

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

**PROTÓTIPO DE UM SISTEMA DE SEGURANÇA**  
**RESIDENCIAL COM LINUX EMBARCADO**

**DANIEL BAUMANN**

**BLUMENAU**  
**2008**

**2008/1-09**

**DANIEL BAUMANN**

**PROTÓTIPO DE UM SISTEMA DE SEGURANÇA**

**RESIDENCIAL COM LINUX EMBARCADO**

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

Prof. Miguel Alexandre Wisintainer - Orientador

**BLUMENAU  
2008**

**2008/1-09**

# **PROTÓTIPO DE UM SISTEMA DE SEGURANÇA RESIDENCIAL COM LINUX EMBARCADO**

Por

**DANIEL BAUMANN**

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

Presidente: \_\_\_\_\_  
Prof. Miguel Alexandre Wisintainer – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Mauro Marcelo Mattos, Dr. – FURB

Membro: \_\_\_\_\_  
Prof. Antonio Carlos Tavares – FURB

Blumenau, 08 de julho de 2008

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

## **AGRADECIMENTOS**

A Deus, pela sabedoria, inteligência e força, fazendo-me chegar até aqui sem desistir, guiando e me acompanhando todos os dias.

Aos meus pais, Ervino e Sonia pela formação e educação que recebi. Obrigado pela compreensão, confiança e força que sempre recebi de vocês.

À Dani pelo incentivo, paciência e ajuda na revisão deste trabalho.

Aos meus amigos, pelos empurrões e cobranças.

Aos professores e colegas da FURB, pela amizade.

Ao Co-fundador, projetista de hardware e webmaster da ACME Systems, senhor Sergio Tanzilli por disponibilizar uma versão de Linux específica para meu modelo de Fox Board com todas as funcionalidades necessárias para o desenvolvimento do protótipo.

Ao meu orientador Miguel Alexandre Wisintainer, pelos incentivos, ajuda em momentos de dificuldade, por ter disponibilizado sua placa Fox Board nacional e por ter acreditado na conclusão deste trabalho.

A única maneira de enfrentar os obstáculos maiores é ultrapassando os obstáculos menores.

Júlio Verne

## RESUMO

Este trabalho apresenta um protótipo de um sistema de segurança residencial integrando em um só dispositivo sensores, câmera e acionamento de tomadas, tornando possível monitorar ambientes de forma remota. Para o mesmo se fez necessário implementar um hardware, responsável por gerenciar os sensores, a câmera e comunicar-se com o usuário através da internet e implementar um software em *Scripts Common Gateway Interface* (CGI), responsável por gerenciar e exibir dados e imagens ao usuário através de um navegador.

Palavras-chave: Dispositivos remotos. Sistema embarcado. Captura de imagem.

## **ABSTRACT**

This work presents a prototype of home security system integrating into one device camera sensors and connect electrical outlet, making it possible to monitor environments of remote way. For the same was necessary to implement a hardware responsible for managing the sensors, the camera, communicate with the user through the Internet and implement a software on Scripts Common Gateway Interface (CGI), responsible for managing and displaying data and images to the user through a browser.

Key-words: Remote device. Embedded system. Image capture.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de sistema cliente-servidor .....	17
Figura 2 - Arquitetura da rede internet .....	19
Quadro 5 - Trecho de código em C para CGI.....	20
Figura 3 – Versão nacional da placa Fox Board.....	21
Quadro 1 – Características de software da placa Fox Board.....	21
Quadro 2 – Características de hardware da placa Fox Board.....	22
Figura 4 – Descrições dos principais componentes da placa Fox Board.....	22
Quadro 3 – Pinagem e respectivas funções na Fox Board .....	23
Quadro 4 – Pinagem e respectivas funções na Fox Board .....	23
Figura 5 – Esquemático do hardware. ....	28
Figura 6 – Conexão com a Rede.....	29
Figura 7 – Esquemático do circuito de sensores .....	30
Figura 8 – Esquemático do circuito tomadas e alarme.....	30
Figura 9 – Pinos J6 utilizados como I/O .....	31
Figura 10 – Circuitos de tomadas, sensores e alarme.....	31
Figura 11 – Conexão de dispositivos a Fox Board nacional .....	32
Figura 12 – Diagrama de atividades da programação do leitor de sensores.....	33
Quadro 6 – Descrição das atividades do leitor de sensores.....	34
Figura 13 – Dados registrados pelo arquivo logsensores.txt.....	35
Figura 14 – Diagrama de arquitetura lógica do sistema .....	35
Figura 15 – Software flashFox destacando os comandos pra instalação da imagem do linux. 36	
Figura 16 – Localização da J8 na placa Fox Board.....	36
Figura 17 – <i>Webcompiler</i> da Acme Systems.....	37
Figura 18 – Acessando placa Fox Board utilizando protocolo FTP.....	37
Quadro 7 – Trecho de código que abre a PortG e configura os pinos de I/O.....	38
Quadro 8 – Trecho de código que grava dados dos sensores caso sejam acionados.....	39
Quadro 9 – Trecho de código que aciona ou desaciona a tomada 1.....	39
Quadro 10 – Trecho de código da página tcc.html.....	40
Figura 19 – Tela inicial do protótipo .....	40
Figura 20 – Tela do servidor de <i>streaming</i> Servfox .....	41
Figura 21 – Tela de visualização de câmera.....	42

Figura 22 – Tela de visualização do <i>log</i> de sensores.....	42
Figura 23 – Tela de controle das tomadas, iluminação e alarme.....	43
Figura 24 – Protótipo de um sistema de segurança residencial com linux embarcado .....	45

## LISTA DE SIGLAS

*AC - Alternate Current*

*CCD - Charge Coupled Device*

*CGI - Common Gateway Interface*

*CPU - Central Processing Unit*

*DC - Direct Current*

*FTP - File Transfer Protocol*

*HTML - HyperText Markup Language*

*HTTP - HyperText Transfer Protocol*

*I/O - Input/Output*

*I2C - Inter-Integrated Circuit*

*LAN - Local Area Network*

*LED - Light Emitting Diode*

*LKM - Loadable Kernel Modules*

*MAC - Medium Access Control*

*MIME - Multipurpose Internet Mail Extensions*

*NA - Normalmente Aberto*

*NF - Normalmente Fechado*

*RAM - Random Access Memory*

*RF - Requisitos Funcionais*

*RNF - Requisitos Não Funcionais*

*SPI - Serial Peripheral Interface*

*TCP/IP - Transmission Control Protocol/ Internet Protocol*

*UML - Unified Modeling Language*

URI - *Uniform Resource Identifier*

USB - *Universal Serial Bus*

WAE - *Web Application Extension*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 OBJETIVOS DO TRABALHO .....	13
1.2 ESTRUTURA DO TRABALHO .....	14
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
2.1 DOMÓTICA.....	15
2.2 CLIENTE-SERVIDOR .....	16
2.3 LINUX.....	17
2.4 PROTOCOLO HTTP .....	18
2.5 CGI .....	20
2.6 FOX BOARD .....	20
2.7 SENSORES .....	24
2.8 TRABALHOS CORRELATOS .....	25
<b>3 DESENVOLVIMENTO DO PROTÓTIPO.....</b>	<b>27</b>
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	27
3.2 ESPECIFICAÇÃO .....	27
3.2.1 Especificação do hardware.....	28
3.2.2 Especificação do software.....	32
3.3 IMPLEMENTAÇÃO .....	35
3.3.1 Técnicas e ferramentas utilizadas.....	36
3.3.2 Operacionalidade da implementação .....	40
3.4 RESULTADOS E DISCUSSÃO .....	44
<b>4 CONCLUSÕES.....</b>	<b>46</b>
4.1 EXTENSÕES .....	46
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>48</b>

# 1 INTRODUÇÃO

A preocupação com a segurança residencial é cada vez maior, tendo em vista o crescimento da violência e abuso sexual contra crianças, violência contra mulher, maus-tratos contra idosos e principalmente furtos. Existem várias maneiras de proteger-se, seja com o uso de câmeras, alarmes, sistemas de vigilância eletrônica ou vigilantes. Grande parte dos sistemas de vigilância eletrônica existentes no mercado possui monitoramento local e/ou estrutura de cabeamento específico.

A internet está entrando em fase de popularização e a queda dos preços dos microcomputadores vem fazendo cada vez mais parte das residências (MARTINS, 2007). Algumas residências são projetadas e/ou adaptadas para receberem redes de computadores, ficando mais viável utilizar esta estrutura de rede do que cabeamento específico para o sistema de segurança.

Baseado no texto acima, surge a idéia de desenvolver um protótipo de sistema de segurança residencial baseado em software embarcado em um hardware denominado Fox Board (ACME SYSTEMS, 2006), desenvolvido em uma versão nacional pelo professor e orientador Miguel Wisintainer. Trata-se de uma placa com várias entradas e saídas, com sistema operacional Linux, *Universal Serial Bus* (USB) e acesso a internet. Este dispositivo será gerenciado de qualquer microcomputador conectado a internet, através de uma interface web, utilizando o protocolo *HyperText Transfer Protocol* (HTTP), podendo visualizar imagens captadas pela câmera, acionar relés e verificar *status* dos sensores, tais como: *reed-switch* e sensor óptico.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um protótipo de um sistema de segurança residencial de monitoramento de imagens, sensores e acionamento de tomadas utilizando o hardware Fox Board, dotado do sistema operacional Linux.

Os objetivos específicos do trabalho são:

- a) criar um servidor web dentro da Fox Board que possibilite o controle de entrada e saída dos pinos de uso geral da Fox Board e visualização de imagem capturadas

por uma câmera;

- b) capturar imagens de uma câmera acoplada a porta USB da placa Fox Board;
- c) construir um circuito eletrônico de potência para acoplar os pinos da placa Fox Board aos sensores, alarme e tomadas.

## 1.2 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em quatro capítulos. O capítulo dois apresenta a fundamentação teórica necessária para o desenvolvimento do presente trabalho. Estando estruturado da seguinte forma: Domótica, Cliente-Servidor, Linux, protocolo HTTP, Fox Board, CGI, sensores e trabalhos correlatos.

No capítulo três são apresentados os principais requisitos do protótipo, especificação da estrutura, do hardware e do software. Em seguida é apresentada a implementação, técnicas e ferramentas utilizadas e operacionabilidade da implementação. Ao final deste capítulo são expostos os resultados e discussão.

Por fim no capítulo quatro são apresentadas as conclusões e sugestões para extensão do protótipo.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os aspectos teóricos relevantes à elaboração do presente trabalho, tais como: conceito e aplicações em domótica, arquitetura cliente-servidor, sistema operacional Linux, protocolo HTTP, apresentação e descrição da placa Fox Board, CGI bem como sensores e por último, são expostas algumas das características dos trabalhos correlatos que serviram de base para o desenvolvimento da ferramenta proposta.

### 2.1 DOMÓTICA

Segundo Brugnera (2007), domótica é uma tecnologia que paralela a automação industrial, ela faz um gerenciamento dos equipamentos domésticos, que por sua vez são controlados por uma central computadorizada. O termo domótica é resultado da junção das palavras *domus* (que significa casa) com robótica (que é o controle automatizado de algo). Desta forma permite o uso de dispositivos para automatizar as rotinas e tarefas da casa, distinguindo de controles normais por ter uma central que comanda tudo, sendo acoplada em um computador e/ou internet. Aplicações em domótica:

- a) automação: programar tarefas diárias individuais ou em conjunto (macros) de forma automática, permitindo reduzir o tempo gasto em rotinas;
- b) iluminação: utilizando aparelhos apropriados que permitem gerenciar os gastos de eletricidade. Juntamente com sensores de movimento e de luz, as luzes de um cômodo da casa que se encontra vazio não ficarão mais acesas, as luzes exteriores acendem automaticamente quando começa a escurecer. Para a casa ter uma aparência de estar habitada (quando não se encontra em casa), basta programar as luzes para acender a determinadas horas e em determinados cômodos. Podendo otimizar o consumo de energia tendo em conta a presença/ausência e horários;
- c) climatização: programação de horários para ativar/desativar equipamentos de aquecimento, ventilação ou ar condicionado, permitindo manter um nível de conforto. Poupança de energia com o funcionamento de acordo com os horários, presença e temperatura exterior e não esquecendo a comodidade de poder efetuar uma chamada para casa para se certificar de que realmente desligou o

aquecimento;

- d) segurança: a domótica pode atuar em diversos níveis de segurança. Sistemas auxiliados por sensores permitem detectar fugas de gás, inundações, incêndios em fase inicial, cortando imediatamente as entradas e avisando ao dono e a profissionais de manutenção ou bombeiros do sucedido de forma a serem tomadas providências. A segurança ao nível de detecção de intrusos também é relevante e levada em consideração pelo sistema. Câmeras ligadas a alguma rede de comunicação tais como, internet e celular, permitem monitorar os ambientes da casa remotamente. Estando o sistema de monitoramento integrado ao sistema de alarme, as próprias câmeras podem funcionar como sensores de presença identificando qualquer situação de invasão, acionando o alarme e gravando as imagens;
- e) comunicação: apoiando-se no avanço das novas tecnologias computacionais e de telecomunicações, a domótica vem oferecer ainda mais vantagens. Não só permite visualizar e ouvir a partir da internet diversos ângulos de uma casa, como também permite a comunicação com o sistema, desligando a televisão que ficou ligada, abaixando a cortina quando o sol incidir ou acender as luzes exteriores quando se estiver próximo de casa. O sistema áudio e vídeo e os meios multimídia vão ao encontro de pessoas que necessitam de cuidados especiais, permitindo a vigilância, e estes por sua vez têm um meio à sua disposição para se comunicar e interagir com o mundo exterior.

## 2.2 CLIENTE-SERVIDOR

Segundo Tarouco (1996), a tecnologia cliente-servidor é uma arquitetura na qual o processamento da informação é dividido em módulos ou processos distintos. Um processo é responsável pela manutenção da informação (servidores) e outros responsáveis pela obtenção dos dados (os clientes).

Já a aplicação cliente é quem estabelece a conexão com o servidor, envia mensagens para o mesmo e aguarda pelas mensagens de resposta. A figura 1 apresenta um exemplo de sistema cliente-servidor.

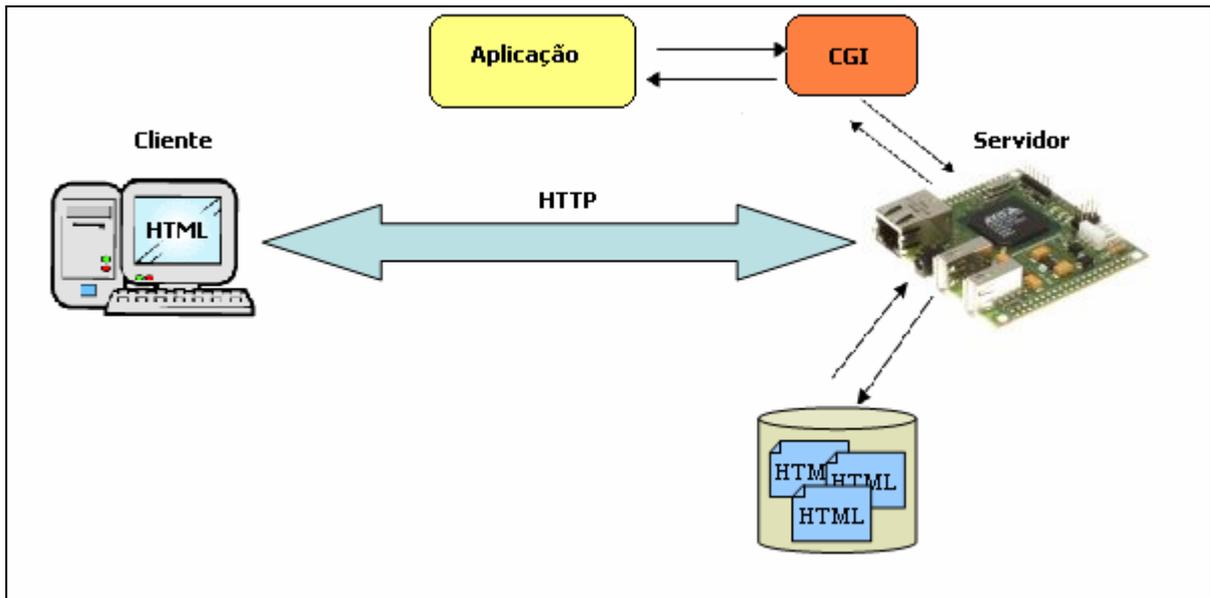


Figura 1 – Exemplo de sistema cliente-servidor

Os protocolos da camada de aplicação fornecem serviços do nível mais alto, tais como: servidores web, servidores de nomes, correio eletrônico, *login* remoto telnet e *Secure Shell* (SSH), transferência de arquivos, etc. O que essas aplicações têm em comum é o uso da arquitetura do tipo cliente-servidor, protocolo de sessão, cookies, etc.

### 2.3 LINUX

Segundo Campos (2006), o linux é um sistema operacional desenvolvido por Linus Torvalds a partir do código fonte do sistema Minix, que é uma versão simplificada do sistema operacional Unix, adotando uma licença livre onde todos os interessados podem usá-lo e retribuí-lo. Com o linux pode-se formar um ambiente moderno, seguro e estável, sendo hoje, um dos *kernels* de sistema operacional mais portados, rodando em *desktops*, servidores e sistemas embarcado.

O linux tem núcleo monolítico, possuindo funções tais como: agendamento de processos, gerenciamento de memória, operações de entrada e saída e acesso ao sistema de arquivos.

*Drivers* de dispositivo e extensões do núcleo tipicamente rodam com acesso total ao hardware, embora alguns rodem em espaço de usuário. Ao contrário dos núcleos monolíticos padrões, os *drivers* de dispositivo são facilmente configurados como módulos *Loadable Kernel Modules* (LKM), e carregados e descarregados enquanto o sistema está rodando.

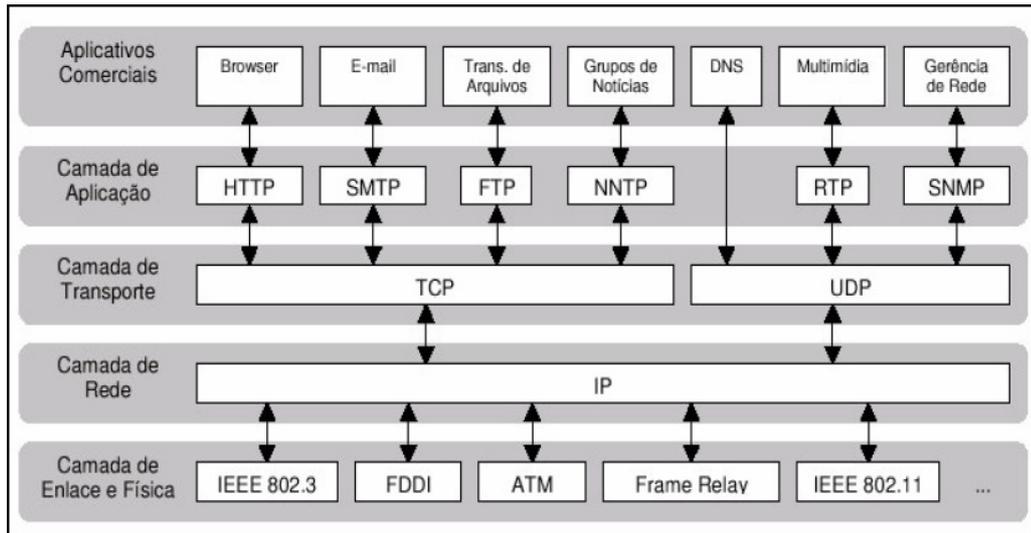
Também ao contrário de núcleos monolíticos padrões, *drivers* de dispositivo podem ser pré-inseridos sob certas condições. Essa última característica foi adicionada para corrigir o acesso a interrupções de hardware, e para melhorar o suporte a multiprocessamento simétrico.

## 2.4 PROTOCOLO HTTP

Segundo Hilgenstieler (2003, p. 19), o protocolo HTTP é um protocolo do nível de aplicação que possui objetividade e rapidez necessárias para suportar sistemas de informação distribuídos, cooperativos e de hipermídia. Suas principais características são as seguintes:

- a) propicia busca de informações e atualizações;
- b) envia mensagens em um formato similar aos utilizados pelo correio eletrônico da internet e pelo *Multipurpose Internet Mail Extensions* (MIME);
- c) possui comunicação entre os agentes usuários e *gateways*, permitindo acesso a hipermídia e a diversos protocolos do mundo internet;
- d) obedece ao paradigma de cliente-servidor, isto é, um cliente estabelece uma conexão com um servidor e envia um pedido, o qual o analisa e responde. A conexão deve ser estabelecida antes de cada pedido de cliente e encerrada após a resposta.

A figura 2 mostra uma visão geral das camadas implementadas pelo modelo de referência *Transmission Control Protocol/ Internet Protocol* (TCP/IP) e seus principais protocolos e aplicações.



Fonte: Péricas (2003, p. 37).

Figura 2 - Arquitetura da rede internet

Os principais métodos são:

- GET:** recupera todas as informações identificadas no recurso da rede. Se o recurso for um processo executável, ele retornará a resposta do processo e não o seu texto. Existe o GET condicional que traz o recurso apenas se o mesmo foi alterado depois da data da última transferência;
- HEAD:** semelhante ao método GET, só que neste caso não há a transferência da entidade para o cliente. Este método é utilizado para testar a validade e acessibilidade dos *links* de hipertexto;
- POST:** Envia dados para serem processados, por exemplo, dados de um formulário *HyperText Markup Language* (HTML) para o recurso especificado. Os dados são incluídos no corpo do comando;
- PUT:** coloca a entidade abaixo do recurso especificado no pedido. Se esta entidade não existe é criada. Se existe, apenas é atualizada;
- DELETE:** solicita que o servidor origem apague o recurso identificado no *Uniform Resource Identifier* (URI);
- LINK:** estabelece uma ou mais relações de *links* entre o recurso identificado pelo URI e outros recursos existentes, não permitindo que o corpo da entidade enviada seja subordinada ao recurso;
- UNLINK:** remove uma ou mais relações de links existentes entre o recurso identificado no URI.

## 2.5 CGI

Segundo Weinman (1997), o CGI é uma forma de comunicação que o servidor web emprega para enviar informações entre o navegador e um programa hospedado em um servidor web. Assim, designam-se por *scripts* CGI os pequenos programas que interpretam parâmetros e geram a página depois de processá-los, ou seja, são programas de computador rodando em um servidor web e que podem ser invocados a partir de uma página através do navegador. O quadro 5 mostra um trecho de uma criação de um CGI para acender e apagar um *Light Emitting Diode* (LED) na placa Fox Board utilizando linguagem C.

```
int main(int argc, char *argv[]) {
    char value[128];

    printf("Content-type: text/html\n\n");
    printf("<h1>Red led remote control</h1>");

    if (read_parameter("cmd",value,10)) {
        if (strstr("ledon",value)) gpioclearbits(PORTA, PA3);
        if (strstr("ledoff",value)) gpiosetbits(PORTA, PA3);
    }

    printf("<a href=?cmd=ledon?>ON</a>");
    printf(" | ");
    printf("<a href=?cmd=ledoff?>OFF</a>");

    return 0;
}
```

Fonte: Acme Systems (2006).

Quadro 5 - Trecho de código em C para CGI

## 2.6 FOX BOARD

Segundo Acme Systems (2006), a placa Fox Board possui como núcleo o sistema operacional Linux, com tamanho reduzido e baixo consumo, especialmente projetada para integrar facilmente a um projeto do cliente, com padrão *Local Area Network* (LAN) e protocolos da internet.

Possui dois *slots* de 20X2 pinos para conectar em uma placa específica desenvolvida pelo cliente, tornando um sistema Linux embarcado pronto para ser utilizado e acessado via

internet. O sistema operacional é livre e aberto para customização, ajustando-o para uma melhor integração com o sistema desenvolvido pelo cliente. Esta placa reduz os esforços do projeto para dispositivos que necessitam de acesso à internet.

Na figura 3 é apresentada a versão nacional placa Fox Board desenvolvida pelo professor e orientador Miguel Alexandre Wisintainer.

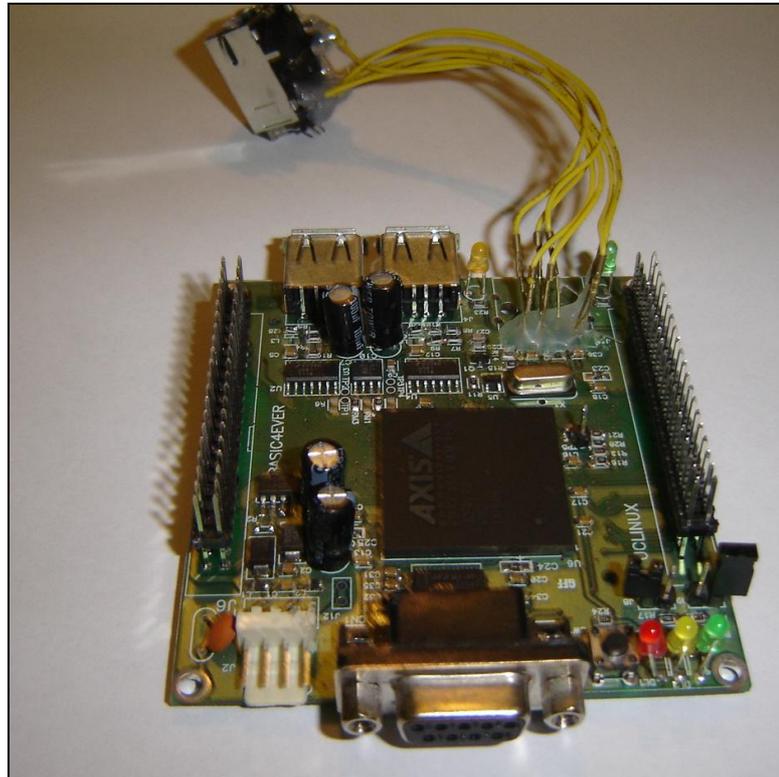


Figura 3 – Versão nacional da placa Fox Board

O quadro 1 descreve com mais detalhes as características de software da placa.

Núcleo	Linux versão 2.4.31 ( <i>default</i> ) or 2.6.15
Servidores	HTTP (WEB), FTP, SSH, TELNET
<i>Driver</i>	USB Pen driver, FTDI e PROLIFIC USB para conversor serial
SDK	<i>Open source</i> SDK para sistema Linux
Linguagens	C, C++, PHP, PYTHON, etc
Ferramentas	Compilador GNU C compiler, serviço disponível no <i>site</i> da ACME Systems

Fonte: adaptado de Acme Systems (2006).

Quadro 1 – Características de software da placa Fox Board

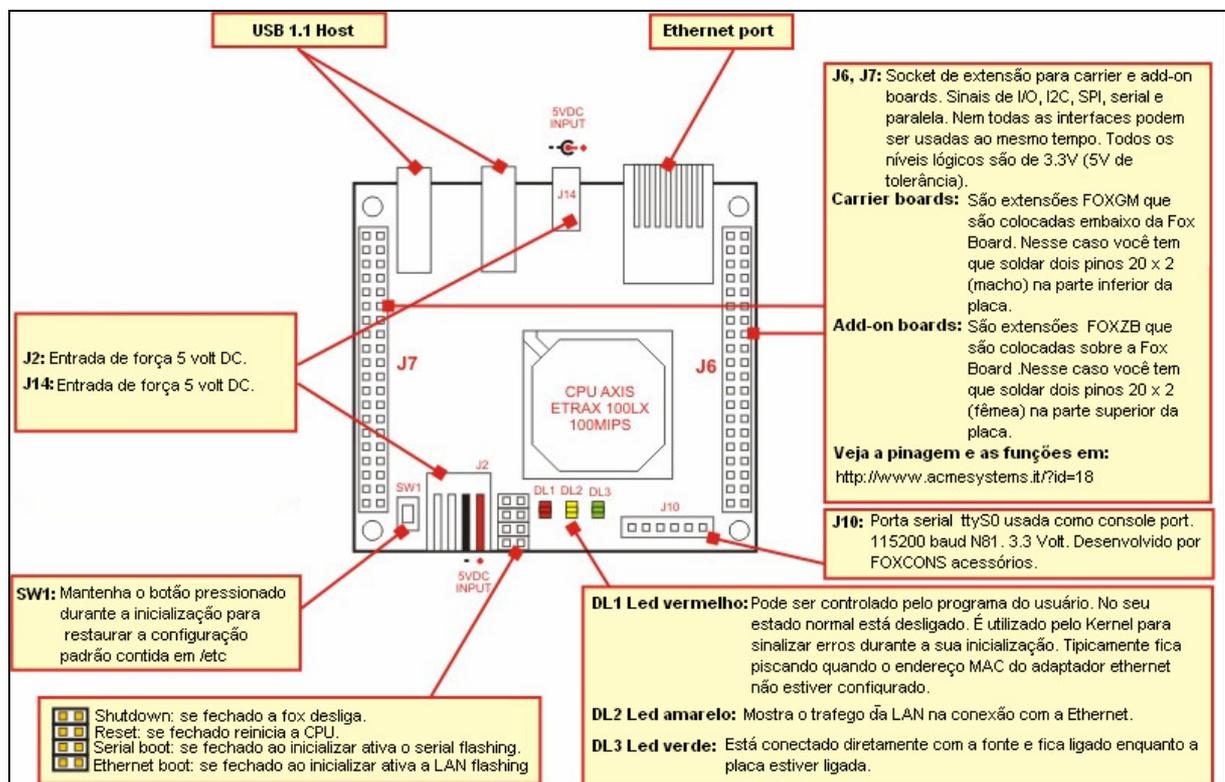
O quadro 2 descreve com mais detalhes as características de hardware da placa.

Tamanho	66 x 72 mm (2.6 x 2.8 polegadas)
CPU	Axis ETRAX 100LX 32 bit, RISC, 100MHz (100MIPS)
Memória	FOX MCM 4 + 16: 4MB FLASH 16MB (Random Access Memory) RAM
Alimentação	5 Volt 280mA (1 watt)
Portas	1 Ethernet (10/100 Mb/s) 2 USB 1.1 1 serial console port
Extensões	2 sockets IDE, SCSI, serial, paralela, linhas I/O, I2C bus interface
Peso	37 gramas
Faixa de temperatura	0-70 °C

Fonte: adaptado de Acme Systems (2006).

Quadro 2 – Características de hardware da placa Fox Board

Na figura 4 são apresentadas descrições dos principais componentes da placa Fox Board.



Fonte: adaptado de Acme Systems (2006).

Figura 4 – Descrições dos principais componentes da placa Fox Board

Nos quadros 3 e 4 são apresentadas as pinagens e suas respectivas funções na placa Fox Board.

J2				J10			
Pin #	I/O Line	Default function	Def. Dir	Pin #	I/O Line	Default function	Def. Dir
1		N.C.		1	+3.3V	Vcc (3.3V)	
2		GND GND		2	RTS0	Request to send /dev/ttyS0	O
3		GND GND		3	TXD0	Transmit data /dev/ttyS0	O
4		+5V Vcc (5V)		4	RXD0	Receive data /dev/ttyS0	I
				5	CTS0	Clear to send/dev/ttyS0	I
				6	GND	GND	

Fonte: Acme Systems (2006).

Quadro 3 – Pinagem e respectivas funções na Fox Board

J7				J6			
Pin #	I/O Line	Default function	Def. Dir	Pin #	I/O Line	Default function	Def. Dir
1		GND GND		1	+3.3V	Vcc (3.3V)	
2		GND GND		2	+3.3V	Vcc (3.3V)	
3	IOG22	IOG22 Input/Output line G22	I	3	OG31	RTS3 Request to send /dev/ttyS3	O
4	IOG23	IOG23 Input/Output line G23	I	4	IG30	RXD3 Receive data /dev/ttyS3	I
5	IOG20	IOG20 Input/Output line G20	I	5	OG30	TXD3 Transmit data /dev/ttyS3	O
6	IOG21	IOG21 Input/Output line G21	I	6	IG31	CTS3 Clear to send/dev/ttyS3	I
7	IOG18	IOG18 Input/Output line G18	I	7	OG7	RTS2 Request to send /dev/ttyS2	O
8	IOG19	IOG19 Input/Output line G19	I	8	IG6	RXD2 Receive data /dev/ttyS2	I
9	IOG16	IOG16 Input/Output line G16	I	9	OG6	TXD2 Transmit data /dev/ttyS2	O
10	IOG17	IOG17 Input/Output line G17	I	10	IG7	CTS2 Clear to send/dev/ttyS2	I
11	OG26	VMO line on USB2		11		NMI Non maskable Interrupt	I
12	OG29	OG29 Output line G29	O	12		+5V Vcc (5V)	
13	OG25	OG25 Output line G25	O	13	IOG9	IOG9 Input/Output line G9	I
14	IG25	VM line on USB2		14	IOG8	IOG8 Input/Output line G8	I
15	OG28	OG28 Output line G28	O	15	IOG11	IOG11 Input/Output line G11	I
16	OG27	VPO line on USB2		16	IOG10	IOG10 Input/Output line G10	I
17	IG28	SPEED line on USB2		17	IOG13	IOG13 Input/Output line G13	I
18	IG27	RCV line on USB2		18	IOG12	IOG12 Input/Output line G12	I
19	IG26	VP line on USB2		19	IOG15	IOG15 Input/Output line G15	I
20	IG29	OE line on USB2		20	IOG14	IOG14 Input/Output line G14	I
21	IOG24	IOG24 Input/Output line G24	I	21	OG2	OG2 Output line G2 (I2C_RESET)(1)	O
22	IOG0	IOG0 Input/Output line G0 (I2C_RES)(1)	O	22	OG5	OG5 Output line G5	O
23		SPEED line on USB1		23	OG1	OG1 Output line G1	O
24		RCV line on USB1		24	IG1	IG1 Input line G1	I
25		OE line on USB1		25	OG4	OG4 Output line G4	O
26		VP line on USB1		26	OG3	OG3 Output line G3	O
27		RESET Reset line (Out)	O	27	IG4	IG4 Input line G4	I
28		INTA INTA (Out)	O	28	IG3	IG3 Input line G3	I
29		+5V Vcc (5V)		29	IG2	IG2 Input line G2	I
30		IRQ Interrupt request line	I	30	IG5	IG5 Input line G5	I
31	PA7	DCD2 Data carrier detect /dev/ttyS2	I	31	PB1	I2C_CLK IC2 Clock	O
32	PA6	DSR2 Data set ready /dev/ttyS2	I	32	PB0	I2C_DATA IC2 Data	I
33	PA5	R12 Ring indicator /dev/ttyS2	I	33	PB3	VMO line on USB1	
34	PA4	DTR2 Data terminal ready /dev/ttyS2	O	34	PB2	VPO line on USB1	
35	PA3	DL1 Connected to DL1 Red led	O	35	PB5	VM line on USB1	
36	PA2	DL2 Connected to DL2 Yellow led	O	36	PB4	PB4 Input/Output line B4	I
37	PA1	SW1 Connected to SW1	I	37	PB7	PB7 Input/Output line B7	O
38	PA0	PA0 Input/Output line A0	I	38	PB6	PB6 Input/Output line B6	O
39		+3.3V Vcc (3.3V)		39		GND GND	
40		+3.3V Vcc (3.3V)		40		GND GND	

Fonte: Acme Systems (2006).

Quadro 4 – Pinagem e respectivas funções na Fox Board

## 2.7 SENSORES

Segundo Capelli e Braga (2002), sensores são dispositivos que detectam movimentos e ações que ocorrem nos processos e projetos de mecatrônica, sendo elementos dotados e encarregados de gerar informações para os sistemas de automação. Existem muitos tipos de sensores, cada um para uma atividade e aplicação específica.

Os principais tipos de sensores são:

- a) sensor magnético ou reed-switch: é um sensor que é acionado quando entra em contato com um campo magnético. Geralmente, é constituído de um material denominado ferro-magnético, ou seja, Ferro, Níquel e etc. O seu funcionamento ocorre da seguinte maneira, quando um ímã entra em contato com o sensor este atrai um par de chapas que fecha o circuito acionando uma determinada carga;
- b) sensor capacitivo: este tipo de sensor funciona seguindo os princípios de funcionamento do capacitor, como o próprio nome sugere. Ele se opõe às variações de tensões do circuito como o capacitor. O sensor capacitivo é constituído de duas chapas metálicas separadas por um material dielétrico que no caso é o ar, pois suas chapas são colocadas uma ao lado da outra diferente do capacitor que possui suas placas uma sobre a outra. O acionamento do sensor ocorre quando um corpo constituído de material não magnético se aproxima aumentando a sua capacitância. Quando isso ocorre, o circuito de controle detecta a variação na capacitância e aciona um determinado dispositivo. Geralmente, este tipo de sensor é utilizado para medir níveis de água ou para serem empregados em esteiras numa linha de produção;
- c) sensor indutivo: o sensor indutivo funciona seguindo os conceitos de funcionamento do indutor. O indutor é um componente eletrônico composto por um núcleo no qual está uma bobina em sua volta. Quando uma corrente percorre esta bobina um campo magnético é formado. Por sua vez, o campo magnético é concentrado no centro do núcleo fazendo com que se armazene energia por algum tempo. Ou seja, mesmo que a corrente pare de circular pela bobina ainda restará um certo tempo para cessar a corrente. O sensor indutivo utiliza este tipo de funcionamento para ser acionado e informar o sistema da presença de algum corpo. O núcleo do sensor indutivo é aberto e assim sendo o campo magnético passa pelo ar em uma intensidade menor. Porém quando um corpo metálico é

aproximado seu campo magnético passa pelo corpo aumentando sua intensidade acionado o circuito;

- d) sensor óptico: este tipo de sensor é constituído por dois componentes denominados, emissor de luz e receptor de luz. Geralmente, os emissores de luz são os famosos LEDs eletrônicos ou lâmpadas comuns. Já o receptor é um componente eletrônico foto-sensível tais como fototransistores ou fotodiodos. O funcionamento ocorre da seguinte maneira: uma onda é gerada por um circuito oscilador e esta é convertida em luz pelo emissor. Quando um corpo se aproxima este reflete a luz do emissor para o receptor acionando o circuito de controle;
- e) sensor de pressão ou chave fim de curso: este tipo de sensor está presente em muito dispositivos mecânicos e pneumáticos. Estes sensores são utilizados para detectar o fim de um curso de um determinado dispositivo. Estes dispositivos podem ser atuadores mecânicos tais como cilindros e alavancas. Seu funcionamento mostra-se muito simples, pois seu acionamento é totalmente mecânico. Este sensor pode ser Normalmente Aberto (NA) ou Normalmente Fechado (NF);
- f) *encoder*: são dispositivos eletromecânicos que convertem posicionamentos mecânicos em sinais eletrônicos digitais para o sistema de controle. Este equipamento é capaz de medir deslocamentos angulares e lineares em máquinas e robôs. Este tipo de sensor utiliza o princípio de funcionamento dos *mouses* dos computadores, ou seja, processo opto-eletrônico.
- g) sensores CCD: Basicamente *charge coupled device* (CCD) é um circuito eletrônico constituído de milhões de sensores microscópicos sensíveis à luz. A quantidade destes sensores expressa a resolução, ou seja quanto mais sensores, maior é a resolução da imagem. Cada um desses sensores representa um *pixel* e como ele não distingue cores, somente intensidade de luz, sobre eles são colocados filtros de cores básicas, que são azul, verde e vermelho (POVARESKIM, 2008).

## 2.8 TRABALHOS CORRELATOS

Algumas ferramentas desempenham papel semelhante ao proposto no presente trabalho, cada qual com suas peculiaridades na área de automação e domótica, os quais são:

- a) sistema para automação e controle residencial via e-mail: trabalho desenvolvido por Censi (2001), que consiste em um sistema de automação residencial que executa tarefas através de comandos recebidos por *e-mail*, bem como comandos digitados localmente. Utiliza os protocolos *Simple Mail Transfer Protocol* (SMTP) e *Post Office Protocol* (POP);
- b) protótipo de sistema de monitoramento remoto utilizando TCP/IP sobre Ethernet (802.3): trabalho desenvolvido por Montibeller Junior (2005), no qual se aplica uma alternativa em outra plataforma sobre o dispositivo de vigilância, integrando também câmera e sensores, onde o mesmo se conecta a rede TCP/IP, tornando assim possível monitorar ambientes em qualquer lugar do mundo. Foi implementado um hardware que gerencia as câmeras e sensores e um servidor implementado em Java para exibir os dados ao cliente;
- c) Robotbuilder: desenvolvido por Mammana e Cardoche (2007), sendo um robô sobre uma plataforma de carrinho de brinquedo, no qual se faz processamento de imagem e segue objetos. Utiliza servos-motores, câmera e uma rede de conexão sem fio, montados sobre a placa Fox Board, no desenvolvimento foram utilizadas as linguagens C e Java.

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo, são detalhadas a especificação e implementação do protótipo, através de diagramas e trechos do código fonte implementado.

#### 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O trabalho será composto de hardware e software. Abaixo são detalhados os requisitos funcionais (RF) e os requisitos não funcionais (RNF).

- a) monitorar os estados dos sensores (RF);
- b) executar o servidor web (RF);
- c) utilizar o sistema operacional Linux (RNF);
- d) possibilitar a visualização das imagens captadas pela câmera e sinais dos sensores (RF);
- e) receber os dados enviados pelo cliente através da rede (RF);
- f) enviar dados ao cliente através da rede (RF);
- g) gerenciar o dispositivo (RF);
- h) criação de CGI utilizando a linguagem de programação C e código HTML (RNF);
- i) a aplicação web do cliente deve ser independente de sistema operacional (RNF).

#### 3.2 ESPECIFICAÇÃO

O Protótipo é composto de um hardware no qual são acoplados os sensores, câmera, alarme e relês para acionarem tomadas e também um software que gerencia os mesmos. Para realizar este trabalho se faz necessário dividir sua especificação em duas partes. São elas especificação do hardware e do software embarcado na placa Fox Board.

### 3.2.1 Especificação do hardware

O hardware tem a função de coletar informações dos sensores e câmera para enviar ao software embarcado que apresenta as informações capturadas e em qualquer navegador, como por exemplo, Internet Explorer ou Mozilla Firefox.

O dispositivo é constituído de cinco partes fundamentais: placa Fox Board, câmera, sensores, circuito de potência e tomadas, figura 5.

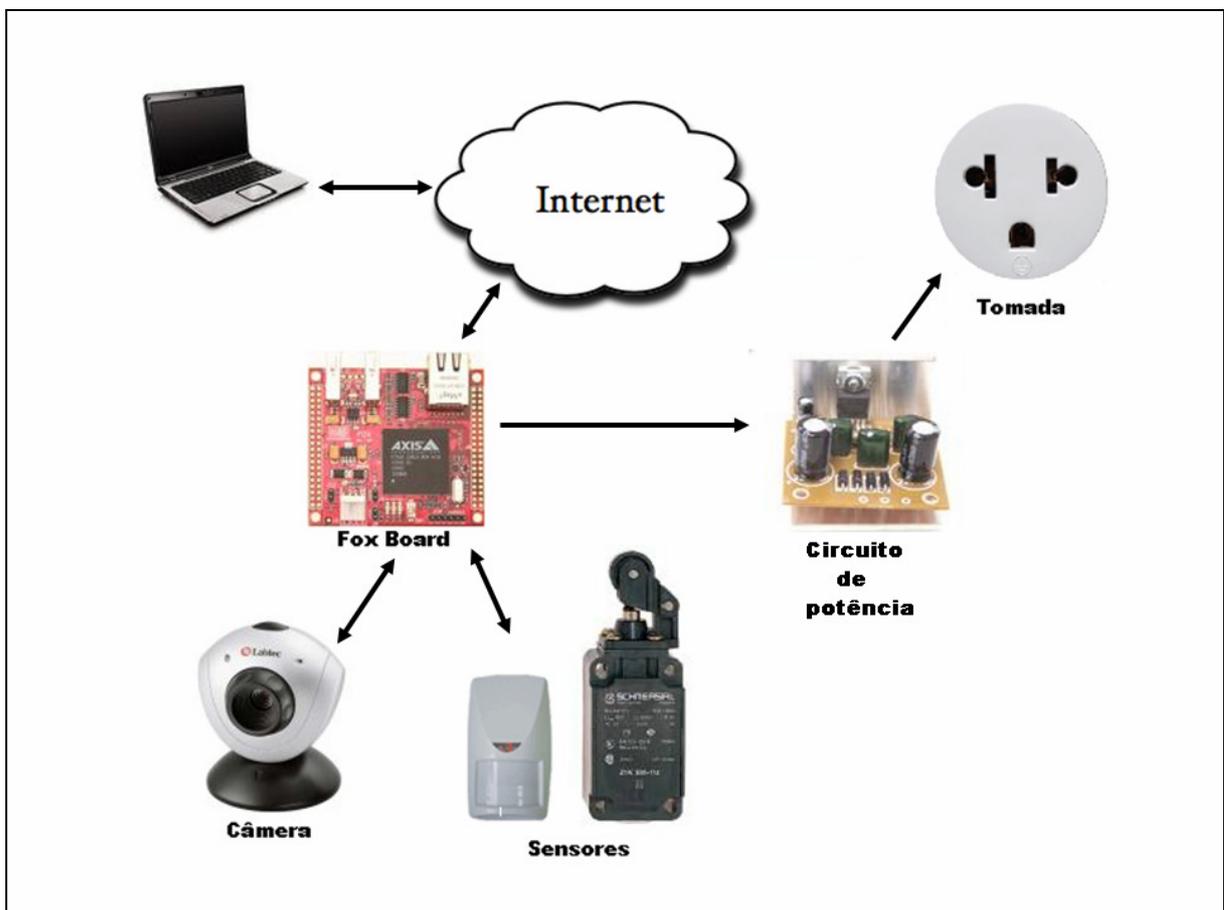


Figura 5 – Esquemático do hardware.

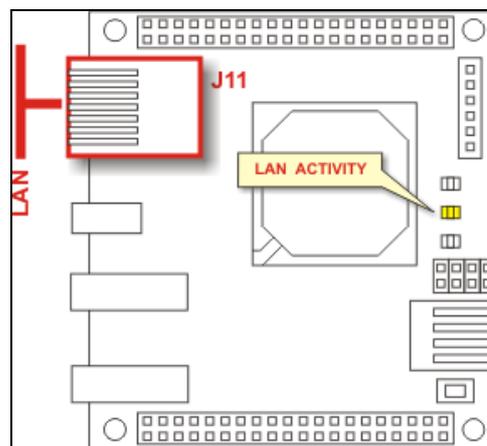
A descrição dos componentes da figura 5 dá-se da seguinte forma:

- placa Fox Board: é responsável por gerenciar a interface com a rede TCP/IP, acionamento das entradas e saídas, leitura dos dados coletados pelos sensores e câmera;
- câmera: conectado a Fox Board através de USB realiza a captura de imagens, o modelo utilizado é a WebCam Pro Labtec, que tem uma resolução de vídeo de até

640 x 480 *pixels* e resolução de imagem de até 1280 x 960 *pixels*, possui uma taxa de até 30 quadros por segundo;

- c) sensores: conectado a Fox Board através dos pinos de uso geral (*I/O lines*), são responsáveis por fazer leituras do ambiente, alterando seu nível lógico quando o mesmo muda de estado;
- d) circuito de potência: conectado a Fox Board através dos pinos de uso geral (*I/O lines*), é responsável por capturar o sinal dos pinos da J6 e por intermédio dos transistores acionar relês ;
- e) tomada: conectada ao circuito de potência, fornece energia a qualquer eletro-eletrônico e/ou eletrodoméstico.

A Fox Board tem um conector RJ45 fêmea para conexão de rede Ethernet 10/100Mbit (J11), o led amarelo apresenta o tráfego pela rede, conforme apresentado na figura 6. É possível ligar diretamente em um computador através de um cabo Ethernet *crossover* ou um *hub* normal com cabo de LAN.



Fonte: Acme Systems (2006).

Figura 6 – Conexão com a Rede.

Os circuitos eletrônicos foram especificados através da ferramenta Proteus Profissional. Na figura 7 é apresentado o esquemático do circuito de sensores, no qual cada sensor é ligado a uma entrada da J6 na Fox Board.

A figura 8 mostra esquemático do circuito de tomadas e alarme, no qual cada relê e o alarme são acionados ou desacionados por intermédio de um transistor ligado nas saídas da J6 na Fox Board, podendo assim ligar eletrodomésticos ou eletrônicos nos relês.

Na figura 9, são apresentados os pinos da J6 que são utilizados como entradas e saídas para sensores e tomadas.

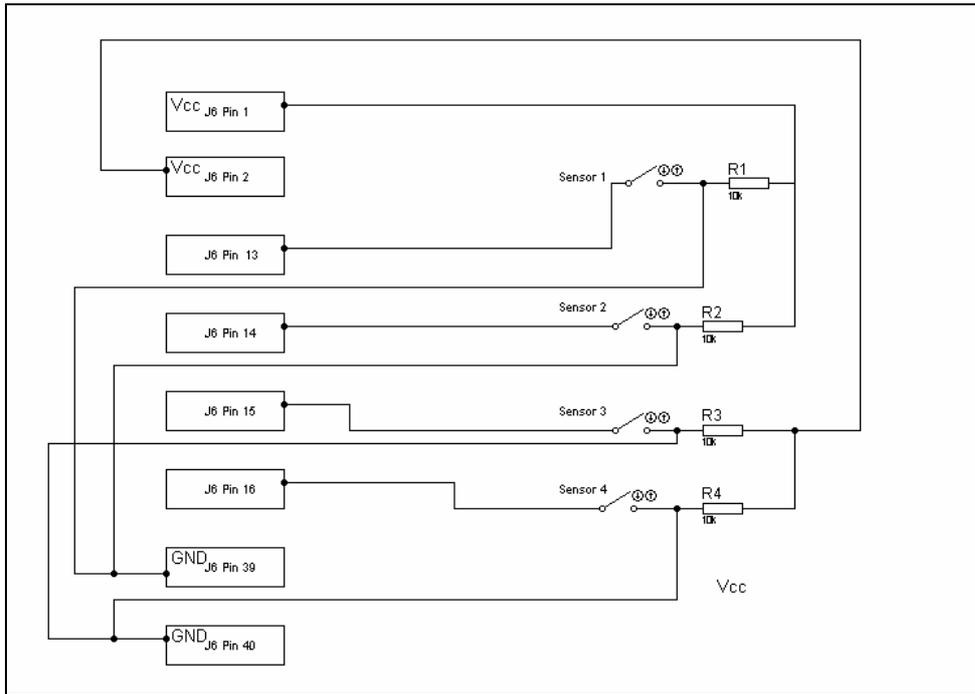


Figura 7 – Esquemático do circuito de sensores

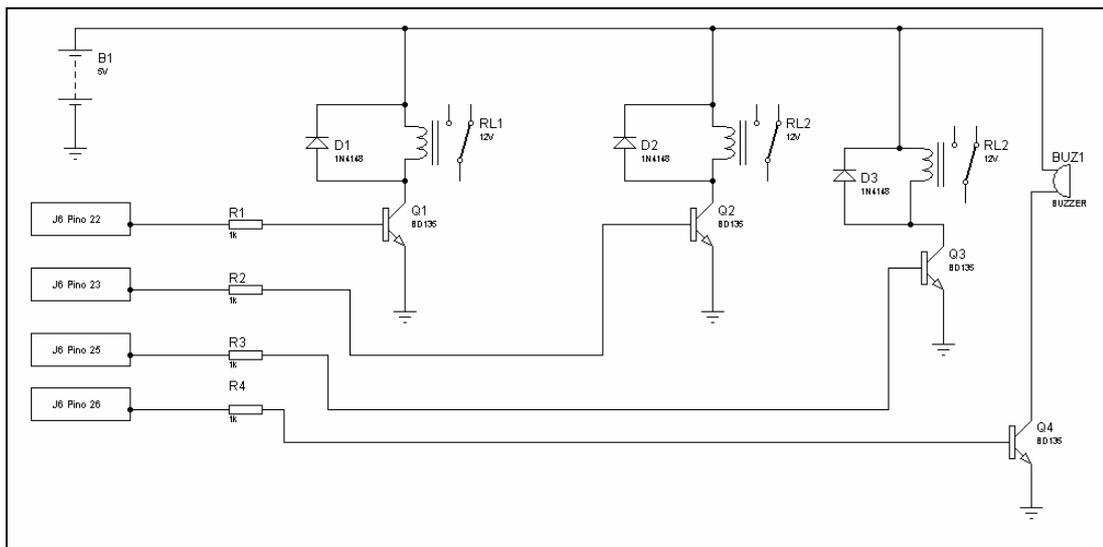


Figura 8 – Esquemático do circuito tomadas e alarme

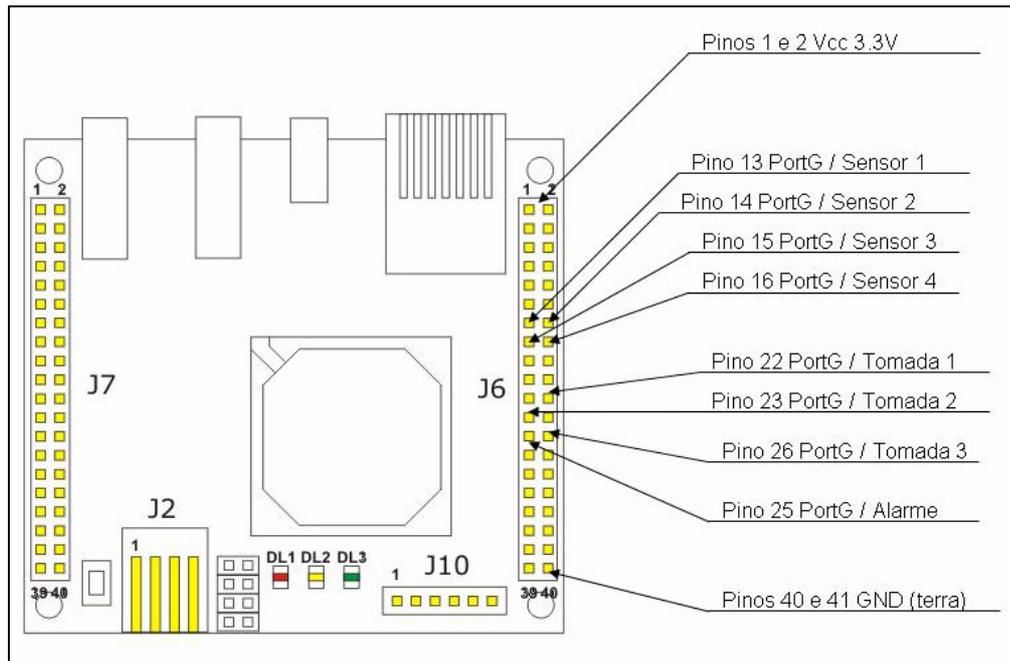


Figura 9 – Pinos J6 utilizados como I/O

A seguir são apresentadas as principais partes do hardware do protótipo, na figura 10 é mostrado na área de cobertura da cor verde o circuito de potência com relês que acionam e desacionam tomadas comandadas pelos transistores ligados nos pinos 22, 23 e 26 da J6 na Fox Board.

Na área de cobertura da cor vermelha é mostrado o circuito de alarme também comandado por um transistor ligado no pino 25 da J6 na Fox Board. Por fim na área de cobertura da cor azul é apresentado o circuito que recebe o sinal dos sensores ligado a resistores que por sua vez são ligados nos pinos 13, 14, 15 e 16 da J6 na Fox Board.

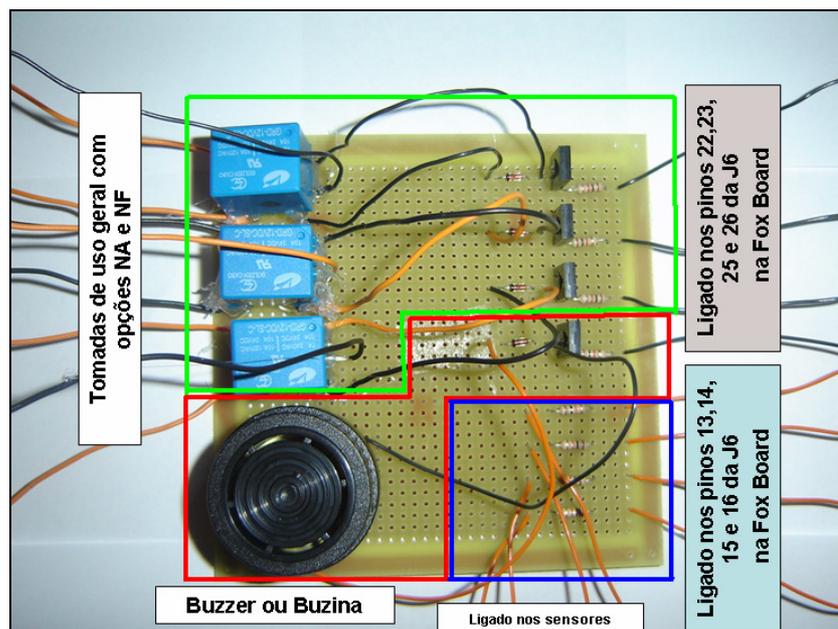


Figura 10 – Circuitos de tomadas, sensores e alarme

Na figura 11 é apresentado a placa Fox Board nacional conectada a rede, câmera e circuito de tomadas e sensores.



Figura 11 – Conexão de dispositivos a Fox Board nacional

### 3.2.2 Especificação do software

A especificação do software foi feita através da ferramenta Enterprise Architect, utilizando diagramas da Unified Modeling Language (UML), gerando o diagrama de atividades. Para facilitar o entendimento foi utilizado a *web application extension* (WAE), uma extensão da UML para aplicações web (CONALLEN, 2003). A WAE permite dar uma idéia dos principais componentes utilizados na aplicação web assim como sua navegabilidade.

É apresentada na figura 12 a interação do programa responsável por leitura e registro de acionamento dos sensores, este programa roda em segundo plano podendo ser inicializado no momento do boot.

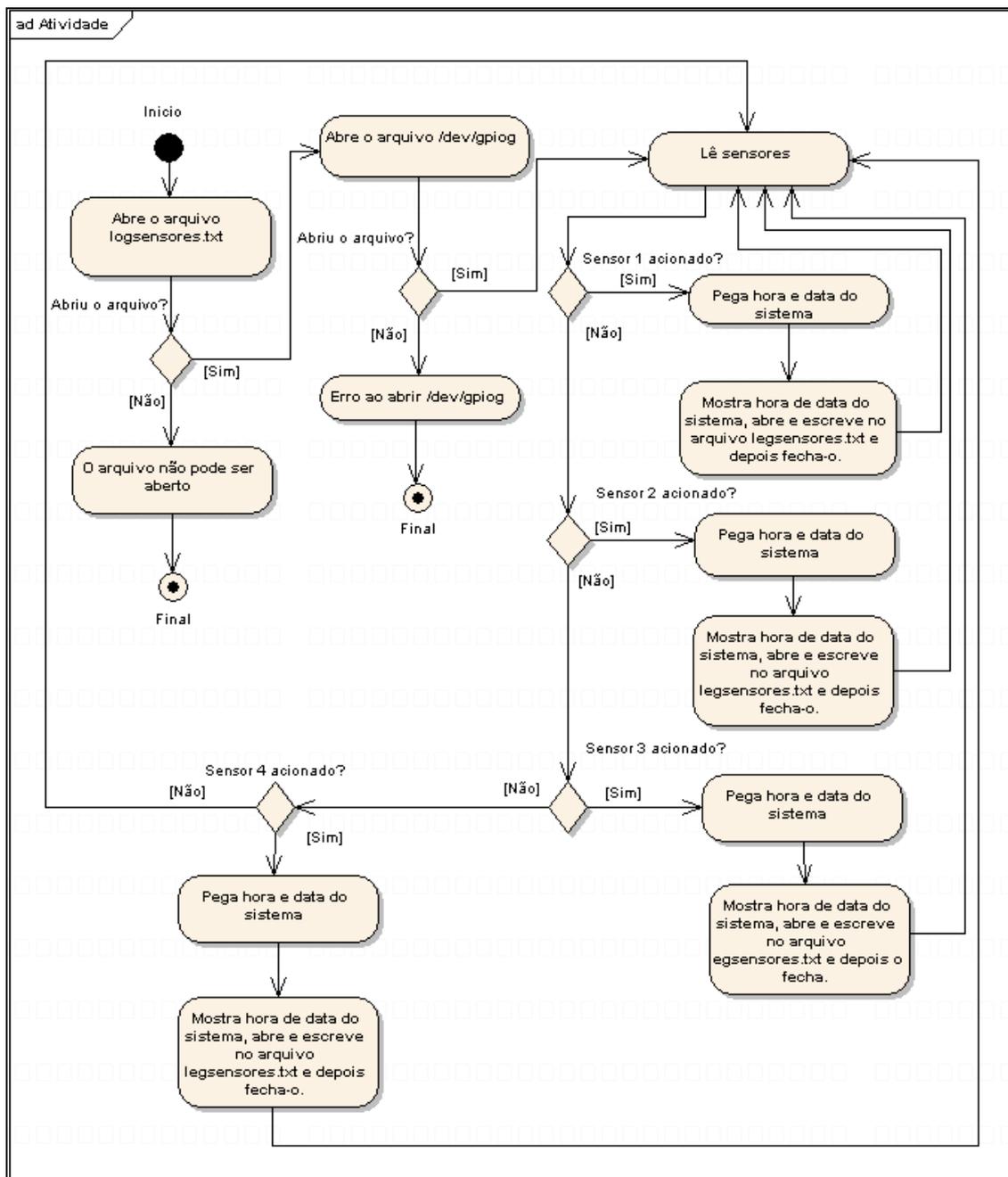


Figura 12 – Diagrama de atividades da programação do leitor de sensores

A descrição mais detalhada do diagrama de atividades do leitor de sensores é apresentada no quadro 6.

Atividade	Descrição
Abre o arquivo logsensores.txt	Abre o arquivo logsensores.txt para registrar data, hora e qual sensor foi acionado em determinado momento.
O arquivo não pode ser aberto	O programa mostra a mensagem que o arquivo não pode ser aberto e o programa é encerrado.
Abre o arquivo /dev/gpiog	Este arquivo possui o método para fazer uso da Port G da placa Fox Board.
Erro ao abrir /dev/gpiog	Caso o arquivo não possa ser aberto ele apresenta a mensagem, erro ao abrir /dev/gpiog e o programa é encerrado.
Lê sensores	O programa fica em <i>loop</i> esperando que algum sensor seja acionado
Pega hora e data do sistema	Quando algum dos sensores for acionado então o programa pega data e hora. Fazendo isso com cada um dos sensores
Mostra hora e data do sistema, abre e escreve no arquivo logsensores.txt e depois o fecha.	O programa imprime na tela o sensor data e hora que o mesmo foi acionado e grava a mesma informação no arquivo logsensores.txt e depois o fecha. Fazendo isso com cada um dos sensores

Quadro 6 – Descrição das atividades do leitor de sensores

Na figura 14 é apresentado o diagrama de arquitetura lógica do protótipo também conhecido como diagrama navegacional segundo a WAE, tendo como página inicial o arquivo `tcc.html` possuindo ligações com **cam.html**, **rspca**, **logsensores.txt** e **tomadas.c**.

O programa **rspca** é um programa específico da Fox Board escrito utilizando CGI, responsável por inicializar e finalizar os *drivers* da câmera, bem como definir dois tipos diferentes de resolução.

A página **cam.html** carrega o arquivo **JWebcamPlayer.jar** responsável pela visualização da imagem da câmera na placa Fox Board e também possui ligações com **tcc.html**, **rspca**, **logsensores.txt** e **tomadas.out**.

**Tomadas.c** é um programa feito em CGI responsável por ativar ou desativar as tomadas através de uma página da internet. Já o programa **lesensores.c** faz uma varredura nas portas de entrada e saída da Fox Board registrando o momento e qual sensor foi acionado, conforme figura 13.

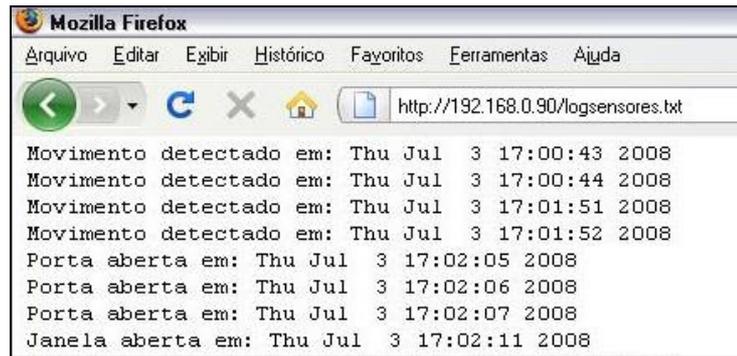


Figura 13 – Dados registrados pelo arquivo logsensores.txt

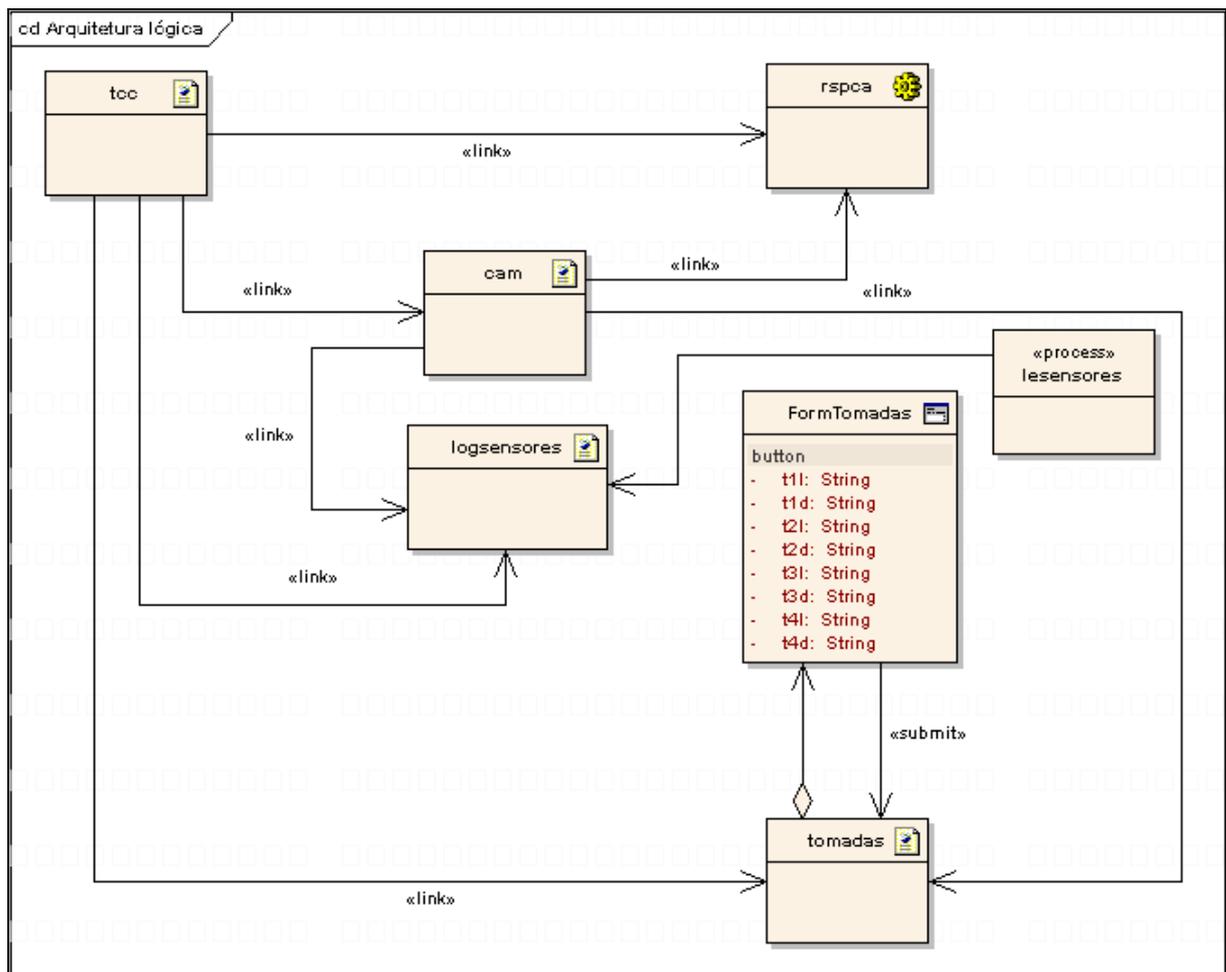


Figura 14 – Diagrama de arquitetura lógica do sistema

### 3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas no desenvolvimento deste protótipo, serão apresentados trechos do código fonte desenvolvido. Também é demonstrada a operacionalidade da implementação, através de um estudo de caso.

### 3.3.1 Técnicas e ferramentas utilizadas

A instalação do sistema operacional linux na placa Fox Board é feita através da rede utilizando Windows XP, tendo como requisitos os softwares WinPcap 4.0 e flashFox.exe apresentado na figura 15, destacando os comandos necessários para gravação da imagem do linux na placa.

```
D:\ACHE\WORK>flashFOX -d 2 -F -i my_image
```

```

*****
e100boot made by axis se, ported for windows by acmesystems.it
feedback, bugs : info@acmesystems.it
*****

Using device Broadcom NetXtreme Gigabit Ethernet Driver
Using internal boot loader INTERNAL_NW - Network boot (default).
Starting boot...
We're doing a flash write, this may take up to a few minutes...

Device ID = 0x00001b49
This bootloader was built by blogic on Mon Mar 5 10:29:02 CET 2007.
Checksum of bootloader is 0x000a0ac0
Waiting for load info.
Checksum of file is 0x00001ebd
Got load info.
SET_REGISTER
0xb0000000
0x000095f8
SET_REGISTER
0xb0000004
0x00000004
SET_REGISTER
0xb000000c
0x09603737
SET_REGISTER
0xb0000008
...

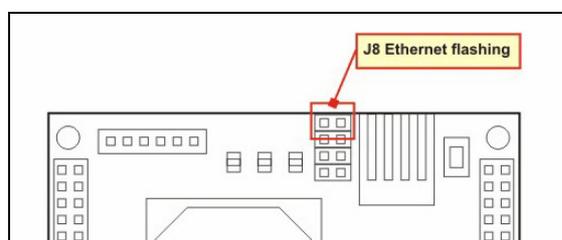
```

→ Caminho e nome da imagem do linux  
→ Enviar toda a imagem do linux  
→ Dispositivo de rede escolhido para enviar a imagem do linux para Fox Board.

Fonte: adaptado de Acme Systems (2006).

Figura 15 – Software flashFox destacando os comandos pra instalação da imagem do linux

Antes de iniciar a instalação é necessário colocar um *jumper* na J8 da placa Fox Board (figura 16) e reiniciá-la.



Fonte: Acme Systems (2006).

Figura 16 – Localização da J8 na placa Fox Board

Na implementação do software foram utilizadas as linguagens C, CGI compiladas através do *webcompiler*, disponível no site do fabricante Fox Board (ACME SYSTEMS, 2006), mostrado na figura 17.

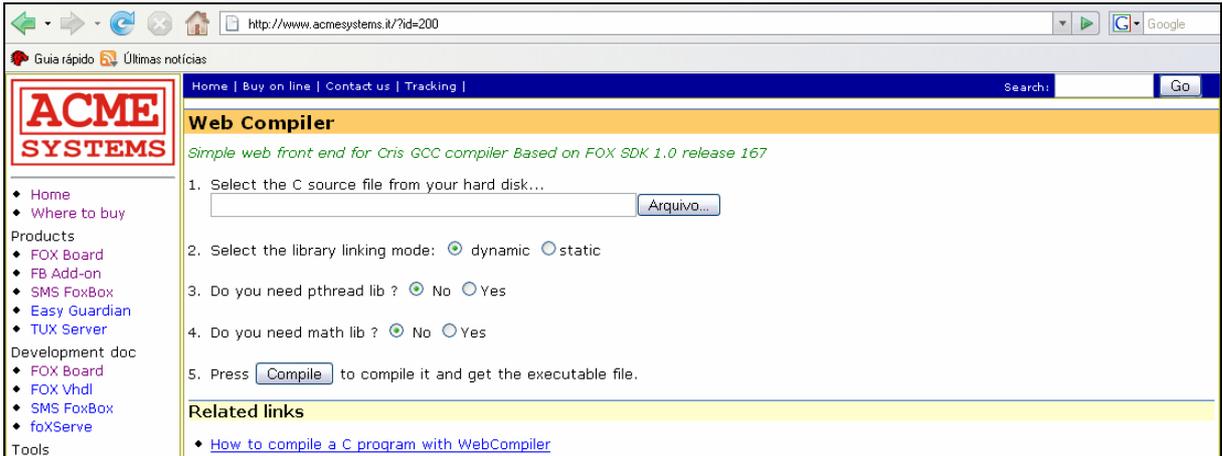


Figura 17 – Webcompiler da Acme Systems

Para o servidor web funcionar é necessário configurar o arquivo **boa.conf**, encontrado em **/etc/httpd/conf/**, destacando as linhas:

- a) 8 *DocumentRoot /usr/html*: o parâmetro *DocumentRoot* define a pasta pública usada pelo servidor web;
- b) 26 *ScriptAlias /cgi/ /etc/httpd/cgi/*: o parâmetro *ScriptAlias* define a pasta que estarão os scripts CGI. Se a pasta **/etc/httpd/cgi/** não existe ela deve ser criada.

Após feita a configuração do servidor web o mesmo deve ser reiniciado utilizando o comando **/etc/init.d/httpd restart**.

Os arquivos C e CGI compilados e os arquivos HTML são enviados para Fox Board através do protocolo *File Transfer Protocol* (FTP) (figura 18).

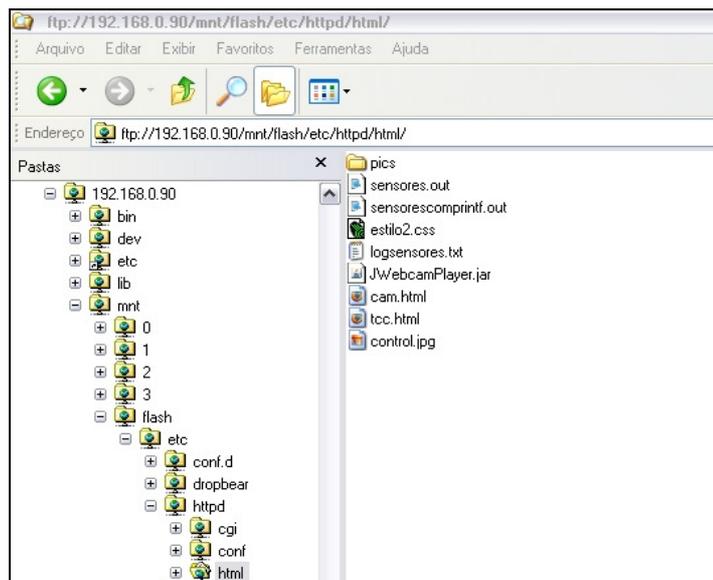


Figura 18 – Acessando placa Fox Board utilizando protocolo FTP

No desenvolvimento do software foram utilizados os seguintes recursos da placa Fox

Board:

- a) entradas e saídas da PortG;
- b) utilização dos protocolos HTTP, FTP e Telnet;
- c) servidor web;
- d) USB;
- e) gravação de informações na memória FLASH, pois este tipo de memória permite armazenar dados por longos períodos sem precisar de alimentação elétrica.

Todas as informações de data de hora de acionamento dos sensores são salvas na memória flash da Fox Board em um arquivo chamado logsensores.txt, conforme foi apresentado na figura 13, e esta organizado da seguinte forma: identificação do sensor, dia da semana, mês, dia, hora, minuto segundo e ano. Cada vez que um dos sensores é acionado ele grava estas informações no arquivo. O quadro 7 apresenta como é aberto o método que aciona as entradas e saídas da PortG e também como são setados os pinos utilizados no programa lesensores.c que roda em segundo plano.

```
//Abrindo a /dev/gpiog
if ((fd = open("/dev/gpiog", O_RDWR)) < 0) {
    printf("Erro ao abrir /dev/gpiog\n");
    exit(0);
}
//Seta a linha 9 da PortG pin # 13
ioma sk1=1<<9;
//Seta a linha 8 da PortG pin # 14
ioma sk2=1<<8;
//Seta a linha 11 da PortG pin # 15
ioma sk3=1<<11;
//Seta a linha 9 da PortG pin # 16
ioma sk4=1<<10;
```

Quadro 7 – Trecho de código que abre a PortG e configura os pinos de I/O

No quadro 8 é mostrado outro trecho de código do programa lesensores, quando o sensor 1 é acionado ele pega os dados de data e hora e salva no arquivo logsensores.txt. Destacando a função `value=ioctl(fd, _IO(ETRAXGPIO_IOCTLTYPE, IO_READBITS))`, que é um *bitmask* que retorna o estado atual da linha de entrada.

```

//lendo a entrada
value=ioctl(fd, _IO(ETRAXGPIO_IOCTLTYPE, IO_READBITS));
char *aux[30];
//se o sensor 1 for acionado
if ((value&iomask1)==0) {
    //pega a data e hora do sistema
    ptr = localtime(&lt);
    //aux recebe a data e hora
    sprintf(aux, asctime(ptr));
    printf("Sensor 1 acionado em: ");
    printf(aux);
    //escreve o conteudo da variavel source no arquivo
    fputs("Sensor 1: ", source);
    fputs(aux, source);
    lt = time (NULL);
    //fecha arquivo
    fclose(source);
    abrearquivo();
}

```

Quadro 8 – Trecho de código que grava dados dos sensores caso sejam acionados

O programa tomadas.out foi desenvolvido em CGI, e é responsável por submeter as requisições para a Fox Board quando o usuário clicar em um dos botões da página, que por sua vez aciona ou desaciona os pinos de saídas. No quadro 9 é mostrado um trecho do código do programa.

```

void commandExecute(char *query_string) {
    char *p;
    //Tomada 1
    if ((p=strstr(query_string, "Tomada+1+Ligar")) {
        //printf("Liga tomada 1\n");
        int iomask;
        iomask=1<<5;
        ioctl(fd, _IO(ETRAXGPIO_IOCTLTYPE, IO_SETBITS), iomask);
        return;
    }
    if ((p=strstr(query_string, "Tomada+1+Desligar")) {
        //printf("Desligar tomada 1\n");
        int iomask;
        iomask=1<<5;
        ioctl(fd, _IO(ETRAXGPIO_IOCTLTYPE, IO_CLRBITS), iomask);
        return;
    }
}

```

Quadro 9 – Trecho de código que aciona ou desaciona a tomada 1

No quadro 10 é mostrado um trecho de código da página tcc.html destacando as ligações para as páginas logsensores.txt e tomadas.out.

```

<a href="logsensores.txt" target="_blank">Log sensores
<span>
Registros de data e hora
que os sensores foram acionados.
</span>
</a>
</li>

<li>
<a href="/cgi/tomada.out" target="_blank"> Tomadas
<span>
Ligar ou desligar tomadas
remotamente.
</span>
</a>
</li>
</ul>

```

Quadro 10 – Trecho de código da página tcc.html

### 3.3.2 Operacionalidade da implementação

Esta seção descreve o funcionamento do protótipo através de um estudo de caso. Para utilizar o protótipo será necessário ligá-lo a uma fonte de energia e a internet através de um conector RJ45. Ao ligar o protótipo o usuário acessa a página inicial do protótipo, por exemplo, 192.168.0.90/tcc.html, conforme figura 19.

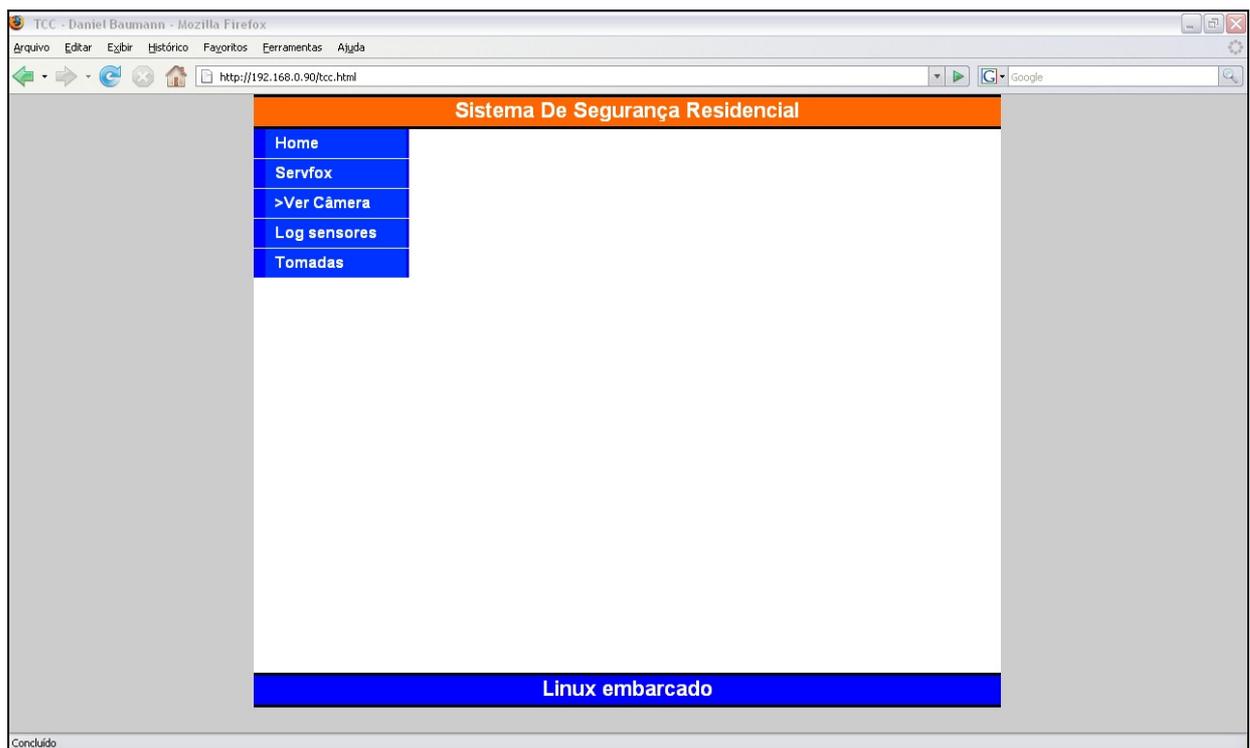


Figura 19 – Tela inicial do protótipo

Para visualizar a câmera primeiro o usuário precisa iniciar o servidor de *streaming* Servfox clicando no menu a opção Servfox, então será aberto uma aba com as opções de resolução e inicialização ou finalização do servidor (caso já esteja inicializado). A página da Servfox é apresentada na figura 20.

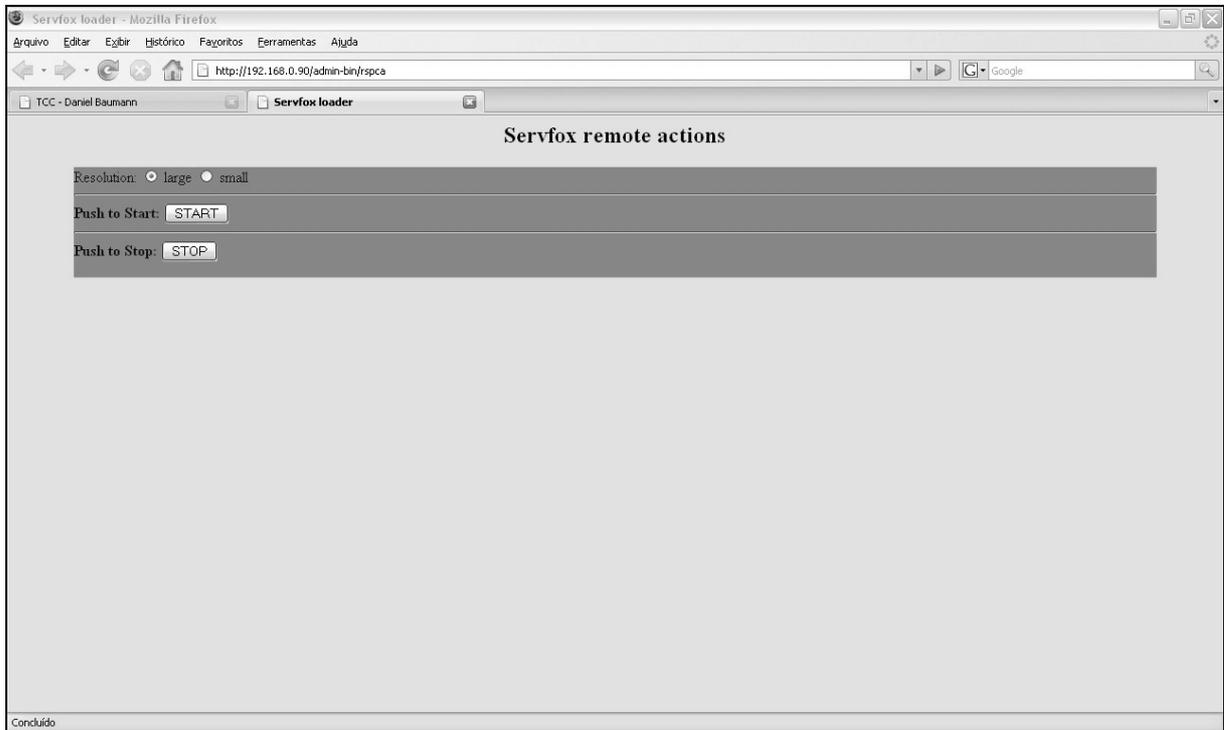


Figura 20 – Tela do servidor de *streaming* Servfox

Após inicializado a Servfox o usuário pode fechar a aba e voltar para página anterior. Clicando em Ver Câmera, no menu, será feito o redirecionamento para a página `cam.html` onde serão exibidas as imagens da câmera, figura 21.



Figura 21 – Tela de visualização de câmera

Caso o usuário deseje visualizar quais sensores foram acionados em determinado momento basta clicar no menu em Log sensores, na figura 22 é destacado no retângulo vermelho o log dos sensores.

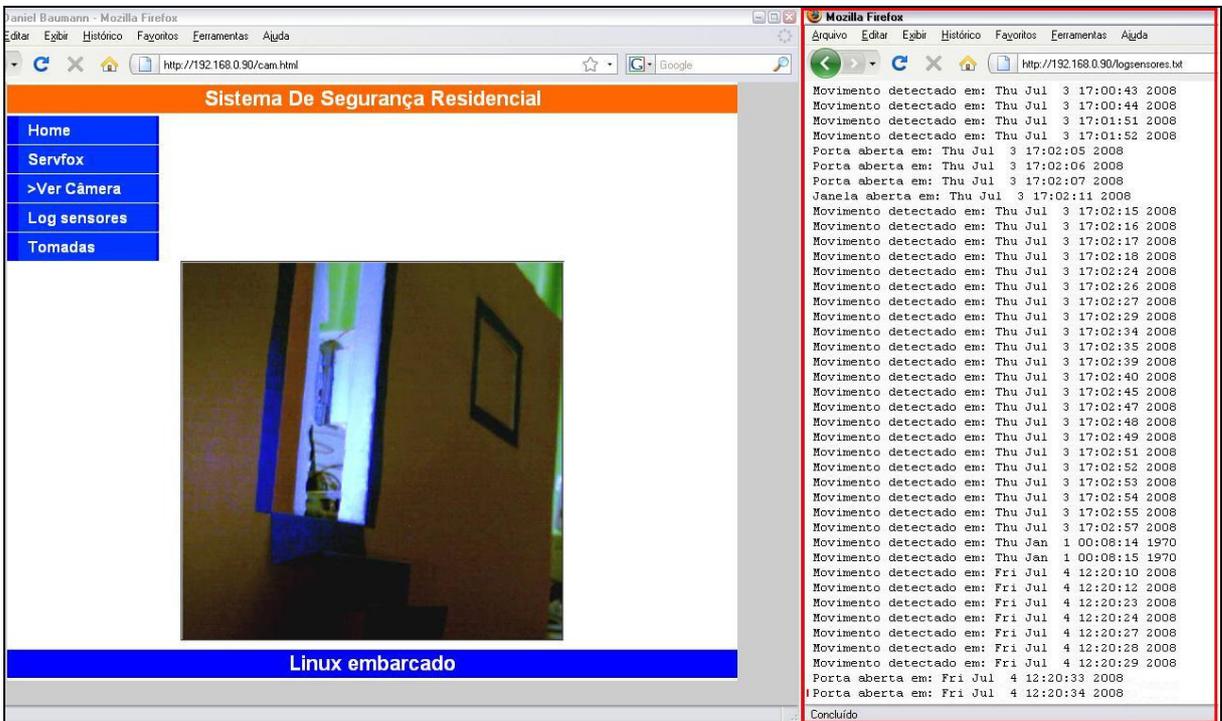


Figura 22 – Tela de visualização do log de sensores

Por fim se o usuário desejar acionar ou desacionar tomadas, luzes ou alarme deve

clicar no menu em Tomadas, o que possibilita ligar eletrodomésticos, luzes ou alarmes destacando os botões de apagar e acender luzes, figura 23.

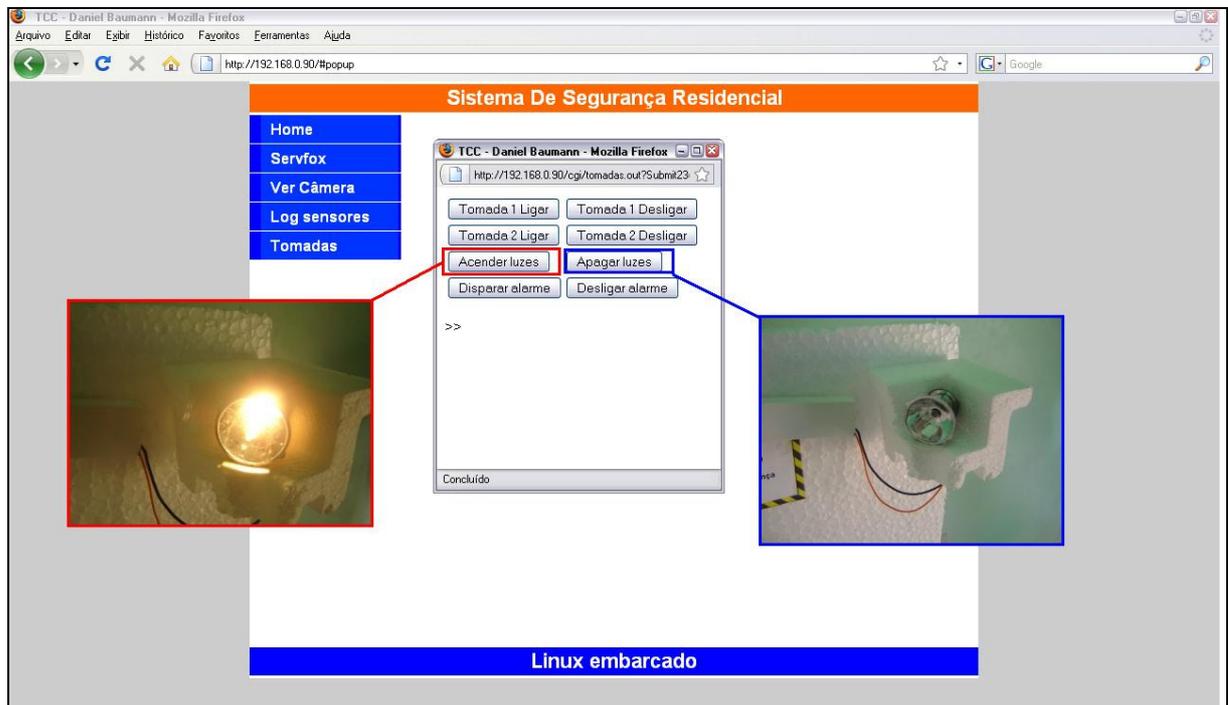


Figura 23 – Tela de controle das tomadas, iluminação e alarme

### 3.4 RESULTADOS E DISCUSSÃO

Considerando-se a proposta do protótipo, os resultados obtidos foram bons. A Fox Board apresentou ótimo desempenho, tornando possível desenvolver um trabalho com Linux embarcado, LAN, USB e protocolos da internet, citando também o servidor web facilitando o desenvolvimento do protótipo.

A câmera ligada a Fox Board apresentou um resultado satisfatório em relação ao tempo de resposta (tempo inferior a um segundo, podendo variar com tráfego na rede) desde o momento em que captura uma imagem até a mesma ser mostrada no navegador. O tempo de resposta do momento que é pressionado um botão no navegador para acionar ou desacionar tomadas também foi inferior a um segundo.

Apesar da limitação de 4MB FLASH e 16MB RAM do modelo de Fox Board utilizado não foi encontrado problema algum de falta de espaço, já que foram utilizados apenas códigos fonte em C e HTML e compilador desenvolvido pela Acme Systems, contudo se o equipamento ficar 24 horas ligado registrando *log* de sensores o espaço na memória *flash* acabará. No decorrer do trabalho foram encontradas algumas dificuldades, sendo a maior delas encontrar uma imagem do Linux para Fox Board MCM que possuísse *driver* de câmera e que acionasse as entradas e saídas. Geralmente estavam disponíveis duas versões de imagem do Linux sendo que cada uma possuía apenas uma funcionalidade. A princípio foi tentado com insucesso instalar o *driver* de câmera, mas a versão do *kernel* era tão básica que não oferecia suporte para instalações de *driver*. Procurou-se uma solução através de opiniões em um fórum de usuários da placa, mas as primeiras possíveis soluções também falharam.

Um dos desenvolvedores da placa Fox Board o senhor Sergio Tanzilli ficou sabendo do problema por intermédio do professor Miguel Alexandre Wisintainer. O senhor Tanzilli então informou que o problema era que a versão do Linux que possui *driver* de câmera não possuía uma biblioteca chamada *glibc* (biblioteca para linguagem C do projeto GNU). Identificado o problema e comparado com o modelo de Fox Board utilizado verificou-se que não existia uma imagem do Linux para a placa Fox Board MCM com *driver* de câmera e *glibc*. Por uma gentileza do senhor Tanzilli foi compilada e disponibilizada uma nova versão do Linux com todas as funcionalidades necessárias para realização do protótipo.

Os desenvolvedores da placa disponibilizam no seu site uma documentação detalhada facilitando o aprendizado e compreensão. Infelizmente a ferramenta Proteus não possui como componente a placa Fox Board tornando mais difícil a especificação do hardware.

No decorrer do projeto ocorreu um problema de não inicialização do linux na placa Fox Board da Acme Systems utilizada no protótipo. Com isto foi necessário a substituição por uma versão nacional da mesma placa, desenvolvida pelo professor e orientador Miguel Alexandre Wisintainer.

Grande parte do tempo de trabalho foi voltado para o estudo da placa Fox Board e resolução de imprevistos encontrados durante os testes e simulações.

Foram reaproveitados alguns códigos e softwares já implementados na versão do linux utilizada, como o servidor de *streaming* Servfox que inicia os *drivers* da câmera e o JWebcamPlayer.jar que é responsável pela visualização da câmera no navegador. Destacando ainda utilização do *driver* SPCA5xx-LE (LIBLAND, 2007). Trata-se de resultado de uma engenharia reversa de protocolos e funcionalidades de *chips* de câmeras, com a finalidade de acionar câmeras em sistemas operacionais linux embarcados. Na figura 24 é apresentado o protótipo.

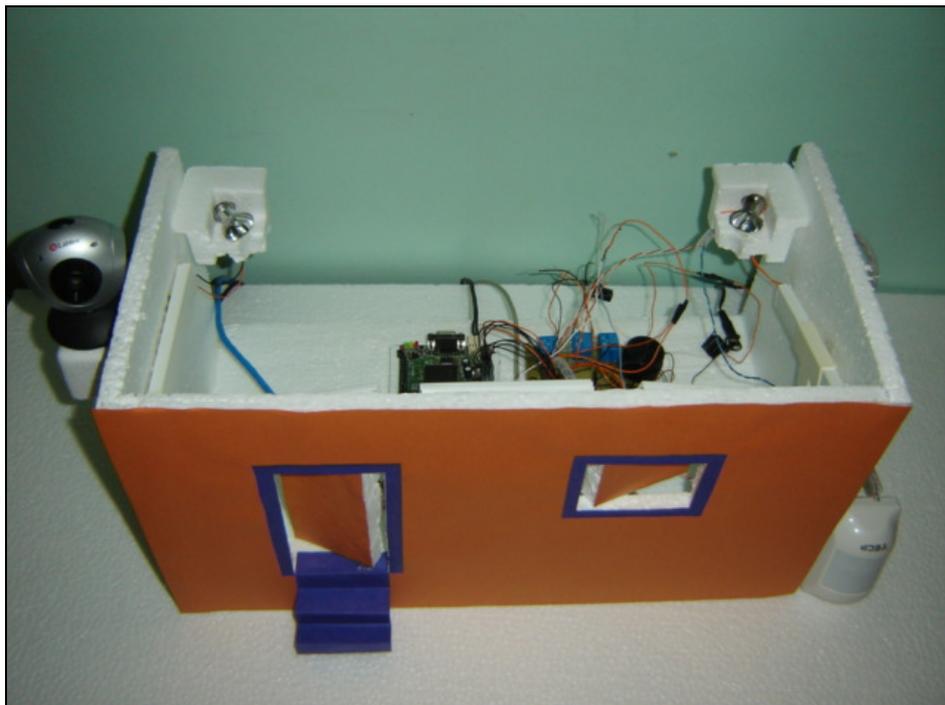


Figura 24 – Protótipo de um sistema de segurança residencial com linux embarcado

## 4 CONCLUSÕES

Este trabalho apresentou um protótipo de um sistema de segurança residencial com Linux embarcado. Verificou-se após a análise do problema que a preocupação com a segurança residencial juntamente com a domótica está se popularizando, produzindo cada vez mais hardwares visando atender este mercado.

Com o protótipo o usuário pode visualizar a distância e em tempo real como esta a movimentação na sua residência através da câmera, verificar quando quiser data e hora e qual sensor foi acionado, e também poderá acionar tomadas na sua residência remotamente.

O tempo gasto para encontrar a causa do problema e possíveis soluções de como conciliar todas as funcionalidades em uma só imagem de Linux foi muito grande, mesmo com ajuda de especialistas, o que prejudicou muito o tempo de desenvolvimento do protótipo.

Uma desvantagem do software do protótipo é que para visualizar *log* de sensores e acionar e desacionar tomadas é redirecionado para uma outra pagina web impossibilitando assim concentrar toda a aplicação em uma só página.

Dentre as principais vantagens do protótipo desenvolvido destaca-se o emprego de linux embarcado utilizando a placa Fox Board. O mercado ainda é novo no Brasil, mas tem uma boa relação de custo benefício e suficiente material disponível a respeito. Uma outra vantagem do protótipo é que ele é independente de plataforma já que é uma aplicação web. Basta apenas o usuário possuir acesso a internet e um navegador, de preferência o Mozilla Firefox, pois os arquivos HTML estão formatados para este navegador.

### 4.1 EXTENSÕES

Como sugestão para futuros trabalhos:

- a) utilizar o modelo de Fox Board LX832 com 8MB FLASH e 32MB RAM para poder fazer uso da ferramenta FoxServe (KDEV, 2007), que possui PHP 5.0.5 e SQLite;
- b) utilizar applet que capture vídeo e/ou imagem e armazene;
- c) instalar motor passo para movimentar a câmera;

- d) substituir os cabos por rede sem fio;
- e) detector de movimento pela câmera;
- f) log de sensores com listagem por sensores;
- g) isolamento com acopladores ópticos.

## REFERÊNCIAS BIBLIOGRÁFICAS

ACME SYSTEMS. **Fox Board**. Ladispoli, [2006?]. Disponível em: <<http://www.acmesystems.it/?id=14>>. Acesso: em 17 set. 2007.

BRUGNERA, Mauro Ricardo. **Domótica**. Novo Hamburgo, [2007]. Disponível em: <<http://www.unibratex.com.br/jornadacientifica/diretorio/FEEVALE+MRB.pdf>>. Acesso em: 24 abr. 2008.

CAMPOS, Augusto. **O que é Linux: BR-Linux**. Florianópolis, 2006. Disponível em <<http://br-linux.org/linux/faq-linux>>. Acesso: em 30 out. 2007.

CAPELLI, Alexandre; BRAGA, Newton. **Sensores: os olhos mecânicos da mecatrônica industrial**. São Paulo: Saber, 2002. Disponível em: <<http://www.mecatronicafacil.com.br/oficina/old/diversos/sensores.htm>>. Acesso em: 18 set. 2007.

CENSI, Angela. **Sistema para automação e controle residencial via e-mail**. 2001. 59 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciência Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CONALLEN, Jim. **Desenvolvendo aplicações web com UML**. Rio de Janeiro: Campus, 2003.

HILGENSTIELER, Fernando. **Protótipo de software para monitoração do cabeçalho do protocolo HTTP em uma rede TCP/IP**. 2003. 51 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <[http://www.bc.furb.br/docs/MO/2003/278729\\_1\\_1.pdf](http://www.bc.furb.br/docs/MO/2003/278729_1_1.pdf)>. Acesso em: 12 set. 2007.

KDEV. **Fox Serve**. Como, 2007. Disponível em: <<http://www.kdev.it/HTDOCS/index.html>>. Acesso em: 26 maio 2008.

LIBLAND. **Spc5xx and Linux Embedded**. [S.l.], 2008. Disponível em: <<http://mxhaard.free.fr/embedded.html>>. Acesso em 11 jun. 2008.

MAMMANA, Jean Charles.; CARDOCHE, Jérôme. **Robotbuilder**. [S.l.], 2007. Disponível em: <<http://www.robotbuilder.org/>>. Acesso em: 12 set. 2007.

MARTINS, Rodrigo. **Começa a popularização da internet**. São Paulo, 2007. Disponível em: <[http://www.link.estadao.com.br/index.cfm?id\\_conteudo=11580](http://www.link.estadao.com.br/index.cfm?id_conteudo=11580)>. Acesso em: 11 set. 2007.

MONTIBELLER JUNIOR, Ariberto. **Protótipo de sistema de monitoramento remoto utilizando TCP/IP sobre Ethernet (802.3)**. 2005. 65 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciência Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <[http://www.bc.furb.br/docs/MO/2005/305466\\_1\\_1.pdf](http://www.bc.furb.br/docs/MO/2005/305466_1_1.pdf)>. Acesso em: 12 set. 2007.

PÉRICAS, Francisco Adell. **Redes de computadores: conceitos e a arquitetura internet**. Blumenau: EdiFURB, 2003.

POVARESKIM. Povareskim Soft & System Ltda. São Paulo, 2008. Disponível em: <<http://www.povareskim.com.br/chromephoto/dicas/dicas.html>>. Acesso em: 04 jun. 2008.

TAROUCO, Liane M. R. **Sistemas Cliente/Servidor**. Porto Alegre, [1996]. Disponível em: <[http://penta.ufrgs.br/redes296/cliente\\_ser/tutoria\\_.htm](http://penta.ufrgs.br/redes296/cliente_ser/tutoria_.htm)>. Acesso em: 22 set. 2007.

WEINMAN, Willian E. **Manual de CGI**. Tradução Daniel Vieira. São Paulo: Makron Books, 1997.