

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA DA USP - PMR
ENGENHARIA MECATRÔNICA

Felipe Prétola Lordêllo

Renato Gomes Cohen

*Sistema de medição unificado do consumo de água
individualizado em condomínios*

São Paulo

2011

Felipe Prétola Lordêllo

Renato Gomes Cohen

*Sistema de medição unificado do consumo de água
individualizado em condomínios*

Monografia apresentada ao Curso de Engenharia
Mecatrônica da USP, como requisito para a obten-
ção do grau de BACHAREL em Engenharia Me-
catrônica.

Orientador: Prof. Dr. Marcos Ribeiro Pereira Barretto
Doutor em Engenharia Mecânica - Universidade de São Paulo

São Paulo

2011

Lordêllo, Felipe

Sistema de medição unificado do consumo de água individualizado em condomínios / F.P. Lordêllo , R.G. Cohen – São Paulo, 2011
54.p

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Hidrometria 2. Wireless 3. Água (Medição) I. Cohen, Renato Gomes II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III. t..

Felipe Prétola Lordêllo

Renato Gomes Cohen

*Sistema de medição unificado do consumo de água
individualizado em condomínios*

Monografia apresentada ao Curso de Engenharia
Mecatrônica da USP, como requisito para a obten-
ção do grau de BACHAREL em Engenharia Me-
catrônica.

Aprovado em 04 de julho de 2011

BANCA EXAMINADORA

Prof. Dr. Marcos Ribeiro Pereira Barretto

Doutor em Engenharia Mecânica - Universidade de São Paulo

Prof. Dr. Lucas Antonio Moscto

Livre Docente da Universidade de São Paulo e Doutor em Engenharia Elétrica

Prof. Dr. Edson Gomes

Doutor em Engenharia Mecânica

Aos nossos pais e amigos que tanto nos apoiaram e contribuíram para a conclusão deste trabalho.

Ao nosso orientador, pela dedicação e paciência com que sempre nos tratou.

Resumo

A medição individualizada do consumo de água em apartamentos está prevista e sendo adotada em boa parte dos municípios brasileiros, obrigatoriamente, na forma de lei. A busca recai sobre a busca de uma tarifação mais justa e que privilegie aqueles que mais economizam, assim como já acontece com outras contas, como a de telefone, a de luz, etc.

Mesmo ainda não aprovada na cidade de São Paulo, o projeto de lei 551/03 obrigará todos os condomínios, os novos e os antigos, a disponibilizar hidrômetros para efetuar uma medição individualizada para cada apartamento. Isso já é considerado e implementado de modo crescente pelas construtoras, visando maior conveniência e economia para os clientes que residirão nos edifícios.

No caso de apartamentos novos, o projeto hidráulico de prumadas e tubulações já é realizado prevendo de antemão os requisitos para que haja a medição individualizada, e contando com todas as facilidades de leitura oferecidas às empresas abastecedoras.

Já nos apartamentos antigos, isso não é tão simples. Há mais de uma prumada chegando em cada um dos apartamentos, o que significa que essa medição precisará ser realizada em cada prumada. Como os medidores teriam que ser instalados em lugares de fácil acesso para a leitura, o nosso projeto consiste na medição e na transmissão dessa informação via Wireless para um único ponto de acesso, localizado no piso térreo do condomínio e de lá, possivelmente para uma central de gerenciamento e controle.

Abstract

The individual metering of water consumption of apartments is under development in major part of the Brazilian cities, mandatorily, by law enforcement. This is motivated by an attempt to achieve a more fair charging which charters customers engaged in saving resources, as it already works for the phone and power bills, for instance.

Even though the laws project 551/03 hasnt been approved yet, it longs to enforce every apartment complex, the new and the old ones, to offer hydrometers for an individual metering of each apartments consumption. This procedure has been adopted and keeps growing among the house builders, aiming a wider convenience and more saving for the residents.

For new apartments, the hydraulic project of water lines is made in such way that it considers the individual metering and facilities to be provided to the water service companies.

On the other hand, for old apartments, its not so simple. There are several water lines sourcing each apartment, and so we should have one meter for each line. This could hamper the metering readout, therefore, our project is based on metering and wireless transmitting the consumption information to a central system, where it could be displayed as well as treated in many ways. This would improve managing and controlling the waters consumption.

Agradecimentos

Agradecemos a todos os professores pela dedicação e apoio irrestrito ao longo de todos esses anos de Escola Politécnica. O esforço de cada um de vocês foi fundamental em cada etapa do processo de conhecimento e formação como Engenheiros.

Em especial, agradecemos ao nosso orientador, Professor Doutor Marcos Pereira Barretto pela sugestão do tema de nosso Trabalho de Conclusão de Curso.

Agradecemos também o apoio dos colegas de faculdade, da família e dos amigos do bloco do troleibus que sempre nos ajudaram a tentar superar os problemas, dando todo o suporte possível para o trabalho.

“Missão dada é missão cumprida!”.

Capitão Nascimento

(personagem de Wagner Moura no filme

“Tropa de Elite”)

Lista de Figuras

1.1	<i>Esquema de instalação em condomínios</i>	12
2.1	<i>Exemplo de mensagem serial síncrona a ser transmitida</i>	17
2.2	<i>Exemplo de mensagem serial assíncrona a ser transmitida</i>	18
2.3	<i>Diferentes camadas da pilha TCP/IP</i>	20
2.4	<i>Módulo XBeeNet</i>	21
2.5	<i>Esquema elétrico da placa XBeeNet - parte 1</i>	22
2.6	<i>Esquema elétrico da placa XBeeNet - parte 2</i>	23
2.7	<i>Esquema elétrico da placa XBeeNet - parte 3</i>	23
2.8	<i>Esquema elétrico da placa XBeeNet - parte 4</i>	24
3.1	<i>Esquema simplificado dos componentes do projeto</i>	25
3.2	<i>Modelo de leitura de um hidrômetro</i>	26
3.3	<i>Representação do mecanismo de um hidrômetro analógico</i>	26
3.4	<i>Esquema de instalação de um hidrômetro convencional em residência . . .</i>	27
3.5	<i>Hidrômetro digital</i>	28
3.6	<i>Estrutura física dos PICs da família 18</i>	29
3.7	<i>Esquema elétrico da placa de leitura</i>	30
3.8	<i>Diagrama de casos de uso do PIC contador</i>	31
3.9	<i>Diagrama estados do PIC contador</i>	32
3.10	<i>Diagrama de atividade do PIC contador</i>	33
3.11	<i>Diagrama de componentes do PIC contador</i>	33
3.12	<i>Diagrama de classes do PIC contador</i>	34

3.13	<i>Diagrama de casos de uso do PIC TCP/IP</i>	34
3.14	<i>Diagrama estados do PIC TCP/IP</i>	35
3.15	<i>Diagrama de atividade do PIC TCP/IP</i>	36
3.16	<i>Diagrama de classes do PIC TCP/IP</i>	37
3.17	<i>O equipamento RC1140</i>	38
3.18	<i>Regulação da tensão</i>	38
3.19	<i>Esquema de mensagem</i>	39
3.20	<i>Esquema para conexão RS232</i>	40
4.1	<i>Página Home do WebSite</i>	50
4.2	<i>Página LCD do WebSite</i>	51
4.3	<i>Página Hidrometros do WebSite</i>	51
4.4	<i>Esquema de carregamento da página do WebSite</i>	52

Lista de Tabelas

3.1	Orçamento do módulo ‘Leitor’	42
3.2	Orçamento do módulo ‘Receptor’	43
3.3	Orçamento do módulo ‘TCP/IP’	44
3.4	Orçamento final do projeto	45

Sumário

Lista de Figuras	6
Lista de Tabelas	8
1 Introdução	11
1.1 Motivação	13
1.2 Objetivos	13
1.3 Organização do trabalho	14
2 Revisão Bibliográfica	15
2.1 UML	15
2.2 Comunicação serial	16
2.2.1 Interface serial RS232	18
2.3 Hardware e Módulo XBeeNet	19
3 O projeto	25
3.1 Hidrômetro	25
3.2 Leitor	28
3.3 Transmissão de dados Wireless	37
3.4 Módulo TCP/IP	40
3.5 Central de controle	41
3.6 Orçamento	41
4 Pilhas TCP/IP e módulo XBeeNet	46
4.1 Configuração das Pilhas TCP/IP	46

SUMÁRIO	10
4.1.1 <i>HardwareProfile.h</i>	48
4.1.2 <i>MainDemo.c</i>	49
4.1.3 <i>LCDBlocking.h</i> e <i>LCDBlocking.c</i>	49
4.2 WebSite gerado pela XBeeNet	50
5 Conclusão	53
Referências Bibliográficas	55

1 Introdução

Atualmente a medição de água residencial é feita através de hidrômetros, que contam os giros e de uma forma mecânica transformam a rotação para volume de água em m^3 . Essa verificação é feita manualmente por fiscais da companhia de água que devem passar em todas as residências e anotar as alterações no popularmente conhecido “relógio de água”.

Para apartamentos a verificação é parecida; o fiscal mede a vazão geral do prédio e a conta é dividida por todos os moradores do condomínio. O problema desse modo é que não se faz distinção entre os apartamentos, sendo que se alguém consumir pouca água terá que pagar a mesma quantia que os outros. Além desse fator de justiça social, em muitos casos onde a medição é feita de modo individual foi verificado que houve uma redução no desperdício de água.

Por isso, em diversas regiões do país projetos de leis estão sendo aprovados para tornar obrigatória a individualização da medição de vazão em condomínios.

A reforma em apartamentos antigos pode ser muito custosa e apesar de uma tendência pela individualização, muitos moradores ainda relutam pela mudança, visto que perderiam a privacidade e a segurança, tendo que abrir suas casas para os fiscais fazerem a medição.

Pensando nisso, o projeto consiste em construir um sistema de medição de água que possa transmitir a informação via *wireless* para uma central que fica no térreo do prédio. Essa central facilita o trabalho dos fiscais que não precisariam passar em todos os apartamentos anotando os valores e garante uma maior segurança para os moradores.

A transmissão *wireless* diminui consideravelmente o custo, já que não se faz necessário o uso de cabos e da instalação do hidrômetro em algum lugar de fácil acesso, podendo ficar em um lugar de menor custo. A figura 1.1 a seguir mostra uma representação ilustrativa do que se pretende nesse projeto.

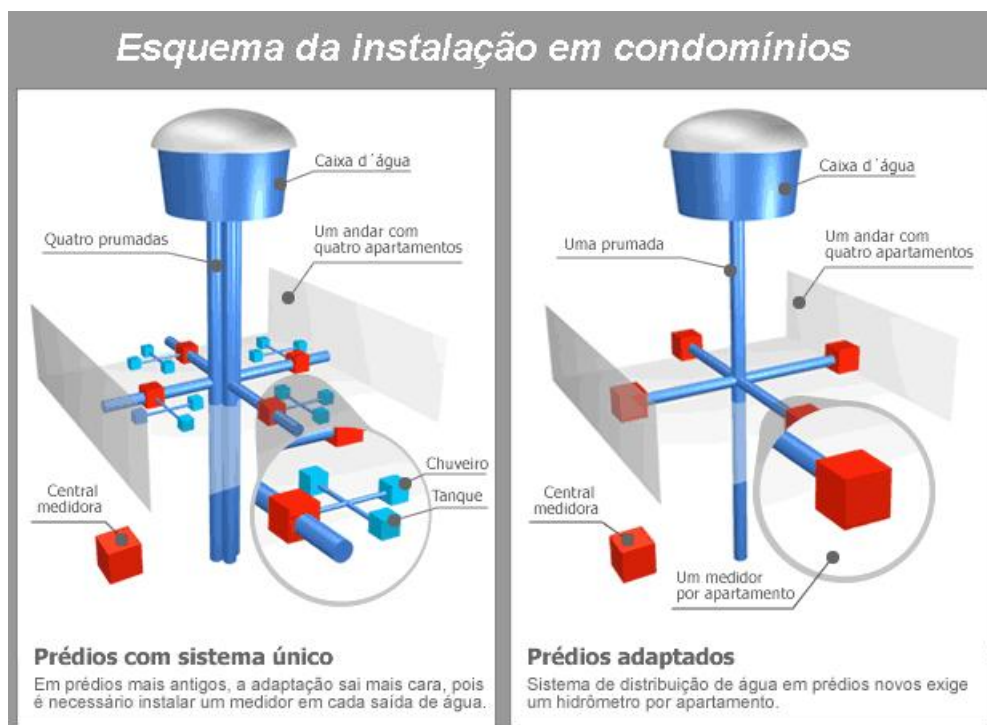


Figura 1.1: *Esquema de instalação em condomínios*

Em algumas cidades já foi implementado o sistema individualizado, e o preço chegou a valores exorbitantes. Isso é devido à necessidade enorme de reformas nas instalações. Apesar da alta variação do valor entre as cidades, o preço médio de instalação pode chegar até a R\$1.500,00 por apartamento. O projeto pretende reduzir o custo da instalação e trazer uma proposta em seu lugar muito mais acessível a todos.

Com os equipamentos que descritos aqui e os materiais a serem utilizados, o custo por apartamento será de aproximadamente R\$500,00, dispensando qualquer custo com locação de aparelhos. Basta lembrar que os custos com manutenção são irrisórios e a energia elétrica despendida por esse sistema também é baixíssima, já que todos os componentes operam com correntes muito baixas, na ordem de miliampères.

É importante ainda lembrar, que não desenvolve-se aqui uma solução pensando em casos emergenciais onde há paradas temporárias ou intermitentes de energia, ficando aqui a sugestão para que se desenvolva futuramente soluções nesse sentido.

1.1 Motivação

O monitoramento à distância de equipamentos e ambientes é uma preocupação crescente nos projetos de automação industrial e até mesmo residencial. Isso tem como consequência direta uma garantia adicional de que os sistemas permanecem operacionais e disponíveis, sem elevar tanto os custos com pessoal.

Em especial, a área de controle de processos e medições já está bastante consolidada no mercado e conta com uma variedade de técnicas, procedimentos e equipamentos que permitam alcançar o objetivo de leituras do consumo de água, energia e gás. Isso já não é novidades.

No entanto, a digitalização dessas informações através do uso de microcontroladores e a sua posterior disponibilização em aplicações *web*, bancos de dados ou mesmo o direcionamento direto para computadores centrais das companhias distribuidoras traz vantagens inovadoras para a sociedade.

A tecnologia utilizada como base para o desenvolvimento de nosso projeto é a comunicação entre sistemas embarcados, ora por meio da rádio frequência, ora por meio da conexão destes mesmos sistemas com a internet. Por intermédio dessa tecnologia, objetiva-se a integração entre os sistemas atuais de medição e a respectiva publicação das informações recolhidas, através de redes TCP/IP.

Em resumo, esta monografia apresenta o projeto de um sistema microcontrolado para a medição e o monitoramento unificado do consumo de água de diferentes apartamentos ou prumadas, utilizando comunicação *wireless* como meio de comunicação entre os sensores de aquisição e um controle central. Além disso, apresenta a configuração das pilhas TCP/IP em sistemas embarcados para que fosse possível disponibilizar os dados na Internet através de um servidor HTTP.

1.2 Objetivos

Apesar dos novos sistemas de medição individualizada já ser implementado nos novos projetos de condomínios de apartamentos e casas, a maioria dos apartamentos dos grandes centros urbanos, já antigos, seguem efetuando uma única medição do consumo

de água para todo o condomínio, que posteriormente é rateado entre os moradores.

Trata-se de um incômodo desconforto para todos os residentes que almejam pagar exatamente por aquilo que consomem.

Ao mesmo tempo, há também um desconforto para as empresas fornecedoras de recursos, como a SABESP, por exemplo. Como identificar os responsáveis pelos maiores desperdícios na sociedade e, ao mesmo tempo, como poupar tempo e desgaste de seus funcionários que muitas vezes caminham o dia inteiro para efetuar medições ao longo da cidade?

Dessa forma, o objetivo desse projeto é conseguir efetuar o controle do consumo de água em condomínios de modo unificado e que disponibilize a informação em uma central, independente do número de prumadas que essa possa haver por apartamento. A possibilidade de visualização remota dos dados através de redes ou da internet através da implantação do protocolo TCP/IP são os objetivos últimos desse projeto.

1.3 Organização do trabalho

Esse trabalho está dividido em capítulos de modo a facilitar a compreensão total do mesmo. No capítulo 2 faz-se a revisão bibliográfica, que basicamente é uma revisão teórica dos assuntos mais importantes tratados no trabalho.

Com a base teórica mais bem elucidada, no capítulo 3 inicia-se o projeto referente a essa monografia. Neste capítulo está explicado tudo sobre o projeto, desde os materiais utilizados e o motivo pela escolha de cada parte do projeto até o funcionamento do trabalho.

As pilhas TCP/IP e o módulo XBeeNet usado no projeto são mais complexos e portanto tem-se um capítulo reservado para esse tema.

O último capítulo possui a conclusão do projeto e é muito importante pois traz a opinião dos autores sobre o trabalho e o seu desenvolvimento.

2 Revisão Bibliográfica

2.1 UML

UML é a sigla em inglês de Unified Modeling Language e é usada para desenvolvimento de softwares. A partir dos requisitos do sistema que se deseja criar traça-se diversos diagramas que facilitam o trabalho dos programadores.

Existem diversos diagramas, mas iremos abordar apenas alguns deles nesse trabalho que são:

- Diagrama de Casos de Uso;
- Diagrama de Estados;
- Diagrama de Atividades;
- Diagrama de Componentes;
- Diagrama de Classes.

O diagrama de casos de uso é o mais simples e indica os ‘atores’ que vão interagir com o sistema planejado. Neste diagrama visualiza-se os requisitos do sistema e normalmente o planejamento do software começa por este.

O próximo diagrama é o de estados, e indica como o sistema deve funcionar de acordo com os seus estados. Em um sistema complexo o programa terá diversos estados, só saindo desses com funções específicas.

Dado isso, o diagrama de atividades deve ser feito para que entenda-se como o programa funcionará cronologicamente. É muito importante a sua análise correta para que o programa não contenha erros temporais.

Temos também o diagrama de componentes, que indica como as classes do sistema irão interagir entre si. Esse diagrama é muito útil para programas complexos e

com muitas classes, principalmente se existe uma iteração entre duas ou mais linguagens de programação.

Já o diagrama de classes mostra todas as funções e variáveis de cada classe citada no diagrama acima. Feito esse diagrama o programador pode já começar a executar o trabalho se baseando também nos outros diagramas.

2.2 Comunicação serial

A maioria das mensagens digitais são mais longas que alguns poucos bits. 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, um canal irá passar apenas um bit por vez. A transmissão bit-serial é normalmente chamada de transmissão serial, e é o método de comunicação escolhido por diversos periféricos de computadores.

A taxa de transferência refere-se à velocidade com que os dados são enviados através de um canal e é medido em transições elétricas por segundo. Na norma EIA232, ocorre uma transição de sinal por bit, e a taxa de transferência e a taxa de bit (bit rate) são idênticas. Nesse caso, uma taxa de 19200 bauds corresponde a uma transferência de 19200 dados por segundo, ou um período de aproximadamente $52 \mu\text{s}$ ($1/19200 \text{ s}$).

Outro conceito é a eficiência do canal de comunicação que é definido como o número de bits de informação utilizável (dados) enviados através do canal por segundo. Ele não inclui bits de sincronismo, formatação, e detecção de erro que podem ser adicionados a informação antes da mensagem ser transmitida, e sempre será no máximo igual a um. A figura 2.1 contém um exemplo dos dados em uma comunicação serial síncrona.

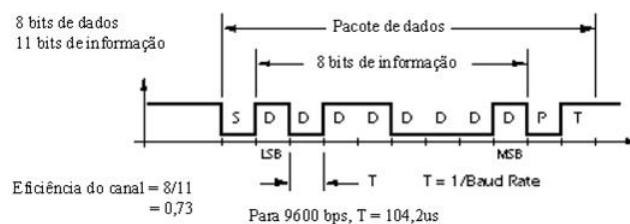


Figura 2.1: *Exemplo de mensagem serial síncrona a ser transmitida*

Geralmente, dados serializados não são enviados de maneira uniforme através de um canal. Ao invés disso, pacotes com informação regulares são enviados seguidos de uma pausa. Os pacotes de dados binários são enviados dessa maneira, possivelmente com comprimentos de pausa variável entre pacotes, até que a mensagem tenha sido totalmente transmitida. O circuito receptor dos dados deve saber o momento apropriado para ler os bits individuais desse canal, saber exatamente quando um pacote começa e quanto tempo decorre entre bits. Quando essa temporização for conhecida, o receptor é dito estar sincronizado com o transmissor, e a transferência de dados precisa torna-se possível. Falhas na manutenção do sincronismo durante a transmissão irão causar a corrupção ou perda de dados.

Duas técnicas básicas são empregadas para garantir a sincronização correta. Em sistemas síncronos, canais separados são usados para transmitir dados e informação de tempo. O canal de temporização transmite pulsos de clock para o receptor. Através da recepção de um pulso de clock, o receptor lê o canal de dado e armazena o valor do bit encontrado naquele momento. O canal de dados não é lido novamente até que o próximo pulso de clock chegue. Como o transmissor é responsável pelos pulsos de dados e de temporização, o receptor irá ler o canal de dados apenas quando comandado pelo transmissor, e portanto a sincronização é garantida.

Em sistemas assíncronos, a informação trafega por um canal único. O transmissor e o receptor devem ser configurados antecipadamente para que a comunicação se estabeleça a contento. Um oscilador preciso no receptor irá gerar um sinal de clock interno que é igual (ou muito próximo) ao do transmissor. Para o protocolo serial mais comum, os dados são enviados em pequenos pacotes de 10 ou 11 bits, dos quais 8 constituem a mensagem. Quando o canal está em repouso, o sinal correspondente no canal tem um nível lógico 1. Um pacote de dados sempre começa com um nível lógico 0 (start bit) para

sinalizar ao receptor que uma transmissão foi iniciada. O start bit inicializa um temporizador interno no receptor avisando que a transmissão começou e que serão necessários pulsos de clocks. Seguido do start bit, 8 bits de dados de mensagem são enviados na taxa de transmissão especificada. O pacote é concluído com os bits de paridade e de parada (stop bit). A figura 2.2 mostra o modo de comunicação assíncrono.

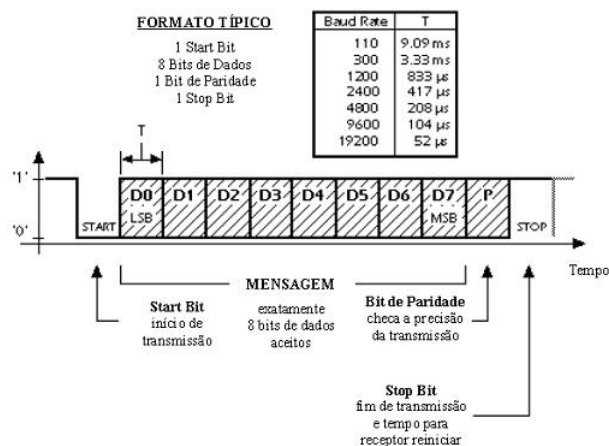


Figura 2.2: *Exemplo de mensagem serial assíncrona a ser transmitida*

2.2.1 Interface serial RS232

RS é uma abreviação de Recommended Standard. Ela relata uma padronização de uma interface comum para comunicação de dados entre equipamentos, criada no início dos anos 60, por um comitê conhecido atualmente como Electronic Industries Association (EIA). Naquele tempo, a comunicação de dados compreendia a troca de dados digitais entre um computador central (mainframe) e terminais de computador remotos, ou entre dois terminais sem o envolvimento do computador. Estes dispositivos poderiam ser conectados através de linha telefônica, e conseqüentemente necessitavam um modem em cada lado para fazer a decodificação dos sinais.

Dessas idéias nasceu o padrão RS232. Ele especifica as tensões, temporizações e funções dos sinais, um protocolo para troca de informações, e as conexões mecânicas.

Há mais de 30 anos desde que essa padronização foi desenvolvida, a EIA publicou três modificações. A mais recente, EIA232E, foi introduzida em 1991. Ao lado da mudança de nome de RS232 para EIA232, algumas linhas de sinais foram renomeadas e várias linhas novas foram definidas. Embora tenha sofrido poucas alterações, muitos

fabricantes adotaram diversas soluções mais simplificadas que tornaram impossível a simplificação da padronização proposta.

As maiores dificuldades encontradas pelos usuários na utilização da interface RS232 incluem pelo menos um dos seguintes fatores:

- ausência ou conexão errada de sinais de controle, resultam em estouro do buffer (overflow) ou travamento da comunicação;
- função incorreta de comunicação para o cabo em uso resultam em inversão das linhas de Transmissão e Recepção assim como a inversão de uma ou ou mais linhas de controle (handshaking)

2.3 Hardware e Módulo XBeeNet

Nesse capítulo, descreveremos o estudo específico sobre as Pilhas TCP/IP realizado a fim de viabilizar uma solução para agregar as leituras de cada apartamento em um mesmo lugar. Isto é, uma vez que temos um microcontrolador que recebe serialmente as informações provenientes de todos os apartamentos, via M-Bus, como fazê-lo alimentar essas informações a um computador ou mesmo à rede, via Protocolo TCP/IP?

A resposta encontrada foi a utilização de um PIC da família 18 com características e capacidade singular para armazenar a pilha de desenvolvimento TCPIP, especialmente desenvolvida pela Microchip.

Conforme mostrado na figura 2.3, a pilha é dividida em múltiplas camadas, onde cada camada tem a capacidade de acessar os serviços de uma ou mais camadas submetidas a ela.

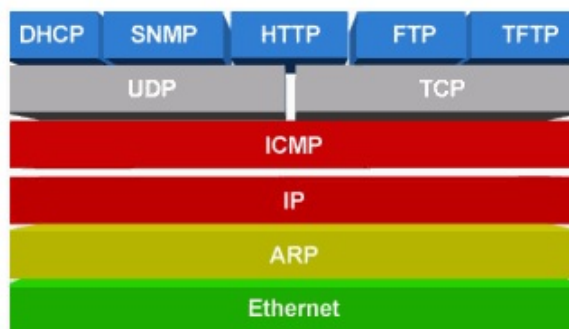


Figura 2.3: *Diferentes camadas da pilha TCP/IP*

A pilha é modular assim como demonstrado na figura 2.3 e é totalmente escrita em linguagem C. Certas implementações, por mais simples que sejam, podem chegar a consumir entre 28 e 34 kBytes de memória. Dessa forma, fica praticamente impossível o uso de microcontroladores cuja memória Flash seja inferior a 32 kBytes, parte-se da premissa que devemos sempre utilizar PICs com no mínimo 64k Bytes de memória Flash.

Caso as aplicações evoluam e a complexidade envolvida aumente, consequentemente, devemos pensar no acoplamento de memórias EEPROM responsáveis por armazenar o desenvolvimento e interface Web dentro dela, liberando assim memória do PIC para que armazene códigos de controle dos periféricos.

O microcontrolador PIC18F87J60 possui um periférico de comunicação Ethernet de 10 Mbps como principal *feature* integrada. Além disso, é compatível com o protocolo IEEE 802.3, possui memória Flash de 128 KBytes, 5 timers internos (sendo 2 deles de 8 bits e 3 de 16 bits) e periféricos de comunicação digital (2 portas A/E/USART e 1 MSSP SPI/I2C), estes últimos extremamente úteis para a recepção da informação via comunicação serial assíncrona com o M-Bus.

Tal PIC possui 80 pinos, o que tornou inviável uma confecção própria do módulo TCP/IP, por requerer instalações, equipamentos e capacidades técnicas que nós simplesmente não dispunhamos. Além disso, hoje, no mercado há centenas de módulos disponíveis e prontos para serem usados. Neles, podemos concentrar os esforços na programação das pilhas TCP/IP propriamente e deixar de lado as complicações da construção de hardware.

Dentre os módulos disponíveis, foi escolhido um bastante completo, o módulo

XBeeNet, desenvolvido pela MicroGenios ©, mostrado na figura 2.4.

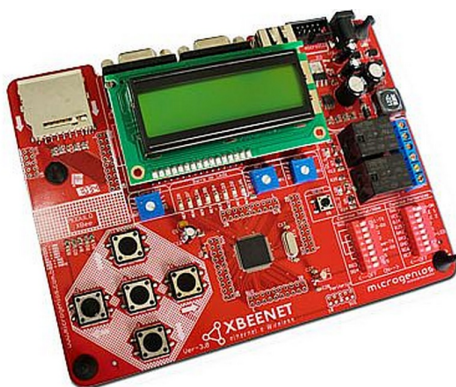


Figura 2.4: *Módulo XBeeNet*

O módulo XbeeNET é uma ferramenta de desenvolvimento profissional microcontrolada para automações industriais e residenciais. Este módulo possui muitos recursos que ajudam o desenvolvedor a criar aplicações avançadas, incluindo: Ethernet, Teclados, Leds, LCD, relés , dentre outros.

O módulo recebeu esse nome devido a suas versões prévias que contava com um módulo de rádio frequência ZigBee acoplado a ele, por onde poderia enviar e receber informações. Por não ser a nossa intenção trabalhar com essa ferramenta, ela foi descartada.

O módulo XBeeNET foi desenvolvido pensando nas aplicações TCP/IP. Compatível com a pilha TCP/IP Microchip, escrita em linguagem C, pode ser compilada pelo C18 Compiler. Aplicações típicas incluem servidor Web HTTP, controle remoto via Web, correio eletrônico, Agent SNMP, Servidor Wap, etc.

Abaixo, disponibiliza-se uma lista dos periféricos I/O que estão presentes nesse módulo.

- PIC18F87J60 (128K Flash, 3808 Byte de RAM, 55 I/Os programáveis, MAC integrado e PHY 10Bases-T.;
- Conector RJ45 (ethernet) com transformador integrado;
- Memória SPI EEPROM 25LC512;
- Canal Serial RS232 integrado;

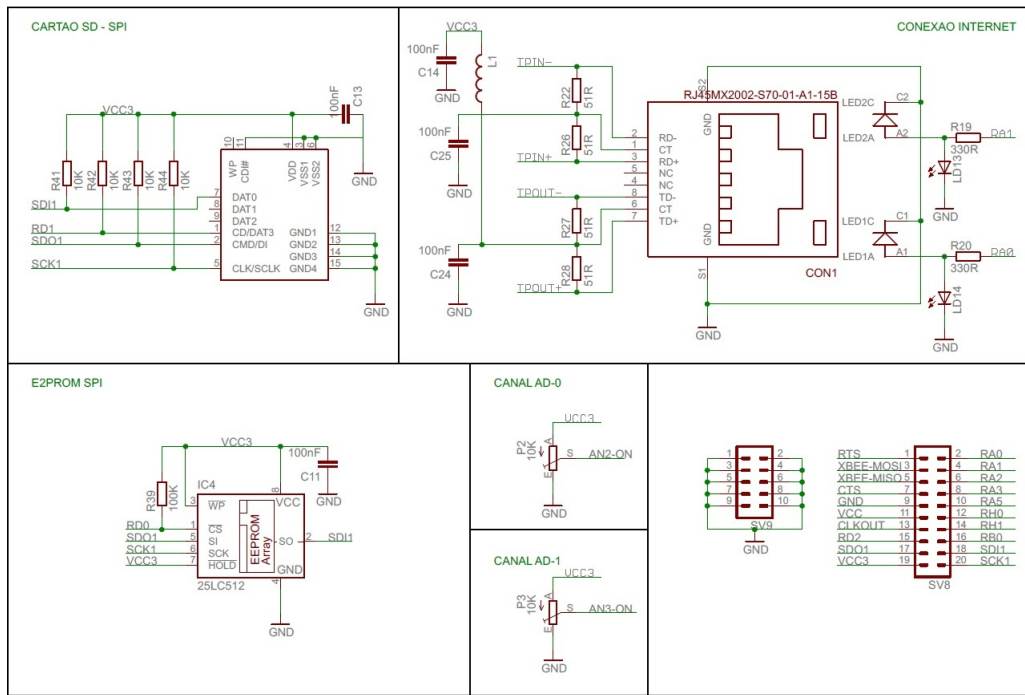


Figura 2.6: Esquema elétrico da placa XBeeNet - parte 2

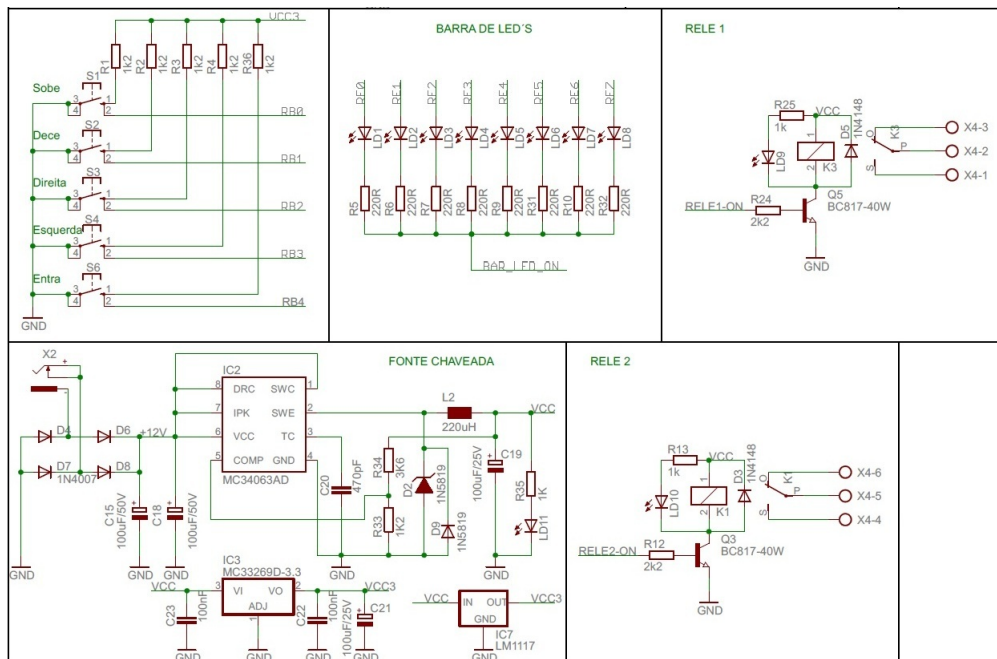


Figura 2.7: Esquema elétrico da placa XBeeNet - parte 3

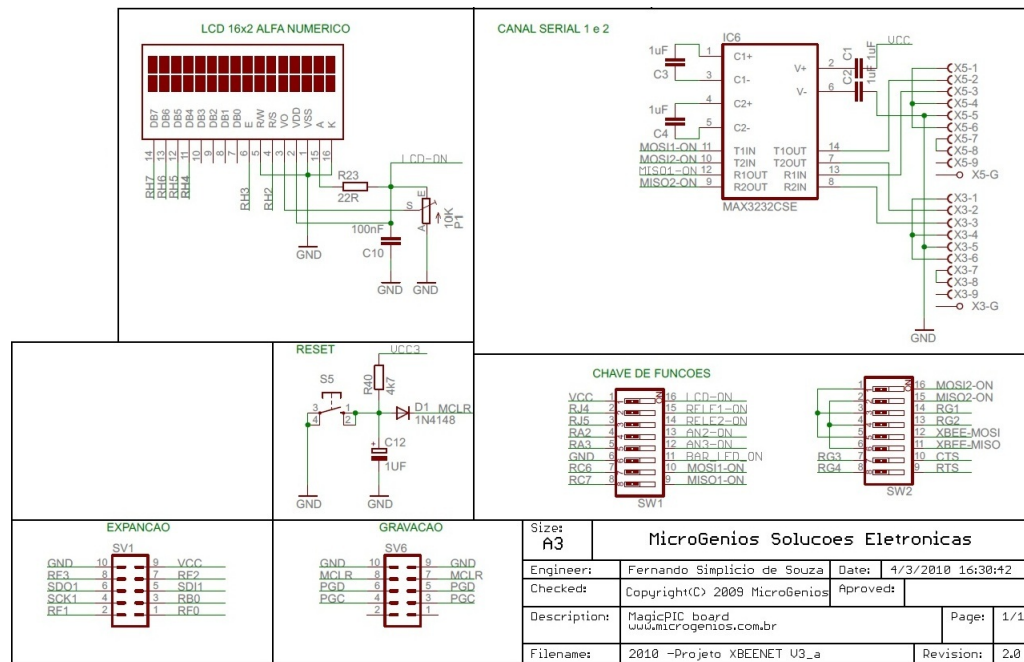


Figura 2.8: Esquema elétrico da placa XBeeNet - parte 4

3 O projeto

Para facilitar a execução do projeto pode-se dividi-lo em diversos subsistemas que serão estudados separadamente a fim de facilitar o desenvolvimento do projeto. O esquema representativo das partes pode ser identificado na figura 3.1.

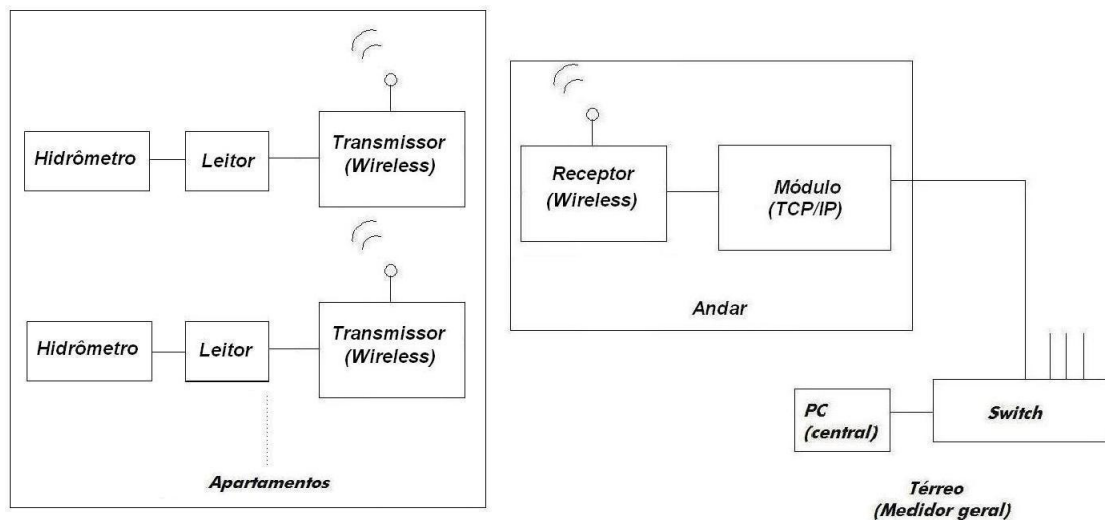


Figura 3.1: Esquema simplificado dos componentes do projeto

3.1 Hidrômetro

Apesar da construção do hidrômetro não fazer parte do projeto, torna-se importante o seu estudo para entender o seu funcionamento a fim de se fazer uma leitura perfeita.

A sua leitura tem precisão de décimos de litros, o que pode ser considerada razoavelmente alta, visto que trabalha em uma faixa de m^3 por hora. Sua leitura máxima chega a milhares de m^3 , e passando desse valor o contador é zerado, sendo importante apenas a diferença mensal entre o valor dos medidores para a cobrança. Essa característica é importante para que um tratamento digital seja implantado. Um modelo da leitura de um hidrômetro pode ser visto na figura 3.2.

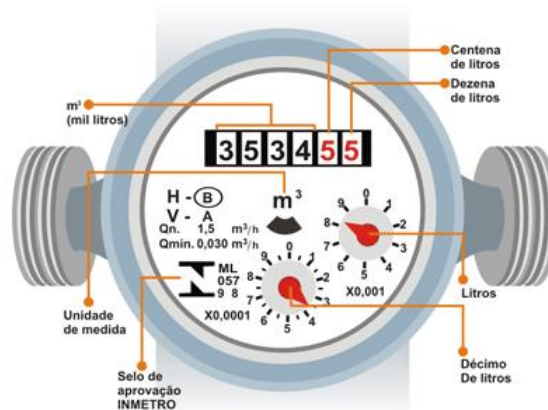


Figura 3.2: *Modelo de leitura de um hidrômetro*

Existem vários tipos de hidrômetros, podendo eles serem diferenciados quanto a alguns requisitos, como temperatura, vazão máxima de água, forma de aquisição de dados, entre outros.

Em relação à forma de leitura, o hidrômetro pode ser classificado em dois tipos: os digitais e os analógicos. O funcionamento é semelhante, transformando rotações da turbina em vazão, sendo que um deles faz a transformação através de um chip eletrônico e o outro através de um trem de engrenagens, de acordo com a figura 3.3.

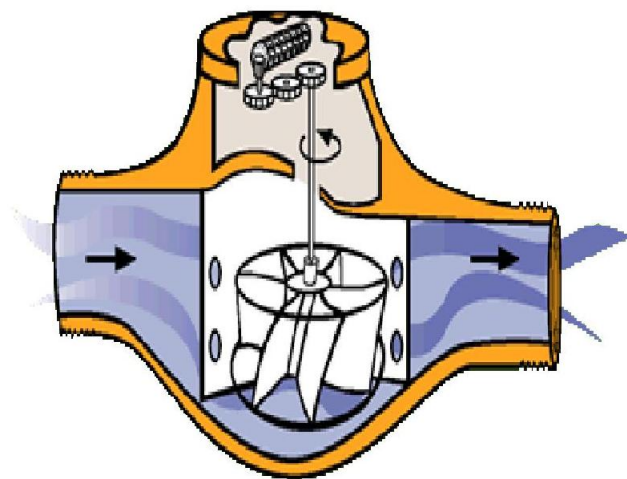


Figura 3.3: *Representação do mecanismo de um hidrômetro analógico*

Outra característica importante dos hidrômetros é a vazão máxima. No caso de uso residencial são comumente utilizados hidrômetros que permitem vazões máximas

variando de $1,5 \text{ m}^3/\text{h}$ a $3 \text{ m}^3/\text{h}$. Não há necessidade de vazões superiores a essas.

Importantes propriedades da água, como sua densidade, variam de acordo com a temperatura, que se torna um importante quesito a influenciar nas medições de consumo. Existem já no mercado dois diferentes tipos de hidrômetros residenciais desenvolvidos para medir o consumo de água de acordo com a faixa de temperatura de trabalho. Um deles é mais adequado para medir a temperaturas superiores a 40°C , e o outro para temperaturas abaixo desse valor.

Em residências a sua instalação é relativamente simples, visto que a medição é feita no momento da coleta na rua, ou seja, antes de chegar à caixa d'água. Essa situação está ilustrada na figura 3.4.

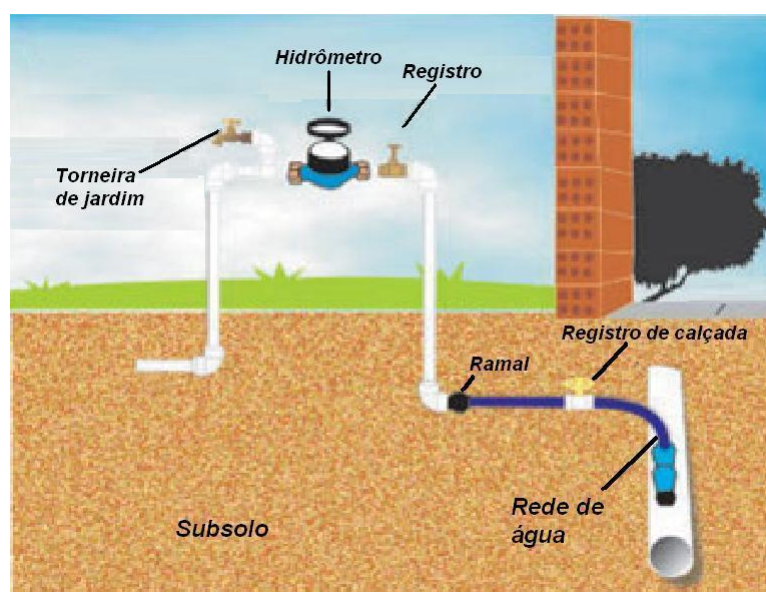


Figura 3.4: *Esquema de instalação de um hidrômetro convencional em residência*

Em prédios a medição atualmente é feita do mesmo modo, porém para a individualização ser feita muitas mudanças são necessárias. Nesse caso o hidrômetro precisa ser instalado na entrada do apartamento, depois da saída da caixa d'água, tornando imprescindível o uso de diversos medidores. A figura 1.1 exemplifica muito bem este tipo de diferença.

No caso do projeto iremos considerar que a medição será feita na entrada do apartamento, portanto antes de qualquer aquecimento de água. O tipo de hidrômetro escolhido será o digital. Sabendo que o mesmo envia pulsos elétricos a cada rotação da

turbina, podemos substituir os hidrômetros por sensores de contato a fim de facilitar os testes. Um exemplo de um hidrômetro digital pode ser visto na figura 3.5.



Figura 3.5: *Hidrômetro digital*

O funcionamento do sensor é simples. Ao selar-se o contato o sensor de contato muda seu estado de normalmente aberto para fechado e pode assim representar pulsos elétricos quando pressionado repetidamente.

3.2 Leitor

Para o processo de leitura e interpretação dos dados o melhor equipamento a ser usado é um microcontrolador. Existem diversas empresas fabricantes e um número enorme de dispositivos, variando desde a capacidade de processamento até a quantidade de funções que ele realiza.

No caso precisamos de um equipamento que além de realizar a contagem dos pulsos provenientes dos relógios do hidrômetro, consiga também converter os dados para valores conhecidos (em m^3 , por exemplo), e transmiti-los.

As funções necessárias podem ser consideradas básicas e estão presentes na maioria dos microcontroladores. Sendo assim, escolhemos o PIC18F452 que satisfaz a todos os requisitos de nosso projeto, contando com 28 portas de I/O, interrupções e USART, para que se possa efetuar tranquilamente a comunicação serial com outros aparelhos.

Disponibiliza-se a figura 3.6 que mostra a representação física em detalhes dos

PICs da família 18, com 40 pinos.

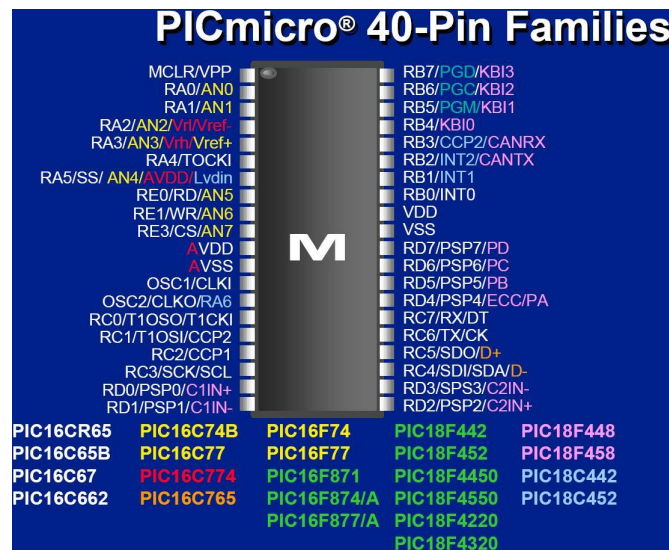


Figura 3.6: Estrutura física dos PICs da família 18

Para a programação e a gravação in-circuit dos PICs a serem utilizados nesse projeto, faz-se o uso da placa MPLab ICD2©. Com ela, adquire-se a facilidade de programar e gravar através da entrada USB.

O programa escrito e detalhado na seção 5.1 implementa todas as funções esperadas e possíveis para um leitor diretamente conectado a um hidrômetro.

Os microcontroladores podem ser programados em linguagem Assembly ou C, e posteriormente compilados e salvos em arquivos do tipo “.hex”, que então são transferidos e gravados na memória FLASH do PIC. Para tanto, utilizou-se o programa “**MikroC PRO for PIC 3.8**”, do fabricante mikroElektronika©.

Com esse código produzido, precisamos montar a placa com todos os equipamentos necessários para a leitura dos dados e a transmissão para o sistema wireless. O esquema elétrico dessa placa está na figura 3.7 e mostra como é feita a comunicação com o módulo de rádio-frequência, com o LCD, o modo de usar o botão e o LED, além do cristal de 20MHz que estamos usando.

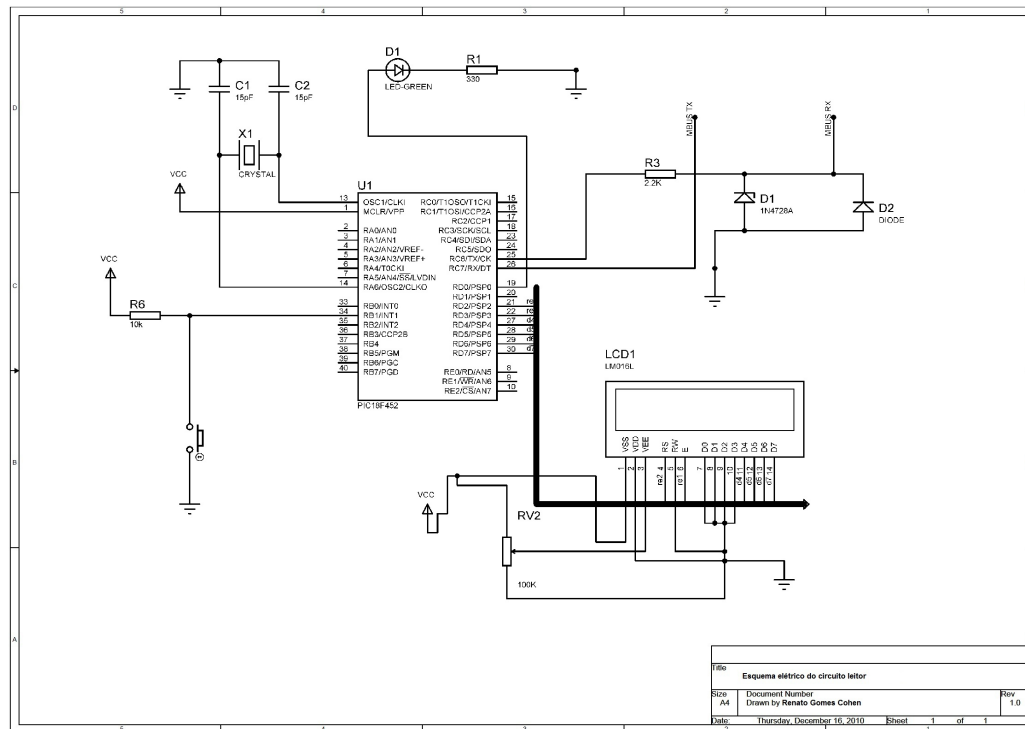


Figura 3.7: *Esquema elétrico da placa de leitura*

Outro ponto importante é a saída RX e TX do PIC que serão ligadas nos pinos TX e RX do módulo de rádio-frequência. Como veremos mais adiante, o M-Bus atua com voltagem de 0-3.6V, portanto precisamos diminuir a tensão de saída TX do pic de 5V para esses valores. Por isso o esquema da figura 3.7 com o Diodo Zener de 3.6V e o Diodo comum entre as duas entradas, fazendo com que a comunicação seja efetivada.

Para o envio das informações estamos utilizando uma interrupção de tempo, sendo assim enviamos as informações a cada intervalo de tempo para o módulo. Se faz necessário o uso de outra interrupção de maior prioridade para a contagem dos valores, no nosso caso a entrada do botão.

Por aceitar duas interrupções, sendo uma prioritária em relação a outra, é que escolhemos o PIC18F452. Essas interrupções são muito importantes para que possamos ter um melhor controle tanto na transmissão de dados quanto na contagem dos valores. A contagem dos valores tem maior prioridade em relação a do envio já que é muito mais importante não perder nenhum pulso na entrada do que deixar de enviar uma vez as informações. E como controlamos o tempo do envio das informações enviadas podemos repetir o envio a cada intervalo de tempo definido.

O LED mostrado na figura 3.7 é utilizado para mostrar quando ele entra na interrupção de tempo, piscando a cada vez. Ele é útil para verificarmos o tempo em que as mensagens estão sendo enviadas.

Para deixar claro que estamos passando os valores do PIC para a serial estamos usando um LCD, onde disponibilizamos o volume daquele leitor, podendo assim comparar com o valor final.

Antes de criar o software para esse programa, vamos projetá-lo usando a UML (Unified Modeling Language). Essa é uma linguagem que facilita muito o trabalho de programação tornando-a mais fácil e prática. Esses diagramas estão listados nas figuras 3.8 até 3.16.

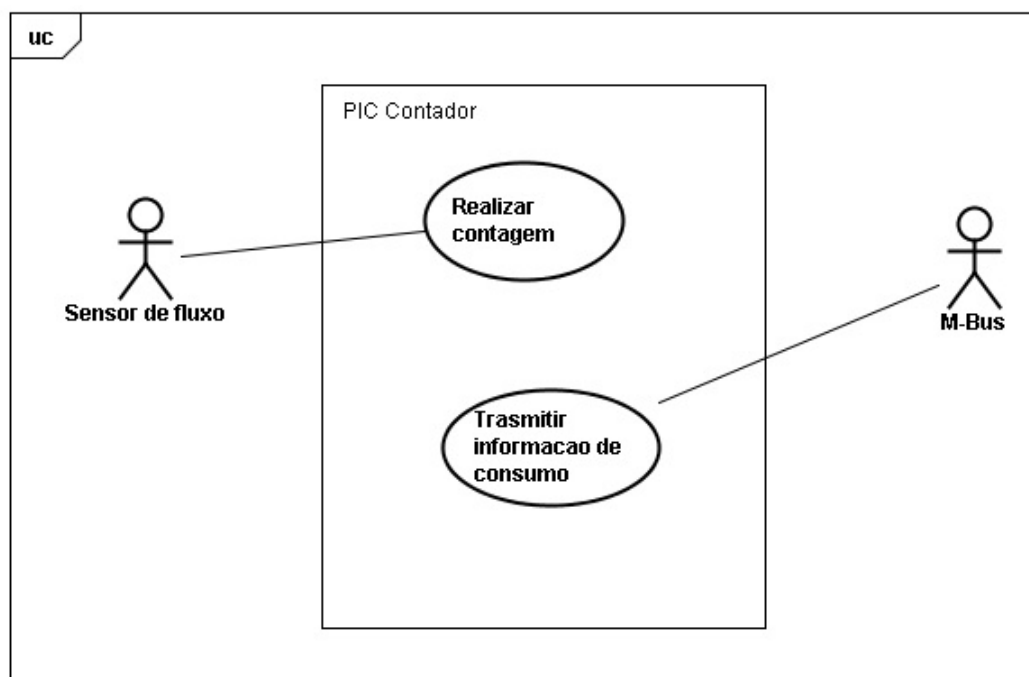
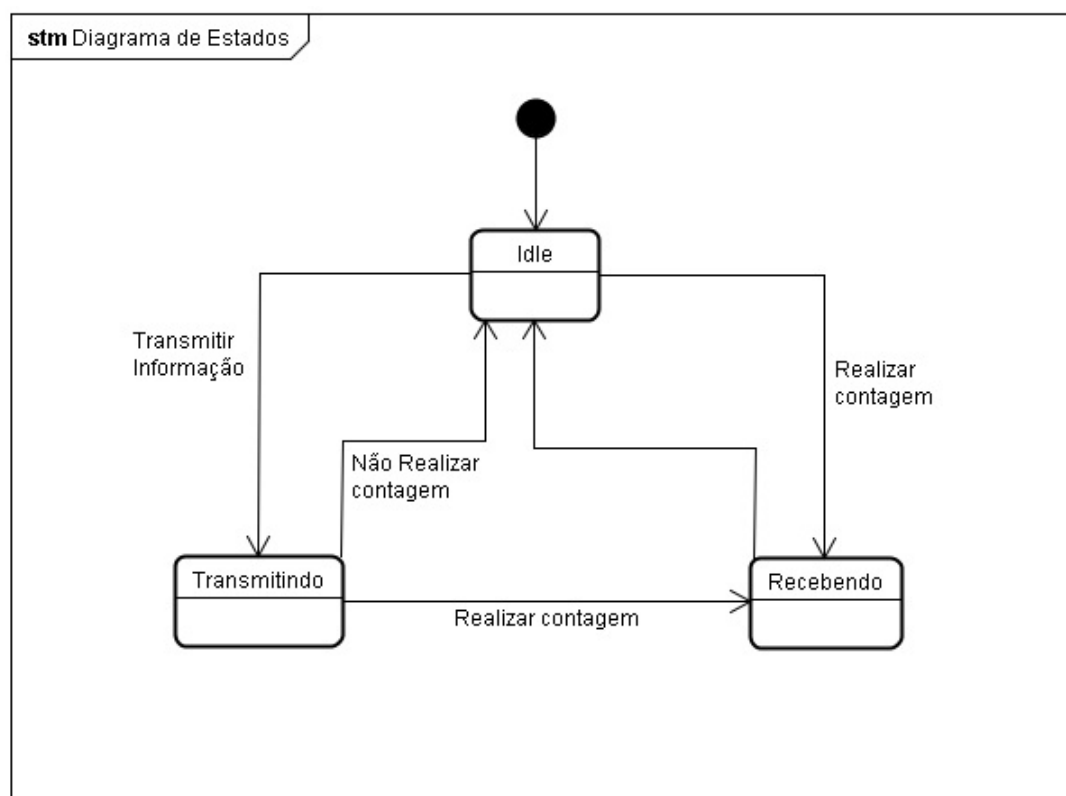


Figura 3.8: *Diagrama de casos de uso do PIC contador*

Figura 3.9: *Diagrama estados do PIC contador*

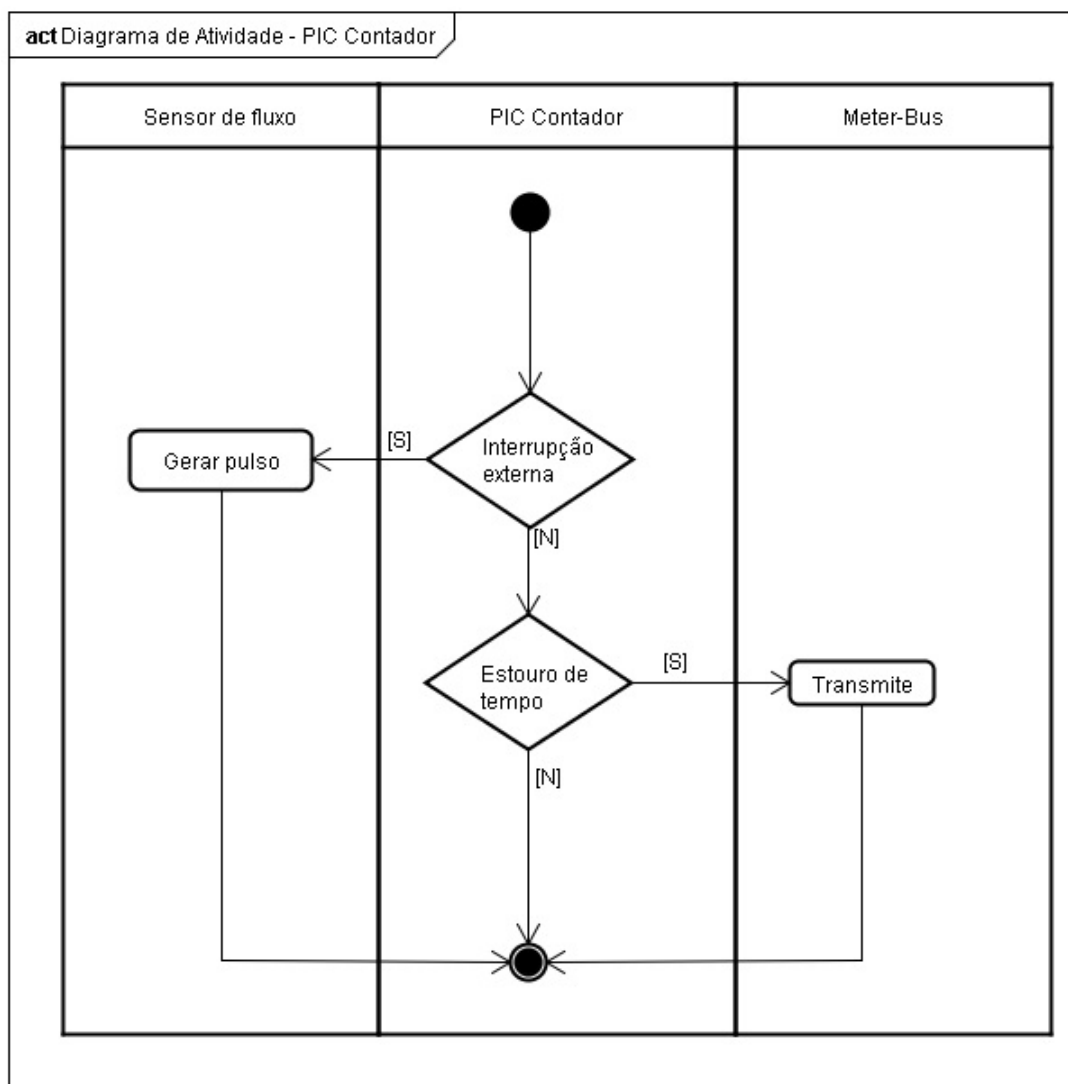


Figura 3.10: *Diagrama de atividade do PIC contador*

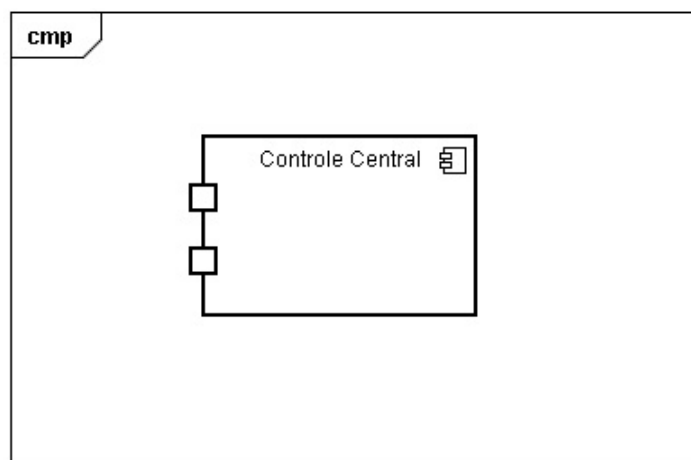


Figura 3.11: *Diagrama de componentes do PIC contador*

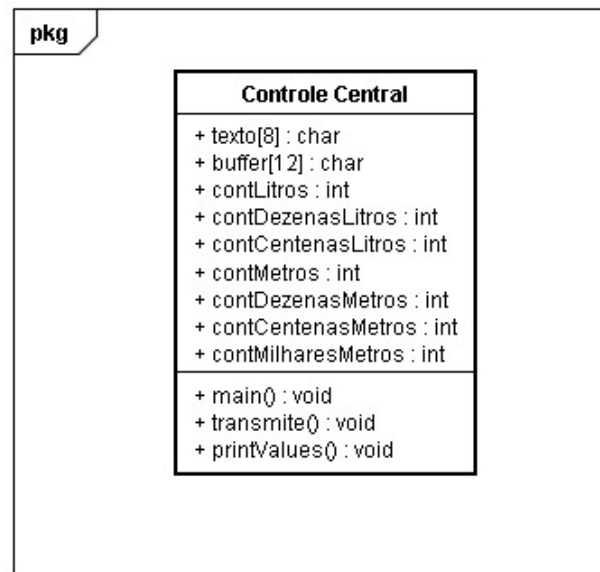


Figura 3.12: Diagrama de classes do PIC contador

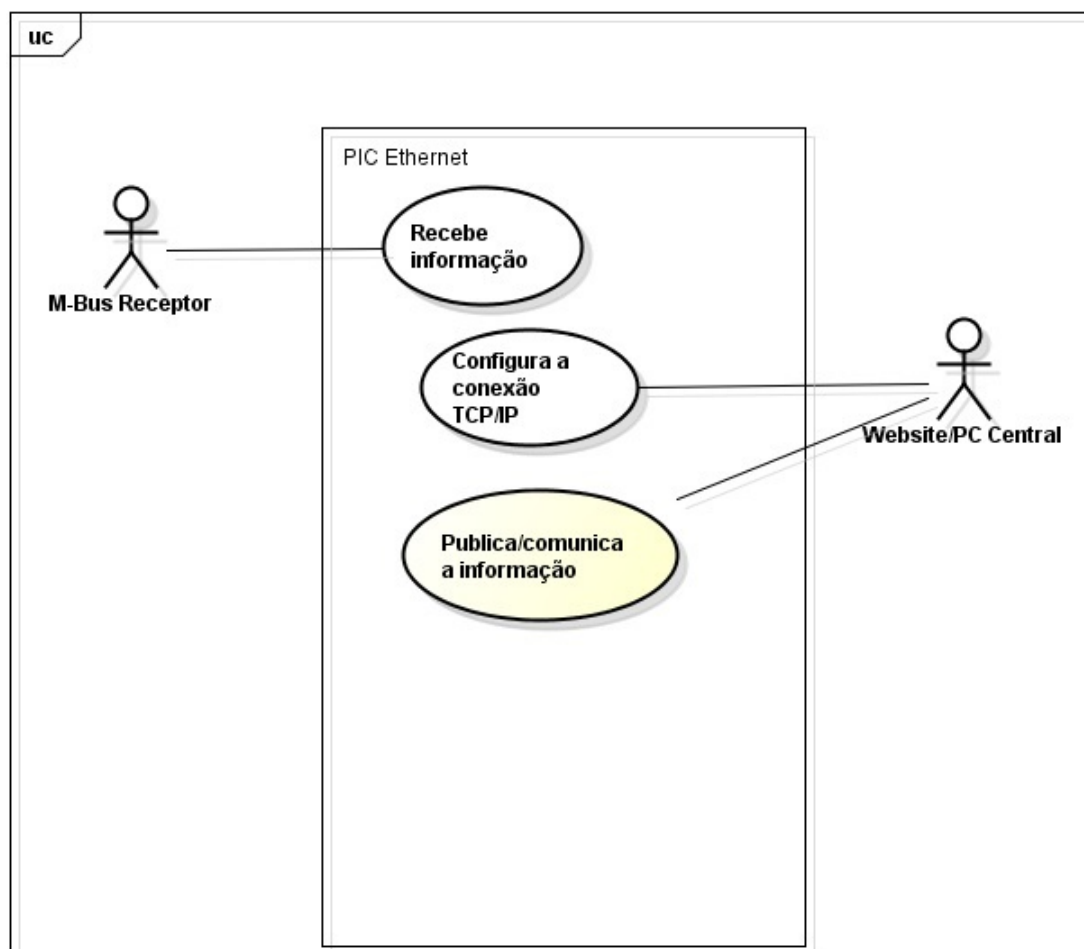


Figura 3.13: Diagrama de casos de uso do PIC TCP/IP

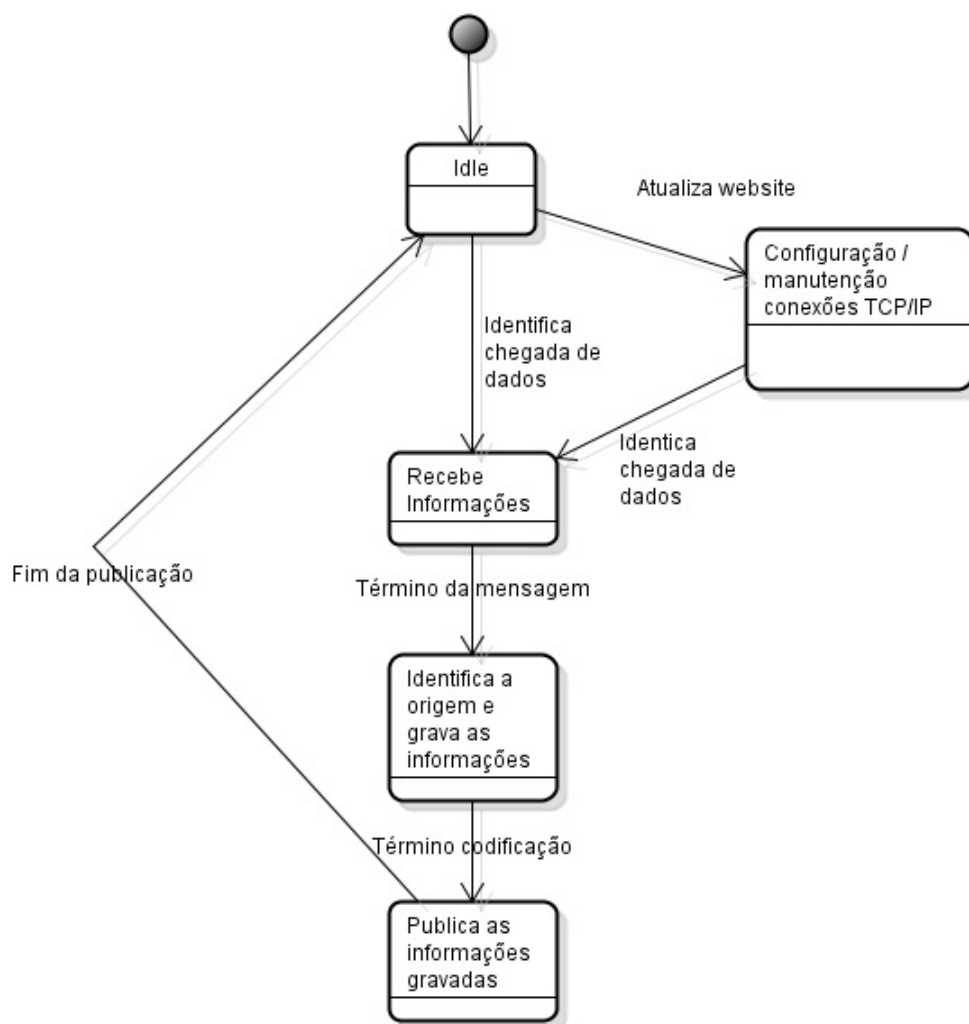


Figura 3.14: Diagrama estados do PIC TCP/IP

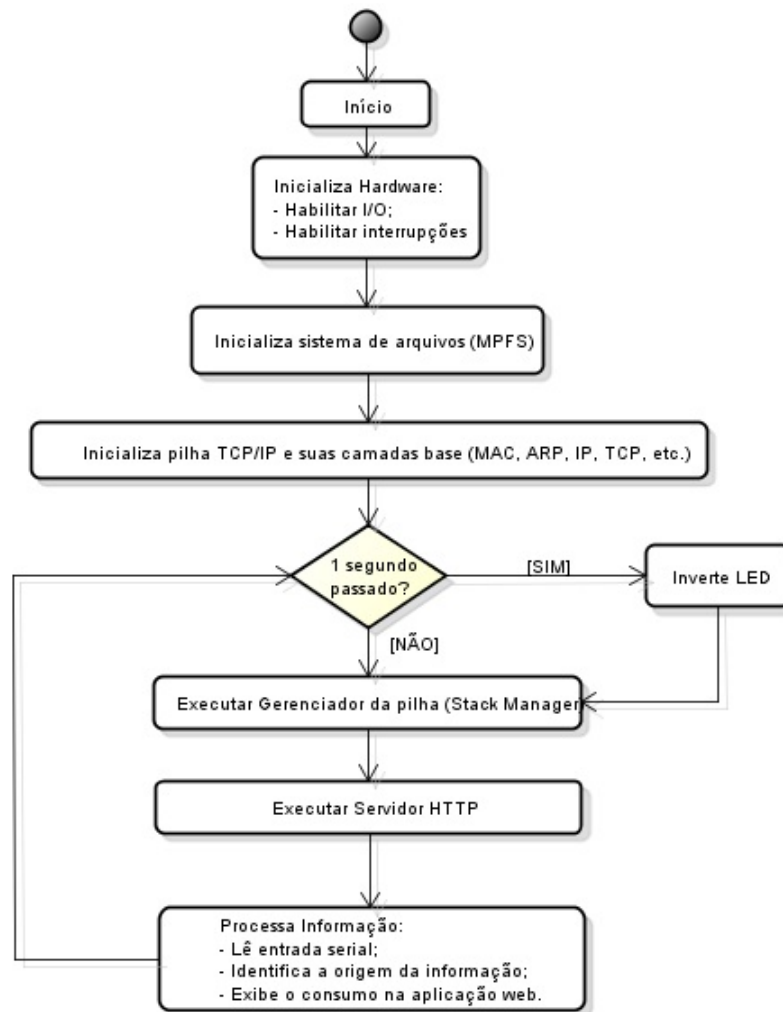


Figura 3.15: Diagrama de atividade do PIC TCP/IP

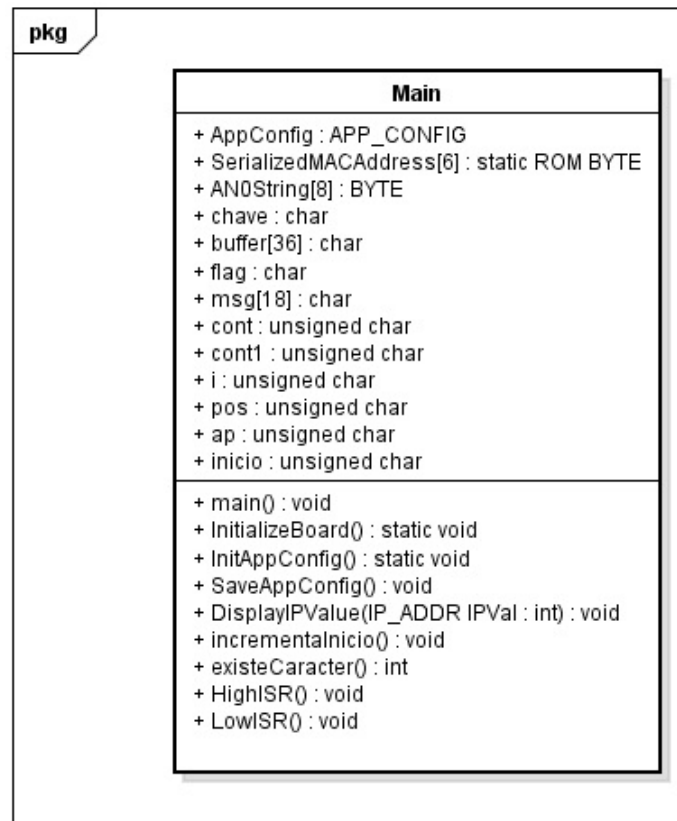


Figura 3.16: *Diagrama de classes do PIC TCP/IP*

3.3 Transmissão de dados Wireless

Feita a leitura e a conversão de giros efetivamente em volume de água consumido, passamos para a próxima etapa do projeto na qual se deve implementar a comunicação serial do PIC com o equipamento wireless que transmitirá ao receptor comum central do andar o consumo de água de cada apartamento.

Para a transmissão de dados sem fio escolhemos o RC1140, que é um módulo de transmissão wireless de alto desempenho. O protocolo de comunicação utilizado é o ISO EN13757-4:2005, amplamente usado para esse tipo de funcionalidade na Europa. Esse módulo escolhido está na figura 3.17.



Figura 3.17: O equipamento RC1140

O equipamento precisa ser alimentado com uma voltagem de 2.0-3.6V, e portanto devemos alterar a voltagem da fonte para alimentar o módulo corretamente. Para isso utilizamos o circuito da figura 3.18 que regula a tensão para o valor necessário.

O pino 'CONFIG' do módulo deve estar em nível de tensão alto para que o módulo não fique no modo de configuração, sendo assim, ligamos uma resistência de 10k Ohm do pino até o Vcc do módulo (3.6V). O circuito elétrico dessa alimentação está na figura 3.18.

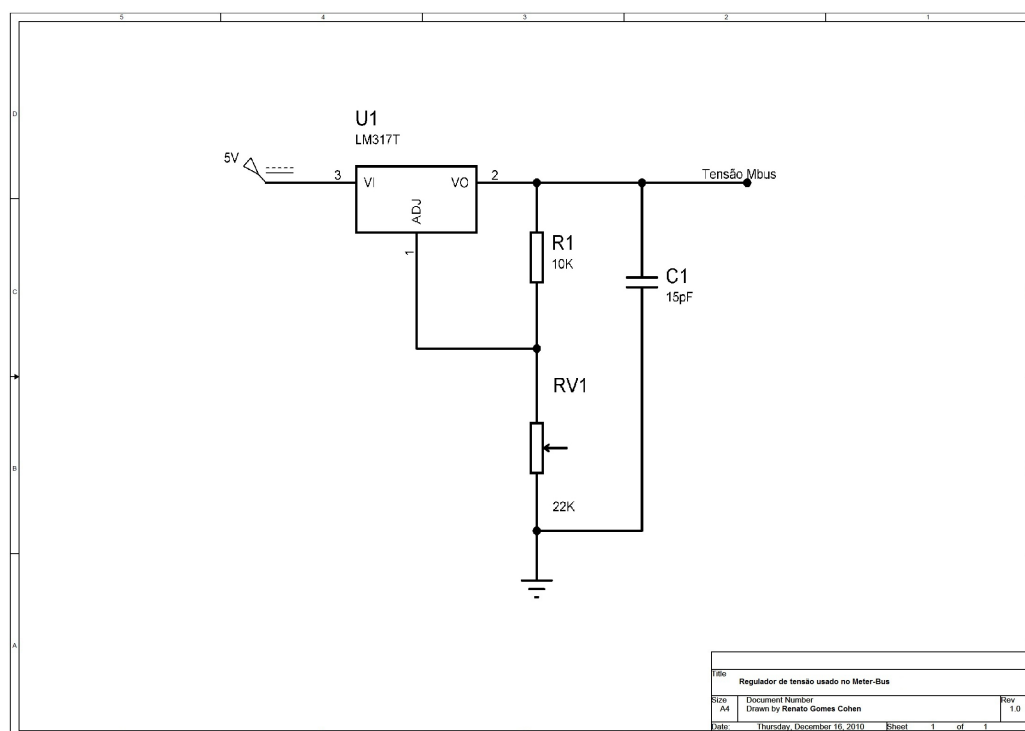


Figura 3.18: Regulação da tensão

O Meter-Bus tem basicamente 2 modos, o 'Slave' (transmissor) e o 'Master'

(receptor ou concentrador). Cada módulo tem o seu próprio endereço interno composto por 8 bits e para que o envio seja bem sucedido os módulos precisam estar acoplados, ou seja, o transmissor deve setar os valores do endereço do receptor em sua memória interna. O mesmo deve acontecer com o concentrador, sendo que podem ser acoplados até 8 transmissores diferentes .

Este equipamento utiliza frequências variadas de comunicação wireless, sendo possível utilizar 12 canais diferentes. Assim ele pode ser usado em muitas prumadas do edifício.

Para fazer a transmissão precisamos enviar a mensagem com um padrão para o módulo e entender como ele retransmite a mensagem recebida. Para o transmissor devemos enviar a informação de acordo com a figura 2.21, sendo que L é o byte do tamanho da mensagem (length) e o appl-data é a mensagem que estamos querendo enviar. Já que não iríamos utilizar o CI, deixamos o mesmo desabilitado.

Para o caso da recepção de dados percebemos pela figura que muitos outros dados foram adicionados pelo Meter-Bus para que seja identificado de onde a mensagem partiu (HEADER), que é endereço interno do módulo que está enviando. O START, CI, CRC e STOP bits estão desabilitados. O esquema da mensagem está na figura 3.19.

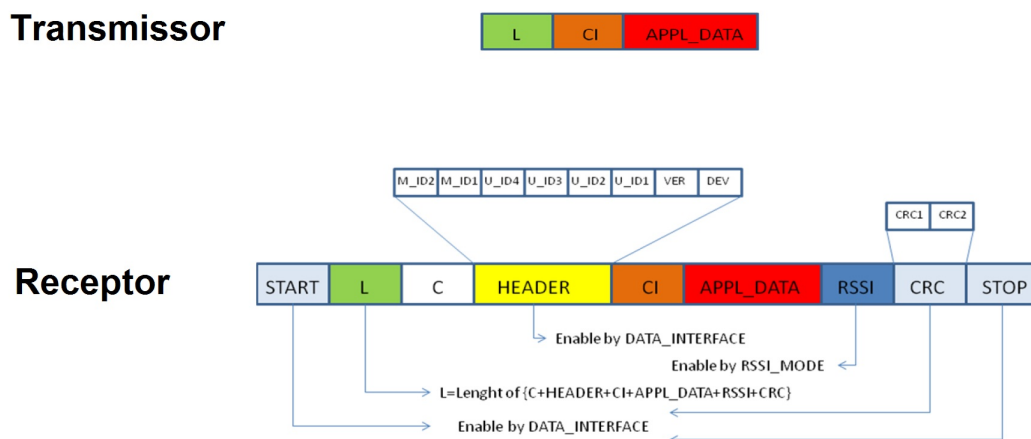


Figura 3.19: *Esquema de mensagem*

Os níveis de tensão da saída TX do módulo é de 0V para nível lógico 0 e 3.6V para nível lógico 1. Devemos transformar esses níveis de tensão para transmitir a

informação via RS232 para a placa XBeeNet e vamos usar o MAX232 para isso. O esquema de montagem desse dispositivo segue na figura 2.22. Sendo assim basta ligarmos a saída RS232 montada na placa do Meter-Bus receptor na entrada RS232 da placa XBeeNet para enviarmos a informação. A figura 3.20 contem o circuito elétrico para conexão RS232.

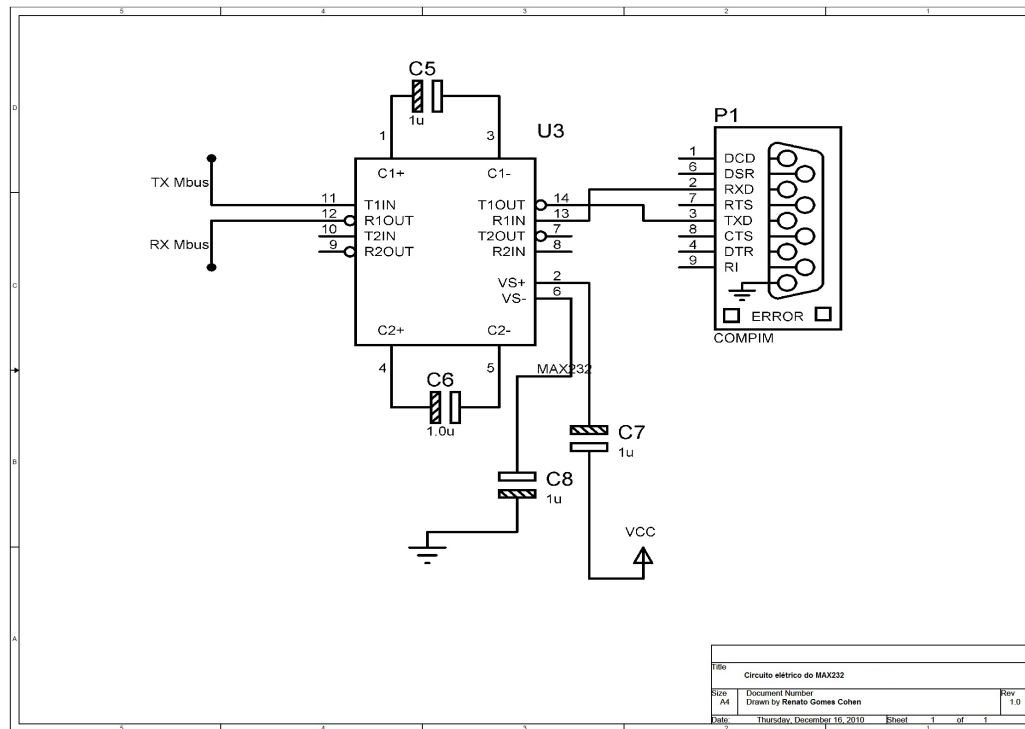


Figura 3.20: Esquema para conexão RS232

O módulo é muito poderoso e poderia até encriptar os dados para que os mesmos não possam ser 'roubados' ou alterados por outros dispositivos. No nosso exemplo não utilizamos essa encriptação pois não estamos trabalhando com usuários hostis.

3.4 Módulo TCP/IP

Iremos medir o consumo de água em diversos andares do edifício, o que torna necessário a criação de uma rede. Como o equipamento escolhido para transmissão de dados via wireless não possui essa ferramenta, usaremos um módulo com protocolo TCP/IP. O módulo escolhido é o XBeeNet que está na figura 2.4, fabricado pela Microgenios©.

Pode-se ver na imagem que o módulo possui entradas RS232 e Ethernet, 8 LEDs, 5 botões e um LCD para facilitar os testes. Com esse módulo gera-se uma página

da internet e a partir dela podemos alterar os LEDs, escrever no LCD e pegar os valores recebidos pelo Meter-Bus para mostrar ao usuário.

3.5 Central de controle

O projeto necessita de uma interface com o usuário para que os dados possam ser verificados tanto pelo morador quanto pela companhia de água responsável. Assim usaremos um computador que irá abrir a página da WEB gerada na placa XbeeNet.

Esse computador não necessita de nenhuma configuração especial, apenas precisamos ligar o cabo de rede dele na placa geradora do WEBSite para que as informações sejam trocadas corretamente.

3.6 Orçamento

Com o intuito de comercializar esse projeto precisamos analisar a sua viabilidade econômica. Sendo assim iremos fazer um orçamento de material do projeto. Nesse caso não estamos contabilizando os valores de desenvolvimento.

Como já discutimos o projeto pode ser separado em partes, e portanto iremos orçá-lo do mesmo modo. Neste caso chamamos de ‘Leitor’ o conjunto PIC Contador e Meter-Bus que envia. Contabilizamos as peças usadas no projeto e calculado sobre os seus preços unitários verificamos que o custo de cada prumada do apartamento é de R\$71,25, como podemos ver na tabela abaixo.

Componente	Quantidade	Preço Unitário	Preço Total
PIC18F452	1	R\$ 18,00	R\$ 18,00
Meter-Bus	1	R\$ 3,00	R\$ 3,00
Display LCD 16x2	1	R\$ 18,00	R\$ 18,00
Placa de fenolite tipo ilha	1	R\$ 20,00	R\$ 20,00
Cristal 20 MHz	1	R\$ 0,75	R\$ 0,75
LM317T	1	R\$ 1,20	R\$ 1,20
Trimpot 100K	1	R\$ 1,00	R\$ 1,00
Trimpot 22k	1	R\$ 1,80	R\$ 1,80
Resistores em geral	5	R\$ 0,15	R\$ 0,75
Capacitores em geral	3	R\$ 0,10	R\$ 0,30
Diodo Zener 3,6V	1	R\$ 0,15	R\$ 0,15
Diodo comum	1	R\$ 0,15	R\$ 0,15
LED	1	R\$ 0,15	R\$ 0,15
Estanho para solda	1	R\$ 2,00	R\$ 2,00
Fios	1	R\$ 4,00	R\$ 4,00
Fonte 5V	1	R\$ 25,00	R\$ 25,00
TOTAL	-	-	R\$ 96,25

Tabela 3.1: Orçamento do módulo ‘Leitor’

Temos que montar a placa que recebe os dados via wireless e a retransmite para a placa XBeeNet via serial RS232. Além de conter um menor número de componentes que o ‘Leitor’, essa placa pode-se comunicar com até outras oito placas diferentes que estão nas prumadas, reduzindo assim o custo. A lista desses equipamentos está na tabela a seguir.

Componente	Quantidade	Preço Unitário	Preço Total
Meter-Bus	1	R\$ 3,00	R\$ 3,00
Placa de fenolite tipo ilha	1	R\$ 20,00	R\$ 20,00
LM317T	1	R\$ 1,20	R\$ 1,20
Trimpot 22k	1	R\$ 1,80	R\$ 1,80
Resistores em geral	3	R\$ 0,15	R\$ 0,45
Capacitores em geral	1	R\$ 0,10	R\$ 0,10
Capacitores eletrolíticos 1uF	5	R\$ 0,15	R\$ 0,75
Diodo Zener 3,6V	1	R\$ 0,15	R\$ 0,15
Diodo comum	1	R\$ 0,15	R\$ 0,15
Conector RS232	1	R\$ 0,50	R\$ 0,50
MAX232	1	R\$ 2,90	R\$ 2,90
Estanho para solda	1	R\$ 2,00	R\$ 2,00
Fios	1	R\$ 4,00	R\$ 4,00
Fonte 5V	1	R\$ 25,00	R\$ 25,00
TOTAL	-	-	R\$ 62,00

Tabela 3.2: Orçamento do módulo ‘Receptor’

Para o nosso trabalho foi comprada a placa XBeeNet, pois foi usada para fins acadêmicos. Porém como estamos orçando apenas os equipamentos e não a mão de obra, estão listados os componentes necessários para a confecção da placa na tabela abaixo.

Componente	Quantidade	Preço Unitário	Preço Total
PIC18F87J60	1	R\$ 32,00	R\$ 32,00
Display LCD 16x2	1	R\$ 18,00	R\$ 18,00
Placa de fenolite tipo ilha	1	R\$ 20,00	R\$ 20,00
Cristal 25 MHz	1	R\$ 0,90	R\$ 0,90
EEPROM 25LC512	1	R\$ 5,00	R\$ 5,00
MAX232	1	R\$ 2,90	R\$ 2,90
Conector RJ45	1	R\$ 0,50	R\$ 0,50
Conector RS232	1	R\$ 0,50	R\$ 0,50
Trimpot 100K	1	R\$ 1,00	R\$ 1,00
Resistores em geral	10	R\$ 0,15	R\$ 1,50
Capacitores em geral	10	R\$ 0,10	R\$ 1,00
Capacitores eletrolíticos 1uF	5	R\$ 0,15	R\$ 0,75
LED	5	R\$ 0,15	R\$ 0,75
Estanho para solda	1	R\$ 3,00	R\$ 3,00
Fios	1	R\$ 5,00	R\$ 5,00
Fonte 5V	1	R\$ 25,00	R\$ 25,00
TOTAL	-	-	R\$ 117,80

Tabela 3.3: Orçamento do módulo ‘TCP/IP’

Assim como a placa anterior, utilizaremos apenas um módulo por andar, reduzindo assim o custo total do projeto caso o número de prumadas aumente. Esse módulo ‘TCP/IP’ contém bem menos componentes do que a placa XBeeNet, visto que ela é muito mais completa e com muito mais funções do que necessitamos para o trabalho. O custo final das peças do trabalho está listado abaixo.

Módulo	Quantidade	Preço Unitário	Preço Total
Leitor	2	R\$ 96,25	R\$ 192,50
Receptor	1	R\$ 62,00	R\$ 62,00
TCP/IP	1	R\$ 117,80	R\$ 117,80
TOTAL	-	-	R\$ 372,30

Tabela 3.4: Orçamento final do projeto

Esse valor é apenas uma estimativa, podendo ter uma pequena variação entre os preços dependendo da quantidade de peças que forem compradas e da loja. Como a maioria dos apartamentos tem duas entradas de água, podemos dizer que esse custo é basicamente o de cada apartamento.

Pensando nisso o custo de equipamento do projeto está viável em relação às reformas feitas em condomínios, onde basta substituírmos o botão pela saída digital do hidrômetro para termos uma solução viável.

4 Pilhas TCP/IP e módulo XBeeNet

Após a implementação da digitalização da leitura dos hidrômetros e da transmissão da informação através das placas de Metter-Bus, chega-se ao momento de armazenar a informação dentro de uma central, que também será responsável por enviar as informações seja para a Web, ou para um outro computador.

Tal equipamento foi batizado de Módulo TCP/IP e sua descrição e função no projeto foram devidamente descritas no capítulo 3, seção 3.4.

4.1 Configuração das Pilhas TCP/IP

Uma vez determinadas todas as regras de comunicação e transmissão de dados, qualquer sistema que tenha a capacidade de implementá-la estará apto a ingressar na rede. Isso significa que não existe distinção sobre o hardware por detrás dos protocolos, podendo ele ser um PC, um super-computador ou um micro-controlador.

Estudar todas as normas de todas as camadas OSI para conseguir se estabelecer um sistema capaz de atendê-la, isto é, capaz de acessar o meio físico (camadas OSI 1 e 2) através do protocolo IEEE 802.3 (Ethernet), capaz de integrar uma rede (camada OSI 3) através do protocolo IP (Internet Protocol), capaz de trocar informações (camada OSI 4) através do protocolo TCP, capaz de compreender a informação (camada OSI 7) através do protocolo HTTP e da linguagem HTML para implementar um servidor Web (ou do protocolo XML para implementar Web Service) e, finalmente, de uma aplicação capaz de utilizar tudo isso para interagir com o usuário.

Outra saída (no caso a adotada por nós nesse projeto) é a obtenção de todos os protocolos necessário já implementados na linguagem apropriada (geralmente C) para o micro-controlador desejado e, sabendo como utilizar tais pilhas, implementar a aplicação.

A segunda opção foi escolhida por possuir menor tempo de desenvolvimento e, principalmente, por evitar a reinvenção da roda. Atestando a estabilidade do sistema já construído e compreendendo seu funcionamento e limitações, abre-se possibilidades para

o surgimento de novas aplicações viáveis e, quiçá, inovadoras.

A rotina principal (main) inicializa as pilhas (MAC, ARP, UDP, TCP) e entra no loop de operação. No loop, pisca-se um led do sistema, checa-se se há pacotes recebido, verifica o tipo do pacote e chama o método apropriado para processá-lo.

Após isso, executa o servidor HTTP que gerencia e serve cada conexão aberta nos sockets (pode-se executar um servidor FTP no lugar).

Executa um NBNS (NetBios Name Server), que é similar a um DNS (domain name server), respondendo a requerimentos que ocorram. Por fim, pode-se rodar um cliente TCP genérico.

Para dar suporte a essas operações, quatro outras funções importantes foram observadas a serem destacadas.

Ressalta-se que a verificação e processamento de pacotes ocorre por pooling (uma vez por ciclo de programa) e não por interrupção. Assim, o aplicativo do desenvolvedor deve ser implementado de maneira multi-tasking cooperativo, isto é, não deve consumir excessivamente a CPU.

A pilha TCP/IP, foi desenvolvida e fornecida gratuitamente pela Microchip e são compatíveis com as famílias 8, 16 e 32 bits de microcontroladores PIC.

Para configurá-la de acordo com a nossa necessidade de aplicação é necessário ter o conhecimento de alguns arquivos chave que a integram e conhecer exatamente onde configurar tais recursos.

Os arquivos que utilizamos e alteramos para obter nossa aplicação funcionando corretamente, e suas respectivas funções estão listados abaixo:

- *HardwareProfile.h* e *HardwareProfile.c* - nesse arquivo, está presente toda a configuração física do hardware no qual o microcontrolador está inserido. Aqui caracteriza-se os vários modelos de kits e módulos para aplicações da pilha TCPIP no PIC, com ênfase para o módulo *XBeeNet* e o PIC18F87J60, utilizados em nosso projeto;
- *LcdBlocking.h* e *LcdBlocking.c* - caracteriza-se nesses códigos o equipamento de LCD que está sendo utilizado, seu modo de operação (4 ou 8 bits), o Port dos bits e seus sentidos, e as funções de acesso e manipulação de variáveis do LCD, como a

inicialização, a escrita e a atualização;

- *HttpPrint.h* e *HttpPrint.c* - arquivos que relacionam diretamente as variáveis que desejamos trabalhar e mostrar em nossa aplicação Web às variáveis e métodos internos do microcontrolador;
- *MainDemo.h* e *MainDemo.c* - método main e loop interativo com a máquina de estados que mantém a funcionalidade de nossa aplicação, bem como a configuração dos estados iniciais de portas I/O do PIC e de seus principais registradores;
- *TCPIPConfig.h* e *TCPIPConfig.c* - códigos responsáveis pelo endereçamento e opções de transporte entre as diferentes camadas da pilha TCPIP;

Não é nossa intenção aqui entrar em detalhes sobre cada alteração feita em cada um desses códigos a fim de obter êxito em nossa configuração de Hardware.

Nas próximas seções descreveremos resumidamente na mesma ordem que implementamos, os passos tomados para a configuração de alguns desses arquivos, no caso, os mais relevantes.

4.1.1 *HardwareProfile.h*

Inicialmente, configuramos o arquivo *HardwareProfile.h* de definição física da placa para que reconhecesse todos os periféricos de nossa placa. Iniciamos assim, definindo a placa que estávamos usando. Definimos nosso projeto através do nome de nosso projeto ou através do modelo do microcontrolador que estamos utilizando.

A seguir definimos em nosso código os bits de configuração de nosso chip, também chamados de *fuses* do microcontrolador. Feito isso, definimos a configuração de *clock* do microcontrolador, na qual se basearão todos os contadores e frequências dos processos internos do chip. Obviamente, a frequência é definido pelo cristal ressonador que utilizamos no módulo, nesse caso 25 Mhz:

Nesse momento, podemos definir em quais portas estão conectados as diversas entradas e saídas I/O do hardware. No nosso caso há a necessidade de definir as portas de LEDs, teclas, display LCD, memórias EEPROM, Relés e canal de expansão ENC28J60 (que também pode ser conectado à placa). Este último, permite que estabeleçamos uma

conexão TCPIP através de um canal SPI e nesse caso, não seria necessário um microcontrolador da Microchip, outros microcontroladores poderiam ser usados para estabelecer uma comunicação Ethernet.

Essas são todas as configurações físicas de nossa placa *XBeeNet* que precisávamos adaptar. Partamos agora para a configuração das condições iniciais de portas e registradores do microcontrolador.

4.1.2 *MainDemo.c*

Como já dito anteriormente, nesse arquivo ficam definidas as condições iniciais das portas I/O do microcontrolador, os valores caso venhamos a utilizar os diversos registradores, interrupções, *timers*. Como pode ser dedutível pelo nome, dentro desse arquivo escrevemos o nosso código principal *main*, juntamente com o loop infinito e a máquina de estados responsável por estruturar nossa aplicação principal.

Para a recepção dos dados via serial utilizamos a interrupção de maior prioridade, que armazena esses dados em um vetor circular. O seu funcionamento é um pouco diferente de um vetor comum. Ele adiciona na pilha os valores recebidos, desde que a pilha tenha espaço suficiente para esses dados, onde ao chegar ao final do tamanho do vetor ele volta ao começo. Ele escreve nas posições vazias do vetor, que somente são zeradas quando o mesmo é lido, sendo assim não perdendo informações.

4.1.3 *LCDBlocking.h* e *LCDBlocking.c*

A fim de que pudéssemos completar a configuração de nosso módulo XBeeNet ainda restava setar as configurações específicas de nosso display LCD.

Apesar das pré-configurações deixadas pela Microchip ainda necessitávamos atualizar alguns métodos de acesso ao display LCD, reconfigurando o modo de 4 bits, a função de inicialização, a função de escrita e, por último, de atualização do LCD.

4.2 WebSite gerado pela XBeeNet

A placa de desenvolvimento citada acima está comunicando com o computador e gerando uma página da WEB para servir como a interface com o usuário. Através do site podemos alterar os LEDs e mudar o valor do LCD, além é claro de ver os valores medidos dos hidrômetros. A figura 4.1 mostra o modelo de página da WEB gerada pela placa XBeeNet.



Figura 4.1: *Página Home do WebSite*

No nosso site obrigamos o usuário a fazer o login e a senha para acessar as páginas Leds, LCD e Hidrômetros. Para facilitar a execução do projeto esses valores estão sendo impressos na tela nos lugares de login e senha, que nessa linguagem indica que são variáveis.

Indicamos também na página principal a versão da Stack TCP/IP que estamos usando e a data de compilação do projeto, que são substituídos nas outras variáveis (version e bulldate).

Na página do LCD podemos atualizar o LCD e escrever uma mensagem com até 32 caracteres. O nosso programa então divide a mensagem e escreve os primeiros 16 caracteres na primeira linha do LCD e o restante na segunda linha. O desenho da página da WEB está na figura 4.2.

Figura 4.2: *Página LCD do WebSite*

Como podemos ver na figura 4.3 a página dos hidrômetros atualiza automaticamente assim que os valores são recebidos pela porta serial. Estamos utilizando para teste apenas dois apartamentos, mas isso poderia ser expandido para o número de apartamentos necessário.

Figura 4.3: *Página Hidrometros do WebSite*

Para ficar com esse formato a página Web precisou usar um arquivo do tipo .css (Cascading Style Sheet Document) que seta as informações como plano de fundo, cor da letra, tamanho da letra, entre outros.

Todas as variáveis do site citadas acima que estamos alterando na pilha TCP/IP devem estar em um arquivo do tipo '.xml'. Temos que ter um outro arquivo do tipo '.cgi' que deve estar vazio para receber os valores.

A figura 4.4 mostra como carregamos a página da Web no nosso módulo XBeeNet. Estamos usando o programa 'Microchip MPFS Generator' para a compilação do

código HTML.



Figura 4.4: *Esquema de carregamento da página do WebSite*

5 Conclusão

Com a conclusão do projeto, nós acreditamos ter conseguido provar a viabilidade na adaptação de leitura individualizada do consumo de água em instalações mais antigas. Trata-se de uma solução razoavelmente simples, mas que no entanto necessita de mais adaptações para que se torne uma solução comercial, visto que nosso objetivo inicial era somente o desenvolvimento de uma solução acadêmica.

Com a tecnologia disponível no mercado e um conhecimento razoável de programação, em especial de microcontroladores PIC, é possível montar e instalar com certa facilidade cada um dos sistemas explicados nesse trabalho.

Sabemos que o escopo desse trabalho não é mais uma grande preocupação nas instalações novas, que já fazem uso de soluções prontas, disponibilizando um medidor de água, luz e gás para cada apartamento. Ainda assim, nosso trabalho pode ser útil e aproveitado mesmo pelas instalações mais recentes e modernas caso o objetivo seja gerenciar as informações de todos os usuários em um mesmo ponto, seja ele um computador ou mesmo através da *Web*.

Como sugestão para possíveis futuros trabalhos que venham a desenvolver nessa mesma linha, sugerimos que seja implementado o banco de dados que abrigue todas as informações de consumo, periodicamente, a fim de facilitar a gestão do consumo para cada usuário.

Podemos verificar que além de inovador, o projeto é muito prático, tornando até possível a comercialização do produto logo após a sua construção.

O projeto foi bastante prático e nos permitiu estar em contato intenso com conteúdos de diversas disciplinas da graduação. Em especial, conceitos de Eletrônica Analógica e Digital, na construção dos circuitos de alimentação, na interpretação de diversos *Datasheets* e na depuração de diversos erros.

Não podemos esquecer da extrema importância da disciplina de Microcontroladores, igualmente, que nos forneceu a base para toda a programação desenvolvida aqui.

De modo similar, não menos importante foi a matéria de Sistemas de Informação, abrindo-nos o horizonte para a integração de sistemas embarcados com aplicações Web.

Desenvolver tal projeto foi bastante desafiador e demandou bastante trabalho, mas foi sem dúvidas uma experiência ímpar de aprendizado mais avançado e aplicado de todas as disciplinas acima.

Estamos muito contentes com o resultado final e esperamos ter contribuído, ainda que de uma forma simples, para o desenvolvimento da automação residencial.

Referências Bibliográficas

- [1] PEREIRA, Fábio, *Microcontroladores PIC - Programação em C*, 5ª edição, Érica, São Paulo.
- [2] SEIXAS FILHO, Constantino; SZUSTER, Marcelo *Programação Concorrente em Ambiente Windows: uma visão de automação*, 1ª edição, UFMG.
- [3] GUEDES, G. *UML: uma abordagem prática*, 3ª edição, Editora Novatec, 2008.
- [4] <http://www.pura.poli.usp.br/main> , visitado em 01/06/2010 às 21:05.
- [5] <http://www.objectsbydesign.com/tools/umltoolsbyCompany.html> , Unified Modeling Language, visitado em 15/03/2011, às 22:20.
- [6] <http://radiocrafts.com/index.php?sideID=443>, visitado em 15/05/2011, às 09:27.
- [7] <http://www.ajmartani.com.br/servicos.htm>, visitado em 30/06/2011, às 09:30.
- [8] http://www.labcenter.co.uk/products/vsm_verview.cfm, Proteus VSM, visitado em 30/12/2010, às 14:50.
- [9] http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2505¶=en535724, Microchip: TCP/IP download and support, visitado em 01/06/2011, às 16:21.
- [10] <http://ww1.microchip.com/downloads/en/AppNotes/00833c.pdf>, Microchip TCP/IP Stack Application Note, visitado em 01/06/2011, às 18:55.
- [11] ww1.microchip.com/downloads/en/DeviceDoc/39762a.pdf, PIC18F97J60 Family Device Data Sheet, visitado em 25/05/2011, às 18:55.
- [12] http://www.radiocrafts.com/uploads/rc11x0-mbus_data_sheet_2_21.pdf, Radiocrafts RC11X0 M-Bus Datasheet, visitado em 19/03/2011, às 14:32.
- [13] http://www.radiocrafts.com/uploads/mbus_user_manual_1_40.pdf, Radiocrafts RC1140 M-Bus User Manual, visitado em 21/03/2011, às 10:49.

-
- [14] <http://www.datasheetcatalog.org/datasheet/texasinstruments/lm317.pdf>,
LM117/LM317A/LM317 3-Terminal Adjustable Regulator Datasheet, visitado
em 04/07/2011, às 18:17.