

# OPC UA – Python GUI Client

## Intro

Il client è stato realizzato sfruttando le seguenti API OPC UA in Python:

<https://github.com/FreeOpcUa/python-opcua>.

Per lo sviluppo dell'interfaccia grafica si è fatto uso del framework PyQt5:

<https://www.qt.io/qt-for-python>.

Il Client offre la possibilità di visualizzare tutti gli endpoint offerti da un server OPC UA e connettersi ad esso in maniera sicura, tramite l'utilizzo di certificati che possono essere importati o generati attraverso l'applicazione stessa.

Dal Client è possibile navigare l'Address Space del server a cui ci si è connessi, visualizzando per ogni nodo in esso contenuto attributi e references.

È inoltre possibile creare delle sottoscrizioni e aggiungervi dei monitored items per ricevere aggiornamenti periodici di variabili di interesse.

The screenshot displays the OPC UA Client application window. The interface is divided into several panes:

- Address Space:** A tree view on the left showing the hierarchy of the OPC UA server. The 'Level' node is selected.
- Subscriptions:** A table in the center showing a list of monitored items. The table has columns: ObjectName, VariableName, Value, and Timestamp.
- Attributes:** A table on the right showing the attributes of the selected 'Level' node. The table has columns: Attribute, Value, and DataType.
- References:** A table at the bottom right showing the references of the selected 'Level' node. The table has columns: ReferenceType, NodeId, BrowseName, and TypeDefinition.

ObjectName	VariableName	Value	Timestamp
Boiler	Operating Pressure	100.0	2020-06-19T17:25:43.094827
LevelIndicator	Level	67.313533456073	2020-06-19T17:32:38.192517
Motor#1	Power	4137.441056660541	2020-06-19T17:32:38.192391
Motor#1	Working Voltage	24.0	2020-06-19T17:31:03.611983
TempSensor#1	Temperature	118.94507835880398	2020-06-19T17:32:38.192469
TempSensor#1	Minimum Voltage	2.1	2020-06-19T17:31:31.133953
TempSensor#1	Maximum Voltage	3.6	2020-06-19T17:31:36.011367
Valve	Pressure	0.3283911446245462	2020-06-19T17:32:38.192426
Valve	Exit Voltage	330.0	2020-06-19T17:31:13.429538

Attribute	Value	DataType
AccessLevel	CurrentRead, CurrentWrite	Byte
ArrayDimensions	[]	UInt32
BrowseName	Level	QualifiedName
DataType	Double	NodeId
Description	Level	LocalizedText
DisplayName	Level	LocalizedText
Historizing	False	Boolean
MinimumSamplingInterval	0.0	Double
NodeClass	Variable	Int32
NodeId	ns=1;i=1027	NodeId
UserAccessLevel	CurrentRead, CurrentWrite	Byte

ReferenceType	NodeId	BrowseName	TypeDefinition
HasTypeDefinition	ns=1;i=1028	0:AnalogItemTy...	Null
HasProperty	ns=1;i=1029	0:EngineeringU...	PropertyTy...

### Nota:

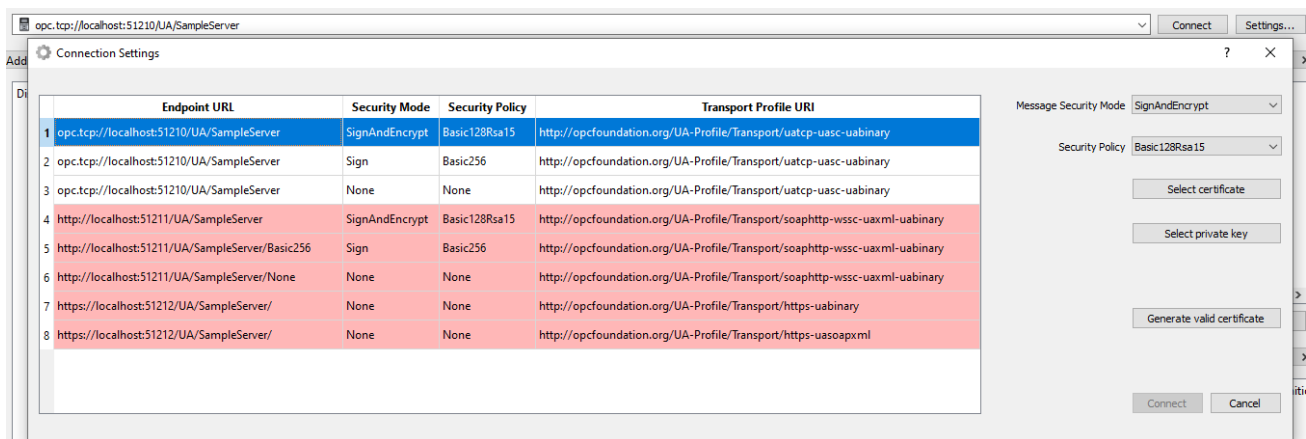
Per avviare il server lanciare dalla directory server il comando `npm install` (se non lo si è fatto prima), e successivamente `npm start`. Per avviare il client bisogna prima installare i requirements attraverso il comando `pip install -r requirements.txt`, e poi lanciare dalla directory client il comando `python mainwindow.py`.

## Connessione

Nella barra in alto è possibile inserire l'URI del server a cui ci si vuole collegare.

Cliccando sul bottone Connect ci si collega utilizzando le ultime impostazioni memorizzate associate all'URI in questione mentre, cliccando su Settings..., si apre una finestra che mostra tutti i possibili Session Endpoint offerti dal server. Quelli non supportati dal client vengono visualizzati in rosso e non sono selezionabili.

Se necessario, è possibile selezionare un certificato o generarne uno nuovo valido per l'applicazione, cliccando sull'apposito bottone, per la creazione del canale sicuro.



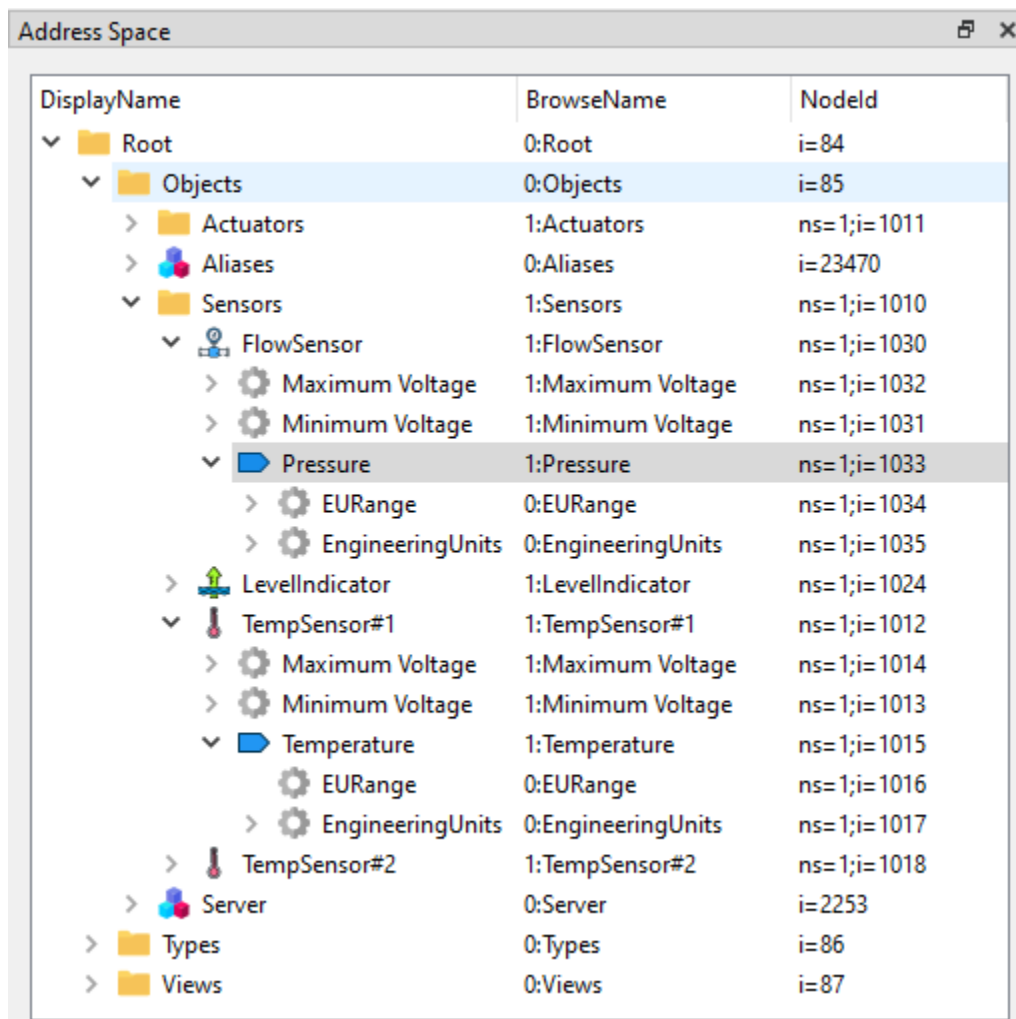
In qualsiasi momento è possibile disconnettersi dal server mediante l'apposito bottone.

## Address Space

Una volta connessi è possibile esplorare l'Address Space del server tramite una visualizzazione ad albero.

Per ogni nodo è possibile visualizzare display name, browse name, namespace index e node id.

Attraverso questa vista è possibile effettuare la browse sui diversi nodi.



DisplayName	BrowseName	NodeId
Root	0:Root	i=84
Objects	0:Objects	i=85
Actuators	1:Actuators	ns= 1;i= 1011
Aliases	0:Aliases	i=23470
Sensors	1:Sensors	ns= 1;i= 1010
FlowSensor	1:FlowSensor	ns= 1;i= 1030
Maximum Voltage	1:Maximum Voltage	ns= 1;i= 1032
Minimum Voltage	1:Minimum Voltage	ns= 1;i= 1031
Pressure	1:Pressure	ns= 1;i= 1033
EURange	0:EURange	ns= 1;i= 1034
EngineeringUnits	0:EngineeringUnits	ns= 1;i= 1035
LevelIndicator	1:LevelIndicator	ns= 1;i= 1024
TempSensor#1	1:TempSensor#1	ns= 1;i= 1012
Maximum Voltage	1:Maximum Voltage	ns= 1;i= 1014
Minimum Voltage	1:Minimum Voltage	ns= 1;i= 1013
Temperature	1:Temperature	ns= 1;i= 1015
EURange	0:EURange	ns= 1;i= 1016
EngineeringUnits	0:EngineeringUnits	ns= 1;i= 1017
TempSensor#2	1:TempSensor#2	ns= 1;i= 1018
Server	0:Server	i=2253
Types	0:Types	i=86
Views	0:Views	i=87

## Attributi e References

Ogni volta che si seleziona un nodo nell'address space vengono visualizzate tutte le sue informazioni: attributi e references.

Se il nodo selezionato appartiene alla Node Class Variable, ovvero è una Data Variable o una Property, il suo valore viene colorato sulla base del suo status code:

- Good / Success
- Uncertain / Warning
- Bad / Failure

L'utente ha anche la possibilità di modificare il valore di ogni singolo attributo, se consentito dal server.

Le references visualizzate sono quelle uscenti dal nodo.

The screenshot shows two windows from a software application. The top window, titled 'Attributes', contains a table with three columns: 'Attribute', 'Value', and 'DataType'. The bottom window, titled 'References', contains a table with four columns: 'ReferenceType', 'NodeId', 'BrowseName', and 'TypeDefinition'.

Attribute	Value	DataType
AccessLevel	CurrentRead, CurrentWrite	Byte
ArrayDimensions	[]	UInt32
BrowseName	1:Exit Speed	QualifiedName
DataType	Double	NodeId
Description		LocalizedText
DisplayName	Exit Speed	LocalizedText
Historizing	False	Boolean
MinimumSamplingInter	0.0	Double
NodeClass	Variable	Int32
NodeId	ns= 1;i= 1037	NodeId
UserAccessLevel	CurrentRead, CurrentWrite	Byte
UserWriteMask		UInt32
Value		Double
Value	6500.0	Double
Server Timestamp	None	DateTime
Source Timestamp	2020-06-09T13:52:38.981914	DateTime
ValueRank	Scalar	Int32
WriteMask		UInt32

Refresh

ReferenceType	NodeId	BrowseName	TypeDefinition
1 HasTypeDefiniti...	PropertyType	0:PropertyType	Null

## Subscriptions e Monitored Items

Il Client fornisce la possibilità di effettuare diverse sottoscrizioni, e di settare per ciascuna di esse i parametri previsti dallo standard.

L'utente può inoltre creare dei monitored items, settando anche in questo caso i relativi parametri, e associarli a una data sottoscrizione.

Subscription Settings

Requested Publishing Interval

500

Requested Max Keep-Alive Count

3000

Requested LifeTime Count

10000

Max Notifications per Publish

10000

Priority

4

Create

Cancel

Monitored Item Settings

Sampling Interval

500

Queue Size

0

Discard Oldest

☒

☒ Data Change Filter

Data Change Trigger

Status/Value

Deadband Type

Percent

Deadband Value

20,00

Add

Cancel

Subscriptions				
Sub1 Sub2 Sub3 Sub4 +				
	ObjectName	VariableName	Value	Timestamp
1	Boiler	Operating Pressure	100.0	2020-06-19T17:25:43.094827
2	FlowSensor	Pressure	1.1733629292605023	2020-06-19T17:28:32.531281
3	FlowSensor	Minimum Voltage	4.5	2020-06-19T17:26:04.275688
4	FlowSensor	Maximum Voltage	24.0	2020-06-19T17:26:08.745144
5	LevelIndicator	Level	58.66551815103618	2020-06-19T17:28:32.531393
6	TempSensor#2	Temperature	5.628781942812662	2020-06-19T17:28:32.531351