

Università degli Studi di Padova
Dipartimento di Tecnica e Gestione dei sistemi industriali (DTG)
Corso di Laurea in Ingegneria Meccatronica

Interoperabilità tra nodi embedded e PLC industriali tramite OPC UA su Ethernet

Anno accademico: 2025/2026

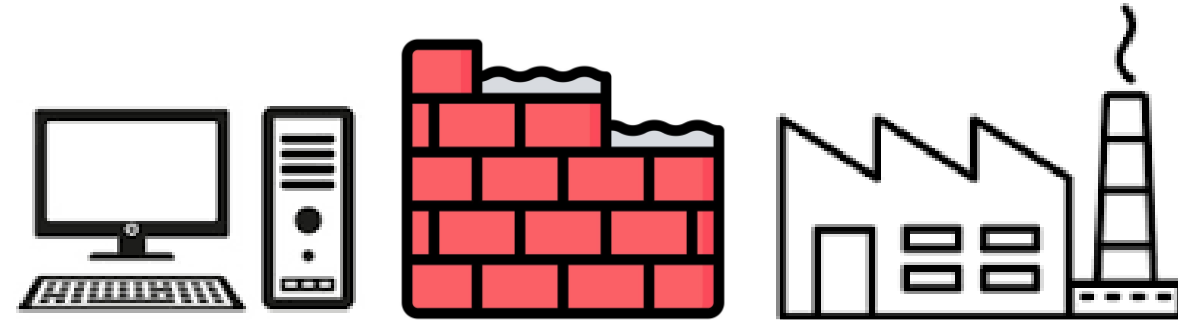


**UNIVERSITÀ
DI PADOVA**

**Filippo Lucietti, Giovanni Milani,
Cesare Pellizzari, Davide Zanellato**



- **Separazione dei livelli:** Mancanza di comunicazione tra livello campo e livello informatico
- **Limiti dei protocolli proprietari:** Spesso costosi, chiusi e legati ad uno specifico fornitore
- **Collo di bottiglia dei dati:** Difficoltà nell'estrarre dati fisici da indirizzare in analisi



OBIETTIVO: Implementazione del **protocollo OPC UA** al fine di consentire **l'interoperabilità tra dispositivi eterogenei**.

COME RAGGIUNGERLO: Configurazione di una **Raspberry Pi** in qualità di **OPC UA Client** e di un **PLC industriale** (Siemens S7-1200) come **OPC UA Server**, collegati direttamente tramite **Ethernet**.

VALIDAZIONE RISULTATI: verifica della corretta **trasmissione dei dati** generati nella scheda utilizzando il software **UaExpert**.



OPC UA (Open Platform Communications Unified Architecture) è uno **standard di comunicazione industriale**

È lo standard fondamentale per far **comunicare macchinari** di produttori **differenti**



Garantisce l'**indipendenza dalla piattaforma**, permettendo a sistemi diversi di dialogare (es. Raspberry Pi e PLC Siemens)

È **affidabile e robusto** grazie a protocolli di sicurezza nello scambio dei dati

Client: Raspberry Pi

- Architettura ARM basata su Linux al fine di garantire massima stabilità.
- Ideale per analisi ed esecuzione dei moduli scritti in codice C.
- Implementazione logica tramite libreria open-source *open62541*.



open62541

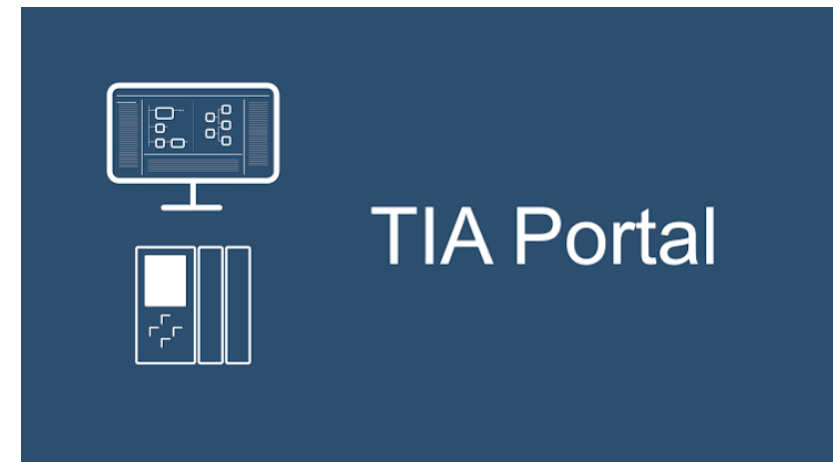
Server: Siemens S7-1200

- PLC compatto che presenta un'interfaccia Profinet integrata.
- Utilizzo di funzioni di diagnostica integrata e protocolli di sicurezza.
- Configurazione di rete e sviluppo attraverso l'ambiente TIA Portal.



TIA Portal permette di programmare il PLC, di creare le interfacce grafiche con cui gli operatori interagiscono con le macchine e di gestire la comunicazione di rete tra i vari componenti

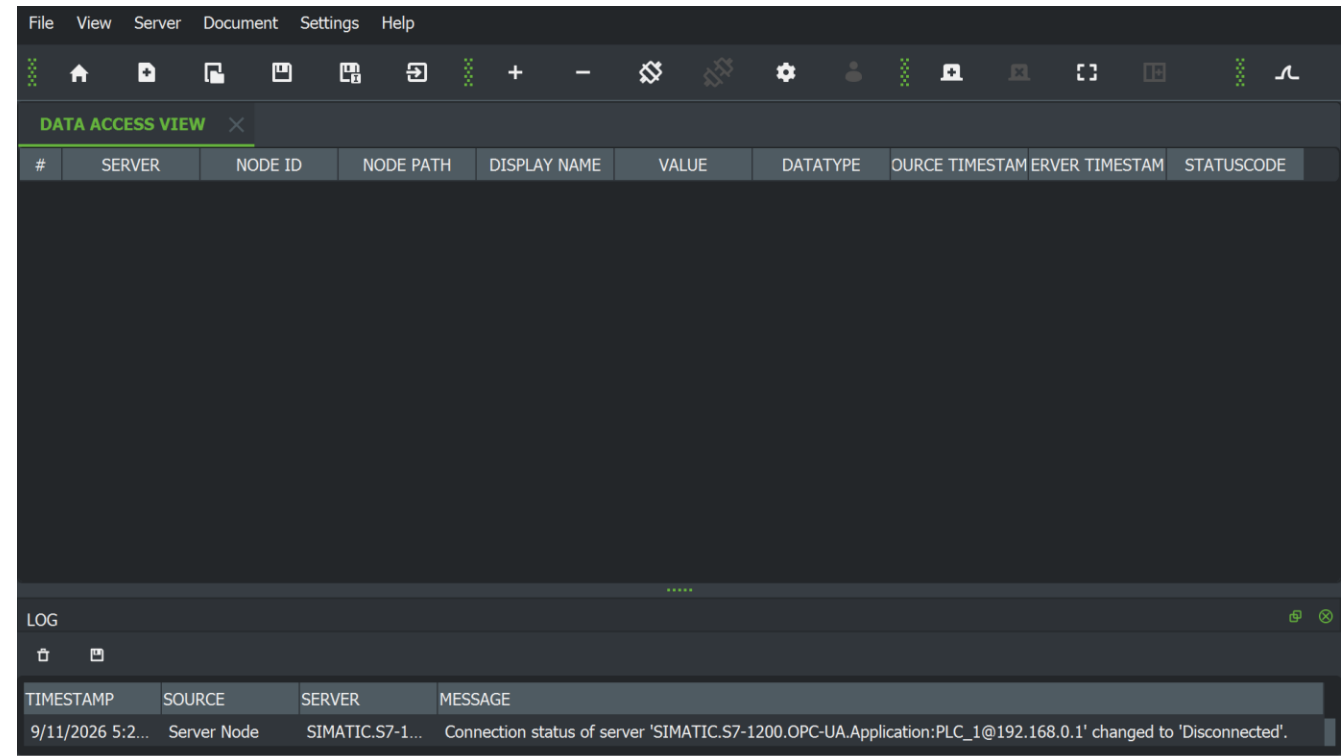
- **Configurazione Hardware:** Inserimento e definizione della CPU nel progetto software.
- **Allineamento Firmware:** Aggiornamento della versione di sistema per sbloccare i pacchetti di comunicazione avanzati.
- **Scopo dell'utilizzo:** Impiegato per configurare i parametri di rete della CPU e l'attivazione del Server OPC UA.

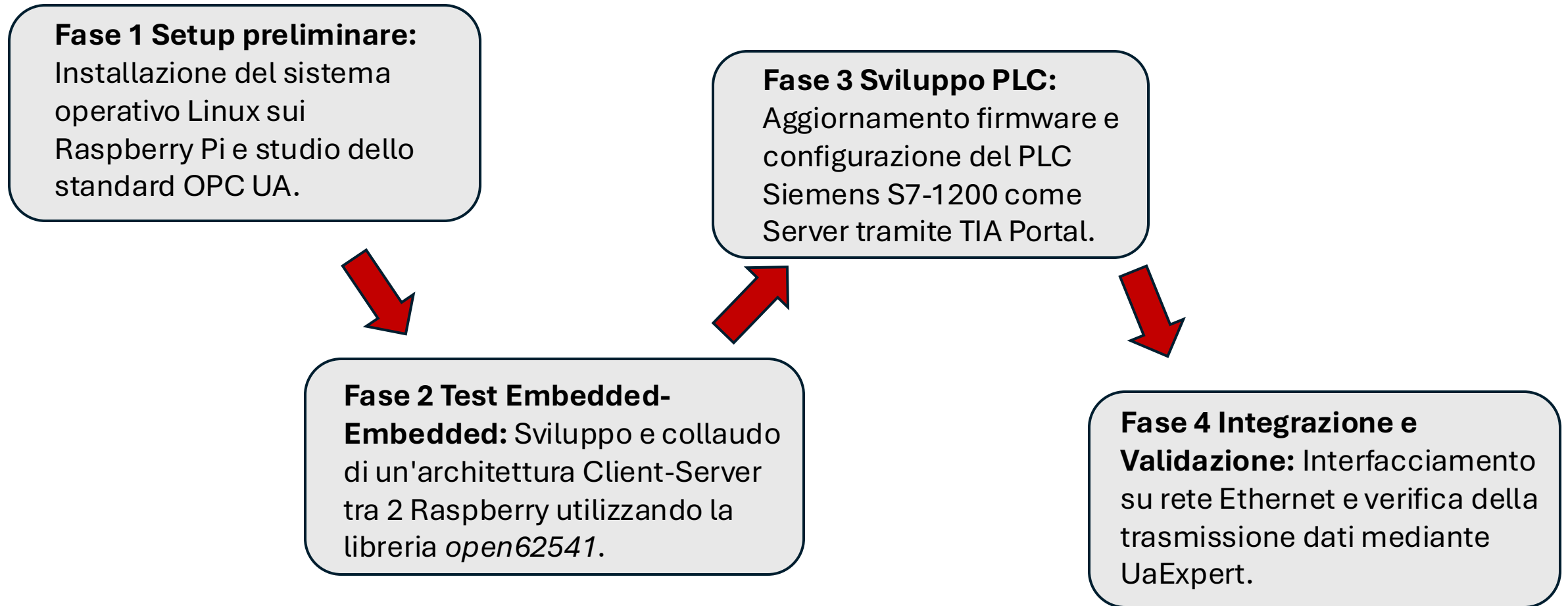


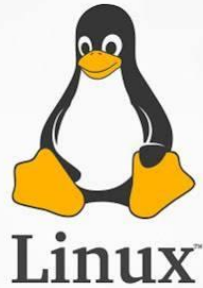
UaExpert è un **client OPC UA** universale e indipendente. Permette di navigare l'Address Space del server collegato.



Utilizzato per **verificare la connettività** del server senza rischiare interferenze date da errori di codice del client e per **visualizzare le variabili** esposte.

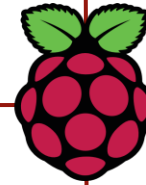






Il sistema operativo
Raspberry Pi OS
installato è
una distribuzione
Linux

La libreria open-source **open62541** è stata
clonata tramite Git



PROGETTO OPC UA

```
├── src
│   ├── myClient.c
│   └── myServer.c
├── open62541
│   ├── include
│   │   └── open62541.h
│   └── src
│       └── open62541.c
├── build
└── CMakeLists.txt
```

**Struttura del
codice:** libreria
separata dal
resto del codice
e CMake

Configurazione dell'interfaccia
di rete **Wi-Fi (WLAN)/Ethernet**
per il collegamento tra le
due schede



```
//creazione del nodo e inizializzazione variabile
UA_VariableAttributes vAttr = UA_VariableAttributes_default;
UA_Double valoreIniziale = 0.0;

//Inserimento valore nel variant
UA_Variant_setScalar(&vAttr.value, &valoreIniziale, &UA_TYPES[UA_TYPES_DOUBLE]);

//nome visualizzato in UaExpert
vAttr.displayName = UA_LOCALIZEDTEXT("en-US", "TEMPERATURA.SENSORE");

//imposta i permessi: nodo leggibile e scrivibile dal client
vAttr.accessLevel = UA_ACCESSLEVELMASK_READ | UA_ACCESSLEVELMASK_WRITE;

//inserimento nodo nel server
UA_Server_addVariableNode(server,
    UA_NODEID_STRING(1, "TEMPERATURA.SENSORE"),
    UA_NODEID_NUMERIC(0, UA_NS0ID_OBJECTSFOLDER),
    UA_NODEID_NUMERIC(0, UA_NS0ID_ORGANIZES),
    UA_QUALIFIEDNAME(1, "TEMPERATURA.SENSORE"),
    UA_NODEID_NUMERIC(0, UA_NS0ID_BASEDATAVARIABLETYPE),
    vAttr, NULL, NULL);

//avvio loop di rete del server
UA_StatusCode retval = UA_Server_run(server, &running);
```

Nodo configurato in lettura e scrittura

Inserimento variabile nel server

Funzione di open62541: avvia un loop che rimane in ascolto sulla rete

```
//definizione nodo di destinazione: ID del nodo (NodeId), parametri
//(indice namespace, stringa identificativa del nodo)
const UA_NodeId nodeId = UA_NODEID_STRING(1, "TEMPERATURA.SENSORE");

/*in caso di server OPC UA su PLC sostituire la riga precedente con:
const UA_NodeId nodeId = UA_NODEID_NUMERIC(4,13);*/

//ciclo lettura temperatura mandata al server ogni secondo
for (int i = 0; i < 50; i++) {
    UA_Double temperature = read_temperature_CPU();

    //Variant con una copia del dato
    UA_Variant value;
    UA_Variant_init(&value);
    UA_Variant_setScalarCopy(&value, &temperature, &UA_TYPES[UA_TYPES_DOUBLE]);

    //scrittura sul server: fino ad ottenimento risposta o scadenza time out
    retval = UA_Client_writeValueAttribute(client, nodeId, &value);

    //libera memoria allocata da setScalarCopy
    UA_Variant_clear(&value);

    sleep(1);
}
```

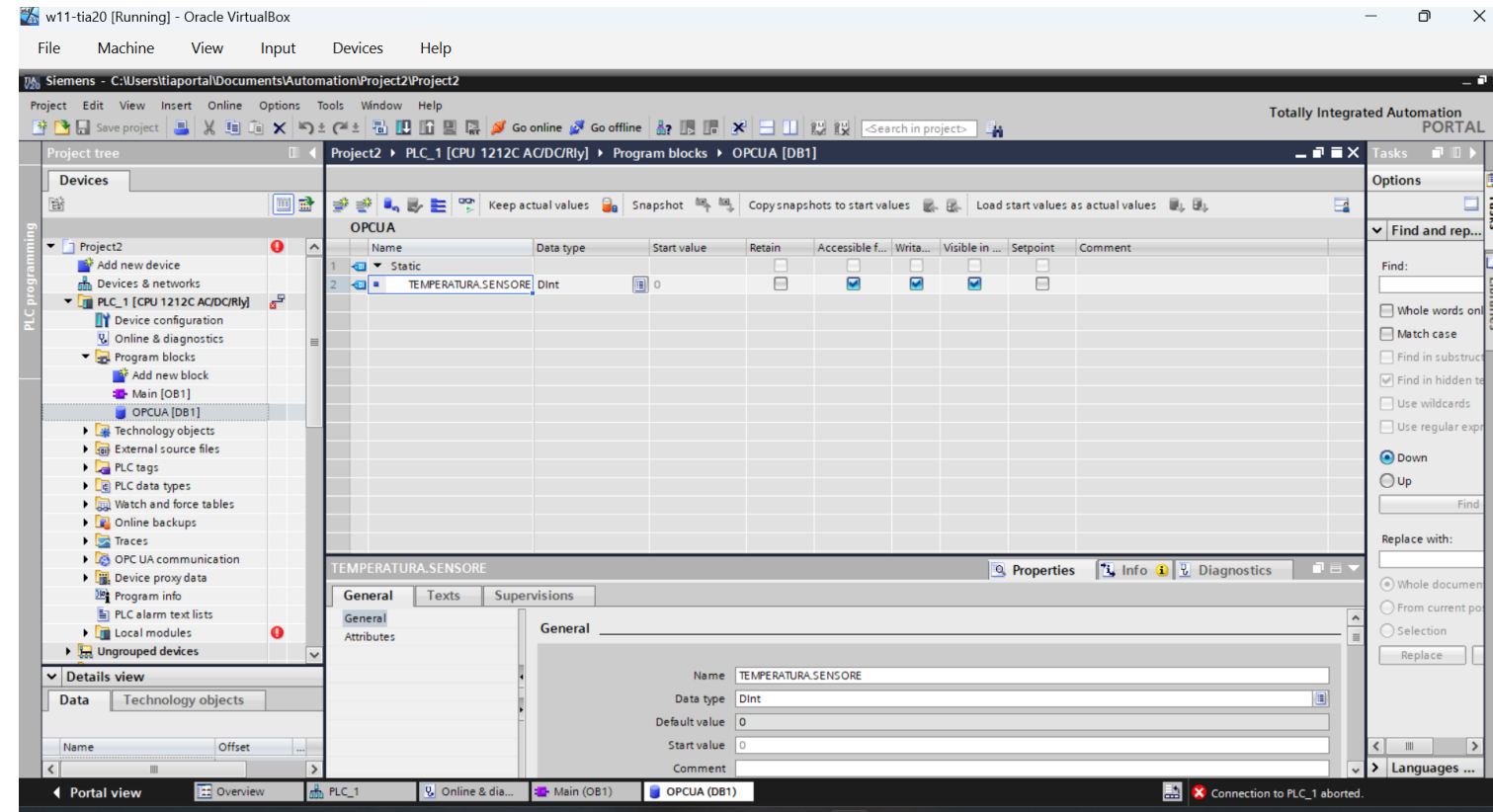
Collegamento al nodo tramite namespace e identificativo

Acquisizione temperatura CPU

Scrittura sul nodo

Periodo di campionamento di 1 secondo

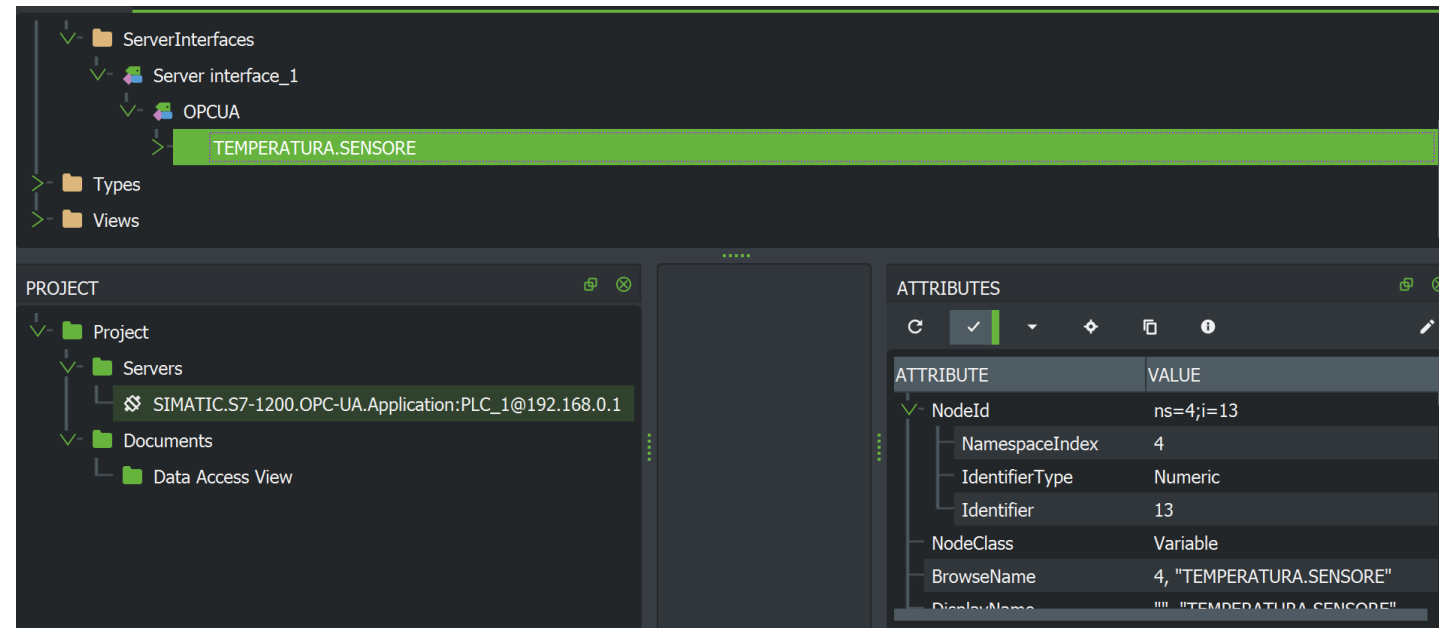
- **Abilitazione Nativa:**
Attivazione del Server OPC UA direttamente dalle proprietà della CPU.
- **Data Block (DB):** Creazione di un blocco dati dove allocare le variabili da esporre in rete.
- **Accessibilità Esterna:**
Configurazione della variabile con permessi di lettura/scrittura per i client in rete.



Connessione del client **UaExpert** al server del PLC.

Navigazione all'interno dell'Address Space per **individuare la variabile** all'interno del Data Block (DB).

Estrazione delle coordinate identificative del nodo
(**Namespace e Identifier**)
necessarie per la configurazione della Raspberry Client.



Collegamento fisico: connessione diretta tra la porta del Raspberry e l'interfaccia profinet del PLC tramite **cavo Ethernet**.

Configurazione di rete: assegnazione di **indirizzi IP** di entrambi i dispositivi all'interno della **stessa sottorete** così da garantire la possibilità di connettersi.

Connessione client: Avvio dell'eseguibile del Client su Linux e monitoraggio tramite log a terminale per confermare il **corretto invio** della temperatura.



Conclusione: Dimostrata con successo l'interoperabilità tra dispositivi eterogenei tramite lo standard OPC UA.

Sviluppo Futuro 1: *Chiusura dell'anello di controllo: generare un comando sul PLC in base al dato di temperatura ricevuto e riportare l'informazione alla Raspberry.*

Sviluppo Futuro 2: Sostituzione della semplice variabile con un sensore reale collegato alla Raspberry.

