



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

ELIAS MACÊDO MACIEL JÚNIOR

**ACOPLAMENTO DE ATUADORES AO PROJETO DO
SIMULIHM**

Campina Grande, Paraíba.
Outubro de 2012

ELIAS MACÊDO MACIEL JÚNIOR

ACOPLAMENTO DE ATUADORES AO PROJETO DO
SIMULIHM

*Trabalho de Conclusão de Curso submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.*

Orientadora:

Professora Maria de Fátima Queiroz Vieira, Ph. D.

Campina Grande, Paraíba.
Outubro de 2012

ELIAS MACÊDO MACIEL JÚNIOR

ACOPLAMENTO DE ATUADORES AO PROJETO DO SIMULIHM

Trabalho de Conclusão de Curso submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Aprovado em / /

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professora Maria de Fátima Queiroz Vieira, Ph. D.
Universidade Federal de Campina Grande
Orientadora, UFCG

Dedico este trabalho à minha namorada
Giovanna, companheira fiel, dedicada,
corajosa e exemplar.

AGRADECIMENTOS

Agradeço a Deus, em primeiro lugar, pela minha vida e pelo dom da perseverança, que me permitiu concluir este trabalho.

Agradeço também à minha mãe, Rita e meu pai Elias, por ter se esforçado tanto para me proporcionar uma boa educação, por ter me alimentado com saúde, força e coragem, as quais que foram essenciais para superação de todas as adversidades ao longo desta caminhada.

Agradeço a Professora Fátima pela paciência e compreensão e por ser uma excelente orientadora.

Agradeço também a toda minha família, que com todo carinho e apoio, não mediu esforços para eu chegar a esta etapa da minha vida.

Enfim, agradeço a todos que de alguma forma, passaram pela minha vida e contribuíram para a construção de quem sou hoje.

RESUMO

Este trabalho consistiu em utilizar as tecnologias de redes sem fio e microcontroladores, com a finalidade de construir uma interface física para interação com o simulador virtual atualmente em desenvolvimento no Laboratório de Interfaces Homem Máquina (LIHM) do DEE UFCG, o qual foi concebido para o treinamento de operadores de subestações do sistema elétrico do sistema elétrico. Para isso foi realizada uma pesquisa bibliográfica para fundamentar a escolha da tecnologia de rede sem fio e dos componentes de hardware adequados ao projeto. Em seguida foi feita a especificação e desenvolvimento do hardware dos atuadores visando a integração com o simulador. O hardware foi concebido utilizando a tecnologia de microcontroladores e contou com o uso de simuladores para validar o projeto. Foram realizados testes de funcionalidade para verificar a abrangência do resultado tanto em simulação quanto com as montagens, resultando em sugestões para futuros trabalhos.

Palavras-chave: Redes sem fio, Microcontroladores, Simulador para treinamento de operadores.

ABSTRACT

This work consisted in using the technologies of wireless networks and microcontrollers, aiming to build a physical interface and integrate it into a virtual simulator currently under development at the Laboratory of Human Machine Interfaces (LIHM) DEE UFCG, for training for substations operators. The project begun with a bibliographic review to support choosing the wireless networking technology to be adopted as well as choosing the adequate hardware components. The following step consisted in developing the specified hardware with the aid of a simulating tool which was essential to test the solution before implementing it. Finally, from the discussion of the test results, it is proposed in this report a set of suggestions for future work.

Keywords: Wireless Networks, Microcontrollers, Operator training Simulator.

LISTA DE ILUSTRAÇÕES

Figura 1 – Sala de controle virtual de subestação de um sistema elétrico	4
Figura 2 – Ambiente tutor no SimuLIHM	4
Figura 3 - Antena tipo chicote ou whip	6
Figura 4 - Antena tipo chip.....	7
Figura 5 – Rede mesh zigbee.....	9
Figura 6 – Criando um novo projeto no PIC c.....	11
Figura 7 – Configurações gerais do PIC.....	11
Figura 8 – Linhas de código geradas apartir das escolhas.	12
Figura 9 – Diagrama de blocos da arquitetura do projeto.....	16
Figura 10 - Painel que será replicado no PROJETO (02J5).	17
Figura 11 – Diagrama de caso de uso.....	18
Figura 12 – Circuito simulado no ambiente PROTEUS.....	21
Figura 13 - INTERFACE DO APLICATIVO X-TCu.....	23
Figura 14 – Pinos do xbee	23
Figura 15 – Montagem com componentes reais	24
Figura 16 - Montagem com componentes reais.....	24
Figura 17 - Diagrama de blocos do protótipo.....	25
Figura 18 – Placas para a montagem do protótipo.....	25
Figura 19 – Montagem do protótipo.....	26
Figura 20 – Diagrama de pinos do PIC17F877	32
Figura 21 – Diagrama de fluxo.....	33
Figura 22 - \ Diagrama de fluxo	36

LISTA DE TABELAS

Tabela 1 - comparação dos principais protocolos de redes sem fio	15
Tabela 2 - Especificação de parâmetros do XBee e XBee-Pro.....	31

SUMÁRIO

1	Introdução.....	1
1.1	Objetivo	1
1.2	Motivação	2
1.3	Organização do relatório.....	2
2	Apresentação dos conceitos utilizados	3
2.1	Apresentação do SimuLIHM.	3
2.2	Redes sem fio.....	5
2.2.1	Rede ZigBee padrão IEEE 802.15.4.....	5
2.3	O Microcontrolador PIC 16F877a	9
2.3.1	Programando o PIC utilizando o ambiente de desenvolvimento PIC C.	11
3	Especificação e Desenvolvimento da Interface	13
3.1	Protocolo de rede e microcontrolador.....	13
3.1.1	Escolha do microcontrolador.....	13
3.1.2	Escolha do protocolo de rede sem fio.....	15
3.2	Especificação do projeto.....	16
3.2.1	Descrição da arquitetura.....	16
3.2.2	Especificação do painel a ser simulado.....	17
3.2.3	Especificação do Software embarcado.....	18
3.2.4	Especificação do Microcontrolador.....	19
3.2.5	Especificação dos módulos XBee.....	20
3.3	Implementação do Projeto.....	20
3.3.1	O projeto de Hardware.....	20
3.3.2	Simulação com o Proteus PCB design.....	21
3.3.3	Os módulos XBee.....	22
3.3.4	Montagem do protótipo.....	24
4	Testes e resultados.....	27
4.1	Software embarcado.....	27
4.2	Hardware.....	27
5	Considerações e sugestões para trabalhos futuros.....	29
	Bibliografia.....	30
	Anexo 1 - Especificação de parâmetros do XBee e XBee-Pro	31
	Anexo 2 - Diagrama de pinos do PIC17F877.....	32
	Apêndice 1 – Programas em C do <i>software</i> embarcado no PIC.....	33
	PIC 1 – Protótipo.....	33
	PIC 2 – Planta.....	36

1 INTRODUÇÃO

Para acompanhar a evolução tecnológica no setor elétrico garantindo segurança e confiabilidade ao sistema elétrico, as empresas têm a necessidade de continuamente capacitar seus profissionais. O treinamento para a capacitação destes profissionais na maioria das vezes não é viável em painéis reais de subestações do sistema elétrico, neste caso, é comum que as empresas utilizem simuladores dos painéis reais para tal treinamento.

Em paralelo com o avanço ocorrido na área de microprocessadores e microcontroladores, novas soluções tecnológicas foram propostas para a interconexão de dispositivos. Dentre estas se destaca a comunicação entre dispositivos através de redes sem fio (RSF), distribuídas geograficamente, que se comunicam através de um link de rádio. Estas redes diferem das redes de computadores tradicionais nos aspectos descritos a seguir. Tipicamente possuem um grande número de nós distribuídos, impõem restrições de consumo de energia, e devem possuir mecanismos para autoconfiguração e adaptação, de modo a poder superar falhas de comunicação e a perda de nós. Uma RSF tende a ser autônoma e é utilizada em situações nas quais se pretende alcançar um elevado grau de operação para as tarefas atribuídas à rede (Loureiro ET AL, 2003).

1.1 OBJETIVO

Visando aumentar o realismo das atividades de treinamento com simuladores, neste trabalho, se propõe projetar e construir uma interface entre os operadores e o ambiente de simulação. Esta interface deve permitir o envio e recepção de comandos entre os operadores em treinamento e o ambiente simulado, emulando a operação sobre os painéis de controle do ambiente de subestações de um sistema elétrico. Os comandos gerados, a partir da manipulação de um conjunto de chaves, serão transmitidos para o simulador SimuLIHM (Torres, Carvalho, Vieira ET AL, 2011) através de uma rede sem fio.

1.2 MOTIVAÇÃO

O Grupo de interfaces Homem-Máquina (GIHM), da Universidade Federal de Campina Grande (UFCG), está trabalhando no desenvolvimento de um simulador composto de um ambiente virtual tridimensional e de um sistema supervisor, os quais integrados constituem a plataforma de treinamento SimuLIHM (Torres, Carvalho, Vieira ET AL, 2011).

O ambiente foi desenvolvido para o treinamento de operadores das subestações do sistema elétrico, os quais podem estar localizados remotamente em relação ao ambiente de treinamento. A atuação do operador no ambiente virtual carece de realismo quando realizando manobras específicas sobre o sistema, o que resultou na realização deste trabalho.

1.3 ORGANIZAÇÃO DO RELATÓRIO.

Este texto inicia com uma revisão dos conceitos utilizados ao longo do trabalho, seguido da especificação e descrição da implementação da interface, finalizando com a apresentação dos testes realizados, considerações finais e sugestões para trabalhos futuros.

2 APRESENTAÇÃO DOS CONCEITOS UTILIZADOS

Neste capítulo são sintetizados os conceitos necessários à compreensão do texto deste relatório os quais foram fundamentais ao desenvolvimento do trabalho.

2.1 APRESENTAÇÃO DO SIMULIHM.

O ambiente para o treinamento de operadores em painéis e supervisorio apoiado por um módulo tutor feito pelo grupo do laboratório de interface homem-máquina (LIHM) [4] se insere no contexto de desenvolvimento de um simulador para o treinamento de operadores de sistemas elétricos, o qual oferece a opção de treinamento tanto em painéis quanto em supervisorio. Ele descreve a representação do supervisorio no ambiente do simulador (SimuLIHM), e do módulo tutor; ambos desenvolvidos para apoiar o planejamento, a execução e avaliação das sessões de treinamento. (Torres, Carvalho, Vieira ET AL, 2011).

O SimuLIHM consiste de um conjunto de ferramentas para apoiar a realização de treinamentos e a avaliação de seus resultados. Sua arquitetura é modular e distribuída e oferece dois ambientes: um ambiente voltado para os treinandos (operadores) e outro para o usuário tutor. Os ambientes podem ser acessados via internet ou intranet, permitindo a realização de treinamentos à distância e a participação simultânea de grupos de operadores, os quais podem interagir entre si e com o tutor. (Torres, Carvalho, Vieira ET AL, 2011).

Na versão atual, o SimuLIHM oferece aos usuários um ambiente de simulação, baseado em realidade virtual, que reproduz uma típica sala de controle de uma subestação distribuidora de energia elétrica (Figura 1). (Torres, Carvalho, Vieira ET AL, 2011).



FIGURA 1 – SALA DE CONTROLE VIRTUAL DE SUBESTAÇÃO DE UM SISTEMA ELÉTRICO

Nesse ambiente virtual tridimensional (3D), os operadores, em treinamento, podem deslocar-se, interagindo a partir de um mouse ou teclado; e realizar suas tarefas de forma semelhante ao que fariam no ambiente real. (Torres, Carvalho, Vieira ET AL, 2011).

Na Figura 2 podemos verificar o ambiente do tutor no SimuLIHM.

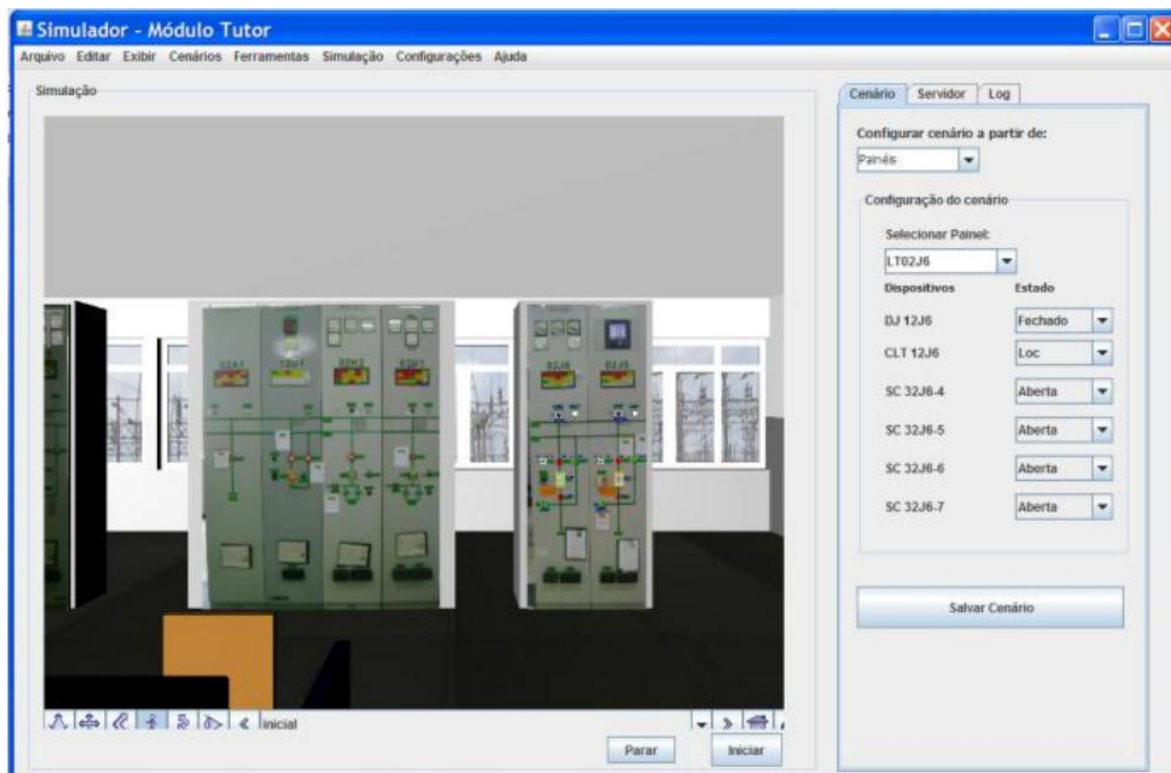


FIGURA 2 – AMBIENTE TUTOR NO SIMULIHM

2.2 REDES SEM FIO.

Uma rede sem fio (*Wireless*) é um sistema que interliga vários equipamentos fixos ou móveis utilizando o ar como meio de transmissão. É semelhante a uma rede local com fio convencional (que segue o padrão IEEE 802.3), com exceção de que a rede sem fio usa ondas de rádio, em vez de cabos. O que significa que é possível uma transmissão de dados via ondas de rádio, sem a necessidade de conexão física por meio de cabos entre computadores.

As redes *wireless* são soluções normalmente aplicadas onde uma infraestrutura de cabeamento convencional (cobre ou fibra óptica) não pode ou não é desejada ser utilizada.

Embora ainda persistam algumas dúvidas e discussões sobre a confiabilidade e eficiência das redes sem fio no que diz respeito à segurança na transmissão da informação, existe um consenso sobre sua fácil configuração, eficiente controle e gerenciamento de dispositivos e simplicidade para alterações do layout.

2.2.1 REDE ZIGBEE PADRÃO IEEE 802.15.4

Originalmente o ZigBee [1] foi pensado para uso em automação industrial em que sua grande maioria as informações são pequenas e básicas. Com isso a taxa de transmissão é baixa, comparadas com outras tecnologias de transmissão sem fio como os padrões IEEE 802.11b ou IEEE 802.11g (*Wi-Fi*), que operam na mesma frequência 2,4GHz

Os dispositivos baseados na tecnologia ZigBee operam na faixa ISM (*Industrial, Scientific and Medical*) que não requer licença para funcionamento, incluindo as faixas de 2,4GHz (Global), 915MHz (América) e 868MHz (Europa) e com taxas de transferência de dados de 250kbps em 2,4GHz, 40kbps em 915MHz e 20kbps em 868MHz.

O padrão ZigBee foi criado para economizar energia ao máximo. Com isso, é possível criar dispositivos sensores remotos alimentados com pilhas ou baterias comuns, que durarão meses ou mesmo anos sem precisarem ser substituídas. Isso porque, os módulos ZigBee quando não estão transmitindo/recebendo dados, entram num estado de dormência ou em "*Sleep*", consumindo o mínimo de energia.

O alcance médio descrito no *datasheet* do fabricante dos módulos XBee da Digi International Inc. usado no projeto em ambientes internos/urbano é próximo a 100m, em linha visível ambiente externo é de 1,6 KM.

Os módulos de ZigBee contém interface de comunicação serial assíncrona, pelo menos da porta serial, o módulo pode comunicar com qualquer dispositivo com a mesma tensão lógica, ou através de dispositivo que regule o nível lógico, como exemplo RS- 232 USB (*Universal Serial Bus*) ou placas de interface.

No Anexo 2 pode-se verificar com maiores detalhes em relação ao desempenho, alimentação e geral as especificações dos parâmetros dos módulos da Digi International Inc. utilizados para o desenvolvimento do projeto.

A escolha do módulo se deu pela facilidade de aquisição dos módulos da Digi, já que todos os módulos de ZigBee seguem o mesmo padrão da IEEE 802.15.4. Com referência aos módulos da Digi, existem dois modelos que se diferenciam principalmente pela potência XBee e o XBee-Pro esse com maior potência. Os dois modelos tem os mesmos pinos e mesmo tipos de antenas..Tipo chicote ou whip, um pedaço de fio de 2,5 cm Figura 3 tipo *chip* a mais compacta, Figura 4.



FIGURA 3 - ANTENA TIPO CHICOTE OU WHIP



FIGURA 4 - ANTENA TIPO CHIP

Topologias de redes ZigBee.

Nas redes ZigBee podemos encontrar dois tipos de dispositivos, FFD (*Full Function Device*) e RFD (*Reduced Function Device*) [5].

FFD – Dispositivos de funções completas. Os dispositivos são mais complexos e necessitam de um hardware mais potente, para conseguir implantar as pilhas de protocolos, consomem mais energia, nesse caso o uso deles é aconselhado quando se tem uma alimentação em rede elétrica contínua ou com uso de energia solar com baterias. Na rede ZigBee eles podem ser configurados para assumir o papel de Coordenador, Roteador ou de um dispositivo final, mas pelo consumo maior de energia se deve ponderar o seu uso. São implementados em microcontroladores com no mínimo 32kB de memória de programa, com certa quantidade de memória RAM, para que consiga implantar as tabelas de rotas e configurações de parâmetros nos casos de serem configurados como Coordenador e Roteador (Digi International Inc.).

RFD – Dispositivos de Funções Reduzidas. Os dispositivos nesse modo são bem mais simples, os recursos, por exemplo, podem ser implementados em microcontroladores de 8bits com memória de programa de 6kB. Nesse modo os dispositivos ficam restritos à comunicação com dispositivos FFD, nos modos de coordenador ou roteador. Assim sendo, só podem assumir o papel de dispositivos finais, por não serem capazes de fazer uma tabela de rotas (Digi International Inc.).

Como mencionado os dispositivos ZigBee podem assumir três modos de configurações na rede: ZigBee Coordenador (ZigBee Coordinator), ZigBee Roteador (ZigBee Router) e ZigBee Dispositivo Final (ZigBee End Device).

ZigBee Coordenador, o coordenador só pode ser implementado por um dispositivo FFD. Ele é o responsável pela inicialização da rede, distribuição de endereços, manutenção da rede e reconhecimento de todos os Nós da rede e somente ele pode servir de ponte para outras Redes ZigBee.

ZigBee Roteador, como o coordenador só pode ser implementado por dispositivos FFD. Contém características de um Nó normal na rede, mas alguns com diferenciais, podendo exercer a função de roteador intermediário entre nós, com a ausência de Coordenador. Com um Roteador na rede ZigBee ela pode ser expandida, os Roteadores podem ser usados como amplificador quando os dispositivos não conseguem se comunicar.

ZigBee Dispositivo Final são os atuadores ou sensores. Pode ser implementado nos dois dispositivos FFD ou RFD, sendo os que consomem menos energia, pois em sua configuração é possível ativar o modo *sleep*, assim, enquanto ele não transmite o uso da energia é mínimo.

Redes Mesh Sem Fios

A comunicação usando a topologia *Mesh* estabelece uma rede complexa e colaborativa. Os elementos da rede funcionam como repetidores, para alcançar rádios que não se comunicam a longas distancias. A rede pode crescer indefinidamente, cada ponto podendo fazer uma ligação com o próximo ponto da rede e assim sucessivamente. Se ocorrer de um ponto repetidor falhar outro ponto que possa fazer a ligação entre os pontos passa a fazer automaticamente, e reestabelece o circuito da rede *Mesh*. A função de reestabelecer a rede quando algum ponto para de funcionar chama-se de auto cura (*self-healing*).

A rede ainda pode se organizar de maneira a otimizar o trafego de dados, com outros caminhos possíveis. Na Figura 5 pode-se verificar uma rede Mesh, com o coordenador, roteadores e dispositivos finais, a figura está mostrando as possíveis ligações da rede com XBee.

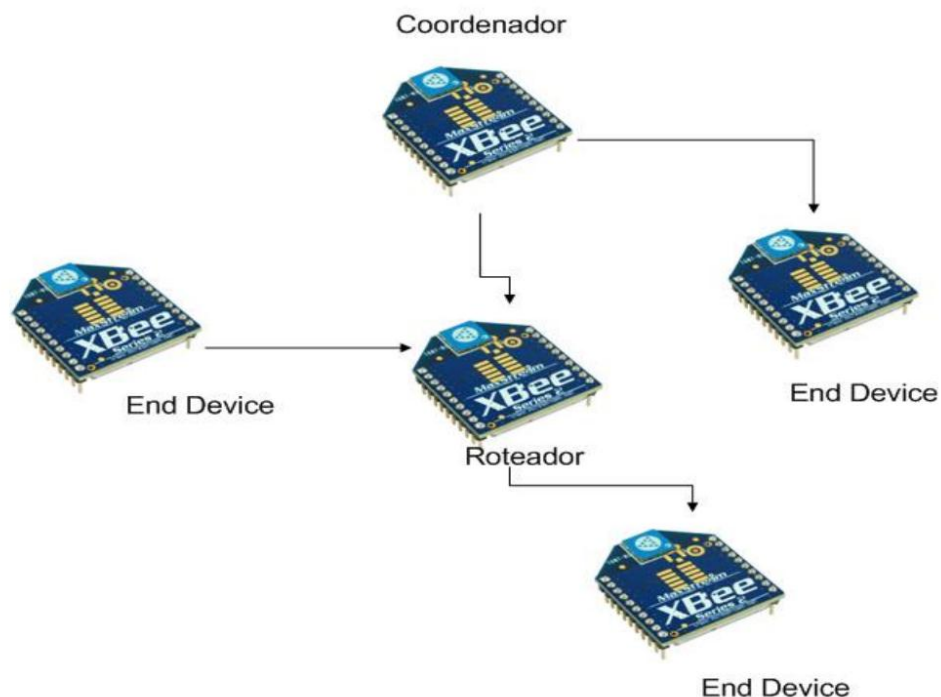


FIGURA 5 – REDE MESH ZIGBEE

2.3 O MICROCONTROLADOR PIC 16F877A

O PIC16F877 [3] é um microcontrolador fabricado pela *Microchip Technology*. Sua frequência de operação (clock) chega até 20 MHz. Pode trabalhar com uma fonte de alimentação de 2 volts a 5,5 volts (Souza, ET AL, 2003).

O microcontrolador possui quarenta pinos que possibilitam a montagem de um hardware complexo e capaz de interagir com diversos recursos e funções ao mesmo tempo. Possui uma via de programação com 14 bits e 35 instruções, 33 portas configuráveis com entradas e saídas, contém 15 interrupções disponíveis.

A memória de programação *E2PROM FLASH*, que permite a gravação rápida do programa diversas vezes no mesmo chip, sem a necessidade apagar por meio de luz ultravioleta, como acontece nos microcontroladores de janela.

A memória de programa com 8kbytes, com capacidade de escrita e leitura pelo próprio código interno. Uma memória E2PROM (não volátil) interna com 256 bytes, memória RAM com 368 bytes. três *timers* (2x8 bits e 1x16 bits).

A comunicação serial pode ser realizada em três modos SPI, PC e USART. Contém conversores analógicos de 10 bits (8x) e comparadores analógicos (2x).

As entradas e saídas desse microcontrolador são unidas em portas chamadas de PORTA, PORTB, PORTC, PORTD e PORTE.

Uma grande vantagem da família PIC é o de todos os modelos conterem um conjunto de instruções bem parecidas, e também apresentarem muita semelhança nas características básicas. No Anexo 2 podemos observar o diagrama de pinos do PIC16F877

2.3.1 PROGRAMANDO O PIC UTILIZANDO O AMBIENTE DE DESENVOLVIMENTO PIC C.

Programar um microcontrolador PIC pode ser uma tarefa bastante complexa se tentarmos programar em nível de hardware. Para resolver isso existem ambientes de desenvolvimento que induzem de forma intuitiva e de alto nível o programador facilitando muito o processo.

O ambiente de desenvolvimento PIC C é um software da *custom computer servisse* (CCS) que auxilia o programador evitando a digitação de várias linhas de código com uma interface em janelas bastante amigável. As figuras 6, 7 e 8 mostram a interface do PIC C.

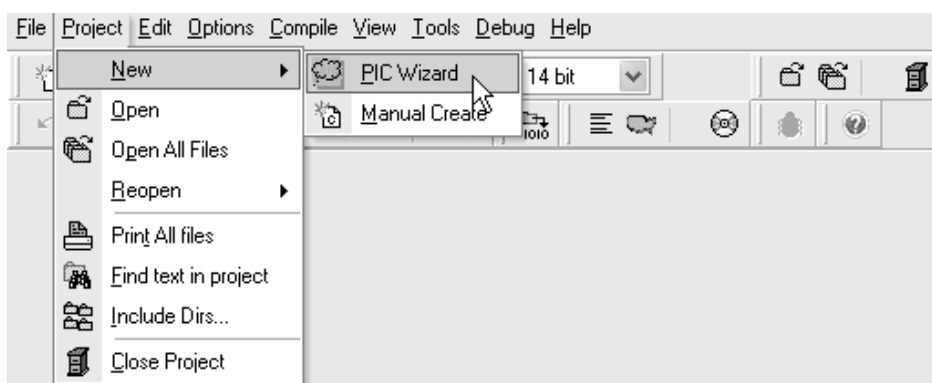


FIGURA 6 – CRIANDO UM NOVO PROJETO NO PIC C

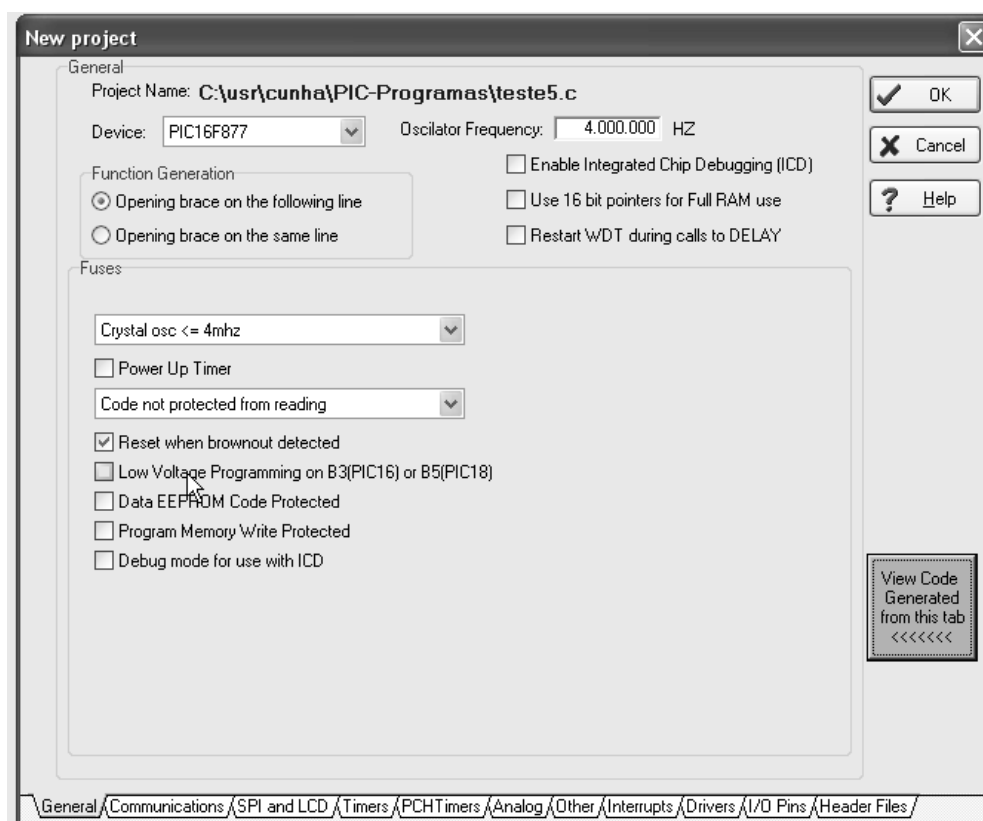
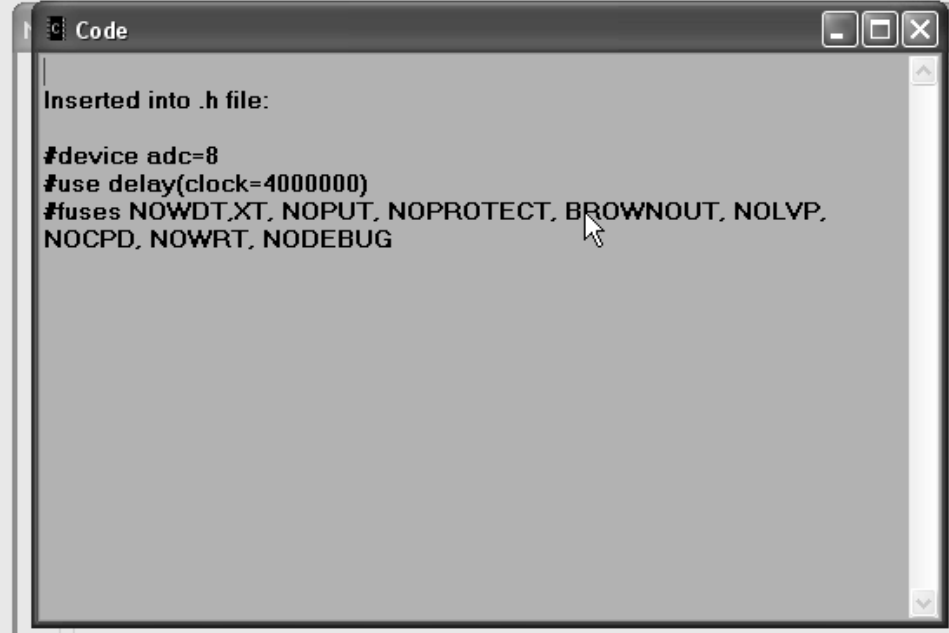


FIGURA 7 – CONFIGURAÇÕES GERAIS DO PIC



```
Code
Inserted into .h file:
#device adc=8
#use delay(clock=4000000)
#fuses NOWDT,XT, NOPUT, NOPROTECT, BROWNOUT, NOLVP,
NOCPD, NOWRT, NODEBUG
```

FIGURA 8 – LINHAS DE CÓDIGO GERADAS APARTIR DAS ESCOLHAS.

Para criarmos um projeto basta selecionar com o mouse como mostra a figura 7 em seguida aparecerá a janela para a configuração do PIC onde se pode configurar a frequência de *clock*, os *timers*, as entradas analógicas, as interrupções, entre outros como mostra a Figura 7. Na Figura 8 temos um código gerado após as configurações gerais serem escolhidas.

3 ESPECIFICAÇÃO E DESENVOLVIMENTO DA INTERFACE

Neste capítulo descreveremos as etapas da especificação e desenvolvimento do projeto. O desenvolvimento do projeto foi baseado inicialmente em pesquisa bibliográfica para a melhor escolha tanto do protocolo de rede a ser utilizado como dos componentes de hardware necessários. Em seguida foi feita a implementação do hardware e o estudo experimental através de testes de funcionalidade. As etapas do projeto são listadas abaixo.

- Estabelecer critérios para propor os requisitos que irão apoiar a escolha do tipo de comunicação sem fio a ser utilizada.
- Especificar o *hardware* a ser construído.
- Implementar o *hardware*.
- Realizar testes de funcionalidade
- Integrar com o ambiente do simulador e realizar testes de integração

3.1 PROTOCOLO DE REDE E MICROCONTROLADOR.

3.1.1 ESCOLHA DO MICROCONTROLADOR.

Originalmente um microcontrolador era escolhido por seu desempenho de processamento, família tecnológica e capacidade de memória. Como a unidade de processamento e a memória ocupavam quase toda a área de silício, sobrava muito pouco espaço para os periféricos, que eram poucos e limitados em recursos.

Atualmente, com a evolução dos processos de produção e tecnologia dos microcontroladores, a área ocupada pela unidade de processamento e memória foi muito reduzida, o que abriu, literalmente, espaço para a incorporação de diversos periféricos junto ao microcontrolador.

Os microcontroladores disponíveis no mercado, de todos os fabricantes, oferecem uma ampla gama de opções de periféricos, incluindo ADCs (conversores analógico-digitais), DACs (conversores digital-analógicos), PWMs (moduladores por largura de pulso), além de interfaces I2C, SPI, USB, Ethernet, e outras.

O primeiro passo é determinar qual é a menor arquitetura capaz de atender os requisitos do sistema, uma vez que os microcontroladores de 8 bits apresentam menor custo que os de 16 ou 32 bits. Nesse ponto vale verificar se é possível a utilização de um microcontrolador de 8 bits considerando algum sacrifício no tamanho do código e na velocidade de execução.

A escolha da linguagem de programação, linguagem de alto nível ou assembler, poderá causar impacto no desempenho do sistema e, dessa forma, também influenciar na seleção da arquitetura. Para aplicações em escala, o custo do sistema pode ser o critério mais importante.

A familiaridade do projetista com uma determinada marca ou família de microcontroladores pode ser uma vantagem, assim como uma desvantagem. Será uma vantagem se o microcontrolador mais adequado fizer parte desse conhecimento, o que irá reduzir os tempos de desenvolvimento, e uma desvantagem se o componente escolhido esbarrar com a natural resistência que os projetistas apresentam quando se trata de alterar o modo como executam seu trabalho.

Os critérios de escolha para este projeto foram: a capacidade de processamento e armazenamento, a quantidade de portas I/O, programação em linguagem de alto nível e principalmente a familiaridade com o microcontrolador por isso o microcontrolador escolhido foi o PIC 16F877a da Microchip.

3.1.2 ESCOLHA DO PROTOCOLO DE REDE SEM FIO.

Para a escolha do protocolo de rede a ser utilizado foram levados em conta principalmente a economia de energia, o alcance e a facilidade de acesso e implementação.

A Tabela 2 mostra um estudo comparativo entre os principais protocolos de redes sem fios utilizados no mercado [2].

TABELA 1 - COMPARAÇÃO DOS PRICIPAIS PROTOCOLOS DE REDES SEM FIO

Standard	Bluetooth	UWB	ZigBee	Wi-Fi
IEEE spec.	802.15.1	802.15.3a *	802.15.4	802.11a/b/g
Frequency band	2.4 GHz	3.1-10.6 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 GHz
Max signal rate	1 Mb/s	110 Mb/s	250 Kb/s	54 Mb/s
Nominal range	10 m	10 m	10 - 100 m	100 m
Nominal TX power	0 - 10 dBm	-41.3 dBm/MHz	(-25) - 0 dBm	15 - 20 dBm
Number of RF channels	79	(1-15)	1/10; 16	14 (2.4 GHz)
Channel bandwidth	1 MHz	500 MHz - 7.5 GHz	0.3/0.6 MHz; 2 MHz	22 MHz
Modulation type	GFSK	BPSK, QPSK	BPSK (+ ASK), O-QPSK	BPSK, QPSK COFDM, CCK, M-QAM
Spreading	FHSS	DS-UWB, MB-OFDM	DSSS	DSSS, CCK, OFDM
Coexistence mechanism	Adaptive freq. hopping	Adaptive freq. hopping	Dynamic freq. selection	Dynamic freq. selection, transmit power control (802.11h)
Basic cell	Piconet	Piconet	Star	BSS
Extension of the basic cell	Scatternet	Peer-to-peer	Cluster tree, Mesh	ESS
Max number of cell nodes	8	8	> 65000	2007
Encryption	E0 stream cipher	AES block cipher (CTR, counter mode)	AES block cipher (CTR, counter mode)	RC4 stream cipher (WEP), AES block cipher
Authentication	Shared secret	CBC-MAC (CCM)	CBC-MAC (ext. of CCM)	WPA2 (802.11i)
Data protection	16-bit CRC	32-bit CRC	16-bit CRC	32-bit CRC

* Unapproved draft.

• Acronyms: ASK (amplitude shift keying), GFSK (Gaussian frequency SK), BPSK/QPSK (binary/quadrature phase SK), O-QPSK (offset-QPSK), OFDM (orthogonal frequency division multiplexing), COFDM (coded OFDM), MB-OFDM (multiband OFDM), M-QAM (M-ary quadrature amplitude modulation), CCK (complementary code keying), FHSS/DSSS (frequency hopping/direct sequence spread spectrum), BSS/ESS (basic/extended service set), AES (advanced encryption standard), WEP (wired equivalent privacy), WPA (Wi-Fi protected access), CBC-MAC (cipher block chaining message authentication code), CCM (CTR with CBC-MAC), CRC (cyclic redundancy check).

Fonte: The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON).

O protocolo de rede escolhido foi o ZigBee pelo baixo consumo de energia, um alcance suficiente para a aplicação e por ter uma melhor acessibilidade para o projetista em questão.

3.2 ESPECIFICAÇÃO DO PROJETO.

Definidos, o microcontrolador e o protocolo de rede sem fio a serem utilizados podemos especificar os componentes necessários á montagem do protótipo.

3.2.1 DESCRIÇÃO DA ARQUITETURA.

O hardware no projeto é responsável pela comunicação *Full-Duplex* entre painéis de subestação simulados e o ambiente virtual SimuLIHM. A comunicação sem fio é implementada por módulos XBee onde um dos módulos tem função de coordenador da rede e os demais são *end points*. Neste projeto usaremos apenas um painel simulado deixando a rede em aberto para que possam ser adicionados posteriormente outros painéis simulados. Um dos módulos XBee é conectada à porta serial de um microcontrolador onde será feito o tratamento dos dados que saem e chegam ao painel simulado o outro módulo (coordenador) é conectada através de um conversor USB/Serial a um computador que hospeda o simulador SimuLIHM. O objetivo é que as modificações realizadas no painel sejam verificadas no simulador em realidade virtual em tempo real e vice-versa.

Como pode ser verificado no diagrama de blocos na Figura 9

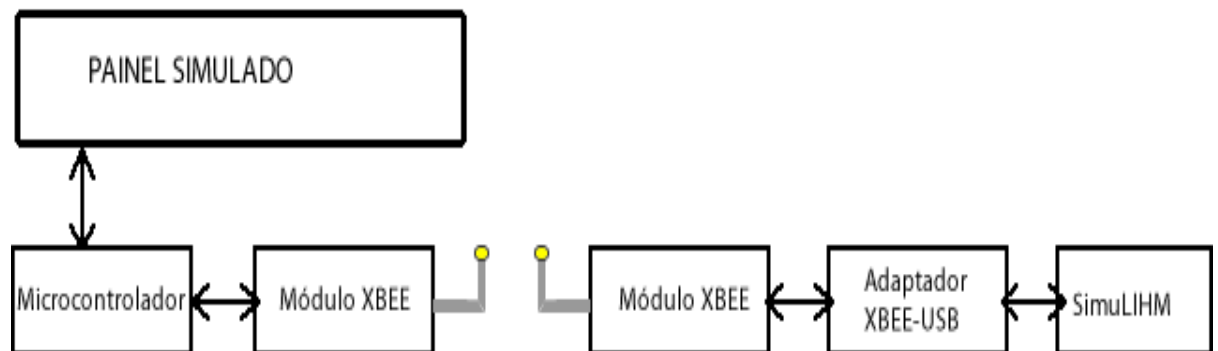


FIGURA 9 – DIAGRAMA DE BLOCOS DA ARQUITETURA DO PROJETO.

3.2.2 ESPECIFICAÇÃO DO PAINEL A SER SIMULADO.

O painel simulado consiste em uma réplica de um dos painéis que já se encontram no ambiente virtual SimuLIHM. O painel escolhido segue na Figura 10.



FIGURA 10 - PAINEL QUE SERÁ REPLICADO NO PROJETO (02J5).

O painel LT02J5 é utilizado para supervisão e manobras envolvendo a linha de transmissão LT02J5 que é uma linha de transmissão de 69KV utilizada para o fornecimento de energia elétrica para uma grande empresa.

Podemos destacar neste painel a chaves tipo Giro-Pressão-Giro (GPG) que comandam os disjuntores. O funcionamento destas chaves é descrito da seguinte forma: para comandar a abertura de equipamentos deve-se girar o embolo móvel do dispositivo, seguido de uma pressão e seguido de mais um giro no mesmo sentido; para executar o fechamento do dispositivo de campo através da chave GPG os movimentos são semelhantes só que em sentido contrário ao explicado anteriormente.

Ainda as chaves GPG tem LEDs sinalizadores que funcionam da seguinte forma: quando a chave fecha um disjuntor e o disjuntor em campo é realmente fechado um LED vermelho é acendido dentro da chave. Quando a chave abre um disjuntor e o mesmo é aberto em campo acende um LED branco dentro da chave e quando a chave abre ou fecha mas, o disjuntor no campo não concorda há uma discordância entre chave e disjuntor então um LED branco pisca dentro da chave indicando que há discordância.

As chaves GPG serão representadas no painel simulado por chaves, LEDs e botões.

3.2.3 ESPECIFICAÇÃO DO SOFTWARE EMBARCADO.

O software embarcado foi desenvolvido em linguagem C [6], utilizando a plataforma do MPLab distribuído pela Microchip Technology Inc e compilado no *PCW Compiler* fornecido pela *Custom Computer Services, Inc.*(*Custom, 2008*), o *software* embarcado é responsável pela identificação das informações e repasse ao Xbee, de modo serial.

Quando o treinando aciona as chaves o *software* trata os dados e os envia para a porta serial o diagrama de caso de uso é mostrado na Figura 11.

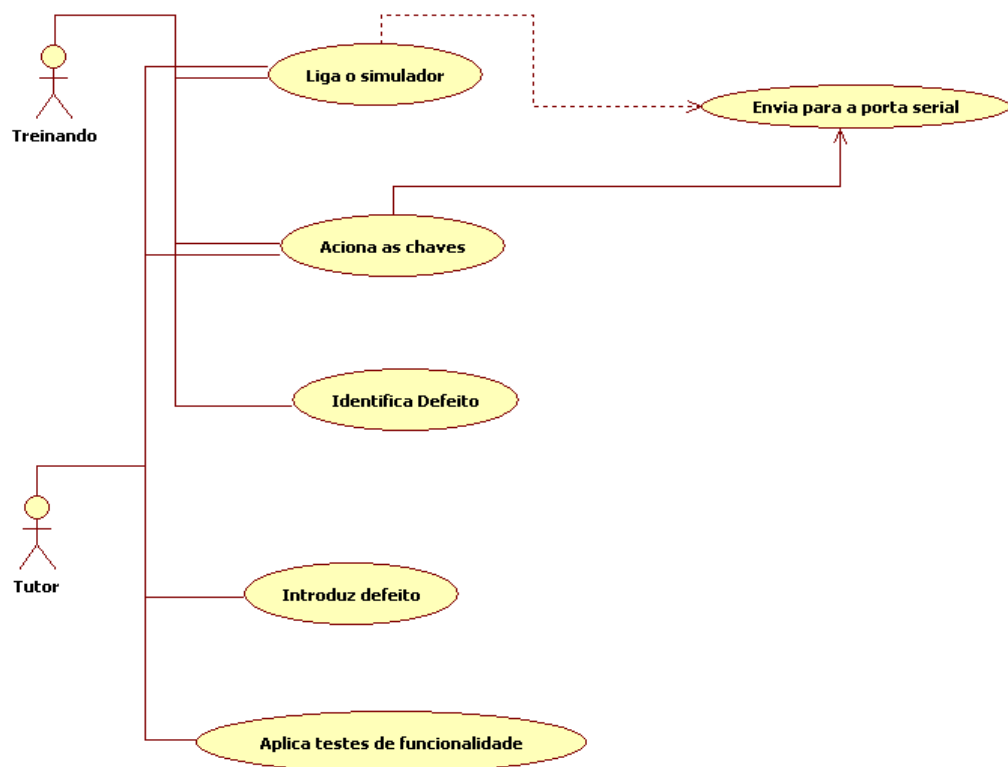


FIGURA 11 – DIAGRAMA DE CASO DE USO

Descrição dos casos de uso do *Software*:

Caso de uso: Liga o simulador

Ator: Tutor, Treinando.

Descrição: O tutor e o treinando podem ligar o simulador através da energização do mesmo.

Caso de uso: Envia para a porta serial

Ator: Tutor, Treinando.

Descrição: Quando se executa qualquer ação nas chaves esta ação é transmitida para a porta serial.

Caso de uso: Aciona as chaves

Ator: Tutor, Treinando.

Descrição: Quando se quer atuar na planta faz-se isso através das chaves.

Caso de uso: Identifica defeito.

Ator: Treinando.

Descrição: Quando um defeito proposital é introduzido no simulador o treinando deve identifica-lo.

Caso de uso: Introduce defeito

Ator: Tutor.

Descrição: O tutor aciona uma chave que simula um defeito no simulador.

Caso de uso: Aplica testes de funcionalidade.

Ator: Tutor

Descrição: O tutor testa o simulador antes do uso.

3.2.4 ESPECIFICAÇÃO DO MICROCONTROLADOR.

O software embarcado é armazenado na memória do PIC, onde é feita conversão de informações recebidas pelas portas de entrada do microcontrolador. Além da comunicação serial com XBee. O microcontrolador PIC 16F877a da Microchip foi escolhido para uso no projeto por ter comunicação serial e portas de entrada, suficientes para o desenvolvimento desta aplicação e pela melhor familiarização do projetista com o mesmo.

Para gravação e testes do software embarcado foram utilizadas placas de desenvolvimento para microcontroladores da família PIC.

3.2.5 ESPECIFICAÇÃO DOS MÓDULOS XBEE.

O XBee foi utilizado no projeto para fazer a rede sem fio de comunicação do painel para o supervisor e vice-versa. Como o XBee contém várias configurações, no projeto foram utilizadas duas configurações, um XBee-PRO é configurado como ZIGBEE COORDINATOR API e o módulo XBee – PRO acoplado no painel simulado funcionando como ZIGBEE END POINT.

3.3 IMPLEMENTAÇÃO DO PROJETO.

3.3.1 O PROJETO DE HARDWARE.

A implementação do hardware deu-se em duas partes. Uma através de um simulador de circuitos eletrônicos e outra através do hardware propriamente dito.

3.3.2 SIMULAÇÃO COM O PROTEUS PCB DESIGN.

A Figura 12 mostra dois PIC's 16F877a simulados no Proteus [8] sendo que um representa o hardware a ser projetado (esquerda) e o outra a planta no SimuLIHM (direita).

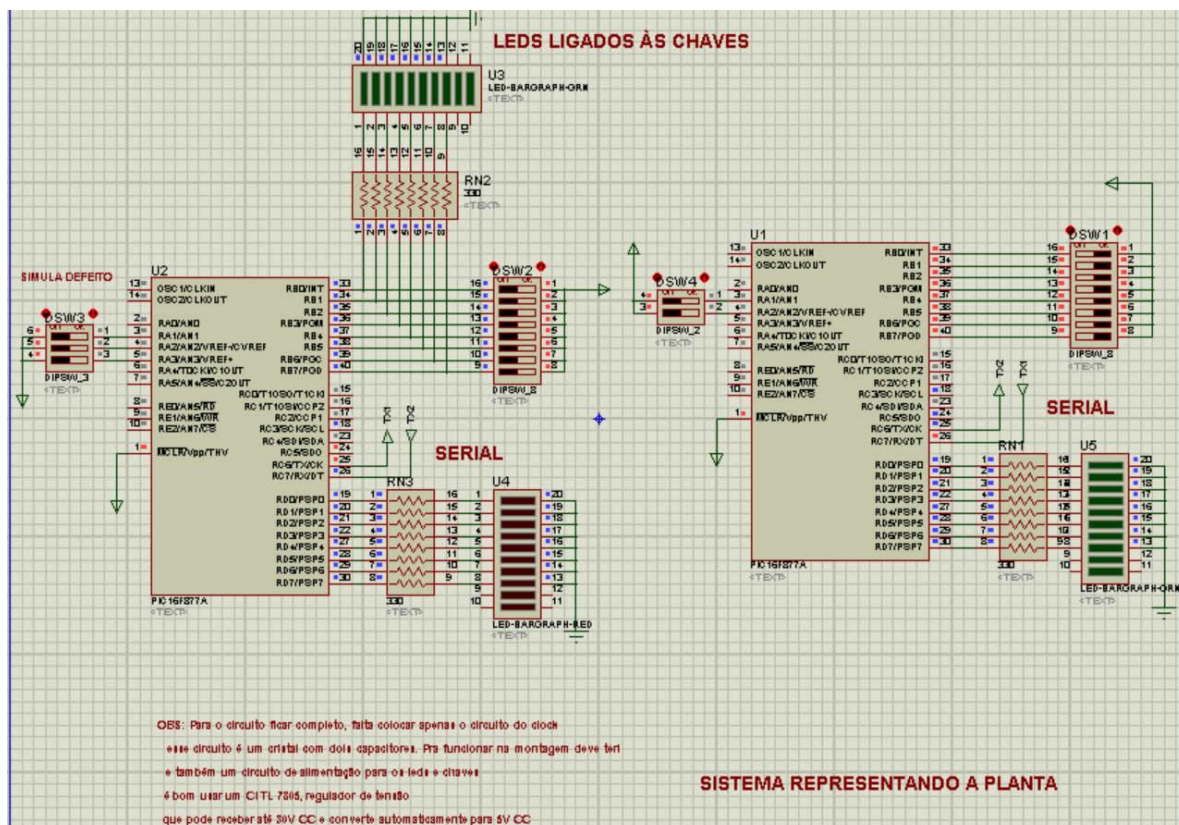


FIGURA 12 – CIRCUITO SIMULADO NO AMBIENTE PROTEUS

Em um primeiro momento foram montados dois circuitos. Um para simular o protótipo e outro para simular a planta em que o protótipo atua.

O circuito na Figura 13 representa as chaves GPG do painel real onde, as chaves representam os botões e os LEDs representam o estado dos mesmos. Quando uma chave é acionada no PIC 1 (esquerda) um LED no PIC 2 (direita) é *setado* indicando que houve atuação remota. Quando a chave no PIC 1 e o LED no PIC 2 estão em discordância um LED de alerta pisca no PIC 1 indicando a discordância.

Foram adicionadas também três chaves de defeito cuja função é de simular defeitos nas chaves para testes e treinamento dos futuros operadores.

3.3.3 OS MÓDULOS XBEE.

O XBee foi utilizado no projeto para fazer a rede sem fio de comunicação dos controles com o computador.

Como o XBee contém várias configurações, no projeto foram utilizadas duas configurações, um XBee-PRO é configurado como ZIGBEE COORDINATOR API e o módulo XBee no protótipo foi configurado como ZIGBEE END DEVICE no aplicativo X-CTU da Digi, que contém os firmwares dos módulos XBee e XBee-PRO, um exemplo da tela de configuração pode ser vista na Figura 13. Nessas configurações os firmwares são definidos com parâmetros da estrutura de rede que ele vai utilizar se Coordenador, Roteador ou Dispositivo final.

A utilização dos módulos XBee no projeto propiciou facilidade na montagem da rede, o protocolo ZigBee se encarrega de fazer toda a estrutura da rede, após a configuração dos módulos com sua respectiva função na rede. Um detalhe a ser levado em conta na utilização do XBee é a série do módulo, na série 1 (um) ele faz a comunicação sem o empacotamento de dados, na série 2 (dois) ele utiliza o empacotamento de dados. A série utilizada no projeto foi a série 2 onde, a transmissão da informação do XBee é feita através de pacotes, pacotes esses que contém, além da informação, garantia de entrega, endereço de origem e endereço de destino. Assim que o XBee recebe os dados seja pelas entradas ou pela serial ele monta o pacote com a informação a transmitir e o destino que precisa ser alcançado.

Para fazer a conexão do XBee com microcontrolador são usados os pinos 2 para RX e o pino 3 para TX, fazendo a transmissão serial de informações. A tensão usada na alimentação do módulo XBee é 3,3V DC, conectado no pino 1 e GND no pino 10, para ver com maior detalhes as conexões usadas ver a Figura 14.

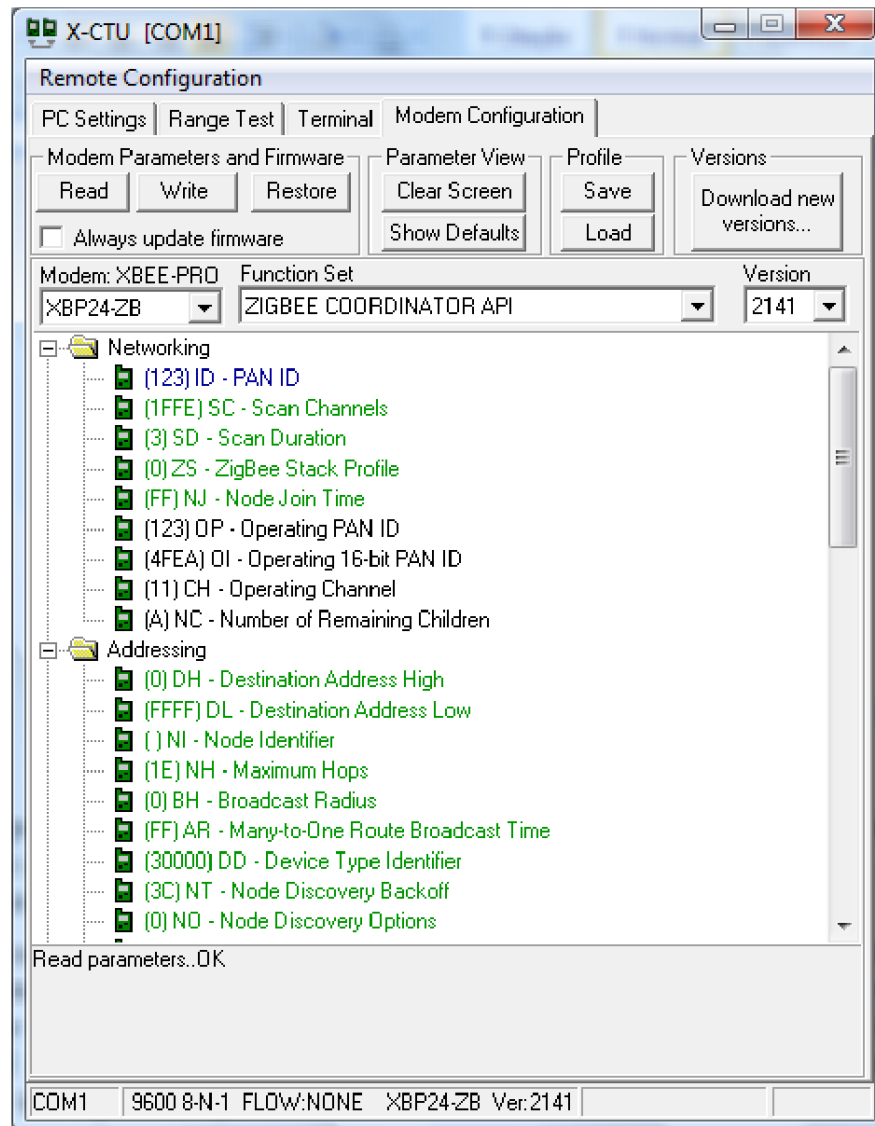


FIGURA 13 - INTERFACE DO APLICATIVO X-TCU

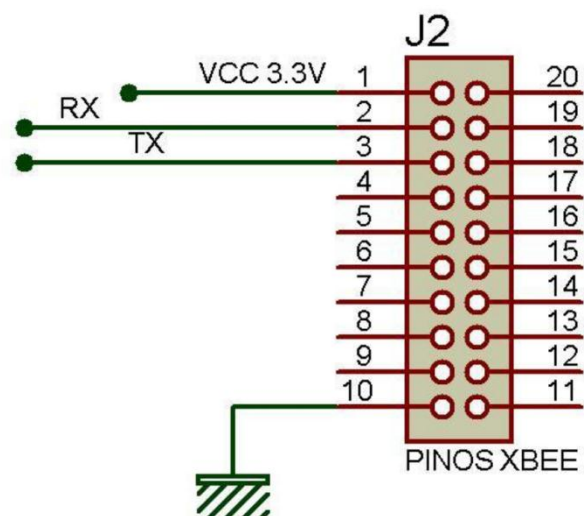


FIGURA 14 – PINOS DO XBEE

3.3.4 MONTAGEM DO PROTÓTIPO.

Após simulações com PROTEUS pudemos então montar o circuito da Figura 12 com componentes reais como podemos ver nas Figura x e x.

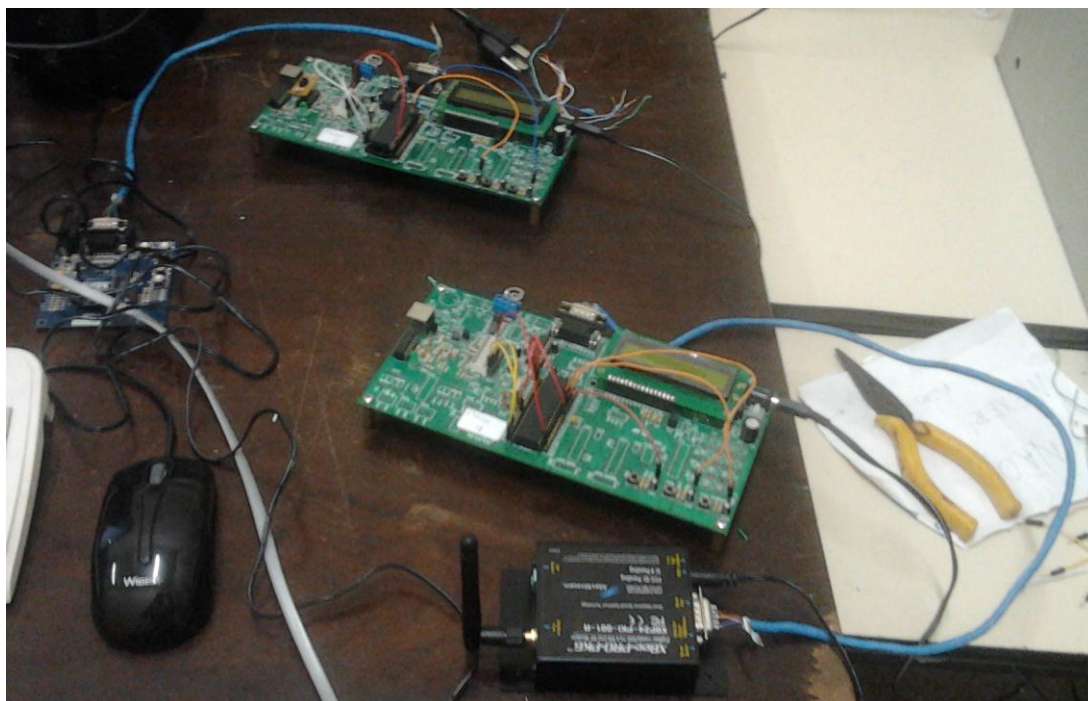


FIGURA 15 – MONTAGEM COM COMPONENTES REAIS

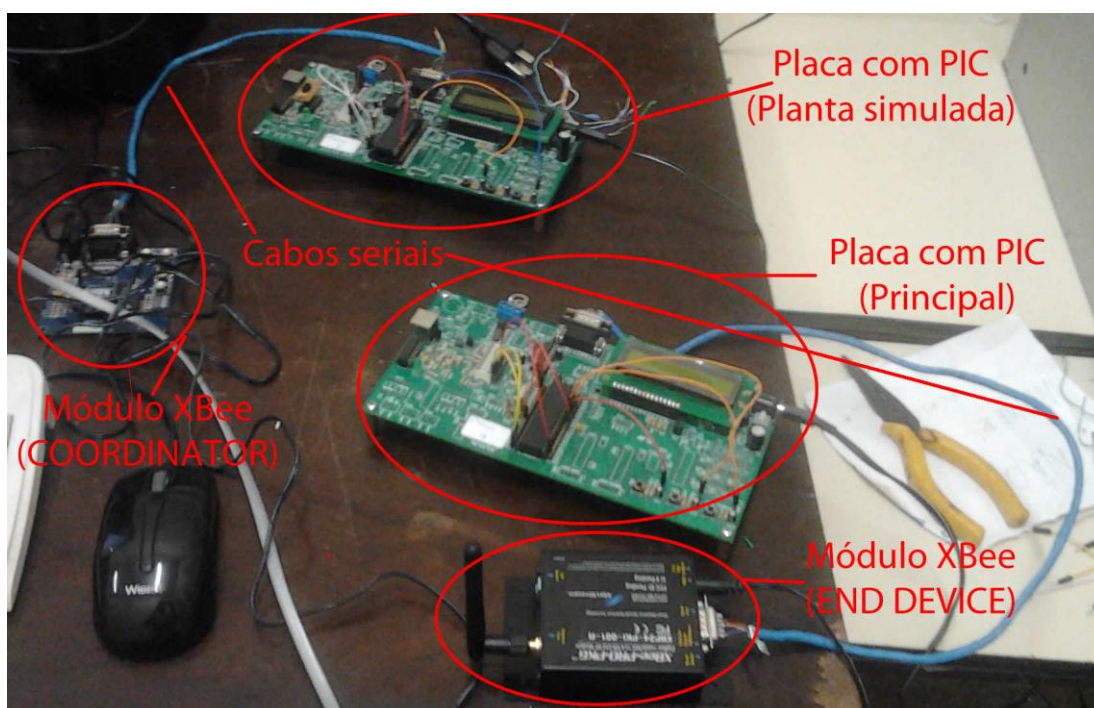


FIGURA 16 - MONTAGEM COM COMPONENTES REAIS

Após a programação do microcontrolador e o ajuste das configurações de rede temos no diagrama da Figura 15 como fica o protótipo. Foram feitas placas de circuito

impresso [7] para os LED's e chaves (Figura 16) então se pôde montar o protótipo que é mostrado na Figura 17.

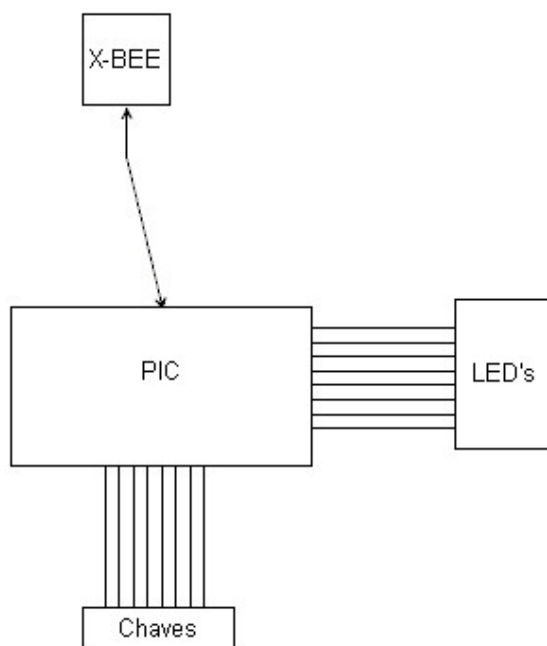


FIGURA 17 - DIAGRAMA DE BLOCOS DO PROTÓTIPO

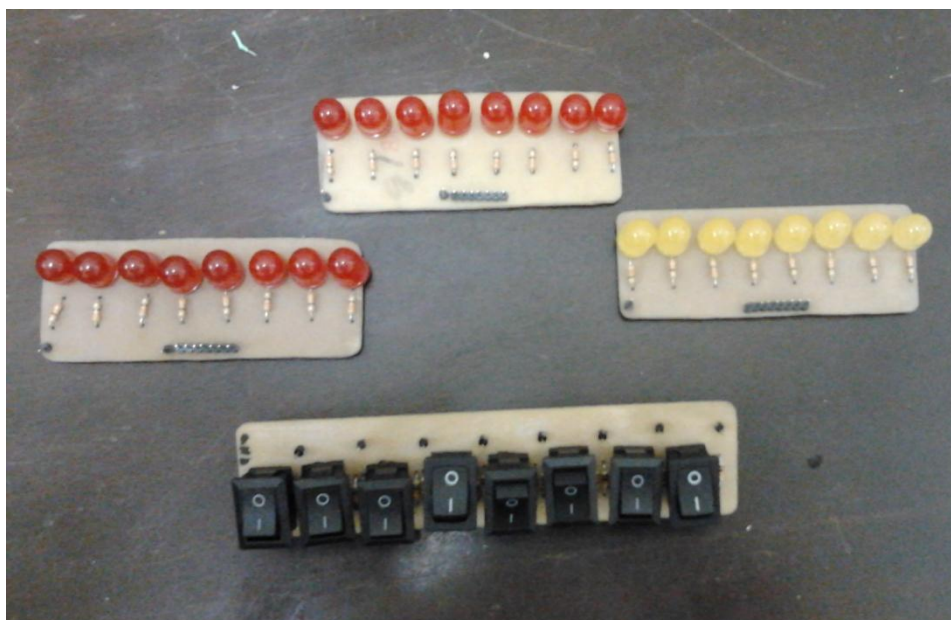


FIGURA 18 – PLACAS PARA A MONTAGEM DO PROTÓTIPO

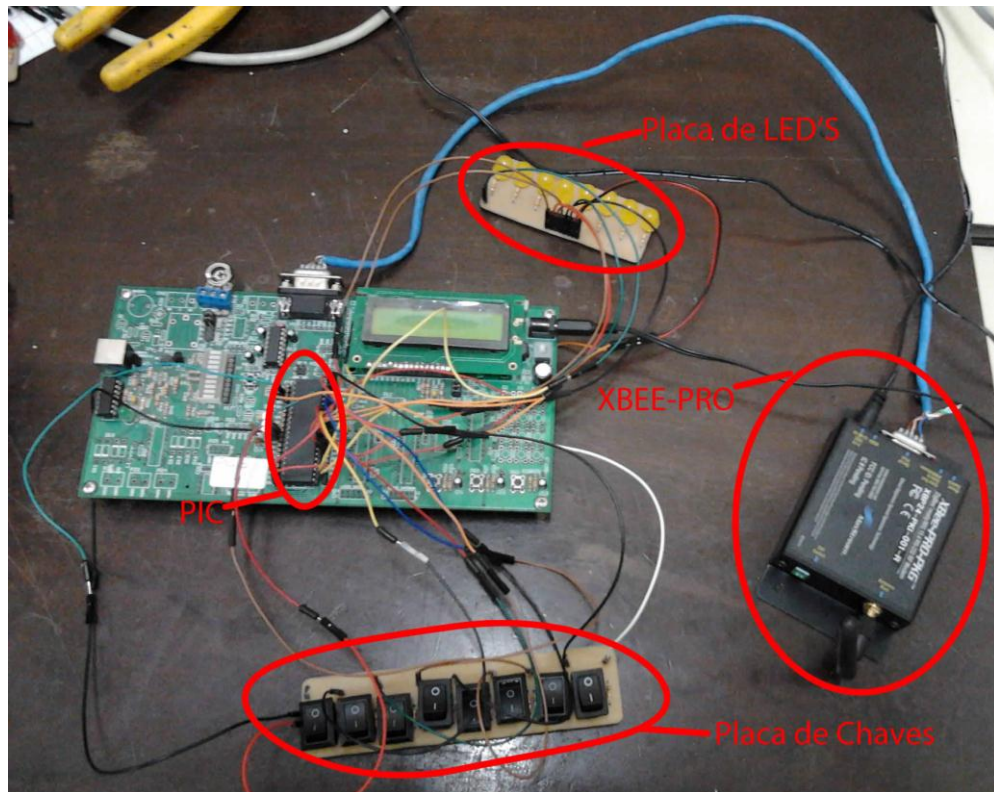


FIGURA 19 – MONTAGEM DO PROTÓTIPO

Os LED's amarelos indicam discordância das chaves e da planta.

4 TESTES E RESULTADOS.

4.1 SOFTWARE EMBARCADO.

Foram realizados testes da programação no PIC tanto dentro do ambiente do PROTEUS como no protótipo.

Primeiramente, foi testado no PROTEUS onde se inseria uma palavra de 8 bits através das chaves com o objetivo de que o valor das chaves fossem mostrados nos LED's da planta simulada. Em seguida foram feitas alterações na palavra de 8 bits e observou-se o comportamento da saída na planta através dos LED's. Os resultados foram satisfatórios, a saída da planta simulada mostrou a palavra de 8 bits sem erros e quando a entrada era alterada havia alteração nos LED's da planta em tempo real. Esses testes permitiram verificar que a lógica de programação do *software* embarcado estava de acordo com o que se esperava.

Após os testes no PROTEUS, gravamos os programas em PIC's reais e preparamos o *hardware* e aplicamos os mesmos testes descritos anteriormente. Os resultados foram parecidos apenas pela diferença de um pequeno atraso (*delay*) na chegada do dado na planta simulada. Isso acontece por perdas Ôhmicas no circuito real não compromete a correta lógica de programação do *software*.

4.2 HARDWARE.

As placas feitas para os LED's e chaves foram testadas com um voltímetro através de testes de continuidade antes de poderem ser acopladas ao protótipo. Os LED's antes de soldados às placas foram testados com uma bateria de 5 V e um resistor de 330 ohms em série. Nenhum LED apresentou problema assim como nenhuma placa.

Os dois módulos XBee foram testados separadamente antes da integração com os microcontroladores da seguinte maneira: Cada módulo foi conectado à porta serial de dois computadores (PC) e, com o X-TCU, enviavam-se *strings* através dos módulos e foi estabelecida uma conexão. Os resultados foram satisfatórios à medida que não houve perda de dados.

A comunicação entre o microcontrolador e o é feita através de cabo serial macho - macho. Entre eles não ocorreu problemas para conseguir comunicar já que nos dois componentes tem a transmissão serial integrada. Após ajustar os valores para a transmissão, foram feitos testes onde se enviava através da saída serial de um PIC para o módulo palavras de 8 bits e o outro módulo recebia a informação e passava via cabo serial para outro PIC que mostrava através de LED's a palavra depois os módulos foram invertidos então o que recebia passou a enviar. Nestes testes detectou-se um atraso perceptível, apesar de pequeno, de aproximadamente 0,1 s entre o acionamento da chave e a atuação na planta. Este atraso não interfere no projeto, pois a aplicação não exige um atraso menor.

A garantia de entrega dos dados entre os módulos se dá pelo protocolo IEEE 802.15.4, que tem garantia de entrega. A distância conseguida entre os módulos XBee foi de aproximadamente 20 m em área com paredes. Nessas distâncias não houve perda de dados.

Após a montagem do protótipo foram aplicados testes onde se acionavam as chaves e verificavam-se a resposta na planta simulada, também foi testada a saída no protótipo que é responsável por indicar se há discordância entre a planta e o protótipo (Vide Figura 13). Para isto é inserida uma chave na planta onde que determina entre atuar diretamente nela através de chaves e a atuação remota através do protótipo.

O teste procedeu da seguinte maneira:

- Entrou-se com um valor nas chaves do protótipo que pôde ser verificado nos LED's da planta.
- Em seguida foi acionada a chave para atuação direta na planta
- Foram modificadas as saídas da planta através de chaves. De modo que o valor das saídas da planta fosse diferente do valor de entrada do protótipo;
- Foram verificados os LED's na saída do protótipo que foram acionados piscando, indicando que houve discordância entre o protótipo e a planta.

Os resultados foram os esperados e não ocorreu nenhum tipo de contratempo.

5 CONSIDERAÇÕES E SUGESTÕES PARA TRABALHOS FUTUROS.

Tornar a interação com o ambiente simulado SimuLIHM mais realista, permitindo a imersão dos operadores de subestações do sistema elétrico em treinamento no seu ambiente típico de trabalho foi o que deu sentido a este trabalho, que tem a função de complementar o projeto SimuLIHM.

Os objetivos almejados pelo projeto foram parcialmente alcançados à medida que não foi feita a integração do protótipo com o SimuLIHM por prazos limitados, ficando esta última parte sugerida para futuros trabalhos.

Apesar de não ter acontecido o acoplamento ao SimuLIHM os resultados do trabalho se mostraram satisfatórios, o *hardware* se mostrou funcional em função de mostrar que se pode dar continuidade a partir do encaminhamento aqui mostrado.

Em futuros trabalhos pode-se acoplar o Hardware ao SimuLIHM, simular outras chaves e recursos dos painéis e expandir para todos os painéis necessários.

BIBLIOGRAFIA

- [1] Eady, F. (2007). *Hands-on ZigBee - implementing 802.15.4 with microcontrollers*. Elsevier.
- [2] Lee, J.-S., Su, Y.-w., & Sehn, C. C. (5 a 8 de Novembro de 2007). A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. *The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, p. 46 a 51.
- [3] Microchip Technology Inc. (1999). PIC16F87X., (p. 200).
- [4] Torres, F. T., da Costa, R. C., & Q. Vieira, M. d. (18 a 21 de Setembro de 2011). AMBIENTE PARA O TREINAMENTO DE OPERADORES EM PAINÉIS E SUPERVISÓRIO APOIADO POR UM MÓDULO . *X SBAI – Simpósio Brasileiro de Automação Inteligente*, p. 857 a 862.
- [5] LOUREIRO, A.A.F., NOGUEIRA, J.M.S., RUIZ, L.B., MINI, R.A.F., NAKAMURA, E.F. FIGUEIREDO, C.M.S. Redes de sensores sem fio. *Simpósio Brasileiro de Redes de Computadores*. Natal -RN. 2003
- [6] PEREIRA, F. PIC: Programação em C. São Paulo: Érica, 2003
- [7] SEDRA, Adel S.; SMITH Kenneth C. *Microeletrônica*. São Paulo Person Education do Brasil, 2007
- [8] Labcenter.(s.d). Proteus PCB design software. Acesso em 11 de novembro de 2012. Disponível em <http://www.labcenter.com/>

ANEXO 1 - ESPECIFICAÇÃO DE PARÂMETROS DO XBEE E XBEE-PRO

TABELA 2 - ESPECIFICAÇÃO DE PARÂMETROS DO XBEE E XBEE-PRO

Especificação	XBee ZNet 2.5	XBee PRO ZNet 2.5
Desempenho		
Alcance em áreas externas	40m	100m
Alcance em campo aberto	120m	1600m
Potencia de transmissão de saída	2mW (+3dBm), boost mode enabled 1,25mW (+1dBm), boost mode disabled	63mW (+18 dBm) 10mW (+10 dBm) for International variant
Taxa de transmissão RF	250,000 bps	250,000 bps
Taxa de transmissão serial	1200 - 230400 bps	1200 - 230400 bps
Alimentação		
Vdd	2,1 – 3,6 V	3,0 – 3,4 V
Corrente (máxima potência)	40mA (@ 3,3 V, boost mode enabled) 35mA (@ 3,3 V, boost mode disabled)	295 mA (3,3 V)
Corrente operacional	40mA (@ 3,3 V, boost mode enabled) 38mA (@ 3,3 V, boost mode disabled)	45mA (3,3V)
Geral.		
Frequência da banda	ISM 2,4 GHz	ISM 2,4 GHz

Fonte: Datasheet XBee®/XBee-PRO® ZB OEM RF Modules, Digi International Inc.

ANEXO 2 - DIAGRAMA DE PINOS DO PIC17F877

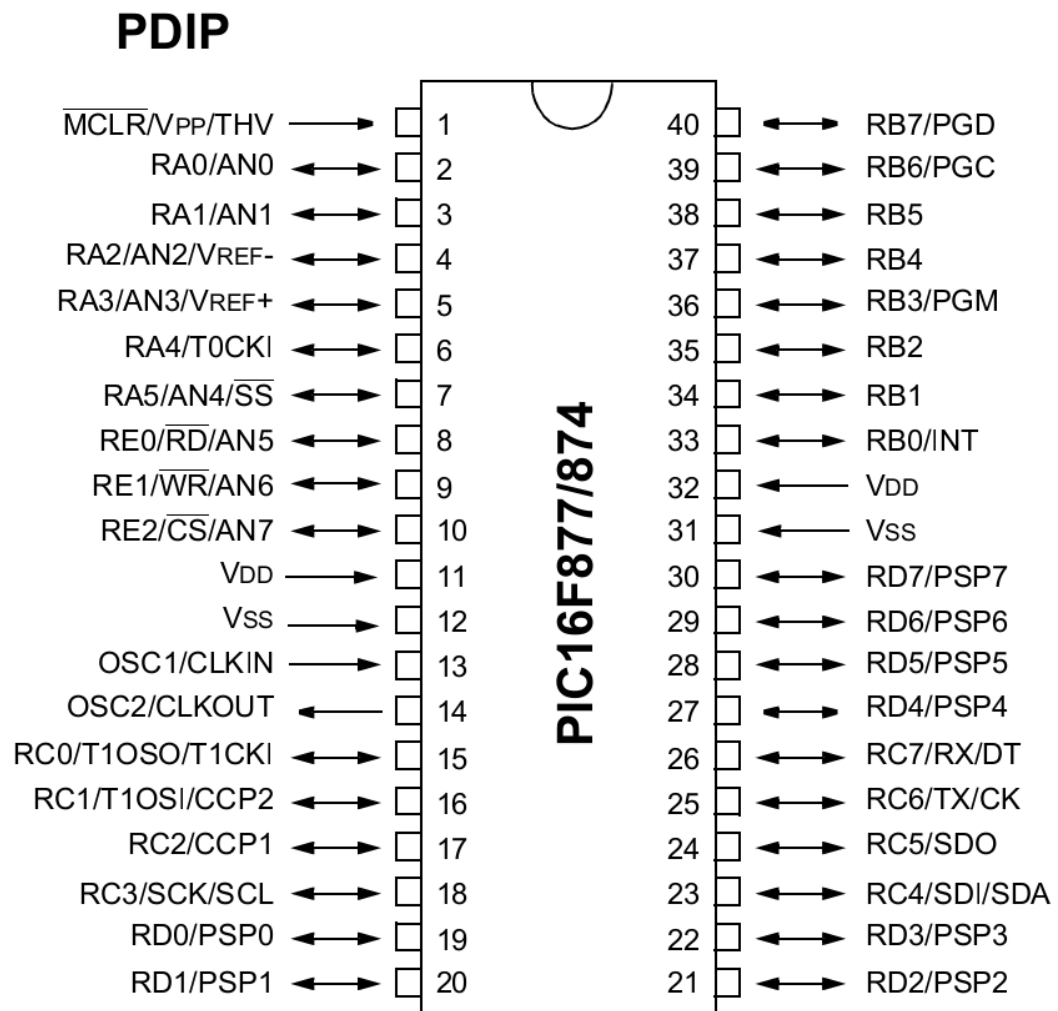


FIGURA 20 – DIAGRAMA DE PINOS DO PIC17F877

APÊNDICE 1 – PROGRAMAS EM C DO *SOFTWARE*

EMBARCADO NO PIC.

PIC 1 – PROTÓTIPO.

Este *software* tem como principal função enviar os estados das chaves via porta serial, receber os dados da porta serial, comparar com os estados das chaves e enviar a comparação para os LED's. o diagrama de fluxo deste código é mostrado na Figura 21.

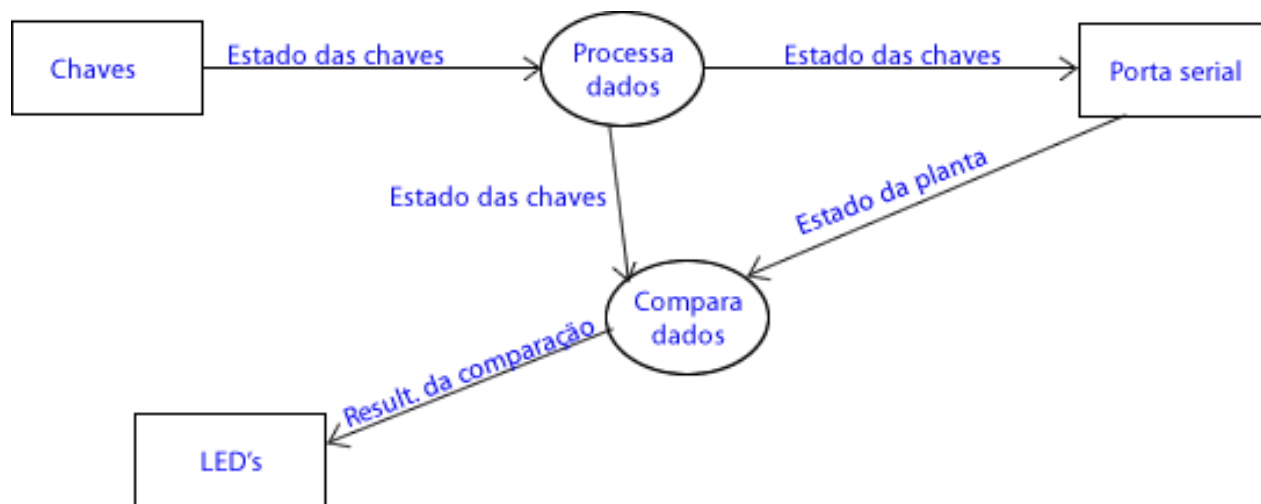


FIGURA 21 – DIAGRAMA DE FLUXO

O código é mostrado a seguir:

```

#include <16F877A.h>
#device adc=8

#FUSES NOWDT           //No Watch Dog Timer
#FUSES XT              //Crystal osc <= 4mhz
#FUSES PUT             //Power Up Timer
#FUSES NOPROTECT      //Code not protected from reading
#FUSES NODEBUG        //No Debug mode for ICD
#FUSES NOBROWNOUT     //No brownout reset
  
```

```

#FUSES NOLVP                //No low voltage prgming, B3(PIC16) or
B5(PIC18) used for I/O
#FUSES NOCPD                //No EE protection
#FUSES NOWRT                //Program memory not write protected

#use delay(clock=4000000)
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)

int8 dado = 0;
int8 serial = 0;
int8 saida = 0;
INT1 controle1=0;
INT1 controle2=0;
INT1 controle3=0;

#int_RDA                //INTERRUPÇÃO DA RECEPÇÃO SERIAL
void RDA_isr(void) //TODA VEZ QUE CHEGA UM DADO NA SERIAL
{
    //ELE É SALVO NA VARIÁVEL "serial"
    serial = getc(); //ESSE DADO JÁ VEM COMPLETO DE 8 bits
}

void main()
{
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);

```

```

enable_interrupts(INT_RDA);
enable_interrupts(GLOBAL);

while(TRUE)
{
    dado = input_B();          //PEGA O VALOR DAS CHAVES E MONTA
UMA VARIÁVEL
    controle1=input(PIN_A1);
    controle2=input(PIN_A2);
    controle3=input(PIN_A3);

    if(controle1==0)
        putc(dado);          //envia o conteudo das chaves para a planta via
serial.

    if(controle2==0)
        saida = (serial ^ dado); //FAZ A COMPARAÇÃO ENTRE O QUE ESTÁ
NAS CHAVES
                                //COM O DADO QUE CHEGOU DA SERIAL. A
COMPARAÇÃO
                                //É UMA OPERAÇÃO BIT-A-BIT XOR. SÓ VALE 1,
QUANDO FOREM DIFERENTES

        {output_D(saida);//MOSTRA O RESULTADO DA OPERAÇÃO BIT-A-
BIT NA PORTA_B
        delay_ms(500);
        output_D(0);
        delay_ms(500);
        }
    }
}

```

PIC 2 – PLANTA

Este código é responsável por receber estímulos da entrada serial e envia-los para os LED's ou mandar para os LED's o estado das chaves. A decisão entre enviar os dados das chaves ou os dados da porta serial é definida através de outras chaves. A saída então é enviada para a porta serial e para os LED's. O diagrama de fluxo deste código é mostrado na Figura 22

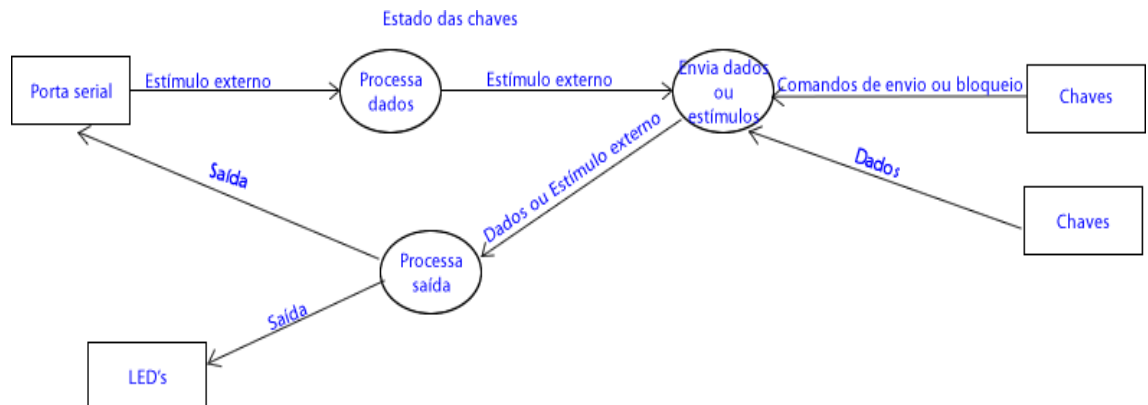


FIGURA 22 - \ DIAGRAMA DE FLUXO

O código é mostrado a seguir

```

#include <16F877A.h>
#device adc=8

#FUSES NOWDT           //No Watch Dog Timer
#FUSES HS              //High speed Osc (> 4mhz)
#FUSES PUT             //Power Up Timer
#FUSES NOPROTECT      //Code not protected from reading
#FUSES NODEBUG        //No Debug mode for ICD
#FUSES NOBROWNOUT     //No brownout reset
#FUSES NOLVP          //No low voltage prgming, B3(PIC16) or
B5(PIC18) used for I/O
#FUSES NOCPD          //No EE protection
#FUSES NOWRT          //Program memory not write protected

#use delay(clock=4000000)

```

```

#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)

int8 dado = 0;
int8 chaves=0;
int1 controle=0;

#int_RDA //INTERRUPÇÃO DA RECEPÇÃO SERIAL
void RDA_isr(void) //TODA VEZ QUE CHEGA UM DADO NA SERIAL
{
    //ELE É SALVO NA VARIÁVEL "serial"
    dado = getc(); //ESSE DADO JÁ VEM COMPLETO DE 8 bits
}

void main()
{
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);

    while(1)
    {

        chaves = input_B(); //RECEBE OS DADOS DAS CHAVES NA PORTA_B
        controle=input(PIN_A1);

        if(controle==1)

```

```
{
    output_D(chaves); //ENVIA O VALOR DAS CHAVES PARA OS LEDS
    NA PORTA_D

    putc(chaves);//ENVIA PARA A PORTA SERIAL O VALOR LIDO DAS
    CHAVES NA PORTA_B
}
if(controle==0)
{
    output_D(dado); //ENVIA O VALOR DAS CHAVES PARA OS LEDS
    NA PORTA_D

    putc(dado);//ENVIA PARA A PORTA SERIAL O VALOR LIDO DAS
    CHAVES NA PORTA_B
}
}
}
```