UNIVERSIDADE SAGRADO CORAÇÃO

JOSÉ RAFAEL SANSALONE ZEMINIAN

CONSTRUÇÃO DE UM SISTEMA MÓVEL PARA MELHORIA DA ACESSIBILIDADE DE DEFICIENTES FÍSICOS MOTORES EM SUAS RESIDÊNCIAS

BAURU
2010

UNIVERSIDADE SAGRADO CORAÇÃO

JOSÉ RAFAEL SANSALONE ZEMINIAN

CONSTRUÇÃO DE UM SISTEMA MÓVEL PARA MELHORIA DA ACESSIBILIDADE DE DEFICIENTES FÍSICOS MOTORES EM SUAS RESIDÊNCIAS

Trabalho de conclusão de curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação do Prof.° Ms. Anderson Francisco Talon.

BAURU
2010

JOSÉ RAFAEL SANSALONE ZEMINIAN

CONSTRUÇÃO DE UM SISTEMA MÓVEL PARA MELHORIA DA ACESSIBILIDADE DE DEFICIENTES FÍSICOS MOTORES EM SUAS RESIDÊNCIAS.

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade Sagrado Coração como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Ms. Anderson Francisco Talon.

Banca Examinadora:

_____
Prof. Ms. Anderson Francisco Talon
Universidade Sagrado Coração

_____
Prof. Dr. Elvio Gilberto da Silva
Universidade Sagrado Coração

_____
Prof. Esp. Henrique Pachioni Martins
Universidade Sagrado Coração

Bauru, 8 de Dezembro de 2010.

Resumo

Nos últimos anos, tem-se notado uma preocupação progressiva com as questões de acessibilidade de pessoas idosas e com deficiência física aos espaços, sejam eles de uso público ou não. A criação de sistemas voltados para a automação residencial tem despertado o interesse de muitos profissionais no âmbito comercial, além de se mostrar um aliado bastante forte na adaptação dessas residências às necessidades especiais de seus moradores. Uma casa automatizada também pode ser denominada de casa inteligente, ou seja, quando existe algum tipo de controle interativo com seus dispositivos controlados e monitorados, visando oferecer a acessibilidade, conforto e segurança desejada. Existem várias formas de fazer esse controle. Neste projeto buscamos desenvolver um software para dispositivos móveis, que fica conectado a um Web Service, permitindo assim que o portador de necessidades especiais controle e verifique o estado dos dispositivos controlados. Esses dispositivos podem ser equipamentos eletrônicos ou sensores em portas e janelas. Para cumprir o objetivo do projeto foi necessário construir um circuito digital responsável por controlar os equipamentos. Dessa forma, o Web Service ficou hospedado em um computador recebendo os sinais enviados através do dispositivo móvel e enviando esse sinal a porta paralela, que por sua vez está conectada ao circuito.

Palavras chave: acessibilidade, casa inteligente, circuitos digitais, deficientes físicos, dispositivos móveis.

Abstract

Recently, we can see a gradual concern about accessibility of eld people and of physical deficiency people, this concern can be for public space or not. The development of systems for home automation has attracted many professionals, it shows a very strong partner in home adaptation to special needs. A automatic house can be called an intelligent house, its means, a house where the devices can be controlled and monitored in order to offer accessibility, comfort and security. There are many ways to make this control. This project will develop a software for mobile devices what is connect to an Web Service, allowing the people with special needs control and check the devices status. These devices can be electronic devices or sensors in doors and windows. To achieve the desired goal, it's be necessary to build a digital circuit that is responsible to controlling the equipments. The Web service is hosted in a computer receiving signals by the mobile device and sending these signals to electronic devices or sensors.

Keywords: accessibility, home automation, digital circuit, people with special needs, mobile devices.

Lista de ilustrações

Sumário

1. Introdução

Dentro do universo das deficiências, o portador de deficiência física motora é um dos indivíduos que mais são penalizados pela falta de acessibilidade do espaço urbano e edificado, seja ele público ou privado, pois sua mobilidade muitas vezes depende do uso de cadeira de rodas, e o ambiente construído não se encontra adaptado para garantir seus direitos e sua segurança.

Segundo Carvalho (2001), em vários segmentos da sociedade são notórias as inúmeras dificuldades enfrentadas pelos portadores de deficiências. Sendo que as barreiras, principalmente físicas, encontradas por grande parte destas pessoas estão presentes em suas próprias residências e seguem para áreas públicas e ambientes de trabalho.

O termo deficiência é apresentado pela Associação Brasileira de Normas técnicas (ABNT) através da Norma Brasileira (NBR) 9050:

“Deficiência: Redução, limitação ou inexistência das condições de percepção das características do ambiente ou de mobilidade e de utilização de edificações, espaço, mobiliário, equipamento urbano e elementos, em caráter temporário ou permanente”.

Segundo SANTOS (2004), vários são os ambientes de interação do ser humano, contudo a habitação se apresenta como um dos mais importantes. Neste espaço particular o homem se apropria do espaço, impondo-o às suas necessidades, buscando encontrar sua identidade, fazendo prevalecer seu direito à privacidade e ao convívio familiar.

Ainda segundo SANTOS (2004), devido à importância apresentada por este ambiente, é fundamental que ele atenda às suas necessidades e especialmente, em se tratando de um morador que depende de uma cadeira de rodas, garanta seu uso e deslocamento.

Com o rápido avanço da tecnologia, o que proporcionou a miniaturização e a redução de custo de componentes eletrônicos, tornaram acessíveis a pessoas comuns os benefícios da automação residencial.

As residências automatizadas podem proporcionar conforto e tornar a vida de seus moradores muito mais fácil, no entanto para alguns de seus moradores esses sistemas não são apenas questão de conveniência, são ferramentas indispensáveis no

cotidiano doméstico. Além de ser possível manter as contas de água e luz, por exemplo, sob controle, a automação residencial também se preocupa com os portadores de necessidades especiais já que a tecnologia pode garantir a independência desses deficientes físicos através de sistemas de acessibilidade.

Ainda segundo a NBR 9050, o termo acessibilidade pode ser definido como: “Possibilidade e condição de alcance, percepção e entendimento para utilização, com segurança e autonomia de edificações, espaço, mobiliário, equipamento urbano e elementos.”

Os moradores dessas residências inteligentes poupam tempo e esforço ao delegarem tarefas domésticas do dia-a-dia para seus sistemas de automação. Assim sendo, a casa poderá também gerenciar a segurança de seus habitantes.

Uma residência tradicional pode ser um dos principais pesadelos dos portadores de necessidades especiais. O que para alguns são pequenas inconveniências, para deficientes podem ser grandes obstáculos, assim precisamos pensar em alternativas de fácil acesso para tornar a vida dessas pessoas mais confortável.

Hoje, a qualquer hora e em qualquer lugar, as pessoas sentem a necessidade de estarem informados, seja através de jornais televisivos, escritos ou através de uma simples consulta a páginas da internet. Isso proporcionou um enorme avanço da tecnologia móvel e atualmente tornou-se cotidiano a possibilidade de checar a caixa de e-mail, ler as últimas notícias ou ainda atualizar a agenda de compromissos diretamente com o servidor de sua empresa através do celular no momento em que estiver andando pelas ruas. Sendo assim, a tecnologia móvel teria muito a somar na área de automação residencial.

A residência automatizada poderá proporcionar uma melhora significativa na qualidade de vida dos deficientes considerando suas limitações físicas, informativas e sociais desde que esteja projetada adequadamente de modo a garantir o mínimo necessário para sua segurança e conforto, já que aumentará e muito a segurança, a comodidade, proporcionando assim independência ao seu morador.

1.1. Justificativa

Atualmente diversas técnicas e modelagens estão sendo estudadas para o desenvolvimento de sistemas para automação de residências e de ambientes com o intuito de buscar maior segurança e conforto. Por possuir uma ampla área de aplicabilidade, essas técnicas são estudadas tanto no meio acadêmico como também no meio comercial.

A automação residencial pode propiciar para as pessoas inseridas nestes ambientes diversas facilidades, diminuindo trabalhos repetitivos e rotineiros como acender lâmpadas, ligar aparelhos eletrônicos ou ainda até deixar de se preocupar com a temperatura e iluminação de um ambiente.

Existem casos em que o ambiente totalmente ou quase totalmente automatizado é de grande importância ou até mesmo necessário. Podem-se citar casos de pessoas com necessidades especiais e idosos, que por sua vez podem esquecer ou até não conseguir desligar lâmpadas e a televisão já que possuem certas dificuldades.

O intuito desse trabalho é apresentar uma forma alternativa de desenvolver um ambiente automatizado, utilizando para isso dispositivos móveis (como celulares com acesso a internet), Web Services e a interface de conexão paralela com os diversos dispositivos do ambiente.

1.2. Objetivos

1.2.1. Objetivo Geral

Desenvolver um sistema para dispositivos móveis que poderá proporcionar uma melhor qualidade de vida, incluindo segurança, comodidade e rapidez, às pessoas portadoras de deficiências físicas.

1.2.2. Objetivos Específicos

- Construir um circuito digital que possibilite controlar diversos dispositivos eletrônicos através da comunicação com uma porta paralela de um microcomputador;

- Construir um aplicativo para dispositivos móveis que se comunique através de um Web Service com o circuito criado tendo como intuito proporcionar maior acessibilidade para o controle dos dispositivos.

1.2.3. Organização do trabalho

O capítulo 2 apresenta uma visão geral sobre a linguagem Java e suas subdivisões, o capítulo 3 descreve a utilidade e o funcionamento dos Web Services. Já no capítulo 4 fazemos uma rápida introdução a álgebra booleana necessária ao entendimento dos circuitos lógicos, no capítulo 5 falamos sobre multiplexadores e demultiplexadores, que são largamente utilizados na confecção de circuitos digitais. No capítulo 6 são descritas as etapas dos processos envolvidos no desenvolvimento do projeto, no capítulo 7 são descritos os resultados alcançados com o desenvolvimento do projeto e por fim, no capítulo 8 falamos sobre as considerações finais bem como os possíveis trabalhos futuros a serem realizados.

2. Java

A linguagem Java permite o desenvolvimento de aplicativos para diversas plataformas, desde dispositivos pequenos como telefones celulares, até computadores de grande porte como os mainframes. Como a linguagem, ao longo dos anos, vem sofrendo aprimoramentos e com o conseqüente aumento do número de bibliotecas disponíveis para utilização, foram criadas três divisões, também chamadas de ambientes de desenvolvimento, na plataforma a partir da versão 2 da linguagem.

- J2SE (Java 2 Standard Edition): É o ambiente mais utilizado, destinado ao desenvolvimento de aplicativos para desktop e estações de trabalho (Developer Resources for Java Technology).
- J2EE (Java 2 Enterprise Edition): É a versão destinada ao desenvolvimento de grandes aplicações, voltadas para redes, Internet. Assim, ela contém bibliotecas especialmente desenvolvidas para o acesso a servidores, sistemas de e-mail e banco de dados (Developer Resources for Java Technology).
- J2ME (Java 2 Micro Edition): Versão destinada ao desenvolvimento para dispositivos com pequena capacidade de memória e processamento. Essa plataforma contém conFigurações e bibliotecas trabalhadas especialmente para a atuação nesses dispositivos (Developer Resources for Java Technology).

Podemos observar essas subdivisões na Figura 2.1.

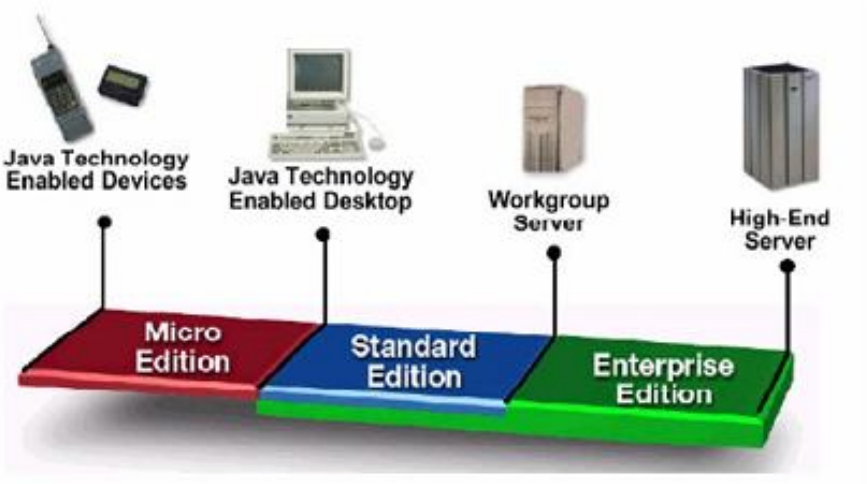

Figura 2-1 - Edições de Pataforma JAVA 2
Fonte: Sun Microsystem
http://java.sun.com (acesso em 08/mar/2010)

Outras definições importantes:

- JVM (Java Virtual Machine): A JVM é responsável por interpretar os arquivos .class (pré-compilados) da linguagem Java. Ela é considerada o coração da linguagem Java, pois simula uma máquina dentro de um computador. Em alguns casos, a JVM também compila alguns trechos do código (just-in-time compiler), a fim de acelerar o processo de execução (Developer Resources for Java Technology, 2010).
- JSDK (Java Software Development Kit): É o ambiente de desenvolvimento da linguagem Java. Ele inclui a JVM, compilador, appletviewer, algumas bibliotecas básicas, entre outras coisas (Developer Resources for Java Technology).
- JRE (Java Runtime Environment): Pacote aconselhado somente para executar as aplicações, contendo nele a JVM e algumas bibliotecas. Não é possível compilar código Java, apenas este pacote (Developer Resources for Java Technology).

Como a JVM pode ser incorporada a aparelhos das redes de geração 2,5 (2,5G), a tecnologia J2ME tornou-se uma ferramenta de desenvolvimento para aplicativos de aparelhos desta geração. Empresas como Motorola, Nokia, Siemens e algumas operadoras telefônicas estão investindo nesta tecnologia.

2.1.1. Estrutura J2ME

Na computação tradicional temos grandes computadores com uma alta capacidade de memória, processamento, display e interfaces ricas com o usuário além de outros diversos aspectos que são considerados padrões. Os pequenos dispositivos como telefones celulares, PDAs não seguem essas características, pois a maioria desses dispositivos possui uma capacidade computacional extremamente reduzida em comparação com os computadores tradicionais. Desta forma não poderíamos utilizar um aplicativo desenvolvido para computadores tradicionais nestes pequenos dispositivos. Justamente por este motivo o J2ME tenta criar um padrão para tais dispositivos tão diferentes entre si, sendo que a SUN, desenvolvedora da tecnologia, unificou essa estrutura com seus outros pacotes, o J2SE e J2EE.

A Figura 2.2 representa uma visão geral dos componentes da tecnologia JME e como ela se relaciona com as demais tecnologias Java.

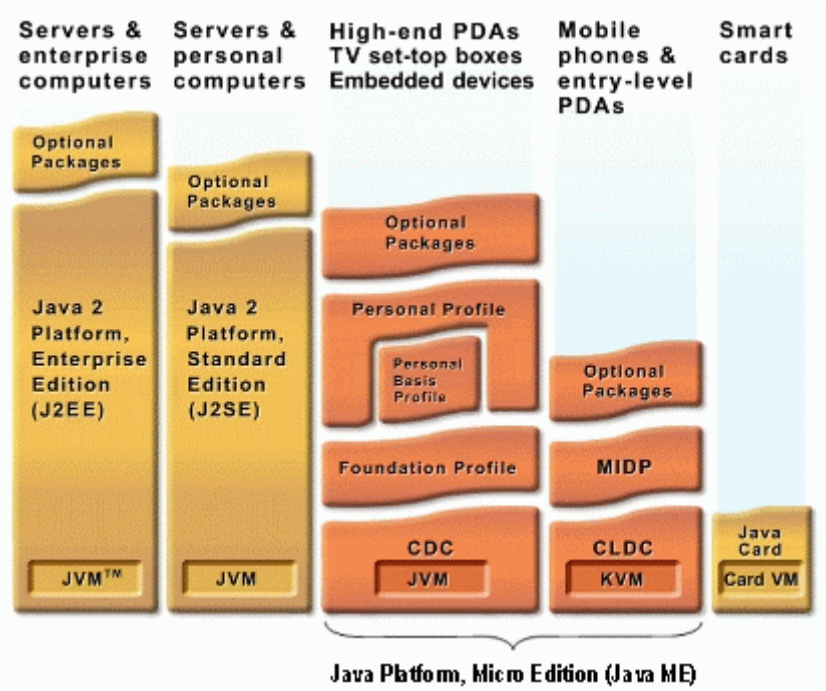

Figura 2-2 - Tecnologias Java
Fonte: Sun Microsystem – Datasheet Java2 Platform, Micro Edition
http://java.sun.com/javame/technology/index.jsp (acesso em 08/mar/2010)

É importante ressaltar que a arquitetura J2ME não substitui o sistema operacional dos dispositivos utilizadores, mas trata-se de camadas acima dele, conforme podemos observar na Figura 2.3.

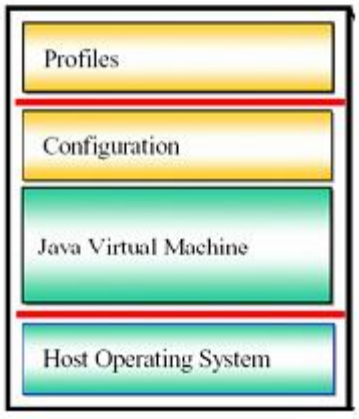

Figura 2-3 - Camadas J2ME
Fonte: Sun Microsystem
http://java.sun.com/products/cldc/wp/KVMwp.pdf (acesso em 08/mar/2010)

A primeira camada acima do sistema operacional é a Java Virtual Machine responsável por interpretar e transformar os códigos fontes da linguagem Java para que ser “entendida” pelo sistema operacional.

A segunda camada são as configurações e tem a função de definir uma classe de Hardware a com qual o aplicativo estará trabalhando. Através dela são definidas as especificações mínimas do ambiente de hardware em que o software será inserido, como por exemplo a interface, memória, processamento entre outras.

Já a terceira camada corresponde aos perfis, que nada mais são que as APIs (Application Program Interface) que complementam e traz novas funcionalidades a camada de configurações.

3. Web Services

Web Services é uma solução utilizada para a integração e comunicação entre sistemas e plataformas diferentes, ou seja, essa tecnologia possibilita que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Os Web Services são componentes que permitem às aplicações enviar e receber dados em diversos formatos, sendo o mais utilizado, o formato XML (Web Services Architecture, 2004).

"Um Web Service é um sistema de software projetado para suportar interações interoperáveis máquina-a-máquina através de uma rede. Ele possui interface descrita em um formato processável pela máquina. Outros sistemas interagem com o Web Service em uma maneira prescrita por sua descrição usando mensagems SOAP, tipicamente transportadas através do protocolo HTTP com uma serialização XML em conjunção com outros padrões relacionados à Web" (Web Services Architecture, 2004).

Para a utilização de um Web Service não é preciso saber nada sobre a plataforma, modelo de objeto, linguagem de programação que foi usada para implementar o serviço, basta saber como solicitar o serviço para que possam receber a resposta da solicitação adequadamente.

3.1.1. Arquitetura de Web Services

"A arquitetura de Web Services é uma arquitetura de interoperabilidade: ela identifica aqueles elementos globais da rede de serviços Web os quais são necessários para assegurar a interoperabilidade entre Web Services" (Web Services Architecture, 2004).

Segundo KREGER (2001), a arquitetura de um Web Service é baseada na interação de três personagens: Provedor de serviços, Consumidor de serviços e o Registro dos serviços, sendo que a interação destes personagens envolve as operações de publicação, pesquisa e ligação.

O provedor de serviços é a entidade que cria o Web Service disponibilizando o serviço para que alguém possa utilizá-lo, porém para isso é necessário descrever o Web Service em um formato padrão que seja compreensível para qualquer um.

O Consumidor de serviços pode ser “qualquer” um que utiliza o Web Service criado por um provedor de serviços. Este conhece a funcionalidade do Web Service, a partir da descrição disponibilizada pelo provedor de serviços, recuperando seus detalhes através de uma pesquisa sobre o registro publicado.

Ainda temos o Registro dos serviços que é a localização central onde o provedor de serviços pode relacionar seus Web Services, e no qual o consumidor de serviços pode pesquisá-los.

Para o funcionamento de um Web Service é necessário existir, no mínimo, dois agentes trocando informações (um solicitando o serviço e outro provendo o serviço desejado). Existem também agentes de descobrimento de Web Services disponíveis na Web, que funcionam como verdadeiros catálogos de serviços disponíveis (Web Services Architecture, 2004).

Ao solicitar o serviço, o solicitante interage com o provedor de serviços, que possui a aplicação propriamente dita. Eventualmente, o solicitante poderá procurar informações em uma das agências de descobrimento de serviços. A partir daí, o provedor de serviços disponibiliza ao cliente somente a descrição dos serviços e o serviço propriamente dito, conforme observamos na Figura 3.1.

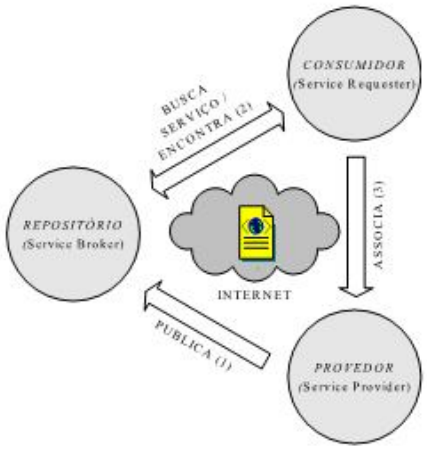

Figura 3-1 - Arquitetura básica Web Service
Fonte: http://www.inf.ufsc.br/~bertoni/artigos/dissertacao.pdf (acesso em 20/03/2010)

Segundo GLASS (2000), ”As aplicações do futuro serão construídas a partir de Web Services dinamicamente selecionados em tempo de execução, baseados no custo, qualidade e disponibilidade”.

4. Álgebra Booleana

Para LOURENÇO *et. al. (2007, p. 37)* a álgebra booleana classifica as informações em dois tipos: verdadeiras na qual se atribui o símbolo matemático 1 (um) e falsas na qual o símbolo é o 0 (zero), facilitando assim o manuseio matemático da informação. Além disso a álgebra booleana tem como base três operações que são: AND, OR e NOT, das quais derivam várias outras. A partir dessas operações é possível desenvolver simples circuitos eletrônicos até o mais avançado computador.

Para desenvolvermos um circuito digital, é necessário ainda conhecer três assuntos fundamentais que são as variáveis lógicas, tabela-verdade e níveis lógicos.

4.1.1. Variáveis Lógicas

Segundo LOURENÇO *et. al.* (2007), as variáveis lógicas são normalmente representadas por letras e seu uso permite escrever expressões algébricas, podendo assumir apenas os valores 1 (um) ou 0 (zero). Normalmente atribui-se o valor 1 (um) às variáveis quando representam elementos ativos e o valor 0 (zero) para situações inversas.

4.1.2. Tabela Verdade

LOURENÇO *et. al.* (2007) diz que em alguns casos, as funções lógicas são extremamente complexas e de difícil análise, sendo possível assim utilizar a tabela-verdade, que é uma representação em forma de tabela das funções lógicas, que facilita a representação e a análise das mesmas.

Utilizando o exemplo da Figura 4.1, sendo S uma lâmpada e A e B duas chaves ligadas em série, pode-se observar que a lâmpada só acenderá se as duas chaves estiverem fechadas, ou seja, A = 1 e B = 1 (representada pela última linha da tabela).

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Figura 4-1 - Exemplo tabela-verdade
Fonte: LOURENÇO *et. al.* 2007, p. 42

4.1.3. Níveis Lógicos

No sistema binário, os números 0 (zero) e 1 (um) representam quantidade e na lógica representam uma qualidade ou situação que pode ser representada através de dois níveis lógicos distintos conforme Figura 4.2 e Figura 4.3. Pode-se assim perceber a grande vantagem de trabalhar com circuitos lógicos, já que são utilizados apenas dois níveis lógicos, o que permite a construção de circuitos mais simples e confiáveis. (LOURENÇO *et. al.* 2007)

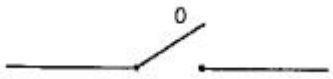

Figura 4-2 - Nível Lógico 0, representado por uma chave aberta
Fonte: LOURENÇO *et. al.* 2007, p. 43

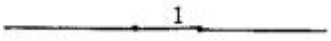

Figura 4-3 - Nível Lógico 1, representado por uma chave fechada
Fonte: LOURENÇO *et. al.* 2007, p. 45

5. Circuitos Combinacionais

De acordo com LOURENÇO *et. al.* (2007), um circuito combinacional é constituído por um conjunto de portas lógicas que determinam os valores das saídas a partir dos valores atuais das entradas, sendo um subsistema digital, ou seja, é uma pequena parte de um sistema maior e mais complexo.

Ainda segundo LOURENÇO *et. al.* (2007), um circuito combinacional "é aquele que executa uma expressão booleana através da interligação das várias portas lógicas existentes, sendo que as saídas dependem única e exclusivamente das entradas", conforme podemos observar na Figura 5.1.

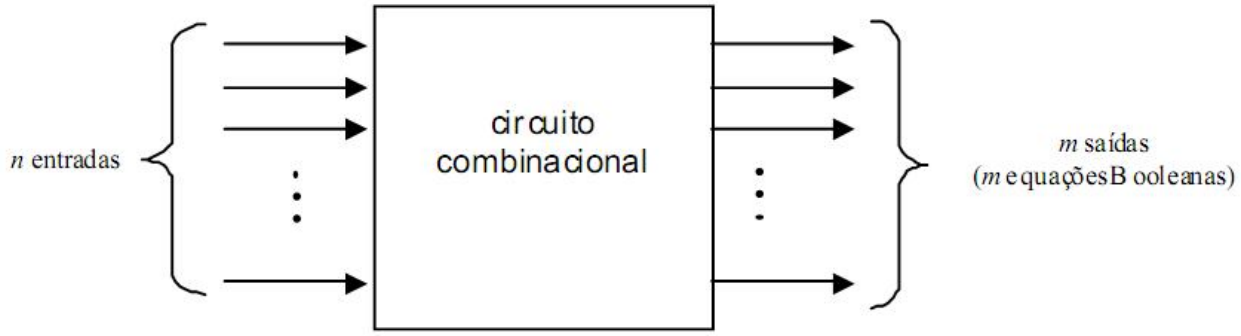

Figura 5-1 - Diagrama genérico de um circuito combinacional.
Fonte: LOURENÇO *et. al.* 2007

No caso, cada combinação de valores na entrada pode ser vista com uma informação diferente e cada conjunto de valores de saída representa o resultado da operação realizada pelas portas lógicas contidas no circuito.

Podermos citar como exemplos mais comuns de circuitos combinacionais os codificadores, decodificadores, multiplexadores, demultiplexadores, somadores e subtratores.

5.1.1. Multiplexadores

O multiplexador ou MUTEX (como é mais conhecido) é um componente eletrônico que codifica as informações de duas ou mais fontes de dados através de suas entradas em apenas um canal de saída, sendo essa operação denominada multiplex (ou multiplexação), que significa seleção.

Segundo LOURENÇO *et. al.* (2007), o multiplexador pode ser definido como: "um circuito combinacional dedicado que tem a finalidade de selecionar, através das

variáveis de seleção, uma de suas entradas, conectando-a eletronicamente à sua única saída".

Podemos observar na imagem 5.2 a representação de um multiplexador que possui *n* entradas.

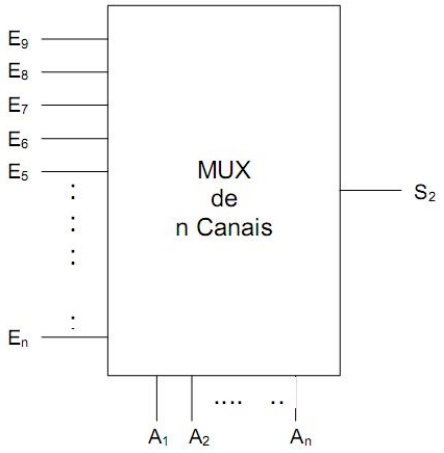

Figura 5-2 - Representação de um multiplexador
Fonte: LOURENÇO *et. al,* 2007

O multiplexador pode assim transmitir os sinais de suas entradas em uma única "linha" que seria sua saída. Como exemplo, se usarmos um multiplexador de quatro entradas, sendo a entrada E1 = 0, a entrada E2 = 1, entrada E3 = 1 e entrada E4 = 0 teríamos a saída S = 0110.

Utilizando as portas lógicas (AND e OR) da álgebra de Boole, temos a representação do multiplexador mais simples que possui 2 canais de entrada conforme a imagem 5.3.

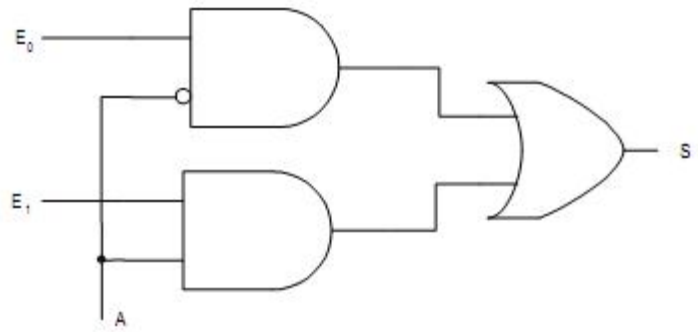

Figura 5-3 - Circuito lógico do multiplexador de dois canais
Fonte: LOURENÇO *et. al,* 2007

No esquema da Figura 5.3 temos as entradas E0 e E1, a variável de seleção A e a saída S.

Podemos encontrar multiplexadores disponíveis comercialmente com 2, 4, 8 ou 16 canais de entradas, porém podemos ainda necessitar de um multiplexador com um número maior de canais de entradas, ou ainda multiplexar vários canais simultaneamente.

Ainda segundo LOURENÇO *et. al.* (2007), a solução para tal problema pode ser facilmente resolvida através da associação desses multiplexadores, na qual podemos associá-los de forma paralela um ao outro, ou ainda sem série dependendo da necessidade.

A associação paralela permite que a capacidade de canais simultâneos de entrada seja ampliada, ou seja, através dela torna-se possível selecionar informações digitais de vários bits simultaneamente.

Neste tipo de associação devemos utilizar um multiplexador com um número de canais de entrada igual ao número de informações a serem multiplexadas, sendo o número de multiplexadores igual ao número de bits destas informações.

O exemplo da Figura 5.4 seria utilizado caso fosse necessário multiplexar quatro informações diferentes, sendo cada uma delas composta de três bits. Teríamos, portanto quatro multiplexadores associados paralelamente, sendo que cada multiplexador possui quatro canais de entrada, um para cada informação e os três bits que compõem a informação seriam distribuídos simultaneamente um para cada multiplexador.

A associação em série entretanto permite que a capacidade de canais de entrada seja ampliada, sendo uma variação da associação paralela, pois para ampliar a capacidade de entradas basta multiplexar os multiplexadores de entrada através de um multiplexador de saída.

No exemplo da Figura 5.5, poderíamos construir um multiplexador de 16 canais através da associação em série de quatro multiplexadores com quatro canais de entradas cada um e também um quinto multiplexador, também com quatro canais de entradas, conectando as saídas dos demais multiplexadores aos seus canais de entrada.

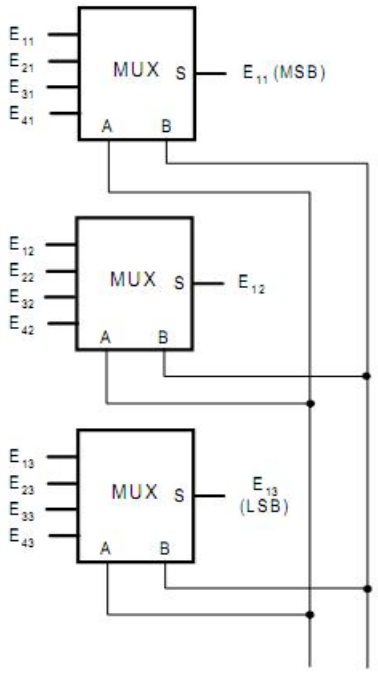

Figura 5-4 - Multiplexação de quatro informações de 3 Bits
Fonte: LOURENÇO *et. al,* 2007

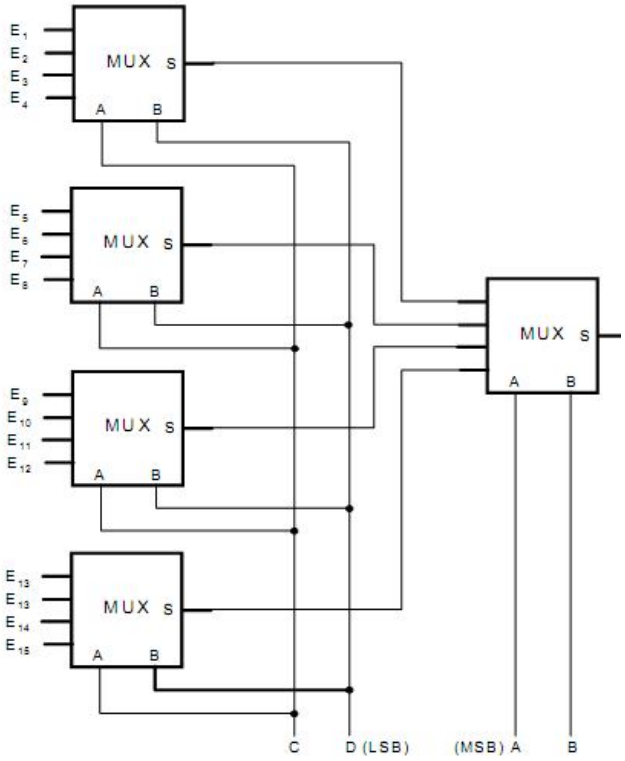

Figura 5-5 - Multiplexador 16 canais através de associação em Série
Fonte: LOURENÇO *et. al,* 2007

Os multiplexadores têm inúmeras funções, como sinalizar informações de vários bits, desenvolver expressões booleanas, ou selecionar informações digitais para serem transmitidas a outro sistema digital. Também pode ser chamado de circuito lógico universal, já que ele pode ser usado como uma solução de projeto para qualquer tabela verdade, desde que o número de variáveis seja igual ao número de entradas de seleção.

5.1.2. Demultiplexadores

O demultiplexador ou DEMUTEX (como é mais conhecido), é praticamente o inverso do multiplexador.

De acordo com LOURENÇO *et. al.* (2007), o demultiplexador é definido como: "um circuito combinacional dedicado que tem a finalidade de selecionar, através das variáveis de seleção, qual de suas saídas deve receber a informação presente em sua única entrada".

Observamos na imagem 5.6 a representação de um demultiplexador que possui *n* saídas.

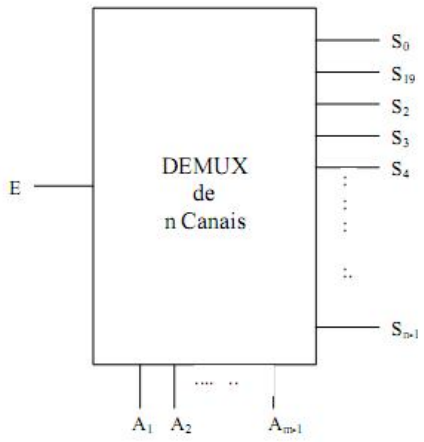

Figura 5-6 - Representação de um demultiplexador
Fonte: LOURENÇO *et. al,* 2007

Assim como os multiplexadores, encontramos os demultiplexadores também com 2, 4, 8 ou 16 canais, sendo que podemos observar na Figura 5.7 a representação do modelo mais simples, ou seja, com dois canais. Neste caso temos a entrada E, a variável de seleção A e as saídas S0 e S1. Podemos notar ainda que os demultiplexadores utilizam as portas lógicas AND e NOT.

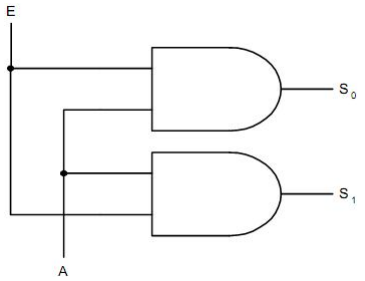

Figura 5-7 - Circuito lógico do demultiplexador de dois canais.
Fonte: LOURENÇO *et. al,* 2007

Segundo LOURENÇO *et. al.* (2007), assim como os multiplexadores, também podemos ter a associação paralela ou em série de demultiplexadores, possibilitando também a ampliação do número dos canais de saída para uma única entrada ou ainda ampliar o número de entradas para se obter mais de um canal de saída ativa simultaneamente.

A associação paralela permite que a capacidade de canais simultâneos de saída seja ampliada, ou seja, através dela torna-se possível demultiplexar informações digitais de vários bits simultaneamente.

Neste tipo de associação devemos utilizar um demultiplexador com um número de canais de saída igual ao número de informações a serem demultiplexadas, sendo o número de demultiplexadores igual ao número de bits destas informações.

O exemplo da Figura 5.8 seria utilizado caso fosse necessário demultiplexar quatro informações diferentes, sendo cada uma delas composta de três bits. Teríamos, portanto quatro demultiplexadores associados paralelamente, sendo que cada demultiplexador possui quatro canais de saída, um para cada informação e os três bits que compõem a informação seriam distribuídos simultaneamente um para cada demultiplexador.

A associação em série, entretanto permite que a capacidade de canais de saída seja ampliada, sendo uma variação da associação paralela, pois para ampliar a capacidade de entradas basta conectar os demultiplexadores de saída em um demultiplexador de entrada.

No exemplo da Figura 5.9, poderíamos construir um demultiplexador de 16 canais através da associação em série de quatro demultiplexadores com quatro canais de saída cada um e também um quinto demultiplexador, também com quatro canais de

saída, conectando as entradas dos demais demultiplexadores aos seus canais de saída.

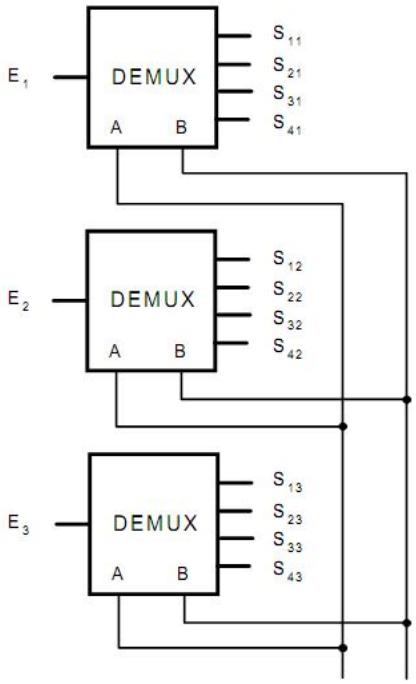

Figura 5-8 - Demultiplexação de quatro informações de 3 Bits
Fonte: LOURENÇO *et. al,* 2007

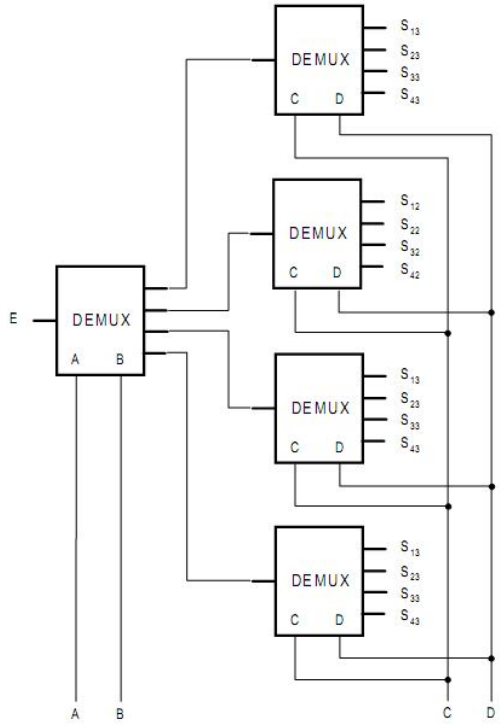

Figura 5-9 - Demultiplexador de 16 canais através de associação em Série
Fonte: LOURENÇO *et. al,* 2007

Através dos demultiplexadores podemos, por exemplo, selecionar os circuitos que devem receber uma determinada informação digital, converter informações seriais em paralelas, entre outras.

6. Metodologia

O projeto visa proporcionar ao portador de deficiência física motora maior acessibilidade e mobilidade dentro de sua própria residência além de minimizar grande parte das dificuldades encontradas como, por exemplo, verificar, ligar ou desligar um determinado dispositivo como TV ou lâmpadas sem a necessidade de se locomover até o local, uma vez que muitas dessas residências não se encontram adaptadas às condições e necessidade desse morador.

Resumidamente, o projeto acompanha o diagrama da Figura 6.1, ou seja, o morador, portador de deficiência física, transmitirá dados para controlar os dispositivos eletrônicos desejados através de seu celular a um Web Service disponível em seu computador, sendo essa comunicação por uma rede sem fio instalada e conFigurada em sua residência. O Web Service por sua vez irá disponibilizar as informações obtidas a um circuito digital que ficará conectado a interface paralela desse mesmo computador, que será o responsável por controlar tais dispositivos.

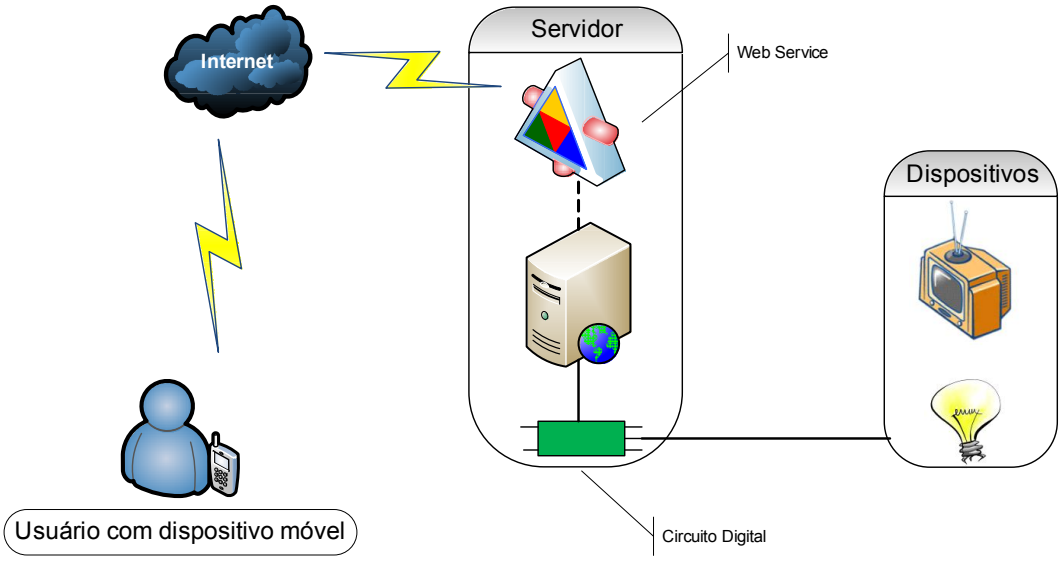

Figura 6-1 - Diagrama funcionamento do projeto

A realização do projeto foi dividida nas seguintes etapas:

A partir das informações coletadas no levantamento bibliográfico, foi possível elaborar o esquema do circuito digital proposto para o desenvolvimento do projeto. Para essa etapa também foi necessário estudar o funcionamento dos componentes

eletrônicos empregados na construção do circuito principal, além do padrão de pinagem do conector DB25, que possui 25 pinos conforme Figura 6.2, para a porta paralela do computador. Nesse conector, um pino está em nível lógico 0 (zero) quando sua tensão elétrica está entre 0 e 0.4 volts e encontra-se em nível lógico 1 (um) quando a tensão está entre 3.1 e 5 volts.

Os pinos do conector são ativados conforme o *byte* enviado a porta paralela, ou seja, para ativar a saída D0 deve-se enviar o *byte* 1 (um), para ativar a saída D1 deve-se enviar o *byte* 2 (dois), as saídas D0 e D1 deve-se enviar o *byte* 3 (três), para ativar todas as saídas envia-se o *byte* 255 e assim por diante.

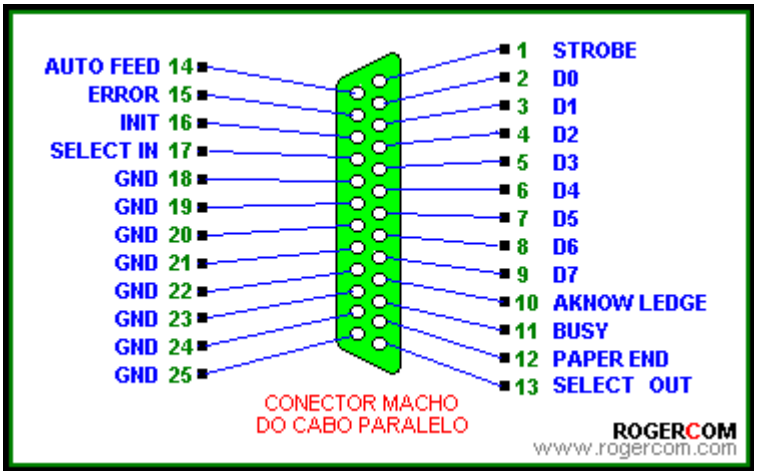

Figura 6-2- Pinagem conector paralelpo (DB25)
Fonte: http://www.rogercom.com (acesso em 01/11/2010)

O circuito proposto seguiu o esquema da Figura 6.3 na qual se pode observar a utilização de um componente chamado *Buffer Octal* (74LS541) que tem seu diagrama apresentado na Figura 6.4 e a finalidade de proteger a porta paralela além de fornecer os níveis de corrente elétrica suficientes para ativar os demais componentes.

Esse componente trabalha com uma corrente de 0.1 mA e tensão máxima de 7V, possui 20 conexões, possuindo oito entradas (representadas pelas siglas A1 à A8) e outras oito saídas (representadas pelas siglas Y1 à Y8). Essas entradas foram conectadas cada uma ao seu respectivo pino no conector DB25, ou seja, o pino D0 do conector paralelo foi ligado a entrada A1 do *Buffer Octal*, já o pino D1 do conector foi ligado a sua entrada A2 e assim por diante até ter as oito entradas conectadas além de conectar também o pino terra, representando pelas letras GND. Na Figura 6.4 à um dos

pinos entre 18 e 25 do conector DB25, que são seus pinos negativos, também conhecidos como terra. Já as saídas Y1 à Y8 foram conectadas cada uma a um resistor de 470 ohm, componente que pode ser observado na Figura 6.5 e tem a função de regular a tensão quando a mesma passar por ele evitando assim que os componentes conectados a ele não sofram qualquer tipo de dano.

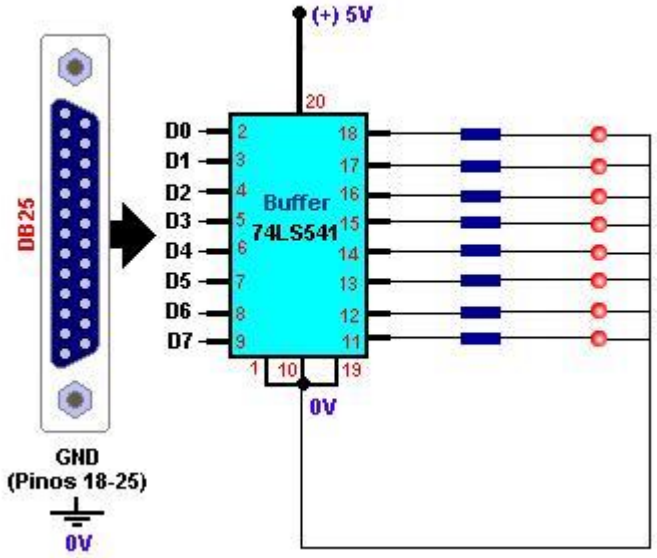

Figura 6-3- Esquema Circuito

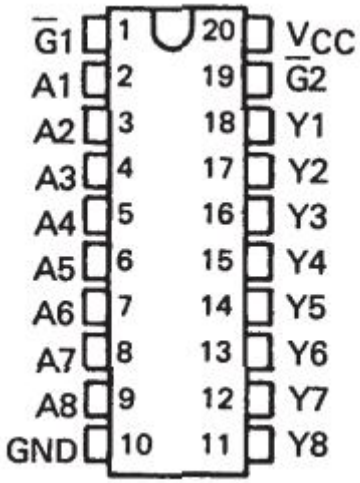

Figura 6-4 - Diagrama componente 74LS541
Disponível em: http://www.alldatasheet.co.kr/datasheet-pdf/pdf_kor/28025 (acesso em 01/11/2010)

Figura 6-5- Resistor 470 Ohm
Disponível em: http://luttech.wordpress.com (acesso em 01/11/2010)

Esses oito resistores estão conectados a um componente emissor de luz chamado LED, que operam normalmente com uma tensão entre 3 e 6V e corrente entre 40 e 50 mA e pode ser observado na Figura 6.6, através de seus terminais positivos, chamados de ânodo, tendo sua outra extremidade, o pólo negativo que é chamado de cátodo conectado também a um dos pinos entre 18 e 25 do conector DB25 que são os pinos com pólo negativo. Esses LEDS simbolizam os dispositivos a serem controlados.

Figura 6-6- LED
Disponível em: http://www.societyofrobots.com/electronics_led_tutorial.shtml (acesso em 01/11/2010)

Nesse protótipo pode-se controlar até 8 dispositivos diferentes já que o mesmo possui apenas 8 saídas, porém caso seja necessário pode-se, através do uso de multiplexadores e demultiplexadores aumentar esse número para até 32 dispositivos a serem controlados.

Na Figura 6.7 pode-se observar o circuito construído em uma placa já perfurada com 10 x 10 cm de dimensão, utilizando os componentes citados anteriormente.

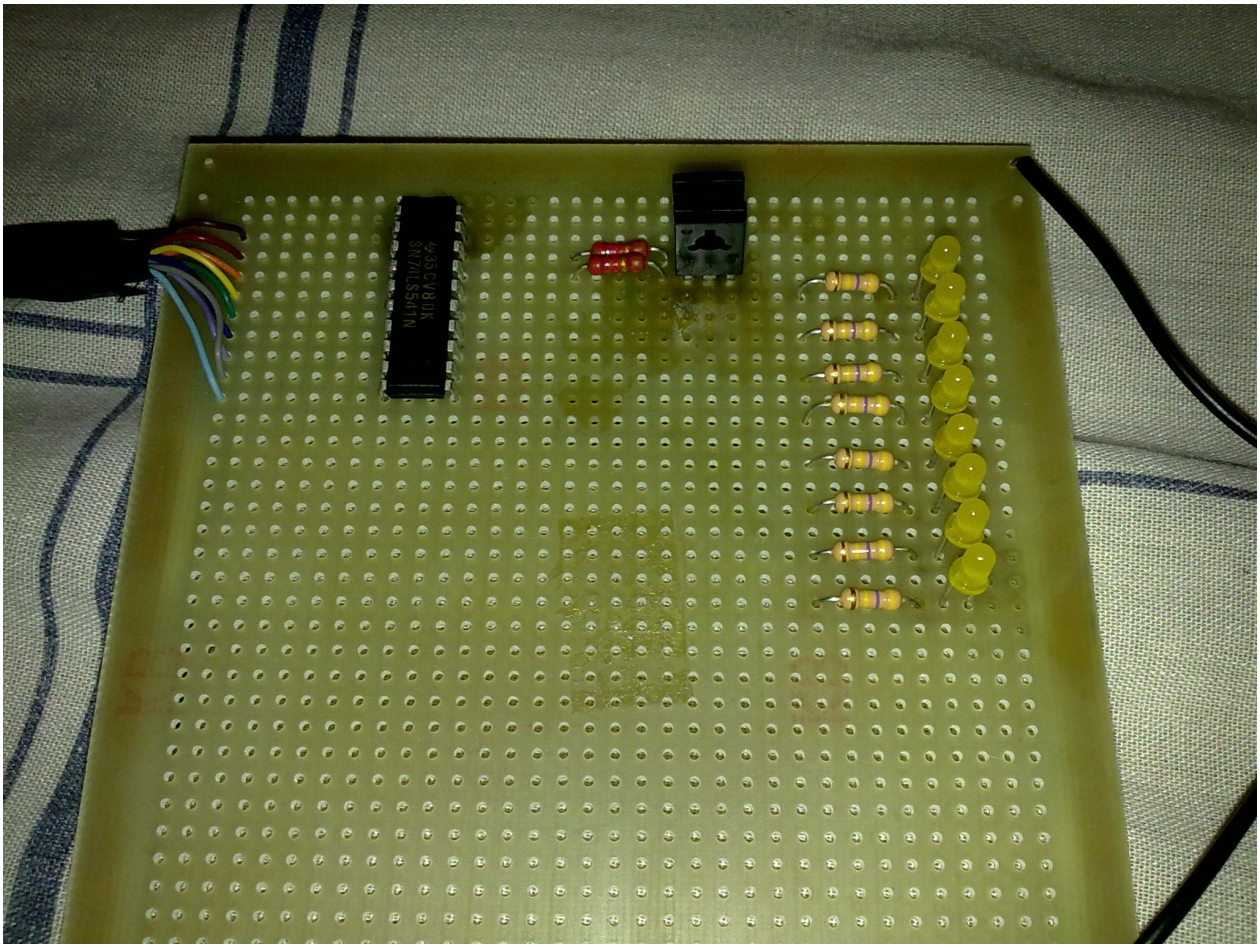

Figura 6-7- Circuito desenvolvido

Após a conclusão da construção do circuito foi desenvolvido um Web Service, em Java, que armazena as informações enviadas pelo aplicativo do dispositivo móvel, disponibilizando essas informações ao aplicativo localizado no microcomputador do usuário. Nessa etapa o Web Service dispõe de um número de variáveis iguais ao número de dispositivos controlados, sendo assim, nesse caso ele possui 8 variáveis chamadas *Disp1, Disp2, Disp3, Disp4, Disp5, Disp6, Disp7, Disp8* e mais duas funções relacionadas a cada uma dessas variáveis chamadas *setDisp*, que recebe como parâmetro um valor *booleado* (*true ou false)* indicando o estado do dispositivo e uma segunda função chamada *getDisp* que retorna ao solicitante o estado atual do dispositivo desejado.

O Web Service utiliza uma linguagem padronizada chamada WSDL (*Service Web Definition Language)* para descrever seus serviços e interfaces disponibilizados independente de sua plataforma ou linguagem de programação. Cada documento

WSDL criado contém especificações em XML que fazem referência aos tipos de dados, operações e outras informações disponíveis no serviço.

Os principais elementos disponíveis no arquivo WSDL são:

- *Namespaces*: é um espaço para nomes, definidos no interior do arquivo XML e tem a função de maximizar a taxa de reutilização dos componentes de um documento WSDL, utilizando-se de atributos para fazer referência a outros elementos, seja dentro ou fora do documento;
- Elemento <*definitions*>: É o elemento raiz do documento WSDL. Ele contém atributos que servem para definir os *namespaces* utilizados;
- Elemento <*types*>: Contem os tipos de dados que estão presentes na mensagem;
- Elemento <*message*>: Define os dados a serem transmitidos. Cada elemento *message* recebe um ou mais elementos <*part*>, que formam as partes reais da mensagem. O elemento <*part*> define o conteúdo da mensagem representando os parâmetros que são passados e a resposta que o serviço retorna;
- Elemento <*import*>: Funciona como uma separação entre as várias partes de uma definição WSDL. Ele possui dois atributos, os quais definem a localização do documento importado e o seu *namespaces;*
- Elementos <*operation*> e <*portType*>: Os elementos <*portType*> possuem um conjunto de operações, as quais possuem um atributo nome e um atributo opcional para especificar a ordem dos parâmetros usados nas operações;
- Elemento <*binding*>: Mapeia os elementos *operation* em um elemento *portType*, para um protocolo específico. Ele associa o elemento *portType* ao protocolo SOAP, utilizando-se de um elemento de extensão SOAP chamado <*wsdlsoap:binding*>*,* através de dois parâmetros: protocolo de transporte e o estilo de requisição;
- Elementos <*service*> e <*port*>: definem a localização real do serviço, tendo em vista que o mesmo pode conter várias portas e cada uma delas é especifica para um tipo de ligação descrita no elemento *binding*;

Pode-se observar na Figura 6.8 uma pequena parte do documento WSDL criado.

Figura 6-8- Documento WSDL Criado

Para o funcionamento do Web Service, é necessário que se tenha um servidor para hospedar o serviço. O servidor utilizado para o funcionamento do Web Service foi o APACHE TOMCAT que é distribuído como *Software livre,* e é de fácil instalação e conFiguração.

Após a conclusão da montagem do circuito e o desenvolvimento do Web Service a próxima etapa foi o desenvolvimento do aplicativo responsável por ligar e desligar os dispositivos (LEDS) conectados ao circuito. Este aplicativo fica localizado no computador no qual também está conectado o circuito desenvolvido.

Seu funcionamento seria basicamente se conectar ao Web Service disponível e fazer a leitura dos estados dos dispositivos controlados e enviar o byte correspondente ao circuito, ligando ou desligando um determinado dispositivo. Primeiramente, essa aplicação foi desenvolvida totalmente por meio da tecnologia Java, utilizando uma biblioteca gratuita chamada *Jnpout32.dll*, porém durante os testes realizados a

biblioteca não funcionou adequadamente exigindo assim outra alternativa para o acesso a porta paralela.

Com a exigência de outra alternativa que atendesse a necessidade, o escopo do aplicativo proposto foi alterado, ou seja, ao invés de construí-lo utilizando totalmente a linguagem Java, ele foi construído em duas etapas. Uma etapa, que faz a conexão e a leitura dos dados com o Web Service (desenvolvida em Java), já a segunda etapa foi desenvolvida com a linguagem de programação C++.

Portanto existe um aplicativo, em Java, que faz a conexão ao Web Service através do protocolo TCP/IP, requisitando através das funções *getDisp()* os estados de cada um dos dispositivos controlados e contém também uma *thread* que executa a cada 5 segundos o aplicativo desenvolvido em C++. Nesse caso o aplicativo em C++ também utilizou uma biblioteca gratuita e específica para o Sistema Operacional do ambiente utilizado chamada *inpout32.dll* e recebe como parâmetro a somatória dos bits de cada um dos dispositivos ligados, ou seja, o *byte* correspondente aos dispositivos controlados é escrito através da função *out32(),* no endereço, em hexadecimal, *0x378*, que é o endereço normalmente utilizado para a escrita na porta paralela, ligando ou desligando os dispositivos.

Durante os testes realizados com o aplicativo desenvolvido, em um ambiente que dispunha do Sistema Operacional Windows, foi observado que em uma quantidade razoável de vezes a execução do aplicativo apresentava falhas por existir um conflito entre a biblioteca utilizada e o Sistema Operacional.

Após a falha descoberta se fez necessário alterar o Sistema Operacional do microcomputador que estava sendo utilizado, no qual, foi instalado Linux através de sua distribuição chamada UBUNTU, que foi escolhida por ser uma das mais amigáveis, simples e fáceis de conFigurar.

Com a instalação também se fez necessário conFigurar o ambiente com os devidos compiladores da linguagem escolhida para o aplicativo para realização dos testes. Nesse ponto, como o Sistema Operacional utilizado, no caso o Linux, não necessita de nenhuma biblioteca para realizar o acesso a porta paralela, foi necessário realizar pequenas alterações no código fonte do aplicativo.

Assim como anteriormente, o aplicativo recebe como parâmetro o *byte* correspondente aos estados dos dispositivos, e através da função *Outb*() escreve esse valor no endereço utilizado pela porta paralela (0x378).

Após os ajustes necessários o aplicativo foi compilado e executado em todas as suas vezes com sucesso. Pode-se observar na Figura 6.9 a execução do aplicativo, primeiramente passando como parâmetro o *byte* 255 que ativa todos os dispositivos, depois passando o *byte* 3, como na Figura 6.10, que ativaria os dois primeiros dispositivos.

Figura 6-9 - Execução do aplicativo passando o *byte 255* como parâmetro e o circuito com todos os LEDS ativos

Figura 6-10 - Execução do aplicativo passando o *byte* 3 como parâmetro e o circuito com os dispositivos 1 e 2 ativos

Foi desenvolvido também um segundo aplicativo, que se encontra instalado no dispositivo móvel do usuário e tem a finalidade de enviar os dados (*bytes* que ligam ou desligam os dispositvos) para o Web Service. O aplicativo foi desenvolvido com a tecnologia J2ME e tem como meio de comunicação a rede sem fio disponível.

Para o desenvolvimento desse aplicativo foi necessário instalar e conFigurar o SDK pertinente ao sistema operacional do aparelho utilizado, nesse caso, foi utilizado um *smartphone* da marca *NOKIA*, modelo *5800 ExpressMusic* com Sistema Operacional *Symbian* em sua versão S60 5ª edição. Quando o SDK é instalado, grande parte da conFiguração necessária para sua utilização junto com a plataforma de desenvolvimento, nesse caso o *Netbeans,* foi feita automaticamente.

O aplicativo para o dispositivo móvel faz a conexão com o Web Service, utilizando também o protocolo TCP/IP do microcomputador em que o serviço encontra-se disponibilizado. A partir daí ele faz chamadas às funções disponíveis no Web Service, ou seja, em um primeiro momento o aplicativo utiliza a função *getDisp* para

obter os estados atuais de cada um dos dispositivos e posteriormente utiliza a função *setDisp* para enviar os novos estados para o Web Service.

Na Figura 6.11 podem-se observar as telas do aplicativo para o dispositivo móvel. Primeiramente (na imagem à esquerda), após a requisição dos dados ao Web Service, nenhum dos dispositivos controlados estavam ativos, e na imagem à direita, foram selecionados os dispositivos 1 e 2 para serem ligados.

Figura 6-11 - Aplicativo do dispositivo móvel sem nenhum dispositivo ativo e com os dispositivos 1 e 2 ativos.

7. Resultados

Após todos os aplicativos, o circuito e as configurações necessárias para o funcionamento do conjunto, foram realizados testes com o intuito de identificar possíveis erros durante a construção do circuito e também do desenvolvimento dos aplicativos.

Os testes consistiram na execução do projeto por várias vezes, ou seja, os dispositivos foram ativados e desativados inúmeras vezes, o microcomputador que hospeda o Web Service também foi reiniciado simulando uma possível queda de energia. Durante a realização desses testes em um ambiente corretamente conFigurado, ou seja, com a rede *Wireless* funcionando e o Web Service ativo, não ocorreram quaisquer erros ou falhas, o que proporciona resultados extremamente satisfatórios.

Pode-se fazer uma observação pela necessidade de utilização de um Sistema Operacional baseado em Linux uma vez que, em ambientes que utilizam Windows, o aplicativo de controle da porta paralela não correspondeu às expectativas por necessitar de bibliotecas de terceiros que não se encontravam totalmente estáveis para utilização, o que gerou conflitos entre a biblioteca e o Sistema Operacional.

A segunda observação fica por conta da utilização de kits de desenvolvimento (SDK) específicos para cada tipo e/ou marca de dispositivo móvel, o que gera tempo adicional na migração entre as plataformas, ou seja, para utilizar o aplicativo de um dispositivo de uma determinada marca em um dispositivo de uma segunda marca é necessário recompilar o aplicativo utilizando a nova plataforma.

8. Considerações Finais

O uso das tecnologias móveis está em constante discussão, e o principal exemplo talvez sejam os celulares, que não são mais apenas aparelhos para comunicação via voz. O projeto realizado proporciona ao deficiente físico motor melhoras significativas nas rotinas e funções desempenhadas durante o seu dia-a-dia, permitindo assim que se tenha maior segurança, mobilidade e comodidade dentro de sua própria residência, que em muitas vezes não se encontra adaptada às necessidades desse morador.

8.1. Trabalhos futuros

Podemos, após a finalização do projeto em questão, realizar diversas melhorias no projeto como, por exemplo:

- Modificar o circuito lógico para que seja possível receber dados e não apenas emitir, permitindo ao dispositivo controlado retornar uma informação de um possível erro ocorrido;
- Alterar a interface de conexão paralela do computador com os circuitos responsáveis pelo controle dos dispositivos eletrônicos para conexões USB ou ainda sem fio;
- Criar formulários, no aplicativo do dispositivo móvel, para cadastros das descrições e outras informações dos dispositivos controlados permitindo assim maior flexibilidade ao usuário, bem como inserir uma imagem da planta da residência com os respectivos dispositivos controlados, melhorando a interatividade com o sistema;
- Melhorar os mecanismos para tratamento de erros e falhas de conexão com o Web Service;
- Desenvolver mecanismo para controle através de comandos por voz, para também proporcionar melhores condições de vida ao usuário portador de deficiência visual.

Referências

ABNT. (2004). Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos. Rio de Janeiro, RJ, Brasil.

CARVALHO, L. R. (2001). Ergonomia e o trabalho do portador de necessidade motora. Florianópolis, SC.

GLASS, G. (2000). Applying Web services to applications.

KREGER, H. Web Services Conceptual Architecture. IBM Software Group.

LOURENÇO, A. C., CRUZ, E. C., FERREIRA, S. R., & JÚNIOR, S. C. (2007). Circuitos Digitais (2 ed.). São Paulo: Editora Érica Ltda.

NOVO, E. M. (1992). Sensoriamento Remoto (2 ed.). São Paulo: EDGARD BLUSHER LTDA.

SANTOS, L. K. (2004). Diretrizes de arquitetura e design para adaptação da habitação de interesse social ao cadeirante. Curitiba, PR.

TÉCNICAS, A. B. (n.d.). Acessibilidade de pessoas portadoras de deficiências a edificações. Rio de Janeiro.

Developer Resources for Java Technology. (n.d.). Disponível em: http://java.sun.com Acesso em 29 mar. 2010.

WEB Services Architecture, Web Services Architecture (2004).Disponível em: http://www.w3.org/TR/ws-arch acesso em: 03 mar. 2010.