

INSTITUTO FEDERAL DE SANTA CATARINA

BRUNO DE BORBA

SERVIÇO DE DESCOBERTA PARA O PROTOCOLO MQTT EM UM SISTEMA DE
MONITORAMENTO DE GRUPOS MOTOR-GERADOR BASEADO EM INTERNET
DAS COISAS

FLORIANÓPOLIS – SC

JUNHO/2018

BRUNO DE BORBA

SERVIÇO DE DESCOBERTA PARA O PROTOCOLO MQTT EM UM SISTEMA DE
MONITORAMENTO DE GRUPOS MOTOR-GERADOR BASEADO EM INTERNET
DAS COISAS

Dissertação submetida ao Programa de Pós-Graduação em Mecatrônica do Instituto Federal de Santa Catarina como parte dos requisitos para a obtenção do título de Mestre em Mecatrônica.

Orientador: Roberto Alexandre Dias

FLORIANÓPOLIS – SC

JUNHO/2018

CDD 629.8
B726s

Borba, Bruno de
Serviço de descoberta para o protocolo MQTT em um sistema de monitoramento de grupos motor-gerador baseado em internet das coisas [DIS] / Bruno de Borba; orientação de Roberto Alexandre Dias – Florianópolis, 2018.

1 v.: il.

Dissertação de Mestrado (Mecatrônica) – Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Inclui referências.

1. Internet das coisas. 2. Serviço de descoberta MQTT. 3. Grupo Motor- Gerador. 4. Monitoramento. I. Dias, Roberto Alexandre. II. Título.

Sistema de Bibliotecas Integradas do IFSC
Biblioteca Dr. Hercílio Luz – Campus Florianópolis
Catalogado por: Edinei Antonio Moreno CRB 14/1065

SERVIÇO DE DESCOBERTA PARA O PROTOCOLO MQTT EM UM SISTEMA DE
MONITORAMENTO DE GRUPOS MOTOR-GERADOR BASEADO EM INTERNET
DAS COISAS

BRUNO DE BORBA

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Mecatrônica e
aprovada em sua forma final pelo Orientador e pela Banca Examinadora

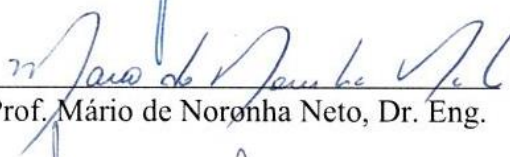
Florianópolis, 08 de Junho de 2018



Prof. Roberto Alexandre Dias, Dr. Eng.



Prof. Valdir Noll, Dr. Eng.



Prof. Mário de Noronha Neto, Dr. Eng.



Prof. Mauricio Edgar Stivanello, Dr. Eng.

RESUMO

Sistemas de monitoramento de condição projetados para grandes plantas de energia são muito caros para serem aplicados em ativos de menor valor, como é o caso dos GMGs (Grupos Motor-Gerador). Para viabilizar o monitoramento on-line destes ativos de menor custo é preciso que o hardware possua um valor reduzido, o que implica em limitações de capacidade de processamento e comunicação. O conceito de Internet das Coisas (IoT - *Internet of Things*) e o uso de protocolos de comunicação como o MQTT (*Message Queuing Telemetry Transport*) são ideais para lidar com tais limitações, sendo uma alternativa coerente e que torna viável economicamente o monitoramento em ativos de geração de menor custo.

Este trabalho propõe um Serviço de Descoberta para o protocolo MQTT e a utilização deste serviço na disponibilização dos dados coletados em um sistema de monitoramento de condição baseado em Internet das Coisas para Grupos Motor-Gerador usando o protocolo Modbus RTU (*Remote Terminal Unit*) sobre um barramento serial RS-485. Um protótipo do sistema proposto foi construído para testes, validações e comparações com sistemas existentes no mercado.

O Serviço de Descoberta foi desenvolvido sobre o *payload* (carga útil) do protocolo MQTT de forma a não modificar a especificação do protocolo, permitindo o uso do serviço com aplicativos MQTT já existentes. Foi possível, por exemplo, montar representações gráficas de forma automática através dos dados disponibilizados pelo serviço proposto.

Os resultados são de grande valor para a AQTech, para o setor elétrico, para os donos/usuários de grupos motor-gerador e para a sociedade. Como resultado econômico destaca-se o benefício para a AQTech, possibilitando a aplicação dos conceitos desenvolvidos em soluções de monitoramento, e benefício para o setor elétrico e para os donos/usuários de grupos motor-gerador, pois visa aumentar a disponibilidade e a vida útil dos ativos de geração. No contexto social contribui para um melhor aproveitamento de recursos, para redução da indisponibilidade da energia elétrica e para a geração distribuída em horário de ponta.

Palavras-chave: Internet das Coisas. Serviço de Descoberta. MQTT. Grupo Motor-Gerador. Monitoramento.

ABSTRACT

Condition monitoring systems designed for large power plants are too expensive to apply to small energy assets like generator sets (gensets). To enable the online monitoring of these smaller assets low cost hardware is needed, which implies limitations in processing and communication capacity. The concept of IoT (*Internet of Things*) and the use of communication protocols like MQTT (*Message Queuing Telemetry Transport*) are ideal to deal with those limitations, being a coherent alternative and economically viable to monitor low cost generation assets.

This work proposes a MQTT protocol Discovery Service and its use for the provision of data collected in an IoT-based condition monitoring system for generator sets using Modbus RTU protocol over RS-485 serial bus. A prototype of the proposed system was built for testing, demonstration of the system and benchmarking.

The discovery service was developed over the payload of the MQTT protocol, so the protocol specification was not modified, allowing the use of the service with existing MQTT applications. It was possible, for example, to build graphical components automatically using the data provided by the proposed service.

The results of this project are valuable to AQTech company, to electric sector, to owners/users of gensets and to the society. As an economic result, the benefit to AQTech is highlighted, enabling the application of the concepts developed in industrial monitoring solutions, and benefit to the electric sector and owners/users of gensets, as it aims to increase the availability and lifespan of generation assets. In the social context it contributes for a better use of resources, to reduce the electricity unavailability and to distributed generation during peak hours.

Key-words: Internet of Things. Discovery Services. MQTT. Generator set. Monitoring.

LISTA DE FIGURAS

Figura 1 - Diagrama de classes parcial do software Pulsar.....	23
Figura 2 – Supervisório WEB de um projeto piloto.	26
Figura 3 – Modelo de gerenciamento de grupos motor-gerador com Serviço de Descoberta sobre MQTT	31
Figura 4 – Exemplo de componente gráfico para visualizar o sinal de teste.	34
Figura 5 – Kit de desenvolvimento Wemos D1 mini.	36
Figura 6 – Página WEB de configuração do protótipo.	38
Figura 7 – Mensagens recebidas pelo MQTT Cloud.	39
Figura 8 – Dashboard gerado no aplicativo MQTT Dash.	40
Figura 9 – Componente gráfico montado automaticamente em um aplicativo de testes.....	41
Figura 10 – Protótipo com ESP12E, fonte DC regulada e Arduino.	42
Figura 11 – Vista superior de protótipo com ESP07, módulo GPRS e RTC.	42
Figura 12 – Vista inferior de protótipo com ESP07, protótipo com ESP12E, fonte DC regulada e Arduino.	43
Figura 13 – Bancada com protótipos e sistema CloudMQTT sendo exibido em um monitor.	43

LISTA DE TABELAS

Tabela 1 – Definição dos tópicos de configuração do Serviço de Descoberta.....	32
Tabela 2 – Exemplo de tópicos de configuração.	33
Tabela 3 – Comparação entre categorias de soluções similares e o protótipo desenvolvido...	46

LISTA DE ABREVIATURAS E SIGLAS

GMG – Grupo Motor-Gerador

IaaS – Infrastruture as a Service

IoT – Internet of Things

MQTT – Message Queuing Telemetry Transport

PaaS – Platform as a Service

PCHs – Pequenas Centrais Hidrelétricas

UHEs – Usinas Hidrelétricas de Energia

RTU – Remote Terminal Unit

SaaS – Software as a Service

M2M – Machine to Machine

OPC-UA – Open Platform Communications - Unified Architecture

DPWS – Devices Profile for Web Services

CLP – Controladores Lógico Programáveis

PSV – Platform Supply Vessel

OSI – Open System Interconnection

ISO – International Organization for Standardization

IEC – International Electrotechnical Commission

PRF – Proof of a new International Standard

IEEE – Institute of Electrical and Electronics Engineers

HTTP – Hypertext Transfer Protocol

AWS – Amazon Web Services

UPnP – Universal Plug and Play

POSIX – Portable Operating System Interface

ANSI C – American National Standards Institute for the C programming language

SUMÁRIO

1. INTRODUÇÃO	9
1.1 Objetivos	10
1.1.1 Objetivo Geral	10
1.1.2 Objetivos Específicos.....	10
1.2 Justificativa e Relevância	11
2. REVISÃO BIBLIOGRÁFICA	13
2.1 Estado da arte	13
2.2 Grupos Motor-Gerador	14
2.3 Sistemas de Supervisão, Controle e Aquisição de Dados	15
2.4 Internet das Coisas	16
2.5 Serviços de Descoberta	18
2.6 Computação em Nuvem.....	18
2.7 Paradigmas de Manutenção.....	19
2.8 Software Pulsar.....	21
3. DEFINIÇÃO DO PROBLEMA	23
4. SOLUÇÃO PROPOSTA.....	28
4.1 Arquitetura de monitoramento	29
4.2 Definição do Serviço de Descoberta para MQTT	31
5. RESULTADOS.....	36
5.1 Validações em laboratório.....	36
5.2 Benchmarking.....	44
6. CONCLUSÕES	48
ANEXO A – ESQUEMÁTICO DO PROTÓTIPO	53
ANEXO B – LAYOUT DO PROTÓTIPO	54
ANEXO C – CÓDIGO FONTE DO PROTÓTIPO.....	55

1. INTRODUÇÃO

O conceito de Internet das Coisas consiste em poder conectar uma variedade de objetos entre si [1]. A Internet das Coisas é um conceito de tornar a Internet e a comunicação entre objetos ou “coisas” pervasiva, onde eles são capazes de interagir e de cooperar entre si para alcançar um objetivo [2]. Para tornar a IoT viável e permitir a conexão de centenas de milhares de coisas, é necessário que tais coisas sejam de baixo custo, o que implica baixa capacidade de processamento, armazenamento e comunicação [3].

Para permitir a comunicação entre diversos dispositivos é preciso também estabelecer um ou mais protocolos de comunicação entre eles. Como esses dispositivos são de capacidade limitada, esses protocolos devem ser adequados para lidar com baixa largura de banda, redes de alta latência e instabilidade de comunicação.

Essas características de dispositivos de baixo custo e redes de comunicação limitadas se adequam às necessidades do mercado em monitoramento de GMGs, uma vez que o custo é um fator crítico que pode inviabilizar o monitoramento e normalmente a comunicação ocorre em redes celulares ou Wi-Fi.

O conceito de IoT está em crescimento acelerado. O número de “coisas” conectadas deve atingir 212 bilhões de entidades implantadas a nível mundial até o final de 2020 [4]. Em 2022, espera-se que o tráfego de comunicação M2M (*Machine to Machine*) deva constituir até 45% de todo o tráfego da Internet [4] [5] [6]. O instituto McKinsey reportou que o número de máquinas conectadas cresceu 300% nos últimos 5 anos [7]. O monitoramento de tráfego em uma rede celular nos Estados Unidos mostrou um crescimento de 250% de tráfego de dados M2M em 2011 [8]. Ainda, o impacto anual total causado pelo IoT é estimado entre 2.7 trilhões e 6.2 trilhões de dólares em 2025 [7].

Todos esses números e projeções apontam para um crescimento potencialmente significativo e de ritmo rápido da Internet das Coisas no futuro próximo. Este crescimento proporciona uma oportunidade única para fabricantes de equipamentos tradicionais transformarem seus produtos em “coisas inteligentes” [9].

Muitos dos sistemas tradicionais utilizam o protocolo Modbus para permitir o monitoramento e controle dos processos. Na área de GMGs, este protocolo está praticamente consolidado (*de facto Standard*) nos CLPs (Controladores Lógico Programáveis). Existem

diversos modelos de CLPs no mercado, cada um com suas particularidades e com diferentes mapas Modbus.

Devido a essa variedade de modelos de CLPs de diferentes fabricantes, os sistemas de monitoramento compatíveis devem possibilitar a leitura de diferentes sinais com diferentes características, como número de algarismos significativos, unidades de engenharia, etc. Além disso, o servidor que recebe os dados monitorados dos diferentes dispositivos deve também lidar com essas diferenças no processamento dos dados.

A importância dos Serviços de Descoberta surge exatamente neste contexto. O objetivo principal de um Serviço de Descoberta é reduzir os esforços de gerenciamento por parte do usuário, automatizando a troca de informações entre os dispositivos e reduzindo ao máximo a necessidade de intervenção humana.

Diversos protocolos já fazem uso de Serviços de Descoberta, como por exemplo o OPC-UA (*Open Platform Communications - Unified Architecture*) e o DPWS (*Devices Profile for Web Services*). O protocolo MQTT, por outro lado, ainda não possui um Serviço de Descoberta em sua especificação, porém possui diversas outras características que o tornam adequado para uso em dispositivos de IoT, limitados em banda e capacidade de processamento.

A seguir, são apresentados os objetivos deste trabalho, sua justificativa, relevância e metodologia utilizada.

1.1 Objetivos

1.1.1 Objetivo Geral

- O objetivo principal deste trabalho é propor um Serviço de Descoberta para o protocolo MQTT sem alterar a especificação do protocolo e utilizar este serviço na disponibilização dos dados coletados em um sistema de monitoramento baseado em Internet das Coisas para Grupos Motor-Gerador usando o protocolo Modbus RTU sobre um barramento serial RS-485.

1.1.2 Objetivos Específicos

- Estudar implementações de Serviço de Descoberta para a concepção da proposta de implementação sobre o MQTT;
- Desenvolver uma abordagem de Serviço de Descoberta compatível com a especificação atual do MQTT e com as plataformas de implementação existentes;

- Avaliar a funcionalidade do Serviço de Descoberta proposto dentro do cenário de gerenciamento de pequenos grupos moto-gerador;
- Desenvolver um sistema protótipo no modelo proposto para validação dos conceitos e da viabilidade do sistema em termos de custo.

1.2 Justificativa e Relevância

A energia elétrica é um recurso essencial para a sociedade e para o desenvolvimento do país. A geração de energia elétrica é atividade que demanda grandes investimentos. Para garantir o retorno de investimentos é preciso estender ao máximo a disponibilidade e vida útil dos ativos, o que torna a gestão de ativos um fator crítico de sucesso dos empreendimentos. Uma das principais ferramentas da engenharia de manutenção é a monitoração online das unidades geradoras, assunto que tem sido alvo de inúmeros estudos acadêmicos e corporativos nas últimas décadas.

A utilidade prática do projeto está na aplicação da solução de monitoramento proposta, através de um sistema protótipo, para o monitoramento online de grupos motor-gerador através das variáveis disponibilizadas em seus controladores lógico programáveis. O monitoramento destes ativos de geração deve possuir um custo muito reduzido frente a soluções de monitoramento mais complexas, comumente aplicadas em ativos de geração de maior porte.

Os resultados deste projeto são de grande valor para a empresa AQTech, para o setor elétrico, para os donos/usuários de grupos motor-gerador e para a sociedade. Como resultado econômico destaca-se o benefício para a empresa AQTech, possibilitando a aplicação dos conceitos desenvolvidos em soluções de monitoramento industriais, e benefício para o setor elétrico e para os donos/usuários de grupos motor-gerador, pois visa aumentar a disponibilidade e a vida útil dos ativos de geração. No contexto social contribui para um melhor aproveitamento de recursos, para redução da indisponibilidade da energia elétrica e para a geração distribuída em horário de ponta.

A empresa AQTech possui know-how em monitoramento de geradores de energia elétrica, atuando no mercado nacional e internacional, principalmente com grandes empresas do setor de geração de energia hidrelétrica. Nos últimos anos a AQTech tem expandido seu mercado de plantas de geração centralizadas de PCHs (Pequenas Centrais Hidrelétricas) e UHEs (Usinas Hidrelétricas de Energia) para abranger também a geração distribuída de menor porte, como é o caso dos grupos motor-gerador por combustão, geradores eólicos e placas solares.

A proposta de trabalho se acopla às necessidades de mercado identificadas na empresa e representa uma solução muito atrativa para os clientes que buscam maximizar a disponibilidade e vida útil de seus ativos de geração. O tema proposto está diretamente alinhado a evolução tecnológica da empresa AQTech.

Eventos como o apagão de 2001 e a crise energética iniciada em 2012 impulsionaram no setor elétrico o uso de grupos motor-gerador como fonte de energia independente das concessionárias [11]. As estimativas indicam aproximadamente 200 mil grupos geradores no Brasil no ano de 2015 [12], tornando o mercado atrativo em termos de volume quando comparado com o setor elétrico onde há poucos geradores de grande porte.

Para viabilizar economicamente o desenvolvimento deste sistema (cuja principal característica é o baixo custo do hardware) é preciso considerá-lo em alta escala. Em um sistema concebido para aplicação em alta escala, o conceito de Serviços de Descoberta é extremamente desejável.

Esse conceito, como visto anteriormente, visa reduzir os esforços necessários para o gerenciamento das aplicações no lado do servidor, uma vez que o próprio dispositivo fornece os dados necessários para o funcionamento do sistema. Este conceito é aplicado, por exemplo, em redes de computador que configuram de forma automática uma rede (zeroconf), permitindo que os usuários conectem uma variedade de dispositivos, como computadores, impressoras e smartphones sem precisar configurá-los.

Em um sistema que deve lidar com uma variedade de modelos de CLPs e que deve, por exemplo, progressivamente suportar mais modelos de CLPs, o conceito de zeroconf é de extrema importância, pois reduz os esforços de gerenciamento, conseqüentemente reduzindo os custos para a empresa provedora do serviço e aumentando a agilidade na integração de novos geradores no sistema de gerenciamento.

2. REVISÃO BIBLIOGRÁFICA

2.1 Estado da arte

Foi identificada uma solicitação de implementação de um Serviço de Descoberta de servidores MQTT-SN (MQTT for Sensor Network) através de pacotes UDP multicast ou técnica similar [29]. O protocolo MQTT-SN difere em algumas características do protocolo MQTT para se adequar a dispositivos embarcados ainda mais limitados e é tipicamente utilizado em dispositivos que não implementam a pilha TCP/IP, por exemplo, em barramento serial.

O propósito deste trabalho difere desta solicitação, uma vez que não objetiva a identificação de servidores MQTT acessíveis em uma rede de computador, mas busca identificar os dados disponíveis em um Broker MQTT e as configurações associadas a cada sinal monitorado.

Ainda, o protocolo MQTT-SN objetiva a redução de uso de banda e memória, por exemplo, utilizando um identificador ao invés de transmitir o tópico em cada publicação. O protocolo permite ainda que o registro do tópico seja pré configurado diretamente no gateway. Dessa forma, o tópico não precisa ser transmitido nem mesmo na primeira transmissão pelo dispositivo.

Tais limitações não se adequam ao trabalho proposto, cuja consequência é ampliar o uso de banda e memória, transferindo além dos dados de monitoramento em si, configurações do respectivo sinal monitorado, afim de automatizar configurações e reduzir esforços de gerenciamento.

Uma proposta que se assemelha mais ao trabalho proposto foi identificada e se trata de uma plataforma de automação residencial de código aberto [30], porém esta proposta suporta apenas alguns tipos de sensores e não padroniza a publicação de configurações de sinais, como tipo de dado, unidade de engenharia, etc., o que enfraquece a interoperabilidade da solução.

O objetivo é muito próximo ao deste trabalho, que é o de minimizar os esforços de configuração, porém as configurações dos sinais possuem um grande foco na forma com que o sinal é exibido (ventilador, luz, chave) e não nas características dos sinais monitorados em si, como unidade de engenharia, número de Algarismos significativos, etc.

Este sistema, da mesma forma que a solução proposta neste trabalho, possui tópicos de configuração que são obrigatórios e tópicos de configuração que são opcionais. O sistema

utiliza também o *payload* do protocolo MQTT para trafegar as configurações dos sinais, portanto sem alterar o protocolo MQTT e permitindo o uso de *Brokers* já disponíveis na internet. Essa mesma proposta é utilizada no protocolo proposto nesta dissertação.

Este sistema, suporta as seguintes categorias de dados:

- Sensores binários;
- Câmeras;
- Aberturas (portas, janelas, etc.);
- Ventiladores;
- Luzes;
- Sensores;
- Chaves;

2.2 Grupos Motor-Gerador

Os Grupos Motor-Gerador (GMGs) são compostos de um motor à combustão, normalmente diesel, acoplado a um alternador [10]. Eventos como o apagão de 2001 e a crise energética iniciada em 2012 impulsionaram no setor o uso de grupos motor-gerador (GMG) como fonte de energia independente.

Estes grupos geradores são usados nos mais diversos segmentos de mercado, como por exemplo, no setor de petróleo, na indústria em geral, nos centros urbanos e rurais e também no próprio setor elétrico. A operação dos grupos geradores visa principalmente o horário de ponta, a energia de emergência (sistemas auxiliares das usinas, por ex.) e também a geração contínua em plantas de autoprodução de energia com ou sem comercialização de excedentes.

A empresa WEG, por exemplo, fornece geradores para embarcações PSV (*Platform Supply Vessel*) que transportam equipamentos para plataformas da PETROBRÁS. Em contato com a WEG foi indicado que a estimativa é que a quantidade de navios PSV aumente significativamente nos próximos anos, portanto, aumentando também o fornecimento de grupos geradores. Algumas companhias de energia estão também voltando suas atenções no segmento de geração distribuída com grupos geradores à biogás conectados na rede de distribuição em baixa tensão.

A grande maioria desses grupos motor-gerador que operam com um CLP (controlador lógico programável) suportam o protocolo Modbus RTU. O Modbus é um protocolo de mensagem de camada de aplicação criado em 1979 pela Modicon, posicionado no nível 7 do

modelo OSI (*Open System Interconnection*), e fornece comunicação cliente/servidor entre dispositivos conectados em diferentes tipos de barramentos ou redes [14].

O protocolo Modbus é um protocolo de requisição/resposta em que cada serviço é especificado por um código de função. Por ser livre de taxas de licenciamento e se adequar a diversos meios físicos (como RS-232, RS-485 e Ethernet), o Modbus é utilizado em milhares de equipamentos e é uma solução de comunicação muito barata de se utilizar.

2.3 Sistemas de Supervisão, Controle e Aquisição de Dados

Sistemas de supervisão, controle e aquisição de dados (SCADA - *Supervisory Control and Data Acquisition*) são sistemas utilizados para supervisionar e controlar outros sistemas, embora muitas aplicações destes sistemas sejam totalmente passivas (apenas monitoram outros sistemas, sem atuar nos mesmos). Sistemas neste conceito tem sido utilizado ao longo de décadas para monitoramento e controle de processos industriais.

Uma das principais características de sistemas SCADA é a possibilidade de centralizar informações de diversos processos de uma instalação, possibilitando uma visão ampla e centralizada dos processos envolvidos.

Estes são alguns potenciais benefícios em adotar um sistema SCADA proprietário, amplamente utilizado e bem aceito na indústria:

- Um sistema rico em funcionalidades e extensivamente testado.
- Suporte a diversos protocolos industriais padronizados.
- O uso de uma ferramenta SCADA normalmente reduz em grande parte o esforço que seria necessário para desenvolver uma interface de supervisão e controle própria. Este esforço ainda pode ser terceirizado.
- Confiabilidade e robustez. Sistemas SCADA são amplamente utilizados na indústria há décadas.
- Suporte técnico e manutenção através de outras empresas, o que reduz os esforços internos para manter o sistema.

Existem ainda soluções SCADA de software livre, porém a utilização desses sistemas é limitada na indústria, pois pode reduzir alguns dos benefícios de um sistema proprietário como confiabilidade, robustez e suporte técnico.

Alguns recursos de software costumam ser comuns nos sistemas SCADA:

- Interface de supervisão personalizada com diferentes tipos de mostradores gráficos;
- Alarmes;
- Histórico dos dados monitorados em base de dados.
- Suporte a protocolos industriais padronizados via rede ou interface serial.

2.4 Internet das Coisas

O principal conceito relacionado a Internet das Coisas é tornar a comunicação entre objetos (coisas) pervasiva, capazes de interagir e de cooperar entre si [1] [2].

A arquitetura mais simples para a Internet das Coisas consiste em um dispositivo conectado a algum serviço na nuvem (seção 2.5) através de uma rede Wifi ou GPRS. Este dispositivo pode realizar a leitura de variáveis através de algum transdutor ou protocolo de comunicação, podendo inclusive atuar sobre o sistema se necessário.

Existem diversas alternativas de hardware adequados para uso com Internet das Coisas. Recentemente surgiram diversas alternativas de kits de desenvolvimento que utilizam o SoC (*System on Chip*) ESP8266 [25]. Trata-se de um sistema barato, de baixo consumo de energia, compacto, de bom desempenho de processamento e com Wifi integrado.

Existem sistemas mais limitados que também podem ser utilizados para esta finalidade, como kits Arduino com recursos de comunicação via Ethernet ou Wifi, e sistemas mais complexos e caros, como a BeagleBone ou a Raspberry Pi. O SoC ESP8266 possui capacidade de processamento intermediária entre esses tipos de soluções e é comercializado por um valor muito baixo, além de ser compatível com a plataforma de desenvolvimento Arduino. Essas características contribuíram significativamente para a popularização da plataforma.

Para a comunicação com a internet, diversos protocolos atendem aos requisitos de baixa capacidade de processamento, redes de alta latência e baixa largura de banda: CoAP (*Constrained Application Protocol*), MQTT (*Message Queuing Telemetry Transport*), WAMP (*Web Application Messaging Protocol*), XMPP (*Extensible Messaging and Presence Protocol*) e SNMP (*Simple Network Management Protocol*), dentre outros [3].

O CoAP é um protocolo da sétima camada do modelo OSI (protocolo de aplicação) com intuito de ser utilizado em dispositivos simples. A especificação do protocolo é a RFC 7252. O CoAP foi projetado para facilmente se integrar na WEB enquanto atende a requisitos como suporte a *multicast* e baixo *overhead*.

WAMP é um protocolo WebSocket aberto que suporta tanto o modo “*publish-subscribe*” quanto o modo RPC (*Remote Procedure calls*). Estão sendo realizados trabalhos para tornar o WAMP um protocolo padrão IETF (*Internet Engineering Task Force*) [17].

XMPP é um protocolo aberto baseado em XML. Seu objetivo principal é para troca de mensagens, mas com o crescimento do conceito de IoT a “XMPP Standards Foundation” está trabalhando em extensões para uso com IoT [18].

SNMP é um protocolo de aplicação de gerenciamento especificado pelo RFC 1157. Este protocolo é largamente utilizado no gerenciamento e monitoramento de redes.

O MQTT foi criado em 1999 pela IBM e é um protocolo de comunicação também em nível de aplicação, otimizado para rodar em redes TCP/IP de baixa confiabilidade, baixa banda e alta latência. O protocolo foi desenvolvido para necessitar de poucos recursos de hardware e para rodar em dispositivos embarcados mais simples.

O protocolo MQTT foi padronizado pelo OASIS em 2014 e atualmente está na versão 3.1.1, sendo livre de royalties desde 2010 [15].

Um sistema microprocessado pode obter informações a partir de comunicação serial SPI/UART/etc. ou diretamente de sensores e publicar as informações via protocolo MQTT em um *Broker*. O *Broker* é um dispositivo responsável por distribuir as mensagens aos clientes interessados e a partir daí os dados podem ser coletados por diversos dispositivos, uma vez que o MQTT é um protocolo baseado em “*publish-subscribe*”.

O protocolo MQTT é padronizado pela ISO/IEC PRF 20922 e considerado um ótimo protocolo para dispositivos de IoT. Um exemplo fora de contexto que mostra a aceitação do protocolo MQTT é o fato de a empresa Facebook ter apostado neste protocolo no software Facebook Messenger, que utiliza este protocolo em parte do código fonte [13]. No contexto de IoT, a Amazon adotou MQTT, HTTP e Websockets como protocolos padrões em sua plataforma AWS (Amazon Web Services) IoT [3].

Já no quesito de segurança, até a versão 3.1.1 o MQTT não implementa qualquer tipo de criptografia (SSL pode ser utilizado independentemente), existindo apenas um método de autenticação com nome de usuário e senha [3].

Como alternativa para o quesito segurança, estão surgindo dispositivos de IoT com suporte a criptografia acelerada por hardware, como é o caso do ESP32 que suporta AES, SHA2, RSA-4096 e outros [19], o que permite o uso de criptografia sem penalizar tanto a CPU.

Devido às diversas características apresentadas, principalmente quanto a adoção do protocolo por grandes empresas em grandes plataformas de software (o que pode indicar um comportamento de início de padronização no mercado), o protocolo MQTT foi selecionado como alvo para a realização de um estudo mais detalhado nesta pesquisa, integrando a solução proposta na arquitetura de monitoramento.

2.5 Serviços de Descoberta

Serviços de Descoberta são sistemas que detectam dispositivos e os serviços oferecidos por esses dispositivos de forma automática em uma rede de computadores, permitindo a descoberta, configuração e comunicação entre eles de forma autônoma, minimizando os esforços de gerenciamento. Alguns exemplos de Serviços de Descoberta são o WS-Discovery e o UPnP (*Universal Plug and Play*).

O WS-Discovery é um protocolo baseado em multicast. Seu modo de funcionamento consiste em um cliente que busca por servidores que suportem determinado serviço. O cliente envia uma mensagem a um grupo de servidores via multicast e os servidores que correspondem à mensagem enviada respondem diretamente ao cliente.

Quando um servidor ingressa em um grupo de servidores, ele envia a todos os dispositivos uma mensagem de anúncio, a fim de reduzir a necessidade por varreduras periódicas.

Já o UPnP é um conjunto de protocolos que permitem que uma variedade de dispositivos (como smartphones, computadores e impressoras), se detectem, se configurem, e troquem informações entre si em uma rede automaticamente. O UPnP roda sobre o protocolo IP e é direcionado principalmente para uso residencial.

2.6 Computação em Nuvem

Segundo a empresa Amazon, “Para simplificar, a computação em nuvem é o fornecimento de serviços de computação – servidores, armazenamento, bancos de dados, rede, software, análise e muito mais – pela Internet (a nuvem)” [31].

Diversos sistemas fazem uso da computação em nuvem, seja para enviar e receber e-mails, ouvir música em serviços de streaming ou assistir a vídeos em diversas plataformas. Esse conceito é amplamente utilizado em todas as plataformas de hardware, como computadores, smartphones ou tablets.

São inúmeras as vantagens da computação em nuvem, como o custo reduzido para iniciar o negócio, visto que não é necessária a compra de hardware e software para a criação de datacenter locais e nem de toda a infraestrutura necessária para manter um sistema com boa disponibilidade e segurança. Outras vantagens incluem a escalabilidade, visto que para aumentar a capacidade de um serviço contratado normalmente basta acessar a plataforma e efetuar alguns cliques. Recursos como armazenamento, largura de banda, capacidade de processamento e memória são facilmente reconfigurados nos serviços disponíveis comercialmente, eliminando a necessidade de investimentos em infraestrutura e reduzindo drasticamente o tempo necessário para redimensionamento do sistema.

A disponibilidade e a confiabilidade também costumam ser muito mais elevadas em serviços de computação na nuvem, visto que este é o foco do serviço.

A computação em nuvem atualmente é classificada em 3 tipos de serviço: IaaS (*Infrasctruture as a Service*), PaaS (*Platform as a Service*) e SaaS (*Software as a Service*).

Os serviços de computação em nuvem também costumam ser classificados quanto a sua implantação: pública, privada ou híbrida.

Segundo a empresa Amazon, “Nuvens públicas são de propriedade de um provedor de serviços de nuvem de terceiros e operadas por ele, que por sua vez fornece recursos de computação, como servidores e armazenamento pela Internet” ... “Uma nuvem privada é aquela em que os serviços e a infraestrutura são mantidos em uma rede privada” ... “Nuvens híbridas combinam nuvens públicas e privadas ligadas por uma tecnologia que permite que dados e aplicativos sejam compartilhados entre elas” [31].

2.7 Paradigmas de Manutenção

A NBR 5462 considera a manutenção como a “combinação de todas as ações técnicas e administrativas, incluindo as de supervisão, destinadas a manter ou recolocar um item em um estado no qual possa desempenhar uma função requerida” [20].

As filosofias de manutenção tipicamente são divididas em corretiva, preventiva e preditiva. Na primeira, as ações da equipe de manutenção são desencadeadas pelas ocorrências de falhas (estragou-conserte). Na segunda, os ajustes são realizados de maneira periódica, independente do estado da máquina. Já na filosofia preditiva, as ações são baseadas no estado ou condição do ativo. Esta característica torna a manutenção preditiva – também conhecida por

CBM (*Condition Based Maintenance* – Manutenção Baseada em Condição) – a mais meticulosa e ativa das três filosofias de manutenção [21].

A manutenção baseada em condição pode contribuir significativamente para a melhora do desempenho da Plantas de geração de energia.

No Brasil, os resultados da aplicação da monitoração online como ferramenta preditiva de manutenção ainda são tímidos. Estudos identificam diversas lacunas nas abordagens teóricas e práticas utilizadas no país [22]. Estas limitações são intensificadas à medida que o sistema elétrico brasileiro migra para a geração distribuída, porque se adiciona à equação a dispersão dos pontos de geração. Quando os geradores estão instalados em múltiplos locais geograficamente distantes da equipe de manutenção, tanto a monitoração online quanto as próprias atividades de gestão do ativo são dificultadas.

Sistemas de IoT são uma alternativa viável para o monitoramento on-line destes ativos, principalmente pelo foco em baixo custo, uma vez que a geração distribuída implica em diversos pontos de geração de energia de menor porte, e por consequência, de menor investimento.

Como exemplo de uma aplicação da AQTech está uma PCH composta por dois gerados de turbinas do tipo Francis, com energia assegurada na ordem de 7MW. Os geradores apresentavam faltas intermitentes de origem desconhecida. Em algumas situações os geradores perdiam o sincronismo com a rede e precisavam ser desconectadas, interrompendo a geração e causando prejuízos econômicos, fora o desgaste acentuado dos geradores causado pelo estresse na operação.

Foram realizados diversos procedimentos de manutenção nas unidades geradoras por parte da equipe de manutenção, porém não havia informação suficiente para diagnosticar o problema, que persistiu mesmo após essas manutenções.

Foi instalado um equipamento de monitoramento permanente (on-line) da empresa AQTech em um dos geradores, monitorando um pequeno grupo de variáveis: tensões e correntes trifásicas, tensão de campo e deslocamentos radiais do eixo.

Foram identificadas diversas faltas na operação do gerador monitorado:

- Deslocamento excessivo do eixo do gerador durante a partida da máquina, chegando a saturar a leitura realizada através de um sensor de proximidade (proxímetro);

- Regulador de tensão com baixo desempenho;
- Curto circuito trifásico nos terminais do gerador por um determinado instante de tempo;
- Perda de excitação do gerador, medida através da saída do regulador de tensão.

Essas faltas ocorriam de forma intermitente e não eram conhecidas até o emprego do sistema de monitoramento permanente. O sistema permitiu ainda visualizar que a perda de excitação do gerador era seguida pela perda de sincronismo com a rede.

O regulador de tensão sofreu assistência técnica, assim como o transformador de saída do gerador que causava curto circuito devido ao rompimento do dielétrico por problemas no óleo de isolamento.

Os benefícios do sistema de monitoramento foram alcançados rapidamente, motivando a expansão do sistema para as duas máquinas e ampliando o número de variáveis monitoradas.

Tais prejuízos na operação poderiam ter sido evitados ou minimizados caso o sistema de monitoramento já estivesse instalado nas unidades geradores antes dos “sintomas” se manifestarem, interrompendo a geração.

2.8 Software Pulsar

O software Pulsar é um software multiplataforma desenvolvido pela empresa AQTech Engenharia e Instrumentação S.A. O termo multiplataforma refere-se a um software que pode ser executado em mais de uma plataforma de hardware e software. No contexto da informática, plataforma refere-se a dispositivos computacionais utilizados para interação com o sistema, seja hardware ou software. Esse conceito normalmente é aplicado à portabilidade entre sistemas operacionais, porém o conceito também contempla a execução do software em diferentes plataformas de hardware, como em processadores x86, ARM e Altera / NIOS II.

Este software foi utilizado no cenário de validação da solução desenvolvida neste trabalho com o objetivo de simular um Controlador Lógico Programável (CLP) de um Grupo Motor-Gerador (GMG). Para isso foi preciso configurar um dispositivo Modbus RTU sobre uma interface serial RS-485 e um modelo de cálculo que simulasse as variáveis disponíveis em um CLP real. Após a configuração, o software Pulsar foi executado em um notebook com um conversor USB-RS485. O modelo de CLP simulado no software Pulsar foi o “PCC1302” da empresa Cummins.

O software Pulsar foi desenvolvido em C++ e sua função primária é adquirir amostras de sinais digitalizados e processá-las matematicamente. O processamento matemático é realizado através de modelos de cálculo, representáveis em diagramas de blocos, que executam diversas funções para tratar adequadamente cada sinal e obter informações relevantes do monitoramento.

O software Pulsar deve ainda armazenar em disco informações pertinentes ao monitoramento dos sinais, para que essas informações possam ser transmitidas posteriormente a um datacenter e analisadas adequadamente por um especialista. O software dispõe ainda de diversos serviços implementados via TCP/IP em protocolo proprietário para visualização e controle do processo de monitoramento.

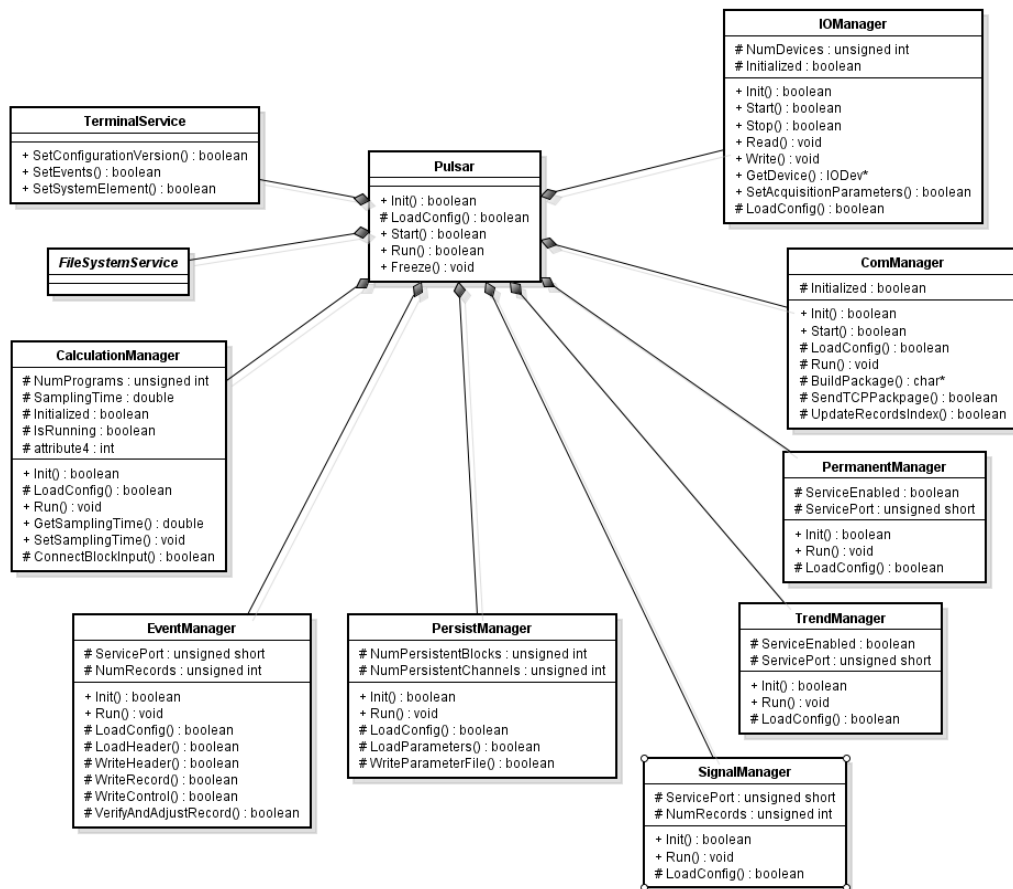
Dependendo da arquitetura de hardware em que o software está sendo executado, é possível ainda acionar alarmes ou saídas digitais e com isso interromper o funcionamento de equipamentos de alto valor agregado em caso de falha.

O software foi desenvolvido visando os padrões ANSI C (American National Standards Institute for the C programming language) e POSIX (Portable Operating System Interface), especificado pela ISO/IEC 9945 (ISO/IEC/IEEE 9945:2009), o que torna o software compatível com a maioria dos compiladores.

O Software Pulsar possui classes de gerenciadores, que são responsáveis por gerenciar instâncias de objetos que são especializações de uma determinada classe-base. Por exemplo, existe o gerenciador de dispositivos de comunicação, que é responsável por gerenciar as instâncias de objetos que controlam o envio de dados ao datacenter. Cada especialização da classe-base representa um dispositivo que pode ser controlado pelo software Pulsar e enviar dados ao datacenter, como um modem GPRS ou um modem para comunicação via satélite.

O diagrama de classes ilustrado na Figura 1 mostra parte do software Pulsar. A documentação não é apresentada de forma integral devido a confidencialidade do software, que é comercial.

Figura 1 - Diagrama de classes parcial do software Pulsar.



Fonte: autor.

3. DEFINIÇÃO DO PROBLEMA

Para garantir o retorno de investimentos em empreendimentos de geração de energia elétrica, é primordial a aplicação de filosofias de manutenção preditivas em detrimento de políticas corretivas e até mesmo preventivas. Neste sentido, a monitoração online das unidades geradoras desempenha papel fundamental na transição destas filosofias, pois fornece os meios necessários para estimar o estado do ativo e prever as falhas antes que elas aconteçam [10].

Em plantas geradoras centralizadas a potência nominal costuma ser da ordem de MW, em alguns casos até GW. Já no caso de GMGs (grupos motor-gerador) a potência costuma ser da ordem de kW. Neste mercado, o custo do ativo é muito menor em relação a uma planta hidrelétrica, mesmo quando comparado a PCHs. Por esta razão, o custo do monitoramento deve ser muito reduzido para que não represente uma parcela muito grande em relação ao custo do ativo de geração, inviabilizando o monitoramento.

Ainda dentro deste contexto, os locais de instalação dos grupos motor-gerador são os mais diversos. Em alguns casos há disponibilidade de rede Wi-Fi, mas na maioria dos casos é preciso utilizar redes celulares para a transmissão dos dados de monitoramento. Em ambos os casos a comunicação pode ser considerada instável, de alta latência e eventualmente de baixa largura de banda.

Devido a essas características, os conceitos de IoT e o protocolo MQTT se adequam perfeitamente a este tipo de aplicação, pois é preciso lidar com baixa capacidade de processamento (baixo custo de hardware) e redes de alta latência, baixa banda de comunicação e alta instabilidade.

Para viabilizar economicamente o desenvolvimento deste sistema (cuja principal característica é o baixo custo do hardware) é preciso considera-lo em alta escala. Como visto anteriormente, o conceito de computação em nuvem está perfeitamente alinhado a essas necessidades, visto que traz benefícios como flexibilidade, segurança e escalabilidade.

Em um sistema concebido para aplicação em alta escala, o conceito de redução de esforços de configuração é extremamente desejável. Esse conceito visa reduzir os esforços necessários para o gerenciamento das aplicações no lado do servidor, uma vez que a própria aplicação fornece os dados necessários para o funcionamento do sistema. Este conceito é aplicado, por exemplo, em redes de computador que criam de forma automática uma rede (zeroconf), permitindo que os usuários conectem uma variedade de dispositivos, como computadores, impressoras e smartphones sem precisar configura-los.

Quando é realizada uma publicação de dados via protocolo MQTT, é preciso publicar também o “local” onde essas informações devem ser armazenadas para serem retransmitidas aos dispositivos que estão cadastrados para receberem esses dados. No protocolo MQTT este “local” é chamado de “tópico”. Comumente os *brokers* disponíveis na internet implementam listas de controle de acesso (ACL – *Access Control List*) para gerenciar as permissões de usuários em uma hierarquia de tópicos.

No protocolo MQTT todos os dados são transmitidos em forma de texto. Ao transmitir, por exemplo, um número inteiro, este número deve ser convertido para texto e encapsulado no protocolo. Se uma aplicação deseja utilizar este número como inteiro após receber a transmissão deve primeiro converter o texto em inteiro. Isto implica que a aplicação deve conhecer o tipo de dado que está sendo recebido em cada tópico caso queira criar regras sobre o valor recebido.

Se considerarmos um caso onde uma aplicação utiliza o protocolo MQTT para receber dados de diversos modelos de equipamentos, por exemplo CLPs de grupos motor-gerador, é comum que pelo menos parte dos dados seja diferente entre os modelos de equipamentos. Mesmo quando os dados se referem a mesma variável, por exemplo a tensão da bateria, os diferentes modelos podem fornecer os dados com diferentes números de algarismos significativos ou diferentes unidades de engenharia.

Aplicações mais simples que não tratam o valor ou o tipo de informação que está sendo recebida podem simplesmente apresentar a informação ao usuário em forma de texto em uma interface simples.

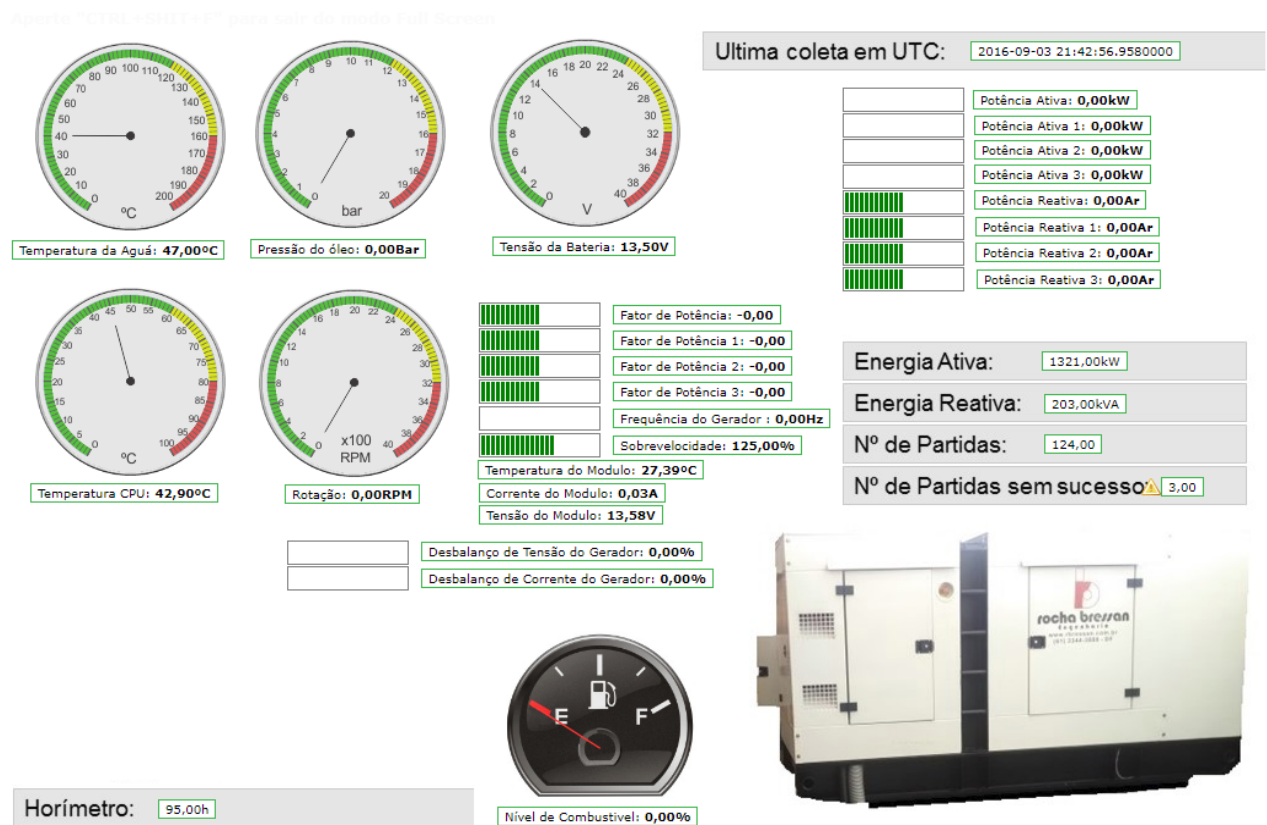
Para que seja possível apresentar as informações considerando o tipo de dado que está sendo recebido o usuário deve manualmente informar para a aplicação as informações referentes ao sinal monitorado. Por exemplo, se a aplicação está recebendo o nível de combustível do tanque em uma variável que varia de 0% a 100% e o usuário deseja utilizar um componente gráfico para apresentar esta informação em forma de um ponteiro analógico, o usuário deve manualmente configurar a aplicação para mostrar esta variável desta forma. Se imaginarmos que cada aplicação possui algumas dezenas de variáveis monitoradas e que o sistema de monitoramento possui centenas ou milhares de grupos motor-gerador monitorados, esta é uma atividade que demanda muito esforço manual.

A proposta é que a aplicação não publique somente os dados de monitoramento, mas publique também informações referentes ao tipo de dado que está sendo monitorado, como o tipo de variável (texto, inteiro, ponto flutuante, etc.), a grandeza (nível, tensão, corrente, pressão, etc.), a unidade de engenharia, o maior e o menor valor esperado para a variável (útil para componentes gráficos em supervisórios), etc.

De posse dessas informações a aplicação pode criar regras e apresentar as informações ao usuário de forma mais amigável. Ao definir uma variável como tensão do gerador, por exemplo, a aplicação pode ser capaz de criar automaticamente um componente gráfico de marcador de tensão. Ao extrapolarmos esta regra para diferentes tipos de variáveis, a aplicação pode ser capaz de montar dinamicamente um *dashboard* de forma automática e apresentar as informações ao usuário de forma muito mais agradável e inteligível.

A Figura 2 mostra um exemplo de tela de um supervisório WEB criado em um projeto piloto na empresa AQTech Engenharia e Instrumentação S.A. para o monitoramento de um grupo motor-gerador.

Figura 2 – Supervisório WEB de um projeto piloto.



Fonte: autor.

Percebe-se que foram utilizados diferentes componentes gráficos para a exibição dos sinais. Por exemplo, para a exibição da tensão da bateria foi utilizado um componente que imita um ponteiro analógico marcando de 0V até 40V. Para a exibição da potência ativa do gerador foi utilizado um componente do tipo “barra de progresso” com limite superior em 110% da potência nominal do grupo motor-gerador. Foram utilizados ainda componentes simples de exibição em texto e um marcador de combustível do tipo “ponteiro analógico” marcando de 0 a 100%.

Todos esses componentes foram configurados manualmente para cada sinal deste projeto piloto em específico. Extrapolando o esforço despendido para a criação desta tela de supervisório a outros geradores de tamanhos e modelos distintos o esforço demandado seria enorme, inviabilizando a escalabilidade do projeto, característica crucial para a viabilidade econômica.

Foram gastas cerca de 40 horas para o desenvolvimento da interface supervisória da Figura 2. Ainda, na ferramenta utilizada, para realizar a “cópia” da interface supervisória deste modelo de CLP para outros geradores com CLP de mesmo modelo é preciso em torno de 8

horas de desenvolvimento e validações, pois como o processo é realizado manualmente, todo o desenvolvimento deve ser validado.

Neste exemplo é preciso ainda preparar o servidor para receber os dados em uma base de dados relacional e configurar cada sinal manualmente para que a análise possa ser realizada posteriormente de forma otimizada.

Em um sistema cuja viabilidade econômica está vinculada à escala de desenvolvimento, os esforços de gerenciamento do lado do servidor para criar simples interfaces supervisorias já tornam o projeto inviável, enaltecendo a importância dos Serviços de Descoberta, que visam reduzir ao máximo os esforços de gerenciamento do lado do servidor por intervenção humana.

4. SOLUÇÃO PROPOSTA

O projeto foi executado no contexto de uma dissertação de mestrado realizado no Instituto Federal de Santa Catarina (IFSC) em uma parceria com a empresa AQTech Engenharia e Instrumentação S.A e contou com o apoio do Grupo de Sistemas Embarcados e Distribuídos NERsD do IFSC.

A metodologia utilizada para o desenvolvimento do projeto foi de natureza aplicada, objetivando gerar conhecimentos para aplicação prática. A pesquisa pode ser classificada também como do tipo exploratória, envolvendo experiências práticas com o problema pesquisado para o levantamento de informações. A pesquisa foi baseada em um estudo de caso, focando em uma unidade do sistema protótipo desenvolvida ao longo do projeto.

As etapas para o desenvolvimento da pesquisa envolveram pesquisa bibliográfica, definição do problema, estudo de trabalhos relacionados, proposta de solução e análise de resultados. Técnicas de simulação foram tomadas como base de validação das soluções em laboratório.

O acompanhamento foi realizado através de reuniões ao longo da execução do projeto. Nas reuniões foram elaborados registros do andamento do projeto e das adequações realizadas ao longo de toda a execução para se chegar ao objetivo esperado.

O desenvolvimento e integração seguiram princípios do QRPD - *Quality Rapid Product Development* (desenvolvimento rápido de produtos com qualidade). O enfoque principal está no uso da prototipação como técnica de antecipação de problemas e de refinamento das características construtivas e funcionais dos equipamentos.

O projeto contemplou as seguintes etapas:

1. Estudo de tecnologias de Serviços de Descoberta existentes;
2. Benchmarking de produtos;
3. Especificação de hardware e software;
4. Definição arquitetural da solução;
5. Compra de componentes;
6. Prototipação física;
7. Implementação do software;
8. Validação do protótipo em laboratório;

Inicialmente é apresentada uma proposta de arquitetura de sistema de monitoramento que permite a leitura dos dados dos CLPs, a transmissão dos dados via internet para um sistema executado no conceito de computação em nuvem e a disponibilização destas informações em portal WEB. Em seguida é apresentada a definição do Serviço de Descoberta proposto.

4.1 Arquitetura de monitoramento

Os dados de monitoramento podem ser obtidos do CLP (controlador lógico programável) do grupo motor-gerador através do protocolo Modbus RTU via barramento serial RS-485 na maior parte dos casos. Normalmente é esta a interface que se tem disponível para os diversos tipos de CLP já observados pela AQTech em diversas aplicações. Alguns modelos de CLP utilizam também a interface RS-232 para Modbus RTU, porém esses modelos são minoria no mercado segundo a experiência da empresa.

O Sistema de Monitoramento deve também transferir os dados via internet através do protocolo MQTT. O *Broker* é um servidor que gerencia a troca de mensagens entre os *publishers* (quem publica os dados) e os *subscribers* (quem recebe as informações). Essa publicação pode ser dar por rede Wi-Fi, rede GPRS ou alguma alternativa de rede TCP/IP suportada pelo sistema de monitoramento.

Nesta arquitetura, o trabalho pesado de processamento e armazenamento para gerenciar grandes números de *publishers* e *subscribers* fica por conta do *Broker*, que pode ser executado na nuvem, deixando os sistemas embarcados com menos responsabilidades, o que permite utilizar dispositivos de menor custo para o monitoramento.

Os *subscribers* podem ser aplicativos *Mobile*, aplicativos WEB ou qualquer outro programa que seja capaz de se conectar ao *Broker* via protocolo MQTT e receber informações, inclusive uma plataforma SCADA (*Supervisory Control and Data Acquisition*).

O sistema proposto dispõe de um Serviço de Descoberta (*Discovery Service*) sobre o protocolo MQTT sem modificar a especificação do protocolo. Para isso as “facilidades” do Serviço de Descoberta são gerenciadas no *payload* (carga útil) do protocolo MQTT, de forma semelhante aos dados em si. Através deste sistema não é preciso modificar a especificação do protocolo MQTT, de forma que a implementação do Serviço de Descoberta pode ser realizada sobre aplicações já desenvolvidas e disponibilizadas na internet.

Este sistema permite que uma empresa provedora de serviços de gerenciamento de grupos geradores possa gerenciar diversos equipamentos de diversos clientes, de marcas e

modelos diferentes, sem se preocupar tanto com a implantação do sistema em novos modelos de equipamentos.

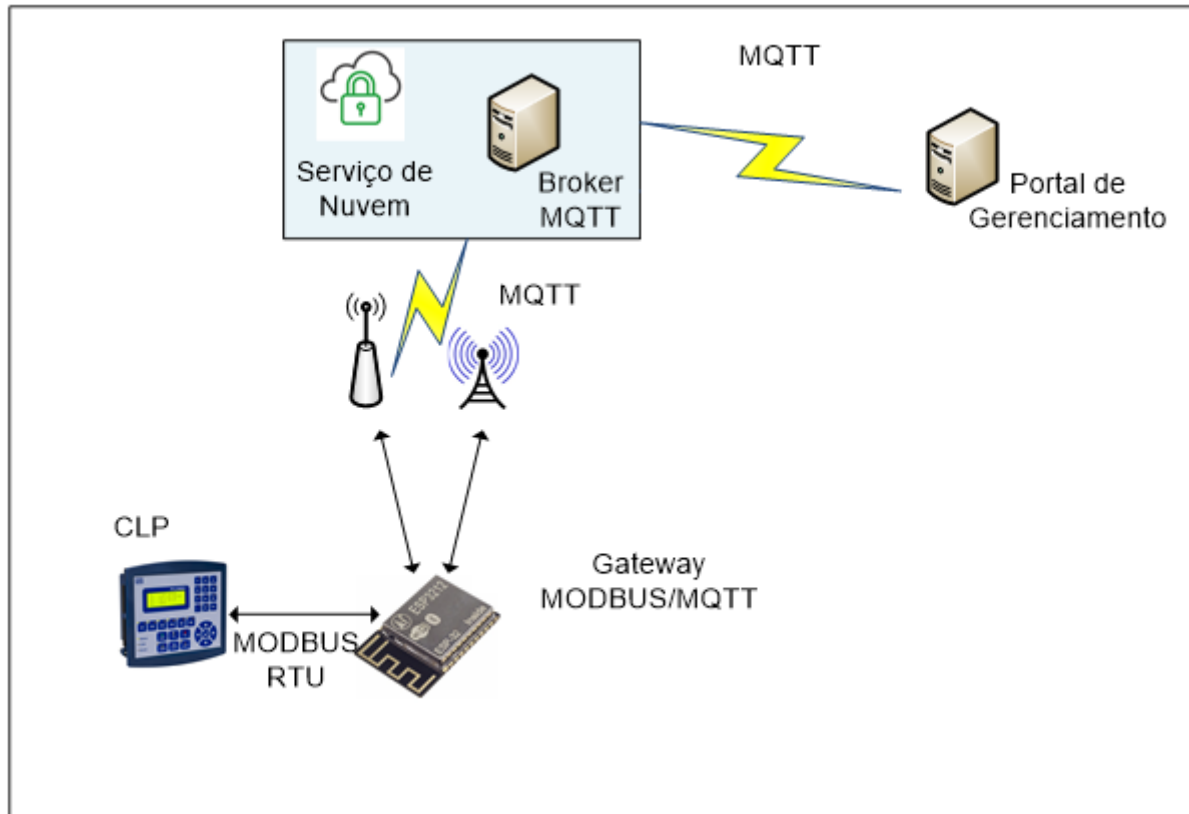
O Serviço de Descoberta permite que os dispositivos a serem gerenciados possam se anunciar automaticamente para a aplicação de gerenciamento, descrevendo as suas funcionalidades e comandos, minimizando o esforço de administração do prestador de serviços. Uma vez que o sistema de monitoramento publique suas facilidades, o sistema de gerenciamento deve ser capaz de interpretar essas “funcionalidades” e apresentar as informações de monitoramento de forma correta.

Existem diversos “*brokers*” MQTT disponíveis na internet, muitos com listas de controle de acesso que são capazes de controlar de forma hierárquica o acesso aos dados dos dispositivos, de forma que somente usuários autorizados possam obter informações do sistema monitorado com base em suas credenciais de acesso.

Como os grupos geradores são de pequeno porte e podem ter mobilidade, o ambiente a ser adotado está aderente ao paradigma da Internet das Coisas e a aplicação de gerenciamento opera com recursos de computação em nuvem.

A Figura 3 mostra a arquitetura de monitoramento proposta para o sistema protótipo. Neste modelo, o controlador Lógico Programável (CLP) controla o grupo gerador (que normalmente utiliza combustível diesel) e se comunica via protocolo MODBUS RTU (o mais comum entre os fabricantes de grupos geradores nacionais) com um módulo de gerenciamento que por sua vez, se comunicará através do protocolo MQTT (*Message Queuing Telemetry Transport*) com um middleware (*broker*) atuando em computação em nuvem. Este broker pode disponibilizar os dados para uma aplicação WEB de gerenciamento remoto acessível a partir de computadores pessoais e/ou dispositivos móveis ou até mesmo para um software desktop executado no dispositivo do cliente.

Figura 3 – Modelo de gerenciamento de grupos motor-gerador com Serviço de Descoberta sobre MQTT



Fonte: autor.

O protocolo MQTT atua no modo *Publisher/subscriber*. Quando o sistema de monitoramento do grupo motor-gerador é inicializado, ele publica no *broker* a sua lista de facilidades encapsulada nos tópicos do protocolo MQTT. Estas facilidades são disponibilizadas na aplicação de gerenciamento apenas para usuários autenticados com credenciais que possuam acesso ao sistema monitorado.

4.2 Definição do Serviço de Descoberta para MQTT

O sistema proposto prevê que a aplicação publique no *broker* informações referentes ao sinal monitorado assim que a conexão é estabilizada, após a autenticação. Essas informações são publicadas em tópicos (*payload*), de forma a não alterar a especificação do protocolo MQTT.

A Tabela 1 define os tópicos de configuração, seus possíveis valores e a descrição de cada um dos tópicos. Esta definição de tópicos se refere à revisão “0” do serviço proposto e será sempre desta forma para esta revisão. Desta forma, as aplicações cliente são capazes de utilizar os respectivos tópicos de forma automática.

Tabela 1 – Definição dos tópicos de configuração do Serviço de Descoberta.

Tópico	Valor	Descrição
revisao	0	Revisão do Serviço de Descoberta. Define os demais campos disponíveis para o Serviço de Descoberta especificado e deve ser incrementado em versões futuras.
descricao	-	Descrição da grandeza monitorada.
tipodado	0 – Booleano (0-1) 1 – Texto (ASCII) 2 – Uin8 3 – Int8 4 – Uint16 5 – Int16 6 – Uint32 7 – Int32 8 – Uint64 9 – Int64 10 – Float (32 bits) 11 – Double (64 bits)	Tipo de dado do valor monitorado.
grandeza	0 – Tensão elétrica 1 – Corrente elétrica 2 – Potência 3 – Frequência 4 – Ângulo 5 – Tempo 6 – Pressão 7 – Temperatura 8 – Nível 9 – Chave (aberto / fechado) 10 – Rotação	Grandeza física monitorada.
unidade	-	Unidade de engenharia do sinal monitorado.
algarismos	-	Número de algarismos significativos do sinal.
maximo	-	Maior valor esperado para o sinal.
minimo	-	Menor valor esperado para o sinal.

A variável monitorada deve ser transmitida no tópico “valor” (ausente na tabela, pois não é um tópico de configuração), que corresponde ao valor da grandeza monitorada, obtida através de algum transdutor, protocolo de comunicação, etc.

O tópico “valor” é atualizado periodicamente conforme o monitoramento, enquanto que os outros tópicos devem ser publicados pelo menos uma vez na inicialização do sistema, pois seus valores não são normalmente alterados durante o monitoramento. O serviço proposto não limita a publicação desses outros tópicos somente ao momento da conexão, ou seja, os dispositivos devem estar preparados para lidar com a alteração desses tópicos durante a operação do sistema. Esses tópicos são chamados de “tópicos de configuração” e devem ser publicados com a “*flag retain*”¹ do protocolo MQTT.

Dessa forma, as aplicações que se inscreverem nos tópicos recebem os últimos dados publicados mesmo após o sistema de monitoramento já estar em execução. Se os “tópicos de configuração” forem publicados sem a “*flag retain*”, somente as aplicações que já estiverem inscritas nestes tópicos no momento da publicação irão receber essas informações.

O Serviço de Descoberta define que somente os dois primeiros campos são obrigatórios: *valor* e *revisão*. Sinais do tipo “booleanos” e “texto”, por exemplo, não possuem unidade de engenharia, Algarismos significativos e nem valores máximos e mínimos esperados, pois essas informações não fazem sentido para esses tipos de dado. Entretanto, o sistema deve publicar o máximo de informações possível. Dessa forma as aplicações podem fazer uso destas informações sobre os dados monitorados.

A Tabela 2 mostra um exemplo de tópicos de configuração de uma aplicação de testes.

Tabela 2 – Exemplo de tópicos de configuração.

Tópico	Valor do tópico
/AplicacaoX/TensaoVa/revisao	0
/AplicacaoX/TensaoVa/descricao	Sinal de tensão da Fase A do gerador
/AplicacaoX/TensaoVa/tipodado	11
/AplicacaoX/TensaoVa/grandeza	0
/AplicacaoX/TensaoVa/unidade	kV
/AplicacaoX/TensaoVa/algarismos	3
/AplicacaoX/TensaoVa/maximo	15
/AplicacaoX/TensaoVa/minimo	0

¹ A “flag retain” é uma informação transmitida durante o “Publish” no protocolo MQTT e faz com que o dado, quando transmitido, permaneça em memória no Broker MQTT. Quando um cliente MQTT se inscreve no respectivo tópico, ele automaticamente recebe o último valor transmitido ao tópico. “MQTT Version 3.1.1”, Seção 3.3.1.3 (pg. 34), <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>.

No exemplo dado, os valores medidos do sinal configurado devem ser transmitidos no tópico “/AplicacaoX/TensaoVa/valor”. Os “tópicos de configuração” do sinal definem que o sinal transmitido é do tipo ponto flutuante de precisão dupla (double), a grandeza medida é uma tensão elétrica, possui unidade “kV”, 3 algarismos significativos e varia de 0kV a 15kV.

De posse dessas informações é possível que um programa supervisor que conheça os tópicos do Serviço de Descoberta monte dinamicamente um componente gráfico para visualizar esse sinal, por exemplo, conforme a Figura 4.

Figura 4 – Exemplo de componente gráfico para visualizar o sinal de teste.



Fonte: autor.

Perceba que o sinal é mostrado em um componente gráfico de aparência típica de voltímetros analógicos (a grandeza monitorada é de tensão elétrica), o valor mínimo é 0 kV, o valor máximo é de 15 kV e existe ainda um mostrador digital na parte inferior do componente gráfico com 3 algarismos significativos, mostrando o valor do sinal já na unidade de kV. Toda a informação necessária para montar este componente gráfico está disponível nos tópicos de configuração do Serviço de Descoberta proposto.

O tópico de configuração “revisao” permite que novas versões do Serviço de Descoberta sejam propostas futuramente, permitindo a expansão dos tópicos de configuração e dos valores possíveis para os respectivos tópicos.

A proposta do Serviço de Descoberta prevê ainda que seja possível que sinais sejam configurados em uma revisão superior à revisão de outros sinais à medida que novas revisões sejam lançadas. As aplicações que implementam o Serviço de Descoberta devem preferencialmente ser retro compatíveis, ou seja, aplicações que suportam a revisão 2 do Serviço de Descoberta devem preferencialmente suportar também a revisão 1 e a revisão 0.

Muitos sistemas SCADA possuem funcionalidades relacionadas ao armazenamento e análise dos sinais monitorados em base de dados. Para que seja possível utilizar os recursos de base de dados adequadamente, é essencial que se conheça o tipo de dado do sinal monitorado (inteiro, ponto flutuante, etc.). Essa informação está disponível no tópico de configuração “tipodado”.

A utilização do tipo de dado correto na base de dados traz benefícios como a menor utilização de espaço de armazenamento, ganhos de recursos de processamento e memória, a simplificação de consultas SQL, etc. O Serviço de Descoberta proposto permite, portanto, que os dados sejam persistidos em memória de forma mais adequada, facilitando a análise posterior.

5. RESULTADOS

O protótipo desenvolvido possui o objetivo de validação teórica dos conceitos apresentados neste trabalho. Portanto, foi utilizado um ambiente simulado de validação disponibilizando dados de monitoramento através do protocolo Modbus RTU em serial RS-485, semelhante à arquitetura encontrada em instalações reais. Neste caso, do ponto de vista do sistema de monitoramento, o simulador de dados é um CLP com dados reais de campo, permitindo a plena utilização dos conceitos apresentados neste trabalho.

5.1 Validações em laboratório

O sistema protótipo desenvolvido utiliza o kit de desenvolvimento “Wemos D1 Mini” [24], que possui o chip “ESP8266EX” [25]. Este chip é muito utilizado por entusiastas em projetos de IoT, pois se integra facilmente à interface de desenvolvimento do Arduino [26], permitindo o uso de diversas bibliotecas disponíveis na interface. Outras características que contribuem para a popularidade do chip é o baixo custo e Wifi integrado (Wireless padrão 802.11 b/g/n), normalmente com antena em circuito impresso nas placas que integram o chip.

O kit de desenvolvimento possui 16 pinos, antena Wifi embutida, conector micro USB e botão de *reset*. Para utilizar esta plataforma basta conectá-la ao computador através de um cabo USB e instalar o driver do conversor “USB-TTL” embutido no próprio kit de desenvolvimento.

A Figura 5 mostra o kit de desenvolvimento.

Figura 5 – Kit de desenvolvimento Wemos D1 mini.



Fonte: <https://i.ebayimg.com/images/g/d08AAOSwEzxYey3m/s-l1600.jpg>.

Alguns pinos do kit de desenvolvimento são de propósito geral (podem ser utilizados para diversos fins), como comunicação serial. O pino “5V” está diretamente conectado à alimentação da USB e serve para alimentar o regulador de tensão do kit de desenvolvimento. O pino “3V3” está conectado à saída do regulador de tensão do kit de desenvolvimento e alimenta o chip ESP8266EX, que trabalha em 3,3V (alimentação e lógica).

Os pinos “Rx” e “Tx” estão conectados ao conversor “USB TTL” integrado ao kit, de forma que é possível gravar e executar programas no kit de desenvolvimento diretamente pelo cabo USB, sem necessidade de nenhum circuito auxiliar. Também é possível utilizar esses pinos quando não estiver com um cabo USB conectado ao kit para comunicação serial com outros equipamentos. O kit de desenvolvimento possui ainda 4MB de memória Flash, que pode ser dividida para uso como memória de programa ou como memória em um sistema de arquivos.

A leitura dos dados do CLP em protocolo Modbus RTU em barramento serial RS-485 half-duplex é realizada pelo protótipo através de um *transceiver* RS485, modelo MAX3485 [27], conectado aos pinos “Rx” e “Tx”. O controle de direção da RS-485 half-duplex é realizado através de uma GPIO (entrada/saída de propósito geral).

Para os casos em que o CLP disponibiliza o protocolo Modbus via RS-232 é possível substituir o *transceiver* MAX3485 pelo MAX3232 [28]. Neste caso não é preciso utilizar a GPIO de controle de direção e nenhuma modificação é necessária no firmware do equipamento.

O protótipo desenvolvido utiliza uma fonte de alimentação de 7-40Vdc. Esta faixa é compatível com as tensões disponíveis em grupos motor-gerador, que são alimentados por uma ou mais baterias automotivas em série. O protótipo possui também uma interface serial para *debug* e uma interface RS-485 para conexão com o CLP dos grupos motor-gerador.

A configuração do protótipo é realizada via Wi-Fi por um computador ou smartphone através de um navegador WEB. A página de configuração permanece disponível somente quando o hardware é reiniciado em modo de configuração (com o botão CONFIG pressionado).

O hardware monitora também a tensão da bateria através de um conversor AD integrado ao microcontrolador, disponibilizando também esta informação via protocolo MQTT para monitoramento remoto.

A gravação do firmware no protótipo pode ser realizada diretamente através da interface serial do microcontrolador quando o protótipo é reiniciado em modo de gravação de firmware (com o botão FLASH pressionado).

O sistema possui ainda 3 LEDs (diodo emissor de luz) com o objetivo de indicar o estado do equipamento. É possível identificar facilmente problemas de conexão (Wi-Fi ou GPRS) ou um problema de comunicação serial (Modbus).

Quando o botão “Configurar” está pressionado no momento em que o equipamento é energizado ou reiniciado o software inicia o modo de configuração via Wi-fi. Neste modo, o equipamento cria uma rede Wi-fi onde equipamentos como smartphones e notebook podem se conectar e configurar o monitoramento através de um navegador WEB.

Quando o equipamento já está configurado e é energizado ou reiniciado com o botão “Configurar” não pressionado, o firmware entra no laço principal, que envolve a aquisição de dados através da serial RS-485 via Modbus RTU e através do conversor AD (tensão da bateria). Em seguida os dados são transmitidos via MQTT e o protótipo aguarda para realizar essa rotina novamente.

A Figura 6 mostra a página de configuração do protótipo acessada através de um smartphone.

Figura 6 – Página WEB de configuração do protótipo.

Configurações do Sistema

Lista de redes localizadas

Nome (SSID)	Senha (S/N)	Potência (dBm)
VIVO-7E50	Sim	-83
HAKUNA_MATATA	Sim	-67
#NET-CLARO-WIFI	Não	-68
HAKUNA_MATATA 2.4-2	Sim	-86

Wireless:

Senha:

Modelo do CLP:

ID Modbus: (1-255)

Baudrate:

Paridade:

Stop bits:

Usuario MQTT:

Senha MQTT:

Serial HW: de2481

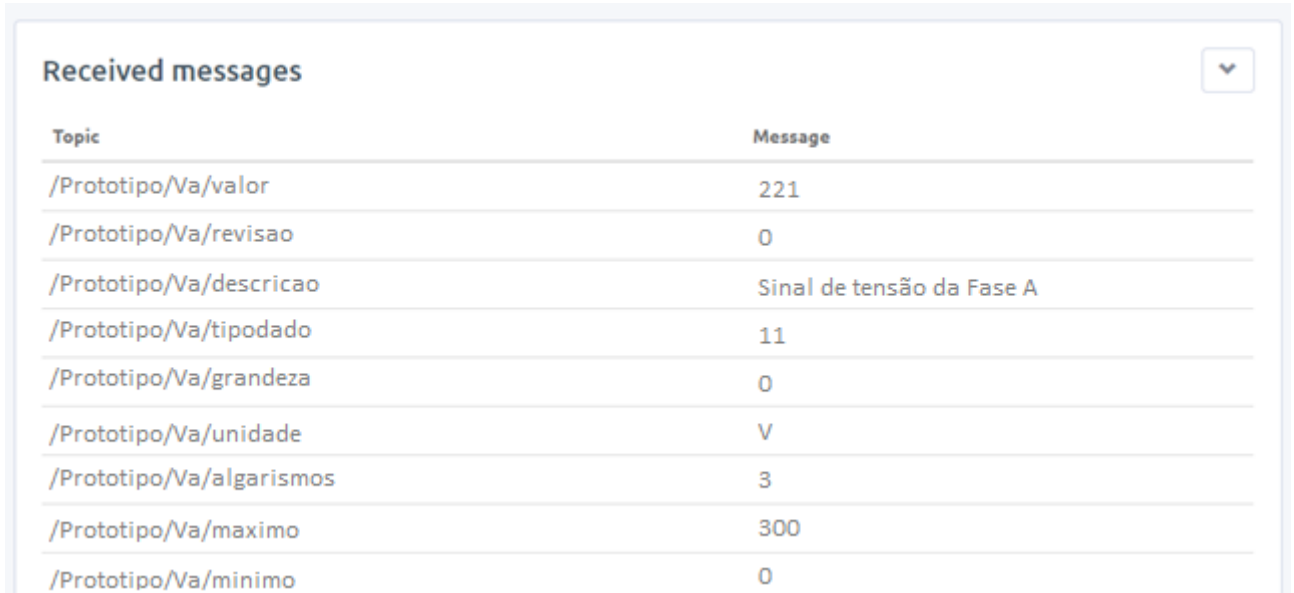
Firmware: 1.0.0

Fonte: autor

Os dados de monitorados são transmitidos a uma plataforma chamada CloudMQTT (www.cloudmqtt.com), que funciona como um Broker MQTT agindo em computação na

nuvem e configurável via WEB. A Figura 7 mostra as mensagens do protótipo desenvolvido sendo recebidas por essa plataforma.

Figura 7 – Mensagens recebidas pelo MQTT Cloud.



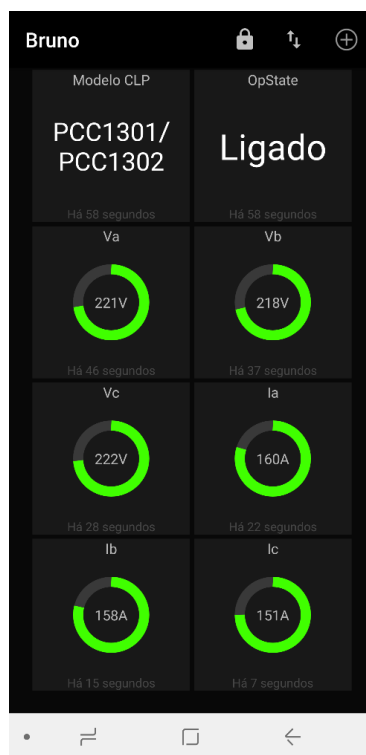
Topic	Message
/Prototipo/Va/valor	221
/Prototipo/Va/revisao	0
/Prototipo/Va/descricao	Sinal de tensão da Fase A
/Prototipo/Va/tipodado	11
/Prototipo/Va/grandezza	0
/Prototipo/Va/unidade	V
/Prototipo/Va/algarismos	3
/Prototipo/Va/maximo	300
/Prototipo/Va/minimo	0

Fonte: autor

A figura mostra a variável “Va” com o seu valor e todos os campos de configurações definidos por esta proposta com seus respectivos valores. O campo “valor” corresponde à última leitura deste sinal. O campo “tipodado” está com o valor 11, que corresponde a ponto flutuante de precisão dupla (double). O campo “grandezza” está com valor 0, que corresponde à “tensão elétrica”. Os outros campos definem ainda a unidade do sinal (V), o número de algarismos significativos (3), o maior valor esperado para o sinal (300 V) e o menor valor esperado (0 V).

A Figura 8 mostra um dashboard com os dados de monitoramento em um cenário de testes. A aplicação “MQTT Dash” foi utilizada para gerar este *dashboard*. Esta aplicação é um cliente MQTT que permite a visualização dos dados monitorados de diferentes maneiras e está disponível na Play Store para smartphones com sistema operacional Android. Existem inúmeras alternativas de aplicativos que implementam o protocolo MQTT, porém esta ferramenta foi escolhida por ser popular, fácil de se utilizar e com boa apresentação gráfica.

Figura 8 – Dashboard gerado no aplicativo MQTT Dash.



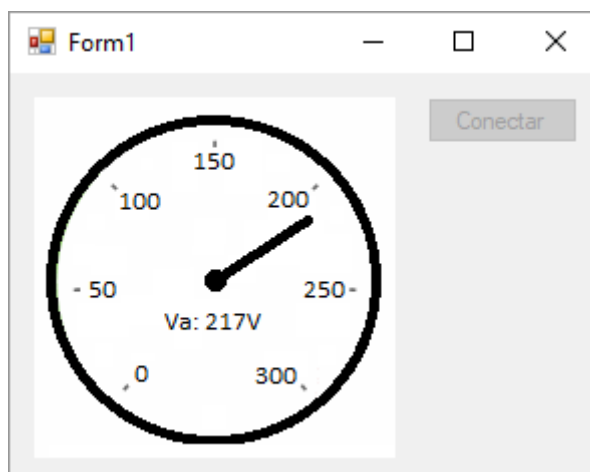
Fonte: autor

O aplicativo “MQTT Dash” não implementa o Serviço de Descoberta proposto neste trabalho, porém permite que o usuário configure diversos componentes gráficos para mostrar os sinais monitorados de diferentes formas, como texto, círculos preenchidos parcialmente, etc. Este aplicativo foi utilizado, portanto, para mostrar que todas essas configurações poderiam ser realizadas automaticamente através do Serviço de Descoberta proposto neste trabalho, reduzindo muito os esforços de gerenciamento.

Para comprovar a eficácia do Serviço de Descoberta proposto no auxílio à criação de interfaces supervisórias automaticamente, foi criado um software em C# utilizando a plataforma de desenvolvimento Visual Studio 2017 Community. O objetivo deste software foi simplesmente de replicar o resultado obtido através do aplicativo “MQTT Dash”, porém de forma automática, fazendo uso dos tópicos de configuração.

A Figura 9 mostra o resultado obtido para o sinal “Va”, sendo exibido em um componente gráfico do tipo “Gauge”. Por questões de simplificação o software desenvolvido exibe apenas um componente gráfico, montado dinamicamente a partir dos valores recebidos via MQTT.

Figura 9 – Componente gráfico montado automaticamente em um aplicativo de testes.



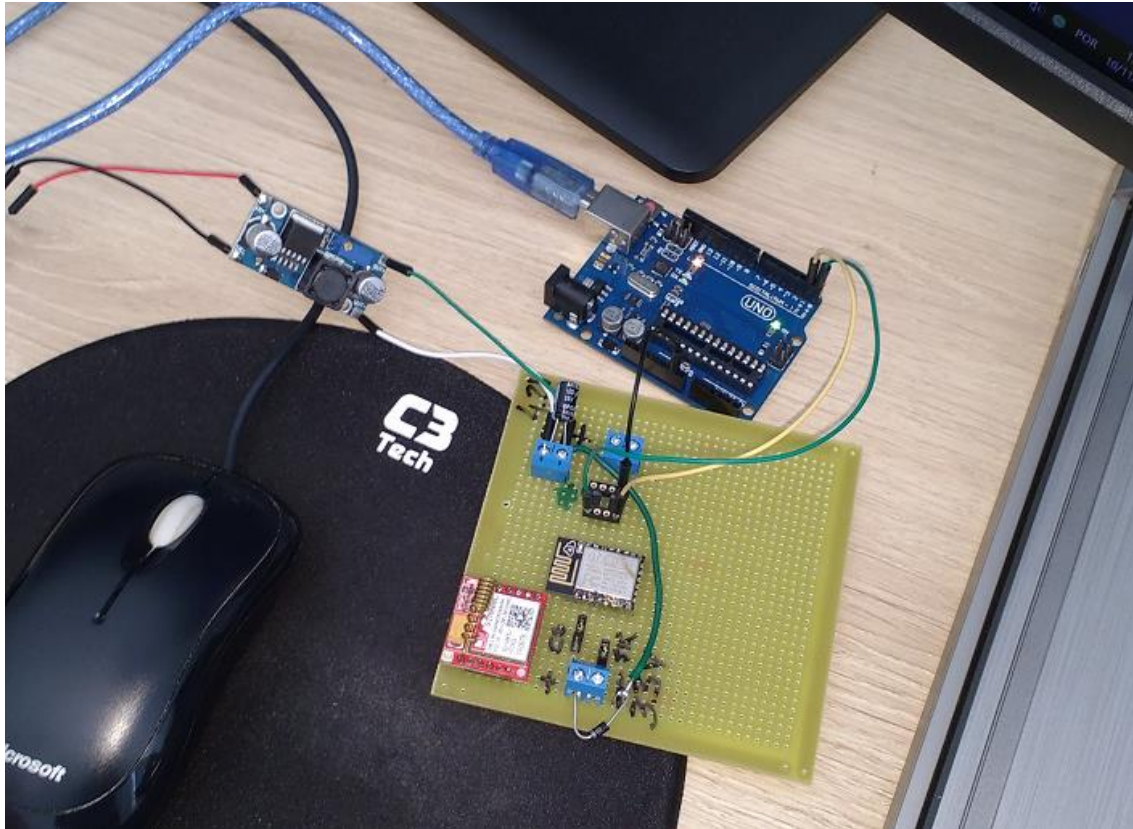
Fonte: autor.

O componente gráfico foi montado através de técnicas simples de programação, utilizando comandos de desenho como DrawLine (desenhar reta), DrawString (desenhar textos) e DrawEllipse (desenhar elipse) sobre uma área da janela do programa. Note que o programa de testes fez uso dos tópicos de configuração “mínimo”, “máximo”, “algarismos”, “unidade” e “tipodado” para montar o respectivo componente gráfico, além do próprio valor monitorado através do tópico “valor”, tudo através do protocolo MQTT e de forma autônoma.

Como um servidor Modbus, para simular os dados de monitoramento, foram utilizados hardware e software da empresa AQTech Engenharia e Instrumentação S.A. O software Pulsar foi configurado para atuar simulando um CLP de modelo PCC1302 da Cummins.

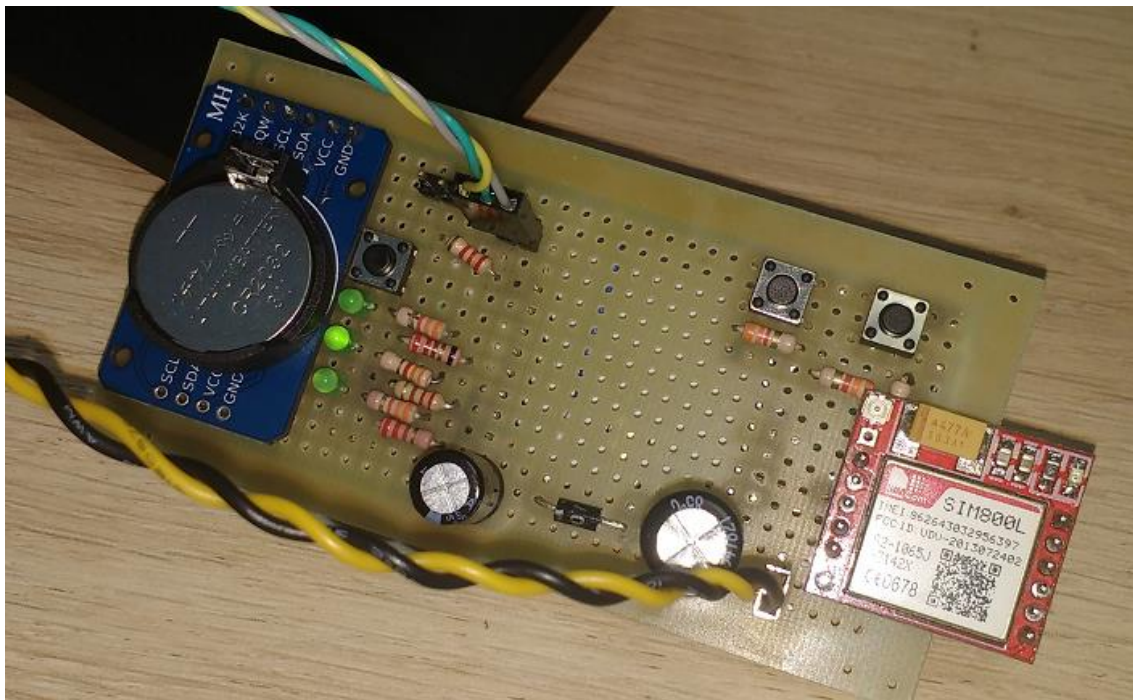
Foram utilizadas ainda diversas variações do kit de desenvolvimento ao longo do projeto para diversos testes. A Figura 10, Figura 11, Figura 12 e Figura 13 registram o processo de desenvolvimento e testes dos protótipos.

Figura 10 – Protótipo com ESP12E, fonte DC regulada e Arduino.



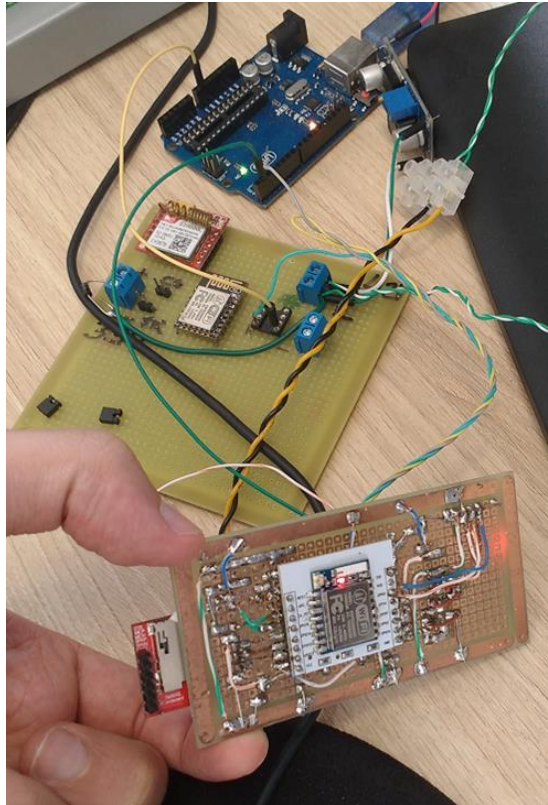
Fonte: autor.

Figura 11 – Vista superior de protótipo com ESP07, módulo GPRS e RTC.



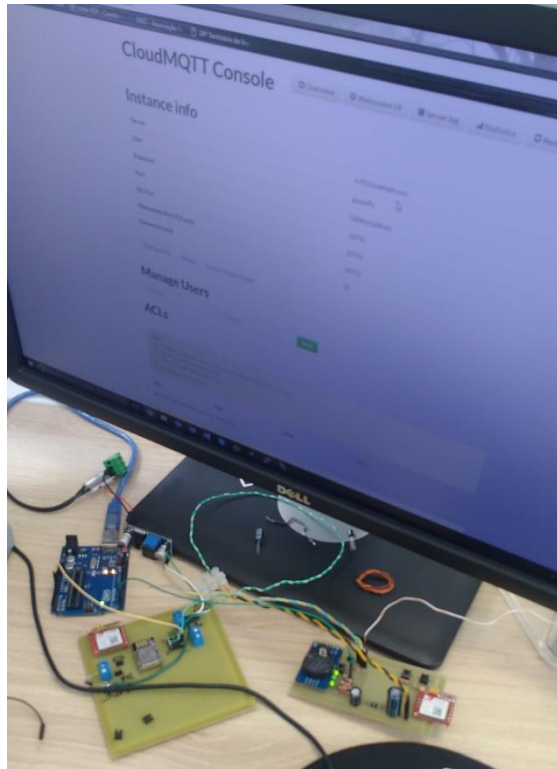
Fonte: autor.

Figura 12 – Vista inferior de protótipo com ESP07, protótipo com ESP12E, fonte DC regulada e Arduino.



Fonte: autor.

Figura 13 – Bancada com protótipos e sistema CloudMQTT sendo exibido em um monitor.



Fonte: autor.

5.2 Benchmarking

O mercado de monitoramento de grupos geradores (a diesel, gás natural, dentre outros combustíveis) é pulverizado de soluções com diversas limitações: algumas são caras e compatíveis somente com GMGs de uma marca específica. Outras têm custo mais acessível, mas configuram solução de simples telemetria – pouca memória e sem inteligência local – que não atendem as necessidades dos clientes de maneira efetiva. Em outros casos existem produtos que, embora pareçam nacionais, configuram uma integração ou adaptação de sistemas importados e projetados para outros fins, com domínio da tecnologia pulverizado em múltiplas empresas (uma domina o hardware, outra o software e uma terceira a infraestrutura de datacenter, por exemplo). Estes produtos compõem soluções ineficazes para o monitoramento remoto de ativos de geração de baixo custo.

As principais características não atendidas em sua totalidade pelos produtos existentes no mercado são as seguintes:

1. **Compatível com vários grupos geradores:** O sistema deve ser compatível com grupos geradores de várias marcas, modelos e capacidades, integrando os dados monitorados em uma mesma plataforma de software;
2. **Tamanho reduzido:** As dimensões do equipamento devem ser reduzidas para possibilitar a instalação dentro da carenagem do grupo gerador, que tipicamente tem espaço limitado;
3. **Custo efetivo:** o investimento no produto deve ser factível e sustentável para o monitoramento de ativos de geração de potência e preços reduzidos.
4. **Baixo consumo de energia:** normalmente a única fonte de energia elétrica é a própria bateria do GMG, bateria automotiva utilizada para partir o motor. O sistema de monitoramento deve possuir baixo consumo de energia para não interferir no funcionamento do GMG.
5. **“Ligar e Usar”:** o equipamento deve ser de fácil instalação, exigindo a menor quantidade possível de informações do cliente e automatizando o máximo possível as configurações.
6. **Fácil de configurar no servidor:** os esforços de configuração do lado do servidor para receber os dados de monitoramento de um equipamento específico devem ser os menores possíveis, viabilizando aplicações em escala.

7. Permitir facilmente a compatibilização com novos modelos de CLP: a plataforma de hardware e software deve estar preparada para integrar dados de novos modelos de CLPs com agilidade e pouco esforço de desenvolvimento.

Aplicações existentes no país são baseadas em produtos que não cumprem estas necessidades de maneira completa e efetiva. Estes sistemas similares se encaixam em uma das classificações abaixo:

1. Solução do tipo “1-rastreador veicular”: São soluções projetadas para rastreamento de frotas (carros e caminhões) e adaptadas para monitoramento de geradores. Tipicamente são soluções com hardware fraco e preço baixo, mas com recursos limitados por terem sido desenvolvidos para outro fim;
2. Solução do tipo “2-datalogger industrial genérico”: São sistemas de telemetria projetados para aplicações industriais diversas e que são adaptados para o monitoramento de grupos geradores;
3. Solução do tipo “3-proprietária de fabricantes de CLP”: São soluções que, integradas aos CLPs de grupos geradores, funcionam como sistemas de monitoramento remoto. Tipicamente são modems ou acessórios opcionais que são dependentes do CLP para funcionar;
4. Solução do tipo “4-proprietária de fabricantes de grupo gerador”: Engloba os sistemas projetados por fabricantes de grupos geradores. Tipicamente são difíceis de instalar, não são aplicáveis a outras marcas e tem preço alto.

O protótipo desenvolvido atende plenamente a todos os critérios identificados para este segmento de aplicação: A compatibilidade com vários modelos de CLP é garantida através da implementação do “mapa modbus” no firmware do protótipo, que pode ser atualizado através de métodos OTA (*Over the Air*), sem a necessidade de intervenção humana, atendendo dessa forma também o critério de compatibilização com novos modelos de CLPs com agilidade e pouco esforço de desenvolvimento.

O tamanho reduzido é atendido facilmente (mesmo não tendo sido desenvolvida uma versão comercial do protótipo), pois os componentes utilizados possuem um tamanho bem reduzido. O custo efetivo é atendido facilmente devido à simplicidade do projeto de hardware e ao custo reduzido dos componentes utilizados (mesmo não tendo sido desenvolvida uma versão comercial do protótipo).

O baixo consumo de energia é atendido plenamente, pois o protótipo utiliza um microcontrolador de baixo consumo e com função “deep-sleep”, que reduz o consumo praticamente à zero enquanto aguarda o intervalo de tempo até a próxima leitura.

O critério “ligar e usar”, que implica na facilidade de instalação e configuração inicial, é atendido dentro do que é possível. Como o protótipo desenvolvido utiliza rede Wi-Fi, é preciso configurar no mínimo o nome da rede e a senha para acesso à internet e envio de dados de monitoramento ao servidor.

Além disso, é preciso configurar também o modelo do CLP e a interface serial que irá coletar dados via Modbus RTU do CLP. Essa configuração consiste no modelo do CLP, o ID do CLP (identificador do CLP no barramento serial), e configurações da porta serial, como baudrate, bit de paridade e número de stop bits. Por fim, é preciso configurar o nome de usuário e senha (credenciais de acesso) cadastrados no servidor para o respectivo equipamento.

No lado do servidor, por outro lado, a configuração é extremamente reduzida. É preciso configurar apenas as credenciais de acesso do equipamento (usuário e senha) e a lista de controle de acessos, para que o equipamento em questão possa somente enviar dados aos respectivos tópicos do protocolo MQTT, sem possuir permissões para interferir nos demais monitoramentos.

A Tabela 3 resume a análise comparativa entre as categorias de soluções encontradas no mercado e a solução desenvolvida neste projeto.

Tabela 3 – Comparação entre categorias de soluções similares e o protótipo desenvolvido

Critério		Classificações de soluções similares				Protótipo
		Tipo “1”	Tipo “2”	Tipo “3”	Tipo “4”	
1	Compatível com vários grupos geradores	Sim	Sim	Não	Não	Sim
2	Tamanho reduzido	Sim	Sim	Sim	Não	Sim
3	Custo efetivo	Sim	Sim	Não	Não	Sim
4	Baixo consumo de energia	Sim	Sim	Sim	Sim	Sim
5	“Ligar e Usar”	Não	Não	Sim	Não	Sim
6	Fácil de configurar no servidor	Não	Não	Não	Não	Sim
7	Permitir facilmente a compatibilização com novos modelos de CLP	Não	Não	Não	Não	Sim

O protótipo desenvolvido se destaca principalmente nos dois últimos critérios avaliados (critérios 6 e 7), pois faz uso do Serviço de Descoberta proposto neste trabalho, permitindo a

automatização da maior parte das configurações tipicamente necessárias para o funcionamento do sistema, e permite ainda a atualização remota do firmware através de métodos OTA, sem a necessidade de intervenção humana, que em conjunto com o Serviço de Descoberta proposto permite facilmente a compatibilização com novos modelos de CLP com o menor esforço possível.

As soluções do tipo “1” e “2” possuem características muito próximas dos critérios descritos como importantes para este segmento de mercado, porém normalmente sua configuração ou parametrização é realizada por desenvolvedores em laboratório de forma fixa (*hard code*), sem nenhuma ou com muita pouca flexibilidade.

Quando é preciso instalar o mesmo equipamento em outros modelos de CLP, por exemplo, normalmente é preciso regravar o firmware em laboratório e reconfigurar o servidor para lidar com o novo firmware. O cliente, portanto, normalmente não consegue de forma autônoma desinstalar o equipamento de um GMG e instalar o mesmo equipamento em outro GMG com modelo de CLP diferente.

O protótipo desenvolvido supre essas necessidades, pois é possível parametrizar o equipamento em uma interface WEB simples através do próprio celular ou notebook do cliente. Caso o modelo de CLP não seja suportado pela versão de firmware presente no equipamento, é possível ainda atualizar o firmware via OTA (sem intervenção do cliente).

Ainda, o servidor recebe através dos tópicos de configuração do Serviço de Descoberta proposto as características dos sinais monitorados e a configuração do equipamento, podendo de forma autônoma lidar com mudanças de parametrização realizadas pelo cliente.

6. CONCLUSÕES

O objetivo principal do trabalho foi atingido, pois foi proposto um Serviço de Descoberta para o protocolo MQTT utilizando o próprio *payload* do protocolo, sem alterar sua especificação. Ainda, o serviço proposto foi utilizado em um sistema protótipo de monitoramento de grupos motor-gerador baseado em Internet das Coisas utilizando o protocolo Modbus RTU sobre um barramento serial RS-485 para obtenção das variáveis monitoradas. Ainda, o protótipo foi desenvolvido com componentes comerciais de baixo custo, comprovando a viabilidade econômica do sistema.

A solução proposta define que os tópicos de configuração são publicados no Broker MQTT assim que o sistema é iniciado. Essas informações são publicadas em tópicos do MQTT, de forma a não modificar a especificação do protocolo. Com esta estratégia foi possível utilizar aplicações já existentes no mercado com a proposta de Serviço de Descoberta abordada sem nenhuma modificação.

A funcionalidade do Serviço de Descoberta proposto foi avaliada dentro de um ambiente de testes em laboratório e os resultados foram considerados satisfatórios. Foi possível, por exemplo, gerar componentes gráficos para análise dos sinais monitorados de forma automática com base nas informações publicadas pelo Serviço de Descoberta.

Outras aplicações que utilizam os dados de monitoramento podem utilizar também as informações presentes nos tópicos de configuração dos sinais monitorados, por exemplo, para melhorar a experiência do usuário em interfaces SCADA de acordo com cada sinal monitorado. Para isso, basta que as aplicações se inscrevam nos tópicos especificados nesta proposta e utilizem essas informações para personalizar a visualização do sinal de acordo com as informações obtidas.

O serviço proposto traz outros benefícios além de facilitar a criação de *dashboards*. É possível, por exemplo, que um servidor na nuvem receba os dados monitorados de diversas aplicações e as armazene em uma base de dados de forma adequada e automatizada. Com isso a transferência de dados para o servidor, mesmo em novos modelos de CLPs, seria totalmente automatizada e reduziria significativamente os esforços de gerenciamento do lado do servidor.

O software seria capaz de armazenar os dados monitorados com o tipo correto de grandeza, como inteiro ou ponto flutuante, fazendo uso dos recursos da base de dados para tipos específicos de dados de forma otimizada.

Essa automação pode ser realizada utilizando o campo “tipodado” do Serviço de Descoberta proposto. Armazenar informações com o tipo de dado incorreto na base de dados (como números em forma de texto) é considerado no mínimo deselegante, além de anular funcionalidades e otimizações da base de dados para um melhor aproveitamento de recursos, como espaço em disco, memória e processamento.

Como sugestão para trabalhos futuros, seria possível refinar a proposta do Serviço de Descoberta e implementar uma plataforma de software que faça pleno uso dos recursos disponibilizados pelo Serviço de Descoberta proposto, por exemplo montando *dashboards* automaticamente, armazenando os dados monitorados em uma base de dados com os tipos de dados corretos, gerando valores estatísticos calculados sobre os sinais numéricos, correlacionando sinais com mesmo tipo de grandeza, etc.

As representações gráficas das quantidades medidas poderiam ser exibidas com base nas características dos sinais, como grandeza de engenharia, unidade, etc. Baseado no valor máximo e mínimo esperado para cada sinal, seria possível montar as representações gráficas, como barras de progresso, ajustadas de acordo com cada sinal.

Ainda como trabalho futuro, seria possível desenvolver comercialmente o protótipo desenvolvido neste trabalho, a fim de se chegar em um produto com características industriais e que faça uso do Serviço de Descoberta proposto. É necessário, por exemplo, a realização de ensaios de tipo (como compatibilidade eletromagnética, resistência a descargas elétricas, a choques mecânicos, ensaios de temperatura, etc.) e certificações.

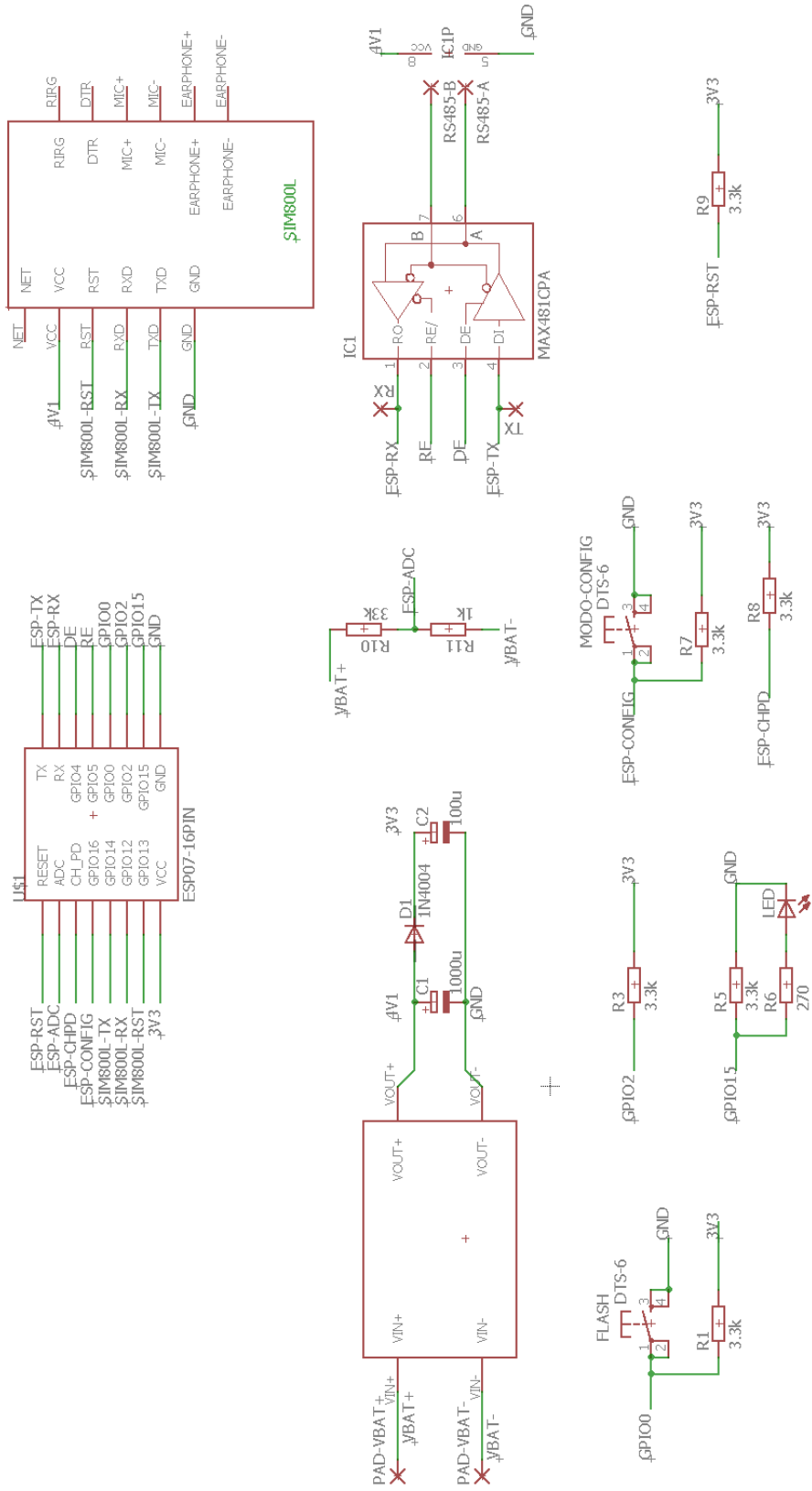
REFERÊNCIAS

- [1] Martins, I. R., Zem, J. L., “Estudo dos protocolos de comunicação MQTT e COAP para aplicações Machine-to-Machine e Internet das Coisas”, 2015.
- [2] Singh, D., Tripathi, G. e Jara, A. J., “A survey of Internet-of-Things: Future vision, architecture, challenges and services”. 2014 IEEE World Forum on Internet of Things, WF-IoT 2014, pp. 287–292, 2014.
- [3] Torres, A. B. B., Rocha, A. R., Souza, J. N., “Análise de desempenho de Brokers MQTT em sistemas de baixo custo”. Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat). Universidade Federal do Ceara (UFC), 2016.
- [4] J. Gantz and D. Reinsel, “The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east,” IDC iView: IDC Anal. Future, vol. 2007, pp. 1–16, Dec. 2012.
- [5] D. Evans, “The Internet of things: How the next evolution of the Internet is changing everything”, CISCO, San Jose, CA, USA, White Paper, 2011.
- [6] S. Taylor, “The next generation of the Internet revolutionizing the way we work, live, play, and learn”, CISCO, San Francisco, CA, USA, CISCO Point of View, 2013.
- [7] J. Manyika et al., “Disruptive Technologies: Advances that Will Transform Life, Business, and the Global Economy”. San Francisco, CA, USA: McKinsey Global Institute, 2013.
- [8] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, e J. Wang, “A first look at cellular machine-to-machine traffic: Large scale measurement and characterization”, in Proc. ACM SIGMETRICS Perform. Eval. Rev., pp. 65–76, 2012.
- [9] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. e Ayyash, M., “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”, IEEE Communication Surveys & Tutorials, Vol. 17, No. 4, 2015.
- [10] Ferreira, M. P., Freitas, F. L., Matsuo, T. K., Borba, B., Fonseca, J. E. R., “Monitoramento Online como Ferramenta para Otimização da Manutenção de Geradores: Uma Tecnologia 100% Nacional em Arquitetura Distribuída”, XI Simpósio de Automação de Sistemas Elétricos, Campinas – SP, 2015.

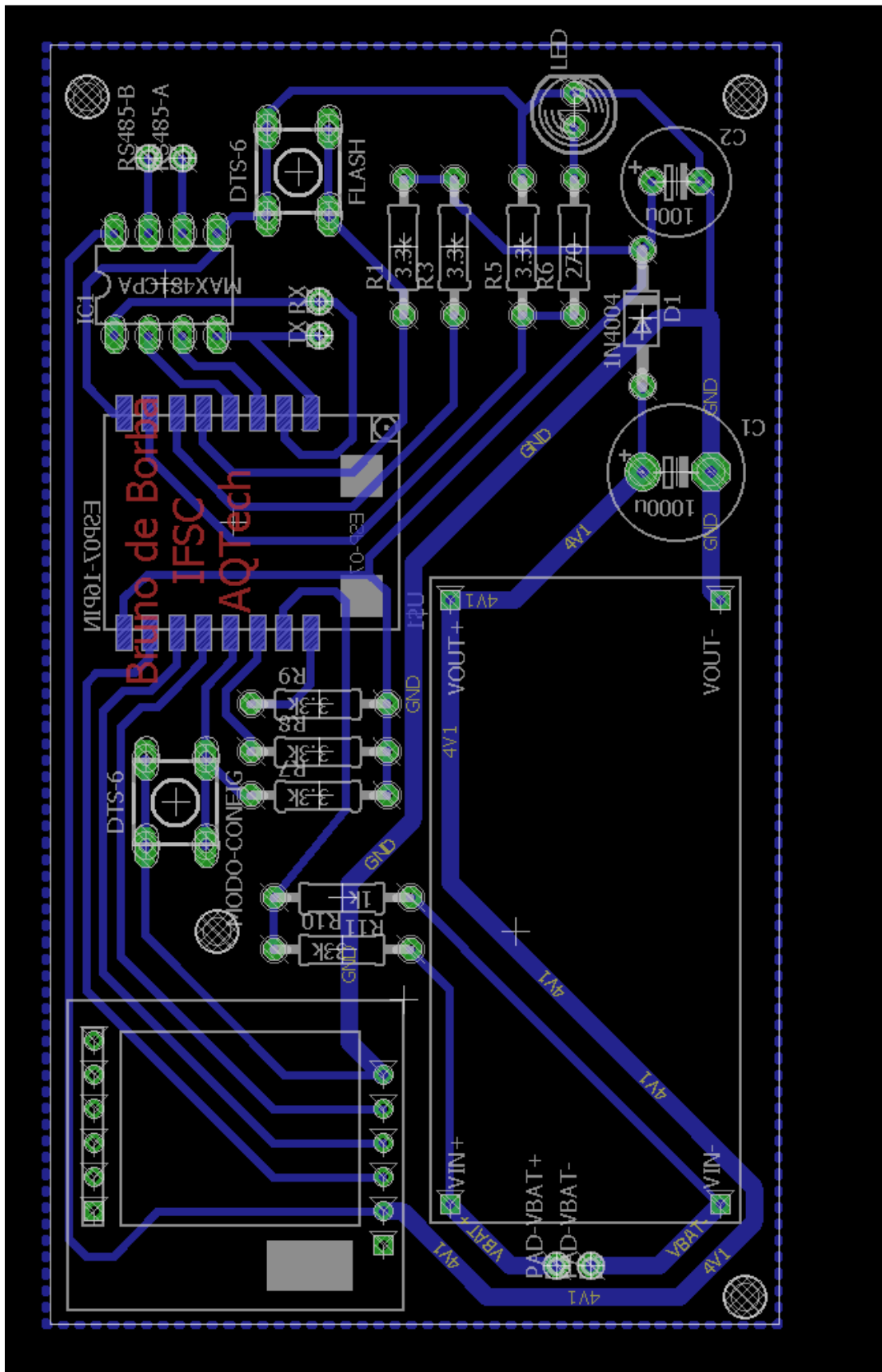
- [11] M.S. Crestani. “Dificuldades e oportunidades da crise”. Revista Eletricidade Moderna, São Paulo, ano 43, n.490, p.6. 2015.
- [12] Freitas, F. L., Ferreira, M. P., Matsuo, T. K., Borba, B., Fonseca, J. E. R., Neves, L. P. C., Buzzatti, M., Ávila, S. L., “Nacionalização de produto para monitoramento de grupos geradores: produto plataforma para o setor elétrico”, 2015.
- [13] Zhang, L. (2011). Building Facebook Messenger. <https://www.facebook.com/notes/facebook-engineering/building-facebookmessenger/10150259350998920>. Acesso em 24/10/2016.
- [14] Modbus Application Protocol Specification V1.1b3. http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf. Acesso em 26/10/2016.
- [15] MQTT Version 3.1.1. Edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS Standard. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>. Acesso em 22/11/2016.
- [16] Modbus FAQ. <http://www.modbus.org/faq.php>. Acesso em 26/10/2016.
- [17] Web Application Messaging Protocol. <http://wamp-proto.org>. Acesso em 01/11/2016.
- [18] Tech pages/IoT XepsExplained. http://wiki.xmpp.org/web/Tech_pages/IoT_XepsExplained. Acesso em 02/11/2016.
- [19] ESP32 Datasheet. https://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em 22/11/2016.
- [20] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR 5462 - Confiabilidade e Manutenibilidade. Rio de Janeiro, 1994
- [21] HIGGINS, LINDLEY R. WIKOFF, DARRIN J. Maintenance Engineering Handbook. 2008. 7ª edição. McGraw-Hill. Nova York. 1244 p
- [22] FREITAS, FABRIZO LEAL ET AL. Monitoração permanente de geradores: Abordagem para o diagnóstico preditivo. Seminário do Estado da Arte em Sistema de Monitoramento Aplicados a Maquinas Rotativas, Brasília, 2008.
- [24] D1 mini. https://wiki.wemos.cc/products:d1:d1_mini. Acesso em 23/11/2017.

- [25] ESP8266EX Datasheet. http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. Acesso em 23/11/2017.
- [26] ESP8266 core for Arduino. <https://github.com/esp8266/Arduino>. Acesso em 23/11/2017.
- [27] 3.3V-Powered, 10Mbps and Slew-Rate-Limited True RS-485/RS-422 Transceivers. <https://datasheets.maximintegrated.com/en/ds/MAX3483-MAX3491.pdf>. Acesso em 23/11/2017.
- [28] MAX3232 3-V to 5.5-V Multichannel RS-232 Line Driver/Receiver with ± 15 -kV ESD Protection. <http://www.ti.com/lit/ds/symlink/max3232.pdf>. Acesso em 23/11/2017.
- [29] <https://issues.oasis-open.org/browse/MQTT-267>. Acesso em 20/02/2018.
- [30] <https://home-assistant.io/docs/mqtt/discovery/>. Acesso em 20/02/2018.
- [31] <https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/>. Acesso em 26/02/2018.

ANEXO A – ESQUEMÁTICO DO PROTÓTIPO



ANEXO B – LAYOUT DO PROTÓTIPO



ANEXO C – CÓDIGO FONTE DO PROTÓTIPO

O código fonte do software desenvolvido para o protótipo está armazenado no repositório <https://github.com/brborba/mqtt-ds.git>.