Python GUI Widget OPC UA Client

**Intro**

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

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

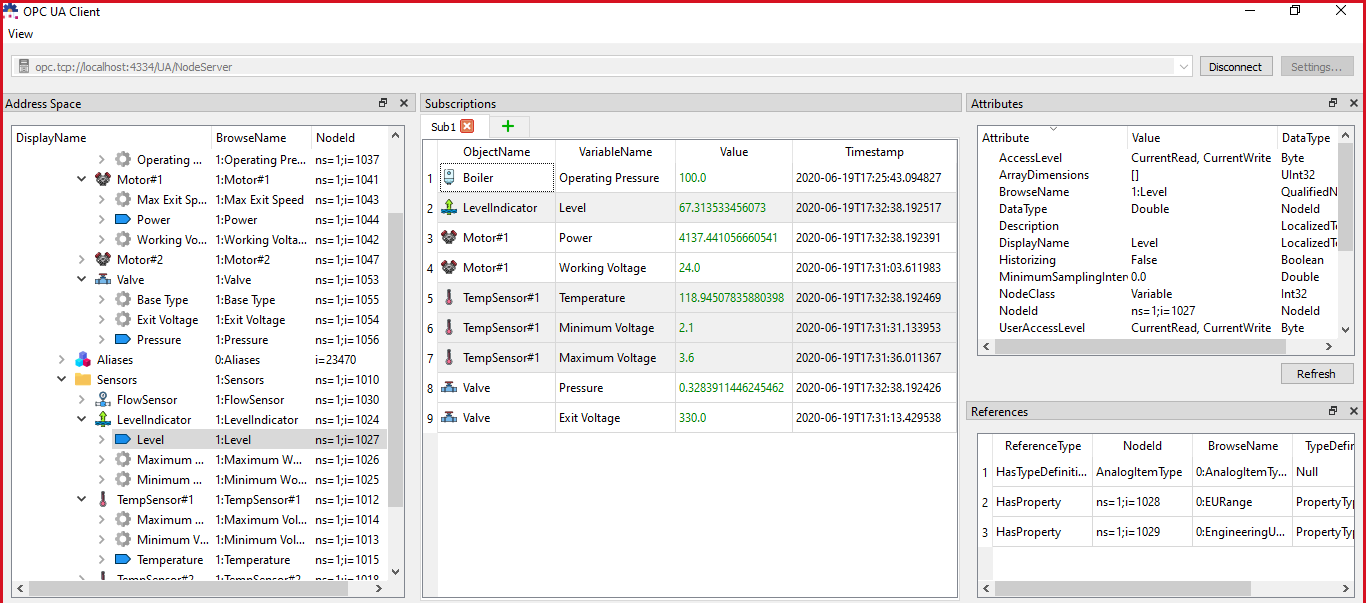
Per lo sviluppo dell’interfaccia grafica si è fatto uso del framework PyQt5:

<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.

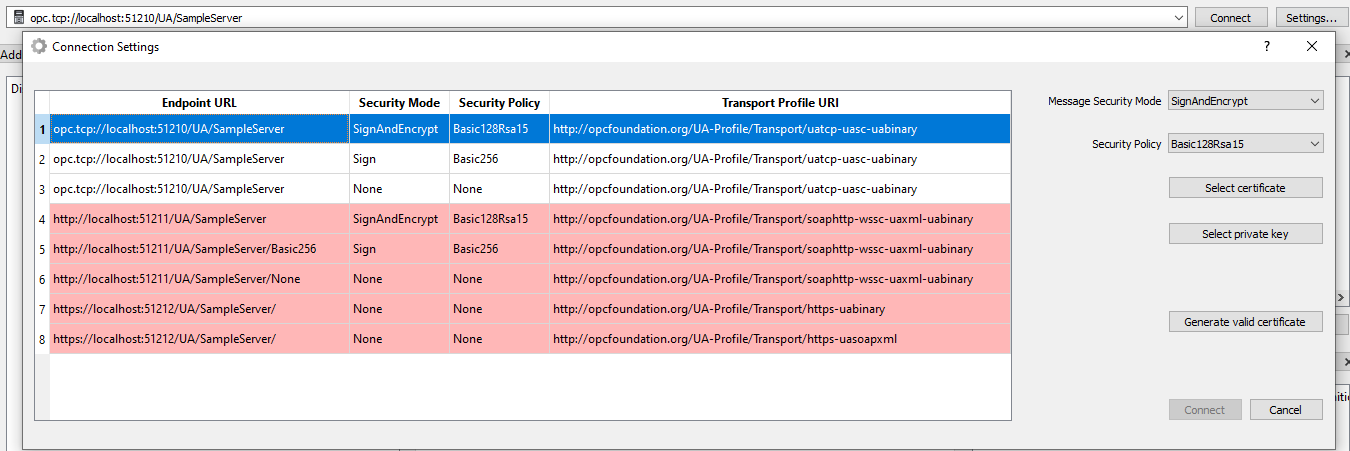
****

**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.**

****

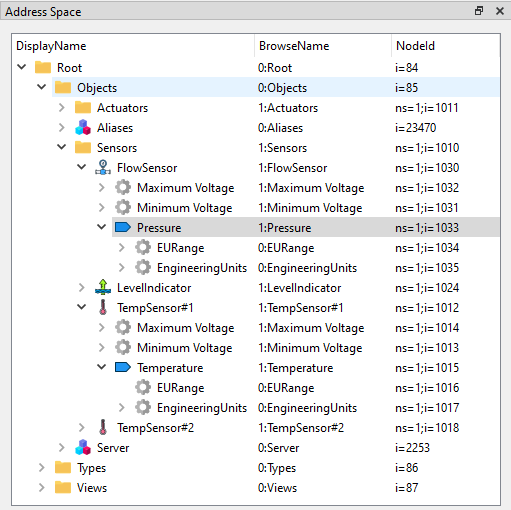
**In qualsiasi momento è possibile disconnettersi dal server mediante l’apposito bottone.**

**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.**

****

**Attributi e Reference**

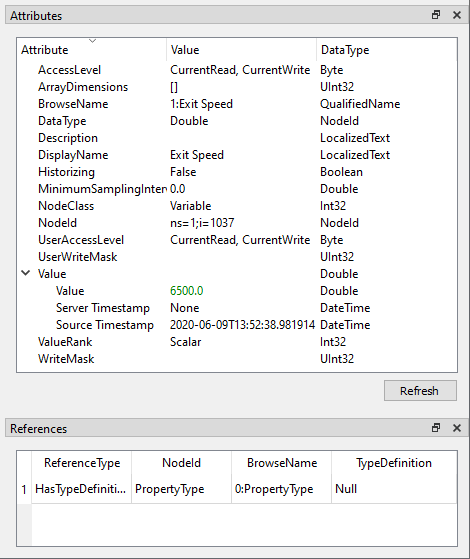
**Ogni volta che si seleziona un nodo nell’address space vengono visualizzate tutte le sue informazioni: attributi e reference.**

**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:**

* **Verde → Good / Success**
* **Giallo → Uncertain / Warning**
* **Rosso → Bad / Failure**

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

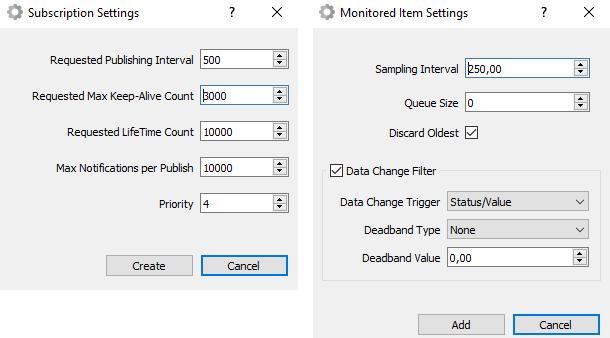
**Le reference visualizzate sono quelle uscenti dal nodo.**

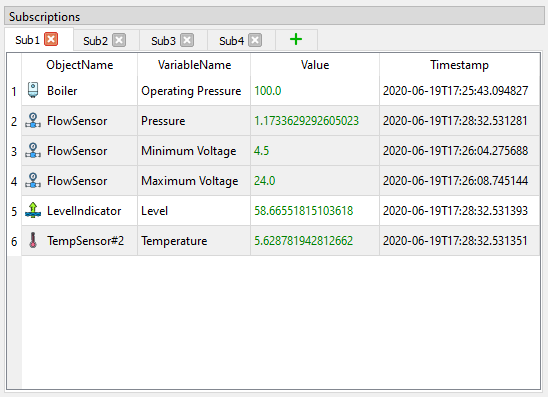
****

**Subscriptions and 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.**

****

****

**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 requiremements.txt*, e poi lanciare dalla directory client il comando *python mainwindow.py* .**