

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTOS ACADÊMICOS DE ELETRÔNICA E MECÂNICA
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

BRUNO VIEIRA DIAS DA SILVA
MILENE IDALGO BRAGA

**PROTÓTIPO DE SISTEMA AUTOMATIZADO DE INSPEÇÃO VISUAL
DA TURBIDEZ DA ÁGUA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2016

BRUNO VIEIRA DIAS DA SILVA
MILENE IDALGO BRAGA

PROTÓTIPO DE SISTEMA AUTOMATIZADO DE INSPEÇÃO VISUAL DA TURBIDEZ DA ÁGUA

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso Superior de Tecnologia em Mecatrônica Industrial, dos Departamentos Acadêmicos de Eletrônica e Mecânica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Me. Sidney Carlos Gasoto

CURITIBA
2016

TERMO DE APROVAÇÃO

BRUNO VIEIRA DIAS DA SILVA
MILENE IDALGO BRAGA

PROTÓTIPO DE SISTEMA AUTOMATIZADO DE INSPEÇÃO VISUAL DA TURBIDEZ ÁGUA

Este trabalho de conclusão de curso foi apresentado no dia 23 de novembro de 2016, como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Milton Luiz Polli
Coordenador de Curso
Departamento Acadêmico de Mecânica

Prof. Me. Sérgio Moribe
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Ubiradir Mendes Pinto
UTFPR

Prof. Dr. Luiz Carlos de Abreu Rodrigues
UTFPR

Prof. Me. Sidney Carlos Gasoto
Orientador - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

À minha querida mãe, Saionara Romariz Idalgo, que sempre me apoiou e incentivou nessa caminhada de superação e que me inspirou a cada dia com sua garra e determinação.

Aos meus queridos pais Edir e Jeany, meus avós, irmã, e a toda a família.

AGRADECIMENTO

Ao professor Sidney Carlos Gasoto pela paciência e solicitude em orientar o desenvolvimento deste projeto.

RESUMO

BRAGA, Milene Idalgo; DA SILVA, Bruno Vieira Dias. **Protótipo de Sistema Automatizado de Inspeção Visual da Turbidez da Água**. 2016. 77 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial), Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

A tecnologia da automação industrial alcança dia após dia níveis cada vez mais sofisticados. Naturalmente, indústrias de grande porte e sofisticadas requerem tecnologias avançadas para grandes produções. Entretanto, pequenas indústrias muitas vezes procuram por uma tecnologia de automação adequada ao seu poder de investimento. Com a ascensão dos microempreendedores da indústria artesanal cervejeira de Curitiba, o desenvolvimento de equipamentos de custo e porte adequado à inspeção da turbidez de produtos líquidos de microindústrias, é demanda digna de atenção. Neste trabalho, foi desenvolvido um sistema automático de inspeção da turbidez de líquidos que é capaz de separar produtos reprovados ou aprovados numa inspeção. Foram combinadas as técnicas da Nefelometria e da Turbidimetria para a inspeção visual. O protótipo deste sistema integra o software Vision Builder da National Instruments ao microcontrolador Arduino. O funcionamento foi satisfatório, no entanto, com confiabilidade dentro de uma faixa limitada de níveis de turbidez. Ainda assim, por ter baixo custo de aquisição, este pode ser um sistema de interesse relevante para segmentos bem específicos de mercado e até mesmo para novos estudos.

Palavras chave: Turbidez. Automatização. Nefelometria. Turbidimetria. Arduino. Vision Builder.

ABSTRACT

BRAGA, Milene Idalgo; DA SILVA, Bruno Vieira Dias. **Prototype of Automated System for Water Turbidity's Visual Inspection**. 2016. 77 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial), Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

The technology of industrial automation reaches levels day after day increasingly sophisticated. Of course, large and sophisticated industries require sophisticated technologies for large productions. However, small industries often seek for a proper automation technology to its power investment. With the rise of micro-entrepreneurs of brewing craft industry of Curitiba, the development of equipment with cost and adequate size to inspection of turbidity of liquid of small industries' products is something that is worthy demand attention. In this work, an automatic inspection system of the liquid turbidity that is able to separate a failed or approved product on an visual inspection was developed. Technicals from nephelometry and turbidimetry for visual inspection were combined. The prototype of this system integrates the Vision Builder software from National Instruments to the Arduino microcontroller. The operation was satisfactory, however, with reliability within a limited range turbidity levels. Still, for having low cost, this may be a relevant interest system for very specific market segments and even for new studies.

Keywords: Turbidity. Automation. Nephelometry. Turbidimetry. Arduino. Vision Builder

LISTA DE FIGURAS

Figura 1 – Turbidímetro de Mesa ou de Laboratório	12
Figura 2 – Turbidímetro portátil	12
Figura 3 – Turbidímetro primitivo.....	13
Figura 4 – Turbidímetro Industrial	13
Figura 5 – Esquema do Sistema Nefelométrico e Turbidímetro	16
Figura 6 – Exemplos de níveis de turbidez da água em UNT's.....	18
Figura 7 – Carrossel Seletor.....	20
Figura 8 – Caixa Controladora	22
Figura 9 – Porta USB tipo B <i>standard</i> e entrada Jack.....	23
Figura 10 – Saída para cabo flat	23
Figura 11 – Placa Microcontroladora e Placa de Potência	24
Figura 12 – Webcam Logitech HD C270 acoplada ao sistema	25
Figura 13 – Webcam Logitech HD C270.....	25
Figura 14 – Circuitos de iluminação	29
Figura 15 – Posicionamento da câmera e das fontes de iluminação	30
Figura 16 – LED's a 90° com filete de placa de acrílico	31
Figura 17 – Em foco, LED's a 180° com placa acrílico à frente.....	32
Figura 18 – Circuito de potência do motor de passo	34
Figura 19 – Circuito de ligação do LCD.....	40
Figura 20 – Portas seriais universais (USB) tipo A <i>standard</i>	42
Figura 21 – Configuração de espera de <i>string</i> pela serial no VBAI	43
Figura 22 – Configuração de envio de mensagem pela serial no VBAI	44
Figura 23 – Primeira parte do programa de inspeção do VBAI	45
Figura 24 – Segunda parte do programa de inspeção do VBAI	46
Figura 25 – Mensagem pelo LCD. Seq.1	47
Figura 26 – Mensagem pelo LCD. Seq.2	48
Figura 27 – Mensagem pelo LCD. Seq.3	48
Figura 28 – Mensagem pelo LCD. Seq.4	49
Figura 29 – Mensagem pelo LCD. Seq.5	50
Figura 30 – Mensagem pelo LCD. Seq.6	50

LISTA DE TABELAS

Tabela 1 – Proporção da diluição de café em água	28
Tabela 2 – Medições antes e depois da implementação de difrator de luz	33
Tabela 3 – Valores obtidos em três sequências de inspeção	52
Tabela 4 – Média aritmética dos valores obtidos em três sequências de inspeção ..	53
Tabela 5 – Desvio padrão e erro de precisão em três sequências de inspeção com iluminação a 90°	54
Tabela 6 – Desvio padrão e erro de precisão em três sequências de inspeção com iluminação a 180°	55

SUMÁRIO

1	INTRODUÇÃO	10
1.1	PROBLEMA	10
1.2	OBJETIVOS	11
1.2.1	Objetivo geral	11
1.2.2	Objetivos específicos	11
1.3	JUSTIFICATIVA	12
1.4	PROCEDIMENTOS METODOLÓGICOS	14
1.5	ESTRUTURA DO TRABALHO	15
2	FUNDAMENTAÇÃO TEÓRICA	16
3	DESENVOLVIMENTO DO PROTÓTIPO	20
3.1	COMPONENTES DO SISTEMA	20
3.1.1	Componentes físicos do sistema	20
3.1.1.1	Carrossel Seletor	20
3.1.1.2	Caixa Controladora	22
3.1.1.3	Câmera	24
3.1.1.4	Computador	26
3.1.2	Componentes virtuais do sistema	26
3.2	ETAPAS DO DESENVOLVIMENTO E CONSTRUÇÃO DO SISTEMA	26
3.2.1	Conexão da câmera com o Vision Builder	26
3.2.2	Primeiros testes de inspeção da turbidez	27
3.2.3	Iluminação e posicionamento da câmera	28
3.2.4	Automatização do transporte dos frascos	33
3.2.4.1	Circuito de potência do motor de passo	34
3.2.4.2	Rotina de programação para o motor de passo	35
3.2.5	A comunicação dos periféricos teclado e LCD com o Arduino	39
3.2.5.1	LCD	39
3.2.5.2	Teclado	41
3.2.6	Comunicação Arduino / Vision Builder	41
3.2.7	Programa final de inspeção do Vision Builder	44
3.2.8	Programação final do Arduino	47
3.2.9	Menu de interface homem máquina	47
4	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	52
5	CONSIDERAÇÕES FINAIS	57
	REFERÊNCIAS	59
	APÊNDICE - TRANSCRIÇÃO INTEGRAL DO PROGRAMA EM LINGUAGEM C++ FEITO PARA O MICROCONTROLADOR ARDUINO	60

1 INTRODUÇÃO

A turbidez ou turvação de um fluido orgânico ou inorgânico é a propriedade física inversamente proporcional à sua transparência. A associação de micro organismos ou partículas, microscópicas ou não, em suspensão num líquido e interferindo diretamente na quantidade de luminosidade que o atravessa gera a turbidez do fluido. É um indicador de qualidade de grande relevância, por exemplo, para a indústria vinícola, cervejeira, farmacêutica, e a outras várias áreas industriais, bem como para a indústria laboratorial e de análises clínicas. Não há como deixar de observar que a cervejaria artesanal em Curitiba erigiu, nos últimos anos, como nunca antes. A ascensão desse ramo fabril na cidade, a está levando a ser reconhecida nacional e internacionalmente como um polo da indústria artesanal cervejeira. Inúmeros já são os microempreendedores em Curitiba que estão embarcando nesse promissor nicho de mercado. Muitos começam esse sonho fabricando dentro da própria casa, numa chácara ou numa garagem. Demandam por máquinas de fabricação de cerveja e de controle da produção e da qualidade. Por tudo isso, a aferição da turbidez da água - matéria prima das microcervejarias - é digna de atenção, e especialmente enfocada para o microempreendedor. Tal é o tema a ser explorado neste trabalho, cujo projeto é a construção de um protótipo de sistema automatizado de inspeção visual da turbidez água.

1.1 PROBLEMA

Diante da importância da aferição da turbidez em processos industriais, levantou-se a seguinte questão: como desenvolver um protótipo de sistema de aferição da turbidez da água para rotar ou classificar uma produção industrial em série em função dessas características?

Uma hipótese ou pressuposto seria utilizar o software Vision Builder da National Instruments que, configurado para essa finalidade e associado a uma câmera de visão, ou webcam, captaria e analisaria as imagens de frascos em série contendo os líquidos controlados. Após, uma placa microcontroladora Arduino,

através dos sistemas de transporte, rotearia os frascos a um destino programado em função da turbidez aferida. O transporte poderia ser baseado num sistema suspenso do tipo *suspended conveyor system*.

1.2 OBJETIVOS

Nesta seção são apresentados os objetivos geral e específicos do trabalho, relativos ao problema anteriormente apresentado.

1.2.1 Objetivo geral

Desenvolver um sistema automático em série de inspeção visual da turbidez da água em frascos de ensaio, e roteá-los em função desta característica.

1.2.2 Objetivos específicos

- Conhecer o funcionamento dos softwares que serão utilizados na metodologia, e implementar programações;
- Caracterizar os princípios de funcionamento dos circuitos de potência e seus motores de passo;
- Descrever os princípios de medição da turbidez da água;
- Analisar métodos de aquisição de imagens com boa qualidade no que tange aos contrastes que possibilitem a identificação do nível de turbidez da água;
- Identificar a forma mais viável de integrar a comunicação entre si dos aparelhos que comporão o sistema;
- Avaliar a relação custo/benefício da implantação de um sistema como esse em função da demanda de uma linha de produção em série.

1.3 JUSTIFICATIVA

Existem no mercado diversos equipamentos que realizam a aferição da turbidez da água, chamados de turbidímetros. Os portáteis são em geral mais baratos que os industriais, que tendem a ser mais complexos. Portanto, para uma produção em série, um turbidímetro portátil pode não ser o mais recomendável por demandar um operador para manuseá-lo e um tempo de ciclo de análise muito superior ao industrial. No entanto, para produções de pequeno e médio porte, ou até mesmo para laboratórios de análises clínicas, dependendo da demanda, um turbidímetro automático em série de baixo custo pode ser a alternativa mais viável.

Naoum (2015), em seus Métodos de Avaliação Laboratorial, elencou a Nefelometria e a Imunoturbidimetria como relevante técnica para aferir a concentração de certas proteínas plasmáticas numa amostra.

A seguir, uma classificação geral dos tipos de turbidímetros:

- Turbidímetro de Mesa ou de Laboratório (Figura 1);



Figura 1 – Turbidímetro de Mesa ou de Laboratório
Fonte: UNIVERSIDADE... (2015b)

- Turbidímetro portátil (Figura 2);



Figura 2 – Turbidímetro Portátil
Fonte: UNIVERSIDADE... (2015b)

- Turbidímetro primitivo com precisão subjetiva (Figura 3);



Figura 3 – Turbidímetro primitivo
Fonte: UNIVERSIDADE... (2015b)

- Turbidímetro Industrial (Figura 4);



Figura 4 – Turbidímetro Industrial
Fonte: Tecnisis (2015)

Micro e pequenas empresas demandam, eventualmente, por um sistema de inspeção automático da turbidez em série para aumentar sua capacidade produtiva. No entanto, a aquisição de um Controlador Lógico Programável – CLP – e mais um software de sistema supervisorio, como o LabVIEW, por exemplo, para realizar a interface homem-máquina, além do software de inspeção Vision Builder, é possivelmente um investimento muito alto em relação ao risco de situações desfavoráveis de mercado, que podem, inclusive, levar à falência da empresa em função da contração de dívidas. Portanto, essa é uma situação de demanda reprimida em função do alto custo para o porte dessas empresas. O presente trabalho visa à pesquisa de uma solução para tal demanda.

1.4 PROCEDIMENTOS METODOLÓGICOS

1. Estudar sobre a natureza da turbidez da água e suas técnicas de aferição.
2. Primeiramente, conectar-se-á e comunicar-se-á uma câmera do tipo *webcam* (facilmente encontrada no mercado e a baixo custo) com o software Vision Builder, da National Instruments, via PC, bem como calibrá-la para a perfeita aquisição de imagens;
3. Após, realizar-se-á um estudo de como operar o software Vision Builder, buscando manuais do fabricante e realizando testes com o fim de configurar e programar uma sequência de tarefas no aludido software que seja capaz de realizar uma inspeção da transparência ou da ausência de transparência da água contida em frascos de ensaio, isto é, sua turbidez. Paralelamente à ação anterior, estudar-se-á a necessidade e, se for o caso, definir-se-á quais as formas de iluminação artificial que otimizam a tarefa da câmera e do Vision Builder, documentando todos os testes e resultados;
4. Identificar e realizar a comunicação da placa Arduino com o Vision Builder, via PC, e este conectado e comunicando com a câmera. O Arduino é uma placa microcontroladora programada através da linguagem C++, contendo também funções extras bastante úteis. É muito popular em projetos eletrônicos face sua aplicação bastante flexível;
5. Em sequência ao trabalho, construir-se-á um sistema seletor de frascos do tipo carrossel ou com função genérica, para simular a seleção dos frascos em ambiente industrial, ainda que o presente estudo seja implementado numa linha de produção que já possua seu sistema próprio de transporte e seleção de peças. Para tanto, deverão ser adquiridas peças e componentes reciclados de impressoras desmontadas, pois em Curitiba já existem comércios especializados em componentes mecatrônicos reciclados. Com uma placa de circuito aberto, projetar-se-á e soldar-se-á um circuito eletrônico de potência para alimentar o(s) motor(es) do sistema seletor e integrá-lo(s) com os comandos do Arduino.
6. Programar-se-á uma rotina para o Arduino para o controle geral do sistema: tanto do sistema seletor, quanto da sequência de inspeções realizadas pelo Vision Builder. Para a IHM do sistema, Interface Homem Máquina, será

implementado um display de LCD e um teclado numérico, ambos em comunicação direta com o Arduino.

7. Testes finais para que operem todos os aparelhos integrados e em tempo real;

1.5 ESTRUTURA DO TRABALHO

O trabalho terá a estrutura abaixo apresentada.

Capítulo 1 - Introdução: serão apresentados o tema, a orçamentária do estudo, o problema e a premissa, os objetivos da pesquisa, a justificativa, os procedimentos metodológicos, e a estrutura geral do trabalho.

Capítulo 2 – Fundamentação teórica: serão elencados sinteticamente conceitos, estudos anteriores e conhecimentos científicos acerca do tema a ser explorado neste trabalho.

Capítulo 3 – Desenvolvimento do protótipo: serão apresentados os componentes do sistema e as etapas do desenvolvimento e construção do sistema.

Capítulo 4 – Apresentação e Análise dos Resultados: após o desenvolvimento do protótipo e a realização de testes com base nos procedimentos metodológicos, serão feitas as devidas análises e descritos os resultados obtidos relacionados à inspeção visual da turbidez da água.

Capítulo 5 – Considerações finais: serão retomados a pergunta de pesquisa e os seus objetivos e será apontado como foram solucionados, respondidos, atingidos, por meio do trabalho realizado. Além disto, serão sugeridos trabalhos futuros que poderão ser realizados a partir do estudo realizado.

2 FUNDAMENTAÇÃO TEÓRICA

Em síntese, Simms (1973) elucida que a turbidez ou turvação de um fluido orgânico ou inorgânico é a propriedade física inversamente proporcional à sua transparência.

Primitivamente, as formas ou técnicas de medir a turvação de um líquido resumiam-se a uma escala gráfica submersa, ou justaposta num distanciamento gradativo da superfície do fluido. O principal problema ao aferir tal medida encontrava-se na acuidade visual do observador, o que tornava a tarefa um tanto quanto subjetiva e bastante imprecisa (SIMMS, 1973).

Ainda assim, segundo Simms (1973), buscando uma padronização nessas técnicas de medição, surgiram unidades de medida, como a JTU (Unidade de Turvação de Jackson), por exemplo, baseada no método da vela de Jackson (*Jackson Candle Method*), entre outras.

Mais tarde, baseadas em graduações da diluição de formazina com água, diz Ziegler (2002), surgiram unidades mais bem definidas e padronizadas pelo mundo inteiro. Entre as mais comuns, destaca-se a NTU (ou UNT – Unidade Nefelométrica de Turbidez), padronizada nos EUA, que também pode aparecer descrita como mg/l de SiO_2 .

Santos (2009), disserta que a diferença entre a turbidimetria e a nefelometria está no ângulo de leitura ou do sensor óptico em relação à fonte de luz alinhada com a amostra sob medição. Enquanto na turbidimetria o ângulo é de 180° em relação à fonte alinhada à amostra, na nefelometria o ângulo é de 90° (Figura 5).

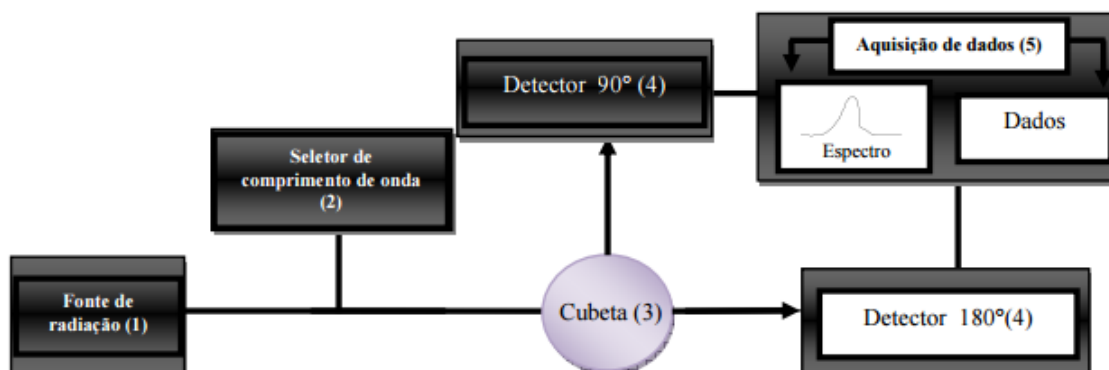


Figura 5 – Esquema de Sistema Nefelométrico e Turbidímetro
 Fonte: Santos (2009)

Já para Naoum (2015), biólogo, “a Nefelometria é uma técnica para medir as concentrações de Ig A, Ig M e Ig G e outras proteínas plasmáticas de uma amostra.”

Segundo a UNIVERSIDADE... (2015b), descreve-se a seguir os procedimentos para leitura manual da turbidez da água usando um turbidímetro nefelométrico.

- 1 - As amostras devem ser coletadas em um recipiente de vidro ou polietileno limpo.
- 2 - As amostras devem ser analisadas logo após a coleta, pois a turbidez pode mudar se a amostra for armazenada.
- 3 - Tubos de amostra sujos, arranhados ou lascados podem causar resultados (leituras) de testes altos. Os tubos devem ser lavados com ácido periodicamente e ligeiramente lubrificados com silicone para mascarar possíveis imperfeições do vidro. Os tubos que estiverem muito arranhados ou lascados não devem ser utilizados. Identifique os tubos para eles possam ser orientados de maneira consistente na câmara de medição de turbidez
- 4 - Cor excessiva na amostra (maior que 15 unidades de cor) pode causar leituras de turbidez baixas, uma vez que a cor irá absorver a luz.
- 5 - Antes de proceder à leitura de turbidez a amostra deve ser gentilmente invertida algumas vezes para misturar. Tome cuidado para não produzir bolhas de ar; elas provocam leituras de turbidez altas. Uma seringa acoplada a um rolha de borracha é a melhor maneira de removê-las. Se houver aderência de bolhas às paredes laterais do tubo de amostra tampe-o e, gentilmente, role-o afim de eliminá-las.
- 6 - Partículas de carbono causam resultados de turbidez baixos, pois também absorvem luz.
- 7 - As partículas que causam turbidez são com frequência carregadas eletricamente. Portanto, campos elétricos ao redor de motores podem afetar as leituras.
- 8 - Vibrações podem aumentar a dispersão de luz e, por consequência, provocar leituras de turbidez altas. O turbidímetro deve ser colocado sobre uma bancada de superfície firme.
- 9 - Após terminar o manuseio e o preparo da amostra, remova as marcas de dedos no tubo de amostra com um pano seco e limpo antes de inseri-lo na câmara do turbidímetro.
- 10 - Os padrões de turbidez têm prazo de validade. Observe-os. Padrões do tipo formazina são válidos por período muito limitado. Devem ser eliminados após a calibração. (UNIVERSIDADE..., [2015b])

Em relação ao princípio de funcionamento do medidor digital de turbidez da água (turbidímetro ou nefelômetro), segue:

A turbidez é a medida da dificuldade de um feixe de luz atravessar uma certa quantidade de água, conferindo uma aparência turva à mesma. Essa medição é feita com o turbidímetro ou nefelômetro, que compara o espalhamento de um feixe de luz ao passar pela amostra, com o de um feixe de igual intensidade, ao passar por uma suspensão padrão. Quanto maior o espalhamento, maior será a turbidez.

Os valores são expressos, normalmente, em Unidades Nefelométricas de Turbidez - UNT, ou em mg/l de SiO₂ (miligramas por litro em Sílica). O turbidímetro da figura pode medir a turbidez em três escalas: 0 - 20, 0 - 200 e 0 - 1.000 UNT's sendo que, na escala de 0 - 20, a resolução é de 0,01 UNT.

A Cor da água (>15 unidades de cor) e as partículas de carbono interferem na medida da turbidez devido às suas propriedades de absorverem a luz. As amostras devem ser analisadas logo após a coleta (que deve ser feita em recipiente de vidro ou de PVC bem limpos), pois a turbidez pode mudar se a amostra for armazenada por um certo tempo. (UNIVERSIDADE..., [2015a])

Ainda, sobre as causas da turbidez (Figura 6):

As principais causas da turbidez da água são: presença de matérias sólidas em suspensão (silte, argila, sílica, coloides), matéria orgânica e inorgânica finamente divididas, organismos microscópicos e algas. A origem desses materiais pode ser o solo (quando não há mata ciliar); a mineração (como a retirada de areia ou a exploração de argila); as indústrias; ou o esgoto doméstico, lançado no manancial sem tratamento. O exame microscópico e observações in loco do manancial podem ajudar a determinar as causas da turbidez. Esses materiais se apresentam em tamanhos diferentes, variando desde partículas maiores (> 1 um), até as que permanecem em suspensão por muito tempo, como é o caso das partículas coloidais (diam.=10⁻⁴ a 10⁻⁶ cm). (UNIVERSIDADE..., [2015a])



Figura 6 – Exemplos de níveis de turbidez da água em UNT's.
Fonte: UNIVERSIDADE... (2015a)

Há que se distinguir entre a matéria suspensa ou sedimento, que precipita rapidamente, daquela que se precipita vagarosamente e que provoca a turbidez. A turbidez é encontrada em quase todas as águas de superfície, em valores elevados (até 2.000 mg/l de SiO₂), mas está normalmente ausente nas águas subterrâneas.

As águas de lagos, lagoas, açudes e represas apresentam, em geral, baixa turbidez, porém variável em função dos ventos e das ondas que, nas rasas, podem revolver os sedimentos do fundo. Via de regra, após uma chuva forte, as águas dos mananciais de superfície ficam turvas, graças ao carreamento dos sedimentos das margens pela enxurrada. Assim, os solos argilosos e as águas em movimentação, ocasionam turbidez. (UNIVERSIDADE..., [2015a])

E Naoum (2015) explica, também, a importância da turbidez em exames laboratoriais na área biológica.

A Nefelometria é uma técnica para medir as concentrações de Ig A, Ig M e Ig G e outras proteínas plasmáticas de uma amostra. Um aparelho específico que mede a turbidez é utilizado e mede a difração (desvio) da luz ao passar por uma solução contendo complexos imunológicos. Já a Imunoturbidimetria mede a diminuição da luz ao passar por um complexo antígeno-anticorpo, em outras palavras, a turbidimetria mede o quanto a solução antígeno-anticorpo absorve da luz e o quanto ela deixa passar. Essa técnica, assim como a nefelometria, é usada para medir a concentração plasmática de diversas proteínas. A principal diferença entre nefelometria e turbidimetria é que na nefelometria a luz difundida ou seja aquela que atravessa a solução é medida, enquanto que na turbidimetria a luz não difundida (a absorvida) é medida. (NAOUM, [2015])

3 DESENVOLVIMENTO DO PROTÓTIPO

3.1 COMPONENTES DO SISTEMA

3.1.1 Componentes físicos do sistema

Como componentes principais do sistema pode-se discriminar quatro principais elementos: o carrossel seletor, a caixa controladora, a câmera, e o computador. Há subcomponentes que também serão descritos a seguir.

3.1.1.1 Carrossel Seletor

Para realizar a tarefa de seleção e transporte dos frascos de ensaios foi construído um carrossel seletor que possui capacidade para até oito (8) frascos de ensaio (Figura 7).

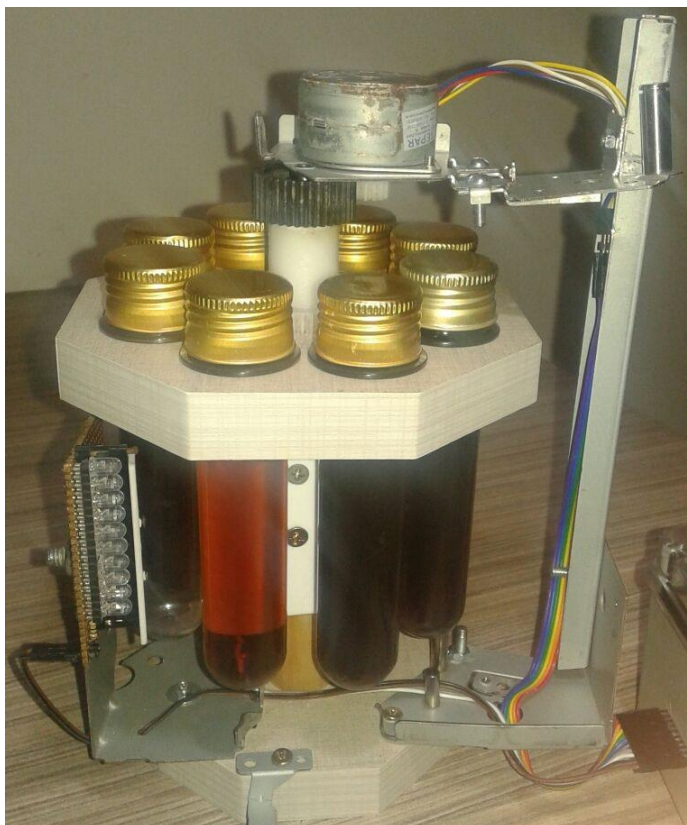


Figura 7 – Carrossel Seletor
Fonte: Aatoria própria

O carrossel é movimentado pela combinação de duas engrenagens movidas por um motor de passo unipolar alimentado com C.C. e tensão de 20V fornecidos pela caixa controladora. A engrenagem acoplada ao eixo principal do carrossel (coroa) possui 37 dentes, e a engrenagem acoplada ao eixo do motor de passo (pinhão) possui 18 dentes. Com essa combinação de engrenagens, pôde-se aumentar o torque do sistema bem como a suavidade dos movimentos. A suavidade se explica comparando-se a rotação de uma volta completa de 360 graus do eixo principal com uma volta completa do eixo do motor de passo. Para um dado ângulo de rotação do eixo principal, o eixo do motor de passo precisa rotacionar um ângulo ainda maior e uma maior quantidade de passos do motor são necessários. Portanto, cada passo movimenta um ângulo menor do eixo principal comparado ao ângulo do eixo do motor, e, por consequência, o tranco de cada passo torna-se menos perceptível, ou seja, uma rotação mais suave e uniforme do que se o motor estivesse acoplado diretamente no eixo principal. O torque foi aumentado em mais de duas vezes, o que é demonstrado pela fórmula (1):

$$\frac{T_c}{T_p} = \frac{d_c}{d_p} \quad \therefore \quad \frac{T_c}{T_p} = \frac{37}{18}$$

$$T_c = 2,055 \times T_p \quad (1)$$

Onde: T_c = Torque obtido na coroa (engrenagem de roda maior)

T_p = Torque aplicado pelo motor no pinhão (engrenagem de roda menor)

d_c = Número de dentes da coroa

d_p = Número de dentes do pinhão

Integram o carrossel, ainda, duas fileiras de LEDs de alto brilho, que, em conjunto com a câmera, aliam as técnicas da turbidimetria e da nefelometria para a inspeção da turbidez da água.

3.1.1.2 Caixa Controladora

A caixa controladora (Figura 8) é composta por uma tela de LCD (*Liquid Cristal Display*) de 16x2 caracteres e um teclado matricial numérico para atuar como interface homem máquina - HMI (*Human Machine Interface*).



Figura 8 – Caixa Controladora

Fonte: Autoria própria

Possui uma entrada USB (*Universal Serial Bus*) tipo B *standard*, uma entrada para plug de alimentação tipo Jack (Figura 9), e um conector de saída para cabo flat de 10 vias (Figura 10).



Figura 9 – Porta USB tipo B *standard* e entrada Jack
Fonte: Autoria própria



Figura 10 – Saída para cabo flat
Fonte: Autoria própria

Seus componentes internos são duas placas elétricas (Figura 11). A primeira é a placa microcontroladora Arduino Mega 2560, responsável pelo controle central dos processos de todo o sistema. A segunda trata-se de uma placa de potência para alimentar a tela de LCD e, principalmente, o motor de passo.



Figura 11 – Placa Microcontroladora e Placa de Potência
Fonte: Autoria própria

3.1.1.3 Câmera

A câmera utilizada é a webcam Logitech HD Webcam C270 (Figuras 12 e 13). Trata-se de uma câmera com resolução boa para o software que a utiliza, o Vision Builder, com resolução de vídeo de 1280x720, e conexão USB 2.0 tipo A.



Figura 12 – Webcam Logitech HD C270 acoplada ao sistema
Fonte: Autoria própria



Figura 13 – Webcam Logitech HD C270
Fonte: Autoria própria

3.1.1.4 Computador

O computador utilizado é um notebook HP com processador Intel Core i3, CPU 1.7 GHz, com 4GB de memória RAM, e sistema operacional Windows 8.1.

3.1.2 Componentes virtuais do sistema

Como componentes virtuais do sistema, além do sistema operacional Windows como requisito, elenca-se o software Vision Builder for Automated Inspections – VBAI – 2013 da empresa National Instruments, e também a rotina de programação em linguagem C++ da placa microcontroladora Arduino.

O Vision Builder é um software voltado à inspeção de qualidade de produtos em uma linha de fabricação. Realiza inspeções visuais às mais variadas necessidades: busca por falhas na usinagem, como por exemplo, fissuras, descontinuidades, buracos, formas e tamanhos dentro de limites padrões até os centésimos de milímetro, dependendo da ampliação da imagem, ausência ou existência de corpos estranhos, e aplicação de filtros e tratamentos digitais para quantificar um determinado aspecto da imagem que se deseja verificar.

A programação em C++ é realizada em ambiente próprio, isto é, um software próprio do Arduino/Genuino disponível para download gratuito em <https://www.arduino.cc/en/Main/Software>, em que é possível editar, compilar e carregar tais programas para o microcontrolador, sempre pela porta USB, por onde também é alimentado com tensão de trabalho de 5V.

3.2 ETAPAS DO DESENVOLVIMENTO E CONSTRUÇÃO DO SISTEMA

3.2.1 Conexão da câmera com o Vision Builder

Por se tratar de uma câmera com tecnologia PnP (*Plug & Play*), para realizar a conexão da câmera com o Vision Builder, seleciona-se a opção 'Acquire

Image (1394, GigE, or USB)', listada na aba de aquisição de imagens (*Acquire Images*). Esta aba está discriminada na caixa de passos da inspeção (*Inspection Steps*), que está habilitada, por sua vez, na interface de configuração da inspeção (*Configuration Interface*). Automaticamente o software deverá ter reconhecido a câmera ligada à porta serial universal e a elencará na lista de câmeras disponíveis que captarão as imagens.

3.2.2 Primeiros testes de inspeção da turbidez

Como teste experimental, as primeiras imagens capturadas e salvas para simulação foram de frascos contendo café em diferentes diluições. Cada frasco é preenchido com 35 ml. Nesse primeiro experimento, encheu-se um frasco com café puro, no estado no qual foi coado, isto é, sem diluições secundárias a não ser a diluição original intrínseca ao ato de coar.

Num segundo frasco, foi preenchido seu volume com 50% de café e 50% de água. No terceiro, preparou-se uma diluição idêntica à do segundo frasco (50%-50%) e preencheu-se 50% do frasco, completando o restante com água. Ou seja, neste terceiro frasco, a quantidade de café representa 25% ($0,5 * 0,5 = 0,25$) da solução contra 75% de água pura. No quarto frasco e assim por diante, repetiu-se o mesmo método de diluição, sendo que a proporção de café neste quarto frasco ficou então em 12,5% ($0,25 * 0,5 = 0,125$). É uma sequência de diluição exponencial, o que não demonstrou ser um método ideal nesse momento, em que se buscou testar primeiro a funcionalidade do sistema. Esse método foi então descartado e preparou-se um segundo lote de soluções de café diluído nas seguintes proporções (Tabela 1):

Tabela 1 – Proporção de diluição de café em água

	Café	Água
Frasco 1	0	100%
Frasco 2	1/7	6/7
Frasco 3	2/7	5/7
Frasco 4	3/7	4/7
Frasco 5	4/7	3/7
Frasco 6	5/7	2/7
Frasco 7	6/7	1/7
Frasco 8	100%	0

Fonte: Autoria própria

A seguir, para a inspeção destas soluções, foi editada uma nova inspeção. Após adquirir as imagens, e para que o software possa mensurar a turbidez da solução fotografada, foi adicionada aos passos da inspeção a etapa *Vision Assistant 1*, que tratou a imagem com um filtro de extração de cores, mais precisamente o filtro *HSI – Intensity Plane*. Tal filtro transforma as cores da imagem numa escala de cinza, isto é, uma imagem “preta e branca” como é coloquialmente conhecida. A imagem fica, então, em escala de cinza no padrão RGB, com *range* de 0 a 255, onde 0 representa o preto puro e 255 o branco puro. A partir daí, o programa tem condições de mensurar a quantidade de luz que atravessa o frasco fotografado, baseado na escala de branco ao longo latitudinal do frasco.

Delimita-se então o campo, ou a área específica da imagem na qual a inspeção deverá mensurar a escala de cinza. Nessa mesma etapa de delimitação, definem-se também os limites aceitáveis na escala de cinza para considerar a imagem aprovada ou reprovada. Ressalta-se que é nesta etapa em que a configuração desejada para cada caso específico deverá ser realizada individualmente pelo usuário de acordo com a sua necessidade em particular.

3.2.3 Iluminação e posicionamento da câmera

Foi idealizada e aplicada uma combinação de métodos de aquisição de imagens para a inspeção da turbidez, quais sejam, a turbidimetria e a nefelometria.

Para tanto, foram implementadas ao carrossel seletor duas fileiras de LED's de alto brilho na cor branca. Sabendo que os pinos de saída do Arduino fornecem uma tensão aproximada de 5V, e que a queda de tensão sobre cada LED está em torno de 3,45V, foi calculada uma corrente teórica de 15,5 mA para cada LED para maximizar a sua emissão de luz dentro de um nível seguro de operação. No entanto, constatou-se que cada pino do Arduino consegue fornecer não mais do que 40 mA. Portanto, tem-se que a corrente real sobre cada LED está em torno de 4 mA. Ainda assim, o funcionamento foi satisfatório. Foram construídos os seguintes circuitos elétricos esquematizados abaixo para ambas as fileiras (Figura 14):

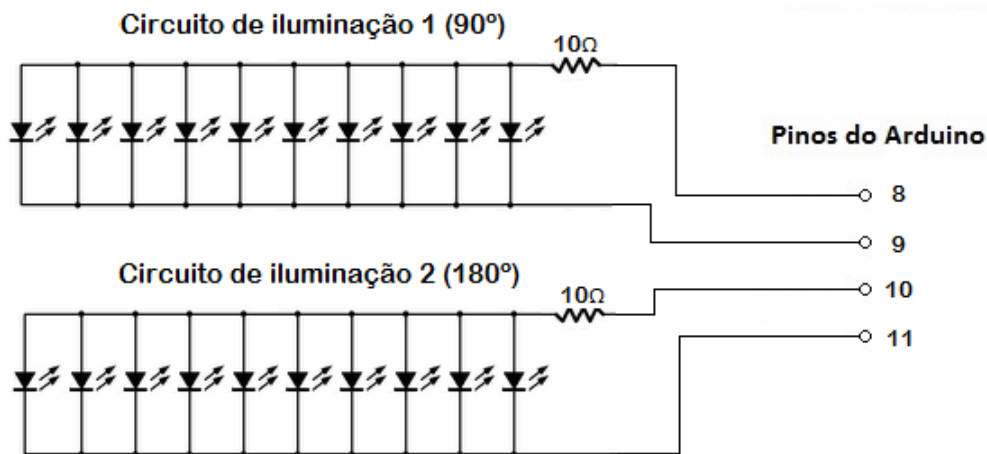


Figura 14 – Circuitos de iluminação
Fonte: Autoria própria

Uma das fileiras, com leds de 3mm, foi posicionada a exatamente 180 graus da câmera, que capta, então, a luz que atravessa diretamente o frasco. Tal é o método da turbidimetria. A segunda fileira, com leds de 5mm, foi posicionada a 90 graus da câmera, que capta, portanto, a reflexão da luz por entre o frasco. Tal é o método da nefelometria. A luminosidade produzida por leds de 3mm ou 5mm é a mesma, com a única diferença do tamanho. A figura abaixo ilustra o esquema do posicionamento descrito (Figura 15).

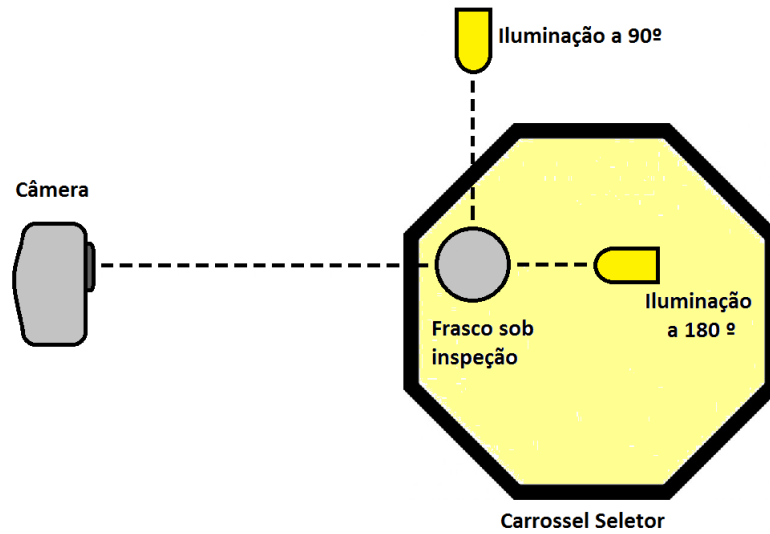


Figura 15 – Posicionamento da câmera e das fontes de iluminação
Fonte: Autoria própria

Após os primeiros testes com essa iluminação direta, observou-se, principalmente em relação à iluminação a 180 graus, uma inconsistência nas medições. A falta de uniformidade na luz emitida foi a principal causa. Para contornar este problema, foi implementada à frente das fileiras de LED's, uma pequena placa de acrílico branca com a função de difratar a luz, tornando-a a mais uniforme possível ao longo da fileira emissora (Figuras 16 e 17).



Figura 16 – LED's a 90° com filete de placa de acrílico
Fonte: Autoria própria

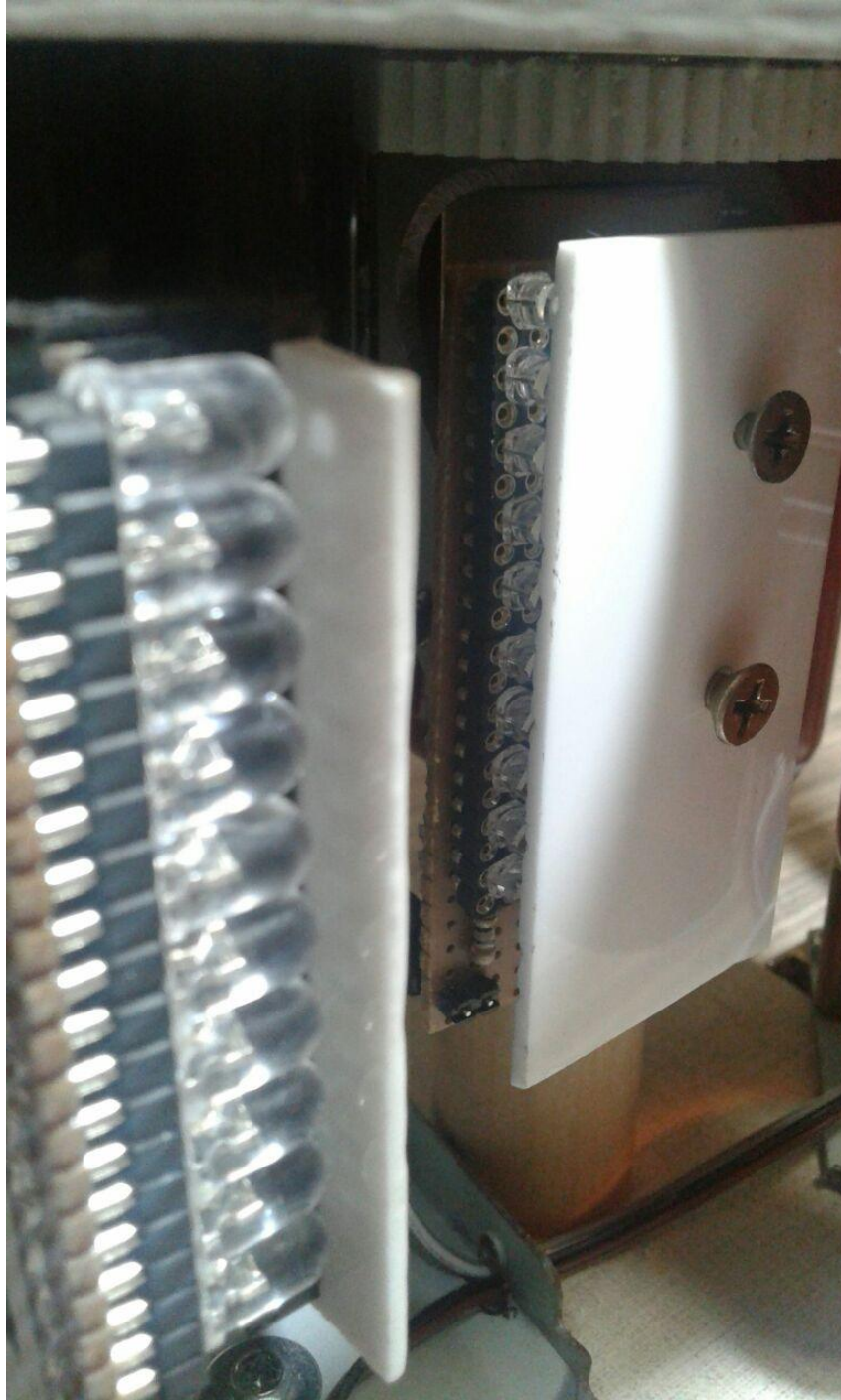


Figura 17 – Em foco, LED's a 180° com placa de acrílico à frente
Fonte: Autoria própria

Essa solução se revelou eficaz frente às incoerências das medições com incidência de luz direta conforme se observa na tabela a seguir (Tabela 2):

Tabela 2 – Medições antes e depois da implementação de difrator de luz

Frasco	Sem difrator de luz		Com difrator de luz		
	Porcentagem de água	Escala de cinza (0-255)	Porcentagem do range	Escala de cinza (0-255)	Porcentagem do range
1	100%	248	97%	247	97%
2	86%	212	83%	130	51%
3	71%	219	86%	122	48%
4	57%	235	92%	125	49%
5	43%	195	76%	108	42%
6	29%	183	72%	89	35%
7	14%	173	68%	56	22%
8	0%	162	64%	30	12%

Fonte: Autoria própria

As medições realizadas antes da aplicação do difrator de acrílico demonstraram que, do frasco 2 ao frasco 4, houve um aumento na leitura de luz quando na verdade deveria haver uma diminuição, uma vez que a porcentagem de água também diminuiu, indicando soluções mais turvas. Após a aplicação do difrator, as mensurações passaram a ficar coerentes com o aumento gradativo de café diluído.

3.2.4 Automatização do transporte dos frascos

Para a automatização do transporte dos frascos, por se tratar de um carrossel, foram implementados um conjunto de duas engrenagens e um motor de passo conforme detalhado no item 3.1.1.1. Para o funcionamento do motor de passo, foi necessário o desenvolvimento de um circuito de potência para a alimentação e uma rotina de programação no microcontrolador Arduino, necessária para coordenar o movimento do motor passo a passo.

3.2.4.2 Circuito de potência do motor de passo

Para realizar a ponte entre o comando de movimento enviado pelo microcontrolador e a necessária corrente contínua de 20V proveniente da fonte de alimentação, utilizou-se o CI (Componente Integrado) ULN2003. O circuito de potência desenvolvido foi esquematizado abaixo (Figura 18).

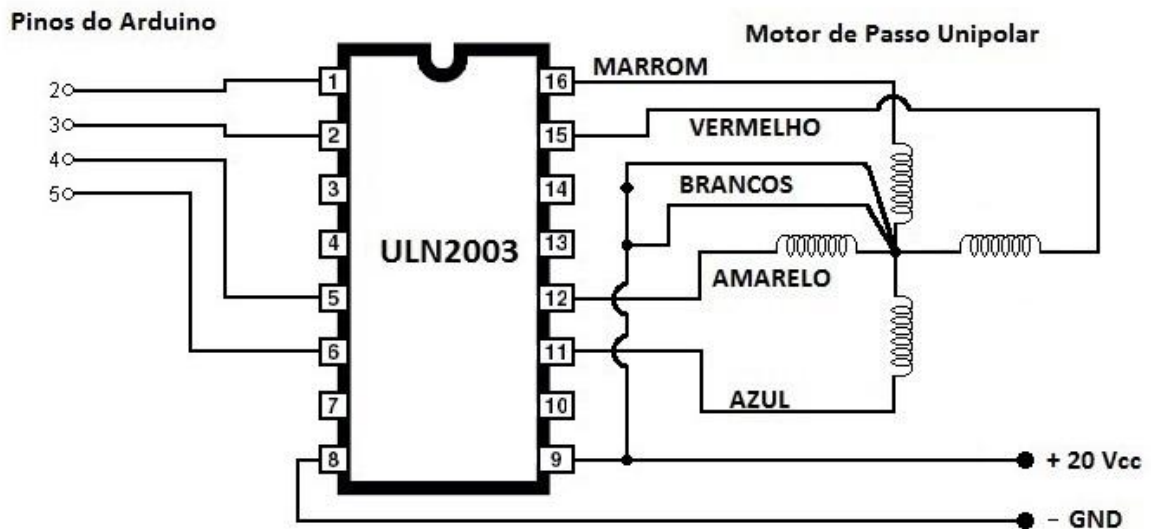


Figura 18 – Circuito de potência do motor de passo
Fonte: Autoria própria

O motor de passo utilizado para girar, primeiramente, precisa ter seus fios brancos energizados com 20V constantemente. Para girar no sentido horário, é preciso coordenar a ligação dos demais fios com o GND (*ground*) na seguinte sequência: azul, vermelho, amarelo, marrom, ou seja, elevar o sinal de saída para 5V dos pinos 5, 3, 4, 2, do Arduino nessa mesma sequência. A partir do momento que qualquer dos pinos de 1 a 7 do ULN2003 recebem sinal alto, ou seja, 5V, os pinos de 16 a 10 são respectivamente conectados ao pino 8 (GND), em virtude das conexões dos transistores internos do CI. Para girar no sentido horário, o procedimento é o mesmo, invertendo-se tão somente a sequência descrita.

3.2.4.3 Rotina de programação para o motor de passo

Para o comando do motor de passo, foram desenvolvidas duas sub rotinas.

A primeira, que segue transcrita abaixo, é uma camada mais básica da programação que comanda diretamente a modificação das portas de saída 2 a 5 em função do passo escolhido. Existem 8 combinações possíveis de níveis alto e baixo das quatro portas de saída. Essa primeira sub rotina apenas recebe qual das 8 combinações foi requisitada, isto é, qual dos 8 passos foi requisitado, e efetua a modificação das portas.

```
void motorpasso(int passo, int veloc)
{
  if(passo==1)
  {
    digitalWrite(5, LOW);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(2, LOW);
    delay(veloc);
  }
  if(passo==2)
  {
    digitalWrite(5, LOW);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(2, HIGH);
    delay(veloc);
  }
  if(passo==3)
  {
    digitalWrite(5, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
    digitalWrite(2, HIGH);
    delay(veloc);
  }
  if(passo==4)
  {
    digitalWrite(5, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
    digitalWrite(2, HIGH);
    delay(veloc);
  }
}
```

```

}
if(passo==5)
{
  digitalWrite(5, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(2, HIGH);
  delay(veloc);
}
if(passo==6)
{
  digitalWrite(5, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(2, HIGH);
  delay(veloc);
}
if(passo==7)
{
  digitalWrite(5, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(2, LOW);
  delay(veloc);
}
if(passo==8)
{
  digitalWrite(5, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(2, LOW);
  delay(veloc);
}
}
}

```

A segunda sub rotina, transcrita abaixo, é uma camada superior da programação. Ela atua diretamente sobre a sub rotina apresentada anteriormente, e, portanto, indiretamente sobre o motor de passo.

```

void gira(char sentido[12])
{
  int x;
  if(sentido=="horario")
  {
    if(terco==1)
    {
      for(x=0;x<25;x++) // inicia no passo x e termina no passo x+1
      {

```

```
    passo++;
    if(passo>8)
    {
        passo=1;
    }
    motorpasso(passo, 16);
}
}
if(terco==2) //inicia no passo x e termina no passo x+1
{
    for(x=0;x<25;x++)
    {
        passo++;
        if(passo>8)
        {
            passo=1;
        }
        motorpasso(passo, 16);
    }
}
if(terco==3) // inicia no passo x e termina no passo x
{
    for(x=0;x<24;x++)
    {
        passo++;
        if(passo>8)
        {
            passo=1;
        }
        motorpasso(passo, 16);
    }
}
terco--;
if(terco<1)
{
    terco=3;
}
frasco++;
if(frasco>8)
{
    frasco=1;
}
}
if(sentido=="antihorario")
{
    terco++;
    if(terco>3)
    {
        terco=1;
    }
}
```

```

if(terco==1)
{
for(x=0;x<25;x++)
{
passo--;
if(passo<1)
{
passo=8;
}
motorpasso(passo, 16);
}
}
if(terco==2)
{
for(x=0;x<25;x++)
{
passo--;
if(passo<1)
{
passo=8;
}
motorpasso(passo, 16);
}
}
if(terco==3)
{
for(x=0;x<24;x++)
{
passo--;
if(passo<1)
{
passo=8;
}
motorpasso(passo, 16);
}
}
frasco--;
if(frasco<1)
{
frasco=8;
}
}
}

```

Esta sub rotina foi necessária não somente para facilitar a programação da rotina principal, mas principalmente para contornar um problema matemático. Isto porque para uma volta completa do pinhão, que tem 18 dentes, o motor dá 96

passos. Para rotacionar o ângulo entre um frasco e outro, são necessários movimentar 4,625 dentes (37 dentes de uma volta completa divididos por 8 posições de frascos). Portanto, tem-se a equação (2):

$$\frac{96 \text{ passos}}{18 \text{ dentes}} \times \frac{37 \text{ dentes}}{8 \text{ posições de frascos}} = 24,6666 \text{ passos por posição.} \quad (2)$$

A cada três rotações de posicionamento são necessários 74 passos ($3 \times 24,6666$). Então, para resolver a dízima periódica, o programa envia a ordem para o motor realizar 25 passos em duas rotações, e, na terceira, envia a ordem para realizar 24 passos, totalizando os 74 passos. Após uma sequência de testes de 128 inspeções ininterruptas, verificou-se que a diferença de 0,66 passos entre uma rotação de 25 e 24 passos é irrelevante para as pequenas diferenças de medição, que estão associadas muito mais a outras imprecisões do sistema como um todo.

3.2.5 A comunicação dos periféricos teclado e LCD com o Arduino

3.2.5.2 LCD

O LCD (*Liquid Cristal Display* – “Tela de Cristal Líquido”) utilizado neste protótipo é o de modelo LCM1602B com *backlight* azul. O circuito de ligação foi esquematizado abaixo (Figura 19).

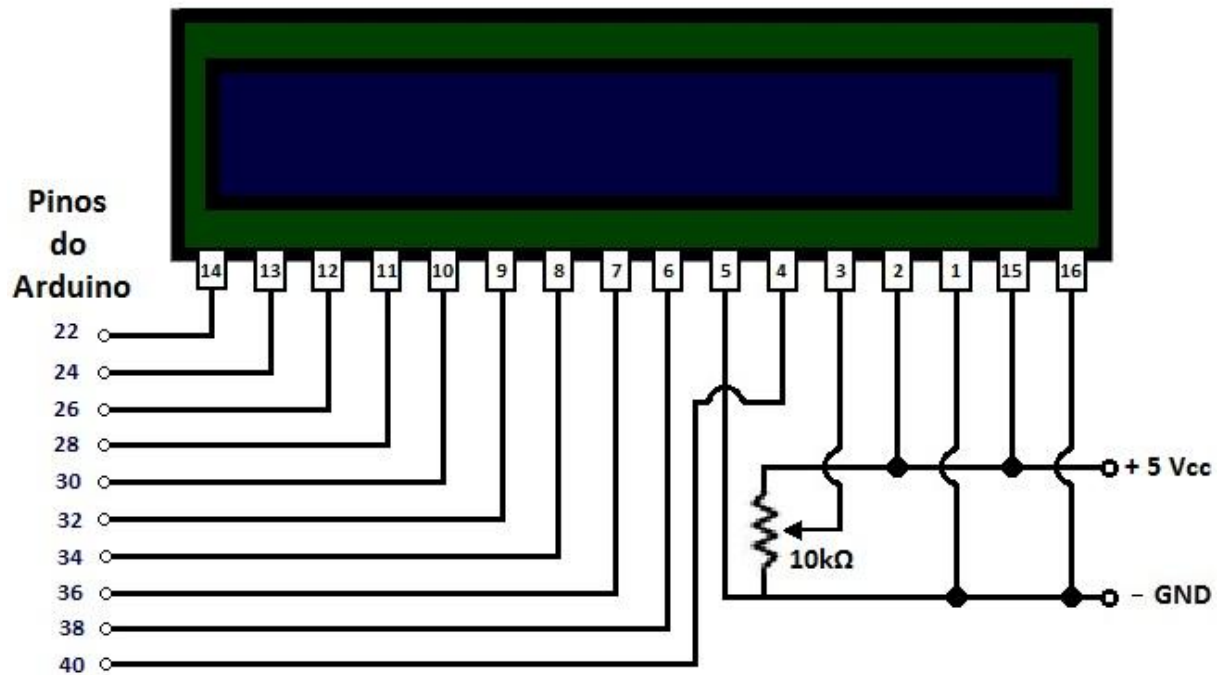


Figura 19 – Circuito de ligação do LCD
Fonte: Autoria própria

O potenciômetro de $10k\Omega$ ligado ao pino 3 do LCD serve como regulador do contraste da tela.

Para a comunicação deste *display* com o microcontrolador, o Arduino já possui uma biblioteca específica para esse tipo de LCD 16x2. Deve-se incluí-la no início do programa e inicializá-la declarando quais os pinos do Arduino escolhidos para a interface com o *display* vide segmento do programa transcrito abaixo:

```
#include <LiquidCrystal.h> // inclui a biblioteca do LCD
LiquidCrystal lcd(40, 38, 36, 34, 32, 30, 28, 26, 24, 22); //inicializa e declara os pinos
```

Após, na função *main* (função principal), antes de programar qualquer comando de escrita no LCD, deve-se, primeiro e apenas uma vez, escrever a ordem:

```
lcd.begin(16, 2);
```

3.2.5.3 Teclado

Para a comunicação do teclado com o microcontrolador, foram necessárias a criação de duas sub rotinas. Uma, de camada mais baixa, responsável pela leitura dos pinos conectados ao teclado, e decodificação da tecla correspondente. A essa, deu-se o nome de 'char teclado()' e pode ser vista na íntegra juntamente com a rotina de programação integral do Arduino transcrita no apêndice deste trabalho. A segunda sub rotina, nomeada de 'int scanf()' representa uma camada superior de programação, pois é responsável por efetuar a leitura e memorização do caractere da tecla pressionada utilizando-se para tanto da sub rotina 'char teclado()'. Pode igualmente ser vista no apêndice.

3.2.6 Comunicação Arduino / Vision Builder

Numa tentativa inicial, buscou-se a comunicação do Arduino com o Vision Builder via OPC (*Open Platform Communications*). Pesquisou-se acerca do *Arduino OPC Server* específico para Arduino desenvolvido por Ildefonso Martínez Marchena, e disponibilizado para download gratuito na página <http://www.st4makers.com/download-opc-server-for-arduino>. Além deste servidor OPC, foi necessária também a instalação de uma biblioteca adicional OPC no compilador do Arduino. Feito isso, de fato o *Arduino OPC Server* reconhece as variáveis do Arduino. No entanto, o Vision Builder não reconhece a pasta de variáveis do *Arduino OPC Server*. Ele reconhece apenas a pasta do *KepServer* ou a do *NIOPC Server* próprio da National Instruments. Esses dois últimos, por sua vez, também não reconhecem as variáveis OPC do Arduino. Diante das infrutíferas tentativas, esse meio de comunicação foi, então, descartado.

Em nova tentativa de comunicação serial, o Vision Builder habilitou a opção de comunicação serial estando o Arduino conectado à porta COM6 (Figura 20).



Figura 20 – Portas seriais universais (USB) tipo A *standard*

Fonte: Autoria própria

Existem, no Vision Builder, duas possibilidades de se programar a inspeção com a comunicação serial: o envio de um sinal pela serial em uma etapa qualquer da inspeção, e a espera de um sinal da serial em uma etapa qualquer. Ele fica aguardando um determinado sinal para poder prosseguir com o programa de inspeção. Enquanto o sinal não for recebido, a inspeção fica em espera.

Assim sendo, para este projeto programou-se da seguinte forma: antes de iniciar a sequência de passos da inspeção, o primeiro passo é o de aguardar a chegada do sinal do Arduino autorizando o início da inspeção. A inspeção é realizada primeiramente dentro dos parâmetros de iluminação a 90 graus da câmera. Feita a inspeção a 90 graus, o Vision Builder envia um sinal de aprovação ou reprovação da inspeção ao Arduino, e fica aguardando novamente o sinal de autorização do Arduino para prosseguir com a segunda etapa da inspeção, pois o microcontrolador precisa trocar a iluminação de 90 para 180 graus, para só depois autorizar o Vision Builder a prosseguir com a segunda parte da inspeção dentro dos parâmetros de iluminação a 180 graus. Após o envio da autorização, o Arduino, por sua vez, fica aguardando o sinal do Vision Builder com o resultado da inspeção.

Com relação ao tipo de sinal enviado e recebido, trata-se de uma *string*, que pode ter um ou mais bytes de tamanho, enviada e recebida sempre em código

decimal da tabela ASCII (*American Standard Code for Information Interchange*; “Código Padrão Americano para o Intercâmbio de Informação”). Para indicar o final de uma mensagem enviada ou recebida, optou-se pela utilização tão somente do *Carriage Return* (Retorno de Carro) como pode ser observado nas figuras 21 e 22 onde, em ‘*Termination Character*’, se selecionou a opção ‘CR (\0D)’. Conforme a tabela ASCII, *Carriage Return* é o código decimal 13, que é enviado como último caractere de uma *string*. Portanto, ao receber o número 13 em decimal pela serial, o microcontrolador ou o computador identificam que termina ali a *string* recebida.

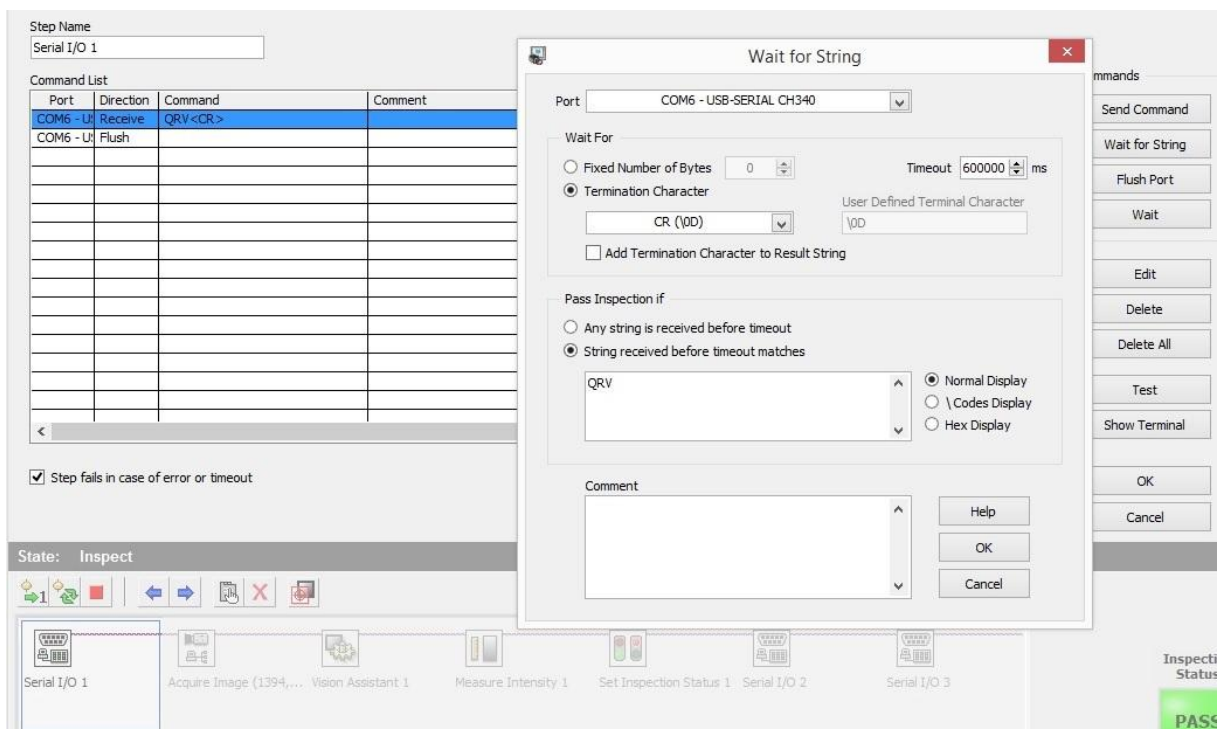


Figura 21 – Configuração de espera de *string* pela serial no VBAI
Fonte: Autoria própria

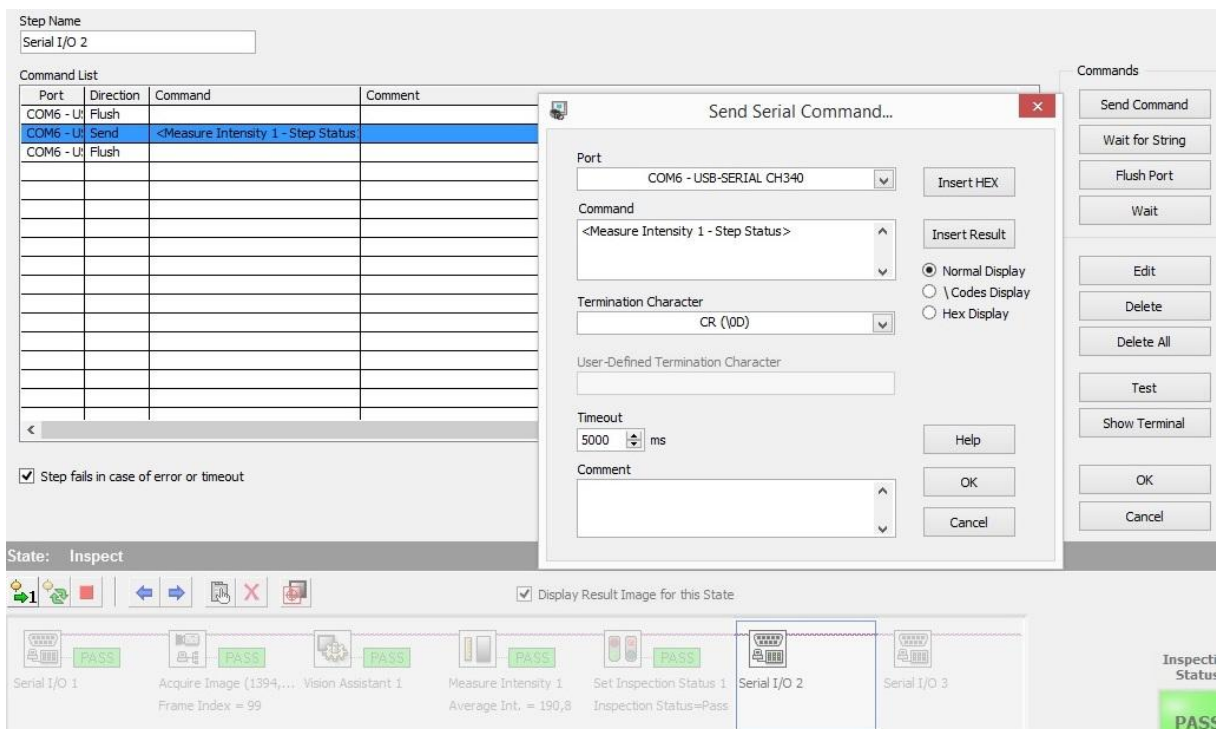


Figura 22 – Configuração de envio de mensagem pela serial no VBAI

Fonte: Autoria própria

Após o envio ou o recebimento de uma mensagem pela porta serial, é importante “limpá-la” para que não haja a possibilidade de que algum “resto” da última mensagem seja misturado com uma nova mensagem, confundindo a comunicação. Para tanto, adiciona-se o comando *‘Flush Port’* conforme Figuras 21 e 22.

3.2.7 Programa final de inspeção do Vision Builder

O programa final de inspeção do Vision Builder ficou da seguinte forma:

- 1º passo: espera pelo envio do sinal de autorização do Arduino para prosseguimento da inspeção;
- 2º passo: aquisição da imagem pela câmera;
- 3º passo: tratamento da imagem, aplicando o filtro HSI para transformar as cores em graus de escala de cinza RGB com *range* de 0 a 255;

- 4º passo: mensuração do grau de luminosidade com base nos índices da escala de cinza dentro da região de interesse previamente configurada. Feita a mensuração, o valor é imediatamente comparado com os valores limites estabelecidos pelo usuário para iluminação a 90 graus, retornando o *status* *PASS* se aprovado, ou *FAIL*, se reprovado;
- 5º passo: seta o *status* geral da inspeção como *PASS*, ou *FAIL* baseado no resultado do último passo;
- 6º passo: envia pela serial o resultado positivo ou negativo da inspeção realizada no 4º passo (Figura 23);



Figura 23 – Primeira parte do programa de inspeção do VBAI

Fonte: Autoria própria

- 7º passo: espera pelo envio do sinal de autorização do Arduino para prosseguimento da inspeção;

- 8º passo: aquisição da imagem pela câmera;
- 9º passo: tratamento da imagem, aplicando o filtro HSI para transformar as cores em graus de escala de cinza RGB com *range* de 0 a 255;
- 10º passo: mensuração do grau de luminosidade com base nos índices da escala de cinza dentro da região de interesse previamente configurada. Feita a mensuração, o valor é imediatamente comparado com os valores limites estabelecidos pelo usuário para iluminação a 180 graus, retornando o *status* *PASS* se aprovado, ou *FAIL*, se reprovado.
- 11º passo: seta o *status* geral da inspeção como *PASS*, ou *FAIL* baseado no resultado do último passo;
- 12º passo: envia pela serial o resultado positivo ou negativo da inspeção realizada no 10º passo (Figura 24);

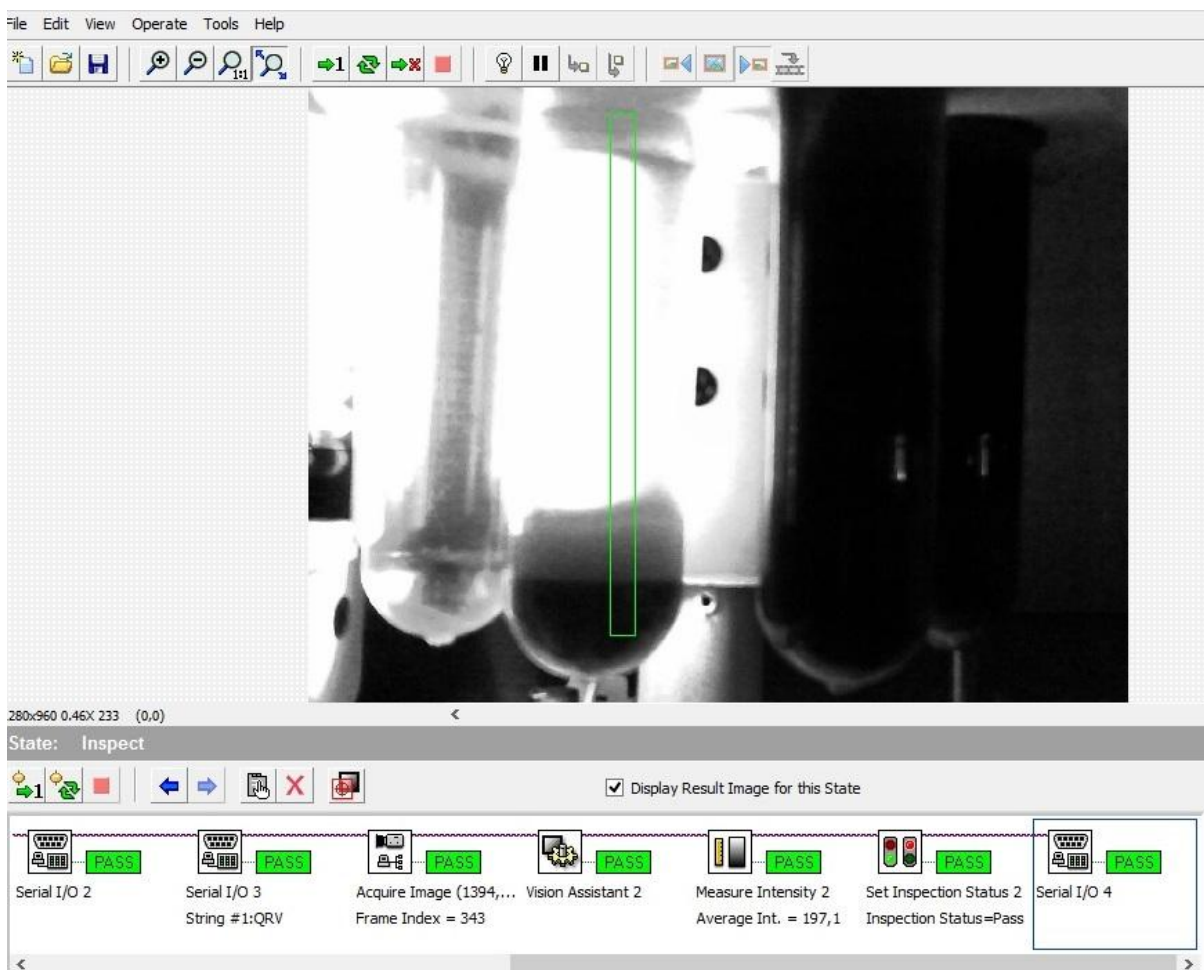


Figura 24 – Segunda parte do programa de inspeção do VBAI
Fonte: Autoria própria

3.2.8 Programação final do Arduino

A rotina de programação final do Arduino pode ser vista na íntegra no apêndice deste trabalho. Em síntese, foram criadas 10 (dez) sub rotinas para auxiliar as 2 (duas) rotinas principais, que são: 'void Main()' e 'void Loop()'. O Arduino roda sempre começando pela rotina 'void Main()'. Após terminar a rotina 'void Main()', o Arduino roda a rotina 'void Loop()' até o fim e recomeça a 'void Loop()' novamente infinitas vezes num *loop* sem fim como o próprio nome diz. Já as sub rotinas podem ser chamadas a rodar a qualquer tempo dentro das rotinas principais e até mesmo dentro umas das outras.

3.2.9 Menu de interface homem máquina

Para a interface homem máquina, foi criado um Menu de opções apresentado através do LCD, que recebe as opções escolhidas através do teclado. O Menu é apresentado da seguinte forma: após os textos de inicialização, o LCD pede para que o usuário informe ao sistema qual é o frasco parado na posição de inspeção, teclando-se um número de 1 a 8 (Figura 25). A partir dessa informação, o sistema consegue “se localizar” com relação ao posicionamento do carrossel.

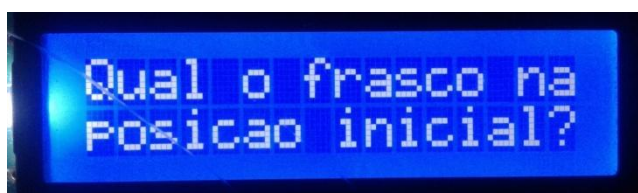


Figura 25 – Mensagem pelo LCD. Seq. 1
Fonte: Autoria própria

Após, as seguintes opções são dadas ao usuário (Figura 26):



Figura 26 – Mensagem pelo LCD. Seq. 2
Fonte: Aatoria própria

Caso o usuário tecele a opção 1, outras opções lhe serão apresentadas (Figura 27):



Figura 27 – Mensagem pelo LCD. Seq. 3
Fonte: Aatoria própria

Nesse momento, o usuário tem a opção de girar o carrossel manualmente no sentido horário ou anti horário teclando os números 1 ou 3 a depender do sentido escolhido. Quando o frasco desejado estiver na posição de inspeção, o usuário pressiona a tecla 5, e a inspeção se inicia unicamente para este frasco. Terminada a inspeção, o resultado é apresentado, e o programa retorna para o início pedindo para que o usuário selecione o modo de operação. Aí então, se o usuário optar pela opção 2 – Contínuo, as seguintes tarefas são realizadas: o Arduino comanda o início da inspeção, inspecionando primeiramente com o método de iluminação a 90 graus e posteriormente com iluminação a 180 graus. As seguintes mensagens são apresentadas (Figura 28):

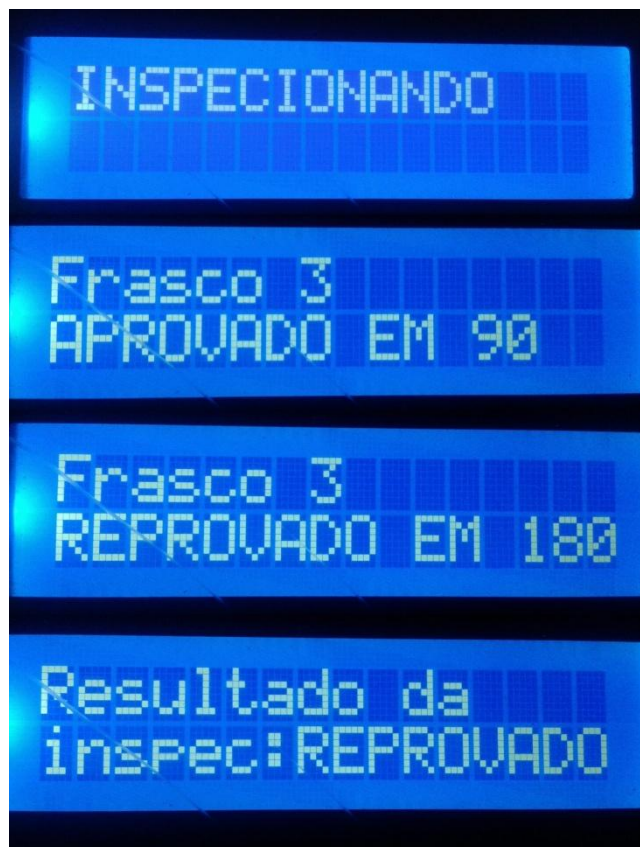


Figura 28 – Mensagem pelo LCD. Seq. 4
Fonte: Autoria própria

Terminada a inspeção deste frasco, o sistema gira no sentido horário e segue inspecionando o próximo frasco repetindo a mesma sequência, e assim por diante até o último dos oito frascos. Após a inspeção dos oito frascos, o sistema apresenta o resultado final das inspeções na tela (Figura 29):



Figura 29 – Mensagem pelo LCD. Seq. 5
Fonte: Autoria própria

Apresentado o resultado final, o sistema passa então a rotear os frascos em função da aprovação ou reprovação deles nas inspeções (Figura 30).



Figura 30 – Mensagem pelo LCD. Seq. 6
Fonte: Autoria própria

Por tratar-se de um carrossel e de um protótipo, o roteamento apresentado é didático, isto é, faz-se uma demonstração de que o sistema é capaz de rotear os frascos em função do resultado da inspeção, porém não há um

roteamento real, uma vez que não há saídas A ou B do transportador dos frascos, justamente por se tratar de um transporte circular, sem fim.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Finalizada a construção do protótipo, foram registrados os resultados de 3 (três) diferentes amostras. Foram três sequências de inspeção idênticas para os oito frascos, isto é, foram usados os mesmos frascos contendo as mesmas soluções apresentadas na Tabela 1, sem qualquer alteração feita entre uma sequência de inspeção e outra. Cumpre salientar que todas as inspeções foram realizadas em ambiente com ausência total de luz, no intuito de excluir qualquer possibilidade de interferência externa. Convém lembrar, também, que o frasco 1 contém 100% de água (o mais límpido), e que o frasco 8 contém 0% de água e 100% de café (o mais turvo). Como se trata de mensurações de uma escala de cinza com range de 0 a 255, onde 255 representa o branco e 0 representa o preto, os seguintes valores foram obtidos (Tabela 3):

Tabela 3 – Valores obtidos em três sequências de inspeção

Ângulo de iluminação	90 graus			180 graus		
	1ª	2ª	3ª	1ª	2ª	3ª
Inspeção						
Porcentagem de água						
100%	164	187	165	197	201	196
86%	91	85	88	159	152	159
71%	58	60	60	96	94	97
57%	58	58	60	94	97	97
43%	57	57	56	94	90	91
29%	53	54	54	37	38	30
14%	52	52	53	11	16	15
0%	49	50	51	7	10	7

Fonte: Autoria própria

Calculando-se a média aritmética dos dados obtidos de cada frasco obteve-se (Tabela 4):

Tabela 4 – Média aritmética dos valores obtidos em três sequências de inspeção

Ângulo de iluminação	90 graus				180 graus				
	Inspeção	1ª	2ª	3ª	Média Arit.	1ª	2ª	3ª	Média Arit.
Porcentagem de água									
100%	164	187	165	172	197	201	196	198	
86%	91	85	88	88	159	152	159	157	
71%	58	60	60	59	96	94	97	96	
57%	58	58	60	59	94	97	97	96	
43%	57	57	56	57	94	90	91	92	
29%	53	54	54	54	37	38	30	35	
14%	52	52	53	52	11	16	15	14	
0%	49	50	51	50	7	10	7	8	

Fonte: Autoria própria

Ocorre que, além da média aritmética, há uma variação maior ou menor entre cada uma das três amostras de inspeção. Para a compreensão dessa variação que satisfaça o propósito desse estudo, faz-se mister, então, uma análise não somente da simples média aritmética dos valores obtidos, mas também da margem ou faixa média de variação dos valores obtidos, isto é, da dispersão dos dados. Assim, será possível observar que para cada diluição, o sistema é capaz de realizar uma série de inspeções em condições idênticas, mas que apresentam além de um valor médio, uma variação média desse valor. Para enxergar essa dispersão média, calcula-se, portanto, o desvio padrão com a seguinte fórmula (3):

$$\sigma = \sqrt{\frac{\sum x_i^2}{n} - \left(\frac{\sum x_i}{n}\right)^2} \quad (3)$$

Onde: σ = Desvio padrão

x = valor medido em cada inspeção

n = quantidade de inspeções

i = Número de cada um dos valores medidos $\leq n$

A partir daí, como exemplo o frasco com 100% de água e iluminação a 90 graus, tem-se que:

$$\sigma = \sqrt{\frac{(164^2+187^2+165^2)}{3} - \left(\frac{164+187+165}{3}\right)^2} \quad (4)$$

$$\sigma = 10,614$$

Então, pode-se dizer que o valor medido pelo sistema, na escala de cinza, com relação ao frasco 1 (100% de água) e iluminação a 90 graus é de, em média 172 unidades, com variação média de 10,614 unidades para mais ou para menos. Por conseguinte, infere-se que o erro de precisão dessa inspeção é de:

$$\frac{10,614}{172} = 6,17\% \quad (5)$$

Levando esses cálculos para todas as outras inspeções da tabela, elaboraram-se as seguintes (Tabelas 5 e 6):

Tabela 5 – Desvio padrão e erro de precisão em três sequências de inspeção com iluminação a 90°

Ângulo de iluminação	90 graus			Desvio Padrão	Erro de precisão
	Inspeção	1ª	2ª		
Porcentagem de Café (em ml.)					
0	164	187	165	10,614	6,17%
5	91	85	88	2,449	2,78%
10	58	60	60	0,943	1,59%
15	58	58	60	0,943	1,61%
20	57	57	56	0,471	0,83%
25	53	54	54	0,471	0,88%
30	52	52	53	0,471	0,90%
35	49	50	51	0,816	1,63%

Fonte: Autoria própria

Tabela 6 – Desvio padrão e erro de precisão em três sequências de inspeção com iluminação a 180°

Ângulo de iluminação		180 graus			
Inspeção	1 ^a	2 ^a	3 ^a	Desvio Padrão	Erro de precisão
Porcentagem de Café (em ml.)					
0	197	201	196	2,16	1,09%
5	159	152	159	3,3	2,11%
10	96	94	97	1,247	1,30%
15	94	97	97	1,414	1,47%
20	94	90	91	1,7	1,85%
25	37	38	30	3,559	10,17%
30	11	16	15	2,16	15,43%
35	7	10	7	1,414	17,68%

Fonte: Autoria própria

Plotando-se os dados acima, gerou-se o seguinte gráfico (Gráfico 1):

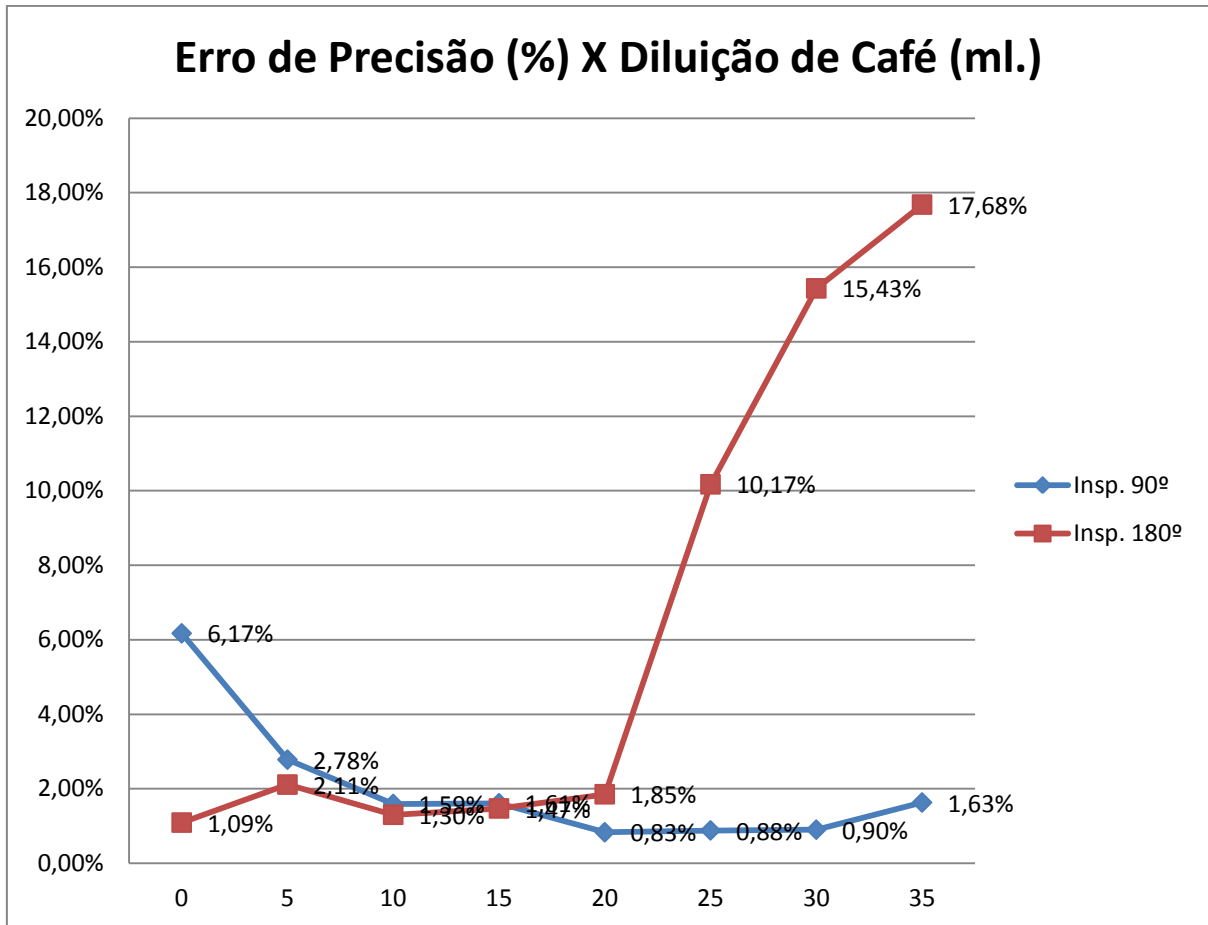


Gráfico 1 – Erro de Precisão (%) X Diluição de Café (ml.)

Fonte: Autoria própria

Do gráfico, é possível inferir um queda abrupta na precisão das inspeções a partir da solução com mais de 57% de café diluído, especificamente com relação ao método de iluminação a 180 graus, representada pela ascensão da linha vermelha (aumento do erro). Da mesma forma, a imprecisão também aumenta nas inspeções de líquidos muito límpidos no caso da iluminação a 90 graus, e especificamente em relação a soluções com menos de 14% de café diluído. Desta forma, a faixa de melhor confiabilidade das inspeções encontra-se entre 14% e 57% de café diluído em água.

5 CONSIDERAÇÕES FINAIS

Os possíveis ganhos à área técnica que esta pesquisa pode propiciar incluem, entre outros, a flexibilização dos sistemas de inspeção visual de peças usinadas ou manufaturadas. Isso porque, agregando a possibilidade de inspeção concomitante da turbidez de produtos líquidos a um sistema que já realiza a inspeção de usinagens através do software Vision Builder, reduzir-se-iam, também, os custos de uma possível expansão da linha de produção, caso fosse implantado um sistema dedicado de análise da turbidez dos produtos líquidos. Outro possível ganho à área técnica, e que motivou em grande parte a presente pesquisa, seria um baixo custo de implementação voltado às micro e pequenas empresas.

Para o problema apresentado na proposta de trabalho, a solução foi alcançada, pois as premissas se tornaram realidade. O sistema de fato funciona, e atingiu o seu objetivo de inspecionar líquidos contidos em frascos, e roteá-los em função dos resultados das inspeções. A comunicação entre os equipamentos funcionou satisfatoriamente dentro do esperado. Não obstante, ressalvas há que ser feitas. Após a apresentação e análise dos resultados, em se tratando da relação custo/benefício de um sistema honesto como este, há que se ponderar no que tange à necessidade específica do usuário deste sistema.

Apesar de o objetivo geral ter sido alcançado, não foi possível, por exemplo, recriar ou reconstruir um turbidímetro propriamente dito, pois que para isso seria necessário criar uma escala de referência para o sistema, isonômica a uma escala de unidades padrão de turbidez, como a UNT. Por conseguinte, para se aferir a turbidez de um líquido na métrica de uma unidade padrão, uma das opções seria a de primeiramente usar a substância formazina (abordado na fundamentação teórica), para que se pudesse estabelecer uma escala de medidas que traduzisse essa escala padrão. Porém, por não ter acesso a essa substância, nem mesmo perícia para manuseá-la, essa foi uma opção inviável para os autores. Uma segunda opção seria a de adquirir um aparelho turbidímetro para que pudesse mensurar a turbidez de algum líquido, em seguida colocar esse mesmo líquido para a inspeção do sistema aqui desenvolvido, e, a partir daí, elaborar uma escala análoga, isonômica, que desse condições ao sistema de calcular suas medições numa métrica padrão de turbidimetria. Nesse caso, porém, a aquisição de tal aparelho também se

mostrou inviável em função do alto custo. Ainda assim, por mais que o sistema não seja capaz de medir a turbidez em unidades padrão, ele é capaz de medi-la dentro de limites impostos por uma solução padrão, por exemplo, e que pode ser definida pelo próprio usuário, configurando o programa de inspeção no software Vision Builder.

Também é necessário verificar se as necessidades do possível usuário se enquadram dentro das margens de precisão do sistema, pois como foi detalhadamente descrito e analisado, existe uma faixa maior e menor de precisão para líquidos com turbidez próxima de zero ou do máximo, e no meio termo. Satisfeitas as necessidades desse possível usuário, com certeza o custo é um atrativo, uma vez que a aquisição do software Vision Builder, de um microcontrolador Arduino e periféricos básicos de IHM são muito mais baratos quando comparados ao investimento num CLP, num turbidímetro industrial, e ainda num software de supervisão além do Vision Builder, que com certeza também satisfariam a necessidade do usuário, mas que podem ser inviáveis no caso de uma micro ou pequena empresa em função do alto custo.

Todavia, o protótipo comporta maiores explorações em estudos futuros principalmente com relação às potencialidades do Vision Builder em aferir a turbidez em unidades padrão de turbidimetria, como a UNT – Unidade Nefelométrica de Turbidez, o que não foi possível até aqui. Também, ainda, como alternativa ao uso do Vision Builder, até por uma questão de custos da aquisição do software, pode-se desenvolver ou estudar o desenvolvimento de outro software de análise visual, baseado em bibliotecas de uso livre por exemplo.

REFERÊNCIAS

NAOUM, Paulo Francisco. **Métodos de Avaliação Laboratorial**. Disponível em: <<http://www.ciencianews.com.br/aulavirt/metodos.pdf>>. Acesso em: 10 nov. 2015.

SANTOS, Wagner Bezerra dos. **Construção e aplicação de um Turbidímetro / Nefelômetro Microcontrolado Portátil**. Disponível em: <http://www.bdtd.ufscar.br/htdocs/tedeSimplificado//tde_busca/arquivo.php?codArquivo=2798>. Acesso em: 11 nov. 2015.

SIMMS, R. **Forward scatter optical turbidimeter apparatus**. U.S. Patent n. 3,713,743, 30 jan. 1973.

TECNISIS. **Turbidímetro em Linha Turbiscat**. Disponível em: <<http://www.tecnisis.pt/produtos/detail.html?categoria=15&node=15%7C42%7C&ref=180&level=p>>. Acesso em: 09 nov. 2015.

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO. **Turbidez**. Disponível em: <<http://www.ufrj.br/institutos/it/de/acidentes/turb.htm>>. Acesso em: 07 nov. 2015a.

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO. **Como Fazer a Leitura**. Disponível em: <<http://www.ufrj.br/institutos/it/de/acidentes/nef.htm>>. Acesso em: 07 nov. 2015b.

ZIEGLER, Andrew C. Issues related to use of turbidity measurements as a surrogate for suspended sediment. In: **Turbidity and other sediment surrogates workshop**. 2002.

APÊNDICE – Transcrição integral do programa em linguagem C++ feito para o microcontrolador Arduino

```
#include <LiquidCrystal.h> // include the library code
LiquidCrystal lcd(40, 38, 36, 34, 32, 30, 28, 26, 24, 22);
```

```
char teclado()
{
  digitalWrite(43,LOW);
  digitalWrite(47,LOW);
  digitalWrite(51,LOW);
  if(digitalRead(41)==LOW)
  {
    if(digitalRead(41)==LOW)
    {
      digitalWrite(51,HIGH);
      if(digitalRead(41)==LOW)
      {
        return '0';
      }
    }
    else
    {
      digitalWrite(47, HIGH);
      digitalWrite(51, LOW);
      if(digitalRead(41)==LOW)
      {
        return '#';
      }
    }
  }
}
if(digitalRead(45)==LOW)
{
  digitalWrite(47,HIGH);
  digitalWrite(51,HIGH);
  if(digitalRead(45)==LOW)
  {
    return '7';
  }
}
else
{
  digitalWrite(43,HIGH);
  digitalWrite(47,LOW);
  if(digitalRead(45)==LOW)
  {
    return '8';
  }
}
else
{
  digitalWrite(47, HIGH);
```

```
    digitalWrite(51, LOW);
    if(digitalRead(45)==LOW)
    {
        return '9';
    }
}
}
}
if(digitalRead(49)==LOW)
{
    digitalWrite(47,HIGH);
    digitalWrite(51,HIGH);
    if(digitalRead(49)==LOW)
    {
        return '4';
    }
    else
    {
        digitalWrite(43,HIGH);
        digitalWrite(47,LOW);
        if(digitalRead(49)==LOW)
        {
            return '5';
        }
        else
        {
            digitalWrite(47, HIGH);
            digitalWrite(51, LOW);
            if(digitalRead(49)==LOW)
            {
                return '6';
            }
        }
    }
}
if(digitalRead(53)==LOW)
{
    digitalWrite(47,HIGH);
    digitalWrite(51,HIGH);
    if(digitalRead(53)==LOW)
    {
        return '1';
    }
    else
    {
        digitalWrite(43,HIGH);
        digitalWrite(47,LOW);
        if(digitalRead(53)==LOW)
        {
            return '2';
        }
    }
}
```

```

    }
    else
    {
        digitalWrite(47, HIGH);
        digitalWrite(51, LOW);
        if(digitalRead(53)==LOW)
        {
            return '3';
        }
    }
}
else
{
    digitalWrite(39,LOW);
    if(digitalRead(37)==LOW)
    {
        return '*';
    }
}
return 0;
}

int scanf()
{
    int memory = 0;
    while(memory == 0)
    {
        memory = teclado();
        if(memory != 0)
        {
            delay(50);
            if(teclado() != memory)
            {
                memory = 0;
            }
        }
    }
    while(teclado() != 0)
    {
        ;
    }
    return memory;
}

void motorpasso(int passo, int veloc)
{
    if(passo==1)
    {
        digitalWrite(5, LOW);
        digitalWrite(3, HIGH);
    }
}

```

```
digitalWrite(4, HIGH);
digitalWrite(2, LOW);
delay(veloc);
}
if(passo==2)
{
digitalWrite(5, LOW);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
digitalWrite(2, HIGH);
delay(veloc);
}
if(passo==3)
{
digitalWrite(5, LOW);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
digitalWrite(2, HIGH);
delay(veloc);
}
if(passo==4)
{
digitalWrite(5, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
digitalWrite(2, HIGH);
delay(veloc);
}
if(passo==5)
{
digitalWrite(5, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
digitalWrite(2, HIGH);
delay(veloc);
}
if(passo==6)
{
digitalWrite(5, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);
digitalWrite(2, HIGH);
delay(veloc);
}
if(passo==7)
{
digitalWrite(5, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);
digitalWrite(2, LOW);
```



```

    delay(veloc);
}
if(passo==8)
{
    digitalWrite(5, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(2, LOW);
    delay(veloc);
}
}

int terco, passo, frasco;
void gira(char sentido[12])
{
    int x;
    if(sentido=="horario")
    {
        if(terco==1)
        {
            for(x=0;x<25;x++) // inicia no passo x e termina no passo x+1
            {
                passo++;
                if(passo>8)
                {
                    passo=1;
                }
                motorpasso(passo, 16);
            }
        }
        if(terco==2) //inicia no passo x e termina no passo x+1
        {
            for(x=0;x<25;x++)
            {
                passo++;
                if(passo>8)
                {
                    passo=1;
                }
                motorpasso(passo, 16);
            }
        }
        if(terco==3) // inicia no passo x e termina no passo x
        {
            for(x=0;x<24;x++)
            {
                passo++;
                if(passo>8)
                {
                    passo=1;
                }
            }
        }
    }
}

```

```
    }
    motorpasso(passo, 16);
  }
}
terco--;
if(terco<1)
{
  terco=3;
}
frasco++;
if(frasco>8)
{
  frasco=1;
}
}
if(sentido=="antihorario")
{
  terco++;
  if(terco>3)
  {
    terco=1;
  }
  if(terco==1)
  {
    for(x=0;x<25;x++)
    {
      passo--;
      if(passo<1)
      {
        passo=8;
      }
      motorpasso(passo, 16);
    }
  }
}
if(terco==2)
{
  for(x=0;x<25;x++)
  {
    passo--;
    if(passo<1)
    {
      passo=8;
    }
    motorpasso(passo, 16);
  }
}
if(terco==3)
{
  for(x=0;x<24;x++)
  {
```

```

    passo--;
    if(passo<1)
    {
        passo=8;
    }
    motorpasso(passo, 16);
}
}
frasco--;
if(frasco<1)
{
    frasco=8;
}
}
}

void leds(char state[8], int fileira)
{
    if(fileira==90)
    {
        if(state=="liga")
        {
            digitalWrite(8, HIGH);
            digitalWrite(9, LOW);
        }
        if(state=="desliga")
        {
            digitalWrite(8, LOW);
            digitalWrite(9, LOW);
        }
    }
    if(fileira==180)
    {
        if(state=="liga")
        {
            digitalWrite(10, HIGH);
            digitalWrite(11, LOW);
        }
        if(state=="desliga")
        {
            digitalWrite(10, LOW);
            digitalWrite(11, LOW);
        }
    }
}
}
}

```

```

String autoriz, inspResult90, inspResult180;
int chave;
String inspecionar()
{

```

```

if(chave==9)
{
  inspResult90="";
  inspResult180="";
  String result90, result180, resultfinal;
  result90="";
  result180="";
  resultfinal="";
  leds("liga",90);
  delay(100);
  autoriz="QRV";
  Serial.println(autoriz);
  delay(50);
  Serial.println("");
  lcd.clear();
  lcd.setCursor(0, 0);      // set the cursor to colum 0, line 0
  lcd.print("INSPECIONANDO");
  while(inspResult90!="8482856913" && inspResult90!="706576836913")
  {
    while(Serial.available())
    {
      inspResult90 += Serial.read();
    }
  }
  delay(200);
  if(inspResult90=="8482856913")
  {
    result90="APROVADO EM 90";
    lcd.clear();
    lcd.setCursor(0, 0);      // set the cursor to colum 0, line 0
    lcd.print("Frasco");
    lcd.setCursor(7, 0);      // set the cursor to colum 0, line 0
    lcd.print(frasco);
    lcd.setCursor(0, 1);      // set the cursor to colum 0, line
    lcd.print(result90);
    inspResult90="";
  }
  if(inspResult90=="706576836913")
  {
    result90="REPROVADO EM 90";
    lcd.clear();
    lcd.setCursor(0, 0);      // set the cursor to colum 0, line 0
    lcd.print("Frasco");
    lcd.setCursor(7, 0);      // set the cursor to colum 0, line 0
    lcd.print(frasco);
    lcd.setCursor(0, 1);      // set the cursor to colum 0, line
    lcd.print(result90);
    inspResult90="";
  }
  }
  delay(3000);

```

```

leds("desliga",90);          // TROCA DE LEDS 90 --> 180
leds("liga",180);
delay(100);
autoriz="QRV";
Serial.println(autoriz);
delay(50);
Serial.println("");
lcd.clear();
lcd.setCursor(0, 0);        // set the cursor to colum 0, line 0
lcd.print("INSPECIONANDO");
while(inspResult180!="8482856913" && inspResult180!="706576836913")
{
  while(Serial.available())
  {
    inspResult180 += Serial.read();
  }
}
delay(1000);
if(inspResult180=="8482856913")
{
  result180="APROVADO EM 180";
  lcd.clear();
  lcd.setCursor(0, 0);      // set the cursor to colum 0, line 0
  lcd.print("Frasco");
  lcd.setCursor(7, 0);     // set the cursor to colum 0, line 0
  lcd.print(frasco);
  lcd.setCursor(0, 1);     // set the cursor to colum 0, line
  lcd.print(result180);
  inspResult180="";
}
if(inspResult180=="706576836913")
{
  result180="REPROVADO EM 180";
  lcd.clear();
  lcd.setCursor(0, 0);     // set the cursor to colum 0, line 0
  lcd.print("Frasco");
  lcd.setCursor(7, 0);     // set the cursor to colum 0, line 0
  lcd.print(frasco);
  lcd.setCursor(0, 1);     // set the cursor to colum 0, line 0
  lcd.print(result180);
  inspResult180="";
}
delay(3000);
leds("desliga",180);       // DESLIGA OS LEDS 180
if(result90=="APROVADO EM 90" && result180=="APROVADO EM 180")
{
  resultfinal="APROVADO";
}
else
{

```

```

        resultfinal="REPROVADO";
    }
    return resultfinal;
}
}

```

```

String result1, result2, result3, result4, result5, result6, result7, result8;
void arqResult(String result)

```

```

{
    if(frasco==1)
    {
        result1=result;
    }
    if(frasco==2)
    {
        result2=result;
    }
    if(frasco==3)
    {
        result3=result;
    }
    if(frasco==4)
    {
        result4=result;
    }
    if(frasco==5)
    {
        result5=result;
    }
    if(frasco==6)
    {
        result6=result;
    }
    if(frasco==7)
    {
        result7=result;
    }
    if(frasco==8)
    {
        result8=result;
    }
}

```

```

String aprovados;
void relacionaAprovados()
{
    if(result1=="APROVADO")
    {
        aprovados+=1;
    }
}

```

```
if(result2=="APROVADO")
{
    aprovados+=2;
}
if(result3=="APROVADO")
{
    aprovados+=3;
}
if(result4=="APROVADO")
{
    aprovados+=4;
}
if(result5=="APROVADO")
{
    aprovados+=5;
}
if(result6=="APROVADO")
{
    aprovados+=6;
}
if(result7=="APROVADO")
{
    aprovados+=7;
}
if(result8=="APROVADO")
{
    aprovados+=8;
}
}
```

String reprovados;

```
void relacionaReprovados()
{
    if(result1=="REPROVADO")
    {
        reprovados+=1;
    }
    if(result2=="REPROVADO")
    {
        reprovados+=2;
    }
    if(result3=="REPROVADO")
    {
        reprovados+=3;
    }
    if(result4=="REPROVADO")
    {
        reprovados+=4;
    }
    if(result5=="REPROVADO")
```

```

{
  reprovados+=5;
}
if(result6=="REPROVADO")
{
  reprovados+=6;
}
if(result7=="REPROVADO")
{
  reprovados+=7;
}
if(result8=="REPROVADO")
{
  reprovados+=8;
}
}

void repeticao()
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Frasco");
  lcd.setCursor(7, 0);
  lcd.print(frasco);
  lcd.setCursor(7, 1);
}

void setup()
{
  lcd.begin(16, 2);          // set up the LCD's number of columns and rows:
  pinMode(37, INPUT);      //Linha do asterisco
  pinMode(39, OUTPUT);     //Coluna do asterisco
  pinMode(41, INPUT);      //Linha 0
  pinMode(43, OUTPUT);     //Coluna 0
  pinMode(45, INPUT);      //Linha 1
  pinMode(47, OUTPUT);     //Coluna 1
  pinMode(49, INPUT);      //Linha 2
  pinMode(51, OUTPUT);     //Coluna 2
  pinMode(53, INPUT);      //Linha 3
  pinMode(13, OUTPUT);     //led da placa
  pinMode(11, OUTPUT);     //bit para o LED 180 graus
  pinMode(10, OUTPUT);     //bit para o LED 180 graus
  pinMode(9, OUTPUT);      //bit para o LED 90 graus
  pinMode(8, OUTPUT);      //bit para o LED 90 graus
  pinMode(5, OUTPUT);      //bit para o motor de passo
  pinMode(4, OUTPUT);      //bit para o motor de passo
  pinMode(3, OUTPUT);      //bit para o motor de passo
  pinMode(2, OUTPUT);      //bit para o motor de passo
  digitalWrite(37, HIGH);
  digitalWrite(41, HIGH);
}

```



```

digitalWrite(45, HIGH);
digitalWrite(49, HIGH);
digitalWrite(53, HIGH);
digitalWrite(10, LOW);
digitalWrite(8, LOW);
digitalWrite(7, LOW);
digitalWrite(6, LOW);
digitalWrite(5, LOW);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
digitalWrite(2, LOW);
passo=1;
motorpasso(passo, 20);
terco=3;
frasco=0;
chave=0;
inspResult90="";
inspResult180="";
lcd.clear();
lcd.setCursor(1, 0);      // set the cursor to colum 1, line 0
lcd.print(" UTFPR");
delay(2000);
lcd.clear();
lcd.setCursor(0, 0);     // set the cursor to colum 0, line 0
lcd.print("Trabalho de");
delay(1250);
lcd.setCursor(0, 1);    // set the cursor to colum 0, line 1
lcd.print("  Diplomacao");
delay(2000);
Serial.begin(9600);
lcd.clear();
lcd.setCursor(0, 0);     // set the cursor to colum 0, line 0
lcd.print("Qual o frasco na");
lcd.setCursor(0, 1);    // set the cursor to colum 0, line 1
lcd.print("posicao inicial?");

while(frasco!=1&&frasco!=2&&frasco!=3&&frasco!=4&&frasco!=5&&frasco!=6&&frasco!
=7&&frasco!=8)
{
  frasco=scanf();
  frasco=frasco-48; //TRANSFORMA DE CARACTER ASCII P/ NUMERO DE 0 A 9
}
lcd.clear();
lcd.setCursor(0, 0);     // set the cursor to colum 0, line 0
lcd.print(frasco);
}
void loop()
{
  String buf;
  int x, modo, opcao;

```

```

modo=0;
opcao=0;
buf="";
autoriz="";
aprovados="";
reprovados="";
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);      // set the cursor to colum 0, line 0
lcd.print("Selecione o");
lcd.setCursor(0, 1);     // set the cursor to colum 0, line 1
lcd.print("modo de inspecao");
delay(3000);
lcd.clear();
lcd.setCursor(0, 0);     // set the cursor to colum 0, line 0
lcd.print("1 - Avulso");
lcd.setCursor(0, 1);     // set the cursor to colum 0, line 1
lcd.print("2 - Continuo");
while(modo!=1&&modo!=2)
{
  modo=scanf();
  modo=modo-48;
}
if(modo==1)
{
  modo=0;
  lcd.clear();
  lcd.setCursor(0, 0);   // set the cursor to colum 0, line 0
  lcd.print("1 e 3 - Gira");
  lcd.setCursor(0, 1);   // set the cursor to colum 0, line 1
  lcd.print("5 - Confirma");
  while(opcao!=5)
  {

    opcao=scanf();
    opcao=opcao-48;
    if(opcao==1)
    {
      gira("horario");
    }
    if(opcao==3)
    {
      gira("antihorario");
    }
  }
  opcao=0;
  chave=9;
  buf=inspeccionar();
  chave=0;
  arqResult(buf);
}

```

```

lcd.clear();
lcd.setCursor(0, 0);      // set the cursor to colum 0, line 0
lcd.print("Resultado da");
lcd.setCursor(0, 1);     // set the cursor to colum 0, line 1
lcd.print("inspec:");
lcd.setCursor(7, 1);     // set the cursor to colum 7, line 1
lcd.print(buf);
delay(3000);
}
if(modos==2)
{
  modo=0;
  result1="";
  result2="";
  result3="";
  result4="";
  result5="";
  result6="";
  result7="";
  result8="";
  for(x=0;x<8;x++)
  {
    delay(500);
    chave=9;
    buf=inspeccionar();
    chave=0;
    arqResult(buf);
    lcd.clear();
    lcd.setCursor(0, 0);   // set the cursor to colum 0, line 0
    lcd.print("Resultado da");
    lcd.setCursor(0, 1);  // set the cursor to colum 0, line 1
    lcd.print("inspec:");
    lcd.setCursor(7, 1);  // set the cursor to colum 7, line 1
    lcd.print(buf);
    gira("horario");
    delay(2000);
  }
  delay(1000);
  lcd.clear();
  lcd.setCursor(3, 0);    // set the cursor to colum 3, line 0
  lcd.print("RESULTADO");
  lcd.setCursor(5, 1);    // set the cursor to colum 5, line 1
  lcd.print("FINAL");
  delay(2500);
  lcd.clear();
  lcd.setCursor(0, 0);    // set the cursor to colum 0, line 0
  lcd.print("APROVADOS");
  lcd.setCursor(0, 1);    // set the cursor to colum 0, line 1
  lcd.print("FRASCOS");
  relacionaAprovados();
}

```

```

lcd.setCursor(8, 1);      // set the cursor to colum 8, line 1
lcd.print(aprovados);
delay(4000);
lcd.clear();
lcd.setCursor(0, 0);     // set the cursor to colum 0, line 0
lcd.print("REPROVADOS");
lcd.setCursor(0, 1);     // set the cursor to colum 0, line 1
lcd.print("FRASCOS");
relacionaReprovados();
lcd.setCursor(8, 1);     // set the cursor to colum 8, line 1
lcd.print(reprovados);
delay(4000);
lcd.clear();
while(frasco>1)
{
  gira("horario");
}
lcd.setCursor(0, 0);     // set the cursor to colum 0, line 0
lcd.print("Roteando");
lcd.setCursor(7, 1);     // set the cursor to colum 7, line 1
lcd.print("Aprovados");
delay(2500);
if(result1=="APROVADO")
{
  repeticao();
  lcd.print(result1);
  delay(3000);
}
gira("horario");
if(result2=="APROVADO")
{
  repeticao();
  lcd.print(result2);
  delay(3000);
}
gira("horario");
if(result3=="APROVADO")
{
  repeticao();
  lcd.print(result3);
  delay(3000);
}
gira("horario");
if(result4=="APROVADO")
{
  repeticao();
  lcd.print(result4);
  delay(3000);
}
gira("horario");

```

```
if(result5=="APROVADO")
{
  repeticao();
  lcd.print(result5);
  delay(3000);
}
gira("horario");
if(result6=="APROVADO")
{
  repeticao();
  lcd.print(result6);
  delay(3000);
}
gira("horario");
if(result7=="APROVADO")
{
  repeticao();
  lcd.print(result7);
  delay(3000);
}
gira("horario");
if(result8=="APROVADO")
{
  repeticao();
  lcd.print(result8);
  delay(3000);
}
lcd.clear();
lcd.setCursor(0, 0);      // set the cursor to colum 0, line 0
lcd.print("Roteando");
lcd.setCursor(6, 1);     // set the cursor to colum 6, line 1
lcd.print("Reprovados");
delay(2500);
if(result8=="REPROVADO")
{
  repeticao();
  lcd.print(result8);
  delay(3000);
}
gira("antihorario");
if(result7=="REPROVADO")
{
  repeticao();
  lcd.print(result7);
  delay(3000);
}
gira("antihorario");
if(result6=="REPROVADO")
{
  repeticao();
```

```
    lcd.print(result6);
    delay(3000);
  }
  gira("antihorario");
  if(result5=="REPROVADO")
  {
    repeticao();
    lcd.print(result5);
    delay(3000);
  }
  gira("antihorario");
  if(result4=="REPROVADO")
  {
    repeticao();
    lcd.print(result4);
    delay(3000);
  }
  gira("antihorario");
  if(result3=="REPROVADO")
  {
    repeticao();
    lcd.print(result3);
    delay(3000);
  }
  gira("antihorario");
  if(result2=="REPROVADO")
  {
    repeticao();
    lcd.print(result2);
    delay(3000);
  }
  gira("antihorario");
  if(result1=="REPROVADO")
  {
    repeticao();
    lcd.print(result1);
    delay(3000);
  }
}
}
```