

**INSTITUTO FEDERAL DO ESPÍRITO SANTO - IFES**  
**ENGENHARIA DE CONTROLE E AUTOMAÇÃO**  
**CAMPUS SERRA**

LUCAS MANTUAN AYRES  
LEONARDO GONÇALVES BATISTA

**IMPLEMENTAÇÃO DE SISTEMA INTEGRADO DE**  
**SUPERVISÃO, CONTROLE E ATUAÇÃO**

SERRA, 2019

LUCAS MANTUAN AYRES  
LEONARDO GONÇALVES BATISTA

**IMPLEMENTAÇÃO DE SISTEMA INTEGRADO DE  
SUPERVISÃO, CONTROLE E ATUAÇÃO**

Relatório técnico apresentado como requisito parcial para obtenção de aprovação na disciplina de Redes Industriais, no Curso de Engenharia de Controle e Automação, no Instituto Federal do Espírito Santo.

Prof. M.e Rafael Emerick Zape de  
Oliveira

SERRA, 2019

## 1. Aspectos gerais do sistema

Na Figura 1 é apresentado o esquemático do sistema projetado, cuja estrutura é dividida em camadas 1, 2 e 3. Os níveis das camadas representam os mesmos níveis da pirâmide da automação, onde camada 1 representa o nível de campo industrial, enquanto as camadas 2 e 3 representam, respectivamente, níveis de controle e supervisão. É interessante frisar que neste trabalho o Modbus foi empregado nas interfaces 1 e 2, e 2 e 3 da pirâmide industrial (Figura 1), respectivamente [1]–[5].

**Camada 1:** Os módulos ESP32 operam somente como dispositivos escravos, e simulam o sistema microcontrolado de um dispositivo de campo inteligente.

**Camada 2:** O OpenPLC Runtime é um PLC *open-source* focado em soluções de baixo custo industrial e está instalado em uma Raspberry Pi 3. O software Editor roda em uma máquina Windows, e suporta os cinco códigos definidos pelo padrão IEC 61131-3: FBD, IL, ST, SFC e LD.

**Camada 3:** ScadaBR é um software SCADA (supervisório) *open source* instalado em uma Máquina Virtual utilizado para criar IHMs.

### Hierarquia de Comunicação:

O ScadaBR se comunica com OpenPLC Runtime sobre Modbus/TCP.

O OpenPLC Runtime executa o programa gerado no OpenPLC Editor, e faz interface com a camada superior e inferior via Modbus/TCP.

O firmware instalado nas placas ESP32 contém a biblioteca Modbus TCP para comunicação com o host OpenPLC (Raspberry Pi 3) como dispositivos escravos.

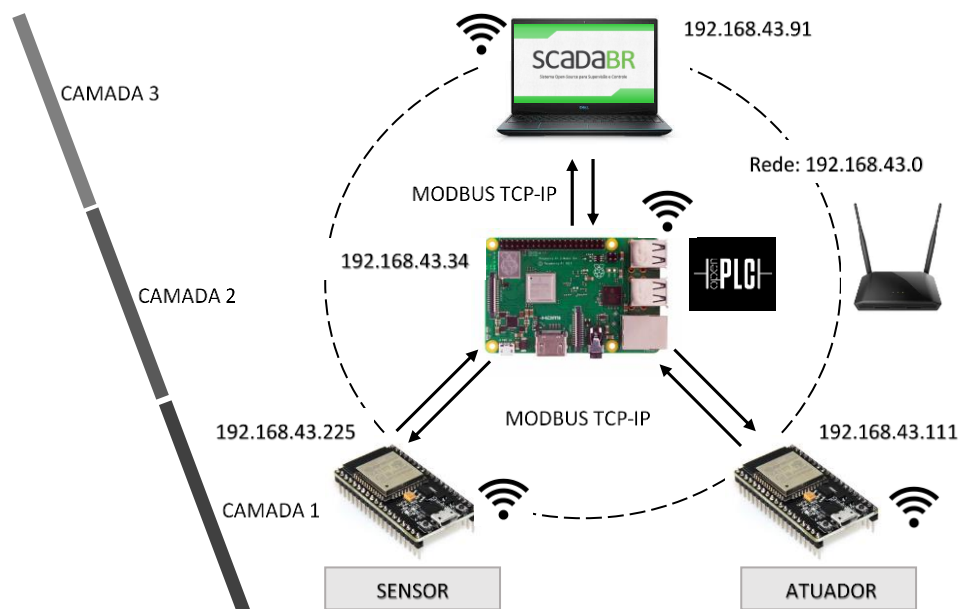


Figura 1 - Esquemático do sistema.

## 2. Mapeamento das Variáveis Modbus

O OpenPLC utiliza o protocolo Modbus/TCP para comunicação com o supervisor. O Modbus é um dos protocolos mais utilizados na indústria, é de código aberto e livre de royalties. Isso torna o OpenPLC compatível com praticamente qualquer software SCADA existente.

O Modbus possui principalmente 4 funções, cada uma responsável pela leitura ou gravação de uma E/S específica. O espaço de endereço de E/S no OpenPLC é mapeado para o espaço de endereço do Modbus de acordo com a Tabela 1 abaixo:

Tabela 1 - Mapeamento de endereços do Modbus para o OpenPLC.

Register Type	Usage	PLC Address	Modbus Address	Register Size	Value Range	Access
Coil Registers	Digital Outputs	%QX0.0 - %QX99.7	0 - 799	1 bit	0 or 1 OFF or ON	read and write
Discrete Input Registers	Digital Inputs	%IX0.0 - %IX99.7	0 - 799	1 bit	0 or 1 OFF or ON	read only
Input Registers	Analog Inputs	%IW0 - %IW99	0 - 1023	16 bits	0 to 65,535	read only
Holding Registers	Analog Outputs	%QW0 - %QW99	0 - 1023	16 bits	0 to 65,535	read and write

O mapeamento para desenvolvimento do sistema, é detalhado a seguir:

#	Nome	Class	Tipo	Localização	Valor Inicial	Opção	Documentação
1	SP	Local	UINT	%QW0			Holding Register - Analog Outputs (R/W)
2	LAMP	Local	UINT	%QW101			Saída analog (PWM) (W) -> ESP32
3	LAMPSCADA	Local	UINT	%IW0			Input Register - Analog Outputs (R)
4	VP	Local	UINT	%IW100			Entrada analog (PWM) (R) --> ESP32
5	VPSCADA	Local	UINT	%QW1			Holding Register - Analog Outputs (R/W)
6	ERRO	Local	INT	%IW1			Input Register - Analog Outputs (R)
7	KP	Local	INT	%QW99			Holding Register - Analog Outputs (R/W)

Figura 2 - Mapeamento das variáveis utilizando o OpenPLC.

### 3. ScadaBR

O ScadaBR é uma aplicação web, sendo necessário abrir o navegador no computador e navegar para o endereço <http://192.168.43.91:8080/ScadaBR>. A Figura 3 apresenta a página do supervisório desenvolvido no ScadaBR.

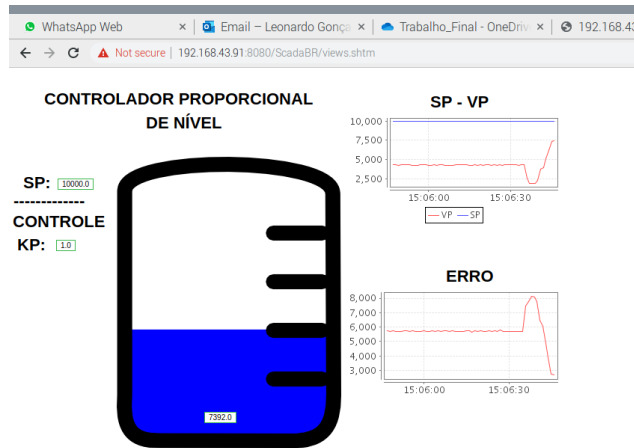


Figura 3 - Interface utilizada no ScadaBR.

### 4. Testes Práticos

Antes de iniciar os testes na estrutura montada, foram feitos testes de conexão utilizando o comando ping (Figura 4). O intuito era identificar os momentos de perdas de conexão para tomar as devidas atitudes para que a planta voltasse ao seu devido funcionamento.

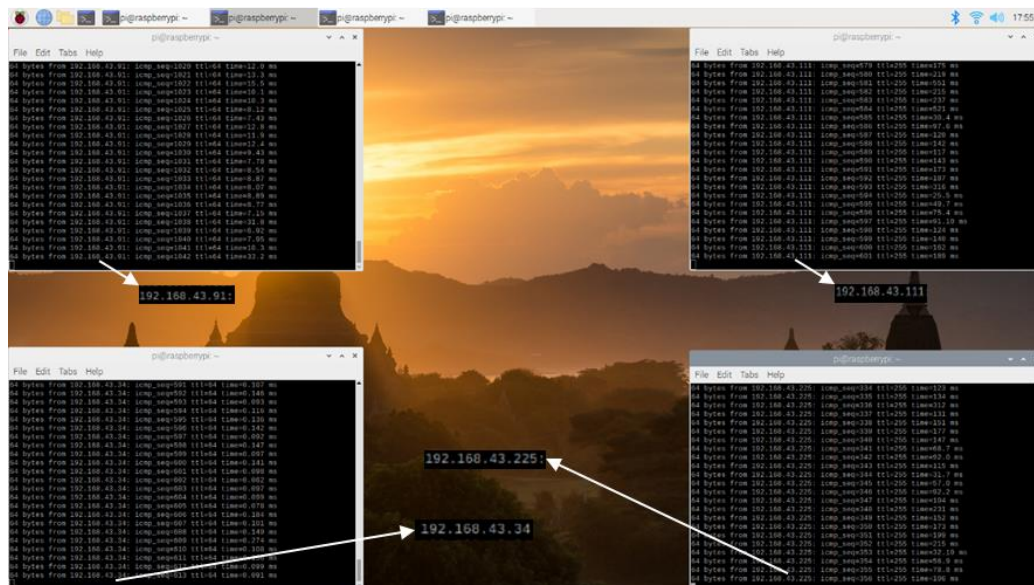


Figura 4 - Testes de conexão entre os dispositivos.

Depois de garantir a devida conexão entre os dispositivos na rede interna, os testes práticos puderam ser realizados. O sistema consistiu em interpretar um controle de nível de um tanque. O elemento que representou um sensor foi um potenciômetro de 10K, onde a variação do potenciômetro evidenciava a variação no nível do tanque. Com isso, o

elemento atuador foi simbolizado por um LED, sendo que, através de um controlador proporcional a intensidade do LED aumentava à medida que o nível se desviava do *Set point*, representando por exemplo uma válvula proporcional aumentando a sua abertura permitindo a passagem de mais líquido para o tanque, pois o nível está abaixo do requerido pelo operador de processo. Os componentes práticos utilizados neste trabalho podem ser verificados na Figura 5.

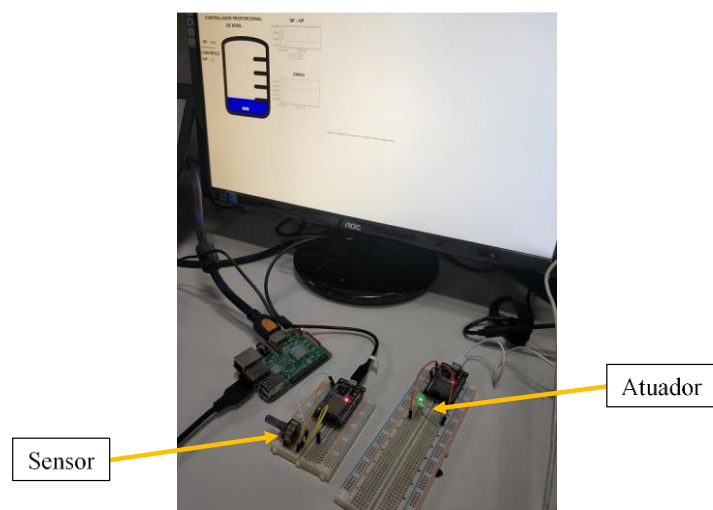


Figura 5 - Componentes práticos utilizados neste trabalho.

## 5. Considerações Finais

O presente documento relata o desenvolvimento de uma aplicação de controle e supervisão, utilizando dois microcontroladores ESP32, cada um com responsabilidades diferentes: um atua em conjunto com o sensor (potenciômetro) e outro realiza interação com o atuador. O emprego desses dispositivos, facilitou a implantação da rede Modbus para interfaceamento entre camadas 1 e 2, uma vez que esses representam o sistema microprocessado do sensor e atuador.

Apesar dos objetivos de o trabalho terem sido alcançados, há pontos que podem ser questionados em uma situação real. Os dispositivos conectados em rede, são interfaceados via Wi-Fi, que se caracteriza por não ser um tipo de rede determinística. Considerando a criticidade de um sistema industrial, o determinismo, deve ser um dos pontos requeridos em uma rede operante, pois sua falta causa atrasos de respostas e pode comprometer o sistema. Uma solução possível para este feito é utilizar uma arquitetura de rede Ethernet com cabeamento CAT5 ou superior.

Repositório Git: [https://github.com/Leonardo-Baptista/Trabalh\\_Final\\_de\\_Redes](https://github.com/Leonardo-Baptista/Trabalh_Final_de_Redes)

## 6. Referências Bibliográficas

- [1] T. Alves, “Modbus Address Mapping.” [Online]. Available: <https://www.openplcproject.com/scada>.
- [2] T. Alves, “OPENPLC ON RASPBERRY PI.” [Online]. Available: <https://www.openplcproject.com/getting-started-rpi>.
- [3] T. Alves, “OPENPLC AND ESP8266.” [Online]. Available: <https://www.openplcproject.com/getting-started-esp8266>.
- [4] T. Alves, “OPENPLC EDITOR.” [Online]. Available: <https://www.openplcproject.com/plcopen-editor>.
- [5] T. Alves, “INSTALLING SCADABR USING A VIRTUAL MACHINE.” [Online]. Available: <https://www.openplcproject.com/reference-installing-scadabr>.