão que representará a resposta ao estímulo provocado. Para facilitar o treinamento e a visão do atleta, foram colocados três LEDs como ilustra a "Figura 4".

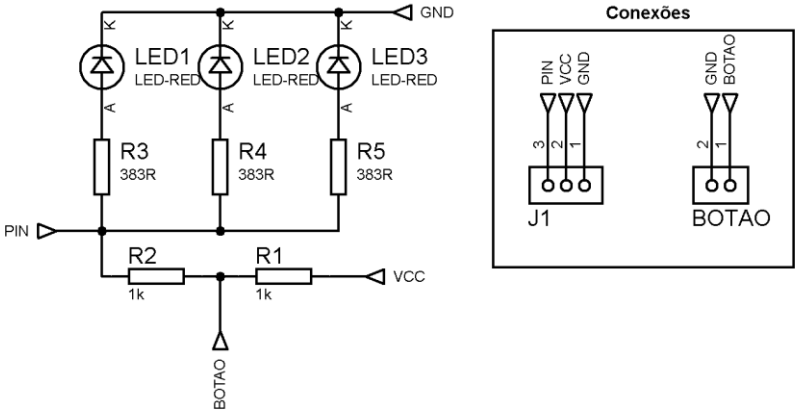

Figura3 – Circuito da placa interativa.

Todas as placas de interação ficarão fixadas em um painel de algum material resistente. Para evitar que a placa quebre com o impacto, o botão será também fixado no painel. Para representar o botão na placa, foi colocado um conector de duas vias para passar os sinais para o botão do painel.

Como o número de portas do microcontrolador é limitado, foi necessário a utilização de pinos bidirecionais de E/S do PIC. Dependendo dos recursos habilitados, existem até cinco portas disponíveis. Nas portas bidirecionais, os sinais são lidos ou escritos via registradores da porta. A direção dos sinais é controlada pelo registrador TRIS. Existe um registrador TRIS para cada porta, com os valores um para leitura e zero para escrita (BOLTON, 2010). Como mostra o circuito da "Figura 2", os três LEDs estão ligados ao pino do microcontrolador juntamente com o botão.

3. DESENVOLVIMENTO DO FIRMWARE

Segundo (PEREIRA, 2003), todo compilador possui uma lista de comandos internos que não são diretamente traduzidos em código, esses comandos são utilizados para especificar determinados parâmetros internos utilizados pelo compilador e são chamados de diretivas do compilador. Sendo assim, o primeiro passo para o desenvolvimento do software embarcado foi a definição das diretivas de compilação iniciais. Podem ser observadas algumas diretivas na "Tabela 1".

Tabela 1 – Diretivas de compilação

Diretivas	Descrição
#define	Definição de constantes de cadeia de caracteres
#include	Incluir arquivos de códigos fontes
#fuses	Define os bits de configuração
#use delay	Informa ao compilador o valor da frequência em Hz

A diretiva *#use delay* informa ao compilador qual o *clock* de operação da CPU do microcontrolador e os valores são informados em Hz. A operação deste sistema é baseada em uma frequência de 20 MHz.

A diretiva *#include* indica arquivos com códigos de programação extras. Esses arquivos de código fonte são adicionados ao programa principal no momento da compilação. No programa desenvolvido foram utilizadas as seguintes inclusões:

```
#include <18F4550.h>
#include <pic18_usb.h>
#include <usb.c>
#include "usb_desc_hid.h"
#include <fuses_config.h>
#include <trismap.h>
```

O arquivo “18F4550.h”, fornecido pelo fabricante, importa funções específicas do PIC 18F4550 tais como manipulação de temporizadores, entrada e saída de dados, etc. Os arquivos “pic18_usb.h” e “usb.c”, fornecidos pelo fabricante, importam funções necessárias para que a comunicação USB funcione corretamente. O “usb_desc_hid.h” contém a implementação dos descritores USB, esse arquivo contém informações específicas como os identificadores de produto e fabricante do dispositivo, necessários para a identificação deste no sistema de interface com o usuário.

Os dois últimos arquivos foram criados neste projeto para facilitar o desenvolvimento e entendimento do código. Na biblioteca “fuses_config.h” são utilizadas diretivas *#fuses* para informar os bits de configuração específicos do microcontrolador PIC18F4550. Na biblioteca “trismap.h” são utilizadas diretivas de *#define* para definir o endereço dos registradores TRIS de cada porta, assim como rotular os bits para facilitar o desenvolvimento.

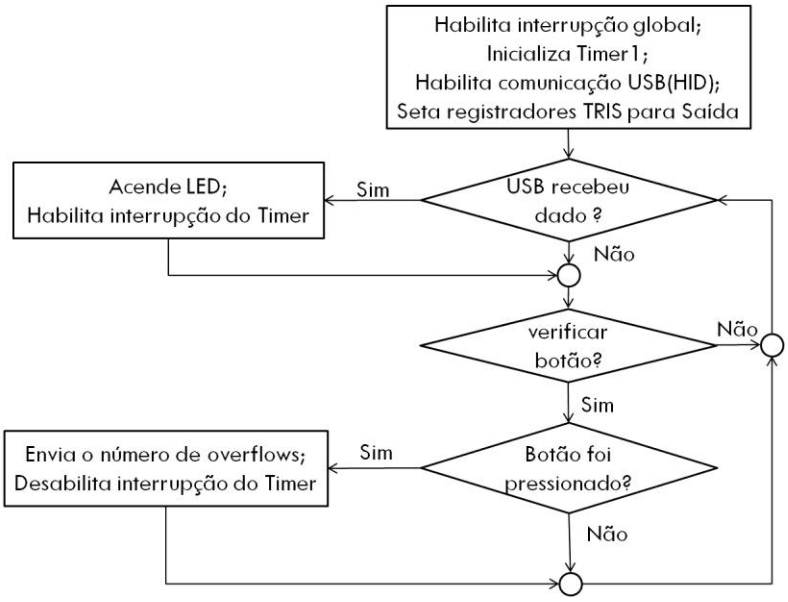

Figura 4 – Fluxograma do programa principal

A "Figura 6" ilustra o fluxograma do programa principal que é baseado em realizar uma comunicação USB com o sistema externo, podendo receber e enviar comandos para que o treinamento seja realizado. Para que todas as interrupções do PIC possam ser manipuladas, inicialmente o registrador da interrupção global deve ser habilitado.

Dentre os temporizadores disponibilizados pelo microcontrolador, será utilizado o *timer 1* para realizar a contagem do tempo de resposta do atleta. O *timer1* é um temporizador/contador de 16 bits, e caso sua interrupção seja habilitada é gerado um *overflow*. Portanto o tempo de resposta será calculado a partir da quantidade de *overflows* gerados pelo *timer 1*. Para a inicialização do *timer 1*, são executados os códigos abaixo:

```
Setup_timer_1(T1_INTERNAL | T1_DIV_BY_8);
Set_timer1(0);
```

Na primeira função são utilizados dois parâmetros de configuração, o "*T1_INTERNAL*" indica que o *timer* será incrementado pelo *clock* interno, e o "*T1_DIV_BY_8*" indica que a frequência do *timer* será dividida por 8. Já a segunda função força o *timer 1* a iniciar com o valor zero. Ainda na fase inicial do programa, é habilitada a comunicação USB, e os registradores TRIS das portas são configurados inicialmente como saída.

De acordo com o fluxograma da "Figura 6", o programa irá permanecer em um *loop* infinito enquanto a placa microcontrolada estiver ligada. Após a comunicação USB ser estabelecida corretamente, deverão ser realizadas duas verificações a cada iteração: verificação de recebimento de dados via USB e verificação de permissão para leitura do botão.

Recebimento de dados via USB

A cada iteração será verificado se foi recebido algum pacote vindo da USB através da função "*usb_kbhit(1)"*. Em caso positivo, o pacote é recebido e tratado para cada tipo de comando definido, como mostra a "Tabela 2".

Tabela 2 – Comandos do treinamento

Comando	Função
0	O treinamento deverá ser interrompido.
1 - 24	Ligar LED referente ao número.

Se o comando for "0", o treinamento será finalizado e todos os LEDs serão desligados. Porém, se o comando for entre "1" e "24"**,** os registradores TRIS de todas as portas serão setados para saída e o LED referente ao número do comando será ligado. Assim que o LED acender, será habilitada a interrupção do *timer 1* para que a contagem seja iniciada.

Permissão para leitura do botão

Já que os pinos do microcontrolador são utilizados tanto para entrada (leitura do botão) como saída (acender LED), o tempo em que cada ação é realizada deve ser dividido. Sendo assim, foi criado um contador que será incrementado a cada iteração para verificar se a leitura do estado lógico do botão pode ser realizada. Ficou definido que para 99% do tempo os registradores TRIS estarão setados para "saída" e somente 1% para "entrada". Esse cuidado foi levado em consideração para que a mudança de um estado para outro fosse imperceptível para o usuário, já que o LED que foi ligado inicialmente deverá ser desligado para que o estado do pino seja conferido.

Assim que houver a permissão para leitura, o registrador TRIS referente ao pino será setado como "Entrada", será lido o estado do botão e o registrador TRIS setado para "Saída" novamente para que o LED seja novamente aceso. Caso o botão seja pressionado, será enviado um pacote via USB contendo o número de overflows contados pelo *timer 1*, a interrupção será desabilitada e o *timer* reiniciado.

4. DESENVOLVIMENTO DO SISTEMA EXTERNO

O sistema de interface com o usuário foi desenvolvido neste projeto com o objetivo principal de acompanhar o treinamento passo a passo, realizar a criação de sequências de comandos para o sistema embarcado e a possibilidade de geração de relatórios. Foi utilizada a linguagem C# para o desenvolvimento da aplicação juntamente com a biblioteca "USB *Library*", disponibilizada na internet e responsável pela comunicação USB do aplicativo. Como ilustra a "Figura 6" a tela principal é dividida em três partes, são elas: a barra de ferramentas, o painel de treinamento e a barra de *status*.

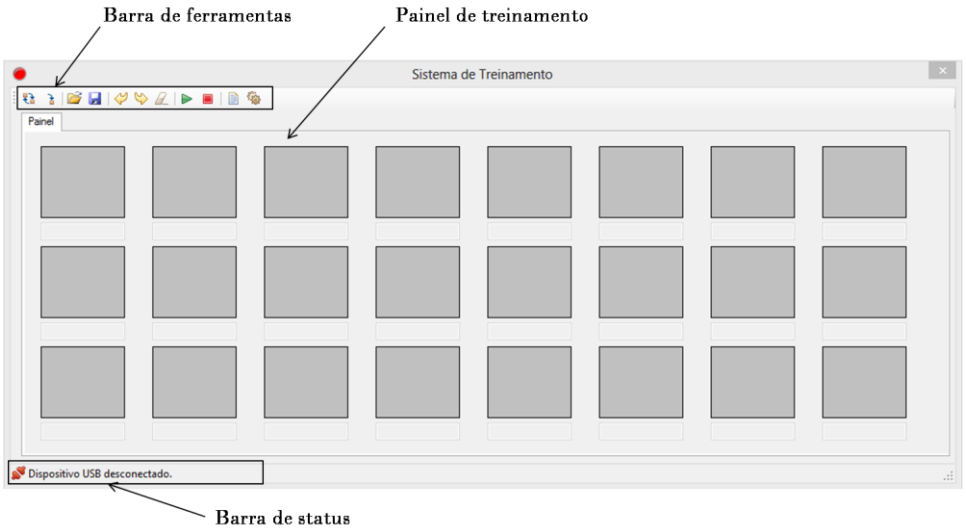

Figura 5 – Tela principal

A barra de ferramentas tem a finalidade de permitir uma ação rápida por parte do usuário, facilitando o acesso a todas as funcionalidades do sistema. As funções que o sistema oferece são: criação de sequências no modo manual e automático, opções de carregar e salvar

as sequências geradas, iniciar e parar o treinamento, geração de relatório e ajustes de configurações do sistema.

Tudo o que será realizado no treinamento poderá ser visualizado no painel de treinamento. Será mostrada a sequência escolhida pelo usuário, o tempo de resposta do atleta e o tempo de acendimento entre um LED e outro poderá ser escolhido. A barra de *status* tem como finalidade informar ao usuário a situação da comunicação USB com o dispositivo, exibindo se ele foi ou não conectado.

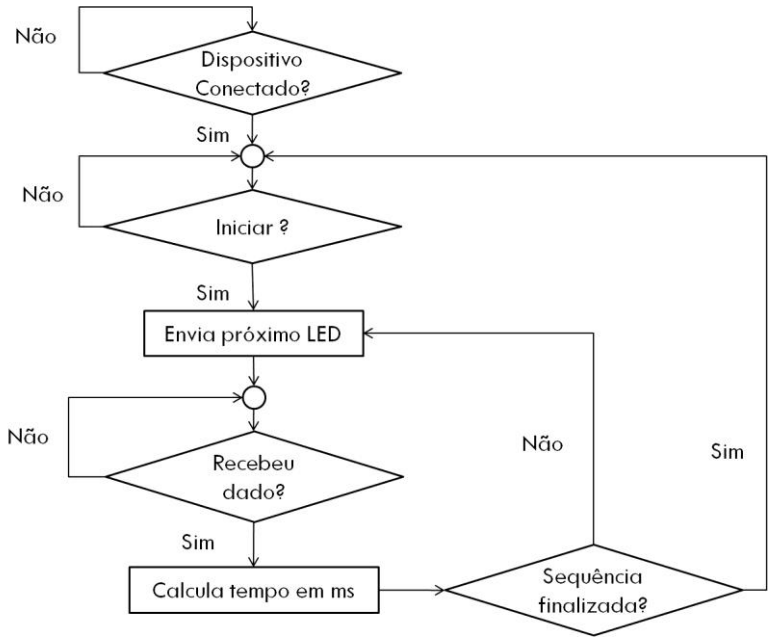

Figura 6 – Fluxograma do sistema externo

De acordo com o fluxograma ilustrado na "Figura 8", o sistema deverá aguardar até que o dispositivo seja conectado, assim o treinamento poderá ser iniciado se alguma sequência foi definida pelo usuário. A sequência possui números de 1 a 24, no modo automático é gerado de forma aleatória, já no modo manual o usuário deverá escolher a ordem da sequência podendo haver repetição de números.

Iniciando o treinamento com a sequência já definida, o próximo passo é enviar o primeiro comando para o dispositivo. Se o comando for o número "5", por exemplo, o quinto LED do painel será aceso, já que os LEDs estão numerados com valores de 1 a 24. A cada comando enviado, o sistema aguardará até que o evento de recepção de dados pela USB seja disparado.

Assim que o for detectada o recebimento de dados, o sistema irá calcular o tempo de resposta do atleta em milissegundos a partir do número de *overflows* enviado pelo *firmware*. Segundo (PEREIRA, 2003), o cálculo total do tempo do *timer 1* é dado por:

$$T = T_{TMR} \times prescaler \times 2^{bits} \times overflows$$

$$T_{TMR} = 4 / f_{CLK}$$

A variável "T_{TMR}" é o período do *clock* que vai na entrada do *prescaler*, que no caso deste projeto equivale a 4 / 20 MHz. Ao passar pelo prescaler ocorre uma divisão de frequência de T_{TMR}, isto equivale a multiplicação do período pelo fator de divisão do *prescaler*, que no exemplo do projeto é 8. Para gerar cada estouro no *timer 1* são necessárias 2^{16} contagens, então basta multiplicar pelo número de estouros recebido. Calculado o tempo de resposta do atleta, o ciclo se repete até toda a sequência escolhida termine, finalizando o treinamento.

5. TESTES UTILIZANDO FERRAMENTA DE SIMULAÇÃO

A simulação do sistema e a criação dos circuitos mostrados neste trabalho foram realizadas com a utilização da ferramenta computacional *Proteus*, da empresa *Labcenter Eletronics*. O uso do *Proteus* foi essencial neste trabalho, para a execução de testes de funcionamento dos circuitos e na depuração do código embarcado, auxiliando na correção de erros de programação.

Para a simulação foi criado um conjunto de *LEDs* e botões para representar um painel a ser utilizado por atletas. Como já mencionado, o painel conterá 24 *LEDs* e 24 botões correspondentes alinhados como uma matriz de 3x8, como mostra a "Figura 7".

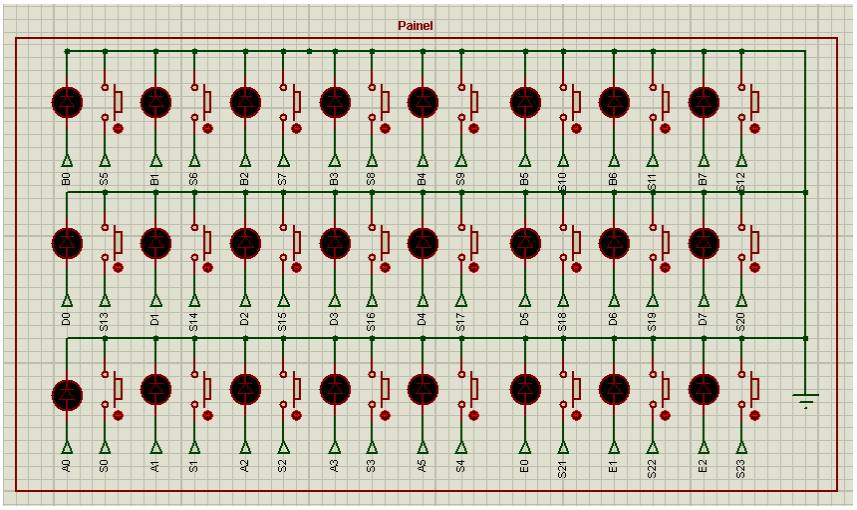

Figura 7 – Painel de treinamento

Como teste inicial foi utilizada a sequência de comandos: "18", "12", "1", "3", "9" que representam os números do conjunto LED/Botão que será acionado. Para cada LED da sequência serão adotados os respectivos tempos de espera para o acendimento: "1", "3", "0", "1", "2". A execução dos comandos pode ser observada na "Figura 8" mostrando a interação entre a simulação no *Proteus* e a aplicação desenvolvida. Pode ser observado na "Figura 8" que o tempo da resposta do atleta obtido é exibido assim que o comando é finalizado.

Assim que o treinamento é finalizado, poderá ser gerado um relatório com informações referentes ao desempenho do atleta. Como mostra a "Figura 9", serão salvas informações como nome do atleta, idade, data e hora do treinamento, e os tempos de respostas para cada LED da sequência utilizada.

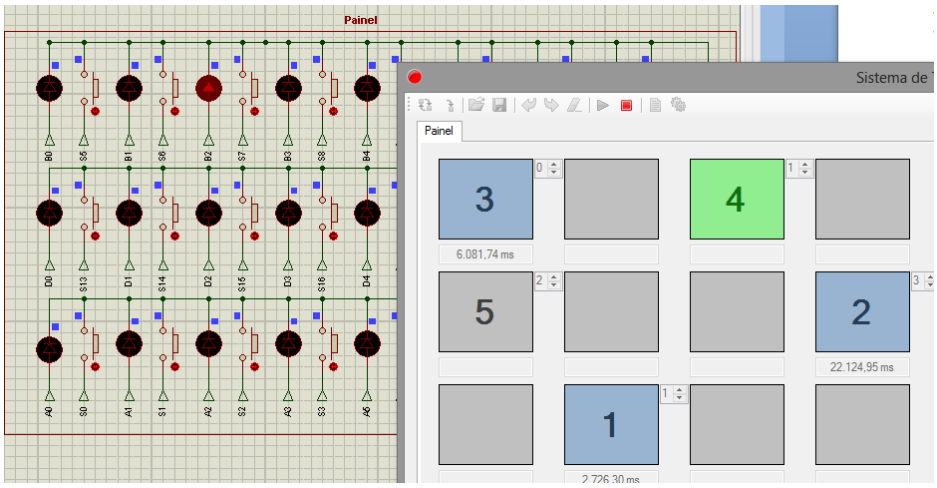

Figura 8 – Execução de comandos

```
==========================================
Atleta: PEDRO BUARQUE
Data: 20/05/2014   15:33
Sexo: Masculino
Idade: 24 anos
==========================================
Botão : Tempo de espera -> Tempo de resposta

18 : 1s -> 2.726,30 ms
12 : 3s -> 22.124,55 ms
1  : 0s -> 6.081,74 ms
3  : 1s -> 3.154,35 ms
9  : 2s -> 4.544,01 ms
==========================================
```

Figura 9 – Relatório de treinamento

6. CONSIDERAÇÕES FINAIS

Este trabalho propôs a construção de uma plataforma que pudesse ser usada para realizar a medição e treinamento do tempo de resposta de atletas, utilizando as funcionalidades do microcontrolador PIC18F4550. Foi realizado o projeto de software e construção dos circuitos de hardware. Os testes foram realizados utilizando ferramentas de simulação e foi possível observar um bom comportamento do sistema, tanto o *firmware* como a aplicação desenvolvida para realizar a interação entre o usuário e o sistema.

Houve uma grande dificuldade inicial para que a comunicação USB funcionasse corretamente. Foi necessário um estudo aprofundado no assunto para poder entender como funcionavam as funções específicas para a comunicação no *firmware*. Para que a comunicação USB do *software* de interação com o usuário funcionasse corretamente, foram testadas muitas bibliotecas disponibilizadas na internet até chegar à escolha ideal.

Com a geração de relatórios, disponibilizada no sistema, que possibilitam a avaliação do atleta durante cada treinamento, tornam esta aplicação muito útil para medir o tempo de

reflexo e treinamento de atletas. Em um contexto mais geral, este mesmo sistema poderá ser adaptado e utilizado como ponto de partida para outros projetos com o intuito de medir tempos de resposta não só de humanos, mas também o tempo entre eventos ou fenômenos que qualquer natureza.

7. REFERÊNCIAS

Livros:
BOLTON, William. Mecatrônica: uma abordagem multidisciplinar. 4. Ed. Porto Alegre: Bookman, 2010.

PEREIRA, Fábio. Microcontroladores PIC: Programação em C - São Paulo: Érica, 2003.

Internet:
Microchip Technology Inc. ***Data Sheet*** **do microcontrolador PIC18F4550**. Disponível em <http://ww1.microchip.com/downloads/en/DeviceDoc/39632b.pdf> Acesso em: 11 de maio de 2014

Proteus. Disponível em <http://www.labcenter.co.uk>. Acesso em: 20 de maio de 2014.

Monografias, dissertações e teses:
SANTOS, Leonardo. Sistema de comunicação USB com microcontrolador - Recife: Junho de 2009.

COBENGE 2014
ENGENHARIA: Múltiplos saberes e atuações
16 a 19 de setembro | Juiz de Fora - MG

RESPONSE TIME MEASUREMENT SYSTEM: AN APPLICATION TO THE MEASUREMENT OF REFLEX TIME AND ATHLETES TRAINING

Abstract: *Time response measurement system can be used on several areas of engineering. For example, the measurement of the response time of elderly persons might help on the cognitive state estimation as also as act as a stimulus to their skills improvement. Computational systems could be used to construct such system but the traditional personal computers input/output interfaces are not adequate for such applications once great dimensions might be required. To solve the problem of the multiple digital interfaces required to activate and measure the response time microcontrolled systems can be used. This paper presents the development of a microcontrolled system capable of to activate 24 different outputs and measure the time until the input be activated. The outputs are composed by a set of LEDs and the inputs are manually push buttons. The complete system will be used to the athletes training. The inputs/outputs set can be programmed to generate training sequences that demands the lateral shift, squatting and jumps to push the respective button. This type of training joints the reducing of the reaction time as also as the athletes skills. First tests will be executed with tennis modality that demands several of the movements previous cited. All work was developed during the course conclusion monograph. This paper demonstrates as the interactions between different areas, physical education and computing engineering, may be motivating to students under engineering graduation.*

Key-words: *Response time, microcontrolled systems, Athlete training*