

**CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**SMART WALKING STICK (SWS)
SISTEMA COLABORATIVO PARA DEFICIENTES VISUAIS**

RAPHAEL FRANCHINI BARROS SOARES

**Marília
2013**

**CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**SMART WALKING STICK (SWS)
SISTEMA COLABORATIVO PARA DEFICIENTES VISUAIS**

Monografia apresentada ao Centro universitário Eurípides de Marília, como parte dos requisitos parciais para obtenção do grau de Bacharel em Ciência da Computação.

Orientador:
Prof. Dr. Fábio Dacêncio Pereira

**Marília
2013**



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Raphael Franchini Barros Soares

SISTEMA COLABORATIVO PARA DEFICIENTES VISUAIS SWS

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 10,0 (Dois)

Orientador: Fábio Dacêncio Pereira

1º. Examinador: Ildeberto de Gênova Bugatti

2º. Examinador: Giulianna Marega Marques

Marília, 05 de dezembro de 2013.

DECIDATÓRIA

Dedico primeiramente a Deus por ter concedido a mim total capacidade, força e inteligência para realização deste trabalho tão importante nesta etapa de minha vida.

Dedico este trabalho aos meus pais, Luis Carlos Barros Soares e Margareth Franchini Soares que mesmo não entendendo muito sobre o meu curso sempre me apoiaram em momentos difíceis.

Dedico este trabalho aos meus Sogros, Ailton Barbosa Correia e Maristela Colombo Correia, pelo apoio e incentivo para realização deste trabalho.

E por fim, porém não menos importante, dedico este trabalho a minha noiva Tatiara Colombo Correia, por todo amor e afeto que me aplica, desde o primeiro ano do curso.

AGRADECIMENTOS

Agradeço a todos que participaram direto e indiretamente da minha formação.

Agradeço ao meu professor orientador Dr. Fábio Dacêncio Pereira pelas ideias e orientações para realização deste trabalho.

Agradeço o Governo do Estado de São Paulo, pela oportunidade de participar do Programa Escola da Família pela razão a qual hoje estar concluído este curso.

Agradeço aos meus Amigos que me acompanharam durante o curso e em especial aqueles que conseguiram concluí-lo.

LISTA DE ILUSTRAÇÕES

Figura 1 - Arduino Uno	14
Figura 2 - A arquitetura do SWS lado A.....	22
Figura 3 - A arquitetura do SWS lado B	23
Figura 4 - Diagrama de blocos do sistema	23
Figura 5 - LV - MaxSonar - EZ4	25
Figura 6 - Principio de Funcionamento de um Sonar	25
Figura 7 - Esquema de ligação Arduino e MaxSonar	26
Figura 8 - Esquema de ligação Arduino e Vibracall	27
Figura 9 - Tela Principal do Sistema de Rastreamento visualizado num desktop	29
Figura 10 - Zoom e localização do dispositivo.....	30
Figura 11 - Esquema de ligação entre Arduino e Bluetooth.....	34
Figura 12 - Teste de requisições ao servidor	39

LISTA DE TABELAS

Tabela 1- Tempo de retorno da posição geográfica	35
Tabela 2 - Consumo médio dos componentes	36
Tabela 3 - Teste Módulo Colaborativo	40

LISTA DE ABREVIATURAS E SIGLAS

SWS – *Smart Walking Stick*

SPP – *Serial Port Profile*

PWM – *Pulse Width Modulation*

GPS – *Global Positioning System*

IDE – *Integrated Development Environment*

RFID – *Radio Frequency Identification*

SMS – *Short Messaging Service*

V – *Volt*

cm – *Centímetro*

SGBD – *Sistema Gerenciador de Bancos de Dados*

JSON – *Java Script Object Notation*

bps – *Bytes por segundo*

mAh – *Miliampere Hora*

RESUMO

Este trabalho propõe o desenvolvimento de uma bengala eletrônica que seja capaz de auxiliar a locomoção de portadores de deficiência visual. O dispositivo será composto por sensores e atuadores onde destacamos: Sonares, Motores Alertas, Módulo GPS, além do Módulo 3G para troca de informações pela rede sem fio e o desenvolvimento de um sistema colaborativo para rastreamento em tempo real do portador do dispositivo, informando sua coordenada geográfica e endereço.

Palavras-chave — Bengala Eletrônica, Sonar, GPS, Arduino, Motor DC.

ABSTRACT

This work proposes the development of an electronic cane that is capable of assistive mobility of the visually impaired. The device is composed of sensors and actuators which include: Radars, Engines Alerts, GPS Module, beyond 3G Module to exchange information over the wireless network and the development of a collaborative system for real-time tracking of the carrier device, informing their coordinated geographical and address.

Keywords - Bengal Electronics, Sonar, GPS, Arduino, DC Motor.

SUMÁRIO

INTRODUÇÃO	12
OBJETIVO	14
METODOLOGIA	14
ORGANIZAÇÃO DA MONOGRAFIA	15
CAPITULO 1 - TECNOLOGIA DE SENSORES E ATUADORES	17
1.1 Conceitos e Aplicações	17
1.2 Atuadores	17
1.3 Sensores	18
1.4 Considerações finais do capítulo	18
CAPITULO 2 - IMPLEMENTAÇÕES CORRELATAS	19
2.1 Bengala Eletrônica via Sensor de Ultrassom	19
2.1.1 Funcionalidades	19
2.1.2 Limitações	19
2.2 SACCOB: Sistema de Alerta Contra Colisão com Obstáculos	20
2.2.1 Funcionalidades	20
2.2.2 Limitações	20
2.3 Considerações finais do capítulo	21
CAPITULO 3 - ARQUITETURA DO SMART WALKING STICK (SWS)	22
3.1 Módulo Sensor	24
3.2 Módulo Atuador	27
3.3 Módulo Emergencial (Colaboração e Rastreamento)	28
3.4 Módulo de Configuração de Cenário	31
3.5 Módulo GPS e Módulo 3G	31
3.6 Módulo de Alimentação	32
3.7 Módulo Bluetooth	33
3.8 Considerações Finais do Capítulo	34
CAPITULO 4 - TESTES, VALIDAÇÕES E RESULTADOS	35
4.1 Tempo de retorno da posição geográfica por meio do Módulo GPS	35
4.2 Duração das Baterias	36
4.3 Distância de Detecção	37
4.4 Ativação do Módulo Alerta com limite de distância configurado	37

4.5	Ativação do Modulo Alerta próximo a ponto de Interesse.....	38
4.6	Disponibilidade do Servidor para envio das informações via Módulo 3G	38
4.7	Funcionamento da Página Web	39
4.8	Marcação de Pontos Interessantes e Envio de SMS	39
4.9	Conexão entre os Módulos Bluetooth.....	40
	CAPITULO 5 - CONCLUSÕES	42
	REFERÊNCIAS BIBLIOGRÁFICAS	43
	APÊNDICE A – CÓDIGO FONTE SERVIDOR WEB SWS.....	46
	APÊNDICE B – CÓDIGO FONTE PLACA ARDUINO	50
	APÊNDICE C – CÓDIGO FONTE MÓDULO 3G, GPS E MÓDULO BLUETOOTH.....	53
	APÊNDICE D – CÓDIGO FONTE TESTE: LEITURA E ENVIO DE DADOS PELA PORTA SERIAL ARDUINO	57
	APÊNDICE E – CÓDIGO FONTE TESTE: CONEXÃO, ENVIO E RECEBIMENTO DE BITS PELO CELULAR NOKIA N95	58
	APÊNDICE F – CÓDIGO FONTE TESTE: MEDIDO A DISTÂNCIA DE UM OBJETO COM O SONAR MAXSONAR EZ4.....	59
	APÊNDICE G – CÓDIGO FONTE TESTE: MEDIDO A DISTÂNCIA DE UM OBJETO COM O SONAR MAXSONAR EZ4 E ACIONAMENTO MOTOR VIBRA	60
	APÊNDICE H – TABELA DE CUSTO DO PROJETO	61

INTRODUÇÃO

De acordo com o Instituto Brasileiro de Geografia e Estatística, estima-se que no último censo mais de 35,7 milhões de pessoas apresentam algum grau de deficiência visual (IBGE, 2013). Seguindo o pensamento do Instituto Benjamin Constant, considera-se portador de deficiência visual o indivíduo que por algum motivo teve sua capacidade de visão reduzida, perca completa da visão em ambos os olhos de forma definitiva e mesmo que se faça uso de lentes corretivas ou até meios cirúrgicos não se consiga resultados em prol da melhoria da visão.

Várias tecnologias vêm sendo pesquisadas e desenvolvidas para auxiliar a locomoção de portadores de deficiência visual e inclusão social dos mesmos, como exemplo, a empresa amazonense Pentop venceu a etapa nacional do Prêmio FINEP de Inovação 2012 na categoria de Tecnologia Assistiva, concedido pela Financiadora de Estudos e Projetos (FINEP) com um sistema de vocalização de etiquetas voltado a pessoas portadoras de deficiência visual (FAPESP, 2013).

Define-se Tecnologia Assistiva, segundo Bersch, um termo ainda novo utilizado para identificar todo o arsenal de recursos e serviços que contribuem para proporcionar ou ampliar habilidades funcionais de pessoas com deficiência e consequentemente promover vida independente e inclusão social. Diante destes argumentos e a crescente inovação tecnológica existente em nosso país, novas Tecnologias Assistivas vem sendo materializadas para tornar a vida de pessoas com deficiências um pouco melhor, propondo facilitar o deslocamento de um portador de deficiência visual por ruas que apresentem vários obstáculos não identificados, a inclusão de portadores de deficiência motora em redes sociais por meio de softwares que fazem leituras de telas ou com softwares que controlam o computador por comandos de voz. Portanto, conclui-se que todas essas tecnologias tentam tornar a vida de portadores de deficiência mais fácil e minimizam a ideia de serem inferiores às outras pessoas que não possuem este tipo de deficiência.

Com uma quantidade significativa da população portadora de deficiência visual, a cada hora que se passa, no mundo novas ideias são desenvolvidas para amenizar a indiferença sofrida por pessoas portadoras de deficiências. Em um rápido levantamento referente à Tecnologia Assistiva, vários aplicativos e ferramentas são baseados em uma plataforma totalmente Open Source denominada Arduino.

Arduino é uma plataforma de computação open-source baseado em uma placa com entradas e saídas tanto digitais como analógicas. Possui um próprio ambiente de desenvolvimento que implementa a Linguagem C. O Arduino pode ser usado para desenvolver objetos interativos autônomos ou pode ser conectado a um software em seu computador (ex. Flash, Processing, MaxMSP). O Ambiente de desenvolvimento (IDE) open-source pode ser obtido gratuitamente (atualmente disponível para Mac OS X, Windows, e Linux) no site do fabricante.

Composta pelo microcontrolador Atmega328. Possui 14 entradas/saídas digitais (das quais 6 podem ser usadas como saídas PWM e duas para comunicação serial RX, TX), 6 entradas analógicas, um cristal oscilador de 16MHz, conexão USB, uma entrada para fonte, soquetes para ICSP, e um botão de reset. A placa contém todo o necessário para usar o microcontrolador. Para uso basta conecta-la a um computador com o cabo USB - AB (vendido separadamente) ou ligue a placa com uma fonte AC-DC (ou bateria). O Arduino Uno seleciona automaticamente a fonte de alimentação (USB ou fonte externa). Esta placa já vem pronta e testada com o microcontrolador ATmega328 pré-carregado com o bootloader.

Um das vantagens de se usar Arduino é que se trata de uma plataforma totalmente livre para um amplo mercado de desenvolvimento facilitado mesmo para pessoas que mesmo não sendo de áreas tecnológicas possam criar seus projetos sem dificuldades em um curto prazo de desenvolvimento.

Segundo Michael (MCROBERTS, 2010), desde que o Projeto Arduino teve seu início em 2005, mais de 150.000 placas Arduino foram vendidas em todo o mundo. O número de placas clone não oficiais sem dúvida supera o de placas oficiais, assim, é provável que mais de 500 mil placas Arduino e suas variantes tenham sido vendidas. Sua popularidade não para de crescer e cada vez mais pessoas percebem o potencial desse projeto de fonte aberta para criar projetos interessantes de forma rápida e fácil, com uma curva de aprendizagem relativamente pequena.



Figura 1 - Arduino Uno

OBJETIVO

Propõe-se como objetivo deste trabalho a intercomunicação de dispositivos externos ao Arduino para elaboração de um sistema que seja capaz de auxiliar a locomoção de deficientes visuais alertando-os de possíveis obstáculos a sua volta e integrar um sistema colaborativo que em tempo real, com auxílio de um GPS, um servidor Web e API Java script do Google Maps V3 possibilitará o rastreamento em tempo real do portador do dispositivo.

METODOLOGIA

O projeto subdivide-se em cinco partes que abordam: (A) pesquisa e funcionamento dos periféricos, (B) interconexão dos periféricos juntamente a placa Arduino, (C) programação para correto funcionamento do sistema, (D) testes de funcionamento e (E) coleta e avaliação dos resultados obtidos.

A. Pesquisa e funcionamento dos periféricos

Neste ponto é importante ressaltar que um bom estudo dos periféricos garantirá um bom funcionamento de todo o sistema futuramente, então, é de alta importância o estudo para identificar pontos positivos e negativos de cada componente utilizado.

B. Interconexão dos periféricos juntamente a placa Arduino

Inicialmente o projeto foi construído em uma placa conhecida como Protoboard, justamente por ser de fácil manuseio. Posteriormente com todos dispositivos já em perfeito funcionamento foi realizado a interconexão destes por fios e soldas para garantir que durante o uso do equipamento nada ocorra, por exemplo, problemas de contato.

C. Programação para correto funcionamento do sistema

A plataforma Arduino fornece um ambiente de desenvolvimento totalmente Open Source, que utiliza uma derivação da linguagem C. Por meio desta linguagem juntamente com a IDE fornecida pelo fabricante foi possível realizar a programação de uma parte do sistema, a outra parte foi programada utilizando a linguagem Python For S60, ambas visando o correto funcionamento do sistema.

D. Teste de funcionamento

Como todo software e hardware, o presente projeto foi testado sob diversas condições e cenários para garantir um bom desempenho para o usuário e também contribuir para correção de possíveis falhas em determinados cenários.

E. Coleta e avaliação dos resultados obtidos

Após a realização da fase anterior todos os dados foram registrados e analisados para garantir que tudo está de acordo com o estipulado e que o resultado esperado foi alcançado de forma consistente.

ORGANIZAÇÃO DA MONOGRAFIA

Esta monografia esta organizada em cinco capitulos para o melhor entendimento. Inicialmente o primeiro capitulo faz uma abordagem das tecnologias que podem ser utilizadas para alcance do objetivo principal, como sensores e atuadores. No segundo capitulo, são abordados dois trabalhos correlacionados ao objetivo principal. No terceiro capitulo é

apresentada a arquitetura do dispositivo bem como suas funcionalidades, no capítulo quarto é mostrado resultados dos testes e validações realizadas para alcançar o objetivo, e por fim, no quinto capítulo é apresentadas as conclusões.

CAPITULO 1 - TECNOLOGIA DE SENSORES E ATUADORES

1.1 Conceitos e Aplicações

Atualmente existem várias tecnologias e tipos de sensores e atuadores para diversos fins, por exemplo: sensor de pressão barométrica utilizado para realizar medições do volume dos tanques de combustíveis de um posto de gasolina, sensores de calor utilizados para identificar um possível incêndio. Já nos atuadores podem-se destacar os motores vibracall presente na grande maioria dos telefones móveis utilizados para alertar uma ligação. Outro atuador bem conhecido, o marca-passo, utilizado para manter o batimento cardíaco de forma regular, enfim, cada um com uma ou mais funções pré-estabelecidas.

Segundo Thomazini (2008, p.17)

Atuadores são dispositivos que modificam uma variável controlada. Recebem um sinal proveniente do controlador e agem sobre o sistema controlado. Geralmente trabalham com potência elevada. Já sensores, termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente, que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza que precisa ser medida, como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc.

1.2 Atuadores

Para um melhor entendimento abaixo é listado alguns tipos de atuadores, utilizados comercialmente:

- Atuadores Hidráulicos: Utilizam um fluído para movimentar seu braço central, comumente utilizado por robôs que operam com grandes cargas, por exemplo, robôs de uma linha de produção veículos;
- Atuadores Pneumáticos: Utilizam gás para atuar, como exemplo, o funcionamento de uma suspensão pneumática de um ônibus ou caminhão;
- Atuadores Eletromagnéticos: São os motores elétricos, vibracall, motores de passo, servo motores, dentre outros.

1.3 Sensores

Os sensores são classificados em dois grandes grupos: Sensores Digitais e Sensores Analógicos. Sensores digitais são aqueles que durante a execução de sua função podem assumir somente dois valores zero ou um em sua saída de resposta. Já os analógicos podem assumir vários valores para sua saída durante execução (THOMAZINI, 2008).

Dentre os diversos tipos de sensores destacam-se os mais comuns:

- Sensores de Pressão;
- Sensores de Temperatura;
- Sensores de som;
- Sensores de corrente elétrica;
- Sensores de movimento;
- Sensores de orientação;
- Sensores de Objetos;
- Sensores de Movimento;

1.4 Considerações finais do capítulo

Vários tipos de atuadores e sensores existem no mercado, deve-se fazer a escolha de qual deles atenderá suas necessidades na implementação de um projeto. Em grande parte os sensores tanto quanto atuadores são de propósitos únicos, então, vale ressaltar que a combinação de um sensor com outro sensor e a combinação destes com outros atuadores, levam a um resultado satisfatório atendendo todas as necessidades do projeto. Neste projeto foi utilizado um sensor para distância e para atuadores os motores vibracall.

CAPITULO 2 - IMPLEMENTAÇÕES CORRELATAS

2.1 Bengala Eletrônica via Sensor de Ultrassom

Este projeto teve como objetivo principal o desenvolvimento de um sistema capaz de facilitar a locomoção de deficientes visuais em determinadas regiões com a utilização de um emissor de ondas sonoras, com isso, evitando que o portador de deficiência visual colida com anteparos localizados a sua frente. Segundo seus idealizadores, o intuito principal do trabalho é o foco na parte de eletrônica envolvida no projeto, por exemplo, a utilização de temporizadores, sensores, sonares, resistores entre outros componentes eletrônicos utilizados. Foi apontado como grande diferencial no desenvolvimento da proposta, a criação dos componentes utilizados ao contrário de outras bengalas eletrônicas existentes no mercado que utilizam placas de hardware livre para serem desenvolvidas (ALESSI et al, 2010).

2.1.1 Funcionalidades

- Detecção de objetos a uma distância fixa;
- Alertas vibratórios indicam obstáculos.

2.1.2 Limitações

- Como a grande parte dos componentes foi criada pelos desenvolvedores a ampliação do projeto torna-se algo complicado uma vez que a arquitetura pronta não suporta a expansão de suas funcionalidades;
- É utilizado pilhas e baterias para alimentar o sistema, tornando a arquitetura pesada e limitando a cada vez que se esgotam as pilhas é necessário realizar a troca, que resulta em gastos desnecessários uma vez que a utilização de baterias de recarregáveis resolveria

esta particularidade;

- A detecção de objetos tem um limite fixo, por exemplo, um metro. Isso gera problemas ao portador de deficiência visual quando o mesmo se encontrar em um ambiente onde os objetos estiverem posicionados a uma distância menor que este limite o dispositivo passa a ser um problema, pois não irá desligar os motores alerta causando confusão ao portador de deficiência visual.

2.2 SACCOB: Sistema de Alerta Contra Colisão com Obstáculos

Este projeto desenvolvido por Sefrin propõe o desenvolvimento de um sistema para facilitar a locomoção de pessoas portadoras de deficiência visual. O objetivo principal é a criação de um sistema que estende a área de Domótica com a utilização de sensores RFID que tem suporte a algoritmos de triangulação. Fixados aos objetos presentes no ambiente e no próprio portador de deficiência, estes sensores são responsáveis por fazer o mapeamento da distância entre o portador de deficiência visual e os obstáculos. Quando o portador de deficiência estiver prestes a colidir com algum obstáculo o mesmo é alertado por meio de alertas sonoros provenientes da Sintetização de voz.

2.2.1 Funcionalidades

- Mapeamento do local onde o sistema foi implementado: ao obter a distância entre os objetos e o portador dos sensores, verifica se a distância entre os dois é menor que a mínima, caso positivo, são acionados os sinais sonoros para alertá-lo, evitando possíveis colisões.

2.2.2 Limitações

- Limitado a ambientes: é necessário fazer a preparação do ambiente para o correto funcionamento do sistema, ou seja, para cada ambiente é necessário a aquisição de novos equipamentos e a instalação dos mesmos;

- Alto custo para desenvolvimento do sistema devido à tecnologia utilizada.

2.3 Considerações finais do capítulo

Várias tecnologias estão sendo desenvolvidas para auxiliar a locomoção de deficientes visuais, uma vez que o número de portadores deste tipo de deficiência esta aumentando. Por sua vez algumas destas tecnologias não atendem todas as necessidades, por este motivo é que foi desenvolvido este projeto, para atender outras necessidades que as soluções atuais não atendem, ou se já foram, seu custo é elevado dificultado sua aquisição. Um exemplo de tecnologias adicionais é a integração de um módulo GPS que permite fazer o rastreamento em tempo real do portador de deficiência visual, envio de sua coordenada por mensagens de texto e capacidade de adaptar o dispositivo para diferentes cenários visando aumentar a segurança e o bem estar familiar.

CAPITULO 3 - ARQUITETURA DO SMART WALKING STICK (SWS)

No desenvolvimento da proposta foram utilizados alguns componentes para alcançar o resultado esperado: desenvolver um módulo capaz de detectar e alertar o portador do dispositivo que a sua frente existem obstáculos, e também um sistema colaborativo para rastreamento e marcação de pontos que poderiam indicar obstáculos para que assim, quando outro portador do dispositivo estiver por perto, o mesmo será alertado que a sua frente há um obstáculo antes mesmo da detecção pelos sensores.

Dentre todos os componentes utilizados podem-se destacar os sensores, atuadores, placas Bluetooth, Módulo GPS e Módulo 3G, que por meio da integração destes junto à placa Arduino produziram um resultado satisfatório. Todos os módulos pertinentes à arquitetura podem ser observados na figura 2 e 3.

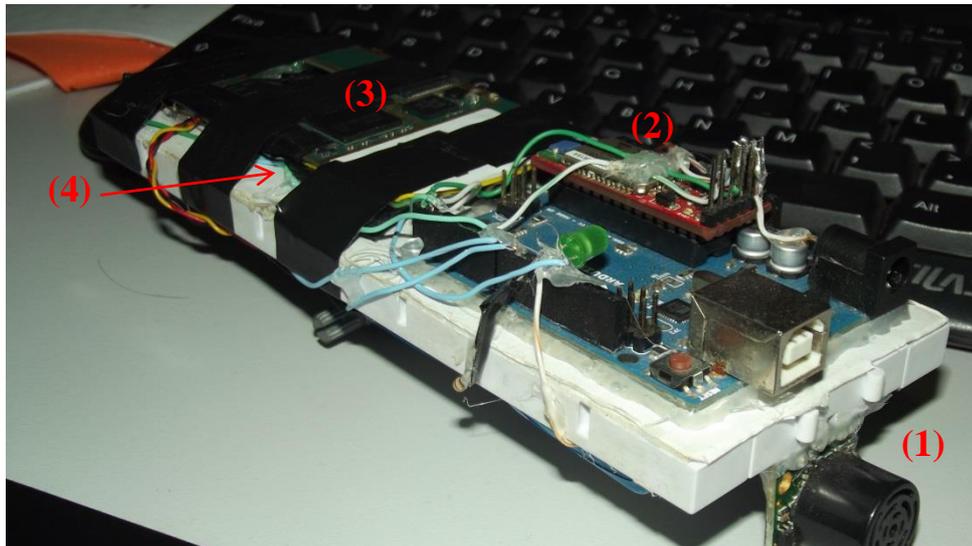


Figura 2 - A arquitetura do SWS lado A

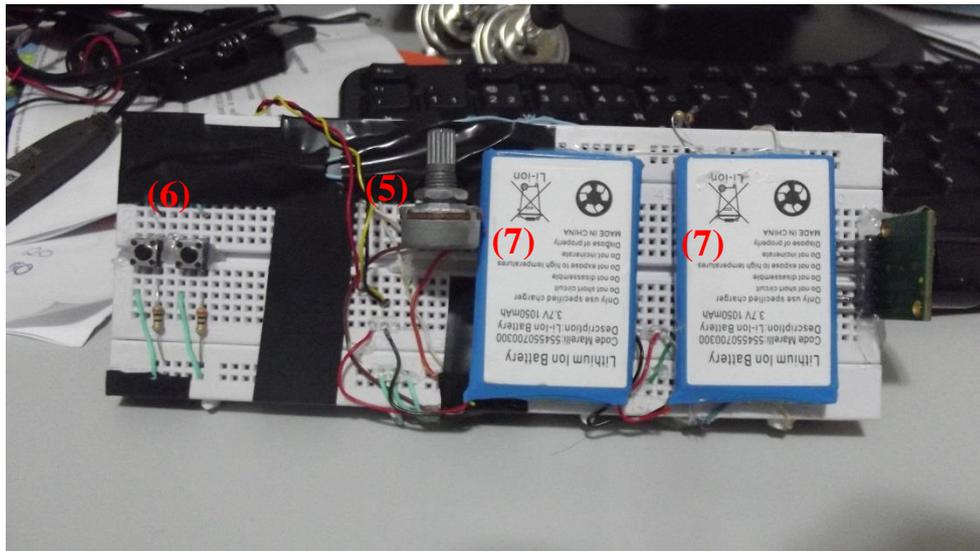


Figura 3 - A arquitetura do SWS lado B

Todos os Módulos relacionados à arquitetura do SWS são apresentados nas figuras 2 e 3 respectivamente, enumeradas de uma a sete. Cada Módulo tem função individual e a comunicação entre eles pode ser observada na figura 4 que apresenta o diagrama de blocos do sistema e o fluxo de comunicação, na sequencia é apresentado um resumo por Módulo, explicando superficialmente o que cada Módulo é responsável a fazer.

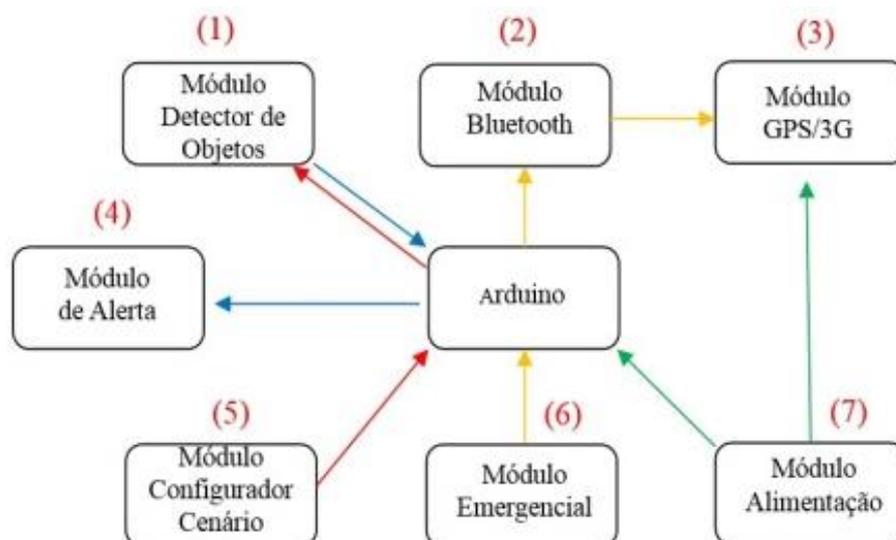


Figura 4 - Diagrama de blocos do sistema

1. Módulo Detector de Objetos: faz uso de um sensor ultrassônico para detecção de obstáculos à sua frente, quando encontrado, envia um sinal à placa Arduino que a partir desse sinal aciona o Módulo de Alerta;
2. Módulo Bluetooth: pelo fato da placa Arduino ser externa aos Módulos GPS/3G a placa Bluetooth tem a função de fazer a ponte de comunicação entre essas duas partes;
3. Módulo GPS/3G: responsáveis por três tarefas muito importantes da arquitetura, determinar a posição geográfica do dispositivo, enviar a posição para um Web Service que processa esta informações e por último este Módulo tem função de enviar SMS's em caso de emergência;
4. Módulo de Alerta: composto por Atuadores que alertam o portador quando objetos são detectados a sua frente;
5. Módulo Configurador de Cenário: composto por um potenciômetro que no momento em que seu eixo central é deslocado altera sua resistência e informa a placa Arduino, que por sua vez repassa para ao Módulo Detector de Objetos qual é a nova distância mínima para o objeto ser detectado;
6. Módulo Emergencial: composto por um botão que ao ser acionado se comunica com a placa Arduino que por sua vez por meio do Módulo Bluetooth envia uma requisição ao Módulo GPS/3G que interpreta o comando e dispara um SMS de socorro com a posição geográfica de onde se encontra o portador de deficiência para números de celulares previamente cadastrados;
7. Módulo de Alimentação: composto por baterias responsáveis por toda a alimentação da arquitetura.

3.1 Módulo Sensor

No desenvolvimento da proposta para realizar-se a detecção de obstáculos à frente do portador foi utilizado sensores ultrassônicos, mais precisamente o sensor LV-MAX SONAR-EZ4 fabricado pela Maxbotix, conforme mostra a figura 5.

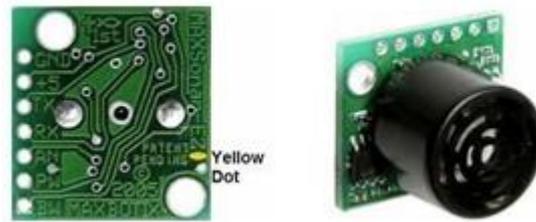


Figura 5 - LV - MaxSonar - EZ4

O funcionamento do sensor é baseado na emissão de uma onda ultrassônica no meio contido, então essas ondas se propagam até se chocarem com algum objeto, então esta onda é refletida para a direção contrária, ou seja, retornam ao emissor. Tomando por base que a velocidade de propagação do som no ar é de aproximadamente 340 M/s estima-se a distância que o objeto se encontra à frente do emissor, com a seguinte fórmula:

$$\text{Dist} = (T / 2) / 340$$

Onde Dist é o resultado da equação e nos mostra a distância calculada, T nos indica o tempo total da propagação da onda no meio contido (soma-se o tempo de ida mais o tempo de volta, por este fato que este valor é dividido pela constante dois). Por fim há uma divisão por uma constante (340) que se refere à velocidade de propagação das ondas sonoras no ar. A figura 6 mostra o funcionamento básico da emissão de ondas ultrassônicas num ambiente aberto.

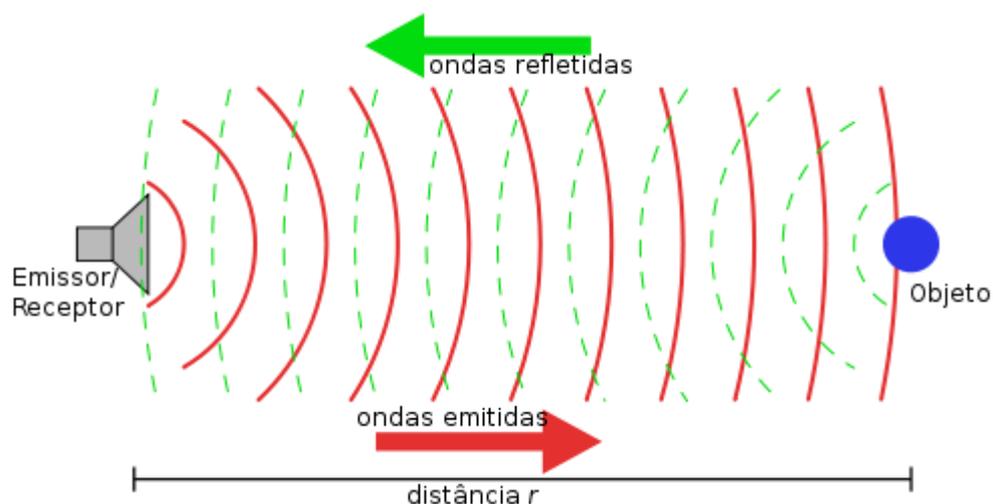


Figura 6 - Princípio de Funcionamento de um Sonar

Como mostra a figura 5, o dispositivo é composto por sete pinos de saída (GND, +5

V, TX, RX, AN, PW, BW) (MAXBOTIX, 2006), mais na implementação do projeto só foi utilizado às saídas GND que se trata do polo negativo do sistema, portanto este pino foi ligado diretamente ao pino GND presente na placa Arduino. Já o pino +5 v é o pino de alimentação do dispositivo (segundo o Datasheet do dispositivo o mesmo opera na escala de 2.5 V a 5.5 V) foi ligado ao pino de alimentação 5 V presente na placa Arduino e por fim o pino PWM (abreviação ou Modulação de Largura de Pulso, que pode ser entendida como uma técnica para obtenção de resultados analógicos em meios digitais), ligado diretamente à entrada PWM presente na placa Arduino, mais precisamente a entrada de número onze (Arduino, 2012). A figura 7 mostra o esquema de ligação entre a placa Arduino e o sensor utilizado.

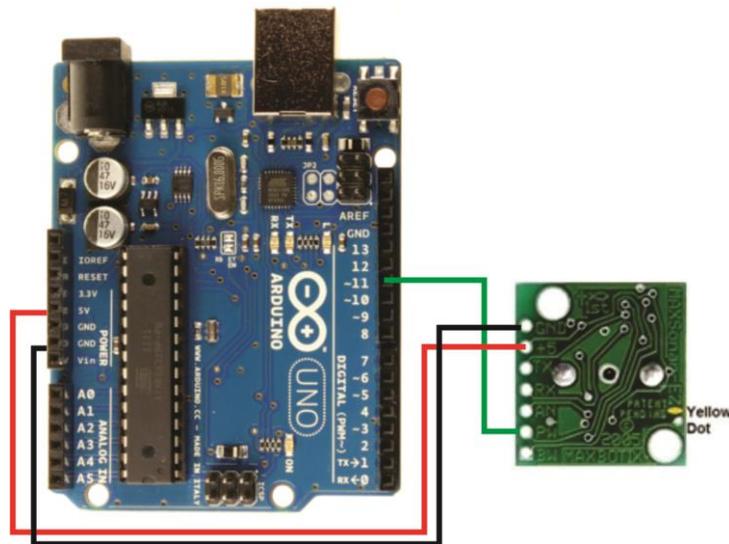


Figura 7 - Esquema de ligação Arduino e MaxSonar

No Apêndice F pode ser visualizado o código fonte que faz a leitura da saída PWM do Módulo. Segundo Maxbotix, este pino gera uma representação de largura de pulso de faixa, sendo assim é possível calcular a distância usando o fator de escala de 147us por polegada. Após a medição desta saída é feita a divisão acima e na sequência a conversão de polegadas para centímetros efetuando a multiplicação da constante que representa a polegada por 2,54 que é a representação de uma polegada em centímetros.

3.2 Módulo Atuador

Motores elétricos são máquinas destinadas a fazer a conversão de energia elétrica em mecânica, ou seja, movimentar seu eixo central presente na maior parte dos motores elétricos (WIKIPÉDIA, 2013). No desenvolvimento do trabalho foram utilizados motores elétricos vibracall, presente nos celulares com a função de alertar o usuário de alguma alteração de estado. A figura 8 apresenta o tipo de motor vibracall escolhido e utilizado já interligado juntamente a placa Arduino.

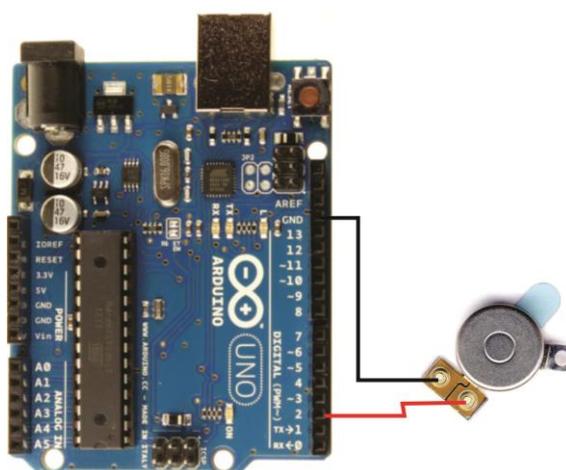


Figura 8 - Esquema de ligação Arduino e Vibracall

Operando lado a lado com o Módulo Sensor o Módulo Atuador recebe um sinal de energia elétrica proveniente da placa Arduino por uma porta digital e começa a vibrar alertando ao usuário que à sua frente existe um objeto que pode atrapalhar sua locomoção. A emissão do sinal pelo Arduino para que seja ativado o motor vibracall depende da distância em que se encontra o objeto, ou seja, se a distância detectada for menor que o valor pré-definido, sinais de energia elétrica são enviados caso contrário nada acontece. No Apêndice G, está o código fonte de testes do acionamento deste motor, tomando como padrão para acionamento o valor da medida detectada pelo Módulo Sonar, se esta distância for menor que 100 cm ou um metro o motor é acionado. Contudo, notou-se que o usuário pode estar em um cenário onde a distância dos objetos contidos neste cenário pode estar a uma distância menor que o estipulado 100 cm e o dispositivo ao invés de auxiliar sua locomoção o deixaria confuso, pois o vibracall ficará ligado o tempo todo. Para solucionar este problema foi criado

o Módulo de Configuração do Cenário permitindo alterar esse valor de ativação para o dispositivo poder operar em diversos cenários.

3.3 Módulo Emergencial (Colaboração e Rastreamento)

Para assegurar a tranquilidade dos familiares, foi desenvolvido um sistema capaz de rastrear o dispositivo em tempo real com a utilização do Módulo GPS e 3G. Além destes módulos foi também utilizado a API Java Script do Google Maps V3 (projetada especialmente para ser mais rápida e aplicada a dispositivos móveis), permitindo a incorporação de mapas a uma página Web que informa a posição do real dispositivo. Esta API trata-se de um serviço gratuito, disponível para qualquer Web site que seja gratuito para os consumidores (Google Developers, 2013). O funcionamento deste Módulo ocorre da seguinte forma: o Módulo GPS faz a leitura das coordenadas geográficas (Latitude e Longitude) de onde se encontra o dispositivo, em posse dessas informações é ativado o Módulo 3G que recebe estas informações do módulo anterior e as envia pela rede sem fio 3G a um Servidor Web implementado utilizando as tecnologias Open Source: WampServer, ambiente de desenvolvimento Web sob a plataforma Windows (WAMPSEVER), composto pelo Apache2 sendo o mais utilizado dentre os servidores Web livres (APACHE, 2012), PHP, linguagem de Script de uso geral amplamente utilizada para o desenvolvimento Web (PHP, 2013) e o Mysql, que se trata de um SGBD que utiliza a linguagem SQL para execução de comandos (MYSQL, 2013). Deve-se destacar também a utilização do JSON que se trata de uma forma leve para troca de informações de forma mais limpa e de fácil compreensão (JSON), JQuery, biblioteca Java Script com muitos recursos utilizáveis, no caso deste projeto utilizado para atualizar a posição do portador do dispositivo sem a necessidade de se recarregar a página Web (JQUERY, 2013) e uma ferramenta denominada NOIP, que permite fazer o acesso a um computador que possui um IP dinâmico (é alterado a cada vez que o usuário se conectar a internet) sem a necessidade de se contratar um serviço que forneça um IP Estático. Com a união destas tecnologias permitiu-se processo de desenvolvimento da proposta.

Na sequência, este servidor Web recebe as informações provenientes do Módulo 3G e as armazena no banco de dados MySQL. A cada nova informação recebida, instantaneamente a tela principal é atualizada com a nova posição do dispositivo. O uso para familiares é feito

acessando a seguinte página Web: smartwalking.zapto.org:4000, a partir de um computador ou até um Smartphone que a tela principal será carregada com a última posição geográfica válida do dispositivo, indicada por um marcador em vermelho conforme ilustrado na figura 9.

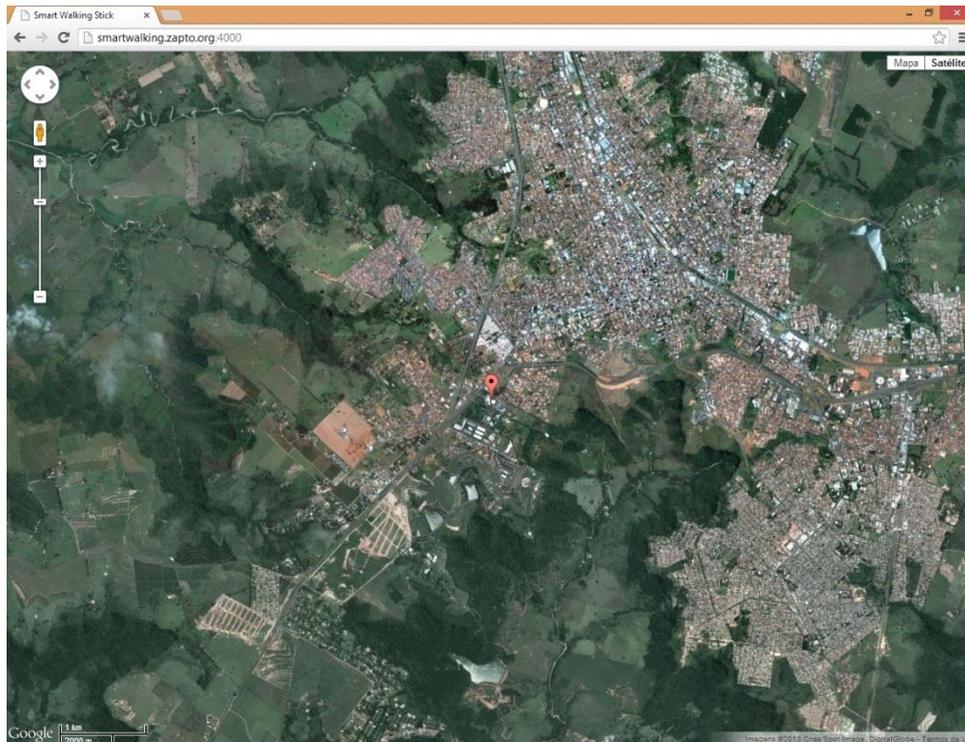


Figura 9 - Tela Principal do Sistema de Rastreamento visualizado num desktop

A Tela Principal por padrão se inicia com zoom distante. Para fazer a aproximação para melhor visualização, basta clicar com o botão direito do mouse no marcador vermelho (indicador de posição) que a aproximação será imediata, mais se preferir a mesma pode ser feita com rolagem do botão Scroll, afastando ou aproximando o mapa de acordo com o sentido do movimento. Por fim, para que seja mostrado o endereço de onde se encontra o portador de deficiência basta posicionar o mouse em cima do marcador de posição que o endereço é exibido na tela.



Figura 10 - Zoom e localização do dispositivo

Além das funcionalidades acima, o sistema conta com dois botões na arquitetura, ambos com funções distintas: O primeiro botão é utilizado para fazer a marcação (quando houver interesse) de um obstáculo que possa atrapalhar o deslocamento de outros portadores do dispositivo. O funcionamento se dá da seguinte forma: ao ser pressionado o botão, as coordenadas geográficas são captadas e enviadas ao servidor Web de forma diferenciada, sendo armazenada numa tabela especial. Quando outro dispositivo estiver perto destas coordenadas são enviados dados ao dispositivo que então alerta o portador que a sua frente existe um obstáculo antes mesmo de que o Módulo Sonar possa o detectar garantindo mais segurança ao portador do dispositivo, vale ressaltar que estas marcações não são visíveis na página Web e sua precisão depende muito do Módulo GPS.

Já o segundo botão tem a função de enviar um SMS para números de celulares previamente cadastrados com a seguinte mensagem: “Preciso de Ajuda. Estou próximo à: xxxxx, yyyyyy”, onde xxxxx representa a latitude e yyyyyy a longitude, com o intuito de garantir mais segurança aos familiares uma vez que nem sempre é possível fazer o acompanhamento do portador pela página web devido a atividades corriqueiras do dia a dia. Todo o código fonte da página web pode ser visualizado no Apêndice A.

Uma informação importante deve ser levada em consideração. O funcionamento dos Módulos aqui descritos depende de um pacote de acesso à internet e créditos para envio de SMS que são fornecidos pelas operadoras pertencentes à região, não chegando a R\$ 15,00 mensais.

3.4 Módulo de Configuração de Cenário

Segundo Maxbotix, o dispositivo tem a capacidade de realizar a detecção de objetos numa escala de 0 a 254 polegadas, aproximadamente 645 cm, com uma resolução de uma polegada. Objetos a uma distância menor do que seis polegadas (12 cm) são tratados como se estivesse a uma distância de 12 cm. Como o usuário pode a cada momento estar em cenários diferentes, foi projetado um método capaz de calibrar a distância que o objeto será detectado, ou seja, se o usuário estiver num local onde os objetos estão mais próximos, por exemplo, em sua casa, a detecção deverá ser mais precisa ao contrário do que acontece quando o usuário estiver num local aberto, onde os objetos estão mais dispersos.

Para realizar o cálculo da distância que o sensor identifica os objetos a sua frente, foi utilizado um componente denominado Potenciômetro, que se trata de um resistor com resistência variável que à medida que se gira o eixo central sua resistência é alterada fazendo com que a quantidade de corrente elétrica que passa pelos terminais seja maior, possibilitando determinar um valor X que é tomado como base para ativação do módulo Atuador, com a leitura da saída analógica deste potenciômetro.

3.5 Modulo GPS e Módulo 3G

Estes módulos são responsáveis por determinar a posição geográfica de onde se encontra o portador do dispositivo e envio destas informações ao servidor Web para que seja possível fazer o seu rastreamento em tempo real. Ao utilizar placa Arduino para intercomunicação dos periféricos pertencentes à arquitetura do SWS, notou-se que a mesma possui apenas uma porta serial para comunicação, visto que ao utilizar um módulo GPS específico e um Módulo 3G é necessário ao menos duas portas serial. Então foi utilizado um celular fabricado pela Nokia, modelo N95 8G que em sua arquitetura está presente tanto o

Módulo GPS quanto um Módulo 3G necessários, fazendo a comunicação entre estes módulos com a placa Arduino por meio do Módulo Bluetooth que utiliza apenas esta única porta serial atendendo a necessidade e tornando o projeto mais acessível. Todo o esquema de ligação destes Módulos com a placa Arduino foi realizado com a ajuda da linguagem de programação Python for S60.

Segundo Scheible (2010, p.6-7),

Symbian OS, é um sistema operacional projetado para dispositivos móveis. Ele inclui bibliotecas associadas, estruturas de interface com o usuário e serve de referência a implementações de ferramentas comuns. Como descendente da EPOC da Psion, ele é executado exclusivamente em processadores ARM. APIs Symbian OS são publicamente disponíveis e qualquer pessoa pode desenvolver softwares para Symbian OS. S60 é uma plataforma de software para telefones móveis baseados [...].

Foi grande a dificuldade para se programar Scripts utilizando esta linguagem, uma vez que são escassos os materiais referentes à mesma e a falta de uma IDE para desenvolvimento, mesmo que baseada na linguagem Python não é possível testar estes Scripts por que as importações pertencentes a esta derivação não funcionam na IDE Python. Contudo, conseguiu-se implementar os Scripts necessários para fazer toda a parte de envio de informações ao servidor Web, envio de SMS e captura da posição geográfica com auxílio do Módulo GPS, com a instalação de um emulador de Symbian OS e teste de Scripts diretamente do celular. Nokia Developer, mostra todos os passos para instalação e alguns exemplos de como iniciar a programação no ambiente.

3.6 Módulo de Alimentação

Nesta última década com a grande evolução de dispositivos eletrônicos, como notebooks, celulares, câmeras digitais, MP3, equipamentos médicos, aumentou-se muito a demanda por pilhas e baterias cada vez menores, com grande durabilidade e melhor desempenho. Basicamente as baterias podem ser classificadas em baterias primárias que não são capazes de serem recarregadas devido a sua composição química, e as baterias secundárias destinadas a equipamentos que demandam de um alto rendimento. Este segundo tipo é o mais utilizado atualmente, pois, é possível realizar o carregamento várias vezes pelo usuário (SOCIEDADE BRASILEIRA DE QUÍMICA). Para uso na arquitetura foi escolhido o

tipo de bateria secundária, por ser do tipo recarregável e sua capacidade de armazenamento ser superior aos tipos primários de baterias, e por consequência a alimentação de todo o sistema e mantida por mais tempo sem a necessidade de recarregá-las em um curto prazo de utilização.

3.7 Módulo Bluetooth

Bluetooth é uma tecnologia de comunicação sem fio de baixo custo. Atualmente milhões de dispositivos possuem este protocolo de comunicação visando à substituição de fios e manter um alto nível de segurança entre a troca de informações (BLUETOOTH, 2013).

Segundo Oliveira, criado em 1998 pelas maiores potências industriais da época, foi lançando a ideia de Bluetooth como uma proposta mais acessível para a conectividade sem fio.

Este dispositivo trabalha por meio da comunicação via rádio, e opera na faixa de frequência que varia de 2.4 GHz a 2.485 GHz, por trabalhar nesta faixa de frequência várias perdas de sinais podem ser ocasionadas pelo fato desta faixa de rádio ser reservada para uso geral em aplicações industriais, científicas e médicas. Para contornar esta situação o sistema conta com o mecanismo frequency hopping, que troca a faixa de operação de rádio para amenizar interferência e perda de sinal (TELECO, 2013).

Acoplado ao Arduino por meio da comunicação serial, este módulo tem como principal função fazer a troca de informações entre o Arduino e o a placa Bluetooth presente na arquitetura do celular Nokia N95 8 Gb por meio do perfil de comunicação SPP, que emula uma comunicação serial para troca de informações, no caso da implementação toda a comunicação é feita com a passagem de códigos pré-definidos em formato de caracteres ou Strings.

O modem utilizado é o Modem Bluetooth BlueSMiRFGold fabricado pela SparckFun, funciona através da comunicação em série RX/TX podendo ser operado com velocidade de transmissão na faixa de 2400 a 115200 bps, possui um alcance de até 106 metros em espaços abertos sem obstáculos e sua alimentação pode ser realizada partindo de 3,3 V até 6 V em todos seus pinos.

A figura 11 mostra o esquema de ligação entre o modem e a placa Arduino.

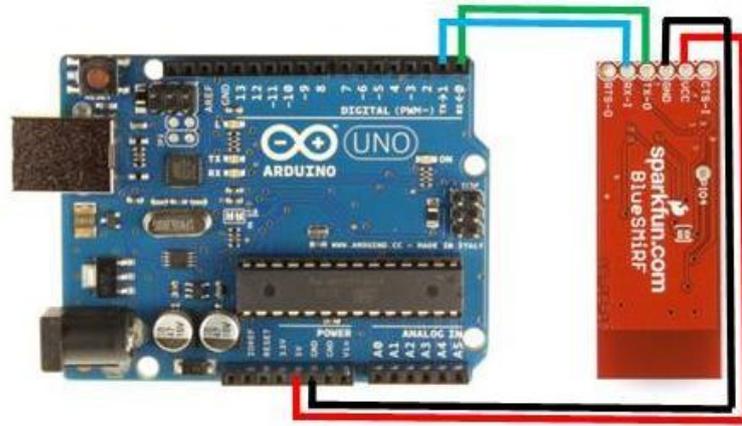


Figura 11 - Esquema de ligação entre Arduino e Bluetooth

Para iniciar os testes de comunicação foram desenvolvidos dois Scripts de comunicação, um hospedado na placa Arduino que faz a leitura da porta serial, caso algum dado for enviado a mesma captura estes dados e toma alguma ação, neste caso tratando-se de testes as informações são mostradas na própria IDE do Arduino e na sequência é encaminhada uma Sting de confirmação de recebimento. Já o segundo Script fica hospedado no celular Nokia n958 Gb e tem por obrigatoriedade fazer a conexão com a placa Bluetooth e caso a conexão seja estabelecida começa a enviar dados pela serial, no caso enviando os bits zero ou um e também fazer a leitura da resposta enviada pelo Arduino confirmando que os bits enviados foram capturados. Tanto o Script hospedado no Arduino quanto o hospedado no celular Nokia N95 8GB pode ser observados nos apêndices D e E respectivamente.

3.8 Considerações Finais do Capítulo

Neste capítulo foi apresentado individualmente os componentes pertinentes à arquitetura do Smart Walking Stick bem como cada funcionalidade desempenhada. Logo em seguida foi realizada a ligação para correto funcionamento dos módulos, o correto sincronismo entre os mesmos, garantindo que cada módulo realize a sua função para que em conjunto consigam proporcionar resultados satisfatórios para o usuário final.

CAPITULO 4 - TESTES, VALIDAÇÕES E RESULTADOS

Para assegurar o correto funcionamento de todas as funcionalidades implementadas e exibidas neste trabalho foram realizados vários testes e validações conforme descritas nessa seção.

4.1 Tempo de retorno da posição geográfica por meio do Módulo GPS

No momento em que o Script é iniciado no Celular Nokia N95 8 Gb foram realizados testes para averiguar qual é o tempo médio para obtenção das coordenadas geográficas. Foi atribuída uma variável do tipo inteira que é incrementada a medida que os segundos se passam. Na tabela 1 é mostrado o resultado dos testes realizados a céu aberto. A primeira coluna mostra o número de tentativas já a segunda o tempo de obtenção da coordenada geográfica pelo Módulo. Por fim é realizado o cálculo de média para obter a média de tempo de retorno da posição geográfica.

Tabela 1- Tempo de retorno da posição geográfica

TENTATIVA	TEMPO TOTAL (Seg.)
1	389
2	256
3	300
4	239
5	270
6	290
7	150
MÉDIA	270,57

O tempo até o retorno da coordenada geográfica varia a cada nova solicitação, lembrando que esta solicitação se trata da primeira solicitação, ou seja, quando o Script é executado pela primeira vez, depois de retornado a primeira posição geográfica as demais solicitações obtém retorno imediato.

Observou-se também que o ambiente influencia neste teste. Ao tentar obter a posição

geográfica em ambientes fechados o tempo passa dos 10.000 segundos diferente do que acontece quando o celular Nokia N95 8GB está localizado num ambiente a céu aberto.

4.2 Duração das Baterias

Conforme apresentado no capítulo da arquitetura do SWS, as baterias utilizadas são do tipo secundário que permite quando descarregadas a possibilidade de serem recarregadas evitando a sua substituição.

Na arquitetura é utilizado um par de baterias cada uma com a Capacidade de 1050 Mah, resultando num total de 2100 Mah. Hoje sabe-se que o consumo de bateria varia de acordo com a carga de utilização de circuitos, por exemplo, se tratar de um telefone celular a duração da bateria varia de acordo com a utilização: se está ligado a conexão de dados, Bluetooth ligado, frequência de ligações, todos estes fatores colaboram para que a duração da bateria seja reduzida ao contrário de quando o celular fica em modo de espera (Stand By), onde a duração da bateria se prolonga por muitas horas.

Como a arquitetura do SWS apresenta vários componentes eletrônicos interligados entre si, seguindo as informações fornecidas pelos fabricantes a tabela abaixo mostra a consumo médio de cada componente.

Tabela 2 - Consumo médio dos componentes

COMPONENTE	QUANTIDADE	CONSUMO TOTAL (Mah)
BLUETOOTH	1	25
SONAR	1	2
VIBRACAL	3	85
SAIDAS ARDUINO	1	320
NOKIA N95	1	225
LED	1	20
TOTAL		677

Estes valores se referem aos dispositivos quando expostos a frequência máxima de funcionamento, porém como estes componentes não trabalham com carga máxima o consumo

vária muito. Se as baterias conseguem fornecer 2100 Mah e o consumo médio dos componentes sendo 677 Mah, tem-se uma duração média de 3 horas e 20 minutos.

Porém não é o que acontece pelo fato dos dispositivos variarem a carga de trabalho como é o caso do GPS que necessita de mais energia quando contido num ambiente mais fechado. Outro fator que influencia no consumo é que a maioria dos dispositivos ligados às saídas do Arduino possui alimentação independente da placa que conseqüentemente reduz o consumo informado na tabela. Portanto, cada vez que se utiliza o dispositivo as baterias são recarregadas por completo e com auxílio de um cronometro conta-se o tempo até a carga se esgotar. Na maioria dos testes a mesma apresentou uma autonomia considerada boa, chegando próximo de 5 horas e 40 minutos.

4.3 Distância de Detecção

Foram realizados alguns testes para averiguar se a distância detectada pelo Módulo Sonar é correspondente a distância real. Para realizar este teste foi utilizada a placa Arduino ligado ao Módulo Sonar. Realizando o upload do código fonte para a placa Arduino utilizou-se uma trena e mediou-se a distância X de certo porto a uma parede localizada a sua frente. Anotando-se esta distância X, então o Módulo Sonar foi disposto exatamente a esta distância, observou-se a distância retornada pelo Módulo. Nos testes realizados a distância utilizada variou desde 50 até 550 cm pelo fato do sensor conseguir detectar objetos de 12 a 645 cm de distância.

Os resultados foram satisfatórios uma vez que a variação entre a distância real e a detectada variou em média 7 cm, tanto para distancias medidas de 1 metro, 2 metros, 3 metros e 5,5 metros.

4.4 Ativação do Modulo Alerta com limite de distância configurado

Nesta etapa foi testado se após configurar uma distância mínima pelo Módulo de Configuração de Cenário o Módulo Alerta seria ativado caso a distância mínima seja atingida. Como o deficiente visual não pode visualizar a tela do celular Nokia N95 8 Gb e a mesma serve apenas para testes, então o dispositivo foi posicionado frente a um obstáculo qualquer e configurou-se a distância mínima, deslocando-se o eixo central do potenciômetro para detectar objetos a esta distância. Diante do cenário, vários objetos foram inseridos e, portando

o dispositivo o usuário se locomoveu pelo cenário. Como já esperado o dispositivo conseguiu alertar que havia obstáculos a sua frente forçando o portador do SWS a desviar do obstáculo conseguindo alcançar um dos objetivos do trabalho.

4.5 Ativação do Módulo Alerta próximo a ponto de Interesse

Testes foram realizados para verificar se ao ser marcado um ponto de interesse num determinado caminho ao retroagir por este caminho o Módulo Alerta obrigatoriamente deveria ser ativado antes mesmo de o Módulo Sonar conseguir identificar o obstáculo. Para testar tal funcionalidade, o dispositivo foi testado numa rua onde uma lixeira presente na calçada foi marcada como ponto de interesse. Seguiu-se até o final da rua e foi feito o caminho contrário, e muito antes de chegar perto da lixeira o Módulo Alerta se ativou causando um pouco de confusão. O teste foi repetido da mesma forma e com as mesmas condições por algumas vezes e notou-se que com à medida que os testes eram realizados cada vez mais perto da lixeira o Módulo Alerta se ativava. A causa deste fato está ligado justamente à precisão do Módulo GPS, quando ao passar mais tempo exposto a céu aberto o número de satélites que fornecem a posição geográfica ao Módulo GPS é maior, o que leva a ter maior precisão e consequentemente conseguir ativar o Módulo Alerta com melhor eficiência.

4.6 Disponibilidade do Servidor para envio das informações via Módulo 3G

Conforme os testes realizados em uma região onde a disponibilidade de conexão 3G nem sempre é possível, na maioria das vezes a conexão estabelecida é do tipo 2G, que por sua vez tem a velocidade inferior, então foi criado um Script PyS60 que dentro de um loop envia várias requisições ao servidor Web, caso a resposta seja positiva é incrementado uma variável inteira (ContSucess), caso contrário outra variável de mesmo tipo é incrementada (ContFail). Para o laço de repetição (loop) foi atribuído um valores inteiro 10, 100 e 300, que representa a quantidade de repetições destas requisições. Ao final do laço são mostrados na tela os valores das variáveis indicando quantas requisições foram positivas e quantas foram negativas. A figura 12 mostra a tela do Nokia N95 8 Gb após a realização de três testes do mesmo tipo.

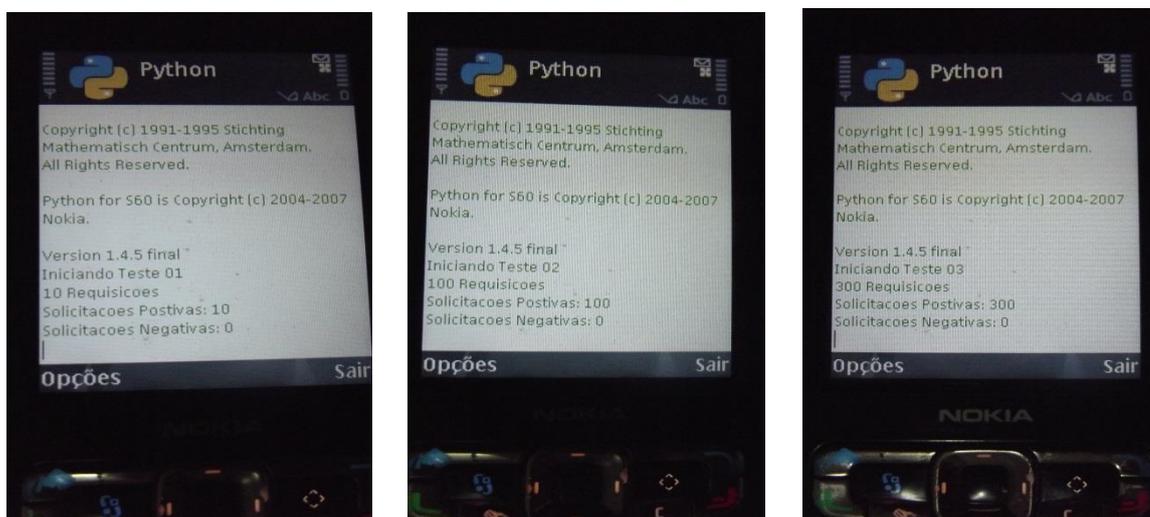


Figura 12 - Teste de requisições ao servidor

4.7 Funcionamento da Página Web

A página Web que mostra o rastreamento do dispositivo se mostrou estável, tanto quanto o acesso por meio de um computador, tablet ou celular. Foi testado se a atualização da posição exibida na tela pelo marcador é imediata, também se mostrou estável como a página. O tempo de testes foi medido a cada nova inserção de coordenadas geográficas enviadas pelo módulo 3G algo em torno de 4000 novas coordenadas capturadas e a cada nova inserção bem sucedida o marcador se atualizou sem problemas.

4.8 Marcação de Pontos Interessantes e Envio de SMS

Como já apresentando a arquitetura do SWS, a mesma dispõe de dois botões que ao serem pressionados executam funções distintas. O primeiro botão tem a função de marcar um ponto interessante que indicará que naquela posição existe um obstáculo. Então para saber se no momento em que o botão é pressionado a informação realmente esta sendo enviada a tabela que armazena as posições marcadas foi limpa, o módulo GPS foi desabilitado e foi estabelecida uma coordenada fixa para que somente fossem enviadas informações relativas a este ponto quando o botão fosse pressionado. Os resultados não foram diferentes dos já

esperados. A tabela 3 mostra o resultado do teste realizado. A primeira coluna mostra a quantidade de vezes consecutivas que o botão foi pressionado, a segunda coluna a quantidade de informações que foi registrada na tabela e pôr fim a terceira coluna o percentual de perdas.

Tabela 3 - Teste Módulo Colaborativo

VEZES PRECIONADO	INSERIDO	PERDA (%)
20	20	0
2	2	0
15	15	0
45	43	4,44
11	11	0
2	2	0
7	7	0

Os testes foram satisfatórios, pois, no primeiro teste teve-se uma perca de 4,44%, pois houve perda de comunicação com o servidor, já no segundo teste este valor foi zero.

Os testes para o segundo botão (que envia uma mensagem de socorro em caso de situação anômala) foi algo próximo ao realizado ao primeiro botão. Ao invés da tabela, a caixa de entrada de mensagens do celular cadastrado para receber as mensagens de socorro foi limpa e então o botão foi pressionado consecutivamente por 30 vezes com o intervalo de 5 segundos e se observou a quantidade de mensagens recebidas.

O teste foi repetido por quatro vezes e o botão foi pressionado 30, 12, 17 e 45 vezes respectivamente. Em todos os testes as quantidades de mensagens na caixa de entrada correspondiam exatamente à quantidade de vezes que o botão foi pressionado. Portando neste segundo teste não houve perda de informações.

4.9 Conexão entre os Módulos Bluetooth

Em conformidade aos testes realizados com Scripts utilizados (Apêndice D e E) para testar a comunicação entre os dois módulos Bluetooth observou-se que quando se realizava

um loop de troca informações entre os Bluetooth's, ou seja, o celular envia um bit, o modulo Bluetooth presente no Arduino recebe este bit e retorna com outro bit de resposta que o Módulo Bluetooth do celular Nokia N95 8Gb recebe e devolve outro bit para o outro módulo Bluetooth fechando o ciclo.

Este loop de troca de informações acarretava no travamento do Módulo Bluetooth por que congestionava o canal de comunicação SPP. Para solucionar este problema foi pré-estabelecido que o sincronismo entre os dois módulos ocorra a cada segundo corrigindo o problema.

CAPITULO 5 - CONCLUSÕES

O desenvolvimento da proposta promoveu o estudo de vários componentes eletrônicos, como é o caso do LV-Max Sonar – EZ4, o funcionamento de um Bluetooth, um pouco sobre servidores Web, bancos de dados e até a arquitetura de um simples celular, entre os demais periféricos interligados ao Arduino.

Com a interconexão destas tecnologias é possível tornar a vida de pessoas portadoras de deficiência mais agradável, promovendo a inclusão social das mesmas, com uma tecnologia que ao final do desenvolvimento seu custo financeiro apresenta-se reduzido conforme pode ser observado na tabela do Apêndice H, sendo assim várias classes sociais podem fazer aquisição do SWS.

Algumas partes ainda têm de ser melhoradas, por exemplo, a arquitetura e interconexão dos periféricos, visando promover um dispositivo mais atrativo, um estudo do maior tempo que o dispositivo ficará em funcionamento para calcular-se a capacidade necessária real das baterias para o dispositivo ficar mais leve e o estudo de novas tecnologias para melhor detecção de obstáculos em movimento.

O estudo de outras tecnologias relacionadas ao GPS será realizado para que o Módulo GPS seja cada vez mais preciso na detecção dos objetos. Quando o mesmo é exposto num local fechado fica impossível determinar a posição geográfica do dispositivo porque os sinais dos satélites não conseguem atingir o Módulo, tornando impossível determinar sua coordenada geográfica conseqüentemente não é possível obter seu endereço.

Por fim, como trabalhos futuros pretende-se integrar o SWS com um dispositivo microfone que por meio de comando de voz informará ao dispositivo o ponto inicial e final da rota a ser percorrida, então o mesmo irá calcular a melhor rota para a situação e ambiente iniciando a rota informando por comando de voz as instruções para que se consiga chegar ao destino com sucesso, vale lembrar que neste percurso todos os Módulos da arquitetura continuam funcionando para proporcionar maior segurança neste trajeto.

REFERÊNCIAS BIBLIOGRÁFICAS

AJAX. “w3schools.com, AJAX Tutorial”. Disponível em <<http://www.w3schools.com/ajax/>>. Acessado em 15 de novembro de 2012.

ALESSI, Alessandro. PASSOS, Ivan. RIBEIRO, Jader A. L. (2010) Disponível em <<http://www.pessoal.utfpr.edu.br/msergio/Monog-10-1-Bengala-eletronica.pdf>>. Acessado em: 30 out. 2012.

APACHE. “The Apache Software Foundation”. 2012. Disponível em <<http://www.apache.org/>>. Acessado em: 20 nov. 2012.

ARDUINO. “ARDUINO – PWM”. Disponível em <<http://arduino.cc/en/Tutorial/PWM>>. Acessado em: 11 jan. 2013.

ATMEL. “ATmega 328”. Disponível em <<http://www.atmel.com/devices/atmega328.aspx?tab=overview>>. Acessado em: 22 dez. 2012.

BERSCH, Rita, 2008. “Introdução à Tecnologia Assistiva”. Disponível em <<http://proeja.com/portal/images/semana-quimica/2011-10-19/tec-assistiva.pdf>>. Acessado em: 12 nov. 2012;

BLUETOOTH. “A Look at the Basics of Bluetooth Wireless Technology”. 2013. Acessado em <<http://www.bluetooth.com/Pages/Basics.aspx>>. Acessado em: 3 mar. 2013.

GOOGLE DEVELOPERS, “Google. API Java script do Google Maps V3”. 2013. Disponível em <<https://developers.google.com/maps/documentation/javascript/?hl=pt-br>>. Acessado em: 1 jan. 2013.

FAPESP. “Facilidades para deficientes visuais”. Disponível em <<http://revistapesquisa.fapesp.br/2013/03/15/facilidades-para-deficientes-visuais/>>. Acessado em: 1 maio 2013.

IBGE – Instituto Brasileiro de Geografia e Estatística, Censo 2010. Disponível em <ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2010/Caracteristicas_Gerais_Religiao_Deficiencia/caracteristicas_religiao_deficiencia.pdf em 20/05/2013>. Acessado em: 22 abr. 2013.

Instituto Benjamin Constant “Os Conceitos de Deficiência”. Disponível em <<http://www.ibc.gov.br/?itemid=396>>. Acessado em: 21 maio 2013.

JAVASCRIPT. “JAVASCRIPT KIT”. Disponível em <<http://www.javascriptkit.com/>>. Acessado em: 25 nov. de 2012.

JQUERY. “jQuerywriteless, do more.”. Disponível em <<http://jquery.com/>>. 2013. Acessado em: 27 nov. de 2012.

JSON. “Introduction JSON”. Disponível em <<http://www.json.org/>>. Acessado em: 25 nov. 2012.

MAXBOTIX. “EZ-MaxSonar-EZ4 Datasheet”. 2005. Disponível em: <<http://www.maxbotix.com/uploads/LV-MaxSonar-EZ1-Datasheet.pdf>>. Acesso em: 01 fev. 2013.

MCROBERTS, Michael. “Arduino Básico”. 1ª Ed., São Paulo: Novatec Editora, 2010.

MD-Soft. “Python for S60”. Disponível em <http://www.md-soft.info/en/python_for_s60.html#a762>. Acessado em: 12 dez. 2012.

MYSQL. “MySQL, The world’s most popular open source database”. Disponível em <<http://www.mysql.com/>>. Acessado em: 20 nov. 2012.

NETWORKS, Roving. “AdvancedUser Manual: rn-bluetooth-um Version 4.77”. 2009. Disponível em: <<https://www.sparkfun.com/datasheets/Wireless/Bluetooth/rn-bluetooth-um.pdf>>. Acesso em: 5 nov.2012.

NOKIA. “Especificações detalhadas para o Nokia N95”. Disponível em <<http://www.nokia.com/br-pt/produtos/celular/n95/especificacoes/>>. Acessado em: 2 jan. 2013.

NOKIA DEVELOPER. “NOKIA Developer”. Disponível em <http://developer.nokia.com/Community/Wiki/Archived:Introdu%C3%A7%C3%A3o_ao_Py_Symbian>. Acessado em: 25 dez. 2012.

OLIVEIRA, Ricardo A. Rabelo. “Bluetooth e Multimídia”. Disponível em <<http://laplace.dcc.ufmg.br/npdi/uploads/c8962954-8be0-43ed.pdf>>. Acessado em: 23mar. de 2013.

PHP. “PHP: Hypertext Preprocessor”. Disponível em <<http://php.net/>>. 2013. Acessado em: 20 nov. 2012.

SCHEIBLE, Jurgen. “Mobile Python – Rapidprototypingofapplicationsonthe mobile platform”. England: John Wiley& Sons, Ltda., 2007.

SEFRIN, Felipe S. (2012) “SACCOB: Sistema de Alerta Contra Colisão com Obstáculos”. Disponível em <http://www.inf.unioeste.br/~tcc/2012/TCC_Felipe_S.pdf>. Acessado em: 30 out. 2012.

SOCIEDADE BRASILEIRA DE QUÍMICA. “Pilhas e Baterias: Funcionamento e Impacto Ambiental”. Disponível em <<http://qnint.sbq.org.br/qni/visualizarConceito.php?idConceito=45&alterarIdioma=sim&novoIdioma=pt>>. Acessado em: 2 fev. 2013.

SOURCEFORGE. “Source Force, PyS60”. Disponível em <<http://sourceforge.net/projects/pys60/files/pys60/1.4.5/>>. Accessed me: 25 out. 2012.

SPARKFUN. “Sparkfun, Bluetooth Modem – Blue SMIRF Gold”. Disponível em

<<https://www.sparkfun.com/products/10268>>. Acessado em: 22 fev. 2013.

TELECO. “Teleco, Inteligência em Telecomunicações”. 2013. Disponível em <http://www.teleco.com.br/tutoriais/tutorialss/pagina_2.asp>. Acessado em: 5 maio 2013.

THOMAZINI, Daniel. “Sensores Industriais Fundamentos e Aplicações”. São Paulo: Érica, 2008.

WAMPSEVER. “WAMPSEVER, a Windows web development environment”. Disponível em <<http://www.wampserver.com/en/>>. Acessado em: 20 nov. 2012.

WIKIPÉDIA. “Motor Elétrico”. 2013. Disponível em <http://pt.wikipedia.org/wiki/Motor_el%C3%A9trico#Tipos_de_motores>. Acessado em: 5 maio 2013.

APÊNDICE A – CÓDIGO FONTE SERVIDOR WEB SWS

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="cache-control" content="max-age=0" />
  <meta http-equiv="cache-control" content="no-cache" />
  <meta http-equiv="expires" content="0" />
  <meta http-equiv="expires" content="Tue, 01 Jan 1980 1:00:00 GMT" />
  <meta http-equiv="pragma" content="no-cache" />
  <title>Smart Walking Stick</title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no"/>
<style type="text/css">
html { height: 100% }
body { height: 100%; margin: 0; padding: 0 }
  #map-canvas { height: 100% }
</style>
<script type="text/javascript" src="jquery-1.9.1.js"></script>

  <script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAqNl2RtVE0NiMl-
dsVqZEmWETtClTDiUk&sensor=false">
</script>
<script type="text/javascript">
var latLng1;
  var map;
  varlatLng;
  var marker;
  varlatini;
  varlogini;
  var Mercado = 0;
  varlatmkr;
  varlogmkr;
  var lati;
  varlongi;

  //Quando o Documento HTML estiver carregado
jQuery(document).ready(function() {

```

```

        //Ao clicar em um elemento do tipo button
        jQuery("#button").click(function(){
            //Requisição Ajax
            jQuery.ajax({
url: "http://smartwalking.zapto.org:4000/pagina.php", //URL de destino
dataType: "json", //Tipo de Retorno
success: function(json){ //Se ocorrer tudo certo
                lati =json.lat;
                longi = json.long;

                Marca()

            }
        });
    });
});

functionclickButton()
{
    document.getElementById('button').click();
}

setInterval(function(){clickButton()},3000);

vargeocoder = new google.maps.Geocoder();
functiongeocodePosition(pos) {
    geocoder.geocode({
        latLng: pos
    }, function(responses) {
        if (responses &&responses.length> 0) {
            updateMarkerAddress(responses[0].formatted_address);
        }
    });
}

functionupdateMarkerAddress(str) {

```

```
marker.setTitle(str);
    }

function initialize() {

    //posição inicial
    latini = -22.2325370436;
    logini = -49.9666965934;
    latLng = new google.maps.LatLng(latini,logini);
    varmapOptions = {

        center: latLng,
        zoom: 14,
        scaleControl: true,
        mapTypeId: google.maps.MapTypeId.SATELLITE

    };

    map = new Google.maps.Map(document.getElementById("map-
    canvas"),mapOptions);
    map.setTilt(45);

    }

    google.maps.event.addDomListener(window, 'load', initialize);

functionMarca() {

    latmkr = lati;
    logmkr = longi;
    latLng1 = new google.maps.LatLng(latmkr,logmkr);
    geocodePosition(latLng1);
    if (Marcado == 0){

        marker = new
        google.maps.Marker({position:
        latLng1,map: map,title:
        "User",animation:
        google.maps.Animation.DROP});
        google.maps.event.addListener(marker,
        'position_changed', function() {
```

```
        window.setTimeout(function() {
            map.panTo(marker.getPosition());
        }, 1000);
    });
    google.maps.event.addListener(marker,
    'click',function() {
        map.setZoom(17);map.setCenter(marker.getPosition());
    });
    Marcado =1;
    }
else{
    marker.setPosition(latLng1);
}

</script>
</head>
<body>

    <input type="button" id="button" onclick="Marca()" value="Atualizar"
    style="display: none;" />
    <div id="map-canvas" style="width: 100%; height: 100%"/></div>

</body>
</html>
```

APÊNDICE B – CÓDIGO FONTE PLACA ARDUINO

```

int pwPin = 11; // PINO DO SONAR
int pinm = 2; //PINO MOTOR 1
int pinm2 = 13; //PINO MOTOR 2
int pinm3 = 7; //PINO MOTOR 3
int num1,num2,num3;
int ledGPS = 12; //PINO LED GPS QUANDO CONECTADO
int ligaled;
int button1State = 0; //VARIAVEIS PARA CONTROLAR ESTADO DO BOTAO
int button2State = 0;
const int button1Pin = 8; //PINOS DOS BOTOES
const int button2Pin = 9;
int travabt01 = 0; // TRAVA PARA OS BOTOES
int travabt02 = 0;
long pulse, inches, cm; // VARIAVEIS PARA CALCULAR A DISTACIA

void mededistancia() { //FUNÇÃO QUE CALCULA A DISTANCIA
                        //DETECTADA

    pulse = pulseIn(pwPin, HIGH); //CASO A DISTANCIA SEJA MENOR QUE
                                    //O VALORESTABELECIDO PELO

    //147uS per inch // POTENCIOMETRO ATIVA OS 3
                        //MOTORES ALERTAS

    inches = pulse/147;
    //change inches to centimetres
    cm = inches * 2.54;

    if (cm < num1){
        digitalWrite(pinm3,HIGH); //FAZ A VERIFICAÇÃO PARA ATIVAR OS MOTORES
        digitalWrite(pinm2,HIGH);
        digitalWrite(pinm,HIGH);}
    else{
        digitalWrite(pinm,LOW);
        digitalWrite(pinm2,LOW);
        digitalWrite(pinm3,LOW);}
    }

void distmaior() {
int sensorValue = analogRead(0); //FAZ A LEITURA DO VALOR DA RESISTENCIA
                                    // VINDA PELO

```

```

num1 = sensorValue/2;          //POTENCIOMETRO (0 - 1024) DEPOIS DIVIDE
                                POR 2 E SOMA-SE 20PARA FICAR NA FAIXAR

num1 = num1+20;                // DE DETECÇÃO DO SONAR

}

void setup() {
  Serial.begin(9600);          //INICIALIZAÇÃO DOS PINOS, COMUNICAÇÃO SERIAL E
  pinMode(pinm,OUTPUT);       //VARIÁVIES
  pinMode(pinm2,OUTPUT);
pinMode(pinm3,OUTPUT);
  pinMode(pwPin, INPUT);
  pinMode(ledGPS,OUTPUT);
  pinMode(button1Pin, INPUT);
  pinMode(button2Pin, INPUT);
  travabt01 = 0;
  travabt02 = 0;
}

void loop() {

  if (Serial.available()) {    //FAZ LEITURA DOS DADOS VINDOS VIA SERIAL,
    ligaled = Serial.read();   //CASO 1 O MOSULO GPS JÁ CONECTOU
  if (ligaled == '1')
    digitalWrite(ledGPS,HIGH);
  if (ligaled == '0')
    digitalWrite(ledGPS,LOW);
  if (ligaled == 'PO')
    {
      digitalWrite(pinm3,HIGH); //CASO VENHA PO ELE ATIVA OS
      digitalWrite(pinm2,HIGH); //MOTORES POIS ESA PASSANDO PERTO
      digitalWrite(pinm,HIGH);  //DEUM PONTO DE INTERESSE
    }

  }

  button1State = digitalRead(button1Pin); //LE O ESTADO DO BOTAO
  button2State = digitalRead(button2Pin); //(PRECIONADO OU NÃO)

  if (button1State == HIGH && travabt01 == 0) { //VERIFICA SE FOI

```


APÊNDICE C – CÓDIGO FONTE MÓDULO 3G, GPS E MÓDULO BLUETOOTH

```

import positioning, appuifw, e32
import os.path
import socket
import httplib, urllib
import messaging
import location

positioning.select_module(positioning.default_module())
positioning.set_requestors([{"type": "service", "format": "application", "data": "test_app"}])

arquivo = 'C:/DadosGPS.txt'

def c(data):
    if data['position']['latitude'] ==
data['position']['latitude']:
        mylatitude = data['position']['latitude']
        mylongitude = data['position']['longitude']
        if mylatitude != 'NaN':
            arq = open('C:/DadosGPS.txt', 'w')
            global lat
            global longi
            lat = str(mylatitude)
            longi= str(mylongitude)
            texto =
str(mylatitude)+str(mylongitude)
            arq.write(texto)
            arq.close()
            sock.send('1')
            params = urllib.urlencode({'lati': lat,
'longi': longi})
            headers = {"Content-type":
"application/x-www-form-urlencoded", "Accept": "text/plain"}
            conn =
httplib.HTTPConnection("smartwalking.zapto.org:4000")
            conn.request("POST", "/set_text.php",
params, headers)

```

```

        conn.close()
    else:
        sock.send('0')

class MinhaAplicacao(object): # Define uma classe MinhaApp

    def __init__(self):
        # Cria um objeto "trava" irÃ; "segurar"
        # a nossa aplicaÃ$Ãfo rodando
        self.trava = e32.Ao_lock()

    def conecta(self):

        global sock
        global apo
        sock=socket.socket(socket.AF_BT,socket.SOCK_STREAM)
        target=('00:06:66:08:61:AA',1)
        sock.connect(target)
        print 'Bluetooth ok!'
        apo = socket.access_point(5)
        socket.set_default_access_point(apo)
        apo.start()
        print 'Internet ok!'

    def enviasms(self):

        arq = open('C:/DadosGPS.txt', 'r')
        texto = arq.read()
        print(texto)
        arq.close()
        data = 'Preciso de ajuda. Estou em: '+texto
        print data
        nbr1 = "XXYYYYYYYYY" #NUMERO DO TELEFONE COM DDD
        txt = data
        messaging.sms_send(nbr1, txt)
        print 'Mensagem enviada'

```

```

def leserial(self):

    ch = sock.recv(1)
    print ch
    if ch == '0':
        self.enviasms(self)
    if ch == '1':
        enviaservidor(self)

def legps(self):

    positioning.select_module(positioning.default_module())

positioning.set_requestors([{"type":"service","format":"application","data"
:"gps_app"}])

    positioning.position(callback=c,
interval=1000000,partial=0)
    print "Iniciando..."

def enviaservidor(self):
    try:
        num1 = lat;
        num2 = longi;
        params = urllib.urlencode({'lati': num1, 'longi':
num2})
        headers = {"Content-type": "application/x-www-form-
urlencoded","Accept": "text/plain"}
        conn =
httplib.HTTPConnection("smartwalking.zapto.org:4000")
        conn.request("POST", "/set_text.php", params, headers)
        conn.close()
    except:
        print 'Falha ao enviar dados ao Servidor'

def executa(self):
    self.conecta()

```

```
self.location()
self.legps()
while 1 :

    self.leserial()

appuifw.app.exit_key_handler = self.sai
self.trava.wait()

def location(self):

    (self.mcc, self.mnc, self.lac, self.cellid) =
location.gsm_location()

    conn =
httpplib.HTTPConnection("smartwalking.zapto.org",4000)
    conn.request("GET",
"/posi_cel.php?hex=0&mcc="+str(self.mcc)+"&mnc="+str(self.mnc)+
"&lac="+str(self.lac)+"&cid="+str(self.cellid)+"")
    r1 = conn.getresponse()
    print r1.status, r1.reason

def sai(self):
# envia o sinal para "trava"
    apo.stop()
    sock.send('0')
    sock.close()
    self.trava.signal()

if __name__ == '__main__':
    aplicacao = MinhaAplicacao()
    aplicacao.executa()
```

APÊNDICE D – CÓDIGO FONTE TESTE: LEITURA E ENVIO DE DADOS PELA PORTA SERIAL ARDUINO

```
int bitrecebido = 0; //variável inteira que receba o bit via serial
void setup() { //método de inicialização

Serial.begin(9600); // abre a comunicação serial com velocidade de
                    9600 Bps

}

void loop() { //método de repetição

    if (Serial.available() > 0) { //verifica se há dados chegando pela
serial, caso sim (maior que 0)
bitrecebido = Serial.read();// captura esse bit
    Serial.print("Dado recebido: ");//imprime Dado recebido:
    Serial.println(bitrecebido, DEC);//imprime o bit na forma decimal
Serial.print("Bit recebido com sucesso!"); //envia string de
confirmação de recebimento pela porta serial
delay(1000); //espera 1 segundo.

}
}
```

APÊNDICE E – CÓDIGO FONTE TESTE: CONEXÃO, ENVIO E RECEBIMENTO DE BITS PELO CELULAR NOKIA N95

```
import socket
global sock

defconecta():
    sock = socket.socket(socket.AF_BT, socket.SOCK_STREAM)
    target=('00:06:66:08:61:AA',1)
    sock.connect(target)
    print 'Conexao Ok!'
    enviasecial()

defleserial():
    retorno = sock.recv(255)
    print retorno
    fechacon()

defenviasecial():

    sock.send('1')
    sock.send('0')
    leserial()

deffechacon():
    sock.close()

conecta()
```

APÊNDICE F – CÓDIGO FONTE TESTE: MEDIDO A DISTÂNCIA DE UM OBJETO COM O SONAR MAXSONAR EZ4

```
intpwPin = 11; // PINO DO SONAR
longpulso, polegada, cm; // VARIÁVEIS PARA CALCULAR A DISTANCIA

voidmededistancia(){ //método para calcular a distancia
pulso = pulseIn(pwPin, HIGH); //faz a leitura na porta pwm
//147uS por polegada Segundo datasheet
polegada = pulso/147;
//converte polegada para centímetros
cm = polegada * 2.54;
//imprime a distancia
Serial.print(cm);
Serial.print(" de distancia");
//pula uma linha a cada leitura
Serial.println();

}

void setup() { //metodo de inicilaização

Serial.begin(9600); //abre a serial
pinMode(pwPin, INPUT); //declara a pino sendo de entrada para ler o
MaxSonar

}

void loop() { //metodo de repetição que faz as leitura

mededistancia()

}

}
```

APÊNDICE G – CÓDIGO FONTE TESTE: MEDIDO A DISTÂNCIA DE UM OBJETO COM O SONAR MAXSONAR EZ4 E ACIONAMENTO MOTOR VIBRA

```

intpwPin = 11; // PINO DO SONAR
long pulso, polegada, cm; // VARIÁVEIS PARA CALCULAR A DISTANCIA
intpinm = 2; //PINO MOTOR

voidmededistancia(){ //método para calcular a distancia
pulso = pulseIn(pwPin, HIGH); //faz a leitura na porta pwm
    //147uS por polegada Segundo datasheet
polegada = pulso/147;
    //converte polegada para centímetros
    cm = polegada * 2.54;

if (cm < 100) { //se distancia menor que 1 metro
    digitalWrite(pinm,HIGH); //ativa o pino motor vibra
    }
else{
    digitalWrite(pinm,LOW); //deixa o pino desativado, motor nao vibra
    }

}

void setup() { //metodo de inicilaização

Serial.begin(9600); //abre a serial
pinMode(pwPin, INPUT); //declara a pino sendo de entrada para ler o
MaxSonar
pinMode(pinm,OUTPUT); //declara o pino sendo uma saída para ativar o
vibracall

}

void loop() { //metodo de repetição que faz as leitura
mededistancia()
}

```

APÊNDICE H – TABELA DE CUSTO DO PROJETO

Componente	Valor R\$
Arduino Uno R3	55,00
Nokia N95 8 Gb	180,00
BlueSmirf Gold	70,00
Motores Vibracall	15,00
Bateria Íons de Lítio	15,00
Botões	2,00
Potenciometro	1,50
LV-Max Sonar EZ4	150,00
Fios e Conectores	10,0