

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUIS GUILHERME MACHADO CAMARGO
PEDRO ALBERTO DE BORBA
RICARDO FARAH
STEFAN CAMPANA FUCHS
TELMO FRIESEN

MAPEAMENTO DE AMBIENTES COM O ROBÔ BELLATOR

ANÁLISE TECNOLÓGICA

CURITIBA

2013

LUIS GUILHERME MACHADO CAMARGO
PEDRO ALBERTO DE BORBA
RICARDO FARAH
STEFAN CAMPANA FUCHS
TELMO FRIESEN

MAPEAMENTO DE AMBIENTES COM O ROBÔ BELLATOR

Análise tecnológica apresentada à Unidade Curricular de Oficina de Integração 3 do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação.

CURITIBA

2013

SUMÁRIO

1 ANÁLISE TECNOLÓGICA	3
1.1 VISÃO GERAL DO PROJETO	3
1.2 REQUISITOS	4
1.2.1 Estação base	4
1.2.2 Sistema de comunicação	5
1.2.3 Sistema embarcado	5
1.3 ANÁLISE DE OPÇÕES TECNOLÓGICAS	6
1.3.1 Estação base	6
1.3.1.1 Biblioteca para desenhos 2D	6
1.3.1.2 Linguagem de programação	7
1.3.1.3 Sistema operacional	10
1.3.2 Sistema de comunicação	10
1.3.3 Sistema embarcado	12
1.3.3.1 Imagens instantâneas do ambiente	12
1.3.3.2 Movimentação do robô	13
1.3.3.3 Odometria	14
1.3.3.4 Detecção de obstáculos	18
1.3.3.5 Microcontrolador	20
1.3.3.6 Placa de circuito impresso	21
REFERÊNCIAS	23

1 ANÁLISE TECNOLÓGICA

Nesta seção está explicitada, primeiramente, uma visão geral do projeto. Em seguida, há uma discussão detalhada a respeito dos requisitos de cada parte fundamental – estação base, sistema de comunicação e sistema embarcado. Por fim há uma enumeração das alternativas tecnológicas pesquisadas e das escolhidas para o preenchimento dos requisitos.

1.1 VISÃO GERAL DO PROJETO

O projeto, de um ponto de vista geral, consiste em um robô controlado manualmente e que seja capaz de efetuar mapeamento em duas dimensões de ambientes. Um usuário humano monitorará e controlará um computador – a estação base – a partir do qual poderão ser enviados comandos de movimentação, via teclado, ao robô. Informações sobre o posicionamento do robô e dos obstáculos detectados por ele serão recebidas na estação base em tempo real. Imagens instantâneas de uma câmera posicionada no robô – aspecto explicado mais à frente – poderão ser visualizadas pelo utilizador.

O sistema de comunicação deverá ter, ao menos, alcance de 20 metros sem fios. Visto que toda a comunicação entre a estação base e o sistema embarcado será feita por um único canal, a velocidade e o tipo de fluxo de transmissão de dados devem ser adequados para, simultaneamente, o envio de comandos de movimentação ao robô, recebimento de dados de leituras de sensores e recebimento de imagens da câmera.

O sistema embarcado é constituído, em suma, pelo robô. Ele deve ser capaz de se mover para frente e para trás e girar para a esquerda e direita. A visualização em tempo real do ambiente pelo utilizador, com o objetivo de facilitar o controle de movimentação manual, poderá ser feita através de imagens instantâneas geradas por uma câmera fixa instalada no robô.

O robô deve ser capaz de obter dados para cálculos (na estação base) de sua velocidade e deslocamento. Um aspecto desejável em relação a isso é a atenuação de erros em decorrência de escorregamento, giros em falso ou trepidação de rodas, visando, dessa forma, a

utilização futura do robô em condições não ideais de terreno. Obstáculos próximos – em uma distância mínima de 30 cm e máxima de 150 cm – devem ser detectados de modo a possibilitar a confecção de um mapa 2D em tempo real na estação base.

1.2 REQUISITOS

1.2.1 Estação base

Esta seção descreve os requisitos da estação base, que foram elaborados de forma a satisfazer os objetivos do projeto.

- O *software* será executado em um computador pessoal.
 - O programa poderá ser executado em computadores pessoais de desempenho médio (de acordo com os padrões atuais). Não haverá necessidade de uma máquina de alto desempenho e custo relativo para executar o *software*.
 - O *software* primariamente será executado em um único sistema operacional, sendo este Linux ou Windows. É desejável que o desenvolvimento (incluindo a escolha das ferramentas) seja feito de forma a simplificar o uso multiplataforma do *software* futuramente.
 - Preferencialmente bibliotecas e ferramentas livres (e gratuitas) deverão ser utilizadas no desenvolvimento do *software*.
- O *software* deve possuir uma interface gráfica.
 - Um utilizador, através da interface gráfica, será capaz de controlar o robô enviando comandos de movimentação especificados pelo teclado.
 - O usuário receberá a imagem em tempo real de uma câmera fixa instalada no robô.
 - Os dados instantâneos de velocidade e posição do robô serão mostrados ao usuário na interface gráfica.
 - Um mapa 2D do caminho percorrido e dos obstáculos detectados pelo robô será gerado, na interface gráfica, à medida em que houver movimentação do mesmo. O caminho percorrido pelo robô será representado visualmente por pontos, gradualmente posicionados no mapa. Os obstáculos serão representados por marcações nas localidades onde houve detecção de objetos por sensores do robô. Todos os pontos representados no mapa serão gerados a partir de amostras obtidas em intervalos de tempo discretos.

- O mapa 2D gerado na interface gráfica poderá ser salvo em um arquivo, podendo ser posteriormente carregado.

1.2.2 Sistema de comunicação

Esta seção descreve os requisitos do sistema de comunicação entre a estação base e o sistema embarcado.

- Distância entre robô e estação base.
 - O sistema de comunicação deve possuir, ao menos, alcance de 20 metros sem fios – de modo que ambientes de tamanho razoável possam ser mapeados.
- Velocidade e direção do fluxo de transmissão de dados.
 - Toda a comunicação entre a estação base e o sistema embarcado será feita por um único canal sem fios e, portanto:
 - A velocidade de transmissão do canal de comunicação deve ser suficiente para, simultaneamente, o envio de comandos de movimentação ao robô, recebimento de dados de leituras de sensores e recebimento de imagens da câmera;
 - O fluxo de dados deve ser bidirecional (*full-duplex*).
- Protocolo de transporte.
 - A tecnologia utilizada para a comunicação deve permitir fácil utilização do protocolo de transporte TCP. Como as leituras de sensores devem obrigatoriamente ser recebidas na estação base na mesma ordem em que forem enviadas pelo robô (e também os comandos de movimentação enviados pela estação base devem chegar ao robô em ordem), o uso desse protocolo de transporte simplificará muito a implementação do protocolo de aplicação ponto a ponto. O TCP possui ainda outro aspecto interessante: além de garantir a ordem de chegada, existem mecanismos de detecção de perdas de pacotes – que efetuam o reenvio destes caso necessário.

1.2.3 Sistema embarcado

Esta seção descreve os requisitos do sistema embarcado (robô).

- Imagens instantâneas do ambiente.

- O robô, através de uma câmera fixa, deverá ser capaz de enviar à estação base imagens instantâneas do ambiente onde ele se encontra.
- Movimentação do robô.
 - O robô deve ser capaz de mover-se para frente, para trás e girar para a esquerda e direita.
- Controle de posicionamento e velocidade.
 - O robô deve ser capaz de obter dados que permitam calcular sua velocidade e posição atual (deslocamento e sentido em relação à posição inicial). Deve ser capaz de enviar os dados à estação base.
- Detecção de obstáculos.
 - O robô deverá ser capaz de detectar obstáculos próximos – com distância de no mínimo 30 cm e no máximo 150 cm – localizados ao seu redor, determinando a distância de cada objeto detectado.

1.3 ANÁLISE DE OPÇÕES TECNOLÓGICAS

Nesta seção está apresentada a análise das opções tecnológicas plausíveis para o atendimento dos requisitos. As alternativas pesquisadas e as escolhidas para cada parte do projeto estão explicitadas a seguir.

1.3.1 Estação base

As alternativas pesquisadas para a estação base estão apresentadas nesta subseção.

1.3.1.1 Biblioteca para desenhos 2D

Tendo em vista que um dos requisitos é a geração de um mapa em duas dimensões na estação base, deve-se escolher uma biblioteca que permita realizar o desenho de formas geométricas em 2D (via código-fonte) e que possa ser integrada facilmente à interface gráfica. Ela deve também possuir meios simples de obter informações do mouse e teclado, para interatividade com o usuário.

Uma biblioteca interessante disponível em Java que possui o recurso de produzir desenhos dinâmicos com integração a interfaces gráficas é o Processing (PROCESSING, 2013), de

código livre. Essa biblioteca foi a principal encontrada que seria capaz de satisfazer as necessidades de desenho do mapa 2D de forma simples. Por possuir inúmeras funções de desenho em alto nível, o trabalho de renderização dos gráficos seria consideravelmente simplificado. Além disso, na biblioteca existem recursos que permitem o recebimento de informações de posicionamento do mouse e de comandos do teclado. Por ser constituído basicamente de um *Applet* Java, o Processing pode facilmente ser integrado a componentes do Swing – biblioteca de interface gráfica (GUI) do Java.

Outra biblioteca para a confecção de desenhos em 2D é o Cairo (CAIRO, 2013), que é *open-source*, disponível nas linguagens C e C++. Ele possui recursos em alto nível para renderização de formas e interação com o usuário, assim como o Processing. Nos aspectos gerais as duas ferramentas são muito semelhantes. A integração do Cairo com a interface gráfica, porém, é dependente na biblioteca externa de GUI utilizada.

Um aspecto importante a notar é que ambas as bibliotecas foram desenvolvidas e otimizadas para terem bom desempenho em máquinas atuais – o que é desejável tendo em vista os requisitos. Na Tabela 1 está presente uma comparação entre as duas bibliotecas.

Tabela 1: Comparação entre Bibliotecas para desenhos 2D.

Característica	Cairo	Processing
Linguagem de programação	C e C++	Java
Integração com interface gráfica	Sim (depende da biblioteca de GUI utilizada)	Sim (com a biblioteca Swing do Java)
Ferramentas de interação com o usuário	Sim	Sim

Escolha da equipe: O Processing foi adotado como solução para desenhos 2D. Deve-se ressaltar que a escolha da biblioteca de desenhos foi feita em conjunto com a escolha de linguagem de programação, tendo em vista a interdependência de ambas. Em razão dos motivos apresentados na próxima subseção – que se constituem de principal argumento para a escolha do Processing – o Java foi adotado. Um ponto interessante do Processing é a simplicidade de integração a interfaces gráficas (Swing) do Java.

1.3.1.2 Linguagem de programação

Na etapa de avaliação das opções tecnológicas, a escolha de uma boa linguagem de programação que atenda aos requisitos é fundamental. Abaixo está presente uma lista dos aspectos desejáveis da linguagem a ser utilizada neste projeto:

- Deve ser multiplataforma (ao menos compatível com Linux e Windows sem muitas modificações);
- Deve possuir orientação a objetos;
- Deve possuir recursos multiplataforma e de código livre para o desenvolvimento de interfaces gráficas;
- Deve ter a disponibilidade de ferramentas de código livre e multiplataforma para a criação *visual* da interface gráfica, dessa forma agilizando o processo de desenvolvimento;
- Deve possuir recursos, integrados ou em bibliotecas externas de código livre, para o desenvolvimento de desenhos dinâmicos (para a geração do mapa 2D). Os desenhos devem ser facilmente integráveis à interface gráfica.

Abaixo está presente uma descrição de duas linguagens, o C++ e Java, atualmente utilizadas em inúmeras aplicações, e que são potenciais alternativas ao projeto – tendo em vista a experiência e conhecimento dos integrantes a respeito de ambas. A Tabela 2 sumariza os recursos de cada uma.

Java

O Java (JAVA, 2013) é uma linguagem concebida de início como sendo orientada a objetos. A maneira com que é feita a compilação e execução do código permite que, muito facilmente, programas sejam rodados em diferentes plataformas (Linux, Windows, Mac, entre outros). O processo de compilação do código gera os chamados *bytecodes*, que são instruções a serem interpretadas pela *Java Virtual Machine* (JVM). A grande vantagem é que o JVM possui disponibilidade multiplataforma, e a manutenção dele pelos desenvolvedores é frequente.

Há disponibilidade, na API do Java, da biblioteca Swing – que contém recursos completos para a criação de interfaces gráficas (GUI) interativas. Existem ferramentas visuais de código aberto que consideravelmente agilizam o processo de desenvolvimento de interfaces Swing, entre elas o NetBeans (NETBEANS, 2013) e o Eclipse (ECLIPSE, 2013), através de plugins ou extensões.

Para o preenchimento do requisito de confecção de desenhos em 2D com integração à interface gráfica, a biblioteca do Processing (explicada anteriormente na Subseção 1.3.1.1) está disponível nessa linguagem.

C++

O C++ é uma linguagem orientada a objetos, que foi desenvolvida a partir da linguagem C. A compilação de código no C++ deve ser feita especificamente para cada plataforma

em que um programa desenvolvido for utilizado. De uma perspectiva prática, certas seções de código frequentemente necessitam de adaptações manuais para cada plataforma e sistema operacional, o que gera retrabalho e gastos de tempo adicionais.

Recursos para desenvolvimento visual de interfaces gráficas estão disponíveis através de bibliotecas e ferramentas externas. O C++ não possui recursos de interface gráfica na própria API e, portanto, deve-se notar que este é um aspecto complicador ao portar programas entre diferentes sistemas.

Para a confecção de desenhos 2D e incorporação dos mesmos à interface gráfica, a biblioteca Cairo (explicada anteriormente na Subseção 1.3.1.1) pode ser utilizada com essa linguagem. A possibilidade de haver integração com a interface, porém, é dependente da biblioteca de GUI utilizada.

Tabela 2: Comparação entre linguagens de programação.

Característica	C++	Java
Multiplataforma (Linux e Windows)	Sim (com adaptação)	Sim (sem adaptação)
Orientação a objetos	Sim	Sim
Recursos multi-plataforma e <i>open-source</i> para desenvolvimento de interface gráfica (GUI)	Sim (com bibliotecas externas)	Sim (integrado à API da linguagem)
Ferramentas <i>open-source</i> e multi-plataforma para criação visual de interface gráfica	Sim (ferramentas externas)	Sim (ferramentas externas)
Recursos <i>open-source</i> para desenvolvimento de desenhos dinâmicos, facilmente integráveis à interface gráfica	Sim (biblioteca externa, integração à interface gráfica dependente da GUI utilizada)	Sim (biblioteca externa)

Escolha da equipe: O Java foi a linguagem escolhida para o desenvolvimento do *software* da estação base, uma vez que preenche satisfatoriamente os requisitos do projeto. Ressalta-se novamente que a escolha do Java foi feita em conjunto com a escolha da biblioteca do Processing. Notavelmente, no Java há a facilidade em portar, sem adaptações consideráveis na maioria dos casos, programas para diferentes plataformas – processo este que é mais complexo no C++. Com relação ao quesito de desempenho em computadores atuais, a linguagem escolhida é satisfatória, visto que há manutenção constante da implementação das bibliotecas e da máquina virtual do Java pelos desenvolvedores – que buscam, entre outros aspectos, otimizar a linguagem para as tecnologias disponíveis atualmente.

1.3.1.3 Sistema operacional

Estando escolhida a linguagem de programação, o próximo passo é escolher um sistema operacional que seja compatível com ela e que satisfaça os requisitos da seção 1.2.1. Os aspectos desejáveis, portanto, são:

- Deve ser compatível com o Java;
- Deve ser compatível com ferramentas de desenvolvimento *visual* de interface gráfica;
- Deve ser gratuito e de código aberto.

A Tabela 3 apresenta uma comparação entre dois sistemas operacionais, Linux e Windows – sobre os quais a equipe tem considerável experiência e possibilidade de uso nos próprios computadores pessoais.

Tabela 3: Comparação entre sistemas operacionais.

Característica	Linux	Windows
Compatível com Java	Sim	Sim
Compatível com ferramentas visuais de desenvolvimento	Sim	Sim
Gratuito e de código aberto	Sim	Não

Escolha da equipe: Ambos os sistemas comparados são compatíveis com o Java e com ferramentas visuais de desenvolvimento de interface gráfica (NetBeans e Eclipse). Porém, o Linux é o único gratuito e de código aberto, e portanto foi o sistema operacional escolhido para o desenvolvimento do projeto da estação base.

1.3.2 Sistema de comunicação

Como foi explicitado nos requisitos, o sistema de comunicação ter as seguintes características:

- Alcance de, no mínimo, 20 metros sem fios;
- Velocidade suficiente para, simultaneamente, o envio de comandos de movimentação ao robô, recebimento de dados de leituras de sensores e recebimento de imagens da câmera;
- Fluxo de dados bidirecional (*full-duplex*);

- Utilização simples do protocolo TCP.

Para o cálculo da taxa de transmissão mínima necessária, faz-se uma estimativa inicial. Prevê-se que o envio de imagens da câmera é o que mais utilizará os recursos da conexão. Supondo serem usadas a máxima qualidade e taxa de amostragem suportadas por uma câmera VGA comum (resolução 640x480, RGB 24 bits, 30 FPS), além de uso de compressão JPEG com 90% de qualidade, haverá uso de aproximadamente 2,4 MB/s ou 19,2 Mbits/s (RESOLUTION, 2013).

O envio de comandos ao robô, supondo que cada comando tenha 2 KB e hajam 10 comandos por segundo (estimativa exaregada), gastará em torno de 160 Kbits/s. O recebimento de informações do robô e leituras dos sensores, supondo que cada pacote tenha 2 KB (com exagero) e haja recebimento de 10 pacotes por segundo, utilizará 160 Kbits/s na banda da conexão. Portanto, o valor mínimo desejável da taxa de transferência é de $19,2 + 0,16 + 0,16 = 19,52$ Mbits/s.

Na Tabela 4 está presente uma comparação entre diferentes tecnologias de comunicação sem fios, que potencialmente podem satisfazer às necessidades do projeto.

Tabela 4: Comparação entre tecnologias de comunicação sem fios.

Característica	802.11g (Wi-Fi)	Rádio Frequência (RF)	Bluetooth
Distância máxima de alcance	50-100 metros	30-100 metros	10 metros
Velocidade de transmissão máxima	54 Mbits/s	2 Mbits/s	1 Mbits/s
Fluxo de dados <i>full-duplex</i>	Sim	Sim	Sim
Possibilidade e simplicidade de uso de TCP	Sim	Não	Não

Escolha da equipe: O Wi-Fi é o recurso mais atrativo em todos os aspectos que foram comparados, preenchendo satisfatoriamente os requisitos do sistema de comunicação. Sua velocidade e alcance são suficientes para satisfazer as necessidades do projeto, e o fluxo de dados pode ser *full-duplex*. Notavelmente, o Wi-Fi é o único sistema comparado que oferece a possibilidade (com simplicidade) de uso do protocolo TCP – o que é um requisito importante para o desenvolvimento ágil e satisfatório do projeto.

É importante ressaltar que uma conexão Wi-Fi 802.11g dedicada será utilizada para a comunicação entre o robô e a estação base, de modo que a velocidade de conexão possa ser utilizada com maior eficiência sem interferências de outros utilizadores.

1.3.3 Sistema embarcado

Nesta seção serão apresentadas as alternativas pesquisadas para o sistema embarcado, levando-se em conta os requisitos da Seção 1.2.3.

1.3.3.1 Imagens instantâneas do ambiente

Para a obtenção de imagens do ambiente onde o robô se encontra, optou-se por utilizar uma *webcam* USB conectada à placa TS-7260 já presente no robô. Essa placa possuirá o *hardware* de comunicação Wi-Fi do robô, e deve-se lembrar que um único canal sem fios será utilizado (como explicitado nos requisitos). Mostra-se adequada, portanto, a utilização de uma câmera que possa ser conectada por USB à placa, de modo que as imagens possam ser transmitidas por esse canal Wi-Fi. Abaixo estão listadas as características desejáveis da *webcam*:

- Possuir conexão USB 2.0;
- Ser compatível com Linux;
- Ser capaz de, no mínimo, produzir imagens em resolução VGA (640x480), RGB 24 bits a 30 fps, para que a visualização possa ser feita com qualidade satisfatória.

A compatibilidade com o Linux pode ser garantida com a escolha de uma *webcam* em conformidade com o padrão UVC (*USB Video Class*) e compatível com o *driver* Video4Linux 2 (V4L2), presente nos *kernels* do Linux a partir da versão 2.5. Uma lista de dispositivos que seguem esse padrão está disponível em (TOOLS, 2013). Três câmeras de custo baixo e com disponibilidade no Brasil foram selecionadas a partir da lista, como apresentado na Tabela 5.

Tabela 5: Comparação entre *webcams* USB.

Característica	Genius FaceCam 2000	Microsoft VX-500	Genius iSlim 1300
Conexão USB 2.0	Sim	Sim	Sim
Compatível com Linux	Sim	Sim	Sim
Resolução máxima de vídeo	1620X1200	640x480	1280X1024
RGB 24 bits	Sim	Sim	Sim
Taxa de amostragem	30 fps	30 fps	30 fps
Custo	R\$ 82,00	R\$ 79,50	R\$ 34,99

Percebe-se que as características técnicas de todas as três câmeras são satisfatórias para o preenchimento dos requisitos. Porém, há certas diferenças relacionadas a custo e resolução.

Escolha da equipe: A *webcam* escolhida foi a Genius iSlim 1300, principalmente tendo em vista o seu custo muito reduzido em relação às outras. Além disso, possui resolução muito satisfatória. Em Curitiba, há disponibilidade desse modelo em pronta entrega.



Figura 1: *Webcam* Genius iSlim 1300.

Fonte: (GENIUS, 2013)

1.3.3.2 Movimentação do robô

Uma vez que o sistema de movimentação do robô, incluindo motores, acionadores, drivers de potência e rodas já estão instalados no robô e atendem aos requisitos, não houve nova pesquisa sobre esses componentes. Um chassi de 40 cm de largura por 50 cm de comprimento, duas rodas de tração e uma roda guia estão presentes atualmente. As rodas de tração estão dispostas na parte posterior do robô, possuindo 20 cm de diâmetro e 4 cm de largura. O chassi está equipado com 2 motores Bosch FPG 12V, 2 baterias Unybatt 12V-7,2 Ampére-hora e duas pontes H L298 (MARIN et al., 2012). A disposição dos itens no robô pode ser vista na figura 2.

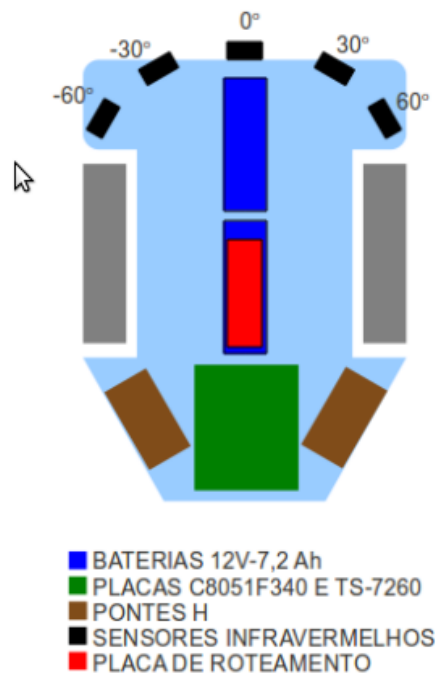


Figura 2: Disposição dos itens no robô

Fonte: (MARIN et al., 2012)

1.3.3.3 Odometria

Para a obtenção da aceleração, velocidade e posição do robô, diversas tecnologias podem ser escolhidas. Abaixo serão descritas as principais opções:

- **Encoder:** Ligado ao eixo da roda do robô, efetua a contagem das rotações realizadas por ela, permitindo assim calcular a distancia percorrida. Se dois encoders forem instalados, um em cada roda, a direção do movimento poderá ser obtida a partir de cálculos baseados na contagem de voltas de cada roda.
- **GPS:** Utiliza sinais de satélites para obter as coordenadas geográficas do robô. A direção e o sentido do movimento podem ser obtidos a partir da comparação das leituras atuais com as anteriores.
- **Acelerômetro:** Pode utilizar a tecnologia chamada MEMS para medir a aceleração do componente. A velocidade e deslocamento lineares podem ser obtidos por integração numérica da aceleração.
- **Giroscópio:** Pode utilizar a tecnologia chamada MEMS para medir a aceleração angular do componente. A velocidade angular e ângulo de rotação podem ser obtidos por

integração numérica da aceleração angular.

- **Bússola:** Utiliza os campos magnéticos da terra para obter a orientação geográfica absoluta do robô.

Na Tabela 6 está presente uma comparação entre as tecnologias apresentadas.

Tabela 6: Comparação entre tecnologias para odometria.

Característica	Encoder	GPS	Acelerômetro	Giroscópio	Bussola
Sujeito a influências externas	Deslizamentos	Não	Não	Não	Ruídos de campos magnéticos diversos
Ambiente de operação	Interno / Externo	Externo	Interno / Externo	Interno / Externo	Interno / Externo
Posicionamento	Relativo	Absoluto	Relativo	Relativo	Absoluto
Acumulo de erro para calculo da posição	Sim	Não	Sim (duas integrações)	Sim (duas integrações)	Sim

Da Tabela 6, vê-se que *encoders* estão sujeitos a erros causados por deslizamentos nas rodas, e que GPS apenas funciona em ambientes externos. A bússola pode ser influenciada por campos magnéticos diferentes do da terra – como por exemplo o gerado pelos motores. O acelerômetro e o giroscópio por sua vez acumulam o erro de duas integrações¹ para obtenção da posição.

Escolha da equipe: Os *encoders* estão sujeito apenas aos erros de deslizamentos e acumulam menos erros na obtenção da posição do que os giroscópios e acelerômetros. Sendo assim eles foram a escolha como principal fonte de dados para odometria. O GPS não foi escolhido pois opera apenas em ambientes internos. A bussola por sua vez poderá sofrer influencias do campo magnético gerado pelos motores do robô. Como a utilização apenas dos *encoders* pode levar a erros no posicionamento devido a deslizamentos, utilizaremos também um acelerômetro e um giroscópio como fonte de dados auxiliar para possibilitar o aumento da exatidão e confiabilidade dos dados obtidos dos *encoders*. Caso discrepâncias consideráveis ocorram entre os dados obtidos pelos sensores, escorregamentos das rodas podem ser detectados e mitigados, atenuando dessa forma erros na determinação do posicionamento.

¹As acelerações linear e angular devem ser integradas duas vezes numericamente para cálculo da posição: A primeira para determinação das velocidades linear e angular; A segunda para determinação do deslocamento e ângulo de rotação atuais.

Por exemplo, em caso de escorregamento roda, um *encoder* fornece uma medição de velocidade maior do que a que corresponde à realidade do movimento do robô. O acelerômetro e o giroscópio, por sua vez, não sofrem influências do escorregamento das rodas, e tenderão a fornecer uma medida mais próxima à realidade. Discrepâncias nas medições podem ser dessa forma detectadas, e procedimentos de atenuação de erros (como por exemplo, descarte de certas medidas dos *encoders*) poderão ser executados no *software* da estação base.

Os *encoders* que serão utilizados (HEDS-9700) já se encontram acoplados ao robô, logo os motivos para a escolha do modelo não serão analisados. Quanto ao acelerômetro e giroscópio, está apresentado a seguir, na Tabela 7, um comparativo entre as opções de menor custo disponíveis no mercado. Na Tabela estão listados apenas os modelos que possuem placas de desenvolvimento, pois acelerômetros e giroscópios geralmente são vendidos em encapsulamento LGA ou BGA (que são de difícil soldagem).

Tabela 7: Comparação entre acelerômetros/giroscópios para odometria.

Modelo	Fabricante	Acel.	Giro.	Faixa	Interface	Preço
STEVAL-MKI009V1	STMicroelectronics	3x	-	$\pm 2g$ ou $\pm 6g$	I2C / SPI	\$23.94
ATAVRSBIN1	Atmel	1x	-	-	I2c	\$26.25
KIT3803 MMA7660FC	Freescale	3x	-	$\pm 1.5g$	I2C	\$35.0
ATAVRSBIN1	Atmel	-	1x		I2C	\$26.25
MPU-6050	IvenSense	3x	3x	$\pm 2g$ ou $\pm 4g$; $\pm 250^\circ/seg$ ou $\pm 500^\circ/seg$	I2C	\$8.78
MKI086V1	STMicroelectronics	-	1x	$\pm 30^\circ/seg$	Analog	\$31.50
STEVAL-MKI094V1	STMicroelectronics	-	3x	$\pm 400^\circ/seg$	Analog	\$31.50
ATAVRSBIN1	Atmel	1x	1x		I2C	\$26.25
DM240316	Zena	3x	3x		RF	\$99.99

MPU-6050

Com base na Tabela 7, o modelo MPU-6050 da IvenSense foi escolhido, principalmente devido ao seu baixo custo: \$8.78. Este modelo possui um acelerômetro e um giroscópio (ambos de 3 eixos), além entradas para uma bússola externa de 3 eixos, tudo integrado a um único chip (EVENSENSE, 2013). A faixa de operação para o acelerômetro é de $\pm 2g$ ou $\pm 4g$ e para o giroscópio é de $\pm 250^\circ/seg$ ou $\pm 500^\circ/seg$. A sensibilidade do acelerômetro

é de 16384 LSB/g ou 8192 LSB/g . A sensibilidade do giroscópio é de $131 \text{ LSB}/(^{\circ}/\text{seg})$ ou $65.5 \text{ LSB}/(^{\circ}/\text{seg})$. A interface de comunicação do módulo suporta o protocolo I2C. O módulo contendo o chip MPU-6050 e alguns componentes necessários para seu funcionamento pode ser visto na figura 3.

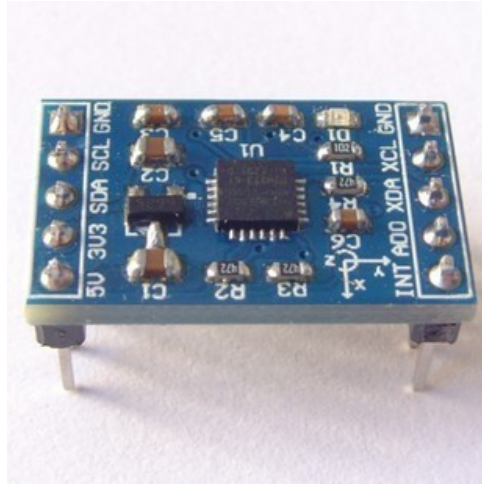


Figura 3: Placa de desenvolvimento contendo o chip MPU-6050

Fonte: (EVENSENSE, 2013)

Encoder Óptico HEDS-9700

Como foi explicitado anteriormente, os *encoders* ópticos já existentes no robô serão utilizados. Ele está equipado com duas unidades do modelo HEDS-9700. Quanto ao funcionamento, este modelo gera em sua saída uma onda quadrada à medida em que as rodas são rotacionadas, sendo 1800 pulsos gerados em uma rotação completa. A forma de onda da saída do *encoder* pode ser vista na Figura 4. Pode-se ver na Figura que o *encoder* possui duas saídas, A e B com defasamento ϕ entre elas. O sentido de rotação pode ser determinado pela informação de qual sinal (A ou B) está mais adiantado em fase (AGILENTTECHNOLOGIES, 2002). A leitura e contagem das rotações do encoder serão feitas conforme foi desenvolvido no projeto anterior (descrito em Marin et al. (2012)).

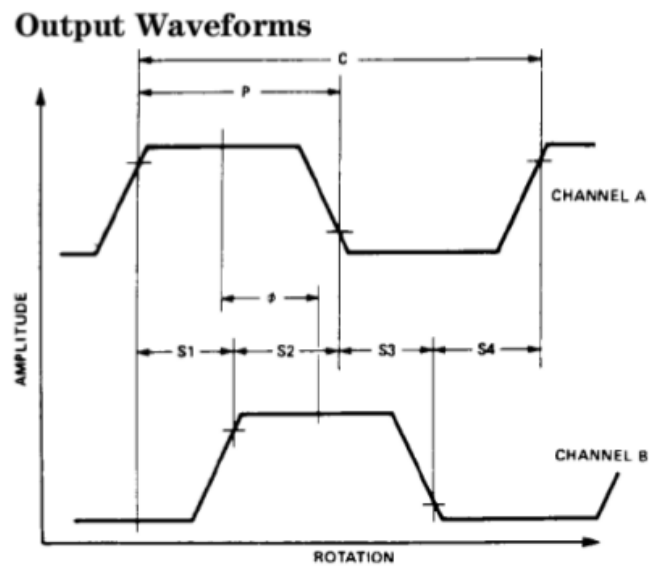


Figura 4: Forma de onda na saída do encoder

Fonte: (AGILENTTECHNOLOGIES, 2002)

1.3.3.4 Detecção de obstáculos

Sensor de proximidade Infra Vermelho IR 2Y0A02F98

A detecção de obstáculos, que é um requisito para o projeto robô, será feita pelos sensores de Infra Vermelho já existentes nele. Estão presentes 5 unidades do modelo IR 2Y0A02F98 (SHARPCORPORATION, 2006). Uma característica interessante deste sensor é que ele sofre pouca influência das cores dos objetos, devido ao método de medição baseado em triangulação. Na figura 5 pode-se verificar o fato. A linha tracejada é a resposta para reflexão em uma superfície cinza e a linha contínua é a resposta para reflexão em uma superfície branca.

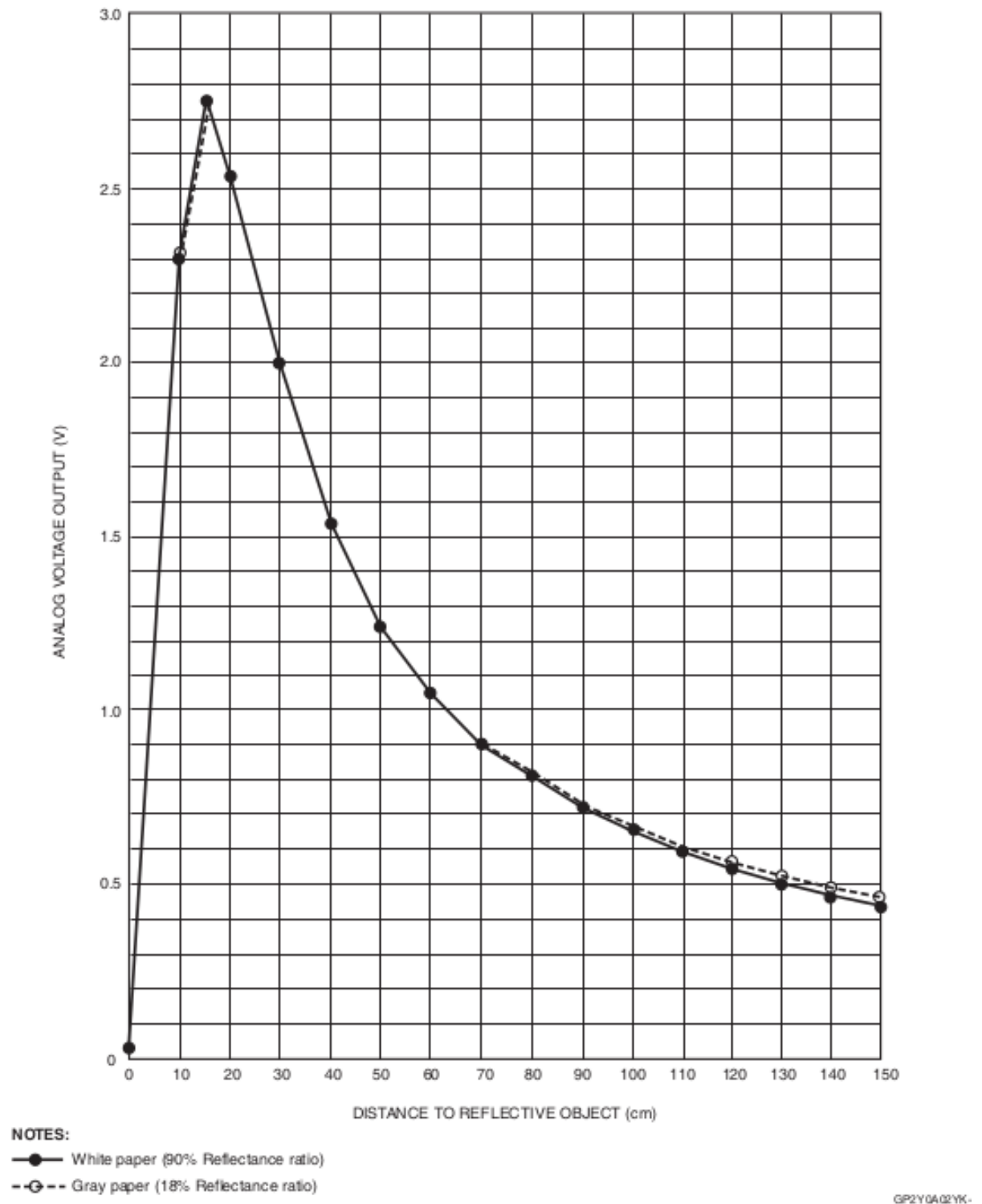


Figura 5: Curva de resposta do sensor Infra Vermelho IR 2Y0A02F98

Fonte: (SHARPCORPORATION, 2006)

1.3.3.5 Microcontrolador

A interface entre sensores e atuadores (sistemas de acionamento dos motores) com a placa TS-7260 ² será feita por um sistema microcontrolado. Este sistema deve possuir as interfaces adequadas para comunicação com todos os componentes. Na Tabela 8 estão listados os requisitos mandatórios para escolha do microcontrolador, e na Tabela 9 estão listados os requisitos desejáveis (porém não obrigatórios).

Tabela 8: Requisitos mandatórios para escolha do microcontrolador.

Requisito	Justificativa
Interface I2C	Comunicação com acelerômetro e giroscópio
Geração de PWM em 4 canais	Acionamento dos motores pelas pontes H
Interface Serial	Comunicação com a placa TS-7260
Interrupções em 2 canais (com capacidade de processamento de no mínimo 2865 interrupções/segundo em cada canal ³)	Leitura do valor dos <i>encoders</i>
Conversor AD em 5 canais	Leitura dos sensores de IR

Tabela 9: Requisitos desejáveis para escolha do microcontrolador.

Requisito desejável	Justificativa
Desenvolvimento em plataforma livre	Diminuição do custo de softwares para desenvolvimento
2 Interfaces seriais ou JTAG	Utilização para <i>debug</i> ou <i>logs</i>
Solução integrada em um único chip	Redução do tamanho da placa e da quantidade de componentes, diminuindo assim o custo e melhorando a organização e disposição dos mesmos

Na Tabela 10 estão listados diversos microcontroladores que foram pesquisadas para o projeto. Todos os modelos atendem aos requisitos da Tabela 8.

Escolha da equipe: Dentre as opções listadas na Tabela 10, o modelo LPC2103 foi escolhido. A escolha foi feita baseando-se principalmente no custo do microcontrolador. Nota-se, a primeira vista ao efetuar-se uma comparação com o SIM3C146, que essa última opção

²Deve-se ressaltar que os atuadores (sistemas de acionamento dos motores) e a placa TS-7260 já existem no robô (MARIN et al., 2012).

³Valor calculado com base no tamanho das rodas, supondo velocidade máxima de deslocamento de 1m/s.

possui desempenho melhor e custo ligeiramente menor. Porém a escolha do LPC2103 justifica-se pela melhor documentação e mais ampla disponibilidade de recursos para o mesmo. A documentação fornecida pelo fabricante desse microcontrolador é mais completa e, pelo fato do modelo estar há mais tempo no mercado, a quantidade de informações e recursos disponíveis na *internet* é maior.

LPC2103

O LPC2103 é um microcontrolador baseado na arquitetura ARM7TDMI-S da NXP (NXP, 2013). Este microcontrolador pode operar em até 70MHz executando a 63MIPS . O Microcontrolador possui 2 interfaces I2C, 2 interfaces seriais, até 14 saídas de PWM, até 13 canais de interrupções externas, 8 canais de conversão para um conversor analógico digital de 10 bits, 32kbytes de memória FLASH para código e 8kbytes de memória RAM. Ele também suporta *debug* via JTag por meio de um *debugger* externo. O custo desse microcontrolador é de \$6.16 (CORPORATION, 2013). O LPC2103 está disponível em encapsulamento LQFP com 48 pinos.

O microcontrolador escolhido pode também ser programado utilizando o protocolo ISP, por meio de ferramentas livres como o *lpc21isp* (LPC21ISP, 2013). Para a geração do código hexadecimal utilizado pelo *lpc21isp*, basta efetuar a compilação de código em C utilizando o GCC (GCC, 2013). Pode-se notar que o LPC2103, além de atender aos requisitos mandatórios da Tabela 8, também atende aos requisitos desejáveis que foram expostos na Tabela 9.

1.3.3.6 Placa de circuito impresso

A placa de circuito impresso será projetada utilizando ferramentas livres, como o gEDA (GEDA, 2013) e PCB (PCB, 2013). Após o projeto da placa os arquivos *Gerber* serão enviados à empresa Stick (IMPRESSOS, 2013) para impressão. A soldagem dos componentes será realizada pela própria equipe.

Deve-se ressaltar que como, em geral, na soldagem de um *chip* com encapsulamento LQFP há complexidade e riscos consideráveis (como por exemplo, queima do *chip* ou da placa), a equipe pesquisou informações sobre *kits* de desenvolvimento que tenham o LPC2103 já soldado na placa. Em Curitiba, existe a empresa eSysTech (ESYTECH, 2013) que fabrica tais *kits*. Através de contatos da UTFPR, confirmou-se que na universidade há algumas unidades (fabricadas por essa empresa) que podem ser emprestadas à equipe para a realização do projeto, caso haja necessidade.

Tabela 10: Comparativo entre microcontroladores.**Fonte:** Dados obtidos de (CORPORATION, 2013)

uC	STM32F103C6T7A	PIC32MX320F128H	LPC2103
Fabricante	STMicroelectronics	Microchip Technology	NXP Semiconductors
Arquitetura	ARM® Cortex™-M3	MIPS32® M4K™	ARM7
Core	32bits	32-Bit	16/32-Bit
Velocidade	72MHz	80MHz	70MHz
MIPS	90	124.8	63
I2C	1	2	2
PWM	12	5	14
UART	2	2	2
Einterrupt	16	5	13
FLASH	32k	128k	32k
RAM	10k	16k	8k
Adc	10x12b	16x10b	8x10b
JTAG	sim	sim	sim
Custo	\$6.27	\$6.26	\$6.16

uC	MCF52210CAE66	AT32UC3C264C	SIM3C146
Fabricante	Freescale Semiconductor	Atmel	Silicon Laboratories Inc
Arquitetura	Coldfire V2	AVR	ARM® Cortex™-M3
Core	32-Bit	32-Bit	32-Bit
Velocidade	66MHz	66MHz	80MHz
MIPS	75.9	98.34	100
I2C	2	3	2
PWM	4	8	8
UART	3	1	2
Einterrupt	7	7	16
FLASH	64k	64k	64k
RAM	16k	16k	16k
Adc	8x12b	11x12b	28x12b
JTAG	sim	sim	SIM3C146-B-GQ
Custo	\$7.1	\$9.14	\$6.1

REFERÊNCIAS

- AGILENTTECHNOLOGIES. **Small Optical Encoder Modules HEDS-9700 Series**. 2002. Disponível em:
<<http://www.digchip.com/datasheets/parts/datasheet/021/QEDS-9871-pdf.php>>. Acesso em: 28 de Fevereiro de 2013.
- CAIRO. 2013. Disponível em: <<http://www.cairographics.org/>>.
- CORPORATION, D. **DigiKey Corporation**. 2013. Disponível em:
<<http://www.digkey.com>>. Acesso em: 18 de Fevereiro de 2013.
- ECLIPSE. 2013. Disponível em: <<http://eclipse.org/>>.
- ESYTECH, S. **eSysTech Embedded Systems**. 2013. Disponível em:
<<http://www.esys tech.com.br>>. Acesso em: 18 de Fevereiro de 2013.
- EVENSENSE. 2013. Disponível em:
<<http://www.invensense.com/mems/gyro/mpu6050.html>>.
- GCC. 2013. Disponível em: <<http://gcc.gnu.org/>>. Acesso em: 18 de Fevereiro de 2013.
- GEDA. 2013. Disponível em: <<http://www.gpleda.org>>.
- GENIUS. 2013. Disponível em:
<<http://geniusnet.com/wSite/ct?xItem=16764&ctNode=161>>.
- IMPRESSOS, S. C. **Stick Circuitos Impressos**. 2013. Disponível em:
<<http://www.stick.ind.br>>. Acesso em: 18 de Fevereiro de 2013.
- JAVA. 2013. Disponível em: <www.java.com>.
- LPC21ISP. 2013. Disponível em: <<http://sourceforge.net/projects/lpc21isp/>>. Acesso em: 18 de Fevereiro de 2013.
- MARIN, A. J.; BORGES, J. C. N.; WERGRZN, Y. A. **Desenvolvimento de robô móvel e análise qualitativa de algoritmos de navegação fuzzy**. Curitiba, 2012.
- NETBEANS. 2013. Disponível em: <<http://netbeans.org/>>.
- NXP. **LPC2103 Overview**. 2013. Disponível em:
<<http://www.nxp.com/products/microcontrollers/arm7/LPC2103FBD48.html>>. Acesso em: 28 de Fevereiro de 2013.
- PCB. 2013. Disponível em: <<http://pcb.gpleda.org>>.
- PROCESSING. 2013. Disponível em: <www.processing.org>.

RESOLUTION, M. C. digital camera. 2013. Disponível em:
<<http://web.forret.com/tools/megapixel.asp?width=640&height=480>>.

SHARPCORPORATION. **GP2Y0A02F98YK Datasheet**. 2006. Disponível em:
<http://www.mindsensors.com/index.php?module=documents&JAS_DocumentManager_op=downloadFile&JAS_File_id=335>. Acesso em: 1 de Agosto de 2011.

TOOLS, L. U. driver . 2013. Disponível em: <<http://www.ideasonboard.org/uvic>>.