

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

***Sistema de Monitoramento de Nível Baseado em
Processamento Digital de Imagens***

Autor:

Francinei Gomes de Moraes

Orientador:

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Examinador:

Prof. Antônio Cláudio Gómez de Sousa, M.Sc.

Examinador:

Prof. Heraldo Luis Silveira de Almeida, D.Sc.

DEL

Março de 2009

DEDICATÓRIA

Dedico este trabalho ao povo brasileiro que contribuiu de forma significativa à minha formação e estada nesta Universidade. Este projeto é uma pequena forma de retribuir o investimento e confiança que me foram depositados.

AGRADECIMENTO

Aos meus pais, grandes mestres que sempre incentivaram e apostaram no estudo e conhecimento como um meio e ferramenta de transformação.

Agradeço também ao Engenheiro Eletrônico Henrique Oscar Duran Lira pela enorme contribuição por seus conhecimentos de Processamento Digital de Imagens, pois sem estes meu trabalho levaria muito mais tempo para ser concluído.

RESUMO

O presente projeto trata do desenvolvimento de um sistema que visa monitorar o processo de destilação fracionada do petróleo fazendo uso de técnicas de processamento digital de imagens. Para um planejamento e desenvolvimento o mais profissional possível, fiz o uso dos conhecimentos adquiridos em Engenharia de Software: nesta cadeira aprendi como devem ser todas as etapas do desenvolvimento de um projeto bem como as documentações e normas a serem seguidas.

Mesmo já havendo outras tecnologias para tal monitoramento, não encontrei nenhuma que fosse baseada em processamento de imagens, o projeto também apresenta um baixo custo se comparado com os já existentes, é inovador e demonstra a capacidade de desenvolvimento e inovação do profissional de Engenharia Eletrônica formado pela Universidade Federal do Rio de Janeiro.

Ao longo do projeto fiz a montagem de alguns protótipos de hardware, bem como a montagem do ambiente em diferentes configurações a fim de descobrir as dependências do sistema (iluminação e distância entre objetos) e como contorná-las.

Como resultados descobri algumas configurações de câmera, iluminação, distância entre objetos, nas quais o sistema apresentou um acerto satisfatório determinando, assim, o sucesso do projeto.

Palavras-Chave: processamento de imagens, monitoração de processos, controle de processos, sistemas de tempo real.

ABSTRACT

This project is about the development of a system to monitor oil distill using image processing techniques. For a professional planning and development, I have used knowledge acquired in Software Engineering: in this course I have learned the steps for project development, the needed documentation and rules to be followed.

Even though there are others techniques to oil monitoring, I have not found out one that was based on image processing, not to mention low-cost: the system developed in this project is cheap compared to others. It is novel and it shows the development and innovation capabilities of Electronics Engineers graduating from the Federal University of Rio de Janeiro.

During the project I built some hardware prototypes, and I organized the environment in different configurations in order to find all systems dependences and to know how to avoid them.

As a result of this work, I found out some configurations for the camera, illuminations and distances between objects, in which the system presented the best results, determining the project success.

Keywords: image processing, process monitoring, process control, real-time systems.

SIGLAS

UFRJ – Universidade Federal do Rio de Janeiro.

CAT – Circuito Adaptador de Tensão.

CA – Circuito Atuador.

CLP – Circuito Lógico Programável.

LPT – *Line Print* Terminal (Porta paralela).

MTH – Módulo de Teste de Hardware.

CT – Circuito de Teste.

Sumário

1. O PROJETO DE MONITORAMENTO.....	ERRO! INDICADOR NÃO DEFINIDO.
1.1. TEMA	1
1.2. DELIMITAÇÃO.....	1
1.3. JUSTIFICATIVA	1
1.4. OBJETIVOS.....	2
1.5. METODOLOGIA	3
1.6. DESCRIÇÃO.....	3
1.7. COMPONENTES DO SISTEMA.....	4
1.8. FUNCIONAMENTO GERAL DO SISTEMA.....	5
2. O SISTEMA DE MONITORAMENTO.....	8
2.1. DIAGRAMA DE CASOS DE USO DO SISTEMA DE MONITORAMENTO.....	8
2.2. DIAGRAMA FÍSICO DO MÓDULO CONTROLADOR.....	9
2.2.1. O Módulo Controlador	10
2.2.2. Caso de uso Módulo Controlador.....	11
2.2.3. Diagrama de Etapas do Módulo Controlador.....	13
2.2.4. Máquina de Estados Módulo Controlador.....	14
2.2.5. Diagrama de Estados do Módulo Controlador	14
2.2.5.1. Legenda da Figura 2.4:.....	15
2.2.5.2. Definição dos Estados da Figura 2.4:.....	15
2.2.6. Classe do Módulo Controlador	16
2.2.7. Diagrama de Atividades do Módulo Controlador	17
2.3. MÓDULO DE LEITURA E IDENTIFICAÇÃO DO CÓDIGO DE BARRAS	18
2.3.1. Caso de uso Módulo Código de Barras	18
2.3.2. Classe do Módulo Código de Barras.....	19
2.3.3. Diagrama de Atividades do Módulo Código de Barras	21
2.3.4. Garrafa Coletora.....	21
2.3.5. Padrão de Codificação	22
2.3.6. Testes de Desempenho	23
2.4. MÓDULO DE MEDIÇÃO DO MENISCO.....	26
2.4.1. Caso de Uso do Módulo Menisco	27
2.4.2. Diagrama Físico do Módulo Menisco.....	27
2.4.3. Classe do Módulo Menisco	30
2.4.4. Diagrama de Atividades do Módulo Menisco.....	32
2.4.5. Infra-Estrutura do Sistema Menisco.....	33
2.4.5.1. Calibragem do Sistema Ótico-Menisco.....	34
2.4.5.2. Calibragem do Sistema Físico-Menisco.....	35

2.4.5.3. Resultados dos Testes de Desempenho.....	36
2.5. MÓDULOS AUXILIARES	39
2.5.1. Programa de avaliação de desempenho do algoritmo de detecção do menisco - nível de líquido no recipiente (Figura 2.20).....	39
2.5.2. Avaliação dos Circuitos de Hardware	40
2.6. MÓDULOS DE HARDWARE	42
2.6.1. Circuito Adaptador de Tensão – CAT.....	42
2.7. A INTERFACE DO SISTEMA PRINCIPAL E SEU FUNCIONAMENTO	43
3. O PROCESSAMENTO DE IMAGEM	48
3.1. FUNDAMENTOS PARA PROCESSAMENTO DIGITAL DE IMAGENS.....	48
3.1.1. Imagem em Escala de Cinza.....	48
3.1.2. Definição Matemática de Imagem	49
3.1.3. Imagem Digital.....	49
3.1.4. Representação de Imagens.....	49
3.1.5. Propriedades de um Pixel	51
3.2. ETAPAS PARA O PROCESSAMENTO DIGITAL DE IMAGENS	53
3.2.1. Aquisição de Imagens	53
3.2.2. Realce de Imagem.....	54
3.2.3. Segmentação de Imagens	60
3.2.4. Classificação de Imagens.....	61
3.3. PROCESSAMENTO DIGITAL DE IMAGEM E O PROJETO	62
3.3.1. Detecção do Menisco.....	63
3.3.2. Etapas do Projeto	64
3.3.2.1. Aquisição da Imagem Real	64
3.3.2.2. Imagem em Níveis de Cinza	65
3.3.2.3. Imagem Binarizada (Limiarização ou Thresholding).....	66
4. ELABORAÇÃO DOS CIRCUITOS	67
4.1. CIRCUITO TEÓRICO	67
4.2. CIRCUITO PRÁTICO	68
5. CONCLUSÃO	72
APÊNDICE A.....	73
A.1. DIAGRAMAS DOS CIRCUITOS.....	73
A.2. DIAGRAMA DO CIRCUITO DE TESTE DA LPT.....	75

Lista de Figuras

Figura 1.1: Visão Geral do Sistema de Monitoramento.....	5
Figura 2.1: Modulo Controlador	7
Figura 2.2: Diagrama de Casos de Uso do Sistema de Monitoramento	8
Figura 2.3: Diagrama Físico do Módulo Controlador	9
Figura 2.4: Diagrama de Etapas do Módulo Controlador	13
Figura 2.5: Diagrama de Estados do Controlador	14
Figura 2.6: Diagrama de Atividades do Módulo Controlador.....	17
Figura 2.7: Diagrama de Atividades do Módulo Código de Barras	21
Figura 2.8: Garrafa Coletora	22
Figura 2.9: Modelo de Código de Barras do tipo intercalado 2 de 5.....	23
Figura 2.10: Código de Barras intercalado 2 de 5 invertido	23
Figura 2.11: Tomada de fotos com o eixo focal alinhado com o centro da garrafa	24
Figura 2.12: Tomada de fotos com o eixo focal alinhado com o fundo	24
Figura 2.13: Tomada de fotos com a câmera inclinada enquadrando toda a garrafa	25
Figura 2.14: Programa de Testes para identificação do Código de Barras	25
Figura 2.15: Diagrama Físico do Módulo Menisco.....	29
Figura 2.16: Diagrama de Atividades do Módulo Menisco	32
Figura 2.17: Distorção apresentada com o uso da câmera de maior abertura	33
Figura 2.18: Calibragem do Sistema Menisco	34
Figura 2.19: Um exemplo da Amostragem de Teste	35
Figura 2.20: Início da Marcação.....	39
Figura 2.21: Programa experimental para simulação dos sinais enviados aos hardwares	41
Figura 2.22: Simulador do Sistema Principal.....	41

Figura 2.23: Diagrama de Blocos do Circuito CAT	42
Figura 2.24: Diagrama Esquemático de Entrada/Saída	42
Figura 2.25: Controlador UI.....	44
Figura 2.26: Aguardando a Gota	44
Figura 2.27: Gota Detectada	45
Figura 2.28: Aguardando Leitura do Código	45
Figura 2.29: Código de Barras Lido	46
Figura 2.30: Início Acompanhar Nível	46
Figura 2.31: Nível Variando.....	47
Figura 2.32: Esperando Nova Garrafa	47
Figura 3.1: Conjunto E.....	48
Figura 3.2: Produto Cartesiano $E \times K$	48
Figura 3.3: Imagem em Níveis de Cinza.....	49
Figura 3.4: Representação Vetorial [5].....	50
Figura 3.5: Representação Matricial [5]	50
Figura 3.6: Vizinhaça 4 de um Pixel.....	51
Figura 3.7: Vizinhaça 8 de um Pixel.....	51
Figura 3.8: Estrutura do Arquivo PNG [6].....	52
Figura 3.9: Estrutura do Arquivo bitmap [6].....	53
Figura 3.10: Estrutura do Arquivo TIFF [6]	53
Figura 3.11: Câmera Digital.....	54
Figura 3.12: Histograma de Imagens [8].....	55
Figura 3.13: Função Linear	56
Figura 3.14: Função Raiz Quadrada	56
Figura 3.15: Função Logarítmica	57

Figura 3.16: Função Inversa.....	57
Figura 3.17: Mascara de Roberts.....	58
Figura 3.18: Mascara de Sobel.....	59
Figura 3.19: Mascara da Média.....	59
Figura 3.20: Bico Real.....	64
Figura 3.21: Garrafa Real.....	64
Figura 3.22: Bico em Tons de Cinza.....	65
Figura 3.23: Garrafa em Tons de Cinza.....	65
Figura 3.24: Bico Binarizado.....	66
Figura 3.25: Garrafa Binarizada.....	66
Figura 4.1: Circuito de Linhas Pares.....	67
Figura 4.2: Circuito de Linhas Ímpares.....	67
Figura 4.3: Circuito de Linhas Pares.....	68
Figura 4.4: Circuito de Linhas Ímpares.....	69
Figura 4.5: Painel de Leds.....	70
Figura 4.6: Circuito Adaptador de Tensão Vista Frontal.....	71
Figura 4.7: Circuito Adaptador de Tensão Vista Traseira.....	71
Figura A.1: Diagrama Esquemático do CAT.....	73
Figura A.2: CAT Simulado no ORCAD.....	74
Figura A.3: Circuito de Teste.....	75

Lista de Tabelas

Tabela 1: Câmera posicionada no fundo da garrafa	36
Tabela 2: Câmera posicionada no meio da garrafa.....	37
Tabela 3: Câmera inclinada posicionada garrafa.....	38
Tabela 4: Descrição da Pinagem da LPT com o CAT	43

Capítulo 1

Introdução

1.1. Tema

O trabalho trata do desenvolvimento de um sistema visando a monitoração automatizada, via processamento digital de imagens, do processo de destilação fracionada do petróleo. No desenvolvimento deste trabalho serão usados conhecimentos de Eletrônica Analógica, Eletrônica Digital, Engenharia de Software, Processamento Digital de Imagens e Linguagens de Programação.

1.2. Delimitação

O projeto que foi desenvolvido envolve o monitoramento da destilação fracionada do petróleo, usando como técnica o processamento digital de imagem. A demanda por este produto é de empresas e indústrias do segmento petrolífero. Outra característica importante é o bom índice de acerto apresentado pelo sistema baseado em processamento digital de imagem.

1.3. Justificativa

Industrialmente, o petróleo é destilado em grandes quantidades, produzindo frações conhecidas como diesel, gasolina, nafta, etc. Cada uma destas frações tem uma faixa de temperatura característica e propriedades físicas e químicas padronizadas. No entanto, um dado petróleo, dependendo de sua origem geoquímica e do seu processo de formação, não necessariamente produz todas as frações ou não as produz nos valores padronizados das propriedades de cada fração. Sendo assim, uma mistura de petróleo é composta, acrescida de outras substâncias produzidas na refinaria, de forma que as diferentes frações sejam produzidas de acordo com as especificações e em volume adequado à demanda do mercado (ASTM, 2002).

Para determinar em laboratório os diferentes cortes ou frações de subprodutos existentes em uma determinada amostra, bem como o volume que cada uma das faixas produz, utiliza-se a destilação em batelada, sendo um dos métodos o padrão ASTM D5236 (ref. ASTM, 2002). Esta determinação é vital não só para a destilação futura do

mesmo óleo executada em regime de produção em uma refinaria, mas também na formação de seu preço, pois o valor do petróleo é função do volume produzido de suas frações mais nobres.

O processo de destilação em batelada já possui uma monitoração automatizada, todavia os instrumentos utilizados apresentam certas dificuldades, tais como: pouca robustez; pouca intercambialidade e dificuldade de operação. Surge daí a proposta da monitoração através de técnicas de processamento de imagens, pois estas apresentam vantagens (Seção 1.4) em relação às técnicas correntes.

1.4. Objetivos

No processo de destilação em batelada ao qual este projeto se aplica, as diferentes frações do petróleo são separadas em frascos localizados em um carrossel na parte inferior da unidade de destilação utilizada. As frações são coletadas em intervalos determinados a partir do instante em que se desprende a primeira gota, indicando que o processo de destilação teve início. A cada intervalo, que corresponde a encher o frasco coletor parcialmente (quando um subproduto termina) ou totalmente, o carrossel de troca de frasco sofre um movimento de rotação, trocando o frasco recipiente do petróleo destilado.

Além disto, é preciso determinar a temperatura do início da destilação, ou seja, a temperatura mínima da primeira faixa ou fração. Como a retirada é contínua, e pela própria dinâmica do processo, o início da destilação corresponde ao instante da formação da primeira gota no condensador lateral (ASTM, 2002).

Com a instrumentação atual, a identificação da formação da primeira gota apresenta algumas dificuldades em termos de instrumentação. Os dispositivos existentes se baseiam na interrupção de um feixe de luz, e apresentam três grandes inconvenientes:

- a) são específicos para cada unidade (eletrônica dedicada).
- b) a regulagem do feixe ao caminho da gota é difícil e instável, pois a queda da gota é sujeita a perturbação.
- c) os instrumentos não apresentam recursos que facilitem a regulagem.

Note-se que a perda da queda da primeira gota equivale à perda de todo o processo de destilação até aquele instante e que um erro na determinação da gota pode comprometer todo o trabalho de medida.

O objetivo principal ao qual este projeto se propõe é: criação de um sistema automático empregando técnicas de processamento de imagens para realizar a detecção

da queda da primeira gota, detecção do código de barras que virá na parte inferior da garrafa, e medição do menisco (nível de líquido na garrafa coletora).

O referido projeto apresenta as seguintes vantagens frente às técnicas atuais:

- a) uso de instrumentação padrão de mercado (câmera e computador), sendo as diferenças entre os equipamentos de destilação, tratadas por programa.
- b) a regulagem é muito mais fácil, não sendo necessário o ajuste do equipamento ao caminho da gota, podendo o operador utilizar a própria imagem gerada pelo dispositivo para enquadrar/regular o posicionamento da câmera.

1.5. Metodologia

Para o desenvolvimento do projeto busquei como metodologia inicial o uso de técnicas aprendidas na cadeira de Engenharia de Software. Sendo assim montei as seguintes documentações:

- a) Plano de Gerenciamento de Projeto de Software;
- b) Especificação de Requisitos de Software.

No primeiro documento foi feito o levantamento de itens tais como finalidade, escopo, objetivo, cronograma, orçamento, papéis e responsabilidades, plano para aquisição e alocação de recursos, plano de relatórios, plano para treinamento etc.

No segundo foi feito o levantamento de itens tais como finalidade, escopo, perspectivas do produto, funções do produto, restrições, pressupostos e dependências, interfaces, requisitos funcionais e não funcionais.

Também lancei mão da reutilização de componentes, artefatos de software, hardware e documentações produzidas em projetos anteriores, pois estes já haviam sido testados, usados, validados e apresentavam as características necessárias ao nosso projeto.

Com o auxílio da documentação produzida e da reutilização de componentes e artefatos, foi possível a economia de tempo, dinheiro e a minimização dos erros encontrados ao longo do processo de desenvolvimento do sistema.

1.6. Descrição

No Capítulo 2 será apresentada uma explicação e decomposição detalhada do projeto. Sua forma de apresentação é uma composição dos documentos: Plano de

Gerenciamento de Projeto de Software, Especificação de Requisitos de Software e outras informações que julgamos necessárias.

No Capítulo 3 serão apresentados os fundamentos do processamento digital de imagens bem como as técnicas utilizadas no projeto e as imagens resultantes.

No Capítulo 4 serão apresentados os esquemáticos de alguns circuitos do projeto bem como fotos do sistema montado em laboratório.

No Apêndice A serão apresentados o diagrama do Circuito Adaptador de Tensão, Circuito de Teste.

1.7. Componentes do sistema

O projeto desenvolvido fará parte de um sistema maior, ou seja, será um subsistema, e como tal poderá ser acionado de maneira remota. O operador do sistema Supervisório é quem fará tal acionamento; segue abaixo uma decomposição do sistema de monitoramento desenvolvido.

- **Unidade de destilação:** Unidade dedicada à destilação do petróleo;
- **Supervisório:** Software do sistema de supervisão do processo de destilação através do CLP;
- **CLP:** Controlador Lógico Programável. Tem a função de realizar a interface entre os sensores da unidade de destilação e o sistema Supervisório;
- **CA:** Circuito Atuador. Este circuito tem a função de acionar a iluminação e as câmeras de monitoramento do menisco, código de barras e gota;
- **CAT:** Circuito Adaptador de Tensão. Este circuito tem a função de casar o nível lógico da porta paralela (LPT), nível TTL, com o nível lógico do CLP;
- **Sistema de Monitoramento por Vídeo:** Software projetado para analisar e identificar os requisitos da destilação por monitoramento em vídeo.

A Figura 1.1 mostra o esquema da integração das partes que compõem o sistema de monitoramento: a unidade de destilação em batelada, *o subsistema de controle* composto por câmeras, CA, CAT e CLP (Controlador Lógico Programável), o sistema Supervisório.

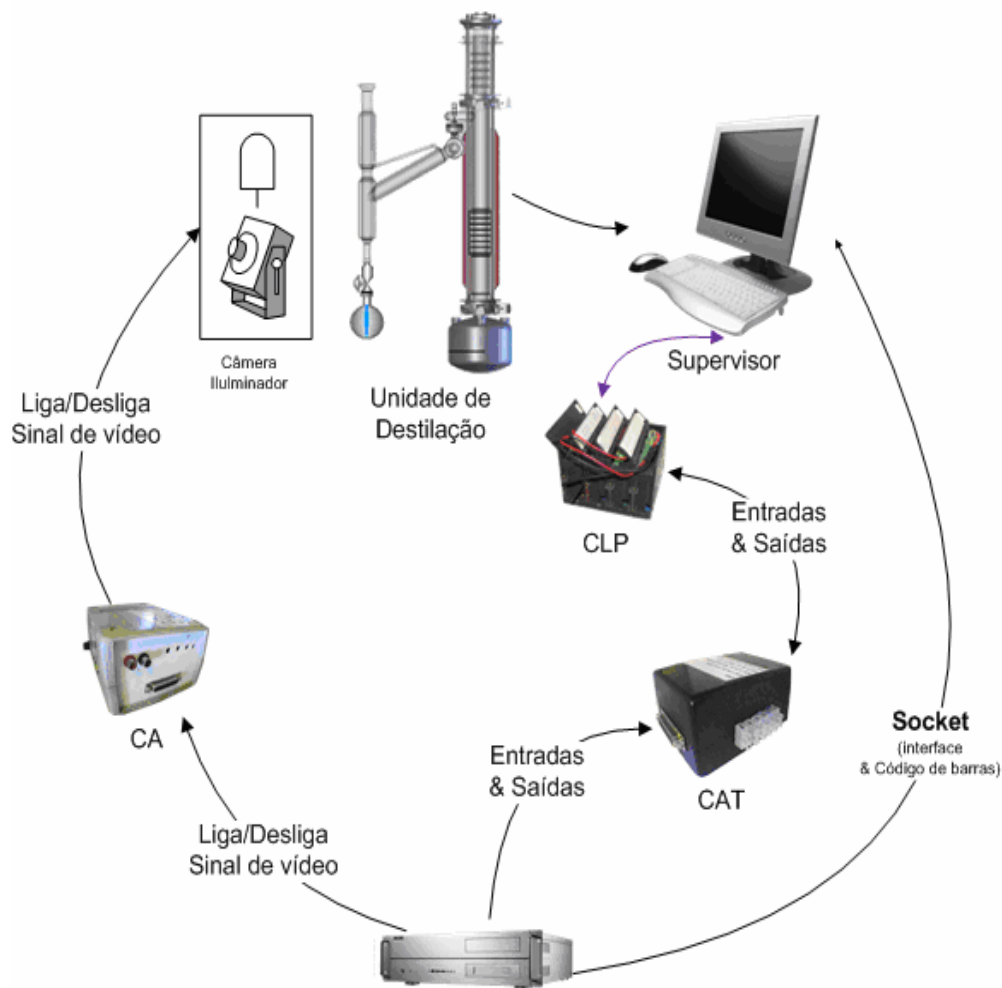


Figura 1.1: Visão Geral do Sistema de Monitoramento

1.8. Funcionamento Geral do Sistema

O sistema de monitoramento compõe-se de diversos módulos (Figura 1.1) que serão descritos brevemente nesta seção.

O sistema Supervisório é o encarregado de processar todos os dados coletados pelos sensores do processo de destilação. Com estes dados o sistema Supervisório controla a destilação, alterando variáveis como, por exemplo, temperatura e vazão segundo a necessidade. Este sistema é executado em um computador exclusivo para não ser sobrecarregado e dedicar-se-á somente ao controle do processo de destilação.

Os sensores não interagem diretamente com o Supervisório, mas sim com um *Controlador Lógico Programável (CLP)* através de uma rede do tipo *FieldBus*. O CLP faz o papel de servidor central na rede *FieldBus*, controlando o tráfego das informações, repassando-as depois para o sistema Supervisório. Os sensores da destilação podem ser vistos como os clientes desta rede. Devido ao sistema de monitoração ser considerado como mais um sensor da unidade, este também se conectará com o CLP. Esta comunicação será efetuada através da porta paralela do computador onde estará sendo executado o *Sistema de Monitoração por Imagem* (computador diferente ao do *Sistema Supervisor*). Em função de a porta paralela reconhecer e enviar sinais elétricos de nível TTL (5 volts) e o CLP trabalhar com sinais de 24 volts, foi projetado um circuito *adaptador de tensão (CAT)*. Este adaptador eleva ou diminui o sinal elétrico para a correta comunicação entre o CLP e o computador do *Sistema de Monitoração por Imagem*.

Quando o Módulo Controlador é executado, a primeira tarefa que o sistema realiza é solicitar ao operador do sistema que realize as calibrações dos módulos de processamento de imagens. *A calibragem consiste basicamente em delimitar as áreas de busca para os algoritmos de processamento de imagens.*

Em seguida é realizada a verificação de todos os módulos de hardware que compõem o sistema. Se o teste de hardware for realizado com sucesso, o sistema é habilitado para iniciar o processo de monitoramento da destilação, senão o operador do sistema será notificado da(s) falha(s) em nível de hardware, e o processo de destilação não poderá ser iniciado até que a falha seja solucionada.

Passando pela fase de teste de hardware, o sistema fica aguardando o sinal vindo do CLP (o Supervisório é que manda o sinal de início para o CLP), que indica o começo da monitoração do processo de destilação. Quando o Supervisório indica ao sistema o começo da monitoração, a câmera e iluminação posicionadas no bico do destilador são acionadas simultaneamente. Neste ponto, o Módulo Controlador cria a *thread* do Módulo da Gota, que é responsável pela indicação da queda da primeira gota, e começam a ser processados os quadros de imagens vindos do bico do destilador. O algoritmo de detecção de movimento, quando detecta a primeira gota, informa ao Módulo Controlador da queda gota dando início a uma série de ações por parte do Módulo Controlador: a câmera e a iluminação do bico são desligadas, um pulso é enviado ao Supervisório, seguindo o caminho LPT, CAT e por último CLP, a fim de indicar a este a queda da primeira gota; por último a *thread* do Módulo gota é destruída.

Detectada a queda da primeira gota, inicia-se o ciclo de monitoração do menisco. Neste ponto, o Módulo Controlador fica aguardando o recebimento do sinal de nova garrafa vindo do Supervisorio via CLP. Este sinal corresponderá à leitura do código de barras, seguido da leitura do menisco. Quando o CLP envia o sinal de nova garrafa, o Módulo Controlador liga a câmera do menisco/código de barras, liga a iluminação do código de barras e cria a *thread* do Módulo Código de Barras, que será o módulo de processamento de imagens responsável pela detecção do código de barras.

Sendo detectado o código de barras pelo Módulo Código de Barras, o mesmo notificará o fato ao Módulo Controlador, que irá suspender a *thread* do Módulo do Código de Barras, desligará a iluminação e a câmera do menisco/código de barras, ligará a câmera e a iluminação do menisco, criará a *thread* do Módulo Menisco, e a partir deste momento será monitorado o menisco. A cada incremento de volume detectado por esse módulo, uma notificação é feita ao Supervisorio pelo Módulo Controlador (sinal pulso de menisco), que calcula assim o volume presente na garrafa. Quando o subproduto acaba ou a garrafa fica cheia, um sinal de nova garrafa é enviado pelo CLP, dando início ao processo descrito anteriormente (a partir da leitura do código de barras).

O processo descrito nos parágrafos anteriores continua até que seja sinalizado pelo Supervisorio, via CLP, o fim processo de monitoração. Neste ponto as câmeras e iluminações são desligadas, e as *threads* são todas destruídas.

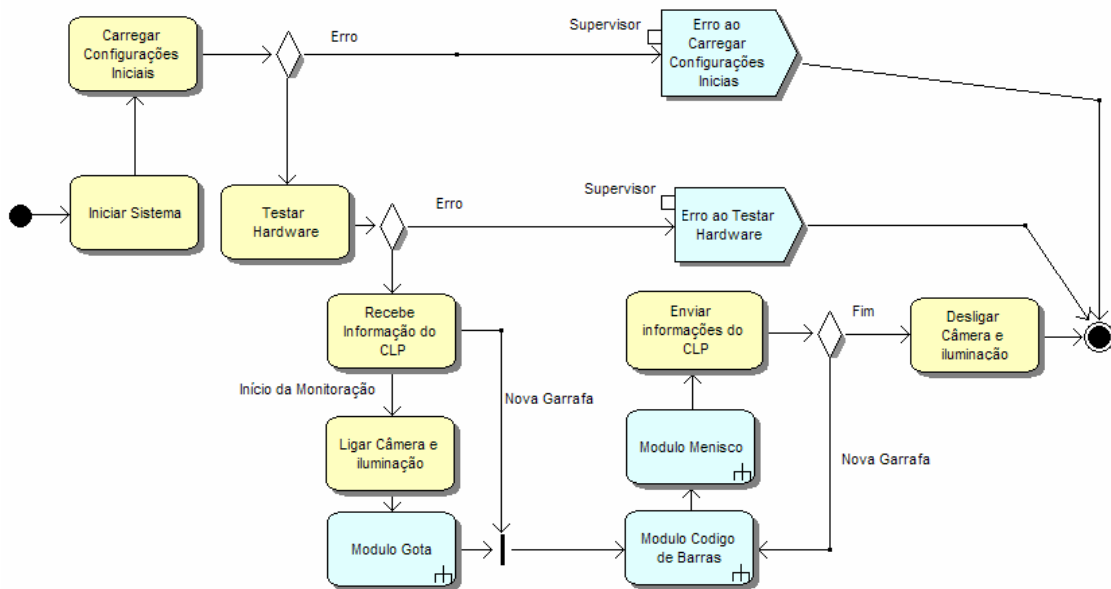


Figura 2.1: Módulo Controlador

Capítulo 2

O Sistema de Monitoramento

O software do sistema de monitoramento divide-se em quatro módulos independentes: Módulo de Detecção da 1a. Gota, Leitura e Identificação do Código de Barras, Monitoramento do Menisco e Módulo Controlador (módulo central do processamento) além de outros sub-módulos para teste de hardware, para calibragem do sistema, gerenciamento de rede, disco e outros recursos.

2.1. Diagrama de Casos de Uso do Sistema de Monitoramento

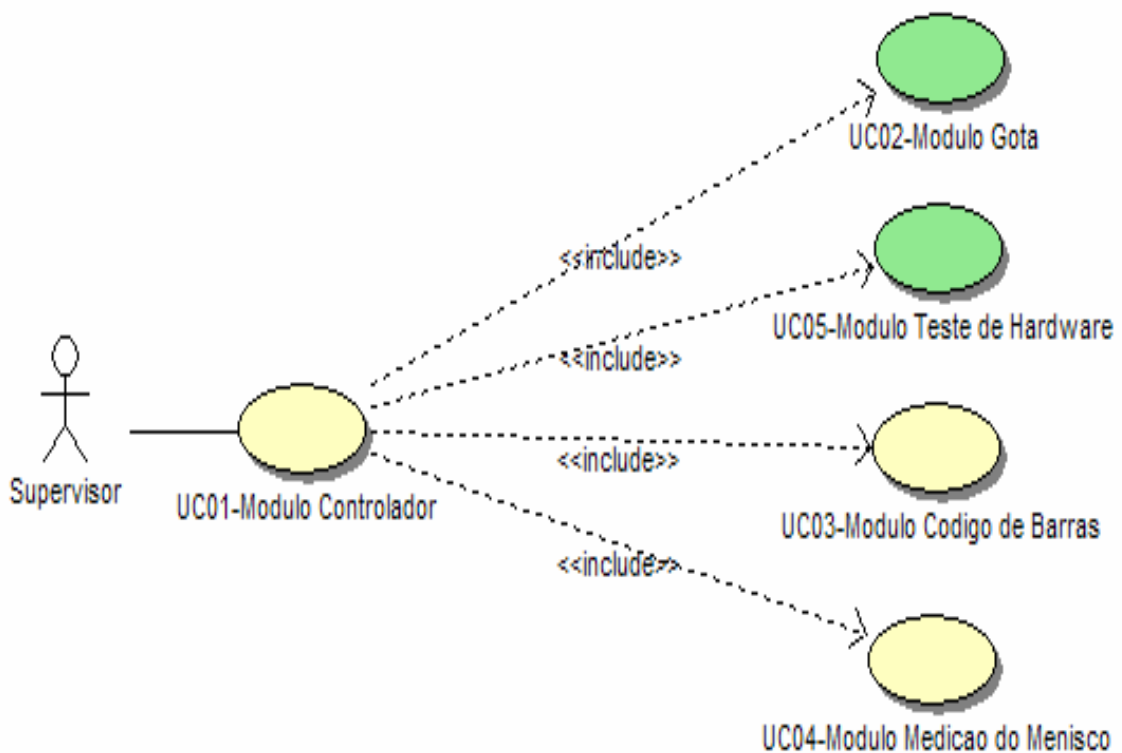


Figura 2.2: Diagrama de Casos de Uso do Sistema de Monitoramento

OBS: Os Módulos em bege foram desenvolvidos neste projeto, já os módulos em verde foram desenvolvidos em [1], porém o Modulo Codigo de Barras foi desenvolvido em

conjunto. Para maiores detalhes sobre os Módulos de Gota e Teste de Hardware ler [1] juntamente com este projeto.

2.2. Diagrama Físico do Módulo Controlador

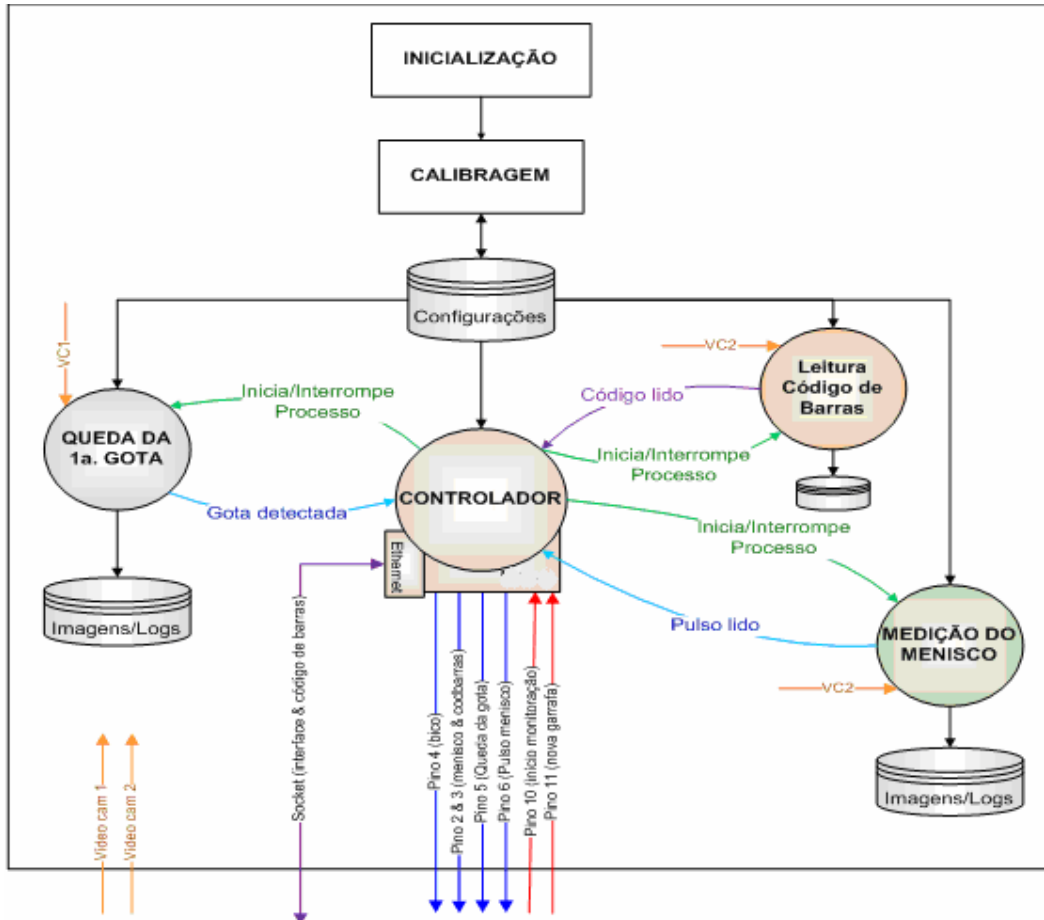


Figura 2.3: Diagrama Físico do Módulo Controlador

2.2.1. O Módulo Controlador

Descrição:

O Módulo Controlador é um módulo de interface que tem como função controlar, via porta paralela (LPT), o acionamento das câmeras e iluminações (atuadores) utilizadas no processo de destilação a batelada, além de enviar dois sinais, aviso de queda da 1ª gota e pulso a cada deslocamento do menisco, ao CLP (Circuito Lógico Programável), via Circuito Adaptador de Tensão (CAT). A partir de dois sinais enviados pelo CLP à LPT, via CAT, dar-se-á a seqüência de controle dos atuadores, tendo como interface com a LPT o CA. Com base em um protocolo de comunicação fornecido, o Módulo Controlador foi implementado como uma máquina de estados, que a partir de uma seqüência de entrada e estado, gera uma seqüência de saída; seu diagrama de estados será apresentado abaixo. O Controlador foi desenvolvido em uma plataforma Windows XP, que não permite o acesso direto aos registradores da LPT; como contorno do problema foi usada a *dll impout32* para o acesso dos registradores da LPT.

2.2.2. Caso de uso Módulo Controlador

Objetivo: Controlar o processo de destilação fracionada do petróleo

Pré-Condições: não há

Fluxo Principal:

1. Usuário chama o executável.
2. O Módulo Controlador carrega configurações iniciais (coloca os níveis lógicos da porta paralela em baixo, seta variáveis do sistema) [E0].
3. O Módulo Controlador realiza a rotina de teste de hardware[E1].
4. O Módulo Controlador espera informação do Módulo CLP [E2].
5. O Módulo CLP envia a informação de início de processo de monitoração.
6. O Módulo Controlador liga a câmera e a iluminação do bico do destilador.
7. O Módulo Controlador chama o Módulo Gota.
8. O Módulo Gota processa a queda da primeira gota.
9. O Módulo Gota seta para *true* uma variável interna do Módulo Controlador indicativa de detecção de queda de primeira gota.
10. O Módulo Controlador desliga a câmera e a iluminação do bico do destilador.
11. O Módulo Controlador envia um sinal via porta paralela para o Módulo CLP indicando a queda da primeira gota.
12. O Módulo CLP envia ao Módulo controlador o sinal de *nova garrafa*.
13. O Módulo Controlador recebe o sinal de *nova garrafa*.
14. O Módulo Controlador liga a câmera e a iluminação do código de barras.
15. O Módulo Controlador chama o Módulo Código de Barras.
16. O Módulo Controlador entra em espera.
17. O Módulo Código de Barras faz a leitura do código de barras.
18. O Módulo Código de Barras seta para *true* uma variável interna do Módulo Controlador indicativa de leitura do código de barras realizada com sucesso.
19. O Módulo Controlador desliga somente a iluminação do código de barras.
20. O Módulo Controlador liga a iluminação do menisco.
21. O Módulo Controlador chama o Módulo Menisco.
22. O Módulo Controlador fica recebendo as informações indicativas de incremento de nível vindas do Módulo Menisco.
23. O Módulo Controlador, para cada informação de incremento de nível, envia para o Módulo CLP um pulso indicativo de incremento de volume.

24. O Módulo CLP faz a contagem de pulsos.
25. O Módulo CLP envia sinal de troca de garrafa ao atingir a quantidade de pulsos pré-definidos ou acabar o subproduto antes de encher a garrafa.
26. O Módulo Controlador retorna ao passo 12, caso o processo de destilação ainda não tenha acabado.
27. O Módulo Controlador inicia o processo de encerramento do monitoramento, caso o Módulo CLP envie o sinal indicativo de fim do processo de destilação.
28. O Módulo Controlador desliga todas as câmeras e iluminações e o processo é encerrado.

Pós-Condições:

1. Não há.

Fluxos de exceção:

1. *Os fluxos de exceção não foram tratados.*

2.2.3. Diagrama de Etapas do Módulo Controlador

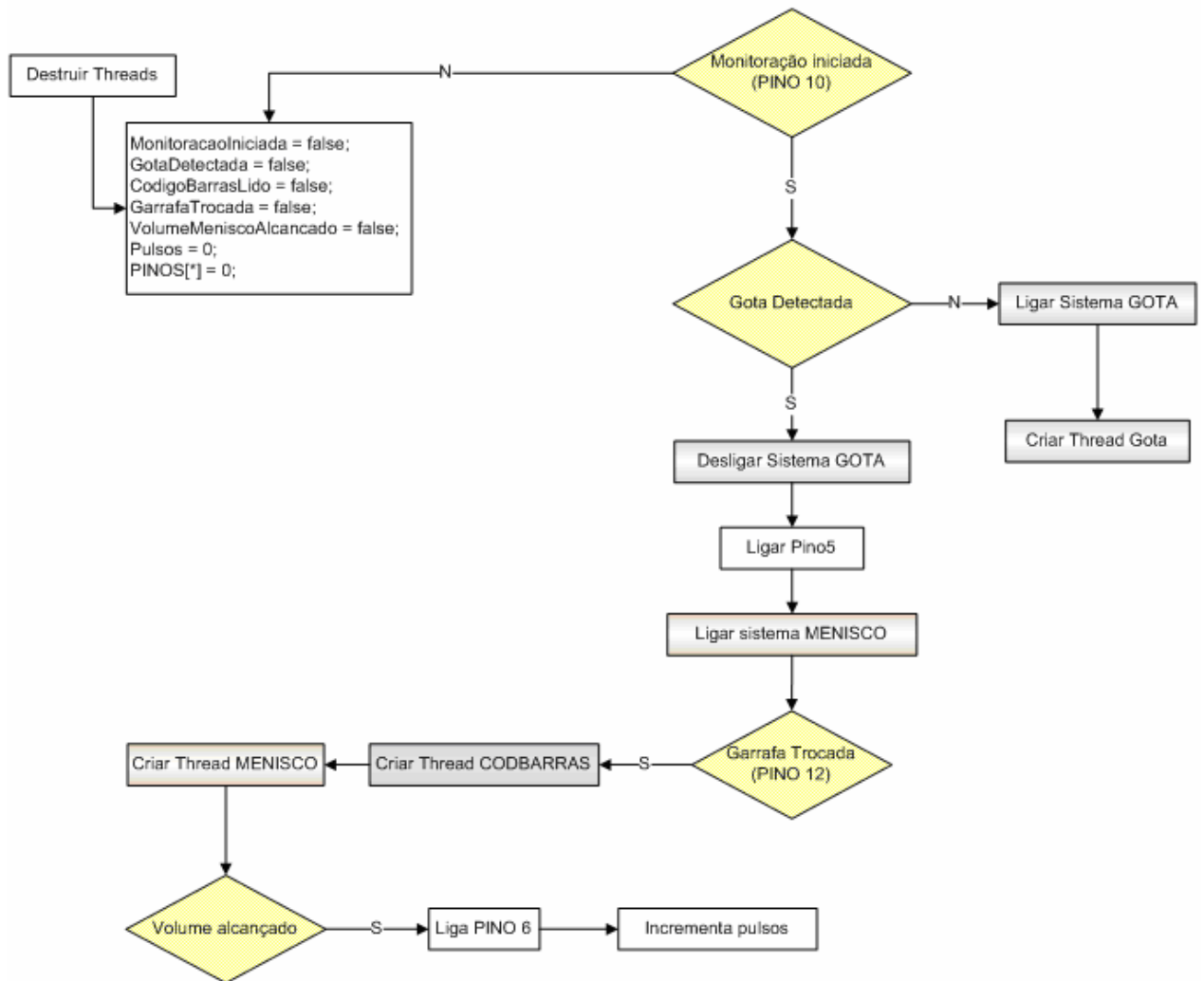


Figura 2.4: Diagrama de Etapas do Módulo Controlador

Onde PINO 6, PINO 10 e PINO 12 da Figura 2.3, referem-se aos pinos 6, 10 e 12 da porta paralela do PC.

2.2.4. Máquina de Estados Módulo Controlador

O Módulo Controlador foi implementado como sendo uma máquina de estados (Figura 2.5) possuindo dois métodos, são eles: “DefineEstado” e “DetectaTransição”. Toda vez que “DetectaTransição” é acionado, o método “DefineEstado” é executado para definir o estado novo. Há várias condições (“if”) dentro do método “DefineEstado”. Cada condição define um estado e retorna um número associado.

O método “DetectaTransição”, então, verifica se este número retornado é o mesmo do estado antigo. Se a sentença for verdadeira, o programa não realiza nenhuma operação. Se for falso, o método percorre algumas condições para certificar qual transição foi feita. Com isso, é gerada uma mensagem referente à transição, que permanece até que o nível lógico de algum objeto seja trocado.

2.2.5. Diagrama de Estados do Módulo Controlador

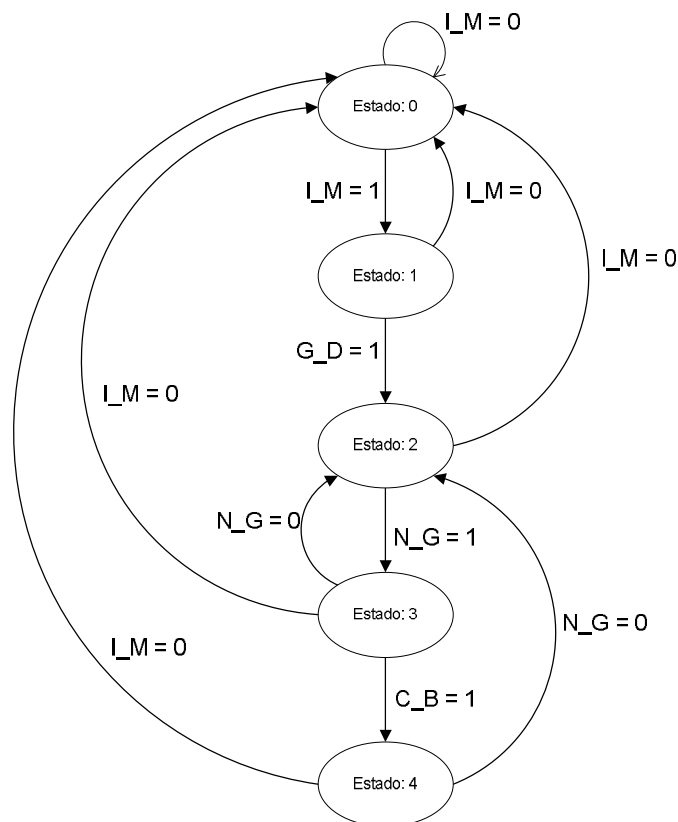


Figura 2.5: Diagrama de Estados do Módulo Controlador

2.2.5.1. Legenda da Figura 2.4:

- I_M: Início da monitoração;
- N_G: Nova garrafa;
- G_D: Gota detectada;
- C_B: Código de barras.

2.2.5.2. Definição dos Estados da Figura 2.4:

- Estado 0: “Aguardando início da monitoração”;
- Estado 1: “Monitoração iniciada”:
 - A câmera e iluminação do bico são ligadas;
 - Fica aguardando a queda da primeira gota.
- Estado 2: “Gota detectada”:
 - A câmera e iluminação do bico são desligadas;
 - Fica aguardando o posicionamento da garrafa.
- Estado 3: “Garrafa posicionada”:
 - Câmera e iluminação do código de barras são ligadas;
 - Fica aguardando leitura do código de barras.
- Estado 4: “Leitura do código de barras realizada”
 - A câmera e iluminação do código de barras são desligadas;
 - A câmera e iluminação do menisco são ligadas;
 - Medindo nível do menisco;
 - Fica aguardando a sinalização da troca de garrafa.

2.2.6. Classe do Módulo Controlador

```
#ifndef ControladorH
#define ControladorH
//-----
#include <Classes.hpp>
#include "ControladorUI.h"
#include "Global.h"
#include "Gota.h"
#include "CodigoBarras.h"
#include "Menisco.h"
#include "CamInterface.h"
#include "UTestaHardware.h"
//-----
class CONTROLADOR
{
private:
    TfrmControlador *frmControlador;
    Gota *gota;
    TWinSemaforoBin *sinc_fim_gota, *sinc_barras;
    CodBarras *cod_barras;
    Menisco *menisco;
    CamInterface *cam_interface;
    TTestaHardwareLPT *teste_h;
//    Variável da classe de teste de hardware
protected:
    void __fastcall Execute();
public:
    bool terminated;
    int estado;
    __fastcall CONTROLADOR(bool CreateSuspended);
    __fastcall ~CONTROLADOR(void);
    void __fastcall ResetVariaveis(int modo);
```

```

void __fastcall LigarCameraIluminacao(int sistema, int modo);
void __fastcall AvisaCLP(int msg);
void __fastcall EscreveMensagem(int msg);
bool __fastcall TestaCircuitos(void);
void __fastcall Gota(void);
void __fastcall CodigoBarras(void);
void __fastcall Menisco(int op);
void __fastcall Controle(void);
void __fastcall Finaliza(void);
void GravaTXT();

};
//-----
#endif

```

2.2.7. Diagrama de Atividades do Módulo Controlador

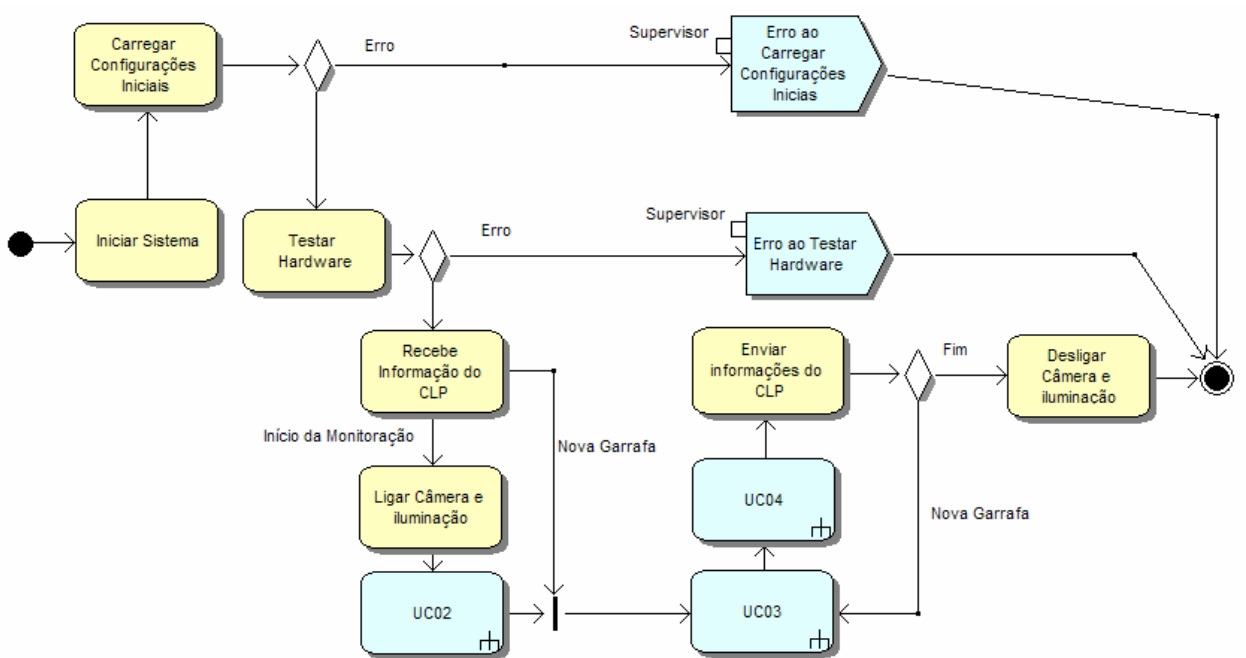


Figura 2.6: Diagrama de Atividades do Módulo Controlador

2.3. Módulo de Leitura e Identificação do Código de Barras

Descrição:

Os fracionamentos do composto destilado deverão ser colhidos em garrafas diferentes. Para identificar estas garrafas e, conseqüentemente, seus conteúdos, optou-se por utilizar uma etiqueta com um número de dois algoritmos codificado como código de barras.

Este módulo visa ler e identificar este código de barras que é impresso na parte inferior da garrafa. Após o posicionamento de uma nova garrafa, o algoritmo identifica o valor e o retorna ao Controlador.

2.3.1. Caso de uso Módulo Código de Barras

Objetivo:

1. Leitura de código de barras via processamento de imagem para associação das garrafas com seus subprodutos.

Pré-Condições:

1. Módulo Controlador ativo.
2. Módulo Gota ter detectado a queda da primeira gota.
3. Módulo Controlador envia sinal de inicialização do Módulo Código de barras.

Fluxo Principal:

1. Módulo Código de Barras identifica o código de barras.
2. Módulo Código de Barras envia ao Módulo Controlador o sinal de detecção do código de barras.
3. O Módulo Código de Barras é encerrado

Pós-Condições:

1. Serão desligadas a iluminação e a câmera do código de barras, e serão ligadas a câmera e a iluminação do menisco.

Fluxos de exceção:

2. *Os fluxos de exceção não foram tratados.*

OBS: Está sendo usado temporariamente um módulo comercial (Seção 2.3.)

2.3.2. Classe do Módulo Código de Barras

```
//-----  
#ifndef TCodigoBarrasH  
#define TCodigoBarrasH  
//-----  
  
#include <Graphics.hpp>  
#include <Dialogs.hpp>  
#include <SysUtils.hpp>  
#include <Printers.hpp>  
#include <algorithm>  
#include <fstream>  
#include <string>  
#include <vector>  
#include <map>  
#include <cstring>  
  
typedef Graphics::TBitmap TGBitmap;  
class TCodigoBarras{  
  
    class TCodBParam {  
    public:  
        int largBarra;  
        int largBarraMin;  
        int largBarraMax;  
        int largImg;  
        int altCodBMin;  
        int numDigCodB;  
        float limiarSim;  
        Byte limiarBin;  
        TRect dimImgImp;  
        TCodBParam(): largBarra(5),largBarraMin(1),largBarraMax(15),  
            largImg(750),altCodBMin(25),numDigCodB(2),  
            limiarSim(0.65),limiarBin(90),
```

```

        dimImgImp(50,50,5650,900){};
    };
private:
    //Atributos Privados
    bool    seg;
    TGBitmap    *imgCapturada,*imgBin,*imgRot,*imgSeg,
                *imgGerada,*imgComConexos;
    AnsiString  strCodBarrasRec;
    AnsiString  strCodBarrasGer;
    TRect    rectCodBarras;

    //Métodos Privados
    bool validarSeg(int X,int Y,int XFim, int *YFim, TGBitmap *pImgSeg);
    void gerarImgSeg(TRect pRect, TGBitmap *pImgSeg);
public:
    TCodBParam param;
    //Construtores
    TCodigoBarras(TGBitmap *pImgCap);
    TCodigoBarras();
    ~TCodigoBarras();
    //Métodos Públicos
    void binarizar(TGBitmap *pImgBin);
    void rotacionar(TGBitmap *pImgRot);
    bool segmentar(TGBitmap *pImgSeg);
    bool reconhecer(TGBitmap *pImgRec);
    void compConexos(TGBitmap *pImgComConexos);
    void gerarImgCodB(AnsiString pCodBarraDec);
    void gerarImgCodB(AnsiString pCodBarraDec,
                      int pLargBarra, int pLarg);
    void imprimir(TGBitmap *pImgImprimir, AnsiString pCodBarraDec);
    void imprimir(TGBitmap *pImgImprimir, AnsiString pCodBarraDec,
                  TRect r);
    void atribuirCodBarras(AnsiString pCodBarras);
    void atribuirImgCap(TGBitmap *pImgCapturada);

```



```

AnsiString obterCodBarrasRec();
AnsiString obterCodBarrasGer();
TRect obterRectCodBarras();
TGBitmap* obterImgCap();
TGBitmap* obterImgBin();
TGBitmap* obterImgRot();
TGBitmap* obterImgSeg();
TGBitmap* obterImgGerada();
};
#endif

```

2.4.3. Diagrama de Atividades do Módulo Código de Barras

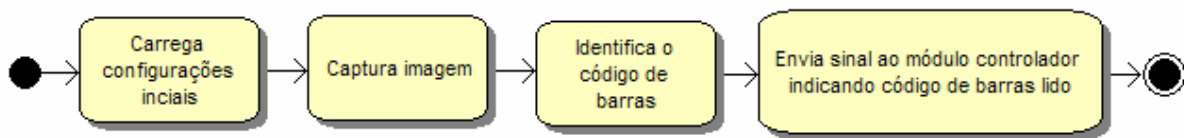


Figura 2.7: Diagrama de Atividades do Módulo Código de Barras

2.3.4. Garrafa Coletora

Para o desenvolvimento dos módulos de código de barras e menisco, foi utilizada uma garrafa construída segundo as especificações abaixo:

- Raio interno: **5,1 cm**
- Altura útil: **13 cm**
- Cor: **transparente**
- Altura do fundo falso para o código de barras: **5 cm**
- Capacidade: **1,0 litro**

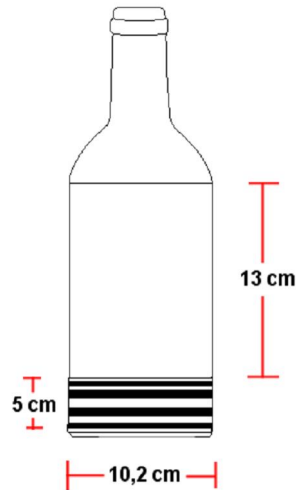


Figura 2.8: Garrafa Coletora

Problemas encontrados nas garrafas fornecidas:

- Deformações na área em volta do fundo: Estrias em torno da garrafa na altura do fundo. Estas estrias distorcem a imagem nesta área.
- Deformações no fundo: O fundo inserido na garrafa não é plano, possui ondulações concêntricas. Estas deformações mascaram a medida de volume nesta área.

2.3.5. Padrão de Codificação

O tipo de código de barras escolhido para o projeto, definido pela equipe anterior, foi o intercalado 2 de 5 (Figura 2.9), que é popular na indústria de transportes. Sua característica mais importante é ser bem compacto. A concepção da direção horizontal (Figura 2.10) visa otimizar a área de visualização na garrafa.



Figura 2.9: Modelo de código de barras do tipo intercalado 2 de 5



Figura 2.10: Código de barras intercalado 2 de 5 invertido

2.3.6. Testes de Desempenho

O algoritmo desenvolvido foi revisado e modificado em uma versão experimental para testes. O objetivo inicial foi verificar o desempenho do algoritmo e identificar falhas.

Para a avaliação de desempenho do algoritmo desenvolvido, foram feitas duas séries de tomadas de imagens seguindo as configurações abaixo. Uma série de imagens repete as configurações para resolução de 320x240 e outra série para imagens de 640x480.

Para cada experimento, foram utilizadas as tiras de código de barras impressas em um conjunto de 40 tiras sequenciadas como:

- De 01 a 10 -> código de barras de 5 linhas*
- De 11 a 20 -> código de barras de 6 linhas
- De 21 a 30 -> código de barras de 7 linhas
- De 31 a 40 -> código de barras de 8 linhas

**Unidade relacionada ao tamanho das barras e espaços*

Experimento 1

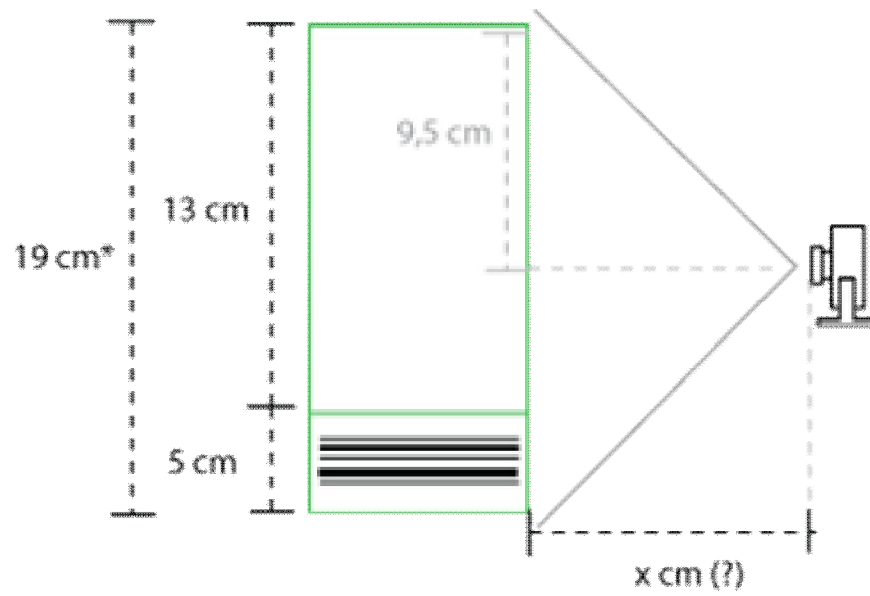


Figura 2.11: Tomada de fotos com o eixo focal alinhado com o centro da garrafa

*19 cm é a altura da garrafa incluindo a altura da *divisão de vidro*

Experimento 2

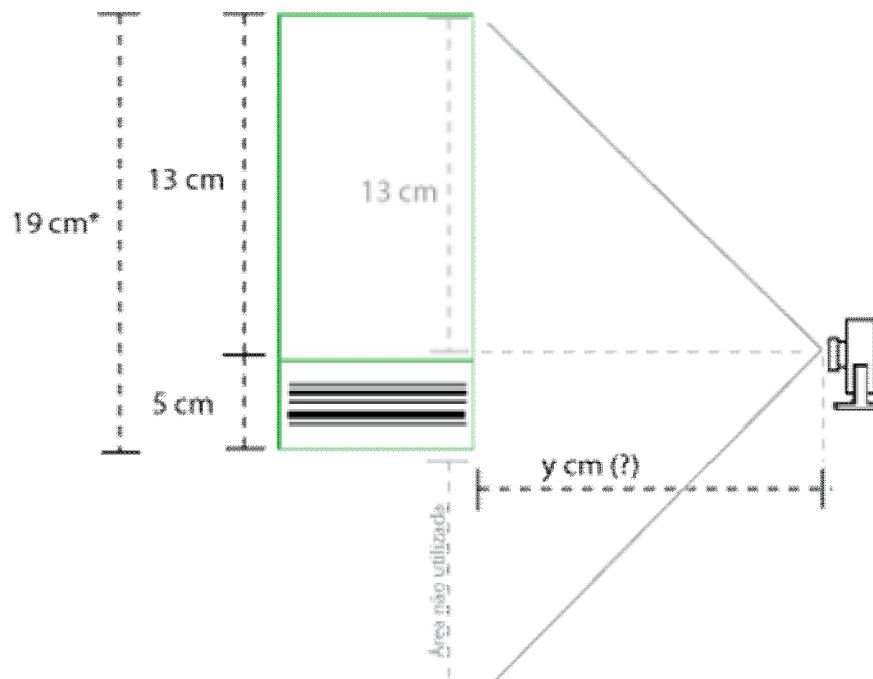


Figura 2.12: Tomada de fotos com o eixo focal alinhado com o fundo

Experimento 3

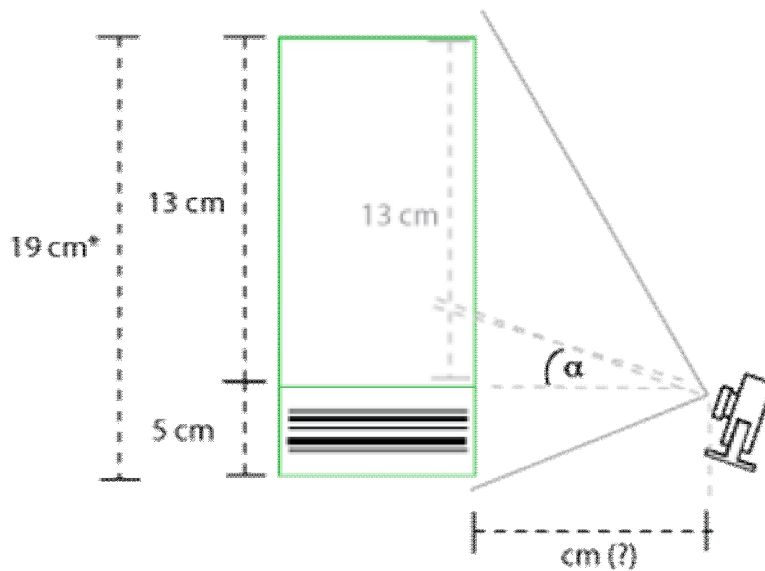


Figura 2.13: Tomada de fotos com a câmera inclinada enquadrando toda a garrafa

Observação: Trabalhei no código do programa abaixo (Figura 2.14), pré-existente. Foram feitas algumas alterações, mas os resultados não estão bons (índice de acertos baixo). Por isso usei um componente comercial que não possui interface, ele recebe um bitmap e informa o código de barras para um arquivo tipo txt que será enviado para o Supervisor, via OPC.

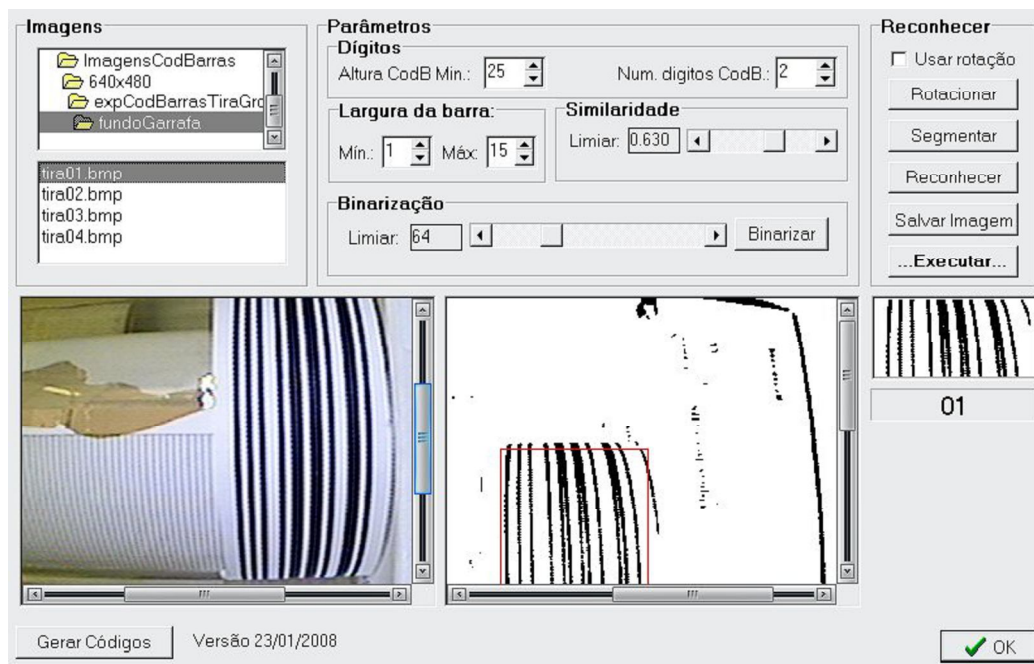


Figura 2.14: Programa de testes para identificação do código de barras

Os testes preliminares mostraram que o algoritmo tem um desempenho melhor utilizando imagens com resolução de 640x480.

2.4. Módulo de Medição do Menisco

Descrição:

Este módulo tem como objetivo identificar mudanças incrementais de volume e informar estas mudanças ao Controlador. Toda vez que a posição do menisco atinge um deslocamento Δh em centímetros, será enviado um sinal para o *sistema Supervisório* através do Módulo Interface *CLP/CAT* e serão armazenados um determinado número (valor parametrizado) de quadros na fila chamada de *quadros pós análise*. Estes quadros correspondem aos quadros nos quais foi detectado o deslocamento procurado e quadros anteriores e posteriores a este evento. A garrafa coletora será trocada toda vez que atingir seu limite máximo de armazenamento ou quando o *sistema Supervisório* determinar o começo da coleta de uma nova subfração de petróleo. É no momento em que a garrafa está sendo posicionada que a *análise* pausa seu funcionamento e o Módulo de *Log* começa a funcionar. O Módulo de *Log* é o encarregado de salvar em disco os quadros que se encontram na fila citada acima, e de processar as exceções ou mensagens que possam ter sido geradas durante a execução da detecção do menisco. Quando o sistema Supervisório envia o sinal indicando que a nova garrafa se encontra posicionada, o Módulo de Log será pausado e o processo se repetirá desde o primeiro passo indefinidamente, até o sistema Supervisório enviar o sinal de final da destilação.

2.4.1. Caso de Uso do Módulo Menisco

Objetivo:

1. Monitoramento da variação de nível de subproduto contido no interior da garrafa.

Pré-Condições:

1. Módulo Controlador ativo.
2. Módulo Código da Gota já detectou a queda da primeira gota.
3. Módulo Código de Barras já identificou o código de barras.
4. Módulo Controlador envia sinal de criar *thread* (Módulo Menisco).

Fluxo Principal:

1. Módulo Menisco lê as configurações iniciais.
2. Cria *thread* Captura de Quadros.
3. *Thread* Captura de Quadros começa a salvar quadros uma fila.
4. Calibragem da localização da garrafa.
5. Cria *Thread* Análise dos Quadros.
6. *Thread* Análise dos Quadros pega quadros da fila para processá-los [E1].
7. *Thread* Análise de Quadros envia notificações tais como número do quadro onde os níveis são detectados, hora, data para a fila notificação.
8. *Thread* Análise de Quadros envia o quadro com o nível detectado, bem como alguns quadros anteriores e posteriores ao nível detectado.
9. *Thread* Geração dos Logs é criada.
10. *Thread* Geração dos Logs retira informações das filas para armazenamento em arquivos de log.

Pós-Condições:

1. Módulo Controlador recebe sinal indicativo de nova garrafa.
2. Nova garrafa posicionada para monitoramento.

Fluxos de exceção:

E1[O subproduto termina antes de a garrafa completar o nível].

E0.1 O Módulo CLP envia um sinal indicativo de nova garrafa ao Módulo Controlador.

E0.2 O Módulo Controlador envia um sinal para o Módulo Menisco entrar em suspensão.

E0.3 O Módulo Controlador chama o Módulo Gota.

E0.4 O Módulo Gota envia ao Módulo Controlador o sinal indicativo de detecção e queda de primeira gota.

E0.5 O Módulo Controlador chama o Módulo Código de Barras.

E0.6 O Módulo Código de Barras envia ao Módulo Controlador o sinal indicativo de leitura de código de barras.

E0.7 O Módulo Controlador envia ao Módulo Menisco o sinal indicativo para sair do estado de suspensão.

E0.8 O Módulo Menisco retorna ao passo 3.

2.4.2. Diagrama Físico do Módulo Menisco

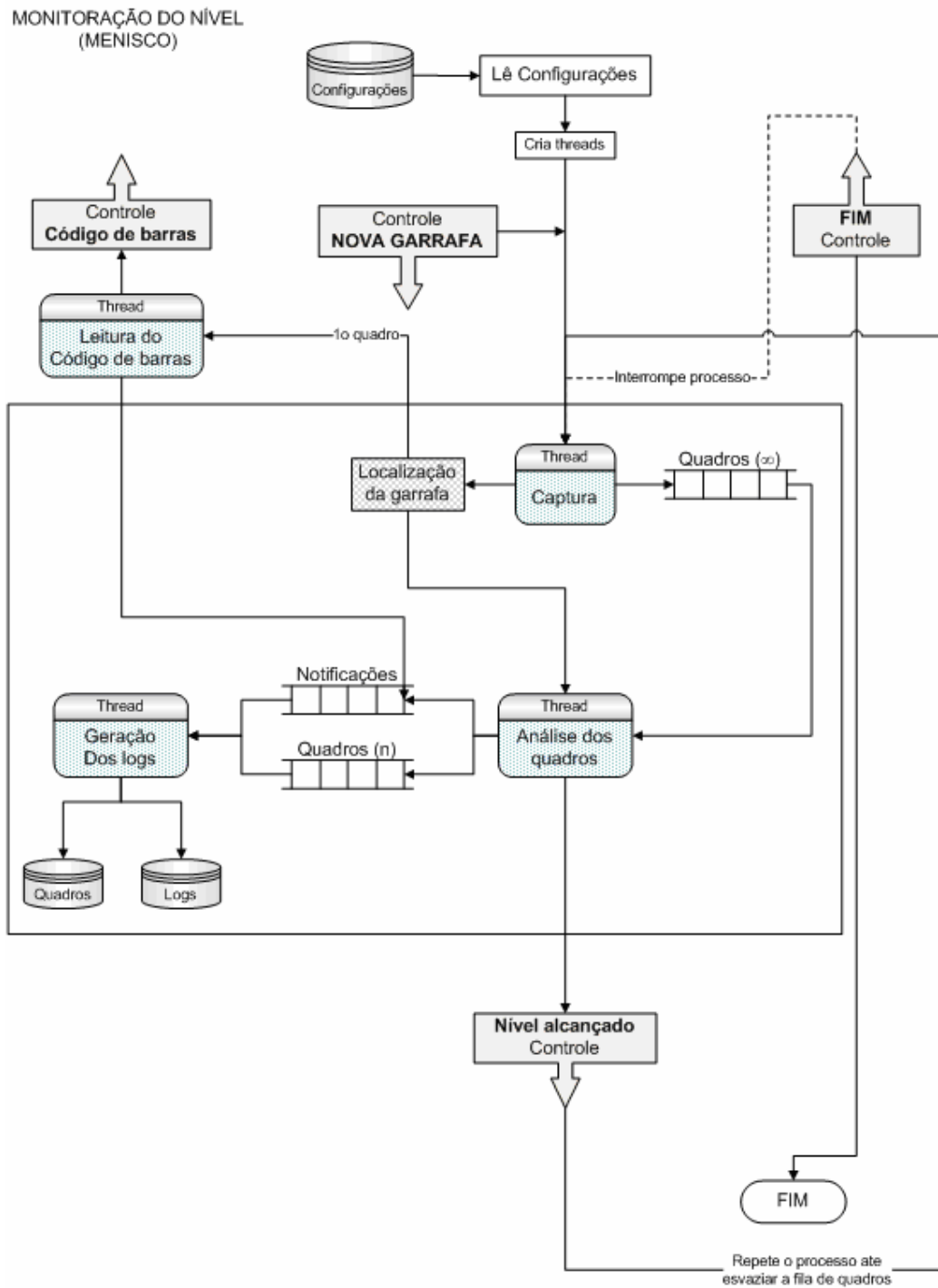


Figura 2.15: Diagrama Físico do Módulo Menisco

2.4.3. Classe do Módulo Menisco

```
//-----  
/*  
    DetectorMenisco.h  
    Deteccao do menisco (versao 1)  
*/  
//-----  
#ifndef DetectorMeniscoH  
#define DetectorMeniscoH  
//-----  
#ifndef TCoordsDefined  
/*  
    TCoords  
    Coordenadas da area de busca  
*/  
struct TCoords { int x1, y1, x2, y2; };  
#define TCoordsDefined  
#endif  
  
#include "Projecoes.h"  
#include "ImgFX.h"  
#include "FileUtils.h"  
/*  
    JANELA_ESQUERDA  
    Define janela de busca a esquerda  
  
    JANELA_CENTRAL  
    Define janela de busca central  
  
    JANELA_DIREITA  
    Define janela de busca 'a direita  
*/  
#define JANELA_ESQUERDA    0
```

```

#define JANELA_CENTRAL      1
#define JANELA_DIREITA     2
/*
TDetectorMenisco
Define classe para deteccao do menisco
*/
class TDetectorMenisco: public TProjcoes
{
    int tempTipo;
    int dbCnt;
    TCoords coordsCalibragem;
    double vetor_interpolacao[60];
    BOOL vetorDefinido;
    double volMax;
    TBmp *mascara, *img_bin;
    bool debug, mascaraDefinida, usaMascara;
    int janela;
    int tamanho_da_janela; // divisor
    BOOL garrafaVazia;
    int alturaGarrafaPX;
public:
    int __fastcall AlturaGarrafaPX(void) { return alturaGarrafaPX; }
    __fastcall TDetectorMenisco(int limBin=200, int limProjs=2, double
volumeTotal = 1200.0);
    __fastcall ~TDetectorMenisco(void);
    void __fastcall DefineCoords(int x1, int y1, int x2, int y2);
    void __fastcall DefineCoords(TCoords &coords);
    TCoords &getCoords() { return coordsCalibragem; }
    int __fastcall VolLidoPX(int *proj, int size, int lim);
    void __fastcall DefineVetorInterpolacao(void);
    void __fastcall DefineVetorInterpolacao(double *vetor);
    BOOL __fastcall VetorDefinido(void) { return vetorDefinido; }
    int __fastcall IniciaIdentificacaoDoVolume(Graphics::TBitmap *imgOrg);
    int __fastcall IdentificaVolumeEmPixels(Graphics::TBitmap *imgOrg);

```

```

double __fastcall CalculaVolumeRealComVetor(Graphics::TBitmap *imgOrg);
double __fastcall CalculaVolumeRealSemVetor(Graphics::TBitmap *imgOrg);
int __fastcall DefineMascara(Graphics::TBitmap *bmp);
double __fastcall ValorInterpolado(double volLido, int ndx);
void __fastcall SetDebug(bool modo);
double __fastcall VolumeTotal(void);
double __fastcall VolumeTotal(double vol);
void __fastcall MascaraSub(Graphics::TBitmap *bmp2);
TRect __fastcall DefineMascaraRect(void);
void __fastcall DefineJanela(int jan);
void __fastcall DefineTamanhoDaJanela(int tam);
void __fastcall UsaMascara(bool modo);
BOOL __fastcall UsaMascara(void);
void __fastcall GarrafaVazia(BOOL estado);
BOOL __fastcall GarrafaVazia(void);
void __fastcall TempTipo(int tt) { tempTipo = tt; }
int __fastcall TempTipo(void) { return tempTipo; }
};
#endif

```

2.4.4. Diagrama de Atividades do Módulo Menisco

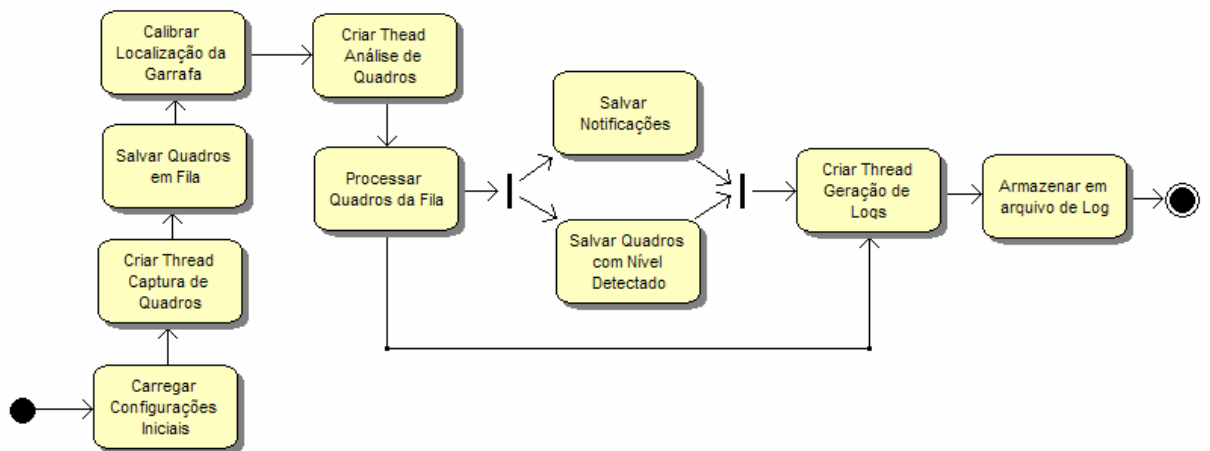


Figura 2.16: Diagrama de Atividades do Módulo Menisco

2.4.5. Infra-Estrutura do Sistema Menisco

Para o desenvolvimento do algoritmo do menisco foram utilizados dois tipos de câmeras. Uma com ângulo de abertura de cerca de 52° com luminosidade 1,0 lux e outra com ângulo de abertura de cerca de 92° e luminosidade de 0,1 lux. Uma vantagem na utilização da câmera de maior ângulo de abertura é a melhor resolução da imagem capturada, porém como desvantagem, uma grande distorção pode ser observada (Figura 2.17).

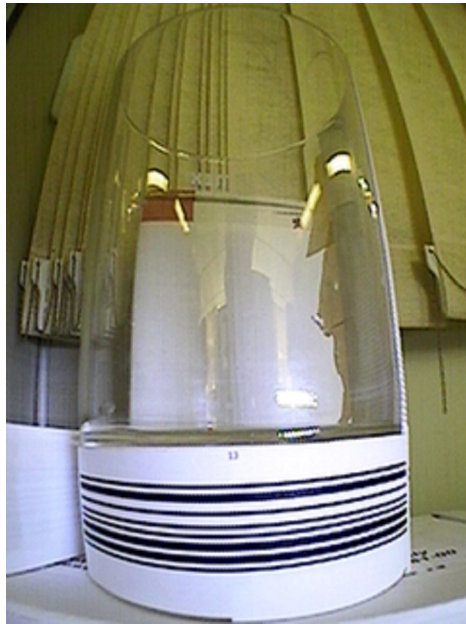


Figura 2.17: Distorção apresentada com o uso da câmera de maior abertura

Outra característica desta câmera é a sensibilidade luminosa de 0,1 lux. Isto permite uma focalização com baixa luminosidade; isto é, a imagem é capturada com a iluminação ambiente.

Esta característica, porém, satura a imagem quando exposta a uma iluminação artificial (iluminadores). Como a iluminação da área de monitoração é necessária, ainda estão sendo estudadas novas idéias para a solução deste problema.

A vantagem da câmera de menor ângulo de abertura é a distorção mínima apresentada, porém como desvantagem, percebe-se uma perda na resolução da imagem capturada.

Os testes que estão sendo realizados visam observar:

- Resolução mínima para identificação do menor volume
- Configuração ideal - posição da câmera. *Ver configurações de experimentos na definição do Módulo de Código de Barras (Seção 2.3.2.).*
- Comportamento de líquidos de densidades diferentes

2.4.5.1. Calibragem do Sistema Ótico-Menisco

A calibragem para este sistema define o retângulo de busca para a monitoração do volume. A altura deste retângulo irá definir a altura real da área útil da garrafa (volume total). O contraste deve ser ajustado para se obter a máxima área clara no interior da garrafa, preservando as bordas.

A figura abaixo (Figura 2.18) mostra a interface de um dos programas utilizados no processo de calibragem do Sistema Ótico-Menisco. Através desta interface podemos calibrar a área útil da imagem para ser usada no momento de varredura em busca da detecção do nível de líquido no recipiente (menisco).

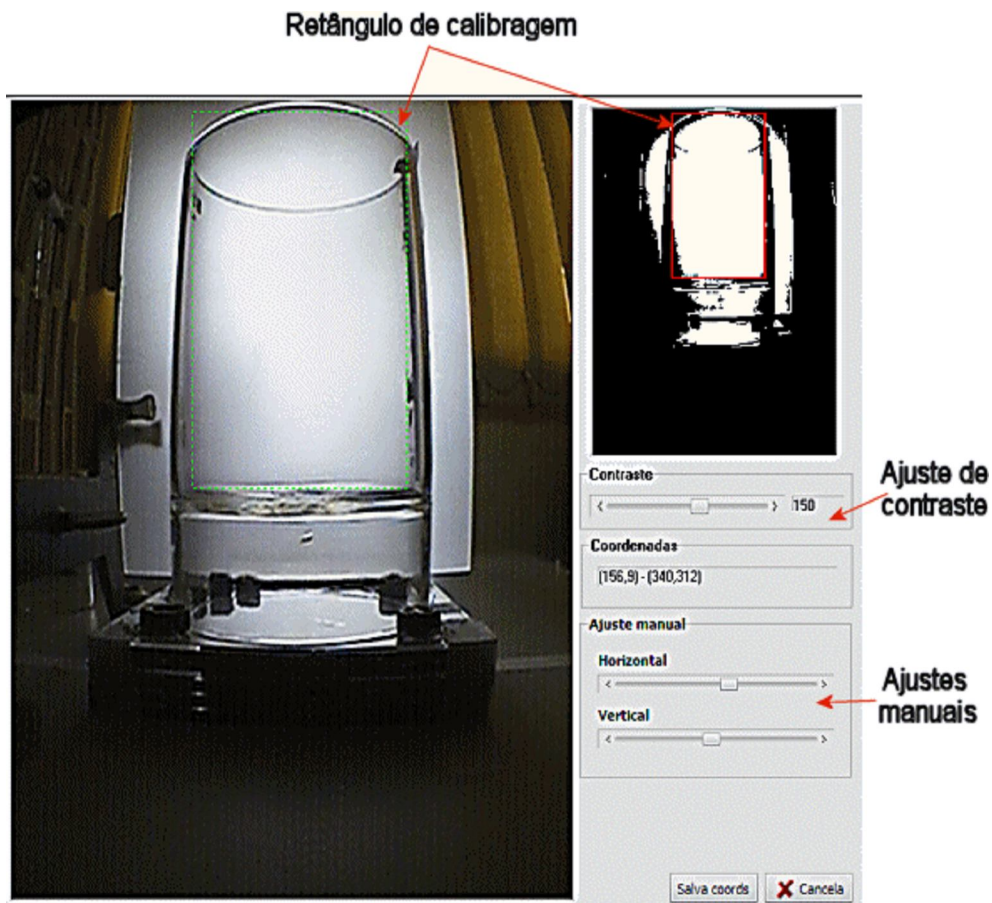


Figura 2.18: Calibragem do Sistema Menisco

2.4.5.2. Calibragem do Sistema Físico-Menisco

Este procedimento deverá ser realizado apenas uma vez e serve para armazenar valores utilizados na varredura da imagem realizada pelo algoritmo do Módulo Menisco. Para isto, é necessário tomar uma seqüência de fotos de 20 ml em 20 ml até completar 1000 ml completando o volume total da garrafa. Este procedimento deverá ser feito nas seguintes condições:

- Usar líquido de densidades diferentes (transparente, alaranjado e escuro)
- Tirar as fotos nos três tipos de configuração (Seção 2.3.6.).
- A iluminação não deve ocultar as bordas do líquido (evitar saturação)
- Devem-se evitar reflexos
- Usar as duas câmeras disponíveis (repetir o experimento para cada câmera)

Com base nestas recomendações foram realizados alguns testes revelando um ótimo desempenho na identificação de volumes.

A figura abaixo (Figura 2.19) mostra a interface de um dos programas utilizados para fazer a calibragem do menisco.

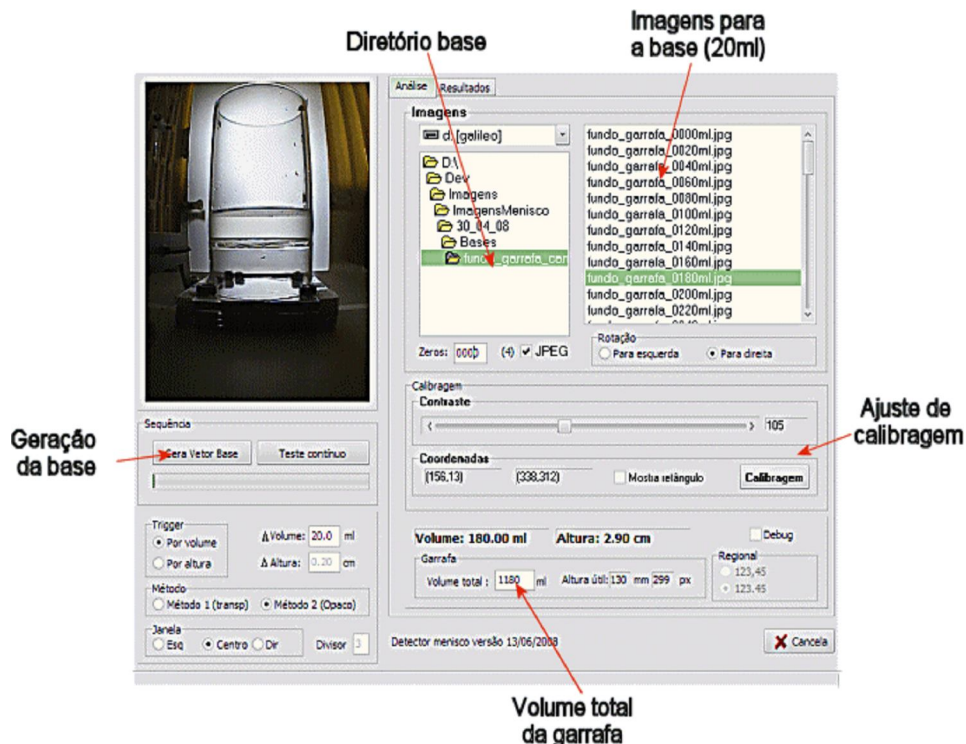


Figura 2.19: Um exemplo da Amostragem de Teste

2.4.5.3. Resultados dos Testes de Desempenho

Tabela 1: Câmera posicionada no fundo da garrafa

Vol (ml)	Algoritmo	Diferença
20,00	20	0,00
40,00	40	0,00
60,00	60	0,00
80,00	80	0,00
100,00	99,74	-0,26
120,00	119,74	-0,26
140,00	140	0,00
160,00	163,69	3,69
180,00	175,79	-4,21
200,00	195,79	-4,21
220,00	227,63	7,63
240,00	240	0,00
260,00	260	0,00
280,00	279,74	-0,26
300,00	299,74	-0,26
320,00	320	0,00
340,00	343,69	3,69
360,00	360	0,00
380,00	380	0,00
400,00	403,69	3,69
420,00	420	0,00
440,00	439,74	-0,26
460,00	459,74	-0,26
480,00	480	0,00
500,00	495,79	-4,21
520,00	520	0,00
540,00	540	0,00
560,00	560	0,00
580,00	580	0,00
600,00	600	0,00
620,00	620	0,00
640,00	635,79	-4,21
660,00	660	0,00
680,00	680	0,00
700,00	703,69	3,69
720,00	720	0,00
740,00	731,85	-8,15
760,00	760	0,00
780,00	780	0,00
800,00	799,74	-0,26
820,00	815,79	-4,21
840,00	840	0,00
860,00	860	0,00
880,00	879,74	-0,26
900,00	895,79	-4,21
920,00	919,74	-0,26
940,00	940	0,00
960,00	960	0,00
980,00	980	0,00
1000,00	999,74	-0,26
1020,00	1020	0,00
1040,00	1031,9	-8,10
1060,00	1059,7	-0,30
1080,00	1075,8	-4,20
1100,00	1095,8	-4,20
1120,00	1120	0,00
1140,00	1135,8	-4,20
1160,00	1155,8	-4,20
1180,00	1175,8	-4,20

Tabela 2: Câmera posicionada no meio da garrafa

Vol (ml)	Algoritmo	Dif.erença
20,00	20	0,00
40,00	164,99	124,99
60,00	77,34	17,34
80,00	92,31	12,31
100,00	114,76	14,76
120,00	134,72	14,72
140,00	152,18	12,18
160,00	22,49	-137,51
180,00	180	0,00
200,00	202,46	2,46
220,00	220	0,00
240,00	239,96	-0,04
260,00	260	0,00
280,00	280	0,00
300,00	299,96	-0,04
320,00	314,97	-5,03
340,00	344,95	4,95
360,00	360	0,00
380,00	380	0,00
400,00	400	0,00
420,00	419,96	-0,04
440,00	439,96	-0,04
460,00	460	0,00
480,00	479,96	-0,04
500,00	489,98	-10,02
520,00	519,96	-0,04
540,00	540	0,00
560,00	567,45	7,45
580,00	580	0,00
600,00	592,48	-7,52
620,00	617,46	-2,54
640,00	640	0,00
660,00	660	0,00
680,00	677,47	-2,53
700,00	699,96	-0,04
720,00	720	0,00
740,00	740	0,00
760,00	762,46	2,46
780,00	780	0,00
800,00	800	0,00
820,00	817,47	-2,53
840,00	840	0,00
860,00	860	0,00
880,00	880	0,00
900,00	897,47	-2,53
920,00	920	0,00
940,00	940,02	0,02
960,00	959,97	-0,03
980,00	980,04	0,04
1000,00	1000	0,00
1020,00	1017,5	-2,50
1040,00	1040	0,00
1060,00	1062,5	2,50
1080,00	1080	0,00
1100,00	1097,5	-2,50
1120,00	1112,5	-7,50
1140,00	1140	0,00
1160,00	1160	0,00

Tabela 3: Câmera inclinada posicionada no fundo da garrafa

Vol (ml)	Algoritmo	Diferença
20,00	20	0,00
40,00	40	0,00
60,00	61,01	1,01
80,00	78,38	-1,62
100,00	100	0,00
120,00	120	0,00
140,00	140	0,00
160,00	160	0,00
180,00	183,64	3,64
200,00	200	0,00
220,00	221,01	1,01
240,00	241,01	1,01
260,00	260	0,00
280,00	281,01	1,01
300,00	300	0,00
320,00	321,01	1,01
340,00	340	0,00
360,00	360	0,00
380,00	381,01	1,01
400,00	400	0,00
420,00	420	0,00
440,00	441,01	1,01
460,00	460	0,00
480,00	480	0,00
500,00	501,01	1,01
520,00	518,38	-1,62
540,00	538,38	-1,62
560,00	561,01	1,01
580,00	578,38	-1,62
600,00	600	0,00
620,00	620	0,00
640,00	640	0,00
660,00	660	0,00
680,00	680	0,00
700,00	700	0,00
720,00	720	0,00
740,00	740	0,00
760,00	760	0,00
780,00	778,39	-1,61
800,00	795,75	-4,25
820,00	820	0,00
840,00	840	0,00
860,00	855,76	-4,24
880,00	878,38	-1,62
900,00	900	0,00
920,00	915,76	-4,24
940,00	940	0,00
960,00	960	0,00
980,00	980,04	0,04
1000,00	1000	0,00
1020,00	1020	0,00
1040,00	1040	0,00
1060,00	1058,4	-1,60
1080,00	1080	0,00
1100,00	1100	0,00
1120,00	1115,8	-4,20
1140,00	1140	0,00
1160,00	1160	0,00

2.5. Módulos Auxiliares

Ambientes de testes para avaliação de desempenho

2.5.1. Programa de avaliação de desempenho do algoritmo de detecção do menisco - nível de líquido no recipiente (Figura 2.20)

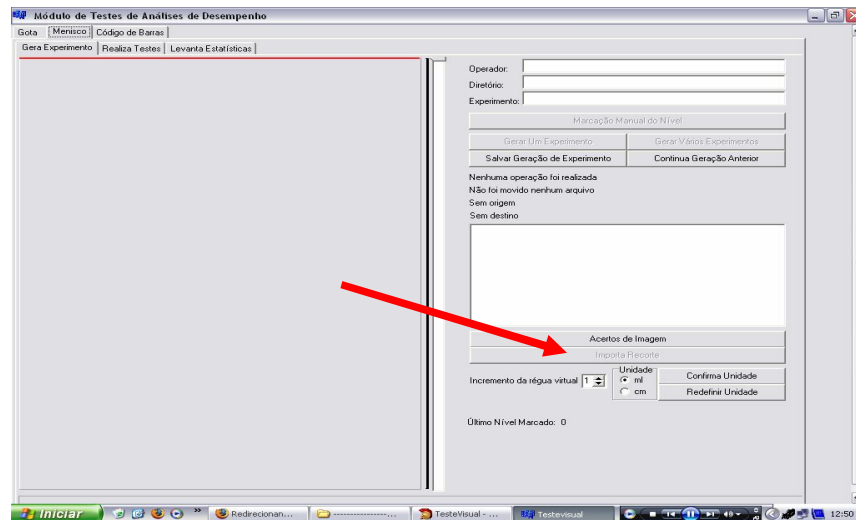


Figura 2.20: Início da Marcação

Com o objetivo de avaliar o desempenho do algoritmo de detecção do menisco e encontrar a configuração ideal do sistema, desenvolveu-se o programa de apoio, Módulo de Testes do Menisco.

Numa primeira etapa o sistema de monitoramento foi montado em diferentes configurações e a captura de quadros foi iniciada. Estas imagens foram divididas em diretórios, “experimentos”, sendo que as fotos contidas em cada diretório correspondiam a uma determinada configuração do Sistema de Monitoramento, ou seja, posição de câmeras, iluminação etc.

Numa segunda etapa, com o uso do Módulo de Testes do Menisco cada um dos diretórios citados foram selecionados para uma marcação manual dos quadros neles contidos, pois bem, em cada diretório continha um número de imagens onde o nível de líquido era o mesmo. Essas imagens foram obtidas manualmente enchendo-se de 20 em 20 ml o recipiente e salvando-se as imagens em um diretório. Esta é uma das finalidades do Módulo de Teste do Menisco, possibilitar marcar e salvar a imagem em um respectivo diretório.

Na terceira etapa foi rodado, sobre cada diretório, o algoritmo de detecção do menisco, o qual lê os quadros destes diretórios para detectar o nível de líquido do recipiente, e assim fazer a marcação automática destes quadros, e então armazenar o resultado nos respectivos arquivos de configuração do sistema.

Na quarta etapa, através da aba de estatística do Módulo de Teste do Menisco, rodou-se a rotina de avaliação de desempenho. Esta usa as informações armazenadas nos arquivos de configuração do sistema para montar uma tabela de estatísticas. Em outras palavras, a rotina de estatística compara se os quadros marcados automaticamente estão dentro da tolerância de cada quadro marcado manualmente, levantando assim a estatística de desempenho do algoritmo de detecção do menisco.

A partir desta observação foi possível avaliar o desempenho do algoritmo. Estes procedimentos são importantes, uma vez que o desenvolvimento do algoritmo foi realizado sobre uma pequena base de dados que não contempla todas as possibilidades de eventos. Estas etapas servem também para se avaliar as condições ambientais ótimas, que são mandatórias para o processo final.

2.5.2. Avaliação dos Circuitos de Hardware

Para avaliar o funcionamento dos circuitos (hardware) montados no laboratório, foi desenvolvido um programa de testes simulando os eventos que serão controlados pelo sistema (Figura 2.21).

Eventos:

- Recebimento do sinal de início da operação
- Ativação do sistema gota (câmera / iluminador)
- Envio de sinal da queda da gota
- Desativação do sistema gota
- Recebimento do sinal de garrafa-posicionada
- Ativação do sistema de leitura do código de barras (câmera / iluminador)
- Ativação do sistema do menisco (câmera / iluminador)
- Envio do sinal do diferencial de volume
- Recebimento do sinal de mudança para nova-garrafa
- Recebimento do final de operação
- Desativação do sistema

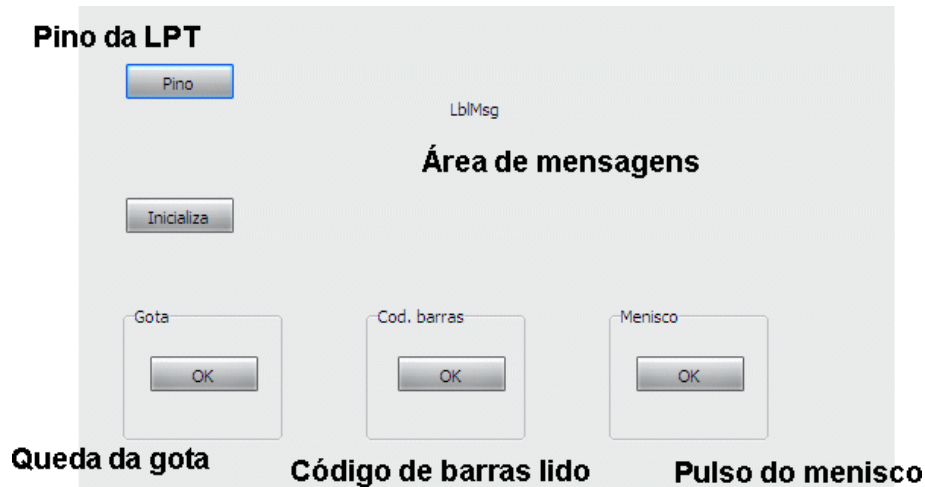


Figura 2.21: Programa experimental para simulação dos sinais enviados aos circuitos

Simulador do Sistema

Para validar a concepção *multi-thread* para o sistema, foi implementado um programa para simular o comportamento de cada módulo independente e a interação dos módulos com o Módulo Controlador (Figura 2.22).

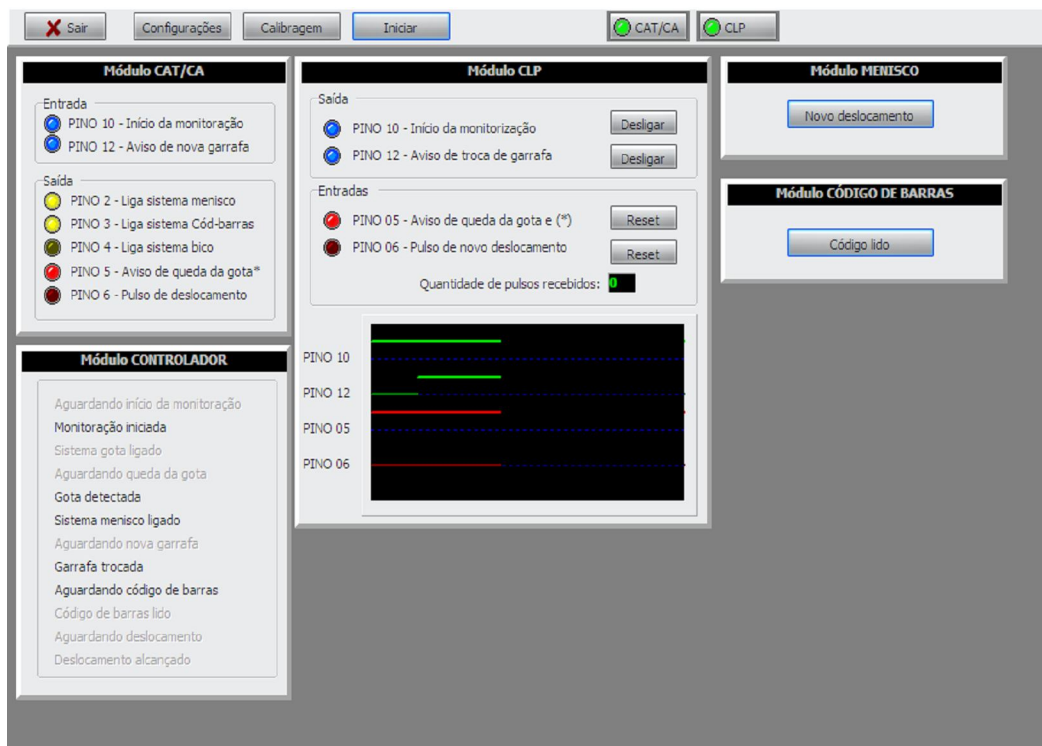


Figura 2.22: Simulador do Sistema Principal

2.6. Módulos de Hardware

2.6.1. Circuito Adaptador de Tensão – CAT

Este circuito tem a função de casar o nível lógico da porta paralela (LPT), nível TTL, com o nível lógico do CLP, que é 0V/24V. A especificação funcional deste circuito é a seguinte: quando a LPT escreve os níveis lógicos 0V ou 5V nos seus pinos (saída da LPT e entrada do CAT), teremos como saída do CAT para o CLP os níveis 24V ou 0V respectivamente. Quando o CLP envia os níveis lógico 0V ou 24V para o CAT, este envia os níveis lógicos 0V ou 5V respectivamente para os pinos de entrada da LPT.

O isolamento entre a LPT e o CAT foi realizado com o emprego do buffer 74LS541, visando proteger a LPT do computador.

Diagrama de Blocos

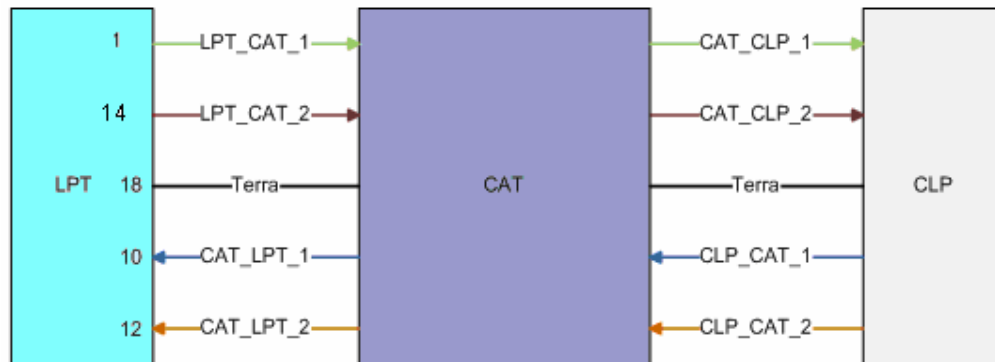


Figura 2.23: Diagrama de Blocos do Circuito CAT

Lógica de Relacionamento da Entrada com a Saída (CAT):

$$\begin{aligned}
 \text{LPT_CAT_1: } & \left\{ \begin{array}{l} \xrightarrow{\quad} 24\text{V} \\ \xrightarrow{5\text{V}} 0\text{V} \end{array} \right\} : \text{CAT_CLP_1} \\
 \text{LPT_CAT_2: } & \left\{ \begin{array}{l} 0\text{V} \xrightarrow{\quad} 24\text{V} \\ 5\text{V} \xrightarrow{\quad} \end{array} \right\} : \text{CAT_CLP_2} \\
 \text{CLP_CAT_1: } & \left\{ \begin{array}{l} 0\text{V} \xrightarrow{\quad} \\ 24\text{V} \xrightarrow{\quad} 5\text{V} \end{array} \right\} : \text{CAT_LPT_1} \\
 \text{CLP_CAT_2: } & \left\{ \begin{array}{l} 0\text{V} \xrightarrow{\quad} \\ 24\text{V} \xrightarrow{\quad} 5\text{V} \end{array} \right\} : \text{CAT_LPT_2}
 \end{aligned}$$

Figura 2.24: Diagrama Esquemático de Entrada\Saída

Tabela 4: Descrição da Pinagem da LPT com o CAT

Pino	Descrição	Tipo
10	Começo da Monitoração (Acknowledge)	Entrada
12	Aviso de Nova Garrafa (Busy)	Entrada
1	Aviso Queda da Primeira Gota (*)	Saída
14	Pulso a Cada Deslocamento do Menisco	Saída
18	Terra	Comum

(*) também usado para sinalizar a conclusão do processamento de cada garrafa.

2.7. A interface do Sistema Principal e seu Funcionamento

As Figuras (2.25 a 2.32) mostram a interface principal do sistema e todas as possíveis mensagens que o operador do sistema verá durante a operação do sistema. As mensagens são:

- Circuito OK: Indicando que o teste de hardware foi realizado com sucesso.
- Monitoração iniciada. Bico ligado. Aguardando Gota: Indicando que o sistema está aguardando o sinal para o monitorando da queda da primeira gota.
- Gota detectada. Bico desligado. Aguardando posicionamento da garrafa: Indicando que o sistema está esperando o sinal para o início do monitoramento do nível de líquido no recipiente (menisco).
- Monitorando Menisco: Indicando que o menisco está sendo monitorado.
- Código de Barras lido: Indicando que o código de barras já foi lido.
- Garrafa finalizada: Indicando que o monitoramento de determinada garrafa foi finalizado.
- Aguardando posicionamento de nova garrafa: Indicando a espera do posicionamento de uma nova garrafa.
- Garrafa posicionada: Indicando o recebimento do sinal de que nova garrafa foi posicionada.
- Aguardando leitura do código de barras e menisco: Indicando que o sistema está na etapa da leitura do código de barras.
- Código de barras lido: Indicando que o sistema conseguiu ler o código de barras.
- Volume alcançado: Indicando os níveis de líquido detectado.

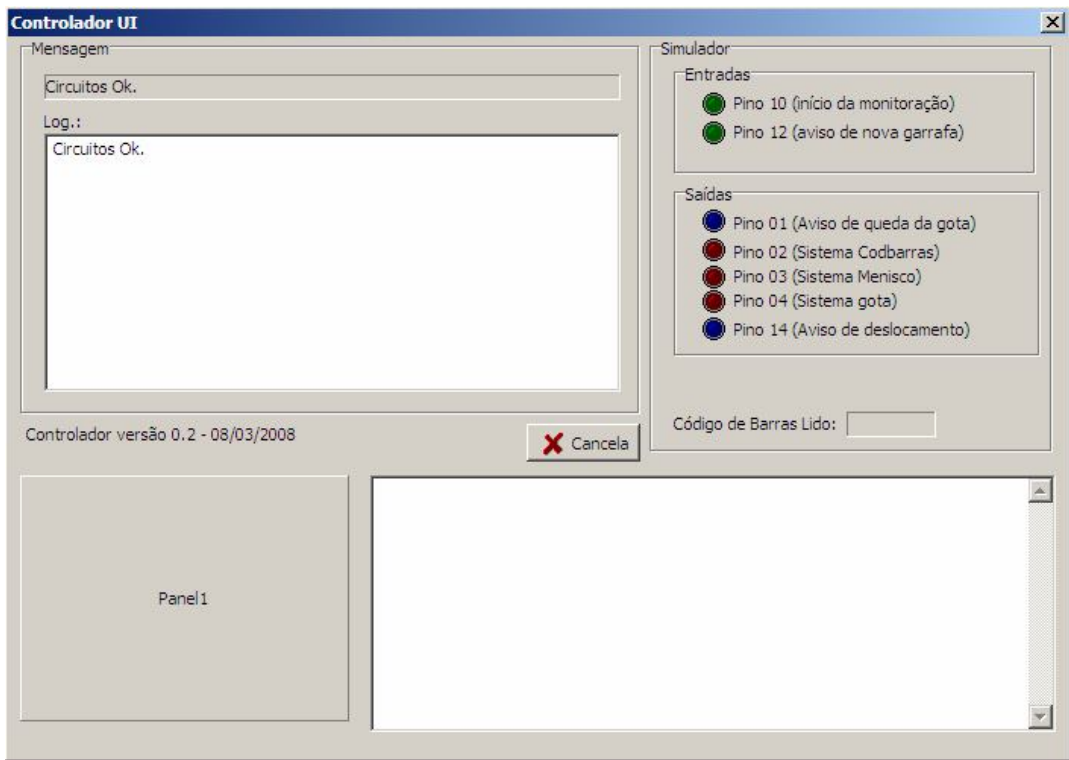


Figura 2.25: Controlador UI

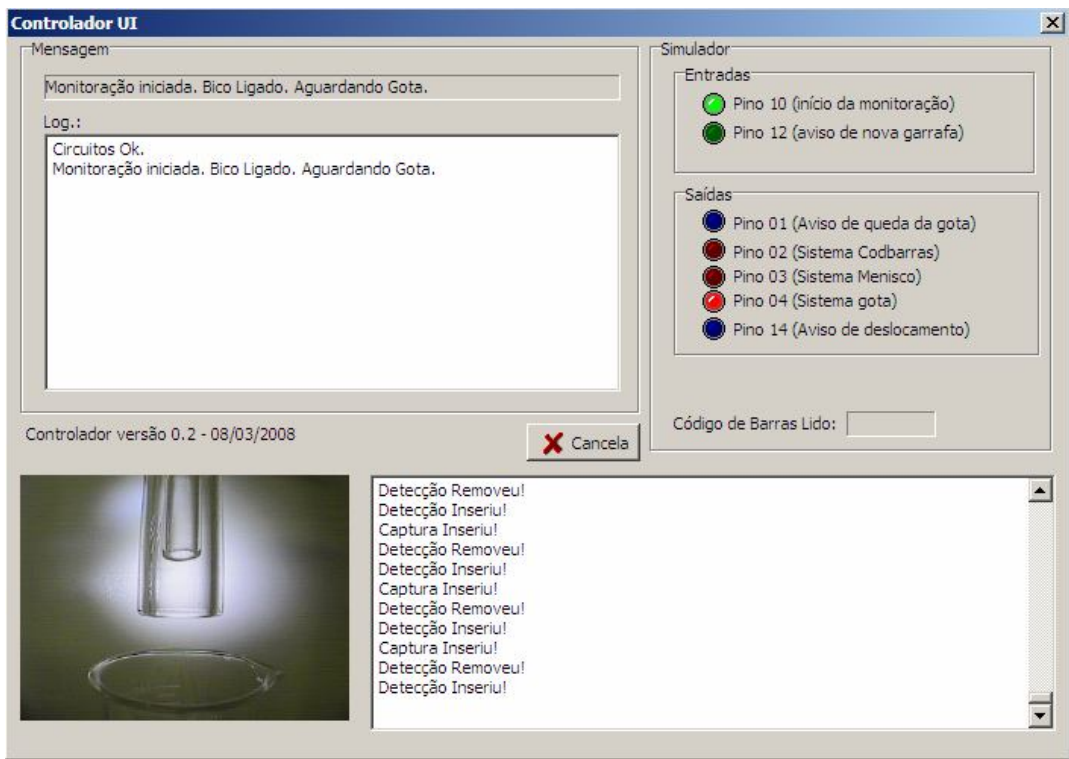


Figura 2.26: Aguardando a Gota

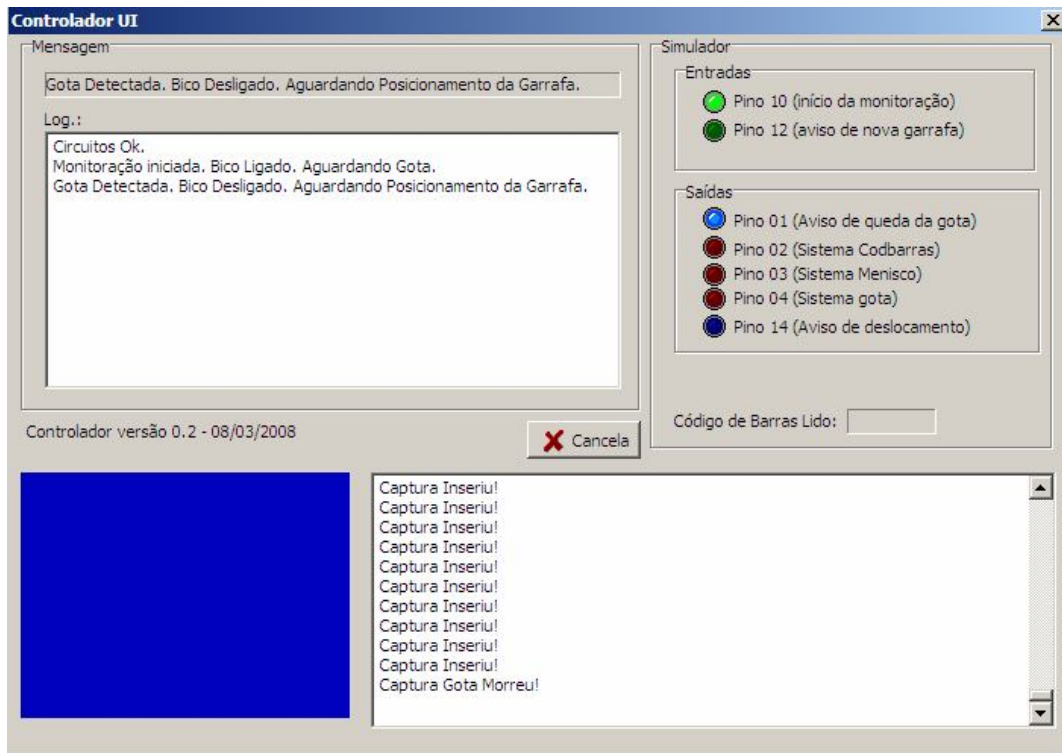


Figura 2.27: Gota Detectada

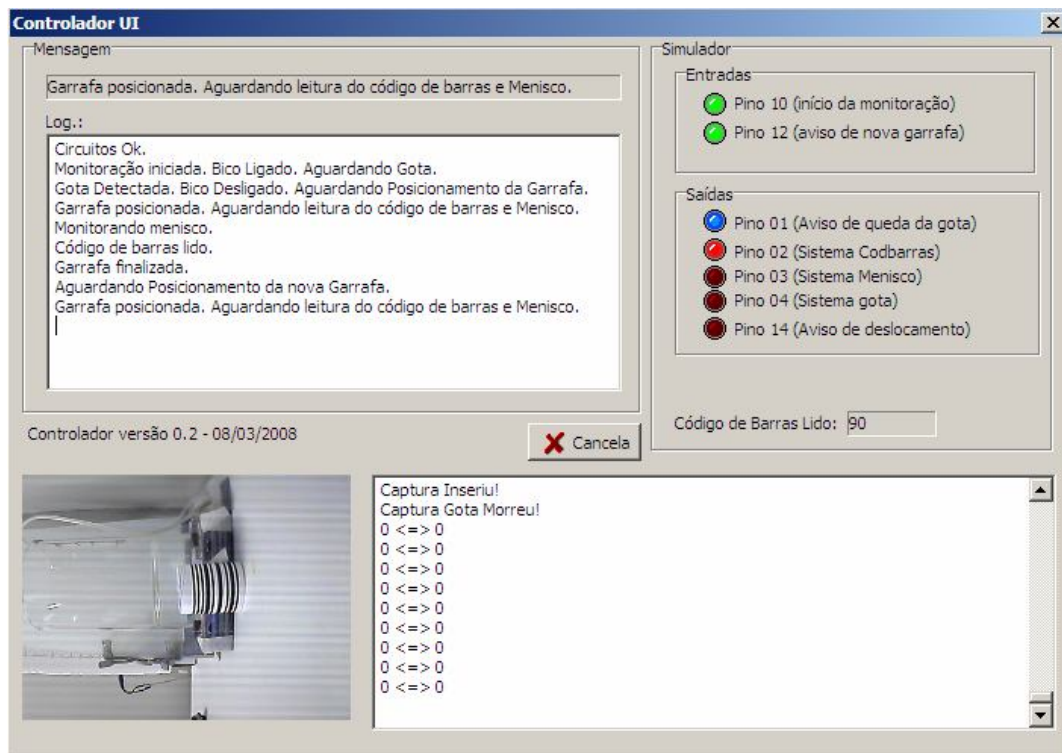


Figura 2.28: Aguardando Leitura do Código

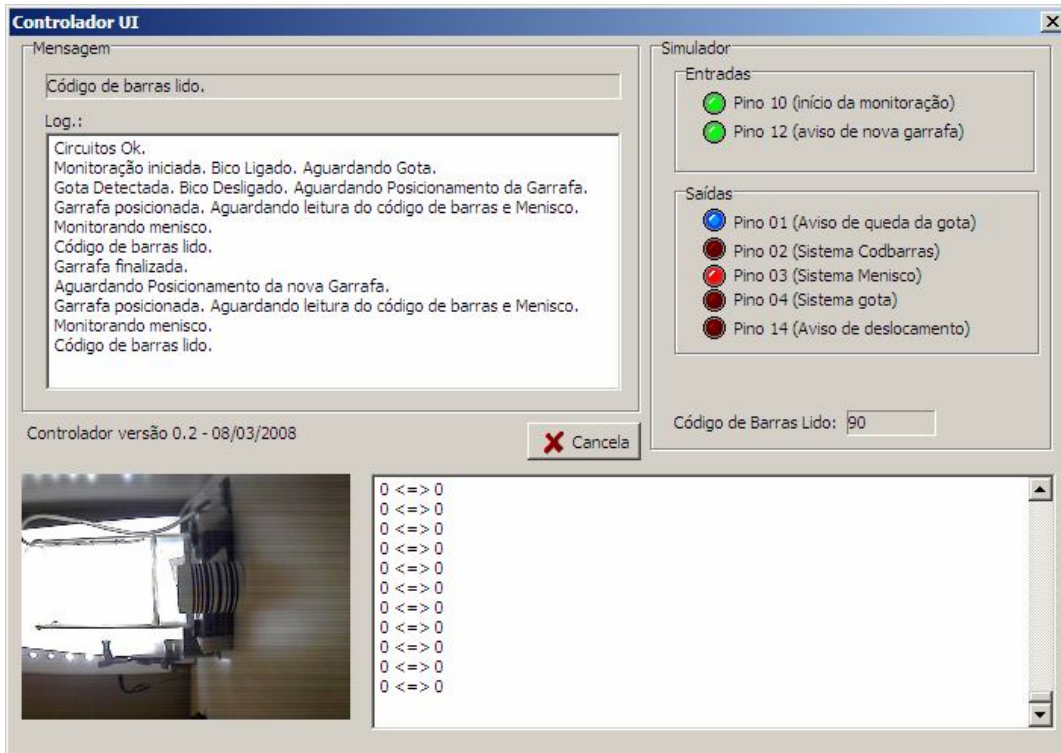


Figura 2.29: Código de Barras Lido

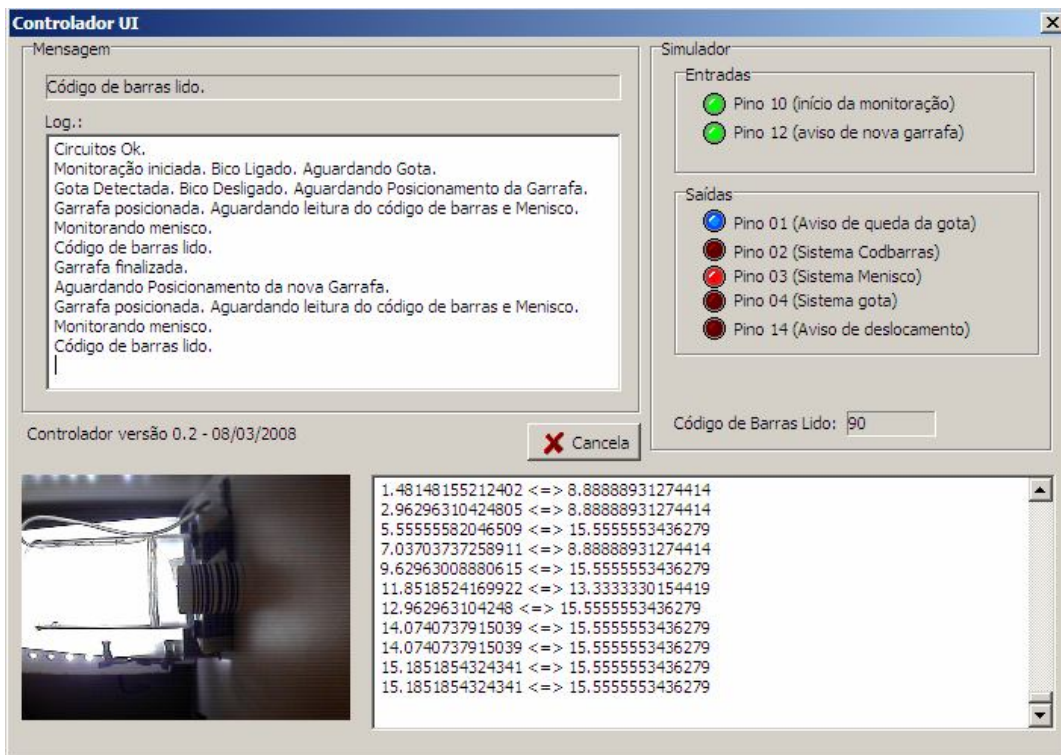


Figura 2.30: Início Acompanhar Nível

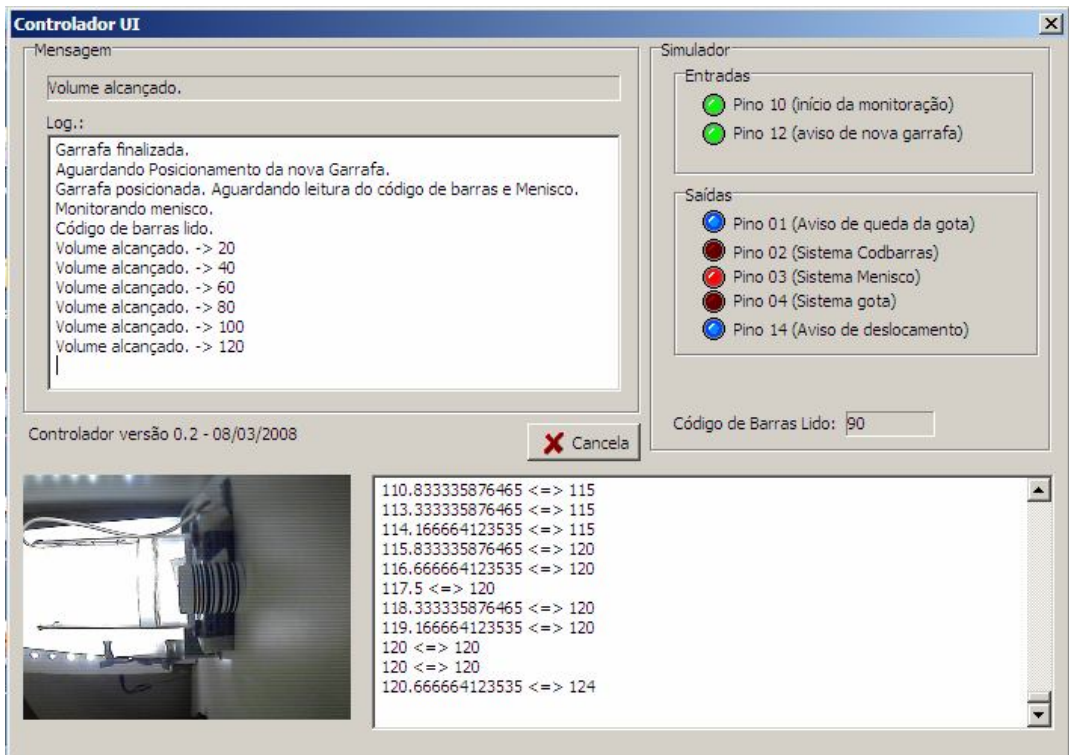


Figura 2.31: Nível variando

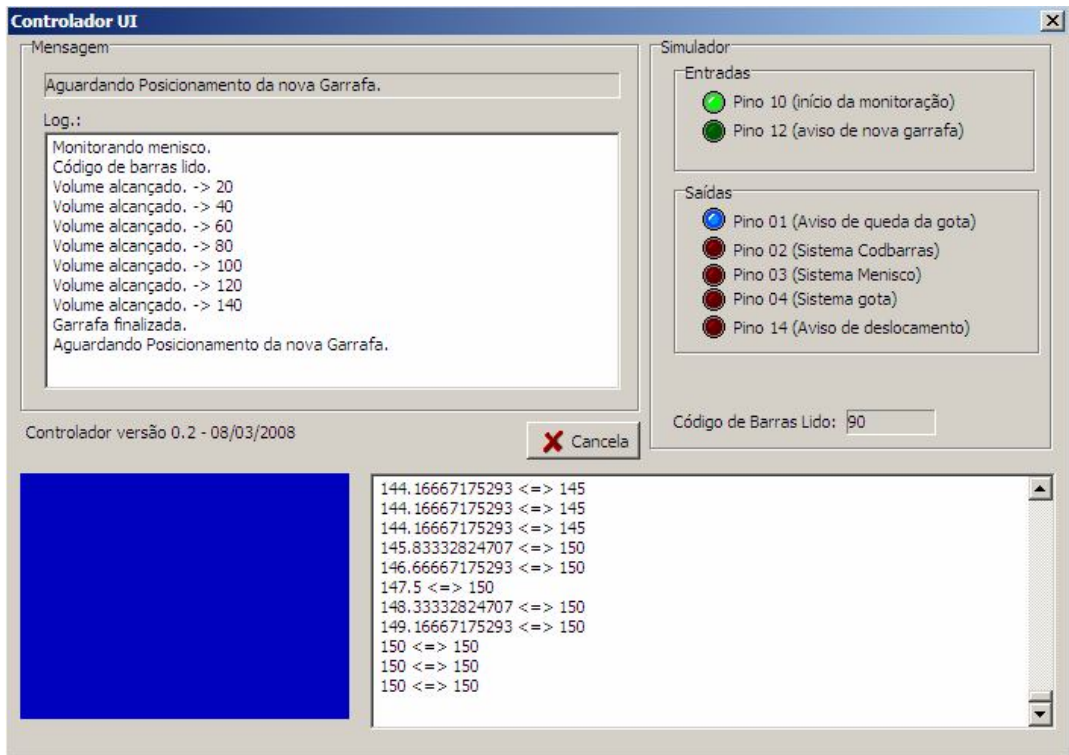


Figura 2.32: Esperando Nova Garrafa

Capítulo 3

O processamento de imagem

Fundamentação Teórica

Este capítulo descreve os fundamentos, as etapas do processamento digital de imagens e como as técnicas de processamento de imagem foram aplicadas ao projeto.

3.1. Fundamentos para Processamento Digital de Imagens

Esta seção apresenta as propriedades da imagem, passando pela definição, a forma de representação, o armazenamento e as propriedades do pixel.

3.1.1. Imagem em Escala de Cinza

Segundo [2], a definição de imagem em escala de cinza pressupõe a existência de dois conjuntos: O conjunto **E** formado por quadrados adjacentes, dispostos ao longo de um número de linhas e colunas (Figura 3.1), e o conjunto **K**, formado pelos níveis de cinza ou escalas de cinza, com **K** variando de 0 a 255.

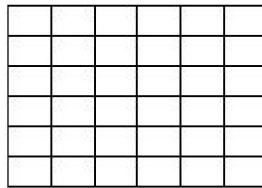


Figura 3.1: Conjunto E

Para [2] um pixel é um elemento do produto cartesiano de um quadrado localizado em **E** e um nível de cinza localizado em **K**, ou seja, o par ordenado (x, s) , onde x é um elemento do conjunto **E** e s um elemento do conjunto **K** (Figura 3.2). O mesmo autor define a imagem em escala de cinza como sendo um gráfico de mapeamento de **E** para **K**.

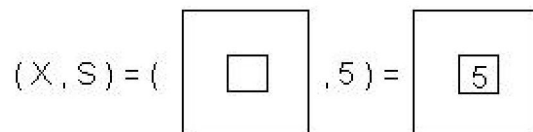


Figura 3.2: Produto Cartesiano E x K

3.1.2. Definição Matemática de Imagem

Segundo [3] uma imagem pode ser definida matematicamente como uma função $f(x, y)$, onde o valor nas coordenadas espaciais xy corresponde ao brilho (intensidade) da imagem nessa coordenada (Figura 3.3).



Figura 3.3: Imagem em Níveis de Cinza

3.1.3. Imagem Digital

Uma imagem de um objeto real é em princípio contínua tanto na sua variação espacial como nos níveis de cinza [3]. Visando o seu processamento computacional a imagem deve ser digitalizada, ou seja discretizar tanto no espaço quanto na amplitude.

A digitalização das coordenadas espaciais é chamada amostragem da imagem e a digitalização da amplitude é chamada de quantização dos níveis de cinza [4].

3.1.4. Representação de Imagens

Existem duas maneiras de representar uma imagem: representação vetorial e representação matricial. Estas representações diferem na natureza dos dados que compõe a imagem.

Segundo [5] uma imagem vetorial é uma imagem de natureza geométrica, ou seja, ela é definida em função de elementos geométricos e parâmetros (Figura 3.4).



Figura 3.4: Representação Vetorial [5]

Segundo [5] uma imagem matricial é uma imagem de natureza discreta, ou seja, a imagem é formada de elementos independentes, dispostos na forma de uma matriz, cada um contendo uma unidade de informação da imagem (Figura 3.5). Esta representação não armazena nenhuma informação geométrica dos objetos contidos na imagem o que torna difícil a manipulação de sua estrutura. No entanto é capaz de representar qualquer tipo de imagem.

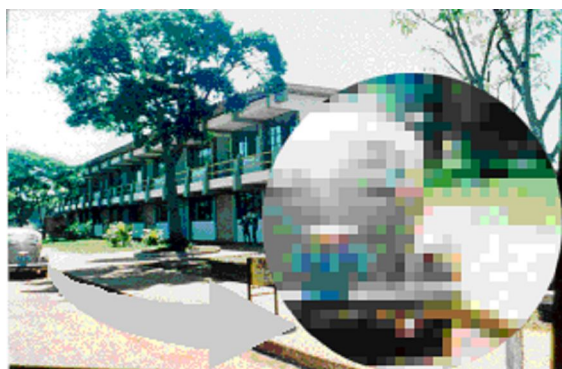


Figura 3.5: Representação Matricial [5]

3.1.5. Propriedades de um Pixel

Uma propriedade importante de um pixel é sua vizinhança, esta seção define a vizinhança 4 e a vizinhança 8 de um pixel.

•Vizinhança 4

A vizinhança 4 de um pixel P é definida pelo conjunto dos pixels adjacentes a P, não levando em conta os pixels localizados nas diagonais passando por P (Figura 3.6).

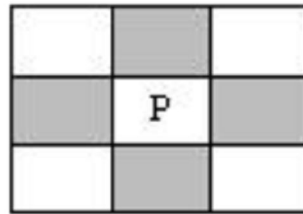


Figura 3.6: Vizinhança 4 de um Pixel

•Vizinhança 8

A vizinhança 8 de um pixel P é formada pelo conjunto de todos os pixels que são adjacentes a P (Figura 3.7).

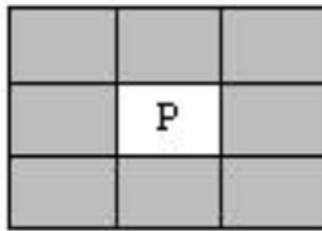


Figura 3.7: Vizinhança 8 de um Pixel

3.1.6. Formatos de Imagens

O formato de uma imagem refere-se ao modo como os dados da imagem estão armazenados em arquivos e como se faz a interpretação desses dados para a visualização da imagem. Neste projeto foram considerados os seguintes formatos de armazenamento de imagem:

- JPEG;
- PNG;
- BitMap;
- Tiff;

•Formato JPEG

Segundo [6] o formato JPEG é o mais utilizado no armazenamento e na transmissão de imagens estáticas em multimídia e na *internet*. Dentre as principais características do JPEG estão:

- Grande capacidade de compressão;
- Padronização internacional;
- Pequena perda perceptível
- Ampla aceitação pela comunidade da *internet*

•Formato PNG

O formato PNG, segundo [6] é indicado para imagens que necessitam de compressão sem perdas, é um formato bastante aplicado na *internet* e em editoração de imagens. Dentre as principais características do PNG estão:

- Formato livre;
- Permite paleta de cores;
- Permite cores com oito, 16, 32, e 48 *bits* por *pixel*;
- Permite transparência em oito e 16 *bits*;
- Permite múltiplas plataformas;

Estrutura de dados do arquivo PNG:

Os primeiros oito bits são reservados para assinatura PNG

- IHDR - Cabeçalho da imagem
- PLTE - Paleta de cores usadas na imagem
- IDAT - Chunks de dados da imagem
- IEND - Indicador de fim de dados



Figura 3.8: Estrutura do Arquivo PNG [6]

•Formato BMP

O formato *bitmap* foi desenvolvido pela *Microsoft* para aplicações no sistema Windows. É um formato de estrutura muito simples, tornando mínimas as possibilidades de erros na interpretação dos dados. Os arquivos possuem geralmente a

extensão. BMP e aceitam imagens com um, quatro, oito, 16, 24 e 32 *bits* por *pixel* [6]. A Figura 3.9 abaixo mostra a estrutura do arquivo *bitmap*.

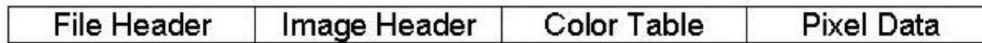


Figura 3.9: Estrutura do Arquivo Bitmap [6]

•Formato TIFF

O formato TIFF foi desenvolvido em 1986 em uma tentativa de criar um padrão para imagens geradas por equipamentos digitais. O formato é capaz de armazenar imagens em preto ou branco, escalas de cinza e em paletas de cores com 24 ou com 32 *emphbits*. O TIFF é reconhecido por praticamente todos os programas de imagem [6]. A Figura 3.10 ilustra a estrutura do arquivo TIFF.

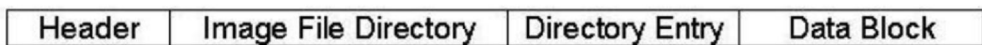


Figura 3.10: Estrutura do Arquivo TIFF [6]

3.2. Etapas para o Processamento Digital de Imagens

Esta seção apresentará os passos para o processamento digital de imagens, enfatizando as etapas abordadas neste projeto, Aquisição de imagem, Realce de imagem, Segmentação de imagem e Classificação de imagem.

3.2.1. Aquisição de Imagens

Para obtenção de imagens digitais são necessários dois elementos: dispositivos físicos captadores de imagem e digitalizador. Dispositivos físicos são sensíveis a espectros de energia eletromagnética e o digitalizador converte o sinal elétrico desses dispositivos para o formato digital. Estes elementos são chamados de sistemas de imageamento. O exemplo mais conhecido deste sistema é câmera digital (Figura 3.11), outros são *scanners* e sensores presentes em satélites. Detalhes sobre aquisição de imagem podem ser vistos em [7].



Figura 3.11: Câmera Digital

3.2.2. Realce de Imagem

O melhoramento de imagem é obtido através de técnicas, tais como o melhoramento de contraste e filtragem aplicadas, com finalidades específicas enfatizando características de interesse ou recuperando imagens que sofreram algum tipo de degradação devido à introdução de ruído, perda de contraste ou borrimento.

A aplicação dessas técnicas, designadas como realce de imagem, são transformações radiométricas que modificam o valor dos níveis de cinza dos pontos da imagem.

•Melhoramento de Contraste

Melhoramento de contraste busca melhorar a qualidade visual da imagem através da manipulação dos níveis de cinza. Uma imagem possui valores de intensidade de *pixel*, variando de 0 a 255. Quanto mais espalhados os *pixels* da imagem neste intervalo melhor é o seu contraste. Para [3] contraste consiste numa diferença local de luminância e pode ser definido como a razão dos níveis de cinza médios do objeto e do fundo.

O processo de melhoramento de contraste transforma a escala de cinza de forma pontual, ou seja, o novo valor do ponto depende somente do valor original do ponto. Uma função de transferência mapeia o valor de um ponto para um novo valor. Essa função é definida da seguinte forma: $g(x, y) = T(f(x, y))$ onde, $f(x, y)$ é o valor do nível de cinza original, T é a função de transferência e $g(x, y)$ é o novo valor do ponto.

Uma boa forma de avaliar o contraste de uma imagem é analisar seu histograma. O histograma é um gráfico que representa a distribuição dos *pixels* para cada nível de

cinza da imagem. No eixo horizontal fica a escala de cinza e no eixo vertical fica a quantidade de *pixels*.

A (Figura 3.12) ilustra duas imagens com seus respectivos histogramas, note que se trata da mesma imagem. Observando o histograma da primeira imagem pode-se notar que a distribuição dos *pixels* concentra-se próximo ao nível zero, exemplificando uma imagem com baixo contraste. A segunda imagem possui alto contraste, seus *pixels* estão melhor distribuídos no histograma, possibilitando um melhor discernimento das informações contidas na imagem.

Funções de transferências modificam histograma para obter uma imagem com melhor contraste. As funções a seguir foram retiradas de [8].

- Função Linear $\Rightarrow g(x, y) = a * f(x, y) + b$, onde os parâmetros a e b são valores de ajuste da função que variam de 0 a 255 (Figura 3.13). Esta função é utilizada para redistribuir os *pixels* da imagem de forma linear.

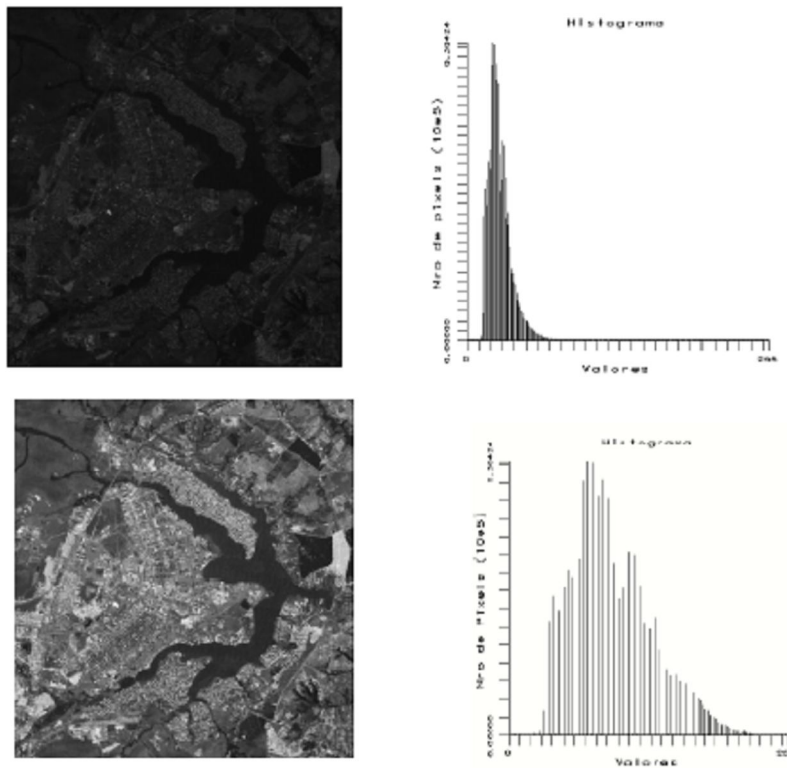


Figura 3.12: Histograma de Imagens [8]

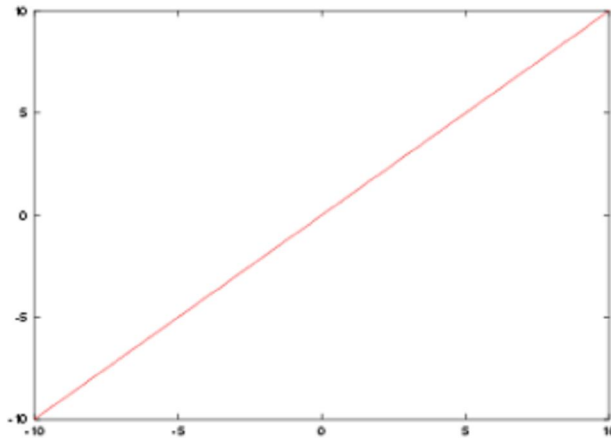


Figura 3.13: Função Linear

- Raiz Quadrada => $g(x, y) = a * [f(x, y)]^{1/2}$, onde o parâmetro a é o fator de ajuste da função que varia de 0 a 255. Esta função é utilizada para realçar as áreas escuras da imagem. Este comportamento é facilmente verificado no gráfico da Figura 3.14, note que a inclinação da função é maior próximo a áreas escuras da imagem;

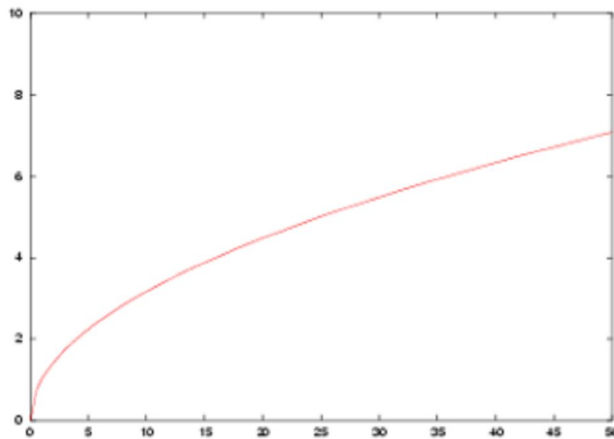


Figura 3.14: Função Raiz Quadrada

- Logaritima => $g(x, y) = a * \log(f(x, y) + 1)$, onde o parâmetro a é o fator de ajuste da função que varia de 0 a 255. Esta função tem um comportamento semelhante ao da raiz quadrada, porém realçando um intervalo menor de áreas escuras (Figura 3.15);

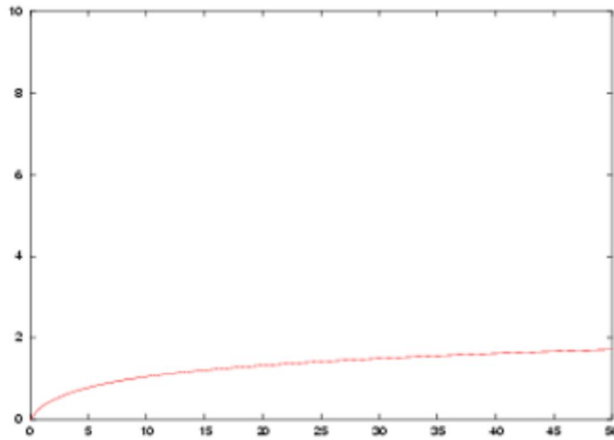


Figura 3.15: Função Logarítmica

- Inversa $\Rightarrow g(x, y) = -(a \cdot f(x, y) + b)$, onde os parâmetros a e b são valores de ajuste da função que variam de 0 a 255. Esta função inverte os níveis de cinza da imagem, ou seja, valores de pixels pretos são mapeados para brancos, valores brancos são mapeados para pretos e valores intermediários são mapeados para valores intermediários (Figura 3.16);

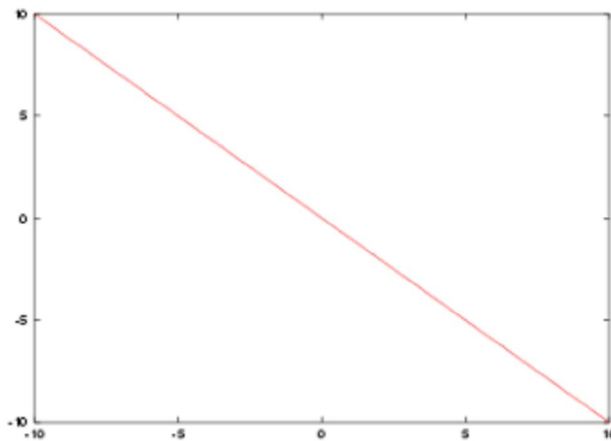


Figura 3.16: Função Inversa

•Filtragem

O processo de filtragem procura extrair informações como as bordas da imagem ou corrigir algumas degradações na imagem, tais como: borrões, ruídos inseridos pelo processo de imageamento ou na transmissão da imagem. Os filtros são divididos em duas categorias: filtros no domínio do espaço e filtros no domínio da frequência. Detalhes sobre filtragem no domínio da frequência em [7].

A filtragem no espaço é considerada uma operação local, ou seja, o nível de cinza de um ponto depende do original e de sua vizinhança. O princípio de funcionamento de tal filtro está baseado em máscaras de deslocamento as quais são matrizes com pesos associados em cada posição. A máscara com centro na posição (x, y) é colocada sobre o *pixel* a ser modificado na imagem. O *pixel* correspondente na imagem é substituído por um valor que considera os *pixels* vizinhos e os pesos correspondentes na máscara. A soma de todos os produtos dos pesos da máscara pelos *pixels* correspondente na imagem resulta em um novo valor de cinza que substituirá o *pixel* central.

•Filtros Detectores de Bordas

Segundo [3], a borda é caracterizada por mudanças abruptas de descontinuidade na imagem. Através do processo de detecção de bordas pode-se localizar e distinguir os objetos presentes na imagem, bem como descobrir algumas de suas propriedades, tais como forma e tamanho.

Os detectores de bordas se baseiam na idéia de derivadas. A derivada mede a taxa de variação de uma função. Em imagens esta variação é maior nas bordas e menor em áreas constantes. Ao percorrer a imagem marcando os pontos onde a derivada possui uma variação maior, ou seja, pontos de máximo, no fim todas as bordas terão sido marcadas [9].

Como as imagens são bidimensionais, usam-se derivadas parciais nas direções vertical y e horizontal x, que são representadas por um vetor gradiente.

- Operador gradiente de Roberts

O operador gradiente de Roberts utiliza uma matriz (2 X 2), nas direções horizontal Gx e vertical Gy [9].

$$\begin{array}{cc}
 \begin{array}{|c|c|}
 \hline
 0 & 1 \\
 \hline
 -1 & 0 \\
 \hline
 \end{array} &
 \begin{array}{|c|c|}
 \hline
 1 & 0 \\
 \hline
 0 & -1 \\
 \hline
 \end{array} \\
 \mathbf{G_x} & \mathbf{G_y}
 \end{array}$$

Figura 3.17: Mascara de Roberts

O gradiente é calculado pela seguinte fórmula: $GR = [G(x \ 1/2) + G(y \ 1/2)] \ 1/2$

• Operador gradiente de Sobel

O operador gradiente de Sobel utiliza matriz (3 X 3) [9].

1	2	1
0	0	0
-1	-2	-1

Gx

-1	0	1
-2	0	2
-1	0	1

Gy

Figura 3.18: Mascara de Sobel

O gradiente é calculado pela seguinte fórmula: $GR = [G(x \ 1/2) + G(y \ 1/2)] \ 1/2$

•Filtros de Ruídos

Para [9] os ruídos são um conjunto de *pixels* aleatórios diferentes dos dados da imagem.

• Filtro da Média

O filtro da média utiliza uma máscara que percorre a imagem e substitui cada *pixel* da imagem pela média de seus vizinhos. O objetivo desse filtro não é de eliminar o ruído e sim suavizá-lo. A máscara utilizada é uma matriz (3 X 3) baseada na vizinhança 8 [10].

$$\frac{1}{9} * \begin{matrix} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \end{matrix}$$

Figura 3.19: Mascara da Média

• Filtro da Mediana

O objetivo do filtro da mediana é substituir o *pixel* central de uma matriz, geralmente 3 X 3, pelo valor que estiver na mediana desses *pixels* ordenados em ordem crescente de nível de cinza [10].

Este filtro pode ser facilmente implementado da seguinte forma:

- 1) Para todos os pontos da imagem;
- 2) Escolha um ponto e o atribua a x;

2.1) Coloque em ordem crescente x e sua vizinhança 8, guarde esses valores em um conjunto X ;

2.2) O novo valor do ponto é o valor que está na mediana do conjunto X ;

Este filtro baseia-se no fato de que se o ponto for um ruído, ao colocar em ordem crescente este ponto e sua vizinhança, o ruído tende a ficar em um dos extremos do conjunto X , sendo substituído pelo valor mediano do conjunto X .

3.2.3. Segmentação de Imagens

Por segmentação de uma imagem entende-se a extração ou identificação dos objetos contidos na imagem, onde o objeto é toda característica com conteúdo semântico relevante para a aplicação desejada [3]. A segmentação faz parte de um passo maior que é a classificação da imagem. A etapa de segmentação divide a imagem em regiões, sem considerar o processo de classificação. Para [3], a região em uma imagem é um conjunto de pontos ligados, ou seja, de qualquer ponto da região pode-se chegar a qualquer outro ponto por um caminho inteiramente contido na região, baseado na vizinhança 4 ou vizinhança 8.

Segmentação Baseado em Crescimento de Regiões

A segmentação baseada no crescimento de regiões pode ser feito de duas formas: por varredura ou por busca em todas as direções. Ambos os métodos baseiam-se no fato das regiões a serem detectadas apresentarem propriedades locais aproximadamente constantes como, por exemplo, o nível de cinza.

• Crescimento de regiões por busca em todas as direções.

No crescimento por busca, em todas as direções, uma única região é crescida por vez.

Algoritmo:

Inicialmente $região(p) = 0$ para todo o ponto p da imagem.

1) $k = 0$;

2) escolha um ponto x , tal que $região(x) == 0$, se não existe ponto nesta condição, então pare: fim;

3) inicie nova região: $k = k + 1$; $região(x) = k$;

4) ache todos os pontos p da vizinhança de x , tal que $\text{região}(p) == 0$ e p possa ser adicionado à região de x sem violar o critério de homogeneidade: faça $\text{região}(p) = \text{região}(x)$ e guarde estes pontos em um conjunto X ;

5) se X está vazio, vá para 2; caso contrário, escolha e extraia um ponto x de X ;

6) vá para 4;

Onde:

x é o ponto (x, y) ;

Os vizinhos de P é a sua vizinhança 4; região (p) indica a região do ponto p .

• **Crescimento de regiões segundo a varredura.**

Neste método, a imagem é percorrida da esquerda para direita e de cima para baixo comparando um ponto inicial da imagem (x, y) com todos os outros pontos da imagem, adicionando ou não esses pontos a região do ponto inicial.

Algoritmo:

1) Percorra a imagem segundo a varredura;

2) Escolha um ponto x na imagem e o rotule;

3) Compare x com todos os outros pontos da imagem;

3.1) Adicione a região de x os pontos que não violarem o critério de homogeneidade;

3.2.4. Classificação de Imagens

A classificação é o processo de extração de informações da imagem. Consiste na divisão da imagem em classes, ou seja, segmentação e posterior identificação destas classes. Esta pode ser feita de forma não automática, onde um especialista humano extrai informações baseando-se na inspeção visual da imagem ou de forma automática, feita por computador [8]. A classificação de imagem por computador pode ser realizada usando técnicas de redes neurais. A rede é treinada de forma que ela possa identificar e agrupar todos os *pixels* em classes. O aprendizado da rede pode ser feito de duas formas: aprendizado supervisionado e aprendizado não supervisionado.

• **Aprendizado supervisionado**

Para [3] quando existem amostras disponíveis de classificação conhecida, define-se que o problema de aprendizado é supervisionado. A rede, tendo esta amostra conhecida inicialmente, tenta encontrar na imagem todos os pixels que podem ser

agrupados a esta amostra conhecida, formando uma classe. Os pixels podem ser agrupados a uma mesma classe se por um critério de similaridade eles são iguais ou semelhantes. Este critério é chamado de limiar, podendo ser a intensidade luminosa do pixel usada para decidir a igualdade ou semelhança.

- Aprendizado não supervisionado

No aprendizado sem supervisão as amostras na área de treinamento não são rotuladas [3]. Neste caso a rede tem que aprender sozinha quais pixels podem fazer parte de uma mesma classe, sem nenhum conhecimento inicial.

3.3. Processamento Digital de Imagem e o Projeto

No projeto desenvolvido existem dois momentos distintos nos quais as técnicas de processamentos de imagem são utilizadas, são eles: Detecção da queda da primeira gota e acompanhamento do nível do líquido no interior do recipiente.

As etapas para o processamento da imagem nos dois momentos descritos acima apresentam semelhanças. São elas:

1. Aquisição da imagem por intermédio das câmeras de monitoramento.
2. Conversão das imagens de coloridas (**RGB**) para *tons de cinza*.
3. Conversão das imagens de tons de cinza para preto e branco (*processo de binarização*).
4. Comparação (subtração pixel a pixel) da imagem capturada com a imagem salva no momento em que o sistema foi calibrado. A calibragem é feita manualmente determinando a região da imagem a ser monitorada, ou seja, faz-se a marcação da região retangular com auxílio das interfaces de calibragem do sistema (Seção 2.4.5.1.).
5. Contagem da quantidade de valores que após a subtração ficaram acima de um limiar estipulado pelo operador do sistema.
6. Comparação percentual entre a quantidade acima descrita e o total de pixels no interior da região que está sendo monitorada; estando este percentual acima de um valor pré-determinado considera-se que a gota ou o nível de líquido foram detectados.

3.3.1. Detecção do Menisco

As etapas para a detecção do nível de líquido seguem como as descritas no (Seção 3.3.), contudo a região retangular descrita naquele item é a região utilizada para a marcação do contorno da garrafa (Seção 2.4.5.1.).

Para encontrar o nível do líquido temos uma outra região retangular menor, que também pode ser marcada manualmente. Este retângulo percorre as imagens adquiridas no sentido de cima para baixo sempre na parte central das mesmas; a comparação é feita com a primeira imagem capturada no início do processo de monitoramento do nível de líquido, para que este processo ocorra é necessário que o sistema execute as 6 etapas descritas na Seção 3.3.

As imagens que foram salvas no ato de calibragem correspondem a variações de 20 em 20 ml, e para cada variação de volume temos uma variação em altura (em pixel) em relação ao topo da garrafa, estes dados, volume total e altura em pixel, ficam salvos em um vetor de dados.

O algoritmo seguido pelo retângulo de varredura de imagem salva os deslocamentos (em pixel) sofridos pelo retângulo de varredura em um outro vetor, ao encontrar o nível; o algoritmo faz uma busca no vetor de dados para saber se o deslocamento sofrido pelo retângulo de busca, ou melhor, se a distância em pixels à borda superior do recipiente corresponde a um dos valores já salvos no vetor de dados.

Estando os valores próximos e dentro de uma faixa de tolerância o algoritmo informa como volume encontrado, o valor que consta no vetor de dados. Valores intermediários de volume também são informados visto que o incremento de volume e o incremento em pixel não seguem uma relação linear.

3.3.2. Etapas do Projeto

3.3.2.1. Aquisição da Imagem Real

As Figuras (3.20 e 3.21) correspondem às imagens recebidas das câmeras que compõem o sistema de monitoramento. Elas ainda não sofreram processamento, ou melhor, são as imagens de entrada da etapa de processamento.



Figura 3.20: Bico Real



Figura 3.21: Garrafa Real

3.3.2.2. Imagem em Níveis de Cinza

As Figuras (3.22 e 3.23) correspondem as Figuras (3.20 e 3.21) após a primeira etapa de processamento. Podemos ver que as imagens agora estão em tons de cinza e não mais coloridas como anteriormente.

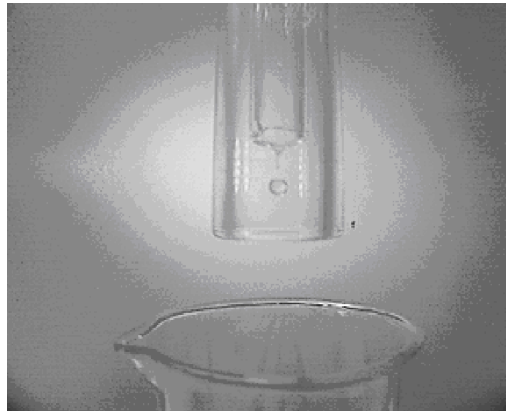


Figura 3.22: Bico em Tons de Cinza

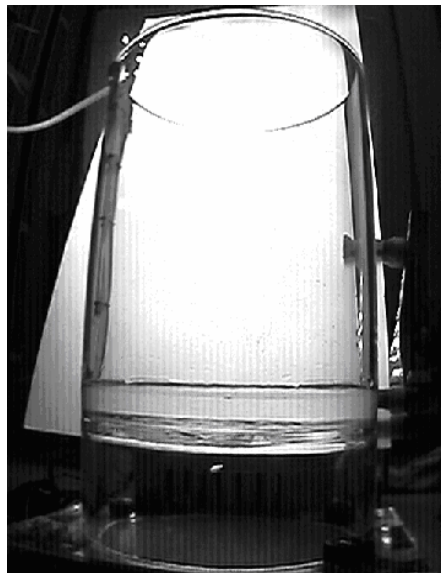


Figura 3.23: Garrafa em Tons de Cinza

3.3.2.3. Imagem Binarizada (Limiarização ou Thresholding)

As Figuras (3.24 e 2.25) correspondem às Figuras (3.20 e 3.21) após a última etapa de pré-processamento. Podemos ver que elas apresentam somente dois níveis de cinza (o branco e o preto); é sobre essas imagens que os algoritmos de detecção de queda de primeira gota bem com o de detecção de nível serão rodados.

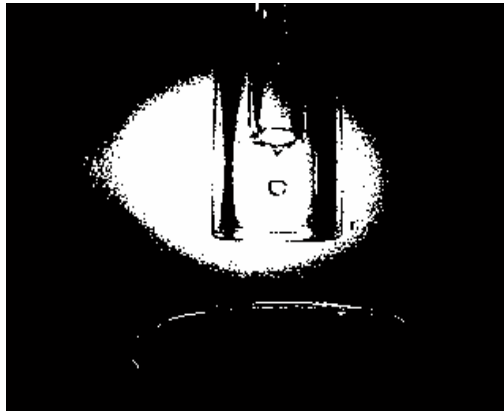


Figura 3.24: Bico Binarizado



Figura 3.25: Garrafa Binarizada

Capítulo 4

Elaboração dos circuitos

O projeto de montagem do painel de “leds” partiu da necessidade de uma iluminação que fosse suficiente tanto no aspecto do brilho quanto na área de cobertura da garrafa utilizada nos experimentos.

4.1. Circuito Teórico

Os circuitos abaixo (Figura 4.1 e Figura 4.2) correspondem ao modelo teórico de circuitos que pretendia-se montar para o Painel de Leds (Figura 4.5), contudo os circuitos que foram implementados são apresentados nas Figuras (4.3 e 4.4).

Linhas Impares:

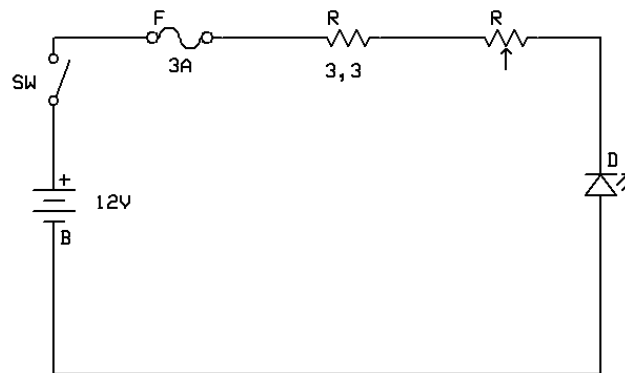


Figura 4.1: Circuito de Linhas Pares

Linhas Pares:

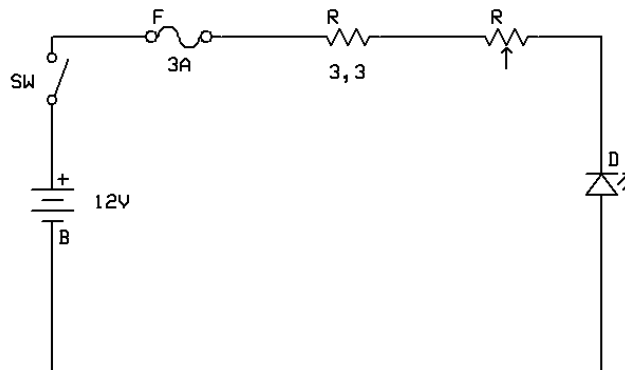


Figura 4.2: Circuito de Linhas Ímpares

Utilizando uma fonte de tensão de 12,0 V e consultando as especificações de corrente e queda de tensão nos “leds” utilizados no projeto, foi possível calcular a faixa

de valores dos resistores, levando-se em conta que são 128 “leds” para cada circuito (totalizando 256). Para isso, divide-se a queda de tensão no resistor ($12,0\text{ V} - 3,8\text{ V}$) pela corrente que passa no próprio ($128 \times i\text{ mA}$). Onde “i” é a corrente recomendada pelo fabricante para cada “led”, que varia de 20,0 mA a 30,0 mA, obtendo uma faixa de valores para os resistores.

	Nº de Leds	I/Led(A)	Itotal(A)	Vfonte(V)	Vled(V)	R(ohm)	Ppot(W)>=
Linhas Pares:	128	0,02	2,56	12,0	3,8	3,203	20,992
Linhas Ímpares:	128	0,02	2,56	12,0	3,8	3,203	20,992

Onde:

- **I/Led(A)** é a corrente em cada “led”,
- **Itotal(A)** é a corrente total no circuito,
- **Vfonte(V)** é a tensão utilizada para alimentar o circuito,
- **Vled(V)** é a queda de tensão no conjunto de 128 “leds”,
- **R(ohm)** é o valor da resistência utilizado no projeto,
- **Ppot(W)>=** é o valor da potência mínima no potenciômetro.

4.2. Circuito Prático

O circuito abaixo (Figura 4.3) foi usado para montagem das linhas pares do Painel de Leds (Figura 4.5).

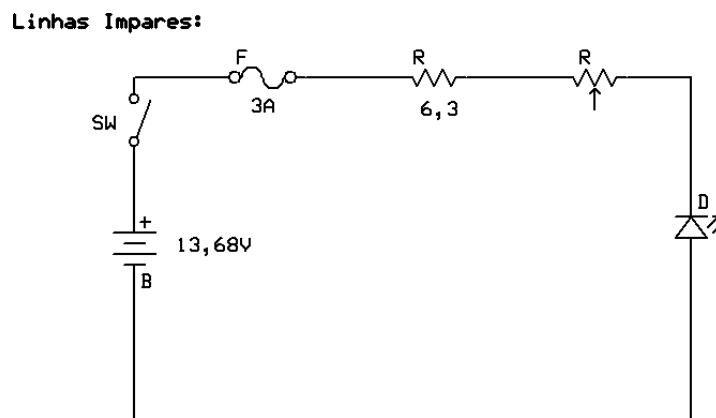


Figura 4.3: Circuito de Linhas Pares

O circuito abaixo (Figura 4.4) foi usado para montagem das linhas ímpares do Painel de Leds (Figura 4.5).

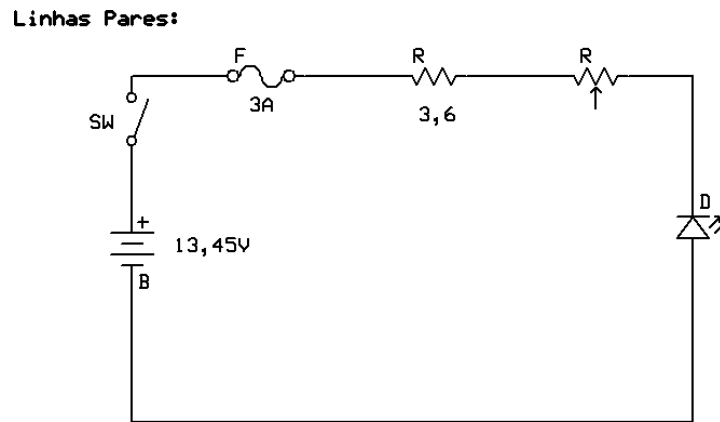


Figura 4.4: Circuito de Linhas Ímpares

Após a escolha dos resistores e a montagem do painel de “leds”, foi feita a verificação das variáveis calculadas na parte teórica ($I_{Led}(A)$, $I_{total}(A)$, $V_{fonte}(V)$, $V_{led}(V)$, $P_{pot}(W) \geq$) utilizando multímetro. Os resultados obtidos estão na tabela abaixo:

	Nº de Leds	$I_{Led}(mA)$	$I_{total}(mA)$	$V_{fonte}(V)$	$V_{led}(V)$	$R(ohm)$	$P_{pot}(W) \geq$
Linhas Pares:	128	0,020	2,583	13,45	4,15	3,6	24,025
Linhas Ímpares:	128	0,012	1,507	13,68	4,18	6,3	14,325

Observação: devido à dificuldade de obter resistores nesses valores que dissipem esta potência, foram utilizados resistores diferentes para os circuitos.

A figura abaixo (Figura 4.5) mostra a imagem do Painel de Leds construído em laboratório, no intuito de gerar uma iluminação mais contínua, facilitando assim o processamento das imagens do nível de líquido no recipiente.

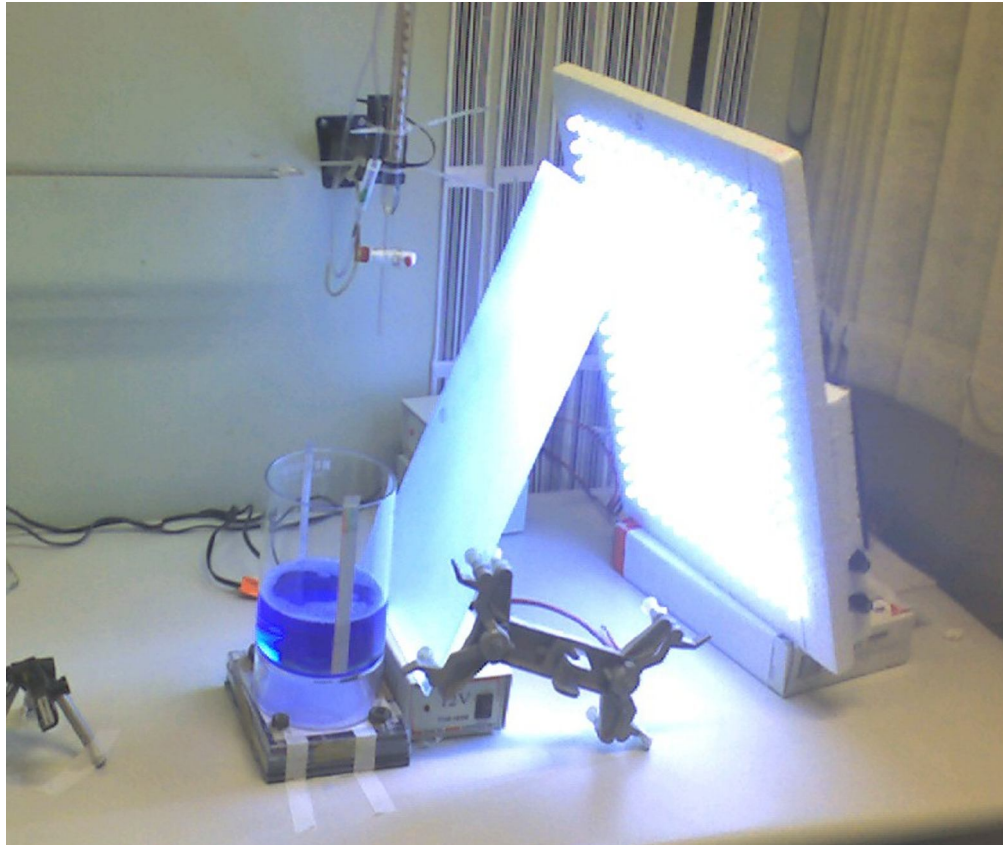


Figura 4.5: Painel de Leds

A figura abaixo (Figura 4.6) corresponde ao Módulo Adaptador de Tensão montado em laboratório, este módulo visa o casamento do nível lógico da porta paralela do computador com a entrada do Circuito Lógico Programável (CLP).



Figura 4.6: Circuito Adaptador de Tensão Vista Frontal

A figura abaixo (Figura 4.7) corresponde ao Circuito Adaptador de Tensão visto pela parte traseira, indicando a entrada de alimentação do mesmo.

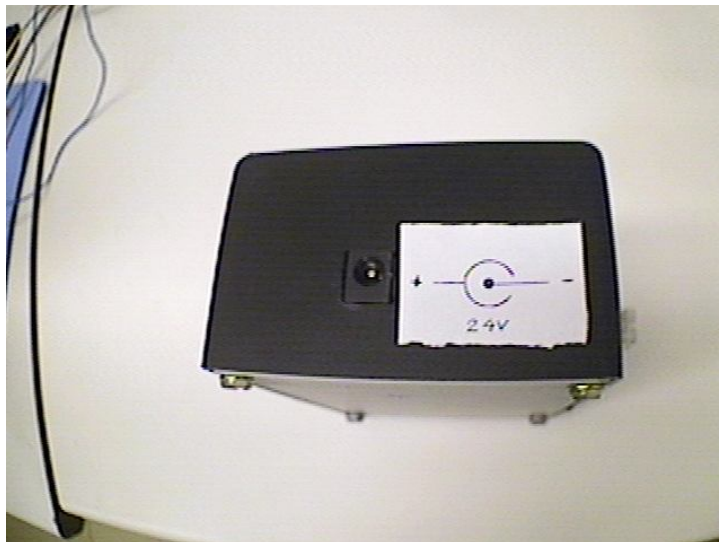


Figura 4.7: Circuito Adaptador de Tensão Vista Traseira

Capítulo 5

Conclusão

Através do desenvolvimento do sistema de monitoramento da destilação fracionada do petróleo foi possível aprender sobre as técnicas de processamento digital de imagens. Apliquei o paradigma de desenvolvimento de projetos baseados em orientação a objetos. Ao longo desse projeto pude perceber a importância da etapa de planejamento e estudos antes de tentar efetivamente a implementação.

O processamento digital de imagens mostrou-se uma poderosa ferramenta no desenvolvimento de sistemas de monitoramento, pois é de baixo custo comparado com tecnologias existentes para tal monitoramento. Tive como aspectos críticos a dependência das variáveis de ambiente tais como: iluminação, distância da câmera à unidade de destilação, perturbações mecânicas do hardware e ou da superfície do líquido no interior do recipiente.

Por fim o projeto proporcionou uma primeira experiência como engenheiro, pois ao longo do desenvolvimento do mesmo foi possível exercitar as etapas de planejamento, pesquisa, aprendizagem e implementação. Sempre que me deparava com problemas desconhecidos buscava meios de solucioná-los; revi o planejamento quando necessário e descobri o quanto é difícil, porém gratificante, o desenvolvimento de trabalhos envolvendo pesquisa.

Apêndice A

A.1. Diagramas dos Circuitos

A Figura A.1 corresponde ao esquemático do circuito montado em laboratório com o objetivo de implementar o Circuito Adaptador de Tensão (CAT), sendo este responsável pelo casamento do nível lógico da porta paralela do computador com as entradas do Circuito Lógico Programável (CLP).

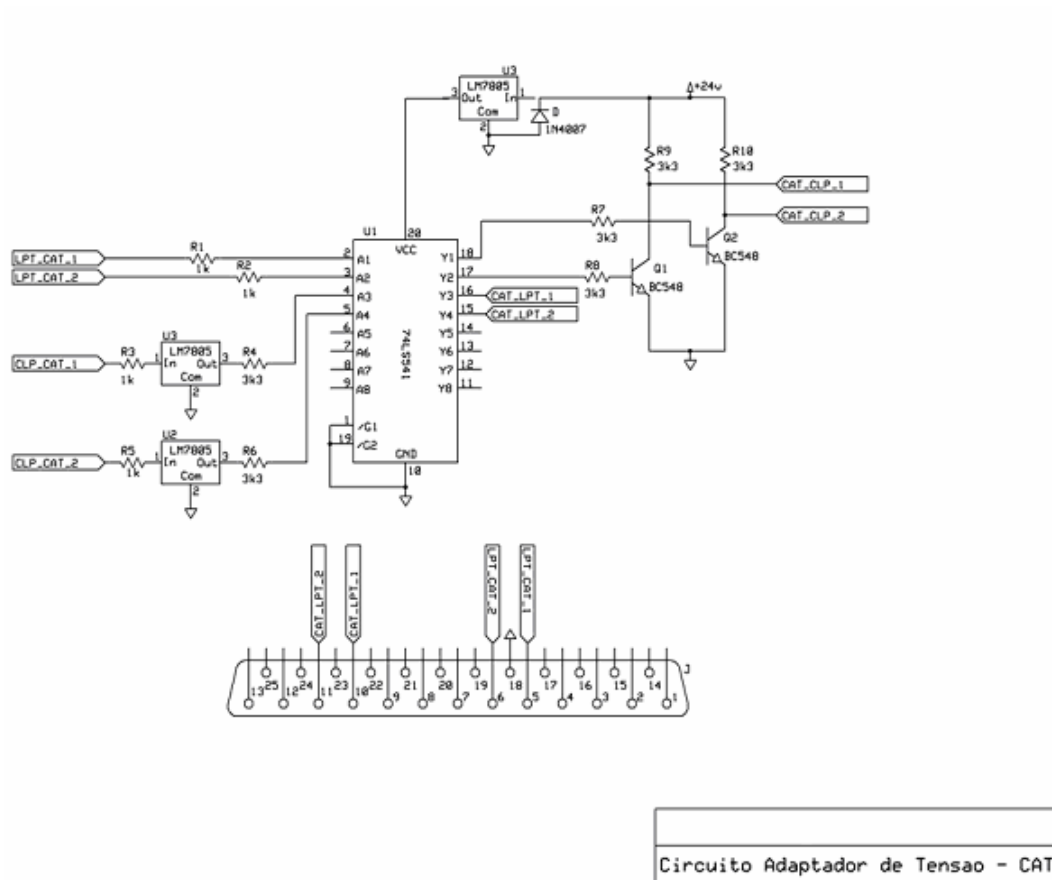


Figura A.1: Diagrama Esquemático do CAT

A Figura A.2 corresponde ao mesmo esquemático apresentado na Figura A.1, contudo este esquemático foi montado com software Orcad , próprio para simulação de sistemas, onde foram feitas as devidas simulações antes da montagem efetiva do protótipo.

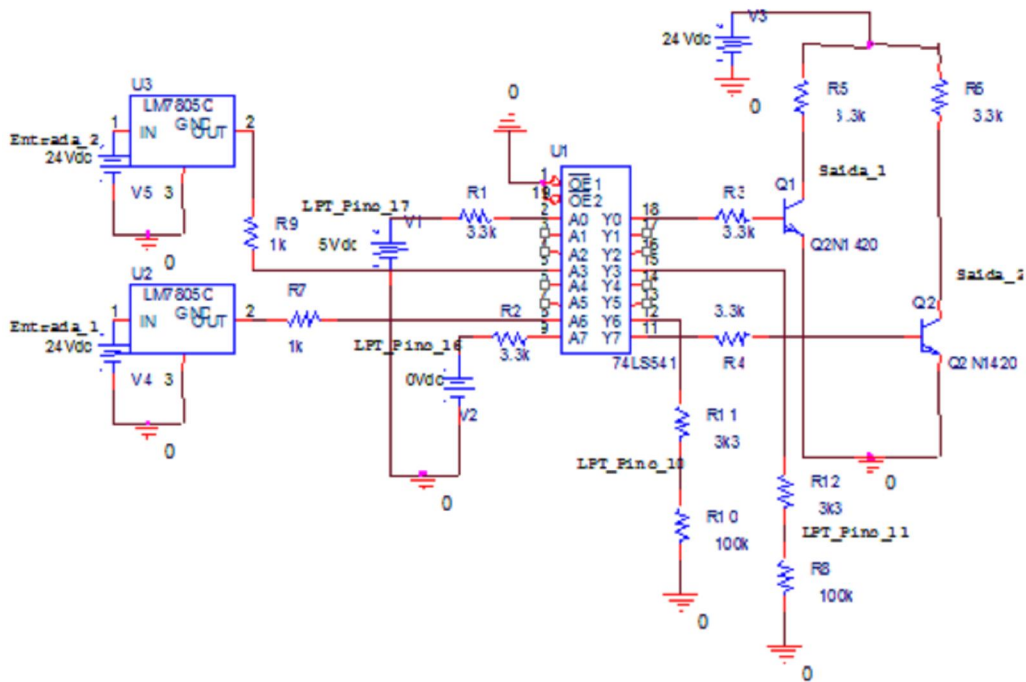


Figura A.2: CAT Simulado no ORCAD

A.2. Diagrama do Circuito de Teste da LPT

A Figura A.3 corresponde ao diagrama esquemático do circuito de teste da LPT, o objetivo de mesmo é como o próprio nome diz realizar o teste dos pinos da porta paralela, seguindo a seqüência de saídas que o programa principal gera para os pinos da LPT.

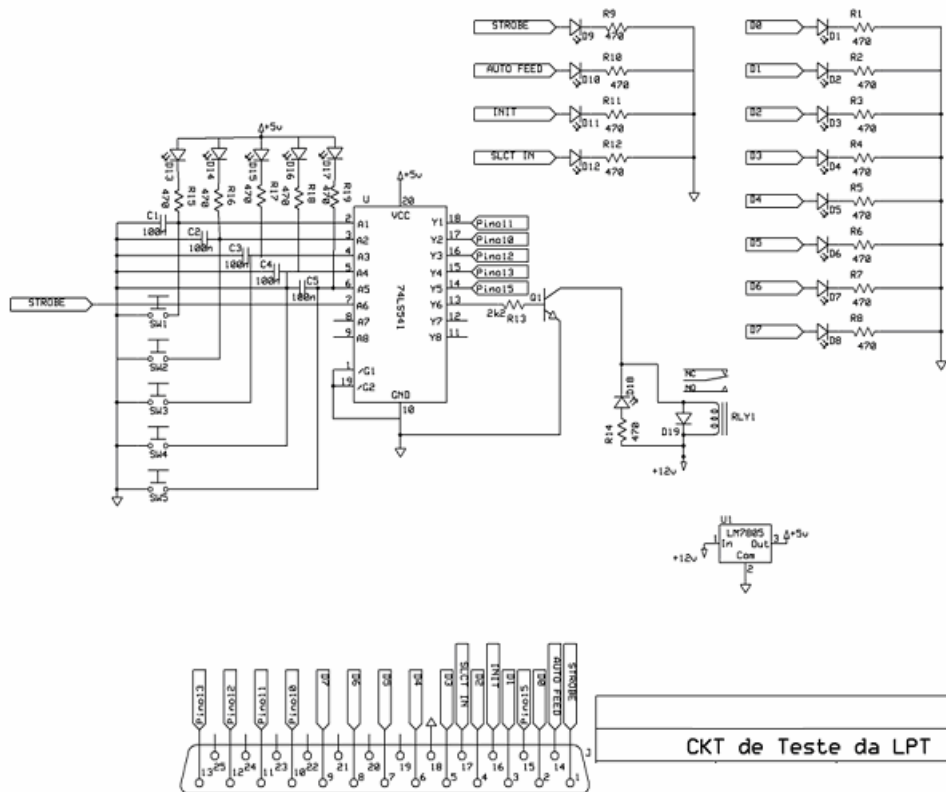


Figura A.3: Circuito de Teste

Bibliografia

- [1] PACHECO, F. R. *Monitoramento da Destilação Fracionada do Petróleo Via Processamento Digital de Imagens*, Universidade Federal do Rio de Janeiro – UFRJ. Março de 2009.
- [2] BANON, G. J. F. *Formal Introduction to digital image processing*. INPE, São José dos Campos JULHO 2000.
- [3] MASCARENHAS, N. A. & VELASCO, F. R.D. *Processamento Digital de Imagens*. Ministerio da ciência e Tecnologia -MCT. Instituto de Pesquisa Espaciais - INPE. Janeiro de 1989.
- [4] BRITO, S. F. & CARVALHO, J. M. *Sistemas de Processamento digital de imagens para Fins didáticos/científicos: Estudo, seleção e implementação de algoritmos de segmentação*. Relatório de iniciação científica. Agosto de 1998.
- [5] SCHNEIDER, B. O. *Apostila de Computação Gráfica*. Departamento de Ciência da Computação. Universidade Federal de Lavras. 9 de Agosto de 2001.
- [6] RAMOS, O. T. H. *Análise Comparativa entre os principais formatos de armazenamento de imagens*. Dissertação de Mestrado pela Universidade Presbiteriana Mackenzie. São Paulo, dezembro de 2000.
- [7] GONZALES, R. C. & WOODS, R. E. *Digital Image Processing*. University of Tennessee Perceptics Corporation, 1992.
- [8] MARIA, L. G. F. *Processamento Digital de Imagens*. INPE, Junho de 2000.
- [9] MARTA, D. S. *Algoritmos para detecção de bordas*. UFSC - dezembro de 1998.
- [10] FACON, J. *Princípios Básicos da Visão por Computador e do Processamento de Imagens*. Pontificia Universidade Católica do Paraná. Curso e Mestrado em Informatica Aplicada, 2002.