

ALEXANDRE AUGUSTO HAMDAN SIQUARA GARCIA

**Desenvolvimento do Módulo Central
de Controle microprocessado do
sistema de automação residencial
SCAR para integração e
processamento de dados de
memória e periféricos.**

*Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia de São Carlos,
da Universidade de São Paulo*

*Curso de Engenharia Elétrica com ênfase em
Eletrônica*

ORIENTADORA: Profa. Dra. MARIA STELA VELUDO DE PAIVA

**São Carlos
2008**

Dedicatória

Dedico este trabalho aos meus amigos companheiros de desenvolvimento do projeto SCAR, Bruno Kimura, Carolina Bermudez, Daniel Medeiros e Matheus Fujita.

Agradecimentos

Agradeço acima de tudo a **Deus**,

Aos meus pais, **Ana** e **Jorge**, por terem me guiado até aqui com tanta sabedoria, paciência e amor;

Agradeço à minha **família**, por estar sempre presente nos momentos que mais precisei;

Agradeço aos meus amigos **Bruno Kimura**, **Daniel Medeiros**, **Matheus Fujita** e **Felipe Zanone**, simplesmente por serem meus amigos;

Aos meus **colegas de república** que me agüentaram por cinco anos e me ajudaram em vários momentos;

Agradeço também à minha orientadora, profa. **Maria Stela**, pela dedicação e compreensão na realização desse trabalho.

Muito obrigado!
Alexandre Garcia

Sumário

Lista de Tabelas.....	6
Lista de siglas.....	7
Resumo.....	8
Abstract.....	9
Capítulo 1 – Introdução.....	10
1.1 Automação Residencial – Residências Inteligentes.....	10
1.2 Motivações e Vantagens.....	10
1.2.1 Segurança.....	11
1.2.2 Acessibilidade, Conforto e Assistência.....	11
1.2.3 Racionalização no uso de Energia Elétrica.....	11
1.3 O projeto SCAR.....	12
1.4 Objetivo.....	14
Capítulo 2 - Fundamentação Teórica.....	15
2.1 O conceito de Domótica.....	15
2.2 Estrutura para Automação Residencial.....	16
2.3 Convergência de padrões.....	17
2.4 Retrofitting.....	18
Capítulo 3 – O Módulo Central de Controle.....	19
3.1 Introdução.....	19
3.2 Implementação.....	19
3.3 O Sistema Operacional e a estrutura de software.....	20
3.4 O controle do Painel de Gerenciamento de Dispositivos.....	22
3.4.1 O controle de telas do sistema.....	23
3.4.2 Decodificação das coordenadas XY do sensor Touchscreen.....	24
3.5 O Gerenciamento das Interfaces de Controle de Dispositivos (ICD).....	27
3.6 O controle da Interface de gerenciamento remoto.....	31
Capítulo 4 – Futuras Implementações.....	33
Capítulo 5 – Conclusão.....	34
Anexos.....	35
A. Sistemas embarcados para o hardware do sistema central.....	35
A.1 A arquitetura ARM.....	35
A.2 A arquitetura PC/104.....	36
A.3 A arquitetura mini-ITX.....	38
B. Redes de conexão de dispositivos.....	39
B.1 Rede Baseada no sistema de distribuição de Energia Elétrica.....	39
B.1.1 Protocolo CEBus & Home PnP.....	39
B.1.2 Protocolo X-10.....	40
B.2 Controle de Periféricos por cabos dedicados.....	42
B.2.1 O Protocolo USB.....	42
B.2.2 O Protocolo FireWire (IEEE 1394).....	42
B.2.3 O Protocolo CAN.....	43
B.3 Redes de controle sem fio.....	44
B.3.1 Protocolo Bluetooth.....	45
B.3.2 ZigBee.....	45
B.3.3 Wi-Fi (IEEE 802.11).....	45
B.3.4 Infravermelho.....	46
B.3.5 Comparativo da comunicação sem fio.....	46
C. Lista de funções para controle da interface USB em GNU/Linux.....	48
D. Inicialização do dispositivo Bluetooth através de interface USB-Serial.....	48
Referências Bibliográficas.....	49
Apêndices.....	50
A. Espalhamento espectral.....	50
B. Sistema operacionais multitarefas.....	51
B.I Multitarefa de antecipação (preemptivo).....	51
B.II Multitarefa de cooperação.....	51
C. Modelo de camadas ISO.....	52

Lista de figuras

FIGURA 1 - CONCEITO DE AUTOMAÇÃO RESIDENCIAL [1].....	10
FIGURA 2 – ESQUEMA DE FUNCIONAMENTO DO PROJETO SCAR.....	13
FIGURA 3 – FUNCIONAMENTO DE UMA RESIDÊNCIA INTELIGENTE, ADAPTADO [1].....	15
FIGURA 4 - RACK AGREGANDO EM UM ÚNICO LOCAL O SISTEMA CENTRALIZADO [3].....	16
FIGURA 5 – REDE CABEADA COM NÓ PROVEDOR DE ACESSO SEM FIO [3].....	16
FIGURA 6 – REDE DOMÉSTICA DE AUTOMAÇÃO [3].....	17
FIGURA 7 – ESTRUTURA DO MÓDULO CENTRAL DE CONTROLE.....	20
FIGURA 8 – DIAGRAMA DE ESTADOS DE FUNCIONAMENTO DO SOFTWARE.....	21
FIGURA 9 – COMUNICAÇÃO SERIAL MÓDULO CENTRAL E PAINEL LCD-TOUCHSCREEN.....	22
FIGURA 10 – ESTRUTURA DE MEMÓRIA DE TELAS DO SISTEMA.....	23
FIGURA 11 – FRAME DE TRANSMISSÃO DE TELA MCC PARA PGD.....	23
FIGURA 12 – ÁREAS MAPEADAS DA PELÍCULA TOUCHSCREEN.....	24
FIGURA 13 – ASSOCIAÇÃO MATRIZ TOUCHSCREEN DE TELA 1 E ARQUIVOS DE COMANDO DO SISTEMA.....	25
FIGURA 14 - ASSOCIAÇÃO MATRIZ TOUCHSCREEN DE TELA 2 E ARQUIVOS DE COMANDO DO SISTEMA.....	25
FIGURA 15 – FLUXOGRAMA DE DECODIFICAÇÃO DOS COMANDOS DO ARQUIVO ESPELHO.....	26
FIGURA 16 – ESTRUTURA DE DADOS DE CADA DISPOSITIVO DA ICD DISTRIBUÍDA NO FRAME DE COMUNICAÇÃO USB.....	27
FIGURA 17 – ESTRUTURA DE COMUNICAÇÃO COM AS INTERFACES DE CONTROLE DE DISPOSITIVOS (ICDS).....	28
FIGURA 18 – ESTRUTURA DE BUSCA POR DISPOSITIVO E ILUSTRAÇÃO DA ATUALIZAÇÃO DO BYTE DE PRÉÂMBULO.....	29
FIGURA 19 – FLUXOGRAMA DE CONTROLE DA COMUNICAÇÃO REMOTA VIA BLUETOOTH.....	32
FIGURA 20 - MICROPROCESSADOR ARM [19].....	35
FIGURA 21 – PLACA PC/104 [19].....	37
FIGURA 22 – CAPACIDADE DE EMPILHAMENTO DE PLACAS DA ARQUITETURA PC/104 [13].	37
FIGURA 23 – PLACA PC MINI-ITX [14].....	38
FIGURA 24 – CICLOS DO PROTOCOLO X-10 NA LINHA DE ENERGIA, ESQUEMA DE IDENTIFICAÇÃO SETORES [1].....	41
FIGURA 25 – PACOTES X-10 [1].....	41
FIGURA 26 – REDE FIREWIRE PARA CONEXÃO DE DISPOSITIVOS MULTIMÍDIA [1].....	43
FIGURA 27 – REDE DE DISPOSITIVOS RESIDENCIAIS MONTADA SOBRE PROTOCOLO CAN [1].....	44
FIGURA 28 – FUNCIONALIDADES DOS PROTOCOLOS SEM FIO. [1].....	47
FIGURA 29 – PILHA DA CAMADA OSI [18].....	52

Lista de Tabelas

TABELA 1 – IDENTIFICAÇÃO DOS DISPOSITIVOS IMPLEMENTADOS.....	30
TABELA 2 – EVOLUÇÃO DA ARQUITETURA PC/104.....	36
TABELA 3 – COMPARATIVO ENTRE O PROTOCOLO USB E FIREWIRE [1].....	43
TABELA 4 – LISTA DE FUNÇÕES PRINCIPAIS PARA CONTROLE DE INTERFACE USB EM GNU/ LINUX.....	48

Lista de siglas

ANSI – *American National Standards Institute*
ARM – *Advanced RISC Machine*
CAN – *Controller Area Network*
CEBus – *Consumer Electronics Bus*
CSMA – *Carrier Sense Multiple Access*
CDCR – *Collision Detection and Resolution*
DI – Dispositivos Inteligentes
EIA – *Electronic Industries Alliance*
EST – *Energy Saving Trust*
PGD – Painel de Gerenciamento de Dispositivos
I2C – *Inter Integrated Circuit*
ICD – Interface de Controle de Dispositivos
IEEE – *Institute of Electrical and Electronics Engineers*
IGR – Interface de Gerenciamento Remoto
HVAC – *Heating, Ventilation & Air Conditioning*
LCD – *Liquid Crystal Display*
MARC – *Medical Automation Research Center*
MCC – Módulo Central de Controle
OSI – *Open System Interconnection*
PAN – *Personal Area Networks*
SCAR – Sistema de Controle e Automação Residencial
SO – Sistema Operacional
ULA – Unidade Lógica Aritmética
USB – *Universal Serial Bus*

Resumo

Inserido num contexto de um sistema de automação residencial intitulado SCAR (Sistema de Controle e Automação Residencial), o presente trabalho descreve o desenvolvimento de um módulo de controle baseado em um computador padrão IBM/PC, capaz de integrar toda a rede de módulos periféricos desse sistema e fazer o controle dos dispositivos (lâmpadas e tomadas). Este módulo comunica-se com os dispositivos que compõem os periféricos através de interface USB, e envia dados a um módulo que permite o controle dos dispositivos da casa por meio de um painel de gerenciamento, com tela LCD e *Touchscreen*. Além disso, controla o *driver* de comunicação *Bluetooth*, que permite que um celular operando em plataforma JAVA, possa controlar de forma remota os dispositivos da residência.

Palavras Chaves: Domótica, *TouchScreen*, *Bluetooth*, USB, GNU/Linux

Abstract

Inserted in a context of a residential automation system named SCAR (Residencial Automation and Controlling System), this work describes the development of a control module based in a PC computer that is able to integrate all the network peripherals modules of the system and control all the devices (lamps and sockets). This module communicates with the devices that compose the peripherals through the USB interface and send data to a module which has *LCD Screen* and *TouchScreen*, that controls the devices. It controls as well as the *Bluetooth* communication driver which allows to control remotely residential devices with a cellphone operating in JAVA plataform.

Capítulo 1 – Introdução

1.1 Automação Residencial – Residências Inteligentes

O termo residências inteligente vem atraindo crescente interesse, já que possibilitam a atuação supervisionada e não supervisionada de dispositivos eletrônicos, exercendo tarefas complexas e interagindo com usuários e com o meio físico. A adoção de sistemas computacionais, redes de dados e dispositivos de automação em residências têm crescido muito nos últimos anos, impulsionada pela introdução da computação pessoal e do aparecimento da Internet [1].

Um sistema de automação residencial permite controlar a residência remotamente, poupando tempo com tarefas repetitivas (como acender e apagar lâmpadas), economizando energia, dinheiro e aumentando o conforto. A revolução da automação residencial está baseada no fato de permitir a comunicação entre os vários aparelhos presentes em uma residência, como ilustrado na figura 1.

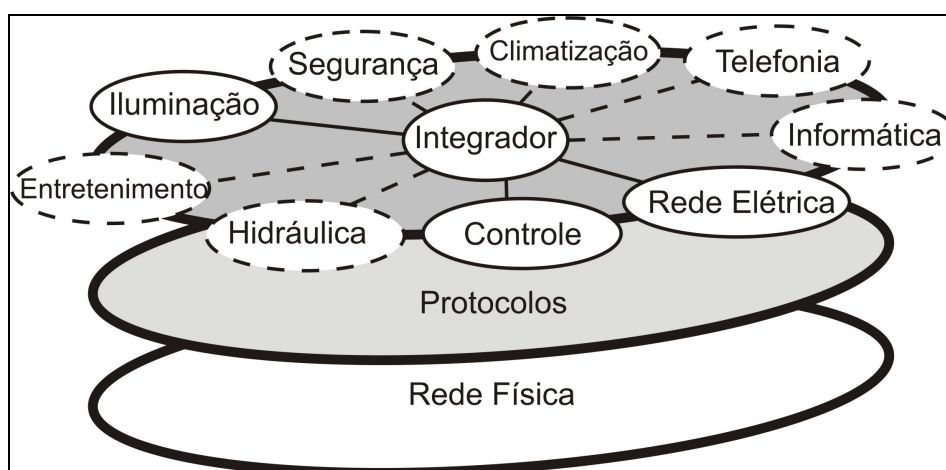


Figura 1 - Conceito de automação residencial [1]

1.2 Motivações e Vantagens

As motivações para implementação de um sistema de automação residencial estão baseadas em três aspectos: o primeiro relacionado à segurança do ambiente e das pessoas presentes na casa, o segundo relacionado ao conforto e acessibilidade do usuário para controle dos dispositivos da casa e o terceiro diz respeito à conservação e uso eficiente do recurso de energia elétrica na residência.

1.2.1 Segurança

O aspecto de segurança implementado por um sistema de automação residencial está em primeiro lugar relacionado à segurança dos usuários, como por exemplo, ao desligar as tomadas do quarto de uma criança evitando assim os riscos de acidente. Um segundo aspecto está relacionado à segurança patrimonial que engloba sistema de prevenção a roubos, no qual é feita a integração de câmeras, sistema de iluminação, sistema de acesso entre outros e, também sistema de prevenção a incêndios através da análise inteligente de diversos sensores como de temperatura e fumaça e, atuação sobre iluminação, rede elétrica da residência entre outros.

1.2.2 Acessibilidade, Conforto e Assistência

O conceito de Residência Inteligente torna possível ajudar idosos e/ou deficientes físicos a viver independentemente utilizando-se de instalações que possuam tecnologias para um controle fácil dos meios da residência, associada com atuadores que facilitam diversas atividades e que permitem até controle de dispositivos remotamente.

Nesse sentido o *Medical Automation Research Center* (MARC), da Universidade de Virginia, fundou um projeto de Residências Inteligentes para avaliar a habilidade de uso de um sistema com vários sensores e fornecer controles do monitoramento da saúde [5]. Na Universidade da Flórida foi também desenvolvido um projeto, o *Gator Tech Smart House*[6], orientado especificamente para pessoas idosas, com o objetivo de criar ambientes Inteligentes, tais como, residências que possam perceber o contexto do ambiente e então assisti-las. A idéia é que a tecnologia alerte aos parentes e o pessoal da saúde sobre alterações no padrão normal de atividades do idoso.

1.2.3 Racionalização no uso de Energia Elétrica

A redução do consumo de energia elétrica de uma residência pode ser alcançada através da utilização de programas de gerenciamento de energia. Inicialmente pode até parecer contraditório reduzir o consumo instalando novos dispositivos eletrônicos, mas o fato é que isso leva a uma melhor racionalização e uso inteligente, o que acaba resultando em economia. Um programa gerenciador, aliado a sensores e atuadores, pode otimizar a utilização de todos os equipamentos da residência, principalmente os de maior potência (maior consumo). Um exemplo é a ativação de equipamentos de grande porte como sistemas de HVAC, em horários pré-determinados, que é um primeiro passo na racionalização, juntamente com o controle de portas, janelas e persianas que permitem uma maior utilização da luz natural, porém de maneira inteligente, balanceando a incidência do

sol para não prejudicar o ar-condicionado.

Uma tendência muito em breve, assim como já acontece para a indústria, é a tarifação da energia pelas concessionárias, levando em conta o horário do uso. Neste contexto é interessante programar, por exemplo, eletrodomésticos, como o lava-louças e a máquina de lavar, para operar em horário de menor custo da energia. Tais sistemas de gestão podem incluir técnicas de auto-aprendizagem temporal, ou seja, eles podem memorizar o esquema de funcionamento de aparelhos como climatizadores e sistemas de iluminação, e assim criar bases de dados para calcular o tempo necessário de inicialização para obtenção do conforto adequado. Durante um período de não utilização, o sistema deverá ser capaz de se ajustar para situações como “noturno”, “férias” ou “não funcionamento”. Durante períodos de utilização, ele deve ser capaz de conservar energia e otimizar o uso dos equipamentos a partir dos critérios pré-estabelecidos de conforto.

Segundo estimativas do *Energy Saving Trust* (EST) [4], Grã Bretanha, equipamentos de alta tecnologia responderão por quase metade do consumo total de energia em um domicílio britânico típico, em 2020. O EST afirma que os eletrônicos superarão os eletrodomésticos e a iluminação como maiores consumidores domésticos de energia. Hoje é muito comum os aparelhos novos ficarem em modo *standby* ao invés de serem desligados, alguns até nem mesmo dispõem de um botão de desligar. O estudo estima que, em 2020, os televisores em *standby* consumirão 1,4 por cento de toda a eletricidade usada em domicílios. Segundo o EST, os domicílios britânicos poderiam economizar 37 libras ao ano em suas contas de energia se desconectassem os aparelhos das tomadas (retirassem do modo *standby*).

1.3 O projeto SCAR

O projeto SCAR foi idealizado a partir de um trabalho realizado na disciplina SEL 373 – “Projeto de Sistemas Digitais” do Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos, SEL – EESC/USP. Este é constituído de quatro partes, são elas: o Módulo Central de Controle (MCC) que será apresentado nesse trabalho, um Painel de Gerenciamento de dispositivos (Fujita [16]), Interfaces de Gerenciamento de Dispositivos (ICDs) (Bermudez [17]) e uma Interface de Gerenciamento Remoto (IGR) (Medeiros [15]), como apresentado na figura 2.

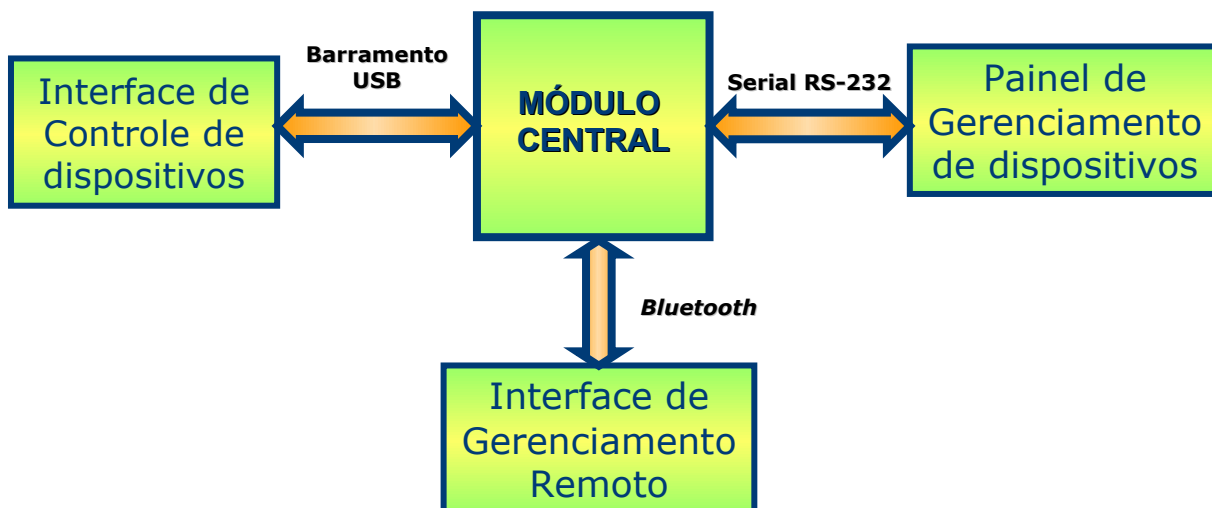


Figura 2 – Esquema de funcionamento do projeto SCAR

Módulo central de controle

O Módulo Central de Controle é pensado como um sistema microprocessado, com *hardware* e *software*, dotado de uma quantidade de memória para manipulação de dados (dezenas a centenas de *megabytes* de memória RAM) e disponibilidade de memória de armazenamento de dados na ordem de *gigabytes*. Essas quantidades de memórias são importantes, pois é no módulo central que são armazenadas as telas do sistema.

A função dos *softwares* de controle dentro do MCC é gerenciar os demais subsistemas e protocolos do projeto SCAR.

Painel de Gerenciamento de Dispositivos

Com a finalidade de propiciar métodos fáceis e intuitivos de controle dos dispositivos da residência, percebeu-se a necessidade de implementação de uma interface de comunicação e controle que além de ser de fácil utilização e intuitiva, tornar-se-ia o elemento principal de controle dos dispositivos e recursos da casa. Pensou-se assim, em uma interface gráfica (baseada em uma tela de LCD) com sensor *TouchScreen* para entrada de dados no sistema. Tal interface foi desenvolvida utilizando-se um microcontrolador ATMEL® 89S52 (Fujita [16]). Essa interface foi concebida para ser interligada ao controle do Módulo Central de Controle, funcionando como uma interface de entrada e saída de dados do sistema central, que os processa e os envia para essa interface.

Interface de Controle de Dispositivos

O controle de uma vasta rede de dispositivos inteligentes, distribuídos ao longo da residência, exige que o processamento e o controle dos dispositivos sejam feitos de maneira distribuída e com certo grau de autonomia, permitindo que cada nó ou ramificação da rede possa ter um grau de processamento independente (autonomia). Nesse contexto, surgiu a

necessidade de desenvolvimento de um módulo de controle de dispositivos que suportasse de maneira satisfatória a diversidade de dispositivos de uma residência, e que ao mesmo tempo apresentasse baixo custo. Assim, fez-se uma escolha por construir essa interface sobre a plataforma de microcontroladores PIC, que reúne uma grande diversidade de meios de comunicação, tendo inclusive uma interface USB interna (protocolo adotado para rede residencial de dispositivos do projeto SCAR) permitindo fácil implementação da comunicação com o Módulo Central de Controle.

Interface de Gerenciamento Remoto

Permitir o controle e gerenciamento remoto de dispositivos é um ponto chave no aspecto de acessibilidade e conforto do sistema de uma residência inteligente. Para tanto foi pensada numa forma de controle remoto dos recursos das residências, que interagisse diretamente com o Módulo Central de Controle e que permitisse operar em quaisquer dispositivos móveis, como celular ou PDA. Assim, optou-se por utilizar aplicativos executados em plataforma JAVA, visto que essa linguagem apresenta a facilidade de ser interpretada e, portanto multi-plataforma, podendo um mesmo código ser executado em qualquer dispositivo de qualquer fabricante, bastando apenas a execução através da Máquina Virtual JAVA. O protocolo sem fio escolhido para a comunicação foi o *Bluetooth*, uma vez que este está presente na maioria dos aparelhos celulares atuais, que é certamente o equipamento móvel mais comum e utilizado nos dias de hoje.

1.4 Objetivo

O objetivo do presente trabalho foi o de desenvolver um Módulo Central de Controle (MCC) para gerenciamento dos subsistemas do projeto SCAR, como ilustrado na figura 2. O módulo funciona como um integrador do sistema, sendo responsável pelo controle dos demais módulos implementados: sistemas de interface de controle para o usuário (Painel de Gerenciamento de Dispositivos – PGD e Interface de Gerenciamento Remoto – IGR) e sistema de controle de iluminação e rede elétrica (Interface de Controle de Dispositivos – ICD).

Capítulo 2 - Fundamentação Teórica

2.1 O conceito de Domótica

A palavra domótica originou-se do latim *domus* que significa casa. Trata-se dos conceitos modernos de engenharia para sistemas residenciais e equipamentos a eles ligados. Com o uso de novas tecnologias pretende-se criar sistemas de controle automático de uma residência ou edifício, tornando as residências inteligentes [1].

Os dispositivos inteligentes (DIs) são a parte fundamental de uma residência inteligente. São equipamentos eletro-eletrônicos que além de exercerem as funções para as quais foram concebidos, agregam *hardware* e *software* adicionais que lhes provêm recursos extras, permitindo o controle e gerenciamento remoto e sua interconexão em rede.

Uma residência inteligente pode extrair informações sobre as condições físicas do ambiente por meio da utilização de sensores e detectores. Pode interpretar tais informações, planejar ações e realizar inferências utilizando a capacidade de raciocínio embutida no controlador central ou nos DIs. Finalizando o ciclo, pode executar ações por meio dos atuadores, afetando as condições do ambiente, produzindo novas situações, conforme apresentado na figura 3.

O termo “inteligente” é utilizado neste trabalho como diferenciador entre o estado mais simples dos sistemas em questão e o seu estado de valores agregados ou evoluído. O termo “inteligente” é usado para descrever um dispositivo que é construído com a possibilidade de se autogerir e até mesmo gerir outros equipamentos. No entanto, não se quer fazer qualquer menção à capacidade humana, e nem mesmo envolver conceitos definidos em áreas como inteligência artificial ou ciência cognitiva. O termo é apropriado para descrever uma capacidade dos dispositivos e do próprio sistema, que por combinarem em algum grau autonomia, tomada de decisão e inferência, podem ser chamados de inteligentes [1].

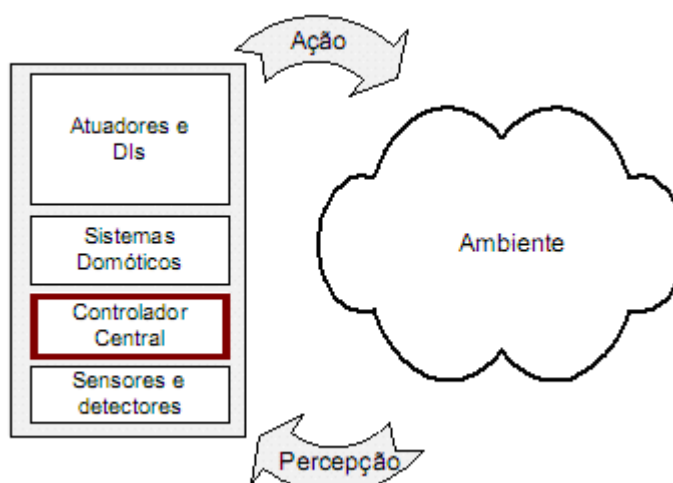


Figura 3 – Funcionamento de uma residência inteligente, adaptado [1]

2.2 Estrutura para Automação Residencial

Um fator a ser considerado quando se decide instalar um sistema de automação residencial é designar um espaço para absorver todos os equipamentos principais do sistema e prover o cabeamento adequado a todos os pontos da residência (figura 4). Um ambiente de um metro quadrado, conhecido como sala de controle (*central wiring closet*), seria ideal para acomodar tudo, porém muitos não têm condições de reservar um espaço deste para tal função. Desse modo, pode-se utilizar um *rack* para acomodar os equipamentos e uma caixa, semelhante à de disjuntores, com todos os eletrodutos para distribuição dos cabos, como ilustrado nas figuras 4 e 5. Com isso, computador (elemento principal do módulo central de processamento), modems, *hubs*, moduladores de vídeo, amplificadores de áudio, sistema de telefonia (PABX), sensores e emissores de infravermelho, acopladores de impedância, equipamentos de segurança e de automação e controle ficarão organizados e dispostos em um único local [1].

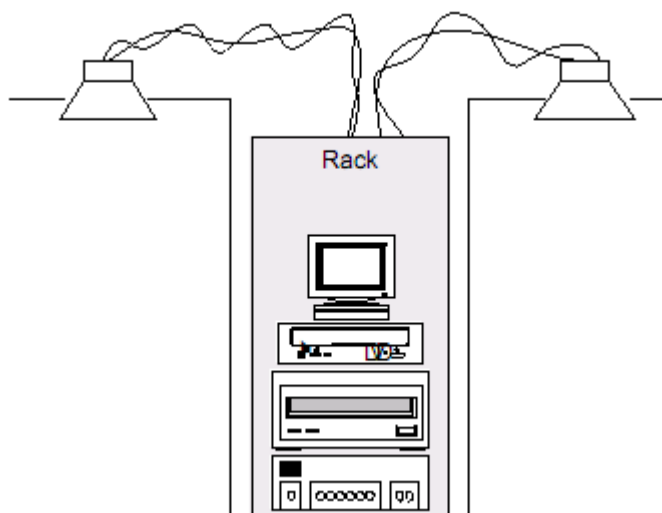


Figura 4 - Rack agregando em um único local o sistema centralizado [3]

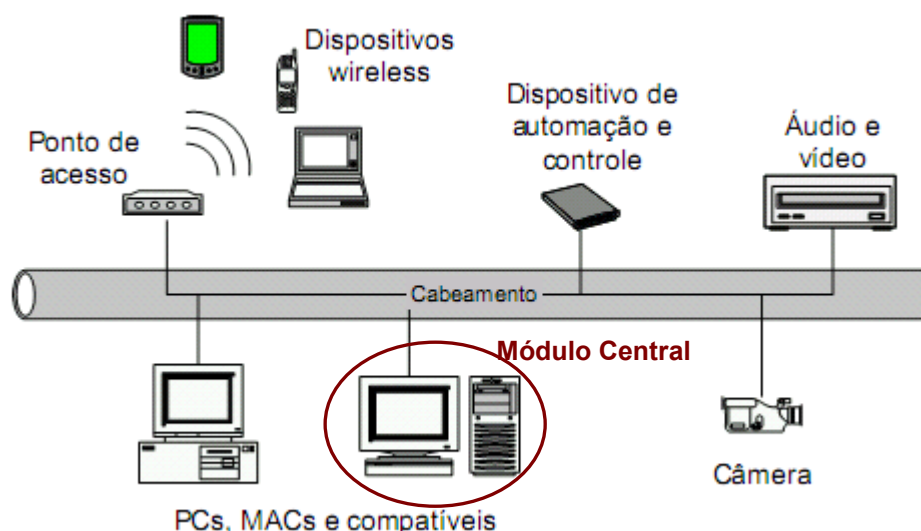


Figura 5 – Rede cabeada com nó provedor de acesso sem fio [3]

2.3 Convergência de padrões

A convergência dos diversos sistemas acelera a possibilidade de tornar a residência em um centro de lazer e entretenimento. Onde antes existia apenas uma TV e um videocassete, hoje em dia existem elaborados sistemas multimídias e de automação residencial, que estão ficando cada vez mais acessíveis (figura 6). Iluminação e sonorização de diversos ambientes podem ser controladas por um número crescente de dispositivos, tais como controles remotos e painéis de controle, acoplados com sistema *TouchScreen*. Estes controles possuem interfaces que vêm se tornando progressivamente mais amigáveis, permitindo que todos em casa operem o sistema.

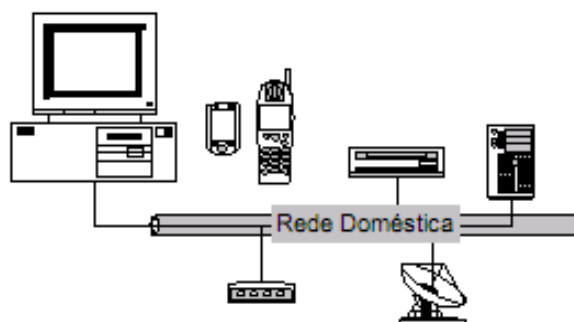


Figura 6 – Rede doméstica de automação [3]

Um dos pontos principais no desenvolvimento de sistemas de automação para residências inteligentes é a integração dos mesmos. Por exemplo, todos os sensores utilizados no sistema de intrusão podem ser utilizados também para controlar a iluminação dos compartimentos. Da mesma forma, todos os sensores de temperatura atuam permanentemente como detectores de incêndio. Assim, os sistemas de segurança de hoje em dia podem fazer mais do que pedir ajuda. Numa rede de comunicações de dados utilizada para este fim, os mesmos sensores que detectam movimento, fogo ou líquidos podem ser utilizados para ativar um número variado de ações. Caso haja necessidade, o sistema de segurança pode controlar o sistema de ar-condicionado e ventilação (desligando-os em caso de incêndio), bem como destrancar portas, janelas e atuar em outros sistemas na residência, também enviar sinais para ligar e desligar a iluminação de acordo com os estado dos detectores de movimento e auxiliar o funcionamento das câmeras de vigilância. Utiliza-se para isso um sistema centralizado de controle adequado a esta realização, coordenando todos os elementos dispostos na rede de DIs interligada.

2.4 Retrofitting

Ao contrário do que muitos imaginam o conceito de residência inteligente e automação residencial não fica restrito apenas a novas residências, que já estão sendo construídas pensando nas funcionalidades do sistema. Obviamente que esse pré-planejamento ajuda muito na concepção total do projeto, porém grande parte do mercado para automação residencial estará nas residências já construídas, sendo que a maioria delas não foi planejada sob o ponto de vista de uma residência inteligente.

O termo *Retrofitting* é utilizado quando se introduz uma modificação em algo previamente construído. No contexto de automação residencial, esse termo se insere na adaptação de uma residência que não foi pensada em termos de automação residencial (necessidades de cabeamento, arranjo de iluminação, arranjo de HVAC - *Heating, Ventilation & Air Conditioning* etc), para que esta possa receber a estrutura necessária para construção de uma rede de dispositivos inteligentes (DIs).

Com isso, insere-se no contexto do projeto de automação residencial, soluções que viabilizem a adaptação de ambientes para instalação da rede de DIs, por exemplo, o uso de tecnologia sem fio (em substituição às necessidade de cabeamento) para transmissão de dados, ou mesmo tecnologias que permitam a utilização da rede elétrica já presente na residência. Essas tecnologias são melhor explicadas no Anexo B.

Capítulo 3 – O Módulo Central de Controle

3.1 Introdução

O Módulo Central de Controle tem por função gerenciar os diversos meios e recursos da residência, permitindo que todos os subsistemas residenciais (implementados pelo projeto SCAR) operem de maneira integrada. Este agrega uma estrutura de *hardware* e *software* e exerce também a função de *gateway*, permitindo a interconexão de diferentes redes com os diferentes protocolos e dados, interligando-se através dele a rede de dispositivos, *intranet*, Internet, rede de serviços telefônicos entre outras. A função do MCC é fazer o controle geral do sistema e interconexão dos módulos do projeto SCAR.

O MCC do projeto SCAR gerencia as telas do sistema, interpreta os comandos vindos do Painel de Controle e da Interface de Acesso Remoto via *Bluetooth* e também controla o *status* dos dispositivos residenciais. Com alta capacidade de armazenamento de dados, é nesse módulo que ficam armazenadas as telas do sistema e são então enviadas ao Painel de Gerenciamento de Dispositivos para exibição. O MCC também processa ainda os dados das coordenadas XY vindos da película *TouchScreen*.

A rede de dispositivos inteligentes é implementada a partir do barramento USB que interliga diversas interfaces de gerenciamento de dispositivos e tem o módulo central com controlador *Host* da rede USB. MCC também implementa comunicação *Bluetooth* via interface USB-Serial.

3.2 Implementação

Para implementação do módulo central foi utilizado um microcomputador do tipo IBM/PC como unidade central de gerenciamento do projeto SCAR. A escolha deveu-se à disponibilidade do computador para implementação, e também por este já ser o instrumento de desenvolvimento dos aplicativos, permitindo assim a realização do desenvolvimento e teste no mesmo equipamento. Outra vantagem é a disponibilidade das interfaces de comunicação, necessárias para o sistema como a porta serial padrão RS-232 e USB (função *host*), sem que houvesse assim, a necessidade de agregar ou desenvolver módulos adicionais para essa função, facilitando a implementação dos meios de comunicação, e permitindo um enfoque maior nas estruturas de controle lógico dos dispositivos (controle de dados do ICD-USB) e meios de acesso (LCD-*Touchscreen* e Celular-*Buetooth*). A figura 7 ilustra a comunicação do Aplicativo Gerenciador SCAR com os demais módulos do sistema, mostrando a estrutura dos aplicativos (camada superior), os protocolos de comunicação (camada intermediária) e módulos controlados pelo sistema (camada inferior).

O sistema operacional utilizado no módulo central foi um GNU/Linux baseado em Debian, com *Kernel 2.6.8*. Trata-se de um conjunto de *softwares* e *driver* com código fonte aberto, o que facilita o desenvolvimento de aplicativos sobre essa plataforma. Sua estrutura de sistema operacional já oferece *drivers* de suporte aos protocolos de comunicação adotados, e suas funcionalidades de sistema operacional já se encarregam do gerenciamento de memórias e controle de chamada dos dispositivos.

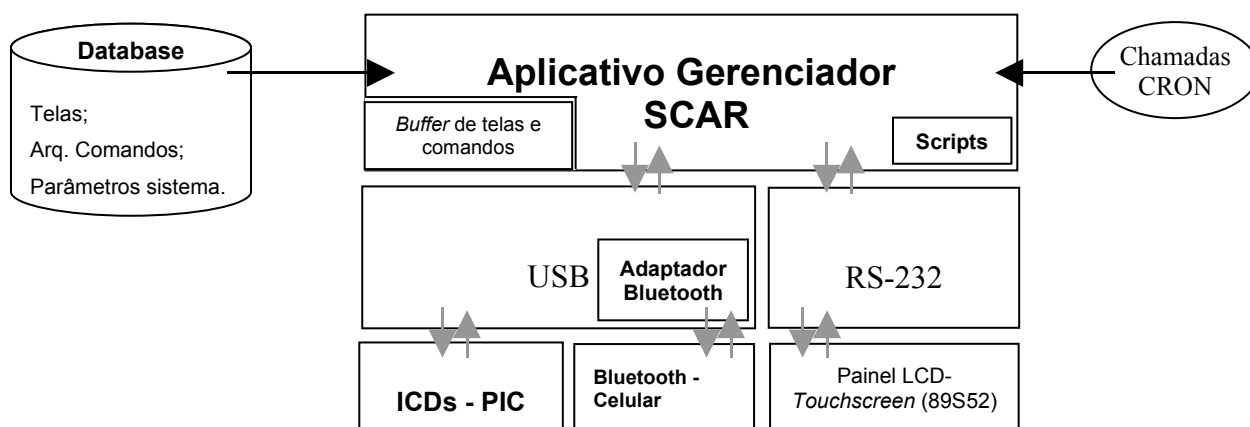


Figura 7 – Estrutura do Módulo Central de Controle

3.3 O Sistema Operacional e a estrutura de *software*

A arquitetura do sistema operacional GNU/Linux é bastante estável e segura [7]. Ele trabalha em modo protegido, oferece proteção de memória para os seus aplicativos e é multitarefa preemptiva (apêndice B), além de ser multi-usuário. O GNU/Linux foi desenvolvido para empresas, universidades, computadores pessoais (PCs) e vem sendo utilizado em larga escala, inclusive em aplicações embarcadas.

Os aplicativos desenvolvidos para o Módulo Central de Controle foram programados em linguagem C padrão ANSI[8][9], executados em *console* (sem bibliotecas gráficas), com a utilização também de linguagem Bash-Script chamadas em *software* e executada em *console* BASH[10].

Aplicativos do *Shell* do sistema operacional também são utilizados no MCC, são eles: o agendador de tarefas CRON utilizado para pré-programação de rotinas de controle de lâmpadas e tomadas, o configurador de comunicação serial SETSERIAL e o *software* ECHO usado para o envio de comandos diretamente aos *buffers* das entradas e saídas.

Para a escrita do software utilizou-se o pacote GNU C *Library* versão 6 (também chamada *libc*) com bibliotecas para desenvolvimento que inclui suporte a comunicação serial, controle do sistema, armazenamento de dados e controle USB. Os aplicativos foram

compilados utilizando-se o compilador GCC-3.3 [11]. A figura 8 apresenta o diagrama de estado do software de gerenciamento do projeto SCAR, a comunicação com cada módulo gerenciado é explicada nos tópicos seguintes 3.4, 3.5 e 3.6.

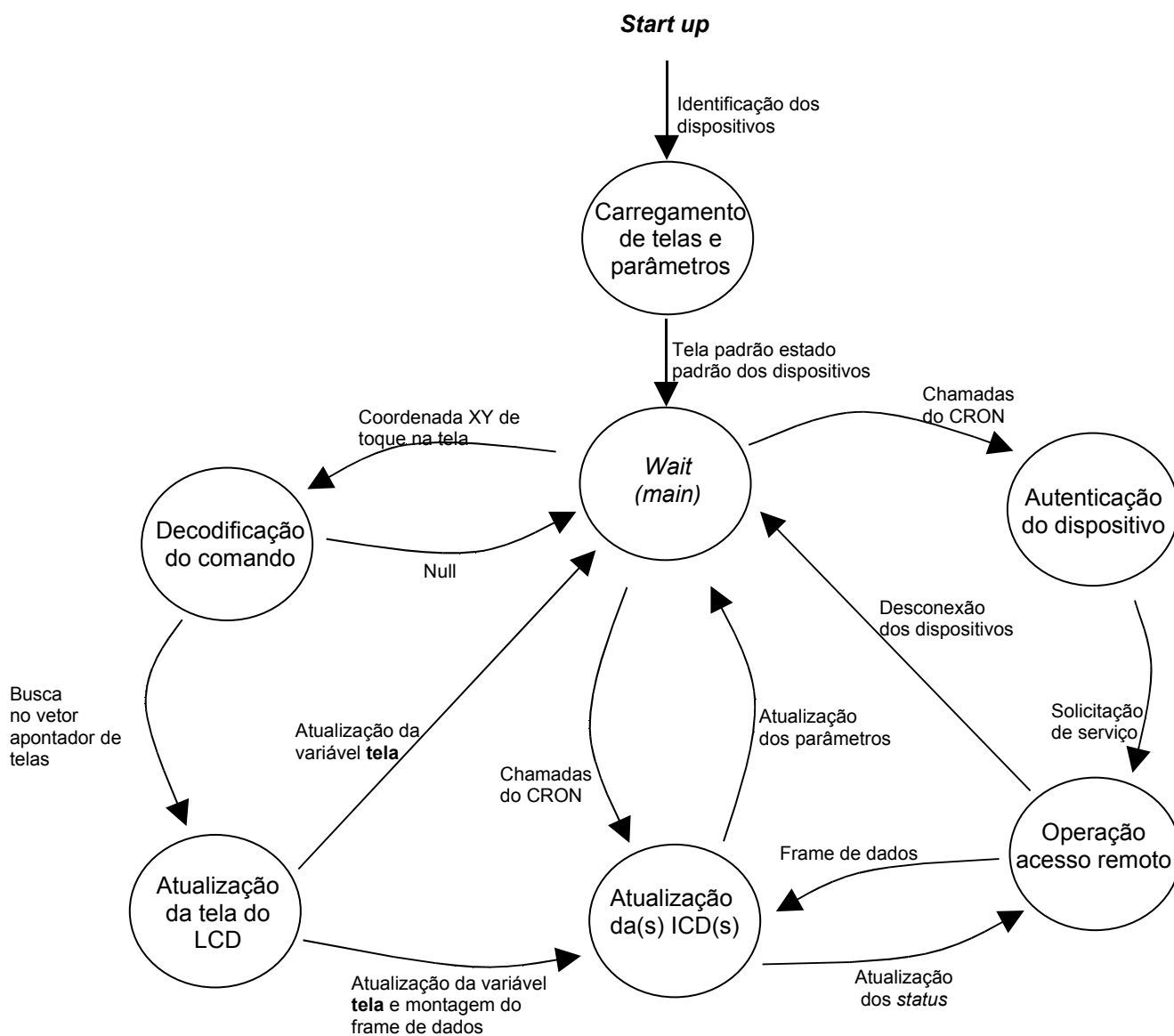


Figura 8 – Diagrama de Estados de funcionamento do software

3.4 O controle do Painel de Gerenciamento de Dispositivos

O Módulo Central de Controle exerce duas funções ligadas ao Painel de Gerenciamento de Dispositivos (PGD): a primeira é o armazenamento das telas a serem apresentadas no LCD do painel, e a segunda refere-se à decodificação das coordenadas XY da matriz *TouchScreen* enviadas ao MCC pelo controlador do painel. Dessa forma, o processamento das telas do sistema e a interpretação dos comandos de toque estão a cargo do MCC, sendo este encarregado pelo envio das telas para exibição, ao controlador do painel, e devendo também receber do microcontrolador do PGD os *bytes* das coordenadas XY do sensor *Touchscreen* (figura 9).

A transmissão de dados entre o sistema central e o painel de controle é feita sem codificação, utilizando-se caracteres específicos (vide Fujita [16]) para sinalização da comunicação, que é realizada através de interface serial RS-232, configurada com taxa de transmissão de 9600 *bauds*, com palavras de 8 bits sem controle de paridade.

A formação da imagem de uma tela no LCD do projeto SCAR (modelo Samsung® UG-32F11, Fujita [16]) exige 28800 bytes (320 pixels x 240 linha x 3 /8). Para a transmissão entre MCC e PGD, mais 4 bytes de controle são somados junto aos dados para formação da tela, o que resulta em uma estrutura de 28800+4 bytes na transmissão. No sentido inverso o PGD envia ao MCC dois bytes referenciando as coordenadas X e Y, respectivamente, do toque na película *touchscreen*. A figura 8 ilustra a transmissão de dados entre o módulo central e o painel LCD-*Touchscreen*.

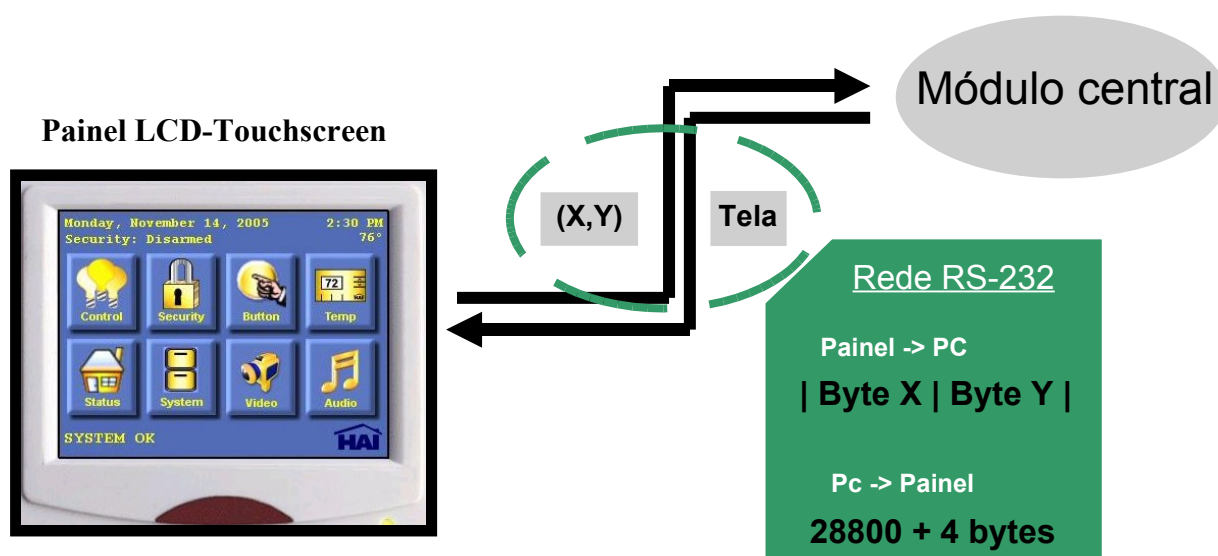


Figura 9 – Comunicação Serial Módulo Central e Painel LCD-*Touchscreen*

3.4.1 O controle de telas do sistema

As informações da configuração do sistema de automação implementado ficam centralizadas em um arquivo de configuração. Esse arquivo possui a quantidade de telas de trabalho que o sistema possui e o número de dispositivos mapeados no sistema. Ao carregá-lo, o *software* de controle do MCC aloca um ponteiro de dados, que por sua vez armazena os ponteiros para cada tela do sistema, cada um com 28800bytes de memória alocada do sistema.

Conforme o usuário navega pelas telas do sistema, o módulo central envia ao Painel de Gerenciamento a tela a ser exibida, e durante esse processo o ponteiro da tela corrente bem como o da tela anterior é armazenado em variáveis, permitindo assim, o controle da tela que é exibida e o controle da tela anterior (auxiliando a função *escape* para retorno de tela) (figura 10). Essa estrutura de gerenciamento de telas permite orientar o sistema no sentido de pré-carregar as telas futuras prováveis de serem exibidas (conforme o encadeamento das telas), permitindo uma resposta mais rápida do sistema.

As telas do sistema ficam armazenadas em arquivos texto no HD com 28800bytes, com os dados em formato hexadecimal.

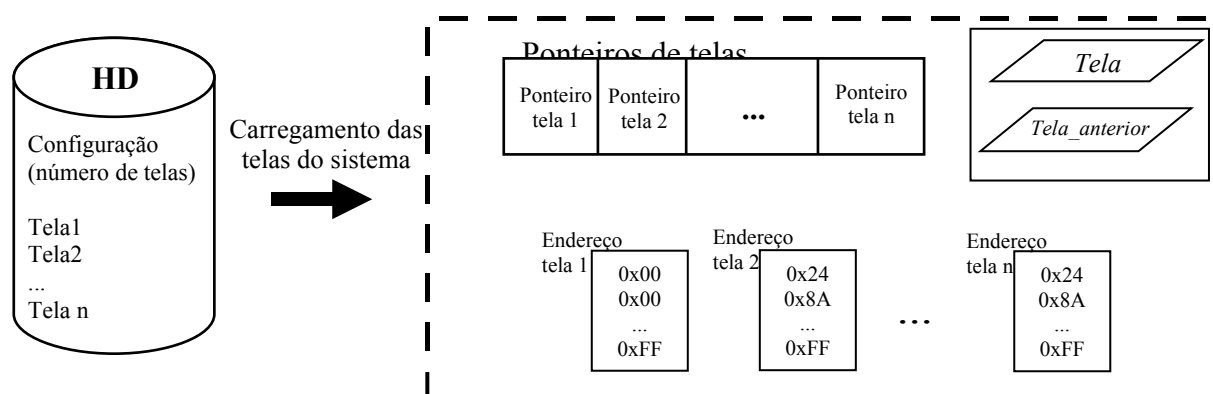


Figura 10 – Estrutura de memória de telas do sistema

A transmissão de dados de uma tela entre o Módulo Central de Controle e o Painel de Gerenciamento de Dispositivos é feita através de um conjunto de bytes formado por um byte de *start*, 2 bytes de cabeçalho (reservados para futuras implementações), os 28800 bytes do conjunto de dados da tela e um byte de *stop*, que também sinaliza se a tela deve ser imediatamente exibida ou armazenada no buffer do PDG (figura 11). A transmissão do conjunto de dados de formação da tela (28800bytes) é feita de 40 em 40bytes, com confirmação de controle entre cada rajada de dados (Fujita [16]).

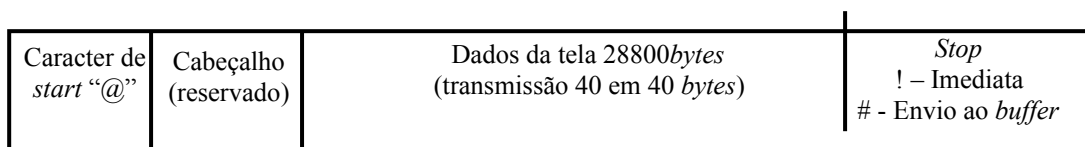


Figura 11 – Frame de transmissão de tela MCC para PGD

3.4.2 Decodificação das coordenadas XY do sensor *Touchscreen*

A decodificação das coordenadas XY (da região pressionada na película do sensor *TouchScreen*) e associação dessas região à execução de uma operação, são feitas pelo Módulo Central de Controle, que recebe via comunicação serial dois bytes do Painel de Gerenciamento de Dispositivos, as coordenadas X e Y respectivamente. A figura 12 apresenta as áreas mapeadas pelo MCC para o *TouchScreen*, num total de 36 áreas de contado na tela.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

Figura 12 – Áreas mapeadas da película *TouchScreen*

Após a recepção das coordenadas e da decodificação da posição na matriz da tela, o *software* de controle determina qual comando está associado a essa região, conforme a tela corrente em exibição. Feita a decodificação da região selecionada (pressionada), o MCC busca o comando associado àquela posição, conforme ilustrado pelas figuras 13 e 14. Para isso, cada tela do sistema tem um arquivo de comando (arquivo espelho) correspondente, com 36 linhas indicando os comandos do sistema associados às respectivas áreas de toque da tela. Assim como os arquivos de tela, os arquivos de comandos também são do tipo texto, porém apresentam um número variável de dados conforme a diversidade de comandos listados. Seu tamanho varia entre centenas de *bytes* a alguns *kilobytes*, podendo chegar a alguns casos a dezenas de *kilobytes*, dependendo dos textos complementares de identificação de comando (textos de comentários incluídos no arquivo) e que são úteis no processo de manutenção do sistema. A inserção de textos de comentários é feita no arquivo após o caracter “#” que indica que o restante de linha deve ser ignorado para a interpretação de comandos. A figura 15 apresenta o algoritmo de busca, execução e tratamento de um comando solicitado.

As operações que executam a partir de um comando selecionado podem ser as seguintes: a apresentação de uma nova tela no painel (retorno, avanço etc), a leitura ou alteração do *status* de um dispositivo controlado, ou ambas as operações simultaneamente.

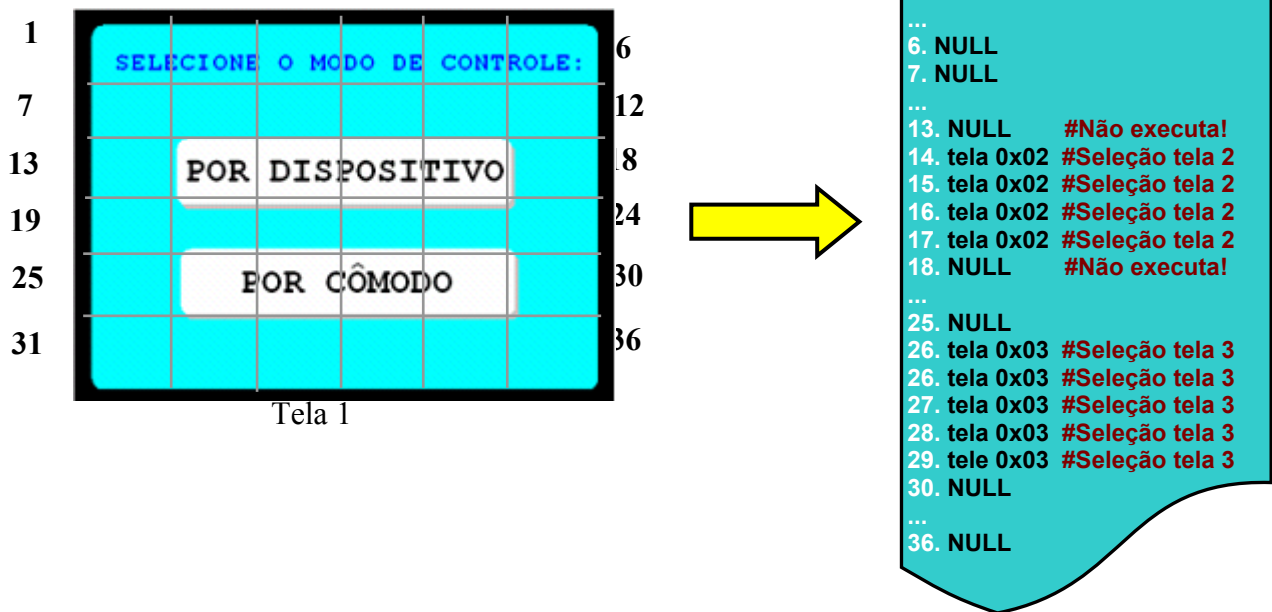


Figura 13 – Associação matriz *Touchscreen* de tela 1 e arquivos de comando do sistema

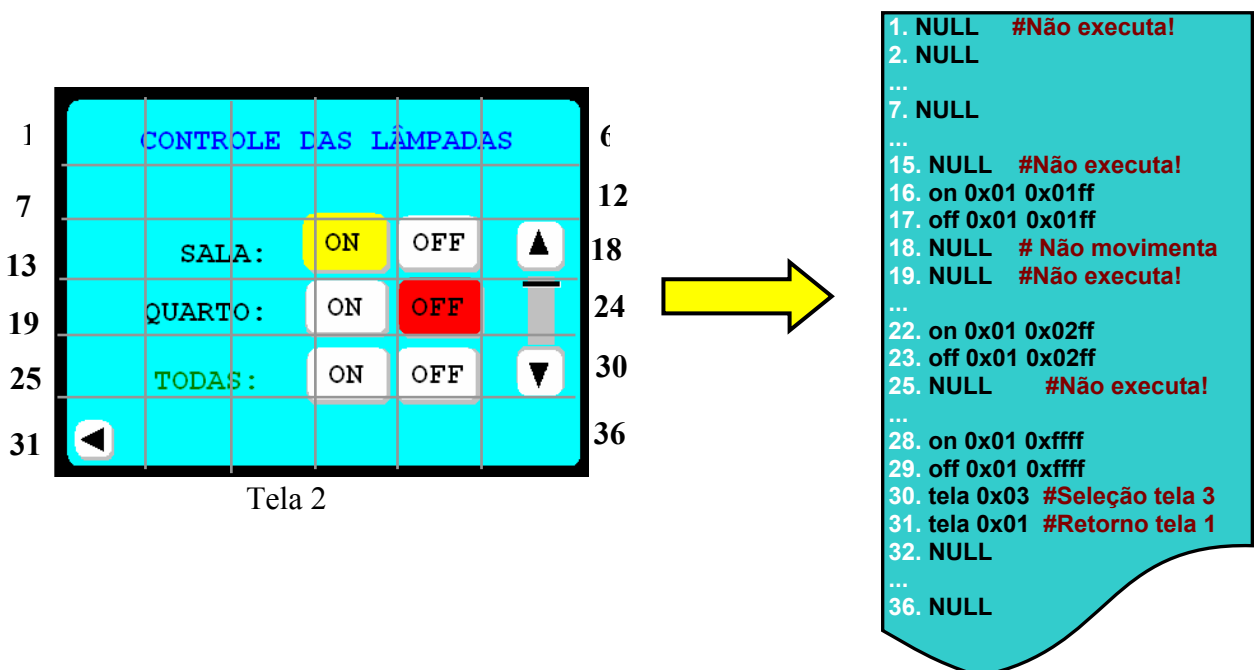


Figura 14 - Associação matriz *Touchscreen* de tela 2 e arquivos de comando do sistema

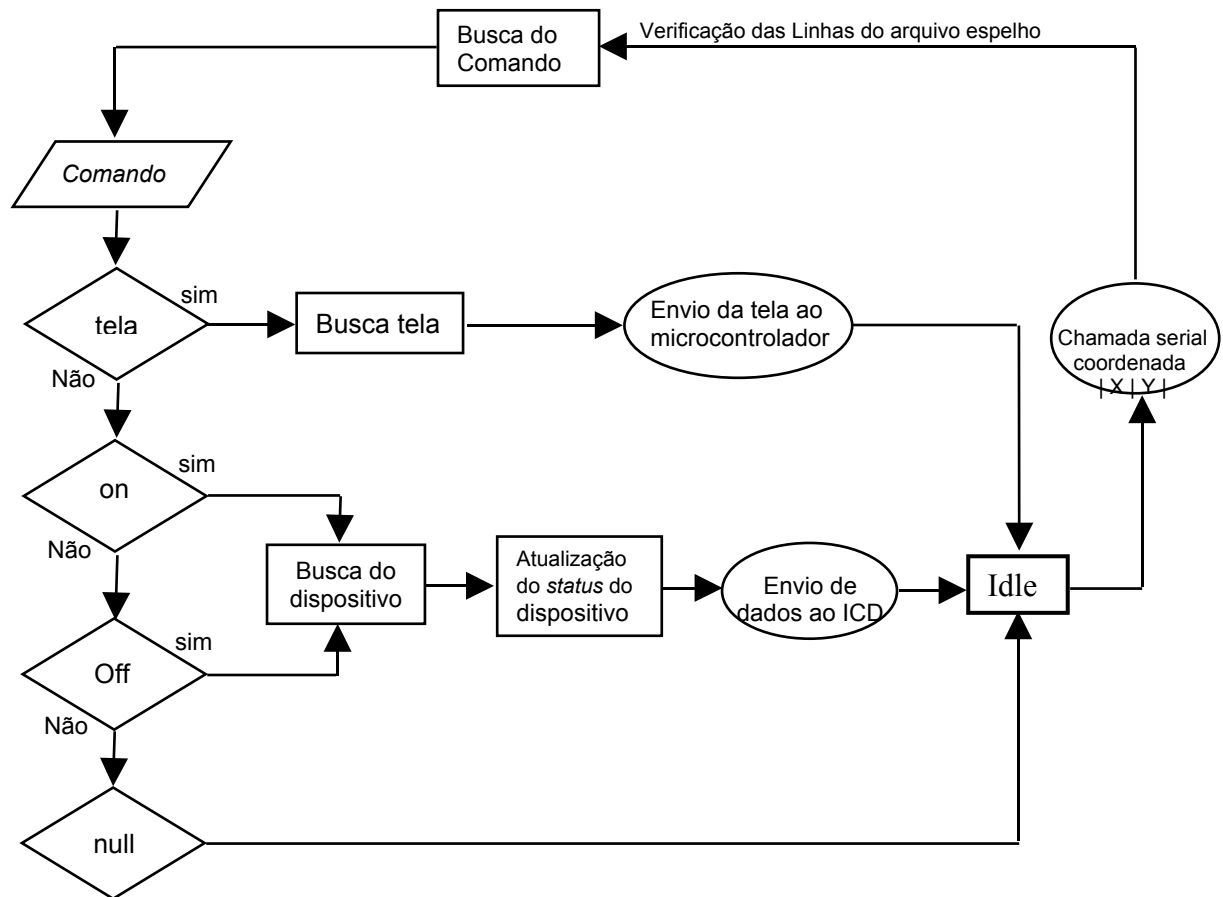


Figura 15 – Fluxograma de decodificação dos comandos do arquivo espelho

3.5 O Gerenciamento das Interfaces de Controle de Dispositivos (ICD)

A implantação da rede de comunicação USB para o gerenciamento de dispositivos residencial é facilitada pelo uso de um computador do tipo PC como elemento do MCC, uma vez que o PC já possui o *hardware* para conexão USB e também o Sistema Operacional GNU/Linux, que já tem *drivers* nativos para comunicação USB.

O reconhecimento dos dispositivos conectados à rede é feito de maneira automática pelo Sistema Operacional, que carrega os parâmetros iniciais da ICD assim que detectada (vide monografia Bermudez [17]).

Duas bibliotecas de funções para manipulação das interfaces USB operando como dispositivos de armazenamento estão disponível no conjunto estendido da biblioteca libC, e são elas: *usb.h* e *usbdevice_fs.h*. Essas bibliotecas apresentam um lista de funções para controle e envio de dados pela interface. A lista das funções utilizadas para implementação encontra-se no anexo C.

A estrutura de transmissão de dados entre o computador e microcontrolador da Interface de Controle de Dispositivos é feita sobre a estrutura do frame USB para transmissão por interrupção, que tem o tamanho máximo de 1024 *bytes* (Bermudez [17]). Dentro dessa estrutura, os dados relativos à informação de cada dispositivo ligado ao ICD são agrupados de 4 em 4 *bytes*, sendo um *byte* de *start* entre cada grupo.

Os grupos de dados de cada dispositivo são ordenados no frame pelo *byte* referente ao **tipo** de dispositivo e pelo número de identificação na ICD (2º *byte* de identificação). A figura 16 apresenta o frame da transmissão USB para uma ICD que controla lâmpadas e tomadas. A rede de dispositivos residenciais via barramento USB tem capacidade de controle de até 127 dispositivos (figura 17) [17].

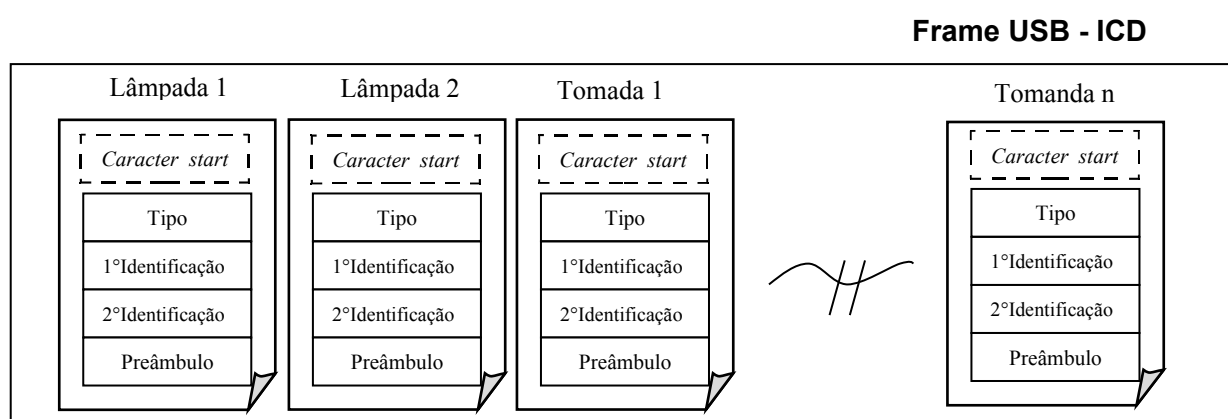


Figura 16 – Estrutura de dados de cada dispositivo da ICD distribuída no frame de comunicação USB

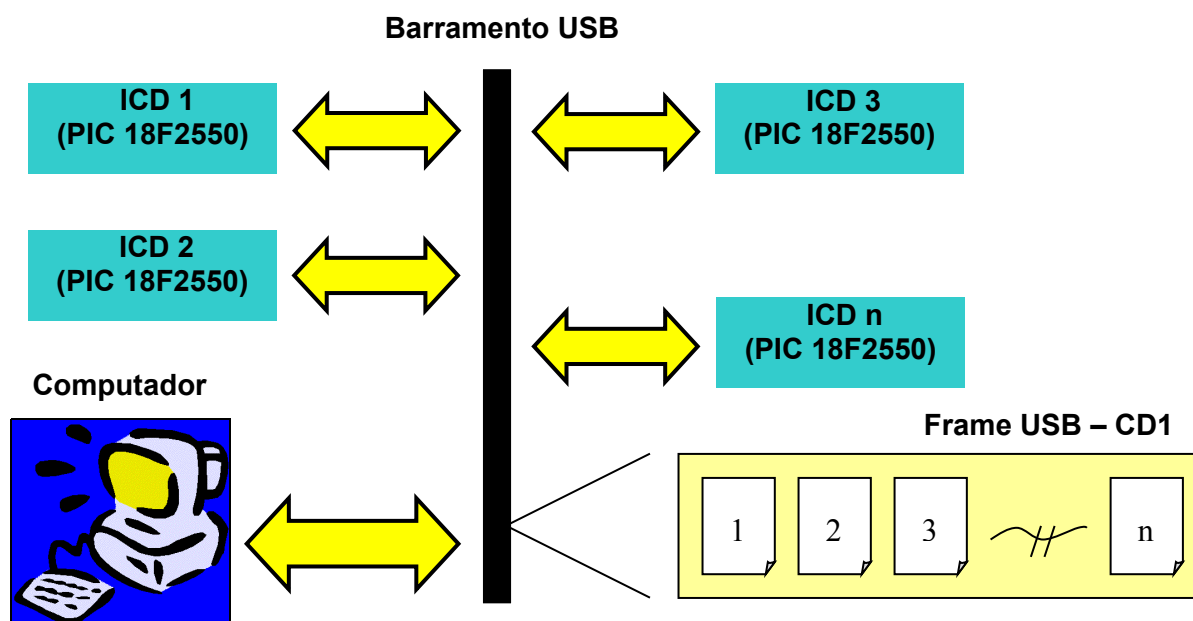


Figura 17 – Estrutura de comunicação com as Interfaces de Controle de Dispositivos (ICDs)

Conforme o comando solicitado, o Módulo Central de Controle lê os *status* dos dispositivos da ICD, ou pode modificar o estado de um dispositivo na ICD alterando seu *status* no *byte* de preâmbulo.

A sequência para modificação do *status* de um dispositivo inicia com a entrada de um comando proveniente do Painel-Touchscreen ou celular-Bluetooth. O comando é então decodificado e a partir disso o sistema realiza uma busca pelo dispositivo selecionado, para então atualizar seu *byte* de preâmbulo e assim modificar o estado do dispositivo na ICD.

O processo de busca por dispositivo no sistema é dividido em duas etapas: na primeira é feita uma busca indireta por ICD, analisando-se o valor do 1º *byte* de identificação do dispositivo, que serve para indicar de maneira indireta em qual delas ele está instalado. Nessa etapa busca-se em um vetor com 256 posições, cada uma associada ao número de uma ICD, e no qual é armazenado o ponteiro para a estrutura de dados dela. Na segunda etapa de busca, já dentro da estrutura de dados (com apontamento indireto por ICD), é feita uma identificação através do TIPO de dispositivo e pelo número do dispositivo presente no 2º *byte* de identificação, fazendo-se uma busca concatenada entre as duas variáveis. A figura 18 apresenta a estrutura de busca por dispositivos.

Após a busca, é feita a atualização do *byte* de preâmbulo associado ao dispositivo modificado, e em seguida é transmitido ao ICD os novos dados, atualizando-se nessa operação todos os dados dos dispositivos gerenciados por ele, independentemente de terem sido alterados ou não.

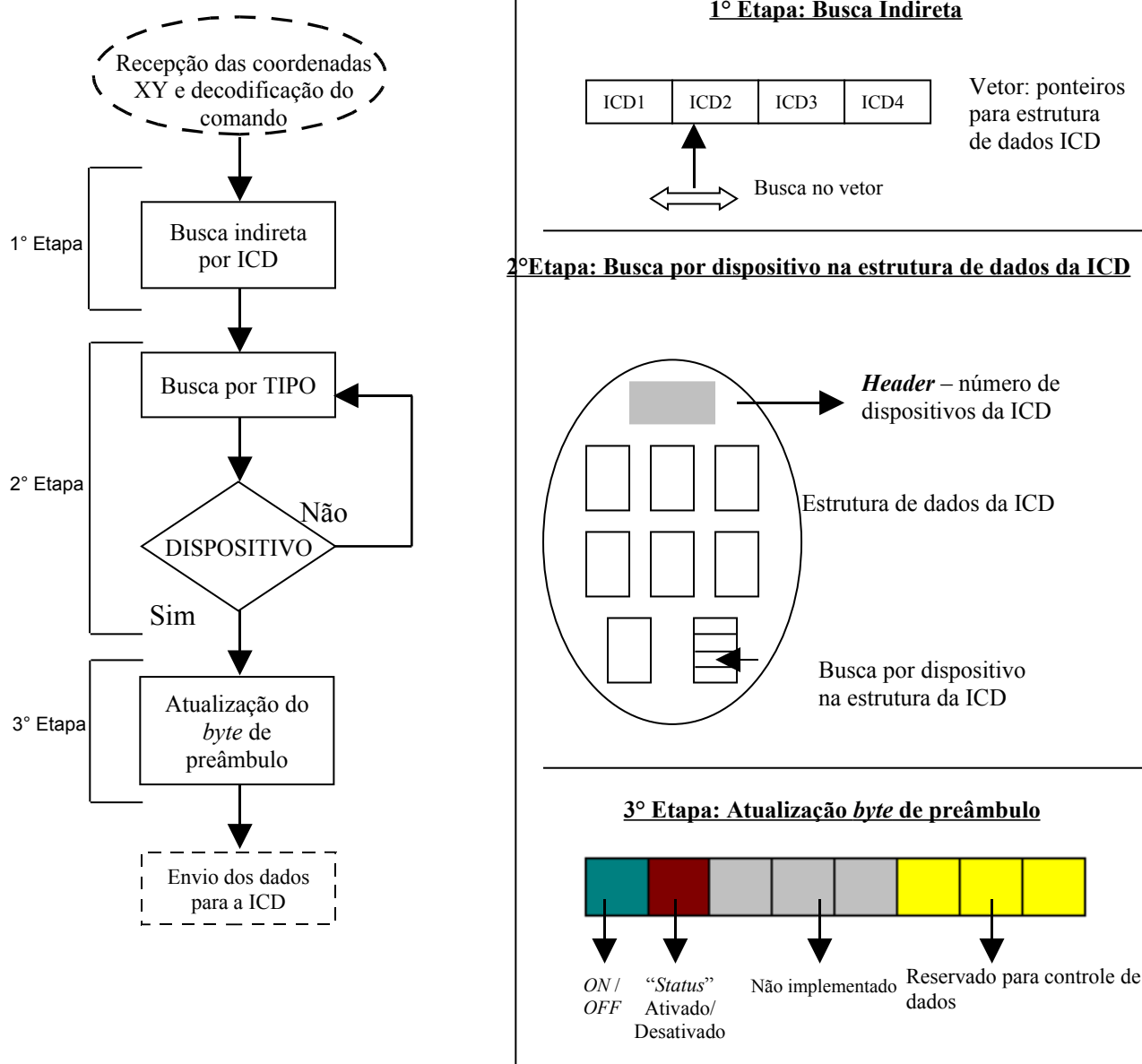


Figura 18 – Estrutura de busca por dispositivo e ilustração da atualização do byte de préambulo

A tabela 1 apresenta os códigos de identificação dos dispositivos implementados no projeto SCAR. Esses códigos são utilizados para identificação dos dispositivos nos arquivos de comando de cada tela, na busca por dispositivos e na identificação dos dispositivos para a comunicação remota via *Bluetooth* apresentada na seção 3.6.

Tabela 1 – Identificação dos dispositivos implementados

Tipos de dispositivos (Tipo)	
Endereços	Dispositivos
0x00	Reservado
0x01	Lâmpada
0x02	Tomadas
0x03... 0xfe	Disponível para futuras aplicações
0xff	Reservado
Identificação das ICDs (1º byte de identificação - 0xAAyy)	
Endereços	Identificação
0x00yy	Reservado
0x01yy... 0xfeyy	Numeração das ICDs
0xffyy	Broadcast para todas as ICDs
Identificação dos dispositivos (2º byte de identificação - 0xyyAA)	
Endereços	Identificação
0xyy00	Reservado
0xyy01... 0xyyfe	Identificação de cada dispositivo
0xyyff	Todos os dispositivos da ICD

3.6 O controle da Interface de gerenciamento remoto

A implementação da interface de comunicação *Bluetooth* foi feita através de um adaptador ligado à porta USB do computador. O próprio GNU/Linux já oferece os *drivers* para utilização de adaptador USB *Bluetooth* com protocolo RFCOMM, que virtualiza a comunicação como uma interface serial de comunicação de dados, o que permite que a mesma estrutura de gerenciamento do Painel-*Touchscreen* seja utilizada para o controle da interface do acesso *Bluetooth*.

Ao “inicializar” o *software* de controle, esse busca o dispositivo virtual de comunicação *Bluetooth* no caminho `/dev/rfcomm0`, e caso não seja encontrado o *software*, prossegue carregando os módulos (*drivers*) de comunicação USB-*Bluetooth*. Caso algum dispositivo *Bluetooth* seja localizado o *software* tenta contato e faz a autenticação pelo MAC *Address* do celular, verificando se esse tem permissão para acessar a rede. A cada 1 minuto, configurado a partir de chamada do CRON, é feita uma nova verificação de dispositivos celulares que possuam solicitação de serviços *Bluetooth*. No anexo D são apresentadas as chamadas em BASH. A configuração da interface para comunicação *Bluetooth* é feita dentro do *software* através do comando “`system();`” e executa o métodos para localização dos dispositivos *Bluetooth*.

Na comunicação com o celular o Módulo Central responde às solicitações do mesmo. Quando o usuário abre o programa de gerenciamento residencial no celular (vide Medeiros [15]) esse envia ao MCC um *byte* de controle. Caso o Módulo Central autentique o celular, envia a ele um código de ativação indicando sua aceitação na rede de automação residencial. Após essa etapa o celular envia comandos ao MCC solicitando atualização do *status* dos dispositivos e passa também a fazer chamada através dos comandos *on* e *off*, e dos códigos de identificação dos dispositivos (os mesmos dos arquivos de comando). O módulo central interpreta o comando, inicia a busca do dispositivo e em seguida executa a operação que pode ser o envio ao celular do *status* do dispositivo, a atualização dos *status* do dispositivo na ICD, o controle da tela LCD-*touchscreen* ou ambos. No estabelecimento da comunicação, a estrutura lógica de controle segue a mesma estrutura usada para o Painel de Gerenciamento de Dispositivo e Interface de Controle de Dispositivos (descrito nas seções 3.4 e 3.5). A diferença está na forma como os dados são transmitidos. A figura 19 apresenta o diagrama de gerenciamento e comunicação com o dispositivo remoto via USB-Serial.

Os dispositivos da rede residencial já são pré-programados no celular do usuário com os respectivos códigos de referência (vide monografia Medeiros [15]).

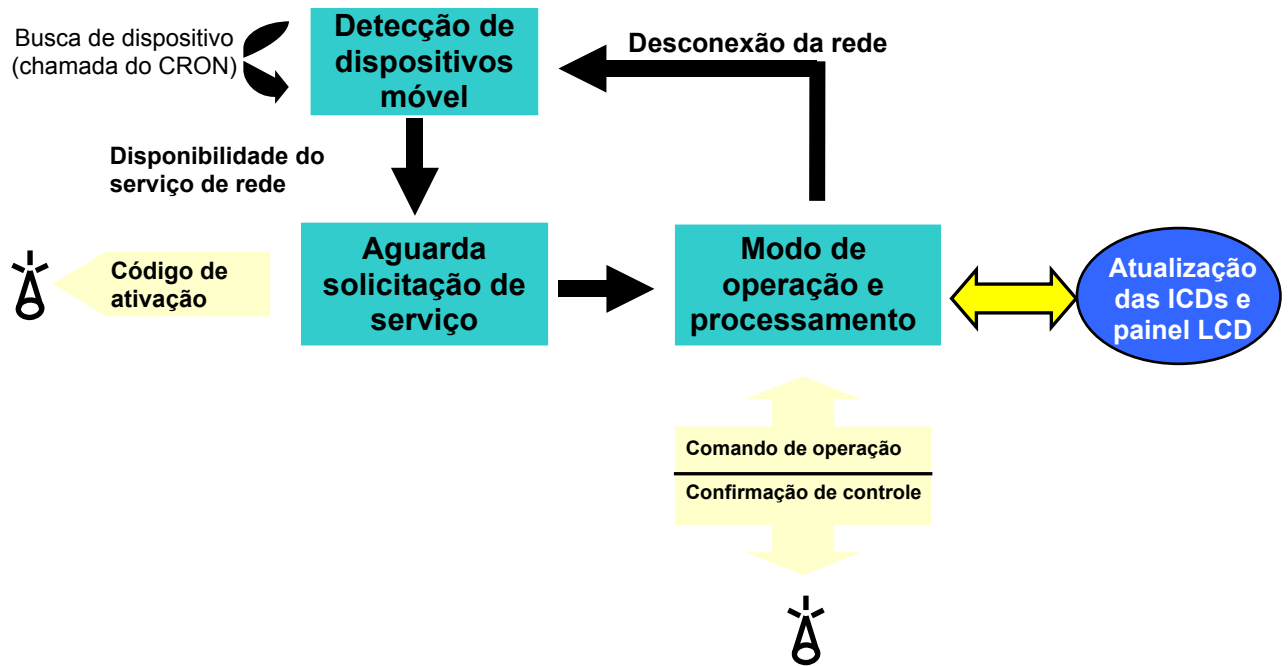


Figura 19 – Fluxograma de controle da comunicação remota via *Bluetooth*

Capítulo 4 – Futuras Implementações

Soluções alternativas em *hardware* para implantação do módulo central podem ser pensadas, dependendo do porte da residência, para substituição do computador padrão IBM/PC (com suas necessidade de refrigeração, espaço físico e consumo de energia elétrica). Nesse contexto, enquadrando-se dentro dos conceitos de *retrofitting* (seção 2.4), soluções com Sistemas Embarcados são alternativas para implementação, como por exemplo o PC/104, computador ITX, ou mesmo dispositivos baseados em microcontroladores com tecnologia ARM apresentados no anexo A.

Além das aplicações sugeridas nesse trabalho, o Módulo Central de Controle deverá permitir a implantação e controle de novos recursos que virão a ser incluídos numa residência como *Set-Top-Box* para o sistema de televisão digital, bem como sistema de áudio e vídeo da residência, aproveitando-se da capacidade de armazenamento de dados que o MCC tem.

Por fim surge a necessidade de implementação de uma interface que permita ao usuário configurar certos parâmetros do módulo central com maior facilidade (evitando que esse seja sistema inacessível no estilo caixa preta), na qual o proprietário da residência possa fazer customização sem a necessidade de um técnico especializado. Como exemplo poderia ser uma interface via *Browser*, acessada via *intranet*, utilizando-se de um servidor HTTP com o Apache. Essa mesma aplicação pode ainda evoluir para se prover controle residencial via *internet*, havendo nesse último caso a necessidade de discussões quanto às questões de segurança.

Capítulo 5 – Conclusão

O conceito de automação residencial tem se mostrado cada vez mais presente na sociedade nos dias de hoje. O envelhecimento da população de modo geral traz à necessidade de soluções que garantam assistência e acessibilidade a essas dentro de uma residência. Outro aspecto atual e que é tema de grande debate é economia e racionalização de energia e formas de economizá-la, como no caso de uma residência inteligente.

No âmbito de um sistema de automação residencial o Módulo Central de Controle traz a facilidade para controlar e conectar diferentes tipos de dispositivos de maneira inteligente e com sinergia, a facilidade de implementação (funcionando como *gateway* do sistema), e traz também a facilidade para o suporte de novas tecnologias e meios de controle.

A implantação do Módulo Central de Controle sobre a plataforma de um computador se mostrou adequada uma vez que essa reúne todas as necessidades de *hardware* (Serial RS-232, USB, armazenamento de dados etc.), além de funcionar de plataforma para desenvolvimento e testes. O sistema operacional mostrou-se confiável para execução da aplicação e não apresentou nenhum problema ou dificuldade para implantação dos protocolos de comunicação desejados.

Algumas dificuldades foram encontradas para garantir uma operação em tempo real do sistema como um todo, o que leva à necessidade de melhorias (aumento da velocidade) na atualização dos parâmetros do sistema e no tratamento e envio de dados, tais como aumento da largura de banda, aumento do *buffer* de telas entre outras medidas.

O desenvolvimento do sistema como um todo mostrou-se viável tecnologicamente e financeiramente apesar das dificuldades. Nesse sentido, a tecnologia de Sistemas de Automação Residencial mostra-se convergente para a política governamental de incentivo no segmento de desenvolvimento de *software*. Essa área apresenta-se como um grande nicho de mercado para incubação de pequenas empresas, com forte caráter de desenvolvimento e pesquisa e para exportação de produtos.

Anexos

A. Sistemas embarcados para o hardware do sistema central

A.1 A arquitetura ARM

A arquitetura ARM (*Advanced RISC Machine*) foi desenvolvida pela *Acorn Computers Limited* de Cambridge, Inglaterra, entre outubro de 1983 e abril de 1985, e foi o primeiro processador desenvolvido para uso comercial. O projeto foi baseado no processador Berkeley RISC I.

A arquitetura ARM é uma arquitetura de processador de 32 bits, e é usada principalmente em sistemas embarcados (*Embedded System*). Embora operem a frequências relativamente baixas, se comparados aos processadores x86 (na maioria dos casos apenas 300, 400 ou 500 MHz), os chips ARM são de baixo custo e de baixo consumo elétrico, por isso são extremamente populares em celulares, PDAs, pontos de acesso, *modems* ADSL, centrais telefônicas, sistemas de automatização em geral entre outros. Cerca de 75% de todos os processadores de 32 bits usados em sistemas embarcados são processadores ARM[12] (figura 20).

Um exemplo é a possibilidade de usar um processador ARM e 4MBytes de memória para executar um sistema embarcado GNU/Linux, o que abre grandes possibilidades em termos de suporte a *drivers*, conectividade de rede, facilidades de *softwares* e ferramentas de desenvolvimento. Com um pouco mais de memória é possível também executar o *Windows Mobile* ou o sistema Operacional Symbian.



Figura 20 - Microprocessador ARM [19]

Existem diversos modelos de processadores baseados na arquitetura ARM, com dezenas de fabricantes no mercado. Alguns modelos possuem inclusive interfaces internas como controladores USB, I2C, memória flash interna entre outros, classificados então como microcontroladores.

Características:

- Arquitetura *Load-Store*, na qual as instruções (soma, subtração, etc) somente processarão valores que estiverem nos registradores e sempre armazenarão os resultados em algum registrador;
- Instruções fixas de 32 bits (com exceção das instruções *Thumb* compactas de 16 bits) alinhadas em 4 bytes consecutivos da memória, com execução condicional, com poderosas instruções de carga e armazenamento de múltiplos registradores, capacidade de executar operações de deslocamento e na ULA com uma única instrução executada em um ciclo de máquina;
- Formato de instruções de 3 endereços (isto é, os dois registradores para operandos e o registrador de resultado são independentemente especificados);
- 15 registradores de 32 bits para uso geral;
- Manipulação de periféricos de I/O como dispositivos mapeados na memória, com suporte a interrupções;
- Conjunto de instruções aberto a extensões através de co-processador, incluindo a adição de novos registradores e tipos de dados ao modelo do programador;
- *Pipelines* de 3 e 5 estágios;
- Interfaces como USB e CAN, agregados no próprio chip, dependendo do modelo e fabricante.

A.2 A arquitetura PC/104

O PC/104 (figura 21) é um padrão de computador para sistemas embarcados.

A primeira versão do PC/104 foi criada em 1987, porém até 1992 o projeto estagnou. Em 1992, o IEEE iniciou a padronização do “Projeto PC/104” reduzindo o tamanho da placa-mãe a partir das especificações já existentes do grupo IEEE 996 (especificação do modelo PC/AT/ISA), com intuito de criar uma nova especificação para aplicação em sistemas embarcados. As especificações do projeto PC/104 serviram então, para o rascunho da especificação IEEE 996.1 que é o padrão para *Compact Embedde-PC Module*.

Desde então a arquitetura evoluiu em três grandes alterações conforme tabela 2:

Tabela 2 – Evolução da arquitetura PC/104

Modelo	Barramento	Ano do modelo
PC/104	ISA (padrão AT ou XT)	1992
PC/104-Plus	ISA ou PCI	1997
PCI-104	PCI	2003



Figura 21 – Placa PC/104 [19]

O Padrão PC/104 a partir da versão *plus* permite que até cinco módulos (placas) sejam empilhadas (acrescentadas) ao sistema (assim como se acrescenta placas a um computador convencional), permitindo uma expansão do sistema. Para isso é necessário uma placa principal responsável pelo controle do barramento (*host board*). Atualmente, a arquitetura é disponibilizada no mercado por mais de 200 diferentes fabricantes, os mais variados tipos de placas como módulos CPUs, controladores de vídeo, interfaces de rede, placas de entrada e saída de som, placas de aquisição de dados e outras interfaces para aplicações específicas.

As especificações elétricas do sistema são similares à de um computador convencional, porém, com baixo consumo que não passa de 39 Watts por módulo. Periféricos de expansão podem operar com alimentação proveniente do próprio barramento, com consumo de até 2Watts.

As principais diferenças entre o PC/104 e o padrão convencional de PC (IEEE P996) são:

- **Área reduzida do sistema:** a área da placa do sistema é de 90.17×95.89 mm;
- **Facilidade de expansão:** o padrão PC/104 permite que sejam empilhados outros módulos (placas) do sistema com novos periféricos trazendo novas funcionalidade, conforme (figura 22)[13];

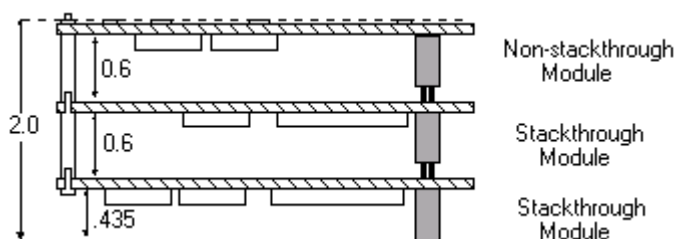


Figura 22 – Capacidade de empilhamento de placas da arquitetura PC/104 [13]

A.3 A arquitetura mini-ITX

As placas mãe mini-ITX (figura 22) foram desenvolvidas pela VIA Technologies Inc. a fim de se tornar uma alternativa de custo menor para placas de sistemas embarcados. Medindo 170 mm x 170 mm a Mini-ITX é a menor placa mãe para padrões X86 existente. Possui total compatibilidade com os sistemas operacionais GNU/Linux e Windows.

O padrão mini-ITX é largamente usado na indústria, em sistemas embarcados, em equipamentos de comunicação de dados e equipamentos médicos entre outros.

As placas mãe mini-ITX possuem diversos tipos de periféricos *onboard*, da mesma forma que alguns modelos de computador convencional, já vindo equipada com o processador x86, barramento PCI, saídas VGA/TV, áudio, USB, serial, *Ethernet* e leitor de *Compact Flash* (Figura 23).

O sistema funciona com processadores x86 com baixo consumo de potência, permitindo que o sistema opere inclusive sem a necessidade de ventiladores exaustores.



Figura 23 – Placa PC mini-ITX [14]

B. Redes de conexão de dispositivos

B.1 Rede Baseada no sistema de distribuição de Energia Elétrica

Redes de transmissão de dados baseadas no mesmo meio físico da distribuição de energia elétrica são comumente chamadas de redes *powerline*. Esse tipo de rede permite que se utilize a fiação elétrica já existente na residência como meio de transporte de pacotes de dados. Dessa forma, permite a implementação de uma rede de serviços envolvendo os dispositivos da casa, sem a necessidade de se acionar uma nova infra-estrutura de cabeamento, que em residências já construídas traria, muitas vezes, a necessidade de adaptações físicas (obras civis) o que pode se tornar inviável.

Dificuldades técnicas da transmissão por rede elétrica:

A rede elétrica, devido à sua natureza, não foi idealizada para se transmitir dados, mas sim otimizada para a transmissão de energia elétrica em frequências típicas de 50 ou 60 Hz. Deste modo, as redes de distribuição de energia são meios inapropriados para a passagem de sinais de alta frequência. Outros fatores também dificultam a utilização da rede elétrica como meio de transmissão, tais como a própria topologia da rede com várias ramificações, as características físicas dos condutores, bem como os tipos de equipamento ligados à rede elétrica. A seguir são listados os principais obstáculos ao uso da rede elétrica:

- **Ruído:** A rede elétrica é normalmente montada sem blindagem e assim fica susceptível a inúmeras fontes de ruído;
- **Descontinuidade de impedância:** ao longo da rede, independentemente do uso de tomadas, criam-se pontos de reflexão de sinal e interferência espectral.
- **Carga:** Todos os eletrodomésticos conectados nas tomadas, mesmo que desligados, constituem uma carga resistiva e reativa dissipando a energia transmitida em alta frequência [1].

B.1.1 Protocolo CEBus & Home PnP

O protocolo *Consumer Electronics Bus* (CEBus) é um padrão aberto que especifica uma arquitetura de rede baseada no modelo OSI (ver anexo A), e tem como principal objetivo a interconexão em rede de dispositivos utilizando a rede elétrica. Foi desenvolvido pelo *Electronic Industries Alliance* (EIA) e pode operar em diferentes meios.

O CEBus sobre a rede elétrica utiliza a tecnologia de espalhamento espectral inicializando a modulação numa determinada frequência, alterando-a durante o ciclo. Ao invés de utilizar os sistemas de *frequency hopping* ou *direct sequence* (ver apêndice A),

amplamente adotados em sistemas que utilizam espalhamento espectral, o CEBus *Powerline Carrier* varia a frequência de transmissão a cada bit transmitido. A transmissão é inicializada com pulsos de 100kHz, aumentando linearmente até 400 kHz durante um intervalo de tempo de 100 microssegundos. Ambos os pulsos criam dígitos similares, sendo desnecessária a pausa entre eles.

O dígito 1 é criado num estágio inferior ou superior com duração de 100 microssegundos, e o dígito 0 é criado num estágio inferior ou superior com duração de 200 microssegundos. Conseqüentemente a taxa de transmissão é variável e depende do número de caracteres 1's e 0's, e se for considerada uma média idêntica para cada um dos dígitos, a taxa seria de 7500bps.

Para evitar colisões de dados, utiliza-se o protocolo *Carrier Sense Multiple Access/ Collision Detection and Resolution* (CSMA/CD), impondo ao nó que aguarde o meio estar livre para enviar o pacote.

Cada pacote transmitido contém o endereço do remetente e do destinatário. As transmissões CEBus são baseadas em *string* ou pacotes de dados que podem variar de tamanho dependendo da quantidade de dados incluídos. O tamanho mínimo do pacote é de 64 bits, e se considerarmos uma média da taxa de transmissão em 7500bps, esse demoraria cerca de 9ms para ser transmitido.

B.1.2 Protocolo X-10

O X-10 é um protocolo de comunicação remoto designado para comunicação entre transmissores e receptores X-10 através da fiação de rede elétrica convencional (*powerline*).

O sistema foi inicialmente introduzido em 1978 pela *Sears Home Control System* e pela *RadioShack Plug and Power System*. Devido ao grande sucesso e robustez, várias outras empresas como a General Electric, Philips, RCA e Leviton, começaram a produzir versões compatíveis. Mas, com o intuito de manter um padrão entre todos os sistemas denominados X-10, alguns módulos foram patenteados e suas licenças oferecidas.

No sistema X-10, os transmissores enviam comando tais como *turn on*, *turn off* ou *dim*, precedidos pela identificação da unidade receptora a ser controlada[1]. Esses comandos são enviados por *broadcast* que entra na rede elétrica da residência através de um ponto de acesso ou uma tomada comum. Cada receptor está relacionado com uma identificação de unidade e só reage aos comandos que lhe são endereçados. O X-10 está especificado para um total de 256 endereços diferentes. São 16 códigos de unidade (1-16) combinados com 16 códigos de compartimentos ou setores (A-P) (figura 24). Deste modo, não existem restrições ao usar múltiplos transmissores sobre a mesma rede, sendo que cada um usará num código de setor. Podem ser encontrados problemas durante as

comunicações, no caso da área de rede suportada ser superior a 185 metros quadrados. Este problema pode ser solucionado com a utilização de uma *bridge* ou de um amplificador de sinal.

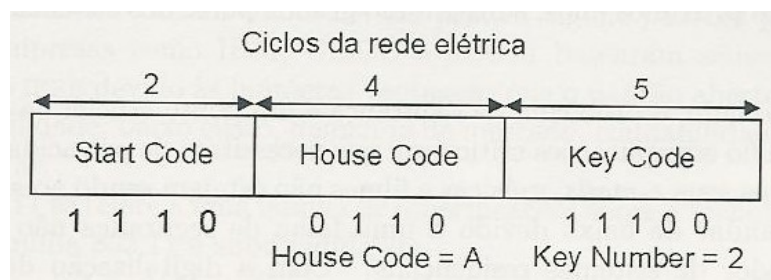


Figura 24 – Ciclos do protocolo X-10 na linha de energia, esquema de identificação setores [1]

Os pacotes X-10 são enviados quando o nível de tensão AC passa pelo valor zero, conforme figura 25. O “um” binário indica a presença de 1ms de um *burst* (rajada) de sinal de 120 Hz no ponto zero AC. O “zero” binário indica a falta deste *burst*. A situação ideal é transmitir o sinal o mais próximo possível do zero AC, ou no máximo 200 microsegundos deste ponto. Estes *bursts* devem ser aplicados três vezes para coincidir com a passagem por zero das três fases que compõe o sistema de transmissão de energia.

A transmissão completa do código engloba 11 ciclos da linha de energia ou 22 dígitos binários. Na verdade, o código é enviado duas vezes totalizando 44 bits transmitidos para cada comando. Os primeiros quatro ciclos são os códigos de início, *Start Code*, 1110. Os quatro seguintes correspondem ao *House Code* e os últimos cinco são os códigos da aplicação /unidade, chamados de *Key Code* [1].

São inúmeros os dispositivos compatíveis com a tecnologia X-10: equipamentos de teste, filtros anti-ruídos, transmissores, receptores, controladores de rede, temporizadores, sensores e detectores diversos. Atualmente tem sido construídas interfaces controladoras dos X-10, com interfaceamento serial RS-232, o que permite a interconexão com a interface de controle de dispositivos.

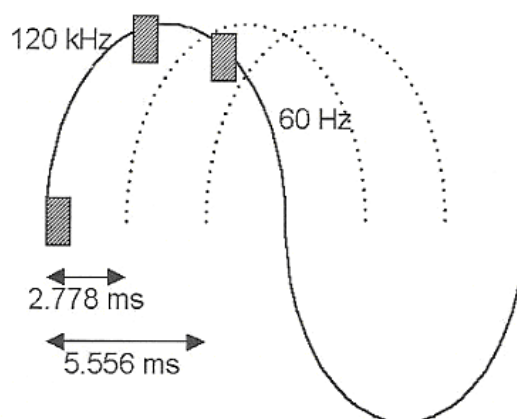


Figura 25 – Pacotes X-10 [1]

B.2 Controle de Periféricos por cabos dedicados

B.2.1 O Protocolo USB

Idealizado em 1995 por um grupo de empresas de tecnologia, o padrão USB permite a conexão de até 127 equipamentos por controlador *host*. Em sua primeira versão a 1.0 e em seguida a 1.1, permitia velocidades de transmissão de 1.5 e 12 Mbps respectivamente, tudo isso sem a necessidade de desligar os equipamentos, e o *host* para conexão com reconhecimento automático dos aparelhos adicionados, conhecido com *hot plug-in*. Anos mais tarde surgiu a versão 2.0 com total compatibilidade com as versões anteriores, a mesma capacidade de gerenciamento de dispositivos, mas velocidade de 480Mbps. [1][14]

B.2.2 O Protocolo FireWire (IEEE 1394)

O *FireWire*, criado pela Apple no início da década de 90, foi adaptado em 1995 e padronizado pela norma IEEE 1394. Sua Capacidade de comunicação pode atingir até 30 vezes a velocidade do USB 1.0, podendo conectar dispositivos como *drivers* de disco, câmeras digitais, televisão digital, computadores, etc. Comparando com o USB 2.0, o IEEE 1394 apresenta praticamente a mesma velocidade, tendo um custo superior devido à implantação de uma complexo protocolo e da taxa de sinalização.

O protocolo não é exclusivo para computadores, nem existe a necessidade de um para que dois dispositivos IEEE1394 possam funcionar. Muitos equipamentos de áudio e vídeo estão sendo produzidos com interfaces IEEE1394, permitindo a transmissão de sinal de um para outro na forma digital, removendo as diversas etapas de conversão analógico-digital, aumentando substancialmente a qualidade da imagem e do som (figura 26).

A utilização do IEEE 1394 com protocolo para construção das redes domésticas é bastante promissora, principalmente no que diz respeito à união de computadores, equipamentos de áudio e vídeo e eletrodomésticos que demandam alta taxa de transmissão de dados, como é o caso de aplicações multimídia.

Como no protocolo USB, não é necessário inicializar a máquina para detectar os dispositivos *FireWire*, já que isso acontece no ato da conexão física. Os produtos atuais podem operar a taxas de 400 Mbps contra 12 Mbps do USB versão 1.0. Mesmo se observarmos a versão USB 2.0 que alcança 480 Mbps, o *Firewire2* (IEEE1394b), na nova revisão do protocolo, deverá atingir em breve velocidade entre 800 e 3200 Mbps.

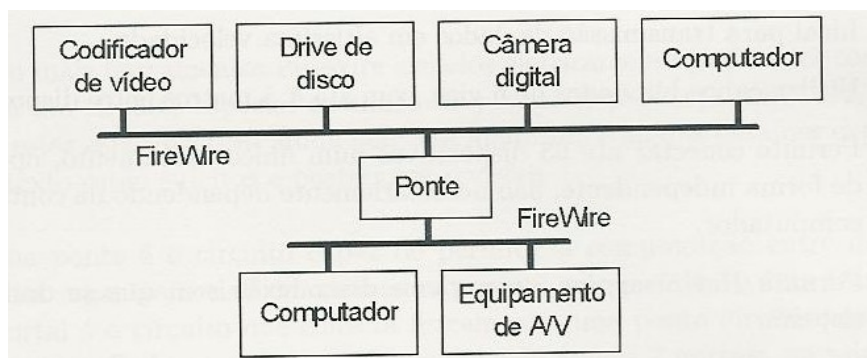


Figura 26 – Rede FireWire para conexão de dispositivos multimídia [1]

Características:

- Dados Digitais transmitidos em formato serial;
- Taxa de transmissão de 100, 200 e 400 Mbps;
- Ideal para transmissão de dados em altíssima velocidade;
- Utiliza cabos blindados de 6 vias, com 4,5 metros entre dispositivos;
- Permite conectar até 63 dispositivos num único barramento, operando de forma independente, não necessariamente dependendo de controle do computador;
- Permite *Hot Swapping* – permitindo a conexão e desconexão sem que seja necessário desligar o sistema.

Tabela 3 – Comparativo entre o protocolo USB e FireWire [1]

Padrão	USB 1.0	USB 2.0	IEEE 1394a	IEEE 1394b
Status	Disponível	Disponível	Disponível	Em planejamento
Máxima taxa (Mbps)	1,5-12	480	400	800 a 3200
Máximo número de dispositivos conectados	127	127	63	63
Precisa de host	Sim	Sim	Não	Não
Digital	Sim	Sim	Sim	Sim
Hot plugable	Sim	Sim	Sim	Sim
Timing	Isócrono	Isócrono	Assíncrono ou Isócrono	Assíncrono ou Isócrono

B.2.3 O Protocolo CAN

O CAN (*Controller Área Network*) é um protocolo de comunicação serial, que foi originalmente desenvolvido pela Bosch, para indústria automobilística, no início dos anos oitenta. A intenção, na época, era desenvolver um barramento de dados com grande imunidade às interferências elétricas, e habilidade em detectar erros, oferecendo ao mesmo tempo uma grande taxa de transmissão.

O CAN permite que controladores, sensores e atuadores possam se comunicar entre si, em tempo real, em até 1Mbps utilizando somente um par trançado com linha de dados serial.

Devido ao sucesso na indústria automobilística, de automação industrial e predial, algumas empresas estão desenvolvendo interfaces para utilização na automação residencial. A arquitetura só especifica duas camadas do modelo OSI (apêndice C), a camada física e a de enlace, enquanto que as mais altas são definidas durante a implementação do sistema.

A utilização do sistema CAN torna-se uma alternativa econômica de rede dados para aplicações residenciais interligando sensores, atuadores e todo o controle em tempo real de dados através de par trançado, fibra óptica ou cabo coaxial. Nos meios físicos de cobre, o protocolo se mostrou resistente à interferência eletromagnética e com uma surpreendente detecção de erros e falhas. Devido ao aumento de popularidade da arquitetura, vários fabricantes passaram a produzir dispositivos CAN e, como resultado os preços para implementação do protocolo estão sendo reduzidos. A figura 27 apresenta um diagrama de uma rede residencial utilizando o CAN.

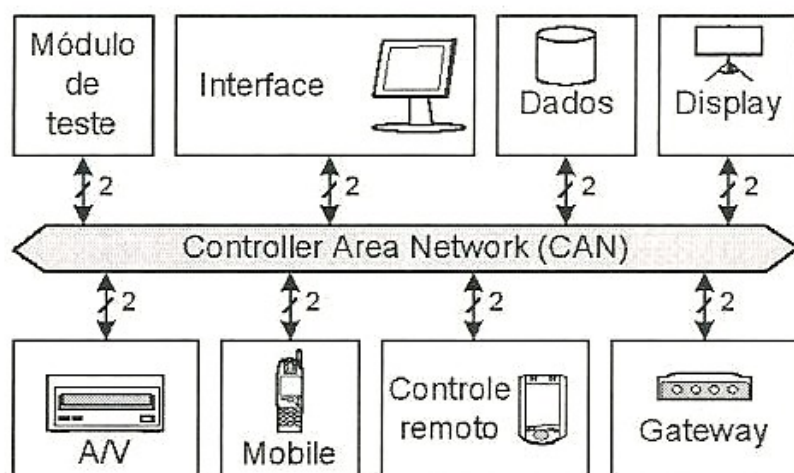


Figura 27 – Rede de dispositivos residenciais montada sobre protocolo CAN [1]

B.3 Redes de controle sem fio

Nos últimos anos, a comunicação sem fio ganhou espaço considerável no mercado de transmissão de dados, deixando de existir apenas nas comunicações de longa distância, para fazer parte de ambientes locais (residenciais).

Há uma tendência moderna de se implantar cada vez mais as redes sem fio em ambientes residenciais, motivada pela facilidade da instalação e, muitas vezes, pela inviabilidade do uso de redes cabeadas em projetos de *retrofitting*. Outros fatores relacionam-se com a mobilidade e flexibilidade que as comunicações sem fio oferecem e pela redução do custo dos equipamentos.

No ambiente residencial, muitas soluções proprietárias têm sido usadas para possibilitar a comunicação entre dispositivos inteligentes com o propósito de comando

remoto, eliminando os fios e tornando mais flexível e prático o uso desse equipamento. Muitas soluções são na maioria das vezes baseadas em transmissores e receptores na faixa de 433 MHz, e esses, em algumas vezes, não seguem qualquer normalização. A indústria desenvolveu soluções baseadas em redes *Bluetooth* e *HomeRF*, sendo que este último serviu de base para o protocolo *ZigBee* (*Home RF Lite*) que tem se mostrado muito promissor para aplicações em automação residencial.

B.3.1 Protocolo *Bluetooth*

Uma das aplicações mais promissoras para o protocolo *Bluetooth* é na área de automação residencial e na fabricação dos *Internet Appliances*. Na área de automação o *Bluetooth* entra como protocolo principal para comunicação com dispositivos de acesso sem fio, tais como celulares, PDAs, PocketPCs entre outros [1].

B.3.2 *ZigBee*

Devido a atrasos no desenvolvimento do *Bluetooth*, um grupo de empresas lideradas pela Philips começou a desenvolver uma outra solução para as WPANs, cujas principais características são o baixo custo e o baixo consumo de energia. Denominada *ZigBee* é também conhecida por *PURLNet*, *HomeRF Lite*, *Firefly*, *RFEasyLink* ou *RF Lite*. Atualmente opera a uma velocidade compreendida entre 20kbps e 250kbps, o que o torna muito mais lento que *Bluetooth*, porém com alcance maior, entre 1 e 100 metros (teoricamente a versão classe 1 do *Bluetooth* também tem alcance de 100 metros, porém não há bons resultados em testes realizados a esta distância [1]), tornando-o interessante para automação residencial. Outra característica bastante interessante é a do baixo consumo, devido ao fato dos dispositivos poderem ficar um grande período em modo inativo.

B.3.3 Wi-Fi (IEEE 802.11)

O IEEE constituiu um grupo de pesquisa para criar padrões abertos a fim de alavancar o desenvolvimento de rede de computadores utilizando a tecnologia sem fio. Para isso foi criado o IEEE 802.11, que nasceu em 1990 mas ficou inerte por aproximadamente sete anos, devido principalmente à baixa taxa de transferência alcançada na época, algo por volta de dezenas de kbps. Com o desenvolvimento da tecnologia e o aumento da velocidade de transmissão de dados, que ficou mais próximo da velocidade obtida no padrão *Ethernet*, o padrão IEEE 802.11 passou a ser visto como promissor para conexão de rede de computadores.

B.3.4 Infravermelho

O infravermelho é um sistema de transmissão de dados utilizado em sistemas ponto a ponto, ou ponto a multiponto em curta distância. Esse sistema é largamente difundido no uso em controle remoto de diversos aparelhos.

Constituído de um sistema elétrico simples, apresenta baixo custo para ser implementado, porém tem séria limitação para uso já que o sinal é facilmente obstruído por objetos que se interponham entre o emissor e o receptor, não tendo capacidade de atravessar superfícies sólidas como paredes.

O protocolo de comunicação via infravermelho sofre atualmente forte concorrência com padrões mais modernos como *Bluetooth* e *ZigBee*, que apesar de mais complexos para implementação, superam as limitações do infravermelho.

B.3.5 Comparativo da comunicação sem fio

Características de custo, consumo de energia e alcance, tornam o protocolo *ZigBee* atrativo para construção de uma rede de sensores e atuadores de DI (dispositivos inteligentes). Ao passo que, o protocolo Wi-Fi mostra-se interessante para funções de conectividade de dispositivos e recursos que demandam grande largura de banda, independentemente da complexidade de implementação e do consumo de energia, como por exemplo, aplicações multimídia da casa, rede de computadores e outros dispositivos que demandam fluxo considerável de dados. O protocolo *Bluetooth* tem quase se tornado um padrão para dispositivos e acessórios de mão como celulares, PDAs e até mesmo Notebook, os quais, além do Wi-Fi (quase sempre presente) também estão trazendo suporte *Bluetooth*. A figura 28 mostra uma comparação entre as tecnologias, ilustrando aspectos como de alcance, aplicação e taxas de dados.

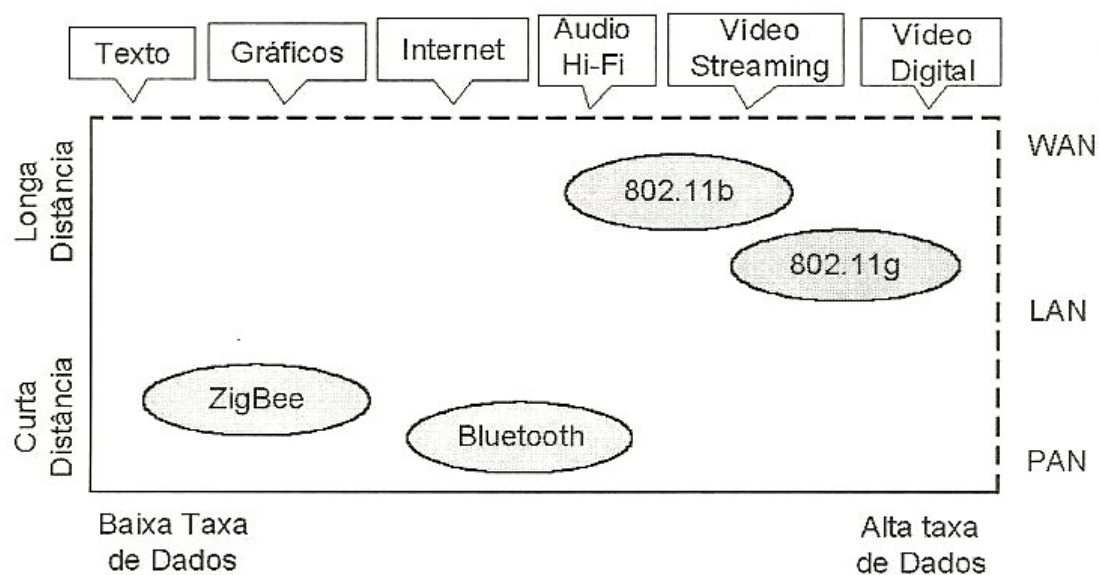


Figura 28 – Funcionalidades dos protocolos sem fio. [1]

Fazendo uma comparação sobre os protocolos de comunicação sem fio pode-se observar que cada um tem características particulares que o torna mais apropriado para um tipo de aplicação específica dentro do sistema de automação residencial. Certamente ao se pensar no ambiente de uma residência inteligente estará presente pelo menos dois desses protocolos.

C. Lista de funções para controle da interface USB em GNU/Linux

Tabela 4 – Lista de funções principais para controle de interface USB em GNU/Linux

Nome da função	Funcionalidade
<code>list_devices()</code>	Lista os dispositivos conectados ao <i>host</i>
<code>find_device(int fabricante, int produto)</code>	Busca por um dispositivo a partir do fabricante e do código do produto
<code>usb_get_busses()</code>	Requisita informações do barramento
<code>usb_control_msg()</code>	Envia dados de controle para o dispositivo especificado.
<code>usb_close()</code>	Encerra a comunicação com um dispositivo

D. Inicialização do dispositivo *Bluetooth* através de interface USB-Serial

A Chamada de códigos a seguir é utilizada para carregar e iniciar uma comunicação *Bluetooth* via interface USB-Serial virtualizada. Essa sequência de inicialização é executada através de um script para inicialização *Bluetooth* chamada através do CRON.

```
# Módulos do Kernel a serem carregados
# Inicialização automática acrescentando as linha em /etc/rc.d/rc.local
root@scar:/$ modprobe bluetooth
root@scar:/$ modprobe rfcomm
root@scar:/$ modprobe hci_usb

# Carregamento da interface de comunicação
root@scar:/$ hciconfig -a hci0 up

# Localização de dispositivos
root@scar:/$ hcitool scan

# Caso algum dispositivo tenha sido localizado
root@scar:/$ rfcomm connect 0 <address> 1
```

Referências Bibliográficas

- [1] **Bolzani, C.A.M.** – “Residências Inteligentes”, 2004;
- [2] **Neto, J.S.C.** – “Edifícios de Alta Tecnologia”, 1994;
- [3] **Bolzani, C.A.M** – “Desenvolvimento de um Simulador de Controle de Dispositivos Residenciais Inteligentes: Uma Introdução aos Sistemas Domóticos”, 2004;
- [4] <http://www.energysavingtrust.org.uk/>, acessado em NOVEMBRO/2007;
- [5] <http://marc.med.virginia.edu/>, acessado em NOVEMBRO/2007;
- [6] **Hela S., Mann W., El-Zabadani H., King J., Kaddoura Y., Jansen E.** - “The Gator Tech Smart House: A Programmable Pervasive Space”, 2005;
- [7] **Anunciação, Heverton Silva** - Linux – Guia prático em português, 1999;
- [8] **Schildt H.** – “C Completo e Total”, 1997;
- [9] **Laureano M.** – “Programando em C: para Linux, Unix e Windows”, 2005;
- [10] **Neves J. C.** – “Programação Shell Linux”, 2006;
- [11] <http://gcc.gnu.org>, acessado em DEZEMBRO/2007.
- [12] **ARM Holdings plc** – “ARM Product Backgrounder”, 2005;
- [13] http://www.pc104.org/technology/reg_info.html, acessado em 2007;
- [14] <http://www.itx.net/ITX-Motherboards.asp>, acessado em 2007;
- [15] **Medeiros, Daniel Berto de** – “Desenvolvimento de uma interface de gerenciamento remoto de dispositivos via *Bluetooth* e acesso telefônico por DTMF para o sistema de automação residencial SCAR”, Trabalho de Conclusão de Curso, Departamento de Engenharia Elétrica – EESC- USP, 2007
- [16] **Fujita, Matheus Nobuochi Shiotsuki** – “Desenvolvimento de um painel de gerenciamento de dispositivos do sistema de automação residencial SCAR, utilizando um LCD gráfico com película *Touchscreen*, operado por microcontroladores da família 8051 e PIC”, Trabalho de Conclusão de Curso, Departamento de Engenharia Elétrica – EESC – USP, 2007
- [17] **Bermudez, Carolina Cristine Colombo** – “Desenvolvimento de uma interface microcontrolada de gerenciamento dos dispositivos do sistema de automação residencial SCAR baseada em PIC, com comunicação com o módulo central via USB”, Trabalho de Conclusão de Curso, Departamento de Engenharia Elétrica – EESC – USP, 2007
- [18] <http://www.techelp.com.br/files/guias/isoosi/>, acessado em NOVEMBRO/2007
- [19] http://pt.wikipedia.org/wiki/Arquitetura_ARM, acessado em NOVEMBRO/2007;
- [20] <http://en.wikipedia.org/wiki/PC104>, acessado em NOVEMBRO/2007;

Apêndices

A. Espalhamento espectral

O princípio das técnicas de modulação usando espalhamento espectral é aumentar a quantidade de bits utilizados para transmitir uma mesma informação, de modo a espalhar o espectro de frequência do sinal. Desta forma, aumenta-se a banda de frequência na qual o sinal é transmitido. Normalmente, utiliza-se um código de espalhamento que é multiplicado pelos bits de informação. Os códigos de espalhamento são muito usados em transmissões digitais, principalmente em WLANs, uma vez que, pelo fato de espalharem o sinal na frequência, proporcionam uma série de vantagens que melhoram consideravelmente a performance de transmissão. Estas vantagens são:

- Imunidade com relação a ruídos e interferências;
- Imunidade a distorções devido ao multipercursos;
- Imunidade a interferências e de desvanecimentos de banda estreita;
- Diversos usuários podem compartilhar a mesma banda de frequência, com baixa interferência;
- Podem ser usados para a criptografiação dos sinais;

A seguir, suas técnicas de espalhamento espectral serão mais detalhadamente analisadas.

DS (*Direct Sequency*)

De acordo com esta técnica de espalhamento, os bits de informação são multiplicados por uma sequência de espalhamento chamada de sequência de Barker de 11 bits que é dada por (1 0 1 1 0 1 1 0 0 0). Desta forma, por exemplo, em um dos modos de DS utilizados, tem-se que cada bit do sinal original passa a ser representado no sinal a ser transmitido por 11 bits. Esta modulação aumenta a banda de frequência ocupada pelo sinal, o que significa o espalhamento deste sinal na frequência.

FH (*Frequency Hopping*)

O método de *Frequency Hopping* baseia-se na rápida variação da frequência de transmissão do sinal em um padrão pseudoaleatório, sendo que cada frequência é utilizada para a transmissão somente durante um intervalo de tempo muito curto, chamado de tempo de permanência (*dwell time*). Este tempo de permanência pode ser ajustado e deve ser menor do que 400 ms. A rápida variação da frequência de transmissão é chamada de saltos de frequência. Estes saltos são determinados por um gerador de números pseudoaleatórios instalado em cada uma das estações do sistema, onde cada número pseudoaleatório corresponde a uma frequência.

B. Sistema operacionais multitarefas

Existem dois tipos de sistemas operativos multitarefa: os de antecipação e os de cooperação.

B.I Multitarefa de antecipação (preemptivo)

Os sistemas operativos com multitarefa de antecipação (ou multitarefa preemptiva) esforçam-se por conseguir uma "democratização" dentro do processador. Os seus *kernels* mantêm em memória um registro com todos os processos em execução. A esse registro dá-se o nome de árvore de processos, em virtude da estrutura de dados interna ser geralmente uma árvore.

B.II Multitarefa de cooperação

A multitarefa de cooperação pouco difere do sistema preemptivo. A única diferença é que, neste caso, há uma preempção explícita, sendo que o programa instrui o *kernel* a fazer o *task switching* em certos pontos do código, e o *kernel* apenas passa ao processo seguinte quando o programa o permite.

C. Modelo de camadas ISSO

A figura 29 ilustra a pilha da camadas OSI.



Figura 29 – Pilha da Camada OSI [18]

Camada Física

A camada 1 compreende as especificações do hardware utilizado na rede (compreendidos em aspectos mecânicos, elétricos e físicos - todos documentados em padrões internacionais). Exemplos: Ethernet 802.3, RS-232, RS-485;

Camada de Enlace

Gerencia o enlace de dados. Responsável pelo acesso lógico ao ambiente físico da rede, como transmissão e reconhecimento de erros.

Camada de Rede

Estabelece uma conexão lógica entre dois pontos, cuidando do tráfego e roteamentos dos dados da rede.

Camada de Transporte

Controla a transferência de dados e transmissões. Protocolos de transporte (TCP) são utilizados nesta camada.

Camada de Sessão

Reconhece os nós da rede local LAN e configura a tabela de endereçamento entre fonte e destino, isto é, estabelece as sessões, no qual o usuário poderá acessar outras máquinas da rede.

Camada de Apresentação

Transfere informações de um *software* de aplicação da camada de sessão para o sistema operacional. Criptografia, conversão entre caracteres ASCII e EBCDIC, compressão e descompressão de dados são algumas funções acumuladas nesta camada.

Camada de Aplicação

É representada pelo usuário final no modelo OSI, selecionando serviços a serem fornecidos pelas camadas inferiores, entre eles, o correio eletrônico, transferência de arquivos, etc[18].