

INSTITUTO FEDERAL DO ESPÍRITO SANTO
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

ANDREW SIMÕES NETO
BRUNO VIANA DE MELLO

SISTEMA INTEGRADO DE SUPERVISÃO CONTROLE E ATUAÇÃO

SERRA
2019

INSTITUTO FEDERAL DO ESPÍRITO SANTO
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

ANDREW SIMÕES NETO
BRUNO VIANA DE MELLO

REDES INDUSTRIAIS
SISTEMA INTEGRADO DE SUPERVISÃO CONTROLE E ATUAÇÃO

Relatório técnico referente a atividade prática apresentado como requisito parcial para aprovação na disciplina de Redes Industriais do curso de Engenharia de Controle e Automação do Instituto Federal do Espírito Santo – IFES, Campus Serra.

SERRA
2019

Sumário

1 Resumo.....	1
2 Objetivos	1
3 Implementação do sistema	1
3.1 Planta: Controle de luminosidade.....	1
3.1.1 Sensores e atuadores	3
3.2 Sistema de controle.....	4
3.3 Sistema de supervisão.....	5
4 Desempenho e segurança da rede.....	5
5 Resultados e discussões	6
6 Conclusão	7

1 Resumo

Este relatório descreve em detalhes todos os procedimentos adotados para configuração e execução de um sistema integrado de supervisão controle e atuação de uma planta didática com utilização dos *softwares* de controle OpenPLC e de supervisão ScadaBR.

2 Objetivos

Implementar um sistema didático e de baixo custo que contemple os três primeiros níveis da pirâmide da automação: “chão de fábrica”, controle e automação. A planta a ser controlada deve ser simples e de baixo custo, com objetivos apenas didáticos e deve operar em malha fechada. A leitura da variável de processo e o sistema de atuação devem ser executados por elementos diferentes, bem como o sistema de controle deve estar implementado em hardware distinto. Todos os níveis devem se comunicar através de uma rede sem fio dotada de algum nível de segurança e com alguma garantia de comunicação e determinismo utilizando algum protocolo de rede industrial (aka, TCP/IP, Modbus, Hart, CAN, OPCUA, etc).

3 Implementação do sistema

A implementação do sistema se deu da seguinte maneira:

- Montagem de uma planta para controle de luminosidade com uso de LED e LDR;
- Instalação do OpenPLC em uma Raspberry PI 2, com uso de dois escravos utilizando-se ESP32;
- Instalação do ScadaBR em ambiente *Windows* para supervisão;

A conexão entre os hosts foi implementada via *wifi*, com uso de roteador TP-Link, com exceção da Raspberry PI, que não possui módulo de rede sem fio e foi conectada via cabo ethernet. Como padrão de utilização dos softwares citados, o protocolo de comunicação adotado foi o Modbus.

Maiores detalhes sobre cada elemento serão expostos nos tópicos a seguir.

3.1 Planta: Controle de luminosidade

A planta utilizada neste trabalho é um sistema de controle de luminosidade, sendo o atuador um LED e o sensor um LDR conforme figura 1.

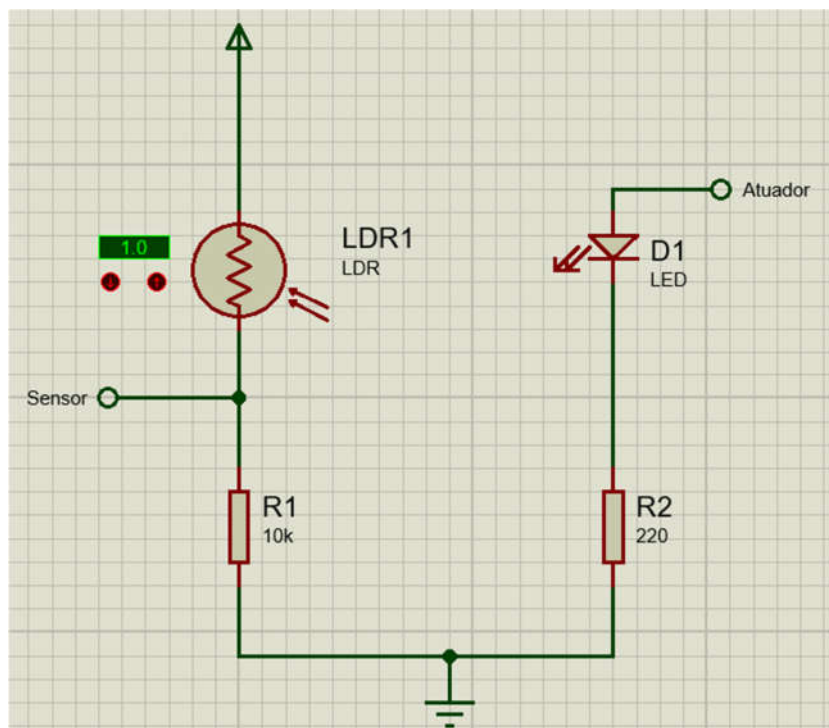


Figura 1 – Diagrama esquemático da planta a ser controlada

Quanto maior a iluminação gerada pelo LED sobre o LDR, menor será sua resistência e maior será a tensão sobre o resistor R1, a qual serve como sinal de saída da planta. Este sinal de saída é enviado para o controlador e comparado com o *set-point*. A diferença entre os dois valores serve como parâmetro de entrada para o sistema de atuação que irá aumentar ou diminuir a tensão aplicada sobre o LED afim de variar sua luminosidade.

O sistema de controle funciona como ilustrado no diagrama de blocos da figura 2.

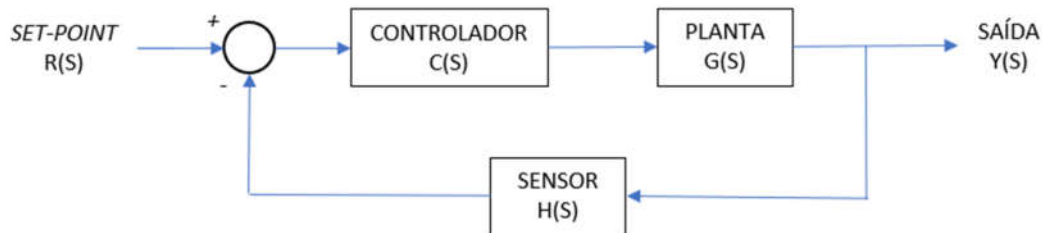


Figura 2 – Diagrama de blocos do sistema de controle

A montagem física dos componentes pode ser visualizada na figura 3.

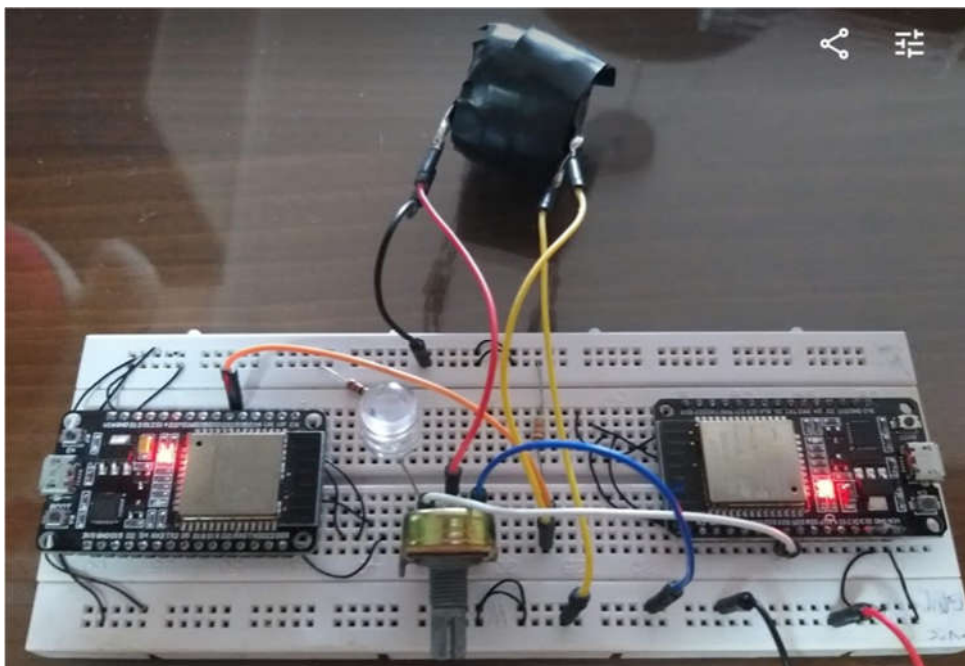


Figura 3 – Montagem física da planta

Para evitar distúrbios na medição causados pela iluminação ambiente, o LDR e o LED de controle foram encapsulados em uma pequena câmara escura. Para possibilitar a visualização do sistema de atuação foi inserido um LED externo adicional, que recebe o mesmo sinal do LED de controle.

O controle eficaz desta planta depende de sua correta modelagem, bem como da sintonia adequada do controlador. Contudo, por este não ser o foco deste trabalho e por questão de simplificação, esta etapa não foi executada

3.1.1 Sensores e atuadores

Os sensores e atuadores foram implementados nas ESP32 utilizadas como escravos do PLC. Foram utilizados 2 microcontroladores, um responsável pela leitura do sinal de saída e um responsável pelo sinal de controle, ambos utilizando as portas de comunicação analógicas. Conforme já mencionado, a comunicação com o “master” se dá via wifi. A figura 4 apresenta os detalhes sobre as entradas e saídas dos escravos configurados no PLC.

Slave Devices

List of Slave devices attached to OpenPLC.

Attention: Slave devices are attached to address 100 onward (i.e. %IX100.0, %IW100, %QX100.0, and %QW100)

Device Name	Device Type	DI	DO	AI	AO
Sensor	ESP32	%IX100.0 to %IX100.7	%QX100.0 to %QX100.7	%IW100 to %IW100	%QW100 to %QW100
Controle	ESP32	%IX101.0 to %IX101.7	%QX101.0 to %QX101.7	%IW101 to %IW101	%QW101 to %QW101

Figura 4 - Detalhes dos escravos do PLC

3.2 Sistema de controle

O equipamento designado para controle foi uma Raspberry PI 2 com o OpenPLC instalado. A lógica do controlador do sistema foi implementada em linguagem Ladder conforme ilustrado na figura 5.

#	Nome	Class	Tipo	Localização	Valor Inicial	Opção	Documentação
1	PID0	Local	PID				
2	Setpoint	Local	INT	%QW1			
3	EN_PID	Local	BOOL	%QX0.3			
4	KP	Local	INT	%QW2			
5	TR	Local	INT	%QW3			
6	TD	Local	INT	%QW4			
7	VP	Local	INT	%IW100			
8	SC	Local	INT	%QW101			
9	QUO	Local	INT	%QW5			

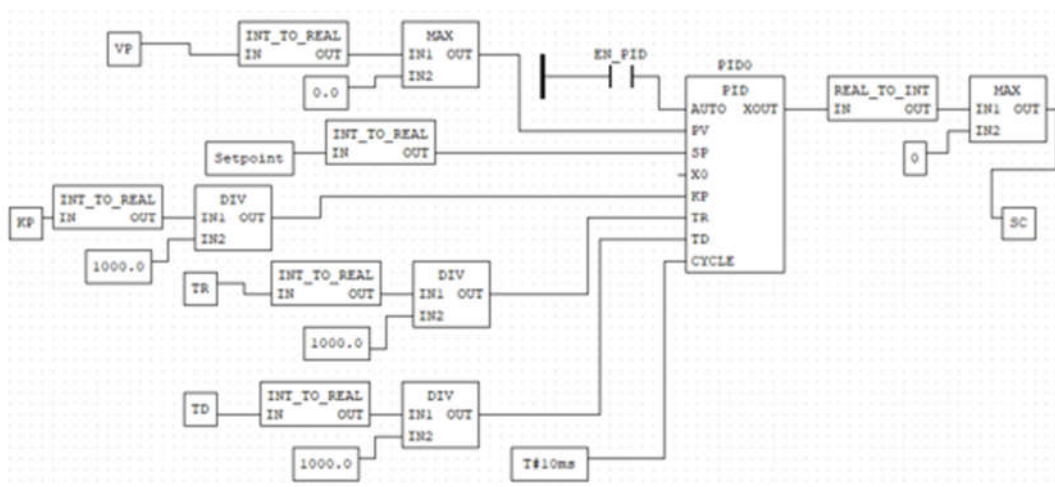


Figura 5 - Lógica do controlador em Ladder

O controlador principal adotado foi o PID, cuja saída é relacionada com a proporção, acúmulo e taxa de variação do erro conforme a equação:

$$PID = K_p * e(t) + K_i * \int_0^t e(t) + K_d * \frac{de(t)}{dt}$$

Um detalhe que foi necessário ser levado em conta, foi com respeito ao formato e ordem de grandeza dos sinais, que precisaram passar por conversões e operações aritméticas para compatibilização. Um exemplo é o sinal de saída do controlador, que é recebido como ponto flutuante e deve ser convertido para inteiro com bloqueio dos valores negativos (não pode ser convertido para inteiro sem sinal devido incompatibilidade com o sistema).

3.3 Sistema de supervisão

O sistema de supervisão foi implementado com o uso do ScadaBR. Foi desenvolvida uma única tela para controle do processo. Esta tela possui controles para os ajustes do controlador e do set-point, bem como permite a visualização das variáveis de saída. Uma amostra da tela de controle pode ser vista na figura 6.



Figura 6 - Tela de interface do sistema supervisório

4 Desempenho e segurança da rede

Durante os testes operacionais foram detectados alguns problemas e restrições de comunicação, principalmente no que diz respeito ao tempo de acesso aos hosts, como foi o caso da perda de comunicação entre os escravos e o PLC devido ao período de *pooling*. Outro problema detectado foi o curto espaço operacional devido à presença de fontes de interferência (paredes e outros objetos) e à baixa potência do sinal.

Os sistemas de segurança adotados foram apenas os padrões dos sistemas, a saber, chave WAPA2 PSK para conexão dos clientes e necessidade de autenticação nos sistemas operativos. O próprio roteador utilizado permite ainda a implementação de *firewall* e a filtragem de IP's contudo esses recursos não foram utilizados.

Dadas as devidas ressalvas, a rede apresentou desempenho satisfatório, com boa vazão e velocidade de resposta, devido, em parte, por ser dedicada somente ao sistema de controle. Como medida de controle adicional, os IP's dos equipamentos foram definidos como fixos no servidor de DHCP do roteador: PLC – 192.168.1.101, Escravos – 192.168.1.102 e 192.168.1.103. A figura 7 ilustra alguns dos resultados de desempenho da rede.


```

C:\Users\bruno>ping 192.168.1.101
Disparando 192.168.1.101 com 32 bytes de dados:
Resposta de 192.168.1.101: bytes=32 tempo=52ms TTL=64
Resposta de 192.168.1.101: bytes=32 tempo=21ms TTL=64
Resposta de 192.168.1.101: bytes=32 tempo=10ms TTL=64
Resposta de 192.168.1.101: bytes=32 tempo=15ms TTL=64

Estatísticas do Ping para 192.168.1.101:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
    Aproximar um número redondo de vezes em milissegundos:
        Mínimo = 10ms, Máximo = 52ms, Média = 24ms

C:\Users\bruno>tracert 192.168.1.101
Rastreamento a rota para 192.168.1.101 com no máximo 30 saltos
    1  12 ms  30 ms  33 ms  192.168.1.101
Rastreamento concluído.

C:\Users\bruno>tracert 192.168.1.102
Rastreamento a rota para 192.168.1.102 com no máximo 30 saltos
    1  14 ms  13 ms  23 ms  192.168.1.102
Rastreamento concluído.

C:\Users\bruno>tracert 192.168.1.103
Rastreamento a rota para 192.168.1.103 com no máximo 30 saltos
    1  39 ms  26 ms  6 ms  192.168.1.103
Rastreamento concluído.

pi@raspberrypi:~$ ping 192.168.1.102
PING 192.168.1.102 (192.168.1.102) 56(84) bytes of data:
64 bytes from 192.168.1.102: icmp_seq=1 ttl=255 time=111 ms
64 bytes from 192.168.1.102: icmp_seq=2 ttl=255 time=31.7 ms
64 bytes from 192.168.1.102: icmp_seq=3 ttl=255 time=52.4 ms
64 bytes from 192.168.1.102: icmp_seq=4 ttl=255 time=68.6 ms
64 bytes from 192.168.1.102: icmp_seq=5 ttl=255 time=101 ms
64 bytes from 192.168.1.102: icmp_seq=6 ttl=255 time=120 ms
64 bytes from 192.168.1.102: icmp_seq=7 ttl=255 time=41.7 ms
64 bytes from 192.168.1.102: icmp_seq=8 ttl=255 time=70.5 ms
64 bytes from 192.168.1.102: icmp_seq=9 ttl=255 time=88.6 ms
64 bytes from 192.168.1.102: icmp_seq=10 ttl=255 time=98.8 ms
64 bytes from 192.168.1.102: icmp_seq=11 ttl=255 time=28.9 ms
^C
--- 192.168.1.102 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 25ms
rtt min/avg/max/mdev = 28.939/73.977/120.428/30.811 ms

```

Figura 7 - Desempenho da rede

5 Resultados e discussões

A instalação e configuração dos programas foi realizada de maneira bem intuitiva seguindo-se os tutoriais disponibilizados nos sites dos desenvolvedores. Nas ESP32, por exemplo, apenas foi necessário redefinir os pinos de entrada/saída sem ser necessários grandes modificações no *firmware* padrão.

Alguns problemas surgiram na execução do ScadaBR, como incompatibilidade com a versão Java instalada no computador e necessidade de encerramento de programas que utilizam a mesma porta de comunicação. Também foi necessária a alteração do período de *pooling* dos escravos no OpenPLC, pois o período padrão, de 100ms, se mostrou incompatível com o roteador utilizado. Ademais, a dinâmica da planta, da maneira em que foi configurada, demonstrou ser demasiadamente rápida para os tempos de operação do sistema de controle, o que impossibilitou seu controle efetivo. Modificações, portanto, seriam necessárias, mas conforme já mencionado, não foram implementadas para não fugir ao foco do trabalho.

Feitas as devidas ressalvas, os resultados obtidos foram satisfatórios.

6 Conclusão

A automação dos processos, sejam eles industriais ou não, está cada vez mais presente no nosso cotidiano. No decorrer do tempo, novas soluções vêm surgindo, com melhora na eficácia e redução dos custos. Os *softwares* e *hardwares* utilizados neste trabalho são um claro exemplo disso, pois de certa maneira, permitem que um sistema de controle possa ser implementado até mesmo por pessoas leigas com poucos recursos financeiros para as mais diversas aplicações.

Contudo, vale lembrar que tais aplicações demandam que a busca pelo conhecimento seja contínua. O ScadaBR por exemplo é um software com uma infinidade de possibilidades, mas nem tudo se encontra pronto para uso, exigindo conhecimentos em outras áreas, tais como programação.

Quanto à operacionalidade e segurança da rede, percebe-se um grande ganho na versatilidade do sistema com o uso de redes sem fio. Todavia, faz-se necessário tomar medidas de segurança adequadas e controlar as variações no ambiente que possam provocar interferências.