**ROMÂNIA**

**MINISTERUL APĂRĂRII NAŢIONALE**

**ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”**

**FACULTATEA DE SISTEME INFORMATICE ȘI SECURITATE CIBERNETICĂ**

**Specializarea: CALCULATOARE ȘI SISTEME INFORMATICE PENTRU APĂRAREA ȘI SECURITATEA NAȚIONALĂ**



**Sistem IOT de management al clădirilor**

CONDUCĂTOR ŞTIINŢIFIC:

Conf. dr. Ing. Laurențiu MĂRGĂRIT

ABSOLVET:

**Stud. Sg. Maj. Calugaru Alina-Diana**

**ABSTRACT**

Automation is a vital component that numerous industries heavily rely upon to boost productivity and minimize expenses. This paper explores the theoretical aspects and presents the necessary steps for the installation and evaluation of OPC UA Technology, a technology widely adopted by the majority of large-scale industries to automate a wide range of processes.

The adoption of OPC UA Technology can result in significant benefits, such as increased production capacity, improved product quality, reduced downtime, and enhanced safety. Furthermore, this technology enables seamless integration between various systems and devices, leading to streamlined communication and data exchange, and ultimately resulting in better decision-making and operational efficiency.

**REZUMAT**

Contents

[LISTA DE FIGURI 5](#_Toc137410818)

[ACRONIME. ABREVIERI. NOTAŢII 6](#_Toc137410819)

[1.Introducere 7](#_Toc137410820)

[1.1 Prezentare generala 7](#_Toc137410821)

[1.2. Standardul OPC UA 7](#_Toc137410822)

[1.3.Conceptele de baza ale standardului OPC UA 8](#_Toc137410823)

[1.4. Serviciile OPC UA 10](#_Toc137410824)

[1.5. Securitatea in OPC UA 10](#_Toc137410825)

[1.6. Interoperabilitatea între diferite platforme și limbaje de programare 11](#_Toc137410826)

[1.7. Studii de caz și exemple de utilizare 12](#_Toc137410827)

[2. Instalare si configurare 13](#_Toc137410828)

[2.1 Dispozitivul Raspberry Pi 13](#_Toc137410829)

[2.2 Arhitectură și specificații tehnice 13](#_Toc137410830)

[2.3. Sisteme de operare și software 14](#_Toc137410831)

[2.4. Instalarea sistemului de operare pe Raspberry PI 15](#_Toc137410832)

[2.5. Conectare si configurare Raspberry Pi la device 17](#_Toc137410833)

[2.6. Introducere VNC Viewer 17](#_Toc137410834)

[2.7. Testare Server OPC UA 19](#_Toc137410835)

[3. Prezentarea Modulelor 22](#_Toc137410836)

[4. Implementare Server OPC UA 22](#_Toc137410837)

[5. Implementare Client OPC UA 22](#_Toc137410838)

[6. Retinere istoric in baza de date 22](#_Toc137410839)

[7. Reprezentarea rezultate in Power BI 22](#_Toc137410840)

# LISTA DE FIGURI

# ACRONIME. ABREVIERI. NOTAŢII

|  |  |  |
| --- | --- | --- |
| 1 | IoT | Internet of Things |
| 2 | OPC | Open Platform Comunications |
| 3 | OPC UA | Open Platform Communications Unified Architecture |
| 4 | OLE | Object Linking and Embedding |
| 5 | DCOM | Distributed Component Object Model |
| 6 | UACP | OPC UA Binary Protocol |
| 7 | SOA | The service oriented architecture |
| 8 | TLS | Transport Layer Security - |

# 1.Introducere

## 1.1 Prezentare generala

IoT reprezinta o tehnologie emergenta care permite conectarea dispozitivelor si obiectelor fizice la internet, permitandu-le sa comunice si sa interactioneze intre ele pentru a oferi beneficii si servicii noi si imbunatatite. Acest document consta in analiza, dezvoltarea si evaluarea unui sistem de achizitie de date ierarhizat, care permite colectarea informatiilor preluate de la senzori analogici si digitali care sunt conectati la o retea de comunicatii.

Intr-un sistem IoT, transferul de date este deosebit de important, deoarece aceste date trebuie sa fie transmise de la dispozitivele de colectare a datelor la serverul central intr-un mod eficient si fiabil. In acest scop, Tehnologia OPC UA este o solutie ideala pentru a face aceasta conexiune si pentru a gestiona datele colectate de senzori si dispozitivele IoT.

In aceasta lucrare, imi propun sa realizez configurarea, implementarea si testarea unui sistem IoT cu server OPC UA, care pot colecta si analiza datele de la senzori si dispozitive, si poate oferi o interfata de utilizare usor de utilizat pentru controlarea si monitorizarea dispozitivelor IoT conectate la retea.

## 1.2. Standardul OPC UA

OPC (Open Platform Communications) reprezintă un set de standarde și specificații dezvoltate în industria automatizării pentru a facilita interoperabilitatea între dispozitivele realizate de producători diferiți. Scopul principal al standardului OPC este de a permite comunicația între sisteme și dispozitive din diferite medii și platforme tehnologice, precum și de a reduce costurile și complexitatea integrării acestora. Acesta a fost dezvoltat de catre compania Microsoft, iar initial standardul a fost dezvoltat sub numele de OLE for Process Control (OPC).

De-a lungul timpului standardul OPC a fost depasit de evolutia tehnologiei, astfel granular s-a facut trecerea la o noua arhitectura numiata OPC UA. Principala schimbare intre OPC si OPC UA a fost protocolul de cominicatie, astfel s-a trecut de la COM/DCOM la TCP/IP pur binar sau alternativ SOAP. Pe lângă multe alte îmbunătățiri, OPC UA acceptă și o descriere semantică a datelor.

Securitatea este un alt motiv important pentru care s-a dezvoltat noua arhitectura, astfel OPC UA include mecanisme de securitate dezvoltate, inclusiv criptare, autentificare, autorizare si audit, astfel asigura o comunicatie sigura intre dispozitivele si sistemele din diferite medii si platforme.

Un alt avantaj al noii arhitecturi este modularitatea si extensibilitatea, astfel se pot adauga cu usurinta noi servicii in viitor si se ofera support pentru mai multe platforme si medii, cum ar fii Windows, Linux si alte sisteme incorporate.

## 1.3.Conceptele de baza ale standardului OPC UA

OPC UA este un standard de comunicare industrială care utilizează modelarea structurilor de date pentru a transmite informații între diferite dispozitive și sisteme. Aceste structuri de date sunt modelate folosind tehnici orientate pe obiect, incluzând ierarhii între tipuri și moșteniri.

Informațiile despre tipurile de date sunt expuse și pot fi accesate în același mod ca și instanțele de date. În terminologia OPC UA, modelul meta folosit pentru modelarea structurilor de date se numește modelul Spațiului de Adrese (Address Space). Prin intermediul modelului Spațiului de Adrese, dispozitivele și sistemele pot interacționa în mod eficient și fiabil, indiferent de tipurile de date și de dispozitivele implicate în comunicare.

Nu există limitări când vine vorba despre cum să fie modelate structurile de date. Pe baza modelului informaţional existent deja în standard se pot defini tipuri care se mapează foarte bine pe modelul real.

Limitările în modelarea structurilor de date în OPC UA sunt minime, deoarece se poate defini o varietate de tipuri de date care se pot mapează bine pe modelul real. Modelarea se bazează pe conceptele de noduri şi referinţe între acestea, iar nodurile pot fi de diferite clase, cum ar fi Object, Variable, Method etc., în funcţie de scopul lor. Toate nodurile au un set de atribute commune:

* NodeId, care identifică în mod unic un nod dintr-un server OPC UA.
* NodeClass, care identifică clasa nodului
* BrowseName, care este numele după care se identifică nodul atunci când se caută prin spaţiul de adrese
* DisplayName, care conţine numele nodului care ar trebui să fie afişat în interfaţa clientului
* Description, care este un atribut opţional prin care se descrie nodul.

Legatura dintre doua noduri este reprezentata de catre referinta, aceasta fiind de mai multe tipuri:

* HasComponent, care indica ca un nod este componenta a altui nod. Acest tip de referinta descrie o relatie de compozitie intre doua noduri, unde un nod este o parte sau un element al altui nod;
* HasProperty, care indica ca un nod are o proprietate asociata. Acest tip de referinta se foloseste pentru a defini o relatie intre un nod si proprietatile sale.
* HasSubtype, care indica ca un nod este un subtip al altui nod. Acest tip de referință se utilizează pentru a stabili o ierarhie de tipuri între noduri.
* HasEventSource, care indica ca un nod este sursa de evenimente. Acest tip de referință se folosește pentru a defini o relație între un nod și evenimentele pe care le poate genera.
* HasNotifier, care indica ca un nod este un modificator de evenimente. Acest tip de referință se utilizează pentru a defini o relație între un nod și nodurile care pot primi notificări despre evenimentele sale.

In figura Fig 1.1 se poate observa legatura dintre noduri, attribute si referinte:

Diagram

Description automatically generated

Fig. 1.1 Noduri, atribute şi referinţe

## 1.4. Serviciile OPC UA

Arhitectura orientata pe servicii (SOA) urmeaza paradigma cerere/raspuns, astfel OPC UA defineste un set fix de servicii cu parametri si comportament exact definiti. Aceste servicii sunt generice. SOA este un tip de proiectare software care se axeaza pe dezvoltarea serviciilor in scopul construirii aplicatiilor. Aceste aplicatii sunt concepute ca o colectie de servicii interoperabile si modularizate, care pot fi accesate si reutilizate intr-un mod standardizat. Serviciile de baza sunt impartite in asa-numitele „seturi de servicii”:

* SecureChannel Service Set, care preia configuratia finala si configuratia de Securitate pentru a stabili o conexiune sigura;
* Session Service Set, care creeaza si administreaza conexiuni specifice utilizatorului intre aplicatii;
* NodeManagement Service Set, care modifica spatiul de adrese al servarului (in cazul in care acest lucru este permis);
* View Service Set, care navigheaza si urmareste referintele (ierarhice) in spatiul de adrese al servarului, cauta si filtreaza informatii;
* Attribute Service Set, care citeste si scrie atributele unui nod, in special atributul valoare, dar si date sau evenimente istorice;
* Method Service Set, invoca metodele pe care un server le ofera la nodurile din spatiul sau de adrese;
* MonitoredItem Service Set, creaza un set de atribute ale nodurilor pentru a fi monitorizate de server si pentru care modificarile ar trebui raportate;
* Subscription Service Set, creaza, modifica sau sterge elementele monitorizate;
* Query Service Set, efectueaza o cautare filtrata pentru informatii in spatiul de adrese al serverului;

## 1.5. Securitatea in OPC UA

OPC UA are doua straturi de securitate incorporate. Primul este stratul de transport(Transport Layer Security - TLS), care stabileste un canal de comunicatie intre server si client, iar pentru acest lucru se foloseste de criptare, semnaturi si certificate digitale. Al doilea strat este cel de aplicatie(Application Security) care are rolul de autentifica utilizatorul si verifica permisiunile.

Performanta aplicatiilor poate fi incetinita de toate aceste mecanisme de securitate, astfel OPC UA ofera optiunea utilizatorului de a le activa sau dezactiva in functie de preferinte.

1. Transport Layer Security-asigura confidentialitatea si integritatea datelor transferate intre dispozitivele OPC UA, astfel se utilizeaza criptografie asimetrica si certificate digitale. Identitatea si autenticitatea sunt asigurate si ele de TLS.
2. Application Security- acest strat se asigura autentificarea utilizatorilor si controlul accesului la resursele sistemului. OPC UA utilizeaza un model baza pe roluri, unde fiecare utilizator are atribuit unul sau mai multe roluri cu permisiuni specifice. Autentificarea utilizatorilor poate fi realizata prin diferite mecanisme, cum ar fi utilizare de nume de utilizator de parole, certificare digitale sau alte metode de autentificare acceptate.

Aceste două straturi de securitate în OPC UA lucrează împreună pentru a proteja comunicarea și accesul la resursele sistemului industrial, asigurând un mediu sigur și de încredere pentru transferul de date între dispozitive.

## 1.6. Interoperabilitatea între diferite platforme și limbaje de programare

În contextul OPC UA (OPC Unified Architecture), interoperabilitatea între diferite platforme și limbaje de programare este realizată prin utilizarea standardului OPC UA însuși. OPC UA definește un set comun de specificații și protocoale care permit comunicația și schimbul de date între diferite sisteme OPC UA, indiferent de platformă sau limbaj de programare.

OPC UA utilizează protocoale de transport standard precum TCP/IP sau HTTPS pentru a permite comunicarea între clientul și serverul OPC UA. Acest lucru permite implementarea clientului și serverului OPC UA în diferite limbaje de programare, cum ar fi C/C++, Java, C#, Python și altele.

De asemenea, OPC UA definește un model de date structurat și un mecanism de descriere a informațiilor și serviciilor disponibile într-un sistem OPC UA, numit Address Space (Spațiu de adresare). Acest model comun de date permite clientului și serverului să înțeleagă și să interacționeze cu obiectele și serviciile definite în sistemul OPC UA.

Astfel, prin implementarea corectă a specificațiilor OPC UA și utilizarea bibliotecilor și SDK-urilor OPC UA disponibile pentru diferite platforme și limbaje de programare, se poate asigura interoperabilitatea între diferitele componente și aplicații OPC UA.

## 1.7. Studii de caz și exemple de utilizare

OPC UA (OPC Unified Architecture) este utilizat într-o gamă largă de domenii și aplicații. Iată câteva studii de caz și exemple de utilizare a OPC UA:

1. Integrarea sistemelor industriale: OPC UA este utilizat pentru integrarea și comunicarea între diferite echipamente și sisteme industriale, cum ar fi mașini de prelucrare, roboți, senzori, SCADA (Supervisory Control and Data Acquisition) și sisteme de control industrial. Acesta permite interoperabilitatea între diversele dispozitive și platforme dintr-o fabrică sau sistem industrial.
2. Monitorizarea și controlul clădirilor inteligente: OPC UA poate fi folosit pentru a integra și gestiona diferite sisteme din clădiri inteligente, cum ar fi sistemele de iluminat, climatizare, securitate și gestionarea energiei. Astfel, se poate obține un control centralizat și o monitorizare eficientă a clădirilor și a resurselor acestora.
3. Interconectarea sistemelor de automatizare: OPC UA facilitează integrarea și comunicarea între diversele sisteme de automatizare, cum ar fi PLC-uri (Programmable Logic Controllers), HMI-uri (Human Machine Interfaces), MES (Manufacturing Execution Systems) și ERP (Enterprise Resource Planning). Acest lucru permite schimbul de date în timp real, coordonarea proceselor și îmbunătățirea eficienței operaționale.
4. Monitorizarea și diagnosticarea utilajelor: OPC UA poate fi utilizat pentru a colecta și transmite date de la utilaje și echipamente către sisteme de monitorizare și diagnosticare. Aceasta permite detectarea și diagnosticarea în timp util a defecțiunilor, planificarea întreținerii preventive și optimizarea performanței utilajelor.
5. Interoperabilitatea între sisteme IT și OT: OPC UA facilitează comunicarea și integrarea între sistemele de tehnologie informațională (IT) și cele de tehnologie operațională (OT). Acesta permite transferul sigur și eficient de date și informații între sistemul de gestionare a afacerii și echipamentele sau procesele de producție.

Acestea sunt doar câteva exemple de utilizare a OPC UA în diverse domenii. Cu flexibilitatea sa și abilitatea de a se integra cu diferite platforme și limbaje de programare, OPC UA devine un standard comun în industrie pentru asigurarea interoperabilității și conectivității între sistemele și dispozitivele industriale.

# Instalare si configurare

Pentru realizarea proiectuluis-a ales platforma Raspberry Pi 3 Model B+, iar in acest capitol voi detalia pasii efectuati pentru instalarea si configurarea si testarea acestei platforme. In continuare vor fi prezentate etapele pe care le-am urmat pentru a contrui si testa modelul informational pentru sistemul experimental.

## 2.1 Dispozitivul Raspberry Pi

Raspberry Pi este o serie de calculatoare monocarton, de dimensiuni reduse și de cost redus, care au fost create cu scopul de a aduce calculatoarele și programarea la un nivel accesibil și de a încuraja învățarea și inovația în domeniul tehnologic.

Fiecare Raspberry Pi este o placă de circuit integrat care conține un procesor, memorie, porturi de intrare/ieșire și alte componente necesare pentru a funcționa ca un computer complet. Aceste dispozitive sunt concepute pentru a rula sisteme de operare precum Linux și oferă suport pentru o varietate de limbaje de programare, permițând utilizatorilor să creeze și să ruleze propriile aplicații și proiecte.

## 2.2 Arhitectură și specificații tehnice

In legatura cu aspectele care au in vedere arhitectura modelului Raspberry Pi 3 Model B+ se poate mentiona faptul ca acesta este echipat cu un processor Broadcom BCM2837B0, care este un procesor ARM Cortex-A53 quad-core cu o frecvență de 1.4 GHz. Dispozitivul are 1 GB de memorie LPDDR2 SDRAM și oferă multiple opțiuni de conectivitate, inclusiv Ethernet 10/100/1000 Mbps, Wi-Fi 802.11ac și Bluetooth 4.2/BLE. Are porturi USB 2.0, port Ethernet, port HDMI, jack audio de 3,5 mm și slot pentru card microSD. De asemenea, dispune de un port camera CSI, un port display DSI și 40 de pini GPIO pentru conexiuni externe. Pentru stocare, se folosește un card microSD, iar grafica este asigurată de GPU VideoCore IV cu suport pentru OpenGL ES 2.0. Dispozitivul Raspberry Pi Model 3B+ este alimentat prin intermediul unui cablu micro USB și poate rula diferite sisteme de operare, cum ar fi Raspbian, Ubuntu sau Windows 10 IoT Core. Are dimensiuni compacte de 85 mm x 56 mm x 17 mm.

A close-up of a green circuit board

Description automatically generated with medium confidence

Fig. 2.1 Raspberry Pi 3 model B+

## 2.3. Sisteme de operare și software

Sistemele de operare și software reprezintă elemente cheie în utilizarea dispozitivului Raspberry Pi model 3B+. Acesta este compatibil cu diverse sisteme de operare, iar cel mai recomandat este Raspbian, bazat pe distribuția Debian. Raspbian oferă o interfață grafică intuitivă, un set diversificat de pachete software și unelte de dezvoltare.

În plus, există și alte distribuții Linux disponibile pentru Raspberry Pi, cum ar fi Ubuntu, Fedora sau Arch Linux, care pot fi adaptate la nevoile specifice ale proiectului. Pentru cei interesați de platforma Windows, există opțiunea Windows 10 IoT Core, un sistem de operare specializat pentru dispozitivele IoT.

Raspberry Pi model 3B+ suportă o gamă largă de limbaje de programare, precum Python, C/C++, Node.js și Java. Acestea permit dezvoltarea aplicațiilor și proiectelor pe dispozitiv. Există, de asemenea, mediile de dezvoltare integrate (IDEs) populare, cum ar fi Thonny, Geany sau Eclipse, care facilitează procesul de dezvoltare.

Placa Raspberry Pi, alimentată cu o tensiune de 5V, poate fi utilizată ca un calculator normal. Se pot conecta periferice precum tastatura și mouse-ul, iar de asemenea poate fi conectată la un monitor pentru afișarea mediului de lucru. De obicei, se folosește impreună cu sistemul de operare Raspbian care are o interfață grafică, facilitând astfel interacțiunea cu acesta. Versiunea cea mai nouă a plăcii Raspberry Pi este Raspberry Pi 4, care poate rula sistemul de operare Windows 10 fără prea mari probleme. Toate specificațiile pe care le are placa Raspberry Pi combinate cu avantajele aduse de sistemul de operare Raspbian sunt principalele motive pentru care această platformă a fost alesă pentru a găzdui aplicația Server OPC UA.

## 2.4. Instalarea sistemului de operare pe Raspberry PI

În vederea utilizării plăcii Raspberry Pi, primul pas esențial este instalarea și configurarea sistemului de operare corespunzător. Pentru a realiza acest lucru, am început prin a căuta distribuția oficială a sistemului de operare destinată plăcii Raspberry Pi, iar am ales să descarc Raspberry Pi OS, care este o variantă optimizată și compatibilă cu placa.

Următorul pas important a fost selecționarea sistemului de operare dorit din opțiunile disponibile. După ce am făcut această alegere, am procedat la instalarea sistemului de operare pe cardul microSD. Acest card a fost apoi introdus în slotul corespunzător al plăcii Raspberry Pi, pregătind astfel placa pentru utilizare.

Prin efectuarea acestor pași, am asigurat ca placa Raspberry Pi dispune de sistemul de operare necesar și este pregătită pentru a fi utilizată în diverse aplicații și proiecte.

Pentru a realiza conexiunea la rețeaua Wi-Fi, voi utiliza un cablu Ethernet pentru a conecta Raspberry Pi la același Wi-Fi la care este conectat și laptop-ul meu. Înainte de a scrie sistemul de operare pe cardul microSD al plăcii Raspberry Pi, va trebui să configurez detaliile despre Wi-Fi.

Aceste detalii includ setarea numelui rețelei (hostname) și a parolei. Prin intermediul utilitarului Putty, voi putea să mă conectez la placa Raspberry Pi și să interacționez cu ea. Este important să specific aceste detalii înainte de a instala sistemul de operare, pentru a asigura o conexiune corectă și securizată între placa Raspberry Pi și rețeaua Wi-Fi.

A screenshot of a computer

Description automatically generated with medium confidence

Fig. 2.2 Imaginea Raspberry Pi OS

În situația în care nu am configurat un hostname înainte de a scrie sistemul de operare pe placa Raspberry Pi, putem realiza conexiunea utilizând adresa IP a plăcii. Pentru a identifica adresa IP a acesteia, putem folosi o aplicație numită Advanced IP Scanner. Această aplicație este un scanner de rețea rapid și eficient, caracterizat printr-o interfață ușor de utilizat.

Cu ajutorul Advanced IP Scanner, în doar câteva secunde, putem localiza toate calculatoarele din rețeaua locală, fie că sunt conectate prin fir sau prin Wi-Fi, și putem scana porturile acestora. Prin intermediul acestui program, avem acces facil la diverse resurse de rețea, cum ar fi protocolul HTTP, HTTPS, FTP și foldere partajate. Mai mult decât atât, ne permite să detectăm toate adresele IP disponibile în rețeaua Wi-Fi.

Utilizând această aplicație, putem obține adresa IP a plăcii Raspberry Pi și, astfel, ne va fi posibilă conectarea și administrarea acesteia prin intermediul protocolului SSH, utilizând un client precum PuTTY sau prin alte mijloace corespunzătoare.

A screenshot of a computer

Description automatically generated with medium confidence

Fig. 2.3 Aplicatia Advanced IP Scanner

## 2.5. Conectare si configurare Raspberry Pi la device

Placa Raspberry Pi va fi controlată în mod remote utilizând protocolul SSH. Pentru a realiza conexiunea, am folosit utilitarul PuTTY, selectând opțiunea SSH și introducând hostname-ul configurat anterior. După ce am stabilit conexiunea prin PuTTY, am executat comanda "sudo raspi-config" în linia de comandă pentru a accesa instrumentul de configurare a software-ului Raspberry Pi.

La rularea comenzii menționate, am fost prezentat cu meniul "Raspberry Pi Software Configuration Tool" în consolă, unde am selectat "Interface Options". Apoi, am continuat să selectez opțiunea "VNC" (acces remote grafic utilizând Real VNC) în pasul următor. După finalizarea acestor setări, am utilizat VNC Viewer pentru a accesa interfața grafică a plăcii Raspberry Pi și a o controla într-un mod mai intuitiv.

## 2.6. Introducere VNC Viewer

VNC Viewer este un sistem de partajare a ecranului dezvoltat pentru a permite controlul de la distanță al unui alt computer. Acesta oferă posibilitatea de a utiliza ecranul, tastatura și mouse-ul unui computer de la distanță, ca și cum am fi în fața sa fizic.

Funcționând pe modelul client-server, VNC Viewer are o componentă server instalată pe computerul remote (cel care va fi controlat), iar VNC Viewer, sau clientul, este instalat pe dispozitivul de pe care dorim să efectuăm controlul (acesta poate fi un alt computer, o tabletă sau un telefon mobil). Atunci când serverul și Viewer-ul sunt conectate, serverul transmite o copie a ecranului computerului remote. Protocolul Remote FrameBuffer (RFB) guvernează formatul datelor care circulă între client și server în sistemul VNC, permițând astfel vizualizarea și controlul computerului remote de la distanță.

VNC Viewer este compatibil cu sisteme de operare precum Windows, macOS, Linux și altele. RFB a început ca un protocol simplu, dar a fost îmbunătățit pe parcurs, adăugând funcții cum ar fi transferul de fișiere, compresia avansată și măsuri de securitate puternice. Compatibilitatea transparentă între diversele clienți și servere VNC este posibilă datorită capacității lor de a negocia o conexiune care utilizează cea mai bună versiune a protocolului RFB, precum și opțiuni de securitate și compresie suportate de ambele părți.

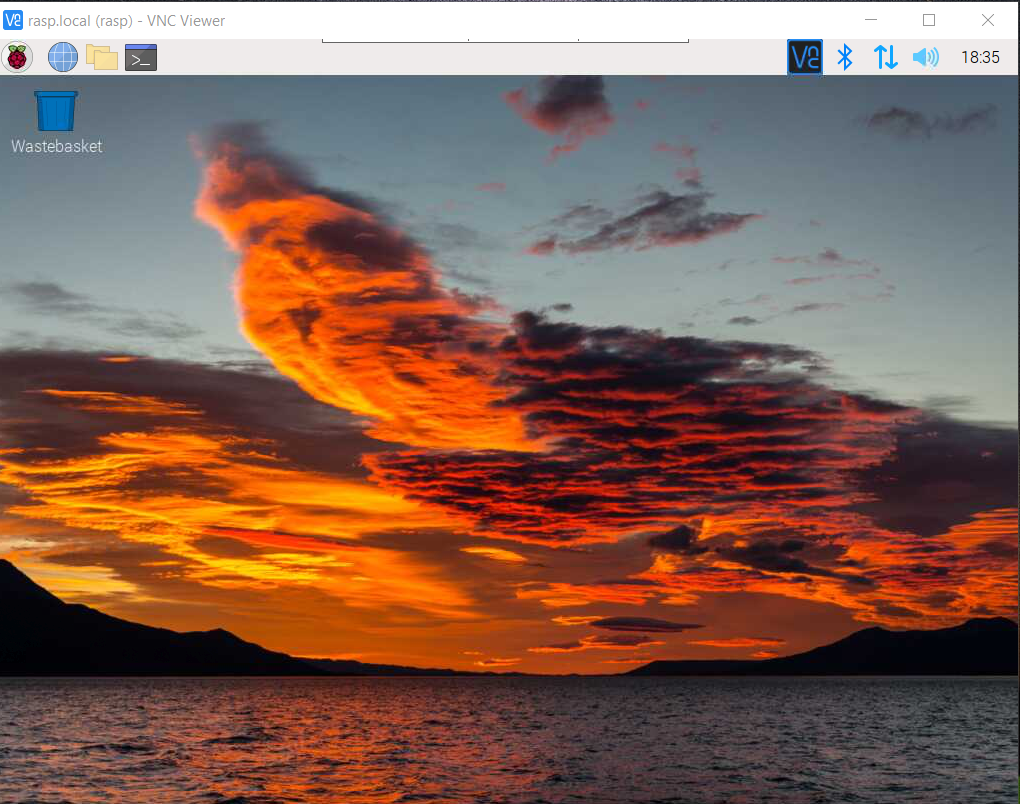


Fig. 2.4 Aplicatia VNC

## 2.7. Testare Server OPC UA

În continuare, am efectuat o testare de tip server-client pentru a simula receptarea și colectarea datelor de la un server, simulând datele înregistrate de un senzor de umiditate și temperatură.

În primul rând, am configurat server-ul pe placa Raspberry Pi, utilizând interfața VNC Viewer. Pentru a actualiza "mașina virtuală", am rulat comanda "sudo apt-get update". Apoi, am executat comanda "sudo apt-get install libxml2-dev libxmlsec1-dev libffi-dev" pentru instalarea unor biblioteci necesare. Analizând comanda de mai sus, putem identifica termeni precum "libxml2-dev", "libxmlsec1-dev" și "libffi-dev". Biblioteca "libxml2-dev" se ocupă de parsarea documentelor XML, în timp ce "libxmlsec1-dev" este o bibliotecă standard în limbajul C care include funcții pentru semnarea și criptarea fișierelor XML, asigurând securitatea acestora. "Libffi-dev" oferă o interfață de programare portabilă și înalt nivel pentru diverse convenții de apelare.

După finalizarea instalării acestor biblioteci, am rulat comanda "sudo pip3 install cryptography" pentru a instala biblioteca "cryptography", conform denumirii acesteia. În continuare, am rulat comanda "sudo pip install freeopcua" pentru a instala tool-ul pentru server-ul OPC UA, pe care îl vom utiliza pe placuța Raspberry Pi.

Prin aceste etape, am pregătit placa Raspberry Pi pentru a funcționa ca server OPC UA, permitând receptarea și colectarea datelor de la senzorul simulat.

Ulterior, trebuie să obținem adresa IP a plăcii Raspberry Pi. Prin intermediul comenzii "ipconfig" sau "ifconfig", putem determina adresa IP, în acest caz fiind: 169.254.15.16. Această adresă IP va trebui înlocuită în codul serverului.

Pentru a transfera codul serverului pe placa Raspberry Pi, folosesc aplicația WinSCP. Interfața grafică a aplicației este împărțită în două părți. Partea stângă este destinată celui care trimite fișierele sau aplicațiile, iar partea dreaptă este destinată celui care primește aceste fișiere sau aplicații.

Partea stângă a aplicației este deja conectată direct la laptop, astfel nu este necesară nicio modificare. În schimb, în partea dreaptă, trebuie să realizăm conexiunea cu placa Raspberry Pi prin intermediul adresei IP, a numelui de utilizator și a parolei setate la momentul scrierii sistemului de operare pe cardul microSD al plăcii.

A screenshot of a computer

Description automatically generated

În ceea ce privește partea de cod, am utilizat limbajul de programare Python. Am declarat trei parametrii, respectiv presiune, temperatură și umiditate. Acești parametrii sunt transmiși în mod continuu către client într-o buclă infinită, atâta timp cât există o conexiune stabilă. De exemplu, pentru a declara parametrul temperatură, am folosit următoarea sintaxă:

Temp = Param.add\_variable(addspace, "Temperature", 0)

De asemenea, am declarat și parametrii pentru presiune și timp folosind sintaxa similară. Acești parametrii vor fi actualizați constant în cod și trimiși către client pentru a fi procesați sau afișați corespunzător.

Este important de menționat că aceasta este o parte din codul serverului și există și alte aspecte și funcționalități care pot fi implementate într-un proiect mai complex

A computer screen shot of a program

Description automatically generated with low confidence

Fig. 2.5 Rularea serverului OPC UA

În continuare, pentru a vizualiza valorile transmise către client, am utilizat aplicația UaExpert. Prin intermediul acestei aplicații, am adăugat placa Raspberry Pi folosind adresa sa IP pentru a stabili conexiunea.

După adăugarea plăcii în aplicația UaExpert, am putut observa în secțiunea "Parameters" parametrii transmiși de către server-ul nostru. Acești parametrii pot include informații precum presiunea, temperatura, umiditatea sau alte măsurători pe care le-am definit în codul serverului.

Prin vizualizarea acestor parametrii în aplicația UaExpert, avem posibilitatea de a monitoriza și analiza în timp real valorile transmise de pe placa Raspberry Pi către client, oferind astfel o perspectivă asupra stării sistemului și a datelor colectate.

A screenshot of a computer

Description automatically generated

Fig. 2.5 Aplicatia UaExpert cu parametrii servarului

Cu ajutorul funcției de drag and drop, am plasat parametrii de temperatură, presiune și timp într-o poziție adecvată în cadrul interfeței aplicației. Astfel, am asigurat că acești parametrii sunt vizibili și ușor accesibili pentru utilizator.

Prin simpla acțiune de tragere și plasare a parametrilor în locația dorită în interfața aplicației, am organizat informațiile într-un mod clar și intuitiv. Acest lucru facilitează monitorizarea și urmărirea valorilor de temperatură, presiune și timp în timp real, oferind o perspectivă rapidă asupra datelor colectate de placa Raspberry Pi.

Astfel, utilizatorul poate avea o experiență mai plăcută și eficientă în vizualizarea și interpretarea datelor furnizate de placa Raspberry Pi prin intermediul aplicației UaExpert.

A screenshot of a computer

Description automatically generated with medium confidence

Fig. 2.5 Aplicatia UaExpert cu valorile parametrilor

### 

# Prezentarea Modulelor

# Implementare Server OPC UA

# Implementare Client OPC UA

# Retinere istoric in baza de date

# Reprezentarea rezultate in Power BI