



Universidade Federal da Paraíba

Centro de Energias Alternativas e Renováveis

Departamento de Engenharia Elétrica

ANDRÉA WILLA RODRIGUES VILLARIM

**DESENVOLVIMENTO DE SISTEMA SUPERVISÓRIO PARA
DISPOSITIVOS COMPUTACIONAIS PORTÁTEIS**

João Pessoa, Paraíba
Junho de 2016

ANDRÉA WILLA RODRIGUES VILLARIM

DESENVOLVIMENTO DE SISTEMA SUPERVISÓRIO PARA
DISPOSITIVOS COMPUTACIONAIS PORTÁTEIS

*Trabalho de Conclusão de Curso submetido ao
Departamento de Engenharia Elétrica da
Universidade Federal da Paraíba como parte
dos requisitos necessários para a obtenção do
título de Engenheiro Eletricista.*

Área de Concentração: Eletrônica

Orientador:

Professor Carlos Alberto de Souza Filho, M. Sc.

João Pessoa, Paraíba
Junho de 2016

ANDRÉA WILLA RODRIGUES VILLARIM

DESENVOLVIMENTO DE SISTEMA SUPERVISÓRIO PARA
DISPOSITIVOS COMPUTACIONAIS PORTÁTEIS

*Trabalho de Conclusão de Curso submetido ao
Departamento de Engenharia Elétrica da Universidade
Federal da Paraíba como parte dos requisitos
necessários para a obtenção do título de Engenheiro
Eletricista.*

Área de Concentração: Eletrônica

Aprovado em ____ / ____ / _____

Professor Avaliador
Universidade Federal da Paraíba
Avaliador

Professor Carlos Alberto de Souza Filho, M. Sc.
Universidade Federal da Paraíba
Orientador, UFPB

Dedico este trabalho à minha família, aos que fazem parte do Departamento de Engenharia Elétrica da UFPB e a todos que lutaram ao meu lado durante a minha formação.

AGRADECIMENTOS

Agradeço primeiramente a Deus, meu tudo, que me ajuda a trilhar cada caminho, me concedendo alegrias e dificuldades que me fazem crescer e ser uma pessoa melhor a cada dia.

Agradeço a Maria Santíssima, minha Mãezinha e Senhora, de quem sou serva fiel e que está sempre intercedendo por mim e cuidando de cada passo meu.

Agradeço a minha família, pois é meu céu na terra. Agradeço todos os esforços dos meus pais que desde que nasci fazem tudo por mim. Nunca mediram esforços para que eu chegasse onde estou hoje e sei que são meus maiores torcedores. Agradeço a minha irmã Mariana, que é a maior companheira que tenho na vida e está comigo para tudo. Aos meus irmãos, André e Juliana, que mesmo à distância, estão sempre rezando pelo meu sucesso. A todos os familiares que contribuíram para que eu conseguisse realizar o meu sonho.

Agradeço aos meus amigos de curso, meu PPG amado. Eles que estiveram comigo durante as diversas noites de estudos e com os quais eu construí verdadeiros laços.

Aos amigos que sempre estiveram comigo dividindo as alegrias e ajudando a superar as tristezas.

A minha Paróquia Nossa Senhora de Fátima, que me proporcionou presentes que levo pra vida e que me acompanham em tudo que faço. Lá conheci inúmeros amigos com os quais divido os meus dias. Nada mais justo do que homenageá-la com o meu curso, nomeando a turma como Turma Nossa Senhora de Fátima.

Aos meus professores que me ajudaram a ser a profissional que estou me tornando, em especial ao meu orientador Carlos Alberto, que desde o início do curso mostrou seu interesse e empenho por nós alunos e me orientou por anos, partilhando comigo seus conhecimentos e seu tempo.

“Onde houver trevas que eu leve a luz.”

São Francisco de Assis

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema supervisório para dispositivos computacionais portáteis como *tablets* e *smartphones*. O trabalho de funcionários que fazem rondas periódicas em uma empresa para verificar o estado das variáveis do processo pode ser facilitado com a utilização desse modelo de supervisório projetado. O sistema é capaz de coletar os valores das variáveis de uma linha produtiva industrial convencional, interpretar as informações adquiridas e apresentá-las ao usuário de forma simples e dinâmica. O usuário deve ser capaz de interagir com o processo, modificando valores de componentes do mesmo. O sistema supervisório é apresentado como um aplicativo para dispositivos móveis que interage com o restante do sistema por meio do protocolo de comunicação Bluetooth. Os testes e resultados são apresentados para validar o projeto desenvolvido.

Palavras-chave: Sistema Supervisório, Automação Industrial, Dispositivos Móveis.

ABSTRACT

This paper presents the development of a supervisory system for portable computing devices such as tablets and smartphones. The work of employees who make regular rounds in a company to check the status of process variables can be facilitated with the use of this designed supervisory model. The system is able to collect the values of variables in conventional industrial production line, interpret the information acquired and present them to the user in a simple and dynamic way. The user must be able to interact with the process and modify component values thereof. The supervisory system is presented as an application for mobile devices that interact with the rest of the system through the Bluetooth communication protocol. The tests and results are presented to validate the designed project.

Keywords: Supervisory System, Industrial Automation, Mobile Devices.

LISTA DE ILUSTRAÇÕES

Figura 1: Pirâmide de Automação.....	15
Figura 2: Diagrama de blocos das possíveis implementações do projeto.....	25
Figura 3: Comunicação entre os sensores e atuadores do CLP através da Interface de Entrada e Saída....	26
Figura 4: IES para sinal analógico.....	27
Figura 5: IES para sinal digital.....	27
Figura 6: Diagrama de estados do microcontrolador.....	29
Figura 7: Diagrama de caso de uso.....	30
Figura 8: Telas do dispositivo.....	32
Figura 9: Aplicativo e sistema desenvolvido.....	33
Figura 10: Diagrama de estados da UVGC.....	34
Figura 11: Painel utilizado para teste.....	35
Figura 12: Plataforma de testes para UAC1 com variável digital.....	36
Figura 13: Tela do aplicativo com Sensor 1 em OFF.....	36
Figura 14: Tela do aplicativo com Sensor 1 em ON.....	37
Figura 15: Comando para escrita em uma bobina e respectiva resposta do CLP.....	38
Figura 16: Resposta para escrita em uma bobina.....	38
Figura 17: Plataforma experimental da UAC2.....	39
Figura 18: Comando para leitura de uma bobina e respectiva resposta do CLP apresentando erro.....	40
Figura 19: Resposta para ler de uma bobina reportando erro.....	41
Figura 20: Comando para leitura de uma bobina e respectiva resposta do CLP para bobina desligada.....	41
Figura 21: Resposta para leitura de uma bobina desligada.....	42
Figura 22: Comando para leitura de uma bobina e respectiva resposta do CLP para bobina ligada.....	42
Figura 23: Resposta para leitura de uma bobina ligada.....	43
Figura 24: Painel com o CLP e o potenciômetro.....	44
Figura 25: Comando para escrita em um registrador e respectiva resposta do CLP.....	44
Figura 26: Resposta para escrita em um registrador.....	45
Figura 27: Acompanhamento dos registradores para comandos de leitura e escrita.....	46
Figura 28: Comando para leitura de um registrador e respectiva resposta do CLP.....	46
Figura 29: Resposta para leitura de um registrador.....	47
Figura 30: Tela do aplicativo para variáveis analógicas.....	47
Figura 31: Tela do aplicativo para variáveis analógicas com variação no zoom.....	48

LISTA DE SÍMBOLOS

API - *Application Programming Interface*

ba – Botão Atuador

CLP – Controlador Lógico Programável

CRC – *Cyclic Redundancy Check*

DC – *Driver* de Comunicação

IDE – *Integrated Development Environment*

IES – Interface de Entrada e Saída

IHM – Interface Homem-Máquina

IHM – Laboratório de Acionamento e Controle

LED – *Light Emitting Diode*

PC – *Personal Computer*

PCI – Placa de Circuito Impresso

RS – *Recommended Standart*

RTU – *Remote Terminal Unit*

Sb-OFF – *Switch Button* - OFF

Sb-ON – *Switch Button* – ON

SCADA – *Supervisory Control and Data Acquisition*

SIG – *Special Interest Group*

ta – Tela atualizada

tL – Terminada leitura

UAC – Unidade de Aquisição e Controle

UART – Universal Asynchronous Receiver/Transmitter

UVGC – Unidade de Visualização, Gerenciamento e Controle

va – Valor armazenado

vd – Valor decodificado

ve – Valor enviado

vr – Valor recebido

SUMÁRIO

Agradecimentos.....	v
Resumo.....	vii
Abstract.....	viii
Lista de Ilustrações.....	ix
Lista de Símbolos.....	x
Sumário.....	xi
1 Introdução.....	12
1.1 Objetivo.....	14
1.1.1 Objetivo Geral.....	14
1.1.2 Objetivos Específicos.....	14
2 Embasamento Teórico.....	15
2.1 Sistemas Supervisórios.....	15
2.2 Sistemas Supervisórios Para Dispositivos Móveis.....	18
2.3 Protocolo de Comunicação Modbus.....	19
2.4 Protocolo de Comunicação Bluetooth.....	21
2.5 Plataforma de Desenvolvimento Android.....	23
3 Descrição do Sistema.....	24
3.1 Hardware.....	25
3.2 Software.....	28
3.3 Descrição da Interface.....	31
3.4 Análise dos Resultados.....	35
4 Considerações Finais.....	49
Bibliografia.....	50
APÊNDICE A.....	52

1 INTRODUÇÃO

A grande preocupação das empresas é aumentar a produtividade, se tornar mais eficaz e mais lucrativa. Nos dias atuais, se um processo não for realizado de forma eficiente para assim obter uma boa produtividade, a empresa não obterá o sucesso esperado e dificilmente sobreviverá no mercado (MACEDO, 2012). Devido a isso, a busca por soluções sofisticadas e de ponta é o passo inicial para conquistar o mercado (MAITELLI, 2003).

Em um processo produtivo industrial, os instrumentistas ou supervisores realizam rondas periódicas para coletar valores de variáveis da planta, como corrente, tensão, temperatura entre outros. Eles coletam visualmente essas informações utilizando instrumentos específicos para cada medição.

É muito importante um sistema que permita ao técnico ou supervisor visualizar de forma rápida as informações desejadas de uma planta, pois uma falha não identificada inicialmente pode gerar uma parada na linha de produção e assim acarretar uma perda financeira considerável. Logo, um sistema de monitoramento eficiente permitiria a detecção do problema e sua correção de forma mais efetiva.

Os sistemas supervisórios podem supervisionar ou monitorar processos de uma planta industrial, por meio do acompanhamento de variáveis da planta que está sendo automatizada (JURIZATO; PEREIRA, 2003). O trabalho do projetista que desenvolverá o sistema consiste na criação de telas gráficas, etapa que varia de acordo com o processo a ser controlado, além da definição dos comandos a serem executados (MAIA, 1998).

Uma pesquisa mundial constatou que a população está com um crescente acesso aos aplicativos móveis. O uso de dessas aplicações aumentou 58% em 2015 em relação ao ano anterior (MOBILE TIME, 2016). Diversas empresas estão buscando incorporar aplicações móveis a seu dia-a-dia para acelerar seus negócios. As empresas visam sempre aumentar o capital e os dispositivos portáteis se tornam indispensáveis em um mundo onde a palavra “mobilidade” está a cada dia mais conhecida (LECHETA, 2013).

A utilização de supervisório em dispositivos portáteis permite a visualização do estado da planta enquanto o operador realiza uma ronda ou manutenção no sistema. Logo, possibilita a visualização e modificação dos pontos de operação mesmo em

deslocamento ou à distância. Isto gera uma facilidade para identificação de falhas, monitoração de estados e armazenamento de variáveis.

Hoje já é possível a implementação de supervisórios nas mais diversas aplicações, tanto nos variados segmentos do comércio e da indústria como nas áreas residenciais. São encontrados em sistemas de segurança de residências, nos hospitais, na climatização de escritórios e laboratórios, nas pequenas fábricas, além dos muitos processos industriais.

Uma das áreas em que cresce a aplicação dos supervisórios é a biomédica. A utilização de sistemas supervisórios ajuda na prevenção de diversas doenças, como exemplo da úlcera por pressão. Este é um sério problema que atinge principalmente indivíduos acamados e é dado geralmente pela pressão de contato corpo com a superfície do leito por um longo tempo. Para auxiliar a prevenção, é necessário reduzir o contato nas áreas do corpo que apresentam maiores chances de desenvolver a doença, como cotovelos e ombros. Essa ação pode ser realizada por um profissional de forma manual, mas requer tempo e conhecimento sobre assunto. Então o processo consiste em monitorar e controlar a pressão nas áreas de maior preocupação utilizando o supervisório (NOVELLETO, ET AL., 2014).

Os sistemas supervisórios estão sendo utilizados nas diversas áreas da indústria para visualizar os dados do processo, armazená-los e controlar a planta de acordo com uma estratégia determinada. A necessidade de gerenciar a elevação artificial do petróleo, por exemplo, torna essencial a obtenção de informações sobre o processo. Cada método de elevação tem suas próprias variáveis a serem monitoradas, então diversas empresas estão desenvolvendo softwares específicos para cada caso (MEDEIROS, 2006).

É possível também comandar um circuito a partir de um dispositivo móvel. O usuário envia comandos a partir do dispositivo e assim realiza o monitoramento e controle do sistema. Esse tipo de aplicação pode ser utilizado em situações mais simples que a anterior, como o controle de um motor de indução a partir de uma comunicação sem fio entre o supervisório e o servidor (LIN; CHANG-CHIEN, 2006).

Diante do exposto, o projeto tem como objetivo desenvolver um sistema que realize a supervisão de um processo produtivo industrial. O sistema possui um hardware voltado para a aquisição e controle das variáveis da planta e um software capaz de interpretar as informações recebidas e apresentá-las ao usuário.

1.1 OBJETIVO

1.1.1 OBJETIVO GERAL

O objetivo principal do projeto é desenvolver um aplicativo que funcione como supervisor de um processo industrial. O aplicativo deve monitorar e controlar as variáveis digitais e analógicas de uma planta e interagir com o usuário de forma simples e dinâmica. O sistema deve acompanhar as mudanças ocorridas nas variáveis e apresentá-las ao usuário quando forem detectadas. A comunicação entre o sistema e o dispositivo móvel é realizada por protocolo Bluetooth.

1.1.2 OBJETIVOS ESPECÍFICOS

- Desenvolvimento de um circuito para adquirir os valores das variáveis digitais e analógicas do processo.
- Desenvolvimento de um aplicativo para sistema operacional Android capaz de apresentar ao usuário os valores finais das variáveis recebidas de forma dinâmica.
- Desenvolvimento de um sistema de comunicação via Bluetooth em hardware e software, que permita a troca de informações entre o dispositivo portátil e o sistema desenvolvido.
- Desenvolvimento de uma placa de circuito impresso que possua o circuito desenvolvido para comunicação serial e por Bluetooth.

2 EMBASAMENTO TEÓRICO

O sistema coleta o estado de variáveis de um processo produtivo e apresenta ao usuário por meio de um aplicativo para dispositivos computacionais portáteis. Esse aplicativo atua como supervisor do processo industrial. A comunicação entre a planta industrial e o sistema pode ser realizada utilizando um microcontrolador ou por meio do protocolo Modbus. O celular realiza a comunicação via Bluetooth para restringir a distância de possível acesso às informações e assim proteger o sistema contra invasões. O aplicativo é desenvolvido para sistema operacional Android.

Podemos observar na Figura 1 a pirâmide de automação e a partir dela podemos relacionar os dispositivos de campo com a supervisão através de um microcontrolador ou CLP, que no caso deste último se comunica através do protocolo de comunicação Modbus com o sistema que realiza a supervisão.



Figura 1: Pirâmide de Automação.

2.1 SISTEMAS SUPERVISÓRIOS

Os sistemas SCADA (*Supervisory Control and Data Acquisition*) são aplicativos que permitem que as variáveis de um processo produtivo ou instalação física sejam coletadas, interpretadas, armazenadas e apresentadas ao usuário. Essas informações

podem ser monitoradas por meio de telas gráficas com atualizações instantâneas das variáveis do processo.

Os dados são recebidos de um CLP ou controlador, por exemplo, e os softwares supervisórios auxiliam no processo de controle da qualidade e de movimentação de dados para gerenciamento e diretrizes. Desta forma, a escolha do software de supervisão é importante na estratégia de automação de uma empresa (BOARETTO, 2008).

Os sistemas SCADA são utilizados com frequência na automação industrial, pois servem para aumentar a praticidade de controle do processo de produção. Embora a utilização de salas de controle com painéis centralizados facilite a obtenção das informações, já que vão estar concentradas no mesmo ambiente, muitas vezes esses locais apresentam grandes extensões com diversos instrumentos, o que dificulta o trabalho do operador. O supervisório veio para diminuir a dimensão dos painéis e aperfeiçoar a interface homem/máquina (CARNEIRO, 2005).

Um sistema SCADA também permite ao operador monitorar e controlar um processo distribuído em lugares distantes da sala de controle, logo a redução dos custos de visitas rotineiras podem ser verificados, já que se torna desnecessária a presença do operador para a ronda periódica (BOYER, 1993).

Existem diversos ambientes para desenvolvimento de sistemas supervisórios disponíveis comercialmente, dentre os quais pode-se citar:

- *Wonderware InTouch*: Possui uma interface de fácil comunicação com o usuário e uma linguagem de programação versátil. Permite a criação e programação de objetos e instalações que se assemelham muito a forma real e possui uma biblioteca com representação de diversos equipamentos industriais.

O *InTouch* está presente em mais de um terço das instalações industriais do mundo e permite aos operadores monitorar e controlar os processos em tempo real. Permite a representação em gráficos mais complexos e possui uma biblioteca de símbolos pronta para utilização (SCHNEIDER ELECTRIC, 2016).

O *InTouch* pode garantir a segurança do processo impedindo que operadores executem atividades não autorizadas, pois permite definir níveis de acesso ao sistema. Além disso, permite que o sistema não esteja restrito a apenas uma empresa, mas que tenha uma base instalada com diversas empresas capazes de integrá-lo (SA AUTOMAÇÃO, 2016).

- *SIMATIC WinCC*: Possibilita ao administrador designar e controlar acessos de usuários. O administrador pode configurar a qualquer momento – incluindo

em tempo de execução – até 128 grupos de usuários, cada um contendo até 128 usuários individuais e designar a eles direitos exclusivos de acesso às funções do WinCC (SIEMENS, 2016).

É possível também criar interfaces com configuração personalizada para qualquer aplicação, armazenar mensagens e disponibilizar as informações conforme desejado. É possível utilizar elementos gráficos prontos que só necessitam ser integrados ao sistema. Além disso, permite proteger conhecimento industrial e tecnológico em elementos encapsulados e usá-los na frequência necessária com diferentes parâmetros (SIEMENS, 2016).

- *FactoryTalk View*: Sistema supervisório que apresenta uma arquitetura flexível que inclui um software de configuração para desenvolvimento e testes de aplicações IHM e um servidor que armazena componentes de projeto como telas gráficas e *TAGs* e mostra essas variáveis aos clientes. Possui uma fácil manutenção do sistema com reconhecimento automático de novo *firmware* do controlador, acesso e controle seguros de uma IHM a partir de um servidor *web* e possibilidade de teste de IHM que permite aos desenvolvedores validar telas que usem parâmetros diversos (ROCKWELL AUTOMATION, 2016).

- *ScadaBR*: O *ScadaBR* é um supervisório disponibilizado em sistema *Open Source* (*software* livre). É bastante utilizado nas áreas de automação de processos industriais, redes de distribuição (água e energia), automação predial e residencial e diversas aplicações de sensoriamento. Possui suporte para mais de 20 protocolos de comunicação, sendo compatível com *hardwares* de centenas de fabricantes em todo o mundo. Além disso, é uma camada de integração com sistemas de *software* de terceiros (externos), através de servidores da *web* (SCADABR, 2016).

Cada um dos sistemas citados possui maiores vantagens para diferentes aplicações. Assim, é importante a escolha de qual utilizar, levando em consideração as necessidades de cada situação e avaliando critérios como a facilidade de aquisição, segurança, dentre outros. Além disso, alguns *softwares* apresentam a possibilidade de desenvolvimento de um supervisório para dispositivos portáteis, como será abordado a seguir.

2.2 SISTEMAS SUPERVISÓRIOS PARA DISPOSITIVOS MÓVEIS

A indústria, visando aumentar a produtividade, insere rapidamente a utilização de dispositivos portáteis na automação de seus processos, incorporando-os aos trabalhos de técnicos e operadores (MEIRELLES, 2004). Isso torna o desenvolvimento de aplicações portáteis para a área industrial um ramo cada vez mais crescente.

O desenvolvimento da computação móvel tem proporcionado o surgimento de novas aplicações e tecnologia. O preço acessível de aparelhos como celulares e *tablets* possibilitou para as empresas a utilização desses dispositivos como plataformas computacionais para instalação de sistemas supervisórios (PEROZZO; PEREIRA, 2006).

A automação industrial está sendo beneficiada com o desenvolvimento da computação móvel. Atualmente existem diversas pesquisas e aplicações que aliam o uso desses dispositivos à realidade industrial, principalmente como sistemas supervisórios (MACHADO, 2012).

Existem diversas plataformas comerciais disponíveis para dispositivos móveis, dentre as quais pode-se citar:

- **Elipse:** É totalmente configurável pelo usuário e permite a monitoração de variáveis em tempo real. Além disso, é possível realizar acionamentos e enviar ou receber informações de equipamentos de aquisição de dados.

O Elipse SCADA possui uma maneira para o usuário acessar todos os elementos do sistema e suas propriedades navegando em uma árvore hierárquica que fornece uma visão geral do aplicativo, organizando a configuração e documentação do mesmo. Além disso, permite a edição *online*, onde é possível alterar o aplicativo sem parar a execução, possui diferentes versões para atender demandas específicas e *software* gratuito disponível (ELIPSE, 2016).

- **LabVIEW:** A abordagem exclusiva da *National Instruments* de desenvolvimento baseado em plataforma para aplicações em engenharia e ciências criou um ambiente de criação interativo e de fácil utilização. A plataforma inclui bibliotecas para a área da engenharia com funções de *software* e interfaces de *hardware*, além do compartilhamento de recursos. Além disso, possibilita a execução em multiplataformas e dispositivos embarcados (NATIONAL INSTRUMENTS, 2016).

O módulo LabVIEW DSC (*Datalogging and Supervisory Control*) une os benefícios da programação gráfica às aplicações SCADA. O DSC possui visualizador de dados históricos, fornece ferramentas para gráficos de tendência histórica ou em tempo-real. Além disso, inclui um bando de dados próprio e permite a integração com estruturas já existentes, possibilitando a utilização de outros bancos de dados (NATIONAL INSTRUMENTS, 2016).

Para aliar a tecnologia de supervisórios aos dispositivos móveis, a *National Instruments* desenvolveu modos de monitorar remotamente as medições. O *Data Dashboard* permite o desenvolvimento de aplicações móveis e personalizadas do *software* LabVIEW. Possibilita o compartilhamento de valores via rede ou serviços da *web* e os exibe em gráficos, indicadores de texto, LEDs, dentre outras formas gráficas de visualização das variáveis (NATIONAL INSTRUMENTS, 2016).

- Proficy SCADA: O sistema supervisório da GE (*General Electric*) está presente em diversas indústrias e permite estender a sala de controle utilizando dispositivos móveis e possibilitando limitar o acesso aos dados coletados apenas a pessoas autorizadas previamente. Uma vantagem desse *software* é o “gravador” que possibilita ao operador analisar graficamente eventos ocorridos antes do acesso (GE AUTOMATION, 2016).

- Pocket Genesis: Possibilita a utilização de dispositivos móveis para aplicações que requerem a visualização de variáveis, tendências, alarmes e relatórios utilizando uma rede sem fio. A segurança do Windows protege o acesso a informações importantes e alterações de parâmetros de entrada (POCKET GENESIS, 2016).

Possui um mecanismo que armazena dados e os envia logo após detectar uma conexão de rede, sem verificar a degradação do sinal, e uma possível adaptação de visualização em tempo de execução do sistema supervisório. Visualiza dados de tendência pelo tempo que se desejar, seja ao longo de segundos, dias ou até mesmo anos. Possui a função “*replay*” para dados a partir de bancos padrão, como Microsoft SQL Server, usando tipos de gráficos populares. Além disso, permite a comparação simultânea de dados históricos e dados em tempo real (POCKET GENESIS, 2016).

2.3 PROTOCOLO DE COMUNICAÇÃO MODBUS

Outro aspecto, na comunicação serial entre equipamentos, é o protocolo que irá efetuar a conexão e troca de mensagem entre eles. Protocolo é um conjunto de regras e convenções para conversação (TANENBAUM, 1994). Esse conjunto de regras define como ocorrerá a comunicação entre dois equipamentos e a ordem que eles devem seguir ao formular as mensagens, de forma que sejam compreendidas por ambos os lados que participam da comunicação (CUNHA, 2000).

O protocolo de comunicação Modbus foi desenvolvido em 1979 pela Modicon e é atualmente um dos mais utilizados na indústria para comunicação entre os equipamentos. A comunicação é do tipo mestre/escravo e pode ser estabelecida via RS-232, RS-485 ou Ethernet (SACRAMENTO, 2015).

O padrão RS-232 é utilizado em comunicação entre dois equipamentos, no caso do Modbus, entre um mestre e um único escravo. A velocidade máxima é aproximadamente 115Kbps e a distância máxima entre os dispositivos é próxima aos 30m. O RS-485 é um dos mais utilizados pelo protocolo Modbus e na indústria em geral. Em alguns casos é possível operar com taxas de até 50Mbps e aceita conexão entre dispositivos com até 1200m de distância. Além disso, permite até 32 dispositivos conectados ao barramento (NATIONAL INSTRUMENTS, 2013).

O padrão Ethernet, quando referido ao protocolo Modbus, apresenta variações de acordo com as instalações da empresa, como a taxa de transferência máxima que pode chegar a 10Mbps ou a 10Gbps. Em relação às distâncias, podem chegar a 100m ou até valores próximos a 200m. É possível utilizar cabos de fibra óptica, o que permite ao sistema possuir maiores valores de distância e melhores taxa de comunicação

O protocolo Modbus apresenta dois possíveis modos de transmissão, ASCII e RTU, que definem com que forma os comandos e respostas serão transmitidos pela rede. No projeto, foi utilizado o modo RTU que apresenta como padrão de mensagens o exposto na Tabela 1.

Tabela 1: Padrão Modbus para troca de mensagens.

Endereço	Função	Dados	CRC-16
8 bits	8 bits	Nx8bits	16 bits

Cada mensagem possui um byte, dividido em dois caracteres hexadecimais de quatro bits cada. Inicialmente, é enviado o endereço do escravo com o qual se deseja estabelecer a comunicação. Em seguida, a função a ser realizada, que tem seus valores

definidos pelo protocolo. Os dados dependem de cada função e seu tamanho é variável, porém definido de acordo com a função escolhida. Por fim, é enviado um CRC-16 para detecção de erro. Nesse caso, cada byte é enviado com um bit de início e dois de parada ou, ao invés de dois de parada, apenas um após um bit de paridade.

O protocolo Modbus trata os equipamentos como escravos e a sua atividade na rede consiste em responder às questões emitidas pelo mestre, que pode ser um PC ou outro equipamento. O mestre pode se comunicar com um escravo em particular e esperar pela sua resposta, ou, pode enviar um só comando a todos os escravos (MORAES, 2000).

O protocolo Modbus define uma estrutura de mensagem que os controladores reconhecerão e utilizarão independente do tipo de rede acima deles. Também descreve o processo que um controlador usa para requisitar o acesso aos demais dispositivos, como responderá a pedidos dos mesmos e como serão descobertos erros na comunicação. Ou seja, o protocolo fornece um formato comum para os comandos e respostas, assim como para os dados de mensagem (MORAES, 2000).

Outra definição para o Modbus é que ele apresenta o padrão interno que os controladores usam para analisar gramaticalmente as mensagens. Ele permite que um controlador reconheça uma mensagem dirigida a ele, determine qual função a ser executada e interprete os dados contidos na mensagem. Se uma resposta for requisitada, o controlador irá desenvolver a mensagem de resposta e a enviar seguindo o mesmo protocolo (SCHNEIDER ELETRONICS, 2016).

2.4 PROTOCOLO DE COMUNICAÇÃO BLUETOOTH

O Bluetooth surgiu como uma tecnologia para comunicação de dispositivos eletrônicos que não necessitava cabeamento para realizar a conexão dos mesmos (BILLO, 2003). O padrão Bluetooth foi desenvolvido pelo grupo SIG (*Special Interest Group*), um grupo composto por empresas líderes em computação e telecomunicações. É um padrão de comunicação usado em escala mundial, que permite a comunicação sem fio entre os dispositivos e apresenta baixo consumo de energia. A transmissão é feita por radiofrequência, ou seja, um dispositivo pode detectar outro dentro do limite de proximidade estabelecido, independente das posições em que se encontrem.

Quando dois ou mais dispositivos Bluetooth desejam se comunicar, eles precisam parear, ou seja, se identificar. A comunicação estabelecida forma uma rede denominada *piconet*. A rede pode possuir de dois a oito dispositivos conectados. Para que os dispositivos saibam quais fazem parte da sua *piconet* é necessário que o aparelho que deseja se conectar a uma determinada *piconet* envie um sinal chamado *Inquiry*. Os dispositivos que recebem o sinal respondem informando sua identificação e os dados de sincronização. Assim, um aparelho pode emitir um sinal denominado *Page* para se conectar a outro. Nesta comunicação, o dispositivo que iniciou a conexão se torna o mestre, enquanto os demais são denominados escravos (BLUETOOTH SIG, INC., 2016). O mestre deve regular a transmissão de dados e o sincronismo entre os dispositivos.

O IEEE (*Institute of Electrical and Eletronics Engineers, Inc.*) atua como órgão regulatório de trabalho para desenvolver padrões de comunicação em redes com o objetivo de possibilitar a compatibilidade entre dispositivos de diferentes fabricantes. O grupo IEEE 802 é responsável por toda a área LAN e é subdividido em outros grupos, onde se encontra o padrão Bluetooth IEEE 802.15.1.

Hoje, o Bluetooth IEEE 802.15.1 já possui diversas atualizações e é definido pelo IEEE 802.15 como um padrão. Em 2005, a união entre a UWB (*Ultra Wideband*) e o Bluetooth SIG impulsionou o padrão Bluetooth 802.15.1 para sistemas de maior velocidade. Quando um dispositivo desse padrão é ligado, ele tenta operar como um escravo de um dispositivo mestre que já esteja ligado (RINCO, 2005).

O Bluetooth opera na faixa ISM (*Industrial, Scientific, Medical*) e funciona a 2,4GHz, uma faixa de frequência aceita em grande parte do planeta. Como a faixa ISM pode ser utilizada por muitos sistemas de comunicação, o Bluetooth divide sua frequência em vários canais para evitar interferências. O dispositivo que estabelece a comunicação muda de canal rapidamente. Esse método é chamado “salto de frequência” e possibilita uma largura da banda de frequência pequena, diminuindo as interferências (SIQUEIRA, 2006).

Os dispositivos podem ser diferenciados em três classes, que determinam o alcance para comunicação Bluetooth. Os de classe 1 são dispositivos com alcance de até 100m, os de classe 2 podem trocar mensagens até 10m de distância e os de classe 3 só mantêm a comunicação à 1m.

Os protocolos do padrão IEEE 802.15.1 são necessários para implementar os *profiles* e seus modos de utilização. Cada *profile* determina mensagens e procedimentos definidos pelo grupo Bluetooth SIG como funcionalidades. (RINCO, 2005).

O Bluetooth apresenta diversas vantagens de utilização, dentre elas a facilidade ao conectar dispositivos, já que a grande maioria dos celulares, *tablets* e muitos computadores possuem essa tecnologia, apresenta um baixo consumo de energia, e, para finalizar, a aquisição de módulos Bluetooth pode ser realizada por um baixo custo.

2.5 PLATAFORMA DE DESENVOLVIMENTO ANDROID

Ao se tratar de dispositivos móveis, sempre existiram muitas limitações tanto do *software* quanto do *hardware*. Nesse contexto, é desenvolvida a plataforma Android para mudar esta realidade e revolucionar o mercado de dispositivos móveis (GOOGLE, 2008).

O Android é uma plataforma lançada no final de 2007. Pode ser definido como uma pilha de *softwares* para aparelhos portáteis, que inclui sistema operacional, *middleware* e diversos aplicativos. Para desenvolvimento de aplicações para Android, é utilizado o Android SDK (GOOGLE, 2008).

O Android SDK é uma plataforma com *software* livre, que aceita o uso de API gráfica e programação em Java. É possível instalar o *Android Development Tools* (ADT) como *plugin* de acordo com a IDE utilizada. O Android SDK permite a criação de interfaces rápidas e organizadas pela linguagem *Extensible Markup Language* (XML) (FRANCISCATO, 2008).

O sistema operacional Android está presente em mais de um bilhão de *smartphones*. Cada versão recebe o nome de sobremesas e surge com alguma inovação. A mais nova versão é denominada Android 6.0 *Marshmallow* e traz inovações em algumas áreas, como a de segurança, onde o usuário pode compartilhar o dispositivo por meio do modo de usuário visitante (ANDROID, 2016).

3 DESCRIÇÃO DO SISTEMA

O sistema desenvolvido deve atuar como um supervisor de uma planta industrial e coletar dados do processo produtivo, interpretá-los e apresentar ao usuário. O sistema deve ser capaz de interpretar variáveis digitais, como resposta de sensores de presença em uma esteira, ou variáveis analógicas, como a temperatura sob a qual está atuando determinada parte do processo. A comunicação pode ser estabelecida utilizando-se um microcontrolador ou através do protocolo de comunicação Modbus, que só necessita de um *driver* de comunicação para se conectar com o CLP. Esse protocolo é utilizado, pois está presente no CLP disponível para desenvolvimento do projeto. Os dados finais são apresentados por meio de um aplicativo Android de fácil interação com o usuário que se comunica com o sistema através do protocolo Bluetooth.

Diante de todas as funcionalidades expostas, nesta seção será apresentada a descrição necessária para o desenvolvimento do sistema que está dividido em *hardware* e *software*.

3.1 HARDWARE

O sistema proposto pode ser representado pelo diagrama de blocos ilustrado na Figura 2.

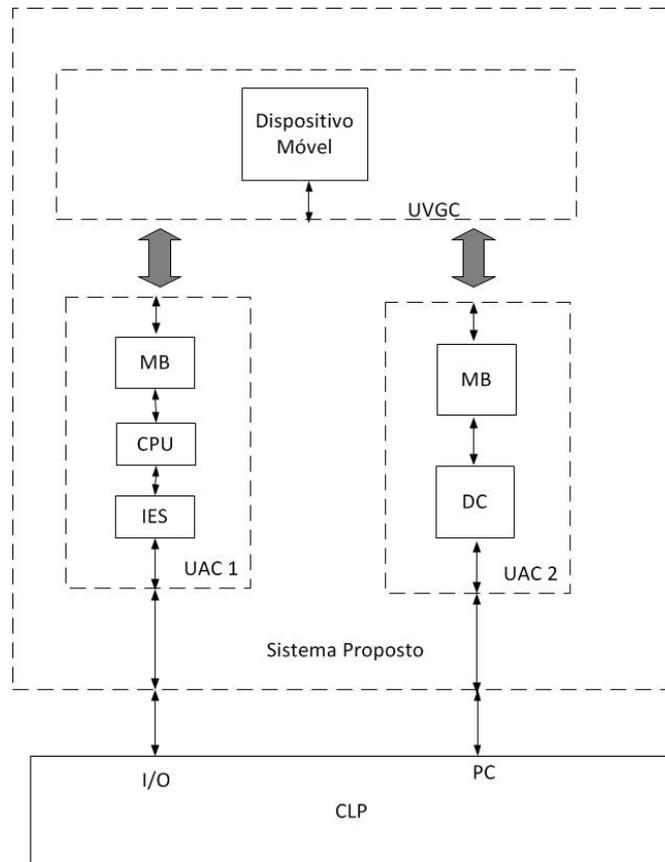


Figura 2. Diagrama de blocos das possíveis implementações do projeto.

O sistema pode ser dividido em uma Unidade de Aquisição e Controle (UAC), que coleta as informações dos sensores e CLP e as interpreta, e uma Unidade de Visualização, Gerenciamento e Controle (UVGC), que recebe os dados enviados pela UAC. A UVGC é a parte do sistema que permitirá ao usuário visualizar as informações desejadas e interagir com o processo.

São propostos dois possíveis modos de implementação da UAC, sendo diferenciados pelo modo de comunicação entre o projeto desenvolvido e os dispositivos existentes a nível de chão de fábrica. A primeira configuração apresentada UAC1 coleta os dados dos sensores ou envia aos atuadores sem utilizar um protocolo de comunicação específico. Na segunda forma, UAC2, o sistema se comunica com o CLP através do protocolo Modbus.

A UAC1 adquire os dados dos sensores e atuadores e utiliza um microcontrolador (CPU) para interpretá-los. Em seguida, o microcontrolador se comunica com o dispositivo móvel, por meio de um módulo Bluetooth (MB), onde os dados são exibidos para o usuário. Para tornar possível a conexão do microcontrolador com o CLP, é necessário um estágio intermediário entre eles, a Interface de Entrada e Saída (IES). O microcontrolador utilizado foi o Arduino Mega 1280, pois possui uma linguagem de programação simples e mais de uma porta para comunicação serial, possibilitando a comunicação do Arduino com a IES e com o módulo Bluetooth.

A configuração UAC1 possibilita que os valores coletados sejam obtidos de um CLP ou que sejam obtidos diretamente dos sensores e atuadores da planta. A conexão física do sistema com a instalação está ilustrada na Figura 3.

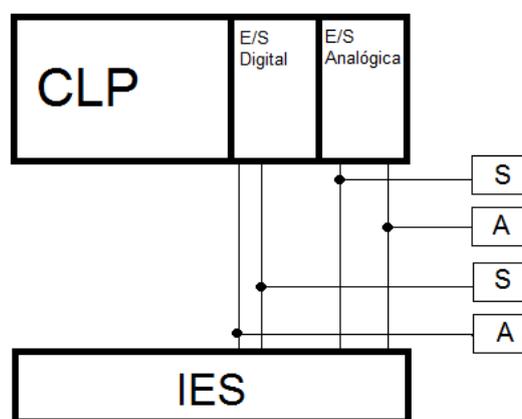


Figura 3. Comunicação entre os sensores e atuadores com o CLP através da Interface de Entrada e Saída.

Os níveis de tensão de um sinal enviado pelo CLP podem variar de 0V a 24V para um sinal digital e de 0V a 10V para um sinal analógico, logo é preciso que a IES seja desenvolvida para possibilitar a conexão desses sensores ao Arduino, que possui tensão de funcionamento de 3,3V ou 5V.

A IES é um circuito composto por um buffer e um divisor resistivo, que permite que a conexão do CLP e do microcontrolador seja realizada sem que haja uma modificação na impedância do circuito existente na planta a ser monitorada. As configurações para uma entrada de sinal analógico e para uma de sinal digital estão ilustradas nas Figuras 4 e 5.

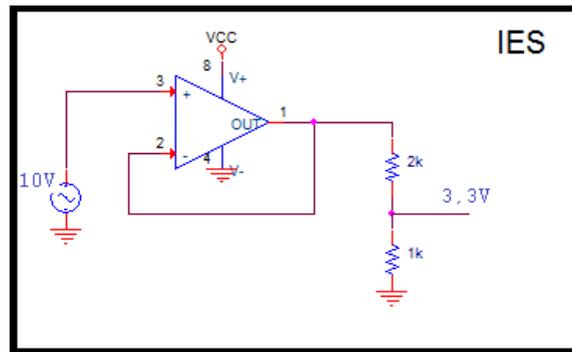


Figura 4. IES para sinal analógico.

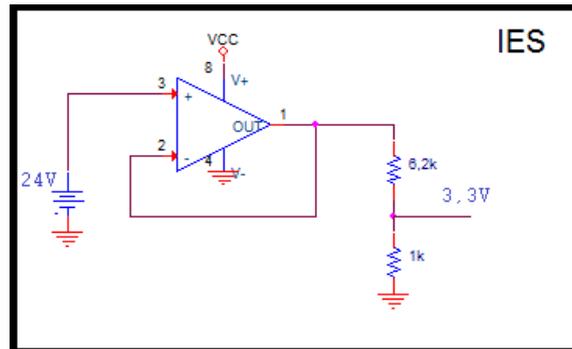


Figura 5. IES para sinal digital.

A segunda configuração (UAC2) dispensa a utilização do microcontrolador, pois estabelece a comunicação do sistema com a planta a partir do protocolo de comunicação determinado pelo CLP. O CLP utilizado foi o TPW-03 da WEG, que utiliza o protocolo Modbus. Ou seja, o comando pode ser enviado pelo dispositivo portátil e a resposta também interpretada por ele, sem a necessidade de um microcontrolador para executar essa função.

A comunicação entre o dispositivo portátil e o sinal enviado pelo CLP é realizada através de um módulo Bluetooth (MB), que recebe o sinal da planta por meio de um *driver* de comunicação (DC) e o repassa ao dispositivo portátil. O protocolo Modbus utiliza como camada física o padrão RS-485, então foi utilizado como *driver* o ADM485, que é um transceptor diferencial adequado para a comunicação bidirecional de dados de alta velocidade em linha de transmissão de um barramento de multipontos. Ele foi utilizado como transceptor do sinal UART do microcontrolador para o padrão RS-485.

O CLP é conectado ao microcontrolador ou ao DC por meio do barramento serial. O sinal enviado pelo controlador lógico e recebido por ele é dado através dos pinos TX e RX do Arduino e ADM485. O sinal proveniente do CLP é transmitido por dois canais, A e B, que conduzem sinais semelhantes, mas opostos, o que torna

vantajoso realizar a subtração entre eles para observar o valor da saída e diminuir o ruído.

Como a comunicação entre a UVGC e a UAC é realizada pelo módulo Bluetooth, é necessário que o dispositivo móvel tenha suporte para esse tipo de conexão. Nesse caso, o dispositivo móvel utilizado foi o *smartphone* Galaxy S4. O módulo Bluetooth utilizado limita a distância de alcance do sistema, o que torna o projeto menos vulnerável a invasões. O módulo utilizado foi o JY-MCU v1.03 da HC, que opera a 3,3V e é compatível com o sinal proveniente do Arduino.

As duas configurações apresentadas possuem vantagens específicas. A primeira, utilizando o microcontrolador, pode ser utilizada em qualquer tipo de CLP ou diretamente conectada a sensores e atuadores. Como a implementação não depende de um protocolo específico, não necessita modificações em software e hardware para diferentes tipos de planta industrial.

A segunda configuração, comunicando-se pelo protocolo Modbus, torna o sistema dependente do CLP, que necessita ter como padrão esse tipo de comunicação. Mas apresenta como vantagem não necessitar de um microcontrolador para compor o sistema. Ou seja, a arquitetura é mais simples e o valor gasto na montagem do circuito é inferior ao da primeira implementação, já que só necessita do *driver* de comunicação, que para o protocolo Modbus possui um baixo custo, logo o valor é menor que o necessário para a UAC1.

O Apêndice A ilustra o desenvolvimento da placa de circuito impresso que possui as duas possibilidades de implementação do sistema inseridas na mesma PCI. A mesma placa pode ser usada também para os dois tipos de variáveis trabalhadas, ou seja, para variável analógica e digital.

3.2 SOFTWARE

O aplicativo que permite ao sistema se comunicar com o operador foi desenvolvido para sistema Android utilizando a plataforma Eclipse, que possui seu código livre, além de ser uma IDE (*Integrated Development Environment*) que compreende vários tipos de linguagem e que aceita a instalação de emuladores para testar o desenvolvimento da plataforma. Devido a possibilidade de programar linguagem Android, essa IDE foi escolhida para utilização nesse projeto de

desenvolvimento de um aplicativo para esse sistema. Foram utilizados para testes o emulador disponibilizado pelo Eclipse e o *smartphone* Samsung Galaxy S4.

A UAC, independente do tipo de implementação, adquire os valores das variáveis do processo produtivo, que em seguida são interpretados e apresentados ao usuário por meio do aplicativo desenvolvido.

Na UAC1, os valores coletados dos sensores e atuadores são fornecidos como sinais que podem assumir nível 0V ou VCC, para variáveis digitais, ou valores definidos dentro de uma faixa de tensão, para variáveis analógicas. Esses valores são adquiridos e analisados pelo microcontrolador Arduino, que enviará para o dispositivo móvel o que deverá ser apresentado ao usuário.

Inicialmente o Arduino é configurado quanto a taxas de transferência e definição de pinos. Em seguida, aguarda para saber se será realizada a aquisição analógica ou digital. Se for requisitada uma variável analógica, o microcontrolador lê o valor analógico e quando finalizada (tL) envia para o dispositivo móvel através do módulo Bluetooth. Caso seja uma variável digital, o Arduino confere se a entrada referente está em nível lógico alto ou baixo e terminada a leitura (tL) executa o procedimento de envio. Após finalizado esse procedimento, o microcontrolador volta para esperar qual variável será lida e repetir o processo. A Figura 6 representa o diagrama de estados referente a essa configuração.

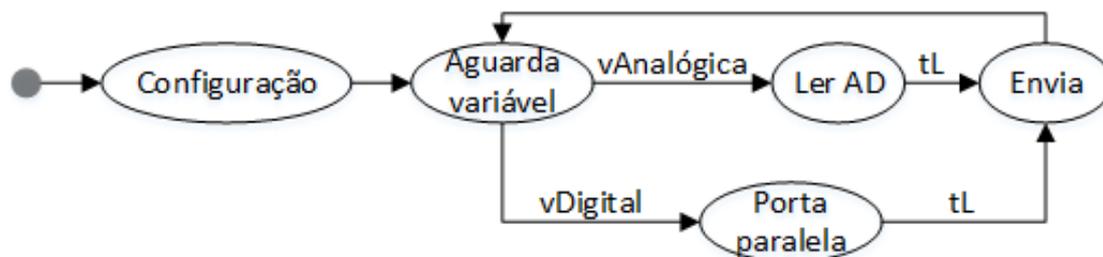


Figura 6: Diagrama de estados do microcontrolador

Na UAC2, o protocolo Modbus é utilizado para comunicação entre o dispositivo portátil e o chão de fábrica. O Modbus possui comandos pré-definidos que o aplicativo deve enviar para o CLP de acordo com a função a ser executada. Os comandos para ler valores de bobinas que contém os valores digitais e escrever nas mesmas ou ler valores analógicos em registradores e escrever neles devem ser enviados pelo *smartphone* e o CLP irá enviar a resposta seguindo o mesmo protocolo.

Na segunda configuração, cabe ao software desenvolvido para o aplicativo interpretar os valores enviados pelo CLP e definir qual o valor de interesse para o

usuário. Em seguida, deve apresentá-lo de forma simples e dinâmica através de imagens e gráficos.

As funcionalidades do sistema estão ilustradas no diagrama de caso de uso da Figura 7.

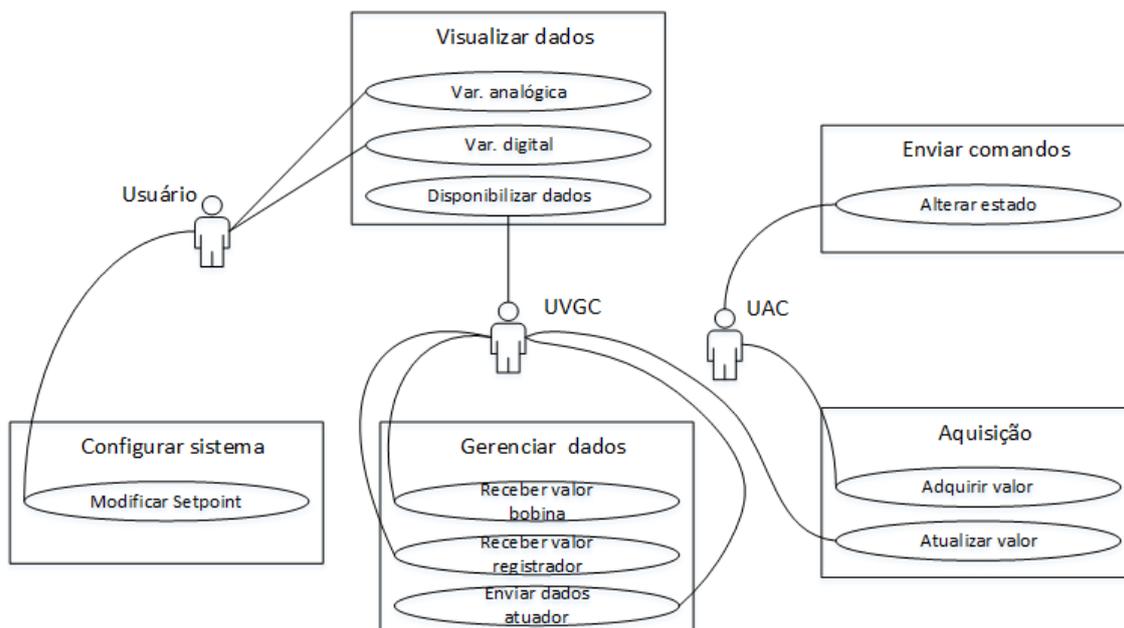


Figura 7: Diagrama de caso de uso.

A UAC envia as requisições ou comandos e adquire os valores necessários. A UVGC deve receber os valores adquiridos e apresentá-los ao usuário. Além disso, deve enviar os dados para os atuadores, para isso, envia os dados para a UAC que irá se comunicar com a planta. A UVGC deve manter a atualização das variáveis sempre apresentáveis. O usuário deve ser capaz de visualizar os valores e configurar o *setpoint* do sistema.

A comunicação estabelecida por meio do protocolo Modbus envia os valores determinados em hexadecimal, seja para escrever ou ler de uma bobina. A sequência de comandos deve ser respeitada e o endereço requisitado define qual sensor é desejado acompanhar ou qual atuador deverá modificar. É necessário também o envio de um CRC-16, que indica se os comandos estão de acordo com o esperado ou se ocorreu algum erro durante a tentativa de comunicação. Essa sequência segue o padrão apresentado na seção “Protocolo de Comunicação Modbus”.

Quando o TPW-03 recebe um comando da UAC, o CLP executa o comando e envia uma resposta para a UAC. Se houver um erro no conteúdo do comando, o TPW-03 apresenta uma resposta de erro para a UAC. A sequência de comandos a ser enviados

começa com o endereço do escravo com o qual a comunicação será estabelecida. Em seguida, é enviado o código da função que será realizada. Os dados enviados em seguida contêm as informações necessárias de acordo com o tipo de função a ser executada. Por fim, é enviado o CRC-16, que é o método utilizado para detecção de erro, que mostra caso tenha ocorrido falha na comunicação.

Após enviado um comando, o CLP envia uma resposta à UAC seguindo um padrão semelhante ao enviado pela unidade de aquisição. O controlador lógico envia em resposta o mesmo endereço e código de função recebidos e os dados de resposta, que depende de cada informação requisitada. Ele calcula um novo CRC e envia indicando se a comunicação foi realizada com sucesso, caso tenha acontecido algum erro, o CLP acrescenta 80H ao código da função para o sistema principal ser informado do erro ocorrido.

Foram utilizadas funções de leitura e escrita de bobinas, para que o sistema pudesse se comunicar com sensores e atuadores digitais. Para as variáveis analógicas do processo foram utilizadas as funções de leitura e escrita de registradores. Cada uma dessas funções possui um código que deve ser enviado após o endereço do escravo. Depois da função é necessário definir qual o endereço de início da leitura ou escrita e em quantos bytes se deseja escrever ou realizar a leitura. Os endereços disponíveis para cada situação estão descritos no manual do fabricante do CLP (WEG, 2010).

O aplicativo, nas duas formas de implementação, é habilitado para o protocolo Bluetooth, pois a comunicação estabelecida pelo usuário deve ocorrer durante rondas na empresa, quando o mesmo irá se posicionar próximo ao equipamento a ser inspecionado e conferirá o valor das variáveis do processo. A comunicação por meio do protocolo Bluetooth limita a distância de conexão entre o aparelho portátil e o sistema proposto, logo previne contra possíveis invasões e torna o sistema mais seguro.

3.3 DESCRIÇÃO DA INTERFACE

O aplicativo possui uma tela inicial onde o usuário pode escolher que tipo de variável será analisado, se digital ou analógico. Nesse caso, pode ser especificado de acordo com a planta industrial a variável, como temperatura, detecção de presença no sensor, dentre outros parâmetros. As telas de Menu Inicial, de visualização de uma

variável digital e de visualização de uma variável analógica estão representadas na Figura 8.

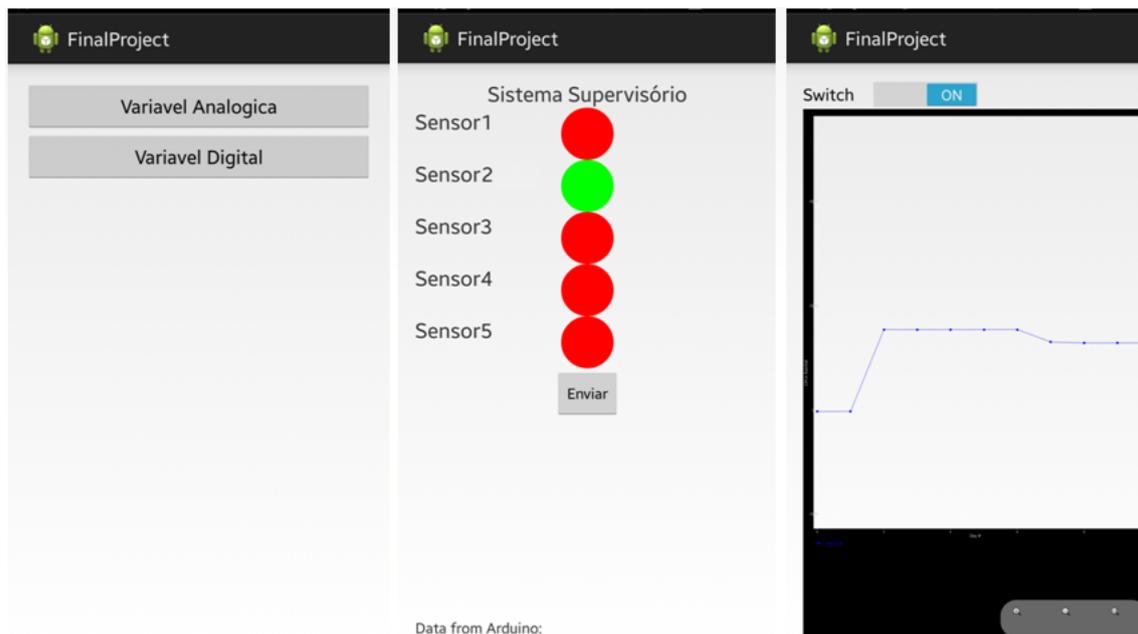


Figura 8: Telas do aplicativo.

As variáveis coletadas podem ser digitais ou analógicas. Os sensores digitais são apresentados ao usuário por meio de um layout gráfico onde formas geométricas representam cada sensor da linha produtiva. As variáveis analógicas são expostas em forma de gráficos de linha, onde o usuário pode visualizar dinamicamente as alterações ocorridas em tempo real, além de modificar o zoom e o parâmetro de tempo.

Para variáveis digitais, foi desenvolvido um *layout* para sensores que exhibe o estado do sensor no momento de visualização. Caso o estado mude durante a exibição para o operador, o aplicativo acompanha a mudança e expõe para o usuário a situação atual de cada sensor. Por exemplo, caso um sensor indutivo localizado em uma esteira detecte algum objeto, o aplicativo irá exibir essa detecção modificando o objeto relacionado ao determinado sensor em questão.

As variáveis analógicas são enviadas em 12 bits que são interpretados e apresentados ao usuário em forma de gráfico de linha com o valor recebido convertido para decimal. O gráfico inicia o recebimento quando requisitado pelo operador e é atualizado em tempo real de forma dinâmica. O gráfico desenvolvido possibilita ao usuário aproximar ou afastar com o comando de zoom e também, voltar e verificar um valor passado.

O aplicativo foi desenvolvido para uma aplicação geral de variáveis analógicas e digitais, mas pode ser facilmente adaptado para as necessidades individuais de cada planta industrial. Assim, é possível especificar cada parâmetro a ser acompanhado e controlado. O sistema em teste, para variáveis digitais, pode ser analisado na Figura 9.

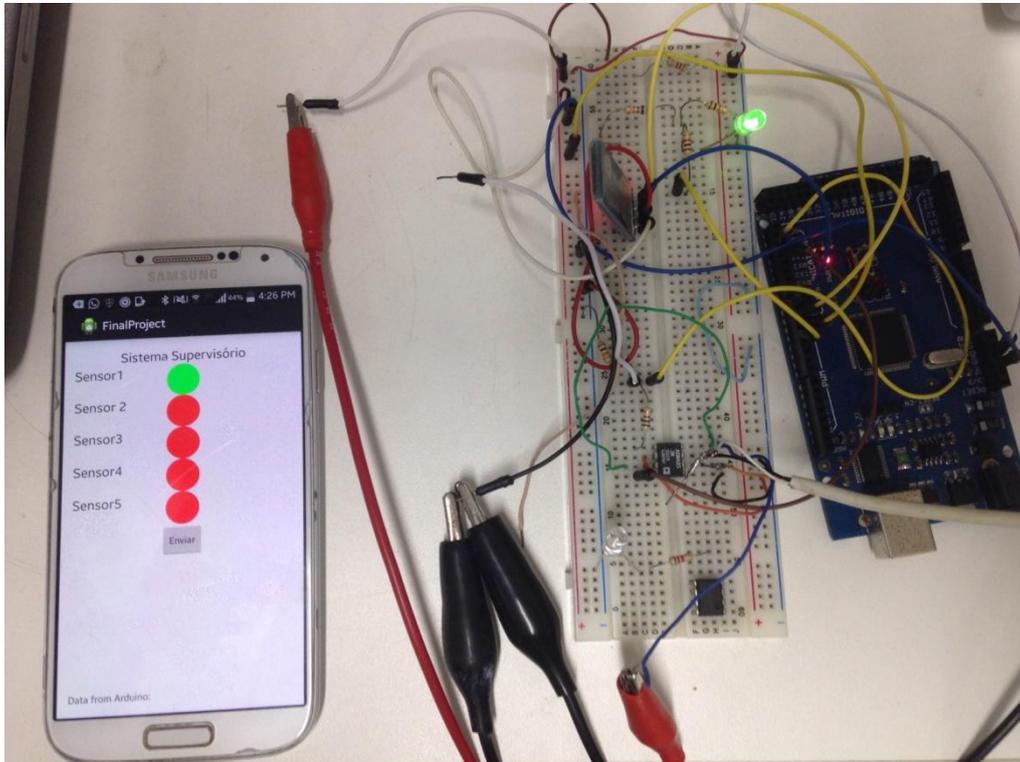


Figura 9: Aplicativo e sistema desenvolvido.

A Figura 10 ilustra o diagrama de estados do processo. O software é iniciado na tela do Menu Principal, onde o usuário pode selecionar para acompanhar uma variável digital, selecionando o botão bDigital, ou uma variável analógica, clicando no bAnalógico.

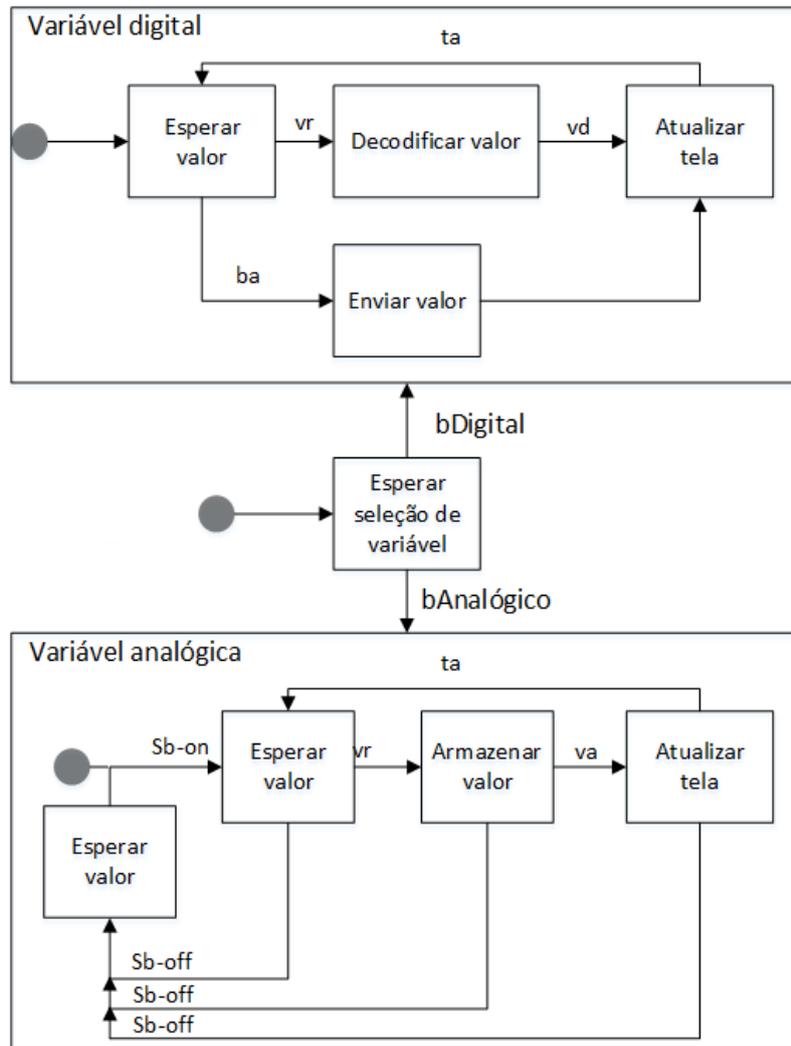


Figura 10: Diagrama de estados da UVGC.

Caso seja selecionada a visualização de uma variável digital, o software espera receber o valor do sistema. Assim que o valor for recebido (vr) o software decodifica esse valor e após o valor ser decodificado (vd) o supervisor atualiza a tela. Após a tela atualizada volta para a espera de valores. Se durante esse tempo, for selecionado o botão de enviar para um atuador (ba), o sistema envia o valor e após enviado (ve) atualiza a tela, caso necessário.

Caso selecionado o botão para variável analógica, o sistema espera receber um valor. Após ser recebido (vr), o valor é armazenado e quando concluída esta etapa (va), o software atualiza a tela. Vale salientar que o gráfico é ativado por um botão que varia o estado em “ON” ou “OFF”, e caso o botão seja variado para o estado OFF (Sb-off) durante a exibição, o gráfico para de atualizar e mantém os resultados já adquiridos na tela.

3.4 ANÁLISE DOS RESULTADOS

Para analisar os resultados, foi inicialmente testado o funcionamento do circuito para variável digital. Foram realizados testes nos sensores óptico, difuso, capacitivo e indutivo do painel DeLorenzo modelo DLB PTSCS do Laboratório de Acionamento e Controle (LAC), ilustrado na Figura 11. Esses sensores respondem com apenas dois níveis de tensão para o controlador. Foi analisado se o sinal que o Arduino recebia era nível lógico alto ou baixo e visualizado o resultado no acendimento de um LED. Em seguida, foi adicionada ao teste a comunicação por Bluetooth para visualizar o estado do sensor no aplicativo. O sinal recebido pelo Arduino é interpretado e caso o sensor detecte alguma superfície o microcontrolador envia a palavra “ON” pela serial para o módulo Bluetooth, caso contrário, envia a string “OFF”.

Esses sensores analisados foram indutivo, capacitivo, ótico de reflexão, ótico difuso e ótico receptor. A saída dos sensores do painel é conectada a IES que faz o condicionamento do sinal que em seguida, é enviado ao microcontrolador e ao dispositivo móvel através do módulo Bluetooth. Essa plataforma de testes pode ser observada na Figura 12.

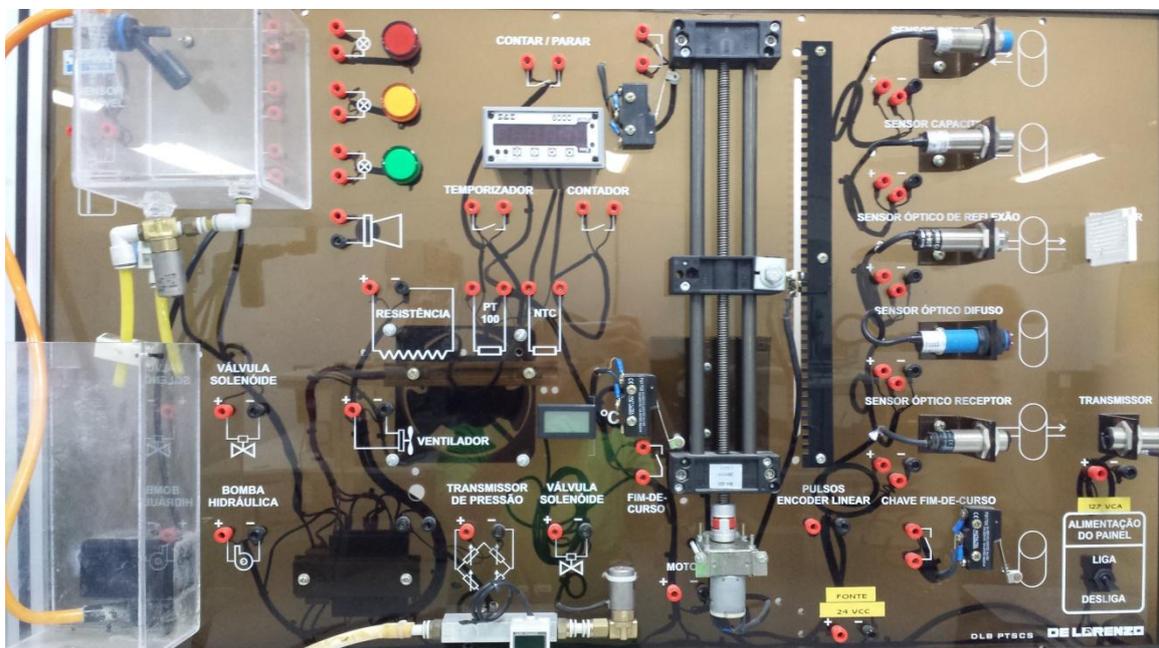


Figura 11: Painel utilizado para testes.

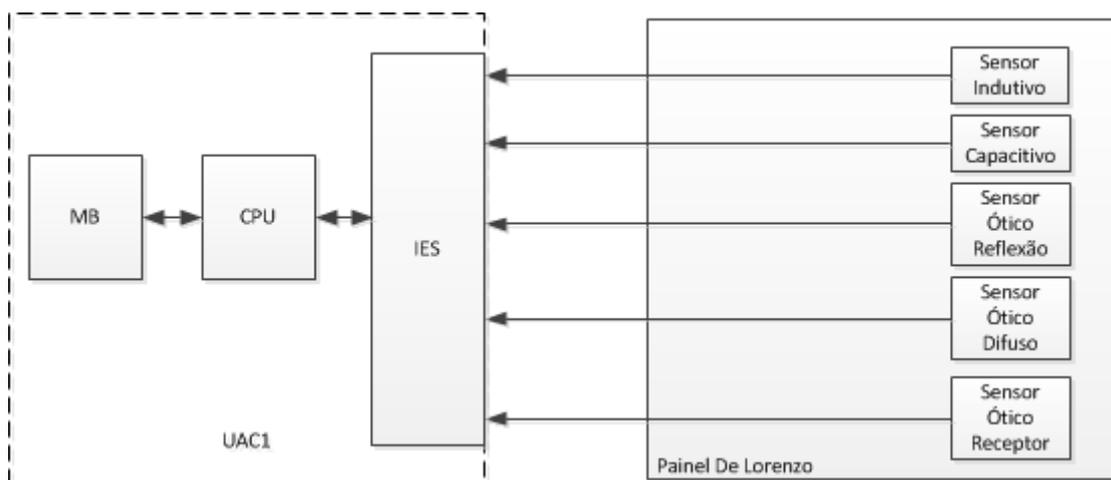


Figura 12: Plataforma de testes para UAC1 com variável digital.

O resultado enviado pelo módulo Bluetooth para o dispositivo móvel é analisado pelo aplicativo e executada a atividade determinada. O teste final com variável digital adquirida diretamente do painel foi analisar a mudança de estado dos diferentes sensores no aplicativo. As Figuras 13 e 14 ilustram a tela do dispositivo no momento da realização do teste para diferentes estados do Sensor 1.

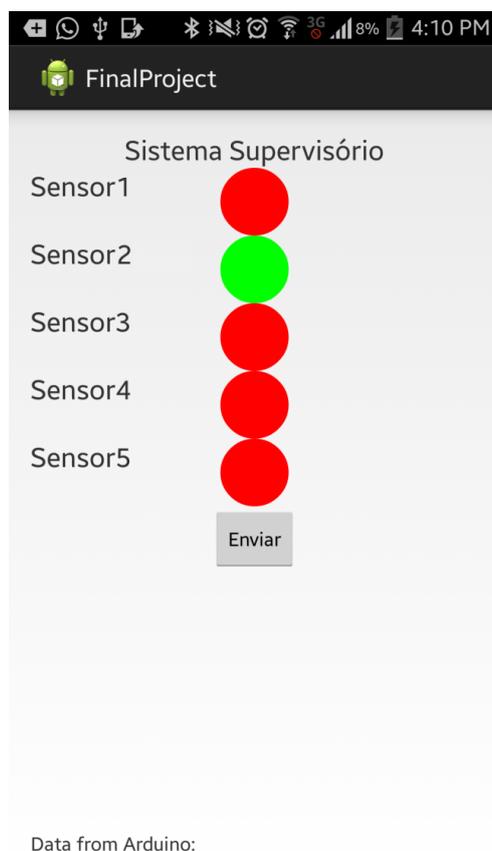


Figura 13: Tela aplicativo com Sensor 1 em OFF.

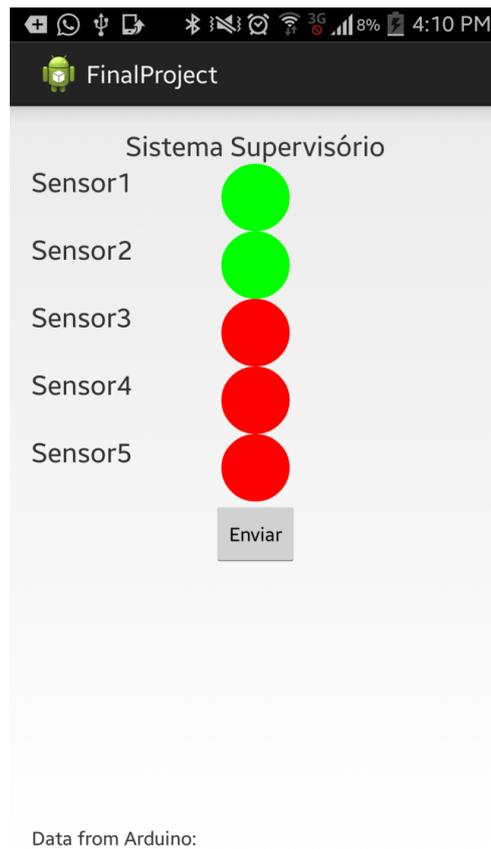


Figura 14: Tela do aplicativo com Sensor 1 em ON.

Em seguida, foi analisado o Protocolo Modbus para escrita em uma bobina do CLP, ou seja, na prática, ligar ou desligar um atuador. Foi seguida a sequência de comandos do protocolo e enviado primeiro o endereço do escravo em hexadecimal 0x01H, que corresponde ao primeiro escravo. Em seguida, enviou-se o código da função 0x05H, que representa a função “escrever em uma única bobina”. O campo seguinte, campo de dados, é formado pelo endereço da bobina e valor da mesma, cada um deles composto por dois bytes. O endereço utilizado para teste foi a terceira bobina do primeiro escravo, representada pelo valor 0x01H seguido de 0x02H. O valor da bobina é dividido em dois bytes, para ativar a bobina o valor enviado é 0xFFH e para desativar 0x00H. Para finalizar a sequência de comandos para escrever em uma bobina, são enviados mais dois bytes de CRC-16, que nesse caso específico é 0x062CH.

O teste foi realizado no CLP WEG TWP-03 e conferido o resultado com o acendimento de um LED no próprio controlador que representa o estado da bobina. Além disso, é possível conferir através da resposta do CLP para o sistema desenvolvido, que segue o mesmo padrão do comando enviado. Ao analisar, deve-se considerar que a cada byte enviado pelo Arduino é precedido de um bit de início e precedido de um de

parada. Já a resposta enviada pelo CLP possui o mesmo bit de início e dois bits de parada, como citado no anteriormente protocolo. O sinal de requisição e resposta pode ser observado na Figura 15 e na Figura 16 uma visão detalhada da resposta.

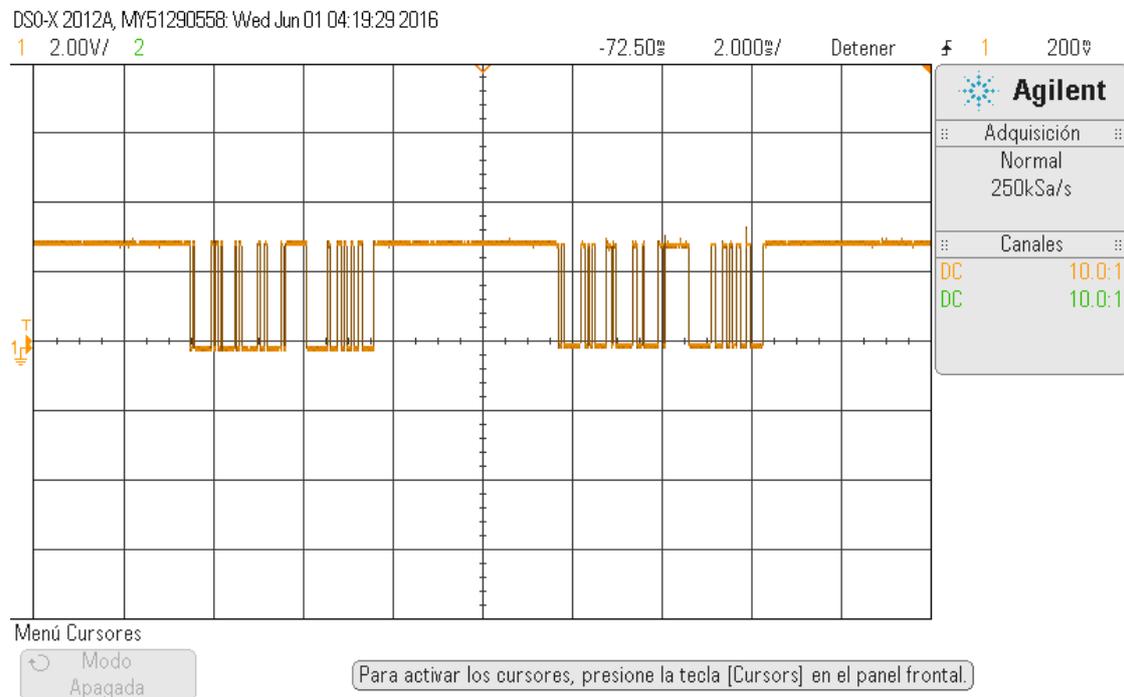


Figura 15: Comando para escrita em uma bobina e respectiva resposta do CLP.

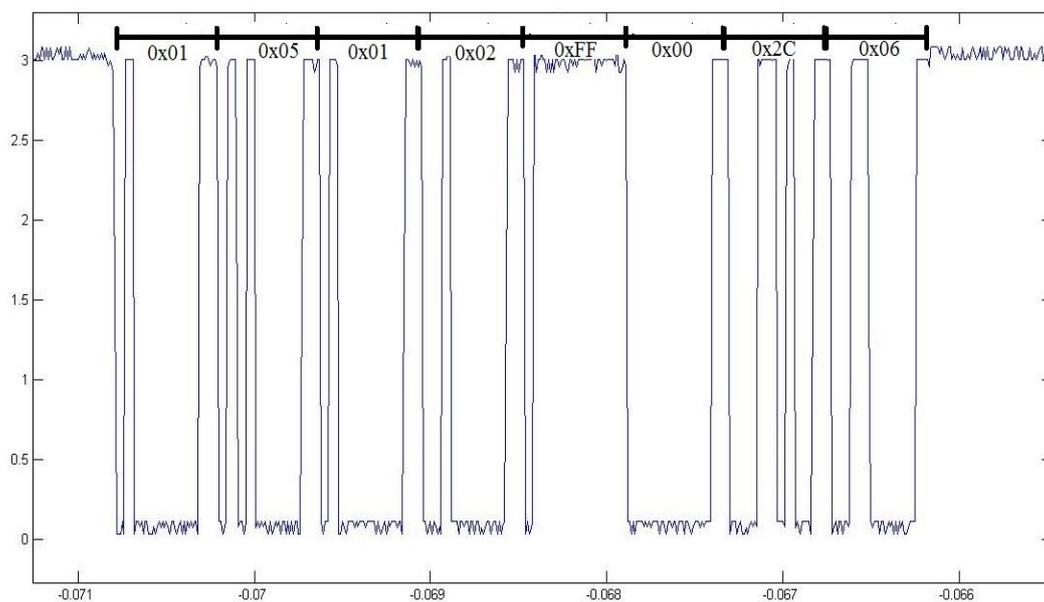


Figura 16: Resposta para escrita em uma bobina.

O protocolo Modbus foi utilizado em seguida para realizar a leitura de uma bobina do CLP. Utilizando linguagem de programação Ladder, foi desenvolvido um sistema que controlasse o estado da bobina por um painel de chaves conectado ao CLP. Ao relacionar o estado de uma chave com o da bobina, foi possível alterar o estado da

mesma e testar o funcionamento da leitura. A chave Q0 representada na Figura 17 foi relacionada com a entrada I0 do CLP e o valor da bobina era alterado pela chave. Os canais A e B enviam a requisição e resposta por Modbus para o CLP e para a UAC2, respectivamente.

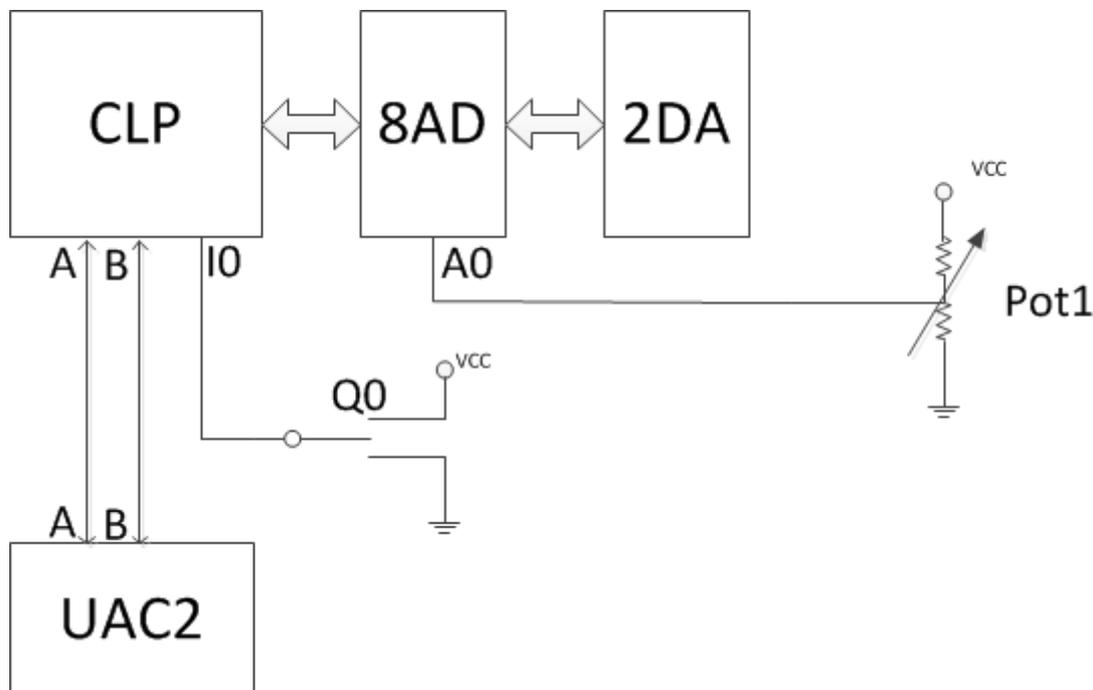


Figura 17: Plataforma experimental da UAC2.

Para leitura da bobina, o valor inicial é o mesmo utilizado anteriormente para função de escrita, que é o endereço do escravo 0x01H. O código da função muda de acordo com a função desejada, nesse caso, para ler os dados de uma bobina é utilizada a função 0x01H. O campo de dados que segue indica a partir de qual bobina será iniciada a leitura, que foi definida a primeira bobina do CLP para ser a inicial, com o valor 0x0100H. O campo de dados é finalizado com a quantidade de bobinas que serão lidas a partir da inicial. A quantidade de bytes é igual à quantidade de bobinas dividido por 8, para ler um único byte, definiu-se o número de bobinas como 0x08H. Por fim, o CRC-16 nesse caso deve ser 0x303CH.

Na função de leitura, o CLP envia como resposta o endereço do escravo e o código da função, iguais aos enviados como requisição se não existir erro. Como sequência, são enviados os dados referentes às bobinas escolhidas e para terminar, o CRC referente aos valores enviados. Os dados de resposta, considerando a função de leitura descrita acima, devem conter o valor em um byte, sendo o bit menos significativo a representação do estado da bobina que terá o valor alterado, já que foi

solicitado o valor da primeira bobina. A resposta deverá variar o bit de acordo com a posição da chave relacionada com a bobina, variando em nível alto ou baixo. O valor do CRC também vai variar de acordo com o estado da bobina.

Inicialmente, o código enviado estava encontrando erro no valor do CRC, então a resposta do CLP era indicando que houve falha na comunicação, ou seja, acrescentando o valor 0x80H ao código da função. Essa o comando e resposta podem ser observados na Figura 18 com a resposta detalhada na Figura 19. Depois de identificado e corrigido o erro no envio do CRC, pôde-se observar a resposta correta enviada pelo CLP, que estão expostas nas Figuras 20 e 21, para bobina desligada, e 22 e 23 para quando estiver ligada.

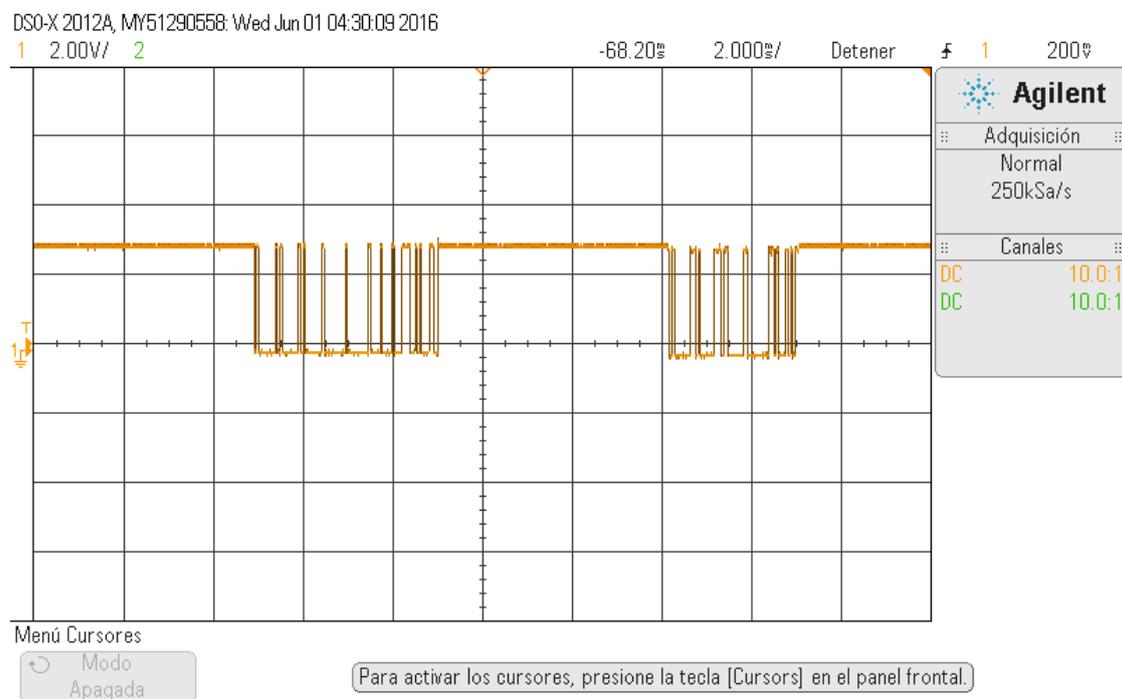


Figura 18: Comando para leitura de uma bobina e respectiva resposta do CLP apresentando erro.

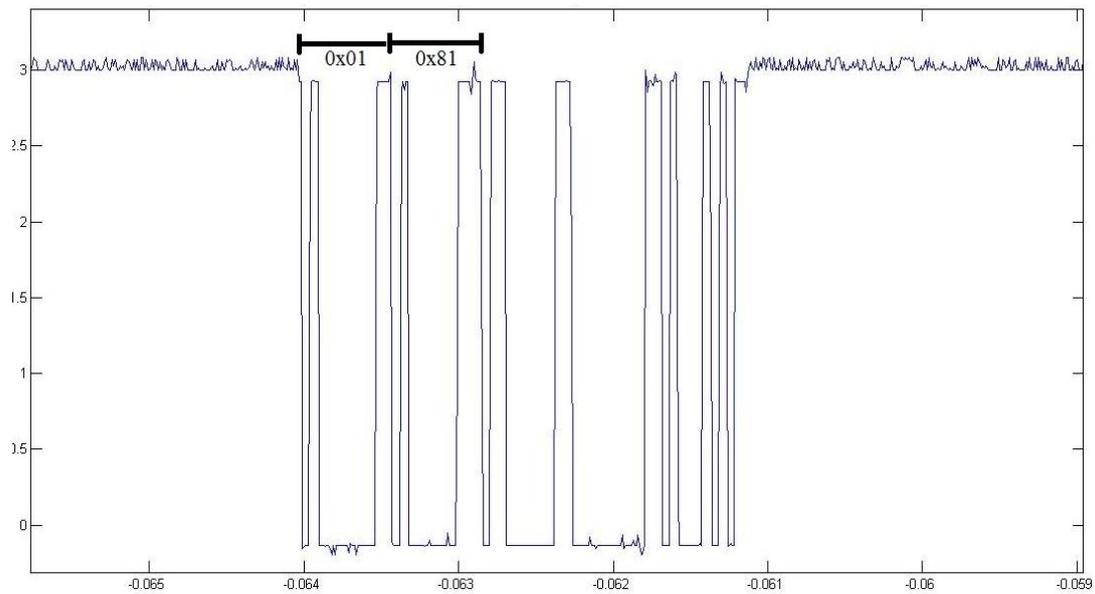


Figura 19: Resposta para ler de uma bobina reportando erro

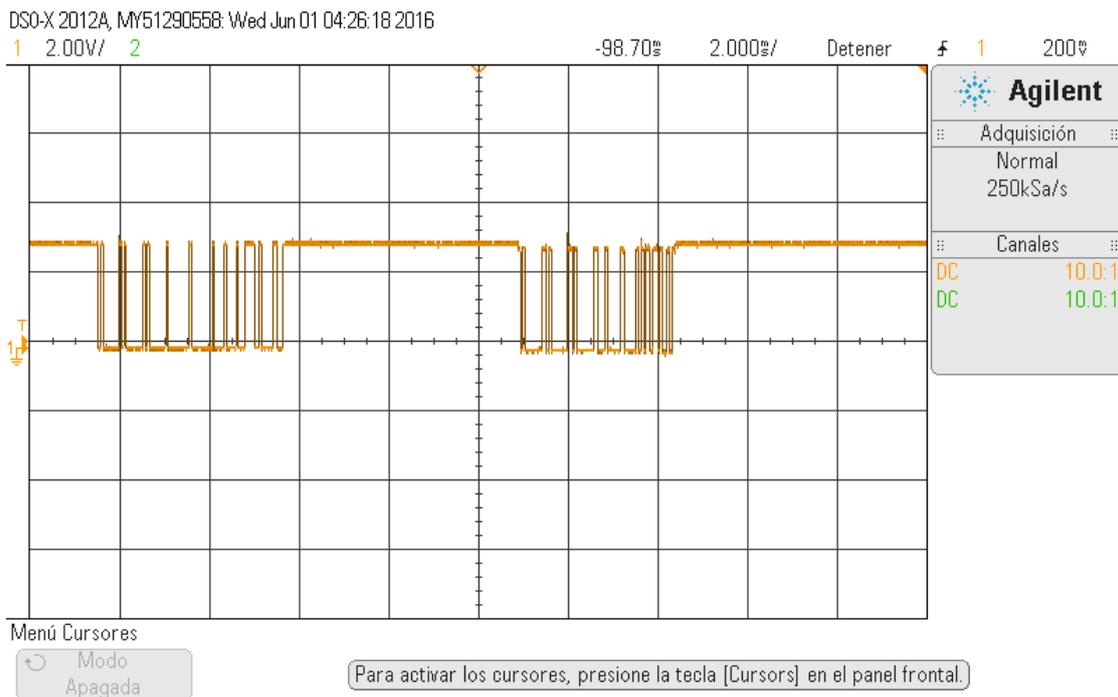


Figura 20: Comando para leitura de uma bobina e respectiva resposta do CLP para bobina desligada.

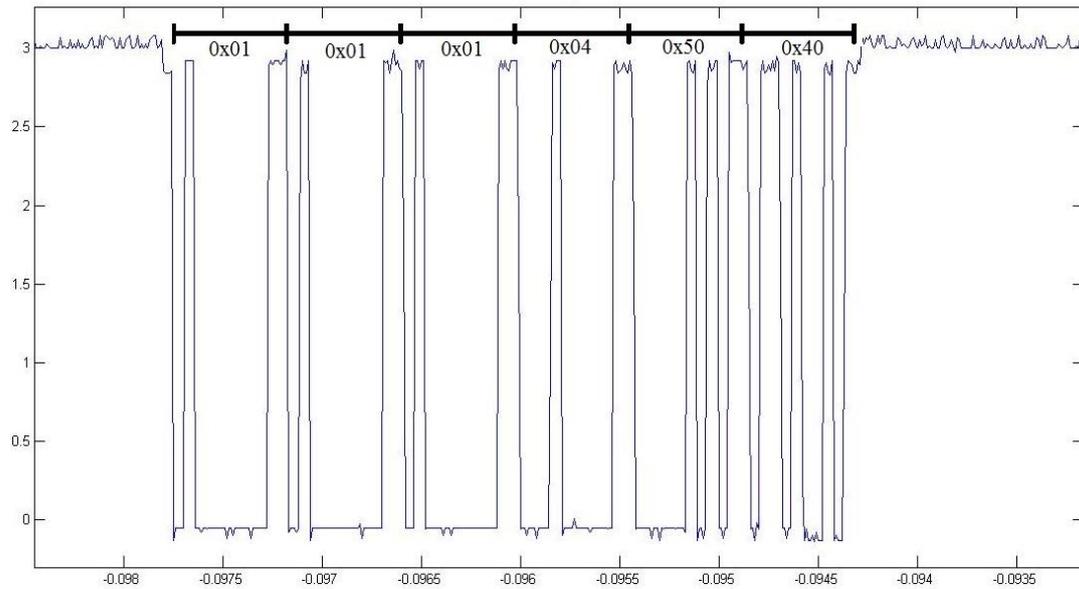


Figura 21: Resposta para leitura de uma bobina desligada.

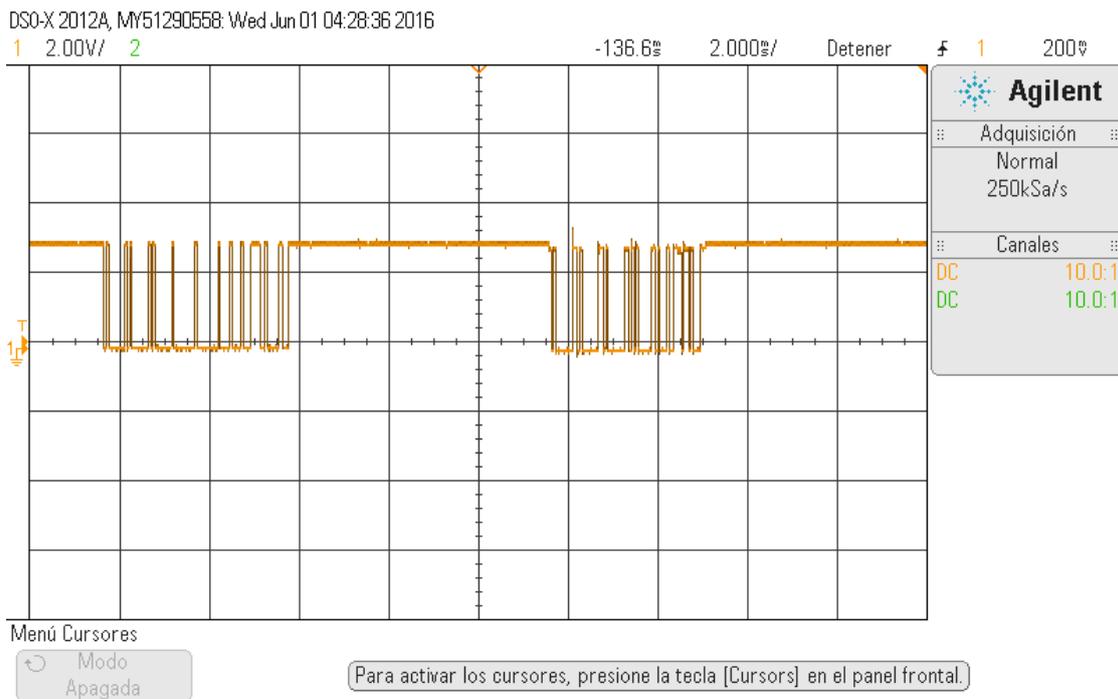


Figura 22: Comando para leitura de uma bobina e respectiva resposta do CLP para bobina ligada.

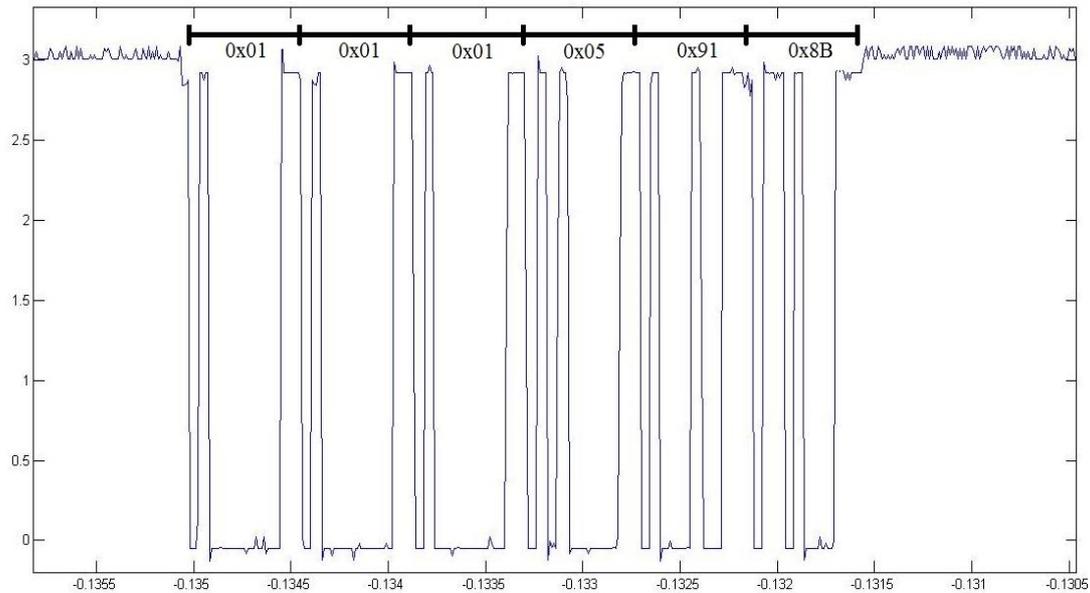


Figura 23: Resposta para leitura de uma bobina ligada.

Depois de realizados os testes para variável digital foi desenvolvido um procedimento semelhante a este para as variáveis analógicas. O procedimento de envio da variável analógica para o atuador foi realizado escrevendo o valor desejado em um registrador do CLP. Para escrever em um registrador é enviado o endereço do escravo 0x01H e em seguida o código da respectiva função 0x06H. O valor do registrador no qual será escrito é enviado em seguida, nesse caso foi utilizado o 0x4340H, que é o valor associado ao registrador D8. Em seguida, é enviado o valor que se deseja escrever, nesse caso 0x08H e por fim o CRC-16. A resposta do CLP confirmando a comunicação é mandada com a mesma sequência e valores recebidos por ele.

Foi possível observar a escrita em registrador analisando a requisição e resposta do CLP, além de desenvolvido um programa em Ladder no programa da WEG, TPW03-PCLINK, para acompanhar o valor do registrador. O valor do registrador é modificado através de um potenciômetro conectado ao conversor AD do CLP. Esse potenciômetro faz parte do *kit* didático da WEG disponível no laboratório. O painel com o potenciômetro e o CLP utilizados está ilustrado na Figura 24 e o diagrama de blocos na Figura 17. Essas duas formas de visualização estão expostas nas Figuras 25 e 27. E a resposta pode ser confirmada na Figura 26.

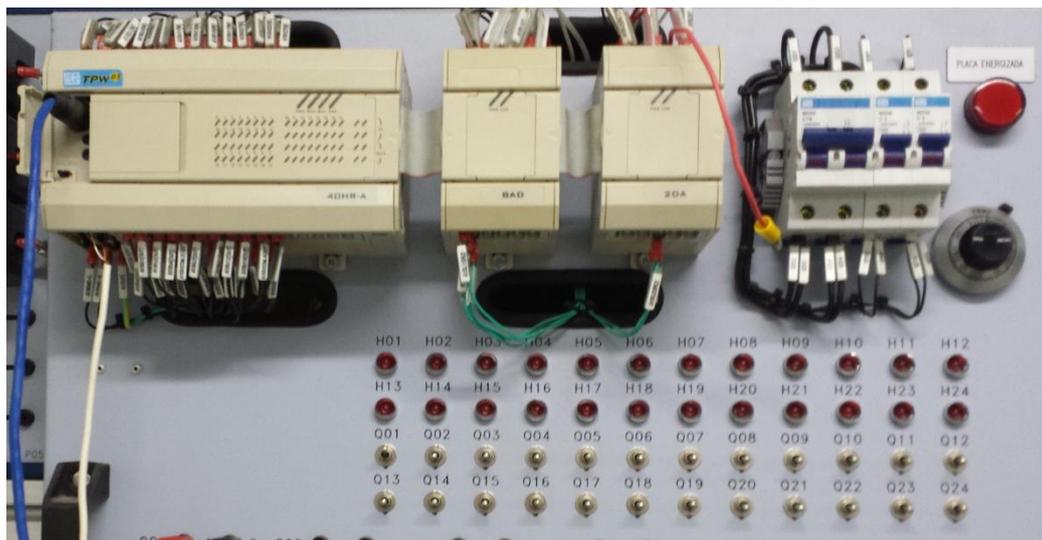


Figura 24: Painel com o CLP e potenciômetro.

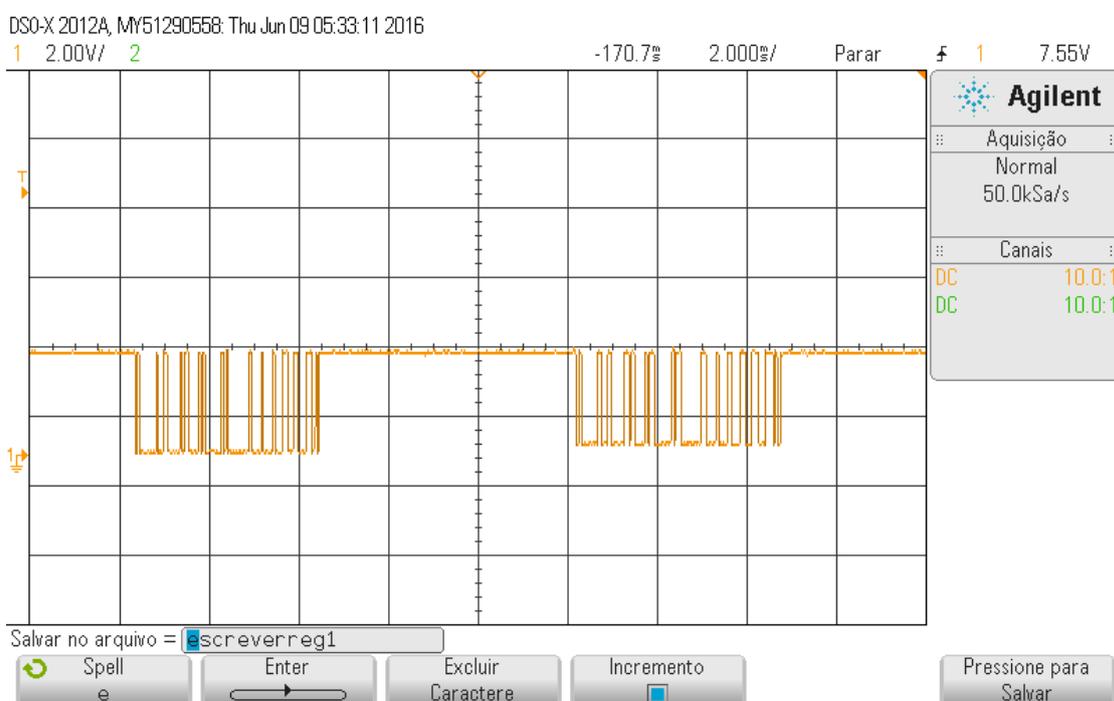


Figura 25: Comando para escrita em um registrador e respectiva resposta do CLP.

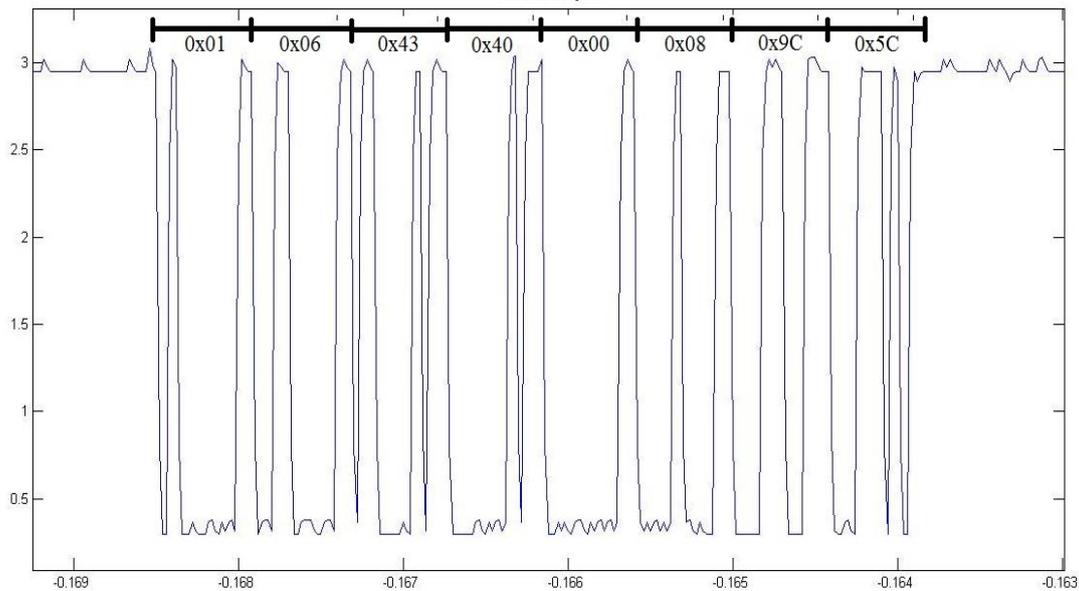


Figura 26: Resposta para escrita em um registrador.

Na aquisição dos resultados para variáveis analógicas, o endereço do escravo enviado é o mesmo dos enviados para as demais funções, 0x01H. Em seguida, o código para leitura de um registrador, que é dado por 0x03H. O campo de dados começa com o endereço de início, que foi definido como o primeiro registrador D0, escolhido de forma arbitrária, e o código de acesso de acordo com o manual do CLP é 0x4338H. O final do campo de dados é um valor que representa metade da quantidade de bytes a serem lidos, nesse caso definiu-se como 0x0001H para serem lidos 2bytes, pois o CLP possui 12 bits para conversão AD. O último valor é como padrão o CRC de acordo com os dados escolhidos.

Para teste, foi desenvolvido um programa em Ladder para variável analógica que adquire o valor da entrada analógica do CLP e move o valor para o registrador D0. O valor analógico varia entre 0 e 4000, pois varia 12 bits dos 16 requisitados, devido ao conversor AD do CLP, além de trabalhar com uma margem de erro, já que não chega a 4096 no valor máximo. O valor alterado pelo usuário durante as análises foi enviado ao circuito desenvolvido para análise.

Os bytes de resposta do CLP começam com o endereço do escravo e o código da função. Em seguida, o controlador envia o valor da quantidade de bytes que foi requisitada e depois os dados com a resposta. Por fim, é enviado o CRC para confirmar que não houve erro. A resposta foi observada inicialmente no osciloscópio e em seguida, observado o gráfico no celular com valores que variavam de 0 a 4000. Esses valores podem ser convertidos de acordo com a variável que será analisada. O teste

observado na Figura 26 é para um valor analógico aproximadamente 520 armazenado no registrador e é possível analisar melhor a resposta enviada pelo CLP na Figura 29. Na Figura 27, é possível observar uma aproximação desse valor, já que o dado exibido na tela oscila um pouco. Além disso, é ilustrada nas Figuras 30 e 31 a tela do dispositivo enquanto modificava-se o valor da entrada analógica do CLP através do potenciômetro.

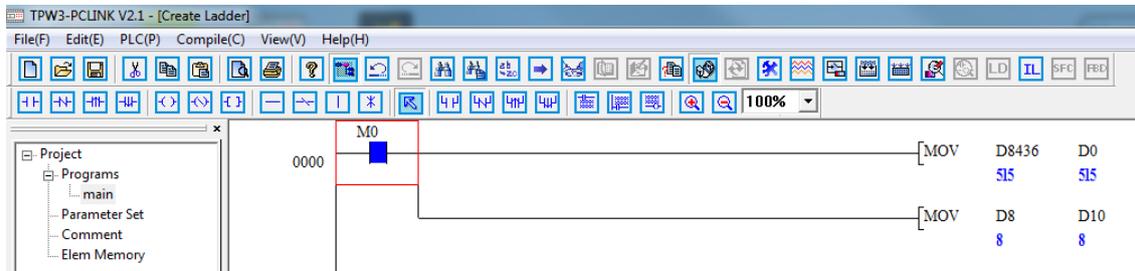


Figura 27: Acompanhamento dos registradores para comandos de leitura e escrita.

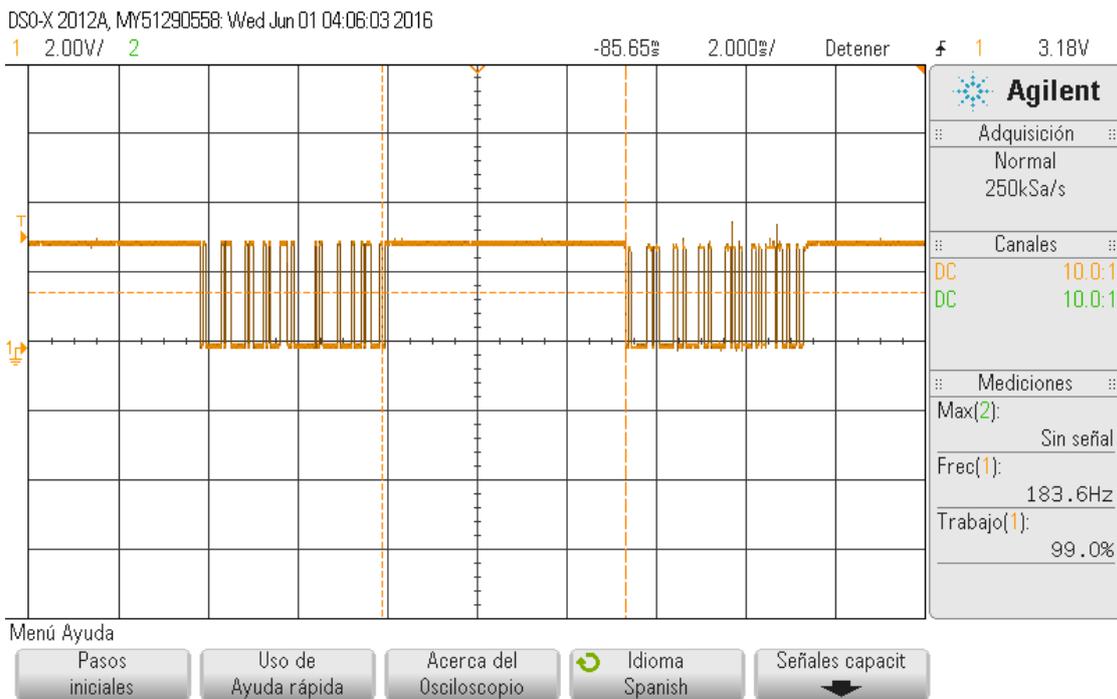


Figura 28: Comando para leitura de um registrador e respectiva resposta do CLP.

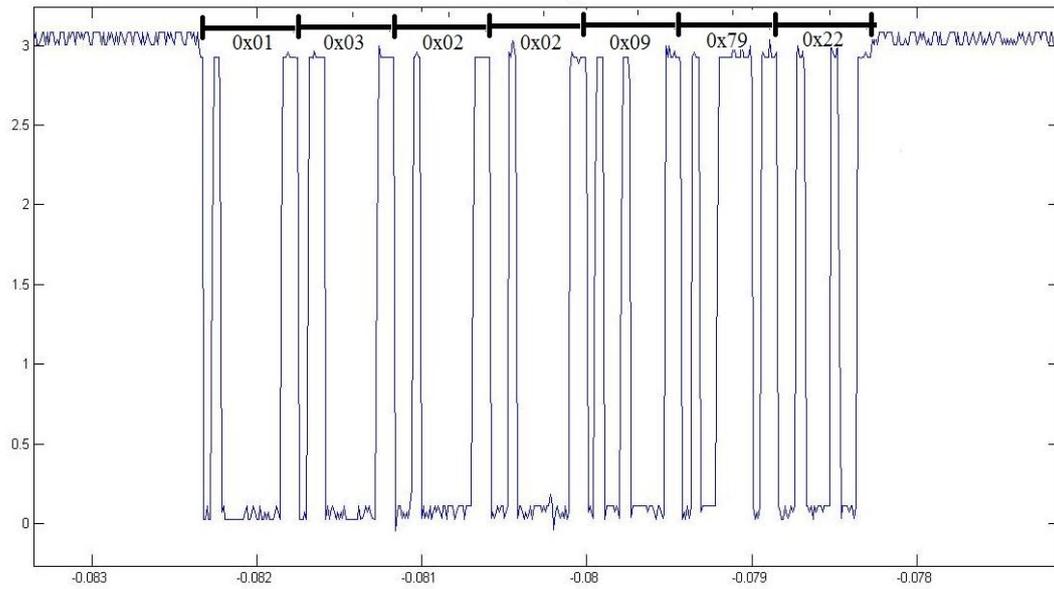


Figura 29: Resposta para leitura de um registrador.

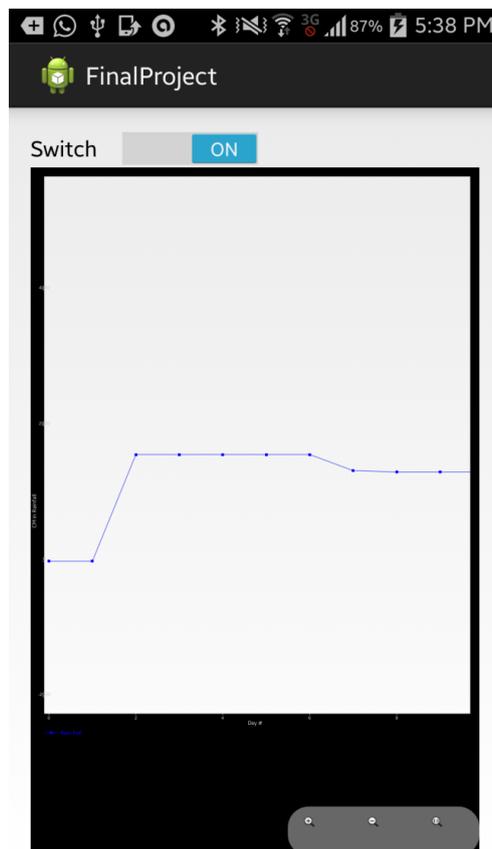


Figura 30: Tela do aplicativo para variáveis analógicas.

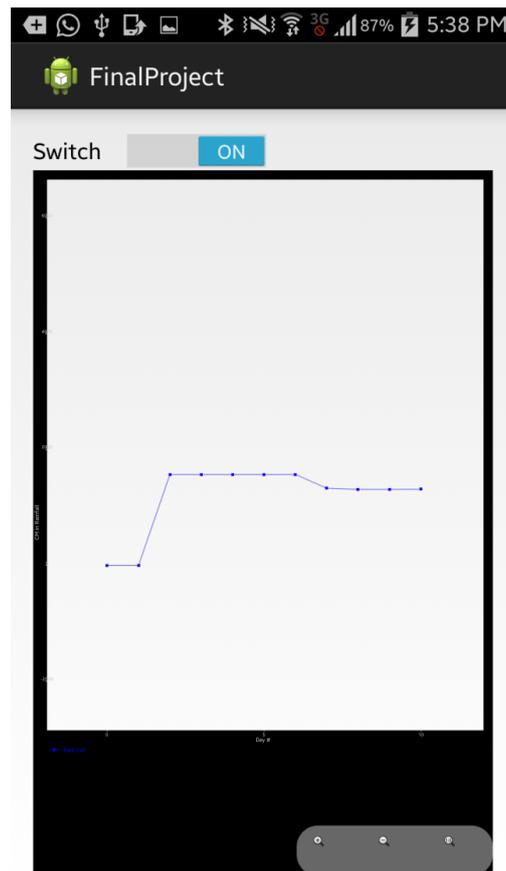


Figura 31: Tela do aplicativo para variáveis analógicas com variação no zoom.

Como pode-se observar nas Figuras 30 e 31, os valores referentes a variação da entrada analógica foram captados pelo dispositivo portátil, que disponibilizou a informação para o usuário através de um gráfico. Desta forma é possível acompanhar a evolução temporal da variável analógica do processo através do dispositivo portátil.

4 CONSIDERAÇÕES FINAIS

Esse projeto desenvolveu um sistema supervisorio capaz de coletar as variáveis digitais e analógicas de uma planta industrial, interpretá-las e apresentá-las ao usuário de forma simples e dinâmica. A exibição pode ser atualizada à medida que o processo produtivo sofra variações.

O sistema apresenta duas possíveis implementações. Na primeira, o projeto é capaz de ser instalado em qualquer tipo de processo industrial, já que não depende de um protocolo específico. Na segunda opção, o sistema se comunica por protocolo de comunicação existente no CLP, necessitando de um drive de comunicação adequado a cada tipo de protocolo utilizado.

Com os resultados, foi possível observar que o operador ou supervisor não necessitam levar diversos equipamentos para verificar os estados das máquinas de um processo, nem precisam conhecer o procedimento necessário para aquisição desses dados. O aplicativo pode se adaptar às diferentes realidades e tornar o trabalho do usuário mais rápido e fácil.

As dificuldades encontradas se deram devido ao curso não oferecer disciplinas sobre programação em Java, então todo o projeto começou a ser desenvolvido do básico da programação, para em seguida evoluir. Além disso, o módulo Bluetooth utilizado apresentava instabilidade durante a comunicação com o celular, então inicialmente foi testado o protocolo Modbus utilizando o Arduino, pois o resultado é o mesmo, logo se tornou mais fácil de analisar.

Para trabalhos futuros, está proposto desenvolver um protótipo do projeto e buscar a possível instalação em alguma indústria. Além disso, para trabalhos futuros pretende-se incluir registros de valores armazenados, envio de informações remotas e acréscimo de comentários à evolução de variáveis analógicas.

BIBLIOGRAFIA

- BILLO, E. A. Uma pilha de protocolos Bluetooth adaptável à aplicação. Trabalho de conclusão de curso de graduação. Universidade Federal de Santa Catarina. Paraná, 2003.
- BOARETTO, NEURY. Sistemas Supervisórios. Instituto Federal Santa Catarina, Joinville, 2008.
- CARNEIRO, S.A. Supervisórios Intouch. Centro Federal de Educação Tecnológica do ES. Serra, 2005.
- JURIZATO, L. A.; PEREIRA, P. S. R. Sistemas supervisórios. Nova Odessa, *Network Technologies*, v. 1, p. 2, 2003.
- LECHETA, R. R. Google Android-3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. Novatec Editora, 2013.
- LIN, K. C.; CHANG-CHIEN, L.R., "A Remote Supervisory Control for Motor Driving System Using Windows Mobile-based Pocket PC," *Industrial Electronics and Applications, 2006 IST IEEE Conference on* , vol., no., pp.1,6, 24-26 May 2006.
- MACEDO, M. M. Gestão da produtividade nas empresas. Revista Organização Sistêmica 1.1 (2012): 110-119.
- MACHADO, E. M. Framework para criação de sistemas supervisórios dinâmicos em dispositivos móveis. (Programa de Pós-Graduação em Computação) Porto Alegre, 2012. 72 f.:Il.
- MAIA, W. U. Sistema Integrado de Operação e Diagnóstico de Falhas para Sistemas de Energia Elétrica: SODF. 1998. Disponível em: <<http://uvirtual.eps.ufsc.br/disserta98/maia>>. Acesso em: 02 mai. 2016.
- MAITELLI, André L. Controladores Logicos Programáveis; disponível em <<http://www.dca.ufrn.br/~maitelli/FTP/clp/>> Acesso em: 02 mai. 2016.
- MEDEIROS, A. A. D, SOUZA, R. B., NASCIMENTO, J. M. A., MAITELLI, A. L., GOMES, H. P. Sisal, um sistema supervisório para elevação artificial de petróleo. *Rio Oil & Gas Expo and Conference*. 2006.
- MEIRELLES, L.; TAROUÇO, L.; ALVES, C. Telemática aplicada à aprendizagem com mobilidade. Revista Novas Tecnologias na Educação. Porto Alegre, 2004.
- MOBILE TIME. Uso dos aplicativos móveis cresceu 58% em 2015. Disponível em <<http://www.mobiletime.com.br/05/01/2016/pesquisa-uso-dos-aplicativos-moveis-cresceu-58-em-2015/425202/news.aspx?noticiario=TT>> Acesso em: 03 mai. 2016.
- NATIONAL INSTRUMENTS, 2013. Uma rápida comparação das interfaces de comunicação serial RS-232, RS-422 e RS-485. Disponível em <<http://digital.ni.com/public.nsf/allkb/DE153F74C4BF3AD8862576AB006C1AAF>> Acesso em: 08 jun. 2016.
- NATIONAL INSTRUMENTS, 2016. Ambiente gráfico de desenvolvimento de sistemas LabVIEW. Disponível em <<http://www.ni.com/labview/pt/>> Acesso em: 08 mai. 2016.
- NOVELETTO, F.; LEAL, A. B.; FILHO, P. B. Modelagem e controle de um sistema automatizado para prevenção de úlceras por pressão. São Caetano do Sul, 2014.
- PEROZZO, R.F.; PEREIRA, C.E. *Framework for Building Supervisory Systems in Mobile Devices. Emerging Technologies and Factory Automation*, 2006. ETFA '06. IEEE Conference on, vol., no., pp.167,172, 20-22. Setembro, 2006.

- PROFICY HMI/SCADA – CIMPPLICITY. GE *Automation*. Disponível em <<http://www.geautomation.com/products/proficy-hmi-scada-cimplicity/n2819>> Acesso em: 08 mai. 2016.
- POCKET GENESIS. Disponível em <<https://getconnected.iconics.com/Marketing/Brochures/Pocket-GENESIS.aspx>> Acesso em: 10 mai. 2016.
- RINCO, M. A. G. Protótipo de Comunicação Bluetooth IEEE 802.15.1 em Tecnologia Móvel. Centro Universitário LaSalle, UNILASALLE, 2005.
- ROCKWELL AUTOMATION. Disponível em <<http://www.rockwellautomation.com/rockwellsoftware/products/factorytalk-view-me.page#latest-releases>> Acesso em: 25 abr. 2016.
- SA AUTOMAÇÃO. Soluções em automação industrial. Disponível em <http://www.sa.online.pt/produtos/produtos_show.htm?idf=17&idp=94> Acesso em: 22 abr. 2016.
- SACRAMENTO, P. N. C. Concepção/Desenvolvimento de aplicações de domótica. (2015).
- SCADA. Elipse Software. Disponível em <<http://www.elipse.com.br/port/scada.aspx>> Acesso em: 02 mai. 2016
- SCADABR. Disponível em <<http://www.scadabr.com.br/>> Acesso em: 27 abr. 2016.
- SCHNEIDER ELECTRIC. Wonderware InTouch. Disponível em <<https://www.wonderware.com/hmi-scada/intouch/>> Acesso em: 01 mai. 2016.
- SIEMENS. Disponível em <<http://www.automation.siemens.com>> Acesso em: 18 abr. 2016.
- SIQUEIRA, T.S. Bluetooth – Características, protocolos e funcionamento. Universidade Estadual de Campinas. Campinas, 2006.
- WEG. Manual do Micro Controlador Lógico Programável Série: TPW-03, 2010. 136p.

APÊNDICE A

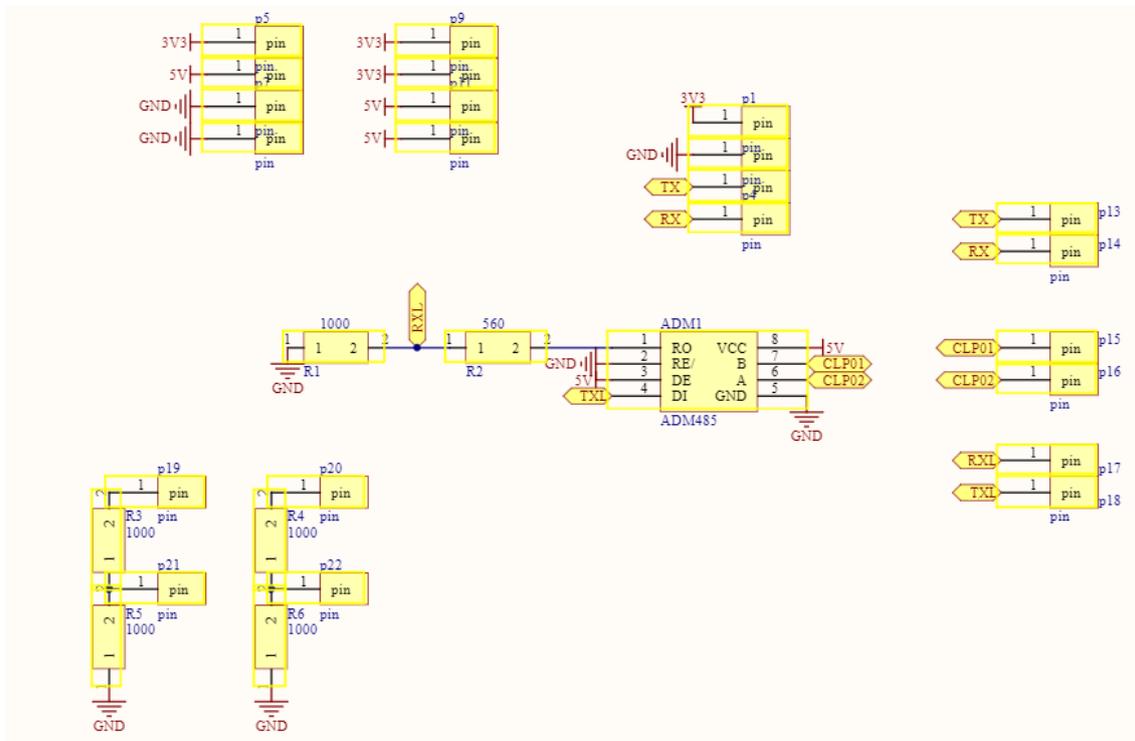


Figura 1: Esquemático da PCI.

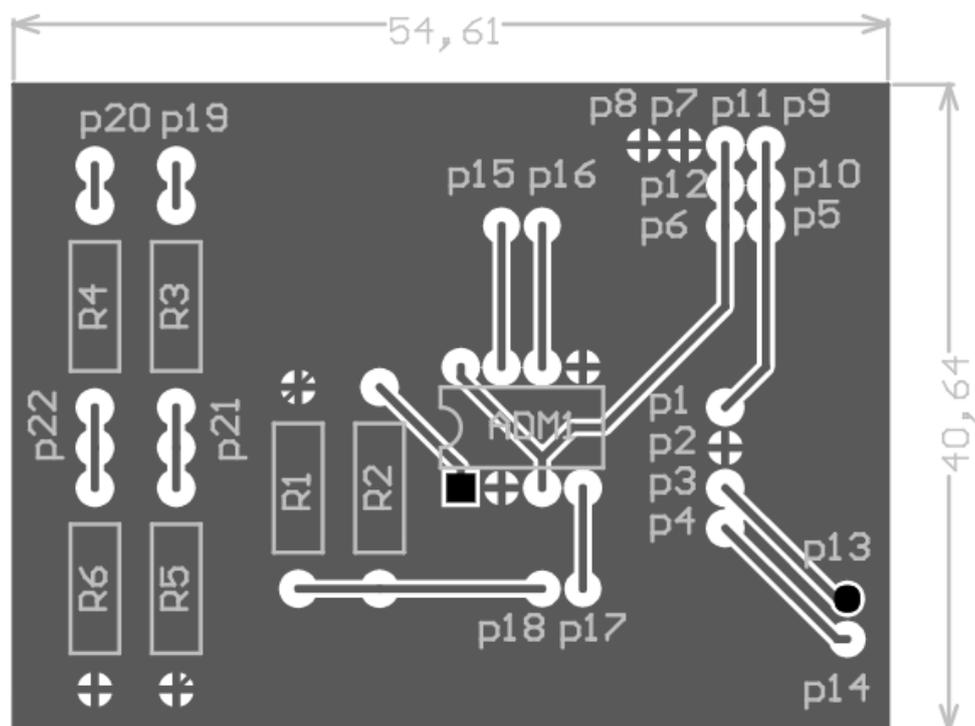


Figura 2: Layout da PCI.