

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

PRIMEIRO ENTREGÁVEL

CURITIBA

2013

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

MAPEAMENTO DE AMBIENTES COM O ROBÔ BELLATOR

Primeiro entregável 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	PRIMEIRO ENTREGÁVEL 1 - 13/03/2013	3
2	MODELAGEM UML	4
2.1	REQUISITOS FUNCIONAIS	4
2.2	REQUISITOS NÃO FUNCIONAIS	4
2.2.1	Casos de uso	5
2.3	DESCRIÇÃO DAS CLASSES	6
2.3.1	Pacote visual	6
2.3.2	Pacote controle	6
2.3.3	Pacote comunicação	6
2.3.4	Pacote interface gráfica	7
2.3.5	Pacote interface gráfica	7
2.4	DIAGRAMA DE CLASSES	8
3	DIAGRAMA DE BLOCOS DO HARDWARE	11

1 PRIMEIRO ENTREGÁVEL 1 - 13/03/2013

Conforme estabelecido na relação de deliverables do documento de análise tecnológica, este primeiro entregável consiste nos seguintes itens:

1. Versões iniciais dos diagramas de casos de uso e de classes (estação base).
2. Versão inicial do diagrama em blocos (hardware).
3. Explicação inicial de cada bloco (hardware).

2 MODELAGEM UML

2.1 REQUISITOS FUNCIONAIS

1. A estação base deve ler a posição do robô e atualizar a interface gráfica automaticamente. Representado pelo requisito funcional: **"Estação base mostra na interface gráfica a posição do robô - RF01"**
2. A estação base deve ler as posições dos novos obstáculos detectados e atualizar a interface gráfica automaticamente. Representado pelo requisito funcional: **"Estação base mostra na interface gráfica os novos obstáculos detectados pelo robô - RF02"**
3. A estação base deve mostrar na interface gráfica a imagem captada pela webcam. Representado pelo requisito funcional: **"Estação base atualiza a imagem captada pela webcam - RF03"**
4. O usuário pode controlar os seguintes quatro movimentos do robô: movimentar para frente, para trás, para a esquerda ou para a direita. Representado pelo requisito funcional: **"O usuário pode movimentar o robô - RF04"**
5. A estação base deve ser capaz de estabelecer conexão com o robô, informando o usuário caso a conexão ocorra com sucesso ou não. Representado pelo requisito funcional **"O usuário pode estabelecer a conexão entre o robô e a estação base - RF05"**

2.2 REQUISITOS NÃO FUNCIONAIS

1. O usuário poderá salvar o caminho percorrido pelo robô. Representado pelo requisito não funcional: **"O usuário pode salvar o caminho percorrido - RNF01"**
2. O usuário poderá carregar o caminho percorrido pelo robô. Representado pelo requisito não funcional: **"O usuário pode carregar o caminho percorrido - RNF02"**
3. A imagem transmitida pela câmera do robô deve ser colorida. Representado pelo requisito não funcional: **"O robô deve enviar os dados de vídeo coloridos para a estação base - RNF03"**

4. O robô deve transmitir as imagens de sua câmera em tempo real. Representado pelo requisito não funcional: **"O robô pode transmitir os dados de vídeo captados pela câmera em tempo real - RNF04"**
5. O método de entrada do usuário deve se dar por meio de interface gráfica com o auxílio do mouse. Representado pelo requisito não funcional: **"O usuário pode interagir com o robô por meio do mouse - RNF05"**
6. O método de entrada do usuário deve se dar por meio de interface gráfica com o auxílio do teclado. Representado pelo requisito não funcional: **"O usuário pode interagir com o robô por meio do teclado - RNF06"**

2.2.1 Casos de uso

1. Movimentação do robô pelo usuário. Representado pelo caso de uso: **"Movimentar robô - UC01"**
2. Parada do robô solicitada pelo usuário. Representado pelo caso de uso: **"Parar o robô - UC02"**
3. Solicitação de estabelecimento de conexão com o robô. **"Estabelecer conexão com o robô - UC03"**
4. Alteração na velocidade do robô solicitada pelo usuário. **"Alterar velocidade do robô - UC04"**
5. Alteração da posição do robô na interface gráfica segundo os dados lidos do robô. Representado pelo caso de uso: **"Mostrar posição do robô na interface gráfica - UC05"**
6. Inclusão de novos obstáculos e suas posições lidos do robô. Representado pelo caso de uso: **"Mostrar posição dos novos obstáculos detectados na tela - UC06"**
7. Consulta da documentação do robô solicitada pelo usuário. Representado pelo caso de uso: **"Consultar documentação do robô - UC07"**

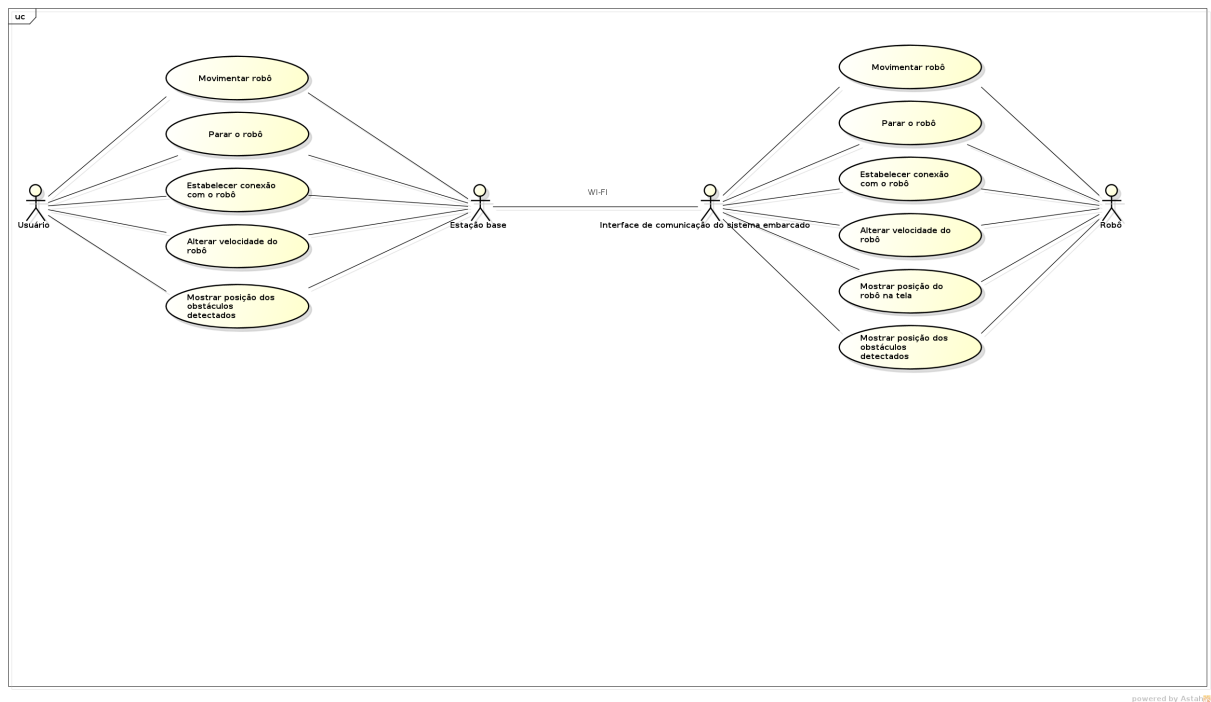


Figura 1: Diagrama de casos de uso

2.3 DESCRIÇÃO DAS CLASSES

2.3.1 Pacote visual

O software da estação base do robô foi dividido em cinco pacotes: visual, controle, comunicação, controle.robô e interface gráfica. Estes serão descritos com suas respectivas classes na Tabela 1.

2.3.2 Pacote controle

Este pacote consiste de toda a parte da estação base que controla as informações essenciais do robô. Conta com as seguintes classes: Mapa, Obstáculos, Robo, ControleSensores, Posinfo, SensorIR e ControleCamera. Na Tabela 2 estão descritas as classes deste pacote.

2.3.3 Pacote comunicação

Este pacote consiste em toda a parte de comunicação da estação base com o robô e conta com as seguintes classes: ClientCommandInterpreter, ClientConnection, ClientReceiver, ClientSender, ServerCommandInterpreter, ServerListener, ServerSender, ServerReceiver e Message. Na Tabela 3 estão descritas as classe deste pacote.

Tabela 1: Pacote visual

Classe	Descrição
Viewer2D	Responsável por exibir os objetos Drawable2D. Possui recursos de pan, zoom e rotate.
Drawable2D	Interface genérica para objetos visuais que podem ser desenhados em um Viewer2D.
EscalaDrawable	Responsável por desenhar uma escala gráfica.
RoboDrawable	Responsável por desenhar o robô na interface gráfica.
RoboTrilhaDrawable	Responsável por desenhar a trilha percorrida pelo robô na interface gráfica.
ObstaculosDrawable	Responsável por desenhar os pontos de cada obstáculo na interface gráfica.
EscalaDrawableProp	Contém as propriedades visuais de desenho da escala.
RoboDrawableProp	Contém as propriedades visuais de desenho do robô
RoboTrilhaDrawableProp	Contém as propriedades visuais de desenho da trilha do robô.
ObstaculosDrawableProp	Contém as propriedades visuais de desenho dos obstáculos.
Ponto	Representa um ponto de coordenadas cartesianas.

2.3.4 Pacote interface gráfica

Este pacote consiste em toda a interface gráfica do sistema e conta com as seguintes classes: JanelaConexao, JanelaPrincipal e JanelaSensores. Na Tabela 4 estão descritas as classes deste pacote.

2.3.5 Pacote interface gráfica

Este pacote consiste de toda a parte do sistema embarcado que gerencia os sensores e atuadores do robô. Conta com as seguintes classes: SensorsManager, WebcamManager e EnginesManager. Na Tabela 5 serão descritas as classes deste pacote.

Tabela 2: Pacote de controle

Classe	Descrição
Mapa	Responsável por representar o mapa. Armazena as informações essenciais do robô e dos obstáculos detectados.
Obstaculos	Responsável por conter os obstáculos detectados pelo robô.
Robo	Responsável por representar o robô, este contém largura, comprimento e centro de movimento (ponto central entre as duas rodas).
ControleSensores	Responsável em atualizar a posição do robô e dos pontos que representam os obstáculos, de acordo com as leituras feitas pelos sensores.
Posinfo	Responsável por conter as informações de uma posição do robô.
SensorIR	Responsável por representar um sensor IR do robô.
ControleCamera	Responsável por controlar as imagens da câmera e o status do recebimento das imagens.

2.4 DIAGRAMA DE CLASSES

A Figura mostra o diagrama de classes tanto da estação base, quanto do software embarcado.

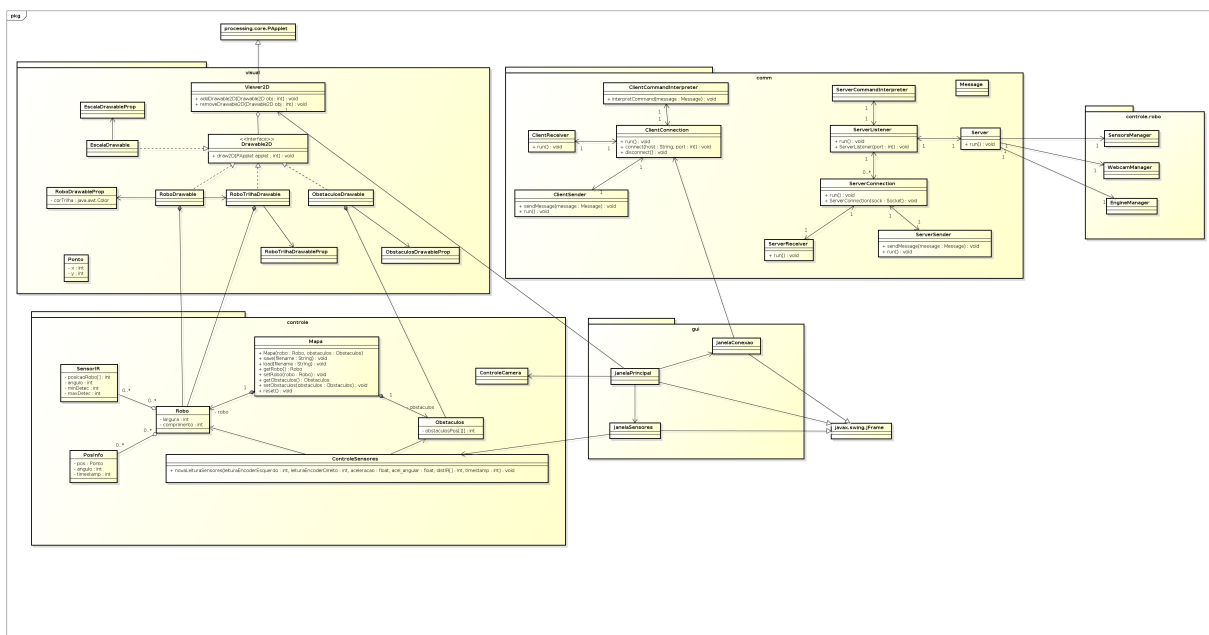
**Figura 2:** Diagrama de classes

Tabela 3: Pacote de comunicação

Classe	Descrição
ClientCommandInterpreter	Responsável pela interpretação dos comandos do cliente. Os comandos recebidos são inseridos em uma fila, de modo a serem posteriormente executados pela thread.
ClientConnection	Responsável por efetuar a gerência da conexão do cliente (estação base) com o servidor (robô).
ClientReceiver	Responsável por receber mensagens de um host de uma conexão.
ClientSender	Responsável por enviar mensagens ao host de uma conexão.
ServerCommandInterpreter	Responsável pela interpretação dos comandos do servidor. Os comandos recebidos são inseridos em uma fila, de modo a serem posteriormente executados pela thread.
ServerListener	Responsável por escutar as novas conexões de clientes.
ServerSender	Responsável por enviar mensagens ao host de uma conexão.
ServerReceiver	Responsável por receber mensagens de um host de uma conexão.
Message	Contém uma mensagem a ser enviada por um Sender.

Tabela 4: Pacote de interface gráfica

Classe	Descrição
JanelaConexao	Responsável por conter a janela com as informações e configurações da conexão com o Bellator.
JanelaPrincipal	Responsável por desenhar a janela principal da interface gráfica do Bellator.
JanelaSensores	Responsável por desenhar a janela responsável pela configuração dos sensores.

Tabela 5: Pacote de controle de robô

Classe	Descrição
SensorsManager	Responsável por gerenciar a coleta de informações dos sensores do robô.
WebcamManager	Responsável por gerenciar a captura de imagens da webcam.
EnginesManager	Responsável por gerenciar a ação dos motores do robô.

3 DIAGRAMA DE BLOCOS DO HARDWARE

Na figura 3 mostra-se o diagrama de blocos do sistema embarcado e suas conexões com o restante do robô. A seguir está também uma descrição para cada um dos blocos da placa de circuito impresso do sistema embarcado.

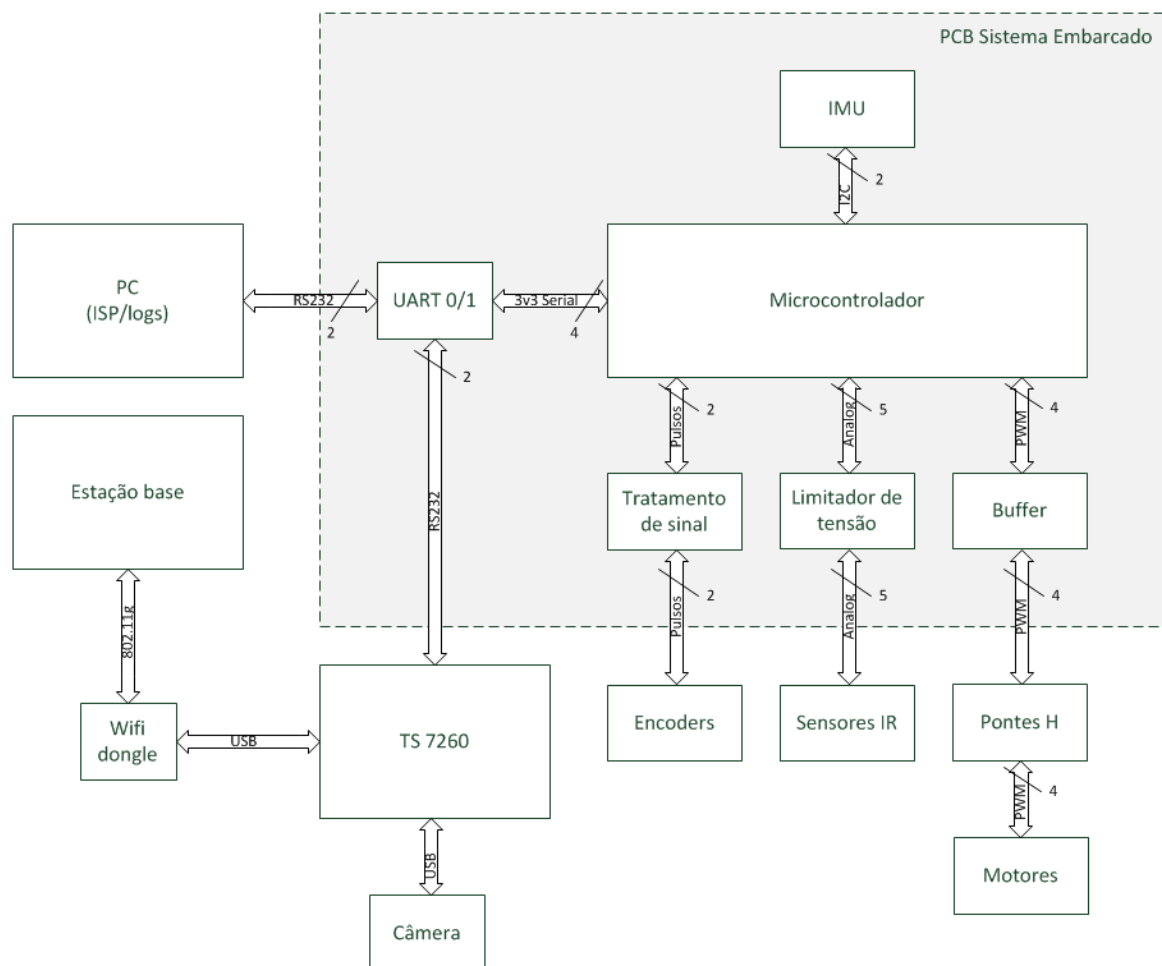


Figura 3: Diagrama de blocos do hardware

1. **Microcontrolador:** Este bloco fará a leitura dos sensores: encoders, infra-vermelhos, acelerômetro e giroscópios. Além disso possui a implementação do protocolo de comunicação para interação com o linux embarcado da placa TS-7260.

2. UART 0/1: Responsável por ajustar os níveis de tensão para comunicação serial no padrão RS-232 com a placa TS-7260.
3. Buffer: Responsável por fornecer corrente e elevar os níveis de tensão de saída do microcontrolador de 3,3V para 5,0V. Esse buffer é conectado às pontes H já existentes no robô.
4. IMU: possui o acelerômetro e o giroscópio e se comunicará com o microcontrolador por meio do protocolo I2C.
5. Limitador de tensão: Necessário pois os sinais de saída dos sensores de infravermelho que já existem no robô não estão limitados em 5V, podendo a saída ultrapassar 5,0V e danificar o microcontrolador.
6. Tratamento de sinal: Composto com um filtro RC passa baixas e um schmitt trigger para remover qualquer falha que possa ocorrer na geração dos pulsos no encoder. A frequência de corte do filtro pode ser obtida pela velocidade máxima que o robô pode atingir e que foi suposta em 1m/s nos requisitos de hardware mencionados anteriormente.