



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

RELATÓRIO FINAL – FEVEREIRO 2006

**Desenvolvimento de Sistema de Aquisição de Dados
Meteorológicos**

Orientadora: Prof.a Eliane Aparecida F. Amaral Fadigas

Coordenador: Prof. Dorel Soares Ramos

Aluno: Ricardo Ping de Tung Gabriel - Nº USP: 3313853



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

folha de aprovação

Ricardo Ping de Tung Gabriel
Desenvolvimento de sistema de aquisição de dados.

Projeto de formatura apresentado ao Departamento de Energia e Automação Elétricas da Escola Politécnica da Universidade de São Paulo para a obtenção do grau de Bacharel em Engenharia Elétrica. Área de concentração: Produção de Energia Elétrica.

Aprovado em:

Banca Examinadora

Prof. Dr. _____

Instituição _____ Assinatura: _____

Prof. Dr. _____

Instituição _____ Assinatura: _____

Prof. Dr. _____

Instituição _____ Assinatura: _____

Prof. Dr. _____

Instituição _____ Assinatura: _____

Prof. Dr. _____

Instituição _____ Assinatura: _____



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

Sumário

Introdução	4
Objetivos	8
Detalhamento do hardware	10
Detalhamento do Software	13
Calculo do CRC	17
Estudo Modbus	23
Estudo Normas IEC.....	28
Firmware.....	35
Conclusão	38



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

Índice de equações

Equação 1. Densidade do ar.	29
Equação 2. Média em 10 minutos.	31
Equação 3. Desvio padrão em 10 minutos.	31

Índice de tabelas

Tabela 1. Cálculo do CRC.	18
--------------------------------	----

Índice de figuras

Figura 1. Diagrama de blocos do hardware 1.	12
Figura 5. Arquivo de saída.	20
Figura 6. Tela de apresentação do software.	23



Introdução

O mundo vive um surto de investimento em fontes alternativas de energia, motivada principalmente pela alta nos preços do petróleo e na alta dependência de uma única fonte energética. Algumas medidas estão sendo tomadas com o intuito de diversificar a matriz energética global e diminuir a dependência do petróleo.

Uma delas é o protocolo de Kyoto, segundo o qual os países desenvolvidos têm uma meta de redução na emissão de gases do efeito estufa (principalmente o CO₂) esta redução pode ser feita tanto direta (troca do uso de combustíveis fósseis por fontes renováveis) quanto pela compra de créditos de carbono.

O Brasil também dispõe de um programa de incentivo às fontes alternativas cujo objetivo é a diversificação da matriz energética e também redução de emissão dos gases do efeito estufa. Este programa se chama Proinfa (programa de incentivo às fontes alternativas de energias), e prevê na primeira etapa a instalação de 1 300 MW de energia proveniente dos ventos. Os aerogeradores atuais encontram-se num excelente estado de avanço tecnológico sendo hoje em dia dentre as



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

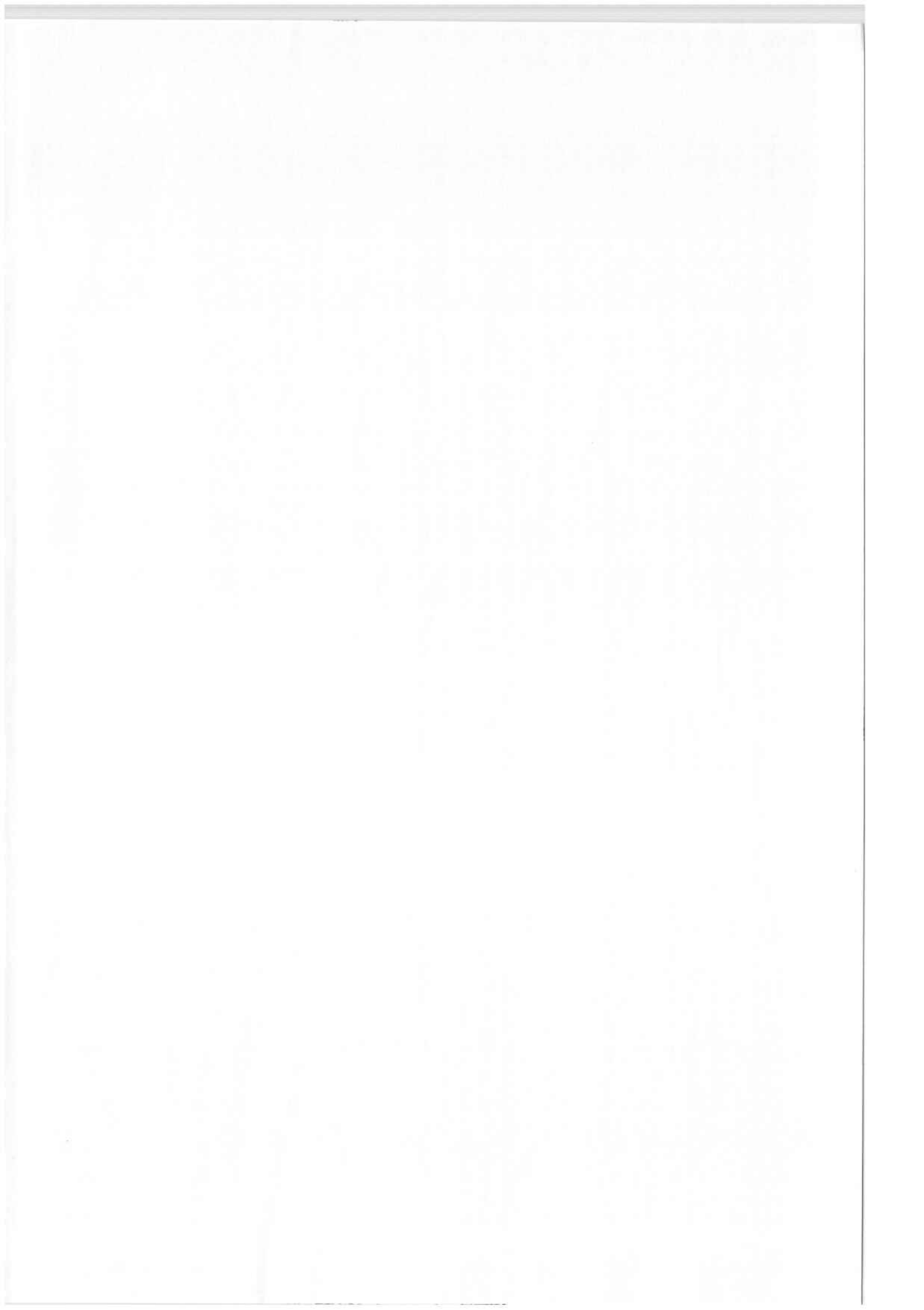
tecnologias alternativas de geração de energia a mais viável tecnologicamente e economicamente.

Apesar de tanto incentivo que existe para a disseminação desta tecnologia não é possível instalar centrais eólicas em qualquer lugar. É necessário um minucioso estudo de regime de ventos conhecido comumente como levantamento de potencial eólico. Este levantamento identifica regiões que oferecem uma elevada velocidade média do vento, pois a energia obtida é proporcional ao cubo da velocidade.

O projeto inicial de uma central eólica de grande porte exige um investimento inicial na compra de equipamentos de medição e contratação de empresas de certificação que garantem o levantamento de potencial eólico dentro de uma precisão necessária ao desenvolvimento do projeto.

Existe um investimento grande nesta e é possível que se chegue a um resultado contrário às expectativas iniciais, ou seja, que o projeto é inviável. O custo deste levantamento torna-se um contra-estímulo ao desenvolvimento dos projetos eólicos. Eles passam, portanto a depender de incentivos governamentais e estudos apenas em locais sabidamente viáveis.

Um barateamento no custo deste levantamento funcionaria como um grande estímulo ao empreendimento de projetos eólicos, pois possibilitaria





ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

a prospecção de áreas com potencial eólico, mas que hoje em dia é desconhecido.

Este projeto se insere neste contexto ao oferecer subsídios para coletar e analisar ventos utilizando um sistema completo de medição.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

Objetivos

O objetivo deste trabalho é desenvolver um sistema de aquisição de dados meteorológicos composto por hardware e software. O hardware (conhecido como datalogger) foi desenvolvido anteriormente no GEPEA como projeto de iniciação científica e consiste num circuito microcontrolado que coleta informações através de entradas analógicas e digitais.

O software, escopo deste projeto, consiste num aplicativo que roda num computador pessoal com sistema operacional Windows e faz a interface entre o datalogger e o usuário.

Este software pode ser entendido como uma interface homem máquina (IHM) uma vez que fornece um modo amigável de coletar dados e configurar o datalogger utilizando funcionalidades gráficas e periféricos comumente utilizadas em PCs.

As funcionalidades principais, que serão detalhadas no decorrer deste trabalho, são:

“Download” de dados – transferência dos dados armazenados na memória do datalogger para a memória do computador.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

Criação do arquivo de saída – criar um arquivo tipo texto que consolide as informações coletadas do datalogger e que seja compatível com outros softwares semelhantes.

Configuração do datalogger – determinar modos de funcionamento, frequência de varredura, tipos de sensores etc.

Apresentação de dados – Apresentar os dados ao usuário de forma amigável, sob forma de tabelas.



Detalhamento do hardware

O hardware consiste num sistema composto por: microcontrolador (8051), conversor A/D (7 canais), memória de massa (ram 512 kbytes), relógio (RTC) e interface serial RS232. O funcionamento deste hardware está descrito a seguir.

Num barramento de dados/endereços estão conectados o conversor A/D, a memória de massa e o relógio. O microcontrolador faz a interligação destes 3 elementos e comunica-se com o computador através de uma interface serial.

Os dados externos são adquiridos por sensores (velocidade de vento, direção, temperatura, umidade e pressão) que estão conectados aos canais analógicos e digitais do datalogger.

Os canais analógicos estão interligados ao conversor A/D por onde podemos digitalizar os sinais. Os sinais digitalizados são transferidos para o microcontrolador utilizando o barramento de dados/endereço. Os canais digitais estão ligados diretamente ao microcontrolador. Onde funcionam apenas da forma de contador. Ou seja, a cada pulso medido no canal digital do datalogger é incrementado o contador do mesmo.



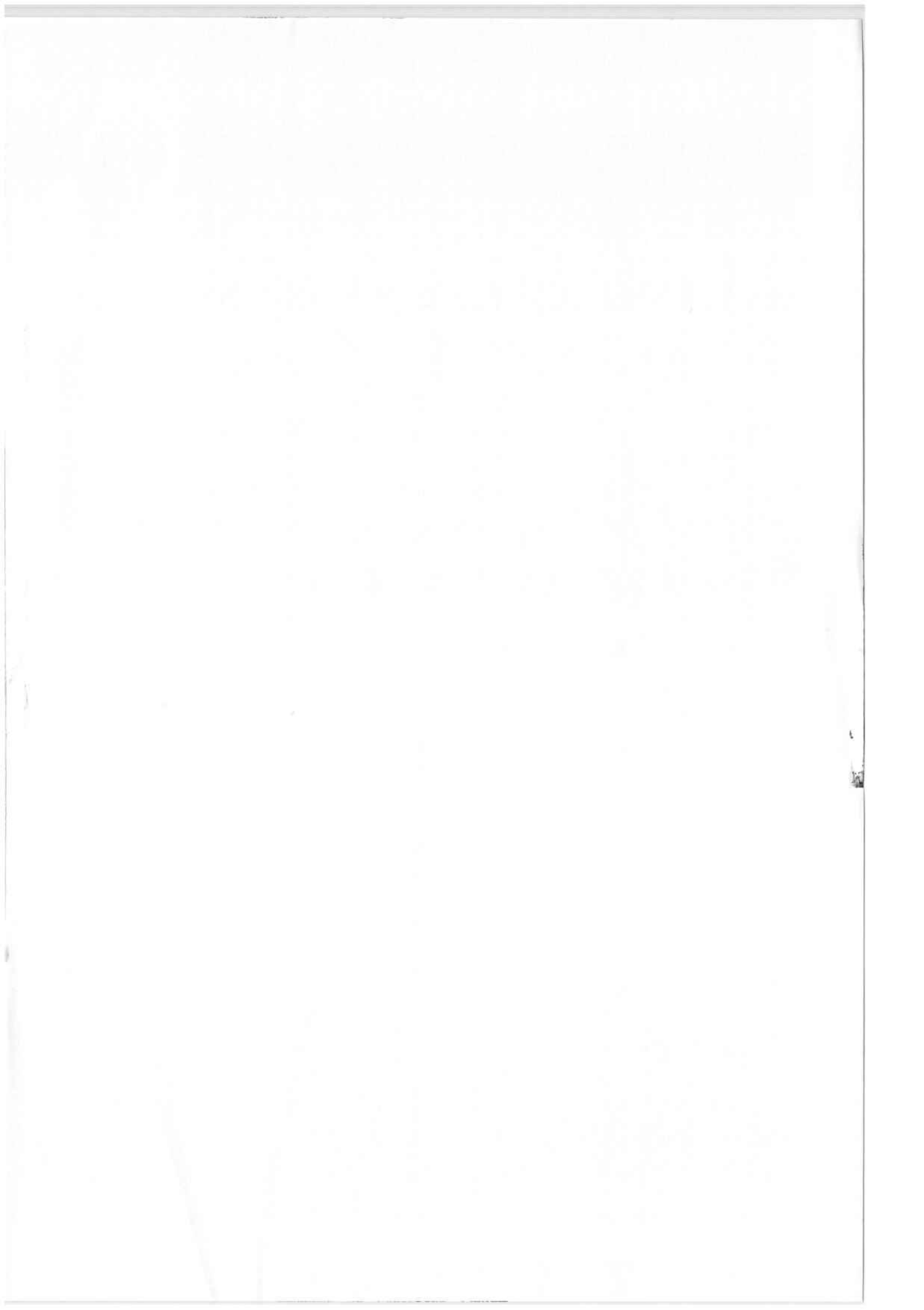
ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

O relógio (RTC – real time clock) tem duas funções no datalogger. A primeira é garantir uma hora correta para cada uma das medições feitas. A segunda é gerar sinais de interrupção ao microcontrolador a cada 10 minutos (que é o período padrão para coleta de dados de vento). Quando o microcontrolador recebe este sinal de interrupção todas as grandezas medidas são digitalizadas e armazenadas numa memória de massa junto do um rótulo da data e hora da medição.

A interface do datalogger com o computador é feita utilizando-se a porta serial do 8051. À ele foi conectado um conversor TTL RS232 que permite que o padrão de conexão seja o mesmo tanto no computador quanto no datalogger. Conectando o datalogger ao computador é possível transferir o conteúdo da memória de massa para o PC para realizar análises, gráficos, previsões e dimensionamento de sistemas.

Podemos ver mais detalhes no diagrama de blocos abaixo:



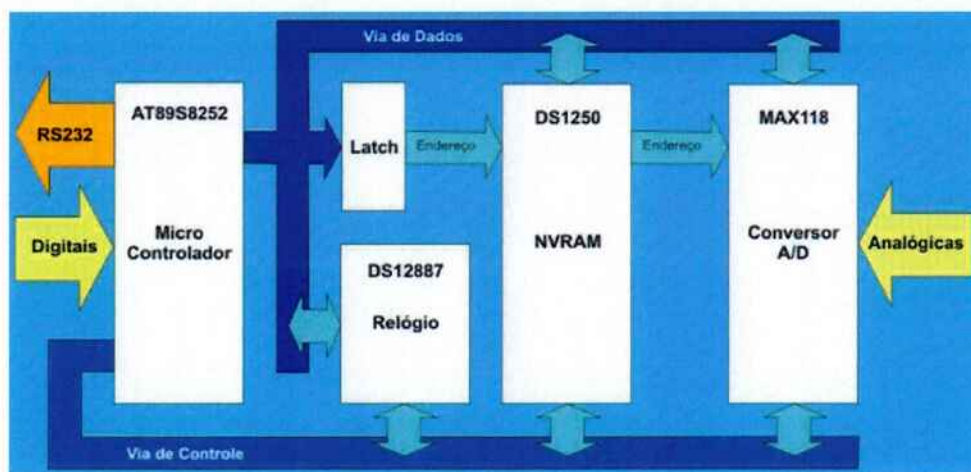


Figura 1. Diagrama de blocos do hardware 1.

Abaixo está um diagrama de blocos que mostra qual o caminho percorrido pelos dados. Desde o sinal dos sensores até o momento em que os dados efetivamente entram no computador.

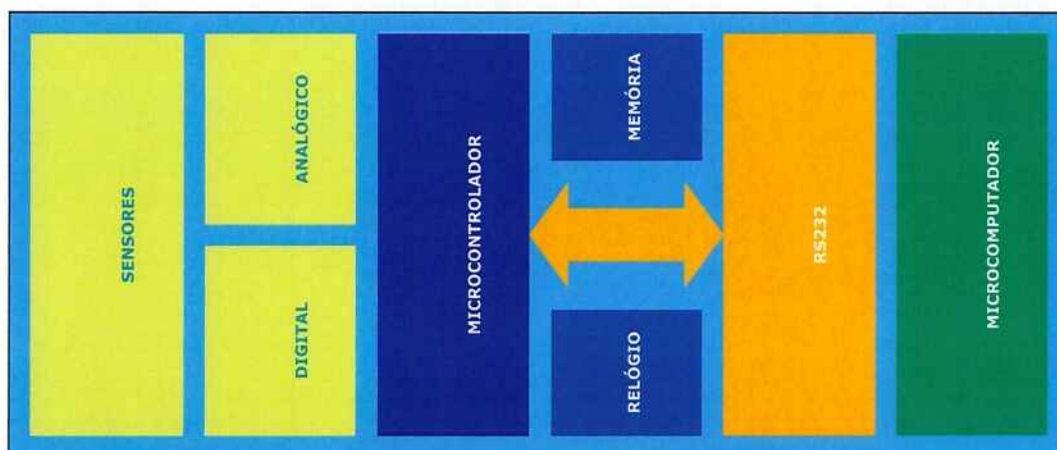


Figura 2. Diagrama de blocos do hardware 2.



Detalhamento do Software

O software está sendo desenvolvido utilizando-se a linguagem Visual Basic 6.0 pela facilidade que existe na sua manipulação e compatibilidade com a grande maioria dos PCs que utilizam o sistema operacional Windows.

O visual basic oferece uma plataforma de desenvolvimento bastante robusta e com muitos recursos, por isso é possível utilizá-lo para criar softwares relativamente complexos sem ser preciso, no entanto, conhecimentos profundos de programação.

Existem três módulos fundamentais: comunicação, processamento e interface com usuário. Responsáveis respectivamente por interface com o datalogger e comunicação serial, encapsulamento e desencapsulamento modbus e, por último, criação de arquivos de saída juntamente com telas de visualização e interação.

O módulo de recepção e transmissão da porta serial, responsável final pela comunicação com o datalogger, está configurado para utilizar a porta serial, comum em um PC doméstico. Utiliza-se do padrão RS232 para transmissão devida a sua robustez, ampla disseminação no mercado, facilidade de implementação e uso.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

O módulo de análise do protocolo se subdivide em dois módulos menores, a "formata modbus" e a "interpreta modbus". A formatação modbus é utilizada para criar a seqüência de bytes seguindo o padrão modbus. Essa seqüência será mais tarde enviada pela porta serial (a isso chamamos de encapsulamento). A interpretação faz justamente o oposto da formatação, ele recebe uma seqüência de bytes que foi formatada pelo datalogger e retira os dados pertinentes (a isso chamamos de desencapsulamento).

A interface com o usuário é feita por meio dos módulos "parâmetro usuário" e "arquivo saída". O "arquivo saída" gera um arquivo texto conhecido como "raw data" (dados brutos) que é um formato padrão utilizado na medição de dados eólicos. "Parâmetro usuário" ilustra todas as funcionalidade do software que dependem de ação do usuário.

A seguir podemos ver um diagrama de blocos que ilustra bem o funcionamento do software. Analisando este diagrama da esquerda para a direita notamos todos os caminhos que a informação percorre desde o datalogger até a tela do microcomputador.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

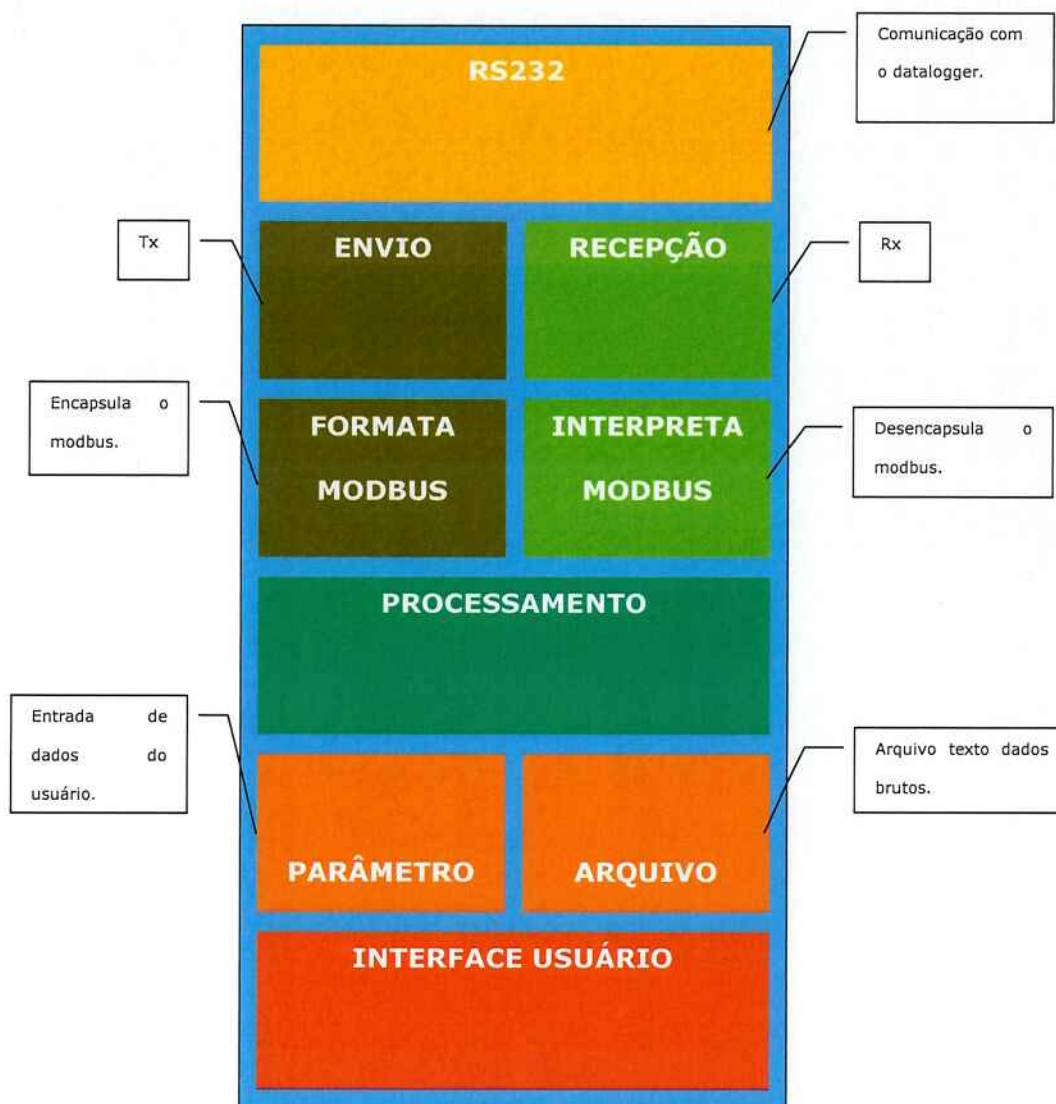


Figura 3. Diagrama de Blocos do software.

A comunicação serial foi implementada utilizando-se a biblioteca Mscmm:
MSComm.CommPort – Seleciona a porta que será utilizada para a comunicação (comumente a COM1);

MSComm.Settings – Seleciona os parâmetros da comunicação Baud Rate (taxa de transmissão em bits/segundo), paridade, quantidade de bits de



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

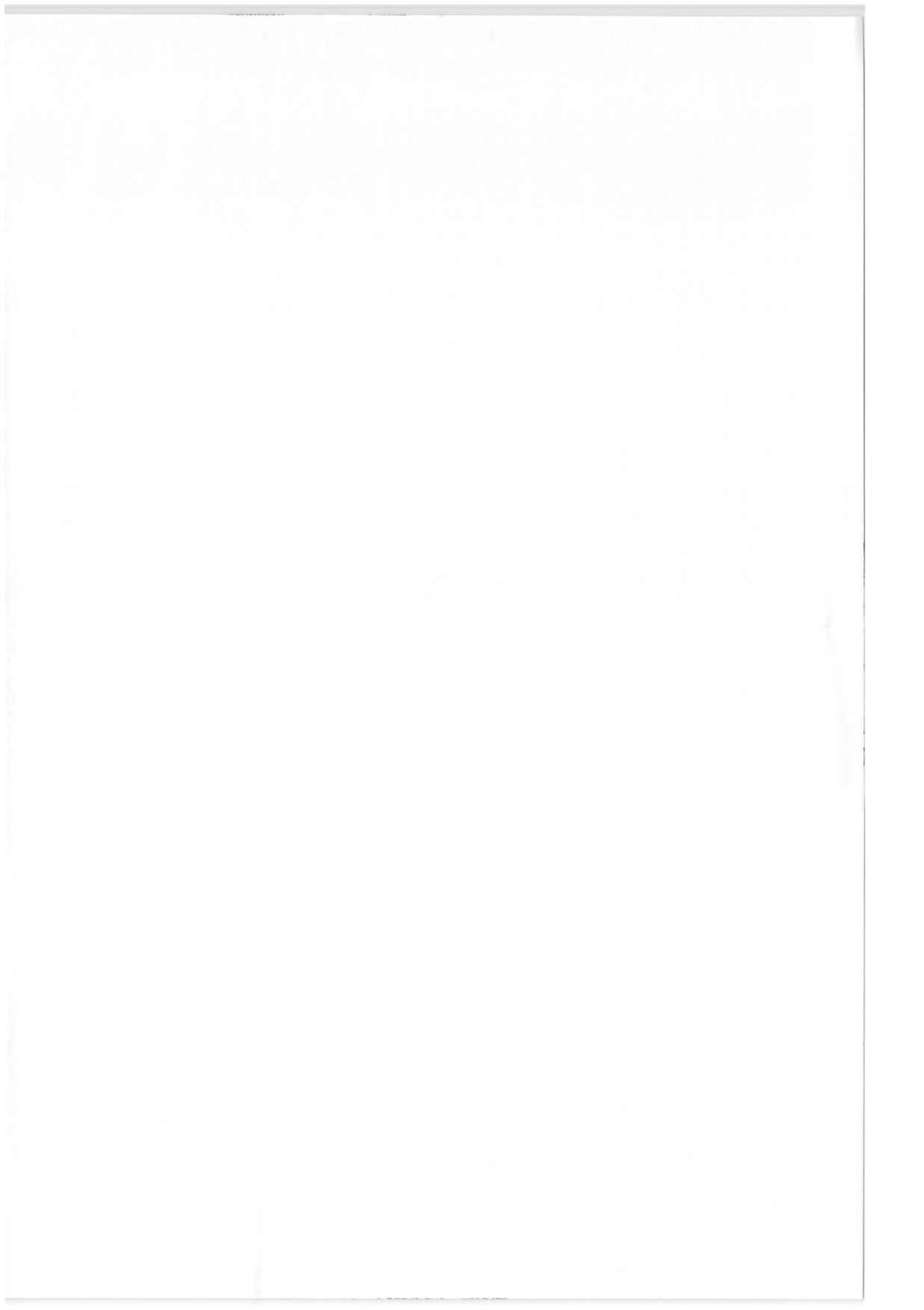
PEA 2507 – Projeto de Formatura II

dados, e bits de parada (o padrão é: Baud rate = 9600 bps, Data bits = 8 bits, Paridade = sem, Stop bit = 2 bits);

MSComm.PortOpen – Abre ou fecha a porta de comunicação;

MSComm.Input – Recebe o caractere;

MSComm.Output – Envia o caractere.





Calculo do CRC

Um dos pontos críticos para a implementação do protocolo modbus é a geração do CRC (cylindrical rule check). A primeira alternativa utilizada foi o uso de uma rotina descrita pelo fluxograma abaixo:

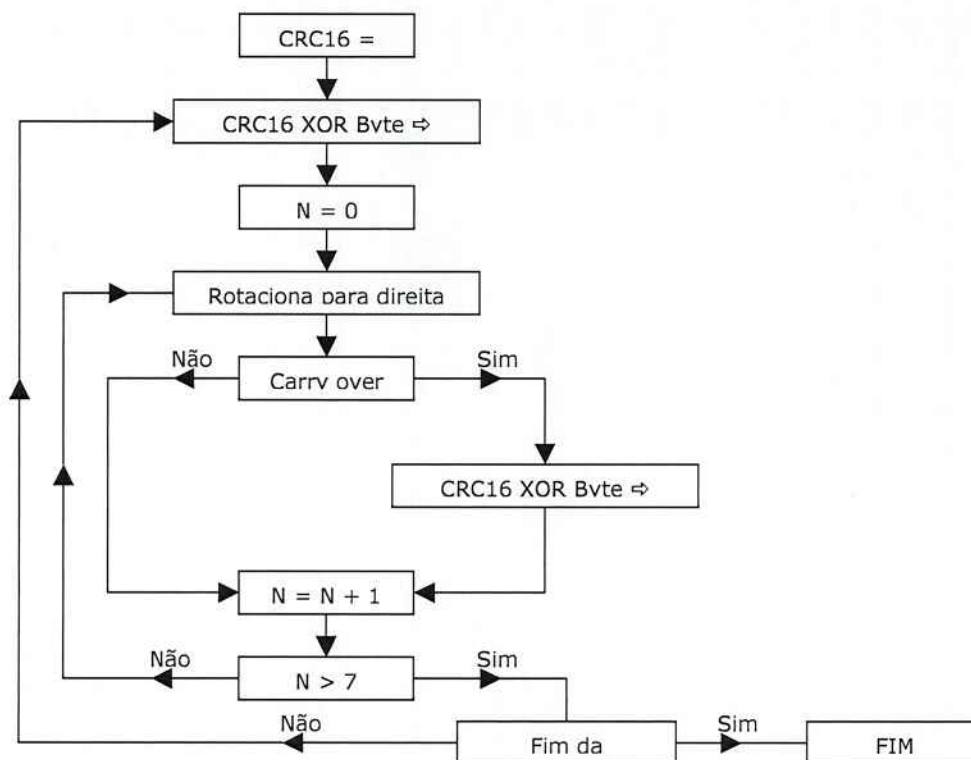


Figura 4. Fluxograma do cálculo do CRC.

Esta alternativa, no entanto, mostrou-se devagar e gastava processamento, tanto no computador quanto no microcontrolador. Foi estudada outra alternativa que consistia em duas tabelas com os dados de



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

CRC já previamente calculados e a cada byte processado era relacionado um CRC na tabela correspondente.

O método que se utiliza de duas tabelas "lookup tables" está descrito quadro abaixo:

A seguir a comparação entre os dois métodos:

CRC = 90 F1 Hex (estado inicial)

Byte de entrada 75 Hex

Tabela 1. Cálculo do CRC.

CRC_Low corrente = F1 Hex

CRC_High corrente = 90 Hex

Byte de entrada = 75 Hex

Índice de procura = (CRC_Low)

EXOR (Byte de entrada) = 132 Dec

Novo CRC_High = Tab_High[índice] =

Tab_High[132] = 63 Hex

Novo CRC_low = Tab_Low[índice]

EXOR CRC_High = 90 Hex

Novo CRC = 63 90 Hex

Podemos ver que no método das duas tabelas utilizamos 3 linhas de execução enquanto que no outro método (calculado) utilizamos nove linhas de execução. Portanto podemos ver que o primeiro método é mais eficiente e por isso foi escolhido para ser usado neste projeto.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

A tela de apresentação do programa contém uma grande caixa de texto superior contida dentro do campo "Recebe" que simula um terminal. Por ele é recebido um grande arquivo de texto com todo o conteúdo da memória de massa do datalogger. Esta caixa de texto pode ser salva criando assim um arquivo tipo texto com todos os dados coletados. Logo abaixo desta caixa de texto existe uma série de caixas consecutivamente dia, mês, ano, hora, minuto, segundos e os 8 canais. Nesta caixa de texto o microcontrolador envia os dados a cada 10 segundos e estes por sua vez são apresentados "instantaneamente".

Existem seis botões Ler Memória, Atualização Instantânea, Apagar Memória, Iniciar Leitura, Interromper Leitura e Configurar Relógio que tem suas funções descritas a seguir:

Ler Memória: Recupera todos os dados contidos na memória, desde o mais antigo até o ultimo registrado. Estes dados são apresentados na caixa de texto grande que pode ser salva em um arquivo de texto. O formato deste arquivo é mostrado na figura abaixo:



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

Arquivo	Editar	Pesquisar	Ajuda										
2004	11	28	14	12	10	255	255	255	255	255	255	255	255
2004	11	28	14	12	20	255	255	255	255	255	255	255	255
2004	11	28	14	12	30	255	255	255	255	255	255	255	255
2004	11	28	14	12	40	255	255	255	255	255	255	255	255
2004	11	28	14	12	50	255	255	255	255	255	255	255	255
2004	11	28	14	13	0	255	255	255	255	255	255	255	255
2004	11	28	14	13	10	255	255	255	255	255	255	255	255
2004	11	28	14	13	20	255	255	255	255	255	255	255	255
2004	11	28	14	13	30	255	255	255	255	255	255	255	255
2004	11	28	14	13	40	255	255	255	255	255	255	255	255
2004	11	28	14	13	50	255	255	255	255	255	255	255	255
2004	11	28	14	14	0	255	255	255	255	255	255	255	255
2004	11	28	14	14	10	255	255	255	255	255	255	255	255
2004	11	28	14	14	20	255	255	255	255	255	255	255	255
2004	11	28	14	14	30	255	255	255	255	255	255	255	255
2004	11	28	14	14	40	255	255	255	255	255	255	255	255
2004	11	28	14	14	50	255	255	255	255	255	255	255	255
2004	11	28	14	15	0	255	255	255	255	255	255	255	255
2004	11	28	14	15	10	255	255	255	255	255	255	255	255
2004	11	28	14	15	20	255	255	255	255	255	255	255	255
2004	11	28	14	15	30	255	255	255	255	255	255	255	255
2004	11	28	14	15	40	255	255	255	255	255	255	255	255
2004	11	28	14	15	50	255	255	255	255	255	255	255	255
2004	11	28	14	16	0	255	255	255	255	255	255	255	255
2004	11	28	14	16	10	255	255	255	255	255	255	255	255
2004	11	28	14	16	20	255	255	255	255	255	255	255	255
2004	11	28	14	16	30	255	255	255	255	255	255	255	255
2004	11	28	14	16	40	255	255	255	255	255	255	255	255
2004	11	28	14	16	50	255	255	255	255	255	255	255	255

Figura 5. Arquivo de saída.

Podemos observar que a primeira coluna consiste no ano, a segunda ao mês, a terceira ao dia, a quarta à hora, a quinta ao minuto, a sexta ao segundo e a sétima em diante dos dados lidos nos canais 1, 2, 3, 4, 5, 6, 7 e 8.

Este formato é compatível com o arquivo de saída de outros dataloggers e permite que o arquivo seja aberto em planilhas de texto tais como excel, open office e lotus 123 e desta forma fazer gráficos e projeções de forma simples.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

Atualização Instantânea: Acionando este botão o microcontrolador envia a cada 10 segundos os dados instantâneos lidos nos canais de 1 a 8. São preenchidos automaticamente os registros de ano, mês, dia, hora, minuto e segundo de forma que podemos acompanhar a variação dos mesmo assim que os dados são coletados.

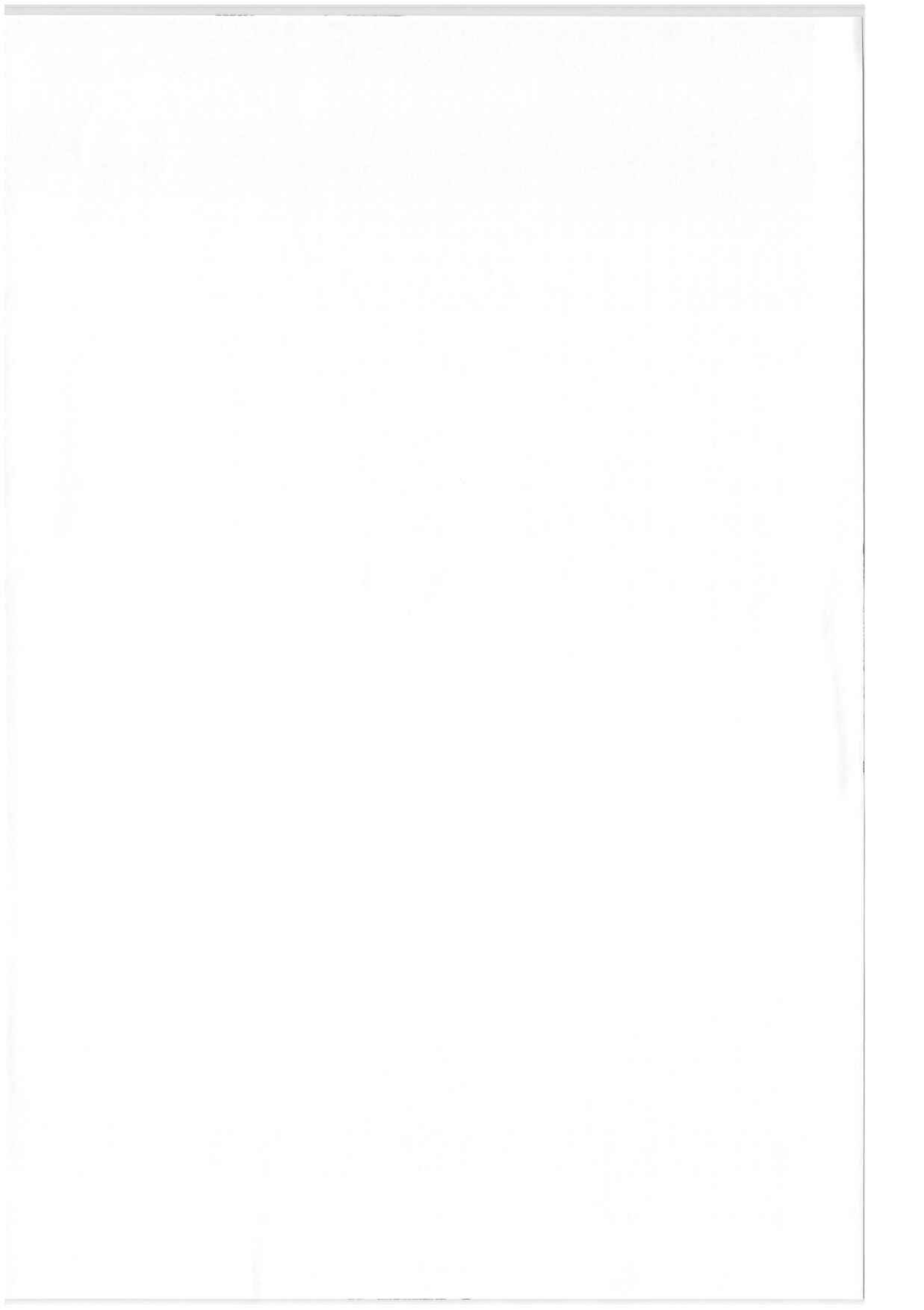
Apagar Memória: Botão que apaga a memória contida no datalogger. Acionando este botão os dados antigos passam a ser sobrescritos pelos dados novos.

Iniciar Leitura: Este botão permite o início da coleta de dados.

Interromper Leitura: Este botão faz o datalogger interromper a leitura dos sensores.

Configurar Relógio: Permite a configuração dos dados do relógio interno do datalogger.

Na posição superior da janela podemos encontrar um "Menu" de configurações: O menu "Arquivo" contém os seguintes submenus "Abrir arquivo de armazenamento", "Fechar arquivo de armazenamento", "transmitir arquivo de texto" e "Sair". Os submenus "Abrir arquivo de armazenamento" e "Fechar arquivo de armazenamento" servem respectivamente para criar um arquivo de texto que armazene todas as informações que são recebidas pela porta serial. Ou seja servirá mais





ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

adiante para criar o arquivo de texto de saída com os dados recebido pelo datalogger.

Ao lado do menu arquivo encontramos o menu "Porta de comunicação" com seus submenus: "Abrir porta" e "Propriedades".

O submenu "Abrir porta" habilita a porta serial tanto para recepção quanto para envio de dados. Ao clicar no submenu "Propriedades" abre-se uma nova janela de onde é possível configurar os parâmetros da porta serial.

Na nova janela (a de propriedades) encontramos uma série de parâmetros que podemos ajustar na transmissão: "Porta", "Velocidade", "Bits de dados", "Paridade", "Stop bits", "Eco" e "Controle de fluxo".

Existe também na parte superior da janela cinco ícones para acesso rápido das funções já contidas no menu do menu.

A seguir a tela de apresentação do programa de interface com o datalogger:



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II



Figura 6. Tela de apresentação do software.

Estudo Modbus

O Modbus é um protocolo do tipo mestre escravo em que existe apenas 1 mestre e até 247 escravos na mesma rede.

Somente o mestre inicia uma comunicação. Nunca um escravo se comunicará com outro escravo e o escravo nunca irá iniciar uma transmissão de dados sem que haja a requisição do mestre. Existem dois modos de comunicação entre o mestre e os escravos: broadcast e unicast. No modo unicast o mestre “conversa” com apenas um escravo. Depois de receber e processar a mensagem o escravo retorna uma mensagem para o mestre. No modo broadcast o mestre pode enviar uma requisição para



todos os escravos. Neste caso nenhum dos escravos irá enviar uma mensagem de volta para o mestre.

Veja a seguir uma ilustração dos modos de transmissão:

Modo Unicast:

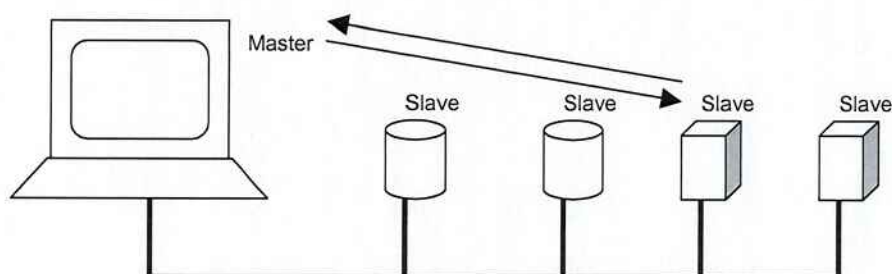


Figura 7. Modbus modo unicast.

Modo Broadcast:

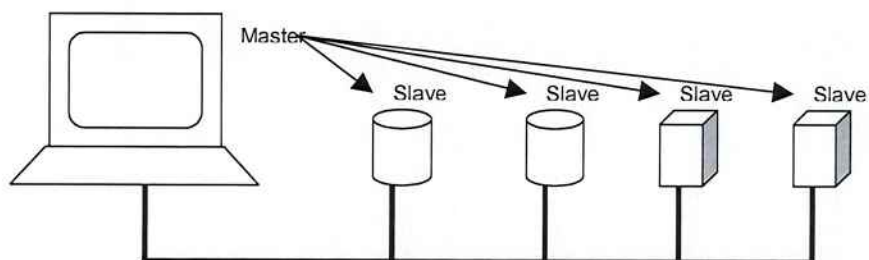


Figura 8. Modbus modo broadcast.

Podemos representar a atuação do mestre através do diagrama de estados. O mestre manda uma requisição para o(s) escravo(s) e caso seja uma requisição em broadcast o mestre apenas aguarda o tempo de espera limite ate voltar ao seu estado normal. Caso seja uma requisição para um escravo específico espera a resposta do mesmo.



Em vermelho podemos ver a resposta do mestre em caso de erros, em preto podemos acompanhar a atuação do mestre em situações normais.

Diagrama de estados para o mestre:

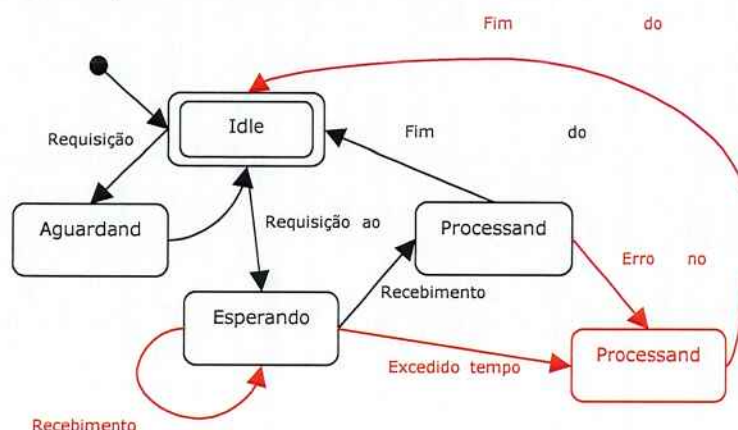


Figura 9. Diagrama de estados para o mestre.

No caso do escravo, podemos ver em seu diagrama de estados, que o fica apenas em estado de "alerta" (idle) até que receba um frame. Ao receber este frame faz um cheque de integridade do mesmo e verifica se a requisição foi encaminhada para si.

Caso o frame encontra-se correto, então a requisição é processada e o escravo envia uma resposta ao mestre. se a requisição for do tipo broadcast o escravo apenas processa a mensagem e logo em seguida volta ao estado de alerta sem enviar resposta para o mestre.

Podemos acompanhar em vermelho os erros possíveis de acontecer: erro no frame, erro no endereçamento, erro durante o processamento.



Diagrama de estados para o escravo:

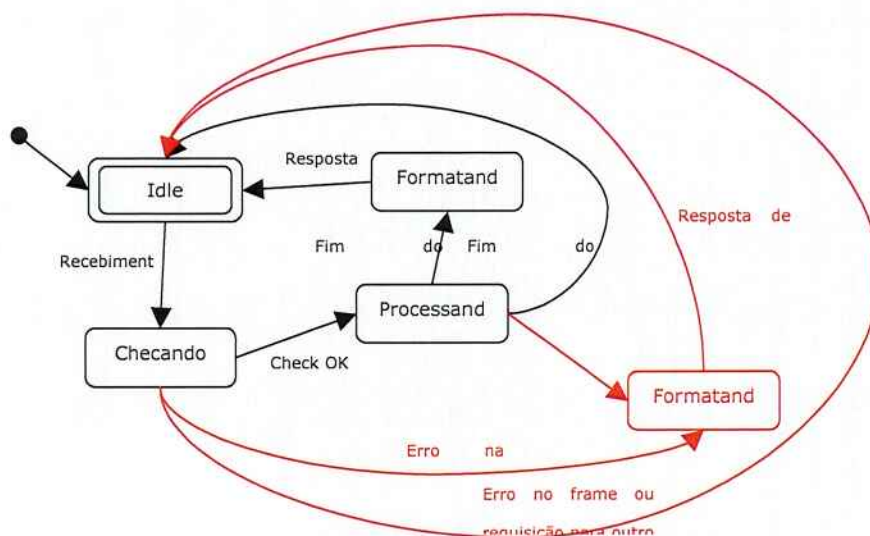


Figura 10. Modbus para o escravo.

O frame do protocolo modbus é composto por 1 byte de endereço, um byte de função, até 252 bytes de dados e mais dois bytes de CRC.

No byte de endereço o mestre determina qual dos escravos irá receber o frame. Ele pode ser endereçado de 1 até 247 sendo que o endereço 0 é reservado para o modo broadcast.

No byte de função é colocado um código binário de 0 até 256. Ele serve para indicar uma determinada ação ao escravo. A lista de funções de um equipamento que utiliza o protocolo modbus é definida livremente por seu fabricante, a norma não estabelece nenhuma regra para utiliza-las.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

A área para dados pode conter até 247 bytes e destina-se a troca de dados entre mestre e escravo e vice-versa.

O CRC é composto de dois bytes e é o responsável pelo "check-sum", ou seja com estes dois bytes podemos saber se os dados que foram recebidos tiveram algum erro e assim descartar e requisitar um novo frame.

Podemos ver a seguir uma ilustração de um frame modbus:

Início	Endereço	Função	Dados	CRC	Fim
$\geq 3,5$ char	8 bits	8 bits	N x 8 bits	16 bits	$\geq 3,5$ char



Estudo Normas IEC

Neste trecho foi destacado o que havia de mais importante da norma IEC 61400-12 *Wind turbine generator systems. Part 12: Turbine power performance testing*. No que diz respeito à metodologia de coleta de dados de vento.

Velocidade do vento:

A velocidade do vento deve ser medida com um anemômetro de copo instalado na altura do cubo do rotor numa torre meteorológica.

O anemômetro deve ser montado entre na altura do cubo do rotor podendo variar numa faixa de 2,5%. Deve estar localizado preferencialmente no topo de um tubo vertical e circular fixado no alto de uma torre meteorológica.

Direção de vento:

A medição da direção do vento deve ser feita com um sensor de direção montado numa torre meteorológica na altura do cubo do rotor podendo variar dentro de uma faixa de 10%.

Densidade do ar:



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

A densidade do ar deve ser calculada a partir da temperatura e pressão usando a equação abaixo. Em altas temperaturas é recomendada também a medição da umidade relativa do ar.

Equação 1. Densidade do ar.

$$\rho_{10\text{ min}} = \frac{B_{10\text{ min}}}{R \cdot T_{10\text{ min}}}$$

onde:

$\rho_{10\text{ min}}$ - densidade média num período de 10 minutos;

$T_{10\text{ min}}$ - temperatura média absoluta do ar num período de 10 minutos;

$B_{10\text{ min}}$ - pressão média do ar num período de 10 minutos;

R - constante dos gases 287,05 J/(kg x K)

O sensor de temperatura deve ser montado pelo menos 10 metros acima do nível do chão. Pode ser montado na torre meteorológica perto da altura do cubo do rotor para dar uma boa representação da temperatura no centro do rotor.

O sensor de pressão deve ser montado na torre meteorológica perto do cubo do rotor para se ter uma boa representação da pressão no centro do



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

rotor. Se o sensor de pressão não está montado próximo a altura do cubo, a pressão medida deve ser corrigida para a altura do cubo de acordo com ISSO 2533.

Precipitação:

Para diferenciar medidas de períodos chuvosos e secos, a precipitação pode ser monitorada durante o período de medição e documentada no relatório.

Sistema de aquisição de dados:

Deve ser usado um sistema de aquisição de dados digital tendo uma taxa de amostragem por canal de pelo menos 0,5 Hz, para coletar e armazenar os dados pré-processados.

Coleta de dados:

Os dados devem ser coletados continuamente numa taxa de amostragem de pelo menos 0,5 Hz. Temperatura, pressão e precipitação podem ser coletados com amostragens numa taxa menor mas no mínimo 1 amostra por minuto.

O sistema de aquisição de dados deve armazenar todos os dados mostrados ou conjunto de dados pré-processados como descrito abaixo ou ambos. O conjunto de dados pré-processados deve conter as seguintes informações:



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

valor médio;

desvio padrão;

valor máximo;

valor mínimo.

Seleção de dados:

Os dados selecionados devem ser baseados em períodos de 10 minutos retirados de medições contínuas. Os valores médios e desvio padrão para cada período de 10 minutos devem, quando retirados de conjunto de dados pré-processados, ser calculado de acordo com as equações abaixo:

Equação 2. Média em 10 minutos.

$$X_{10\min} = \frac{1}{N_k} \cdot \sum_{k=1}^{N_k} X_k$$

Equação 3. Desvio padrão em 10 minutos.

$$\sigma_{10\min} = \sqrt{\frac{1}{N_k N_s - 1} \sum_{k=1}^{N_k} (N_s (X_{10\min} - X_k)^2 + \sigma_k^2 (N_s - 1))}$$

Onde:

N_k - número de conjuntos de dados pré-processados dentro de um período de 10 minutos;

X_k - o parâmetro médio dentro do período de pré-processamento;



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

X_{10min} - parâmetro médio dentro de 10 minutos;

N_s - número de amostras de conjunto de dados pré-processados;

σ_k - desvio padrão do parâmetro pré-processado;

σ_{10min} - desvio padrão dos dados pré-processados durante um período de 10 minutos.

Se levarmos em consideração 5 sensores (velocidade, direção, temperatura, pressão e umidade) e que devemos fazer amostragem dos sinais de velocidade e direção a cada 2 segundos (0,5 Hz) e os demais uma amostragem a cada 2 minutos, teremos a cada 10 minutos $300\text{ s} \times 2 = 600$, para dados de direção e velocidade de vento, e $5 \times 3 = 15$ dados para os sensores de temperatura pressão e umidade tomado um período de medição de 10 minutos. A cada dia teremos $615 \times 144 = 88.560$ dados. Como cada dado ocupa um espaço de memória de 1 byte então teremos necessidade de 88,560 Kbytes para cada dia inteiro de coleta. Como a capacidade de armazenamento é de 512 Kbytes então temos uma autonomia de aproximadamente 5 dias e 17 horas.

Essa autonomia representa o tempo que o datalogger pode ficar em campo coletando dados sem que haja necessidade de fazer o "download" dos mesmo para um computador. Em outras palavras, para essa



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

configuração, devemos conectar o data logger ao computador no mínimo 1 vez a cada 5 dias.

Outra solução seria armazenar somente os dados pré-processados, ou seja, para cada sensor devemos ter apenas guardados os dados máximo, mínimo, médio e desvio. Utilizando este método teremos 4 medidas armazenadas a cada 10 minutos por sensor o que nos dá $5 \times 4 = 20$ registros a cada 10 minutos], que nos resulta numa quantidade diária de $20 \times 144 = 8.820$ dados e teremos assim uma necessidade diária de 8,820 Kbytes de memória por dia e portanto uma autonomia (dado 512 Kbytes de memória) de 58 dias sem necessidade de realizar o "download" dos dados.

A segunda solução parece ser a mais interessante uma vez que possibilita uma grande autonomia, porém deve-se considerar que haverá um sobrecarga no processamento (pois será necessário executar operações matemáticas como multiplicação e divisão) que podem comprometer o desempenho do datalogger.

Não apenas o desempenho pode ser comprometido como a complexidade do firmware cresce muito. Seria preciso criar rotinas especiais para cálculo de média, cálculo de desvio padrão, cálculo de máximas e de mínimas e o 8051 não oferece uma facilidade para a programação dos mesmos.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II



Firmware

O firmware é o software que controla o microcontrolador. Sua função é gerenciar todas as atividades dos periféricos do datalogger como selecionar elementos ativos no barramento de dados, estabelecer a comunicação entre o computador e o datalogger, etc.

O firmware neste projeto foi desenvolvido na linguagem assembly devido a dificuldade em se encontrar compiladores em linguagens mais amigáveis para este tipo de equipamento.

Esta escolha apresenta, entretanto algumas vantagens, sendo duas principais. A primeira é que programando diretamente em assembly (sem o uso de compiladores em linguagens de alto nível) conseguimos um código muito mais eficiente e, portanto com menos instruções. A segunda vantagem é evitar erros que o compilador insere no código e que, portanto que não temos controle.

Algumas funções em assembly são de vital importância para o funcionamento do sistema e por isso abaixo estão listados algumas destas funções implementadas com seus devidos comentários.

Init_ser: Função que configura a porta serial. Ela atribui os parâmetros necessários ao microcontrolador para que a sua porta serial funcione



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

seguindo os parâmetros de Baud Rate 2400 bps, 1 start bit, 1 stop bit, 8 bits de dados e nenhum bit de paridade

Txb: Função que transmite um byte pela porta serial. Para ser usada o byte a ser transmitido deve se localizar no acumulador. Assim se desejamos transmitir o byte 09 hexadecimal colocamos este byte no acumulador e em seguida chamamos a função txb.

Rxb: Função de recebimento de byte. Esta função é ativada pela interrupção serial e é executada toda vez que recebe algum dado pela porta serial. Uma rotina inserida dentro desta própria função determina qual a ação a ser executada de acordo com o byte recebido.

CRC16: Função que calcula o CRC para o protocolo modbus. Assim como explicado anteriormente o CRC consiste em dois bytes que são inseridos no final dos dados a serem transmitidos que permite avaliar se houve erro na transmissão de dados.

SelRel, SelMem, SelAD: Estas três funções são respectivamente seleciona relógio, seleciona memória e seleciona conversor A/D. Tem por função tornar o elemento ativo no barramento de dados e endereço. O Datalogger contém uma via de dados e endereço compartilhada entre todos os elementos. Apenas um dos elementos pode se comunicar por vez



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

com o microcontrolador. Estas função tornam então ativo o elemento que desejamos comunicar e conseqüentemente desativa os demais.

LeRel: Função que lê o ano, mês, dia, hora, minuto e segundo contidos no relógio e armazena em variáveis auxiliares para ser usado posteriormente em outras rotinas.

LeAD: Função semelhante ao *LeRel* porém faz uma leitura do conversor A/D e seus 8 canais, armazena-os em variáveis auxiliares para uso posterior em outras rotinas.

EscMem: Função que salva os seguintes registros na memória RAM de forma ordenada: ano, mês, dia, hora, minuto e segundo e os 8 canais do conversor AD. Esta rotina acessa as variáveis auxiliares das funções *LeRel* e *LeAD* para em seguida armazena-las na memória RAM.

LerMem: Rotina que recupera os valores armazenados na memória RAM e manda pela porta serial desde o primeiro registro até o ultimo armazenado.

SetRel: Função que configura o relógio com ano, dia, mês hora, minuto e segundo iniciais. Após isso o relógio atualiza automaticamente.

SetAlr: Função que configura o próximo instante para alarme. Este alarme serve como sinal de interrupção para o microcontrolador. E sinaliza que os dados devem ser coletados e armazenados.



Conclusão

Neste trabalho foi desenvolvido a interface de comunicação entre um datalogger e o computador. Foi escopo deste projeto o firmware (software embarcado no microcontrolador) e o software de interface homem máquina que funciona num microcomputador. O software permite uma interface amigável para configurar o datalogger e ler os dados armazenados em sua memória de massa. Outra funcionalidade é que este software permite a gravação de um arquivo tipo texto que é compatível com arquivos de saída de outros dataloggers disponíveis no mercado. É possível também a abertura deste arquivo em planilhas eletrônicas como excel, lotus 1 2 3 etc.

Os principais problemas enfrentados foram dificuldades com o circuito uma vez que o mesmo encontra-se no modo de protótipo numa placa universal. Por diversas vezes houve problemas de mau contato e rompimentos de fios que atrapalhou o andamento do trabalho. Outros problemas foram devido a não familiaridade com o software do visual basic e muitas vezes tarefas simples levaram muito tempo para serem realizadas.

Este trabalho desenvolvido permite que seja aplicado em outros projetos, tanto acadêmicos quanto comerciais, principalmente na parte de



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

aprimoramento do produto, quanto em aplicação do mesmo. Uma sugestões são o uso deste datalogger para coleta de dados de vento e levantamento de potencial eólico ou desenvolvimento de novas funcionalidades para o datalogger.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

Bibliografia

IEC, International Standard. *IEC 61400-12 Wind turbine generator systems part 12: Wind turbine performance testing*. First edition 1998-02

MICROSOFT, Corporation. *Microsoft visual basic 6.0 guia do programador*. Makron 1998.

MODBUS.ORG. *MODBUS over serial line v1.0*. www.modbus.org

PREDKO, Myke. *Programming and customizing the 8051 microncontroller*. McGraw-Hill 1999.

SILVA, P. C. *Sistema para tratamento, armazenamento e disseminação de dados de vento*. Tese de mestrado, Rio de Janeiro, UFRJ, 113p.

STEWART, James W. *The 8051 microcontroller: hardware, software and interfacing*. Prentice Hall 1999.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PEA 2507 – Projeto de Formatura II

São Paulo, 22 de Fevereiro de 2006

Dorel Soares Ramos – Coordenador

Eliane Aparecida Faria Amaral Fadigas – Orientadora

Ricardo Ping de Tung Gabriel - Aluno