

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**CURSO DE AUTOMAÇÃO INDUSTRIAL**

**RICARDO FERNANDES CUNHA**

**MICRO AUTOMAÇÃO E CONTROLE DE TANQUE DE HIDRÓXIDO DE  
SÓDIO A 3% UTILIZANDO O ARDUINO UNO**

**TRABALHO DE CONCLUSÃO DE CURSO**

**CORNÉLIO PROCÓPIO**

**2014**



**RICARDO FERNANDES CUNHA**

**MICRO AUTOMAÇÃO E CONTROLE DE TANQUE DE HIDRÓXIDO DE  
SÓDIO A 3% UTILIZANDO O ARDUINO UNO**

Trabalho de conclusão de curso de graduação apresentado à disciplina de Trabalho de Diplomação de Curso Superior de Automação Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Luiz Marcelo Chiesse da Silva

**CORNÉLIO PROCÓPIO**

**2014**

**RICARDO FERNANDES CUNHA**

**MICRO AUTOMAÇÃO E CONTROLE DE TANQUE DE HIDRÓXIDO  
DE SÓDIO A 3% UTILIZANDO O ARDUINO UNO**

Trabalho de conclusão de curso apresentado às 18:30h do dia 24 de julho de 2014 como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof.(a) Dr.(a) Luiz Marcelo Chiesse da Silva  
Professor(a) Orientador(a)  
UTFPR/ Campus Cornélio Procópio

---

Prof.(a) Ms.Rafael Rodrigues da Silva  
Professor(a) Convidado(a)  
UTFPR/ Campus Cornélio Procópio

---

Prof.(a) Ms. Ângelo Feracin Neto  
Professor(a) Convidado(a)  
UTFPR/ Campus Cornélio Procópio

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



# MICRO AUTOMAÇÃO E CONTROLE DE TANQUE DE HIDRÓXIDO DE SÓDIO A 3% UTILIZANDO O ARDUINO UNO

## RESUMO

CUNHA, Ricardo Fernandes. **Micro automação e controle de tanque de hidróxido de sódio a 3% utilizando o Arduino uno.** 2014. N° de folhas 50. Trabalho de conclusão de curso. Tecnologia em Automação Industrial - Universidade tecnológica Federal do Paraná.

Esta monografia apresenta o projeto de automação do controle de um dos processos necessários para produção industrial do café solúvel, denominado comumente de “solução de hidróxido de sódio a 3%”, comum nas empresas alimentícias. Este projeto objetiva a obtenção das medidas das principais grandezas físicas deste processo industrial, utilizando o microcontrolador Arduino Uno para controlar o nível e temperatura da solução, além dos atuadores e um display de *LCD*, implementado em um painel sinóptico, para a automação do sistema.

**Palavras chave:** Controle. Arduino. Nível. Temperatura. Hidróxido de sódio.

## ABSTRACT

### **3% SODIUM HYDROXIDE TANK CONTROL AND MICRO AUTOMATION USING THE ARDUINO UNO**

This work presents a process control automation project from one of soluble coffee production processes, commonly called 3% sodium hydroxide solution, commonly used in food industry. This project aims obtain the main physical entities measures from this industrial process, using the Arduino Uno microcontroller to level and temperature solution control, beyond the actuators and a *LDC* display, implemented in a synoptic panel, for the system automation.

**Keywords:** Control. Arduino. Level. Temperature. Sodium Hydroxide.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Fluxograma do processo de recebimento do hidróxido de sódio na fábrica..	18
Figura 2 - Identificação dos componentes da placa.....	21
Figura 3 - Circuito integrado ATMEGA328. ....	24
Figura 4 - Mapeamento da pinagem do ATMEGA 328.....	25
Figura 5 - Protótipo desenvolvido. ....	27
Figura 6 - Fluxograma do processo. ....	28
Figura 7 – Fluxograma do processo. ....	28
Figura 8 - Excitação das bobinas do motor monofásico. ....	29
Figura 9 - Válvula solenoide utilizada no protótipo.....	30
Figura 10 - Sensor ultrassônico HC-SR04. ....	31
Figura 11 - Gráfico de sequência de disparo do sensor.....	31
Figura 12 - Esquema de ligação do sensor ultrassônico.....	32
Figura 13 - Sensor de temperatura LM35.....	32
Figura 14 - Esquema de ligação do LM35 .....	33
Figura 15 - Diagrama de instrumentação do projeto.....	34
Figura 16 - Módulo relé utilizado no projeto. ....	35
Figura 17 - Esquema de ligação do módulo relé 12V. ....	36
Figura 18 - Display LCD 1602A.....	36
Figura 19 - Esquema de ligação do LCD no Arduino Uno. ....	37
Figura 20 - Ambiente de desenvolvimento do Arduino Uno. ....	38
Figura 21 - Diagrama esquemático de controle dos componentes.....	39
Figura 22 - Instalação do sensor ultrassônico no vaso comunicante.....	40
Figura 23 - Módulo relé acionando bobinas dos contatores.....	41
Figura 24 - Esquema elétrico dos atuadores trifásicos.....	41
Figura 25 - Esquema de acionamento das bobinas solenoides.....	42



## LISTA DE ACRÔNIMOS

IDE – Ambiente de Desenvolvimento Integrado.

USB-Universal Serial Bus.

LED - Diodo Emissor de Luz.

LCD – Tela de Cristal Líquido.

ISO – Organização Internacional de Padrões.

PIC – Controlador de Interface Programável.

EEPROM – Memória Programável Apagável Eletricamente.

FTDI – Dispositivo Internacional de Tecnologia do Futuro.

RISC - Computador com Conjunto Reduzido de Instruções.

SPI – Interface Periférica Serial.

USART – Transmissor Receptor Síncrono e Assíncrono Universal.

SD – Segurança Digital.

PWM – Modulação por Largura de Pulso.

PVC – Policloreto de Vinil.

AC – Corrente Alternada.

DC – Corrente Contínua.

ICSP – In-Circuit System Programming.

## SUMÁRIO

1 – INTRODUÇÃO.....	12
1.1 - APRESENTAÇÃO DO TRABALHO.....	12
1.2 - APRESENTAÇÃO DO PROBLEMA.....	13
1.3 - JUSTIFICATIVA.....	14
2 – OBJETIVO.....	14
2.1 - OBJETIVO GERAL.....	14
2.2 - OBJETIVO ESPECÍFICO.....	15
3 – MÉTODOS DE PESQUISA.....	16
4 – FUNDAMENTAÇÃO TEÓRICA.....	17
4.1 - FUNCIONAMENTO ATUAL DO PROCESSO DE LIMPEZA DOS EQUIPAMENTOS DA CAFÉ IGUAÇU.....	17
4.2 - MOTIVOS PARA ESCOLHA DO ARDUINO UNO.....	19
4.3 - O MICROCONTROLADOR.....	19
4.4 - ESTRUTURAS DOS COMANDOS DE PROGRAMAÇÃO DO ARDUINO. .....	21
4.5 - CI ATMEL ATMEGA 328.....	24
4.6 - DESENVOLVIMENTO DO PROTÓTIPO.....	25
4.7 - FLUXOGRAMA DO PROJETO.....	28
4.8 - MOTORES MONOFÁSICOS AC.....	28
4.9 - VÁLVULA SOLÉNOIDE.....	30
4.10 - O SENSOR ULTRASSÔNICO HC-SR04.....	30
4.11 - SENSOR DE TEMPERATURA LM35.....	32
4.12 - PAINEL SINÓPTICO.....	34
4.13 - MÓDULO RELÉ 4 CANAIS 12V.....	35
4.14 - DISPLAY DE LCD.....	36
4.15 - AMBIENTE DE DESENVOLVIMENTO INTEGRADO.....	38
4.16 - CÓDIGO INSERIDO NO ARDUINO.....	38
4.17 - APRESENTAÇÃO DO CIRCUITO ELÉTRICO DO SISTEMA.....	39
5 – CONSIDERAÇÕES FINAIS.....	43
6 – REFERÊNCIAS.....	44
ANEXO A – Código Desenvolvido para o projeto.....	46



## **1 – INTRODUÇÃO.**

### **1.1 - APRESENTAÇÃO DO TRABALHO.**

O trabalho proposto apresenta um projeto utilizando o microcontrolador Arduino Uno para o controle de um sistema industrial da Cia. Iguçu de Café Solúvel. O sistema a ser controlado consiste em um tanque industrial que recebe hidróxido de sódio a 3%, da central de produtos químicos da própria empresa, no qual o microcontrolador Arduino Uno controla a temperatura do hidróxido de sódio em aproximadamente 55°C através de um sistema de aquecimento, além do controle do nível da solução no interior do tanque, mantendo-a em um *set point* determinado, e do controle da entrada e saída do fluido, acionando motores que são necessários para obtenção da solução em condições aceitáveis. Esta solução é utilizada para a limpeza dos equipamentos e tubulações de escoamento do extrato de café, como pasteurizadores, resfriadores e demais equipamentos para a manufatura deste produto, aumentando a eficiência da produção de café solúvel devido à diminuição de intervenções mecânicas.

O aquecimento do hidróxido de sódio a 3% é necessário para que este produto químico tenha ação efetiva na limpeza nas tubulações de inox e equipamentos, evitando o progressivo acúmulo de resíduos do extrato de café, o que pode causar entupimentos. De acordo com a ficha técnica de informações de segurança deste produto químico (FISPQ), a temperatura da solução acima de 60°C inicia um processo de corrosão das tubulações e equipamentos, como tanques de armazenamento, constituídos na grande parte de aço inoxidável (inox), ocasionando principalmente vazamentos nas juntas e tubulações (FICHA TÉCNICA DE SEGURANÇA DO PRODUTO QUIMICO, 2014).

Se esta solução está sujeita a uma temperatura ambiente em torno de 35°C, não se obtém uma limpeza eficiente, causando desperdício deste produto e entupimento progressivo das tubulações. Portanto, a automação deste processo é de fundamental importância na obtenção do produto final (café solúvel), principalmente no aspecto financeiro, pois se trata de um produto de valor econômico considerável e que necessita de um tratamento correto para minimizar prejuízos e gastos. A utilização de um microcontrolador substitui a necessidade do projeto e implementação de sistemas de

automação de alto custo, principalmente em um processo simples, mas de fundamental importância para empresa.

Este projeto abrange o desenvolvimento de um protótipo microcontrolado de um sistema e de um painel sinóptico para controle e visualização do nível e temperatura da solução contida no interior do tanque, através de um display *LCD*, bem como a visualização do acionamento de todos os atuadores do sistema através de *led,s* indicativos, e alarmes de segurança. O objetivo é sintetizar todo o processo necessário, para o aquecimento e armazenagem do produto de modo que a solução de hidróxido de sódio esteja à disposição para utilização nos equipamentos de obtenção do café solúvel.

## **1.2 - APRESENTAÇÃO DO PROBLEMA.**

Controle de um processo industrial da empresa Cia. Iguazu de Café Solúvel, utilizando o microcontrolador Arduino Uno. Este objetiva controlar a entrada e saída, bem como o nível e a temperatura próximo de 55°C de uma solução denominada de hidróxido de sódio a 3%, ou seja, o hidróxido de sódio é diluído em água, para ser efetivamente utilizado nos equipamentos da fábrica, pois sua constituição original é 50% de hidróxido de sódio e 50% de água. O principal objetivo deste composto no processo de obtenção do café solúvel é a limpeza das tubulações e equipamentos, como pasteurizadores e resfriadores, por onde passa o extrato de café.

A necessidade de se utilizar hidróxido de sódio 3% se deve ao fato de que o extrato de café acumula-se no interior das tubulações de inox, criando incrustações, devido a altas temperaturas e à pressurização no interior dos equipamentos, causando em função do tempo, o entupimento progressivo. Desta forma, adota-se o procedimento de executar a limpeza dos dutos e equipamentos de processamento do extrato de café com a solução de hidróxido de sódio a 3%, após um período de produção.

O microcontrolador Arduino Uno é muito utilizado em aplicações residenciais, tais como acionamento automático de lâmpadas ou sistemas de monitoramento por celulares. Este trabalho consiste na utilização do Arduino Uno em uma aplicação industrial, onde as variáveis como nível, temperatura, abertura e fechamento de válvulas

e acionamento de motor e alarmes de segurança são processos comuns nas indústrias petroquímicas e alimentícias.

### **1.3 - JUSTIFICATIVA**

Com o aumento da necessidade de tornar os ambientes industriais mais confiáveis, com relação à medição de variáveis e ao controle automático, com o menor custo aplicado, é possível observar que o Arduino Uno é uma alternativa econômica, flexível e de fácil implementação para o controle de processos industriais apresentando ótima estabilidade e confiabilidade. A reprodução de um sistema supervisório utilizando o painel sinóptico é uma forma simples de supervisão, porém eficiente e de baixo custo para atender projetos de automação, como o descrito nesta proposta.

Este projeto abrange o conteúdo de diversas disciplinas ministradas durante o curso de tecnologia da Automação Industrial, sendo alguns conteúdos relacionados diretamente ao tema proposto, tais como, sistemas digitais, fundamentos de circuitos elétricos, eletrônica de potência, instrumentação e controle de processos industriais. Além disso, os aprimoramentos tecnológicos na área de microcontroladores e a criação de plataformas open-source viabilizaram o desenvolvimento, implementação e a melhoria de projetos de automação residencial ou industrial como esse.

## **2 – OBJETIVO.**

### **2.1 - OBJETIVO GERAL.**

Utilização do microcontrolador Arduino Uno no controle de um sistema industrial para a limpeza de dutos e equipamentos destinados à produção de café solúvel (indústria alimentícia). O sistema a ser controlado consiste em um tanque de recebimento de hidróxido de sódio (soda cáustica) a 3%, que provem da central de produtos químicos das instalações de uma fábrica de café solúvel. O controle deve ser executado para a utilização do hidróxido de sódio, que dessa forma obtém-se melhor eficiência, com seu respectivo aquecimento a uma temperatura de 55°C.

Este processo é necessário para a limpeza por onde passa o extrato de café, que no caso a vários equipamentos que precisam deste tratamento. Caso a temperatura do hidróxido de sódio seja superior a 60°C, inicia-se um processo de corrosão nos dutos de inox devido à reação da solução com o mesmo, causando problemas como vazamentos nas tubulações e nos equipamentos do processo de fabricação do extrato de café solúvel. Já o contrário, quando o hidróxido de sódio trabalha dentro dos equipamentos na temperatura ambiente, ocorre baixa eficiência na limpeza das tubulações, como consequência há o desperdício de material. Esta limpeza ocorre de acordo com a norma regulamentadora ISO22.000 de segurança para a produção de alimentos industrializados. Além do controle de aquecimento, este projeto, mostra que o Arduino Uno é capaz de controlar outra variável como nível, e o acionamento de atuadores como motores e válvulas solenoides.

## **2.2 - OBJETIVO ESPECÍFICO.**

- Estudar o processo de aquecimento na indústria;
- Buscar referencias bibliográficas sobre microcontrolador Arduino Uno;
- Desenvolver protótipo;
- Desenvolver o painel sinóptico;
- Sintetizar o processo de aquecimento no painel sinóptico;
- Desenvolver lógica de programação para efetuar o controle do sistema;
- Dimensionar circuitos;
- Efetuar testes no protótipo;
- Mostrar a necessidade de atenção neste tipo de processo.

### **3 – MÉTODOS DE PESQUISA**

Este projeto será dividido em três etapas.

1º Etapa: nesta etapa será executada a pesquisa de campo pela observação dos sistemas industriais que exigem o controle de nível e temperatura e acionamento de válvulas e acionamento de motores elétricos e visualização por sinóptico em seus processos para o controle da produção de um produto ou subproduto;

2º Etapa: na segunda etapa do projeto, será efetuada a pesquisa bibliográfica, a fim de buscar informações importantes para o desenvolvimento do projeto, corroborando para completo entendimento de funcionamento dos sensores, atuadores e motores elétricos bem como o microcontrolador que será necessário dispor no sistema;

3º Etapa: compreende os testes do software criado no ambiente de desenvolvimento do microcontrolador Arduino Uno. Será verificado a funcionalidade do protótipo e também as observações e testes finais no mesmo. Após isso, será testado o painel sinóptico e todas as condições e informações que devem conter no painel, para enfim controlar as variáveis necessárias para a obtenção da solução com características aceitáveis principalmente nível e temperatura.



## **4 – FUNDAMENTAÇÃO TEÓRICA.**

### **4.1 - FUNCIONAMENTO ATUAL DO PROCESSO DE LIMPEZA DOS EQUIPAMENTOS DA CAFÉ IGUAÇU.**

Atualmente o processo de limpeza dos dutos de inox e equipamentos como pasteurizadores e resfriador da Café Iguaçu é feita em vários setores da fábrica. O hidróxido de sódio chega até a empresa, no caso a Cia Iguaçu de Café Solúvel em caminhões de outra empresa terceirizada. Logo em seguida o operador, da Café Iguaçu, conecta na saída do caminhão uma tubulação flexível, que é ligada a uma moto bomba. Dessa forma o operador abastece um tanque que fica localizado na caldeira de biomassa, este local chama-se especificamente central de produtos químicos.

Quando da necessidade, de utilização do hidróxido de sódio, o mesmo é enviado a um tanque geral de uso exclusivo do setor de produção. Nesse tanque é feita toda a distribuição do hidróxido de sódio para vários outros setores da fábrica que necessitam da solução. Enfim em tanques menores localizados em cada setor da fábrica, o hidróxido de sódio fica a disposição do operador local, para poder ser utilizado e fazer a limpeza nos equipamentos quando necessário. Em alguns setores simplesmente o hidróxido de sódio é utilizado na temperatura ambiente em torno de 25°C a 35°C e a limpeza ocorre sem nenhum tipo de aquecimento. Somente contém nesses setores o controle de nível através de chave boia, e o acionamento por moto bomba para deslocar a solução até os equipamentos e também as válvulas de abertura de saída do tanque são totalmente manuais.

Já em outros setores existem sistemas de controle automático, utilizando topologia profibus, que fazem a indicação da temperatura, nível e porcentagem de abertura de válvulas de controle, todo acionamento do sistema via supervisor. Para o referido controle é utilizando uma válvula de controle de vapor eletropneumática da Spirax sarco, sensor de temperatura, transmissor de temperatura da Smar e transmissor de pressão Yokogawa para indicar o nível da solução, tornando um sistema de simples utilização, oneroso demais para ser aplicado em todos os setores da fábrica, a partir desde momento surgiu-se a ideia de criar um sistema de menor custo de implementação como o descrito neste projeto. A seguir a figura 1 exhibe o fluxograma do processo de recebimento e distribuição do hidróxido de sódio na fábrica para melhor entendimento.

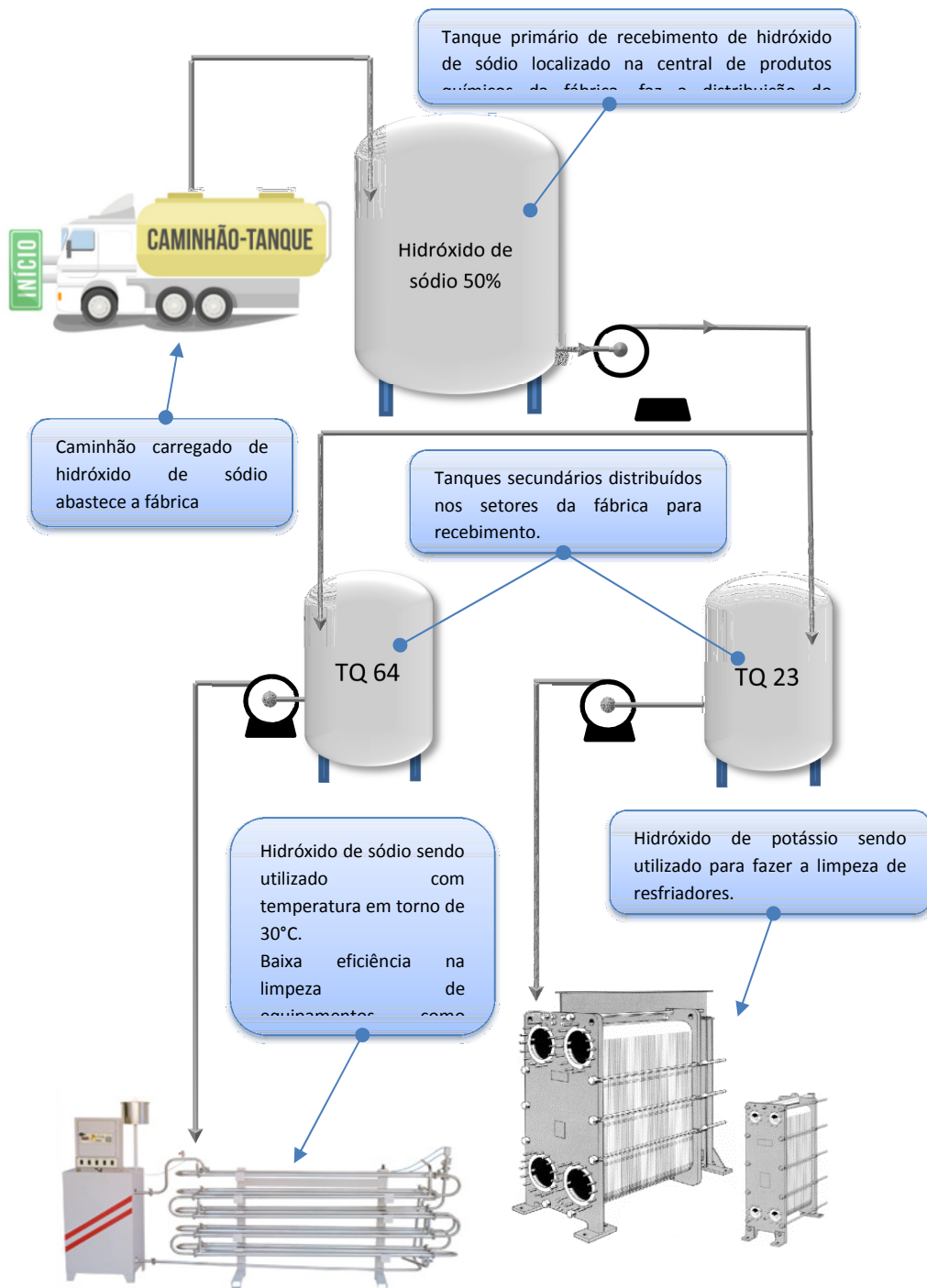


Figura 1 - Fluxograma do processo de recebimento do hidróxido de sódio na fábrica.

## **4.2 - MOTIVOS PARA ESCOLHA DO ARDUINO UNO.**

A plataforma Arduino, possui código aberto, ou seja, software, bibliotecas pré-definidas para o acionamento de sensores e componentes e hardware no caso a placa, por isso torna-se interessante quando se deseja fazer algum projeto, pois é fácil encontrar referências teóricas principalmente na internet, sobre suas aplicações, sensores e shields que podem ser conectados, aumentando sua funcionalidade, dessa forma, o Arduino torna-se um importante instrumento de aprendizado para técnicos tecnológicos e engenheiros, devido à relativa facilidade de programá-lo diferente de outros sistemas como o PIC, que são mais rebuscados e apresenta certa dificuldade em encontrar referências bibliográficas.

Possui sensores e shields especialmente desenvolvidos para conectar em suas entradas e saídas, o que proporciona agilidade no momento da montagem de qualquer projeto, além de software, conhecido como ambiente de desenvolvimento integrado, onde o código desenvolvido que aplicará funcionalidade ao microcontrolador é escrito. Outro aspecto importante do microcontrolador Arduino Uno se verifica no momento da aquisição deste componente, que possui preço relativamente baixo, comparado a outros microcontroladores disponíveis no mercado.

## **4.3 - O MICROCONTROLADOR.**

O Arduino é “um microcontrolador de placa única e um conjunto de software para programá-lo. O sistema consiste em um projeto simples de hardware livre para o controlador, com um processador Atmel AVR e suporte embutido de entrada e saída. O software possui uma linguagem de programação padrão e do bootloader que roda na placa” (MICHAEL ROBERTS, 2011, p.22). Os componentes externos são sensores que monitoram toda e qualquer oscilação sejam eles temperatura, pressão e nível por exemplo.

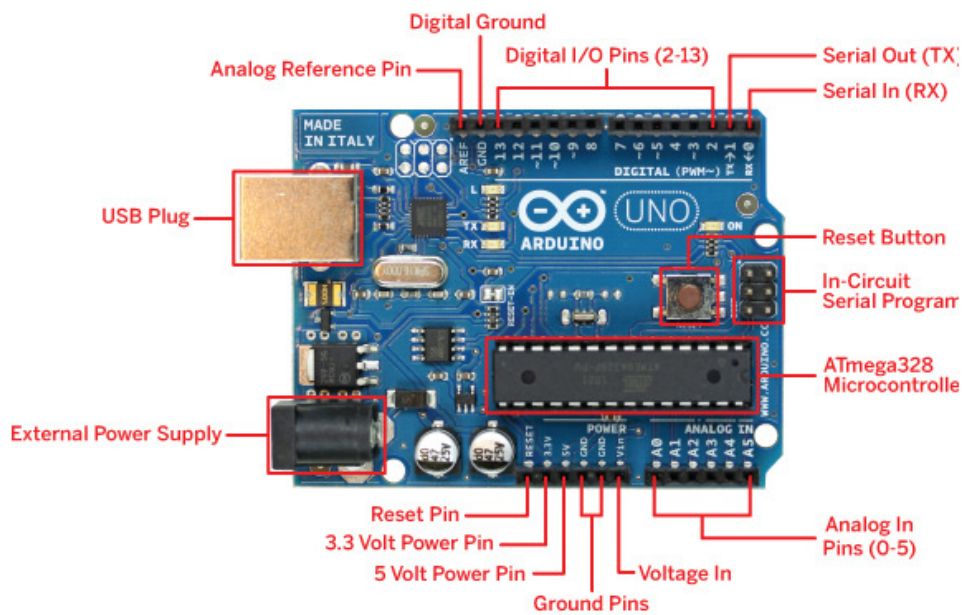
As entradas analógicas são processadas pelo microcontrolador, onde ações são executadas pelo seu desenvolvedor, que geralmente efetuando alguma ação através de atuadores como motores e bobinas, ou são geradas outras saídas, como alarmes ou acionamento de *led's*. A linguagem de programação utilizada no microcontrolador

Arduino conhecida como wiring, e os códigos inseridos são programados num ambiente de desenvolvimento conhecido como IDE, baseado em processing, muito parecida com C++, mas pode-se dizer que o Arduino possui uma linguagem de programação especialmente desenvolvida para ele. A alimentação da placa pode ser feita por cabo USB, que também, através deste cabo o programa que é criado é descarregado para o Arduino. Como alternativa pode-se utilizar alimentação externa de até 12V utilizando para isso bateria ou pilha, observando que a placa possui um conector específico. (WHAT ARDUINO CAN DO, 2014).

Basicamente para se conseguir programar no Arduino, existem duas regras imutáveis para inicializa-lo no qual sem elas o sistema não funciona. São as funções primárias “void setup ( )” e a função “void loop ( )”, ambas são funções executadas pelo software de desenvolvimento de programação do Arduino Uno. Outro aspecto relevante sobre a programação do microcontrolador é com relação a sua portabilidade, uma vez carregado o programa na IDE, através do cabo USB, o mesmo fica gravado, podendo ser desligado a qualquer momento, isso é possível porque dentro da memória do microcontrolador possui um software conhecido como firmware.

Ao religar o microcontrolador, o ultimo programa carregado funciona normalmente, pois está gravado em sua memória, tornando isso um aspecto positivo, levando em consideração o fato de não necessitar refazer o programa e dispensando a utilização do computador para carregar ou desenvolver a lógica de programação mais uma vez. (INTERFACING WITH HARDWARE, 2013). Existem outras plataformas genéricas do microcontrolador Arduino espalhadas pelo mundo, confeccionadas manualmente, e as mesmas são comercializada, em sites facilmente encontrados. Exemplos são o Roboduino, e Freeduino.

Nos controladores Arduino Uno, temos a presença do chip Atmega 328, com 2KB de memoria RAM, 32 KB de memória flash e 1KB de memoria EEPROM, salvo em outros Arduino com maior capacidade, como por exemplo, o Arduino Mega. A velocidade do clock é de 16MHZ, possuindo 14 pinos digitais que podem ser dispostos como entradas ou saídas e seis pinos de entradas analógicas que também podem ser utilizados como saídas digitais. Outra característica da placa é o chip FTDI conversor de sinal de RS232 para USB responsável pela comunicação do Arduino com o computador que também possui seis pinos ICSP, para programação do Arduino em AVR (ARDUINO UNO. 2013). Logo abaixo a figura 2 ilustra os principais componentes do microcontrolador Arduino Uno.



**Figura 2 - Identificação dos componentes da placa**

(Fonte – Arduino Brasil Website, 2014).

As existências de outros controladores no mercado não chamaram tanto a atenção como o microcontrolador Arduino Uno, isso devido ao grande crescimento popular entre os estudantes de tecnologia, além do custo benefício que o equipamento proporciona onde o mesmo pode ser montado à mão, chegando a custar menos que R\$50,00. Outro aspecto importante que deve ser ressaltado se prende ao fato, que o Arduino Uno, pode trabalhar com outras plataformas operacionais como Linux, Macintosh e Windows.

#### **4.4 - ESTRUTURAS DOS COMANDOS DE PROGRAMAÇÃO DO ARDUINO.**

Neste tópico são apresentadas as principais funções e estruturas dos comandos do software do Arduino Uno. Com os códigos a seguir foi desenvolvida a lógica de programação de acionamento dos sensores e atuadores do projeto.

- ***Void setup:*** Função obrigatória, na criação de todos os programas inseridos na IDE do Arduino mesmo quando não for necessária. Utilizada para inicializar variáveis, bibliotecas, definir os pinos como entrada ou saída digital,

inicialização de comunicação serial entre outros. A função `setup` é executada apenas uma vez, uma no momento em que se energiza o microcontrolador ou quando é pressionado o botão reset.

- ***Void loop:*** A função `loop` também é obrigatória na criação de todos os programas inseridos na IDE do Arduino e deve ser criada depois da função `setup`, mesmo quando não for necessário. Como o próprio já informa, esta função gera laços consecutivos, ou seja, todos os comandos inseridos nessa função são lidos sucessivamente o que permite uma atualização constante das informações dos sensores de entrada e de saída.
- ***Ciclo if:*** A esta estrutura de controle é atribuída a função de inserir uma condição dentro da lógica de programação que está sendo desenvolvida ou seja quando em conjunto com operadores aritméticos como, igual, maior, menor ou diferente de algum valor, então executa-se uma determinada condição.
- ***Variável int:*** A variável do tipo `int`, comporta um valor inteiro de 2 bytes, exatamente um valor compreendido entre -32768 até 32767.
- ***Variável long:*** Essa variável suporta guardar grandes valores numéricos entre o que correspondente à 4 bytes ou seja de -2147483648 até 2147483647.
- ***Variável float:*** Essa variável é muito utilizada para leituras analógicas, pois permite melhor resolução do que a variável do tipo `int`. O intervalo de valores dessa variável compreende de  $-3.40 \times 10^{33}$  até  $3.4028235 \times 10^{33}$ .
- ***PinMode:*** Esta função quando acionada, define o estado sendo entrada ou saída de um dos pinos do microcontrolador. Para o correto funcionamento desta função na lógica, ela torna-se inerente a função `void setup`.

- **DigitalWrite:** A atribuição dada a esta função especial, permite a configuração de um ou mais pinos onde suas saídas podem obter valores lógicos de 1 sendo HIGH igual a 5V ou 0 sendo LOW igual a 0V.
- **AnalogRead:** Possibilita a leitura analógica de um sensor instalado em um dos 6 pinos de entrada analógica do microcontrolador Arduino. Com o conversor analógico digital, possui uma resolução de 10bits, ou seja, ocorre um mapeamento da tensão de entrada de 0 à 5V compreendendo onde ocorre a conversão para valores inteiros entre 0 a 1023.
- **delayMicroseconds:** Provoca um pausa na execução da lógica criada na IDE do Arduino especificada em microssegundos.
- **Serial.begin:** Instrução que define a taxa de dados em bits por segundo, para a transmissão de dados serial, onde as taxas de comunicação mas utilizadas com o computador são: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, até 115200.
- **Include:** Utilizado para incluir bibliotecas externas, especialmente desenvolvidas para aplicações com o Arduino. No caso de projeto foi inserido a biblioteca para o acionamento do display.
- **Define:** É uma função que permite que o programador de um nome a um valor constante antes que o programa ser compilado.
- **PulseIn:** É uma função que lê o pulso alto ou baixo de um pino especificado, ou seja lê o tempo entre a chamada até o momento em que o pino entra em nível alto ou baixo.
- **lcd.setCursor:** Especifica a posição e linha e coluna que o caractere deve se posicionar no display de cristal liquido (LANGUAGE REFERENCE, 2014).

#### 4.5 - CI ATMEL ATMEGA 328.

Responsável pelo gerenciamento do microcontrolador Arduino Uno, o Atmega328 da empresa ATMEL, figura 3, possui microchip baseado na arquitetura RISC que é provido de periféricos de entrada e de saída, memória flash para armazenamento das instruções inseridas, sendo que as placas oficiais do microcontrolador Arduino, são constituídos principalmente pelos chips Atmega168 e Atmega328, diferenciando-se entre eles, somente em capacidade de memória e número de pinos.



**Figura 3 - Circuito integrado ATMEGA328.**

(Fonte: Arduino Website, 2014).

O chip Atmega328 possui 40 pinos, portando 14 pinos que podem ser configurados como entrada ou saída, definidos de acordo com os códigos inseridos no software de desenvolvimento criado pelo programador, como por exemplo, a função `pinMode( )`, `digitalWrite( )` e `digitalRead( )`, com um detalhe importante, quando configurado a porta com a função `digitalWrite( )` sua porta pode fornecer até 40mA, para atuar os componentes de potência.

Outros pinos tem funções especializadas como, o pino 0 (RX) e o pino 1(TX) que são utilizados para fazer comunicação serial, conhecidos como porta serial USART. Já os pinos 3, 5, 6, 9, 10, 11 são capazes de gerar sinais PWM, para acionamento e controle de velocidade de motores elétricos, utilizando a função `analogWrite( )`, com esse comando pode-se gerar um sinal alternado com valores altos e baixos ou seja 0V ou 5V, onde o tempo de nível alto é controlado.

Já outros pinos, são utilizados para comunicação SPI, sendo o mesmo, um padrão de comunicação, utilizados para conectar cartão de memória SD, que a título de exemplo, os pinos para realizar a comunicação são os pinos 10(SS), 11(MOSI), 12(MISO) e 13(SCK). Também o pino AREF, utilizado para referencia de tensão para as entradas analógica, executado pela função `analogReference( )`.



Seis pinos funcionam como entrada analógica, interligadas com um conversor analógico-digital de 10 bits fazendo à conversão da leitura do sinal entre um valor de 1024 possibilidades, sendo que por padrão medem de 0 a 5 V, então, através de entradas analógicas podem-se utilizar sensores que façam a medição de alguma grandeza física como temperatura, que posteriormente é transformado em um valor de tensão padrão que pode ser lida pela entrada analógica.

Outras características dos pinos da placa Arduino Uno, é a possibilidade de fazer a comunicação I2C utilizando a biblioteca Wire. O Arduino também possui pino exclusivo para reset, do microcontrolador, onde sua função é reiniciar o código salvo em sua memória ao início da primeira linha de execução. (ATMEL-AVR, 2014), na figura 4 está ilustrado o mapeamento dos pinos do Atmega328.



Figura 4 - Mapeamento da pinagem do ATMEGA 328.

(Fonte: ATMEL 8-bit AVR Datasheet, 2014).

#### 4.6 - DESENVOLVIMENTO DO PROTÓTIPO.

Para mostrar a viabilidade do Arduino em controlar um processo industrial, foi construído um protótipo que possui um tanque constituído de policloreto de vinil (PVC) que contém 33cm de altura por 13,5cm de largura e seu diâmetro corresponde a 48cm. O projeto possui motor AC 110V que tem a função de agitar a solução contida no interior do tanque e também as suas variáveis controladas, como nível e temperatura por sensores. Para que a solução contida no interior do tanque atinja a temperatura de 55°C,

é utilizado um aquecedor elétrico modelo DWT 1000W de potência, contendo em sua extremidade, uma estrutura de metal no formato helicoidal. O aparelho introduzido dentro do tanque é o responsável pelo aquecimento da solução até a temperatura desejável.

Para que haja controle de temperatura no sistema, optou-se pelo sensor de temperatura *LM35* e o controle do nível da solução, conta com o sensor ultrassônico *HC-SR04*, devido ao seu custo reduzido. Outro fator que deve ser levado em consideração, é que este tipo de sensor dispensa a utilização de sensores digitais, eletromecânicos, que geralmente ocupam grandes espaços físicos nas instalações, nas quais são utilizados. O sensor de nível ultrassônico indica o nível exato do líquido no interior, mostrando em porcentagem o preenchimento do líquido no interior do tanque sendo as respectivas leituras de nível e temperatura que são indicadas no display de *LCD* montado no painel sinóptico.

O sensor de nível controla a entrada da solução no interior do tanque, através da válvula acionada por eletroímã. Seu comando aciona a válvula caso o sensor indique nível abaixo de 90%. Senso assim, a válvula de entrada, abre a alimentação da solução, preenchendo o recipiente até atingir 90% do tanque. No mesmo instante, a válvula de saída permanece fechada. Quando a válvula de entrada for acionada para alimentar o tanque, no mesmo instante o *led* indicativo localizado no painel sinóptico ascenderá, para indicar seu acionamento.

Após atingir o nível desejável, o motor agitador do tanque, que é provido de um motor AC 127V, onde seu objetivo é homogeneizar o líquido através do movimento das hélices do motor. O agitador também é sinalizado por *led* indicando seu acionamento no painel sinóptico. No mesmo instante que o agitador é ligado, o aquecedor elétrico entra em funcionamento e o sensor de temperatura *LM35* monitora, e informa no display de *LCD*, qual a temperatura do líquido. Sendo o set-point 55°C ao atingir esse valor, a resistência de aquecimento é desligada.

Por fim com o nível e temperatura controlados dentro das especificações determinadas, o Arduino comanda a abertura da válvula de saída, e a bomba de saída do tanque, e o mesmo é enviado para o destino final, sendo feita a limpeza das tubulações de inox, de pasteurizadores e resfriadores. Com a temperatura estabilizada, melhora-se a eficiência da limpeza dos equipamentos, em consequência disto, evitam-se paradas por entupimento ou vazamentos nos equipamentos e tubulações.

Outra funcionalidade que o projeto apresenta, é no momento da drenagem, ao atingir 10% do nível, o sistema para automaticamente todos os atuadores, e aguarda o comando de reinício do ciclo, através do acionamento do botão reset localizado no painel. Também foi implementado sistema de alarme visual. Caso ocorra alguma falha que faça o nível do tanque indicar acima de 90% um led vermelho instalado no painel sinóptico começa a piscar, impedindo o acionamento da resistência de aquecimento do tanque, com o objetivo de evitar a queima da mesma, caso o nível seja um erro de medição do sensor ultrassônico e não existir líquido no interior do tanque, também acusa falha. O alarme também sinalizará caso a temperatura de aquecimento ultrapassar 60°C, tomando com ação o desligamento do aquecedor elétrico, com o objetivo de evitar a corrosão dos equipamentos de aço inox em contato com hidróxido de sódio aquecido.

Neste projeto, como mostra a figura 5, foram utilizadas todas as portas digitais e analógicas do microcontrolador para desenvolver o protótipo sendo que o display de *LCD* utiliza para seu funcionamento sete portas digitais do Arduino, já o sensor ultrassônico utiliza duas portas digitais, os atuadores utilizam cinco portas digitais, o sensor de temperatura uma porta analógica e os *leds* do painel utilizando cinco portas analógicas, que são transformadas em saídas digitais. Também foram utilizados os pinos de força do microcontrolador (GND) e (5V). O pino reset embarcado na placa do microcontrolador foi utilizado com a função de fazer o reinício do ciclo do processo, através do pressionamento de um botão instalado no painel sinóptico ele executa esta função.



**Figura 5 - Protótipo desenvolvido.**

#### 4.7 - FLUXOGRAMA DO PROJETO.

Neste tópico é apresentado o fluxograma do processo, que o protótipo efetua a partir do momento que o operador pressiona o botão iniciar, onde pode ser visto representado pela figura 6. Com esse desenho esquemático pretende-se mostrar as ações que o microcontrolador irá tomar quando em funcionamento.

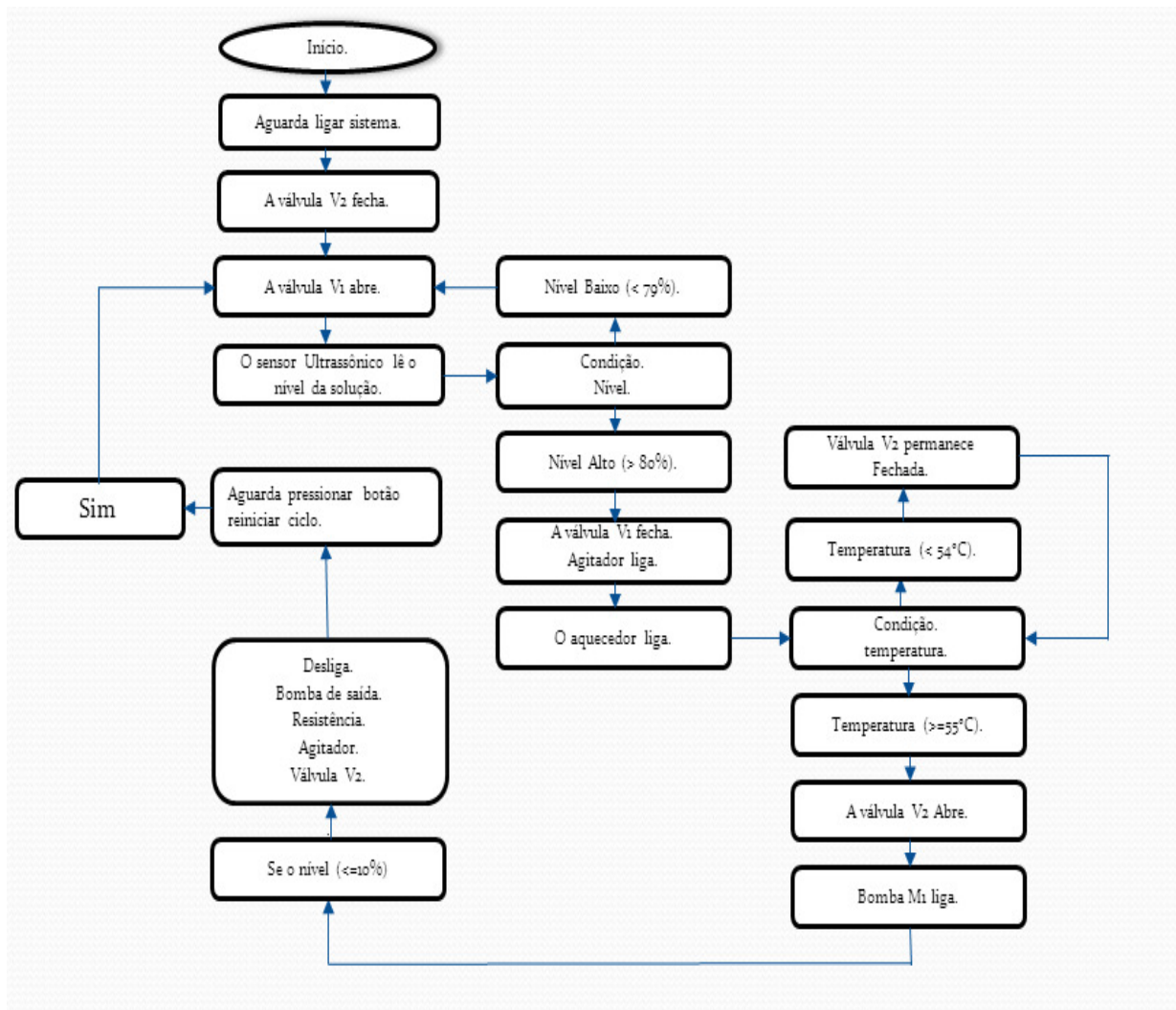


Figura 6 - Fluxograma do processo.

#### 4.8 - MOTORES MONOFÁSICOS AC.

Segundo (Claiton Moro Franchi 2012, p.17) “O motor elétrico é um dispositivo que transforma energia elétrica em mecânica, em geral, energia cinética, ou seja, num

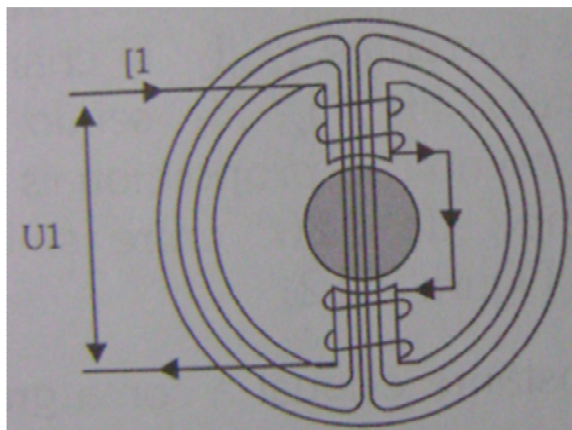
motor, a simples presença da corrente elétrica, garante o movimento em um eixo, que pode ser aproveitado de diversas maneiras, dependendo da aplicação do motor”.

Em protótipos são geralmente utilizados motores de corrente contínua devido ao seu tamanho relativamente pequeno e seu custo também é proporcional ao seu tamanho. Já nesse protótipo foi utilizado motor AC devido à necessidade do projeto exigir um motor mais robusto para produzir uma pressão na válvula solenoide de saída, possibilitando uma vazão de saída considerável.

Este projeto exige utilizar dois motores AC, o primeiro tem a função de homogeneizar o líquido interno, atuando como agitador garantindo que toda a solução atinja 55°C uniformemente. O segundo tem a função de bombear a solução do interior do tanque para fora, ou seja, injetar hidróxido de potássio nas instalações da fábrica como, por exemplo, resfriadores e pasteurizadores.

Segundo (Claiton Moro Franchi, 2012, p.53) “Quando uma bobina é percorrida por uma corrente elétrica, cria-se um campo magnético dirigido conforme o eixo da bobina e com seu valor proporcional à corrente”. A figura 7 exhibe um enrolamento monofásico atravessado por uma corrente  $I$ , e o campo  $H$  é criado por ela.

O enrolamento é constituído por um par de polos (norte e sul) cujos efeitos se somam para estabelecer o campo  $H$ . O fluxo magnético atravessa o rotor entre os dois polos e se fecha através do núcleo do estator.



**Figura 8 - Excitação das bobinas do motor monofásico.**

(Fonte – Livro de Acionamentos Elétricos).

#### 4.9 - VÁLVULA SOLÉNOIDE.

A válvula solenoide, figura 8, tem a função no projeto de permitir a entrada e a saída da solução do interior do tanque (PVC) totalizando duas peças, que são posicionadas uma na entrada e outra na saída do tanque, responsáveis pelo controle de fluxo da solução no interior do recipiente.

A válvula solenoide contém para o correto funcionamento o respectivo núcleo móvel, (plunger) e seu obturador e o corpo dotado de um orifício, no qual o núcleo móvel se posiciona. É o obturador que permite ou impede à passagem de fluxo do fluido contido no interior da tubulação em função da atração, ou não, do núcleo móvel, no momento em que, uma corrente percorre a bobina de cobre revestida de verniz, energizando este elemento, dessa forma a corrente elétrica cria um campo magnético que atrai o elemento ferroso. (VALVULA E ACESSÓRIO CONTROLE DE FLUIDO, 2014).



**Figura 9 - Válvula solenoide utilizada no protótipo.**

#### 4.10 - O SENSOR ULTRASSÔNICO HC-SR04.

Este projeto conta com um sensor ultrassônico que será responsável por controlar o nível da solução no interior do tanque, a figura 9 ilustra o sensor utilizado no protótipo.





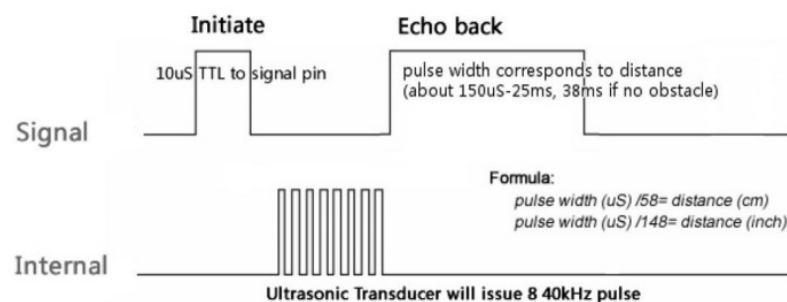
**Figura 10 - Sensor ultrassônico HC-SR04.**

O sensor de distância ultrassônico tem a capacidade de medir distâncias de 2cm a 5m. O componente possui um circuito pronto com emissor e receptor ultrassônico. As suas características são descritas na listagem a seguir.

- Fonte de alimentação: 5V.
- Corrente de operação: 2mA.
- Ângulo de efeito: 15°.
- Alcance: 2 cm ~ 500 cm.
- Precisão: 0,3 cm.

Para entendimento do funcionamento deste sensor, observe a figura 10 a seguir.

### Sequence chart



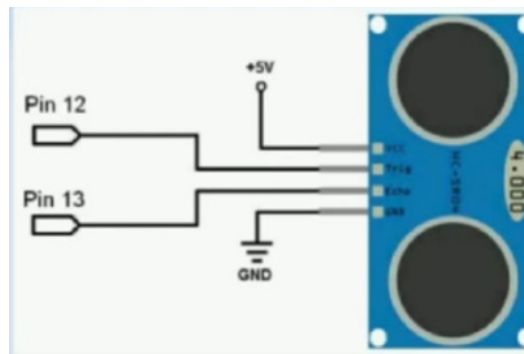
**Figura 11 - Gráfico de sequência de disparo do sensor.**

(Fonte – Ultrasonic Module Datasheet Catalog, 2014).

O sensor HC-SR04 gera oito pulsos de 40KHZ internamente enquanto o pino trigger dispara um sinal de 10µS para gerar o calculo interno. Para fazer esse calculo, o

sensor tem como parâmetro que a velocidade do som é em torno de 340m/s, logo esse valor deve ser dividido por dois, onde se obtém o valor de 170m/s, ou seja, 17000 cm/s.

Dessa forma pode-se calcular, se um objeto estiver 1700 cm o sensor terá um tempo de resposta de 1s, ou seja, 1cm será percorrido em 0,058ms ( ULTRASONIC RANGING MODULE, 2014). Na figura 11, ilustra o esquema de ligação do sensor.



**Figura 12 - Esquema de ligação do sensor ultrassônico.**

(Fonte – Novatec Website, 2014).

#### 4.11 - SENSOR DE TEMPERATURA LM35.

O sensor escolhido para monitorar a temperatura da solução, foi o LM35, devido principalmente ao baixo custo, além de apresentar ótima precisão de medição. As especificações do sensor atende plenamente as exigências desse projeto. Logo abaixo a figura 12, ilustra o LM35 utilizado no protótipo.



**Figura 13 - Sensor de temperatura LM35.**

O sensor de temperatura LM35 trata-se de um circuito integrado, fabricado pela National Semiconductor, em síntese, o sensor gera uma tensão de saída linearmente



proporcional à temperatura em graus Celsius quando alimentado entre 4 a 30Vdc, sendo que seu sinal de saída gera 10mV para cada grau Celsius de temperatura.

Outra vantagem do LM35 se deve ao fato de não necessitar de calibração externa para fornecer valores de temperatura com variações de 0,25°C até 0,75°C na faixa de temperatura de -55°C a 150°C (NATIONAL SEMICONDUCTOR, 2014).

O modelo de carcaça utilizado pelo sensor é TO-92, com aspecto de um transistor, tornando este modelo de sensor um dos mais baratos do mercado, porém com ótima funcionalidade. Na figura 13 é apresentado o esquema de ligação do sensor de temperatura.

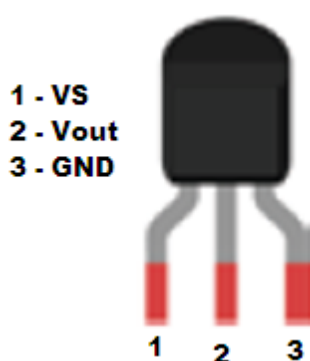


Figura 14 - Esquema de ligação do LM35

(Fonte – Novatec Website, 2014).

A seguir apresenta-se um resumo das características do sensor.

- Calibrado diretamente em graus Celsius;
- Linear: +10.0 mV/°C (fator de escala);
- 0,5°C de exatidão a 25°C;
- Faixa de funcionamento de (-55°C) à (+150°C);
- Adequado para aplicações de monitoramento remoto;
- Opera em tensões de 4 a 30 Vdc;
- Dreno de corrente menor que 60µA;
- Saída com baixa impedância: 0,1 Ω /1mA.

#### 4.12 - PAINEL SINÓPTICO.

O projeto inclui a criação do painel sinóptico, que sintetiza todo funcionamento do sistema proposto, quando o mesmo estiver em operação. O painel sinóptico é um sistema de monitoramento barato e eficiente, onde o operador obtém informações necessárias para poder iniciar todo o processo de aquecimento da solução no caso o hidróxido de sódio a 3% sem a necessidade de se ausentar da sala de operação, para acompanhar o ciclo.

No painel sinóptico o operador pode visualizar informações como, abertura e fechamento de válvulas de alimentação e de saída do produto, acionamento da bomba de saída e do agitador do tanque, tudo indicado através de *led's* que se acenderão quando o atuador estiver ligado e apaga-se quando o atuador desligar-se. Além dessas informações o operador pode visualizar a temperatura do interior do tanque e também o nível, indicado em porcentagem no display de cristal líquido instalado, a figura 14 ilustra o diagrama de instrumentação utilizado.

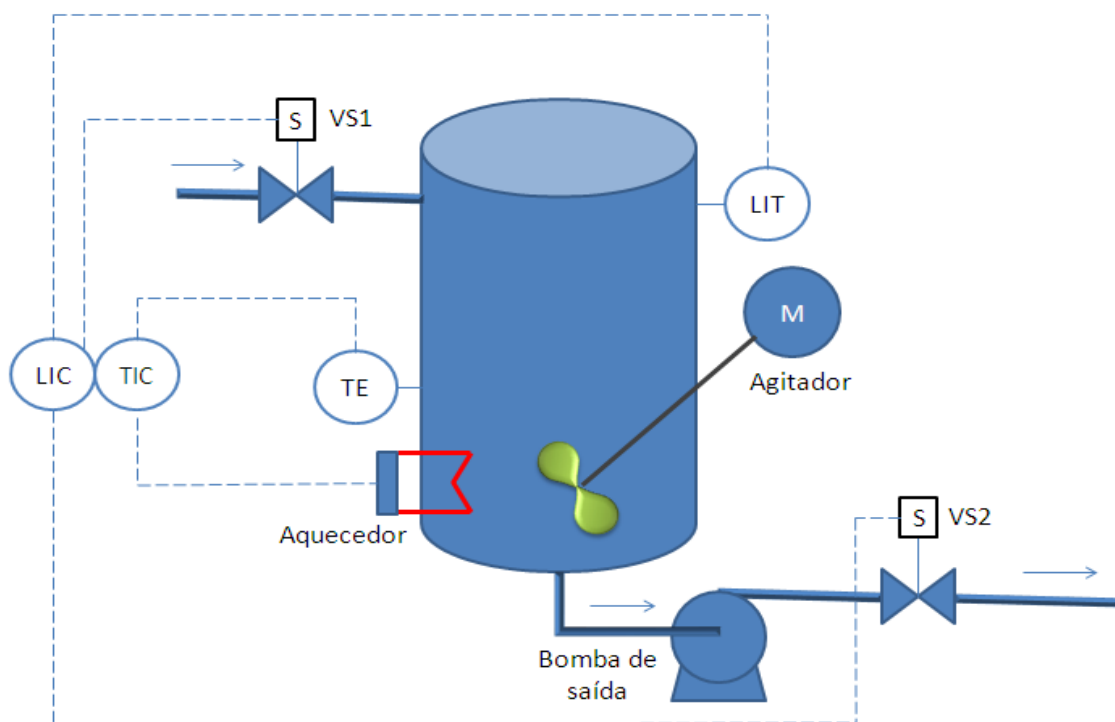


Figura 15 - Diagrama de instrumentação do projeto.

#### 4.13 - MÓDULO RELÉ 4 CANAIS 12V.

O módulo foi instalado no projeto, devido à necessidade de acionamento de equipamentos que consomem uma quantidade de corrente maior que 40mA. Acima desse valor as portas de saída do Arduino Uno podem sofrer algum dano. A seguir a figura 15 apresenta o modulo utilizado no projeto, e na figura 16 ilustra-se o esquema de ligação do componente eletrônico ao Arduino. Estes módulos são conhecidos também como *bricks*, possuem quatro relés operados a 12V, drenando 5mA de corrente das portas digitais do microcontrolador, na qual, coloca em operação o circuito integrado ULN2003, e os contatos dos quatro relés suportam uma corrente de no máximo 10A. (MANUAL TÉCNICO, 2014).

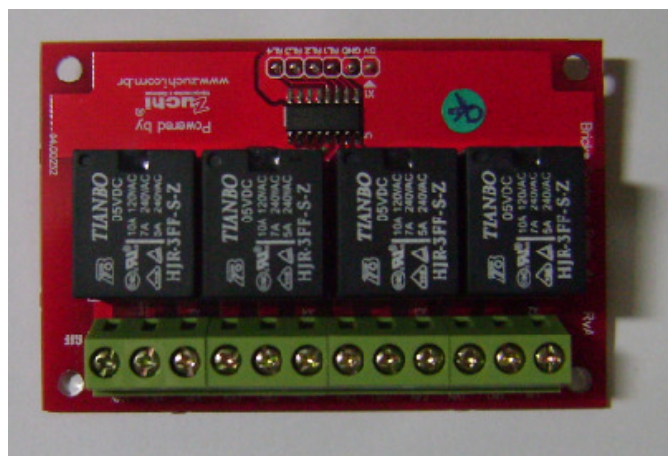


Figura 16 - Módulo relé utilizado no projeto.

A seguir apresenta-se um resumo das características do módulo relé.

- Saídas com opções de NA ou NF.
- Quatro saídas por contato 10A em 127V.
- Conectores parafusáveis, para acionamento das cargas.
- Acionamento através de sinal lógico de 5V.
- Alimentação de 5V para operação.

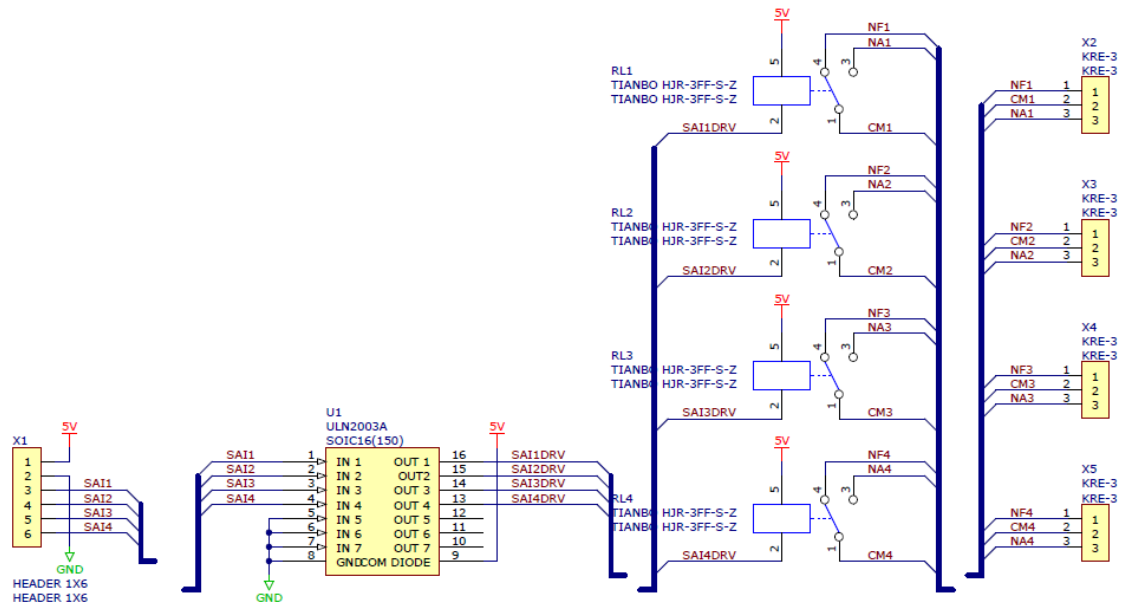


Figura 17 - Esquema de ligação do módulo relé 12V.

(Fonte – Bricks Relés Datasheet Catalog, 2014).

#### 4.14 - DISPLAY DE LCD.

O display de LCD utilizado no projeto tem como objetivo indicar simultaneamente a temperatura e nível da solução no interior do taque. O modelo utilizado “Hitachi HD44780 1602A” são os mais populares do mercado atualmente como mostra a figura 17.

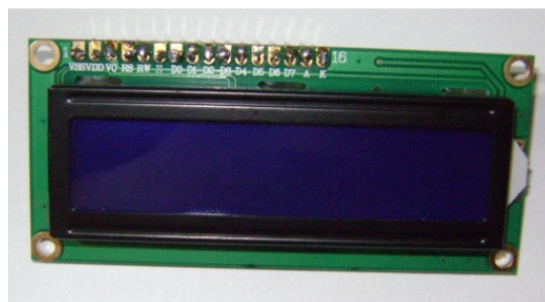
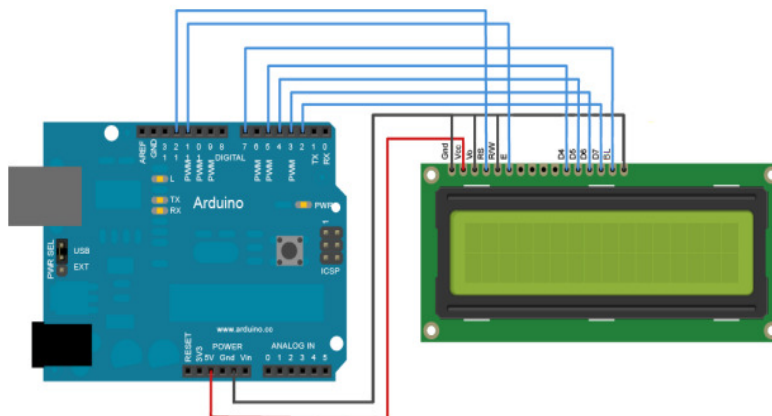


Figura 18 - Display LCD 1602A.

Esse display tem capacidade para exibir 16 caracteres por 2 linhas com “backlight” (luz de fundo) azul, 8 “bits” de dados para comunicação (em paralelo) e necessita de uma fonte de alimentação simples para sua utilização. Na figura 18 a seguir está ilustrado o esquema de ligação do display de cristal líquido (JHD204, 2005).



**Figura 19 - Esquema de ligação do LCD no Arduino Uno.**

(Fonte – Novatec Website, 2014).

Segundo (Claiton Moro Franchi 2012, p.207) o funcionamento do display consiste em “utilizar as propriedades de modulação da luz nos cristais líquidos, sendo o display formado de pixels, cada um, preenchido com cristais líquidos. Esses pixels são dispostos à frente de uma fonte de luz. Os cristais são posicionados em camadas impressadas entre filtros polarizados. Os dois painéis polarizados são alinhados em 90 graus um do outro, o que bloqueia a luz. O primeiro filtro polarizará as ondas de modo que obtenha uma orientação, já o segundo filtro bloqueará a luz”.

A seguir apresenta-se a tabela 1 que exibe uma lista com a função associada à descrição dos pinos do display LCD.

PIN NO	Symbol	Fuction
1	VSS	GND
2	VDD	+5V
3	V0	Contrast adjustment
4	RS	H/L Register select signal
5	R/W	H/L Read/Write signal
6	E	H/L Enable signal
7	DB0	H/L Data bus line
8	DB1	H/L Data bus line
9	DB2	H/L Data bus line
10	DB3	H/L Data bus line
11	DB4	H/L Data bus line
12	DB5	H/L Data bus line
13	DB6	H/L Data bus line
14	DB7	H/L Data bus line
15	A	+4.2V for LED
16	K	Power supply for BKL(0V)

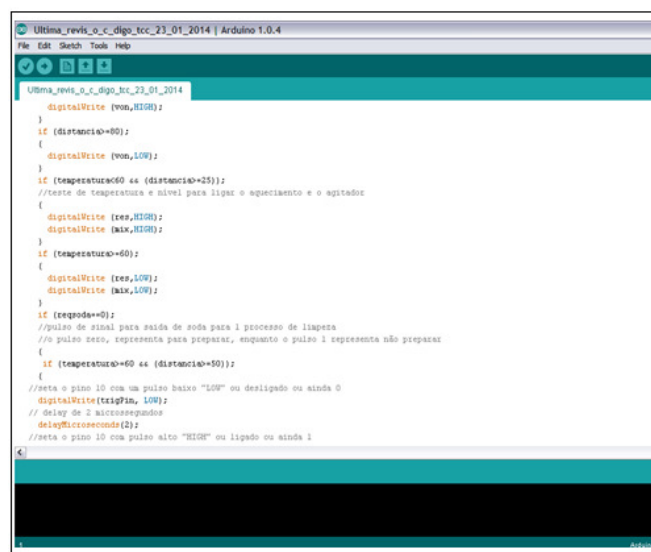
**Tabela 1 – Esquema de pinagem do LCD**

(Fonte: Datasheet Catalog, 2014).

#### 4.15 - AMBIENTE DE DESENVOLVIMENTO INTEGRADO.

A programação inserida no software do Arduino Uno conhecido como IDE, para desenvolver o controle do sistema, consiste em uma lógica de programação bem organizada visualmente e criada na linguagem *Processing*, muito semelhante com a linguagem de programação C++. Sua interface gráfica é construída em Java como mostra a figura 19, simplificando, o software é intuitivo, e sua função principal é permitir o desenvolvimento da lógica de programação e enviá-lo à placa Arduino através do cabo USB.

Processing é uma linguagem de programação open-source desenvolvida por estudantes da Universidade Harvard. Atualmente estudantes e profissionais que se habilitam em criar projetos de animações e interações. Este software ensina os fundamentos de programação de computadores dentro de um contexto visual (PROCESSING, 2014).



```
Ultima_revis_o_c_digo_tcc_23_01_2014 | Arduino 1.0.4
File Edit Sketch Tools Help

Ultima_revis_o_c_digo_tcc_23_01_2014
digitalWrite (pin,HIGH);
}
if (distancia<=00);
{
digitalWrite (pin,LOW);
}
if (temperatura<=44 (distancia<=25));
//teste de temperatura e nivel para ligar o aquecimento e o aptador
{
digitalWrite (pin,HIGH);
digitalWrite (pin,HIGH);
}
if (temperatura>=60);
{
digitalWrite (pin,LOW);
digitalWrite (pin,LOW);
}
}
if (tempo=0);
//pulo de sinal para saída de onda para o processo de liapeta
//o pulso zero, representa para preparar, enquanto o pulso 1 representa não preparar
{
if (temperatura<=60 && (distancia<=50));
{
//seta o pino 10 com um pulso baixo "LOW" ou desligado ou ainda 0
digitalWrite (pin, LOW);
// Delay de 2 milissegundos
delay (microseconds(2));
//seta o pino 10 com pulso alto "HIGH" ou ligado ou ainda 1
```

Figura 20 - Ambiente de desenvolvimento do Arduino Uno.

#### 4.16 - CÓDIGO INSERIDO NO ARDUINO.

É extremamente fácil inserir o código na IDE do Arduino Uno, onde somente é necessário à instalação do software em um computador específico, seguindo os passos de instalação que está localizado no site do fabricante disponibiliza na internet gratuitamente, mais precisamente no site, [www.arduino.cc](http://www.arduino.cc). No caso deste projeto foi baixado à versão para Windows XP.

Depois de baixado o software, deve-se fazer a digitação da lógica de programação no software. Por fim, conectar o cabo USB entre o computador e o microcontrolador. Já com a lógica desenvolvida e finalizada, descarrega-se o programa no microcontrolador clicando no ícone denominado download, que fica localizado na barra de ferramentas do software, que por fim, salva o código na memória permanente do microcontrolador. O código desenvolvido para o projeto no ambiente de desenvolvimento integrado do microcontrolador Arduino Uno, encontra-se no anexo A desta monografia.

#### 4.17 - APRESENTAÇÃO DO CIRCUITO ELÉTRICO DO SISTEMA.

Neste tópico é apresentado o esquema elétrico geral do sistema utilizado no projeto, com o objetivo de melhorar o entendimento das ligações entre os componentes do protótipo, onde se pode ver a ilustração na figura 20.

Na interligação, deste desenho esquemático, foi interconectado o microcontrolador Arduino Uno, juntamente com display de *LCD*, sensor de temperatura *LM35*, e também, sensor de nível ultrassônico *HC-SR04*, que por fim conecta o microcontrolador aos dois módulos reles de saídas *ULN2003*.

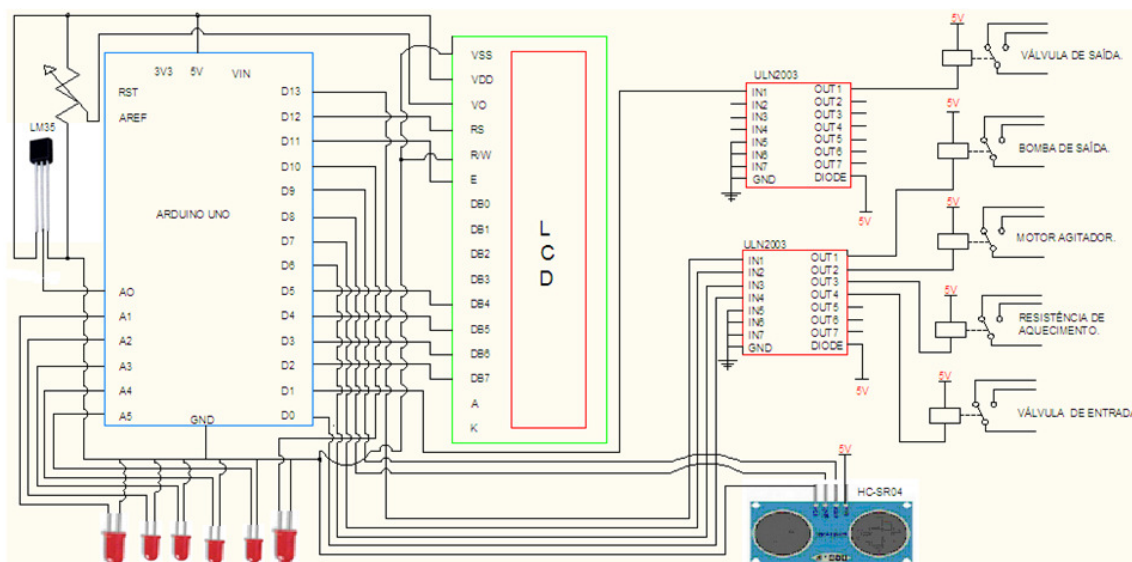


Figura 21 - Diagrama esquemático de controle dos componentes.

#### 4.18 – POSSÍVEL APLICAÇÃO DO PROJETO PARA A INDÚSTRIA.

Neste tópico é apresentado o esquema elétrico, de como ficaria, a aplicação do protótipo no ambiente industrial, para acionamento de motores elétricos e das válvulas solenoides, salvo os problemas de isolamento dos componentes eletrônicos, devido a agressividade do ambiente industrial, onde a aplicação de projeto efetivamente na indústria demanda vários estudos, sobre adaptação nesses ambientes. Os esquemas elétricos a seguir mostra a parte de força dos atuadores, como o motor agitador e a bomba de saída de produto.

Também é sugerida uma modificação, que seria na forma de como o sensor ultrassônico poderia ser instalado. Para que o sensor ultrassônico não sofra a ação dos gases, provenientes do hidróxido de sódio, que é altamente corrosivo pode-se instalar o ultrassônico num vaso comunicante ao lado externo do tanque, com uma proteção de náilon ou similar onde a função seria de evitar o contato direto dos gases diretamente com o sensor de distância como mostra a figura 21, mas como dito anterior mente, a implementação na indústria efetivamente, ainda demanda muitos estudos e adaptações para o correto funcionamento.

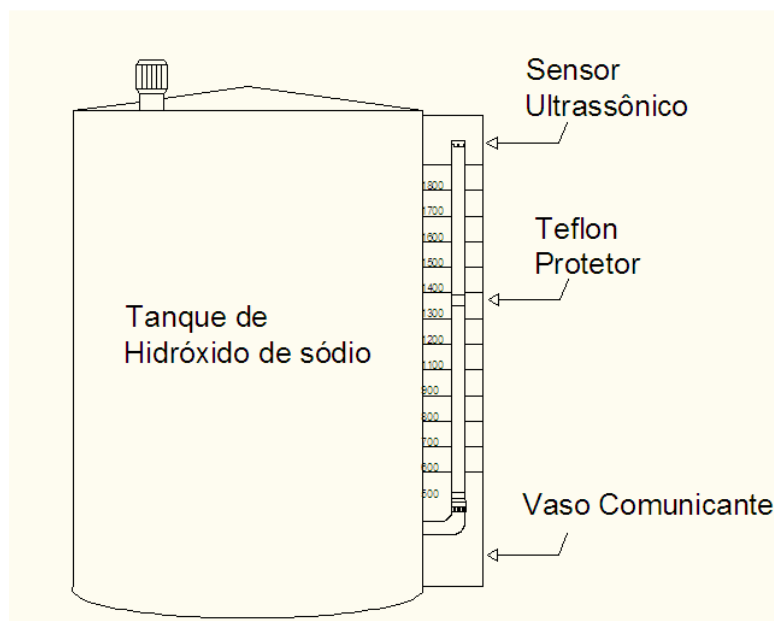
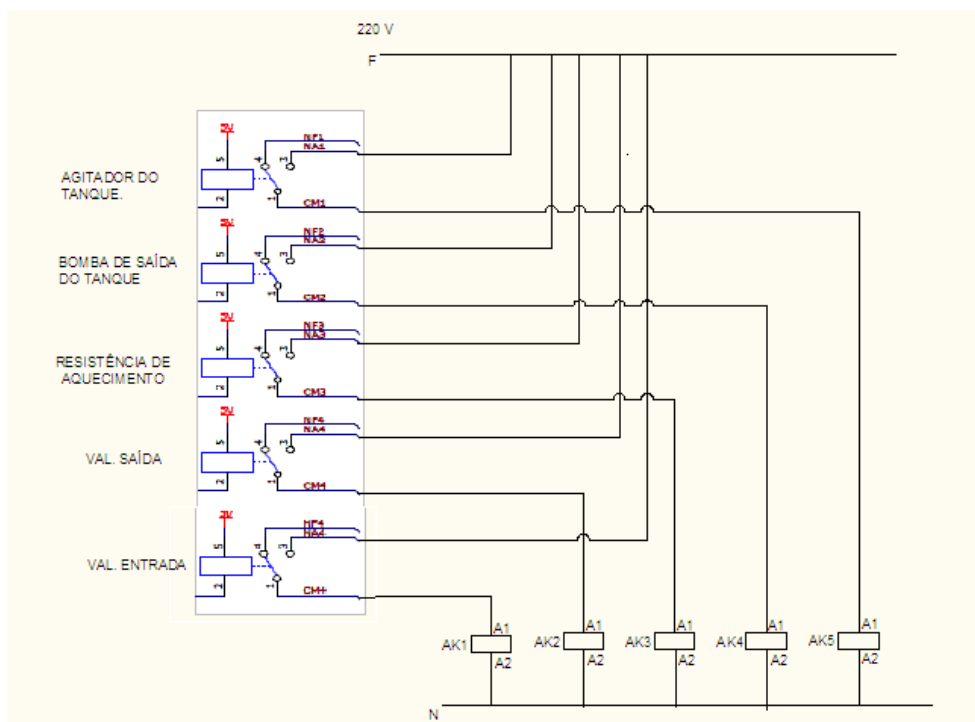


Figura 22 - Instalação do sensor ultrassônico no vaso comunicante.

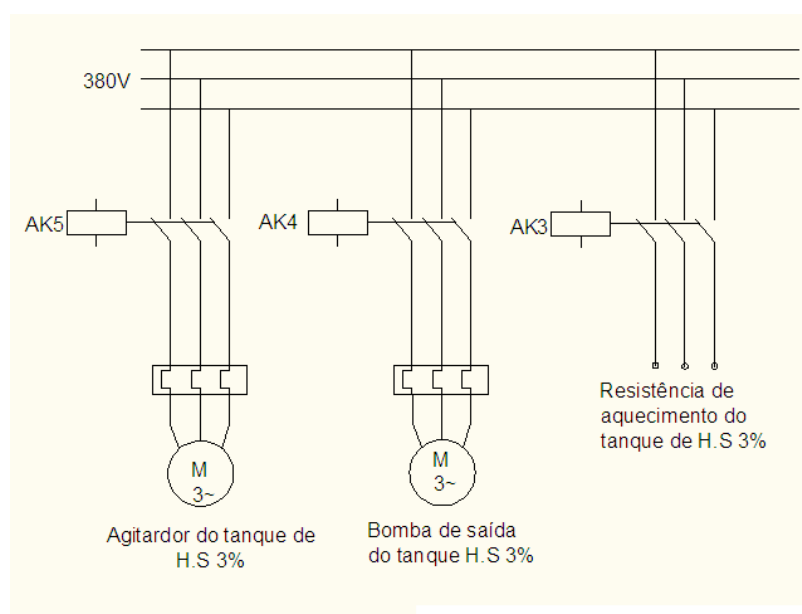
A figura 22 ilustra-se os módulos relés acionando as bobinas solenoides dos contadores AK1 (válvula de entrada), AK2 (válvula de saída), AK3 (resistência de aquecimento), AK4 (bomba de saída da solução). AK5( motor agitador).





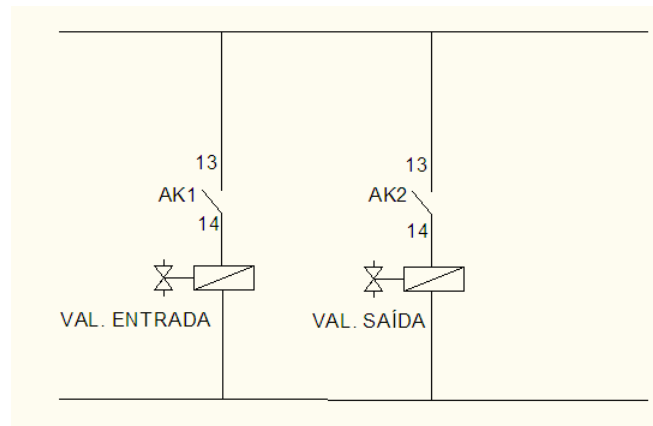
**Figura 23 - Módulo relé acionando bobinas dos contatores.**

Na figura 23 ilustra-se o esquema elétrico de potência dos atuadores trifásicos, quando da energização do contator (AK1) o motor agitador do tanque de hidróxido de sódio a 3% é acionado, quando da energização do contator (AK2) ocorre o acionamento da bomba de saída da solução, e quando o contator (AK3) for acionado, a resistência de aquecimento do tanque é ligada.



**Figura 24 - Esquema elétrico dos atuadores trifásicos.**

Na figura 24 no momento em que as bobinas dos contatores (AK1) e (AK2) são energizadas os contatos auxiliares 13 e 14 dos respectivos contatores, dessa forma, energizam-se as válvulas solenoides, de entrada e de saída do tanque de hidróxido de sódio a 3%.



**Figura 25 - Esquema de acionamento das bobinas solenoides.**

## **5 – CONSIDERAÇÕES FINAIS.**

Com a montagem do protótipo pode-se observar que o processo de aquecimento da solução do hidróxido de sódio a 3% necessita de atenção especial primeiro por se tratar de um produto caro, e a proposta atenta para o aspecto financeiro, pois desenvolve um sistema que otimiza e padroniza, o aquecimento desta solução preocupado com a máximo aproveitamento que este produto pode oferecer respeitando as propriedades físicas dos materiais constituídos de inox.

Em segundo lugar que o projeto, trata o assunto com relação à segurança, pois o monitoramento e controle da temperatura, intertravado com alarmes é fundamental para se evitar acidentes, pois o aquecimento desproporcional pode desencadear acidentes com esse produto. E também neste trabalho mostrou-se a necessidade de um controle de nível com monitoração com alarmes de segurança, pois o transbordamento desse produto, também pode causar sérios acidentes, pois trata-se de um produto degradante e corrosivo, não somente com os materiais constituídos de inox.

## 6 – REFERÊNCIAS.

MCROBERTS, Michael. **Arduino básico**. [São Paulo]: Novatec, 2011.

FRANCHI, Claiton Moro. **Acionamentos Elétricos**. [São Paulo]: Érica, 2012.

FICHA TÉCNICA DE SEGURANÇA DO PRODUTO QUIMICO. Disponível em:  
<[http://www.bbquimica.com.br/bbq/produtos/content/hidroxido\\_sodio.pdf](http://www.bbquimica.com.br/bbq/produtos/content/hidroxido_sodio.pdf)>. Acesso em:  
27 jan. 2014.

*INTERFACING WITH HARDWARE*. [S.l.]: Arduino 2013. Disponível em:  
<<http://www.arduino.cc/playground/Main/InterfacngWithHardware>>  
Acesso em: 22 jul. 2013

LANGUAGE REFERENCE. Disponível em:  
< <http://www.arduino.cc/en/Reference/HomePage> >. Acesso em 22 fev. 2014.

SIEMENS. Disponível em:  
<[http://www.siemens.com.br/medis/Files/2910\\_20060412165714.pdf](http://www.siemens.com.br/medis/Files/2910_20060412165714.pdf). >  
Acesso em: 23 jul2013.

ARDUINO UNO. Disponível em: <<http://www.arduino.cc/en/main/arduinoBoardUno>>.  
Acesso em: 01 jul.2013.

ATMEL-AVR. Disponível em: < <http://www.atmel.com/Images/doc8161.pdf>> Acesso  
em: 11 fev. 2014.

*ULTRASSONIC RANGING MODULE*. Disponível em:  
<http://user.ece.utexas.edu/~valvano/Datasheets/HCSR04b.pdf>. Acesso em: 14 jun. 2013.

*LIQUID CRYSTAL*. Disponível em:  
<<http://arduino.cc/en/Tutorial/LiquidCrystalDisplay>>. Acesso em: 3 jun. 2013.

WHAT ARDUINO CAN DO. Disponível em: < <http://www.arduino.cc> > Acesso em 21 jan. 2014.

VALVULA E ACESSÓRIO CONTROLE DE FLUIDO. Disponível em:  
<[http://www.norless.com.br/downloads/Valvulas\\_solenoideas.pdf](http://www.norless.com.br/downloads/Valvulas_solenoideas.pdf)> Acesso em 5 jan. 2014.

ULTRASONIC RANGING MODULE. Disponível em:  
< <http://www.webtronico.com/documentos/HC-SR04.pdf>>. Acesso em: 03 jan. 2014

NATIONAL SEMICONDUCTOR. Disponível em:  
<<http://www.datasheetcalalog.org/datasheet/nationalsemicoductor/DS005516.PDF>>  
Acesso em: 06 jan. 2014

MANUAL TÉCNICO. Disponível em:  
<[http://www.serialink.com.br/lab/Arduino/Datasheet\\_Modulo\\_Rele\\_4\\_Canais\\_SerialLi nk.pdf](http://www.serialink.com.br/lab/Arduino/Datasheet_Modulo_Rele_4_Canais_SerialLi nk.pdf)>. Acesso em: 07 jan. 2014

PROCESSING. Disponível em:  
< <http://www.processing.org/> >. Acesso em 06 fev. 2014

**ANEXO A** – Código Desenvolvido para o projeto.

```
#include <LiquidCrystal.h> // declara a utilização da biblioteca LiquidCrystal.h
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // define os pinos que envia dados ao lcd.
#define echoPin 9 // pino 9 recebe o pulso do eco.
#define trigPin 8 // pino 8 envia o pulso para gerar o eco.
long proporcao; // variável para guardar o valor do cálculo do nível.
```

```
int pinotemp = A0; // define pino para sensor de temperatura.
int valorLido = 0; // variável para tratar a temperatura.
float temperatura = 0; // variável para armazenar o valor da temperatura.
```

```
int vonPin = 0; // válvula de entrada.
int vsPin = 1; // válvula de saída.
int msPin = 13; // motor de saída.
int resPin = 6; // resistência de aquecimento.
int mixPin = 7; // agitador.
```

```
int ledPinres = A1; // led correspondente ao acionamento da resistência.
int ledPinvon = A2; // led correspondente ao acionamento do válvula de entrada.
int ledPinvs = A3; // led correspondente ao acionamento da válvula de saída.
int ledPinms = A4; // led correspondente ao acionamento do motor de saída.
int ledPinmix = A5; // led correspondente ao acionamento do agitador.
int ledPinalarm = 10; // alarme de nível alto ou temperatura alta.
```

```
long distancia; // variável para ajustar valores 0% a 100%.
```

```
void setup() // esta função é executado apenas uma vez.
{
  pinMode(echoPin, INPUT); // define o pino 13 como entrada (recebe).
  pinMode(trigPin, OUTPUT); // define o pino 12 como saída (envia).
  lcd.begin(16, 2); // Iniciando o objeto "lcd" de 2 linhas e 16 colunas.
  pinMode(distancia, INPUT); // seta como entrada.

  pinMode (vonPin,OUTPUT); // define pino 0 como saída digital.
  digitalWrite (vonPin,HIGH); // define nível alto para porta.
  pinMode (vsPin,OUTPUT); // define pino 1 como saída digital.
  digitalWrite (vsPin,LOW); // define nível baixo para porta.
  pinMode (resPin,OUTPUT); // define pino 6 como saída digital.
  digitalWrite (resPin,LOW); // define nível baixo para porta.
  pinMode (pinotemp,INPUT); // define pino A0 como entrada.
  pinMode (mixPin, OUTPUT); // define pino 7 como saída digital.
  digitalWrite (mixPin, LOW); // define nível baixo para porta.
  pinMode (msPin, OUTPUT); // define pino 13 como saída digital.
  digitalWrite (msPin, LOW); // define nível baixo para portal
  pinMode (ledPinres, OUTPUT); // define pino A1 como saída digital.
  digitalWrite (ledPinres, LOW); // define nível baixo para portal A1.
  pinMode (ledPinvon, OUTPUT); // define pino A2 como saída digital.
  digitalWrite (ledPinvon, HIGH); // define nível baixo para portal A2.
  pinMode (ledPinvs, OUTPUT); // define pino A3 como saída digital.
  digitalWrite (ledPinvs, LOW); // define nível baixo para porta A3.
```

```

pinMode(ledPinms, OUTPUT); // define pino A4 como saída digital.
digitalWrite(ledPinms, LOW); // define nível baixo para porta A4.
pinMode (ledPinmix, OUTPUT); // define pino A5 como saída digital.
digitalWrite (ledPinmix, LOW); // define nível baixo para porta A5.
pinMode (ledPinalarm, OUTPUT); // define pino 10 como saída digital.
digitalWrite (ledPinalarm,LOW); // define nível baixo para porta 10.

}

void loop() // esta função é executado a cada 1 segundo.
{
  // Inicia o objeto "lcd" de 2 linhas e 16 colunas.
  lcd.begin(16, 2);
  Serial.begin(9600);
  // seta o pino 12 com um pulso baixo "LOW"
  digitalWrite(trigPin, LOW);
  // delay de 2 microssegundos.
  delayMicroseconds(2);
  // seta o pino 12 com pulso alto "HIGH".
  digitalWrite(trigPin, HIGH);
  // delay de 10 microssegundos.
  delayMicroseconds(10);
  // seta o pino 12 com pulso baixo novamente.
  digitalWrite(trigPin, LOW);

  //pulseInt é uma função que verifica, o tempo que pino echoPin fica em HIGH.
  long duration = pulseIn(echoPin,HIGH); // variável para calcular a distância.

  long distancia =((duration/58)*10); //faz o sensor ler em centímetros de distância.

  //cálculo para converte em porcentagem (%) o valor exibido no display.
  long proporcao = (((distancia-300)*-3.7)/10);
  // atribui valor da porta A0 e guarda o valor na variável (valorLido).
  valorLido = analogRead(pinotemp);

  // converte para graus celsius (°C) o valor exibido no display.
  temperatura = (valorLido * 0.00488)*100;

  // Serial.println verifica os valores que estão sendo gerados dentro dessas variáveis.
  Serial.print("temperatura");
  Serial.println(temperatura);

  Serial.print("valorLido");
  Serial.print(valorLido);

  Serial.print("distancia");
  Serial.println(distancia);

  Serial.print("duration");
  Serial.println(duration);

```



```

Serial.print("proporcao");
Serial.println(proporcao);

lcd.clear();
lcd.setCursor(0,0); // seta o cursor para: (coluna = 0, linha = 0)
lcd.print("Nivel:");
lcd.setCursor(6,0); // seta o cursor para: (coluna = 13, linha = 0)
lcd.print(proporcao);
lcd.setCursor(9,0); // seta o cursor para: (coluna = 13, linha = 0)
lcd.print("%");
lcd.setCursor(0,1); // seta o cursor para: (coluna = 0, linha = 0)
lcd.print("Temp.");
lcd.setCursor(6,1); // seta o cursor para: (coluna = 13, linha = 0)
lcd.print(temperatura);
lcd.setCursor(12,1); // seta o cursor para: (coluna = 13, linha = 0)
lcd.print("C");

// condição para gerar o alarme de segurança.
if (proporcao>=90 || temperatura >=60 || proporcao <=-1 )
{
  digitalWrite (ledPinalarm, HIGH); // nível alto no pino 10.
  delay (1000); // mantém estado alto por 1 segundo.
  digitalWrite (ledPinalarm,LOW); // nível baixo pino 10.
  delay (1000); // matem estado baixo por 1 segundo.
  digitalWrite (resPin, LOW); // desliga resistência.
  digitalWrite (mixPin, LOW); // desliga agitador.
  digitalWrite (ledPinres,LOW); // desliga led da resistência.
  digitalWrite (ledPinmix,LOW); // desliga led do agitador.
}

// condição: se o nível for (>=80%) e (<=89%).
if (proporcao>=70 && proporcao<=89)
{
  digitalWrite (vonPin, LOW); // fecha válvula de entrada.
  digitalWrite (mixPin, HIGH); // liga o agitador.
  digitalWrite (ledPinmix, HIGH); // liga led correspondente do agitador.
  digitalWrite (resPin, HIGH); // liga resistência.
  digitalWrite (ledPinres, HIGH); // liga led correspondente da resistência.
  digitalWrite (ledPinvon, LOW); // desliga led da válvula de entrada.
}

// condição: se nível for (>=80%) e temperatura da solução for (>=55°C).
if (proporcao >=70 && temperatura >55)
{
  digitalWrite (mixPin,HIGH); // liga o agitador.
  digitalWrite (resPin,LOW); // desliga a resistência.
  digitalWrite (ledPinres,LOW); // desliga o led resistência.
  digitalWrite (msPin,HIGH); // liga bomba de saída.
  digitalWrite (vsPin,HIGH); // abre válvula da saída.
}

```

```
digitalWrite (vonPin,LOW); // fecha válvula de entrada.
digitalWrite (ledPinvs,HIGH); // liga led válvula de saída.
digitalWrite (ledPinms, HIGH); // liga led motor de saída.
}

//condição: se nível for <10%.
if (proporcao <10)
{
digitalWrite (resPin,LOW); // mantém desligada resistência.
digitalWrite (msPin,LOW); // desliga bomba de saída.
digitalWrite (vsPin,LOW); // fecha válvula de saída.
digitalWrite (mixPin,LOW); // desliga o agitador.
digitalWrite (ledPinvs, LOW); // desliga led da válvula de saída.
digitalWrite (ledPinms, LOW); // desliga led motor de saída.
digitalWrite (ledPinmix, LOW); // desliga led do agitador.
}

delay (1000);
}
```

