

Controle de Processo Industrial Genérico com interface entre CLP e Microcontrolador

André Filipe Obenaus – Júlio Joel da Costa Neto – Pedro Arthur S. P. Cogliatti

Engenharia de Controle e Automação
Universidade Federal de Santa Catarina – Campus Blumenau
Blumenau, Brasil

Resumo — Este artigo tem por fim demonstrar de maneira didática como utilizar um controlador lógico programável (CLP) em conjunto com um microcontrolador de licença livre. A montagem será utilizada para controlar um processo industrial genérico em escala reduzida, e a necessidade de utilizar o microcontrolador é para possibilitar o uso de sensores e atuadores mais baratos e amplamente disponíveis no ambiente acadêmico e o no mercado, gerando assim maquetes funcionais de conteúdos teóricos ensinados em sala de aula.

Palavras-chave — *aprender fazendo; controle; CLP; ensino prático; modelagem; sistema de nível*

I. INTRODUÇÃO

O ensino prático nas engenharias, além de auxiliar o aluno na compreensão dos conceitos das disciplinas, também visa simular situações reais da indústria para preparar o aluno ao mercado de trabalho, porém, existe uma lacuna entre o ambiente acadêmico e fabril devido ao fato de nem sempre ser viável reproduzir os processos estudados em um laboratório. Uma maneira de contornar esse problema é utilizando simulações em software, que modelam com grande confiabilidade casos clássicos de estudos técnicos. Mas, apesar dos grandes avanços em softwares de modelagem e simulação, eles ainda não substituem totalmente o “aprender fazendo”, de modo que a construção de modelos reais e funcionais continua sendo o método que melhor preenche a lacuna mencionada anteriormente.

Este artigo visa proporcionar uma base teórica e prática para a construção de um sistema de nível de tanque, processo clássico da indústria, descrito por [1]. Na Seção II estão apresentados o desenvolvimento teórico, equações, imagens e definições que serviram de base para o trabalho. Na Seção III, é listado o material utilizado, e suas características. A Seção IV contém as informações referentes ao desenvolvimento prático e simulação do processo. O artigo é concluído na Seção V.

II. DESENVOLVIMENTO TEÓRICO

O processo desenvolvido pode ser dividido em duas partes. Uma envolve o problema de controle de nível de líquido em um tanque, onde é necessário descrever matematicamente o comportamento do tanque e do controle que aciona uma bomba responsável pela vazão de entrada no sistema. A outra é implementada no CLP, que dá o comando de início para o controle, além de ativar outras etapas do processo descrito a seguir.

A. Modelo Matemático da Planta de Controle

Em análise de sistemas que envolvem fluxo de fluidos, é necessário dividir os regimes de fluxo em regime laminar e regime turbulento, de acordo com o valor do número de Reynolds (Re) [1].

Para $Re > 4000$, o sistema é turbulento. Para valores de $Re \leq 2300$, o regime é considerado laminar. A região intermediária é considerada de transição, onde ambos os regimes são possíveis, dependendo, entre outros fatores, de rugosidade do tubo e uniformidade do fluxo.

A importância dessa distinção é que fluxos laminares podem ser representados por equações diferenciais lineares, ao passo que fluxos turbulentos são representados por equações diferenciais não lineares.

Processos industriais, de maneira geral, têm escoamentos turbulentos, porém, se a região de operação for limitada, as equações que descrevem esses processos podem ser linearizadas. Utilizam-se conceitos de resistência e capacitância para sistemas de nível de líquido que permite descrever suas características dinâmicas de modo simples [1]. A Fig. 1 descreve o diagrama de funcionamento do sistema:

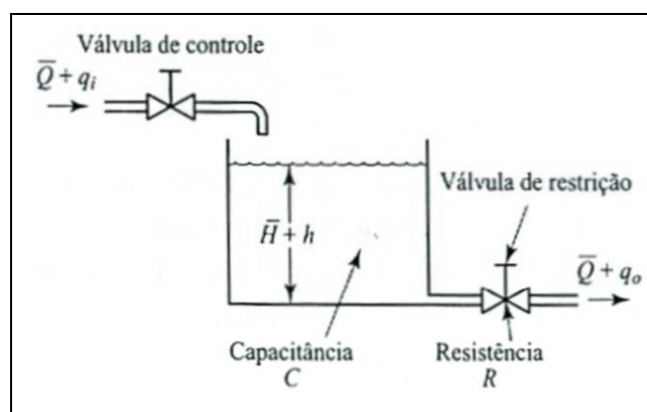


Fig. 1. – Diagrama de sistema de nível de primeira ordem [1].

Onde, para em regime permanente (RP):

\bar{Q} = vazão em volume em RP

q_i = desvio da vazão de entrada em relação ao RP

q_o = desvio da vazão de saída em relação ao RP

\bar{H} = altura do nível em RP

h = desvio de nível a partir do seu valor em RP

Considerando um fluxo ao longo de uma tubulação curta ou através de uma restrição, a resistência R é definida como a variação na diferença de nível necessária para causar a variação unitária na vazão, isto é:

$$R = \frac{\text{variação na diferença de nível (m)}}{\text{variação na vazão em volume } (\frac{m^3}{s})} \quad (1)$$

Para um fluxo turbulento no regime permanente, a resistência R_t é aproximadamente:

$$R_t = \frac{2\bar{H}}{\bar{Q}} \quad (2)$$

Esta relação é verdadeira para a condição de operação nas proximidades de um ponto P. Se definirmos uma pequena variação a altura de RP como h a pequena variação correspondente da vazão como q , a validade da equação de R_t pode ser vista na Fig. 2 abaixo.

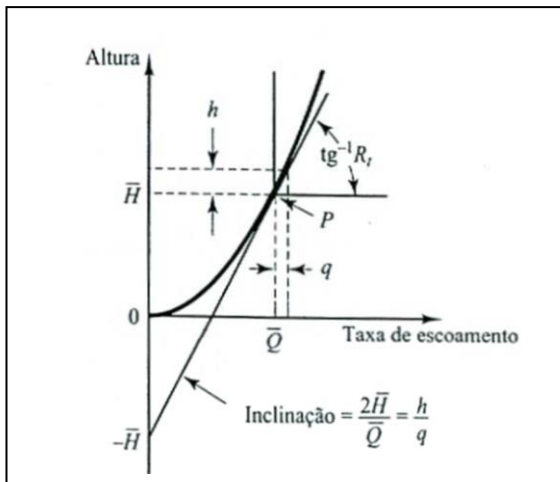


Fig. 2. – Curva real de operação [1].

Como se pode observar, em um ponto P de operação, se as variações de altura e vazão forem suficientemente pequenas, a inclinação da curva real no ponto é igual à inclinação da reta tangente ao ponto.

A capacitância C de um reservatório é definida como a variação na quantidade de líquido armazenada necessária para causar a mudança unitária no potencial (altura), então,

$$C = \frac{\text{variação no volume armazenado (m}^3\text{)}}{\text{variação na altura (m)}} \quad (3)$$

A capacitância do reservatório é igual à sua seção transversal. Se esta for constante, a capacitância será constante para qualquer altura do nível. Para o caso de um tanque cilíndrico, a área A da seção transversal é,

$$A = C = \frac{\pi D^2}{4} \quad (4)$$

Onde D é o diâmetro do tanque.

Agora que definimos as variáveis do sistema, e restringimos sua condição de operação para um regime linearizado, podemos descrever seu comportamento com a equação a seguir:

$$C dh = (q_i - q_o) dt \quad (5)$$

Da equação (1):

$$q_o = \frac{h}{R} \quad (6)$$

Para um sistema linear, R é constante e reescrevemos (5) como:

$$RC \frac{dh}{dt} + h = Rq_i \quad (7)$$

Aplicando a transformada de Laplace em ambos os membros da equação e considerando condições iniciais nulas, obtemos:

$$(RCs + 1)H(s) = RQ_i(s) \quad (8)$$

Para q_i como entrada e h como saída, a função de transferência do sistema é:

$$\frac{H(s)}{Q_i(s)} = \frac{R}{RCs + 1} \quad (9)$$

B. O Controle Proporcional-Integral

O método escolhido para efetuar o controle do sistema é o proporcional-integral (PI), formado por duas componentes. A componente proporcional depende apenas da diferença entre o ponto de ajuste e a variável de processo. Esta diferença é referida como o termo de erro. O ganho proporcional determina a taxa de resposta de saída para o sinal de erro. Em geral, aumentando o ganho proporcional irá aumentar a velocidade da resposta do sistema de controle. No entanto, se o ganho proporcional é muito grande, a variável de processo começará a oscilar. Se o ganho é aumentado ainda mais, as oscilações ficarão maiores e o sistema ficará instável e poderá oscilar até mesmo fora de controle. A componente integral soma o termo de erro ao longo do tempo. O resultado é que mesmo um pequeno erro fará com que a componente integral aumente lentamente. A resposta integral irá aumentando ao longo do tempo a menos que o erro seja zero, portanto, o efeito é o de conduzir o erro de estado estacionário para zero. O regime permanente de erro é a diferença final entre as variáveis do processo e da referência [8].

A equação de um controle PI no domínio da frequência pode ser escrita como:

$$G(s) = K_p + \frac{K_i}{s} \quad (10)$$

Onde K_p é o ganho proporcional e K_i o ganho do integrador.

C. Mapa de Entradas e Saídas e Diagrama de Estados

O processo tem seu início definido por comandos dados ao CLP, que inicia a rotina de controle.

Antes de implementar o código, porém, é importante mapear as entradas e saídas do sistema, bem como construir seu diagrama de estados, de modo a ter o processo bem documentado. O mapeamento de entradas e saídas pode ser visto na Tabela 1, sendo que a letra em frente ao rótulo indica se a porta usada é digital ou analógica.

TABELA I – Mapa de entradas e saídas

Entradas			Saídas		
D	B1	Botão de partida	D	M1	Bomba
D	BE	Botão de emergência	D	HE	Aquecedor
A	SD	Sensor Ultrassônico	D	M2	Aerador
D	BD	Botão de desligar	D	LUV	Luz UV
			A	SPoint	Valor de referência
			D	AL	Alarme

O diagrama de estados finito é encontrado na Fig. 3.

III. MATERIAIS

A. O Controlador Lógico Programável

O Controlador Lógico Programável (CLP) é um equipamento composto de componentes eletrônicos, memória programável ou não programável que contém dados e programas com a finalidade de ler e executar instruções, interagindo com um sistema que deve ser controlado, através de dispositivos de input e output do tipo digital ou analógico (norma IEC 61131-2).

O CLP é um computador que realiza funções de controle em vários níveis de complexidade. O aspecto interessante do CLP é que ele pode ser programado e utilizado por pessoas sem um grande conhecimento no uso do computador. É projetado para trabalhar em ambiente industrial com variação de temperatura, umidade, vibrações, distúrbios elétricos e outras variantes que existem no meio industrial [5].

O CLP utilizado é o SIMATIC S7-1200 da marca Siemens, disponível para utilização dos acadêmicos no LABCID – Laboratório de Circuitos e Sistemas Digitais.

B. O Microcontrolador e Placa de Prototipagem

O microcontrolador é um circuito integrado encapsulado e destinado ao processamento de sinais digitais. Em geral, possuem todos os periféricos necessários em um único chip. Seu tamanho é reduzido, mesmo contendo vários periféricos, como memória, barramento, *timers*, portas de comunicação, conversores analógico-digital, etc. Possuem desempenho menor que os microprocessadores, mas são ideais em aplicações que necessitam de menores dimensões, tempo e custos. É amplamente utilizado em automação industrial, residencial e predial, eletrodomésticos, brinquedos, eletrônicos, e em qualquer situação onde seja necessário o controle de um dispositivo de sinais eletrônico [10].

O microcontrolador utilizado é o ATmega328, instalado na placa de prototipagem Arduino UNO.

Arduino é uma plataforma de prototipagem de código aberto baseado em hardware e software *easy-to-use*. Placas Arduino são capazes de ler entradas e transformá-las em uma ou mais saídas.

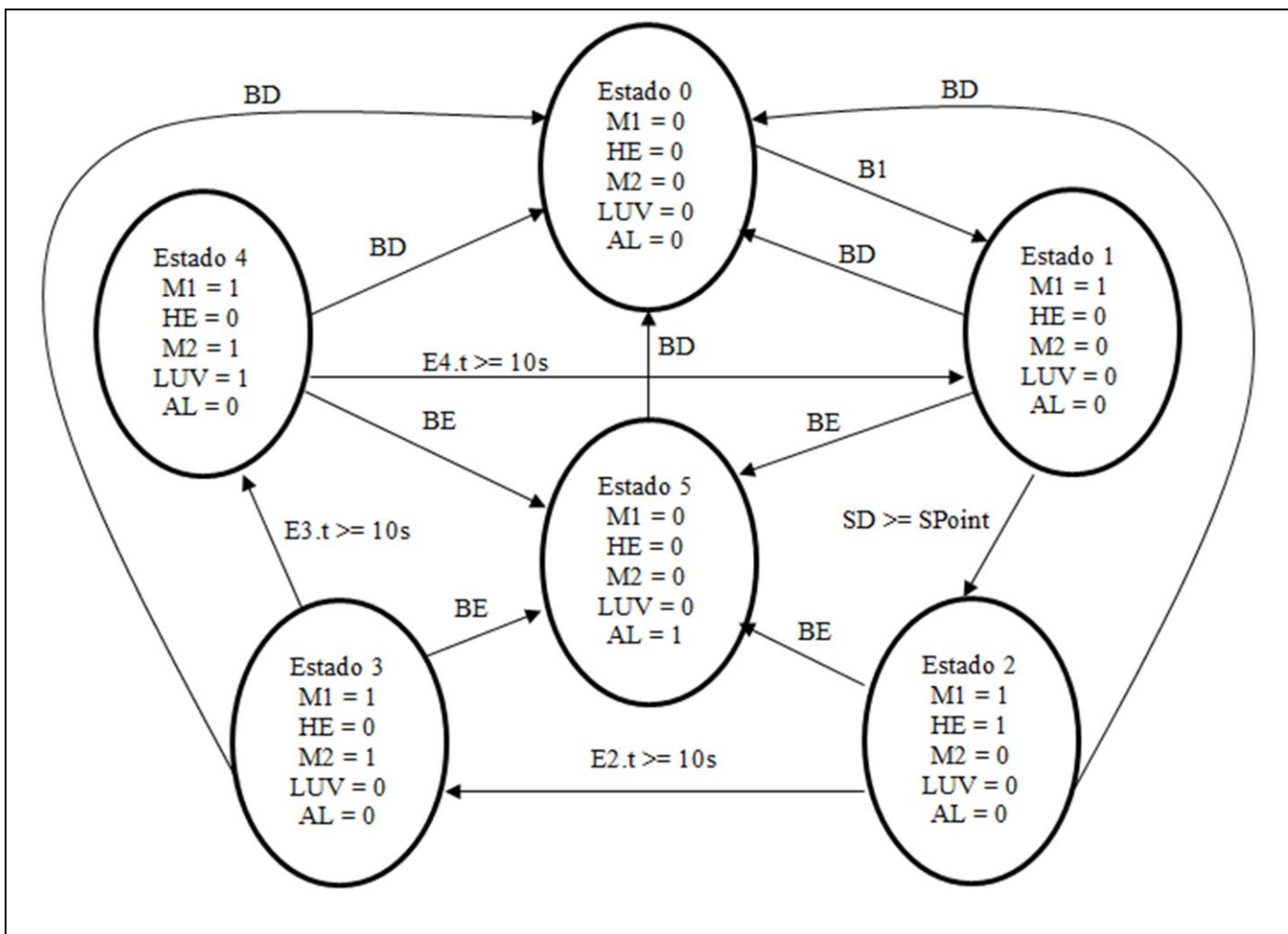


Fig. 3 – Diagrama de máquina de estados finitos.

Para fazer isso se usa a linguagem de programação Arduino (baseada em conexões por fios), e o software Arduino (IDE), com base em processamento.

As placas Arduino são completamente *open-source*, capacitando os usuários para construí-las de forma independente e adaptá-las às suas necessidades específicas. O software também é *open-source* [9].

C. Motobomba e Sensor de Nível

A bomba utilizada para encher o tanque é uma bomba de gasolina para automóveis, de 12 volts CC.

A rotação do motor é controlada com uma ponte H, baseado no chip L298N, construído para controlar cargas indutivas como relés, solenoides, motores DC e motores de passo.

O sensor usado para medir o nível é o sensor ultrassônico HC-SR04, comumente utilizado com a plataforma Arduino.

Quando um pulso de disparo é enviado para a entrada *trigger*, o sensor emite uma rajada de oito sinais ultrassônicos de 40 kHz, e o tempo de eco desses sinais é proporcional à distância de medição. Parâmetros do sensor são encontrados na Tabela II.

TABELA II – Parâmetros do sensor

Tensão de operação	5 V CC
Corrente de operação	15 mA
Frequência de operação	40 Hz
Alcance máximo	4 m
Alcance mínimo	2 cm
Ângulo de medição	15 graus
Sinal de disparo de entrada	10 μ s pulso TTL
Sinal de eco de saída	Sinal TTL de entrada e o alcance proporcional
Dimensões	45x20x15 mm

IV. DESENVOLVIMENTO PRÁTICO

Uma foto do sistema construído pode ser vista na Fig. 4.

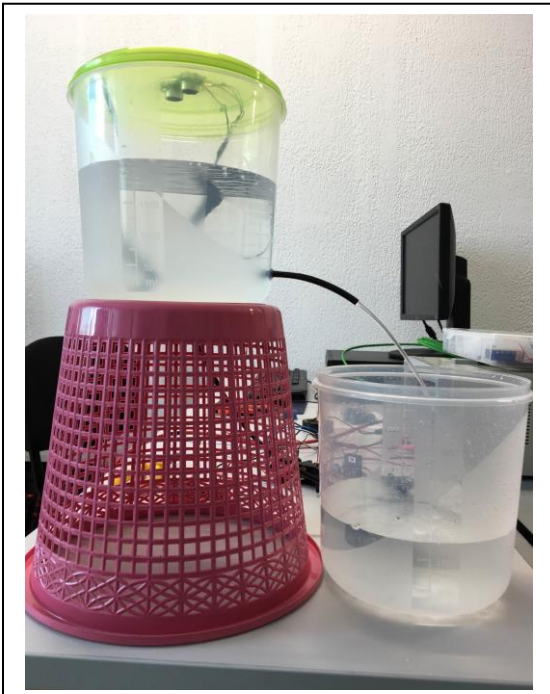


Fig. 4 – Tanque de controle e reservatório.

A. Parâmetros C e R_t

Utilizando um pote plástico cilíndrico de quatro litros como o tanque e outro como reservatório, podemos definir os parâmetros da planta, conforme visto na seção II.

O parâmetro C foi definido como sendo a área da seção do tanque. Tendo o diâmetro de 18 cm então:

$$C = 25,45 \cdot 10^{-3} \text{ m}^2 \quad (11)$$

A vazão de saída e regime permanente pode ser definida como:

$$\bar{Q} = C_d \cdot A_o \cdot \sqrt{2 \cdot g \cdot \bar{H}} \quad (12)$$

Onde A_o é a área do tubo de vazão e C_d é o coeficiente de descarga, uma constante adimensional obtida experimentalmente.

Portanto, R pode ser calculado como:

$$R_t = \frac{2\bar{H}}{C_d \cdot A_o \cdot \sqrt{2 \cdot g \cdot \bar{H}}} \quad (13)$$

A saída do tanque é feita por um tubo de 10 cm de comprimento por 4 mm de diâmetro, sendo classificado como um tubo curto.

Considera-se um tubo curto aquele em que seu comprimento (L) supera três vezes o seu diâmetro e fica abaixo de quinhentas vezes o seu diâmetro.

Da referência [2], obtemos valores práticos de C_d que para $L/D = 25$, $C_d = 0,715$.

Com $\bar{H} = 10 \text{ cm}$;

$$R_t = 15891,5 \frac{\text{s}}{\text{m}^2} \quad (14)$$

A função de transferência fica:

$$\frac{H(s)}{Q_i(s)} = \frac{15891,5}{404,44s + 1} \quad (15)$$

O diagrama de blocos para o sistema de controle é apresentado na Fig. 5.

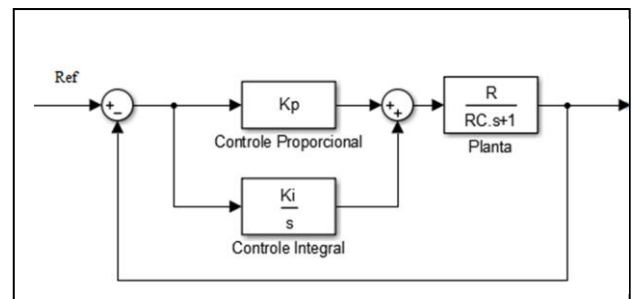


Fig. 5 – Diagrama de blocos do controle e planta.

O atuador tem uma resposta muito mais rápida do que a planta para que seja possível efetuar um controle, ou seja, seus polos estão muito afastados da origem, reduzindo o sua influencia na resposta total do sistema. Por este motivo, vamos desconsiderar a parcela de ganho do atuador, conforme o esquema da Fig. 5.

B. Controle com Arduino

a) O microcontrolador foi programado utilizando IDE próprio da plataforma, que utiliza linguagem C, e o código de controle utilizado foi o de uma biblioteca PID encontrada em [3], onde zeramos o termo relacionado ao controle diferencial.

A determinação dos termos K_i e K_p foi feita utilizando a ferramenta computacional Matlab. O requisitos para o ajuste foi de menor tempo de subida sem que haja *overshoot*. Devido à natureza de medição do sensor, convertemos os valores de C e R_t para cm^2 e s/cm^2 , e usando a opção de ajuste do software, obtivemos:

$$K_i = 0,5 \quad (16)$$

$$K_p = 45 \quad (17)$$

A simulação feita no computador da resposta no tempo do sistema é apresentada na Fig. 6.

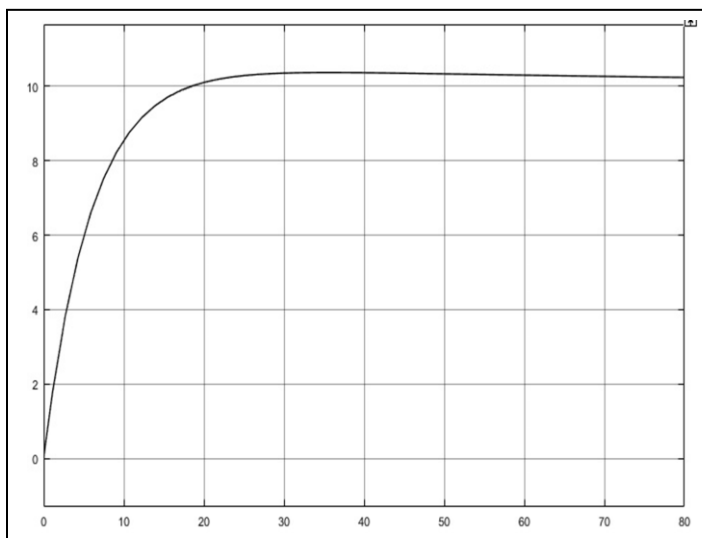


Fig. 6 – Gráfico da resposta do sistema.

O modelo físico atinge a altura de referência no tempo previsto pela simulação, em torno de vinte segundos, e leva um tempo mais longo para estabilizar, o que também era esperado.

C. Controle no CLP

Três linguagens foram utilizadas na programação do CLP, a linguagem Ladder, texto estruturado e FBD (diagramas de blocos). Todas estão definidas na norma IEC 61131-3.

A linguagem Ladder foi criada para melhorar a aceitação do CLP no mercado, pois emula a lógica tradicional de controle que era representada por diagrama de relés. É uma linguagem gráfica de alto nível que representa um circuito elétrico [4]. Foi a linguagem utilizada para programar as principais partes do diagrama de estados da Fig. 3.

O FBD é uma linguagem gráfica onde os elementos aparecem interligados como em circuitos elétricos. Usa

blocos padrão semelhantes aos do Ladder e permite aos usuários construir seus próprios blocos de função de acordo com a exigência da aplicação, o que se chama de encapsulamento.

O bloco pode ser programado em qualquer outra linguagem de programação [4]. Esta linguagem foi utilizada para converter valores em centímetros para valores de tensão, para comunicar o valor de referência para o Arduino e também fazer a leitura dos valores medidos e converter em unidades convenientes de se apresentar ao usuário.

O texto estruturado, ou em inglês *structured text* (ST), é uma linguagem de texto de alto nível, similar a BASIC e PASCAL. Permite a implementação de procedimentos complexos, difíceis de representar em linguagens gráficas. Possibilita o uso de laços de repetição (*while-do*) e de condições (*if-then-else*) [4]. Essa linguagem foi utilizada para fazer uma média das medições enviadas pelo sensor, para reduzir o efeito de erros de precisão e ruído.

Além da programação, o CLP permite criar uma interface homem-máquina (IHM), que pode ser implementado em uma tela do tipo *touchscreen*, onde é possível acompanhar e controlar parâmetros que o programador julgue necessário. A interface criada pode ser vista na Fig. 7.

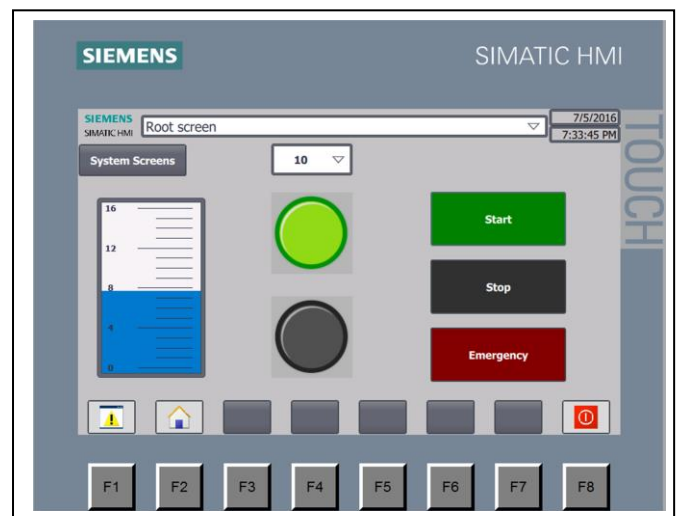


Fig. 7 – IHM com indicador de nível e LED de operação ativado.

Além da IHM, também foi desenvolvida a aplicação *webserver*, que permite controlar o processo através de um computador com acesso a mesma rede do CLP. Essa funcionalidade cria uma página em HTML, na qual o programador pode liberar ou restringir o acesso de usuários mediante o uso de senha, assim como liberar ou restringir o acesso e controle de parâmetros e informações conforme a necessidade.

As páginas da aplicação podem ser vistas nas Fig. 8 e Fig. 9.

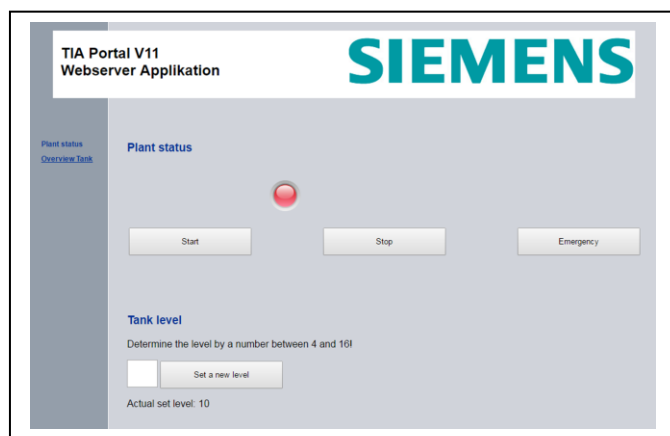


Fig. 8 – Página de comando do processo.

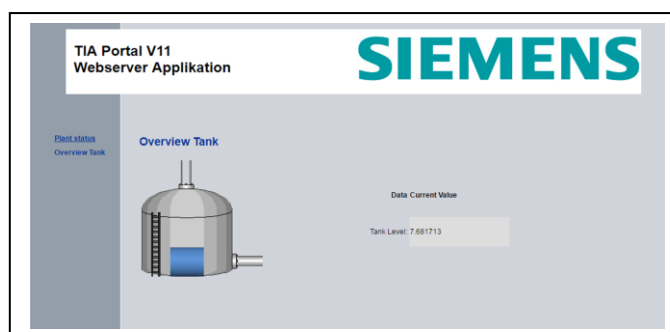


Fig. 9 – Página de visão geral do processo.

V. CONCLUSÃO

Durante a montagem do projeto, muitas dúvidas de natureza prática surgiram, e foram posteriormente solucionadas através de pesquisa e ajuda do corpo docente. O controle PI foi implementado com sucesso, apesar de pequenas variações causadas pela precisão do sensor. O controle do CLP operou conforme projeto, sem ressalvas. O artigo avançou o entendimento dos autores nos conteúdos aqui abordados, trazendo sentido prático para conteúdos antes muito abstratos. A sugestão para trabalhos futuros é usar o controle PI industrial do CLP, que apresenta melhor desempenho, utilizando o Arduino apenas como interface para os sensores.

AGRADECIMENTOS

Os autores agradecem o corpo docente da Engenharia de Controle e Automação – UFSC – Campus Blumenau, pelos recursos e apoios prestados ao desenvolvimento deste projeto.

REFERÊNCIAS

- [1] K. Ogata, Engenharia de Controle Moderno, 5ª ed. São Paulo: Pearson Prentice Hall, 2010, pp 92-95.
- [2] J. M. Azevedo Netto, Manual de Hidráulica, 8ª ed. São Paulo: Edgar Blüchler, 1998, pp.84-85.
- [3] (<http://playground.arduino.cc/Code/PIDLibrary>) PID Library, acessado em 06/07/2016.

- [4] M.R. Petry – Notas de aula - Informática Industrial, 2016.
- [5] F. Prudente, PLC S7-1200 – Teoria e Aplicações, 1ª ed. Rio de Janeiro: LTC, 2014.
- [6] B.P. Lathi, Sinais e Sistemas Lineares, 2ª ed. Porto Alegre: Bookman, 2008.
- [7] G.F. Franklin; J.D. Powell; A. Emami-Naeini. Sistemas de Controle para Engenharia. 6. ed. Porto Alegre: Bookman, 2013.
- [8] (<http://www.ni.com/white-paper/3782/pt/>) National Instruments - Explicando a Teoria PID, acessado em 12/07/2016.
- [9] (<https://www.arduino.cc/en/Guide/Introduction>) What is Arduino, acessado em 12/07/2016.
- [10] A.S. de Oliveira, F. S. de Andrade, Sistemas Embarcados: hardware e firmware na prática, 2ª ed. São Paulo: Érica, 2010, pp 34.