

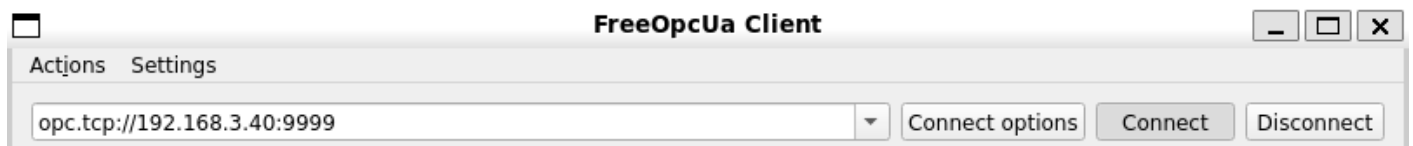
# Visualisation et traitement des données de la Ligne

Une fois les données stockées on va vouloir maintenant les visualiser.

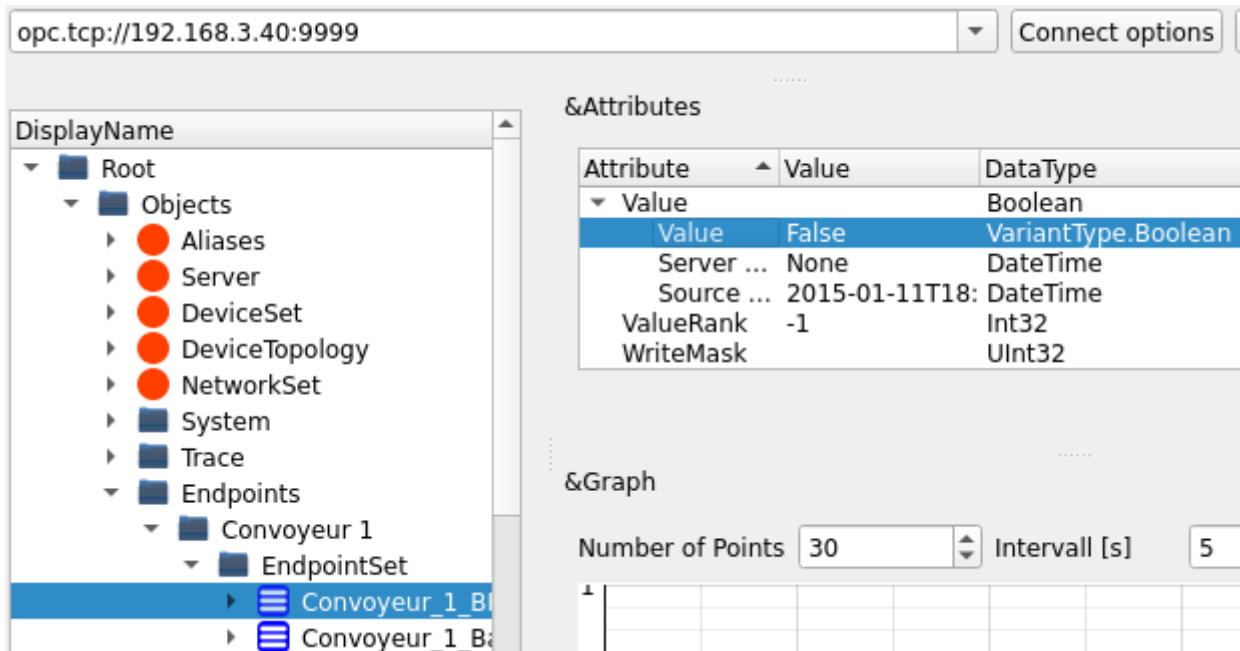
## OPC-UA

Il est possible de visualiser les données concentrées sur l'IoT Gateway via le protocole OPC-UA.

- Installer le logiciel client graphique FreeOPC-UA GUI <https://github.com/FreeOpcUa/opcua-client-gui>
  - Sur Linux (VM WSL Ubuntu) dans un terminal taper `pip3 install opcua-client`, lancer avec `opcua-client`
  - Sur Windows
    - télécharger et extraire [Winpython](#) par ex. vers `C:\Users\user\Downloads\WPy64-31241`
    - Lancer le terminal WinPython `YOUR_INSTALL_PATH\WinPython Terminal.exe`
    - m à j pip `python.exe -m pip install --upgrade pip`
    - installer avec `pip install opcua-client`
    - lancer avec `opcua-client.exe`
- Lancer FreeOPC-UA et se connecter à l'IoT Gateway sur le port OPC-UA
  - `opc.tcp://192.168.3.40:9999` (ou le serveur de lien avec Mappsy `opc.tcp://192.168.3.40:4840`)
  - Cliquer sur Connect



- Inspecter les variables de la ligne
- Dans le panneau de droite, observer les variables changer d'état



# Grafana

Un des outils que nous pouvons utiliser pour visualiser les données est un serveur grafana qui est hébergé sur l'IOT gateway. Pour utiliser Grafana, il faut tout d'abord ajouter les bases de données influxdb pour pouvoir importer et afficher les données. Voici un exemple de configuration de BDD dans Grafana :

Name  Default ☐

Query Language  
InfluxQL

HTTP

URL

Access  [Help >](#)

Allowed cookies

Timeout

Auth

Basic auth ☐ With Credentials ☐

TLS Client Auth ☐ With CA Cert ☐

Skip TLS Verify ☒

Forward OAuth Identity ☐

Custom HTTP Headers  
[+ Add header](#)

InfluxDB Details

Database Access  
Setting the database for this datasource does not deny access to other databases. The InfluxDB query syntax allows switching the database in the query. For example: `SHOW MEASUREMENTS ON „internal“ OR SELECT * FROM „„internal“..“database“ LIMIT 10`  
To support data isolation and security, make sure appropriate permissions are configured in InfluxDB.

Database

User

Password  [Reset](#)

HTTP Method

Max time interval

Max series

[Back](#) [Explore](#) [Delete](#) [Save & test](#)

Les BDD ont déjà été importé dans grafana.

## Dashboard

Voilà un exemple de Dashboard permettant d'afficher les données



## IoT Dashboard

# Node-RED

Intégration des Andons, boutons tactiles et traitement de données

## Interconnexion données Bosch-HLP

### Serveur OPC-UA sous Node-RED

Dans le cadre de l'intégration de la Ligne connectée Bosch Rexroth avec le MES HLP Mappsy, le serveur Node-RED mis en place par Théo Kielwasser et Bosch pour le traitement de données a été augmenté d'un Flow supplémentaire. Ce Flow créé un serveur OPC-UA sur le port standard `192.168.3.40:4840`. Le serveur OPC-UA de l'IoT Gateway lui, est sur le port `192.168.3.40:9999`. Cela permet de formater/convertir les données Bosch pour qu'elles correspondent au "format standard" HLP/Mappsy.

Pour visualiser ces données il suffit d'utiliser FreeOPC-UA pour se connecter à

`opc.tcp://192.168.3.40:4840` (adapter les instructions ci-dessus

<https://innovation.iha.unistra.fr/books/ligne-flexible-connectee-4h/page/visualisation-et-traitement-des-donnees-de-la-ligne#bkmrk-opc-ua> ).

Le principe d'interconnexion est simple. Pour chaque poste, on publie une variable booléenne OPC-UA `PIECES_COUNTER_N` dans `Root/Objects/postes`. Cette variable est `true` lorsqu'il n'y a pas de palet au poste `N` et `false` lorsqu'un palet est présent. Lorsqu'un palet quitte le poste la variable change donc de l'état `false` à `true`.

Le boîtier HLP i4 se connecte au serveur Bosch via le réseau local de la ligne via son interface ETH2 (câble rose connecté au switch du poste de préparation). Il souscrit au serveur OPC-UA

`192.168.3.40:4840`. Le boîtier est connecté au cloud HLP via son interface ETH1 et le routeur de la ligne (câble gris connecté à la prise B0TP11-29 en bas du poteau, prise sans étiquette).



Chaque fois que la variable passe à true, HLP incrémente une variable de comptage dans l'infrastructure de données de la ligne sur son cloud.

opc.tcp://192.168.3.40:4840

DisplayName	BrowseName	NodeId
Root	0:Root	i=84
Objects	0:Objects	i=85
Aliases	0:Aliases	i=23470
Server	0:Server	i=2253
VendorName	1:VendorName	ns=1:s=VendorName
postes	1:postes	ns=1:s=postes
PIECES_COUNTER_1	3:PIECES_COUNTER_1	ns=3:s=PIECES_COUNTER_1
PIECES_COUNTER_2	3:PIECES_COUNTER_2	ns=3:s=PIECES_COUNTER_2
PIECES_COUNTER_3	3:PIECES_COUNTER_3	ns=3:s=PIECES_COUNTER_3
PIECES_COUNTER_4	3:PIECES_COUNTER_4	ns=3:s=PIECES_COUNTER_4
PIECES_COUNTER_5	3:PIECES_COUNTER_5	ns=3:s=PIECES_COUNTER_5
andon_poste_1	3:andon_poste_1	ns=3:s=andon_poste_1
andon_poste_2	3:andon_poste_2	ns=3:s=andon_poste_2
andon_poste_3	3:andon_poste_3	ns=3:s=andon_poste_3

Attribute	Value	DataType
AccessLevel	CurrentRead,	Byte
AccessLevelE	3	UInt32
ArrayDimens	None	UInt32
BrowseName	3:PIECES_COUN	QualifiedName
DataType	Boolean	NodeId
Description	LocalizedText(Lo	LocalizedText
DisplayName	LocalizedText(Lo	LocalizedText
Historizing	False	Boolean
MinimumSam	0.0	Double
NodeClass	2	Int32
NodeId	ns=3:s=PIECES	NodeId
UserAccess	CurrentRead,	Byte
UserWriteMa	3	UInt32
Value	False	Boolean
VariantType	Boolean	Boolean

Flow de stockage dans des variables globales

Flow de publication des variables globales sur le l'OPC-UA 4840

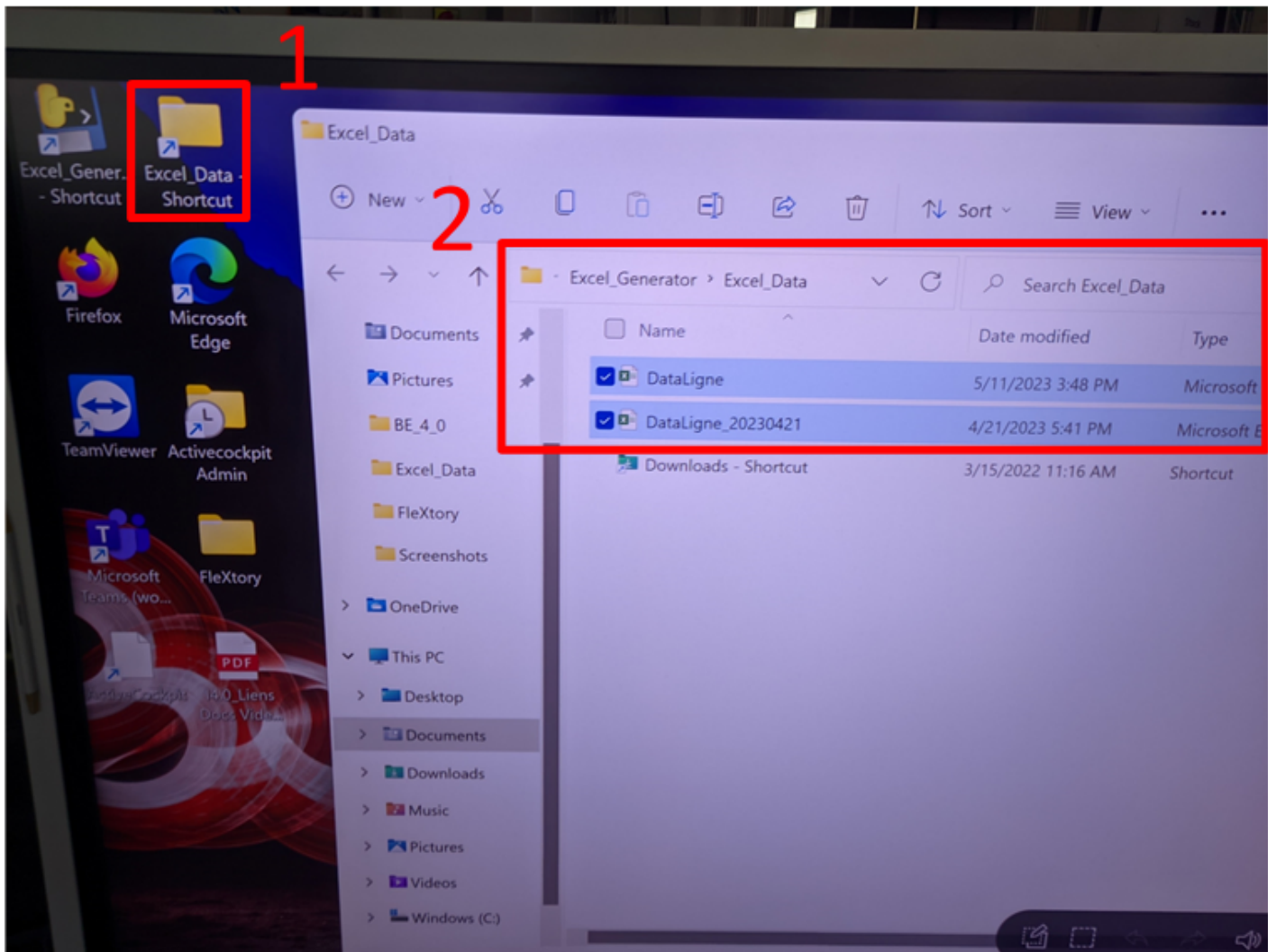
Sources :

- Tuto : <https://flowfuse.com/blog/2023/07/how-to-deploy-a-basic-opc-ua-server-in-node-red/>
- Serveur OPC-UA dans serveur Node-RED (Flow) : <https://flows.nodered.org/node/node-red-contrib-opcua-server>
- Exemple de flow : <https://github.com/BiancoRoyal/node-red-contrib-opcua-server/blob/master/examples/server-with-context.json>
- Upstream : <https://flows.nodered.org/node/node-red-contrib-opcua>
- A voir ? <https://support.elsist.biz/fr/articoli/opc-ua-client-con-node-red/>

# Export des données dans un tableur







Revision #9

Created 11 May 2023 14:07:21 by admin\_idf

Updated 12 March 2025 10:51:34 by admin\_idf