

**Centro Universitário Positivo – UnicenP
Núcleo de Ciências Exatas e Tecnológicas – NCET
Engenharia da Computação
Ronnie Kindreich**

Pedômetro

**Curitiba
2005**

Centro Universitário Positivo – UnicenP
Núcleo de Ciências Exatas e Tecnológicas – NCET
Engenharia da Computação
Ronnie Kindreich

Pedômetro

Monografia apresentada à disciplina de Projeto Final, como requisito parcial à conclusão do Curso de Engenharia da Computação. Orientador: Professor José Carlos da Cunha.

Curitiba
2005

Termo de Aprovação

Ronnie Kindreich

Pedômetro – Sistema de avaliação de marcha

Monografia aprovada como requisito parcial à conclusão do curso de Engenharia da Computação (noturno) do Centro Universitário Positivo, pela seguinte banca examinadora:

Prof. José Carlos da Cunha

Prof. Valfredo Pilla Junior

Prof. Edson Pedro Ferlin

Curitiba, 07 de novembro de 2005

Agradecimentos

Gostaria de agradecer primeiramente a Deus que me deu vida para chegar até esse momento.

Quero agradecer também a meus pais, José Kindreich e Margarida Edwiges Kindreich, que me financiaram e me apoiaram nos momentos de dificuldade no decorrer desses anos de curso, ao mestre Maurício Perretto pelos conselhos preciosos e ao meu amigo e orientador professor José Carlos da Cunha.

Sumário

1. INTRODUÇÃO E MOTIVAÇÕES DO PROJETO.....	1
2.ESPECIFICAÇÃO TÉCNICA.....	3
2.1 Aspectos funcionais do sistema.....	3
2.2 Módulos do sistema.	5
3.REVISÃO BILIOGRÁFICA.....	7
3.1 Conhecimento teórico de hardware.....	7
3.1.1 Aspectos físicos.....	7
3.1.1.1 Pocição e deslocamento.....	7
3.1.1.2 Velocidade média e velocidade escalar média.....	8
3.1.1.3 Velocidade instantânea e velocidade escalar.....	9
3.1.1.4 Aceleração.....	10
3.1.2 Teoria de microcontroladores.....	11
3.1.2.1 Microcontroladores com núcleo 8051.....	11
3.1.2.2 Teoria básica de microcontroladores da família 8051.....	11
3.1.2.3 Microcontrolador MSC1211Y5.....	12
3.1.3 Porta Serial.....	14
3.1.3.1 Modos de comunicação serial RSR232.....	14
3.1.3.2 Modo síncrono de comunicação.....	14
3.1.3.3 Modo assíncrono de comunicação.....	15
3.1.3.4 Canais Simplex, Half-Duplex e Full-Duplex.....	15
3.1.3.5 O modo simplex.....	15
3.1.3.6 O modo Half-Duplex.....	15
3.1.3.7 O modo Full-Duplex.....	16
3.1.4 Interface serial no MSC1211Y5.....	16
3.2 Conhecimento teórico de software.....	16
3.2.1 As bases da orientação a objetos.....	16
3.2.1.1 Conceitos básicos de orientação a objetos.....	17
3.2.2 Conceito de UML.....	18
3.2.2.1 Diagramas da UML.....	19
3.2.2.1.1 Diagrama de classe.....	19
3.2.2.1.2 Diagrama de caso de uso.....	20
3.2.2.1.3 Diagrama de seqüência.....	20
3.2.3 Requisitos de sistemas.....	20
3.2.3.1 Especificação de requisitos.....	20
3.2.3.2 Engenharia de requisitos.....	20
4.ESPECIFICAÇÃO DO HARDWARE.....	22
4.1 Funções do hardware.....	22
4.2 Componentes utilizados.....	22
4.3 Diagrama em blocos do hardware do sistema (com descrições).....	23
4.3.1 Aquisição.....	23
4.3.2 Processamento.....	25
4.3.3 Armazenamento e visualização.....	26
4.3.4 Comunicação com o PC.....	28
4.4 Ambientes de desenvolvimento do Hardware.....	28
5.ESPECIFICAÇÃO DE SOFTWARE.....	29
5.1 Software embarcado no microcontrolador.....	29
5.1.1 Software embarcado no microcontrolador do hardware do tênis.....	30
5.1.2 Fluxograma do software embarcado.....	30
5.2 Software embarcado no hardware do monitor do atleta.....	33
5.2.1 Fluxograma com descrições.....	33
5.3 Software de interface gráfica para o usuário.....	34
5.3.1 Fluxograma com descrições.....	34
5.3.2 Levantamento de requisitos do usuário.....	36

5.3.2.1 Missão do produto.....	36
5.3.2.2 Lista de funções.....	37
5.3.3 Diagramas de casos de uso do sistema.....	38
5.3.3.1 Diagrama de seqüência dos casos de uso do cenário 1.....	39
5.3.3.2 Diagrama de seqüência dos casos de uso do cenário 2.....	40
5.3.3.3 Diagrama de seqüência dos casos de uso do cenário 3.....	42
5.3.3.4 Diagrama de seqüência dos casos de uso do cenário 4.....	43
5.3.3.5 Diagrama de seqüência: Atores -> Sistema.....	43
5.3.3.5.1 Contratos das operações do sistema.....	44
5.3.4 Diagrama de classes do sistema.....	45
5.3.4.1 Descrição das classes de interface, negócio e controle do sistema.....	46
5.3.4.1.1 Classes de interface com o usuário.....	46
5.3.4.1.2 Classe controladora de operações do sistema.....	47
5.3.4.1.3 Classe de negócio do sistema.....	48
5.4 Protótipo de telas do software de interface com o usuário.....	49
6. CRONOGRAMA DE DESENVOLVIMENTO DO PROJETO.....	51
7. ESTIMATIVA DE CUSTOS.....	52
7.1 Estimativa de custos do hardware do sistema.....	52
7.2 Estimativa de custos do software do sistema.....	52
8. ESPECIFICAÇÃO DE VALIDAÇÃO DO PROJETO.....	54
8.1 Especificação de validação do hardware do sistema.....	54
8.2 Especificação de validação dos softwares do sistema.....	54
8.2.1 Validação do software embarcado no microcontrolador do sistema.....	55
8.2.2 Validação do software de interface gráfica com o usuário.....	55
9. RESULTADOS.....	56
10. CONCLUSÃO.....	59
11. REFERÊNCIAS BIBLIOGRÁFICAS.....	60
12. ANEXOS.....	61

Lista de Figuras

Figura 1 – Visão geral do sistema.....	3
Figura 2 – Eixos de trabalho do acelerômetro.....	4
Figura 3 – Disposição do acelerômetro.....	4
Figura 4 – Visão geral do projeto.....	5
Figura 5 – A determinação da posição de um objeto.....	8
Figura 6 – Velocidade de acessos externos do MSC1211Y5 comparados com o 8051.....	13
Figura 7 – Diagrama de blocos do projeto de hardware do sistema.....	23
Figura 8 – Estrutura interna do acelerômetro ADXL202JE.....	23
Figura 9 – Saídas típicas (PWM) do ADXL202JE.....	24
Figura 10 – microcontrolador MSC1211Y5PAGT utilizado no projeto.....	25
Figura 11 – Atualização do sistema de três em três passos.....	26
Figura 12 – Memória auxiliar de descarga	26
Figura 13 – diagrama geral de blocos do hardware acoplado no tênis do atleta.....	27
Figura 14 – Diagrama geral de blocos do monitor do atleta.....	27
Figura 15 – Diagrama geral de blocos do hardware auxiliar de descarga de memória.....	28
Figura 16 – Fluxograma do software embarcado no controlador do hardware do tênis.....	31
Figura 17 – Fluxograma do microcontrolador do hardware do monitor do atleta.....	33
Figura 18 – fluxograma software de interface com o usuário.....	35
Figura 19 - Casos de uso identificados no Cenário 1.....	38
Figura 20 – Diagrama de seqüência dos casos de uso da figura 19.....	39
Figura 21 - Casos de uso identificados no Cenário 2.....	40
Figura 22 – Diagrama de seqüência dos casos de uso da figura 21.....	40
Figura 23 - Casos de uso identificados no Cenário 3.....	41
Figura 24 – Diagrama de seqüência dos casos de uso da figura 23.....	42
Figura 25 – Casos de uso identificados no cenário 4.....	42
Figura 26 – Diagrama de seqüência dos casos de uso da figura 25.....	43
Figura 27 – Diagrama de seqüência: Atores -> Sistema.....	43
Figura 28 – Diagrama de Classes do sistema.....	49
Figura 29 – Interface gráfica do software.....	50
Figura 30 – Informações do percurso em um ponto escolhido pelo usuário.....	50
Figura 31 - Gráfico de acelerações amostradas para uma bateria aleatória de teste - 4º Passo realizado.....	56
Figura 32 - Gráfico de acelerações amostradas para uma bateria aleatória de teste - 4º Passo realizado.....	56
Figura 33 – Gráfico de Acelerações válidas pelo hardware do sistema para uma passo aleatório de uma bateria de testes.....	57
Figura 34 – visão geral do protótipo do sistema desenvolvido.....	58

Lista de tabelas

Tabela 1 – Lista de funções do sistema.....	38
Tabela 2 – Casos de uso identificados no cenário 1.....	38
Tabela 3 – Casos de uso identificados no cenário 2.....	39
Tabela 4 – Casos de uso identificados no cenário 3.....	41
Tabela 5 – Casos de uso identificados no cenário 4.....	42
Tabela 6 – Descrição dos métodos da interface do sistema.....	47
Tabela 7 – Descrição dos métodos da classe de controle do sistema.....	47
Tabela 8 – Descrição dos métodos da classe de negócio do sistema.....	48
Tabela 9 – Cronograma de desenvolvimento do projeto.....	51
Tabela 10 – Estimativa e Custos para o desenvolvimento de uma unidade de hardware do projeto.....	52
Tabela 11 – Estimativa e Custos para o desenvolvimento dos softwares do sistema.....	52

Lista de Siglas

UNICENP – Centro Universitário Positivo

V - Volts

KHz - Kilo Hertz

MHz - Mega Hertz

A/D - Analógico/Digital

Freq. - Freqüência

Rx - módulo de recepção

Tx - módulo de transmissão

PC - Personal Computer (Computador Pessoal)

|v_m| - velocidade escalar média

Δx – deslocamento

Δt - intervalo de tempo

RF – Rádio freqüência

RESUMO

O presente trabalho descreve o desenvolvimento de um sistema para o monitoramento de atividades físicas, tanto de pessoas que realizam simples caminhadas, até atletas que realizam atividades físicas mais intensas, como corridas de nível profissional.

O sistema descrito apresenta uma solução utilizando acelerômetros para a obtenção de informações como aceleração, velocidade e distância desenvolvidas por um atleta durante a realização de alguma atividade física, sem a utilização de GPS (que torna os dispositivos de mercado caros) que possibilita referenciar o atleta durante a realização da atividade.

Após a realização de inúmeros testes com o protótipo desenvolvido, demarcando-se distâncias a serem percorridas e realizando comparações com sistemas encontrados no mercado, observamos boas respostas do sistema em termos de velocidade, tempo e distância percorrida.

ABSTRACT

This project presents the development of a system based on physical activities monitoring for professional level to home intended use public.

The system described presents a solution using accelerometers to obtain information about a measured distance, speed and other analysis during any kind of physical activity with out use of GPS (that turn the solutions more expensive) that would reference the user during the activities.

After many tests and evaluations with the developed prototype, fixing milestones and distances to be measured and comparing the results with other models available on the market, it was possible to observe good results out of the system. Good speed, time and distance values.

1. INTRODUÇÃO E MOTIVAÇÕES DO PROJETO

Muitas pessoas que realizam simples caminhadas diárias ou atletas que realizam atividades de corridas mais intensas, tem o desejo de monitorar seu desempenho durante a realização de algum percurso, saber qual a distância total que conseguiu percorrer, qual foi a sua velocidade durante o percurso, saber em que ponto do percurso ganhou ou perdeu desempenho para poder trabalhar em cima de seus pontos fortes e fracos, otimizando seus desempenhos futuros.

Os sistemas existentes no mercado, que possibilitam a aquisição desses tipos de informações (velocidade, aceleração distância percorrida) desejadas por essas pessoas, são muito precisos, mas são muito caros para serem adquiridos por pessoas que apenas realizam caminhadas diárias ou a maioria dos atletas amadores. Esses dispositivos existentes são caros por utilizarem tecnologias como o GPS para fazer a referência do atleta no percurso.

Existem ainda no mercado, os dispositivos que não utilizam GPS para fazer o referenciamento do atleta durante a realização da atividade física. Esses dispositivos são mais acessíveis e tem um melhor preço de mercado, mas tem o inconveniente de não terem uma boa precisão. Essa falta de precisão é facilmente verificada se levarmos em conta que, para utilizar esses dispositivos, precisamos fixar uma distância dos passos que efetuaremos durante a corrida, e esses passos não podem variar. Obviamente, quando caminhamos ou corremos, nossos passos ou passadas variam, daí a explicação para a falta de precisão encontrada nesses dispositivos.

Uma das motivações para esse projeto foi o desenvolvimento de um sistema com menor custo, que fosse acessível a todas as pessoas, desde simples praticantes de caminhadas diárias até atletas que realizam atividades de corridas mais intensas.

Além da motivação financeira descrita, a motivação técnica desse projeto foi o desenvolvimento do sistema utilizando acelerômetros para a aquisição de informações (como a aceleração, a velocidade a distância das passadas do atleta), sem a utilização de dispositivos de referência (GPS) como são a maioria dos sistemas encontrados no mercado.

Essa foi a dificuldade maior do projeto, utilizar conhecimentos matemáticos (equações diferenciais, derivadas e integrais) e físicos (aceleração e velocidade instantâneas), unidos a dispositivos de um custo mais baixo que os dispositivos

referenciais (como o GPS) para a obtenção das informações de monitoramento mais requisitadas por atletas corredores com uma precisão aceitável.

Com o sistema desenvolvido, os atletas poderão monitorar acelerações, velocidades e distâncias percorridas durante seus percursos de treinamento e corridas, possibilitando-os uma oportunidade de melhorar seus desempenhos futuros.

2. ESPECIFICAÇÃO TÉCNICA

Apartir desse capítulo da documentação, especificaremos o hardware e o software do sistema que foi desenvolvido, apresentando diagramas em blocos, fluxogramas e estudos de casos para os softwares que serão desenvolvidos para o PC e para o microcontrolador. Faremos também uma estimativa de custos e investimentos que serão necessários para o desenvolvimento de uma unidade desse projeto.

2.1 Aspectos funcionais do sistema

A figura 1 apresenta o sistema como um todo e a figura 2 apresenta as partes do mesmo que merecem destaque, que são os sensores utilizados para a captação dos sinais que possibilitarão o cálculo da aceleração, daí poderemos derivar e integrar essas informações de aceleração para calcular a velocidade e a distância dos passos do atleta durante a realização do percurso. Com essas informações, teremos condições de calcular a distância total que foi percorrida pelo atleta.

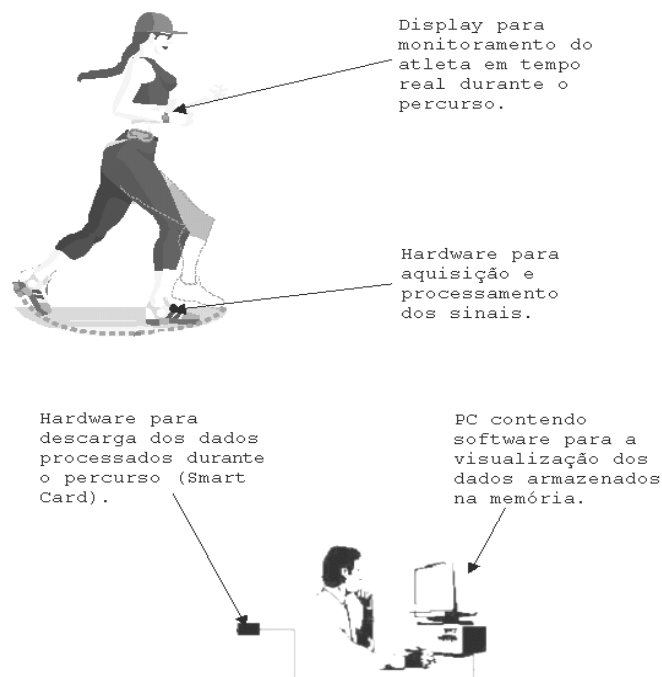


Figura 1 – Visão geral do Sistema



Figura 2 – Eixos de trabalho do acelerômetro (aquisição de sinais)

A parte do hardware do sistema acoplado no tênis do atleta (figura 1) é interligada ao display por RF para que os dados processados durante o percurso possam ser visualizados em tempo real no display de seu monitor.

Ao término de sua corrida ou caminhada, além de ter monitorado seu desempenho em tempo real, o atleta terá a possibilidade de descarregar a memória auxiliar, que armazenou todos os dados processados durante o percurso, em um PC que conterá um software que irá processar as informações da memória auxiliar, possibilitando ao atleta uma visualização global de seu desempenho no percurso realizado.

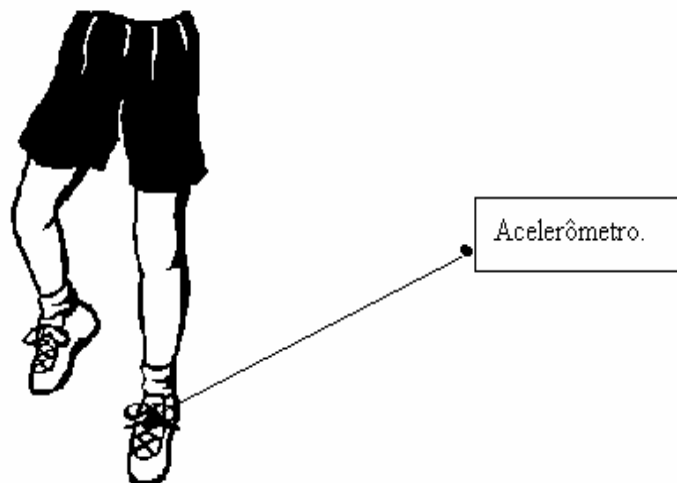


Figura 3 – Disposição do acelerômetro

Dentro da parte do hardware que irá ser acoplado no tênis do atleta (figura 3), teremos o dispositivo que é responsável por captar os sinais que indicam que o atleta deu um passo se estiver caminhando, ou passada se estiver correndo.

Inicialmente foi escolhido este local para o acoplamento desta parte do sistema por questões de ordem física do dispositivo, pois essa parte do hardware é muito grande, impossibilitando o seu acoplamento dentro do tênis do atleta.

A verificação de eficiência do sistema foi feita mediante testes comparativos, com dispositivos comerciais existentes que sejam capazes de medir e avaliar as grandezas que estão envolvidas nesse projeto, que são velocidade, aceleração, distância percorrida e distância das passadas do atleta, de forma a comprovar a eficiência do sistema desenvolvido.

2.2 Módulos do sistema

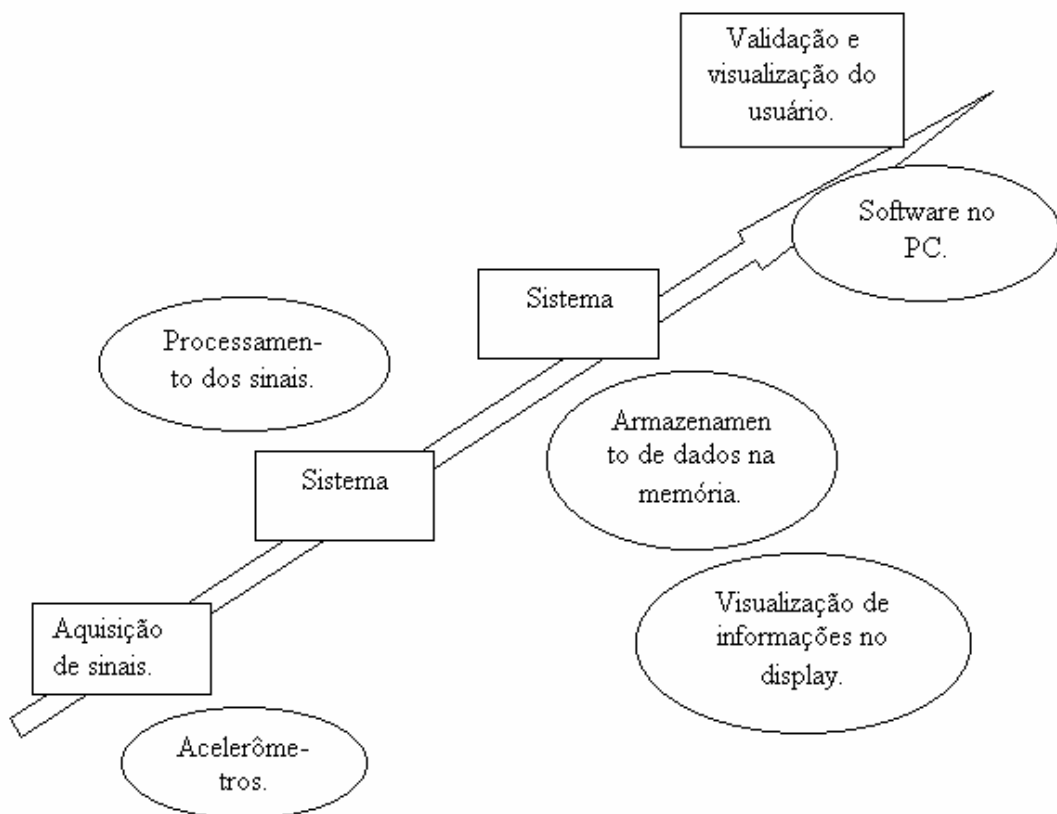


Figura 4 – Visão geral dos módulos do projeto

Dentre os módulos do sistema, o hardware é basicamente composto, na parte de aquisição (ver Figura 4), por um acelerômetro que faz aquisição de sinais referentes aos passos ou passadas realizadas pelo atleta durante o desenvolvimento do percurso.

Na parte de processamento de sinais (Figura 4), teremos um microcontrolador baseado na arquitetura 8051 que, sincronizado com o acelerômetro, processará e interpretará as informações recebidas desse sensor, contará o tempo que foi necessário para a realização do passo do atleta, realizando os processamentos necessários com essas informações, gravando-as na memória auxiliar de descarga e mostrando-as no display do atleta.

Conectado ao PC, teremos um pequeno dispositivo, ligado na porta serial ou paralela (ver figura 1), que terá a finalidade de enviar para o PC o conteúdo da memória auxiliar, que contém as informações que foram processadas e armazenadas durante o desenvolvimento do percurso pelo usuário do sistema. Essas informações receberão tratamento do software desenvolvido, plotando gráficos de desempenho no percurso realizado, mostrando velocidades em determinados pontos escolhidos, etc. Observa-se uma visão geral do sistema nas figuras 1 e 4.

Considerações:

1. Os sinais de saída gerados pelo acelerômetro poderão receber tratamento para a retirada de ruídos, conforme necessidades que poderão ser observadas na prática;
2. A memória auxiliar de descarga poderá ser mudada, dependendo de possíveis necessidades práticas do projeto;
3. Possivelmente o monitor do atleta também conterà um microcontrolador, que terá a função de fazer a recepção e mostrar as informações no display do atleta;
4. A memória auxiliar de descarga, a princípio estará na parte do hardware do sistema que se encontra no tênis do atleta, mas, dependendo de necessidades práticas (tamanho do produto desenvolvido) ela poderá ser acoplada no hardware do monitor do atleta, assim teremos mais uma função para o microcontrolador do módulo do monitor do sistema, que inicialmente só serve para receber e mostrar as informações processadas no display e o hardware do tênis ficará menor;
5. O acelerômetro escolhido para fazer a aquisição dos sinais também poderá ser trocado, dependendo de seu comportamento mediante a testes de ordem prática.

3. REVISÃO BIBLIOGRÁFICA

Esse capítulo apresentará o conhecimento teórico necessário para o desenvolvimento de todos os blocos de hardware que compõem esse projeto e também os conhecimentos necessários para o desenvolvimento do software orientado a objetos que foi desenvolvido para interface com o usuário.

3.1 Conhecimento teórico de Hardware

Esta seção do capítulo 3 apresentará os conhecimentos básicos necessários para o desenvolvimento do hardware do sistema.

3.1.1 Aspectos físicos

Quando começamos a pensar no desenvolvimento desse projeto, tentamos imaginar como poderíamos fazer para medir a distância dos passos de uma pessoa ou atleta que está realizando alguma atividade física, caminhada, cooper ou corridas sem um ponto de referência do atleta durante a realização de seu percurso.

Sem saber a distância das passadas do atleta durante a realização do percurso, não temos como saber a sua velocidade e aceleração instantâneas num determinado momento do percurso e por conseguinte, não sabemos a distância total percorrida por ele até o ponto em questão.

Para solucionar esses problemas, recorreremos a alguns conceitos físicos como velocidade, aceleração, posição, deslocamento e distância percorrida que ajudarão a compreender melhor as grandezas físicas envolvidas nesse projeto e também darão a base necessária para a geração de algumas soluções para esses problemas.

3.1.1.1 Posição e Deslocamento

- **Posição**

Basicamente, quando tentamos localizar um objeto, estamos tentando determinar sua posição relativa a um ponto de referência, em geral, a origem de um eixo [HALLIDAY, 1996]. A figura 5 ilustra essa idéia.

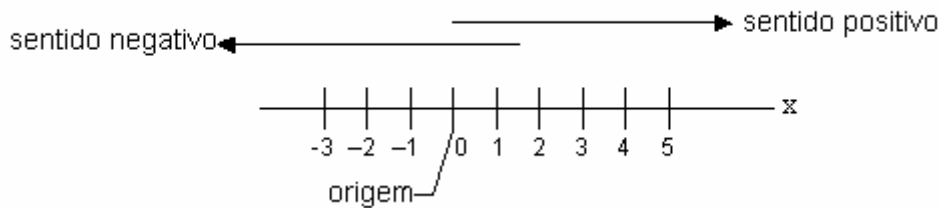


Figura 5 – A determinação da posição de um objeto

Na figura 5, o sentido positivo do eixo é crescente na escala numérica, ou seja, para a direita e o sentido negativo é o oposto [HALLIDAY, 1996].

- **Deslocamento**

A variação de uma posição x_1 para uma posição x_2 chama-se deslocamento [HALLIDAY, 1996].

Quando consideramos valores, um deslocamento no sentido positivo, (para a direita na figura 5), é um número positivo e no sentido contrário (para a esquerda na figura 5) é negativo [HALLIDAY, 1996].

A equação 1 mostra como podemos calcular o deslocamento de um objeto.

$$\Delta x = x_2 - x_1.$$

Equação 1 – Deslocamento de um objeto

3.1.1.2 Velocidade média e velocidade escalar média

- **Velocidade média**

A velocidade média é basicamente, a razão do deslocamento Δx (equação 1), ocorrido durante um determinado intervalo de tempo Δt (equação 2), por esse intervalo de tempo [HALLIDAY, 1996].

$$\Delta t = (t_2 - t_1)$$

Equação 2 – Intervalo de tempo

$$V = \Delta x / \Delta t = (x_2 - x_1) / (t_2 - t_1)$$

Equação 3 – Cálculo da velocidade média

- **Velocidade escalar média**

A velocidade escalar média $|v_m|$ de uma partícula depende basicamente, da distância total percorrida no intervalo de tempo Δt (equação 2) [HALLIDAY, 1996].

$$|v_m| = \text{distância total} / \Delta t$$

Equação 4 – Cálculo da velocidade escalar média

Enquanto a velocidade média (equação 3) é função do deslocamento Δx (equação 1) de uma partícula, a velocidade escalar média (equação 4) é função da distância total percorrida [HALLIDAY, 1996].

A velocidade escalar média difere também, da velocidade média, porque não considera o sentido de deslocamento, e por conseguinte não possui sinal algébrico [HALLIDAY, 1996].

3.1.1.3 Velocidade instantânea e Velocidade escalar

Apresentamos duas maneiras de descrever a velocidade com que algo se move: velocidade média (equação 3) e velocidade escalar média (equação 4), ambas medidas em relação a um intervalo de tempo Δt (equação 2).

Agora será conceituada a velocidade instantânea e a velocidade escalar.

- **Velocidade instantânea**

A velocidade instantânea em um ponto qualquer é igual a velocidade média (equação 3), quando o intervalo de tempo Δt (equação 2) tende a zero [HALLIDAY, 1996].

Basicamente, a medida que Δt (equação 2) diminui, a velocidade média (equação 3) tende a um valor limite, que é a velocidade naquele instante [HALLIDAY, 1996]. É

importante salientar que a velocidade instantânea (equação 5) é um vetor, tendo direção e sentido associados [HALLIDAY, 1996].

A velocidade instantânea é dada pela equação 5.

$$\mathbf{V} = \lim_{\Delta t \rightarrow 0} \Delta \mathbf{x} / \Delta t = d\mathbf{x} / dt.$$

Equação 5 – Cálculo da velocidade instantânea

- **Velocidade Escalar**

A velocidade escalar é basicamente, o módulo da velocidade instantânea (equação 5), ou seja, é a velocidade sem qualquer indicação de direção e sentido [HALLIDAY, 1996].

O velocímetro de um carro por exemplo, mede a velocidade escalar (equação 4), e não a velocidade (equação 5), porque ele não tem informações a cerca da direção e do sentido do movimento do veículo [HALLIDAY, 1996].

3.1.1.4 Aceleração

Quando a velocidade de uma partícula varia, dizemos que ela está sob uma aceleração (ou está acelerada) [HALLIDAY, 1996].

A aceleração média é dada pela equação 6.

$$\bar{a} = \frac{v_2 - v_1}{t_2 - t_1} = \frac{\Delta v}{\Delta t}$$

Equação 6 – Cálculo da aceleração

$$a = \frac{d(v)}{dt}$$

Equação 7 – Cálculo da aceleração instantânea

Existe também a aceleração instantânea ou simplesmente aceleração, que é a taxa de variação da velocidade num determinado instante [HALLIDAY, 1996].

A aceleração instantânea é dada pela equação 7.

3.1.2 Teoria de Microcontroladores

Microcontrolador é um dispositivo utilizado para controlar e monitorar funções durante um processo [PEREIRA, 2000].

A partir do advento dos circuitos integrados TTL, pode-se delinear três gerações no que diz respeito à implementação de controladores [DENYS, 2003].

Na primeira geração estão os projetos envolvendo circuitos integrados TTL, na sua maioria. O alto consumo de energia, a grande quantidade de chips envolvidos e a dificuldade em se realizar reengenharia tornou a segunda geração atraente aos projetistas.

O advento dos microprocessadores tornou versátil o projeto de circuitos destinados ao controle: é a segunda geração de controladores. Boa parte das funções, antes implementadas por hardware, passou a ser implementadas por software.

A terceira geração veio para integrar em um único chip boa parte dessa estrutura. Microcontroladores integram as funções de um microprocessador, memória de dados e de instruções e ainda, dependendo da complexidade, portas seriais e paralelas bidirecionais, conversores A/D, timers, watchdog e outros.

3.1.2.1 Microcontroladores com núcleo 8051

Para o desenvolvimento desse projeto, utilizaremos o microcontrolador MSC1211Y5 [Texas, 2004] que tem núcleo baseado na arquitetura 8051 [intel, 1980].

3.1.2.2 Teoria básica de microcontroladores da família 8051

A partir da década de 80, a família MCS-51 da Intel obteve grande sucesso, com microcontroladores de uso geral com capacidades de memória e I/Os diferenciados. A família MCS-51 pode incorporar memória de programa e dados internamente com a possibilidade de expansão de até 64K bytes de programa e

mais 64 Kbytes de dados. Permite o acesso a portas internas de I/O, canal de comunicação serial UART full duplex, interrupções com estrutura “nesting” com 5 fontes mascaráveis e dois níveis de prioridade, timers/counters de 16 bits, oscilador interno, frequência de clock típica de 12 MHz. A família MCS-51 permite facilidades de software que permitem a execução de complexas operações aritméticas e lógicas (multiplicação, divisão, permuta e deslocamento de bits, etc). Esta família trabalha com bancos de registradores nominais e também com bits endereçáveis na RAM.

Algumas vantagens apresentadas pelo microcontroladores com arquitetura 8051:

- Popular: prontamente disponível e amplo suporte. Gama completa de produtos de suporte estão disponíveis de graça e comercialmente.
- Eficaz: Instruções especializadas significam que menos bytes precisam ser buscados e menos jumps condicionais são processados.
- Baixo custo: alto nível de integração do sistema em um único componente. Poucos componentes são necessários para se criar um sistema que funcione.
- Ampla gama de produtos: uma única família de microcontroladores cobre as opções que outros fornecedores só conseguem cobrir com um número razoável de diferentes e incompatíveis famílias. Desse modo, o 8051 proporciona economia real em termos de custo de ferramentas, treinamento e suporte para software.
- Compatibilidade: opcodes e código binário são os mesmos para todas as variações do 8051, diferente de outras famílias de microcontroladores.
- Multi-Sourced: mais de 12 fabricantes, centenas de variedades.
- Aperfeiçoamentos constantes: melhorias na manufatura aumentam a velocidade e potência anualmente. Há ainda versões de 16 bits vindo de diversos fabricantes.

3.1.2.3 Microcontrolador MSC1211Y5 [Texas, 2004]

O microcontrolador MSC1211Y5 [Texas, 2004] é baseado na arquitetura 8051 [Intel, 1980] descrita na seção 3.2.1.1 desse capítulo, com algumas otimizações.

Além do microcontrolador, apresenta interno a seu ship, resumidamente, 32 Kb de memória flash que podem ser particionados de acordo com a aplicação em questão, conversores ADC e DAC, acumulador de 32 bits, duas portas de comunicação serial, watch dog Timer interno que só necessita ser programado,

alguns registradores a mais do que um 8051 normal [intel, 1980], interface de comunicação para barramento SPI entre outras vantagens.

Apresenta também a vantagem de podermos programar a velocidade de acessos a memória externa, oque significa que podemos sincronizar acessos externos desse microcontrolador com algum dispositivo que desejamos nos comunicar, memórias externas por exemplo.

A figura 6 mostra um comparativo de velocidades de acessos à memória externa que são efetuados eventualmente por esses microcontroladores.

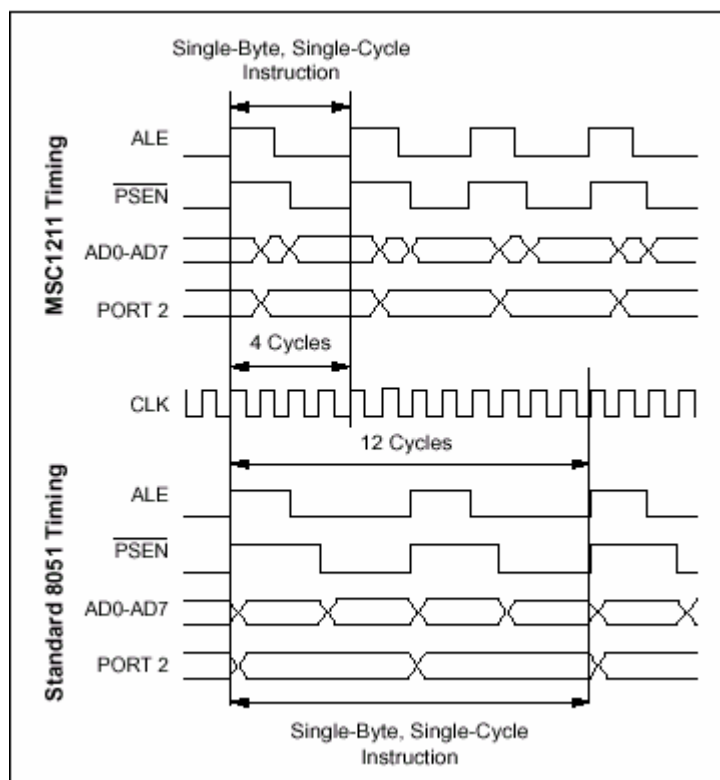


Figura 6 – Velocidade de acessos externos do MSC1211Y5 [Texas, 2004] em comparação com o 8051 padrão [Intel, 1980]

Cabe uma última observação importante e vantajosa desse microcontrolador, que é o fato de podermos habilitar ou não os componentes que são internos a esse ship (ADC, DAC, entre outros), diminuindo assim o consumo de energia do dispositivo.

3.1.3 Porta Serial

Diante da necessidade de comunicar equipamentos a grande distância, foi criada a transmissão serial [BJARNE, 2000].

Atualmente, o meio mais utilizado para o transporte serial de informação é a linha telefônica, privada ou pública, que com a ajuda de aparelhos dedicados permite a ligação de dois ou mais computadores, por exemplo, em países diferentes, bastando para tal a disponibilidade da linha telefônica e seus sistemas próprios (centrais, antenas e até mesmo satélites)

Na transmissão serial o envio de um certo caractere (vários bits) é feita de tal forma que, cada bit de cada caractere é transmitido de forma seqüencial, um após o outro.

Para que vários sistemas se comuniquem, foi criado um código binário para cada caractere, de tal forma que exista compatibilidade. Atualmente usa-se o código ASCII (American Standart Code for Interchange of Information), neste código, cada caractere possui seu corresponde em binário, incluindo-se aí vários caracteres de controle e sinais especiais.

3.1.3.1 Modos de comunicação serial RS232

Existem basicamente duas formas de comunicação serial RS232 que serão apresentadas a seguir: O modo síncrono de comunicação e o modo assíncrono de comunicação.

3.1.3.2 Modo Síncrono de Comunicação

Este modo necessita de um sincronismo entre dois sistemas em comunicação, este sincronismo é gerado por um conjunto de bits, denominado bits de sincronismo, que ao serem recebidos pelo elemento receptor, ajustam seu relógio (clock) interno para receberem um conjunto de bits referentes aos dados. Logo após o último bit de dado, o transmissor envia um conjunto de bits chamado bits de parada, que ao serem detectados pelo receptor informam que acabaram os bits de dados. Estes bits de parada podem conter ou não informações a respeito dos bits transmitidos, para permitirem ao receptor confirmar se recebeu os bits corretamente.

3.1.3.3 Modo Assíncrono de Comunicação

Neste modo não existe a necessidade de gerar sincronismo, cada caractere é transmitido individualmente, e para cada caractere (transmitido bit a bit) existe bits de início de transmissão (Start bit) e bits de fim de transmissão (Stop bit).

O Start bit é reconhecido pela transição do nível presente na linha de 1 para 0. Neste instante, o clock interno do sistema efetua uma varredura da linha de tempos em tempos para detectar o nível na mesma, nível que será associado a cada bit de forma conveniente. Ao reconhecer o sétimo bit, o sistema fica esperando o Stop bit, que é a transição de 0 para 1, ou a permanência em nível 1, se já estava em 1.

Neste ponto, o sistema entra em repouso e fica na espera de um novo Start bit, para iniciar a recepção de um novo caractere.

Neste modo de transmissão, deve-se garantir que o transmissor e o receptor operem com a mesma taxa de transmissão e recepção.

3.1.3.4 Canais Simplex, Half-Duplex e Full-Duplex

Existem basicamente, três maneiras de interligar dispositivos digitais que serão apresentadas nos tópicos a seguir: modo simplex, Half-Duplex e Full-Duplex.

3.1.3.5 O modo simplex

O modo simplex de interligação de dispositivos possui um elemento que apenas transmite e outro que apenas recebe. Exemplo: terminais de dados e impressoras.

3.1.3.6 O modo Half-Duplex

O modo Half-Duplex ou Semiduplex, permite elementos que recebem e transmitem dados, embora as duas operações não possam ocorrer simultaneamente.

3.1.3.7 O modo Full-Duplex

E o último modo é o Full-Duplex, onde os sistemas podem transmitir e receber dados simultaneamente.

3.1.4 Interface serial no MSC1211Y5

No MSC1211Y5 [Texas, 2004] com núcleo baseado na arquitetura 8051 padrão [Intel, 1980], a interface serial é do tipo Full-Duplex, isto significa que o microcontrolador pode receber e transmitir dados simultaneamente, sendo que para isto existem registros especiais. Este registro chama-se SBUF (Serial Buffer) e uma escrita no mesmo implica em automática transmissão do dado escrito, assim como um dado que chegue no pino de recepção, independente do controle do usuário (desde que o canal serial esteja habilitado e corretamente ajustado) (Pereira, 2000).

Existe na realidade, dois registros com o mesmo nome SBUF. Sendo um para recepção e outro para transmissão. O reconhecimento é feito pelo sistema através das instruções que acessarão o mesmo.

Cabe ressaltar que o MSC1211Y5 [Texas, 2004] apresenta duas interfaces de comunicação serial que podem ser utilizadas simultaneamente. As portas seriais do msc1211 [Texas, 2004] tem o mesmo funcionamento da porta serial de um microcontrolador 8051 padrão [intel, 1980].

3.2 Conhecimento teórico de software

Essa seção do capítulo apresentará o conhecimento teórico básico de orientação a objetos e modelagem UML necessário para o desenvolvimento do software de interface gráfica do sistema com o usuário do mesmo.

3.2.1 As bases da orientação a objetos

Diretamente derivada dos conceitos de programação e do projeto orientado a objetos, a análise e o desenho orientado a objetos são certamente a mais nova das abordagens de desenvolvimento de sistemas. A tecnologia de objetos apresenta componentes chaves que fundamentam a mudança de enfoque no processo de

modelagem e desenvolvimento de aplicações, trazendo benefícios intrínsecos à filosofia [FURLAN, 1998].

O sucesso em desenvolvimento de software depende em grande parte do conhecimento que não só envolve programação e habilidades de gerenciamento, mas também conhecimento e compreensão das mais recentes indústrias de software [FURLAN, 1998].

A tecnologia de objetos oferece modularidade de seus elementos podendo-se tomar um subconjunto existente e integrá-lo de uma maneira diferente em outra parte do sistema – uma aplicação no universo de objetos consiste de um conjunto de blocos de construção autocontidos e pré-definidos que podem ser localizados, reparados ou substituídos. A construção de código autocontido propicia o teste completo antes de ser utilizado dentro da lógica do sistema de informação [FURLAN, 1998].

3.2.1.1 Conceitos básicos de orientação a objetos

O primeiro conceito básico é o do próprio objeto. Um objeto é uma ocorrência específica (uma instância) de uma classe e é similar a uma entidade/tabela no modelo relacional somente até o ponto onde representa uma coleção de dados relacionados com um tema em comum [FURLAN, 1998].

Um objeto possui tudo que é necessário para conhecer a si próprio – há o encapsulamento e de operações e atributos atribuindo-lhe vida própria.

Em seguida temos o conceito de mensagem. Objetos se comunicam através de mensagens, isto é, um sinal enviado de um objeto a outro requisitando um serviço através da execução de uma operação [FURLAN, 1998].

Adjacente ao conceito de mensagem, temos o conceito de polimorfismo, cuja palavra é originária do grego "muitas formas". Tais formas se referem a vários comportamentos que uma mesma operação pode assumir, assim como a capacidade de uma variável referir-se a objetos diferentes que preenchem responsabilidades dependendo da mensagem que lhes é passada [FURLAN, 1998].

Outro conceito importante é o de classe. A classe é uma coleção de objetos que podem ser escritos com os mesmos atributos e as mesmas operações. Representa uma idéia ou um conceito simples e categoriza objetos que possuem propriedades

similares, configurando-se em um modelo para a criação de novas instâncias [FURLAN, 1998].

Na criação de classes, há a possibilidade de ocorrer uma conexão semântica de elementos do modelo entre pai e filho na qual uma classe filha (subclasse) herda as propriedades de seu pai (superclasse) direta ou indiretamente. Cada classe pode ter suas propriedades particulares herdadas diretamente da classe pai ou substituídas/mascaradas nessa transição, assim somente propriedades diferentes serão declaradas na classe filha. Esse mecanismo que acabou de ser descrito é chamado de herança [FURLAN, 1998].

Adicionalmente, há o conceito de dependência e coesão entre classes.

Dependência refere-se ao conceito que uma classe possui de outra – o objetivo é o de minimizar a dependência para evitar impactos em uma classe decorrentes de modificações em outra classe. Coesão é uma medida de integridade conceitual de uma classe – o objetivo, nesse caso, é o de maximizar a coesão para assegurar agrupamento de operações e com isso reduzir esforços de manutenção [FURLAN, 1998].

3.2.2 Conceito de UML

A UML é a linguagem padrão para especificar, visualizar, documentar e construir artefatos de um sistema e pode ser utilizada com todos os processos ao longo do ciclo de desenvolvimento através de diferentes tecnologias de implementação [FURLAN, 1998].

A UML é um passo natural na escala de evolução de objetos com o objetivo de:

1. Fornecer aos usuários uma linguagem visual expressiva e pronta para uso visando o desenvolvimento de modelos de negócio;
2. Fornecer mecanismos de extensibilidade e de especialização para apoiar conceitos essenciais;
3. Ser independente de linguagens de programação e processos de desenvolvimento
4. Prover uma base formal para entender a linguagem de modelagem;
5. Encorajar o crescimento no número de ferramentas orientadas a objeto no mercado;

6. Suportar conceitos de desenvolvimento de nível mais elevado tais como colaboração, estrutura de trabalho, padrões e componentes;

7. Integrar as melhores práticas;

3.2.2.1 Diagramas da UML

O modo de descrever os vários tipos de modelagem pela UML é através da notação definida pelos seus vários tipos de diagramas. Um diagrama é uma apresentação gráfica de uma coleção de elementos de modelo, freqüentemente mostrado como um gráfico conectado de arcos (relacionamentos) e vértices (outros elementos do modelo) [FURLAN, 1998].

Existem vários diagramas propostos pela UML. Serão apresentados apenas os modelos que serão implementados dentro do escopo desse projeto.

3.2.2.1.1 Diagrama de classe

Gráfico bidimensional de elementos de modelagem que denota a estrutura estática de um sistema e as classes representam coisas que são manipuladas por esse sistema. O diagrama de classes é considerado estático porque a estrutura descrita é sempre válida em qualquer ponto no ciclo de vida do sistema [FURLAN, 1998].

Existem quatro tipos principais de relacionamentos no diagrama de classe [FURLAN, 1998]:

1. Generalização/especialização: Indica basicamente, o relacionamento entre um elemento mais geral e um elemento mais específico.
2. Agregação: Usado para denotar relacionamentos todo/parte (por exemplo um item de compra é parte de um pedido).
3. Associação: Utilizado para denotar relacionamentos entre classes não correlatas (por exemplo, um cliente pode alugar várias fitas de vídeo).
4. Dependência: É um relacionamento entre elementos, um independente e outro dependente, onde uma mudança no elemento independente afetará o elemento dependente.

3.2.2.1.2 Diagrama de caso de uso

Os casos de uso descrevem a funcionalidade do sistema percebida por atores externos. Um ator interage com o sistema podendo ser um usuário, dispositivo ou outro sistema [FURLAN, 1998].

3.2.2.1.3 Diagrama de seqüência

Apresenta a interação de seqüência de tempo dos objetos que participam da interação. As duas dimensões de um diagrama de seqüência consistem basicamente, na dimensão vertical (tempo) e na dimensão horizontal (objetos diferentes) [FURLAN, 1998].

3.2.3 Requisitos de sistemas

Os requisitos são as características que definem os critérios de aceitação de um produto [PRESSMAN, 1995].

3.2.3.1 Especificação de requisitos

Os requisitos podem ser dos seguintes tipos: explícitos, implícitos e normativos.

- Os requisitos explícitos são aqueles escritos em um documento que arrola os requisitos de um produto, ou seja, uma especificação de requisitos;
- Os requisitos normativos são aqueles que decorrem de leis, regulamentos, padrões e outros tipos de a que o tipo de produto devem obedecer;
- Os requisitos implícitos são expectativas dos clientes e usuários, que são cobradas por estes, embora não documentadas.

3.2.3.2 Engenharia de requisitos

Ao conjunto de técnicas de levantamento, documentação e análise forma a engenharia de requisitos [PRESSMAN, 1995].

Um dos problemas básicos da engenharia de software é o levantamento e documentação dos requisitos dos produtos de software [PRESSMAN, 1995].

Quando este levantamento é bem feito, os requisitos implícitos são minimizados. Quando a documentação é bem feita, os requisitos documentados tem maiores chances de serem corretamente entendidos pelos desenvolvedores.

A boa engenharia de requisitos faz com que consigamos entender melhor os problemas do cliente, diminuimos também a possibilidade de resolvermos o problema errado, algum problema que não era o que o cliente queria que fosse resolvido. É uma situação absurda querer resolver um problema sem seu perfeito entendimento [PRESSMAN, 1995].

4. ESPECIFICAÇÃO DO HARDWARE

Esse capítulo apresenta o desenvolvimento de todos os módulos de hardware que compõem o sistema, suas funcionalidades e componentes que foram utilizados em cada módulo.

4.1 Funções do hardware

O hardware tem funções totalmente autônomas em relação ao software desenvolvido, ou seja, não depende de nenhuma configuração por parte do usuário.

O hardware tem a função de fazer a aquisição dos sinais do acelerômetro. O microcontrolador deve processar esses sinais, interpretando-os utilizando-se de recursos físicos, como aceleração e velocidade instantâneas, e matemáticos, como funções diferenciais, derivadas e integrais, gerando as informações que serão necessárias (aceleração, velocidade e distância das passadas do atleta), gravando-as na memória auxiliar de descarga do sistema e enviando-as para o monitor do atleta (por RF) para poderem ser mostradas no display.

Também foi desenvolvido um hardware auxiliar ligado na porta serial do PC. Esse hardware tem a função de enviar para o PC o conteúdo da memória auxiliar que contém as informações do atleta referentes ao percurso realizado.

4.2 Componentes utilizados

- Acelerômetro ADXL202JE [Analog, 2003];
- Microcontrolador MSC1211Y5pagt [Texas, 2004];
- Latch 74LS373;
- Display LCD;
- Transmissores RF DP1201 [XEMICS, 2004];
- Capacitores e resistores;
- Memória SRAM BQ4016 com 1M byte de capacidade de armazenamento;
- Decodificadores 74LS138.

Poderão ser utilizados conforme necessidades práticas do projeto:

- Amplificadores de instrumentação INA 128 e 129;
- Filtros ativos passa-baixas (TLC04).

4.3 Diagrama em blocos do hardware do sistema (com descrições)

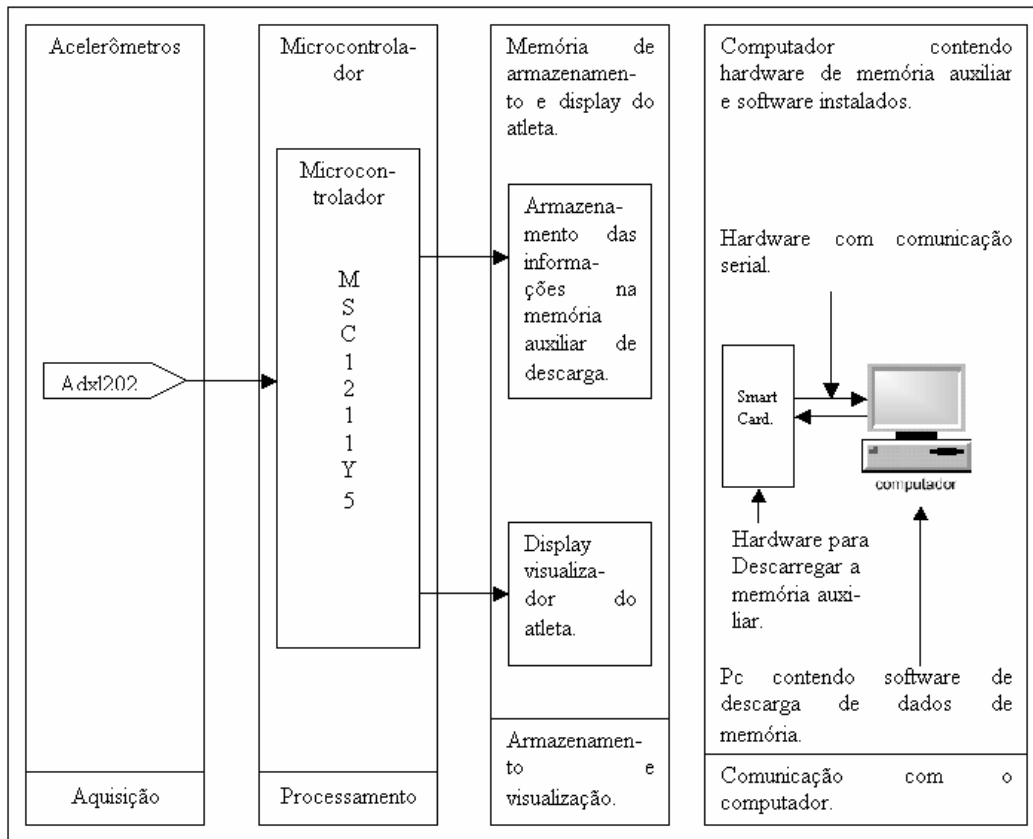


Figura 7 – Diagrama de blocos do projeto de hardware do sistema

4.3.1 Aquisição

A aquisição do sinal (figura 7) é feita utilizando-se o acelerômetro ADXL202JE [analog, 2003] acoplado no tênis do atleta, como ilustrado na Figura 3.

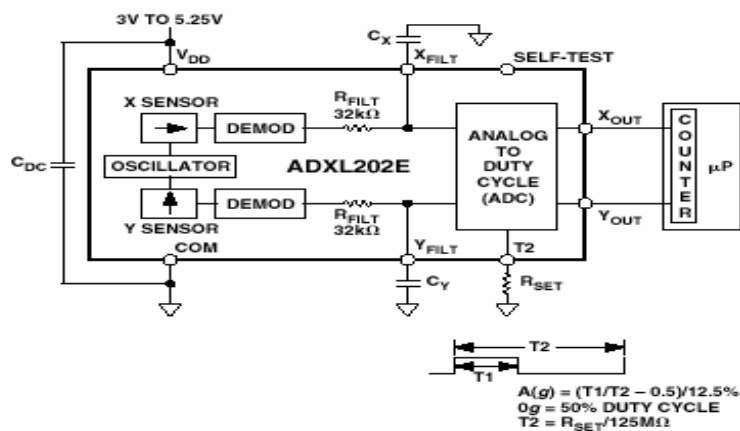


Figura 8 – Estrutura interna do acelerômetro ADXL202JE

O acelerômetro (figura 8) foi utilizado para medirmos a distância das passadas do atleta durante o percurso. A figura 9 mostra as saídas PWM típicas dos eixos x e y que são disponíveis no acelerômetro ADXL202JE [analog, 2003].

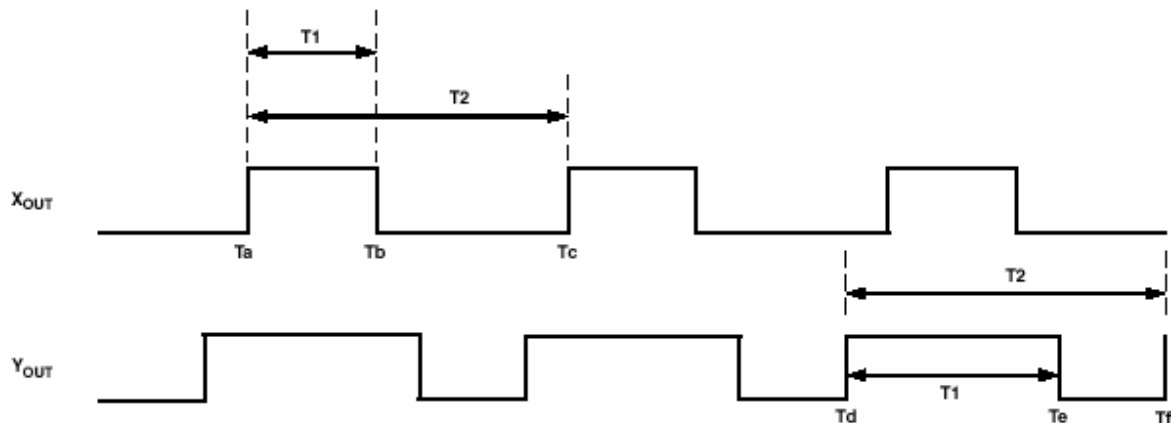


Figura 9 – Saídas típicas (PWM) do ADXL202JE

Da folha de dados do fabricante do ADXL202JE [analog, 2003], descobrimos como medir a aceleração, interpretando a saída PWM desse componente:

$$a = (T1/T2 - 0.5) / 12.5\%$$

Equação 8 – Cálculo da aceleração utilizando a saída PWM do ADXL202JE.

Para calcular a aceleração usando o ADXL202JE [analog, 2003], primeiramente teremos que definir o tempo de Duty Cycle (T2 na figura 9) das saídas PWM desse componente.

Além de definir o tempo de Duty Cycle do acelerômetro, teremos que calibrar a banda de passagem (bandwidth) do sinal, a freqüência do sinal, levando em conta o número máximo de passadas que um atleta de nível profissional consegue realizar em um determinado tempo (um segundo, por exemplo).

A princípio, para efeitos de testes práticos, a freqüência do sinal foi calibrada para 10Hz. Cabe ressaltar que todos esses ajustes são realizados no próprio acelerômetro, que apresenta pinos específicos para ajuste de bandas de passagem, freqüência de sinais e definições de tempos de Duty Cycle.

O hardware que é acoplado no tênis do atleta (figura 1 e figura 11) poderá estar tanto no pé esquerdo quanto no direito, onde o atleta preferir.

4.3.2 Processamento

Todo processamento é realizado pelo microcontrolador de 8 bits MSC1211Y5 [Texas, 2004], que é baseado na arquitetura de microcontroladores da família 8031/51 [Intel, 1980]. Esse microcontrolador é ilustrado na figura 10.

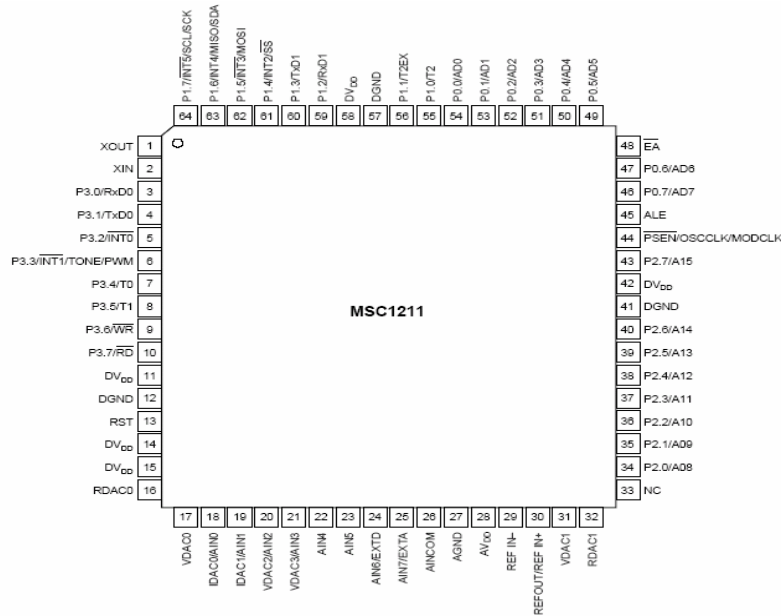


Figura 10 – microcontrolador MSC1211Y5 [Texas, 2004] utilizado no projeto

O MSC1211Y5 [Texas, 2004] estará sincronizado com o acelerômetro decodificando a saída PWM do canal x do mesmo.

A princípio, foi definido que o microcontrolador contaria os pulsos de saída PWM do acelerômetro de três em três passos. A razão disso é que temos apenas um acelerômetro acoplado em um dos pés do atleta (ver figura 3), que é o responsável por fornecer a aceleração no eixo x do atleta (o canal x de saída PWM do acelerômetro, a saída PWM y do acelerômetro é descartada) e ainda temos que utilizar suas saídas analógicas x e y, para conseguirmos detectar e informar ao microcontrolador que um passo foi efetuado pelo atleta.

Realizando testes práticos com o ADXL202JE, definimos um limiar para as suas saídas analógicas. Quando esse limiar é atingido, sabemos que um passo foi efetuado pelo atleta.

Quando o sinal DC das saídas analógicas do acelerômetro (eixo x e y) atingir o limiar definido para indicar um passo, um circuito comparador de nível sinaliza para alguma interrupção de prioridade alta do microcontrolador que o passo foi finalizado.

Se tentarmos imaginar o atleta correndo ou caminhando com o dispositivo no pé direito, como ilustrado na figura 11 por exemplo, ele fará o primeiro passo com o pé direito, o segundo passo não é marcado pelo sistema (o acelerômetro está no pé direito com o restante do sistema, portanto pisadas com o pé esquerdo nesse caso não são captadas pelo sistema) e o terceiro passo novamente com o pé direito, indicará que três passos foram efetuados pelo atleta.

Quando o microcontrolador tiver recebido dois avisos do circuito comparador de nível ele saberá que três passos foram realizados, e então atualizará o display do atleta com as informações de aceleração, velocidade, distância percorrida que foram calculadas e processadas pelo microcontrolador durante a realização dos três passos e gravará as essas informações na memória auxiliar de descarga.

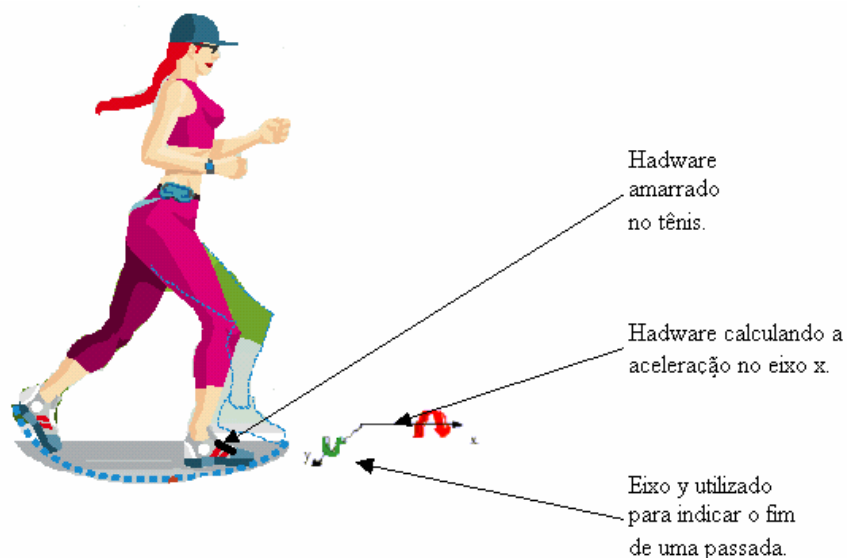


Figura 11 – Atualização do sistema de três em três passos

4.3.3 Armazenamento e visualização

A memória de descarga contém as informações coletadas durante a corrida, possibilitando ao atleta a visualização de dados em um computador pessoal.

A figura 12 ilustra a memória utilizada como memória auxiliar de descarga de dados do atleta.



Figura 12 – Memória auxiliar de descarga

O diagrama de blocos da figura 13 mostra com maiores detalhes todos os blocos de componentes que são necessários para o desenvolvimento dessa parte do projeto.

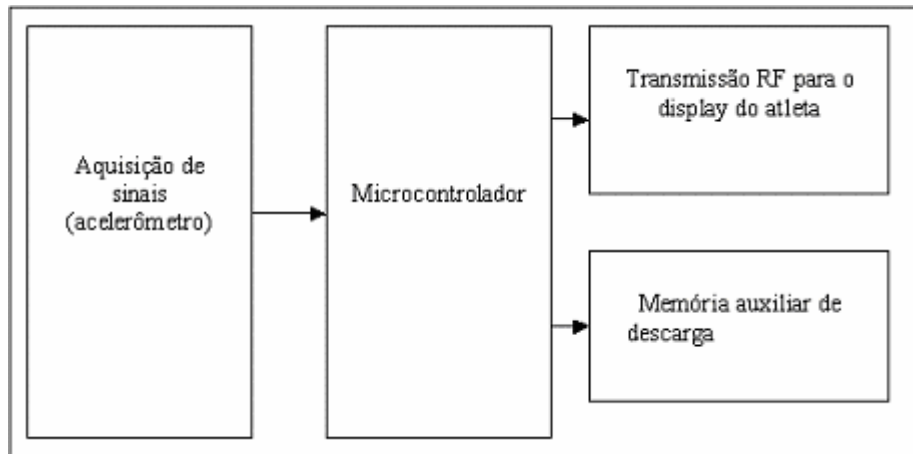


Figura 13 – diagrama geral de blocos do hardware acoplado no tênis do atleta

A visualização das informações processadas (aceleração, velocidade e distância dos passos do atleta) é feita no display do monitor do atleta em tempo real, durante a realização de seu percurso. O diagrama de blocos da figura 14 ilustra com maiores detalhes os componentes necessários para o desenvolvimento dessa parte do hardware.

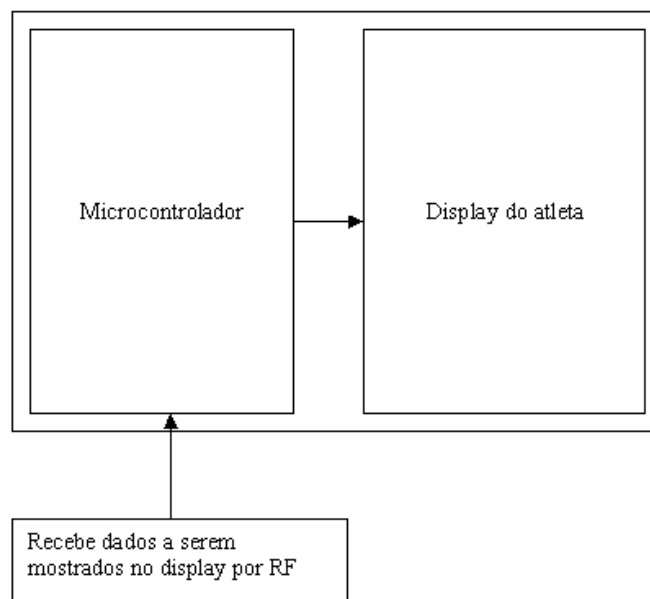


Figura 14 – Diagrama geral de blocos do monitor do atleta

4.3.4 Comunicação com o PC

Esse bloco foi o último a ser desenvolvido no projeto. Teremos um pequeno hardware que realizará a descarga da memória do hardware do atleta no computador. Esse dispositivo estará acoplado ao PC pela porta serial ou paralela.

O software a ser desenvolvido fará a recepção dos dados da memória, possibilitando a plotagem de gráficos e visualização de valores em determinados pontos, como velocidade instantânea, aceleração, distância percorrida, etc.

O diagrama geral de blocos da figura 15 mostra a idéia geral que foi desenvolvida nessa parte do hardware do sistema.

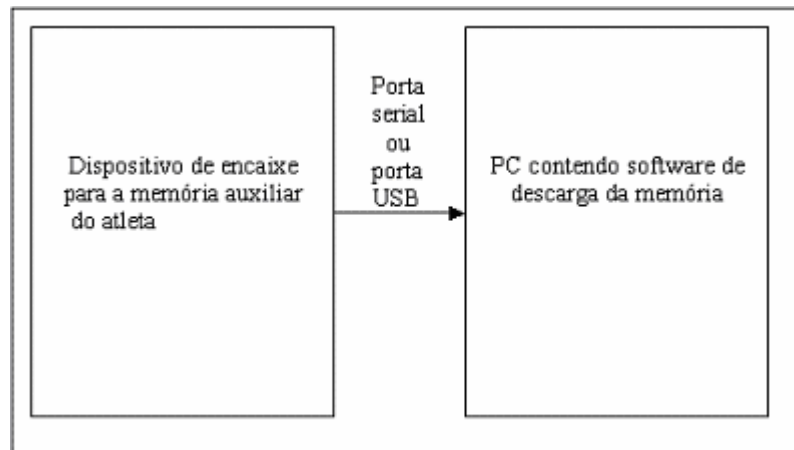


Figura 15 – Diagrama geral de blocos do hardware auxiliar de descarga de memória

4.4 Ambientes de desenvolvimento do Hardware

O ambiente utilizado para programar o microcontrolador MSC1211Y5 [Texas, 2004] foi o RIDE do fabricante Raisonance.

Para o desenho do diagrama esquemático do circuito foi utilizado o software Orcad Capture e para o desenvolvimento da placa de circuito impresso foi utilizado o software Orcad Layout.

5. ESPECIFICAÇÃO DE SOFTWARE

Esse tópico do documento realiza um estudo completo do funcionamento dos softwares embarcados nos microcontroladores MSC1211Y5 [Texas, 2004] que foram utilizados para o desenvolvimento desse projeto e também o software de interface gráfica para o usuário.

5.1 Software embarcado no microcontrolador

O microcontrolador MSC1211Y5 [Texas, 2004] tem a flexibilidade de ser programado tanto na linguagem assembly, baseado na arquitetura do microcontrolador 8051, quanto na linguagem C [Texas, 2004];

A linguagem assembly não é muito indicada para o desenvolvimento de aplicações que envolvam contagens de tempo grandes, pois os registradores do 8051 são de apenas 8 bits, o que dificulta a contagem de tempos relativamente longos [DENYS, 2003].

A linguagem C é mais indicada para o desenvolvimento desse tipo de aplicações que envolvem contagem de tempo, pois quando declaramos uma variável no programa, podemos escolher o tipo de dado que ela irá comportar e o compilador agrupará os registradores de forma a suportar o tamanho dos dados que foram declarados [Mizrahi, 1993]. Por exemplo, se declarmos:

```
long int var;
```

O compilador irá agrupar internamente na hora da compilação do programa os registradores do microcontrolador de forma a suportar dados para esse tamanho que foi declarado.

Se utilizarmos a linguagem assembly, não teremos essa facilidade de trabalho com as variáveis que temos quando programamos o microcontrolador na linguagem C, por esse motivo específico, os softwares dos microcontroladores do hardware do ténis e do monitor serão desenvolvidos em linguagem C.

5.1.1 Software embarcado no microcontrolador do hardware do tênis

O software embarcado no microcontrolador do hardware do tênis do atleta (ver figuras 1 e 11) que foi construído para calcular, interpretar e processar as informações de aceleração disponibilizadas nas saídas PWM do canal x do acelerômetro ADXL202JE [Analog, 2003] foi desenvolvido em linguagem C.

5.1.2 Fluxograma do software embarcado

O fluxograma básico do software embarcado no microcontrolador do hardware do tênis do atleta, que foi implementado para fazer a aquisição dos sinais de aceleração do ADXL202JE [Analog, 2003], interpretando-os e processando-os durante a realização do percurso pelo atleta é apresentado na figura 16.

Observando o fluxograma da figura 16, vemos que o programa é basicamente um laço infinito que conta o tempo T1 (ver figura 9) para o cálculo da aceleração instantânea (equação 7).

A idéia geral do software do microcontrolador do hardware do tênis é fazer o maior número possível de aquisições de sinais do ADXL202JE [Analog, 2003] interpretando-os e tirando a aceleração média do atleta a cada três passos que são realizados pelo mesmo. Após o atleta ter efetuado três passos, o hardware atualiza o seu display com as novas informações de distância percorrida, aceleração e velocidade médias e grava as informações na memória auxiliar do atleta.

Observado ainda o fluxograma da figura 16, observamos a utilização do watch dog timer e da interrupção 2 do microcontrolador MSC1211Y5 [Texas, 2004].

Como o programa está em laço infinito, calculando o tempo T1 (figura 9), a única forma de irmos para o final da execução do software é se ocorrer algum erro durante a utilização do sistema, assim, o programa ativará o watch dog timer, emitirá uma mensagem para o atleta e encerrará a sua execução.

A interrupção 2 do microcontrolador está ligada ao circuito comparador de nível que indicará ao controlador que um passo foi realizado pelo atleta, e assim, após ter dados três passos, o sistema atualizará seu display e gravará as informações na memória auxiliar do atleta.

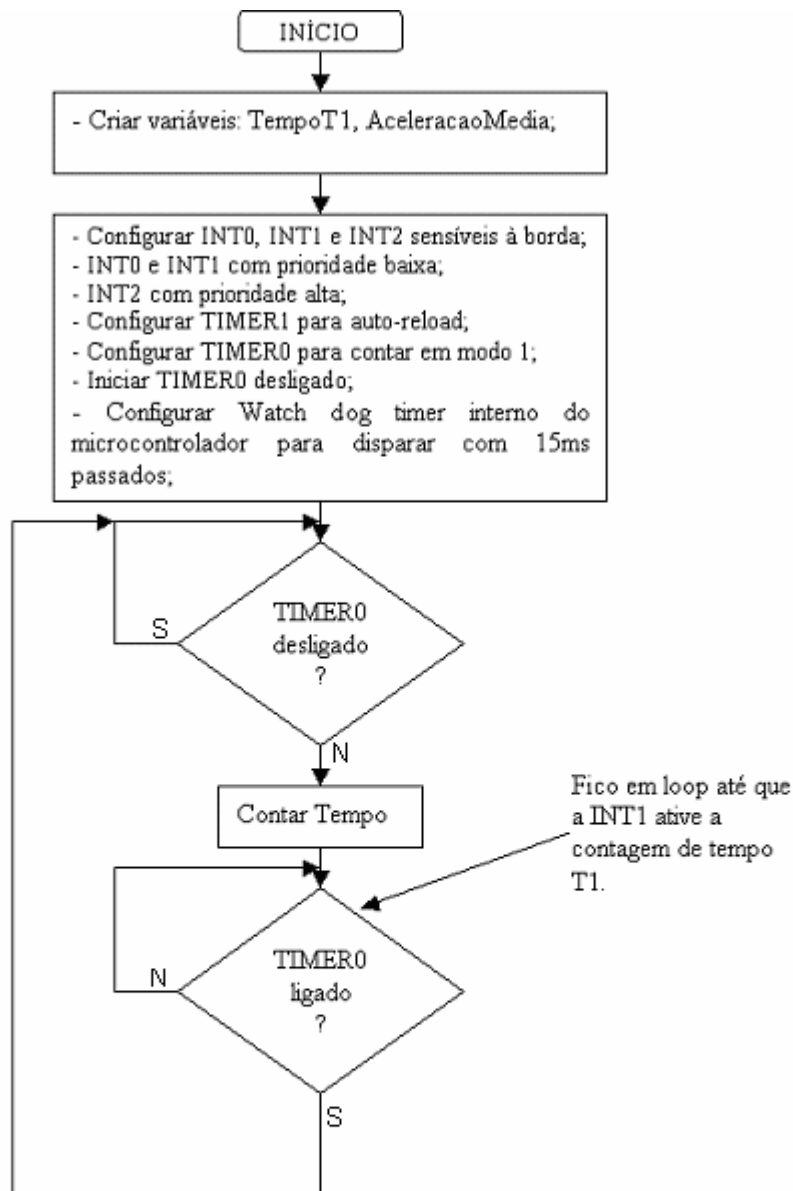


Figura 16 – Fluxograma do software embarcado no controlador do hardware do tênis do atleta

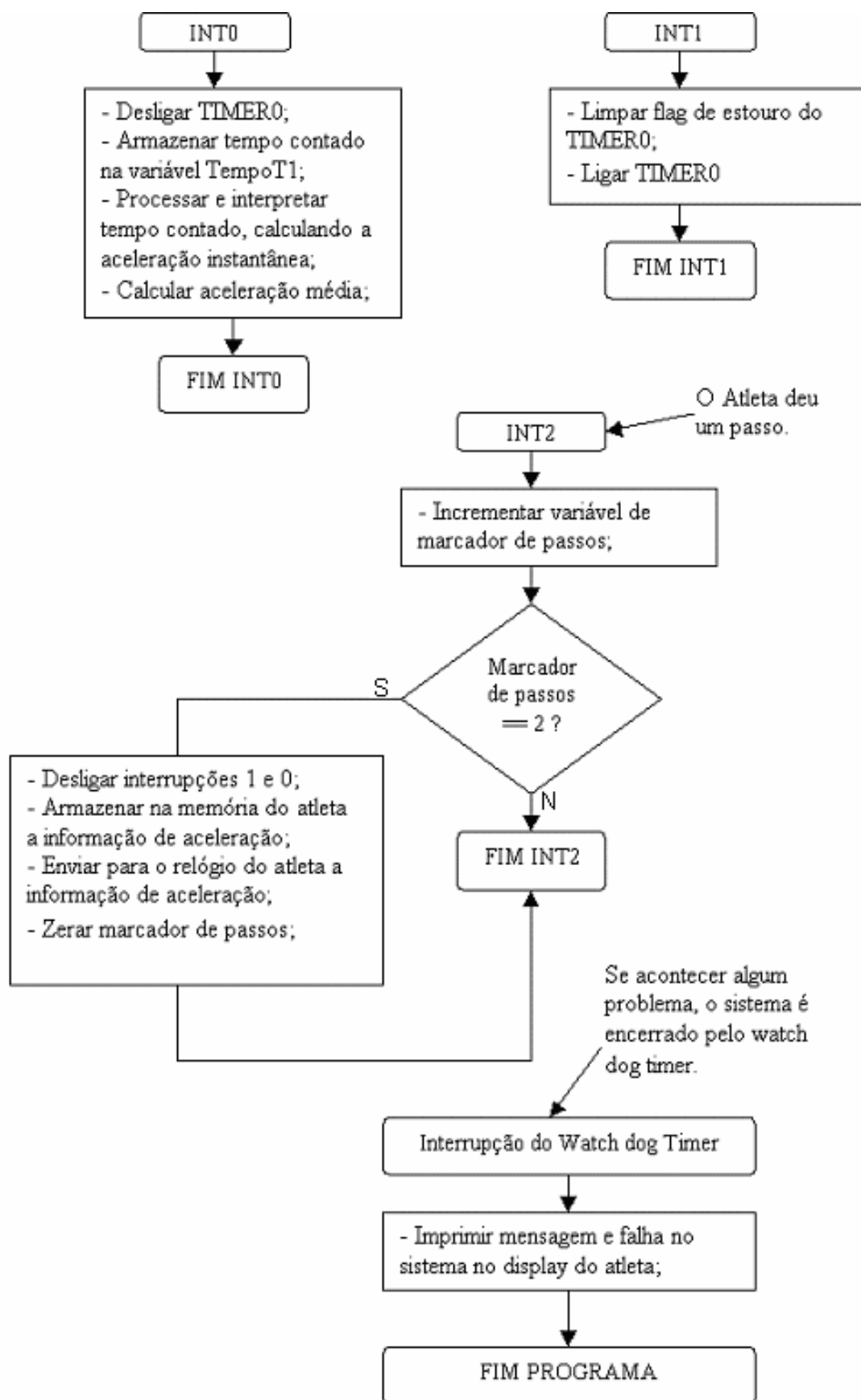


Figura 16 (continuação) – Fluxograma do software embarcado no controlador do hardware do tênis do atleta

5.2 Software embarcado no hardware do monitor do atleta

O software embarcado no microcontrolador do monitor do atleta foi construído utilizando-se a linguagem C.

5.2.1 Fluxograma com descrições

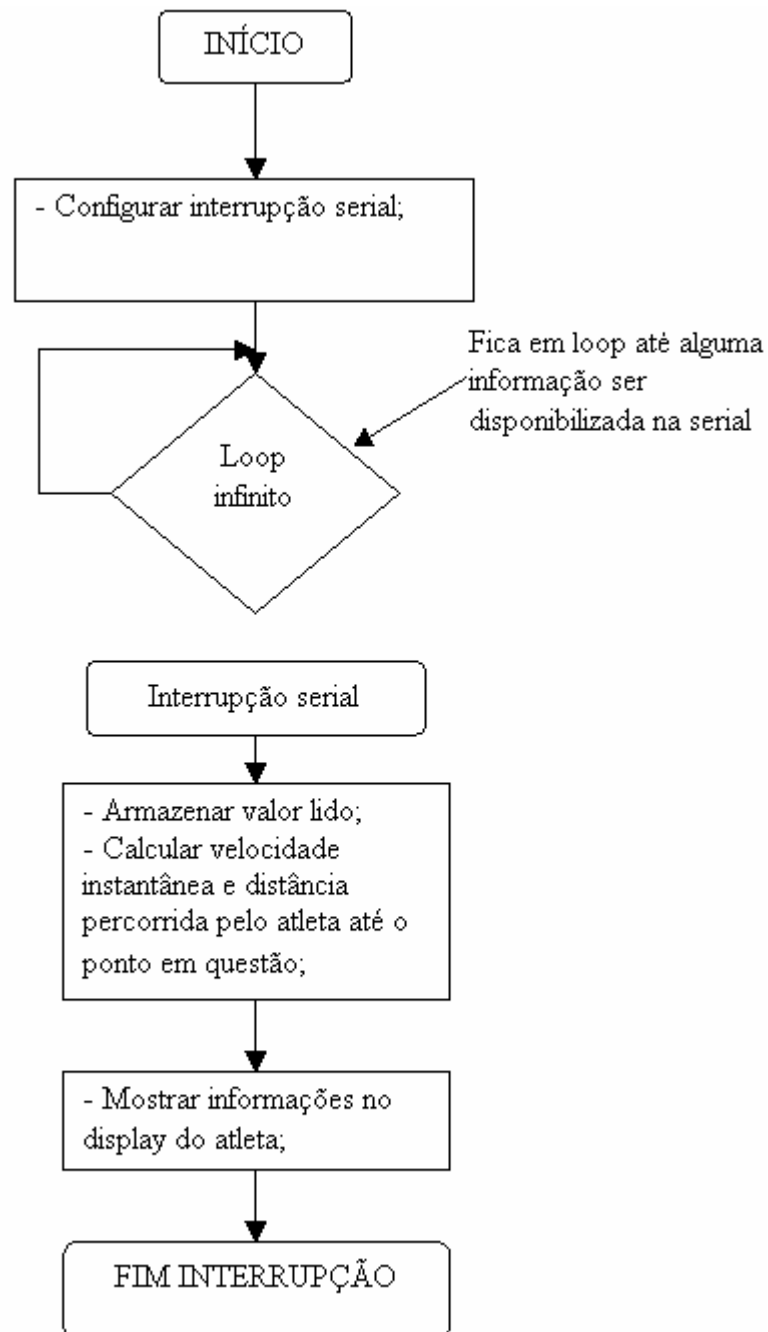


Figura 17 – Fluxograma do microcontrolador do hardware do monitor do atleta

O fluxograma básico do software embarcado no microcontrolador do hardware do monitor do atleta, que é responsável por receber as informações de aceleração que foram processadas pelo hardware do tênis do atleta é apresentado na figura 17.

Observando o fluxograma da figura 17 vemos que, esse software simplesmente fica em loop infinito, recebendo via RF as informações de aceleração que foram processadas pelo hardware do tênis do atleta e, com essa informação, deverá calcular a velocidade instantânea do atleta e a distância total percorrida até o ponto em questão. Feito os cálculos, o programa atualizará o display do atleta com as novas informações.

5.3 Software de interface gráfica para o usuário

O software de interface gráfica para o usuário, foi desenvolvido utilizando-se conceitos de orientação a objetos com a linguagem de programação C++.

5.3.1 Fluxograma com descrições

O fluxograma básico do software de interface para o usuário que foi implementado é apresentado na figura 18. O fluxograma mostra a idéia principal do funcionamento do software de interface com o usuário que foi implementado.

Basicamente, como descrito no fluxograma da figura 18, a memória auxiliar de descarga do atleta, é lida pelo hardware auxiliar acoplado ao PC (figura 15), que conterà as informações que foram processadas durante a corrida, plotando gráficos de velocidade e aceleração médias pela distância total percorrida e gráficos de velocidades e acelerações instantâneas em todos os pontos que foram amostrados pelo sistema durante a realização do percurso pelo atleta.

A princípio, a memória auxiliar do atleta apresentará apenas informações de aceleração nos pontos que foram amostrados pelo hardware. O software carregará o seu banco de dados com as informações de aceleração contidas na memória, processando-as e montando os gráficos para o usuário do aplicativo desenvolvido.

Como já foi descrito no capítulo 1, o usuário desse sistema poderá verificar seu desempenho em pontos do percurso que foram amostrados pelo hardware, o que lhe possibilitará melhorar seus desempenhos futuros.

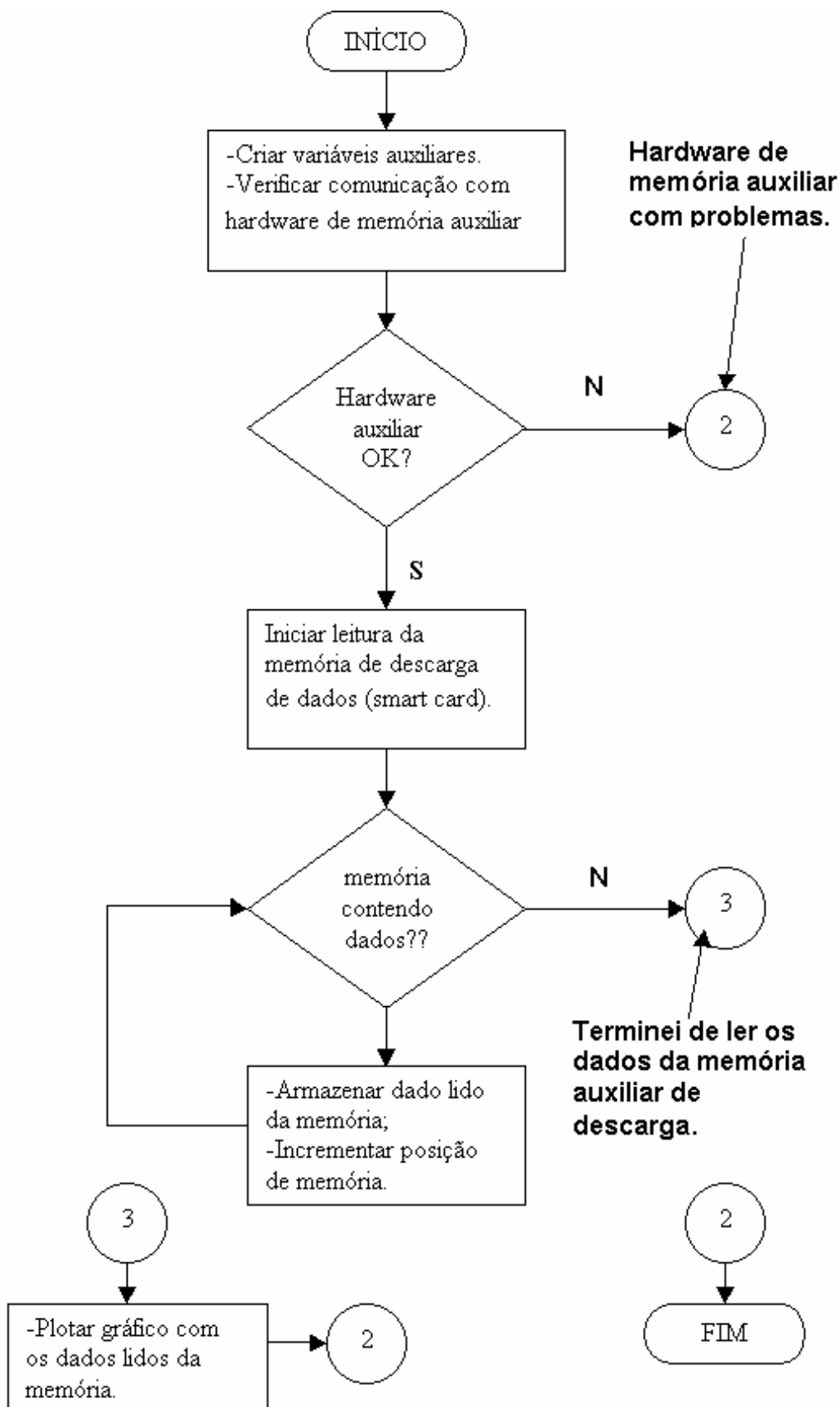


Figura 18 – fluxograma contendo a idéia básica do software de interface com o usuário

5.3.2 Levantamento de requisitos do usuário

5.3.2.1 Missão do produto

O sistema deverá apresentar ao usuário informações de distância total percorrida, velocidade e acelerações médias e instantâneas referentes a algum percurso que foi realizado por um atleta com a utilização do sistema.

Entrada no sistema:

O usuário do sistema deverá dirigir-se ao PC no qual estão instalados o software de interface gráfica e o hardware auxiliar de memória de leitura, com a memória auxiliar de descarga que foi utilizada pelo atleta durante a corrida em mãos. Deverá encaixar a memória com os dados da corrida no hardware auxiliar de forma correta e inicializar o software do sistema.

- Gráficos de velocidade média x distância percorrida

O usuário do sistema deverá selecionar a opção “Gráfico de velocidade média”, confirmando a operação.

- Gráficos de aceleração média x distância percorrida

O usuário do sistema deverá selecionar a opção “Gráfico de aceleração média”, confirmando a operação.

- Gráficos de velocidades amostradas

O usuário do sistema deverá selecionar a opção “Gráfico de velocidades amostradas”, confirmando a operação.

- Gráficos de acelerações amostradas

O usuário do sistema deverá selecionar a opção “Gráfico de acelerações amostradas”, confirmando a operação.

5.3.2.2 Lista de funções

Tabela 1 – lista de funções do sistema

Número de ordem	Nome da função	Necessidades	Benefícios	Escondido/Evidente
1	Inicialização	Fornecimento de informações a outras funções para inicializar o sistema.	Diminuição no tempo das operações e eventos por parte do usuário. Segurança do sistema.	Escondido.
2	Gráficos de velocidade	Geração de cálculos necessários para a obtenção das velocidades nos pontos amostrados pelo sistema durante a corrida.	Visualização de informações referentes a velocidades instantâneas que foram desenvolvidas pelo atleta durante a realização de seu percurso.	Evidente.
3	Gráficos de aceleração	Geração de cálculos necessários para a obtenção das acelerações nos pontos amostrados pelo sistema durante a corrida.	Visualização de informações referentes a acelerações instantâneas que foram desenvolvidas pelo atleta durante a realização de seu percurso.	Evidente.
4	Gráficos de velocidades amostradas	Geração de cálculos necessários para a obtenção das velocidades nos pontos amostrados pelo sistema.	Visualização das amostras de velocidade adquiridas pelo sistema durante a realização do percurso pelo atleta.	Evidente.
5	Gráficos de acelerações amostradas	Geração de cálculos necessários para a obtenção das acelerações nos	Visualização das amostras de aceleração adquiridas pelo sistema durante a	Evidente.

		pontos amostrados pelo sistema.	realização do percurso pelo atleta.	
--	--	---------------------------------	-------------------------------------	--

Tabela 1 (continuação) – lista de funções do sistema

5.3.3 Diagramas de casos de uso do sistema

Cenário 1: Montagem de gráfico de velocidade média x distância percorrida

1. Usuário do sistema encaixa memória auxiliar de descarta que contém as informações da corrida no hardware auxiliar ligado ao PC.
2. Usuário inicializa software do sistema que está instalado no PC em questão.
3. Usuário seleciona opção para montar gráfico que mostra a velocidade média x distância percorrida pelo atleta.

Casos de uso identificados:

Tabela 2 – casos de uso identificados no cenário 1

Caso de uso	Passos
Inicializar	1, 2
Plotar gráfico de velocidades	2, 3

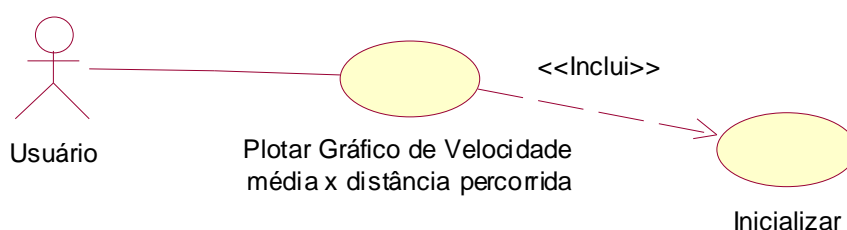


Figura 19 - Casos de uso identificados no Cenário 1

Descrição dos casos de uso identificados no cenário 1:

Caso de uso: Inicializar

Usuário inicializa sistema “abrindo” o software, que lerá o conteúdo da memória auxiliar que está encaixada no hardware auxiliar do PC (ver figura 1 e 15) e carregará o banco de dados do software do sistema.

Caso de uso: Plotar Gráfico de velocidades

Usuário após inicializar o sistema, seleciona a opção que plota gráfico de velocidade média x distância total percorrida pelo atleta.

5.3.3.1 Diagrama de seqüência dos casos de uso do cenário 1

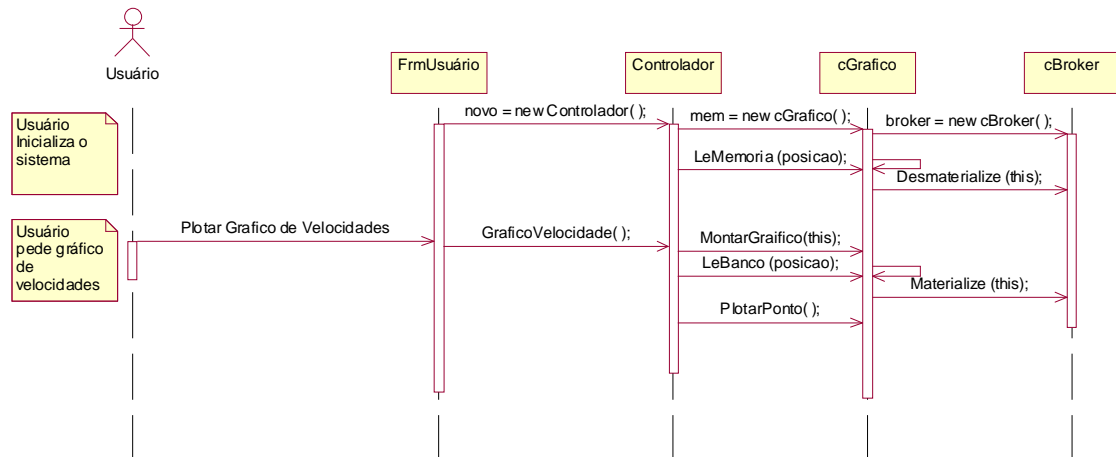


Figura 20 – Diagrama de seqüência dos casos de uso da figura 19

Cenário 2: Montagem de gráfico aceleração média x distância percorrida pelo atleta

1. Usuário do sistema encaixa memória auxiliar de descarga que contém as informações da corrida no hardware auxiliar ligado ao PC.
2. Usuário inicializa software do sistema que está instalado no PC em questão.
3. Usuário seleciona opção para montar gráfico que mostra a aceleração média x distância percorrida pelo atleta.

Casos de uso identificados:

Tabela 3 – casos de uso identificados no cenário 2

Caso de uso	Passos
Inicializar	1, 2
Plotar gráfico de Acelerações	2, 3

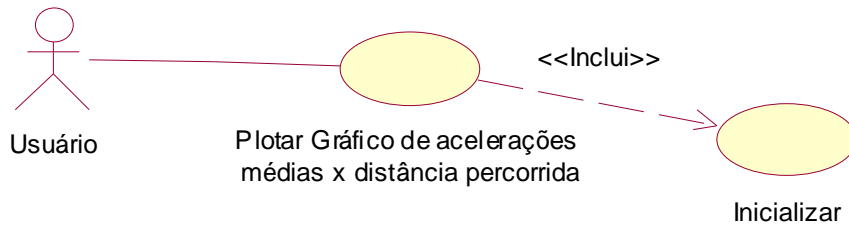


Figura 21 - Casos de uso identificados no Cenário 2

Descrição dos casos de uso identificados no cenário 2:

Caso de uso: Inicializar

Ídem cenário 1.

Caso de uso: Plotar Gráfico de Acelerações

Usuário após iniciiar o sistema, seleciona a opção que plota gráfico de aceleração média x distância total percorrida pelo atleta.

5.3.3.2 Diagrama de seqüência dos casos de uso do cenário 2

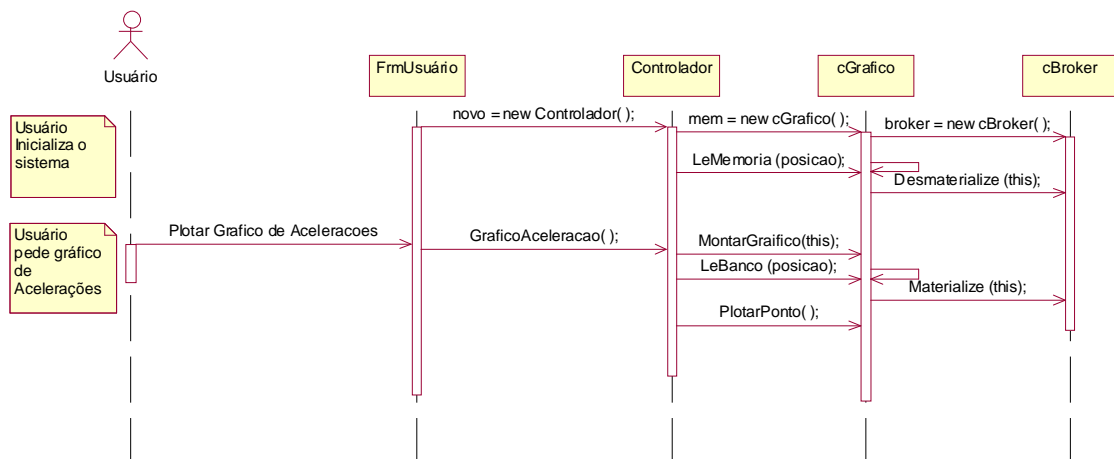


Figura 22 – Diagrama de seqüência dos casos de uso da figura 21

Cenário 3: Montar gráfico de amostras de velocidade

1. Usuário do sistema encaixa memória auxiliar de descarga que contém as informações da corrida no hardware auxiliar ligado ao PC.
2. Usuário inicializa software do sistema que está instalado no PC em questão.
3. Usuário seleciona opção para montar gráfico que mostra as velocidades instantâneas nos pontos que foram amostrados pelo sistema durante a corrida realizada pelo atleta.

Casos de uso identificados:

Tabela 4 – casos de uso identificados no cenário 3

Caso de uso	Passos
Inicializar	1, 2
Plotar gráfico de velocidades amostradas	2, 3

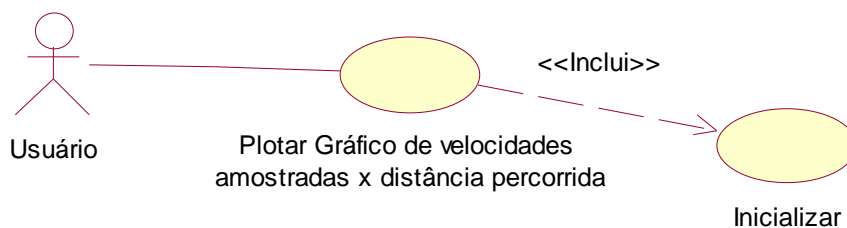


Figura 23 - Casos de uso identificados no Cenário 3

Descrição dos casos de uso identificados no cenário 3:

Caso de uso: Inicializar

Idem cenários 1 e 2.

Caso de uso: Gráfico de velocidades amostradas

Usuário após inicializar o sistema, seleciona a opção que plota gráfico de velocidade amostras pelo sistema durante a corrida.

5.3.3.3 Diagrama de seqüência dos casos de uso do cenário 3

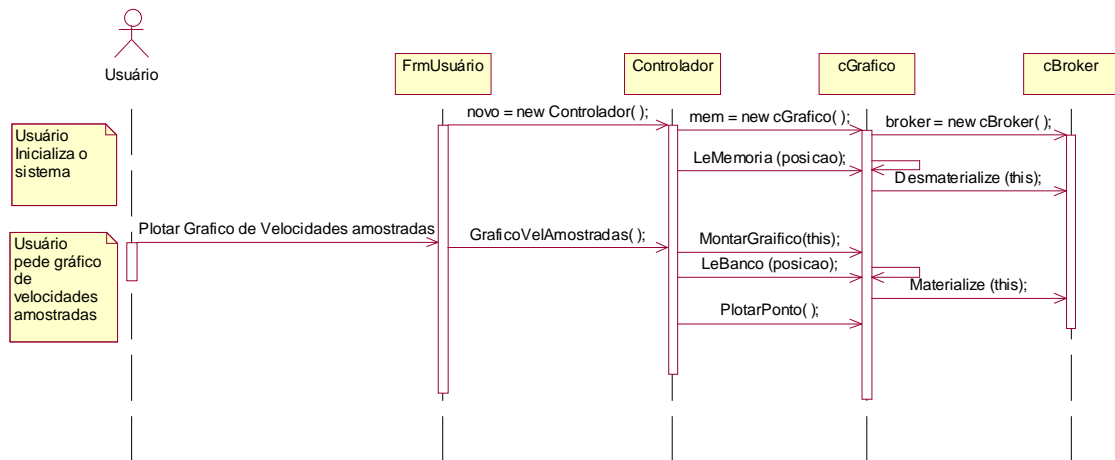


Figura 24 – Diagrama de seqüência dos casos de uso da figura 23

Cenário 4: Montar gráfico de acelerações amostradas

1. Usuário do sistema encaixa memória auxiliar de descarga que contém as informações da corrida no hardware auxiliar ligado ao PC.
2. Usuário inicializa software do sistema que está instalado no PC em questão.
3. Usuário seleciona opção para montar gráfico que mostra as acelerações instantâneas nos pontos que foram amostrados pelo hardware do sistema durante a realização da corrida.

Casos de uso identificados:

Tabela 5 – casos de uso identificados no Cenário 4

Caso de uso	Passos
Inicializar	1, 2
Plotar gráfico de acelerações amostradas	2, 3

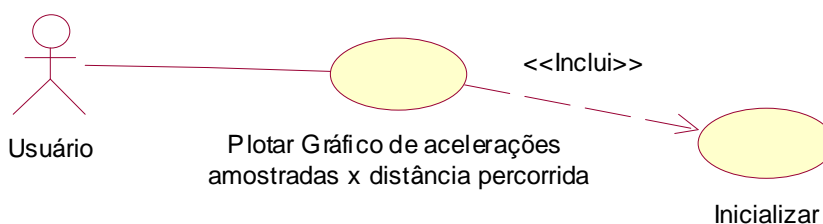


Figura 25 – Casos de uso identificados no cenário 4

Descrição dos casos de uso identificados no cenário 4:

Caso de uso: Inicializar

Idem cenários 1 e 2.

Caso de uso: Gráfico de acelerações amostradas

Usuário após inicializar o sistema, seleciona a opção que plota gráfico de acelerações amostras pelo sistema durante a corrida.

5.3.3.4 Diagrama de seqüência dos casos de uso do cenário 4

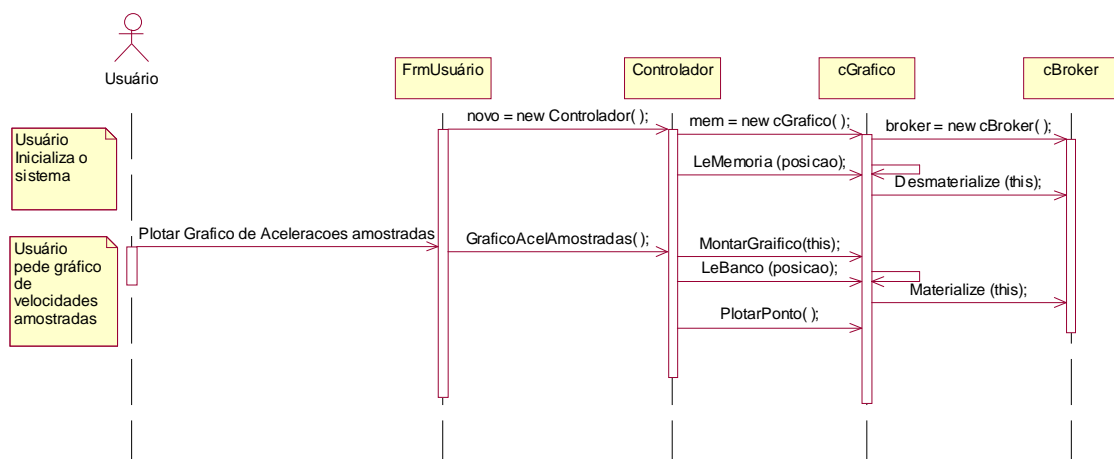


Figura 26 – Diagrama de seqüência dos casos de uso da figura 25

5.3.3.5 Diagrama de seqüência: Atores -> Sistema

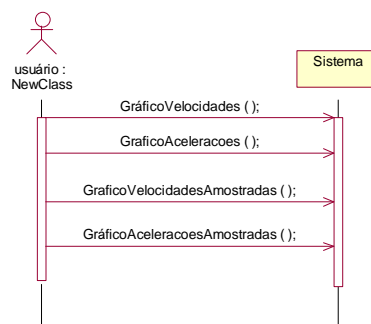


Figura 27 – Diagrama de seqüência: Atores -> Sistema

5.3.3.5.1 Contratos das operações do sistema

- **Contrato da operação “GráficoVelocidades”**

Contrato: GráficoVelocidades ();

Responsabilidades: Plotar gráfico de velocidade média x distância total percorrida pelo atleta.

Tipo: Sistema

Referências: Casos de uso (figura 19) do cenário 1.

Notas: Diagrama de classes (figura 28).

Excessões:

Saída: Visualização do gráfico com as informações de velocidade.

Pré – Condições: Banco de dados do sistema carregado ao inicializar o sistema.

Pós – Condições: Se um novo gráfico, foi criado um novo gráfico (instanciação de um novo gráfico).

- **Contrato da operação “GráficoAcelerações”**

Contrato: GráficoAcelerações ();

Responsabilidades: Plotar gráfico de aceleração média x distância total percorrida pelo atleta.

Tipo: Sistema

Referências: Casos de uso (figura 21) do cenário 2.

Notas: Diagrama de classes (figura 28).

Excessões:

Saída: Visualização do gráfico com as informações de Acelerações.

Pré – Condições: Banco de dados do sistema carregado ao inicializar o sistema.

Pós – Condições: Se um novo gráfico, foi criado um novo gráfico (instanciação de um novo gráfico).

- **Contrato da operação “GráficoVelocidadesAmostradas”**

Contrato: GráficoVelocidadesAmostradas ();

Responsabilidades: Montar gráfico contendo informações de velocidades nos pontos que foram amostrados durante o percurso.

Tipo: Sistema.

Referências: Casos de uso (figura 23) do cenário 3.

Notas: Diagrama de classes (figura 28).

Excessões:

Saída: Visualização do gráfico com as informações de velocidades.

Pré – Condições: Banco de dados do sistema carregado ao inicializar o sistema.

Pós – Condições: Se um novo gráfico, foi criado um novo gráfico (instanciação de um novo gráfico).

- **Contrato da operação “GráficoAceleracoesAmostradas”**

Contrato: GráficoAceleracoesAmostradas ();

Responsabilidades: Montar gráfico contendo informações de aceleração nos pontos que foram amostrados durante o percurso.

Tipo: Sistema.

Referências: Casos de uso (figura 25) do cenário 4.

Notas: Diagrama de classes (figura 28).

Excessões:

Saída: Visualização do gráfico com as informações de Acelerações.

Pré – Condições: Banco de dados do sistema carregado ao inicializar o sistema.

Pós – Condições: Se um novo gráfico, foi criado um novo gráfico (instanciação de um novo gráfico).

5.3.4 Diagrama de classes do sistema

O software foi desenvolvido no modelo de três camadas: Interface, controle e classes de negócios.

A figura 28 mostra o diagrama de classes do sistema que foi desenvolvido, levando em conta as classes de negócio, interface e controle.

5.3.4.1 Descrição das classes de interface, negócio e controle do sistema

Esse item do capítulo apresenta uma breve descrição das principais funcionalidades das classes de interface, negócio e controle do sistema que foi desenvolvido.

5.3.4.1.1 Classes de interface com o usuário

No sistema desenvolvido, temos uma interface com o usuário (figura 28 – classe FrmUsuário) na qual o mesmo tem disponível as opções de montagem de gráficos.

A interface conhece o controlador do sistema, e assim, quando o usuário solicitar a montagem de algum dos possíveis gráficos, a interface irá fazer uma solicitação ao controlador.

A descrição dos métodos da classe de interface do sistema com o usuário encontram-se a seguir na tabela 6.

Tabela 6 – Descrição dos métodos da interface do sistema

Método	Funções do método
void Grafvelocidade()	Envia uma solicitação ao controlador para a montagem de um gráfico de velocidade média x distancia percorrida pelo atleta
void GrafAceleracoes()	Envia uma solicitação ao controlador para a montagem de um gráfico de aceleração média x distancia percorrida pelo atleta
void GrafVelAmostradas()	Envia uma solicitação ao controlador para a montagem de um gráfico de velocidades instantâneas das amostras adquiridas pelo sistema durante a realização do percurso pelo atleta.
void GrafAcelAmostradas()	Envia uma solicitação ao controlador para a montagem de um gráfico de

	acelerações instantâneas das amostras adquiridas pelo sistema durante a realização do percurso pelo atleta.
--	---

Tabela 6 (continuação) – Descrição dos métodos da interface do sistema

5.3.4.1.2 Classe controladora de operações do sistema

O controlador do sistema receberá solicitações da interface que foi acionada pelo usuário. O controlador gerenciará as solicitações que recebe, passando-as essa solicitação para o único responsável em resolver o problema, a classe que manipulará as informações e montará os gráficos.

A descrição dos métodos da classe de controle do sistema encontra-se a seguir, na tabela 7.

Tabela 7 – Descrição dos métodos da classe de controle do sistema

Método	Funções do método
void GrafVel()	Envia a solicitação de montagem de gráfico de velocidade média x distância percorrida pelo atleta da interface ao seu destinatário correto.
void GrafAcel()	Envia a solicitação de montagem de gráfico de aceleração média x distância percorrida pelo atleta da interface ao seu destinatário correto.
void GrafVelAmostr()	Envia a solicitação de montagem de gráfico de velocidade dos pontos que foram amostrados pelo hardware dos sistema.
void GrafAcelAmostr()	Envia a solicitação de montagem de gráfico de aceleração dos pontos que foram amostrados pelo hardware do sistema.

5.3.4.1.3 Classe de negócio do sistema

A classe de negócios do sistema (figura 28 - Gráficos), receberá as tarefas que deverá fazer da classe de controle do sistema, o Controlador (figura 28).

A descrição dos métodos da classe de negócio do sistema encontra-se a seguir, na tabela 8.

Tabela 8 – Descrição dos métodos da classe de negócio do sistema

Método	Descrição do método
int LeMemoria()	Lê dados armazenados na memória auxiliar do atleta e invoca método especialista em gravar dados lidos da memória no banco de dados do sistema.
void GrafVelocidade()	Monta gráfico de velocidade média x distância percorrida pelo atleta.
void GrafAceleração()	Monta gráfico de aceleração média x distância percorrida pelo atleta.
void GrafVelAmostradas()	Monta gráfico de velocidades com as amostras que foram adquiridas pelo hardware durante a realização do percurso.
void GrafAcelAmostradas()	Monta gráfico de acelerações com as amostras que foram adquiridas pelo hardware durante a realização do percurso.
int LeBanco()	Método que lerá as informações que estão armazenadas no banco de dados do sistema.

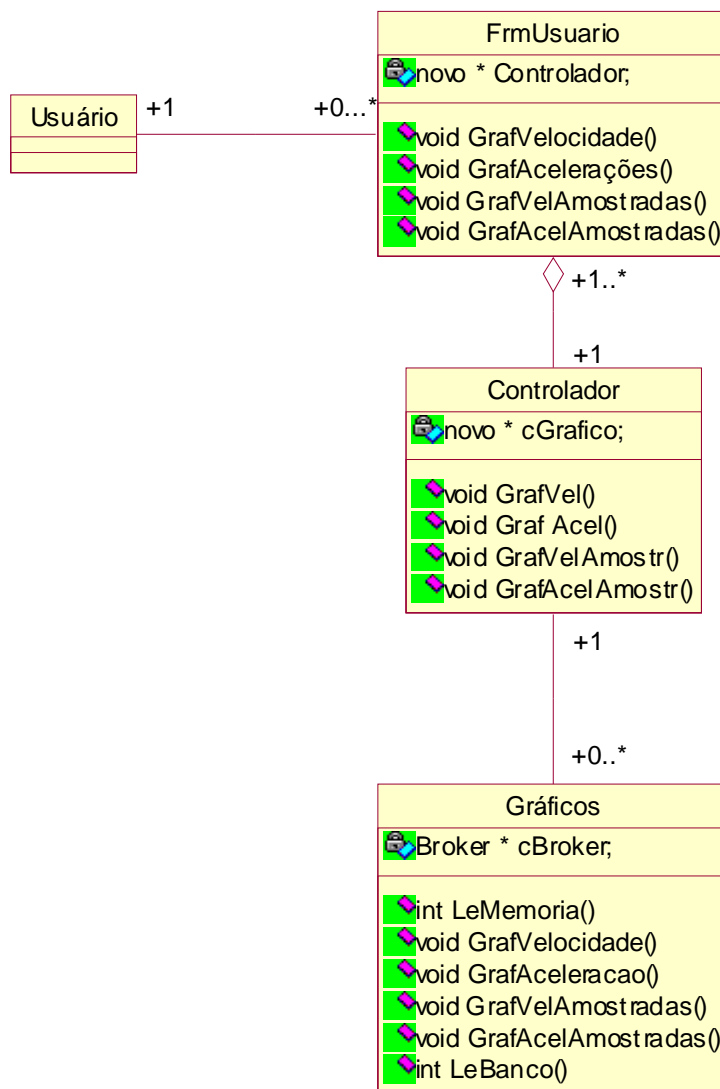


Figura 28 – Diagrama de Classes do sistema

5.4 Protótipo de telas do software de interface com o usuário

A interface gráfica do software de interface com o usuário que foi desenvolvida é mostrada na figura 29.

O usuário do sistema simplesmente, após ter encaixado no hardware de memória auxiliar de descarga a memória que contém as informações da corrida, selecionará a opção de gráfico que deseja visualizar e após isso apertar o botão “Plotar”. O software montará e disponibilizará ao usuário o gráfico que foi solicitado.

Além das opções mostradas na figura 29, o usuário, após ter montado um gráfico, poderá clicar sobre pontos de interesse no gráfico e o software abrirá outra

janela contendo informações de velocidades, acelerações e distância da passada do atleta, referentes ao ponto do gráfico no qual o usuário clicou.

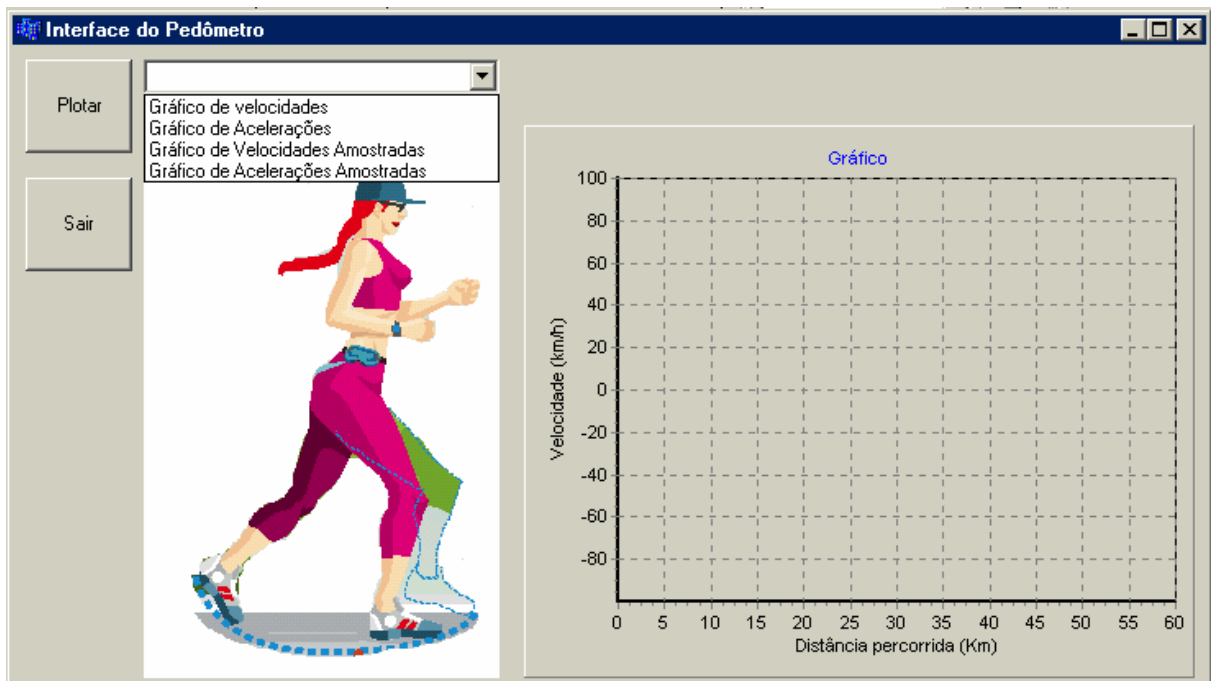


Figura 29 – Interface gráfica do software

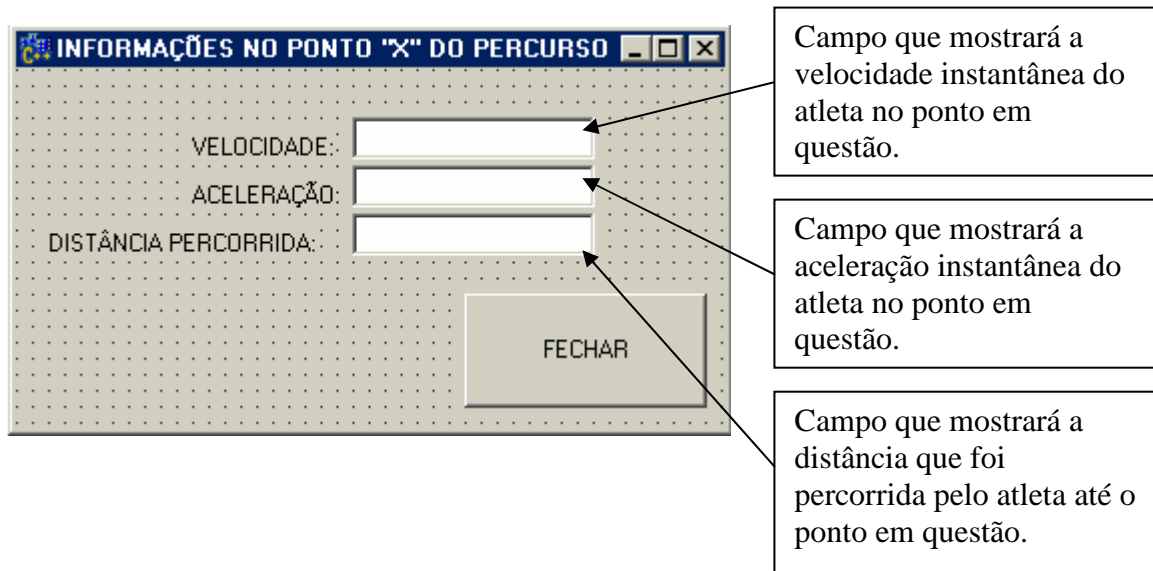


Figura 30 – Informações do percurso em um ponto escolhido pelo usuário

6. CRONOGRAMA DE DESENVOLVIMENTO DO PROJETO

A tabela 9 apresenta um cronograma básico com datas importantes para entrega de documentações e de implementações dos blocos do sistema apresentado.

Tabela 9 – Cronograma de desenvolvimento do projeto

Data	Atividade
28/03/05	-Entrega da especificação técnica do projeto a ser desenvolvido. -Início do desenvolvimento do hardware acoplado no tênis do atleta. -Início do desenvolvimento do software embarcado no microcontrolador.
02/05/05	-Entrega da monografia e do resumo/abstract do artigo para congresso.
20/06/05	-Finalização do hardware que será acoplado no tênis do atleta. -Finalização da implementação do software embarcado no microcontrolador.
21/06/05	-Início da implementação do monitor do atleta. -Início da implementação e modelagem (UML) do software de interface com o usuário.
18/07/05	-Finalização da implementação do hardware do monitor do atleta. -Finalização de modelagem (UML) e implementação do software de interface com o usuário.
19/07/05	-Início da implementação do hardware auxiliar de memória de descarga de dados.
08/08/05	-Finalização da implementação do hardware auxiliar de memória de descarga.
09/09/05	-Finalização do desenvolvimento do sistema. -Testes gerais com o sistema desenvolvido.
03/10/05	-Finalização dos testes com o sistema.
10/10/05	Apresentação do projeto implementado e qualificação para a fase final.
07/11/05	Entrega da documentação completa em espiral para a banca examinadora, em 3 vias, contendo a monografia, manual técnico, manual do usuário e artigo científico.
21 ^e 28/11/2005	Defesa formal do projeto, com apresentação oral para a banca examinadora.
12/12/05	Entrega da documentação completa, revisada e corrigida, encadernada no padrão da biblioteca (capa dura) em duas vias, contendo a monografia, manual técnico, manual do usuário e artigo científico;

7. ESTIMATIVA DE CUSTOS

Essa seção apresenta estimativas de investimento/custo para o desenvolvimento de uma unidade desse sistema, tanto investimentos para hardware quanto para software. Foram estipulados R\$ 50 para cada hora de trabalho gasta pelo desenvolvedor dos módulos desse sistema.

7.1 Estimativa de custos do hardware do sistema

Levando em conta os componentes utilizados e a quantidade de horas de trabalho para o desenvolvimento do sistema, a tabela 10 apresenta uma estimativa de investimento/custo para o desenvolvimento de uma unidade do projeto.

Tabela 10 – Estimativa e Custos para o desenvolvimento de uma unidade de hardware do projeto

Recurso	Quantidade	Custo (R\$)	Horas de trabalho sobre os componentes	Custo das horas de trabalho(R\$)
Acelerômetro ADXL202JE	1	90,14	20	1000
Microcontrolador MSC1211Y5pagt	2	60	15	750
DP1201	2	60	10	500
Latch 74ls373	1	1,12	0	0
Display	1	36	5	250
Cristal 11,059MHz	1	2,00	0	0
Custos totais	--	369,26	50	2500

7.2 Estimativa de custos do software do sistema

A tabela 11 mostra a estimativa de investimento/custo para o desenvolvimento do software de interface para o usuário e o software embarcado no microcontrolador do sistema em questão.

Tabela 11 - Estimativa e Custos para o desenvolvimento dos softwares do sistema

Recurso	Quantidade	Custo (R\$)	Horas de trabalho sobre os softwares	Custo das horas de trabalho (R\$)
Compilador Borland C++ Builder 6	1	1,170	20	1000
Compilador RIDE	1	350	15	750

Compilador Reads51	1	350	5	250
Modelagem de software (UML)	--	--	14	700
Custos totais	--	1870	54	2700

Tabela 11 (continuação) - Estimativa e Custos para o desenvolvimento dos softwares do sistema

8. ESPECIFICAÇÃO DE VALIDAÇÃO DO PROJETO

O projeto foi validado mediante testes do sistema que serão realizados na pista de atletismo do UNICENP e nos laboratórios de engenharia de computação (UNICENP) com a supervisão do professor José Carlos da Cunha.

8.1 Especificação de validação do hardware do sistema

A validação do hardware do sistema refere-se mais especificamente a parte do hardware que é acoplada no tênis do atleta, onde se encontra o sensor que é utilizado para monitorar as passadas do atleta (o acelerômetro) e o microcontrolador que é responsável por fazer a aquisição dos sinais da saída PWN do canal x desse sensor, interpretando-os e realizando os cálculos (aceleração e velocidade instantâneas e distância percorrida pelo atleta durante o percurso).

Para validar essa parte do sistema, mais especificamente a medição de acelerações e velocidades alcançadas pelo atleta durante a realização do percurso, utilizaremos alguns sistemas existentes no mercado que apresentam boa precisão e que sejam capazes de calcular e medir essas grandezas físicas. Para validar a distância percorrida pelo atleta, demarcaremos distâncias fixas que serão percorridas com o sistema desenvolvido, verificando a precisão dos cálculos do sistema no display do monitor do atleta.

As demais partes do hardware do sistema, o monitor do atleta e a memória auxiliar de descarga de dados, foram validados durante a utilização do sistema nos testes práticos (o monitor do atleta deve mostrar as mesmas informações do pedômetro comercial utilizado para os testes e o hardware de descarga da memória de dados foi validado mediante a testes com a interface gráfica do usuário, que irá ler a memória e montará o gráfico de velocidade (Km/h) x distância percorrida (Km). O gráfico montado pelo software deverá mostrar a distância percorrida pelo atleta, assim como deveria ter sido mostrado durante a corrida em seu monitor).

8.2 Especificação de validação dos softwares do sistema

Nessa seção apresentaremos os métodos e recursos que serão utilizados para validar os softwares que deverão ser desenvolvidos para o sistema, o software embarcado no microcontrolador e o software de interface gráfica com o usuário do sistema.

8.2.1 Validação do software embarcado no microcontrolador do sistema

Esse software foi validado juntamente com o hardware acoplado no tênis do atleta. O microcontrolador é responsável por processar, interpretar e realizar os devidos cálculos (deverá calcular a aceleração, a velocidade e a distância percorrida pelo atleta) com as informações da saída PWM do canal x do acelerômetro). Se as informações que aparecem no monitor do atleta conferirem com as do pedômetro que utilizado para os testes, saberemos que o software embarcado no microcontrolador está processando de forma correta os dados de saída do acelerômetro utilizado, validando o software do microcontrolador.

8.2.2 Validação do software de interface gráfica com o usuário

A validação desse software foi feita verificando-se as leituras e as montagens dos gráficos realizadas por esse aplicativo sobre o conteúdo lido da memória auxiliar de descarga de dados. Observando os gráficos montados pelo software, poderemos saber se as informações contidas nesses gráficos são verdadeiras, pois teremos monitorado a corrida no monitor do atleta, e assim, saberemos qual foi a distância percorrida pelo atleta, sua velocidade escalar média e sua aceleração média.

Sendo assim, validaremos esse software conferindo as informações que foram visualizadas pelo atleta durante a corrida em seu display e as informações contidas nos gráficos montados por esse software. Isso também ajudará a validar o restante do sistema, pois se as informações visualizadas no monitor do atleta baterem com as informações contidas nos gráficos plotados pelo software de interface, saberemos que o microcontrolador localizado no hardware do tênis do atleta (ver figura 11) gravou de forma correta os dados que foram processados e calculados durante a corrida na memória auxiliar de descarga de dados, que o microcontrolador do monitor do atleta recebeu os dados vindos do microcontrolador do tênis do atleta e os mostrou de forma correta no display do monitor do atleta (ver figura 12) e que o hardware conectado ao PC (ver figura 13) também está correto, validando assim, todo o hardware do sistema e os softwares desenvolvidos.

9. RESULTADOS

Após a realização de inúmeros testes, baseados em múltiplas baterias de pequenas caminhadas (pequenos percursos, com o número de passos realizados variando entre 5 e 30 passos) variando-se o tamanho e velocidade dos passos, observamos um comportamento padrão na curva característica de acelerações fornecidas pela aquisição de amostras do acelerômetro. Esse padrão de informações é mostrado nas figuras 1 e 2.

Conhecendo-se a curva característica de aceleração do movimento, como ilustrado nas figuras 1 e 2, conseguimos calcular a velocidade e a distância média de cada passo, realizando integrais sobre a área da curva no tempo dispendido para realizar o evento do passo [SWOKOWSKI, 1994].

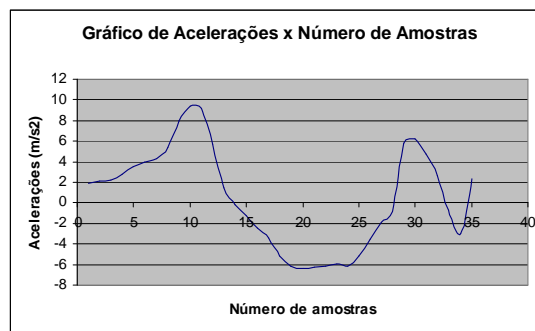


Figura 31 - Gráfico de acelerações amostradas para uma bateria aleatória de teste - 1º Passo realizado

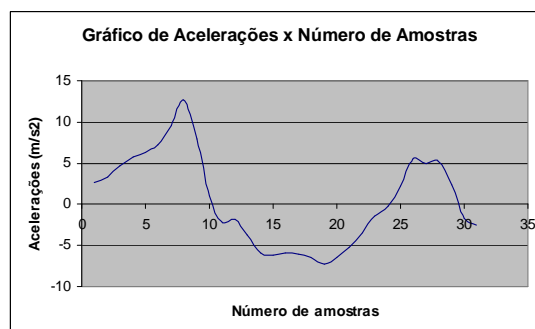


Figura 32 - Gráfico de acelerações amostradas para uma bateria aleatória de teste - 4º Passo realizado

Os testes realizados envolveram integração sobre toda a curva de acelerações (levando-se em conta as acelerações e desacelerações do passo), sobre as

acelerações da curva e sobre as desacelerações da curva característica de acelerações dos passos (figuras 1 e 2).

Os resultados mais satisfatórios obtidos, em termos de distância e velocidade para cada passo, foram os que utilizaram apenas a parte positiva da curva de acelerações nos cálculos.

Sendo assim, foi feita uma aproximação na curva de acelerações, considerando que a parte positiva da curva é igual a desaceleração da curva na realização do passo, o que gerou resultados satisfatórios em termos de distância e velocidade do passo efetuado. A curva característica apresentada na figura 3 nos mostra a aproximação realizada para os cálculos de velocidades e distâncias.

A figura 3 ilustra o que o firmware do microcontrolador do projeto faz com as amostras de aceleração que são obtidas durante a realização do passo pelo usuário. Basicamente, o microcontrolador faz a aquisição das amostras do acelerômetro tratando-as como um detector de pico [PERTENCE, 1996] implementado via software, descartando assim, amostras com valores de aceleração menores que o valor da maior amostra obtida até então e toda a parte negativa (desacelerações) da curva.

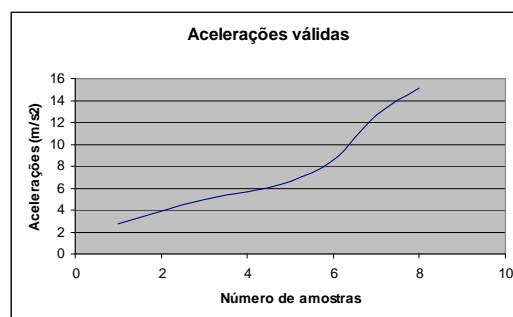


Figura 33 – Gráfico de Acelerações válidas pelo hardware do sistema para uma passo aleatório de uma bateria de testes

Como não estamos trabalhando com tempo contínuo, não podemos fazer integração das curvas apresentadas, portanto, trabalhamos com somatórios, que simulam uma integral para tempo amostrado [HAYKYN, 2001].

O somatório que é realizado pelo microcontrolador em curvas do tipo mostrado na figura 3 é duplicado após termos calculado a distância e a velocidade, pois trabalhamos apenas com metade da curva (as desacelerações são descartadas).

Para as distâncias percorridas utilizando o sistema desenvolvido, observou-se uma resposta razoável. Com relação a distância percorrida, observamos uma taxa de erro entre 10 e 15 metros para cada 300 metros percorridos, uma taxa de erro relativamente baixa se compararmos com os dispositivos encontrados no mercado que utilizam GPS para fazer o referenciamento do usuário durante a realização do percurso. Esses dispositivos de mercado apresentam uma taxa de erro média em torno de 20 metros para qualquer percurso desenvolvido.

Devido a possíveis mudanças de hardware, as placas de hardware não foram confeccionadas em uma máquina de prototipação comercial, foram feitas em placas padrões, o que comprometeu um pouco o tamanho físico do sistema. Ao término dos ajustes do projeto, teremos um hardware de aquisição de sinais que será amarrado no tênis do usuário do sistema e o monitor também terá seu tamanho físico reduzido em aproximadamente 60%.

A figura 31 apresenta uma visão geral do protótipo do sistema que foi desenvolvido até o presente momento. Observamos da esquerda para a direita da figura 31, o monitor do atleta, o hardware de descarga da memória de dados e o módulo de aquisição de sinais.



Figura 34 – visão geral do protótipo do sistema desenvolvido

10. CONCLUSÃO

Os sistemas encontrados no mercado, que não utilizam GPS para referenciar o atleta durante a realização da atividade física são muito imprecisos. Durante os testes realizados com pedômetros comerciais, observamos claramente a falta de precisão nas distâncias marcadas por esses dispositivos. Isso acontece pelo fato da distância de cada passo que é computada por esses dispositivos ser fixa, o que gera um erro muito grande na distância final que foi percorrida por usuários desses sistemas.

É perfeitamente possível desenvolver um sistema para monitorar atividades físicas como caminhadas, coopers e corridas com boa resolução e resposta utilizando acelerômetros, conhecimentos físicos e matemáticos, sem a utilização de GPS para referenciar o atleta durante o desenvolvimento do percurso.

11. REFERÊNCIAS BIBLIOGRÁFICAS

D. HALLIDAY, R. RESNIK; J. WALKER. *Fundamentos de física*. 4. ed. Rio de Janeiro, Editora Livros técnicos e científicos S.A., 1996.

SILVA, V.P.. *Aplicações práticas do microcontrolador 8051*. 2. ed. São Paulo, Érica, 2000.

NICOLOSI, D.E.C.. *microcontrolador 8051 detalhado*. 2. ed. São Paulo, Érica, 2003.

PRESSMAN, ROGER S.. *Engenharia de software*. 2.ed. São Paulo, Makron Books , 1995.

FURLAN, J.D.. *Modelagem de objetos através da UML*. 2. ed. São Paulo, Makron books, 1998.

MIZRAHI, VICTORINE VIVIANE. *Treinamento em linguagem C*. 3. ed. São Paulo, Makron Books , 1993.

STROUSTRUP, BJARNE. *A linguagem de programação C++*, 3. ed. Porto Alegre, Bookman, 2000.

SWOKOWSKI, EARL W., *Cálculo com Geometria Analítica*. 2. ed., São Paulo, Makron Books, 1994.

PERTENCE JUNIOR, ANTONIO, *Amplificadores Operacionais*. 5.ed., São Paulo, BOOKMAN, 1996.

HAYKYN, SIMON S., *Sinais e Sistemas*. 1. ed., São Paulo, Bookman, 2001.

12. ANEXOS