

Automação residencial de baixo custo: um protótipo com acesso web

Ivan Vieira Ferreira da Silva
Engenheiro Eletricista pelo Centro Universitário CESMAC.
e-mail: ivanvieirafs@gmail.com

Sérgio Silva de Carvalho
Mestre em Ciência da Computação pela UFPE.
e-mail: ssc1511@gmail.com

RESUMO

Este trabalho trata da automação residencial, especificamente do desenvolvimento de um protótipo de domótica com controle via web, utilizando tecnologias com padrões abertos e de baixo custo, como o padrão de rede serial RS485, a plataforma microprocessada Arduino e o servidor web Apache. Este trabalho inclui todo o desenvolvimento de um protótipo de domótica de baixo custo, dos hardwares utilizados até os softwares desenvolvidos.

Palavras-chave: Automação residencial. Domótica. Padrões abertos. RS485. Arduino. Servidor web. Microcontroladores.

INTRODUÇÃO

A automação residencial, conhecida também como *Domótica*¹, tem como principal objetivo facilitar o dia-a-dia de seus usuários, através da gestão dos recursos habitacionais de suas residências, visando uma maior segurança, conforto, praticidade e economia. Atualmente, a automação residencial encontra-se em plena evolução tecnológica, com soluções modernas e futuristas para qualquer tipo de imóvel e classe social.

As soluções atuais para a automação de uma residência demandam um alto custo para sua implementação. Recursos como controle de iluminação com cenas pré-programadas, controle de temperatura de ambientes, fechaduras biométricas, integração com sistemas de áudio e vídeo e acionamento elétrico de persianas e janelas, são exemplos do que pode ser feito com a tecnologia atual voltada para automação residencial.

Segundo estimativa da Associação Brasileira de Automação Residencial (Aureside), os preços desses recursos caíram pela metade nos últimos quatro anos. “A

¹ Domótica – O termo “Domótica” resulta da junção da palavra latina “Domus” (casa) com “Robótica” (controle automatizado de algo).

automação custava cerca de 5% do valor do imóvel. Agora representa 3%.” Afirma o engenheiro José Roberto Muratori, fundador da Aureside (LEAL, 2011).

Porém, mesmo com essas quedas de preços, o custo para se ter uma casa automatizada ainda é alto, quando se trata de uma automação simples para uma residência de classe média, visto que esta classe, ainda não vê com bons olhos a automação residencial devido ao seu alto investimento. Contudo, soluções de baixo custo são alternativas para este tipo de automação, de pequenas e médias residências, buscando assim se ter uma maior aceitação da automação residencial nesta classe social.

Portanto, o presente trabalho visa mostrar uma solução de baixo custo para tal problema. Usando chips microcontroladores e uma rede RS-485 para comunicação entre eles, será desenvolvido um sistema capaz de controlar a iluminação e equipamentos elétricos de uma residência, através da rede local ou *Internet*.

Ele foi desenvolvido através de estudos bibliográficos, publicações *online* e de experimentos práticos em circuitos eletrônicos utilizando *protoboard*². Nos estudos bibliográficos, as informações necessárias foram coletadas a partir de livros, revistas e monografias. No que se referem a publicações online, as informações foram coletadas de sites e artigos da *Internet*. Por fim, com todas as informações necessárias para o desenvolvimento do trabalho coletadas, os conhecimentos adquiridos foram postos em prática em circuitos eletrônicos utilizando *protoboard*.

1. DESENVOLVIMENTO DO PROTÓTIPO

Inicialmente o protótipo seria composto por microcontroladores Arduino, uma rede RS485 e um servidor web utilizando um *Shield Ethernet* da plataforma Arduino, porém, este servidor não foi implementado, devido a alguns recursos que o mesmo não possuía e que eram necessários para o protótipo. Preferiu-se então a implementação de um servidor web a partir de um microcomputador com o *Apache Web Server* instalado, pois o mesmo possui os recursos desejados e é um servidor bastante utilizado no mundo todo.

² Protoboard – Conhecida também como “matriz de contatos” ou “placa de ensaio”, protoboard é uma placa com furos e conexões condutoras para montagem de circuitos elétricos experimentais.

O protótipo é baseado numa rede de comunicação mestre/escravo, onde um microcomputador dotado de um *software* escrito em linguagem VB.NET será o mestre e microcontroladores da plataforma Arduino (programados em linguagem C) serão os escravos. Para a interação entre o usuário e os equipamentos da residência, foi desenvolvido um *software* escrito em php, rodando a partir do servidor *web*, tornando capaz o controle e supervisão do sistema através de qualquer computador ou dispositivo que contenha um browser e esteja conectado na rede local ou *Internet*.

Serão descritas agora suas características de *hardware* e *software*, bem como os problemas encontrados em seu desenvolvimento e suas soluções.

2. HARDWARE

O *Hardware* é composto por microcontroladores e relés, rede de comunicação que utiliza circuitos integrados conversores de sinais seriais de padrões RS232 e RS485, e de um microcomputador utilizado como servidor *web* e mestre da rede RS485.

Na Figura 1 pode ser visto um diagrama do modelo do protótipo de domótica, com todos os seus componentes de *hardware*.

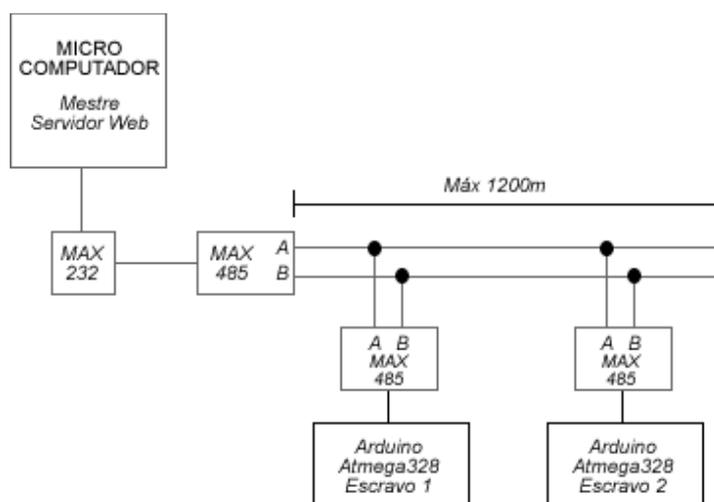


Figura 1 – Diagrama do protótipo de domótica.

2.1 Os Microcontroladores (escravos)

Como dito anteriormente, os microcontroladores utilizados foram os da plataforma Arduino, especificamente o modelo Atmega328, que compõe a placa Duemilanove. Os microcontroladores da plataforma Arduino foram escolhidos principalmente por sua facilidade de programação, característica essencial para um rápido desenvolvimento de projetos microcontrolados.

Além da facilidade de programação, outra característica que ajudou na sua escolha foi a gravação descomplicada do programa no microcontrolador, que não necessita de gravadores específicos, necessitando apenas de um cristal oscilador e de alguns resistores.

Os microcontroladores são usados para o controle dos equipamentos elétricos da residência. Foi criado um módulo de acionamento e leitura, composto por quatro relés e um sensor de temperatura, como pode ser visto na Figura 2. Foram utilizadas quatro portas de saída digital do microcontrolador para o acionamento de lâmpadas e tomadas e quatro portas de entrada digital para a leitura de sensores de porta e interruptores de lâmpadas. Foi utilizada apenas uma porta analógica do microcontrolador, configurada como entrada analógica para a leitura de um sensor de temperatura.

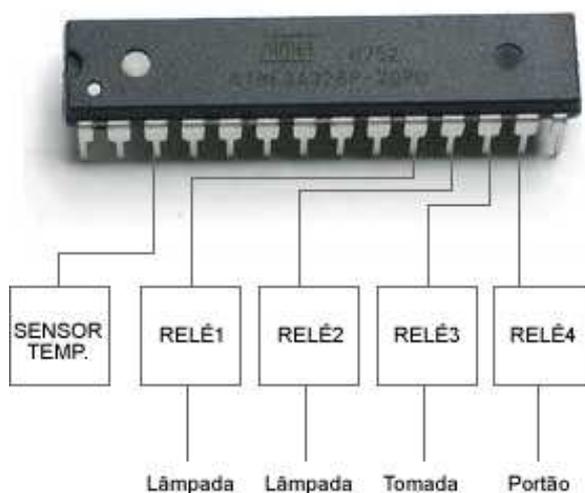


Figura 2 – Exemplo do módulo de acionamento e leitura.

O módulo de acionamento foi idealizado para ser introduzido em uma caixa de tomadas, interruptor ou caixa de passagem, portanto deveria ter no máximo dois relés e não

quatro como os que foram vistos na Figura 2. Idealmente o seu projeto deveria ser em placa de circuito impresso, mas o mesmo foi montado em *protoboard*, como todo o resto do protótipo, ficando como sugestão para trabalhos futuros a miniaturarização do mesmo em placa de circuito impresso.

2.2 A rede de comunicação

A rede de comunicação escolhida foi a rede RS485, por ser uma rede robusta e bastante utilizada, por poder operar no modo multiponto, possuir capacidade de comunicação com cabos de grandes comprimentos e facilidade de conversão do padrão RS232 para o padrão RS485, características essenciais para a rede deste protótipo.

Para que os dispositivos pudessem se comunicar por esta rede, foram necessários circuitos integrados conversores, que serviram para converter os sinais do padrão RS232 do mestre (microcomputador) para o padrão RS485 e os sinais TTL dos escravos (Arduíno) para RS485.

A seguir, pode ser visto o modo de ligação destes circuitos integrados na rede de comunicação, tanto para o mestre quanto para os escravos. A Figura 3 mostra a ligação do CI MAX232 para converter os sinais da interface RS232 do mestre para os sinais TTL. Já a Figura 4, mostra a ligação do CI MAX485 para converter os sinais TTL dos microcontroladores e do MAX232 do mestre para os sinais do padrão RS485.

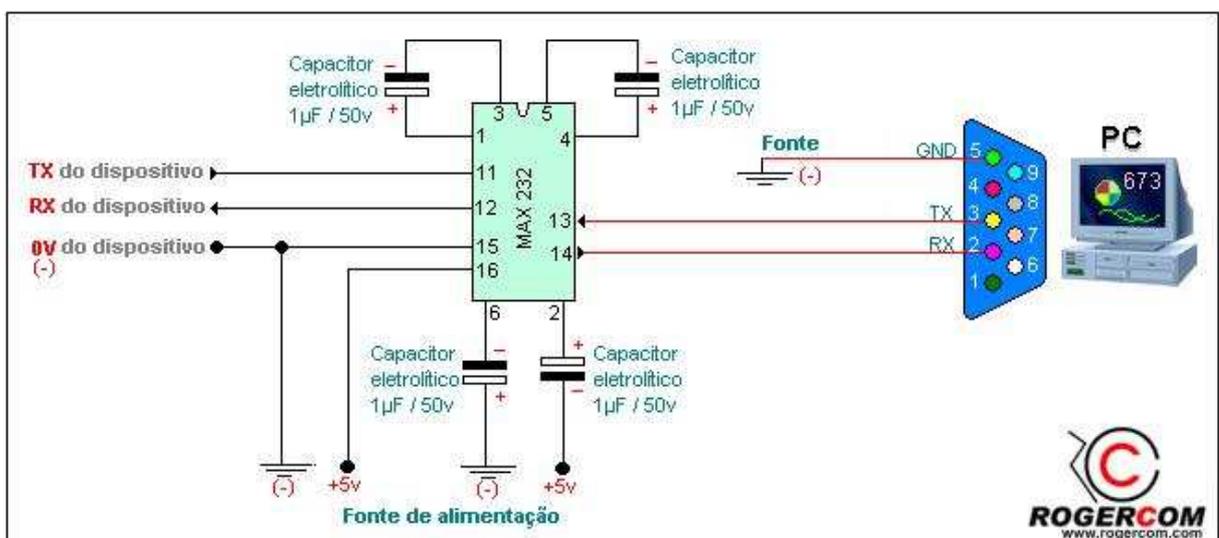


Figura 3 – Ligação do CI MAX232 na interface RS232 para conversão TTL/RS232.
Autor: Rogercom.com

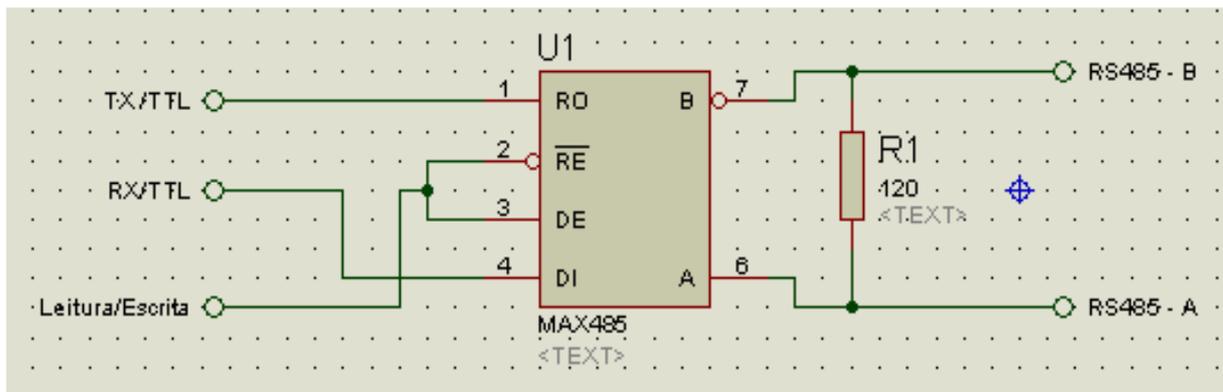


Figura 4 – Ligação do CI MAX485 para conversão de TTL/RS485.
 Autor: Rogercom.com

Neste tipo de rede, por definição do padrão RS485, é possível conectar até 32 dispositivos na mesma. Neste protótipo, foram utilizados apenas dois dispositivos, ou seja, dois escravos, mas nada impede que sejam adicionados mais dispositivos a esta rede.

2.3 O Mestre e o Servidor Web

Inicialmente, o servidor *web* seria feito utilizando um *Shield Ethernet* Arduino, componente da plataforma Arduino que adiciona basicamente uma comunicação *ethernet* no microcontrolador. Após alguns testes com o *Shield ethernet*, sua utilização foi descartada, pois o mesmo apresentou uma implementação complicada e recursos limitados, como por exemplo, o suporte apenas da linguagem HTML, limitando o seu uso apenas para simples aplicações, diferente de uma página mais elaborada, com uma maior interação entre o usuário e o sistema, como é o caso de uma página escrita na linguagem PHP sendo executada a partir de um servidor *web* padrão em um microcomputador qualquer.

Portanto, um microcomputador foi utilizado para ser o servidor *web*, pois nele seria capaz de ser instalado o *Apache Web Server*, um servidor *web* padrão de páginas para *Internet*, que pode ser utilizado tanto para páginas *web* escritas em HTML quanto para páginas escritas em PHP. Este mesmo microcomputador foi utilizado como mestre da rede, tendo a capacidade de interagir com os escravos e a página escrita em PHP, utilizando um *software* escrito na linguagem VISUAL BASIC.NET. As principais características do microcomputador utilizado, bem como os *softwares* nele instalados, são:

- a) Processador de 3,14GHz;
- b) Memória RAM de 1GB;
- c) Porta Serial padrão RS232;
- d) Porta *Ethernet* 10/100 Mbps;
- e) Servidor Apache *Web Server* versão 2.2.17;
- f) PHP versão 5.3.5 instalado junto com o Apache;
- g) Circuito conversor RS232/RS485 (MAX232 e MAX485).

Vale ressaltar que para este servidor poderia ter sido escolhido algum outro microcomputador de menor capacidade de processamento, mas necessariamente ser dotado de uma porta serial RS232 (ou conversor USB/RS232) e uma porta *Ethernet*. A seguir, será visto mais detalhadamente todos os *softwares* desenvolvidos neste protótipo.

3. SOFTWARE

A parte de *software* do protótipo de domótica é justamente o protocolo de comunicação da rede, que é composta de dois *softwares*, um do mestre e um dos escravos, além de mais um software escrito em PHP que serve como interface entre o usuário e o sistema de automação.

O protocolo de comunicação é baseado no envio de *strings*³ de comandos pela rede RS485. Estas *strings* de comandos são formadas por um conjunto de caracteres que formam um pacote de informações. A Figura 5 mostra a estrutura de uma *string* de comando utilizada no protótipo.

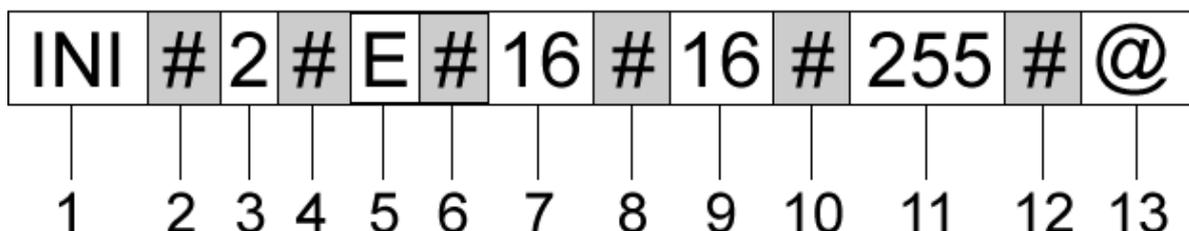


Figura 5 – Modelo da string de comando utilizada no protocolo do protótipo.

³ *String* - Em programação e em linguagens formais, uma cadeia de caracteres (também conhecida como samblagem ou string) é uma seqüência ordenada de caracteres (símbolos) escolhidos a partir de um conjunto pré-determinado.

Como foi visto na Figura 5, a *string* é dividida em pequenos blocos, onde cada bloco possui caracteres com funções específicas. Os números de 1 a 13 na Figura 5 representam os blocos de comandos da string. Pode ser visto a seguir, na Tabela 1, um detalhamento da *string* de comando, onde são mostradas as funções de cada um de seus blocos.

Tabela 1 – Detalhes dos blocos de uma *string* de comandos.

BLOCO	CARACTERES	SIGNIFICADO	FUNÇÃO
1	INI	Início de uma string válida	Utilizado para informar os escravos que o início da string é válida.
2	#	Separação	Utilizado apenas para organização da string.
3	2 ou 3	Endereço	Informa aos escravos qual endereço a string de comandos pertence.
4	#	Separação	Utilizado apenas para organização da string.
5	E ou L	Leitura ou Escrita	Utilizado para sinalizar se a string é de escrita ou de leitura.
6	#	Separação	Utilizado apenas para organização da string.
7	01 a 16	Entradas digitais	Utilizado para informar o estado de 4 saídas digitais. 01 a 16 em decimal corresponde a 4bits.
8	#	Separação	Utilizado apenas para organização da string.
9	01 a 16	Saídas digitais	Utilizado para informar o estado de 4 entradas digitais. 01 a 16 em decimal, corresponde a 4bits.
10	#	Separação	Utilizado apenas para organização da string.
11	001 a 255	Entrada ou Saída Analógica	Utilizado para a leitura ou escrita analógica. 001 a 255 em decimal, corresponde a 8bits.
12	#	Separação	Utilizado apenas para organização da string.
13	@	Fim de uma string válida	Utilizado para informar o fim de uma string válida.

Os três *softwares* do protótipo que tratam estas strings são:

a) Página PHP – *Software* escrito na linguagem PHP, utilizado como interface de comunicação entre o usuário do sistema de automação e os equipamentos nele instalados.

b) *Software* do Mestre – *Software* escrito na linguagem VISUAL BASIC.NET, utilizado para acessar a portal serial RS232 do mestre e repassar as informações tanto para os escravos quanto para a página PHP.

c) *Software* Escravos – *Software* escrito na linguagem C, utilizado dentro dos microcontroladores arduino para controlar os equipamentos a eles ligados, além de torná-los capazes de se comunicar com o mestre.

3.1 Páginas PHP

As páginas *web* escritas em PHP não fazem parte diretamente do protocolo de comunicação, mas é uma parte essencial do protótipo, que serve como meio de acesso para o usuário controlar e supervisionar o sistema de automação pela rede local e *Internet*.

A princípio, estas páginas teriam a função de mestre no protótipo, acessando informações diretamente da porta serial do servidor. Foram feitos testes de funcionamento de uma página escrita em PHP acessando diretamente a porta serial. Os resultados destes testes não foram satisfatórios, pois tanto no envio quanto na recepção de dados utilizando a porta serial do servidor, a página em PHP se mostrou demasiadamente lenta na execução do programa, comprometendo todo o sistema.

Contudo, as páginas em PHP ficaram responsáveis apenas pela interação do usuário com o controle e supervisão do sistema, deixando a função de mestre do protocolo de comunicação para outro *software* específico, escrito em linguagem Visual Basic .NET.

Como pode ser visto na Figura 6, a página principal escrita em PHP possui três opções. A opção um, Controle de Equipamentos, leva o usuário para outra página que lista todos os equipamentos que podem ser controlados no momento pelo sistema. Ao lado de cada equipamento listado, existem duas opções, as opções ligar e desligar. No final da lista, possui um botão, chamado de “Alterar estado”, que ao ser pressionado carrega as opções escolhidas pelo o usuário e em seguida monta uma *string* de comandos, gravando-a em um arquivo de texto chamado comando_php.txt, para que o mestre possa ler e enviá-la aos escravos. Além de gravar a *string* montada no arquivo comando_php.txt, a página PHP grava mais um arquivo, chamado str_phpenvia.txt, que avisa ao mestre que existe uma nova *string* para ser enviada.

Já na opção dois, Supervisão do Sistema, o usuário será levado para uma página que solicitará ao *software* mestre uma *string* de comandos de leitura, contendo informações atuais dos estados das entradas e saídas dos escravos. Esta página grava um arquivo chamado *str_phprecebe.txt*, que serve para avisar ao mestre que existe uma nova solicitação de *string* de leitura. Após esta solicitação, o mestre faz a leitura nos escravos e grava a *string* em um arquivo chamado *comando_mestre.txt*, para que a página de supervisão possa lê-la.



Figura 6 – Página principal das páginas escritas em PHP.

A opção três na página principal, Configurações do Sistema, não foi implementada devido ao pouco tempo para a conclusão do protótipo, ficando como sugestão para trabalhos futuros. A página configurações do sistema seria desenvolvida com o propósito de adicionar ou remover equipamentos e sensores que poderiam ser controlados e supervisionados pela rede. A seguir, pode ser visto na Figura 7 o fluxograma das páginas em escritas em PHP.

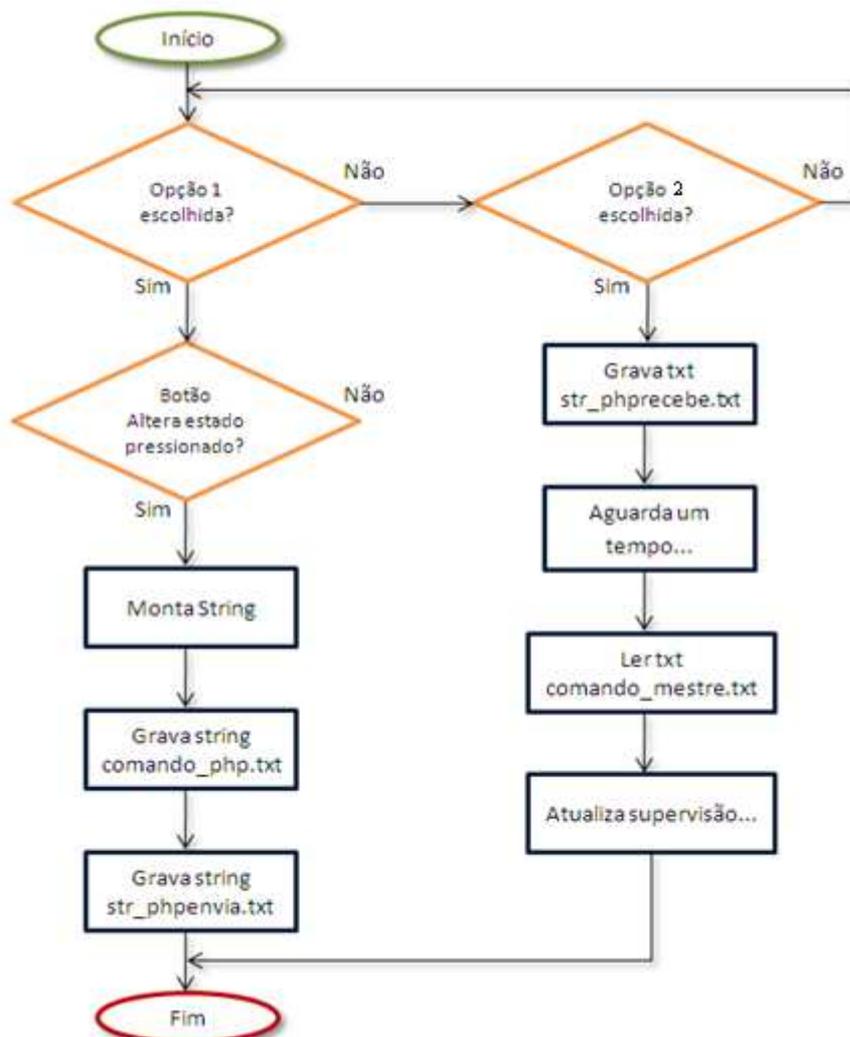


Figura 7 – Fluxograma das páginas PHP.

3.2 O software do mestre

O mestre do protocolo de comunicação do protótipo foi desenvolvido utilizando a linguagem Visual Basic.NET. Este *software* é responsável por todo o fluxo de informação na rede. Sua principal função é acessar a portal serial RS232 do servidor e permitir a troca de informações entre as páginas PHP e os escravos. O seu funcionamento é bastante simples. Ele ficará permanentemente verificando alterações nos arquivos de texto `str_phpenvia.txt` e `str_phprecebe.txt`, arquivos estes criados pelas páginas PHP, e que foram comentados anteriormente. O conteúdo desses dois arquivos indicará para o mestre se ele enviará ou receberá uma nova *string* de comandos dos escravos.

Caso estes arquivos indiquem uma requisição de uma nova *string* de leitura ao mestre, ele irá fazer uma requisição de leitura aos escravos e aguardará um tempo para que

eles respondam com a *string* de leitura solicitada. Após receber a *string* solicitada, o mestre irá gravá-la em um arquivo de texto chamado comando_mestre.txt, que será lido pela página PHP para poder atualizar sua tela de supervisão. Se agora os arquivos de texto indicar ao mestre que ele terá que enviar uma nova *string* de comandos de escrita nos escravos, ele irá ler o arquivo de texto chamado comando_php.txt, que contém a última *string* de comandos de escrita solicitada pela página de controle de equipamentos, e em seguida enviará a *string* para os escravos. Na Figura 8 pode ser visto o fluxograma do *software* mestre.

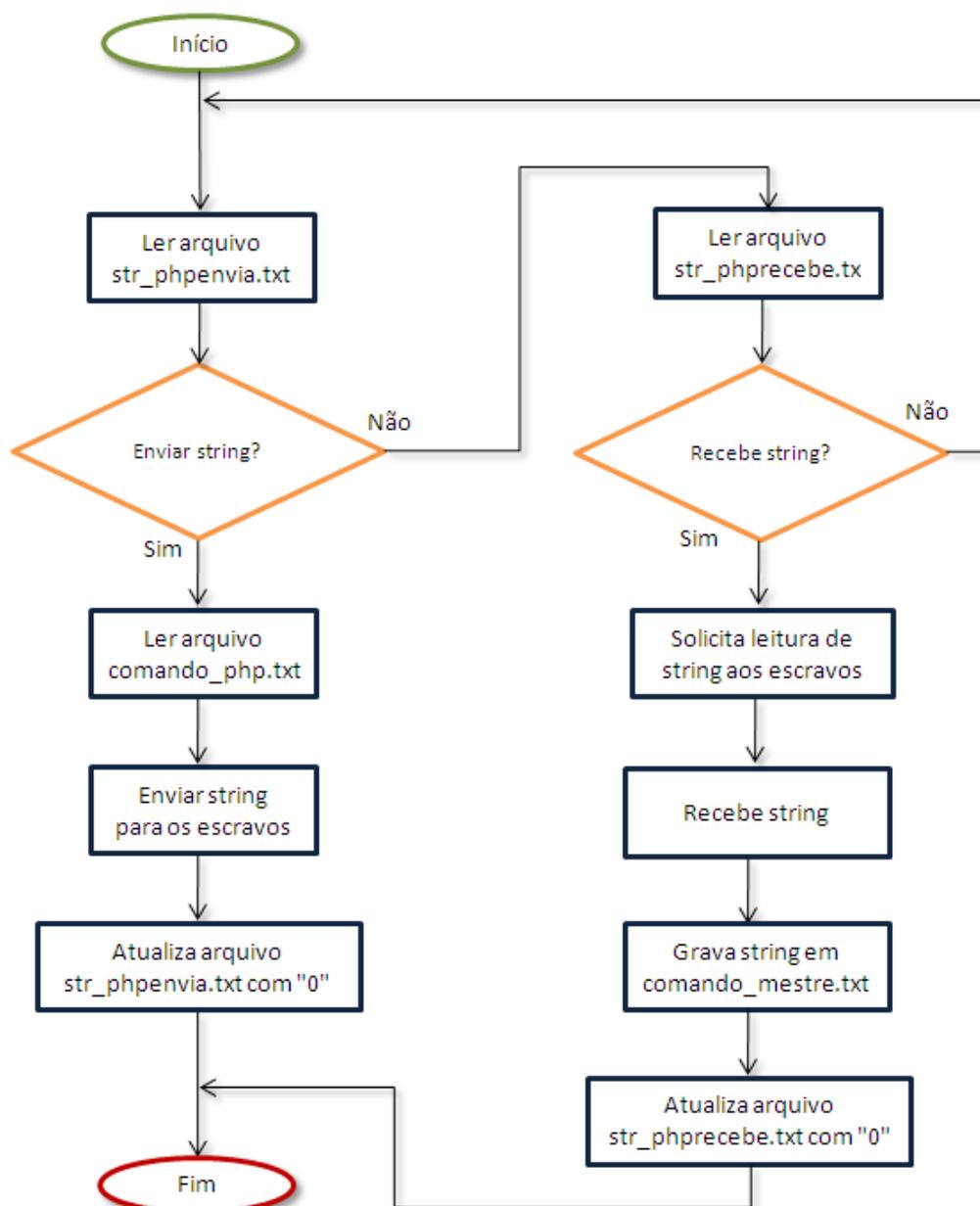


Figura 8 – Fluxograma do *software* do mestre.

3.3 O Software dos escravos

O *software* dos escravos é um único *software* para todos, mudando apenas algumas informações, como o endereço do escravo na rede e se ele é um módulo que permite entrada ou saída analógica.

O *software* é escrito em linguagem C. Ele ficará permanentemente aguardando uma *string* de comandos do mestre. Caso receba, ele verificará no bloco de endereço da *string* se contém seu endereço. Se o endereço for o seu, ele tratará a *string* de comandos e irá verificar se a mesma é de leitura ou escrita, através do bloco cinco da *string* de comandos.

Na Figura 9 pode ser visto o fluxograma do *software* escravo.

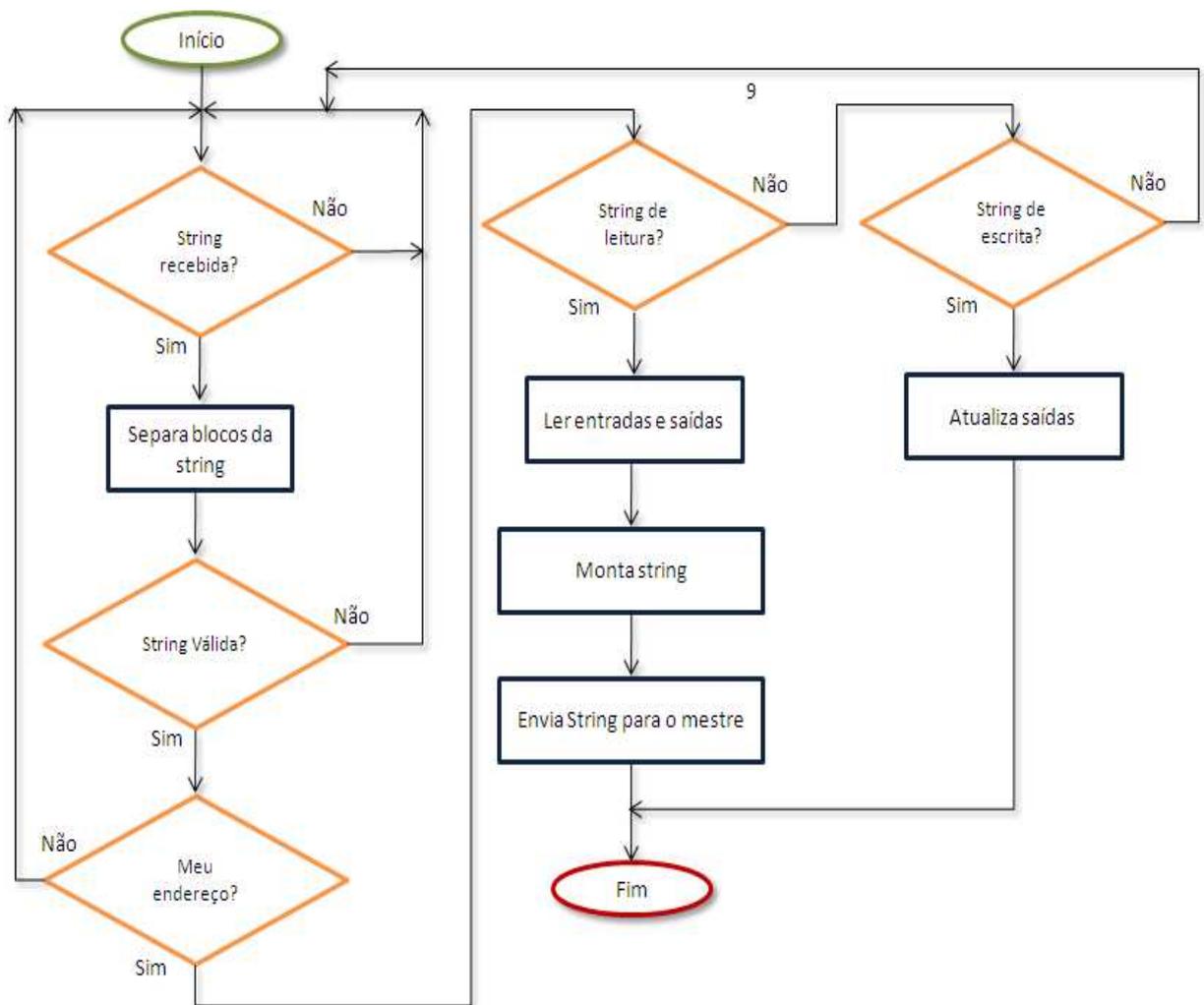


Figura 9 – Fluxograma do *software* dos escravos.

4. ACESSO REMOTO VIA WEB

As páginas escritas em PHP funcionaram perfeitamente pela rede local e pela Internet. Para a visualização das páginas pela rede local não foi preciso nenhuma configuração. As páginas escritas em PHP foram adicionadas à pasta raiz de páginas *web* do servidor Apache, onde na versão utilizada esta pasta é chamada de “WWW”.

Para o acesso via rede local, bastou digitar o endereço `http://localhost/index.php:8888`, onde “*localhost*” é o endereço local padrão do servidor *web* Apache, e `index.php`, a página principal que contém as opções de controle de equipamentos e supervisão do sistema. O caractere “:” seguido dos números 8888, avisam ao navegador que a porta utilizada não é a padrão 80 do protocolo de *Internet* HTTP, e sim a 8888, porta configurada nas opções do servidor Apache.

Porém, para o acesso via *web*, foi preciso configurar o roteador de acesso à *Internet* para redirecionar todas as requisições feitas na porta 80 do roteador, para a porta 8888 do IP do servidor *web*. Esta configuração é bastante simples e diferente em cada tipo de roteador, por isso não foi descrita neste trabalho, pois não faz parte de seu escopo.

CONCLUSÃO

Mesmo com alguns problemas encontrados e o curto tempo para o término do protótipo, o seu desenvolvimento possibilitou alcançar os objetivos deste trabalho, mostrando-se capaz de realizar suas funções básicas esperadas, como o controle e supervisão dos equipamentos conectados aos escravos (microcontroladores) através da *Internet*. Os problemas encontrados, como o funcionamento inadequado das páginas php servindo como mestre da rede, e os recursos limitados do *Ethernet Shield* como servidor *web*, foram contornados através de novas soluções, como o uso de um outro *software* escrito em VB.NET, para servir como mestre da rede, e também o uso de um microcomputador como servidor *web* dedicado.

Contudo, estas soluções não afetaram diretamente o objetivo geral deste trabalho, mas demandaram um tempo considerável para que fossem encontradas e implementadas, comprometendo assim o andamento do cronograma e conseqüentemente a conclusão do protótipo no tempo esperado. Apesar do comprometimento do cronograma, o desenvolvimento do protótipo foi finalizado, mas não por completo, pois alguns recursos não foram implementados a tempo, como a página de configurações do sistema e os módulos de acionamento confeccionados em placas de circuito impresso, ficando como sugestão para serem desenvolvidos em trabalhos futuros.

Pôde-se concluir então, que é possível automatizar pequenas residências de classe média, utilizando soluções e tecnologias com padrões abertos e de baixo custo, sendo possível o seu controle via rede local e *Internet*.

REFERÊNCIAS

ATMEL, Semicondutores. Disponível em:

<<http://http://www2.atmel.com/about/corporate/default.aspx>> Acesso em 20 de abril de 2011.

BOLZANI, Caio Augustus Morais. **Artigo Desmistificando a Domótica**. Disponível em:

<<http://www.aureside.org.br/artigos/default.asp?file=01.asp&id=74>>. Acesso em 20 de março de 2011.

CANZIAN, Edmur. **MINICURSO Comunicação Serial RS232**. São Paulo. 2011.

Disponível em: <http://http://www.professores.aedb.br/arlei/AEDB/Arquivos/rs232.pdf>.

Acesso em 20 de abril de 2011.

CERNE TECNOLOGIA, Projetos Eletrônicos. Disponível em: <http://www.cerne-tec.com.br>.

Acesso em 5 de maio de 2011.

GIMENEZ, Salvador Pinillos. **Microcontroladores 8051: Teoria de Hardware e Software, Aplicações em controle digital e Laboratório/Simulação**. Pearson Education do Brasil LTDA. São Paulo, 2002.

HERMES, Pedro da Paschoa Filho. **Objetivos da Automação Residencial**. Disponível em:

<<http://www.lpaarquitectura.com.br/dicas/objetivos-da-automacao-residencial>>. Acesso em 20 de março de 2011.

KOPELVSKI, Mykon Max. **Teoria de CLP**. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo. 2010. Disponível em:

<http://www.cefetsp.br/edu/maycon/arqs/ap_clp_rev00.pdf> Acesso em 28 de abril de 2011.

LAMBRECHT, Ariel. **Rede de Dados e Integração da Manufatura por Computador**. 2006. Disponível em: <<http://www.ebah.com.br/content/ABAAAABUGAA/x10>>. Acesso em 10 de maio de 2011.

LEAL, Renata. **INFO EXAME – Viva na casa do futuro hoje**. Disponível em: <<http://info.abril.com.br/noticias/tecnologia-pessoal/viva-na-casa-do-futuro-hoje-17012011-2.shl>>. Acesso em 22 de março de 2011.

LEÃO, Prof. Vitor. **RS485 – Especificação e Utilização**. Universidade Federal da Bahia. 2007. Disponível em: <www.e-science.unicamp.br/angra/.../publicacao_616_RS485.pdf>. Acesso em 30 de maio de 2011.

NETO, Aristides Domingues Leite; MENON, Renato de Oliveira. **Monitoramento e Controle Residencial via Software**. 2004. Monografia (Graduação em Engenharia da Computação). Faculdade de Engenharia de Sorocaba – FACENS. Disponível em: <http://www.cseg.eng.br/repositorio/tcc/TCC_MCRVS.pdf> Acesso em 28 de abril de 2011.

NUNES, Renato Jorge Caleira. **Análise comparativa de tecnologias para domótica**. 2002. Disponível em: <<http://www.scribd.com/doc/26476683/02-JEACI02>> Acesso em 10 de março de 2011.

PINTO, Paulo Henrique. **Funcionamento de um Controlador Lógico Programável (CLP)**. 2008. Disponível em: <http://www.pharmaster.com.br/artigos/docs/20080703_7439_Funcionamento%20de%20um%20CLP.pdf> Acesso em 22 de março de 2011.

PINHEIRO, José Maurício Santos. **Falando de Automação Predial**. 2004. Disponível em: <http://www.projetoderedes.com.br/artigos/artigo_falando_de_automacao_predial.php> Acesso em 22 de março de 2011.

RIBEIRO, Marcos Antônio. **Automação Industrial**. 4. Ed. SALVADOR, Tek Treinamento & Consultoria LTDA, 1999.

SILVA, Eduardo de Moraes V; SOBRINHO, Fillipe Y Saad; AQUINO, José Adriany V de. **Protocolo RS485**. 2006. Monografia (Ciência da Computação). Departamento de Ciência da computação. Universidade de Brasília. Disponível em: <http://www.cic.unb.br/~bordim/TD/Arquivos/G10_Monografia.pdf>. Acesso em 28 de março de 2011.

SOUZA, Cláudio de Medeiros Chaves. **SISTEMA EMBARCADO: Uma aplicação voltada para o monitoramento das faltas à terra de uma subestação Eletrobrás Chesf**. 2010. Monografia (Graduação em Análise de Sistema.). Centro de Estudos Superiores de Maceió. Disponível em: <<http://www.cesmac.com.br>> Acesso em 23 de março de 2011.

SOUZA, Vitor Amadeu. **Artigo Comunicação RS232 e RS485**. 2011. Disponível em: <<http://www.cerne-tec.com.br>> Acesso em 27 de maio de 2011.

UOL Economia. **Brasil é “quase” um país de classe média**. 2010. Disponível em: <<http://economia.uol.com.br/ultimas-noticias/redacao/2010/05/12/brasil-e-quase-um-pais-de-classe-media-diz-mantega.jhtm>> Acesso em 10 de junho de 2011.

VARGAS, Alessandra Antunes. **Estudo sobre Comunicação de Dados via Rede Elétrica para Aplicações de Automação Residencial/Predial**. 2004. Monografia (Engenharia da Computação). Instituto de Informática. Universidade Federal do Rio Grande do Sul. Disponível em: <<http://www.ece.ufrgs.br/~fetter/plt/TrabalhoConclusaoAlessandra.pdf>>. Acesso em 22 de abril de 2011.