

Sistema para envio remoto de comandos para inversores

A system for remote issuance of commands to inverters

RESUMO

Andreas Anael Pereira Gomes
andreasgomes@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Winderson Eugenio dos Santos
winderson@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

João Pedro da Matta Galera da Silva
joaosilva.2016@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Gustavo Ramos Pereira
gpereira@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Nesse trabalho é descrita a implementação de um sistema que permite o envio remoto de comando para inversores presentes em estações fotovoltaicas presentes em seis *campi* da UTFPR, permitindo a sincronização de limite e fator de potência entre os dados presentes no servidor localizado no campus Curitiba Sede da UTFPR e o que efetivamente está em uso nos 36 inversores espalhados pelo estado. A solução foi desenvolvida utilizando a linguagem de programação Python e o *framework* Django do lado do servidor, enquanto no computador local de cada campus foram utilizados a linguagem de programação Node-RED e um banco relacional em SQLite. Um computador local presente em cada estação fotovoltaica se comunica com o servidor, obtendo os valores de fator e limite de potência que devem ser enviados aos inversores da rede e então retorna uma resposta ao servidor confirmando a operação.

PALAVRAS-CHAVE: Inversores elétricos. Energia solar. Aplicações *Web*.

ABSTRACT

This work describes the implementation of a system that allows the remote issuing of commands to inverters present at photovoltaic stations across six of UTFPR's campuses, enabling the synchronization of the power limit and the power factor effectively in use by the inverters with the data stored in server located at the main campus of UTFPR, in Curitiba. This solution was implemented with the Python programming language and the Django framework on the server side, while the local computer for each campus uses the Node-RED programming language and a relational database using SQLite. These local computers act as the middleman between the server and the inverters, acquiring the power factor and limit values that need to be issued to each inverter from the server, and sending confirmation back to the server when the operation succeeds.

KEYWORDS: Electric inverters. Solar energy. Web applications.

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

A rede de estações de pesquisa em energia solar foi implantada em um projeto de pesquisa desenvolvido pela UTFPR e pela COPEL, com estações solarimétricas localizadas em seis *campi* da UTFPR (LABORATÓRIO DE ENERGIA SOLAR, 2019). Cada estação coleta dados meteorológicos e registra dados de irradiação solar, além de módulos de avaliação de diferentes tecnologias de painéis fotovoltaicos, com inversores da família NHS *On Grid* SOLAR GMSR1 (NHS SOLAR, 2020) acoplados a cada tipo de tecnologia. Como os inversores são acoplados à rede elétrica da COPEL e suportam que o usuário configure limite e fator de potência para despacho, há interesse em disponibilizar de forma unificada um mecanismo que permita ao usuário configurar esses valores remotamente.

MATERIAL E MÉTODOS

O funcionamento do sistema é baseado em três tarefas distintas, executadas em paralelo no computador local chamado de ST-One (SMART-TECH, 2018): “busca remota de estado”, “envio de comandos aos inversores” e “atualização remota de estado”. Há um servidor que disponibiliza ao ST-One de cada campus os valores de fator de potência e limite de potência que devem ser configurados a cada inversor. O código do ST-One foi programado em Node-RED (IBM, 2019), com tabelas locais armazenadas em SQLite (HIPPI, R. D., 2019), enquanto a aplicação *Web* do servidor foi desenvolvida em Python (PYTHON SOFTWARE FOUNDATION, 2020) utilizando o *framework* Django (DJANGO SOFTWARE FOUNDATION, 2020).

BUSCA REMOTA DE ESTADO

Nesta tarefa, o ST-One busca a informação do servidor através de uma requisição *GET* em um *Uniform Resource Locator* (URL) específico a cada dez minutos. A requisição terá como resposta dados no formato *JavaScript Object Notation* (json). A quantidade de entradas desse json depende dos parâmetros solicitados durante a requisição, filtrando inversores por campus ou tecnologia. O Quadro 1 apresenta os campos presentes no json retornado pela requisição *GET*.

Quadro 1 – Tipografia das seções

Campo	Descrição	Escopo
<i>campus_cod</i>	Código do <i>campus</i> .	Dois caracteres.
<i>inv_name</i>	Nome do inversor.	Quatro caracteres.
<i>power_factor</i>	Valor de fator de potência configurado para o inversor.	0.80 a 1.00, reais com duas casas.
<i>power_limit</i>	Limite de despacho de potência configurado para o inversor.	0 a 100, inteiros.
<i>update_time</i>	Data e hora da alteração de fator ou de limite de potência desejada.	Formato de data: aaaa-mm-dd HH:MM:SS
<i>status</i>	Se a operação foi confirmada por um inversor ou não.	“Updated” caso sim ou “Applied”, caso não.
<i>pf_type</i>	Especifica se o fator de potência é indutivo, capacitivo ou unitário.	“Capacitive”, “Inductive” ou “Unitary”.

Fonte: Autoria própria (2020).

Após obter uma resposta, o programa processa o json e compara essas informações com as salvas em um banco de dados local no ST-One feito em SQLite em duas tabelas: “*inv_pow_factor*” e “*inv_pow_limit*”. Os Quadros 2 e 3 apresentam seus valores.

Quadro 2 – Campos da tabela de fator de potência *inv_pow_factor* dos ST-Ones

Campo	Descrição	Escopo
<i>inv_name</i>	Nome do inversor.	Dois caracteres.
<i>inv_code</i>	Endereço da rede RS485 para o inversor.	Quatro caracteres.
<i>update_time</i>	Horário extraído do json. Data e hora que foi feita uma alteração no servidor para dado inversor.	Formato de data: aaaa-mm-dd HH:MM:SS
<i>power_factor</i>	Valor de fator de potência configurado para o inversor.	0.80 a 1.00, reais com duas casas.
<i>pf_type</i>	Especifica se o fator de potência é indutivo, capacitivo ou unitário.	“Capacitive”, “Inductive” ou “Unitary”.
<i>pow_factor_command</i>	Valor que o inversor usa como entrada para realizar o comando.	1 a 20 e 80 a 100.
<i>effective_factor</i>	Valor de fator de potência efetivamente em uso no inversor.	-0.80 a -0.99 e 0.80 a 1.00.
<i>status</i>	Utilizado para controle se o comando precisa ser enviado para o inversor ou se é necessário enviar a confirmação para o servidor.	“Updated”, “Applied” ou “Pending”.

Fonte: Autoria própria (2020).

Quadro 3 – Campos da tabela de fator de potência *inv_pow_limit* dos ST-Ones

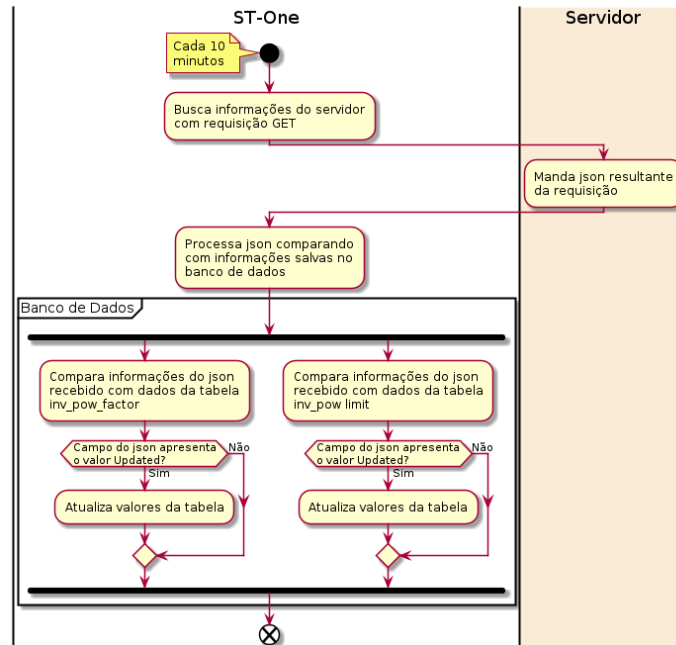
Campo	Descrição	Valores
<i>inv_name</i>	Nome do inversor.	Dois caracteres.
<i>inv_code</i>	Endereço da rede RS485 para o inversor.	Quatro caracteres.
<i>update_time</i>	Horário extraído do json. Data e hora que foi feita uma alteração no servidor para dado inversor.	Formato de data: aaaa-mm-dd HH:MM:SS
<i>power_limit</i>	Valor de limite de potência configurado para o inversor.	0 a 100, inteiros.
<i>effective_limit</i>	Valor de limite de potência efetivamente em uso no inversor.	0 a 100, inteiros.
<i>status</i>	Utilizado para controle se o comando precisa ser enviado para o inversor ou se é necessário enviar a confirmação para o servidor.	“Updated”, “Applied” ou “Pending”.

Fonte: Autoria própria (2020).

Se o campo “*status*” do json apresentar o valor “*Updated*”, ocorrerá atualização das tabelas locais “*inv_pow_factor*” e “*inv_pow_limit*”. A atualização dessas tabelas é feita comparando se o campo “*update_time*” apresenta uma data mais recente que a atual e checando se os dados de potência são diferentes dos atualmente em operação nos inversores — em caso afirmativo, os dados serão atualizados e o campo “*status*” para aquele inversor é configurado como

“Updated” nas tabelas onde isso se aplica. A Figura 1 apresenta o diagrama de estados para essa tarefa.

Figura 1 – Diagrama de estados para busca remota de estado



Fonte: Autoria própria (2020).

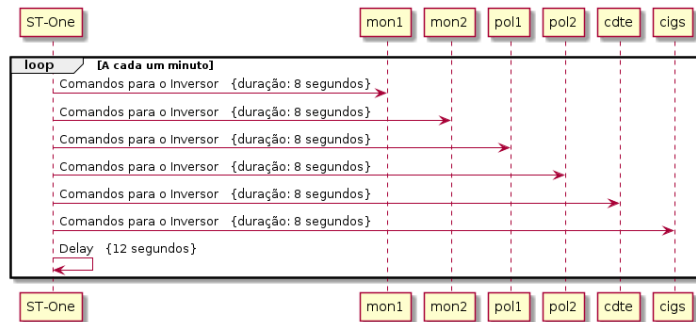
ENVIO DE COMANDOS AOS INVERSORES

Os comandos de ajuste de fator de potência e limitação de potência são enviados, se necessário, dentro do ciclo que se repete a cada minuto. Como os inversores e o ST-One estão ligados em uma rede RS485 do tipo *Half Duplex*, apenas um pacote de dados pode ser transmitido por vez, seja este um comando enviado pelo ST-One ou uma resposta enviada por um inversor.

A definição do próximo endereço para comunicação e a requisição de dados são comandos, logo, ocupam a rede. Sendo assim, o modo como o gerenciamento de tempo da rede RS485 funciona consiste em alocar um tempo para cada operação, mesmo que ela não venha a ocorrer.

A cada minuto, para cada inversor, esta sequência de comandos é realizada: alocação de endereço de inversor, requisição de dados, ajuste de fator de potência e ajuste de limite de potência. A temporização entre cada envio foi definida considerando a quantidade de comandos e respostas que precisam trafegar pela rede para respeitar a requisição de dados de minuto em minuto para todos os inversores existentes na rede de cada campus. A Figura 2 apresenta o diagrama de sequências para essa comunicação da rede RS485 para um campus entre seu ST-One e todos os inversores da usina fotovoltaica.

Figura 2 – Diagrama de sequência para a comunicação da rede RS485



Fonte: Autoria própria (2020).

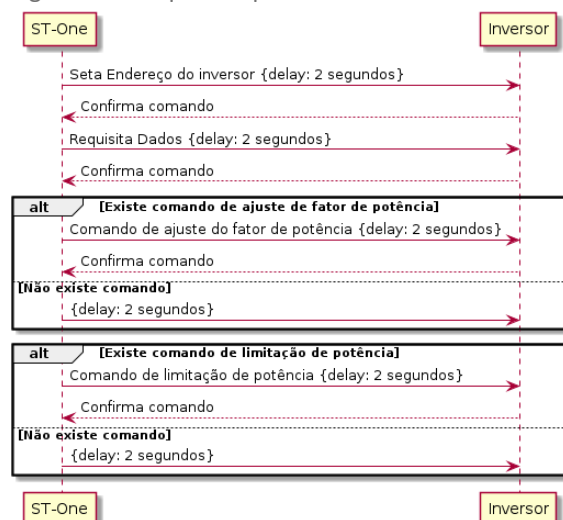
O processo é o mesmo para o envios de comandos de ajuste de fator de potência e ajuste de limitação: é consultado no armazenamento local do ST-One se o campo “status” das tabelas “*inv_pow_limit*” e “*inv_pow_factor*” apresenta o valor “*Updated*” e, caso sim, o comando é montado com os dados presente em “*pow_factor_command*” para o fator de potência ou “*power_limit*”, para limite de potência, e então enviado ao inversor. Esse processo se repetirá todo minuto, até que seja obtido uma confirmação do inversor de que o comando foi aceito.

Para seu respectivo comando aceito com sucesso, os campos “*effective_factor*” e “*effective_limit*” das tabelas “*inv_pow_factor*” e “*inv_pow_limit*”, respectivamente, são atualizados com o valor que acabou de ser aceito. O campo “status” então mudará de “*Updated*” para “*Pending*” e ficará com este valor até a tarefa de atualização remota de estado ser executada.

O campo “*exec_time*” da tabela “*command_history*” será atualizado com a data e hora de quando o inversor respondeu um comando de potência, para uma dada entrada. Se o comando for recusado, o campo “*event*” será atualizado com o valor “*Error*”. Se o comando for aceito, o campo será atualizado com o valor “*Set*”.

A Figura 3 apresenta o diagrama de sequência para a tarefa de envio de comandos de inversor para um inversor qualquer. Isso é repetido a cada minuto, para cada inversor, como representado anteriormente na Figura 2.

Figura 3 – Diagrama de sequência para o envio de comandos aos inversores



Fonte: Autoria própria (2020).

ATUALIZAÇÃO REMOTA DE ESTADO

Essa é a tarefa responsável por enviar ao servidor a informação de quais valores de fator de potência e limite de potência estão efetivamente em uso nos inversores de cada campus. A tarefa tentará ser realizada ou no momento que ocorre uma confirmação de resposta de um inversor ou a cada dez minutos no caso da presença campo “*status*” com valor “*Pending*” na tabela “*inv_pow_factor*” ou “*inv_pow_limit*” para qualquer inversor.

As tabelas locais são consultadas e, no caso da presença campo “*status*” com valor “*Pending*” na tabela “*inv_pow_factor*” ou “*inv_pow_limit*” para um dado inversor, será construído um json que será enviado à uma URL para a atualização dos valores. O Quadro 4 apresenta os campos desse json.

Quadro 4 – Campos do json enviado ao servidor

Campo	Descrição	Valores
<i>campus</i>	Código do <i>campus</i> .	Presentes no Quadro 1.
<i>inv</i>	Nome do inversor.	Presentes no Quadro 2.
<i>power_factor</i>	Valor de fator de potência efetivamente em uso.	0.80 a 1.00.
<i>pf_type</i>	Especifica se o fator de potência é indutivo, capacitivo ou unitário.	“ <i>Capacitive</i> ”, “ <i>Inductive</i> ” ou “ <i>Unitary</i> ”.
<i>power_limit</i>	Valor de limite de potência efetivamente em uso.	0 a 100.

Fonte: Autoria própria (2020).

Os campos “*power_factor*”, “*pf_type*” e “*power_limit*” são populados utilizando os valores presentes nos campos “*effective_factor*” e “*effective_limit*” das tabelas “*inv_pow_factor*” e “*inv_pow_limit*”. Os campos “*inv*” e “*campus*” são populados de acordo com a informação “*inv_name*” presente nas tabelas e com a informação local de campus gravada na memória do ST-One.

Após gerar o json, ele é enviado através de uma requisição *POST* por HTTP, com os *headers* “*Content-type: application/json*” e “*labens-token: <token-utilizado>*” e, se for obtido como resposta do *POST* o código de estado “*200 OK*”, o campo “*status*” das tabelas locais do inversor que gerou o json que possuíam o valor “*Pending*” mudará para “*Applied*”. Do lado do servidor, o campo “*status*” muda de “*Updated*” para “*Applied*” após receber o json.

RESULTADOS E DISCUSSÕES

Usuários com o nível de permissão necessário podem acessar os inversores através do Django *Admin*, na página *Web* do LABENS, acessível em praticamente qualquer navegador *Web*. A Figura 4 apresenta a página de lista e adição de inversores, com alguns disponíveis.

Clicando em um dos itens da Figura 4, a página com as opções presentes na Figura 5 abre, o que permite ao usuário modificar os parâmetros de fator e limite de potência. Os inversores utilizados no projeto possuem um limite de fator de potência entre 0.80 e 1.00 (indutivo ou capacitivo), mas os campos “*FpMin*” e “*FpMax*” permitem que o sistema seja ajustado para modelos de inversor com

especificações diferentes. O campo “UpdateStatus” da Figura 5 dá a segunda função da página: permite ao usuário saber se o comando foi ou não efetivado pelo inversor, sendo atualizado para “Applied” assim que o ST-One responsável pelo inversor escolhido mandar um *POST* com os valores efetivamente em uso.

Figura 4 – Lista de inversores no painel do Django Admin



Fonte: Autoria própria (2020).

Figura 5 – Diagrama de sequência para o envio de comandos aos inversores

Campus:	<input type="text" value="ct"/>
Nome:	<input type="text" value="mon1"/>
Fp:	<input type="text" value="0,91"/>
FpTipo:	<input type="text" value="Inductive"/>
FpMin:	<input type="text" value="0,8"/>
FpMax:	<input type="text" value="1,00"/>
LimPot:	<input type="text" value="95"/>
UpdateStatus:	<input type="text" value="Applied"/>
UpdateTime:	Data: <input type="text" value="31/07/2020"/> Hoje
	Hora: <input type="text" value="05:31:30"/> Agora

Fonte: Autoria própria (2020).

CONCLUSÕES

O sistema projetado cumpre sua função, permitindo envio remoto de comandos a inversores localizados em seis cidades do estado do Paraná, sendo apenas necessário que o usuário administrador possua conexão com a Internet

AGRADECIMENTOS

Os autores agradecem a UTFPR pelo apoio e infraestrutura disponibilizada para o desenvolvimento desta pesquisa e a COPEL-Distribuição pelo apoio e financiamento dos recursos para realização deste projeto de P&D "ANEEL PD 2866-0464/2017 - Metodologia Para Análise, Monitoramento e Gerenciamento da GD por Fontes Incentivadas".

REFERÊNCIAS

LABORATÓRIO DE ENERGIA SOLAR. **Rede de Estações de Pesquisa em Energia Solar**. LABENS, 2019. Disponível em: <https://labens.ct.utfpr.edu.br/sobre/rede-de-estacoes-de-pesquisa-em-energia-solar/>. Acesso em 1 de setembro de 2020.

NHS SOLAR. **Inversor NHS On Grid SOLAR GMSR1**. NHS solar, 2018. Disponível em: <https://www.nhssolar.com.br>. Acesso em 1 de setembro de 2020.

SMART-TECH. **ST-ONE**. ST-One, 2018. Disponível em: <https://st-one.io/landing>. Acesso em 1 de setembro de 2020.

IBM. **Node-RED v1.0**. Node-RED, 2019. Disponível em: <https://nodered.org/about/>. Acesso em 1 de setembro de 2020.

HIPP, R. D. **SQLite v3.27.0**. SQLite, 2019. Disponível em: <https://www.sqlite.org/index.html>. Acesso em 1 de setembro de 2020.

PYTHON SOFTWARE FOUNDATION. **Python v3.8.5**. Python, 2020. Disponível em: <https://www.python.org/doc/>. Acesso em 1 de setembro de 2020.

DJANGO SOFTWARE FOUNDATION. **Django v3.1.0**. Django, 2020. Disponível em: <https://djangoproject.com>. Acesso em 3 de setembro de 2020.