

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

PROTÓTIPO DE CONTROLE DE ACESSO PARA
ACADEMIAS DE GINÁSTICA UTILIZANDO
MICROCONTROLADOR PIC E O PADRÃO RS485

VILSON DO CARMO

BLUMENAU
2005

2005/2-27

VILSON DO CARMO

**PROTÓTIPO DE CONTROLE DE ACESSO PARA
ACADEMIAS DE GINÁSTICA UTILIZANDO
MICROCONTROLADOR PIC E O PADRÃO RS485**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciências
da Computação — Bacharelado.

Prof. Miguel Alexandre Wisintainer - Orientador

**BLUMENAU
2005**

2005/2-27

**PROTÓTIPO DE CONTROLE DE ACESSO PARA
ACADEMIAS DE GINÁSTICA UTILIZANDO
MICROCONTROLADOR PIC E O PADRÃO RS485**

Por

VILSON DO CARMO

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Miguel Alexandre Wisintainer – Orientador, FURB

Membro: _____
Prof. Mauro Marcelo Mattos – FURB

Membro: _____
Prof. Antonio Carlos Tavares – FURB

Blumenau, 30 de Outubro de 2005

Dedico este trabalho a todos os amigos,
especialmente aqueles que me ajudaram
diretamente na realização deste.

AGRADECIMENTOS

A Deus, pelo seu imenso amor e graça.

À minha família, que sempre me incentivou nos estudos.

Aos meus amigos, pelos empurrões e cobranças.

Ao meu orientador, Miguel Alexandre Wisintainer, por ter acreditado na conclusão deste trabalho.

Diz-me, e eu esquecerei; ensina-me e eu
lembrar-me-ei; envolve-me e eu aprenderei.

Autor desconhecido.

RESUMO

Este trabalho tem por objetivo implementar um protótipo para controle de acesso de alunos e funcionários em uma academia de ginástica. O protótipo permite a conexão de uma ou mais catracas com controle por intermédio de cartão com código de barras, interligadas a um microcomputador através de uma rede de comunicação serial utilizando o padrão RS485. O microcomputador possui um software que controla as catracas por meio de mensagens enviadas pelo canal de comunicação serial, permitindo ou não a liberação da referida catraca. Este software permite também que professor cadastre os dados dos alunos, o pagamento das mensalidades e as séries de exercícios que este deverá realizar. Permite também que o aluno, por meio de uma senha própria, acesse as séries de exercícios que deverá executar, tendo também a possibilidade de imprimi-las.

Palavras-chave: Serial. Microcontrolador

ABSTRACT

This work has for its objective to implement a prototype to control students' and employees' access to a gym club. The prototype allows the connection of one or more pivot barrier, controlled by a barcode card, linked to a microcomputer through a serial communication net that uses the RS485 pattern. The microcomputer uses software that controls the pivot barrier by means of messages sent through the serial communication channel and allows or not the liberation of the referred pivot barrier. This software also enables the teacher to include students' data, payment of monthly fees and the series of exercises that each student should take. It also allows the students to access their own series of exercises, by means of a personal code, with the possibility to print them.

Word-keys: Serial. Microcontroller.

LISTA DE ILUSTRAÇÕES

Quadro 1 – Caracteres do protocolo orientado a caracter	17
Figura 1 – Estrutura do protocolo orientado a caracter	17
Figura 2 – Estrutura do protocolo orientado a bit.....	18
Figura 3 – Catraca eletromecânica CA-1G	23
Figura 4 – Diagrama de atividades do protocolo de comunicação.....	29
Figura 5 – Formato do protocolo de comunicação	29
Quadro 2 – Caracteres utilizados no protocolo de comunicação do protótipo.....	30
Figura 6 – Circuito do leitor de código de barras da TECPOINT.....	31
Figura 7 – Protótipo do circuito eletrônico confeccionado em uma placa universal	32
Quadro 3– Configuração de registros do microcontrolador	35
Quadro 4 – Desvio para rotina principal	35
Quadro 5 – Rotina principal para o programa do microcontrolador	35
Quadro 6 – Sequência de execução da rotina principal.....	36
Quadro 8 – Sequência de execução da rotina de interrupção	38
Quadro 10 – Configuração dos parâmetros de comunicação	39
Quadro 11 – Rotina de tratamento do dado da porta serial	40
Quadro 12 – Rotina de transmissão de dados.....	40
Quadro 13 – Rotina para cálculo do CRC	41
Figura 9 – Tela inicial.....	42
Figura 11 – Tela para seleção do aluno para cadastro de série de exercícios.....	44
Figura 12 – Tela para cadastro de série de exercícios	45
Figura 13 – Tela para registro de pagamento de mensalidade	46

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 OBJETIVOS DO TRABALHO	12
1.2 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 COMUNICAÇÃO DE DADOS.....	14
2.1.1 Padrão RS232.....	15
2.1.2 Padrão RS485.....	15
2.1.3 Protocolos.....	16
2.1.3.1 Protocolo orientado a caracter	17
2.1.3.2 Protocolo orientado a bit.....	18
2.2 MICROCONTROLADORES	19
2.2.1 Linguagem de programação para microcontroladores.....	20
2.2.1.1 Linguagem Assembly para microcontroladores	20
2.2.1.2 Linguagem Basic para microcontroladores	21
2.2.1.3 Linguagem C para microcontroladores	21
2.3 CÓDIGO DE BARRAS	21
2.4 LEITOR DE CÓDIGO DE BARRAS.....	22
2.5 CATRACA ELETROMECAÂNICA	23
2.6 AUTOMAÇÃO DE ACADEMIAS DE GINÁSTICA	23
2.7 TRABALHOS CORRELATOS.....	24
3 DESENVOLVIMENTO DO TRABALHO	26
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	26
3.2 ESPECIFICAÇÃO	26
3.2.1 Especificação do circuito eletrônico	27
3.2.2 Especificação do software para o microcontrolador	28
3.2.3 Especificação do software para o PC.....	28
3.2.4 Especificação do protocolo de comunicação	28
3.3 IMPLEMENTAÇÃO	30
3.3.1 Técnicas e ferramentas utilizadas.....	30
3.3.1.1 Leitor do código de barras	31
3.3.1.2 Implementação do protótipo do circuito eletrônico.....	32

3.3.1.3 Implementação do software para o microcontrolador	33
3.3.1.4 Implementação do software para o PC	39
3.3.1.5 Testes e validação	41
3.3.2 Operacionalidade da implementação	42
3.4 RESULTADOS E DISCUSSÃO	47
4 CONCLUSÕES.....	49
4.1 EXTENSÕES	50
REFERÊNCIAS BIBLIOGRÁFICAS	51
APÊNDICE A – Parte 1 do esquema eletrônico	53
APÊNDICE B – Parte 2 do esquema eletrônico	54
APÊNDICE C – Especificação do Software do microcontrolador	55
APÊNDICE D - Especificação do Software para o PC (Diagrama de Atividades 1)	56
APÊNDICE E - Especificação do Software para o PC (Diagrama de Atividades 2)	57
APÊNDICE F - Especificação do Software para o PC (Diagrama de Seqüência 1)	58
APÊNDICE G - Especificação do Software para o PC (Diagrama de Seqüência 2).....	59
APÊNDICE H - Especificação do Software para o PC (Diagrama de Estados).....	60

1 INTRODUÇÃO

Atualmente, com o avanço tecnológico, a automação tornou-se possível em todas as áreas. Praticamente todos os setores podem ser parcial ou totalmente automatizados, proporcionando de uma maneira geral, maior segurança, agilidade, economia, conforto e um conseqüente aumento em qualidade de vida. O surgimento do microcontrolador contribuiu enormemente para a disseminação da automação, principalmente fora dos ambientes industriais. Com a utilização de microcontrolador, projetos envolvendo eletrônica tornam-se bastante facilitados, pois somente este componente, pode substituir circuitos eletrônicos contendo um grande número de outros componentes. Segundo Silva Júnior (1998, p. 1), microcontrolador “é um componente que possui todos os periféricos dos microprocessadores comuns embutidos em uma só pastilha, facilitando assim o desenvolvimento de sistemas pequenos e baratos, embora complexos e sofisticados”.

Hoje, há muitas academias de ginásticas nas cidades, e elas são freqüentadas diariamente por um grande número de pessoas. Torna-se interessante, portanto, controlar o acesso dessas pessoas para o interior dessas academias, tanto para possibilitar um maior controle por parte de sua administração, quanto para oferecer maior segurança aos demais usuários. Este controle dá-se com o emprego de catracas eletrônicas, através das quais o usuário só terá seu acesso liberado, caso possua um cachá ou cartão com um código de barras impresso e devidamente cadastrado no sistema e em dia com suas obrigações para com a academia.

Para confeccionar um protótipo, capaz de realizar este controle de acesso, utilizou-se um circuito leitor de código de barras, através do qual, o usuário passará seu cartão. Este leitor foi desenvolvido e é comercializado pela empresa TECPOINT (TECPOINT, 2004). Este circuito captura o código e o envia por um canal serial utilizando o padrão *Recommended*

Standard 232 (RS232) para o microcontrolador, que por sua vez, o envia para um microcomputador (PC) por meio de outro canal de comunicação serial e com a utilização do padrão de comunicação *Recommended Standard 485*(RS485). Aos alunos ainda será possível, através do PC, imprimir as séries de exercícios que deverão realizar, previamente cadastradas por seu instrutor.

Este protótipo envolve vários assuntos distintos, que devem ser dominados, por quem deseja trabalhar com automação, como o desenvolvimento de circuitos com microcontroladores e sua programação, bem como técnicas de interligação entre hardwares diferentes.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um sistema para controlar o acesso de alunos de uma academia de ginástica às áreas de treinamento, através de cartões com código de barras.

Os objetivos específicos do trabalho são:

- a) disponibilizar aos alunos a consulta de suas respectivas fichas, com informações a respeito das séries de exercícios que devem realizar;
- b) permitir aos instrutores o controle das atividades físicas dos alunos;
- c) permitir à administração o controle de pagamento dos alunos;
- d) disponibilizar um sistema integrado de hardware e software para comunicação serial utilizando o padrão RS485.

1.2 ESTRUTURA DO TRABALHO

Este trabalho, inicialmente no capítulo 1 apresentará uma introdução e os objetivos a serem alcançados. No capítulo 2 será apresentada toda fundamentação teórica dos assuntos que serão abordados pelo trabalho e os trabalhos correlatos pesquisados. No capítulo 3 será apresentado o desenvolvimento do trabalho, abordando seus requisitos, especificações e a implementação dos softwares. E ao final, no capítulo 4 será apresentada a conclusão e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados alguns aspectos teóricos relacionados ao trabalho, tais como: comunicação de dados, código de barras, catracas eletromecânicas, automação de academias, linguagens para programação de microcontroladores e ao final, são apresentados alguns trabalhos correlatos.

2.1 COMUNICAÇÃO DE DADOS

Segundo Loesh, Stringari e Tafner (1996) a comunicação é formada de meios e regras pelas quais, a mensagem, o elemento causador desta comunicação, é o componente principal. Porém, só é válida a comunicação se a mensagem chegar ao seu destino com a mesma qualidade que saiu de sua origem.

Uma das formas de se estabelecer a comunicação entre dois ou mais equipamentos digitais, é através da comunicação serial.

Afirma Canzian (2001) que por não ser prático nem econômico transferir todos os bits de uma mensagem simultaneamente, a mensagem é “quebrada” em partes menores e transmitida seqüencialmente. A transmissão bit-serial converte a mensagem em um *bit* por vez através de um canal. Cada *bit* representa uma parte da mensagem. Os bits individuais são então rearranjados no destino para compor a mensagem original. Em geral, em um canal irá passar apenas um *bit* por vez. A transmissão bit-serial é normalmente chamada de transmissão serial.

Para se garantir que uma mensagem saia do transmissor e chegue ao receptor integralmente e que este, por sua vez entenda a mensagem, faz-se necessário o

estabelecimento de padrões elétricos e protocolos de comunicação, os quais serão vistos a seguir.

2.1.1 Padrão RS232

De acordo com Canzian (2001) ruídos e distúrbios elétricos momentâneos podem causar mudanças nos dados quando estão trafegando pelos canais de comunicação. Com o objetivo de eliminar ou atenuar sensivelmente os erros gerados por ruídos elétricos, bem como aumentar a potência do sinal, permitindo desta forma uma maior distância entre as fontes emissoras de sinal e as receptoras, implantou-se o padrão RS232.

Segundo Cunha (2000), um PC utiliza a tecnologia RS232, definida da estrutura IBM-PC, como forma de acesso ao meio físico. Este padrão, entretanto, estabelece apenas o padrão relativo à questão elétrica.

Para maiores informações a respeito do padrão RS232 o trabalho de CUNHA (2000) poderá ser consultado.

2.1.2 Padrão RS485

Este padrão, analogamente ao RS232, tem como propósito garantir a imunidade do sinal quanto às questões de ruído e distância. De acordo com Matias (2002), este padrão é o mais utilizado em ambiente industrial nos dias de hoje. Ele é chamado normalmente de RS485, entretanto seu nome oficial é *Electronic Industry Association (EIA) EIA-485*, que reflete o nome do comitê que no ano 1998 foi padronizado. Ainda, segundo Matias (2002), o padrão RS485 é baseado na transmissão diferencial de dados que é ideal para transmissão em altas velocidades, longas distâncias e em ambientes propícios a interferências

eletromagnéticas. Segundo Axelson (2000, p. 186), este padrão representa um baixo custo de implantação e pode transmitir dados a uma velocidade de até 10Mbs, podendo suportar distâncias de 1200 metros, porém quanto maior for esta distância menor será sua taxa de transmissão ou número de nós, sendo que cada nó representa um equipamento participante da rede.

2.1.3 Protocolos

Segundo Loesh, Stringari e Tafner (1996), protocolo é um conjunto de regras pré-estabelecidas, cuja função é fazer com que a comunicação de dados entre equipamentos seja realizada com segurança e de forma ordenada. Estas regras obedecem a uma seqüência lógica e normalmente padronizada. Os protocolos podem ser classificados como assíncronos ou síncronos, de acordo com seus controles de transmissão.

No controle assíncrono, de acordo com Loesh, Stringari e Tafner (1996, p. 32) estes protocolos utilizam *bits* de *start* e *stop* para delimitar um caracter, tornando a comunicação pouco eficiente. Neste modo, o código usado é normalmente o *American Standart Code for Information Interchange(ASCII)*.

No controle síncrono, seguindo ainda afirmação de Loesh, Stringari e Tafner (1996, p. 32), são aqueles em que tanto o transmissor, quanto o receptor trabalham com o mesmo *clock*, não sendo, portanto, necessário utilizar *bits* de *start* e *stop*.

A seguir serão mostrados dois protocolos: orientados a caracter e orientados a *bit*.

2.1.3.1 Protocolo orientado a caracter

De acordo com Loesh, Stringari e Tafner (1996), são protocolos que utilizam caracteres especiais para o controle de operação e do tráfego de suas mensagens. Estes protocolos são os mais antigos e são usados com frequência em linhas de longa distância. O *Binary Synchronous Communication* (BSC) é um exemplo de protocolo orientado a caracter muito utilizado atualmente.

Há 16 caracteres que controlam a conversação do protocolo, e todos estão relacionados nos caracteres especiais do código ASCII. A seguir, o quadro 1 apresenta os caracteres mais comuns utilizados e suas funções, de acordo com Loesh, Stringari e Tafner (1996).

Caracter	Cód.(hex)	Função
SOH (<i>Start of Header</i>)	01	Identifica o início do cabeçalho
STX (<i>Start of Text</i>)	02	Identifica o início do texto e o fim do cabeçalho. Também serve para zerar o BCC.
ETX (<i>End of Text</i>)	03	Fim do texto. É sempre transmitido como último caracter. Também indica que logo após segue o BCC.
EOT(<i>End of Transmission</i>)	37	Indica o fim da transmissão.
ETB (<i>End of Transmission Block</i>)	26	Fim de um bloco. Iniciado com um SOH ou um STX.
ENQ (<i>Enquiry</i>)	2D	Obter resposta de estação remota. Ou seja, pede uma resposta em X segundos à estação, ou requer uma retransmissão da última mensagem.
ACK0 e ACK1 (<i>Acknowledge</i>)	1070 e 1061	Reconhecimento positivo de uma mensagem recebida e está pronto para receber outra mensagem.
NAK (<i>Negative Acknowledge</i>)	3D	Reconhecimento negativo de uma mensagem recebida, ficando para a estação transmissora a função de uma retransmissão.
SYN (<i>Synchronous Idle</i>)	32	Mantém o sincronismo entre o transmissor e o receptor.
ITB (<i>End of Intermediate Transmission Block</i>)	1F	Caracter usado para dividir o texto da mensagem transmitida em blocos.

Quadro 1 – Caracteres do protocolo orientado a caracter

O formato típico de um bloco de protocolo orientado a caracter é mostrado na figura 1 a seguir.

SYN	SYN	SOH	CABEÇALHO	STX	TEXTO	ETX	BCC
-----	-----	-----	-----------	-----	-------	-----	-----

Fonte: Loesh, Stringari e Tafner (1996, p. 34)

Figura 1 – Estrutura do protocolo orientado a caracter

O formato anterior representa uma conexão entre dois equipamentos. Inicialmente o bloco é enviado com caracteres de sincronismo SYN, seguidos do caracter de SOH que pode indicar o envio de endereço do transmissor. Após o SOH, é enviado o cabeçalho com o endereço do receptor de destino, após isso é enviado a mensagem ou texto, que é precedida com o caracter STX. Após o texto, pode seguir o caracter ETB, significando que, apesar de ter sido preenchido todo campo disponível no bloco, o texto ainda não terminou, ou o caracter ETX, indicando o fim do texto. E por fim, é colocado o bloco *Block Check Character* (BCC), que são um ou mais caracteres para verificação de erros (LOESH; STRINGARI; TAFNER, 1996).

2.1.3.2 Protocolo orientado a bit

De acordo com Loesh, Stringari e Tafner (1996), protocolos orientados a *bit* são protocolos que não utilizam caracteres especiais para delimitar blocos de mensagem. Todo o controle é tratado em nível de *bit*, ou seja, campos formados por combinações binárias bem definidas, não existindo caracteres que designem funções especiais. Segundo Loesh, Stringari e Tafner (1996) ainda, o formato típico do protocolo orientado a *bit* é mostrado na figura 2.

FLAG	ENDEREÇO	CONTROLE	TEXTO	FSC	FLAG
------	----------	----------	-------	-----	------

Figura 2 – Estrutura do protocolo orientado a bit

FLAG é um caracter para delimitar o inicio e o fim da mensagem *Synchronous Data Link Control* (SDLC). É constituído pela seqüência 01111110. Também é utilizado para disparar o algarismo de verificação de erros. ENDEREÇO é o campo formado por 8 bits, que codificam o controle da comunicação. TEXTO é a mensagem propriamente dita e por último *Frame Check Sequence* (FCS) é um campo formado por 16 bits para verificação de erros

empregando a técnica *Cyclic Redundancy Cheking* (CRC) (LOESH; STRINGARI; TAFNER, 1996).

2.2 MICROCONTROLADORES

Os microcontroladores tornaram-se muito populares atualmente, pois praticamente qualquer aparelho eletrônico moderno é controlado por um microcontrolador. Com sua utilização foi possível construir sistemas mais compactos e mais versáteis, com um custo relativamente baixo (SILVA JÚNIOR, 1998).

Existem diversas empresas fabricantes de microcontroladores, entre elas, a Microchip Technology Inc.(MICROCHIP, 2004), que criou a família de microcontroladores *Peripheral Interface Controller* (PIC). Trata-se de microcontroladores com arquitetura Harvard que, de acordo com Silva Júnior (1998, p. 3), prevê várias vias de comunicação entre *Central Processing Unit* (CPU) e periféricos, permitindo a realização de várias operações simultaneamente, o que implica em um aumento considerável na velocidade de execução e permite ainda que memória de dados e programa tenham tamanhos diferentes. O PIC possui tecnologia *Reduced Instruction Set Computer* (RISC), isto é, possui um número reduzido de instruções (no caso do PIC16F877A apenas 35).

O PIC16F877A pode operar com *clock* de 20MHz, o que caracteriza a execução de um ciclo de máquina em 200 nanosegundos. Possui ainda vários periféricos, que facilitam o desenvolvimento de sistemas, como exemplo, 8K de memória *FLASH* para armazenamento do programa de controle, canal de comunicação serial *Universal Synchronous Serial Receiver Transmitter* (USART), 256 bytes *Electric Erasable Programmable Randomic Memory* (EEPROM), 33 pinos que podem ser configurados como portas de *Input/Output* (I/O) e 368 bytes de memória *Random Acess Memory* (RAM), para dados.

2.2.1 Linguagem de programação para microcontroladores

As linguagens mais utilizadas atualmente para a programação de microcontroladores são as linguagens Assembly, C e Basic. Antes que se escolha entre uma destas linguagens, é necessário examinar cuidadosamente os requisitos de velocidade de execução, de espaço de memória a ocupar e o tempo disponível para o desenvolvimento do programa (MIKROELEKTRÔNICA, 2003). Adiante, segue uma breve introdução sobre cada uma destas linguagens.

2.2.1.1 Linguagem Assembly para microcontroladores

Segundo Pereira (2003) a linguagem Assembly consiste em uma forma alternativa de representação dos códigos de máquina usando mnemônicos, ou seja, abreviações de formas usuais que descrevem a operação efetuada pelo comando em código de máquina.

A linguagem Assembly é de baixo nível, isto é, não possui nenhum comando, instrução ou função além daqueles definidos no conjunto de instruções do processador utilizado. Isto implica em um trabalho extra do programador para desenvolver rotinas e operação que não fazem parte do conjunto de instruções do processador produzindo, por conseguinte, programas muito extensos e complexos com um fluxo muitas vezes difícil de ser seguido. Contudo, é a linguagem que oferece os melhores resultados quando se pretende poupar espaço de memória e aumentar a velocidade de execução do programa (MIKROELEKTRÔNICA, 2003).

2.2.1.2 Linguagem Basic para microcontroladores

Trata-se de uma linguagem de alto nível. Dispõe de instruções para implementar comunicação serial, *Pulse Width Modulation* (PWM), leitura de entrada analógica, geração e medida de impulsos, geração de tom e gravação e leitura de memória EEPROM. Possui entrada para botões com eliminação de ruídos e auto-repetição, conhecidos por *debouncing*. Entretanto, é uma linguagem de execução lenta, pois executa aproximadamente apenas 2000 instruções por segundo (CUENCA; MARTÍNEZ; USATEGUI, 1998, p. 280).

2.2.1.3 Linguagem C para microcontroladores

Esta linguagem, segundo Pereira (2003), devido a sua proximidade com o hardware e o Assembly, é uma linguagem extremamente eficiente. É considerada como a linguagem de alto nível mais eficiente atualmente disponível.

De acordo com Cuenca, Martinez e Usategui (1998), linguagens de alto nível como Basic e C devem permitir a inserção de código em Assembly para aumentar a performance do programa, em casos onde se requer rapidez na execução da função.

2.3 CÓDIGO DE BARRAS

Segundo Grossman e Zyngier (1991) o código de barras é um símbolo composto de barras paralelas de largura e espaçamentos variados. Estabeleceu-se que as barras são as regiões que não refletem a luz ao equipamento leitor, e os espaços em branco entre elas são regiões que refletem. Afirmam Grossman e Zyngier (1991) que existem vários padrões que normalizam os códigos de barra, entre eles o código 39, código 128, código *Universal*

Product Code (UPC) e European Article Numbering (EAN).

O código de barra mais popular do mundo é o UPC, é o formato encontrado nos produtos de supermercados. Ele se encontra disponível desde a década de 70, este formato é conhecido no mundo inteiro.

2.4 LEITOR DE CÓDIGO DE BARRAS

O leitor de código de barras é um equipamento com capacidade para capturar um código de barras impresso em um cartão, crachá ou algum outro tipo de documento e disponibilizá-lo em uma porta de saída, serial ou paralela, para comunicação com outros equipamentos, como um PC por exemplo.

O leitor é formado por um sensor ótico, uma canaleta e uma placa decodificadora. Na canaleta, cuja principal função é servir como guia para o cartão, é fixado o sensor ótico. Este sensor lê os sinais provenientes do código de barra impresso nos cartões e os envia para a *Unidade de Processamento Central (CPU)* da placa decodificadora. A placa decodificadora captura os sinais do sensor e os codifica de acordo com alguns padrões existentes, como o padrão 2 de 5 intercalados, 39, EAN, entre outros.

A placa decodificadora ainda tem a função de enviar o código para outro dispositivo, por um canal serial ou uma saída paralela, dependendo do modelo. A placa decodificadora também pode receber comandos para configuração de seus parâmetros como velocidade, paridade e dados de bits e stop bits.

O leitor de código de barras não possui um protocolo de comunicação específico. Ao se passar o código de barras, ao seu final, este código é imediatamente enviado pela porta de saída (TECPOINT, 2004).

2.5 CATRACA ELETROMECAÂNICA

Catracas Eletromecânicas são equipamentos utilizados para controle de acesso. Possuem, solenóides de travamento, de forma que somente permitem o giro da catraca quando estes são acionados. Possuem sensores de giro, para indicar que a catraca sofreu um giro, ou seja, ocorreu a passagem de uma pessoa. Essas catracas permitem a passagem em único sentido ou em ambos os sentidos.

Este tipo de catraca deve ser interligada a coletores de dados para o controle de acesso, como leitores de código de barras ou cartões magnéticos. Elas podem ser utilizadas para controle de entradas de pessoas em clubes sociais, instituições de ensino, empresas, parques temáticos, eventos, entre outros (PASSO AUTOMAÇÃO, 2003). A figura 3 a seguir mostra um exemplo de catraca eletromecânica.



Fonte: <http://www.passo.com.br/eletromec.htm>

Figura 3 – Catraca eletromecânica CA-1G

2.6 AUTOMAÇÃO DE ACADEMIAS DE GINÁSTICA

A automação de academias de ginástica tem como objetivo facilitar sua administração.

Pode tanto controlar o fluxo de usuários, quanto as atividades exercidas por eles. Pode-se, por exemplo, controlar as atividades e serviços, dados dos funcionários e alunos, controle de pagamentos, fluxo de caixa, fornecedores, exames médicos, frequência e horário de acesso. Sistemas mais complexos permitem criar relatórios mais específicos, emitir gráficos estatísticos e permitir integração com catracas eletrônicas, de modo a controlar o acesso de usuários. Estes sistemas podem ainda fornecer ferramentas para avaliação dos alunos, efetuando exames de composição corporal, perimetria, ergonomia e teste postural. Pode oferecer ferramentas para controlar as séries de exercícios executados pelos alunos, permitindo inclusive que as mesmas sejam impressas. Pode emitir dados para a utilização na sala de exercícios cardiorespiratórios, volumes de oxigênio, pressão arterial e frequência cardíaca, de modo que o aluno pode ser monitorado caso esteja fazendo pouco esforço ou esforço excessivo na bicicleta, esteira, entre outros. Alguns sistemas permitem também a emissão de carteiras com códigos de barras, para os alunos (SYSNET, 2003).

2.7 TRABALHOS CORRELATOS

Existem vários trabalhos que envolvem o assunto estudado, cada um aborda um determinado assunto que muito auxiliou no desenvolvimento do protótipo, como o de CUNHA (2000), no qual, é abordado o assunto sobre RS485 e um modelo de protocolo de comunicação.

APS é um software comercial, utilizado em uma academia de ginástica, situada na cidade de Indaial, que serviu como exemplo de interface e funcionamento para o desenvolvimento do software para o PC, para o controle da academia.

O Trabalho de MORETTI (1999) descreve o funcionamento do código de barras, servindo como uma das fontes para consulta do referido assunto. Encontrou-se em (SYSNET,

2003) um software comercial, denominado *Auto-Gym for Windows*, desenvolvido para administrar academias de ginástica, permitindo controlar as atividades de alunos e professores. Este software é apresentado em módulos específicos para as diversas áreas que são abrangidas dentro de uma academia, de forma que foi possível adquirir um bom entendimento sobre seu funcionamento e alguns aspectos mínimos que a interface do trabalho deve abranger.

3 DESENVOLVIMENTO DO TRABALHO

Neste capítulo serão descritos os requisitos, as especificações dos softwares tanto para o microcontrolador quanto para o PC. Serão apresentados também, alguns detalhes do software para o microcontrolador, bem como demais detalhes para a construção do protótipo. Em seguida, será apresentado a operacionalidade do sistema e, por fim, os resultados obtidos durante os testes de validação.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O protótipo deverá possibilitar a automação da academia de ginástica, no que se refere ao acesso dos usuários e o acesso desses à suas respectivas séries de exercícios. Para isto o protótipo deverá possibilitar a comunicação entre um PC e uma ou mais catracas eletromecânicas. O protótipo deverá permitir o cadastro de alunos e funcionários, bem como alteração ou exclusão dos mesmos. Deve permitir somente aos instrutores, por meio de senha, cadastrar séries de exercícios aos alunos. Deverá permitir ao aluno, por meio de senha, o acesso e a impressão de sua ficha com a série de exercícios que deverá executar. O sistema deverá também permitir a liberação da catraca, somente a usuários devidamente cadastrados e em dia, no caso de alunos, com a mensalidade da academia.

3.2 ESPECIFICAÇÃO

A especificação foi dividida em quatro partes, sendo primeiro a especificação do circuito eletrônico do protótipo, após, a especificação do software para o microcontrolador, a especificação do software para o PC e por último, a especificação do protocolo de

comunicação.

3.2.1 Especificação do circuito eletrônico

Os desenhos nos apêndices A e B mostram a especificação do esquema eletrônico do protótipo. Foi executado com o auxílio da ferramenta para confecção de desenhos para eletrônica OrCAD Capture 9.00. O apêndice A mostra todos os componentes utilizados para viabilizar o perfeito funcionamento do microcontrolador e a comunicação com o padrão RS485, além da conversão do padrão RS232 para nível *Transistor Transistor Logic* (TTL).

No detalhe “A” os componentes U1 (MAX232) e U2 (SN75176) fazem a conversão dos sinais padrão RS232 do PC para o padrão RS485 e vice-versa, que trafegam pela rede RS485.

O componente U3 (SN75176) faz a conversão dos sinais entre a rede RS485 e o microcontrolador, sendo que este último opera em nível TTL.

No detalhe “B”, o componente U4 (MAX 232) faz a conversão do sinal padrão RS232 para o nível TTL para ser interpretado pelo microcontrolador.

No detalhe “C”, o led D1 e o relé (RL1), são acionados pelo pino 20 do microcontrolador, através do transistor Q1. Este relé possui um “contato seco”, com o qual se aciona o solenóide, que por sua vez, libera a catraca.

O led D2, é acionado diretamente pelo pino 21 do microcontrolador, e sua função é simplesmente sinalizar o momento de cadastramento dos cartões.

Como o microcontrolador, ao ser alimentado, executa um *reset* interno, colocando todos seus pinos de saída em nível lógico 1, optou-se por utilizar uma lógica negativa, isto é, acionar as cargas com nível lógico 0. Desta forma, garante-se que essas cargas não sejam acionadas no momento da alimentação do circuito.

No apêndice B, o esquema eletrônico apresenta um display de cristal líquido (DP1), através do qual aparecerá o código gerado pelo cartão e a indicação de catraca liberada. Neste mesmo esquema aparece também, o regulador de tensão LM7805 (U1), com a função de regular a tensão de 5 volts em corrente contínua, para garantir uma alimentação estável para todo o circuito.

3.2.2 Especificação do software para o microcontrolador

A especificação do software para o microcontrolador é mostrada no apêndice C, seu desenvolvimento foi executado com a técnica de linguagem de modelagem *Unified Modeling Language* (UML), através do diagrama de atividades, utilizando a ferramenta Enterprise Architect.

3.2.3 Especificação do software para o PC

A especificação do software para o PC, foi desenvolvida usando a UML, através de três diagramas distintos: diagrama de atividades mostrados nos apêndices D e E, diagrama de seqüência mostrados nos apêndices F e G e o diagrama de estados, apresentado no apêndice H. Para a confecção desse diagramas, utilizou-se a ferramenta Enterprise Architect.

3.2.4 Especificação do protocolo de comunicação

O protocolo de comunicação entre o PC e a catraca foi baseado no protocolo orientado a caracter. O diagrama de atividades da figura 4 apresenta sua estrutura. Ele é bastante simples, possui o formato mostrado na figura 5.

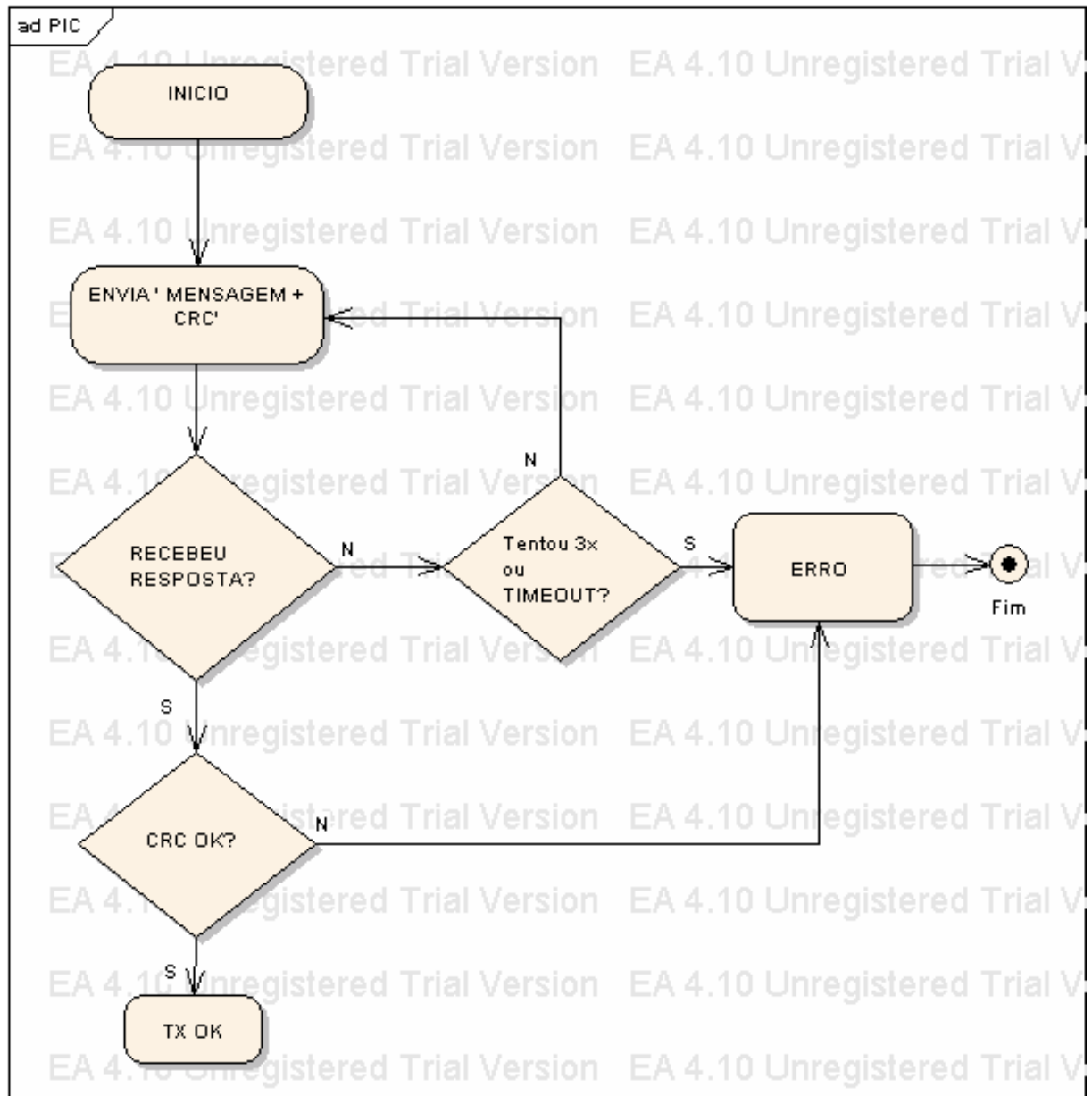


Figura 4 – Diagrama de atividades do protocolo de comunicação

SOH	ENDER_C	FUNCAO	CRC
-----	---------	--------	-----

Figura 5 – Formato do protocolo de comunicação

Nesse formato a palavra “SOH” representa o início da comunicação, “ENDER_C” representa o endereço da catraca, “FUNCAO” representa o código que indicará ao microcontrolador a função que deverá executar. Por último, “CRC” são dois caracteres para verificação de erro nos dados transmitidos. O Quadro 2 a seguir mostra os caracteres e suas respectivas funções.

Caracter	Código (Hex)	Função
SOH	01	Indica início de transmissão
SYN	16	Estabelece comunicação com a catraca.
STX	02	Solicita transmissão do código do cartão
ACK	06	Sinal de reconhecimento (catraca pode ser liberada)
NACK	15	Sinal de não reconhecimento(catraca não pode ser liberada)
CAD	0E	Solicita passagem do cartão para seu cadastro
FCAD	0F	Indica fim de cadastro

Quadro 2 – Caracteres utilizados no protocolo de comunicação do protótipo

3.3 IMPLEMENTAÇÃO

Nesta seção serão apresentados os principais componentes e as ferramentas utilizadas para a implementação do protótipo.

3.3.1 Técnicas e ferramentas utilizadas

Nesta seção serão apresentadas as técnicas e as ferramentas utilizadas para a construção do protótipo. Será dividido em partes, sendo que a primeira mostrará o circuito leitor de código de barras, a segunda apresentará o protótipo do circuito eletrônico contendo o microcontrolador, a comunicação padrão RS485 e o acionamento do solenóide da trava eletromecânica, a terceira e última parte apresentará a implementação do software para o microcontrolador.

3.3.1.1 Leitor do código de barras

O circuito leitor de código de barras utilizado, desenvolvido pela empresa TECPOINT é mostrado na figura 6. Ele é fixado na catraca eletromecânica, de forma a permitir que o usuário passe seu cartão com o código de barras pela canaleta, na qual está fixado o leitor óptico do circuito. Possui ainda um *buzzer*, que gera um *beep* quando ocorre um erro na leitura do cartão e um *led* bicolor que quando verde, indica a captura do código com sucesso e quando vermelho, indica erro na leitura.

O circuito possui também, um conector, cujo cabo é conectado ao circuito eletrônico e pelo qual recebe uma tensão contínua de 5 volts para sua alimentação. Por esse cabo, também, em outros dois fios, trafegam os dados para a comunicação com o microcontrolador.

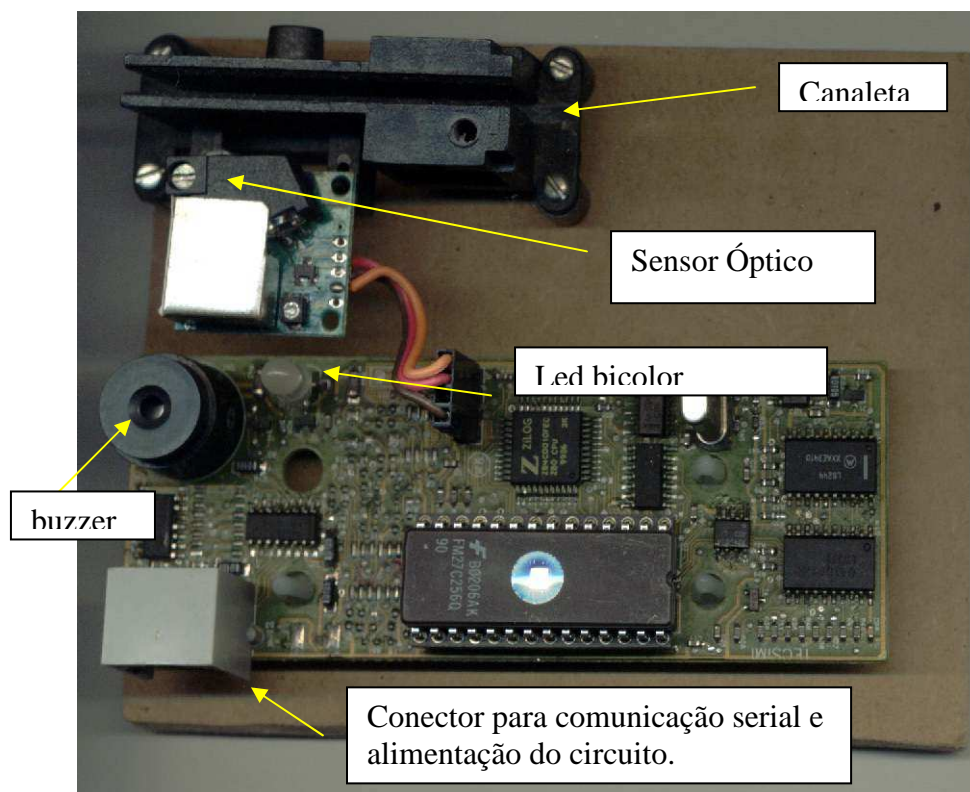


Figura 6 – Circuito do leitor de código de barras da TECPOINT

3.3.1.2 Implementação do protótipo do circuito eletrônico

O protótipo desenvolvido é apresentado na figura 7, foi montado de acordo com a especificação feita na seção 3.2.1. Sua montagem foi realizada fixando todos os componentes do circuito em uma placa universal, ou seja, uma placa de circuito impresso que permite a soldagem dos componentes diretamente na placa, e a interligação de seus pinos podem ser realizados por meio de fios de cobre bem finos, conhecidos por *wire-up*.

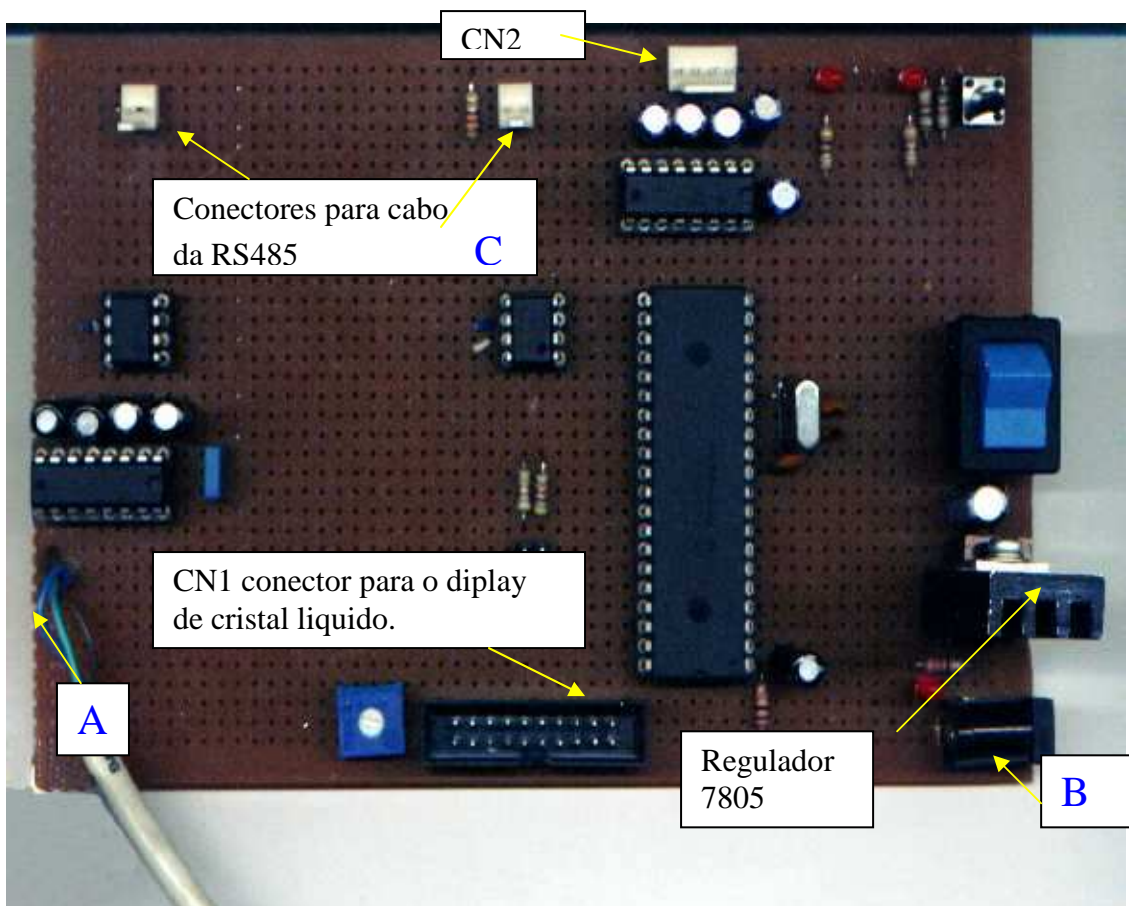


Figura 7 – Protótipo do circuito eletrônico confeccionado em uma placa universal

O conector CN2 possui quatro pinos para conexão do leitor de código de barras. O detalhe A mostra um cabo com três fios, utilizado para estabelecer a comunicação serial com o PC. O detalhe “B” mostra o conector para entrada de alimentação de 12 volts, que é posteriormente regulado para 5 volts através de um regulador de tensão LM7805, que alimenta todo o circuito. No detalhe C são mostrados dois conectores, nos quais é ligado o

cabo para a comunicação serial entre as catracas e o PC. Esta implementação foi executada apenas com o propósito de demonstração. Em uma aplicação real, como as catracas ficam normalmente distantes alguns metros do PC, o conversor de RS485 para RS232 (detalhe “A” no apêndice A) deveria ser montado próximo ao PC, e o restante do circuito próximo a catraca. A figura 7 destaca ainda, um conector para a conexão do display de cristal líquido (CN1). Esse display é mostrado na figura 8.

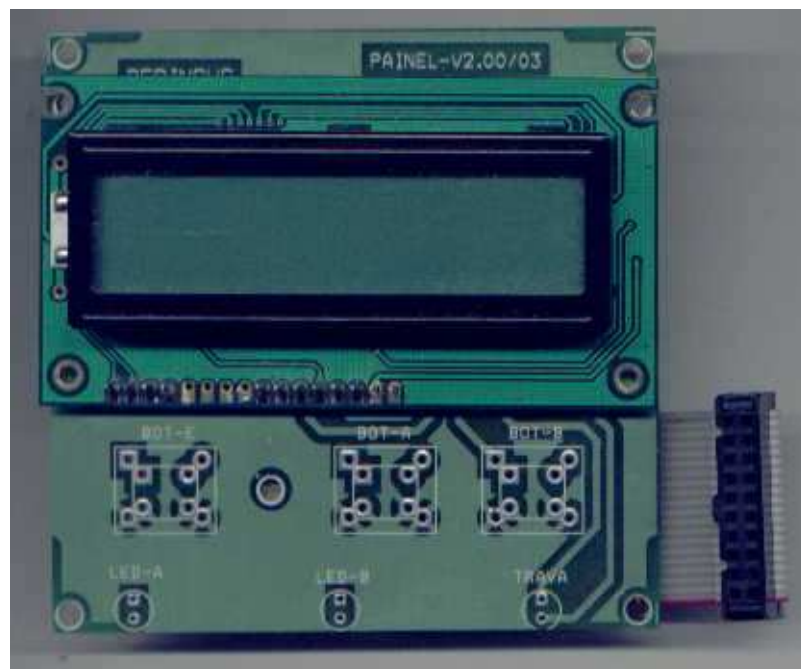


Figura 8 – Display alfanumérico

3.3.1.3 Implementação do software para o microcontrolador

O software foi implementado, seguindo a especificação feita na seção 3.2.2. Este software foi desenvolvido em linguagem Basic para microcontroladores, para tanto, utilizou-se a ferramenta PicBasic Compiler, compilador desenvolvido pela empresa MICROENGINEERING LABS, INC (MICROENGINEERING LABS, 2005) integrado ao

ambiente de programação MPLAB IDE v7.20 da empresa MICROCHIP (MICROCHIP, 2004).

O software tem por finalidade detectar a passagem do cartão, armazenar o código gerado por este cartão e enviá-lo para o PC, quando este requisitar. Tem ainda a atribuição de acionar um relé elétrico por um determinado tempo, afim de possibilitar o giro da catraca eletromecânica.

O protótipo possui duas portas para comunicação serial, sendo uma constituída pelo hardware interno do microcontrolador e a outra implementada por software, que se utiliza dos pinos PORTC.0 e PORTC.1 como pinos para transmissão e recepção, respectivamente.

Ao canal serial, formado pelo hardware interno do microcontrolador que utiliza os pinos PORTC.6 (TX) e PORTC.7 (RX) está conectado o leitor de código de barras. A chegada de um dado por este canal gera uma interrupção, e a rotina a ela associada é imediatamente executada.

Como no microcontrolador PIC quando ocorre uma interrupção, o programa sempre desvia para o mesmo endereço, e nesta implementação foram utilizadas duas interrupções, sendo uma pela USART e outra pelo TIMER1, faz-se necessário então, identificar qual fonte gerou a interrupção, para que seja feito o seu tratamento adequado.

Utilizou-se a interrupção pela USART, para que, quando o cartão for passado pelo leitor, seu código completo seja imediatamente armazenado no microcontrolador, sem correr o risco do código ser perdido total ou parcialmente por estar o microcontrolador ocupado com outra tarefa. A interrupção pelo TIMER1, ocorre ao final da passagem do cartão, para determinar o tempo em que o código gerado por este cartão permanecerá disponível na memória do microcontrolador. Após este tempo, este código é zerado.

Inicialmente, são configurados alguns parâmetros do microcontrolador para seu funcionamento. O quadro 3 a seguir mostra essa configuração.

RCSTA = \$90	' habilita recepcao USART
TXSTA = \$24	' Seta transmissaoUSART
SPBRG =129	' taxa de transmissao 9600 baunds
TRISD= %00000000	' configura todos pinos da PORTD como saida

Quadro 3– Configuração de registros do microcontrolador

O programa inicia desviando para a rotina principal, para evitar o endereço da rotina de interrupção. O quadro 4 mostra um esquema desse desvio.

Goto START
Rotina de interrupção;
START:
Rotina principal;
End.

Quadro 4 – Desvio para rotina principal

Na rotina principal, mostrada no quadro 5, são inicializados os conteúdos das variáveis, configuradas as interrupções, e mantido as porta de saída em nível alto, para que as cargas, isto é, os leds e o relé não sejam acionados. Em seguida a rotina fica no aguardo da chegada de dados pelo canal serial.

index_in = 0	; zera índice para vetor do código cartão
index_out = 0	; zera índice auxiliar.
INTCON = %11000000	; hab. Interrupções
PIE1.5 =1	; liga interrupção USART
GOSUB ZERAR	; zera vetor código do cartão.
HIGH TRAVA	; desliga solenóide da catraca.
HIGH LED	; apaga led
TEMPO_LED=0	; inicializa contador de tempo led/solenóide ligados
TMR1L=0	; zera registrador do Timer1/baixo
TMR1H=0	; zera registrador do Timer1/alto
T1CON=%00110001	; habilita interrupção timer1
LCDout \$fe,1	; limpa display
low PIR1.0	; zera flag de interrupção do timer1
DISPLAY:	
B2= ACK	; carrega B2 com ACK(6)
PASSO_1:	
.....	
SERIN2 PORTC.1\PORTC.0,84,500,PASSO_1,[B0,B1,B3,CRC16A_H,CRC16A_L]	; aguarda a chegada de dados pela serial.

Quadro 5 – Rotina principal para o programa do microcontrolador

A função SERIN2 tem como função a recepção de dados pelo canal serial. Ela

configura os pinos PORTC.1 e PORTC.0 como pinos de transmissão e recepção, respectivamente, para o caso mostrado no quadro 5. Tem ainda, como particularidade, a execução de *timeout*. Assim sendo, na instrução implementada na rotina principal, se no tempo estipulado, no caso específico, 500 milissegundos, não houver a chegada de um dado pelo canal de recepção, ocorrerá o *timeout* e o programa passará a ser executado a partir do endereço indicado, no caso “PASSO_1”. Havendo a chegada de dados pela recepção, a função SERIN2, os armazena nas variáveis passadas como parâmetro e o programa passa a executar a instrução seguinte.

O quadro 6, apresenta a seqüência do programa.

```

if( CRC16A_H <>CRC16A_HIGH or CRC16A_L<>CRC16A_LOW) then
; se cálculo do CRC for igual ao enviado pelo PC

    if (B0=SOH and B1=ENDER_1 and B3=SYN) then
        ..
    if (B0=SOH and B1=ENDER_1 and B3=STX) then
        ..
    if (B0=SOH and B1=ENDER_1 and B3=ACK) then
        ..
    if (B0=SOH and B1=ENDER_1 and B3=NACK) then
        ..
    if (B0=SOH and B1=ENDER_1 and B3=FALHA) then
        ..

```

Quadro 6 – Seqüência de execução da rotina principal

No trecho acima, são implementadas as funções para tratamento dos dados chegados na recepção serial. As instruções respeitam o protocolo de comunicação especificado no item 3.2.4 deste documento. Basicamente respondem às solicitações enviadas pelo PC, se devidamente reconhecido o endereço da catraca.

Conforme mencionado anteriormente, a comunicação entre o leitor de código de barras e o microcontrolador, geram uma interrupção pela USART, a interrupção pode, porém, ocorrer pelo TIMER1. No trecho de código do quadro 7 é apresentado o início da rotina executada quando ocorre uma interrupção e como é identificada a fonte que a gerou. Algumas

instruções da rotina foram implementadas em *Assembly*, o que é perfeitamente permitido pelo PICBASIC.

Asm		; inicia um programa em assembly
InterrupcaoUSART		; endereço de interrupcao
movwf wsave		; salva registrador W
swapf STATUS, W		; alterna nibbles de STATUS
		; e armazena em W
clrf STATUS		; limpa STATUS
movwf ssave		; salva W em ssave
movf PCLATH, W		; Move PCLATH para W
movwf psave		; salva PCLATH
movf FSR, W		
movwf fsave		; salva FSR
btfsc PIR1,0		; se houve interrupcao por timer
goto TEMPO		; desvia para TEMPO
INT_USART:		; senão interrupção por USART
Endasm		; fim de instruções assembly

Quadro 7 – Início da rotina de interrupção

A cada caracter que chega à porta de recepção serial do microcontrolador, enviado pelo leitor de código de barras, é gerada uma interrupção e o dado deve ser então armazenado. Para o armazenamento do código, constituído por uma seqüência de 16 caracteres enviados pelo leitor, criou-se uma variável do tipo fila. A cada interrupção gerada um caracter é armazenado em uma posição desta variável. O quadro 8 a seguir, mostra o trecho do código que executa esta função.

```

asm
incf _index_in, F ; incrementa index_in
movf _index_in, W ; salva em W
sublw _buffer_size-1 ; verifica se index_in
btfss STATUS,C ; é igual ou maior que o tamanho
; do vetor
clrf _index_in ; ser for, zera index_in

movlw Low _buffer ; pega o endereço da primeira
; posição do vetor buffer[0]
addwf _index_in, W ; soma o index_in a este endereço
movwf FSR ; e o salva no registro FSR

movf RCREG, W ; armazena em W o caracter rece-
; bido pela serial
movwf INDF ; armazena na posição de memória
; apontada por FSR
endasm ; fim de assembly
INTCON = 0 ; desabilita interrupções
index_out=index_out+1 ; incrementa index_out

IF (index_out => buffer_size) Then index_in = 0
index_out = 0 ; limita tamanho do índice
endif

vetor[index_out-1]=buffer[index_out]; salva no vetor
; auxiliar o caracter
.

```

Quadro 8 – Sequência de execução da rotina de interrupção

Finalmente, antes que a rotina de interrupção seja finalizada, os registradores inicialmente salvos, devem ter seus valores restaurados para os valores que continham no momento em que ocorreu a interrupção. O quadro 9 mostra essa situação.

```

asm
SAIDA:
movf fsave, W ;recupera valor de FSR
movwf FSR ;carrega em FSR
movf psave, W ; recupera valor de PCLATCH
movwf PCLATH ; carrega em PCLATCH
swapf ssave, W ; troca nibbles de ssave
movwf STATUS ; recupera valor de STATUS
swapf wsave, F ; troca nibbles de wsave
swapf wsave, W ; troca novamente e salva
; em W ( restaura valor de W)
retfie ; sai da interrupção.
EndAsm

```

Quadro 9 – Recuperação dos valores dos registradores

3.3.1.4 Implementação do software para o PC

O software para controle e cadastro dos usuários, foi desenvolvido de acordo com a especificação apresentada no item 3.2.3. Foi desenvolvido em ambiente de desenvolvimento Delphi 7.0. Para base de dados utilizou-se o banco de dados Interbase 6.0.

Além de todos os componentes, já comuns no Delphi 7.0, para construir a interface de operação para o usuário e o acesso ao banco de dados, foi necessário utilizar um novo componente, não pertencente originalmente ao Delphi 7.0. Este componente é o `ApdCompPort1` (TECPOINT, 2004), utilizado para estabelecer a comunicação serial entre o PC e o microcontrolador. Ele oferece todo o suporte necessário para realizar a comunicação serial, com um equipamento externo, inclusive o controle do sinal *Request to Send* (RTS), utilizado para o controle de sentido de transmissão de dados necessário para emprego do padrão RS485. Ele contém ainda, parâmetros que devem ser configurados, para o perfeito funcionamento da comunicação. O quadro 10 mostra as instruções que configuram esses parâmetros e seus valores adotados nesse projeto.

```
.....  
Comm.ComNumber:=1; // define porta de comunicação COMM1  
Comm.Baud:=9600;   // taxa de transmissão 9600 bauds  
Comm.Parity:=pNone; // sem paridade  
Comm.DataBits:=8; // dados com 8 bits  
Comm.StopBits:=1; // possui 1 stop bit  
.....
```

Quadro 10 – Configuração dos parâmetros de comunicação

O componente `ApdCompPort1`, força a execução de uma rotina, sempre que um caracter chega à porta de recepção do PC. O quadro 11 apresenta essa rotina.

```

procedure TfPrincipal.CommTriggerAvail(CP: TObject; Count: Word);
var
  I : Word;
  C : char;

Begin

  for I := 1 to Count do      // de 1 até o número de caracteres que chega na
                             // recepção
  begin
    C := Comm.GetChar; // armazena em C o caracter recebido
    memo1.Lines.Text:=memo1.Lines.Text+C;// acumula caracteres
    codigo:= memo1.Lines.Text; // salva na variável global codigo.
  end;

```

Quadro 11 – Rotina de tratamento do dado da porta serial

Para transmissão de dados utiliza-se a estrutura mostrada no quadro 12.

```

Transmite(SOH,slave,dado,CRC16A_HIGH,CRC16A_LOW); //
// chama rotina para transmissão.
.
procedure TfPrincipal.Transmite(const Posi1: byte;Posi2: byte;Posi3: byte;Posi4:
byte;Posi5: byte);
begin
  Comm.Output:=char(Posi1); // envia valor passado através de Posi1
  Comm.Output:=char(Posi2); // envia valor passado através de Posi2
  Comm.Output:=char(Posi3); // envia valor passado através de Posi3
  Comm.Output:=char(Posi4); // envia valor passado através de Posi4
  Comm.Output:=char(Posi5); // envia valor passado através de Posi5
end;

```

Quadro 12 – Rotina de transmissão de dados

O Quadro 13 a seguir, mostra a rotina para o cálculo do CRC. Ela é utilizada tanto no software do microcontrolador, quanto no software para o PC. É um CRC de 16 *bits*, ou seja, o cálculo de um ou mais caracteres sempre resultará em dois valores distintos de 8 *bits* cada. Esta rotina, foi baseada em uma rotina desenvolvida em Pascal por VIGNEAULT (2004).

```

CALCULA_CRC:
    for I=0 to TAMANHO_DADO ; para calcular o crc dos caracteres
    vetor_crc[I]=vetor[I] ; armazena dados num vetor auxiliar
    Next I
CALCULA_CRC2:
    CRC=0
    FOR I=0 to TAMANHO_DADO
    TEMP =vetor_crc[I] * $100 ; multiplica valor por 256
    CRC = (CRC ^ TEMP) ; OU Exclusivo entre CRC e TEMP
    For J = 0 To 7 ; para cada bit
    TEST=(CRC & $8000) ; rotaciona byte
    If (CRC & $8000)>0 Then ; se for igual a 1
    CRC = ((CRC * 2) ^ $1021) & $FFFF ; executa equação 1
    Else ; senão
    CRC = (CRC * 2) & $FFFF ; executa equação 2
    EndIf
    Next J
    Next I
    CRC16A =(CRC & $FFFF)
    CRC16A_HIGH=CRC16A/256 ; byte mais significativo do crc
    CRC16A_LOW=CRC16A//256 ; byte menos significativo do crc
RETURN

```

Quadro 13 – Rotina para cálculo do CRC

3.3.1.5 Testes e validação

Ao longo do desenvolvimento do protótipo foram realizados testes de funcionamento do sistema. Foram passados pelo leitor de código de barras, diversos cartões, alguns cadastrados e outros não, para verificar se atendiam aos requisitos especificados. E também foram realizados os testes para o cadastro de novos cartões. O protótipo funcionou a contento, o tempo de resposta entre a passagem do cartão e a liberação de passagem, para apenas uma catraca conectada foi inferior a 1 segundo. Os cadastros foram realizados com sucesso, e ao final foram impressas algumas fichas de séries de exercícios anteriormente cadastradas pelo instrutor.

3.3.2 Operacionalidade da implementação

Quando o sistema é executado, é exibida a tela principal mostrado na figura 9. Inicialmente o sistema faz a conexão com as catracas, marcando as que estão habilitadas. Após isso permite ao usuário digitar seu login e sua senha.



Figura 9 – Tela inicial

Após a digitação, se for pressionado o botão cancelar, o sistema volta à condição inicial. Pressionando o botão “OK”, estando o usuário cadastrado o sistema verifica seu *status*, caso contrário aparecerá a mensagem “Usuário não cadastrado”. Sendo o usuário, cadastrado como instrutor, aparecerá então a tela mostrada na figura 9, porém, sem a tela menor, “Login para acesso a academia”. Pressionando o botão “Cadastrar”, surgirá a tela mostrada na figura 10, na qual, o instrutor ou administrador pode cadastrar um novo instrutor, um novo aluno ou apenas alterar seus dados já existentes. Por meio da caixa de diálogo “STATUS” pode-se definir qual o *status* do usuário; instrutor ou aluno. Ao ser definido seu *status*, apenas os atributos pertinentes a esse tipo de usuário serão habilitados, permitindo

assim sua digitação. Ao final, pressionando o botão “OK”, aparecerá uma caixa de mensagem solicitando a passagem do cartão pela catraca, para cadastro de seu código. Após a passagem do cartão, aparecerá uma caixa de mensagem informando se o cadastro foi executado com sucesso ou não.

Cadastro Aluno/Funcionário

Status:

Nome: CPF:

Endereço:

Bairro: Cidade:

Estado Civil: Data de Nascimento: Profissão:

Fone Residencial: Fone Comercial: Celular:

E-mail: Dia de Vencimento: Mensalidade:

Login: Início das Atividades: Cargo:

Senha: Cartão:

Objetivo:

Observações:

NOVO

EDITAR

Anterior

Proximo

GRAVAR

SAIR

Figura 10 – Tela de cadastro de alunos e instrutores

Pela tela da figura 9 ainda, sendo pressionado o botão “Cadastrar/ Série de exercícios” aparecerá a tela da figura 11. Esta tela permite selecionar o nome do aluno, para o qual se pretende cadastrar sua série de exercícios.

The image shows a software window titled "Consulta Aluno". At the top, there is a search field labeled "Nome:". Below it is a table with two columns: "CPF" and "NOME". The table contains four rows of data. At the bottom of the window, there are two buttons: "OK" and "SAIR".

CPF	NOME
33.333.333-33	MARIA MATARRARY
11.111.111-11	DIANA DA SILVA
22.222.222-22	JOÃO DE ALBUQUERQUE
44.444.444-44	CLEMENTINA DE JESUS

Figura 11 – Tela para seleção do aluno para cadastro de série de exercícios

Escolhido o nome do aluno, com o uso do *mouse*, pela barra de rolagem e pressionando o botão “OK”, aparecerá a tela da figura 12. Nela aparecerá o nome do aluno, além de outros de seus atributos. Pressionando com o *mouse* sobre uma figura, ela é habilitada, permitindo também que os campos, logo abaixo, sejam preenchidos com o peso que o aluno deve utilizar. O instrutor pode escolher diversas figuras para formar toda a série de exercícios que o aluno deverá executar.

Séries de Exercícios

CPF.: 222.222.222-22 JOÃO DE ALBUQUERQUE

Séries.: 8 series de 10 minutos Aquecimento.: 10 minutos

Data Ini.: 01/03/2005 Data Fin.:

01-Crucif. 30° 1º Grupo Pectorais	02-Supino 30°	03-Peck Deck	04-Cross-Over	05-Supino Máquina	06-Supino Declinado	07-Crucifixo reto
08-Puxador Costas 2º Grupo Dorsal	09-Puxador Frente	10-Puxador Fechado	11-Remador	12-Rem. Uni-Lateral	13-Puxador Máquina	14-Barra Fixa
15-Hack 3º Grp. Membros Inf	16-Leg-Press 45°	17-Leg-Press Reto	18-Extensor Coxa	19-Flexora Coxa	20-Flexora Unilateral	21-Cad Adutora/Abdu
22-Glúteos Máquina	23-Stiff	24-Panturrilha em Pé	25-Panturrilha Sentad	26-Remada Alta 4º Grupo Ombros	27-Desenv. Halteres	28-Elevação Lateral
29-Elevação Frontal	30-Rosca Scott 5º Grp. Bí./Triceps	31-Rosca Direta	32-Rosca Banco 30°	33-Rosca Alternada	34-Tríceps Teste	35-Tríceps Francês
36-Tríceps Puxador	Nova Serie		Consulta		GRAVAR	
	Editar		Imprimir		SAIR	

Figura 12 – Tela para cadastro de série de exercícios

Para gravar a série, basta pressionar o botão “GRAVAR”, que a série será cadastrada. Para fechar esta tela, basta pressionar o botão “SAIR”.

O instrutor poderá ainda escolher a opção “Mensalidade”. Pressionando esta opção aparecerá a tela da figura 13. De maneira idêntica à tela da figura 11, pode ser escolhido o aluno cuja mensalidade se deseja quitar. Após selecionado o aluno, basta marcar a opção

“mensalidade paga” com o *mouse* e pressionar o botão “CONFIRMA”.

The screenshot shows a window titled "Consulta Aluno - Mensalidades". At the top, there is a search field labeled "Nome:" containing the text "MARIA MATARRARY". Below this is a table with the following data:

CPF	NOME	Em Dia
33.333.333-33	MARIA MATARRARY	NÃO
11.111.111-11	DIANA DA SILVA	
22.222.222-22	JOÃO DE ALBUQUERQUE	SIM
44.444.444-44	CLEMENTINA DE JESUS	SIM

At the bottom of the window, there is a checkbox labeled "MENSALIDADE PAGA" which is currently unchecked, and a green button labeled "CONFIRMA".

Figura 13 – Tela para registro de pagamento de mensalidade

Se na tela inicial (figura 9) o usuário possuir o *status* de aluno, ao apertar o botão “OK”, aparecerá uma tela idêntica a da figura 12. Mostrando habilitados somente os exercícios previamente cadastrados pelo instrutor, o botão “IMPRIMIR”, para realizar a impressão da ficha de exercícios e o botão “CONSULTA”, para a consulta da série realizada anteriormente.

Por fim, após a passagem do cartão, poderá surgir no display, junto à catraca, a frase “ERRO NA LEITURA DO CARTÃO”, caso ocorra algum erro na leitura do cartão, situação que provocará também um aviso sonoro.

Ao passar o cartão pelo leitor, se o aluno estiver com a mensalidade atrasada, mas com um prazo inferior a 10 dias, aparecerá no display a frase “Falar com a Administração” e a catraca será liberada. Se o tempo de atraso, entretanto, for superior a 10 dias, a mesma frase será apresentada, mas a catraca não será liberada, impedindo dessa forma, o acesso do aluno.

3.4 RESULTADOS E DISCUSSÃO

Este trabalho atingiu seu objetivo de realizar o cadastro e controle de acesso de usuários em uma academia de ginástica. Foi implementado de forma a permitir a conexão para até três catracas simultaneamente, o que foi considerado, em discussão com o orientador, um proprietário de academia e um instrutor, um número suficiente para atender academias de pequeno e médio porte. Porém, esse número pode ser aumentado, implicando é claro, em um aumento do tempo de espera de resposta para liberação da catraca.

Foram realizados diversos testes com o protótipo. Entretanto, como havia somente um leitor de código de barras, devido a seu custo, foi necessário simular dois outros leitores. Para essa simulação foi ligado em duas portas do microcontrolador dois *jumpers*. Ao iniciar o processamento, o microcontrolador lê o estado dos *jumpers*, e com isso, forma o endereço da catraca. Infelizmente não foi possível a utilização das catracas eletromecânicas para executar os testes, também por conta de seu custo.

Inicialmente, o objetivo era implementar todo o software para o microcontrolador em Basic. Porém, como era necessário utilizar dois canais de comunicação, sendo que em um dos quais deveria gerar interrupção, as instruções em Basic não funcionaram adequadamente, o que forçaram seu desenvolvimento em Assembly dentro do próprio Basic.

O componente *ApdCompPort1*, apresentou problemas quanto ao controle do pino RTS, não alternando seu nível lógico de saída corretamente, o que não permitiu o perfeito funcionamento dos componentes U2 e U3 (apêndice A).

Para o desenvolvimento do protótipo, foram pesquisados alguns trabalhos correlatos, como a utilização do padrão RS485 e do PIC16F877A. Esses trabalhos, porém, tinham objetivos diferentes, de forma que não é possível tentar estabelecer comparações entre esses trabalhos e o aqui exposto. Outros trabalhos consultados, com o objetivo de tomar

conhecimento da forma de gerenciamento de uma academia, foram executados fora do ambiente acadêmico, sendo que esses, de cunho comercial, apresentavam interfaces complexas com o objetivo de atender a todas as necessidades de uma academia de ginástica como tabelas de nutrição, cadastro de laudos médicos, diversas modalidades de atividades físicas, entre outros. Como não era o objetivo deste trabalho, oferecer uma interface tão complexa, foram tiradas apenas informações suficientes para implementar uma interface simples e objetiva. O que também não permite uma comparação entre esses trabalhos.

4 CONCLUSÕES

Através deste trabalho foi possível explorar bem o funcionamento de protocolos de comunicação, microcontroladores e comunicação serial entre vários equipamentos distintos. Foi possível verificar a versatilidade dos microcontroladores e como estes componentes tornam possível a implantação de sistemas compactos e a um baixo custo, para a resolução de problemas de automação relativamente complexos.

Com o desenvolvimento desse trabalho pode-se verificar também o quanto a automação pode beneficiar nossas vidas, tornando algumas tarefas mais cômodas de serem realizadas e de maneira mais eficiente.

Ao longo de seu desenvolvimento, foram utilizadas diversas ferramentas, sem as quais, com toda a certeza, o trabalho teria seu desenvolvimento enormemente dificultado. Com alguma ressalva para o Basic, pelo motivo já mencionado, mas que, entretanto, oferece ele mesmo ferramenta para melhorar seu desempenho, exigindo assim, apenas um pouco mais de trabalho e dedicação por parte do programador. A utilização do ambiente Delphi 7.0 juntamente com o banco de dados Interbase, facilitou muito a construção da interface para o usuário.

A automação é sem dúvida um assunto extremamente interessante, e a interação entre PC e demais equipamentos por meio de comunicação serial deve ser dominado por quem opta por esta área de atuação. Este trabalho serve como um exemplo inicial, como uma base, em cima da qual outros projetos de automação possam ser desenvolvidos.

4.1 EXTENSÕES

Como extensão para esse trabalho sugere-se:

- a utilização de módulos *Rádio Frequency* (RF) operando na faixa de 2.4Ghz, para estabelecer a comunicação entre o PC e as catracas;
- Adição de memória EEPROM para o armazenamento da base de dados, permitindo a realização do controle de acesso sem a utilização do PC.

REFERÊNCIAS BIBLIOGRÁFICAS

AXELSON, Jan. **Serial port complete**: programming and circuits for RS-232 and RS-485 links e networks. 3 th. Madison: Lakeview Research, 2000.

CUENCA, Martin E.; MARTÍNEZ, Ângulo I.; USATEGUI, José M. **Microcontroladores Pic**: la solución em um chip. 2. ed. Madrid: Paraninfo, 1998.

CUNHA, Judson Michel. **Protótipo de rede industrial utilizando padrão serial RS-485 e protocolo MODBUS**. 2000. 119 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CANZIAN, Edmur. Comunicação Serial – RS232. **Saber Eletrônica**, São Paulo, n.4, 2001. 1 CD-ROM.

GROSSMAN, Fábio; ZYNGIER, Mauro Luiz. **Código de barras**: da teoria à prática. 2. ed. São Paulo: Nobel, 1991.

LOESH, Cláudio; STRINGARI, Sérgio; TAFNER, Malcon Anderson. **Comunicação de dados usando linguagem C**: aplicação em DOS e windows. 1. ed. Blumenau: Editora da Furb, 1996.

MATIAS, Juliano. Entenda o protocolo elétrico RS-485. **Mecatrônica Atual**, São Paulo, n. 4, p. 7, Jun. 2002.

MICROCHIP. **A leading provider of microcontroller & analog semiconductors**. Massachusetts, 2004. Disponível em: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>>. Acesso em: 6 ago. 2005.

MICROENGINEERING LABS, INC. Colorado Springs, 2005. Disponível em: <<http://www.melabs.com>>. Acesso em: 8 ago. 2005.

MIKROELEKTRÔNİKA. **Microcontroladores Pic**. Beograd, 2003. Disponível em: <<http://www.mikroelektronika.co.yu/portuguese/product/books/picbook/capitulo1.htm#Microcontroladores%20versus%20Microprocessadores>>. Acesso em: 12 out 2005.

MORETTI, Alfredo Angelo. **Código de Barras**. 1999. 47 f. Monografia (Pós Graduação em Nível de Especialização em Administração de Materiais) – Instituto Nacional de Pós Graduação, Universidade Regional de Blumenau, Blumenau.

PASSO AUTOMAÇÃO. **Catracas**. Canoas, 2003. Disponível em: <<http://www.passo.com.br/catraca.htm>>. Acesso em: 12 out. 2005.

PEREIRA, Fábio. **Microcontroladores PIC** : programação em c. 3.ed. São Paulo: Érica, 2003.

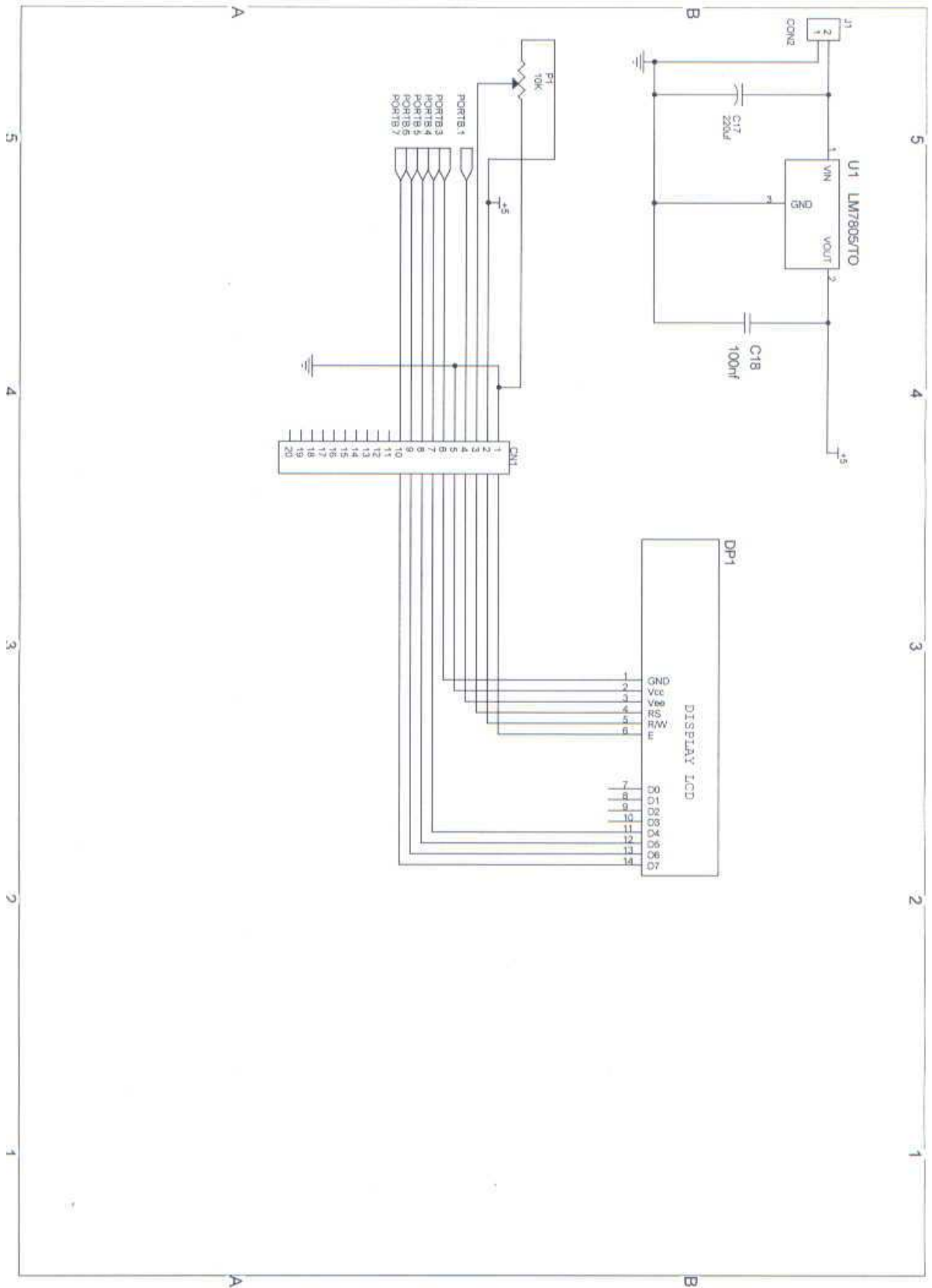
SILVA JÚNIOR, Vidal Pereira da. **Microcontroladores PIC** : teoria e prática. São Paulo: Érica, 1998.

SYSNET. **Auto-gym for Windows**. São Paulo, [2003]. Disponível em:
<<http://www.sysnetweb.com.br>>. Acesso em: 6 ago. 2005.

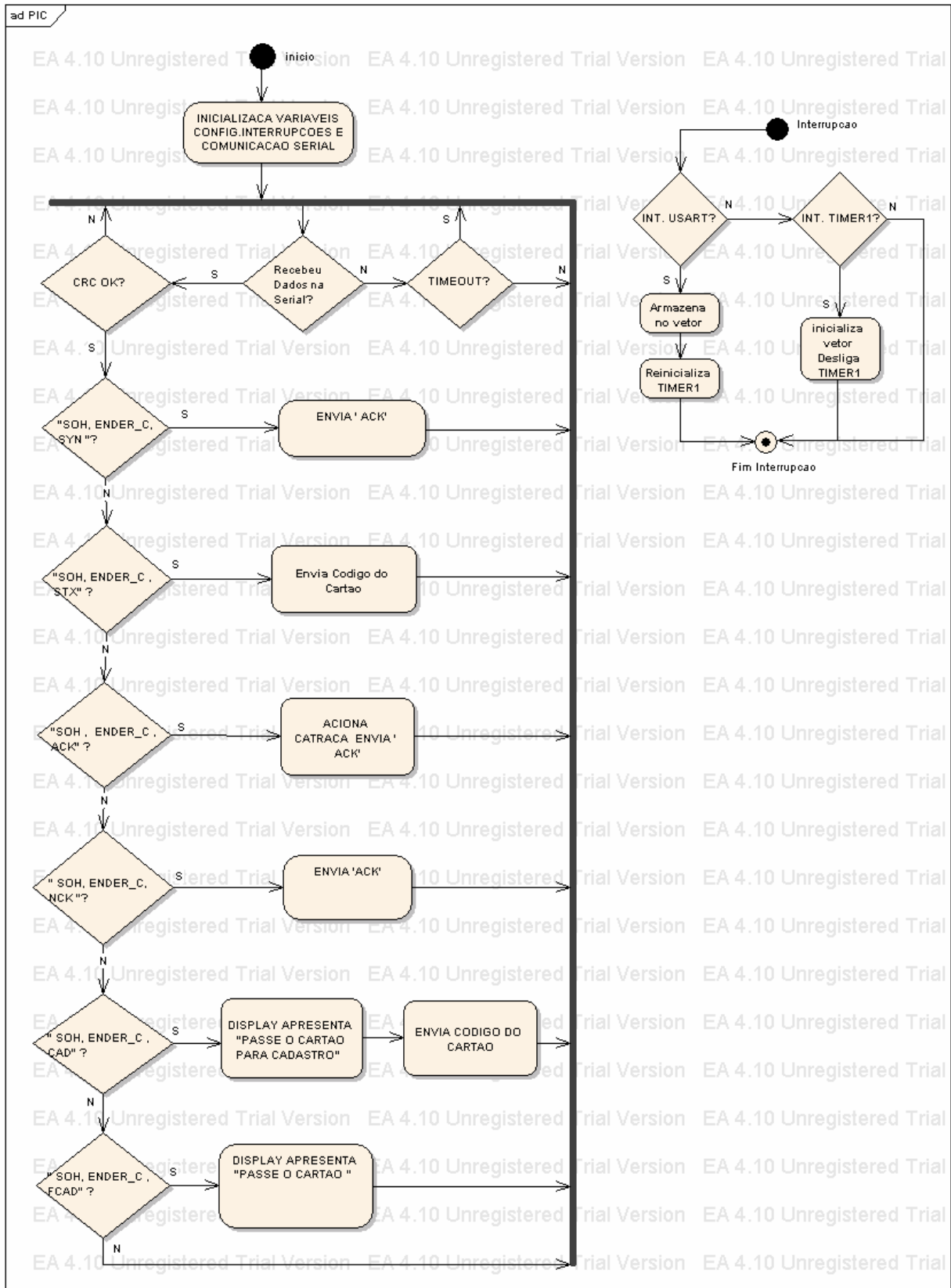
TECPOINT. **Kit barras**. São Paulo, [2004]. Disponível em:
<<http://www.tecpoint.com.br/kitbarras.asp>>. Acesso em: 6 ago. 2005.

VIGNEAULT, Greg. **Ideas for website**. [S.I.], 2004. Disponível em:
<<http://massmind.org/techref/language/delphi/swag/crc0003.html> >. Acesso em: 10 out. 2005.

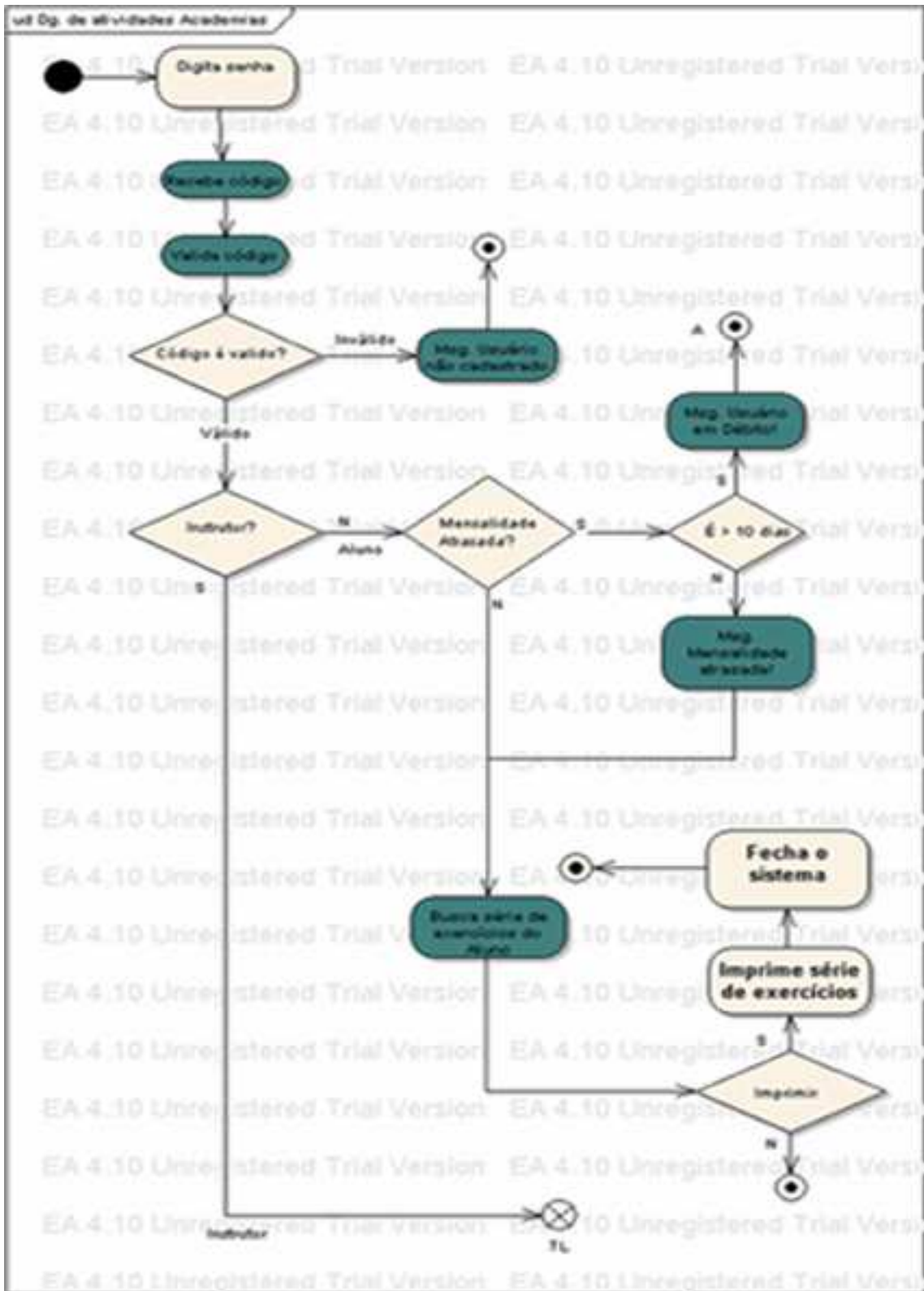
APÊNDICE B – Parte 2 do esquema eletrônico



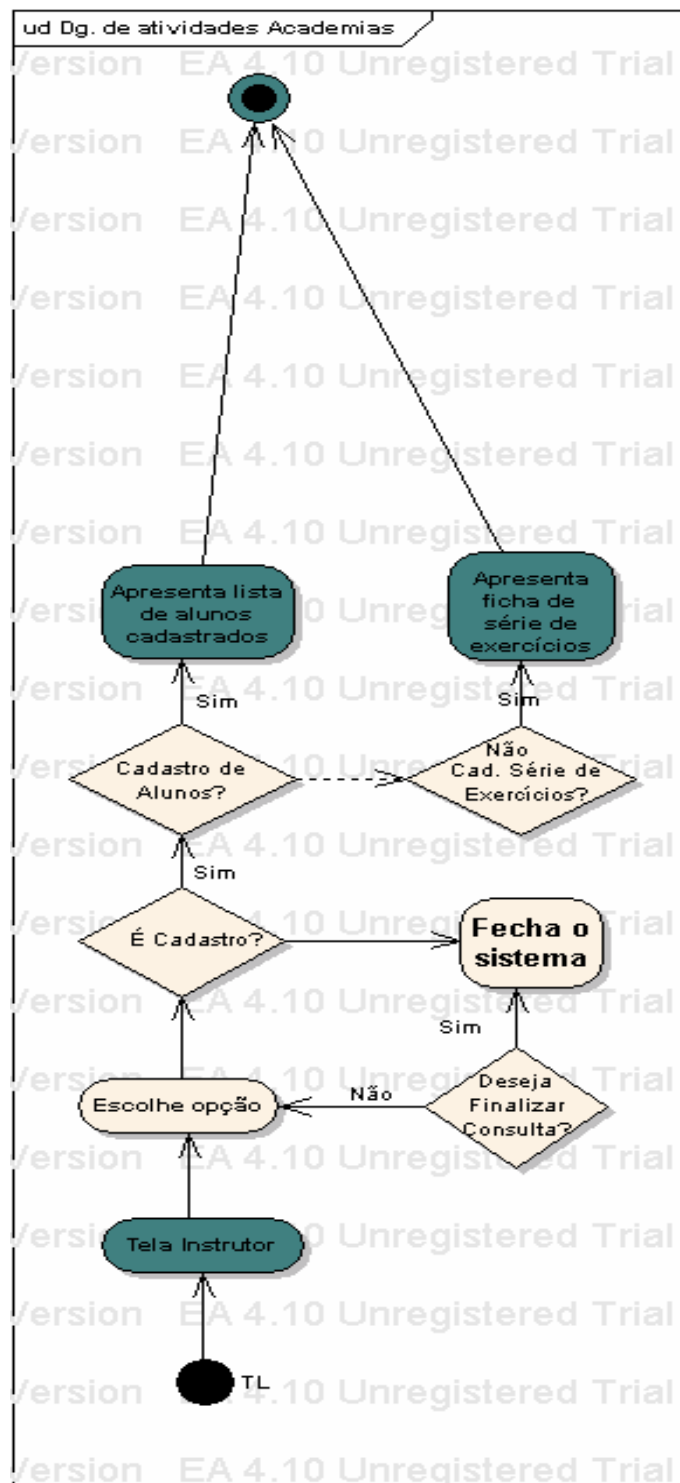
APÊNDICE C – Especificação do Software do microcontrolador



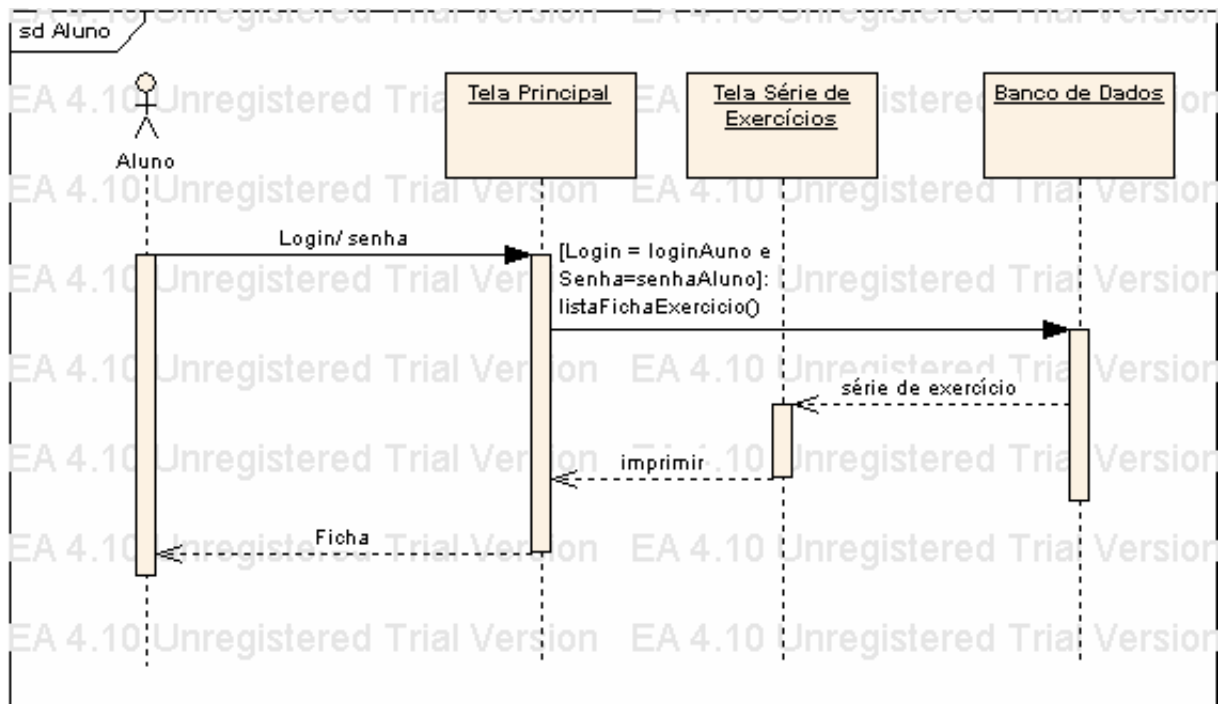
APÊNDICE D - Especificação do Software para o PC (Diagrama de Atividades 1)



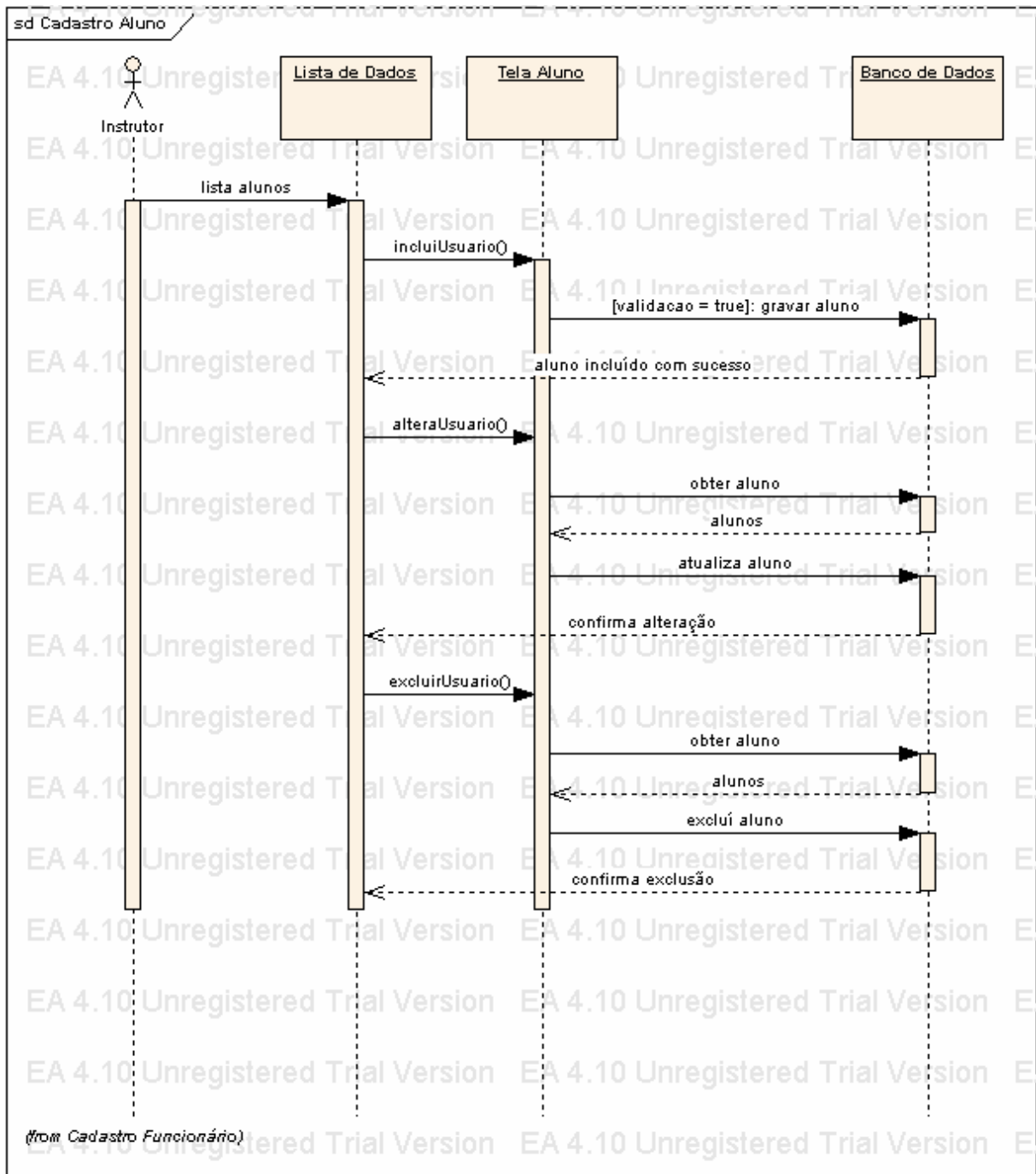
APÊNDICE E - Especificação do Software para o PC (Diagrama de Atividades 2)



APÊNDICE F - Especificação do Software para o PC (Diagrama de Seqüência 1)



APÊNDICE G - Especificação do Software para o PC (Diagrama de Seqüência 2)



APÊNDICE H - Especificação do Software para o PC (Diagrama de Estados)

