

# RELAZIONE FINALE ELABORATO INDUSTRIAL INFORMATICS

Arena Gianluca - Lo Castro Alex

Il nostro elaborato finale, disponibile su GitHub all'indirizzo <https://github.com/ykn18/industrial-informatics-opc-ua-python>, consiste nello sviluppo di un'applicazione formata da client e server che comunicano attraverso il protocollo OPC UA. Entrambi i componenti sono stati sviluppati in Python utilizzando la libreria FreeOPCUA, disponibile all'indirizzo <https://github.com/FreeOpcUa/python-opcua>. Nelle sezioni successive spieghiamo in cosa consiste l'applicazione e come utilizzare la GUI.

## OPC-SERVER

Il server gira su un Raspberry Pi 3 B+ a cui sono collegati un LED e un bottone. La struttura del server, al file `opc-server/server.py`, è piuttosto minimale.

Dopo aver creato un'istanza del server OPC UA, settiamo il suo URL e decidiamo quali endpoint esporre in base alla sicurezza supportata. Nel nostro caso i meccanismi di sicurezza supportati sono "None", "Sign" con `SecurityPolicyType "Basic256Sha256_Sign"` e "SignAndEncrypt" `SecurityPolicyType "Basic256Sha256_Sign"`. Dopo aver caricato i certificati, generati in OpenSSL e self-signed, creiamo il nostro namespace e aggiungiamo un nodo Object che conterrà le variabili OPC UA che rappresentano il bottone e il led.

A questo punto istanziamo anche le due variabili che rappresentano bottone e LED reali ed entriamo nel loop principale.

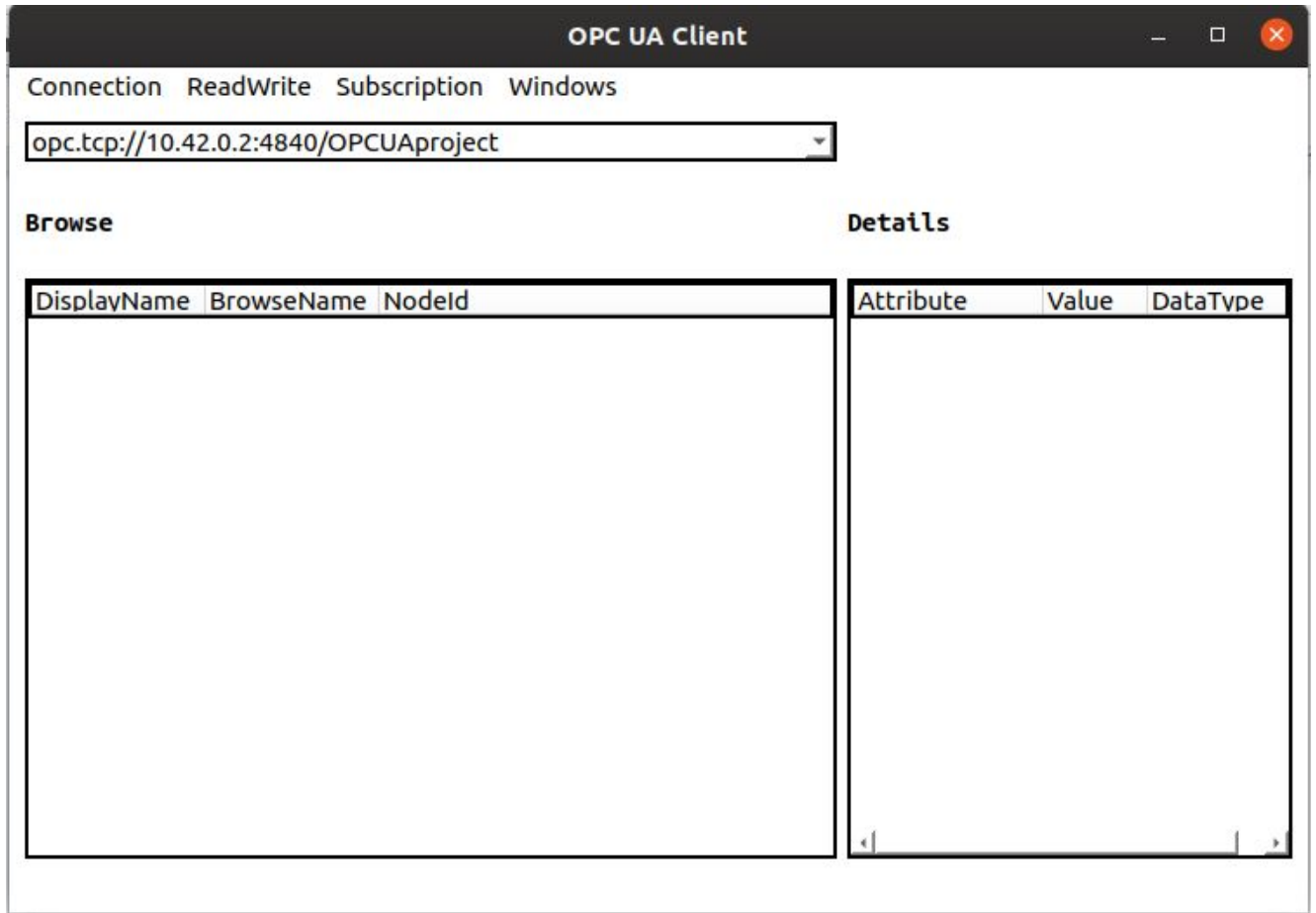
Ogni 100ms il server:

- Controlla il valore della variabile LED nell'address space e accende/spegne il LED reale di conseguenza;
- Controlla se il bottone reale è premuto o meno e aggiorna la variabile button nell'address space di conseguenza.

Oltre che con il nostro client, il server è stato testato con Prosys OPC UA Browser e Unified Automation UaExpert.

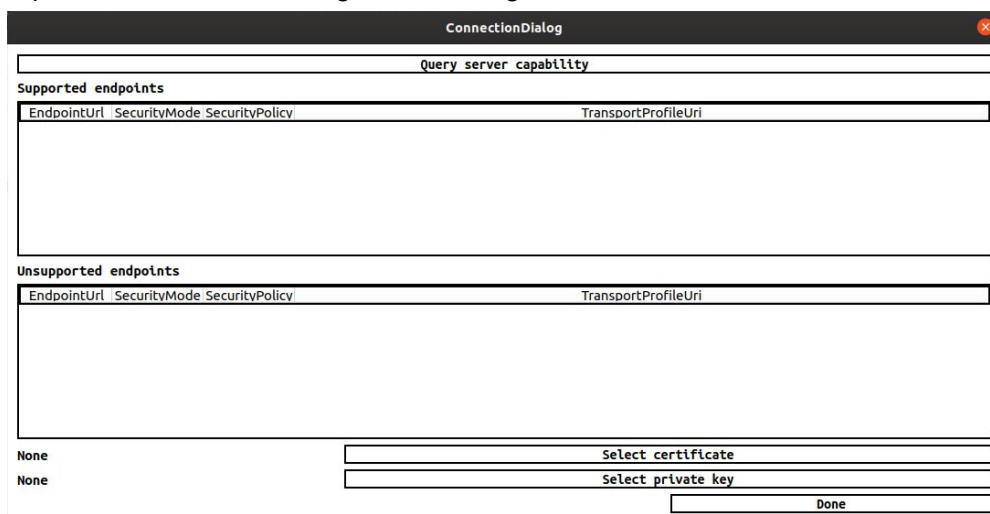
## OPC-CLIENT

La GUI del client opc è realizzata usando PyQt5. Per lanciare l'applicazione bisogna andare nella cartella `opc-client` e digitare da terminale il comando Python per eseguire il file `app.py`. La schermata principale si mostra come nella seguente figura:



Dopo aver inserito l'url del server, nella sezione Connection della menuBar è possibile:

- Chiedere gli endpoints disponibili nel server
- Selezionare l'endpoint e caricare il certificato e la chiave qualora si scelga un endpoint che include meccanismi di sicurezza. La finestra in cui si possono fare queste operazioni si presenta come nella seguente immagine:



Cliccando su Query server capability, viene fatta una get\_endpoints() al server.

Gli endpoints sono suddivisi in supported endpoint, ovvero selezionabili in quanto supportati dalla libreria FreeOpcUA, e unsupported.

Questa suddivisione è dovuta al fatto che la libreria FreeOpcUA permette l'uso esclusivo della codifica binaria e a livello di trasporto supporta TCP.

- Connettersi all'endpoint scelto
- Disconnettersi

Appena viene stabilita la connessione, nella sezione Browse, è possibile navigare nell'address space del server.

The screenshot shows the OPC UA Client window with the connection URL set to `opc.tcp://localhost:4840/OPCUAproject`. The **Browse** section displays a tree view of the address space. The **Details** section shows the attributes of the selected node, `MyVariable`.

DisplayName	BrowseName	NodeId
Root	0:Root	i=84
Objects	0:Objects	i=85
Server	0:Server	i=2253
Device0001	2:Device0001	ns=2;i=7
MyObject	2:MyObject	ns=2;i=12
MyStringVariable	0:MyString...	ns=2;g=1be5b...
MyDateTimeVar	2:MyDateTi...	ns=2;i=16
MySin	2:MySin	ns=2;i=14
MyStringVariable	2:MyString...	ns=2;i=15
<b>MyVariable</b>	<b>2:MyVariable</b>	<b>ns=2;i=13</b>
multiply	2:multiply	ns=2;i=23
myStronglyTyp...	2:myStrong...	ns=2;i=18
myarrayvar	2:myarrayvar	ns=2;i=17
mymethod	2:mymethod	ns=2;i=20
myproperty	2:myproperty	ns=2;i=19
myEmptyFolder	2:myEmpty...	ns=2;i=6
Types	0:Types	i=86
Views	0:Views	i=87

Attribute	Value	Data Type
AccessLevel	Curre...	Byte
ArrayDim...	[]	UInt32
BrowseN...	2:My...	QualifiedI
DataType	Double	NodeId
Description	MyVa...	Localized
DisplayNa...	MyVa...	Localized
Historizing	False	Boolean
Minimum...	0.0	Double
NodeClass	Varia...	Int32
NodeId	ns=2;...	NodeId
UserAcce...	Curre...	Byte
UserWrit...		UInt32
Value		Double
Value	9.9	Double
Server ...	None	DateTime
Source...	None	DateTime
ValueRank	Scalar	Int32

Nel momento in cui viene selezionato un nodo nell'address space, nella sezione Details è possibile osservare gli attributi associati a quel nodo.

Per visualizzare le reference del nodo selezionato, bisogna andare nella sezione Windows della menuBar e cliccare su References. Una tabella come nella seguente immagine verrà visualizzata:

The screenshot shows the References window with a table containing one reference entry.

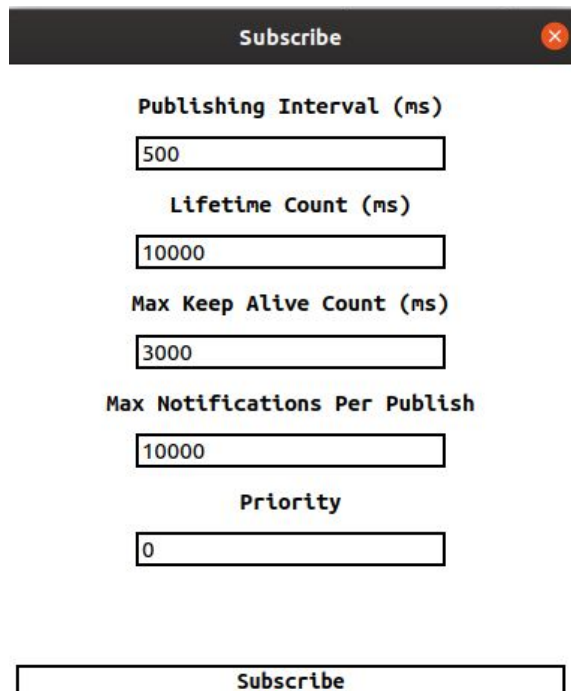
ReferenceType	NodeId	BrowseName	TypeDefin
1 HasTypeDefinition	BaseDataVariableType	0:BaseDataVariableType	Null

Nella sezione ReadWrite della menuBar è possibile:

- Fare una Read sul nodo selezionato. La Read aggiorna la sezione Details della finestra principale. Nel caso in cui il nodo sia un DataVariable viene stampato il valore nella finestra dei logs (visualizzabile andando su Windows->Logs)
- Fare una Write sul nodo selezionato solo se si tratta di un DataVariable node.

Nella sezione Subscription è possibile:

- Creare una Subscription, settando i vari parametri nella finestra.



**Subscribe**

Publishing Interval (ms)  
500

Lifetime Count (ms)  
10000

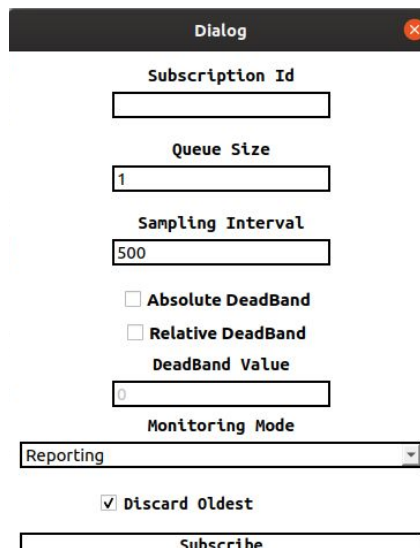
Max Keep Alive Count (ms)  
3000

Max Notifications Per Publish  
10000

Priority  
0

Subscribe

- Eliminare una Subscription. Nel caso in cui la subscription ha associati dei monitored items anch'essi verranno eliminati.
- Aggiungere i monitored items alle varie subscription attraverso la seguente finestra:



**Dialog**

Subscription Id  
[Empty]

Queue Size  
1

Sampling Interval  
500

☐ Absolute DeadBand  
☐ Relative DeadBand

DeadBand Value  
0

Monitoring Mode  
Reporting

☒ Discard Oldest

Subscribe

- Eliminare il monitored item del nodo selezionato.

Per visualizzare le subscriptions create con i relativi monitored items, bisogna cliccare su Windows->Subscriptions.

La schermata visualizzata sarà come nella seguente figura:

MainWindow							
Subscription				Monitored Items			
	Subscription Id	Publishing Interval		DisplayName	Value	Timestamp	Subscription Id
1	78	500		1 MySin	0....	2020-06-12T0...	78

Il client, oltre che col nostro server, è stato testato con i server demo della OPC Foundation e con OPC UA C++ Demo Server della Unified Automation.